

Lappeenrannan-Lahden teknillinen yliopisto LUT  
School of Engineering Science  
Tietotekniikan koulutusohjelma

Diplomityö

**Sami Tarkiainen**

**WORDPRESS-VERKKOSIVUSTON SAAVUTETTAVUUS -  
SAAVUTETTAVAN TEEMAN KEHITYS**

Työn tarkastajat: Associate Professor Jussi Kasurinen  
Assistant professor Antti Knutas

Työn ohjaaja: Associate Professor Jussi Kasurinen

# TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT

School of Engineering Science

Tietotekniikan koulutusohjelma

Sami Tarkiainen

## **WordPress-verkkosivuston saavutettavuus – Saavutettavan teeman kehitys**

Diplomityö 2019

46 sivua, 8 kuvaa

Työn tarkastajat: Associate Professor Jussi Kasurinen

Assistant professor Antti Knutas

Hakusanat: saavutettavuusohjeistus, wcag, wordpress

Keywords: web accessibility, wcag, wordpress

Tämän diplomityön tarkoituksena on tutkia mitä vaatimuksia Euroopan Unionin (EU) asettama saavutettavuusdirektiivi tuo uuden verkkosivuston suunnitteluun ja toteutukseen. Selvitän mitä saavutettavuus ja saavutettavuusohjeistus (Web Content Accessibility Guidelines, WCAG) tarkoittavat, mitä niillä pyritään saavuttamaan, sekä keneen ne vaikuttavat suoraan ja välillisesti.

Työssäni tarkastelen myös WordPress-julkaisujärjestelmää ja avaan sen rakennetta sekä toiminnallisuuksia. Tämän jälkeen selvitän saavutettavuusohjeistuksen vaatimia käytännön toimenpiteitä, joilla saavutettavuus toteutetaan sivustolle kokonaisuudessaan. Lisäksi avaan saavutettavuuden testausta ja siihen liittyviä. Arvoin kahden verkosta löytyvän ilmaisen testaustyökalun toiminnallisuuden sekä niiden toimivuuden.

Lopuksi esittelen case-esimerkillä käytännössä, kuinka saavutettavuus voidaan teknisesti toteuttaa WordPress-julkaisujärjestelmällä toteutetun verkkosivuston teemaan. Työn kirjoitushetkellä saavutettavuus on erittäin ajankohtainen aihe case-esimerkin

kohdeyritykselle, kun saavutettavuusdirektiivi vaatii julkishallinnon ja virastojen omistamat verkkosivustot täyttämään saavutettavuusohjeistuksen vaatimukset ja kriteerit. Teeman toteutuksen jälkeen testaan toteutuksen onnistumista käyttäen hyväksi todettuja testaustyökaluja. Tämän jälkeen arvioin kehitystyön ja testauksen onnistumista kokonaisuutena.

## **ABSTRACT**

Lappeenranta-Lahti University of Technology LUT  
School of Engineering Science  
Degree Programme in Software Engineering  
Sami Tarkiainen

### **Accessibility for WordPress-website – Accessible theme development**

Master's Thesis 2019

46 pages, 8 figures

Examiners: Associate Professor Jussi Kasurinen  
Assistant professor Antti Knutas

Keywords: web accessibility, wcag, wordpress

The purpose of this thesis is to examine what requirements the European Union (EU) Accessibility Directive sets to the design and implementation of a website. I will explain what accessibility and Web Content Accessibility Guidelines (WCAG) mean, what they aim to achieve, and who they affect directly and indirectly.

In my thesis, I will also go through the basics of WordPress Content Management System (CMS) and open its structure and functionality. I will also outline practical steps provided by the accessibility guidelines to implement accessibility for the site. In addition, I will tell how accessibility can be tested and measured. I will showcase two publicly available free to use tools for testing the accessibility and evaluate their functionality of.

Finally, I will use the case study to demonstrate how accessibility can be implemented within a WordPress theme. At the time of writing, accessibility is a very timely topic for the case examples target company, where the Accessibility Directive requires websites owned by government to meet the requirements and criteria of the accessibility guidelines. After the

theme has been implemented, it will be tested with the testing tools presented in this work. After this I will evaluate how development process and testing succeeded overall.

## **ALKUSANAT**

Aivan aluksi haluan antaa erityiskiitoksen Janne Tallqvistille. Hänen halunsa ja tarve saada oma diplomityö kasaan, motivoi myös minua. Ilman hänen panostustansa ei siis tämäkään työ olisi koskaan valmistunut.

Aloitin opiskeluni Lappeenrannan teknillisessä yliopistossa vuonna 2010 ja lähes 10 vuotta myöhemmin sain diplomityön valmiiksi ja valmistuin. Kuka olisi uskonut näiden vuosien aikana, kun en siihen itsekään enää uskonut. Näköjään kaikki on mahdollista, kun siihen ei tarpeeksi usko.

# SISÄLLYSLUETTELO

<b>1</b>	<b>JOHDANTO.....</b>	<b>4</b>
1.1	TYÖN TAUSTA.....	4
1.2	TYÖN TOTEUTUS.....	5
1.3	TYÖN RAJAUKSET.....	6
1.4	TYÖN RAKENNE.....	6
<b>2</b>	<b>SAAVUTETTAVUUS.....</b>	<b>7</b>
2.1	SAAVUTETTAVUUSOHJEISTUS (WCAG 2.0).....	7
2.1.1	<i>Ohjeistuksen sisältö ja periaatteet.....</i>	<i>8</i>
2.1.2	<i>Saavutettavuusohjeistuksen ongelmat.....</i>	<i>9</i>
2.2	SAAVUTETTAVUUSDIREKTIIVI JA LAKI SUOMESSA.....	9
2.2.1	<i>Keitä laki velvoittaa?.....</i>	<i>10</i>
2.2.2	<i>Lain toteutuksen aikataulut Suomessa.....</i>	<i>11</i>
2.2.3	<i>Saavutettavuusseloste.....</i>	<i>11</i>
2.3	SAAVUTETTAVUUDESTA HYÖTYVÄT KÄYTTÄJÄRYHMÄT.....	11
2.3.1	<i>Näkörajoitteiset.....</i>	<i>12</i>
2.3.2	<i>Kuulorajoitteiset.....</i>	<i>12</i>
2.3.3	<i>Fyysisesti tai motorisesti rajoittuneet.....</i>	<i>12</i>
2.3.4	<i>Kognitiivisesti rajoittuneet.....</i>	<i>13</i>
2.4	AVUSTAVAT TEKNOLOGIAT.....	13
2.4.1	<i>Näppäimistöseläminen.....</i>	<i>13</i>
2.4.2	<i>Ruudunlukuohjelmat.....</i>	<i>14</i>
2.4.3	<i>Esitustavan muuttaminen.....</i>	<i>14</i>
2.4.4	<i>Pistenäytöt.....</i>	<i>14</i>
2.5	VERKKOSIVUSTON SAAVUTETTAVUUDEN TESTAAMINEN.....	15
2.5.1	<i>AChecker.....</i>	<i>15</i>
2.5.2	<i>Wave - Website Accessibility Evaluation tool.....</i>	<i>16</i>
<b>3</b>	<b>WORDPRESS-JULKAISUJÄRJESTELMÄ.....</b>	<b>18</b>
3.1	TEEMAT.....	18
3.2	LISÄOSAT.....	19
3.3	SISÄLLÖN SYÖTTÖ.....	19
<b>4</b>	<b>SAAVUTETTAVUUDEN TOTEUTUS.....</b>	<b>21</b>
4.1	SIVUPOHJAT.....	21
4.2	TEKSTIN KOON JA KIRJASINTYYPIN MÄÄRITTÄMINEN.....	24

4.3	EI-TEKSTUAALINEN SISÄLTÖ.....	25
4.4	VÄRIEN KÄYTTÖ SIVUSTOLLA.....	26
4.5	LOMAKKEET JA TAULUKOT.....	27
4.6	YMMÄRRETTÄVÄ SISÄLTÖ.....	28
<b>5</b>	<b>CASE MAINOSTOIMISTO GROTESKI - SAAVUTETTAVAN TEEMAN</b>	
	<b>KEHITYS.....</b>	<b>30</b>
5.1	SAAVUTETTAVUUDEN TOTEUTUS TEEMAAN.....	31
5.2	SAAVUTETTAVUUDEN TESTAUS.....	32
5.2.1	<i>Työkalut lähdekoodin testaamiseen.....</i>	<i>33</i>
5.2.2	<i>Näppäimistöseläminen.....</i>	<i>35</i>
5.2.3	<i>Toimivuus ruudunlukuohjelmalla.....</i>	<i>36</i>
5.3	KEHITYSTYÖN JA TESTAUKSEN ONNISTUMINEN.....	37
5.4	TULEVAISUUS.....	37
<b>6</b>	<b>YHTEENVETO.....</b>	<b>39</b>
	<b>LÄHTEET.....</b>	<b>40</b>



## **SYMBOLI- JA LYHENNELUETTELO**

CSS	Cascadian Style Sheet
CMS	Content Management System
DOM	Document Object Model
HTML	Hypertext Markup Language
PHP	PHP Hypertext Preprocessor
UNCRPD	Convention on the Rights of Persons with Disabilities
W3C	World Wide Web Consortium
WCAG	Web Content Accessibility Guidelines
WP	WordPress

# 1 JOHDANTO

Verkkosivuja ja erilaisia verkkopalveluita käytetään maailmalla vuosi vuodelta enemmän ja enemmän. Yli 45% kaikista maailman ihmisistä käyttää verkkoa tietokoneellaan, tabletillaan, puhelimellaan tai digi-tv:llään päivittäin. Samaan aikaan lähes kuudenneksella maailman ihmisistä on vaikeuksia käyttää verkkopalveluita, koska niitä ei ole joko suunniteltu tai toteutettu oikein. Vaikka verkkopalveluiden tilaaja haluaa palveluunsa mahdollisimman suuren kohdeyleisön ja kävijämäärän, ei monestikaan ajatella muuta kuin mitä palvelun visuaalinen ulkoasu ja käytettävyys on suurimmalle osalle kävijöistä. [1]

Internetin käytöstä on tullut maailmalla ja yhteiskunnassamme jopa välttämättömyys. Erilaiset verkkopalvelut eivät enää tarjoa vain nuorille vapaa-ajankäyttömahdollisuuksia vaan niiden käyttöä odotetaan kaikenikäisiltä, jotta henkilö saa palvelua esimerkiksi pankista. Internetin tärkeys kansalaisten jokapäiväisessä elämässä puolestaan eriarvoistaa käyttäjiä jatkuvasti, joilla ei ole pääsyä verkon palveluihin. [2]

Käyttäjät, joilla on jonkinlaisia rajoitteita jäävät usein kokonaan palvelun kohderyhmän ulkopuolelle, vaikka muuten voisivat hyvin olla potentiaalisia palvelun käyttäjiä. Rajoitteita ovat mm. sokeus ja heikkonäköisyys, kuurous ja huonokuuloisuus, oppimisvaikeudet, kognitiiviset rajoitteet, liikuntakyvyn rajoitteet, puhevaikeudet, valoherkkyys sekä näiden yhdistelmät. Suurena huolena maailmalla on myös väestönikäntyminen. Vanhuksilla on tarve käyttää verkon palveluita, mutta isolla osalla heistä voi esiintyä jopa useita edellä mainituista rajoitteista. [3] [4]

Viime vuosina tähän ongelmaan on kuitenkin pyritty saamaan ratkaisu poliittisella tasolla. Euroopan unionin asettama saavutettavuusdirektiivi kokoaa yhteen tarpeelliseksi näkemänsä kriteerit ja minimivaatimukset käyttäjien eriarvoisuuden poistamiseksi verkossa, jolloin jokaisella verkon käyttäjällä on yhtäläiset mahdollisuudet lukea ja käyttää verkon sisältöä.

## 1.1 Työn tausta

Tämän työn kohdeorganisaationa toimiva Mainostoimisto Groteski Oy tuottaa asiakkailleen keskimäärin 40-50 verkkopalvelua vuosittain. Web-kehittäjiä on yrityksessä kaksi ja web-

suunnittelijoita yksi. Tavallisesti projektiin osallistuu vielä projektijohtaja, joka varmistaa tiedonkulun sekä yrityksen sisällä, mutta myös asiakkaalle. Yrityksen koosta huolimatta, se pystyy oman arvion mukaan keskimääräistä parempaan kehitykseen kuin alalla muut vastaavan kokoiset mainostoimistot, varsinkin paikallisella tasolla.

Kohdeyritys toteuttaa kaikki verkkosivustot ja -kaupat WordPress-julkaisujärjestelmällä käyttäen itse kehitettyä teemaa, joka perustuu maailmalla suosittuun Underscores.me kehysteemaan. Teema ei sisällä suurta määrää erilaisia ominaisuuksia, vaan vain pohjan sille, että kehitystyö asiakastoissa on nopeaa ja mutkatonta.

Teeman kehitys pyritään pitämään jatkuvana ja aina kun uudet ominaisuudet katsotaan tarpeelliseksi, tuodaan ne nopealla aikataululla osaksi teemaa. Teeman kehitykseen käytetään vuodessa noin 5-10 henkilötyöpäivää. Viimeiseksi kehitystyön kohteena oli verkkokaupan parempi liittäminen ulkoasun osalta osaksi teemaa.

