

**Woocommerce-pohjaiselle verkkokaupalle räätälöidyn teeman
kehitys**



Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutus, Hämeenlinnan korkeakoulukeskus
syksy, 2021

Trung Ta

TIIVISTELMÄ

Opinnäytetyön tavoitteena oli toteuttaa tyhjästä Wordpress-teema, joka on täysin yhteensopiva Woocommerce-pohjaisen verkkokaupan kanssa ja Omari oli valittu teeman nimeksi. Opinnäytetyössä selvitettiin myös kustomoidun teeman hyödyllisyyttä www-sivuja kehittävän yrityksen kannalta ja pohdittiin aloitusteeman valintaa vertaamalla Underscore-pohjateemaan. Opinnäytetyön toimeksiantaja oli Suomen nettimyynti Oy, joka toimii verkkokaupan alalla.

Opinnäytetyö oli toiminnallinen, jonka käytännön osa käsitteli pääosin kustomoidun teeman toteutusta. Teematoteutuksen tekniikka koostui Wordpressin antamasta ohjeistuksesta, omista ohjelmointikokemuksista ja internetlähteistä.

Työn tuloksena syntyi kustomoitu Omari-teema, jota voidaan käyttää pohjana verkkokaupan käyttöliittymän suunnittelussa. Tuloksen lisäksi tutkimuksessa havaittiin, että Underscore-aloitusteema toimi paremmin, eikä saavutettavuutta ja hakukoneoptimointia käsittelevään tutkimuskysymykseen pystytty vastaamaan. Tämä johtui opinnäytetyön pääaiheesta, sillä pyydettiin että luodaan räätälöityä teemaa tyhjästä, jota käytetään pohjana jatkokehitykseen, eikä tutkimustyöstä käsitteli CSS- ja HTML-seikkoja. Työn tulokset vastaavat kuitenkin täysin toimeksiantajan odotukseen, sillä teemaa voitaisiin käyttää verkkokauppaan pohjana tulevaisuudessa.

Avainsanat kustomoitu teema, woocommerce, wordpress-teema

Sivut 39 sivua ja liitteitä 1 sivu

Author Trung Ta

Year 2021

Subject Custom theme development for Woocommerce-based e-commerce

Supervisors Lasse Seppänen

ABSTRACT.

The aim of the thesis was to implement a Wordpress theme from scratch that is fully compatible with Woocommerce-based e-commerce and Omari was chosen for theme name. The thesis also investigated the usefulness of a custom theme for a company developing web pages and comparing its starter theme with Underscore base theme. The client of the thesis was Suomen nettimyynti Oy, which is specialized in ecommerce branch.

The thesis was functional mainly dealt with the implementation of the customized theme. The theme implementation technique was implemented following the instructions provided by Wordpress, some technique came from own programming experiences, and from Internet sources.

The result of this thesis was a fully functional Omari theme that can be used as a basis for designing the webshop's user interface. In addition, the study found the superiority of the Underscore starter theme, also in terms of accessibility and search engine optimization could not be implemented as one of the research questions had posed. This was due to the main topic of the thesis, where to create a customized theme from scratch, which is used as a web template for further development, and the research work did not deal with CSS and HTML. Lastly the result of the work has fully met the client's expectations, as the theme could be used in their ecommerce website in the future.

Keywords custom theme, woocommerce, wordpress theme

Pages 39 pages and appendices 1 page

Sanasto

Woocommerce	muokattava Wordpressiin rakennettu avoimen lähdekoodin verkkokauppa-alusta
Aloitusteema	tyhjä teemamalli, jossa on yleensä tarvittavat perussuunnittelut tai ei olleenkaan, oman teeman kehittämiseksi.
Omari-teema	opinnäytetyön lopputyö eli Wordpress-teema, jota kehitetään tyhjästä omaa aloitusteemaa käyttäen
Lisäosat	ohjelmisto, joka on kirjoitettu PHP-ohjelmointikielellä ja sisältyy lukuisia Wordpress-sivuille käytettäviä toimintoja ja ominaisuuksia
GPL	General Public License on ilmainen tekijäoikeuslisenssi ohjelmistoille ja muille teoksille
Grid-järjestelmä	Bootstrap-kehiksen ruudukkojärjestelmä, joka koostuu kahdestatoista sarakkeista
WCAG	Web Content Accessibility Guidelines eli verkkosisällön esteettömyysohjeet, joista selittävät kuinka verkkosisältöjä saadaan vammaisten saatavuutta helpommin.
SEO	Search Engine Optimization on verkkosivujen parantamisprosessi, josta nostaa yritykseen liittyvät tuotteet ja palvelut hakukoneissa
Branch	Github-alustaan kuuluva toiminto, jonka avulla voidaan kehittää uudet ominaisuudet, korjata virheet pelkäämättä ytimen koodin menettämistä
Teemakirjasto	Wordpressin teemahakemisto, johon voidaan lisätä ja ladata uusia teemoja

Sisälllys

TIIVISTELMÄ.....	1
ABSTRACT.	2
Sanasto	3
1 Johdanto	7
2 Avoin lähdekoodi.....	8
2.1 Sisällönhallintajärjestelmät.....	8
2.2 Wordpress-avoin lähdekoodi.....	9
2.3 Woocommerce-avoin lähdekoodi.....	10
3 Wordpress-teeman kehitysjärjestelmät.....	11
3.1 Bootstrap -sovelluskehys	11
3.2 WAVE-saavutettavuuden tarkastustyökalu	13
3.3 Yoast SEO –hakukoneoptimoinnin lisäosa	14
3.4 Tietoturvan seikat	16
3.5 Git–versionhallinta	17
4 Erilaiset räätälöityä teemaa rakentavat pohjat.....	19
4.1 Joints WP.....	19
4.2 Bones-teema	20
4.3 Underscore-aloitusteema	20
5 Räätälöidyn teemapohjan toteutus	22
5.1 XAMPP-Asennus.....	22
5.2 Wordpress.....	23
5.2.1 Wordpress-sivulle tietokannan luominen.....	23
5.2.2 Wordpress-asennus	24
5.2.3 Github-alustaan integraatio	25
5.3 Aloitusteeman luonti ja aktivointi.....	26
5.3.1 Otsake-tiedosto (header.php) ja alaosio (footer.php).....	27
5.3.2 Tyylitaulukkojen järjestely	28
5.3.3 Bootstrap-kehiksen lisäys.....	28
5.3.4 Valikkojen luonti.....	29
5.3.5 Wordpress The Loop	30
5.3.6 Sivumallit Page Templates	31
5.4 Omari-aloitusteeman integraatio Woocommerce-verkkokauppaan	31
5.4.1 Woocommerce-verkkokaupan asennus.....	31

5.4.2	Woocommerce-tuen ilmoittaminen	32
5.4.3	Woocommerce-pohjan muunnos	32
5.5	Omari-teeman tietoturvat ja käännös	33
5.5.1	Tietoturvat.....	33
5.5.2	Käännös	33
5.5.3	Omari-teeman hallinta	34
5.6	Omari-teeman lisäys Wordpress-teemakirjastoon.....	35
6	Tulokset	37
7	Yhteenveto	39
	Lähteet.....	41

Kuvat, ohjelmakoodit ja taulukot

Kuva 1	Sisällönhallintajärjestelmien markkinaosuus (vihreä palkki) ja harmaasävyllä lasketaan ei-sisällönhallintajärjestelmää käytettävän sivuston osuutta 11/2021	9
Kuva 2	Wordpressin valiooteema Twenty Twenty-One ja sen tiedostot.....	9
Kuva 3	Woocommerce-projektin esittely Githubissa.....	10
Kuva 4	Bootstrapin CSS:n- ja Javascriptin esimerkkikoodeja	12
Kuva 5	Saavutettavuuden aloitus kirjoittamalla sivun osoitetta tyhjään kenttään	14
Kuva 6	Yoast-lisäosan haun ulkoasun muokkaus artikkelille	16
Kuva 7	Haavoittuneiden www-sivujen määrä eri sisällönjärjestelmissä vuonna 2017...17	
Kuva 8	Github Desktop -käyttöliittymä	18
Kuva 9	JointsWP malliversio	19
Kuva 10	Bonesin koodikirjasto	20
Kuva 11	XAMPP-testiympäristön eri asennusvaihtoehtoja.....	22
Kuva 12	XAMPP-kehitysympäristön aktivointia	23
Kuva 13	Omari-tietokannan luominen	24
Kuva 14	Wordpress-asennuksen näyttö.....	24
Kuva 15	Tietokannan määritelmä.....	25
Kuva 16	Projekti on ladattu onnistuneesti Github-palvelimeen.	26
Kuva 17	Omari-teeman perustiedot.....	27
Kuva 18	Otsakkeen- ja alaosion lisäys	27
Kuva 19	Rivittää tyylitiedostoa wp_enqueue_style()-funktioilla	28

Kuva 20 Omari-valikon asetukset	30
Kuva 21 Etusivun ja blogin erottaminen	30
Kuva 22 Woocommerce-verkkokaupan tuotemallit	32
Kuva 23 Omarin tarvittavat käännösteksti Poedit-ohjelmistossa	34
Kuva 24 Theme Check ilmoittaa käyttäjälle korjattavat virheet.	35
Kuva 25 Ilmoitus teeman läpäisemisestä Wordpressin tarkistuksessa.....	36
Kuva 26 Omari-teemasta perustuvasta etusivu	38

Liitteet

Liite 1	Aineistonhallintasuunnitelma
---------	------------------------------

1 Johdanto

Olen työskennellyt verkkokauppiaana jo useita vuosia, joten Wordpress- eli sisällönhallintajärjestelmän kanssa minulla on melko paljon kokemuksia verkkosivujen ylläpitämisessä, mutta en ole vielä perehtynyt järjestelmiin syvällisesti, varsinkaan avoimiin lähdekoodeihin. Suomen Nettimyynä Oy on päätoimialtaan laaja postimyynä ja verkkokauppa ja yrityksen nykyisessä verkkokaupassa käytetään Flatsome-teemaa oletuksena, joka ei anna riittävää tilaa jatkokehitykselle ja luovuudelle. Tämän takia yritys on antanut minulle toimeksiannon Wordpress-teeman työhön asti, ja se on tämän opinnäytetyön pääaihe.

Lopputeeman pitää olla täysin yhteensopiva Woocommerce-lisäosan kanssa, ja sen lisäksi teema toimii responsiivisesti kaikissa laitteissa Bootstrap-sovelluskehystä hyötyen. Lisäksi saavutettavuudesta ja hakukoneoptimoinnista on säädettävä niin, että sivuston sisällöt ovat saatavilla kaikille ihmisille ja niitä sisältöjä löydetään jokaisessa hakukoneissa. Käydään myös teeman tietoturvasuutta ja sen jatkokehitystä läpi, varsinkin tietoturvan osalta voidaan jakaa pienempiin osiin, koska siinä on monia tapoja turvata Wordpress-sivustoa. Projektia tehdessäni tallennan ja synkronoin kaikki koodipätkät ja kansiot Github-versiohallintajärjestelmään, näin työ voidaan jatkaa opinnäytetyöteon jälkeen.

Teemakehityksen yhteydessä pyrin vastaamaan seuraaviin tutkimuskysymyksiin, joiden avulla voidaan ottaa selväksi lopputyön hyödyllisyyttä, eroja muihin aloitusteemoihin. Opinnäytetyössä selvitetään myös saavutettavuuden- ja hakukoneoptimoinnin vaikutusta sivuston sijoitukseen hakukoneissa. Tutkimuskysymykset ovat:

- Miten ohjelmistokehitykseen erikostuva yritys hyödyntää itsetehtyä Wordpress-teemaa kehittäkseen asiakkaiden nettisivuja?
- Mitä eroja on itsetehdyllä aloitusteemalla ja Underscore-alustaa käyttämällä aloitusteemalla?
- Miten saavutettavuus ja hakuoptimointi nostaa sivuston sijoitusta hakukoneissa?

