



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Paasila Timo-Jaakko

KOTISIVUJEN SUUNNITTELU JA TOTEUTUS WORDPRESSILLÄ

Tekniikka ja liikenne
2013

TIIVISTELMÄ

Tekijä	Timo-Jaakko Paasila
Opinnäytetyön nimi	Kotisivujen suunnittelu ja toteutus Wordpressillä
Vuosi	2013
Kieli	suomi
Sivumäärä	53 + 3 liitettä
Ohjaaja	Pirjo Prosi

Opinnäytetyö tehtiin Vaasassa pääkonttoriaan pitävälle Contact & Travel Finland Oy:lle, joka on keskittynyt telemarkkinointiin sekä isännöinti- ja tilipalveluihin. Opinnäytetyön aihe oli tehdä kolmet web-kotisivut CTF-konsernin yrityksille. Työn teettäminen opiskelijalla osoittautui edulliseksi tavaksi hankkia uudet sivut.

Web-sivut toteutettiin Wordpress-sisällönhallintaohjelmalla. Wordpressin käyttö vaati palveluntarjoajalta tietokannan ja palvelintilan, jotka toimeksiantaja hankki ennen työn aloittamista. Wordpressin asentamisen jälkeen web-sivujen runko suunniteltiin mahdollisimman tarkasti, jolloin toteutusvaihe voitiin aloittaa ilman ongelmia. Työn tekemiseen annettiin melko vapaat kädet, joten sivujen rakentamiseen voitiin käyttää omaa luovuutta niin teknisen kuin visuaalisenkin toteuttamisen suhteen.

Valmiit web-sivut täyttivät kaikki toimeksiantajan vaatimukset. Alkuperäisiin suunnitelmiin lisättiin vielä joitain elementtejä, esimerkiksi diaesitykset tehostamaan sivujen visuaalista ulkonäköä. Sivujen yleisilme onnistuttiin saamaan miellyttävän näköiseksi ja samaan aikaan modernien kotisivujen vaatimusten mukaiseksi. Sisältörakenne saatiin selkeäksi ja helposti lähestyttäväksi. Wordpress asetti tietynlaiset rajoitukset toteuttamisen suhteen, mutta siitä huolimatta lopputulos oli onnistunut.

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Tietotekniikan koulutusohjelma

ABSTRACT

Author	Timo-Jaakko Paasila
Title	Designing and Implementing Websites with Wordpress
Year	2013
Language	Finnish
Pages	53 + 3 Appendices
Name of Supervisor	Pirjo Prosi

This thesis was done for a company named Contact & Travel Finland Oy (CTF). Its head office is located in Vaasa. CTF is focused on telemarketing, account and property manager services. The goal of this thesis was to design and implement three web-homepages for the companies of the CTF concern. Hiring a student to implement these homepages was an economical way for CTF.

Web-pages were implemented using Wordpress which is a popular content management system. Database and server space was needed before installing Wordpress. After installation the homepages were designed carefully which made implementation much easier. CTF gave full freedom to design and implement these homepages, so it was allow to use own creativity and imagination with technical and visual issues.

Completed web-sites filled all requirements. Some elements were added to original plans, for example a slider to boost the visual look. The basic appearance was comfortable and filled modern homepage requirements. The content structure was clear and easy to approach. Wordpress set some limits so it was not possible to implement all plans. Despite that the homepages were successful.

Keywords	Wordpress, content management, CSS, homepages, design and implementation
----------	--

KÄYTETYT LYHENTEET

MySQL	MySQL-tietokanta on hyvin suosittu web-palveluiden tietokantana.
CSS	Cascading Style Sheets, dokumentille voi määrittellä useita tyyliohjeita, jotka yhdistetään tietyllä tavalla yhdeksi säännöstöksi.
HTML	Hypertext Markup Language, avoimesti standardoitu kuvauskieli, jolla voidaan kuvata hyperlinkkejä sisältävää tekstiä.
PHP	Hypertext Preprocessor, ohjelmointikieli, jota käytetään erityisesti Web-palvelinympäristöissä dynaamisten web-sivujen luonnissa.
URL	Uniform Resource Locator, käytetään osoittamaan WWW-sivuja.
ZIP	Tiedostonpakkausmenetelmä.
FTP	File Transfer Protocol, TCP-protokollaa käyttävä tiedostonsiirtomenetelmä kahden tietokoneen välille.
JPEG	Joint Photographic Experts Group, häviöllistä pakkausta käyttävä bittikarttagrafiikan tallennusformaatti.
ICO	Microsoft Windows -käyttöjärjestelmän käyttämä tiedostomuoto ohjelmakuvakkeissa.
GIF	Graphic Interchange Format, häviötön bittikarttagrafiikan tallennusformaatti. Pystyy näyttämään vain 256 väriä kerralla.
PNG	Portable Network Graphics, häviötön bittikarttagrafiikan tallennusformaatti. Kehitettiin GIF-formaatin korvaajaksi.
IP	IP-osoite, Internetin Protokolla-osoite. Numerosarja, joka yksilöi jokaisen Internet-verkkoon kytketyn tietokoneen.
ID	Identifier, on tietojenkäsittelyssä annettava yksilöllinen tunniste. Esimerkiksi henkilötunnus on numeromuotoinen tunniste.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO.....	7
2	CONTACT & TRAVEL FINLAND OY	8
3	KEHITYSTYÖKALUT	9
	3.1 Sisällönhallintaohjelma.....	9
	3.2 Wordpress	9
	3.3 HTML	11
	3.4 CSS	12
	3.5 PHP	14
4	WORDPRESSIN ASENTAMINEN	16
	4.1 Tietokannan luonti	16
	4.2 MySQL-asetusten määrittäminen	18
	4.3 Tiedostojen siirto palvelimelle.....	19
	4.4 Wordpressin asennus selaimella	20
5	KOTISIVUJEN SUUNNITTELU JA ESIVALMISTELU.....	21
	5.1 Teeman valinta.....	21
	5.1.1 Responsive-teema	22
	5.1.2 Child theme eli lapsiteema	22
	5.2 Sivuston ulkoasu	24
	5.2.1 Asettelu	24
	5.2.2 Valikkorivi	25
	5.2.3 Etusivu	26
	5.2.4 Favicon.....	27
	5.3 Wordpress-lisäosat.....	27
	5.3.1 Jetpack.....	28
	5.3.2 Contact Form 7.....	28
	5.3.3 Better WP Security.....	29
	5.3.4 Meta Slider	29
	5.3.5 Admin menu editor	29
	5.4 Selainten yhteensopivuus.....	29

6	KOTISIVUJEN TOTEUTUS.....	31
6.1	Sivuston rungon muutokset.....	31
6.2	Sivuston rakenne ja asettelu.....	33
6.2.1	Header	33
6.2.2	Footer	34
6.2.3	Widgetit.....	35
6.2.4	Lomakkeet.....	36
6.2.5	Sliderit.....	38
6.3	Ulkoasun muokkaus.....	39
6.3.1	Taustakuva	39
6.3.2	Valikon muokkaus	40
6.3.3	Napin ja tekstikentän muokkaus	42
6.3.4	Wrapper.....	44
6.4	Tietoturva.....	46
7	TESTAUS.....	48
8	JOHTOPÄÄTÖKSET JA POHDINTA	50
	LÄHDELUETTELO.....	53

1 JOHDANTO

Kotisivut ovat nykyään melkein jokaisella hiemankin suuremmalla yrityksellä, koska niistä haetaan yhä enemmän tietoa internetin kautta. Omat kotisivut ovat yrityksen näyteikkuna ja mahdollisuus laajentaa asiakaskuntaa. Huolellisesti suunnitellut ja toteutetut sivut voivat nousta merkittävään rooliin uusien asiakkaiden hankinnassa, tiedottamisessa ja myös imagon nostattamisessa.

Tämän opinnäytetyön toimeksiantaja on Vaasassa pääkonttoriaan pitävä Contact & Travel Finland Oy. Yritys toimii 13:lla eri paikkakunnalla ja sen toimialat ovat myynti ja markkinointi. Konserni tarvitsee kolmet uudet kotisivut yrityksilleen ja niiden teettäminen opinnäytetyön muodossa oli edullisuutensa vuoksi hyvä vaihtoehto.

Alun perin oli tarkoitus hankkia jokaiselle sivulle erikseen palvelintilat ja domainit, mutta koska se olisi tullut liian kalliiksi, päätettiin kaikki kolme sivustoa sijoittaa samalle palvelimelle yhden domainin alle. Tällä tavalla kuluja pystyttiin karsimaan huomattavasti, eikä ratkaisu haitannut työn suorittamista millään tavalla.

Itse työssä keskitytään pääasiassa minulle jo entuudestaan tuttuun Wordpress-sisällönhallintaohjelman toimintaan ja sillä toteutettavien sivujen kehitykseen. Ennen Wordpressin käyttöönottoa se asennetaan palvelimelle, jonka jälkeen aloitetaan sivujen suunnittelu. Suunnittelu- ja toteutusvaiheeseen syvennyttään hyvin tarkasti, jotta lopputuloksesta tulee onnistunut. Työhön annettiin melko vapaat kädet toteuttamisen osalta, joten luovuutta ja osaamista pystyttiin käyttämään sivujen ulkonäköä sekä rakennetta koskien.

Toteutusvaiheen jälkeen web-sivujen toiminta testataan huolellisesti virheiden varalta. Lopuksi pohditaan lopputulosta ja työn onnistumista.

2 CONTACT & TRAVEL FINLAND OY

Contact & Travel Finland Oy perustettiin vuonna 2007 ja se toimii tällä hetkellä 13:lla eri paikkakunnalla. Työntekijöitä on noin 150 eri puolilla Suomea. Vaasan toimipiste sijaitsee lähellä kaupungin ydinkeskustaa. /1/

Yrityksellä on vuosien kokemus tuloksetta ja kustannustehokkaasta levikki-markkinoinnista, ja on siksi yksi Suomen suurimmista alan ammattilaisista. Palveluja tarjotaan niin yksityisille kuin yrityksillekin. /1/

Contact & Travel Finland Oy panostaa erityisesti työntekijöiden kouluttamiseen, jotta myyntijärjestelmä olisi mahdollisimman laadukas. Työntekijöiden rekrytoinnissa kiinnitetään erityisesti huomiota hakijan kykyyn suoriutua hyvin puhelimen välityksellä tehdystä myyntityöstä ja markkinoinnista. /1/

3 KEHITYSTYÖKALUT

Tässä luvussa käydään läpi työssä käytetty sisällönhallintaohjelma, CSS-muotoilukieli ja tarvittavat ohjelmointikielet.

3.1 Sisällönhallintaohjelma

Sisällönhallintaohjelma on ohjelma, jolla voidaan luoda, muokata, tallentaa ja julkaista sisältöä. Sillä on mahdollista hallita suuriakin web-sivustoja, eikä sen käyttö vaadi ohjelmointitaitoja, joten sitä oppii käyttämään myös ilman kokemusta. Sisällönhallintaohjelma on web-sovellus, joka pitää asentaa palvelimelle, jotta sitä voidaan käyttää. Tietokanta tarvitaan, jotta sisältöä voidaan tallentaa. /10/

Sisällön muokkaaminen on yksinkertaistettuna seuraavanlaista: käyttäjä kirjautuu verkkoselaimellaan ohjelman hallintapaneeliin, kirjoittaa tekstiä jollekin sivulle ja julkaisee sen, jolloin luotu sisältö on internetissä kaikkien saatavilla. Ulkonäkömuutoksia voidaan tehdä yksinkertaisesti hallintapaneelin ulkoasuasetuksia muuttamalla tai muokkaamalla tyylitiedostoa. /10/

Sisällönhallintaohjelmaa voidaan käyttää isojen tai pienien web-sivujen ylläpitoon, mutta myös blogien pitäminen on todella suosittua. Erilaisilla käyttöoikeuksilla voidaan joukolle sivuston ylläpitäjiä antaa erilaisia oikeuksia sisällön tai hallintapaneelin muutoksiin. /10/

Wordpress valittiin tämän työn sisällönhallintaohjelmaksi, koska se on työn tekijälle jo ennestään tuttu. Wordpress on myös suosituin sisällönhallintaohjelma, joten tukiverkko on varsin kattava.

3.2 Wordpress

Wordpress on ilmainen, yli 70 kielelle käännetty, sisällönhallintaohjelma, jota on kehitetty jo vuodesta 2003 lähtien. Se oli alun perin tarkoitettu blogien luomista ja ylläpitoa varten. Wordpress perustuu avoimeen lähdekoodiin ja se on kirjoitettu PHP-kielellä. Koodia saa siis vapaasti muokata omien tarpeidensa mukaan. Wordpress käyttää MySQL-tietokantaa tietojen tallentamiseen, joten sille on olta-

va tietokanta ennen kuin se voidaan asentaa palvelimelle. Wordpress kattaa tällä hetkellä jopa 17 % internetistä. /2/

Wordpressin pääkehittäjiä ovat Matt Mullenweg ja Ryan Boren, jotka haarauttivat sen omaksi projektikseen sen edeltäjän b2/cafelogin lopettamispäätöksen jälkeen. Eri versioiden julkistamisen kautta Wordpressiin tuli jatkuvasti uusia ominaisuuksia ja sitä alettiin pitää sisällönhallintaohjelmana. Suurimmat syyt olivat staattisten sivujen tulo sekä kehittynyt pohjatiedostojärjestelmä. /2/

Wordpress on tunnettu helposta ja nopeasta asennuksestaan. Myös sen ja lisäosien päivittäminen on todella yksinkertaista, joten käytön aloittaminen ei vaadi kovin paljoa tietämystä. Palvelimella sijaitseviin tiedostoihin pääsee käsiksi käyttämällä jotain FTP-tiedonsiirtoa tukevaa ohjelmaa. /2/

Yksinkertaisuutensa takia Wordpressillä on helppoa ja nopeaa julkaista mitä tahansa sisältöä netissä. Joustavuuden ansiosta sillä voidaan luoda kotisivuja, blogeja ja monia muita web-sivuja. Tuhansilla lisäosilla ja teemoilla Wordpressistä voidaan tehdä aivan itsensä näköinen. Nämä ovat syitä, miksi se on sisällönhallintaohjelmana niin suosittu. /2/

Käyttäjiä on mahdollista luoda useita ja niille voi antaa eri oikeudet, joten monen eri tason ylläpitäjät voivat päivittää sivustoa omien oikeuksiensa rajoissa. Median-siirto on tehty yksinkertaiseksi vedä ja pudota –tekniikalla. /2/

Viimeisin suuri päivitys oli versio 3.0, joka mahdollisti muun muassa usean blogin ylläpidon ilman erillisiä asennuksia. Tällä hetkellä uusin versio on 3.5.1. ja sen käytön vaatimukset palvelimelle ovat, että PHP:n versio on vähintään 5.2.4 ja MySQL:n vähintään 5.0. Vanhemmilla Wordpress-versioille kelpaavat hieman vanhemmat PHP- ja MySQL-versiot. Vanhempien Wordpress-versioiden tukeminen on kuitenkin lopetettu aina tietyin väliajoin. Tietoturvan kannalta on aina syytä päivittää uusimpaan versioon. /2/

3.3 HTML

Lyhenne HTML tulee sanoista HyperText Markup Language. Se on avoimesti standardoitu yksinkertainen kuvauskieli, jota voidaan kirjoittaa millä tahansa tekstieditorilla. HTML keskittyy ulkoasun kuvaamisen sijaan web-sivujen rakentamiseen eli sisällön asetteluun sivuilla. Kaikki web-sivustot käyttävät HTML:ää josain määrin. /3/

HTML:llä on kaksi tärkeää ominaisuutta ja ne ovat hyperteksti sekä universaalisuus. Hyperteksti tarkoittaa kykyä luoda toiselle web-sivustolle johtava linkki. Universaalisuus mahdollistaa minkä tahansa tietokoneen HTML-dokumenttien lukemisen, oli sitten kyseessä Mac-, Unix- tai PC-tietokone. Tästä huolimatta eri yhdistelmät suhtautuvat web-sivuihin hieman eri tavalla ja tuloksena saattaa olla erilainen näkymä. Tähän vaikuttaa esimerkiksi tietokoneen malli ja käytettävä selain. /3/

HTML käyttää kulmasulkeita, joiden sisällä on jokin koodin tunniste eli tagi. Esimerkiksi sivun otsikon aloitustagi `<title>` ja lopetustagi `</title>` (**Kuva 1**). Tagien väliin sijoitetaan haluttu otsikko, joka sitten näytetään web-sivuilla. /3/

```
<html>
  <head>
    <title>Tähän tulee otsikko</title>
  </head>
  <body>
  ...
  </body>
</html>
```

Kuva 1. HTML-rakenne.

