

Sonja Rask

Kotisivujen teko Weebly-työkalun avulla

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinöörityö

17.5.2018

Tekijä Otsikko	Sonja Rask Kotisivujen teko Weebly-työkalun avulla
Sivumäärä Aika	23 sivua 17.5.2018
Tutkinto	insinööri (AMK)
Koulutusohjelma	tietotekniikka
Suuntautumisvaihtoehto	ohjelmistotekniikka
Ohjaaja	lehtori Sami Sainio
<p>Insinööriyön tarkoituksena oli toteuttaa espoolaiselle yritykselle selkeät kotisivut, joiden tuli lisäksi olla helposti asiakkaan muokattavissa. Kotisivut on jo noin 70 prosentilla yrityksiä, ja ne ovat oleellinen osa yrityksen yleiskuva. Verkkosivuilla tarjotaan usein tietoa yrityksen palveluista, jotta asiakkaat voivat tutustua yrityksen toimintaan yrityksen kannalta kustannustehokkaasti.</p> <p>Työssä tutustuttiin HTML-merkintäkieleen, CSS-tyyliohjeisiin ja JavaScript-tekniikoihin, jotka ovat yleensä internetsivujen ulkoasun ja toiminnallisuuden taustalla. Verkkosivut rakentuvat elementeistä, ja HTML:ää käytetään määrittämään elementtien paikat. Tyyliohjeet antavat selaimelle ohjeet siitä, miten verkkosivut näytetään näytöllä. JavaScript-kieli on tarkoitettu tuomaan interaktiivisuutta verkkosivuille, kuten ponnahdusikkunoita, joihin voi reagoida.</p> <p>Yritys oli tilannut verkkotunnuksen ja webhotellipalvelun Louhi.fi-sivustolta. Kotisivut päätettiin toteuttaa webhotellipalvelun tarjoamalla graafisella käyttöliittymällä, Weeblyllä.</p> <p>Työn tuloksena syntyi asiakkaan tilaamat kotisivut, joita on helppo muokata kotisivutyökalulla Louhen webhotellissa. Sivut sisältävät yleistä tietoa yrityksen toiminnasta, yhteystiedot tärkeimpiin henkilöihin, osoitteen ja kartan sekä yleisimpien palveluiden kuvaukset.</p>	
Avainsanat	kotisivut, internetsivut, Weebly, css, html, julkaisujärjestelmä

Author Title	Sonja Rask Creating a website with Weebly
Number of Pages Date	23 pages 17.5.2018
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation Option	Software Engineering
Instructor	Sami Sainio, Senior Lecturer
<p>The client of this thesis was a company from Espoo, GCE Service. They didn't have webpages for their services, so the objective was to create simple ones. The client's request was to make the webpages easy to edit, so they could make changes to them. About 70% of businesses have websites, so it is major part of the company.</p> <p>In this thesis we get to know to Hypertext Markup Language, Cascading Style Sheets and JavaScript, that are techniques behind websites. Websites are built with elements and HTML is used to determine their places in the webpage. Stylesheets give the browser instructions how to show the page on the display. JavaScript is meant to give websites interactivity, such as popup-windows that you can react to.</p> <p>The company had ordered a domain and web hotel service from Louhi.fi. The website was decided to create with the graphic content management system, Weebly, offered by Louhi.</p> <p>The result of this thesis is a website for the client, with the possibility to easily edit them in the web hotel. The webpages contain information about the company's services, contact information, address and a map.</p>	
Keywords	website, homepage, html, css, Weebly, cms, content management system

Sisällys

Lyhenteet

1	Johdanto	1
2	Tekniikka kotisivujen ulkoasun takana	1
2.1	HTML-kuvauskieli	1
2.2	CSS-tyyliohjeet	3
2.3	JavaScript ja interaktiivisuus	5
2.4	HTML:n ja CSS:n historiaa	5
3	Palvelin ja kotisivutyökalu	8
3.1	Kotisivutyökalujen historiaa	8
3.2	Julkaisujärjestelmien tietoturva	10
3.3	Louhi palvelimena	11
3.4	Weebly-sisällönhallintajärjestelmä	12
4	Kotisivujen toteutus	13
5	Pohdinta	16
	Lähteet	17

Lyhenteet

HTML	Hypertext Markup Language. Hypertekstin merkintäkieli. Koodikieli, jolla tehdään kotisivuja.
CSS	Cascading Style Sheets. Tyylisäännöstö, jonka avulla muokataan kotisivujen ulkoasua.
W3C	World Wide Web Consortium. Kansainvälinen yhteisö, joka kehittää internetin standardeja.

1 Johdanto

Insinööriyön tavoitteena on tehdä toimivat ja selkeät verkkosivut espoolaiselle General Construction Equipment Servicelle. Yritys on valinnut valmiiksi oman domainin, eli verkkotunnuksen, gceservice.fi. Verkkotunnus-ta ylläpitää Louhi (Louhi.fi), jonka kautta yritys on etukäteen tilannut webhotellipalvelun. Tämän työn tarkoituksena on tehdä verkkosivut, joita yrityksen on mahdollisimman helppo itse tarvittaessa muokata ja ylläpitää. Kaikki päätökset sivujen sisällöstä ja ulkoasusta tehdään yhdessä yrityksen johdon ja taloushallinnon kanssa.

Yritys sijaitsee Espoossa ja on perustettu vuonna 2014. GCE Service tarjoaa asiakkailleen maansiirtokoneiden huolto- ja korjauspalveluita. Toimeksiantajalla ei ole ollut aikaisemmin kotisivuja, jotka ovat nykyään tärkeitä yrityksille. Lähes kaikilla yrityksillä on nykyään verkkosivut, joissa on vähintäänkin yrityksen yhteystiedot.

2 Tekniikka kotisivujen ulkoasun takana

2.1 HTML-kuvauskieli

Lähes kaikki internetsivut on tehty käyttäen HTML-kieltä. HTML eli Hypertext Markup Language, suomeksi hypertekstin merkintäkieli, on kuvauskieli, jolla rakennetaan internetsivuja. Sen avulla määritellään, missä eri elementit sijaitsevat sivuilla, esimerkiksi missä ovat kuvien tai otsikoiden paikat. Selaimet, esimerkiksi Google Chrome tai Mozilla Firefox, lukevat ja kääntävät HTML-koodit internetsivuiksi näytölle. (1, s. 4.)

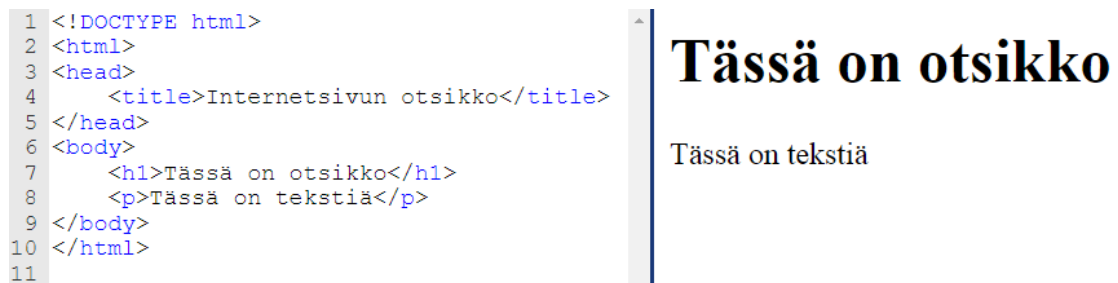
HTML-koodissa tärkeintä ovat elementit. Kaikki HTML-sivut alkavat alkutagilla `<html>` ja päättyvät lopputagiin `</html>`. HTML-elementti on juurielementti, joka tarkoittaa sitä, että kaikki muut elementit HTML-koodin sisällä ovat näiden tagien sisäpuolella. (1, s. 12; 2, s. 70.)