2 Avoin lähdekoodi

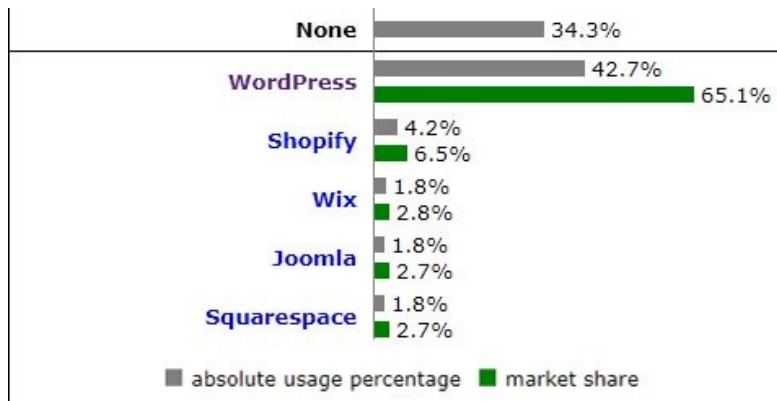
Tästä lopputyö koostuu pääosin Wordpress- ja Woocommerce-järjestelmistä, jotka toimivat avoimina lähdekoodeina. Seuraavissa alaluvuissa käydään läpi mitkä ovat sisällönjärjestelmät ja Wordpress- ja Woocommerce-avoinlähdekoodit.

2.1 Sisällönhallintajärjestelmät

Sisällönhallintajärjestelmä on tietojärjestelmä, joka käsittelee julkaisujen luomista, tallentamista, noutamista, kuvauksia, kommentteja ja ulkoasua (Williams et al., 2015). Järjestelmässä pystytään erotelemaan www-sivujen sisältö, rakenne ja ulkoasu toisistaan, ja myös päivitykset voidaan suorittaa suorassa selaimissa. Sisällönhallintajärjestelmän helppokäyttöisyyden myötä käyttäjät voivat muokata sivujen sisältöjä ja ulkoasua ilman koodausosaamista.

Kirjoitushetkellä kolme käytetyintä sisällönhallintajärjestelmää on Wordpress (65,1 %), Shopify (6,5 %), Wix (2,8 %) (Usage Statistics and Market Share of Content Management Systems, September 2021, n.d.). Wordpress on yltänyt korkeimpaan sijoitukseen tai sijoittunut korkeimmalle jo useita vuotta, perusteena mm. alustan käyttäjänystävällisyydestä, helppoudesta ja edullisuudesta. Lisäksi Wordpress-alustan käyttäjät ovat laajoja bloggaajista pienyrittäjiin, myös tavallisiin ihmisiin, jotka tarvitsevat pienimuotoiset www-sivut esim. harrastuksille tai seuroille. Shopifyn ja Joomla:n sidosryhmät taas ovat pääosin verkkokauppiaita ja suuryrityksiä, joiden verkkosivuillaan vaaditaan suuria RAM-muistia ja näiden sisällönhallintajärjestelmien käyttöönotto on yleensä maksullinen, kun taas Wordpress on ilmainen. Huippuaseman takia Wordpress-järjestelmä on tullut verkkohyökkäyksen ykköseksi kohteeksi esim. tilaston mukaan yli 40000 Alexassa huipulla listattua sivua on altistunut kyberhyökkäykseen (Jerkovic & Sinkovic, n.d.). Tämän korjaamiseksi Wordpressin pitää julkaista alustalleen päivitykset säännöllisesti.

Kuva 1 Sisällönhallintajärjestelmien markkinaosuus (vihreä palkki) ja harmaasävyllä lasketaan ei-sisällönhallintajärjestelmää käytettävän sivuston osuutta 11/2021



2.2 Wordpress-avoin lähdekoodi

Wordpressin rakenne koostuu pääosin PHP-tiedostoista ja MySQL-tietokannasta, ohjelmisto on kehitetty alkuperäisesti Matt Mullenweg ja Mike Little vuonna 2003 (Williams et al., 2015). Wordpress-alusta toimii avoimena lähdekoodina, jotka on lisensoitu GPL:n alla eli General Public License, jonka avulla muilla sovelluskehittäjillä on oikeudet Wordpress-alustan muokkaamiseen ja kopioimiseen alkuperäisiä tekijöiden nimiä huomioiden. Tämän takia Wordpress-alusta on kehittynyt suureksi ja tullut yksi suosituimmaksi sisällönhallintajärjestelmäksi, sen lisäksi markkinassa on yli tuhansia Wordpressin lisäosia kehittäjiä, joiden rakenteet sitovat ja perustuvat Wordpressin ytimeen koodirakenteeseen.

Kuva 2 Wordpressin valioiteema Twenty Twenty-One ja sen tiedostot

```

/*
Theme Name: Twenty Twenty
Text Domain: twentytwenty
Version: 1.8
Requires at least: 4.7
Requires PHP: 5.2.4
Description: Our default theme for 2020 is designed to take full advantage of t
endless layouts using the group and column blocks. The centered content column
what your content will look like, even before you publish. You can give your si
on your site are automatically calculated based on the colors you pick, ensurin
Tags: blog, one-column, custom-background, custom-colors, custom-logo, custom-m
options, threaded-comments, translation-ready, block-patterns, block-styles, wi
Author: the WordPress team
Author URI: https://wordpress.org/
Theme URI: https://wordpress.org/themes/twentytwenty/
License: GNU General Public License v2 or later
License URI: http://www.gnu.org/licenses/gpl-2.0.html

All files, unless otherwise stated, are released under the GNU General Public
License version 2.0 (http://www.gnu.org/licenses/gpl-2.0.html)

This theme, like WordPress, is licensed under the GPL.
Use it to make something cool, have fun, and share what you've learned
with others.
*/

```


2.3 Woocommerce-avoin lähdekoodi

Woocommerce on yksi ladatuimmista Wordpressissa listatusta lisäosista, joka toimii verkkokaupan alustana ja helpottaa kaikenlaisien tuotteiden myymistä netissä.

Woocommerce-alusta toimii myös avoimena lähdekoodina, jonka kehittivät Mark Forrester, Magnus Jepsoni ja Adii Pienari vuonna 2007 (About WooCommerce - WooCommerce.Com, n.d.). Avoin lähdekoodi antaa muiden kehittäjien muokkaamaan koodirakenteensa, kirjoitushetkellä yli 350 kehittäjää on aktiivisesti sitoutunut tämän alustan kehittämiseen maailmaa ympäri (WooCommerce Features, n.d.). Woocommerce-koodikirjasto löytyy Githubista eli versiohallintajärjestelmästä (kuva 3) ja on täysin muokattava GPL:n lisenssillä, lisäksi Woocommerce-alustaan on kehitty ja luotu yli 400 lisäosaa, joiden avulla edistää verkkokaupan myyntiä.

Kuva 3 Woocommerce-projektin esittely Githubissa.

☰ README.md



license GPL-3.0-or-later
stable 5.9.0
downloads 160M
rating ★★★★★
Run CI failing
codecov 45%

Welcome to the WooCommerce repository on GitHub. Here you can browse the source, look at open issues and keep track of development. We recommend all developers to follow the [WooCommerce development blog](#) to stay up to date about everything happening in the project. You can also follow [@DevelopWC](#) on Twitter for the latest development updates.

If you are not a developer, please use the [WooCommerce plugin page](#) on WordPress.org.

Documentation

- [WooCommerce Documentation](#)
- [WooCommerce Developer Documentation](#)
- [WooCommerce Code Reference](#)
- [WooCommerce REST API Docs](#)

Reporting Security Issues

To disclose a security issue to our team, [please submit a report via HackerOne here](#).

Support

This repository is not suitable for support. Please don't use our issue tracker for support requests, but for core WooCommerce issues only. Support can take place through the appropriate channels:

- If you have a problem, you may want to start with the [self help guide](#).
- The [WooCommerce.com premium support portal](#) for customers who have purchased themes or extensions.
- [Our community forum on wp.org](#) which is available for all WooCommerce users.
- The [WooCommerce Help and Share Facebook group](#).

3 Wordpress-teeman kehitysjärjestelmät

Wordpress-teeman toteutus koostuu yleensä mallitiedoista, jotka ovat PHP-tiedostoja. Nämä tiedostoja sisältyy taas muita pienosia mm. mallitunnisteet, HTML- ja PHP-koodit, jotka tulevat yhteen muodostamaan ydintä teemaa. Näiden lisäksi teeman parantamiseksi voidaan käyttää kolmannen osapuolen lisäosia, joiden kautta parantamaan käyttökokemuksia sekä hienontaa teeman ulkoasua.

3.1 Bootstrap -sovelluskehys

Projektiin käytetään Bootstrap-sovelluskehystä, jonka avulla lopputeema toimii responsiivisesti kaikissa laitteissa. Bootstrap on verkkosovelluskehys, jonka perusteella kehittäjät voivat rakentaa verkkokäyttöliittymiä (Jakobus, 2018). Bootstrap toimii myös avoimena lähdekoodina, jonka kehittivät alun perin Mark Ottolla ja Jacob Thorntonilla. Kirjoitushetkellä Bootstrap on julkistanut uusimman versionsa v5.1.1, jossa sovelluskehyyksen pienvirheitä saatiin korjattua ja Bootstrapin dokumentaatiota parannettua. Bootstrap-sovelluskehys on nykyään kiistattomasti yksi kaiken aikojen suosituimmista projekteista Githubissa (Niska, 2014).

Halutaan että lopputeeman responsiivisuus on optimoitu niin, että sivusto näkyy parhaallaan kaikissa laitteissa television näytöstä puhelimen näyttöön. Kun opinnäytetyön tekeminen aloitettiin, mietitiin, käytetäänkö Bootstrapia lopulliseen työhön. Päädyttiin siihen tulokseen, että työhön kuuluu myös Bootstrap-kehystä, sillä sen helppokäyttöisyys ja hyvä dokumentaatio nostavat projektia toiselle tasolle. Bootstrapin käyttööntoon ei vaadi koodausta eikä muita monimutkaisia asennusvaiheita, vaan sovellus voidaan ladata Bootstrapin kotisivustaan ja latauksen jälkeen tulee selkeitä ohjeita miten sovellus asennetaan suoraan omalle kehitysympäristölle esim. Visual Studio -tekstineditoriin.

Kuva 4 Bootstrapin CSS:n- ja Javascriptin esimerkkikoodeja

```
✓ CSS
# bootstrap-grid.css
# bootstrap-grid.css.map
# bootstrap-grid.min.css
# bootstrap-grid.min.css.map
# bootstrap-grid.rtl.css
# bootstrap-grid.rtl.css.map
# bootstrap-grid.rtl.min.css
# bootstrap-grid.rtl.min.css.map
# bootstrap-reboot.css
```

Bootstrap on suunniteltu melko kaikille käyttäjille, jotka tavoittelevat helppokäyttösuutta ja yksinkertaisuutta responsiivisuudessa www-sivujen suunnittelussa. Sovelluskehystä käytettiin ja ladattiin yli miljoonaa kertaa ja tuhannet suuryritykset luottavat Bootstrapiin, mukaan lukien NASA, Walmart ja Bloomberg (Jakobus, 2018). Alustan asennettua käyttäjät pääsevät nopeasti projektin tekemiseen Bootstrapin CSS-kehystä hyötyen.

