

Jouni Nygård

WWW-SIVUJEN SISÄLLÖNHALLINTAJÄRJESTELMÄT

Esimerkkinä henkilökohtainen www-portfolio

Opinnäytetyö

KESKI-POHJANMAAN AMMATTIKORKEAKOULU

Mediatekniikan koulutusohjelma

Marraskuu 2011

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Yksikkö Ylivieska	Aika 8.11.2011	Tekijä/tekijät Jouni Nygård
Koulutusohjelma Mediatekniikka		
Työn nimi WWW-SIVUJEN SISÄLLÖNHALLINTAJÄRJESTELMÄT Esimerkkinä henkilökohtainen www-portfolio		
Työn ohjaaja Mikko Himanka		Sivumäärä 35
Työelämäohjaaja		
<p>Opinnäytetyön aiheena oli perehtyminen sisällönhallinnan teoriaan ja järjestelmiin. Työ rajattiin käsittelemään www-sisällönhallintaa ja järjestelmien rajapintana toimi PHP-kieli. Opinnäytetyön ensimmäisissä luvuissa käytiin läpi sisällönhallinnan määritelmät, jonka jälkeen tarkasteltiin aiheeseen liittyviä teknologioita ja ohjelmointikieliä.</p> <p>Käytännönosuus koostui sopivan järjestelmän valinnalla alustaksi web-portfolioille. Järjestelmän valinta tapahtui luomalla vaatimusmäärittely ja vertailemalla eri järjestelmiä keskenään. Portfoliolle sopivaksi järjestelmäksi päätettiin lopulta Wordpress ja sivusto siirtyi opinnäytetyön jälkeen jatkokehitykseen.</p>		

Asiasanat

sisällönhallinta, sisältö, internet, julkaisu, toiminnot, käyttöliittymä.

ABSTRACT

CENTRAL OSTROBOTHNIA UNIVERSITY OF APPLIED SCIENCES Ylivieska	Date 8.11.2011	Author Jouni Nygård
Degreeprogramme Media Technology		
Name of thesis CONTENT MANAGEMENT SYSTEMS FOR WEBSITES Case personal web portfolio		
Instructor Mikko Himanka	Pages 35	
Supervisor		
<p>The subject of the thesis was familiarization with the theory of content management and systems. The task was restricted to web content management and system interfaces that utilised the PHP scripting language. In the initial chapters of the thesis, we glanced at the definitions of content management and then examined the related technologies and programming languages.</p> <p>The practical part of the thesis consisted of selecting a suitable platform for a web-portfolio. The selection process started by creating a specification of requirements and then comparing different systems with each other. This resulted in Wordpress being chosen as a suitable system for the portfolio at which point the existing site was then moved into further development.</p>		

Key words

content management, content, internet, publishing, functionality, user interface.

TIIVISTELMÄ

ABSTRACT

SISÄLLYS

1 JOHDANTO	1
2 SISÄLLÖNHALLINTA	3
2.1 Sisällönhallinnan määritelmiä	3
2.2 Sisällönhallintajärjestelmä	6
3 WWW-SISÄLLÖNHALLINNAN LÄHTÖKOHDAT	9
3.1 Staattiset ja dynaamiset sivut	10
3.2 Web-kehitysteknologiat	12
3.2.1 HTML -sivunkuvauskieli	13
3.2.2 CSS -kieli	14
3.2.3 PHP -ohjelmointikieli	15
3.2.4 SQL	16
3.2.5 WampServer	16
3.2.6 phpMyAdmin	18
4 SISÄLLÖNHALLINTAJÄRJESTELMÄN TOIMINTA	20
4.1 Front-end ja back-end	20
4.2 Toimintaperiaatteet	20
5 WEB-PORTFOLIO	23
5.1 Ohjelmistot	23
5.1.3 Drupal	24
5.1.4 Joomla!	26
5.1.5 Wordpress	27
5.2 Valinta ja käyttöönotto	28
6 LOPPUARVIOINTI	32

LÄHTEET

1 JOHDANTO

Opinnäytetyössä tarkastellaan sisällönhallinta- ja julkaisujärjestelmien toimintaperiaatteita ja arkkitehtuuria. Tavoitteena on perehtyä eri ohjelmien tarjoamiin mahdollisuuksiin ja www-sisällönhallinnan peruseräisiin. Opinnäytetyö tulee olemaan tutkimustyö, jonka käytännönosuus toimii apuna perehtymisessä sisällönhallintaohjelmistoihin. Tämän vuoksi esimerkkityö on varsin suppea osa tätä opinnäytetyötä, se tarjoaa vain perehtymisen asiaan käytännössä. Esimerkkityö on henkilökohtaisen portfoliosivuston alusta. Portfolio tulee toimimaan jatkossa apuna tulevassa työhaussa, tällä tavoin mahdollisen työnantajan on helppoa tutustua työnhakijan osaamiseen ja graafisiin kykyihin. Se tulee sisältämään tähänastiset työnäytteet, jotka ovat nopeasti tarkasteltavissa. Tavoitteena on myös varmistaa tuki eri mediaformaateille, kuten kuva ja video.

Opinnäytetyön ensimmäisten lukujen pääpaino on sisällönhallinnan teoriassa, joka kattaa sisällönhallinta- ja julkaisujärjestelmien määritelmät, käyttötarkoitukset ja nykyisen tilanteen. Seuraavissa luvuissa perehdymme www-sisällönhallinnan määritelmiin ja teknologioihin, joihin opinnäytetyötä varten valitut sovellukset perustuvat.

Ennen esimerkkityön esittelyä, otetaan yleinen katsaus muutamiin tarjolla oleviin ohjelmistoihin. Täysin kattavaa listaa ei tässä opinnäytetyössä ole, sillä vapaassa levityksessä olevia sisällönhallintajärjestelmiä on kymmeniä ja ne ovat jatkuvan kehitystyön alaisia. Aihealueen rajoittamiseksi opinnäytetyössä perehdytään ainoastaan avoimen lähdekoodin ohjelmistoihin.

Esimerkkityön alussa valitsen muutaman avoimen lähdekoodin ohjelmiston lähempään tarkasteluun, joiden tarjoamia toimintoja testaan portfolioa varten. Seuraavaksi luon lyhyen vaatimusmäärittelyn, jolla pyrin selittämään millainen ohjelmisto sopisi parhaiten käyttötarkoituksiini. Viimeinen osuus keskittyy koko työn arviointiin. Opinnäytetyön päätyttyä, käyttötarkoitukseen parhaiten soveltuva sisällönhallintaohjelma ja sen avulla luotu sivusto siirtyy aktiiviseen käyttöön.

Lähdemateriaalin löytäminen opinnäytetyölle oli haaste sekin. Kirjallisuutta työlle ei paikallisesti löytynyt ollenkaan. Suomalaisista kirjakaupoissa materiaalia oli sentään tarjolla, vaikkakin lähinnä englanninkielisenä. Päälähteenä opinnäytetyölle toimii suosittu Bob Boikon teos Content Management Bible, joka kattaa suuren osan sisällönhallinnan teoriasta. Muut lähteet ovat internetistä poimittua kirjallisuutta ja artikkeleja, muun muassa: Bradford Lee Edenin teos Content Management Systems.

2 SISÄLLÖNHALLINTA

Aivan aluksi on hyvä käsitellä sisällönhallinnan (engl. Content Management) perusmääritelmät. Bob Boikon teos Content Management Bible on varsin kattava sisällönhallinnan määritelmässä ja tarjoaa paljon tätä lukua varten. Aihealueen laajuus vaatii, että esitän asiat lyhyesti ja pääpiirteittäin.

Sisällönhallinta sekoitetaan usein ajatukseen luoda suuria ja helposti ylläpidettäviä internetsivuja. Sisällönhallinta ei kuitenkaan rajoitu pelkästään internetsivuihin, vaikka nykypäivänä se yleisesti liitetään niihin. Tämä opinnäytetyö rajoittuukin vain käsittelemään www-sisällönhallinnan teoriaa. Sisällönhallinta tarkoittaa informaation luomisen ja jakamisen hallintaa, joka on käsitteenä laaja kokonaisuus. Ann Rockley toteaa sisällönhallinnasta: Tehokas sisällönhallinta on toistettava menetelmä, jolla kyetään yksilöimään sisältövaatimukset etukäteen. Tämä tapahtuu luomalla johdonmukaista sisältöä uusiokäyttöön, hallitsemalla tätä sisältöä tietyssä lähteessä ja kokoamalla sisältöä vastaamaan asiakkaiden tarpeita. (Boiko 2005, 65–67.)

Ensimmäiseksi on hyvä määritellä mitä tarkoitetaan sisällöllä sisällönhallinnassa. Sisältö ei ole yksinkertaisesti vain kasa erilaista mediaa, vaan sille paljon laajempi määritelmä.

2.1 Sisällönhallinnan määritelmiä

Bob Boiko määrittelee tiedon ja informaation teoksessaan seuraavasti: Tietokone maailman ihmiset ovat lähestulkoon yksimielisiä siitä, miten määritellään tieto. (engl. data) Tieto on pieniä katkelmia informaatiosta, jotka liitetään toisiinsa tietueiksi ja talletetaan tietokantoihin. Boikon mukaan informaatiolle taas löytyy monta eri käsitettä mutta hän itse mieltää informaatioksi kaikki tyypilliset talletetun median muodot. Miten tämä taas liittyy sisältöön? Raaka informaatio muuttuu sisällöksi kun sille annetaan käytännöllinen muoto, jolle on yksi tai useampi tarkoitus. Sisältö on siis informaatiota. Informaation voimme merkitä tiedolla niin, että tietokone ky-

kenee organisoimaan ja systematisoimaan sen keräämisen, hallinnoimisen ja julkaisemisen. (Boiko 2005, 7–11.)

Tieto keksittiin, koska se on paljon helpommin käsiteltävissä kuin informaatio. Se on pientä, yksinkertaista ja kaikki sen asiayhteydet tunnetaan. Tieto tekee tietokoneohjelmien luomisen mahdolliseksi. Informaatio puolestaan on laajaa, monimutkaista ja pursuaa asiayhteyksiä, jotka ovat tärkeitä sen käyttötarkoitukselle mutta tietokoneelle täysin käsittämättömiä. Boikon yhteenvedossa sisältö on kompromissi tiedon käytettävyyden ja informaation rikkauden välillä. Sisältö on informaatiota, jota ympäröi metatieto. Tämä metatieto on periaatteessa yksinkertaistettu versio informaation sisällöstä ja tarkoituksesta. (Boiko 2005, 12.)

Boiko esittää asiat kirjassaan hiukan vaikeaselkoisesti. Sain kuitenkin käsityksen, että tieto on konkreettista dataa ja se ei jätä mitään tulkinnan varaan. Toisin kuin informaatio, jonka merkitys voi muuttua käyttäjän näkökulman tai informaation käytön mukaan. On olemassa sanonta, yksi kuva kertoo enemmän kuin tuhat sanaa. Tämä sanonta kuvaa hyvin informaatiota, jolla on yleisesti paljon asiayhteyksiä. Informaation rajaaminen käyttökohteen mukaan tapahtuu merkitsemällä se metatiedoilla, jotka auttavat hahmottamaan informaation käyttöä. Tällaista digitaalista informaatiota on helpompi käsitellä ja hallinnoida.

