

# **Verkkosivustojen saavutettavuus osana palveluiden kehittämistä**

Janne Hyyryläinen

Opinnäytetyö

Marraskuu 2019

Tekniikan ala

Insinööri (AMK), tieto- ja viestintätekniikan tutkinto-ohjelma

Tekijä(t) Hyyryläinen, Janne	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä marraskuu 2019
	Sivumäärä 31	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: X
Työn nimi <b>Verkkosivustojen saavutettavuus osana palveluiden kehittämistä</b>		
Tutkinto-ohjelma Insinööri (AMK), tietotekniikan tutkinto-ohjelma		
Työn ohjaaja(t) Kari Niemi		
Toimeksiantaja(t) Into-Digital		
Tiivistelmä <p>Euroopan parlamentti ja neuvosto julkaisi lokakuussa 2016 direktiivin, jossa asetettiin vähimmäisvaatimukset julkishallinnon digitaalisten palveluiden saavutettavuudelle. Saavutettavuusdirektiivissä vähimmäisvaatimuksien lähtökohtana on käytetty W3C:n Web Content Accessibility Guidelines 2.1 -ohjeistuksen AA-tasoa. Saavutettavuusdirektiiviä noudatetaan portaittain, riippuen palvelun julkaisuajankohdasta ja tyypistä. Ensimmäisten digitaalisten palveluiden määräpäiväksi asetettiin 23.9.2019.</p> <p>Opinnäytetyön toimeksiantajana toimi Into-Digital, jonka tavoitteena oli kehittää palvelujan verkkopalveluiden saavutettavuuden osalta. Tämän tavoitteen saavuttamiseksi Into-Digitalin tavoitteena oli tutustua siihen mitä saavutettavuus pitää sisällään ja kehittää työkalu, jonka avulla voidaan hallita verkkopalveluiden saavutettavuuteen liittyviä projekteja, sekä testata verkkopalveluiden käyttöliittymien saavutettavuutta.</p> <p>Työkalu toteutettiin WordPress sisällönhallintajärjestelmän päälle, joka mahdollisti helpon datan ja käyttäjien hallinnan. Muita työkalussakäytettyjä teknologioita ja palveluita olivat Vue.js, Google Lighthouse, What CMS ja Digital Ocean Dropletit.</p> <p>Verkkosivustojen käyttöliittymien saavutettavuusanalyysien tekemiseksi työkalussa käytettiin Google Lighthouse -analysointia, jota voi käyttää palvelinympäristössä. Toimeksiantajan palvelinrajoitteiden takia, Google Lighthousea piti käyttää erillisellä palvelimella, jota varten käytettiin Digital Oceanin Dropletteja, joiden päälle voitiin asentaa Apache 2 -palvelin.</p> <p>Projektin tuloksena saatiin kehitettyä toimiva työkalu, jota on käytetty saavutettavuusprojektien toteuttamisessa.</p>		
Avainsanat ( <a href="#">asiasanat</a> ) saavutettavuus, Google Lighthouse, WordPress, Vue.js		
Muut tiedot ( <a href="#">salassa pidettävät liitteet</a> )		

Author(s) Hyyryläinen, Janne	Type of publication Bachelor's thesis	Date November 2019 Language of publication: Finnish
	Number of pages 31	Permission for web publication: x
Title of publication <b>Website accessibility as part of developing services</b>		
Degree programme Information and Communications Technology		
Supervisor(s) Niemi, Kari		
Assigned by Into-Digital		
Abstract  <p>In October 2016, the European parliament and council published an accessibility act regarding digital services made by the public administration. Accessibility act uses level AA of the Web Content Accessibility Guidelines 2.1, created by W3C, as a minimum requirement. Accessibility act is enforced step by step, depending on the type of service and the first date was set to 23 October 2019.</p> <p>The thesis was assigned by Into-Digital, whose objective was to improve their services on the accessibility of the web services. To reach this objective, Into-Digital researched accessibility and what it contains and developed a tool to control accessibility projects and to test the accessibility of the web service's user interface.</p> <p>The tool was made on top of the WordPress content management system, which allowed easy data and user management. Other used technologies and services were Vue.js, Google Lighthouse, What CMS and Digital Ocean droplets.</p> <p>To analyze web user interfaces of web services, the tool used Google Lighthouse in a server environment. Because of the assigner's server limitations, Google Lighthouse had to be used in a different server. For that reason, Apache 2 was installed on the Digital Oceans Droplet environment.</p> <p>As a result of the project, a working tool was successfully developed, and it has been used on completing company's accessibility projects.</p>		
Keywords/tags ( <a href="#">subjects</a> ) accessibility, Google Lighthouse, WordPress, Vue.js		
Miscellaneous ( <a href="#">Confidential information</a> )		

