

KYMENLAAKSON AMMATTIKORKEAKOULU

Viestinnän koulutusohjelma/ Verkkomedian suuntautumisvaihtoehto

Outi Heinonen

VERKKOJULKAISUSIVUSTON SUUNNITTELU JA TOTEUTUS
WORDPRESS-JULKAISUJÄRJESTELMÄÄ HYÖDYNTÄEN

Opinnäytetyö 2010

TIIVISTELMÄ

KYMENLAAKSON AMMATTIKORKEAKOULU

Viestintä

HEINONEN, OUTI	Verkkojulkaisusivuston suunnittelu ja toteutus WordPress-julkaisujärjestelmää hyödyntäen
Opinnäytetyö	58 sivua + 4 liitesivua
Työn ohjaaja	Marko Siitonen
Toimeksiantaja	Non-Stop Studiot OY/AB
Huhtikuu 2010	
Avainsanat	WordPress, julkaisujärjestelmät, sisällönhallinta, verkkojulkaisut

Verkkosivujen ylläpidon helpottamiseksi yhä useammassa yrityksissä on siirrytty sisällönhallinta- tai julkaisujärjestelmien käyttöön. Järjestelmissä sisällöt ovat erotettu sivuston rakenteesta ja toiminnallisuudesta, jolloin sisällön päivittäminen ei vaadi peruskäyttäjältä teknistä osaamista. Järjestelmät tukevat useita käyttäjiä ja käyttöoikeuksia, jolloin verkkosivuja voi hallita useampi organisaation jäsen yhtä aikaa.

Opinnäytetyössä käsitellään aluksi julkaisujärjestelmiä yleiseltä kannalta, jonka jälkeen siirrytään tarkastelemaan aihetta WordPress-julkaisujärjestelmän näkökulmasta. Tutkimuksen tarkoituksena oli tarkastella WordPress-julkaisujärjestelmän toimintaa, teknisiä vaatimuksia, ominaisuuksia ja mahdollisuuksia tavalla, joka hyödyttäisi opinnäytetyön tuotannon teknistä toteutusta. Lisäksi tutkimusta voidaan hyödyntää arvioidessa WordPressin soveltuvuutta muihin verkkosivuprojekteihin.

Opinnäytetyön tutkimuksen pohjalta suunniteltiin ja toteutettiin Non-Stop Studiot Oy:lle ViewMe-verkkojulkaisupalvelun markkinointisivusto avoimen lähdekoodin WordPress-julkaisujärjestelmällä. Myös erillisellä sivustolla toimivan verkkojulkaisusovelluksen ulkoasu tuli päivittää markkinointisivuston kanssa yhteneväiseksi, jotta nämä kaksi sivustokokonaisuutta voitiin yhdistää yhdeksi kokonaisuudeksi. Verkkojulkaisusovelluksella tarkoitetaan palvelua, jonka kautta asiakas toimittaa verkkokirjaksi, -lehdeksi tai -esitteeksi julkaistavan PDF-tiedoston yrityksen käännettäväksi. Interaktiivinen flash-julkaisu on animoitu siten, että sivuja voidaan kääntää oikean paperiversion tapaan. Julkaisua voidaan käyttää ViewMeOnline-sivutilasta tai siirtää asiakkaan omalle palvelimelle. Vaikka opinnäytetyö painottui tekniseen toteutukseen, kuului tuotantoon myös logon, visuaalisen ilmeen ja käyttöliittymän suunnittelu. Opinnäytetyössä pohditaan teknisen toteutuksen, visuaalisen suunnittelun ja käyttöliittymäsuunnittelun lisäksi sivuston viestintää, sen merkitystä, kehitysideoita ja tulevaisuutta nyky-yhteiskunnassa.

Opinnäytetyön tavoitteet saavutettiin onnistuneesti. WordPress-järjestelmän käyttötarkoituksista ja mahdollisuuksista saatiin tutkimuksen kautta runsaasti tietoa, jota voitiin soveltaa opinnäytetyön teknisessä toteutuksessa. Sivuston ilmeestä saatiin tavoitteiden mukaisesti pelkistetty, mutta tyylikäs. Logosta tuli viestinnällisesti onnistunut ja toimiva. Käyttöliittymä onnistuttiin rakentamaan hyvin ja mahdolliset ongelmat saatiin ratkottua yli odotusten.

ABSTRACT

KYMENLAAKSON AMMATTIKORKEAKOULU

University of Applied Sciences

Media Communication

HEINONEN, OUTI

Website design and implementation for a digital publishing platform using WordPress publishing system

Bachelor's Thesis

58 pages + 4 pages of appendices

Supervisor

Marko Siitonen

Commissioned by

Non-Stop Studiot Ltd.

April 2010

Keywords

WordPress, Publishing platforms, Content Management, Web publishing

Many companies have started to use publishing platforms or content management systems (CMS) to make the updating of their websites easier. Content Management Systems separate the content from the code structure. Ordinary users no longer need the technical coding skills to add content to their websites. CMS also supports multiple users and authors allowing several people to contribute and update the websites at the same time.

The first part of this thesis is a general overview of CMS platforms. The second part examines, in detail, the WordPress-publishing platform: how it works as well as its specific functions, requirements and features. These research results can, however, even be implemented in other similar WordPress (CMS) website projects.

On the basis of the examination, a website marketing ViewMe was designed for Non-Stop Studios Ltd. ViewMe is a digital interactive publishing platform made partly with the Open Source WordPress (CMS) platform. The visual appearance of the digital interactive publishing system also had to be updated. The challenge was to fuse two platforms, WordPress (CMS), PHP and MySQL database into one functioning web system. This interactive digital publishing system is a service whereby customers can upload their books, magazines and brochures in the PDF format. The result is an interactive Flash publication that can be hosted for the client on the ViewMeOnline or on their own web server. One of the popular and appealing features is that the pages are animated to turn like a real book. Although the thesis is primarily based on the technical implementation, the project also called for the design and production of the ViewMe logo, graphic and typographic design elements and the user interface design. The thesis discusses the technical implementation, graphic design and user interface design in addition to the site's communications, the importance of development ideas and the future of today's society.

The primary goals for this project were met successfully. The research provided plenty of vital information about WordPress and the possibilities of implementing such publishing platforms in the future. The idea was to create a simple looking but elegant web site and user interface. The logo was effective and its message was clear. The user interface works efficiently and is clear and easy to use.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO	6
2	TAUSTA	6
3	JULKAISUJÄRJESTELMÄ	7
3.1	Julkaisujärjestelmän määritelmä	7
3.2	Julkaisujärjestelmän käyttötarkoitukset	8
3.3	Hyödyt ja haitat	9
3.4	Avoimen lähdekoodin julkaisujärjestelmät	11
3.5	Julkaisujärjestelmän tekninen toiminta ja vaatimukset	11
3.5.1	WWW-palvelin ja -ohjelmisto	12
3.5.2	PHP	12
3.5.3	MySQL	13
4	WORDPRESS JULKAISUJÄRJESTELMÄ	14
4.1	Perusominaisuudet ja -toiminnot	14
4.2	Asentaminen	16
4.3	Hallintasivu	17
4.4	Asetukset	18
4.5	Käyttäjähallinta ja käyttöoikeudet	23
4.6	Teemat	25
4.7	Lisäosat	27
4.8	Tietoturva	28
5	VERKKOJULKAISUSIVUSTON SUUNNITTELU	31
5.1	Suunnittelun lähtökohdat	31
5.2	Logon ja sivuston viestinnällinen tarkoitus	32
5.3	Logon suunnittelu	33
5.4	Graafisen käyttöliittymän suunnittelu	36
6	VERKKOJULKAISUSIVUSTON TEKNINEN TOTEUTUS	40
6.1	Markkinointisivuston tekninen toteutus WordPressillä	40

6.2 Sovellussivuston ulkoasun päivittäminen	44
6.3 Sivustojen yhdistäminen yhdeksi kokonaisuudeksi	46
6.4 WordPress-järjestelmään lisäosien valinta	49
6.5 WordPress-järjestelmän asetusten määrittely	51
6.6 WordPressin tietoturvan parantaminen	52
6.7 Teknisen toteutuksen ongelmat	53
7 YHTEENVETO	54
LÄHTEET	55
LIITTEET	
Liite 1. Logokuvion kehitys	
Liite 2. Logokuvion valmiiseen logoon	
Liite 3. Entinen ViewMe:n etusivu ja ensimmäinen protosivu	
Liite 4. Toinen protosivu ja uusi ViewMe:n etusivu	

1 JOHDANTO

Vuoden 2009 lopulla otin sähköpostitse yhteyttä useisiin yrityksiin löytääkseni opin-
näytetyön aiheen. Tavoitteenani oli löytää aihe verkkosivuston suunnittelun ja toteu-
tuksen parista, joka toteutettaisiin julkaisujärjestelmää hyödyntäen. Sopiva työ löytyi
Kotkasta Non-Stop Studiot Oy:lta.

Opinnäytetyöni lähtökohtana oli suunnitella ja toteuttaa markkinointiin soveltuva
verkkojulkaisupalvelusivusto avoimen lähdekoodin WordPress-julkaisujärjestelmällä,
ja yhdistää jo olemassa ollut verkkojulkaisusovellussivusto palvelusivustoon. Työhöni
kuului myös palvelun logon suunnittelu ja automaattisten sähköpostiviestien ulkoasun
päivittäminen.

Opinnäytetyöraportissani tarkastelen julkaisujärjestelmiä yleisellä tasolla ja syvennyn
WordPressin toiminnallisuuteen sekä sen käyttöön toteutuksessa. Pohdin myös sivus-
ton käytettävyyttä ja graafista ilmettä.

Opinnäytetyö on toteutettu Kouvolassa ja Kotkassa joulukuun 2009 – huhtikuun 2010
välisenä aikana.

2 TAUSTA

Opinnäytetyöni aiheen sain Non-Stop Studiot Oy:lta. Yritys on vuonna 1992 perustet-
tu pieni kotkalainen julkaisutoimisto, jonka omistajana ja yrittäjänä toimii Paul
Hitchman. Yritys on erikoistunut asiakasjulkaisujen ja niihin liittyvien palveluratkai-
sujen toteuttamiseen. Yritys tarjoaa asiakkailleen mm. kirjoja, luetteloita, jäsenhake-
mistoja perinteisinä painettuina tuotteina tai vaihtoehtoisesti verkkojulkaisuina.

Työn lähtökohtana oli yrityksen ViewMe-verkkojulkaisupalvelun kehittäminen.
ViewMe on verkkosovellus, jonka kautta asiakas lähettää verkkolehdeksi, -kirjaksi tai
-esitteeksi julkaistavan PDF-tiedoston Non-Stop-Studiot Oy:lle. Ladatut tiedostot siir-
tyvät automaattisesti tiettyyn kansioon. Yritys kääntää erillisellä ohjelmalla tiedoston
interaktiiviseksi verkkojulkaisuksi. Kun asiakkaan verkkojulkaisu on valmis, lähetään
asiakkaalle sähköpostissa linkki, jonka kautta hän pääsee lukemaan julkaisuaan Inter-
netissä. Verkkosovelluksella tarkoitetaan Internetissä toimivaa ohjelmistoa, jolla on
verkkosivulle soveltuva käyttöliittymä. PDF (Portable Document Format) on Adoben

kehittämä tiedostomuoto. Se pohjautuu PostScript-kieleen ja on käyttäjärjestelmäriippumaton tiedostomuoto, jota käytetään pääasiallisesti sähköiseen julkaisemiseen, tulostamiseen ja painamiseen. Verkkolehti, -kirja tai -esite on Internet-selaimesta luettava interaktiivinen flash-julkaisu, jonka sivuja voidaan käänellä samaan tapaan kuin paperiversion. Verkkojulkaisuun voidaan tekstin ja kuvien lisäksi liittää linkkejä, videopätkiä ja ääntä. Flash on Adoben kehittämä ohjelma, jonka avulla voidaan luoda flash-multimediaesityksiä verkkosivuille.

ViewMe-sovellus oli toiminnallisuudeltaan valmis, mutta sivuston ulkoasu ja käytettävyys olivat hätäisesti suunniteltu. Sovelluksella ei myöskään ollut markkinointisivustoa tai palvelusivustoa, josta saisi ohjeistusta tai tietoa verkkojulkaisupalvelusta. Tarkoituksena oli suunnitella ja toteuttaa sovelluksen rinnalle toinen sivusto, jonka avulla sovellusta markkinoitaisiin. Markkinointisivustoa kutsutaan nimellä ViewMe.

Asiakas toivoi markkinointisivuston toimivan julkaisujärjestelmällä, johon yhdistettäisiin kirjautuminen verkkojulkaisusovellukseen. Julkaisujärjestelmän avulla asiakas pystyisi päivittämään helposti sivustoa Internet-selaimesta käsin tekstieditoria käyttäen. Julkaisujärjestelmä vaihtoehtoina oli joko Joomla tai WordPress, koska molemmat ovat ilmaisia avoimen lähdekoodin ohjelmistoja. Päädyimme WordPressiin, koska kyseinen julkaisujärjestelmä oli asiakkaalle ennestään tuttu. Itse olin aiemmin toteuttanut sivustoja Joomla-pohjalle, joten minulla oli mahdollisuus tutustua uuteen järjestelmään ja kehittyä teknisesti pätevämmäksi.

3 JULKAISUJÄRJESTELMÄ

3.1 Julkaisujärjestelmän määrittely

Suomessa julkaisujärjestelmä (engl. publishing platform) rinnastetaan usein sisällönhallintajärjestelmään (engl. Content Management system). Puhuttaessa julkaisujärjestelmästä tarkoitetaan yleensä kevyttä tai keskiraskasta sisällönhallintajärjestelmää. Se mikä erottaa julkaisujärjestelmän sisällönhallintajärjestelmästä, on hallintaominaisuuksien määrä. Mitä enemmän järjestelmässä on sisältöihin liittyviä hallintaominaisuuksia, sitä lähempänä ollaan sisällönhallintajärjestelmää kuin julkaisujärjestelmää. (Tolvanen 2008.)

Sisällönhallintajärjestelmä on yleiskäsitys tietojärjestelmälle. Sisällönhallinnalla voidaan viitata eri tavoin painottuneisiin tietojärjestelmiin. Erilaisia järjestelmätyyppejä ovat mm. dokumenttienhallintajärjestelmät, verkkokauppajärjestelmät, aineistonhallintajärjestelmät ja wiki-järjestelmät.

Suurin osa markkinoilla olevista julkaisujärjestelmistä noudattavat samaa perusrakennetta, joka koostuu seuraavista kerroksista: tiedon tuotantovälineet, tiedon tallennusjärjestelmä, tiedon julkaisu, version hallinta ja käyttöoikeuksienhallinta. (Pohjanoksa, Kuokkanen, Raaska 2007, 212.)

Sisällönhallintajärjestelmään kuuluvat kaksi eri osaa: julkaisuohjelmisto ja hallintaohjelmisto. Julkaisuohjelmiston avulla välitetään tallennettu sisältö jakelukanavien kautta käyttäjille. Hallintaohjelmisto on sisällönhallinnan ydin, joka huolehtii sisällön osien välisistä linkityksistä. (Salmela 2002, 251.)

Sisällönhallinta- ja julkaisujärjestelmät mahdollistavat sisällön tuottamisen sivustolle siten, ettei ylläpitäjän tarvitse muokata HTML-tiedostoja käsin, siirtää niitä verkossa olevalle WWW-palvelimelle. HTML (Hypertext Markup Language) tai XHTML (eXtensible Hypertext Markup Language) on avoin standardoitu kuvauskieli, ei ohjelmointikieli. HTML:n avulla kerrotaan, missä järjestyksessä elementit esitetään sivulla. XHTML eroaa HTML:stä tiukemmillä muotoilusäännöillä. WWW-palvelin on tietokone, jossa on WWW-palvelinohjelmisto ja joka jakaa verkkosivuja Internetissä. Kaikki Internet-sivustot sijaitsevat jollakin WWW-palvelimella.

Helppoudessaan julkaisujärjestelmä toimii siis siten, että sivuston ylläpitäjä kirjautuu järjestelmän hallintasivustolle, päivittää artikkelin tekstieditorilla ja tallentaa muutokset nappia painamalla. Järjestelmä tallentaa sisällön automaattisesti tietokantaan tai tietokantatyypisesti erilleen julkaisujärjestelmästä. Tämä helpottaa verkkojulkaisua huomattavasti ja estää myös mahdollisia päivittäjän tekemiä virheitä, jotka voisivat rikkoa sivuston rakenteen tai muun sivuston toiminnallisuuden.

3.2 Julkaisujärjestelmän käyttötarkoitukset

Julkaisujärjestelmiä käytetään niin Internetissä kuin myös yritysten intranetissä. Intranet on tietyn ryhmän käyttöön eristetty lähiverkko. Tavallisesti sillä tarkoitetaan organisaation lähiverkkoa, jota käytetään sisäiseen viestintään. Julkaisujärjestelmän tarkoi-

tuksena on tarjota vaivaton tapa tiedon julkaisuun. Tämä mahdollistaa sen, että sivuston ylläpitäjä voi keskittyä sisältöön ilman teknisiä taitoja. Julkaisujärjestelmän valintaan vaikuttavat yrityksen tai yksityishenkilön käyttötarpeet.

Kevyet julkaisujärjestelmät ovat kehitetty pk-yrityksille tai pienille organisaatioille. Järjestelmien lisenssihinnat ovat 5 – 10 000 euroa. Ultrakevyet julkaisujärjestelmät ovat tarkoitettu lähinnä pk-yrityksille, yhdistyksille ja järjestöille. Ultrakevyitä järjestelmiä saa sovellusvuokrauksella, joten IT-osaamista ei tarvita. Sovellusvuokrauksen hinnaksi tulee vain kymmeniä euroja kuussa. (Pohjanoksa 2007, 215.)

Keskiraskaat julkaisujärjestelmät ovat suunnattu suurille yrityksille, joiden julkaisu-toiminta perustuu suoraviivaiseen verkkosisällönhallintaan. Sivustoja hallinnoi tällöin enimmäkseen muutamat sadat henkilöt. Keskiraskaiden julkaisujärjestelmien lisenssihinnat ovat 20 – 200 000 euron tuntumassa. Parhaat avoimenlähdekoodin julkaisujärjestelmä sijoittuvat myös keskiraskaiden joukkoon. (Pohjanoksa 2007, 215.)

Raskaat julkaisujärjestelmät ovat kehitetty suurien yritysten käyttötarpeisiin, jolloin sivustojen ylläpitoon osallistuu parhaimmassa tapauksessa tuhansia henkilöitä. Raskaat järjestelmät ovat suunniteltu siten, että niistä käsin voidaan hallinnoida satojentuhansien sivujen laajuisia kokonaisuuksia ja ne ovat yleensä integroitu tiiviisti yrityksen muihin järjestelmiin. Raskaan sarjan julkaisujärjestelmien kehittäjät ovat suuria tietotekniikkayrityksiä, jotka pyytävät järjestelmän peruskäyttöön oikeuttavasta lisenssistä 100 – 300 000 euroa. (Pohjanoksa 2007, 214.)

3.3 Hyödyt ja haitat

Julkaisujärjestelmää käyttävän näkökulmasta toivottuja ominaisuuksia ovat julkaisujärjestelmän helppokäyttöisyys, yksinkertaiset ajastukset ja mahdollisimman monipuoliset mahdollisuudet muokata sisältöjen lisäksi sivuston rakenteita ja ulkoasua. Kuitenkaan sisällönhallinnan kanssa näillä jälkimmäisillä vaatimuksilla ei ole juuri tekemistä keskenään. Julkaisujärjestelmät eivät ole olemassa verkkosivustojen kokoamista eivätkä ulkoasun muokkaamista varten. (Tolvanen 2008.)

Julkaisujärjestelmien suurimpia hyötyjä ovat sisällöntuotannon ja julkaisuprosessin nopeus sekä helppous. Tämä mahdollistaa hajautetun sisällöntuotannon, jolloin myös sisältöjen muokkaus saadaan lähemmäksi sisältöjen tuottajia. Hankintasyitä raskaam-

malle sisällönhallintajärjestelmälle ovat mm. hajautetun sisällöntuotannon tehostaminen, monikielisten sisältöjen hallinta, brändinhallinta, metatietojen hallintamahdollisuudet, laajan verkkopalvelukokonaisuuden hallinta ja monipuolisempien asiointipalveluiden mahdollistaminen. (Tolvanen 2008.)

Julkaisujärjestelmä otetaan käyttöön yleensä siinä vaiheessa, kun huomataan staattisen verkkosivuston kasvaneen liian vaikeasti ylläpidettäväksi. Laajan staattisen sivuston ylläpito on työlästä, sillä silloin on huolehdittava itse sivuston ja linkityksien toimivuudesta sekä vakioelementtien paikoista. Julkaisujärjestelmän avulla voidaan siirtää perustoiminnot ohjelmiston hoidettavaksi, joka säästää aikaa sisällön tuottamiseen ja muuhun ylläpitoon. Julkaisujärjestelmän avulla voidaan muuttaa esimerkiksi kaikkien sivuston sivujen navigaatorakenne eli linkkien paikka ja järjestys samanlaiseksi yhdestä paikasta yhdellä kertaa. Staattisella sivustolla navigaatorakenteen muutos täytyisi tehdä jokaiselle sivulle erikseen. Julkaisujärjestelmä hoitaa myös tiedon tallennuksen ja siirron tietokantaan automaattisesti.

Julkaisujärjestelmään voidaan asettaa erilaisia käyttövaltuuksia, joiden avulla voidaan määritellä ketkä saavat lukea, muokata tai käsitellä tiettyä sisällön osaa. Suurissa yrityksissä sivuston muokkausoikeuksia voidaan rajata esimerkiksi osastoittain luomalla erilaisia käyttöoikeusryhmiä. Käyttäjän lisääminen tiettyyn ryhmään antaa käyttäjälle automaattisesti enemmän oikeuksia tai päinvastoin. Tällä voidaan varmistaa myös se, että vahingossa ei muuteta muiden alueelle kuuluvia sivustoja. (Salmela 2002, 188–190.)

