



TAMPEREEN
AMMATTIKORKEAKOULU

BOOTSTRAP-SOVELLUSKEHYKSEN HYÖDYNTÄMINEN WORDPRESS-TEEMAN KEHITYKSESSÄ

Jussi Vainionpää

Opinnäytetyö
Maaliskuu 2018
Tietojenkäsittely
Web-palvelut



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittely
Web-palvelut

VAINIONPÄÄ, JUSSI:

Bootstrap-sovelluskehityksen hyödyntäminen Wordpress-teeman kehityksessä

Opinnäytetyö 30 sivua
Maaliskuu 2018

Tämän opinnäytetyön tavoitteena oli suunnitella ja rakentaa oma WordPress-teema mainostoimisto Trival Oy:n käyttöön ja tutkia Bootstrap-sovelluskehityksen hyötyjä teeman kehityksessä. Trival toteuttaa nettisivuja asiakkailleen ja yrityksellä oli tarve omalle WordPress-teemalle, jota voi hyödyntää useissa tulevilla projekteilla ilman, että yrityksen tarvitsee miettiä kolmansien osapuolien toteuttamien teemojen lisenssimaksuja tai tietoturvaongelmia. Teeman tuli olla sellainen, että loppukäyttäjä eli asiakas voi helposti muokata sitä.

Trival on käyttänyt aikaisemmin ilmaisia sekä maksullisia kolmannen osapuolen toteuttamia teemaratkaisuja pohjina nettisivujen teossa ja muokannut näistä teemoista asiakkaan imagoon ja brändiin sopivan ilmeen. Kaikki Trivalin toteuttamat asiakassivustot ovat nykyään responsiivisia, eli ne skaalautuvat mobiililaitteille sopiviksi. Luonnollisesti tämän opinnäytetyön tuotoksena syntyneen Trival-teemankin täytyi tukea responsiivisuutta. Bootstrap oli täydellinen ratkaisu tähän.

Kehityksen apuna toimi suureksi osaksi WordPressin oma dokumentaatio ja internetlähteet sekä muutama fyysinen kirjajulkaisu web-suunnittelusta. Työssä käytettiin HTML-, CSS-, PHP- ja JavaScript-tekniikoita. Kehitysympäristö toteutettiin suoraan uusimman PHP 7:n päälle, jotta teeman pohjarakenne on heti ajan tasalla eikä se käytä vanhaa tekniikkaa.

Teema on raportoinnin valmistuessa käytännössä valmis, mutta sitä ei voida vielä ottaa käyttöön tuotannossa, koska WordPressiin julkaistu isompi päivitys toi mukanaan sellaisia muutoksia, joihin teema täytyy jatkokehityksessä sulauttaa.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Information Systems
Option of Web Services

VAINIONPÄÄ, JUSSI:

Using a Bootstrap-Framework in the Development of a Wordpress Theme

Bachelor's thesis 30 pages

March 2018

The purpose of this thesis was to design and build a WordPress theme for the marketing business Trival Ltd. and to find out how well the CSS framework Bootstrap will work together in developing a WordPress theme. Trival Ltd. designs websites for their customers and the business had a need for their own WordPress theme which they could use in future projects without having to worry about license fees or security threats of third-party themes. The theme also needed to be easily customized by the end-user, ie. the customer.

Trival has used free and paid third-party themes as the base theme on which they design their own layouts which are designed purely with the customers' brand and image in mind. All new websites developed by Trival are responsive so naturally the theme developed for this thesis had to be responsive as well. Bootstrap was a perfect solution to this.

A big help in developing this theme was WordPress' own documentation and different Internet sources as well as a few published books about web design. Techniques that were used include HTML, CSS, PHP and JavaScript. The development environment for the theme was built on PHP version 7 from the start so that the whole frame of the theme would be up to date and the theme would not use any deprecated functions.

The theme was essentially ready at the time this report was finished, but it can't be used in development yet because the newest big update to WordPress changed some functions of the customizer and it broke the theme. This is a priority to be fixed in the theme so Trival can start using the theme in development.

Key words: wordpress, content management system, website, web design, web development

SISÄLLYS

1	JOHDANTO.....	6
1.1	Työn tausta.....	6
1.2	Työn tavoite	6
2	WORDPRESS-TEEMAN SUUNNITTELU	7
2.1	Valmiin teeman käyttö.....	7
2.2	Oman teeman luonti.....	7
2.3	Lapsiteemat	8
3	CSS-SOVELLUSKEHYKSET	10
3.1	Valmiiden sovelluskehysten tarve web-kehityksessä.....	10
3.2	Tyylien nollaus	10
3.3	Responsiiviseen web-kehitykseen tarkoitettu ruudukko.....	11
3.4	CSS-kehysten valinta	12
4	BOOTSTRAP.....	13
4.1	Käyttöönotto	13
4.2	Bootstrap 3 & 4.....	13
4.3	Laitetuki	13
4.4	Reboot.....	15
4.5	Käyttö tuotannossa.....	15
4.5.1	Responsiivisuus.....	16
4.5.2	Taulukot	18
4.5.3	Kuvat.....	20
4.5.4	Helppokäyttöisyys ja käytettävyys.....	21
4.5.5	Scrollspy.....	22
5	TRIVAL-TEEMA	23
5.1	Kehitys	23
5.2	Rakenne	24
5.2.1	Header	24
5.2.2	Sisältö.....	25
5.2.3	Footer	26
5.3	Mukautin.....	28
6	POHDINTA.....	29
	LÄHTEET.....	30

LYHENTEET JA TERMIT

Bootstrap	CSS-sovelluskehys, joka on kehitetty web-suunnittelijoiden avuksi nopeuttamaan modernien verkkosivustojen toteuttamista mobiililaitteille skaalautuviksi.
breakpoint	Responsiivisessa web-suunnittelussa käytetty termi, jolla tarkoitetaan näyttökokoa, jossa sivuston näkymä vaihtuu. Määritellään yleensä pikseleissä näytön resoluution leveytenä.
CSS	Cascading Style Sheets on tyyliohje verkkosivustolle, johon määritellään WWW-dokumenttien käyttämät tyylit.
HTML	HyperText Markup Language on standardoitu kuvauskieli, jota käytetään verkkosivujen rakenteen luomiseen.
lapsiteema	Lapsiteema on WordPress-teeman alateema, joka hakee tyyli-tiedostonsa ja rakenteensa isäntäteemaltaan. Lapsiteemalle luodaan oma tyylitiedosto ja tarvittavat rakennetiedostot, jotka yliajavat isäntäteeman tiedostot.
MySQL	MySQL on relaatiotietokantaohjelmisto, jota käytetään yleisesti dynaamisten verkkosivujen tietokantana.
PHP	Hypertext Preprocessor on ohjelmointikieli, jota käytetään verkkosivujen ja verkkopalveluiden kehittämisessä.
responsiivisuus	Responsiivisuudella tarkoitetaan sivustoa, joka skaalautuu automaattisesti eri näyttökoolle sopivaksi. Responsiivinen sivusto on yksi kokonaisuus eikä mobiilinäkymää ole rakennettu erikseen.
Webhotelli	Webhotelli on palvelintila, joka vuokrataan verkkosivustoa varten.

1 JOHDANTO

1.1 Työn tausta

Mainostoimisto Trival Oy luo asiakkailleen verkkosivustoja Wordpress-sisällönhallintajärjestelmällä. Yritys on käyttänyt valmiita teemaratkaisuja verkkosivujen pohjana ja luonut valmiiden teemojen päälle asiakkaan tarpeisiin ja imagoon sopivan ulkoasun. Trival Oy:n tavoitteena on siirtyä käyttämään yrityksen itse luomia teemaratkaisuja sivustojen pohjina lisätäkseen kustannustehokkuutta asiakasprojekteissa.

1.2 Työn tavoite

Tämän opinnäytetyön tavoitteena on selvittää Bootstrap-sovelluskehiksen käytön hyötyjä ja mahdollisia haittoja Wordpress-teeman kehityksessä. Sovelluskehystä käytetään verkkosivujen toteuttamisen apuna. Se tekee verkkosivun kehittämisestä nopeampaa ja helpompaa ja näin ollen lisää kehitystyön kustannustehokkuutta.

