

Pekka Suni

WordPressin soveltuvuus mobiililaitteille

Opinnäytetyö

CENTRIA AMMATTIKORKEAKOULU

Sähkötekniikan koulutusohjelma

Toukokuu 2015

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Yksikkö Ylivieska	Aika Toukokuu 2015	Tekijä/tekijät Pekka Suni
Koulutusohjelma Mediatekniikka		
Työn nimi WordPressin soveltuvuus mobiililaitteille		
Työn ohjaaja Hannu Puomio		Sivumäärä 13
Työelämäohjaaja Matti Mehtälä		
<p>Opinnäytetyöni aiheena oli tutkia WordPressin soveltuvuutta mobiililaitteille. Opinnäytetyössä ohjelmin Wordpressiä käyttämällä fitness-aiheisen sivuston, jonka avulla selvitin miltä WordPressillä luodut web-sivut näyttävät mobiililaitteiden näytöllä ja kuinka hyvin ne toimivat mobiililaitetta käytettäessä. Tämä aihe oli ajankohtainen mobiililaitteiden laajan suosion vuoksi.</p> <p>Työ toteutettiin käyttämällä WordPressin versiota 4.1.0 ja WinSCP-ohjelmaa, sekä PHP-ohjelmointikieltä, JavaScript-ohjelmointikieliä ja CSS-tyylimäärittelyä.</p> <p>Opinnäytetyön tuloksena syntyi selkeä ja toimiva web-sivusto.</p>		

Asiasanat

Lihaskunnon kehityspalvelu, WordPress

ABSTRACT

Unit Ylivieska	Date May 2015	Author Pekka Suni
Degree programme Mediatechnology		
Name of thesis The suitability of WordPress for mobile devices		
Instructor Hannu Puomio		Pages 13
Supervisor Matti Mehtälä		
<p>The subject of this thesis was to investigate the suitability of WordPress to mobile devices. In the thesis a fitness related web was programmed using WordPress to find out how WordPress-based web-sites look like on mobile devices. The topic was timely due to the wide popularity of mobile devices.</p> <p>The work was implemented using WordPress version 4.1.0 and WinSCP program, as well as the PHP programming language, JavaScript programming language and cascading style definitions.</p> <p>As a result of the thesis, a clear and functional web-site was created.</p>		

Key words Muscle strength development service, WordPress
--

**TIIVISTELMÄ
ABSTRACT
SISÄLLYS**

1	JOHDANTO	1
2	OHJELMOINTIKIELET	2
	2.1 HTML	2
	2.2 PHP	3
	2.3 CSS	4
	2.4 JAVASCRIPT	4
3	WORDPRESS JULKAISUALUSTANA	6
	3.1 Teemat	6
	3.2 Lapsiteemat	7
	3.3 Lisäosat	8
	3.4 Wordpress mobiililaitteissa	9
4	SIXPACK	10
	LÄHTEET	12

1 JOHDANTO

Opinnäytetyöni aiheena oli tutkia WordPressin soveltuvuutta mobiililaitteille. Tarkemmin sanottuna tutkimuksen kohteena oli selvittää kuinka hyvin WordPressillä luotujen sivujen käyttäminen onnistuu mobiililaitteilla, muuttuuko sivuston rakenne tai ulkonäkö ja kuinka sivujen päivittäminen tai muokkaaminen onnistuu mobiililaitteilla. Opinnäytetyöni aihe on tullut ajankohtaiseksi viimevuosina lisääntyvien älylaitteiden ja niiden kehityksen vuoksi. Käytössäni oli WordPressin versio 4.1.0. Valitsin tämän aiheen lehtori Hannu Puomion ehdotuksesta. Aihe oli jokseenkin entuudestaan tuttu edellisen vuoden kesäharjoittelusta.