HTML-elementin sisällä on aina kaksi osaa: HEAD-elementti ja BODY-elementti. HEAD-elementti, eli otsikko-osa, alkaa alkutagilla `<head>` ja loppuu lopputagiin `</head>`. Sen sisälle kirjoitetaan metatietoja, kuten sivun otsikko tai kuvaus, tyylitietoja eli CSS-koodia

ja skriptejä, joita on esimerkiksi ulkoinen JavaScript-koodi. Otsikko-osan elementit eivät ole sivujen sisältöä, vaan kuvailevat sivustoa tietyillä tavoilla. (1, s. 12; 2, s. 70; 3.)

BODY-elementti, eli runko-osa, alkaa heti HEAD-elementin jälkeen ja loppuu juuri ennen HTML-elementin lopputagia. Niin kuin aiemmatkin esitellyt elementit, myös tämä elementti alkaa <body>-tagilla ja loppuu </body>-tagiin. Tämän elementin sisälle tulee internetsivun sisältö. (1, s.12; 2, s. 70.)

Kuvan 1 esimerkissä näkyy, miten nämä kolme elementtiä toimivat. Kuvasta huomaa, että title-elementin sisällä oleva otsikko ei näy itse sivulla, vaan sen näkee esimerkiksi selaimessa välilehdellä. Esimerkissä ei ole annettu mitään tyylitietoja, joten käytössä ovat selaimen oletusarvot h1-elementin tyylille ja p-elementin tyylille. (4, s. 2.)



Kuva 1. HTML-koodin yksinkertainen esimerkki.

HTML-dokumenttien elementit, joita käytetään internetsivuston varsinaisen sisällön esittämiseksi, jaetaan kahteen ryhmään: lohkoelementteihin ja sisäelementteihin. Lohkoelementit ovat suurempia rakennekokonaisuuksia, lohkoja, joista sivun runko yleensä muodostetaan. Näitä ovat esimerkiksi otsikkoelementit (h1, h2), kappaleet (p) ja listat (ul, ol). Sisäelementit ovat pienempiä kokonaisuuksia, joita ovat esimerkiksi kuvat ja lomakkeiden kentät. (2, s. 73.)

Elementteihin liittyy määritteitä, attribuutteja, jotka kirjoitetaan elementin alkutagin sisään käyttäen välilyöntiä erottamaan ne elementin nimestä. Jotkin määritteet ovat pakollisia. Esimerkiksi kuva-elementissä (img) on pakollisena määrite sille, mistä kuva haetaan näytölle, mikäli halutaan, että kuva on näkyvissä. Tällainen kuva-elementti määritteineen olisi esimerkiksi muotoa . Määritteiden arvot ovat aina tekstiä, eli ne eivät voi sisältää tageja. (2, s. 73–74)

Määritteillä voidaan myös luokitella elementtejä tai antaa niille tunnisteita. Luokitellakseen elementin on sille annettava määrite class. Luokan nimeksi voi antaa ihan millaisen nimen tekijä haluaa, mieluiten aina jokin sana jokin kuvaa luokkaa parhaiten, esimerkiksi muodossa `<h2 class="varoitus">`. Kun elementille on asetettu luokka, sitä voidaan käyttää tyylisäännöstoissä tai JavaScript-koodissa. Tällöin haluttu tyyliohje tai interaktiivinen tapahtuma tapahtuu vain elementeille, joilla on kyseinen luokka. Tunniste annetaan elementille muodossa `<h2 id="varoitus">`, ja se on paljon yksilöllisempi kuin luokka. (2, s. 76.)

HTML-elementtejä, erityisesti lohkoelementtejä, voidaan myös ryhmitellä ja muodostaa suurempia kokonaisuuksia. Näistä ryhmittelyistä ei ole hyötyä yksinään, mutta kun otetaan käyttöön CSS-muotoilut, niitä voidaan hyödyntää. (1, s. 12.) Ryhmittely tehdään joko div-tagilla tai span-tagilla. Div-tagilla voidaan auttaa elementtien sijoittelussa tai taustaväriässä. Yleensä sitä käytetään esimerkiksi palstoittamaan sivua, koska se on ikään kuin suorakaiteen muotoinen. Span-tagii toimii suhteellisen samalla tavalla, mutta se ei tee rivitystä, eli sitä ei voi käyttää palstoittamiseen. (1, s. 15.)

2.2 CSS-tyyliohjeet

Cascading Style Sheets, suomennettuna kaskadisit tyyliohjeet, on merkintätapa, jonka avulla määritellään verkkosivujen ulkoasua. Ohjeita kutsutaan myös tyylisäännöstoiksi. (2, s. 300.) Se antaa verkkosivun avaavalle selaimelle ohjeet siihen, miten verkkosivujen elementit näytetään näytöllä. Tyyliohjeita ovat muun muassa värit, koot ja vaikkapa reu-
nuksien paksuudet. (7.)

CSS-määrytykset voidaan liittää kahdella eri tavalla HTML-sivuille. Yksi tapa on kirjoittaa erillinen CSS-tiedosto, joka liitetään HTML-sivun HEAD-elementtiin. Tällainen CSS-tiedosto mahdollistaa verkkosivujen yhtenäisen ulkoasun, jolloin yhtä määrytystä muuttamalla kaikkien sivujen ulkoasu muuttuu. Tämä on helppo tapa pitää verkkosivujen ulkoasu yhtenäisenä. Toinen tapa liittää määrytyksen sivuille on kirjoittaa ne suoraan HTML-koodiin. (6.)

Tyyliohjeet koostuvat valitsimista, attribuuteista ja attribuuttien arvoista. Valitsimelle annetaan mahdollisimman kuvaava nimi, yleensä käytetään kielenä englantia. Valitsin määrää sen elementin, jota tyyliohje koskee. Tämän jälkeen valitaan attribuutti eli omi-

naisuus valitussa elementissä, jota halutaan muokata, esimerkiksi väri eli color. Sen jälkeen annetaan attribuutille arvo, eli jokin väri. Värejä voi kirjoittaa kirjaimin tai voidaan käyttää esimerkiksi heksadesimaaliarvoja, jolloin voidaan valita suuremmasta valikoi-
masta värejä. Attribuutti erotetaan arvosta aina kaksoispisteellä, ja arvon jälkeen pääte-
tään koodi puolipisteellä. (2, s. 302.)

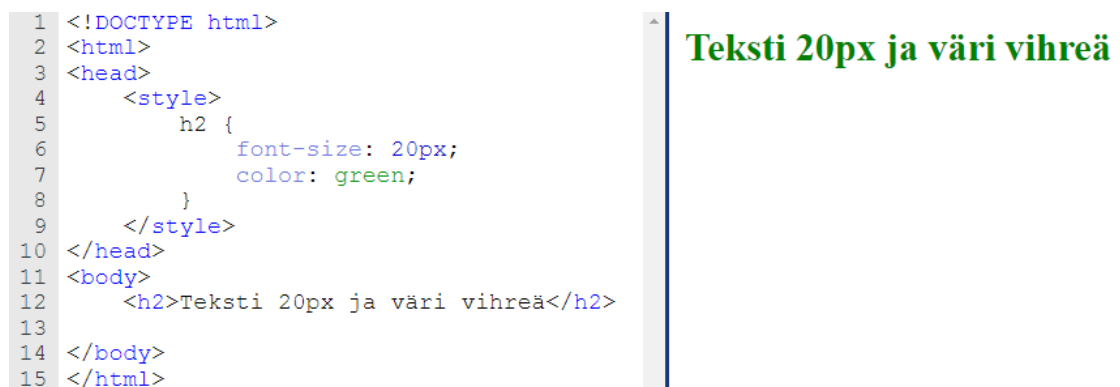
Jos valitsimen elementti on luokiteltu johonkin luokkaan, silloin valitsimen eteen laitetaan piste (.). Tällöin vain elementit, jotka on luokiteltu kyseiseen luokkaan, saavat sen oh-
jeen. Jos taas valitsimen elementille on annettu tunniste, sitä voidaan yksilöllisesti muo-
kata tyyliohjeella, kunhan valitsimen eteen laitetaan ristikkomerkki (#). (2, s. 302.)

