

Antti Oinonen

VERKKOSIVUJEN TOTEUTUS SISÄLLÖNHALLINTA- JÄRJESTELMÄLLÄ

Opinnäytetyö
Tietotekniikan koulutusohjelma


Helmikuu 2013




MIKKELIN AMMATTIKORKEAKOULU

Mikkeli University of Applied Sciences

KUVAILULEHTI

 <p>MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences</p>		Opinnäytetyön päivämäärä 22.2.2013	
Tekijä(t) Antti Oinonen		Koulutusohjelma ja suuntautuminen Tietotekniikan koulutusohjelma	
Nimeke Verkkosivujen toteutus sisällönhallintajärjestelmällä			
Tiivistelmä <p>Tässä opinnäytetyössä tarkasteltiin sisällönhallintajärjestelmiä verkkosivujen hallinnointijärjestelmänä. Läpi käytiin, kuinka sisällönhallintajärjestelmät muodostuvat, mitä komponentteja ne tarvitsevat toimiakseen sekä mitä sisällönhallintajärjestelmiltä voi odottaa. Samalla tarkasteltiin missä tapauksissa sisällönhallintajärjestelmät toimivat paremmin, ja miksi tulisi käyttää sisällönhallintajärjestelmää verkkosivuja luotaessa. Pikaisesti tutkittiin, kuinka verkkosivut yleensä toimivat. Ohjelmointikielistä tarkasteltiin CSS:ää ja HTML:ää, ja kuinka niitä tulisi käyttää, ja mitä niillä saavutetaan.</p> <p>Läpi käytiin, kuinka Joomla sisällönhallintajärjestelmällä verkkosivujen toteuttaminen tapahtuu. Tämä toteutettiin lokaalina ratkaisuna. Työssä toteutettiin myös aiemmasta järjestelmästä tietojen tuonti uudempaan järjestelmään, tämä tehtiin yhteistyössä MHG Systems Oy:n kanssa. Aiempi versio siirrettiin uudemmalle julkaisulle, ja lisättiin haluttuja toimintoja uuteen julkaisuun käyttäen erinäisiä lisäosia Joomla kokoonpanossa. Käymme myös läpi Joomla toimintaperiaatteet ja luomme muutaman artikkelin tyhjiin kokoonpanoon alkuun pääsemiseksi. Tutustumme myös Joomla omiin toimintoihin, uppoutumatta sen syvemmin lisäosien tutkiskeluun.</p> <p>Toteutettiin kaksi erilaista ratkaisua sisällönhallintajärjestelmistä onnistuneesti. Toinen tehtiin valmiista tietokannoista ja siirrettiin uudelle julkaisulle ja toinen toteutettiin luomalla uusi kokoonpano. Samalla tutkittiin, miksi tulisi käyttää sisällönhallintajärjestelmää.</p>			
Asiasanat (avainsanat) Sisällönhallinta, WWW-sivut, CSS, HTML, Joomla			
Sivumäärä 35	Kieli suomi	URN	
Huomautus (huomautukset liitteistä) Liite 1			
Ohjaavan opettajan nimi Reijo Vuohelainen		Opinnäytetyön toimeksiantaja MHG Systems Oy	

DESCRIPTION

 <p>MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences</p>		Date of the bachelor's thesis 22 February 2013	
Author(s) Antti Oinonen		Degree programme and option Information Technology	
Name of the bachelor's thesis Website implementation via content management systems			
Abstract <p>This bachelor's thesis studied content managements systems as an administrative tool for a website. This covered how content management systems were built, what components they required to function and what can be achieved with them. This thesis explored why to choose content management system over traditional web development and what content management systems could offer. The thesis also covered the basic principles of website development. This included the basics of CSS and HTML programming.</p> <p>As a part of this study I installed one of the better known content management systems, Joomla, and demonstrated how to build a website with it. The work was done on a local host. I also studied how to do a migration from an existing website to a Joomla-based website. This was done in co-operation with MHG Systems Oy. We transferred an older publication to a newer one and added some functions to the site by using multiple extensions. This thesis also examined how to install Joomla content management system and how to create content with it. The thesis showcased the basic functions of Joomla-based web-site without too detailed study of the world of extensions.</p> <p>In this thesis two different websites were successfully made with Joomla content management system demonstrating the usefulness of content management systems. One was done to an empty build, while the other was a migration process from an existing website.</p>			
Subject headings, (keywords) Content management, website, CSS, HTML, Joomla			
Pages 35	Language Finnish	URN	
Remarks, notes on appendices Appendix 1			
Tutor Reijo Vuohelainen		Bachelor's thesis assigned by MHG Systems Oy	

SISÄLTÖ

1	JOHDANTO	1
2	VERKKOSIVUJEN RAKENTAMINEN.....	2
2.1	HTML	2
2.2	CSS	3
2.3	Verkkosivujen rakenne	4
3	SISÄLLÖNHALLINAJÄRJESTELMÄ ELI CMS	6
3.1	Sisällönhallintajärjestelmien käytettävyys ja toiminta periaate.....	7
3.2	CMS:n rakenne	8
3.2.1	Front end	10
3.2.2	Back end.....	11
4	JOOMLA.....	12
4.1	Joomla-migraatio ja päivitykset.....	13
4.2	Laajennukset	14
5	JOOMLA-SISÄLLÖNHALLINTAJÄRJESTELMÄN KÄYTTÖÖNOTTO.....	15
5.1	Alkuvaatimukset	16
5.2	Joomla kokoonpanon asentaminen tai päivittäminen	18
5.3	Joomla ohjauspaneeli.....	22
5.4	Joomla template ja CSS:n muokkaaminen	23
5.5	Käyttäjien hallinnointi	25
5.6	Sisällön luominen	27
5.7	Kielten lisääminen	30
5.8	Käyttämäni lisäosat.....	30
6	PALVELIMELLE MUUTTAMINEN.....	32
7	YHTEENVETO	32
	LÄHTEET	35

LIITE 1: Joomla:n järjestelmävaatimukset eri versioissa

1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on tutustua CMS-järjestelmiin (*Content Management System*, sisällönhallintajärjestelmä), joista erityisesti Joomla-järjestelmään, ja toteuttaa sen avulla verkkosivut MHG Systems Oy -yritykselle käyttäen Joomlaa verkkosivujen pohjana. Opinnäytetyön tekeminen aloitettiin vuoden 2012 loppupuolella. Teoriaosuus kattaa CMS-järjestelmien toimintaperiaatteet ja ajatuksia siitä, miksi tulisi valita CMS perinteisen verkkosivujen tekemisprosessin sijaan. Käytännön osuudessa toteutetaan MHG Systemsille verkkosivut, joihin luodaan pohjalle toimintoja helpottaakseen verkkosivujen hallinnointia. Yritykselle luodut verkkosivut toteutetaan päivittämällä aiemmasta alustasta, joten tukena luotiin ja käytiin läpi perustoimintoja myös tyhjään kokoonpanoon.

Joomla valittiin alustaksi sen takia että yrityksessä parilla henkilöllä oli jo aiempaa kokemusta Joomla-sisällönhallintajärjestelmästä. Osatekijänä oli varmasti myös Joomla'n vapaa ohjelmisto sekä sen avoin lähdekoodi, eli ohjelmiston pystyy saamaan ilmaiseksi ja sen muokkaaminen ja uudelleenkirjoittaminen on tarvittaessa mahdollista. Kun sisällönhallintajärjestelmä otetaan käyttöön, sivujen ylläpitäminen helpottuu ja hallinnointi vie jatkossa vähemmän aikaa. Teknisen osaamisen tarve vähentyy, käytettäessä sisällönhallintajärjestelmiä. Tämä johtuu siitä, että moni toiminto on tehty graafiseksi tai intuitiivisten käyttöliittymien taakse.

Käytännön osuus toteutetaan aluksi paikallisena (*local host*) versiona, minkä jälkeen se siirretään palvelimelle, jotta kaikki toiminnot saadaan testattua. Paikallisena versiona työskentelyssä muutosten tarkkailu ja niiden tekeminen on hiukan helpompaa, kun taas palvelimelle suoraan luodussa versiossa muutosten tekeminen ja tiedostojen lähettely vie oman aikansa. Siksi toteutetaan Joomla paikallisena. Muutosten tekeminen CMS:ään käsitellään, jotta se saadaan toimimaan palvelimella.

2 VERKKOSIVUJEN RAKENTAMINEN

Ennen kun voi alkaa hallinnoida verkkosivujen sisältöä, tulee siis tietää kuinka verkkosivut rakentuvat. Verkkosivut rakentuvat HTML:stä (*hypertext markup language*) tai XHTML:stä (*extended hypertext markup language*), sekä jonkinlaisesta tekstinmuotoilusta CSS (*cascading style sheets*), jotka valmiina muodostavat kokonaisuuden, jossa on muotoiltua tekstiä. Verkkosivuilla tarvitaan myös oma domain-nimi (verkkosoite) ja palvelin (fyysinen sijoituspaikka tiedostoille).

Useimmiten verkkosivut rakentuvat myös useammasta kuin yhdestä sivusta. On siis kotisivu (*homepage*), jolle päästään navigoimalla verkkosivujen osoitteeseen, joka on sivuston kannalta sen tärkein sivu. Aloitussivu auttaa sivustolla vierailevaa henkilöä muodostamaan mielipiteen sivustosta ja se on tärkeässä roolissa sivuston saavutettavuuden kannalta. Kotisivu usein määrittääkin onko sivuilla siis selkeä rakenne, ja kuinka helposti pääsee käsiksi haluttuun materiaaliin. Monesti sivuston aloitussivu toimii vain suurena valikkona, josta navigoidaan syvemmälle sivuston hierarkiaan, kunnes päästään haluttuun osioon. Useimmiten verkkosivuilla onkin alisivuja (*subpages*), joihin päästään navigoimalla sivustolla sijaitsevista valikoista (*menus*) tai lisäämällä osoiteriville alisivun tunnus. Tunnus muodostuu vinoviivasta (”/”) ja sivun tunnuksesta. Sivun domain-nimi säilyy kuitenkin samana, esimerkiksi www.esimerkki.fi on aloitussivu ja www.esimerkki.fi/mallisivu on sen alisivu. (Heng 2012.)

Verkkosivuja tehtäessä tulee myös harkita, tarkastellaanko sivuja myös älypuhelimista tai tableteista ja varautua näihin tapauksiin. Tulee siis välttää isoja latausaikoja, esimerkiksi isot ja/tai tarkat kuvat lisäävät huomattavasti latausaikoja. Huomioitavaa on myös se että tableteissa ja älypuhelimissa kiinteät kuvat (*fixed images*), kuvat joiden paikka ja koko ovat lukittu samaan arvoon, voivat aiheuttaa pitkiä latausaikoja tai muuten rikkoa sivuston ulkoasun. Nämä asiat tulisi huomioida ohjelmoitaessa sivuja.

2.1 HTML

HTML on verkkosivujen kuvauskieli. HTML toimii itse tekstinä, johon tulee lisämuuttujia. HTML muodostuu tageista, kutsutaan myös HTML elementeiksi (< >), jonka sisällä on jokin termi. Yleisimpiä elementtejä tagejen sisällä verkkosivuilla ovat <doctype>, <h1>, <p>, ja <a>. Käytämme näitä termejä, sillä on hyvin toden-

näköistä että sivulta löytyy yksi tai useampi näistä termeistä. Näillä siis luodaan yksinkertainen verkkosivu, jossa on tekstiä, kuvia ja hyperlinkkejä. (W3Schools 2012b.)

`<Doctype>`-tagi kuvaa minkälainen dokumentti on kyseessä (*document type*). `<H1>` kuvaa otsikkoa (*header*), `<p>` tarkoittaa kappaletta (*paragraph*), `` kuvaa (*image*) ja `<a>` tarkoittaa linkkiä (*anchor*). Yllä mainitut tagit saavat arvoja tagien väliin tai joskus jopa itse tagin sisään, tarkentaen mitä dataa haetaan. Kuvassa 1 on esimerkiksi sivu ja sen HTML. (W3Schools 2012a.)



Kuva 1. HTML ja kuinka se toistuu selaimella (W3Schools 2008)

HTML-elementit voivat myös saada tyyllittelyjä. Tyyllittelyt antavat tarkennuksia elementeille. Tyyllittelyt mahdollistavat erilaisten fonttien, värien ja muiden tyyllittelyjen lisäämisen. Tyyllittelyt auttavat siis personoimaan verkkosivuja ja näin saadaan sivut erottumaan muusta massasta. (Kyrnin 2011.)

2.2 CSS

CSS on niin ikään kuvauskieli. CSS perustuu luokkaohjelmointiin. Sitä ei juuri käytetä itsenäisesti vaan sitä käytetään HTML:n tukena. CSS:n tarkoitus on personoida HTML:ää. Miksi siis käyttää CSS vaikka HTML:ään voi itsessään lisätä tyyllittelyjä niiden elementteihin? CSS:ää tulisi käyttää kun verkkosivut ovat kasvaneet, eikä haluta enää jatkossa erikseen muotoilla jokaista otsikkoa tai elementtiä. Vaan kirjoitetaan HTML:n tueksi CSS, johon alustetaan arvoja. Tämän avulla voidaan määrittää esimerkiksi se että kaikki otsikot ovat jotain tiettyä fonttia.

CSS:n toimimiseen tarvitaan viittaus HTML:ssä CSS:ään, jotta se osataan ottaa käyttöön. Lisäksi tarvitsee määritellä tiedostopolku CSS:ään. (taulukko 1.)

Taulukko 1. CSS:ää käytetään HTML:n tukena

default.htm
<pre><html> <head> <title>Esimerkki dokumentti</title> <link rel="stylesheet" type="text/css" href="style.css" /> </head> <body> <h1>Ensimmäinen otsikko</h1> </body> </html></pre>
style.css
<pre>body{ background-color: #FF0000; }</pre>

Pieniä eroja löytyy myös HTML:ään verrattuna. CSS:ää ei tarvitse kirjoittaa tageilla vaan laitetaan kaikki arvot elementeille sulkujen sisään. CSS:ssä osien nimet ovat valitsin, ominaisuus ja arvo. (HTML.net 2006.)

body{ background-color: #000000 }

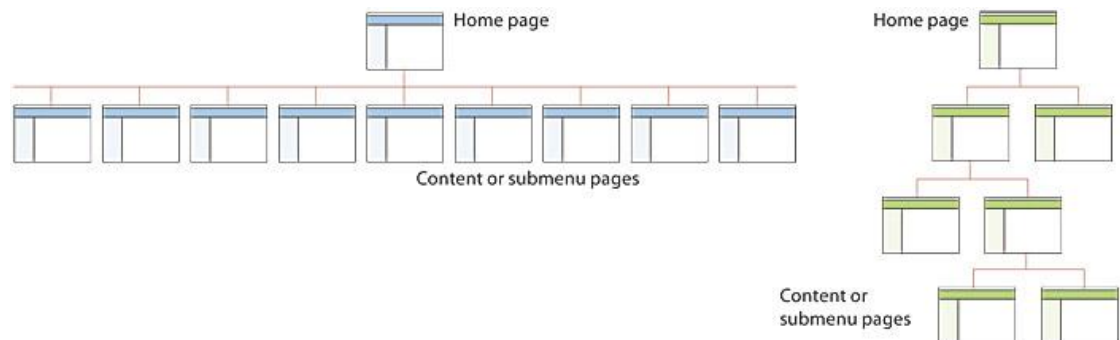
Body on valitsin, background-color on ominaisuus ja #000000 arvo.

