

# **WordPress-alustalle rakennetun verkkosivuston nopeuttaminen**

Tuomo Räisänen

Opinnäytetyö

Toukokuu 2020

Liiketalous

Tradenomi (AMK), tietojenkäsittelyn tutkinto-ohjelma

Tekijä(t) Räisänen, Tuomo	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Toukokuu 2020
	Sivumäärä 42	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi <b>WordPress-alustalle rakennetun verkkosivuston nopeuttaminen</b>		
Tutkinto-ohjelma Tietojenkäsittelyn tutkinto-ohjelma		
Työn ohjaaja(t) Tuikka, Tommi		
Toimeksiantaja(t) Saksola, Arto		
Tiivistelmä <p>Työssä tarkastellaan verkkosivujen nopeuden vaikutuksia, verkkosivuilla käytettävien resurssien vaikutuksia sivujen nopeuteen ja etsitään keinoja verkkosivujen nopeuttamiseen. Opinnäytetyön tavoitteena oli etsiä keinoja WordPress-alustalle rakennetun verkkosivun nopeuttamiseksi ja toteuttaa yrityksen asiakkaan verkkosivujen nopeuttaminen löydettävien keinojen avulla. Nopeuttamisella pyrittiin parantamaan sivuston käyttökokemusta. Hitaat verkkosivut heikentävät sivujen käyttökokemusta, karkottavat käyttäjät ja voivat heikentää hakukonenäkyvyyttä.</p> <p>Opinnäytetyö toteutettiin kehittämistutkimuksena yrityksen asiakkaalle. Nykyisten verkkosivujen nopeus mitattiin sekä nykyisellä palvelimella, että uudella palvelimella, jonne uusi sivusto tulisi sijoittumaan. Uusi sivusto päätettiin rakentaa, koska nykyisen sivuston tekninen toteutus ei enää ollut riittävä, eikä se olisi sellaisenaan enää palvellut nykyisiä tai tulevia tarpeita. Verkkosivut rakennettiin kokonaan uudelleen etsien samalla keinoja nopeuttaa sivustoa. Optimoinnit tehtiin löydösten perusteella. Tulokset mitattiin käyttäen Googlen PageSpeed Insights ja Pingdom Website Speed Test -nopeusmittareita.</p> <p>Verkkosivujen optimoinnit pienensivät sivuston kokoa ja lyhensivät sivun lataamiseen kuluva aikaa merkittävästi. Muita hyötyjä olivat prosessorin ensimmäisen toimitusjakson lyhentyminen, interaktiivisuutta edeltävän ajan lyheneminen sekä suurimman mahdollisen ensimmäisen toiminnon viiveen lyhentyminen. Näiden lisäksi opinnäytetyössä esitellään aiheeseen liittyviä tärkeitä termejä ja työkaluja optimoinnin tekemiseen.</p> <p>Tuloksista hyötyvät henkilöt, jotka haluavat parantaa WordPress-alustalla olevien verkkosivujen nopeutta, henkilöt, joille termit eivät ole vielä tuttuja sekä saman aihepiirin opinnäytetyöntekijät.</p>		
Avainsanat (asiasanat) WordPress, verkkosivut, optimointi		
Muut tiedot (Salassa pidettävät liitteet)		

Author(s) Räisänen, Tuomo	Type of publication Bachelor's thesis	Date May 2020 Language of publication: Finnish
	Number of pages 42	Permission for web publication: x
Title of publication <b>Ways to speed up a website built on the WordPress platform</b>		
Degree programme Bachelor's degree programme in Business Information Technology		
Supervisor(s) Tuikka, Tommi		
Assigned by Saksola, Arto		
Abstract  <p>The work examines the effects of website speed, the effects of resources used on websites to the page speed, and seeks ways to speed up websites. The objectives of the thesis were to look for ways to speed up a website built on the WordPress platform and to implement the found means to the company's customer's website. Speeding up the website would aim to improve the user experience of the site. Slow websites drives away customers and reduces search engine visibility.</p> <p>The thesis was implemented as a development study. The speed of existing website was measured on the current web server, as well as the new web server that the new website would be built on. The new site was built because the current technical implementation of the current site was not technically up-to-date. The website was completely rebuilt, searching for ways to speed up the site at the same time. The optimizations made were based on findings. Results were measured using Google PageSpeed Insights and Pingdom Site Speed Test speed testing tools.</p> <p>Website optimizations reduced the size of the site and shortened the time it took to load the page. Other benefits were shortening the time at which the page's main thread is quiet enough to handle input, shortening the amount of time it takes for the page to become fully interactive, - as well as shortening the maximum potential First Input Delay. In addition, the thesis introduces important terms related to the topic and tools for performing optimization.</p> <p>The results will benefit people who want to improve the speed of web pages on the WordPress platform, people who are not yet familiar with the terms, and people working on the same topic.</p>		
Keywords/tags (subjects) WordPress, website, optimization		
Miscellaneous (Confidential information)		

## Sisältö

<b>Käsitteet .....</b>	<b>8</b>
<b>1 Johdanto .....</b>	<b>10</b>
<b>2 Tutkimusasetelma .....</b>	<b>11</b>
2.1 Opinnäytetyön tavoitteet ja rajaukset .....	11
2.2 Tutkimusmenetelmät .....	13
<b>3 Teoreettinen viitekehys .....</b>	<b>14</b>
3.1 Miksi verkkosivujen nopeus on tärkeää .....	14
3.1.1 Käyttäjien säilyttäminen .....	15
3.2 Sijoitus hakutuloksissa .....	16
<b>4 Selaimen käyttämät resurssit ja niiden lataaminen .....</b>	<b>16</b>
4.1 HTML-dokumentin käsittely .....	17
4.2 Critical Rendering Path (CRP) .....	18
4.2.1 Document Object Model (DOM) .....	19
4.2.2 CSS Object Model .....	20
4.2.3 Render Tree, Layout ja Paint .....	23
<b>5 Optimointi .....</b>	<b>24</b>
5.1 Resurssien minifointi .....	24
5.2 CSS .....	24
5.3 JavaScript .....	26
5.4 Kuvatiedostot .....	27
5.4.1 Lazyload .....	28
5.5 Välimuisti .....	28
<b>6 Tutkimuksen toteutus .....</b>	<b>29</b>
6.1 Toteutuksessa esiintyvät käsitteet .....	29

	5
6.2 Alkutilanne.....	30
6.3 Sivuston siirto uudelle palvelimelle.....	34
6.4 Uuden sivuston mittaukset .....	37
6.5 Lisäosat .....	40
6.6 Kuvien optimointi ja välimuistin käyttöönotto .....	40
6.7 HTML, CSS ja JavaScript –tiedostojen minifointi.....	43
6.8 CSS ja Javascript –tiedostojen optimointi .....	46
<b>7 Tulokset .....</b>	<b>49</b>
<b>8 Pohdinta.....</b>	<b>51</b>
<b>Lähteet .....</b>	<b>53</b>

## **Kuviot**

Kuvio 1. Suomalaisten internetin käytön yleisyys ja käyttö eri laitteilla 2019. ....	12
Kuvio 2. Experiment Impact on Daily Searches Per User. ....	15
Kuvio 9. Critical Rendering Path. ....	18
Kuvio 3. HTML-tiedoston sisältöä. ....	19
Kuvio 4. Document Object Model -prosessi. ....	20
Kuvio 5. CSS-tiedoston sisältöä.....	21
Kuvio 6. CSS Object Model -prosessi. ....	21
Kuvio 7. Osa CSS Object Model –puuta. ....	22
Kuvio 8. DOM ja CSSOM yhdistettynä renderöintipuuksi. ....	23
Kuvio 10. Esimerkki media query -säännöstä.....	25
Kuvio 11. Esimerkki media query -määrittämisestä link-tagin sisällä. ....	25
Kuvio 12. Async-avainsana käytössä.....	27
Kuvio 13. Yhteenveto sivusta PageSpeed Insight –työkalua käyttäen. ....	31
Kuvio 14. Laboratoriodata sivusta. ....	31

Kuvio 15. Yhteenveto sivusta Pingdom Pingdom Website Speed Test –työkalua käyttäen. ....	32
Kuvio 16. Resurssien koko tyyppin mukaan Pingdom-työkalussa. ....	33
Kuvio 17. HTTP-pyyntöt resurssityypin mukaan Pingdom-työkalussa. ....	34
Kuvio 18. Sivun siirrettynä uudelle palvelimelle. Mittaustulokset Pingdomin testiä käyttäen. ....	35
Kuvio 19. Sivun siirrettynä uudelle palvelimelle. Resurssien koko tyyppin mukaan Pingdom-testissä. ....	36
Kuvio 20. Sivun siirrettynä uudelle palvelimelle. HTTP-pyyntöt resurssityypin mukaan Pingdom-testissä. ....	37
Kuvio 21. Uusien sivujen laboratoriodata PageSpeed-testissä. ....	38
Kuvio 22. Uuden sivuston yleiskuva mitaustuloksista Pingdom-testissä. ....	38
Kuvio 23. Uuden sivuston resurssien koko Pingdom-testissä. ....	39
Kuvio 24. Uuden sivuston HTTP-pyyntöt Pingdom-testissä. ....	39
Kuvio 25. Kuvien optimoinnin ja välimuistin vaikutus PageSpeed-työkalussa. ....	41
Kuvio 26. Kuvien optimoinnin ja välimuistin vaikutus Pingdom-työkalun yhteenvedossa. ....	41
Kuvio 27. Kuvien optimoinnin ja välimuistin vaikutus resurssien kokoon Pingdom-testissä. ....	42
Kuvio 28. Kuvien optimoinnin ja välimuistin vaikutus HTTP-pyyntöjen määrään Pingdom-testissä. ....	43
Kuvio 29. HTML, CSS ja JavaScript -tiedostojen minifoinnin vaikutus PageSpeed-testissä. ....	44
Kuvio 30. Yhteenveto HTML, CSS ja JavaScript -tiedostojen minifoinnin vaikutuksesta Pingdom-testissä. ....	44
Kuvio 31. HTML, CSS ja JavaScript -tiedostojen minifoinnin vaikutus resurssien kokoon. ....	45
Kuvio 32. HTML, CSS ja JavaScript -tiedostojen minifoinnin vaikutus HTTP-pyyntöjen määrään. ....	45

Kuvio 33. CSS ja Javascript-tiedostojen optimoinnin vaikutus PageSpeed-testissä. .....	47
Kuvio 34. CSS ja Javascript –tiedostojen optimoinnin vaikutus Pingdom-testissä. .....	47
Kuvio 35. CSS ja Javascript –tiedostojen optimoinnin vaikutus resurssien kokoon. .....	48
Kuvio 36. CSS ja Javascript –tiedostojen optimoinnin vaikutus HTTP-pyyntöjen määrään.....	48
Kuvio 37. Tulos PageSpeed-testissä optimointien jälkeen.....	50
Kuvio 38. Laboratoriodata PageSpeed-testissä optimointien jälkeen. ....	50
Kuvio 39. Yhteenveto Pingdom-testissä optimointien jälkeen. ....	51

## Taulukot

Taulukko 1. Tulokset.....	51
---------------------------	----

## Käsitteet

### CRP

Lyhenne sanoista Critical Rendering Path eli vaiheiden sarja, joiden aikana selain kääntää HTML, CSS ja JavaScript -tiedostot pikseleiksi tietokoneen ruudulle.