Esimerkkikoodin 1 tyyliohjeen mukaan näytölle ilmestyy otsikkoelementti h2, jonka kir-
jaisin-koko on 20 pikseliä ja jonka väri on vihreä.

```
valitsin {                                h2 {
                                           font-size: 20px;
                                           color: green;
}                                           }
```

Esimerkkikoodi 1. Tyyliohjeiden perusta ja esimerkki.

Kuvassa 2 on havainnollistettu yksinkertainen esimerkki sekä HTML:stä että sen esi-
merkkikoodin 1 mukaisesta tyyliohjeesta.



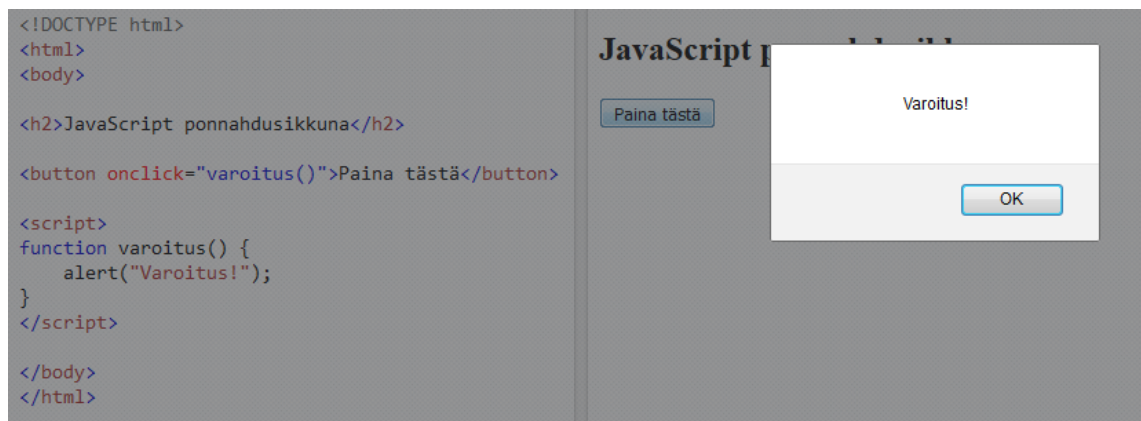
Kuva 2. Esimerkkikoodin 1 tulos.

2.3 JavaScript ja interaktiivisuus

HTML-dokumenttiin voi myös liittää JavaScript-koodia, jolla tehdään interaktiivisia toimintoja verkkosivuille. Interaktiivinen toiminto tarkoittaa sellaista toimintoa, joka vastaa käyttäjän toimintaan. Esimerkiksi ponnahdusikkunat, joissa on jokin kuittaustoiminto, tai painikkeet, joita painaessa jotain tapahtuu sivulla, voidaan tehdä JavaScriptillä. (7.) Nimensä mukaisesti JavaScript on skriptikieli, mikä tarkoittaa, että koodia ei käännetä kääntäjällä ohjelmaksi kuten sovelluksia, vaan selain tulkitsee skriptin ja toteuttaa sen näytölle. (8.)

JavaScriptin voi kirjoittaa suoraan HTML-dokumenttiin `<script>`- ja `</script>`-tagien väliin. Nämä tagit sijoitetaan joko HEAD-elementin sisään tai BODY-elementin sisään. CSS-tiedoston tavoin myös JavaScript-tiedosto voidaan liittää verkkosivulle ulkoisesta lähteestä. Tällöin script-tagiin lisätään lähde määrite `<script src="javaskriptini.js"></script>`. (9.)

Kuvassa 3 on pieni esimerkki JavaScript-koodia. Tässä esimerkissä on tehty painike, jota painamalla näytölle ilmestyy ponnahdusikkuna, jossa lukee varoitusteksti. Ponnahdusikkunassa on myös kuittauspainike, jolla ikkunan saa pois.

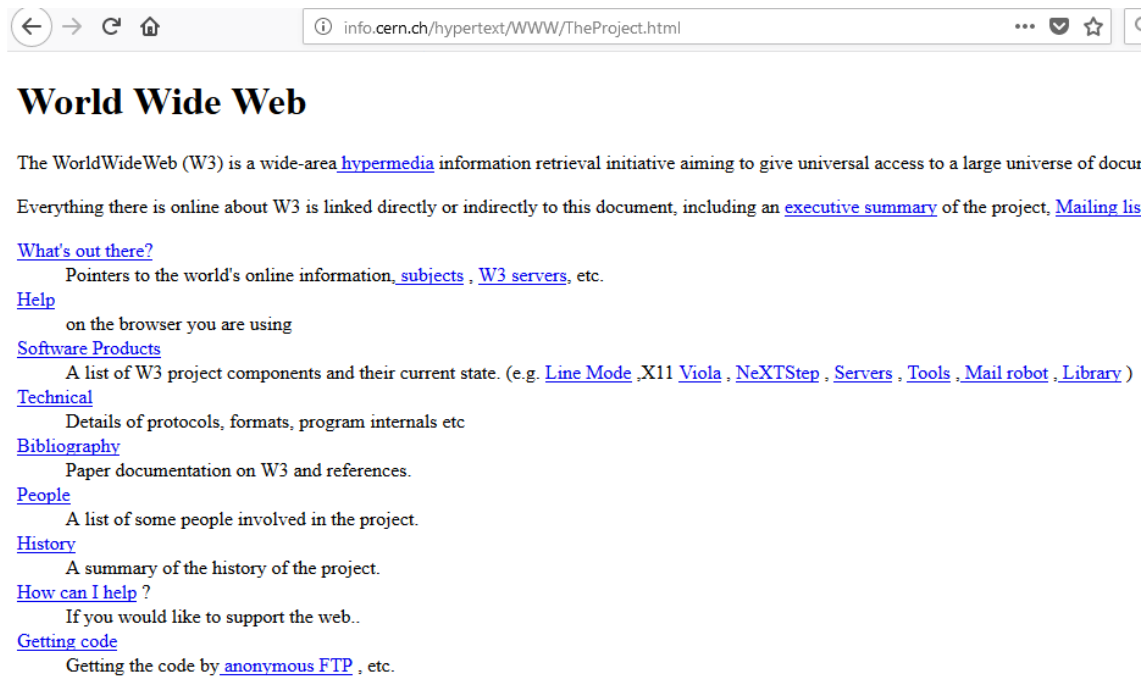


Kuva 3. Esimerkki ponnahdusikkunasta.