Lisäksi alustan Grid-järjestelmä nopeuttaa sivuston sarakkeiden suunnittelusta, näin voidaan määritellä eri kokoja erilaisille näytöille. Järjestelmä jakaa näyttöä kahteentoista sarakkeeseen esim. komennolla `'col-sm'` on tarkoitettu näytölle, joka on yhtä suuri tai pienempi kuin 576px, `'col-md'` 768px ja `'col-lg'` 992 px. Näiden oletuksien perusteella ruudunkokoja voidaan muokata vapaasti käyttäjien tarpeen mukaisesti, näin kehittäjät eivät tarvitse luoda ruudukkojärjestelmää alusta asti, mikä on ajanvietettä ja monimutkaista. Lisäksi Bootstrap sisältyy erilaisia valmiita HTML-elementtejä mm. otsikkoja, luetteloja, taulukkoja, painikkeita ja lomakkeita, joita on muotoiltu etukäteen, myös www-sivujen tiettyjä komponentteja esim. pudotusvalikkoja, painikeryhmiä, navigointipalkkeja on käsitelty ja muotoiltu CSS-tyylillä.

Bootstrap on kätevä juuri tähän lopputyöhön, sillä lopputeeman responsiivisuus toimii ja näkyy paremmalla tavalla, sen lisäksi voidaan säästää paljon aikaa responsiivisuuden suunnittelussa Bootstrapin Grid-järjestelmän avulla.

3.2 WAVE-saavutettavuuden tarkastustyökalu

Saavutettavuus on digipalvelun esteettömyyttä, jota käsitellään ja optimoidaan tässä lopputyössä. Esteettömyys on tarkoitukseltaan se, että kulku eri paikkoihin on esteetön, vaikka liikkuisi pyörätuolilla tai ei näkisi kunnolla (saavutettavuusvaatimukset.fi, 2021). Tämä periaate pätee verkkosivujen ja mobiilisovelluksen kehitykseen, sillä näitä digipalveluja suunnitellessa kehittäjien on syytä ottaa huomioon nämä saavutettavuusseikat, niin lopputuote vastaa WCAG-ohjeistukseen eli Web Content Accessibility Guidelines, suomeksi Verkkosisällön saavutettavuusohjeisiin. Saavutettavuus on taidetta rakentaa verkkosivuja käyttäjille, jolla on rajoitteita esim. fyysisissä, näössä, kognitiivisessa tai kuulossa (Palani, 2019), joten hyvin verkkosivujen tai mobiilisovelluksen optimointi parannettaisi kaikkien käyttäjien käytettävyyttä ja käyttökokemuksia verkkopalveluissa.

Lisäksi saavutettavuus parantaa yhdenvertaisuutta, sillä jokaisella on sama mahdollisuus käyttää digipalveluja, ja siksi siitä lisäksi rajoitteisten ihmisten itsenäisyyttä ja osallistua yhteiskunnan aktiviteetteihin useammin (saavutettavuusvaatimukset.fi, 2021).

Saavutettavuutta suunniteltaessa pitää ottaa huomioon seuraavaa kolme asiaa: tekninen toteutus, helppokäyttöisyys, sisältöjen selkeys ja ymmärrettävyys. Näitä ovat mm. HTML-standardia ja WCAG-ohjeistus on noudatettu, lähdekoodi on virheetön. Palvelun pitää toimii kaikissa laitteissa, myös avustavilla teknologioilla on plussa. Helppokäyttöisyyteen pyritään asentamaan verkkosivuja niin, että kaikki sivun toimintoja ja sisältöjä löytyy helposti, näin varmistaa käyttäjien saavan käyttää sivuja vaivattomasti. Lisäksi sivuston kielen käyttö on myös yksi tärkeimmistä faktoista, joka edistää saavutettavuutta, sivustoon kannattaa käyttää ymmärrettävää kielen käyttöä sekä selkokielen käyttö on suositeltu. Tekstin fontti on tarpeeksi suuri lukemiseen ja taustan abstraktitaso on selkeä, myös se, että sisältöjä tarjotaan kuvioina, videoina tai äänenä on lukijoille ja katsojille saavutettavaa.

Sivuston saavutettavuutta voidaan tarkistaa useilla työkaluilla, jotka ovat saatavilla verkossa. Tässä opinnäytetyössä käytetään WAVE-lisäosa eli Web Accessibility Evaluation Tool loppusivuston saavutettavuuden tarkistamiseen. WAVE on ilmainen verkkotyökalu, jolla havaita sivustosta syntyviä saavutettavia virheitä ja lisäosa on saatavilla osoitteesta <https://wave.webaim.org/>. Verkkosivujen saatettavuuden tarkastelu voidaan aloittaa kirjoittamalla sivujen osoitetta kenttään (kuva 5), niin tarkastus kestää hetken aikaa ennen

tuloksien näyttöä. Lisäksi WAVE tarjoaa erilaisille selaimille laajennukset, joita kiinnittää selaimien yläpalkkeihin, näin voidaan nopeuttaa saatettavuuden tarkastelua napsauttamalla WAVE-kuvaketta.

Kuva 5 Saavutettavuuden aloitus kirjoittamalla sivun osoitetta tyhjäan kenttään



Vaikka teeman suunnittelussa ei välttämättä tarvitse saatettavuuden tarkastelua, sivusto käyttöliittymälle taas saavutettavuus on olennaista ja tämän tarkistamiseksi WAVE-lisäosa on kätevä ja hyödyllinen monin tavoin.

Sivustolla ei käytetä tiedon ja toiminallisuuksien esittämiseen, sekä tekstin ja taustan kontrasti on riittävä. Standardiin pitää noudattaa lomakkeiden ja painikkeiden teksti, mutta logo ei välttämättä tarvitse noudattaa kontrastivaatimuksia. Normaalien tekstin osalta kontrastin pitää olla vähintään 4,5:1 ja ison tekstin puolelta 3:1 (*Värit Ja Kontrastit - Saavutettavasti.Fi, n.d.*).

Värien käyttö verkkosivulla on suositeltava, sillä WCAG-kriteeristö ei kiellä värikästä ulkoasua, tärkeintä on se, että heikkonäköinen ja värisokea henkilö saa hakea tiedon ja ymmärtää toimintaohjeet. Verkkosivun tekstin ja taustan välillä pitää olla riittävä tummuusero, näin heikkonäköiset henkilöt voivat tunnistaa tekstiä. Myös verkkosivujen painikkeiden täytyy noudattaa kontrastivaatimuksia esim. Lähetä- ja Tilaa-painikkeissa tekstin ja taustan tummuuskontrasti on riittävä (*Värit Ja Kontrastit - Saavutettavasti.Fi, n.d.*).

3.3 Yoast SEO –hakukoneoptimoinnin lisäosa

Hakukoneoptimointi eli Search Engine Optimization lyhennellä SEO viittaa sivustojen ja niiden sivujen optimointiin niin, että koko sivustoa nostetaan hakutuloksissa (Kent, 2020). Suosituimmat hakukoneet kuten Google ja Bing käyttävät botteja, jotka käyvät kaikki

sivuston sivuja läpi, joita indeksoidaan keräämällä tiedoilla. Tämän takia hakukoneoptimointia suunniteltaessa kannattaa ottaa huomioon sivuston tekstejä mm. otsikot, kuvaukset, tiedostonimet ja linkit. Kaikkea näitä tiedostoja tallennetaan ensisijaisesti hakukoneiden tietokantoihin ja niiden perustella päättää sivuston sijoituksia hakukoneen tuloksissa. (*What Is SEO / Search Engine Optimization?*, n.d.)

Hakukonenäkyvyys on periaatteessa ilmainen, mutta sivustojen sijoitusjärjestys riippuu sen yksityiskohtaisesta optimoinnista. Hakukoneoptimointi on myös jatkuva prosessi ja osa digitaalisista markkinoinneista, sillä ihmiset ovat laittaneet kyselyjä hakukoneisiin triljoonia kertoja vuosittain, joista löytyy useita kaupallisiin tarkoituksiin hakuja. Näin olleen verkkosivuun pääsee enemmän asiakkaita ja myynti olisi lähtenyt nousuun, jos sivusto on sijoitettu parempaan järjestykseen. (*What Is SEO / Search Engine Optimization?*, n.d.)

Hakukoneoptimointia suorittaessa seuraavat seikat otetaan huomioon, joista mm. tekniikka, sisältö ja maine. Sivuston laadukas koodi ja sen valitsema sisällönhallintajärjestelmä ovat olennaisia. Mikäli koodeja on kirjoitettu virheellisesti, niitä kirjataan hakemistoon ja hakukoneet olettavat kyseisten sivujen olevan vaarallisia, niin niiden sijoituksia laitetaan heti alemmaksi. Sisällön pitää olla runsasta, uutta ja aitoa, jotka vastaavat hakukoneiden oletuksiin asetuksiin. Hakukoneet arvostavat sivuston tuottamia aitoja sisältöjä, joten blogin lisäys sivustoon olisi plussa, myös blogiin pystytään tuottamaan jatkuvasti laadukkaita sisältöjä ja julkaisuja nostaa silti sivuston hakukonenäkyvyyttä. Hakukoneoptimoinnin parantamiseksi linkkivaihto on myös hyvä laittaa tarkasteltavaksi, koska se on yksi tärkeimmistä faktoista sivuston näkyvyyden nostamiseen. Mikäli sivuston linkki onnistutaan laittaa hyvämaineeseen sivuihin, sijoitusjärjestystä parannettaisiin.

Kuva 6 Yoast-lisäosan haun ulkoasun muokkaus artikkelille

Artikkelit (post) ^

Yksittäisen Artikkelin asetukset

Näytä Artikkelit hakutuloksissa?

Käytössä Ei käytössä

Näytä SEO-asetukset kohteelle Artikkelit

Käytössä Ei käytössä

SEO otsikko

Title Page Separator Site title

Meta selustus

Tässä voit muokata metakuvaustasi

3.4 Tietoturvan seikat

Wordpress on kirjoitushetkellä maailmaan käytetyin sisällönhallintajärjestelmä, mikä tekee alustan kyberhyökkäyksen ykköskohteeksi ja tilaston mukaan noin 83% Wordpress-sivuista oli hyökätty vuonna 2017. Wordpress koostuu PHP-ohjelmointikielistä ja tietokantaa rakennetaan MySQL-tekniikalla, jonka sisällön koko ja monimutkaisuus perustuu kolmannen osapuolen koodista mm. laajennukset ja teemat. Kun taas Wordpress-sivustoja asennetaan verkkopalvelimelle ja tiedostoja hallinnoidaan esim. FTP-tiedostosiirron kautta (Connelly, 2011). Nämä ovat Wordpress-alustan perusrakenteita, joita käytetään miljoonien www-sivujen rakentamiseen. Tämän takia Wordpress-sivustot haavoittuvat helposti, jos niiden tietoturvat on laiminlyöty.

Wordpressin tietoturvaan suhtaudutaan erittäin vakavasti, mutta sama kuin muita sisällönhallintajärjestelmiä mahdolliset tietoturvaongelmat saattavat ilmetä (WordPress.Org, 2021). Tietoturvan käsite periaatteessa ei ole täydellistä turvallista järjestelmää, vaan pyritään vähentämään hyökkäysriskejä ulkopuolelta, jotka saattavat haittaa järjestelmää ja suojata haavoittuvuudesta. Suurin osa Wordpress-alustaa kohdistuvista hyökkäyksistä liittyy järjestelmän pääsynhallintaan, ohjelmiston haavoittuvuuksiin ja kolmannen osapuolen integraatioihin.