### CSSOM

Lyhenne sanoista CSS Object Model

### DNS

Lyhenne sanoista Domain Name System. Internetin nimipalvelujärjestelmä, joka muuntaa verkkotunnuksia IP-osoitteiksi.

### DOM

Lyhenne sanoista Document Object Model

### HTML

Lyhenne sanoista HyperText Markup Language

### SEO

Lyhenne sanoista Search Engine Optimization. Tapa, jolla pyritään parantamaan verkkosivuille tulevan liikenteen laatua ja määrää sekä parantamaan oman brändin tai tuotteen näkymistä orgaanisissa hakutuloksissa.

### TCP

Lyhenne sanoista Transmission Control Protocol. Kyseessä on tietoliikenneprotokolla tietokoneiden väliseen luotettavaan tiedonsiirtoon.



**TLS**

Lyhenne sanoista Transport Layer Security. Salausprotokolla, jolla voidaan suojata Internet-sovellusten tietoliikenne IP-verkkojen yli.

**WordPress**

Avoimeen lähdekoodiin perustuva www-sisällönhallintaohjelmisto.

# 1 Johdanto

Hitaat verkkosivut karkottavat käyttäjät, huonontavat sivuston käyttökokemusta sekä sijoitusta hakukonetuloksissa. Verkkosivujen nopeuden optimointi on tärkeässä asemassa, jotta yrityksen sivut pystyisivät palvelemaan käyttötarkoitusta mahdollisimman hyvin. Tämän vuoksi kehittäjän on hyvä osata sivustojen optimointi.

Opinnäytetyön aiheena on verkkosivujen nopeuttaminen. Kehittämistutkimuksessa etsittiin keinoja nopeuttaa vanhoja verkkosivuja. Työssä tarkasteltiin yrityksen nykyisiä verkkosivuja, keinoja niiden päivittämiseen ja nopeuttamiseen. Työn aikana huomattiin, ettei nykyisiä sivuja kannattanut lähteä kehittämään pidemmälle vaan oli parempi ratkaisu rakentaa täysin uudet sivut. Uusien sivujen rakentamisen myötä tutkittiin keinoja optimoida sivuston nopeutta. Työ keskittyy vain sivujen optimointiin, ei niiden rakentamiseen.

Tutkimustuloksista saadaan hyötyä asiakasyritykselle paremmin toimivien sivujen muodossa näin parantaen sivuston käyttökokemusta ja tulevaisuuden kehitystarpeita. Lisäksi vahvistetaan omaa ammatillista tietotaitoa sekä löydetään mahdollisesti uusia työkaluja tuleville projekteille. Lopputuloksena saatiin kattava termien esittely, erilaisia työkaluja mitata sivuston nopeutta ja työkaluja sivuston optimointiin.

## 2 Tutkimusasetelma

### 2.1 Opinnäytetyön tavoitteet ja rajaukset

Tämä opinnäytetyö tehdään kehittämistyönä jyvaskyläläiselle ICT-alan yritykselle Tri-media Oy:lle, jonka ydintoimintana ovat yritysilmien ja ulkoasujen suunnittelu, käyttöliittymäsuunnittelu, verkkosivujen toteutus sekä muut räätälöidyt verkkopalvelut. Tri-media Oy:n asiakkaalla – keski-suomalaisella pk-yrityksellä – on WordPress-julkaisualustalle rakennettu verkkosivusto, jonka latausnopeutta halutaan parantaa ja samalla tehdä ulkoasuun päivityksiä. Latausnopeutta parantamalla pyritään parantamaan sivustolla vierailevien käyttäjäkokemusta ja välttämään Googlen antamia rangaistuksia hitaille verkkosivustoille. Rangaistuksena on hakukonenäkyvyyden heikkeneminen, jos sivuston latausnopeudet ovat hitaita (Phan & Wang 2018).

Opinnäytetyön tarkoituksena on selvittää, mitä keinoja on parantaa WordPress-julkaisualustan päälle rakennetun sivuston latausnopeutta ja samalla parantaa sivuston saamia tuloksia latausnopeusmittareissa. Opinnäytetyön tavoitteena on nostaa sivuston nopeus mahdollisimman nopeaksi täten parantaen käyttäjäkokemusta. Mittareina käytetään Googlen PageSpeed Insights -työkalua sekä Pingdom Website Speed Test -työkalua.

Koska suomalaiset käyttävät internetiä yleisimmin matkapuhelimella (kuvio 1), keskittään mittauksissa vain mobiilipuolen tuloksiin, eikä pöytäkoneversion tuloksia käsitellä (Tilastokeskus 2019).

	Käyttää internetiä <sup>1)</sup>	Käyttää internetiä useasti päivässä	Käyttää internetiä				
			Matkapuhelimella <sup>1)</sup>	Kannettavalla tietokoneella <sup>1)</sup>	Pöytäkoneella <sup>1)</sup>	Tabletilla <sup>1)</sup>	
	% -osuus väestöstä						
16-24	100	97	98	80	44	40	
25-34	100	96	98	76	48	43	
35-44	100	97	97	79	40	62	
45-54	98	90	93	74	35	52	
55-64	96	82	84	71	38	54	
65-74	80	57	54	50	27	35	
75-89	41	23	22	21	14	11	
Miehet	90	80	80	66	41	45	
Naiset	89	78	79	66	31	43	
Kaikki	90	79	80	66	36	44	

Kuvio 1. Suomalaisten internetin käytön yleisyys ja käyttö eri laitteilla 2019 (Tilastokeskus 2019)

Googlen PageSpeed Insight -työkalua käytettäessä mittaustulosten pisteytykset on määritelty seuraavasti:

- 90 – 100: sivusto on nopea
- 50 – 89: sivusto on keskitasoa
- 0 – 49: sivusto on hidas.

Nykyisen sivuston mittaustulokset ovat PageSpeedin mukaan keskitasoa: 60/100.

Pingdom Website Speed Test -työkalulla mittaustulokseksi saadaan C-luokka, joka myös on keskitasoa. Pingdomin latausnopeusmittarissa pisteytykset ovat välillä A (parempi) – F (huonompi). Suorituskykyä ei kuitenkaan suoraa seurata mittareiden antaman pisteytyksen mukaan, vaan varsinainen latausaika ja sivuston koon optimointi on tavoitteena.

Opinnäytetyössä keskitytään verkkoselaimessa ajettavan koodin parannuksiin. Palvelinpuolen parannuksia ei käsitellä, vaikka ne vaikuttavatkin selainpuolen toimintaan. Palvelimella tapahtuva koodin käsittely on jätetty työstä pois, koska halutaan mahdollisimman syvällisesti ymmärtää selainpuolen toimintaa. Sivusto myös päädyttiin rakentamaan kokonaan uusiksi ja koska opinnäytetyössä halutaan keskittyä valmiin sivun optimointiin, ei sivun rakentamista käsitellä.

## 2.2 Tutkimusmenetelmät

Opinnäytetyön tutkimuskysymyksenä on: **Millä keinoilla WordPress-julkaisualustalle rakennettua verkkosivustoa voidaan nopeuttaa?**

Opinnäytetyössä etsitään keinoja nopeuttaa asiakkaan verkkosivustoa. Opinnäytetyössä tutkitaan yksittäistä tapausta ja sen prosesseja, tehdään analysointia jo aineistoa kerätessä sekä pyritään ymmärtämään syvällisesti tutkittavaa ilmiötä. Tämän vuoksi tutkimusote on kvalitatiivinen eli laadullinen (Kananen, 2008, 24–25).

Työssä määritellään ongelma ja etsitään vaihtoehtoisia ratkaisuja kehittämistutkimuksen avulla. Tutkimus alkaa nykytilan kartoituksella ja ongelman määrittelyllä. Sen jälkeen edetään vaihtoehtojen etsintään ja arvioidaan vaihtoehtoiset ratkaisut. Ratkaisujen arvioinnin jälkeen edetään kokeiluun ja arvioidaan kokeilun tulokset. Työ etenee suunnitelmallisesti ja seuraa Kanasen kehittämistutkimuksen syklien vaiheita. (Kananen, 2015, 39–42.) Kehittämistutkimuksessa suoritetaan sekä ennen että jälkeen mittaukset, jotta tuloksia voidaan kunnolla arvioida. Mittauksissa käytetään erilaisia tilastoja ja numeroita, joten mittauksen tutkimusote on kvantitatiivinen, eli määrällinen (Jyväskylän yliopisto – Määrällinen tutkimus 2015).

### 3 Teorettinen viitekehys

Teoreettisena viitekehystenä käytetään tietoa tutkimuksista, jotka vastaavat kysymyksiin: Miksi verkkosivujen nopeus on tärkeää? Mitkä asiat vaikuttavat verkkosivujen nopeuteen? Mitä yritys hyötyy nopeista verkkosivustoista?

#### 3.1 Miksi verkkosivujen nopeus on tärkeää

Verkkosivustojen jatkuvan toiminnollisuuksien lisääntymisen myötä sivujen suorituskyvyn ylläpitäminen tulee vaikeammaksi. Ongelmien laajuudet vaihtelevat pienistä sivuston käyttöä hidastavista viiveistä suurempiin ongelmiin, jotka voivat estää sivuston käytön kokonaan. (Wagner 2019.)

