

Websivujen julkaisu WordPress-julkaisujärjestelmällä
CASE: McCann Worldgroup Helsinki Oy

Juha Nordlund

Opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

2010



Tietojenkäsittelyn koulutusohjelma

<p>Tekijät Juha Nordlund</p>	<p>Ryhmä tai aloitusvuosi 2007</p>
<p>Opinnäytetyön nimi Websivujen julkaisu WordPress-julkaisujärjestelmällä CASE: McCann Worldgroup Helsinki Oy</p>	<p>Sivu- ja liitesivumäärä 36 + 11</p>
<p>Ohjaaja tai ohjaajat Sirpa Marttila</p>	
<p>Tämä opinnäytetyö on tutkimus WordPress-julkaisujärjestelmän käytöstä websivujen julkaisussa. Case-esimerkkinä tässä opinnäytetyössä toteutettiin ja kuvattiin mainostoimisto McCann Worldgroup Helsinki Oy:n websivut WordPressillä.</p> <p>WordPress on alun perin blogien tekemiseen kehitetty sovellus, joka perustuu vapaaseen lähdekoodiin, eli se on ilmainen ja sen edelleen kehittäminen on mahdollista. WordPressiä on alettu käyttää myös websivujen julkaisussa, koska sen sisältämät sisällönhallintaan liittyvät toiminnot ovat hyvin rakennettuja. WordPress vaatii toimiakseen tietokannan ja se asennetaan palvelinympäristöön.</p> <p>WordPress:sa sivusto rakennetaan php-elementeistä. Sivuston ulkoasu ja rakenne määritellään teemassa, johon voi lisäksi liittää toiminnallisuuksia. Teemassa on keskeistä template:n käyttö. Template on php-tiedosto, joka on osa sivuston muodostamaa kokonaisuutta.</p> <p>Opinnäytetyön käytännön osassa rakennettiin olemassa olevat HTML-pohjaiset McCann Worldgroup Helsinki Oy:n websivut WordPressiin php-elementeistä. Tehtiin McCann:lle oma teema ja toiminnallisuus, jolla työntekijät pystyvät päivittämään itse omia henkilötietojaan sekä lisäämään uusia henkilöitä, joita näytetään sivustolla olevassa swf-esityksessä.</p> <p>WordPress-sovellusta kehitettiin lisäämällä painikkeita hallintapaneeliin. Painikkeiden alle tehtiin toiminnallisuudet, jotka hakevat työntekijöiden tiedot xml-tiedostosta lomakkeeseen sekä tallentavat tehdyt muutokset xml-tiedostoihin.</p> <p>Sisällön rakentamiseen käytettiin loop-toimintoa, jonka avulla saatiin WordPressin sisällönhallintatoiminnot käyttöön.</p>	
<p>Asiasanat WordPress, php, sisällönhallintajärjestelmä</p>	

Information technology

<p>Authors Juha Nordlund</p>	<p>Group or year of entry 2007</p>
<p>The title of thesis Publishing websites with WordPress CASE: McCann Worldgroup Helsinki Oy</p>	<p>Number of pages and appendices 36 + 11</p>
<p>Supervisors Sirpa Marttila</p>	
<p>The purpose of this thesis was to investigate website publishing using WordPress. The case company in this study was the advertising agency McCann Worldgroup Helsinki Oy.</p> <p>WordPress is based on open source code that makes it free for use and for software development. The application was originally developed for making blogs. WordPress is also known as content management system and that is why it can also be used for making websites. The application must be installed to a server and it needs a database.</p> <p>WordPress is based on using php elements that are php files. Themes are used for making the layout of the site and adding functionality to it. A theme consists of templates that are smaller parts of the whole site.</p> <p>McCann Worldgroup Helsinki Oy had websites that were based on HTML. The site was rebuilt into WordPress from php elements. The McCann WordPress site needed its own theme and a functionality for updating personal information for the employees and adding new persons to a swf presentation that is shown on the site.</p> <p>WordPress needed modification and the application was enhanced by adding buttons to the admin panel. Functionality was added to buttons so that it is possible to get a person's information from the xml file that will be displayed in a form and saved after submit into a xml file.</p> <p>Loop function was used for displaying the content to get the full advance of the content management system.</p>	
<p>Key words WordPress, php, content management system</p>	

Sisällys

1	Johdanto	1
1.1	Tutkimussuunnitelma	2
2	WordPress julkaisujärjestelmä.....	3
2.1	Historia	3
2.2	Blogit.....	4
2.2.1	Twitter ilmiö.....	5
2.2.2	Blogityökalut	5
2.3	Keskeisiä termejä.....	5
2.4	WordPressin asennus.....	6
2.5	Yleistä WordPressin käytöstä	7
2.5.1	Tiedosto-oikeudet.....	7
2.5.2	Käyttöliittymä.....	8
2.5.3	Dashboard	8
2.6	Sivun rakenne.....	9
2.6.1	Templaten kutsu	11
2.7	Teema.....	11
2.7.1	Teeman muokkaus	12
2.8	Sivun sisältö.....	13
2.8.1	Sisällön hallinta	14
2.8.2	Sisällön syöttäminen.....	14
2.9	Toiminnallisuudet.....	15
2.9.1	Toimintojen hallinta.....	15
2.9.2	Widgetit.....	15
2.10	Pluginit.....	16
2.11	Loop -toiminto	16
2.12	Sivuston testaus	16
3	Websivujen julkaisu WordPressillä Case: McCann Worldgroup Helsinki Oy... 18	
3.1	Sivuston vaatimukset	18
3.1.1	Toiminnalliset vaatimukset	18
3.2	Sivun rakenne.....	18
3.3	Teeman rakennus	19
3.3.1	Header.php.....	19

3.3.2	Index.php.....	21
3.3.3	Footer.php.....	21
3.4	Sisällön laitto	21
3.4.1	Latest.php	22
3.4.2	News.php.....	23
3.4.3	People.php.....	23
3.4.4	Clients.php.....	23
3.5	Toiminnallisuudet ja WordPressin muokkaus	23
3.5.1	Painike hallintavalikkoon.....	23
3.5.2	Henkilön muokkaus ja tallennus	24
3.5.3	Henkilön lisäys.....	28
3.6	Sivuston testaus selaimilla ja iPhonella.....	29
4	Tulokset.....	30
5	Pohdinta	32
Liite 1.	WordPressin käyttöliittymä.....	37
Liite 2.	Dashboardin hallintapaneeli	38
Liite 3.	McCann websivun layout.....	39

1 Johdanto

Tämä dokumentti on opinnäytetyö, joka kuuluu IT-tradenomin tutkinnon opintoihin HAAGA-HELIA Ammattikorkeakoulussa. Opinnäytetyönä tehtävä tutkimus käsittelee WordPress-julkaisujärjestelmän käyttöä websivujen julkaisutyökaluna. Opinnäytetyön teoriaosassa käsitellään WordPressin käyttöä yleisellä tasolla. Kerrotaan sivuston rakentamisen vaiheita ja minkälaisesta kokonaisuudesta WordPressillä toteutettu sivusto muodostuu. Case-esimerkkinä toteutetaan ja kuvataan McCann Worldgroup Helsinki Oy:n websivujen julkaisu WordPressillä. Toteutettavalle sivustolle tehdään toiminnallisuuksia, joiden toteuttamista varten muokataan WordPress-sovellusta.

WordPress on blogien julkaisuun kehitetty sovellus, jota on alettu käyttää myös websivustojen rakentamiseen. WordPress on ilmainen, avoimen lähdekoodin sovellus, jota kehitetään jatkuvasti WordPress-yhteisön toimesta. WordPressiin on saatavana paljon erilaisia vapaasti käytettäviä teemoja. Teemat sisältävät tyyli tiedoston lisäksi toiminnallisuuksia sekä rakenteellisia komponentteja. WordPressiin on saatavana paljon erilaisia ilmaisia plugin laajennuksia, joiden avulla sivustolla voi lisätä toiminnallisuuksia. Plugin:ja on mahdollista tehdä myös itse sekä jakaa niitä muille.

Tutkimuksen tavoitteena on selvittää WordPress-julkaisujärjestelmän käyttöä yleisellä tasolla websivujen julkaisussa. Tutkimukseen liittyvän käytännön osan tavoitteena on julkaista McCann Worldgroup Helsinki Oy:n websivut WordPressiä käyttäen, sekä taustoittaa aihetta teoriatiedoilla. Tavoitteena on selvittää, miten sivusto rakennetaan ja kuinka sen ulkoasuun vaikutetaan WordPressillä. Tutkimuksen tavoitteena on selvittää teoriassa ja käytännössä: Miksi tehdä WordPressillä websivusto? Minkälainen rakenne websivulla on kun se tehdään WordPressillä? Mikä on teema ja mitä sillä tehdään? Mitä ovat plug-init ja mitä niillä tehdään? Miten sivuston saa näkymään selaimissa ja iPhonessa?

McCann Helsingin websivujen julkaisussa tavoitteena on tehdä toimiva ja validi sivusto, jolle luodaan oma teema. Sivuston tulee olla suunnitellun layoutin mukainen. WordPressiin halutaan toiminto, jolla yrityksen työntekijät voivat päivittää henkilötietojaan sekä lisäämään uusia henkilöitä sivustolla näkyvään swf-esitykseen. Julkaistava sivusto toteutetaan niin, että se toimii selaimissa ja iPhonessa.

1.1 Tutkimussuunnitelma

Tutkimuksen tavoitteena on selvittää Wordpress-julkaisujärjestelmän käyttöä websivujen julkaisussa yleisesti. Tutkimukseen liittyvän käytännön osuuden tavoitteena on julkaista McCann Worldgroup Helsinki Oy:n omat websivut Wordpressiä käyttäen, sekä taustoittaa aihetta teorian tiedoilla. Tavoitteena on selvittää, miten sivusto rakennetaan ja kuinka sen layout luodaan Wordpressillä. Tutkimuksen tavoitteena on selvittää teoriassa ja käytännössä: Miksi tehdä Wordpressillä websivusto? Selvitetään sivuston rakennetta ja elementtien käyttöä? Selvitetään mikä on teema ja kuinka sitä muokataan? Kerrotaan plugineista ja niiden tehtävästä sekä siitä miten sivuston saa näkymään selaimissa ja iPhonessa? Selvitetään miten henkilötietojen päivitystoiminto voidaan toteuttaa?

McCann Helsingin websivujen julkaisussa tavoitteena on tehdä toimiva ja validi sivusto, jolle luodaan oma teema, joka määrittää sisällön asettelun sekä siihen liittyvät toiminnallisuudet ja tyylitiedoston. Sivuston tulee olla suunnitellun layoutin mukaiset, joka kaikkein tärkein asia, koska kyseessä on mainostoimisto. Lisäksi tarkoituksena on tehdä sivuilta kytkös Twitterin statukseen niin, että kun henkilötietojen statusta päivitetään yrityksen sivulla, status päivittyy myös Twitteriin. Sivusto toteutetaan niin, että se toimii selaimissa ja iPhonessa.

Tutkimusmenetelminä käytetään aiheeseen liittyvää teorian tietoa, jota sovelletaan käytännön tutkimuksessa jossa toteutetaan ja julkaistaan sivusto WordPressillä. Tutkimus koostuu taustateoriasta sekä tehtävästä sivustosta. Tutkimus tehdään käytännön kokeilujen ja saatavana olevan ohjeistuksen sekä kirjallisuuden ja muiden julkaisujen pohjalta.

Wordpress asennetaan palvelimelle, johon sivusto rakennetaan. Sivuston teko tapahtuu pääosin Wordpressin käyttöliittymästä. Joitakin palvelimella olevia tiedostoja voi muokata Wordpressin käyttöliittymästä ja joitakin täytyy ladata palvelimelta ftp-yhteydellä omalle työkoneelle muokkausta varten ja sen jälkeen siirtää muokattu tiedosto takaisin palvelimelle. Tietokanta, jonka Wordpress vaatii toimiakseen, on mySql-pohjainen. Sivuston koodin validiutta tarkastetaan W3C:n validaattorin avulla. Sisällön syöttämisessä voidaan hyödyntää olemassa olevan sivuston HTML-koodia.

Teorian tietoa etsitään kirjallisista ja sähköisistä julkaisuista. Tietoa Wordpressistä ja sen käytöstä saa parhaiten Wordpressin omilta websivuilta, joista löytyy paljon ohjeistusta ohjelman käyttöön. Tutkimuksen edetessä tehdyt havainnot kirjataan ylös. Havainnot analysoidaan opinnäytetyön tutkimusosassa.