HTML:stä on kehitetty paranneltu versio, XHTML. Se on voimakkaampi, joustavampi ja sitä on mahdollista laajentaa omien tarpeiden mukaan. CSS-kuvauskielen kannalta XHTML on paljon parempi ratkaisu. Tästä huolimatta se ei ole vielä syrjäyttänyt HTML:ää, mutta mahdollisesti tulevaisuudessa XHTML tulee yleistymään. /3/

Tässä työssä HTML:ää tarvitaan ainoastaan lomakkeissa ja joidenkin sivujen taustakkoasettelussa eli siihen ei tulla sen tarkemmin keskittymään.

3.4 CSS

CSS on lyhenne sanoista Cascading Style Sheets. HTML:n keskittyessä rakentamiseen, hoitaa CSS kuvaamisen eli CSS-tiedosto sisältää tyylisäännöt esimerkiksi web-sivuille. CSS on hyvin monipuolinen ja sillä voidaan tehdä huomattavia muutoksia sivuston ulkonäköön. Tyyli-tiedostoon voidaan lisätä niin paljon haluttuja tyyllisääntöjä kuin vain nähdään tarpeelliseksi. /4/

CSS ei ole ohjelmointikieli, joten sillä ei voida esimerkiksi luoda ehdollisia ohjeita, kuten jos jokin ehto täyttyy, suoritetaan tietty muotoilu. CSS:stä ei olla tulevaisuudessakaan tekemässä ohjelmointia tukevaa, koska sille ei nähdä tarvetta. CSS:ssä ei voi myöskään käyttää ohjelmoinnista tuttuja vakioita helpottamaan muutoksien tekemistä. /4/

Web-sivujen yhteydessä CSS:n tarkoitus on vaikuttaa vain tiettyihin ulkoasun ominaisuuksiin ja loput jätetään pääosin selaimen ja käyttäjän asetusten varaan. Pelkällä HTML:llä voidaan myös tehdä monia ulkonäkömuutoksia, mutta CSS on huomattavasti monipuolisempi ja joustavampi, koska se on nimenomaan suunniteltu ulkoasun määrittelyä varten. CSS:n ominaisuudet voidaan yrittää tiivistää yhteen lauseeseen, mutta tämä ei anna koko kuvaa muotoilukielen mahdollisuuksista. Tyyllisäännöllä voidaan siis vaikuttaa web-sivujen asemointiin, tekstin muotoiluun, värien ja kuvien esitykseen, sekä moniin erilaisiin muotoiluasetuksiin. /4/

Yhteen tyyllisääntöön voidaan sisällyttää useita määrittelyjä. Sääntö koostuu valitsimesta, yhdestä tai useammasta ominaisuudesta ja tämän arvosta. Kuvassa 2 muokataan päävalikon ulkonäköä muuttamalla sen taustaväri liukuväriksi valkoisesta siniseen ja alle luodaan varjo. Valitsin on tässä tapauksessa *.menu* ja ominaisuutena esimerkiksi *background-color*, jonka arvona on *transparent*. Selaintyteensopivuuden varmistamiseksi jotkin säännöt lisätään monta kertaa eri etuliitteellä. Esimerkiksi *-moz-* -etuliite varmistaa Firefoxin kanssa toimivuuden ja *-o-* puolestaan Operan. Valikon kulmat pyöristetään *border-radius*-ominaisuudella ja

leveys eli *width* määritetään 600 pikselin mittaiseksi. *Margin* määrittää valikon ympärillä olevan tyhjän tilan, mutta tässä tapauksessa sitä ei ole, koska arvo on 0.

```
.menu {
    background-color: transparent;
    background-image: -webkit-gradient(linear, left top, left bottom, from(rgba(255,0,0,0)), to(rgba(0,0,255,0)));
    background-image: -webkit-linear-gradient(top, rgba(255,0,0,0), rgba(0,0,255,0));
    background-image: -moz-linear-gradient(top, rgba(255,0,0,0), rgba(0,0,255,0));
    background-image: -ms-linear-gradient(top, rgba(255,0,0,0), rgba(0,0,255,0));
    background-image: -o-linear-gradient(top, rgba(255,0,0,0), rgba(0,0,255,0));
    background-image: linear-gradient(top, rgba(255,0,0,0), rgba(0,0,255,0));
    width: 600px;
    margin: 0;
    border-radius: 10px;
}
```

Kuva 2. Valikkorivin muokkaamista CSS-tiedostossa.

Selain voi käyttää vain yhtä tyylytiedostoa kerrallaan, mutta *@import*-säännöllä on mahdollista ottaa mukaan myös toisia tyylytiedostoja, jotka täydentävät jo olemassa olevaa säännöstöä. /4/

Yksi CSS:n haasteista on eri selainten erilainen suhtautuminen tyyllisääntöihin. Web-sivut saattavat näyttää erilaisilta riippuen käytettävästä selaimesta, joten kehitysvaiheessa kannattaa käyttää useaa eri selainta, jotta sivut näyttäisivät mahdollisimman samanlaisilta. Vielä 2000-luvun alussa tilanne yhteensopivuuden suhteen oli erittäin huono, varsinkin Internet Explorer 6:n ja sen edeltäjien kanssa. Nykyään tilanne on huomattavasti parempi ja monet suosituimmista selaimista näyttävät sivustot melkein identtisinä, kunhan CSS-tiedosto on oikein rakennettu. Selainten yhteensopivuuteen keskitytään vielä syvemmin viidennessä luvussa. /4/

Tyylytiedosto kannattaa tarkistuttaa eli validoida virheiden varalta. Sen voi suorittaa kätevästi osoitteessa <http://jigsaw.w3.org/css-validator/>. Tarkistin ilmoittaa mahdollisista virheistä, jolloin ne voidaan paikantaa ja korjata. Joskus selain saattaa noudattaa virheellistä tyyllisääntöä, vaikka sen ei pitäisi. Myöhemmässä vaiheessa selaimen päivittyessä sitä ei enää ehkä noudatetakaan, jolloin seuraa ongelmia. Sen vuoksi virheet tulee käydä läpi, mutta kaikkien korjaaminen ei ole täysin välttämätöntä, koska kyseessä on vain suositukset. /4/

Tässä työssä CSS on suurimmassa roolissa, sillä suurin osa sivun ulkonäkömuutoksista hoidetaan sen avulla.

3.5 PHP

PHP-nimistä ohjelmointikieltä käytetään erityisesti dynaamisten webdokumenttien luomisessa. Lyhenne PHP viittaa yleisesti itse ohjelmointikieleen ja myös teknisiin ratkaisuihin, jotka mahdollistavat PHP-kielisten ohjelmien suorittamisen. Tällöin voidaan puhua PHP-ympäristöstä. PHP on saatavilla täysin ilmaiseksi. /5/

PHP voidaan upottaa HTML-koodin sisälle (**Kuva 3.**) käyttämällä PHP:n omaa merkintää eli `<?php` ja `?>` tageja. Koodi tulkitaan web-palvelimella ja näitä tulkittavia ohjelmointikieliä kutsutaan skriptikieliksi. Lyhyesti sanottuna PHP-ohjelmointiin tarvitaan tavallinen tekstieditori, web-palvelin, PHP-tulkki ja web-selain. /5/

```
<html>
  <head>
    <title>PHP-esimerkki</title>
  </head>
  <body>
    <?php echo "Hei!"; ?>
  </body>
</html>
```

Kuva 3. PHP-koodia sijoitettuna HTML-koodin sisälle.

Ilmaisuuuden lisäksi PHP:n etuna on, että se on suunniteltu erityisesti web-sovellusten kehittämiseen. Ohjelmointikielenä PHP on helposti opittava, joten sillä saa nopeasti aikaan tuloksia. C-kielestä lainattu syntaksi on tehokas ja monipuolinen, eikä kovinkaan monimutkainen. Suosion kasvamisen myötä PHP:tä on alettu käyttää yhä isommissa järjestelmissä. Suosio on myös lisännyt sopivien web-hotellien tarjontaa sekä mahdollistaa suuren määrän dokumentteja ja avuliaita ihmisiä verkossa, joiden avulla löytää helposti ratkaisun ongelmaan. /5/

Tässä työssä PHP-kieltä ei käytetä kovin paljon, vaikka Wordpress on kirjoitettu kokonaan sillä. Vaatimuksena tälle työlle on PHP-kielen perusteiden ymmärtäminen, jotta lisättävän koodin sijoittaminen Wordpressin PHP-tiedostoihin onnistuu ongelmitta.

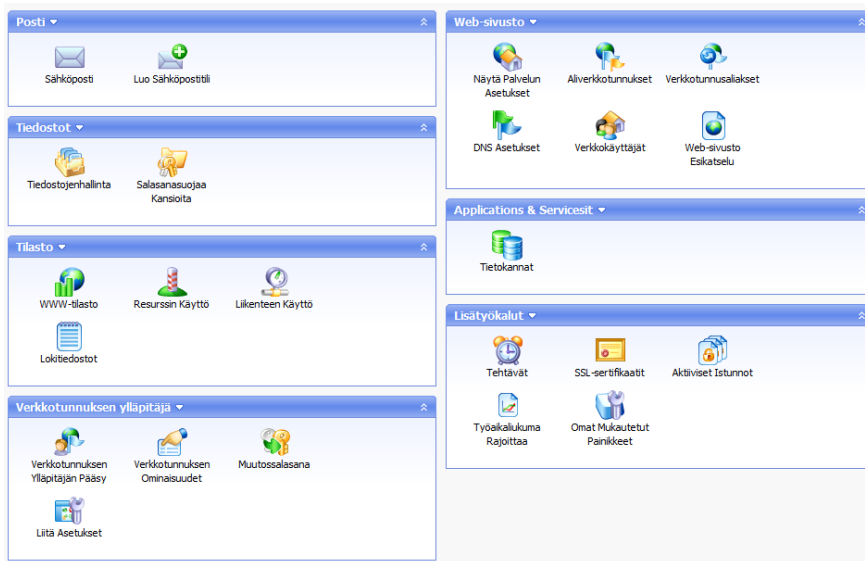
4 WORDPRESSIN ASENTAMINEN

Tässä luvussa selostetaan yksityiskohtaisesti kaikki neljä Wordpressin asentamisen vaihetta. Niitä ovat tietokannan luonti, MySQL-asetusten määrittäminen, tiedostojen siirto palvelimelle ja itse asennus selaimella.

4.1 Tietokannan luonti

Ennen Wordpressin asennusta luodaan tietokanta palvelimen Parallels Plesk Panel –hallintapaneelilla (**Kuva 4.**). Sinne kirjaututaan palveluntarjoajan antamilla tunnuksilla, minkä jälkeen avataan lista tietokannoista ja valitaan kohta ”lisää uusi tietokanta”. Sille annetaan haluttu nimi eli tässä tapauksessa wp-database (**Kuva 5.**). Tietokanta on nyt luotu eli sille voidaan lisätä tietokantakäyttäjää. Tekstikenttiin annetaan käyttäjänimi ja salasana (**Kuva 6.**). Luodulle käyttäjälle annetaan automaattisesti kaikki käyttöoikeudet tietokantaan, joten niitä ei tarvitse erikseen määrittellä.

MySQL-tietokannan hallintatyökalu on phpMyAdmin, mutta sitä ei tarvitse käytännössä käyttää ollenkaan, ellei tietokantataulujen luonti epäonnistu asennuksen aikana. Tällöin ne voidaan kätevästi poistaa hallintatyökalulla. Kaikki on nyt tietokannan osalta valmista Wordpressin asennusta varten.



Kuva 4. Parallels Plesk Panel –hallintapaneeli.



Parallels® Panel

Home Setup Sitebuilder Account Help & Support

Koti > Tietokannat >

Lisää uusi tietokanta

Lisää uusi tietokanta

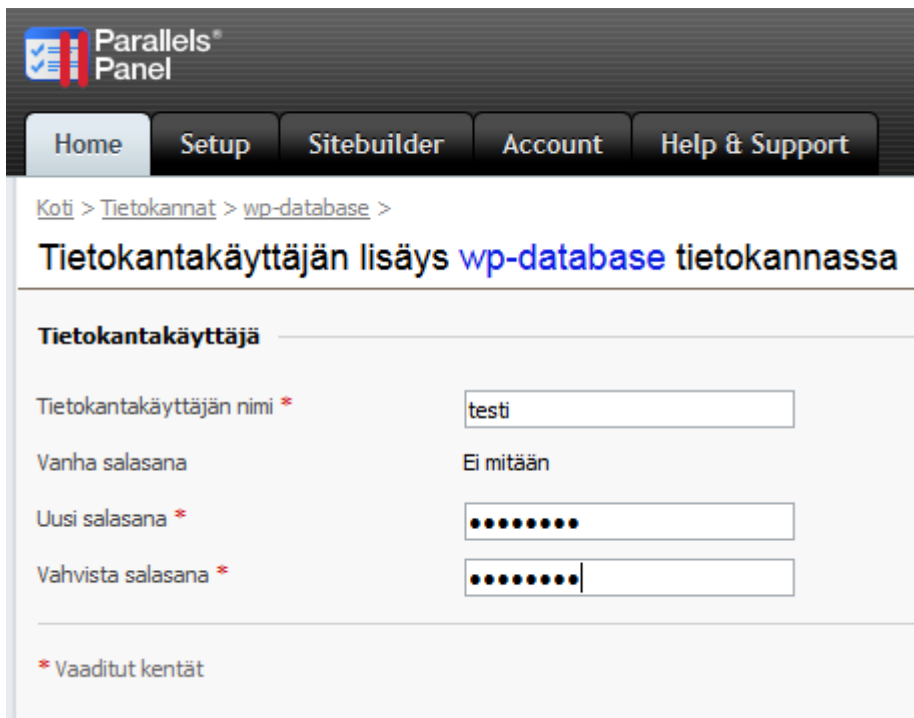
Tietokannan nimi *

Tyyppi

Tietokantapalvelin Paikallinen MySQL palvelin (MySQL-oletusarvo)

* Vaaditut kentät

Kuva 5. Tietokannan luonti.