Julkaisujärjestelmät mahdollistavat myös usean ylläpitäjän työskentelyn samanaikaisesti toisistaan riippumatta. Ennen julkaisujärjestelmiä sisällöt ovat siirretty staattisille verkkosivuille erillisen ylläpitäjän ”webmasterin” kautta, joka on koostanut julkaisut teknisesti oikein ja siirtänyt ne WWW-palvelimelle (Kirjavainen 2009, 5). Staattinen sivu on tiedosto WWW-palvelimella. Tiedostoa ei pysty muokkaamaan selaimesta käsin, se täytyy tehdä HTML-koodia ymmärtävässä ohjelmassa tietokoneella. Tämän jälkeen tiedosto on siirrettävä itse palvelimelle. Julkaisujärjestelmät ovat dynaamisia, koska niiden sisältöä voidaan muokata selaimesta käsin, tietoa ei tarvitse itse siirtää palvelimelle, sillä järjestelmä hoitaa sen. Ilman julkaisujärjestelmää tai erillistä ylläpitäjää eri julkaisijat voisivat tallentaa toistensa julkaisuiden päälle tai rikkoa sivuston toiminnan.

Sivuston ylläpitäjän on myös mahdollista lisätä julkaisujärjestelmiin erilaisia lisäosia kuten analysaattoriohjelmia, joiden avulla voidaan seurata verkkosivuston käyttöä ja käyttäjien käyttäytymistä (Pohjanoksa 2007, 221). Tilastotiedoista on apua esimerkiksi verkkokaupalle, sillä kauppias voi seurata analysaattoriohjelman avulla asiakkaidensa mielenkiintoa tuotteitansa kohtaan ja siten kohdistaa markkinointiansa vaikkapa tietyille kohderyhmälle.

3.4 Avoimen lähdekoodin julkaisujärjestelmät

Kaupallisen lähdekoodin julkaisujärjestelmät ovat saaneet kilpailijoikseen joukon avoimen lähdekoodin julkaisujärjestelmiä. Avoimen lähdekoodin julkaisujärjestelmiä ovat mm. Drupal, Joomla, WordPress, Mambo, MODx, PHP-Nuke, phpWCMS, phpWebSite, Post Nuke, Texpattern, typo3 ja XOOP.

Kaupallisen ja avoimen lähdekoodin raja-aita on kuitenkin madaltunut huomattavasti viime vuosina. Tämä johtuu siitä, että suuret ohjelmistokehitysyrietykset käyttävät oman suljetun koodin lisäksi avointa koodia kasatakseen kokonaispaketteja. Myös useat markkinoilla olevat tuotteet ovat uudelleen brändättyjä avoimen lähdekoodin komponenteista. (Tolvanen 2008.)

Nykyisin monet verkkosivuja toteuttavat yritykset käyttävät liiketoiminnassaan avoimen lähdekoodin julkaisujärjestelmiä. Yritykset suunnittelevat verkkopalvelun ulkoasun, käyttävät julkaisujärjestelmää alustana tekniselle toteutukselle ja syöttävät annettun sisällön sivustolle. Järjestelmän ylläpitoliittymän koulutus kuuluu myös usein pakettiin.

3.5 Julkaisujärjestelmän tekninen toiminta ja vaatimukset

Monet avoimen lähdekoodin julkaisujärjestelmät ovat toteutettu avoimella PHP-ohjelmointikielellä. PHP (Hypertext Preprocessor) on dynaamisten verkkosivustojen luomiseen käytetty skriptikieli, jonka avulla luodaan verkkosivuille erilaisia toimintoja mm. verkkopalveluun kirjautuminen (Rantala 2002, 12). Sisällön tallentamiseen monet julkaisujärjestelmät käyttävät MySQL-tietokantaa, joka on myös avoimen lähdekoodin järjestelmä. MySQL-tietokanta on nimensä mukaisesti järjestelmä, jolla voidaan hallita tietokantoja eli sen avulla voidaan tallentaa tai hakea tietoa varastosta. (Heinisuo & Rauta 2007, 37.)

Edellä mainituista tekniikoista johtuen julkaisujärjestelmää ei voida asentaa mihin tahansa kotisivutilaan tai kehitysympäristöön. Tätä varten tarvitaan valmiille tuotteelle WWW-palvelin esimerkiksi niin sanotusta web-hotellipalvelusta, joka tukee PHP:tä ja sisältää MySQL-tietokannan. Kehitysvaiheessa tarvittavat ohjelmistot voidaan asentaa myös omalle työkoneelle tai omalle palvelimelle.

3.5.1 WWW-palvelin ja -ohjelmisto

Palvelimella voidaan tarkoittaa fyysistä tietokonetta, joka on yhteydessä johonkin verkkoon. Palvelimeksi voidaan kutsua myös tietokoneohjelmaa, joka ottaa vastaan tietoja asiakasohjelmilta ja vastaa sovitulla tavalla niiden lähettämiin tietopyyntöihin. (Heinisuo 2007, 12.)

Palvelinohjelmistoja on lukuisia, joista suosituimpia ovat Apache ja Microsoftin IIS. Nämä kaksi palvelinohjelmistoa eivät sisällä automaattisesti PHP-tukea, joten se täytyy asentaa itse. Myös MySQL-tuki ja halutut laajennukset on asennettava manuaalisesti. Saatavilla on myös paketteja, joissa on valmiiksi asennettuna Apache, PHP ja MySQL. Yksi tällainen ohjelma on esimerkiksi XAMPP. Valmiiksi asennetut paketit ovat testiympäristössä käyttökelpoisia, mutta tuotantoympäristöön on tehtävä asennukset alusta alkaen itse mm. tietoturvasyistä.

3.5.2 PHP

PHP on tulkittava kieli, jota WWW-palvelin ymmärtää. PHP-komentoja voi kirjoittaa suoraan HTML-sivujen sisään mihin kohtaan tahansa. Verkkosivulla oleva PHP-koodi ajetaan joka kerta juuri ennen kuin WWW-palvelin lähettää sivun selaimelle. Kun sivustolla oleva koodi suoritetaan, selaimessa näkyy vain ohjelman tulostus. PHP-koodi nähdään ainoastaan silloin, kun käsitellään palvelimella olevia tiedostoja. PHP suoritetaan aina WWW-palvelimessa ja sivut on ladattava WWW-palvelimelta. Tästä syystä PHP-koodia sisältävät sivut eivät toimi, kun käyttäjä avaa tiedoston suoraan tietokoneen levyiltä. (Heinisuo 2007, 13.)

PHP:llä toteutetut julkaisujärjestelmät tarvitsevat siis toimiakseen WWW-palvelimen ja -ohjelmiston, joka ymmärtää PHP-komentoja. PHP:n avulla julkaisujärjestelmiin voidaan luoda toimintoja, jotka esimerkiksi helpottavat sivuston ylläpitäjän työtä ku-

ten mahdollisuus sisällönmuokkaukseen selaimesta käsin. Sivustoista saadaan dynaamiset ja helposti ylläpidettävät.

3.5.3 MySQL

MySQL on relaatiotietokantoja käsittelevä tietokannanhallintajärjestelmä, jonka kyse-lykielenä toimii SQL. Relaatiotietokanta on yksi tietokantatyyppeistä, jossa tieto on järjestetty tauluihin ja taulujen keskinäiset suhteet eli relaatiot ovat merkittävässä osassa. MySQL-tietokantoja käytetään julkaisujärjestelmien lisäksi mm. lomakepalveluissa, sähköisen kaupan järjestelmissä ja yritysten Internet-palveluissa tiedon säilyttämiseen. MySQL:ää voidaan käyttää useista ohjelmointikielistä käsin kuten PHP:stä, C:stä, C++:sta, Perl:stä, Pythonista ja Tcl:stä. (Heinisuo 2007, 38.)

MySQL toimii sekä omana erillisenä sovelluksena, että palvelimena toisille ohjelmille. Toisin sanoen MySQL käyttää asiakas-palvelin arkkitehtuuria. PHP:n ja MySQL:n tapauksessa tämä tarkoittaa sitä, että tietokantaa ei käsitellä suoraan PHP:stä käsin, vaan PHP:llä otetaan yhteys MySQL-tietokantaan. PHP lähettää SQL-kielisiä komentoja palvelimelle ja MySQL vastaa esimerkiksi lähettämällä pyyntöä vastaavan sisällön tietokannasta. (Heinisuo 2007, 40.)

Yhdellä MySQL-palvelimella voi olla useita tietokantoja ja tietokannoilla useita tauluja, joihin tieto tallennetaan. Tämä mahdollistaa monimutkaisten käyttöoikeuksien luomisen, jossa jokaisella käyttäjällä tai ryhmällä voi olla eritasoiset oikeudet palvelimen tietokantoihin ja tauluihin. (Heinisuo 2007, 40.)

MySQL-tietokanta on yhtä tärkeä osa julkaisujärjestelmää kuin PHP. Ilman tietokantaa sivustolla ei olisi tietoa ylläpitäjistä, sivuston linkeistä tai sisällöstä. Tietokantaan tallennetaan julkaisujärjestelmän kaikki tekstisisältö, kuvien sijaintitiedot, ylläpitäjien tunnukset ja salasanat sekä muut mahdolliset tiedot. PHP:llä luodaan yhteys näihin tietoihin. Jos yhteyttä ei oteta tietokantaan, sivulla ei ole sisältöä, on olemassa pelkät HTML:llä ja CSS:llä muodostetut sivunraamit. CSS on tyylitiedosto, jonka avulla määritellään HTML:llä luotujen elementtien koot, sijainnit, värit yms.

4 WORDPRESS JULKAISUJÄRJESTELMÄ

4.1 Perusominaisuudet ja -toiminnot

WordPress (lyhennettynä WP) on PHP:llä ja MySQL:llä toteutettu ilmainen avoimen lähdekoodin julkaisujärjestelmä, jonka kehitys aloitettiin vuonna 2003. WordPressin juuret juontuvat edeltäjänsä b2/cafelog ohjelmiston ajoilta jo vuodelta 2001. WordPress oli alun perin suunnattu lähinnä blogeille, mutta ohjelmistokehityksen myötä sovelluksesta kehittyi kokonainen julkaisujärjestelmä. WordPress on myös GPL- lisensoitu. GPL (GNU General Public License) on vapaa ohjelmistolisenssi, jonka tarkoituksena on taata käyttäjälle oikeus kopioida, muuttaa ja jakaa edelleen ohjelmia ja niiden lähdekoodia. (About WordPress 2010.)

WordPressissä on lukuisia perusominaisuuksia ja -toimintoja, joiden avulla sivuston käytöstä on pyritty tekemään mahdollisimman helppoa. Lisäksi saatavilla on tuhansia lisäosia, widgettejä ja ulkoasuteemoja, joita käyttäjät ovat kehittäneet ja jakaneet muiden käyttäjien kanssa. Lisäosien ja widgettien avulla sivustolle voidaan luoda esimerkiksi kuvagallerioita tai esittää videoita. Tämän suuntaisen kehityksen on mahdollistanut WordPressin vapaasti muokattava lähdekoodi. WordPressin omien ominaisuuksien ja vapaan lähdekoodin tarkoituksena on taata eri käyttötarpeisiin soveltuva järjestelmä.

Tärkein lähtökohta toimivalle järjestelmälle on verkkosivujen koodistandardien noudattaminen. WordPressin mukaan standardien noudattaminen varmistaa koodin yhteensopivuuden eri selaimissa ja järjestelmän toimivuuden tulevaisuudessa (Features 2010). Kyseisistä syistä WordPress noudattaa W3C standardia. The World Wide Web Consortium (lyh. W3C) on kansainvälinen yhteisö, joka ylläpitää ja kehittää WWW:n standardeja tai suosituksia (W3C 2009).

WordPressiin on saatavilla useita ilmaisia teemoja, joiden avulla on mahdollista päivittää ulkoasua aivan yhtä helposti kuin muokata artikkelia. Tekstin muotoilu on suunniteltu käyttäjäystävälliseksi esimerkiksi siten, että jos käyttäjä vahingossa lisää tekstieditorissa useamman rivinvaihdon, koodillisesti lisätään vain yksi rivinvaihto. Typografiset erikoismerkit ovat myös huomioitu. WordPressissä käytetään menetelmää, joka muuttaa ASCII-merkistön mukaiset erikoismerkit koodillisesti korrektiin muotoon. (Features 2010.)

WordPressin pitkälle kehittynyt käyttäjähallinta sallii 10 eri käyttäjätason luomisen. Jokaiselle tasolle voidaan määritellä eri oikeudet katsella ja muokata julkaisuja. Käyttäjähallinta on tarpeen varsinkin silloin, jos sivustolla sallii rekisteröitymisen. (Features 2010.)

WordPress tukee myös paluuviihteitä (engl. *trackback*) ja päivitysilmoituksia (engl. *pingback*) (Features 2010). Paluuviihteiden avulla bloggaaja voi kertoa toiselle blogin kirjoittajalle kommentoineensa kirjoitusta omassa blogissaan. Paluuviihteitä lähetetään kopioimalla ilmoitettu paluuviihteosoite kommentoitavalta sivustolta ja liittämällä osoite oman blogin paluuviihtekenttään. Kun kirjoitus julkaistaan omassa blogissa, blogialusta lähettää päivitysilmoituksen automaattisesti blogiin, josta lähde on peräisin. (Lipsanen 2008.)

Järjestelmä pystyy suodattamaan suurenosan roskapostista, koska WordPressillä on ns. musta lista yleisimmistä roskapostin lähettäjistä ja avoimista välityspalvelimista, joiden kautta voidaan lähettää roskapostia. Tämän lisäksi saatavilla on lukuisia lisäosia, jotka voivat tehostaa roskapostitarkastusta. (Features 2010.)

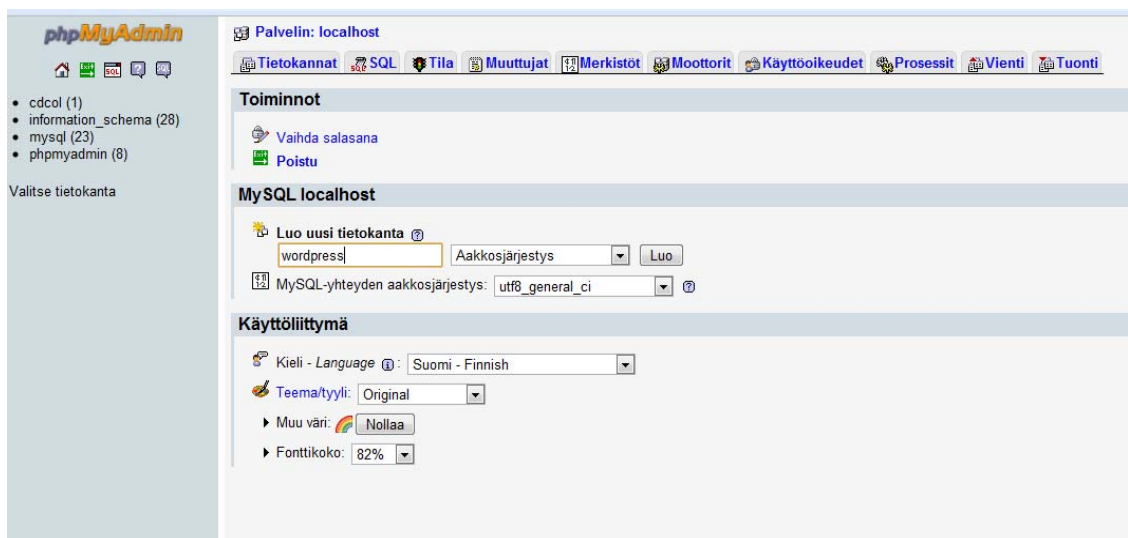
WordPressiin voi tuoda sisältöä Movable Type, Textpattern, Greymatter, Blogger ja b2 järjestelmistä. WordPress tukee myös XML-RPC rajapintaa, jota käyttävät Blogger API, MetaWeblog API ja MovableType API (Features 2010). Ohjelmointirajapinta eli API on käyttöliittymä, jonka avulla eri järjestelmät voivat tehdä pyyntöjä toisilleen ja vaihtaa tietoja keskenään. XML-RPC on tekniikka, jossa etäkutsuja lähetetään eri järjestelmien välillä HTTP-protokollan avulla.

Vaikka WordPress on tarkoitettu alun perin blogisivustoksi, voidaan järjestelmää nykyisin käyttää myös perinteisen sivuston tavoin. Blogikirjoitukset ja kommentit voidaan estää kokonaan tai sallia joillakin yksittäisillä sivuilla. WordPress sopii yhtä hyvin yksittäiselle henkilölle kuin myös suuremmalle organisaatiolle järjestelmän monipuolisuuden ansiosta. WordPress on käyttäjän näkökulmasta jo varsin hyvä perusominaisuuksiltaan, sillä käyttäjälle tärkeät ominaisuudet kuten artikkelien muokkaus, käyttäjähallinta, roskapostin suodatus ja tietoturva ovat kehittyneitä.

4.2 Asentaminen

WordPress vaatii toimiakseen WWW-palvelimen, joka tukee vähintään PHP:n versiota 4.3 ja MySQL:n versiota 4.1.2. WWW-palvelimena suositellaan käytettäväksi joko Apachea tai Nginxiä. WordPressin pitäisi toimia myös muilla palvelimilla, jotka tukevat PHP:tä ja MySQL:ää (Requirements 2010). Asennuksen ensimmäisessä vaiheessa on ladattava WordPressin uusin versio osoitteesta <http://wordpress.org/download>. Asennuspaketin voi ladata joko pakattuna ZIP tai TAR-tiedostona. Asennuspaketti on purettava tämän jälkeen ohjelmalla, joka osaa avata kyseisiä pakkaustiedostoja. (Installing WordPress 2010.)

Seuraavaksi purettu kansio siirretään WWW-palvelimelle. WordPressille on asennusta varten luotava tietokanta. Se miten tietokanta luodaan, riippuu asennusympäristöstä. Kuva 1 esittää kuinka MySQL-tietokanta luodaan PhpMyAdmin-sivuston kautta. Sivusto toimii kehitysympäristöön tarkoitettulla XAMPP WWW-palvelimella.



Kuva 1. Tietokannan luonti WordPressiin.

Helpoin tapa asentaa WordPress on käynnistää ohjattu asennus avaamalla selaimesta asennussivusto `/wp-admin/setup-config.php`. Asennussivu ohjaa käyttäjää luomaan `wp-config.php` tiedoston, nimeämään blogin ja luomaan käyttäjätunnuksen. `Wp-config.php` tiedostosta määritellään WordPressin tietokantaan pääsy eli tietokannan nimi (engl. Database Name), tietokannan käyttäjänimi (engl. Database Username), tietokannan salasana (engl. Database Password), tietokannan isäntä (engl. Database Host) eli tietokoneen nimi, jossa tietokanta sijaitsee. Ulkopuoliset webhotellin tarjo-

ajat voivat tarjota WordPressin valmiiksi asennettuna, jolloin edellä vaaditut tiedot saadaan palveluntarjoajalta. (Editing wp-config.php 2010.)

Kuva 2 esittää kuinka ohjatun asennuksen avulla luodaan WordPressin tarvitsema wp-config.php tiedosto. Aiemmin PhpMyAdmin sivustolla perustettiin tietokanta, jonka nimi kirjoitetaan tietokannan nimi (engl. Database Name) kohtaan. Muihin kenttiin kirjoitettavat tiedot riippuvat asennusympäristöstä. Esimerkiksi testiympäristö XAMPPia käytettäessä tietokannan käyttäjänimi (engl. Database Username) on oletuksena root ja salasana (engl. Database Password) tyhjä. Tietokannan isäntä on localhost, joka tarkoittaa paikallista tietokonetta. Localhostin tilalle voisi kirjoittaa myös tietokoneen nimen tai IP-osoitteen, jossa WWW-testipalvelin toimii.

The image shows the WordPress installation database configuration screen. At the top is the WordPress logo. Below it is a note: "Below you should enter your database connection details. If you're not sure about these, contact your host." The form consists of five rows, each with a label, an input field, and a description:

- Database Name:** Input field contains "wordpress". Description: "The name of the database you want to run WP in."
- User Name:** Input field contains "root". Description: "Your MySQL username"
- Password:** Empty input field. Description: "...and MySQL password."
- Database Host:** Input field contains "localhost". Description: "99% chance you won't need to change this value."
- Table Prefix:** Input field contains "wp_". Description: "If you want to run multiple WordPress installations in a single database, change this."

At the bottom left of the form is a "Submit" button.

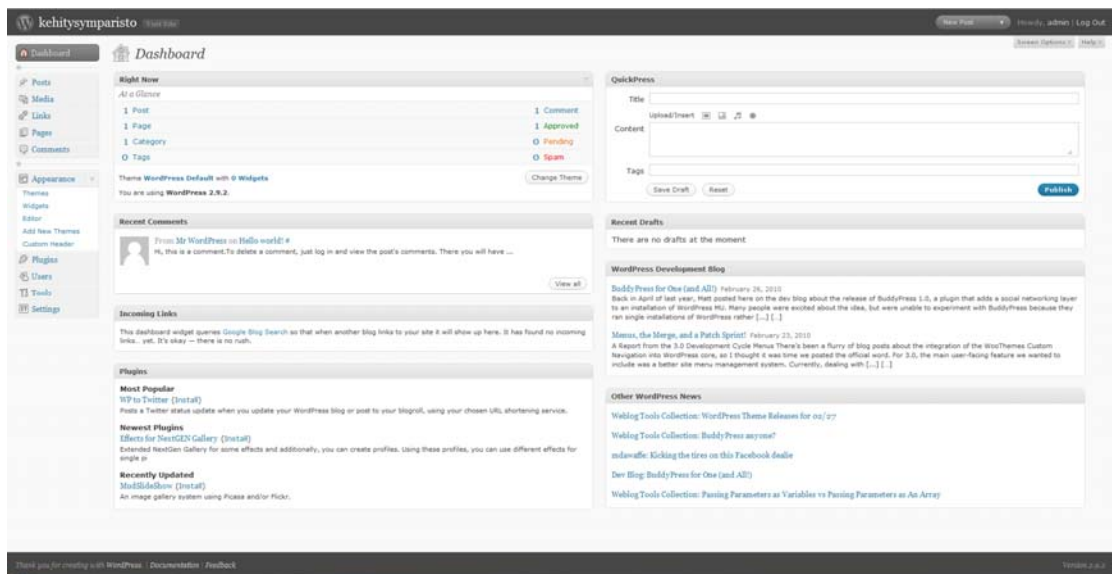
Kuva 2. WordPressin config.php-tiedoston luonti.

4.3 Hallintasivu

Hallintasivustolla (engl. Administrator Panel) voidaan hallita WordPress sivuston ominaisuuksia, joita ovat mm. sivun luominen, linkitys ja ulkoasuteeman vaihto. Hallintasivustolle pääsee kirjautumaan esimerkiksi kirjoittamalla selaimen osoitekenttään sivuston osoitteen perään /wp-admin.