Toimeksiantaja McCann Worldgroup Helsinki Oy saa käyttöönsä WordPressillä toteutetun sivuston, johon toteutetaan henkilötietojen päivitystoiminto sekä uuden henkilön lisäystoiminto. Tutkimustuloksesta on hyötyä sellaiselle, joka haluaa oppia WordPressin käyttöä yleisellä tasolla sekä tekemään sillä websivuston.

2 WordPress julkaisujärjestelmä

Sosiaalinen verkostoituminen internetissä on ajan ilmiö. Ihmiset käyttävät yhä enemmän aikaa kommunikointiin toisten kanssa verkossa. Informaation jakaminen muille verkossa on osa arkipäivää ja tietoa etsitään yhä enemmän verkon avulla. Erilaiset henkilökohtaiset tai yhteisölliset sivustot ovat saavuttaneet suuren suosion ja keskeinen tekijä niiden perustamisessa sekä käytössä on helppous.

WordPress-julkaisujärjestelmä on blogien tekemiseen kehitetty alustariippumaton avoimen lähdekoodin sovellus. WordPress on julkaistu GPL-lisenssillä eli käytännössä tämä tarkoittaa sitä, että WordPressin saa ladata käyttöönsä ilmaiseksi. Sovelluksessa oleva kehittynyt template-järjestelmä on mahdollistanut käytön myös websivujen tekemisessä. WordPress käyttää PHP-ohjelmointikieltä sekä MySQL-tietokantaa, joten se asennetaan palvelinympäristöön. (WordPress 2010.)

WordPressistä voidaan käyttää myös nimitystä sisällönhallintajärjestelmä. Käytännössä tämä tarkoittaa sitä, että esim. sisältöä syöttäessä websivulle ei tarvitse huolehtia itse muusta kuin sisällön kirjoittamisesta ja järjestelmä tekee kaiken työn eli muotoilee sisällön ja laittaa sen oikeaan paikkaan. Tämä edellyttää sitä, sivustoon liittyvät muotoilut ja asetelut on määriteltävä WordPressissä.

2.1 Historia

WordPressin tarina lähti liikkeelle vuonna 2003. Sitä edelsi vuonna 2001 julkaistu b2/cafelog, jolle WordPressistä tuli virallinen seuraaja. Idean kehittäjillä oli halu luoda hyvin rakennettu, vakaa järjestelmä joka soveltui persoonalliseen käyttöön. Sovelluksen arkkitehtuuri tuli olla hienostunutta ja järkevästi rakennettua. Sovellus perustui PHP-ohjelmointiin ja MySQL-tietokannan käyttöön ja sen haluttiin olevan avoimen lähdekoodin sovellus. (WordPress 2010.)

WordPressistä on julkaistu lähes vuosittain uusi versio, joissa on aina uusia ominaisuuksia. Sovellusta kehitetään jatkuvasti satojen kehittäjien toimesta ympäri maailmaa. Toiminta on aina perustunut yhteisön tarpeisiin. (WordPress 2010.)

2.2 Blogit

WordPress kehitettiin alun perin blogien tekemiseen. Blogi mielletään usein verkossa olevaksi päiväkirjaksi, jossa kerrotaan jostakin aiheesta, jota muut käyttäjät voivat lukea ja kommentoida. Yksinkertaisuudessaan blogi on sitä, mutta se voi olla myös paljon muutakin. Sosiaalisen verkostoitumisen myötä yritykset ja mainostajat ovat alkaneet lanseeraamaan kampanjoita, jotka on suunnattu erityisesti blogien käyttäjille siihen ympäristöön, missä he aikaansa viettävät.

Blogi on puhekieleen tullut lyhennös sanasta weblogi. Blogi yleisellä tasolla on jonkun henkilökohtainen websivu, joka on verkossa. Blogi voi koostua monenlaisesta sisällöstä. Esim. kirjallisesta sisällöstä ja kuvista. Sisältö voi olla päiväkirjamaista, pohdintaa, kuulumisten vaihtoa ystävien kesken tai yleistä tietoa kaikille. (Vedje & Gustafsson 2005, 3.)

”Blogi voi olla reaaliaikainen sanomalehti, ajan tasalla oleva valokuvaportfolio tai toimia tiimityössä yhteydenpitovälineenä ja edistymisen seurannan apuvälineenä”(Castro 2005, vii.).

Blogin lukemiseen ja luomiseen tarvitaan tietokone, yhteys internettiin ja webselain. Tekstisisällön laittaminen blogiin edellyttää jonkin tekstinkäsittelyohjelman käyttökokemusta. Kuvien laittaminen sivulle vaatii hieman enemmän käytännön taitoja tietokoneen käytöstä. (Vedje & Gustafsson 2005, 3.)

Blogissa yhdistyvät tietokanta ja websivu järkevällä tavalla. Tietokoneohjelma joka blogia pyörittää, tekee kaiken työn ja antaa kirjoittajan keskittyä vain kirjoittamiseen. Tietokoneohjelma laittaa päivämäärän kirjoituksiin ja arkistoi ne aikajärjestykseen. Halutessaan aikaisempia kirjoituksia voi katsoa jälkeenpäin. (Castro 2005, vii.)

2.2.1 Twitter ilmiö

Twitter on osa sosiaalisen verkostoitumisen ilmiötä. Twitterissä, kuten muissa vastaavissa sivustoissa, luodaan oma henkilökohtainen sivu. Se kuitenkin eroaa suurista kilpailijoistaan ratkaisevalla tavalla. Sivulle kirjoitetaan viestejä, jotka saavat olla korkeintaan 140 merkkiä pitkiä. Twitterissä on omat toimintatavat, säännöt ja potentiaali käytön suhteen. (Smith & Llinares 2009, vi-vii.)

Twitterin idea perustuu lyhyen viestin kirjoitukseen, jossa voi kertoa esim. maailmalle mitä teet parhaillasi. Vastaavasti voit lukea lyhyesti mitä muu maailma tekee viesteistä, jotka ovat korkeintaan 140 merkkiä pitkiä. Käyttö on helppoa ja nopeaa, joka on vaikuttanut Twitterin nopeaan suosion kasvuun. Yritykset ovat hyvin kiinnostuneita Twitterin mahdollisuuksista, joka on johtanut markkinointikampanjoihin sekä lisäsovellusten kehitykseen. (Smith & Llinares 2009, 3-4.)

2.2.2 Blogityökalut

Blogien tekemiseen on olemassa useita työkaluja mm Blogger, LiveJournal ja WordPress. WordPressin etuna on alustariippumattomuus eli se ei vaadi tiettyä järjestelmää toimiakseen. Keskeistä blogityökaluissa on käytön helppous ja sisällönhallinta ominaisuudet. Yksinkertaisen blogin luominen käy nopeasti ja helposti. Ohjelmointia ei tarvitse hallita eikä myöskään tehdä, perustaakseen blogin. Sivulle saa lisättyä jopa valmiita toimintoja blogityökalusta ilman ohjelmoinnin tuntemusta. Pieniä ulkonäöllisiä muutoksia blogiin voi tehdä suoraan blogityökalusta, mutta vähänkin suuremmat muutokset ulkonäköön ja toimintoihin vaativat (X)HTML - ja PHP -ohjelmoinnin tuntemusta. Blogista on mahdollista rakentaa täysin omanlaisensa ratkaisun sekä ulkonäöllisesti että toiminnallisesti

2.3 Keskeisiä termejä

Archives eli arkistointi on tehokas tapa jäsenellä ja säilyttää tietoa blogissa. Kirjoitetut viestit saavat päivämäärätunnisteen, jolla niitä voi etsiä jälkepäin. Etsiä voi myös kirjoittajan nimen tai aakkosellisen haun mukaan. (WordPress 2010, Codex.)

Blogroll on lista linkeistä. Blogin pitäjä voi listata itselleen tärkeitä linkkejä listaan, josta halukkaat voivat käydä niitä klikkaamassa. (WordPress 2010, Codex.)

Feeds on toiminto, jolla voi katsoa uusimpia päivityksiä sivulta ja lähettää uudesta sisällöstä tiedon johonkin toiseen blogiin. (WordPress 2010, Codex.)

Plugin on scriptti eli ohjelmakoodia, joka lisää blogille jonkin toiminnallisen lisän. Joitakin käyttövalmiita Plugin toimintoja saa ladattua verkosta ilmaiseksi. Plugin toimintoja voi tehdä myös itse. (WordPress 2010, Codex.)

Post eli kirjoitus ja sisältö on koko blogin olemassaolon päätarkoitus. Blogien käyttäjät lukevat ja kirjoittavat viestejä. (WordPress 2010, Codex.)

Syndication on toiminto, joka ilmoittaa haluamasi blogien uusista päivityksistä. Idea on se, että käyttäjän ei tarvitse itse tutkia kaikkia blogeja tarkistaakseen onko päivityksiä tehty. Suurin osa blogeista tukee tällaista toimintoa. (WordPress 2010, Codex.)

Template on jokin osa sivun pohjakokonaisuudesta, joista sivun rakenne muodostetaan. Sivun rakentaminen WordPressillä perustuu Template:in käyttöön. (WordPress 2010, Codex.)

2.4 WordPressin asennus

WordPress asentuu helposti ja nopeasti, jos on vähänkään aikaisempaa kokemusta jonkin ohjelmiston asentamisesta. WordPress täytyy asentaa palvelimelle ja se tarvitsee toimiakseen MySQL –tietokannan.

1. Asennus alkaa WordPressin lataamisella WordPressin websivuilta osoitteesta <http://www.WordPress.org/>. Lataamisen jälkeen puretaan zip-tiedosto.
2. Luodaan MySQL tietokanta sellaiselle käyttäjälle, jolla on käyttöoikeudet tietojen muokkaamiseen.
3. Muutetaan wp-config-sample.php tiedoston nimi wp-config.php:ksi.

4. Avataan wp-config.php –tiedosto, jollakin tekstieditorilla, ja laitetaan siihen tietokantaan liittyvät tiedot niille osoitettuihin paikkoihin. Tietokannan tiedot voi myös laittaa web-selaimen kautta wp-admin/setup-config.php tiedoston kautta suoraan palvelimelle.

5. Siirretään WordPressin tiedostot palvelimelle siihen paikkaan, jossa WordPressiä halutaan käyttää. Joko suoraan juurihakemistoon, jolloin WordPress toimii esim. osoitteessa <http://www.palvelin.com/>. Tai luodaan juurihakemiston alle kansio WordPressiä varten. Tällöin WordPress toimisi esim. osoitteessa <http://www.palvelin.com/WordPress/>.

6. Installointi suoritetaan ajamalla install.php –tiedosto palvelimella, siinä paikassa johon WordPress on asennettu. Install.php löytyy wp-admin –kansioista. Tämän jälkeen WordPress on asennettu ja valmis käyttöön. (WordPress 2010, Codex.)

2.5 Yleistä WordPressin käytöstä

WordPress asennetaan palvelimelle, josta sitä käytetään web-selaimen kautta. Hallinnolliset toiminnot edellyttävät kirjautumista WordPressiin sisään tunnuksilla, jotka on saatu sähköpostiin asennusvaiheessa annettuun sähköpostiosoitteeseen. Hallintaan liittyvät tiedostot löytyvät wp-admin –kansion alta. WordPressistä ei kuitenkaan pääse käsiksi palvelimella oleviin tiedostoihin suoraan, muuten kuin template -editorilla. Template –editorilla muokataan templateja, jotka ovat php-elementtejä, joista sivusto koostetaan. Jos esim. templatien tietoja muuttaa ja haluaa tallentaa muutokset, täytyy tiedoston oikeudet muuttaa palvelimelta ftp-yhteyden avulla.

2.5.1 Tiedosto-oikeudet

Oletuksena kaikilla tiedostoilla on vain lukuoikeudet ja tiedoston omistajalla on kirjoitusoikeus. Tiedosto-oikeudet täytyy muuttaa, jotta niiden muokkaaminen on mahdollista WordPressin käyttöliittymästä. Jos tiedostoa yrittää tallentaa muokkauksen jälkeen ja sen oikeuksia ei ole muutettu, WordPress ei anna tallentaa kyseistä tiedostoa. Tiedostojen oikeuksia pääsee muokkaamaan ottamalla ftp-yhteyden palvelimeen, johon WordPress on asennettu. Wp-content kansion alta löytyy kansio themes, jonka alta löytyy kaikki käytössä olevat teemat. Valitaan tema, jonka tiedostoja halutaan muokata, ja muokataan niiden tiedostojen oikeuksia, jotka halutaan tallentaa WordPressissä. Riippuen ftp-ohjelmasta, jota käytetään, mutta useimmissa tiedostojen oikeuksia pääsee muokkaamaan painamalla hiiren oikeaa painiketta halutun tiedoston kohdalla, jolloin aukeavasta ikkunasta valitaan tiedosto-oikeudet.