Samana teeman runkoa voidaan käyttää usean eri verkkosivuston kehityksessä luomalla lapsiteemoja, jotka käyttävät isäntäteeman tyylejä ja rakennetta pohjana. Lapsiteeman tarkoitus on yksilöidä ulkoasu asiakkaan käyttötarpeisiin sopivaksi. Kun kehittäjällä on käytössä yksi teema, joka hyödyntää Bootstrap-sovelluskehystä, kaikki tämän teeman lapsiteemat hyödyntävät sitä myös. Jatkossa tämän teeman pohjalta kehitettäessä verkkosivuja ei siis tarvitse miettiä sivuston runkoa ja sen rakenteen luomista vaan kehittäjä voi keskittyä suunnittelemaan sivuston ulkoasua.

Tämän opinnäytetyön tuotteina syntyy Trival Oy:n sisäiseen käyttöön oma teema, jota yritys hyödyntää tulevissa asiakasprojekteissa luomalla lapsiteemoja tämän teeman pohjalta. Teeman lisäksi toisena tuotteena syntyy Trival Oy:lle teeman ja Bootstrapin käyttöohjedokumentti, jota mahdolliset uudet työntekijät voivat lukea ja oppia näin käyttämään tätä teemaa tulevaisuuden projekteissa.

2 WORDPRESS-TEEMAN SUUNNITTELU

2.1 Valmiin teeman käyttö

Wordpress perustuu sisällönhallintajärjestelmänä helppokäyttöisyydessään valmiiden teemojen käyttöön. Valmiita teemoja on julkisesti saatavilla tuhansia erilaisia sekä ilmaisia että maksullisia. Maksullisten teemojen etuna on yleensä ostettuun lisenssiin sisältyvä oikeus teeman kehittäjältä saatavaan tekniseen tukeen ja oikeus ladata teemaan julkaistavia päivityksiä. Maksulliset teemat myös sisältävät enemmän toimintoja kuin ilmaiset teemat.

Trival on käyttänyt asiakasprojekteissaan muutamaa erilaista maksullista teemaa, joista web-suunnittelijat ovat muokanneet asiakkaiden tarpeisiin ja imagoon sopivat ulkoasut ja rakenteet. Trival on myös käyttänyt muutamia ilmaisia teemoja asiakasprojekteissa, mutta niiden käyttöä on pyritty välttämään teeman julkaisijoiden tekemien harvojen päivitysten takia. Jos teemasta löytyy tietoturvaheikkous, on se saatava korjattua mahdollisimman pian. Asiakkaiden sivustojen tietoturva on mainostoimistolle ensimmäinen ja tärkein kriteeri verkkosivuston toiminnassa.

Verkkosivustolle vuokrataan palvelintila, jota kutsutaan webhotelliksi. Erilaisten palveluntarjoajien webhotellit saattavat poiketa ominaisuuksiltaan toisistaan. Trival pyrkii käyttämään uusissa asiakasprojekteissa mahdollisimman kattavia ja monipuolisia webhotelleja. Wordpressin nopeuden kannalta on tärkeää, että webhotelli mahdollistaa verkkosivuston kehittämisen PHP:n versiolla 7. Valmiiden teemojen on siis tuettava myös PHP:n versiota 7.

2.2 Oman teeman luonti

Wordpress-teeman voi luoda kuka tahansa. Oman teeman luominen vaatii kuitenkin ymmärrystä PHP-ohjelmointikielestä, MySQL-tietokannoista ja Wordpressin omista toiminnallisuuksista. Wordpress tarjoaa teemojen kehittäjille kattavan dokumentaation, jossa kerrotaan yksityiskohtaisesti, mistä teema koostuu ja miten teema saadaan taipumaan kehittäjän omiin tarpeisiin (WordPress.org. Theme handbook n.d.).

Oman teeman luomiseen tarvitaan vain kaksi tiedostoa: `index.php` ja `style.css`. WordPress käyttää `index.php`-tiedostoa hakeakseen sivuston sisällön ja `style.css`-tiedostoa luomaan sivuston visuaalisen ilmeen. `Style.css`-tiedosto sisältää myös teeman tiedot: teeman nimen, teeman luontipäiväyksen ja teeman tekijän tiedot. Teeman luonnissa kuitenkin suositellaan käyttämään useampia tiedostoja tukemaan kokonaisuutta kuten `header.php`, joka sisältää teeman yläosan tiedot ja `footer.php`, joka sisältää teeman alaosan tiedot. (Friedman 2013, 46.)

Tyylitiedoston ensimmäisinä tietoina on oltava teeman tiedot kommenttien sisällä. Kuvassa 1 on kuvattuna Trival-teeman tiedot tyylitiedostossa.

```

/*
Theme Name: WP Bootstrap Trival
Theme URI: http://www.trival.fi
Author: Jussi Vainionpaa
Description: WordPress Bootstrap Trival theme is a theme developed for Trival Oy to use in future development projects
Version: 1.0
License: GNU General Public License v2 or later
License URI: LICENSE
Text Domain: wp-bootstrap-trival
*/

```

KUVA 1: Trival-teeman tiedot tyylitiedostossa

2.3 Lapsiteemat

Lapsiteemoja luodaan, kun valmiiseen teemaan tehdään jotain muutoksia. Lapsiteema käyttää isäntäteeman rakennetta ja sen ulkoasutiedostoja. Teemojen kehittäjät julkaisevat aika ajoin päivityksiä teemoihinsa. Jos päivityksessä on muokattu isäntäteeman jotain ulkoasutiedostoa, se saattaisi hajottaa verkkosivuston ulkoasun. Kun sivusto on toteutettu käyttäen lapsiteemaa, sen tiedostoja ei päivitetä teeman päivityksen yhteydessä ja näin ollen, vaikka itse isäntäteeman päivitys hajottaisi ulkoasun, lapsiteeman tiedostoihin tehdyt muutokset pysyvät ennallaan. Kuvassa 2 on kuvattu isäntä- ja lapsiteeman hierarkia. Ainoa poikkeus on `style.css`-tiedosto, joka käyttää molempien teemojen tiedostoja. Lapsiteeman tyylitiedostoon kirjoitetut säännöt yliajavat isäntäteeman tyylitiedoston säännöt.

Toinen lapsiteeman isoista hyödyistä on tehtyjen muutosten helppo erottaminen. Lapsiteeman omasta kansioista on suoraan nähtävissä, mihin teeman osiin on tehty muutoksia. Esimerkiksi kuvassa 2 lapsiteemalle on luotu tiedostot `style.css`, `header.php` ja `footer.php`, mutta ei tiedostoa `index.php`. Tästä nähdään suoraan, että `index.php`-tiedostoon ei ole tehty muutoksia (Król 2015.)



KUVA 2: Lapsiteeman ja isäntäteeman hierarkia

Verkkosivuston kehityksessä eteen tulee usein ongelmia. Pienikin virhe koodissa voi hajottaa koko sivuston toiminnan ja varsinkin, jos kyseessä on jo julkinen sivusto, virhe on saatava korjattua mahdollisimman nopeasti. Kun sivusto on luotu lapsiteemaa käyttäen, voi kehittäjä huoletta poistaa ongelman aiheuttavan tiedoston lapsiteeman kansioista. Sivusto palaa tällöin käyttämään isäntäteeman vastaavaa tiedostoa, ja sivuston kehittäjä voi rauhallisin mielin alkaa tutkia ongelman aiheuttanutta koodia. (Król 2015.)

3 CSS-SOVELLUSKEHYKSET

3.1 Valmiiden sovelluskehysten tarve web-kehityksessä

CSS-sovelluskehyksellä tarkoitetaan jonkun tahon luomaa pakettia, joka koostuu valmiista CSS-säännöistä ja mahdollisesti myös JavaScript-tiedostoista. Sovelluskehukset on luotu noudattamaan web-kehityksen standardeja. Yleisimmin CSS-sovelluskehukset käyttävät hyödykseen tyylien nollausta, responsiiviseen web-kehitykseen tarkoitettua ruudukkoa, web-kirjasimia sekä viivagrafiikan muodossa olevia kuvakkeita (Wikipedia. CSS Framework 2014).