## Sisältö

<b>Käsitteet ja lyhenteet.....</b>	<b>3</b>
<b>1 Työn lähtökohdat .....</b>	<b>4</b>
1.1 Toimeksiantaja .....	4
1.2 Tehtävä ja tavoitteet.....	4
<b>2 Verkkopalveluiden saavutettavuus.....</b>	<b>6</b>
2.1 Mitä on saavutettavuus ja mihin se perustuu?.....	6
2.2 Saavutettavuusdirektiivi.....	7
<b>3 Toteutus.....</b>	<b>8</b>
3.1 Projektin aloitus.....	8
3.2 Suunnitelma .....	8
3.2.1 Käytetyt teknologiat .....	8
3.2.2 Projektienhallinnan työkalut.....	9
3.2.3 Datan hallinta.....	10
3.2.4 Testit.....	11
3.3 Tekninen toteutus.....	12
3.3.1 Projektiympäristö ja sen pystyttäminen .....	12
3.3.2 Käyttöliittymä.....	12
3.3.3 Analysoitavan sivuston sisällönhallintajärjestelmän tunnistaminen...	19
3.3.4 Yoast sitemap-tiedoston hakeminen .....	22
3.3.5 Lighthouse API .....	23
3.3.6 Lighthouse datan näyttäminen .....	26
3.4 Testaaminen.....	27
<b>4 Toteutuksen analysointi .....</b>	<b>29</b>
4.1 Toteutuksen hyödyllisyys .....	29

4.2	Jatkokehitysmahdollisuudet.....	29
-----	---------------------------------	----

<b>5</b>	<b>Yhteenveto ja pohdinta .....</b>	<b>30</b>
----------	-------------------------------------	-----------

<b>Lähteet .....</b>	<b>31</b>
----------------------	-----------

## **Kuviot**

Kuvio 1. Projektien listaamiseksi käytettävä taulukko. ....	13
Kuvio 2. Taulukon perustaminen Vue-ohjelmassa.....	14
Kuvio 3. Taulukon näyttämiseksi käytettävä v-client-table-komponentti .....	14
Kuvio 4. Projektien tietojen näyttämiseen käytettävä ACF-lomake.....	16
Kuvio 5. Acf_form funktion käyttäminen .....	17
Kuvio 6. Saavutettavuus analyysien tekemiseen käytettävät työkalut.....	18
Kuvio 7. Saavutettavuustyökalun ilmoitukset .....	19
Kuvio 8. Taulukko saavutettavuusanalyysien listaamiseksi .....	19
Kuvio 9. What CMS rajapintapyyntö .....	21
Kuvio 10. Yoast! -lisäosan luoma sitemap-tiedosto .....	22
Kuvio 11. Sivukarttojen juuritiedoston hakeminen.....	23
Kuvio 12. Esimerkki rajapintaan lähetettävästä objektista.....	26
Kuvio 13. Datan iterointi ja analyysien suorittaminen .....	26
Kuvio 14. JSON-datan tallentaminen tiedostoon.....	26
Kuvio 15. Lighthouse-raporttien datan visualisointi .....	27

## **Taulukot**

Taulukko 1. Esimerkki saavutettavuusprojektien massaluomiseen käytettävästä datasta.....	14
--	----

## Käsitteet ja lyhenteet

ACF	Advanced Custom Fields, WordPress lisäosa, joka mahdollistaa dynaamisten web-komponenttien ohjelmoinnin.
API	Application programming interface, ohjelmointirajapinta.
CMS	Content Management System, sisällönhallintajärjestelmä
Droplet	DigitalOceanin tarjoama virtuaalikone palvelu.
Lighthouse	Google Lighthouse, Googlen tarjoama palvelu, jolla voi testata verkkosivun käyttöliittymän saavutettavuusasteen.
NPM	Node Package Manager
Node.js	JavaScript kirjasto, jolla voidaan toteuttaa mm. palvelimia.
Saavutettavuus	Saavutettavuudella tarkoitetaan digitaalisten palveluiden esteettömyyttä, jolla pyritään mahdollistamaan, että palvelua voi käyttää, vaikka käyttäjällä olisi esimerkiksi fyysisiä rajoitteita.
Toimeksiantaja	Into-Digital
URL, URL-osoite	Uniform Resource Locator, verkko-osoite.
Vue, Vue.js	JavaScript kirjasto, joka mahdollistaa dynaamisen reagoivien web-komponenttien ohjelmoinnin.
WCAG	Web Content Accessibility Guidelines, Verkkosisällön saavutettavuusohjeet.
W3C	The World Wide Web Consortium
Yoast!	WordPress lisäosa, verkkosivun hakukoneoptimointiin.