Parallels® Panel

Home Setup Sitebuilder Account Help & Support

Koti > Tietokannat > wp-database >

Tietokantakäyttäjän lisäys wp-database tietokannassa

Tietokantakäyttäjä

Tietokantakäyttäjän nimi *

Vanha salasana Ei mitään

Uusi salasana *

Vahvista salasana *

* Vaaditut kentät

Kuva 6. Käyttäjän lisääminen.

4.2 MySQL-asetusten määrittäminen

Ennen asentamista Wordpress ladataan ZIP-pakettina Wordpressin kotisivuilta. Paketti puretaan kotikoneelle, jonka jälkeen avataan sen sisältä löytyvä wp-config-sample.php-tiedosto. Sinne tehdään muutamia välttämättömiä muutoksia, jotta Wordpressin asentaminen olisi ylipäättään mahdollista.

Tiedostoon lisätään luodun tietokannan nimi kohtaan `define('DB_NAME', 'wp-database');`, jotta asennusohjelma tietää, mitä tietokantaa asennuksessa käytetään. Tietokannankäyttäjän tunnus ja salasana tulee kirjoittaa kohtiin `define('DB_USER', 'testi');` ja `define('DB_PASSWORD', 'testipassu');` täysin oikein. Autentikointi-kohtiin (**Kuva 7.**) sijoitetaan esimerkiksi Wordpressin omalla tarkoitukseen suunnitellulla generaattorilla luodut merkkijonot.

Jotta sama tietokanta voidaan jakaa useille eri Wordpress-asennuksille, on kunkin asennuksen taulujen edessä oltava oma etutunniste kohdassa `$table_prefix = 'esim_';`. Kun tässä työssä samaa tietokantaa käyttää kolme eri sivustoa, ovat niiden etutunnisteet esimerkiksi wp-myynti_, wp-rekry_ ja wp-tili_. Tietoturvan kannalta ei suositella käytettävän pelkkää oletuksena olevaa wp_-etutunnistetta sen haavoittuvuuden vuoksi.

Muokattu tiedosto nimetään uudelleen muotoon wp-config.php.

```
define('AUTH_KEY', 'oma uniikki lauseesi');
define('SECURE_AUTH_KEY', 'oma uniikki lauseesi');
define('LOGGED_IN_KEY', 'oma uniikki lauseesi');
define('NONCE_KEY', 'oma uniikki lauseesi');
define('AUTH_SALT', 'oma uniikki lauseesi');
define('SECURE_AUTH_SALT', 'oma uniikki lauseesi');
define('LOGGED_IN_SALT', 'oma uniikki lauseesi');
define('NONCE_SALT', 'oma uniikki lauseesi');
```

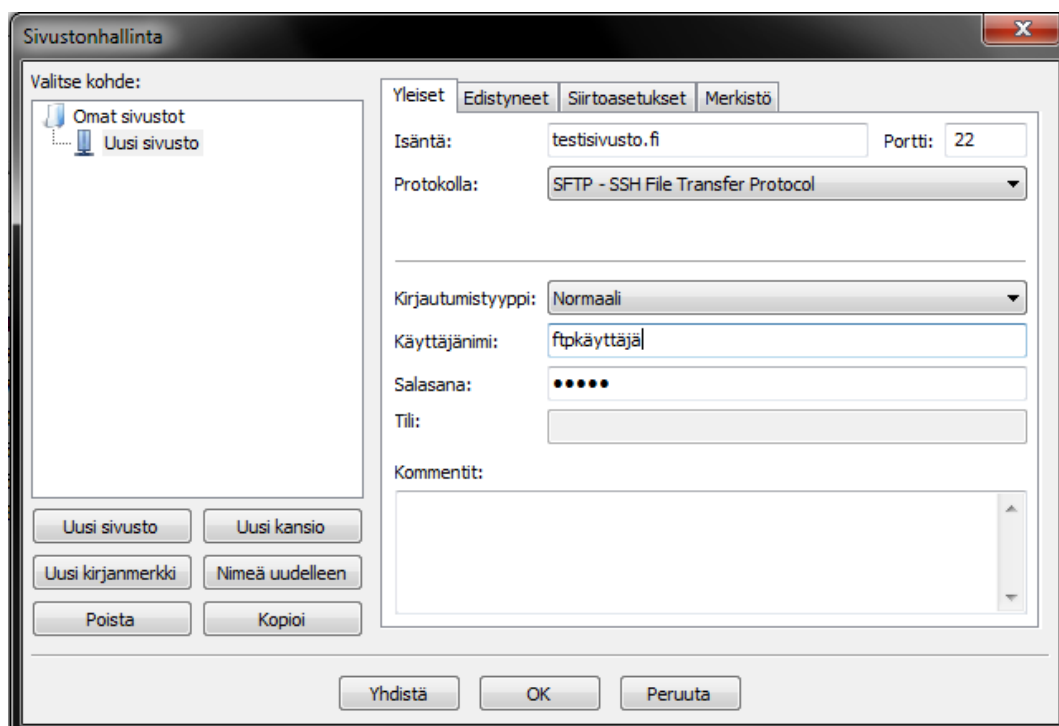
Kuva 7. Autentikoinnin uniikit avaimet.

4.3 Tiedostojen siirto palvelimelle

Seuraavaksi Wordpress-kansion sisältö siirretään palvelimelle FileZilla-ohjelman avulla. Palveluntarjoajalta saaduilla FTP-tunnuksilla kirjaudutaan palvelimelle (**Kuva 8.**), jonka jälkeen kaikille kolmelle asennukselle luodaan omat kansiot palvelimen hakemistoon ja suoritetaan tiedostojen siirto kotikoneelta. Ainut pieni ero kansioiden sisällössä on edellä mainittu wp-config.php-tiedosto.

Kansioiden nimet valitaan sen mukaan, millainen tuleva web-sivusto tulee olemaan. Koska sivuista tulee isäntäsivujen www.esim.fi ”alisivustoja”, on niiden kaikkien tuleva URL-osoite muotoa www.esim.fi/kansionnimi. Tästä syystä esimerkiksi rekrytointiin keskittyvien sivujen kansion nimeksi annetaan rekrytointi, jolloin URL tulee olemaan www.esim.fi/rekrytointi. Tällä tavalla vältetään myöhemmässä vaiheessa turhalla vaivalta, kun osoiterakennetta jouduttaisiin muuttamaan Wordpressistä käsin huomattavasti monimutkaisemmalla tavalla.

Wordpress on nyt valmis asennettavaksi.



Kuva 8. Palvelemille kirjautuminen FileZillalla.

4.4 Wordpressin asennus selaimella

Selaimen osoitekenttään kirjoitetaan `http://www.esim.fi/kansionnimi/wp-admin/install.php`, jolla ajetaan asennus-skripti. Selaimen avautuu sivu, jonne syötetään halutun sivuston otsikko, käyttäjänimi ja salasana (**Kuva 9**). Käyttäjänimeksi ei tule koskaan asettaa pelkkää admin-tunnusta, koska se on haavoittuvainen hakkereiden hyökkäyksille. Sähköpostikenttään kirjoitetaan asentajan oma sähköpostiosoite. Koska sivustolle halutaan luonnollisesti näkyvyyttä, jätetään ruksi kohtaan ”Salli sivuston näkyminen hakukoneissa”.

Wordpress on asennettu onnistuneesti eli nyt voidaan kirjautua Wordpressin sisälönhallintaohjelmaan osoitteessa `http://www.esim.fi/kansionnimi/wp-admin` käyttämällä käyttäjätunnusta ja salasanaa.

Tervetuloa

Tervetuloa kuuluisaan viiden minuutin asennukseen! Muutaman vaiheen jälkeen pääset käyttämään omaa henkilökohtaista julkaisualustaasi. Ehtiessäsi voit [ReadMe-tiedoston](#).

Tarvittavat tiedot

Anna aluksi seuraavat tiedot. Älä huolehdi, voit kyllä muuttaa niitä myöhemmin.

Sivuston otsikko	<input type="text" value="CTF-Rekryointi"/>
Käyttäjänimi	<input type="text" value="testi-admin"/> <small>Käyttäjänimet voivat sisältää vain kirjaimia, numeroita, välilyöntejä, alaviivoja, väliviivoja, pisteitä ja @-symbolin.</small>
Salasana, kahdesti	<input type="password"/> <input type="password"/> <input type="button" value="Salasanan vahvuus"/> <small>Vihje: Salasanan kannattaa olla vähintään seitsemän merkkiä pitkä. Jos haluat tehdä salasanasta mahdollisimman turvallisen, käytä isoja ja pieniä kirjaimia, numeroita ja symboleja kuten ! " ? \$ % ^ &).</small>
Sähköpostiosoite:	<input type="text" value="testi@testi.fi"/> <small>Varmista sähköpostiosoitteesi ennen kuin siirryt seuraavaan vaiheeseen.</small>
Tietosuoja	<input checked="" type="checkbox"/> Salli sivuston näkyminen hakukoneissa.
<input type="button" value="Asenna WordPress"/>	

Kuva 9. Wordpressin asentaminen selaimella.

5 KOTISIVUJEN SUUNNITTELU JA ESIVALMISTELU

Tässä luvussa perehdytään siihen, mistä tulevat web-sivut rakentuvat, ja mitä kaikkea onnistuneeseen lopputulokseen tarvitaan.

Suunnitteluvaiheessa on otettava monta asiaa huomioon, vaikka uusia ideoita tulee vielä itse kehitysvaiheessakin. Toteuttaminen on huomattavasti helpompi aloittaa, kun tärkeimmät asiat ovat valmiiksi mietitty. /9/

5.1 Teeman valinta

Wordpressiin on saatavilla useita eri teemoja, joiden ominaisuudet vaihtelevat. Ilmaisia vaihtoehtoja on ladattavissa satoja. Maksullisissa teemoissa on hieman enemmän ominaisuuksia ja niiden ulkonäkö on oletuksena näyttävämpi kuin ilmaisissa.

Teeman valinta aloitettiin tutustumalla niiden suosioon Wordpressin kotisivuilla. Voitiin olettaa, että suosituimmat teemat ovat toiminnaltaan vakaimpia, yhteensopivia useimpien lisäosien kanssa ja niitä myös päivitetään säännöllisin väliajoin. Teemojen perusulkonäkö on melkein sama teemasta riippumatta, joten myös sen takia kannattaa pysyä tutuissa ja turvallisissa valinnoissa, eikä lähteä kokeilemaan satunnaisia hyvännäköiseksi havaittuja teemoja, jotka voisivat myöhemmässä toteutusvaiheessa osoittautua huonoiksi valinnoiksi.

Kun teeman vakaus oli varmistettu, selvitettiin sen ominaisuudet. Koska tämän työn kotisivut ovat melko yksinkertaiset, ei teemalta vaadittu kovin paljoa, kunhan perusominaisuudet löytyivät.

Teema asennettiin muutamalla klikkauksella Wordpressin teemanhallinnassa. Asennus ei vaatinut käyttäjältä mitään toimenpiteitä, joten teemaa voitiin käyttää välittömästi.

5.1.1 Responsive-teema

Tässä työssä päädyttiin käyttämään Responsive-teemaa, joka on eniten ladattu ilmainen teema. Se on todella monipuolinen ja joustava, ja se on käännetty 40:lle eri kielelle. Siinä on useita ominaisuuksia, jotka täyttävät tämän työn web-sivujen vaatimukset. Sivujen asettelumalleja löytyy 9 erilaista, joiden avulla jokaiselle välilehdelle voidaan valita sopiva. Näille sivuille voidaan lisätä niin sanottuja vimpaimia eli widgettejä, riippuen asettelumallista. Widgetit ovat pienoisohjelmia, jotka näyttävät esimerkiksi hakukentän tai kalenterin. Widgeteille on varattu 11 eri paikkaa riippuen asettelumallista, mutta niitä ei ole kuitenkaan pakko käyttää. /7/

Sivuille voidaan asettaa taustakuva ja header-logo yksinkertaisesti hallintapaneelin avulla. Headerin alle voidaan asettaa oletuksena oleva päävalikko, mutta sen lisäksi voidaan lisätä myös ylä-, ala- ja päävalikon alivalikko. Valittavana on siis 4 erilaista valikkoa, joita voi kaikkia muokata haluamallaan tavalla. Välilehtien järjestystä pystyy vaihtamaan ja tarpeen vaatiessa niitä voidaan muuttaa vetovalikoiksi. Valikoiden ulkonäköä muutetaan tyylitiedostoa muokkaamalla, joka mahdollistaa muun muassa kulmanpyöristyksen, värien vaihdon ja lasiefektin.

Kaikki yksityiskohtaisemmat ulkonäkömuutokset on tehtävä tyylitiedostoon koodia muuttamalla. Esimerkiksi valitun asettelumallin sivupohjien alle voidaan lisätä varjostus tai koko pohjan väriä voidaan muuttaa. Muokausmahdollisuuksia on lukuisia, joten vaihtoehtoja oikean tyylin löytämiseksi on paljon.

Responsivella on todella monipuolinen tuki ympäri internetiä johtuen teeman suosiosta. Pieneenkin ongelmaan löytää helposti vastauksen sen omalta tukifoorumilta.

5.1.2 Child theme eli lapsiteema

Lapsiteeman käyttöä suositellaan, jos web-sivun CSS-tyylitiedostoon tai PHP-tiedostoihin tehdään muutoksia. Kun isäntäteemaan asennetaan päivitykset, korvataan vanhat tiedostot uusilla eli kaikki tehdyt muutokset pyyhitään ja sivuston ul-

koasu palautetaan oletustilaan. Lapsiteeman ollessa käytössä, säilyvät muutokset päivittämisestä huolimatta. /8/

Lapsiteeman luominen onnistuu luomalla uusi kansio esimerkiksi FileZilla-ohjelmalla palvelimella olevan Wordpress-asennuskansion themes-kansioon. Nimeksi annetaan esimerkiksi responsive-child. Luotuu lapsiteeman kansioon vie-dään tyhjä styles.css-tiedosto, joka löytyy siis myös isäntäteemasta. Tähän tyh-jään tiedostoon tehdään muutoksia, jotta Wordpress tunnistaa lapsiteeman. Tie-dostosta tulee käydä ilmi, että se on yhteydessä isäntäteemaan. /8/

CSS-tiedoston alkuun kirjoitetaan header, joka nähdään kuvassa 10.

```
/*
Theme Name: Responsive-child
Version: 0.1.0
Author: TJ
Template: responsive
*/
@import url("../responsive/style.css");
```

Kuva 10. Responsive-lapsiteeman CSS-tiedoston header.