Kuva 3 esittää kuinka hallintasivusto on jaettu neljään osioon. Osiot ovat ylätunniste (engl. header), päävalikko (engl. main navigation), työalue (engl. work area) ja ala-

tunniste (engl. footer). Ylätunnisteella on tummanharmaa tausta, jonka vasemmassa reunassa ilmoitetaan sivuston nimi ja tarjotaan linkki pääsivulle, pois hallintasivustolta. Ylätunnisteen oikeassa reunassa on pikalinkit eniten käytettyihin toimintoihin ja linkki käyttäjäprofiiliin. Päävalikko on vasemmassa reunassa, josta löytyy hallint ominaisuudet. Sivun keskellä on työalue, joka näyttää eri tietoja riippuen siitä, mikä linkki on aktivoitu päävalikosta. Työalueella voidaan esittää esim. tekstieditori, jonka avulla käyttäjä voi kirjoittaa uuden artikkelin sivustolle. Sivuston alareunassa tummanharmaalla taustalla on alatunniste, josta löytyvät linkit dokumentaatioihin ja palautteeseen. (Administration Panels 2010.)



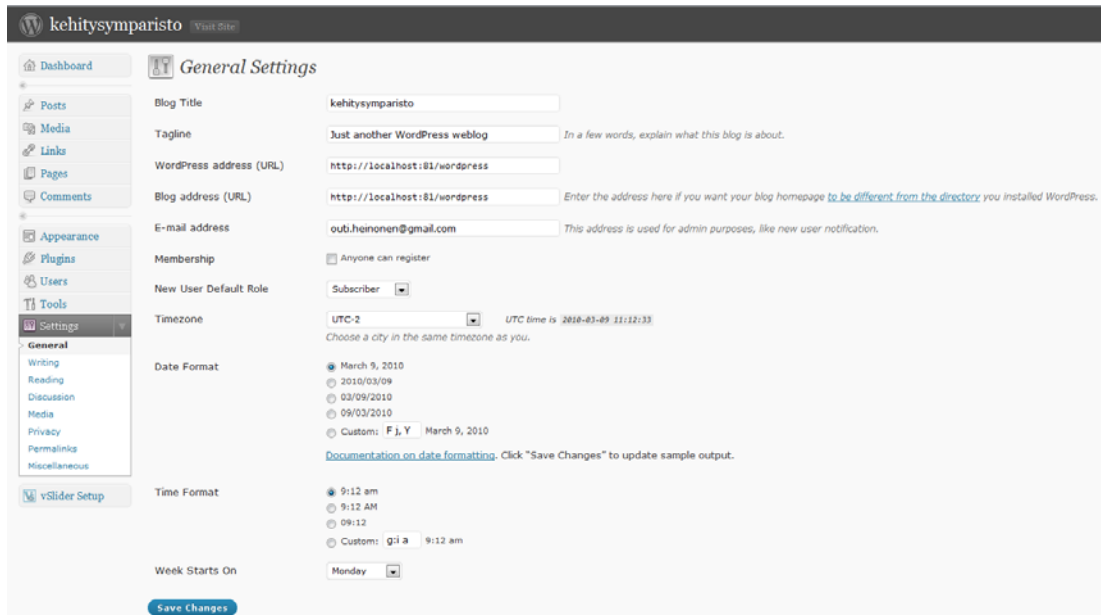
Kuva 3. WordPressin hallintasivusto.

4.4 Asetukset

WordPressin asetukset löytyvät hallintasivulta settings-välilehdeltä. Asetukset ovat jaettu ryhmiin yleiset (engl. general), kirjoitus (engl. writing), luku (engl. reading), keskustelu (engl. discussion), media, yksityisyys (engl. privacy), staattiset linkit (engl. permalinks) ja sekalaiset (engl. miscellaneous).

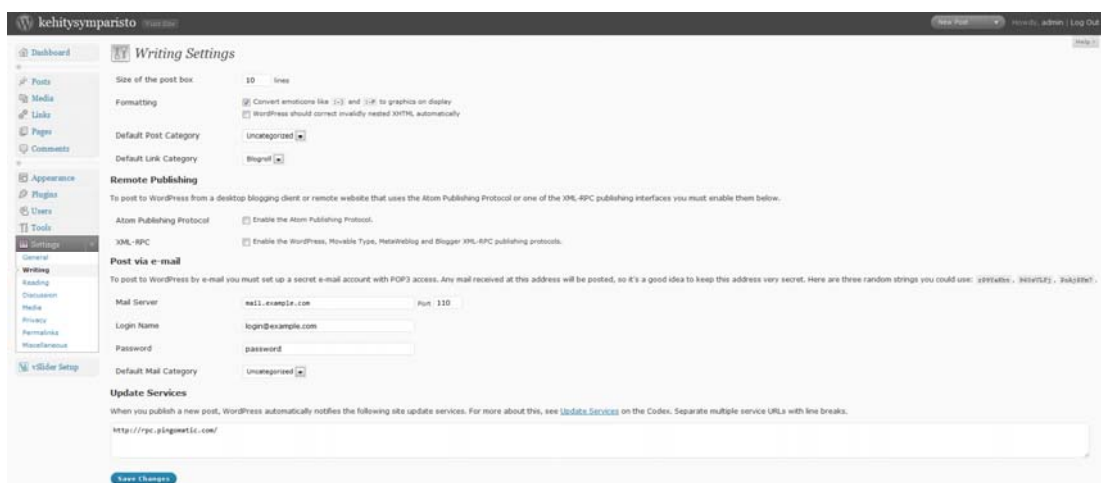
Kuva 4 esittää yleisien asetusten välilehteä, josta määritellään sivuston nimi, sivuston kuvaus, sivuston sijaintiosoitteet, ylläpitäjän sähköpostiosoite, sallitaanko rekisteröityminen, uuden käyttäjän oletusrooli, aikavyöhyke, päivämäärän ja ajan esitysmuoto sekä viikon aloituspäivä. Oletusulkoasueteemoissa sivuston nimi ja kuvaus näkyy sivuston ylätunnisteessa eli headerissa. Uuden käyttäjän rooli tulee merkittäväksi, jos sivustolle sallitaan rekisteröityminen, silloin on tärkeää määrittellä, mitä muutoksia

tuntemattomat voivat tehdä sivustolla. Aikavyöhyke, päivämäärä ja aika ovat tärkeitä mm. blogikirjoitusten kannalta.



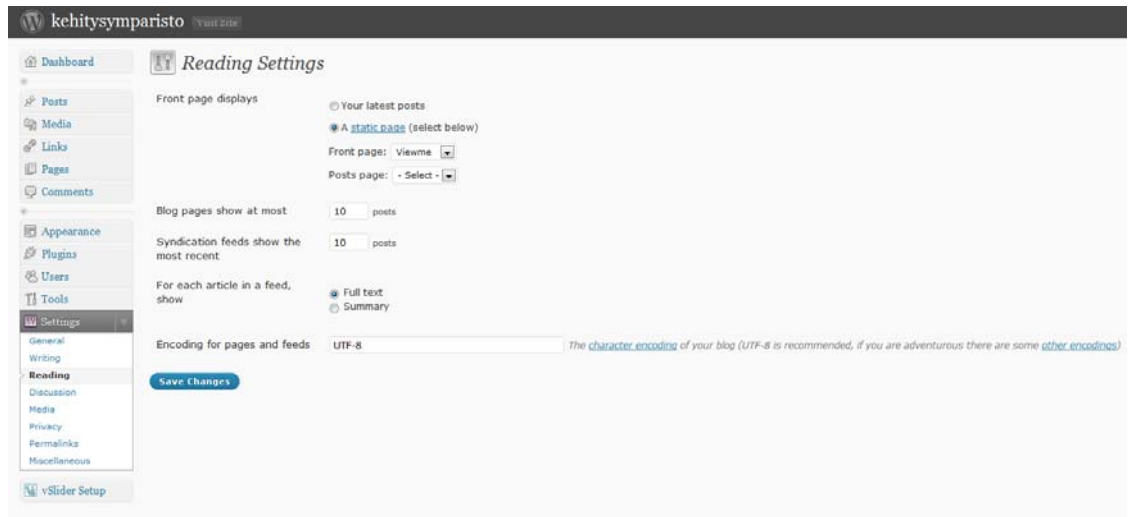
Kuva 4. WordPress-asetuksien yleiset asetukset.

Kuva 5 esittää kirjoitusasetuksien määrittämiä, joita ovat tekstilaatikon koko kirjoittaessa, ilmoitetaanko hymiöt tekstinä vai pikkukuvina, tarkastetaanko viesteihin kirjoitetun XHTML-koodin oikeellisuus, mikä on viestien ja linkkien oletus kategoria, sallitaanko etäjulkaisu Atom tai XML-RPC tekniikoilla ja sallitaanko julkaisu sähköpostin kautta. Sähköpostin kautta julkaisu on riskialtista, sillä WordPress julkaisee kaikki postilaatikkoon saapuneet viestit. Kyseisestä syystä sähköpostiosoitteen on oltava salainen ja tästäkin huolimatta riski väärinkäytölle säilyy.



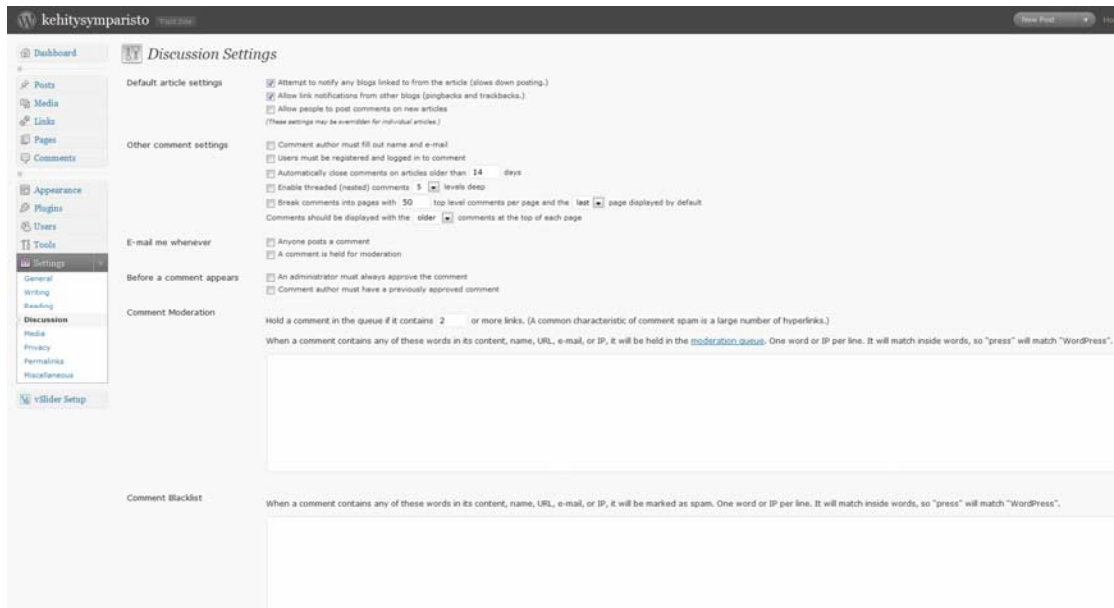
Kuva 5. WordPress-asetuksien kirjoitusasetukset.

Kuva 6 esittää lukuasetuksien välilehteä, josta määritellään näytetäänkö etusivulla viimeisimmät blogiviestit vai jokin sivu staattisena. Tämä asetus on hyödyksi, jos sivustoa käytetään blogin sijasta staattisena sivustona. Jos sivua käytetään blogin tapaan, on hyvä määritellä kuinka monta blogiviestiä näytetään kerrallaan ja näytetäänkö kerralla koko blogiviesti vai tiivistelmä.



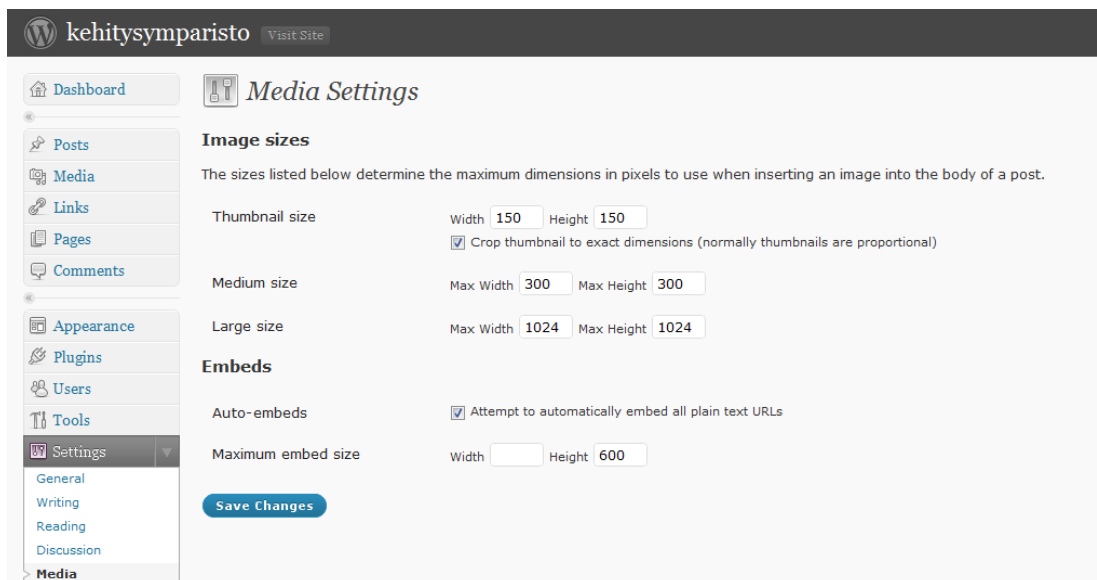
Kuva 6. WordPress-asetuksien lukuasetukset.

Kuva 7 esittää keskusteluasetukset välilehteä, josta hallinnoidaan blogikirjoitusten kommentointiasetuksia. Asetuksista määritellään sallitaanko paluuviihteet ja päivitysilmoitukset, sallitaanko uusien artikkelien kommentointi, tarvitseeko kommentoijan ilmoittaa nimi ja sähköposti, vaaditaanko kommentointiin rekisteröitymistä, kuinka kauan kommentit säilytetään, kuinka monta kommenttia näytetään sivulla, onko ensimmäisenä listassa uusin vai vanhin viesti, lähetetäänkö ylläpitäjälle sähköposti uuden kommentin saapumisesta, täytyykö ennen kommentin julkaisua ylläpitäjän hyväksyä se, suodatetaanko kommentteja, sallitaanko avatarit jne. Kommenttien suodatuksella tarkoitetaan sanoja, lauseita tai linkkejä, jotka ovat kielletty. Tämän avulla voidaan estää osa epäasiallisista viesteistä. Avatar tarkoittaa kuvaa, joka on linkitetty tiettyyn käyttäjään. Kuva näytetään esim. käyttäjän kommenttien yhteydessä.



Kuva 7. WordPress-asetuksien keskusteluasetukset.

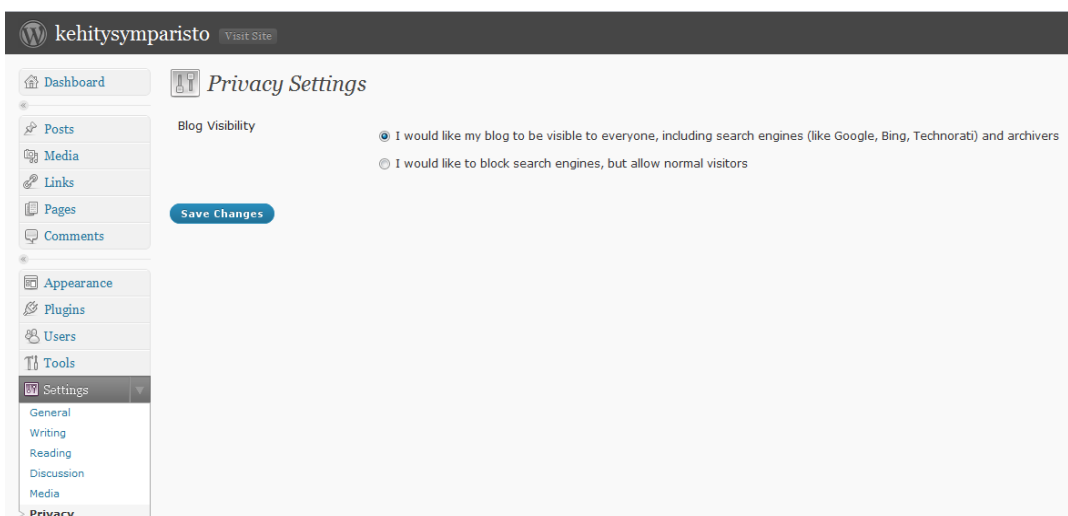
Kuva 8 esittää media-välilehteä, josta hallinnoidaan pienten, keskisuurien ja suurien kuvien kokoja. Pienet kuvat ovat lähinnä tarkoitettu esikatselukuviksi. Keskikokoiset ja suuret kuvat ovat tarkoitettu sivustolla normaaliin katselukäyttöön. Upotettua mediaa varten on myös koko asetukset. Upotetulla medialla voidaan tarkoittaa esim. videotiedostoa, joka halutaan näyttää osana sivua.



Kuva 8. WordPress-asetuksien media-asetukset.

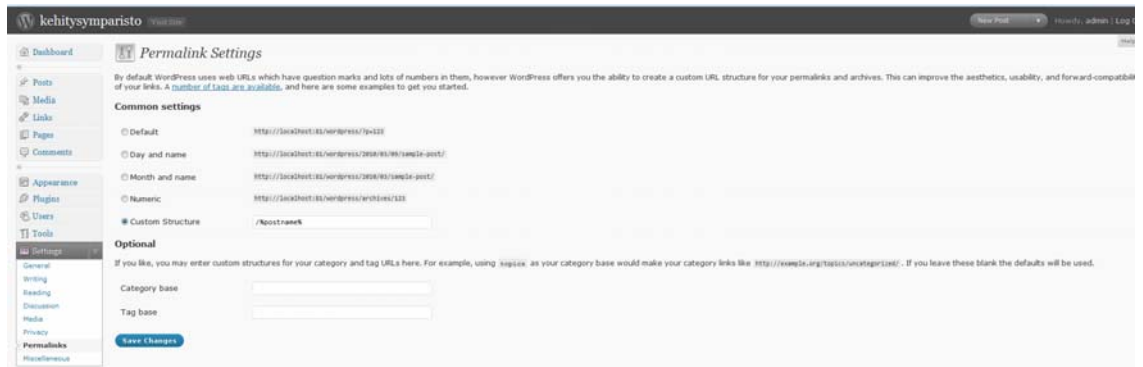
Kuva 9 esittää kuinka yksityisyysasetuksista määritetään sivuston näkyvyyttä hakukoneissa. Vaihtoehtoja on vain kaksi. Toinen valinta rajoittaa näkyvyyttä pelkästään hakukoneilta, kun toinen taas ei rajoita sitä mitenkään. Hakukoneilta rajoittaminen tulee

kyseeseen esimerkiksi tilanteessa, jossa yksityinen henkilö pitää blogia vain itseään varten. Hän ei välttämättä halua sivuston olevan helposti kaikkien saatavilla muuttaman hakusanan takana.



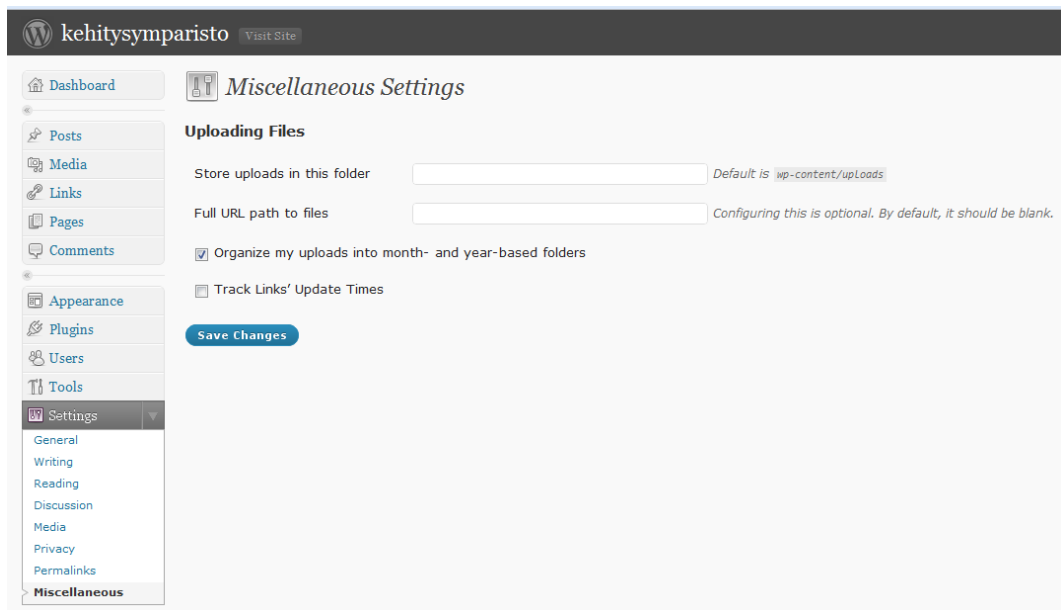
Kuva 9. WordPress-asetuksien yksityisyysasetukset.

Kuva 10 esittää kuinka staattiset linkit-osiosta hallinnoidaan WordPressin URL-osoitteen nimeämisperusteita. URL:ää (Uniform Resource Locator) käytetään osoittamaan WWW-sivujen osoitteita. Oletusasetuksena WordPress sivuston sivut saavat tunnisteeksi satunnaisen merkkijonon. Jos sivuston osoite olisi esimerkiksi `www.testi.fi` ja sivustolla olisi sivu nimeltä etusivu. Linkki etusivulle voisi olla esim. `www.testi.fi/?=123`. Asetuksista voi vaihtaa tämän sivun nimeämisperiaatteen päivä ja nimi, kuukausi ja nimi, numeeriseen tai itse määriteltyyn muotoon. URL-osoitteen voi määrittää myös kategorioille, joihin blogiviestin voi liittää niitä kirjoittaessa. Yksi hyvä ja selkeä vaihtoehto on esimerkiksi määrittellä itse, että jokainen sivu saa nimekseen artikkelin nimen. Tämä tapahtuu siten, että ”Custom Structure” kenttään kirjoitetaan `%postname%`. Nimeä ei haeta suoraan artikkelin otsikkokentästä, vaan sen alapuolelta permalink-kentästä. Esimerkiksi jos etusivun nimi olisi etusivu, tulisi linkiksi `www.testi.fi/?=123` sijaan `www.testi.fi/etusivu`. Tämä on paljon siistimpi ja haku-koneystävällisempi vaihtoehto.