Informaation välittäminen tietokoneelle tapahtuu muuttamalla se tietokoneen ymmärtämäksi koodikieleksi, joka tunnetaan yleisemmin formaattina. (engl. format) Tietokonemaailmassa formaatti kattaa kaksi laajaa mutta läheistä käsitettä: tallennusmuoto (tiedostomuoto) ja renderöintimuoto (näyttömuoto). Tallennusmuodolla tarkoitetaan informaatiota, joka on muutettu tietokoneen ymmärtämäksi binäärikoodiksi. Otetaan esimerkiksi painettu kuva, joka pitäisi saada digitaaliseen muotoon. Skannerin tulee muuttaa pienet väripisteet koodijanaksi joka muodostuu yksöistä (1) ja nolista (0). Tietokoneen tiedostot ovat perinteinen tapa tallentaa binäärimuodossa olevaa informaatiota, siksi tätä kutsutaan myös tiedostomuodoksi. On olemassa useita binäärikoodiin pohjautuvia tiedostomuotoja: tiedoston sisältämä tieto on tallennettu siihen binäärimuotoon millä ohjelmalla se on luotu ja mikä sitä käyttää. Jokaisella binäärimuodolla on oma tapansa esittää informaatiota, joka sille on koodattu. Useimmat kuvatiedostot sisältävät ylinnä koodin, joka määrittelee millainen tiedosto on kyseessä ja kuinka iso se on. Binäärimuoto on tärkeä

siksi, että käyttäjä joutuu usein vaihtamaan sen toiseen. Esimerkiksi, käyttäjä saattaa saada kuvatiedoston Encapsulated PostScript -formaattissa (EPS) ja joutuu web-julkaisua varten muuttamaan sen Joint Photographic Experts Group -formaattiin (JPEG). Sisällön esittäminen binäärimuodossa ei ole käyttäjäystävällinen ratkaisu, ymmärrettävistä syistä. Ratkaisuna tälle toimii renderöintimuoto, joka käsitteenä sisältää typografisia ominaisuuksia ja median eri muotoja. Sisältö täytyy saada muotoiltua lukijalle ymmärrettäväksi. Tässä tapauksessa, muotoilulla viitataan sisällön visuaaliseen esitystapaan, jossa täytyy ottaa huomioon kaksi asiaa. Ensimmäiseksi: muotoilun tulee pysyä yhtenäisenä julkaisukohtaisesti. Toiseksi: muotoilu tulee pitää erillisenä sisällöstä, jotta sisältö soveltuisi uusiokäyttöön eri tuotannoissa. Muotoilu on sisällön ensimmäinen laatukriteeri joka tulee ottaa huomioon, seuraavana on rakenne. (Boiko 2005, 13–17.)

Luvun alkupuolella poikettiin käsitteessä tietokanta. Mikä se siis on? Tietokanta on kokoelma toisiinsa liittyviä tietoalkioita, joita hallinnoidaan yhtenä yksikkönä. Määritelmä on tarkoituksellisesti laaja, koska eri ohjelmistotuottajien tarjoamalla tietokantajärjestelmillä on paljon eroavaisuuksia. Esimerkiksi, Oracle Corporation määrittelee tietokantansa kokoelmana fyysisiä tiedostoja, joita hallinnoidaan vain yhdellä tietokantaohjelmistolla. Microsoft taas määrittelee SQL Server -tietokannan kokoelmana taulukoita, joka sisältää tietoa ja muita objekteja. Tietokantaobjekti on nimetty tietorakenne, joka on tallennettu tietokantaan. Objekti voi olla joko taulukko, näkymä, tai indeksi. (Oppel 2005, 22.)

Sisällön rakenteella viitataan tapaan arkistoida informaatiota. Sisällön rakenne kattaa informaation osat ja niiden suhteet toisiinsa. Esimerkiksi: internetsivun otsikko viittaa sivun sisältöön, samoin kuin elokuvan nimi viittaa sen sisältämiin kohtauksiin. Siinä missä formaatilla viitataan esitystapaan ja muotoiluun, rakenteella viitataan hallintaan. Hyvin rakennetulta sisällönperustalta löytyy seuraavia ominaisuuksia: Sisältö jakautuu tarkasti määriteltyihin kategorioihin, eli sisältötyyppeihin. Jokaisessa sisältötyypissä sisältö jakautuu helposti käsiteltäviin komponentteihin. Komponenteissa sisältö taas jakautuu määriteltyihin elementteihin. Elementit liittyvät toisiinsa joko saman tai muiden komponenttien kautta, hahmotelmien, indeksien, ristiviittausten ja sekvenssien mukaan. Kattavien rakenteiden luominen on hyvä tapa ymmärtää ja hallita sisältöä. (Boiko 2005, 21–23.)

Määritelmässä viimeisenä on toiminnallisuus. Ohjelmistojen kohdalla voidaan puhua tietokonepohjaisista prosesseista, jotka tunnetaan toimintoina. Näiden tietokoneohjelmien tarjoamat toiminnot ovat periaatteessa prosesseja, jotka tapahtuvat käyttäjän ja tietokoneen välillä. Lyhyesti sanottuna, toiminnoilla tarkoitetaan tapaa työskennellä tietokoneen avulla, jossa käyttäjä esimerkiksi tekee kyselyn tietokoneelle tai vastaa tietokoneen antamaan kyselyyn. (Boiko 2005, 31.) Tällainen käyttäjän ja tietokoneen välinen vuorovaikutelma tunnetaan yleisemmin käyttöliittymänä, joka tarjoaa käyttäjälle mahdollisuuden vaikuttaa ohjelmistojen sisäisiin prosesseihin ja toimintoihin.

Aikaisemmin toiminnot rajoittuivat laajoihin ohjelmistoihin mutta tänä päivänä se on jakautunut kevyisiin koodinpätkiin ja sekoittunut muuhun mediaan niin vahvasti, että on jo vaikea sanoa milloin puhua informaatiosta ja milloin toiminnoista (Boiko 2005, 40.)

Jotta informaatiosta saataisiin sisältöä, sille pitää löytyä tarkoitus jonka mukaan se järjestellään kategorioihin. Informaation tarkoitus on alku organisoimiselle, jolla luodaan järjestelmä täynnä nimityksiä ja kategorioita. Sisältö on siis informaatiota, joka järjestellään jonkin tarkoituksen ympärille, johonkin käyttöön. Avaintekijänä sisällön käyttämiselle ja organisoimiselle on nimeäminen. Nimitykset tarjoavat yksinkertaisia, muistettavia ja käytännöllisiä säilöjä, joihin voi kerätä ja yhdistää muutoin vertailukelvottomia informaation palasia. Yksinkertaisuudessaan sisällönhallinnalla saadaan teokseen ohjattavuutta ja mahdollisuus jakaa informaatiota ja toimintoja. Kyseessä on tietämys siitä, mitä henkilö pystyy tarjoamaan, kuka sitä tarvitsee ja missä mittakaavassa, ja kuinka he haluavat sen toimitettavan. (Boiko 2005, 45–65.)

2.2 Sisällönhallintajärjestelmä

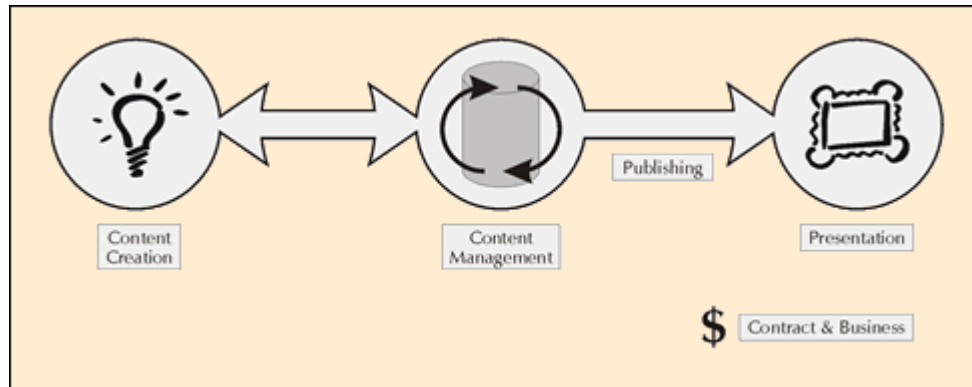
Täysin varmaa ei ole, kenestä sai alkunsa ajatus sisällönhallintajärjestelmästä. Monet sisällönhallinnan analyttikot ovat sitä mieltä, että suurin kiitos kuuluu yritykselle nimeltä Vignette, joka julkaisi Vignette StoryServer ja StoryBuilder – ohjelmistot 90 – luvun puolivälissä (Doyle 2011; Gilbane 2000). Sisällönhallintajärjestelmät syntyivät samoin kuin mikä tahansa muu tuote tai palvelu jolle löytyi tarve. Kun

internetsivut vaihtuivat pienistä, epävirallisesti suunnitelluista sivustoista massiivisiin ja nopeasti muuttuviin sivustoihin, tarve varteenotettaville hallintatyökaluille tuli kiireelliseksi. Tuotantoyhtiöt astuivat esiin tarjoamaan vastausta tälle tarpeelle ja näin sai alkunsa nimitys sisällönhallintajärjestelmä. (engl. Content Management System tai CMS) Sisällönhallintajärjestelmä auttaa pitämään sivustoja järjestelmällisenä ja korkealaatuisina, samalla kun ne jatkavat nopeaa kasvuaan. (Boiko 2005, 66.)

Seuraavaksi tarkastellaan sisällönhallintajärjestelmien kehitystä aivan www-tuotannon alkupuolelta. Alun perin sivujen luonti tapahtui käyttämällä alkeellisia tekstieditoreja HyperText Markup Language -sivunkuvauskielen (HTML) kirjoittamiseen. Ajan saatossa tuotantoyhtiöt loivat helposti käytettäviä työkaluja ihmisten avuksi, joilla ei ollut paljon teknistä ymmärrystä. Parhaimmillaan nämä työkalut automatisoivat vain erinäisiä tehtäviä, kuten vaikkapa auttamalla muistamaan HTML -kieltä. Myöhemmin nämä työkalut kehittyivät What You See Is What You Get (WYSIWYG eli mitä näet, sitä saat) -hallintaympäristöiksi, joilla kyettiin luomaan kokonaisia internetsivuja. Nykyään jotkin näistä työkaluista sisältävät juuri riittävästi hallintaominaisuuksia kyetäkseen toimimaan nimellisinä web-sisällönhallintaohjelmistoina organisaatioille, joiden sivut ovat pienet eivätkä sisällä ylimääräisiä julkaisuja. Ne kuitenkin sisältävät sisällönhallintajärjestelmän perusperiaatteet: kokoaminen, hallinta ja julkaisu. Joten ymmärtämällä kuinka nämä työkalut toimivat, on jo iso askel ymmärtämään monimutkaisempien järjestelmien toimintaa. (Boiko 2005, 72.)

Prosessina www-sisällönhallinnan voi mieltää kolmeen osa-alueeseen: Kokoaminen, hallinta ja julkaisu. Kokoaminen tapahtuu joko luomalla informaatiota tai hakemalla sitä valmiista lähteestä. Verkkotyökalut tarjoavat tavallisesti rakenteellisen julkaisu ympäristön, joka sisältää perusjulkaisumallipohjat, tyylit, opastuksen ja ominaisuuksia, jotka auttavat luomaan alkeellisen web-sivuston nopeasti, vaikka ymmärrys HTML:stä olisikin vähäistä. Hallinnan osalta verkkotyökalut taas tarjoavat mahdollisuuden luoda säilytyspaikan, joka sisältää tietokanta-arkiston ja tiedostoja, jotka sisältävät sisällön osia ja hallinnointitietoa. Sisällön julkaisemisella tarkoitetaan sen tekemistä käytettäväksi muille. Tämä tapahtuu ottamalla tallella olevia sisällön osia ja rakentamalla kohdistettuja julkaisuja, kuten esimerkiksi internetsivuja, tulostettavia dokumentteja ja sähköisiä uutiskirjeitä. Julkaisut sisältä-

vät asianmukaisesti koottuja sisällön osia, hyvän toiminnallisuuden, liitännäistietoja ja navigointimahdollisuudet. (Boiko 2005, 72.)



KUVIO 1. Esitys sisällönhallinnan prosessista (steptwo 2003).

James Robertson tuo mukaan esittelyn yhtenä osa-alueena www-sisällönhallinnan prosessissa. (KUVIO 1) Hänen mukaansa sisällönhallintajärjestelmät tarjoavat useita ominaisuuksia, jotka parantavat sivuston laatua ja tehokkuutta. Ne esimerkiksi rakentavat sivuston navigoinnin lukemalla rakenteen suoraan sisällön arkistosta. Ne myös mahdollistavat useiden eri selainten tuen. Sisällönhallintajärjestelmällä voidaan tehdä sivustoista dynaamisia ja interaktiivisia, mikä parantaa sivuston vaikutusta. Julkaistujen sivujen on täytettävä tietyt vaatimukset, jotta niillä olisi arvoa käyttäjille. On tärkeää täsmentää nämä vaatimukset, jos pyydät palveluntarjoajaa suunnittelemaan verkkosivujen ulkoasun ja asettelun. Vaatimuksiksi sivustoille hän asettaa seuraavat: Käytettävyys, saavutettavuus, selainten tuki, rajoitetut toiminnot asiakkaille, nopeus, kelvollinen HTML, tehokas navigointi ja metadata. (Robertson 2003.)