Tässä tutkimuksessa oli suurta apua opiskeluaikana käydyistä Centrian kursseista muun muassa WordPressin yleiseen käyttöön liittyen, ohjelmointikielen sekä myös Word-tekstinkäsittelyohjelman käytöstä.

Työssä käytettiin lähteinä ohjelmointikieliä ja WordPressin käyttöä käsitteleviä verkkojulkaisuja.

2 OHJELMOINTIKIELET

WordPress, kuten myös kaikki muutkin käyttämämme tietokonesovellukset on ohjelmoitu jollakin ohjelmointikielellä. Ohjelmointikieliä on satoja erilaisia ja niitä voi käyttää erilaisiin käyttötarkoituksiin tehtävän laadun mukaan. Jotkut kielet on luotu opiskelijoille, jotta he oppisivat ohjelmoinnin alkeet helpommin, kun taas toisia käyttävät ammattilaiset luoden ohjelmistoja meille muille. (Ohjelmointi 2015.)

Ohjelmoijan täytyy tehdä hyvin täsmälliset toimintaohjeet tietokoneelle, sillä tietokone ei osaa arvailla mitä sovelluksen seuraavaksi tulisi tehdä, joten tämän perusteella voidaankin todeta, että tietokone on juuri niin viisas, kuin ohjelman tekijä on. (Ohjelmointi 2015.)

2.1 HTML

WordPress pystyy lukemaan HTML:ää ilman lisäosia ja jopa sivuille ja artikkeleille voi kirjoittaa edellä mainittua koodauskieltä. HTML sisältää monenlaisia elementtejä, joiden erottelu onnistuu tagien avulla. Lähes aina tageja käytetään pareittain. Ensin laitetaan alkutagi `<>`, jonka jälkeen tulee www-sivulla näkyvä teksti tai jokin muu olio ja lopuksi lopputagi `</>`. Tagien, niin alku- kuin lopputagien sisälle tulee sama elementin nimi, joka viestittää selaimelle miten tagien väliin sijoitettu teksti tai muu olio esitetään. (Kerkkänen 2009.)

HTML (Hypertext Markup Language) kehitys alkoi vuonna 1989 Tim Berners-Leen johdolla tiedeyhteisöjen tarpeisiin ja se toimi halpana ja nopeana jakeluväylänä uusimmille tutkimustuloksille. Internetin alkuaikoina web-sivut olivat hyvin yksinkertaisia ja sisälsivät lähinnä pelkkää tekstiä. Tärkeimpinä prioriteetteina olivatkin selkeys ja hyvä luettavuus. Netscape-selain julkaistiin vuonna 1994 ja se lisäsi huomattavasti web-sivujen visuaalisuutta. Microsoft kehitti Netscapen kilpailijaksi Internet Explorerin josta seurasi, että molemmat valmistajat alkoivat kehittää HTML-kieltä oman selaimen tarpeisiin.

Vasta myöhemmin organisaatio nimeltä W3 määritteli HTML-standardin. (W3C 2015.)

2.2 PHP

PHP on lyhenne sanoista Hypertext Preprocessor. Se on avoimen lähdekoodin ohjelmisto, jonka kehittämisen aloitti Rasmus Lerforf vuonna 1994 kirjoittamalla pienen kokoelman C-kielisiä CGI-skriptejä. PHP:tä on kehitetty alusta alkaen varta vasten web-ympäristön sovelluksia varten ja se voidaankin sijoittaa suoraan sivun HTML-koodin sekaan, eikä erillistä kehitysympäristöä välttämättä tarvita, joten PHP-koodia voidaankin kirjoittaa vaikkapa Windowsin Notepad-ohjelmalla. Koodi käsitellään palvelimella sivun pyynnön yhteydessä. (Ohjelmointi 2015.)

Yleisimmin PHP:tä käytetään erilaisten web-sovellusten ohjelmointiin. Sen avulla voidaan lisätä sivustoille erilaisia toimintoja, kuten lomaketietojen tarkastusta ja lähetystä, sisällön lataamista tietokannoista jne. (Ohjelmointi 2015.)