Tiedosto-oikeudet määritellään numeroilla tai valitsemalla hiirellä ikkunasta halutut kohdat eli ruksataan laatikoista ne kohdat, jotka halutaan valita. Tiedostolle määritetään oikeudet erikseen omistajalle, ryhmälle sekä julkiselle käyttäjälle. Jokaisessa ryhmässä on mahdollista antaa oikeudet lukemiseen, kirjoittamiseen ja suorittamiseen. Lukuoikeus on kaikissa ryhmissä oletuksena. Jotta tiedostoja pääsee muokkaamaan ja tallentaminen on mahdollista, täytyy oikeuksia sallia lisää.

Tiedosto-oikeudet voidaan määrittää myös numeraalisesti. Numerokoodi muodostuu kolmesta numerosta. Ensimmäinen numero kertoo tiedosto-oikeudet tiedoston omistajalle. Toinen numero kertoo tiedosto-oikeudet ryhmälle ja kolmas numero kertoo tiedosto-oikeudet julkiselle käyttäjälle. Koodissa numero neljä ilmaisee lukuoikeutta, numero kuusi luku- ja kirjoitusoikeutta ja luku seitsemän luku-, kirjoitus- ja suoritusoikeutta. Eli kun oletuksena kaikilla tiedostoilla on vain lukuoikeus ja omistajalla kirjoitusoikeus, ilmaistaan sama asia numeraalisesti koodilla 644. Jos kaikki oikeudet sallitaan kaikille käyttäjille, ilmaistaan asia koodilla 777. (WordPress 2010, Codex.)

2.5.2 Käyttöliittymä

WordPressin käyttöliittymä ei poikkea ratkaisevasti mistään totutusta käyttöliittymämallista, joita yleensä sovelluksissa käytetään. Alkuun saattaa aiheuttaa ongelmia ruudulla näkyvä informaation määrä, mutta kun siihen tottuu, käyttöliittymä on selkeä. Päästäkseen sisään hallintatoimintoihin, täytyy WordPressiin kirjautua admin -tunnuksilla. Liitteessä 1. on kuva WordPressin käyttöliittymästä.

2.5.3 Dashboard

Dashboard on sivuston hallinnan kotisivu, joka aukeaa ensimmäisenä kun kirjaudutaan sisään. Dashboardiin löytyy myös painike sivun vasemmasta yläkulmasta, joka toimii samaan tapaan kuin webselaimen home -näppäin. Varsinainen hallintapaneeli on sivun vasemmassa reunassa ja siitä löytyvät kaikki käytettävät toiminnot, joita WordPressistä sisältää. Hallintapaneeli on jaoteltu aiheittain kahteen osaan. Sisällölliset asiat ovat omana kokonaisuutena ja toiminnalliset asiat omana kokonaisuutena. Liitteessä 2. on kuvattu Dashboardin hallintapaneeli. Dashboardin yläreunassa on visit site -painike, josta voi katsoa miltä sivusto näyttää selaimessa.

Sisältö:

- Posts
- Media
- Links
- Pages
- Comments

Toiminnot:

- Appearance
- Plugins
- Users
- Tools
- Settings

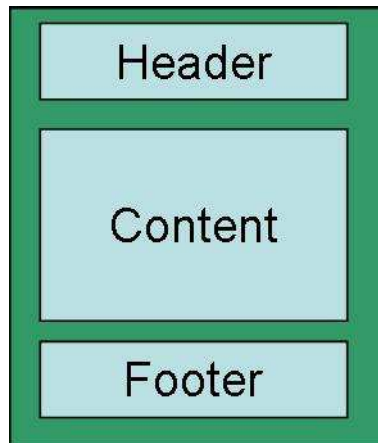
2.6 Sivun rakenne

WordPressissä sivun rakenne muodostuu itsenäisistä elementeistä, jotka yhdessä muodostavat sivun. CSS-tyylitiedostossa määritellään esim. käytettävien fonttien koot ja väri. Sivun rakenteelliset elementit eli templatet ovat .php -päätteisiä tiedostoja ja voivat sisältää mm. XHTML koodia. Templateja käytetään yleensä index.php -tiedoston avulla. Index.php:ta käytetään sivuston perustana siitä johtuen, että webselain lataa aina index -nimisen tiedoston oletuksena. Index.php:tä voi myös käyttää sisäisenä sivuna, jolle tuodaan varsinainen sivun sisältö. Tällöin home.php otetaan käyttöön ja WordPress käyttää sitä automaattisesti sivuston perustana.

Minkä tahansa php -elementin voi tuoda templateen include -komennolla. Esim. index.php:ssä määritetään missä kohdassa sivua esim. content.php näytetään. WordPressissä joidenkin templateiden nimet ovat standardeja. Eli esim. header.php ja footer.php ovat varattuja nimiä. Käytännössä näihin voi tehdä hieman erilaisia kutsuja, kuin muihin normaaleihin templateihin.

Perussivumalli koostuu kolmesta elementistä. Kuvassa 1. on esitetty yksinkertainen kolmesta elementistä koostuva sivumalli.

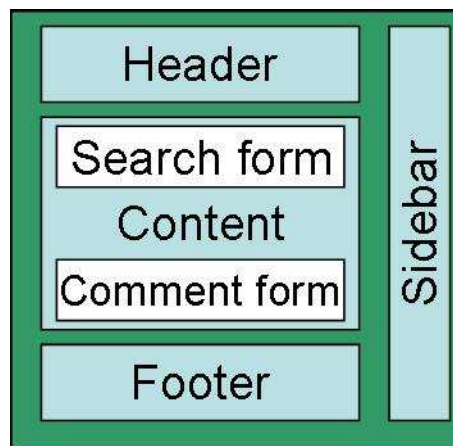
- Header
- Content
- Footer



Kuva 1. Perussivumalli

Monimutkaisempi sivumalli voi koostua useasta elementistä Kuvassa 2. on esitetty sivurakenne, joka koostuu kuudesta erillisestä elementistä.

- Header
- Content
- Comment Form
- Sidebar
- Search Form
- Footer



Kuva 2. Kuudesta eri elementistä koostuva sivumalli

(WordPress 2010, Codex.)

Kun sivu tehdään (X)HTML –koodilla, sen täytyy sisältää html- , head- ja body –tagit. Sivua voidaan tuoda sellaisenaan WordPressiin esim. index.php:lle. WordPressin varsinainen hyöty siitä saavutetaan kuitenkin, vasta kun se jaetaan usealle templatelle esim. header -, content - ja footer -templatelle. (X)HTML –tagit pysyvät samoissa paikoissa kuin ne alkuperäisessä tiedostossa olivatkin. Nyt vain alkutagi ja lopputagi voivat olla eri PHP –elementillä. Esim. html –tagi alkaa headerr.php:ssa ja loppuu footer.php:ssa. Tiedostoa muokatessa, tulee tarkastaa että alku- ja lopputagit ovat oikeissa paikoissa.

2.6.1 Templaten kutsu

Templateen liittyvä kutsun muoto riippuu siitä, onko templaten nimi koodattu WordPressiin. Jotkin nimet ovat määriteltä standardiksi esim. header.php ja index.php. Templaten voi nimetä myös haluamakseen, mutta silloin ei saa käyttää WordPressiin koodattuja eli varattuja nimiä. Esim. some_content2.php tai textform.php käyvät hyvin.

Kutsu WordPressiin koodatun standardin mukaiseen templateen on seuraavanlainen:

```
<?php get_header(); ?>
```

Kutsu laitetaan esim. index.php –tiedostoon siihen kohtaan, jossa halutaan tuoda header –elementti näkyviin.

Kutsu sellaiseen templateen, jonka nimi ei ole koodattu WordPressiin:

```
<?php include (TEMPLATEPATH . ”/some_content2.php”); ?>
```

(Blakeley Silver 2008, 70-71.)

2.7 Teema

Teema muodostaa sivuston ulkoasun tyylin. Teema tarkoittaa sivun ulkoasun lisäksi siihen liitettyjä toimintoja. Teeman muodostaa CSS-tyylitiedostosto, templateet ja toiminnot. WordPressissä on mukana kaksi erilaista teemaa, jotka ovat selkeitä ja yksinkertaisia. Jos WordPressin valmiit teemat eivät miellytä, vapaasti käytettäviä teemoja voi ladata ilmaiseksi verkosta. Toinen vaihtoehto on luoda oma teema juuri sellaiseksi kuin itse haluaa. (WordPress 2010.)

WordPressissa teema on sivun design johon WordPress generoi sisällön. Teeman sisällä on monta yksilöllistä template -tiedostoa. Nämä tiedostot muodostuvat pääosin XHTML ja PHP –koodista, määrittäen sivun, sivun sisällön ja toiminnallisuuden. Teemaa muodostuu tyylitiedostosta, PHP-skriptistä, Javaskriptistä ja kuvista. Nämä elementit yhdessä template:in kanssa muodostavat teeman kokonaisuuden. (Blakeley Silver 2008, 62-63.)

Teemaan liittyy ulkoisen annin lisäksi toiminnallisuus. Esim. postien organisointi ja arkistointi voivat olla jossakin teemassa toteutettu selkeämmin ja tehokkaammin kuin jossain toisessa. Sivuston tai blogin ulkoasulla on suuri merkitys käyttäjiä ajatellen, riippumatta niiden sisällöstä. Vaikka sivu sisältäisi paljon hyödyllistä informaatiota, sen löytäminen voi olla vaikeaa, jos tieto on organisoitu huonosti. Käyttäjä saattaa myös muodostaa mielikuvan sisällön oikeellisuudesta sivun ulosannin perusteella. Kun taas vaatimaton sisältö voi saada käyttäjiltä enemmän huomiota, jos se esitetään käyttäjää kiinnostavalla tavalla.

2.7.1 Teeman muokkaus

WordPressissä on valmiina kaksi teemaa, Classic ja Default. Näitä saa käyttää vapaasti omissa sivustoissa ja blogeissa. Teemaa voi esikatsella WordPressissä ennen kuin tekee päätöksen haluaako sitä käyttää. WordPress generoi miltä sivu näyttää oikeasti ja laittaa jopa oikean sisällön siihen, mikäli se on olemassa jollakin templatella. (WordPress 2010, Codex.)

Jos valmiit teemat eivät miellytä sellaisenaan, voi niitä muokata haluamaansa muotoon. Teeman voi luoda myös kokonaan alusta asti, jolloin saa käyttöönsä yksilöllisen ratkaisun. Monet kehittäjät tekevät omia teemoja jakeakseen niitä muille. Teeman tekijä yleensä kommentoi koodia, jotta jatkokehitys ja muokkaus on helpompaa. On hyvin yleistä, että halutaan lisätä valmiiseen teemaan jotain omaa. WordPressissä on muokkaus mahdollisuus otsikko-osaan, jolla saa nopeasti muutettua valmista pohjaa. Tällä tavoin saa muutettua mm. otsikon fonttia ja taustaväriä. Tämä on kuitenkin rajoittunutta muokkausta ja esim. kuvan liittäminen otsikkoon ei onnistu tällä menetelmällä. Teeman muokkaaminen on siis ainoa vaihtoehto, jos haluaa luoda jotain yksilöllisempää sivumallia sekä ulkoasua.

Teeman muokkaus on helpointa siten, että kopio esim. default –teeman kaikki tiedostot palvelimelta uuteen kansioon, jolle on antanut uutta teemaa kuvastavan nimen. Toinen vaihtoehto on monistaa default –kansio ja antaa sille uusi nimi. Tämän jälkeen avataan kopioitu kansio ja annetaan muutamille tiedostoille uusi nimi. Home.php, page.php, category.php, archive.php ja single.php -nimiset tiedostot täytyy uudelleen nimetä vaikka liittämällä jokaisen alkuun jonkin etuliitteen esim. alkup_home.php. Toinen vaihtoehto on poistaa nämä edellä mainitut tiedostot kokonaan. Tämä täytyy tehdä siitä johtuen, että alkuperäisessä teemassa, josta tiedostot on kopioitu, on liitetty toiminnallisuuksia ja viittauksia tämän nimisiin tiedostoihin. Lisäksi on hyvä kopioida tai monistaa myös index.php ja CSS-tyylitiedosto ja nimetä ne samaan tapaan. (Blakeley Silver 2008, 67.)

