

Opinnäytetyö (AMK)

Liiketalouden koulutusohjelma / Tietojenkäsittely

NLIIBK14

2016

Jere Sumell

# RESPONSIIVISEN BOOTSTRAP -TEEMAN LUONTI WORDPRESS-SISÄLLÖNHAL- LINTAJÄRJESTELMÄÄN

-Tapaus Vasaramäen tilitoimisto Oy

Jere Sumell

# RESPONSIIVISEN BOOTSTRAP- TEEMAN LUONTI WORDPRESS SISÄLLÖNHAL- LINTAJÄRJESTELMÄLLE

- Tapaus Vasaramäen Tilitoimisto Oy

Eryteisesti erilaisten älylaitteiden suosion kasvun myötä ihmiset selaavat verkkosivuja erikokoisilla näyttöpäätteillä. Responsiivinen verkkosuunnittelu on nykyaikainen menetelmä toteuttaa verkkosivu, mikä toimii usean kokoisessa laitteessa, samalla kun yritys saavuttaa taloudellista hyötyä siitä, että toteutetaan vain yhden verkkosivut.

Yrityksen työntekijöillä ei välttämättä ole HTML-osaamista, joten on järkevää valita kotisivujen alustaksi jokin julkaisujärjestelmän tyylinen sisällönhallintajärjestelmä. Sisällönhallintajärjestelmät sallivat erilaisia käyttäjärooleja ja tästä johtuen myös useampi ylläpitäjä on mahdollinen. Tässä opinnäytetyössä perehdyttiin yhteen sisällönhallintajärjestelmään, ja yhteen front-end -kehityskehikseen ja näiden yhteensovittamiseen. Asiaan perehtyminen oli konstruktivistista, ja se tapahtui tutustumalla alan kirjallisuuteen sekä Internet-lähteisiin.

Tuloksena saatiin käsitys responsiivisesta www-suunnittelusta, perustietoutta sisällönhallintajärjestelmistä, ja työn edetessä myös PHP-kieli tuli tutummaksi.

Tuloksena yritys innostui sisällönhallintajärjestelmän käyttöönotosta sekä etsimään vaihtoehtoisia teemoja kotisivuilleen.

## ASIASANAT:

julkaisujärjestelmät, CMS, sisällönhallinta, WordPress, Bootstrap, kehys, www-kehitys, teema

JERE SUMELL

# CREATING RESPONSIVE WORDPRESS -THEME TAKING ADVANTAGE ON TWITTER BOOTSTRAP -FRAMEWORK

-CASE-STUDY VASARAMAEEN TILITOIMISTO OY

After a massive growth in the popularity of different types of smart devices, people are browsing the Internet with devices with screens of various sizes. Responsive WEB development is the modern solution to create company websites compatible with all viewing devices. At the same time, companies save resources when their WEB pages are created with responsive WEB design because employees do not necessarily have expertise in creating pure HTML code. For this reason, it is a good idea for companies to use a Content Management System (CMS) as a website platform where employees can work with different roles in the CMS.

The purpose of this thesis was to create a functional, ready to use, responsive CMS theme for the client organization, a small-sized accounting firm. CMS platforms were researched and WordPress was chosen because of its leading market position combined with Twitter Bootstrap as a front-end framework.

The result was a simple, ready-to-use, responsive WordPress theme that the client organization can start using.

## KEYWORDS:

CMS, WordPress, Bootstrap, framework, web development, theme.

# SISÄLTÖ

<b>KÄYTETYT LYHENTEET</b>	<b>7</b>
<b>1 JOHDANTO</b>	<b>7</b>
1.1 Tehtävän määrittely	7
1.2 Vasaramäen tilitoimisto Oy:n esittely	8
<b>2 CMS-SISÄLLÖNHALLINTAJÄRJESTELMÄ</b>	<b>9</b>
2.1 Sisällönhallintajärjestelmän käytön edut liiketoiminnan kannalta	10
2.2 WordPress	10
2.3 WordPressin asennus ja käyttöönotto	11
2.4 WordPressin ympärille rakennettu liiketoiminta Suomessa	11
<b>3 RESPONSIIVINEN WEB-SUUNNITTELU</b>	<b>13</b>
<b>4 RESPONSIIVISEN BOOTSTRAP WORDPRESS-TEEMAN TOTEUTTAMINEN SEKÄ KÄYTTÖÖNOTTO</b>	<b>17</b>
4.1 WordPress-tiedostorakenteesta	17
4.2 1 – Tavallisen HTML5-dokumentin pohjakoodin luonti	18
4.3 Vaihe 2 - Twitter Bootstrap koodia HTML5-tiedostoon.	18
4.4 GRID-taulukointijärjestelmä	19
4.5 Sivuston tekninen toteutus	21
4.6 Vaihe 4 – Tavallisen CSS3-tyylitiedoston luontia	23
4.6.1 CSS3-mediakyselyistä	24
4.7 Vaihe 5 – HTML-koodin sijoitus oikeisiin WordPressin php –tiedostoihin	24
4.8 Vaihe 6–Varsinaisen sisällön ohjelmointi	24
4.9 Vaihe 7 – Uuden teeman käyttöönotto	25
<b>5 KOETUT ONGELMAT</b>	<b>27</b>
<b>6 YHTEENVETO</b>	<b>29</b>
<b>7 LÄHTEET</b>	<b>31</b>

## KUVAT

Kuva 1. Kappaleen relatiivisen sijainnin ja kuvan absoluuttisen sijainnin määrittely CSS-tyylitiedostossa.	14
Kuva 2. Esimerkki CSS-tyylimäärittelystä, jolla voidaan piilottaa kuva näytöltä tietyn kokoisella laitteella ilman kuvan lataamista.	15
Kuva 3. Esimerkki navigointilistan toteuttamisesta CSS-tiedostossa.	16
Kuva 4. WordPress-tiedostorakenne.	17
Kuva 5. HTML-pohja.	18
Kuva 6. Kaksi pylvästä Containerin sisään.	20
Kuva 7. Kustomoitava otsikkokuva käyttöön.	22
Kuva 8. WordPress-sivujen dynaaminen listaus.	22
Kuva 9. Mediakyselyt CSS3-tiedostossa.	23
Kuva 10. Sisällön ohjelmointia.	25
Kuva 11. WordPress-valikko.	25
Kuva 12. WordPress-teeman käyttöönottoikkuna.	26

## KÄYTETYT LYHENTEET

CMS	sisällönhallintajärjestelmä, Content Management System; sisällön julkaisun, arkistoinnin ja muun ylläpitämisen helpottamiseksi käytetään useita eri sisällönhallintajärjestelmiä, tässä opinnäytetyössä keskitytään markkinajohtajaan WordPress-järjestelmään
CSS	porrastetut tyyliarkit eli HTML-kielessä esitetyn rakenteen ehdotettu ulkoasu, mikä voi sijaita itsessään HTML-tiedostossa tai ulkoisessa tyylitiedostossa, Cascading Style Sheets
Framework	kehikko, kehys, viitekehys, runko, runkorakenne, puite, käytän tässä opinnäytetyössä termiä "kehys": yleisesti kehys on aito tai konseptuaalinen rakenne , joka keskittyy tarjoamaan tukea tai opastusta jonkin hyödyllisen rakennelman laajentamiseksi
Front-end	Web-suunnittelussa front-end viittaa ulkoasuun, julkisivuun, miltä sivusto näyttää loppukäyttäjän silmille
GPL	vapaiden ohjelmien julkaisemiseen tarkoitettu lisenssi
HTML	hypertekstin merkintäkieli, HTML-kielellä kuvataan dokumentin, eli www-sivuston rakenne: Hypertext Markup Language
Javascript	ohjelmointikieli www-sivuston toiminallisuuden ohjelmointiin
Open Source	vapaasti saatavilla ja kehitettävissä oleva avoin lähdekoodi

PHP	olio-ohjelmointikieli, jonka avulla voidaan valmistaa palvelin-pohjaisia olioovelluksia. opinnäytetyössä tarkasteltava WordPress on toteutettu PHP-kielellä
URL	merkkijono, jota käytetään osoittamaan verkkosivun sijainti Internetissä
WYSIWYG	What you see is what you get (Mitä näet on se, mitä lopulta saat)

# 1 JOHDANTO

## 1.1 Tehtävän määrittely

Yritysten kannalta näkyvyys verkossa on avainasemassa, ja se on jo pitkään korvannut perinteiset luettelomediat. Responsiivisuus on päivän sana www-kotisivun kehitystyössä vuonna 2016. Maailman suurin hakukoneyhtiö Google on linjannut politiikassaan, että se pistää etusijalle hakutuloksissaan nimenomaan responsiivisesti toteutetut verkkosivut (Mobile marketing engine).