Robertsonin artikkeli havainnollistaa sisällönhallinnan prosessit esimerkkien avulla, jotka auttavat kyllä ymmärtämään prosessien määritelmiä ja hän pysyy paljolti samalla linjalla Boikon määritelmien kanssa. Omasta mielestäni hänen määritelmänsä esittelystä kuuluu kuitenkin samaan yhteyteen julkaisun kanssa. Julkaisulahan tarkoitetaan sisällön saattamista muiden käyttöön ja Robertson mielestäni vain tarkentaa miten tämä tapahtuu käytännössä.

3 WWW-SISÄLLÖNHALLINNAN LÄHTÖKOHDAT

Verkkosivujen luominen käyttämällä ainoastaan HTML- ja Cascading Style Sheets -kieliä (CSS) on yleensä sekoitus logiikkaa, esittelyä ja sisältöä saman koodin sisällä. Tämä on kuitenkin aikaa vievää ja joustamatonta. Esimerkiksi lisäämällä uusia navigointimahdollisuuksia tai muuttamalla tekijänoikeustietoja alatunnisteseen, muutokset täytyy synkronoida jokaisessa HTML -tiedostossa. Asiakkaat eivät välttämättä kykene hallitsemaan verkkosivujensa sisältöä itse, koska heillä tulisi olla tuntemusta HTML -ohjelmointikielestä. Sisällönhallintajärjestelmä toimii ratkaisuna yllä mainittuihin ongelmiin, mutta se ei ole verkkosivujen rakennusohjelma. Rakennusohjelmia käyttävät ihmiset, jotka haluavat rakentaa verkkosivustoja oppimatta sivuntuotannon teknisiä seikkoja. He käyttävät valmiita suunnittelumalleja ja valitsevat rajoitettuja toimintoja, mitä ohjelma sattuu tarjoamaan. Tällainen sivustontuotanto on joustamatonta ja sitä käytetään usein yksityisten sivujen rakentamiseen. Sisällönhallintajärjestelmät palvelevat parhaiten ammattilaisverkko-vastaavia, jotka luovat asiakkailleen yksilöllisiä ulkoasuja ja tarvittavia toimintoja. (Hauschildt 2010, 7-9.)

Sisällönhallintajärjestelmä tarjoaa tavan hallita suuria määriä verkossa olevaa tietoa ilman taakkaa koodata kaikki tieto kullekin sivulle HTML:llä käsin. Sisällönhallintajärjestelmät ovat tärkeitä esimerkiksi kirjastoille, koska pelkästään julkisen web-sisällön määrä on saavuttanut pisteen, jossa ylläpito on ongelma. Jos esimerkiksi verkkosivuilla on useampi tekijä, ongelmaksi saattaa muodostua tekijöiden erimielisyydet vaikkapa sivun ulkoasusta. Sisällönhallintajärjestelmä tarjoaa keskitetyn hallintajärjestelmän näytettävälle sisällölle. Sisällönhallintajärjestelmän haittoja on muutamia, joita tulee harkita tarkoin. Osa sisällönhallintajärjestelmistä pakottaa käyttäjän pitämään tiettyjä malliteemoja, jolloin on mahdollista muuttaa vain ulkoasun peruselementtejä ja väri vaihtoehtoja. (Eden 2006, 5; Battles 2008.)

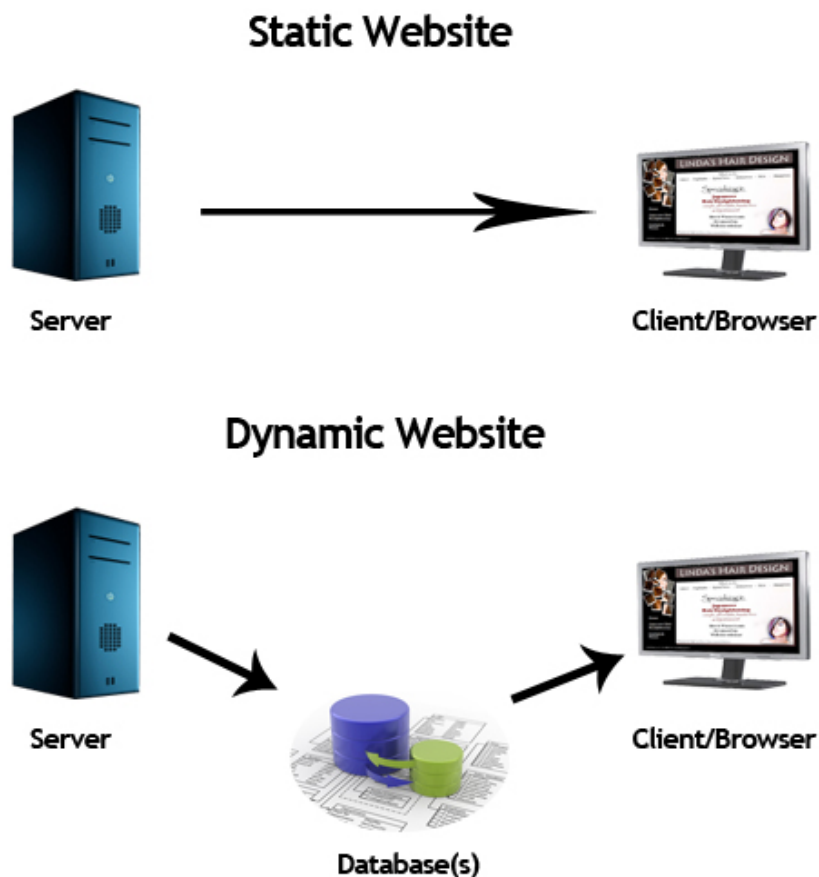
James Robertson tuo www-sisällönhallinnan hyötyjä esille enemmän liiketoiminnan näkökulmasta mutta ne kannattaa ottaa huomioon joka tapauksessa. Robertson toteaa, että kun verkkosivut ovat saatu asennettua ja asiakkaat alkavat vyöryä sisään, ylläpitäjä haluaa tehdä muutoksia sivulle lennosta. Nämä muutokset voivat

olla hyvinkin pieniä, kuten vaikkapa yhteystietojen vaihto. Uudet tuotteet tai liiketoiminnan suunnan vaihto, tarkoittaa suurempien muutoksien tekemistä sivuille. Todennäköisintä on, että ylläpitäjä on tyytyväinen nykyiseen muotoiluun ja tarvittava muutos on vain pieni pätkä tekstiä. Vaikka sivujen suunnittelijat olisivatkin tehneet hyvää työnjälkeä, ei ylläpitäjä halua turvautua heihin kaikissa pienissä muutoksissa. Sadat pienet muutokset vuodessa ovat varmasti turhauttava ja taloudellisesti kannattamaton vaihtoehto molemmille osapuolille. Tässä kohtaa sisällönhallintajärjestelmä tulee mukaan. Se tarjoaa käyttäjälleen yksinkertaisen ja vähemmän teknisen keinon päivittää sivujen sisältöä. Tämä toteutuu usein, mutta ei aina, web-pohjaisella käyttöliittymällä. Tällä tavoin ylläpitäjä ei enää ole täysin riippuvainen web-suunnittelijoista sivuja koskevissa muutoksissa. Hallintatyökalut mahdollistavat muun muassa kyvyn tarkkailla sivuston ylläpitäjien tekemisiä ja näin välttää mahdollisia sekaannuksia. Näillä työkaluilla voi varmistaa, että jokainen sivuja päivittävä työntekijä voi muokata vain niitä sivuston osia, joista he ovat vastuullisia. Muiden hallintatyökalujen ohella tämä vapauttaa käyttäjän keskittymään vain sanoihiin, ei tekniikkaan. (Robertson 2003.)

3.1 Staattiset ja dynaamiset sivut

Sisällönhallintajärjestelmiin pohjautuvat dynaamiset sivustot on helppo tunnistaa. Dynaamisen sivuston verkko-osoitteen pääte on esimerkiksi: .cgi, .php tai .asp, jota seuraa kysymysmerkki ja muut parametrit, joiden avulla järjestelmä rakentaa sivua. Tästä voi päätellä, että sivusto käyttää jonkinlaista sisällönhallintajärjestelmää. Web-palvelin on suunnitellut nämä dynaamiset sivustot lennosta käyttämällä tietokantaan talletettua sisältöä. Vastakohta dynaamiselle sivustolle on staattinen sivusto. Staattiset sivustot eivät ole sivustoja jotka eivät koskaan muutu, vain sen sivuston sivut ovat staattisia. Niitä ei voi muuttaa, ellei joku tallenna päivitettyä HTML-tiedostoa palvelimelle. Dynaamisilla sivustoilla on omat puolensa ja haittansa. Dynaaminen sivusto voivat kasvaa paljon helpommin ja ovat paljon tehokkaammin hallittavissa pienellä joukolla ihmisiä, kuin mihin pieni staattinen sivusto kykenee. Toisaalta, dynaamiset sivustot usein päätyvät käyttämään hyväksi havaittua, valmista ulkoasutyylä. Sisällönhallintajärjestelmät voivat olla myös kalliita ja muutokset voivat edellyttää ohjelmoijan asiantuntemusta. (Addison 2004, 223)

Staattiset sivustot ovat juuri sitä - staattisia. Ne toimitetaan selaimelle suoraan palvelimelta johon ne ovat talletettu ja aivan kuten toimitettu postipaketti, mitä näet, sitä saat. Syy miksi staattisilla sivustoilla ei ole muutoksia on, että kaksi keskeistä sivuston osaa, informaatio ja muotoilu, ovat molemmat kiinteästi kytketty osaksi sivuston sivuja ja sitten tallennettu palvelimelle. Siellä ne vain odottavat kunnes jokin selain lähettää niille kutsun. (KUVIO 2) Staattisilla sivustoilla on hyviä puolia. Niitä on nopeaa ja halpaa kehittää, ja myös niiden ylläpito on edullista, koska ne eivät vaadi paljon tallennustilaa. Vähäisen tallennuskapasiteetin ansiosta sivustot ovat myös nopeita ladata, koska palvelimilta ei mene kauaa niiden suorittamiseen. Mutta haittojakin on. Yksinkertaisetkin päivitykset ja muutokset edellyttävät asiantuntemusta ja panostusta. Sivut voivat nopeasti seisahtua vähäisten tai jopa ole-mattomien sisällönmuutosten seurauksena. Tarjotut toiminnot, sekä asiakkaan ja että ylläpitäjän, ovat hyvin rajalliset. Staattisilla sivustoilla ei myöskään ole käyttä-jäkohtaisia näkymiä, ne eivät joustu, eivätkä skaalaudu. (sitedynamics 2009; Boiko 2005, 75.)



KUVIO 2. Mallikuva staattisesta ja dynaamisesta sivustosta (sitedynamics 2009).

Dynaamiset sivustot tarjoavat vierailijalle mahdollisesti erilaisen kokemuksen joka kerta kun he vierailevat sivustolla. Sisältöä voidaan helposti muuttaa valvojan tai sivuston omistajan toimesta. Sivuston muotoiluelementit ja informaatio ovat palvelimella erillään. Varsinaiset sivut ovat kuin tyhjät kuoret, joille piirretään sisältöä palvelimen ja selaimen välillä toimivasta tietokannasta. Tämän rakenteen ihanteellisuus tulee siitä, että sivuston omistaja hallinnoi tietokannan informaatiota, ilman teknisiä kompastuskiviä. Dynaamisen sivuston yksi positiivisimmista puolista on päivitysten ja muutosten helppo toteuttaminen omistajan tai ylläpitäjän toimesta. Sisältö on aina olennaista ja tuoretta tarjoten käyttäjille syyn palata. Dynaamisen sivuston tarjoamien toimintojen mahdollisuudet eivät ole yhtä rajoitettuja kuin staattisen. Tarjoaa omistajalle reaaliaikaisen, suoran yhteyden selaimiin ja mahdollisuuden asiakaspalautteeseen. Dynaamisen sivuston haittoja ovat muun muassa: Ne tarvitsevat enemmän asiantuntemusta ja aikaa kehittyä. Dynaamisen sivuston ylläpito voi maksaa hieman enemmän kuin staattisen sivuston. Kustannukset riippuvat sivuston vaatimuksista.(sitedynamics 2009.)