Default –teemasta kopioidun CSS-tyylitiedoston alussa on määritelty teemaan liittyviä tietoja. Mm. teeman nimi, URI ja kuvaus. Nämä kannattaa muuttaa vastaamaan oman teeman tietoja, koska kun teema halutaan ottaa käyttöön WordPressissä, se löytäminen on helpompaa, varsinkin jos WordPressiin on ladattu useita teemoja. Oman teeman tiedot tulevat näkyviin kysyisen teeman kohdalle WordPressissä, kun ne on kirjoitettu style.css –tiedoston alussa oleviin kommentoituihin riveihin. Jos halutaan luoda täysin oma teema, kannattaa sen CSS-tyylitiedoston alkuun kopioida jonkin toisen teeman tyylitiedoston alussa olevat kommentoidut määrittämissrivit ja muuttaa tiedot oman teeman mukaisiksi. (Blakeley Silver 2008, 68.)

Teeman muokkaamisella pyritään saamaan aikaan hyvän näköinen sivu tai blogi, joka toimii järkevästi.

2.8 Sivun sisältö

Yleensä sivu tai blogi muodostuu erilaisista sisällöllisistä elementeistä. Teksti on tärkeä elementti, koska se yleensä ilmaisee esitetyn asian yksityiskohtaisesti. Blakeley Silverin mukaan websivun rakennus kannattaa aloittaa tekstin tuomisella sivulle. Se voi olla joko oikeaa sivulle tulevaa sisältöä tai sitten jotain muuta tekstiä. Ideana on antaa tekstile riittävästi tilaa ja tuoda muut elementit vasta sen jälkeen sivulle.

Kuvat toimivat hyvin tekstin ohella, joskus jopa itsenäisinä elementteinä. Usein esim. tuotekuvat tuovat lisäinformaatiota käyttäjälle. Sivun layout voi perustua jonkin kuvan ympärille, mutta yritysten websivuilla yleensä layout rakennetaan suuremmasta kokonaisuudesta, johon kuvia liitetään täydentämään kokonaisuutta. Lähes jokaisella sivulla on jokin kuva.

Animaatio ja liikkuvat elementit mielletään mainoksiin ja bannereihin. Niitä käytetään tehokkaasti huomion herättämiseen ja tarinan kertomiseen. Liikettä voi tuoda myös omalle sivulle, mutta kuitenkin hillitysti. Jos liikettä on liian paljon, se voi hidastaa sivun käyttöä ja jopa hankaloittaa informaation löytymistä. Hyvällä animaatiolla tuodaan käyttäjälle lisää informaatiota niin, että sivun käyttötarkoitus ja käytettävyys säilyvät.

Video –tiedostot ovat tehokkaita ja mielenkiintoisia elementtejä sivulle. Ne voivat parhaimmillaan olla koko sivuston perusta esim. Youtube. Huonosti toteutettuna video –tiedoston käyttö hidastaa ja hankaloittaa käytettävyyttä ja voi ohjata käyttäjän pois toiselle sivulle. Tiedostokoot kasvavat nopeasti video –tiedostojen myötä.

2.8.1 Sisällön hallinta

Hallintapaneelissa sivun sisältöön liittyvä asiat ovat suoraan Dashboardin alla olevassa kehystetyssä laatikossa. Liitteessä 2. on kuvattu hallintapaneeli.

Posts –painikkeen alta löytyy toiminnot olemassa olevan postin editointiin, uuden postin luomiseen, postin tagien laittamiseen ja kategorioimiseen.

Media –painikkeen alta pääsee media kirjastoon tai lisäämään uutta mediaa.

Links –painikkeen alta voi editoida linkkejä, lisätä uusia linkkejä ja kategorioida linkkejä.

Pages –painikkeen alta voi editoida sivuja ja lisätä uusia sivuja.

Comments –painikkeen alta voi lisätä kommentteja sivulle.

2.8.2 Sisällön syöttäminen

Sisällön syöttämiseen on kaksi mahdollisuutta, joko koodi- tai visuaalinen näkymä.

Visuaalisessa näkymässä tekstiä voi kirjoittaa tekstieditorin tapaan. Pieniä muokkauksia voi tehdä suoraan tekstiin sekä lisätä kuvia ja muuta mediaa. Suuremmat muotoilut tulee kuitenkin tehdä tyylitiedostossa.

Koodinäkyvässä WordPressiin syötetään (X)HTML ja PHP-koodia. Olemassa olevia template tiedostoja voi muokata editorilla ja tallentaa muutokset, jos tiedosto-oikeudet kyseisen tiedoston osalta sen sallivat. Uuden templatien voi luoda joko kopiomalla jonkin olemassa olevan templatien palvelimella, tai viemällä palvelimelle uuden php –päätteisen tiedoston. Uusi tiedosto ilmestyy WordPressiin template – editorin oikean puoleiseen reunaan, muiden template:in joukkoon. Myös täysin uuden templatien tiedosto-oikeudet täytyy muuttaa, jotta WordPress antaa tallentaa muutokset.

2.9 Toiminnallisuudet

Sivun tai blogin käyttöä voidaan tehostaa lisäämällä siihen toiminnallisuutta.

Toiminnallisuudella tarkoitetaan asioita, jotka tuovat käyttäjälle jotain helpotusta tai ekstraa sivuston käyttämiseen. Esim. sivulla voisi olla ajan tasalla oleva kalenteri ja kello.

Toiminnallisuus voidaan liittää myös teemaan. Jollakin teemalla voi olla sille ominainen toiminto, jota ei muissa teemoissa ole. Esim. tapa organisoida ja näyttää vanhoja posteja. (WordPress 2010.)

2.9.1 Toimintojen hallinta

Hallintapaneelissa toimintoihin liittyvät asiat ovat omassa kehystetyssä laatikossa, sisältöön liittyvien asioiden alapuolella. Liitteessä 2. on kuvattu hallintapaneeli.

Appearance –painikkeen alta löytyy teemat, Widgetsit, template-editori, uusien teemojen lisäys ja headerin eli otsikko-osan kustomointi.

Plugins –painikkeen alta näkee olemassa olevat pluginit, voi lisätä uusia tai editoida plugineja.

Users –painikkeen alta voi muokata käyttäjiä, lisätä käyttäjiä ja voi katsoa omaa profilia.

Tools –painikkeen alta löytyy työkalut, import –toiminto, export –toiminto ja päivitys.

Settings –painikkeen alta löytyy asetukset yleisesti, kirjoitukseen, lukemiseen, keskusteluun, mediaan, yksityisyyteen, linkkeihin ja sekalaisiin toimintoihin.

2.9.2 Widgetit

Widgetit ovat helppo tapa lisätä toiminnallisuutta sivulle. WordPressissä on useita erilaisia Widgettejä, joita voi vetää hiirellä sivulle ilman että tarvitsee koodata mitään. Sivulle voi tuoda mm. kalenterin, arkiston, linkkikokoelman ja sivuluettelon. Jotta Widgetteja voi käyttää, täytyy käytössä olevan teeman asetuksissa määrittää, että Widgetit ovat käytössä. (WordPress 2010, Codex.)

2.10 Pluginit

Pluginit tunnetaan myös nimellä laajennusosa. Pluginit ovat skriptejä eli koodeja, jotka tekevät jotain toiminnallista sivustolle. Esim. säätietopalvelu tai postin organisointia. Pluginit ovat vapaaehtoisten kehittäjien tekemiä, jotka pitävät haasteista ja ongelmien ratkaisemisesta. (WordPress 2010, Codex.)

WordPressin yhteisössä on jatkuvasti kasvava plugin kirjasto, jota kehittäjät tukevat. Pluginien avulla voi laajentaa serverillä toteutettavan WordPress -sivun toimintoja. Esim kalenteri, kuvagalleria tai ostoskori voisivat olla tällaisia toimintoja. Lähes mitä vain mieleen tulee, WordPressiin löytyy plugin, joka auttaa ongelman ratkaisussa. (Blakeley Silver 2008, 8.)

2.11 Loop -toiminto

Index.php -tiedostossa sekä muissa template:ssa voidaan käyttää loop -toimintoa. Loop toistaa jonkin sisällön halutuilla sivuilla. Loop:iin voidaan laittaa PHP tai XHTML -koodia, joka voidaan laittaa toistumaan tiettyihin tageihin. (WordPress 2010, Codex.)

Loop -toiminnon avulla saadaan WordPressin tehokkuus käyttöön. Luomalla ja julkaisemalla sisällöstä post muotoista, voidaan WordPressissä määrittää kuinka paljon julkaisuja kerralla näytetään. Loop -toimintolla voidaan näyttää haluttu määrä posteja.

2.12 Sivuston testaus

Sivuston testaus täytyy tehdä useammalla kuin yhdellä selaimella. Eri selaimilla sivut toimivat hieman eritavalla, joten on hyvä testata ajoissa, miltä sivu näyttää eri selaimissa. Testausta kannattaa tehdä jo hyvinkin alkuvaiheessa ja jatkaa sitä koko ajan, jotta mahdolliset ongelmatilanteet saa korjattua. Mahdolliset ongelmat on helpompi korjata alussa kuin aivan lopussa. W3C -validaattoria kannattaa käyttää (X)HTML:n ja CSS:n validointiin.

Blakeley Silver mukaan kannattaa validoida vaiheittain kun työstää teemaa. Kun lisää jotain koodia, kannattaa ensin katsoa miltä sivu näyttää Firefoxilla. Sen jälkeen validoi koodin validaattorilla, jonka jälkeen voi katsoa miten InternetExplorer näyttää sivun. Kannattaa huomioida myös muut mahdolliset selaimet, joita tiedät kohderyhmän käyttävän. Lopuksi

validoi vielä uudestaan jos se on tarpeen. Sitten voi lisätä koodia ja toistaa tarkistusprosessi aina uudestaan jokaisen vaiheen jälkeen kunnes teema on valmis.

”Jos sivu ei näytä selaimessa siltä, miltä sen pitäisi näyttää tai sivu ei toimi niin kuin sen pitäisi toimia tai sivu ei validoidu, täytyy pysähtyä ja selvittää mistä ongelma johtuu.”(Blakeley Silver 2008, 94).

3 Websivujen julkaisu WordPressillä Case: McCann Worldgroup Helsinki Oy

3.1 Sivuston vaatimukset

Sivuston layoutin suunnittelun on tehnyt AD Jani Nummela mainostoimisto McCann Worldgroup Helsinki Oy:sta. Ensisijaisen tärkeää on toteuttaa sivuston ulkoasu suunnitellun mukaisena, joka on kuvattu liitteessä 3. Koska kyseessä on mainostoimisto, sivuston ulkoasu on tärkeämpi kuin mikään muu yksittäinen tekijä. Seuraavaksi tärkeimpiä vaatimuksia ovat toimiva sivusto ja validi koodi.

3.1.1 Toiminnalliset vaatimukset

Mainostoimiston työntekijöille haluttiin mahdollisuus päivittää omia henkilötietoja sivustolle. Tiedot näkyvät sivun alaosassa olevassa Flashilla toteutetussa swf-esityksessä, jossa jokaisella työntekijällä on oma laatikko omille tiedoilleen. Toimintoa varten haluttiin WordPressin valikkoon uusi painike, josta pääsee henkilötietoja käsittelevään sivuun. Sivulle haluttiin saada lista henkilöistä, josta valitsemalla avautuu uusi ikkuna jossa on valitun henkilön tiedot. Henkilötietoihin haluttiin muokkausmahdollisuus sekä muokatun profiilin tallennus. Kokonaan uuden henkilön lisäys listaan haluttiin myös mahdolliseksi. Henkilötietoja varten oli olemassa xml-tiedosto, jossa oli listattuna kaikki työntekijät sekä lisäksi jokaisella työntekijällä oli henkilökohtainen xml-tiedosto omille tiedoilleen. Näitä xml-tiedostoja tuli käyttää henkilötietojen muokkaustoiminnon toteutuksessa.

Sivuston haluttiin käyttävän loop-toimintoa sekä post:a, jotta sisällönhallintaominaisuudet saadaan käyttöön. Sisältö muoto saadaan tällä tavoin post:ssa samanlaiseksi ja julkaisut näkyvät kronologisessa järjestyksessä. Sivuston tulee toimia selaimissa ja iPhonessa.