Kuva 10. WordPress-asetuksien staattiset linkit.

Kuva 11 esittää Sekalaiset-välilehteä, josta löytyvät tiedostojen latausmääritykset kuten mihin kansioon ladatut tiedostot esim. kuvat tallennetaan. Jos kansion nimeä muuttaa, on ”Full URL path to files” kohtaan määriteltävä koko tiedostopolun URL-osoite, jossa tiedostot sijaitsevat. Tiedostojen järjestelyä varten on valittavissa järjestelläänkö tiedostot kansioihin kuukauden ja päivämäärän vai päivitysajan mukaan.



Kuva 11. WordPress-asetuksien sekalaiset-välilehti.

4.5 Käyttäjähallinta ja käyttöoikeudet

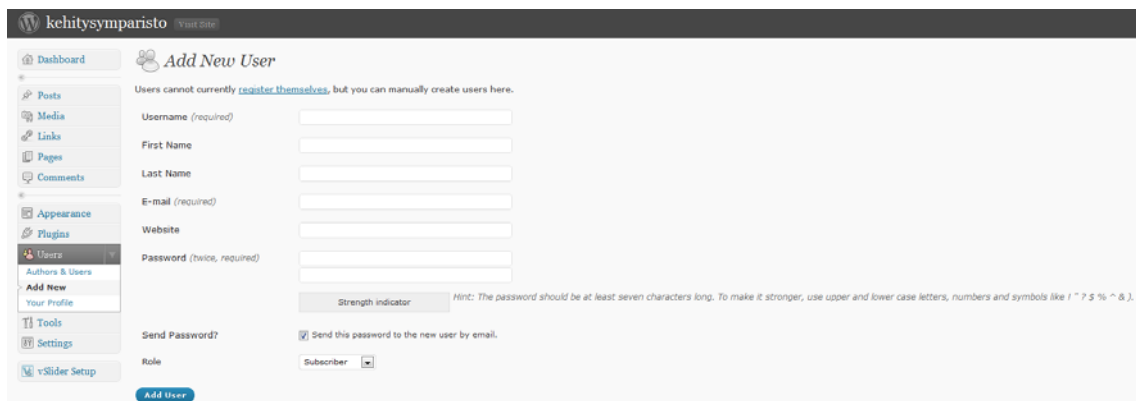
WordPressin käyttäjähallintaan ja käyttöoikeuksiin liittyvät asetukset löytyvät hallintasivulta Users-välilehdeltä. Sivulta voidaan lisätä, muokata tai poistaa yksittäisiä käyttäjiä tai käyttäjryhmiä sekä määritellä henkilöille tai ryhmille eriasteisia käyttöoikeuksia. (Authors and Users Subpanels 2010.)

Kuva 12 esittää käyttäjät-välilehteä (engl. Authors & users), jonka kautta nähdään lista kaikista sivuston käyttäjistä. Sivun kautta voidaan poistaa tai muokata valittua käyttäjää tai käyttäjäryhmää. Useaa käyttäjää voidaan hallita yhtä aikaa Bulk Action toiminnon avulla. Toiminnon avulla voidaan siirtää yhtäaikaaisesti useampi käyttäjän tiettyyn käyttäjäryhmään. Sivun oikeassa yläreunassa on etsintäikkuna, jonka avulla voidaan etsiä käyttäjiä käyttäjänimen, henkilön nimen, sähköpostin tai verkkosivun mukaan.



Kuva 12. WordPressin käyttäjät.

Lisää uusi (engl. Add New) alalistasta voidaan lisätä uusi käyttäjä. Käyttäjälle pakollisia määrittämiä ovat käyttäjätunnus, sähköpostiosoite ja salasana. Myös käyttäjän rooli eli käyttäjäryhmä on valittava. Erilaisia käyttäjäryhmiä ovat administrator, editor, author, contributor ja subscriber. Kuva 13 esittää sivua, josta uusi käyttäjä lisätään WordPressiin.

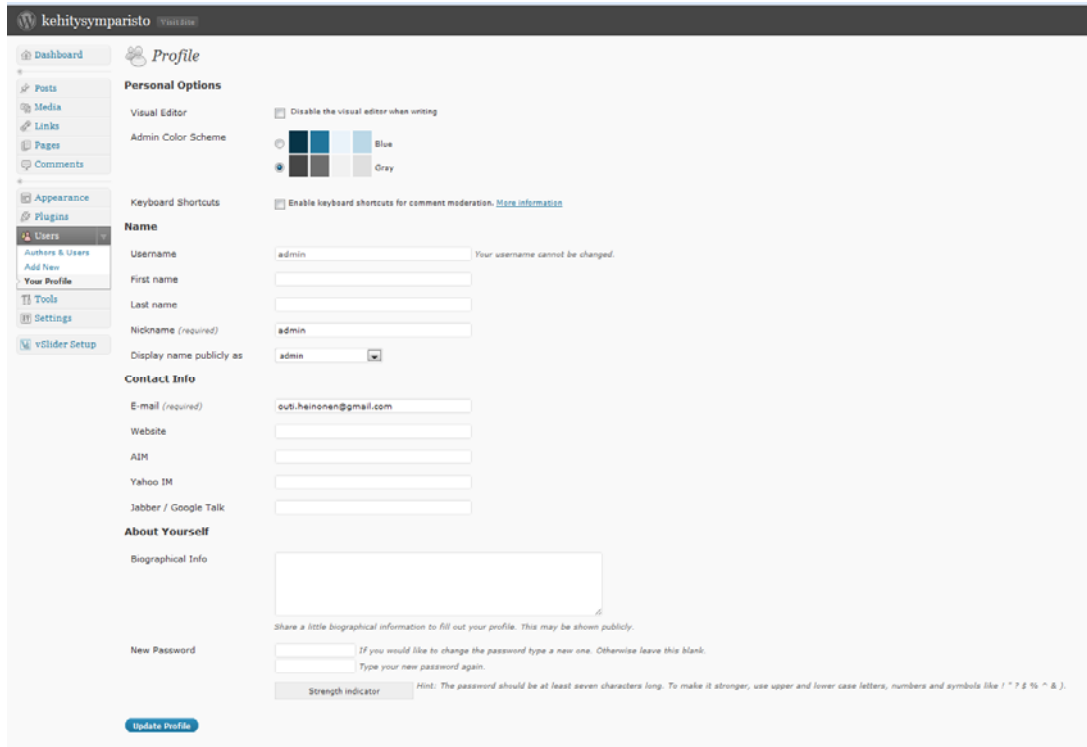


Kuva 13. Uuden käyttäjän lisääminen WordPressiin.

Jokaisella käyttäjäryhmällä on omat oikeutensa. Administrator on ylläpitäjä, jolla oikeus kaikkiin sivuston asetuksiin. Editorilla on oikeus kirjoittaa, julkaista, muokata joko omia tai toisten artikkeleita. Authorilla on oikeus kirjoittaa, muokata ja julkaista ainoastaan omia artikkeleita. Contributor voi kirjoittaa ja muokata omia artikkeleita,

mutta hänellä ei ole julkaisu-oikeutta. Subscriber voi ainoastaan tarkastella sisältöä. (Authors and Users Subpanels 2010.)

Kolmannesta alalinkistä Sinun profiilisi (engl. Your profile) voidaan tarkastella ja muokata omaa profiilia. Kuva 14 esittää kyseistä välilehteä, josta käyttäjä voi muokata henkilökohtaisia asetuksia. Käyttäjä voi vaihtaa admin-sivun väriteeman, muokata yhteystietoja, lisätä henkilökohtaisia tietoja ja uusia salasanaansa.



Kuva 14. Profiilin muokkaaminen WordPressissä.

4.6 Teemat

Teema tarkoittaa WordPressin ulkoasujärjestelmää, joka sisältää useita tiedostoja verkkosivuston graafisen käyttöliittymän tuottamiseksi. Teemojen tiedostoja kutsutaan nimellä template files. (Using Themes 2010.)

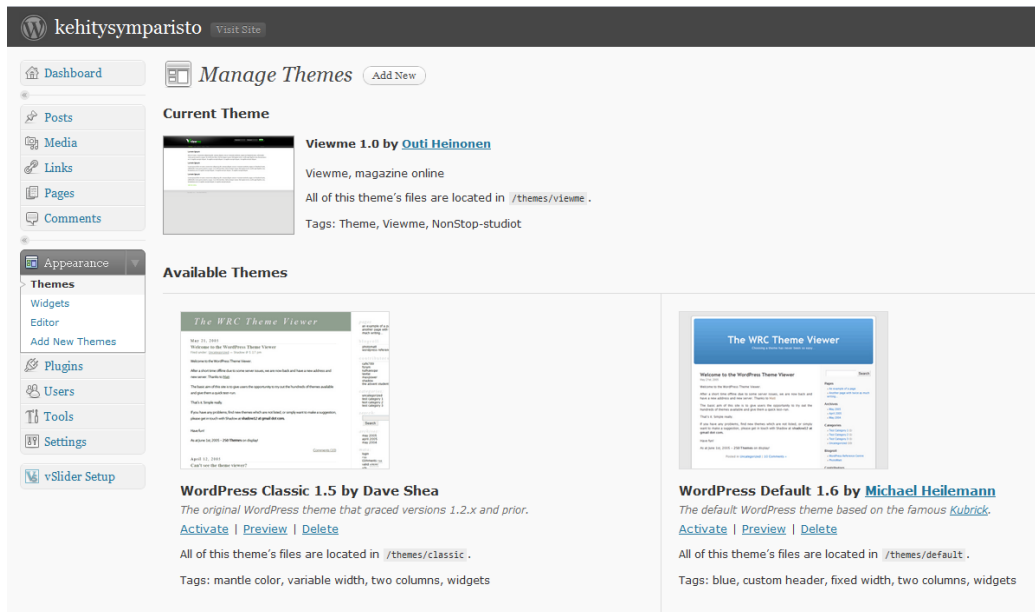
Teemat koostuvat tyypillisesti seuraavista tiedostoista:

- 404.php – tiedostoa ei löydy
- archive.php – blogiviestien listaus kategorian mukaan
- archives.php – vanhojen blogiviestien listaus
- comments.php – kommentointien listaus

- footer.php – alatunniste, sivun alareunassa
- header.php – ylätunniste, sivun yläreunassa
- links.php – linkkien listaus
- index.php – pääsivupohja
- page.php – staattinen sivupohja ilman blogia
- comments-popup.php – kommentit esitetään ponnahdusikkunoissa
- single.php – näyttää yhden blogisivun kommentteineen
- searchform.php – hakukentän sivupohja
- search.php – hakutuloksien sivupohja
- sidebar.php – sivuvalikko
- style.css – tyylitiedosto

Teema voi myös koostua pelkästään index.php tiedostosta. Kaikkien tiedostojen käyttö ei ole pakollista. Yleensä teeman pääsivupohja index.php kutsuu muita sivunosia esim. header.php:tä, page.php:tä, sidebar.php:tä ja footer.php:tä eli sivuston yläreunaa, staattista sisältöä, sivuvalikkoa ja alareunaa. Myös muut sivunosat voivat kutsua toisiinsa, jolloin esim. yllämainitun index.php:n kutsuma sivuvalikko voi kutsua linkkejä. Tiedostojen jakaminen osiin mahdollistaa erilaisten sivupohjien esittämisen samalla sivustolla esimerkiksi osan staattisina sivuina ja osan blogisivuina. Peruskäyttäjän ei välttämättä tarvitse tietää tai ymmärtää järjestelmän logiikkaa, sillä peruskäyttäjiä varten on runsaasti valmiita sivupohjia. Sivupohjien käyttöönotto ei vaadi koodaustaitoja.

Peruskäyttäjä voi ladata uusia teemoja ilmaiseksi wordpress.org sivustolta tai ostaa kolmansilta osapuolilta. Teeman latauksen ja tiedoston purkamisen jälkeen, teemakan-sio siirretään Word-Pressin kansioon wp-content/themes. Tämän jälkeen uusi teema löytyy hallintasivulta Appearance-linkin alta Themes-välilehdeltä. Teeman käyttöönotto tapahtuu activate-linkkiä painamalla. Kuva 15 näyttää missä teemat sijaitsevat hallinta sivulla.



Kuva 15. Teeman käyttöönotto WordPressissä.

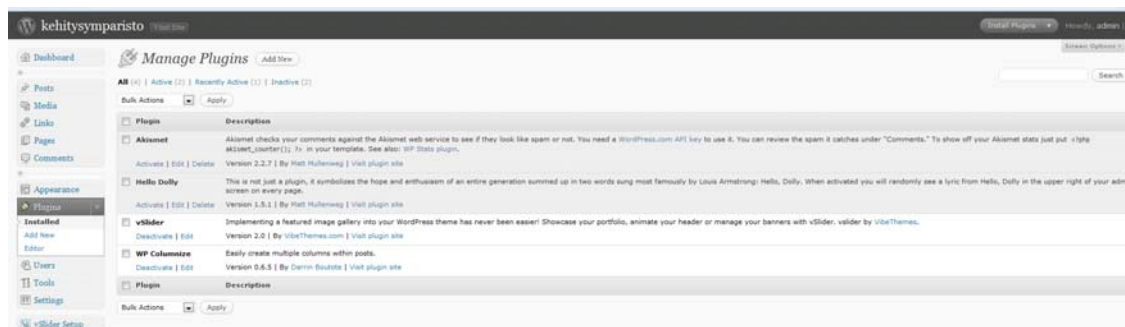
4.7 Lisäosat

Lisäosien (engl. Plugin) tarkoituksena on laajentaa WordPressin ominaisuuksia, jotta käyttäjät voisivat muokata järjestelmästä tarpeisiinsa sopivan. Järjestelmän mukana tulevat lisäosat Askimet ja Hello Dolly. Askimet tarkastaa järjestelmään lähetettyjä viestejä roskapostien varalta, siirtää epäilyttävät viestit tarkastuslistalle ja poistaa ne tietyn ajan kuluttua ellei ylläpitäjä toisin määrää. Toimiakseen lisäosa tarvitsee API-koodin, joka tarkastaa posteja ulkopuolisen palvelimensa avulla. Saadaksesen API-koodin on käyttäjän luotava tunnus wordpress.org sivustolle. Hello Dolly lisäosa sijoittaa hallintasivun oikeaan yläreunaan satunnaisia lauseita Hello, Dolly kappaleesta. Tämä on WordPressin ensimmäinen virallinen lisäosa, jolla on enemmänkin symbolista kuin käytännön arvoa. (Plugins 2010.)

WordPressin sivustolta on saatavilla runsaasti muiden käyttäjien kehittämiä lisäosia, joiden avulla sivustolle voi lisätä vaikkapa videoita ja kuvagallerioita. Käytännössä kuka tahansa, jolta löytyy vaadittavaa tietotaitoa voi kehittää WordPressiin oman lisäosan.

Valmiiden lisäosien asentaminen WordPressiin käy näppärästi. Ladattu ja purettu kansio siirretään WordPressin kansioon wp-content/plugins. Tämän jälkeen lisäosa aktivoidaan Appearance-linkin alta Plugins-välilehdeltä. Se kuinka lisäosa saadaan toimimaan aktivoinnin jälkeen, riippuu lisäosasta ja sen käyttötarkoituksesta. Esimerkik-

si jos käyttäjä haluaa lisätä videolisäosan sivustolle, saattaa lisäosa vaatia toimiakseen tietyn kutsukoodin artikkeliin, jossa video näytetään. Jotkut lisäosat saattavat myös vaatia käyttäjää lisäämään sivupohjaan PHP-lauseita. Lauseiden lisääminen käy WordPressistä käsin menemällä Appearance-linkin alta löytyvälle Editor-sivustolle. Lisäosien asentaminen voi kuulostaa hankalalta, mutta yleensä kehittäjät tarjoavat oppaan asennuksen kulusta. Pääsääntöisesti oppaat löytyvät englanninkielellä. Kuva 16 esittää välilehteä, josta näkyy lista lisäosista ja ovatko ne aktivoituna.



Kuva 16. Lisäosien aktivointi WordPressissä.

4.8 Tietoturva

WordPressiin liittyviä tietoturvariskejä on ollut joitakin vuosien 2007–2010 varrella. Vuonna 2007 WordPress julkaisi tiedotteen sivustoonsa kohdistuneesta palvelinmurrosta, jonka yhteydessä ohjelmistoversion 2.1.1 lataustiedostoon lisättiin haitallista koodia (WordPress Dangerous 2.1.1, upgrade to 2.1.2 2007). Muutama kuukausi tämän jälkeen julkaistiin helppokäyttöinen hyväksikäyttötyökalu, jonka avulla WordPressistä oli mahdollista saada luottamuksellisia tietoja tai suorittaa omia komentoja. Hyökkäysmenetelmät eivät toimineet enää versiota 2.2.3 vastaan. (WordPress-haavoittuvuuksiin julkaistu helppokäyttöinen hyväksikäyttötyökalu 2007.)

Vuoden 2008 huhtikuussa WordPress julkaisi ohjelmistoversion 2.5.1, koska aiemmista versioista oli löydetty vakavia turvallisuusriskejä. Aiempien versioiden tietoturva-aukot mahdollistivat hyökkääjälle ylläpitotunnuksen pääsynvalvonnan ohittamisen sopivaa evästä (engl. cookie) käyttämällä ja PHP-koodin suorittamisen palvelimella. Tätä tietoturva-aukkoa pystyttiin käyttämään ainoastaan, jos sivustolla oli uusien käyttäjien rekisteröityminen sallittuna. Myös syötteentarkistuksen puutteet mahdollistivat HTML- ja skriptikoodin tulkkauksen ja suorittamisen käyttäjän selaimesta. (WordPress-ohjelmistossa haavoittuvuuksia 2008.). Syyskuussa 2008 WordPressin

ohjelmistoversiosta 2.6.2. löydettiin jälleen kaksi riskiä, jotka yhdessä mahdollistivat käyttäjätunnusten kaappaamisen. Ensimmäinen tietoturva-aukko mahdollisti minkä tahansa käyttäjätunnuksen salasanan uudelleenasettamisen ja toinen mahdollisti järjestelmän luomien satunnaisten salasanoiden selvittämisen. (WordPress 2.6.2 julkaistu 2008.)

Seuraava tietoturvariski löydettiin vuoden 2009 elokuussa. Riski koski kaikkia WordPress versioita, jotka olivat vanhempia kuin versio 2.8.3. Tietoturva-aukko mahdollisti ylläpitotyökalujen oikeudettoman käytön, jolloin hyökkääjän oli mahdollista päästä käsiksi luottamuksellisiin tietoihin, muokata lisäosien asetuksia tai ladata sivustolle JavaScript-koodia. (WordPress-turvallisuuspäivitys 2009.) Saman vuoden lokakuussa WordPress järjestelmään tarjottiin uutta turvallisuuspäivitystä. Päivitys korjasi palvelunestohyökkäykset ja korjasi muita turvallisuusriskejä. (WordPress-blogialustan turvallisuutta parantava päivitys 2009.)

Helmikuussa 2010 WordPressistä löydettiin tietoturvariski, joka salli kirjautuneiden käyttäjien lukea muiden käyttäjien roskakoriin siirrettyjä viestejä. Hyökkääjällä oli mahdollisuus päästä käsiksi arkaluontoisiin tietoihin. Ongelma korjattiin WordPressin versioon 2.9.2. (WordPress-ohjelmistossa haavoittuvuus 2010.)

Edellä mainitut tietoturvatiedotukset ovat saatu viestintävirastossa toimivalta kansalliselta CERT-FI tietoturvaviranomaiselta. Organisaation tehtävänä on tietoturvaloukkauksen ennaltaehkäisy, havainnointi, ratkaisu ja tietoturvauhkista tiedottaminen. Sivustolta saadut tiedot osoittavat WordPressissä olleen vuosittain keskiarvoisesti kaksi tietoturvariskiä. Luku on suuri verrattaessa esimerkiksi Joomla-julkaisujärjestelmään. CERT-FI on tiedottanut Joomlaa koskevista tietoturvariskeistä kerran vuoden 2007 aikana, jolloin hakutoiminnon avulla pystyttiin suorittamaan PHP-komentoja (Haavoittuvuus Joomla-julkaisujärjestelmässä 2007). Vuoden 2008 aikana raportoitiin mahdollisuudesta poistaa järjestelmän ensimmäinen käyttäjätunnus (Joomla-järjestelmiä vastaan hyökätään aktiivisesti 2008). Vuoden 2008 aikana CERT-FI sai useita ilmoituksia kolmansien osapuolien kehittämistä lisäosista, jotka altistivat Joomlaan hyökkäyksille (Hyökkäyksiä Joomla! sisällönhallintajärjestelmiä vastaan 2008).

WordPress ja Joomla ovat kumpikin suosittuja avoimen lähdekoodin ohjelmistoja, mutta järjestelmät ovat kehitetty eri käyttötarkoituksiin. WordPress on kehitetty lähinnä blogisivustoksi, kun taas Joomla on tarkoitettu toimimaan staattisten sivustojen ta-

paan, joten tietoturvaongelmat ovat myös erilaisia. WordPressissä haavoittuvuudet liittyivät useimmiten käyttäjätileihin, kun taas Joomlaassa kolmansien osapuolien kehittämiin lisäosiin. WordPress on tietoturva-aukoistaan huolimatta blogisivustona useita muita blogi- tai julkaisujärjestelmiä kehittyneempi. Esimerkiksi Joomlaassa ei ole yhtä kehittyntä blogi- tai kommentointijärjestelmää kuin WordPressissä. Joomlaan voi halutessaan asentaa lisäosia, jotta järjestelmä toimisi kuten WordPressin blogi- tai kommentointijärjestelmä, mutta lisäosat saattavat altistaa järjestelmän helpommin tietoturvahyökkäyksille. Lisäosiin ei myöskään välttämättä kehitetä päivityksiä. Sisäänrakennetun blogijärjestelmän ja säännöllisesti julkaistavien blogi- ja tietoturva-päivityksien ansiosta WordPress on blogisivustona tietoturvallisempi ja kehittyneempi kuin esim. Joomla. Käytettäessä WordPressiä staattisen sivuston tavoin on se yhtä turvallinen kuin Joomla, sillä silloin käyttäjähallintamäärittämiä voidaan supistaa WordPressissä pienemmiksi.