Kuva 7 Haavoittuneiden www-sivujen määrä eri sisällönjärjestelmissä vuonna 2017



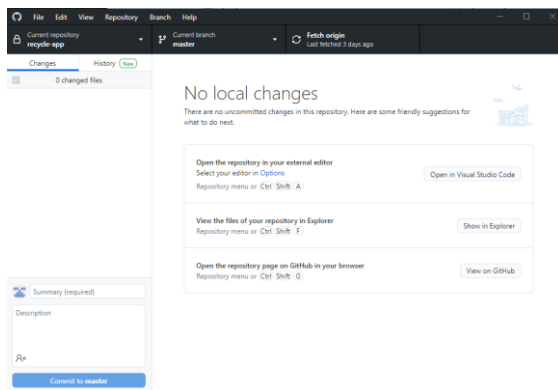
3.5 Git-versionhallinta

Git on ilmainen ja avoimeen lähdekoodiin perustuva versionhallintajärjestelmä, jota on suunniteltu käsittelemään erilaisia projekteja pienistä suurempaan tehokkaasti ja nopeasti (Git, n.d.). Git-versionhallintajärjestelmän myötä kirjoitettuja koodeja pystytään säilyttämään ja niitä koodeja voidaan jakaa muille kehittäjille, myös kaikilla kehittäjillä on mahdollista osallistua muihin projekteihin häiritsemättä alkuperäiskoodin rakenteita.

Github on pilvipohjainen alusta, jonka kautta voidaan ylläpitää ja säilyttää Git-projekteja. Github on pohjimmiltaan helpottaa Git-versionhallinnan käyttöä yksilöille ja ryhmille (Kinsta, 2021). Github-alusta antaa käyttäjilleen useita versionhallintatyökaluja, joiden avulla voidaan päivittää koodeja paikallisilta koneilta tai testata uusia ominaisuuksia. Github sallii myös käyttäjään luomaan uusia branchejä eli aktiivisia osia, joiden avulla voidaan työstää tai kokeilla uusia koodipätkiä pelkäämättä se, että alkuperäiskoodia sekoitetaan.

Github tarjoaa kirjoitushetkellä sekä ilmaiset että maksulliset versiot, joista alustan käyttäjillä on mahdollista päästä Githubin tarjoamia kooditiloihin, joka on pilvipohjainen kehitysympäristö ja sopii sekä pienempiin että isompiin projekteihin. Codespaces eli suomeksi kooditilat ovat saatavilla sekä Visual Studio Code -tekstieditorilla tai selainpohjaisessa versiossa (Developing with GitHub Codespaces, n.d.).

Kuva 8 Github Desktop -käyttöliittymä



Git-versiohallintajärjestelmällä on juuret Linux kernel -kehitysympäristöstä, jonka kehitti Linus Torvalds vuonna 2005 (Laster, 2017). Versionhallinnan lisäksi Gitiä käytetään ohjelmiston bugien ja vihreiden etsimiseen, alustan myötä kehittäjät voivat osallistua muiden projekteihin helposti kutsulla, jota lähetetään Githubin asetuksista. Järjestelmän kautta kehittäjät voivat antaa palautetta toisille ja suorittaa vikailmoituksia ongelmia ilmetessä.

4 Erilaiset räätälöityä teemaa rakentavat pohjat

Tässä opinnäytetyössä luodaan loppuvaiheessa Woocommerce-pohjaiselle verkkokauppa Wordpress-teemasta, mutta sen aloitusteemaa voidaan luoda monin tavoin. Näitä ovat Joints WP, Bones, HTML Blank ja Underscore, joista käydään läpi seuraavissa alaluvuissa ja otetaan selville mitkä ovat näiden teemojen hyvä ja huono puoli räätälöityyn alustateemaan verrattuna.

4.1 Joints WP

Periaatteessa Joints WP on Wordpress-teema, jota on rakennettu Foudation 6 -alustassa, mikä antaa käyttäjälleen tarvitsemansa työkaluja monimutkaisen ja responsiivisuuden verkkosivujen luomiseen ilman aloitusteeman tekoa (Getting Started - JointsWP - A Blank, Foundation 6 WordPress Theme, n.d.)

Joins WP -teemaa käytetään usein aloitusteemana, sillä teema ei ole itsensä lopputuotetta, vaan teema näyttää oletuksena ilman CSS-muotoilua (kuva 9) ja sen käyttäjät voivat suorittaa muokkauksia pohjakoodin perusteella. Teeman asetus onnistuu helposti Gulp-asetustiedoston avulla, myös Saas-tiedostoja voidaan koota nopeasti muutamalla komennolla ja JS-tiedostojen minifiointi onnistuu ongelmitta Node.js-kehitysympäristössä.

Kuva 9 JointsWP malliversio

- [Foundation Features](#)
- [Archive Styles](#)
 - [Blog](#)
 - [Grid Archive](#)
- [Navigation Options](#)
 - [Top Bar Navigation](#)
 - [Off-Canvas Navigation](#)
 - [Off-Canvas Top Bar](#)
 - [Title Bar Navigation](#)
- [Extras](#)
 - [Gravity Forms](#)
- [Download](#)
- [JointsWP Demo](#)
- [Foundation Features](#)
- [Archive Styles](#)
 - [Blog](#)
 - [Grid Archive](#)
- [Navigation Options](#)
 - [Top Bar Navigation](#)
 - [Off-Canvas Navigation](#)
 - [Off-Canvas Top Bar](#)
 - [Title Bar Navigation](#)
- [Extras](#)
 - [Gravity Forms](#)
- [Download](#)
- [Menu](#)

4.2 Bones-teema

Bones on toinen Wordpress-teema, jota on suunniteltu kehittäjille. Teemaa on rakennettu uusimpien Web-standardien tekniikoilla ja siinä on vankka perusta, josta hyödyntää minkä tahansa Wordpress-alustaan pohjautuvia projektia (Bones - An HTML5, Mobile-First Starter Theme for Rapid WordPress Development. - Themble, n.d.).

Bones-kehukset on rakenteiltaan hyvin orgaanisoitu ja teemalla on Mobile First - lähestymistapa eli websivun responsiivisuuden tärkeys on teeman ykkönen prioriteetti. Bones-teemassa on asennettu valmiiksi Saas-tyylikirjastoa, joka optimoi responsiivisuutta eri näyttöjen koossa.

Bones-teema on ilmainen ja sen koodikirjasto voidaan ladata suoralta kehittäjän github-alustasta (kuva 10). Bone-teema sallii käyttäjää oman teeman luomista Bonesin koodia käyttäen ja teemaa voidaan käyttää kaupalliseen tarkoitukseen.

Kuva 10 Bonesin koodikirjasto

library	Initial finnish translations	6 years ago
post-formats	Corected dubbel by in byline.	7 years ago
.gitattributes	added gitattributes to help reduce linebreak issues	8 years ago
.gitignore	COMposer.json	4 years ago
404.php	bringing main element back into the fold	7 years ago
CHANGELOG.md	changelog	8 years ago
archive-custom_type.php	bringing main element back into the fold	7 years ago
archive.php	Merge pull request eddiemachado-zz#738 from JeremyEnglert/master	7 years ago
comments.php	Reply button translatable	7 years ago
composer.json	COMposer.json	4 years ago
favicon.ico	updating screenshot and favicon	9 years ago
favicon.png	adding new favicon setup to include windows 8 tile support	9 years ago
footer.php	on footer.php .footer-links tag doesn't exist	7 years ago
functions.php	Make use of Google font protocol agnostic	6 years ago

4.3 Underscore-aloitusteema

Underscore-aloitusteema on yksi kirjoitushetkellä suosituimmista aloitusteemoista, jota on kehitetty yli tuhannella työtunneilla (A 1000-Hour Head Start: Introducing The _s Theme – ThemeShaper, n.d.). Underscore lyhennellä _s on valmis Wordpressiin sopiva aloitusteema, jota käytetään uusien teemojen kehittämiseen. Teemaa on hyvin suunniteltu HTML5-

sapluunalla, selkeästi kommentoitu eli tiettyjä funktioita voidaan aktivoida poistamalla näitä kommentteja.

Underscore sisältyy valmis CSS-tyylikirjasto, GPL-lisenssi on kirjoitettu license.txt-tiedostoon, niin pystytään tarkistamaan Underscoren käyttöä. Teema on ilmainen ja sitä voidaan ladata suoralta Underscoren sivulta.

5 Räättälöidyn teemapohjan toteutus

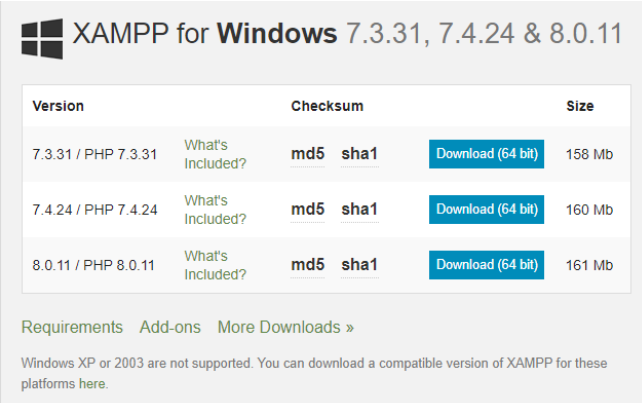
Woocommerce-pohjaiselle verkkokaupalle toteutetaan räätälöityä teemaa itsetehdyllä aloitusteemalla. Projektin aloittamiseksi asennetaan ensin paikalliselle koneelle Xampp-kehitysympäristöä, johon Wordpress-alusta laitetaan pystyyn. Näiden asennusten lisäksi käytetään Visual Studio Code -editoria koodin muokkaamiseen, myös Github Desktop -ohjelmistoa saadaan asennettua koneelle, näin varmistetaan että valmiita koodeja säilytetään ja synkronoidaan jatkuvasti Github-palvelimeen.

Työn lopuksi räätälöidy teema toimii responsiivisesti Bootstrap-kehiksen myötä ja on yhteensopiva Woocommerce-pohjaiselle verkkokaupalle.

5.1 XAMPP-Asennus

XAMPP-testiympäristö voidaan ladata sivuiltaan, ohjelmisto on saatavilla Windows-, Linux- ja OS X-järjestelmään. Sillä koneen käyttöjärjestelmä on Windows, joten ladataan tähän yhteensopivaa versiota koneeseen. Kirjoitushetkellä XAMPP tarjoaa kolmea versiota Windows käyttäjälle mm. XAMPP 7.3.31, johon toimii PHP 7.3.31, 7.4.24/ PHP ja 8.0.11 / PHP 8.0.11. (kuva 11).

