

Verkkosivujen toteutus avoimen lähdekoodin ratkaisulla

Sami Ryyppö

Opinnäytetyö
Tietojenkäsittelyn
koulutusohjelma
2018



Tekijä(t) Sami Ryyppö	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Verkkosivujen toteutus avoimen lähdekoodin ratkaisulla	Sivu- ja liitesivumäärä 31 + 1
<p>Opinnäytetyön tarkoituksena oli uudistaa visuaalisesti ja teknisesti toimeksiantajan verkkosivut käyttäen sen teknisenä alustana WordPress-sisällönhallintajärjestelmää.</p> <p>Opinnäytetyö toteutettiin Suomalainen Putkimies Oy:lle, joka tarjoaa kiinteistöhoitoon liittyviä palveluita. Projekti alkoi elokuussa 2018 potentiaalisten toimeksiantajien prospektoinnilla ja päättyi verkkosivuston julkaisemiseen marraskuussa 2018.</p> <p>Opinnäytetyössä tutkittiin verkkosivustoja ja niiden kehitykseen käytettäviä teknologioita, avointa lähdekoodia ja esitettiin case-esimerkkinä avoimeen lähdekoodiin perustuva WordPress-sisällönhallintajärjestelmä. Taustateorian lähteinä käytettiin kirja- ja verkkojulkaisuja.</p> <p>Toimeksiannon toteutuksessa graafisessa suunnittelussa käytettiin Adobe XD -ohjelmaa ja toteutuksessa WordPress-sisällönhallintajärjestelmää ja Advanced Custom Fields -lisäosaa.</p> <p>Lopputuloksena opinnäytetyössä syntyi kolme samaa graafista ulkoasua käyttävää dynaamista verkkosivustoa, joita toimeksiantaja pystyy päivittämään WordPress-sisällönhallintajärjestelmän kautta.</p>	
Asiasanat WordPress, WWW-sivut, verkko-ohjelmointi, avoin lähdekoodi	

Sisällys

1	Johdanto	1
1.1	Keskeiset käsitteet	2
2	Verkkosivustojen toteutus avoimen lähdekoodin avulla	3
2.1	Verkkosivustot.....	3
2.1.1	HTML	4
2.1.2	CSS	5
2.1.3	JavaScript	6
2.1.4	PHP	7
2.2	Avoimen lähdekoodin ohjelmistot.....	9
2.2.1	Historia lyhyesti.....	9
2.2.2	Miksi avoimen lähdekoodin ohjelmia kehitetään.....	10
2.2.3	Avoimen lähdekoodin ohjelmistot ohjelmistokehityksessä	11
2.3	Case esimerkki avoimen lähdekoodin ohjelmistosta: WordPress	12
2.3.1	WordPressin historia	12
2.3.2	Teemat.....	13
2.3.3	Lisäosat	14
3	Opinnäytetyön toimeksianto	15
3.1	Yrityksen esittely	15
3.2	Tavoite, ongelmat ja kehittämistehtävä	15
4	Toimeksiannon toteutus	17
4.1	Vaatimusmäärittely	17
4.2	Sisällönhallintajärjestelmän valinta.....	18
4.3	Graafinen suunnittelu	18
4.4	Verkkosivuston sisältö.....	20
4.5	Käytetyt lisäosat.....	21
4.6	Staattiset sivupohjat	22
4.7	Konvertointi WordPress teemaksi	24
4.8	Testaaminen	26
5	Pohdinta.....	28
	Lähteet	29
	Liitteet.....	32
	Liite 1. Verkkosivuston etusivun graafinen ulkoasu	32

1 Johdanto

Nykyään helppokäyttöiset ja selkeät kotisivut toimivat tärkeänä myyntikanavana yritykselle. Verkkosivustojen kautta palveluita etsivät asiakkaat tekevät päätöksiä, mikäli kyseinen yritys sopii heidän tarpeisiinsa. Vaivattomasti löydettävissä olevat tiedot yrityksen palveluista sekä hintatiedoista ovat avainasemassa uusien asiakkuuksien kartuttamiseen. Näyttävät, helposti lähestyttävät kotisivut luovat yritykselle uskottavuutta sekä parantavat verkossa tapahtuvaa asiakkaan ja yrityksen välistä vuorovaikutusta.

Opinnäytetyön tavoitteena on suunnitella ja toteuttaa kiinteistönhoidon monialayritykselle uudet verkkosivut. Verkkosivujen kautta toimeksiantaja voi saavuttaa uusia kohtaamisia potentiaalisten asiakkaiden kanssa. Toimeksiantajana opinnäytetyölle toimii Suomalainen Putkimies Oy. Yritys tarjoaa erilaisia kiinteistöhoitoon liittyviä palveluita.

Opinnäytetyön taustateorian tarkoituksena on antaa lukijalle ymmärrys verkkosivuista ja verkkosivujen kehitykseen käytettävistä teknologioista sekä avoimesta lähdekoodista. Taustateorian tarkoituksena on vastata tutkimuskysymyksiin:

- Mitä ovat verkkosivut ja niiden kehittämiseen käytettävät teknologiat?
- Miksi avoimen lähdekoodin ohjelmistoja kehitetään?
- Miksi avoimen lähdekoodin työkaluja käytetään ohjelmistokehityksessä?

Taustateoriaosuudessa käsitellään myös WordPress-sisällönhallintajärjestelmää. Järjestelmä toimii keskeisenä osana opinnäytetyössä toteutettavaa verkkosivustojen uudistamista.

Opinnäytetyön toiminnallisessa osuudessa kerrotaan verkkosivustojen toteutuksesta toimeksiantajalle WordPress-sisällönhallintajärjestelmällä ja ACF-lisäosalla. Toimeksiannon toteutus koostuu sivuston graafisesta suunnittelusta, toiminnallisuuksien määrittelystä ja itse toteutuksesta. Kappaleessa kerrotaan myös verkkosivuston testaamisesta eri toteutusvaiheissa.

1.1 Keskeiset käsitteet

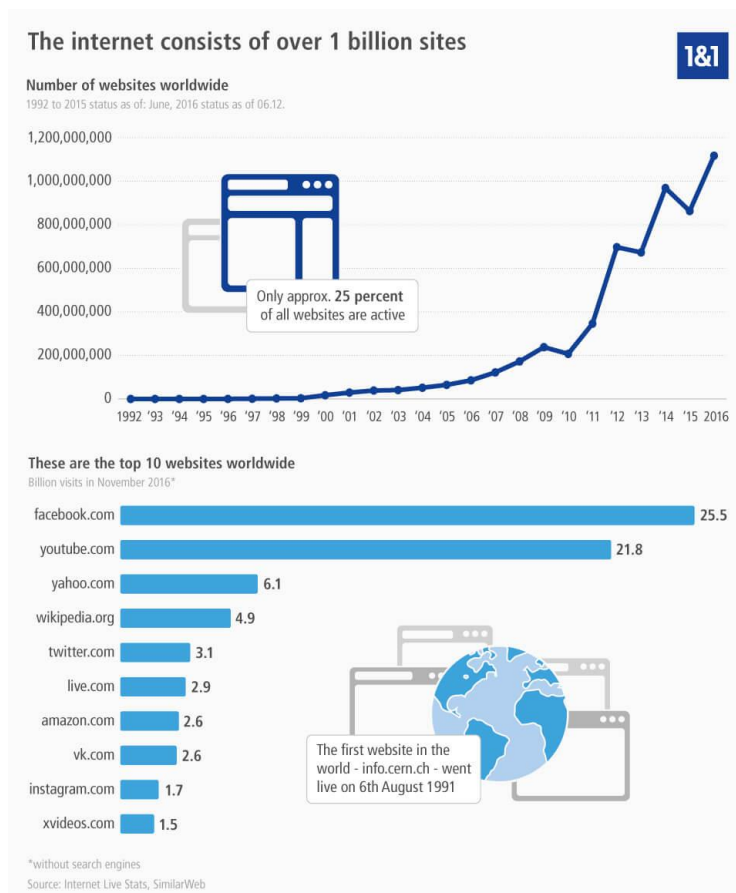
AJAX-kutsu	Verkkosivulla tapahtuva asynkroninen pyyntö, joka ei johda sivun uudelleenlataukseen.
Appletit -lisäosat	Pieniä sovelluksia, joita käytetään osana isompaa sovellusohjelmaa.
Emulaattori	Laitteisto tai ohjelma, jota mahdollistaa ohjelmien tai käyttöjärjestelmien muunlaisella laitteella kuin mille ne on alun perin suunniteltu.
Flash	Internet-sivuilla käytetty sovellusohjelma videoiden ja äänen toistamiseen.
GPLv2-lisenssi	Vapaiden ohjelmistojen julkaisemiseen tarkoitettu lisenssi, joka antaa kenelle tahansa oikeuden käyttää, kopioida, muuttaa ja jakaa ohjelmia ja niiden lähdekoodia.
Hakukone	Web-pohjainen ohjelma, joka indeksoi internetistä löytyviä verkkosivustoja ja tarjoaa käyttäjille tiedonhakuja.
Hyperteksti	Teksti, joka sisältää linkin muuhun dokumenttiin tai tekstiin.
Kuvapankki	Verkkosivustoja, joista voidaan ladata tai ostaa kuvia.
Linuksen laki	Linuksen lailla selitetään yhteiskunnan toiminnassa kolmea perustoimintoa: eloonjäämistä, sosiaalisia suhteita ja viihdettä.
World Wide Web	World Wide Web tai WWW tarkoittaa maailmanlaajuisesta internet-tietojärjestelmää.
World Wide Web Consortium	Kansainvälinen yhteisö, joka kehittää ja määrittelee avoimia standardeja, joilla varmistetaan verkon pitkäaikainen kasvu.

2 Verkkosivustojen toteutus avoimen lähdekoodin avulla

Tässä kappaleessa kerrotaan verkkosivustoista ja niiden kehittämiseen käytettävistä teknologioista. Kappaleessa käsitellään myös avoimen lähdekoodin ohjelmistoja ja kerrotaan Case-esimerkkinä avoimella lähdekoodilla julkaistusta WordPress-sisällönhallintajärjestelmästä, joka toimii olennaisena osana verkkosivuston toteutusta.

2.1 Verkkosivustot

Verkkosivustot eli web-sivu tai www-sivu ovat kokoelma Internetissä julkaistuja hypertextidokumentteja. Verkkosivut ovat tiedostoja, jotka selaimen käyttäjä näkee mennessään verkko-osoitteeseen kirjoittamalla hakukenttään verkkosivun osoitteen, napsauttamalla linkkiä tai laittamalla kyselyn hakukoneeseen. Verkkosivut voivat sisältää kaikenlaista tietoa: kuvia, tekstejä ja grafiikkaa. Verkkosivustoilla on yleensä kotisivu, jolla esitetään verkkosivuston tarjonta. Jokaisella verkkosivustolla on myös oma identifioiva osoite, jota voidaan käyttää internet-yhteydellä selaimen kautta. (BusinessDictionary; Freeservers.) Kuvassa 1 esitetään verkkosivustojen määrällistä kehitystä vuoden 1992 jälkeen ja kymmenen suosituinta verkkosivustoa kävijämäärän perusteella marraskuussa 2016.



Kuva 1. Verkkosivustojen määrä maailmassa ja kymmenen suosituinta sivustoja (1&1 IONOS.)

Verkkosivut jaetaan staattisiin ja dynaamisiin sivuihin. Staattiset verkkosivut ovat tiedostoja, joiden sisältö esitetään aina samanlaisena kaikille sivustovierailijoille. Staattisten verkkosivujen sisältö on muuttumatonta ja sisältö muuttuu ainoastaan, kun lähdekoodiin tehdään muokkauksia. Dynaamiset verkkosivustot ovat sivustoja, joiden ulkoasu voidaan esittää vaihtelevana eri sivustovierailijoille. Dynaamiset verkkosivustot rakennetaan sivuston lähdekoodista sivustovierailijan selaimen lähettämästä pyynnöstä. Ulkoasu, joka sivustovierailijalle näytetään, voi vaihdella käyttöjärjestelmän, selaimen tai muiden vaikuttavien tekijöiden vaikutuksesta. (Bulgar 2016.)

