

Internetsivujen kehittämisen standardisoiminen mainostoimistossa

Lauri Hokkanen

Opinnäytetyö
Huhtikuu 2015

Mediatekniikan koulutusohjelma
Tekniikan ja liikenteen ala





Tekijä(t) HOKKANEN, Lauri	Julkaisun laji Opinnäytetyö	Päivämäärä 01.04.2015
	Sivumäärä 40	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty:
Työn nimi Internetsivujen kehittämisen standardisointi mainostoimistossa		
Koulutusohjelma Mediatekniikka		
Työn ohjaaja(t) NIEMI, Kari		
Toimeksiantaja(t) Suomen Pienyrittäjien Mainostoimisto Oy		
Tiivistelmä <p>Opinnäytetyön tarkoituksena oli tehdä Suomen Pienyrittäjien Mainostoimiston käyttöön perehdytyskansio tuleville harjoittelijoille ja uusille työntekijöille- sekä määrittelydokumentti alihankkijoille. Perehdytyskansio toimii pohjana tuleville harjoittelijoille ja työntekijöille. Perehdytyskansiossa kerrotaan yleisesti, miten internetsivut muodostuvat ja mitä lisäosia yleisimmin käytetään. Määrittelydokumentti toimii ohjeistuksena alihankkijoille, jotta heiltä tulleet työt olisivat SPYMin laatustandardien mukaisia.</p> <p>Perehdytyskansiossa on määritetty yleisesti, miten WordPress-sivujen peruselementit SPY-Millä tehdään. Peruselementtejä ovat esimerkiksi navigaatio, napit ja lomakkeet. Näiden lisäksi määritellään käytetyimmät lisäosat sekä WordPressin tietoturva.</p> <p>Alihankkijoiden määrittelylistassa määritellään, miten sivuston on toimittava ja mitä lisäosia sivustolla on käytettävä. Määrittelydokumentin liitteenä toimitetaan myös tarkistuslista asioista, joiden on oltava kunnossa, kun sivusto on valmis.</p> <p>Opinnäytetyön lopputuotteena tehdyt perehdytyskansio sekä määrittelydokumentti onnistuivat hyvin. Perehdytyskansion avulla uudet harjoittelijat ja työntekijät saavat hyvän käsityksen, mikä on SPYMin tapa toteuttaa internetsivuja. Määrittelydokumentin avulla alihankkijoiden on helpompi toteuttaa SPYMin standardien mukaisia internetsivuja.</p> <p>Perehdytyskansiota ja määrittelydokumenttia on tarkoitus laajentaa niin, että perehdytyskansio sisältää valmiiksi koodattuja elementtejä, jotka voisi helposti kopioida WordPress-teemaan. Määrittelydokumenttia kehitetään jatkuvasti yhdessä alihankkijoiden kanssa.</p>		
Avainsanat (asiasanat) Määrittelydokumentti, WordPress, HTML, PHP		
Muut tiedot		



Author(s) HOKKANEN, Lauri	Type of publication Bachelor's thesis	Date 01.04.2015
	Number of pages 46	Language of publication: Finnish
		Permission for web publication:
Title of publication Standardization of websites in advertising agency		
Degree programme Media Engineering		
Tutor(s) NIEMI, Kari		
Assigned by Suomen Pienyrittäjien Mainostoimisto Oyj Abstract <p>The purpose of the thesis was to create a guidebook for the trainees and new employees of Suomen Pienyrittäjien Mainostoimisto (SPYM) and a definition document to the agency's subcontractors. The guidebook contains basic information of how websites are formed and what plugins are commonly used. The definition document will act as a guideline for subcontractors to ensure that their work adheres to the quality standards of SPYM.</p> <p>The guidebook goes through how WordPress sites are made in SPYM in general. Basic elements are, for example, navigation, buttons and forms; in addition, the most common plugins and security of WordPress are discussed in the thesis.</p> <p>The definition document for subcontractors describes how a website must work and what plugins need to be installed. A checklist is attached to the definition document with a list of matters that need to be made when the website is ready.</p> <p>Final products, the guidebook and definition document, succeeded well. By using the guidebook and the definition document it will be easier to new trainees, employers and subcontractors to follow SPYM's practices to create websites.</p> <p>The guidebook and definition document are intended to expand, so that the guidebook contains pre-coded modules, which can be copied into WordPress theme. The definition document is constantly being developed with the subcontractors</p>		
Keywords/tags Guidebook, WordPress, HTML, PHP		
Miscellaneous		

Sisältö

1	Työn lähtökohdat	2
1.1	Toimeksiantaja	2
1.2	Tavoitteet ja työn rajaus	3
2	Web-sivustojen kehitys	3
2.1	Yleistä	3
2.2	Sisällönhallintajärjestelmät	5
2.3	WordPressin kehitys	6
2.4	Internetsivuston kehitys ja toteutus	9
2.4.1	Suunnittelu	9
2.4.2	Toteutus	10
3	Esiselvitys SPYMin nykytilasta	12
3.1	Miten internetsivut tehdään nyt	12
3.2	Kehitystyön ongelmakohdat ja parannusehdotukset	13
3.3	Käytettävät tekniikat	14
3.3.1	Responsiivisuus	14
3.3.2	WordPress-teema	16
4	Perehdytyskansio	18
4.1	HTML, CSS ja PHP	18
4.1.1	Yleisiä ohjeita	18
4.1.2	Navigaatio	18
4.1.3	Napit	19
4.1.4	Taulukot	21
4.1.5	Listat	24
4.2	WordPress	25
4.2.1	Lomakkeet	25
4.2.2	Lisäkentät	26
5	Hakukoneoptimointi ja tietoturva	31
5.1	Hakukoneoptimointi	31
5.1.1	Title-tagin eli otsikko-tagin	31
5.1.2	Metakuvaus	31
5.1.3	URL eli sivun osoite	32
5.1.4	Sisällön otsikot	32
5.1.5	Validi HTML-koodi	32
5.1.6	Sivukartta	32
5.2	Tietoturva	33
6	Pohdinta	34

Lähteet	36
---------------	----

Kuviot

Kuvio 1. Kuinka sisällönhallintajärjestelmät jakaantuvat eri alustoille.....	6
Kuvio 2. Esimerkinäkymä siitä, miten sisältö mukautuu media queryjen avulla.	15
Kuvio 3. Media query -muutospisteiden käyttö	16
Kuvio 4. Toggle-navigaation vaiheet.....	19
Kuvio 5. Esimerkinapit	21
Kuvio 6. Taulukko	22
Kuvio 7. Taulukko mobiilinäkymässä	23
Kuvio 8. Esimerkki Font Awesome -fontti-ikonista listamerkinä.	25
Kuvio 9. Advanced Custom Fieldsin Repeater Field.....	27
Kuvio 10. Advanced Custom Fieldsin Kuva-kenttä.	28
Kuvio 11. Advanced Custom Fieldsin Tiedot-kenttä.	28
Kuvio 12. Advanced Custom Fieldsin Location rules.	29
Kuvio 13. Advanced Custom Fieldsin lisäkentät.	29
Kuvio 14. Yhteystiedot-sivun näkymä.	30
Kuvio 15. Esimerkki hakutulossivusta.....	32

1 Työn lähtökohdat

1.1 Toimeksiantaja

Opinnäytetyön toimeksiantaja oli jyvaskyläläinen Suomen Pienyrittäjien Mainostointo (SPYM). SPYM perustettiin vuonna 2008, ja se työllistää tällä hetkellä 14 työn-

tekijää. SPYM tarjoaa markkinointiratkaisuja pienille ja keskisuurille yrityksille. Markkinointiratkaisuja ovat esimerkiksi www-sivut, sähköiset suoramarkkinointikampanjat ja erilaiset sosiaalisen median sovellukset. SPYM suunnittelee ja toteuttaa markkinointiviestintää myös perinteisempiin medioihin, kuten esimerkiksi sanomalehtiin sekä mainosflyereihin.

1.2 Tavoitteet ja työn rajaus

Tarkoituksena oli tehdä SPYMin käyttöön perehdytyskansio tuleville harjoittelijoille ja uusille työntekijöille sekä määrittelydokumentti alihankkijoille. Perehdytyskansiossa kerrotaan, miten SPYMillä tehdään internetsivut ja minkälaiset määritelmät sivuston on täytettävä, minkälaisista moduuleista sivusto koostuu ja mitä lisäosia käytetään tietyissä tilanteissa, ja miten määritellään tietoturvaan liittyvät kohdat. Perehdytyskansio toimii hyvänä apuvälineenä uusille harjoittelijoille, tuleville työntekijöille tai muille SPYMin nykyisille työntekijöille, jotka haluavat tutustua siihen, mistä osa-alueista lopullinen internetsivu muodostuu.

SPYMillä on laaja alihankkijaverkosto, jota käytetään silloin, kun SPYMin omat sisäiset resurssit eivät riitä. Määrittelydokumentin tarkoitus on yhtenäistää ja parantaa alihankkijoilta tilattua koodia. Määrittelydokumentin avulla alihankkijoille annetaan tarkat ohjeet siitä, miten heidän pitää sivustot kasata ja minkälaisia toiminnallisuuksia sivustosta pitää löytyä. Alihankkijoiden määrittelydokumentin liitteenä toimitetaan myös tarkistuslista asioista, joiden on oltava kunnossa sivuston ollessa valmis.