Maailmalla on kehitetty lukuisia HTML-kehyskiä (frameworks) www-sivujen kehittäjille, jotta kehitystyö olisi helpompaa, ja kotisivuista saataisiin näyttävämpiä pienemmällä vaivannäöllä. Tässä opinnäytetyössä tarkasteltava ohjelmistokehys on Twitter Bootstrap. Bootstrapin kilpailijoita ovat mm. Foundation, Zimit sekä Ink.

Toisaalta yrityksen työntekijöiden kannalta voi olla mielekästä käyttää aikaa johonkin muuhun työtehtävään kuin kotisivujen jatkuvaan päivittämiseen HTML-ohjelmakoodin työstämiseen. Tähän soveltuvat sisällönhallintajärjestelmät, jotka toimivat alustana (platform), jonka päälle kotisivujen ulkoasu on rakennettu.

Tämän opinnäytetyön tavoite on tutkia Wordpress-sisällönhallintajärjestelmää ja laatia toimiva ulkoasuteema yritykselle Vasaramäen Tilitoimisto Oy käyttäen Twitter Bootstrap-kehystä. Lisäksi opinnäytteessä tarkastellaan näiden teknologioiden yhteensovittamisessa ilmenneitä etuja ja haittoja niin tekniikan, kuin liiketoiminnan kannalta. WordPress valikoitui henkilökohtaisten positiivisten kokemuksen ansiosta kyseisestä sisällönhallintajärjestelmästä, ja sen käyttöä puoltaa myös se, että se oli markkinaosuuksissa ylivoimaisesti johtava sisällönhallintajärjestelmä vuonna 2015 (Open Source CMS, 2015).

Twitter Bootstrap valikoitui tässä opinnäytetyössä käytettäväksi kehikseksi siitä syystä, että se on kasvanut maailman johtavaksi ja suosituimmaksi HTML-kehikseksi kehittäjien keskuudessa verrattain nopeassa ajassa sen ilmestymisen vuoden 2011 jälkeen (Bootstrap).



## 1.2 Vasaramäen tilitoimisto Oy:n esittely

Tämän opinnäytetyön toimeksiantajaksi varmistui Vasaramäen tilitoimisto OY, joka kaipasi uusia nettisivuja toimeksiannon hetkellä syyskuussa 2015.

Vasaramäen tilitoimisto OY:n toiminta alkoi vuonna 2006 Turussa ja sen yhtiömuoto on osakeyhtiö. Yrityksen toimitusjohtaja Jarkko Rislakki oli tuolloin vielä auktorisoidussa tilitoimistossa töissä, mutta idea luoda puolisolle mahdollisuus tehdä osa-aikaisesti töitä kotona johti yrityksen perustamiseen. Silloisessa Jarkon työpaikassa toiminta oli niin vähäistä, että sen ei katsottu synnyttävän intressiristiriitaa.

Yrityksen toiminta-ajatuksena on tarjota aivan pienille yrityksille, toiminimille ja yksinyrittäjille edullinen ja kevyt kirjanpito palvelu. Liikevaihto on viime vuosien kuluessa ollut arviolta noin 120 000 euroa, josta liikevoitto on noin 10 %, mikä kuvastaa Jarkko Rislakin mukaan alan normaalitilannetta hyvin. Liikevaihto on kasvanut noin 10 % vuodessa. Koska yrityksessä on kaksi työntekijää, yrityksen liikevaihto per työntekijä on noin 60 000 euroa.

Yritys pitää kilpailuvalttina kevyen kulurakenteen. Lisäksi Jarkko Rislakin pitkää kokemusta taloushallinnon parissa, sekä myös Jarkon kiinnostusta uuden tietotekniikan suomiin mahdollisuuksiin. Yritys sijoittuu suhteessa kilpailijoihinsa nähden kartalle juuriksi, että se tarjoaa kevyen ja edullisen vaihtoehdon pienille yrityksille.

Vasaramäen tilitoimisto ei ole käyttänyt markkinointiin juuri lainkaan varoja, vaan sen kasvu on perustunut pitkälti puskaradion toimintaan, ja siihen, että nykyiset asiakkaat ovat suositelleet kyseistä yritystä toisille yrittäjille, joille tilitoimiston löytyminen on ajankohtaista.

Yrityksen nykyiset nettisivut on perustettu vuonna 2006, jolloin Jarkko Rislakin tuttu teki ne omalle palvelimelleen. Yrityksen paineet uudistaa nettisivut juuri nyt perustuu pitkälle responsiivisuuteen, sillä Jarkko Rislakin mukaan sivuja selataan paljon myös muilta päätelaitteilta kuin pelkästään tietokonenäytöltä. Lisäksi Jarkko haluaisi saada sivut paremmin omaan hallintaan, tekstin, grafiikan ja layoutin suhteen.

Vasaramäen tilitoimiston kotisivut ovat perua vuodelta 2006, eivätkä ne käytä pohjanaan mitään sisällönhallintajärjestelmää. Tämän opinnäytetyön myötä olisi tarkoitus ottaa käyttöön responsiivinen kotisivuratkaisu, mikä käyttää pohjalla WordPress-sisällönhallintajärjestelmää.

## 2 CMS-SISÄLLÖNHALLINTAJÄRJESTELMÄ

CMS:llä (Content Management System) tarkoitetaan tietojärjestelmää, mikä on suunniteltu organisaation kokonaisvaltaiseen sisällönhallintaan, eikä niinkään keskittymään ainoastaan johonkin osa-alueeseen. Lisäksi esimerkiksi ulkoasun suunnittelu (design) on eroteltuna sisällöstä, mikä luo joustavuutta käyttäjille (The University of Kansas).

Sisällönhallintajärjestelmä (CMS) on järjestelmä, mikä sallii loppukäyttäjän lisätä, poistaa, päivittää tai muuttaa verkkosivun sisältöä hallintapaneelin kautta ilman, että tarvitsee välttämättä kajota ohjelmakoodiin (Stack Exchange, 2011).

Modernit sisällönhallintajärjestelmät ovat verkkosovelluksia, jotka sisältävät ainakin yhden tietokannan. Markkinoilla on useita CMS-järjestelmiä saatavilla (Stack Exchange, 2011). Järjestelmän valinta riippuu lähinnä käyttötarkoituksesta. Tässä opinnäytetyössä on valittuna WordPress-järjestelmä, sillä case-yritys halusi hyödyntää liiketoiminnassaan nimenomaan sitä. WordPress on alun perin suunniteltu bloggaus-tarkoitukseen, mutta se voi toimia myös staattisempienkin sisältöjen alustana. WordPress on nk. avoimen lähdekoodin sovellus (open source), ja lähdekoodi on kaikkien tarkasteltavissa vapaasti netissä ilmaiseksi. (Templatemonster) Tämä takaa myös sen edun, että ohjelma on periaatteessa kenen tahansa ohjelmointitaitoisien kehitettävissä ja paranneltavissa.

Tyypillisesti CMS käsittää kaksi elementtiä: Sisällönhallintasovelluksen (CMA) ja sisällönjakelusovelluksen (CDA) (Techtarget, 2016). Sisällönhallintasovellus (CMA) näyttyy tavallisesti käyttäjälle hallintapaneelina, sillä se on nk. ”front-end”-komponentti CMS-järjestelmässä (Techtarget,2016). WordPress tarjoaa nk. WYSIWYG-näkymän käyttäjälle hallintapaneelina, eli käyttäjä näkee editoidessaan sivustoa saman, mikä näkyy lopullisessa sivustossa sivustolla vieraillevalle. Sisällönjakelusovellus (CDA) hoitaa sisällön jakelulogiikan.

Sisällönhallintajärjestelmät on tavallisesti suunniteltu siten, että sen käyttäjiksi voidaan määritellä useita käyttäjiä erilaisine rooleineen, mikä mahdollistaa myös useamman ylläpitäjän roolissa järjestelmässä asioivan käyttäjän (Comentum, 2010).

## 2.1 Sisällönhallintajärjestelmän käytön edut liiketoiminnan kannalta

Perinteisen sisällön, kuten tekstin, äänen, kuvan, videon tai minkä tahansa muun sisällön tuottaminen on pysynyt hyvin pitkään ihmisen historiassa kovin samankaltaisena perusteiltaan ohittaen tekniikan lukuun ottamatta sitä, että videon tuottamisen historia ei ole yhtä pitkä kuin tekstin. Internet ja digitalisoituminen mahdollistavat tiedon, datan ja sisällön jakelun uudella tavalla. Jatkossakin täytyy olla jokin keino hallita tätä kasvavaa sisältömäärää jollain tavalla. Tässä suhteessa sisällönhallintajärjestelmät omaavat melko kirkkaan tulevaisuuden, vaikka niiden kuolemasta on puhuttu vuosikausia aika ajoin (Econtentmag, 2016).