2.3 Verkkosivujen rakenne

Verkkosivuja rakennettaessa, tulee myös tarkastella, mihin tarkoitukseen verkkosivut tulevat. Onko sivut tarkoitettu henkilökohtaiseen tarkoitukseen, tiedonjakamiseen tai mahdollisesti yritystoimintaan? Kun sivujen tarkoitus on selvitetty, tulee seuraavaksi pohtia sivujen rakennetta. Tärkeimmät lähtökohdat ovat sivuilla käytetyt termit ja niiden yhteneväisyys. Vaihdeltaessa sanoja ja viljellessä useita erilaisia termejä käyttäjät helposti eksyvät sivuilla tai kadottavat mielenkiintonsa, koska heidän on vaikea löytää halutut sivut tai tiedot selattaessa sivuja.

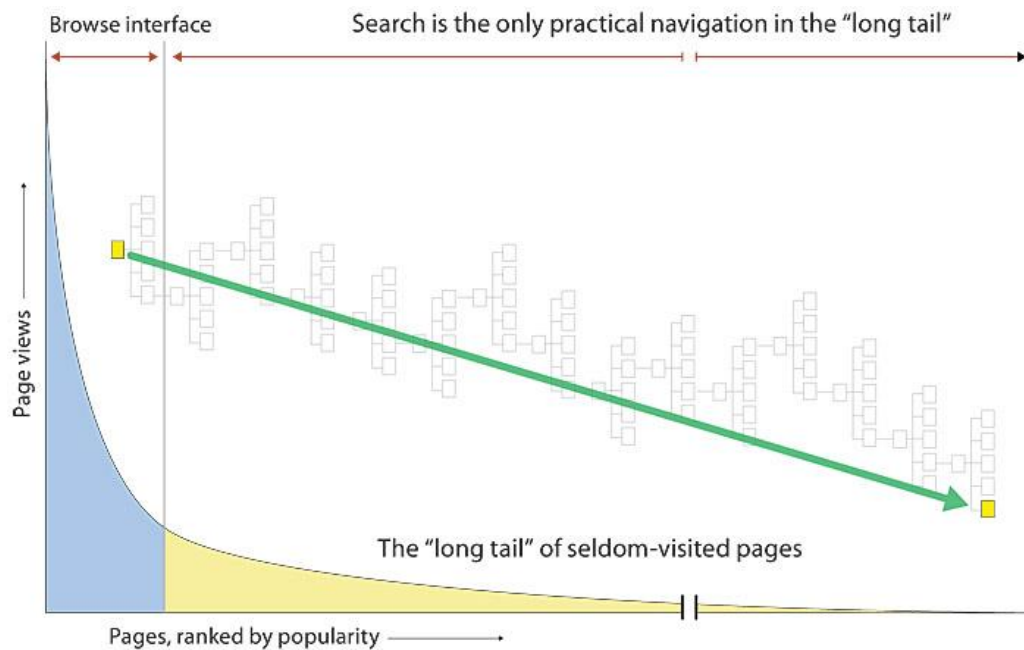
Verkkosivujen rakenteessa tulisi ottaa huomioon myös se, että verkkosivut, jotka on tarkoitettu portfolioiksi, eroavat huomattavasti sivuista, joiden on tarkoitus toimia tie-

donlähteinä, tai jopa yrityksen kotisivuista. Kaikissa lähtökohtana on automaattisesti painottaa pääsyä sisältösivuille (*content page*), joilla oletetaan olevan paljon liikennettä. Kyseisille sivuille luodaan valikkolinkki tai helposti huomattavissa oleva hyperlinkki (*hyperlink*), jonka valitsemalla pääsee kyseiselle sivulle. Tämä voidaan toteuttaa esimerkiksi kuvilla, jos tekstiä ei haluta käyttää tai mikäli yrityksessä halutaan painottaa logojen mieleenpainuvuutta.



Kuva 2. Vasemmalla malli sivustosta, jossa vähän valikkoja. Oikealla malli sivustosta, jossa paljon valikkoja. (Lynch ja Horton 2009.)

Erilaisissa malleissa on kuitenkin usein erilaisia ongelmia. Vähennettäessä valikkojen määrää on ongelmana valikoiden koko. Valikot valtaavat usein ison osan sivuista, mikäli niihin tulee paljon linkkejä. Näin ne voivat myös alkaa muistuttaa pelkkää listausta ja saa täten sivuston näyttämään siltä, että asiaa ei pystytä tiivistämään parempaan muotoon. Toisesta ääripäästä lähdettäessä taas on useita valikkoja, joilla on paljon alavalikkoja ja halutulle sivulle pääseminen vaatii usean valikon valitsemisen ja sivun latautumisen (kuva 2). Näin vaarana on mielenkiinnon lopahtaminen tai hermostuminen, koska sivulle pääseminen vaatii paljon aikaa ja vaivaa. Näiden kahden väliltä tulisikin löytää tasapaino, joka toimii parhaiten oman sivuston kohdalla.



Kuva 3. Sivujen näyttökerrat suhteessa suosioon (Lynch ja Horton 2009.)

Sivuston kasvaessa suureksi sisällösivuja on useita kymmeniä, sivuille tulisi harkita lisättävän hakutoiminto (*search*). Hakutoiminnolla tarkoitetaan työkalua, johon kirjoitetaan hakusanoja sisällöstä, jota halutaan etsittävän sivustolta. Useimmiten se on jokin tekstikenttä (*text box*), jonka lisänä on näppäintoiminto (*button*), joka käynnistää haun. Tällä autetaan sisällösivuille pääsyä (*access*) ja sivuston luettavuutta. Tämä myös mahdollistaa helpomman pääsyn sisällösivulta toiselle. Näin ollen ei aina tarvitse haluttaessaan mennä toiselta sisällösivulta toiselle sisällösivulle, palata sivuston aloitussivulle ja edetä sieltä eteenpäin. Haku-toiminto myös helpottaa pääsyä sisällösivuille, jotka eivät ole yhtä suosittuja kuin muut (kuva 3). Toisaalta hakutoiminnon lisäksi kannattaa jättää näkyviin joitain valikoita tai hyperlinkkejä myös sisällösivuille, koska tämä helpottaa vierailijoiden navigointia ja parantaa sivuston käyttömukavuutta. (Lynch ja Horton 2009.)

3 SISÄLLÖNHALLINAJÄRJESTELMÄ ELI CMS

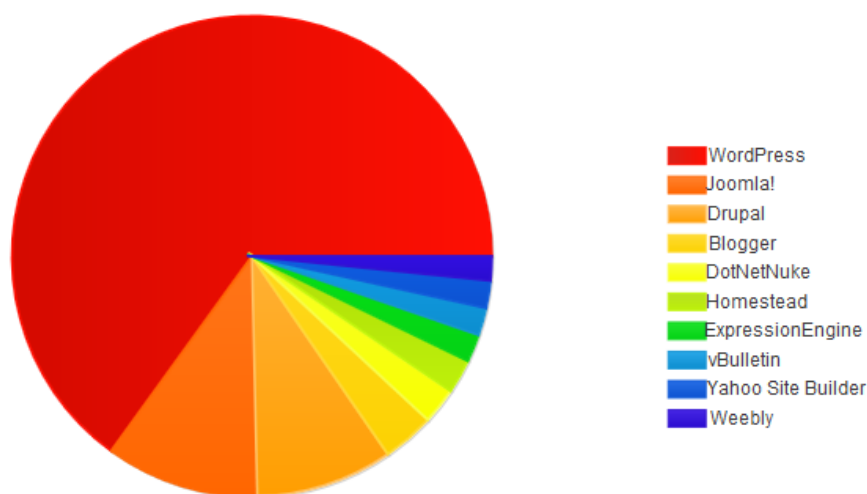
Nimensä mukaisesti sisällönhallintajärjestelmää (CMS) käytetään hallinnoimaan sisältöä verkkosivuilla. Sisällöllä tarkoitetaan tässä yhteydessä tietoa, jota ei tule sekoittaa dataan. Datalla tarkoitetaan pikemminkin asiaa, jonka tietokone voi ymmärtää, kun taas tiedolla käsitettä, joka on lähempänä ihmistä ja vain ihminen osaa tulkita tämän. Sisällöllä on myös joitain perusoletuksia. Sisällöllä on merkitystä ja se halutaan jakaa muiden kanssa. Useimmiten sisältö on päässyt paisumaan ja näin ollen sille tarvitaan

oma hallintajärjestelmä. Sisällönhallintajärjestelmien perimmäinen tarkoitus onkin taata sisällön helppo saatavuus ja tiettyjen toimintojen, pääsääntöisesti päivitysten, automatisointi. Ne on pääasiallisesti suunniteltu helpottamaan sivustojen ylläpitämistä ja päivittämistä. Useimmiten CMS:t sisältävät myös graafisen käyttöjärjestelmän. Tämä on huomattavasti käyttäjäystävällisempi kuin perinteinen verkkosivujen rakenne, sillä näin ollen kokonaisuus on helpompi hahmottaa, eikä järjestelmä säikäytä pois aloittelevia käyttäjiä. Tämän vuoksi oman tietämyksen ja teknisen osaamisen ei tarvitse olla niin korkealla tasolla käytettäessä sisällönhallintajärjestelmiä verrattuna perinteiseen verkkosivujen hallintaan, vaikka perustietämys olisi hyväksi. Tämä auttaa muun muassa templatejen (eräänlainen sivustopohja) tekemisessä sekä CSS:n luomisessa. (Boiko 2005, 65–79 , Creativesites 2009.)

3.1 Sisällönhallintajärjestelmien käytettävyys ja toiminta periaate

Sisällönhallintajärjestelmä toimii ohjelmiston tavoin. CMS antaa luoda sisältöä, hallita sitä, muokata sitä ja laittaa se muiden näkyville ja tarkasteltaviksi. Sisällönhallintajärjestelmällä voidaan toteuttaa muitakin toimintoja kuin verkkopalveluja, mutta keskitytään pääasiallisesti verkkotoimintoihin. Muutamia esimerkkejä sisällönhallintajärjestelmistä ovat mm. Joomla, Drupal ja Wordpress. Suosituin CMS tammikuussa 2013 oli Wordpress (kuva 4). Vertailussa ei ollut mukana sivustoja, joissa ei käytetä sisällönhallintajärjestelmiä (Built With 2009).

CMS Distribution in Top Million Sites



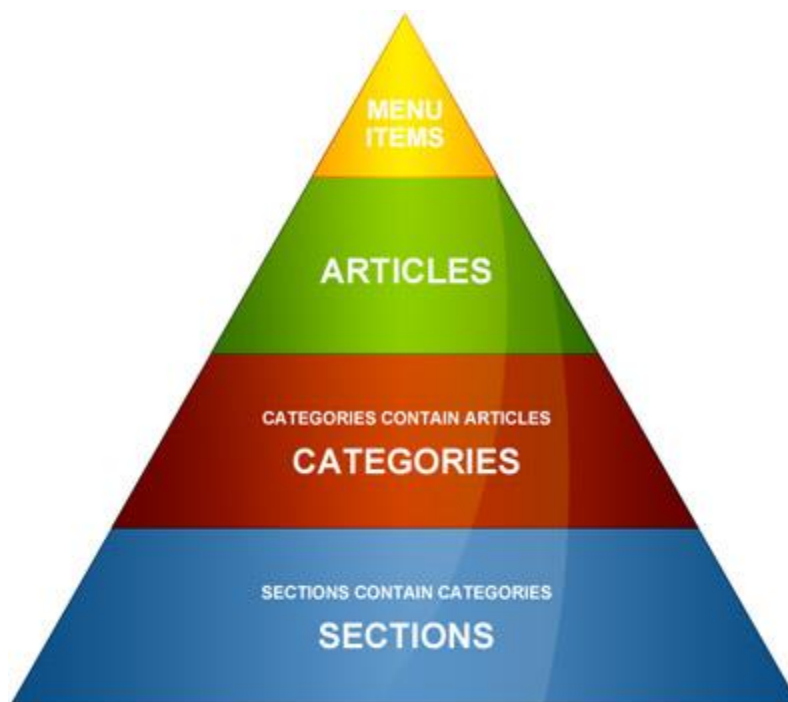
Kuva 4. Sisällönhallintajärjestelmien osuus miljoonan-suosituisimman sisällönhallintajärjestelmää käyttävän sivun mukaan (Built With 2009)

Lähinnä kuitenkin tarkastellaan Joomlaa vaikkakin Wordpress olisi hyvä valita suosionsa kannalta. Aloittaessa työskentely täysin tyhjältä pohjalta olisi hyvä valita suosituin vaihtoehto, sillä mitä todennäköisemmin sille tuki- ja ongelmanratkaisutilanteita löytynee paljon enemmän. Täten mahdollisten ongelmien ratkaiseminen helpottuu huomattavasti.

Yksinkertaisuudessaan verkkosivut ovat siis sarja sivuja, joihin voidaan navigoida valikoiden välityksellä. Nämä valikot saavat arvonsa (sijainti, näkyvyys yms.) toisista elementeistä ja ne voidaan täten toistaa käyttäjälle. Kaikki sivuilla näkyvät tiedot tallennetaan tietokantaan, josta ne haetaan tähän tarkoitukseen käytettävällä ohjelmointikielellä. Tämä kaikki on suhteellisen automatisoitua ja julkaisijalle näkyvkin vain graafinen liittymä, jolla sitä hallitaan. Peruskäyttäjällä on harvoin tarvetta päästä käsiksi sivuston tietokantoihin. Usein ne ovat tekstimuodoissaan myös epäselkeitä ja sisältävät arvoja joihin on saatettu käyttää lyhenteitä, mikä olisi epäkäytännöllistä hallinnoin kannalta. (Boiko 2005, 81–109)

3.2 CMS:n rakenne

Sisällönhallintajärjestelmä rakentuu useasta osasta. Se voidaan jakaa kahteen isoon osaan: front end ja back end. Front end on näkymä, johon päästään käsiksi menemällä suoraan halutun verkkosivun osoitteeseen. Tämä näkymä kuvaa siis sitä kuinka sivuston vierailijat pääasiallisesti näkevät sivuston. Back end taas tarkoittaa CMS:n hallintipuolta, jonka avulla toteutetaan myös front end -näkymä sellaiseksi kuin se halutaan. Back endiin on mahdollista päästä lisäämällä vinoviiva ja sen jälkeen, useimmiten sisällönhallintajärjestelmästä riippuva tunnus, esim. www.esimerkki.fi antaisi pääsyn sivuston front endiin, kun taas www.esimerkki.fi/administrator sivuston back endin, joka on Joomla:n ensisijaisesti käyttämä tunnus. Back end -kirjautumistunnus on riippuvainen CMS:stä, joka on käytössä.