Verkkosivustoilla on monia käyttötarkoituksia. Sivustoja käytetään tiedon jakamisesta liiketoiminnan kehittämiseen ja vapaa-aikaan. Googlen hakukone on isolle osalle käyttäjistä ensimmäinen informaationsatama ja melkein kaikelle tiedolle on olemassa oma verkkosivunsa internetissä. Liiketoiminta on myös viimeisen kymmenen vuoden aikana siirtynyt enemmän ja enemmän verkkoon. Nykyisin käyttäjät ostavat tuotteita ja palveluita Internetistä halvempien hintojen ja käytännöllisyyden takia. Vapaa-aika on myös merkittävä osa Internetin käyttötarkoitusta – pelattavista peleistä ja kasinoista aina verkon 'surffaamiseen'. (Sadie 2011.) Verkon surffaamisella tarkoitetaan yleisesti ajankäyttöä internetissä.

Verkkosivustot rakentuvat pääasiallisesti HTML, CSS, JavaScript ja PHP -tiedostoista. Näistä teknologioista kerrotaan lisää seuraavissa alaluvuissa.

2.1.1 HTML

Hyper Text Markup Language tai HTML on web-sivujen merkitsemiseen ja määrittämiseen käytetty syntaksi. HTML koostuu sarjasta lyhyitä kuvauksia, joita kutsutaan tunnisteksiksi. Tunnisteet, joista HTML koostuu muodostavat verkkosivuilla näkyvät osiot. Osioissa voidaan näyttää sisältötekstiä, kuvia ja videoita. (Shannon 2012; West 2012, 7-8.)

Kuten aiemmin on mainittu, HTML-syntaksi on kokoelma sanoja ja symboleita. Nämä sanat ja symbolit muodostavat tiedoston, jonka selain kääntää näkyvään muotoon. HTML-tiedostoissa käytettävät tunnistekset eivät näy, kun verkkosivustoa tarkastellaan selaimen kautta. Selain kääntää nämä tunnistekset verkkosivuilla esitettäväksi rakennepalikoiksi, joita kokonaisuutena kutsutaan verkkosivuiksi. (Shannon 2012.)

HTML kehittyi jatkuvasti. Viimeisin julkaistu kehitysversio on HTML5 (kuva 2), joka tarjoaa uusia toiminnallisuuksia web-sivujen kehitykseen. Päivitettyihin toiminnallisuuksiin lukeutuvat muun muassa LocalStorage käyttö, jonka avulla kehittäjät voivat tallentaa tietoa sivus-

tovierailijan käyttämälle laitteelle. HTML5 mahdollistaa myös videoiden näyttämisen selaimessa ilman lisäosia. Sovelluskehityksen alalla HTML5 verkkosivujen kehittämistä kutsutaan webin kehittämiseksi avoimilla tekniikoilla. (West 2012, 8; Wikipedia.)

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

Kuva 2. HTML5-dokumentti

2.1.2 CSS

Cascading Style Sheet tai CSS on suunnittelukieli, jonka tarkoituksena on määritellä, miten web-sivuja esitetään. CSS määrittelee web-sivuilla käytettyjen HTML-elementtien ulkoasua. CSS avulla hallitaan tekstien väriä, fonttien tyyliä, lomakkeiden ulkoasua ja kaikkia ulkoasua määrittäviä graafisia piirteitä. CSS:llä voidaan määritellä HTML-elementtien käyttäytymistä eri kokoisilla näytöillä. (TutorialsPoint.)

World Wide Web Consortium tai W3C ylläpitää CSS-kielen määritelmiä. W3C-organisaatiossa työskentelevä CSS-työryhmä luo dokumentteja, jotka ratifioidaan suosituksiksi. W3C ei ylläpidä CSS suunnittelukieltä vaan organisaatiot ja yritykset luovat kyseisen ohjelmiston. (TutorialsPoint.)

Viimeisin julkaistu versio on CSS3, joka tarjoaa uusia toiminnallisuuksia. Osa näistä ominaisuuksista on suunniteltu korvaamaan Flash ja Appletit -lisäosat. Uusia CSS3 ominaisuuksia ovat muun muassa 2D ja 3D -animaatiot ja Media Queryt. Media Queryillä voidaan verkkosivuja suunnitella responsiiviseksi, joka tarkoittaa, että verkkosivut mukautuvat käytettävän näyttökoon mukaan. Verkkosivuston käyttäjän ei tarvitse erikseen suurentaa ja pienentää näkymää, vaan verkkosivustolla näytettävät elementit mukautuvat optimaalisesti erikokoisille laitteille. (Wikikirjasto.) Kuvassa 3 esitetään CSS:llä määritettyjä ulkoasullisia määritteitä verkkosivustolle.


```

@font-face {
  font-family: 'Liberations-sans';
  src: url(fonts/LiberationSans-Bold.ttf);
  src: url(fonts/LiberationSans-BoldItalic.ttf);
  src: url(fonts/LiberationSans-Italic.ttf);
  src: url(fonts/LiberationSans-Regular.ttf);
}

h1, h2, h3, h4 {
  color: #00A5FD;
}

* {
  margin: 0;
  padding: 0;
  max-height: 1000000px;
  box-sizing: border-box;
}

body {
  font-family: 'Liberation-sans', sans-serif;
  font-size: 18px;
}

```

Kuva 3. CSS3 määrittelyä

2.1.3 JavaScript

JavaScript tai JS on komentosarjapohjainen ohjelmointikieli. Sitä käytetään pääasiallisesti kehittämään käyttäjän ja verkkosivuston välistä vuorovaikutusta. JavaScriptillä voidaan luoda dynaamista sisältöä kuten animaatioita, kuvaesityksiä ja muita sivustovierailijalle näytettäviä toiminnallisuuksia. JavaScriptin ideana on, että käyttäjä pystyy käyttämään verkkosivun toiminnallisuuksia, ilman sivun uudelleenlataamista. (Morris 2018; Techopedia.)

JQuery on kevyt JavaScript-kirjasto. JQueryn tarkoitus on helpottaa JavaScriptin käyttämistä verkkosivuilla valmismetodeilla, jotka normaalisti vaativat usean rivin perinteistä JavaScript-koodia. JQuery yksinkertaistaa useita monimutkaisia JavaScript-toimintoja kuten AJAX-kutsut ja verkkosivuston manipuloinnin valmismetodien avulla. (W3Schools). Kuvassa 4 esitetään yksinkertainen esimerkki JavaScriptin käytöstä JQueryn valmismetodeilla.

```

$(function() {
    initMobileMenu();
    hideLastDivider();
});

function initMobileMenu() {
    $('#toggle').click(function () {
        $('.navbar').slideToggle();
    });
}

function hideLastDivider() {
    $('.divider-wide').last().hide();
}

```

Kuva 4. JavaScriptin käyttöä JQueryn valmismetodeilla

2.1.4 PHP

PHP: Hypertext Preprocessor tai PHP on avoimen lähdekoodin ohjelmointikieli, jota käytetään verkkokehityksessä. Kielen tavoitteena on antaa kehittäjille mahdollisuus kirjoittaa nopeasti ja helposti dynaamisesti luotavia sivuja. PHP määritetään skriptikieliseksi, koska PHP-ohjelmat suoritetaan palvelinpuolella ja niiden sisältö lähetetään käyttäjälle vasta suorittamisen jälkeen. (NTC Hosting; PHP.net.)

PHP on suunniteltu toimimaan vuorovaikutuksessa HTML:n kanssa. HTML-sivuilla PHP-ohjelmakoodi esitetään PHP:ssä käytettyjen tunnisteiden sisällä (kuva 5). PHP-kääntäjä kääntää kaiken tunnisteiden sisällä olevan ohjelmakoodin ennen selaimelle lähettämistä. Kaikki ohjelmakoodi, joka ei ole PHP-tunnisteiden sisällä, lähetetään selaimelle alkuperäisessä esitettyssä muodossa. (NTC Hosting.)

```

<div class="company-information-box">
  <div class="company-information-box-heading">
    <h2>
      <?php the_field('company_heading', 'options'); ?>
    </h2>
  </div>
  <?php if( have_rows('company_information', 'options') ): ?>

  <div class="company-info-boxes">

    <?php while( have_rows('company_information', 'options') ): the_row();

    $heading = get_sub_field('company_box_heading', 'options');
    $content = get_sub_field('company_box_text', 'options');

    ?>

    <div class="info-box">
      <h2 class="info-box-heading">
        <?php echo $heading; ?>
      </h2>
      <p>
        <?php echo $content; ?>
      </p>
    </div>

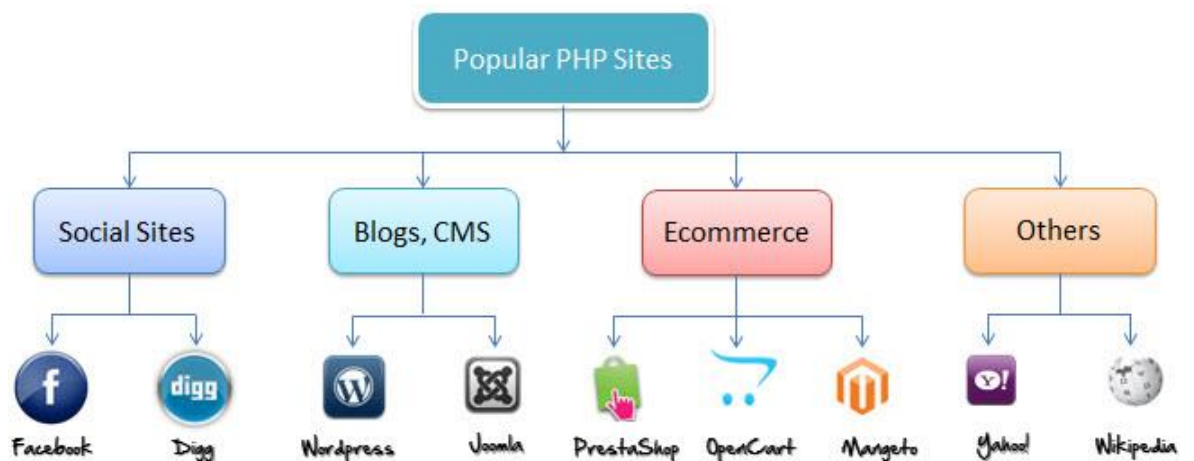
    <?php endwhile; ?>

  </div>
</div>

```

Kuva 5. PHP-ohjelmointikoodia HTML-dokumentissa

Maailmassa on yli 20 miljoonaa PHP:llä kehitettyä verkkosivustoa ja ohjelmaa (Guru99). Kuvassa 6 esitetään suosituimpia PHP:tä käyttäviä verkkosivustoja ja ohjelmia.



Kuva 6. Suosituimpia PHP:tä käyttäviä verkkosivustoja ja ohjelmia (Guru99.)

2.2 Avoimen lähdekoodin ohjelmistot

Avoimen lähdekoodin ohjelmistot (OSS) ovat ohjelmistoja, jotka on julkaistu kaiken yleisön nähtäväksi, kopioitavaksi, muokattavaksi ja uudelleenjaettavaksi. Avoimen lähdekoodin ohjelmistoja voidaan käyttää maksamatta rojalteja tai maksuja, mutta alkuperäisten kehittäjien työtä on kunnioitettava hyvän maun mukaisesti. Avoimen lähdekoodin ohjelmistoja kehittävät ja ylläpitävät avoimen lähdekoodin järjestöt. Näihin järjestöihin kuuluvat sekä kuulumattomat kehittäjät, että muut yksityishenkilöt osallistuvat omalla työpanoksellaan avoimen lähdekoodin projektien kehittämiseen. (Fitzgerald, Kesan, Russo, Shaikh & Succi 2011, 24.)