Osa kohdeyrityksen asiakkaista on toimijoita, joiden jatkossa julkaistavat verkkopalvelut täytyy olla saavutettavuusdirektiivin ohjeistuksen mukaisia. Myös joidenkin asiakkaiden jo julkaistut sivustot tulee päivittää vastaamaan ohjeistuksia lainsäädännön luoman aikataulun puitteissa.

Tämän työn kirjoitushetkellä, ongelmana kuitenkin on, että kohdeyrityksessä ei ole riittävästi asiantuntijuutta, jotta se pystyisi toteuttamaan saavutettavan verkkopalvelu-uudistuksen asiakasyritykselle. Tätä prosessia helpottamaan, työn tuloksena tulen kehittämään teemasta uuden version, joka teknisesti täyttää kaikki saavutettavuuden vaatimat kriteerit. Yritys arvioi saavutettavien sivustojen määrän kasvavan tulevaisuudessa ja siksi haluaa panostaa saavutettavuuden osaamiseen ja asiantuntijuuteen.

## **1.2 Työn toteutus**

Tämän diplomityön tavoitteena on tarjota Mainostoimisto Groteski Oy:lle tarvittava tieto ja ohjeistus saavutettavan verkkopalvelun toteutukseen, sisältäen sivuston suunnittelun, kehityksen ja sisällöntuotannon. Työn tavoitteena on jatkokehittää yrityksen WordPress-

teemaa, jotta se täyttää saavutettavuuden vaatimukset, jotta asiakastöiden läpivienti olisi jatkossa tehokkaampaa.

Työn teoriaosuudessa tutkin saavutettavuutta yleisesti, sen lainsäädäntöä, vaatimuksia sekä niiden aiheuttamia toimenpiteitä, jotka verkkosivustolle on tehtävä. Olennaisena osana käsittelen ja avaan kansainvälisen saavutettavuusohjeistuksen sisältöä sekä kuinka ohjeistuksen eri kriteereitä toteutetaan WordPress-julkaisujärjestelmään.

### **1.3 Työn rajaukset**

Työssä huomioidaan Suomessa sekä Euroopassa yleisesti käytössä oleva World Wide Web Consortiumin (W3C) toteuttama saavuttavuusohjeistus WCAG 2.0. Työssä ei käsitellä muita maailmalla käytössä olevia ohjeistuksia tai vaatimuksia. Työn case-esimerkissä päähuomiona on WordPress-sivuston tema ja erityisesti sen tekninen toteutus, eikä sisältö.

### **1.4 Työn rakenne**

Työ rakentuu yhteensä kuudesta luvusta. Ensimmäisessä luvussa käsittelen työn taustaa, toteutustapaa ja rakennetta. Esittelen luvussa myös kohdeyrityksen ja kuinka se liittyy työhön ja sen tavoitteeseen. Toinen luku käsittelee saavutettavuutta teoreettiselta kannalta, mikä on sen tausta ja keneen se vaikuttaa ja ketkä siitä hyötyvät. Kolmannessa luvussa käsittelen WordPress-julkaisujärjestelmää, sen historiaa ja logiikkaa. Neljäs luku perehtyy syvemmin WordPress-julkaisujärjestelmään. Selvitän, kuinka saavutettavuusohjeistuksen eri kriteerit toteutetaan osaksi WordPress-alustaa. Luvussa käsittelen myös saavutettavuutta sisällön osalta ja käyn läpi erilaisten sisältöelementtien ja teknisen koodin tuomat haasteet. Viides luku keskittyy esittelemään saavutettavan WordPress-teeman kehitystä ja testausta käyttäen case-esimerkkiä. Luvussa arvioin kuinka hyvin kehitys- ja testaustyö onnistui ja oliko se riittävää. Lisäksi pohdin kohdeyrityksen näkökulmasta, mitä huomioita työ antoi tulevaisuudelle. Viimeisessä kappaleessa esittelen työn johtopäätökset.

## 2 SAAVUTETTAVUUS

Yhdistyneiden kansakuntien vammaisten ihmisoikeuksia koskeva yleissopimus (Convention on the Rights of Persons with Disabilities, UNCRPD) tuli voimaan vuonna 2008. Sopimuksella pyrittiin parantamaan ihmisten yhdenvertaisuutta. Myöhemmin sillä on ollut suuret vaikutukset nykyisiin paikallisen sekä Euroopan unionin tason yhtenäisiin uudistuksiin vammaisten perusoikeuksista tieto- ja viestintätekniikassa. [5]

Saavutettavuuden taustan ja kokonaisuuden, joka siihen liittyy, ymmärtäminen on olennainen osa työtä, jotta on mahdollista hahmottaa sen tarpeellisuus. Saavutettavuudella tarkoitetaan, että mahdollisimman monella olisi yhtäläiset oikeudet käyttää verkkopalveluita. Vaikka saavutettavuudella pääasiassa pyritään helpottamaan vammaisten ja toimintarajoitteisten henkilöiden verkon käyttöä, hyödyttää se kuitenkin kaikkia. Ottamalla huomioon myös voimakkaan väestön ikääntymisen sekä palveluiden siirtymisen verkkoon, saavutettavuus mahdollistaa näin suuremmalle joukolle ihmisiä pääsyn, nykyään verkkoon siirtyneisiin palveluihin. Ongelma ei koske vain pientä osaa ihmisiä, vaan jo Suomessa ihmisiä, joilla esiintyy joitakin vaikeuksia ja rajoitteita, on jo noin miljoona. [6]

Saavutettavuus ei kuitenkaan ole ominaisuus, jonka voi vain laittaa päälle, vaan se vaatii verkkosivuston toteutukselta joukon toimenpiteitä; hyvää suunnittelua ja teknistä toteutusta, selkeää ja ymmärrettävää kieltä sekä käyttäjien huomiointia.

### 2.1 Saavutettavuusohjeistus (WCAG 2.0)

Web Content Accessibility Guidelines (WCAG), eli saavutettavuusohjeistus on W3C:n ylläpitämä kansainvälinen virallinen ohjeistus verkkosivustojen ja -sisältöjen saavutettavuudesta. Ensimmäinen versio ohjeistuksesta julkaistiin vuonna 1999 ja nykyinen WCAG-ohjeistuksen versio 2.0 vuonna 2008. Ihan uudesta ohjeistuksesta ei siis ole kyse. [1] [6]

Versio 2.0 on suunniteltu teknisesti neutraaliksi, eli se ei pyri ottamaan kantaa tekniikkaan, jolla verkkosivusto on toteutettu. Sen tavoitteena on myös olla helposti ihmisen tai ohjelman testattavissa. WCAG-ohjeistus toimii Suomessa saavutettavuuslainsäädännön pohjana.

Ohjeistukseen kuuluu yhteensä 13 ohjetta, jotka on jaettu neljään periaatteeseen, jotka esittelen myöhemmin. Ohjeet sisältävät kaikkiaan 78 onnistumiskriteeriä, jotka puolestaan luokitellaan kolmeen tasoon; A, AA sekä AAA. Tasoilla erotellaan kriteerit sen mukaan mitä tasoa verkkosivuston saavutettavuuden halutaan noudattavan. Suomessa saavutettavuuslainsäädännön mukaisesti esimerkiksi julkishallinnon verkkosivustojen on täytettävä kaikki A- ja AA-tason kriteerit. Verkkosivustojen, joiden pääasiallisena kohderyhmänä on kehitysvammaiset, on puolestaan syytä noudattaa AAA-tason kriteerejä. [1]

Sivuston on läpäistävä kaikki yksittäiset tavoitetason kriteerit, jotta sen saavutettavuustaso on täytetty. Jos sivusto esimerkiksi täyttää kaikki AA-tason kriteerit, paitsi yhden kriteerin tasolla A, on sivuston saavutettavuusluokitus tällöin A.

Ohjeistus sisältää periaatteiden ja onnistumiskriteereiden lisäksi tekniikoita, joilla onnistumiskriteerit voidaan täyttää. Jokainen tekniikka sisältää kuvauksen ja esimerkit sen käytöstä.

### **2.1.1 Ohjeistuksen sisältö ja periaatteet**

Saavutettavuusohjeistuksen vaatimukset on jaoteltu neljään peruseriaatteeseen, jotka ovat havaittava, hallittava, ymmärrettävä ja lujatekoinen (toimintavarma). Osa vaatimuksista liittyy vain verkkosivuston tekniseen toteutukseen, eli tekniseen toteutukseen, mutta osa koskettaa myös sisällöntuotantoa, suunnittelua sekä sivuston ylläpitoa. Tästä syystä kaikkien verkkosivuston toteutuksessa ja ylläpidossa mukana olevien on syytä tutustua vaatimuksiin.

Havaittavalla tarkoitetaan sitä, että kaikki verkkosivuston sisältö esitetään niin, että se on mahdollisimman helposti käyttäjän nähtävissä. Havaittavan periaatteen vaatimukset on suunnattu pääosin suunnittelijoille, jotta sivuston suunnittelussa osataan huomioida mm. värimaailma, fonttikoko, riviväli jne.

Toinen periaate, hallittava on suunnattu enemmän sivuston teknisille toteuttajille, mutta osittain myös suunnittelijoille. Hallittavan periaatteen kriteereillä pyritään parantamaan

yhtäläistä mahdollisuutta käyttää sivustoa, esimerkiksi ilman hiirtä pelkällä näppäimistöllä, mutta myös niin, että käyttäjällä on riittävästi aikaa suorittaa haluamansa toiminnot.

Ymmärrettävä periaate on tarkoitettu pääosin sisällöntuottajille, koska sen vaatimusten lähtökohtana on, että sivuston kieli ei ole liian monimutkaista ja on helposti ymmärrettävissä. Se kuitenkin sisältää myös joitakin vaatimuksia teknisille toteuttajille, jotka toteuttavat sivuston sivupohjat. Tavoitteena on, että käyttöliittymä on ymmärrettävää ja loogista käyttää.

Lujatekoinen tai usein käytetty termi toimintavarma periaatteena antaa sivuston lähdekoodille vaatimukset, jotta se on teknisesti oikein toteutettu. Tällöin sivusto on käytettävissä mahdollisimman isolla joukolla apuvälineitä ja ohjelmia.

### **2.1.2 Saavutettavuusohjeistuksen ongelmat**

Saavutettavuusohjeistus sisältää myös paljon ongelmia, joka on ymmärrettävää. Aihe on niin laaja, että ohjeistuksen kirjoittaminen mahdollisimman ymmärrettävässä muodossa ja auttaen kaikkia on erittäin vaikea tehtävä. Vuosien aikana ohjeistus on päivittynyt useaan otteeseen ja edelleenkin se kehittyy kaiken aikaa vastaten suurempaan joukkoon ongelmia.

Ongelma ohjeistuksessa liittyy sen kriteereiden tulkinnanvaraisuuteen ja monimutkaisuuteen. Ohjeet on myös pyritty kirjoittamaan mahdollisimman objektiivisiksi ja selkeiksi, jolloin ohjeissa on käytetty paljon uusia termejä, joiden omaksuminen tuottaa monille haasteita. Vaikka ohjeistus on siis laadittu helpottamaan sivuston saavutettavuuden testausta, ei kuka tahansa sitä pysty toteuttamaan. [7]

## **2.2 Saavutettavuusdirektiivi ja laki Suomessa**

Euroopan parlamentin ja neuvoston direktiivi 2016/2102 julkisen sektorin elinten verkkosivustojen ja mobiilisovellusten saavutettavuudesta tuli voimaan 22.12.2016. Direktiiville vastineen Suomessa eduskunta hyväksyi 12.2.2019. Suomessa laki digitaalisten palvelujen tarjoamisesta (306/2019) ja lain sähköisestä asioinnista viranomaistoiminnassa

annetun lain muuttamisesta (307/2019) astui voimaan kaksi kuukautta myöhemmin, 1.4.2019. Suomessa saavutettavuusvaatimusten neuvonnasta ja valvonnasta vastaa Etelä-Suomen aluehallintovirasto AVI. [8] [9]

Direktiivin tärkeimpänä tavoitteena on edistää kaikkien ihmisten mahdollisuutta toimia yhdenvertaisesti yhteiskunnassa, joka digitalisoituu vauhdilla, sekä luoda koko Euroopan kattavat yhdenmukaiset vaatimukset julkisen sektorin verkkosivustojen ja mobiilisovellusten saavutettavuudelle. Lisäksi direktiivillä pyritään parantaa digitaalisten palveluiden laatua sekä parantaa Euroopan unionin saavutettavuuden toteuttamisen sisämarkkinoita. [6]

### **2.2.1 Keitä laki velvoittaa?**

Suomessa laki koskettaa julkisen hallinnon omistamia verkkosivustoja sekä mobiilisovelluksia. Pääasiassa laki velvoittaa viranomaisen asemassa toimivat organisaatiot muuttamaan verkkosivustot ja mobiilisovellukset saavutettavuusvaatimusten mukaisiksi. Viranomaisten lisäksi, laki vaatii julkisoikeudellisten laitosten ja joidenkin järjestön verkkosivustojen olevan saavutettavia aikatauluvaatimuksen mukaisesti. Näiden lisäksi myös erityisavustusta saavat verkkopalvelut, tietyin ehdoin, ja osa yksityisestä sektorista, kuuluvat lain piiriin.

