

Anssi Savolainen

Verkkosivuston suunnittelu ja toteutus WordPressillä

Case: Ylivieskan kaupungin koulutusorganisaatiot

**Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Mediatekniikan koulutusohjelma
Joulukuu 2016**

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Yksikkö Ylivieska	Aika Joulukuu 2016	Tekijä/tekijät Anssi Savolainen
Koulutusohjelma Mediatekniikka		
Työn nimi Verkkosivuston suunnittelu ja toteutus WordPressillä		
Työn ohjaaja Hannu Puomio	Sivumäärä 23	
Työelämäohjaaja Mikko Hannula		
<p>Opinnäytetyön aiheena oli suunnitella ja toteuttaa verkkosivusto Ylivieskan alueen koulutusorganisaatioille käyttäen WordPress-julkaisualustaa. Sivuston tarkoituksena oli koota kerätysti yhteen paikkaan organisaatioiden yleistiedot sekä turvallisuusohjeistuksia koulujen henkilökunnille.</p> <p>Lisäksi työhön kuului sivuston käytön sekä muokkauksen opastus tulevalle, sivustoa ylläpitävälle henkilölle.</p>		
Avainsanat CSS, Käyttöliittymä, Sisällönhallinta		

ABSTRACT

CENTRIA UNIVERSITY OF APPLIED SCIENCES Ylivieska	Date December 2016	Author Anssi Savolainen
Degree programme Media technology		
Name of thesis Designing and building a website using WordPress		
Instructor Hannu Puomio		Pages 23
Supervisor Mikko Hannula		
<p>The subject of this thesis was to design and implement a website for educational organisations on the Ylivieska area using WordPress content management system. Purpose of the website was to gather basic informations and safety instructions to one place for the personnel of the organisations.</p> <p>The project also consisted of guiding the website's future administrator to using and modifying the website.</p>		
Key words CSS, User interface, Content management		

KÄSITTEIDEN MÄÄRITTELY

Sisällönhallintajärjestelmä	Järjestelmä, jolla pyritään helpottamaan verkkopalvelun sisällön hallitsemista mahdollisimman tarkoituksenmukaisesti.
CSS	(lyh. Cascading Style Sheets) on tyylikieli, jolla voidaan määritellä www-dokumentin tyyliohjeita, kuten kirjaisinmuotoiluja ja värejä.
HTML	Avoimen standardin kuvauskieli, jolla luodaan hypertekstiä sisältäviä www-dokumentteja, joita luetaan selainohjelmalla. Hypertekstillä tarkoitetaan dokumentissa olevia linkkejä, joista painamalla pääsee toisiin dokumentteihin.
WWW-palvelin	Tarkoitetaan tietokonetta tai ohjelmistoa, joka jakaa HTML-kielellä luotuja dokumentteja asiakasohjelmille ja -koneille.

TIIVISTELMÄ
ABSTRACT
KÄSITTEIDEN MÄÄRITTELY
SISÄLLYS

1 JOHDANTO.....	1
2 WORDPRESS-SISÄLLÖNHALLINTAJÄRJESTELMÄ.....	2
2.1 Asennus.....	3
2.2 Ohjausnäkyvä.....	4
2.3 Asetukset.....	5
2.4 Käyttäjäroolit.....	6
2.5 WordPressin muokattavuus.....	7
3 SIVUSTON SUUNNITTELU.....	8
3.1 Yleistä.....	8
3.2 Teema.....	9
3.3 Ulkoasu ja rakenne.....	10
4 SIVUSTON TOTEUTUS.....	12
4.1 Sivut ja sisältö.....	12
4.2 Lapsiteema.....	15
4.3 Lisäosat.....	16
4.4 Käyttäjät.....	18
5 YHTEENVETO.....	21
LÄHTEET.....	22
KUVIOT	
KUVIO 1. Ohjausnäkyvä.....	5
KUVIO 2. Twenty Thirteen teema.....	9
KUVIO 3. Sivustolla käytetyt värit.....	10

1 JOHDANTO

Tämän opinnäytetyön aiheena oli suunnitella ja toteuttaa verkkosivusto käyttäen WordPress-julkaisualustaa sekä pohtia sen soveltuvuutta tiedonjakoon keskittyneeseen sivustoon. Verkkosivut tulevat Ylivieskan alueen eri koulutusorganisaatioiden henkilökuntien sekä opiskelijoiden vanhempien käyttöön. Sivuilla esitellään kukin organisaatio ja selvitetään niiden erilaisia turvallisuus- ja muita tarpeellisia ohjeistuksia. Organisaatioihin kuuluu alueen peruskoulut, lukiot, päiväkodit sekä opistot.

Opinnäytetyö koostuu neljästä pääluvusta. Ensimmäisessä pääluvussa esittelen WordPress-sisällönhallintajärjestelmän yleisesti sekä kerron sen eri käyttöönottovaiheet. Toisessa pääluvussa kerron opinnäytetyöni suunniteluvaiheista, miten työ aloitettiin, millä työkaluilla ja tiedoilla. Toinen pääluku sisältää myös yleistä tietoa verkkosivujen suunnittelusta ja toteutuksesta. Kolmas pääluku sisältää asiaa ja pohdintaa verkkosivujen teknisestä toteutuksesta. Kolmannessa pääluvussa kerron opinnäytetyön sivuston sisällön toteutuksesta, käytetyistä lisäosista, sivuston käyttäjistä sekä lapsiteeman käyttöönotosta. Neljäs pääluku sisältää yhteenvedon opinnäytetyöstäni.

Koska digitaalinen media on kokoajan muuttuvaa ja helposti päivitettävää nykyaikaan, on lähteinä opinnäytetyössä käytetty enimmäkseen eri verkkosivustoilla olevaa materiaalia. Eniten käytettyinä lähteinä ovat olleet WordPressin omat kotisivut sekä sen sähköinen ohjekirja WordPress Codex.

2 WORDPRESS-SISÄLLÖNHALLINTAJÄRJESTELMÄ

WordPress on nykyaikainen henkilökohtainen avoimen lähdekoodin julkaisualusta, jonka painopisteinä ovat esteettisyys, web-standardit ja helppo käytettävyys (WordPress 2016b). WordPress on tehty PHP-ohjelmointikielellä ja se käyttää MySQL-tietokantajärjestelmää tietojen tallentamiseen. Ensimmäinen WordPress versio julkaistiin vuonna 2003. WordPressin edeltäjänä toimi vuonna 2001 alkunsa saanut b2/cafelog niminen palvelu, jota käytettiin erilaisten blogien tuottamiseen. (WordPress Codex 2016a.) Alunperin WordPress luotiinkin ainoastaan blogien tuottamista varten, mutta WordPressin kehittäjät kuitenkin myöhemmin huomasivat, että palvelua voitaisiin käyttää paljon suurempien ja monimutkaisempienkin sivustojen rakentamiseen ja ylläpitämiseen.

Uusien ohjelmistoversioiden myötä WordPressiin julkaistiin ominaisuus käyttää erilaisia lisäosia eli plugineita, joiden avulla sivustoja voitiin rakentaa ja kehittää entistä monipuolisimmiksi. Nykyään ladattavat pluginit ovat olennainen osa WordPressiä, eri plugineita löytyykin WordPressin virallisilta kotisivuilta jo yli 46 tuhatta tätä kirjoittaessa (WordPress 2016d). Osa monimutkaisemmista ja enemmän ominaisuuksia tarjoavista plugineista ovat kuitenkin maksullisia. Pian pluginien lisäys ominaisuuden jälkeen WordPress laajeni niin sanotuilla staattisilla sivuilla sekä uudella sivuston muokkauspaneelilla. Tämän paneelin avulla sivujen rakentaminen ja päivitys oli entistä helpompaa. (WordPress Codex 2016a.)

