



SAVONIA

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

VERKKOSIVUJEN LASKURI LISÄOSA WORDPRESSIIN

TEKIJÄ: Jani Kellomäki

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma/Tutkinto-ohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Jani Kellomäki	
Työn nimi Verkkosivujen laskuri lisäosa WordPressiin	
Päiväys	15.12.2020
Sivumäärä/Liitteet	31
Ohjaaja(t) Keijo Kuosmanen, Janne Koponen ja Joanna Niininen	
Toimeksiantaja/Yhteistyökumppani(t) Sollertis Oy	
<p>Tiivistelmä</p> <p>Opinnäyte toteutettiin yhteistyössä Sollertis Oy:n kanssa. Projektissa oli päämääränä luoda WordPress-sisällönhallintajärjestelmään helposti lisättävä lisäosa, jolla asiakkaat voivat laskea uusien verkkosivujensa hinnan. Ensimmäinen projektin vaihe oli tutkia, kuinka WordPress alustalle luodaan lisäosa sekä kuinka julkiseen WordPress-alustan lisäosa valikkoon saisi lisättyä oman lisäosan niin, että muutkin voisivat sitä mahdollisesti käyttää. Lisäosa lisättiin toimeksiantajan verkkosivulle, joten lisäosaa luodessa piti myös miettiä mahdollisia tietoturvaan liittyviä ongelmia sekä ulkoasua.</p> <p>Lisäosan kehityksen jaoin hallintapaneelin koodaukseen sekä käyttäjä näkymään ja sen ulkoasun muokkaukseen. Jokainen lisäosa WordPress alustalle toteutetaan PHP-kielillä. Tässä lisäosassa käytin PHP kielen lisäksi myös html- ja javascript kieltä sekä ulkoasua varten css kieltä. Toteutuksessa piti myös tutustua sekä käyttää WordPress hallintajärjestelmän valmiita toiminnallisuuksia sekä ominaisuuksia. Valmiita toimintoja hyödyntäen lisäosasta sai toimivan ja helposti muokattavan.</p> <p>Itse lisäosan toteutus onnistui WordPress toimintojen itseopiskelun jälkeen hyvin ja se mahdollisti toiminnallisuuksien kehittämisen, joita lisäosaan haluttiin. Ulkoasusta tuli selkeä ja käyttäjäystävällinen. Lisäosa vaatii vielä jatkokehitystä tulevaisuutta varten, jotta se toimii kaikilla teemoilla sekä erilaisten lisäosien kanssa ilman yhteensopimattomuusongelmia. Lisäosan kehityksessä kävi selväksi, että WordPress alustan kehittäjille on todella paljon materiaalia kehityksessä hyödyntämiseen. Hyvät materiaalit ja dokumentoinnit auttoi kehityksessä ja opiskelussa.</p>	
Avainsanat	
WordPress, lisäosa, julkaisu järjestelmä, sisällönhallintajärjestelmä	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Jani Kellomäki			
Title of Thesis Website price counter plugin for WordPress			
Date	December 15, 2020	Pages/Appendices	31
Supervisor(s) Keijo Kuosmanen, Janne Koponen and Joanna Niininen			
Client Organisation /Partners Sollertis Oy			
<p>Abstract</p> <p>The thesis was implemented in co-operation with Sollertis Oy. The goal of the project was to create an plugin that could be easily installed to the WordPress content management system, with which customers can calculate the price of their new website. The first phase of the project was to study how to create a plugin for the WordPress platform and how to add your own plugin to the public WordPress platform plugin's menu so that others could possibly use it as well. The plugin was added to the client's website, so when creating the plugin, one had to think about possible security-related problems as well as the layout of the plugin.</p> <p>Plugin development was split into control panel coding and user view and customization of its layout. Each plugin for the WordPress platform is coded with PHP coding language. This plugin, in addition to PHP, was also created with html and javascript language, and css language for the layout. In the implementation one also had to get acquainted with and use the ready-made functionalities and features of the WordPress management system. Utilizing ready-made functions, the plugin became functional and easily customizable.</p> <p>The implementation of the plugin itself was successful after studying the WordPress functions. Self-studying made possible to implement the functionalities which were required and made a clear and user-friendly look for the plugin. The plugin still needs further development for the future, so that it works with different themes as well as with various different plugins without any compatibility issues. In plugin development, it became clear that WordPress platform developers had a lot of material in development to utilize. Good materials and documentation helped with the development and learning.</p>			
<p>Keywords CMS, platform, WordPress, webpage</p>			

ESIPUHE

Opinnäytetyön aiheena on toteuttaa helposti muokattavissa oleva verkkosivulaskuri, jolla asiakas voi laskea tulevan verkkosivun hinnan. Työ helpottaa yrityksen toimintaa sekä asiakkaan "elämää". Samalla opinnäytetyössä kehitän osaamistani WordPress alustalla koodaamisessa sekä kehitän samalla omaa ohjelmointitaitoa.

Työn teoria osiossa käydään läpi yleisiä tietoja WordPressistä sekä tutustutaan WordPressin julkiseen lisäosavalikoimaan ja kuinka sinne olisi mahdollista lisätä oma lisäosa. Tutkitaan myös WordPressissä jo itsessään olevia ominaisuuksia, joita hyödynnetään ohjelmointi vaiheessa. Ohjelmointi osuudessa käydään läpi itse lisäosan kehitystä sekä käytettyjä tekniikoita. Lopussa käydään läpi opittuja asioita sekä haasteita, joita vastaan tuli.

LYHENTEET JA MÄÄRITELMÄT

XSS = Cross-site scripting, XSS antaa hyökkääjälle mahdollisuuden injektoida komentosarjoja verkkosivuille (WordPress, 2020)

WAMP = web-kehitysympäristö Windows käyttöjärjestelmälle.

MYSQL = Tietokantajärjestelmä

RSS = verkkosyötemuoto, mahdollistaa digitaalisen sisällön julkaisemisen.

FTP = Tiedonsiirtomenetelmä tietokoneiden välillä, FTP avulla otetaan yhteys palvelimeen.

Funktio = Koodi, jota voi käyttää monta kertaa, funktio voi tehdä erilaisia toimintoja.

Hook = hook, eli koukku on WordPress toiminto, jolla voi muokata wordpressin oletustoimintoja.

API = Sovellusohjelmointirajapinta, eri WordPress rajapinnat sisältävät erilaisia toiminnallisuuksia joita voi hyödyntää lisäosan kehityksessä.

SQL Query = Tietokantakysely, jonka avulla voidaan haeta tietoa tietokannasta.

SISÄLTÖ

1	JOHDANTO	6
2	WORDPRESS SISÄLLÖNHALLINTAJÄRJESTELMÄ	7
2.1	WordPressin asennus	7
2.2	Käyttöönotto.....	8
2.3	Hallintapaneeli	8
2.4	Lisäosat.....	11
2.5	Sivuston rakennus.....	12
2.6	Ylläpito ja kehitys.....	14
3	LISÄOSAN KEHITYS WORDPRESS-ALUSTAAN	15
3.1	WordPress toimintafunktiot.....	15
3.2	API rajapinnat	16
3.2.1	Shortcode API.....	16
3.2.2	Options API	17
3.2.3	Settings API.....	18
3.3	Lisäosan tietokanta	18
3.4	Lisäosan tietoturva.....	19
3.5	Lisäosan lisääminen julkiseen lisäosat-valikkoon.....	20
4	VERKKOSIVULASKURIN TOTEUTUS.....	22
4.1	Lisäosan hallinta	22
4.2	Tietokanta.....	24
4.3	Tiedon lisääminen ja muokkaaminen	25
4.4	Käyttäjä näkymä	26
4.5	Tuotteiden hintojen lasku	27
4.6	Valintojen tallennus.....	27
5	VERKKOSIVULASKURIN RAKENNE.....	28
6	YHTEENVETO.....	29
	LÄHTEET JA TUOTETUT AINEISTOT	30

1 JOHDANTO

Opinnäytetyö käsittelee verkkosivulaskuri lisäosan rakentamista WordPress sisällönhallintajärjestelmään Sollertis Oy:lle. Työssä tutkitaan sisällönhallintajärjestelmän toiminnallisuuksia sekä mietitään lisäosankehityksen kannalta oleellisia aiheita. Teoriaosiossa käydään yleisesti läpi WordPressin toimintaa ja käyttöönottoa asennuksesta lähtien. Sekä tutkitaan tarkistusprosessia, joka lisäosille suoritetaan, kun niitä lisätään julkiseen WordPress laajennusvalikkoon. Teoriaosion jälkeen käydään läpi verkkosivulaskurin kehityksen vaiheita ja lisäosan ominaisuuksia. Lopussa on pohdinta, jossa käydään läpi työssä vastaan tulleita haasteita ja onnistumisia.

2 WORDPRESS SISÄLLÖNHALLINTAJÄRJESTELMÄ

WordPress vuonna 2003 aloitti toimintansa blogi alustana, joka kehittyessään sai suuren suosion sisällönhallintajärjestelmänä, nykypäivänä WordPress alustalla julkaistaan kaikenlaisia sivuja. Yli 455 miljoonaa sivua ympäri maailmaa on tehty WordPress alustaan. Tämä tarkoittaa 35% markkinaosuutta kaikista maailman verkkosivuista. 69% WordPress sivuista käytetään sisällönhallintaan ja vain 6% nykypäivänä on pelkästään blogaamiseen tarkoitettuja sivustoja. Wordpressiä käyttävät kaiken kokoiset yritykset sekä yksityishenkilöt. (Rashty, 2020).

