

---

**UUDISTETTAVIEN VERKKOSIVUJEN TOTEUTUS COSS  
RY:LLE**



Ammattikorkeakoulun opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

Visamäki, Syksy 2015

Mikko Sivula



Visamäki  
Tietojenkäsittelyn koulutusohjelma  
Multimedia & e-learning

---

<b>Tekijä</b>	Mikko Sivula	<b>Vuosi</b> 2015
<b>Työn nimi</b>	Uudistettavien verkkosivujen toteutus COSS ry:lle	

---

## TIIVISTELMÄ

Opinnäytetyön toimeksiantajana toimi COSS ry, Suomen avoimien tietojärjestelmien keskus. Työn tarkoituksena oli suunnitella ja toteuttaa uudistetut verkkosivut COSS ry:lle. Työskentelyssä otettiin huomioon nykyaikaiset standardit ja trendit. Verkkosivut suunniteltiin yhdessä henkilön kanssa, joka loi yhdistykselle uuden visuaalisen ilmeen.

Työn teoriaosuudessa käytiin läpi työssä käytetyt menetelmät, ohjelmointikielet ja verkkosivujen suunnittelu, joita sovellettiin käytännön osuudessa. Käytännön osuus käsitteli sivuston suunnittelun ja toteutuksen ja kuinka projekti eteni. Työn tietolähteinä käytettiin suurimmaksi osaksi Internetiä, mutta myös alan kirjallisuutta. Opinnäytetyö toteutettiin kehittämistyönä.

Työn lopputuloksena oli täysin responsiivinen verkkosivusto COSS ry:lle, WordPress-sisällönhallintajärjestelmää käyttäen. Sivusto suunniteltiin ja ohjelmoitiin tyhjältä pohjalta ja siinä otettiin huomioon nykyaikaiset trendit ilman, että sivuston käytettävyys kärsi. Työssä käytettiin hyväksi Bootstrap-sovelluskehystä, jonka avulla sivujen kehitykseen voitiin laittaa enemmän aikaa. Työssä on myös huomioitu Odoo-järjestelmä, joka oli alkuperäinen vaihtoehto sisällönhallintajärjestelmäksi.

**Avainsanat** verkkosivut, WordPress, Bootstrap, Web-ohjelmointi

**Sivut** 36 s.

Visamäki  
Business information technology  
Multimedia & e-learning

---

<b>Author</b>	Mikko Sivula	<b>Year</b> 2015
<b>Subject of Bachelor's thesis</b>	Implementation of website Renewal for COSS Association	

---

## ABSTRACT

The client of this thesis was COSS Association, which is the Centre for Open Systems and Solutions. The object of this work was to design and develop a renewed Web-site for COSS Association. Modern standards and trends were taken into account in the work. The website was designed in co-operation with the person who created the association's new visual identity.

The theory part of this thesis contained procedures used in the work, programming languages and website designing that were implemented in the practical part. The practical part contained designing and developing the website as well as how the project proceeded. The Internet was mainly used as an information source, but also literature from the field was used in the work. The thesis was carried out by using development work.

The result of the work is a fully responsive Web-site for COSS Association by using the WordPress Content Management System. The site was designed and developed from scratch while taking the newest trends in consideration, without affecting the site's usability. Bootstrap-framework was used into the work, which allowed more time to be spent on development. Odo ERP was also taken into consideration in the work, which was the original choice for the content management system.

**Keywords** website, WordPress, Bootstrap, Web-development

**Pages** 36 p.

# SISÄLLYS

1	JOHDANTO.....	1
2	TOIMEKSIANTAJA JA LÄHTÖKOHDAT.....	2
2.1	COSS ry.....	2
2.2	Lähtökohdat ja tavoitteet.....	2
3	OHJELMOINTIKIELET.....	3
3.1	Web-ohjelmointi.....	3
3.2	HTML.....	3
3.3	HTML5.....	4
3.4	CSS.....	5
3.5	CSS3.....	6
3.6	PHP.....	6
3.7	JavaScript.....	7
4	VERKKOSIVUJEN SUUNNITTELU.....	8
5	SISÄLLÖNHALLINTAJÄRJESTELMÄT.....	10
5.1	Odoo.....	10
5.2	WordPress.....	10
5.2.1	Teemat.....	11
5.2.2	Templatet.....	11
5.2.3	Teeman luonti.....	12
5.2.4	Vimpaimet.....	12
6	BOOTSTRAP.....	13
6.1	Asennus ja käyttö.....	13
6.2	Grid System eli ruudukkojärjestelmä.....	14
6.3	CSS.....	15
6.4	Komponentit.....	15
6.5	JavaScript.....	16
7	SIVUSTON TOTEUTUKSEN VAIHEET.....	17
7.1	Toteutuksen aloitus.....	17
7.2	Sivuston siirto julkiselle Web-palvelimelle.....	17
7.3	Sivuston sisältö.....	18
7.4	Testaus.....	18
7.5	Julkaisu ja jatkokehitys.....	18
8	SIVUSTON SUUNNITTELUPROSESSI.....	19
8.1	Rakenne, visuaalisuus ja typografia.....	19
8.2	Navigointipalkki.....	20
8.3	Slideshow ja sisältöalue.....	21
8.4	Sivupalkit ja alapalkki.....	22
8.5	Uutiskirjeen tilaus.....	23
8.6	Lisäosat.....	23

8.6.1	Yhteydentottolomake .....	23
8.6.2	Kalenteri .....	24
8.6.3	Yoast SEO .....	24
8.6.4	Facebook- ja Twitter-syöte .....	24
8.6.5	MailPoet .....	25
9	SIVUSTON TEKNINEN TOTEUTUS .....	26
9.1	Functions.php .....	26
9.2	Header ja navigointipalkki .....	26
9.3	Slideshow .....	27
9.4	Pääsisältöalueet .....	28
9.5	Sivupalkit ja alapalkki .....	30
9.6	CSS .....	32
9.7	Sivukartta ja sivunumerot .....	32
10	YHTEENVETO .....	33
	LÄHTEET .....	34

Uudistettavien verkkosivujen toteutus COSS ry:lle

## KÄSITELUETTELO

### **Bootstrap**

Ilmainen työkalu Web-sivujen ja Web-applikaatioiden luontiin.

### **CDN** (Content Delivery Network)

Sisältöverkko, joka jakaa tiedostoja käyttäjille palvelimelta.

### **CMS** (Sisällönhallintajärjestelmä)

Järjestelmä, jonka avulla on mahdollista hallita sisältöä.

### **CSS** (Cascading Style Sheets)

Tyyliohjeet, joita käytetään erityisesti Web-dokumenttien muotoiluun.

### **DOM** (Document Object Model)

Dokumenttioliomalli, joka kuvaa dokumenttien loogisen rakenteen.

### **ERP** (Enterprise Resource Planning)

Yrityksen toiminnanohjausjärjestelmä.

### **HTML** (Hypertext Markup Language)

Merkintäkieli, jonka avulla luodaan Web-dokumentteja eli internetsivuja.

### **JavaScript**

Oliopohjainen Web-ohjelmointikieli.

### **jQuery**

Ilmainen avoimen lähdekoodin JavaScript-kirjasto.

### **Layout**

Tekstin, kuvien ja elementtien sommitelma.

### **MySQL**

Relaatiotietokantaohjelmisto, jota käytetään Web-ohjelmoinnissa.

### **Notepad++**

Lähdekoodieditori Windowsille.

### **PHP** (Hypertext Processor)

Ohjelmointikieli, jonka avulla voidaan luoda dynaamisia Web-sivuja sekä palvelinympäristökoodia.

### **Responsiivinen**

Optimaalinen esim. WWW-sivun näkymä, ilman että tarvitsee suurentaa kuvaa eri päätelaitteilla.

### **Selektori**

Kaava, jonka avulla voidaan valita elementti tai elementit joita halutaan muotoilla.

### **Tagi**

Elementti, jota ympäröi pienempi kuin ja suurempi kuin merkit.

Uudistettavien verkkosivujen toteutus COSS ry:lle

**Template**

Esimuotoiltu mallinne.

**Vimpaimet (Widgets)**

WordPressin vimpainten avulla voidaan lisätä sisältöä ja ominaisuuksia sivupalkkeihin.

**WinSCP**

FTP-asiakasohjelma tiedostojensiirrolle.

**XAMPP**

Apache-palvelinympäristö, jonka avulla voidaan testata ja kehittää verkkosivuja.

## 1 JOHDANTO

Nykyaikana verkkosivut ovat yksi tärkeimmistä markkinoinnin ja näkyvyyden osa-alueista yrityksille ja yhdistyksille. Hakukoneiden algoritmien jatkuva kehittyminen ja nykyaikaisten sivujen suosimisen takia on erittäin tärkeää, että sivut pysyvät ajan tasalla niin sisällön kuin ulkoasun kannalta.

Tämän opinnäytetyön aiheena on uudistettavien verkkosivujen toteutus COSS ry:lle, joka toimii toimeksiantajana opinnäytetyössä. COSS ry on Suomen avoimien tietojärjestelmien keskus ja toimii Tampereella. Opinnäytetyössä toteutetaan uudistetut verkkosivut COSS ry:lle käyttäen nykyajan standardeja ja uusimpia teknologioita. Yhdistyksen tämän hetkiset sivut ovat epäselvät ja täynnä sisältöä. Sivustolle mentäessä ei välttämättä edes selviä heti, mikä sivusto on kyseessä ja mitä he tekevät. Sivusta olisi tarkoitus saada selkeämmät ja nykyajan vaatimukset täyttävät. Toimeksiantajan toiveena oli, että sivut olisivat helposti päivitettävissä ja sisällön lisääminen kävisi helposti ilman ohjelmointikielten osaamista. Alkuperäisen suunnitelman mukaan toimeksiantaja olisi halunnut sivustolle pienimuotoisen verkkokaupan, jossa he voisivat myydä omia tuotteitaan, mutta tämän työstäminen jäi sivuston tulevaan jatkokehitykseen.

Opinnäytetyön käytännön osuudessa verkkosivut luodaan käyttäen viimeisimpien Web-kehityksen teknologioita ja toimeksiantajan valitsemalla sisällönhallintajärjestelmällä. Osana sivuston ulkoasun ja rakenteen suunnittelusta vastasi Heini Suvisuo, jonka kanssa tehtiin yhteistyötä koko opinnäytetyön ajan. Yhteyshenkilönä toimivat aluksi Anna Wilkman ja Timo Väliharju COSS ry:ltä. Anna Wilkmanin tilalle vaihtui projektin aikana pääyhteyshenkilöksi Moona Puha.

Opinnäytetyössä vastaan keskeisiin kysymyksiin, jotka olivat ”Miten sivuista saadaan modernit ja samalla yksinkertaiset?”, ”Mikä sisällönhallintajärjestelmä sopii parhaiten toimeksiantajalle?” ja ”Miten hyvin sivusto vastaa toimeksiantajan tarpeita?”.



## 2 TOIMEKSIANTAJA JA LÄHTÖKOHDAT

### 2.1 COSS ry

COSS ry (Centre for Open Systems and Solutions, suom. Suomen Avoimien Tietojärjestelmien Keskus) on voittoa tavoittelematon yhdistys, joka keskittyy avoimen lähdekoodin, avoimen datan, avoimien rajapintojen sekä avoimien standardien edistämiseksi.

COSS ry on perustettu vuonna 2003 Teknologiakeskus Hermia Oy:n toimesta. Siitä lähtien COSS ry on kehittynyt suomalaisten tietotekniikkayritysten avoimuuden foorumiksi ja itsenäiseksi yhdistykseksi. COSS ry on myös yksi Euroopan vanhimmista ja aktiivisimmista avoimuuden järjestöistä.

COSS ry:n toimintaan kuuluu mm. tukea julkishallintoa tietojärjestelmien avoimuuden edistämässä, löytää ratkaisuja avoimen lähdekoodin käyttöön palveluissa ja yritystoiminnassa, ylläpitää suhteita alan suomalaisten ja kansainvälisten järjestöjen välillä ja jakaa vuosittain Open World Hero -palkinnon. (COSS ry.)

### 2.2 Lähtökohdat ja tavoitteet

Opinnäytetyöprojekti oli tarjolla jobstep.net-sivustolla, jossa haettiin opiskelijoita tekemään COSS ry:n sivujen uudistus. Hakemuksessa haettiin henkilöä tekemään verkkosivujen layoutin suunnittelu, graafinen ohjeisto ja sivuston tekninen toteutus. Tässä projektissa verkkosivujen layoutin suunnittelu tehtiin yhteistyössä Heini Suvisuon kanssa.

Sivuston suunnittelu alkoi tapaamisella COSS ry:n tiloissa Tampereella. Tapaamisessa katselmoitiin tämän hetkinen sivusto ja mietittiin, mitä voisi parantaa, poistaa ja kehittää. Tapaamisessa päädyttiin siihen tulokseen, että sivusto toteutetaan Odoo-järjestelmän website builder -työkalulla, joka myöhemmin vaihdettiin WordPress-sisällönhallintajärjestelmään.

Tavoitteena sivuston tuli olla tämän hetken standardit täyttävät sivustot, mutta samalla myös mahdollisimman yksinkertaiset ja helppokäyttöiset, niin loppukäyttäjälle kuin hallinnoijalle. Sivuston tuli olla näiden lisäksi myös näyttävät.