Kuten luvun alusta voi päätellä, dynaamiset sivustot tunnetaan usein myös tietokantaohjattuina sivustoina, joka luo sivuston lennosta käyttäjän lähettäessä pyynnön. Palvelin vastaanottaa kyselyn käyttäjän painaessa klikkaamalla linkkiä. Linkki aktivoi tietokannasta mallisivun, joka sisältää tavanomaista HTML-kieltä, kuin myös ohjelmointi skriptiä, objekteja ja muita ohjelmia jotka tulkkavat pyynnön, ottavat yhteyden tietokantaan ja noutavat asianmukaisen sisällön. Boikon mukaan dynaamiset internetsivut ja sisällönhallintajärjestelmät eivät aina kulje yksissä kansissa, vaan olemassa puhtaasti itsenäisiä ja dynaamisia sivustoja ilman sisällönhallintajärjestelmää. (Boiko 2005, 75–76.)

3.2 Web-kehitysteknologiat

Tässä luvussa käydään läpi ne teknologiat, joille opinnäytetyöhön valitut web-sisällönhallintajärjestelmät pohjautuvat. Tavoitteena on käydä läpi vain ohjelmointikieliset ja tekniikat, jotka ovat oleellimmat tälle opinnäytetyölle. Web-sisällönhallintajärjestelmille asetin rajapinnaksi Hypertext Preprocessor – kielen

(PHP), vaikka kyseisiä järjestelmiä luodaan useilla eri ohjelmointikielillä, kuten vaikkapa Javalla.

3.2.1 HTML -sivunkuvauskieli

Vuonna 1992, Tim Berners-Lee, brittiläinen ohjelmistosuunnittelija ja tietotekniikan asiantuntija, loi HTML:n asiakirjojen tekemistä varten. Hän suunnitteli myös HyperText Transfer Protocol -protokollan (HTTP), jolla nämä asiakirjat kyettiin lähettämään, ja ensimmäisen selaineditorin, nimeltään World Wide Web (WWW). WWW tuli pian National Center for Supercomputing Applicationsin (NCSA) tietoon, jossa ohjelmointitiimi päätti luoda paremman selaimen. Näin syntyi Mosaic, ensimmäinen selain, joka tukee monta eri multimedialla. Mosaic johti lopulta nykypäiväisten selainten luontiin. (Schafer 2010, 39.)

HTML-sivut kyetään näyttämään lähes millä tahansa tietokoneella, jolla on nykyaikainen käyttöjärjestelmä joka sisältää selaimen. HTML-tiedosto sisältää vain tekstitunnisteita (eli muotoilukoodeja), joten HTML-tiedoston luonti onnistuu nopeasti ja helposti käyttämällä ainoastaan tekstieditoria ja selainta. (Hart-Davis 2004, 1.)

Kuten luvun nimestä voi jo päätellä, kyseessä ei ole ohjelmointikieli. HTML on sivunkuvauskieli, joka koostuu laajasta joukosta tunnisteita, jotka puolestaan määrittelevät verkkosivujen rakenteen. Kaikki nykyaikaiset käyttöjärjestelmät sisältävät selaimia, joten HTML-sivuja voidaan näyttää lähes millä tahansa tietokoneella. HTML-tiedosto sisältää <> -merkein ympäröityä tekstiä, jota kutsutaan muotoilukoodiksi. Tästä syystä, HTML-tiedoston luominen onnistuu nopeasti ja helposti käyttämällä vain tekstieditoria ja selainta. Tunniste on kulmasulkeiden ympäröimä nimi, joita tulee kaksi kappaletta. Ensimmäinen tunniste on avaava <esimerkki> ja seuraava on sulkeva tunniste, joka määritellään heittomerkillä </esimerkki>. Tunnisteiden väliin tulee sivun elementti, joka voi olla tekstiä tai grafiikkaa. Tunnisteilla voi myös olla ominaisuuksia, jotka entisestään määrittelevät niiden muotoilua tai toimintoja. Tunnisteiden <html> ja </html> välillä oleva teksti määrittelee sivun. Tämä myös kertoo selaimelle, mistä kohtaa HTML -kieli alkaa ja mihin se päättyy. (Hart-Davis 2004, 3; w3schools 2011.)

HTML 5 tulee olemaan seuraava suuri versio HTML:stä. Se on suunniteltu korvaamaan HTML 4.01 ja XHTML. On realistista olettaa, että HTML 5:n valmistuminen kestää vielä kauan, ehkä jopa vuosia. Jotkut selaimet tukevat jo muutamia HTML 5:n ominaisuuksia, kuten uutta selain-tietokantaa ja välimuistin mekanismeja. Se tarjoaa myös ohjelmointirajapintoja (engl. Application Programming Interface, API), jotka antavat uusia elementtejä käytettäväksi ja joilla helpotetaan "vedä ja pudota" -toimintoja. Uudet elementit tarjoavat keinoja liittää tietyntyyppistä sisältöä, kuten audio- ja videoelementtejä, ilman turvautumista liitännäissovelluksiin. Audioelementin avulla, sivulle voidaan upottaa vaikka MP3 -tiedostoja. Videoelementti on yhä kiistelty aihe, koska selain valmistajat kamppailevat keskenään selainten yhteisestä formaatista. (Duckett 2010, 637–639.)

Duckettin tieto on hiukan vanhentunutta. Nykypäivänä merkittävimmät selaimet antavat jo kattavan tuen HTML 5 -ominaisuuksille. HTML 5 on joka tapauksessa alkanut hitaasti valtaamaan alaa vanhemman version tieltä. Videoelementin formaatista on kuitenkin erimielisyyksiä valmistajien keskuudessa vielä tänäkin päivänä.

3.2.2 CSS -kieli

Internet sai alkunsa HTML:stä ja tekstimuotoisista -asiakirjoista. Vuosien saatossa, internetistä on tullut kotitalouksien ja teollisuuden tukipilari, josta lopulta valmistui kannattavia julkaisualustoja CSS:n ansiosta. CSS mahdollistaa web-suunnittelijat ja -ohjelmoijat virittämään elementtejä julkaisuja varten, sekä verkkoon että useille eri medioille. Tyylitietojen etu on, että käyttäjän tarvitsee muuttaa määrittelyjä vain kerran ja tämä muutos vaikuttaa jokaiseen elementtiin, joka käyttää kyseistä tyylitiedostoa. Näin käyttäjän ei tarvitse koodata jokaista elementtiä erikseen. Tyylitiedot tarjoavat helpon tavan päivittää asiakirjan muotoilua ja ylläpitää useiden asiakirjojen yhdenmukaisuutta. (Schafer 2010, 397–398.)

CSS on Worldwide Web Consortiumin (W3C) määrittelemä www-kuvauskielen standardi, jolla verkkosivut määritellään noudattamaan samaa tyyliä ja sovelta- maan tätä mallia useille sivuille. CSS on tekniikka, joka suurimmaksi osaksi on kirjoitettava manuaalisesti, jos sillä aikoo luoda mielenkiintoa herättäviä töitä. On-

gelmana on, että useiden suunnittelijoiden on vaikea sitoutua kirjoittamaan koodia saadakseen työnsä tehtyä. (Schmitt 2008, 13.)

CSS on kieli, jolla määritellään verkkosivujen esitystapa. Tämä sisältää määrittelyt väreille, sivun asettelulle ja fonteille. Esitystavan voi mukauttaa erilaisille laitteille, kuten suurille tai pienille näytöille, tai tulostimille. CSS on riippumaton HTML:stä ja sitä voidaan soveltaa mille tahansa eXtensible Markup Language –pohjaiselle (XML) kuvauskielelle. Erottaminen HTML:stä helpottaa ylläpitämään sivustoja, jakamaan tyylitietoja sivujen välillä ja räätälöimään sivut erilaisiin ympäristöihin. (W3C 2011.)

3.2.3 PHP -ohjelmointikieli

Nykypäivän PHP on oikeastaan jatkotuote Personal Home Page / Forms Interpreterille (PHP/FI). Vuonna 1994, Rasmus Lerdorf laati ensimmäinen inkarnaation PHP:stä. Tämä koostui joukosta yksinkertaisia Common Gateway Interface -binäärejä (CGI), jotka oli kirjoitettu C -ohjelmointikielellä. Alun perin Lerdorf käytti tätä vierailijoiden seurantaan verkossa ja hän nimitti näitä skriptipaketteja henkilökohtaisiksi kotisivutyökaluiksi (engl. Personal Home Page Tools), joihin usein viitattiin nimellä PHP Tools. Ajan myötä toimintoja alettiin vaatia lisää ja Lerdorf uudisti PHP Toolsin tuottaen paljon suuremman ja rikkaamman version. Tämä uusi malli kykeni muun muassa vuorovaikuttamaan tietokantoihin. Se tarjosi myös rakenteet, joilla käyttäjät kykenivät kehittämään yksinkertaisia, dynaamisia web-sovelluksia, kuten vieraskirjoja. Kesäkuussa 1995, Lerdorf julkaisi PHP Tools lähdekoodin, joka salli kehittäjien käyttää sitä mielensä mukaan. Tämä kannusti käyttäjiä myös korjaamaan virheitä koodissa ja parantamaan sen laatua. Viimeisin versio, PHP 5, julkaistiin heinäkuussa 2004, pitkän kehitystyön ja useiden esijulkaisujen jälkeen. Sen taustalla toimii Zend Engine 2.0 -ydin, joka tarjoaa uuden objektimallin ja kymmeniä uusia ominaisuuksia. PHP:n nykyinen kehitystiimi sisältää kymmeniä kehittäjiä, samalla kun kymmenet muut työstävät PHP -liitännäisiä ja sitä tukevia hankkeita. Vaikka arvio perustuu tilastoihin aikaisemmilta vuosilta, on turvallista olettaa, että PHP:tä käyttää kymmenet tai ehkä jopa sadat miljoonat verkkotunnukset ympäri maailmaa. (PHP 2011.)

3.2.4 SQL

Structured Query Language (SQL) on tietokantakieli. Kaikki nykyaikaiset relaatio-tietokannat, kuten Access, FileMaker Pro, Microsoft SQL Server ja Oracle käyttävät SQL:ää perustana ohjelmistoilleen. Itse asiassa, nämä ohjelmat ovat usein ainoa tapa, joilla käyttäjä voi vaikuttaa itse tietokantaan. Kaikki graafiset käyttöliittymät, jotka tarjoavat tiedonsyöttö ja -manipulointi toimintoja, eivät ole mitään muuta kuin SQL -kääntäjiä. Ohjelma muuntaa käyttäjän tekemät graafiset toimet, tietokannan ymmärtämiksi SQL -komennoiksi. SQL on yksinkertainen kieli. Sillä on rajallinen määrä komentoja ja nämä komennot ovat helposti luettavissa englanniksi. (Chapple 2011.)

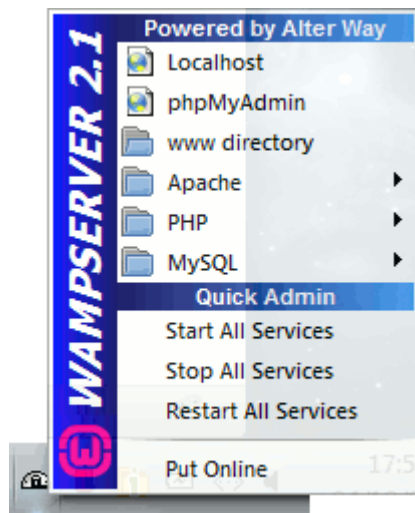
SQL-komennot voidaan jakaa kahteen alakieleen. Data Definition Language (DDL) sisältää komentoja, joilla voidaan luoda ja tuhota tietokantoja, ja tietokannan objekteja. Kun tietokannan rakenne on määritelty Data Definition Languageella (DDL), ylläpitäjät ja käyttäjät voivat hyödyntää Data Manipulation Languagea (DML) ja sen avulla lisätä, hakea tai muokata tietoja. (Chapple 2011.)