2.1 WordPressin asennus

WordPressin asennuksen ensivaiheena on palvelinympäristön valinta. WordPressin käyttöä varten voi luoda oman palvelinympäristön käyttäen pilvipalveluita tai ostaa webhotellipaketin palvelun tarjoajalta. WordPress on mahdollista myös asentaa omalle tietokoneelle WAMP-sovelluksen avulla, tämä toimii omalla tietokoneella virtuaalipalvelimen tapaan. WAMP-ympäristöön voi asentaa ja testata WordPressin toimintoja sekä ominaisuuksia lokalisoidusti omalla tietokoneellasi. WordPress tarvitsee toimiakseen uuden PHP version sekä MySQL tietokannan.

Kun palvelinympäristö on valmis, voi aloittaa itse WordPressin asennuksen. Asennuspaketin voi ladata ilmaiseksi WordPressin kotisivuilta. Tiedostot siirretään palvelimen web-juurihakemistoon, joka usein on "httpdocs" nimellä. Tämän jälkeen voi aloittaa asennuksen suoraan menemällä WordPress asennuksen juureen selaimella.

WordPressin asennus alkaa kielen valinnalla, käyttäjä voi valita haluamansa kielen listasta. Kielen jälkeen alkaa itse WordPressin asennus, ensimmäisessä vaiheessa on tietokannan luonti. Tietokannalle annetaan nimi, käyttäjätunnus, salasana, palvelin sekä taulujen etuliite (kuva 1). Jatketaan näppäimestä pääsee syöttämään halutut asetukset tietokantaa varten.

Kun asetukset ovat kunnossa, pääsee asennusvaiheessa jatkamaan eteenpäin. Asetusten jälkeen alkaa WordPressin asennus. Asennuksen päätyttyä voi asetetuilla tunnuksilla kirjautua hallintapaneeliin oman selaimen kautta.



Tervetuloa asentamaan WordPressiä. Tarvitset seuraavat tietokantaan liittyvät tiedot ennen asennuksen aloittamista.

1. Tietokannan nimi
2. Tietokannan käyttäjätunnus
3. Tietokannan salasana
4. Tietokantapalvelin
5. Tietokantataulujen etuliite (jos haluat asentaa samaan tietokantaan useamman WordPress-sivuston)

Käytämme näitä tietoja luodaksemme wp-config.php-tiedoston. **Jos automaattinen tiedoston luonti ei onnistu, ei hätää. Voit myös avata tiedoston wp-config-sample.php tekstieditoriin, täyttää tarvittavat tiedot ja tallentaa sen nimellä wp-config.php.** Tarvitseko lisää apua? [Sitä löytyy.](#)

Olet saanut todennäköisesti nämä tiedot palveluntarjoajaltasi. Muussa tapauksessa sinun pitää olla ensin palveluntarjoajaasi yhteydessä tietojen saamiseksi. Jos olet valmis...

Jatketaan!

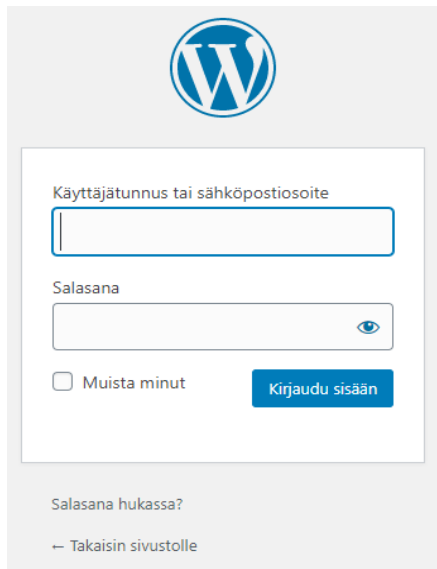
KUVA 1 Asennus vaihe yksi

2.2 Käyttöönotto

WordPressin käyttöönotto vaiheessa pitää päättää, mitä haluaa verkkosivultaan. Tuleeko siitä normaali verkkosivu ilman verkkokauppaa, verkkokaupan sisältävä sivu vai pelkkä blogialusta. WordPressin teema valikoimassa on tuhansia ilmaisia ja maksullisia teemoja, joista voi valita omaan käyttöön parhaiten sopivan. Teeman valinnassa kannattaa miettiä, millaisen ulkoasusta haluaa. Valintaa helpottamiseksi monet teemat ovat tehneet esimerkkejä kyseisellä teemalla luoduista sivuista.

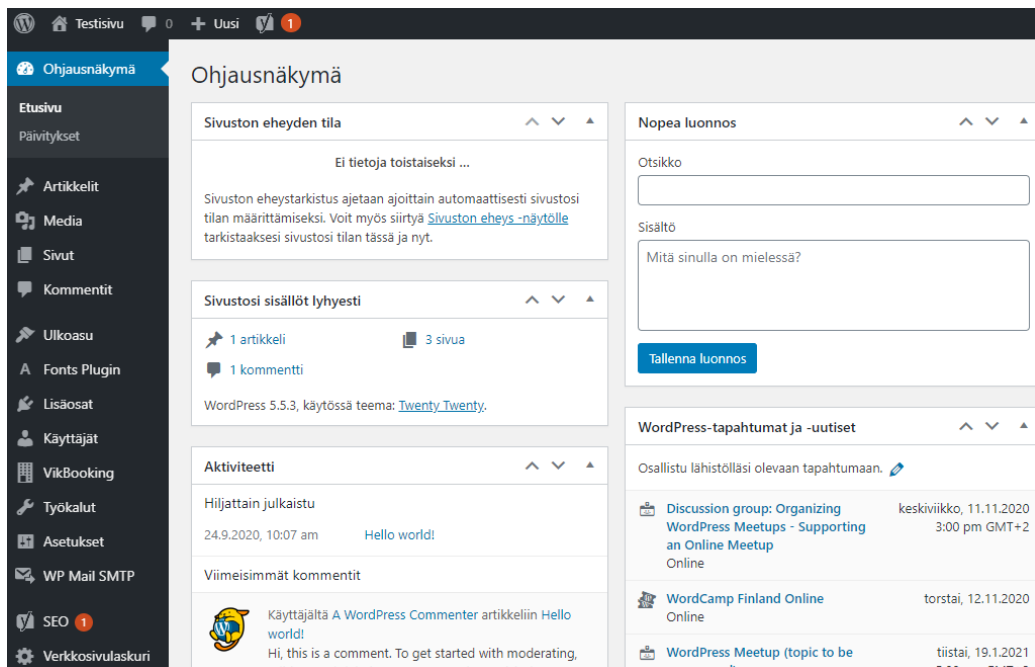
2.3 Hallintapaneeli

WordPress hallintapaneeliin kirjaututaan wp-login.php-tiedoston avulla. Osoitekentän kautta www.domain.fi/wp-login tai vaihtoehtoisesti [/wp-admin](http://www.domain.fi/wp-admin) (Kuva 2). Kirjautumisessa käytetään asennusvaiheessa asetettua käyttäjää sekä salasanaa. Kirjautumisruudussa on myös mahdollisuus nollata salasana, jos se on päässyt unohtumaan.



KUVA 2 Hallintapaneeliin kirjautuminen

WordPress hallintapaneeli sisältää käyttäjälle monia erilaisia mahdollisuuksia sekä ominaisuuksia muokata sivuaan halutunlaiseksi (Kuva 3). Käyttäjä voi julkaista uusia sivuja sekä postauksia eli artikkeleita sivullensa reaali-ajassa tai ajastettuna julkaisuna. Sivut ja postaukset tulevat verkkosivulle näkyviin julkaisun yhteydessä. WordPress mahdollistaa myös yksityiset ja salasana-suojatut sivut sekä postaukset. Yksityiset sivut näkyvät vain järjestelmään kirjautuneille. Salanasuojatut sivut näkyvät kävijälle vain salasanan asettamisen jälkeen.



KUVA 3 Hallintapaneeli

Hallintapaneelin media osiossa käyttäjä voi lisätä uusia mediatiedostoja sivulleen. Mediatiedostoja voi hyödyntää sivuja rakentaessa. WordPressiin voi ladata eri tyyppisiä mediatiedostoja. Hyväksytyjä tiedostoja on kuvat, dokumentit, äänet sekä videot. Mediat tallentuvat palvelimelle, jolloin ne

ovat käytössä kaikkialla sivulla. Median koko tulee ottaa huomioon. Eri palvelimet antavat käyttäjien lisätä eri kokoisia tiedostoja palvelimelle, tämä saattaa estää suurien tiedostokokojen lisäämisen.

Lisäosat-valikko sisältää asennetut lisäosat, lisää uusi ja lisäosaeditor vaihtoehdot. Asennetuissa lisäosissa näet kaikki omat lisäosat, joita WordPressiin on asennettu. Niitä voi ottaa pois käytöstä ja laittaa päälle. Lisää uusi -vaihtoehdosta pääsee julkiseen lisäosavalikoimaan, josta voi etsiä lisäosia tuhansien lisäosien valikoimasta. Lisää uusi -valikossa voi myös itse lisätä .zip-tiedostosta lisäosan sivulle. Lisäosaeditori sisältää kaikkien aktiivisten lisäosien koodit. Editorissa voi muokata lisäosien koodeja, mutta se ei ole suotavaa, sillä niiden muokkaaminen saattaa rikkoa lisäosan.