Perinteisesti kotisivujen valmistus on vaatinut vähintään HTML-ohjelmakoodin tuntemusta, eli sivujen ylläpitoon ollut lähinnä ainoastaan yritysten IT-henkilöiden tehtävä (Library Hi Tech 2006, Seadle 2006 mukaan). Kun yrityksessä on käytössä sisällönhallintajärjestelmä, sivuston ylläpitäjän ei tarvitse niinkään käyttää aikaa ohjelmakoodin tuottamiseen tai tuntea sitä parhaassa tapauksessa lainkaan, vaan tämän on mahdollista saada sivusto päivitetynksi haluamallaan tavalla muutamalla klikkauksella (Accrinet, 2013).

Lisäksi perinteisesti kotisivujen ylläpidosta on vastannut minimissään yksi henkilö ja tämänkin henkilön vaatimukseen on sisältynyt tavallisesti HTML-kielen tuntemus. Koska sisällönhallintajärjestelmän käyttö mahdollistaa erilaisten roolien asettamisen usealle järjestelmän käyttäjälle, myös useampi ylläpitäjä on mahdollinen ilman, että kenenkään näistä tarvitsisi tuntea välttämättä teknistä toteutuspuolta lainkaan.

Mielestäni juuri kaksi yllämainittua asiaa ovat ehkä vahvimmat edut, joita yritys saa sisällönhallintajärjestelmän käytöstä, koska näin työntekijöiden aika säästyy varsinaisesti tärkeimpiin työtehtäviin.

## 2.2 WordPress

WordPress on palvelimella toimiva sisällönhallintajärjestelmä, joka on toteutettu ohjelmointikieli PHP:llä. (WordPress). Syyskuussa 2015 mitatun markkinaosuus- tutkimuksen mukaan WordPress omisti CMS-markkinoista vajaan 60 % lohkon (W3Techs, 2015).

WordPress valikoitui tähän opinnäytetyöhön opinnäytetyön tekijän omien positiivisten kokemusten pohjalta kyseisestä järjestelmästä. Lisäksi kävi yrityksen edustajan kanssa keskusteltaessa, että yritys olisi kiinnostunut siirtämään nykyiset kotisivunsa WordPress-alustalle.

WordPressin esiaste julkaistiin vuonna 2001, jolloin Michel Valdrighi julkaisi PHP:llä toteutetun ja MySQL-tietokantoja hyödyntävän julkaisujärjestelmän omaan käyttöönsä. Järjestelmän nimi oli ”b2 cafelog”. Tuotteen julkaisu tapahtui GPL-lisenssillä. Vuonna 2003 Matt Mullenweg ja Mike Little hakkeroi b2 -cafelogia, ja WordPress syntyi. Vuotta 2003 voidaan pitää virallisena WordPress-syntymävuotena (WordPress).

WordPress on sittemmin sen julkaisusta kasvanut pelkästä bloggaustyökalusta kaikenlaisia sisältöjä omaavien verkkosivujen alustaksi. Nykyisin WordPress-sivusto ei ole enää rajattu pelkkään blogiin (Silver 2013,8).

### 2.3 WordPressin asennus ja käyttöönotto

WordPress vaatii palvelimelta sekä PHP- että MySQL-tuen. Vaadittavat versiot kyseisistä teknologioista riippuu siitä, minkä version WordPressistä haluaa asentaa palvelimelle. Suosituksena voi pitää, että oikeastaan aina kannattaa käyttää turvallisuusrisien minimoimiseksi uusinta mahdollista versiota. WordPress-vaatimukset kunkin versionumeron kohdalta saa tarkistettua WordPressin virallisilta kotisivuilta. (WordPress).

Viralliset WordPress-kotisivut löytyvät osoitteesta <https://WordPress.org/>, ja sieltä saa myös ladattua viimeisimmän version WordPress-ohjelmistosta ilmaiseksi. Asennus tapahtuu joko pitkän kaavan mukaan, tai sitten on olemassa myös nk. ”Kuuluisa 5 minuutin asennus.” (Famous 5-minute install) (WordPress).

Tässä opinnäytetyössä ei käydä varsinaista asennusprosessia läpi, vaan tässä opinnäytetyössä keskitytään luomaan responsiivinen kotisivuteema tai osa teemasta WordPress-järjestelmään ja tutkimaan sen etuja ja haittoja liiketoiminnan kannalta.

### 2.4 WordPressin ympärille rakennettu liiketoiminta Suomessa

Vaikka WordPress onkin GPL-lisenssillä jaettava ilmaisohjelmisto, Suomessa ja muissa maissa EU-alueella WordPress on myös merkittävä osa liiketoimintaa. Yksistään

Suomessa on listattuna 41 WordPress-toimintaa harjoittavaa yritystä. ([Vierityspalkki, 2013](#))

Vierityspalkin lokakuussa 2013 tekemässä kartoituksessa WordPress-liiketoimintaa harjoittavat yritykset jaettiin kolmeen kategoriaan. Erikoistuneita tekijöitä, joilla WordPress on merkittävä osa liiketoimintaa, eikä niillä ole juuri muita toiminta-alueita, löytyi kartoituksessa kolme kappaletta. Turussa merkittävää WordPress-liiketoimintaa harjoittaa Aucor.

Selkeästi kartoituksen suurin yritysryhmä oli pienet ja notkeat haastajat. Kolmas kategoria Vierityspalkin tutkimuksessa oli turvalliset ja osaavat tekijät. Niitä löytyi kaikkiaan 9 kappaletta.

### 3 RESPONSIIVINEN WEB-SUUNNITTELU

Responsiivinen verkkosuunnittelu muuttaa ja on jo osittain muuttanut rakentamaamme www-maailmaa. Se on jo määritelty osittain, mutta määrittelee osittain myös jatkossa sen, mitä voimme nähdä verkossa ja miten.

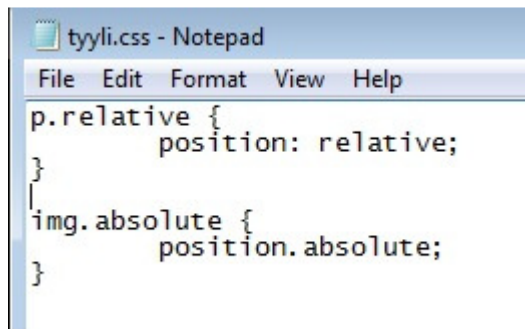
Viime vuosikymmenen aikana verkkosivujen kehitys on liikkunut hyvin rajatulla alueella, vaikka verkkoliikenne ja verkon käyttö yleensä on kehittynyt hyvinkin voimakkaasti ympäri maailman. Syynä tähän kehitykseen on se ajatus, että verkkoa selattaisiin ainoastaan työpöytä-tietokoneilla (Carver, 2015, 19).

Vaikka responsiivinen WEB-kehitys on noussut oikeastaan HTML5:n ja CSS3:n tekniikoiden tulon myötä päivänvaloon selkeämmin, itse asiassa www-historian ensimmäisenä responsiivisena sivustona voidaan pitää 2000-luvun alussa julkaistua Audi-autovalmistajan sivustoa, jonka kehitti yritys nimeltä Razorfish. Sivusto salli erikokoiset päätelaitteet tai oikeastaan erikokoiset selainikkunat ja sivuston sisältö mukautui päätelaitteen tai selainikkunan koon mukaan (Kalbach, 2012). Kyseessä oli siis verkkosivusto, mikä enteili jo tulevaa kehitystä.

Takana on lopulta vuonna 2016 ne ajat, kun verkkosivu tai verkkopalvelu suunniteltiin tietyn kokomuotin mukaan (fixed-width). Verkkoa selataan nykyään mitä erilaisimmilla laitteilla. Erityisesti äylaitteiden suosio on nostanut responsiivisen web-suunnittelun merkitystä huomattavasti. Takana ovat ne ajat, jolloin verkkoa saatettiin selata ainoastaan perinteisen tietokonemonitorin välityksellä. Nykyään ihmiset selaavat nettiä mitä moninaisimmilla äylaitteilla, kuten älypuhelimet, tabletit tai vaikkapa televisiolla. Erään tutkimuksen mukaan vuonna 2014 maailmanlaajuisesti verkon mobiilikäyttäjien lukumäärä miljoonissa mitattuna ylitti ensimmäistä kertaa historian aikana työpöytäkäyttäjien lukumäärän (Smart Insights, 2016).