Kuva 11 XAMPP-testiympäristön eri asennusvaihtoehtoja



XAMPP for Windows 7.3.31, 7.4.24 & 8.0.11

Version	Checksum	Size
7.3.31 / PHP 7.3.31 What's Included?	md5 sha1	Download (64 bit) 158 Mb
7.4.24 / PHP 7.4.24 What's Included?	md5 sha1	Download (64 bit) 160 Mb
8.0.11 / PHP 8.0.11 What's Included?	md5 sha1	Download (64 bit) 161 Mb

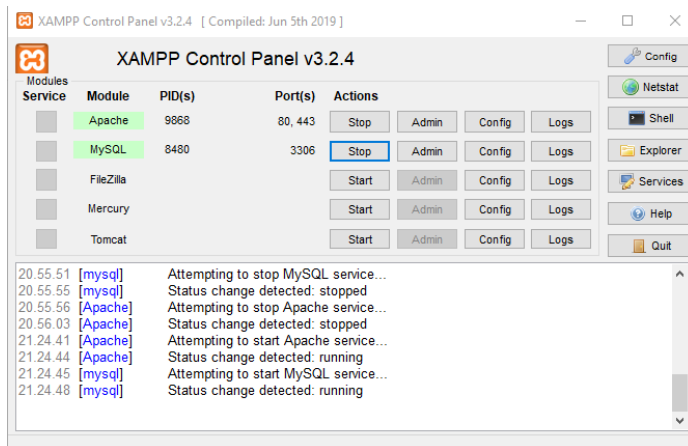
[Requirements](#) [Add-ons](#) [More Downloads »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

Koneelle oli jo asennettu XAMPP 8.0.3, johon toimii PHP 8.0.3. XAMPP-asennettua tuntuu sivuston toimivan hitaasti, sitä hitautta voidaan parantaa muokkaamalla Apachen puolelta php_init-tiedostoa. Oletuksena 512MB on suurin muisti, jota skripti saa käyttää, mutta tällä

nopeudella Wordpress-sivusto ei lähde toimimaan nopeasti, joten tätä muistin määrää korotetaan 1GB:ksi. Lisäksi MySQL-tietokannan puolesta voidaan korostaa innodb_buffer_pool_size eli paikallisen tietokoneen muistimäärän käyttö avaamalla my_ini-tiedostoja ja siitä säädellään jopa 50-80% lisää RAM-muistista. Oletuksena muistin määrä on 512MB, joten vaihdetaan tätä 2G:ksi, jonka määrä on yhä sallitussa rajassa. XAMPP-kehitysympäristöä voidaan aktivoida avaamalla sen ohjauspaneelia, josta Apache ja MySQL pitää olla käynnistettyä (kuva 12)

Kuva 12 XAMPP-kehitysympäristön aktivointia



5.2 Wordpress


Paikalliselle palvelimelle wordpress-asennus onnistuu luomalla alustalle oma tietokanta ja Wordpress voidaan ladata sivultaan. Asennuksen jälkeen Wordpress-koodikirjasto liitetään Github-alustaan jatkokehitykseksi.

5.2.1 Wordpress-sivulle tietokannan luominen







Jo tässä vaiheessa tietokantoja voidaan luoda, jotta saada Wordpress-sivuja toimimaan. Tietokannan luominen onnistuu lähtemällä osoitteeseen localhost/phpmyadmin selaimessa (kuva 13) ja kirjoittamalla tietokannan nimi tyhjään kenttään. Tietokannan nimi saa olla enintään 30 merkkiä, josta käy esim. aakkosnumeeriset, alaviivat (_), dollarimerkki (\$) ja ristikko (#), mutta nimen alkuun on oltava aakkosmerkillä. Tietokannan nimeen valitaan wp_omari ja Unicode-lajitteluun tulee olla utf8mb4_unicode_ci, joka tukee täysin unicodea, jonka avulla kaikki tiedot syötetään tietokantaan onnistuneesti.

Kuva 13 Omari-tietokannan luominen

Tietokannat

Luo tietokanta 

wp_omari utf8mb4_unicode_ci Luo

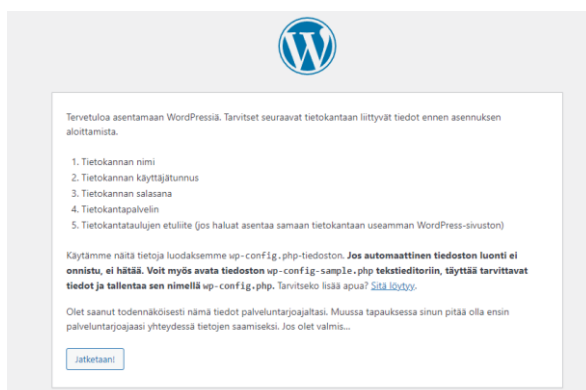
Tietokanta	Aakkosjärjestys	Toiminnot
<input type="checkbox"/> information_schema	utf8_general_ci	 Merkitse käyttöoikeudet
<input type="checkbox"/> mysql	utf8mb4_general_ci	 Merkitse käyttöoikeudet
<input type="checkbox"/> performance_schema	utf8_general_ci	 Merkitse käyttöoikeudet
<input type="checkbox"/> phpmyadmin	utf8_bin	 Merkitse käyttöoikeudet
<input type="checkbox"/> ttkauppa	utf8mb4_unicode_ci	 Merkitse käyttöoikeudet
<input type="checkbox"/> wp_omari	utf8mb4_unicode_ci	 Merkitse käyttöoikeudet

Yhteensä: 6

5.2.2 Wordpress-asennus

Wordpress-asennuspaketti voidaan ladata Wordpress-sivultaan, jossa on saatavilla myös suomenkielinen versio. Zip-tiedoston ladattua sitä voidaan purkaa suoraan C:-aseman xampp/htdocs-kansioon ja kansion nimi voidaan kirjoittaa uudelleen, tässä nimetään kansiota ”omariksi”. Kansion nimen muutettua puretaan Wordpress-asennuspakettia selaimessa osoitteesta localhost/omari, niin näytölle esitetään wordpress-asennuksen etusivua (kuva 14), valitse siitä Jatketään-painike jatkaakseen tietokantaan tietojen syöttöä.

Kuva 14 Wordpress-asennuksen näyttö



Tietokannan nimi on wp_omari, jonka on luotu edellisessä vaiheessa, käyttäjätunnus on oletuksena 'root' ja salasana saa jättää tyhjäksi, ellei sitä määritelty etukäteen ja tietokantapalvelimeen määritetään localhostiksi, jota löytyy tietokannan asetuksessa.

Tietoturvan kannalta tietotaulujen etuliitteen pitää vaihtaa, sillä oletuksena wp_ on kaikissa tiedossa, myös hakkeri on tiennyt sen, joten etuliitteen nimetään oma_:ksi. (kuva 15).

Tietokannan asetusten saatua valmiiksi siirretään Wordpress-sivun alustavan tietojen syöttämiseen mm. Wordpressin sivuston otsikko, käyttäjätunnus, salasana ja sähköpostiosoite. Wordpressin salasanaan pitää valita sellainen merkkijono niin, että muut eivät voi arvata helposti, yleensä Wordpress-salasana koostuu sekä isoista että pienistä kirjaimista, numeroista ja erikoismerkeistä. Näiden kenttien täytettyä Wordpress-asennus on valmis ja käyttäjät voivat päästä pääikkunaan.

Kuva 15 Tietokannan määrittelmä

Kirjoita alle tietokannan tiedot. Jos et ole varma yksityiskohdista, ota yhteys palveluntarjoajaasi.

Tietokannan nimi Sen tietokannan nimi, johon haluat asentaa WP:n.

Käyttäjätunnus Tietokannan käyttäjätunnus.

Salasana Tietokannan salasana.

Tietokantapalvelin Saat luultavasti nämä tiedot web-palveluntarjoajaltasi jos localhost ei toimi.

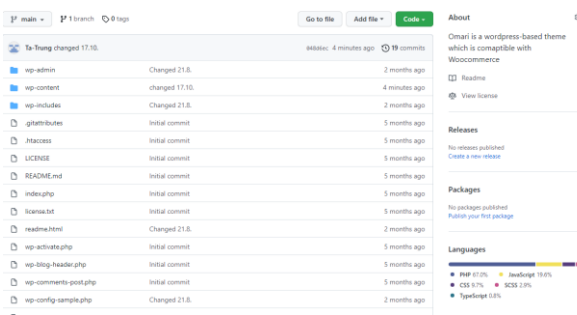
Tietokantataulujen etuliite Jos haluat asentaa useita WordPressejä samaan tietokantaan, muuta tämä.

Paikalliselle tietokoneelle on asettanut onnistuneesti Wordpress-sivustoa. Sivuston pääsyä onnistuu osoitteesta localhost/omari/wp-admin ja samalla XAMPP-testiympäristössä pitää laittaa Apache- ja MySQL päälle

5.2.3 Github-alustaan integraatio

Github Desktop -sovellus voidaan ladata Gitin pääsivulta, johon kirjaudutaan omalla Git-tunnuksella. Jo tässä vaiheessa on syytä linkittää projekti Github-alustaan, jonka avulla kaikki koodeja säilytetään myöhäisvaiheen käyttöön. Sovelluksen ladattua projektiin kohdistuvia koodeja synkronoidaan Git-palvelimeen, niitä koodeja voidaan muokata branchia käyttäen. Tämä toiminto on suhteellisen kätevä ja projektiin pystytään kutsumaan myös työkaverit, sekä kaikki muutoksen teko synkronoidaan master-branchiin helposti Github Desktop -sovelluksen kautta.

Kuva 16 Projekti on ladattu onnistuneesti Github-palvelimeen.



5.3 Aloitusteeman luonti ja aktivointi

Wordpressin ytimeen kuuluu kolme juurihakemistoa, josta on mm. wp-admin-, wp-includes- ja wp-content-hakemisto. Wp-admin-hakemisto sisältyy useita ylläpitotyökaluja, joiden avulla pystytään muodostamaan yhteyttä tietokantaan sekä ylläpitää käyttäjien käyttöoikeutta. Kun taas wp-includes-hakemistosta on erilaisia tiedostoja kuten funktioita, vimpaimia, joita käytetään sivuston ajamiseen ja wp-content sisältyy kaksi alakansiota 'plugins' ja 'themes', joihin voidaan vaikuttaa ja luoda kustomoituja teemoja ja lisäosia.

Tavoitteena luoda räätälöity teema, joten themes-hakemistoon luodaan omari-kansiota ja kaksi tiedostoa 'index.php' ja 'styles.css'. Index.php on kokonaissivun sydän, jonka tehtävänä on ladata kaikki tiedostot käyttäjän pyytäessä, kun taas kaikki tyylilliset muokkaukset tehdään styles.css-tiedostosta.

Näiden tiedostojen luotua alustava teema on periaatteessa valmis. Se näkyy wordpressin pääikkunassa ulkoasu/teemat-puolelta, mutta teemasta puuttuu yhä kuvaa, jota voidaan lisätä suoraan themes-kansioon ja kuvan koko on oltava 1200x900 pikselissä png-muotona ja tämän kuvankaappauksen pitää olla sama kuin sivuston etusivua, näin voidaan lisätä wordpress-teemakirjastoon myöhäisvaiheessa.

Viimeksi vaiheeksi syötetään vielä teeman tietoja, joita näytetään omari-teeman osiossa. Teeman tietoja kirjoitetaan suoraan styles.css-tiedostoon, jossa on mm. teeman nimi, osoite, kehittäjän nimi ja osoite (kuva 17). Tässä vaiheessa pitää jo ottaa huomioon millaista sivustoa ollaan suunnittelemassa esim. onko sivustolla sivupalkki oikealla tai vasemmalla,

kustomoituja menuja, logoja ja niin edelleen. Kaikkea näitä listataan tunnisteeseen, josta näkyy myös teeman tiedostossa.