Avoimen lähdekoodin ohjelmistojen merkitys teknillisessä mielessä on se, että kuka tahansa pystyy osallistumaan ohjelman tai projektin kehitykseen. Termi 'avoin' kuvastaa ideologiaa, jolla avoimen lähdekoodin ohjelmistoja kehitetään ja rakennetaan. Avoimen lähdekoodin ohjelmistot ovat myös todistaneet, että monimutkainen ohjelmistokoodi voidaan rakentaa, ylläpitää, kehittää ja laajentaa monien kehittäjien yhteistyöllä, rakenteettomissa yhteisöissä. (Weber 2004, 2-3)

Tunnetuimpia avoimen lähdekoodin ohjelmistoja ovat Linux, Apache, Mozilla ja OpenOffice (Fitzgerald ym. 2011, 24.) WordPress-sisällönhallintajärjestelmä on myös yksi tunnetuimmista avoimen lähdekoodin ohjelmistoista. WordPressistä kerrotaan lisää tämän opinnäytetyön aikana alaluvussa 2.3, sillä se toimii olennaisena osana toiminnallisen osuuden toteutusta.

2.2.1 Historia lyhyesti

1970-luvulla ohjelmistoja ja laitteita kehittävät yritykset laativat yhä enemmän tekijänoikeuksia lähdekoodin suojaamiseen ja ohjelmistojen ja laitteiden käyttöoikeiden pidättämiseen. Omistusoikeusohjelmistot hallitsivat teknologiamarkkinoilla. (Singh 2018.)

1980-luvun alkupuolella Richard Stallman, joka työskenteli Massachusettsin teknillisessä korkeakoulussa tutkijana, uskoi vapaan ohjelmayhteisön rakentamiseen. Hän aloitti GNU-hankkeen, jonka tarkoituksena oli luoda vapaa käyttöjärjestelmä. GNU-projekti yhdistyi 1990-luvun alussa Linuxin kanssa. Vapaa käyttöjärjestelmä, josta Stallman unelmoi, oli viimein syntynyt näiden kahden hankkeen yhdistelmästä. (Singh 2018.)

Eric Raymond, ilmaisten ohjelmistojen kannattaja, aloitti yhteistyön NetScape-yrityksen kanssa vuonna 1997. Viisi vuotta yhteistyön aloittamisesta Mozilla Firefox julkaistiin.

Mozilla Firefox on avoimeen lähdekoodiin perustuva ohjelmisto ja eräs 2000-luvun suosituimmista selaimista. (Singh 2018.)

Viimeisen kahdenkymmenen vuoden aikana, avoimen lähdekoodin projektit ovat kasvaneet osaksi yleistä tietoutta. Avoimen lähdekoodin yhteisöt ovat aktiivisia ja tutkimuksen mukaan 98% yrityksistä käyttää avoimen lähdekoodin ohjelmistoja tänä päivänä. (Singh 2018.)

2.2.2 Miksi avoimen lähdekoodin ohjelmia kehitetään

Avoimen lähdekoodin ohjelmistoja kehitetään, koska avoin kehittämismalli tuottaa laadukkaita ohjelmia. Ohjelmiston parissa työskentelevät kehittäjät toimivat myös testaaajina. Linuksen lain mukaisesti: empiirisesti avoimen lähdekoodin taipumus on tuottaa laadukkaampaa ohjelmistoa kuin vaihtoehtoisten kehitysmetodien. Suljetuissa ohjelmistoissa ainoastaan ohjelman kanssa toimivat kehittäjät voivat havaita mahdollisia ohjelmistovirheitä. Avoimen lähdekoodin ohjelmistoissa on enemmän silmäpareja, jotka pystyvät diagnosoimaan, havaitsemaan ja korjaamaan esiintyviä ongelmia. (Balter 2015.)

Yksi tärkeä syy avoimen lähdekoodien ohjelmistojen käyttämiselle ovat pienemmät kokonaiskustannukset suljettuihin ohjelmistoihin verrattuna. Suljetuissa ohjelmistoissa yritykset maksavat tuotteen kehittämisestä, ylläpidosta ja tuesta. Avoimen lähdekoodin ohjelmistoissa nämä kustannukset ovat huomattavasti pienempiä, sillä ratkaisut perustuvat jo olemassa oleviin kehitettyihin ohjelmistoihin. Yleisesti avoimeen lähdekoodiin perustuvien ratkaisujen käyttöönotto on edullisempaa, koska ohjelmistojen siirto, mukautus ja jatkokehitys on suhteellisesti edullisempaa. (Balter 2015.) Kuvassa 8 verrataan maksullisten ohjelmistojen ja avoimen lähdekoodiin perustuvien ohjelmistojen kustannuksia lisensoinnin, markkinoinnin, ylläpidon ja tuen osa-alueilla.

Cost	Open source	Proprietary
Licensing	No	Yes
Implementation	Yes	Yes
Maintenance	Yes	Yes
Support	Yes	Yes

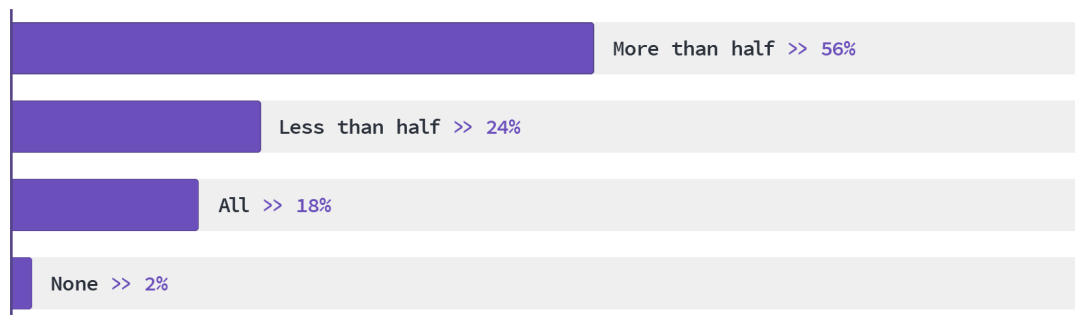
Kuva 7. Avoimen lähdekoodin ja maksullisten ohjelmistojen kustannuksien osa-alueita (Balter 2015.)

Avoimen lähdekoodin ratkaisujen käyttämisellä on kolme etua: avoimen lähdekoodin ohjelmistoissa hyödynnetään kaikkien maailmassa kyseisen ohjelmiston kehittämisen parissa toimivien kehittäjien tietämystä. Kehittäjien määrä on suurempi, joten kehityksessä käytettävä tiedon määrä on laajempi. Avoimen lähdekoodin ohjelmistot myös mukautuvat käyttötapauksien mukaan. (Balter 2015.)

2.2.3 Avoimen lähdekoodin ohjelmistot ohjelmistokehityksessä

Avoimen lähdekoodin ohjelmistoja käytetään ohjelmistokehityksessä, sillä ohjelmistojen taustalla toimii kokonaisia yhteisöjä. Ohjelmistot kehittyvät nopeasti ja ohjelmistossa havaitut virheet korjataan nopeasti yhteisön jäsenien toimesta. Jopa 98% kehittäjistä käyttää avoimen lähdekoodin työkaluja ohjelmistokehityksessä. (Dodd 2017.) Kuvassa 8 esitetään GitLab-verkkopalvelun toteuttaman kyselyn tuloksia avoimen lähdekoodin työkalujen käytämisestä.

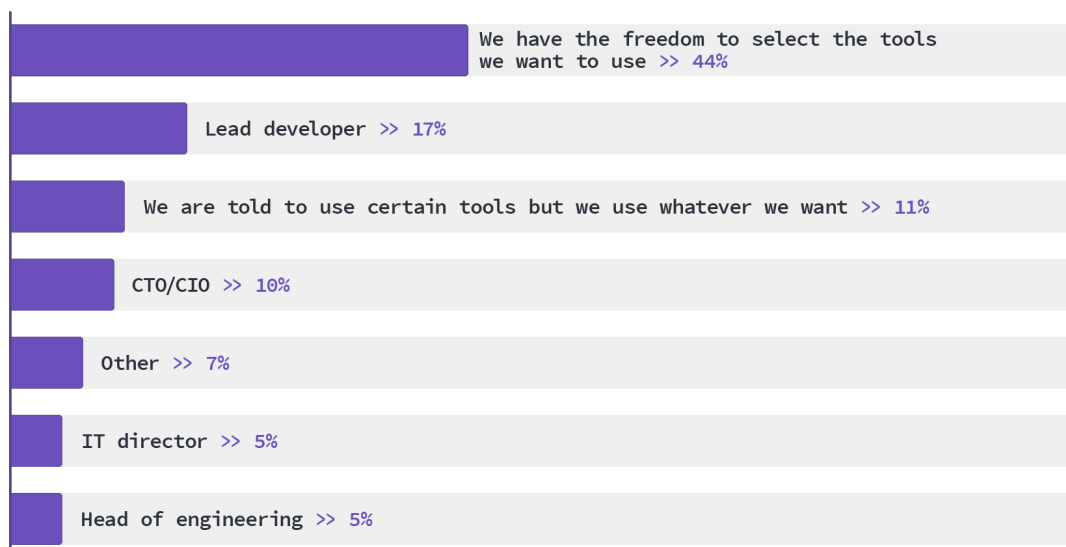
APPROXIMATELY WHAT PERCENTAGE OF YOUR TEAM'S DEVELOPER TOOLS ARE OPEN SOURCE?



Kuva 8. Kysely avoimen lähdekoodin työkalujen käytöstä ohjelmistokehityksessä (Dodd 2017.)

Avoimen lähdekoodin työkaluja ja ohjelmistoja käytetään ohjelmistokehityksessä, koska niitä pidetään luotettavina lähdekoodin ollessa avoin ja kaikkien tarkastettavissa. Kehittäjät voivat itse analysoida ja tarkistaa ohjelmistokoodin turvallisuuden ja tehdä korjauksia sekä parannuksia. Avoimen lähdekoodin ohjelmistot eivät myöskään sitouta kehittäjää ennalta määritettyihin ohjelmistossa käytettäviin toiminnallisuuksiin. Ohjelmia voidaan jatkokehittää ja integroida liiketoiminnan tarpeiden mukaan. GitLabin-verkkopalvelun toteuttaman kyselyn (kuva 9) mukaan, melkein puolella kehittäjistä on vapaus valita kehityksessä käytettävät työkalut ja ohjelmistot. (Dodd 2017.)

WHO IN YOUR ORGANIZATION DECIDES WHICH TOOLS YOU USE?



Kuva 9. Ketkä määrittelevät yrityksissä käytettävät työkalut (Dodd 2017.)

2.3 Case esimerkki avoimen lähdekoodin ohjelmistosta: WordPress

WordPress on PHP-ohjelmointikielellä kirjoitettu avoimeen lähdekoodiin perustuva GPLv2-lisenssillä julkaistu verkkosivustojen sisällönhallintajärjestelmä (CMS). (iTheme; WordPress.org.) Sisällönhallintajärjestelmä tarkoittaa ohjelmistosovellusta, jonka avulla käyttäjä voi muokata, luoda, tuottaa ja julkaista digitaalista sisältöä verkkosivuilla (Sistecore).

2.3.1 WordPressin historia

Vuonna 2001 ranskalainen ohjelmoija Michael Valdrighi kehitti PHP-pohjaisen blogialustan nimeltään b2/cafelog. Valdrighi kuitenkin lopetti järjestelmän kehityksen jo seuraavana vuonna. Tammikuussa 2003 yliopisto-opiskelija Matt Mullenweg yhdessä Mike Littlen kanssa kopioivat b2/cafelog -blogialustan uudeksi projektiksi. Mullenweg ja Little julkaisivat ensimmäisen version uudesta projektista vuonna 2003. Järjestelmää kutsutiin WordPressiksi. (Wilson 2013.)