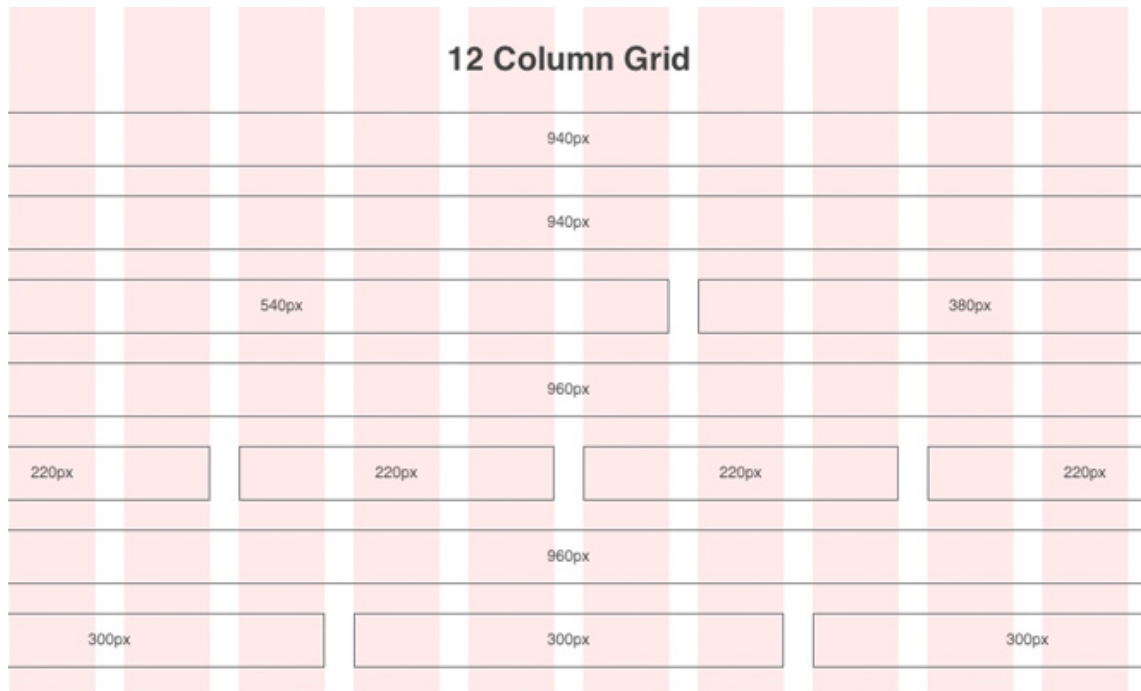
3.2 Tyylien nollaus

Jokainen internet-selain käyttää oletuksena jotain CSS-sääntöjä eri elementeille (Wikipedia. Reset style sheet 2018). Yleisimpiä sääntöjä ovat otsikoiden kirjasinkoot, kappaleiden väliin jäävät marginaalit ja kaikille tuttu sininen väri linkeissä. Web-sivusto, jolla on valkoinen tausta, mustaa tekstiä ja sinisiä alleviivattuja linkkejä on yleensä selkeä merkki siitä, että sivuston kehittäjä ei ole käyttänyt ollenkaan CSS-sääntöjä tyylitelläkseen sivustoa. Selaimet kuitenkin osaavat asettaa otsikoille oikeat kirjasinkoot ja linkeille sinisen värin oletustyylien kautta.

Yleisimpiä käytettyjä web-selaimia ovat tällä hetkellä Googlen kehittämä Chrome sekä Mozillan kehittämä Firefox. Uusissa Windows-tietokoneissa valmiina asennettuna oleva Microsoftin kehittämä Edge-selain omaa myöskin mainitsemisen arvoisen osuuden web-selainten käytöstä. Jokainen näistä edellä mainituista selaimista käyttää omia oletustyyliään. Tästä syystä jotkut sivustot saattavat näyttää erilaisilta eri selaimilla tarkasteltaessa. Web-kehittäjän näkökulmasta erilaiset oletustyyliä ovat yksi suuri päänsärky. Tähän ongelmaan ratkaisu on kuitenkin CSS-tyylien nollaus, jota usein kutsutaan ”reset stylesheet” tai ”CSS reset” termeillä.

3.3 Responsiiviseen web-kehitykseen tarkoitettu ruudukko

Web-kehityksessä voidaan käyttää apuna ruudukkoa (grid), joka mahdollistaa helpomman sivuston rakenteen luomisen. Ruudukko koostuu taulukoiden tapaan riveistä ja kolumneista. Ruudukon avulla web-sivustosta saadaan symmetrisen näköinen ja sisällön sijoittelu sekä sen hallinta helpottuvat. Kuvassa 3 on esitetty yleisimmin käytetty 12 kolumnin ruudukko.



KUVA 3: 12 kolumnin ruudukko (Kuva: Jake Rocheleau 2014)

3.4 CSS-kehyyksen valinta

Opinnäytetyön alkuvaiheessa oli valittava mitä CSS-sovelluskehystä käytettäisiin Wordpress-teeman kehityksessä. Heti alusta oli selvää, että sovelluskehyyksen on oltava hyvin tuettu ja suosittu. Tällä taattaisiin teeman jatkokehityksen kannalta sovelluskehyyksen pysyminen samana, eikä sitä tarvitsisi lähteä vaihtamaan muutaman vuoden käytön jälkeen.

Vaihtoehtoina olivat Foundation, Bulma, Skeleton sekä Bootstrap. Päädyin valitsemaan Twitterin kehittämän Bootstrapin, koska minulla oli sen käyttämisestä jo aikaisemmin kokemusta ja se on pitkään kehitteillä ollut ja hyvin laajalti käytetty sovelluskehys. Oman Wordpress-teeman kehityskin täytyi opetella opinnäytetyötä tehdessä alusta asti, joten uuden sovelluskehyyksen sisäistäminen samaan projektiin tuntui turhan suurelta taakalta.

4 BOOTSTRAP

4.1 Käyttöönotto

Bootstrapin käyttöönotto on tehty web-suunnittelijalle helpoksi. Sovelluskehys koostuu muutamasta CSS- ja JavaScript-tiedostosta, jotka sisällytetään sivuston lähdekoodiin kuten mikä tahansa muukin vastaava tiedosto. Bootstrap tarjoaa vaihtoehdoksi joko tiedostojen linkittämistä paikallisesti tai ulkoisesti BootstrapCDN:n kautta. Trivalin Wordpress-teemaan linkitän Bootstrapin paikallisesti.

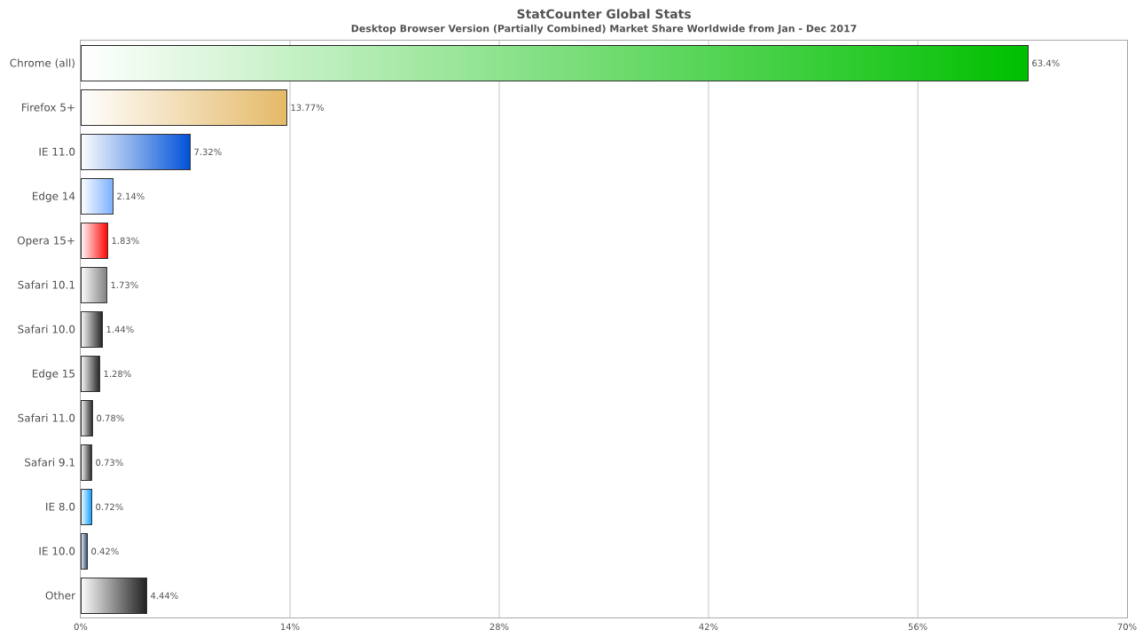
4.2 Bootstrap 3 & 4

Trivalin teeman kehityksen alkaessa Bootstrapista oli saatavilla versio 3 ja versio 4 oli vasta alpha-testausvaiheessa, joten päädyin aluksi käyttämään versiota 3. Kehityksen edessä kuitenkin versio 4 julkaistiin beta-vaiheeseen. Pohdittuani kehittämäni teeman tulevaisuutta päädyin vaihtamaan käyttöön kuitenkin version 4. Näin ollen, kun koko runko on rakennettu jo uusimman teknologian päälle, on tulevaisuudessa helpompi pitää teeman kehitys ajan tasalla.