3.2 Sivun rakenne

Sivuston rakenne oli yksinkertainen, koska sivuston ulkoasu oli helppo jakaa selkeisiin osiin. Sivun muodostuu header, index ja footer –osista. Headerissa on yrityksen logo ja yhteystietopalkki. Indexissa on kaikki sivun sisällöllinen osuus, jossa käytetään post –toimintoa

ja WordPress -looppia. Postilla on helppo lisätä sivulle uutta sisältöä ja WordPress näyttää uusimman postin ylimpänä eli ensimmäisenä.

3.3 Teeman rakennus

Sivuston ulkoasu toteutus tehtiin olemassa olevan sivuston mukaiseksi. Käytännössä tämä tarkoitti sitä, että ulkoasun toteuttaminen WordPressiin vaati joko täysin uuden teeman luomista tai suuria muutoksia jonkin valmiin teeman template:in. HTML –pohjaisen sivuston tyylitiedoston sai käyttöön pienen muokkauksen jälkeen WordPress -sivustossa. Sivustoa varten luotiin McCann:lle oma teema. Siinä käytettiin pohjana Default –teeman templateja, joita muokattiin. Sivuston rakenteen yksinkertaisuuden takia oli mielekästä lähteä muokkaamaan valmista teemaa. Default –teemassa oli valmiiksi kaikki rakenteen toteuttamiseen tarvittavat templatet.

CSS-tyylitiedoston alkuun lisättiin teemaan liittyvät tiedot, joiden avulla teeman käyttöön ottaminen on helpompaa. Jos tietoja ei määrittele tyylitiedostossa, oman teeman löytäminen WordPressissä voi olla vaikeaa, jos teemoja on paljon. Teemaan liittyvät tiedot laitetaan kommenttimerkkien sisään ja ne sijoitetaan aivan tyylitiedoston alkuun.

```
/*  
Theme Name: McCann  
Theme URI: http://www.mccann.fi/  
Description: McCann Worldgroup Helsinki Oy.  
Version: 1.0  
Author: Juha Nordlund  
Author URI:  
Tags: mccann  
*/
```

3.3.1 Header.php

Header.php:in tuli kaikki se tieto mikä on HTML-dokumentin <head></head> -tagien sisällä. Header.php:n alussa määritellään minkälaisesta dokumentista on kyse eli avataan <html> -tagi, sekä käytettävät tyylitiedostot ja sivuston metatiedot.


```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
<title>McCann Worldgroup Helsinki Oy</title>
<link href="resetti.css" rel="stylesheet" type="text/css" />
<link href="style.css" rel="stylesheet" type="text/css" />
<!--[if !IE]>-->
<link rel="stylesheet" href="iphone.css" type="text/css" media="only screen and (max-device-width:
480px)" />
<meta name="viewport" content="width=device-width" />
<!--<![endif]>-->
<!--<link rel="stylesheet" href="iphone.css" type="text/css" /> -->
<script type="text/javascript" src="swfobject.js"></script>
<script src="Scripts/swfobject_modified.js" type="text/javascript"></script>
<script type="text/javascript">
swfobject.embedSWF("mccann_fi.swf", "content", "300", "120", "9.0.0", "expressInstall.swf");
</script>
<style type="text/css" media="screen">
@import url("<?php bloginfo('stylesheet_url');?>");
</style>
</head>

```

Sisällölliset asiat, jotka header.php:lle tulivat olivat yrityksen logo sekä yhteystietopalkki. Koska nämä olivat näkyviä osia sivusta, ne tarvitsivat aloittavan <body> -tagin. Sulkeva </body> -tagi sijoitettiin vasta siihen tiedostoon, jossa sivun sisältö näyttäminen loppuu. McCann – teemassa sivun kokonaisuus muodostuu siis kolmesta templatesta ja nämä kaikki ovat sisällöllisiä elementtejä.

```

<body>

<!-- header start -->
<div id="vrapperi">
<div class="otsake"></div>
<div class="palkki palkkiyla"></div>
<div class="kontakti_boxi">

```

```
<div class="teksti tekstikontaktissa">McCANN WORLDGROUP HELSINKI OY | PIENI  
ROOBERTINKATU 1-3 00130 HELSINKI | TEL. +358 424 6311 | E-MAIL info@mccann.fi ,  
petteri.lillberg@mccann.fi</div>  
</div>  
<!-- header end -->
```

3.3.2 Index.php

Index.php toimii varsinaisen sisällön alustana ja sisältää vain viittaukset elementeistä, joista sisältö muodostetaan. Alkuperäisessä HTML -muotoisessa sivussa kaikki asiat ovat yhdessä dokumentissa. WordPressiin tuotaessa HTML -koodi jaettiin omiin kokonaisuuksiin sivun otsikoiden mukaan ja näistä kaikista tehtiin oma php-tiedosto. Luodut php-tiedostot oli järkevää nimetä kyseisen osion otsikon mukaisesti. Nämä yksittäiset php-tiedostot tuodaan index.php:lle get- ja include-toiminnolla. Index.php:n lopussa suljetaan sekä </body> että </html> -tagit.

```
<?php get_header(); ?>  
<?php include (TEMPLATEPATH . '/latest.php'); ?>  
<?php include (TEMPLATEPATH . '/news.php'); ?>  
<?php include (TEMPLATEPATH . '/people.php'); ?>  
<?php include (TEMPLATEPATH . '/clients.php'); ?>  
<?php get_footer(); ?>  
</body>  
</html>
```

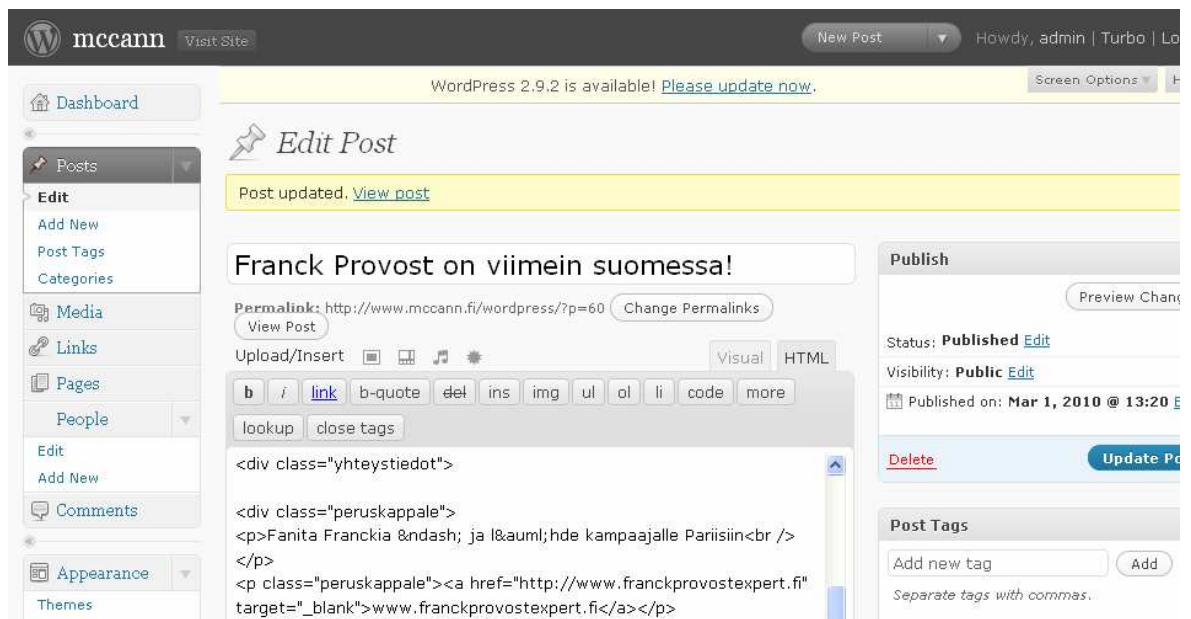
3.3.3 Footer.php

Footer.php:le tuli yrityksen logo ja harmaa palkki. Tämä on järkevää tehdä kuitenkin omaksi elementiksi, sillä sen lisääminen tarvittaviin kohtiin on helppoa tällöin.

3.4 Sisällön laitto

Sivuston sisällön hallinta oli keskeinen asia koko projektissa. HTML -pohjainen sivusto oli hankala ylläpitää ja päivittää. Tästä johtuen haluttiin WordPressillä toteutettu sivusto, joka on helpompi pitää ajan tasalla. WordPress-loopin käyttö tuli vastaan tässä vaiheessa, kun sivuston sisältöä alettiin työstämään. Kun sisältö syötetään post:n muodossa, siihen on mahdollista

kohdistaa toimintoja ja ominaisuuksia WordPressissä. Nimenomaan tällä tavoin oli projektissa tarkoitus toimia. Kuvassa 3. näkyy post:n muokkausikkuna WordPressin käyttöliittymässä.



Kuva 3. Post:n muokkausikkuna WordPressin koodinäkyvässä

Olemassa olevaa HTML-koodia oli mahdollista hyödyntää post:sa. Se ei välttämättä toiminut halutulla tavalla sellaisenaan, mutta sitä muokkaamalla sai samanlaisen lopputuloksen kuin alkuperäinen sisältö. WordPress näyttää postit kronologisessa järjestyksessä eli vanhin julkaisu alimpana ja uusin ylimpänä.

3.4.1 Latest.php

Latest.php sivun sisältö kopioitiin alkuperäisestä HTML-koodista, josta se täytyi ottaa erilleen. WordPressin tehokkuus tulee nimenomaan sisällönhallintaominaisuuksissa esille. Post:n käyttö oli nyt olennaista, jotta saatiin WordPressin hyöty käyttöön. Latest.php:sta tehtiin pohja, jolle jokaisesta kirjoituksesta tehtiin oma post, jotka sitten erotettiin vaakapalkilla toisistaan. Post:hin lisättiin kyseisiin kirjoituksiin liittyvät kuvat ja videot.

Default –teemassa oli toiminto, jolla post:n otsikkoa klikkaamalla aukeaa pelkästään kyseinen post uuteen ikkunaan headerin ja footerin kanssa, jos sellaiset ovat käytössä. Tämä ominaisuus oli hyödyllinen jättää käyttöön McCann –teemaan.

3.4.2 News.php

Tämä osio oli toteutukseltaan samanlainen kuin latest.php:ssa. Tässä elementissä näytetään uutiset viimeisin julkaisu ylimpänä.

3.4.3 People.php

Tämän otsikon mukaiseen kohtaan tuotiin olemassa olevalta sivustolta Flash-esitys, johon henkilötietojen päivitystoiminto olennaisesti liittyi.

3.4.4 Clients.php

Lista asiakkaista, jotka on jaettu palstoihin.

3.5 Toiminnallisuudet ja WordPressin muokkaus

WordPress on vapaan lähdekoodin sovellus, joten jokainen voi vapaasti muokata sovellusta parhaaksi katsomallaan tavallaan. Koodi on nähtävissä wp-admin kansion alla ja se on hyvin kommentoitua. WordPressin muokkaus oli järkevä tapa toteuttaa haluttu henkilötietojen päivitystoiminto.

3.5.1 Painike hallintavalikkoon

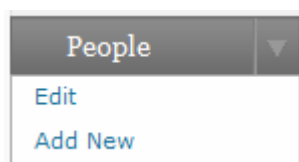
Wp-admin kansioista löytyy menu.php –tiedosto, joka sisältää koodin vasemmassa reunassa olevaan hallintavalikkoon. WordPressistä ei pääse näkemään eikä muokkaamaan näitä tiedostoja suoraan, vaan siihen tarvitaan ftp-yhteys palvelimeen. Haluttu tiedosto kopioidaan palvelimelta omalle koneelle, jossa sitä voi muokata jollakin editorilla.

Pienellä tarkastelulla koodista löytyi painikkeiden paikat, joista oli hyvä ottaa mallia uuden painikkeen luomiseen. Yksinkertaisinta on valita painike, jonka alapuolelle haluaa uuden painikkeen tulevan, ja kopioida olemassa olevan painikkeen koodi siihen. Ennen kuin painikkeen toiminnallisuutta ruvetaan muokkaamaan, kannattaa käydä katsomassa WordPressissä toiko tiedostoon tehty muutos toivottavan tuloksen. Tiedosto siirretään ftp-yhteydellä takaisin palvelimelle wp-admin kansioon. Tällöin kun selain päivitetään, WordPressin hallintavalikkoon on ilmestynyt uusi painike kuten kuvassa 4. näkyy.



Kuva 4 Hallintapaneeliin lisätty people -painike

Menu.php –tiedoston alussa on selitetty menu-painikkeiden koodin syntaksi. Siitä selviää mitä tietoja tarvitaan ja missä järjestyksessä ne syötetään. Uuden painikkeen nimeksi tuli people, koska se on sivuston alaosassa olevan osan nimi. People –painikkeen alle tarvittiin vielä linkit henkilön tietojen muokkaamiseen sekä uuden henkilön lisäämiseen. Linkeille tuli nimeksi edit ja add new. Kuvassa 5. näkyy people –painikkeen alla olevat linkit.