WordPressin käyttäjät voivat asentaa ja vaihtaa sivuillaan käytettävää teemaa, jonka avulla voidaan muuttaa sivuston ulkoasua ja toimivuutta. Jokainen WordPressillä luotu sivusto vaatii jonkinlaisen teeman käytön, joka on luotu käyttäen PHP-, HTML- ja CSS-ohjelmointikieliä. Teemojen asennus tapahtuu WordPressin käyttöliittymästä (engl. Dashboard) tai vaihtoehtoisesti lataamalla teeman kansio suoraan sivuston palvelimelle. Teeman käyttämiä PHP-, HTML-, ja CSS-tiedostoja voidaan muokata, jos sivun ulkoasua halutaan muokata enemmän. Kuten lisäosiakin, WordPressin teemoja on ilmaisia sekä maksullisia, joista ilmaiset löytyvät suoraan WordPressin teema-valikosta ja maksullisia löytyy kauppapaikoista sekä yksittäisiltä WordPress kehittäjiltä. WordPressin kehitys- ja päivitystyö on ollut tiivistä sillä uusi, entistä paremmilla ominaisuuksilla varustettu versio on julkaistu vähintään kerran vuodessa sen julkaisuvuodesta 2003 lähtien (WordPress Codex 2016a).

Muita avoimen lähdekoodiin perustuvia WordPressin tapaisia PHP-pohjaisia sisällönhallintajärjestelmiä ovat esimerkiksi Joomla!, Drupal, Mambo, PHP-Nuke, SilverStripe, TYPO3 ja MODX.

2.1 Asennus

WordPress on hyvin tunnettu sen helpposta asennusprosessista, joka vie keskivertokäyttäjältä aikaa noin viisi minuuttia. Nykyään monet verkkopalveluja tarjoavat organisaatiot tarjoavat myös työkaluja, joilla WordPressin asennus tapahtuu automaattisesti. WordPressin uusimman version on 4.6.1 ja se vaatii toimiakseen www-palvelimen joka tukee vähintään PHP:n versiota 5.6, MySQL versiota 5.6 tai vaihtoehtoisesti MariaDB:n versiota 10.0. Käytettäviksi palvelimiksi suositellaan Apachea ja Nginxiä, jotka tarjoavat varmat ja ominaisuusriikkaat palvelut. (WordPress 2016a.)

Ennen WordPressin asentamista tulee vielä varmistaa että seuraavat asiat ovat kunnossa:

- Pääsy www-palvelimelle
- Tekstieditori
- FTP-ohjelma
- www-selain

(WordPress Codex 2016b.)

WordPressin asennus aloitetaan lataamalla sen uusimman versio. WordPress on saatavilla useilla eri kielillä, myös suomeksi. Suomenkielisen version voi ladata WordPressin suomalaisilta sivuilta osoitteesta <http://fi.wordpress.org/>. Tiedostojen latauksen ja niiden purkamisen jälkeen luodaan WordPressille tietokanta palvelimelle sekä MySQL tai MariaDB käyttäjä, jolla on kaikki oikeudet tietokannan käsittelyyn. Lisäksi voidaan käydä nimeämässä uudelleen wp-config.sample.php niminen tiedosto esimerkiksi wp-config.php muotoon, jonka jälkeen tähän tiedostoon lisätään tietokannan tiedot. Näitä tietoja ovat mm:

- DB_NAME (tietokannan nimi)
- DB_USER (tietokannan käyttäjätunnus)
- DB_PASSWORD (tietokannan salasana)
- DB_HOST (MySQL-palvelin)
- DB_CHARSET (tietokantatauluissa käytettävä merkistö)
- DB_COLLATE (tietokantataulujen tekstikentissä käytettävä merkistön tallennusmuoto)

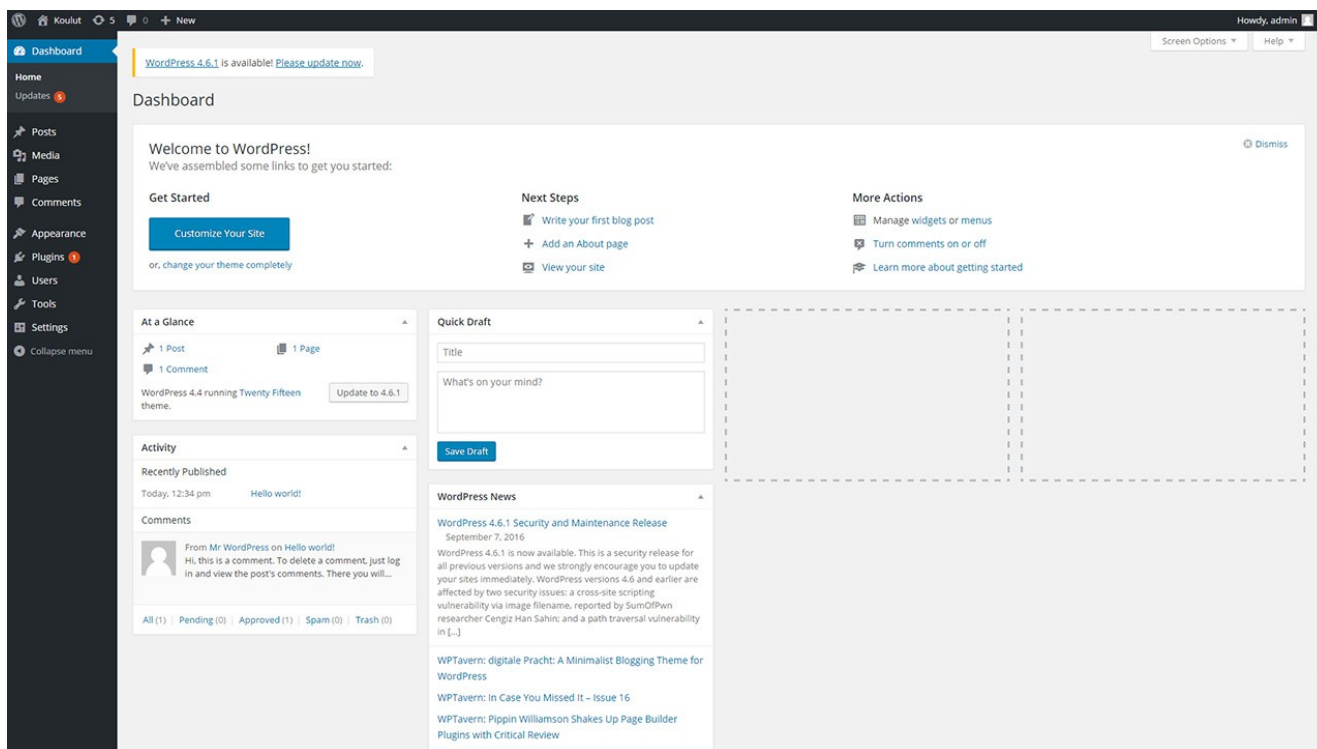
(WordPress Codex 2016b.)

Seuraavassa vaiheessa siirretään ladatut tiedostot WordPressin tulevalle palvelimelle. Jos WordPress sivuston halutaan käyttävän suoraan palvelimen osoitetta esimerkiksi <http://www.esimerkki.fi/>. Halutessaan voi tehdä myös palvelimen juureen uuden kansion, joka nimetään sen mukaan minkä osoitteen haluaa. Esimerkiksi jos sivuston osoitteeksi halutaan <http://www.esimerkki.fi/sivu> tehdään uusi kansio "sivu" johon WordPress tiedostot siirretään. Viimeisenä asennusvaiheena WordPressin asennustiedosto ajetaan palvelimelta. Asennustiedon ajaminen tapahtuu menemällä sille tarkoitettuun osoitteeseen käyttäen www-selainta. Jos WordPressin tiedostot asennettiin "sivu" nimiseen kansioon, asennustiedoston osoite on <http://www.esimerkki.fi/sivu/wp-admin/install.php>. Onnistuneen asennuksen lopuksi WordPress toivottaa uuden käyttäjän tervetulleeksi. (WordPress Codex 2016b.)

2.2 Ohjausnäkyvä

WordPressin käyttöliittymän tärkein ominaisuus on sen Ohjausnäkyvä (engl. Dashboard), josta nähdään kaikki olennaiset työkalut joiden avulla sivuston sisältöä hallitaan. Sivun yläosassa sijaitsee työkalupalkki, josta nähdään jo paljon tietoa kuten ilmoitukset uusista kommentista ja linkit Ohjausnäkyvään sekä omalle sivustolle. Näkyvillä on myös sisäänkirjautuneen käyttäjän nimi, jota painamalla pääsee suoraan muokkaamaan sen käyttäjän asetuksia ja oikeuksia. Sivun yläosan työkalupalkki on aina näkyvissä kun WordPressiin on kirjaututtu sisään jollain tunnuksella, myös silloin kun poistutaan Ohjausnäkyvästä varsinaiselle sivustolle esimerkiksi tarkastelemaan tehtyjä muutoksia.