Suurimpana kohderyhmänä laki vaikuttaa julkisoikeudellisiin laitoksiin. Laki digitaalisten palvelujen tarjoamisesta arvioi onko organisaatio julkisoikeudellinen laitos vai viranomainen käyttäen hankintadirektiivin ja hankintalain määritelmää. Yksinkertaisesti sanottuna, jos organisaatio on hankintalain mukainen hankintayksikkö, sen täytyy noudattaa lain mukaisia saavutettavuusvaatimuksia. Määrittäminen ei kuitenkaan aina ole helppoa.

Alla lista toimijoista, joiden määrittäminen on helppoa ja jotka toimivat viranomaismäärityksen alla:

- ammattikorkeakoulut ja yliopistot
- itsenäiset julkisoikeudelliset laitokset
- julkista hallintotehtävää hoitavat organisaatiot ja yhdistykset



- kunnat ja niiden omistamat organisaatiot sekä yhtiöt
- ministeriöt
- valtiolliset liikelaitokset
- valtion virastot

### **2.2.2 Lain toteutuksen aikataulutus Suomessa**

Kuten luvussa 2.2 kirjoitin, laki digitaalisten palveluiden tarjoamisesta astui voimaan Suomessa 1.4.2019. Saavutettavuusvaatimusten toteutumista ei kuitenkaan voida vaatia tapahtuvan hetkessä, vaan siksi on määritelty, että muutos tapahtuu portaittain. Ensimmäinen määräaika on 23.9.2019, joka koskettaa verkkopalveluita, jotka on julkaistu 23.9.2018 tai sen jälkeen. Ennen 23.9.2018 julkaistuilta verkkopalveluilta puolestaan vaaditaan, että ne täyttävät saavutettavuusvaatimukset viimeistään 23.9.2020. Mobiilisovelluksille on annettu 9 kuukautta enemmän aikaa, eli niiden on täytettävä vaatimukset 23.6.2021. Verkkopalveluiden ja mobiilisovellusten lisäksi takaraja saavutettavuudelle on annettu myös PDF-tiedostoille sekä organisaatioiden intranet-palveluille. Intranet-palveluita ei tarvitse kuitenkaan uusia vain saavutettavuuden takia, jos se on julkaistu ennen 23.9.2018. [6]

### **2.2.3 Saavutettavuusseloste**

Laki vaatii, että jokaiselta sivustolta, jota laki koskee, löytyy saavutettavuusseloste. Selosteen tarkoituksena on kertoa käyttäjille, kuinka saavutettavuus on toteutettu ja jos sivustolla on tehty poikkeuksia, niistä tulee lukea selosteessa. Selosteessa on myös tarjottava tapa antaa palautetta saavutettavuuteen liittyen. [6]

## **2.3 Saavutettavuudesta hyötyvät käyttäjryhmät**

Verkon saavutettavuudesta hyötyvät kaikki verkon käyttäjät, mutta joillekin käyttäjryhmille verkon käyttäminen on erittäin hankalaa tai mahdotonta ilman saavutettavuutta. Rajoittuneisuutta voi esiintyä ikään tai sukupuoleen katsomatta ja rajoitteet voivat olla synnynnäisiä, sairauden tai onnettomuuden seurauksena aiheutuneita. Suurena yksittäisenä ryhmänä on kuitenkin vanhukset, joilla voi esiintyä jopa useita seuraavaksi esiteltäviä rajoitteita. [10] [4]

### **2.3.1 Näkörajoitteiset**

Näkörajoitteisuutta on monen asteista ja se on yksi yleisimmistä esiintyvistä rajoitteista. Iso osa suomalaisista kärsii näkörajoittuneisuudesta jossakin määrin elämänsä aikana. Näkörajoitteisuus liittyy heikentyneeseen näköön yhdessä tai molemmissa silmissä tai jopa sokeuteen. Suomessa jo pelkästään näkövammaisia on noin 80 000, joista jopa yli 10 000 on kokonaan sokeita. Huononäköisille haasteet verkkosivustoilla liittyvät vaikeasti nähtäviin tai erotettaviin elementteihin, kuten tekstisisältöön tai klikattaviin linkkeihin. Kokonaan sokeat puolestaan pystyvät tyypillisesti käyttämään näppäimistöä kirjoittamiseen, joten he tarvitsevat avustavaa teknologiaa sisällön lukemiseen, eikä käyttäjän syötteen vastaanottamiseen. [9][11][4]

Kaikista yleisin näkörajoite liittyy värisokeuteen. Suomen väestöstä noin viisi prosenttia kärsii jonkin asteisesta värisokeudesta. Siksi verkkosivuston ulkoasua suunniteltaessa värimaailman valitseminen on erityisen tärkeää. Varsinkin kun väreillä pyritään esittämään jotakin käyttäjälle. [4]

### **2.3.2 Kuulorajoitteiset**

Kuulorajoitteisuus jakautuu kahteen tasoon. Käyttäjä voi olla joko kokonaan kuuro tai huonokuuloinen. Suomessa kuulovammaisia on noin 740 000, joista täysin kuurojen osuus on noin 8000. Kuulorajoitteiset käyttäjät voivat kuulla joitakin ääniä, mutta eivät välttämättä pysty erottamaan esimerkiksi puhetta, varsinkaan silloin kun puheen lisäksi esiintyy taustäääniä. Kuulorajoitteisille suurimmat haasteet verkkosivustoilla esiintyvät mediasisällöissä, kuten videoissa ja äänitteissä. [9][4]

### **2.3.3 Fyysisesti tai motorisesti rajoittuneet**

Fyysiset ja motoriset rajoitteet jakautuvat mm. lihaksiston heikkouksiin tai tahattomiin liikehdintöihin, tuntoaistin rajoitteisiin tai jopa raajojen menetykset. Tällaiset rajoitteet vaativat sivustolta erityisestä selkeitä klikattavia elementtejä, joita voidaan käyttää myös kosketusnäytöltä tai koko sivuston käyttö tapahtuu näppäimistöltä. [9]

### **2.3.4 Kognitiivisesti rajoittuneet**

Kognitiiviset rajoitteet liittyvät ihmisen kykyyn ymmärtää tietoa, puhetta, lukemaansa tai ongelmia uusien asioiden oppimisessa. Kognitiiviset haasteet eivät aina kuitenkaan liity käyttäjän älykkyyteen. Yleisin ongelma esiintyy kielellisissä haasteissa, jolloin käyttäjä vaatii verkkosivuston sisällöltä helposti ymmärrettävää kieltä, mutta osa liittyy verkkosivuston huonoon käytettävyyteen. Arvioiden mukaan Suomessa on noin 500 000 ihmistä, jotka tarvitsevat helppolukuista kieltä ymmärtääkseen lukemaansa. [9][4]

## **2.4 Avustavat teknologiat**

Avustavat teknologiat nimensä mukaisesti auttavat vammaista tai muuten toimintarajoitteista henkilöä käyttämään verkkosivuja. Avustavat teknologiat ovat monelle käyttäjälle välttämättömiä, jotta verkkopalvelun käyttäminen on ylipäänsä mahdollista, mutta joillekin ne voivat toimia avustamaan sivuston käyttöä. [4]

Avustava teknologia on esimerkiksi ruudunlukuohjelma, joka lukee verkkosivun sisällön ääneen sokealle. Avustavaa teknologiaa voi verrata hyvin apuvälineeseen, jolloin eri rajoitteita varten on erilaisia apuvälineitä. Kun sokea ei pysty lukemaan näytöllä olevaa tekstiä, ruudunlukuohjelma auttaa. Se ei kuitenkaan auta henkilöä, joka ei voi käyttää käsiään tietokoneen operoimiseen.

Avustavat teknologiat jaetaan kahteen eri kategoriaan. Päätelaitteelle asennettaviin ohjelmistoihin sekä päätelaitteeseen kytkettäviin erillisiin laitteisiin. Avustavat teknologiat on toteutettu ymmärtämään oikein toteutettua ja standardoitua HTML-koodia. Siksi sivuston teknisessä toteutuksessa on välttämätöntä ottaa huomioon avustavat teknologiat. Alla käyn läpi käytetyimpiä avustavia teknologioita.

### **2.4.1 Näppäimistöselaaminen**

Näppäimistöselaaminen on avustavista teknologioista yleisimpiä. Käyttäjänä on joko kokonaan sokea tai muuten näkövammainen, mutta myös fyysisesti ja motorisesti rajoittuneet henkilöt, joilla on hankaluuksia käyttää hiirtä verkon selaamiseen.

Näppäimistöseläminen vaatii verkkosivulta erityisesti teknisesti oikein toteutettua HTML-koodia.

Kun näppäimistön tarkoituksena on korvata hiiri, pitää sen toiminnallisuudet siirtää näppäimistölle. Hiiren rullan sijaan käyttäjä voi vierittää sivua näppäimistön nuolinäppäimillä. Näppäimistön sarkaimella (eli TAB-painikkeella) puolestaan valitaan esimerkiksi klikattava linkki, ja Shift ja TAB -näppäinyhdistelmällä liikutaan linkkien välillä taaksepäin. Enter-painiketta käytetään hyväksymään jokin valinta sivustolla.

#### **2.4.2 Ruudunlukuohjelmat**

Ruudunlukuohjelma on päätelaitteelle asennettava erillinen ohjelma, joka auttaa näkövammaisia lukemaan verkkosivuilla olevan sisällön ääneen. Ruudunlukuohjelmat vaativat verkkosivustolta, että sitä pystyy käyttämään näppäimistöllä ja, että sisältö on tehty havaittavaksi. Esimerkiksi kuvien pitää sisältää vaihtoehdoisen tekstin tai kuvauksen, jonka ruudunlukija pystyy lukemaan käyttäjälle. Ruudunlukuohjelma käyttää verkkosivuston dokumenttioliomallia (Document Object Model, DOM), ymmärtääkseen miten ja mistä sivu muodostuu. Eli kun sivusto on teknisesti oikein toteutettu ruudunlukuohjelma tietää ohjelmallisesti, milloin elementti on esimerkiksi otsikko ja milloin linkki. [12]

#### **2.4.3 Esitystavan muuttaminen**

Verkkosivujen esitystavan muuttaminen tapahtuu päätelaitteelle asennettavalla ohjelmalla tai selainlaajennuksella. Esitystavan muuttamisella tarkoitetaan, kun käyttäjä muuttaa sivuilla esiintyvän tekstin fonttia tai kokoa, taustaväriä tai piilottaa jotakin tarpeetonta sisältöä tai graafisia elementtejä. [9]

#### **2.4.4 Pistenäytöt**

Pistenäyttöjen käyttäjät ovat joko sokeita tai kuurosokeita. Pistenäytöt ovat erillisiä laitteita, jotka muodostavat näytölle pistekirjoitusta, jota tunnustelemalla käyttäjä pystyy lukemaan verkkosivuston sisältöä. Usein pistenäytöt tunnistavat verkkosivuston sisällön ruudunlukuohjelman tapaan käyttäen sivuston dokumenttioliomallia ymmärtääkseen sivun rakenteen ja sisällön.