4.3 Laitetuki

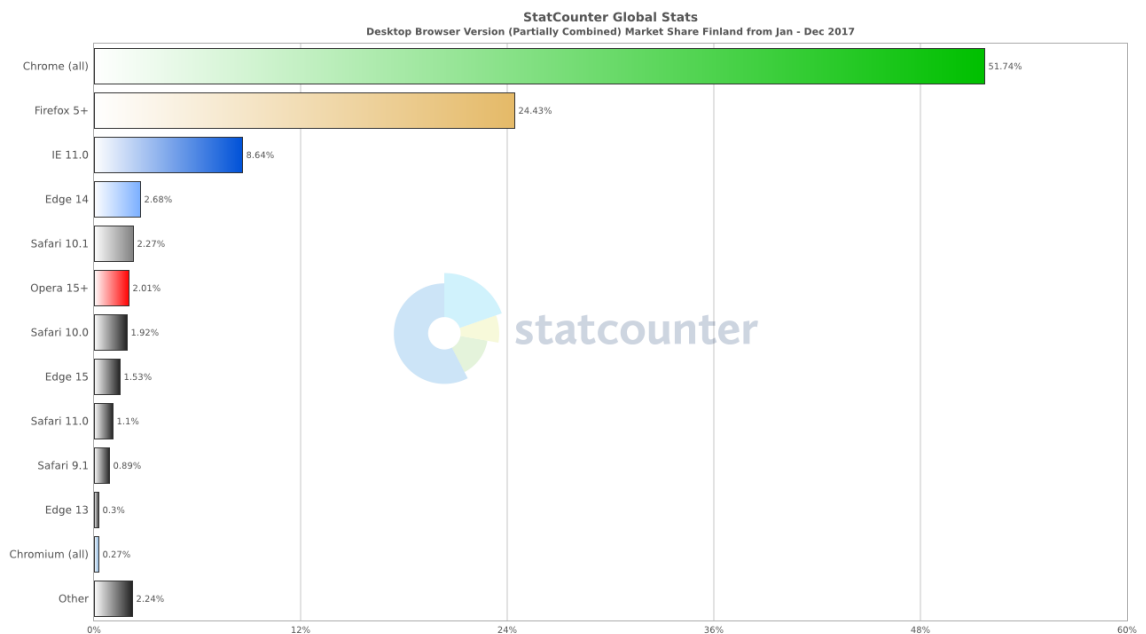
Bootstrapin versio 4 tukee kaikkia uusimpia, moderneja ja käytetyimpiä selaimia. Windowsilla tuki löytyy myös Internet Explorerin versioille 10 ja 11 sekä tuettuna on myös Microsoft Edgen kaikki versiot. (Bootstrap Documentation: Browsers and Devices 2017.)

Perustuen StatCounterin keräämiin tietoihin internetin käyttäjien käyttämistä selaimista ja selainten versioista (Kuva 4) päädyin tulokseen, että Trivalin teeman kehitykseen Bootstrap tukee tarpeeksi montaa eri selainversiota (yli 94 % maailmanlaajuisesti käytetyistä versioista). Vanhojen selainversioiden käyttäjämäärät ovat myös jatkuvassa laskussa, joten teeman selaintuki tulee olemaan riittävä.



KUVA 4: Selainversioiden käyttö maailmanlaajuisesti vuonna 2017 (Kuva: StatCounter 2018)

Suomen sisällä tilanne on vielä maailmanlaajuista tilannettakin parempi. Suomalaisen internetin käyttäjien käyttämistä selainversiosta tuki on jopa yli 96 % (Kuva 5).



KUVA 5: Selainversioiden käyttö Suomessa vuonna 2017 (Kuva: StatCounter 2018)

4.4 Reboot

Luvussa 3.2 käsiteltiin CSS-sovelluskehysten käyttämää tyylien nollausta. Reboot on Bootstrapin käyttämä työkalu tyylien nollaamiseen. Eri selaimet käyttävät oletustyyleinä hieman erilaisia määrittelyjä eri elementeille. Bootstrap muuttaa Rebootin avulla osan selainten käyttämistä oletusarvoista käyttämään rem-arvoja em-arvojen sijasta. Tällä saavutetaan komponenttien väliin jäävän tilan helppo skaalautuminen. (Bootstrap Documentation: Reboot 2017.) Em-arvot käyttävät isäntä-elementtinsä kirjasinkokoa määrittämään arvonsa, kun taas rem-arvot käyttävät dokumentin juureen asetettua kirjasinkokoa määrittämään arvonsa (Liew 2016.) Näin ollen käyttämällä rem-arvoa korvaamaan selainten oletusasetuksia saadaan sivuston kokonaisuus pidettyä oikein skaalautuvana. Rebootilla myös pyritään välttämään käyttämästä yläpuolen marginaalia elementeillä ja kirjasinkokoa arvoina CSS-säännöissä.

Versiossa 4 Bootstrap on lopettanut oletusarvoisten web-kirjasimien käyttämisen ja korvannut kirjasimet käyttäjärjestelmien omia oletuskirjasimia. Tällä pyritään takaamaan sivustolla vierailijoiden käyttökokemuksen mukavuus käyttämällä kirjasimia, jotka löytyvät asennettuina Windowsilta, Macilta, Linuxilta ja muilta yleisimmin käytetyiltä käyttöjärjestelmiltä. Mikäli mitään asetetuista kirjasimesta ei kuitenkaan löydy vierailijan käyttämästä käyttöjärjestelmästä, Reboot käyttää käyttöjärjestelmän itse asettamaa sans serif -oletuskirjasinta. Bootstrap asettaa oletuksena kaikki elementit perimään kirjasimen <body>-tagilta.

Reboot asettaa myös muutamien muiden yleisesti käytettyjen elementtien oletusarvoja uudelleen luodakseen yhtenäisen rungon sivuston kehitykselle.

4.5 Käyttö tuotannossa

Ben Frain toteaa kirjassaan *Responsive Web Design with HTML5 and CSS3*, että CSS-sovelluskehysten käyttöä tulisi välttää tuotannossa (s. 267). Hänen mukaansa sovelluskehysten käyttäminen tuotannossa lisää turhaa käyttämättömän koodin määrää projektissa. Toisena syynä hän toteaa sovelluskehysten, kuten Bootstrapin, suosion johtavan siihen,

että toteutetut sivustoprojektit näyttävät liian samankaltaisilta. Viimeisenä argumenttinaan hän toteaa, että kopioitaessa samaa valmista koodia omaan projektiin, projektin kehittäjä ei arvosta tarpeeksi kaikkea, mitä koodi todellisuudessa tekee.

Frainin toteamukset ovat toki totta, mutta Wordpressin kanssa toimittaessa hänen mainitsemansa ongelmat eivät ole merkittäviä. Wordpressin kanssa käytetään yleisesti välimuistisäosia ja CSS:n sekä Javascriptin minimointi-työkaluja. Näillä pienennetään tiedostojen kokoja ja optimoidaan niiden käyttö uuden vierailijan saapuessa sivustolle. Bootstrapin lisäämä kuorma Wordpress-sivuston lataamisessa on huomaamaton.

Graafisen suunnittelun näkökulmasta olen Frainin kanssa samaa mieltä siitä, että samoilla sovelluskehysillä toteutetut sivustot näyttävät helposti samanlaisilta. Usein internetiä selatessa vastaan tulee sivusto, josta näkee heti päällepäin, että sen taustalla pyörii Bootstrap. Tämä kuitenkin johtuu pääasiassa siitä, että sivuston tekijä on päätenyt käyttämään ainoastaan Bootstrapin tarjoamia oletustyyliä eikä ole käyttänyt aikaa ja vaivaa omien tyylien suunnitteluun. Samannäköisyys ei siis johdu Bootstrapin käytöstä vaan sivuston toteuttajan laiskuudesta.

Koodin arvostamisesta voidaan olla montaa eri mieltä. Toiset kehittäjät arvostavat vain omia luomuksiaan ja eivät luota muiden tekemiin sovelluskehysiin, koska eivät välttämättä tiedä mitä joka käänneessä tapahtuu. Tämän opinnäytetyön kannalta kuitenkin Bootstrap-sovelluskehys on mielestäni oikea valinta. Sen kehityksessä on keskitytty laite- ja selaintukeen, responsiivisten sivustojen skaalautuvuuteen ja käyttäjäkokemuksen parantamiseen. Suunniteltaessa teemaa, jota on tarkoitus käyttää useamman projektin taustalla tulevaisuudessa, on tärkeää, että nämä edellä mainitut ominaisuudet toimivat. Teeman toteuttajana toimii yksi henkilö, joten mielestäni on järkevää käyttää hyödyksi tuhansien ihmisten hyväksi toteamaa sovelluskehystä tukemassa teeman kehitystä.