Google tutki heidän hakukonettaan käyttävien käyttäjien reagoitua hitaaseen sivuston toimintaan hidastamalla hakutuloksien näyttämistä käyttäjille 100 – 400 millisekunnilla. Seurauksena oli 0.2 % - 0.6 % vähemmän hakuja (kuviot 2). Hakujen määrä myös jatkui vähenemistään, mitä pidemmälle kokeilu eteni. (Brutlag 2009.)

Table 1: Experiment Impact on Daily Searches Per User

Type of Delay	Magnitude	Duration	Impact
Pre-header	50 ms	4 weeks	—
Pre-header	100 ms	4 weeks	−0.20%
Post-header	200 ms	6 weeks	−0.29%
Post-header	400 ms	6 weeks	−0.59%
Post-ads	200 ms	4 weeks	−0.30%

Kuvio 2. Experiment Impact on Daily Searches Per User (Brutlag 2009).

Lopputuloksena käyttäjien tekemät hakumäärät putosivat päivittäisellä tasolla 0.5 %, joka tarkoittaisi Googlen päivittäisessä hakumäärässä – 5.5 miljardia (Sullivan 2016) – noin 27 500 000 hakua (Brutlag 2009). Tutkimuksesta voidaan siis päätellä, että hitaammat sivut vähentävät käyttäjien halukkuutta käyttää kyseistä verkkosivustoa.

### 3.1.1 Käyttäjien säilyttäminen

Jos ensikokemus sivustosta on huono, voi olla, ettei käyttäjä pala sivustolle takaisin. Jotta sivuston käyttäjät pysyvät sivustolla ja pystyvät suorittamaan onnistuneesti toiminnot, joita varten sivusto on rakennettu sekä palaavat käyttämään sivustoa uudelleen on sivustojen suorituskyvyn oltava riittävä. Nopeat sivut myös parantavat sivuston konversioastetta. (Wagner 2019.) Konversiolla tarkoitetaan sivuston kävijän tekemää mitattavaa toimenpidettä, kuten vaikka lomakkeen täyttäminen, artikkelin lukeminen tai linkin klikkaus (Koskelo n.d.).

Sivuston hitaalla toiminnalla taas on vastakkainen vaikutus. Esimerkiksi BBC huomasi jokaisen sivun lataamiseen menevän sekunnin lisäävään käyttäjien poistumista sivustolta kymmenellä prosentilla. (Clark 2018.)

### 3.2 Sijoitus hakutuloksissa

Google ilmoitti tammikuussa 2018 ottavansa käyttöön päivityksen, jonka seurauksena hitaasti toimivat nettisivustot sijoittuvat huonommin hakutuloksissa. Tämä uusi päivitys tulisi vaikuttamaan mobiililaitteilla tehtyihin hakuihin ja otettaisiin käyttöön heinäkuun 2018 aikana. (Phan & Wang 2018.) Google oli jo aiemmin – huhtikuussa 2010 (Cutts & Singhal 2010) – ilmoittanut verkkosivujen nopeuden olevan vaikuttava tekijä hakutuloksiin, mutta käytäntö laajentui nyt myös mobiililaitteilla tehtyihin hakuihin (Phan & Wang 2018). Vaikka päivityksen tuomat huonommat hakusijoitukset vaikuttavat vain todella huonosti suoriutuviin sivustoihin, voi tulevaisuudessa tilanne olla toinen.

Korkea sijoitus hakutuloksissa tuo sivustolle enemmän liikennettä ja parantaa oman brändin sekä tuotteiden näkyvyyttä. Näkyvyyttä voi tietysti aina hankkia ostamalla mainostilaa, mutta tämä lisää kuluja suhteessa ilmaiseen orgaaniseen näkyvyyteen, jota hyvä sijoitus hakutuloksissa tuo mukanaan.

Nopeat sivut ovat myös nopeampia indeksoida. Indeksoinnilla tarkoitetaan prosessia, jossa Googlen robotit käyvät sivuston linkit läpi etsien uutta ja päivitettyä sisältöä. Ilman indeksointia, ei linkkiä tai sivua voi löytää hakutuloksista. (Muller & Moz.com staff, n.d.)

## 4 Selaimen käyttämät resurssit ja niiden lataaminen

Verkkosivu koostuu useista eri resurssityypistä ja näistä resursseista monen tehtävänä on parantaa käyttäjän visuaalista kokemusta (Weiss 2018, 238). HTML-resurssi kertoo selaimelle dokumentin rakenteen. CSS-resurssit määrittelevät dokumentin ulkoasun ja JavaScript-resurssit tuovat toiminnallisuutta ja interaktiivisuutta. Lisäksi on vielä erilaiset kirjasintyypit – fontit – joilla pyritään parantamaan lukukokemusta ja



tuomaan visuaalisuutta sivustolle sekä erilaiset kuva-, video- ja ääniresurssit, joilla pyritään parantamaan sivuston asiakontekstia ja visuaalisuutta. (Weiss 2018, 238.)

Kun käyttäjä klikkaa linkkiä tai siirtyy sivustolle, hakee selain HTML-tiedoston ja sen noudettuaan tietää missä muut tarpeelliset resurssit sijaitsevat. Selain aloittaa yhteyden muodostamisprosessin käyttäen nimipalvelinta selvittääkseen mihin IP-osoitteeseen sen on luotava yhteys ja sen jälkeen selain luo TCP ja TLS -protokollia käyttäen yhteyden palvelimelle. Sen jälkeen selain lähettää palvelimelle GET-pyyynnön, johon palvelin vastaa. (Weiss 2018, 238.)

#### 4.1 HTML-dokumentin käsittely

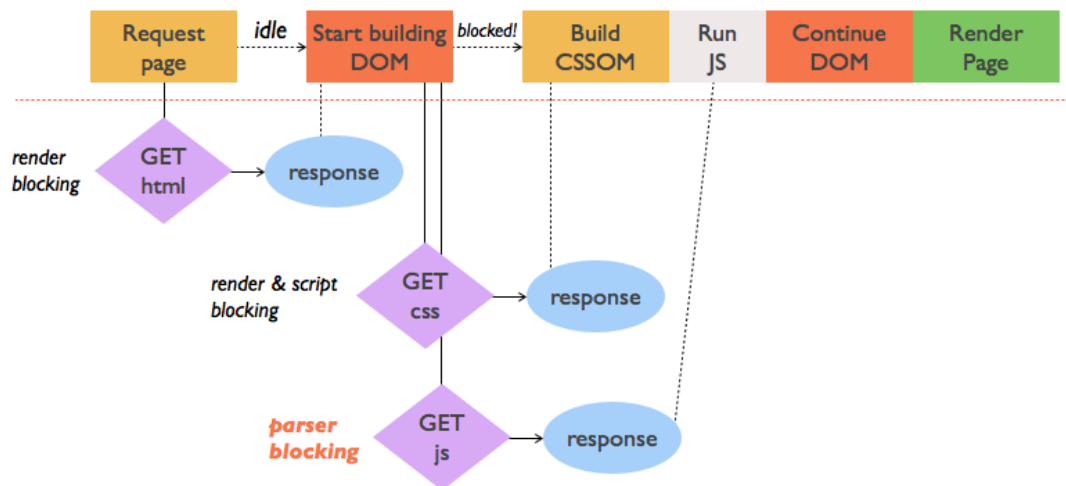
HTML toimii progressiivisesti, eli selaimen ladatessa HTML-dokumentin ei selaimen tarvitse odottaa koko dokumentin latautumista päästäkseen käsittelemään dokumentin sisältöä, vaan dokumentin käsittely alkaa saman tien sen saapuessa, vaikka dokumentin lataaminen olisi vielä kesken. (Weiss 2018, 239.)

Selaimen täytyy käsitellä HTML-dokumentti, jotta siitä pystytään muodostamaan DOM-puu, jonka avulla selain saa selville mitä resursseja tarvitaan piirtämään verkkosivu kokonaisuudessaan ruudulle. HTML:n käsittely alkaa vaiheella, jossa selain pilkkoo HTML-dokumentin tekstijonot tunnisteiksi – tokeneiksi. Tunnisteiden luomisen jälkeen selain pystyy käyttämään niitä käynnistämään resurssien lataamisen. (Weiss 2018, 239.)

Tunnisteiden luonnin jälkeen selaimen esilataaja – preloader – käy tokenit läpi ja selvittää, mitä resursseja todennäköisesti tarvitaan myöhemmin ja käynnistää resurssipyynnöt. Tämän jälkeen selain käy tokenit läpi HTML:n jäsennysääntöjen mukaisesti. Tästä syntyy DOM-silmukoita (nodeja), joista yhdessä muodostuu DOM-puu. (Weiss 2018, 239.)

## 4.2 Critical Rendering Path (CRP)

Critical Rendering Path (kuvio 9) tarkoittaa vaihteita, jotka tapahtuvat selaimen saadessa HTML, CSS ja JavaScript -tavut ja muuntaessa ne renderöidyiksi pikseleiksi ruudulle (Grigorik 2019).



Kuvio 3. Critical Rendering Path (Max 2014).

Verkkosivujen nopeuttaminen tapahtuu optimoimalla CRP:n eri vaihteita, jotta verkkosivu saadaan renderöityä nopeasti ruudulle näkyviin. CRP koostuu viidestä eri vaiheesta – DOM:in ja CSSOM:in rakentaminen, näiden yhdistäminen renderöintipuuksi, asettelu ja maalaaminen –, joita edeltää selaimelta tuleva HTML-sivupyynnö. (Grigorik 2019.)

### 4.2.1 Document Object Model (DOM)

Document Object Model on rajapinta HTML-dokumentille. Selain käyttää sitä määrittämään, mitä ruudulle renderöidään näkyväksi. JavaScript-ohjelmat käyttävät DOMia muokatessaan sivuston sisältöä, rakennetta tai tyyliä. (Grigorik 2019.)