Kuva 17 Omari-teeman perustiedot

```

/*
Theme Name: Omari
Theme URI: https://www.omari.fi
Author: Trung Ta
Author URI: http://www.omari.fi/trung-ta
Description: Omari is a wordpress theme, which is compatible with Woocommerce. By using this theme you can sell anythings online, beautiful design and fully responsive
Version: 1.0.1
Text-domain: omari
License: GNU General Public License v2 or later
License URI: https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html
Tags: e-commerce, custom-menu, custom-logo, featured-images, footer-widgets, theme-options, translation-ready, blog, right-sidebar, sticky-post, threaded-comments
*/

```

5.3.1 Otsake-tiedosto (header.php) ja alaosio (footer.php)

Tässä osiossa luodaan ja jaetaan otsaketta head.php ja alaosiota footer.php erikseen tiedostoon index.php-osasta. Suurin osa luokista kirjoitetaan body-tagiiin kutsumalla body_class()-funktioita, johon lisätään kaikki sivuston elementtejä ja sijoittuu head.php-tiedostoon. Tunnistaakseen sivuston käyttämää kieltä ja tietoa jo tässä vaiheessa näihin liittyviä funktioita pystytään lisäämään käyttämällä language_attributes()- ja bloginfo()-tagia, näin sivu näyttää automaattisesti käyttäjän syöttämiä tietoja ja kieliä.

Footer.php eli alaosion saatua valmiiksi näitä sivuja voidaan kutsua milloin tahansa funktioilla get_header() ja get_footer(), joiden tehtävänä on näyttää otsakkeessa ja alaosiossa olevia tietoja missä vain sivustossa. Jokaiselle sivulle lisätään loppuvaiheeksi kuvauksia kommentoituna, joka on pakollinen osa, jos lopputeema halutaan lisätä Wordpress-teemakirjastoon.

Kuva 18 Otsakkeen- ja alaosion lisäys

```

index.php • header.php • footer.php
wp-content > themes > omari > index.php
1  <?php
2  /**
3   * The main template file
4   *
5   * This is the most generic template file in a WordPress theme
6   * and one of the two required files for a theme (the other being style.css).
7   * It is used to display a page when nothing more specific matches a query.
8   * E.g., it puts together the home page when no home.php file exists.
9   *
10 * @link https://developer.wordpress.org/themes/basics/template-hierarchy/
11 *
12 * @package Omari
13 */
14
15 get_header();
16 >>

```

5.3.2 Tyylitaulukkojen järjestely

Seuraavaksi luodaan tiedostoa, joka monitoroi teeman julkaisun tyyppejä, shortcodeja ja mukautuu verkkosivujen tyylejä. Kaikki näitä asioita pakataan functions.php-tiedostoon, johon voidaan lisätä useita koukkuja eli hooks do_action()- funktiolla. Tämä toiminto käynnistää kaikkiin koukkuun \$hook_name liitetyt toiminnot ja mahdollista uusien toimintojen kutsua. Tässä vaiheessa liitetään css-tyylitiedosto functions.php-tiedostoon, ja se onnistuu luomalla uusi funktio, josta kutsuu wp_enqueue_style()-komentoa, jonka myötä uusia tyyliä voidaan rekisteröidä. Sillä get_stylesheet_uri()-funktioilla määrittelee tyylitiedoston sijaintia (kuva 19), jonka kautta voidaan muotoilla sivustoa css-ohjelmointikielellä.

Kuva 19 Rivittää tyylitiedostoa wp_enqueue_style()-funktioilla

```
function omari_scripts()  
{  
    wp_enqueue_style( 'omari-style', get_stylesheet_uri(), array(), filemtime( get_template_directory() . '/style.css' ), 'all' );  
}  
add_action( 'wp_enqueue_scripts', 'omari_scripts' );
```

5.3.3 Bootstrap-kehiksen lisäys

Bootstrap-koodikirjasto voidaan liittää projektiin kahdella tavalla, joko koko koodi liitetään suoraan projektissa olevaan kansioon tai lisätään ne sisällönjakeluverkolta eli CDN:lta. MaxCDN on yksi tunnetuimmista sisällönjakeluverkon tarjoajista, joka tukee Bootstrapin koodikirjastoa mm. sen kirjoittama css-tyylikoodia ja javascriptiä.

Tässä Omari-teeman suunnittelussa isännöidään Bootstrap-kehystä lataamalla koko koodikirjasto projektiin. Koodin lisäys onnistuu funktioilla wp_enqueue_script ja wp_enqueue_style, joista pyytää Bootstrap-koodikirjastosta css-tyylejä ja javascriptiä. Projektiin Bootstrap-kehiksen lisättyä voidaan tarkistaa asennusta selaimen kehittäjän työkalussa. Jos Bootstrap on onnistuneesti ladattu, selaimen kehittäjän kentällä näkyy seuraava koodiriviä:

```
<script type="text/javascript" src="http://localhost/omari/wp-content/themes/omari/inc/bootstrap.min.js?ver=5.0.0" id="bootstrap-js-js"></script>
```

Bootstrapin myötä verkkosivu tulee olla responsiivinen ja näkyy kaikissa laitteissa sen grid systemin avulla eli sivu jaetaan 12 sarakkeeseen, joita kutsutaan esille col-komennolla. Tämän järjestelmän avulla voidaan asettaa sivun kokoja erilaisille näytöille esim. .col-sm on tarkoitettu näytölle, joka on pienempi kuin 576px, .col-md 768px ja niin edelleen, tämä on siis erittäin kätevää kehittäjälle, niin pystytään määrittelemään eri kokoja eri näytölle nopeasti.

Omari-teemassa käytetään Bootstrap 5 eli uusinta versiota kirjoitushetkellä, johon pitää liittää lisäksi bootstrap-5-wordpress-navbar-walker-koodipätkää, jotta saadaan navigaatiota toimimaan, esimerkiksi halutaan että sivun logo ilmenee kolmannessa sarakkeessa näytössä, joka on pienempi kuin 576px ja taas toisessa sarakkeessa näytössä, joka on pienempi kuin 992px, niin käytetään tätä koodi:

```
<div class="brand col-md-3 col-12 col-lg-2 text-center text-md-left">
```

Bootstrapin liitettävä sivulle on ihan plussa, sillä nopeutetaan responsiivisuutta, mutta sivuston pitää ladata runsaita CSS- ja JavaScript-koodeja, mikä saattaa hidastua kokonaissivua myöhäisvaiheessa.

5.3.4 Valikkojen luonti

Sillä teema on luotu ihan tyhjästä, joten Wordpress-dashboardsissa ei näy valikkojen vaihtoja. Pystyäkseen valitsemaan valikot niitä pitää rekisteröidä ja tämä toiminto onnistuu Wordpress-funktiolla `register_nav_menus()`. Tällä komennolla voidaan rekisteröidä yksittäistä valikkoa tai monia valikkoja lisäämällä `array()`-tagia perään. Sillä ajalla etusivulle luodaan tässä vaiheessa muutamia sivuja mm. etusivu, blogi, yhteydenotto ja tietoja meistä, jotka kuuluvat `omari_main_menu`-valikkoon. Näiden sivujen luotua omari-etusivussa ruveta näkyä valikkoja vaihtoehtojen valitettuna.

Kaikki menut hallinnoidaan ja valitaan Wordpress-hallintapanelin kautta, sama kuin sivuston sisältöjä. Menujen asetuksissa voidaan lisätä ja poistaa valikkoja (kuva 20), samalla `function.php`-tiedostoon saa rekisteröidä lukemattomia valikkoja `array()`-tunnisteella.

Kuva 20 Omari-valikon asetukset

Valikon asetukset

Lisää sivut automaattisesti Lisää päätason sivut automaattisesti tähän valikkoon

Näytä valikko sijainnissa Main menu

Omari Main Menu (Tämänhetkinen asetus: Omari Main Menu)

Omari Footer Menu (Tämänhetkinen asetus: Omari Footer Menu)

5.3.5 Wordpress The Loop

Wordpressin artikkeleita ja verkkosivun viestejä näytetään esille The Loopia käyttäen. The Loop on käytännössä PHP-ohjelmointikielellä kirjoitettu koodipätkä, jonka avulla Wordpress pystyy käsittelemään sivulla olevia viestejä muotoineen, toisella sanoilla viestin sisältämä HTML- ja PHP-koodeja käsitellään The Loopin kutsumisen yhteydessä. The Loopin yleisimmät tagit ovat `the_title`, `the_time` ja `the_category`, jotka pyytävät sivuston otsikkoja, aikoja ja kategoria tarvittaessa.

Omari-sivustoon on luotu kolme artikkelia nimeltään blog 1, 2 ja 3 ja `index.php`-tiedostoon kutsutaan niitä julkaisuja funktiolla `have_post()`. Koodi on kirjoitettu niin, että jos siinä tilassa on julkaisuja, niitä näytetään esim. sivuston otsikkoja ja sisältöjä. Koodin lisättyä säädellään vielä Wordpress-asetuksia lukemisen osalta niin, että etusivulla näyttää sille tarkoitettuja sisältöjä ja blogisivulle näytetään artikkeleita (kuva 21)

```
if( have_posts() ):
    // lataa ne julkaisut
    while( have_posts() ): the_post();
        get_template_part( 'template-parts/content' );
    endwhile;
```

Kuva 21 Etusivun ja blogin erottaminen

[Sivu](#) (valitse alta)

Kotisivu:

Artikkeleiden sivu:

5.3.6 Sivumallit Page Templates

Tässä vaiheessa pyritään erottelmaan kaikki sivut toisistaan Wordpressin sisältämiä sivumalleja käyttäen. Sivumallit eli Page Templates ovat ominaisuudeltaan muuttujia, joiden avulla pystytään muotoilemaan eri sivujen tyylejä ja asetelmia. Oletuksena page.php-tiedosto määrittelemään kaikki Wordpressissa olevia sivuja, ellei niihin aseta muita sivupohjia etukäteen.

Omari-teeman etusivua hallitsee template-home.php-tiedosto ja muita sivuja page.php-tiedostolla. Näitä sivupohjia voidaan vaihtaa Wordpress-hallintapaneelissa ja sivupohjat voidaan luoda Wordpressin antamalla funktiolla esim. page-{slug}-komenolla, josta slug voi olla mitä tahansa nimiä, joka sopii sivun sisältöön esim. page-about ja niin edelleen.

5.4 Omari-aloitusteeman integraatio Woocommerce-verkkokauppaan

Omari-aloitusteema on periaatteessa luotu onnistuneesti, mutta rakenteeltaan ei ole vielä täysin yhteensopivaa Woocommerce-pohjaiselle verkkokaupalle esim. tuoteluettelo, tuotesivu ja kassasivu ei näytä eikä toimi oikein. Tämän takia käytännön osalta käsitellään myös, miten omari-teemaa integroidaan Woocommercelle ja tehdään sivuille tyylejä myös kuvakaruselin lisäys bannerille.

5.4.1 Woocommerce-verkkokaupan asennus

Woocommerce on yksi lukuisista Wordpressin lisäosista, jota asennetaan suoralta Wordpressin-lisäosakirjastosta tai sovellusta on valmis ladattavaksi Woocommerce-sivulta, jota lisätään verkkosivulle käsin. Woocommerce asennettua ja sen asetuksen laitettua pystyy ruvetaan lisätä kauppasivua Omari-teeman päävalikkoon, jotta vaihtoehtoja on saatavilla valittavaksi etusivussa. Sillä verkkokaupassa ei ole yhtään tuotteita oletuksena, joten niitä pitää lisätä ja toiminto onnistuu joko lataamalla vanhasta verkkokaupasta olevia tuotteita tai käyttää Woocommercen tarjoamia tuotemallia. Tarkoituksena on luoda teemaa Woocommerce-pohjaiselle verkkokaupalle, ei välttämättä tarvitse varsinaisia tuotteita, joten ladataan tuotemallia Woocommerce-sivulta ja siitä tehdään tuotteille muutoksia sekä tyyllissä että muodossa.