2 Web-sivustojen kehitys

2.1 Yleistä

Internet kehitettiin 1990-luvun alussa, ja aluksi internetsivut olivat HTML-tageista muodostettuja pelkkiä tekstidokumentteja. Internetsivujen tyylittelyyn tarkoitettu CSS (Cascading Style Sheet) esiteltiin vuonna 1995 ja CSS-tyyliä avulla kehittäjät

pystyivät muotoilemaan internetsivuja kuten esimerkiksi vaihtamaan sivun taustaväriä, ja muuttamaan tekstin kokoa ja tyyliä. CSS-tyylit eivät tähän aikaan vielä olleet kuitenkaan erityisen suosittuja, sillä selaimilla oli vaikeuksia tukea niitä. (Lumsden 2012.)

1990-luvun puolivälin jälkeen internetsivujen jäsentely oli suurimmaksi osaksi taulukkopohjaista. Taulukot kehitettiin aluksi taulukkomuotoisen tiedon esittämiseen, mutta pian suunnittelijat huomasivat, että taulukoiden avulla sisältö pystytään jakamaan riveihin ja sarakkeisiin. Tällä tavalla sisältö on helppolukuisempaa ja paremmin jäseneltyä. (Carlson n.d)

Vuonna 2000 Microsoft julkaisi uuden selaimen nimeltä Internet Explorer 5, joka tuki 99 prosenttia CSS1-tyyleistä. Myös muut selaimet osasivat jo tukea tyyli-tiedostoja. Internetsivujen ulkoasu muuttui taulukkomuotoisesta enemmän monimuotoiseksi. CSS-tyylien avulla internetsivujen sisältö saatiin erotettua tyyleistä, ja myös toisinpäin. Tämän ansiosta internetsivut olivat paljon helpompia ylläpitää ja sivustot olivat paljon joustavampia. Värien ja ikonien käyttö yleistyi ja muuttui monipuolisemmaksi, ja internetsivujen käytettävyys yleisesti parani. (Kelly 2013.)

Nykypäivänä internetiä käytetään monilla erikokoisilla laitteilla ja monilla erikokoisilla näyttöpäätteillä. Erilaisten mobiililaitteiden ja taulutietokoneiden eli tablettien käyttäminen on yleistynyt huomattavasti, ja näiden lisäksi on markkinoille tullut pelikonsoleita, televisiota ja älykelloja. Näillä kaikilla voi olla erikokoiset näyttökoot, joten on tärkeää, että internetsivut pystyvät mukautumaan näihin kokoihin nyt ja tulevaisuudessa. Responsiivinen suunnittelu tarkoittaa sitä, että sivuston ulkoasu muokautuu laitteen näyttökoon ja ominaisuuksien mukaan. Esimerkiksi tabletilla tarkasteltaessa sisältö näkyy kahdella palstalla vierekkäin, kun taas puhelimella sisältö tiipahaa yhteen sarakkeeseen allekkain.

Internetsivujen ylläpitäminen on myös muuttunut alkuajoista huomattavasti. Ennen sivustot kovakoodattiin eli kaikki sisällöt ja tyylit kirjoitettiin suoraan lähdekoodiin. Sivut kehitettiin omalla koneella ja valmis sivu ladattiin serverille. Muokkaaminen

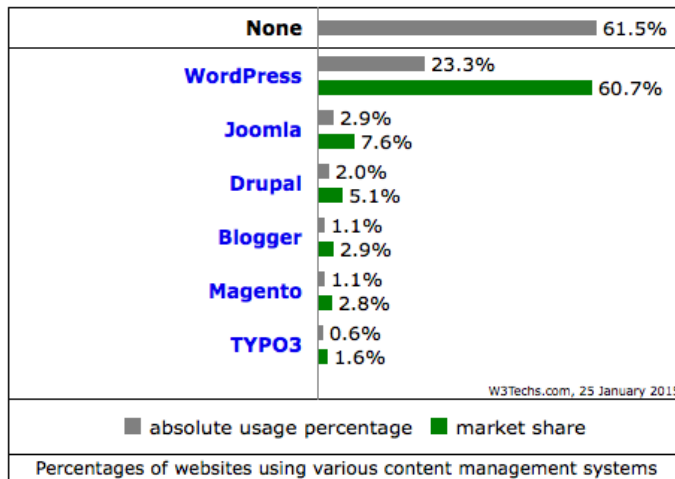
tapahtui samalla tavalla. Halutut sivut ladattiin koneelle, minkä jälkeen ne muokattiin ja ladattiin takaisin serverille. Tämä ei kuitenkaan ole erityisen käyttäjäystävällinen tai nopea ratkaisu, sillä jo pelkän sisällön muokkaaminen vaati hieman koodaus-taitoja tai koodin ymmärtämistä.

Tänä päivänä kovakoodattujen nettisivujen vaihtoehtona ovat erilaiset sisällönhallintajärjestelmät. Erilaiset sisällönhallintajärjestelmät ovat yleistyneet erittäin paljon. Yksi suurimmista syistä yleistymiseen on se, että monet sisällönhallintajärjestelmät ovat ilmaisia, sillä ne perustuvat avoimeen lähdekoodiin eli kaikki koodi ja kehitetyt toiminnallisuudet ovat kaikkien vapaasti käytettävissä. Muita syitä ovat, että sisällönhallintajärjestelmät ovat erittäin nopeita ja helppoja käyttää. Sisällönhallintajärjestelmiin löytyy useita erilaisia ilmaisia teemoja, jotka on helppo muotoilla omaan käyttöön sopiviksi ilman koodaustaitoja. Useimmista sisällönhallintajärjestelmistä löytyy myös ilmainen ja kattava dokumentaatio sekä hyvät keskustelualueet, joissa käydään läpi yleisimpiä asioita kyseisestä järjestelmästä. Sisällönhallintajärjestelmät sisältävät yleensä erilaisia käteviä ominaisuuksia kuten esimerkiksi artikkeleiden kategorisoinnin, kommentoinnin ja helposti käytettävät monipuoliset pluginit eli lisäosat, joiden avulla voidaan laajentaa sisällönhallintajärjestelmän toiminnallisuuksia.

2.2 Sisällönhallintajärjestelmät

Kolme suosituinta sisällönhallintajärjestelmää ovat WordPress, Drupal ja Joomla (Usage of content management systems for websites n.d.). Näistä kolmesta WordPress on ehdottomasti suosituin. Kuviosta 1 nähdään, että WordPressiä käyttää 60,7

% sisällönhallintajärjestelmää käyttävistä sivustoista, ja tämä on 23,3 % kaikista internetsivustoista.



Kuvio 1. Kuinka sisällönhallintajärjestelmät jakaantuvat eri alustoille (Usage of content management systems for websites n.d.)

Kaikki kolme suurinta sisällönhallintajärjestelmää on rakennettu käyttäen PHP:ta ja MySQL-tietokantaa. Ne ovat avoimeen lähdekoodiin perustuvia ja kaikilla on laaja käyttäjä- ja kehittäjäyhteisö.

WordPress aloitti vuonna 2003, ja se kehitettiin aluksi blogialustaksi. Wordpress kuitenkin kehittyi nopeasti ja on saanut paljon uusia ominaisuuksia. Yksi suurimmista syistä WordPressin nopeaan suosioon oli se, että WordPress heti aluksi keskittyi vain olemaan blogialusta, ja sillä oli tarkka kohdeyleisö eli blogikirjoittajat. WordPress on alusta asti ollut erittäin helppo asentaa ja ottaa käyttöön. (Mark 2011.)

2.3 WordPressin kehitys

Vuonna 2003 Matt Mullenweg ja Mike Little kehittivät b2/cafelog-blogisovelluksen päälle omanlaista, yksinkertaista ja tyylikästä julkaisualustaa yksityisille henkilöille. WordPressiä kehitettiin koko ajan eteenpäin ja siihen lisättiin eri ominaisuuksia. Vuonna 2004 julkaistiin versio 1.2 (Mingus), jonka suurin uusi ominaisuus olivat plugginit eli lisäosat. (History n.d.)

WordPress-teemat tulivat käyttöön vuonna 2005 versiossa 1.5 (Strayhorn), ja samana vuonna tuli vielä uusi versio 2.0 (Duke), jonka uusia ominaisuuksia olivat käyttäjäroolit. Samalla uudistettiin hallintapaneelia. (History n.d.)

Tammikuussa 2007 julkaistiin versio 2.1 (Ella). Sen mukana tuli paljon uusia ominaisuuksia kuten esimerkiksi automaattinen tallennus ja oikeinkirjoituksen tarkistaminen. Etusivuksi pystyttiin laittamaan mikä tahansa sivu, ja sivun tilan pystyi vaihtamaan luonnokseksi tai yksityiseksi, jolloin sivu ei näy ulospäin vierailijoille. Tämän version julkaisun jälkeen päätettiin, että WordPressiä päivitetään 3–4 kuukauden välein ja WordPressin käyttäjät pystyivät ehdottamaan ja äänestämään siitä, mitä uusia ominaisuuksia WordPressiin tulisi. Eniten ääniä saaneet ehdotukset sisällytettiin johonkin tulevaan WordPress-versioon. Syyskuussa 2007 julkaistiin versio 2.2 (Getz), jonka mukana tulivat tagien eli asiasanojen lisäys, hakukoneystävälliset URLit ja päivitysilmoitukset, eli Wordpress ilmoittaa, jos WordPressiin tai johonkin lisäosaan on tullut uudempi versio. (History n.d.)