Sivusto toteutettiin täysin puhtaalta pöydältä alusta saakka, ilman valmista teemaa tai mitään muuta. Projektissa käytettiin ainoastaan Internetistä löytyviä koodinpätkiä, joita myös muokattiin erittäin paljon omaan tarpeeseen soveltuviksi.

### 3 OHJELMOINTIKIELET

Tässä osiossa käydään lävitse yleistä asiaa Web-ohjelmoinnista sekä projektissa käytetyistä ohjelmointikielistä ja menetelmistä.

#### 3.1 Web-ohjelmointi

Web-ohjelmoiminen on Internet- tai intranet-Web-sivustojen tai Web-sivujen kehittämisprosessi. Oli kyse sitten nettikaupasta, blogista, videoiden suoratoistopalvelusta tai mistä tahansa muusta Internet-sovelluksesta, sen on täytynyt luoda Web-ohjelmoija.

Web-ohjelmointi voidaan luokitella kolmeen eri pääryhmään: palvelinpuolen ohjelmoiminen (server-side coding), asiakaspuolen ohjelmointi (client-side coding) ja tietokannat.

Asiakaspuolen ohjelmoinnista puhuttaessa, kun henkilö selaa Internet-sivuja, on hän käyttäjä tai asiakas. Ohjelmat jotka käyttäjän selain suorittaa, ovat asiakaspuolen skriptejä. Tämä tapahtuu siten, että ohjelma pyytää tarvittavat tiedostot toteutuakseen, palvelimelta ja sen jälkeen ajaa ne käyttäjän selaimessa. Tunnettuja asiakaspuolen ohjelmointikieliä ovat mm. JavaScript, Ajax ja Flash.

Palvelinpuolen ohjelmoinnissa skriptit ajetaan Web-palvelimella, kun käyttäjä pyytää selaimessa toimintaa. Palvelin tuottaa dokumentin, yleensä HTML-dokumentin, jonka käyttäjän selain voi lukea. Tunnettuja palvelinpuolen ohjelmointikieliä ovat mm. PHP, Perl, Ruby ja Java.

Sivuston toimiakseen Internetissä, tarvitsee sen olla tietokannassa palvelimella. Tietokannassa sijaitsee kaikki tiedostot, joita tarvitaan sivuston ja sen sovellusten toimimiseen. Tunnettuja avoimenlähdekoodin tietokantajärjestelmiä on mm. MySQL, PostgreSQL ja MariaDB. (Shiotsu 2014.)

#### 3.2 HTML

HTML (HyperText Markup Language) on nettisivun perusrakennuskappale, jonka avulla voidaan luoda ja visuaalisesti esittää nettisivu. HTML määrittää sivun sisällön, mutta ei sen toiminallisuutta. (Mozilla Developers HTML A 2015.)

HTML käyttää ennalta määrättyjä elementtejä määrittämään sisältötyyppejä. Elementit sisältävät kaksi yhteensopivaa tagia. Tagi alkaa pienempi kuin -merkillä ("`<`") ja loppuu suurempi kuin -merkillä ("`>`"). Tagi suljetaan samalla tavalla, kuin se alkaakin, mutta pienempikuin merkin eteen tulee kauttaviiva ("`/`"). Kuvassa 1 esimerkki HTML:n kappaleesta, "`<p>`"-tagista (paragraph).

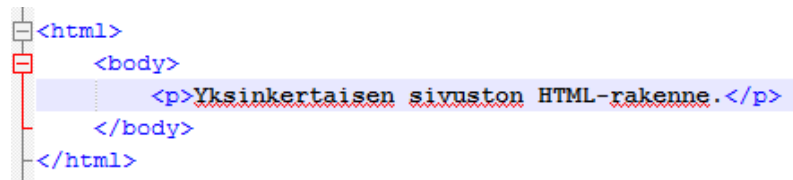
A screenshot of a code editor showing a line of HTML code: `<p>Tämä on kappale. Kappaleet voidaan laittaa p-tagien sisään.</p>`. The code is highlighted in a light blue background.

Kuva 1. Esimerkki HTML:n tagin käytöstä.

## Uudistettavien verkkosivujen toteutus COSS ry:lle

Joillakin elementeillä on erittäin tarkka merkitys, kuten kuvilla, otsikoilla tai listoilla. Jotkin taas ovat vähemmän tarkempia, kuten sisältöalue tai tekstialue, ja joissakin elementeissä on vain sivun tunnistamistietoja, joten niitä ei näytetä. Joka tapauksessa tavalla tai toisella, kaikilla HTML-elementeillä on semanttinen arvo.

Useat elementit voivat pitää sisällään toisia elementtejä, muodostaen hierarkkisen rakenteen. Yksinkertainen sivusto näyttää alla olevan kuvan mukaiselta. Kuvassa 2 on rakennettu yksinkertainen sivurakenne HTML:n avulla.

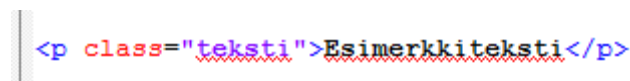


```
<html>
  <body>
    <p>Yksinkertaisen sivuston HTML-rakenne.</p>
  </body>
</html>
```

Kuva 2. Yksinkertaisen sivuston HTML-rakenne.

HTML-dokumentit kirjoitetaan tavallisena tekstinä. HTML:ää voidaan kirjoittaa millä tahansa tekstieditorilla, joka antaa tallentaa sisältöä tavallisena tekstinä (esimerkiksi muistio eli notepad), mutta yleensä suositetaan editoria, joka korostaa syntakseja ja näyttää DOM:it. Tagit voidaan kirjoittaa isoilla kirjaimilla, mutta standardien ylläpitäjä ja kehittäjä, W3C, suosittelee käytettävän pieniä kirjaimia.

Tagit voivat sisältää lisätietoa. Tällaista tietoa kutsutaan attribuutiksi. Attribuutti koostuu yleensä kahdesta osasta, attribuutin nimestä sekä arvosta. Kuvassa 3 näkyy kuinka attribuuttia ja arvoa käytetään.



```
<p class="teksti">Esimerkkiteksti</p>
```

Kuva 3. Esimerkki attribuutin nimestä ja arvosta.

HTML:ssä on mekanismi upotettuihin kommentteihin, joita selain ei renderöi. Kommenttien avulla voidaan jättää tietoa osioista muille kehittäjille tai itselle. Kommentit toteutetaan seuraavalla tavalla:

```
<!-- TÄMÄ ON KOMMENTTI -->
```

HTML-dokumentin on sisällettävä doctype, joka kertoo millä HTML-versiolla sivusto on kirjoitettu. Doctype kirjoitetaan ensimmäiselle riville. HTML5:n doctype ilmoitetaan seuraavalla tavalla:

```
<!DOCTYPE html>
```

Luomalla ja lataamalla nettisivuja Internetiin, niistä tulee osa World Wide Webiä, kun sivut ovat julkaistu. HTML:n avulla kaikki voivat tehdä staattisia sekä dynaamisia nettisivuja. (Mozilla Developers HTML B 2015.)

### 3.3 HTML5

HTML5:n kehitys lähti ajatuksesta, että HTML:ää ja selaimia kehitettäisiin rinnakkain ja, että selainten valmistajat olisivat yhteistyössä. (Korpela

## Uudistettavien verkkosivujen toteutus COSS ry:lle

2011, 24.) HTML5 perustuu HTML:n aiemmille versioille, mutta eroaa silti huomattavasti niistä. HTML5 ei ole vielä valmis, ja joidenkin mukaan sille saadaan virallinen määrittely vasta 2020-luvulla. Yhä useampi selain tukee jo tosin HTML5:n uusia ominaisuuksia, kuten videoiden upottamista ja grafiikan dynaamista luomista. HTML5:n kehityksestä vastaa kaksi organisaatiota, W3C ja WHATWG. (Korpela 2011, 18.)

HTML5:n uusiksi ominaisuuksiksi katsotaan seuraavia ominaisuuksia. (Korpela 2011, 13.)

Taulukko 1. HTML5:n uusia ominaisuuksia

Canvas-elementti	Piirtoalusta, jonka avulla voidaan luoda JavaScriptillä piirtotoimintoja.
Lomakkeet	Uudet lomakkeet ja niiden tietojen tarkistus.
HTML5-muisti	Tietoa voidaan tallentaa paremmin kuin evästeillä
Paikannustiedot	Sisältöä voidaan muuttaa käyttäjän sijainnin mukaan
Upotetut videot	Videoiden näyttäminen suoraan selaimessa
Rakenteiset elementit	Sivuja voidaan jakaa paremmin osiin
Sovellusvälimuisti	Sen avulla selain voi ladata sovelluksen tiedostot offline-käyttöön

HTML5:een siirtyminen ei vaadi mitään toimintoja kehittäjiltä, vaan jo olemassa oleviin sivuihin voidaan alkaa lisäämään HTML5:n ominaisuuksia. Selain tukee, jos tukee HTML5:tä, mutta vaikka ne eivät tukisi sitä, voidaan siihen varautua eri tavoin. (Korpela 2011, 15.)

### 3.4 CSS

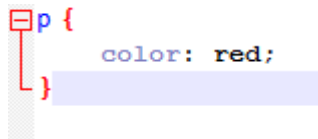
Siinä missä HTML määrittelee vain sisällön rakenteen, toinen suuri Web-teknologia, joka on nimeltään CSS, määrittelee miltä sisältö tulisi näyttää.

CSS, eli Cascading Style Sheet, voidaan kirjoittaa suoraan HTML-dokumenttiin <style> tagien sisään <head> osiossa, mutta normaalisti se kirjoitetaan erilliseen tiedostoon, jotta tyylitietoja voidaan uudelleen käyttää. CSS-tiedoston päätte on ".css". (Mozilla Developers CSS A 2015.)

Ulkoista CSS-tiedostoa käyttämällä saavutetaan monia etuja kuten kaksoiskappaleiden välttäminen, helpompi ylläpito ja se, että voidaan tehdä koko sivustoa koskevia muutoksia yhdessä paikassa.

CSS:ään kirjoitettuja rivejä kutsutaan säännöiksi. Säännöt alkavat selektorilla, jonka avulla valitaan mihin elementtiin sääntö kohdistuu. Selektorilla voidaan valita tagi tai ID- ja class-attribuutti. Selektorin jälkeen tulee ominaisuus ja arvo, jotka erotetaan kaksoispisteellä. Ominaisuuden ja arvon jälkeen tulee puolipiste, joka erottaa sen muista ominaisuus-arvo julistuksista. Ominaisuus ja arvo ympäröidään aaltosulkeilla. Kuvassa 4 CSS esimerkki tag-selektorista. (Mozilla Developers CSS B 2015.)

```
p {  
  color: red;  
}
```



Kuva 4. Esimerkki CSS-säännöstä, joka muuttaa p-tagin tekstin punaiseksi.

Class-attribuutilla voidaan antaa elementille luokkanimi, jonka voi vapaasti valita. Useampi elementti voi saada saman luokka-arvon. Tyyliohjeissa luokan voi valita selektorilla merkkiaamalla pisteen (.) ennen luokan nimeä.

Toisin kuin class-attribuutti ID-attribuutti tulee olla ainutkertainen. CSS-tyyliohjeissa ID voidaan valita selektorilla sijoittamalla risuaita (#) ID:n eteen. (Mozilla Developers Selectors 2015.) Kuvassa 5 näkyy kuinka id- ja class-selektoria käytetään CSS:ssä.

```
#id {  
  color: black;  
}  
  
.class {  
  color: grey;  
}
```



Kuva 5. Ylempänä ID-selektori ja alempana class-selektori CSS-tyyliohjeissa.

### 3.5 CSS3

CSS3 on kehitteillä oleva CSS-tyyliohjekielen uusien versio. Se on taaksepäin yhteensopiva aiempien CSS-versioiden kanssa. Se tuo monta pitkään odotettua uutuutta kuten pyöristettävät kulmat, varjot, liukuvärit, siirtymät tai animaatiot. Kokeilullisia osia pitäisi välttää tai käyttää erityistä varovaisuutta käyttäen, koska molemmat sekä syntaksit että semantiikka voi muuttua tulevaisuudessa. (Mozilla Developers CSS3 2015.)

### 3.6 PHP

PHP on avoimen lähdekoodin palvelinpuolen ohjelmointikieli. Sen voi ladata ilmaiseksi ja käyttää ilmaiseksi. PHP-tiedosto voi sisältää HTML-, CSS-, JavaScript- ja tietysti itse PHP-koodia. PHP-koodi toteutetaan palvelimella ja palautuu selaimelle normaalina HTML:nä. Tiedosto päättyy ".php".

PHP-koodi alkaa "<?php"-merkinällä ja loppuu "?>"-merkinnällä. PHP-tiedosto sisältää yleensä HTML-merkintäkieltä ja itse PHP-koodia. Kuvassa 5 luodaan yksinkertainen PHP-koodi, joka tulostaa sanan "Tervehdys!" selaimen.

```
<!DOCTYPE html>
<html>
  <body>
    <h1>PHP testi</h1>
    <?php
      //Tähän php koodi
      echo "Tervehdys!";
    ?>
  </body>
</html>
```

Kuva 6. Kuva tulostaa tekstin "Tervehdys!" selaimen.

PHP:tä käytetään mm. WordPressissä, Facebookissa ja tuhansissa muissa tunnetuissa Web-palveluissa. (W3Schools PHP.)