Kuva 5 People –painikkeen alla olevat edit ja add new -linkit

3.5.2 Henkilön muokkaus ja tallennus

Lisätyn people –painikkeen alle tehdyn edit –linkin tarkoitus on tuoda sivulle luettelo henkilöistä. Tuodusta listasta valitaan henkilö, jonka tietoja halutaan katsoa tai muokata, jolloin WordPress hakee valitun henkilön tiedot. Henkilöiden listaukseen tarvittiin oma sivu, josta tuli edit-people.php. Tälle sivulle listataan henkilöiden nimet, jotka haetaan palvelimelta ruudukko.xml –tiedostosta. Ruudukko.xml tiedosto oli olemassa entuudestaan ja se oli luotu sivustolla olevaa flashia varten, jossa henkilöiden tiedot näkyvät.

Tietojen hakemiseen XML:sta käytettiin `simplexml_load_file` -toimintoa, joka on php:ssa sisäänrakennettuna. Henkilötiedot `ruudukko.xml`:ssa on jaettu `<ruutu>` -tageihin, joten kun haluttiin saada henkilön nimi, käytettiin komentoa `$ruutu->nimi`. Henkilöiden etu- ja sukunimet olivat samassa `<nimi>` -tagissa, vain välilyönnillä erotettuna. Jotta url:ssa ei tulisi ongelmia välilyöntien kanssa, kannatti välilyönti korvata väliviivalla. Merkin korvaamiseen löytyi myös valmis toiminto `str_replace`. Luetelluista nimistä täytyi tehdä linkkejä, jotta kun valitun henkilön nimeä painaa, henkilön tiedot tulevat näkyviin.

```
<?php
$xml = simplexml_load_file("../ruudukko.xml");
foreach ($xml->ruutu as $ruutu) {
    $string = "$ruutu->nimi";
    $string = str_replace(" ", "-", $string);
    echo "<a href='edit-person.php?nimi=$string'>$ruutu->nimi</a><br />";
}
?>
```

Henkilöiden varsinaisten tietojen näyttämiseen tarvittiin vielä yksi sivu, josta tuli `edit-person.php`. Tälle sivulle tulee edellisen sivun listasta valitun henkilön tiedot. Tässä käytettiin php:n `GET` -metodia, jolla haettiin valitun henkilön nimi.

```
$nimi = $_GET['nimi'];
```

`edit-person.php`:lle tarvittiin myös xml -tiedoston lataaja, jossa käytettiin samaa `simplexml_load_file` -toimintoa kuin henkilöiden nimien listauksessakin. `Get`-metodin mukainen nimi asetettiin loaderiin ja sen perään lisättiin `.xml` -pääte.

```
$xml = simplexml_load_file("../xml/$nimi.xml");
```

Nimistä täytyi poistaa väliviiva sekä muuttaa isot kirjaimet pieniksi. Lisäksi piti ottaa huomioon joitakin erikoismerkkejä, sillä skandinaavisia merkkejä ei saanut käyttää, koska henkilöiden xml -tiedostojen nimet olivat muodostettu niin. Helpoin tapa tähän oli laittaa monta merkin korvausta peräkkäin.

```
$nimi = str_replace("-", "", $nimi);
$nimi = str_replace("ä", "a", $nimi);
$nimi = str_replace("Ä", "A", $nimi);
$nimi = str_replace("å", "a", $nimi);
```

```
$nimi = str_replace("Ä", "A", $nimi);  
$nimi = str_replace("ö", "o", $nimi);  
$nimi = str_replace("ö", "o", $nimi);  
$nimi = strtolower($nimi);
```

Merkistön kanssa voi tulla ongelmia, jos ei ole tiedossa mitä merkistöä mikäkin tiedosto käyttää. Merkistön tulisi olla kaikissa sama, jotta tiedostoja voi käyttää yhdessä. Tässä kohtaa ongelman aiheutti se, että xml-tiedosto ja php-tiedosto käyttivät erilaista merkistöä. Tuotu nimi-muuttuja täytyi muuttaa toiseen muotoon, jotta merkistöongelma poistui.

```
$nimi = utf8_decode($ruutu->nimi);  
$nimi = urlencode($nimi);
```

Henkilön tiedoista voi muokata kaikkea muuta, paitsi ei nimeä. Tätä varten lisättiin nimen kohdalle uusi parametri `readonly = "readonly"`. Tämä siitä johtuen, että nimen mukainen tieto haetaan `ruudukko.xml`:sta sekä jokaisen henkilökohtaisesta XML –tiedostosta. Joten jos henkilön nimeä halutaan jostain syystä muuttaa, täytyy se tehdä luomalla uusi henkilö, tai sitten muokata `ruudukko.xml`:a ja vastaavasti henkilökohtaista XML:a suoraan palvelimelta.

Henkilötietoja varten tehtiin lomake johon tuli tekstikenttä jokaiselle syötettävälle tiedolle. Nämä kentät nimettiin yksilöllisesti ja kuvaamaan siihen tulevaa tietoa. Näihin kenttiin haettiin tiedot `ruudukko.xml` –tiedostosta.

```
<Form Name = "info" Method = "POST" ACTION = "edit-person.php?nimi=<?php echo  
($_GET['nimi']); ?>">  
foreach ($xml->nimi as $nimi) {  
echo ("<td class='formisolu'>Name: </td><br /><td><input name='nimi' type = 'text' size = '40'  
value = '$nimi' readonly = 'readonly'></td><br />");  
}
```

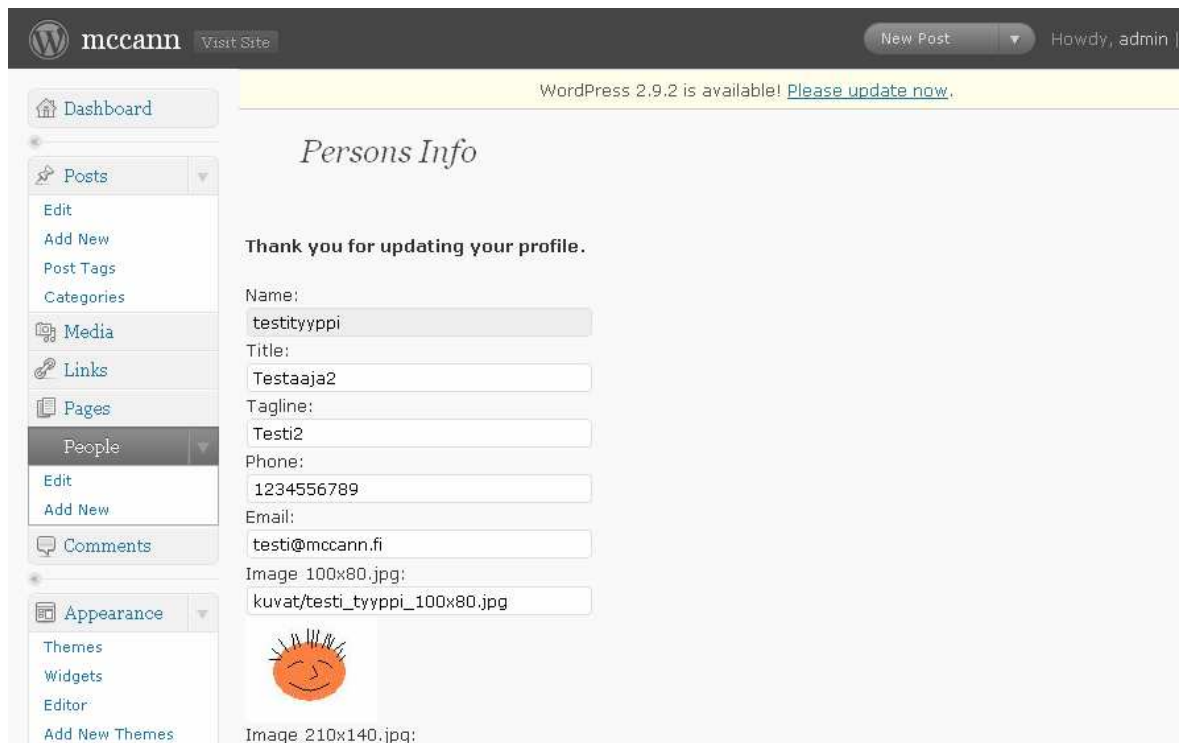
Haettuja tietoja voi muokata ja tallentaa muutokset. Lomakkeen `action` –parametrin arvoksi määritettiin `edit-person.php` sekä haettiin käsittelyssä oleva nimi. Eli kun tallennusta varten luotua `save` –painiketta painaa, sama sivu johon tiedot syötetään, latautuu uudestaan ja antaa ilmoituksen tallennetuista muutoksista. Tämän toteuttamiseen tarvittiin lomakkeeseen kenttä joka tunnistaa tehdyn muutoksen. Tämän kentän tyypiksi laitettiin `hidden` eli se ei näy käyttäjälle, vaan toimii ainostaan tallennuksen apuna.

```
if ($_POST['op'] == "edit")
echo ("<input type='hidden' name='op' value='edit' />");
```

Koska tietojen muutokset tehdään samalle lomakkeelle ja samoihin kenttiin, kuin alkuperäiset tiedot haetaan, tarvittiin jokin joka ilmoittaa käyttäjälle, että tallennus on tehty. Joten kun nimen mukaiseen tiedostoon tallennetaan tietoa xml –muodossa, tulee ruudulle sivun uudelleen latautumisen yhteydessä ilmoitus tallennuksesta käyttäjälle.

```
$xml->asXML("../xml/$nimi.xml");
echo "<b>Thank you for updating your profile.</b><br /><br />";
```

Alla olevassa kuvassa 6. on näkymä henkilötietojen muokkausikkunasta. Tiedot tuodaan xml-tiedostosta tekstikenttiin, joista ne myös haetaan xml-tiedostoon tallennusta varten.



Kuva 6. Henkilötietojen muokkaustoimintoon tehty lomake

3.5.3 Henkilön lisäys

Henkilötietojen muokkaamisen lisäksi haluttiin mahdollisuus lisätä uusia henkilöitä. Toiminnallisesti tämä on hyvinkin samantyyppinen tapahtuma kuin muokkaus- ja tallennustoiminto, mutta nyt täytyi saada luotua uusi henkilö ruudukko.xml:an sekä täysin uusi henkilökohtainen XML –tiedosto.

Henkilön lisäykseen tarvittiin uusi tiedosto new-person.php. Tällä sivulla käytettiin pohjana samaa lomaketta, kuin henkilötietojen muokkauksessa. Tällä kertaa myös henkilön nimi – kenttään oli mahdollista kirjoittaa, joten poistettiin nimen kohdalta readonly –parametri. Suurin muutos henkilötietojen muokkaussivuun oli tehdä toiminto, joka luo syötetyn tiedon mukaisen xml-tiedoston, ja joka saa nimekseen syötetyn nimen arvon ja lisää perään vielä .xml –päätteen.

Eli käyttäjän antamista tiedoista täytyi saada tehtyä kaksi erillistä xml-tiedostoa.

Ruudukko.xml:an, josta kaikkien henkilöiden nimet listataan, tarvittiin vain muokkausta ja henkilökohtainen xml-tiedosto täytyi luoda kokonaan.

Kun uuden käyttäjän tiedot on syötetty kenttiin, jotka tulevat ruudukko.xml:an, haetaan tarvittavista kentistä tiedot myös henkilökohtaisen xml-tiedoston luomiseen. Käyttäjän painaessa save –painiketta xml-tiedostot päivittyvät ja sama sivu latautuu uudestaan, näyttäen ilmoituksen profiilin luomisesta sekä annetut tiedot.

```
$xml = simplexml_load_file("../ruudukko.xml");
```

```
$xmledosto1 = $_POST['nimi'.xml];
```

```
$sxe = new SimpleXMLElement($xml->asXML());
```

```
$ruutu = $sxe->addChild("ruutu");
```

```
$ruutu->addChild("leveys", "1");
```

```
$ruutu->addChild("korkeus", "1");
```

```
$ruutu->addChild("nimi", $_POST['nimi']);
```

```
$ruutu->addChild("xmledosto", $xmledosto1);
```

```
$sxe->asXML("../ruudukko.xml");
```

```
echo "<b>Thank you for creating a new profile.</b><br /><br />";
```

3.6 Sivuston testaus selaimilla ja iPhonella

Sivuston toimivuutta tarkasteltiin koko projektin ajan Internet Explorer 7:lla, FireFox 3.6:lla sekä Safarilla. On selvää että eri selaimet näyttävät sisältöä hieman eritavalla. Testaamista on hyvä tehdä ainakin kahdella sellaisella selaimella, joista tietää että niitä tullaan sivuston katselussa käyttämään. McCann:n työntekijöistä suurimmalla osalla on käytössä Mac-tietokone, joten sivuston tuli toimia ainakin kolmessa eri selaimessa. Pc-käyttäjät käyttävät yleensä joko Internet Exploreria tai FireFoxia ja Mac-käyttäjät Safaria tai FireFoxia.