Maaliskuussa 2008 julkaistiin versio 2.5 (Brecker). Tässä versiossa uudistuksia oli muun muassa hallintapaneelin uudistuminen. Nyt hallintapaneeli oli enemmän widgeteihin pohjautuva. Widgetien avulla pystyttiin seuraamaan artikkeleiden tietoja kuten esimerkiksi sitä, mihin artikkeliin tai sivuun on tullut kommentteja, tai sitä, mitkä ovat WordPressin suosituimpia lisäosia. Widgetien avulla saatiin myös haettua WordPressin ulkopuolelta uutisia. Brecker-päivityksen mukana tuli myös mahdollisuus ladata useita tiedostoja kerralla. Tämä uudistus helpotti ja nopeutti erittäin paljon esimerkiksi kuvagallerioiden tekemistä.

Myöhemmin vuonna 2008 ilmestyi versio 2.6., jossa uutena ominaisuutena oli muun muassa artikkeleiden versiohallinta. WordPress tallentaa jokaisen muokkauksen jälkeen uuden version artikkelista, ja käyttäjä pystyy myöhemmin tarkastelemaan näitä versioita. Muita uusia ominaisuuksia olivat esimerkiksi se, että kuvatestit voitiin näyttää kuvien yhteydessä artikkeleissa tai sivuilla ja se, että kuvagallerioiden sisältö voitiin lajitella. Loppuvuodesta 2008 julkaistiin versio 2.7 (Coltrane). Tämän version tärkeimpiä uusia ominaisuuksia olivat automaattinen päivitys ja mahdolli-

suus asentaa lisäosat helposti sisäänrakennetulla asennussovelluksella. (The History of WordPress 2014; History n.d.)

Versiot 2.8 (Baker) ja 2.9 (Carmen) julkaistiin vuonna 2009. Näiden uusia ominaisuuksia olivat sisäänrakennettu teeman asennussovellus, kuvien muokkausmahdollisuus, artikkelin tai sivun poistaminen roskakoriin ja sieltä palauttaminen. (The History of WordPress 2014; History n.d.)

Vuonna 2010 julkaistu versio 3.0 (Thelonious) oli suuri muutos WordPressille. Theloniuksen mukana tulivat mukautetut artikkelityypit, mukautetut valikot, mukautetut otsikkobannerit, mukautetut taustakuvat ja mahdollisuus hallita useita WordPress-sivuja kerralla eli MultiSite-ominaisuus. Mukautettujen artikkelityyppien avulla käyttäjä pystyy tekemään uusia artikkelityyppejä artikkeleiden ja sivujen lisäksi. Mukautetut valikot mahdollistavat sen, että käyttäjä pystyy yhdistämään valikkoon tai widgetiin artikkeleita, sivuja, kategorioita tai linkkejä. MultiSite-ominaisuuden avulla käyttäjä pystyy tekemään useiden sivujen verkoston yhden WordPress-asennuksen päälle. (The History of WordPress, 2014; History n.d.)

Helmikuussa 2011 julkaistiin versio 3.1 (Reinhardt), jonka mukana tuli muun muassa ylläpitäjän palkki. Ylläpitäjän palkki näkyy sivun yläreunassa, kun käyttäjä on kirjautuneena WordPressiin ja tarkastelee sivustoa julkiselta puolelta. Ylläpitäjän palkissa on linkkejä useisiin hallintapaneelin osioihin. Heinäkuussa 2011 julkaistu versio 3.2 (Gershwin) teki WordPressistä nopeamman ja kevyemmän. Tämän päivityksen myötä uudet minimivaatimukset olivat PHP 5.2.4 ja MySQL 5.0.15. (The History of WordPress, 2014; History n.d.)

Joulukuussa 2011 julkaistu päivitys teki WordPressistä helpomman uusille tulokkaille (The History of WordPress, 2014; History n.d.).

Kesäkuussa 2012 julkaistu versio 3.4 (Green) ja joulukuussa 2012 julkaistu 3.5 (Elvin) toivat WordPressiin teeman muokkaimen ja paransivat median hallintaa (The History of WordPress, 2014; History n.d.).

Vuonna 2013 tuli versio 3.6 (Peterson), johon sisältyivät äänen ja videon tuki sekä paranneltu automaattinen tallennus. Seuraava versio 3.7 (Basie) julkaistiin lokakuussa 2013, mikä paransi automaattista päivittymistä ja yleistä tietoturvaa. Versiossa 3.8 (Parker) tuli uutena ominaisuutena paranneltu ylläpitäjän palkki. (The History of WordPress 2014; History n.d.)

Versio 3.9 (Smith) julkaistiin huhtikuussa 2014. Tämä versio teki sisällön editoinnista sulavampaa ja helpompaa. Syyskuussa 2014 julkaistiin versio 4.0 (Benny). Tässä versiossa muutoksia olivat käyttäjän medialisäyksien parantaminen sekä lisäosien tarkastelun ja asentamisen muokkaaminen helpommaksi. Joulukuussa 2014 julkaistiin versio 4.1 (Dinah), joka paransi sisällönmuokkausta ja hallintapuolen käytettävyyttä. (The History of WordPress 2014; History n.d)

2.4 Internetsivuston kehitys ja toteutus

2.4.1 Suunnittelu

Sivuston rakentaminen WordPressin päälle alkaa samalla tavalla kuin minkä tahansa muun nettisivun kehittäminen. Ensin pitää määritellä ja suunnitella, minkälaista sivustoa ollaan tekemässä ja miksi. Sivun suunnittelun perustana voidaan pitää Tieken (Sähköisen kaupankäynnin aapinen 2003) neljää peruskysymystä:

1. **Miksi?** Mikä on sähköisen kaupankäynnin ja www-sivujen tarkoitus? Miten tämä tukee yrityksen koko toiminnan tarkoitusta eli toiminta-ajatusta?
2. **Kenelle?** Tavoitellaanko nykyisiä vai uusia asiakkaita? Ovatko asiakkaat kuluttajia vai yrityksiä?
3. **Mitä?** Mitä tuotteita ja palveluja tarjotaan ja mitä niistä kerrotaan?

4. **Miten?** Millaisia resursseja toteutus vaatii? Missä aikataulussa? Mitä uusia yhteistyömahdollisuuksia on? Ketkä ovat kilpailijoita?

Ennen sivujen toteuttamista kannattaa miettiä, miksi sivusto toteutetaan. Samalla kannattaa määritellä myös maalit, joihin tähdätään. Vain tarkkojen maalien avulla voidaan arvioida, onko sivusto onnistunut. Selkeitä maaleja ovat esimerkiksi myynnin nostaminen 20 %:lla tai tietty määrä tarjouspyyntöjä kuudessa kuukaudessa. Muuttamalla sivustoa ja tekemällä uudet maalit voidaan mitata, onko sivusto parantunut entisestä.

Suunnitelmassa pitää ottaa huomioon, mitä tuotteita tai palveluita nettisivulla markkinoidaan. Tuotteiden tai palveluiden määrittelyssä voidaan kysellä potentiaalisilta asiakkailta, minkälaisia palveluita tai tuotteita he olisivat valmiita ostamaan verkosta. (Sähköisen kaupankäynnin aapinen 2003, 10.)

Suunnitelmassa pitää olla myös määriteltynä, miten sivusto toteutetaan ja missä aikataulussa. Toteutukseen on hyvä määritellä työryhmä ja työryhmän sisällä vastuhenkilöt ja roolit. (Sähköisen kaupankäynnin aapinen 2003, 11.)

2.4.2 Toteutus

Työnjako

Työnjako voidaan toteuttaa monella tavalla. Periaatteessa yksi ihminen voi tehdä koko prosessin yksin tai prosessi voidaan jakaa monelle oman alan ammattilaiselle: sisällön tekee sisällöntuottaja, ulkoasun suunnittelee www-suunnitteluun erikoistunut graafikko ja teknisen toteutuksen tekee www-kehittäjä tai www-suunnittelija.

Sisältö

Sisällöntuottaja eli copywriter on ammattilainen tekstintuotannossa. Hänen tehtävänsä on tehdä kieliopillisesti oikeaa ja monimuotoista tekstiä. Sisällöntuottaja tekee

sisällön toimeksiannon ja kohderyhmän määrittämisen jälkeen hankittuun tietoon perustuen. (Raninen & Rautio 2003, 130.)

Graafinen suunnittelu

Sivuston ulkoasun tekee yleensä www-suunnitteluun erikoistunut graafikko. Graafikko hahmottelee kuvankäsittelyohjelmalla, miltä lopullinen sivusto näyttää ja miten eri osiot asetellaan sivustolle. (Cezzar n.d.)

Tekninen toteutus

Teknisen toteutuksen tekee yleensä www-suunnittelija tai www-kehittäjä. Teknisen toteutuksen tavoitteena on toteuttaa graafisen suunnittelijan tekemä ulkoasu toimivaksi internetsivuksi, jossa on aikaisemmin suunnitellut toiminnallisuudet. Sivuston toteutuksessa käytetään yleensä HTML-merkintäkieltä, CSS-tyylimäärytyksiä, JavaScript-kirjastoja ja PHP:ta

Kun tekninen toteutus tulee WordPressin päälle, se yleensä tarkoittaa sitä, että tehdään kokonaan uusi WordPress-teema. Teema on kokoelma tiedostoja, joista yhdessä muodostuvat koko sivuston ulkoasu ja toiminnallisuudet. (Theme Development n.d.)