### 3.7 JavaScript

JavaScript on maailman suosituin ohjelmointikieli, jonka avulla luodaan selainskriptejä (client-side scripting). Oikealla tavalla HTML-dokumenttiin liitetty JavaScript-ohjelma suoritetaan Web-selaimessa, kun sitä käsketään esittämään HTML-dokumentti. Esimerkkinä JavaScript-ohjelmasta on yksinkertainen ilmoitus, joka ilmestyy ikkunaan ruudulle, kun jotain painiketta klikataan. Yksinkertaisillakin JavaScript-ohjelmilla saadaan aikaiseksi monipuolisia asioita.

JavaScript perustuu tulkintaan, ei kääntämiseen ja ne ovat normaalisti suhteellisen pieniä, jonka takia JavaScript-koodeista yleensä puhutaan skripteinä eikä ohjelmina. (Korpela 2009.)

JavaScript voidaan sijoittaa HTML-dokumentin <body> ja <head> tagien sisään. HTML:n sisään laitettaessa koodia, JavaScript tulee olla <script> ja </script> tagien sisällä.

JavaScriptiä voidaan myös sijoittaa ulkoisiin tiedostoihin. Ulkoiset tiedostot ovat käteviä, kun samaa koodia käytetään monella eri sivulla. JavaScript-tiedoston päätte on ".js". Ulkoisen tiedoston käyttäminen onnistuu määrittelemällä <script> -tagiin lähde, src-attribuutilla, esimerkiksi <script src="skripti.js"></script>. Ulkoisenkin tiedoston voi sijoittaa joko <head> tai <body> tagien sisään. Skripti toimii kuin se olisi sijoitettu, siihen paikkaan mihin ulkoisen tiedoston lähteen lisää. Ulkoisen tiedoston etuja ovat mm. se erottaa HTML:n ja koodin, se tekee HTML:stä ja JavaScriptistä selkeämpää ja helpommin hallittavaa, ja välimuistiin tallennetut JavaScript-tiedostot voivat nopeuttaa sivun latautumista. (W3Schools JavaScript.)

## 4 VERKKOSIVUJEN SUUNNITTELU

Suuren verkkosivun kehitys on prosessi, jolla voi olla kauaskantoisia rahoitushallinnallisia, henkilökunnallisia ja viestinnällisiä seurauksia organisaatiolle, niin kehityksen aikana kuin pitkään sen jälkeenkin. Liian moni verkkosivu toteutetaan ilman täysin ymmärtämättä sivuston tavoitteita. Huonosti suunnitellun ja hätäisesti kehitetty verkkosivusta tulee usein hylätty. (Lynch 2008, 24.)

Ensimmäinen askel verkkosivuston suunnittelussa on tavoitteiden määrittely. Ilman selkeästi esitettyä tehtävää ja tavoitteita projekti ajautuu, menee puihin tai jatkuu sopivan päätepisteen ohitse. Huolellinen suunnittelu ja selkeä tarkoitus ovat avaimia menestykselliseen sivuston rakentamiseen. (Lynch 2008, 1.) Sivuston suunnittelun alkuvaiheessa määritellään myös sivuston sisällön laajuus, interaktiivisuus ja tarvittava teknologia. (Lynch 2008, 30.)

Ensimmäiset luonnokset olisi hyvä piirtää paperille, ja kun luonnoksista alkaa hahmottumaan selkeä kuva, voidaan aloittaa rautalankamallin luominen. Rautalankamallin luominen kannattaa aloittaa usein navigaation, sivupalkkien ja alapalkkien hahmottamisella. Rautalankamallin avulla voidaan usein saada selkeä ymmärrys siitä, minkälainen layoutin tulee olla, kun suunnitellaan osioita muuallekin kuin etusivulle. (Guglieri 2015.)

Sivukartan luominen on myös yksi päätehtävistä sivustoa suunniteltaessa. Sivukartta on kuin uuden toimistorakennuksen pohjapiirustus. Siitä selviää sivuston jokainen sivu ja kuinka ne ovat yhteydessä toisiinsa. Ilman sivukarttaa sivuston luomisesta voi tulla turhauttavaa, sillä sivukartan avulla kaikilla on yhteneväinen suunta. Jos sivukarttaa ei ole, miten voidaan olettaa, että kaikki sisältö, sivut ja ominaisuudet tulevat sivustolle. (Lopuck.)

Sivuston suunnitelman määrittelyn, rautalankamallien ja sivukartan luomisen jälkeen seuraava askel on luoda visuaalinen tyyli. Lopullisen visuaalisen tyylin määrää yleensä organisaation brandi. Visuaalisen tyylin pitäisi olla yhteydessä organisaation muihin verkkokommunikaatioihin. Organisaation brandilla on suuri rooli tässä osassa prosessia, sillä suunnittelijat haluavat visuaalisesti ilmaista avainideoita suunnittelussa.

Suunnitelmien hyväksymisen jälkeen alkaa sivuston kehittäminen. Kehittämisen aikana ohjelmoidaan sivusto, kehitetään uutta sisältöä, parannellaan vanhaa sisältöä ja luodaan uutta mediaa.

Ennen sivuston julkaisua sivusto tulisi testata sisäisen testijoukon kanssa ja niiden kanssa, jotka tietävät sivuston osoitteen. Sivusto tulisi olla tuotantoympäristössä tässä vaiheessa. Testaus on kriittinen osa prosessia, sillä vääjäämättä kohdataan ongelmia, jotka tarvitaan korjata ennen kuin sivusto julkaistaan. Huonosti toimivat sivut, kirjoitusvirheet tai rikkiäiset elementit eivät tee hyvää organisaation brandille. Sivusto tulisi myös testata usealla eri selaimella ja päätelaitteella.

## Uudistettavien verkkosivujen toteutus COSS ry:lle

Testauksen jälkeen koittaa sivuston julkaisu, kun se on hyväksytty. Projekti ei tosin ole ohitse, kun sivusto julkaistaan, vaan sivustosta saadun palautteen perusteella tehdään muutoksia.

Sivuston julkaisun jälkeen alkaa sivuston ylläpitäminen ja huolto. Verkkosivut ovat eläviä kokonaisuuksia ja vaativat jatkuvaa tarkkailua ja huoltoa. Sisällön päivittäminen, muutosten tekeminen koodiin ja rikkiäisten linkkien korjaus on jokapäiväistä työtä. Kaikki nämä ovat kriittisiä vaiheita verkkosivun suunnitteluprosessissa. (Farris 2014.)



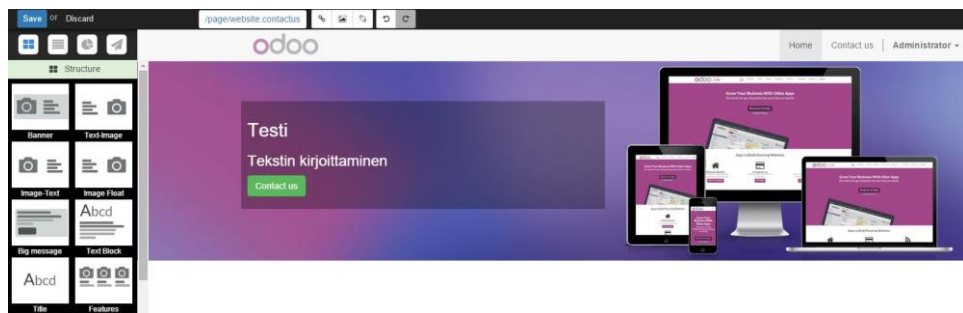
## 5 SISÄLLÖNHALLINTAJÄRJESTELMÄT

Sisällönhallintajärjestelmät ovat Web-pohjaisia sovelluksia, jotka tarjoavat tapoja monelle henkilölle eri käyttöoikeustasoilla hallinnoimaan sisältöä, dataa tai informaatiota. Hallinta viittaa luomiseen, muokkaamiseen, arkistointiin, julkaisemiseen, raportointiin,

Esimerkki sisällönhallintasovelluksesta on Web sovellus, joka tarjoaa mm. ylläpidon hallintapaneelin, toiminnollisuuden, jonka avulla voi luoda, muokata ja julkaista artikkeleja, sivuston koodin päivityksen ja mahdollisuuden tarkastella tilastoja sivulta. (Comentum 2010.)

### 5.1 Odoo

Odoo, entiseltä nimeltään OpenERP, on Odoo S.A.:n kehittämä avoimen lähdekoodin toiminnanohjausjärjestelmä, joka on nopeitten kehittyvä liiketoiminnan ohjelmisto. (Odoo A.) Odoon website builderin avulla, käyttäjät voivat luoda sivuja ”drag and drop” tyyllillä, eli raahaamalla ja tiputtamalla, elementtejä paikoilleen. Odoossa tekstiä voi editoida suoraan riviltä, WYSIWYG-tyyliin (What You See Is What You Get). Järjestelmä käyttää hyväkseen Bootstrap frameworkia. Odooseen on integroitu mm. verkkokauppa, blogi, live chat, newsletter, crm ja tapahtuma lisäosat. (Odoo B.) Kuvassa 7 näkyy Odoon drag & drop tyylinen muokkaustila ja yleisnäkyminen sen käyttöliittymästä.



Kuva 7. Odoon käyttöliittymä

Projektin alkuvaiheilla toimeksiantaja halusi, että sivusto toteutettaisiin Odoon website builder -työkalun avulla. Pitkän tutkiskelun ja kokeilun jälkeen tulimme yhdessä toimeksiantajan kanssa päätökseen, että kyseinen työkalu ei sovi tähän projektiin ja heidän tarkoitukseensa, koska sivusto on sisältö- ja uutispainotteinen. Työkalulla saa aikaan hyvännäköiset sivut, mutta sen toimintojen puutteellisuuden takia päädyttiin WordPress-sisällönhallintajärjestelmään, joka oli jo entuudestaan tuttu toimeksiantajalle.

### 5.2 WordPress

WordPress on ilmainen avoimen lähdekoodin ja tällä hetkellä maailman suosituin julkaisujärjestelmä. Ensimmäinen versio julkaistiin vuonna 2003. WordPress oli alkuun vain blogijärjestelmä, mutta on sittemmin kehittynyt täysimittaiseksi sisällönhallintajärjestelmäksi. Se on rakennettu

## Uudistettavien verkkosivujen toteutus COSS ry:lle

PHP:n ja MySQL:n päälle ja se on rekisteröity GPLv2:n (tai myöhemmän) alle.

WordPressin ydinohjelmiston ovat rakentaneet sadat vapaaehtoiset yhteisön jäsenet ja siihen on saatavilla tuhansia lisäosia ja teemoja, joiden avulla sivustosta voidaan tehdä ihan minkäläinen tahansa. WordPressin on valinnut tällä hetkellä yli 60 miljoonaa ihmistä. (WordPress 2015.)

### 5.2.1 Teemat

Olellisesti WordPressin teemajärjestelmän avulla voidaan sivustolle antaa uusi ”skini”, eli uusi ulkonäkö. WordPressin teemat antavat enemmän valtaa kuin pelkästään uuden ulkonäön luominen. Niiden avulla voidaan hallita ulkonäköä ja sisältö esitetään sivustolla.

WordPress-teema on kokoelma tiedostoja, jotka toimivat yhdessä tuotakseen graafisen käyttöliittymän sivustolle. Näitä tiedostoja kutsutaan template-tiedostoiksi (suom. mallitiedosto) Teemaa muokkaa tapaa, jolla sivustoa esitetään ilman että perustana olevaa ohjelmistoa muokataan. Teemat voivat sisältää muokattuja template-tiedostoja, kuvatiedostoja, tyyliohjeita, muokattuja sivuja, kuten myös tarpeellisia kooditiedostoja (.php).

Uusi teema voidaan lisätä muutamalla eri tavalla; etsimällä ja lataamalla WordPressin ylläpidosta tai lataamalla teematiedostot ulkoisilta nettisivuilta ja siirtämällä ne palveluntarjoajan cPanelista tai käsin FTP-asiakasohjelmalla WordPressin asennuskansioon. Sekä cPanel- että FTP-siirrossa, molemmissa navigoidaan WordPressin asennuskansioon ja sieltä teemat kansioon, johon ladatut teematiedostot siirretään.

Teemojen käyttöönotto onnistuu WordPressin ylläpidosta menemällä ulkonäkö linkin kautta teemat valintaan, josta voi valita haluamansa teeman. (WordPress Codex Themes.)

### 5.2.2 Templatet

Teeman tiedostot ovat modulaarisia, uudelleen käytettäviä mallitiedostoja joiden avulla generoidaan sivuja WordPress-sivustolle. Joitakin teeman mallitiedostoja, kuten header- ja footer-tiedostot, käytetään sivuston jokaisella sivulla, kun taas toisia käytetään vain tietyissä olosuhteissa.

WordPress käyttää query stringiä (muista selittää) päättääkseen mitä mallia tai malleja näytetään sivulla. Query string on tietoa, joka näkyy linkissä kysymysmerkin jälkeen ja voi sisältää useita parametreja. Yksinkertaisesti sanottuna, WordPress tarkastelee mallihierarkiaa, kunnes löytää oikean mallitiedoston.

Käyttäjä voi valita haluaako käyttää tiettyä mallitiedostoa vai ei, poikkeustapauksena index.php. Jos WordPress ei löydä mallitiedostoon sopivaa nimeä, se hyppää seuraavaan tiedostoon hierarkiassa. Jos WordPress ei

## Uudistettavien verkkosivujen toteutus COSS ry:lle

löydä yhtäkään yhteensopivaa mallitiedostoa, käytetään teeman index.php-tiedostoa.(WordPress Developers.)