2.4 HTML:n ja CSS:n historiaa

HTML:n historia alkoi vuonna 1991, kun Tim Berners-Lee kehitti ensimmäisen version HTML:stä, HTML 1.0:n. Hänen ajatuksensa oli, että eri tietokoneilla nähtävillä oleva tieto

olisi kaikkien saatavilla. Ensimmäisessä versiossa oli kaksikymmentäkaksi elementtiä, joista kolmesta on edelleen käytössä HTML 4.0:ssa. HTML 1.0:n jälkeen julkaistiin HTML 2.0. Näissä kahdessa ei ollut paljoa eroa, vain muutama uusi toiminnallisuus. (10; 11.) Kuvassa 4 näkyy ensimmäinen internetsivu, jonka Tim Berners-Lee teki.



Kuva 4. Tim Berners-Leen tekemä ensimmäinen internetsivu (12).

Netscape oli ensimmäinen selain, jossa HTML toimi. Kun markkinoille alkoi tulla uusia selaimia, varsinkin Internet Explorer, internetsivut eivät näyttäneetkään yhtään samalta kuin Netscapessa. Tämän takia alettiin kehittää HTML 3.0:aa. Uuteen HTML:ään haluttiin lisätä paljon uusia ominaisuuksia websivujen suunnittelijoille. Selaimet olivat tuolloin hitaampia käsittelemään suurempia tietomääriä, joten suurimmasta osasta uusista ominaisuuksista luovuttiin, ja lopulta koko HTML 3.0:n suunnittelu jäädettiin. (10.)

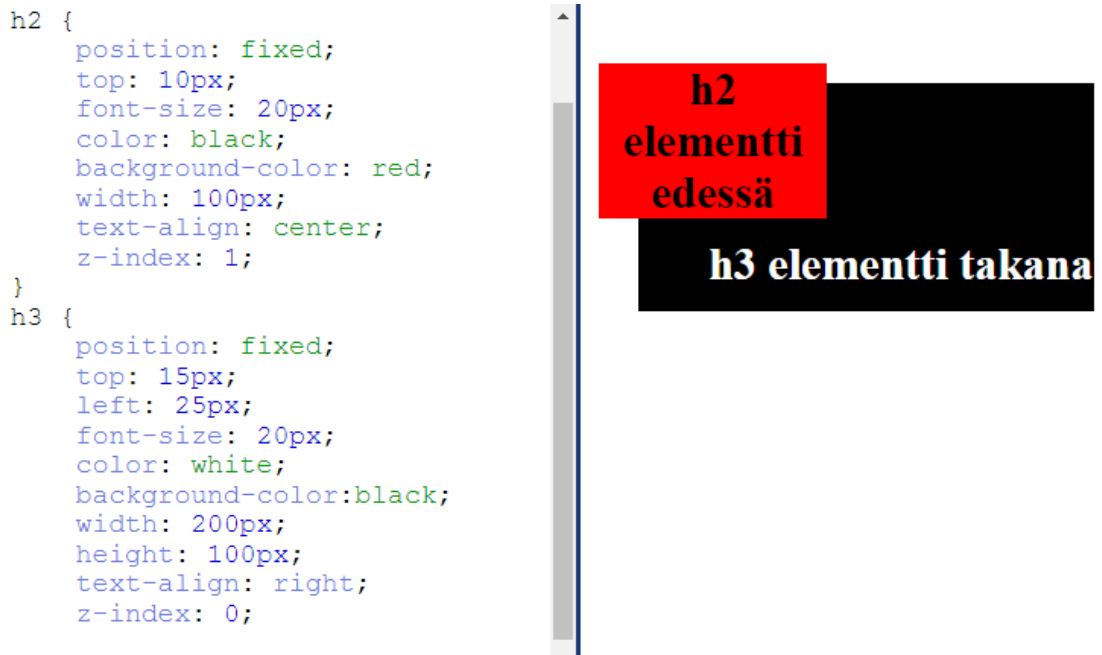
Selaimilla alkoi olla omia tageja sivujen sommitteluun ja ulkoasuun, ja niiden määrä kasvoi kasvamistaan, joten vuonna 1994 World Wide Web Consortium, W3C, kehitti standardin websivujen merkintäkielelle. Tämän standardin nimi oli WILBUR, mutta se kuitenkin myöhemmin muutettiin HTML 3.2:ksi. Netscape-selaimen omat tagit eivät päässeet mukaan tähän uuteen standardiin. Vuonna 1997 HTML 3.0:sta tuli virallinen standardi, ja nykyään jokainen selain tukee tätä versiota. (13.)

HTML 4.0 oli seuraava iso askel kohti nykyistä hypertekstin merkintäkieltä. Joulukuussa 1997, alle vuosi HTML 3.2:n virallistaminen jälkeen, W3C ehdotti uudeksi standardiksi HTML 4.0:aa. Tämä versio oli hyvin erilainen kuin HTML 3.0. Vanhemmassa versiossa kaikki tyyli- ja värit koodattiin suoraan HTML-dokumenttiin, mutta nyt uudessa HTML:n versiossa käytettiin CSS:ää. HTML 4.0:aan kuului muun muassa uudet tagit tyyliohjeille, skripteille ja taulukoille sekä paremmat mahdollisuudet esittää tietoa ihmisille, jotka eivät esimerkiksi näe hyvin. HTML 4.0:sta tuli virallisesti standardi vuonna 1998. (10.)

Ensimmäisen luonnoksen CSS:tä kehitti norjalainen Håkon Wium Lie vuonna 1994. Tyyliohjeiden kehitys eteni hitaasti mutta varmasti ja vuoden 1996 lopulla Microsoft päätti tukea CSS:ää ja W3C virallisti CSS:n standardiksi websivujen tyyliohjeiksi. (14.)

Ensimmäinen CSS-versio, eli CSS1, ei tukenut mitään hienouksia, mihin useimmat käyttäjät ovat varmasti websivuilla tottuneet, kuten esimerkiksi varjostuksia. Tällöin kaikki toteutettiin kuvina, jos sivulla oli mitään, mikä oli koristeellista. (14.)

Kun CSS2 kehitettiin, alkoi viimein tulla enemmän ominaisuuksia eikä kaikkea tarvinnut toteuttaa kuvina. Kun CSS1:ssä asemoitiin sivuja taulukoiden avulla, uudella CSS2:lla saatiin aseteltua websivut paljon helpommin position-ominaisuudella. Tämän lisäksi saatiin myös käyttöön z-index, jonka avulla voidaan määritellä elementtien sijoitus syvyyssuunnassa. Tämä tarkoittaa, että jos elementillä A on suurempi z-index kuin elementillä B, elementti A on niin sanotusti lähempänä käyttäjää ja elementti B jää A:n taakse. (15.) Tämä on havainnollistettu kuvassa 5.



Kuva 5. Elementti h2 punaisella pohjalla on nostettu elementin h3 eteen syvyysuunnassa käyttäen z-indeksiä.

Nykypäivän CSS on CSS3, joka on syntynyt 2000-luvulla. Uusimpia ominaisuuksia on muun muassa selainten yhteensopivuus. Tällöin CSS-koodiin kirjoitetaan etuliite, jonka avulla tyyliohje toimii oikein kyseisellä selaimella. Muita uusia ominaisuuksia, joita ei aiemmissa versioissa ole ollut, ovat esimerkiksi reunukset (borders), joille voi muotoilla pyöreät kulmat ja erilaiset animaatiot. (16.)

3 Palvelin ja kotisivutyökalu

3.1 Kotisivutyökalujen historiaa

Kotisivutyökaluja kutsutaan myös julkaisujärjestelmiksi tai sisällönhallintajärjestelmiksi. Näiden kahden järjestelmän ero on yleensä niillä toteutettavien projektien laajuus. Kun julkaisujärjestelmien projektien budjetit ovat hintaluokiltaan noin 5 000 – 80 000 euroa, sisällönhallintajärjestelmien budjetit yltyvät jopa 500 000 euroon asti. Englanniksi nimitys on yleisesti Content Management System, CMS, tai tarkemmin Web Content Management System, eli WCMS. (17.)