## 2.5 Verkkosivuston saavutettavuuden testaaminen

Saavutettavuuden testaustyökalut ovat päätelaitteille asennettavia erillisiä sovelluksia, selainlaajennuksia tai verkossa toimivia palveluita, joilla voidaan testata täyttääkö sivusto saavutettavuuden vaatimukset. Työkalut voivat joko testata joitakin tiettyjä vaatimuksia esimerkiksi näkövammaisia varten tai ne voivat ottaa huomioon mm. WCAG:n kriteerit laajemmin ja verrata niitä sivuston dokumenttioliomalliin. WCAG 2.0:n lähtökohtana oli, että sen avulla sivuston testaaminen on mahdollista vastaten, täytyykö jokin kriteeri vai ei. [7]

Monet kriteereistä ovat kuitenkin tulkinnanvaraisia ja siksi työkalujen ja ohjelmien on hankala antaa kaiken kattavaa vastausta sivuston saavutettavuuden onnistumisesta ilman ihmisen tulkintaa. Työkalut eivät esimerkiksi pysty määrittämään itse, tarvitseeko kuva alt-attribuutin, eli onko kuva sisällöllisesti oleellinen vai pelkästään koristekuva. Työkalut jaetaankin automaattisiin ja puoliautomaattisiin työkaluihin, jolloin automaattiset työkalut paljastavat vain selkeimmät virheet, jotka on mitattavissa ohjelmallisesti ja jotka tarjoavat lisäksi listan sivun mahdollisista puutteista ja virheistä, jotka ihmisen on hyväksyttävä tai hylättävä. Seuraavaksi esittelen muutamia testaustyökaluja. [7][13]

### 2.5.1 AChecker

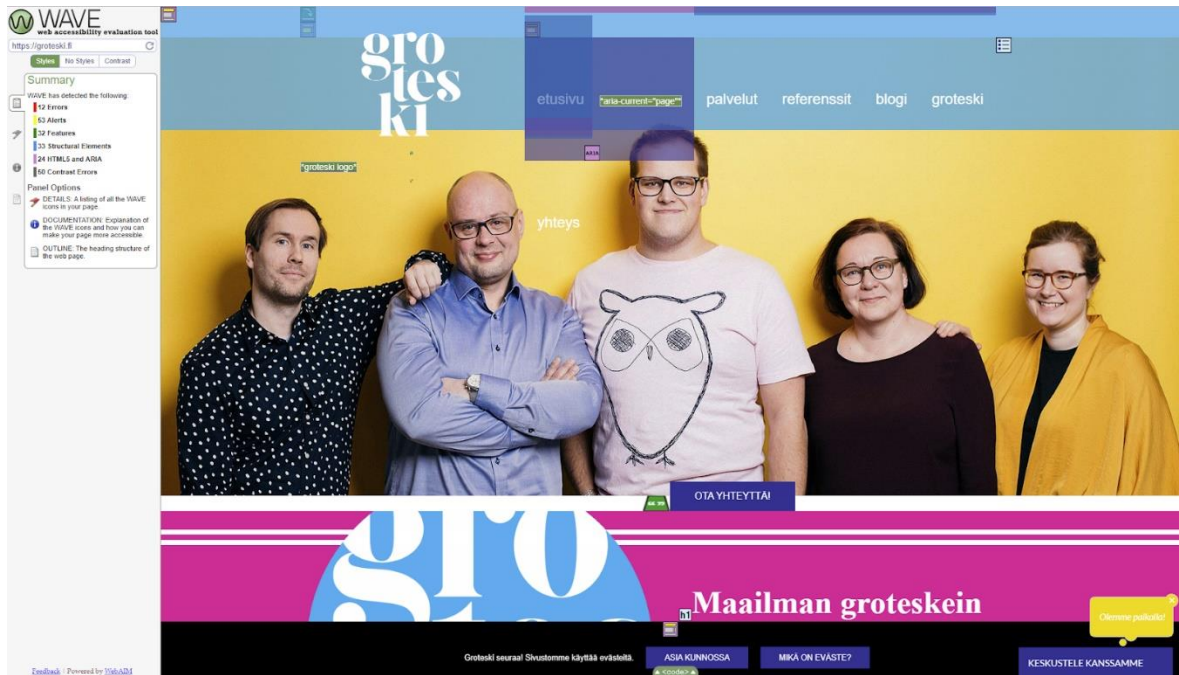
AChecker on avoimen lähdekoodin, puoliautomaattinen, ilmainen palvelu, joka testaa sivuston yksittäisen sivun HTML-sisällön saavutettavuuden ja antaa lopuksi raportin, jossa sivun ongelmat jaetaan kolmeen eri ryhmään:

- Tunnettuihin ongelmiin, jotka palvelu varmasti pystyy määrittämään. Nämä ongelmat täytyy korjata saavutettavuuden täyttämiseksi.
- Todennäköisiin ongelmiin, jotka voivat vaatia ihmisen päätöstä siitä, onko kyseinen ongelma todellinen
- Mahdollisiin ongelmiin, joita palvelu ei pysty tunnistamaan.

AChecker-työkalulla voidaan testata sivua käyttäen useita eri saavutettavuusohjeistuksia WCAG 2.0:n lisäksi. Työkalu tarkistaa sivuston HTML-sisällön oikeellisuuden lähettämällä sen W3C:n validointipalveluun (<http://validator.w3.org/>). Palvelu mahdollistaa myös CSS:n testaamisen niin ikään W3C:n validointipalvelulla (<http://jigsaw.w3.org/css-validator/>).

Palvelu esittää virheet allekkain ja tulostaa lähdekoodin rivin numeron, josta ongelma löytyi. Samalla palvelu kertoo mihin saavutettavuusohjeistuksen kriteeriin ongelma viittaa. Alla esimerkkikuva AChecker-työkalun tuloksista. [14]

The screenshot displays the AChecker Web Accessibility Checker interface. At the top, there are links for 'Login' and 'Register', and the 'Web Accessibility Checker' logo. The main section is titled 'Check Accessibility By:' and offers three methods: 'Web Page URL', 'HTML File Upload', and 'Paste HTML Markup'. The 'Web Page URL' method is selected, and the address 'https://grotoski.fi' is entered in the text box. A 'Check It' button is located below the text box. Below the input section is an 'Options' link. The 'Accessibility Review' section shows the results for the URL. It includes a summary of findings: 'Known Problems (23)', 'Likely Problems (0)', 'Potential Problems (630)', 'HTML Validation (12)', and 'CSS Validation'. The 'Accessibility Review' section also includes a section for '1.1 Text Alternatives: Provide text alternatives for any non-text content' and 'Success Criteria 1.1.1 Non-text Content (A)'. A specific error is highlighted: 'Check 7: Image used as anchor is missing valid Alt text.' The error details include the repair instruction: 'Repair: Add Alt text that identifies the purpose or function of the image.' The error is listed for four instances: 'Line 134, Column 28795:', 'Line 134, Column 32524:', 'Line 134, Column 46861:', and 'Line 134, Column 47451:'. Each instance shows the HTML code for the image: '. Alla esimerkkikuva työkalun tuloksista. [8] [13]



**Kuva 2.** Esimerkki Wave-testaustyökalun tuloksista

### **3 WORDPRESS-JULKAISUJÄRJESTELMÄ**

WordPress julkaisujärjestelmä (Wordpress.org) on julkaistu vuonna 2003 ja se on alun perin tarkoitettu blogien luomiseen ja hallintaan, mutta myöhemmin siitä on kehittynyt myös verkkosivustojen kehittämiseen ja ylläpitoon soveltuva alusta. WordPress, tai lyhennettynä WP, perustuu avoimeen lähdekoodiin, jota kehittää iso joukko vapaaehtoisia kehittäjiä ja on siksi ilmainen. Nykyään WordPress kuuluu maailman suosituimpiin julkaisujärjestelmiin. Huhtikuussa 2019 maailman 10 miljoonasta suosituimmasta sivustosta yli kolmasosa käytti WordPressiä sisällönhallintaan. WordPress on rakennettu PHP-ohjelmointikielellä (PHP: Hypertext Preprocessor) ja verkkosivuston sisältö tallennetaan MySQL-tietokantaan. WordPress ei tallenna yksittäisiä sivuja staattisesti vaan kaikki käyttäjien hakemat sivut ladataan ja rakennetaan palvelimella dynaamisesti. [16]

WordPress käyttää ulkoasun hallintaan teemoja (themes) ja toiminnallisuudet puolestaan ajetaan lisäosien (plugins) ja WordPressiin rakennetun logiikan kautta. Tällä rakenteella WordPress hajauttaa eri ohjelmalliset alueet komponentteihin, joita on helppo jakaa eri asennusten kesken. Tästä syystä WordPress suosittelee, ettei teemoihin tehdä toiminnallisuuksia eikä lisäosiin tyylimäärityä kuin vain siltä osin mikä kuuluu ks. lisäosaan. Seuraavaksi esittelen teemat ja lisäosat lyhyesti. [17]

#### **3.1 Teemat**

WordPress rakentaa sivuston ulkoasun käyttäen teemaa. Vähimmäisvaatimuksena teemalle on, että se sisältää indeksi- ja tyylitiedoston. Indeksitiedosto on ohjelmoitu PHP:llä, jolla sivun Hypertext Markup Language (HTML) rakennetaan dynaamisesti. Tyylitiedosto puolestaan määrittää sivuston ulkoasun käyttäen Cascading Style Sheets -tiedostoa (CSS) tyylisääntöjen luomiseen. Sivuston ulkoasun ja rakenteen monimutkaistuessa mm. eri sivupohjille tarkoitettujen templaattitiedostot, teeman toiminnallisuuksia varten funktiotiedosto sekä ulkoasun hallitsemiseen luodut javascript- ja tyylitiedostot ovat suositeltuja tiedostorakenteen selkeyttämiseen ja parempaan hallintaan.

WordPress sisältää valmiiksi kolme asennettua teemaa; Twenty Nineteen, Twenty Seventeen sekä Twenty Sixteen. Teemana voi siis käyttää jotakin näistä WordPressin oletusteemoista, ladata WordPressin teemakirjastosta ilmaisen tai ostaa sen valmiina



kolmannelta osapuolelta, tai tehdä sen itse. Valmiita ja puolivalmiita teemoja on verkossa huomattava määrä. Helpoimmillaan sivuston ulkoasun rakentaminen hoituu lataamalla WordPressiin valmis teema, johon vaihtaa vain tarvittavat ulkoasuelementit. Tällä tavalla monesti menetetään sivuston kuitenkin räätälöitävyys.

### **3.2 Lisäosat**

Lisäosien tarkoitus on rikastuttaa WordPress-julkaisujärjestelmän ominaisuuksia. Vaikka WordPressin kehittyessä sen ominaisuuksien ja toimintojen määrä on kasvanut suuresti ei se siltikään kykene kaikkeen itse, eikä kannatakaan. WordPress sisältää yleisesti katsoen vain ne tarpeelliset toiminnot, joita suurin osa käyttäjistä tarvitsee. Näitä ovat mm. artikkeleiden, sivujen, median ja käyttäjien hallinta. Näiden lisäksi jokainen sivuston ylläpitäjä pystyy lataamaan WordPressin asentamisen jälkeen ne lisätoiminnot, jotka hän näkee juuri sille sivustolle tarpeellisiksi. Tällä tavalla WordPress on voitu pitää yksinkertaisena, mutta skaalautuvana alustana.

WordPress julkaisujärjestelmän avoimeen lähdekoodin luonteen omaisesti lisäosia on kehitetty lukemattomiin eri tarkoituksiin ympäri maailman ja tämän työn kirjoitushetkellä niitä on WordPressin ylläpitämässä virallisessa lisäosakirjastossa 54 709 kappaletta. [18]


### **3.3 Sisällön syöttö**

WordPress on julkaisujärjestelmä tai kuten moni sanoo, sisällönhallintajärjestelmä, joten sisällönhallinta on pyritty tekemään yksinkertaiseksi, mutta riittävän monipuoliseksi. Onhan se alun perin kehitetty blogien hallintaan, joissa sisältö ja sen muokkaus suuressa roolissa. WordPressissä sisällön syöttö tapahtuu sen oman editorin avulla. Ennen WordPress 5.0 versiota, WordPress sisälsi yksinkertaisen Wordin kaltaisen tekstieditorin, mutta päivityksen myötä WordPress uudisti editorin niin sanottuun Gutenberg-editoriin.


WordPressin Gutenberg-editorissa sisältö rakennetaan modernimmalla tavalla elementeistä, joita voidaan hallita raahaa-pudota-tekniikalla. Myös elementtien valinta tapahtuu uuden käyttöliittymän kautta, jossa rivien jakaminen sarakkeisiin, eli palstoihin on nykyään mahdollista. Monien nykyaikaisten sisällönhallintaeditorien tapaan, Gutenbergiin

on mahdollista luoda omia määrittämiään elementtejä ja täten helpottaa sivuston päivitettävyyttä. Alla kuvankaappaus Gutenberg-editorista, jossa rivi on jaettu kahteen palstaan ja molempiin palstoihin on lisätty sisältöä.

## Etusivu

T ▾ H2 H3 H4 B I  ▾ ⋮

Maailman  
groteskein  
mainostoimisto



Groteski on mikkeliäinen mainostoimisto, joka ajattelee suuria ja tähtää korkealle. Perinteisten mainostoimistopalveluiden lisäksi Groteski toteuttaa mm. nettisivuja ja verkkokauppoja. [Me](#) tiedämme mitä teemme. Kurkkaa palveluihin ja totea itse, kuinka hyviä me olemme!