Käyttäjät -valikosta ylläpitäjä voi lisätä uusia käyttäjiä ja antaa heille tarvittavat oikeudet muutosten tekoa varten. Käyttäjärooleja on 5 eri oikeuksilla olevaa. Rooleista ylin on Ylläpitäjä, joka omaa kaikki oikeudet. Seuraavana on Päätoimittaja, joka voi julkaista ja muokata sisältöä. Päätoimittajan jälkeen tulee kirjoittaja, jolla on vain omien postauksien muokkaus ja uusien julkaisujen oikeudet. Avustaja tulee kirjoittajan jälkeen, joka voi kirjoittaa ja muokata sekä poistaa omia postauksia, mutta ei voi julkaista niitä itse. Viimeisenä tulee Tilaaaja, jolla on oikeudet lukea ja kirjoittaa kommentteja sekä muokata omaa profiilia. (WordPress, Roles and Capabilities, 2020)

Hallintapaneeli sisältää myös Työkalut sekä Asetukset ruudut. Työkalut ruudussa voit tuoda tietoa sivullesi tai viedä tietoa ulos. Tuonti työkalu mahdollistaa artikkelien tuonnin sivulle RSS syötteen kautta tai Tumblerin kautta. (kuva 4).

Blogger Asenna nyt Tiedot	Tuo artikkelit, kommentit ja käyttäjät Blogger-blogista.
Kategorioiden ja avainsanojen muunnin Asenna nyt Tiedot	Muunna kategoriat avainsanoiksi tai avainsanat kategorioiksi.
LiveJournal Asenna nyt Tiedot	Tuo artikkelit LiveJournalista käyttäen heidän API:aan.
Movable Type ja TypePad Asenna nyt Tiedot	Tuo artikkelit ja kommentit Movable Type- tai TypePad -blogista.
RSS Asenna nyt Tiedot	Tuo artikkelit RSS-syöttestä.
Tumblr Asenna nyt Tiedot	Tuo artikkelit ja mediatiedostot Tumblrista käyttäen heidän API:aan.
WordPress Asenna nyt Tiedot	Tuo artikkelit, sivut, kommentit, lisäkentät, kategoriat ja tagit WordPressin vientitiedostosta.

KUVA 4 Asetukset -valikko

Asetukset kohta sisältää osiot Yleinen, Kirjoittaminen, Lukeminen, Keskustelu, Media, Osoiterakenne sekä Tietosuoja. Nämä ovat oletusasetukset WordPressissä. Asetukset -valikkoon voi myös lisätä omia asetuksia.

Yleinen -osiossa käyttäjä voi asettaa sivun URL-osoitteen sekä kuvauksen ja otsikon. Myös uusien käyttäjien rooleja voi muokata tässä osiossa. Yleiset asetukset -ruutu sisältää kaiken pienten asetusten säädön. Kirjoittaminen -ruutu sisältää WordPress artikkelien julkaisun sähköpostin avulla. Asetukseen voi asettaa sähköpostipalvelimen sekä tunnuksen tilille, johon artikkelit lähetetään. WordPress hakee artikkelit sähköpostilaatikosta ja julkaisee ne sivulla.

Lukeminen -asetukset ruudussa voi asettaa artikkeleille sivun, johon artikkelit menevät suoraan. Tämä ominaisuus on hyödyllinen esimerkiksi blogin teossa. Kaikki uudet artikkelit julkaistaan kyseiselle valitulle sivulle. Lukeminen -valikossa on mahdollista myös asettaa hakukonenäkyvyyden päälle tai pois sekä asettaa muita asetuksia artikkelejä varten.

Keskustelu -osiossa voi säätää kommentoinnin asetuksia. Mikäli halutaan pitää kommentit päällä sivulla, kannattaa silloin miettiä asetuksia tarkoin ja asentaa lisäksi tietoturvalisäosa tai google captcha, jolla voi estää kommentointiosion spämmäyksen.

Media-osio sisältää median oletusasetukset eli koon ja tiedostojen sijainnin. Oletuksena WordPress lajittelee tiedostot kansioihin vuoden ja kuukauden mukaan esimerkiksi "uploads/2020/11/". Osoiterakenne -osiossa voidaan määrittää sivun osoitteet esimerkiksi päivämäärän ja nimen mukaan tai artikkelin nimen mukaan. Tässä osiossa voidaan myös asettaa osoiterakenteelle mukautettu rakenne, johon WordPress antaa avainsanoja käytettäväksi (kuva 5).



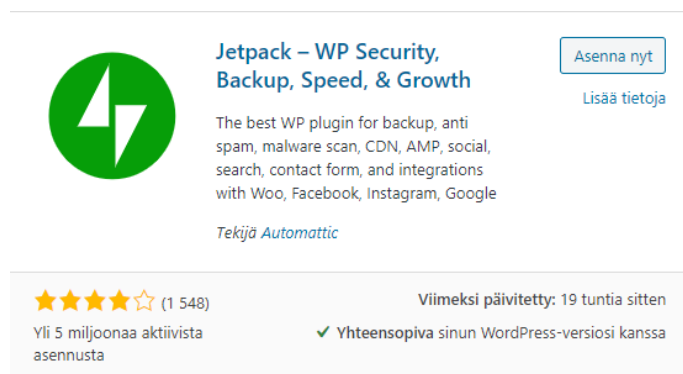
KUVA 5 Osoiterakenne -asetukset

2.4 Lisäosat

WordPress lisäosat ovat valmiita kolmannen osapuolen koodeja, joiden avulla voi lisätä uusia ominaisuuksia sivustolle. Lisäosat WordPress alustalla on kirjoitettu PHP-kielillä, joten ne voidaan lisätä saumattomasti toimimaan WordPress sivustollesi. Lisäosia on moniin eri tarpeisiin esimerkiksi google näkyvyyden lisäämiseen, tietoturvaan tai uusien ominaisuuksien lisäämistä varten.

Lisäosia voi asentaa sivulle suoraan WordPressin hallintapaneelin lisäosat valikosta. Lisäosien valinnassa kannattaa ottaa huomioon latausmäärät sekä viimeinen päivityksen ajankohta (Kuva 6). Latausmäärät kertovan lisäosan suosiosta ja näin voi luottaa, että se on toimiva ja turvallinen. Yleisimmin lisäosiin tulee päivityksiä noin yhden kuukauden välein, riippuen lisäosan toiminnallisuuksista.

Monta kuukautta päivittämättömiä lisäosia ei kannata lisätä sivustolle, sillä se tuo toimimattomuus- ja tietoturvariskejä. Vanhojen ja päivittämättömien lisäosien kautta tunkeutujat voivat löytää aukon verkkosivullesi, joka voi mahdollistaa tunkeutujalle mahdollisuuden esimerkiksi vaihtaa salasanoja tai poistaa tietoja.



KUVA 6 Lisäosan asennus

Lisäosien latauksen jälkeen niitä tulee päivittää aina kun mahdollista. Päivitykset voit tehdä hallintapaneelin päivitykset valikosta. Päivitykset osioon tulee aina ilmoitus, kun lisäosaan on tullut uusi päivitys. Päivittämisen yhteydessä kannattaa ottaa varmuuskopio sivusta, jos päivityksen yhteydessä sivusto menee rikki. UpdraftPlus lisäosa on suosittu ja helppokäyttöinen varmuuskopiointi lisäosa. Lisäosalla voi varmuuskopioida sivun pilveen tai palvelimelle. Lisäosalla voidaan varmuuskopioida sivut asetuksen avulla suoraan ennen päivitysten tekoa. Sivu voidaan myös palauttaa aiempaan varmuuskopio versioon, jos lisäosien tai teeman päivittäessä sivu menee rikki tai jokin ei toimi halutulla tavalla.

2.5 Sivuston rakennus

Sivuston rakennuksen voi aloittaa heti WordPressin asennuksen jälkeen. Rakennuksessa voi käyttää WordPressin oletusteemaa tai asentaa uuden teeman tuhansien eri teemojen valikosta. Teeman mukana tulee erilaisia ominaisuuksia, jotka helpottavat sivun rakennusta sekä tekevät ulkoasusta helposti muokattavan ilman erillistä koodausta.

Uudet teemat ovat yleensä lapsiteemoja, jotka vaativat pääteeman. Pääteema sisältää kaikki vaadittavat WordPress pohjan tiedostot. Pääteemat tuovat erilaisia ominaisuuksia ja jotkin pääteemat sisältävät lapsiteemoja, joita voi suoraan asentaa sivulle. Yksi suosituimmista pääteemoista on Genesis, joka sisältää erilaisia sivun asettelu asetuksia ja sivueditorin lohkoasetukset. Lohkoasetukset helpottavat uusien sivujen luontia. Lohkoasetukset mahdollistavat sivueditorissa suoraan uusien lohkojen lisäämisen. Lohkot voivat sisältää esimerkiksi videon tai kuvan ja tekstin. Pääteema on lapsiteeman pohja, jonka koodia ei muokata. Pääteema päivitetään aina uuden päivityksen

yhteydessä ja sen takia pääteeman koodiin ei saa tehdä omia muokkauksia. Päivityksen yhteydessä uusi koodi ajetaan pääteemaan ja silloin kaikki omat koodit ja muokkaukset lähtevät. Erilliset koodimuokkaukset tehdään lapsiteemaan.

Lapsiteema perii pääteeman ja sen kaikki toiminnot. Lapsiteema tuo sivun ulkoasun muutokset. Lapsiteemoja voi ladata suoraan WordPressin teema -valikosta ilmaiseksi tai ostaa maksullisen teeman netistä ja asentaa sen sivulle. Maksullisissa teemoissa usein on enemmän ominaisuuksia, siksi teemaa valittaessa pitää miettiä mitä sivulta haluaa. Lapsiteema mahdollistaa pääteeman päivityksen vaikuttamatta sivustoosi tekemiin muutoksiin. Lapsiteemaan voi koodata omia uusia ominaisuuksia sekä muokata teeman sisältämiä ominaisuuksia. (Wordpress, developer.wordpress.org, 2020).