Toukokuussa 2004 julkaistiin WordPressin versio 1.2, joka lisäsi toiminallisuuden lisäosien tuelle. Toukokuussa 2005 WordPress-sisällönhallintajärjestelmään lisättiin teemat. Teemat ovat joukko tiedostoja, jotka hyödyntävät WordPress-taustaohjelman toimintoja ja valmiuksia luomalla verkkosivustolle graafisen ulkoasun. (Wilson 2013.)

Vuosien 2007 ja 2009 välisenä aikana WordPress koki huomattavia parannuksia mm. WordPressin sisäisiä toimintoja parannettiin. Käyttäjät pystyi nyt vaihtamaan WYSIWYG

(What You See Is What You Get) ja tekstieditorin välillä. Uusien päivityksien mukana tuli myös mahdollisuus piilottaa julkaisuja hakukoneilta ja määrittellä sivuston etusivuksi mikä tahansa WordPress-sisällönhallintajärjestelmässä luotu sivu. Dataa pystyttiin myös vie-
mään ja tuomaan järjestelmässä saumattomasti. Vuoden 2009 loppuun mennessä WordPress-sisällönhallintajärjestelmässä oli toiminnallisuudet, joilla käyttäjät pystyivät lataamaan järjestelmään useita tiedostoja kerralla, kustomoimaan hallintanäkymää ja päi-
vittämään käytettäviä lisäosia yhdellä napin painalluksella. (Muriuki 2016.)

Vuonna 2011 WordPressin osuus kaikista maailman verkkosivustoista on 12%. Tämä luku nousee häkellyttävään 22% seuraavan kolmen vuoden aikana. (Kinsta.)

Viidessätoista vuodessa WordPressin kasvu on ollut valtavaa: 32% kaikista verkkosivuista käyttävät WordPress-sisällönhallintajärjestelmää. WordPress hallitsee maailmanlaajuisia CMS-markkinaa 50-60% osuudella ja on kaikkien aikojen suosituin CMS-järjestelmä seitsemännen vuoden peräkkäin. (Karol 2018.)

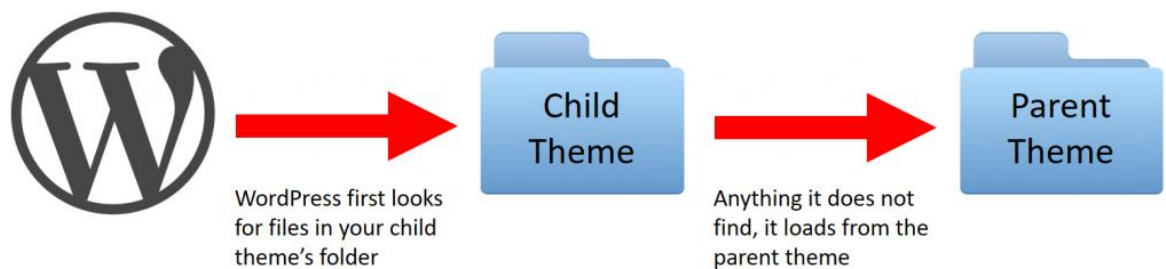
2.3.2 Teemat

WordPress -julkaisujärjestelmässä käytetään teemoja ulkoasun muokkaamiseen. Teemat ovat kokoelma ennalta määritettyjä ulkoasua muokkaavia tyyli- ja rakennetiedostoja, jotka muodostavat verkkosivuston graafisen ulkoasun. Teema voidaan asentaa suoraan palvelimella 'wp-content' kansioon, tai käyttämällä WordPress -sisällönhallintajärjestelmän sisäänrakennettua teeman asennustoimintoa. Asennuksen jälkeen WordPressin sisällönhallintajärjestelmän hallintapaneelissa valitaan käytettäväksi asennettu teema, jonka määritteiden mukaisena verkkosivuston graafinen ulkoasu esitetään. (WordPress.org.)

Teemoja on kahdenlaisia: valmis- ja räätälöityjä teemoja. Monissa webhotelleissa on WordPress valmiiksi asennettuna, jolloin voidaan käyttää valmisteemaa, jota voidaan laajentaa halutuilla lisäosilla. Valmisteemoja löytyy kohtuuhinnalla ja jopa ilmaiseksi. Monet valmisteemat sisältävät kuitenkin turhia toiminnallisuuksia. Räätälöidyt teemat taas poikkeavat valmisteemoista. Räätälöidyt teemat rakennetaan asiakkaiden toiveiden mukaisesti vastaamaan heidän tarpeitaan verkkosivustona. Räätälöidyissä teemoissa etuna on se, että niitä voidaan optimoida asiakkaiden tarpeita varten. Kaikki toiminnallisuudet suunnitellaan etukäteen ja myös ulkoasu suunnitellaan vastaamaan asiakkaan toiveita. (Laukkarinen 2017; Leiniö 2015.)

Lapsiteemat (Child Theme) ovat WordPress-sisällönhallintajärjestelmässä teemoja, jotka perivät toisen teeman toiminnot ja muotoilun. Teemoja, joilta toiminnot periytetään, kutsutaan isäteemoiksi. Lapsiteemat ovat suositeltu tapa muokata olemassa olevaa teemaa, sillä lapsiteemaan tehdyt muokkaukset eivät korvaannu isäteeman päivityksen yhteydessä. Kun teemoja muokataan lapsiteemoja hyödyntäen, on myös pienempi riski, että käyttäjä onnistuu hajottamaan käytössä olevan teeman. (Duò 2018.)

Kuvassa 10 esitetään lapsiteemojen toiminta. WordPress-sisällönhallintajärjestelmä lataa lapsiteeman kansiota olemassa olevat tiedostot. Tiedostot, joita ei löydy lapsiteeman kansiota, ladataan isäteemasta.



Kuva 10. Kuinka lapsiteemat toimivat (Duò 2018.)

2.3.3 Lisäosat

WordPress-julkaisujärjestelmässä käytettävät lisäosat ovat ohjeistoja, jotka määrittävät lisäominaisuuksia ja toimintoja. Ne ovat pääasiassa kooditiedostoja, joita voidaan käyttää WordPress-sisällönhallintajärjestelmän perustoimintojen jatkeena. WordPress-lisäosat ovat kirjoitettu PHP-ohjelmointikielellä ja integroituvat saumattomasti WordPress-julkaisujärjestelmään. (WPBeginner.)

Lisäosia on tarjolla tuhansia erilaisia: lisäosia löytyy yksinkertaisista toiminnoista kuten sivuston päivittämisestä aina tietoturvaan ja käyttäjänhallintaan asti. Ne käytännössä tarjoavat loputtomasti mahdollisuuksia verkkosivuston toimintojen laajentamiseen. (Aranovitch 2016.)

3 Opinnäytetyön toimeksianto

Opinnäytetyön toimeksiantona uudistettiin kolme verkkosivustoa toimeksiantajalle. Tarkoituksena oli parantaa toimeksiantajan verkkosivujen ulkoasua ja yksinkertaistaa verkkosivujen päivittämistä. Päivitettyjen, helposti lähestyttävien verkkosivujen tarkoituksena on hankkia uusia kohtaamisia toimeksiantajan ja potentiaalisten asiakkaiden välille.

3.1 Yrityksen esittely

Suomalainen Putkimies Oy on Suomessa toimiva kiinteistöhoiton monialayritys. Yrityksen toiminnan tarkoituksena on tarjota kohtuuhintaisia kiinteistöhoiton palveluita pääasiallisesti pääkaupunkiseudun alueella. Yritys tarjoaa palveluita sekä B2B (Business To Business, yritykseltä yritykselle), että B2C (Business To Customer, yritykseltä kuluttajalle) liiketoiminnassa. Yrityksen palvelutarjontaan kuuluvat putkiremontit, varaajan vaihdot, öljykattilan ja öljysäiliön purku, lämpökeskuksien asennukset, säädöt ja mittaukset sekä saneeraukset ja uudisrakentaminen.

Suomalainen Putkimies Oy on perustettu vuonna 2011. Nykyään yritys työllistää neljä vakituista työntekijää. Tämän lisäksi yritys työllistää kausityöntekijöitä ja harjoittelijoita. Yrityksen toimitusjohtajana toimii Pauli Ojala.

3.2 Tavoite, ongelmat ja kehittämistehtävä

Suomalainen Putkimies Oy oli tunnistanut tarpeen lähteä kehittämään verkkonäkyvyyttä. Yrityksen vanhat verkkosivustot eivät vastanneet tämän päivän standardeja rakenteen, teknologian ja mobiilikäytettävyyden osalta (kuva 11). Sivustojen sisältö myös kaipasi päivittämistä. Verkkosivustouudistuksen tavoitteena oli ulkoasun uudistaminen, käytettävyyden ja löydettävyyden parantaminen sekä verkkosivuston päivittämisen yksinkertaistaminen.

PUTKIMIES - ESPOO

Nuku yösi rauhassa - tilaa putkimies.



Tunnollinen putkimies

- Tuntihinta 50€/h+alv (24%)
- Minimilaskutus vain yksi tunti
- Kilometrikorvaus 46 senttiä/km (10-25€)
- 25% alennus vesikalusteiden suositushinnoista
- Nopea palvelu
- Takuukorjaukset 2 vuoden ajan

Suomalainen Putkimies Oy

(Y-tunnus 2391760-4)

Pauli.Ojala@suomalainenputkimies.fi

040/7474967

Työajat ja päivystys: Myös arki-illat & viikonloput

Kuva 11. Kuvakaappaus putkimies-espoo.fi vanhasta verkkosivusta

Opinnäytetyön tarkoituksena on uudistaa kolme verkkosivustoa Suomalainen Putkimies Oy:lle. Yrityksellä on tämän lisäksi huomattava määrä muita verkkosivustoja. Opinnäytetyössä kolme uudistettavaa verkkosivustoa ovat:

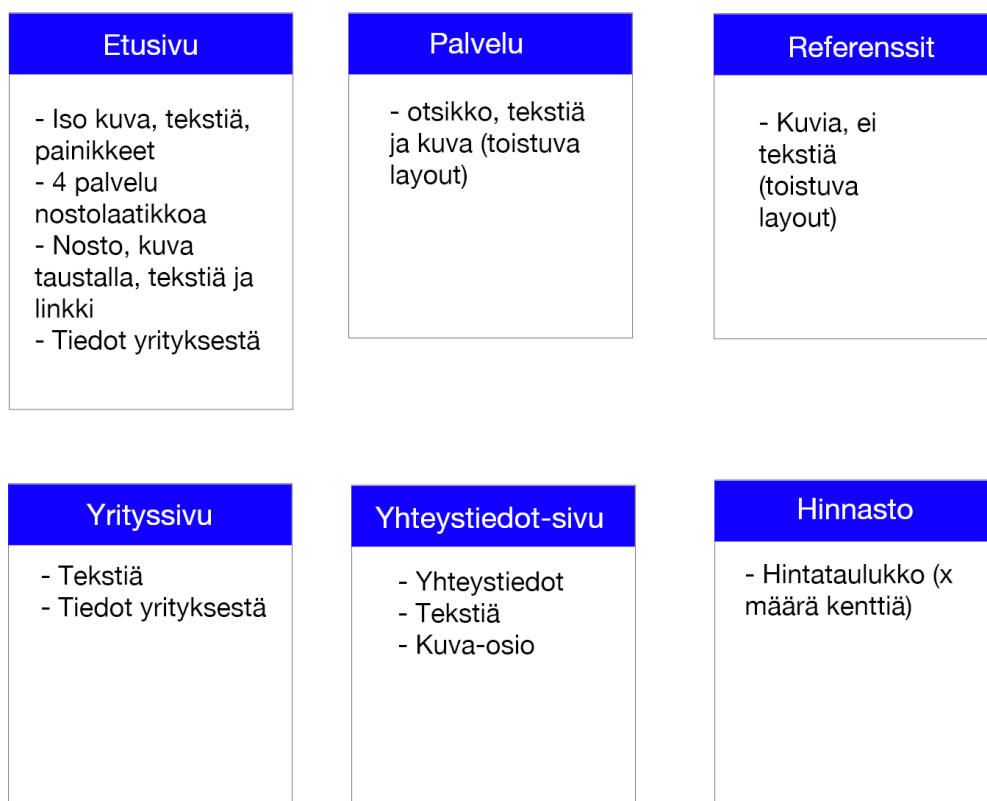
- putkimies-espoo.fi
- putkimies-helsinki.fi
- putkimies-vantaa.fi

Koska sivustoilla ei ole käytössä sisällönhallintajärjestelmää, päivitykset tehdään päivittämällä staattisia sivupohjia. Tämä siis tarkoittaa, että kaikki sivustoille tehtävät muutokset pitää tehdä suoraan ohjelmakoodiin. Tästä syystä verkkosivujen päivittäminen on hidasta ja vaatii päivityksiä tekevältä verkkosivuston ylläpitäjältä perustason ymmärrystä verkko-ohjelmoinnista. Verkkosivustoja ei ole myöskään optimoitu mobiilikäyttäjille

Uusien verkkosivujen kautta yrityksen potentiaaliset asiakkaat voivat löytää tietoja yrityksen palveluista, hintatiedoista ja ottaa yhteyttä yritykseen verkkosivustolla käytössä olevalla lomakkeella. Kaikki kolme uudistettavaa verkkosivustoa käyttävät samaa graafista ulkoasupohjaa ja niiden sisältö räätälöidään toimeksiantajan näkemysten mukaan.