Kuva 22 Woocommerce-verkkokaupan tuotemallit



5.4.2 Woocommerce-tuen ilmoittaminen

Kun luodaan Woocommerce-verkkokaupalle kustomoitua teemaa, siitä on ilmoitettava Woocommercelle saadakseen alustan shortcodeja toimiaan. Ilmoitus onnistuu `add_theme_support`-funktiolla ja tämän funktion alle voidaan määritellä eri vaihtoehtoja esim. tuoteriveille, tuoteruudukon määrä, pikkukuvien koot ja niin edelleen. Woocommerce -tuen tila voidaan tarkistaa Wordpress-hallintapaneelissa eli jos Woocommerce-tilaan näkyy vihreä rastia, Woocommerce-tuki on onnistuneesti lisätty verkkosivulle.

5.4.3 Woocommerce-pohjan muunnos

Woocommcerce-tuotteiden muokkaamiseksi `function.php`-tiedostoon lisätään funktiota `if(class_exists('WooCommerce'))` eli jos Woocommerce on olemassa, haetaan muutoksia `wc-modification.php`-tiedostosta. `Wc-modification` sisältyy Woocommcerce- tuotteita muokkaavia funktioita, joita kutsutaan Wordpressin `add-action()`-funktiota käyttäen. Omari-teemaan luotiin funktiota `omari_wc_modify`, jossa on lukuisia tuotesivuja muokkaavia toimintoja.

```
if( class_exists( 'WooCommerce' ) ){
require get_template_directory() . '/inc/wc-
modifications.php';
}
```

Samalla toiminnolla voidaan muokata myös Woocommerce-alustan sisältämää ostoskorja, oma- ja maksusivua. Suurin osa näiden sivujen muokkaamisista koostuu css-tekniikoista, mikä ei käsitellä tässä opinnäytetyössä. Siitä huolimatta etusivu ja Woocommerce-sivua on saatu muokattua ja on valmis käyttö pohjana toisin projektin kehittämiseen. Käydään

kuitenkin seuraaviksi Wordpressiin kohdistuvia tietoturvia, niin saadaan teemaa parempaan tasoon tietoturvan suhteen.

5.5 Omari-teeman tietoturvat ja käännös

Wordpress on kirjoitushetkellä yksi käytetyimmistä sisällönhallintajärjestelmistä ympäri maailman. Tämä on syytä ottaa huomioon tietoturva-asioissa.

5.5.1 Tietoturvat

Teemaan kohdistuvan tietoturvan parantamiseksi funktio `esc_()` on lisättävä kaikille teeman oleville merkkijoihin, linkkeihin, javascripteihin, attribuutteihin, näillä lisäyksillä haitalliset koodit estettäisiin. Aina kun jotain tietoja tulostetaan esille, niille koodeille pitää laittaa `esc_()`-funktioita oikein esim. omari-teeman etusivun linkkiin on lisätty `esc_url()`-funktioita, niin koodin tulostaessaan sitä ei voidaan muokata vieraiden kirjoittamalla scriptilla.

```
<a href="<?php echo esc_url( home_url( '/' ) ); ?>">
```

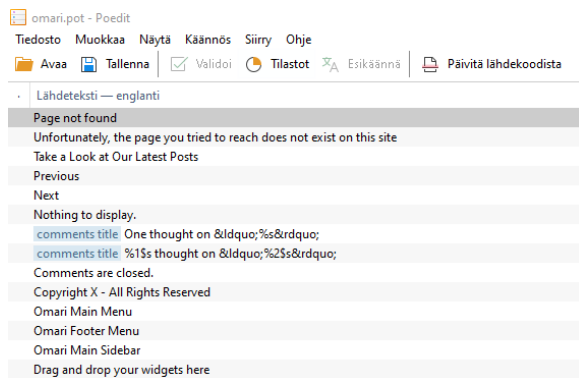
5.5.2 Käännös

Functions.php-tiedostoon on lisätty paria riviä, jotka kutsuvat pot-tiedostoa languages-kansiosta. Pot-tiedosto on tuotu Poedit-ohjelmistolla, jolla pystyy luomaan käännöstiedostoa ja siitä käännetään määräämään kieltä manuaalisesti.

```
load_theme_textdomain( $textdomain, get_stylesheet_directory()
. '/languages/' );
load_theme_textdomain( $textdomain, get_template_directory() .
'/languages/' );
```

Toteuttaakseen käännöstyötä tarvitaan vielä `_()`-funktioita, jonka avulla Poedit-ohjelmisto havaitsee käännökseen tarvittavia tekstejä ja niitä käännetään Poedit-sovelluksessa. Tekstin käännettyä sitä voidaan viedä pot-tiedostoon ja liittää omari-teemaan, näin tekstejä on käännetty toiseen kieleen.

Kuva 23 Omariin tarvittavat käännösteksti Poedit-ohjelmistossa



5.5.3 Omari-teeman hallinta

Omari-teemaa hallinnoidaan ja muokataan customizer.php-tiedostolla. Tällä kautta voidaan rekisteröidä uusia osiota esim. karusellit, Woocommerce-omaisuudet ja alatunnisteet funktiolla `$wp_customize()`, johon lisätään asetukset `add_setting` ja miten aluetta halutaan hallinnoida `add_control`-funktiolla.

Omari-teemaan on määritelty niin, että bannerin painikkeineen, Woocommerce-omaisuudet ja alatunnisteet hallintaan Wordpressin valikkojen osalta. Näihin osioihin on lisätty useita valikkojen vaihtoehtoja käyttämällä `add_control()`:llä olevia vaihtoehtoja kuten teksti, valintaruutuja tai avattavia sivuja.

```
$wp_customize->add_control(
    'set_slider_page1', array(
        'label'           => __( 'Set slider page 1', 'omari'
    ),
        'description'    => __( 'Set slider page 1', 'omari'
    ),
        'section'        => 'sec_slider',
        'type'           => 'dropdown-pages'
    )
);
```

5.6 Omari-teeman lisäys Wordpress-teemakirjastoon

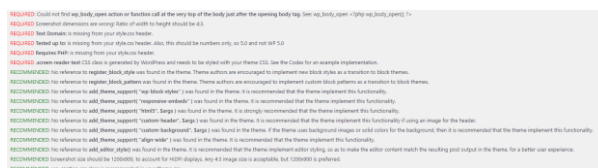
Omari-teeman lisääminen Wordpress-kirjastoon vaatii seuraavat toiminnot: teeman pakkaaminen, teemamaalin luonti ja teeman valmistelu lähettämiseen. Näiden toimintojen lisäksi omari-teeman on oltava yhteensopiva GPL-lisenssin määräyksen mukaisesti, teeman kehittäjä pitää ottaa täysin vastuu kehittämästään teemasta ja viimeksi tekemänsä kustomoitu teema tai lisäosa ei saa seuranta tuotteidensa käyttäjän aktiivisuutta ilman lupiaan.

Omari-teema pakataan alustavasi TGM Plugin Activatio -ohjelmiston avulla. Tämän ohjelmiston myötä voidaan liittää kaikki tarvittavat lisäosat, jotka tulevat omari-teeman yhteydessä käyttäjän asentaessaan teemaa omaksi käytöksi. Tässä tapauksessa omari-teeman asentamiseksi tarvitaan Woocommerce- ja One Click Demo Import -lisäosaa. Asentamiseksi tarvitaan kaksi tärkeintä tiedostoa require-plugins.php- ja class-tgm-plugin-activation.php-tiedostoa, jotka sijoittuvat inc-kansioon. Näiden tiedostojen saamiseksi toimimaan pitää kutsutaan näitä tietoja functions.php-tiedostossa, joita näkyy alla.

```
require_once get_template_directory() . '/inc/class-tgm-
plugin-activation.php';
require_once get_template_directory() . '/inc/required-
plugins.php';
```

Omari-teemaa on pakattu valmiiksi zip-tiedostona, jota voidaan lähete Wordpress-teemakirjastoon tarkistavaksi. Ennen sitä hetkeä teeman tarkastelemiseksi käytetään Theme Check -lisäosaa. Jos siinä teemassa on havainnut jotain epäsojivaa teeman lähettämiseen, Theme Check -lisäosa ilmoittaa niitä virheitä käyttäjälle kuten kuvassa 24:ssa. Hyväksynnän vuoksi näitä punaisia tekstejä täytyy korjata, muuten Wordpress saattaa hylätä teeman lähettämistä sen kirjastoon.

Kuva 24 Theme Check ilmoittaa käyttäjälle korjattavat virheet.



The screenshot shows a list of error messages from the Theme Check plugin. The messages are as follows:

- REQUIRED:** Could not find wp_body_open within a function call at the very top of the body just after the opening body tag. See wp_body_open() in wp-includes/general-template.php.
- REQUIRED:** Screenshot alt-text is missing. A list of valid alt-texts is provided in the documentation.
- REQUIRED:** Text Domain is missing from your plugin header.
- REQUIRED:** Theme URI is missing from your plugin header. This should be a URL, not a file path.
- REQUIRED:** Requires PHP is missing from your plugin header.
- REQUIRED:** Admin header text CSS class is generated by WordPress and needs to be added with your theme CSS. See the Guide for an example implementation.
- RECOMMENDED:** No reference to register_block_patterns was found in the theme. Theme authors are encouraged to implement custom block patterns as a foundation to block themes.
- RECOMMENDED:** No reference to add_theme_support() for 'block-styles' was found in the theme. It is recommended that the theme implement this functionality.
- RECOMMENDED:** No reference to add_theme_support() for 'responsive-embeds' was found in the theme. It is recommended that the theme implement this functionality.
- RECOMMENDED:** No reference to add_theme_support() for 'wide-images' was found in the theme. It is strongly recommended that the theme implement this functionality.
- RECOMMENDED:** No reference to add_theme_support() for 'custom-background' was found in the theme. If the theme uses background images or solid colors for the background, then it is recommended that the theme implement this functionality.
- RECOMMENDED:** No reference to add_theme_support() for 'align-wide' was found in the theme. It is recommended that the theme implement this functionality.
- RECOMMENDED:** No reference to add_editor_style() was found in the theme. It is recommended that the theme implement editor styling, or so to make the editor content match the resulting post output in the theme, for a better user experience.
- RECOMMENDED:** Screenshot alt-text should be 1200x600. Any 4:3 image size is acceptable, but 1200x600 is preferred.
- RECOMMENDED:** New updates url class is recommended in your theme url.

Teeman hyväksytyä teemakirjastoon sähköpostiin lähetetään vahvistusviesti, jossa kerrotaan teeman tarkistuksen tilanteet. Alustavasti omari-teeman rakenne on läpäissyt testin, sillä tulos nähdään kuvasta 25:sta.

Kuva 25 Ilmoitus teeman läpäisemisestä Wordpressin tarkistuksessa.

Results of Automated Theme Scanning: Pass

- RECOMMENDED: No reference to `register_block_style` was found in the theme. Theme authors are encouraged to implement new block styles as a transition to block themes.
- RECOMMENDED: No reference to `register_block_pattern` was found in the theme. Theme authors are encouraged to implement custom block patterns as a transition to block themes.
- RECOMMENDED: No reference to `add_theme_support("wp-block-styles")` was found in the theme. It is recommended that the theme implement this functionality.
- RECOMMENDED: No reference to `add_theme_support("responsive-embeds")` was found in the theme. It is recommended that the theme implement this functionality.