Julkaisujärjestelmillä julkaistaan internetiin tietoa kaikkien saataville. Yleisimmin näitä järjestelmiä käytetään juuri verkkosivujen ylläpitoon ja tekoon ilman minkäänlaista

ohjelmointia. Suurimpia julkaisujärjestelmiä ovat muun muassa WordPress, Drupal ja Joomla! (tästä eteenpäin Joomla). (18.)

Ensimmäinen avoimen lähdekoodin julkaisujärjestelmä kehitettiin vuonna 1995, ja se oli nimeltään Wiki-Wiki. Se mahdollisti monipuolisten websivustojen tekemisen pienellä budjetilla. (19, s. 12.)

Nykyisten suurimpien julkaisujärjestelmien kehitys on alkanut 2000-luvun alussa. Joomla-lan historia on alkanut vuonna 2001, ja aluksi se tunnettiin nimellä Mambo. Kun Mambon suosio julkaisujärjestelmänä alkoi kasvaa, alettiin sen kehittämiseen palkata alan parhaimmista. Vuonna 2005 kehittäjien ja Mambon omistaman ohjelmistoyrityksen välille syntyi riitaa siitä, mihin suuntaan Mambon kehitystä pitäisi viedä. Samana vuonna Mambon kehittäjät tekivät oman ryhmän ja käyttivät Mambon avointa koodia luodakseen Joomla-lan. Käytännössä Joomla-lan ensimmäinen versio oli Mambon viimeinen versio parannuksilla. Tammikuussa 2008 julkaistiin Joomla 1.5, joka kehittäjien sanojen mukaan oli ensimmäinen oikea versio Joomla-lasta. (19, s. 14.)

Toisen suurimman ja tunnetuimman julkaisujärjestelmän, WordPressin, historia alkaa vuodesta 2001. Silloin Michel Valdridgi teki pienen julkaisujärjestelmän nimeltä b2 cafelog, joka oli tarkoitettu blogikirjoitusten julkaisemiseen. Kaksi vuotta myöhemmin Matt Mullenweg ja Mike Little löysivät Valdridgin julkaisujärjestelmän ja kehittivät sen pohjalta uudenlaisen blogityökalun, WordPressin. (19, s. 23.)

WordPressin kehitys on ollut nopeaa. Vuonna 2004 tuli WordPressin toinen versio, 1.2. Seuraavan vuoden sisään tuli kaksi versiota lisää, 1.5 ja 2.0. Vuosien varrella on tullut lukuisia uusia versioita, ja WordPressin käyttö on kehittynyt blogijulkaisujärjestelmästä myös www-sivujen julkaisuun. Nyt vuoden 2018 huhtikuussa viimeisin versio on 4.9.5. (19, s. 23; 20.) Taulukossa 1 vertaillaan WordPressiä sen kahteen julkaisujärjestelmäkilpailijaan Joomla-laan ja Drupaliin.

Taulukko 1. Ilmaisten julkaisujärjestelmien vertailu (21).

	WordPress	Joomla	Drupal
Websivuja tehty	75 miljoonaa	2,8 miljoonaa	1,1 miljoonaa
Latauksia	160 miljoonaa	75 miljoonaa	25 miljoonaa
Marketti-osuus	58,4 %	7,2 %	4,8 %
Ilmaisia teemoja	Yli 3 000	Yli 1 000	Yli 2 000
Ilmaisia lisäosia	44 000	5 000	26 000
Kustannukset	Ilmainen	Ilmainen	Ilmainen
Asennusaika	5 minuuttia	10 minuuttia	10 minuuttia
Tunnettuja käyttäjiä	CNN, Mashable, Time	McDonald's, Ikea, Linux, Harvard University	NBC, Patch
Hyviä puolia	Aloittelijaystävällinen, hakukoneoptimoitu, responsiiviset sivustot	Verkkokauppa mahdollisuus, helppokäyttöisyys	Nopea, monikäyttöinen
Huonoja puolia	Altis hyökkäyksille, hidas	Lisäosat ei välttämättä aina yhteen sopivia	Ei aloittelijalle, vaikea saada apua

Taulukon mukaan WordPress on kaikista suosituin julkaisujärjestelmä, ja se on myös hyvä vaihtoehto aloittelijoille. Sen käyttämiseen ei tarvitse välttämättä opetella ohjelmointia, eikä sen ylläpito ole kallista. (21.)

3.2 Julkaisujärjestelmien tietoturva

Internetissä sivustot joutuvat joskus hyökkäysten kohteeksi. Tämä on otettava huomioon verkkosivuja tehdessä. Julkaisujärjestelmien vanhentuneet versiot ovat haavoittuvaisia verkkosivujen murroille. Tästä syystä julkaisujärjestelmät ja niihin mahdollisesti asennetut laajennukset kannattaa aina pitää päivitetynä ajan tasalle. (22.) Monissa julkaisujärjestelmissä on kolmannen osapuolen tekemiä lisäosia, joihin ei välttämättä tule ollenkaan tietoturvapäivityksiä. Sen vuoksi täytyy olla tarkkana, mitä laajennuksia lataa käytettäväksi. (19, s. 22.)

Toinen tärkeä osa verkkosivun tietoturvaa on palvelintason tietoturva. Tällöin ei ole väliä, millä julkaisujärjestelmällä sivusto on rakennettu. Hyvän tietoturvan palvelimella varmistavat esimerkiksi päivitykset, varmuuskopiointi ja palomuuuri. Tässä insinööriyössä käytössä ollut palvelin on turvallinen, koska järjestelmä on ympärivuorokautisessa valvonnassa ja jokainen webhotelli ja verkkosivu on varmuuskopioitu Louhen varmennusjärjestelmään. (23.)

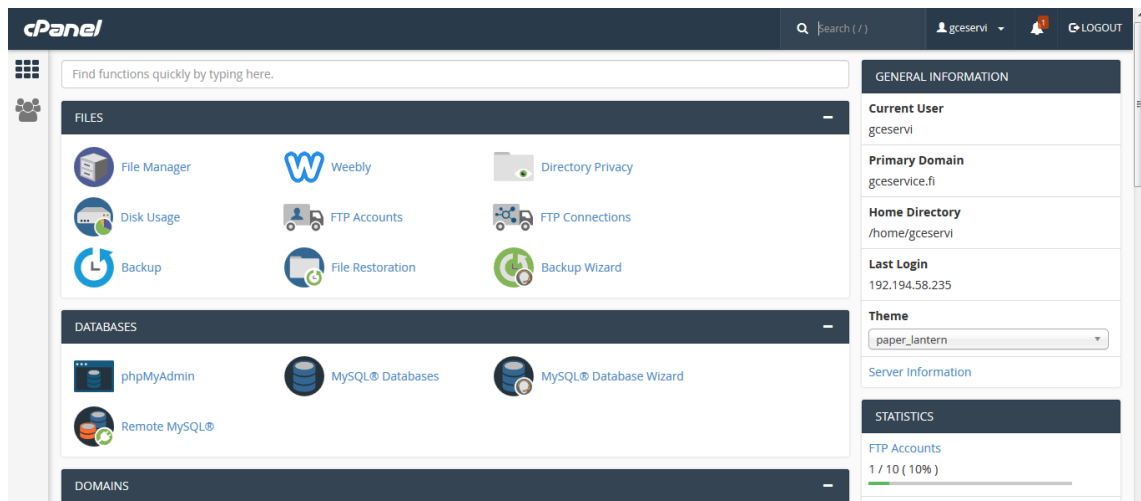
3.3 Louhi palvelimena

Louhi on suomalainen yritys, joka on perustettu vuonna 2003. Se tarjoaa asiakkailleen muun muassa verkkotunnusten rekisteröintiä, webhotelli palvelun ja virtuaalipalvelimia. Työssä tutustuttiin vain webhotelliin, ja GCE Service oli rekisteröinyt verkkotunnuksensa etukäteen Louhen kautta.