Vuonna 2010 Ethan Marcoette käytti vertauskuvallisesti arkkitehtuuria esimerkkinä puhuessaan responsiivisesta webistä. Ennen hänen artikkeliaan on käytetty termiä ”responsiivinen arkkitehtuuri” (Responsive Architecture), mikä vastaa kysymykseen, kuinka erikokoiset tilat voivat vastata nykykansan olemiseen kauttaaltaan (Marcoette, 2010). Arkkitehtuurissa sulautettujen robottien ja jännitykseen liittyvien materiaalien pohjautuen arkkitehdit pitää sellaisten taide-installaatioiden ja seinän struktuurien rakennetta, jotka taipuvat, joustavat ja laajenevat, responsiivisen arkkitehtuurin perustana

(Marcoette,2010). Edellä kuvailtu Marcoetten viittaus arkkitehtuurien tapaan käsittää responsiivisuus voisi hyvin toimia sovellettuna myös responsiivisessa www-suunnittelussa . Tämä vertauskuva sopii hyvin siihen, mitä tulee elementtien aseteluun www-sivulle. Mahdollisuus vuodattaa elementtejä sivun reunalta toiselle tarjoaa mukavan ja siistin ulkoasun mikä on vastaanottavainen sivun eri elementeille. Toisaalta, kun suunnittelija tarvitsee tiukemman kontrollin elementeille, elementit voidaan asemoida käyttäen eri tekniikoita, kuten suhteellista asemointia (relative) tai absoluuttista asemointia (absolute) (Shayhowe). Relatiivisen ja absoluuttisen sijainnin elementille voi määrittellä CSS-tyylitiedostossa, kuten kuvassa 1 on menetelty.

A screenshot of a Notepad window titled 'tyyli.css - Notepad'. The window has a menu bar with 'File', 'Edit', 'Format', 'View', and 'Help'. The text area contains the following CSS code:

```
p.relative {  
    position: relative;  
}  
img.absolute {  
    position: absolute;  
}
```

Kuva 1. Kappaleen relatiivisen sijainnin ja kuvan absoluuttisen sijainnin määrittely CSS-tyylitiedostossa.

Responsiivisuus tarkoittaa www-suunnittelussa sitä, että verkkopalvelu tai yksittäinen verkkosivu toteutetaan alustariippumattomaksi ilman, että loppukäyttäjän tarvitsee ottaa päänsärkyä sen suhteen, että tämän tarvitsisi käyttää jotain tiettyä laitetta tai tiettyä selainta parhaan mahdollisen selauskokemuksen saamiseksi. Hyödyntäen responsiivisia etuja koodi voidaan toteuttaa välttämättä ilman tarvetta palvelin-pohjaisen tai backend-ratkaisuun.

Responsiivinen toteutus ei olisi mahdollista ilman kahta asiaa: CSS3-mediakysejlyä, sekä pysäytyspistettä (breakpoints). Juuri kaksi edellä mainittua ominaisuutta selainriippumattoman responsiivisuuden ja antavat käyttäjälle mahdollisuuden selata verkkosivua haluamansa kokoisella päätelaitteella (Carver, 2015, 20).

Suunnitteleamalla alusta lähtien kotisivuprojektin responsiivisesti ”mobile-first”-lähestymistavalla saavutetaan useita etuja. Ensinnäkin responsiivisuus takaa sen, että

mikäli markkinoille tulee jatkossa uuden kokoinen näyttöpääte, on yritys responsiivisella ratkaisulla jo valmistautunut tuleviinkin näyttöpääte-julkistuksiin, eikä kotisivua tarvitse toistaiseksi uusia (Carver, M 2015). Lisäksi sivustosta muodostuu nopeampi, mikäli sivuston latausajat optimoidaan kunkin näyttöpääteen koon mukaan sopiviksi. (Carver, 2015) Ja lopultahan kaikki haluavat nettisivun latautuvan mahdollisimman nopeasti. Ja kolmantena etuna voidaan pitää sitä, että mikäli responsiivinen toteutus ja erityisesti CSS-tyylitiedosto on sujuvasti suunniteltu, ei tarvitse niinkään ajatella sitä, mitä selainta käyttäjä käyttää, vaan saavutetaan poikkiselaimellisuus (cross-browser) (Carver, M 2015).

Vaikka yrityksellä voikin olla rinnakkain kaksikin verkkosivua, yksi ”perinteinen”, ja yksi mobiililaitteille suunnattu versio, voi olla kätevää ja rahaa säästävää ratkaisu toteuttaa yksi responsiivinen ratkaisu, mikä sitten toimii usealla laitteella (Frain, 2012, 7–10). Yksi näkyvimmistä seikoista responsiivisessa toteutuksessa onkin se, että käyttäjän ei tarvitse muistaa kuin yksi URL, jolla hän pääsee käsiksi haluamaansa sivustoon kaiken kokoisilla laitteilla (Carver, M 2015 20).

Responsiivisessa verkkosivu-ratkaisussa on yhä joitain haasteita, jotka kehittäjiä on hyvä pitää mielessä. Esimerkiksi kuvien saanti responsiiviseksi saattaa tuottaa hankaluuksia. Käytännön toteutus tällä hetkellä saada kuva responsiiviseksi on skaalata kuva sopimaan näyttöpääteeseen käyttämällä `max-width: 100 %` -koodia CSS-tiedostossa, tai sitten mahdollisesti piilottaa koko kuva, kun kuvaa ei tarvita (CSS-Tricks Forums 2014). Kuvassa 2 esitetään koodi, jolla kuvaa ei ladata.

```
@media only screen (max-width: 479px) {  
    background:none;  
}
```

Kuva 2. Esimerkki CSS-tyylimäärittelystä, jolla voidaan piilottaa kuva näytöltä tietyn kokoisella laitteella ilman kuvan lataamista.

Käyttäjä-ystävällisyyttä ajatellen ei ole lainkaan sama, miten navigoinnin on toteuttanut responsiivisessa verkkosivuratkaisussa. Pienemmillä näyttöpääteillä ja erityisesti päätelaitteilla, joissa ei ole näppäimistöä käytettävissä, tulee luoda mieluiten mahdollisimman käyttäjäystävällinen navigointityyli CSS-media-kyselyitä käyttäen yhdistettynä



CSS-koodiin (Hongkiat). Perusteeltaan HTML-tiedostossa navigointipalkki on joukko <li>-tageja, mutta varsinainen käyttäjäystävällisyyden takaava ”taika” tapahtuu CSS-tyylitiedostossa.

```
@media only screen (max-width: 320px) {  
    nav {  
        height: 40px;  
        width: 100%;  
        font-size: 11pt;  
        position: relative;  
        border-bottom: 2px solid #283744;  
    }  
    nav ul {  
        padding: 0;  
        margin: 0 auto;  
        width: 320px;  
        height: 480px;  
    }  
    nav li {  
        display: inline;  
        float: left;  
    }  
}
```

Kuva 3. Esimerkki navigointilistan toteuttamisesta CSS-tiedostossa.

Responsiivinen kotisivuratkaisu kotisivun laatijan näkökulmasta muovaa myös yrityskulttuuria. Responsiivisella webillä ja webkehityksellä on tekemistä myös hyväksynnän kanssa ja se muovaa työskentelytapoja yrityksissä. Tavallisesti tiimit, jotka suunnittelevat responsiivisia ratkaisuja toisille yrityksille tai vaikka yksityishenkilöille, eivät toimi nk. perinteisen ”vesiputous”-mallin mukaan, vaan toiminta on paremminkin ketterää (Carver, M, 2015). Tämä säästää niin aikaa, kuin rahaakin.