Kuva 5. CMS rakenne diagrammina (Joomla Design 2009)

Sisällönhallintajärjestelmällä toteutetut sivut koostuvat eri tasoista. On olemassa kolme tasoa; moduulit (*modules*), valikot (*menu items*) ja artikkelit (*articles*). Artikkelit usein lukeutuvat myös kategorioihin. Nämä julkaistaan ja näin ollen saadaan sisältöä näkyville (kuva 5). Tähän kuitenkin vaaditaan joitakin työkaluja, jotta nämä saataisiin toteutettua. Tarvitaan siis tietokanta, johon tiedostoja tallennetaan ja josta ne noudetaan sekä ohjelma, jolla hallinnoidaan kyseistä tietokantaa. Tämä toteutetaan ohjelmalla joka käyttää SQL:ää (*structured query language*), joka on nykyään standardi ja tietokantojen kanssa keskusteluun kehitelty kuvauskieli. Peruskäyttäjän ei kuitenkaan tarvitse tietää muuta kuin se, että tietokantoihin tallennetaan artikkelien tiedot ja metatiedot (*metadata*). Metatiedoilla tarkoitetaan tietoja, jotka auttavat luokittelemaan ja kuvaamaan artikkeleita. Metatiedot koostuvat metatietoelementeistä ja näiden arvoista (*value*). Esimerkiksi artikkelin kirjoittaja, artikkelin julkaisuaika ja artikkelin otsikko olisivat metatietoelementtejä, ja näille olisivat arvoja Erkki Esimerkki, 1.1.2011 ja Artikkelini. (Boiko 2005, 453–463)

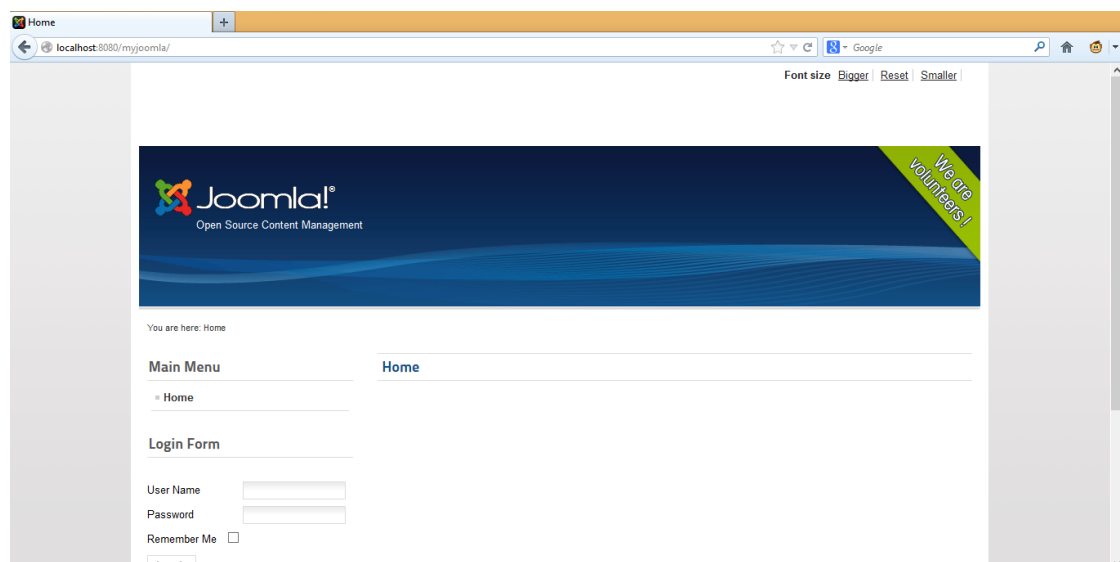
Sivusto tarvitsee myös templatien, josta se saa arvonsa. Tämä sivustopohja on osa, joka sisältää sivuston moduulien tiedot: missä ne sijaitsevat ja miltä ne näyttävät. Templatet sisältävät CSS-tiedostot. Eli periaatteessa voidaan sanoa että sivuston ulkoasu muodostuu kokonaan templatien pohjalta. Sisällönhallintajärjestelmä sisältää vähintään yhden templatelähtöisen alkuperäisessä kokoonpanossaan. Templateja voi-

daan joko ladata lisää tai luoda uusia vanhan pohjalta. Jos luo uuden vanhan pohjalta, tulee huomioda, että mikäli sivustoa päivitetään sisällönhallintajärjestelmän sisäisellä päivitystoiminnolla, voi aiemmat muokkaukset templatesta kadota. Tärkeää olisi nimetä template uudeksi virheiden välttämiseksi. (Softaculous 2013.)

Sisällönhallintajärjestelmän parhaita puolia onkin sen helppo hallinnointi. Paitsi että uusien käyttäjien ja käyttäjien hallinnointi on helppoa, tulee CMS:ään toisinaan päivityksiä jotka parantavat tai lisäävät perustoimintoja järjestelmään. Päivitys toteutetaan sivuston back endin kautta sille erikseen varatussa osiossa.

3.2.1 Front end

Sisällönhallintajärjestelmissä sivuston perusosoitenimellä päästään sen front endiin. Front end on siis sivuston julkisivu ja se, mitä verkkosivuilla halutaan tuoda ilmi (kuva 6). Sisällönhallintajärjestelmien yksistä käyttötavoista on se että sivustolla vierailevat henkilöt voivat tehdä pieniä muutoksia sivulle, mikäli heille on tähän annettu käyttöoikeudet. Vaatii useimmiten rekisteröinnin tai perusoletuksen että vierailevat ovat aina alhaisimman tason käyttäjiä. Kirjautuminen on kuitenkin parempi tapa, sillä korkean liikenteen sivustoilla tämä ei muuten ole käytännöllistä, sillä ilman kirjautumista ei voida tarkistaa kuka muutoksia haluaisi tehdä.

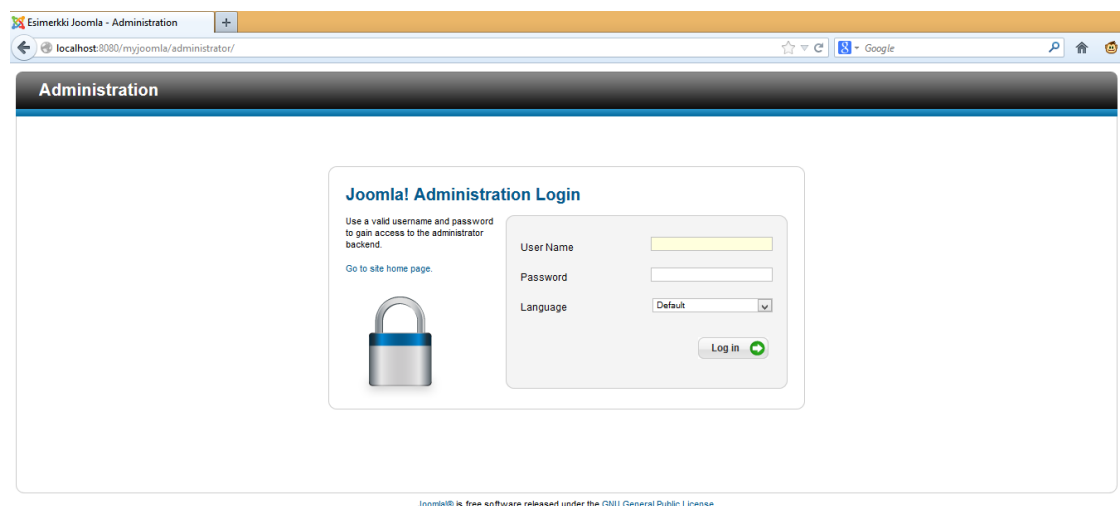


Kuva 6. Joomla, front end

Front end -hallittava sisällönhallintajärjestelmä ei siis yksinään kuitenkaan ole mahdollista, sillä vaikka vierailijoilla olisikin oikeus muokata sivustoa, tulee niille yhä määrittää oikeudet sivuston back endistä. Muita tietojakin tulee hallinnoida ja pitää ne yllä jossain, sitä varten back end on suunnattu.

3.2.2 Back end

Back end on sivuston ylläpitäjille tarkoitettu osa. Täältä ohjataan mitä sivusto sisältää ja kuinka sivusto rakentuu (kuva 7.). Back endin kautta on mahdollista muokata templatea, valikoita, käyttäjiä, komponentteja ja laajennuksia. Back end sisältää myös oman erillisen ohjauspaneelin, josta sivuston sisäinen hallinnointi on helppoa ilman sen kummempaa teknistä osaamista. Sisällönhallintajärjestelmät painottavatkin lähesyttävyyttä ja helppokäyttöisyyttä juuri back endin välityksellä.



Kuva 7. Joomla, back end (ei kirjautuneena sisään)

Back endin ohjauspaneeli sisältää aiemmin selityt valikot (käyttäjät, valikot, artikkelit ym.) ja lisää. Täältä voidaan vaihtaa niin sivustoilla käytettävät tietokantatyypit riippuen tietenkin sisällönhallintajärjestelmän vaatimuksista ja yhteensopivuuksista, tyhjentää sivuston välimuistia kuin korjata erilaisia aiheita mitä huomaa puutteelliseksi, esimerkiksi tekstieditorin vaihtaminen, lisäosien asentaminen yms.

4 JOOMLA

Joomla on yksi suosituimmista sisällönhallintajärjestelmistä. Vain Wordpress päihittää sen suosiolla. Tämä oli yksi tekijä, joka sai minut valitsemaan Joomla-järjestelmän sisällönhallinnan käyttöpohjaksi. Toinen suuri vaikuttaja oli se että yritys, jolle luotiin verkkosivut, MHG Systems Oy, halusi käyttää kyseistä CMS:ää. Osalta henkilöstöstä löytyi jo aiempaa kokemusta sen käytöstä. Joomla'n nimi tulee swahilin kielestä, ja suomeksi käännettynä tarkoittaa ”kaikki yhdessä” tai ”kokonaisuudessaan”. Joomla'n pääpaino on *portaalina* ja verkkosivuina toimimisesta. Portaalilla tarkoitetaan verkkosivua, joka kasaa informaatiota yhdelle sivustolle useista eri lähteistä ja näyttää sen sivustolle luodulla mallilla, joka on globaali sivustolla käytetty malli. Portaalien huomattavin erottava tekijä yritysten, kerhojen ja portfolioiden verkkosivuista on hakuelementtien puuttuvuus. Joomla'n tekniset vaatimukset kattavat vaatimuksia SQL:stä PHP:stä ja verkkopalvelimista (liite 1). Tekniset vaatimukset riippuvat myös siitä mitä Joomla'n versiota käytetään. Nykyään tulisi käyttää jo vähintään, 2.5-versiota, tai uusinta 3.0-versiota. Keskityn lähinnä 2.5-versioon ja sen toimintoihin, sillä verkkosivut on toteutettu sen avulla, koska versiota 3.0 ei ollut vielä julkaistu, kun sivuja alettiin työstää.

Joomla'n saadessa useita julkaisuja, on siihen luotu myös sisäinen päivitystoiminto. Mikäli aiemmin on ollut käytössä vanhempi versio Joomla:sta, se on mahdollista päivittää toiseen sisäisen päivityksen avulla hyödyntäen muutamaa pluginia. Tärkeää päivitysprosessissa on varmuuskopiointi. Tarkoitus on siis ottaa kaikki kansiot ja tietokannat talteen uuteen erilliseen sijaintiin. Tämä sen takia että päivityksen yhteydessä voi olla jokin katkos tai muu ongelma, ja tiedostot voivat vioittua. Tällöin, ilman varmuuskopiointia, menetettäisiin jo aiemmin tehdyt ja toimivat ratkaisut. Mikäli vaihtaa aiemmasta julkaisusta uudempaan tai toisesta sisällönhallintajärjestelmästä toiseen, kutsutaan tätä prosessia *migraatioksi*. Siirryttäisiin siis esimerkiksi Joomla 1.5:stä Joomla 2.5 käyttöön. Migraatiota ei kuitenkaan tule sekoittaa järjestelmän sisäisiin päivityksiin. Järjestelmä voi siis saada päivityksen versiosta 2.5.7 muuttuen järjestelmäksi 2.5.8. Prosessia kutsutaan järjestelmäpäivitykseksi.

Joomla'n suosio selittyy myös osaksi sen yhteisön luomilla komponenteilla, laajennuksilla ja plugineilla. Näitä löytää Joomla'n sivustolta ja niihin pääsee käsiksi Joomla'n sivustoilta. Osa laajennuksista on lisäksi ilmaisia, mikä kannustaa valitsemaan Joomla-

lan käyttöalustaksi, sillä voi etsiä haluttuja toimintoja laajennusvalikoimasta ladata ne ja saada toimimaan, eikä itse tarvitse luoda koodia luodakseen ohjelma / toiminto. Laajennukset eivät kuitenkaan ole ainoastaan Joomla-sisällönhallintajärjestelmän ominaisuuksia, vaan tämä löytyy myös muista sisällönhallintajärjestelmistä. Osa laajennuksista on maksullisia, mutta summat ovat usein pieniä tai kertamaksuluontoisia, joten kustannukset eivät nouse ihan mahdottomiksi. Maksullisuus hieman taistelee avoimen lähdekoodin periaatetta vastaan, johon Joomlaakin kuuluu. Osiin laajennuksista ja kolmannen osapuolen komponenteista löytyy tosin hyvät tukipalstat ja tukevat Joomlaan toimintaa. Tulee siis tarkastella ja ottaa huomioon komponentin hyvät ja huonot puolet, jos on lisäämässä sitä osaksi omaa Joomla kokoonpanoa.

Joomlan ohjauspaneeli koostuu back endin komponenteista ja osista. Sisältää mm. valikoiden hallinnointi (*menu manager*), artikkelien hallinnointi (*article manager*), käyttäjien hallinnointi (*user manager*), Joomlaan tila (*status*) ja laajennusten ja CMS:n päivitykset. Alkuun ohjauspaneeli näyttää käytännölliseltä portilta eri osioihin, mutta jonkun käyttöajan jälkeen alkaa oppia rakenteen paremmin, ja ei tarvitse aina kiertää ohjauspaneelin kautta. Ohjauspaneeli on ikään kuin back endin etusivu.