Vasemmasta reunasta löytyy varsinainen päävalikko, josta on nähtävillä kaikki sivuston hallintaan ja ylläpitoon liittyvät työkalut. Lisänä jo yläpalkissa näkyviin työkaluihin sivupalkista löytyy sivuston ulkoasun asetukset, lisäosat ja niiden asetukset, käyttäjät sekä asetukset WordPressin Ohjausnäkyvän muokkaamiseen. Ohjausnäkyvän keskiosio on varattu varsinaiselle työskentelyaluelle. Keskiosion sisältö muuttuu sen mukaan, mikä osio on valittuna vasemman laidan työkaluvalikosta. Vasemman laidan päävalikossa näkyy myös saatavilal olevat uudet versiopäivitykset lisäosiin, teemoihin sekä WordPressiin. Päivityksistä ilmoitetaan oranssilla ympyrällä ja numerolla sen valikon otsakkeen kohdalta johon on päivityksiä ja montako niitä on.



KUVIO 1. Ohjausnäköala

2.3 Asetukset

Yksi tärkeimmistä asioista joihin kannattaa tutustua ensimmäisenä WordPressin asennuksen jälkeen on sen asetuksiin, jotka löytyvät päävalikosta kohdasta Asetukset (engl. Settings). Asetukset sivuilta löytyy oletuksena useita alasivuja joita ovat:

- Yleiset (engl. General)
- Kirjoittaminen (engl. Writing)
- Lukeminen (engl. Reading)
- Keskustelu (engl. Discussion)
- Media
- Osoiterakenne (engl. Permalinks).

Yleiset kohdasta nähdään yleisiä tietoja sivustonhallinnasta kuten, sivun nimi ja url-osoite, ylläpitäjän sähköpostiosoite, uuden käyttäjän oletusrooli, ajan ja päivämäärän esitysmuodot sekä sivun kielen valinta. Kirjoittamisen ja lukemisen asetuksista löytyy Postauksien oletuskategorian ja -formaatin muoto, mahdollisen sähköpostin välityspalvelimen asetukset sekä etusivun määrittelyyn liittyvät asetukset. Keskustelun asetuksissa voidaan määrittellä asetukset mahdollisille

keskusteluominaisuuksille, kuten erilaisista ilmoituksista sekä erilaisia kommentoinnin asetuksia. Media osiossa määritellään palvelimelle ladattavien kuvien oletusmitat jotka WordPress muokkaa automaattisesti kuvan lisäyksen yhteydessä. Osoiterakenne asetuksissa määritellään sivuston käyttämien linkkien rakenne sekä voidaan määrittellä sivuille halutunlainen rakenne kategorioille.

2.4 Käyttäjäroolit

Oletuksena WordPressistä löytyy muutamia eritasoisia käyttäjärooleja, joita ovat pääkäyttäjä (engl. administrator), päätoimittaja (engl. editor), kirjoittaja (engl. author), avustaja (engl. contributor) sekä tilaaja (engl. subscriber). Kuten käyttäjäroolien nimistä voi jo ehkä huomata, ovat nekin keskittyneet blogien luomiseen ja ylläpitämiseen. Eri käyttäjärooleilla sivuston pääkäyttäjä voi määrittellä, mitä rekisteröityneet käyttäjät voivat sivuilla tehdä. (WordPress Codex 2016c.) Käyttäjien hallinta tapahtuu Ohjausnäkyvän vasemman reunan päävalikosta valitsemalla Käyttäjät-sivun.

Käyttäjät-sivulta näkyy sivustolle rekisteröityneiden käyttäjien nimimerkki, nimi, sähköpostiosoite, rooli sekä mahdollisten kirjoitettujen artikkelien lukumäärä. Käyttäjiä voidaan hallita myös usean kappaleen ryhmissä, jos halutaan esimerkiksi poistaa useampi käyttäjä kerralla. Uuden käyttäjän voi lisätä valitsemalla päävalikosta Käyttäjät, jonka valitsemalla aukeaa alasvetovalikko, josta valitaan Lisää uusi. Uuden käyttäjän luomiseen vaaditaan vähintään käyttäjänimi, sähköpostiosoite, salasana sekä käyttäjärooli.

Tämän opinnäytetyön tapauksessa tarvittiin vain muutamia eri käyttäjärooleja. Pääkäyttäjä-roolin omaavalla tunnuksella on kaikki oikeudet ja mahdollisuudet sivuston muokkaamiseen. Toisena käyttäjänä toimii päätoimittajan roolin omaava tunnus, jonka oikeuksia on rajoitettu vain sivuston materiaalin ylläpitoon eli tällä tunnuksella ei päästä muokkaamaan sivuston eikä WordPressin Ohjausnäkyvän ulkoasua eikä niihin vaikuttavia lisäosia ja asetuksia. Lisäksi jokaiselle organisaatiolle luotiin vielä vähän enemmän rajoitetut tunnukset turvallisuusjohto-henkilöstön käyttöön. Turvallisuusjohdon tunnuksilla voidaan vain muokata jo olemassa olevia oman organisaation sivuja ja niillä sijaitsevaa tekstiä ja kuvaa. Tällä tunnuksella he myös kirjautuvat sisään tarkastellakseen muilta sivuston käyttäjiltä piilotettua sisältöä.

2.5 WordPressin muokattavuus

WordPress on kehitetty sellaiset käyttäjät huomioon ottaen, jotka haluavat luoda omat sivustot ilman varsinaista tietämystä web-ohjelmoinnista. Jos tietämystä ja osaamista kuitenkin löytyy ohjelmointikielistä kuten HTML, PHP ja CSS, on sivuja mahdollista myös muokata paljon isommalla kädellä. WordPressin ulkoasua voidaan siis muokata helposti joko vain valmiin teeman asennuksella sekä erilaisilla lisäosilla tai tarpeen tullen myös teemalle määriteltyä tyyli-tiedostoa muokkaamalla.

WordPressin teemoihin sisältyy valmis tyyli-tiedosto CSS (lyh. Cascading Style Sheets), joka on www-dokumenteille kehitetty tyyliohjeiden laji jolla määritellään sivustolle sen ulkoasu. Tyyliohjeet yhdessä muodostavat säännöstön, jotka ehdottavat minkä näköisenä sivusto esitetään. Tyyliohjeet määritellään .css päätteiseen tekstitiedostoon, yleensä nimenä käytetään style.css. CSS-sääntö koostuu valitsimesta (engl. selector), attribuutista ja sen arvosta (engl. value). Sivuston teeman CSS-tiedostoa voidaan muokata muun muassa vaihtamalla sivuston eri osissa käytettäviä värejä, kirjainten muotoiluja sekä sivuston eri elementtien kokoja ja paikkoja. (Ring 2013.)

Tämän työn yhteydessä tuli tarpeen käyttää kaikkia kolmea aiemmin mainittua ohjelmointikieltä. Kuitenkin enimmäkseen käytettyinä kielinä olivat CSS sekä HTML. HTML-kieltä tarvittiin vain, jos haluttiin tehdä muokkauksia johonkin tiettyyn sivun kohtaan, ilman että se muokkasi muita samankaltaisia sivuja. PHP-kielen käyttö jäi tällä kertaa melko pieneen roolin, koska se käsittää enemmän WordPressin rakennetta koskevaa koodia. PHP-kieltä tarvittiin kuitenkin sen verran, että sen perusominaisuudet tulivat tutuiksi.