Pakollisia tietoja ovat *theme name* ja *template*. Theme namen on hyvä olla samaa muotoa kuin aiemmin mainittu lapsiteeman kansio eli esimerkiksi isäntäteema-child. Template ilmaisee isäntäteeman kansion nimen themes-kansiossa. Nimen pitää olla täsmälleen oikein kirjoitettu isoja ja pieniä kirjaimia myöten. Headerin jälkeen tuleva *@import*-lause on myös välttämätön, jotta isäntäteeman CSS-tiedosto osataan tuoda lapsiteemaan. Tiedostopolulla osoitetaan, missä isäntäteeman styles.css-tiedosto sijaitsee. /8/

Tämän jälkeen lapsiteema on valmis käytettäväksi. Teema otetaan käyttöön Wordpressin ulkoasuasetuksissa. Nyt isäntäteema ei periaatteessa ole enää käytös-sä, vaan se toimii lapsiteeman kautta.

Jos PHP-sivupohjatiedostoihin halutaan tehdä muutoksia, tehdään niistäkin uudet versiot lapsiteeman kansioon teemapäivitysten varalta. Tällaisia tiedostoja ovat esimerkiksi footer.php ja header.php. Näihin tiedostoihin ei kuitenkaan tarvitse tehdä samanlaista headeria, kuin styles.css-tiedostoon, vaan ne voivat olla suoria kopioita isäntäteeman vastaavista tiedostoista. /8/

5.2 Sivuston ulkoasu

Ulkoasu on yksi tärkeimmistä asioista web-sivuilla. Ketään ei kiinnosta vierailta tylsännäköisillä ja huonosti rakennetuilla sivuilla, vaan sieltä poistutaan todella nopeasti. Yrityksen kotisivut ovat osa sen imagoa, joten huonoilla sivuilla annetaan helposti negatiivinen kuva, vaikka todellisuudessa niin ei olisikaan. /9/

Ulkoasu rakentuu monesta eri asiasta. Värivalinnat, taustakuvat, kuvat, diaesitykset, grafiikka, valikkorivi ja itse sivun asettelu vaikuttavat kaikki lopputulokseen. Näiden kesken tulee löytää sopiva tasapaino, jotta mikään ei nouse liian räikeästi muiden yläpuolelle.

Onnistuneen lopputuloksen saavuttamiseksi kannattaa kokeilla erilaisia yhdistelmiä värimaailman ja elementtien muotojen osalta. Tämä saattaa viedä paljon aikaa, mutta se ei ole tämän työn osalta ongelma. Suurimmat muutokset ulkonäköön tehdään CSS-tyylitiedostoa muokkaamalla rivi kerrallaan.

Toimeksiantaja antoi esimerkkisivustot, jotka antavat suuntaa itse toteutukseen. Pääväriksi yritys haluaa heidän pääsivuillaankin näkyvän taivaansinisen.

Sisältösivujen välillä on huolehdittava, että valittu värimaailma ja rakenne pysyy lähes samana sivusta riippumatta, jolloin sivusto näyttää ulospäin mahdollisimman yhtenäiseltä. Pienet muutokset ovat kuitenkin suositeltavia. /6/

5.2.1 Asettelu

Asettelussa tulisi pitää graafinen ja tekstipohjainen sisältö tasapainossa. Erityisesti ensivaikutelman kannalta edellä mainitut asiat ovat tärkeitä. Liiallinen tekstin tai vastaavasti kuvien määrä joissakin kohdissa saattaa johtaa vierailijan poistumiseen sivuilta. Tekstin ja kuvien tasapainoisella yhteiskäytöllä asettelu saadaan

miellyttävän näköiseksi ja samalla haluttu viesti saadaan tehokkaasti perille. Kuvien tulee tukea tekstiä mahdollisimman hyvin. /6/

Wordpressillä toteutettujen sivujen rakenne on yleensä aika samanlainen. Ylä-laidassa on otsikkotila, johon asetetaan esimerkiksi logo. Sen alapuolella on vaakasuuntainen valikkorivi. Jäljelle jäänyt tila on käytännössä varattu itse sisällölle. Tämä asettelumalli sopii mainiosti tämän työn kotisivujen pohjaksi. Sisällön asettelu on hoidettava huolellisesti selkeyden säilyttämiseksi ja jokaisen sivun tulee olla helposti luettavissa.

Web-sivujen vierittäminen pystysuunnassa on normaalia, mutta sivuttaissuuntaista vierittämistä tulisi välttää, jos vain mahdollista. Käytettävyyden kannalta se on todella huono ratkaisu ja saattaa pahimmassa tapauksessa ajaa vierailijan pois. Sivuttaisvierittäminen on kuitenkin helppo estää rajoittamalla sisältötilan leveys tarpeeksi pieneksi, jolloin pienelläkin näytöllä ei tule ongelmia. /6/

Rajoittamalla sisällön leveydeksi 960 pikseliä, pysyvät kuvat ja teksti aina määritetyn alueen sisällä. Kapeampi tila myös tiivistää sisällön, jolloin navigointi sivuilla on helpompaa.

5.2.2 Valikkorivi

Valikko on tärkeä sivujen navigoinnin kannalta. Sen avulla siirrytään sivustolla välilehtiin ja etsitään haluttua tietoa. Valikon nimien on kuvattava välilehteä mahdollisimman tarkasti, jolloin vierailijan on helppo siirtyä haluamaansa välilehteen sen enempää ajattelematta. /9/

Valikosta pitää aina löytyä linkki myös etusivulle, koska vierailija voi eksyä jollekin muulle välilehtisivulle, esimerkiksi hakukoneen kautta. Toinen tärkeä linkki on yhteystiedot, jotka jokainen uskottava yritys tarjoaa vierailijalleen. /9/

Selkeys on tässäkin asiassa tärkeä, joten valikkoa ei kannata tunkea täyteen erilaisia linkkejä. Kaikki päälinkit suositellaan mahdolluttamaan yhdelle riville ja niiden alle voi tarpeen vaatiessa lisätä alivalikoita. Selkeyttä lisää myös se, että aktiivi-

nen sivun valikkonimi näytetään muusta valikosta eriävällä värillä, jolloin vierailija tietää, millä sivulla on sillä hetkellä.

Valikko ei saa viedä liikaa tilaa, mutta se ei saa olla liian pienikään. Pientä linkkiä on hankalaa klikata ja sillä pilataan helposti sivun käyttökokemus. Erityisesti iäkkäillä ihmisillä voi olla hankaluuksia tähdätä hiirellään turhan pieneksi säädettyä valikkorivin linkkiä, puhumattakaan kosketusnäyttöjen tai mobiililaitteiden käyttäjistä. Sivuttaissuuntainen tila kannattaa hyödyntää tehokkaasti. Jos valikon loppuosaan jää paljon tyhjää, venytetään valikossa olevia linkkejä koko leveyden matkalle, koska silläkin parannetaan käyttökokemusta. Turhan tiheään ahdetut linkit johtavat helposti väärän välilehtisivun aukeamiseen.

Linkkien järjestys suunnitellaan mahdollisimman loogiseksi. Ensimmäiseksi tulee aina etusivu, jonka jälkeen loput järjestetään sivustosta riippuen.

5.2.3 Etusivu

Etusivu on tärkein osa sivustoa. Sillä on kaksi tärkeää tehtävää:

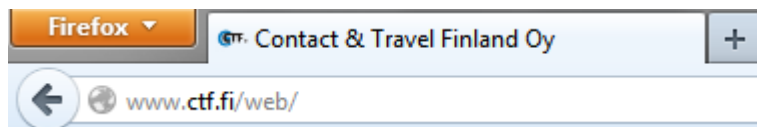
1. Varmistaa, ettei vierailija poistu välittömästi sivuilta
2. Saada vierailija klikkaamaan jotain sivun sisältöä.

Etusivu on kuin tiivistelmä koko sivustosta eli sen pitää antaa kerralla oikea vaikutus ja näyttää, millainen yritys on kyseessä. Ulkoasu suunnitellaan sopivan houkuttelevaksi ja valikot selkeiksi. Sekava asettelu ajaa vierailijan nopeasti pois, jolloin suunnittelussa on epäonnistuttu. /9/

Etusivulle sijoitettavalla liikkuvalla kuvalla, kuten diaesityksellä, luodaan dynaamisuutta ja annetaan kehittyneempi kuva sivustosta. On kuitenkin vältettävä memestä liiallisuuksiin ja pyrittävä pitämään lisättävä sisältö maltillisena. Liikkuvaa kuvaa voidaan käyttää tehokkaasti hyväksi etusivun houkuttelevuuden lisäämiseksi, kun siihen sisällytetään osa etusivun normaalista sisällöstä.

5.2.4 Favicon

Favicon on sivuille määritelty ikoni, joka näytetään esimerkiksi selaimen osoiterivillä, välilehden reunassa (**Kuva 11.**) ja kirjanmerkeissä. Sen koko voi olla 16×16, 32×32 tai 64×64 pikseliä ja tiedostomuotoina JPEG, ICO, PNG tai GIF.



Kuva 11. Favicon selaimen välilehdessä.

Favicon on pieni, mutta todellisuudessa hyvinkin merkittävä yksityiskohta sivustolle. Sillä erotetaan sivusto muista, esimerkiksi kun vierailijalla on selaimessa useita välilehtiä auki samaan aikaan. Onnistuneella faviconin valinnalla sivusto saadaan näyttämään uskottavammalta.

Favicon voidaan luoda itse kuvankäsittelyohjelmalla, mutta helpompaa on käyttää jotain netin favicon-generaattoria. Sen avulla omille sivuille sopivan persoonallisen ikonin luonti onnistuu nopeasti.

Onnistuneen faviconin työstäminen sivuille voi olla yllättävän haastavaa, koska 16x16 pikselin kokoiselle kuvalle melkein jokaisella yksittäisellä pikselillä on merkitystä.

5.3 Wordpress-lisäosat

Wordpressiin on saatavilla tuhansia lisäosia, joita voidaan ladata ja asentaa mielenä mukaan. Osa on teemojen ohella maksullisia ja huomattavasti laajempia verrattuna maksuttomiin. Lisäosilla tuodaan aivan uusia ominaisuuksia sivuille ja niillä voidaan tehostaa koko sivuston toimintaa. Osa on vierailijoita varten ja ovat näkyvä osa ulkoasua, mutta toiset eivät näy ulkopuolisille. Esimerkiksi sivuston kävijälaskuri tai hallintapaneelin valikkorivin muokkaamiseen tarkoitettu lisäosa ovat vain ylläpitäjää varten, eivätkä ne vaikuta sivuston ulkoiseen toimintaan millään tavalla.

Lisäosienkin osalta on hyvä tarkistaa niiden suosio ja arvostelut, sillä kuka tahansa voi luoda lisäosia. Sen takia niiden käyttö voi olla riskialtista ilman tarkkaa tietoa ohjelman tarkoituksesta. Lisäosia kannattaa pääsääntöisesti ladata vain Wordpressin omasta lisäosahakemistosta, koska siellä julkaistavat lisäosat on tarkastettu haattakoodien varalta. Tämä ei täysin takaa turvallisuutta, mutta todennäköisyys vaaralliselle lisäosalle on marginaalinen. Ulkopuolisilta sivuilta ladattavan sisällön kanssa tulee olla erittäin varovainen.

Tämän työn Wordpress-asennuksille tarvitaan muutama lisäosa. Niiden avulla parannetaan tietoturva, saadaan tietoa kävijämääristä, ohjataan kävijäpalaute suoraan haluttuun sähköpostiin, lisätään diaesitys sivustolle ja muokataan Wordpressin hallintapaneelin valikkoa. Kaikki asennetut lisäosat ovat suosituimpien joukossa, joten pelkoa haattakoodista ei ole.

5.3.1 Jetpack

JetPack on Wordpressin kehittämä lisäosa, jolla on useita ominaisuuksia. Suurin osa niistä on maksullisia, mutta tässä työssä tarvitaan ainoastaan sen tarjoamaa kävijälaskuria. Se saadaan käyttöön, jos omistaa wordpress.com-tunnukset. Wordpress-sivut linkitetään wordpress.com-sivuston kanssa, jonka jälkeen kävijälaskuri on otettu toimintaan. Laskurin avulla voidaan seurata, paljonko sivulla on ollut vierailijoita tiettyinä ajanjaksona. Tämä ominaisuus on erityisen hyödyllinen yrityksille, jotka haluavat tietää sivujensa mainostuksen hyödyn.

5.3.2 Contact Form 7

Contact form on Wordpressin yksi suosituimmista sähköpostilomakkeiliitännäisistä. Sitä päivitetään säännöllisesti, joten sen käyttö on turvallista. Tässä työssä tätä lisäosaa käytetään monessa paikassa. Lomakkeen täyttäjän painettua lähetä-nappia, lomakkeeseen täytetyt tiedot lähetetään lisäosassa määritettyyn sähköpostiosoitteeseen. Sähköpostiin saapuvan viestin muotoilua voidaan säädellä jonkin verran lisäosan asetuksissa.

5.3.3 Better WP Security

Tämä lisäosa on erittäin hyödyllinen tietoturvan parantamiseksi. Wordpressin suojausta voidaan helposti parantaa pelkällä hiiren klikkauksella. Monet asetukset voivat kuitenkin sotkea koko asennuksen, ellei niiden kanssa olla tarkkana. Mahdollisista ongelmista ilmoitetaan selvästi, joten asennusta ei saa sekaisin vahingossa. Jos asetuksia kuitenkin halutaan kytkeä päälle, otetaan tietokannasta ensin varmuuskopio. Se onnistuu myös tällä lisäosalla, joka lähettää varmuuskopion määritettyyn sähköpostiin.

Lisäosan päänäkymässä luetellaan kaikki potentiaaliset uhat eri värikoodeilla, joista nähdään selvästi, missä suojauksissa on vielä parantamisen varaa. Kaikkia suojausasetuksia ei ole pakko pistää päälle, koska niissä kuvattujen kaltaiset uhat ovat melko epätodennäköisiä. Omaa harkintaa kannattaa käyttää asetuksia säätäessä.