Louhella on neljä webhotellipakettia valikoimassaan: Mini, Perus, Rajaton ja Voima. Jokaisessa vaihtelee levytilan määrä ja se, kuinka paljon liikennettä voi kuukaudessa käyttää. Myös sähköpostilaatikoiden määrä vaihtelee. (23.)

GCE Service valitsi Webhotelli Rajaton -sopimuksen, jolla saa verkkosivuille 20 Gt tilaa. Se on suuri määrä muutaman sivun verkkosivuille. Siihen kuuluu myös sähköposti, tietokannat sekä rajattomasti liikennettä. Sen hinnaksi tulee kuukaudessa 11,99 euroa, joka on yrityksen budjettiin sopiva hinta. Yritys halusi tämän webhotellisopimuksen sen hinnan ja rajattomuuden takia.

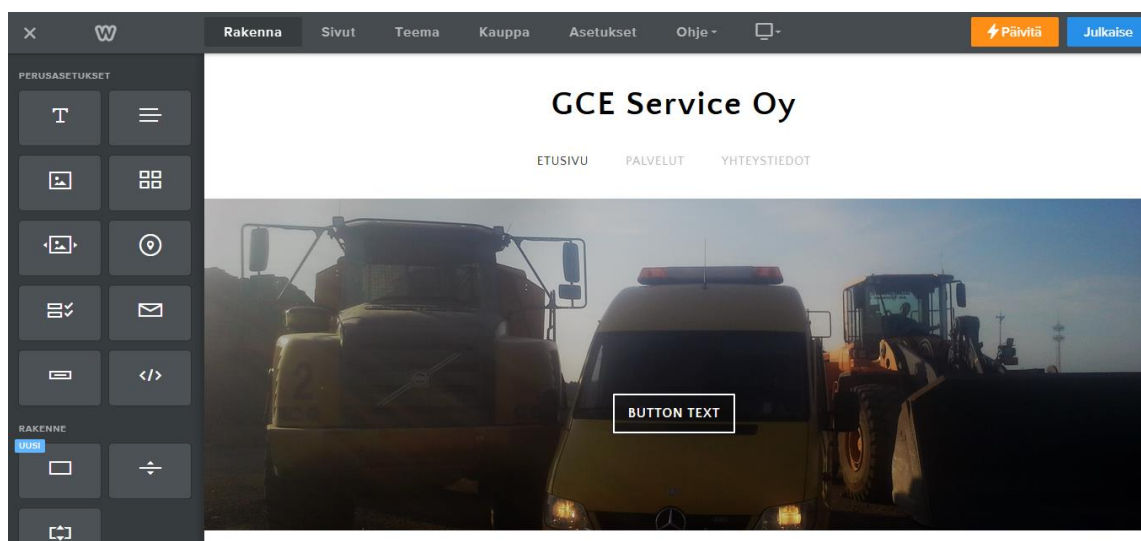
Louhessa kirjaututaan ensin OmaLouhi-sivustolle, jossa voi hallita omia sopimuksia, tilata lisää palveluita ja kirjautua edelleen webhotelliin tai sähköpostiin. Webhotelliin kirjaututaan toisilla tunnuksilla, jotka toimitetaan sähköpostiin. Ensimmäisenä avautuu näkymä Cpanel-sivustolle (kuva 6), jossa voi hallinnoida muun muassa verkkotunnuksia, tietokantoja ja tiedostoja. Niitä ei tässä opinnäytetyössä käydä läpi.



Kuva 6. Cpanel-näkymä, kun kirjaudutaan webhotelliin (24).

3.4 Weebly-sisällönhallintajärjestelmä

Kotisivut tehtiin graafisella kotisivujen tekoon tarkoitettulla käyttöliittymällä nimeltään Weebly. Niiden tekemiseen ei tarvita ollenkaan koodausta, mikä on asiakkaalle eduksi, jotta se voi itse myöhemmin muokata sivuja. Perusajatuksena on, että vasemmalla olevasta valikosta valikoidaan, mitä halutaan sivustolle liittää, ja se raahataan hiirellä toivotuun kohtaan, minkä jälkeen sitä voi vielä muokata klikkaamalla. Jotkin toiminnot kyseisestä valikosta ovat ilmaisia ja jotkin ovat maksullisia. Tätä työtä tehdessä käytössä olivat vain ilmaiset toiminnot, sillä yritys ei tarvitse käyttöönsä esimerkiksi HD-laatuista videota.

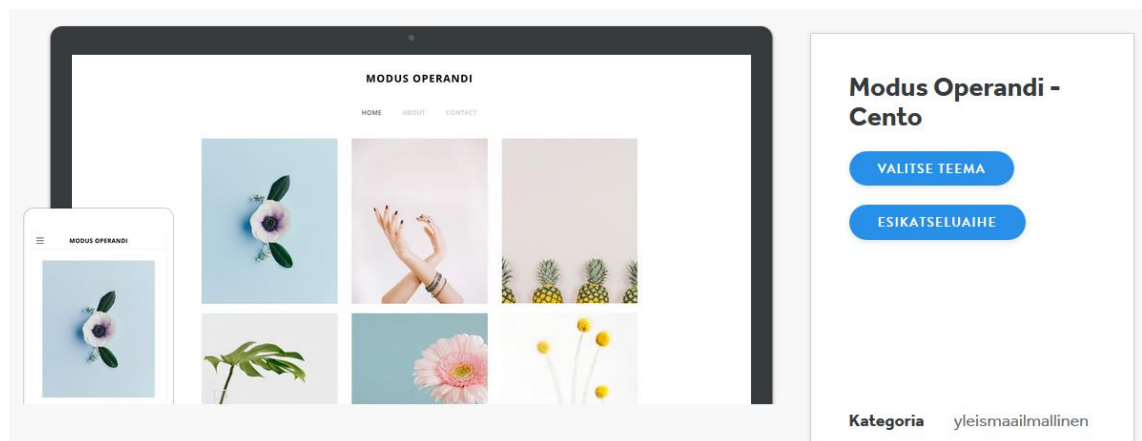


Kuva 7. Weeblyn näkymä, kun muokataan sivustoa (24).

Weeblyn aloitusnäkyssä on viisi välilehteä erilaisille toiminnoille. Nämä toiminnot ovat Rakenna, Sivut, Teema, Kauppa ja Asetukset. Niiden jälkeen samalla rivillä on myös Ohjeet ja painike, josta voi vaihtaa näkymää työpöytänäköymän ja mobiilinäkymän välillä. Oranssi painike sivussa on päivityspainike, jos haluaa maksaa ja päivittää sivustoja suu-remmaksi. Viimeisenä ylhäällä oikealla on sininen painike, Julkaise, jota painamalla sivusto julkaistaan internetiin.