# 1 Työn lähtökohdat

## 1.1 Toimeksiantaja

Into-Digital on yritys, joka on suuntautunut digitaalisen markkinointiin ja digitaalisten palveluiden kehittämiseen. Into-Digital on harjoittanut toimintaansa vuodesta 2002 lähtien, ja se on toteuttanut tuhansia digimarkkinoinnin projekteja ja on tunnettu erityisesti WordPress osaamisestaan. Into-Digitalin osaamiseen kuuluu myös mobiilisolvellukset, Drupal-toteutukset, sekä digitaalinen mainonta kuten display-mainokset ja verkkosivuilla toimivat bannerit.

Into-Digital on Ida Fram Group -konsernin tytäryhtiö, työllistää noin 15 työntekijää ja toimii kahdessa toimipisteessä, Helsingissä ja Jyväskylässä. Into-Digitalin myynti, projektinjohto ja hallinto toimivat Helsingissä ja tekninen tuotanto Jyväskylässä.

## 1.2 Tehtävä ja tavoitteet

Vuonna 2016 Euroopan parlamentti ja neuvosto julkaisivat direktiivin (EU 2016/2102) julkisen sektorin elinten digitaalisten palveluiden saavutettavuudesta. Saavutettavuusdirektiivin avulla pyritään luomaan verkkopalveluita, joita kuka tahansa voi käyttää mahdollisista fyysisistä rajoitteista, kuten sokeudesta tai puutteellisista motorisista taidoista riippumatta.

Toimeksiantajan tavoitteena oli tarjota asiakkailleen palvelu, jonka avulla asiakkaan verkkopalvelut päivitetään vastaamaan saavutettavuusdirektiivin vaatimuksia. Kasvan tarpeen takia toimeksiantaja tarvitsi sisäisen työkalun saavutettavuusprojektien projektinhallintaan ja verkkosivustojen saavutettavuusasteen määrittämiseksi.

Työkalun avulla toimeksiantajan tulee pystyä lisäämään järjestelmään tarvittaessa kerralla satoja uusia projekteja, tarkastelemaan projektin valmiusastetta, hallitsemaan projektiin liittyviä dokumentteja ja muita lisätietoja sekä määrittämään verkkosivuston käyttöliittymän saavutettavuusaste.

## **Projektienhallinta**

Työkalun yksi tärkeimmistä ominaisuuksista on saavutettavuusprojektien projektinhallinta. Koska digitaalisten palveluiden saavutettavuuteen liittyviä projekteja tulee jatkuvasti lisää, on tärkeää, että niitä pystytään hallitsemaan järkevästi.

Palveluun pitää pystyä lisäämään uusia yksittäisiä asiakaskohtaisia projekteja, ja vaihtoehtoisesti projekteja pitää pystyä luomaan myös Excelistä tuotavalla datalla. Excel-datan tuonti mahdollistaa satojen uusien projektien luonnin kerralla. Tämän ominaisuuden avulla projektipäälliköt voivat koostaa datapaketin esimerkiksi kaikista eduskunnan virastoista ja luoda niille kaikille omat projektit järjestelmään mahdollisimman pienellä vaivalla.

Projekteille on määritelty työvaiheita, joiden avulla projektin eri osapuolet pystyvät kuvaamaan projektin edistymistä ja minimoimaan riskin, että jokin projekti jää pitkäksi aika kesken, koska kukaan ei tiedä, mitä toimenpiteitä projektille on tehty.

Projekteille pitää pystyä lisäämään myös yksi tai useampi myynnin ja tekninen vastuuhenkilö. Nämä henkilöt ovat vastuussa projektien läpiviennistä, ja näiden tietojen avulla tulevaisuudessa pystytään selvittämään, ketkä ovat työskennelleet missäkin projektissa.

Työkalun avulla pitää myös pystyä hallitsemaan projekteihin liittyviä dokumentteja, kuten kirjallisia saavutettavuusanalyseja ja jakamaan projektiin liittyviä lisätietoja kuten millä sisällönhallintajärjestelmällä sivusto on toteutettu.

## **Sivustojen analysointi**

Koska saavutettavuusprojekteja voi tulla lyhyellä aikavälillä useita, on tärkeää, että verkkosivustojen saavutettavuusaste ja puutteet pystytään määrittämään mahdollisimman nopeasti. Tämän takia työkaluun pyritään toteuttamaan mahdollisimman monipuolisesti testejä, joilla voidaan analysoida sivustojen saavutettavuutta ja toimintaa automatisoidusti.