Menettelyllisiin ohjelmointikieliin verrattuna, SQL on varsin erilainen. Menettelyperiaatteisiin perustuvia ohjelmia ovat C++, Visual Basic, Pascal, ja muut kolmannen sukupolven ohjelmointikielet, joiden avulla ohjelmoija kirjoittaa tietokoneelle täsmälliset ohjeet siitä, mitä tehdä, jotta määritelty tavoite toteutuu. SQL on kyselykieli. Tämä tarkoittaa sitä, että käyttäjä saa haluamiaan tuloksia, vain kertomalla mitä haluaa. Ohjelma suorittaa pyynnön käyttämällä sille ohjelmoituja toimintoja ja palaa sitten tuloksien kanssa. (Wilton & Colby 2005, 12.)

3.2.5 WampServer

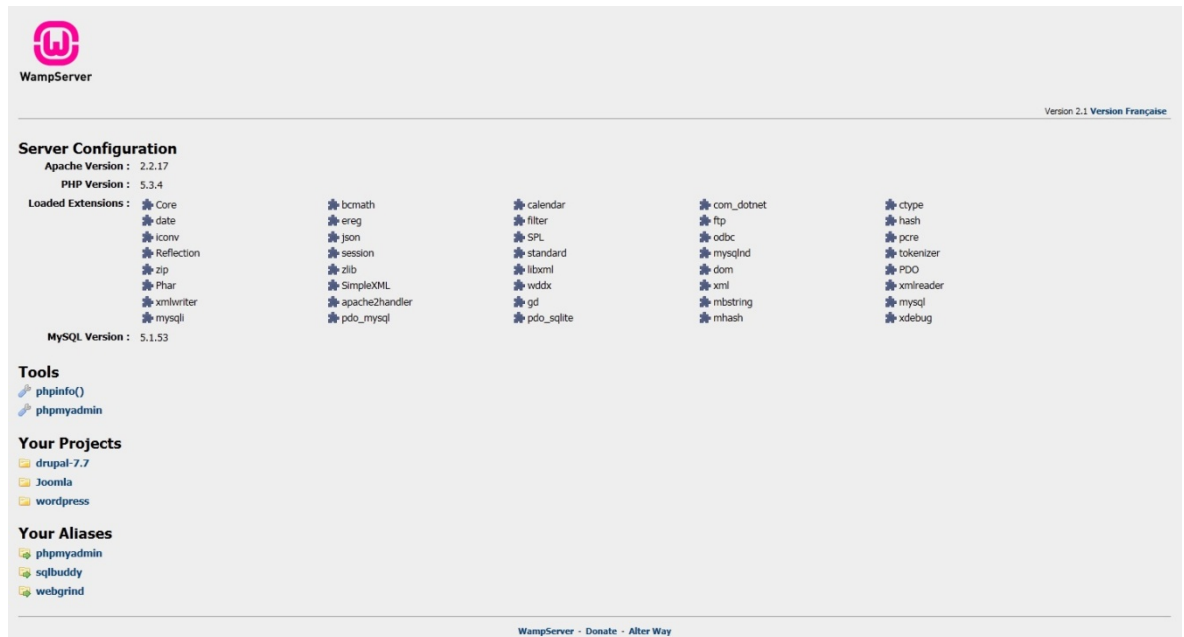
WampServer on web-kehitysympäristö Windows-käyttöjärjestelmälle, eli ei siis sisällönhallintajärjestelmä, vaan työssä apuna käytettävä sovellus. Se mahdollistaa web-sovelluksien luonnin Apachen, PHP:n ja MySQL -tietokannan avulla. Se sisältää myös PHPMyAdminin, jolla voi helposti hallita tietokantoja. WampServer asentuu automaattisesti asennusohjelman avulla ja sen käyttö on hyvin intuitiivis-

tä. Sen voi asentaa palvelimelle ilman, että käyttäjän tarvitsee edes koskea asetustiedostoihin. WampServer on pakettiratkaisu, jonka avulla voi luoda oman tuotantopalvelimen. Kun WampServer on asennettu, siihen on mahdollisuus lisätä uusia Apache, MySQL ja PHP - julkaisuja. Ohjelman mukana työpöydälle tulee myös pikakuvake, jolla voi hallita palvelinta ja sen asetuksia. (WampServer 2011.)



KUVIO 3. WampServer pikakuvake. (Wampserver 2011).

Opinnäytetyössä WampServer toimii virtuaalisena palvelimena portfoliosivustolle. Ohjelma asentuu moitteettomasti tietokoneelle, eikä vaadi erillisiä säätöjä ensialkuun. WampServer – hallintatyökalu avautuu internetselaimella, osoitteessa localhost ja käyttää tietokonetta näin palvelimena. WampServer on automaattisesti yhteydettömässä työskentelytilassa mutta sen voi tarvittaessa nopeasti asettaa online -tilaan napsauttamalla pikakuvakkeesta. Hallintanäkymässä on kaikki virtuaalipalvelimelle asennetut julkaisut, projektit ja työkalut. WampServerillä pääsee myös käsiksi PHP- ja MySQL -asetustiedostoihin, jolla voidaan muun muassa vaikuttaa mahdollisten virheilmoitusten näkyvyyteen, pakettien lähetys- ja vastaanottorajoi- tuksiin, muistin käyttöön ja tiedostokokoon.



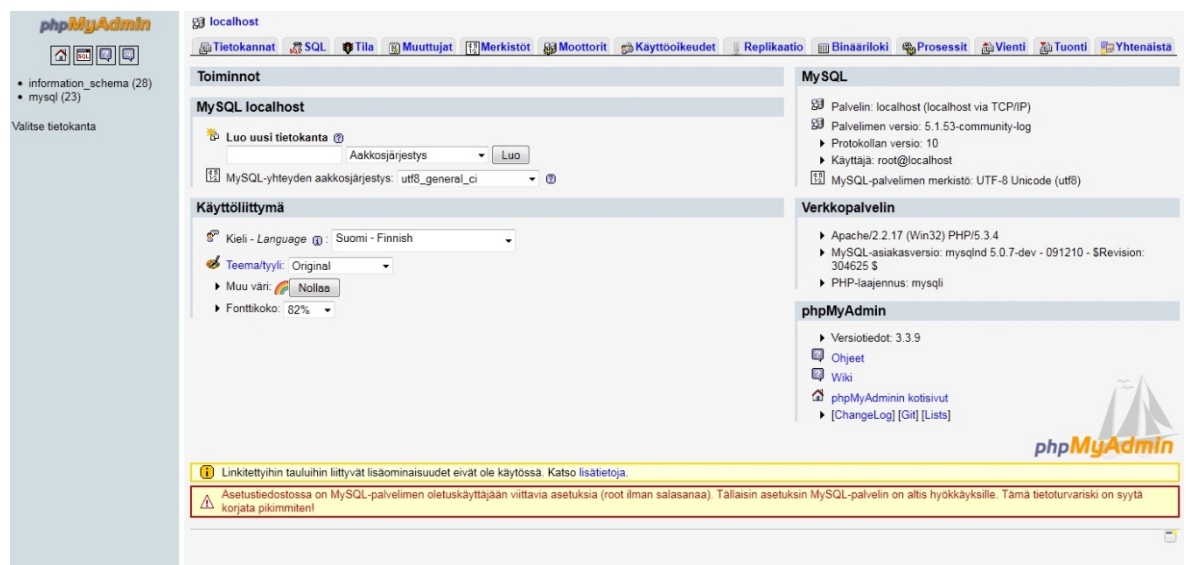
KUVIO 4. WampServer hallintanäkymä.

3.2.6 phpMyAdmin

Tobias Ratschiller, IT-konsultti ja myöhemmin Maguma ohjelmistoyhtiön perustaja, alkoi työstää PHP -pohjaista web-käyttöliittymää MySQL:lle vuonna 1998, Peter Kuppelwieserin MySQL-WebAdmin -teoksen innoittamana. Hän kuitenkin luopui hankkeesta vuonna 2000 ajanpuutteen vuoksi. Siihen mennessä phpMyAdminista oli jo muodostunut yhdeksi suosituimmista PHP -sovelluksista ja MySQL hallintatyökaluista, joilla molemmilla oli suuri käyttäjäkunta ja paljon avustajia. Jotta kasvavaa korjausten määrää voitaisiin koordinoida, Olivier Müller, Marc Delisle ja Loïc Chapeaux rekisteröivät phpMyAdmin -projektin SourceForge.nettiin ja ottivat sovelluksen kehittämisen haltuunsa vuonna 2001. (phpmyadmin 2011.)

phpMyAdmin on web-sovellus, joka on kirjoitettu PHP -kielellä ja sisältää XHTML, CSS ja JavaScript -koodia. Sovellus tukee useita toimenpiteitä MySQL:lle. Ja useimmin käytettyjä toimintoja on mahdollista suorittaa käyttöliittymän avulla, kuten hallintatietokannat, taulukot, kentät, suhteet, indeksit, käyttäjät, käyttöoikeudet ja niin edelleen. Käyttäjällä on mahdollisuus suorittaa myös suoria SQL -komentoja. Tarjoamalla tehokkaan ja graafisen käyttöliittymän MySQL:n hallintaan

phpMyAdmin on yksi suosituimmista avoimen lähdekoodin sovelluksista. Vaikka useimmat kehittäjät käyttävät päivittäin phpMyAdminin perusominaisuuksia, harvat ovat tietoisia siitä mahdollisuuksien määrästä, mitä sen edistyneimmät ominaisuudet tarjoavat. Miljoonien kehittäjien käyttämä MySQL on suosituin avoimen lähdekoodin tietokanta. Se tukee lukuisia suuria ja dynaamisia sivustoja, ja sovelluksia. MySQL on hankkinut tämän suuren suosion sen avoimen lähdekoodin, suorituskyvyn, luotettavuuden ja kestävyuden ansiosta. Sillä on myös tuki useille eri alustoille. phpMyAdmin on osaltaan auttanut tuon suosion saavuttamisessa. Se on alan standardihallintatyökalu, jonka avulla tietokannan hallinta on helppoa, niin kokeneille kuin aloitteleville kehittäjille. Web-käyttöliittymän tulo MySQL:ään on tehnyt phpMyAdminista välttämättömän välineen MySQL ja web -kehittäjille, jotka voivat olla ammattilaisia tai vasta harrastelijoita. (Delisle 2009, 1-8; phpmyadmin 2011.)



KUVIO 5. phpMyAdmin – hallintatyökalu.

4 SISÄLLÖNHALLINTAJÄRJESTELMÄN TOIMINTA

Kuinka siis kehitysteknologiat liittyvät toisiinsa sisällönhallintajärjestelmässä? Tässä luvussa käydään läpi kuinka tässä opinnäytetyössä käsiteltävät www-sisällönhallintajärjestelmät hyödyntävät edellisessä luvussa esiteltyjä teknologioita. Jaetaan teknologioiden toiminta kahteen back-endiin ja front-endiin. Selvitetään kuinka käyttäjä voi vaikuttaa sovellusten toimintaan. Luvussa käsitellään myös itse sisällönhallintajärjestelmien toimintaa yleisellä tasolla, selvittämällä mitä yhteisiä toimintaperiaatteita nämä järjestelmät jakavat keskenään.

4.1 Front-end ja back-end

Back-endissä toimii siis sisällönhallintajärjestelmän taustalla toimivat teknologiat, kuten esimerkiksi palvelimella toimiva tietokanta ja sen hallintaympäristö, joka tässä tapauksessa on MySQL. Myös PHP- ja HTML -koodit voidaan lukea tähän samaan kategoriaan, sillä järjestelmä rakentaa muun muassa sivuston perusrakenteen ja navigointimahdollisuudet valmiiksi ilman käyttäjän apua.

Käyttäjän hallintanäkymä on puolestaan järjestelmän front-end. Front-end tarkoittaa käyttöliittymää ja sen tarjoamia toimintoja joilla käyttäjä voi vaikuttaa taustalla toimiviin sovelluksiin. Jos käyttäjä esimerkiksi lisää uutta sisältöä sivustolle, järjestelmä tallettaa sisällön tietokantaan, josta se on aina noudettavissa. Käyttäjän on mahdollista vaikuttaa myös sivuston ohjelmointiin suoraan hallintanäkymästä, ilman tiedostonhakua palvelimen kansioista.