### 5.2.3 Teeman luonti

Luomalla WordPress-teeman, saadaan sivustosta ainutlaatuisen näköinen ja kaikki irti WordPressin ominaisuuksista. Sillä saadaan myös helpotettua sivuston päivitystä ilman, että sivuston omistajan tarvitsee osata HTML-, CSS- tai PHP-kieltä.

Minimissään WordPress-teema koostuu kahdesta tiedostosta; style.css ja index.php, mutta tyypillisesti template sisältää comments.php, comments-popup.php, footer.php, header.php ja sidebar.php tiedostot. Käyttämällä näitä template-tiedostoja ja sijoittamalla template-tageja index.php tiedostoon, saadaan ne näkymään sivustolla. Kuvassa 8 on esimerkki, jossa näkyy kuinka template-tiedostot saadaan toimimaan, esimerkiksi get\_header()-funktiolla. (WordPress Codex Theme Development.)

```
<?php
/*
   Template Name: Etusivu
 */
?>
<title>Etusivu | Coss Ry</title>
<?php get_header(); ?>
<div class="header">
  <div class="containerwrap">
    <?php get_template_part('carousel', 'bootstrap'); ?>
  </div>
</div>
```

Kuva 8. Esimerkki templatien funktioista

### 5.2.4 Vimpaimet

WordPressin vimpaimien avulla lisäävät sisältöä ja ominaisuuksia sivupalkkeihin. Esimerkiksi oletusvimpaimet, jotka tulevat WordPressin mukana ovat kategoriat, avainsanapilvi, haku jne. Lisäosat usein lisäävät omat vimpaimensa. Vimpaimia voi, ilman ohjelmointitietämystä, lisätä, poistaa ja uudelleen järjestää menemällä WordPressin ylläpidossa ulkoasu valikosta vimpaimet valintaan. (WordPress Codex Widgets. 2015.)

## 6 BOOTSTRAP

Bootstrap, entiseltä nimeltään Twitter Bootstrap, on ilmainen HTML5 ja CSS3 framework (sovelluskehys), joka on suunniteltu auttamaan kehittäjiä sovellusten ja sivustojen kehityksessä. Bootstrapin kehitti Mark Otto ja Jacob Thornton Twitteriltä ja sen ensimmäinen versio julkaistiin elokuussa 2011 GitHubissa. Bootstrap sisältää HTML- ja CSS-pohjaisia suunnittelumalleja usein käytettyihin käyttöliittymä komponentteihin kuten mm. typografia, kaavakkeet, painikkeet, taulut, navigaatiot, alavetovalikot, hälytykset ja monia muita komponentteja, ja myös JavaScript-laajennuksen.

Bootstrapin avulla responsiivisen layoutin luominen käy paljon helpommin kuin ilman sitä.

Bootstrapin suurin etu on, että sen mukana tulee työkaluja joiden avulla voidaan luoda joustavia ja responsiivisia sivu layouteja kuten myös yleisiä käyttöliittymä komponentteja. Lisäksi käyttämällä hyväksi Bootstrapin API:a voidaan luoda edistyneitä käyttöliittymä komponentteja, kuten navigaation reaaliaikaista seuranta, ilman että kirjoitetaan riviäkään JavaScriptiä. Bootstrapin muita etuja on lueteltu taulukossa 2. (Tutorial Republic.)

Taulukko 2. Bootstrapin etuja

Ajan säästö	Aikaa voidaan säästää käyttämällä Bootstrapin valmiiksi määritettyjä malleja ja luokkia, jonka vuoksi voidaan keskittyä muuhun kehitystyöhön.
Responsiiviset ominaisuudet	Käyttämällä Bootstrapia, voidaan luoda helposti responsiivisia malleja. Bootstrapin responsiivisuus tekee sivustoista oikean kokoiset eri laitteilla ja eri resoluutioilla, ilman että tarvitsee muuttaa merkintöjä.
Yhtenäinen malli	Kaikki Bootstrapin komponentit jakavat saman muotoilun ja tyylittelyn käyttämällä keskitettyä kirjastoa, jotta muotoilut ja layoutit sivustoilla seuraavat yhtenäistä mallia läpi kehityksen.
Helppo käyttää	Bootstrapia on erittäin helppo käyttää. Kuka tahansa, jolla on perustietotaso HTML:stä ja CSS:stä voi alkaa kehittämään Bootstrapin avulla sivuja.
Yhteensopiva selainten kanssa	Bootstrap on luotu nykyaikaiset selaimet mielessä ja se on yhteensopiva kaikkien nykyaikaisten selainten kanssa kuten Mozilla Firefox, Google Chrome, Safari, Internet Explorer ja Opera.
Avoin lähdekoodi	Bootstrap on täysin ilmainen ladata ja käyttää.

### 6.1 Asennus ja käyttö

Bootstrapin voi asentaa monella eri tavalla, mutta tässä työssä asennetaan se liittämällä koodiin CDN:stä (Content Delivery Network).

## Uudistettavien verkkosivujen toteutus COSS ry:lle

Käyttämällä MaxCDN nimistä palvelua, voidaan Bootstrap helposti lisätä sivustolle. CDN:n avulla Bootstrapin käyttämisestä tulee nopeampaa loppukäyttäjille, sillä Bootstrap tallentuu välimuistiin ja se latautuu sieltä. Bootstrap ladataan lähimmältä loppukäyttäjää sijaitsevalta palvelimelta, mikä myös johtaa nopeampaan latausaikaan.

Bootstrapin CSS ja JavaScriptit liitetään sivustoon samalla tavalla kuin muut CSS- ja JavaScript-tiedostot. Kuvassa 8 näkyy kuinka Bootstrap otetaan käyttöön sivustolla ja kuvassa 9 on esimerkki kuinka Bootstrap saadaan käyttöön sivustolla ja kuvassa 10 esimerkki Bootstrapin avulla luodusta yksinkertaisesta sivusta.

```
<title>Bootstrap - Testi</title>

<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">

<!-- Valinnainen -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap-theme.min.css">

<!-- Compiled ja minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>
```

Kuva 9. Bootstrapin käyttöönotto sivustolla.

```
<!DOCTYPE html>
<html lang="fi">
  <head>
    <title>Bootstrap Esimerkki</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
    <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>
  </head>
  <body>
    <div class="container">
      <h1>Bootstrap testisivusto</h1>
      <p>Tämä osa on container luokan sisällä.</p>
      <p>Container luokka tarjoaa responsiivisen sisällön.</p>
    </div>
  </body>
</html>
```

Kuva 10. Bootstrapin avulla luotu yksinkertainen verkkosivu.

Bootstrap käyttää HTML- ja CSS-ominaisuuksia, jotka vaativat HTML5 doctypen, jonka takia se täytyy sisällyttää sivun alkuun, mukaan lukien kieliattribuutti ja oikea charset, eli merkistöstandardi.

Varmistamalla oikeanlainen renderöinti kaikilla laitteilla ja kosketus zoomaus, pitää myös meta tagin sisään sisällyttää osat, jotka kertovat seuraamaan ruudun leveyttä ja asettamaan zoomaustason, kun sivu ensimmäisen kerran latautuu selaimessa.

(W3Schools Bootstrap A.)

## 6.2 Grid System eli ruudukkojärjestelmä

Bootstrapin grid system (ruudukkojärjestelmä) antaa luoda 12 palstaa sivustolle. Kaikkia 12 palstaa ei ole pakko käyttää erikseen, vaan palstat voidaan tehdä leveämpiin ryhmiin, esimerkiksi 4/4/4 tai 3/6/3 palstajajoilla.

## Uudistettavien verkkosivujen toteutus COSS ry:lle

Bootstrapin grid systemissä on neljä luokkaa: xs (puhelin näytöt), sm (tablettien näytöt), md (työpöytäkoneiden näytöt) ja lg (isot näytöt). Luokkien avulla voidaan luoda dynaamisempia ja joustavampia layouteja. Grid Systemiä voidaan käyttää kuvan 11 mukaisesti.

```
<div class="row">
  <div class="col-**-**"></div>
</div>
<div class="row">
  <div class="col-**-**"></div>
  <div class="col-**-**"></div>
  <div class="col-**-**"></div>
</div>
<div class="row">
  ...
</div>
```

Kuva 11. Bootstrapin ruudukkojärjestelmä

Seuraavan esimerkin avulla voidaan luoda kolme saman levyistä palstaa, jotka alkavat tabletista ja skaalautuvat suuriin näyttöihin. Puhelimella palstat kasaantuvat automaattisesti päällekkäin.

```
<div class="row">
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-4">.col-sm-4</div>
</div>
```

Jos halutaan luoda eri levyisiä palstoja, jaetaan sivusto niin, että palstojen summaksi tulee 12, kuten seuraavassa esimerkissä.

```
<div class="row">
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-8">.col-sm-8</div>
</div>
```

(W3Schools Bootstrap B.)

### 6.3 CSS

Bootstrapin CSS asetukset sisältävät yleismaailmalliset asetukset, oleelliset HTML elementit tyyliteltyinä ja paranneltuina laajennettavilla luokilla ja edistyneellä ruudukkojärjestelmällä. (Bootstrap CSS.)

### 6.4 Komponentit

Bootstrap sisältää toistakymmentä uudelleen käytettävää komponenttia, joiden avulla voidaan luoda alavetovalikoita, syöttöryhmiä, navigaatioita, hälytyksiä ja paljon muuta. Bootstrap sisältää esimerkiksi yli 250 symbolia, kuten euro-merkki, plus-merkki, miinus-merkki ja lukuisia muita. (Bootstrap Components.)

## 6.5 JavaScript

Käyttämällä Bootstrapin toistakymmentä jQuery –lisäosaa saadaan sivuston komponenteista elävämpiä. Lisäosat voidaan liittää yksi kerrallaan tai kaikki kerrallaan. (Bootstrap JavaScript.)

## 7 SIVUSTON TOTEUTUKSEN VAIHEET

Tämä sivusto toteutettiin aloittamalla ohjelmointi ensiksi paikallisesti tietokoneella, jonka jälkeen vasta alkuun luodut sivut sijoitettiin julkiselle puolelle tuotantoympäristöön Web-palvelimelle. Testituotantoympäristöstä, sivut siirrettiin COSS ry:n sen aikaisten sivujen kehittäjien palvelimelle, josta saatiin sisältö käyttöön.

Tärkeimmät ohjelmat projektissa olivat Notepad++, XAMPP ja WinSCP. Notepad++-ohjelmaa käytettiin lähdekoodin editoimiseen, XAMPP:in avulla pystytettiin paikallinen Apache-palvelin ja WinSCP:llä siirrettiin tiedostot FTP-palvelimelle.

Sisällönhallintajärjestelmäksi valittiin WordPress, joka on jo ennestään tuttu toimeksiantajalle. Sivusto piti alun perin toteuttaa Odoon-järjestelmällä, mutta sen vajavaisuuden takia päädyttiin yhdessä toimeksiantajan kanssa heille tuttuun sisällönhallintajärjestelmään. Pitämällä sisällönhallintajärjestelmän WordPressinä, saadaan myös sivuston aikaisempi sisältö säilytettyä, eikä näin ole tarvetta siirtää sisältöä toiseen järjestelmään.

### 7.1 Toteutuksen aloitus

Ennen sivuston kehitystyön alkamista, sivustosta toteutettiin muutama luonnos paperille ja rautalankamalliversio sivuston pääpiirteistä. Sivustoa aloitettiin toteuttamaan paikallisessa verkkoympäristössä XAMPP-ohjelman avulla, tällä tavoin ei ole heti tarvetta ”oikealle” Web-palvelimelle. Paikalliselle palvelimelle asennettiin WordPress-sisällönhallintajärjestelmä, jonka jälkeen aloitettiin sivuston pohjan rakennus. Projektia työstettiin XAMPP:n avulla siihen saakka, kunnes sivuston perusrunko oli valmis, minkä jälkeen sivuston siirto voitiin siirtää julkiselle palvelimelle.

### 7.2 Sivuston siirto julkiselle Web-palvelimelle

Sivuston perusrungon valmistuessa tuli vastaan raja XAMPP-ohjelman käytössä sillä sivusto oli paikallisesti koneella, jolla sivustoa toteutettiin eli toimeksiantajalla ei siis ollut mahdollisuutta päästä tarkastelemaan sivuston kehitystä muutoin kuin kuvakaappausten kautta. Tässä vaiheessa sivusto tuli siis siirtää julkiselle puolelle Web-palvelimelle. Tuotantoympäristöön tehtiin samat asennukset kuin paikalliselle palvelimelle eli WordPressin asennus. WordPressin asennuksen jälkeen oli mahdollista siirtää paikallisella puolella tehty WordPress-teema julkiselle palvelimelle. Siirto tapahtui käyttämällä WinSCP-tiedostonsiirto -ohjelmaa ja lataamalla luotu teema Web-palvelimen WordPressin teemakansioon. Julkisella puolella oleva sivuston linkki lähetettiin yhteyshenkilöille, minkä jälkeen sivustosta saatiin arvokasta palautetta ja kehitysideoita.