3 Esiselvitys SPYMin nykytilasta

3.1 Miten internetsivut tehdään nyt

Tällä hetkellä SPYMin nettisivujen tekoprosessi etenee niin, että ensin asiakas ottaa yhteyttä myyjään ja he tekevät yhdessä suunnitelman, joka käytiin läpi luvussa 2.4.1. Tämän jälkeen myyjä laittaa asian eteenpäin projektipäällikölle, joka aikatauluttaa työn sisällöntuottajalle, graafikolle ja tekniselle toteuttajalle. Myyjä on myös pääasiallinen yhteyshenkilö.

Sisällöntuottaja voi suunnitella ja toteuttaa sisällön ensimmäiseksi. Sisältö tehdään yhteistyössä asiakkaan kanssa tai haastattelemalla asiakasta.

Graafinen suunnittelija toteuttaa aluksi ulkoasuvedoksen, joka toimitetaan asiakkaalle. Asiakas voi kertoa vedoksesta mielipiteensä ja tarvittaessa muutosehdotukset. Vedoskierrosten jälkeen, kun ulkoasu on hyväksytty, prosessi siirtyy tekniselle toteutukselle.

Tekninen toteutus voidaan toteuttaa sisäisesti tai ulkoistaa tarkkaan valituille alihankkijoille.

Tällä hetkellä SPYMillä sisäisesti tehtäviin töihin on käytettävissä yksi WordPress-teema, ja siitä on olemassa kaksi eri versiota: SPYM WP sekä SPYM WP Responsive. SPYM WP on staattinen WordPress-teema, ja SPYM WP Responsive on valmiiksi responsiivinen. Teema on tehty CSS Gridframeworkin päälle, ja teemat sisältävät WordPress Custom fields -funktiolla tehdyt lisäkentät, yleisimmin käytetyt lisäosat ja muutaman esimerkkisivun.

Ulkoistetuissa töissä SPYM lähettää alihankkijoille sopimuksen yhteydessä Tekniset vaatimukset internetsivuille -listan, jossa on lyhyesti määritelty, mitä vaatimuksia internetsivun on täytettävä. Tekniset vaatimukset -lista ei ole erityisen tarkka, ja vaatimukset jäävät erittäin yleiselle tasolle. Tämän vuoksi alihankkijoilta tullut lopullisen

tuotteen taso on ollut hieman vaihtelevaa, ja SPYMillä on pitänyt tehdä paljon korjauksia ja muutoksia sivuille.

3.2 Kehitystyön ongelmakohdat ja parannusehdotukset

Prosessin suunnittelu toimii hyvin. Myyjien ja asiakkaan pitämät suunnittelupalaverit ja workshopit tuottavat hyvän toimeksiannon, jonka mukaan tuotanto voi tehdä työnsä. Ongelmakohtia ovat koodatun lopputuloksen vaihteleva jälki ja alihankkijoilta tilattujen töiden puutteellisuus tai poikkeava tyyli.

SPYMillä pitäisi olla sisäisesti käytössä perehdytysdokumentti siitä, miten sivustot tehdään ja miksi juuri näin tehdään. SPYMin sisäisessä perehdytysdokumentissa kerrottaisiin, mikä sivuston elementtien tarkoitus on, miten ne tehdään ja miksi ne tehdään sillä tavalla. Dokumentissa koodi olisi kommentoituna, sisältäisi myös yleisiä ohjeita koodin rakentamiseen ja mahdollisesti linkkejä sivuille, esimerkiksi WordPress Codexiin, josta saisi halutessaan haettua lisätietoja.

Alihankkijoille pitäisi olla tarkemmat ohjeet siitä, miten tietyt elementit pitää toteuttaa, ja näitä ohjeita olisi pakko noudattaa. Alihankkijoille toimitettava määrittelydokumentti sisältäisi lyhyet, tarkat ohjeet siitä, miten elementit on tehtävä, mitä lisäosia käytettävä ja miten sivuston on toimittava. Tässä dokumentissa ei esiteltäisi elementtien toimivuutta eikä koodin tarvitse olla niin hyvin kommentoitu kuin perehdytyskansiossa, sillä alihankkijoina käytetään työnsä osaavia ammattilaisia.

Alihankkijoille toimitettaisiin myös tarkistuslista, jossa olisi lueteltuna kaikki vaatimukset, joita valmiilta sivulta odotetaan. Alihankkijan olisi palautettava tämä lista valmiin nettisivun liitteenä täytettynä. Täyttämällä tarkistuslistan alihankkija lupaa, että sivusto on siltä osin tarkistettu toimivaksi. Esimerkiksi kun alihankkija rastittaa kohdan "Sivusto on selainoptimoitu ja toimii yleisimmillä selaimilla (Mozilla Firefox, Google Chrome, Safari, IE 8+)", sivuston on toimittava näillä selaimilla, ja jos ei toimikaan, hän on velvollinen sen korjaamaan mahdollisimman pian. Tarkistuslistan

avulla saadaan alihankkijat tarkastamaan sivut tarkemmin ja nopeammin, ja samalla laatu pysyy parempana.

3.3 Käytettävät tekniikat

3.3.1 Responsiivisuus

Yleistä responsiivisuudesta

Responsiivisella web-suunnittelulla tarkoitetaan sitä, että suunniteltu sivusto on laitteesta tai näyttökoosta riippumaton. Tämä ei kuitenkaan tarkoita sitä, että sivusto näyttäisi kaikilla eri laitteilta samalta, esimerkiksi samanlaisella palstajaolla, vaan sitä, että sisältö ja käytettävyys säilyvät hyvinä. Responsiivisessa web-suunnittelussa panostetaan yleensä selkeään luettavuuteen ja hyvään käytettävyyteen. (LePage n.d; Knight 2009.)

Tekniikat

Responsiiviset sivut toteutetaan käyttäen media query -tekniikkaa. Media queryjen avulla tunnistetaan käyttäjän selaimen ikkunan leveys, ja tälle leveydelle voidaan määrittellä omat CSS-tyylit. Yleisimmät media queryt ovat max- ja min-width. Media queryyn määritetään leveydet ja annetaan tietyille ikkunan leveyksille muutospisteet. Muutospisteissä sisältö näytetään parhaalla mahdollisella tavalla käyttäen uusia CSS-tyylimäärittelyjä. Responsiivisessa suunnittelussa muutospisteitä kutsutaan breakpointeiksi (Bradley 2014).

Alla on esimerkki media queryjen merkinnästä. Esimerkissä muutospisteitä eli breakpointteja ovat 1024px, 768px, 480px ja 320px.

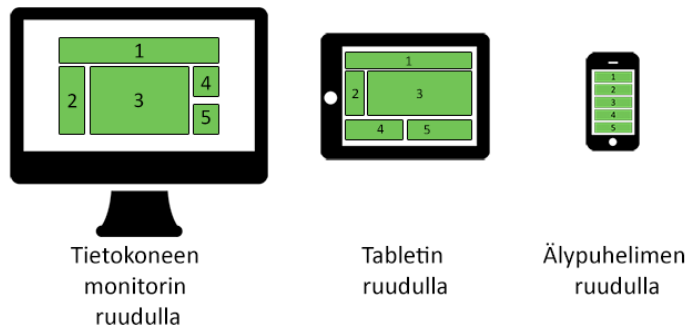
```
@media screen and (max-width: 1024px){  
/* Tylimäärittelyt, kun selaimen ikkuna on enintään 1024px leveä */  
}  
@media screen and (max-width:768px){  
/* Tylimäärittelyt, kun selaimen ikkuna on enintään 768px leveä */  
}
```

```

@media screen and (max-width: 480px){
/* Tyylimäärittelyt, kun selaimen ikkuna on enintään 480px leveä */
}
@media screen and (max-width: 320px){
/* Tyylimäärittelyt, kun selaimen ikkuna on enintään 320px leveä */
}

```

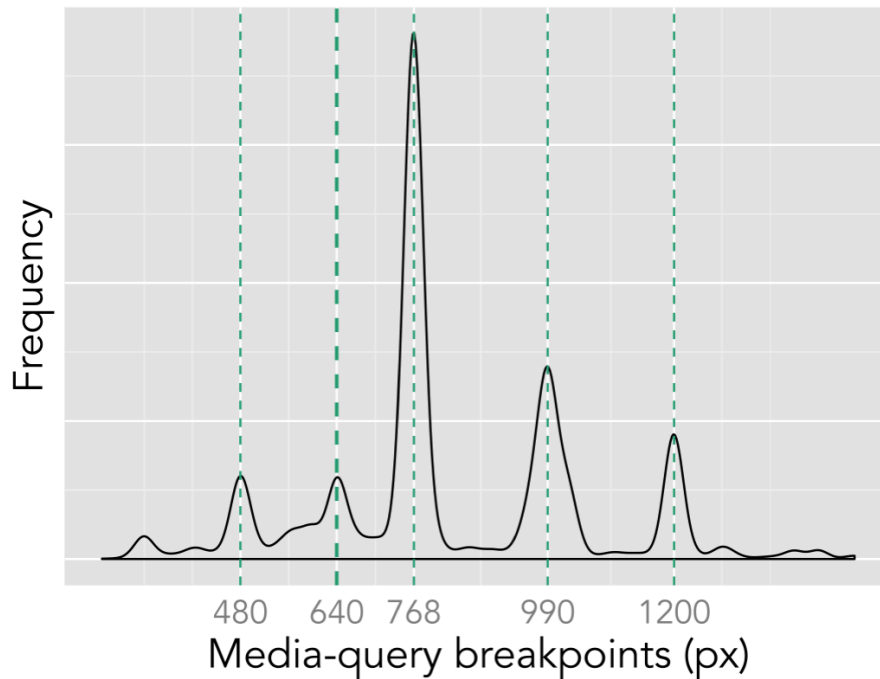
Alla olevassa kuviossa on esitetty, miten sisältö mukautuu media queryjen avulla.