4.2 Toimintaperiaatteet

Sisällönhallintajärjestelmät löytyvät suoraan kehittäjiensä kotisivuilta tai suomenkielisiltä tukisivustoilta. Ladatut järjestelmät tulevat pakattuina tiedostoina ja näiden tiedostojen sisältö puretaan, tässä tapauksessa WampServerin www -aliansioon,

joka pitää palvelinpohjan virkaa. Seuraava vaihe asennuksessa on käynnistää Wampserver selaimessa, tässä tapauksessa Mozilla Firefox. Kirjoittamalla osoitekenttään localhost, pääsee käsiksi hallintatyökaluun. (KUVIO 4) Tässä kohtaa voi nähdä myös sisällönhallintajärjestelmät, mikäli ne on tallennettu palvelimen kansioihin. Ennen sovellusten asentamista, pitää niille luoda tietokannat phpMyAdminilla, jolla määritellään myös tietokantojen käyttöoikeudet ja tunnukset. Tietokantojen tunnukset eivät ole sama asia kuin sisällönhallintajärjestelmän tunnukset. Tietokantatunnukset toimivat vain viitteinä jotka mahdollistavat järjestelmän pääsyn tietokantaan.

Kun sisällönhallintajärjestelmät on purettu palvelimen kansioihin ja niiden asennus tapahtuu käynnistämällä ne selaimessa. Asennusvaiheessa järjestelmät pyytävät tietokantojen viitetiedot ja luovat järjestelmän pääkäyttäjän tunnukset. Muut asennuksen aikana tapahtuvat toimenpiteet riippuvat täysin järjestelmästä, näitä voivat olla muun muassa kielivalinta ja yhteensopivuuden tarkistus muiden sovellusten kanssa. Joomlailla kielen voi valita asennuksen aikana, kun taas muiden järjestelmien suomenkieliset versiot ovat ladattavista järjestelmien suomalaisilta kehittäjä sivustoilta.

Pääkäyttäjällä on mahdollisuus hallinnoida koko sivuston ulkoasua ja sisältöä, ja muiden käyttäjäoikeuksia. Oikeuksien määrittely auttaa sivuston ylläpidossa, käyttäjä voi antaa toiselle henkilölle ylläpitäjän käyttöoikeudet, joilla käyttäjä voi valvoa ja tarpeen mukaan poistaa kommentteja. Muita oikeuksia voi olla muun muassa tiedoston julkaisuoikeudet sivuston yleisnäkömään. Yleisnäkömän hallintatyökalut vaihtelevat sen mukaan mitä oikeuksia kirjautuneella käyttäjällä on järjestelmässä.

Järjestelmät jakavat sisällön erilleen sivun muista toiminnoista, ja luotu sisältö voidaan joko julkaista sellaisenaan sivuston julkinäkömään tai tallettaa uusiokäyttöä varten. Luotu sisältö siis pysyy tallella palvelimen tietokannassa, ellei ylläpitäjä halua poistaa sitä lopullisesti. Suurin ero järjestelmien välillä löytyy juurikin ylläpitäjälle tarjotuista toiminnoista. Suorilla toiminnoilla sivuston ulkoasuun ei pysty vaikuttamaan kuin rajallisesti, mutta järjestelmiltä löytyy tekstieditori jolla pystyy muokkaamaan CSS -tyylitiedostoa ja näin käyttäjä kykenee vaikuttamaan laajemmin ulkoasun rakenteeseen. Samalla tavoin pääsee käsiksi myös sivupohjiin, jotka löytyvät palvelimen kansioista PHP -tiedostoina.

Tämä jättää kuitenkin sivuston grafiikan suhteellisen koskemattomaksi. Tähän voi vaikuttaa suunnittelemalla oman ulkoasun ja tuomalla sen sisällönhallintajärjestelmään uutena teemana. Uusia teemoja voi myös ladata valmistajan sivustoilta. Teemojen tuonti riippuu käytettävästä järjestelmästä, osalla uuden teeman voi hakea ja ladata suoraan järjestelmällä. Muussa tapauksessa se tapahtuu lataamalla teeman pakkaustiedosto internetistä ja purkamalla se järjestelmän teemakansioon, josta teeman voi järjestelmällä ottaa käyttöön. Yksi teema sisältää useita sivupohjia. Teemoja ja lisäosia on valmistajien sivuilla satoja, ellei jopa tuhansia joista varmasti löytyy sopiva kattamaan yksinkertaisimmat tarpeet.

Lisäosien asennus ja järjestelmän päivitys hoituu edellä mainituilla tavoilla. Tämäkin on järjestelmäkohtainen asia. Useat järjestelmät tarkistavat automaattisesti ovatko ne ajan tasalla ja näiden päivittäminen voi tapahtua automaattisesti käyttäjän pyynnöstä. Toisinaan järjestelmä joudutaan päivittämään manuaalisesti, lataamalla uusin versio valmistajan sivuilta. Päivittäessä järjestelmää on hyvä kytkeä mahdolliset lisäosat ja teemat pois päältä, koska ne eivät välttämättä tue uutta versiota. Samalla kannattaa ottaa järjestelmästä ja sisällöstä varmuuskopio.

5 WEB-PORTFOLIO

Esimerkkityönä siis toimii Internetissä toimiva henkilökohtainen portfolio, jonka avulla on tarkoitus tutustua www-sisällönhallintaan käytännössä, sekä muutamaaan web-sovellukseen. Tarkoitus on siis perehtyä sovellusten toiminnallisiin ominaisuuksiin ja luoda niiden pohjalta yksinkertainen web-portfoliomalli, joka siirtyy työn päätyttyä aktiiviseen käyttöön, työnhakua avustamaan. Esimerkkityössä ei keskitytä sivujen visuaaliseen suunnitteluun, vaikka sitä hiukan sivutaankin. Pääasia on, että käytettävä ohjelmisto täyttää web-portfoliolle asetetut tekniset vaatimukset.

Tekniset vaatimukset saadaan sivulle tulevista työnäytteistä. Nämä työnäytteet ovat periaatteessa mediatekniikan alaa koskevia teoksia, kuvia ja videoita. Lähtökohtana on selvittää, miten esimerkkityöhön valitut sovellukset tukevat näitä digitaalisen median muotoja. Miten ne ovat esiteltävissä ja hallittavissa sovellusten avulla. Web-portfolion työstöä testataan useammalla sovelluksella tarkoituksena selvittää mikä sovelluksista olisi työn tarkoitukselle optimaalisin.

5.1 Ohjelmistot

Tässä opinnäytetyön vaiheessa käyn ensin läpi esimerkkityöhön käytettävät työkalut, niiden roolit ja perustelut niiden valinnalle. Budjetin takia työhön on käytetty ainoastaan vapaassa levityksessä olevia sovelluksia ja työalustana toimii koti-pc, joka pitää tässä tapauksessa palvelimen virkaa. Opinnäytetyön jälkeen lähitulevaisuudessa portfolion alusta on tarkoitus siirtää omalle palvelimelleen ja aktiiviseen käyttöön.

Koska vapaassa levityksessä olevia www-sisällönhallintasovelluksia on kymmeniä jo pelkästään PHP -kielelle, joka toimii yhtenä rajapintana sopivaa sovellusta valittaessa, on aiheellista tutustua tarjontaan ja päättää mitkä näistä PHP-kieleen pohjautuvista sovelluksista otetaan käyttöön esimerkkityötä varten. Internet tarjoaa runsaasti lähteitä ja arvosteluja eri ohjelmistoille. Pienellä tutkimustyöllä valitsin ehdokkaiksi kolme sovellusta: Drupal, Joomla! ja Wordpress. Näistä ehdokkaista

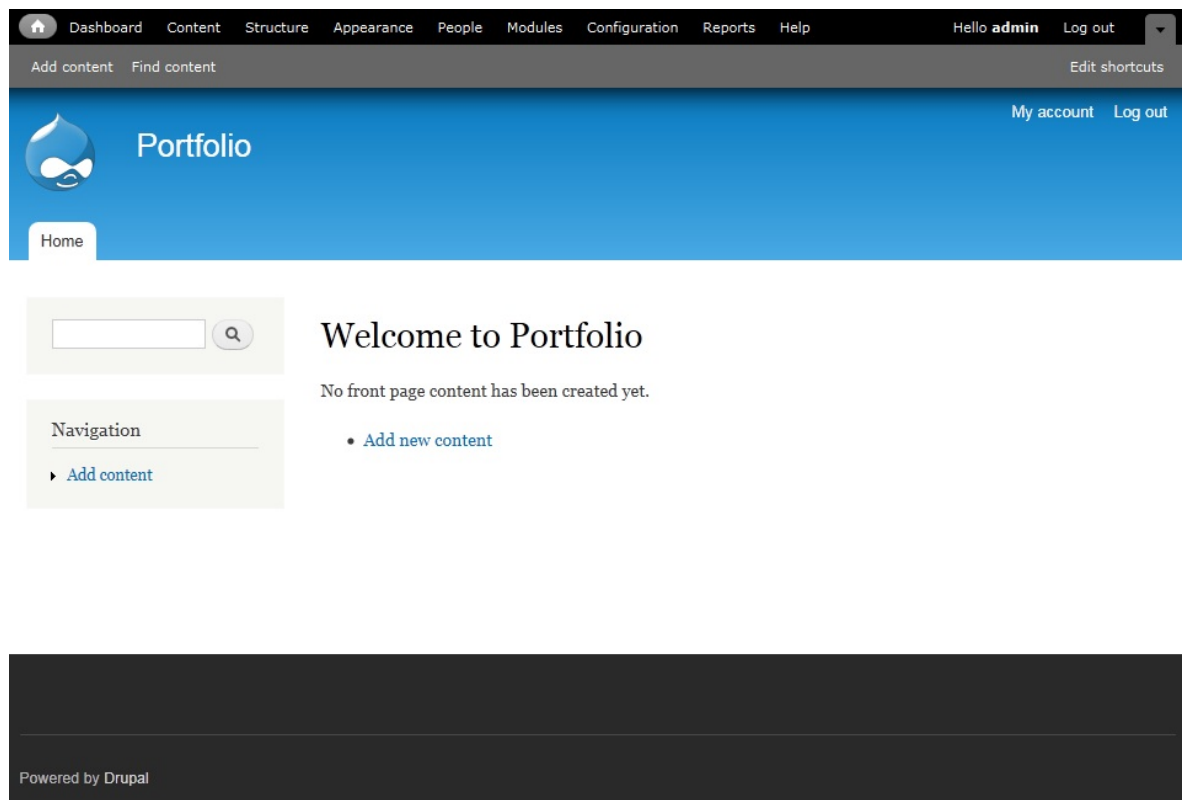
käydään internetissä ahkeraa keskustelua. Varsinkin kahdella ensimmäisellä on laaja käyttäjäyhteisö ja ne ovat nähtävästi aina taistelleet kärkipaikasta. Kaikki ehdokkaat ovat avoimen lähdekoodin sovelluksia ja laajasti kehitettyjä. Tämä on omissa yhteisöissään toimivien tuhansien jäsenien ansiota. Yhteisöissä he auttavat kehittämään ja laajentamaan sovellusten käyttömahdollisuuksia. Vaikka Wordpressiä pidetään altavastaaajana CMS-sodassa, se on ehdottomasti blogisovellusten kuningas tehokkuudessaan. Kullakin sovelluksella on omat puutteensa ja vahvuutensa, joita käsitellään tarkemmin tulevissa luvuissa. Kaikilla valituilla sovelluksilla voi kuitenkin luoda internetsivuja käyttötarkoituksesta riippumatta. Kaikki riippuu sovellusten kyvyistä ja käyttäjän perehtyneisyydestä järjestelmään. Sovellusten hyödyllisyys on siis käyttäjäkohtainen ja tämä ymmärretään usein väärin. Jos et esimerkiksi pidä Drupalin käyttämisestä, se ei tee siitä vähemmän arvokasta, se juuri tekee siitä vähemmän hyödyllisen sinulle. (Stiffler-Dean 2009; opensourcecms 2011.)