### 7.3 Sivuston sisältö

Sivuston todellista ulkonäköä ei päässyt tarkastelemaan ilman sisältöä, joten edessä oli sisällön tuominen sen hetkisiltä sivuilta uudistetuille sivuille. Sivusto siirrettiin vanhalta Web-palvelimelta uudelle palvelimelle, jonka yritys nimeltä Seravo Oy oli pystyttänyt. Seravo Oy oli COSS ry:n sen hetkisten sivujen toteuttaja, joten heillä oli tietokannat, tiedostot jne. sisältö sivuilta valmiina. Sivusto oli siis kopio COSS ry:n sivuilta, mutta toisessa osoitteessa. Tunnusten vastaanoton jälkeen, pystyttiin jälleen ottamaan yhteys uuteen palvelimeen WinSCP-ohjelmalla, jonka avulla teemakansio saatiin siirrettyä uudelle palvelimelle. Tämän jälkeen sivuston kehittämisessä päästiin jälleen uudelle tasolle sisällön ansiosta. Tästä eteenpäin, kehitys oli sisällön sovittamista luodulle teemalle.

### 7.4 Testaus

Sivuston testausta tapahtui alusta saakka koko ajan. Aina kun jokin osa saatiin valmiiksi ohjelmoitua, sitä testattiin erityisen paljon mm. eri resoluutioilla ja laitteilla. Sivustolle ei toteutettu erikseen testausta vaan sivustoa testattiin yhdessä COSS ry:n kanssa.

Sivuston testauksessa käytettiin hyväksi myös selainten elementtien tarkastelu ominaisuutta, jonka avulla pystyttiin helposti ja nopeasti debugaamaan ongelmia, sekä testaamaan eri CSS-tyylejä ilman että koodin tarvitsee tehdä muutoksia. Kuvassa 12 näkyy Google Chromen Tarkastele elementtiä-työkalu.



Kuva 12. Google Chromen elementtien tarkasteluominaisuus

### 7.5 Julkaisu ja jatkokehitys

Verkkosivusto saatiin tehtyä valmiiksi, mutta sivustoa ei ehditty julkaisemaan tämän työn aikana. Jatkotoimenpiteistä ei ole sovittu toimeksiantajan kanssa. Verkkosivujen kehitys ei tosin koskaan pysähdy julkaisuun vaan jatkuu pitkälle sivuston julkaisun jälkeen. Yhdistys oli kuitenkin tyytyväinen sivustoon ja sen sopivuuteen heidän käyttöönsä.

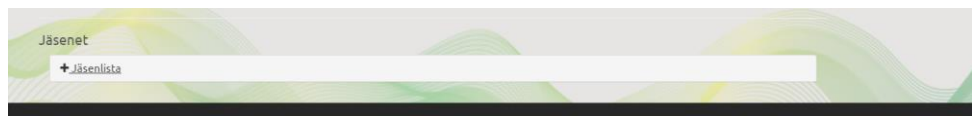
## 8 SIVUSTON SUUNNITTELUPROSESSI

Sivuston yksi päätavoitteista oli saada niistä selkeämmät ja yksinkertaisemmat ilman, että sivuston ulkonäkö kärsisi. Sivuston pääalue on tiedotus uutisten ja artikkeleiden kautta ja tiedon jakaminen sisällön kautta, mikä teki sivuston rakentamisesta haasteellisen.

### 8.1 Rakenne, visuaalisuus ja typografia

Sivustoja suunniteltaessa ja rakentaessa, sivuston rakenne täytyy olla hyvin tarkkaan mietitty, sillä sivuston rakenteen avulla voidaan niistä rakentaa joko helposti tai vaikeasti hahmottavat. Jokainen osa-alue, navigointipalkista alapalkkiin vaikuttaa käyttäjäystävällisyyteen. Sivujen tulisi olla esteettömät ja hyvän näköiset eli sivujen tulisi näyttää hyvältä, oli käyttäjän näyttöpäätteen resoluutio mikä tahansa. Käyttämällä Bootstrap-frameworkia, saadaan helposti tehtyä responsiivinen rakenne sivustolle. Bootstrapin komponenttien ja toimintojen avulla, sivustosta saadaan myös selkeä, yksinkertainen ja moderni.

Yhdistyksen visuaalinen ilme sivustolle syntyi pääasiallisesti valituista väreistä, fonteista, uudesta logosta ja työskentelyn aikana syntyneestä graafisesta elementistä. Yhdistyksellä ainoa toive värien suhteen oli, että vihreä väri aikaisemmilta sivuilta säilytettäisiin. Vihreään sävyyn tehtiin hieman värisäätöä raikkaammaksi ja se tulee esille usein sivustolla. Myös hieman tummempaa vihreää käytetään sivustolla. Sivuston muu värimaailma hyödyntää viimeisimpiä trendejä. Viimeisimmät trendit noudattavat neutraaleja värejä, joita sivustolle lisättiinkin, kuten harmaata mustaa ja valkoista. Sivustoa pyrittiin myös elävöittämään graafisella elementillä, joka näkyy sivujen alhaalla juuri ennen alapalkkia. Kuvassa 13 näkyy graafinen elementti, jota käytettiin sivustolla.




Kuva 13. Sivuston graafinen elementti

Fontiksi sivulle valittiin Ubuntu Font Family. Ubuntu Font Family on yksilöllinen ja räätälöity fontti, jolla on persoonallinen ulkonäkö ja tunne. Kirjasintyyppinä Ubuntu Font Familyssä on sans-serif. Se on selkeä fontti sekä työpöytäkoneissa, että mobiililaitteissa. Fontti on myös oiva valinta nimensä perusteella sivustolle siksi, että kyseessä on avoimen lähdekoodin yhdistys.

Sivustolle liitettiin myös skaalautuvia vektorikuvaikoneja sisältävä kirjasto nimeltä Font Awesome. Font Awesome sisältää yhteensä 585 ikonia, joita voidaan käyttää hyväksi sivustolla. Tässä työssä ikoneja käytettiin Bootstrapin ikonien sijaan. Ikoneja käytettiin pääasiallisesti uutisissa helpottamaan, selkeyttämään ja lyhentämään uutisen julkaisu ajankohtaa, kirjoittajaa, kategoriaa ja kommenttien lukumäärää. Ikonien avulla ei tarvinnut kirjoittaa mitään mikään arvo tarkoittaa, vaan niiden avulla saatiin helposti ilmaistua nämä kaikki pienessä tilassa, kuten alla olevassa kuvassa

14 näkyy. Font Awesomen avulla saatiin sivustolle myös sosiaalisen median painikkeet.



9.9.2015 Moona Puhja COSS tiedottaa 9 kommenttia

Kuva 14. Ikonien avulla ilmaisu säästää tilaa.

## 8.2 Navigointipalkki

Navigointipalkki on yksi sivuston tärkeimmistä ominaisuuksista, sillä se määrittää kuinka helposti ja nopeasti sivulla pystyy navigoimaan eri sivujen välillä, sekä kuinka helposti loppukäyttäjälle on tiedossa, millä sivulla ollaan. Tämän sivuston navigointipalkista luotiin seuraavan lainen:

- Etusivu
- Ajankohtaista
  - Uutiset
  - Coss tiedottaa
  - Jäsenet tiedottavat
- Avoin kunta
- Avoimuus
- Coss
- Kalenteri
- Ota yhteyttä

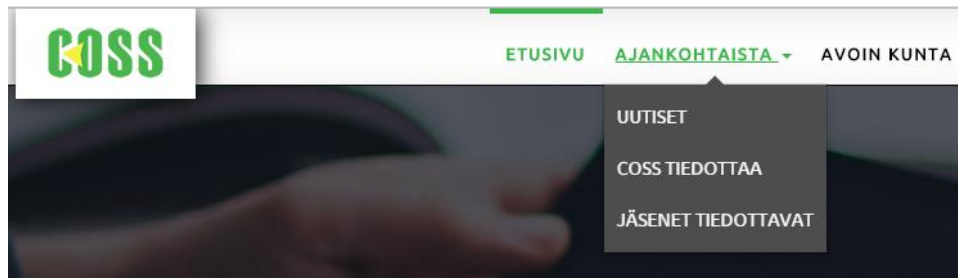
Navigoitavien sivujen määrä tulisi pitää suurimmillaan n. seitsemässä sivussa. Tätä suurempaa määrää ihmisen on vaikea hahmottaa ja navigointi sivustolla hidastuu ja hankaloituu. Suuren sisäsivujen määrän takia navigointipalkki pyrittiin pitämään mahdollisimman yksinkertaisena ja selkeänä. Suuren sivumäärän takia navigointipalkista ei avaudu kuin ainoastaan Ajankohtaista-linkistä alavetovalikko. Käyttäjän avattua alavetovalikon laatikon päälle ilmestyy pieni kolmio osoittamaan ylös mihin päälinkkiin alavetovalikko kuuluu.

Navigointipalkki seuraa käyttäjän rullaamista sivulla eli kun käyttäjä rullaa sivuja alaspäin, pysyy palkki mukana ylhäällä. Alkuperäinen idea sivustoa suunniteltaessa oli häivyttää valikko alaspäin rullatessa ja tuoda se takaisin näkyväksi ylöspäin rullatessa.

Navigointipalkin vasempaan reunaan sijoitettiin COSS ry:n logo, joka aluksi on kehystetty ja suurempi, mutta sivua alaspäin rullatessa kehukset lähtevät häiventyvästi pois, jotta se ei häiritse sivustolla selaamista. Häiveefekti toteutettiin CSS3:n avulla.

Efektit ovat kätevä tapa lisätä sivustolle näyttävyyttä ja tämän vuoksi navigointipalkkiin lisättiinkin pieni, mutta silmille mieluisa efekti. Kun käyttäjä vie hiiren sivulinkin päälle, ns. hoverefekti, ilmestyy linkin päälle vihreä palkki animoidusti vierien. Tämä sama vihreä palkki pysyy myös aktiivisten linkkien päällä. Aktiivinen linkki maalantuu myös saman väriseksi kuin pieni palkki linkin päällä. Animointi toteutettiin CSS3:n avulla. Kuvassa 15 navigaatiopalkin värit, ominaisuudet ja alavetovalikko.

## Uudistettavien verkkosivujen toteutus COSS ry:lle



Kuva 15. Sivuston navigaatio

Mobiilinäkyvässä linkit ovat vertikaalisti horisontaalin sijaan, sillä näin säästyy leveydestä ja kapeammalla näytöllä onnistuu selaamaan sivustoa. Mobiilinäkyvässä valikko täytyy erikseen avata painikkeen kautta, jonka jälkeen sivut listautuvat päällekkäin vertikaalisti. Mobiilinävygoinnin taustaväri on sama väri kuin alasvetovalikossa normaalissa navigoinnissa eli tumma. Navigointipalkin mobiilinäkyvyys toteutettiin kätevästi Bootstrap-frameworkin avulla, joka sisältää responsiivisen ruudukkojärjestelmän. Kuvassa 16 navigaatio, kun ruudun koko pienenee.



Kuva 16. Mobiili-näkymän navigaatio

### 8.3 Slideshow ja sisältöalue

Etusivun slideshow eli kuvarulla toteutettiin Bootstrap-frameworkin carousel-toiminnon avulla. Carouselin avulla kuvarullaan voidaan helposti lisätä kuva ja siihen tekstiä. Tämän sivuston kuvarulla kertoo mille sivustolle käyttäjä on saapunut, toinen kuva mainostaa uutiskirjeen tilaamista ja kolmas kuva kehottaa liittymään jäseneksi. Jokaisessa kuvassa on tekstiä, jonka alla on painike, jota kautta pääsee lukemaan lisää asiasta. Kuvarulla ei ole muilla sivustoilla, sillä se ei sopinut muualle kuin etusivulle.

Sisältöalue vaihtelee riippuen siitä mihin sivu esimerkiksi liittyy. Etusivun sisältöalue eroaa täysin kaikista muista sivuista, koska siihen yritettiin saada ”nostetta”. Etusivua lukuun ottamatta sisäisivut käyttävät 3/6/3 tai 8/4 palstajakoa.

Sisäisivujen sisältöalueen ylhäällä näkyy myös sivukartta, joka näyttää sivuston hierarkian ja millä sivulla käyttäjä juuri on. Linkkejä klikkaamalla

## Uudistettavien verkkosivujen toteutus COSS ry:lle

käyttäjää voi palata edellisille sivuille klikkaamalla sitä ilman, että tarvitsee painaa navigointipalkista. Tämä helpottaa sivustolla navigoimista ja helpottaa etenkin mobiilinäkymässä sivuston käytettävyyttä.

### 8.4 Sivupalkit ja alapalkki

Sivustolla on kaksi erilaista sivupalkkia, jotka sijaitsevat sisäsivujen joko oikealla puolella tai vasemmalla puolella. Molemmat sivupalkit näkyvät sisällöltään raskailla sivuilla, jotta sivustot olisivat miellyttävämmät. Oikealle puolelle sijoitettavassa sivupalkissa näkyvät viisi uusinta uutista, uutiskirjeen tilaus mahdollisuus sekä Twitter- ja Facebook-painikkeet. Vasemman puoleisessa sivupalkissa näkyy puolestaan sivun kaikki alisivut ja pääsivun nimi. Sivupalkkien avulla saadaan sivustosta myös paljon eläväisempi ja samalla myös sijoitetaan tarpeellista tietoa käyttäjälle kätevästi. Kuvassa 17 sivuston informatiivinen sisäsivu.



Kuva 17. Sivuston sivupalkit elävöittävät muuten tyhjää tilaa.