Kuvio 2. Esimerkinäkymä siitä, miten sisältö mukautuu media queryjen avulla.

Quickleft teki tutkimuksen, jossa hän latsi ja vertaili 8000 sivuston CSS-tiedostoja. Tästä tutkimuksesta kävi ilmi, että käytetyimmät muutos pisteet olivat 480px, 640px, 768px, 990px ja 1200px.

Kuviosta 4 huomataan, että käyrä on aika leveä 990 pikselin kohdalla. Tämä johtuu siitä, että käyttäjät ovat käyttäneet paljon erilaisia määrittelyjä välillä 960px, 970px, 980px ja 990px, 992px, 1,000 ja 1,024px. Piikit käyrässä voidaan osittain selittää sillä, että kohdat 768px, 992px ja 1200px ovat valmiiksi määritellyjä muutos pisteitä Bootstrap frameworkissa. (McPherson n.d.)



Kuvio 3. Media query -muutospisteiden käyttö. (McPherson n.d)

Tarkkojen muutospisteiden käyttö voi johtua kehittäjien halusta kunnioittaa joidenkin tiettyjen puhelinvalmistajien näyttökokoja tai siitä, että useat kehittäjät käyttävät CSS frameworkien mukana tulevia valmiita muutospisteitä. (McPherson n.d.) Ennalta määriteltyjen muutospisteiden koko ei ole hyvä ratkaisu, sillä nykyään kehitetään koko ajan uusia laitteita, joiden näyttökoot voivat vaihdella erittäin paljon. Parempi tapa on antaa sisällön määrittellä, miten ulkoasu asettautuu. Käytännössä tämä tarkoittaa sitä, että toteutusvaiheessa pienennetään selaimen ikkunaa ja tehdään muutospisteet tarvittaville ikkunan leveyksille. (LePage n.d.)

3.3.2 WordPress-teema

WordPress-teema rakentuu teematiedostoista, jotka määrittelevät, miten internet-sivu näkyy ulospäin. Teematiedostot sisältävät HTML-koodia, sekä PHP:llä kirjoitettuja WordPressin omia komentoja eli Template tageja. Template tagien avulla haetaan tietoa MySQL-tietokannasta tai voidaan sisällyttää teeman tiedostoja sivupohjaan.

Haettu tieto sisältää hallintapaneelin kautta syötetyt sisällöt kuten tekstisisällön, otsikot, kuvat, videot, kategorianimet ja kirjoittajan tiedot.

Tiedostojen sisällyttäminen toimii yksinkertaisilla PHP-tageilla. Esimerkiksi tagi `<?php get_header(); ?>` sisällyttää tiedoston header.php ja tagi `<?php get_footer(); ?>` sisällyttää tiedoston footer.php (Theme Development n.d.)

Suurin osa teeman toiminnallisuuksista tulee Functions.php:n kautta. Functions.php toimii hieman samantyyllisesti kuin lisäosa, ja se ladataan automaattisesti jokaisella julkipuolen sivulla ja hallintapaneelissa. Functions.php:n kautta voidaan ladata ulkopuolisia kirjastoja, fontteja ja tyylitiedostoja sekä lisätä WordPressiin uusia ominaisuuksia.

Teeman yleisimmät tiedostot ovat seuraavat:

- Style.css – Päätyylitiedosto, sisältää teeman perustiedot.
- Header.php - Sivuston yläosa, ja sisältää myös HTML-elementin `<head>`, jossa haetaan esimerkiksi sivuston metadata.
- Index.php – Sivupohjatiedosto, yleensä etusivun näkymä. Jos muita sivupohjia ei löydy, tämä toimii kaikkien sivujen sivupohjana.
- Comments.php – Hakee ja näyttää artikkelien kommentit.
- Single.php – Yksittäisen artikkelin teematiedosto.
- Page.php – Yksittäisen sivun teematiedosto.
- Category.php – Näyttää kategorialistauksen
- Archive.php – Näyttää arkistolistauksen.
- 404.php – 404-virhesivu eli sivupohja, kun haluttua sivua ei löydy.
- Footer.php – Sivuston footerin eli alaosan pohja.
- Sidebar.php – Sivuston sivupalkki. Yleensä sivupalkissa on oletuksena sivuston vimpaimia.

(Theme Development n.d.)

4 Perehdytyskansio

4.1 HTML, CSS ja PHP

4.1.1 Yleisiä ohjeita

Varmista, että koodi on WordPressin koodausstandardien mukaista. Standardien mukainen koodi auttaa välttämään yleiset koodivirheet, helpottaa lukemista ja koodin muokkaamista. Hyvin kirjoitettu koodi on helppolukuista myös muille ja ongelmien etsiminen nopeutuu ja helpottuu.

Lisätietoja: <https://make.wordpress.org/core/handbook/coding-standards/>

4.1.2 Navigaatio

Navigaatio on sivuston tärkein elementti, sillä sen avulla käyttäjä pystyy liikkumaan sivustolla. Navigaation on oltava selkeä ja helppokäyttöinen jokaisella näyttökoolla.

Käyttäjälle on tärkeää, että hän tietää koko ajan, millä sivulla hän sillä hetkellä on. Näytä aktiivinen sivu navigaatiossa erilaisella taustavärillä tai tekstitehosteella. Jos käyttäjä on jonkin sivun alisivulla, on myös pääsivusta ilmettävä, millä sivulla hän on.

Jos sivusto on responsiivinen, käytä toggle-navigaatiota, jossa navigaatio on aluksi piilotettuna, mutta klikkaamalla ikonia valikko aukeaa. Kuviossa 3 on esitetty toggle-navigaation vaiheet.

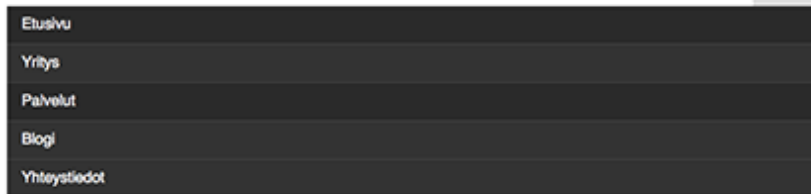
1. Navigaatio desktop-näkymässä

Etusivu Yritys Palvelut Blogi Yhteystiedot

2. Kiinni oleva toggle-navigaatio mobiilinäkymässä



2. Auki oleva toggle-navigaatio mobiilinäkymässä



Kuvio 4. Toggle-navigaation vaiheet

Jos navigaatioissa on käytössä alavalikko, linkin vieressä on oltava siitä merkki, esimerkiksi pieni alaspäin osoittava nuoli. Desktop-näkymässä alavalikko voi tulla esiin kursorin hover-efektillä, mutta koska hover-efektiä ei voi käyttää mobiilissa, on mobiilinäkymässä alavalikon tultava esiin klikkaamalla alavalikkoikonina.

4.1.3 Napit

Käytä nappien tekemiseen yleisiä luokkia. Tee ensin yksi luokka nimeltä btn ja määrittele tälle luokalle kaikki napille tarvittavat määrittelyt. Tämän jälkeen määrittele luokat pienemmälle (luokka btn-small), isommalle (luokka btn-large) ja täysleveälle napille (luokka btn-full). Jos sivulla käytetään paljon eri värisiä nappeja, tee jokaiselle värilliselle napille oma luokkansa, jossa määrittelet taustaväriin.

Alla on esiteltyä HTML-koodi nappien muodostamiseksi.

HTML

```
<!--Pieni nappi--!>
<a class="btn small" href="#" >Pieni nappi</a>
```

```

<!--Normaali nappi--!>
<a class="btn medium" href="#">Normaali nappi</a>

<!--Suuri nappi--!>
<a class="btn large" href="#">Suuri nappi</a>

<!--Täysleveä nappi--!>
<a class="btn full" href="#">Täysleveä nappi</a>

<!--Sininen normaali nappi--!>
<a class="btn medium blue" href="#">Sininen normaali nappi</a>

<!--Punainen suuri nappi--!>
<a class="btn large red" href="#">Punainen suuri nappi</a>

<!--Normaali nappi pelkillä reunoilla--!>
<a class="btn medium border" href="#">Normaali nappi pelkillä reunoilla</a>

```

Alla on esiteltyä CSS-tyylit, joilla muotoillaan HTML-koodi.

CSS

```

.btn{
    margin: 10px;
    position: relative;
    text-decoration: none;
    text-align: center;
    display: inline-block;
    padding: 0.88889em 1.77778em 0.94444em;
    font-size: 0.88889em;
    background-color: #1A3543;
    color: #FFF;
}

.small{
    padding: 0.77778em 1.55556em 0.83333em;
    font-size: 0.72222em;
}

.large{
    padding: 1em 2em 1.05556em;
    font-size: 1.1em;
}

.full{
    padding-right: 0px;
    padding-left: 0px;
    width: 100%;
}

.red{
    background: red;
}

```

```

}

.blue{
background: blue;
}

.border{
border: 1px solid black;
background:transparent;
color: #000;
}

```

Alla olevassa kuviossa on esiteltyä, miltä aiemmat koodimäärittelyt näyttävät internetsivulla.