Lapsiteeman voi myös halutessaan luoda itse. Tämä mahdollistaa täyden hallinnan verkkosivusta. Pienellä HTML, CSS ja PHP koodiosaamisella voidaan luoda sivuista juuri sellainen kuin halutaan. Teeman luonti aloitetaan luomalla uusi kansio teemalle sekä sen alle CSS tiedosto eli tyylitiedosto. Tiedostoon lisätään WordPress teemoissa olennainen tyylitaulukon otsikko (Kuva 7). Tyylitiedoston otsikko sisältää Teeman nimen, lisätietoja teemasta, teeman tekijän nimen ja osoitteen sekä teeman version. Template kohdalle annetaan pääteema, joka on käytössä sivulla.

```
/*
Theme Name:   Janin oma Teema
description:  >-
    Genesis lapsiteema, kustomoitu by Jani
Author:       Jani Kellomäki
Template:     Genesis
Version:      1.0.0
*/
```

KUVA 7 Tyylitiedoston otsikko

Teeman asennuksen jälkeen, verkkosivun rakennus voi alkaa. Eri teemat tuo erilaisia vimpainalueita sivulle. Vimpainalueille voi lisätä esimerkiksi sliderin tai tekstieditorin sekä kuvia (Kuva 8). Vimpain alueet tekevät sivun rakennuksesta yksinkertaista sekä suoraan mobiiliresponsiivista. Teemat eroavat usein huomattavasti juuri näiden vimpainalueiden kautta, siksi on tärkeää valita oikea teema, jotta vältytään erilliseltä koodaukselta. Verkkosivuja rakentaessa pieni css koodin ymmärtäminen ja osaaminen auttaa paljon ulkoasua muokatessa. Sekään ei ole vaadittua, kun sivut rakennetaan WordPress alustalle valmiin teeman avulla.

Home Top (Slider)	>
Home Top (Callout)	>
Home Top (News Row)	>
Home Row #1 (Left)	>
Home Row #1 (Right)	>
Home Row #2	>
Home Row #3 (Left)	>
Home Row #3 (Right #1)	>
Home Row #3 (Right #2)	>

KUVA 8 Vimpainalueet

2.6 Ylläpito ja kehitys

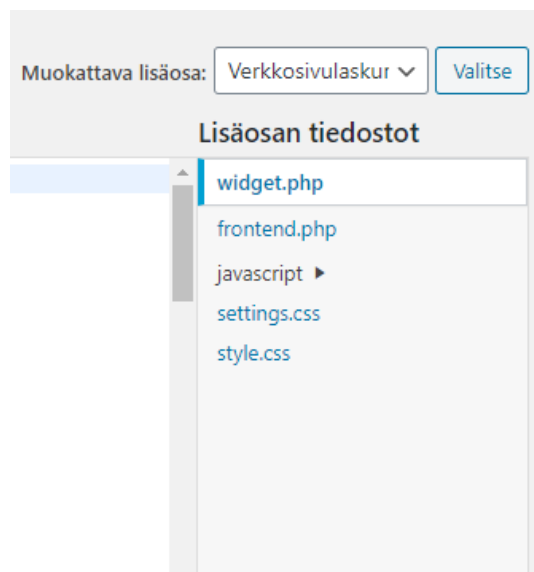
WordPress alustalla tehdyt verkkosivut tulee pitää aina ajantasalla, jotta sivut toimivat ja pysyvät tietoturvasivoina. Tähän kuuluu lisäosien päivittäminen sekä teeman ja WordPressin päivittäminen uuteen versioon. Ylläpidossa ja päivittämisessä tulee ottaa huomioon yhteensopimattomuusongelmia lisäosien ja teemojen välillä. Lapsiteemoja ei päivitetä, kuitenkin tulee ottaa huomioon, jos lapsiteema on itse tehty, joutuu sitäkin tietyissä tilanteissa päivittämään tekniikan kehittyessä. Lisäosia päivittäessä, tulee ottaa huomioon yhteensopimattomuusongelmat ja niiden välttäminen sekä mahdolliset tietoturva ongelmat lisäosissa.

Lisäosat ovat kolmannen osapuolen koodeja ja vaikka ne olisivatkin päässeet läpi WordPressin lisäosatarkastuksesta, voi kyseisiin lisäosiin silti tulla myöhemmin tietoturvaongelmia. Tietoturvaongelman huomattessa, pitää kyseinen lisäosa ottaa välittömästi pois käytöstä ja odottaa mahdollista päivitystä lisäosaan. Tietoturvaongelmat saattaa huomata oudoista käyttäjistä WordPressin tietokannassa tai muutoksista verkkosivuilla.

3 LISÄOSAN KEHITYS WORDPRESS-ALUSTAAN

WordPress lisäosat mahdollistavat uusien ominaisuuksien luomisen, mutta aina lisäosavalikosta ei löydy sitä oikeaa. Tässä tilanteessa oman lisäosan kehitys on paras vaihtoehto. Oman lisäosan avulla voidaan luoda halutut ominaisuudet sekä parantaa siten sivuston toimintaa. WordPress sovellusrajapinta sisältää paljon valmiita funktioita, jotka helpottavat lisäosan koodausta. Rajapinnat sisältävät ominaisuuksia, joita kehittäjät voivat käyttää lisäosaa koodatessa. Tällaisia ominaisuuksia ovat muun muassa lyhytkoodin luonti tai tietokantaan tiedon lisääminen. Rajapinnat mahdollistavat sen, ettei sinun tarvitse tehdä muutoksia WordPressin ydinkoodiin.

Lisäosaa voi suoraan muokata ja kehittää WordPressin omasta lisäosaeditorista. Editori sisältää kaikki lisäosan kansiot ja tiedostot (kuva 9). Lisäosan muokkaukset tulevat sivulle päivittäessä reaaliajassa, eikä tarvitse tuoda tiedostoja joka kerta uudestaan WordPressiin. Uusia tiedostoja tai kansioita ei voi luoda WordPressin kautta, vaan tulee ottaa FTP yhteys palvelimelle tai käytettävä palveluntarjoajan control panelia tiedoston lisäksi.



KUVA 9 Lisäosaeditor

3.1 WordPress toimintafunktiot

WordPress sisältää valmiita toimintafunktioita, joita ovat Koukut eli "Hookit". Koukkuja on kahta tyyppiä, toiminnot (action hook) ja suodattimet (filter hook). Koukkujen avulla voidaan lisätä tai muokata WordPressin toimintaa. Toiminnot ovat tehtäviä, esimerkiksi viestin lähettäminen tai tietokantaan tiedon lisääminen. (WordPress, 2020)

Uuden toiminnon lisääminen tapahtuu oman funktion luonnilla. Kun uusi toiminto eli funktio on määritelty, on se "hookattava" eli kiinnitettävä ja rekisteröitävä WordPressin kanssa.

Tämä tapahtuu `add_action()` -funktiolla (Kuva 10). `Add_action`- funktiolle annetaan neljä arvoa; Hookin eli koukun nimi, funktion nimi, prioriteetti ja argumenttien määrä (Kuva 11). (WordPress, 2020)

```
13
14 function oma_funktio( $post_ID, $post, $update ) {
15     // do stuff here
16 }
17
18 add_action( 'save_post', 'oma_funktio', 10, 3 );
19
20
```

KUVA 10 Uuden toiminnon koukuttaminen

```
23 add_action ( 'hook_name', 'your_function_name', [priority], [accepted_args] );
```

KUVA 11 `add_action`- funktion arvot

3.2 API rajapinnat

WordPress API on lyhenne sanoista WordPress Application Programming Interface eli sovellusliittymä. Sovellusliittymä on käyttöliittymä, jonka avulla sovellus voi käyttää kolmannen osapuolen toimintoja (Google, 2020.). WordPress sisältää monia API-osioita, joista jokainen sisältää eri toimintoja. Yhdessä WordPress API sisältää koko lisäosa/teema/laajennus-käyttöliittymän. WordPressin toimintaa pystyy laajentamaan kyseisten sovellusliittymien avulla. (Wordpress, https://codex.wordpress.org/WordPress_API%27s, 2020).

3.2.1 Shortcode API

Shortcode API on toiminto, jonka avulla koodista voidaan luoda lyhytkoodi. Kyseisen lyhytkoodin voi lisätä suoraan sivun tekstiin, sillä Shortcode API hoitaa suoraan kaikki hankalat jäsentelyt puolestasi. Tämä tekee lyhytkoodin lisäyksestä yksinkertaista. Uuden toiminnon voi lisätä ["shortcode nimi"] avulla, jossa lyhytkoodin nimi laitetaan "[]" tagien väliin.

Lyhytkoodin lisäämiseen sivulle tarvitaan funktio, jonka sisällä on halutut toiminnallisuudet.

`Add_shortcode` toiminnolla rekisteröidään lyhytkoodi, toiminto sisältää lyhytkoodin nimen sekä funk-

tion nimen, jota kutsutaan. Funktion luonnin jälkeen, sivulle lisätään lyhytkoodi suoraan tekstieditorin [oma_lyhytkoodi] avulla, joka palauttaa sivulle funktion sisällön. Tässä tapauksessa lyhytkoodi palauttaa sivulle: "Hei tämä on lyhytkoodilla tuotu teksti." (Kuva 12).

```
function Shortcode_example( $atts ){
    return "Hei tämä on lyhytkoodilla tuotu teksti.";
}
add_shortcode( 'oma_lyhytkoodi', 'Shortcode_example' );
```

KUVA 12 Lyhytkoodi

3.2.2 Options API

Options API on tapa tallentaa ja hakea tietoja tietokannasta. Rajapinta tekee tiedon luonnin, käytön, päivittämisen ja poistamisen helpoksi. Tiedot tallennetaan WordPressin tietokannassa olevaan wp_options-tauluun. Lisäosan kehityksessä käytetään kyseistä wp_options taulua tietojen haussa ja lisäämisessä.