4 Kotisivujen toteutus

Ihan ensimmäiseksi Weeblyyn kirjaututtua käytiin läpi teemoja, joiden avulla saatiin hahmotettua kotisivujen tyyliä. Teemat ovat valmiiksi tehtyjä ulkoasuja, joissa on esimerkiksi oletusvärit ja -rakenteet. Niitä on valmiina Weeblyssä monia sekä erillinen vaihtoehto, josta voi ladata tyylin itse. Valinta kohdistui lopulta teemaan nimeltä ”Modus Operandi” (kuva 8). Tämä teema on yksinkertainen, ja siitä on helppo lähteä kokoamaan internet-sivuja. Taustaväri on valkoinen ja navigaatiopalkin värit ovat harmahtavat, tummemmalla näytetään sen sivun nimi, jolla tällä hetkellä ollaan. Navigaatiolinkin väri vaihtuu tummemmaksi myös silloin, kun hiiren kohdistin on sivun nimen päällä.



Kuva 8. Modus Operandi -teema ennen muokkausta (24).

Teemat valitaan Weeblyn ”Teema”-välilehdellä, josta voi myös muokata sivuston kaikkien tekstien kirjaisintyyliä. Kirjaisintyyliin ei vielä tehty muutoksia.

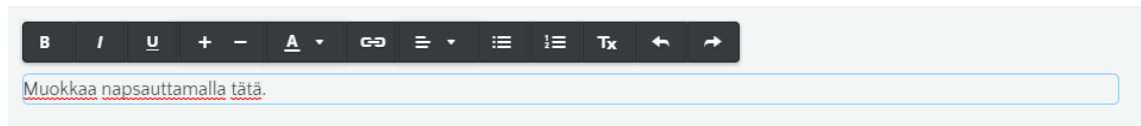
Teeman valitsemisen jälkeen suunniteltiin websivujen sisältö. Yritys toivoi kolme erillistä sivua, joista yhdessä on etusivun lisäksi enemmän tietoa sen palveluista ja toisessa tärkeimmät yhteystiedot. Verkkosivuille lisättiin etusivun lisäksi Palvelut-sivu ja Yhteystiedot-sivu. Sivujen muokkaaminen, lisääminen ja poistaminen onnistuu Weeblyn ”Sivut”-välilehdeltä (kuva 9). Sivun nimeä klikkaamalla avautuvat vasemmalla muokkauspalkissa sivujen muokkaukseen tarkoitetut toiminnot. Täällä on mahdollista muokata sivun nimeä, joka näkyy linkkinä navigaatiopalkissa, poistaa sivu sekä valita, haluaako sivun olevan julkinen vai täytyykö kirjautua sisään. Koska kyseessä on informatiiviset sivut asiakkaille, ovat kaikki sivut julkisia. Tulevaisuudessa yritys saattaa tarvita esimerkiksi työntekijöilleen sivun, johon vain heillä on oikeus kirjautua.



Kuva 9. Sivut-välilehden näkymä (24).

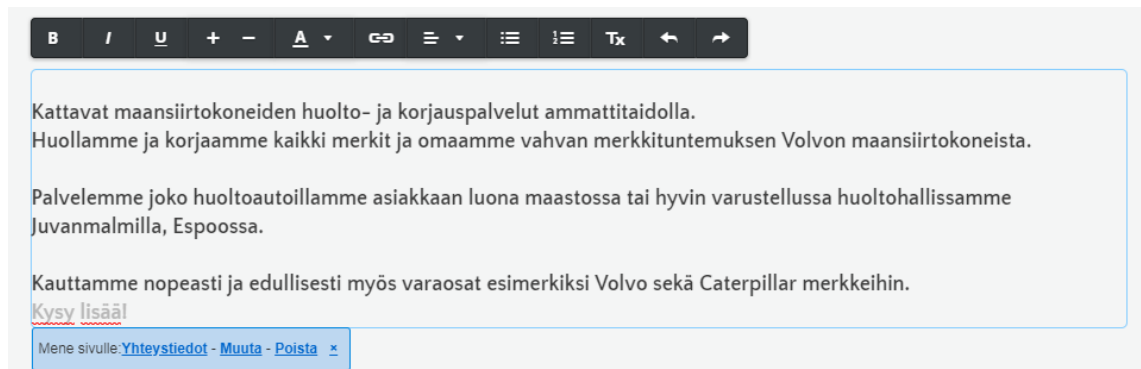
Sivujen luomisen jälkeen jokaiselle sivulle lisättiin otsikkokuva, joka on sivuilla heti navigaatiopalkin alapuolella. Etusivulle laitettiin isompi kuva, jossa on kaksi maansiirtokoneetta ja yksi huoltoauto. Kahdelle muulle sivulle lisättiin hieman pienemmät kuvat, jotta esimerkiksi yhteystiedot-sivulla pääsee heti käsiksi yhteystietoihin. Tämä, ja tekstin lisääminen, onnistuu Weeblyn ”Rakenna”-välilehdeltä.

Seuraavaksi kotisivuille lisättiin tekstiä. Etusivulle kirjoitettiin yleistä tietoa yrityksen toiminnasta ja sijainnista. Etusivulla on myös linkki, jonka tekstinä on 'Kysy lisää!', jota klikkaamalla pääsee suoraan yhteystiedot-sivulle. Teksti laitettiin kotisivulle vasemmasta sivupalkista raahaamalla otsikkokuvan alle. Ennen kirjoittamista tekstiosion yläpuolelle ilmestyi muokkauspalkki (kuva 10), jossa oli paljon toimintoja tekstin muokkaukseen.



Kuva 10. Weeblyn tekstiosio ja muokkauspalkki (24).

Suunniteltu teksti kirjoitettiin tekstiosioon ja linkki lisättiin helposti valitsemalla tekstistä se osa, jonka halutaan ohjaavan toiselle sivulle. Tekstin valitsemisen jälkeen painettiin 'luo linkki'-painiketta muokkauspalkissa, minkä jälkeen avautui uusi palkki (kuva 11), josta voi valita, millaisen linkin haluaa. Vaihtoehtona oli myös vakiosivu, josta pystyi valitsemaan jo luodun sivun. Tähän siis valittiin sivu yhteystiedoille, jotka oli luotu jo aiemmin.



Kuva 11. Linkin lisäämisen jälkeen linkkiä painettaessa näytetään, mille sivulle linkki ohjautuu (24).

Yhteystiedot-sivulle lisättiin yrityksen tärkeimpien henkilöiden nimet, asema yrityksessä ja heidän yhteystietonsa, yrityksen osoite sekä kartta yrityksen sijainnista, kartan avulla saatiin sivulle lisää ilmettä, ja siitä on myös apua asiakkaille. Nämä kaikki lisättiin helposti vasemmalta palkista vetämällä, niin kuin muutkin sivuston osiot. Karttaa lisätessä täytyi vielä klikata karttaa ja kirjoittaa yrityksen osoite sille varattuun paikkaan.

Viimeisenä lisättiin tekstiä Palvelut-sivulle, jossa kerrotaan tarkemmin yrityksen palveluista, joita ovat muun muassa maansiirtokoneiden määräaikaishuollot, hydraulikan ja sähkövianhaut, niiden korjaukset ja säädöt. Tämä tehtiin samalla tavalla kuin muutkin tekstin lisäämiset, vetämällä teksti osio sivulle vasemmasta palkista.

Lopuksi vielä muokattiin sivuston otsikon kirjaisinta. Oletus kirjaisinmalli oli nimeltään Aller, mutta se oli liian pelkistetty ja pieni fontti, joten kirjaisin vaihdettiin Quattrocento Sansiksi. joka on enemmän asiakasyrityksen näköinen.