Upottaminen HTML:n sekaan onnistuu lisäämällä PHP- koodi `<?php ?>`- elementin väliin. Sivun tiedostopäätteeksi tulee `.php`. Myös merkkejä `<? ?>`, `<% %>` (ns. ASP-tagit) ja `<script language="php"> </script>` voidaan käyttää riippuen palvelimen asetuksista. Rivien lopetus tapahtuu aina puolipisteellä (;), paitsi viimeinen rivi, jolle se ei ole pakollinen. (Keski-Suomen Verkkomedia 2009.)

2.3 CSS

CSS on lyhenne sanoista Cascading Style Sheets, joka voi tarkoittaa tyyლისivuja, tyylipohjaa, muotoilumallia tai muotoilusääntöä. CSS:llä siis määritellään sivun ulkoasu. Sillä voidaan määritellä muun muassa web-sivun taustakuva ja -väri, reunukset, marginaalit sekä fontin koko ja väri. Esimerkiksi jos HTML:ssä on merkintä `<h1>Otsikko</h1>`, CSS-määrittely `H1 { color: #FFFFFF }` määrittelee otsikon väriksi valkoisen. (Heikeniemi 2002.)

Tavoitteena CSS:llä on erottaa ulkoasu ja sisältö toisistaan. Se pyrkii luomaan yksinkertaista koodia, jonka päivittäminen on helppoa. Se myös mahdollistaa monipuolisempia määrittelyjä kuin HTML ja saman tyylipohjan käyttö on mahdollista erityyppisissä asiakirjoissa (HTML, XHTML, XML). (Perälä 2003.)

CSS- tekniikan kehitti organisaatio nimeltä W3C. Sen ensimmäinen versio CSS1 julkaistiin vuonna 1996. Päivitetty ja W3C:n suosittelema versio julkaistiin vuonna 1998 ja se kulki nimellä CSS2. Nyt käytössä on vuonna 2003 julkaistu versio CSS2.1, joka on hieman korjailtu ja supistettu versio edellisestä. (Rossi 2010.)

2.4 JAVASCRIPT

JavaScript on monipuolinen, tulkettava ohjelmointikieli, jota yleisimmin käytetään selainskriptien ohjelmointiin (client-side scripting), toiminnallisuuden ohjelmointiin ja käyttöliittymän ohjelmointiin. Vaikka siitä luotiin hyvin samankaltainen C-kielen ja Javan kanssa, se on kuitenkin täysin omanlaisensa kieli. JavaScriptin yhteydessä puhutaan useimmiten skripteistä (script) eikä ohjelmista (program), sillä JavaScript-kielen toteutus perustuu tulkintaan (interpretation) eikä kääntämiseen (compilation). (Korpela 2009.)

JavaScriptin vahvuus on sivun muokattavuus W3C:n määrittelemän DOM-olionmallin kautta ja sillä on vaivatonta tehdä yksinkertaisia skriptejä, sekä kokeilla niiden toimivuutta. (Heikeniemi 2002.)

JavaScriptin lisääminen HTML-dokumenttiin onnistuu monilla eri tavoilla. Yksi tapa on kirjoittaa JavaScript-koodi erilliseen tiedostoon ja tallentaa se .js-muotoon. Tiedosto sijoitetaan hakemistoon siten, että siihen pystytään viittaamaan URL:llä. JavaScript-koodin mukaan ottaminen HTML-koodiin onnistuu seuraavasti: `<script type="text/javascript" language="JavaScript" src="URL"></script>`.

Toinen menetelmä on samankaltainen menetelmä, kuin edellinen: JavaScript-koodi on ikään kuin upotettu suoraan HTML-koodin sekaan. `<script type="text/javascript" language="JavaScript"><!--JavaScript-koodi tulee tänne!--></script>`.