Options API sisältää neljä eri funktiota, joilla voi tehdä halutut toimenpiteet. Tietokantaan lisäykseen käytetään "add_option ()" funktiota, jolla luodaan uusi tieto wp_options tauluun. Add_option funktioon syötetään option nimi sekä tieto, joka halutaan tallentaa tietokantaan. Tietokannasta option poistamista varten käytetään "delete_option ()" funktiota, jolle annetaan optionin nimi, joka halutaan poistettavaksi tietokannasta. Tiedon hakemiseen käytetään "get_option ()" funktiota. Funktiolle annetaan optionin nimi, jonka tiedot halutaan hakea. Tietojen päivittämistä varten käytetään "update_option ()" funktiota, jolle annetaan optionin nimi sekä uusi tieto (Kuva 13). (Wordpress, Options_API, 2020)

```
// $option sisältää optionin nimen.
// Luo optio. $autoload tarkoittaa sitä, että haluaako ladata option suoraan kun WordPress menee päälle.
add_option( $option, $value = 'optionin tieto', $autoload = 'yes' );

// Poista optio nimen avulla.
delete_option( $option );

// Hae option tietoja
get_option( $option, $default = false );

// Päivitä option tiedot.
update_option( $option, $uusitieto );
```

KUVA 13 Options API

3.2.3 Settings API

Settings APIa hyödynnetään, kun luodaan sivua, jossa käytetään WordPressin Menu APIa. Setting API:lla voidaan luoda eri välilehtiä ja tekstejä hallintapaneelin sivulle. Opinnäytetyössä kehitetyssä lisäosassa käytettiin hallintapaneelin luontiin ja lisäosan asetuksiin kyseistä settings APIa. Settings API on jaettu neljään eri luokkaan, jolla jokaisella voi luoda erilaisia osioita hallintapaneeliin. Luokka välilehti on luokka, joka sisältää kaikki muut 3 luokkaa sisällensä. Välilehti sisältää luokat osio, kentät ja asetukset.

Asetussivun luonti alkaa kyseisen sivun luonnilla. Asetus-sivun luontia varten käytetään `add_options_page()` funktiota, jolla saadaan luotua sivu hallintapaneelin valikkoon (kuva 14). Sivun luonnin jälkeen tarvitsee sinne tuoda tietoa. Tiedon tuonti onnistuu Settings API:n avulla. `Settings_field` ja `do_settings_field` funktioilla tuomme osiot ja kentät tyhjälle sivulle. Näihin kenttiin ja osioihin voi koodata haluaamaansa tietoa (kuva 15).

```
add_options_page(
    'Settings Admin',
    'Verkkosivulaskuri',
    'manage_options',
    'my-setting-admin',
    array( $this, 'create_admin_page' )
);
```

KUVA 14 `add_options`

```
add_settings_section(
    'setting_section_id',
    'My Custom Settings',
    array( $this, 'print_section_info' ), |
    'my-setting-admin' // Page
);

add_settings_field(
    'price1',
    'Ulkoasu: ',
    array( $this, 'Ulkoasu_settings' ),
    'my-setting-admin',
    'setting_section_id'
);
```

KUVA 15 `add_settings`

3.3 Lisäosan tietokanta

Lisäosalle voi lisätä WordPressin omaan tietokantaan uuden taulun, jota lisäosa käyttää. Uuden taulun lisääminen on kaksivaiheinen prosessi. Taulun lisäyksessä käytetään WordPress `wpdb`-globaali muuttujaa sekä `dbDelta`-funktioita. Nämä muuttujat sallivat muokautetun SQL-käskyn ajamisen ja suorittamisen WordPress tietokannassa. Uuden taulun voi luoda funktiota käyttäen ja sitä kutsuen toisesta tiedostosta (kuva 16). Funktion sisällä käytetään `wpdb`-funktioita, jonka avulla uusi tietokanta voidaan luoda. Ensin annetaan kyseiselle taululle nimi ja sen jälkeen taululle arvot. Kun taulu on määritelty voidaan ajaa `dbDelta`- funktio, joka ajaa taulun tietokantaan.

```

1 function LUO_taulu(){
2
3 global $wpdb;
4
5 $Uusi_taulu = $wpdb->prefix ."Tietokantataulu";
6
7 $sql = "CREATE TABLE $Uusi_taulu (
8     id int(10) NOT NULL AUTO_INCREMENT,
9     time datetime DEFAULT '0000-00-00 00:00:00' NOT NULL,
10    name tinytext NOT NULL,
11    text text NOT NULL,
12    PRIMARY KEY (id)
13 );";
14
15 require_once( ABSPATH . 'wp-admin/includes/upgrade.php' );
16 dbDelta($sql);
17 }

```

KUVA 16 Tietokannan luonti

Oman lisäosan -tiedostossa voi kutsua kyseistä lisäosatiedostoa (JK_database.php), joka luo uuden taulun tietokantaan, kun lisäosa aktivoidaan. (Kuva 17). Koukku "register_activation_hook" ajaa JK_database.php tiedoston "LUO_taulu" funktion. Funktion sisällä on SQL query, joka luo uuden taulun tietokantaan.

```

19 include_once("JK_database.php");
20
21 register_activation_hook(__FILE__, "LUO_taulu");

```

KUVA 17 Tietokanta tiedoston kutsuminen

3.4 Lisäosan tietoturva

WordPress lisäosan kehityksessä tulee ottaa huomioon monta kohtaa tietoturvaan liittyen. Ensimmäisenä tulee miettiä käyttäjärooleja ja sitä, kuka voi tehdä muutoksia lisäosaan. Käyttäjäroolien tarkistus onnistuu WordPressin roolien kautta. WordPressin "current_user_can" funktiolla, voidaan tarkistaa, onko käyttäjällä editointioikeuksia toisiin postauksiin, eli onko hän rooliltaan päätoimittaja tai ylläpitäjä (Kuva 18). Vain kyseisillä rooleilla olevat käyttäjät voivat tehdä muokkauksia. Roolien tarkistus on tärkeää varsinkin, jos sivuille voi kirjautua ja rekisteröityä kuka tahansa.

```

1 if ( current_user_can( 'edit_others_posts' ) ) {
2
3     add_filter( 'the_content', 'Funktio_oma' );
4
5     add_action( 'init', 'Oma_funktio' );
6 }

```

KUVA 18 Roolientarkistus

Toisena tietoturvan kannalta olennaisena asiana lisäosaa kehittäessä on ottaa huomioon, ettei käyttäjät pääse suoraan käsiksi kooditiedostoihin. Käyttäjää voi estää pääsemästä suoraan .php tiedostoihin selaimen URL-osoitteen kautta, tarkistamalla kooditiedoston alussa, onko yhteys suoraan URL-osoitteesta (Kuva 19). Koodissa tarkistetaan, tuleeko yhteys WordPress-ympäristöstä vai suoraan selaimen URL-osoitteesta, jolloin tiedostoihin pääsy evätään. Ilman kyseistä estoa, koodin operaatiot voidaan ajaa hyökkääjän toimesta, joka tuo merkittävän tietoturvariskin sivustolle.

```
10 if ( ! defined( 'ABSPATH' ) ) {  
11     exit;  
12 }
```

KUVA 19 Estä suorayhteys tiedostoon

WordPress sisältää myös monia apufunktioita, jotka auttavat lisäosan tietoturvan parantamisessa. Kehitysympäristö sisältää "sanitizing" ja "escaping" funktiot, joilla voidaan tarkistaa käyttäjän syöttämiä tietoja. Näitä funktioita on tärkeä käyttää, jos tekee lisäosaa, jossa käyttäjät voivat syöttää tietoja. Kun käyttäjä on lisännyt tietoa, tieto otetaan talteen muuttujaan. Muuttujassa käyttäjän antama tieto tarkistetaan ennen kuin sitä käytetään muualla (Kuva 20). (Wordpress, developer.wordpress, 2020)

```
15 <input id="nimi" type="text" name="nimi">  
16 $nimi = sanitize_text_field( $_POST['nimi'] );
```

KUVA 20 Tiedon tarkistus

Toinen apufunktio eli "escape" auttaa suojaamaan tiedot ennen niiden näyttämistä loppukäyttäjälle. Oikeassa muodossa tiedon välittäminen estää sivuston XSS (Cross-site scripting) hyökkäykset. Tiedon välittämisessä "echon" (Kuva 21) sijaan on normaalia käyttää WordPress lokalisointitoimintoa (Kuva 22). (Wordpress, developer.wordpress, 2020)

```
21 echo "Hello world";
```

KUVA 21 PHP tekstin tulostus

```
19 esc_html_e( 'Hello World', 'text_domain' );
```

KUVA 22 Esimerkki escape funktion käytöstä

3.5 Lisäosan lisääminen julkiseen lisäosat-valikkoon

WordPressin laajennushakemisto tarjoaa turvallisia lisäosia kaikille WordPressin käyttäjille. Tämän vuoksi jokainen lisäosa menee oman tarkistusprosessin läpi. Jotta lisäosan saa lisättyä julkiseen laajennushakemistoon, tulee sen täyttää kaikki 18 vaatimusta. Tärkeimpänä ovat lisenssointi sekä lisäosan ylläpito ja toiminnallisuudet.