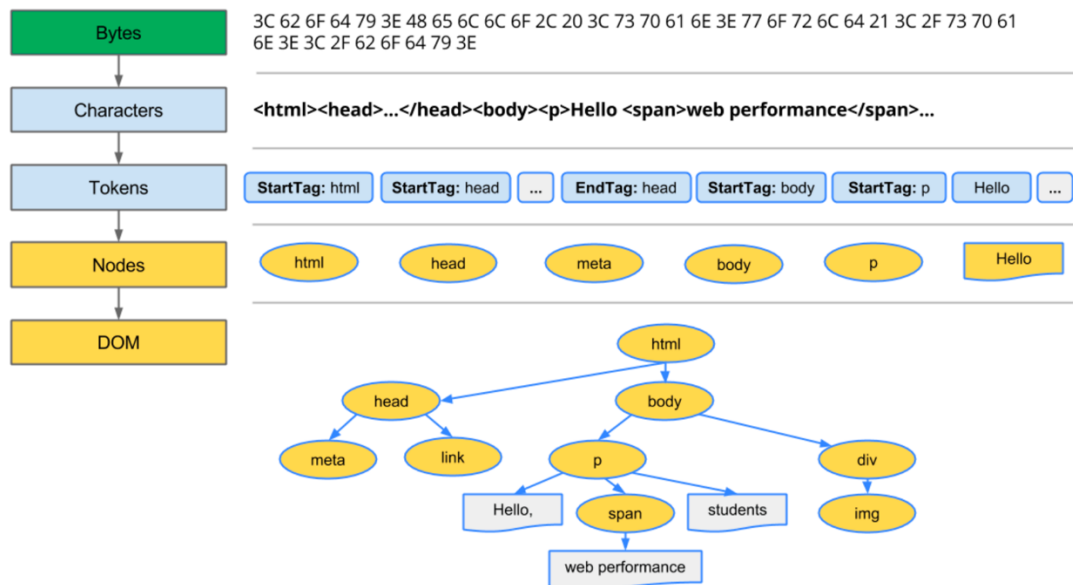
DOM on prosessin lopputulema, joka koostuu neljästä eri vaiheesta. Ensimmäisessä vaiheessa selain lukee HTML-tiedoston tavut ja kääntää ne yksittäisiksi merkeiksi tiedoston määritellyn koodauksen perusteella, esimerkiksi UTF-8 -muotoon. (Grigorik 2019.) HTML-tiedoston sisältöä havainnollistettu kuvassa (kuvio 3).

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <link href="style.css" rel="stylesheet">
    <title>Critical Path</title>
  </head>
  <body>
    <p>Hello <span>web performance</span> students!</p>
    <div></div>
  </body>
</html>
```

Kuvio 4. HTML-tiedoston sisältöä. (Grigorik 2019.)

Tämän jälkeen selain muuntaa merkkijonot erillisiksi tokeneiksi, joilla jokaisella on oma merkityksensä ja sääntönsä, kuten "<html>" ja "<body>" ja muut merkkijonot

kulmasulkeiden sisällä. Tokenit muunnetaan silmukoiksi (nodeiksi), jotka määrittelevät niiden ominaisuudet ja säännöt. Viimeisenä vaiheena DOM rakentuu ja siinä on määritettynä eri objektien väliset suhteet: HTML-objekti on body-objektin vanhempi, joka vuorostaan on div-objektin vanhempi ja niin edelleen. Prosessi (kuvio 4) tapahtuu joka kerta, kun selain käsittelee HTML-koodia. (Grigorik 2019.)



Kuvio 5. Document Object Model -prosessi (Grigorik 2019).

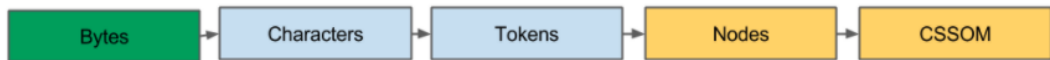
#### 4.2.2 CSS Object Model

CSS Object Model (CSSOM) on joukko rajapintoja, jotka mahdollistavat dokumentin tyylien manipuloinnin JavaScriptin avulla. CCSOM (ks. 7) on hyvin samankaltainen

kuin DOM; toistetaan samoja prosessin (kuvio 6) vaiheita kuin luodessa DOM, mutta HTML-tiedoston sijaan luetaan CSS-tiedostot (kuvio 5). (Grigorik 2019.)

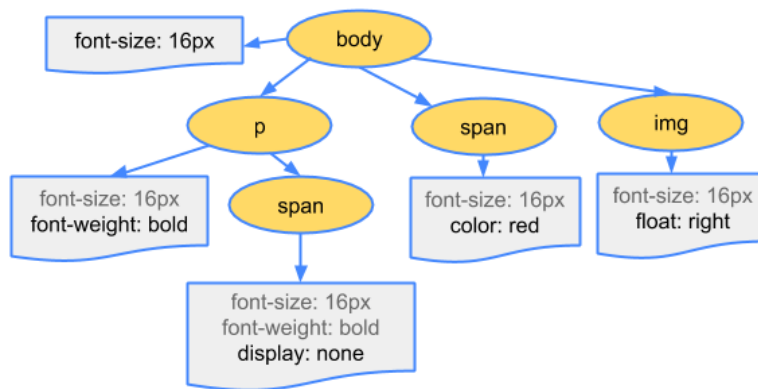
```
body { font-size: 16px }  
p { font-weight: bold }  
span { color: red }  
p span { display: none }  
img { float: right }
```

Kuvio 6. CSS-tiedoston sisältöä (Grigorik 2019).



Kuvio 7. CSS Object Model -prosessi (Grigorik 2019).

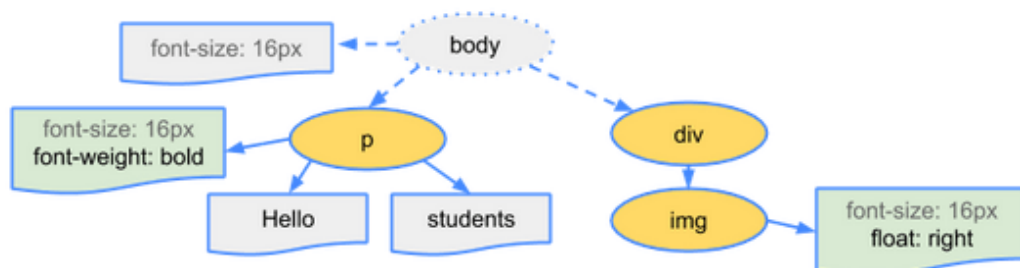
CSS-tiedostossa olevat määrytykset – sekä HTML-koodiin mahdollisesti suoraan kirjoitetut CSS-tyylimäärittelyt – liitetään nodeihin ja tästä muodostuu CSS Object Model (kuvio 7) (Grigorik 2019).



Kuvio 8. Osa CSS Object Model –puuta (Grigorik n.d.).

### 4.2.3 Render Tree, Layout ja Paint

DOM:in ja CSSOM:in puut yhdistetään yhdeksi renderöintipuuksi (kuvio 8), joka sisältää DOM:ista saadut sisältötiedot ja CSSOM:ista saadut tyyli tiedot. Yhdistettyä renderöintipuuta käytetään selainikkunassa näkyvien elementtien asettelun laskentaan ja se myös toimii syötteenä maalausprosessissa (Paint). (Grigorik 2019.)



Kuvio 9. DOM ja CSSOM yhdistettynä renderöintipuuksi (Grigorik 2019).

Asetteluvaiheessa (layout) selain käy renderöintipuun lävitse alusta loppuun ja selvittää jokaisen elementin tarkan koon ja sijainnin sivulla. Tämän kaiken jälkeen - kun selain on tietoinen mitkä nodet näytetään ikkunassa, miltä ne näyttävät, minkä kokoisia ne ovat ja missä ne sijaitsevat – muuntaa maalausvaihe renderöintipuun solmut näytölle pikseleiksi. (Grigorik 2019.)

Renderöintipuun rakentamiseen, asetteluvaiheeseen ja maalaukseen käytettävä aika riippuu HTML-dokumentin koosta, tyyliasetusten määrästä ja monimutkaisuudesta sekä käytettävästä laitteesta (Grigorik 2019).

## 5 Optimointi

Koska on tiedossa, mitä vaiheita tarvitaan sivun tuottamiseksi ja kuinka kriittiset resurssit – HTML, CSS ja Javascript – toimivat keskenään, voidaan pyrkiä optimoimaan CRP:n vaiheita.

### 5.1 Resurssien minifointi

Resurssien minifointi pienentää sivuston kokoa, jolloin myös datasiirtoon tarvittava aika lyhenee, joka taas nopeuttaa sivun renderöintiä. Minifoinnilla saadaan poistettua HTML, CSS ja JavaScript -tiedostoista turhat välilyönnit ja kommentit, jolloin tiedoston koko pienenee. (Grigorik 2019; Max 2014.)

Minifoinnin jälkeen on suositeltavaa vielä pakata kriittiset resurssit käyttäen GZIP-pakkausta. Tiedostojen pakkaaminen pienentää tiedostojen kokoa entisestään ja voidaan saavuttaa jopa 70–90 % pienennys tiedoston koolle. GZIP-pakkaus suoritetaan palvelimella ja on riippuvainen palvelimen asetuksista. Modernit selaimet osaavat käsitellä pakattuja tiedostoja ja pyytävät niitä automaattisesti palvelimelta, ilman että käyttäjän tarvitsee tehdä erityisiä toimia. (Grigorik 2019.)

### 5.2 CSS

Koska CSS estää oletuksena sivun renderöinnin sekä JavaScriptin suorittamisen, on suosituksena listata kaikki CSS-tiedostoihin osoittavat '*link*'-tagit HTML-tiedoston '*head*'-osaan, jotta ne saadaan toimitettua käyttäjälle mahdollisimman nopeasti (Max 2014).

Toinen keino lyhentää aikaa, jonka CSS estää renderöinnin on käyttää media queryja (Max 2014). Media query on CSS-tekniikka, joka käyttää '@*media*'-sääntöä (kuvio 10), jolla voidaan määrittää CSS-tyylejä, jos tietyt ehdot toteutuvat (Responsive Web



Design n.d.).

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: red;  
  }  
}
```

Kuvio 10. Esimerkki media query -säännöstä.

Media query –tekniikkaa käyttämällä voidaan pilkkoa CSS-määrittelyt useampaan tiedostoon, jolloin selain voi käyttää vain niitä tiedostoja joiden ehdot täyttyvät. Lisäämällä '*<link>*' –tagiin media-määrittelyn (kuvio 11) voidaan määrittää, milloin jokin tietty CSS-resurssi on tarpeellinen. (Max 2014; Grigorik 2019.)

```
<link href="large-devices.css" rel="stylesheet" media="(min-width: 992px)">
```

Kuvio 11. Esimerkki media query -määrittämisestä link-tagin sisällä.