WordPressin tietoturvaa voi jokainen järjestelmän ylläpitäjä parantaa muutamalla helpolla konstilla. Asennuksen yhteydessä kannattaa käyttää uusinta versiota. Säännöllinen järjestelmän, lisäosien ja teemojen päivitys on tietoturvan kannalta kaikkein tärkeintä (Virenius 2008).

Ennen asennusta suositellaan myös tietokantataulujen etuliitteen vaihtamista oletuksesta vaikeammin arvattavaksi. Etuliitteen voi muuttaa myös asennuksen jälkeen erillisen WP Prefix Table Changer-lisäosan avulla. (Virenius 2008.)

Tiedostojen ja kansioden kirjoitusoikeudet kannattaa myös tarkastaa, vaikka ne ovat oletuksena määriteltä turvallisiksi. Kirjoitusoikeuksien tarkastamista varten on myös saatavilla lisäosa WP Security Scan. Asennustiedosto wp-config.php-tiedosto kannattaa suojata joko siirtämällä se toiseen kansioon tai lisäämällä .htaccess-tiedostoon seuraavan komennon: `<Files wp-config.php> Order Deny, AllowDeny from All</Files>`. WP-Admin-kansio on hyvä suojata esim. AskApache Password Protect-lisäosalla tai Login Lockdown-lisäosalla. AskApache Password Protect-lisäosa vaatii käyttäjänimen ja salasanan wp-admin-kansioon pyrkiviltä käyttäjiltä. Login Lockdown-lisäosa tallentaa epäonnistuneiden kirjautumisten yhteydessä IP – osoitteen ja ajan. Jos yrityksiä on useita lyhyen ajan sisällä, kirjautuminen estetään kyseisestä IP-osoitteesta. (Virenius 2008.)

Kansioden selaaminen kannattaa estää, koska muuten sivustolta on vapaa pääsy esim. lisäosia sisältävään kansioon. Pääsyn kansioon voi estää lisäämällä tyhjän `index.html`-tiedoston `wp-content/plugins`-kansioon tai lisäämällä `.htaccess`-tiedostoon seuraavan komennon: `Options All -Indexes`. (Virenius 2008.)

Hakukoneita voi estää indeksoimasta `wp`-alkuisten kansioden sisältöä lisäämällä `robots.txt` tiedostoon seuraavan komennon: `Disallow: /wp-*`. WordPressin versionumero on hyvä poistaa lähdekoodista, jos päivitykset ovat jääneet väliin. Hakkereiden ja hakkerointiohjelmien on tällöin hankalampi löytää sivustoa. Varmuuskopioita kannattaa ottaa säännöllisesti. Salasanojen on oltava tarpeeksi pitkiä ja vaikeasti arvattavia. (Virenius 2008.)

5 VERKKOJULKAISUSIVUSTON SUUNNITTELU

5.1 Suunnittelun lähtökohdat

Huovilan (2006, 12–13) mukaan visuaalinen suunnittelun tarkoitus on tukea viestiä, antaa viestin lähettäjälle identiteetti, antaa esitetylle tiedolle järjestys, herättää ja ylläpitää vastaanottajan mielenkiinto. Tästä syystä ennen projektin käynnistämistä on tärkeää tehdä projektimäärittely, joka koostuu tavoitteista, uudelleen suunnittelun vaatimuksista, yrityksen asiakkaiden demograafisista piirteistä, projektin laajuudesta ja ylläpidosta (Goto & Cotler 2003, 42–43). Projektin määrittelyssä on tärkeää myös ymmärtää asiakkaan tarpeet siten, että toteutus vastaa asiakkaan tavoittelemia tuloksia (Hatva 2002, 31).

Projektinmäärittelypalaverissa kävimme läpi asiakkaan tavoitteet mm. kuinka verkkojulkaisusivusto kehittäisi liiketoimintaa. Asiakas halusi verkkosivuston rakenteen ja ulkoasun olevan yksinkertainen, pelkistetty ja selkeä, mutta tyylikäs. Tällöin sivuston sisällöllä olisi suurempi merkitys viestinnän ja markkinoinnin kannalta. Uudelleen suunnittelun lähtökohtana oli logosuunnittelu, verkkosivuston ulkoasun suunnittelu, tekninen toteutus WordPress julkaisujärjestelmällä ja ulkoasun päivittäminen PHP-pohjaiseen sovellussivustoon sekä näiden kahden sivuston yhdistäminen yhdeksi sivustokokonaisuudeksi. Sivuston viestinnällinen tarkoitus määrittelee palvelua käyttävien henkilöiden demograafisia piirteitä, joita ovat esimerkiksi teknisyyt, muutosmyönteisyys, kansainvälisyys ja ekologisuus. Tehtävät projektin suorittamisesta ja- kautui siten, että asiakas vastasi sisällön tuotosta ja minun tehtäväalueekseni jäi graa-

finen suunnittelu ja tekninen toteutus. Asiakas osallistui tiiviisti graafiseen suunnitteluun ja hänellä oli vahvat mielipiteet, joiden mukaan toimin.

5.2 Logon ja sivuston viestinnällinen tarkoitus

Huovilan (2006, 25) mukaan tuotteen, logon tai muun tunnuksen on viestittävä yhteisönsä periaatteita tai tuotteen ominaisuuksia. Tämän pohjalta voitiin määritellä logon ja sivuston viestinnällinen merkitys. Tarkoituksena on viestiä ekologisuutta, vihreää ajattelua, luonnon säästämistä, paperin säästämistä, teknistä kehitystä, uusia lukutapoja ja tulevaisuuden mahdollisuuksia. Logossa haluttiin tämän lisäksi symboloida PDF-tiedoston siirtämistä järjestelmään ja sen julkaisua verkkolehtenä, -kirjana tai -esitteenä.

Markkinoinnissa tunnusmerkin viestiä voidaan vahvistaa liittämällä siihen erilaisia mielikuvia, joiden avulla pyritään erottumaan kilpailijoista. Kun kuluttaja kokee tunnusmerkkiin yhdistyneet mielikuvat ja arvot omaan arvomaailmaansa sopiviksi, hän valitsee kyseisen tuotteen tai palvelun helpommin (Huovila 2006, 26). Palvelulla on myös hyvät mahdollisuudet kansainvälisillä markkinoilla, sillä sivusto on englanninkielinen.

Verkkojulkaisupalvelun markkinoinnissa voidaan vedota ihmisiin ekologisilla myyntipuheilla, sillä luonnonsuojelu on jatkuvasti ajankohtainen aihe. Mediassa puhutaan jatkuvasti maapallon lämpenemisestä, saasteista, fossiilisten energiamuotojen loppumisesta, uusien uusiutuvien energiamuotojen käytöstä ja ekologisimmista ratkaisuista. Luonnon suojelu antaa ihmisille positiivisia mielikuvia, jotka myös sopivat useimpien arvomaailmaan. Verrattaessa kymmentä muuta vastaavaa verkkojulkaisupalvelua oli heidän markkinointistrategiansa aivan toisenlainen.

Uudet lukutavat ja tulevaisuuden mahdollisuudet ovat ajankohtaisia teknistyvässä yhteiskunnassa. Näitä uusia lukutapoja ja tulevaisuuden mahdollisuuksia on myös tarkoitus viestiä sivuston kautta. Aholan, Koivumäen ja Oinas-Kukkolan (2002, 166) mukaan suurin osa ihmisistä omistaa jonkin kannettavan päätelaitteen, jolloin liikkuvuus ja langaton viestintä mahdollistavat verkkopalvelujen saatavuuden lähes kaikkialla. Esimerkiksi uusimmat matkapuhelimet ovat teknisesti selainominaisuuksiltaan tietokoneeseen verrattavia. Tulevaisuudessa arkipäivää voivat olla enemmissä määrin tietokoneelta, puhelimesta tai muusta vastaavasta kämmenlaitteesta luettavat verkkoleh-

det ja -kirjat. Suurta suosiota on saanut esimerkiksi Kindle 2 -sähkökirja, jolla voi lukea Amazonin Kindle-kaupasta ostettuja digitaalisia kirjoja ja lehtiä. Laitteeseen mahtuu yli 1 500 kirjaa, näytössä on hyvä kontrasti ja se toimii ilman taustavaloa, jolloin sitä on luettava oikean kirjan tapaan valon alla (Herttua, I. 2010). Apple on myös julkaisemassa verkkosivujen ja verkkojulkaisujen selaamiseen tarkoitettun Ipad-laitteen. Kun tämälntapaiset laitteet yleistyvät, on verkkojulkaisupalveluilla tulevaisuudessa yhä merkittävämpi rooli.

5.3 Logon suunnittelu

Asiakas määritteli logosuunnittelulle muutaman ehdon. Logon päävärinä käytettäisiin kirkkaan vihreää, joka toimisi mustalla taustalla. Muotoehdotuksia logolle olivat kirja, lehti, silmä tai V-kirjain. V-kirjain viittaisi ViewMe nimeen. Logossa sai myös käyttää markkinointisivuston nimeä.

Visuaalisessa suunnittelussa on otettava huomioon ihmisen havainnointijärjestelmä, joka perustuu ympäröivään maailmaan. Ihminen omaksuu ympäröivästä maailmasta mittasuhteita, muotoja ja värejä. Käyttämällä näitä luonnollisia mittasuhteita sommitelussa voidaan varmistaa hyvä viestintä. Sommitelun apuna voidaan käyttää kultaista leikkausta tai erilaisia matemaattisia malleja. Vastakkain asettelulla, kontrastieroilla, koolla, välimatkalla, suunnalla yms. voidaan luoda jännitteitä mielenkiinnon herättämiseksi. Jännitettä saa aikaiseksi esimerkkinä käyttämällä samassa sommitelmassa pyöreitä ja teräviä muotoja. (Huovila 2006, 28–31.)

Ensimmäisissä hahmotelmissa en kiinnittänyt huomiota mittasuhteisiin, sommitteluun, tai väreihin sen tarkemmin, sillä hahmotelmieni tarkoitus oli löytää suunta muodolle. Ensimmäiset hahmotelmat koostuivat asiakkaan määrittelemistä elementeistä silmistä, kirjoista, lehdistä ja V-kirjaimista, mutta ne eivät muistuttaneet muodoiltaan ja väreiltään juurikaan logon viestinnän tarkoitusta. Ensimmäinen hahmotelma silmästä ja lukijasta ruudukon päällä liittyi ideaan logon animoinnista, jolloin palaset liittyisivät yhteen muodostaen ymmärrettävän kuvion. Tein useita silmä-, lehti-, kirjakuvioita, jonka jälkeen logon kehitys suuntautui yhä enemmän V-kirjaimen käyttöön. Ensimmäinen V-kirjainmainen logosuunnitelma oli kirjahylly, jossa kaatuneet kirjat muodostivat V-kirjaimen. Asiakkaan lisäidea V-kirjaimesta, joka muistuttaisi avattua kirjan selkää, antoi uutta suuntaa suunnittelulle. Tämän jälkeen logon muoto löytyi nopeasti. Ensiksi tein mallin harmaana, jossa oli useita ruohomaisen muotoisia avattuja sivuja.

Muotoa oli hiukan yksinkertaistettava, jotta se toimisi hyvin myös pienenä. Tästä muotoutui sitten lopullinen kuvio, jonka vasemmalla puolella on pieni lovi ja oikealla pystysuunnassa logon läpileikkaava lovi sekä nouseva ruohomainen viiva loven vieressä. Lovi vasemmassa reunassa kuvaa kirjan sivujen väliin tulevaa jakoa, kun kirja on avattuna. Se myös kuvaa PDF-tiedoston sisään latausta. Oikean reunan lovi antaa kasvavalle ruoholle pidemmän muodon ja tekee jaon valkoisen ja vihreän värin välille. Oikean puolen lovi ja viiva kuvaavat myös materiaalin julkaisua järjestelmästä. Valmiiseen kuvioon saatiin jännitettä muutamalla terävällä kulmalla. Liikettä ja mielenkiintoa lisättiin viivoilla. Logokuvion kehityksen kulku ideoista logokuvion valmiiseen muotoon esitetään liitteessä 1.

Jotta logokuvion vierelle saataisiin Viewme-teksti, oli seuraavaksi mietittävä typografiaa. Laajasti määriteltynä typografia käsittää verkko- tai printtijulkaisun graafisen ulkoasun. Alun perin typografialla tarkoitettiin ainoastaan kirjainmerkkien suunnittelua, mutta nykyään typografiaan kuuluu kaikki tekstiin liittyvä suunnittelu. Typografisia ominaisuuksia ovat kirjasintyyppi eli fontti, kirjasimen koko, sanan merkkiväli, rivin pituus, palstan väli, kappaleen muotoilu. Kirjasintyypit jaetaan muodon mukaan antiikvoihin eli päätteellisiin ja groteskeihin eli päätteettömiin. Typografiaa voidaan myös käyttää tukemaan kuvia tai toisin päin. (Raninen ja Rautio 2003, 221.)

Sinkkosen, Kuoppalan, Parkkisen ja Vastamäen (2002, 144) mukaan päätteettömät kirjasimet erottuvat näytöltä paremmin kuin päätteelliset. Syynä tähän on näyttöjen huono tarkkuus, jonka vuoksi päätteelliset kirjaimet voivat sotkea luettavuutta. Päätteelliset kirjasimet puolestaan parantavat paperilla luettavuutta. Pienaakkosten käyttöä suositetaan myös otsikoissa, sillä ne erottuvat paremmin kuin suuraakkoset. Verkkosivulla kursiivi voi sotkea luettavuutta ja alleviivaus voidaan sekoittaa linkkiin, joten näitä ei suositella käytettävän. (Sinkkonen 2002, 144.)

Käytettävästä kirjasimesta oli neljä eri ehdotusta. Ensimmäinen kirjasintyyppi oli Clearfac Gothic MT, joka on päätteetön, tasapaksu ja teräväreunainen. Kirjasintyyppissä on vino e-kirjain. Toinen ehdotus oli Frankfurter SH, joka on päätteetön, tasapaksu, reunoiltaan pyöreä ja pehmeä. E-kirjain on myös vinossa. Kolmas vaihtoehto oli Schueller BQ, joka on tasapaksu, päätteetön, teräväreunainen ja hiukan kursiivi. Neljäs vaihtoehto oli Vag Rounded, joka on päätteetön ja tasapaksu, reunoiltaan pyöreä ja pehmeä. Itse miellyin Frankfurter SH ja Vag Rounded kirjasintyyppeihin pyöreiden

ja selkein muotojen vuoksi. Frankfurter SH:n vino e-kirjain antaa mielestäni logoon liikettä ja siten lisäpotkua. Lukuun ottamatta e-kirjaimen asentoa, Vag Rounded on muodoiltaan samantyylinen kuin Frankfurter SH. Clearfac Gothic MT oli mielestäni liian jyrkää, jolloin se vei huomiota logokuviossa. Schueller BQ oli hiukan kursiivi, joten se ei välttämättä toimisi hyvin pienessä koossa. Frankfurter SH ja Vag Ronded toimivat kumpikin pienessä koossa. Asiakas valitsi kirjasintyypiksi Vag Roundedin.

Seuraava askel logon kehityksessä oli värien määrittely. Ranisen (2003, 230) mukaan väreillä on psykologisia vaikutuksia työtehoon ja mielialaan sekä symbolisia merkityksiä kaikissa kulttuureissa. Logon värit suunniteltiin symboloimaan haluttua viestiä ja mielikuvia. Mustalla taustalla logon värit ovat vihreä ja valkoinen, kun taas valkoisella taustalla värinä ovat vihreä ja musta. Huovilan (2006, 46) mukaan vihreä symboloi luontoa, tasapainoa ja terveyttä, valkoinen kuvastaa puhtautta ja kunniaa, ja musta high-techiä. Värit voivat symboloida eri asioita asiayhteydestä riippuen. Esimerkiksi logokuviossa valkoinen väri symboloi kirjan sivuja ja musta kuvaa tekstin väriä. Psykologisesti vihreä väri myös rauhoittaa ihmistä (Huovila 2006, 45). Kun värit ja muodot olivat päätetty, tarkastettiin logon mittasuhteet. Mitat määritettiin suhteellisesti x- korkeuksina. Typografiset vaihtoehdot, valmis logo ja mittasuhteet esitetään liitteessä 2.

Logolle saatiin pieni graafinen ohjeisto, jossa määriteltiin logon mittasuhteet, käyttö, koot ja värit. Sinkkosen (2002, 154) mukaan musta taustaväri korostaa muita värejä entisestään. Tästä syystä logoa pyritään käyttämään mustalla taustalla. Ohjeistossa määritellään, että logoa saa käyttää tarvittaessa myös valkoisella pohjalla, mutta silloin tekstin ja kuvion valkoiset osat vaihdetaan mustaksi. Logon kuvio voidaan esittää yksin, mutta logon tekstiä ei.

Logon koot:

- Suurin: 66 mm leveä.
- Suositeltu: 44 mm leveä.
- Pienin: 22 mm leveä.

Logon vihreän värin määrittely eri värijärjestelmillä:

- HSB 90 | 100 | 83

- RGB 102 | 212 | 0
- WEB #66D400
- CMYK 62 | 0 | 100 | 0

5.4 Graafisen käyttöliittymän suunnittelu

Lähtökohtana oli rakentaa Wordpress-julkaisujärjestelmällä ViewMe-markkinointisivusto sovellussivuston rinnalle. Sovellussivuston ulkoasu tuli päivittää WordPress-järjestelmän kanssa samanlaiseksi. Nämä kaksi sivustoa tuli yhdistää yhdeksi kokonaisuudeksi siten, että käyttäjä ei kirjautuessaan huomaa sivuston vaihtuvan.

Goton (2003, 99) mukaan viestinnän ja visuaalisen suunnittelun pohjana ovat rautalankamallit, joiden avulla määritellään tärkeimpien sivujen sisältöjen paikat. Rautalankamallien pohjalta tehdään klikkailtava protosivu, joka ei sisällä välttämättä toiminnallisuutta tai valmista ulkoasua. (Goto 2003, 128.)

Ennen graafisen käyttöliittymän suunnittelua oli tutustuttava sovellussivuston tekniseen toimintaan, sillä se määritteli kuinka käyttöliittymä ja ulkoasu voidaan toteuttaa, kun huomioidaan kahden eri järjestelmän yhdistäminen toisiinsa. Tässä tapauksessa ensimmäinen protosivu tehtiin ennen rautalankamalleja, jotta tiedetään voidaanko WordPress-järjestelmään kutsua toisen sivuston yläosa kirjautumisineen. Liitteessä 3 on kuva entisen sovellussivuston etusivusta ja protosivusta, kun sovelluksen yläosa on kutsuttu WordPress-järjestelmässä toimivalle testisivulle. Kun tekninen toteutusmahdollisuus oli varmistettu, voitiin aloittaa sivuston rakenteen ja visuaalisuuden suunnittelu.

Elementtien asettelulla, visuaalisella suunnittelulla ja typografialla pyritään saavuttamaan mahdollisimman hyvä kommunikointi käyttäjän ja tuotteen välillä. Asettelen tansapaino riippuu värien, kontrastien ja tyhjän tilan käytöstä sekä elementtien muodosta, sijoittelusta ja liikesuunnista. Käyttöliittymässä tärkeintä on sen toimivuus ja sisältö. Visuaalisella suunnittelulla voidaan parantaa myös uusien käyttäjien käyttöliittymäosaamista ja vaikuttaa tuotteen käyttötehokkuuteen. (Sinkkonen 2002, 176–177.)

Käyttöliittymän elementtejä ja värejä mietittäessä oli huomioitava logon käyttöohjeet ja sen sijoittaminen sivustolle. Logo sijoitettiin vasempaan yläreunaan, koska Sinkkonen (2002, 120) mukaan katse ohjautuu ensimmäisenä vasempaan yläreunaan, ellei jo-

kin muu kiinnittää käyttäjän huomiota ensin. Koska logo tuli esittää mustalla taustalla, valittiin sivuston yläosan väriksi musta. Oikeaan reunaan sijoitettiin sovellussivustolle kirjautuminen. Kirjautumisnapin väriksi valittiin sama vihreä kuin logossa, jotta käyttäjä kiinnittäisi huomion myös oikeaan reunaan. Asiakas halusi sivuston linkkien olevan huomaamattomia, ehdotti niiden asettamista pelkästään sivuston alareunaan, joka ei olisi ollut käytettävyyden kannalta järkevä ratkaisu. Käyttäjä ei välttämättä olisi osannut etsiä linkkejä alareunasta, jos ne eivät olisi näkyneet joka sivulla heti. Selitin asian asiakkaalle ja linkit sijoitettiin vaakatasoon sivun yläosaan. Sivuston oli tarkoitus olla hyvin pelkistetty, mutta näyttävä. Tästä syystä yläosassa käytettiin heijastuksia ja läpinäkyvyyttä. Sinkkosen (2002, 99) mukaan pysty- ja vaakasuorat alueet, symmetriset osat sekä kuperat ja koverat elementit nähdään helposti kuviona, koska ihminen on tottunut havaitsemaan kolmiulotteisia kohteita taustaa vasten. Sivuston valikossa ja yläosassa pyrittiin heijastuksilla luomaan juuri kolmiulotteisuuden vaikutelmaa. Yksinkertaisen sivuston saavuttamiseksi sisältöalueen väriksi valittiin valkoinen ja alaosan väriksi harmaa. Alaosaan sijoitettiin yhteystiedot. Sinkkosen (2002, 153) mukaan mm. puhtautta ja rehellisyyttä kuvastava valkoinen sopii muiden värien taustalle ja antaa ilmavuutta sommitteluun. Sinkkosen (2002, 154) mukaan harmaa puolestaan muodostaa laajana alueena vaaleille ja tummille väreille neutraalin taustan. Väriinä se kuvastaa mm. arkisuutta, yhtenäisyyttä, turvallisuutta, menestystä, tyyneyttä, kypsyyttä ja toivekuutta (Sinkkosen 2002, 154). Kuutin (2003,100) mukaan käyttöliittymässä värien käyttö tulisi rajata noin viiteen. Sivuston käyttöliittymässä on mustaa, valkoista, harmaata ja vihreää, joten värejä ei ole liikaa ja ne sulautuvat hyvin yhteen.

Kuva 17 esittää ensimmäistä versiota yläosasta. Taustalla on runsaasti valoisuutta. Logon värit eivät ole vielä tässä vaiheessa ohjeen mukaiset. Logo peilautuu kokonaan mustasta. Kirjautumislaatikko on aika suuri, joka herättää huomiota hiukan liikaa.