Kuvio 5. Esimerkinapit

4.1.4 Taulukot

Taulukko on erittäin hyvä keino esittää vertailtavaa tietoa sivuilla. Taulukon luettavuutta kannattaa parantaa muotoilemalla taulukko CSS:ää käyttämällä. Taulukon helppolukuisuutta kannattaa parantaa varmistamalla, että tekstillä on tarpeeksi paddingia ympärillään. Isommissa taulukoissa kannattaa laittaa joka toiselle riville oma taustaväri käyttämällä CSS3 `nth-child(even)`- ja `nth-child(odd)`-valitsimia.

Alla on HTML-koodi taulukon muodostamiseksi:

```

<table>
  <tbody>

```

```

        <tr>
            <td>Lorem</td>
            <td>ipsum</td>
            <td>dolor</td>
        <td>sit</td>
    </tr>
    <tr>
        <td>sit</td>
        <td>amet</td>
        <td>consectetur</td>
    <td>adipiscing</td>
    </tr>
</tbody>
</table>

```

Alla ovat CSS-määrittelyt taulukon muotoilemiseksi:

```

table {
    width: 100%;
    border-collapse: collapse;
}
/* Parittomalle riville taustaväri #eee */
tr:nth-of-type(odd) {
    background: #eee;
}

td{
    padding: 6px;
    border: 1px solid #ccc;
    text-align: left;
}

```

Alla olevassa kuviossa esitettyinä, miltä taulukko näyttää internetsivulla.

Lorem	ipsum	dolor	sit
sit	amet	consectetur	adipiscing

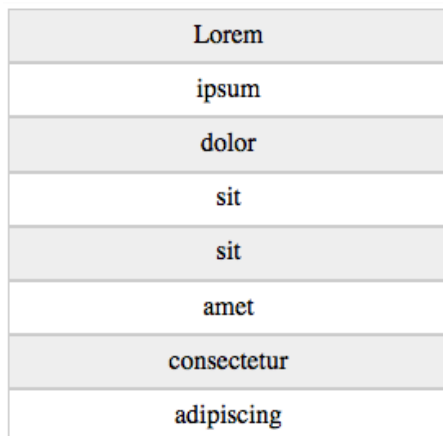
Kuvio 6. Taulukko

Näillä määrittelyillä taulukon leveys mukautuu näyttökoon mukaan, mutta monesti taulukko on silti liian iso pienemmille näytöille. Media queryjen avulla pystytään helposti muokkaamaan taulukot erimuotoisiksi.

Alla ovat esitettynä media query- ja CSS-tyylit, joiden avulla taulukon solut saadaan mobiilinäkymässä allekkain:

```
@media only screen and (max-width: 760px){  
  
  table, tbody, td, tr {  
    display: block;  
  }  
  
  td {  
    text-align:center;  
  }  
  
  tr:nth-of-type(odd) {  
    background: none;  
  }  
  
  td:nth-of-type(odd) {  
    background: #eee;  
  }  
  
}
```

Alla olevassa kuviossa on esitettynä, miltä taulukko näyttää mobiilinäkymässä.



Lorem
ipsum
dolor
sit
sit
amet
consectetur
adipiscing

Kuvio 7. Taulukko mobiilinäkymässä.

4.1.5 Listat

Käytä listojen tekemiseen tageja ` `, `` ja ``. Tagit ` ` tekevät järjestämättömän listan. Alla on esimerkkikoodi yksinkertaisesta listasta:

```
<ul>
  <li>Ensimmäinen</li>
  <li>Toinen</li>
  <li>Kolmas</li>
</ul>
```

Esimerkki näkyy sivuilla:

- Ensimmäinen
- Toinen
- Kolmas

Tagit `` tekevät listasta järjestetyn listan. Esimerkiksi lista:

```
<ol>
  <li>Ensimmäinen</li>
  <li>Toinen</li>
  <li>Kolmas</li>
</ol>
```

Esimerkki näkyy sivulla:

1. Ensimmäinen
2. Toinen
3. Kolmas

Jos haluat vaihtaa listamerkkiä, tee se CSS-tyyliin kautta. Jos haluat käyttää jotain omaa listamerkkiä, esimerkiksi Font Awesomen fontti-ikonia, voit laittaa sen CSS-tyyleihin näin:

```
ul {
  list-style:none;
}

li:before {
  content: "\f10d";
  font-family: FontAwesome;
```

```

color: #000;
font-size: 12px;
padding-right: 0.5em;
}

```

Esimerkki näkyy sivuilla:

```

“ Ensimmäinen
“ Toinen
“ Kolmas

```

Kuvio 8. Esimerkki Font Awesome -fontti-ikonista listamerkinä.

4.2 WordPress

4.2.1 Lomakkeet

Lomakkeet ovat erinomaisia työkaluja yhteystietojen keräämiseen, yhteydenottamiseen tai tuotteiden tilaamiseen. Lomakkeiden täyttämiskynnyksen täytyy olla tarpeeksi alhainen, että käyttäjä jaksaa ja viitsii täyttää lomakkeen. Täyttämiskynnystä voidaan alentaa kysymällä vain tarpeellisimmat asiat käyttäjältä, ja kertomalla mihin tarkoitukseen yhteystietoja käytetään ja mihin ei käytetä. Kukaan ei halua luovuttaa yhteystietojaan, jos on mahdollista, että ne luovutetaan myös kolmansille osapuolille.

Tee lomakkeet aina käyttämällä Contact Form 7 -lisäosaa (<http://contactform7.com/>) ja tee jokaiselle lomakkeelle oma uniikki kiitossivu. Käyttäjä ohjautuu kiitossivulle, kun lisää lisäasetukset kohtaan rivin:

```
on_sent_ok: "location.replace('www.esimerkki.fi/kiitossivu');"
```

Korvaa kohta www.esimerkki.fi/kiitossivu oikealla kiitossivun osoitteella.

Lisätietoa lisäosan käytöstä löydät osoitteesta <http://contactform7.com/faq/>.

4.2.2 Lisäkentät

Oletuksena WordPressin sivuilla ja artikkeleissa on rajallinen määrä kenttiä, joihin voi laittaa tietoa. Monesti kuitenkin tarvitaan useampia kenttiä erityyppisille tiedoille, ja tieto pitää hakea tiettyyn kohtaan sivulle. WordPressillä on oma custom post type -moduuli, mutta se on kankea ja yksinkertainen.

Jos sivustolla tarvitaan lisäkenttiä, käytä Advanced Custom Fields (ACF) -lisäosaa. Advanced Custom Fieldsin etuja WordPressin omiin lisäkenttiin on se, että Advanced Custom Fields on käytettävyydeltään paljon monipuolisempi. ACF:llä pystytään määrittelemään useita erilaisia lisäkenttätyppejä, ja niistä voidaan muodostaa kenttäryhmiä. Lisäkenttien näkyvyys voidaan määritellä sivujen, artikkeleiden, kategorioiden, sivupohjien tai käyttäjien mukaan. ACF antaa myös tehdä ehtoja siitä, miten lisäkentät näytetään.

Yksittäisen kentän tiedot voidaan näyttää koodilla:

```
<?php the_field('field_name'); ?>
```

Parempi vaihtoehto on kuitenkin käyttää ehtolauseita, että vain jos kenttä on täytetty, se tulostetaan sivulle. Ehtolauseeseen sisälle voidaan laittaa myös omia div-elementtejä. Esimerkiksi:

```
<?php
if(get_field('field_name')) { ?>
<div class="content"><?php echo get_field('field_name'); ?> </div>
<?php
}
?>
```

Yllä oleva esimerkkikoodi tulostaa divin "content" ja kentän "field_name" tiedot vain, jos kenttä on täytetty.

Jos sivulle tulee useita elementtejä vierekkäin – esimerkiksi yhteystiedot-sivu, jolla on henkilön kuva ja alla yhteystiedot – käytä ACF:n Repeater Fieldiä. Repeater Fieldillä voidaan tehdä alakenttärühmiä, joita voidaan toistaa peräkkäin.

Repeater Fieldin peruslooppiesimerkki:

```
<?php

// tarkistaa onko repeater fieldissä sisältöä
if( have_rows('repeater_field_name') ):

    // käy läpi sisällön
    while ( have_rows('repeater_field_name') ) : the_row();

// näyttää alakentän sisällön
the_sub_field('sub_field');

    endwhile;

else :

    // Jos sisältöä ei löytynyt

?>
```

Esimerkki yhteystietojen asettelusta ACF Repeater Fieldin avulla

Tee ensin uusi Field Group Advanced Custom Fieldsiin ja sen sisälle Repeater Field.

Add New Field Group

Henkilö

Order	Label	Name	Type
1	Henkilö	henkilo	Repeater

Field Label *
This is the name which will appear on the EDIT page

Field Name *
Single word, no spaces. Underscores and dashes allowed

Field Type *

Instructions
Instructions for authors. Shown when submitting data

Required?
 Yes No

Kuvio 9. Advanced Custom Fieldsin Repeater Field

Tämän jälkeen tee Repeater Fieldiin kaksi kenttää: kuva (ks. kuvio 10.) ja tiedot (ks. kuvio 11.). Valitse kuvan Return valueksi Image url. Tämä valinta palauttaa kuvan urlin ja sen voi tulostaa suoraan src-tagin sisään.

Order	Label	Name	Type
1	Kuva	kuva	Image

Field Label *
This is the name which will appear on the EDIT page
Kuva

Field Name *
Single word, no spaces. Underscores and dashes allowed
kuva

Field Type *
Image

Instructions
Instructions for authors. Shown when submitting data

Required?
 Yes No

Return Value
Specify the returned value on front end
 Image Array Image URL Image ID

Preview Size
Shown when entering data
Full Size

Mediakirjasto
Limit the media library choice
 All Uploaded to post

Conditional Logic
 Yes No

Wrapper Attributes
width % class id

Close Field

Kuvio 10. Advanced Custom Fieldsin Kuva-kenttä.