Kuten jo luvun alussa todettiin, kaikki valitut sisällönhallintajärjestelmät ovat vapaan lähdekoodin sovelluksia. Tämä tarkoittaa, että niiden jakelu tapahtuu GNU – hankkeen yleisen lisenssin (engl. GNU General Public License tai GPL) alaisuudessa. GPL lisenssi on suunniteltu takaamaan käyttäjän vapaus jakaa ja muuttaa sovellusta. Lisenssillä varmistetaan, että ohjelma on vapaa kaikille käyttäjille. GPL lisenssiä sovelletaan pääasiallisesti Free Software Foundationin (FSF) ohjelmiin ja mihin tahansa muuhun ohjelmaan, jonka tekijät sitoutuvat sen käyttöön. Kun GPL:n ehdoissa puhutaan vapaista ohjelmistoista, siinä on kyse vapaudesta, ei hinnasta. GPL -lisenssi on varmistaa, että käyttäjä saa sovelluksen lähdekoodin ja voi muuttaa ohjelmaa, tai käyttää osia siitä omissa vapaisissa ohjelmissaan. (gnu 1991.)

5.1.3 Drupal

Drupal on avoimen lähdekoodin ohjelmisto, jota ylläpitää ja kehittää yli 630000 käyttäjän ja kehittäjän yhteisö. Vapaaehtoistyötä tekevä Drupal -yhteisö antaa tukea IRC-kanavien ja foorumien kautta, kuten myös kasvatusten Drupal - tapahtumissa. Yhteisö on myös luonut dokumentaation Drupalille, joka kattaa suu-

rimman osan sovelluksen kanssa työskentelyyn liittyvistä aiheista. Vapaa lisenssi-pohja takaa, että kaikki voivat osallistua Drupalin kehittämiseen. Dries Buytaert aloitti Drupal -ohjelmiston, käyttämällä sitä keskustelualueena vuonna 1999. Suurin piirtein vuoden kuluttua, kun ihmiset kiinnostuivat käyttämään ja edistämään Drupalia, hanke muutettiin avoimeen lähdekoodiin. Drupal.org tuli verkkoon vuonna 2001, ja yhteisö kehittyi nopeasti vuonna 2005 useiden koodimuutosten ja konferenssien myötä. (drupal 2011.)



KUVIO 6. Drupal -sisällönhallintajärjestelmän päänäkymä.

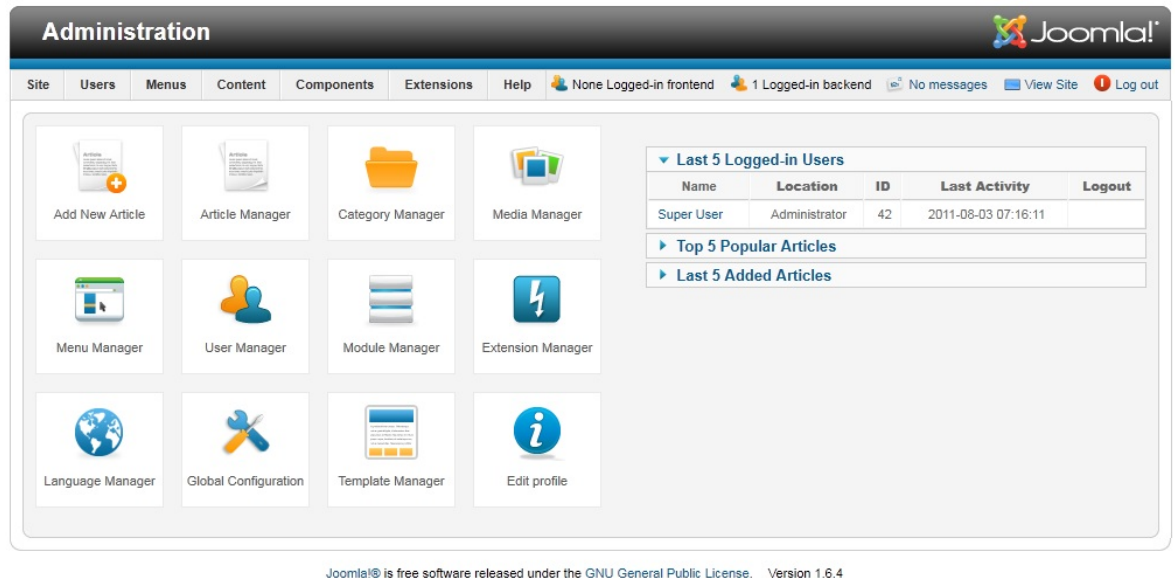
Drupal sopii erityisesti käyttäjille, jotka mieluummin koodaavat sisällön sivulle käsin, kuin käyttäisivät WYSIWYG –editoria. Drupal sopii myös niille, jotka haluavat panostaa sivuston rakennekoodin säätämiseen. Drupal muistuttaa enemmän kehitystyökalua kuin perinteistä sisällönhallintajärjestelmää. Tämä ei kuitenkaan tarkoita sitä, että vain kehittäjät voisivat käyttää järjestelmää mutta, että he suosivat enemmän Drupalia kuin kahta muuta sisällönhallintajärjestelmää. Mielenkiintoista on, että enemmän kehittäjäystävällinen sovellus ei automaattisesti tee siitä käyttä-

jäystävällistä. Itse asiassa kehittäjiä on tehtävä lujasti töitä, että saavat tarpeidensa mukaisia lopputuotteita. Verrattuna muihin ehdokkaisiin, Drupal tarjoaa kymmeniä tunteita ja toimintoja enemmän, joita voidaan käyttää sovelluksen kehittämiseen. Jokaisella tietosolulla on omat komennot ja tunnisteet, jotka voidaan sijoittaa muualle ja näin manipuloida sivuston toimintoja. Tämä mahdollistaa erittäin intensiivisen kokemuksen. Niille jotka eivät ole niin kehittäjämielisiä, tämä voi olla suuri haaste. Mutta kehittäjät, jotka tuntevat koodia, voivat kirjaimellisesti eksyä hienojen sivustojensa kehittämiseen. Drupalilla luodut sivustot voivat käyttäytyä upeasti ja sisältävät hienoja toimintoja, mutta ongelmaksi usein muodostuu se, etteivät sivustot aina ulkoasuiltaan näytä erityisen hienoilta. Loppujen lopuksi on hyvin vähän Drupal -sivustoja jotka näyttävät ja tuntuvat yhtä hyviltä kuin ne toimivat. Tämän voi todeta erityisesti tarkastelemalla teema -osiota Drupalin pääsivustolla. Edistyksellisestä kehityksestä huolimatta, ohjelmisto olisi melkein täydellinen, jos sen käytettävyyttä korostettaisiin ja muotoilun käyttöliittymä olisi parempi. (Stiffler-Dean 2009.)

5.1.4 Joomla!

Suunnittelijat valitsevat Joomla sen moottorin tarjoamien, ulkoasua parantavien ominaisuuksien vuoksi. Uudet tulokkaat ja internetsivujen hallinnoijat pitävät Joomlaista, koska se on erittäin helppokäyttöinen ja sitä on helppo muokata. Näitä ominaisuuksia edesauttaa myös se, että yhä useammat kehittäjät ovat jatkuvasti luomassa uusia työkaluja, joita on helpompi ymmärtää. Joomla on myös monien kehittäjien valinta, koska se tarjoaa suuren kapasiteetin kehitykselle ja räätälöinnille. Uusi malli–näkyvä–käsittelijä -kehys (model-view-controller eli MVC) on rakennettu siten, että asiantuntijat voivat ohittaa CMS -ytimen, ilman muutoksia alkuperäiseen koodiin. Silti Joomla on kaukana täydellisestä. Koodaajan näkökulmasta, se ei ole yhtä joustava kuin Drupal ja siinä missä Drupal voi ajaa useita sivustoja yhdellä taustajärjestelmällä ja tietokannalla, Joomla puuttuu tämä kyky. Se on juuri tarpeeksi tehokas ollakseen hyödyllinen, mutta ei riitä suurempaan kehittelyyn. Suorassa vertailussa Wordpressin kanssa Joomlailla on vielä matkaa olemaan käyttäjäystävällinen. Stiffler-Dean kokee, että ihmiset joilla on ollut käytettävissään molemmat järjestelmät, niin niillä joilla oli vähäinen tuntemus internetsivustojen

toiminnasta, valitsivat aina Wordpressin. Joomla on tarpeeksi yksinkertainen, että sen käyttö voidaan opettaa kelle tahansa, mutta ei tarpeeksi, että kukaan haluaisi oppia sitä. (Stiffler-Dean 2009.)



KUVIO 7. Joomla:n hallintanäkymä.

5.1.5 Wordpress

Wordpress on erinomainen järjestelmä jos tarve on luoda verkkosivusto, jonka avulla voi nopeasti saada ajatuksia ulos Internetiin. Vaikka Wordpress on pääasiallisesti tarkoitettu blogien ylläpitoon, se voidaan ohjelmoida toimimaan monilla muilla mielenkiintoisilla tavoilla. Ajatellaan esimerkkinä työyhteisön keskinäinen tietojen jakaminen ja dokumentointi yrityksen intranetissä. On hyvä tietää, että Wordpress on erittäin helppo käyttää ja asentaa. Stiffler-Dean kertoo, kuinka hänellä on ollut useita asiakkaita, jotka ovat uusia Internetin käytössä ja sivustojen ylläpitämisessä, mutta kuinka he ovat kyenneet käyttämään Wordpressiä nopeammin kuin kahta muuta järjestelmää. Koska Wordpress on jo valmiiksi kehitetty toimimaan blogina, se voidaan asentaa tähän käyttöön välittömästi. Kommenttikentät ovat rakennettu osaksi järjestelmää, kuten myös ping -palvelut, useat bloggaus -profiilit, paluuvitteet ynnä muuta. Näille järjestelmille ei tarvitse aina edes

tehdä mitään muutoksia, sillä ne usein toimivat juuri kuten käyttäjä haluaa. Wordpressiä ei kuitenkaan kannata käyttää kaikkeen. Vaikka Wordpressillä voi luoda esimerkiksi ostoskärry -sovelluksia, näin ei välttämättä kannata tehdä. Järjestelmä on suunniteltu siten, etteivät ihmiset kokisi tarpeelliseksi tehdä muutoksia järjestelmään vaan että se toimisi tarpeeksi hyvin yksinkertaisemmilla vakioasetuksilla. Sovellusta on tarkoitus käyttää tietyllä tavalla ilman muutoksia, koska kaikki muutokset, joita sille tehdään saattavat hyvinkin aiheuttaa järjestelmän rikkoutumisen. Useat käyttäjät valittavat, että Wordpress ei ole kehittäjäystävällinen ja oikeutetusti niin. (Stiffler-Dean 2009.)

KUVIO 8. Wordpress hallintanäkymä.

5.2 Valinta ja käyttöönotto

Ennen tutustumista ohjelmiin käytännössä on aiheellista määritellä vaatimukset web-portfolioille. Pääsääntöisesti on huomioitava tekniset ominaisuudet ja sovellusten tarjoamat toiminnot, ja niiden hyödyntäminen. Portfolion visuaalinen ilme on toisarvoinen kriteeri mutta otettava silti huomioon. Tämä siksi, että sivuston yleinen visuaalinen ilmeikkyyden antaa vierailijoille ensivaikutelman ja ensivaikutelma

avaa mielenkiinnon sivustoa kohtaan. Jos vierailijoiden mielenkiinto ei pysy ensivaikutelmaa pitemmälle, ei portfolio ole hyödyllinen kenellekään. Toissijaisia kriteerejä ovat myös muut käyttömahdollisuudet, kuten blogin pitäminen.

Teknisesti portfolio ei vaadi paljoa sisällönhallintajärjestelmältä. Tärkeintä on, että sillä pystyy hallinnoimaan ja esittämään digitaalista mediaa. Esimerkkityöllä otetaan selvää miten kukin järjestelmä tämän toteuttaa ja mikä näistä menetelmistä olisi paras työn tarkoitukselle. On huomioitava myös, kuinka mediasisällön voi kategorisoida ja miten tätä sisältöä on mahdollista selailla. Myös sivun muut tekniset ominaisuudet ovat tärkeitä, kuten miten helposti sivuja voi ylläpitää ja päivittää tarpeen mukaan. Sovelluksen tekninen joustavuus ei sinänsä ole suuri huolenaihe, sillä esimerkkityössä ei luoda uusia liitännäisohjelmia tai uutta ulkoasua. Jatkuvuuden kannalta tämä on kuitenkin yksi asia, joka kannattaa ottaa huomioon. Todennäköisesti sivun ulkoasu tulee vielä muuttumaan, mutta esimerkkityölle riittää valmiista elementeistä rakennettu kokonaisuus.