Kolmannella ja viimeisellä tavalla HTML-kielessä esiintyvillä niin sanotuilla tapahtumamääritteillä (event attributes) pystyy lisäämään yksittäisiin elementteihin liittyviin tapahtumiin JavaScript-koodia. (Korpela 2009.)

JavaScriptiä sai alkunsa 90-luvun alussa Netscapella työskennelleen Brendan Eichin toimesta. Idean ohjelmointikielen antoivat web-sivujen kehittäjät, jotka eivät olleet tyytyväisiä HTML-kielen hitaaseen ja työlääseen päivitettävyyteen. Ensimmäinen versio otettiin käyttöön jo vuonna 1995 Netscape Navigator 2.0-selaimessa, jolloin selaimen ominaisuuksien muuttaminen sekä yksinkertaisten olio-ohjelmien luonti tuli mahdolliseksi. Jälkeenpäin tuki erilaisille lauseille (kuten switch/case), taulukoille, kuville sekä Java-sovelluksille lisättiin mukaan. Nykysisin käytössä on vuonna 1999 julkaistu versio 1.5 JavaScriptistä. (Peltomäki 1998, 130.)

Alussa kehitettiin CGI-ohjelmointikieli ja tätä käyttämällä sivujen generointi onnistui dynaamisesti palvelimella. Tämä mahdollisti käyttäjän antamien lomaketietojen ja tietokantojen käytön sivuja luodessa. Se oli varsin hidasta, joten HTML-sivuille lisättävää ja vuorovaikutteisuutta parantavaa komentosarjakieltä (scripting language), JavaScriptiä, alettiin kehittää. (Peltomäki 1998, 129.)

3 WORDPRESS JULKAISUALUSTANA

Alun perin WordPress-julkaisualusta luotiin pelkästään monipuoliseksi blogialustaksi, mutta sittemmin siitä on kehittynyt julkaisualusta kaikenlaisille sivustoille. Koska WordPress on erittäin hyvin muokattavissa oleva julkaisualusta, ei rajoituksia sivuston tyypille ole, ja se onkin osasyynä julkaisualustan suurelle suosiolle; WordPress.com-sivuston mukaan sitä käyttää jopa yli 70 miljoonaa sivustoa ja se on käännetty yli 70 kielelle, mukaan lukien suomeksi. WordPressin suuresta suosiosta kertoo myös W3Techin tekemä tutkimus, jonka mukaan WordPress-julkaisualustaa käyttää 23,7 % kaikista web-sivustoista. Julkaisualustoista kakkossijalla on Joomla (2,8 %) ja kolmantena Drupal (2,0 %) (Q-Success 2015.)

WordPressissä kaikki sivuston rakenteeseen, sisältöön ja ulkoasuun liittyvä muokkaaminen onnistuu erittäin monipuolisen hallintasivun kautta. Sen kautta voi muun muassa lisätä mediaa (kuvat, videot), hallita kommenttiosioita (muokkaus, poistaminen), muokata sivuston ulkonäköä (muun muassa teeman vaihtaminen, valikoiden ja vimpaimien lisäys) ja hallita käyttäjätietoja (käyttäjän oikeuksien muokkaaminen, käyttäjän lisäys ja poistaminen).

Sivuston rakentaminen aloitetaan yleensä luomalla sivu tai artikkeli, joka onnistuu hallintasivun kautta. Sivuille ja artikkeleille voi sijoittaa tekstiä, kuvia ja PHP-, HTML- sekä JavaScript-ohjelmointikieltä, joista PHP vaatii lisäosan toimiakseen.