Verkkosivustoille voi tehdä monia erilaisia saavutettavuustestejä, mutta kaikkia ei voi automatisoida, vaan suurin osa testeistä pitää toteuttaa manuaalisesti. Esimerkiksi sivuston tekstin kontrastin taustaväriin nähden voi testata ohjelmallisesti, koska siihen on olemassa yksinkertainen laskukaava. Sivuston navigaation pudotusvalikkojen

toimintaa ei pysty tai edes kannata kokeilla ohjelmallisesti, koska ne toimivat jokaisella sivustolla eri tavalla, minkä vuoksi jokaiselle sivustolle pitäisi tehdä oma ohjelmasen testaamiseksi. Tällaista ohjelmaa ei kuitenkaan kannata toteuttaa, koska navigaation testaaminen manuaalisesti on paljon nopeampaa. Edellä mainitun rajoitteen takia työkaluun ei voi toteuttaa toiminnallisuutta testaavia testejä, vaan automatisoidut testit voivat olla vain sivuston ulkoasuun, sisältöön ja tekniikkaan liittyviä.

Toimeksiantajan kannalta on tärkeää myös, että työkalulla pystyy määrittämään, minkä julkaisujärjestelmän päällä sivusto toimii. Esimerkiksi osa julkisenhallinnon järjestelmistä toimii juuri heitä varten tehdyn julkaisujärjestelmän päällä, mikä saattaa hankaloittaa sivustolle tehtävien korjaustoimenpiteiden toteuttamista.

## **2 Verkkopalveluiden saavutettavuus**

### **2.1 Mitä on saavutettavuus ja mihin se perustuu?**

Eficoden Digipalveluiden saavutettavuusopas (2018, 4) mukaan saavutettavuus tarkoittaa, että kaikki voivat käyttää digipalvelua rippumatta henkilön ominaisuuksista tai toimintarajoitteista, kuten iästä tai vammoista.

Suurella osalla ihmisistä on jokin sairaus tai toimintarajoite, joka voi hankaloittaa tietokoneen ja digitaalisten palveluiden käyttöä. Näitä rajoitteita voivat olla esimerkiksi näkövammot kuten värisokeus, joka voi hankaloittaa tekstin lukemista värikkäillä verkkosivuilla, ihmisen motorisia kykyjä rajoittavat neurologiset sairaudet kuten Parkinsonin tauti, joka voi aiheuttaa käsien vapinaa, jolloin hiiren tarkka käyttäminen voi olla hankalaa, tai Down oireyhtymä, jonka oireet voivat aiheuttaa muun muassa oppimisvaikeuksia.

Koska verkon käyttöä hankaloittavia tekijöitä on paljon erilaisia, Euroopan parlamentti ja neuvosto julkaisivat vuonna 2016 saavutettavuusdirektiivin, jonka tarkoituksena on kehittää kaikista julkishallinnon digitaalisista palveluista sellaisia, että kuka tahansa voi käyttää niitä rajoitteistaan huolimatta.

Digitaalisten palveluiden saavutettavuuden määrittämisessä käytetään W3C:n (World Wide Web Consortium) kehittämää WCAG-standardia (Web Content Accessibility

Guidelines). WCAG:n ohjeiden noudattaminen kehittää digitaalisista palveluista saavutettavampia ihmisille, joilla on rajoitteita, kuten sokeus ja heikko näkö, kuurous ja heikko kuulo, liikunta- ja puherajoitteita, valoherkkyyttä ja oppimisvaikeuksia ja kognitiivisia rajoitteita (Campbell, Cooper, Kirkpatrick & O Connor 2018). WCAG koostuu kolmesta eri tasosta: A, AA ja AAA. Saavutettavuusdirektiiviä koskevien sivustojen tulee noudattaa tasojen A ja AA vaatimuksia (Saavutettavuus n.d.). WCAG:n ohjeissa otetaan kantaa muun muassa tekstin kontrastiin sen taustaelementin väriin nähden, tekstin kokoon, sivuston navigoitavuuteen eri päätelaitteilla ja siihen miten eri mediamuotoja (esimerkiksi kuvat ja videot) tulisi sivustolla esittää käyttäjälle.

WCAG:n tasot A, AA ja AAA pyrkivät kuvamaan kuinka hankalaa kyseisen tason täyttäminen kyseisessä aiheessa on. Esimerkiksi tekstielementtien vähimmäiskontrastisuhde niiden taustaväriin nähden tasolla AA on 4.1 ja tasolla AAA vähimmäiskontrastisuhde on 7 (Contrast and Color Accessibility 2018). Värien kontrastisuhde lasketaan värien kirkkautena, eli luminanssina, toisiinsa nähden. Valkoisen tekstin kontrastisuhde valkoisella taustalla on 1 ja mustan tekstin kontrastisuhde on 21.