Lisäosan tulee olla yhteensopiva GNU General Public lisenssin kanssa. Lisenssin saamiseen tulee jokaisen tiedostolla olla oikeat tekijänoikeusilmoitukset. Lisäosaan tulee lisätä GNU tiedostojen kopioita, jos sitä haluaa käyttää, sekä lisenssi-ilmoitus jokaiseen tiedostoon. Lisäosaa tulee kehittää ja sen pitää toimia jatkossa tulevien WordPress päivityksien kanssa. Koodin tulee olla ihmisen luettavaa, eikä koodia saa peittää, sekä lisäosasa on oltava dokumentoitu sen toiminta.

Kokeiluversiot eivät ole sallittuja. Lisäosa ei saa sisältää toimintoja, jotka aukeavat vain maksamalla tai päivittämällä. Toiminnallisuuksia ei voi poistaa käyttäjiltä, eikä lisäosa voi sisältää pelkästään yhteyden erillisiin API rajapintoihin. Maksulliset toiminnallisuudet ovat sallittuja, tässä tapauksessa lisäosa päivitetään maksulliseen versioon, joka tuo uudet toiminnallisuudet lisäosaan.

Lisäosa ei saa valvoa käyttäjän toimintaa eikä kerätä tietoa ilman käyttäjän antamaa lupaa. Lisäosa ei myöskään saa ottaa yhteyttä ulkoisiin palveluihin ilman valtuutettua suostumusta. Myöskään ulkoisen lähteen koodien käyttö ei ole sallittua, jos se ei tule erillisenä palveluna.

Lisäosat eivät saa sisältää mitään laitonta, epärehellistä eikä loukkaavaa sisältöä. Lisäosat eivät myöskään saa tehdä mitään laitonta. Laitonta sisältöä on muun muassa lisäosat, jotka luo tilejä väärennettyjen arvostelujen luomiseksi, tarjoaa lisää liikennettä sivustoille niiden kautta, jotka käyttävät samaa lisäosaa. Lisäosat eivät myöskään saa ottaa haltuun hallintapaneelia, eikä laittaa sinne liikaa turhia ilmoituksia. Päivitysketjuja, ilmoituksia, hälytyksiä saa näyttää hallintapaneelissa, mutta se ei saa haitata käyttäjän WordPressin hallintaa. Ilmoitukset on oltava suljettavia tai hylättäviä käyttäjän toimesta. (Wordpress, Wordpress, 2020)

4 VERKKOSIVULASKURIN TOTEUTUS

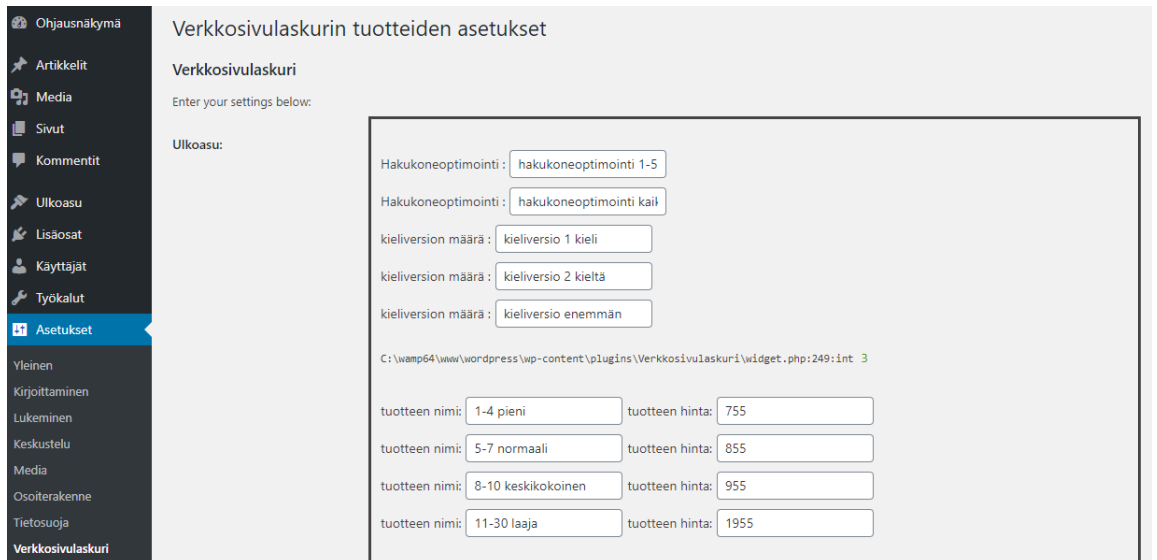
Toteuttaminen ja koodaustyö aloitetaan ensin vaatimusmäärittelyn tekemisellä lisäosan tilaajan kanssa. Vaatimusmäärittelyssä käydään läpi haluttuja toiminnallisuuksia sekä ulkonäköön liittyviä asioita. Tärkeimpänä oli helposti muokattavat valinnat sekä teemaan sopiva ulkoasu. Lisäosan käyttöönoton helpottamiseksi käytin WordPressin omaa shortcode API:a, jolla lisäosan voi lisätä halumaansa kohtaan. Toiminnallisuuksien määrittelyn jälkeen piti miettiä, miten kyseiset toiminnallisuudet olisi paras ja turvallisin toteuttaa. Koodikieliksi toteuttamisessa valitsin verkkosivukehitykseen oleelliset html, css ja javascript kielet, sekä WordPressiin pakollisen PHP kielen. PHP kieltä vaaditaan lisäosan toteutuksessa käytettäviin koukkuihin sekä tietokanta kyselyihin. Javascript kielellä toteutin laskurin toiminnallisuuksia sekä laskin tuotteiden hintoja. Html ja css kielellä toteutin itse laskurin ulkoasun.

4.1 Lisäosan hallinta

Backend on lisäosan hallintapuoli, joka ei näy sivulla kävijöille. Backend koostuu lisäosassa komponenteista, joiden tietoja käytetään asiakasnäkymässä. Hallintapuoli on tärkeä, koska ilman sitä joutuisi lisäosan hallinnoija käsin koodaamaan lisäosaan uudet hinnat sekä nimet, eikä uusien tietojen lisääminen tuotteille onnistuisi yhtä helposti. Lisäosan vaatimuksena oli helppo muokattavuus, joka toteutettiin lisäosan hallintapuolen luonnilla.

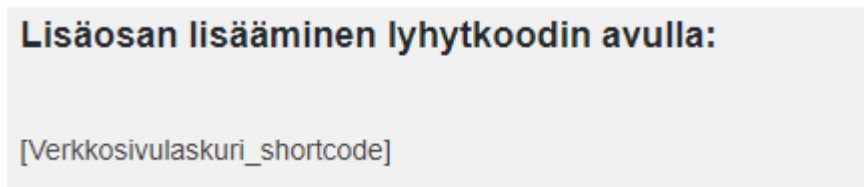
Muokattavuuden toteutin hallintapaneeliin lisäämällä valikoita, joihin hallinnoija voi lisätä tuotteille nimet sekä hinnat. Nimet ja hinnat käyttävät WordPress järjestelmän tietokantaa. Tietokantaan lisätään jokaiselle komponentille oma "option", joka on WordPress hallintajärjestelmän sisäinen koukku eli hook. Koukulla voi helposti hakea tietoa kyseisistä optioneista sekä lisätä uusia ja muokata valmiiksi olevia.

Lisäosan hallintaan pääsee asetukset-valikon Verkkosivulaskuri kohdasta (kuva 23). Kyseinen näppäin aukaisee ruudun, joka sisältää kaikki mahdolliset tuotteiden muokkaukset lisäosaa varten. Lisäosan tuotteet sisältävät hinnat sekä nimet. Nimiä on mahdollista muokata jokaiselle, mutta hinnat lasketaan suoraan ensimmäisen tuotteen valinnasta.



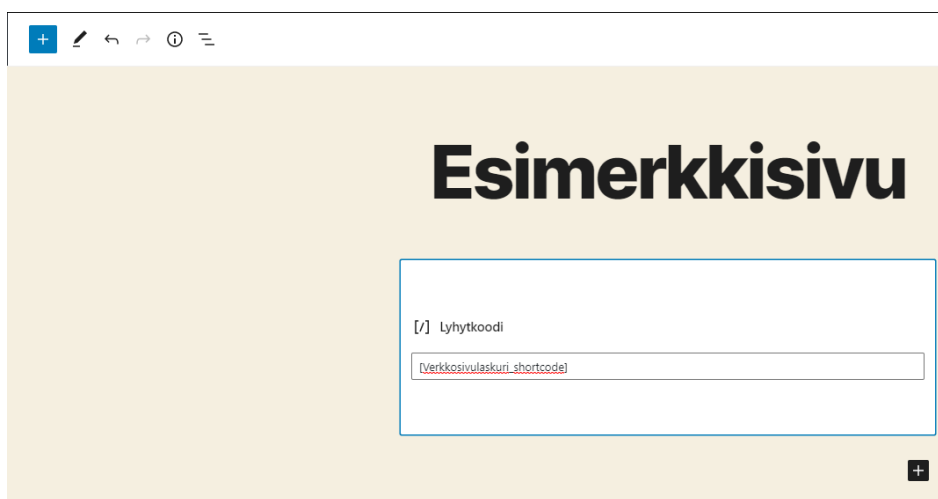
KUVA 23 Lisäosan asetukset-valikko

Hallintapaneeli sisältää myös lyhytkoodin. Lyhytkoodin voi kopioida suoraan hallintapaneelistä ja lisätä omalle sivulle tekstieditoriin tai vimpainalueeseen, näin lisäosan saa helposti ja vaivattomasti lisättyä haluttuun kohtaan verkkosivulla (Kuva 24).