Media queryn käyttämisestä huolimatta selain lataa kaikki CSS-resurssit, mutta resurssit joiden ehdot eivät täyty, latautuvat pienemmällä prioriteetilla ja samaan aikaan renderöinnin kanssa (Max 2014; Grigorik 2019).

CSS-määritykset on myös mahdollista lisätä suoraan HTML-tiedoston koodiin, jolloin erillistä CSS-tiedostoa ei tarvitse hakea, mutta määritykset toimivat vain kyseisessä HTML-tiedostossa ja itse HTML-tiedoston koko kasvaa suuremmaksi. Kokonaiskuorma tiedonsiirrolle voi myös kasvaa, jos samoja määrittelyjä joudutaan tekemään useisiin HTML-tiedostoihin sen sijaan, että ne määritettäisiin samaan CSS-tiedostoon. (Max 2014; Grigorik 2019).

### 5.3 JavaScript

Oletuksena JavaScript-ohjelman suorittaminen pysäyttää DOM:in rakentamisen, kunnes ohjelma on suoritettu. JavaScript-ohjelman suorittaminen myös pysähtyy, kunnes CSSOM on valmiina. Tämän vuoksi on tärkeää määrittää, missä kohtaa `<script>`-tagi – jolla osoitetaan ulkoisen JavaScript-resurssin sijainti tai jonka sisälle JavaScript-ohjelma on suoraan tehty - on lisätty HTML-dokumenttiin. (Max 2014; Grigorik 2019.) Jos JavaScript-ohjelma on kirjoitettu *'inline'*, eli ohjelma on suoraan `<script>`-tagin sisällä, pysähtyy DOM:in rakentaminen, kunnes ohjelma on suoritettu (Grigorik 2019).

Jos tagilla taas osoitetaan JavaScript-resurssin sijainti, pysähtyy DOM:in rakentaminen ja selaimen on odotettava, kunnes resurssi on latautunut ja vasta sen jälkeen JavaScript-ohjelma ajetaan, jonka jälkeen DOM:in rakentaminen vasta jatkuu. Tämä voidaan kuitenkin kiertää lisäämällä `<script>`-tagiin *async*-avainsana (kuvio 12), jolla

kerrotaan selaimelle, ettei ohjelmaa tarvitse suorittaa siinä kohtaa, jossa se on merkitty vaan se voidaan suorittaa, kun resurssi on saatu ladattua. (Grigorik 2019.)

```
<script
  src="https://code.jquery.com/jquery-3.4.1.min.js"
  integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFLBw8HfCJo="
  crossorigin="anonymous" async>
</script>
<script src="<?php echo get_template_directory_uri(); ?>/assets/js/carousel.js" async></script>
```

Kuvio 12. Async-avainsana käytössä.

Jos JavaScriptillä ei muokata DOM:ia tai CSSOM:ia on async-avainsanan käyttäminen tällaiselle resurssille suositeltavaa (Max 2014).

## 5.4 Kuvatiedostot

Kuvien optimoinnilla pyritään jakamaan kuvatiedostot mahdollisimman pienessä koossa menettämättä kuvanlaatua. Kuvat voidaan optimoida ennen sivustolle lataamista tai lisäosaa käyttämällä. Kuvia optimoimalla saadaan pienennettyä sivuston koosta ja nopeutettua sivun lataamista. Optimointiin käytetään kahta erilaista tekniikkaa: häviöllinen, jossa kuvasta poistetaan osa datasta ja häviötön, jossa kuvan data pakataan. Kuvat on myös hyvä skaalata käyttötarkoitukseen sopiviksi. (Jackson 2020.)

### 5.4.1 Lazyload

Lazyload on tekniikka, jolla viivästetään ei-kriittisten resurssien lataamista sivulatauksen yhteydessä ja resurssit ladataan vasta kun niitä tarvitaan. Kuvien osalta se tarkoittaa, että kuvat jotka eivät ole ikkunan näkymässä eivät lataudu ollenkaan, tai varsinaisen kuvan sijaan käytetään väliaikaista kuvaa, joka on kooltaan paljon pienempi kuin varsinainen kuva. Kun käyttäjä siirtyy sisällössä kohtaan, jossa kuva pitäisi olla, ladataan varsinainen kuvatiedosto. (Wagner 2019.)

Tekniikan hyötyjä ovat pienemmät datan siirtomäärät, koska resursseja ladataan vasta tarpeen mukaan näkyviin sekä muiden resurssien pienempi käyttö – kuten prosessointiin menevä aika, akun käyttö ja muut laiteresurssit. Tekniikalla saadaan myös nopeutettua sivun ensilataamiseen menevää aikaa. (Wagner 2019.)

## 5.5 Välimuisti

Jokaisen sivulatauksen myötä palvelin saa pyynnön sivusta ja joutuu käsittelemään pyynnön tehden tarpeelliset prosessit ja lähettämään kaikki tarpeelliset tiedostot selaimelle, jotta sivu pystytään näyttämään niin kuin se on tarkoitettu ja tämä tietysti syö resursseja. Välimuistin tarkoituksena on ohjeistaa palvelinta säilömään sivun tiedostoja palvelimen levyille tai keskusmuistiin, josta se osaa sitten jakaa saman sisällön uudelleen. Tämä vähentää tarvittavan työn määrää, joka joudutaan sivun näyttämiseen tekemään ja sivu saadaan ladattua välimuistista nopeammin näkyviin. (How to Speed up Your WordPress Site 2020.)

## 6 Tutkimuksen toteutus

Tutkimus aloitettiin mittaamalla nykyisen sivuston nopeus. Koska nykyinen sivusto sijaitsee eri palvelimella kuin tuleva uusi sivusto, siirrettiin sivusta kopio samalle palvelimelle, kuin uusi sivusto tulee sijaitsemaan. Tämän jälkeen selvitettiin, mitä muita optimointeja on mahdollista tehdä ja lähdettiin tekemään optimointeja sivustolle, samalla seuraten kuinka muutokset vaikuttavat sivuston suorituskykyyn. Löydökset dokumentoitiin ja parhaat keinot jätettiin käytäntöön.

Kehitystyö tehtiin lokaalissa kehitysympäristössä omalta tietokoneelta. Nopeuden mittaamiseen käytettiin eri työkaluja: Googlen PageSpeed Insights ja Pingdom.comin Website Speed Test -työkalua.

### 6.1 Toteutuksessa esiintyvät käsitteet

**Ensimmäinen sisällön renderöinti:** Ensimmäisen sisällön renderöinti tarkoittaa aikaa joka kuluu, jotta käyttäjä näkee jotain ruudulla. Mitä pienempi aika kuluu, sitä nopeammin käyttäjä näkee ruudulla jotain tapahtuvan. (Tools for Web Developers - First Contentful Paint 2020.)

**Ensimmäisen toiminnon viive:** Ensimmäisen toiminnon viive mittaa aikaa joka kuluu, jotta käyttäjä voi tehdä sivustolla jotain, kuten klikata linkkiä ja sivusto vastaa pyydettyyn toimintoon (Tools for Web Developers – First input delay 2020).

**Ensimmäinen merkityksellinen renderöinti:** Ensimmäinen merkityksellinen renderöinti kertoo, milloin sivun ensisijainen sisältö tulee näkyviin (PageSpeed Insights 2020).

**Nopeusindeksi:** Nopeusindeksi kertoo, kuinka nopeasti sivun sisältö tulee näkyviin (PageSpeed Insights 2020).

**CPU:n ensimmäinen toimettomuusjakso:** Prosessorin ensimmäinen toimettomuusjakso kertoo, milloin sivun pääsäikeen tilanne sallii syötteiden käsittelyn (PageSpeed Insights 2020).

**Interaktiivisuutta edeltävä aika:** Interaktiivisuutta edeltävä aika tarkoittaa aikaa, joka sivulla kestää siihen, että se on täysin interaktiivinen (PageSpeed Insights 2020).

**Suurin mahdollinen viive:** Käyttäjän suurin mahdollinen ensimmäisen toiminnon viive on pisimmän tehtävän kesto millisekunneissa (PageSpeed Insights 2020).

**Page size:** Page size ilmaisee sivun resurssien yhteenlasketun määrän.

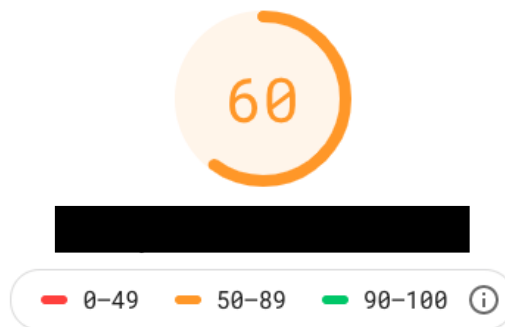
**Load time:** Load time ilmaisee sivun lataamiseen menevän ajan.

**Requests:** Requests näyttää HTTP-pyyntöjen yhteenlasketun määrän.

## 6.2 Alkutilanne

Pelkän nykyisen sivun optimoinnin sijaan päädyttiin rakentamaan sivun toteutus kokonaan uusiksi ja optimoimaan uusi sivusto, koska sivuston nykyinen toteutus on vanhentunut hidastaen sivustoa sekä vaikeutti toteutuksen vuoksi tulevaa kehittämistä ja sivuston päivittämistä.

Mittaustulokset on otettu huhtikuussa 2020. Kuvioissa 13 – 17 esitellään mittaustuloksia nykyisen sivuston nopeudesta.



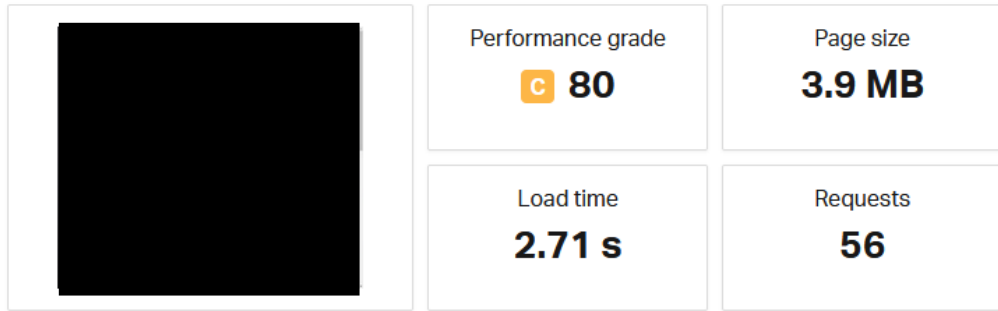
Kuvio 13. Yhteenveto sivusta PageSpeed Insight –työkalua käyttäen (PageSpeed Insights 2020).