Order	Label	Name	Type
2	Tiedot	tiedot	Wysiwyg Editor

Field Label *
This is the name which will appear on the EDIT page
Tiedot

Field Name *
Single word, no spaces. Underscores and dashes allowed
tiedot

Field Type *
Wysiwyg Editor

Instructions
Instructions for authors. Shown when submitting data

Required?
 Yes No

Default Value
Appears when creating a new post

Tabs
Visual & Text

Työkälpalkki
Full

Show Media Upload Buttons?
 Yes No

Conditional Logic
 Yes No

Wrapper Attributes
width % class id

Close Field

Drag and drop to reorder

+ Add Field

Kuvio 11. Advanced Custom Fieldsin Tiedot-kenttä.

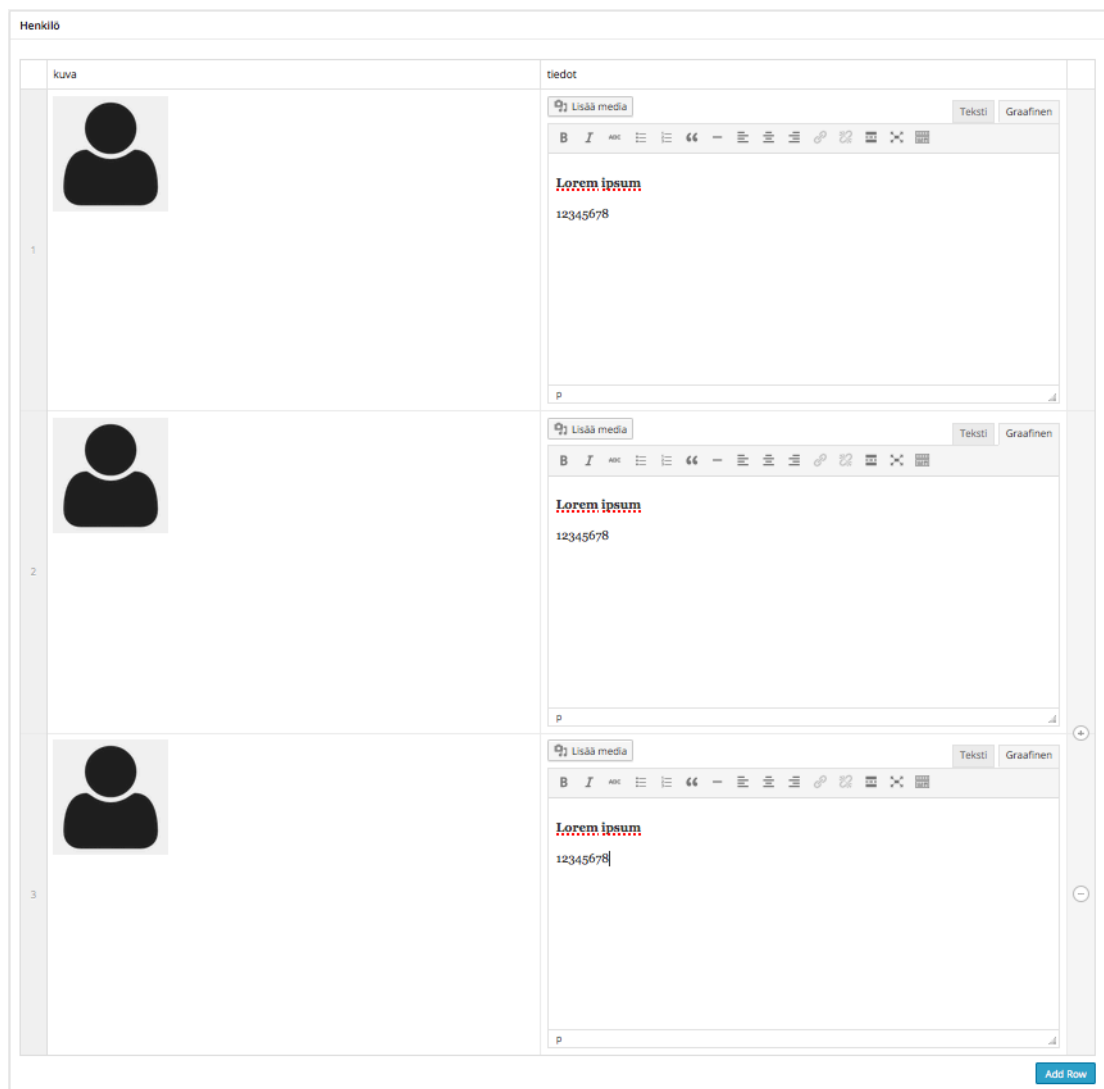
Field groupin alapuolella on kohta Location, jossa voit määrittää, millä sivuilla tai missä artikkeleissa lisäkentät näkyvät (ks. kuvio 12.).



The screenshot shows the 'Location' configuration panel for an Advanced Custom Fields group. It features a 'Rules' section with instructions to create rules for determining which edit screens use the fields. The main configuration area is titled 'Show this field group if' and contains a dropdown menu set to 'Post Type', followed by 'is equal to', another dropdown menu set to 'Artikkeli', and an 'and' button. Below this, there is an 'or' label and an 'Add rule group' button.

Kuvio 12. Advanced Custom Fieldsin Location rules.

Alla olevassa kuvioista nähdään, miltä lisäkentät näyttävät hallintapaneelin puolella.



The screenshot displays the 'Henkilö' (Person) field group configuration in the WordPress admin panel. It shows three rows of field groups, each with a 'kuva' (image) field and a 'tiedot' (details) field. The 'kuva' field contains a placeholder image of a person. The 'tiedot' field contains a rich text editor with the text 'Lorem ipsum' and '12345678'. The interface includes a 'Lisää media' (Add media) button, a 'Teksti' (Text) button, and a 'Graafinen' (Visual) button. The 'tiedot' field also has a 'P' (Paragraph) block selected. At the bottom right, there is an 'Add Row' button.

Kuvio 13. Advanced Custom Fieldsin lisäkentät.

Liitä alla oleva koodi haluttuun sivupohjaan.

```
<?php if( have_rows('henkilo') ): ?>

    <?php while ( have_rows('henkilo') ) : the_row(); ?>

    <div class="henkilo">
        
        <?php the_sub_field('tiedot'); ?>
    </div>

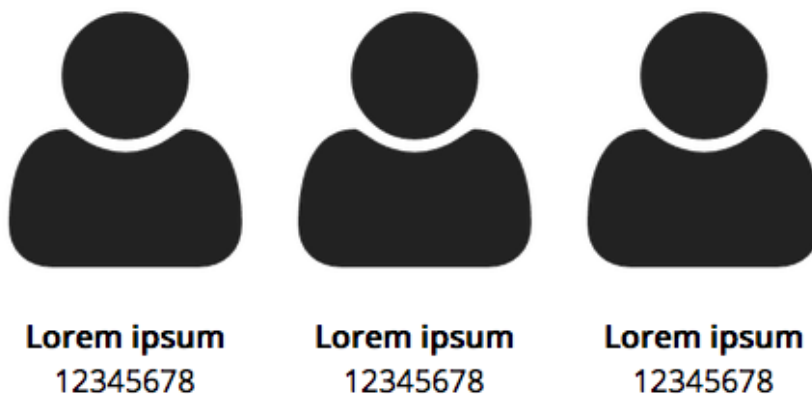
    <?php endwhile; ?>

<?php endif; ?>
```

Alla ovat CSS-tyylit, joiden avulla asetellaan tiedot vierekkäin.

```
.henkilo{
    max-width: 150px;
    float: left;
    text-align:center;
    font-family: 'Open Sans', sans-serif;
}
.henkilo img{
    max-width:150px;
    height:auto;
    display:block;
}
```

Alla olevassa kuviossa näkyy, miltä yhteystiedot näyttävät sivulla.



Kuvio 14. Yhteystiedot-sivun näkymä.

Lisätietoja:

<http://www.advancedcustomfields.com/>

5 Hakukoneoptimointi ja tietoturva

5.1 Hakukoneoptimointi

Hakukoneoptimoinnilla tarkoitetaan toimenpiteitä, jotka tehdään, jotta sivusto näkyisi paremmin hakukonetuloksissa. Hakukoneoptimointi on olennainen osa sivuston rakentamista ja se kannattaa tehdä huolella. Sivuston optimoinnissa tärkeimpiä asioita ovat title-tagin, meta description, URL eli sivun osoite, sisällön otsikot, validi html-koodi sekä Googleen syötetty sivukartta.

Hakukoneoptimoinnissa kannattaa hyödyntää lisäosaa WordPress SEO by Yoast. Tämän lisäosan avulla pystytään helposti määrittämään jokaiselle sivulle oma title, metakuvaus sekä avainsanat, joihin halutaan keskittyä. Lisäosan avulla voidaan myös piilottaa sivuja hakukonenäkymästä.

5.1.1 Title-tagin eli otsikko-tagin.

Title-tagin on sivuston tärkein yksittäinen elementti hakukoneoptimoinnin kannalta. Titlen on kuvailtava koko sivuston sisältö lyhyesti ja siihen kannattaa sijoittaa tärkeitä avainsanoja. Paras titlen pituus on noin 50–60 merkkiä. Kuviossa 5 on esimerkki hakutulossivusta ja siitä, miten title näkyy siinä.