5.3.4 Meta Slider

Responsive-teema vaatii erillisen lisäosan, jotta sivuilla voidaan näyttää diaesitys. Lisäosan avulla määritellään diaesityksen koko, diojen vaihtumisviive ja monia muita asetuksia. Ulkonäöksi voidaan valita neljä erilaista teemaa. Diaesitys voidaan lisätä minne tahansa sivulle, elementtiin tai widgettiin.

5.3.5 Admin menu editor

Tämän lisäosan ainoa tarkoitus on pistää Wordpressin hallintapaneelin valikko haluttuun järjestykseen, jotta navigointi olisi mahdollisimman vaivatonta. Kaikkia Wordpressin valikon linkkejä, kuten käyttäjien määrittämistä ja artikkeleiden luomista tarvitaan niin harvoin, että ne on parempi siirtää syrjään. Asetukset, sivujen sisältö ja lisäosat eritellään omaksi nipukseen, jolloin kaikki tarvittavat linkit löytyvät nopeasti.

5.4 Selainten yhteensopivuus

Nykyään nettiselaimia on saatavilla useita erilaisia. Yleisimmät ovat Google Chrome, Mozilla Firefox, Opera, Internet Explorer ja Safari. Näistä jokainen tul-

kitsee sivujen koodia hieman eri tavalla, joten ne saattavat myös näyttää erilaisilta selaimesta riippuen. Tämä johtaa helposti ongelmiin, jos sivuja kehitetään ainoastaan yhdellä selaimella ja jätetään muut huomioimatta. Varsinkin CSS-tyylitiedoston muokkaaminen saattaa vaikuttaa erilailla eri selaimen sivunäkymään. Esimerkiksi Internet Explorerin vanhemmat versiot eivät tue monen muun selaimen tapaan kulmanpyöristyksiä, joten sivujen ulkonäöllä on pientä eroa selainten välillä. Tällainen pieni kauneusvirhe ei ole kuitenkaan kovin vakava, vaan on syytä keskittyä ehkäisemään suurempia ongelmia.

Yksinkertainen tapa välttää suurimmat ongelmat on asentaa suosituimmat selaimet omalle koneelle ja tarkastaa sivunäkymät aina tasaisin väliajoin sivujen kehittämisen aikana. Tämä ei kuitenkaan ratkaise kaikkea, sillä selaimista on vielä olemassa monia eri versioita, joita ei voida asentaa koneelle samaan aikaan. Toinen ongelma on eri käyttöjärjestelmät, joiden tuntemattomimmat selaimet saattavat näyttää sivut eri tavalla. Onneksi kehittäjien avuksi on olemassa palveluita, joilla voidaan testata sivujen yhteensopivuutta.

Toisenlaisen haasteen asettaa nykyinen älypuhelinien suosio nettiselailussa. Puhelimia on monenlaisia ja myös niiden selaimet vaihtelevat laidasta laitaan. Pahimmassa tapauksessa suurelle näytölle suunnitellut sivut näyttävät aivan toisenlaisilta pienessä puhelimen ruudussa.

Tärkeintä on kuitenkin varmistaa, että ainakin suosituimmat selaimet näyttävät sivut suurin piirtein oikein, koska suurella todennäköisyydellä valtaosa sivuilla vierailevista ihmisistä käyttää jotain suosittua selainta, oli sitten kyseessä pöytätietokone tai älypuhelin. Vaikka kaikkia vierailijoita tulisi pyrkiä kohtelemaan tasapuolisesti, on kuitenkin tärkeää huomioida massojen tarpeet.

Sivujen kehitysvaiheessa päädyttiin käyttämään neljää eri selainta. Pääselaimena käytettiin Firefoxia ja rinnalla Operaa, Internet Exploreria sekä Chromea. Ulkoasu tarkistettiin kolmella jälkimmäisellä aina säännöllisin väliajoin ja ongelmakohdat korjattiin niiden ilmaantuessa.

6 KOTISIVUJEN TOTEUTUS

Tässä luvussa kerrotaan web-sivujen toteutuksen eri vaiheista. Tässä opinnäyte-työssä tehdään siis kolmet eri web-sivut, mutta ne eroavat toisistaan sen verran vähän, että niihin keskitytään kokonaisuutena. Siitä huolimatta toteutusvaihe ku-vaillaan mahdollisimman yksityiskohtaisesti.

Kaikki web-sivuilla näkyvät kuvat muokattiin tai luotiin käyttämällä Adobe Pho-toshop CS5:sta. Internetistä ladatut ilmaiskuvat muokattiin sivuille sopivaan muo-toon ja niihin saatiin lisätä jotain oman mielikuvituksen tuotoksia. Otsikkokuvat luotiin käyttämällä erilaisia fontteja ja efektejä, sekä ottamalla läpinäkyvyys käyt-töön tehostamaan ulkoasua. Sliderien kuvat räätälöitiin jokaiseen slideriin erik-seen.

6.1 Sivuston rungon muutokset

Kaikilla sivuilla käytetään siis Responsive-teemaa tai tarkemmin sanottuna sen lapsiteemaa. Koska se asennettiin jo aikaisemmin, voitiin aloittaa muutosten te-keminen. Ensimmäiseksi muutettiin web-sivujen osoiterakennetta, jotta osoite nä-kyisi mahdollisimman selvästi vierailijoilla. Oletuksena välilehtisivut näytettiin id-numeroina. Esimerkiksi uutiset-välilehtisivu näytti osoiterivillä seuraavalta: <http://www.esim.fi/sivusto/?p=123>. Se kuitenkin haluttiin näyttämään selkeäm-mältä eli <http://www.esim.fi/sivusto/uutiset>. Tämän asetuksen muuttaminen oli helppoa Wordpressin asetuksista.

Seuraavaksi asetettiin itse luotu favicon-ikoni sivuille lisäämällä koodirivi (**Kuva 12.**) lapsiteeman header.php-tiedoston `<head></head>`-tagien väliin. Tämän jäl-keen ikoni näkyi selaimen välilehdessä.

```
<link rel="shortcut icon" type="image/x-icon" href="http://www.esimerkki.fi/kansio/wp-  
content/uploads/2013/03/favicon.ico">
```

Kuva 12. Faviconin lisäämiseen tarvittava koodi header.php-tiedostossa.

Seuraavaksi luotiin kaikki web-sivuille tulevat välilehtisivut ja niille määritettiin haluttu asettelumalli. Asettelumallin valinta riippui välilehden tarkoituksesta.

Esimerkiksi yhdelle web-sivustolle toimeksiantaja halusi työhakemuslomakkeen näkymään jokaisella sivulla, joten kaikille sivuille valittiin asetelumalliksi Content/Sidebar. Tällaiselle sivulle voitiin lisätä widgettejä sivupalkkiin. Itse sisältö vie noin 66 % leveydestä ja sivupalkki loput. Näitä asetuksia pystyttiin vielä muuttamaan tyylitiedostossa.

Nyt kun välilehdet oli luotu, voitiin keskittyä valikon luomiseen. Wordpressin valikko-asetuksissa on mahdollista luoda erilaisia valikoita. Luodun valikon voi sijoittaa neljään eri kohtaan web-sivuilla, mutta tässä työssä käytettiin vain kahta eri paikkaa, päävalikkoa ja ylävalikkoa. Uusi valikko luotiin painamalla ”lisää valikko”. Siihen voitiin nyt lisätä juuri äsken luodut välilehdet, joiden nimeä ja järjestystä sai vapaasti muokata. Kun välilehtien nimet ja järjestys tuntuivat sopivilta, tallennettiin valikko ja sen jälkeen päätettiin, missä kohtaa web-sivuja se näytettäisiin. Nyt valikon runko oli valmis.

Web-sivujen sisältö haluttiin pitää tietyn alueen sisällä sivuttaissuunnassa, joten sivujen sisällön maksimileveydeksi asetettiin tyylitiedostossa 960 pikseliä (**Kuva 13**). Tällainen leveys aiheuttaa paljon tyhjää tilaa sisällön molemmille puolille, mutta se on suositeltu tapa tiivistää sivuja. Olisi aivan turhaa ottaa käyttöön esimerkiksi 1920x1080 resoluution täysi leveys, koska se saisi sivut näyttämään väljiltä ja sekavilta. Tällöin pienempien resoluutioiden käyttäjät saattaisivat nähdä sivuston aivan erilaisina, koska tietyt elementit siirtyisivät jonnekin muualla, mihin ne oli alun perin sijoitettu. 960 pikselin leveydessä on myös se etu, että nykyisissä tietokonenäyttöissä on vähintään 1024 pikseliä leveyssuunnassa. Näin ollen sisältöalue näyttää samalta näytön resoluutiosta riippumatta. Vain sisältöalueen oikealla ja vasemmalla puolella olevan tyhjän tilan määrä vaihtelee.

```
#container {  
    max-width: 960px;  
}
```

Kuva 13. Web-sivuston sisältöalueen maksimileveyden määrittäminen.

6.2 Sivuston rakenne ja asettelu

Kuten edellä jo mainittiin, kaikki kolme web-sivustoa ovat melko samanlaisia, mutta eroja löytyy tiettyjen elementtien sijoittelussa ja asettelussa. Sivustojen yläosat eli headerit, sekä alaosat eli footerit eroavat jonkin verran. Samoin diaesityksen eli sliderin sijoittelu vaihtelee.

6.2.1 Header

Sivujen headerissa on kahdella sivulla otsikkokuva, mutta yhdessä sen tilalla on slider. Myös hakukenttä ja Facebook-linkki sijoitettiin headeriin, tosin pienin eroavaisuuksin.

Headeriin otsikkokuvan sijoittaminen on helppoa Wordpressin teemanhallinnassa, mutta siinä on tietyt rajoitukset, joten kuvan asettaminen hoidettiin tyylitiedoston avulla. Näin otsikkokuvan tila ei ollut niin rajattu eli se pystyttiin halutessa jopa sijoittamaan osittain valikon alle, joka toi moniulotteisemman vaikutelman. Kuvan lisääminen onnistui lisäämällä *background*-sääntö *#container*-valitsimen sisään (**Kuva 14**). Arvoina on lisäksi *no-repeat*, joka estää kuvan toistamisen sivutaissuunnassa, jos kuva on kapeampi kuin 960 pikseliä. *Top left* asettaa kuvan vasempaan ylälaitaan. Otsikkokuvan leveyden oli syytä olla maksimissaan 960 pikseliä eli sivuston sisällön levyinen, jotta turhalta säätämiseltä vältyttiin.

```
#container {
    max-width: 960px;
    background: transparent url(http://www.esim.fi/.../otsikko.png) no-repeat top left;
}
```

Kuva 14. Otsikkokuvan lisääminen.

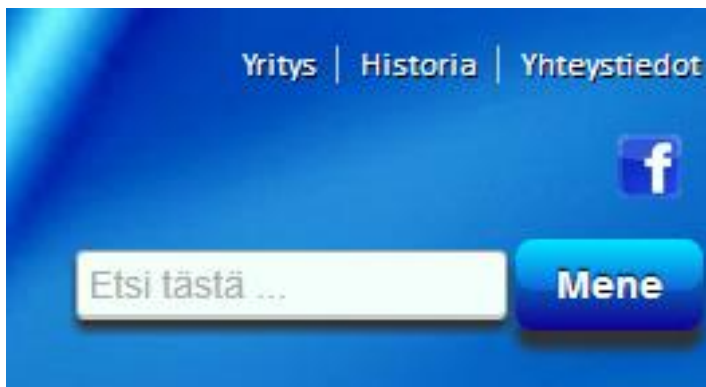
Yhdelle sivustolle toimeksiantaja halusi headeriin hakukentän, jonka avulla vierailija voi etsiä sisältöä web-sivuilta. Sen lisääminen onnistui lisäämällä koodia (**Kuva 15**) *header.php*-tiedostoon. Tämä aktivoi hakukentän ja se ilmestyi tallentamisen jälkeen sivuston yläosaan. Tyylitiedostoon oli tehtävä pieniä muutoksia, jotta hakukentän paikka voitiin määrittää halutuksi.

```
<div id="search-box">
    <?php get_search_form(); ?>
</div><!-- end of #search-box -->
```

Kuva 15. Hakukentän lisääminen sivustolle.

Hieman samalla tavalla hoidettiin Facebook-ikonin siirtäminen footerista headeriin. Tämä onnistui siirtämällä kaikki social-icon-lauseet footer.php-tiedostosta header.php-tiedostoon. Nyt sosiaaliset ikonit eivät enää näkyneet footerissa, vaan headerissa. Ensiksi piti tosin aktivoida haluttu sosiaalinen ikoni lisäämällä WordPressin teeman asetuksiin Facebook-linkki. Ikonin sijoittelu hoidettiin samaan tapaan kuin hakukenttäkin.

Valmis hakukentän ja Facebook-linkin sijoittelu nähdään kuvassa 16.



Kuva 16. Hakukenttä ja Facebook-linkki sijoitettuna headerin oikeaan reunaan.

6.2.2 Footer

Footerista siis poistettiin sosiaaliset ikonit, mutta niiden tilalle asetettiin yrityksen tiedot tekijänoikeustietojen lisäksi. Näitä olivat yrityksen nimi, osoite, puhelinnumero, sähköposti ja y-tunnus. Tiedot oli kätevä sijoittaa footer.php-tiedoston *copyright*-elementin sisään, kuten kuvassa 17.

```

<div class="grid col-300 copyright">
    esc_attr_e('&copy;', 'responsive'); ?> <?php _e(date("Y")); ?><a href="<?php echo home_url('/') ?>" ti-
    tle="<?php echo esc_attr(get_bloginfo('name', 'display')); ?>">
    <?php bloginfo('name'); ?>
    <?php printf('<br>Osoite, Postinumero Toimipaikka<br>Puhelin, sähköposti'); ?>
</div>

```

Kuva 17. Yrityksen tietojen lisääminen footeriin.

Sivuilla näkyvän footerin teksti näytettiin oletuksena mustana, joten sitä muutettiin tyylitiedostossa. Copyright-valitsimen tekstinväri eli *color* muutettiin valkoiseksi *#fff* ja *font-weight* asetettiin lihavoiduksi eli *bold* (**Kuva 18**).

```

.copyright {
    color: #fff;
    font-weight: bold;
    text-align: left;
}

```

Kuva 18. Copyright-elementin tekstin ulkonäön muokkaaminen.

Näillä pienillä muutoksilla footer saatiin näyttämään halutut tiedot. Footerin ulkonäköä oltaisiin voitu myös muuttaa, mutta sille ei nähty tarvetta.

6.2.3 Widgetit

Widgetit ovat olennainen osa Responsive-teemaa ja niitä pyrittiin käyttämään mahdollisimman tehokkaasti hyväksi. Ensimmäisenä ne osuvat vierailijan silmään etusivulla, jossa niitä on eri paikoissa sivustosta riippuen. Niitä käytetään myös monilla muilla välilehtisivuilla.