Alapalkista eli footerista yritettiin saada mahdollisimman tiivispakkaus sivuston sisällöstä. Alapalkista löytyy tietoa yhdistyksestä, uutiskirjeen tilauspainike, yhteystiedot, haku, sosiaalisen median painikkeet sekä linkit sivuston tärkeimmille sivuille. Alapalkki seuraa käyttäjää jokaiselle sivulle, joten on erittäin tärkeää, että siitä löytyy tiiviissä muodossa tiedot. Alapalkki jaettiin neljään eri osaan, mikä toteutettiin Bootstrapin ruudukkojärjestelmällä. Päätelaitteen näytön koon pienentyessä osat menevät päällekkäin. Kuvassa 18 näkyy sivuston koko alapalkki.



Kuva 18. Alapalkissa tulisi olla kaikki oleellinen sivustosta.

## 8.5 Uutiskirjeen tilaus

Uutiskirjeen tilaus toteutettiin COSS ry:n sen hetkisten sivuille räätälöidyn koodin avulla. Uutiskirjeen tilauspainike sijoitettiin sivupalkkiin, alapalkkiin ja etusivulle, jotta uutiskirjeen tilausmahdollisuus saataisiin maksimoitua.

## 8.6 Lisäosat

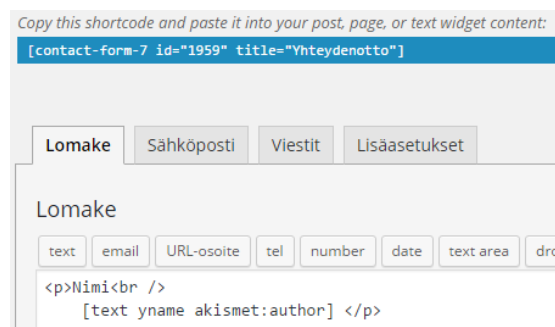
WordPressin suosion takia, siihen löytyy lisäosia lähes mihin tahansa ja jos lisäosaa ei löydy, voidaan sellainen kehittää itse. Tässä projektissa yritettiin käyttää mahdollisimman vähän lisäosia, sillä niiden päivitys on työlästä ja niiden kanssa saa olla tarkkana, jotta tietoturvaongelmia ei tule. Myös lisäosien kehityksen päättyminen täytyy ottaa huomioon. COSS ry:n sivuilla tosin oli lisäosia paljon, mutta kaikki niistä olivat tärkeitä ja niillä oli jokin tehtävä sivuston käytön kannalta katsottuna, eli turhia lisäosia ei juurikaan ollut. Sivustolla käytettiin myös muitakin kuin WordPressin omia lisäosia, kuten Twitter- ja Facebook-syöte.

Yleisesti WordPress-lisäosat ladataan WordPressin lisäosat sivulta, jonka jälkeen lisäosaohjelma puretaan ja ladataan FTP-kansioon WordPressin plugins-kansioon. Kun lisäosa on sijoitettu kansioon, voidaan WordPressin ylläpito puolelta asettaa lisäosa käyttöön, minkä jälkeen sitä voidaan soveltaa sivuilla ohjeiden mukaan. Seuraavissa alaluvuissa käydään lävitse sivuston tärkeimmät lisäosat.

### 8.6.1 Yhteydentottolomake

Yhteydenottolomake luotiin Contact Form 7-lisäosan avulla. Contact Form 7 avulla sivustolle saadaan helposti ja turvallisesti luotua useita yhteydenottolomakkeita ja sitä voi muokata myös oman näköiseksi. Kaa-  
vakkeiden lähetys toimii Ajaxin avulla ja sen muita turvallisuuskeinoja on mm. CAPTCHA ja Akismet spam -suojaus.

Yhteydenottokaavake luodaan WordPressin ylläpidosta, minkä jälkeen lisäosa antaa lyhyen koodin, joka liitetään sivulle haluttuun kohtaan. Kuvassa 19 näkyy miltä Contact Form 7 näyttää WordPressin ylläpidossa.



Kuva 19. Yhteydenottokaavakkeen luonti Contact Form 7:n avulla

### 8.6.2 Kalenteri

Kalenteri lisäosa oli valmiiksi ohjelmoitu edellisille sivuille, joten se kopiointiin suoraan vanhoilta sivuilta uusille. Muutoksia tehtiin CSS:ään, jotta kalenterista saatiin sivuille sopivan näköinen. Kalenteria voidaan selata eri kuukausien välillä käyttäen sivulta löytyviä painikkeita.

### 8.6.3 Yoast SEO

Hakukoneoptimointi on tärkein ominaisuus, jolla sivustot saadaan näkyväksi hakukoneissa. Vaikka WordPress on suoraan alkuperäisessä muodossaan suhteellisen tehokas hakukoneoptimointiin, saadaan Yoast SEO lisäosan avulla hakukoneoptimointi vieläkin tehokkaammaksi.

### 8.6.4 Facebook- ja Twitter-syöte

Sosiaalisen median tärkeyden takia sivustolle luotiin Facebook Page-plugin ja upotettu Twitter-syöte, jotka eivät ole WordPressin lisäosia, vaan Facebookin ja Twitterin tarjoamia pienoishjelmia. Näiden avulla ihmiset löytävät organisaation Facebook- ja Twitter-tilit helposti, sekä näkevät näiden medioiden viimeisimmät aktiviteetit suoraan nettisivuilta.

Facebook lisäosaan tarvitsee pienen osan koodia, joka saadaan generoitua helposti Facebookin kehittäjille suunnatussa palvelussa. Kehittäjän tarvitsee vain tietää organisaation Facebook-sivun osoite ja merkitä se vaadittuun kohtaan, jonka jälkeen palvelu luo automaattisesti HTML- ja JavaScript-koodin, joka sijoitetaan omalle sivulle. JavaScript liitetään heti `<body>` -tagin jälkeen. HTML-koodi sijoitetaan siihen kohtaan missä se halutaan sivuilla näkyvän. Ennen kuin koodi luodaan, voidaan lisäosalle tehdä muokkauksia, kuten mm. näytetäänkö kansikuva, näytetäänkö kaverien naamakuvia ja miten leveä tai korkea lisäosasta halutaan.

Twitter-syöte saadaan luotua lähes samalla tapaa kuin Facebook-syöte Twitterin omilla sivuilla. Ensiksi asetuksissa annetaan halutun käyttäjätunnuksen nimi, jonka jälkeen lisäosasta voidaan muokata halutun lainen, muokkaamalla mm. lisäosan korkeutta, teemaa ja linkkien väriä. Kun asetuksista on saatu mieluisat, painetaan Luo pienoishjelma-painiketta, jonka jälkeen palvelu luo JavaScript- ja HTML-koodin. Koodi sijoitetaan haluttuun paikkaan HTML-rakenteeseen samaan tapaan kuin Facebook-syöte. Twitter-syötteessä JavaScript-koodia ei erikseen tarvitse sijoittaa HTML-rakenteeseen, vaan se liitetään yhdessä HTML-koodin kanssa haluttuun kohtaan HTML-rakenteeseen. Kuvassa 20 yhdistyksen Twitter-tilin pienoishjelman asetukset ja jo valmiiksi luotu koodinpätkä.

## Uudistettavien verkkosivujen toteutus COSS ry:lle

**Asetukset**

Käyttäjätunnus @ cossfi

Vaihtoehdot  Ala näytä vastauksia  
 Laajenna kuvat automaattisesti

Korkeus 600

Teema Vaalea

Linkin väri Oletus (sininen)

Ulkoasun edistyneemmästä muokkauksesta saat tietoa [mukautusdokumentaatiosta](#).

Kieltäydy Twitterin räätälöinnistä [?]

**Tallenna muutokset**

Peruuta

**Esikatselu**

**Twiitit**

**COSS** @cossfi 28. elota  
Open Tampere Challenge avustilaisuus 2.9. klo 16-18!  
Ilmoittaudu mukaan [bit.ly/1lobUGe](#) #opendata  
#innovation #Tampere @OpenFinland  
Laajenna

**eOppimiskeskus** @eOppimiskeskus 28. elota  
Tule mukaan #EduCloud Allfancen EDU-yhteisöön!  
[ow.ly/RtZDM](#) #seoppi  
👉 Käyttäjän COSS uudelleentwiittaama  
Laajenna

**Softpedia Linux** @softpedialinux 27. elota  
#Linux 15.10 Beta Release Now Available for  
Twiittaa käyttäjälle @cossfi

`<a class="twitter-timeline" href="https://twitter.com/cossfi" data-widget-id=">`

Kopioi ja liitä koodi sivustosi HTML-rakenteeseen.

Käyttämällä Twitterin pienoisohjelmia hyväksyt [kehittäjien säännöt](#).

Kuva 20. Coss Ry:n Twitter-tilin pienoisohjelman luonti

### 8.6.5 MailPoet

MailPoetin avulla sivuston ylläpitäjät voivat lähettää uutiskirjeitä, automaattisia vastauksia ja ilmoituksia WordPressistä helposti. MailPoetin avulla voidaan mm. editoida uutiskirje drag & drop tyyliä, lähettää automaattisia ilmoituksia uusista artikkeleista ja automaattisia vastauksia esimerkiksi uusille uutiskirjeen tilaajille. MailPoetin avulla lähetetyt kirjeet näyttävät myös samalta useassa eri sähköpostipalvelussa ja laitteessa. Lähetetyistä uutiskirjeistä on myös mahdollista nähdä tilastoja, kuten avaukset, klikkaukset ja peruutukset. (WordPress Plugins 2015.)



## 9 SIVUSTON TEKNINEN TOTEUTUS

Tässä osiossa käydään lävitse sivuston toteuttamisen teknistä puolta. Osiossa käydään lävitse sivuston tärkeimmät osa-alueet loogisessa järjestyksessä ja kuinka ne ohjelmoitiin WordPressiin.

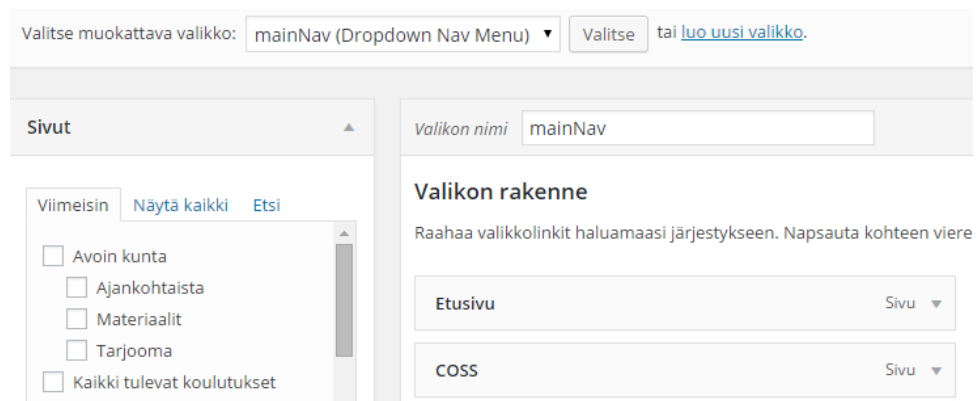
### 9.1 Functions.php

Functions.php-tiedostoon voi sisällyttää sivustolle tarkoitettuja funktioita. WordPress tunnistaa automaattisesti functions.php-tiedoston, josta voidaan kutsua funktioita helposti funktion nimellä. Tämän työn functions.php-tiedosto sisältää n. 400 riviä koodia, jossa on muun muassa funktioita kuten uutissivujen sivutus, Bootstrap navigaation luonti ja luonnollisestikin WordPressin vimpaimien rekisteröintejä.

### 9.2 Header ja navigointipalkki

WordPressissä header.php-tiedosto on yksi oleellisimmista osista WordPress-teemaa. Header-tiedosto sisältää sivuston yläosan merkinäkielen yleensä siihen saakka kunnes alkaa pääsisältöalue. Header-tiedostoon sijoitetaan sivuston meta-tiedot, sivuston otsikko, CSS- ja JavaScript-tiedostot, navigaatio ja koska tässä työssä käytettiin Facebookin Page Plugin-lisäosaa, niin myös JavaScript-koodi lisäosan aktivoimista varten.

Navigointipalkin rakenne luotiin Bootstrapin navigaatiokomponentilla. Navigaatiopalkin luominen oli haasteellinen tehtävä, sillä WordPressiin jouduttiin luomaan uusi Walker-luokka, jonka avulla voidaan käydä lävitse tietorakenteita ja esittää se HTML-muodossa. Internetistä löytyvien ohjeiden avulla saatiin luotua uusi Walker-luokka navigaatiolle. Walkerille tehtiin oma tiedosto nimeltään wp\_bootstrap\_navwalker.php. Walker rekisteröitiin vielä functions.php-tiedostossa, jonka jälkeen uusi navigaatio voitiin luoda WordPressin ylläpidosta. Lopullinen muotoilu navigaatiolle tehtiin CSS:llä. Kuva 21 on WordPressin ylläpidosta valikon rakenteen luomisesta.