4.1 Joomla-migraatio ja päivitykset

Migraatio on siis Joomlaan julkaisusta toisen julkaisun, useimmiten vaihdetaan uudempaan julkaisuun, siirtymistä. Tämä voidaan toteuttaa Joomlaan sisäisesti lisäkomponentin avulla tai kopioimalla tärkeimmät kansiot ja tietokannat ja tuomalla ne uudempaan. Käytettäessä kopiointimenetelmää tulee huomioida suuret muutokset, jotka on listattu Joomlaan sivuilla. Tämä useimmiten aiheuttaa vähintään pienen tietokantojen ja tiedostojen rakenteiden muuttamista, ja useimmiten vaatiikin jo vähän laajempaa ymmärrystä ohjelmoinnista. Muutokset siirryttäessä julkaisusta 1.5:stä 2.5:n käyttöön näkyvät suurimmalta osaksi PHP:ssä. Toinen vaihtoehto migraatiolle on hankkia komponentti, joka toteuttaa tämän. Suorittaa automatisoituna lähestulkoon kaiken, kaikki mitä käyttäjältä vaaditaan, on oikeudet ylikirjoittaa aiemmat datat ja valita haluttavat osat mitä tuodaan uuteen julkaisuun. Tämän pluginin nimi on jUpgrade ja on se löydettävissä ja ladattavissa Joomlaan laajennuskirjastoista.

Migraatiossa jUpgrade komponentilla, joka on Joomlaan suosittelema menetelmä, asennetaan aluksi komponentti. Joomla tulisi myös päivittää viimeisimpään 1.5 jake-

lun versioon etsimällä päivityksiä, löytyy Joomlaan sisäisestä ohjauspaneelistä. Tämän jälkeen tulisi varmistaa että muutama plugin on käytössä, sillä päivityskomponentti käyttää niitä. Tämän jälkeen valitaan komponentti ja aloitetaan migraatio. Ensimmäiseksi Joomla hakee ja tuo datan, joka sisältää mm. käyttäjä-, kategoria-, valikko-, sisältö-, moduuli-, media- ja *newsfeed*-tiedot (sisältää uutisia / pieniä ilmoituksia). Joomla pystyy myös tuomaan vanhat templatet uuteen kokoonpanoon tai voi erikseen ladata ne uudemman julkaisusta ja tuoda kopiointimenetelmällä siihen, perinteinen leikkausliitä -metodi toimii. Itse luotuja templateja jUpgrade komponentti ei kuitenkaan pysty tuomaan, vaikkakin manuaaliset ohjeet tälle löytyvät jUpgrade komponentin ohjesivustoilta.

4.2 Laajennukset

Joomlaan suosio johtuu paljolti yhteisön luomista ja ylläpitämistä kolmannen osapuolen laajennuksista. Joomlaan sivuilta onkin suoraan mahdollista päästä laajennuskirjastoisiin. Laajennukset voivat olla moduuleja, templateja, komponentteja tai plugineja (You!Joomla 2013). Näiden lisääminen ja asentaminen on automatisoitua, ja suurin ongelman aiheuttaja lienee lisäosien ja komponenttien käyttöönotto. Lataaminen tapahtuu Joomlaan kotisivujen tai kolmannen osapuolen sivustojen kautta, ja niiden lisääminen hoidetaan helposti sivuston back endin kautta. Ladatut tiedostot voidaan implementoida kahdella ottaa käyttöön kahdella tapaa, voidaan purkaa tiedostot oikeaan polkuun ja etsiä ne sieltä tai asentaa ne paketista. Molempien niin asentamis- (install) kuin löytämismenetelmät (*discover extensions*) hoidetaan Joomlaan back endin laajennukset-valikon alta, laajennusten hallinta -kohdasta (*extension manager*). Kun ladattu lisäosa on paketissa, käytetään asentamismenetelmää. Ensin etsitään koneelta pakattu tiedosto (*archive*, esim. tiedostomuotoa ”.zip”) hakutoiminnon avulla. Sitten valitaan asenna (*install*) ja sisällönhallintajärjestelmä päivittää itsensä käsittämään kokoonpanon. Jos asennus on tapahtunut kansiokokonaisuuksista, täytyy valita laajennusvalikosta tunnistavälilehti (*discover*). Sitten välilehdellä tulisi näkyä laajennus. Se voidaan asentaa valitsemalla se painamalla asenna-painiketta. Jos laajennusta ei kuitenkaan vielä näy listassa, voi laajennuksia yrittää etsiä siihen varatulla samannimisellä nappulalla, tunnistaa. Tämä tunnistaa laajennukset ja sitten ne voidaan asentaa. (Joomla 2012/2.)

Jossain tapauksissa laajennukset eivät kuitenkaan vielä näy asennuksen jälkeen toiminnassa. Laajennukset voivat käyttää jotain pluginia, koostuu komponenteista, jotka lisäävät ohjelman käytettävyyttä lisäämällä siihen toimintoja, jota ei ole vielä otettu käyttöön. Tällöin tulee tarkistaa että laajennuksessa käytettävät pluginit ovat käytössä. Tähän päästään menemällä laajennusvalikossa plugineihin (*plug-in manager*). Näkyville tulee iso lista plugineja. Pluginit on merkattu joko punaisella pallolla, jossa on rasti tai vihreällä pallolla, jossa on oikein-merkki. Vihreä tarkoittaa että plugin on käytössä, ja punainen sitä, että se ei ole käytössä. Voidaan valita useita komponentteja ja käyttää oikealla ylhäällä sijaitsevia nappuloita muuttamaan plugineiden tilaa. Jotkut laajennukset siis vaativat pluginien olevan toiminnassa voidakseen itse toimia.

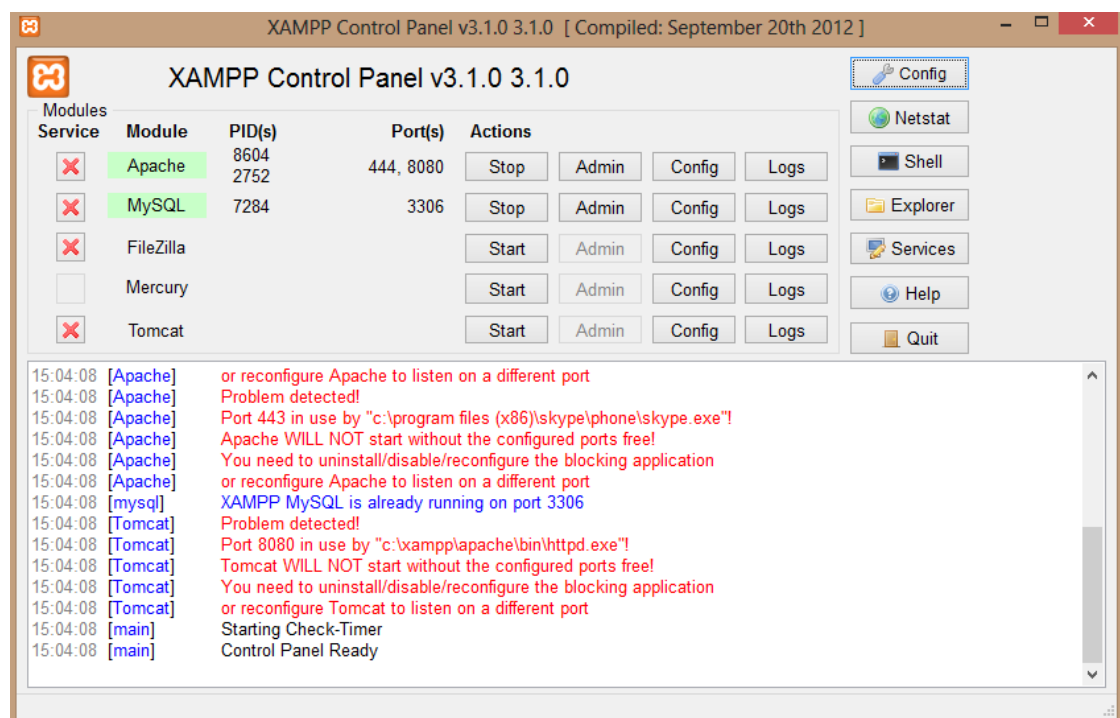
Joissain tapauksissa laajennuksista halutaan myös päästä eroon, koska huomataan että ne ovat vanhentuneet tai niille ei ole enää tarvetta. Mikäli laajennus on komponentti, on siis lisännyt oman merkintänsä komponentti valikkoon, sillä on useimmiten ohjelmiston poistamistoiminta oman valikkonsa alla. Jos se ei kuitenkaan lisännyt merkin-tää komponenttilistalle, vaan se onkin jokin lisä-osa tai muu toiminto, se voidaan poistaa samasta paikasta, mistä se lisätään. Tämä tapahtuu laajennusvalikosta laajennuksista, joiden alta valitaan hallinnoi-välilehti (*manage*). Täältä laajennukset voidaan ottaa pois käytöstä disable-nappulan avulla tai poistaa pysyvästi uninstall-nappulalla.

5 JOOMLA-SISÄLLÖNHALLINTAJÄRJESTELMÄN KÄYTTÖÖNOTTO

Läpi käydään kaksi erilaista tapaa toteuttaa verkkosivut. Toinen tapa on luoda kokonaan uusi Joomla kokoonpano, kun taas toinen tapa on vanhasta järjestelmästä uuteen päivittäminen. Käymme läpi päivittämisen, sillä MHG Systems Oy:lle teettämässäni pohjassa käytin tätä metodia. Yrityksellä oli aiemmin käytössä vanhempi Joomla-julkaisu. Teemme molemmissa kokoonpanoissa ratkaisut lokaalina (local host), jotta ei tarvitse alkuun harkita, minne sivut tulisi sijoittaa tai alkaa etsimään verkkopalvelujen tarjoajia. Eikä tiedostojen siirtämisestä tule sen kummempia ongelmia, kun ne siirretään palvelimelle. Palvelin on verkkosivujen fyysinen sijoituspaikka, josta niihin päästään käsiksi).

5.1 Alkuvaatimukset

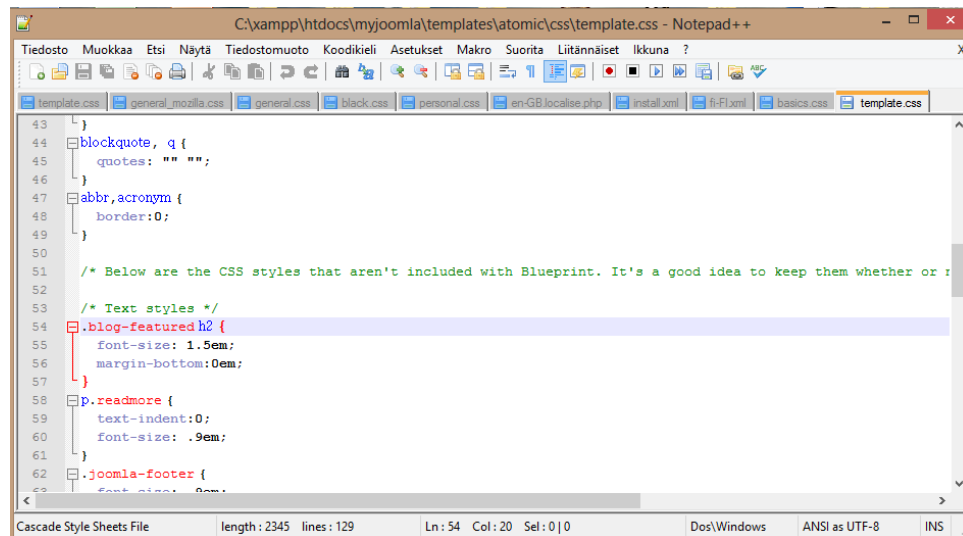
Aluksi tulee tehdä muutama asennusprojekti. Tarvitaan siis ohjelmia, jotka voivat hoitaa *Apachen*, *SQL:n* ja *PHP:n*. Apache on unix-pohjainen verkkopalvelin, jolla hoidetaan HTTP (*Hypertext Transfer Protocol*) palvelut (Your dictionary 2004). PHP on kirjoituskieli (*scripting language*), jolla voidaan ohjelmoida yleisiä toimintoja. Käytetään yhdessä HTML:n kanssa ja yleisimmin nimenomaan verkkosivujen yhteydessä. Näiden lisäksi tulisi myös olla jokin editori, jolla muokata tarvittaessa PHP:tä, CSS:ää tai HTML:ää. Tämä voidaan toteuttaa esimerkiksi käyttöjärjestelmän muistiolla, mutta kyseinen ohjelma ei auta koodin löytämisessä ja virheiden paikallistamisessa. (PHP.net 2010)



Kuva 8. XAMPP ja halutut palvelut

Käymme siis aluksi kaksi ohjelmaa, joilla aiemmin mainitut toiminnot saadaan suoritettua, XAMPP ja Notepad++. XAMPP on ohjelma, joka toimii Apache-palveluna ja yhdistäen siihen PHP:n, SQL:n ja muut halutut toiminnot (kuva 8). Ohjelma on myös ilmaiseksi ladattavissa, ja näin ollen saamme kaikki toiminnot yhden ohjelman taakse. Näin ollen ei tarvita useaa eri ohjelmaa hallinnoimaan toimintoja. Notepad++ puolestaan toimii koodikielen editorina. Tämä sen takia, että mikäli ilmenee sulkuvirheitä tai muita loogisia virheitä, niin editori ilmoittaa virheiden sijainnit siihen tarkoitetulla merkillä, useissa ohjelmissa merkki on huutomerkki tai rasti punaisella pohjalla (kuva

9). Notepad++ sattuu myös olemaan ilmainen ohjelma ja kuuluu GPL:n piiriin (General Public License). Lisenssi tarkoittaa sitä että kenellä tahansa on mahdollisuus muokata ja jakaa kokoonpanoa ja sen kaikkia versioita (GNU 2007).



Kuva 9. Notepad++ ja koodin korostus

XAMPP ja Notepad++ ohjelmat voidaan asentaa perusasetuksineen *wizardien* avustamana. Ohjelman wizard on eräänlainen käyttöliittymä, joka on suunniteltu auttamaan tuotteen asennuksessa, ja useimmiten tuotteen poistamiseen on myös erikseen luotu oma wizardinsa. Wizard toteutetaan keskusteluvalikoiden (*dialog boxes*) välityksellä, joissa on vielä monesti oletusasetukset. Tässä tapauksessa oletusasetukset riittävät. Sen jälkeen joudumme muokkaamaan XAMPP:n toimintaa. XAMPP:ia tulee muuttaa sen takia, että se käyttää sellaista porttia, joka saattaa aiheuttaa ristiriitoja jo valmiiksi koneella olevan ohjelman takia (yksi tällainen ohjelma on Skype). Näin ollen XAMPP ei toimi halutulla tavalla ilman portin muutosta. (Microsoft 2012.)