3.1 Teemat

WordPressissä teema määrittelee sivuston ulkonäön ja niitä on saatavilla tuhansiin eri käyttötarkoituksiin. Teeman vaihtaminen on erittäin helppoa teemahakemiston kautta, mutta se onnistuu myös siirtämällä halutun teeman zip-paketti sivustolle. Teemahakemistoon pääsee WordPressin hallintasivun kautta menemällä ensin Ulkoasuvälilehdelle ja valitsemalla sieltä Teemat-linkki, josta teemoja voi hakea niiden suosion,

avainsanan tai julkaisupäivän perusteella tai vaihtoehtoisesti käyttämällä ominaisuussuodatinta. Teemaa painamalla sen ulkonäköä ja toiminnallisuutta voi tutkia ja sen lataus onnistuu Lataa teema-linkistä. Teeman voi poistaa listausnäkyvässä klikkaamalla ensin halutun teeman tiedot näkyviin ja sitten poista. (Leiniö 2013.)

Sivuston teeman vaihtaminen WordPressiin tuli mahdolliseksi vuonna 2005 version 1.5 mukana. Se oli merkittävä helpotus sivuston ulkonäön muokkaukseen, sillä aiemmin muokkaus tapahtui CSS-tyylisivun kautta, mikä oli huomattavasti monimutkaisempaa ja vaati CSS tietämystä. (Leiniö 2013.)

3.2 Lapsiteemat

Vaikka Wordpressille on olemassa hyvin kattava valikoima erilaisia teemoja, saattaa käyttäjä haluta silti tehdä omia muutoksia. Muutoksia teeman ulkonäköön voi tehdä monella eri tavalla, mutta jos muokkauksen tekee väärin, saattaa se mahdollisesti aiheuttaa ongelmia, joten omien muutosten tekeminen teemaan on suositeltavaa tehdä lapsiteeman kautta. Aina ei toki tarvitse tehdä lapsiteemaa. Jos käyttää esimerkiksi valmisteemaa ja kaikki tarvittavat muutokset sivuston ulkonäköön pystyy tekemään Wordpressin hallintasivun kautta, on lapsiteeman käyttö turhaa. (Leiniö 2013.)

Lapsiteeman lienee suurin hyöty ilmenee isäntäteemaa päivitettäessä: jos muutoksia tehdään isäntäteeman koodiin, ne menetetään sen päivityksen yhteydessä, mutta jos ne tehdäänkin lapsiteeman koodiin, päivitys ei kadota tehtyjä muutoksia. Lapsiteemaa on helppo pitää ajan tasalla käyttämällä isäntäteeman uusimpia ominaisuuksia. Myös kaikki tehdyt muutokset koodiin löytyvät helposti yhdestä paikasta, kun taas alkuperäiseen teemaan tehdyt koodi-muutokset katoavat helposti muun koodin sekaan. (Leiniö 2013.)

Mutta kuinka Lapsiteema sitten tehdään? Ensiksi sille täytyy luoda uusi kansio wp-content/themes-hakemistoon. Kansion nimen voi päättää itse, mutta yleensä nimeksi laitetaan ensin isäntäteeman nimi ja loppuun -child. Esimerkiksi jos käyttää Wordpressiin uusinta oletusteemaa nimeltä TwentyFifteen, sen lapsiteeman nimeksi tulisi TwentyFifteen-child. Lapsiteema tarvitsee tyylitiedoston. Tyylitiedoston alussa määritetään lapsiteeman nimi sekä isäntäteeman hakemisto. Nimen voi valita vapaasti ja se näytetään hallintasivun teemahakemistossa. Templaten arvoksi tulee kirjoittaa sama kuin isäntäteeman hakemisto. Tavallisesti isäntäteeman tyylimääritteet halutaan lapsiteeman mukaan. Nämä tyylimääritteet ladataan CSS:n @import-toiminnolla. Tässä vaiheessa lapsiteema on otettavissa käyttöön hallintasivulla. Sivusto näyttää lapsiteeman käyttöön otettuaan tismalleen samalta, kuin isäntäteemalla, ellei muutoksia koodiin ole tehty. Mutta nyt muutoksia koodiin voi tehdä niitä menettämättä, kun teema päivitetään. (Leiniö 2013.)