Kuva 21. Navigaation rakenteen luominen WordPressin ylläpidossa

### 9.3 Slideshow

Slideshow eli kuvarulla, joka pyörii etusivulla, ohjelmoitiin erillisenä tiedostona nimeltä carousel-bootstrap.php. Se liitettiin etusivun sivupohjaan käyttämällä get\_template\_part()-funktiota. Toiminto ohjelmoitiin käyttämällä Bootstrapin carousel slideshow -komponenttia. Kuvarullaan sisällytettiin kolme kuvaa jotka mainostavat yhdistystä, uutiskirjeen tilausta ja jäseneksi liittymistä. Slideshow-tiedosto sijoitettiin kahden divin väliin etusivun templatessa. Kuvassa 22 näkyy kuinka slideshow liitettiin etusivun templateen.

```
<div class="header">  
  <div class="containerwrap">  
    <?php get_template_part('carousel', 'bootstrap'); ?>  
  </div>  
</div>
```

Kuva 22. Slideshow liitettiin etusivun sivutemplateen WordPressin funktiolla

Kuvat ja tekstit kova koodattiin tiedostoon, koska se oli helpoin ja nopein tapa ohjelmoida slideshow. Halutut kuvat ladattiin ensiksi palvelimelle omaan kansioonsa, jonka jälkeen ne liitettiin CSS:n avulla sivulle slideshow:n luokkiin background-image()-tyyliohjeen avulla. Kuvien alle ohjelmoitiin myös painikkeet Bootstrapin painike-komponentin avulla, joiden avulla käyttäjä pääsee kuvaan liittyvään sivuun. Linkit painikkeisiin toteutettiin WordPressin bloginfo("url")-funktion avulla. Funktiolle annettiin parametri "url", joka palauttaa arvoksi sivuston url-osoitteen. Funktion jälkeen annettiin halutun sisäsivun osoite. Kuvassa 23 esimerkki siitä, kuinka käyttäjä ohjataan yhdistyksen tietosivulle.

```
<div class="hero">  
  <hgroup>  
    <h1>COSS RY</h1>  
    <h3>Suomen avoimien tietojärjestelmien keskus</h3>  
  </hgroup>  
  <a href="<?php bloginfo('url'); ?>/goss-ry"><button cl<  
</div>
```

Kuva 23. Käyttäjät ohjataan slideshowin kuvan osoittamalle sivulle

Carousel-komponentti sisältää myös data-attribuutti valintoja, joiden avulla voidaan muuttaa mm. kuvien näyttöaikaa suoraan HTML:stä. Näyttöaika olikin ainoa valinta mitä tässä työssä muokattiin. Oletusarvo kuvien vaihtumiselle on 5000ms eli 5s, mutta tämä muutettiin 6000ms, jotta käyttäjät ehtivät havaitsemaan kuvan tarkoituksen. Toinen tapa muokata carouselin ominaisuuksia on JavaScriptin avulla, mutta niitä ei tarvinnut tehdä tämän työn aikana. Kuvassa 24 slideshowin attribuutteja ja arvoja.

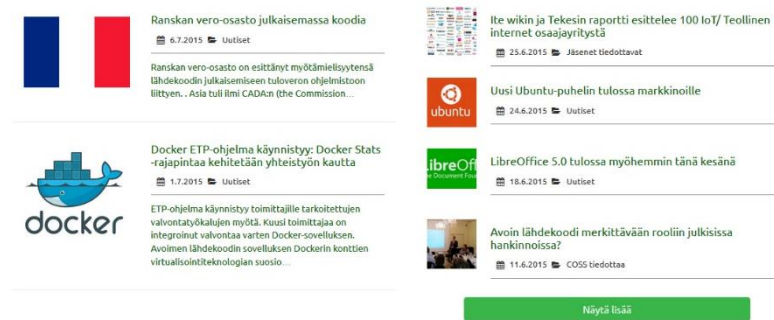
```
<div class="carousel fade-carousel slide" data-ride="carousel" data-interval="6000" id="bs-carousel">
```

Kuva 24. Carouselin data-interval attribuutti muokattiin 6000:een

## 9.4 Pääsisältöalueet

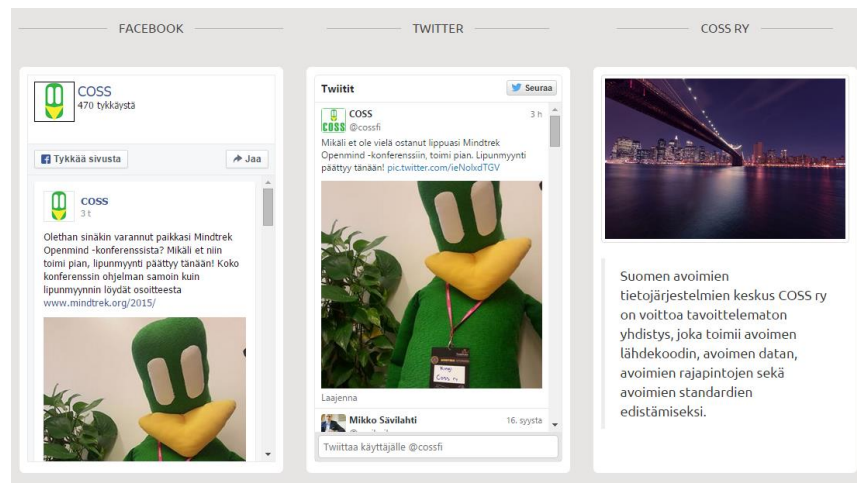
Tämän sivuston pääsisältöalueet käyttävät pääasiallisesti 3/6/3 tai 8/4 palstajakoa lukuun ottamatta etusivua. Etusivulla ei sinänsä ole mitään virallista palstajakoa, vaan sen eri osioissa sekoittuu eri jaot.

Etusivun sisältöalue alkaa jo slideshowsta, sillä slideshow sijaitsee ainoastaan etusivulla. Sisällön kannalta sisältöalue alkaa slideshown alapuolelta, johon luotiin uusimmat uutiset kahdelle palstalle. Uusimpien uutisten hakemisessa käytettiin WordPressin artikkelien haku -funktiota. Uutisosio jaettiin kahteen samankokoiseen palstaan käyttämällä Bootstrapin ruudukkojärjestelmää. Vasemman puoleisessa palstasta haetaan sivuston kaksi uusinta uutista ja oikean puoleisessa palstasta haetaan neljä uusinta uutista alkaen kolmanneksi uusimmasta uutisesta. Oikean palstan uutisten alle luotiin vielä painike Bootstrapin painike-komponentin avulla, jota painamalla pääsee lukemaan kaikki sivuston uutiset. Uutisen päivämäärä ja kategoria ilmaistiin Font Awesomen ikonien avulla. Kuvassa 25 näkyy kuinka uutisosio jaettiin etusivulla kahteen osaan.



Kuva 25. Etusivun uutisosion palstajako

Uutisosion alapuolelle luotiin Facebook- ja Twitter-syötteet, jotka yhdistyvät hyvin uusimpien uutisten jälkeiseksi osioksi. Syötteiden lisäksi osiossa on lyhyt esittelyteksti yhdistyksestä. Osio käyttää kolmen palstan jakoa eli 3/3/3. Facebook- ja Twitter-syötekoodit saatiin käyttöön kyseisten palveluiden kehittäjä sivuilta. Kuvassa 26 etusivun keskiosio, jossa Twitter- ja Facebook-syötteet sekä yhdistyksen lyhyt esittely.



Kuva 26. Sosiaalisen median syötteet, sekä yhdistyksen esittely

Seuraavaksi osioksi muodostui uutiskirjeen tilaus mainostosio. Tilaa uutiskirje toteutettiin jo valmiiksi luodulla vimpaimella sivustolle käyttämällä sen lyhytkoodia.

Viimeiseksi osioksi muodostui yhdistyksen jäsenesittely. Jäsenesittely on kaksi osainen; ensiksi ohjausryhmäjäsenten esittely ja toisena jäsenlista. Listan jäsenistä saa avattua painamalla jäsenlista-tekstistä.

Ohjausryhmäjäsenten logot ja linkit kova koodattiin erilliseen tiedostoon nimeltä ohjausryhma-jasenet.php Bootstrapin ruudukkojärjestelmällä nelipalstaiseksi. Tiedosto tuotiin etusivulle `get_template_part()`-funktion avulla. Ruudun koon pienessä palstajako muuttuu ensiksi kaksijakoiseksi, jonka jälkeen pienimmällä näytöllä palstajako on yksijakoinen eli logot ovat päällekkäin.

Jäsenlista kova koodattiin myös erilliseen tiedostoon nimeltä `jasenlista.php` ja se myös tuotiin etusivulle `get_template_part()`-funktion avulla. Jäsenlistan avauksessa käytettiin Bootstrapin `accordion`-komponenttia, minkä avulla jäsenlista ei vie turhaa tilaa ja käyttäjä saa avattua sen halutessaan painamalla jäsenlistasta. Jäsenlistassa käytettiin myös Font Awesomen ikoneita hyväksi. Jäsenlistan ollessa kiinni tekstin edessä näkyy plus-merkki, mutta sitä painaessa ja jäsenlistan avauduttua, se muuttuu miinus-merkiksi. Toiminto toteutettiin luomalla lyhyt JavaScript-koodi. Kuvassa 27 näkyy avattu jäsenlista.



Kuva 27. Avattu jäsenlista

Lisäksi jäsenlistan taakse lisättiin graafinen elementti, joka elävöittää sivua. Elementistä luotiin png-kuva, jonka jälkeen se ladattiin palvelimelle ja lisättiin alapalkin yläpuolelle hakemalla kuvakansio palvelimelta WordPressin `get_template_directory_uri()`-funktiolla.

Informatiiviset sisäsivut käyttävät 3/6/3 palstajakoa. Sivujen vasemmassa reunassa sijaitsee alasisuvalikko ja oikeassa reunassa sivupalkki, josta näkyy mm. tuoreimmat uutiset. Uutissivu on samanlainen, mutta ilman alasisuvalikkoa, jonka vuoksi on mahdollista käyttää 8/4 palstajakoa. Sivupalkeista luotiin erilliset tiedostot, jotka tuotiin haluttuun templateen `get_template_part()`-funktion avulla.

## 9.5 Sivupalkit ja alapalkki

Sivustolle sijoitettiin kaksi sivupalkkia. Toinen sivustolla olevista sivupalkeista näyttää sivujen alasivuvalikon ja toinen näyttää tuoreimmat uutiset, uutiskirjeen tilauksen ja sosiaalisen median linkit.

Sivupalkin, joka näyttää alasivut, koodi koottiin Internetistä löytyvien ohjeiden perusteella. Koodia muokattiin raskaasti, jotta se sopi tämän työn käyttöön. Sivupalkki käyttää logiikkaa, joka tarkastelee onko sivu, jolla ollaan, päätason sivu vai alatasen sivu. Jos sivu on päätaso, luo se sivupalkkiin päätason sivun nimen otsikoksi. Alasivut ilmestyvät otsikon alle linkkeinä. Jos kyseessä on toisen asteen alasivu, sisentyy tämä oikealle ensimmäisen asteen alasivun alle, jotta navigoinnista tulee helpompaa. Kuvassa 28 osa koodista, jolla sivuvalikko luotiin.

```
if ($children) { ?>
<ul class="nav nav-pills nav-stacked">
<?php
if ($post->ancestors[1]) {
$ancestor_title = get_the_title($post->ancestors[1]);
$ancestor_link = get_permalink($post->ancestors[1]);
?>
<h1><?php echo $ancestor_title; ?></h1>
```

Kuva 28. Osa sivuvalikon koodista

Muotoilussa käytettiin avuksi Bootstrapin nav-pills- ja nav-stacked-luokkia. Lopullinen muotoilu, kuten linkkien väri ja sisennys, tehtiin CSS-tiedostoon.

## COSS

```
Jäsenyys ja jäsenedut
-----


- Jäsenet
- Liity kannatusjäseneksi
- Liity yhteisöjäseneksi


-----
Koulutus ja konsultointi
-----
```

Kuva 29. Lopullinen sivuvalikko sisältösivulla

Tiedostolle annettiin nimi sivu-valikko.php ja se sisällytettiin get\_template\_part()-funktion avulla haluttuihin sivu-templateihin. Kuvassa 30 näkyy kuinka sivuvalikko saatiin lisättyä templateihin.

```
<div class="col-md-3">
<?php echo get_template_part('sivu', 'valikko'); ?>
</div>
```

Kuva 30. Sivuvalikko sisällytettiin usein sisältösivuille

Sivupalkki, joka näyttää mm. tuoreimmat uutiset, ohjelmoitiin käyttämällä WordPressin funktioita. Sivupalkkia varten luotiin uusi tiedosto nimeltä sidebar-sidebar.php. Ohjelmoinnissa käytettiin hyväksi WordPressin ar-

tikkeleiden hakukoodia, mutta haku rajoitettiin viiteen artikkeliin. Tärkein osa sivupalkin luonnissa oli tosin sen muotoilu, joka toteutettiin CSS-tiedostoon. Ongelmana oli kuinka saada mahtumaan uutisten tieto pieneen kokoon. Hyväksi avuksi osoittautui Font Awesome kirjasto, jonka avulla saatiin päivämäärän ilmaisu todella pieneen kokoon.

Artikkelikuva sivupalkissa rajattiin WordPressin kuvan rajausta-funktiolla, joka löytyy ylläpidosta. Artikkelikuvan saa asetettua templateen `the_post_thumbnail()`-funktiolla, jolle annettaessa `thumbnail`-parametri, hakee parametrille asetetun arvon kuvalle. Tässä työssä `thumbnail`-parametrille annettiin 80:n pikselin korkeusarvo ja 150:n pikselin leveysarvo.

Sivupalkkiin lisättiin myös ”Tilaa uutiskirje” tilausosio sekä Twitter- ja Facebook-painikkeet. Alkuperäisen suunnitelman mukaan, sivupalkkiin piti latautua viimeisimmät Facebook päivitykset, mutta koska päivitykset latautuivat joka sivulla, jossa sivupalkki sijaitti, heikensi se sivujen latausaikaa. Upotuskoodit haettiin Facebookin ja Twitterin sivuilta. Uutiskirjeen tilaus toteutettiin upottamalla uutiskirjeentilaus lisäosan koodi sivupalkkiin. Kuvassa 31 vasemman sivupalkin uutisosio.