XAMPP:n palvelinportin muuttaminen tapahtuu seuraavasti. Avataan XAMPP:in polusta seuraava tiedosto ”httpd.conf” Notepad++ ohjelmalla. Käytettäessä oletusasetuksia se löytyy seuraavasta polusta: C:/xampp/apache/conf/httpd.conf. Ollessasi tiedostossa etsi kohta ”Listen 80”, ja muutetaan se ”Listen 8080”, seuraavana etsitään kohta Servername ja sinnekin muutetaan 80 tilalle 8080. Joissain tapauksissa tulee myös vaihtaa SSL kuuntelemaan toista porttia. Se tapahtuu muuttamalla httpd-ssl.conf tiedostoa, ja muuttamalla arvot 443:sta 444:ään. Joissain tapauksissa MySQL ei välttämättä toimi. Tähän saattaa auttaa MySQL:n käynnistäminen erillisellä tiedostolla C:/xampp/mysql_start.bat. Jos ohjelma ei toimi edes muokkausten jälkeen, kannattaa

kokeilla asentaa XAMPP tiedostoista installerin sijaan. Tämän jälkeen käynnistetään XAMPP ja palvelut MySQL ja Apache. Näin tilaksi muuttuu *running* ja palvelut ovat käytössä. Tämän voi tarkistaa selaimella local host osoitteesta, lisäten porttitunnuksen 8080 sen perään, esim. localhost:8080 kirjoitetaan selaimen osoiteriville. Tällä pääsee XAMPP:n ohjauspaneeliin, josta voi asettaa salasanoja nähdä palveluiden tilat (status) yms. Tässä merkittävintä on se että tilasivulla (status) palveluiden tulisi olla vihreitä.

5.2 Joomla kokoonpanon asentaminen tai päivittäminen

Rakennamme Joomla lokaalina parista eri syystä. Näin muutokset näkyvät välittömästi, eikä palvelimen käyttöoikeuksista tarvitse murehtia. Täten emme tarvitse verkkosisäntäpalvelua, ja tiedostojen siirtäminen tapahtuu huomattavasti nopeammin. Eikä Joomla-kokoonpanon muokkaaminen palvelimelle vaadi juuri erityisen paljon muutoksia. Joomla saa asennettua lataamalla se Joomla verkkosivuilta. Työssä käytin julkaisua 2.5.x, koska alkaessani tekemään tätä, ei Joomla 3.0.x vielä ollut. Käymme siis läpi, Joomla 2.5.x julkaisun asennuksen. Joomla järjestelmä on arkistomuodossa (archive format, kuten .zip), joten se tulee purkaa ennen liikkeelle lähtöä. Rakentaessa Joomlaa lokaalina, puretaan se XAMPP-polkuun. XAMPP polusta valitaan htdocs kansio, jonne tehdään kansio, joka nimetään haluamallaan tavalla (Huom. tätä nimeä tarvitaan jatkossa, joten se tulisi muistaa). Kun paketti on purettu, voidaan avata selain ja mennä osoitteeseen localhost:8080/myjoomla. Myjoomla kuvaa kansion nimeä, joka luotiin purkaessa Joomla arkisto siihen). Tämän jälkeen alkaa Joomla alkuarvojen alustaminen.

Joomla alustaminen tapahtuu selaimella täyttämällä tekstikenttiä ja valitsemalla haluttuja vaihtoehtoja. Asennus alkaa sivulla, jolla valitaan kieli. Siirtymällä eteenpäin, painetaan seuraava-painiketta (*next*), joka sijaitsee sivun oikeassa yläreunassa. Seuraavalla sivulla käydään läpi tietokantojen toimivuus, ja sivulla tulisi olla enimmäkseen vihreää, mikä tarkoittaa sitä että palvelut ovat käytössä. Suositusasetuskohdassa voi tosin olla poikkeuksia, sillä oletusasetuksena on, että virheet näytetään (*display errors*). Tästä huolimatta voidaan siirtyä seuraavaan kohtaan. Seuraavalla sivulla on ohjelman GPL, jotta käyttäjät ymmärtävät, mitä oikeuksia heillä on ohjelmiston suhteen. Tämän jälkeen tullaan kohtaan, johon valitaan käyttäjätunnus ja salasana, lokaalina ratkaisuna voidaan käyttää käyttäjätunnuksena ”root” ja salasana voidaan jättää tyhjäksi. Palvelimen nimi lokaalina voidaan myös jättää localhostiksi, joka onkin ole-

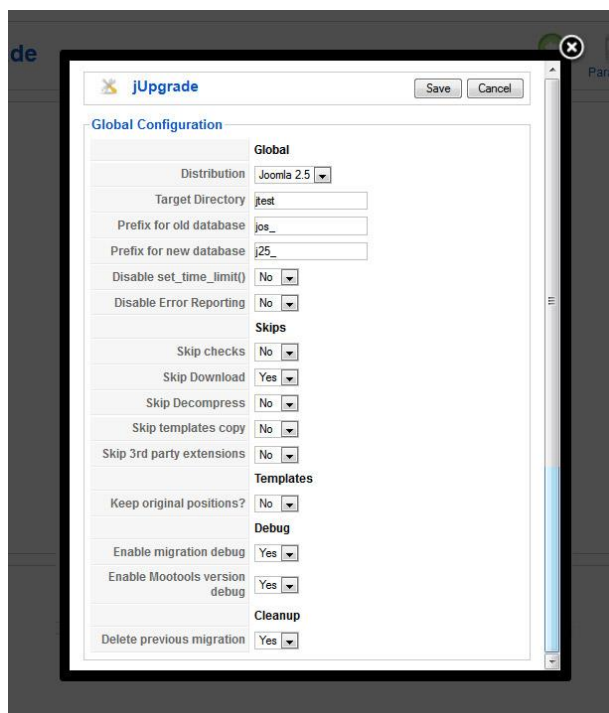
tusarvo. Tietokantatyypinä tulisi olla mysql, koska sillä on isompi tuki erilaisille lisäosille ja liitännäisille verrattuna tavalliseen mysql tietokantatyyppiin. Tietokanta nimetään halutuksi ja tietokannalle annetaan vielä etuliite, jolla jatkossa kyseinen tietokanta tunnistetaan. Lokaalina seuraavan sivun FTP-asetukset (*File Transfer Protocol*) voidaan myös ohittaa. FTP tarkoittaa tiedostonsiirto protokollaa. Jatkamme seuraavalle sivulle, jossa valitaan sivuston nimi (esim. My Website). Tähän tekstikenttään voidaan liittää myös metatietoja, jotka auttavat sen löytämisessä ja tunnistamisessa. Halutut käyttäjätunnukset ja salasana valitaan nyt, ja varmennetaan se sähköpostiosoitteella. Tämän yhteydessä voidaan myös liittää mallirakenne Joomla-kokoonpanosta, tutkiskelemalla ja opiskelemalla valmiin sivuston rakennetta, voi se auttaa hahmottamaan rakennejohdannaisuuksia ja Joomla järjestelmän kokonaisuutta. Mikäli halutaan tehdä tyhjään kokoonpanoon ratkaisut, tulisi mallirakenne jättää pois. Tämän jälkeen tulee sivu, jolla on asennustiedoston poistaminen. Painamalla painiketta asennustiedosto poistetaan automaattisesti. Tämä tulee tehdä, sillä muuten Joomla käynnistyy aina asennusmuodossaan. (Joomla 2012c.)

Asennuksen jälkeen on ensimmäisen kerran mahdollista tutkiskella sivuston rakennetta. Sivulle päästään samalla osoitteella, millä alun perin Joomla asennettiin selaimella, eli tässä tapauksessa localhost:8080/myjoomla. Lisäämällä ositerivin loppuun /administrator päästään sivuston back endiin. Tätä ennen voidaan kuitenkin varmistaa että tietokannat on luotu ja onnistuneesti lisätty Joomla kokoonpanoon. Tämä voidaan toteuttaa XAMPP:n yhteydessä olevalla phpMyAdminilla (XAMPP:n yhteydessä oleva tietokantojen hallinnointipalvelu). Tänne päästään selaimella kirjoittamalla osoiteriviin localhost:8080/PHPyadmin. Täältä voi tutkailla tietokantoja, ja mitä tietokannoissa sijaitsee. phpMyAdminilla voidaan tarvittaessa ottaa talteen tietokannat, varmuuskopiointi on erittäin tärkeää verkkosivujen ylläpidossa. Valitsemalla tietokanta phpMyAdminista, klikkaamalla sen nimeä päästään tarkastelemaan mitä se tällä hetkellä sisältää. Kokoonpano sisältää muutamia tauluja tietokannassa, ja tauluista näkee, että niiden etuliite on aiemmin lisätty etuliite, joka nimettiin Joomla asennussivujen yhteydessä.

Joomlan sisäinen migraatio voidaan hoitaa jUpgrade komponentin avulla. Migraatio on siis julkaisusta seuraavaan siirtyminen tässä tapauksessa julkaisusta 1.5 julkaisuun 2.5 siirtyminen. jUpgrade on erillinen komponentti ja täytyy se aluksi asentaa. Mennään Joomla sivustoille ja laajennusten arkistoihin (*extension directories*) etsitään jUpgrade

ja ladataan se. Sivusto, josta jUpgrade löytyy, vaatii kirjautumisen, mutta itse komponentti ja kirjautuminen ovat molemmat ilmaisia. Tämän jälkeen puramme pakkauksen Joomla-kokoonpanon polkuun komponenttikansioon. Polku löytyy oletusasetuksilla asennetusta kokoonpanosta: C:/xampp/htdocs/myjoomla/components, nimet voivat vaihdella omien nimeämislogiikoiden mukaan. Kun komponentti on ladattu ja purettu koneelle, voimme kirjautua vanhaan Joomla kokoonpanoon ja mennä laajennusvälilehdelle (*extensions*). Valitsemme laajennusten alta tunnista (*discover*) kohdan. Oikeasta yläkohdasta löytyy tunnista painike, jonka avulla komponentti löytyy ja asennetaan. Tämä on yksi tapa asentaa lisäosia, toinen on asenna välilehdeltä, pakkausmuodossaan. Se tosin vaatii että laajennuksella on pakkauksessa asennustiedosto, jos tätä ei kuitenkaan löydy tunnista tapa toimii. Ennen jUpgrade komponentin käyttöä Joomla-kokoonpanon tulisi olla päivitettyinä viimeisimpään versioonsa. Tämä voidaan toteuttaa ohjauspaneelin etusivulta, päivitä-toiminnolla (*update*). Joomla oletuksena etsii päivityksiä, mutta ei asenna automaattisesti niitä. Tähän tarvitaan käyttäjän lupa. (Joomla 2013a.)

jUpgrade komponenttiin pääsee käsiksi komponentti-valikon alta. Komponenteilla on oma valikkonsa ja useimmiten oma ohjauspaneelinsa. Julkaisusta seuraavaan päivittäminen aloitetaan aloita päivitys -kuvalla (*start upgrade*), mutta tätä ennen tulisi tarkastella mitä kaikkea halutaan päivittävän. Oikealla ylhäällä, jossa laajennusten ja Joomla:n omat painikkeet sijaitsevat, on parametrit painike. Valitsemalla tämä, päästään valitsemaan mitä kaikkea halutaan päivittää (kuva 10). Oletusasetukset ovat aivan riittävät tähän tarkoitukseen, jos haluaa nimetä tietokannan tai pöydän tietynnimiseksi, tulee uudet nimet valita nyt. Samalla virheiden raportointi tulisi ottaa pois käytöstä (*disable error reporting*), sillä tätä toimintoa emme tarvitse. Tarvittaessa sen saa takaisin päälle kokoonpanon ollessa valmis. Virheiden raportoinnin takaisin ottaminen tapahtuu yleisten asetusten (*Global config*) valikosta. Palvelin välilehdeltä, jossa kohtaan virheiden raportointi (*error reporting*) valitaan haluttu vaihtoehto.



Kuva 10. jUpgrade komponentin parametrit

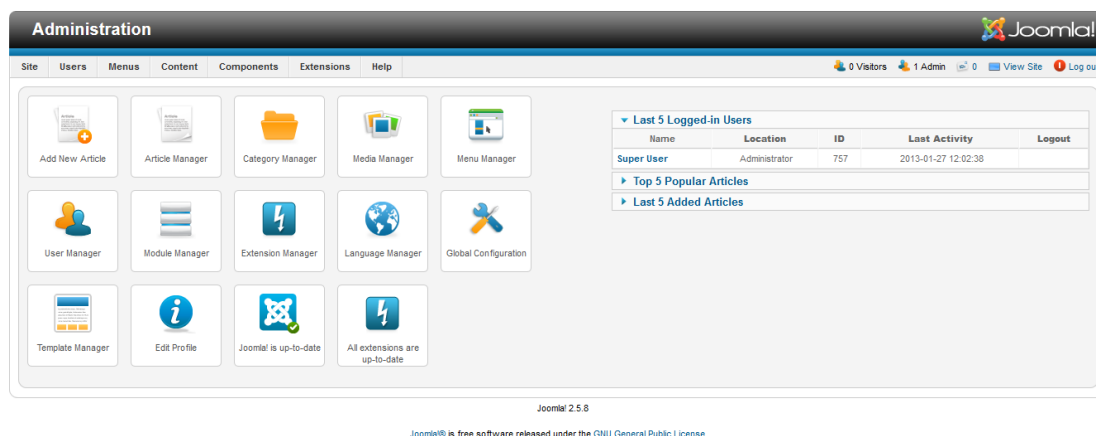
Mikäli komponentti ei toimi esim. migraatio jumittuu johonkin kohtaan, tulisi tarkastaa seuraavat asiat: päivitä Joomla kokoonpanosi viimeisimpään kokoonpanoon. Varmistetaan että käyttäjällä, jolla kirjaudutaan, on superhallinnoijan (*superadmin*) oikeudet. Varmistetaan että järjestelmä – Mootools plugin on käytössä. Tämä tehdään laajennusvalikon alta, pluginien hallinnointi kohdasta (Plug-in manager). Plugin on tarkalta nimeltään ”Systems – MooTools Plugin”. Tämä siis sallitaan (*enable*). Ja nyt voidaan kokeilla uudestaan.

Tämän jälkeen migraatio prosessi on valmis. Uutta kokoonpanoa voi tarkastella Joomla kokoonpanon nimellä. Samalla tulisi tarkistaa että käyttäjät, artikkelit, valikot, moduulit ja muut tärkeät osat säilyivät. Mikäli sivu ei toistu niin kuin pitäisi, niin ei tule vielä huolestua. Voi olla että aiemmassa kokoonpanossa oli käytössä jokin itse tehty, tai jonkun muu luoma template, jota ei ole luotu myöhemmälle julkaisulle. Tämä siis aiheuttaa lisätyötä Joomla kokoonpanon näkymiseksi, sillä nyt tulee asentaa tai luoda uusi template ja määrätä kaikki moduulit käyttämään kyseistä templatea. Toinen asia joka vaikuttaa ulkonäköön on media tiedostojen mahdollinen puuttuvuus. Komponentti ei nimittäin kopioi mediatiedostoja, joten nämä kansiot tulisi erikseen tuoda vanhas- ta kokoonpanosta. Voidaan tehdä kopioi-liitä -menetelmällä uuteen kokoonpanoon samoihin tiedostosijainteihin. Viimeinen tekijä on myös laajennusten toimimattomuus.