4.5.1 Responsiivisuus

Responsiivinen suunnittelu web-kehityksessä on vielä melko tuore termi. ”Responsiivinen web-suunnittelu” on termi, jota käytti ensimmäisen kerran Ethan Marcotte vuonna 2010 (Frain 2015, 2.) Käytännössä termillä tarkoitetaan web-suunnittelua, jossa verkkosivustosta luodaan yksi versio, joka skaalautuu käyttäjän laitteen näyttökoon mukaan.

Responsiivisessa web-suunnittelussa CSS-sääntöjä asetetaan mediakyselyjen sisään. Mediakyselyllä asetetaan ensin haluttu näyttökoko, johon sen sisältämät CSS-säännöt pätevät. Bootstrapin käyttämät mediakyselyt näkyvät kuvassa 6.

```
// Small devices (landscape phones, 576px and up)
@media (min-width: 576px) { ... }

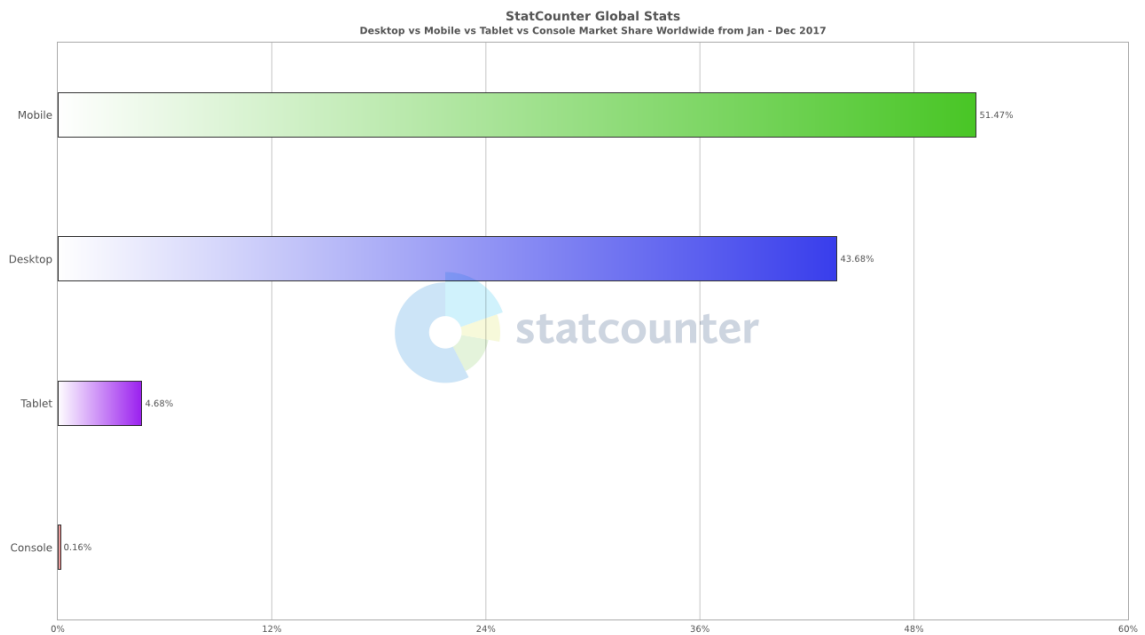
// Medium devices (tablets, 768px and up)
@media (min-width: 768px) { ... }

// Large devices (desktops, 992px and up)
@media (min-width: 992px) { ... }

// Extra large devices (large desktops, 1200px and up)
@media (min-width: 1200px) { ... }
```

KUVA 6: Bootstrapin käyttämät mediakyselyt

Bootstrap käyttää Mobile First -lähestymistapaa pohjanaan. Mobile Firstillä tarkoitetaan sitä, että ensin suunnitellaan sivuston ulkoasu pienimmälle näyttökoolle ja siitä lähdetään suunnittelemaan sivustoa isommille näyttökoille. Internetin selaaminen mobiililaitteilla on kasvanut räjähdysmäisesti viimeisen kymmenen vuoden aikana ja jo vuonna 2017 maailmanlaajuisesti internetin liikenteestä yli 56 % tapahtui älypuhelimilla ja tableteilla. Vain reilut 43 % liikenteestä tapahtui käyttäen tietokoneita (Kuva 7). Tästä voidaan siis päätellä, että verkkosivujen optimointi mobiililaitteille on erittäin tärkeää.



KUVA 7: Tietokoneiden, tablettien, mobiililaitteiden ja konsolien osuudet internetin käyttäjien laitteista vuonna 2017 (Kuva: StatCounter 2018)

4.5.2 Taulukot

Responsiivisen web-suunnittelun yhden ison haasteen muodostavat taulukot. Pienet taulukot, jotka sisältävät vähän dataa, skaalautuvat useimmiten hyvin jokaiselle näyttökoolle. Kun aletaan puhua leveistä taulukoista, joissa on käytetty paljon sarakkeita, päästään ongelman ytimeen. Internetin käyttäjät selaavat mobiilisivuja yleisesti pitäen puhelinta pystyasennossa. Näytön leveys on siis ongelmallinen, kun käsitellään taulukoita, joissa on jopa kymmeniä sarakkeita.

Bootstrap käyttää responsiivisten taulukoiden käsittelyssä vaakatasoon lisättävää vierityspalkkia (Kuva 7). Kapealla näytöllä sormella on helppo vierittää taulukon tietoja oikealta vasemmalle ilman, että taulukon rakenne hajoaa. Tämä ratkaisu ei välttämättä ole kaikista kaunein valinta, mutta web-suunnittelussa on tehtävä kompromisseja ja tässä tapauksessa käyttäjäystävällisyys menee kauneuden edelle. Paremmän näköisen ratkaisun voisi toteuttaa erottamalla jokaisen rivin omalla otsikollaan ja katkaisemalla pitkän rivin useammalle riville mobiilinäkymässä (Kuva 9). Tämä ratkaisu ei kuitenkaan toimi, jos taulukko sisältää muuttuvaa, eripituista dataa (Arora 2017.) Mielestäni Bootstrap on valinnut parhaan mahdollisen lähestymistavan taulukoiden käsittelyyn.

#	Heading	Heading	Heading	Heading	Heading	Heading
1	Cell	Cell	Cell	Cell	Cell	Cell
2	Cell	Cell	Cell	Cell	Cell	Cell
3	Cell	Cell	Cell	Cell	Cell	Cell

KUVA 8: Bootstrapin käyttämä vaakatason vierityspalkki taulukoissa

Työpöytä

Year	First Name	Last Name	Section	Roll number	Marks	Position	Discipline	Sports	Weight	Height	Attendance
Ist	Foo	bar	II	1209112002	95/100	1	A	B	60	5'	95%
IInd	James	Doe	II	1209112002	85/100	2	C	B	69	6'	85%
IIIRD	Jennifer	Jen	III	1209112006	89/100	3	A	B	72	6'	90%

Mobiili

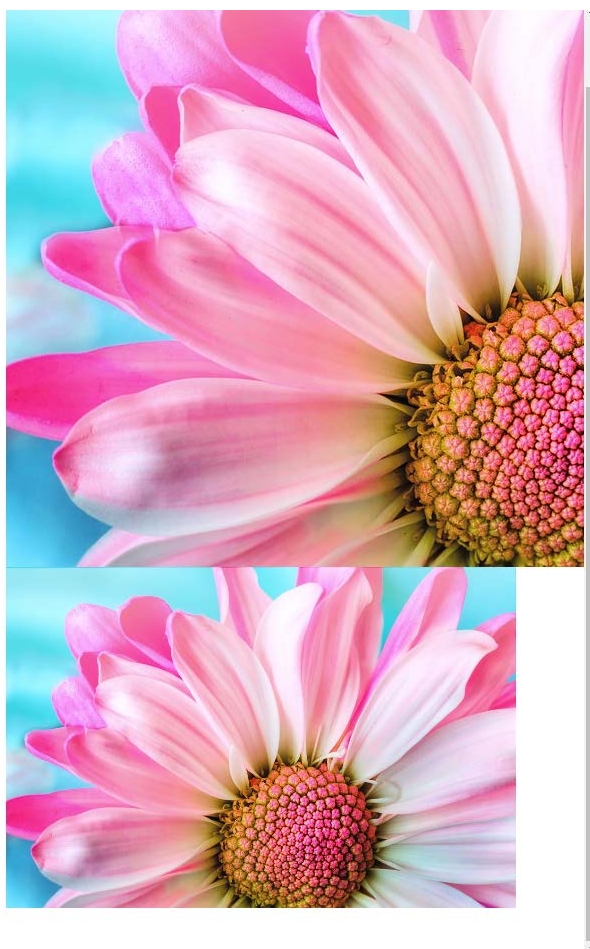
Ist		
First Name Foo	Last Name bar	Section II
Roll Number 1209112002	Marks 95/100	Position 1
Discipline A	Sports B	Weight 60
Height 5'	Attendance 95%	
IInd		
First Name James	Last Name Doe	Section II
Roll Number 1209112002	Marks 85/100	Position 2
Discipline C	Sports B	Weight 69
Height 6'	Attendance 85%	
IIIRD		
First Name Jennifer	Last Name Jen	Section III
Roll Number	Marks	Position