3 SIVUSTON SUUNNITTELU

3.1 Yleistä

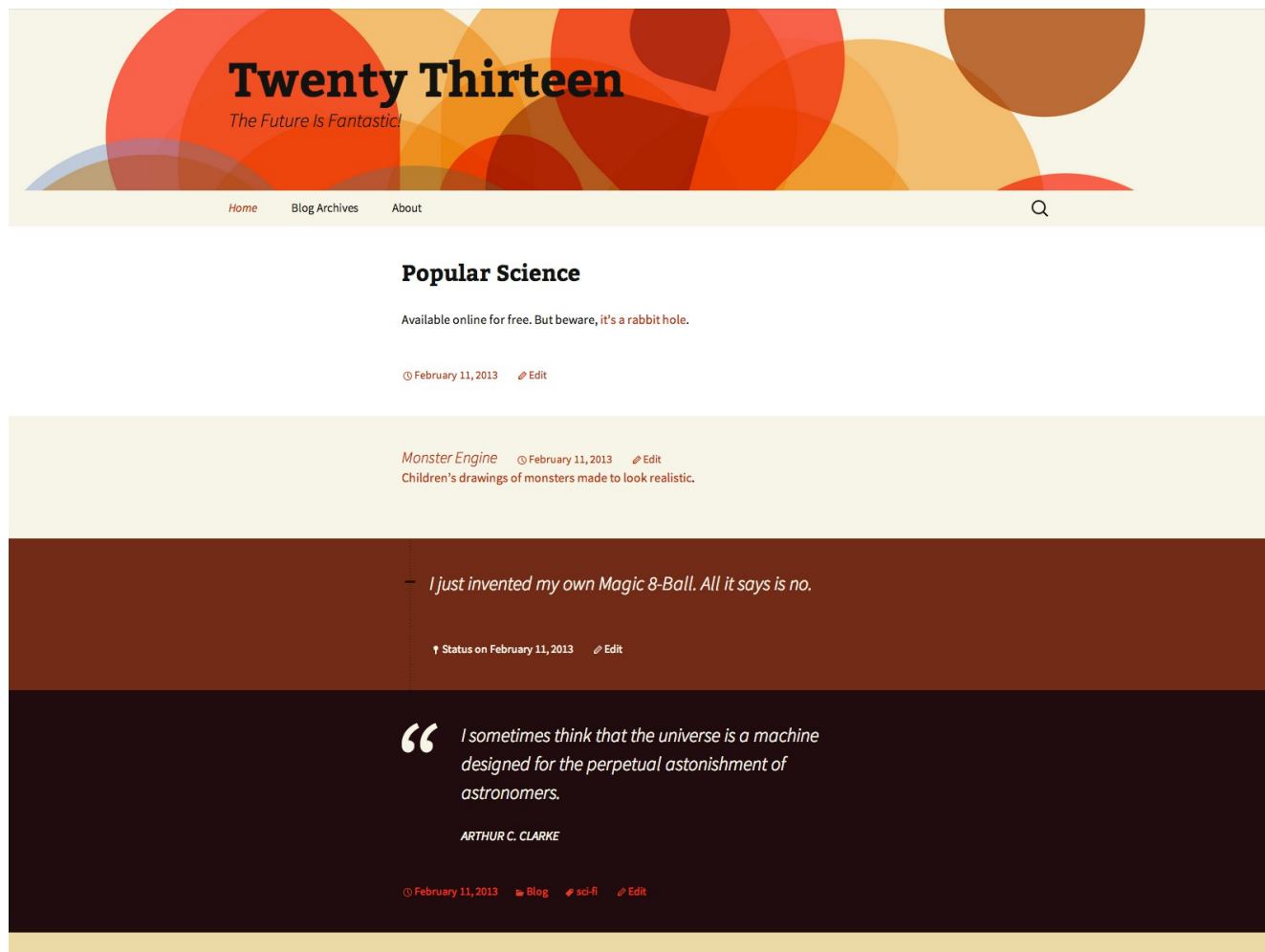
"Kauneus on katsojan silmissä" on vanha sanonta, joka pätee hyvin myös toimivan verkkosivuston suunnitteluun, jossa sivuston toiminnallisuuden arvioivat käyttäjät eivätkä sivuston omistajat. Verkkosivujen käytettävyys koostuu monesta eri muuttujasta, kyse ei ole ainoastaan sivuston ulkoasusta vaan myös millainen se on toiminnaltaan. Hyvä verkkosivun suunnittelu on aina tehty palvelemaan asiakkaan tarpeita, oli se sitten käyttäjän tiedonhakua, viihdettä tai jonkinlaista kanssakäymistä. Jokaisella verkkosivuston sivulla on oltava selvä tehtävä. Internetsivustojen käyttäjät haluavat usein vain nopeasti löytää tarvitsemansa tiedon, joten sivuilla olevan tekstin muotoilu ja rakenne on tärkeää rakentaa helppolukuisiksi. Helppolukuisuutta edesauttaa otsikoiden sekä täydentävien väliotsikoiden käyttö. (Woods 2014.)

Teksien muotoilulla on myös iso vaikutus käyttäjäkokemukseen. Yleisimpiä verkkosivustoilla käytettäviä kirjaisinmuotoiluja ovat sans-serif kirjaisintyyliä kuten Arial ja Verdana, niiden helppolukuisuuden vuoksi. Sans-serif eli groteski-tyyliset fontit ovat pääteviivattomia. Ideaalinen käytettävä kirjaisinmuotoilu on leipätekstissä kooltaan 16px ja muiden tekstien muotoilua olisi syytä vaihtaa korkeintaan kolme kertaa ja kolmen kokoyksikön verran, jotta suunnitelma pysyy virtaviivaisena. Tarkasti mietittyllä väripaletilla sivuston käyttäjäkokemusta voidaan ehostaa entisestään. Toisiaan tukevilla väreillä luodaan tasapainoa ja harmoniaa sivustolle. Lukemista voidaan ehostaa käyttämällä kontrastivärejä tekstin ja taustan muotoiluissa. (Woods 2014.)

Muita tärkeitä muuttujia, jotka kannattaa ottaa huomioon verkkosivuja suunnitellessa, ovat kuvien ja muun sisällön käyttö ja niiden sijoittaminen sivustolle. Oikeanlaisia kuvia, videoita tai muuta grafiikkaa käyttämällä, voidaan usein käyttää jopa tekstiä tehokkaammin jakamaan tietoa. Sisältöä ei kuitenkaan kannata roiskia sivuille miten sattuu, vaan pitää kaikki kuvat ja teksti järjestyksessä esimerkiksi käyttämällä ruudukkomallia, joka jakaa sisällön sarakkeisiin ja kappaleisiin. Myös tyhjän tilan käyttö kannattaa pitää mielessä. Navigointi verkkosivuston eri sivuilla on myös olennainen osa toimivaa suunnittelua. Hyviä tekniikoita ovat muun muassa looginen sivu hierarkia, "bread crumb"-polut ja painettavat napit. Seuraamalla kolmen klikin sääntöä, sivuston rakenne pyritään suunnittelemaan siten, että käyttäjä löytää etsimänsä kolmen klikkauksen sisällä. (Woods 2014.)

3.2 Teema

Kuten aiemmin on käynyt ilmi, WordPress tarjoaa valmiina useita ulkoasuiltaan poikkeavia teemoja jotka ovat suoraan käyttövalmiita sellaisinaan tai käyttäjä voi muokata teemoja omaan käyttöön sopivammaksi. Tässä työssä päädyimme käyttämään valmista teemaa sen sijaan, että teema-tiedosto luotaisiin kokonaan itse. Koska WordPress on ensisijaisesti tehty blogien tuottamiseen, ovat teemat myös enemmän siihen tarkoitukseen sopivampia, poikkeuksia kuitenkin löytyy.



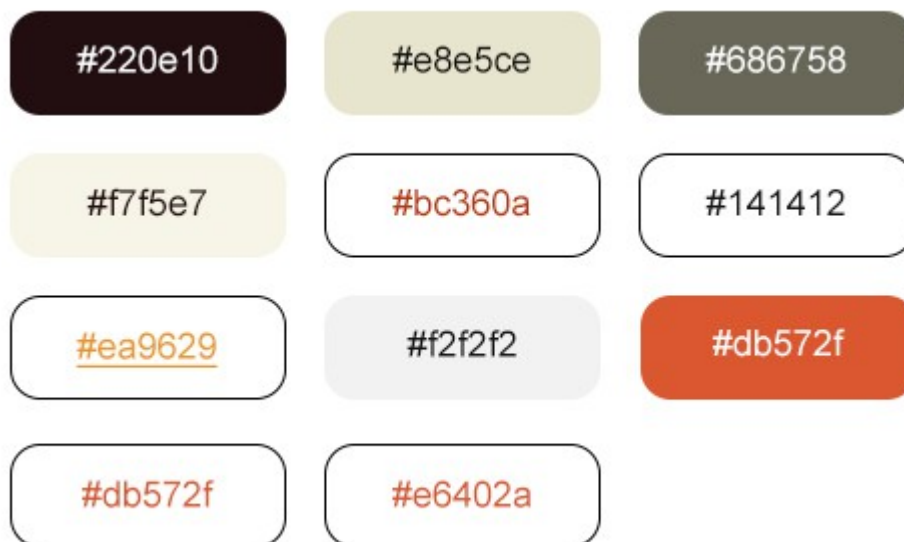
KUVIO 2. Twenty Thirteen teema

Sivustolle halusin valita mahdollisimman selkeän ja yksinkertaisen teeman. Koska sivuston ensisijainen tarkoitus on tiedonjako, oli tärkeää, että tieto on helposti löydettävissä ja luettavissa. Sivustolla käytettäväksi teemaksi valitsin yhden WordPressin omista vakioiteemoista nimeltään Twenty Thirteen, joka tuli tarjolle WordPressiin vuonna 2013 ja on käytössä yli 200 000 käyttäjällä (WordPress 2016c). Kyseinen teema sopi mielestäni hyvin sivuston tarkoitukseen niin ulkoasun kuin käytettävyydenkin osalta. Teeman vakiovärimaailma koostuu erilaisista oranssin ja ruskean sävyistä,