## 2.2 Saavutettavuusdirektiivi

Euroopan parlamentti ja neuvosto julkaisivat 26. lokakuuta 2019, saavutettavuusdirektiiviin, joka määrää EU-maiden julkishallinnon elimiä tekemään digitaalisista palveluistaan saavutettavia, jotta mahdollisimman monilla olisi mahdollisuus käyttää niitä rajoitteistaan huolimatta (EN 2016/2102). Saavutettavuus direktiivin tavoitteena on myös parantaa digitaalisten palveluiden yleistä laatua ja asettaa niille minimivaatimukset, jotka niiden tulee saavuttaa (Saavutettavuusdirektiivi edistää yhdenvertaisuutta N.d.). Saavutettavuusdirektiivi ei koske pelkästään verkkosivustoja vaan myös aikasidonnaista mediaa, kuten videoita, äänitteitä ja podcasteja (Saavutettavuuslaki on täällä tänään! 2019). Saavutettavuusdirektiivi koskee muun muassa kuntia, kouluja ja virastoja. Saavutettavuusdirektiivin vähimmäisvaatimuksina pidetään W3C:n Web Content Accessibility Guideline -ohjeistuksen (WCAG) AA-tasoa.

Saavutettavuusdirektiiviä noudatetaan porrastetusti riippuen palvelun tyyplistä ja alkuperäisestä julkaisuajankohdasta. Mobiilipalveluiden tulee olla saavutettavia 23.6.2021 mennessä, ennen 23.9.2018 julkaistujen verkkopalveluiden 23.9.2020

mennessä ja 23.9.2018 jälkeen julkaistujen verkkopalveluiden 23.9.2019 mennessä. Uuden aikasidonnaisen median tulee olla saavutettavaa 23.9.2020 lähtien, mutta vanhoja medioita ei tarvitse muuttaa saavutettaviksi. Aikasidonnaiset mediat kuten videot voidaan muuttaa saavutettaviksi esimerkiksi tekstityksillä ja viittomakielisillä käännöksillä.

## 3 Toteutus

### 3.1 Projektin aloitus

Ennen työkalun ominaisuuksien määrittelyä työkalusta oli toteutettu demosovellus Node.js teknologialla. Demossa työkaluun pystyttiin lisäämään tekstikenttiä, joihin syöttämällä verkko-osoitteita kyseiset osoitteet lähetettiin Node.js:n päällä toimivalle palvelimelle. Palvelimelle oli asennettu NPM-paketinhallinnasta saatu Google Lighthouse. Palvelimelle lähetetyt osoitteet ajettiin Lighthousen läpi, joka suoritti saavutettavuusanalyysin osoitteille ja tallensi jokaisesta sivusta raportin käyttäjän tietokoneelle.

Kun työkalusta päätettiin toteuttaa tuotannossa käytettävä versio, toimeksiantajan kanssa määriteltiin, mitä ominaisuuksia työkalussa tulisi olla (ks. luku 1.2). Tämän määrittelyn perusteella työkalun käyttöliittymästä piirrettiin rautalankamallit. Kun toimeksiantaja oli kommentoinut rautalankoja, näiden rautalankojen pohjalta toteutettiin sovelluksen alustava käyttöliittymä. Projektia varten Trello-palveluun pystytettiin sille oma seinä, johon kaikki projektin osapuolet pystyivät lisäämään toivomuksia työkalun ominaisuuksista. Trello-seinä toimi projektin aikana myös projektin hallintaan käytettävänä kanban-työkaluna.

### 3.2 Suunnitelma

#### 3.2.1 Käytetyt teknologiat

Projektinhallintaan liittyvien ominaisuuksien takia työkalu päätettiin toteuttaa WordPress-julkaisujärjestelmän päälle koska WordPress tarjoaa kattavat ominaisuudet työkalun sisällön- ja käyttäjähallinnan toteuttamiseksi. Jos työkalu olisi päätetty

toteuttaa Node.js-alustalle kuten alkuperäinen demo (ks. luku 3.1), monet työkalun ominaisuuksista olisi pitänyt ohjelmoida itse alusta asti, mikä olisi pidentänyt työkalun kehittämiseen kulunutta aikaa.

Työkalun käyttöliittymän toteuttamiseen valittiin ACF, joka mahdollistaa helpon datan syötön ja dynaamisten verkkosivujen rakentamisen, sekä Vue.js, jolla datan käsittely ja muotoilu reaaliajassa on todella yksinkertaista.

Verkkosivustojen julkaisujärjestelmien tunnistamista varten käytettiin WhatCMS-palvelun tarjoamaa ilmaista rajapintaa, joka mahdollistaa tuhat API-pyyntöä kuukaudessa.

Toimeksiantajan palvelintarjoajan rajoitteiden takia palvelimelle, jossa työkalu sijaitsee, ei voitu asentaa demosovelluksessa käytettyä NPM-pakettina saatua Google Lighthousea. Tämän takia analyysit suoritetaan erillisellä DigitalOceanin tarjoamalla droplet-virtuaalipalvelimella. Koska palvelimella ei ole graafista käyttöliittymää ja Lighthouse tarvitsee Google Chrome -selaimen toimiakseen, palvelimelle asennettiin Headless Chrome, joka voidaan käynnistää ilman käyttöliittymää.