**Laboratoriodata** ☰ ☰

■ Ensimmäinen sisällön renderöinti	3,9 s	▲ Ensimmäinen merkityksellinen renderöinti	4,1 s
▲ Nopeusindeksi	7,2 s	■ CPU:n ensimmäinen toimitusjakso	4,1 s
■ Interaktiivisuutta edeltävä aika	4,8 s	● Suurin mahdollinen ensimmäisen toiminnon viive	120 ms

Kuvio 14. Laboratoriodata sivusta (PageSpeed Insights 2020).








## Your Results:

[↓ DOWNLOAD HAR](#)[↑ SHARE RESULT](#)

Kuvio 15. Yhteenveto sivusta Pingdom Pingdom Website Speed Test –työkalua käyttäen (Pingdom Website Speed Test 2020).










## Content size by content type

CONTENT TYPE	PERCENT	SIZE
 Image	91.29%	3.6 MB
 Script	3.96%	156.0 KB
 CSS	2.04%	80.4 KB
 Font	1.76%	69.5 KB
 XHR	0.66%	26.0 KB
 HTML	0.28%	11.1 KB
 Redirect	0.01%	310.0 B
Total	100.00%	3.9 MB

Kuvio 16. Resurssien koko tyyppin mukaan Pingdom-työkalussa.

## Requests by content type

CONTENT TYPE	PERCENT	REQUESTS
 Image	52.73%	29
 Script	21.82%	12
 CSS	9.09%	5
 XHR	7.27%	4
 Font	5.45%	3
 Redirect	1.82%	1
 HTML	1.82%	1
Total	100.00%	55

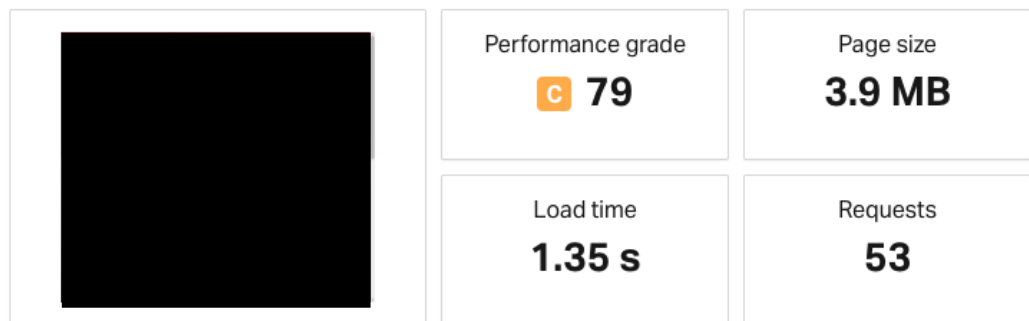
Kuvio 17. HTTP-pyynnöt resurssityypin mukaan Pingdom-työkalussa.

### 6.3 Sivuston siirto uudelle palvelimelle

Nykyinen sivusto sijaitsee eri palvelimella kuin tuleva uudistettu sivusto, joten luotiin nykyisestä sivustosta kopio uudelle palvelimelle. Näin palvelimien eroista aiheutuvat tulokset saatiin jätettyä pois mittauksista.







Sivuston siirto ja toimintaan saattaminen ei kuitenkaan mennyt ilman ongelmia uudella palvelimella yhteensopivuusongelmien vuoksi ja tuloksena on mittaustulosten

saaminen vain Pingdomin testaustyökalua käyttäen. Uuden palvelimen muutos nopeuteen on noin 1.3 sekuntia nopeampi sivun lataaminen. Kuvioissa 18, 19 ja 20 esitellään mittaustuloksia nykyisen sivuston nopeudesta uudella palvelimella.









Kuvio 18. Sivun siirrettynä uudelle palvelimelle. Mittaustulokset Pingdomin testiä käyttäen.

## Content size by content type

CONTENT TYPE	PERCENT	SIZE
 Image	92.58%	3.6 MB
 Script	4.04%	157.2 KB
 CSS	2.22%	86.4 KB
 XHR	0.66%	25.6 KB
 HTML	0.29%	11.3 KB
 Font	0.21%	8.1 KB
Total	100.00%	3.9 MB

Kuvio 19. Sivu siirrettynä uudelle palvelimelle. Resurssien koko tyyppin mukaan Pingdom-testissä.

## Requests by content type

CONTENT TYPE	PERCENT	REQUESTS
 Image	55.77%	29
 Script	23.08%	12
 CSS	9.62%	5
 XHR	7.69%	4
 HTML	1.92%	1
 Font	1.92%	1
Total	100.00%	52

Kuvio 20. Sivun siirrettynä uudelle palvelimelle. HTTP-pyyntö resurssityypin mukaan Pingdom-testissä.

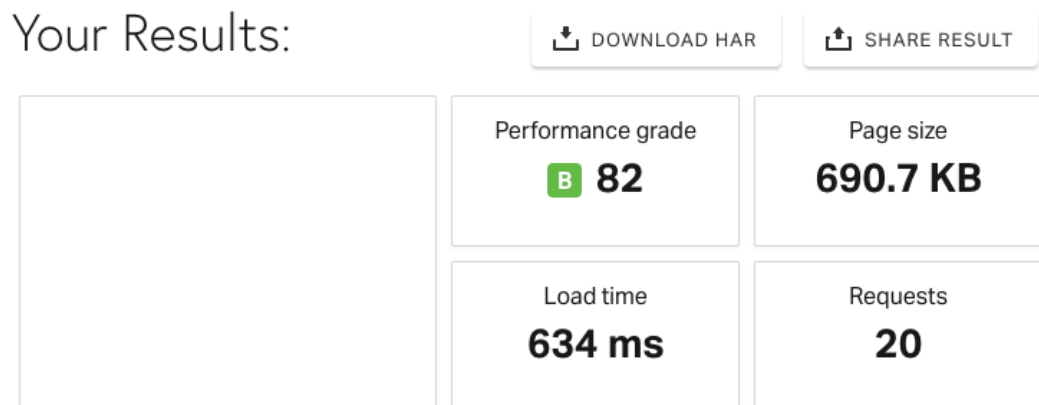
## 6.4 Uuden sivuston mittaukset

Uuden sivuston valmistuttua tehtiin mittaukset, joilla saatiin tietoa miten paljon sivuston uudelleen rakentaminen nopeuttaa sivua. Kuvioissa 21 – 24 esitellään uusien sivujen nopeustestien tuloksia.

**Laboratoriodata** ☰






● Ensimmäinen sisällön renderöinti	1,7 s	● Ensimmäinen merkityksellinen renderöinti	1,7 s
● Nopeusindeksi	2,3 s	■ CPU:n ensimmäinen toimittomuusjakso	3,8 s
■ Interaktiivisuutta edeltävä aika	4,1 s	▲ Suurin mahdollinen ensimmäisen toiminnon viive	430 ms

Kuvio 21. Uusien sivujen laboratoriodata PageSpeed-testissä.








Kuvio 22. Uuden sivuston yleiskuva mittaustuloksista Pingdom-testissä.

## Content size by content type

CONTENT TYPE	PERCENT	SIZE
 Script	39.05%	263.9 KB
 Image	37.51%	253.5 KB
 Font	11.96%	80.8 KB
 CSS	10.52%	71.1 KB
 HTML	0.97%	6.5 KB
Total	100.00%	675.9 KB

Kuvio 23. Uuden sivuston resurssien koko Pingdom-testissä.

## Requests by content type

CONTENT TYPE	PERCENT	REQUESTS
 Image	40.00%	6
 Script	26.67%	4
 CSS	20.00%	3
 HTML	6.67%	1
 Font	6.67%	1
Total	100.00%	15

Kuvio 24. Uuden sivuston HTTP-pyynnöt Pingdom-testissä.

## 6.5 Lisäosat

WordPressin tärkeimpiin ominaisuuksiin kuuluu lisäosa-arkkitehtuuri, joka mahdollistaa valmiiden laajennusten käyttämisen WordPress-sivuilla. Lisäosilla voidaan helpottaa ja automatisoida erilaisia toimenpiteitä, kuten tässä tapauksessa optimointeja. Optimointeihin käytettiin WordPress-lisäosia: EWWW Image Optimizer ja WP Rocket. Lisäosien valinnat tuntuivat turvallisilta, koska niiden ominaisuudet olivat sopivia käyttötarkoitukseen, ne olivat useissa lähteissä suositeltujen lisäosien listalla, niiden jatkokehitys on ollut aktiivista ja aktiivisten käyttäjien määrät suuria. Lisäosat asennettiin WordPress-hallintasivun kautta ja otettiin käyttöön. Käyttöänoton jälkeen lisäosien omista asetuksista valittiin käytettävät optimointivalinnat.

## 6.6 Kuvien optimointi ja välimuistin käyttöönotto

Seuraavaksi optimoitiin sivustolla olevat kuvat ja otettiin välimuisti käyttöön. Kuvien optimointiin käytettiin EWWW Image Optimizer –lisäosaa, jolla saatiin pienennettyä kuvien kokoa. Sivun välimuisti toteutettiin WP Rocket –lisäosalla. Kuvioissa 25 – 28 esitellään kuvien optimoinnin ja välimuistin vaikutusta nopeuteen.



## Laboratoriodata



● Ensimmäinen sisällön renderöinti	1,7 s	● Ensimmäinen merkityksellinen renderöinti	1,7 s
● Nopeusindeksi	2,1 s	● CPU:n ensimmäinen toimittomuusjakso	3,6 s
● Interaktiivisuutta edeltävä aika	3,8 s	▲ Suurin mahdollinen ensimmäisen toiminnon viive	330 ms

Kuvio 25. Kuvien optimoinnin ja välimuistin vaikutus PageSpeed-työkalussa.






## Your Results:

[↓ DOWNLOAD HAR](#)
[↑ SHARE RESULT](#)

	Performance grade	Page size
	<b>B 82</b>	<b>638.2 KB</b>
	Load time	Requests
	<b>661 ms</b>	<b>20</b>






Kuvio 26. Kuvien optimoinnin ja välimuistin vaikutus Pingdom-työkalun yhteenvedossa.