5.1.2 Metakuvaus

Metakuvauksessa kuvaillaan sivuston sisältö lyhyesti ja se näkyy hakutulossivulla title-tagin alapuolella. Myös metakuvaukseen kannattaa sisällyttää avainsanoja ja niiden eri muotoja. Metakuvaus voi olla noin 160 merkkiä pitkä. Kuviossa 15 on esimerkki hakutulossivusta ja siitä, miten metakuvaus näkyy siinä.



Kuvio 15. Esimerkki hakutulossivusta

5.1.3 URL eli sivun osoite

Sivun URLin on oltava selkeälukuinen ja kuvaava. WordPress tekee automaattisesti hakukoneystävällisiä urleja ja ottaa urliksi sivuston otsikon, mutta sitä voidaan muokata helposti jälkeempään.

5.1.4 Sisällön otsikot

Sivustolla saa olla useampi pääotsikko eli h1-tageilla merkitty otsikko. Pääotsikoksi kannattaa valita hyvin koko sivun sisältöä kuvaava otsikko. Sisältö kannattaa muuten jakaa alaotsikoilla eli h2-, h3-, h4-, h5-, h6-tageilla. On suositeltavaa että sivustolla käytetään ainakin otsikkotasoja h1 ja h2.

5.1.5 Validi HTML-koodi

Sivuston koodin oikeellisuus kannattaa tarkistaa W3-validaattorilla. Hakukoneet suosivat internetsivuja, joiden koodi on validia, sillä tällaiset sivut näkyvät varmimmin kaikilla päätelaitteilla oikein.

5.1.6 Sivukartta

Sivukartan luominen helpottaa ja nopeuttaa sivuston läpikäymistä ja kategorisoimista hakukoneelle. Jos sinulla on käytössäsi lisäosa Yoast SEO, niin tämä lisäosa tekee automaattisesti XML-sivukartan osoitteeseen http://www.sivunosoite.fi/sitemap_index.xml/. Sivukartan voi tehdä myös osoitteessa <https://www.xml-sitemaps.com/>.

Kun sivukartta on tehty, se pitää syöttää Googlen Webmasters-tooliin, ja tähän tarkoitukseen tarvitaan Google-tunnukset.

5.2 Tietoturva

WordPress on tällä hetkellä maailman suosituin julkaisujärjestelmä. Suosion mukana on tullut haittapuolena se, että WordPress-sivustot ovat monesti erilaisten hyökkäyksien kohteena. WordPressin tietoturvaa voidaan kuitenkin parantaa muutamalla helpolla tavalla, jotka on listattu alla.

Päivitä Wordpress ja lisäosat

WordPressiin ja lisäosiin tulee uusia päivityksiä tasaisin väliajoin. Uudet päivitykset sisältävät uusien ominaisuuksien lisäksi monesti myös tietoturvaparannuksia. Ennen päivitystä kannattaa ottaa varmuuskopiot asennuksista ja tietokannasta.

Teemat ja lisäosat

Käytä vain turvalliseksi todettuja lisäosia ja teemoja. WordPressillä on oma lisäosa- ja teemahakemisto, joihin käyttäjät voivat lähettää lisäosia ja teemoja. Ennen kuin teema tai lisäosa julkaistaan, ne käyvät läpi tarkan tarkastusprosessin haavoittuvuuksien varalta.

Jos otat jonkin lisäosan pois käytöstä, poista kaikki sen tiedostot. Jos poistat lisäosan hallintapaneelin kautta, varmista, että kaikki tiedostot ovat lähteneet myös Plugins-kansiosta.

Suojaa Wp-config.php

Wp-config.php-tiedosto kannattaa suojata lisäämällä .htaccess -tiedostoon alla olevan koodin, jotta selaimella ei pääse tiedoston sisältöä katsomaan.

```
<files wp-config.php>  
  order allow,deny deny from all  
</files>
```

6 Pohdinta

Opinnäytetyön tarkoituksena oli tehdä Suomen Pienyrittäjien Mainostoimistolle eli SPYMillä perehdytyskansio ja määrittelydokumentti alihankkijoille. Alihankkijoiden määrittelydokumenttiin kuuluu myös tarkistuslista, joka pitää palauttaa tilatun työn ohessa. Kun tilattu työ on valmis, alihankkija tarkistaa työn tarkistuslistan avulla ja merkitsee valintaruutuun, että kyseinen osa sivustolla toimii halutulla tavalla.

Perehdytyskansion avulla uusille harjoittelijoille ja työntekijöille voitaisiin helposti kertoa, miten SPYMillä tehdään internetsivut, ja tällä tavalla heidän perehdyttämistään SPYMin työtapoihin voitaisiin nopeuttaa. Määrittelydokumentin tarkoitus oli parantaa yhteistyötä alihankkijoiden kanssa. Tarkat ohjeet ja vaatimukset selkeyttävät ja nopeuttavat SPYMin ja alihankkijoiden välistä kommunikaatiota. Sen ansiosta alihankkijoilla on yleiset vaatimukset selkeästi kirjallisena, jolloin he tietävät, mitä lopputulokselta vaaditaan. Perehdytyskansion ja määrittelydokumentin yhteinen tarkoitus oli yhtenäistää ja standardisoida internetsivujen tekemistä SPYMillä. Yhteiset menettelytavat parantavat SPYMin internetsivujen laatua ja nopeuttavat projektien läpivientiä.

Mielestäni onnistuin opinnäytetyössä hyvin. Perehdytyskansio on vielä hieman suppea, mutta tarkoitus on laajentaa sitä paljon kattavammaksi. Määrittelydokumentti on hyvä ja kattava. Määrittelydokumentin ohjeistuksen mukaan tehdyt internetsivut, ovat helposti päivitettäviä sekä teknisesti että sisällöllisesti. Kun alihankkija seuraa määrittelydokumentin ohjeita, loppukäyttäjän on helppo päivittää sivuston sisältöä WordPressin hallintapaneelin kautta.

Opinnäytetyössäni olisin halunnut tutkia ja kehittää enemmän mainostoimiston työprosessin läpivientiä ja etsiä myös muita kehityskohteita. Vaikka prosessin läpivienti tällä hetkellä toimii hyvin, olen varma siitä, että sitäkin voitaisiin tehostaa ja parantaa standardisoimalla viestintään ja dokumentointiin liittyviä käytänteitä.

Jatkokehityksen tavoitteena on, että perehdytyskansio tulisi sisältämään yleisten ohjeiden lisäksi myös valmiita elementtejä, jotka voisi helposti kopioida WordPress-teemaan ja ottaa siinä käyttöön. Tällä tavoin voitaisiin nopeuttaa etenkin pienempien projektien läpivientiä. Määrittelydokumenttia tullaan kehittämään yhdessä alihankkijoiden kanssa, ja tarkoitus olisi, että alihankkijoilta tilatut työt olisivat heti tullessaan julkaisuvalmiita. Määrittelydokumenttia ja tarkistuslistaa täydennetään aina, kun alihankkijoilta tulleista töistä löytyy jotain parannettavaa.

Lähteet

Bradley, S. 2012. Where And How To Set Breakpoints In Media Queries. Vanseodesignin internetsivut. Viitattu 30.1.2015.

<http://www.vanseodesign.com/web-design/media-query-breakpoints/>

Carlson, R. N.d. The evolution of web design: 1990 – present. Viitattu 17.1.2015.

<http://www.designjuices.co.uk/2011/09/web-design-evolution/>

Cezzar, J. N.d. What is graphic design? Viitattu 17.1.2015.

<https://www.sokanu.com/careers/graphic-designer/>

History. N.d. WordPressin internetsivut. Viitattu 20.1.2015.

<http://codex.wordpress.org/History>

Kelly, M. 2013. A Look Back at 20+ Years of Website Design. Viitattu 17.1.2015.

<http://blog.hubspot.com/marketing/look-back-20-years-website-design>

LePage, P. N.d. Responsive Web Design Basics. Viitattu 31.1.2015.

<https://developers.google.com/web/fundamentals/layouts/rwd-fundamentals/>

Lumsden, A. 2012. Brief History of the World Wide Web. Viitattu 17.1.2015.

<http://webdesign.tutsplus.com/articles/a-brief-history-of-the-world-wide-web--webdesign-8710>

Mark, J. 2011. How WordPress Took The CMS Crown From Drupal And Joomla. Viitattu 19.1.2015. <http://www.smashingmagazine.com/2011/11/29/wordpress-cms-crown-drupal-joomla/>

McPherson, A. N.d. The 2014 CSS Report: Examining how CSS is being used in the wild. Viitattu 3.2.2015. <http://reports.quickleft.com/css>

Raninen, T. & Rautio, J. 2003. Mainonnan ABC. Porvoo: WS Bookwell Oy.

Sähköisen kaupankäynnin aapinen. 2003. Tietoyhteiskunnan kehittämiskeskus ry (Tieke). Viitattu 29.1.2015.

<http://www.tieke.fi/download/attachments/3441521/Sahkoisenkaupankaynninaapinenpaivitetty03.pdf?version=1&modificationDate=1325505628000&api=v2>

The History of WordPress. 2014. WordPressin internetsivut. Viitattu 21.1.2015.

<http://www.wpbeginner.com/news/the-history-of-wordpress/>

Theme Development. N.d. WordPressin internetsivut. Viitattu 30.1.2015.

http://codex.wordpress.org/Theme_Development

Usage of content management systems for websites. W3Techsin internetsivut. Viitattu 18.1.2015.

http://w3techs.com/technologies/overview/content_management/all