KUVA 9: Taulukon rivien erottaminen otsikoilla ja pitkien rivien katkaisu mobiilinäky-
mässä

4.5.3 Kuvat

Kuvien responsiivisuus on myös yksi tärkeimmistä asioista suunniteltaessa verkkosivua. Lähtökohtaisesti mobiilinäkymään kuvien halutaan skaalautuvan mahdollisimman isoina ruudulle, mutta jos kuvan pakottaa koko ruudun levyiseksi, se näyttää kömpelöltä.

Bootstrapin ratkaisu responsiivisiin kuviin on käyttää CSS:n sääntöjä `max-width: 100 %` ja `height: auto`. `Width: 100 %` pakottaa kuvan leveyden aina 100 % leveyteen sen isäntäelementin leveydestä. Eri näyttökoissa sitten asetetaan tämän isäntäelementin leveys perustuen käytössä olevaan mediakyselyyn. `Height: auto` -säännöllä kuva määrittää automaattisesti oman korkeutensa säilyttämällä alkuperäisen kuvan kuvasuhteen. Kuvat saa responsiivisiksi käyttämällä CSS-luokkaa `.img-fluid` niissä kuvissa, jotka halutaan responsiivisiksi. Kuvassa 10 on kuvattuna `.img-fluid` luokan toiminta.



KUVA 10: Alempaan kuvaan on liitetty `.img-fluid` luokka, joten se skaalautuu näyttökoon mukaan. Ylempi kuva näyttää suurella näytöllä hyvältä, mutta pienellä näytöllä kuva jatkuu ruudun ulkopuolelle.

Valmiina Bootstrap tarjoaa myös niin sanotun pienoiskuvan tyylyttelyn. Pienoiskuva on leikattu, yleensä neliön muotoinen, kuva alkuperäisestä kuvasta, jolle Bootstrap asettaa pyöristetyt reunat ja yhden pikselin vahvuisen reunuksen. Pienoiskuvia käytetään usein kuvagallerioissa ja sen käyttö onnistuu lisäämällä CSS-luokan `.img-thumbnail` niille kuville, joista halutaan tehdä pienoiskuvat.

4.5.4 Helppokäyttöisyys ja käytettävyys

Hyvässä web-suunnittelussa on otettava huomioon sivuston helppokäyttöisyys ja käytettävyys. Tällä tarkoitetaan useimmiten sivuston optimointia myös erilaisille lukulaitteille sopiviksi. Sivusto, jolla sisältöön olennaisesti liittyvät tekstit ovat vain kuvamuodossa on näkövammaiselle samanlainen este kuin liian kapea ovi pyörätuolin käyttäjälle (TIEKE 2017).

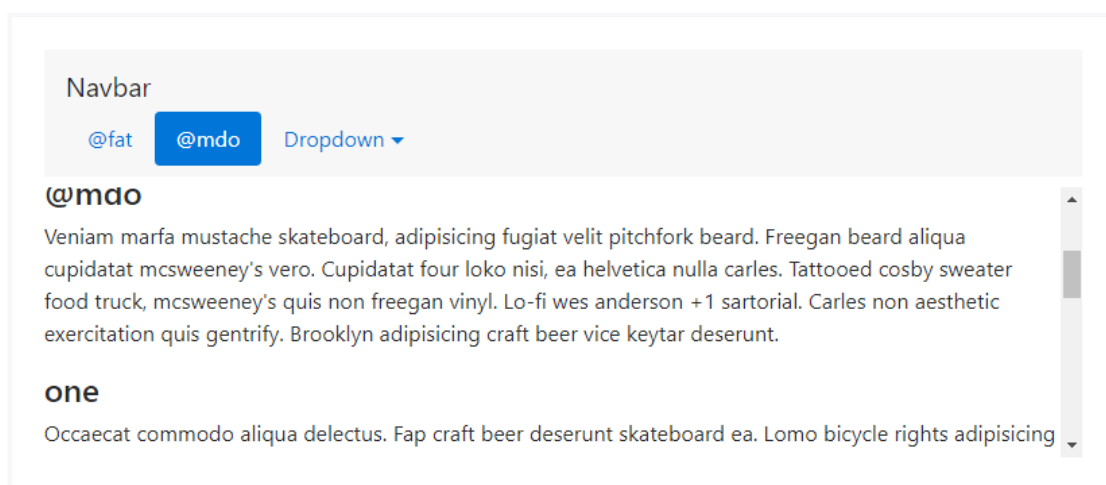
ARIA (Accessible Rich Internet Applications Suite) on World Wide Web Consortiumin (W3C) kehittämä standardi, joka määrittää, miten web-sisällön ja web-aplikaatiot saadaan paremmin vammaisten käytettäväksi. ARIA on suunniteltu erityisesti auttamaan kehittyneempien web-sivustojen käyttöliittymien käyttämisessä (W3C 2017).

Bootstrapin kehityksessä on otettu huomioon lukulaitteet. Komponentit, joita on tarkoitus koskettaa kuten alavetovalikot, ponnahdusikkunat ja avustavat tekstit on kaikki suunniteltu kosketusnäytöille ja näppäimistöllä ja hiirellä käytettäväksi. Näihin komponentteihin on liitetty ARIA-roolit, jotka tekevät ruudunlukulaitteille helpoksi tunnistaa, mikä komponentti on kyseessä.

Ainoastaan ruudunlukulaitteille suunnatut elementit merkataan Bootstrapissa CSS-luokalla `.sr-only` (screen reader only). Tätä CSS-luokkaa käytetään sellaisissa elementeissä, joiden ei haluta näkyvän web-sivuilla, mutta jotka lukulaitteiden halutaan silti lukevan. Yksi käyttöesimerkki on linkki, jolla lukulaite voi hypätä sivuston alkutekstien yli suoraan asiasisältöön.

4.5.5 Scrollspy

Scrollspy on Bootstrapin käyttämä lisäosa, joka nimensä mukaisesti ”vakoilee” käyttäjän skrollausta. Sitä käytetään yleensä päivittämään navigaatioon tyylejä, kun käyttäjä rullaa sivua eteenpäin. Tämä on hyvin yleisesti käytetty tekniikka niin sanotuilla one pager -sivuilla, joilla kaikki sisältö latautuu yhdellä sivulla ja kun navigaatiosta valitaan linkki, sivu vain skrollaantuu oikeaan kohtaan. Monilla sivuilla käytetään navigaatiossa värejä tai muita efektejä ilmaisemaan nykyistä sivua tai kohtaa sivulla.



KUVA 11: Scrollspy päivittää navigaatioon sinisen taustavärin oikeaan kohtaan sitä mukaan, kun sivua skrollataan

Scrollspy:sta ei ole hirveästi hyötyä Trivalin teeman kehityksessä, sillä teema ei tue yhden sivun toteutuksia. Se on kuitenkin hyötynsä takia mainitsemisen arvoinen Bootstrapin ominaisuus, joka antaa Bootstrapille lisäarvoa. Trivalin teema ei tue yhden sivun toteutuksia, sillä niiden hakukoneoptimointi on hankalampaa verrattuna monen sivun toteutukseen.

5 TRIVAL-TEEMA

5.1 Kehitys

Teeman kehitys alkoi luomalla puhtas WordPress-asennus omalle palvelimelle väliaikaiseen kehitysympäristöön. Ensin teemalle luodaan oma kansio wp-content/themes -kansioon sisälle. Tämän teeman nimeksi tuli lopulta WP Bootstrap Trival, joten kansion nimeksi tuli wp-bootstrap-trival. Luvussa kaksi käsiteltiin WordPress-teeman suunnittelua yleisellä tasolla ja tässä luvussa keskitytään enemmän itse opinnäytetyössä syntyneen teeman sisältöön.