Widgettien sisältöön voidaan vaikuttaa teeman asetuksissa, jossa widgettejä hallitaan. Oletuksena kaikki widgetit olivat tyhjiä eli niihin piti lisätä jotain sisältöä, jotta ne näkyisivät sivustolla. Etusivun widgetteihin lisättiin poikkeuksellisesti vain web-sivun esittelytekstiä, mutta muille välilehdille haluttiin esimerkiksi työhakemuslomake ja pienoivalikko.

Pienoisvalikon luomiseksi tehtiin uusi valikko, jonne lisättiin tarvittut linkit. Sen jälkeen widget-asetuksissa vedettiin ”mukautettu valikko” vasemman sivupalkin widgettiin. Lopuksi valittiin valikoksi äsken luotu valikko. Tämän jälkeen valikko nähtiin sivun vasemmassa sivupalkissa.

Samaan tapaan toteutettiin työhakemuslomakkeen sijoittaminen widgettiin. Tarvi- taan vain Contact Formin [*contact-form-7 id="4" title="Työhakemuslomake"*] – koodi, joka sijoitettiin tässä tapauksessa oikea sivupalkki –widgettiin.

6.2.4 Lomakkeet

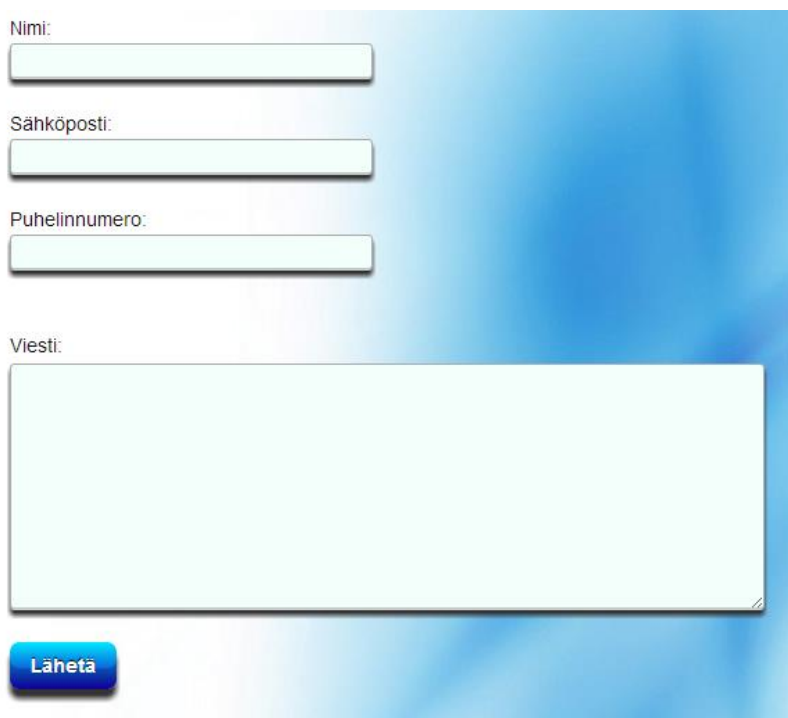
Lomakkeen luominen tapahtui Contact Form –lisäosassa. Lomakkeen tekstikent- tinä on esimerkiksi nimi, sähköposti, puhelinnumero, viesti ja lomaketyypistä riippuen myös syntymäaika sekä työpaikkakunta. Tekstikentät sijoitettiin allek- kain ja niiden sijoittelua muokattiin hieman käyttämällä taulukkorakennetta. Ku- vassa 19 nähdään koko yhteydenottolomakkeen rakenne.

```
<table>
  <tr>
    <td width="50%">
      <p>Nimi:<br /> [text* your-name] </p>
      <p>Sähköposti:<br /> [email* your-email] </p>
      <p>Puhelinnumero:<br /> [text* your-number] </p>
    </td>
    <td width="50%">
      </td>
  </tr>
  <tr>
    <td width="100%" colspan=2>
      Viesti:<br />[textarea* teksti]
      <p>[submit "Lähetä"]</p>
    </td>
  </tr>
</table>
```

Kuva 19. Yhteydenottolomakkeen HTML-koodia.

Taulukko luotiin, jotta tekstikentistä ei tulisi tarpeettoman leveitä. Solun leveydeksi määritettiin 50 % sisältöalueen leveydestä. Ensimmäisen solun sisälle tuli normaalit tekstikentät. Ensin esitetään tekstikentän nimi ja sitten itse kenttä, joka näkyy esimerkiksi `[text* your-name]` -tagin avulla. Tähti-merkki tagin perässä tarkoittaa sitä, että kenttä ei saa olla tyhjänä sitä lähettäessä. Jos kenttä jätetään tyhjäksi ja painetaan lähetä-nappia, ilmoitetaan vierailijalle virheestä.

Taulukon toinen rivi varattiin kokonaan tekstialueelle, jolle annettiin *colspannin* avulla koko taulukon leveys käyttöön. Alapuolelle sijoitettiin lähetä-nappi. Kuvassa 20 nähdään valmis lomake.



The image shows a web form with a light blue background. It contains four input fields and one button. The fields are labeled 'Nimi:', 'Sähköposti:', 'Puhelinnumero:', and 'Viesti:'. The 'Viesti:' field is a large text area. Below the fields is a blue button with the text 'Lähetä'.

Kuva 20. Käyttövalmis lomake web-sivulla.

Kun vierailija on täyttänyt lomakkeen ja painanut lähetä-nappia, lähetetään lomakkeen tiedot sähköpostina ylläpitäjän osoitteeseen. Viestin runko nähdään kuvassa 21.

Nimi: [your-name]

Sähköposti: [your-email]

Puhelinnumero: [your-number]

Viesti:

[teksti]

Kuva 21. Yhteydenottolomakkeen sähköpostiviestin rakenne.

Lomakkeen tiedot sijoitetaan viestin tageihin. Järjestystä voidaan vapaasti muokata ja omaa tekstiä saa lisätä tarpeen mukaan.

6.2.5 Sliderit

Jokaiselle tämän opinnäytetyön web-sivulle lisättiin yksi slider ja ne kaikki ovat eri kohdissa sivustoja. Yksi sijoitettiin headeriin, toinen valikon alle ja kolmas etusivun sisältöalueelle. Kaksi ensimmäistä näkyy jatkuvasti vierailijalle välilehdestä riippumatta ja näiden sliderien leveydeksi määritettiin sisältöalueen leveys eli 960 px. Kolmas slider nähdään vain etusivun sisältöalueella, joten sen koko on pienempi kuin kahdella muulla.

Kaikilla slidereilla on muutama kuva, joita pyöritetään. Niitä voidaan myös vaihtaa manuaalisesti sliderissa näkyvästä nuolesta. Myös sliderin alapuolella olevista kuvien määrän kertovista pisteistä voidaan valita näytettävä kuva.

Sliderin ulkonäkömuutoksiin ei voitu vaikuttaa ollenkaan, vaan oli valittava valmiina olevista teemoista sivustoon sopiva. FlexSlider-niminen tema valittiin kaikille sivuille ja kuvien vaihdon efektiksi vaihdettiin häivytytys.

Kuvien lisääminen slideriin onnistui helposti lisäosan asetuksista. Kuvien koko muutettiin kuvankäsittelyohjelmalla samaksi kuin sliderin asetuksissa, jotta lisäosa ei skaalaisi itse väärän kokoista kuvaa huonompilaatuiseksi.

Sliderin lisääminen eri elementteihin tapahtui lisäämällä PHP-tiedostoihin kuvan 22 koodirivi.

```
<?php echo do_shortcode("[metaslider id=201]"); ?>
```

Kuva 22. PHP-tiedostoihin lisättävä koodi.

Sijoittamisen kanssa oltiin tarkkana, sillä väärässä kohtaa se aiheutti virhetilanteita. Tämän jälkeen muokattu PHP-tiedosto kopioitiin lapsiteeman kansioon, jotta teemapäivitykset eivät kumoaisi muutoksia tulevaisuudessa.

Normaalille sivulle, kuten etusivun sisältöalueelle, ei tarvinnut lisätä kuin pelkkä *[metaslider id=201]*.

6.3 Ulkoasun muokkaus

Nyt kun web-sivustojen rakenne, runko ja asettelu oltiin saatu kuntoon, voitiin aloittaa ulkoasun muokkaaminen. Suurin osa ajasta käytettiin eri muotojen ja tyylien kokeilemiseen, jotta sopivat yhdistelmät löydettäisiin.

Web-sivujen oletusfontti, sen koko ja väri todettiin sopiviksi, joten niitä ei muutettu.

6.3.1 Taustakuva

Taustakuvan asettaminen onnistui helposti teeman asetuksissa. Opinnäytetyön toimeksiantaja halusi värimaailmaksi sinisen, joten taustakuviksi valittiin sinisiä vektorikuvia. Niitä sai ladata vapaaseen käyttöön eri kuvasivustoilta. Taustakuva ladattiin kotikoneelta WordPressin avulla suoraan palvelimelle, jonka jälkeen määritettiin kuvan sijainti- ja vieritysasetukset. Kuva asetettiin pysymään aina keskellä ja paikoillaan, vaikka sivua vieritettäisiinkin. Tämän ansiosta kuvaa ei koskaan toisteta. Taustakuvan koko on 2000x1100 pikseliä, jotta se kattaa periaatteessa kaikki mahdolliset tietokonenäytöt. Nykyään yleisin maksimiresoluutio on 1920x1080, joten taustakuva on tarpeeksi suuri, eikä taustaväri pääse koskaan näkymään ja pilaamaan ulkoasua.

Taustakuvan tiedostokoko pyrittiin pitämään mahdollisimman pienenä, jotta sen lataaminen ei kestäisi kovin kauaa, jos vierailijalla on hidas internet-yhteys.

6.3.2 Valikon muokkaus

Valikon ulkonäköä pystyttiin muokkaamaan monella tavalla. Tyylietiedostossa oli mahdollista määrittää valikon väri ja myös käyttää kahta väriä yhtä aikaa liukuväriä *background-image*-ominaisuuden *linear-gradient*-arvolla. Tämä tarkoittaa siis sitä, että esimerkiksi valikon yläosa on vaaleansininen ja alaspäin mennessä väri vaihtuu tasaisesti tummansiniseen. Kun tämän päälle vielä asetetaan valikon sisään suunnattu himmeä ja läpinäkyvä varjostus eli *box-shadow*:n *inset*, saadaan valikkoon niin sanottu lasiefekti. Insetin perässä tulevalla arvolla asetetaan vielä musta varjo valikon alle.

Valikon leveys riippui sivustosta, mutta kuvassa 23 se määritettiin 690 pikselin levyiseksi. Valikon kulmat pyöristettiin käyttämällä ominaisuutta *border-radius*. Pyöristäminen vaikutti automaattisesti kaikkiin asetettuihin varjoihin, joten kaikkia suunniteltuja ulkonäkömuutoksia ei voitu toteuttaa täysin ilman pieniä ongelmia. Ensimmäiset muutokset valikon ulkonäköön nähdään kuvassa 23.

```
.menu {
    background-color: transparent;
    background-image: -webkit-gradient(linear, left top, left bottom, from(#00e9ff), to(#060082));
    background-image: -webkit-linear-gradient(top, #00e9ff, #060082);
    background-image: -moz-linear-gradient(top, #00e9ff, #060082);
    background-image: -ms-linear-gradient(top, #00e9ff, #060082);
    background-image: -o-linear-gradient(top, #00e9ff, #060082);
    background-image: linear-gradient(top, #00e9ff, #060082);
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr=#00e9ff, endColorstr=#060082);
    clear: both;
    -moz-box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.2) , 0pt 10px 8px #333333;
    -webkit-box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.2), 0pt 10px 8px #333333;
    -o-box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.2), 0pt 10px 8px #333333;
    -ms-box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.2), 0pt 10px 8px #333333;
    box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.2) , 0pt 10px 8px #333333;
    width: 690px;
    margin: 0 auto;
    border-radius:20px;
}
```

Kuva 23. Valikon ulkonäön muokkausta.

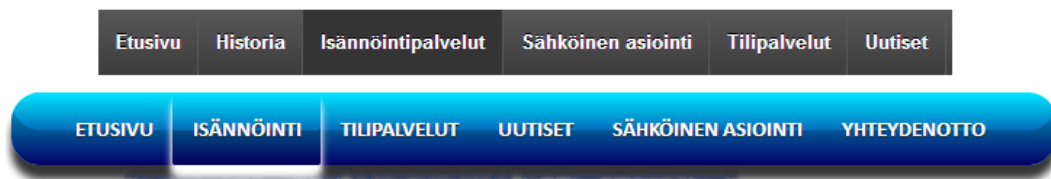
Seuraavaksi muutettiin valikon välilehtilinkkien ulkonäköä tietyissä tilanteissa. Kun hiiri on jonkin linkin päällä eli *hover*, sen ulkonäköä korostetaan jonkin verran. Sama tapahtuu aktiivisen välilehden, *current-menu-item*, linkissä eli se erottuu selvästi muusta valikosta ja helpottaa näin vierailijan asiointia. Muutokset olivat kuitenkin pieniä, sillä pelkillä varjoilla saatiin haluttu lopputulos. Pohjalle tarvittiin sama liukuväri kuin kuvassa 23. Kuvassa 24 asetetaan myös *inset*-varjo eli se tulee jo aikaisemmin määritetyn varjon päälle, joka tummentaa linkin alaosan väriä tarpeeksi paljon. Linkin alle asetettiin valkoinen varjo, jonka ansiosta linkki erottui selvästi valikosta. *Hover*-linkin varjo määritettiin isommaksi, jotta se ei sekoittuisi aktiivisen linkin kanssa.

```
.menu a: hover {
    background-color: transparent;
    background-image: -webkit-gradient(linear, left top, left bottom, from(#00e9ff), to(#060082));
    background-image: -webkit-linear-gradient(top, #00e9ff, #060082);
    background-image: -moz-linear-gradient(top, #00e9ff, #060082);
    background-image: -ms-linear-gradient(top, #00e9ff, #060082);
    background-image: -o-linear-gradient(top, #00e9ff, #060082);
    background-image: linear-gradient(top, #00e9ff, #060082);
    -moz-box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.3) , 0pt 10px 8px #fff;
    -webkit-box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.3) , 0pt 10px 8px #fff;
    -o-box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.3) , 0pt 10px 8px #fff;
    -ms-box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.3) , 0pt 10px 8px #fff;
    box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.3) , 0pt 10px 8px #fff;
    border-radius:2px;
}

.menu .current_page_item a,
.menu .current-menu-item a
{
    background-color: transparent;
    -moz-box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.3) , 0pt 5px 5px #fff;
    -webkit-box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.3) , 0pt 5px 5px #fff;
    -o-box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.3) , 0pt 5px 5px #fff;
    -ms-box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.3) , 0pt 5px 5px #fff;
    box-shadow: inset 0pt -24px 0pt rgba(0,0,0,0.3) , 0pt 5px 5px #fff;
    border-radius:2px;
}
```

Kuva 24. Aktiivisen linkin ja hover-linkin muutokset.