6 Tulokset

Omari-teema on saatu valmiiksi, ja se vastaa tilaajan odotukseen ja tämän opinnäytetyön tavoitteeseen. Teeman toteutuksessa on tullut useita haasteita, joihin ratkaisuja piti kaivaa useista resursseista. Onneksi Wordpress tarjoaa laajoja ohjeita, jossa on funktio-, koodimallit, kustomoidun teeman valmistamiseen. Vaikka Omari-teeman käyttöliittymän ulkoasua ei käsitelty, sen koodin ydin toimii moitteettomasti, joten teeman jatkokehittämiseen ei pidä olla monimutkaista ja verkkosivustoon voidaan helposti lisätä erilaisia tyylejä.

Omalla kustomoidulla teemalla sallitaan käyttäjän täysi teeman hallinta, säästetään kustannuksia ja suojataan verkkosivun salaisuudet ulkopuolelta. Teeman joustavalla koodilla käyttäjä voi räätälöidä sen ulkoasua, lisätä valikkoja tai uusia sivupalkkeja missä vaan verkkosivustossa. Muutoksia voidaan suorittaa nopeasti tekstieditorissa, mikä näkyy heti verkkosivussa päivityksen jälkeen. Oman teeman kehitykseen ei vaadi suurta pääomaa, mutta aika kuluisi nopeasti näihin kehitystöihin. Valmisteen käytöstä saattaa laskuttaa ohjelmistoyritykseltä ison summan laskua. Itse valmistemasta ei maksa paljon, mutta jos siitä käytetään useille asiakkaiden sivuille, summa voi olla moninkertainen. Näin olleen kustomoidun teeman käytöstä on taloudellisempaa. Vielä se oman teeman käyttö parantaa tietoturvallisuutta ja yrityksen salaisuutta, sillä teeman koodintimeen ei pääse kolmas osapuoli.

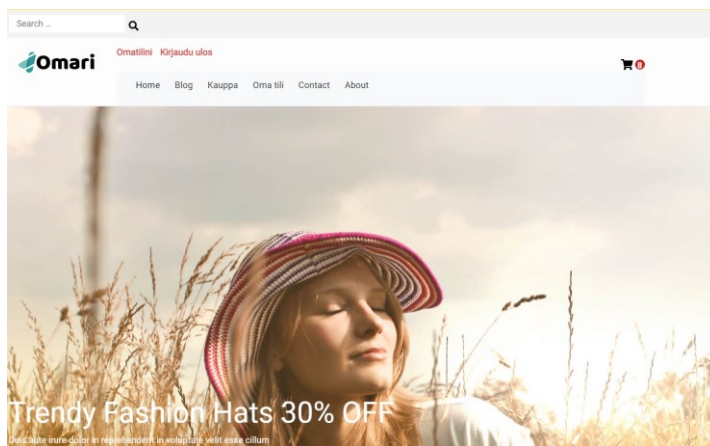
Tyhjästä kehittävän aloitusteeman verrattuna Underscore-aloitusteeman käyttö on kätevämpää, sillä Underscore-tiimi on kehittänyt lähes täysin Wordpressin vaatimukseen täyttävä peruskoodirakenne. Vaikka Underscore-aloitusteeman hyvästä käytettävyydestä oltiin tietoinen, pääteltiin silti kehittää oman aloitusteeman tyhjästä. Siitä työstä opiskeltiin paljon ytimen koodia ja funktioita, jotka voidaan hyödyntää myöhemmin käyttöön.

Vaikka teemasta ei tehty käyttöliittymää, käsitellään verkkosivuun liittyvän saavutettavuuden ja hakuoptimoinnin seikkoja tuloksena. Saavutettavuuden säätely liittyy usein käyttöliittymän muokkaimelle puolelle ja niihin saavutettaviin muokkauksiin tehdään css-ohjelmointikielellä, mitä ei käsitelty tässä opinnäytetyössä, mutta teoriaosuuden puolelta on selvittänyt se, että miten sivusto on saanut saavutettavuutta. Toisaalta

hakukoneoptimointia voidaan suorittaa useilla Wordpress-lisäosilla ja yksi suosituimmista osista on YoastSEO. Tämän lisäosan avulla voidaan optimoida sivustoa, jotta sitä näkyy hakukoneissa ystävällisesti ja Yoast-SEO-lisäosa voidaan myös paketoita teematiedostoon, jota asennetaan uutena teemana teeman purkamisen yhteydessä.

Opinnäytetyön tilaajan on mieltä sitä, että lopputyö vastaa odotukseensa, sillä Wordpress-teema, joka on kehitetty tyhjästä ja on yhteensopiva Woocommerce-verkkokaupalle. Jatkossa teemaa voidaan käyttää yrityksen nykyisen teeman sijalle tai kaupalliseen käyttötarkoitukseen.

Kuva 26 Omari-teemaan perustuvasta etusivu



7 Yhteenveto

Opinnäytetyössä käsiteltiin Wordpress-alustaan sopivan teeman kehitystyötä, joka on vastannut perusteellisesti tämän opinnäytetyön tutkimuskysymyksiin. Oman teeman käyttäminen antaa toimeksiantajalle paljon vaihtoehtoja ja luovuutta verkkosivujen suunnittelussa ja välttää kolmannen osapuolen tekemästä valmisteeman käytöstä. Näin olleen yritys voi säästää kustannukset valmisteeman käytöstä ja samalla parantaa verkkosivujen tietoturvallisuutta kustomoitua teemaa käyttäen.

Opinnäytetyötutkimuksen myötä olen havainnut Underscore-aloitusteeman käytön olevan kätevää, sillä Underscore-koodia kehittävä tiimi oli käyttänyt yli tuhat tuntia aloitusteeman kehittämiseen ja kirjoitetut koodit ovat täysin yhteensopivia Wordpressin määrittämää ohjeistukseen. Tulevissa projekteissa liitän Underscore-aloitusteemaa alustana ja siitä kehitän omaa Wordpress-teemaa.

Opinnäytetyön tarkoituksena on luoda vain teema, joka sopii Woocommerce-verkkokaupalle. Siksi tässä työssä ei käsitelty HTML-koodia ja CSS-tyylejä, joten käytännön osalta ei mennyt syvästi saavutettavuuden ja hakukoneoptimoinnin osuuteen kuten yksi tutkimuskysymyksistä oli esittänyt. Aluksi olin miettinyt tuoneeni nämä seikat esille, mutta opinnäytetyön käytännön kirjoitettua olen havainnut sitä, että tavoitteeni oli luoda vain teema Wordpressille ja Woocommercelle, joka vastaa juuri toimeksiantajan toiveeseen ja saavutettavuuden- ja hakukoneoptimoinnin toimintoja tehdään yleensä käyttöliittymän toteutuksessa.

Opin tästä opinnäytetyöstä paljon asioita teoriaosuudelle liittyvien lähteiden etsimisestä lopputyöhön kohdistuvien koodien kirjoittamiseen. Olen pitänyt tähän saakka verkkokauppaa yli 10 vuotta, mutta minulla on vähän verkkokauppaa rakentavia koodaustaitoja. Tämän opinnäytetyön kirjoittaminen on antanut minulle hyvät mahdollisuudet tutkia verkkokaupan koodaamista, ja siitä opin paljon juuri tästä aiheesta.

Vaikka tutkimuskysymyksiin ei vastattu täydellisesti, koen onnistuneeni tässä opinnäytetyössä. Sillä opin ja kokeilin uusia asioita Wordpress-teeman kehityksessä ja suunnittelussa, jota hyödynnän myöhemmin tulevaisuudessa. Tämän koulutuksen

suoritettuani haen ensisijaisesti työpaikkaa Wordpress-kehittäjinä tai perustan uuden yrityksen, jossa voin kehittää ja ylläpidä www-sivuja, sovelluksia ja ohjelmistoja.

Lähteet

A 1000-Hour Head Start: Introducing The _s Theme – ThemeShaper, n.d. Noudettu osoitteesta <https://themeshaper.com/2012/02/13/introducing-the-underscores-theme/>

About WooCommerce - WooCommerce.Com, n.d. Noudettu osoitteesta <https://woocommerce.com/about/>

Bones - An HTML5, Mobile-First Starter Theme for Rapid WordPress Development. - Themble, n.d. Noudettu osoitteesta <https://themble.com/bones/>

Developing with GitHub Codespaces, n.d. Noudettu osoitteesta <https://code.visualstudio.com/docs/remote/codespaces>

Getting Started - JointsWP - A Blank, Foundation 6 WordPress Theme, n.d. Noudettu osoitteesta <https://jointswp.com/docs/>

Git, n.d. Noudettu osoitteesta <https://git-scm.com/> Kinsta, 2021. Noudettu osoitteesta <https://kinsta.com/knowledgebase/what-is-github/>

Usage Statistics and Market Share of Content Management Systems, September 2021, n.d. Noudettu osoitteesta https://w3techs.com/technologies/overview/content_management

Värit Ja Kontrastit - Saavutettavasti.Fi, n.d. Noudettu osoitteesta <https://www.saavutettavasti.fi/kuva-ja-aani/varit-ja-kontrastit/>

WooCommerce Features, n.d. Noudettu osoitteesta <https://woocommerce.com/features/>

WordPress.Org, 2021 Noudettu osoitteesta <https://wordpress.org/support/article/hardening-wordpress/>

Connelly, 2011. WordPress 3 Ultimate Security. Noudettu osoitteesta <https://ebookcentral-proquest-com.ezproxy.hamk.fi/lib/hamk-ebooks/detail.action?docID=948509>

Jakobus, 2018. Mastering Bootstrap 4 : master the latest version of Bootstrap 4 to build highly customized responsive web apps. Noudettu osoitteesta <https://ebookcentral-proquest-com.ezproxy.hamk.fi/lib/hamk-ebooks/reader.action?docID=5309081&query=>

Jerkovic & Sinkovic, n.d. Vulnerability analysis of most popular open source Content Management Systems with focus on WordPress and proposed integration of artificial intelligence cyber security features. Noudettu osoitteesta <http://www.iaras.org/iaras/journals/ijems>

Kent, 2020. SEO for dummies. Noudettu osoitteesta <https://ebookcentral-proquest-com.ezproxy.hamk.fi/lib/hamk-ebooks/detail.action?docID=5996901>

Laster, 2017. Professional Git. Noudettu osoitteesta <https://hamk.finna.fi/Record/nelli19.4330000000009964>

Niska, 2014. Extending bootstrap. Noudettu osoitteesta <https://ebookcentral-proquest-com.ezproxy.hamk.fi/lib/hamk-ebooks/reader.action?docID=1644016&query=>

Palani, 2019. Advanced Selenium Web Accessibility Testing. Noudettu osoitteesta <https://hamk.finna.fi/Record/nelli19.4100000007820759>

saavutettavuusvaatimukset.fi, 2021. Noudettu osoitteesta <https://www.saavutettavuusvaatimukset.fi/yleista-saavutettavuudesta/>

B. Williams, 2015. Professional WordPress : design and development. Noudettu osoitteesta https://books.google.com/books/about/Professional_WordPress.html?hl=fi&id=8AUaBgAAQBAJ

(*What Is SEO / Search Engine Optimization?*, n.d.). Noudettu osoitteesta <https://searchengineland.com/guide/what-is-seo>

Liite 1: Aineistonhallintasuunnitelma

Opinnäytetyössä ei haastatellut ketä, eikä lähettänyt kyselyjä, joten niitä tietoja ei käsitellä eikä säilytetä sekä omassa tietokoneessa että pilvipalvelussa. Valmisteema on ladattu Github-palveluun ja teema on lisensoitu Suomen nettimyynti Oy nimellä, joka ei enää poista ja teema käytetään jatkossa kaupalliseen käyttöön toimeksiantajan omalla vastuulla.