## Content size by content type

CONTENT TYPE	PERCENT	SIZE
 Script	42.21%	263.9 KB
 Image	32.45%	202.9 KB
 Font	12.93%	80.8 KB
 CSS	11.37%	71.1 KB
 HTML	1.05%	6.5 KB
Total	100.00%	625.3 KB

Kuvio 27. Kuvien optimoinnin ja välimuistin vaikutus resurssien kokoon Pingdom-testissä.

## Requests by content type

CONTENT TYPE	PERCENT	REQUESTS
 Image	40.00%	6
 Script	26.67%	4
 CSS	20.00%	3
 HTML	6.67%	1
 Font	6.67%	1
Total	100.00%	15

Kuvio 28. Kuvien optimoinnin ja välimuistin vaikutus HTTP-pyyntöjen määrään Pingdom-testissä.

### 6.7 HTML, CSS ja JavaScript –tiedostojen minifointi

HTML, CSS ja JavaScript –tiedostot minifointiin käyttäen WP Rocket –lisäosaa. Kuvioissa 29 – 32 esitellään minifoinnin vaikutusta sivuston nopeuteen.

## Laboratoriodata



● Ensimmäinen sisällön renderöinti	1,9 s	● Ensimmäinen merkityksellinen renderöinti	1,9 s
● Nopeusindeksi	1,9 s	● CPU:n ensimmäinen toimittomuusjakso	1,9 s
● Interaktiivisuutta edeltävä aika	2,3 s	● Suurin mahdollinen ensimmäisen toiminnon viive	60 ms

Kuvio 29. HTML, CSS ja JavaScript -tiedostojen minifoinnin vaikutus PageSpeed-testissä.






## Your Results:

[↓ DOWNLOAD HAR](#)
[↑ SHARE RESULT](#)

	Performance grade	Page size
	<b>B 84</b>	<b>406.2 KB</b>
	Load time	Requests
	<b>303 ms</b>	<b>20</b>






Kuvio 30. Yhteenveto HTML, CSS ja JavaScript -tiedostojen minifoinnin vaikutuksesta Pingdom-testissä.

## Content size by content type

CONTENT TYPE	PERCENT	SIZE
 Image	51.59%	202.9 KB
 Font	20.55%	80.8 KB
 CSS	16.43%	64.6 KB
 Script	9.72%	38.2 KB
 HTML	1.70%	6.7 KB
Total	100	393.3 KB

Kuvio 31. HTML, CSS ja JavaScript -tiedostojen minifoinnin vaikutus resurssien kokoon.

## Requests by content type

CONTENT TYPE	PERCENT	REQUESTS
 Image	40.00%	6
 Script	26.67%	4
 CSS	20.00%	3
 HTML	6.67%	1
 Font	6.67%	1
Total	100.00%	15

Kuvio 32. HTML, CSS ja JavaScript -tiedostojen minifoinnin vaikutus HTTP-pyyntöjen määrään.

## 6.8 CSS ja Javascript –tiedostojen optimointi

Optimoinnit toteuttiin WP Rocket –lisäosalla. CSS:n osalta lisäosa käy läpi jokaisen post-tyypin – artikkelit, sivut, tuotteet jne. – sekä taksonomiat ja sen jälkeen muodostaa näille kriittiset CSS-polut, lähettää ne takaisin lisäosalle, joka lisää luodut kriittiset CSS-polut tarvittaville sivuille. (Documentation for WP Rocket 2020.)

JavaScript-tiedostoille lisäosa lisää defer-avainsanan, joka kertoo selaimelle, että tiedoston voi ladata taustalla ja suorittaa, kun se on valmis. Jos JavaScript-tiedosto on jo merkitty async-avainsanalla, ei defer-lisäystä tehdä. (Documentation for WP Rocket 2020.) Kuvioissa 33 – 36 esitellään CSS ja Javascript-tiedostojen optimoinnin vaikutusta sivuston nopeuteen.

## Laboratoriodata



● Ensimmäinen sisällön renderöinti	1,7 s	● Ensimmäinen merkityksellinen renderöinti	1,7 s
● Nopeusindeksi	1,7 s	● CPU:n ensimmäinen toimittomuusjakso	1,7 s
● Interaktiivisuutta edeltävä aika	1,9 s	● Suurin mahdollinen ensimmäisen toiminnon viive	70 ms

Kuvio 33. CSS ja Javascript-tiedostojen optimoinnin vaikutus PageSpeed-testissä.






## Your Results:

[↓ DOWNLOAD HAR](#)
[↑ SHARE RESULT](#)

	Performance grade	Page size
	<b>B 83</b>	<b>408.9 KB</b>
	Load time	Requests
	<b>281 ms</b>	<b>20</b>






Kuvio 34. CSS ja Javascript –tiedostojen optimoinnin vaikutus Pingdom-testissä.

## Content size by content type

CONTENT TYPE	PERCENT	SIZE
 Image	51.24%	202.9 KB
 Font	20.41%	80.8 KB
 CSS	16.32%	64.6 KB
 Script	9.66%	38.2 KB
 HTML	2.37%	9.4 KB
Total	100.00%	396.0 KB

Kuvio 35. CSS ja Javascript –tiedostojen optimoinnin vaikutus resurssien kokoon.

## Requests by content type

CONTENT TYPE	PERCENT	REQUESTS
 Image	40.00%	6
 Script	26.67%	4
 CSS	20.00%	3
 HTML	6.67%	1
 Font	6.67%	1
Total	100.00%	15

Kuvio 36. CSS ja Javascript –tiedostojen optimoinnin vaikutus HTTP-pyyntöjen määrään.

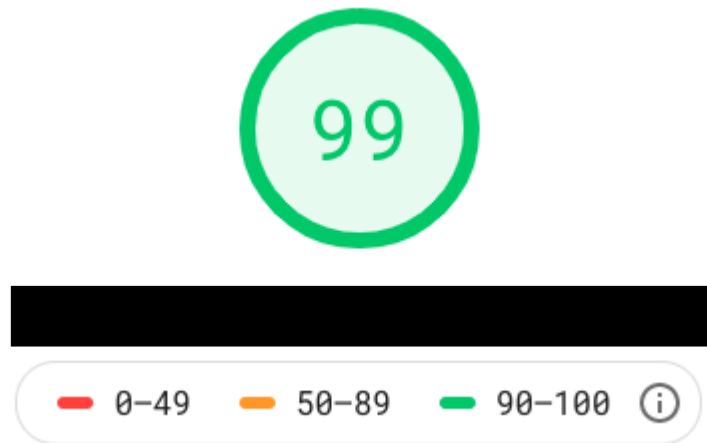


## 7 Tulokset

Sivuston koko pienentyi uudelleen rakentamisen ja optimointien myötä 3.9 megatavusta n. 408 kilotavuun, tarkoittaa noin 3,5 megatavun ja noin 89.5 prosentin pienentymistä.

Sivun lataamiseen menevä aika lyheni noin 1.35 sekunnista noin 0.3 sekuntiin, joka tarkoittaa noin 1.05 sekuntia ja noin 77,78 prosenttia lyhyempää aikaa.

Näiden lisäksi optimointien selkeitä hyötyjä uudella sivustolla olivat: prosessorin ensimmäisen toimitusjakson lyhentymisen noin 3.8 sekunnista noin 1.7 sekuntiin, joka tarkoittaa noin 2.1 sekuntia ja noin 55,26 prosenttia nopeampaa aikaa, interaktiivisuutta edeltävän ajan lyheneminen noin 4.1 sekunnista noin 1.9 sekuntiin, joka tarkoittaa noin 2.2 sekuntia ja 53,66 prosenttia lyhyempää aikaa sekä suurimman mahdollisen ensimmäisen toiminnon viiveen lyhentymisen noin 430 millisekunnista noin 70 millisekuntiin, joka tarkoittaa noin 360 millisekuntia ja noin 83,72 prosenttia lyhyempää aikaa. Optimointien lopullista vaikutusta esitellään kuvioissa 37, 38 ja 39. Tuloksien esittelyä taulukossa 1.



Kuvio 37. Tulos PageSpeed-testissä optimointien jälkeen.

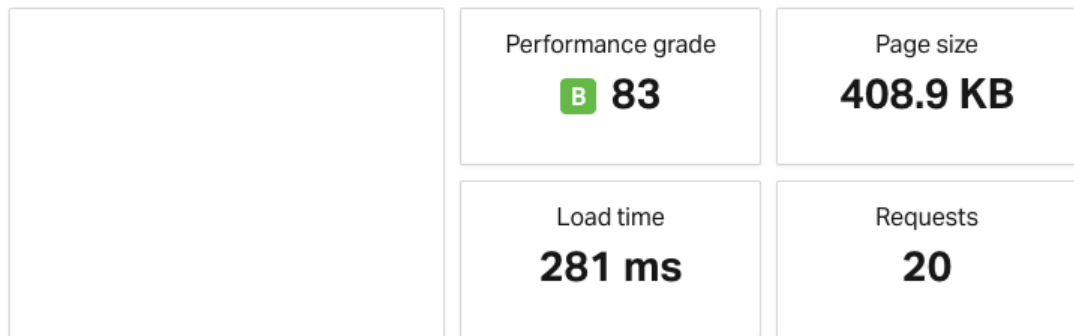
**Laboratoriodata** ☰

● Ensimmäinen sisällön renderöinti	1,7 s	● Ensimmäinen merkityksellinen renderöinti	1,7 s
● Nopeusindeksi	1,7 s	● CPU:n ensimmäinen toimitusjakso	1,7 s
● Interaktiivisuutta edeltävä aika	1,9 s	● Suurin mahdollinen ensimmäisen toiminnon viive	70 ms

Kuvio 38. Laboratoriodata PageSpeed-testissä optimointien jälkeen.

## Your Results:

 DOWNLOAD HAR

 SHARE RESULT


Kuvio 39. Yhteenveto Pingdom-testissä optimointien jälkeen.