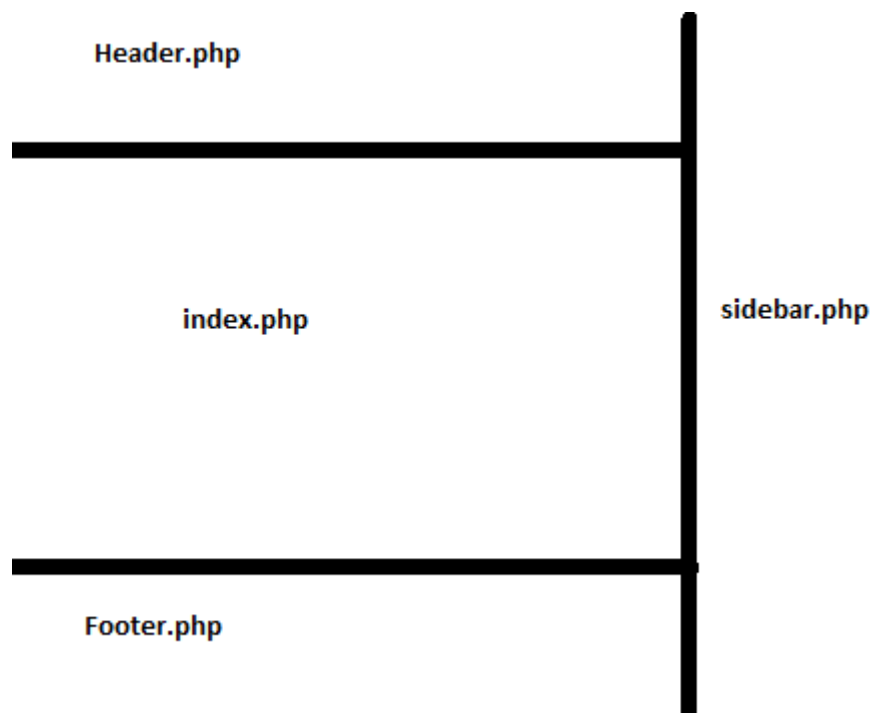
## 4 Responsiivisen Bootstrap WordPress-teeman toteuttaminen sekä käyttöönotto

Tässä luvussa perehdytään WordPress-tekniikkaan ja esitetään, miten yrityksen responsiivisten www-sivujen toteuttaminen on tehty teknisesti ja minkälaisia etuja responsiivinen ratkaisu tarjoaa yrityksen näkökulmasta. Lopputuloksena saadaan toimintakuntoinen tuote case-yrityksen käyttöön lähestyessä eteen tulevia ongelmia käytännön tasolla.

### 4.1 WordPress-tiedostorakenteesta

WordPress-teema sisältää laajan joukon tiedostoja, mikä takaa sen, että teeman layout on äärimmäisen hyvin kustomoitavissa (McCollin & Silver 2013, 72).

Kuvassa 4 on esitettynä runko WordPress-teeman perusrakenteesta, jossa nimettyinä välttämättömiä tiedostoja.



Kuva 4. WordPress-tiedostorakenne.

Lisäksi tässä opinnäytetyössä ohjelmoidaan kuvan 4 osoittamien tiedostojen lisäksi functions.php-tiedosto, mikä mahdollistaa esimerkiksi kustomoitavan otsikkokuvan saamisen käyttöön.

#### 4.2 1 – Tavallisen HTML5-dokumentin pohjakoodin luonti

Suunniteltaessa WordPress-teemaa on hyvä lähteä liikkeelle aivan tavallisesta HTML5-luurangon ohjelmoinnista. (Kuva 5)

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Vasaramäen tilitoimisto OY</title>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
7    <!-- Käytetään CDN -sisällön jakelu-verkosta-->
8    <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
9
10   <!-- jQuery -kirjasto -->
11   <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
12
13   <!-- Viimeisin käännetty Javascript -->
14   <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
15 </head>
16 <body>
17
18 </body>
19 </html>

```

Kuva 5. HTML-pohja.

Rivillä 6 (Kuva 5) saadaan responsiivisuus käyttöön laadittavalla verkkosivulla. Riveillä 8–14 otetaan Bootstrap-resurssit käyttöön CDN-palvelusta.(Content Delivery Network) Bootstrapin voi ladata myös paikallisesti koneelle, mutta käytettäessä verkon yli haettavaa resurssia Bootstrap on aina ajan tasalla, eikä päivittämisestä tarvitse itse huolehtia.

#### 4.3 Vaihe 2 - Twitter Bootstrap koodia HTML5-tiedostoon.

Bootstrap on alun perin Twitter-yhtiön vuonna 2010 kehittämä ja julkaisema front-end -kehys helpomman ja vaivattomamman responsiivisen www-suunnittelun edistämiseksi.

Bootstrap-pääkehittäjät olivat Twitter-työntekijät Mark Otto ja Jacob Thornton. Nykyään sen on ideana olla avoimen lähdekoodin projekti. Aluksi Bootstrap tunnettiin nimellä "Twitter Blueprint" ja se toimi aluksi yhtiön sisäisessä kehitystyössä, mutta alkoi sitten leviämään ulkopuolelle ja tekee sitä yhä (Bootstrap). Twitter Bootstrap julkaistiin ensi kertaa yleisölle Github-palvelussa avoimen lähdekoodin projektina perjantaina 19. elokuuta 2011, ja se on noussut maailman johtavaksi kehitys kehikseksi sittemmin (Bootstrap.)

Twitter Bootstrap hyödyntää HTML5 ja CSS3 tekniikkaa, ja se tarjoaa kehittäjille "Mobile -first"-lähestymistavan. Tämä "Mobile first"-lähestymistapa on oleellinen responsiivisen kehitystyön onnistumisessa (Bootstrap).

Twitter Bootstrap ei ole maailman ainut front-end -kehys. Sen kilpailijoita on mm. Foundation, Zimit, tai Ink muutamia mainittakoon (Modernweb, 2014).

Bootstrap vaatii nk. "containerin" (Container) sisällölle ja ylläpitämään taulukkojärjestelmää. Kyseinen kehys tarjoaa kaksi mahdollisuutta: Joko ".container", mikä on tarkoitettu yhden koon (fixed -width)-sivustoille, tai sitten ".container-fluid", mikä käyttää koko päätelaitteen leveyttä hyödyksi.

#### 4.4 GRID-taulukointijärjestelmä

Näyttöpäätteen ruutu voidaan kuvitella jaettavaksi riveiksi ja pylväiksi. www-sivuston sisältö sijoitetaan riveillä sijaitseviin pylväisiin (<http://getbootstrap.com/css/#grid>). Twitter Bootstrap sisältää variaation suositusta nk. 960 taulukointijärjestelmästä. (grid system) (Cochran, D, 2012). Ensin lienee syytä läpikäydä ensin vähän sitä, jotta ymmärrettäisiin paremmin Bootstrapin tarjoama variaatio.

Luku 960 viittaa pikseleihin ja nimenomaan leveyteen. Näyttöpäätteen rivi on jaettu 12 pylvääseen, joista kukin on 80 pikseliä leveä. Pylvääseen jää marginaalia molempiin suuntiin 10 pikseliä vasemmalle ja oikealle, joten sisällölle jää leveydeksi 60 pikseliä (960 Community). Toinen malli 960-järjestelmässä on 16-pylväikköinen esitys.

Vertailun vuoksi, Bootstrap tarjoaa yksinkertaisemman syntaksin taulukoinnin toteuttamiseen, eli kyseessä on ennalta määritettyjä luokkia (Cochran,D 2012). Responsiivinen ominaisuus tässä taulukoinnissa Bootstrapissa tulee esiin siinä, että ulkoasu on sää-

dettävissä eri kokoisille laitteille pöytätietokoneista tabletteihin yms. Bootstrap hyödyntää 12-pylväikköistä mallia.

Riippuen näyttöpäätteen koosta, pylväikön leveydeksi asetetaan automaattisesta leveydestä aina 62 pikselistä (pienet laitteet) 92 pikseliin (isot laitteet) marginaalin ollessa 30 pikseliä: Vasemmalla ja oikealla on marginaalia 15 pikselin verran. Tässä on siis Bootstrapin tarjoama ero perinteiseen 960-taulukointiasetteluun nähden (Bootstrap).

Ulkoasu luodaan GRID-taulukoinnilla seuraavan muistilistan mukaan:

- Mikäinen rivi määritellään hyödyntäen "container" (fixed-width) tai "container-fluid" (full-width)-luokkia.
- Rivejä käytetään pylväsryhmien luomiseen.
- Sisältö sijoitetaan pylvääseen, ja pylvään pitäisi olla rivin lapsisolmu.
- Kukin rivi sisältää tietyn lukumäärän erilevyisiä pylväitä, kunhan kokonaispylvässumma on 12. Ylimenevät pylväät siirtyvät seuraavalle riville.
- Ennalta määritellyt luokat pylväille on ".col-xs-", ".col-sm-", ".col-md-", ".col-lg-" ja niitä käytetään sen mukaan, minkä kokoisille laitteille sivustoa halutaan suunnitella. Pienemmälle laitteelle käytettävä luokkamäärittelmä toimii myös suuremmilla laitteilla, esimerkiksi keskikokoisille laitteille käytettävä col-md toimii myös suurissa näyttöpäätteissä.

HTML-div-tagin "class"-attribuutin arvona voi käyttää kerralla yhtä tai useampia pylväsluokka-määrittelyjä.

Esimerkki kahdesta pylvästä rivillä sijaitsevasta koodista sijaitsee kuvassa 6:

```

24 <div class="container-fluid">
25   <div class="row">
26     <div class="col-md-6">Vasen pylväs</div>
27     <div class="col-md-6">Oikea pylväs</div>
28   </div>
29 </div>

```

Kuva 6. Kaksi pylvästä Containerin sisään.

#### 4.5 Sivuston tekninen toteutus

Vaikka aiemmassa koodiesimerkissä kuvassa 5 näytetään, kuinka Bootstrap otetaan käyttöön hyödyntäen CDN-jakeluverkostoa (Content Delivery Network), tässä opinnäytetyössä lopulta päädyttiin asentamaan Bootstrap paikallisesti palvelimelle, mikä helpotti CSS-tyylitiedoston kirjoittamista. (CSS-tyylitiedoston ensimmäinen `@import` lause oli helpompi toteuttaa.)

On järkevää ajatella kutakin tekstiosiota ja mitä muuta tahansa elementtiä omana blogikirjoituksena, mikä sitten sijoitellaan sivulle. Lisäksi esimerkin vuoksi case-yrityksen sivulle luodaan yksi tai kaksi WordPress-sivua pohjaan malliksi, jotta jatkossa case-yrityksen sivuston ylläpitäjän olisi mahdollista ilman suurempaa päänvaivaa luoda lisää tarvittaessa templateen sivuja. Yritys säästää näin työntekijöiden aikaa.

Myös ratkaisu, että jokainen teksti olisi oma blogi-kirjoituksensa takaa helpon ylläpidon ja sivuston päivittämisen. Sivujen otsikot ja blogikirjoitusten esittely sivuilla tapahtuu dynaamisesti. Dynaaminen toteutus saavutetaan kirjoittamalla PHP-koodia WordPressin PHP-tiedostoihin.

Luotava mallipohja (template) on jaettu kahteen pylvääseen: ”oikea”, ja ”vasen”. Kaikki blogikirjoitukset, joissa käytetään tagia ”vasen” tai ”vasenpalsta”, näytetään ruudulla vasemmassa sarakkeessa, ja vastoin taas ”oikea” ja ”oikeapalsta” tagit näkyvät sivuston oikean puoleisessa sarakkeessa.

Kustomoitava otsikkokuva tekstillä toteutetaan ohjelmoimalla `functions.php`-tiedosto seuraavan kuvan 7 osoittamalla tavalla:

```

1  <?php
2  // Register Theme Features
3  $defaults = array(
4      'default-image'           => '',
5      'width'                   => 0,
6      'height'                  => 0,
7      'flex-height'             => false,
8      'flex-width'              => false,
9      'uploads'                 => true,
10     'random-default'          => false,
11     'header-text'             => true,
12     'default-text-color'      => '',
13     'wp-head-callback'        => '',
14     'admin-head-callback'     => '',
15     'admin-preview-callback' => '',
16 );
17 add_theme_support( 'custom-header', $defaults );
18 ?>

```

Kuva 7. Kustomoitava otsikkokuva käyttöön.

Lisäksi, jotta teeman otsikkokuva pystytään muokkaamaan yrityksessä, lisätään header.php-tiedostoon seuraava koodirivi:

```
height; ?>" width="<?php echo get_custom_header()->width; ?>" alt="Vasaramäen tilitoimisto Oy">
```

Navigointipalkkiin listataan dynaamisesti kaikki WordPress-sivut, jotka ylläpitäjä on luonut. Tämä tapahtuu lisäämällä seuraava Bootstrap-elementtejäkin omaava koodi header.php-tiedostoon:

```

25 <nav class="navbar navbar-default navbar-static-top" navbar-right">
26 <div class="container">
27 <ul id="menu" class="nav nav-tabs">
28
29 <li role="presentation" <?php if ( is_single() || is_home() ) { echo 'class="current_page_item"'; } ?>
30
31 <a href="<?php echo bloginfo('url'); ?>">Alkuun</a>
32
33 </li>
34 <?php wp_list_pages('echo=1&sort_column=menu_order&depth=1&title_li=6exclude=804'); ?>
35
36 </ul>
37 </div>
38 </nav>

```

Kuva 8. WordPress-sivujen dynaaminen listaus.

#### 4.6 Vaihe 4 – Tavallisen CSS3-tyylitiedoston luontia

CSS3-tyylitiedostossa määritellään käytettävän teeman ulkoasu. Kuvassa 9 on runko, jonka päälle opinnäytetyön CSS3-mallipohjaa lähdettiin rakentamaan.

```
1  /*
2  Theme Name: VasaraX
3  Author: Jere Sumell
4  Description: Turun AMK -Opinnäytetyö 2016
5
6  */
7
8  @import url('css/bootstrap.min.css');
9
10 /*Bootstrap3 -tuetut mediakyselyt*/
11 @media only screen and (max-width:767px) {
12
13 }
14 @media only screen and (min-width:768px) {
15
16 }
17 @media only screen and (min-width:992px) {
18
19 }
20 @media only screen and (min-width:1200px) {
21
22 }
```

Kuva 9. Mediakyselyt CSS3-tiedostossa.

Kommenttirivit 1–6 sisältää metatietoa teemasta. Nämä tiedot näkyvät WordPress-hallintapaneelissa teeman valinnan yhteydessä.

Rivillä 8 otetaan käyttöön Bootstrapin oma CSS-tiedosto, mikä sisältää määrittelyt käytettäville valmiille mediakyselyille.



#### 4.6.1 CSS3-mediakyselyistä

CSS-versionumero 3 on ensimmäinen CSS-versio, jossa uutena ominaisuutena on otettu käyttöön mediakyselyt. Mediakyselyn tunnistaa @media-alusta. Mediakysely on eräänlainen moduuli sallien sisällön renderoinnin vallitseviin olosuhteisiin ja laitteen resurssit kuten näyttökoko on huomioitu (Mozilla Foundation).

#### 4.7 Vaihe 5 – HTML-koodin sijoitus oikeisiin WordPressin php –tiedostoihin

Perusteiltaan voidaan ajatella jakoa kolmeen: Otsikkokoodi sijoitetaan header.php-tiedostoon, alapalkki-koodi footer.php-tiedostoon, ja sitten tärkein WordPress-tiedostoista on index.php, mikä sitoo eri osiot kokonaisuudeksi.

WordPress tarjoaa get\_header(), sekä get\_footer()-metodit, joilla voi sisällyttää otsikkokoodi-tiedoston dokumentin alkuun, ja vastaavasti alapalkin koodin get\_footer() -metodilla index.php:n loppuun.

#### 4.8 Vaihe 6–Varsinaisen sisällön ohjelmointi

Jotta saataisiin blogikirjoitukset näkymään oikein halutulla tavalla sivustolla, se vaatii PHP-koodia. Seuraavassa kuva vasemman sarakkeen PHP-koodista, mikä siis näyttää kaikki blogikirjoitukset, joissa on tunnisteena "vasen" tai "vasenpalsta":

```

1 <?php
2   get_header();?>
3
4 <div class="row">
5   <div class="col-md-6">
6     <?php
7       $vasen= new WP_Query( array( 'tag_slug__in' => array( 'vasenpalsta', 'vasen' ) ) );
8
9
10      while( $vasen->have_posts() ) :$vasen->the_post();?>
11
12        <h2><?php the_title();?></h2>
13        <div class="entrytext">
14          <?php the_content(); ?>
15        </div>
16        <?php endwhile; ?>
17      </div>
18