### 3.2.2 Projektienhallinnan työkalut

Yksi työkalun ominaisuuksista on mahdollisuus seurata saavutettavuusprojektien etenemistä ja arkistoida niihin liittyviä tietoja. Projektien toteutuksen aikana sille voidaan määrittää eri valmiusasteita, joiden avulla pystytään seuraamaan, milloin projektia on viimeksi tehty ja missä työvaiheessa sen edistyminen on. Työkaluun voidaan myös kirjata projektin etenemisen kannalta tärkeät vastuuhenkilöt, kuten myynnin vastuuhenkilö ja mahdollisten korjauksien toteuttaja. Näiden henkilötietojen tarkoituksena on säilyttää tieto siitä, ketkä projektia ovat tehneet, jotta tulevaisuudessa heiltä voi kysyä projektiin liittyviä tietoja, jos niitä ei muualta löydy.

Työkalussa pystytään säilyttämään muitakin projektiin liittyviä tietoja, kuten projekteista annetut kustannusarviot ja asiakkaille lähetetyt koosteet saavutettavuustestien tuloksista. Näiden tietojen avulla projektin toteuttaja esimerkiksi näkee helposti, mitä kaikkea testatuissa sivustoissa pitää korjata ilman, että hänen täytyy käydä läpi kaikkia saavutettavuustestien tuloksia. Projekteille on myös tekstikenttä, johon voi tarvittaessa jättää muita projektiin liittyviä huomioita tai ohjeita.

### 3.2.3 Datan hallinta

Saavutettavuustyökalun datan ja käyttäjien hallitsemiseksi, työkalu toteutettiin WordPress sisällönhallintajärjestelmän päälle. Jokaista omaa projektia varten WordPressissä voitiin luoda oma artikkeli, jonka sisälle voi tallentaa kaikki projektia koskevat tiedot, kuten analyysit ja projektin parissa työskennelleet henkilöt.

Työkalun projektien listaussivun käyttöliittymän luomisessa käytettiin Vue.js JavaScript-kirjastoa, jota varten WordPressiin toteutettiin suljettu rajapinta, jonka kautta JavaScriptillä voitiin noutaa lista kaikista järjestelmän projekteista ja rajoitettu osa niiden tiedoista, kuten millä sisällönhallintajärjestelmällä ne on toteutettu.

Saavutettavuusprojektien datan tallentamiseksi käytettiin Advanced Custom Fields -lisäosaa (ACF), joka mahdollistaa lisäkenttien luomisen WordPressin hallintapaneeliin. Lisäosalla hallintapaneeliin voidaan luoda esimerkiksi kenttä verkkosivuston osoitteelle. ACF:n `acf_form()`-funktio mahdollistaa hallintapaneelien kenttien ja niiden sisällön näyttämisen, sekä käyttöliittymässä. Tätä funktiota käyttämällä pystyttiin toteuttamaan osa yksittäisen projektin sivusta.

Kun käyttäjä haluaa analysoida verkkosivuston, pyyntö analysoinnin suorittamiseksi toteutetaan WordPressin rajapinnassa. Rajapinta noutaa PHP:lla projektiin tallennetut verkko-osoitteet ja lähettää ne ulkoisella palvelimella toimivaan Lighthouse-rajapintaan. Lighthouse-rajapinta palauttaa WordPressin rajapinnalle JSON-objekteja, jotka WordPressin rajapinta tallentaa JSON-tiedostoiksi, jotta analysoitu data on myös tulevaisuudessa käsiteltävissä. Lighthouse-rajapinnan kutsu päätettiin toteuttaa WordPressin rajapinnassa, jotta saavutettavuustyökalun käyttöliittymä voidaan tarvittaessa sulkea, vaikka analyysi olisi vielä kesken. Jos datan tallentaminen toteutettaisiin käyttöliittymätasolla, mutta käyttöliittymä on suljettu, dataa ei voida tallentaa ja analyysit joudutaan suorittamaan uudelleen seuraavalla käyttökerralla. Tallennettu data haetaan tarvittaessa JavaScriptillä käyttöliittymätasolle ja näytetään saavutettavuustyökaluun toteutetulla Lighthouse-selaimella.