jotka antavatkin sivustolle räikeän ja lämpimän tunnelman. Esittelin valitsemani teeman sekä ideoimani muutokset työn tilaajalle ja hän hyväksyi ne, jonka jälkeen aloitin visuaalisten muutosten toteuttamisen sovitun mukaiseksi.

3.3 Ulkoasu ja rakenne

Kun sivustolla käytettävä teema oli valittu, aloitin sivuston suunnittelun sen tulevaa käyttötarkoitusta varten. Aluksi mietin millä teeman oletusvärimaailmaa saataisiin muokattua ja erotettua hieman valitusta vakioiteemasta. Halusin kuitenkin käyttää teemalle määriteltyjä värejä ja tekstityylejä. Aikani pyöriteltyä vakiovärejä eri sivuston elementeissä, päädyin antamaan sivulle vähän enemmän eloa ja kontrastia vaihtamalla osan sivuston elementeistä haalean beigestä tumman ruskeaan.



KUVIO 3. Sivustolla käytetyt värit

Koska valittu teema on suunniteltu blogien tuottamista silmällä pitäen, on siinä valmiina erilaisia siihen tarkoitukseen kehiteltyjä ominaisuuksia, kuten blogitekstien-hakulaatikko, -kommenttiosio sekä -kategoriointi. Näistä blogi ominaisuuksista haluttiin kuitenkin eroon, koska niillä ei ollut käyttötarkoitusta sivustolla.

Alunperin teeman yläpalkin (engl. header) navigointi-osassa sijaitsi WordPress teemoille tyypilliset valikot sivuston eri alisivuille, jotka sai esiin dropdown-valikon muodossa otsakkeita klikkaamalla. Valikot sekä hakupalkki korvattiin paremmin sivuston käyttötarkoitukseen sopivalla niin kutsutulla

leivänmuru (engl. breadcrumb) polkurakenteella. Tämän ilmaisen lisäosana toteutetun ominaisuuden ansiosta sivustolla navigointi on sujuvampaa. Turhaksi nähtyjä ominaisuuksia poistamalla yläpalkkiin saatiin mahdutettua myös erillinen uloskirjautumispainike, joka on tarpeen tulleen myös mahdollista saada näkymään vain sivustolle kirjautuneille. Uloskirjautumispainike on myös toteutettu ilmaisella lisäosalla nimeltä BAW Login/Logout-menu, joka tarjoaa vaihtoehtoiksi pelkän ulos tai sisäänkirjautumis painikkeen tai molemmat yhdessä jolloin kirjautumattomille käyttäjille näkyy Kirjaudu sisään-teksti ja sisään kirjautuneille Kirjaudu ulos-teksti. Teeman alkuperäisessä muodossa header-kuva sekä navigointi-palkki olivat yksi elementti.

Yleensä verkkosivujen alaosassa sijaitsee footer-palkki, johon voidaan sijoittaa aina näkyvissä olevia osioita, kuten tekstiä, kalenteri, yhteystietoja, sivuston tekijän tietoja. Footer-palkki löytyi myös tämän sivun vakionäkymästä. Alun perin muutin footer-palkin sivulle sopivammaksi poistamalla alkuperäisiä blogitoimintoja ja korvaamalla ne sivuston käyttötarkoitukseen liittyvillä tärkeillä yhteystiedoilla ja hyödyllisillä linkeillä. Myöhemmin kuitenkin totesimme, että footer-palkissa olevat tiedot ovat liian piilossa sivun alareunassa, joten siinä sijainneet tekstit ja linkit siirrettiin paremmin näkyvillä olevaan sivupalkkiin sivuston oikeaan reunaan. Samalla annoin sivupalkille uusia muotoilusääntöjä, kuten pyöristetyt kulmat ja marginaali-alueet.

Valittu Twenty Thirteen teema sopi myös käytettävyydeltään erinomaisesti sivustolle, koska siihen on valmiina määriteltynä myös erilaisia näyttöasetuksia, eli sivun ulkoasu mukautuu sen mukaan millä laitteella sivustoa käytetään. Ulkoasu ei kuitenkaan muutu paljoa normaalista näkymästä, joten sivustoa on helppo käyttää millä tahansa laitteella ja kaikki asiat löytyvät sieltä mistä ne löytyvät alkuperäisestä versiostakin. Tämä oli tärkeää ottaa huomioon sivustoa kehiteltäessä, koska oli tiedossa että suuri osa tulevista sivuston käyttäjistä tulee käyttämään jonkinlaista mobiililaitteita sivuston käyttämiseen. Myös värit sekä tekstien muotoilut olivat määriteltynä valmiiksi teemassa, mutta joitain muutoksia on myös niihin tehty, kuten muutettu tekstin kokoa ja tekstimuotoilua sekä värejä. Myös kokonaan uusia tekstimuotoiluita on luotu. Mobiililaitteiden käyttäjät on huomioitu myös sivuston painettavien laatikoiden ja kuvien mitoissa sekä eri sivuilla olevan tekstinmäärän pituuksissa, tarkoituksena on pitää turha skrollailu minimissä.

4 SIVUSTON TOTEUTUS

Sivuston alkuperäiseen suunnitteluun käytin projektityön tietooni saatua noin viikon, jonka aikana tein Adobe Photoshopilla vedos tasoiset kuvanmuokkaukset. Sivujen varsinainen työstäminen alkoi heti kun sain käyttöön tarvittavat hallinta- ja FTP-tunnukset www-palvelimelle. Sivuston suunnittelu ja toteutus ovat kulkeneet käsi kädessä sen aloittamispäivästä lähtien. Jos mieleen on tullut jokin idea, joka sopisi hyvin sivustolle on mietitty onko se miten hankala toteuttaa tai liikaa resursseja vievä ja onko se loppujen lopuksi olennainen asia sivujen käyttöä ajatellen. Mitään suuria kompromisseja sivuston rakentamisessa ei kuitenkaan ole tehty vaan lähes kaikki suunnitellut ominaisuudet on saatu toteutettua.

CSS- ja PHP-ohjelmointikielet, joita en aikaisemmin ollut juurikaan käyttänyt tulivat hyvinkin tutuiksi tämän työn myötä. Hyväksi tavaksi havaitsin itselleni sen, että kun jotain arvoa muokkaa ylisuurella arvolla, huomaa heti sivustolta mitä se arvo määrittelee, jonka jälkeen sen voi säätää paremmin kohdalleen. Nykyaikaiset internet-selaimet tarjoavat myös mahdollisuuden muokata sivujen ulkoasua inspect-toiminnolla, joka avaa uuteen ikkunaan sivustolla käytetyt visuaaliset muotoilut. Muotoilujen arvoja voidaan siis muokata suoraan selaimessa ilman, että sivuston varsinaista CSS-tyylitiedostoa muokataan. Selaimessa tehdyt muutokset pysyvät voimassa vain sen session ajan eli sivun päivittyessä tai uudelleen aukaisun jälkeen sivu palautuu varsinaisen palvelimella sijaitsevan tyyli-tiedoston määrittelemään muotoon.