Sivuston testaamista iPhonella ei juuri ehditty toteuttamaan. Sivusto toimii kyllä jotenkin, mutta iPhonessa ei voida näyttää mm. flash sisältöä, joten sivun toiminta ei ole täydellistä.

4 Tulokset

Opinnäytetyön tuloksena syntyi tutkimus WordPress julkaisujärjestelmän käytöstä websivujen julkaisussa sekä WordPress julkaisujärjestelmällä toteutettu sivusto McCann Wordgroup Helsinki Oy:lle. Tuloksena on: selvitys siitä mikä on WordPress, miten sitä käytetään ja miten se soveltuu websivujen julkaisuun. Teoriatietoa, jota tutkimuksessa käytettiin, hyödynnettiin käytännön osassa ja sivuston toteutuksessa.

Tutkimuksen teoriaosan ideana oli käsitellä yleisesti WordPressin käyttöä websivujen julkaisussa. Selvitettiin keskeisiä termejä sekä käsiteltiin WordPressin asennusta ja käyttämiseen liittyviä yleisiä asioita. Kerrottiin teemasta ja teeman muokkaamisesta sekä template:sta ja template-rakenteesta, loop-toiminnosta ja post:n käytöstä.

Case-esimerkin lähtökohtana oli muuttaa olemassa oleva McCann:in HTML –pohjainen sivusto muotoon, jossa sen ylläpitäminen ja päivittäminen on helpompaa. Sivusto rakennettiin WordPressiin php-elementeistä. Sivuston rakenteen toteuttamiseen tarvittiin yksinkertainen template rakenne. Koska sivu rakentuu php-elementeistä, niiden keskinäinen hallinta sekä sisällön hallinta on tehokasta ja ylläpito helppoa.

WordPress asennettiin yrityksen palvelimelle, josta kaikki yrityksen työntekijät pääsevät sitä käyttämään. Työstövaiheessa sivusto kaikkine toiminnallisuuksineen ja sisältöineen oli palvelimelle tehdyssä alihakemistossa, jotta testausta saatiin tehtyä ilman että se aiheutti häiriötä ja katkoksia McCann:n olemassa olevaan sivustoon. Kun WordPressillä luotu uusi sivusto valmistui, se siirrettiin palvelimen juurihakemistoon. WordPressin toimintoja pääsee käyttämään kirjautumalla sisään admin-tunnuksilla, jotka luotiin sovelluksen asennuksen yhteydessä.

Koska WordPress on avoimen lähdekoodin sovellus, sen kehittäminen ja muokkaaminen on mahdollista. WordPress sovellusta ja käyttöliittymää edelleen kehitettiin palvelemaan yrityksen työntekijöiden tarvetta muokata omia henkilötietojaan. Hallintapaneeliin lisättiin painike, josta henkilöprofiilia on mahdollista tarkastella, muokata tietoja sekä tallentaa muokattu profiili.

Vaikka WordPress tarvitsee periaatteessa asentuaakseen ja toimiakseen tietokannan, tässä projektissa tietokantaa ei tarvinnut käyttää vähäisen informaatiomäärän vuoksi.

Henkilötietojen päivitystoiminto liittyi sivuilla olevaan swf-esitykseen, johon oli tehty jo aikaisemmin xml-tiedostot. Xml-tiedostoihin oli tallennettu kaikki henkilökuntaan liittyvät tiedot, joita swf-esityksessä näytetään. Xml-tiedostot oli rakennettu niin, että yhdessä tiedostossa oli kaikkien työntekijöiden nimi, sekä nimeen liittyvä henkilökohtaisen xml-tiedoston nimi ja hakemistopolku. Jokaiselle työntekijällä oli siis oma henkilökohtainen xml-tiedosto, johon oli tallennettu kyseisen henkilön tiedot. Kyseiset xml-tiedostot eivät sisältäneet mitään arkaluontoista tai salaista tietoa, vaan pelkästään tietoja, joita swf-esityksessä halutaan näyttää.

Projektissa päätettiin käyttää olemassa olevia xml-tiedostoja tiedon hakemiseen sekä tallentamiseen. Jos projektissa olisi käytetty tietokantaa, olisi tallennetut tiedot pitänyt muuttaa kuitenkin xml-muotoiseksi tietokannasta hakemisen jälkeen, sillä sivustolla oleva swf-esitys tarvitsee henkilötiedot xml-muodossa.

Henkilötietojen päivitystoimintoa varten tehtiin HTML-lomake, johon tuli tekstikentät kullekin tietotarpeelle. Tiedot haetaan sekä tallennetaan xml-tiedostoon. Ensin haetaan kaikkien työntekijöiden nimet xml-tiedostosta sivulle, josta valitaan henkilö. Tällöin aukeaa uusi sivu johon haetaan käyttäjätietoja henkilön nimen mukaan henkilökohtaisesta xml-tiedostosta lomakkeen yksilöllisesti nimettyihin tekstikenttiin. Tekstikenttien sisältöä on mahdollista muokata ja tallentaa sitten niihin muokatut tiedot.

Uuden henkilön lisäystoiminnossa käytettiin samaa HTML-lomaketta, joka tehtiin henkilötietojen päivitystoimintoa varten. Uuden henkilön lisäystoiminnossa päivitetään kaikkien henkilöiden tietoja sisältävän xml-tiedoston sisältöä sekä luodaan uusi xml-tiedosto lisätylle henkilölle ja nimetään luotu tiedosto henkilön nimen mukaan sekä poistetaan erikoismerkit ja välilyönti. Kun tallennus on tehty, annetaan käyttäjälle siitä ilmoitus. Lomakkeen tekstikenttiin ei tehty tarkistuksia siitä johtuen, että kyseessä oli jokaisen työntekijän itsensä syöttämää tietoa itsestään. Mahdollinen virheellinen syöte lomakkeeseen huomataan ja korjataan nopeasti.

Tuloksena syntynyttä sivustoa on mahdollista kehittää eteenpäin. Sivustolle on helppo lisätä toiminnallisuutta plugin:en muodossa tai sitten kehittämällä itse sovellusta. Kun sivusto muodostuu elementeistä joihin on kohdistettu toimintoja, kokonaisuuden hallinta on järjestelmällisempää. Päivitys- ja ylläpitotoiminnot ovat huomattavasti helpommat toteuttaa ja niiden tekemiseen ei vaadita välttämättä muuta kuin tekstin muokkaustaitoja.

5 Pohdinta

Opinnäytetyönä tehdystä tutkimuksesta sekä siihen liittyvästä toteutetusta sivustosta päätellen voidaan todeta WordPressin soveltuvan hyvin websivujen julkaisuun. Vaikka WordPress onkin alun perin kehitetty blogien tekemiseen ja ylläpitoon, se soveltuu websivujen julkaisuun todella hyvin sisällönhallintaominaisuuksiensa vuoksi. Tutkimuksen teoriaosassa eri lähteistä selvinneet asiat toteutuivat hyvin käytännön työssä. Aiheesta ei ollut helppo löytää kirjallisuutta, sillä WordPress on julkaisujärjestelmänä melko uusi.

WordPress-sovellus kasvaa ilmiönä ja saa uusia puolia, kun sitä kehitetään WordPress-yhteisön toimesta jatkuvasti. WordPress-yhteisön omat websivut ovat sovelluskehittäjälle korvaamattoman tärkeitä, sillä sivuilta saa tietoa ja oman tietämyksen sekä mielipiteiden jakaminen on mahdollista. WordPressin omat sivut toimivat hyvänä esimerkkinä sivustosta, jotka on toteutettu blogimaiseen muotoon.

Sivusto, joka antaa käyttäjälle mahdollisuuden kertoa omia kokemuksiaan aiheesta tai jakaa mielipiteitään, on yksinkertaisen nerokas. Idea on yksikertainen, mutta sitä ei ole käytetty vielä kovin laajasti yritysmaailmassa. Monilla yrityksillä on websivut, joissa esitellään tuotteita ja palveluja, mutta vuorovaikutus tärkeimpään ryhmään eli asiakkaaseen on jätetty pienemmälle huomiolle tai pois kokonaan. Joillakin sivuilla on mahdollista antaa palautetta, joka lähtee yritykselle sähköpostin muodossa. Tällöin yritys, joko reagoi viestiin tai on reagoimatta. Yleinen ajatusten vaihto ja sitä kautta saatu palaute, joka esitetään kaikille, on mielekkäämpää. Asiakas voi tällöin kokea voivansa vaikuttaa asioihin ja yritys voi kokea toimivansa asiakkaiden haluamalla tavalla ja kehittää toimintaa yhä edelleen. Eri asteisten ideoiden tuominen esille voi olla hyvä jo alkuvaiheessa, ennen kuin tuotantovaiheen jälkeen huomataan, että tuotteelle ei olekaan kysyntää.

WordPress sovelluksena on todella monipuolinen kokonaisuus, joka on selvinnyt tehdyn tutkimuksen myötä. Tässä projektissa käsiteltiin vain osaa WordPressin toiminnallisuuksista ja mahdollisuuksista. Kaiken hyödyn irti saaminen WordPressista vaatii enemmän paneutumista aiheeseen. Kun ensimmäinen projekti WordPressillä on tehty, seuraavan lähestyminen on aivan toisenlaista. Sovelluskehitys on keskeisessä roolissa, kun tehdään toimintoja yrityksen tarpeisiin. Työn edetessä on tullut mieleen useita erilaisia sovelluskohteita, joiden

toteuttamiseen WordPressiä voisi käyttää. WordPress toimii hyvin ja vakaasti, sekä lisäksi sen kehittäminen on mielenkiintoista ja hyvän dokumentoinnin ansiosta mahdollista.

Vaikka aikaisempaa kokemusta php-ohjelmoinnista tai WordPressistä ennen tätä projektia ei ollutkaan, sekä tehty tutkimus että siihen liittyvä toteutettu sivusto toteutuivat lähes suunnitellun mukaisesti. Php-ohjelmoinnin periaate tuli selville hyvin pian tarkastelemalla WordPressin Default-teeman templateja. Php-ohjelmointi osoittautui mielenkiintoiseksi tavaksi tehdä webohjelmointia. Php:n avulla toiminnallisuuden lisääminen html-pohjaiseen sivustoon käy helposti. Erilaisten ideoiden toteutus, joita aikaisemmin ei ole osannut toteuttaa, onkin nyt mahdollista.

Aikaisempi kokemus blogityökaluista olisi ollut tutkimusta tehdessä hyödyksi, jotta olisi ollut selkeämpi käsitys blogin toiminta- ja toteutusperiaatteesta. Tutkimusta tehtäessä eri lähteistä tuli esille asioita blogien käyttömahdollisuuksista, jotka muuttivat suhtautumista koko blogeihin ilmiönä. Aikaisempi käsitys perinteisestä päiväkirjamallisesta blogista, koki suuren muutoksen kohti toimivaa informaation jakamiskanavaa, jonka mahdollisuudet ovat blogin ylläpitäjän käsissä.

Tutkimuksen käytännön osassa WordPress-sovellusta kehitettiin yrityksen tarpeiden mukaiseksi. Sovelluksen kehitykseen liittyvät asiat tulevat olemaan tutkimuksen jälkeenkkin merkittävässä osassa ja olivat oppimiskokemuksena suurta antia. Halu kehittää ja luoda uutta toimivampaa kokonaisuutta on täten mahdollista. Henkilötietojen päivitystoiminnon olisi ehkä voinut toteuttaa myös plugin:na, mutta WordPressin muokkaus tuli kuitenkin toteutustavaksi. Plugin:en tekeminen vaatii edistyneitä ohjelmointitaitoja, joten tämän projektin aikana resurssit eivät riittäneet plugin:en tekemisen opetteluun ja tuotantoon. Plugin:en maailmaan tullaan tutustumaan tarkemmin myöhemmin.