3.3 Lisäosat

Lisäosat ovat pieniä ohjelmia, jotka voidaan lisätä sivustolle lisäämään ja laajentamaan sen toiminnallisuutta. Ne lisäävät sivuston monipuolisuutta ja niiden avulla WordPressiä voi kehittää ja muuttaa lähes miten tahansa. Tämän tutkimuksen tekohetkellä lisäosia oli saatavilla 37836 kappaletta. (WordPress 2015.)

Lisäosat ovat ladattavissa WordPressin omalta kotisivulta osoitteesta <https://wordpress.org/plugins/> tai WordPressin hallintasivun kautta, josta niitä voi etsiä vapaalla sanahauulla, suosion, suositusten tai julkaisupäivän perusteella. Kuten teemojakin, voi lisäosia asentaa kahdella eri tapaa. Lisäosat voi asentaa joko hallintasivun kautta sivuvalikosta kohdasta lisäosat ja valitsemalla sieltä linkki ”lisää uusi”. Toinen tapa on sijoittaa lisäosan tiedostot WordPressin kansioon wp-content/plugins. Tämä ei ole suositeltavaa haittakoodin ilmenemisen vaaran takia, mutta on joskus välttämätöntä, jos lisäosa löytyy ainoastaan muualta kuin WordPressin omasta hakemistosta. (Leiniö 2013.)

Kaikki WordPressiin asennetut lisäosat löytyvät sivupalkista kohdasta lisäosat ja valitsemalla asennetut lisäosat, josta ne myös voidaan ottaa käyttöön, poistaa käytöstä tai poistaa kokonaan WordPressistä. Asennettuja lisäosia voi hakea vapaalla sanahauulla tai suodattaa asennuksen tilan mukaan. Massatoiminnon avulla käyttöönoton, käytöstä poistamisen, päivityksen tai järjestelmästä poistamisen voi halutessaan tehdä useille tai kaikille lisäosille yhtä aikaa. (Leiniö 2013.)

3.4 WordPress mobiililaitteissa

WordPressillä luotu sivu skaalautuu mobiililaitteen näytölle sopivaksi ja näyttää samalta käytettäessä laitetta pystyssä sekä vaakatasossa. Sivuston käytössä olevalla teemalla ei ole väliä ja mobiililaitteen selain kohtelee kaikkia WordPressin teemoja samalla tavalla. Sivun leveys määräytyy sen yläbannerin leveyden mukaan ja mikäli jokin elementti on yläbannerista sivussa, selain siirtää sen sivun alareunaan.

Automattic inc:in kehittämä ilmainen sovellus Android-pohjaisille mobiililaitteille nimeltä WordPress tarjoaa rajoitetun, mutta tehokkaan valikoiman työkaluja WordPressillä luotujen sivujen muokkaamiseen, päivittämiseen sekä tarkkailuun. Se tukee sekä wordpress.comin, että yksittäisten henkilöiden julkaisemia ja ylläpitämiä sivustoja WordPressin versiosta 3.5 ylöspäin. Sisään kirjautuessa ensin sovellus kysyy sivuston www-osoitteen, adminin tunnuksen sekä salasanan. Sovelluksella voi muun muassa lisätä uusia tai muokata jo sivustolle lisättyjä artikkeleita, sivuja ja kommentteja. Tällä WordPress-mobiilisovelluksella voi lisätä sivustolle myös mediaa, kuten kuvia ja videoita. Ne voi lisätä artikkeleihin ja sivuille puhelimen muistikortilta tai halutessaan ottaa suoraan mobiililaitteen kameralla, mikäli sellainen on käytössä. Sivuston статистиikkaa voi myös tutkia, mutta sitä varten täytyy ladata erillinen lisäosa nimeltä Jetpack. Tämä lisäosa mahdollistaa sivuston katselukertojen, vierailijoiden, tykkäyksien ja kommenttien tilastojen tutkimisen.