4.1 Sivut ja sisältö

Aloitin työstämään sivustua kehittämällä aloitussivun, jossa näkyy kaikki sivustolle tulevat koulut sekä päiväkodit. Alunperin suunnitelmissa oli luoda vain yksi pudotusvalikko, josta koulutusorganisaatio valittaisiin, mutta todettiin että tämä menetelmä voisi olla hieman hankala mobiililaitella käytettäväksi. Lopulta päädyttiin listaamaan koulut eri otsikoiden alle ruudukko-muodossa, josta valitaan haluttu koulu. Näin käytettävyys on helppoa mobiililaitteilla sekä tietokoneella. Valitsemalla koulun pääsee sen koulun yleissivulle, jossa näkyy perusyhteystiedot sekä koulusta riippuen, joitain tärkeitä koulukohtaisia tietoja sekä linkit koulun henkilökunnan ja turvallisuusjohdon sivuille.

Etusivulla, sekä muilla samaa ruudukko-tyylistä otsikointipohjaa käytävillä sivuilla ruudukon luomiseen on käytetty ilmaista lisäosaa nimeltä Responsive Grid Shortcode. Kyseinen lisäosa toimii siten, että se määrittelee käytettävän shortcoden, jonka avulla sivun sisältö voidaan helposti jakaa useisiin sarakkeisiin. Sarakkeiden leveydet määritellään käyttäen eri numeroarvoja. Jos halutaan käyttää koko sivun leveys sisällön näyttämiseen on sarakkeiden numeroarvojen summa oltava 12. Esimerkiksi, jos sivu halutaan jakaa kolmeen saman levyiseen sarakkeeseen, annetaan arvoiksi 4 allaolevan esimerkin mukaisesti.

```
[grid_section]
[grid_col size=4]
    Ensimmäisen sarakkeen sisältö
[/grid_col]
[grid_col size=4]
    Toisen sarakkeen sisältö
[/grid_col]
[grid_col size=4]
    Kolmannen sarakkeen sisältö
[/grid_col]
[/grid_section]
```

Responsive Grid Shortcode lisäosa on nimensä mukaan reagoiva (engl. responsive), eli se mukautuu siirtäen sarakkeet allekkain, kun sivun leveyttä muutetaan esimerkiksi käytettäessä pieni resoluutioisia mobiililaitteita.

Sivustolla voidaan liikkua takaisinpäin joko selaimen takaisin-napilla tai sivuston navigointi-palkista löytyvällä leivänmuru-polulla, joka näyttää etusivusta lähtien kuljetun polun nykyiselle sivulle. Leivänmuru-polku on toteutettu ilmaisella lisäosalla nimeltä Instant Breadcrumbs, joka lisää sivuston yläpalkkiin uudeksi menu-valikoksi.

Yläpalkista poistettu hakupalkki on toteutettu kopioimalla alkuperäisen teeman header.php tiedosto lapsiteeman kansioon ja muokkaamalla sen koodista pois komento, joka hakee hakupalkin. Komennon voi joko poistaa kokonaan tai kuten tässä tapauksessa muokata se kommentiksi alla olevan mukaisesti:

```
<!-- <?php get_search_form(); ?> -->
```

Sivuston sisäisissä sivujen linkityksissä on käytetty ns. slugia. Slug on käyttäjä- ja URL-ystävällinen versio postausten, sivujen, kategorioiden ja tagien nimille. Kun sivulle tai postaukselle annetaan otsikko, WordPress luo automaattisesti sille slugin. Slug on sivun URL-osoitteen se osa, joka viittaa juuri sille sivulle (WordPress Support 2016). Esimerkiksi <http://esimerkki.com/tama-on-sivu/> veisi käyttäjän esimerkki.com sivuston alasivulle nimeltä Tämä on sivu. Slugeja käyttämällä vältetään pitkät ja usein melko sekavat linkit.

Alunperin sivuston suunnitelmana oli toteuttaa kaikki alasivut omina sivuinaan. Tämän todettiin kuitenkin olevan hieman hankalaa ja sekavaa ylläpitää, koska alasivuja kertyi yhdelle koululle yli 10. Uusi useita alasivuja sisältävien sivujen rakenne koostuukin nyt vain yhdestä sivusta johon aiemmat alasivut on listattu otsikoilla. Sivulla näkyvät otsikot on rakennettu siten, että kun sen aihealueen linkistä painaa aiemmalta sivulta, menee sivusto automaattisesti sen otsikon kohdalle. Tämä on toteutettu määrittelemällä pääsivun linkille osoitteeksi alasivun osoite sekä otsikko. Alla oleva esimerkki vie käyttäjän alasivulle kohtaan otsikko1:

```
<a href="alasivu/#otsikko1">Otsikko1</a>
```

Vastaavasti alasivulla otsikko pitää määritellä siten, että aikaisempi linkki tietää mihin sen pitää mennä. Otsikko määritellään jollain tekstimuotoilulla, tässä tapauksessa käytettiin `<h2>` muotoilua. Otsikko määritellään allaolevan esimerkin tavoin:

```
<h2 id="otsikko1">Otsikko1</h2>
```

Tätä sivurakennetta käyttämällä annetaan käyttäjälle myös mahdollisuus halutessaan lukea läpi koko sivu ilman, että hänen tarvitsee mennä takaisin pääsivulle ja valita uusi otsikko nähdäkseen sen sisällön. Sivuston oikeassa reunassa sijaitsevan widget-sivupalkin linkitykset on toteutettu samalla käytännöllä.

Lisäksi sivuston sisällöille on määritelty maksimipituus. Maksimipituuden ylittyessä sisältölaatikon oikeaan reunaan tulee esiin vierityspalkki (engl. scroll-bar), jolla sivun sisältöä voidaan rullata alas- ja ylöspäin, muun sivun pysyessä paikallaan. Tämä on toteutettu lisäämällä allaolevat muotoilut sivuston CSS-tyylitiedostoon:

```
max-height: 820px;
```

```
overflow-y: auto;
```

4.2 Lapsiteema

Vaikka WordPressiin on saatavilla tuhansia käyttövalmiita teemoja, usein tulee kuitenkin tarpeeseen tehdä teemaan omia muokkauksia. Teemoja voi muokata melkein rajattomasti, mutta jos jotain menee muokkauksissa pieleen tai teemaan tulee ladattavaksi päivitys, kaikki teematiedostoon tehdyt muutokset palautuvat sen alkuperäiseen muotoon. Tämä aiheuttaisi turhaa harmia ja hukkaa aikaa. Näistä syistä johtuen tässäkin projektissa päätettiin käyttää lapsiteemaa (engl. child theme) ennen kuin sivujen varsinaista teknistä toteutusta päästiin aloittamaan. Lapsiteema koostuu yhdestä kansioista oman WordPress-sivuston palvelimella joka löytyy polusta /wp-content/themes. Tämän uuden kansion sisältä tulee löytymään vähintään kaksi tiedostoa, jotka ovat style.css ja functions.php. (WordPress Codex 2016d.) Uusi lapsiteeman kansio on hyvä nimetä siten, että sen tunnistaa mitä teemaa se käyttää. Esimerkiksi tämän projektin tapauksessa käytössä on Twenty Thirteen -teema nimettiin lapsiteeman kansio muotoon twentythirteen-child.

Seuraavassa vaiheessa luodaan lapsiteeman käyttämät tiedostot. Aluksi tehdään tekstieditorilla style.css-tyylitiedosto, jonka ensimmäiset rivit näyttävät seuraavanlaisilta:

```
/*
Theme name:      (teeman nimi)
Theme URI:       (teeman www-osoite)
Description:     (teeman kuvaus)
Author:         (tekijän nimi)
Author URI:     (tekijän www-osoite)
Template:       (alkuperäisteeman kansio)
Version:       (teeman versionumero)
License:       (lisenssinumero)
License URI:   (lisenssin www-osoite)
Tags:         (avainsanat)
*/
```

Lapsiteeman tyylitiedostosta pitää löytyä tiedot vähintään kohdista Theme name ja Template.