Kuva 17. Yläosan ensimmäinen versio.

Kuva 18 esittää valmista yläosaa. Logoa suurennettiin hiukan, laskettiin lähemmäksi linkkejä ja värit korjattiin ohjeen mukaisiksi. Logon peilautumista pienennettiin. Kirjautumislaatikon tekstikentät muutettiin harmaiksi, jotta ne eivät erottuisi yhtä voimakkaasti kuin valkoiset. Päänavigaation linkkejä vaalennettiin hiukan. Yläosasta on-

nistuttiin suunnittelemaan pelkistetty, mutta heijastuksilla siihen saatiin elävyyttä ja mielenkiintoa. Logo erottuu kirkkaasti mustasta taustasta. Yläosan musta väri heijastuksineen tuo mieleen arvokkuuden, korkean laadun, teknisen osaamisen ja yritysmaailman. Valikko on logon alapuolella, jolloin se huomataan helpommin kuin esimerkiksi asiakkaan ehdottamasta sivun alareunasta. Kirjautumislaatikko erottuneen vihreän kirjautumisnapin ansiosta. Kirjautumiskentän alapuolelta löytyy rekisteröinti-linkki. Käyttäjän kirjautuessa sisään rekisteröinti-linkki häviää ja tilalle avautuu kirjautuneen käyttäjän linkit.



Kuva 18. Valmis yläosa.

Käyttöliittymän suunnittelussa kuten logon suunnittelussa tuli ottaa huomioon myös typografia. Sen lisäksi, että verkkosivulla tulisi käyttää näyttöjen huonon resoluution vuoksi päätteettömiä kirjasimia, on myös muitakin ohjeistuksia. Kuutin (2003, 100) mukaan paras tulos luettavuuden kannalta saavutetaan kontrastierolla, tumma teksti vaalealla pohjalla tai toisinpäin. Sivustolla suositellaan käytettävän enintään kolmea eri kirjasintyyppiä esimerkiksi yksi kirjasintyyppi otsikolle, toinen alaotsikolle ja kolmas leipätekstille. Leipäteksti tulisi tasata vasempaan reunaan ja jättää oikeareuna liehureunaksi. Sopiva rivin pituus on 50–60 merkkiä. (Sinkkosen 2002, 144–146.)

Valittaessa verkkosivulle kirjasinlajeja, kannattaa käyttää ns. yleisimpiä kirjasintyyppisiä, sillä erilaisissa tietokoneissa on erilaiset valikoimat. Arvion mukaan yli 50 %:ssa tietokoneista voidaan olettaa olevan päätteellisistä kirjasimista Book Antiqua, Bookman Old Style, Garamond, Georgia, Palatino, Linotype, Sylfaen ja Times New Roman. Päätteettömistä kirjasimista Arial, Century Gothic, Lucida Sans, Lucida Sans Unicode, Microsoft San Serif, Tahoma, Trebuchet MS ja Verdana. Jos käyttäjän koneelta ei löydy jotakin verkkosivulla määriteltyä kirjasintyyppiä, voidaan sitä varten sivuston tyylitiedossa määritellä varalle toisia kirjasintyyppisiä. Jos käyttäjältä ei löydy yhtäkään tyylitiedossa määriteltyä kirjasinlajia, käytetään selaimen oletus kirjasinta. (Korpela 2008, 194–196.)

Asiakkaalla oli vahva näkemys leipätekstin kirjasintyyppin valinnasta. Asiakas oli mieltynyt päätteelliseen Georgia kirjasimeen eikä käytettävyyden perusteluista huolimatta suostunut vaihtamaan kirjasinta toiseksi. Asiakaan perustelu Georgian valintaan

oli viestinnällinen. Hänen mukaansa kirjasin on tuttu pokkareista ja lehdistä, joiden luettavuus on paperillakin hiukan suttuista. Tämä viestii hänen mukaansa sivuston tarkoitusta ja ideaa julkaista verkossa kirjoja, lehtiä ja esitteitä. Georgian avulla voidaan luoda mielikuvia printtijulkaisuista ja liittää nämä mielikuvat digitaaliseen julkaisuun.

Käyttöliittymän päänavigaation linkit olivat asiakkaan toiveenmukaisesti pyritty tekemään vähemmän huomiota herättäväksi, joten linkkien kirjasin koko jätettiin hyvin pieneksi. Pienessä koossa Georgia olisi päätteidensä vuoksi sotkenut linkkien luettavuutta, joten asiakas myöntyi linkeissä päätteettömään Trebuchet MS kirjasimeen. Trebuchet MS valittiin myös otsikoiden kirjasimeksi.

Sisältöalueella leipäteksti ja otsikot ovat taustaansa tummempia, joka tekee tekstistä helposti luettavaa. Vaikka leipätekstissä käytetään Georgiaa, on se kuitenkin näytöllä isossa koossa yllättävän selkeää. Päänavigaation linkeissä tausta on vastaavasti tumma ja linkit hiukan vaaleampia. Kun kursori viedään linkin päälle, sen väri muuttuu valkoiseksi, jolloin käyttäjä havaitsee selkeästi mille sivulle linkki vie. Kun käyttäjä on painanut sivun aktiiviseksi, näytetään aktivoituneen sivun linkki vihreänä. Vihreällä voidaan myös kiinnittää käyttäjän huomio valikkoon, joka on hiukan huomaamaton.

Sinkkosen (2002, 116) mukaan käyttäjää voidaan ohjata katsomaan haluttuun suuntaan käyttämällä esimerkiksi juuri voimakkaita värejä. Sinkkosen suosituksen mukaan sivustolla käytettävien kirjasinlajien määrä ei ylitä kolmea, sillä sivulla käytetään kahta erilaista kirjasintyyppiä: Georgiaa ja Trebuchet MS:ää. Sisältöaluetta ei voida asettaa rivinpituuden vaatimusten mukaisesti 50–60 merkkiin, sillä julkaisujärjestelmän käyttö on suunniteltava siten, että ylläpitäjä voi halutessaan asettaa myös esim. kuvia sisältöalueelle haluamaansa kohtaan. Julkaisujärjestelmissä on myös riskinä se, että ylläpitäjä voi asettaa sivun tekstit ja kuvat oman mieltymyksen mukaan. Sisällön suhteen käytettävyyttä ei voida taata julkaisujärjestelmissä täysin, ellei ylläpitäjällä ole tarvittavaa tietoa hyvästä sisällön asettelusta.

Ulkoasun suunnittelun ja määritysten jälkeen voitiin tehdä HTML-pohjainen protosivu, jonka avulla Goton (2003, 129) mukaan voidaan varmistaa elementtien asettuminen sivustolla, sisältöjen saumaton yhteen sopivuus ja mahdolliset ongelmat. Koska tässä vaiheessa sivustolle ei ollut sisältöä, voitiin protosivun avulla varmistaa ainoastaan typografia, elementtien sijainnit ja sisältöalueen toimivuus tekstin kanssa sekä si-

vuston skaalatuvuus, jos selainta pienennetään. Protosivun jälkeen voitiin aloittaa varsinainen tekninen toteutus, jonka yhteydessä sivuston harmaaseen alaosaan lisättiin vielä ruohokuvio. Asiakas halusi sijoittaa alaosaan samat linkit kuin yläosan navigaatiossa ja sähköpostiosoitteet kolmeen palstaan allekkain. Liitteessä 4 on kuva toisen vaiheen protosivusta ja viimeistellystä valmiista etusivusta.

Ulkoasun suunnitteluun kuului myös sähköpostiviestien päivittäminen. Sähköposteissa käytettiin logoa valkoisella taustalla ja tekstiksi määriteltiin Georgia samalla värillä sekä koolla kuin verkkosivustolla. Sähköpostiohjelmat saattavat kuitenkin suodattaa määritellyt tyylit ja kuvat. Viesti tulee perille, mutta visuaalisuus ei välttämättä.

Sivuston ulkoasu ja värimaailma onnistui mielestäni hyvin. Käyttöliittymässä muuttasin linkkejä hiukan helpommin näkyviksi ja vaihtaisin leipätekstin päätteettömään kirjaseimeen Georgian sijasta. Sivuston harmaaseen alaosaan en olisi sijoittanut linkkejä uudelleen, olisin sijoittanut vain päivämäärän ja sähköpostiosoitteet yhdelle riville.

6 VERKKOJULKAISUSIVUSTON TEKNINEN TOTEUTUS

6.1 Markkinointisivuston tekninen toteutus WordPressillä

Tekninen toteutus voitiin jakaa kolmeen osa-alueeseen, jotka olivat markkinointisivuston rakentaminen WordPress-järjestelmään, sovellussivuston ulkoasun päivittäminen ja näiden kahden erijärjestelmän yhteensovittaminen yhdeksi sivustoksi. Yksittäisen sivuston ulkoasun rakentaminen tai päivittäminen, voi olla jo itsessään haasteellista, mutta vieläkin haasteellisemmaksi sen tekee se, että nämä kaksi täysin erilaista järjestelmää oli yhdistettävä toisiinsa.

Tekninen toteutus käynnistyi jo ensimmäistä protosivua tehdessä, jolloin testasin yläosan tuontia sovellussivustolta WordPress-järjestelmään. Tätä varten tarvitsisin kehitysympäristön. Yksi vaihtoehto oli ostaa sivutilaa joltakin web-hotellipalvelulta, joka tarjoaa vaadittavat ominaisuudet WordPressin toimivuudelle. Toinen vaihtoehto oli asentaa paikalliselle koneelle palvelinohjelmisto. Päädyin asentamaan omalle koneelleni kehitysympäristöksi XAMPPin. Tämä mahdollistaa sen, että tehdyt muutokset koodissa nähdään heti tallentaessa. Ostettuun sivutilaan tiedostot pitäisi siirtää jokaisen tallennetun muutoksen jälkeen, joten tämä olisi hankaloittanut kehitystä.

Tämän jälkeen kehitysympäristöön oli asennettava WordPress-järjestelmä. Asennus tapahtui luvun 4.2 kuvaamalla tavalla. Ennen asennusta loin WordPressille tietokannan selaimen kautta osoitteesta <http://localhost:81/phpmyadmin>. Yleensä osoitteena paikalliselle koneelle on pelkkä localhost, mutta olin määritellyt sen toimimaan portista 81, koska localhostin oletusportti oli käytössä toisella ohjelmalla. Käynnistin XAMPPin Control Panelista eli hallintapaneelistä Apachen ja MySQL:n. Latasin WordPressin uusimman version 2.9.2 ja siirsin puretun kansion paikallisella koneella olleeseen XAMPPin kansioon `xamplite/htdocs`. Selaimesta avasin <http://localhost:81/wp-admin/setup-config.php> asennustiedoston ja toimin kuten luvussa 4.2 ohjeistetaan.

Seuraavaksi voitiin aloittaa oman teeman kokoaminen kehitysympäristössä. Luvussa 4.6 kerrottiin, WordPressin teemojen sisältävän useita tiedostoja, joiden avulla voidaan toteuttaa tarpeisiin sopiva graafinen käyttöliittymä. Teemakansio `themes` löytyy WordPressin pääkansion sisältä, kansioista `wp-content`. Kansio sisältää muutaman WordPressin oletusteeman, joita tutkimalla pystyin perehtymään tiedostojen rakenteeseen ja pohtimaan markkinointisivuston teeman toteutusta. Käyttöliittymän suunnittelun yhteydessä oli protosivun avulla testattu yläosan erottaminen koodillisesti muusta sivustosta. Tämä mahdollisti sen, että esim. WordPressin yläosan tilalle voitiin kutsua toista yläosaa toiselta sivustolta.

Tutkittuani teemojen toimintaa ja mahdollisuuksia, aloin koota omaa teemaa. Tässä vaiheessa en huomioinut kahden eri sivuston yhdistämistä, vaan pyrin kokoamaan teeman WordPress-ympäristössä toimivaksi. Toisen protosivun yhteydessä oli sivuston ulkoasu, elementtien paikat ja tyyli tiedostot määritelty hyvin pitkälle. Pystyin soveltamaan protosivun HTML-koodia WordPressissä melkein suoraan ja näin sisällyttämään valmiiseen rakenteeseen WordPressin toiminnallisuutta ohjaavat komennot. Muutokset näin saman tien, kun aktivoin teeman luvun 4.6 mukaisesti.

Ensimmäiseksi rakensin ainoastaan `index.php` sivun, johon sisällytin WordPressin tarvitseman koodin. Tämän lisäksi minulla oli erillinen tyyli tiedosto, jossa määrittelin elementtien paikat, kuvat, fontit ja värit. Kun olin varmistanut sivuston toimivuuden, jaoin sivun kolmeksi tiedostoksi: `header.php`, `index.php` ja `footer.php`. Lisäksi `Viewme_template` niminen teemani sisältää myös `404.php`, `comments.php` ja `style.css` tiedostot sekä `images`-kansion kuvia varten. Tiedostojen nimeämisessä on noudatettava

luvussa 4.6 esitettyä listaa, sillä WordPressin toimintoihin on määritelty listan mukaiset tiedostonimet. Järjestelmä muodostaa kyseisellä tavalla nimetyistä tiedoston osista kokonaisen sivun. Jos header.php nimettäisiin vaikkapa ylaosa.php, ei järjestelmä tunnista sitä eikä sivusto toimisi oikein.

Header.php kuvaa nimensä mukaisesti HTML sivuston headeria eli yläosaa, jossa määritellään dokumentille tyypillisiä asioita. Kuva 19 esittää WordPressin header.php tiedostoa. Kuva 20 esittää koodin visuaalisessa muodossa. Kuvassa punaisella näkyvät PHP-tagit liittyvät WordPressin omiin toimintoihin, joiden avulla järjestelmä määrittelee mitä arvoja asetetaan kyseisiin kohtiin. Rivillä yksi kerrotaan millainen dokumentti on kyseessä. Rivillä kaksi määritellään xml sanasto eli standardi, jonka mukaan koodia luetaan. Rivillä neljä head-tagissa osoitetaan sivustolle, josta metataprofiili haetaan. Metataprofiilissa määritellään avainsanoja, joiden avulla hakukoneet löytävät sivuston. Oletettavasti uusilla hakukoneilla on kuitenkin omat kehittyneemmät algoritminsa, jotka luottavat enemmän sivustolla olevaan sisältöön kuin avainsanoihin. Rivillä viisi määritellään metatiedoissa hakukoneita varten esim. sivuston tyyppi ja merkistö. Rivillä seitsemän annetaan sivustolle nimi, joka näkyy selaimen yläreunassa. Rivillä yhdeksän kutsutaan tyylitiedostoa ja rivillä 10 määritellään päivitysilmoitusten URL-osoite. Rivillä 11 kutsutaan WordPressin omaa funktiota, jonka avulla ladataan esimerkiksi lisäosien JavaScript-koodia headeriin. Riviltä 14 alkaa markkinointisivustoon liittyvä HTML-koodi.

```

1 | <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 | <html xmlns="http://www.w3.org/1999/xhtml" <?php language_attributes(); ?>
3 |
4 | <head profile="http://gmpg.org/xfn/11">
5 | <meta http-equiv="Content-Type" content="<?php bloginfo('html_type'); ?>; charset=<?php bloginfo('charset'); ?>" />
6 |
7 | <title><?php wp_title('alaquo:', true, 'right'); ?> <?php bloginfo('name'); ?></title>
8 |
9 | <link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>" type="text/css" media="screen" />
10 | <link rel="pingback" href="<?php bloginfo('pingback_url'); ?>" />
11 | <?php wp_head(); ?>
12 | </head>
13 |
14 | <body>
15 | <div id="container1">
16 |   <div id="header_logo">
17 |     <a href=".." index.php"><div id="logo_link"></div></a>
18 |   </div>
19 |
20 |   <div id="header_login_bg">
21 |
22 |     <div id="header_fields">
23 |       <form>
24 |         <div id="username"><input class="textfield" type="text" value="Username" /> </div>
25 |         <div id="password"><input class="textfield" type="text" value="Password"/></div>
26 |         <div id="button"><a href=""></a></div>
27 |       </form>
28 |     </div>
29 |   </div>
30 |
31 | </div>
32 |

```

Kuva 19. WordPress sivuston header.php ennen sovellussivustoon yhdistämistä.



Kuva 20. WordPress sivuston header.php visuaalisesti.

Kuva 21 esittää index.php tiedoston koodirakennetta. Kuva 22 esittää koodin visuaalissa muodossa, kun ei huomioida header ja footer kutsua. Index.php tiedosto on niin kutsuttu pääsivu. Kaikki markkinointisivuston sivujen sisällöt toimivat tämän tiedoston pohjalta. Jokaisella sivuston sivulla on oma tunnuksensa, jonka avulla index.php tiedosto tietää, mitä sisältöä tulee näyttää. Index.php tiedosto koostuu markkinointisivun navigaatiosta ja sisältöalueesta. Rivillä 32 kutsutaan header.php tiedostoa WordPressin omalla PHP-komennolla. Rivillä 39 kutsutaan linkkejä artikkelin nimen mukaan. Tässä tapauksessa linkeiksi listataan artikkelien nimet, jotka löytyvät hallintasivulta Pages-välilehden alta. Pages tarkoittaa WordPressissä ns. staattisia sivuja. Navigaatiossa ei ole huomioitu blogiartikkeleiden näyttämistä, sillä markkinointisivustolla ei ole tarkoitus käyttää blogiominaisuutta. Blogiartikkeleja kutsutaan järjestelmässä nimellä Posts ja niiden kutsumiseen on oma komentonsa. Rivillä 56 aloitetaan PHP:n if-lause ja rivillä 57 while loop. If- lauseella kysytään onko artikkelia, jos on aloitetaan while loop. Loopin avulla näytetään artikkelien sisältö. Samaa looppia voitaisiin käyttää myös blogikirjoitusten näyttämiseen. Rivillä 62 mahdollistetaan tekstin katkaisu ja lue lisää linkki, mutta toimintoa voidaan käyttää ainoastaan blogikirjoituksissa. Rivillä 65 lopetetaan while loop ja seuraavalla rivillä if-lause. Rivillä 68 mahdollistetaan linkki, jonka avulla WordPressiin kirjautunut voi siirtyä muokkaamaan kyseistä artikkelia. Jos kommentointi on sallittu, näytetään Rivillä 71 kommentit ja kommentin lähetykentät. Rivillä 74 kutsutaan footer.php tiedostoa.

```

32 <?php get_header();?>
33 <div id="container2">
34     <div id="navigation_bg">
35         <div id="navigation">
36             <div id="navigation_left">
37                 <ul>
38                     <li>
39                         <?php wp_list_pages('title_li='); ?>
40                     </li>
41                 </ul>
42             </div>
43             <div id="navigation_right">
44                 <ul>
45                     <li>
46                         <a href="">Register</a>
47                     </li>
48                 </ul>
49             </div>
50         </div>
51     </div>
52
53     <div id="content">
54
55
56         <?php if (have_posts()) : ?>
57         <?php while (have_posts()) : the_post(); ?>
58
59             <div class="post">
60                 <div class="entrytext">
61
62                     <?php the_content('<p class="serif">Read the rest of this page &raquo;</p>'); ?>
63                 </div>
64             </div>
65             <?php endwhile;?>
66             <?php endif; ?>
67             <div id="edit_entry">
68                 <?php edit_post_link('Edit this entry.', '<p>', '</p>'); ?>
69             </div>
70
71             <?php comments_template(); ?>
72         </div>
73     </div>
74 <?php get_footer(); ?>

```

Kuva 21. WordPress sivuston index.php ennen sovellussivustoon yhdistämistä.



Who are we?

We are a small creative agency based in Finland designing and producing high quality customer & client publications. These publications have a wide reader base from North America to Europe and Asia.

1 Register

2 Select product

3 Upload PDF

What is it?

ViewMe & ViewMeOnline together provide a **FREE!** digital publishing service we offer to our existing clients.

We provide a quick and efficient digital publishing solution so you can read your print publications interactively online. See our **Demo** or visit our **Examples** page for more online publications.

If you're **NOT** one of our existing customers then don't worry, we are now offering this popular service at a minimal cost so you too can provide this service to your readers today!

Why use it?

The way we read media online is changing rapidly due to reading devices such as Amazon's Kindle and Apples iPad. Readers now expect that they can read your publications online, in interactive digital format.

- Moderate pricing
- Hosting solutions to suit your needs
- No page limitations
- Add your company logo
- Customize colours to fit your brand
- **No advertising!**

Register now and upload your files today! [Register](#)

Kuva 22. WordPress sivuston index.php visuaalisesti.

Footer.php tiedosto sisältää kutsun, jonka avulla WordPress tunnistaa sivuston alaosaksi. Tiedostossa kutsutaan myös päänavigaation linkkejä samalla komennolla kuin index.php sivulla. Virhesivu 404.php on kopio index.php tiedostosta, johon on lisätty ilmoitustekstiksi: sivua ei löydy. Comments.php tiedosto sisältää ainoastaan WordPressin määrittelemiä toimintoja kommenttien näyttämiseen. Tätä tiedostoa ei välttämättä tarvitsisi, sillä sivustolla ei todennäköisesti sallita artikkelien kommentointia.

6.2 Sovellussivuston ulkoasun päivittäminen

PHP-pohjaista sovellussivustoa voitiin päivittää WordPress-järjestelmän tapaan XAMPPin avulla. Olin heti projektin alussa kopioinut asiakkaalta sovellussivuston lähdekoodit ja MySQL-tietokannan, mutta ulkoasupäivityksen alkaessa syvennyin sovellussivuston toimintaan ja toteutukseen.