4 SIXPACK

Sixpack on web-sivusto, jota rakensin tehdessäni tätä tutkimusta. Se on fitness-sivusto, jolla käyttäjä voi seurata omaa fyysistä kehitystään, kuten painon nousua tai laskua ja treenipainojen kehitystä. Sivusto on suunnattu kaikenlaisille kuntosalin käyttäjille riippumatta siitä, onko tavoitteena vähentää rasvakudoksen määrää kehossa, tai kasvattaa lihasmassaa. Sivusto vaatii rekisteröitymisen, jotta sen ominaisuuksia voi käyttää. Rekisteröityessä sivusto kysyy käyttäjältä perustietoja kuten nimeä, ikää, painoa, sähköpostiosoitetta ja pituutta. Myös oma tavoite, kuntotaso, aikataulu ja harrastukset on hyvä ilmoittaa rekisteröityessä, jotta kunto-ohjelma voidaan räätälöidä niiden mukaan. Käyttäjän halutessaan personal trainer voi myös laatia käyttäjän tavoitteisiin räätälöidyn ruoka-ohjelman.

Sivuston toiminnallisuus muodostuu pitkälti lisäosista. Sivustolle rekisteröityminen tapahtuu WP-Members-lisäosaa käyttämällä. Tässä ilmaisessa lisäosassa rekisteröitymissivua pystyi muokkaamaan haluamallaan tavalla, kuten lisäämällä pakollisia tai vapaaehtoisia tietovaatimuksia. Luonnollisesti WordPressiin piti asentaa myös PHP-koodin käyttöä varten lisäosa, jotta se toimisi artikkeleissa ja sivuilla. Tässä lisäosassa normaalisti käytössä olevat PHP-kielen aloitus- ja lopetuselementit eivät toimi, vaan ne on korvattu `[insert_php]`- ja `[/insert_php]`-elementeillä. User Access Manager-lisäosalla piilotin sivupalkissa sijaitsevat linkit, joita ei ole tarkoitettu rekisteröitymättömille käyttäjille. Rekisteröityessä ne tulevat näkyviin. Julkaisijan nimen näkyminen julkaisuissa oli tarpeetonta, joten käytin sen piilottamiseen lisäosaa nimeltä Show Hide Author. CSS:n muokkaukseen kokeilin kahta lisäosaa (CSS & JavaScript Toolbox, Simple Custom CSS), mutta ne eivät toimineet halutulla tavalla, joten päädyin tekemään CSS-muutokset WordPressin `style.css`-tiedostoon. Muutoksia tein `body`n läpinäkyvyyteen (`opacity`) ja navigointivalikon taustaväriin.

WP Business Intelligence Lite oli erinomainen lisäosa viivakaavioiden lisäämiseen julkaisuihin. Käyttäjä pystyi lisäämään kaavioon tietoa esimerkiksi seuratakseen oman kehonpainon muutoksia, mutta valitettavasti tämä lisäosa ei toiminut lainkaan

käyttämälläni Haunted-lisäosalla. Yritin korvata tämän ominaisuuden JavaScript-koodilla, mutta tuntemattomasta syystä sekään ei toiminut ja sivu näytti tyhjää. Väliaikaisena ratkaisuna päädyin lisäämään edellä mainitun JavaScript-koodin erilliselle php-sivulle, johon pääsisi sivustolle sijoitetun linkin kautta.