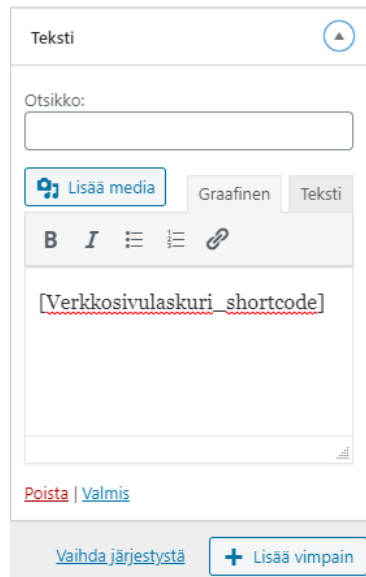
KUVA 24 Lyhytkoodi

Sivulle lisäosan voi lisätä suoraan tekstieditorin kautta (kuva 200) tai sitten teeman vimpaimien kautta (kuva 25). Tekstieditorissa lyhytkoodia lisätessä kannattaa käyttää WordPressin block editoria, joka sisältää lyhytkoodi vaihtoehdon. Lyhytkoodi blockissa lyhytkoodi toimii varmemmin ja se on helpompi lisätä.



KUVA 25 Lyhytkoodi tekstieditoriin

Vimpainalueelle lisätessä, tekstivimpain toimii ja lyhytkoodin voi siihen suoraan kopioida. Vimpainalueelle lyhytkoodin laittaminen mahdollistaa lisäosan lisäämisen esimerkiksi sivupalkkiin tai sivun alareunan footer alueelle (Kuva 26).



KUVA 26 Vimpain lyhytkoodi

Lyhytkoodit on tarkoitettu käytettäväksi WordPressin sivuilla ja vimpainalueissa, mutta joskus voi tulla tarve käyttää niitä myös teeman sisällä (wpBeginner, 2019). WordPress mahdollistaa tämän ja voit suoraan lisätä teeman tiedostoon php koodin, joka ajaa lyhytkoodin (Kuva 27).

```
echo do_shortcode("[Verkkosivulaskuri_shortcode]");
```

KUVA 27 lyhytkoodi php

4.2 Tietokanta

Lisäosa käyttää WordPressin omaa tietokannassa olevaa options taulua. Tauluun lisätään jokainen tuote. Jokaisella tuotteella on oma option_name sekä option_value (kuva 28).

option_name	option_value
hakukone_1	hakukoneoptimointi 1-5
hakukone_2	hakukoneoptimointi kaikille

KUVA 28 Tietokanta taulu

Hallintapaneelissa olevat ruudut, joihin käyttäjä lisää tietoa (kuva 29). Ruudut sisältävät kyseiset option_name:t. Option name:n kautta ruudun tieto lisätään kyseisen option_name:n option_value:en. Eli jos lisätään ruutuun, jonka value on hakukone_1 tietoa, se tieto menee kyseisen nimen "option_value" kohtaan (Kuva 28). Kuvassa hakukone_1 ruutuun on lisätty tieto "hakukoneoptimointi 1-5".

tuotteen nimi:	<input type="text" value="1-4 pieni"/>	tuotteen hinta:	<input type="text" value="755"/>
tuotteen nimi:	<input type="text" value="5-7 normaali"/>	tuotteen hinta:	<input type="text" value="855"/>

KUVA 29 Tietojen syöttäminen

4.3 Tiedon lisääminen ja muokkaaminen

Uuden tiedon lisääminen tietokantaan tapahtuu "update_option" funktiolla. Tämä funktio päivittää tiedon tietokannassa, jos se siellä on, mutta osaa myös lisätä uuden optionin tietokantaan. Ensimmäisen käytön yhteydessä, kun tuotteille annetaan tiedot ja ne tallennetaan, update_option funktio lisää uudet tiedot tietokannan options tauluun. (kuva 30).

```
update_option( 'test_'. $dd.', $this->options['test'.'. $dd.']) );
```

KUVA 30 Tuotteen lisäys

Tietojen tallennukseen käytetään WordPressin omaa "submit_button ()" näppäintä (kuva 31). Nappi kutsuu "Ulkoasu_settings" funktiota, joka sisältää kyseisten laatikoiden koodit. Tiedon lisäyksen vaiheessa lisätyt tiedot tarkistetaan, ennen tietokannan kutsua ja lisäystä.

```
submit_button( __('Lataa tiedot'), 'Ulkoasu_settings' );
```

KUVA 31 Tiedon tallentaminen

Tietoja voi ylikirjoittaa laatikoista ja tallentaa, jolloin tietokannassa tiedot muuttuvat. Tiedon muuttuessa tietokannassa myös käyttäjänäkymässä tiedot muuttuvat.

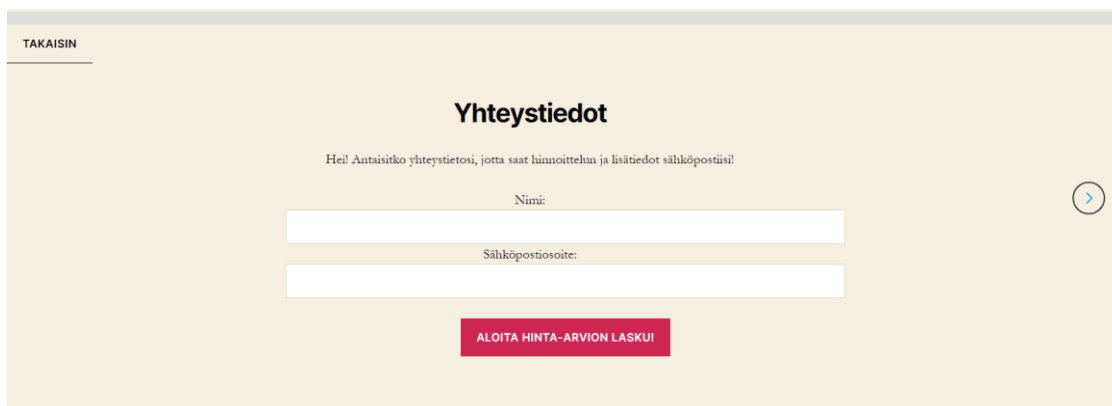
4.4 Käyttäjä näkymä

Käyttäjän näkymä koostuu valinta ruuduista, yhteystietojen lisäyksestä, valikon vaihtonäppäimistä sekä edistymistä kuvaavasta laatikosta. Ulkoasusta luotiin mahdollisimman yksinkertaisen sekä selkeän. Tuotteiden tiedot lisäosa hakee WordPressin options taulusta "wpdb" WordPress luokan avulla. Tämä luokka toimii rajapintana tietokannalle. Haku toimii SQL Queryllä (kuva 32).

```
$wpdb->get_results( "SELECT * FROM " );
```

KUVA 32 Tiedon haku

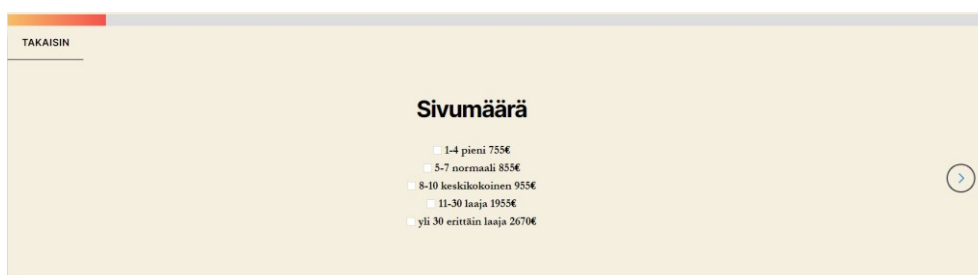
Ensimmäisessä vaiheessa käyttäjä antaa yhteystiedot eli nimen sekä sähköpostiosoitteen (kuva 33). Yhteystietojen syötön jälkeen käyttäjä voi aloittaa verkkosivujen hintojen laskennan.



KUVA 33 Laskurinäkymä

Yhteystietojen syöttö tarkistetaan WordPressin tarkistus hookilla ja lisätään sessioniin. Näitä tietoja käytetään myöhemmässä vaiheessa sähköpostin lähetyksessä. Syötteen tarkistaminen on tärkeää, ettei kukaan pysty "injektoimaan" tietokantaan tietoa syötön yhteydessä tai poistamaan tietoa tietokannasta.

Yhteystiedot sivun jälkeen tulee itse valinnat. Tämä ruutu on tehty selkeäksi, eikä käyttäjän tarvitse muuta kuin valita haluamansa. Ensimmäisessä valinnassa sisältösivujen määrä vaikuttaa seuraavien valintojen hintoihin (kuva 34).



KUVA 34 Valintaruutu 1

Yksi haluttu ominaisuus, joka selkeytti käyttäjälle etenemistä oli "Progress Bar". Progressbar on valikkoruudun yläreunassa oleva oranssi palkki, joka havainnollistaa käyttäjälle etenemisen. Palkki etenee sitä mukaan, kun käyttäjä liikkuu valinnoissa eteen tai takaisin päin. Etenemistä havainnollistava palkki on luotu javascript funktiolla. Funktiota kutsutaan jokaisella painalluksella, eteenpäin kun mennään oranssi palkki kasvaa sekä taakse päin kun mennään, se pienenee saman verran. Etenemisen seuranta on tärkeää, sillä käyttäjät haluavat tietää, kuinka monta valintaa joutuu tekemään ennen kokonaishinnan tietämistä.