Jokainen valituista sisällönhallintajärjestelmistä asentuu palvelimelle automaattisesti pyytämällä ainoastaan käytettävän tietokannan viitetiedot ja tunnukset. Mahdollisten ristiriitojen välttämiseksi loin jokaiselle järjestelmälle oman tietokannan, joihin kuitenkin pääsee pääkäyttäjän tunnuksilla. Jokainen järjestelmä on heti asennuksen jälkeen käyttövalmis.

Tässä vaiheessa huomaa, että Stiffler-Deanin artikkeli kuvasi hyvin järjestelmien eroja. Tässäkin tapauksessa Drupal tarjoaa kilpailijoitaan enemmän toimintoja ja mahdollistaa tarkemmat määrittelymahdollisuudet sivustolle. Joomla ei tarjonnallaan jää kauaksi Drupalista mutta molemmat sisällönhallintajärjestelmät kärsivät käyttöjärjestelmän hankaluudesta, ne eivät ole kovin käyttäjäystävällisiä. Esimerkkityön testisivustoja luodessa näillä järjestelmillä oli hankala löytää haluamiaan toimintoja mutta löytyessään, niillä oli helpompi saada aikaiseksi haluamiaan tuloksia. Wordpress on kirkkaasti joukon käyttäjäystävällisin vaihtoehto, vaikka se tarjoaakin näennäisesti vähiten toimintoja käytettäväksi.

Järjestelmillä ilmeni kuitenkin yhteinen ongelma. Ne eivät tue kovin suurikokoisten tiedostojen siirtämistä palvelimelle. Tiedostokoon ylärajana on kaksi megatavua, mikä ei ole tarpeeksi kaikille video -tiedostoille. Pienellä tutkimustyöllä selvisi, että tämän ongelman voi ohittaa tekemällä muutoksia järjestelmän asetustiedostoihin.

Toisaalta ratkaisuna voisi toimia myös videon lataaminen ulkoiseen palveluun, esimerkiksi Youtubeen tai Vimeoon. Tämä olisi hyvä ratkaisu, sillä se säästäisi palvelimen kapasiteettia ja työnäytteet pääsisivät paremmin esille julkisesti.

Yhteenvetona, testatuista sisällönhallintajärjestelmistä ei löydy huonoja vaihtoehtoja. Kyse on pääasiassa käyttäjän omista mieltymyksistä ja luotavan sivuston käyttötarkoituksesta. Esimerkkityön alustaksi valitsin lopulta Wordpressin, sillä se tarjoaa kaikki tarvittavat toiminnot pienelle portfoliolle ja sivuston kehitykselle. Portfolio ei ole sivustona niin mittava, että se tarvitsisi suuren määrän eri toimintomahdollisuuksia. Wordpress sopii muutenkin portfoliolle, koska se mahdollistaa nopeimman sisällönjulkaisun sivustolle ja on erittäin kevyt käyttää, myös sisällön hallinnointi on Wordpressillä varsin selkeää. Ulkoasun paikanpitäjäksi valitsin Showcase -teeman, joka kuvauksensa mukaan soveltuu juurikin portfoliosivustoille. (KUVIO 10) Teema ei tarjoa paljon toimintoja ulkoasun muuttamiseen mutta se tarjoaa tarpeeksi hillityn ja samalla tyylikkään ulkoasun portfoliolle.

Tärkein vaatimus on saada digitaalista informaatiota sivustolle. Järjestelmällä voidaan lisätä digitaalista mediaa palvelimelle, joko käyttäjän omalta tietokoneelta tai toiselta sivustolta. Tämä tapahtuu Lisää -toiminnon avulla. (KUVIO 9) Tällä toiminnolla media saadaan vasta palvelimelle, eikä se ole vielä itse sivustolla tarkasteltavissa mutta löytyy järjestelmän mediakirjastosta, josta sen voi linkittää artikkeliin. Järjestelmällä ei ole omaa sovellusta videotiedostojen toistoon, vaan tätä varten käyttäjä joutuu lataamaan tiedoston toistoon kykenevän lisäosan. Testasin tätä lisäämällä Flash Video -tiedoston (FLV) järjestelmään ja asensin toistoa varten Hana Flv Player -lisäosan. Tämän jälkeen videotiedoston sijainti pitää upottaa lisäosan tunnisteeseen, jotta sitä pystyisi toistamaan julkinäkymässä. Lisäosasta voin todeta sen olevan aavistuksen viallinen mutta asiansa hoitava. Videon kuvasuhde on väärä, eikä lisäosalla voi tätä määritellä oikeaksi. Video myös toistuu jostain syystä kaksi kertaa. Youtubesta upotetuilla videoilla ei tätä ongelmaa ole. (KUVIO 10) Kuvat voidaan lisätä sivulle samalla tavalla kuin videot, eivätkä ne tarvitse ylimääräisiä lisäosia näkyäkseen. Kannattaa pitää mielessä kuvien koko, sillä liian suuret kuvat rikkovat sivun ulkoasun. Kuvan koon voi kuitenkin erikseen määritellä medianhallinnasta ja 620 pikseliä leveä kuva mahtui vielä sivuston rakenteiden sisään.

KUVIO 9. Median lisäys Wordpressiin.

KUVIO 10. Työnimi BlueBin julkinäkymä.

6 LOPPUARVIOINTI

Opinnäytetyötä aloittaessa, sisällönhallintajärjestelmät olivat täysin uusi tuttavuus. Tämä oli yksi syy, miksi tämä tuli opinnäytetyön aiheeksi. Aihe oli ennestään tuntematon mutta samalla myös levinnyt käytäntö, jonka rooli vain kasvaa sivuston suunnittelussa. Ilman mitään aikaisempaa kokemusta aiheesta, oli hankaluuksia löytää sopivaa aihetta opinnäytetyön ensiluvuille. Perusidea sisällönhallintajärjestelmistä tuli kuitenkin nopeasti tutuksi. Hyvin lyhyesti, kyseessä on järjestelmä jolla hallitaan sisältöä ja tämän opinnäytetyön tapauksessa, digitaalista sisältöä palvelimella.

Syvempi teoria oli kuitenkin pitkään hukassa, eikä suomenkielistä materiaalia löytynyt kuin hyvin niukasti. Bob Boikon Content Management Bible auttoi huomattavasti sisällönhallinnan teorian ymmärtämisessä, ennen varsinaisen kirjoitusprosessin alkua. Vaikka osa määritelmistä oli ennalta tuttuja, Boikon teos selkeytti asioita ennestään ja auttoi ymmärtämään tuntemattomampia käsitteitä, kuten sisällön rakenne, toiminnot ja sisällönhallinnan prosessit. Seuraavaksi oli selvitettävä millaisia hallintajärjestelmiä oli tarjolla, mitkä niiden toimintaperiaatteet ovat ja mihin tekniikoihin ne pohjautuvat.

Opinnäytetyössä tutustuttiin vain muutamamaan PHP -pohjaiseen järjestelmään. Tämä oli kuitenkin pakollinen rajapinta, sillä järjestelmiä on liikaa käsiteltäväksi yhdessä opinnäytetyössä. Tällä kuitenkin saavutettiin käytännön tutustuminen näihin järjestelmiin ja helppokäyttöinen alusta web-portfoliolle, joka tulevaisuudessa tulee muuttumaan niin toiminnoiltaan, kuin ulkoasultaankin. Jatkokehitykselle on siis paljon suunnitelmia mutta opinnäytetyö oli merkittävä apu itseopiskelun kannalta. Mielestäni opinnäytetyöllä saavutettiin alkuperäinen tavoite, vaikkakin se jäi varsin suppeaksi verrattuna aihealueen laajuuteen.

LÄHTEET

Addison, Doug. 2004. *Small Websites, Great Results*.
Scottsdale, AZ, USA: Paraglyph Press.

Battles, Ryan. *Is a Content Management System Right for You?* Www-dokumentti.
Saatavissa:
http://joviawebstudio.com/index.php/blog/is_a_content_management_system_right_for_you/. Muutettu 23.12.2008. Luettu 1.9.2011.

Boiko, Bob. 2005. *Content Management Bible*, 2. painos.
Indianapolis, IN, USA: Wiley Publishing, Inc..

Chapple, Mike. *SQL Fundamentals*. Www-dokumentti. Saatavissa:
<http://databases.about.com/od/sql/a/sqlfundamentals.htm>. Luettu 1.9.2011.

Chapple, Mike. *Introduction to SQL*. Www-dokumentti. Saatavissa:
<http://databases.about.com/od/sql/a/sql.htm>. Luettu 1.9.2011.

Delisle, Marc. 2009. *Mastering phpMyAdmin 3.1 for Effective MySQL Management*.
Olton Birmingham, GBR: Packt Publishing Ltd.

Doyle, Bob. *CMS Genesis: Who Did What When?* Www-dokumentti. Saatavissa:
<http://www.econtentmag.com/NewsLetters/NewsletterReader.aspx?NewsletterID=178#1>. Luettu 1.9.2011.

Drupal, virallinen sivusto. Www-dokumentti. Saatavissa: <http://drupal.org>. Luettu 1.9.2011.

Duckett, Jon. 2010. *HTML, XHTML, CSS and JavaScript*.
Hoboken, NJ, USA: Wrox.

Eden, Bradford Lee. 2006. *Content Management Systems*.
Bradford, GBR: Emerald Group Publishing Ltd.

Gilbane, Frank. What Is Content Management? Www-dokumentti. Saatavissa: <http://gilbane.com/artpdf/GR8.8.pdf>. Muutettu 8.11. 2000. Luettu 1.9.2011.

GNU Operating Systems. GNU General Public License, version 2. Www-dokumentti. Saatavissa: <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>. Muutettu 1.6.1991. Luettu 1.9.2011.

Hart-Davis, Guy. 2004. HTML QuickSteps.
Emeryville, CA, USA: McGraw-Hill Professional Publishing.

Hauschildt, Sofia. 2010. Cms Made Simple 1.6.
Olton Birmingham, GBR: Packt Publishing Ltd.

OpensourceCMS, virallinen sivusto. Www-dokumentti. Saatavissa: <http://php.opensourcecms.com>. Luettu 1.9. 2011.

Oppel, Andrew J.. 2005. SQL Demystified.
Emeryville, CA, USA: McGraw-Hill Professional Publishing.

PHP, virallinen sivusto. History of PHP. Www-dokumentti. Saatavissa: <http://fi2.php.net/manual/en/history.php.php>. Luettu 1.9.2011.

phpMyAdmin, virallinen sivusto. Www-dokumentti. Saatavissa: <http://www.phpmyadmin.net>. Luettu 1.9.2011.

Robertson, James. Why every small website needs a CMS. Www-dokumentti. Saatavissa: http://www.steptwo.com.au/papers/cmb_needcms/index.html. Muutettu 13.1.2003. Luettu 1.9.2011.

Schafer, Steven M.. 2010. HTML, XHTML, and CSS Bible, 5. painos.
Hoboken, NJ, USA: Wiley.

Schmitt, Christopher. 2008. Professional CSS : Cascading Style Sheets for Web Design, 2. painos.
Hoboken, NJ, USA: Wiley.

Site Dynamics, Web Solutions. Static and dynamic websites... what's the difference? Www-dokumentti. Saatavissa:

<http://www.sitedynamics.ca/rootfolder/blog/?p=1>. Muutettu 13.5.2009. Luettu 1.9.2011.

Stiffler-Dean, Tim. Wordpress vs Joomla vs Drupal. Www-dokumentti. Saatavissa: <http://www.goodwebpractices.com/other/wordpress-vs-joomla-vs-drupal.html>. Muutettu 9.3.2009. Luettu 1.9.2011.

WampServer virallinen sivusto. Www-dokumentti. Saatavissa: <http://www.wampserver.com/en/presentation.php>. Luettu: 1.9.2011.

Wilton, Paul; Colby, John. 2005. Beginning SQL. Hoboken, NJ, USA:Wiley.

W3C, virallinen sivusto. Www-dokumentti. Saatavissa: <http://www.w3.org/standards/webdesign/htmlcss#whatcss>. Luettu 1.9.2011.