Jotkin laajennukset eivät siis enää toimi 2.5 julkaisussa, ja jos nämä laajennukset vaikuttivat sivuston ulkonäköön, ei sivusto enää välttämättä näytä samalta.

5.3 Joomla ohjauspaneeli

Joomlan ohjauspaneeli sisältää kaksi erilaista valikkoa. Toinen on back endin aloitus-sivulla näkyvä graafinen valikko, kun taas toinen on jatkuvasti back endissä esillä oleva navigointivalikko (kuva 11). Ylhäällä sijaitsevassa navigointivalikossa on useita vaihtoehtoja, jotka mahdollistavat helpon selaamisen ja graafinen valikko mahdollistaa mm. Joomlan helpon päivityksen, sillä se on automaattisesti esillä ja tarkistaa viimeisimmät päivitykset automaattisesti. Ennen päivityksiä tulisi kuitenkin aina tehdä varmuuskopiot.



Kuva 11. Joomla hallinnoinnin ohjauspaneeli

Tutkimme myös ohjauspaneelin lisäksi tässä vaiheessa Joomla-kokoonpanon yleisiä asetuksia. Valitsemme siis yleiset asetukset -kohdan (*global config*) ja tarkastelemme mitä muutoksia täältä voi tehdä. Kohdat ovat sivusto (*site*), järjestelmä (*system*), palvelin (*server*), oikeudet (*permissions*) ja tekstisuodattimet (*text filters*). Sivustovälilehdeltä löytyy yleisiä asetuksia, kuten sivuston nimi ja asetukset, metadata tiedot, sekä hakukone optimointi. Järjestelmä-välilehdeltä löytyy *debugging* (tarkoittaa virheiden paikantamista) ja välimuistin käyttöasetukset. Palvelin-välilehdeltä löytyy palvelin asetukset, sisältää mm. tietokannan asetukset, postiasetukset ja tietojen lähettämisasetukset. Oikeudet välilehdeltä määritellään mihin käyttäjien oikeudet riittävät, voiko esim. mennä muokkaamaan mitä vaan, vai pelkästään artikkeleja tai estää kokonaan joidenkin toimintojen toteuttaminen alhaisen tason käyttäjiltä. Tekstisuodattimet estävät jotkin tekstin muokkaukseen liittyvät asiat käyttäjäkohtaisesti. Kaikki ai-

emmat kohdat asetetaan haluamallaan tavalla tai jätetään sellaiseksi, koska nykyisellään asetukset ovat riittävät sisällönhallintajärjestelmän testaamiseen. (Joomla 2008.)

5.4 Joomla template ja CSS:n muokkaaminen

Tämän jälkeen olisi hyvä löytää itselle template. Tämä on siis osa, joka määrittelee suurilta osin sivustorakenteen, eli kuinka valikot näkyvät ja toistuvat, missä tekstiä ja moduuleja voi esiintyä sekä minkälainen sivuston muotoilu on. Template siis sisältää CSS-tiedoston, joka määrittelee kuinka HTML muotoillaan. Käymme läpi template käyttöönottamisprosessin ja joitain muokkaustoimintoja templateihin.

Template koostuu pääosin kolmesta komponentista, voi siis olla useampiakin ja useimmiten jotkut osat, kuten CSS, on jaettu pienempiin osiin. Osat ovat HTML, CSS ja PHP. Jollei haluta tehdä kuin muotoilumuutoksia nykyiseen, templateen, riittää CSS:n muokkaaminen. Halutessa lisätä moduuleja, tulee jo muokata useampaa osaa, moduuleja lisätessä muokataan PHP:tä. Templatejen CSS on monesti jaettu alaluokkiin. Useimmiten löytyy yleisesti tuettu CSS, sisältää selaimet kuten Chrome, FireFox ja Opera, kun taas IE:lle on tarkoitettu oma CSS:nsä. Tämä sen takia että vanhemmilla selaimilla selailtaessa sivua eivät sivut toistu samalla tavalla, koska esimerkiksi kulumien pyöristys täytyi toteuttaa lisäosilla vanhoissa IE-selaimissa. CSS joka on ladattu jostain toisesta lähteestä saattaa sisältää myös useaan osaan jaetun CSS:n. Voi siis mm. olla basic.css (sisältää yleiset muotoilut), color.css (sisältää värimaailman muotoilut) ja font.css (sisältää tekstienmuotoilut).

Joomla kokoonpanossa tulee automaattisesti muutama template sen mukana. Kyseiset templatet ovat seuraavanimisiä atomic, beez_20 ja beez5. Joomla sivustoilta on kuitenkin mahdollista myös latailla lisää templateja ja asentaa ne lisäosien tavoin. Templatet löytyvät www.joomla.org sivustolta laajennus (extend) valikon alta. Haku-toiminnolla saadaan templateja näkymään. Toinen hyvä lähde templateille on www.joomla24.com. Vaihtoehtoisesti löytyy myös joitain maksullisia ohjelmia, joilla voidaan toteuttaa Joomla templatet täysin graafisesti, esimerkki tällaisesta ohjelmasta on Artisteer. Käytettäessä templateja tulisi kuitenkin huomioida niiden yhteensopivuus. Joomla template 1.5.x ei mitä todennäköisimmin toimi oikein, Joomla 2.5.x julkaisulla. Sama koskee myös Joomla uusinta julkaisua 3.0.x.

Templatet asennetaan samalla logiikalla kuin muutkin lisäosat, eli laajennus välilehdeltä asennus tai tunnista toiminnolla. Sen jälkeen se voidaan ottaa käyttöön. Templaten käyttöönotto tapahtuu laajennus valikon alta template hallinnointi kohdasta (template manager). Tyylit välilehdeltä näemme mitkä templatet ovat tällä hetkellä käytössä (oletuksena 2.5.x kokoonpanossa ovat Beez_20 ja Bluestork). Käytössä on siis kaksi templatea, koska toinen on tarkoitettu sivuston back endin hallinnointiin, oletuksena Bluestork, kun taas Beez_20 template toimii sivuston front endin hallinnoinnissa. Sivustolla on siis aina käytössä vähintään kaksi templatea. Templaten perässä oleva tunniste kertoo onko kyseinen template tarkoitettu back endille vai front endille. Tunniste sivu (site) on front endille ja hallinnoija (administrator) on back endille. Muokkaa painikkeella (edit, oikealla ylhäällä) pääsee templatien muokkaustoimintoihin. Tästä voidaan mm. lisätä sivulle sen nimi ja kuvaus sivustosta, jotka tulevat näkyviin tai jopa sivuston logo. Nämä voidaan myös ottaa halutessa pois käytöstä, tai ainakin sivustolla käytettävä logo.

Seuraavaksi katsomme kuinka template toimii ja tarkastelemme hiukan kuinka CSS rakentuu. CSS:ään pääsee käsiksi kahdesta eri polusta, joko navigoimalla tietokoneella sijaitsevaan CSS kansioon (löytyy Joomla kokoonpanon template kansioista ja sieltä CSS kansioista esim. polku: C:/xampp/htdocs/myjoomla/templates/beez_20/CSS) tai navigoimalla Joomla back endiin, laajennus valikosta, template hallinnointi (template manager) ja täältä tyylit. Valitaan template jota halutaan käyttää, ja CSS tiedostot löytyvät muotoilutapojen (stylesheets) alta. Täältä voi myös muokata PHP:tä, muokkaa pääsivua tai virhesivua kohdasta (edit main page / error page).

CSS tulee avata Notepad++ ohjelmalla helpottaakseen koodin muokkausta (mikäli tiedostoon meni käsiksi Joomla back endistä, aukaistaan se selaimessa). Tämän jälkeen tarkastelemme mitä luokkia (kyseessä on siis luokkaohjelmoinnilla toimiva ohjelmointikieli) on jo olemassa, ja kuinka luoda uusia. Perusluokat ovat sellaisia kuten body, ul, table, blockquote ja erinäisiä paragraphin muotoja esim. p.readmore. Näille voidaan siis luoda halutessa lisää ominaisuuksia. Tai vaihtoehtoisesti voidaan luoda kokonaan uusia valitsimia.

Muokataan esimerkiksi valitsinta body. Luokaamme tähän background-image ominaisuus ja arvoksi lisäämme jonkin kuvan. Käytetään kuvana jotain lataamamme tai luoma kuvaa. Sijoitamme sen arvo kohtaan ja näemme vaikutukset sivulla. Ladattu tai

liitetty kuva tulisi siirtää templatien kuva kansioon, templatella on useimmiten oma images kansio johon voidaan siirtää kuvia. Muokkaamme beez_20 sijaitsevaa personal.css-tiedostoa. Selvittääkseen mitä CSS:ää tulee muokata, voi käyttää erinäisiä lisäosia ohjelmiin. Itse käytän selaimena Firefoxia, johon löytyy erittäin käytännöllinen työkalu nimeltä Firebug, jolla voi tarkastella verkkosivujen elementtejä ja näin saada selville mitä koodia muokata. CSS:n tulisi näyttää jokseenkin seuraavalta.

```
body {background-image: url(../images/kuva.jpg)}
```

Valitsin voi sisältää muita ominaisuuksia ja arvoja, mutta tässä on siis esimerkki muokkaus. Arvossa url viitataan tiedostoon kyseisessä sijainnissa ”..”-merkkäus polun alussa hyppää ylemmän tason kansioon. Taustakuva näyttää jokseenkin tökeröltä toistuessa useaan kertaan, jollei kuvaa ole tarkoitettu toistumaan useasti. Haluamme siis lisää sille ominaisuuksia, joten muokkaamme lisää. *Background-size:cover* saman valitsimen alla saa aikaiseksi kuvan venyttämisen ruudun mitalle, ja peittääkseen sen. Ominaisuudella ja arvolla *background-repeat:no-repeat*; estämme kuvan toistamisen useaan kertaan. Koodia ei tarvitse aina pyyhkiä olemattomiin, mikäli halutaan jatkossa palata siihen jossain vaiheessa. Useammin pätkän pois kommentoiminen onkin parempi vaihtoehto. Näin näkee mitä aiemmin oli muokattu ja mistä haluttiin eroon, lisäksi kommentoitiin voi lisätä omia mielipiteitä tai syitä miksi näin tehtiin. Kommentointi CSS:n sisällä tapahtuu parilla merkillä. ”/*”-merkintä avaa kommentin, kun taas ”*/” sulkee kommentin. Listan erilaisista CSS:n ominaisuuksista löytyy esimerkiksi sivulta <http://www.blooberry.com/indexdot/CSS/propindex/all.htm>.

5.5 Käyttäjien hallinnointi

Käyttäjiä hallinnoidaan pääasiallisesti Joomla:n back endistä, joissain tapauksissa, voit osin olla että täytyy mennä tietokantoihin, ja nollata sieltä käyttäjätiedot (mikäli esim. superhallinnoijat ovat unohtaneet salasansa). Tämä tapahtuu siis tietokannoista phpMyAdminilla. Etsitään haluttu käyttäjä käyttäjä-taulusta (users) ja muokataan hänen tietojaan. Salasana tulee laittaa MD5 salausmuodossa, ja käyttää sen vastiketta kirjautuessaan sisälle (pari tunnettua taulukossa 2). Tämän jälkeen kun salasana on muutettu tietokantoihin, kirjaudutaan kyseisellä salasanalla ja vaihdetaan se tämän jälkeen johonkin toiseen käyttäjähallinnasta.

Taulukko 2. Salasanat ja niiden salattu muoto

Normaalimuoto	MD5 salaus
admin	21232f297a57a5a743894a0e4a801fc3
secret	5ebe2294ecd0e0f08eab7690d2a6ee69

Käyttäjien hallinnoinnista on mahdollista muokata nykyisiä käyttäjiä, lisätä uusia tai poistaa vanhoja. Kaikki tämä tapahtuu Joomla:n navigointivalikosta, joka sijaitsee siis back endissä sivun ylälaudassa. Valitaan käyttäjät (users), josta käyttäjien hallinnointi (user management). Luomme esimerkkinä yhden uuden käyttäjän, joka on hallinnoija tason käyttäjä, eli lähes samoilla oikeuksilla kuin superhallinnoija. Luodaan käyttäjä täyttämällä kentät, ja asettamalla alkuarvot, kyseinen käyttäjä voi kuitenkin muuttaa asetuksiaan myöhemmin, mikäli hänellä riittää oikeudet tähän. Kun koko sivu on täytetty, luodaan käyttäjä tallentamalla tiedot. Tämän jälkeen sen toimivuuden voi testata kirjautumalla ja täten tarkistamalla toimivuus. (kuva 12.)

Kuva 12. Käyttäjän luominen

Käyttäjän oikeuksien muokkaaminen tapahtuu muokkaamalla olemassa olevaa käyttäjää ja pudottamalla hänen oikeudet alemmalle tasolle. Kirjautuneet (registered) käyttäjät voivat kirjautua front endille, vaikka eivät voikaan muokata artikkeleita, mutta pääsevät näin sisältöön joihin kaikki muut eivät pääse. Kirjoittajat (author) pystyvät luomaan uutta sisältöä, vaikkakin ei suoraan, vaan tämä sisältö tarkastetaan myöhemmin korkeamman oikeuksien omaavan tahon toimesta, ja näin joko julkaistaan tai ei. Editorit pystyvät muokkaamaan sisältöä, mutta eivät pysty muuttamaan sisällön tilaa, mikäli sisältö on julkaisematta, eivät he sitä pysty julkaisemaan. Julkaisijat (publishers) tason käyttäjät pystyvät luomaan, muokkaamaan ja julkaisemaan sisältöä.