	Uusi sivusto	Uusi sivusto uudelta palvelimelta	Muutos palvelimen vaihdolla	Uusi sivusto	Muutos vs vanha sivu uudelta palvelimelta	Uusi sivusto + kaviien optimoinnit + cache	Muutos vs uusi	Muutokset	Muutos vs uusi	CSS + JS optimoinnit	Muutos vs uusi
Ensimmäinen sisältö renderöinti (s)	3,9	1,7	1,7	1,7	0,0	1,9	0,0	1,9	-0,2	1,7	0,0
Ensimmäinen merkityksellinen renderöinti (s)	4,1	1,7	1,7	1,7	0,0	1,9	-0,2	1,7	0,0	1,7	0,0
Hyväksymäaika (s)	7,2	2,3	2,3	2,3	0,0	1,9	0,4	1,9	0,4	1,7	0,2
CPU:n ensimmäinen toimitusajanko (s)	4,1	3,8	3,8	3,8	0,2	1,9	1,9	1,9	1,7	2,1	0,2
Interaktiivisuutta edeltävä aika (s)	4,8	4,1	4,1	4,1	0,7	2,3	1,8	1,9	1,9	2,2	0,3
Suurin mahdollinen ensimmäinen toiminnon viive (ms)	120	40	40	40	80	60	70	70	70	80	10
Page size (KB)	3,90	3,90	0,00	0,69	3,21	0,64	100	60	0,28	0,41	0,28
Load time (s)	2,71	1,35	1,36	0,63	0,72	0,66	-0,05	0,41	-0,05	0,3	0,35
Requests (kpl)	56	53	3	20	36	20	0	20	0	20	0
						Muutos ed.		Muutos ed.		Muutos ed.	
Ensimmäinen sisältö renderöinti (s)	3,9	1,7	1,7	1,7	0	1,9	-0,2	1,7	0,0	1,7	0,0
Ensimmäinen merkityksellinen renderöinti (s)	4,1	1,7	1,7	1,7	0	1,9	-0,2	1,7	0,0	1,7	0,0
Hyväksymäaika (s)	7,2	2,3	2,3	2,3	0,0	1,9	0,2	1,9	0,2	1,7	0,2
CPU:n ensimmäinen toimitusajanko (s)	4,1	3,8	3,8	3,8	0,2	1,9	1,9	1,9	1,7	2,1	0,2
Interaktiivisuutta edeltävä aika (s)	4,8	4,1	4,1	4,1	0,7	2,3	1,8	1,9	1,9	2,2	0,3
Suurin mahdollinen ensimmäisen toiminnon viive (ms)	120	40	40	40	80	60	70	70	70	80	10
Page size (KB)	3,90	3,90	0,00	0,69	3,21	0,64	100	60	0,28	0,41	0,28
Load time (s)	2,71	1,35	1,36	0,63	0,72	0,66	-0,05	0,41	-0,05	0,3	0,35
Requests (kpl)	56	53	3	20	36	20	0	20	0	20	0

Taulukko 1. Tulokset.

## 8 Pohdinta

Opinnäytetyön tarkoituksena oli selvittää, mitä keinoja on parantaa WordPress-julkaisualustan päälle rakennetun sivuston latausnopeutta ja samalla parantaa sivuston

saamia tuloksia latausnopeusmittareissa. Käytännön työssä rakennettiin asiakkaan tarpeiden mukaan sivuston pohja uusiksi ja sivujen optimoinnin viimeistelyyn käytettiin WordPress-lisäosia. Näin parannettiin sivujen nopeutta ja käyttökokemusta.

Lopputuloksena tilaajalle saatiin nopeammin toimivat uudet verkkosivut päivitetyllä toteutuksella, jonka myötä sivuston jatkokehitys on helpompaa ja nopeampaa. Opinnäytetyön tekijä sai paremman kuvan verkkosivujen rakenteista sekä tietoa siitä, miten optimoinnit toimivat ja auttavat nopeuttamaan sivuston toimintaa.

Mittaukset puoltavat, että uusien sivujen toteutus onnistui teknisestä näkökulmasta hyvin. Alkuperäisen suunnitelman mukaan tarkoituksena oli kehittää vanhaa sivustoa, mutta sivuston vanhan teknisen toteutuksen vuoksi tämä ei ollut järkevää ja päädyttiin kehittämään kokonaan uusi sivusto. Uuden sivun rakentaminen mahdollisti ajan myötä kertyneiden väliaikaisratkaisujen karsimisen, mutta lisäsi myös työmäärää.

Mittareiden luotettavuuteen vaikuttaa sen hetkinen palvelun käyttäjämäärä, palvelinten kuorma ja verkkoliikenteen määrä, joten tulokset ovat noin-lukuja. Tällaiset luvut ovat kuitenkin riittäviä erojen näkemiseen. Kokenut alalla ollut tekijä olisi voinut valita eri työkalut, mutta opinnäytetyöntekijän tietämyksen mukaan nämä olivat hyviä valintoja tähän tarkoitukseen.

Tuloksista hyötyvät erityisesti henkilöt, jotka haluavat optimoida WordPress-alustalla olevien verkkosivujen nopeutta. Opinnäytetyöstä on myös hyötyä henkilölle, joille termit eivät ole vielä tuttuja, koska kaikki termit ovat määritelty. Sen lisäksi esitellään työkaluja, joita voi käyttää. Esitellyistä termeistä ja työkaluista on myös hyötyä muille saman aihepiirin opinnäytetyöntekijöille.

## Lähteet

- Brutlag, J. 2009. Speed Matters. Verkkojulkaisu Google AI Blog [www-sivuilla](http://www-sivuilla). Viitattu 26.8.2019. <https://ai.googleblog.com/2009/06/speed-matters.html>.
- Clark, M. 2018. How the BBC builds websites that scale. Artikkele Creative Bloq [www-sivuilla](http://www-sivuilla) 17.1.2018. Viitattu 29.9.2019. <https://www.creativebloq.com/features/how-the-bbc-builds-websites-that-scale>.
- Critical Rendering Path. 2019. E-opas MDN web docs [www-sivuilla](http://www-sivuilla) 26.3.2019. Viitattu 23.8.2019. [https://developer.mozilla.org/en-US/docs/Web/Performance/Critical\\_rendering\\_path#Understanding\\_CRP/](https://developer.mozilla.org/en-US/docs/Web/Performance/Critical_rendering_path#Understanding_CRP/).
- Cutts, M & Singhal, A. 2010. Using site speed in web search ranking. Verkkojulkaisu Google Webmaster Central Blog [www-sivuilla](http://www-sivuilla). Viitattu 26.8.2019. <https://webmasters.googleblog.com/2010/04/using-site-speed-in-web-search-ranking.html>.
- Documentation for WP Rocket. 2020. E-opas docs.ewww.io [www-sivuilla](http://www-sivuilla). Viitattu 30.4.2020. <https://docs.ewww.io/>.
- First contentful paint. 2019. Verkkojulkaisu MDN web docs [www-sivuilla](http://www-sivuilla). Viitattu 30.4.2020. [https://developer.mozilla.org/en-US/docs/Glossary/First\\_contentful\\_paint](https://developer.mozilla.org/en-US/docs/Glossary/First_contentful_paint).
- Grigorik, I. 2019. Critical Rendering Path. Web Fundamentals 29.5.2019. Viitattu 26.8.2019. <https://developers.google.com/web/fundamentals/performance/critical-rendering-path/>.
- How to Speed up Your WordPress Site (Ultimate 2020 Guide). 2020. Artikkele [kinsta.com](http://kinsta.com) [www-sivuilla](http://www-sivuilla). Viitattu 29.4.2020. <https://kinsta.com/learn/speed-up-wordpress>.

- Jackson, B. 2020. How to Optimize Images for Web and Performance. Verkkojulkaisu Kinsta.com www-sivuilla. Viitattu 23.4.2020. <https://kinsta.com/blog/optimize-images-for-web/>.
- Kananen, J. 2015. Kehittämistutkimuksen kirjoittamisen käytännön opas. Miten kirjoitan kehittämistutkimuksen vaihe vaiheelta. Jyväskylä: Jyväskylän ammattikorkeakoulu.
- Kananen, J. 2008. Kvali: Kvalitatiivisen tutkimuksen teoria ja käytänteet. Jyväskylän ammattikorkeakoulu, liiketalous.
- Koskelo, T. N.d. Mikä on konversio? Artikkelit digimarkkinointi.fi www-sivuilla. Viitattu 23.4.2020. <https://www.digimarkkinointi.fi/blogi/mika-konversio>.
- Max, S. 2014. Optimizing the Critical Rendering Path. Artikkelit sitepoint.com www-sivuilla. Viitattu 23.4.2020. <https://www.sitepoint.com/optimizing-critical-rendering-path/>.
- Muller, B. & Moz.com staff. N.d. How Search Engines Work: Crawling, Indexing, and Ranking. E-opas Moz.com www-sivuilla. Viitattu 30.12.2019. <https://moz.com/beginners-guide-to-seo/how-search-engines-operate>.
- Määrällinen tutkimus. 2015. Verkkojulkaisu Jyväskylän yliopiston Koppa www-sivuilla. Viitattu 30.4.2020. <https://koppa.jyu.fi/avoimet/hum/menetelmapolkuja/menetelmapolku/tutkimusstrategiat/maarallinen-tutkimus>.
- Responsive Web Design – Media Queries. N.d. E-opas w3schools.com www-sivuilla. Viitattu 23.4.2020. [https://www.w3schools.com/css/css\\_rwd\\_media\\_queries.asp](https://www.w3schools.com/css/css_rwd_media_queries.asp)
- Sullivan, D. 2016. Google now handles at least 2 trillion searches per year. Artikkelit Searchengineland.com www-sivuilla. Viitattu 30.12.2019. <https://searchengineland.com/google-now-handles-2-999-trillion-searches-per-year-250247>.
- Suomen virallinen tilasto (SVT). 2019. Suomalaisten internetin käyttö 2019. Verkkojulkaisu. Helsinki: Tilastokeskus. Viitattu 30.4.2020. [http://www.stat.fi/til/sutivi/2019/sutivi\\_2019\\_2019-11-07\\_kat\\_001\\_fi.html](http://www.stat.fi/til/sutivi/2019/sutivi_2019_2019-11-07_kat_001_fi.html).
- Wagner, J. 2019. Why performance matters. E-opas Google Developers www-sivuilla. Viitattu 29.7.2019. <https://developers.google.com/web/fundamentals/performance/why-performance-matters/>.
- Wang, Z & Phan, D. 2018. Using page speed in mobile search ranking. Verkkojulkaisu Google Webmaster Central Blog www-sivuilla. Viitattu 29.5.2019. <https://webmasters.googleblog.com/2018/01/using-page-speed-in-mobile-search.html>.

Weiss, Y. 2018. A Guide To Loading Assets on the Web. Freiburg, Saksa:  
New Frontiers In Web Design.