5 Pohdinta

Insinööriyö onnistui suhteellisen hyvin. Taustaosuuden kirjoittaminen oli itselleni hieman haastavaa, mutta onnistuin siinä lopulta. Tavoitteet saavutettiin, kun internetsivujen lopputulos miellytti toimeksiantajaa. Opin myös itse paljon uutta, joten oppimistavoitteet saavutettiin.

Kotisivujen teko on mielenkiintoista, ja olen opiskellut enemmän HTML-koodia ja CSS-tyylejä. Päätin lisätä tietoa myös JavaScriptistä työhön, koska mielestäni se on tärkeä osa nykyaikaisia kotisivuja. HTML- ja CSS-koodista löytyi internetistä todella paljon tietoa.

Sivuja muokatessa huomasin, että joidenkin ominaisuuksien muokkaaminen ei olekaan niin helppoa. Esimerkiksi kun yritin muuttaa navigaatiopalkin linkkien väriä, sen muokkaamiseen ei löytynyt mistään mitään vaihtoehtoa. Ainoa paikka, josta nämä nimenomaiset värit olisi voinut vaihtaa, oli Teema-välilehdellä oleva ”Muokkaa HTML/CSS” -painike. Sieltä löysin koodia, josta tämän olisi voinut vaihtaa, mutta päätimme yrityksen kanssa, että en muuta sieltä mitään, mitä he eivät osaisi itse muuttaa tarvittaessa takaisin. Myöhemmin huomasin myös, että minkään tekstin väriä ei voinut muuttaa kuin CSS-koodia muokkaamalla. Onneksi olimme yhdessä jo valinneet hyvän teeman, jossa oli yrityksen tarpeisiin hyvät värit, ja yritys oli tyytyväinen lopputulokseen.

Julkaisujärjestelmät ovat edistäneet websivujen luontia. Ne ovat paljon helpompi tapa tehdä verkkosivut jopa ilman tietotaitoa HTML:stä tai muista tekniikoista. Olen itse tehnyt verkkosivut pelkästään ohjelmoimalla HTML:ää ja CCS:ää. Tällöin joutuu tekemään kaiken itse alusta asti, ja se on huomattavasti aikaa vievää, joidenkin ominaisuuksien tekeminen on hankalaa ja joutuu monesti pettymään, kun sivu ei toimikaan halutulla tavalla. Alusta asti koodaaminen on myös tietoturvallisuudelle riski, ellei todella tiedä, mitä tekee. Tehdessä tietenkin oppii uutta, mutta silti suosittelen käyttämään julkaisujärjestelmää. Mikäli koodaaminen on tuttua ja haluaa toteuttaa uutta, yhtenä vaihtoehtona voi toteuttaa uusia lisäosia julkaisujärjestelmiin.

Lähteet

- 1 Sarja, Jari. 2012. HTML:n perusteet. Verkkoaineisto. <http://opinnot.internetix.fi/fi/muikku2materiaalit/muut/amatillinen/web/html/html_perusteet.pdf>. Luettu 20.3.2018.
- 2 Korpela, Jukka K & Linjama, Tero. 2005. Web-suunnittelu. Jyväskylä. Docendo Finland.
- 3 Html-sivujen metatietoa. Verkkoaineisto. Netikka. <<http://www.netikka.net/herala/meta.htm>>. Luettu 18.3.2018.
- 4 Korpela, Jukka K. 2008. CSS verkkosivujen muotoilussa. Jyväskylä. WSOYpro/Docendo-tuotteet.
- 5 Linkola, Jussi. WWW-julkaisemisen perusteet – CSS. Verkkoaineisto. <<http://jml.kapsi.fi/jussi/www-julkaisemisen-perusteet/www-julkaiseminen-css/>>. Luettu 16.3.2018.
- 6 HTML ja CSS opas. Verkkoaineisto. Salakapakka. <http://salakapakka.net/opaat/html-ja-css-opas/css_opas.php>. Luettu 15.3.2018.
- 7 Korpela, Jukka K. 2009. JavaScript (ja vastaavat). Verkkoaineisto. <<http://jorpela.fi/webjulk/3.2.html>>. Luettu 21.3.2018.
- 8 Saarikumpu, Osmo. JavaScript-kielen alkeet osa1. Verkkoaineisto. <<https://wepipakki.com/js/opas/alkeet1.htm>>. Luettu 21.3.2018.
- 9 JavaScript Where to. Verkkoaineisto. W3Schools. <https://www.w3schools.com/js/js_where.asp>. Luettu 21.3.2018.
- 10 History of HTML. Verkkoaineisto. Land of Code. <<http://www.landofcode.com/html-tutorials/html-history.php>>. Luettu 9.4.2018.
- 11 Bartels, Angela. 2011. Internet History: HTML Code Evolution 1.0 to 5.0. Verkkoaineisto. <<https://blog.rackspace.com/internet-history-html-evolution>>. Luettu 9.4.2018.
- 12 Berners-Lee, Tim. World Wide Web. Verkkoaineisto. <<http://info.cern.ch/hypertext/WWW/TheProject.html>>. Luettu 9.4.2018.
- 13 Shannon, Ross. 2012. The History of HTML. Verkkoaineisto. <<http://www.yourhtmlsource.com/starthere/historyofhtml.html>>. Luettu 9.4.2018.
- 14 Forsström, Mikko. 2016. CSS:n lyhyt historia. Verkkoaineisto. <<https://fraktio.fi/blogi/cssn-lyhyt-historia/>>. Luettu 9.4.2018.

- 15 Ekonoja, Antti; Lahtonen, Tommi & Mäntylä, Jukka. 2009. Asemointi. Verkkoaineisto. <<http://appro.mit.jyu.fi/doc/css2/index2.html>>. Luettu 9.4.2018.
- 16 CSS3 ominaisuudet. 2014. Verkkoaineisto. Jyväskylän yliopisto. <<https://koppa.jyu.fi/avoimet/hum/tvt/www/css/css3-ominaisuudet>>. Luettu 9.4.2018.
- 17 Tolvanen, Perttu. 2009. Käsitesekamelskaa: julkaisujärjestelmä, CMS, portaali ja sisällönhallintajärjestelmä. Verkkoaineisto. <<https://vierityspalkki.fi/2009/11/03/kasitesekamelskaa-julkaisujarjestelma-cms-portaali-sisallonhallintajarjestelma/>>. Luettu 12.4.2018.
- 18 CMS eli Content Management System. Verkkoaineisto. Omnipartners. <<https://omnipartners.fi/sanakirja/cms-eli-content-management-system/>>. Luettu 12.4.2018.
- 19 Rautiainen, Teemu. 2012. Julkaisujärjestelmät www-kehityksessä. Opinnäytetyö. Jyväskylän ammattikorkeakoulu. Theseus-tietokanta.
- 20 Download WordPress. Verkkoaineisto. WordPress. <<https://wordpress.org/download/>>. Luettu 12.4.2018.
- 21 Spencer, Jamie. Best content management system 2017 – WordPress VS Joomla VS Drupal. Verkkoaineisto. <<https://makeawebsitehub.com/content-management-system-cms-comparison/>>. Luettu 14.4.2018.
- 22 Julkaisujärjestelmien päivittäminen on tärkeää. 2017. Verkkoaineisto. Viestintävirasto. <<https://www.viestintavirasto.fi/kyberturvallisuus/tietoturvanyt/2017/02/ttn201702061225.html>>. Luettu 14.4.2018.
- 23 Webhotellit. Verkkoaineisto. Louhi. <<https://www.louhi.fi/webhotelli>>. Luettu 21.3.2018.
- 24 Oma Louhi. Verkkoaineisto. Louhi. <<https://oma.louhi.fi/main/>>. Luettu 21.3.2018.