Muut käyttäjätasot on tarkoitettu Joomla back endin muokkaukseen. Vastaa front endin julkaisijoita, muutamilla lisäoikeuksilla (mutta vain back endin puolella). Pystyy myös poistamaan joitain osioita, sekä kategorioita. Hallinnoijat saavat melkein kaikki oikeudet, poikkeuksena on kuitenkin globaalien asetusten muokkaaminen ja templatien muokkaaminen. Kaikkeen muuhun he pystyvätkin. Superhallinnoijat pystyvät kaikkiin sivustoon liittyviin muokkauksiin. Huomioitavaa on että vain superhallinnoijat pystyvät luomaan superhallinnoijia, ja muokkaamaan niitä. (Joomla 2011)

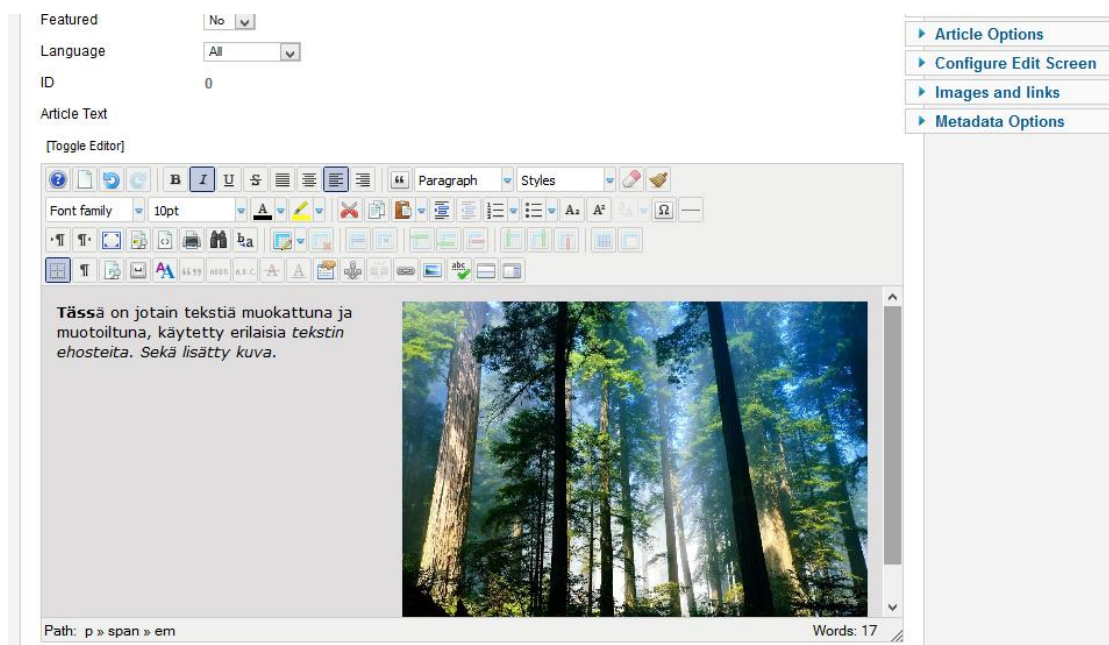
5.6 Sisällön luominen

Sisältöä luodessa tulee ottaa huomioon parikin asiaa: täytyy luoda moduulit, jotka saavat kyseisen sisällön ja valikot jotka johtavat sisältöön. Järjestyksellä on myös huomattavaa merkitystä luotaessa sisältöä Joomla kokoonpanoon, sillä jotkin elementit vaativat jo toisten olemassa olon, ennen kun niitä voi luoda. Elementtien luontijärjestys tulisi olla seuraava: kategoriat, artikkelit, valikot, template ja moduulit. Tämä siis sen takia että artikkelit vaativat kategorioita joihin sijoittua. Valikot saattavat johtaa artikkeleihin (tai kategorioihin), vaaditaan ne jo siis pohjalle. Template luodaan ennen moduuleja sen takia että moduuleja luodessa tulee valita niiden sijainti templatesta. Elementtien alkuperäinen luominen tapahtuu siis lähestulkoon päinvastaisessa järjestyksessä, mitä niiden toistuminen on sivustolla.

Tarkasteltaessa asennettua Joomla kokoonpanoa, näemme että se sisältää jo moduuleja ja valikkoja. Ei kuitenkaan vielä sisältöä, jollei tätä tuotu toisesta kokoonpanosta tai asennettu esimerkki rakennetta. Luomme siis aluksi pari valikkoa lisää, jotka johtavat sisällölle (luomme siis artikkeleita sisällöksi). Voimme siis ajatella että artikkelit ovat alisivuja. Valikkojen luonti tapahtuu valikko kohdasta (menu) ja sieltä valittuna valikoiden hallinnointi (menu manager). Valitsemme päävalikon (main menu), jonka alla on kohta aloitus (home). Tässä kohtaa valitsemme uusi painikkeen avulla uuden valikon luomisen.

Uuden valikon luomisessa tulee tietää, mikä tyyppi tulee valita. Artikkelit vaihtoehdot johtavat sisältöön, jonka jälkeen voidaan määrätä mihin sisältöön, muut vaihtoehdot johtavat selitettyyn sisältöön (esim. käyttäjät ja uutiskirjeet). Täältä voidaan myös

valita luotavaksi sivu, joka näyttää esimerkiksi viimeisimmät artikkelit jostain kategoriasta, joihin artikkelit tulisi luokitella kyseisten toimintojen viimeistelemiseksi. Tämä auttaa myös sivujen ylläpidossa, mikäli halutaan muokata artikkeleja vain yhdestä kategoriasta. Helpottaa myös sisällön paikallistamisessa. Jos yrittää luoda valikon, joka johtaa tiettyyn artikkeliin tulee virhe, sillä artikkeleita ei ole vielä luotuna. Luomme siis esimerkki artikkelin aivan ensiksi, jonka kopioimme ja siirrämme muuttaman toiseen kategoriaan. Artikkelien luomisessa käytetään WYSIWYG-editoria (*What You See Is What You Get*), joka siis näyttää artikkelin sellaisena kun se sivulla onkin, vaikka pienellä varauksella. Artikkelin näyttää enimmäkseen editorissa luodulta, mutta voi saada tunnuspiirteitä CSS:stä, sillä mm. fonttiperhe ja fontin koko voidaan määrittää jo CSS ohittamaan artikkelin sisäiset muutokset. Artikkeleita on helppo luoda ja muokata editorilla, tarvittaessa voi jopa ladata uuden editorin (yksi suositelluista olisi sisältöön suoraan linkittävä editori esim. JCE ladattavissa Joomla:n sivuilta lisäosista). JCE:n avulla sisältöön linkittäminen voidaan suorittaa graafisesti, kun taas ilman sitä pitäisi käyttää artikkeleiden linkkejä. Jos haluaa nähdä tekstin HTML:nä, on se myös helppo muuttaa tähän muotoon ja muokata koodina näin halutessaan. (Kuva 13.)



Kuva 13. Artikkelin luonti JCE-editorilla

Kategorioiden luominen tapahtuu sisältö kohdasta, kategorioiden hallinnoinnista (teemme siis tämän ennen artikkelien luontia, jotta emme joudu myöhemmin siirtelemään artikkeleja kategorioihin). Josta löytyy myös artikkelien hallinnointi. Luodaan

kategoria uusi-painikkeella, sille otsikko, alias (hakukone optimisoinnin ollessa käytössä). Täytämme myös parametrit, kuten metatiedot ja muut. Tilaksi valitaan julkaistu (published). Tämän jälkeen luodaan loput kategoriat. Kun kategoriat on luotu, siirrymme luomaan artikkeleita samalla metodilla. Mutta nyt voimme lisäksi valita artikkeleille kategoriat johon ne lukeutuvat. Näiden luotua on mahdollista palata valikoiden hallinnointiin ja saada valikot tehtyä, jotka vaativat sisältöä toimiakseen (artikkelityyppiset valikot). Luomme yhden yksilöartikkeli-valikon (single article) ja yhden kategoria blogimallisen -valikon (category blog). Valitsemme valikoille kohdeartikkelin ja kategorian ja julkaisemme ne.

Artikkelien, kategorioiden ja valikoiden muokkauksen ja luomisen lisäksi tulee osata luoda moduuleja ja käyttää niitä. Moduulit ovat suuriosa CMS:n hallintaa. Moduuleilla määrätään mitä komponentteja valikoita tai muita osia sivuilla näkyy. Moduuleita hallinnoidaan Joomla sivuston back endistä. Moduulit löytyvät laajennus valikon alta moduulien hallinta kohdasta. Moduuleja voidaan luoda ja ottaa käyttöön samalla logiikalla kun artikkeleitakin. Moduulit voidaan siis julkaista tai jättää julkaisematta (eli poistaa käytöstä toistaiseksi säilyttäen kuitenkin asetukset) tai sitten ne voidaan kokonaan poistaa roskakori vaihtoehdolla (trash). Jos on asentanut lisäosia, jotka ovat komponentteja, usein komponentin asennuksen yhteydessä komponentti luo itsestään moduulin, joka tosin jätetään julkaisemattomaan tilaan. Näin käyttäjä näkee, mikä moduuli komponentista tulee. Joomla julkaisussa ilman asennettuja lisäosia moduulivaihtoehdot ovat artikkeleita, breadcrumbseja (navigointipalkki, jonka avulla voi peruuttaa jollekin tietylle sivulle), valikkoja, hakupalkkeja, kielenvaihto moduuleja, tekstiä (custom HTML) yms.

Moduulia luodessa tulee moduulille antaa nimi (title), valitaan näytetäänkö moduulin nimi ja mihin moduuli sijoitetaan. Moduulin sijoittaminen valitaan templatejen sijaintien perusteella. Samalla voidaan laittaa julkaisuaika moduulille. Moduulille annetaan kieli, monikielisyys hoidetaan siis tämän avulla (jollei käytä kolmannen osapuolen komponentteja). Moduulille voidaan myös määrätä muotoiluja templatien pohjalta. Muotoilut hoidetaan lisäasetusten alta antamalla moduulille, luokkaan liittyvä liite (*module class suffix*). Voidaan siis luoda templateen uusi luokka, jos ei ole vielä olemassa käypää vaihtoehtoa. Alhaalta vasemmalta, moduulin luomissivulla, voidaan valita moduulin toistuvuus, millä sivuilla moduuli siis halutaan näyttää. Esimerkiksi

kielenvaihto moduuli, halutaan mahdollisesti kaikille sivuille, mikäli sivuille päätyy hakukoneen kautta, ja halutaan vaihtaa kieleksi toinen tunnettu kieli.

5.7 Kielten lisääminen

Kielten lisääminen tapahtuu laajennus valikon alta, kohdasta kielten hallinnointi (*language manager*). Täällä nähdään asennetun kokoonpanon kieli, ja asennetaan uudet kielet. Yhden kielen tulee aina olla oletuskielenä. Oletuskieli olisi hyvä valita sivuston kohderyhmän mukaan. Sivustolta löytyy oikealta ylhäältä asenna kieliä -painike (*install languages*), jonka avaamalla saadaan lista jo luotuja käännöksiä Joomlaalle. Asennetut kielet siis lähinnä kääntävät joko yleisten komponenttien sanoja tai termejä, mutta eivät itse artikkeleja. Jos haluttu kieli löytyy listalta, voidaan se yksinkertaisesti asentaa valitsemalla kieli tai kielet. Edetään asenna painikkeella valintojen jälkeen. Tämän jälkeen tulisi saada viesti että asennus suoritettu ja kielen tulisi näkyä sen hallinnoinnissa.

Tapauksissa, jossa käännöskieltä ei löydy, voidaan itse luoda oma käännös. Tämä voidaan toteuttaa luomalla kopiot jo asennetusta kielestä (kopioidaan kansio) ja aletaan kääntää osia, ja vaihdetaan kaikista osista niiden tiedot. Tiedot jotka vaihdetaan, on kielen kirjoittaja, kielitunnukset, käytetään samaa nimeämislogiikkaa, ja muut tärkeät tiedot, jonka jälkeen aletaan kääntää itse ini-tiedostoja, joissa käännökset sisältävät. Tämän jälkeen kieli voidaan löytää tunnista vaihtoehdolla, jota käytetään laajennusten löytämiseen. Kun kieli on asennettu, mennään takaisin kielten hallintaan, josta valitsemme välilehden sisältö (*content*). Huomaamme että toistaiseksi täällä näkyy vain 1 kieli (asennetun kokoonpanon kieli) ja tänne tulee lisätä uudet kielet. Valitaan siis uusi vaihtoehto ja lisäämme kielen, tätä ennen tulisi katsoa nimeämislogiikka jo aiemmin olemassa olevasta kielestä ja laittaa ylös pienten ja suurten kirjaimien sijainti. Eli mitä tunnisteet ja niiden merkkityla (suuri vai pieni kirjain). Lisätään täsmälleen samat tiedot, mutta omasta kielen tunnisteesta ja näin ollen saadaan sisällölle uusi kieli.

5.8 Käyttämäni lisäosat

Lisäosilla on tarkoitus nopeuttaa jatkossa joidenkin osien toteuttamista ja hallinnointia. Lisäosia valitessa tulisi huomioida mitä ominaisuuksia järjestelmän kokoonpanos-

ta sillä hetkellä puuttuu ja lisätä niitä. Lisäosilla voidaan tehdä joko kokonaan uusia asioita, tai ne voivat vain sisältää oikopolkuja jo olemassa oleviin asioihin. Voitaisiin esimerkiksi lisätä täysin uusi komponentti, jolla oma toiminto tai simppele korvaava koodi, jolla muokataan esimerkiksi sähköpostien lähettämistä. Lisäosat riippuvat siis paljolti omasta tarpeesta. Selitän seuraavaksi poimimani lisäosat kokoonpanoon ja mitä ne pääosin tekevät.

Käytin siis lisäosia Falang, jNews ja sekä slideshowCK. Falangilla hoidetaan sivuston käännös säästämällä artikkelien määrää ja moduulien lisäystä. Falang komponentti on käännöskomponentti, jolla voidaan kääntää valikot, artikkelit ja moduulit Joomla kokoonpanoissa asennetuille kielille. Käännökset täytyy tosin tehdä vielä itse, mutta kieli on selkeää ja ymmärrettävissä kun taas automatisoiduilla käännöskoneilla kieli olisi vaikealukuista tai toisinaan jopa käsittämätöntä. Komponentin avulla vältetään myös uusien osien luomiselta, sillä käännöksissä käytetään aiemmin luotuja moduuleja artikkeleita ym. Komponentti lisääkin tietokantaan käännökset aiemmille osille ja kaikki yhteneväisyydet ja linkitykset säilyvät, ainut että teksti tulee kääntää (haluttujen kielin täytyy olla lisätty Joomla kokoonpanoon, muuten niitä ei lisätä käännös tietokantoihin, voi lisätä jälkeempään). SlideshowCK lisäosan avulla voidaan luoda kuva- tai video-galleria erittäin helposti. Voi myös sekoittaa molempia keskenään. Gallerian käyttö tapahtuu valitsemalla halutun tiedostotyyppin ja sittemmin paikantamalla sen mediakansioista. Lisäosan lisääminen tapahtuu moduulin luonnilla ja lisäosan luomisessa annetaan paljon asetusvaihtoehtoja gallerian yksilöllistämiseen. Voidaan myös käyttää luokkamuotoilua CSS:n avulla galleriassa. JNews komponentilla toteutettiin sivuston uutiskirjeiden luominen ja hallinnointi. Komponentti mahdollistaa uutiskirjeiden hie- man laajemman muokkaamisen ja automatisointi on paljon yksinkertaisempaa, mitä Joomla kokoonpanossa olevalla uutiskirjekomponentilla on. Komponenttiin lisätään uutiskirjeiden tilaajat, ja voidaan jakaa näille ryhmiä ja luokitella tilaajia. Uutiskirjeitä voidaan siis lähetellä ryhmittäin. Lisäosalla saadaan myös luotua erilaisia uutiskirje-pohjia, joilla uutiskirjeiden yksilöllistäminen tai ryhmille suuntaaminen helpottuu. JNews komponentti omaa myös tehtävienajoitus ominaisuuden, joten uutiskirjeet on mahdollista ajoittaa jaettavaksi tietyinä ajankohtana. Komponentti tuo siis lisää jous- tavuutta uutiskirjeiden jakamiseen.