 404.php	Jul 4, 2017 09:49 AM
 animated-search.php	Aug 15, 2017 01:41 PM
 archive.php	Jun 29, 2017 10:53 AM
 comments.php	Jun 29, 2017 10:53 AM
 custom-editor-style.css	Jul 20, 2017 02:07 PM
 footer-some.php	Aug 9, 2017 10:05 AM
 footer-widget.php	Jun 29, 2017 10:53 AM
 footer.php	Sep 19, 2017 02:28 PM
 fullwidth.php	Aug 4, 2017 01:01 PM
 functions.php	Sep 19, 2017 03:21 PM
 header.php	Sep 19, 2017 02:54 PM
 index.php	Jun 29, 2017 10:53 AM
 left-sidebar.php	Aug 4, 2017 01:01 PM
 LICENSE	Jun 29, 2017 10:53 AM
 page.php	Jun 29, 2017 10:53 AM
 readme.txt	Jul 5, 2017 11:16 AM
 screenshot.png	Jun 29, 2017 11:04 AM
 search.php	Jul 4, 2017 09:50 AM
 searchform.php	Aug 9, 2017 11:27 AM
 sidebar.php	Jun 29, 2017 10:53 AM
 single.php	Jun 29, 2017 10:53 AM
 style.css	Sep 19, 2017 02:58 PM

KUVA 12: Trival-teeman sisältämiä rakennetiedostoja

Dokumentissa ensin todetaan, että käytetään HTML5:tä kirjoittamalla `<!DOCTYPE html>` ensimmäiselle riville. Tämän jälkeen aukaistaan HTML-dokumentti `<html>`-tagilla. `<html>`-tagiin on sisällytetty dynaamisesti haettava tieto sivuston käyttämästä kielestä käyttämällä WordPressin omaa `language_attributes()`-funktioita. Julkiselle sivustolle tämä tulostaa koodin `<html lang="fi">` edellyttäen, että WordPressin asetuksista on valittu kieleksi suomi.

`<head>`-tagin sisältä löytyy metatietoja ja WordPressin oma funktio `wp_head()`, joka hakee kaikki muut tiedot, jotka on tarkoitettu tulostuvan `<head>`-tagien sisälle kuten CSS- ja JavaScript-tiedostojen linkitykset. Myös sivustolla käytetyt Googlen kirjasimet linkitetään `<head>`-tagin sisään.

`<body>`-tagin sisälle luodaan sivuston käyttäjälle näkyvä sisältö. WordPressin funktiolla `body_class()` haetaan tälle tagille kaikki sen vaatimat CSS-luokat. CSS-luokat haetaan dynaamisesti, jotta myöhemmin voidaan helposti yksilöidä, mikä sivu on kyseessä. WordPress käyttää sivujen kanssa yksilöivää numerointia ja sivujen tunnistenumerot tulostetaan juuri `body`-tagin sisälle. Esimerkiksi jos halutaan ainoastaan tietyllä sivulla tehdä tietty CSS-sääntö, voidaan se kohdistaa sivun tunnistenumeroilla aktivoitumaan ainoastaan kyseisellä sivulla. `Header.php` tiedosto sisältää sivuston logon, navigaation ja kaiken muun sisällön, joka on näiden elementtien yläpuolella.

5.2.2 Sisältö

Sivuston sisältöalue löytyy erillisestä tiedostosta. Sisällön käyttämä ulkoasutiedosto muuttuu sen perusteella, mikä sivupohja sivulle on valittuna WordPressin asetuksissa. Trivalin teemaan olen luonut ensimmäiseen versioon kolme eri sivupohjaa: `fullwidth.php`, `left-sidebar.php` ja `page.php`. `Page.php` on oletus-sivupohja, joka jakaa sisältöalueen kahteen sarakkeeseen. Vasemmanpuoleisessa sarakkeessa on sivuston sisältö ja oikeanpuoleisessa sarakkeessa sivupalkki. `Left-sidebar.php` on muuten samanlainen kuin `page.php`, mutta sivupalkki sijaitsee vasemmalla ja sivuston sisältö on oikealla puolella. `Fullwidth.php` taas luo sivulle ainoastaan yhden sarakkeen, jonka sisällä on sivuston sisältö. `Fullwidth.php` ei sisällä ollenkaan sivupalkkia. Kuvassa 14 on kuvattuna `fullwidth.php`-tiedoston sisältö.

```

<?php
/**
 * Template Name: Koko leveys ilman sivupalkkia
 */

get_header(); ?>

<section id="primary" class="content-area col-sm-12">
  <main id="main" class="site-main" role="main">

    <?php
    while ( have_posts() ) : the_post();

      get_template_part( 'template-parts/content', 'page' );

      // If comments are open or we have at least one comment, load up the comment template.
      if ( comments_open() || get_comments_number() ) :
        comments_template();
      endif;

    endwhile; // End of the loop.
  ?>

  </main><!-- #main -->
</section><!-- #primary -->

<?php
get_footer();

```

KUVA 14: Fullwidth.php-sivupohjan sisältö

Fullwidth.php-tiedoston alussa kutsutaan ensin sivuston yläosa header.php tiedostosta käyttämällä WordPressin funktiota `get_header()`. Tämän jälkeen luodaan `<section>`- ja `<main>` -tageilla sisältöalue. Sisältöalueen sisällä haetaan sivulle sisältö käyttämällä WordPressin Loop-toimintoa (WordPress.org, Codex – The Loop n.d.). Loopissa haetaan sisältö-alueen rakenne vielä tiedostosta `content-page.php`, joka sijaitsee teeman sisällä kansiossa `template-parts`. Loopin jälkeen tagit suljetaan ja haetaan sivuston alaosan sisältö tiedostosta `footer.php` käyttämällä funktiota `get_footer()`.

5.2.3 Footer

Footer.php-tiedosto on viimeinen, jonka selain käsittelee piirtäessään nettisivustoa. Alatunnisteessa suljetaan ensin tagit, jotka on aukaistu sisältöalueen tiedostoissa ja sen jälkeen tarkistetaan, onko vimpainalue aktiivisena. Jos vimpainalue on aktiivinen, footer.php tulostaa sivulle vimpainalueen, jonka olen luonut teemaan alapalkkiin sijoitettavia elementtejä varten. Vimpainalueelle on helppo lisätä ja muokata elementtejä WordPressin hallinnasta. Kuvassa 15 on kuvattu footer.php tiedoston sisältö.

```

<?php if(!is_page_template( 'blank-page.php' ) && !is_page_template( 'blank-page-with-container.php' )): ?>
    </div><!-- .row -->
    </div><!-- .container -->
</div><!-- #content -->
<!-- ALAPALKKI-WIDGET -->
<?php if ( is_active_sidebar( 'footer-4' ) ) : ?>
    <div id="footer_widget" class="widget-area">
        <?php
            $showFooterWidget = get_theme_mod('fwidget_onoff_setting');

            // If option is selected to only show on homepage
            if ($showFooterWidget == '1') {
                // If visitor is on homepage
                if (is_front_page()) {
                    // Print widget area
                    dynamic_sidebar( 'footer-4' );
                }
            } else {
                dynamic_sidebar( 'footer-4' );
            }
        ?>
    </div>
<?php endif; ?>
<!-- ALAPALKKI-WIDGET -->
<footer id="colophon" class="site-footer" role="contentinfo">
<?php get_template_part( 'footer-widget' ); ?>
    <div class="container social-media">
        <?php get_template_part( 'footer-some' ); ?>
    </div>
    <div class="container">
        <div class="site-info">
            <?php echo get_theme_mod('footer_text_setting'); ?>
        </div>