Käytin sivuston RoseCitySisterin luomaa teemaa nimeltä Haunted, josta myöhemmin loin lapsiteeman. Teemaa täytyi muokata, jotta se sopi tämän sivun tarkoitukseen. Vaihdoin taustalle mukautetun taustakuvan, joka sopi fitness -teemaan. Taustakuvan kansio sijaitsi teeman kansiossa osoitteessa /home/sixpack/public_html/wp_content/themes/haunted-child/images. Haunted-teemalla ei ollut yläbanneria lainkaan, joten lisäsin siihen mukautetun bannerin kooltaan 694*398 pikseliä. Bannerin kansio sijaitsi samassa osoitteessa taustakuvan kanssa. Navigointivalikko sulautui uuteen mukautettuun taustakuvaan liian hyvin, eikä linkeistä saanut enää kunnolla selvää. Tähän väliaikaisena ratkaisuna lisäsin valikon taakse värimaailmaan sopivat taustat

Oletuksena sivun ylälaidassa näkyvä asetuspalkki oli hyödytön ominaisuus. Sen piilottaminen muilta, paitsi adminilta onnistui lisäämällä functions.php-tiedostoon seuraava PHP-koodi:

```
add_action('after_setup_theme','remove_admin_bar');
functionremove_admin_bar() {
if (!current_user_can('administrator') && !is_admin()) {
        show_admin_bar(false);
}
}
```

Tein kaikki muutoksen koodiin WinSCP-ohjelman kautta.

LÄHTEET

Jouni Heikeniemi. 2002. Www-dokumentti. Saatavissa:

<http://www.heikniemi.fi/svwww-vukk/k13.html>. Luettu 1.2.2015.

Jouni Heikeniemi. 2002. Www-dokumentti. Saatavissa:

<http://www.heikniemi.fi/svwww-vukk/k33.html>. Luettu 2.2.2015.

Jukka Korpela. 2009. Www-dokumentti. Saatavissa:

<https://www.cs.tut.fi/~jkorpela/webjulk/3.2.html#js-mita>. Luettu 20.3.2015.

Jukka-Pekka Kerkkänen. 2009. Www-dokumentti. Saatavissa:

<http://opinnot.net/tietotekniikka/ohjelmointi/html/index.php>. Luettu 15.3.2015.

Keski-Suomen Verkkomedia. 2009. Www-dokumentti. Saatavissa:

<http://www.php-perusteet.com/>. Luettu 11.2.2015.

Mervi Rossi. 2010. Www-dokumentti. Saatavissa:

<http://www.avkymppi.net/joomla/html/css.html?start=1>. Luettu 20.2.2015.

Ohjelmointi. 2015. Www-dokumentti. Saatavissa:

<http://ohjelmointi.medianurkka.com/?p=21>. Luettu 13.3.2015.

Ohjelmointi. 2015. Www-dokumentti. Saatavissa:

<http://ohjelmointi.medianurkka.com/?p=59>. Luettu 3.3.2015.

Peltomäki, J. 1998. WWW-ohjelmointi. Jyväskylä: Teknolit.

Petteri Perälä. 2003. Www-dokumentti. Saatavissa:

<http://koti.mbnet.fi/~petepe/css1/cssinfo/info.htm#mcss>. Luettu 16.3.2015.

Q-Success. 2015. Www-dokumentti. Saatavissa:

http://w3techs.com/technologies/overview/content_management/all. Luettu 1.3.2015.

Timo Leiniö. 2013. Www-dokumentti. Saatavissa:

<https://wpopas.fi/videot/teemat>. Luettu 16.2.2015.

Timo Leiniö. 2013. Www-dokumentti. Saatavissa:
<https://wpopas.fi/kuinka-luot-lapsiteeman-ja-mita-hyotya-siita-on>. Luettu 15.2.2015.

Timo Leiniö. 2013. Www-dokumentti. Saatavissa:
<https://wpopas.fi/videot/lisaosan-asennus>. Luettu 15.2.2015.

Wordpress. 2015. Www-dokumentti. Saatavissa: <https://wordpress.org/plugins>. Luettu 11.3.2015.

W3C. 2015. Www-dokumentti. Saatavissa: <http://www.w3.org/People/Berners-Lee>. Luettu 10.3.2015.