6 PALVELIMELLE MUUTTAMINEN

Sivuston palvelin muotoon muuttaminen on yksinkertaista. Se tapahtuu kolmessa eri prosessissa; tietokannan siirrossa, kokoonpanon kopioinnissa, ja asetusten muokkaamisessa. Tämän jälkeen sivuston tulisi toimia palvelimella.

Aluksi siis tulee kirjautua phpMyAdmin:iin ja kopioida tietokanta kokonaisuudessaan. Tämän jälkeen tulisi se tuoda palvelimella sijaitsevaan kirjastoon ja lisätä se sinne. Tämän jälkeen tulee saada kansiot ja tiedostot järjestelmästä palvelimelle. Järkevintä olisi pakata Joomla kokoonpano kokonaisuudessaan, lähettää se palvelimelle ja purkaa siellä. Täten vähennetään lähetysaikaa ja varmistetaan että kaikki tiedostot saadaan perille toimivina. Seuraavana on asetusten muokkaaminen palvelin kelpoiseksi. Tämä tapahtuu muokkaamalla Joomla kansiossa sijaitsevaa configuration.php:tä. Täältä kaikki viittaukset localhostiin tulee vaihtaa palvelinmuotoisiksi, niin itse tiedot kuin polkuviittaukset. (Taulukko 3, Joomla2012a.)

Taulukko 3. Configuration.php muokattavat kohdat (Joomla 2012a)

```
var $log_path = '/var/www/vhost/domain.com/home/html/logs';
var $tmp_pth = '/var/www/vhost/domain.com/home/html/tmp';
var $live_site = '';
var $host = 'name your remote host';
var $user = 'your_user_db_name';
var $db = 'your_db_name';
var $password = 'your_db_password'
```

7 YHTEENVETO

Opinnäytetyötä lähdettiin tutkimaan, koska MHG Systems Oy:llä oli tarvetta päivittää verkkosivujaan uuteen malliin sisällönhallintajärjestelmällä. Tämä ei yksinään kuitenkaan riittänyt, sillä näin ollen jäisi pois tyhjän kokoonpanon tutkiminen. Samalla siis päätettiin asentaa ja tutkia tyhjää sisällönhallintajärjestelmää.

Sisällönhallintajärjestelmän alkuaskeleet voivat tuntua erittäin vierailta. CMS:n kokonaisuuden hahmottaminen vie kuitenkin oman aikansa. Kun on oppinut sisällönhallintajärjestelmän hierarkian ja saanut lopullisen kokoonpanon tehtyä, on sen käyttäminen erittäin helppoa. Tämän jälkeen muokkausten tekeminen ja aika, joka itse järjestelmän muokkaukseen menee, on mitätöntä, sillä lähes kaikki toiminnot on saatu automatisoitua.

Sisällönhallintajärjestelmää on suositeltavaa käyttää, mikäli ei usko että teknistä osaamista on tarpeeksi ilman CMS:ää. Myös suurten sivustojen olisi syytä käyttää sisällönhallintajärjestelmää, sillä se auttaa sisällön järjestelyssä ja saatavuudessa. Samalla myös voidaan hiukan rajoittaa tiedostojen rönsyilyä. Sisällönhallintajärjestelmä tulisi valita aiemman osaamisen ja tietojen pohjalta. Eroja järjestelmien välillä on, vaikkakin toimintaperiaate on sama,. Erityisesti kolmannen osapuolen komponenteissa eroja voi löytyä paljon. Hyötyjen kannalta CMS:n käyttäminen olisi kannattavaa

Ongelmakohtia voivat olla järjestelmän vanhennevuus, tarvitsee päivittää uudempaan, joka aiheuttaa lisäosien mahdollisen uusimisen. Jotkin lisäosat eivät kantaudu seuraavalle julkaisulle, ja näin ollen täytyy joitain asioita tehdä uusiksi. Tämä oli myös yksi syy, miksi tein tätä opinnäytetyötä yhteistyössä MHG Systems Oy:n kanssa. Toinen suuri kompastuskivi, mikä tosin on muissakin verkkosivujen hallinnoinnissa, on varmuuskopiointi. Varmuuskopiointi on erittäin tärkeää, sillä joidenkin päivitysten tai asetusten myötä, jokin komponentti ei välttämättä toimi niin kuin pitää. Tämä vuorostaan aiheuttaa kaiken alusta aloittamisen ilman riittäviä varmuuskopioita. Lisää kompastuskiviä voi aiheuttaa käyttäjien opastaminen uuden sisällönhallintajärjestelmän käytössä. Tästä voi myös syntyä yllättäviä lisäkoulutuskustannuksia. Hallinnoijan jo osatessa käyttää järjestelmää, olisi hänen hyvä tutustuttaa uudet käyttäjät järjestelmään. Tai vähintään yhden henkilön hyvin, joka voi taas jatkossa opastaa muita.

Käytännönosuus onnistui hyvin, sillä saatiin toteutettua halutut toiminnot. Yritys ei kuitenkaan ottanut rakennetta aivan vielä käyttöön, sillä ulkoasun muuttaminen on vielä edessä, alkuun on kuitenkin tehty mallirakenne niin kuville kuin rakenteellekin. Rakenne säilyy samana ja yritys jatkaa tämän pohjalta, muokaten lähinnä visuaalista ilmettä. Haasteellisinta käytännön osuudessa olikin visioiden samaistaminen minun ja yrityksen välillä. Alkuun ongelmia tuotti myös käytettyjen ohjelmien toimiminen, joista vastaan tulleita käsittelinkin, mutta tästä eteenpäin projektin eteenpäin vieminen

onnistui enimmäkseen ongelmitta. Kaiken kaikkiaan työn tekninen toteutus oli onnistunut. Tyhjän kokoonpanon asentamisessa, ongelmien ennakointi ja niiden ennaltaehkäisy oli mahdollista, ja täten välttiin turhalta ongelmanratkaiselta asentamisvaiheissa.

LÄHTEET

Boiko, Bob 2005. Content Management Bible. New York: Hungry Minds, Inc.

Built With 2009. CMS Usage Statistics. WWW-artikkeli.
<http://trends.builtwith.com/cms>. Päivitetty 18.1.2013. Luettu 20.1.2013

Creative Sites 2009. Content management system (CMS). WWW-dokumentti.
<http://www.creativesites.eu/content-management-system-cms-joomla/>. Luettu 20.1.2013

GNU 2007. GNU General Public License. WWW-dokumentti.
<http://www.gnu.org/licenses/gpl.html>. Päivitetty 9.2.2013. Luettu 23.1.2013

Heng, Christopher 2012. How to Make / Create a Website: The Beginner's A-Z Guide. WWW-dokumentti.
<http://www.thesitewizard.com/gettingstarted/startwebsite.shtml>. Päivitetty 30.3.2012. Luettu 16.1.2013

HTML.net 2006. Lesson 2: How does CSS work? WWW-artikkeli.
<http://www.HTML.net/tutorials/css/lesson2.php>. Ei päivitystietoja. Luettu 16.1.2013

Joomla 2008a. Control Panel. WWW-artikkeli.
<http://help.joomla.org/content/view/1440/296/>. Päivitetty 3.2.2008. Luettu 23.1.2013

Joomla 2008b. Technical Requirements. WWW-dokumentti.
<http://www.joomla.org/technical-requirements.html>. Ei päivitystietoja. Luettu 16.1.2013

Joomla 2011. User Group Access levels explained in simple terms. WWW-dokumentti.
http://docs.joomla.org/User_Group_Access_levels_explained_in_simple_terms. Päivitetty 25.10.2011. Luettu 29.1.2013

Joomla 2012a. Copying a website from localhost to a remote host. WWW-dokumentti
http://docs.joomla.org/Copying_a_website_from_localhost_to_a_remote_host. Päivitetty 1.9.2012. Luettu 30.1.2013

Joomla 2012b. Installing an Extension. WWW-dokumentti.
http://docs.joomla.org/Installing_an_extension. Päivitetty 4.8.2012. Luettu 22.1.2013

Joomla 2012c. Installing Joomla! WWW-dokumentti
http://docs.joomla.org/Installing_Joomla!. Päivitetty 12.12.2012. Luettu 24.1.2013

Joomla 2012d. Migrating from Joomla 1.5 to Joomla 2.5. WWW-dokumentti
http://docs.joomla.org/Migrating_from_Joomla_1.5_to_Joomla_2.5. Päivitetty 11.1.2013. Luettu 21.1.2013

Joomla Designs 2009. Joomla sections categories and articles. WWW-dokumentti
<http://www.joomladesigns.co.uk/Tutorials/Joomla/Joomla-sections-categories-and-articles.HTML>. Ei Päivitystietoja. Luettu 30.1.2013

Kyrnin, Jennifer 2011. What is HTML. WWW-dokumentti
<http://webdesign.about.com/od/HTMLxHTMLtutorials/a/what-is-HTML.htm>. Päivitetty 10.4.2012. Luettu 16.1.2013

Lynch, Patrick J. & Horton, Sarah 2009. Site Structure. WWW-artikkeli.
<http://webstyleguide.com/wsg3/3-information-architecture/3-site-structure.HTML> .
 Päivitetty 22.10.2010. Luettu 14.1.2013

Microsoft 2012. User Interface Wizard Behavior. WWW-dokumentti.
<http://msdn.microsoft.com/en-us/library/aa372394%28VS.85%29.aspx>. Ei päivitystietoja. Luettu 23.1.2013

PHP.net 2010. Introduction to PHP: What is PHP? – Manual. WWW-dokumentti.
<http://www.php.net/manual/en/intro-what-is.php>. Päivitetty 8.2.2013. Luettu 23.1.2013

Reyes, Joel 2009. How to Evaluate What CMS to Use. WWW-dokumentti.
<http://sixrevisions.com/web-development/how-to-evaluate-what-cms-to-use/>. Ei päivitystietoja. Luettu 20.1.2013

Sundar, Veera 2009. How to Change XAMPP Server Port. WWW-dokumentti.
<http://veerasundar.com/blog/2009/07/how-to-change-xampp-server-port/>. Ei päivitystietoja. Luettu 23.1.2013

Treble Click 2009. XAMPP MySQLdeactivated Fix. WWW-artikkeli.
<http://trebleclick.blogspot.fi/2009/01/xampp-mysql-deactivated-fix.html>. Ei päivitystietoja. Luettu 23.1.2013

W3Schools 2008. Tryit Editor v1.6. WWW-ohjelma.
http://www.w3schools.com/HTML/tryit.asp?filename=tryHTML_intro_. Ei päivitystietoja. Luettu 16.1.2013

W3Schols 2012a. HTML Reference - (HTML5 Compliant). WWW-dokumentti.
<http://www.w3schools.com/tags/default.asp>. Ei päivitystietoja. Luettu 16.1.2013

W3schools 2012b. Introduction to HTML. WWW-artikkeli.
http://www.w3schools.com/html/html_intro.asp. Ei päivitystietoja. Luettu 16.1.2013

You!Joomla 2012. What is a Joomla Extension? WWW-dokumentti.
<http://www.youjoomla.com/joomla-definitions-knowledgebase/what-is-a-joomla-extension.HTML>. Ei päivitystietoja. Luettu 22.1.2013

Your Dictionary 2004. Apache – technical definition. WWW-dokumentti
<http://computer.yourdictionary.com/apache>. Ei päivitystietoja. Luettu 23.1.2013

Softaculous 2012. Default Templates Explained. WWW-dokumentti.
http://demos2.softaculous.com/CMS_Made_Simple/index.php?page=default_template
 s. Ei päivitystietoja. Luettu 30.1.2013

JOOMLAN JÄRJESTELMÄVAATIMUKSET ERI JULKAISUISSA (JOOMLA 2008B)

Requirements for Joomla 2.5.x, 1.7.x and 1.6.x

Software	Recommended	Minimum	More information
PHP	5.3 +	5.2.4 +	http://www.php.net
MySQL	5.0.4 +	5.0.4 +	http://www.mysql.com
Supported Web Servers:			
Apache (with mod_mysql, mod_xml, 2.x + and mod_zlib)		2.x +	http://www.apache.org
Nginx	1.1	1.0	http://wiki.nginx.org/
Microsoft IIS	7	7	http://www.iis.net

Requirements for Joomla 3.x

Software	Recommended	Minimum	More information
PHP (Magic Quotes GPC off)	5.3.1 +	5.3.1 +	http://www.php.net
Supported Databases:			
MySQL (InnoDB sup- port required)	5.1 +	5.1 +	http://www.mysql.com
MSSQL	10.50.1600.1+	10.50.1600.1+	http://www.microsoft.com/sql
PostgreSQL	8.3.18 +	8.3.18 +	http://www.postgresql.org/
Supported Web Servers:			
Apache (with mod_mysql, 2.x + mod_xml, and mod_zlib)		2.x +	http://www.apache.org
Nginx	1.1	1.0	http://wiki.nginx.org/
Microsoft IIS	7	7	http://www.iis.net

JOOMLAN JÄRJESTELMÄVAATIMUKSET ERI JULKAISUISSA (JOOMLA 2008B)

Requirements for Joomla 1.5.x

Software	Recommended	Minimum	More information
PHP*	5.2 +	4.3.10	http://www.php.net
MySQL**	4.1.x +	3.23	http://www.mysql.com
Supported Web Servers:			
Apache***			
(with mod_mysql, mod_xml, 2.x + and mod_zlib)		1.3	http://www.apache.org
Microsoft IIS****	7	6	http://www.iis.net

*Do not use PHP 4.3.9, 4.4.2 or 5.0.4. These releases have known bugs that will interfere with installation. Zend Optimizer 2.5.10 for PHP 4.4.x also has serious bugs and you should ask your host to upgrade to a newer version. Joomla! versions 1.5.15 + are compatible with PHP 5.3. Note that the OpenID library is not yet compatible with PHP 5.3.

**Joomla is not yet compatible with MySQL 6.x.

***In order to use SEO URLs, you will need to have the Apache mod_rewrite extension installed.

****For Microsoft IIS (depending on your setup) you may need the following:

- PHP 5.2 - Installation instructions
- MySQL 5.1 - Installation instructions
- Microsoft URL Rewrite Module - Required for SEO URLs only. For more information, click here. For information about using ISAPI, click here.
- FastCGI - Download for IIS6. Download for IIS7.

For further assistance using Microsoft IIS, visit the Joomla IIS forum.

7.1 Configuration Options

If installing on a local computer, there are a number of packages that will help you get set up quicker than individual installations:

- LAMP (Linux) - Most Linux distributions come with a pre-configured LAMP server.
- WAMP (Windows) - For more information, visit <http://www.wampserver.com>
- MAMP (Macintosh) - For more information, visit <http://www.mamp.info>
- XAMPP (Multi-platform) - Not for live sites. For more information, visit <http://www.apachefriends.org/en/xampp.HTML>

If you need other options, we recommend visiting our installation forums for more information.