### 3.2.4 Testit

Työkalun päätarkoitus on helpottaa sivustojen saavutettavuusanalyysien tekemistä. Analyysien ja tulevien kustannusarvioiden kannalta on tärkeää tietää, millä sisällönhallintajärjestelmällä analysoitavat sivustot on toteutettu. Sivuston sisällönhallintajärjestelmän pystyy selvittämään WhatCMS-palvelulla. Palveluun syötetään sivuston URL-osoite, jolloin palvelu palauttaa tiedon käytetystä sisällönhallintajärjestelmästä. WhatCMS tarjoaa sovelluskehittäjille rajapinnan, mihin kehittäjät pystyvät lähettämään useita osoitteita kerralla. WhatCMS tarjoaa kehittäjille tuhat ilmaista rajapintapyyntöä kuukaudessa, ja jos kehittäjä on valmis maksamaan palvelun käyttämisestä, hän voi lähettää niitä enemmän ja aikaväli pyyntöjen välillä on lyhyempi. Ilmaisella tilillä pyyntöjä voi lähettää kerran kymmenessä sekunnissa. Tämän rajoituksen takia saavutettavuustyökalulle voidaan ohjelmoida CRON-ajastin, joka lähettää tarvittaessa pyynnön WhatCMS:n rajapintaan sallituin väliajoin.

Verkkosivustojen saavutettavuusasteen määrittämiseksi voidaan käyttää Googlen tarjoamaa Lighthouse-järjestelmää. Lighthouse analysoi annetusta URL-osoitteesta löytyneen verkkosivun ja palauttaa käyttäjälleen analyysin, jossa kerrotaan muun muassa sivuston suorituskyvystä, saavutettavuusvajeista, kuten liian matalasta kontrastista tai liian pienestä fontista, sekä hakukoneoptimointiin liittyvistä puutteista. Lighthouseella voidaan siis selvittää muitakin kuin saavutettavuuteen liittyviä asioita. Vaikka monet Lighousen palauttamista tiedoista eivät ole saavutettavuuden kannalta oleellisia, päätettiin toimeksiantajan kanssa, että niitä ei kannata suodattaa, sillä tiedoilla voi mahdollisesti olla tulevaisuudessa käyttöä.

Tavallisin tapa käyttää Lighthousea on Google Chrome-selaimessa, mutta selaimessa voi analysoida ainoastaan auki olevalla välilehdellä olevan verkkosivun. Eli selaimessa ei voi analysoida koko verkkosivustoa kerralla. Tämän rajoitteen takia työkalussa käytetään sovelluskehittäjille tarkoitettua Lighthouse versiota, jonka voi suorittaa ohjelmallisesti tietokoneen komentorivillä. Koska Lighthousea ajetaan ohjelmallisesti, voidaan luoda sovellus, joka käy läpi listan annettuja URL-osoitteita ja suorittaa Lighthouse-analyysit niille automatisoidusti, minkä jälkeen se palauttaa saadun datan joko HTML- tai JSON-formaatissa.

Saavutettavuustyökalulla verkkosivujen analysoinnin tulisi olla mahdollisimman vai-  
vatonta. Koska monilla verkkosivustoilla on kymmeniä tai jopa satoja alasivuja, kaik-  
kien tarvittavien URL-osoitteiden kerääminen Lighthouse-analyysiä varten käsin voi  
olla todella aikaa vievää. Tämän takia työkaluun pitää toteuttaa testi, joka yrittää ke-  
rätä nämä osoitteet käyttäjän puolesta. Helpoin tapa kaikkien osoitteiden löytä-  
miseksi on etsiä sivustolta sivukartta. Sivukartan tarkoituksena on näyttää kaikki  
verkkosivuston sivut ja miten niihin pääsee. Suurin osa maailman verkkosivustoista,  
jotka on toteutettu sisällönhallintajärjestelmää käyttäen, käyttää WordPress-sisällön-  
hallintajärjestelmää (WordPress market share N.d.) ja monet moderneista  
WordPress-sivustoista käyttää YoastSEO-lisäosaa hakukoneoptimointiin. YoastSEO-  
luo sivustolle automaattisest in sivukartan, jota voidaan käyttää sivujen URL-  
osoitteiden keräämiseksi.

### 3.3 Tekninen toteutus

#### 3.3.1 Projektiympäristö ja sen pystyttäminen

Saavutettavuustyökalu päätettiin toteuttaa WordPress-sisällönhallintajärjestelmän  
päälle, koska tällöin analysoidun datan säilöminen olisi mahdollisimman helppoa ja  
nopeaa. WordPressin-teemapohjassa käytetään apuna Gulp.js -JavaScript-kirjastoa,  
jonka päälle voidaan rakentaa automatisoitu ympäristö, joka kääntää teknologioita  
selaimille luettaviin muotoihin. Esimerkiksi SASS-tyylimääritykset kääntyvät selaimien  
ymmärtämään CSS muotoon. Projektiympäristössä käytetään myös edellä mainittuja  
SASS-tyylimäärityksiä, koska niiden rakenne on helpommin luettavissa ja se tarjoaa  
monia funktioita, joita tavallinen CSS ei sisällä. Projektipohjassa on myös käytetty  
NPM-paketinhallintajärjestelmästä saatuja JavaScript-paketteja kuten Vue.js, Ele-  
ment.io -käyttöliittymäelementit ja Moment.js, jota käytetään päivämäärien syntak-  
sien muokkaamiseen.