Kuva 31. Sivupalkin uutisosio

Alapalkki eli footer, tiedostonimeltään `footer.php`, sisältyy jokaiselle sivulle ja se sisällytetään sivutemplateen `get_footer()`-funktiolla.

Alapalkkiin kova koodattiin yhdistyksen tiedot ja sosiaalisen median linkit. Linkeille luotiin oma funktio `functions.php`-tiedostoon, jonka avulla niille voitiin luoda WordPressin ylläpidon vimpain osiosta oma navigaatio. Alapalkki tiedostoon haettiin `functions.php`-tiedostosta siihen tarkoitettu funktio. Tämän avulla linkkejä ei tarvinnut kova koodata vaan niitä pystyi lisäämään ylläpidossa alapalkkien linkkejä varten luotuun navigaatioon. Koko alapalkki muokattiin CSS:llä halutun näköiseksi. Kuvassa 32 näkyy funktio, jonka avulla WordPressin ylläpidossa voidaan lisätä vimpaimia ja kuvassa 33 esimerkki kuinka funktio haetaan `footer.php`:stä.

```
register_sidebar(array(  
    'id' => 'footer1',  
    'name' => 'Footer 1',  
    'before_widget' => '<div id="%1$s" class="widget %2$s">',  
    'after_widget' => '</div>',  
    'before_title' => '<h4 class="widgettitle">',  
    'after_title' => '</h4>',  
));
```

## Uudistettavien verkkosivujen toteutus COSS ry:lle

Kuva 32. Alapalkin funktio functions.php-tiedostossa

```
<?php if ( !function_exists('dynamic_sidebar') || !dynamic_sidebar('footer1') ) : ?>
<?php endif; ?>
```

Kuva 33. Footer.php-tiedostossa haetaan funktio functions.php:stä

## 9.6 CSS

CSS-tiedostot sijoitettiin header.php-tiedostoon. Tässä työssä käytettiin yhteensä neljää eri CSS-tiedostoa. Päätiedostona käytettiin WordPressin havaitsemaa tiedostoa nimeltä style.css. Tämän tiedoston avulla liitettiin muut CSS-tiedostot työhön `@import url('')` komennon avulla. Style.css-tiedoston lisäksi käytettiin Bootstrapin bootstrap.css- ja bootstrap-responsive.css-tiedostoja. Näiden lisäksi Slideshow:lle luotiin täysin oma CSS-tiedosto, jotta sitä olisi helpompi ylläpitää, nimeltään carousel.css.

Style.css-tiedostoon voidaan sisällyttää teeman tietoja, kuten sen nimi, tekijät, kuvaus, versio numero, avainsanoja ja lisenssi. Style.css on myös tiedosto johon muokatut tyyliohjeet laaditaan. Bootstrapin CSS-tiedostoihin ei saa tehdä muutoksia, jotta se toimii oikein. Tämän työn CSS-tiedosto sisälsi n. 1000 riviä tyyliohjeita. Sivuston julkaisun ja palautteen jälkeen tyyliohjeiden määrä varmasti kasvaa lisää. Kuvassa 34 tämän työn style.css-tiedoston teeman tiedot sekä muiden CSS-tiedostojen tuonti.

```
/*
Theme Name: Coss Ry
Theme URI: -
Description: Coss Ry:lle räätälöity teema, jossa lähtökohdina ovat modernius, mutta
samalla myös helppokäyttöisyys. Teema käyttää uusinta Bootstrap -frameworkia, jonka ansiosta
sivut ovat täysin aidosti responsiiviset.
Author: Mikko Siivola & Heini Suvisuo
Author URI: -
Version: 0.9a
Tags: coss ry, responsiivinen, bootstrap, moderni, vaalea, vihreä, nykyaikainen, mobiili, 2015

License: Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)
License URI: http://creativecommons.org/licenses/by-sa/3.0/
*/
@import url('bootstrap/css/bootstrap.css');
@import url('bootstrap/css/bootstrap-responsive.css');
@import url('carousel.css');
```

Kuva 34. Style.css:n teeman tiedot, sekä muiden CSS-tiedostojen tuonti

## 9.7 Sivukartta ja sivunumerot

Edellä mainittujen toimintojen lisäksi tärkeä toiminto sivustolla oli sivukartan luonti sisäsivujen yläreunaan, josta näkee sivuhierarkian kyseiselle sivulle. Toiminto luotiin käyttämällä Internetistä löytyvää ohjetta nimeltä Dimox breadcrumbs. Toiminnon on kehittänyt käyttäjä nimeltä Dimox. Koodia muokattiin sivustolle tarpeelliseksi. Koodi lisättiin functions.php-tiedostoon, josta sitä kutsuttiin haluttuun sivu-templateen. Funktiota käytettiin informaatio sivujen yläreunassa.

Sivunumerointi toteutettiin Bootstrapin pagination komponentin avulla. Lisäksi sivunumeroinnille luotiin funktio, joka listaa sivunumerot. Sivunumeroita käytettiin uutissivuilla.

## 10 YHTEENVETO

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa responsiivinen, nykyaikaisia trendejä ja standardeja hyväksikäyttävä verkkosivusto. Työkentely alkoi hitaasti, mutta sisällönhallintajärjestelmän varmistuttua, työ eteni hyvää vauhtia eteenpäin. Vaikka alku oli hidas ja itse työ ei edennyt hirveästi eteenpäin, etsittiin ja tutkittiin verkkosivujen uusimpia trendejä ja tehtiin muistiinpanoja.

Työssä oli alun perin neljä tutkimuskysymystä, mutta kun sisällönhallintajärjestelmä pysyi lopulta samana, eikä uudelle järjestelmälle ollut tarvetta, poistettiin siitä yksi, joka oli ”Minkälaiset ohjeet asiakas tarvitsee järjestelmän käyttöön?”.

Sivuston suunnittelu tehtiin yhdessä henkilön kanssa, joka oli vastuussa yhdistyksen visuaalisesta ilmeestä. Yhteistyö sujui ilman minkäänlaisia ongelmia ja kommunikointi oli tiivistä. Suunnittelua toteutettiin yhdessä alusta loppuun saakka ja jos toinen sai hyvän idean, kerrottiin siitä toiselle, jonka jälkeen sitä lähdettiin toteuttamaan sivustolla ja katsomaan miltä idea näyttää.

Sivustosta saatiin loppujen lopuksi hyvinkin modernit ja yksinkertaiset. Helpoksi sivuston suunnittelun teki Bootstrap-framework, jonka avulla kehitystyölle jäi enemmän aikaa. Bootstrap-framework oli myös suurin syy minkä takia sivuista saatiin, ei ainoastaan modernin näköiset, mutta myös yhteneväiset ja yksinkertaiset sen komponenttien avulla. Tutkimalla satoja eri nykyaikaisia sivuja saatiin hyväksi pohjaksi tietotaso, jonka avulla sivuston suunnittelua ja toteutusta aloittaessa oli helppo lähteä rakentamaan uutta nykyaikaista sivustoa.

Lopulliseksi sisällönhallintajärjestelmäksi päättyi WordPress. Alkuperäisen suunnitelman järjestelmä, Odoo, päätettiin hylätä sen taipumattomuuden takia. WordPress oli myös jo ennestään toimeksiantajalla käytössä, mikä teki lopullisesta valinnasta helppoa.

Lopullinen sivusto vastasi hyvin toimeksiantajan tarpeita. Sivusto on kaiken kaikkiaan helpompi ja selkeämpi käyttää kuin heidän aiemmat sivut. Toimeksiantajan palautteen ansiosta sivuista saatiin muokattua täysin heidän mieleisensä sivustot. Tällä hetkellä uudet sivut odottavat julkaisua.

Web-ohjelmointi ei ollut kirjoittajalle uusi aihealue, vaan kirjoittajalla oli jo entuudestaan hyvä tausta Web-ohjelmoinnissa. WordPress-sisällönhallintajärjestelmä ei kirjoittajalle ollut täysin tuntematon, mutta ei myöskään äärettömän tuttuakaan. Tavoitteena olikin oppia käyttämään WordPress-sisällönhallintajärjestelmää ja siihen liittyviä lisäosia ja toimintoja, yhdessä Bootstrap-frameworkin kanssa. Kehitystyössä käytettiin muuten pääasiallisesti tuttuja työkaluja ja menetelmiä.

Web-ohjelmointi oli kiinnostanut kirjoittajaa kovasti jo aikaisemminkin, ja opinnäytetyö vahvisti mielenkiintoa alaa kohtaan entisestään.



## LÄHTEET

Bootstrap CSS. Css

<http://getbootstrap.com/css/>

Viitattu 12.8.2015

Bootstrap Components. Components

<http://getbootstrap.com/components/>

Viitattu 12.8.2015

Bootstrap JavaScript. JavaScript

<http://getbootstrap.com/javascript/>

Viitattu 12.8.2015

Comentum. 2010. What is a Content Management System (CMS)?

<http://www.comentum.com/what-is-cms-content-management-system.html>

Viitattu 20.8.2015

COSS ry. COSS

<https://coss.fi/coss-ry/>

Viitattu 14.8.2015

Guglieri, C. 2015. 20 steps to the perfect website layout

<http://www.creativebloq.com/web-design/steps-perfect-website-layout-812625?page=1>

Viitattu 8.9.2015

Farris, J. 2014. 8 Phases of the Web Design Process

<http://www.printmag.com/featured/phases-of-the-web-design-process/>

Viitattu 8.9.2015

Korpela, J. 2009. JavaScript (ja vastaavat)

<https://www.cs.tut.fi/~jkorpela/webjulk/3.2.html>

Viitattu 26.8.2015

Korpela, J. 2011. HTML5: uudet ominaisuudet. Jyväskylä: WSOYpro Oy.

Lopuck, L. The First Steps of Website Design

<http://www.dummies.com/how-to/content/the-first-steps-of-website-design.html>

Viitattu 8.9.2015

Lynch, P. Horton, S. 2008. Web style guide: basic design principles for creating web sites. New Haven & London: Yale University Press.

Mozilla Developers HTML A. 2015. HTML (HyperText Markup Language)

<https://developer.mozilla.org/en-US/docs/Web/HTML>

Viitattu 1.9.2015

Mozilla Developers HTML B. 2015. Introduction to HTML

Uudistettavien verkkosivujen toteutus COSS ry:lle

<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Introduction>

Viitattu 1.9.2015

Mozilla Developers CSS A. 2015. CSS

<https://developer.mozilla.org/en-US/Learn/CSS>

Viitattu 1.9.2015

Mozilla Developers CSS B. 2015. Why use CSS?

[https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting\\_Started/Why\\_use\\_CSS](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_Started/Why_use_CSS)

Viitattu 1.9.2015

Mozilla Developers Selectors. 2015. Selectors

[https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting\\_Started/Selectors](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_Started/Selectors)

Viitattu 1.9.2015

Mozilla Developers CSS3. 2015. CSS3

<https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3>

Viitattu 1.9.2015

Odoo A. About us.

<https://www.odoo.com/page/about-us>

Viitattu 20.8.2015

Odoo B. Website Builder Features

<https://www.odoo.com/page/website-builder-features>

Viitattu 8.9.2015

Tutorial Republic. Bootstrap Introduction

<http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-introduction.php>

Viitattu 12.8.2015

Shiotsu, Y. 2014. Web Development 101: What is Web Development?

<https://www.upwork.com/blog/2014/03/web-development-101-web-development/>

Viitattu 26.8.2015

W3Schools Bootstrap A. Bootstrap Get Started

[http://www.w3schools.com/bootstrap/bootstrap\\_get\\_started.asp](http://www.w3schools.com/bootstrap/bootstrap_get_started.asp)

Viitattu 12.8.2015

W3Schools Bootstrap B. Bootstrap Grids

[http://www.w3schools.com/bootstrap/bootstrap\\_grid\\_basic.asp](http://www.w3schools.com/bootstrap/bootstrap_grid_basic.asp)

Viitattu 12.8.2015

W3Schools JavaScript. JavaScript Where To

[http://www.w3schools.com/js/js\\_where.asp](http://www.w3schools.com/js/js_where.asp)

Viitattu 26.8.2015

W3Schools PHP. PHP 5 Syntax

Uudistettavien verkkosivujen toteutus COSS ry:lle

[http://www.w3schools.com/php/php\\_intro.asp](http://www.w3schools.com/php/php_intro.asp)

Viitattu 19.8.2015

WordPress.

<https://wordpress.org/about/>

Viitattu 14.8.2015

WordPress Codex Themes. Using Themes

[https://codex.wordpress.org/Using\\_Themes](https://codex.wordpress.org/Using_Themes)

Viitattu 3.9.2015

WordPress Codex Theme Development. Theme Development

[https://codex.wordpress.org/Theme\\_Development](https://codex.wordpress.org/Theme_Development)

Viitattu 3.9.2015

WordPress Codex Widgets. WordPress Widgets

[https://codex.wordpress.org/WordPress\\_Widgets](https://codex.wordpress.org/WordPress_Widgets)

Viitattu 16.9.2015

WordPress Developers. Template Hierarchy

<https://developer.wordpress.org/themes/basics/template-hierarchy/>

Viitattu 3.9.2015

WordPress Plugins. 2015. MailPoet Newsletters

<https://wordpress.org/plugins/wysija-newsletters/>

Viitattu 20.8.2015