## Asiakkaita läheltä ja kaukaa

**Kuva 3.** Kuvankaappaus Gutenberg-editorista ja sen toiminnallisuuksista

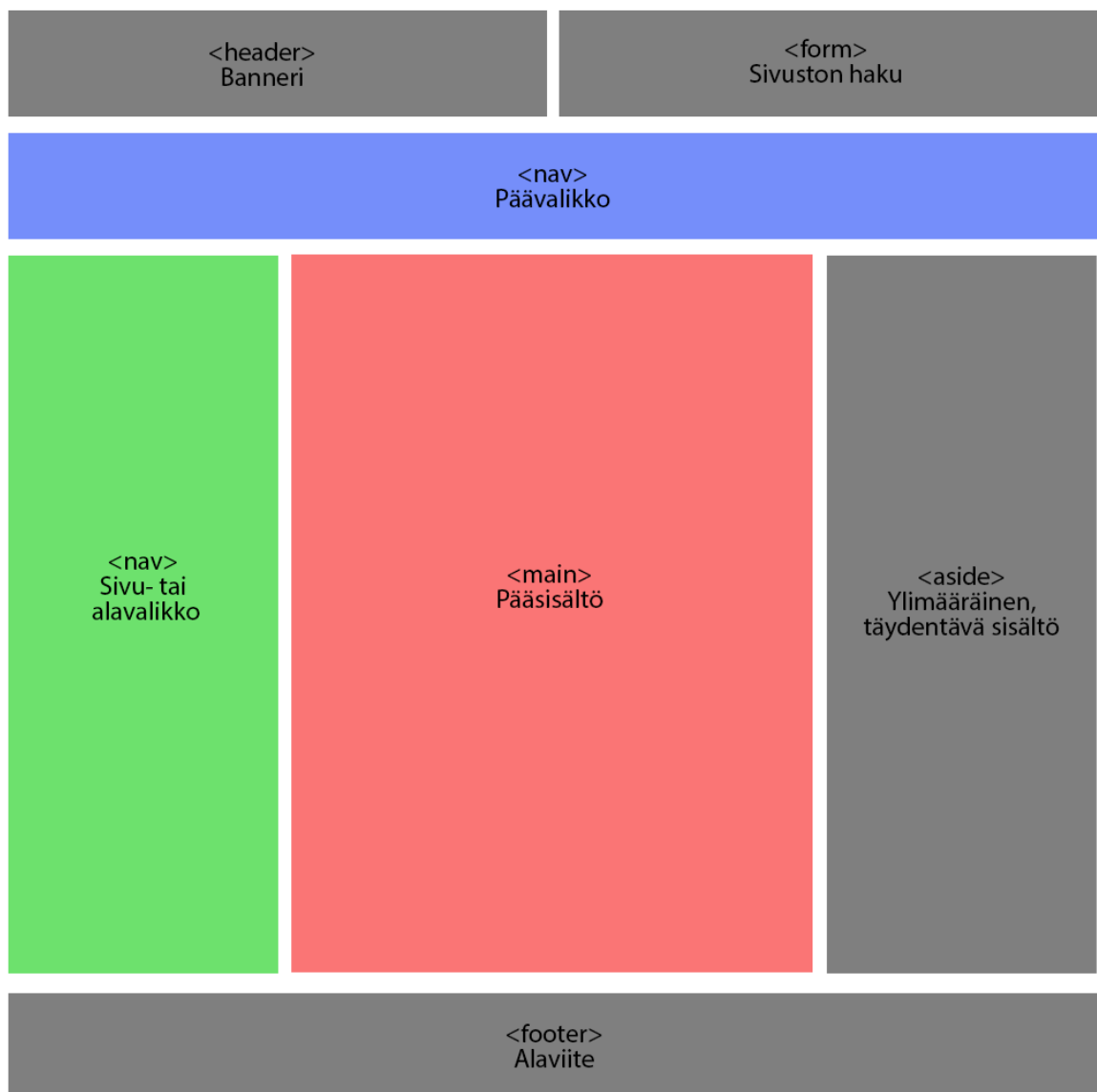
## **4 SAAVUTETTAVUUDEN TOTEUTUS**

Saavutettavuus on otettu vahvasti osaksi WordPressin kehitystä. WordPress kehittyy jatkuvasti ja se on kasannut 70:n vapaaehtoisen henkilön tiimin, joka testaa uusien ominaisuuksien ja ulkoasun saavutettavuutta. WordPress on myös määrittänyt saavutettavan koodausstandardin, jossa kaikkien uusien WordPressin ytimeen lisättävien ominaisuuksien on täytettävä WCAG 2.0 AA-taso. Luonnollisesti WordPress ei kuitenkaan ota kantaa muiden tekemiin teemoihin tai lisäosiin, joten siksi jokaisen kehittäjän on itse varmistettava, että tarvittaessa ne täyttävät saavutettavuuden vaatimukset. [17]

Seuraavaksi esittelen, kuinka saavutettavuus toteutetaan WordPress-pohjaiselle sivustolle, mutta käytännössä ohjeistuksen mukaisesti sivusto voidaan toteuttaa saavutettavaksi ilman julkaisujärjestelmääkin tai käyttäen jotain muuta julkaisujärjestelmää. Teknisesti sivuston toteutus lähtee oikein suunnitelluista ja määritellyistä sivupohjista.

### **4.1 Sivupohjat**

On suotavaa ja jo yleisen käytettävyyden kannalta tärkeää, että sivupohjat ovat yhdenmukaisia ja siten selkeitä ja helppokäyttöisiä. On myös hyvä muistaa, että sivuston toiminnalliset elementit tulisi suunnitella niin, että ovat käyttäjille tutunnäköisiä ja siten helppokäyttöisiä. Esimerkiksi niin, että käyttäjät tunnistavat linkit klikattaviksi linkeiksi. Tällöin sivustoa on loogista käyttää. Lisäksi on tärkeää, että sivupohjan eri tekniset elementit ovat selkeästi käyttäjien, mutta myös ohjelmallisesti erotettavissa. Tämä tarkoittaa sitä, että sivupohjien html-koodi on määritelty tavalla, jonka W3C ohjeistaa. Alla oleva kuva esittää sivupohjan tärkeimmät elementit ja johdonmukaisen järjestyksen. [18]



**Kuva 4.** Sivupohjan rakenne

Kuvassa 3. oleva pääsisältö on `<main></main>` -elementtien välissä. Pääsisällön on syytä erottua selkeästi muusta sisällöstä, jotta sivun tärkein sisältö on mahdollisimman helppoa havaita.

Pääsisältö voi koostua monenlaisesta sisällöstä, mutta on ehdottoman tärkeää, että jokainen sivuston sivu sisältää pääotsikon, joka on merkattu html-koodin otsikkotageilla, `<h1></h1>` -elementtien sisään. Pääotsikoita saa olla vain yksi jokaisella sivulla, mutta väliotsikoita, joita html-koodissa ovat tagit h2, h3, h4, h5 ja h6, puolestaan kehoitetaan käyttämään reilusti. Saavutettavuusohjeistus kuitenkin määrittää, että otsikointia on käytettävä loogisesti, eli

käytännössä h1-tasoinen otsikon jälkeen tulee h2-tasoinen otsikko, eikä esim. h3-tasoinen otsikko.

Otsikoinnissa on mietittävä myös niiden sisältöä niin, että ne ovat riittävän kuvaavia, jota WCAG-ohjeistuksen onnistumiskriteeri (2.4.6) vaatii tekemään. Hyvällä otsikoinnilla on suuri merkitys sivuston helppolukuisuuteen ja sitä myöten saavutettavuuteen mm. ruudunlukuohjelmaa varten. WordPressin Gutenberg-editorissa otsikot löytyvät yleisistä elementeistä, joista elementin valinnan jälkeen pystytään valitsemaan myös otsikon tyyppi. [9]

Sivuston navigaatio, eli valikko, tulee rakentua `<nav></nav>` -elementtien väliin. Navigaatioita voi sivustolla olla useita, kuten päänavigaatio ja alanavigaatio, mutta niiden on erotettava ohjelmallisesti sekä käyttäjälle toisistaan. Valikon elementtien on oltava selkeitä ja johdonmukaisia sekä vastata sivun pääotsikkoa, jolle sivulle se johtaa. Valikolla voi myös toteuttaa WCAG:n kriteerin esittää käyttäjälle millä sivulla hän on korostamalla kyseistä linkkiä valikossa. Toinen vaihtoehto vaatimuksen onnistumiselle on käyttää murupolkua.

Sivuston jokaisella sivulla tulisi pääotsikon lisäksi olla yksilöllinen nimi. Nimen määrittäminen tapahtuu html-koodissa `<head></head>` -osiossa `<title></title>` -elementtien sisällä. Kuten otsikoiden, myös sivun nimen olisi syytä olla riittävän kuvaava, jotta siitä voidaan ymmärtää sivun tärkein sisältö. Nimen lisäksi saavutettavuusohjeistus vaatii, että sivustolle on määritetty kieli html-koodissa. Kieli määritellään html-koodiin heti alkavaan `<html>`-elementtiin suomenkieliseksi näin: `<html lang="fi-FI">`. [19]

Saavutettavuusohjeistus ei vaadi sivustolta, että se sisältää hakutoiminnon, jolla käyttäjä voi etsiä haluamansa sivun. Hakutoimintoa voidaan kuitenkin käyttää saavuttamaan kriteeri 2.4.5, jolloin käyttäjälle annetaan useita mahdollisuuksia löytää haluttu sivu. Papunet myös suosittelee hakutoimintoa, kun sivusto on laaja ja sisältää yli 20 alasivua. WordPressiin on sisäänrakennettu hakutoiminnallisuus. Hakulomake voidaan lisätä sivustolle esimerkiksi sivupalkkiin haku-vimpaimella (widget). Vimpaimelle voidaan antaa otsikko sekä placeholder-teksti, jolla voidaan ohjeistaa käyttäjiä hakulomakkeen käyttöön. Hakulomake näyttää tulokset erillisellä hakutulokset-sivulla. [17][9]

## 4.2 Tekstin koon ja kirjasintyyppin määrittäminen

Saavutettavuusohjeistus ei anna määritystä fontin minimikoolle. Tämä johtuu siitä, että eri fontit voivat olla hyvin erikokoisia, vaikka fonttikoko olisikin sama. Lisäksi niiden luettavuus saattaa vaihdella, vaikka fonttikoko olisi sama. Siksi ohjeistus on jätetty tulkinnanvaraiseksi ja arvio jää tällöin suunnittelijalle onko fonttikoko riittävän suurta ja kirjasintyyppi mahdollisimman selkeä. [4]

WCAG:n kriteeri 1.4.4 vaatii, että käyttäjä pystyy itse määrittämään verkkosivuston fonttikoon, maksimissaan 200-prosenttiin asti. Vaikka käyttäjät pystyvät muuttamaan verkkosivustojen fonttikokoa selaimestaan, niin kaikki heikkonäköiset eivät ole tähän tottuneet. Tästä syystä on erittäin suositeltavaa, että sivustolle lisätään painikkeet, joilla käyttäjä pystyy itse määrittämään fonttikokoa suuremmaksi tarpeen mukaan. Kriteerin mukaan, sivuston rakenne ei saa kärsiä kuitenkaan sivuston sisällön suurentamisesta johtuen. WordPressiin tätä toimintoa ei ole rakennettu oletuksena, mutta käyttäen lisäosaa (WP Accessibility Helper WAH) tai toteuttamalla se itse jQueryllä ja CSS:llä toiminto on helposti lisättävissä sivustolle.

Kriteerin 1.4.4 takia tekstin koon määritystä ei suositella tehtäväksi pikseliarvoilla. Tekstin koon määritys tapahtuu tyylitiedostoon font-size -attribuutilla. Saavutettavuusohjeistus suosittelee sen määrittämiseen käyttämään em-arvoa, prosenttiarvoa tai ennalta määritettyjä kokoja.

Myös sivuston kirjasintyyppin valinnalla on erittäin suuri vaikutus sisällön luettavuuteen. Saavutettavuusohjeistus ei anna tarkkaa määritystä siitä, pitäisikö kirjasintyyppin olla päätteeton vai päättellinen, mutta Papunet suosittelee käyttämään päätteettömiä sans-serif -tyylisiä kirjasintyyppejä. Selkeitä päätteettömiä kirjasintyyppejä ovat mm:

- Arial
- Calibri
- Century Gothic
- Helvetica

- Tahoma

Helppolukuisia päätteellisiä, serif-tyylisiä kirjasintyyppiä ovat puolestaan:

- Times New Roman
- Georgia

WordPressissä kirjasintyyppin ja fonttikoon määrytykset tehdään suoraan tyylitiedostoon, tai kuten aiemmin mainitsin, teeman omiin asetuksiin, jos sellaiset käytetystä teemasta löytyy. Kirjasintyyppin määrytys tapahtuu font-family -attribuutilla tyylitiedostoon.

### 4.3 Ei-tekstuaalinen sisältö

Lähes kaikilla moderneilla sivustoilla käytetään jotain ei-tekstuaalista sisältöä. Ei-tekstuaalista sisältöä on mm. kaikki kuvat, infograafit, videot ja äänitiedostot, joita ruudunlukuohjelma ei pysty lukemaan ääneen ilman, että elementille annetaan tekstivastine. Tästä syystä saavutettavuusohjeistus vaatii kaikilta ei-tekstuaalisilta elementeiltä tekstivastineen. Ei-tekstuaalista sisältöä on myös lomakkeiden kentät sekä lomakkeen Captcha-kentät. [9]

Vaihtoehtoinen teksti tulee englanninkielestä alternative text ja se määrytetään kuvaelementin alt-attribuuttiin. Vaihtoehtoinen teksti kertoo, usein ruudunlukuohjelman lukemana, kuvan sisältämän tiedon niille sivuston käyttäjille, jotka eivät näe kuvaa. Saavutettavuusvaatimusten mukaisesti vaihtoehtoisen kuvan voi jättää tyhjäksi vain, jos kuvan tarkoitus on toimia koristekuvana. Koristekuvalla tarkoitetaan sellaista kuvaa, jonka tavoitteena ei ole välittää mitään informaatiota tai se ei ole merkityksellistä. [10]

WordPressissä kuvien alt-tekstit määrytetään mediakirjaston kautta alla olevan kuvan mukaisesti.

Tiedoston nimi: groteski\_logo\_2018\_valmis\_main.png  
Tiedoston tyyppi: image/png  
Siirretty: 26.9.2018  
Tiedostokoko: 31 KB  
Koko: 1080 kertaa 1080 pikseliä

---

Vaihtoehtoinen teksti   
[Kuvaile kuvan tarkoitusta.](#) Jätä tyhjäksi jos kuva on vain koristeellinen.