4 Toimeksiannon toteutus

Uudistettavien verkkosivujen suunnittelun toteutus aloitettiin ensimmäisessä syyskuun alkupuolella pidetyssä tapaamisessa toimeksiantajan kanssa. Toimeksiantajan kanssa keskusteltiin verkkosivustojen teknisistä vaatimuksista ja ulkoasuista. Tapaamisessa keskusteltiin myös verkkosivujen rakenteesta ja tärkeistä tekijöistä verkkosivuilla. Uusien verkkosivujen rakenne haluttiin yksinkertaiseksi, mutta sivutyyppejä on useita erilaisia. Toimeksiantaja korosti tapaamisessa hinta- ja yhteydenottotietojen näkyvyyttä uusilla verkkosivuilla – näiden tietojen pitäisi olla helposti löydettävissä. Verkkosivustot rakentuisivat etu, palvelu, referenssi, yritys, yhteys ja hinnasto -sivuista (Kuva 12).



Kuva 12. Uusien sivujen rakenne

4.1 Vaatimusmäärittely

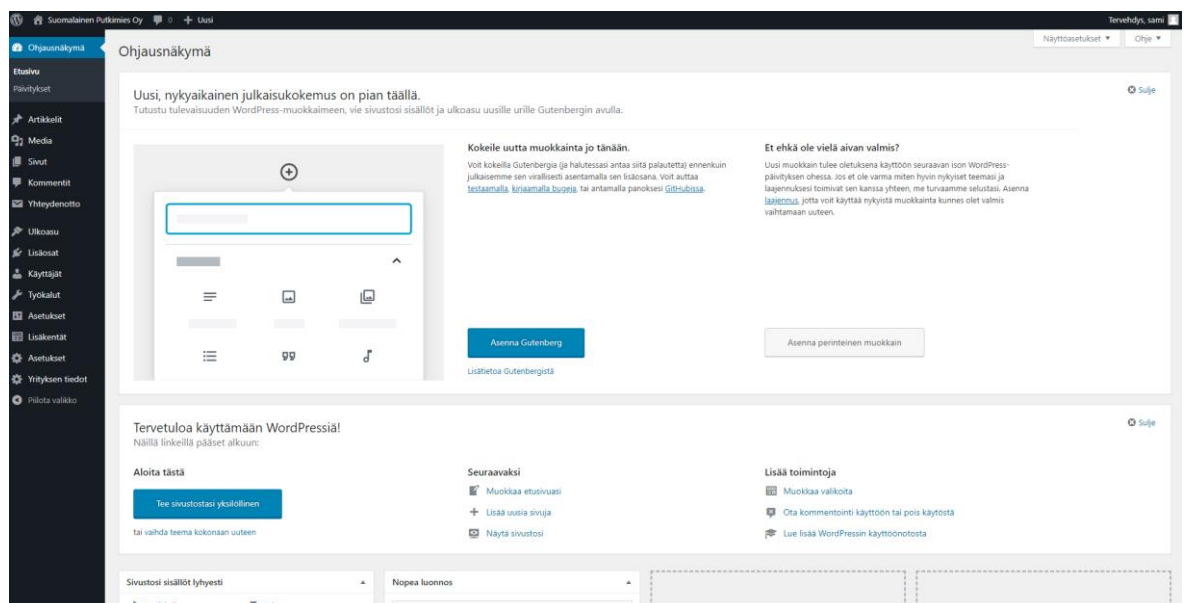
Verkkosivustojen käyttäjäryhmiä ovat yksityishenkilöt ja yritykset, jotka haluavat käyttää Suomalainen Putkimies Oy:n tarjoamia palveluita. Verkkosivustojen käyttöliittymän tulee olla helppokäyttöinen ja selkeä, jotta verkkosivustojen käyttäminen on käyttäjälle luonte-

vaa. Käyttäjän tai yrityksen ei oleteta tuntevan kaikkea yrityksen tarjoamia palveluita etukäteen, joten palvelutarjonnan tulee olla helposti löydettävissä verkkosivuilta. Hintatiedot ovat avainasemassa yrityksen toiminnassa, joten hintatietojen tulee olla helposti löydettävissä ja saatavilla kaikilla päätelaitteilla.

Verkkosivustojen tulee olla yhteensopiva yleisten käytössä olevien web-selainten (mm. Chrome, IE, Firefox, Safari) kanssa. Verkkosivuston on kyettävä palvelemaan käyttäjiä eri päätelaitteilla. Verkkosivustojen täytyy olla responsiivisia eli tarjota helppo ja vaivaton navigointi näyttökoosta riippumatta.

4.2 Sisällönhallintajärjestelmän valinta

Tarkoituksena oli toteuttaa helppokäyttöinen verkkosivusto, jota asiakas pystyy itse päivittämään. WordPressin tarjoamat toiminnallisuudet kattoivat verkkosivustolle asetetut vaatimukset. Tästä syystä WordPress valittiin verkkosivuston sisällönhallintajärjestelmäksi. WordPress-sisällönhallintajärjestelmän avulla sivustolle saadaan käyttöön hallintapaneeli (kuva 13), jonka kautta sivuston tietoja, asetuksia ja sivuja voidaan päivittää. WordPress-sisällönhallintajärjestelmän avulla toimeksiantaja pystyy myös tarpeen mukaan ottamaan käyttöön erilaisia lisäosia, joilla laajentaa sivuston toiminnallisuuksia.



Kuva 13. WordPress-sisällönhallintajärjestelmän hallintapaneeli

4.3 Graafinen suunnittelu

Toimeksiantaja antoi graafiseen suunnitteluun vapaat kädet. Näin ollen uuden verkkosivuston graafinen ulkoasu vastaa täysin opinnäytetyön aikana suunniteltua graafista ulkoasua. Rakenteellisesti graafisessa toteutuksessa otettiin huomioon toimeksiantajan esit-

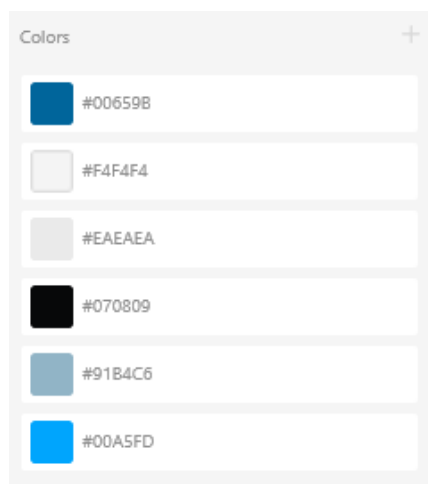
tämät tärkeät tiedot verkkosivustolla. Verkkosivustolta pitää löytyä helposti yhteystiedot, kuten puhelinnumero ja sähköposti. Hintatietojen täytyy olla myös helposti löydettävissä.

Verkkosivuston graafisen suunnittelun toteuttamisessa käytetään Adobe XD -ohjelmaa. Adobe XD ilmainen suunnittelijoille kehitetty suunnitteluohjelma käytettäväksi prototyyppien, verkkosivujen ja mobiilisovelluksien suunnittelemiseen. Tässä opinnäytetyössä käytettiin Adobe XD -ohjelmaa sen yksinkertaisuuden ja helppokäyttöisyyden takia.

Graafisen suunnittelun toteutuksen väripohjana käytettiin Suomalainen Putkimies Oy:n logosta (kuva 14) löytyviä värejä. Tämän lisäksi käytettiin muutamaa lisäväriä verkkosivuston kontrastin parantamiseksi. Hyvän kontrastin tarkoituksena on huomioida, että esimerkiksi heikkonäköiset tai värisokeat henkilöt pystyvät löytämään verkkosivustolta hakemansa tiedon. Kontrastin ideana verkkosivujen toteutuksessa on myös luoda esteettömyyttä toteutetussa verkkopalvelussa ja parantaa verkkosivustolla käytettävien kuvien ja tekstien suhdetta toisiinsa. Verkkosivuston väripaletti esitetään kuvassa 15.



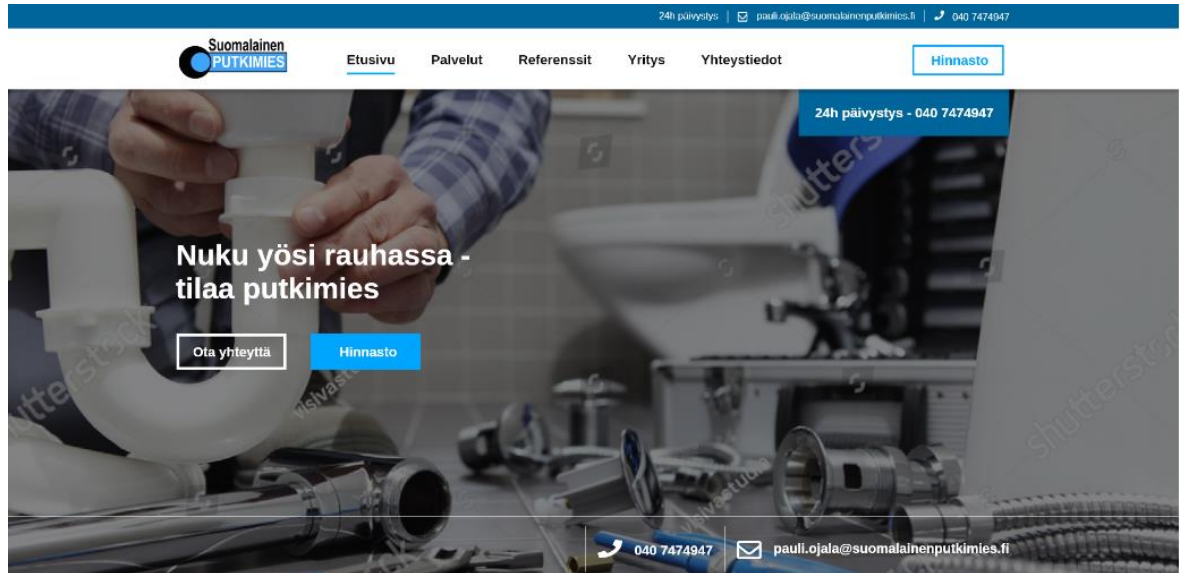
Kuva 14. Suomalainen Putkimies Oy:n logo