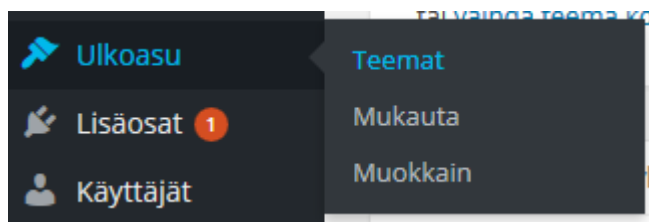
```

Kuva 10. Sisällön ohjelmointia.

Kuvassa 10 on esitettyä osa index.php -tiedostoa. Kuvassa riveillä 1–2 saadaan header.php-tiedostosta otsikkokoodi kutsumalla `get_header()`-metodia. Varsinainen ”taika” tapahtuu riveillä 6-16. Ensin luodaan uusi `WP_Query`-ilmentymä, mikä tallennetaan oliomuuttujaan `$vasen`. Rivillä 10 aloitetaan `while`-silmukka, mikä tulostaa kaikki saatavilla aiemmin määriteltyihin tageihin ja ehtoihin soveltuvat postaukset sivulle.

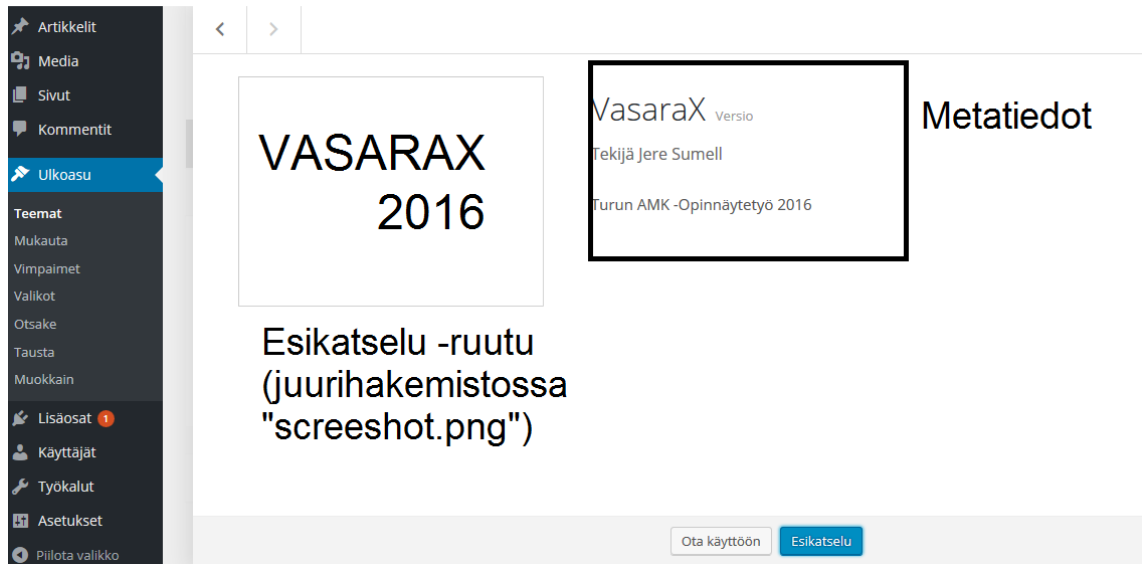
#### 4.9 Vaihe 7 – Uuden teeman käyttöönotto

Teemat sijaitsevat WordPress-alikansiossa ”wp-content/themes”. Edellä kirjoittamaani kansioon siirretään kansio, mikä sisältää teema-tiedostot. Seuraavaksi kirjaudutaan WordPress-hallintapaneeliin ja valitaan ”Ulkoasu -> Teemat”.



Kuva 11. WordPress-valikko.

Kuvassa 7 näkyy polku, jota pitkin sivuston ylläpitäjä voi käydä vaihtamassa haluamansa teeman.



Kuva 12. WordPress-teeman käyttöönottoikkuna.

## 5 KOETUT ONGELMAT

Käytettäessä kehitysympäristönä Windows-alustaa, johon oli asennettuna XAMPP, WordPress sekä Notepad++, kyseisistä ohjelmista ainoastaan WordPressin 5 minuutin asennus tuotti hieman pään vaivaa projektin alussa. Virheilmoitus ”Error establishing database” toistui, mikä johtui lopulta siitä, että tietokantaan ei ollut luotu käyttäjää, jolla on kaikki oikeudet kaikkeen. Itse case-yrityksen sivut sijoitetaan yrityksen verkkosäntöintipalvelulta ostamaan palvelintilaan, mikä tarjoaa WordPress-asennuksen ilmaiseksi palvelimelle, joten vastaavaa tietokantaan liittyvää ongelmaa ei varsinaisesti esiintynyt, kuin työn kehitysympäristössä.

Ensimmäinen versio malliteemasta esiteltiin yritykselle joulun alla 2005, mutta yritys piti layoutia turhan levottomana, ja runko piti ohjelmoida uusiksi alusta lähtien. Lopulta kun dynaaminen toteutus alkoi hahmottua, ja PHP-kielen tultua enimmäkseen tutuksi, suuremmilta ongelmilta vältyttiin. WordPress-sivulistausta ei saatu aluksi toimimaan halutulla tavalla. Kaikki sivujen linkin navigointipalkkiin saatiin listattua, mutta linkkejä klikkaamalla ei aluksi tapahtunut mitään. Edellä mainituista syistä työ viivästyi. Lisäksi PHP-kielen opiskelu otti aikansa.

Työstä muodostui lopulta melko insinöörimäinen, vaikka liiketalouden näkökulma onkin jonkin verran esillä. Toisaalta se heijastaa myös jonkin verran opiskellun opetussuunnitelman sisältöä, sillä opetussuunnitelmassa pääpaino oli nimenomaan teknisissä aiheissa, eikä niinkään varsinaisesti liiketaloudessa.

WordPress tarjoaa joukon funktioita (toimintoja) käytettäväksi mallipohjatiedoissa. Bootstrapissa on oma lähestymistapansa siihen, kuinka HTML on strukturoitu, mikä ei sovellu joidenkin kriitikoiden mielestä hyvin siihen, mitä WordPress tarjoaa oletuksena (Themeshaper 2014).

Lisäksi Bootstrap ei takaa välttämättä toivotunlaista lopputulosta sisällön sijoittelun suhteen, sillä 12-pylväinen Grid-järjestelmä edellyttää sisällön sijoittamisen ennalta määriteltyihin laatikkoihin, mikä ei tuo harmoniaa laadittavalle sivustolle välttämättä. Toisaalta opinnäytetyön käytännön osuuden edetessä Bootstrap alkoi tuntumaan melko luontealta tekniikalta toteuttaa uusi teema, joskin tässä työssä luotu teema on melko simppele.

Kolmantena ongelmana perehdyttäessä enemmän responsiivisen verkkosivusuunniteluun, responsiivisessa verkkosuunnittelussa täytyy ajatella erityisesti kuvien sijoittelua ja skaalautumista valmistetulle sivustolle. Vaikka kyseessä olisi kuinka suuri kuva tahansa, se latautuu palvelimelle kokonaisuudessaan ja silti sen pitäisi skaalautua pieniinkin näyttöpäätteisiin sopivan kokoisena oikeaan kohtaan. Kuvan lataamisen toki voi estää, mikäli haluaa säästää resursseja.

Ongelmana voidaan pitää myös sitä, että pääpaino lopulta oli enemmän ehkä PHP-ohjelmoinnin omaksumisessa WordPress-puolella, kuin itsessään ulkoasun luonnissa Bootstrapilla. Bootstrap hyödyntää Javascriptiä, eikä sisällä PHP -ohjelmakoodia, kun taas WordPress sisältää PHP -koodia, eikä niinkään Javascriptiä. Oman kokemukseni mukaan ohjelmistoteollisuudessa on melko harvinaista, että PHP ja Javascript yhdistettäisiin, tai ainkaan työntekijöiltä ei vaadita molempien hallintaa. Javascript on silti useiden alan ammattilaisten mielestä valtaamassa alaa.

## 6 YHTEENVETO

Tässä opinnäytetyössä tarkasteltiin sisällönhallintajärjestelmistä WordPressiä ja tutustuttiin Twitter Bootstrap-kehukseen. Lisäksi tutkielmassa perehdyttiin WordPressin ja Bootstrapin yhteensovittamiseen liittyviä teknisiä sekä liiketoiminnan kannalta saavutettuja etuja tai menetettyjä etuja.

Työssä perehdyttiin sisällönhallintajärjestelmiin yleisesti, niiden toimintaperiaatteisiin, sekä etuihin liiketoiminnan kannalta sekä kuvailtiin hieman sisällönhallintajärjestelmien tulevaisuuden näkymiä. Lisäksi teoria-osuudessa responsiivista verkkosuunnittelua sekä Twitter Bootstrap-kehystä, sillä käytännön osuudessa luotu WordPress-teeman osanen toteutettiin responsiiviseksi Bootstrap-kehystä hyödyntäen.

Työn edetessä tuli selväksi omiin havaintoihin perustuen, että mikäli yritys suunnittelee sivustonsa alusta pitäen responsiiviseksi, se säästää rahaa jatkossa, mikäli markkinoille tulisivatkin uuden kokoinen näyttöpäätte-laite ja lisäksi sivustosta saadaan käyttäjäystävällisempi (User Experience, UX) optimoidun latausajan tähden.

Responsiivisessa web-suunnittelussa on yhä joitain ratkaisemattomia ongelmia, kuten teoria-osuustarkastelussa esitetystä kuvassa 2 tuli selväksi. Tästä syystä kotisivujen tekstisisältö tulisivatkin olla kunnossa, jotta voitaisiin käyttää vähemmän kuvia, mikäli halutaan säästää resursseja. Toisaalta kuvan voi piilottaa pienemmillä näyttöpäätte-laitteilla lisäämällä CSS3-ohjelmakoodiin mediakyselyiden alle ohjelmakoodia siten, että määrittelee kuvan, jota ei halua näytettäväksi esimerkiksi tabletilla, taustakuvaksi.

Työn käytännön osuuden tavoite oli luoda toimeksiantajalle käyttövalmis tai liki käyttövalmis responsiivinen teema WordPressiin. Vasamären tilitoimisto ei ole toistaiseksi lokakuun 2016 alkuun ottanut työn pohjaa käyttöön, mutta se rohkaisi yritystä tutustumaan markkinoilla jo saatavilla oleviin responsiivisiin teemoihin. Lisäksi yritys innostui sisällönhallintajärjestelmän käyttöönotosta.