Kun yllämainitut tiedot on määritelty teematiedostoon, tulee seuraavaksi määritellä mistä ja miten alkuperäisen teeman sisältö määritellään lapsiteeman käyttöön. Aikaisemmin tähän on käytetty `@import` komentoa, joka lataisi alkuperäisen teeman tiedot lapsiteeman vastaavaan, mutta tämä tapa on todettu hitaaksi. Nykyään teemat linkitetään toisiinsa muokkaamalla `functions.php`-tiedostoa, joka luodaan lapsiteeman kansioon. Tätä tekniikkaa käyttäessä `functions.php`-tiedoston alkuun lisätään `wp_enqueue_scripts` action niminen koukku (engl. hook), joka hakee tiedot alkuperäisen teeman tiedoista. Koodin lisäyksen jälkeen `functions.php`-tiedoston alku näyttää seuraavanlaiselta:

```
<?php
function twentythirteenchild_enqueue_styles() {
    $parent_style = 'twentythirteen';
    wp_enqueue_style( $parent_style, get_template_directory_uri() . '/style.css' );
    wp_enqueue_style( 'twentythirteen-child',
        get_stylesheet_directory_uri() . '/style.css',
        array( $parent_style )
    );
}
add_action( 'wp_enqueue_scripts', 'twentythirteenchild_enqueue_styles' );
?>
```

Lapsiteema on nyt valmis aktivointia varten. Aktivointi tapahtuu kirjautumalla WordPressin hallintisivulle admin-tunnuksilla ja menemällä vasemman päävalikon kautta Ulkoasu > Teemat. Seuraavassa näkymässä näkyy WordPressin vakioiteemat sekä äsken luotu lapsiteema. Valitaan lapsiteema ja aktivoidaan se. Nyt lapsiteema on valmis käytettäväksi. (WordPress Codex 2016d.)

Lapsiteemaa käyttämällä on mahdollista muokata myös muita alkuperäiseen teemaan kuuluvia tiedostoja `style.css` tiedoston lisäksi. Tämä tapahtuu yksinkertaisesti kopioimalla haluttu tiedosto alkuperäisen teeman kansioista lapsiteeman vastaavaan kansioon.

4.3 Lisäosat

Sivuston ehostukseen käytettiin useita ilmaisia lisäosia (engl. plugin) joita on tarjolla WordPressin kotisivuilla yli 46 tuhatta (WordPress 2016d). Yhteensä tässä projektissa käytettyjä lisäosia on 12. Vaikka WordPress onkin hyvin kattava ja tehokas julkaisualusta, siitä kuitenkin puuttuu oletuksena joitain ominaisuuksia, joita tässä projektissa tarvittiin. Koska WordPressiä ei ole oikeastaan tarkoitettu

käsittämään sivustoja jossa on paljon erillisiä sivuja, sen sivujenhallinta voi alkuperäisessä muodossaan olla hieman hankala ja turhauttava käyttää. Onneksi useimmat näistä hieman puutteellisista vakio-ominaisuuksista saatiin kuitenkin korjattua ilmaisia lisäosia käyttäen.

WordPressin lisäosat asennetaan valitsemalla Ohjausnäkyvän vasemman reunan päävalikosta Lisäosat, jonka jälkeen lisäosia voidaan selata kategorioittain tai hakusanoilla. Kun oikeanlainen lisäosa on löytynyt voi sen sivuilta vielä lukea tietoja lisäosasta sekä mahdolliset tarkemmat asennus- ja käyttöohjeet. Seuraavaksi valitaan Asenna nyt, jonka jälkeen vielä tarvitaan palvelimen FTP-tunnukset jotta lisäosa voidaan ladata. Lopuksi kun lisäosan lataus ja asennus on valmis WordPress pyytää käyttäjää aktivoimaan lisäosan, jotta se tulee käyttöön sivustolla. Kaikki asennetut lisäosat näkyvät Lisäosat sivulla, josta niitä voi ottaa pois käytöstä (engl. deactivate), poistaa kokonaan sekä päivittää. Lisäosat asentuvat WordPress sivuston käyttämälle palvelimelle.

Lähdin parantamaan WordPressin sivujenhallinnan käytettävyyttä asentamalla lisäosan nimeltään Swifty Page Manager, jonka avulla luodut sivut saadaan organisoitua paremmin. Tämän lisäosan avulla sivut voidaan sijoittaa sivujen muokkausnäkyvässä uudelleen paremmin sivuston rakennetta kuvaavaan muotoon. Sivujen uudelleen sijoittelu tapahtuu helposti hiirellä raahamalla sivut oikeille paikoilleen. Usein sivuilla on myös alasivuja, kuten tässäkin tapauksessa. Swifty Page Manager lisäosan avulla alasivut sijoittuvat valitun parent-sivun alle järjestykseen pienellä sisennyksellä, erottaen pääsivun alasivuista paremmin. Lisäksi sivujenhallintänäkyvässä alasivuja sisältävien sivujen viereen ilmestyy (+) -tai (-) -merkki jota painamalla alasivut voi piilottaa näkyvästä.

Toinen sivujen näkymisrakenteen muokkaukseen vaikuttava lisäosa, jonka asensin on nimeltään Duplicate post. Tämän lisäosan avulla pääsivu alasivuineen voidaan kloonata, jolloin sivuja ei tarvitse tehdä manuaalisesti yksi kerrallaan. Tämän lisäosan käyttö nopeutti huomattavasti sivuston rakentamista. Kun olin tehnyt yhden koulun yleiset sivut alasivuineen valmiiksi pohjaksi, pystyin tekemään niistä kopiot. Tämän jälkeen tein vielä organisaatiokohtaiset muutokset, kuten sivujen linkitykset sekä nimet, jonka jälkeen vielä raahasin kloonatun organisaation oikealle paikalleen sivunäkyvässä.

Koska osa sivuston sisällöstä haluttiin rajata vain tietyille käyttäjille, täytyi sivujen salaukseen kehitellä jokin hyvä keino. Oletuksena WordPressin sivujen salasanasuojaus toimii siten, että kun käyttäjä kirjoittaa salasanan suojatulle sivulle päästäkseen, salasana suojaa pelkästään kyseistä sivua eikä sen alasivuja. Tämä aiheutti ongelman, jossa käyttäjä pystyi ohittamaan salatun sivun

kirjoittamalla selaimen osoiteriville sen alisivun osoitteen jonka halusi nähdä. Esimerkiksi jos sivu2 on salasanalla suojattu ja sillä on alisivu sivu3, jos käyttäjä kirjoittaa selaimen osoiteriville <http://esimerkki.fi/sivu/sivu3/> hän näkisi sivu3 alisivun ilman salasanaa. Tämän ongelman korjaamiseksi asensin ilmaisen lisäosan nimeltä FT Password Protect Children Pages, joka laajentaa pääsivulle asetetun salasanan myös sen sivun alisivuille.

4.4 Käyttäjät

Sivuston sisältö on rajattu kolmeen eri käyttäjäryhmään, joita ovat: yleiset käyttäjät, henkilökunta sekä turvallisuusjohto. Yleisille käyttäjille eli kaikille sivustolla vieraileville, näkyvillä ovat aloitussivu sekä jokaisen koulun yleiset sivut. Koulujen omilla yleisillä-sivuilla on linkit henkilökunnan sekä turvallisuusjohdon sivuille. Jos käyttäjä haluaa tarkastella henkilökunnalle tarkoitettuja sivuja, on hänen tiedettävä salasana, joka on jokaiselle koululle oma. Varsinaisia käyttäjätunnuksia ei siis tarvita. Kun sivuston käyttäjä etenee minkä tahansa organisaation Henkilökunta-sivuille tulee näkyville ilmoitus, jossa pyydetään antamaan salasana jotta sivujen sisältö on nähtävissä. Tämä ominaisuus on toteutettu lisäämällä allaoleva koodi lapsiteeman käyttämään funtions.php tiedostoon.

```
function my_password_form() {
    global $post;
    $label = 'pwbox-' . ( empty( $post->ID ) ? rand() : $post->ID );
    $o = '<form action="' . esc_url( site_url( 'wp-login.php?action=postpass',
'login_post' ) ) . '" method="post">
' . __( "Tämän sivun sisältö on suojattu. Nähdäksesi sivun anna salasana." ) .
'<br><br>
    <label for="' . $label . '">' . __( "Salasana:" ) . ' </label><input
name="post_password" id="' . $label . '" type="password" size="20"
maxlength="20" /><input type="submit" name="OK" value="' . esc_attr__( "OK" ) .
' " />
    </form>
';
    return $o;
}
add_filter( 'the_password_form', 'my_password_form' );
```

Jos taas käyttäjä haluaa päästä tarkastelemaan koulun turvallisuusjohdon sivuja, tarvitaan käyttäjätunnus ja salasana. Näin haluttiin painottaa eri sivustojen vakavuutta sekä turvallisuutta.