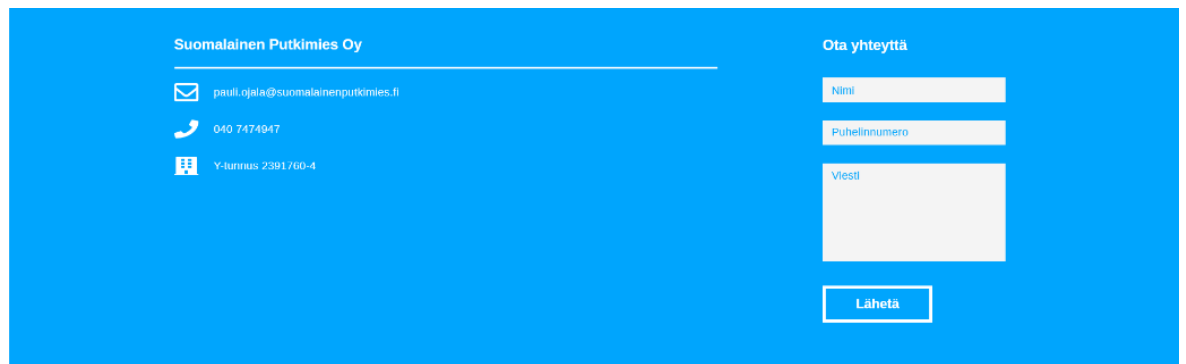
Kuva 15. Väripaletti

Graafisen suunnittelun tarkoituksena oli luoda näkemys verkkosivustosta, jonka mukaisena verkkosivusto toteutetaan. Verkkosivuston graafisen suunnittelussa toteutettiin ensimmäisenä sivuston ylä- ja alatunnisteet, valikko ja nostokuva, sekä nostokuvan päällä esi-

tettävät tekstipaikat ja painikkeet (kuva 16 ja 17). Näiden ympärille rakentuivat sivupohjien rakenteet toimeksiantajan tarpeiden mukaisena. Sivupohjia suunniteltiin kuusi erilaista, kappaleessa 4 esitetyn kuvan 10 mukaisesti. Liitteessä 1 esitetään sivuston etusivun graafisessa suunnitteluvaiheessa suunniteltu ulkoasu.



Kuva 16. Verkkosivuston ylätunniste, valikko ja nostokuva



Kuva 17. Sivuston alatunniste

4.4 Verkkosivuston sisältö

Sivuston sisällön kirjoittaminen ei kuulunut opinnäytetyöhön. Opinnäytetyön toteutuksessa vastattiin ainoastaan verkkosivuston rakenteellisesta toteutuksesta suunnitellun sivustorakenteen ja graafisen ulkoasun mukaisesti. Kaikki verkkosivustolla käytettävät tekstit tulivat toimeksiantajalta.

Toimeksiantajaa avustettiin sivuston sisällön lisäämisessä ja sopivien verkkosivustolla käytettävien kuvien etsinnässä. Verkkosivustolla käytetään toimeksiantajan itse kuvaamia ja kuvapankeista ostettuja kuvia. Toimeksiantajaa ohjeistettiin käyttämään ilmaisia kuva-

pankkeja kuten Pexels.com lähteenä sivustoille käytettäville kuville. Tämän lisäksi toimeksiantajalle kerrottiin maksullisista kuvapankeista, kuten Istockphoto.com.

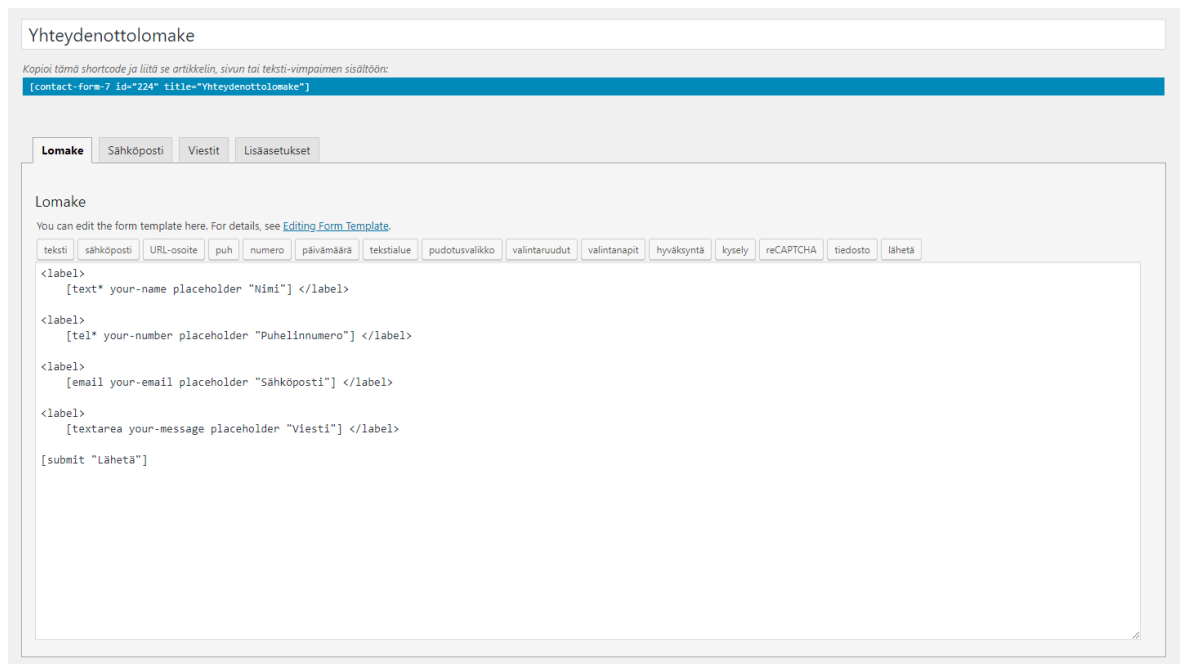
4.5 Käytetyt lisäosat

Sivuston toiminnallisuuksien toteuttamiseen ja käytön helpottamiseen käytettiin muutamaa erilaista lisäosaa. Tärkeimpänä näistä on Advanced Custom Fields (ACF), jota käytettiin sivuston toiminnallisuuksien rakentamiseen. Advanced Custom Fields on kehittäjille kehitetty lisäosa, jonka avulla WordPressissä voidaan lisätä ylimääräisiä sisältökenttiä. Näitä sisältökenttiä kutsutaan mukautetuiksi kentiksi, sillä kentillä määritetyt kuvat, tekstit jne. muokkaavat sivuston esitettyä ulkoasua. ACF poikkeaa muista WordPress-järjestelmässä käytettävistä sivuston rakentamiseen suunnitelluista lisäosista, sillä ACF-lisäosalla kehittäjä ohjelmoi ja määrittelee sivustolla käytettävät toiminnallisuudet sivupohjiin ja hallintapaneeliin.

Verkkosivuston toteutuksessa käytettiin ACF Pro -lisäosaa, joka on maksullinen versio Advanced Custom Fields -lisäosasta. Pro-versio sisältää laajennettuja toiminnallisuuksia. Näihin toiminnallisiin lukeutuvat muun muassa asetusten luominen WordPress-sisällönhallintajärjestelmän hallintanäkymään, sekä 'Repeater-Field' -kenttäryhmä, jonka avulla sivupohjissa voidaan toistaa sisältökenttiä.

ACF Pro -lisäosasta puuttui toiminnallisuus taulukenttien rakentamista varten. Taulukenttiä varten sivustolla käytettiin Advanced Custom Fields: Table Field -lisäosaa, joka laajentaa ACF-lisäosaa lisäämällä taulukkokentät.

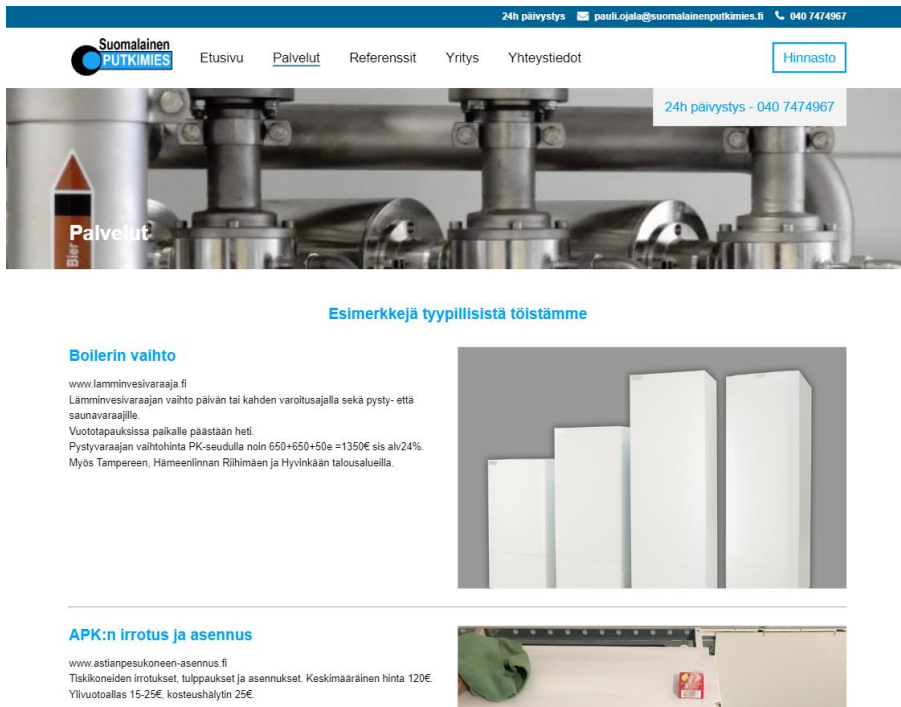
Sivustolle luotiin yhteydenottolomake (kuva 18) Contact Form 7 -lisäosalla. Contact Form 7 on yksinkertainen lisäosa, jolla voidaan luoda rajaton määrä lomakkeita käytettäväksi WordPress-sivustoille. Lisäosalla luodulle yhteydenottolomakkeelle määritettiin kentiksi nimi, puhelinnumero, sähköposti ja viesti. Lomake lisättiin verkkosivustolle käyttäen short-code-toimintoa, joka on lyhyt koodinpätkä, joka voidaan upottaa sivupohjaan. Kun potentiaalinen asiakas ottaa yhteyttä lomakkeen kautta, lähtee viesti sivuston ylläpitäjän sähköpostiin. Lomake ei tallenna yhteydenottoja sivuston tietokantaan.



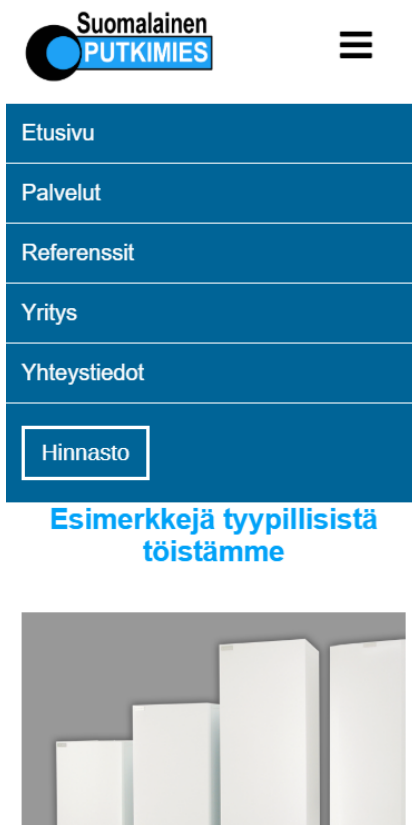
Kuva 18. Näkymä Contact Form 7 -lisäosan lomakehallinnasta

4.6 Staattiset sivupohjat

Verkkosivuston toteutus aloitettiin kehittämällä verkkosivuista staattiset sivupohjat graafisen suunnittelukuvien mukaisesti. Staattiset sivupohjat rakennettiin HTML, CSS ja JavaScript -tiedostoilla. Tarkoituksena oli luoda peruspohja, joka toimii kaikilla päätelaitteilla responsiivisesti ja ei sisällä dynaamisia toiminnallisuuksia. Kehittämällä sivupohjat ensin staattiseksi sivustoksi oli projekti helpompi vaiheistaa jatkokehityksen kannalta. Joitakin graafisessa suunnittelukuvissa suunniteltuja kohtia ei tehty staattisissa sivupohjissa tai niitä muutettiin. Nämä muutokset tehtiin siitä syystä, että verkkosivustoa pyrittiin parantamaan kokonaisvaltaisesti paremmaksi. Kuvissa 19 ja 20 esitetään sivuston staattisten sivupohjien ulkoasua pöytäkoneella ja älypuhelimella.