Testausta tehtiin käytännön osassa selaimilla lähes jatkuvasti projektin aikana. IPhonella testaaminen jäi pienemmälle huomiolle, koska muut projektin osat veivät resursseja enemmän kuin oli odotettu. Websivustojen toiminta mobiililaitteilla tulee olemaan koko ajan tärkeämpää, sillä laitteet yleistyvät ja halpenevat. Yhä enemmän ihmiset käyttävät internettiä puhelimen kautta, joten yritysten täytyy huomioida tämä myös omien palveluiden sekä tuotteiden esittelyssä ja kehittäessä. Sivustojen täytyy pystyä vastaamaan teknologian vaatimuksia, jotta pystytään huomioimaan erilaiset käyttäjät ja käyttäjäryhmät.

Yritysten websivuilla tietojen päivitys on olennaisen tärkeä tekijä, koska websivut ovat merkittävä osa sekä ulkoista että sisäistä markkinointia. Joillakin yrityksillä on tarkoitukseen tehty, räätälöity sovellus, joilla sivuja voi päivittää. Toisilla yrityksellä kaikki päivitykset tehdään jonkin ulkopuolisen toimesta ja jotkut eivät päivitä sivujaan ollenkaan. WordPressillä on helppo luoda päivitysjärjestelmä, joka hoitaa tarvittavat sisällön muotoilut ja asetellut, ja jonka käyttö ei vaadi normaalien tekstinkäsittelytaitojen lisäksi muuta osaamista. Joten päivittäjän tarvitsee vain syöttää sisältö ja painaa päivityspainiketta, jolloin WordPress hoitaa loput.

Sosiaalisen median kasvaessa ja kehittyessä tarvetta sisällönhallintajärjestelmille on. Koska tiedon määrä on suurta, sen hallinta on vaikeaa ja sivustoista tulee sekavia viidakoita, joista ei löydä edes ulos sulkematta selainta. WordPressin avulla luodulla sivustolla voi hallita suurempaakin tietomäärää, kun tiedot tallennetaan tietokantaan sekä haetaan ja esitetään käyttäjälle tarvittaessa, vain silloin kun käyttäjä niin haluaa. Jos tallennettavissa tiedoissa on jotain sellaista, jota ei haluta yleiseen tietoon, on tietojen tallentaminen tietokantaan turvallisempaa.

Tutkimuksen myötä on herännyt ajatuksia asioista, joita voisi tutkia lisää. Jatkotutkimuksen voisi tehdä tutkimalla esim. plugin:en rakentamista tai teeman muokkaamista tarkemmalla tasolla. WordPressillä tehdyt sivustot voivat olla hyvinkin luovasti ja nerokkaasti toteutettuja, vaikka toiminnan taustalla pyörii sama toiminnallinen perusidea. Pluginien avulla toiminnallisuutta voisi laajentaa syndication-tyyppisesti, seuraamaan muiden sivustojen tai blogien tapahtumia sekä ilmoittamaan käyttäjälle tämän määrittelemistä asioista ne, jotka eniten kiinnostavat. Informaatio tulisi suoraan käyttäjälle, eikä sitä tarvitsisi lähteä erikseen etsimään. Käyttäjä voisi näin jakaa saadun tiedon edelleen muille feeds-tyyppisellä toiminnolla.

Blogeja käytetään ajatusten vaihdon ja mielipiteiden jakokanavana. Kun jonkin asian toimintamalli on riittävän helppo, sen käytöstä tulee helpommin suosittua ja sana leviää laajalle nopeasti. Twitter teki oman versionsa blogimallista ja siitä tuli suosittu todella nopeasti. Kerro asiasi lyhyesti ja ytimekkäästi ja jaa se muille. Idea oli yksinkertainen ja siksi sai nopeasti lisää käyttäjiä keskuuteensa. Tietysti herää kysymys, kuka näitä kaikkia viestejä lukee ja kenelle ne on oikein tarkoitettu? Viestit eivät olekaan tarkoitus olla välttämättä kaikille suunnattuja, vaan ne liittyvät ehkä johonkin suurempaan kokonaisuuteen jonka käsittely on aloitettu jo aikaisemmin.

Blogia voidaan pitää mm. projektinhallintavälineenä, jossa jaetaan jonkin tietyn ryhmän sisällä tärkeää tietoa. Ryhmän ulkopuolinen käyttäjä ei välttämättä edes ymmärtäisi viesteistä mitään, koska viestit ovat suunnattu vain tietyille kohderyhmälle. Viestin suuntaamista tietylle

kohderyhmälle hyödynnetään mainonnassa, mutta esim. tv-mainoskampanjalla on haluttujen kontaktien lisäksi paljon hukkakontakteja. Websivuilla olevaa mainontaa voidaan kohdistaa paremmin tietylle käyttäjäryhmälle ja jopa näyttää mainokset käyttäjäprofiiliin annettujen tietojen perusteella. Tällöin mainonta voidaan kohdistaa vain ja ainoastaan sille kohderyhmälle, jolle se on tarkoitettu. Blogimainontaa tehdään myös, mutta paljon potentiaalia on vielä käyttämättä. Vuorovaikutus mainostajan ja asiakkaan välillä jää pois kokonaan perinteisessä mainonnassa, johon blogissa ja blogimainonnassa voidaan nimenomaan vaikuttaa. Tuodaan yritys lähemmäs asiakasta ja annetaan asiakkaalle sellaista tietoa, johon kaivataan vastauksia. Yritys voi laittaa lisäksi omaan blogiin tai sivustoonsa blogroll-listaan linkkejä, joiden avulla voidaan viestittää käyttäjille asioita, jotka vaikuttavat myös mielikuvan luomiseen yrityksen toiminnasta ja arvoista.

Lähteet

Blakeley Silver, T. 2008. WordPress Theme Design. Packt Publishing.

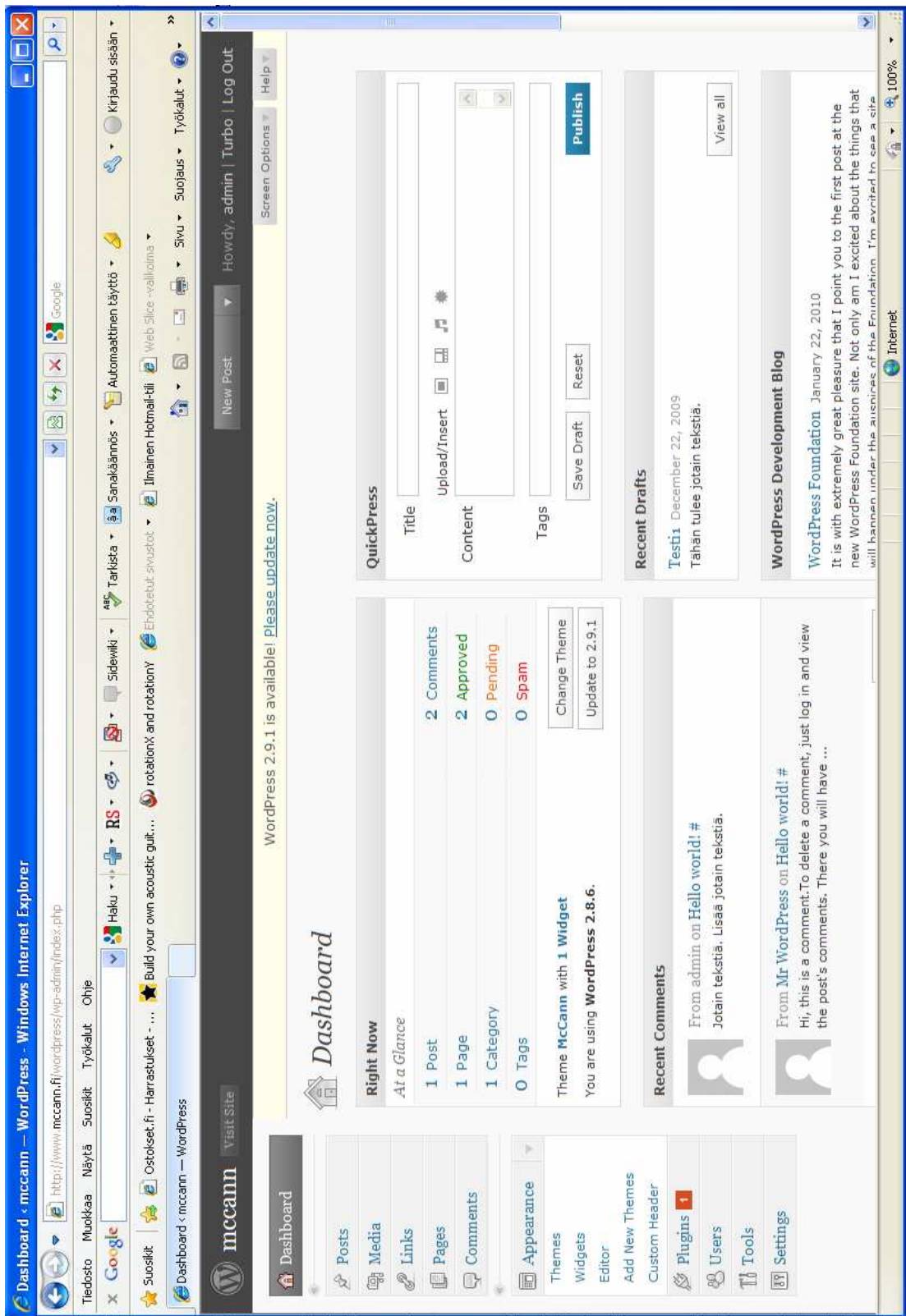
Castro, E. 2005. Publishing a Blog with Blogger. Peachpit Press. Berkeley, CA.

Smith, J. & Llinares, J. 2009. Dominate your market with Twitter. The Infinite Ideas Limited. Oxford, UK.

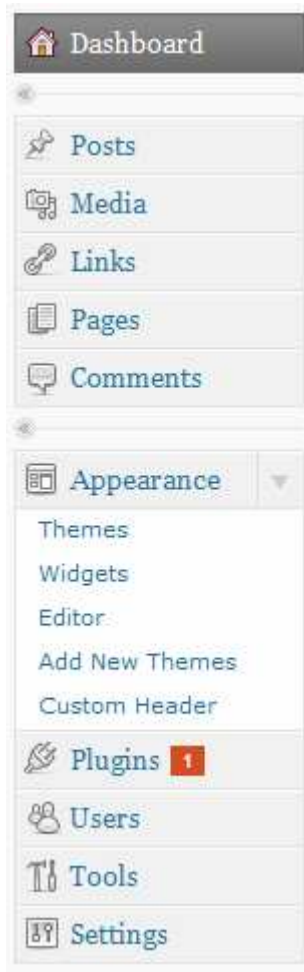
Vejde, O. & Gustafsson, B. 2005. En första blogg för den orädda nybörjaren med Blogger. Milsbo Tryckeri.

WordPress 2010. Luettavissa: <http://www.WordPress.org>

Liite 1. WordPressin käyttöliittymä




Liite 2. Dashboardin hallintapaneeli




Liite 3. McCann websivun layout

McCANN



MCCANN WORLDWIDE HELSINKI OY | PLEIN ROBERTINKATU 1-3 00100 HELSINKI | TEL. 095 4240111 | EMAIL: INFO@MCCANN.FI | PETER@LILINDO@MCCANN.FI



Franck Provost on ymmen Suomessa!

Fanita Franchia – ja lähdä kampaajalle Pariisiin

www.franckprovostexpert.fi


McCann – Vuoden mainostointistoketju Euroopassa


advertising

Kilpailu kansainvälisten kampanjoiden tekijöiden parissa on ollut niin kovan tasoinen, että vuoden 2008 voittaja on joutunut jättämään paikkansa jäljempänä sijalle. Tämä vuonna kilpailu oli tunteiden ja viisujen taistelua, sillä kahdella parhaalla oli joutunut jättämään paikkansa jäljempänä sijalle.

McCann on ollut joka vuosi mukana kilpailussa, vaikka viimeisin voitto on kuulunut muulle.

Green on pitänyt kiikkapöytänsä ja suunta peräkkäin, mutta tänä vuonna McCann on tehnyt uudenlaisia tunteita. Vuotta McCannille toi erityisesti McCannin tuntuu -tuotteille tekemä taideteoskampanja.





advertising

Sergel

T. Oskari Tindari

Matkaohjelmat

Gator

HKT

Nestlé

II

Viking Line

VAASAV

Opel

Marrasinen

Es-sun Mobile

Raché

Mastercard

SER

Lairas

Orosu

Microsoft

McCANN