Lopputuloksena saatiin yleiskäsitys siitä, mitä on responsiivinen verkkosuunnittelu, ja mikä on sen tausta tähän päivään ja tulevaisuuteen, sekä yleiskäsitys responsiivisen suunnittelun mahdollistavista moderneista tekniikoista opiskelemalla aihetta Internetistä ja perehtymällä alan kirjallisuuteen. Lisäksi opinnäytetyössä käytännön osuudessa valmistettiin kohtalaisen yksinkertainen, mutta toimiva WordPress-teema pääosin PHP-kielillä ohjelmoiden ja hyödyntäen Bootstrap-teknologiaa case-yritykselle Vasaramäen

tilitoimisto Oy. Käytännön toteutus jäi melko raakileeksi, ja lopulta toimeksiantaja hyötyy ehkä teoriatasolla tästä opinnäytetyöstä lukemalla sen läpi.

Bootstrapin käyttöönotto edellytti HTML5:tä, sekä CSS3-ohjelmointia. Aihe oli mielenkiintoinen, sillä joidenkin arvioiden mukaan lopullinen HTML5-määrittely tehtäisiin vasta 2020-luvulla. (Korpela, 2011). Mielenkiintoista oli myös se, että opinnäytetyö-tutkielman tutkimusvaiheen mennessä syvemmälle paljastui, että todellisuudessa ensimmäinen responsiivista verkkosivua ja responsiivisen www-sivun nykypäivää ja tulevaisuutta enteilevä verkkosivusto on luotu paljon ennen, kuin HTML-standardi on kehittynyt viidenteen versionumeroon tai vaikkapa paljon ennen, kuin mediakyselyt olivat mahdollisia tyyllitiedostoissa. Lisäksi oma osaamiseni PHP-olio-ohjelmointikielestä karttui työn edetessä.

Opinnäytetyön alussa kirjoitettu sisällönhallintajärjestelmien eduista yrityksen liiketoiminnan kannalta ja johtopäätös sisällönhallintajärjestelmän voimakkaimmista eduista sai käytännön vahvistusta case-yrityksessä.

Lisäksi tämä opinnäytetyö herätti kiinnostuksen vapaalla lisenssillä jaettavien ohjelmistojen, sekä avoimen lähdekoodin ratkaisuihin ja niiden ympärille perustuvaan liiketoimintaan tai oikeastaan siihen, minkälaisia bisnesmahdollisuuksia niihin voidaan liittää ja tässä opinnäytetyössä myöskin selvitettiin, kuinka laajaa toiminta on Suomessa hiipaisten hieman muuta EU-aluetta. On olemassa yrityksiä, joiden liiketoiminta perustuu sisällönhallintajärjestelmien, ja juuri WordPress-järjestelmän ympärille.

## 7 LÄHTEET

Accrinet 2013: 5 Reasons You Should Be Using A Content Management System. Viitattu 9.7.2016 <https://www.accrinet.com/blog/5-reasons-you-should-be-using-a-content-management-system/>.

Bootstrap 2016a. About. Viitattu 9.7.2016 <http://getbootstrap.com/about/>.

Bootstrap 2016b. Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web. Viitattu 9.7.2016 <http://getbootstrap.com/>.

Bootstrap 2016c. Grid system. Viitattu 9.7.2016 <http://getbootstrap.com/css/#grid>.

Carver, M. 2015. Responsive WEB. Manning.

Cochran, D. 2012. Twitter Bootstrap web development how-to: a hands-on introduction to building websites with Twitter Bootstrap's powerful front-end development framework.

Comentum 2010. What is a Content Management System (CMS)?. Viitattu 9.7.2016 <http://www.comentum.com/what-is-cms-content-management-system.html>.

CSS-Tricks Forum, 2014. [Solved] Hiding body background image on smaller screens?. Viitattu 20.9.2016 <https://css-tricks.com/forums/topic/hiding-body-background-image-on-smaller-screens/>

Econtentmag 2016.: What's Next for Content Management. Viitattu 30.8.2016 <http://www.econtentmag.com/Articles/Editorial/Commentary/Whats-Next-for-Content-Management-112164.htm>

Frain, B. 2012. Responsive WEB Design with HTML5 and CSS Packt Publishing Ltd.

Hongkiat Blog: How to create responsive navigation. Viitattu 20.9.2016 <http://www.hongkiat.com/blog/responsive-web-nav/>

Kalbach, J. 2012. The First Responsive Design Website: Audi (circa 2002). Viitattu 29.8.2016 <https://experiencinginformation.WordPress.com/2012/07/22/the-first-responsive-design-website-audi-circa-2002/>



Korpela, J. 2011. HTML5:uudet ominaisuudet Docendo.

Marcotte, E. 2010. Responsive Web design. Viitattu 29.8.2016  
<http://alistapart.com/article/responsive-web-design>

McCollin, R. & Silver, BT. 2013. WordPress Theme Development Beginner's Guide Packt Publishing Ltd.

Mobile marketing engine: Google Will Punish You If You Don't Have A Mobile-Friendly Website. Viitattu 9.7.2016 <http://mobilemarketingengine.com/google-demotion-with-no-mobile-website/>.

Modern web 2014: 8 Bootstrap Alternatives. Viitattu 9.7.2016  
<http://modernweb.com/2014/02/17/8-bootstrap-alternatives/>.

Mozilla Foundation: @media. Viitattu 9.7.2016 <https://developer.mozilla.org/en-US/docs/Web/CSS/@media>

Open source CMS 2015: CMS Market Share. Viitattu 9.7.2016  
<http://www.opensourcecms.com/general/cms-marketshare.php>.

Seadle, M. 2006 mukaan. Content Management Systems, Library Hi Tech

Shayhowe, S. Detailed positioning, Lesson 2. Viitattu 20.9.2016  
<http://learn.shayhowe.com/advanced-html-css/detailed-css-positioning/>

Smith, N 2016. 960 Grid System 2016. Viitattu 9.7.2016 <http://960.gs>.

Silver, TB. 2013 WordPress 3.2 theme design : beginner's guide Packt Publishing Ltd.

Smart Insights 2016: Mobile Marketing Statistics compilation - Statistics on consumer mobile usage and adoption to inform your mobile marketing strategy mobile site design and app development. Viitattu 9.7.2016 <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>.

Stack Exchange: What are Web CMSs and how they work?. Viitattu 30.8.2016  
<http://programmers.stackexchange.com/questions/36380/what-are-web-cmss-and-how-they-work>

Techtarget: Definition - content management system (CMS). Viitattu 9.7.2016  
<http://searchsoa.techtarget.com/definition/content-management-system>.

Template Monster: How to choose the right CMS for your website - Open Source CMS Comparison. Viitattu 30.8.2016 <http://www.templatemonster.com/blog/how-to-choose-the-right-cms-for-my-website/>

The University of Kansas: Benefits of the CMS. Viitattu 9.7.2016 <http://cms.ku.edu/benefits-cms>

Themeshaper 2014r: Why Bootstrap is a bad fit for WordPress Themes. Viitattu 9.7.2016 <http://themeshaper.com/2014/08/19/why-bootstrap-is-a-bad-fit-for-WordPress-themes/>.

Vierityspalkki 2013: WordPress-toteuttajat Suomessa 2014. Viitattu 9.7.2016 <http://vierityspalkki.fi/2014/04/03/WordPress-toteuttajat-suomessa/>.

W3techs 2015. Market share trends for content management systems for websites. Viitattu 9.7.2016 [http://w3techs.com/technologies/history\\_overview/content\\_management](http://w3techs.com/technologies/history_overview/content_management).

Websitebuilderexpert 2016: How to Build a Website - It Doesn't Have To Be Hard. Viitattu 29.8.2016 <http://www.websitebuilderexpert.com/how-to-build-a-website/>

WordPress: About WordPress. Viitattu 9.7.2016 <https://WordPress.org/about/>.

WordPress: History. Viitattu 9.7..2016 <https://codex.WordPress.org/History>.