```

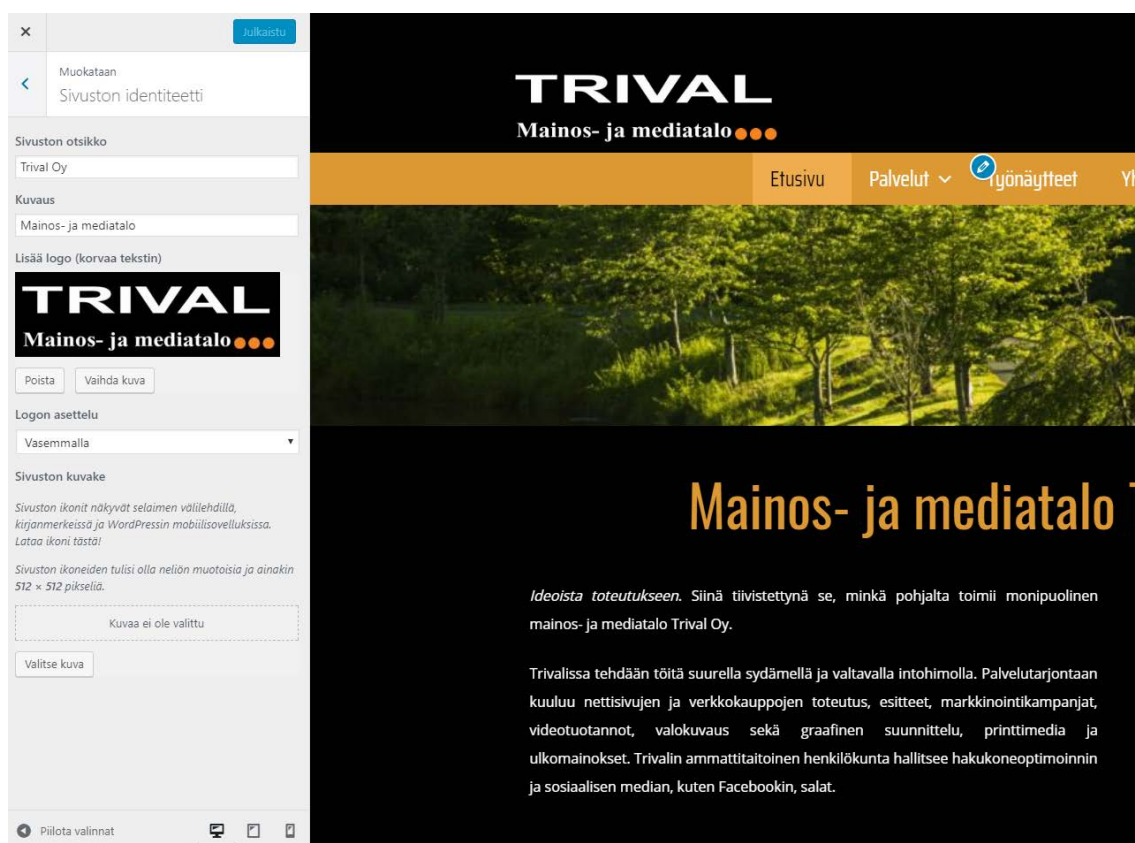
KUVA 15: Footer.php-tiedoston sisältöä

Alapalkkiin on myös upotettu sosiaalisen median kuvakkeet, joita pääsee muokkaamaan helposti WordPressin mukauttimen avulla. Mukauttimesta lisää seuraavassa luvussa. Sosiaalisen median painikkeiden jälkeen alaosassa on enää jäljellä tekijänoikeustekstille varattu paikka, jota sivuston ylläpitäjä pääsee myös muokkaamaan helposti WordPressin ohjausnäkyvästä graafisesta käyttöliittymästä. Aivan lopuksi vielä ennen <body>- ja <html>-tagien sulkemista footer.php tiedostossa kutsutaan wp_footer()-funktioita, joka toimii vastaavalla tavalla kuin wp_header()-funktio, mutta se hakee vuorostaan sivuston alaosaan tulevat mahdolliset JavaScript-koodit.

5.3 Mukautin

Wordpressin yksi keskeisimpiä toimintoja on Mukauta-työkalu. Sen avulla pystytään muokkaamaan sivuston ulkoasua reaaliajassa graafisesta käyttöliittymästä. Mainostoimiston tarpeisiin mukautin on todella kätevä, sillä loppukäyttäjien eli mainostoimiston asiakkaiden on joissain tapauksissa hyvä päästä myös itse muokkaamaan sivuston ulkoasua. Kun muutostyökalut on toteutettu mukauttimeen, ei loppukäyttäjän ole mahdollista hajottaa sivuston rakennetta samalla lailla kuin tilanteessa, jossa asiakas itse muokkaisi suoraan koodia.

Rakennettaessa teemaa pystytään mukauttimeen lisäämään eri toimintoja. Yleisimpiä teemoissa nähtyjä toimintoja ovat sivuston identiteettiin liittyvät asiat kuten logo, navigaation muokkaus tai kirjasinten valinta. Mukauttimesta päästään myös suoraan muokkaamaan vimpaimia. Edistyneemmille käyttäjille mukauttimesta löytyy myös alue oman CSS-koodin lisäämiseksi. Trivalin tapauksessa Oma CSS -osio on tärkeä, kun teemaa aletaan käyttää uudelleen eri projekteissa ja ulkoasuun on tehtävä suurempia muutoksia, joihin mukauttimen valmiit asetukset eivät riitä. Kuvassa 16 on kuvattuna osa Trival-teeman mukauttimesta.



KUVA 16: Trival-teeman mukautin

6 POHDINTA

Yhden aikaisemman WordPress-sivuston toteuttaneena ajatus oman teeman luomisesta tuntui alkuun todella isolta urakalta. Dokumentaatioita lukiessa kuitenkin alkoi syntyä selkeämpi ajatus siitä, miten teeman tekninen toteutus tapahtuu. Kun toimeksiantajan tarpeet ja vaatimukset teeman suhteen olivat ennalta tiedossa, oli helpompi lähteä suunnittelemaan oman teeman kokonaisuutta.

Kehitystyö alkoi kesällä 2017 ja alkuun jouduin perehtymään ihan teeman teon ensimmäisiin askeliin ja WordPressin sisäiseen toimintaan ja logiikkaan. Pienten askelten siivittämänä teema alkoi pikkuhiljaa rakentua ja muotoutua lopulliseen muotoonsa. Kun pääsin mukauttimen muokkaamiseen, ideoita uusien asetusten lisäämiseen tuli koko ajan lisää ja jossain vaiheessa oli pakko vain lopettaa uusien toimintojen lisääminen, jotta teeman pystyi julistamaan valmiiksi. Uusien toimintojen lisääminen mukauttimeen on yksi teeman jatkokehityksen tavoite.

Teeman kehityksen ohessa tein muita työtehtäviä samalla ja aikataulu oli aika kiireinen. Alkuperäisestä deadlinesta opinnäytetyön valmistuminen venyi reilulla kuukaudella. Opinnäytetyön tuotoksena syntynyt teema on raportin kirjoitusvaiheessa 95 % valmis. Sitä ei kuitenkaan voi vielä ottaa käyttöön tuotannossa, koska WordPressin mukauttimeen tuli iso päivitys versiossa 4.9. Päivitys julkaistiin, kun teeman kehitys oli tauolla muiden työkiireiden vuoksi. Muutoksen jälkeen mukauttimella tehdyt muutokset eivät tallennu sivustolle. Tämä on jatkokehityksen kannalta ensimmäinen ja tärkein asia, joka täytyy korjata.

LÄHTEET

Arora, A. 2017. Responsive Design in Tables. Luettu 10.12.2017.

<https://www.codementor.io/aayusharora/responsive-design-in-tables-7f5e3hspy>

Bootstrap Documentation: Accessibility. Luettu 5.1.2018.

<https://getbootstrap.com/docs/4.0/getting-started/accessibility/>

Bootstrap Documentation: Browsers and devices. Luettu 14.11.2017.

<http://getbootstrap.com/docs/4.0/getting-started/browsers-devices/>

Bootstrap Documentation: Reboot. Luettu 20.12.2017.

<https://getbootstrap.com/docs/4.0/content/reboot/>

Cooper, M. 2016. WAI-ARIA Overview. Luettu 5.1.2018

<https://www.w3.org/WAI/intro/aria>

Frain, B. 2015. Responsive Web Design with HTML5 and CSS3. Birmingham: Packt Publishing.

Friedman, J. 2013. Web Designer's Guide to WordPress: Plan, Theme, Build, Launch. Berkeley, CA: New Riders.

Król, K. 2015. WordPress 4.x Complete. Packt Publishing.

Liew, Z. 2016. REM vs EM – The Great Debate. Luettu 4.1.2018.

<https://zellwk.com/blog/rem-vs-em/>

Tietoyhteiskunnan kehittämiskeskus ry. Helppokäyttöisyys. Luettu 5.1.2018.

<https://www.tieke.fi/pages/viewpage.action?pageId=3441008>

Wikipedia. CSS Framework. Luettu 11.3.2018.

https://en.wikipedia.org/wiki/CSS_framework

Wikipedia. Reset style sheet. Luettu 11.3.2018.

https://en.wikipedia.org/wiki/Reset_style_sheet

WordPress.org. Codex – The Loop. Luettu 20.12.2017.

https://codex.wordpress.org/The_Loop

WordPress.org. Theme handbook. Luettu 22.12.2017.

<https://developer.wordpress.org/themes/>