Sovellussivuston pääkansio sisälsi kolme kansiota: css, images ja lib. Css kansiossa oli sivuston css-tyylitiedosto, images-kansiossa sivustoon liittyvät kuvat ja lib-kansiossa komentoja sisältävät tiedostot funktions.php ja java.js. Pääkansion juuressa oli seuraavat tiedostot:

- activate.php – sisälsi toiminnallisuuden, joka aktivoi rekisteröityneen käyttäjän
- activation.html – yhdistyi activate.php tiedostoon ja sisälsi ulkoasun onnitteleville käyttäjän aktivoitessa tilinsä
- admin.php – ylläpitäjän sivu
- demo.html – verkkojulkaisun demo
- foot_bar.html – sivuston alaosa, joka oli tyhjä tiedosto
- header.html – sivuston yläosan kuva
- index.php – näyttää kirjautuneille käyttäjätilin tietoja
- info.html – etusivu ei kirjautuneille käyttäjille, jossa yksi lause tekstiä
- info_log.html – etusivu kirjautuneille käyttäjille, jossa yksi lause tekstiä
- login_bar.php – kirjautumisen toiminnallisuus ja ulkoasu
- menu.html – linkit
- register.php – rekisteröitymisen toiminnallisuus ja ulkoasu
- sessions.php – yhteydenotto tietokantaan
- settings.php – käyttäjätilin tietojen toiminnallisuus ja ulkoasu
- upload_banner.php – tiedoston lataussivu
- uploaded_files.php – käyttäjän ladatut tiedostot-sivu

Koko sivusto oli toteutettu kokonaan nykystandardin vastaisesti taulukkoasettelulla, joten korvasin sivuston div-elementeillä. Taulukkoasettelun jätin ainoastaan kohtiin, joissa tieto oli tarkoitettu esitettävän taulukkomuodossa tai kohtiin, joissa muutokset olisivat vieneet kohtuuttomasti aikaa. Tosin julkaisujärjestelmissä käytetään usein taulukoita sisältöalueella tiedon esittämiseen, sillä se on joissakin tapauksissa helpoin tapa jäsenellä tekstiä ja kuvia editorissa. Sovellussivustolla oli PHP-funktioille oma tiedosto lib-kansiossa, mutta silti suuriosa toiminnoista oli kirjoitettu suoraan sivurakenteeseen sekaan. Pyrin poistamaan sivustolta turhat sivut ja jaottelemaan sisällön kolmeen osaan kuten WordPressissä. Tämä ei kuitenkaan ollut täysin mahdollista, sillä en voinut puuttua sivustojen rakenteeseen kovinkaan paljon rikkomatta toiminnallisuutta.

Muutoksieni jälkeen sivuston sivut rakentuivat seuraavista tiedostoista: activate.php, activation.html, admin.php, header.php, index.php, login_bar.php, menu.php, register.php, sessions.php, settings.php, upload_banner.php ja uploaded_files.php. Lib-kansion tiedostot functions.php ja java.js säilytettiin myös. Kansiot css ja images poistettiin, sillä tyylit ja sivuston ulkoasuun liittyvät kuvat haettiin WordPressiin tehdystä Viewme_template kansioista. Sovellussivuston kansioon luotiin kansiot button_images, joka sisälsi kirjautumisnapit ja email_images, joka sisälsi sähköpostiviesteissä käytettävän logon kuvan.

Header.php tiedosto sisältää HTML-sivun vaatimien alkumääritysten lisäksi joitakin sovellussivuston toiminnallisuuksia. Esimerkiksi ensimmäisellä rivillä kutsutaan sessions.php tiedostoa, jossa määritellään yhteydenotto tietokantaan. Toisella rivillä taas functions.php:ta. Sivurakenne on katkaistu yläosan jälkeen samalla tavalla kuin WordPressin Viewme_template teemassa. Header.php sisällyttää myös kirjautumiskentille tarkoitettuun kohtaan login_bar.php tiedoston, joka näyttää kyseiset kentät ja hoitaa sovellukseen kirjautumisen.

Index.php tiedosto on rakenteeltaan samanlainen kuin WordPressin index.php. Se sisältää navigaation ja sisältöalueen. Tiedostossa on myös määritelty mitä tietoja sisältöalueella näytetään, kun käyttäjä on kirjautunut sisään. Index.php sisällyttää menu.php tiedoston, kun sovellukseen on kirjaututtu. Menu.php sisältää linkit asetuksiin (settings.php), PDF-tiedostojen lataukseen (upload_banner.php) ja omiin ladattuihin tiedostoihin (uploaded_files.php). Ylläpitäjätunnuksella kirjautunut näkee admin.php tiedoston sisällöt esim. sivuston käyttäjät. Kirjautumattomalle näytetään ainoastaan rekisteröintisivu.

6.3 Sivustojen yhdistäminen yhdeksi kokonaisuudeksi

Ensimmäiseksi WordPressin pääkansion juureen luotiin users-niminen alikansio sovellussivuston tiedostoille. Kun markkinointisivuston osoitteesta www.viewme.info siirrytään sovellussivustolle, osoitekentässä lukee www.viewme.info/users. Käyttäjä ei tällöin huomaa sivuston vaihtumista osoitekentästä.

Ensimmäinen haaste sivuston yhdistämisessä oli kutsua WordPressin header.php tiedostossa sovelluksen kirjautumiskenttä. Kuva 23 esittää kuinka WordPressin header.php tiedostossa kutsutaan sovellussivustolta kirjautumiskenttää. Kuva 24 esittää

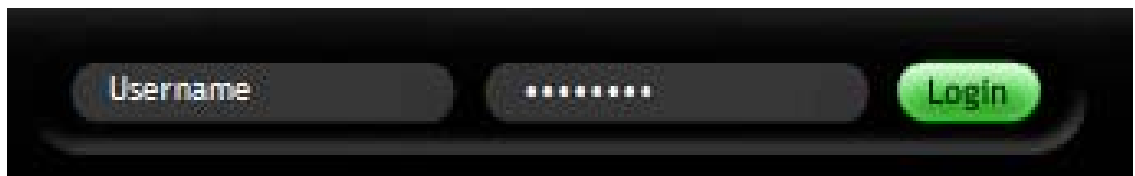
visuaalisesti kohdan, joka tuodaan WordPress-järjestelmään. Rivillä yksi kutsutaan users-kansiosta sovellussivuston sessions.php tiedostoa, jossa on yhteydenottomäärittökset sovelluksen tietokantaan. Tietokannassa ovat tiedot käyttäjätunnuksista ja salasanoista. Sovelluksen tietokannalla ei ole yhteyttä WordPressin tietokantaan, joten sovelluksen tunnuksilla ei voi kirjautua WordPressiin. Kyseessä on kaksi eri järjestelmää ja kaksi eri tietokantaa käyttäjineen. Rivillä 23 sisällytetään login_bar.php kirjautumiskentille tarkoitettuun kohtaan. Vastaavasti sovelluksen header.php tiedostossa kutsutaan samassa kohdassa login_bar.php tiedostoa, tiedostopolku on ainoastaan lyhyempi.

```

1 <?php include(dirname(__FILE__) .'/../../../../../users/session.php'); ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml" <?php language_attributes(); ?>>
4 <head profile="http://gmpg.org/xfn/11">
5 <meta http-equiv="Content-Type" content="<?php bloginfo('html_type'); ?>; charset=<?php bloginfo('charset'); ?>" />
6
7 <title><?php wp_title('slaguo;', true, 'right'); ?> <?php bloginfo('name'); ?></title>
8 <link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>" type="text/css" media="screen" />
9 <link rel="pingback" href="<?php bloginfo('pingback_url'); ?>" />
10 <?php wp_head(); ?>
11 </head>
12
13 <body>
14 <div id="container1">
15     <div id="header_logo">
16         <!-- This is link to the homepages by viewme logo -->
17         <a href="index.php"><div id="logo_link"></div></a>
18     </div>
19
20     <div id="header_login_bg">
21         <div id="header_fields">
22             <!-- This include loginbar from viewme PHP-site -->
23             <?php include(dirname(__FILE__) .'/../../../../../users/login_bar.php'); ?>
24         </div>
25     </div>
26 </div>
27 </body>

```

Kuva 23. WordPressin header.php tiedosto ja sisällönhaku sovellussivustolta.



Kuva 24. Sovelluksesta tuodut kirjautumiskentät visuaalisesti.

WordPressin index.php tiedostossa tarvittiin yhdistämistä vain navigaatioissa, sillä jos käyttäjä siirtyy sovellussivustolta kirjautumisensa jälkeen takaisin WordPressin puolelle, on hänelle näytettävä kirjautuneen käyttäjän linkit edelleen. Rivillä 23 tarkastetaan käyttäjän kirjautumisen tila. Jos käyttäjä ei ole kirjautunut sisään, näytetään rekisteröidy-linkki. Jos käyttäjä on kirjautunut sisään, piilotetaan rekisteröidy-linkki ja tilalle sisällytetään rekisteröityneen käyttäjän linkit sovellussivuston menu.php tiedostosta. Kuva 25 esittää WordPressin index.php tiedostoa, kun sisältöä haetaan sovellussivustolta.

```

1  <?php get_header();?>
2  <div id="container2">
3      <div id="navigation_bg">
4          <div id="navigation">
5              <div id="navigation_left">
6                  <ul>
7                      <li>
8                          <!-- This get links by Pages-->
9                          <?php wp_list_pages('title_li='); ?>
10                     </li>
11                 </ul>
12             </div>
13             <div id="navigation_right">
14                 <!-- If login or register link, li tag get activelink style -->
15                 <ul><?php $active = "";
16
17                     if(isset($_GET["pag"]))
18                     {
19                         $active = $_GET["pag"];
20                     }
21                 <?>
22                 <!-- If login, register link is hidden-->
23                 <?php if(!isset($_SESSION['user'])){ ?><li <?php if($active == "register")
24                     echo 'class="activeLink" ';?><a href="../../users/index.php?pag=register">Register</a></li><?php }?>
25                 <!-- If login, show loginlinks from viewme PHP-site -->
26                 <?php include(dirname(__FILE__) .'/../../users/menu.php'); ?></ul>
27             </div>
28         </div>
29     </div>
30 </div>
31 <div id="content">
32     <!-- This get contents of post and Pages -->
33
34     <?php if (have_posts()) : ?>
35     <?php while (have_posts()) : the_post(); ?>
36
37         <div class="post">
38
39             <h2 id="post-<?php the_ID() ?>"><?php /* the_title(); */?></h2>
40             <div class="entrytext">
41
42                 <?php the_content('Read on...'); ?>
43             </div>
44         </div>
45     <?php endwhile;?>
46     <?php endif; ?>
47     <div id="edit_entry">
48     <?php edit_post_link('Edit this entry.', '<p>', '</p>'); ?>
49     </div>
50
51
52     <!-- This get comments -->
53     <?php comments_template(); ?>
54 </div>
55 </div>
56 <?php get_footer(); ?>

```

Kuva 25. WordPressin index.php ja sisällönhaku sovellussivustolta.

Kun WordPressiin saatiin yhdistettyä toiminnot sovellussivustolta, oli haasteena saada sovellussivustolla näkymään WordPressin päänavigaation linkit. Muutoin vasemmalla olevat linkit häviäisivät kirjautuessa sovellukseen tai siirryttäessä rekisteröintisivulle. Tästä käyttäjä olisi voinut hämmentyä ja käsittää, että kyseessä on kaksi eri sivustoa. Yhtenä vaihtoehtona olisi ollut sijoittaa sovellussivustolle linkit staattisesti, mutta tällöin asiakkaan olisi päivitettävä linkkien nimet kahteen eri järjestelmään. Tämänkaltaisen toteutuksen oli vienyt ideaa dynaamiselta sivustolta. Ongelmaan löytyi kuitenkin helppo ratkaisu. Sovellussivuston index.php sivun ensimmäiselle riville sijoitettiin `require('../../wp-blog-header.php');`-komento. Tämän rivin avulla WordPressin omat komennot pystytään tulkitsemaan sivustolla, joka ei pohjautu WordPress teemaan. Sovellussivustolla pystyttiin kutsumaan päänavigaatiota WordPressin puolelta `<?php wp_list_pages('title_li='); ?>`-komennolla. Myös sovellussivustolle pystyttiin hakemaan sivuston alaosa WordPressin puolelta `<?php get_footer(); ?>`-komennolla, jol-

loin esim. yhteystietojen päivitys helpottuu, kun muutokset voidaan tehdä yhdestä paikasta eli WordPressin puolelta.

Sovellussivuston koodilliseen rakenteeseen ja PHP-funktioihin oli tehtävä joitakin muutoksia. Yksi tällainen muutos oli käyttäjän siirtäminen tietyissä tilanteissa sovellussivustolta markkinointisivuston pääsivulle eli WordPress-järjestelmän puolelle. Näin tuli tehdä käyttäjän kirjautuessa ulos, peruuttaessa rekisteröitymisensä, rekisteröitymisensä jälkeen tai kirjoittaessaan osoitekenttään suoraan kirjautuneelle käyttäjälle tarkoitetun sivuston esim. www.viewme.info/users/index.php?pag=settings. Tämä osoite veisi kirjautuneen käyttäjän hänen omiin asetuksiinsa. Jos ohjausta ei tehtäisi sovellussivustolta Word-Pressin puolelle, kirjautumaton käyttäjä näkisi tyhjän sivun sovellussivuston puolella. Päivitys ei ollut kovin yksinkertainen, sillä sovellussivuston koodiin täytyi tehdä monia pieniä ja vähän suurempia muutoksia. Pienet muutokset saattoivat aluksi vaikuttaa yksinkertaisilta, mutta sitä ne eivät olleet. Sillä ohjelman logiikka ei ollut keskitetty, vaan se oli pieninä palasina eri dokumenteilla, jolloin pienikin muutos saattoi johtaa toiminnallisuuden rikkoutumiseen ja suurempiin muutoksiin.

6.4 WordPress-järjestelmään lisäosien valinta

Valintamessamme lisäosia sivustolle, pohdimme asiakkaan tarpeita tiedon esittämiseen. Lähtökohtana oli kuvien, esimerkki julkaisuiden esittäminen ja tekstin palstointus. Tietoturvan parantamiseksi WordPressiin asennettiin muutama muukin lisäosa, joita käsitellään luvussa 6.6.

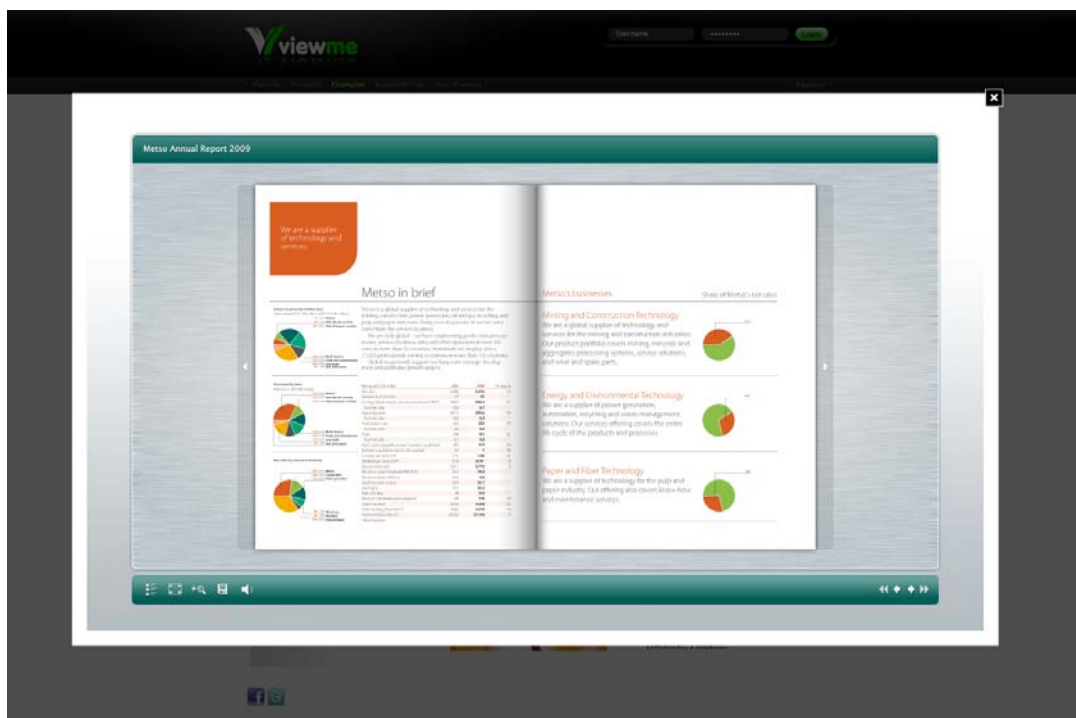
Asiakas halusi etusivulle lisäosan, jonka avulla pystyisi näyttämään useita kuvia diaesityksenä tietyllä viiveellä. Löysimme Vslider-nimisen lisäosan, joka mahdollistaa 10 erilaisen kuvan esittämisen. Lisäosan asennustiedosto ladattiin koneelle, purettiin WordPressin Plugins-kansioon ja aktivoitiin hallintasivulta Plugins-välilehdeltä. Etusivun artikkelin alkuun sijoitettiin lisäosan vaatima [vslider]-komento. Asennuksen jälkeen Vsliderin asetukset löytyivät omalta välilehdeltä WordPressin hallintasivulla. Asetuksista määriteltiin kuva-alueen koko, vaihtumisnopeus, vaihtumisanimaatio, kuvien osoitteet ja mahdolliset linkit. Kuva 26 esittää Vsliderin asetukset-välilehteä.

Kuva 26. Vslider asetusten-välilehti.

Sivuston palstoitukseen käytettiin etusivulla WP Columnize -lisäosaa. Lisäosa vaatii tyyli tiedostoon muutaman määrittelyt palstan leveydelle. Artikkelissa palsta aloitetaan komenolla [column] ja päätetään komenttoon [/column]. Palstat olisi voinut tehdä myös manuaalisesti joko taulukoina tai div-asettelulla. Lisäosa vähentää koodin kirjoitusta artikkelissa ja auttaa ylläpitäjää hahmottamaan, missä kohdin palstoitusta käytetään.

Asiakas halusi sivuston verkkojulkaisuesimerkkien avautuvan lightboxissa, joten kolmanneksi lisäosaksi valittiin WordPress Multibox PlugIn. Lisäosa osaa näyttää kuvia, Flashiä, Realmediaa, Windows Mediaa, MP3 musiikkitiedostoja, HTML sivuja ja PDF-tiedostoja dynaamisessa ja animoidussa ikkunassa. Lisäosa asennettiin luvun 4.7 ohjeistuksen mukaan. Lisäosan asetukset löytyvät WordPressin hallintasivulta Settings-välilehden alta kohdasta Multibox. Asetuksista voidaan valita esim. esityksen taustan tai reunan värit. Multiboxin asetuksissa linkeille voidaan määrittellä tyyli-

dostoa varten arvo, jotta kaikki linkit sivuston linkit eivät avaudu Multiboxissa. Määrittelin arvoksi multi. Artikkelissa verkkojulkaisuesimerkki voidaan pyytää avautumaan Multiboxissa näin: ` Tässä demo `. HTML-kielessä a-tag tarkoittaa linkkiä. Tyyli tiedoston tiettyä tyyliä voidaan kutua class-attribuuttivalitsimen avulla. Anoin a-tagissa linkin class-arvoksi multi. Multiboxin koko määritellään kohdassa `rel="width:1200,height:700"`. Tässä tapauksessa leveys on 1200 ja korkeus 700. Kuva 27 esittää kuinka verkkolehtijulkaisu avautuu sivustolla Multiboxissa.



Kuva 27. Verkkolehtiesimerkin esittäminen Multibox-lisäosassa.

Sivustolle asennettiin Share and Follow -lisäosa, jonka avulla käyttäjä voi jakaa viewme-sivun linkin Facebookiin tai Twitteriin. Lisäosa tukee myös linkin lisäämistä stumble, digg, reddit, hyves, delicious, orkut ja myspace sovelluksiin. Sovelluksien kuvakkeet näkyvät jokaisen sivun alareunassa lisäosan aktivoinnin jälkeen. Lisäksi kuvakkeet voidaan toistaa vasemmassa tai oikeassa reunassa.

6.5 WordPress-järjestelmän asetusten määrittäminen

Asetukset käytiin läpi luvun 4.4 mukaan. Yleisistä asetuksista määriteltiin sivuston nimeksi ViewMe, sivuston kuvaus, sivuston sijaintiosoitteet, ylläpitäjän sähköpos-

tiosoite ja estettiin rekisteröityminen sivustolle. Tämä rekisteröityminen koski siis WordPressin puolella toimivaa markkinointisivustoa. Non-Stop Studiot Oy:n asiakkailla ei ole tarvetta rekisteröityä WordPress-järjestelmässä toimivalle markkinointisivustolle. Verkkojulkaisusovellukseen kirjautuminen toimii siis aivan eri järjestelmästä, jolloin sovellukseen ei voida vaikuttaa WordPressin hallintasivulta.

Kirjoitus- ja keskusteluasetuksilla ei ollut merkitystä, sillä kommentointia tai blogikirjoituksia ei markkinointisivustolla sallittu. Lukuasetuksista määriteltiin etusivuksi ns. staattisen sivun artikkeli. Media- ja sekalaiset-asetukset säilytettiin oletuksena. Yksityisyysasetuksissa sallittiin sivuston näkyminen hakukoneille. Staattiset linkit-asetuksissa määriteltiin ohjeen mukaan ”Custom Structure” kentälle arvoksi %post-name%. WordPress-sivujen linkit saivat osoitteekseen artikkelin nimen. Nimi määritellään artikkelin otsikon alapuolella olevaan permalink-kenttään. Luvun 4.5 käyttäjäasetuksissa määriteltiin vain yksi pääkäyttäjä. Myöhemmin käyttäjiksi voidaan lisätä yrityksen muita jäseniä.

6.6 WordPressin tietoturvan parantaminen

Luvun 4.8 ohjeistuksen mukaan WordPressin tietoturvaa pyrittiin parantamaan käyttöönoton yhteydessä. Tietoturvasyistä asennuksen yhteydessä ladattiin ohjelmiston uusin versio 2.9.2. Luvussa 4.8 suositellaan etuliitteen vaihtamista WP Prefix Table Changer -lisäosalla ja kirjoitusoikeuksien tarkastamista WP Security Scan -lisäosalla. WP Security Scan-lisäosan viimeisimmästä versiosta löytyi myös tietokannan etuliitteen vaihto-ominaisuus. Etuliitettä ei tarvinnut vaihtaa lisäosalla asennuksen jälkeen, sillä jo tuotantoon asennuksen yhteydessä etuliite määriteltiin toiseksi. WP Security Scan-lisäosalla tarkastattettiin sivuston muu turvallisuus.

Ohjeistuksen mukaan asennustiedosto wp-config.php siirrettiin toiseen kansioon. Kansiodien selaaminen on WordPressin versiossa 2.9.2 estetty lisäämällä tyhjä index.html-tiedosto esim. wp-content/plugins-kansioon. Aikaisemmin tämä olisi pitänyt tehdä itse. Myös admin-käyttäjän oletusnimi muutettiin toiseksi ja salasana vaihdettiin.