Kuvassa 25 nähdään valikon oletusulkonäkö ja sen alla valmis muokattu valikko.



Kuva 25. Valikko ennen ja jälkeen muokkauksen.

6.3.3 Napin ja tekstikentän muokkaus

Lomakkeiden alla on aina lähetä-nappi, jota painamalla vieras lähettää täyttämiesä tekstikenttien tiedot ylläpitäjälle. Oletuksena napin ulkoasu oli harmaa ja yksilotteinen, mutta se haluttiin sopimaan web-sivujen sinisen teeman kanssa yhteen. Napin muokkaaminen toteutettiin melkein samalla tavalla kuin valikonkin. Napille annettiin hieman poikkeavat liukuvärit ja tekstin väri muutettiin valkoiseksi. Kuvassa 26 nähdään tyylitiedoston *a.button*-valitsimen ominaisuudet, jotka koskevat kaikkia nappityyppejä. Napin oletusulkonäkö ja muokkauksen jälkeinen ulkonäkö nähdään kuvassa 27.

```
a.button,
input[type='reset'],
input[type='button'],
input[type='submit'] {
    background-color: transparent;
    background-image: -webkit-gradient(linear, left top, left bottom, from(#00e9ff), to(#1600c1));
    background-image: -webkit-linear-gradient(top, #00e9ff, #1600c1);
    background-image: -moz-linear-gradient(top, #00e9ff, #1600c1);
    background-image: -ms-linear-gradient(top, #00e9ff, #1600c1);
    background-image: -o-linear-gradient(top, #00e9ff, #1600c1);
    background-image: linear-gradient(top, #00e9ff, #1600c1);
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr=#00e9ff, endColorstr=#1600c1);
    -moz-box-shadow: inset 0pt -15px 0pt rgba(0,0,0,0.2) , 0pt 7px 5px #333333;
    -webkit-box-shadow: inset 0pt -15px 0pt rgba(0,0,0,0.2) , 0pt 7px 5px #333333;
    -o-box-shadow: inset 0pt -15px 0pt rgba(0,0,0,0.2) , 0pt 7px 5px #333333;
    -ms-box-shadow: inset 0pt -15px 0pt rgba(0,0,0,0.2) , 0pt 7px 5px #333333;
    box-shadow: inset 0pt -15px 0pt rgba(0,0,0,0.2) , 0pt 7px 5px #333333;
    border: 0px solid #ddd;
}
```

```

border-radius: 7px;
color: #fff;
cursor: pointer;
display: inline-block;
font-size: 14px;
font-weight: 700;
line-height: 20px;
margin: 0;
padding: 7px 15px;
text-decoration: none;
text-shadow: 0px -1px 1px #000;
vertical-align: middle;
white-space: nowrap;
}

```

Kuva 26. Aktiivisen linkin ja hover-linkin muutokset.



Kuva 27. Nappi ennen ja jälkeen.

Napilla on vielä samat hover-ominaisuudet kuin valikollakin.

Myös tekstikenttien oletusulkoasu oli napin tapaan tylsä, joten sille annettiin uutta ilmettä lisäämällä varjot ja muuttamalla kentän väriä vaaleansinisiksi. Reunoja pyöristettiin aavistuksen. Samat säännöt koskivat myös tekstialuetta. Muutokset nähdään kuvassa 28 ja lopputulos kuvassa 29.

```

select,
input[type="text"],
input[type="password"],
input[type="email"],
area,
textarea {
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    -moz-box-shadow: inset 0pt -1px 0pt rgba(0,0,0,0.2) , 0pt 4px 4px #333333;
}

```

```

-webkit-box-shadow: inset 0pt -1px 0pt rgba(0,0,0,0.2), 0pt 4px 4px #333333;
-o-box-shadow: inset 0pt -1px 0pt rgba(0,0,0,0.2), 0pt 4px 4px #333333;
-ms-box-shadow: inset 0pt -1px 0pt rgba(0,0,0,0.2), 0pt 4px 4px #333333;
box-shadow: inset 0pt -1px 0pt rgba(0,0,0,0.2) , 0pt 4px 4px #333333;
background-color: #f2ffa;
box-sizing: border-box;
border: 1px solid #aaa;
border-bottom-color: #ccc;
border-radius: 3px;
margin: 0;
outline: none;
padding: 4px 4px;
vertical-align: top;
width: 100%;
}

```

Kuva 28. Tekstikentän ja tekstialueen muokkausta.



Kuva 29. Tekstikenttä ennen ja jälkeen.

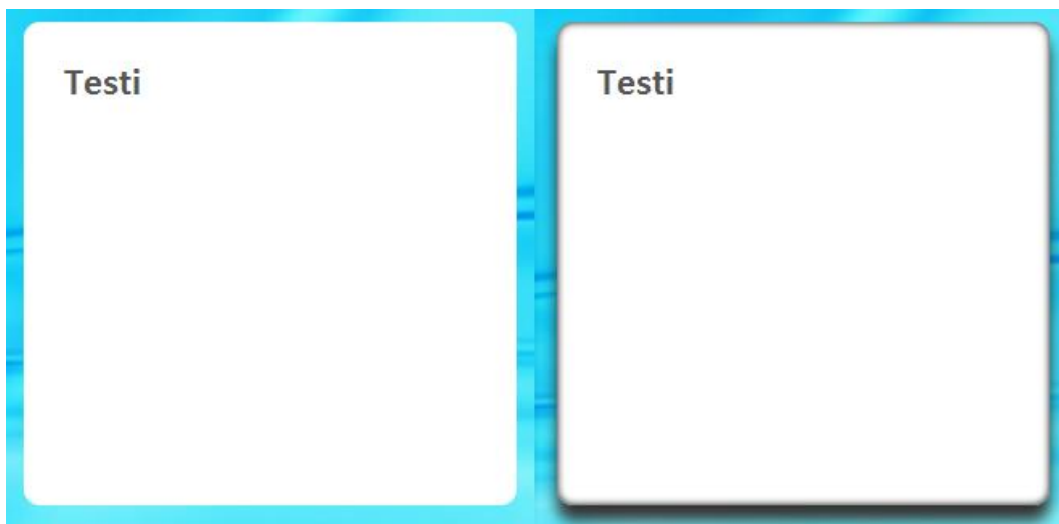
Näiden muutosten jälkeen napit ja tekstikentät sopivat loistavasti web-sivujen yleisilmeeseen.

6.3.4 Wrapper

Wrapperin eli sisältöalueen pohjalle tehtiin myös pieniä muutoksia. Pohjaväriä säilytettiin valkoinen, mutta tähänkin lisättiin varjot ulos ja sisälle. Joitain kulmia pyöristettiin enemmän kuin toisia. Jotta sisältö ei olisi missään vaiheessa kiinni reunoissa, määritettiin *margin* sen verran suureksi, että sisältö on aina sopivalla etäisyydellä reunoista. Muutokset nähdään kuvassa 30. Esimerkki wrapperin muokatusta ulkonäöstä nähdään kuvassa 31.

```
#wrapper {  
    background-color: #fff;  
    border: 0px solid #e5e5e5;  
    clear: both;  
    margin: 20px auto 20px auto;  
    padding: 0px 40px 20px 20px;  
    position: relative;  
    height:auto;  
    min-height: 650px;  
    -moz-box-shadow: inset 0 0 5px black , 0pt 10px 10px #333333;  
    -webkit-box-shadow: inset 0 0 5px black , 0pt 10px 10px #333333;  
    -o-box-shadow: inset 0 0 5px black , 0pt 10px 10px #333333;  
    -ms-box-shadow: inset 0 0 5px black , 0pt 10px 10px #333333;  
    box-shadow: inset 0 0 5px black , 0pt 10px 10px #333333;  
    border-top-left-radius:20px;  
    border-top-right-radius:120px;  
    border-bottom-right-radius:20px;  
    border-bottom-left-radius:20px;  
}
```

Kuva 30. Wrapperin muokkausta.



Kuva 31. Esimerkki wrapperin muutoksista ennen ja jälkeen (kuvassa widgetti).

Wrapperiin on myös mahdollista lisätä taustakuva taustaväriin sijaan samaan tapaan kuin aikaisemmin mainitun containerin kanssa (**Kuva 14.**), ja näin tehtiinkin yhden web-sivun osalta.

Joissakin tapauksissa widgettien taustavärit voitiin poistaa kokonaan, koska wrapperin tausta riitti täyttämään molempien tarpeet, ja näin välttyttiin widgettien taustan tuleminen wrapperin reunojen tai muiden osien päälle. Erityisesti eri selainten välillä edellä mainittu tilanne olisi saattanut olla mahdollinen.

6.4 Tietoturva

Tietoturva oli hoidettava huolellisesti, sillä netistä löytyy aina ihmisiä, jotka haluavat tunkeutua sivustoille ja tehdä mahdollisimman paljon ilkeävaltaa. Jo aikaisemmin esitelty Wordpress-lisäosa Better WP Security hoitaa oikeastaan kaikki tarvittavat tietoturva-asiat. Jo aikaisemmin tietokannan etutunnisteet ja käyttäjätunnus asetettiin vaikeasti arvattaviksi, mutta jäljelle jäi vielä muutamia muutosta tarvitsevia kohtia.

Ensimmäiseksi lisäosa kysyi lupaa muuttaa admin-käyttäjän ID-numeroa. Oletus-ID saattaa lisätä tietoturvariskiä, koska se on helposti arvattavissa.

Seuraavaksi estettiin tietyt mustalla listalla olevat IP-osoitteet, joista ei nyt pääse ollenkaan sivustoille.

Jotta hakkereiden botit eivät pystyisi loputtomasti arvailemaan hallintapaneeliin kirjautumiseen tarvittavia tunnuksia, määritetään lisäosassa, kuinka monesti samasta IP-osoitteesta voidaan yrittää kirjautua, ennen kuin se estetään kokonaan määritetyksi ajaksi.

.htaccess- ja wp-config.php-tiedostojen muokkaaminen Wordpressillä estettiin, joten niihin tehtävät muutokset on pakko tehdä manuaalisesti käyttämällä esimerkiksi FileZillaa.

Kuvassa 32 nähdään yhteenveto lisäosan tietoturva-asetuksista. Vihreät kohdat on kokonaan suojattu, oranssit osittain ja punaisia ei ollenkaan. Siniset on osittain suojattu, mutta saattavat aiheuttaa ongelmia joidenkin teemojen ja lisäosien kans-

sa. Vaikka listassa näkyy muutamia vain osittain suojattuja tai kokonaan suojaamattomia kohtia, ei niiden todettu aiheuttavan varsinaista uhkaa. Tärkeimmät ja suurimman uhan alla olevat kohdat oli kuitenkin suojattu.

Yksi tärkeä asia tietoturvan kannalta on päivittää Wordpress, sen lisäosat ja teema säännöllisin väliajoin. Uusia Wordpress-versioita tulee harvemmin kuin lisäosien päivityksiä, joita saattaa tulla jopa viikoittain. Näiden päivittäminen jätetään kokonaan toimeksiantajan vastuulle. Se ei kuitenkaan vaadi kuin päivitä-napin painamista eli toimenpide ei ole vaikea, eikä vaadi erityisosaamista. Erilaisten tulevien versioiden yhteensopivuusongelmien ilmaantuminen on kuitenkin mahdollista, joten sivujen toteuttaja on valmis auttamaan mahdollisissa ongelmatilanteissa myöhemmässäkin vaiheessa.

1. You are enforcing strong passwords, but not for all users. [Click here to fix.](#)
2. Your WordPress header is still revealing some information to users. [Click here to fix.](#)
3. Non-administrators cannot see available updates.
4. The *admin* user has been removed.
5. The user with id 1 has been removed.
6. Your table prefix
7. You are not scheduling regular backups of your WordPress database. [Click here to fix.](#)
8. Your WordPress admin area is available 24/7. Do you really update 24 hours a day? [Click here to fix.](#)
9. You are blocking known bad hosts and agents with HackRepair.com's blacklist.
10. Your login area is protected from brute force attacks.
11. Your WordPress admin area is not hidden. [Click here to fix.](#)
12. Your .htaccess file is partially secured. [Click here to fix.](#)
13. Your installation is actively blocking attackers trying to scan your site for vulnerabilities.
14. Your installation is not actively looking for changed files. [Click here to fix.](#)
15. Your installation accepts long (over 255 character) URLs. This can lead to vulnerabilities. [Click here to fix.](#)
16. You are allowing users to edit theme and plugin files from the WordPress backend. [Click here to fix.](#)
17. Better WP Security is allowed to write to wp-config.php and .htaccess.
18. wp-config.php and .htaccess are not writeable.
19. Version information is obscured to all non admin users.
20. You have renamed the wp-content directory of your site.
21. You are not requiring a secure connection for logins or for the admin area. [Click here to fix.](#)

Kuva 32. WP Better Security –lisäosan suojauksen yhteenveto.

Lopuksi luotiin muutama uusi käyttäjä Wordpressiin toimeksiantajaa varten. Näille tunnuksille annettiin vähemmän oikeuksia kuin admin-käyttäjälle. Luonti onnistui käyttäjienhallinnassa, jossa syötettiin käyttäjätunnus, sähköposti ja salasana tekstikenttiin. Salasana luotiin generaattorilla, jotta se olisi mahdollisimman turvallinen. Rooli, eli kuinka paljon käyttäjällä on oikeuksia, asetettiin esimerkiksi päätoimittajaksi. Sillä ei ole oikeuksia päivittää tai tehdä kriittisiä muutoksia sivustolle, mutta sisältöä voidaan lisätä.

7 TESTAUS

Nyt kun kaikki kolme web-sivustoa oli valmiita, oli syytä testata niiden toiminta. Erilaisten elementtien ja osien toimintaa testattiin tietenkin koko toteuttamisen ajan, mutta jotain oli silti saattanut jäädä huomaamatta.

Ensin tarkistettiin, että kaikkien sivujen osoiterakenne oli oikea. Toteutuksen alussa välilehtisivujen osoitteet muutettiin ID-numerosta muotoon <http://www.esim.fi/sivusto/uutiset>. Oli käytävä läpi myös kaikki välilehtien sisällä olevat yksittäiset linkit, jotta niiden polku olisi yhteydessä kyseessä olevaan välilehteen.

Seuraavaksi testattiin lomakkeet, jotka ovat alttiita ongelmille. Monen lomakkeen hiominen yhtä aikaa lisäsi mahdollisuutta tehdä jokin yksinkertainen kirjoitusvirhe, joka saattoi estää koko lomakkeen toiminnan. Ensin oli tarkistettava koodi puuttuvien HTML-tagien varalta. Myös lisäosan omat tagit tarkistettiin, jotta ne kaikki oli varmasti oikein kirjoitettu sekä esitysosassa että sähköpostiviestissä. Suurimmassa osassa lisäosan tageja piti olla *-merkki eli tieto oli pakko antaa lomakkeeseen ennen lähetystä, joten tämän merkin olemassaolo piti tarkistaa tiettyissä tageissa.