Otsikko

Kuvateksti

Kuvaus

**Kuva 5.** Vaihtoehtoisen tekstin määrittäminen kuvaan WordPress-julkaisujärjestelmässä

Video- ja äänitiedostoille ei HTML-kielessä pysty määrittämään vaihtoehtoista tekstiä. Niinpä niiden sisältämä tieto on tarjottava käyttäjille myös tavallisena tekstisisältönä. Videoille, jotka eivät ole aikasidonnaisia eli niin sanottuja livelähetyksiä, on lisättävä tekstitykset. Tämä auttaa kuulorajotteisia käyttäjiä, jotka eivät kuule videoissa puhuttua kieltä. Lisäksi video- ja äänitiedostojen toisto ei saa käynnistyä automaattisesti, vaan käyttäjälle on annettava mahdollisuus käynnistää mediasisältö. Myös mediasisällön tapauksessa on tärkeää, että käyttäjät, jotka käyttävät pelkkää näppäimistöä voivat ohjata mediasisällön toistoa.

#### 4.4 Värien käyttö sivustolla

Värien käyttäminen verkkosivustoilla on mietittävä tarkasti jo ennen sivuston toteutusta. Sivustojen värimaailma tulee usein yrityksen, yhteisön tai palvelun graafisesta ohjeistuksesta, mutta sitä on noudatettava vain siltä osin kuin saavutettavuusvaatimukset mahdollistavat. Sivuston värimaailmalla on ensisijaisesti vaikutuksia heikkonäköisten ja värisokeiden käyttäjien mahdollisuuteen käyttää sivustoa sujuvasti, mutta kuten muillakin saavutettavuuden vaatimuksilla, myös värimääritykset vaikuttavat yleiseen käytettävyyteen esimerkiksi pieneltä näytöltä luettaessa.



WCAG-ohjeistuksen kriteeri 1.4.1 vaatii, että värejä ei saa käyttää pelkästään tiedon tai toiminnallisuuksien esittämiseen, vaan siitä on kerrottava myös jollakin muulla käyttäjän ymmärrettävällä tavalla. Tyypillisiä kohtia ilmoittaa haluttu viesti värein ovat esimerkiksi lomakkeiden virheilmoitukset, joissa pelkän värin sijaan käyttäjälle on ilmoitettava virheestä myös tekstillä tai symbolilla.

Toinen WCAG-ohjeistuksessa vaadittu ominaisuus sivustoilta on tekstin ja taustan välinen riittävä tummuusero eli kontrasti. Kriteeri 1.4.3 vaatii, että taustan ja tekstin välinen kontrastiero tulee olla vähintään 4,5:1 kun teksti on normaalikokoista. Teksti määritellään normaalikokoiseksi, jos se on arvoltaan 14 pt. Jos fonttikoko on suurempaa, riittää kontrastieroksi 3:1. Teoriassa kontrastiero tarkoittaa, sitä kuinka monta kertaa valoisampi taustan ja tekstin välinen suhde on. Kontrastiero on mahdollista tarkistaa monella verkosta löytyvällä ilmaisella työkalulla kuten esimerkiksi: <https://contrastchecker.com/>. [6]

WordPressissä värimaailman määrittely voi tapahtua teeman omista asetuksista, jos teemalta sellaiset löytyy. Muussa tapauksessa värimäärittely kuuluu normaalin tyylimäärittelyn tapaan CSS-tiedostoon.

## **4.5 Lomakkeet ja taulukot**

WordPress ei oletuksena tue lomakkeita, mutta käyttäen esimerkiksi suosittua Contact form 7 -lisäosaa, lomakkeiden luominen ja muokkaaminen on mahdollista niin, että ne täyttävät saavutettavuusvaatimukset. Tärkeintä on, että lomakkeen elementeillä on omat nimilaput, eli html-koodin mukaiset `<label></label>` -elementit. Nimilappujen tarkoituksena on tuoda ilmi mitä sitä vastaavaan lomakekenttään tulee syöttää. Nimilapun tulee pitää sisällään myös tieto siitä, jos lomakekenttä on pakollinen.

Nimilappujen lisäksi, lomakkeen tulisi olla riittävän selkeä ja johdonmukainen, jolloin sitä voi kuunnella ruudunlukuohjelmalla sekä käyttää myös pelkällä näppäimistöllä. Jos lomakkeen täyttämisen on ollut ongelmia tai lomake on onnistuneesti lähetetty, on siitä ilmoitettava käyttäjälle tekstipalautteen muodossa.

Contact form 7 -lisäosa tarjoaa erinomaisen pohjan lomakkeelle, mutta oletuksena se ei täysin täytä saavutettavuusvaatimuksia. Nimilappuihin on määritettävä tarkasti for-attribuutilla sitä vastaavan lomakekentän id-attribuutti, alla olevan esimerkin mukaisesti.

```
<p><label for="your-name">Nimi (vaadittu)</label><br />
  [text* your-name id:your-name] </p>

<p><label for="your-email">Sähköposti</label><br />
  [email your-email id:your-email] </p>

<p><label for="your-subject">Aihe</label><br />
  [text your-subject id:your-subject] </p>

<p><label for="your-message">Viesti</label><br />
  [textarea your-message id:your-message] </p>

<p>[submit "Lähetä"]</p>
```

**Kuva 6.** Esimerkkilomake luotuna Contact Form 7 -lisäosalla

Taulukoiden lisääminen onnistuu WordPressin versiolla 5.0 käyttäen sen omaa Gutenberg-editoria. Papunet ohjeistaa kuitenkin tarkkaan harkitsemaan taulukoiden käyttöä ja niitä onkin syytä käyttää vain suuren datamäärän esittämiseen, kun muuta tapaa ei ole. Ongelma taulukoiden kanssa on käyttäjillä, jotka tarvitsevat ruudunlukuohjelmaa sivuston lukemiseen, koska ruudunlukuohjelmilla on vaikeuksia lukea taulukoiden sisältö. Helpottaakseen sitä, täytyy taulukoille määrittää vähintään otsikkokentät <th></th>-elementeillä sekä lisätä seliteteksti <caption></caption>-elementillä. Otsikkokenttien lisäys WordPressin Gutenberg-editorilla kuitenkin ole mahdollista käyttäen normaalia graafista editoria, vaan otsikkoelementit on lisättävä HTML-editorin avulla. [9]

#### 4.6 Ymmärrettävä sisältö

Verkkosivuston eri sivujen sisällön ymmärrettävyyteen vaikuttaa kaikki edellä mainitut tekniset kohdat, mutta jos sisältö ei ole kirjoitettu niin, että se on kaikkien ymmärrettävässä muodossa, ei sivusto ole saavutettava. Onkin tärkeää, että sivuston kieli yksinkertaista ja ymmärrettävää kieltä. Tällöin mahdollisimman monella on mahdollisuus lukea sivuston sisältöä.

WCAG-ohjeistuksen 3. periaate sanoo, että “Tekstin ei pitäisi vaatia peruskoulun opetuksen vaatimukset ylittävää lukutaitoa. Jos se vaatii, käyttäjälle pitää tarjota täydentävää tai vaihtoehtoista sisältöä.”. Käytännössä tämä tarkoittaa sitä, että sivuston tekstisisältö on selkeää yleiskieltä, jonka ymmärtämistä voidaan lisäksi helpottaa lisäämällä sivulle sisältöä tukevaa kuva-, video- ja äänisisältöä.

## **5 CASE MAINOSTOIMISTO GROTESKI - SAAVUTETTAVAN TEEMAN KEHITYS**

Tässä luvussa tulen esittelemään kuinka Mainostoimisto Groteskin käyttämä oma teema kehitettiin vastaamaan saavutettavuusohjeistuksen vaatimuksia tasolla AA. Tason valinta perustui Suomen lainsäädännön vaatimuksiin. Teeman kehityksessä keskityttiin vain tekniseen näkökulmaan, koska teeman tarkoituksena ei ole sisältää sisältöä, vaan tarjota ainoastaan tekninen alusta tulevia asiakastöitä varten. Kaikki muutokset eivät olleet saavutettavuusohjeiden vaatimusten mukaan välttämättömiä, mutta ne katsottiin kuitenkin tarpeellisiksi. Kehityksen jälkeen käyn läpi teemaan testauksen eri työkaluin sekä pohdin, kuinka testaus onnistui ja mitä tietoa se antoi.

Kohdeyritys on pyrkinyt kehittämään verkkosivuprosessin mahdollisimman tehokkaaksi niin, ettei useasti käytettäviä tyylimäärityksiä tarvitse tehdä joka kerta uudestaan. Lisäksi pitäen tavoitteena, että se tuottaa erittäin laadukkaita sivustoja budjetissaan. Nykyisen teeman kehitys aloitettiin noin kaksi vuotta sitten, jonka aikana sitä käyttäen asiakkaille on toteutettu lähes 100 erilaista WordPress-sivustoa. Ennen oman teeman käyttöä, kohdeyritys valitsi asiakasprojekteihin kolmannen osapuolen teemoja. Ne kuitenkin osoittautuivat liian hankaliksi kehittää, teknisesti ylläpitää, mutta myös asiakkaan puolesta hankaliksi käyttää.

Oma teema on ottanut jo kahden vuoden aikana useita harppauksia eteenpäin mm. responsiivisuuden ja verkkokaupan tyylimääritysten toteutuksessa, mutta saavutettavuutta ei oltu otettu huomioon vielä lainkaan. Kehitetty teema on toteutettu maailmalla erittäin suosittu Underscores-kehysteeman (underscores.me) pohjalta. Underscores-teemasta on kuitenkin päätetty poistaa yrityksen mielestä ylimääräisiä asioita mm. mobiilinavigaation toteutus, joka korvattiin itse toteutetulla ratkaisulla.

Koska Groteski-teema oli toteutettu suosittu kehysteeman päälle, oli oletuksena, että se sisältää jo suoraan kaikki tärkeimmät ja suurimmat saavutettavuuden vaatimukset. Tiedossa kuitenkin oli, että kehysteemaa oli viimeksi päivitetty yli 2 vuotta sitten, joten odotettavissa oli, että joitakin muokkauksia oli tehtävä. Tämän työn kirjoitushetkellä yritys ei ollut vielä toteuttanut saavutettavaa sivustoa, jonka takia teeman sekä prosessin kehitystyö oli tulevaisuutta ajatellen erittäin tärkeää.

## 5.1 Saavutettavuuden toteutus teemaan

Tavallisesti kohdeyritys käyttää teemaa yhdessä WPBakery Builder -lisäosan kanssa, mutta kohdeyritys teki päätöksen, että se käyttää jatkossa vain WordPressin omaa Gutenberg-editoria saavutettavien sivustojen toteutukseen. Saavutettavien sivustojen rakentamisessa Gutenbergin ohella yritys puolestaan tulee käyttämään Advanced Custom Field -lisäosaa, joka helpottaa sivuston ylläpitämistä ja sisällön päivittämisessä sekä mahdollistaa sellaisten elementtien lisäämisen, joka ei Gutenberg-editorilla onnistu. Advanced Custom Fields -lisäosan toiminta perustuu lisäkenttien luontiin mm. sivupohjan mukaan, jotka lisätään vastaaviin sivupohjiin PHP:llä.

Teeman kehittäminen saavutettavaksi alkoi muutoksilla sivupohjien PHP-koodiin. Teema sisälsi sivupohjat sivuille, joilla ei ole sivupalkkia (fullwidth.php) sekä sivupalkilliselle (page.php) sivulle. Lisäksi teemalla oli sivupohjat myös blogin etusivulle (home.php), arkistosivulle (archive.php) sekä yksittäiselle blogisivulle (single.php).

Teeman sivupohjat sisälsivät tärkeimmät saavutettavuuteen liittyvät vaatimukset, kuten toisistaan erottuvat ja tunnistettavat elementit, jotka oli toteutettu jo kehysteemaan. Uutena ominaisuutena teeman sivupohjiin toteutettiin murupolku. Murupolku on joukko navigointilinkkejä, jotka esittävät sivuston käyttäjälle, millä sivulla hän on. Samalla murupolku näyttää polun etusivulta käyttäjän vieraillemalle alasivulle. Murupolku toteutettiin yksinkertaisella PHP-funktiolla, joka sijoitettiin teeman functions.php -tiedostoon. Koska saavutettavuuden kriteeri 2.4.8 vaatii, että käyttäjälle näytetään selkeästi, millä sivulla hän on, murupolku päätettiin lisätä kaikille alasivuille. Puolestaan funktiokutsu, joka lisäsi murupolun tulostuksen sivulle, lisättiin header.php-tiedostoon.

Toinen toteutettu muutos koski mobiilinavigaatiota. Hampurilaisikoniin, jolla avataan mobiilinavigaatio, lisättiin aria-hidden="true" sekä aria-label="Avaa mobiilivalikko" -attribuutit. Aria-hidden:in tarkoituksena on, että se piilottaa hampurilaisikonin, mutta puolestaan aria-label on tarkoitettu ruudunlukuohjelmien luettavaksi. Näin ollen, käyttäjät, jotka käyttävät ruudunlukuohjelmaa, tietävät mistä painaa avatakseen mobiilinavigaation.