Kuva 19. Palvelut-sivun näkymä pöytäkoneella



Kuva 20. Sivuston valikko älypuhelimella

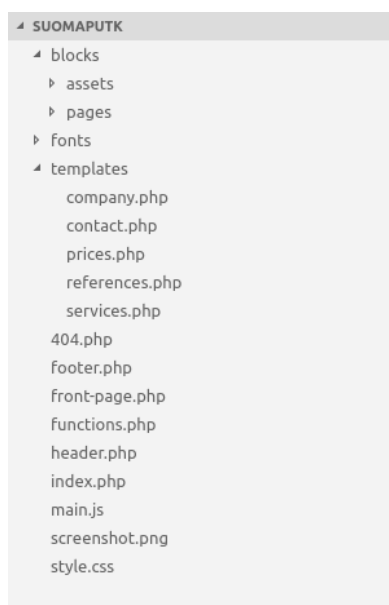
4.7 Konvertointi WordPress teemaksi

Staatisten sivupohjien kääntäminen WordPressillä toimivaksi teemaksi oli olennainen osa opinnäytetyön toteutusta. Teeman luominen aloitettiin luomalla WordPress-sisällönhallintajärjestelmässä uusi teemakansio, josta kaikki teeman tiedostot ladataan. WordPress tarvitsee kaksi tiedostoa, index.php:n ja style.css:n teemakansioon, jotta se tunnistaa kansion tiedostot validiksi teemaksi. Staatiset sivupohjat ladattiin teemakansioon, jonka lisäksi luotiin index.php-tiedosto.

Konvertointi teemaksi aloitettiin jakamalla HTML-tiedostot osioihin, joita pystytään käyttämään uudelleen eri sivupohjissa. Osiot myös muutettiin käyttämään PHP-tiedostopäätettä. Jokainen sivupohja käyttää samaa rakennetta, joka rakentuu seuraavanlaisesti:

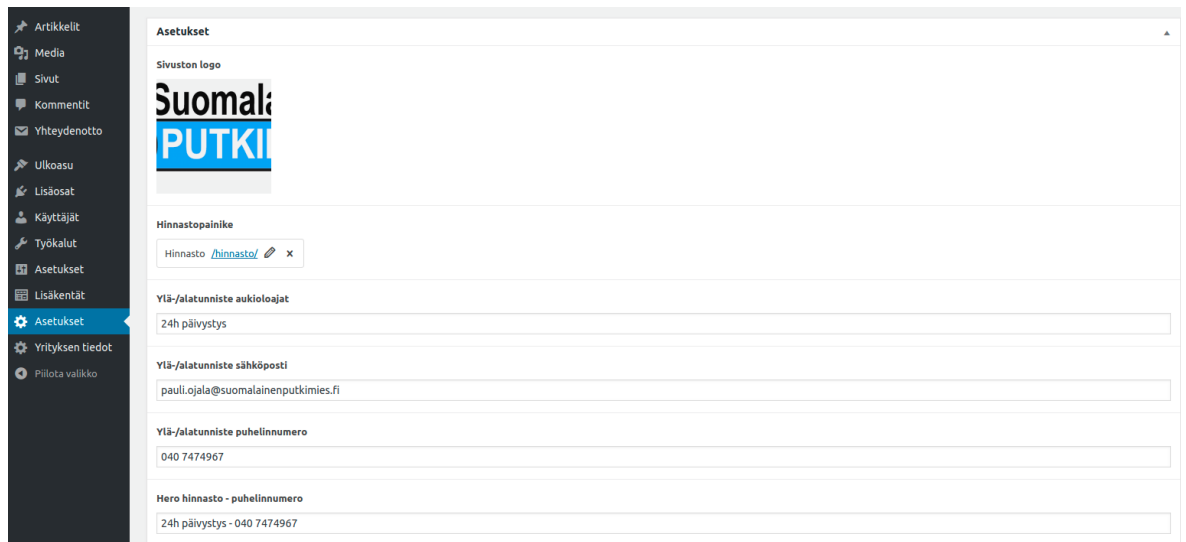
- header.php
- sivutyyppe.php
- sivupohjantiedostot.php
- footer.php

Header.php ja footer.php -tiedostot, jotka ladataan jokaisessa sivupohjassa, sisältävät sivujen toimivuuden kannalta olennaisia rakennepalikoita. Näihin rakennepalikoihin lukeutuvat tyylitiedostot, yleiset määritteet, valikot sekä ylä- ja alatunnisteissa esitettävät tiedot. Sivutyyppe.php ja sivupohjantiedostot.php ovat tiedostoja, joihin on määritetty olennaiset ominaisuudet tietyille sivupohjalle. Jokainen sivupohja generoidaan dynaamisesti neljästä edellä mainituista tiedostoista. Teeman tiedostorakenne esitetään kuvassa 21.



Kuva 21. Teeman tiedostorakenne

WordPress-sisällönhallintajärjestelmän asetuksissa määritettiin ACF-lisäosalla kenttätyyppejä, joita käytetään sivuston ja sivupohjien rakentamisessa. Yleiset asetukset (kuva 22) määritetään hallintapaneelissa. Nämä yleisissä asetuksissa määritetyt tiedot ovat yleispäteviä, jotka toistuvat eri sivupohjissa. Tämän lisäksi luotiin sivukohtaiset kenttätyypit, jotka ladataan sivupohjakohtaisesti.

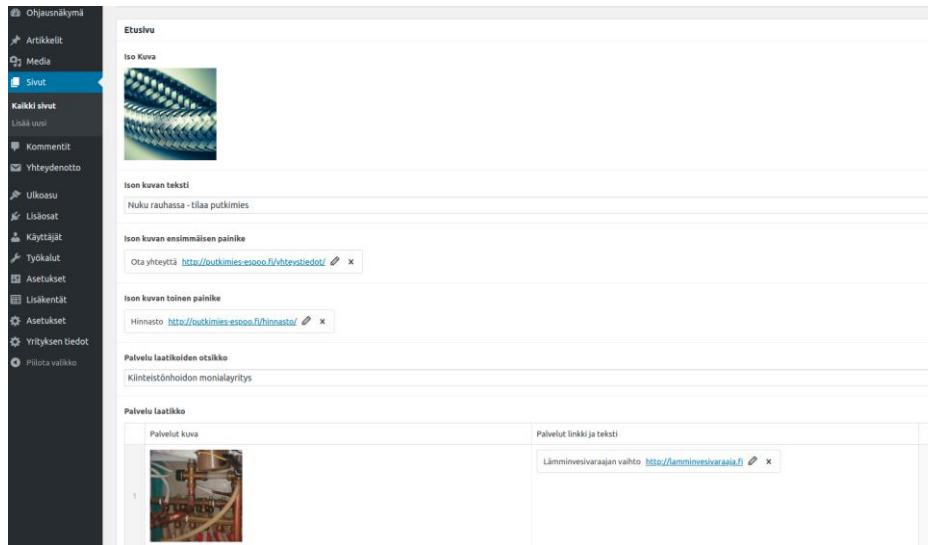


Kuva 22. Teeman yleiset asetukset

Kenttätyyppien määrytykset ladataan niitä koskevissa sivupohjissa. ACF:llä määritettyjen asetusten hakeminen tapahtuu erilaisilla PHP-ohjelmakoodilla, jotka lisätään ladattaviin tiedostoihin (kuva 23). Tämän jälkeen WordPress-sisällönhallintajärjestelmässä voidaan luoda uusi sivu, joka käyttää tiettyä sivupohjaa. Näille sivupohjille voidaan määrittellä ennalta määritettyjen kenttien kautta sivuston hallintapaneelissa (kuva 24) sivustolla esitettäviä tietoja, kuten tekstiä, kuvia ja linkkejä.

```
?>
<div class="box" style="background-image: url(<?php echo $image['url']; ?>);">
  <div class="box-layer">
    <?php if( $link ): ?>
      <h2>
        <?php echo $link['title']; ?>
      </h2>
      <a href="<?php echo $link['url']; ?>" target="<?php echo $link['target']; ?>"><span></span></a>
    <?php endif; ?>
  </div>
</div>
<?php endwhile; ?>
</div>
```

Kuva 23. Hallintapaneelissa määritetyn tiedon hakeminen sivupohjassa



Kuva 24. Etusivun muokkausnäkyminen hallintapaneelissa

Kaikkia toiminnallisuksia ei rakennettu ACF-lisäosalla, vaan jotkut toiminnallisuudet tehtiin WordPress-järjestelmässä käytettävillä toiminnallisuuksilla. Näihin lukeutuivat muun muassa valikossa näytettävät valikkolinkit, joiden hakemiseen käytettiin WordPress-järjestelmässä olevaa sisäänrakennettua ohjelmakoodia. Tämän lisäksi sivuilla esitettävät sivujen otsikot haettiin WordPress-järjestelmästä lyhyellä koodinpätkällä.

4.8 Testaaminen

Testaaminen on laadunvalvontaa, jolla varmistetaan, että järjestelmä toimii vaatimusmääritysten mukaisesti. Testauksen tarkoitus on varmistaa ohjelmiston laatu ja parantaa kehitettävää tuotetta. Yksinkertaistettuna testauksen tarkoitus on ohjelmassa tai tuotteessa esiintyvien virheiden löytäminen. Ohjelmistotestausta on hyvä toteuttaa koko projektin kehityskaaren ajan. Virheiden korjaaminen on sitä nopeampaa, mitä aikaisemmassa vaiheessa kehitystä ne havaitaan. Jos virheitä ei havaita, ne ehtivät vaikuttaa pidempään muiden osien toimintaan. Uudet ja muuttuneet toiminnallisuudet on hyvä testata aina. Lisäksi progressiivinen testaaminen on suositeltavaa laadun varmistamiseksi. (ATR Soft 2018.)

Testaamista toteutettiin sivuston kehityksen aikana iteratiivisesti eli pienissä osissa, joiden tulokset olivat nähtävissä ja kokeiltavissa jokaisen muutoksen jälkeen. Sivuston testaamisen tarkoituksena oli löytää mahdollisia virheitä aikaisessa vaiheessa ja pyrkiä korjaamaan virheet ennen seuraavaa vaihetta. Jokaisen isomman vaiheen toteutuksen jälkeen suoritettiin kattavampi testaus, jossa testattiin sivuston yleisiä toiminnallisuuksia ja rakenteellisia kohtia.

Staattisten sivupohjien kehityksen aikana testausta suoritettiin jatkuvasti, sillä jokaisen sivuston koodimuutos tarkistettiin heti kehityksen aikana selaimessa. Staattisten sivupohjien valmistumisen jälkeen sivusto testattiin yleisillä käytössä olevilla selaimilla ja laitteilla (kuva 25). Kaikkia laitteita ei pystytty testaamaan fyysisesti, sillä laitteita oli saatavilla vain rajoitettu määrä laitteita. Laitteet, joita ei ollut saatavilla, testattiin emulaattoreilla.