4.5 Tuotteiden hintojen lasku

Kaikille tuotteille ei voitu antaa suoria hintoja, koska ne muodostuvat ensinmäisen valinnan mukaan. Ensimmäisessä osiossa käyttäjä valitsee sisältösivujen määrän. Määrä vaikuttaa siihen kuinka paljon seuraavat valinnat maksavat. Esimerkiksi pieni sisältösivujen määrällä myös kieliversiot maksavat vähemmän.

Hintojen laskun toteutin javascript kielellä. Funktio tarkistaa minkä tuotteen käyttäjä on ensinmäisessä ruudussa valinnut. Valinnan tarkastamisen jälkeen funktio asettaa seuraavalle valinnalle v-luen eli hinnan (kuva 35).

```
if ($('#input#Ulkoasu_item_1').is(':checked')) {
    document.getElementById("optio1").value = "380";
}
```

KUVA 35 Hinnan laskenta

4.6 Valintojen tallennus

Jokainen käyttäjän valinta säästetään javascript muuttujaan. Yhdessä muuttujassa on jokaisen valinnan nimi. Toinen muuttuja sisältää, jokaisen valitun tuotteen hinnan sekä niiden yhteen lasketun hinnan. Nämä tiedot säästetään, jotta ne voidaan myöhemmässä vaiheessa lähettää käyttäjälle sähköpostiin, sekä näyttää loppuruudussa (kuva 36).

```
$("#checkbox").change(function() {
    var arr = $("#checkbox:checked").map(function() { return $(this).next().text(); }).get();
    $("#cs-input").text(arr.join(', ' + '\n') + '\n');
});
```

KUVA 36 Tiedon lisääminen loppuruutuun

5 VERKKOSIVULASKURIN RAKENNE

Lisäosa koostuu PHP tiedostoista, tyylitiedostoista sekä javascript tiedostoista. Itse lisäosa on rakennettu PHP- kielellä, kuten muutkin lisäosat WordPress sisällönhallintajärjestelmässä. Lisäosan PHP-tiedostot koostuvat hallintapaneelin luonnista sekä käyttäjän näkymästä.

Lisäosan päätiedosto sisältää kommentoidun osion, joka kertoo WordPressille yleistä tietoa lisäosasta. Tässä kohdassa on annettu lisäosan nimi, lisätiedot, version numero sekä tekijä (kuva 38). Alussa estetään myös pääsy suoraan tiedostoon käsiksi selaimen kautta tietoturvasyistä.

```
<?php
    /*
    Plugin Name: Verkkosivulaskuri
    description: verkkosivujen laskurilisäosa
    Version: 1.0
    Author: Jani Kellomäki
    */

    if ( !defined( 'ABSPATH' ) ) exit;
```

KUVA 38 Kommenttiosio ja tietoturva

Tiedoston alussa kutsutaan myös vaadittavat css sekä javascript tiedostot, joita tarvitaan lisäosan toiminnallisuuksia varten. Lisäosassani ne olivat funktio.js, josta tulee javascript koodit sekä settings.css, josta tulee hallintapuolen ulkoasun koodit (kuva 39).

```
function my_theme_scripts() {
    $ok = plugin_dir_url( __FILE__ );
    wp_enqueue_script( 'my-great-script', $ok . '/javascript/funktio.js', array( 'jquery' ), '1.0.0', true );
}
add_action( 'wp_enqueue_scripts', 'my_theme_scripts' );

function jk_enqueue_custom_admin_style() {
    wp_register_style( 'custom_wp_admin_css', plugin_dir_url( __FILE__ ) . 'settings.css', false, '1.0.0' );
    wp_enqueue_style( 'custom_wp_admin_css' );
}
```

KUVA 39 funktio- ja tyylitiedoston lisäys

WordPress itsessään oli tullut tutuksi töiden kautta, mutta aikaisempaa kokemusta lisäosan koodauksesta ei ollut. Lisäosan kehittäminen vaati paljon itseopiskelua, mutta WordPress lisäosien koodausta varten on paljon kattavia dokumentaatiota kehittäjille. WordPress sisältää myös paljon valmiita funktioita, joiden käyttämisen opetteluun meni aikaa. Kyseiset funktiot ja niiden käyttö oli hyvin dokumentoitu sekä selkeesti kerrottu käyttö. Aihe oli hyvä ja se toi uutta näkökulmaa WordPress sisällönhallintajärjestelmästä. Paljon uutta tuli opittua niin lisäosan rakentamisesta kuin WordPressin toiminnasta yleisesti. Nämä asiat auttavat tulevaisuuden työssä WordPressin parissa.

Alkuun haasteena oli, kuinka saadaan luotua lisäosalle hallintapuoli. Hallintapuolen koodauksessa meni itsessään eniten aikaa WordPress funktioiden dokumenttien lukemisessa ja opiskelussa. Oikeiden funktioiden löytäminen ja käyttämisen oivaltaminen sai hallintapuolen luonnin sujumaan. Itse hallintapuolen lisäksi ongelmana tuli tuotteiden hintojen laskenta. Vain yhteen tuotteeseen annetaan itse hinta, joka määrää seuraavien tuotteiden hinnat. Tämän ongelman ratkaisuun tein monta tapaa, mutta testauksissa tuli ongelma kohtia, jotka piti korjata useaan otteeseen. Hintojen laskuun löysin lopulta hyvän tavan, jolla tuotteille saatiin hinnat.

Alun haasteiden jälkeen, onnistuin kuitenkin omasta mielestä hyvin. Sain tehtyä halutut toiminnot lisäosaan, niin ulkoisesti kuin hallinta puolella. Lisäosan tuotteita on helppo muokata sekä käyttäjän on selkeä ja helppo laskea verkkosivujen hinnat lisäosalla. Lisäosan voi myös mutkattomasti lisätä mihin kohtaan sivua vain.

Lisäosa tehtyjen ominaisuuksien lisäksi vaatii vielä jatkokehitystä. Lisäosa kuitenkin on helposti jatkokehitettävissä ja siihen voi nopeasti lisätä uusia ominaisuuksia tarpeen tullen. Lisäosan käyttöä tutkitaan ja mietitään, millä tavoin sen saisi vielä käyttäjä ystävällisemmäksi. Ulkoasua kehitetään tulevalle sivulle sopivaksi sekä ominaisuuksia tarpeen tullen lisätään.

LÄHTEET JA TUOTETUT AINEISTOT

Google. (8. 12. 2020.). *support.google.com*. Noudettu osoitteesta <https://support.google.com/adsense/answer/32755?hl=fi>.

Leiniö, T. (27. 12. 2013). *Kuinka asennat WordPressin webhotelliin?* Haettu 30. 9. 2020 osoitteesta [wpopas.fi](https://wpopas.fi/kuinka-asennat-wordpressin-webhotelliin/): <https://wpopas.fi/kuinka-asennat-wordpressin-webhotelliin/>

Rashty, D. (5. 11. 2020). *cminds*. Noudettu osoitteesta <https://www.cminds.com/ultimate-guide-wordpress-statistics%E2%80%A8%E2%80%A8/>.

Staff, E. (22. 6. 2012). *How to Add Your Plugin to the WordPress Plugin Directory*. Haettu 30. 9. 2020 osoitteesta www.wpbeginner.com: <https://www.wpbeginner.com/plugins/how-to-add-your-plugin-to-the-wordpress-plugin-directory/>

Wordpress. (9. 12. 2020). *developer.wordpress*. Noudettu osoitteesta <https://developer.wordpress.org/plugins/security/securing-output/>.

Wordpress. (9. 12. 2020). *developer.wordpress*. Noudettu osoitteesta <https://developer.wordpress.org/plugins/security/securing-input/>.

Wordpress. (10. 10. 2020). *developer.wordpress.org*. Haettu 10. 10. 2020 osoitteesta Wordpress: <https://developer.wordpress.org/themes/advanced-topics/child-themes/>

Wordpress. (20. 10. 2020). *developer.wordpress.org*. Haettu 20. 10. 2020 osoitteesta Wordpress: <https://developer.wordpress.org/plugins/security/securing-output/>

WordPress. (14. 10. 2020). *developer.wordpress.org*. Haettu 14. 10. 2020 osoitteesta <https://developer.wordpress.org/plugins/hooks/>: <https://developer.wordpress.org/plugins/hooks/>

Wordpress. (7. 11. 2020). https://codex.wordpress.org/WordPress_API%27s. Haettu 7. 11. 2020 osoitteesta [codex.wordpress](https://codex.wordpress.org).

Wordpress. (14. 12. 2020). *Options_API*. Noudettu osoitteesta https://codex.wordpress.org/Options_API.

Wordpress. (30. 9. 2020). *Roles and Capabilities*. Haettu 30. 9. 2020 osoitteesta [Wordpress.org](https://wordpress.org): <https://wordpress.org/support/article/roles-and-capabilities/>

Wordpress. (7. 12. 2020). *Wordpress*. Noudettu osoitteesta <https://developer.wordpress.org/plugins/wordpress-org/detailed-plugin-guidelines/>.

wpBeginner. (18. 9. 2019). *www.wpbeginner.com*. Noudettu osoitteesta <https://www.wpbeginner.com/wp-tutorials/how-to-add-a-shortcode-in-wordpress/>.

Vuoristo, V. (1. 11. 2009). *Verkkosivut ja sisällönhallinta WordPress-järjestelmällä*. Haettu 30. 9. 2020 osoitteesta theseus.fi: https://www.theseus.fi/bitstream/handle/10024/5271/Vuoristo_Vesa.pdf?sequence=1