Seuraavassa luvussa, testauksen aikana myös huomattiin, että mobiilinavigaation ID-attribuutti ei saa olla sama kuin päänavigaatioissa. Normaalisti monissa toteutuksissa

aikaisemmin käytettiin päänavigaatiossa ja mobiilinnavigaatiossa samaa valikkoa, joka aiheutti saman ID-attribuutin arvon molempiin valikkoihin. Ongelman korjaus onnistui lisäämällä functions.php -tiedostoon toinen navigaationsijainti ja asettamalla siihen mobiilinnavigaatio.

Suurimmat muutokset teemassa kohdistuivat kuitenkin fonttikokoon. Vaikka teema oli toteutettu responsiiviseksi, fonttimääritykset oli suurimmalta osalta tehty pikseliarvoilla. Jo ennen muutosta oli tiedossa, että tämä olisi aiheuttanut ongelmia, kun käyttäjälle annetaan mahdollisuus säätää tekstikokoa sivulla. Yhdenmukaistimme teeman fontin kokomäärityksiä niin, että body-elementin fonttikoko oli 1 rem ja puolestaan kaikkien muiden tekstielementtien kuten otsikoiden kokomääritykset tehtiin käyttäen prosentuaalisia arvoja. Fonttien rem-arvot perustuvat käyttäjän selaimen tai laitteiston määräämään fonttikokoon. Tavallisesti yksi rem vastaa 16 pikseliä, jolloin se sopii hyvin normaalin leipätekstin kooksi.

## **5.2 Saavutettavuuden testaus**

Saavutettavuuden testausta varten toteutettiin yksinkertainen WordPress-sivusto käyttäen kehitettyä Groteski-teemaa. Sisältö ei tässä työssä ollut merkittävää eikä testauksen tarkoitus, siksi työssä päädyttiin käyttämään mainostoimistoyrityksemme omilta sivuilta teksti- ja kuvasisältöä demosivuston toteutukseen. Sivustolle toteutettiin etusivu, tyypillinen sisältösivu sekä yhteys sivu. Näin testistä saatiin riittävän kattava, joka antoi mahdollisimman realistinen ja monipuolinen tulos. Testauksessa käytettiin työkaluissa WCAG 2.0 -version tasoa AA.

Saavutettavuuden testaus ei ole yksinkertainen prosessi, koska mahdollisuuksia testata sivustoa on lukuisia. Testaus voidaan suorittaa käyttäen täysin automaattisia työkaluja, käytännön käyttäjätestausta, asiantuntija-arviota tai puoliautomaattisia työkaluja. Jokaisella tavalla on omat vahvuutensa ja heikkoutensa, jotka on hyvä arvioida testattavan kohteen mukaan. Vaikka automaattisilla ja puoliautomaattisilla työkaluilla on omat rajoitteensa, pidetään niitä kuitenkin tyypillisenä ja riittävänä tapana testata sivuston saavutettavuutta. Tässä tapauksessa, kun testaus kohdistui vain sivuston teemaan, nähtiin tarpeelliseksi käyttää vain puoliautomaattisia työkaluja. [12] [20]

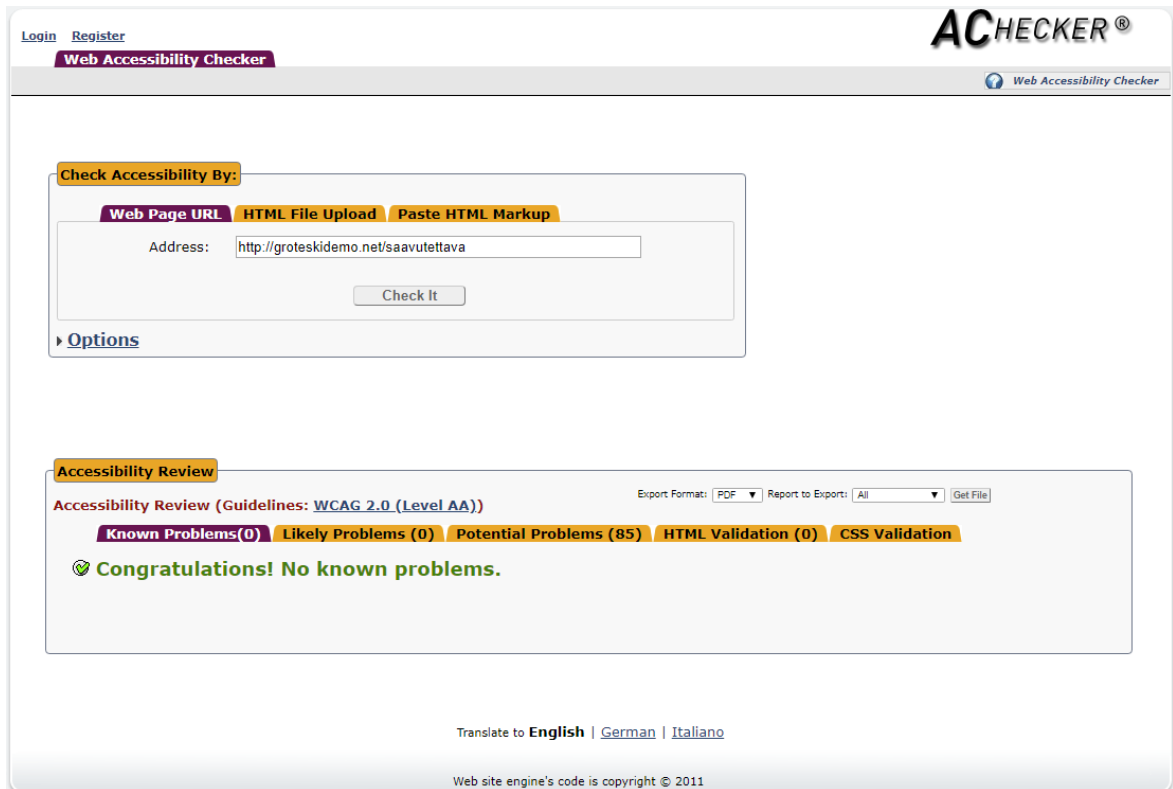
Testaus jaettiin kolmeen osaan. Ensimmäisessä osassa testattiin sivuston teemaa verkosta löytyvillä puoliautomaattisilla palveluilla, jotka testaavat sivuston lähdekoodin ja mm. värien käyttöä. Testauksessa päähuomio kiinnittyi ohjelmien antamiin virheisiin ja varoitukset tai mahdolliset ongelmat vain tarkistettiin. Tämän jälkeen testi suoritettiin käyttäen Mozillan Firefox -selainta, jolla onnistuu sivuston testaus simuloiden näppäimistöselaimista. Lopuksi testattiin, kuinka teema toimii käyttäen ruudunlukuohjelmaa. Ruudunlukuohjelmiksi valittiin Windowsin oma ruudunlukuohjelma sekä NVAccessin NVDA ruudunlukuohjelma. NVDA on ilmainen avoimeen lähdekoodin ruudunlukuohjelma, joka on toteutettu Windows-ympäristöön. Jokaisessa vaiheessa testattiin sivustoa käyttäen edellisessä kohdassa toteutettua saavutettua Groteski-teemaa, mutta myös vanhaa teemaversiota ennen saavutettavuuden toteutusta. [21]

### **5.2.1 Työkalut lähdekoodin testaamiseen**

Teeman testaus suoritettiin kahdella eri verkosta löytyvällä palvelulla, jotka esittelin tässä työssä tarkemmin luvussa 2.5. Ensimmäisenä testasin teeman käyttäen AChecker-työkalua. Testattavan sivuston osoite syötettiin kohtaan Address. Lisäasetuksista testauksen vaadituksi tasoksi valittiin “WCAG 2.0 taso AA” ja testaukseen lisättiin HTML:n validointi. Kuten AChecker-työkalun esittelyssä aiemmin kerroin niin se jakaa tulokset tunnettuihin ongelmiin, todennäköisiin ongelmiin, mahdollisiin ongelmiin ja HTML-validoinnin tuottamiin ongelmiin.

Tunnettuja ongelmia löytyi sivustolta yksi. Mobiilivavigaation id-attribuutti ei ollut uniikki. Tämä ongelma oli onneksi nopea korjata, josta kerroin luvussa 5.1. Korjauksen jälkeen AChecker ei löytänyt enää tunnettuja ongelmia sivustolta. Sama ongelma löytyi myös HTML-validoinnissa, mutta sekin poistui ongelman korjauksen jälkeen. Mahdollisia ongelmia työkalu löysi yhteensä 85 kappaletta. Nimensä mukaisesti nämä kuitenkin olivat vain ohjelman löytämiä arvauksia, jotka voivat aiheuttaa saavutettavuusongelmia sivustolla. Ongelmat liittyivät koristekuvien alt-attribuutin puuttumiseen sekä linkkien tarkoitukseen, kun ohjelma ei tunnista elementtien sisältöä eikä siksi pystynyt itse määrittämään, oliko kuva koristekuva vai tarpeellinen sisällön kannalta.

Alla oleva kuva esittää Achecker-työkalun tarjoamista tuloksista, kun tarvittavat korjaukset teemaan oli toteutettu.



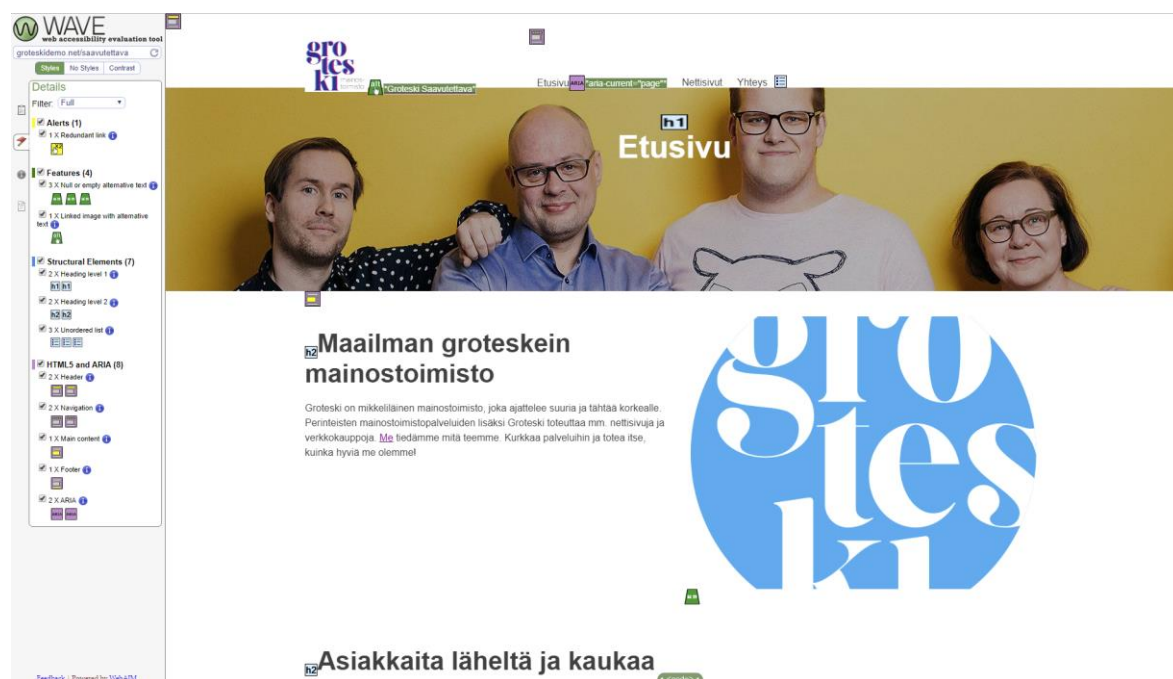
**Kuva 7.** Kuvankaappaus Achecker-työkalun antamista tuloksista

Seuraavaksi testattavan sivuston saavutettavuus testattiin käyttäen Wave-työkalua. Toisin kuin AChecker, Wave-työkalu esittää ongelmat ja varoitukset visuaalisesti sijoittuen virheellisen elementin kohdalle. Wave ei löytänyt sivustolta yhtään teknistä virhettä, mutta kun testaukseen lisättiin värikontrastien tarkastus, löysi työkalu vielä kaksi uutta virhettä. Ensimmäinen virhe oli etusivun pääotsikosta ja sen alla olevan taustakuvasta. Näiden välinen kontrastiero oli liian pieni ja siksi vaikeasti havaittava. Toinen virhe löytyi sivuston mustasta alapalkista, johon oli tulostettu teksti “Webdesign Groteski”. Groteski-teksti oli kirjoitettu punaisella, jolloin sekään ei läpäissyt kontrastin testausta.

Alla olevan kuva esittää Wave-testaustyökalun tuloksia, kun sivustolle oli tehty vaadittavat korjaustoimenpiteet. Tulokset näkyvät ohjelman vasemmassa laidassa, josta voidaan



havaita, ettei virheitä enää löydy. Ohjelma tarjoaa myös tiedon niistä saavutettavuuden ominaisuuksista, jota löytyvät sivulta.