Käyttöjärjestelmä/Selain	Etusivu	Palvelut-sivu	Referenssit-sivu	Yrityssivu	Yhteystiedot-sivu	Hinnasto
Chrome	Testattu	Testattu	Testattu	Testattu	Testattu	Testattu
Firefox	Testattu	Testattu	Testattu	Testattu	Testattu	Testattu
Safari	Testattu	Testattu	Testattu	Testattu	Testattu	Testattu
Internet Explorer 11	Testattu	Testattu	Testattu	Testattu	Testattu	Testattu
Edge	Testattu	Testattu	Testattu	Testattu	Testattu	Testattu
Android	Testattu	Testattu	Testattu	Testattu	Testattu	Testattu
iPhone	Testattu	Testattu	Testattu	Testattu	Testattu	Testattu
iPad	Testattu	Testattu	Testattu	Testattu	Testattu	Testattu

Kuva 25. Sivuston testaaminen eri käyttöjärjestelmillä ja selaimilla

WordPress-teeman luonnissa testaaminen tapahtui myös iteratiivisesti. Teemassa käytetyt sivupohjat rakennettiin yksittäisinä tiedostoina, joilla jokaisella oli omat rakennepalikkansa. Näiden sivupohjien toiminnallisuuksia testattiin päivittämällä niihin testisisältöä ja tekemällä havaintoja mahdollisista ongelmista kehityksessä. Teeman ollessa valmis kokonaisuutena, lisättiin kaikille sivustoille vielä uudemman kerran testisisältöä. Tämän jälkeen sivustoa testattiin uudestaan. Sivuston testaamista suoritettiin myös toimeksiantajan puolelta, joka raportoi sisällönsyötön aikana ilmenneistä ongelmista. Kyseiset ongelmat korjattiin raportoinnin perusteella.

5 Pohdinta

Opinnäytetyön tavoitteena oli uudistaa visuaalisesti ja teknisesti toimeksiantajan verkkosivut. Verkkosivujen uudistamisen taustalla oli luoda uusia verkossa tapahtuvia kohtaamisia toimeksiantajan ja potentiaalisten asiakkaiden välillä.

Taustateoriassa perehdyttiin verkkosivustoihin ja niiden kehittämisessä käytettäviin teknologioihin. Tämän lisäksi työssä tutkittiin avointa lähdekoodia ja tunnettua avoimen lähdekoodin järjestelmää, WordPressiä, joka toimi keskeisenä osana työn toiminnallista osuutta.

Sain toimeksiantajalta vapaat kädet suunnitella ja toteuttaa verkkosivusto oman näkemykseni mukaisena. Tämä vapaus mahdollisti verkkosivuston kehittämisen ja suunnittelun omilla ehdoillani. Lopputuloksena syntynyt verkkosivusto on täysin oma näkemykseni hyvästä verkkosivustosta. Uskon verkkosivuston palvelevan hyvin toimeksiantajan asettamia tavoitteita yrityksen verkkoliiketoiminnalle.

Mielestäni työ onnistui hyvin. Taustateoriassa tutkitut tutkimuskysymykset ja teoria tukevat työn empiiristä osuutta. Työ kokonaisuutena toimii kattavana raporttina onnistuneesta opinnäytetyöstä. Opinnäytetyön tavoite saavutettiin, sillä verkkosivusto saatiin suunniteltua, toteutettua ja julkaistua opinnäytetyön puitteissa. Toimeksiantajan antama palaute verkkosivuista on ollut positiivista. Vaikka verkkosivuston kehittäminen on saatettu päätökseen opinnäytetyön osalta, on toimeksiantaja ilmaissut kiinnostuksensa verkkosivujen jatkokehitykselle.

Opinnäytetyön toteutuksen prosessi on ollut pitkä, monimutkainen ja haastava. Prosessin aikana olen kohdannut useita ylitsepääsemättömiltä tuntuvia haasteita. Näistä kuitenkin selvittiin ja työ saatiin vietyä päätökseen. Henkilökohtaisena tavoitteenani oli syventää ymmärrystäni verkkosivujen kehittämisestä sekä oppia lisää verkkosivujen kehittämiseen käytettävistä teknologioista. Työn aikana olen tutustunut näihin syvällisesti ja oppinut paljon uutta. Taitoni verkkosivustokehityksessä, graafisessa suunnittelussa ja tiedon hankinnassa ovat parantuneet huomattavasti. Mielestäni on hyvä, että valitsin opinnäytetyössä toiminnallisen toteutuksen, sillä toiminnallinen opinnäytetyö tukee ammatillista kehitystäni.

Lähteet

1&1 IONOS. What exactly is a website? Luettavissa:

<https://www.ionos.com/digitalguide/websites/website-creation/what-exactly-is-a-website/>.

Luettu: 22.11.2018.

Aranovitch, M. 2016. WordPress Plugins Explained. Luettavissa:

<https://wpcompendium.org/wordpress-basics/wordpress-plugins-explained/>. Luettu.

23.11.2018.

ATR Soft. 2018. Ohjelmistotestaus parantaa laatua. Luettavissa:

<https://www.atrsoft.com/ohjelmistotestaus-parantaa-laatua/>. Luettu 2.11.2018.

Balter, B. 2015. 6 motivations for consuming or publishing open source software. Luetta-

vissa: <https://opensource.com/life/15/12/why-open-source>. Luettu. 23.11.2018.

Bulkar, M. 2016. The Difference Between Static And Dynamic Websites? Luettavissa:

[https://www.linkedin.com/pulse/difference-between-static-dynamic-websites-mangesh-](https://www.linkedin.com/pulse/difference-between-static-dynamic-websites-mangesh-bulkar)

bulkar. Luettu: 1.1.2018.

BusinessDictionary. Website. Luettavissa:

<http://www.businessdictionary.com/definition/website.html>. Luettu: 1.11.2018.

Dodd, R. 2017. Why more companies are adpting open source technology. Luettavissa:

<https://about.gitlab.com/2017/03/03/why-choose-open-source/>. Luettu. 23.11.2018.

Duò, M. 2018. What is a Child Theme in WordPress? A Plain English Explanation of Their

Importance in Custom Development. Luettavissa: [https://codeable.io/what-is-a-child-](https://codeable.io/what-is-a-child-theme-wordpress/)

theme-wordpress/. Luettu 3.11.2018.

Fitzgerald, B., Keesan, J., Russo, B., Shaikh, M. & Succi, G. 2011. Adopting Open Source

Software : A Practical Guideline. The MIT Press Ser. MIT Press. Cambridge (MA). Luetta-

vissa: <https://ebookcentral.proquest.com/lib/haaga/reader.action?docID=3339329&query=>. Luet-

tu: 25.10.2018.

Freeservers. What Is a Website? Luettavissa:

<http://www.freeservers.com/WebHosting101/WhatIsAWebsite.html>. Luettu: 22.11.2018.

Guru99. What is PHP? Write your first PHP Program. Luettavissa:
<https://www.guru99.com/what-is-php-first-php-program.html>. Luettu 6.11.2018.

iThemes. What is WordPress? Luettavissa: <https://ithemes.com/tutorials/what-is-wordpress/>. Luettu: 1.11.2018.

Karol, K. 2018. WordPress Stats: Your Ultimate List of WordPress Statistics (Data, Studies, Facts – Even the Little-Known). Luettavissa:
<https://www.codeinwp.com/blog/wordpress-statistics/>. Luettu: 2.11.2018

Kinsta. 2018. The History of WordPress, its Ecosystem and Community. Luettavissa:
<https://kinsta.com/learn/wordpress-history/>. Luettu 25.10.2018.

Laukkarinen, R. 2017. WordPress-sivut – räätäilille vai markettiin? Luettavissa:
<https://www.dude.fi/wordpress-sivut-raatalille-vai-markettiin>. Luettu: 22.10.2018.

Leiniö, T. 2015. WordPress-kehitys: räätälöity vai valmisteema? Luettavissa:
<https://www.sofokus.com/fi/blogi/wordpress-kehitys-raataloity-vai-valmisteema/>. Luettu 8.11.2018.

Morris, S. 2018. Everything You Need To Know About JavaScript. Luettavissa:
<https://skillcrush.com/2012/04/05/javascript/>. Luettu 6.11.2018.

Muriuki, F. 2016. History Of WordPress: A What You Must Know Guide. Luettavissa:
<https://www.wpkube.com/wordpress-history/>. Luettu 25.10.2018.

NTC Hosting. PHP in HTML. Luettavissa:
<https://www.ntchosting.com/encyclopedia/scripting-and-programming/php/php-in/>. Luettu: 24.11.2018.

PHP. What Is PHP? Luettavissa: <http://php.net/manual/en/intro-what-is.php>. Luettu 6.11.2018.

Sadie. 2011. Web Design Basics – How are websites being used? Luettavissa:
<https://www.falkondigital.com/blog/web-design-basics-how-are-websites-being-used/>. Luettu: 25.10.2018.

Sitecore. What is a CMS (Content Management System)? Luettavissa:

<https://www.sitecore.com/topics/what-is-a-cms>. Luettu: 22.11.2018.

Shannon, R. What is HTML? Luettavissa:

<https://www.yourhtmlsource.com/starthere/whatishtml.html>. Luettu: 3.11.2018

Singh, V. A 2018. A Brief History Of Open Source. Luettavissa:

<https://medium.com/gitcoin/a-brief-history-of-open-source-3928cb451767>. Luettu 1.11.2018.

Techopedia. JavaScript (JS). Luettavissa:

<https://www.techopedia.com/definition/3929/javascript-js>. Luettu 6.11.2018.

W3Schools. JQuery Introduction. Luettavissa:

https://www.w3schools.com/jquery/jquery_intro.asp. Luettu: 6.11.2018.

Weber, S. 2004. The Success of Open Source. Harvard University Press. Cambridge (MA). Luettavissa:

<https://ebookcentral.proquest.com/lib/haaga/reader.action?docID=3300611&query=>. Luettu 25.10.2018.

West, M. 2012. HTML5 Foundations. John Wiley & Sons, Inc. New York. Luettavissa:

<https://ebookcentral.proquest.com/lib/haaga/reader.action?docID=1120310&query=>. Luettu: 25.10.2018

WordPress. Using Themes. Luettavissa: https://codex.wordpress.org/Using_Themes. Luettu: 25.10.2018.

Wikipedia. HTML5. Luettavissa: <https://fi.wikipedia.org/wiki/HTML5>. Luettu 6.11.2018.

Wikikirjasto. CSS3: Luettavissa: <https://fi.wikibooks.org/wiki/CSS3>. Luettu 6.11.2018.

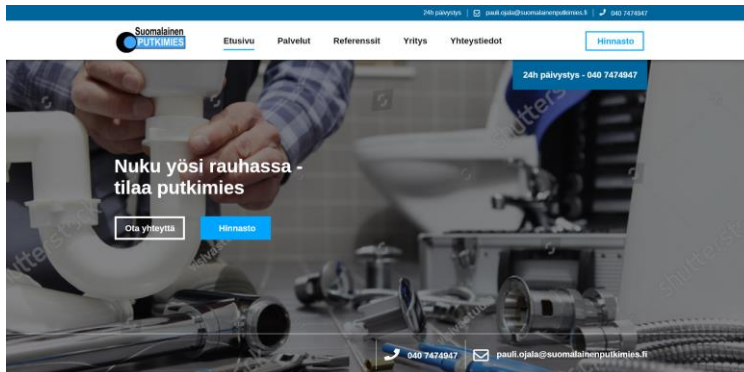
Wilson, B. 2013. A Brief History Of WordPress. Luettavissa: <https://winningwp.com/a-brief-history-of-wordpress/>. Luettu: 1.11.2018.

WordPress.org. GNU Public License. Luettavissa:

<https://fuc.wordpress.org/about/license/>. Luettu 25.10.2018

Liitteet

Liite 1. Verkkosivuston etusivun graafinen ulkoasu



Kiinteistöhoiton monialaryitys