Kirjautumislomakkeen näyttämiseksi luotiin shortcode [login-form], joka näkyy vain käyttäjille, jotka eivät ole kirjautuneet sisään. Shortcode sijoitetaan sivulle jossa kirjautumislomake halutaan näyttää. Shortcoden toiminta on määritelty lapsiteeman functions.php tiedostoon seuraavalla tavalla:

```
function login_form_shortcode() {
    if ( is_user_logged_in() )
        return '';
    return wp_login_form( array( 'echo' => false ) );
}
function devpress_add_shortcodes() {
    add_shortcode( 'login-form', 'login_form_shortcode' );
}
add_action( 'init', 'add_shortcodes' );
```

Henkilökunta sekä turvallisuusjohto-sivujen aloitusnäkyvä on aloitussivun tyyliä mukaileva, eli näkyvillä on kuusi tai useampi linkki jota klikkaamalla päästään uudelle alisivulle.

Käyttäjän kirjoittaessa salasanan tai tunnuksen ja salasanan sivuston yläpalkkiin ilmestyy Kirjaudu ulos-painike, jota painamalla käyttäjä kirjataan ulos sivustolta ja ohjataan etusivulle. Tämä on toteutettu lisäosalla nimeltä Nav Menu Roles jolla voidaan piilottaa menu-elementtejä käyttäjän perusteella. Uloskirjautuessa etusivulle ohjaus on toteutettu lisäämällä functions.php tiedostoon kohta:

```
add_action('wp_logout',create_function('', 'wp_redirect(home_url());exit();'));
```

Jos käyttäjä unohtaa kirjautua ulos manuaalisesti on sivustolle määritelty ajastettu uloskirjautuminen. Näin varmistetaan, ettei rajattujen sivujen sisältöä pääse kukaan ulkopuolinen katselemaan. Kirjautumistiedot poistetaan tässä tapauksessa kun sivusto on ollut käyttämättömänä 30 minuuttia. Ajastettu kirjautumistietojen poisto näyttää functions.php tiedostossa seuraavanlaiselta:

```
function myplugin_cookie_expiration( $expiration, $user_id, $remember ) {
    return $remember ? $expiration : 1800;
}
add_filter( 'auth_cookie_expiration', 'myplugin_cookie_expiration', 99, 3 );
```

Käyttäjätunnuksen ja salasanan sekä pelkän salasanan vaatien sivujen kirjautumistiedot poistetaan cookiasta käyttäjän kirjautuessa ulos sivustolta alla olevan koodin mukaisesti:


```

add_action( 'clear_auth_cookie', function()
{
    setcookie(
        'wp-postpass_' . COOKIEHASH,
        '',
        time() - YEAR_IN_SECONDS,
        COOKIEPATH,
        COOKIE_DOMAIN
    );
});

```

Kun sivuston käyttäjä etenee minkä tahansa organisaation Henkilökunta-sivuille tulee näkyville ilmoitus jossa pyydetään antamaan salasana, jotta sivujen sisältö on nähtävissä. Tämä ominaisuus on toteutettu lisäämällä allaoleva koodi lapsiteeman käyttämään funtions.php tiedostoon.

```

function my_password_form() {
    global $post;
    $label = 'pwbox-' . ( empty( $post->ID ) ? rand() : $post->ID );
    $o = '<form action="' . esc_url( site_url( 'wp-login.php?action=postpass',
'login_post' ) ) . '" method="post">
    ' . __( "Tämän sivun sisältö on suojattu. Nähdäksesi sivun anna salasana." ) .
'<br><br>
        <label for="' . $label . '">' . __( "Salasana:" ) . ' </label><input
name="post_password" id="' . $label . '" type="password" size="20"
maxlength="20" /><input type="submit" name="OK" value="' . esc_attr__( "OK" ) .
' " />
    </form>
    ';
    return $o;
}
add_filter( 'the_password_form', 'my_password_form' );

```

5 YHTEENVETO

Opinnäytetyön aiheena ja tavoitteena oli suunnitella ja toteuttaa toimiva, selkeä ja helppokäyttöinen tiedonjakoon perustuva sivusto Ylivieskan alueen opetusorganisaatioille. Sivusto toteutettiin WordPress-sisällönhallintajärjestelmällä ja palvelimena käytettiin oman koulun palvelinta.

Työn alkumetreillä luotu suunnitelma sivuston ulkoasua ja toimintaa koskien pysyi pääpiirteiltään suunnitelman mukaisena. Työn edetessä ja teknisen sekä teorillisen tietämyksen lisääntyessä sivuston ilme sekä toiminta kuitenkin muuttui vielä hieman käyttäjäystävällisempään muotoon. Sivustolla oleva teksti- ja kuvamateriaali saatiin kouluilta sitä tahtia miten heillä oli aikaa sitä luoda ja lähettää.

Aikataulu työn valmistumisen suhteen muuttui useaan kertaan, erilaisista seikoista johtuen. Sivuston ensimmäinen versio rakenteesta ja materiaalista oli luotu jo melko nopeasti työn aloituksesta. Palvelin, jolla sivusto sijaitti koki yllättävän ongelman sen päivityksen yhteydessä, jolloin sivuston käyttämä tietokanta meni sekaisin. Tästä syystä tietokannassa sijaitseva tieto eli kaikki sivuston kirjoitettu materiaali muuttui käyttökelvottomaksi ja ne piti luoda uudelleen sivustolle. Ulkoasu ja siihen liittyneet lisäosat pysyivät kuitenkin ennallaan, näitä kuitenkin muutettiin samalla. Toinen aikataulu venyttänyt asia oli koulujen hieman hidas materiaalien toimitus, johtuen heidän kiireellisistä aikatauluista, lomista sekä materiaalien laajuudesta.

Kaiken kaikkiaan opinnäytetyö sujui ilman mitään ylitsepääsemättömiä ongelmia. Omasta sekä työn tilaajan mielestä työlle asetetut tavoitteet saavutettiin, jättäen kuitenkin vielä varaa kehittää sivustoa eteenpäin. Opinnäytetyöni oli erittäin mielenkiintoinen ja opettavainen verkkosivujen toiminallisen sekä ulkonäöllisen kehittämisen kannalta. WordPress-sisällönhallintajärjestelmä sekä siihen liittyvät ohjelmointikielet tulivat jo tämän työn aikana hyvin tutuiksi, jättäen kuitenkin vielä varaa uuden oppimiseen.

LÄHTEET

Ring, Z. 2013. An Intro to CSS, or How to Make Things Look Like You Like. Saatavissa: <https://dailypost.wordpress.com/2013/06/21/css-intro/>. Viitattu 10.10.2016.

Woods, S. 2014. 10 Top Pinciples of Effective Web Design. Www-dokumentti. Saatavissa: <http://shortiedesigns.com/2014/03/10-top-principles-effective-web-design/>. Viitattu 24.9.2016.

WordPress. 2016a. Requirements. Www-dokumentti. Saatavissa: <http://wordpress.org/about/requirements/>. Viitattu 22.9.2016.

WordPress. 2016b. About WordPress. Www-dokumentti. Saatavissa: <http://wordpress.org/about/>. Viitattu 22.9.2016.

WordPress. 2016c. Theme Directory. Www-dokumentti. Saatavissa: <https://wordpress.org/themes/twentythirteen/>. Viitattu 24.9.2016.

WordPress. 2016d. Plugin Directory. Www-dokumentti. Saatavissa: <https://wordpress.org/plugins/>. Viitattu 25.9.2016.

WordPress Codex. 2016a. History. Www-dokumentti. Saatavissa: <https://codex.wordpress.org/History>. Viitattu 22.9.2016.

WordPress Codex. 2016b. Installing WordPress. Www-dokumentti. Saatavissa: https://codex.wordpress.org/Installing_WordPress/. Viitattu 22.9.2016.

WordPress Codex. 2016c. Roles and Capabilities. Www-dokumentti. Saatavissa: http://codex.wordpress.org/Roles_and_Capabilities. Viitattu 22.9.2016.

WordPress Codex. 2016d. Child Themes. Www-dokumentti. Saatavissa: https://codex.wordpress.org/Child_Themes. Viitattu 22.9.2016.

WordPress Support. 2016. Slugs. Www-dokumentti. Saatavissa: <https://en.support.wordpress.com/slugs/>. Viitattu 5.11.2016.