**Kuva 8.** Kuvankaappaus Wave-työkalun antamista tuloksista

## 5.2.2 Näppäimistöseläminen

Mozilla Firefoxilla näppäimistöseläminen on mahdollista, kytkemällä asetuksista päälle kohta “Käytä aina kohdistinpainikkeita selataksesi sivuilla”. Testaustapa ei tarjoa tulosraporttia vaan se on vain puhtaasti tarkoitettu helpottamaan näppäimistökäyttöä, ja jolla on helppoa testata, kuinka sivusto soveltuu näppäimistöllä selattavaksi. Toki käyttäjä, joka ei ole aikaisemmin selannut sivustoa pelkällä näppäimistöllä, ei välttämättä tiedä millaista käyttökokemusta pitäisi odottaa.

Demosivuston testaus suoritettiin lukemalla sen sisältö läpi johdonmukaisessa järjestyksessä. Tärkeintä teeman toteutuksessa näppäimistöseläamisen onnistumiselle on oikein kirjoitettu HTML-koodi. Demosivuston teeman HTML-koodi oli testattu ja todettu toimivaksi jo edellisessä luvussa. Looginen rakenne sivustolla, mutta myös linkkien :focus-valitsen näkyminen, jolloin käyttäjälle kerrotaan selkeästi milloin mikäkin linkki on valittu tabulaattoripainikkeella.

### 5.2.3 Toimivuus ruudunlukuohjelmalla

Sivuston ruudunlukumahdollisuuden testaus suoritettiin ensimmäisenä Windowsin omalla ruudunlukuohjelmalla. Sivuston yksinkertaisen sisällön takia lukeminen tapahtui nopeasti eikä ongelmia juurikaan esiintynyt. Ainoa havaittu virhe esiintyi sivuston kielen määrittämisessä, joka oli jäänyt testattavalle sivustolle englanniksi. Vaikka tämä ei liittynyt itse teeman tekniseen toteutukseen, oli se hyvä muistutus saavutettavuusohjeistuksen kriteeristä 3.1.1. Virhe ei kuitenkaan aiheuttanut käytössä muita ongelmia kuin enintään hämmennyksen. WordPressin hallintapaneelista tämä oli nopeasti muokattavissa suomeksi.

Seuraavaksi testaus suoritettiin NVDA-ruudunlukuohjelmalla, joka nopeasti osoittautui huomattavasti monipuolisemmaksi käyttää kuin Windowsin oma ruudunlukuohjelma. NVDA suosittelee ohjelmansa käyttöön selaimeksi Firefoxia, joka tässäkin testissä valikoitui käytettäväksi selaimeksi. Ohjelman käyttö onnistui näppäimistöllä muutamia näppäinyhdistelmiä käyttäen. Luettavuus oli hyvää ja puheen ohjaaminen näppäimistöllä oli intuitiivista käyttää. Toisin kuin Windowsin oma ohjelma, NVDA-ohjelma kertoi navigaatiopalkin kohdalla millä sivulla käyttäjä oli sekä luetteli suoraan kaikki navigaatiopalkin linkit ja niiden lukumäärän. Tämä varmasti helpottaa sivuston kokonaisuuden hahmottamista ja tärkeän sisällön löytymistä, myös alasivuilta, kun käyttäjä pääsee nopeasti juuri sille sivulle, jolle halusi. [21]

Tälläkään testillä sivuston teemasta ei enää paljastunut uusia ongelmia. Vaikka testauksen tarkoituksena ei ollutkaan ruudunlukuohjelmien vertailu, lyhyen testauksen aikana huomattiin selkeästi, kuinka NVDA-ohjelma oli Windowsin omaa ruudunlukuohjelmaa edellä.

Ruudunlukuohjelmaa varten sisällöllä ja esimerkiksi otsikoiden ja linkkien sisällön selkeydellä voi vaikuttaa huomattavasti ruudunlukemisen käyttökokemukseen ja saavutettavuuteen. Teemaan toteutettu toiminnallisuus, jolla käyttäjän on mahdollista hypätä toistuvien elementtien yli, oli testauksen aikana erittäin hyödyllinen.

### **5.3 Kehitystyön ja testauksen onnistuminen**

Oletus oli, että teeman toteutus saavutettavaksi olisi ollut vaikeampaa, vaikka teknisesti kohdeyrityksen Groteski-teemaan oli panostettu jo useita vuosia, jotta se täyttää HTML:n syntaktiset vaatimukset mahdollisimman hyvin. Systemaattisempaa testausta teemalle ei kuitenkaan aiemmin oltu tehty, joten ennen saavutettavuuden kehitystä ei ollut varmaa, mitä oli odotettavissa.

Teknisesti teemaan saatiin saavutettavuuden kehityksen aikana lähes kaikki tarpeellinen toteutettua. Testauksen aikana kuitenkin paljastui muutamia virheitä, jotka helposti saatiin korjattua teemaan. Vaikka saavutettavuuteen liittyy monia vaatimuksia, niiden muodostaman kokonaisuuden ymmärtäminen ei ollut aivan yksinkertaista. Tämä aiheutti myös tässä työssä ongelmia. Onneksi testaustyökalut paljastivat ongelmia, jotka eivät heti toteutusvaiheessa paljastuneet.

Suoritettujen testausten pohjalta on kuitenkin hankala arvioida, oliko teemalle suoritettavat testit riittävät. Teeman testaamisessa olisi voitu lisäksi hyödyntää puoliautomaattisten testien lisäksi oikeaa käyttäjäryhmää, jolloin todellisen tuloksen saaminen olisi ollut varmempaa ja antanut realistisemmän kuvan. Suoritetut testit kuitenkin osoittivat, että verkosta löytyvillä palveluilla on mahdollista saada suoritettua kaikki tärkeimmät saavutettavuuden tarkastukset, ainakin silloin kun sivuston kohderyhmänä suoraan ei ole käyttäjäryhmät, joilla on rajoitteita.

### **5.4 Tulevaisuus**

Testauksen myötä voidaan sanoa, että Groteskin teema on kehitys- ja testaustyön jälkeen saavutettava. Se ei kuitenkaan tarkoita, että sen kanssa jatkossa kehitettävien sivustojen saavutettavuus olisi aivan mutkatonta. On muistettava, että saavutettavuus on paljon muutakin kuin vain tekninen toteutus. Oikein toteutettu teema sekä muu lähdekoodi auttaa huomattavasti, mutta jos sivuston sisältö on kirjoitettu vaikeaselkoiseksi tai rakenne on hankala hahmottaa, ei teknisellä toteutuksella ole juurikaan väliä. Seuraavaksi yrityksen on mietittävä prosessi, jolla saavutettavia verkkosivustoja suunnitellaan. Kuten luvussa 4.

kirjoitin, ulkoasun suunnittelussa on otettava huomioon monia asioita, jotka vaikuttavat huomattavasti sivuston saavutettavuuteen.

Jatkossa kohdeyrityksen on lisäksi kiinnitettävä huomiota ulkoasun suunnitteluun sekä prosessiin kokonaisuutena, asiantuntijuutta unohtamatta. Saavutettavuus ja siihen liittyvä lainsäädäntö on Suomessa vielä niin tuntematon, ettei kohdeyrityksen asiakkailta itsellään ole riittävä osaamista ja tietoa niiden tuomista vaatimuksista verkkopalveluita kohtaan.

## 6 YHTEENVETO

Työn tavoitteena oli selvittää mitä kaikkea laki digitaalisten palveluiden tarjoamisesta tuo saavutettavuuden osalta verkkosivustojen suunnitteluun, kehittämiseen ja sisällönsyöttöön sekä paneutua tarkemmin saavutettavan WordPress-julkaisujärjestelmän teeman kehitykseen case-esimerkillä.

Verkkosivuston saavutettavuus voi kuulostaa monen korvaan monimutkaiselta ja tuottaa vain ylimääräistä turhaa työtä verkkosivustojen suunnittelijoille, kehittäjille ja sisällöntuottajille. Todellisuudessa saavutettavuudesta jollain tasolla hyötyviä ihmisiä on jo Suomessa lähes miljoona. Siksi tämä askel verkkosivustojen kehittämisessä oli otettava viimeistään nyt lain määrittämällä tavalla, jotta mahdollisimman monella on jatkossa yhdenvertainen mahdollisuus toimia digitaalisessa yhteiskunnassa.

Mainostoimisto Grotoskin käyttämän teeman kehittäminen saavutettavaksi oli toteutuksen puolesta helppo prosessi, koska tuntemus teemaan oli erittäin hyvällä tasolla. Suurimmat haasteet tuli saavutettavuusohjeistuksen tulkinnoista. Kuten aiemmin työssäni mainitsin, kansainvälinen saavutettavuusohjeistus WCAG 2.0 jättää sivuston toteuttajille paljon tulkinnanvaraa, miten missäkin tilanteessa tulisi toimia. Tätä prosessia onneksi auttaa huomattavasti verkosta löytyvät testaustyökalut, joilla sivuston saavutettavuuden onnistumisen voi testata varsin nopeasti. Kuten case-esimerkki todisti, ihmeisiin testaustyökalutkaan eivät kuitenkaan pysty vaan nekin jättävät joitakin mahdollisia ongelmakohtia käyttäjälle itselleen arvioitavaksi. Teknisesti saavutettavuus ei vaadi sivustolta ihmeellisiä asioita, vaan sivuston toteutus on jo pitkällä, jos sivuston HTML on W3C:n ohjeistuksen mukaan toteutettu. Teeman toteuttaminen saavutettavaksi on kuitenkin vasta ensimmäinen askel. Ulkoasun suunnittelu ja sivuston sisällöntuotanto saavutettavaksi vaativat omat keinonsa sekä huomionsa.

## LÄHTEET

1. Mucha, J.M., 2018. Combination of automatic and manual testing for web accessibility (Master's thesis, Universitetet i Agder; University of Agder).
2. Noh, K. R., Jeong, E. S., You, Y. B., Moon, S. J., & Kang, M. B., 2015. A study on the current status and strategies for improvement of web accessibility compliance of public institutions, 1(1), 4.
3. Gibson, M., 2012. Opening the Web for all: inclusive and secure design of an online authentication system.
4. ”Diverse Abilities and Barriers”. [verkkoaineisto]. Saatavilla: <https://www.w3.org/WAI/people-use-web/abilities-barriers/> [viitattu 15.9.2019]
5. Ferri, D. and Favalli, S., 2018. Web Accessibility for People with Disabilities in the European Union: Paving the Road to Social Inclusion. *Societies*, 8(2), p.40.
6. “Saavutettavuusvaatimukset”. [verkkoaineisto]. Saatavilla: <https://www.saavutettavuusvaatimukset.fi/> [viitattu 15.9.2019]
7. Alonso, F., Fuertes, J.L., González, Á.L. and Martínez, L., 2010, April. On the testability of WCAG 2.0 for beginners. In Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A) (p. 9). ACM.
8. “Developer resources”. [verkkoaineisto]. Saatavilla: <https://vm.fi/saavutettavuusdirektiivi> [viitattu 15.9.2019]
9. “Papunet - saavutettavuus”. [verkkoaineisto]. Saatavilla: <http://papunet.net/saavutettavuus> [viitattu 12.9.2019]
10. “Developer resources”. [verkkoaineisto]. Saatavilla: <https://journals.ala.org/index.php/ltr/article/viewFile/4687/5574> [viitattu 22.9.2019]
11. Ivory, M.Y., Mankoff, J. and Le, A., 2003. Using automated tools to improve web site usage by users with diverse abilities.
12. “Screen Reader Testing”. [verkkoaineisto]. Saatavilla: <https://soap.stanford.edu/tips/screen-reader-testing> [viitattu 22.9.2019]

13. Elkabani, I., Hamandi, L., Zantout, R. and Mansi, S., 2015, December. Toward better web accessibility. In 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA) (pp. 1-6). IEEE.
14. "Web Accessibility Checker". [verkkoaineisto]. Saatavilla: <https://achecker.ca/checker/index.php> [viitattu 2.10.2019]
15. "WAVE Web Accessibility Tool". [verkkoaineisto]. Saatavilla: <https://wave.webaim.org/> [viitattu 2.10.2019].
16. M. Zilak, A. Keselj and T. Besjedica, "Accessible Web Prototype Features from Technological Point of View," 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2019, pp. 457-462.
17. "Developer resources". [verkkoaineisto]. Saatavilla: <https://developer.wordpress.org/> [viitattu 2.10.2019].
18. "Wordpress plugins". [verkkoaineisto]. Saatavilla: <https://fi.wordpress.org/plugins/> [viitattu 13.10.2019].
19. Brajnik, G., Yesilada, Y. and Harper, S., 2010, October. Testability and validity of WCAG 2.0: the expertise effect. In Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility (pp. 43-50). ACM.
20. S. Lundqvist ja J. Ström, "Web Accessibility in E-learning : Identifying and Solving Accessibility Issues for WCAG 2.0 Conformance in an E-learning Application", Dissertation, 2018.
21. "Using NVDA to Evaluate Web Accessibility". [verkkoaineisto]. Saatavilla: <https://webaim.org/articles/nvda/> [viitattu 26.10.2019].