Koodin tarkistuksen jälkeen kaikki lomakkeet testattiin syöttämällä niihin tietoja, jotka sitten lähetettiin ylläpitäjälle lähetä-napilla. Lomakkeista lähetetty sisältö tarkistettiin sähköpostista, jotta kaikki tieto oli varmasti tullut perille. Yhdestä lisäosan tietokentästä löydettiin kirjoitusvirhe, jonka takia sähköpostiin tullessa viestissä ei näytetty lähettäjän sähköpostiosoitetta ollenkaan. Lopulta kaikki lomakkeet todettiin toimiviksi.

Sliderien toimintaa seurattiin tarkastamalla, että kuvat näytettiin oikein. Myös kuvien välillä oleva viive tarkastettiin, ettei kuva vaihdu liian nopeasti. Yhdestä sliderista puuttui autoplay-asetus, jonka takia kuvat eivät aina vaihtuneet. Se pistettiin päälle, jonka jälkeen slider toimi oikein.

Lopuksi tarkistettiin ulkoasu. Siihen käytettiin neljää suosituinta selainta, jotta näkyvä olisi mahdollisimman samanlainen jokaiselle vierailijalle. Kaikki sivut käy-

tiin läpi Chromella, Firefoxilla, Operalla ja Internet Explorerilla. Merkittäviä eroja ei löydetty, vaikka niitä oli hieman syytä odottaa. Voitiin vain todeta, että selaimet ovat nykyään päässeet parantuneen CSS-tuen myötä huomattavasti lähemmäs toisiaan.

Sivujen toteuttamiseen käytettiin suurimmaksi osaksi kannettavaa tietokonetta, jonka maksimiresoluutio on 1400x900 pikseliä, joten oli vielä syytä tarkistaa suuremman ja pienemmän resoluution näytön yhteensopivuus. Full HD-näytön 16:9-kuvasuhteen 1920x1080 resoluutio ei luonnollisesti muuttanut sivustojen ulkoasua muuta kuin lisäämällä tyhjää tilaa sisältöalueen molemmille puolille. Suurin kysymysmerkki oli vanhemman 4:3-kuvasuhteen sopivuus. Testaamiseen käytettiin 17-tuumaisen näytön 1280x1024 pikselin resoluutiota. Ongelmia ei havaittu eli toteutusvaiheessa määritetyn sisältöalueen leveyden valinta onnistui.

Ainut ongelma havaittiin, kun selaimen ikkunaa kavennettiin sivusuunnassa, jolloin sen leveys alitti sisältöalueen määritetyn 960 pikselin. Tämä muutti sivujen asettelua huomattavasti ja sotki monet elementit keskenään. Ongelman todettiin kuitenkin olevan lähinnä Wordpressin ”ominaisuus”, joten sille ei ollut paljoa tehtävissä. Sen ei kuitenkaan arveltu olevan merkittävä ongelma, sillä harvemmin vierailijat pienentävät selaimen ikkunaa alle kriittisen pikselimäärän. Tärkeintä oli, ettei ongelmia ilmaantuisi ikkunan ollessa täydessä koossaan.

Vaikka tämän työn web-sivuja ei missään vaiheessa suunniteltu mobiililaitteita silmällä pitäen, tarkistettiin niiden yleisilme Samsung Galaxy S2 -älypuhelimella. Sivujen huomattiin olevan yllättävän hyvässä kunnossa, joten niiden selaaminen onnistuisi ainakin jossain määrin tietyillä mobiililaitteilla.

8 JOHTOPÄÄTÖKSET JA POHDINTA

Projekti onnistui suunnitellusti. Vaikka web-sivuilla ei ollut varsinaisia kiveen hakattuja vaatimuksia ja ne toteutettiin pääsääntöisesti vapain käsin, saatiin sivuista toimeksiantajalle sopivat. Sain heiltä projektin eri vaiheissa erilaisia toivomuksia, joita toteutin aina mahdollisuuksien mukaan. Olimme siis tiiviisti yhteydessä koko projektin ajan, jonka ansiosta toimeksiantaja pystyi omalta osaltaan vaikuttamaan kehityksessä.

Projektin suorittamiseen oli noin kaksi kuukautta aikaa. Toimeksiantajalla ei ollut varsinaista kiirettä, vaan oma koulusta valmistuminen painoi päälle, joten tämä työ oli saatava valmiiksi hyvissä ajoin ennen viimeistä valmistumisajankohtaa. Projektin toteuttaminen lähti suhteellisen tiukan aikataulun takia nopeasti liikkeelle, eikä missään vaiheessa tullut pitkää taukoa. Aikaa kului lopulta vajaa kaksi kuukautta.

Web-sivujen suunnitteluun ei käytetty merkittävästi aikaa, mutta kaikki tärkeimmät asiat kirjoitettiin ylös, jotta sivustojen pohjarakenne olisi tiedossa. Web-sivuja ei voi, eikä kannata suunnitella liian pitkälle, koska toteutusvaihe on suurilta osin myös suunnittelua. Tämä projekti onnistui hyvin suunnittelemalla pohjat tarkasti ja sen jälkeen toteutettiin sivut pala kerrallaan.

Wordpressin käyttötaidot kehittyivät tämän projektin parissa entisestään. Palvelimen hakemistoissa seikkaileminen oli täysin uusi kokemus, enkä aluksi tajunnutkaan kuinka tärkeä osa palvelimen tiedostoihin käsiksi pääseminen oli. Aikaisemmin Wordpressin parissa työskennellessäni en ollut koskaan päässyt tutkimaan palvelimen Wordpress-tiedostoja ja se rajoitti työskentelyä todella paljon. Aivan työn alussa en edes tiennyt kuinka pääsisin kirjautumaan palvelimelle, koska FTP-tunnukset olivat jossain palveluntarjoajan web-hotellin hallintapaneelissa piilossa. Myös kirjautumiseen käytettävä ohjelma oli mysteeri, puhumattakaan isäntäsivun osoitteesta (Kuva 4.). Lopulta tuntien yrittämisen ja tiedon etsimisen jälkeen onnistuin, jolloin alun tuskastuminen jäi kerralla taakse.

Myös CSS-taidot ottivat suuren askeleen eteenpäin. Aikaisemmin olin vain ko-keillut erilaisia tyyllisääntöjä onnistumisen toivossa, mutta tämän projektin ansios-
ta perehdyin myös teoriaan, joka auttoi ymmärtämään CSS:ää paremmin. Ulko-
asun muokkaaminen nopeutui huomattavasti projektin edetessä, kun ei tarvinnut
miettiä, mitä mikäkin kohta tekee.

PHP ei ollut kovin suuressa roolissa, mutta siihenkin perehdyin jonkin verran kir-
jallisuuden avulla. Olisin jopa hieman toivonut, että PHP:n kanssa olisi pitänyt
olla enemmän tekemisissä. Nyt se jäi vain satunnaiseen koodin muokkaamiseen ja
sen sijoittamiseen keskelle valmista PHP-koodia.

Henkilökohtaisesti olen todella tyytyväinen sivujen rakenteeseen ja ulkonäköön
(Liite 1, 2 ja 3). Wordpress on kuitenkin pääosin harrastesivujen ylläpitoon tar-
koitettu sisällönhallintaohjelma, joten siinä valossa lievästi rajoittuneella ohjel-
malla tähän lopputulokseen pääseminen tuntuu hyvältä. Tietysti sivut voisivat olla
paljon paremmatkin, mutta koska minulla ei ole kokemusta varsinaisesta ammatti-
laismaisesta web-sivujen suunnittelusta tai ohjelmistoista, voidaan vain kysyä,
mitä olisin saanut aikaan paremmalla kokemuksella ja kunnan työkaluilla. Projek-
tin aikana tutkin monia eri web-sivuja, joita jotkin web-sivuja työkseni tekevät
yritykset olivat toteuttaneet. Totesin, etteivät ne edes olleet, ainakaan asettelultaan
tai ulkonäöltään, kovinkaan paljoa edellä. Monesti sivut oli tehty niin sekaviksi,
ettei niillä tehnyt edes mieli vieraillla.

Tulevaisuudennäkymät eivät näiden kolmen web-sivuston osalta ole tiedossa. Si-
vustot ovat nyt netissä kaikkien nähtävillä, mutta saattaa olla, että toimeksiantaja
haluaa vielä jossain vaiheessa muuttaa joitain asioita. Se, että olenko itse mukana
tekemässä noita mahdollisia muutoksia, ei ole vielä ajankohtainen kysymys. Olen
kyllä valmis muokkaamaan omia luomuksiani tulevaisuudessakin, jos aikaa vain
riittää.

Tämä projekti oli todella mukava ja mielenkiintoinen. Koulusta valmistumisen
jälkeen melkein toivoisin, että pääsisin suunnittelemaan ja toteuttamaan web-
sivuja myös työkseni. Varsinkin ulkonäön muokkaaminen on aina ollut mielen-
kiintoista.

Olen kiitollinen, että sain mahdollisuuden tehdä tämän opinnäytetyön, koska olin jo yhdessä vaiheessa todella epätoivoinen, kun sopivaa aihetta ei löytynyt mistään. Olisi muutenkin ollut epämukavaa työstää jotain, josta ei todellisuudessa ole kiinnostunut tai tietotaitoa ei löydy riittävästi. Nyt kuitenkin pääsin tekemään sitä, missä olin jo valmiiksi hyvä ja tietoakin oli tarpeeksi. Lisätiedon hankkiminenkaan ei tuntunut ollenkaan epämukavalta, kun aihe oikeasti kiinnosti.

LÄHDELUETTELO

/1/ Contact & Travel Finland Oy. Yritys. Viitattu 25.3.2013.

<http://www.ctf.fi/web/yritys.html/>

/2/ Wordpress About. Viitattu 29.3.2013. <http://wordpress.org/about/>

/3/ Castro E. 2007. Kotisivut kuntoon – HTML, XHTML ja CSS. 1.painos. Jyväskylä. Readme.fi.

/4/ Korpela K. J. 2003. CSS-tyylit. 1. painos. Porvoo. Docendo.

/5/ Rantala A. 2005. Web-ohjelmointi. 1. painos. Porvoo. Docendo.

/6/ Häggman C. 2001. Web-Desing. 1. painos. Jyväskylä. Docendo.

/7/ Responsive WordPress Theme. Viitattu 15.4.2013.

<http://themeid.com/responsive-theme/>

/8/ Wordpress support: child themes. Viitattu 16.4.2013.

http://codex.wordpress.org/Child_Themes

/9/ Österlun P. 2010. Kotisivujen suunnittelijan opas. Boston Information Group.

Viitattu 5.4.2013. <http://www.kotisivubisnes.com/p/e-kirjat.html>

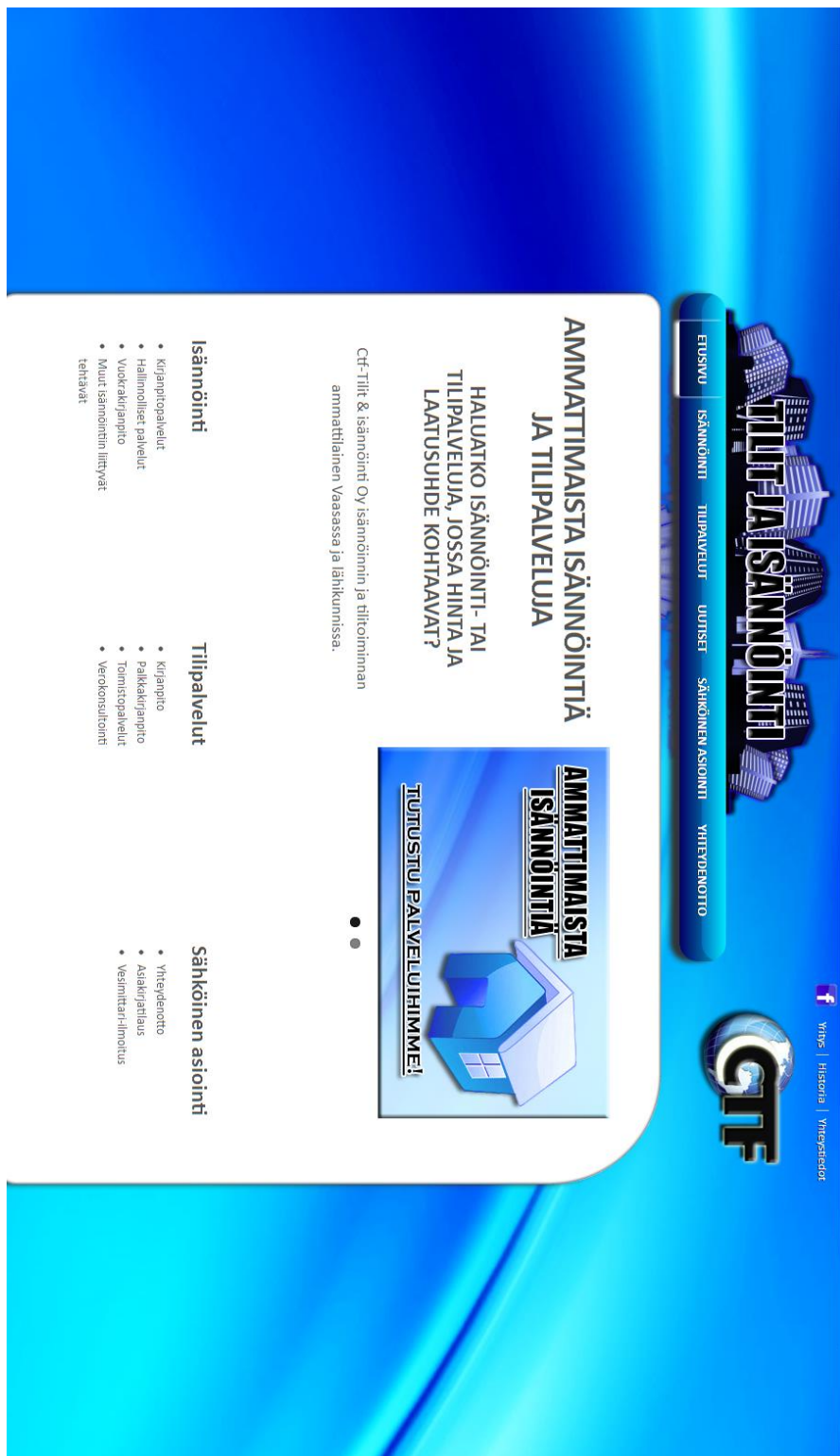
/10/ Software Development Resources. Content management system (CMS). Vii-

tattu 10.5.2013. http://docforge.com/wiki/Content_management_system

Ensimmäisen web-sivuston lähes valmis etusivu



Toisen web-sivuston lähes valmis etusivu



Kolmannen web-sivuston sisältöä vaille valmis etusivu