6.7 Teknisen toteutuksen ongelmat

Markkinointisivuston kokoaminen WordPressin puolelle sujui lähes ongelmitta. Ainoat ongelmat liittyivät käyttökokemattomuuteen järjestelmän suhteen. Testattaessa sivuston perusrakenteen toimivuutta, huomattiin ongelmia Internet Explorer 6:ssa. Ongelmat liittyivät selain version tapaan käsitellä CSS-tyylejä. Muutamilla muutoksilla sivuston perusrakenne saatiin toimimaan, niin hyvin kuin IE 6:ssa se on mahdollista.

Sovellussivuston ulkoasun päivityksessä ongelmia aiheutti jäsentämätön toiminnallisuus. Pienet muutokset vaativat suhteessa suuria muutoksia. Tästä syystä jouduin myös pohtimaan oman opinnäytetyön rajausta, kuinka suuria muutoksia olen valmis tekemään saadakseni käyttöliittymän toimimaan täydellisesti?

Suurimpana haasteena oli ratkaistava miten kahden eri sivuston osat yhdistetään. Yhtenä keskeisenä asiana oli uudelleenohjaus markkinointisivuston etusivulle esimerkiksi tilanteessa, kun käyttäjä ei ole kirjautuneena, mutta pyrkii kirjautuneen käyttäjän sivulle.

Sovellussivuston koodista johtuen muutama ongelma jäi ratkaisematta esim. aktiivista linkkiä kuvaava vihreä väri, ei toimi joka toiminnon jälkeen. Admin-sivustolla tiedostojen poistoon tarkoitettussa delete-linkissä kursori on vääränlainen, tekstiä ei tunnista linkiksi. Sivuston ilmoitukset näytetään varoitusikkunassa (engl. alert box). Esimerkiksi Internet Explorerissa on varoitusikkunassa keltainen varoituskolmio. Sivustolla oli joitakin kirjoitusvirheitä, joita pyrin korjaamaan huomattessani. Salasanaan käy ainoastaan pienet kirjaimet ja numerot 6-8 merkkiä. Jos käyttäjä yrittää luoda käyttäjätilin samalla sähköpostiosoitteella, sivusto ei ilmoita virheestä. Kirjautuessa sovellukseen, väärän käyttäjätunnuksen tai -salasanan syöttö antaa virheilmoituksen kuten pitääkin. Mutta jos käyttäjä yrittää kirjautua sovellukseen joltakin muulta kuin etusivulta, käyttäjä ohjautuu virheilmoitusta varten etusivulle. Vaikka asiat kuulostavat yksinkertaiselta ja helpolta korjata, eivät ne sitä ole. Koodin rakenteeseen jouduttiin tekemään jo hyvin paljon muutoksia, jotta asiat toimisivat näinkin hyvin. Olisi mahdotonta käytettävän ajan suhteen selvittää edellä mainittuja ongelmia, joiden selvittäminen vaatisi paljon uudelleen ohjelmointia, joka ei liity opinnäytteeni aihealajuokseen.

7 YHTEENVETO

Opinnäytetyöni kirjallinen osuus koostui neljästä osasta: julkaisujärjestelmien yleiskatsauksesta, perehtymisestä WordPress-järjestelmään, sivuston graafisesta suunnittelusta ja teknisestä toteutuksesta. Aiheesta kirjoittaminen oli haasteellista, sillä julkaisujärjestelmistä ei löytynyt kovin paljon painettua lähdeaineistoa, puhumattakaan WordPress-järjestelmästä. Ainoa WordPressiä käsittelevä kirja oli englanninkielinen opas WordPress teemojen kokoamisesta. Uskoisin kirjallisuuden vähyyden johtuvan julkaisujärjestelmien nopeasta kehityksestä, jolloin painovalmiiksi saatettu tieto saattaisi olla jo vanhentunutta. Graafiseen suunnitteluun liittyvää painettua kirjallisuutta löytyi todella paljon, jolloin eri lähteiden tietoa pystyi vertailemaan ja soveltamaan omassa toteutuksessa.

Opinnäytetyön produktiivin toteutuksessa oli haasteellista aikataulussa pysyminen, sillä asiakkaan oli hankala irrottautua omista työtehtävistä projektia varten. Graafisena alan ammattilaisena asiakas halusi osallistua tiiviisti sivuston suunnitteluun. Asiakkaalla oli hyvin vahvat näkemykset ja tavoitteet logon ja sivuston ulkoasusta. Tästä syystä sivuston visuaalisuutta työstettiin asiakkaalla, jolloin hän pystyi ohjeistamaan minua.

Sivuston teknisessä toteutuksessa oli haasteellisinta sovellussivuston päivittäminen ja sen yhdistäminen markkinointisivustoon. Tulevaisuutta ajatellen sovellussivuston kehittäminen saattaa olla haasteellista ja jopa mahdotonta. Yksi vaihtoehto olisi kehittää sovellussivuston pohjalta lisäosa WordPressiin. Tämän avulla WordPressin kirjautumista voisi käyttää myös sovelluksessa, ulkoasun päivittäminen olisi tehtävä vain yhteen järjestelmään ja tietoturvallisuus paranisi. Käytännössä tämä olisi hyvin iso projekti, sillä sovellussivusto olisi rakennettava koodillisesti uudelleen. Nykyisiä toimintoja tuskin pystyisi hyödyntämään suoraan.

Saavutin opinnäytetyössäni tavoitteeni, sillä kehityin teknisesti, opin uuden julkaisujärjestelmän, sain kokemusta logosuunnittelusta, graafisesta suunnittelusta ja pääsin tutustumaan työelämän realiteetteihin. Ymmärsin myös PHP-ohjelmointikielen tärkeyden ja tarpeen dynaamisten sivustojen kokoamisessa. Tavoitteenani on oppia PHP:tä paremmin, jotta pystyisin hyödyntämään sitä tulevissa verkkoprojekteissa. Yhteistyö asiakkaan kanssa sujui aikatauluongelmia lukuun ottamatta hyvin. Kaiken kaikkiaan koen onnistuneeni tavoitteissani opinnäytetyöni suhteen. Nähtäväksi jää millaista sisältöä asiakas sivustolle laittaa ja kuinka hyvin palvelu menestyy.

LÄHTEET

Ahola, H., Koivumäki, T. & Oinas-Kukkonen, H. 2002. Markkinointi, liiketoiminta ja digitaalinen media. Vantaa: WSOY.

CERT-FI. 23.7 2007. CERT-FI haavoittuvuustiedote 082/2007: Haavoittuvuus Joomla-julkaisujärjestelmässä. Saatavissa:
<http://www.cert.fi/haavoittuvuudet/2007/haavoittuvuus-2007-82.html> [viitattu 5.3.2010].

CERT-FI. 13.9.2007. Tietoturva nyt: WordPress-haavoittuvuuksiin julkaistu helppo-käyttöinen hyväksikäyttötyökalu. Saatavissa:
http://www.cert.fi/tietoturvanyt/2007/09/P_6.html [viitattu 5.3.2010].

CERT-FI. 29.4.2008. CERT-FI haavoittuvuustiedote 054/2008: WordPress-ohjelmistossa haavoittuvuuksia. Saatavissa:
<http://www.cert.fi/haavoittuvuudet/2008/haavoittuvuus-2008-54.html> [viitattu 5.3.2010].

CERT-FI. 23.8.2008. Tietoturva nyt!: Joomla-järjestelmiä vastaan hyökätään aktiivisesti. Saatavissa: <http://www.cert.fi/tietoturvanyt/2008/08/ttn200808231817.html> [viitattu 5.3.2010].

CERT-FI. 10.9.2008. Haavoittuvuustiedote 118/2008: WordPress 2.6.2 julkaistu. Saatavissa: <http://www.cert.fi/haavoittuvuudet/2008/haavoittuvuus-2008-114.html> [viitattu 5.3.2010].

CERT-FI. 6.11.2008. Tietoturva nyt!: Hyökkäyksiä Joomla! sisällönhallintajärjestelmiä vastaan. Saatavissa:
<http://www.cert.fi/tietoturvanyt/2008/11/ttn200811061617.html> [viitattu 5.3.2010].

CERT-FI. 6.8 2009. CERT-FI haavoittuvuustiedote 074/2009: WordPress-turvallisuuspäivitys. Saatavissa:
<http://www.cert.fi/haavoittuvuudet/2009/haavoittuvuus-2009-074.html> [viitattu 5.3.2010].

CERT-FI. 22.10.2009. CERT-FI haavoittuvuustiedote 103/2009: WordPress-blogialustan turvallisuutta parantava päivitys. Saatavissa:
<http://www.cert.fi/haavoittuvuudet/2009/haavoittuvuus-2009-103.html> [viitattu 5.3.2010].

CERT-FI. 24.2.2010. CERT-FI haavoittuvuustiedote 038/2010: WordPress-ohjelmistossa haavoittuvuus. Saatavissa:
<http://www.cert.fi/haavoittuvuudet/2010/haavoittuvuus-2010-038.html> [viitattu 5.3.2010].

Goto, K. & Cotler, E. 2003. Verkkopalveluprojekti. Helsinki: Edita Prima Oy.

Hatva, A. 2002. Verkkografiikka. Helsinki: Edita Prima Oy.

Heinisuo, R. & Rauta, I. 2007. PHP ja MySQL Tietokantapohjaiset verkkopalvelut. 4., uudistettu painos. Helsinki: Gummerus kirjapaino Oy.

Herttua, I. 2010. Amazon Kindle 2 -sähköinen kirja. Saatavissa:
<http://www.tekniikanmaailma.fi/kuva-ja-aani/testit-ja-pikakokeet/tm-pikakoe-amazon-kindle-2-sahkoinen-kirja> [viitattu 17.3.2010].

Huovila, T. 2006. Look: Visuaalista viestisi. Hämeenlinna: Karisto Oy:n kirjapaino.

Kirjavainen, J. 2009. Opinnäytetyö: Julkaisujärjestelmä yhdistystoiminnan tukena. Saatavissa. <https://publications.theseus.fi/browse-author-items?author=Kirjavainen,+Juho> [viitattu 27.1.2010].

Korpela, J. 2008. CSS verkkosivujen muotoilussa. 1. painos. Porvoo: WS Bookwell.

Kuutti, W. 2003. Käytettävyys, suunnittelu ja arviointi. Saarijärvi: Talentum.

Lipsanen, N. 2008. Mitä ovat paluuviihteet (Trackback)? Saatavissa:
<http://aiheet.domnik.net/ai-2008/03/paluuviite-trackback> [viitattu 1.3.2010].

- Pohjanoksa, I., Kuokkanen, E. & Raaska, T. 2007. Viesti verkossa. Digitaalisen viestinnän käsikirja. Juva: WS Bookwell Oy.
- Raninen, T. & Rautio, J. 2003. Mainonnan ABC. 1. Painos. Porvoo: WSOY.
- Rantala, A. 2002. PHP. 1. Painos. Porvoo: WS Bookwell.
- Salmela, J. 2002. Verkkosisällön hallinta. 1. Painos. Helsinki: Edita Prima Oy.
- Sinkkonen, I, Kuoppala, H, Parkkinen, J, Vastamäki, R. 2002. Käytettävyyden psykologia, Helsinki: Edita Prima Oy
- Tolvanen, P. 2008. Julkaisujärjestelmät Suomessa, markkinakatsaus 2008. Saatavissa: <http://vierityspalkki.fi/2008/03/31/julkaisujarjestelmat-suomessa-markkinakatsaus-2008/> [viitattu 22.1.2010].
- Virenius, M. 2008. WordPress-blogin tietoturva. Saatavissa: <http://www.arkimedia.fi/tietoturva/wordpress-blogin-tietoturva/> [viitattu 5.3.2010].
- W3C. 2009. Consortium. Saatavissa: <http://www.w3.org/Consortium/> [viitattu 1.3.2010].
- WordPress. 2007. WordPress Dangerous 2.1.1, upgrade to 2.1.2. Saatavissa: <http://wordpress.org/development/2007/03/upgrade-212/#more-199> [viitattu 5.3.2010].
- WordPress. 2010. About WordPress. Saatavissa: <http://wordpress.org/about/> [viitattu 15.2.2010].
- WordPress. 2010. Administration Panels. Saatavissa: http://codex.wordpress.org/Administration_Panels#Favorites_Menu [viitattu 1.3.2010].
- WordPress. 2010. Authors and Users SubPanel. Saatavissa: http://codex.wordpress.org/Users_Authors_and_Users_SubPanel [viitattu 4.3.2010].

WordPress. 2010. Editing wp-config.php. Saatavissa:
http://codex.wordpress.org/Editing_wp-config.php [viitattu 28.2.2010].

WordPress. 2010. Features. Saatavissa: <http://wordpress.org/about/features/> [viitattu 1.3.2010].

WordPress. 2010. Installing WordPress. Saatavissa:
http://codex.wordpress.org/Installing_WordPress [viitattu 15.2.2010].

WordPress. 2010. Plugins. Saatavissa: <http://codex.wordpress.org/Plugins> [viitattu 6.3.2010].

WordPress. 2010. Requirements. Saatavissa: <http://wordpress.org/about/requirements/>
[Viitattu 15.2.2010].

WordPress. 2010. Using Themes. Saatavissa:
http://codex.wordpress.org/Using_Themes [viitattu 6.3.2010].

Logokuvion kehitys

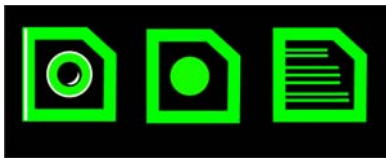
Silmä / lukija ruudukon päällä



Erilaisia silmiä



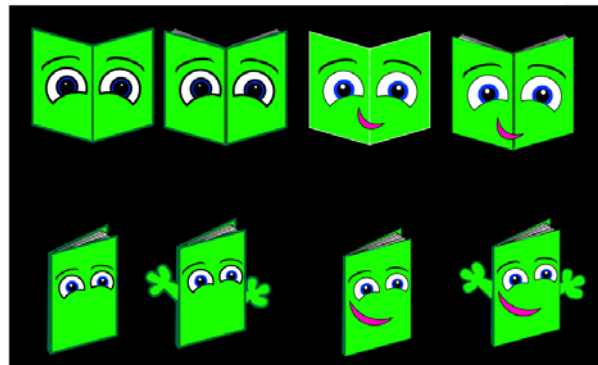
Sivuja



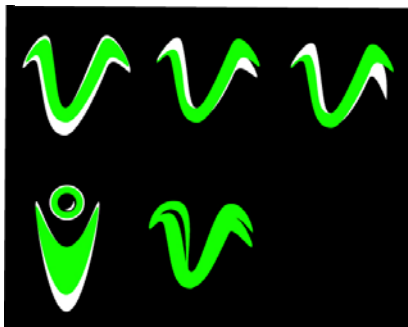
Kirjahylly



Veikeitä kirjoja



V-kirjaimia



Kirjamainen V-kirjain



Kirjamainen V-kirjain pelkistettynä



Muodon viimeistely



Logokuvioista valmiiseen logoon

Clearfac Gothic MT kirjaisintyyppi



Frankfurthe SH kirjaisintyyppi



Schueller BQ Bold kirjaisintyyppi



Vag Rounded kirjaisintyyppi



Valmis logo mustalla



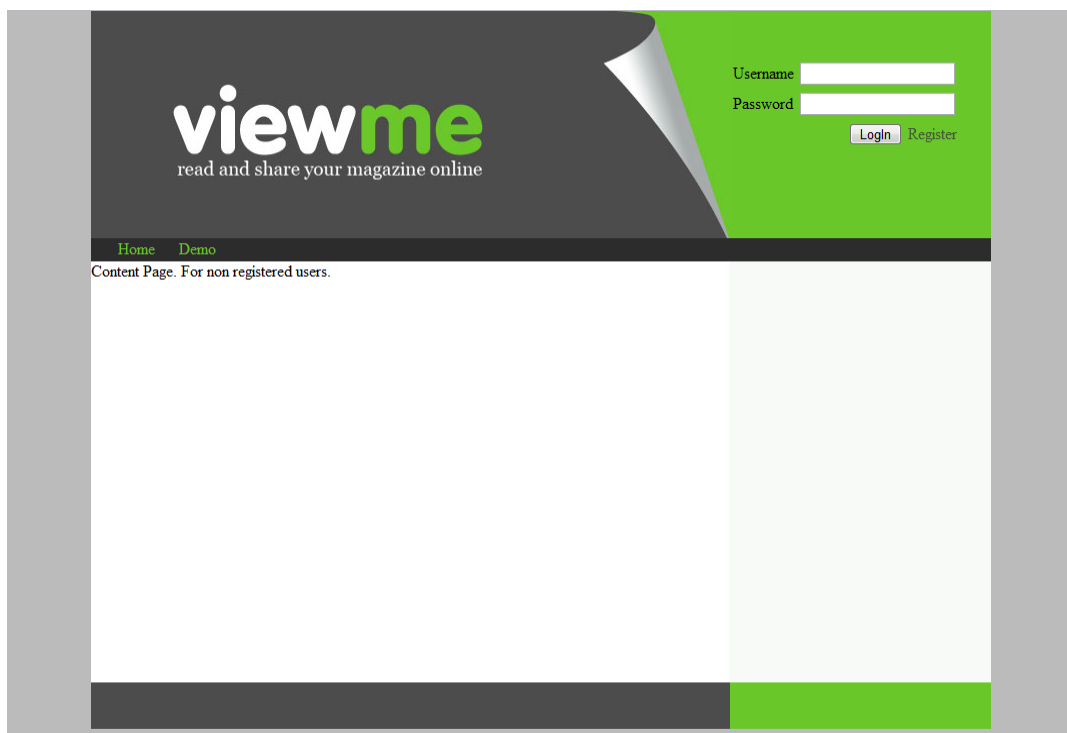
Valmis logo valkoisella



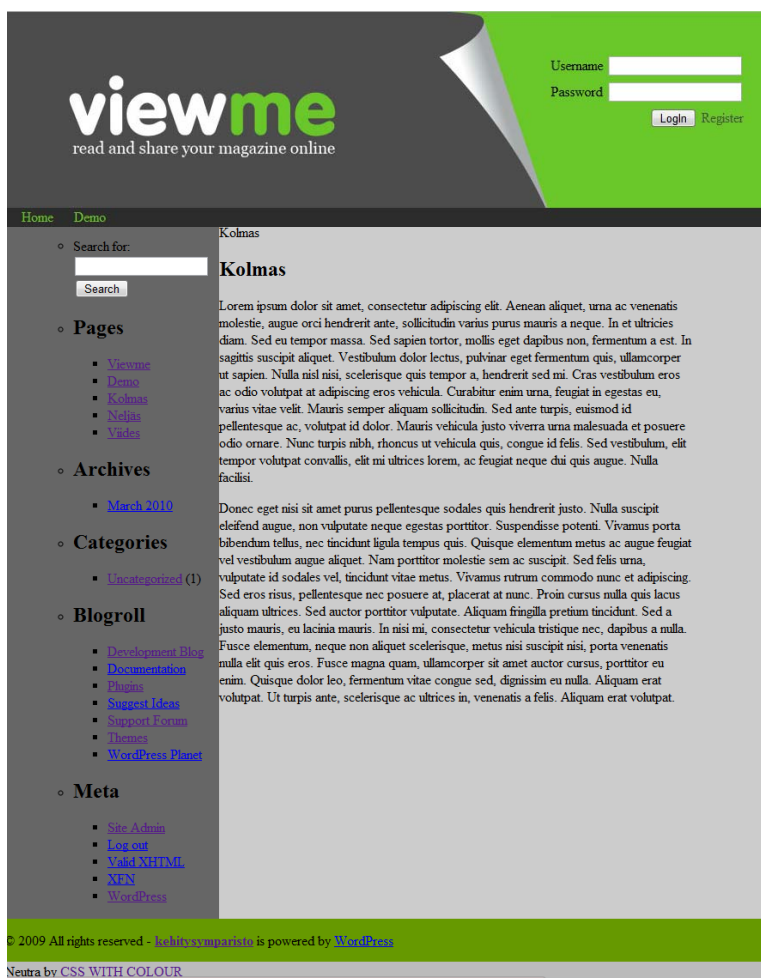
Logon mittasuhteet



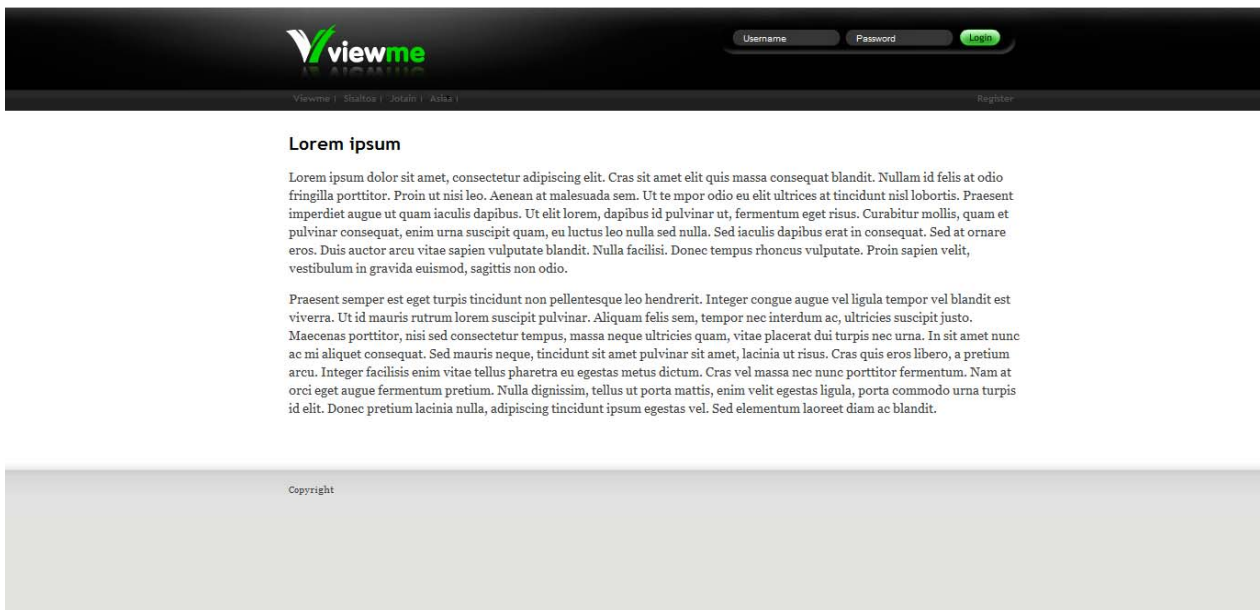
Entinen ViewMe:n etusivu



Ensimmäinen protosivu



Toinen protosivu



Uusi ViewMe:n etusivu