#### 3.3.2 Käyttöliittymä

Työkalun käyttöliittymä koostuu kolmesta päänäkymästä. Projektien listaussivusta,  
yksittäisen projektin sivusta ja Lighthouse-raporttien lukemiseen käytettävästä työka-

lusta. Projekteja varten WordPressiin lisättiin oma postityyppi, projektit, johon jokainen saavutettavuus projekti voidaan tallentaa omana artikkelina. Kun jokaisella projektilla on oma artikkeli, niiden tiedot pystytään tarvittaessa hakemaan helposti projektin post\_id:n avulla. Työkalun käyttöliittymässä on pyritty käyttämään mahdollisimman yksinkertaista värimaalimaa käytettävyyden ja työkalun saavutettavuuden parantamiseksi.

## Projektien listaussivu

Koska työkaluun on tarkoitus voida lisätä satoja eri saavutettavuusprojekteja, projektien listaamiseksi mahdollisimman tehokkaasti päätettiin käyttää taulukkoa (ks. kuvio 1). Taulukossa näytetään projektin tunnistamiseksi tarvittavat tiedot: muokauspäivä, projektin nimi ja projektin vastuuhenkilöt. Näiden tietojen lisäksi taulukossa näytetään projektin status. Taulukossa projekteja voi hakea kaikilla taulukon tiedoilla, projektit voidaan järjestää aakkos- ja päiväjärjestykseen ja kaikki projektit voidaan tarvittaessa tallentaa excel-tiedostoon, jotta niitä voi käyttää myöhemmin muissa projekteissa.

+ Uusi projekti

### Projektit

10 ▾

Filter CMS ▾
Filter Status ▾
📄
+ Tuo projekteja

Muokattu	Projektin nimi	Projektin vastuuhenkilöt	CMS	Status
26.03.2019	Testi projekti 2	Aapo Mäki, Janne Hyyryläinen		Odottaa laajaa raporttia
26.03.2019	Testi projekti			Odottaa automaattitestiä
29.11.2018	Into-Digital			Odottaa automaattitestiä
19.11.2018	New project test			Odottaa automaattitestiä
19.11.2018	Test	Aapo Mäki	WordPress	Odottaa automaattitestiä
19.11.2018	Test 2	Aapo Mäki	WordPress	Odottaa automaattitestiä
19.11.2018	Test 2	Aapo Mäki	WordPress	Odottaa automaattitestiä
19.11.2018	Test	Aapo Mäki	WordPress	Odottaa automaattitestiä
16.11.2018	Testi	Aapo Mäki	Drupal	Odottaa automaattitestiä
09.11.2018	Svenska handelshögskolan	Aapo Mäki	Drupal	Odottaa automaattitestiä

<< < 1 2 3 4 5 > >>

Showing 1 to 10 of 406 records

Kuvio 1. Projektien listaamiseksi käytettävä taulukko.

Taulukon toteuttamiseen käytettiin Vue Tables 2 - JavaScript kirjastoa. Kyseinen kirjasto mahdollistaa reaaliajassa päivittyvien taulukoiden rakentamisen ja se tarjoaa työkalut datan suodattamiseksi. Taulukon perustamiseksi Vue-ohjelmaan määritellään taulukoihin taulukon kolumnit, data kolumnien mukaisena avainarvo pareina ja taulukon asetukset, kuten mitä filttoreitä halutaan aktivoida tai kuinka monta projektia näytetään yhdellä sivulla (ks. kuvio 2).

```
new Vue({
  el: "#data",
  data: {
    columns: ['edit_date', 'project_name', 'project_personnel', 'project_status'],
    tableData: [
      {
        edit_date: "25.3.2019",
        project_name: "Projekti X",
        project_personnel: "Janne Hyyryläinen",
        project_status: "Odottaa automaattitestiä"
      }
    ],
    options: {
      perPage: 10
    }
  }
});
```

#### Kuvio 2. Taulukon perustaminen Vue-ohjelmassa

Kun taulukko on perustettu Vue-ohjelmassa, se voidaan näyttää käyttöliittymässä käyttämällä v-client-table-komponenttia (ks. kuvio 3). Komponentille kerrotaan mitä kolumneja, datataulukkoa ja asetuksia sen tulee käyttää.

```
<div id="data">
  <v-client-table :data="tableData" :columns="columns"
    :options="options"></v-client-table>
</div>
```

#### Kuvio 3. Taulukon näyttämiseksi käytettävä v-client-table-komponentti

Projektien listaussivulla on myös työkalu, jota voidaan käyttää useiden projektien luomiseksi yhdellä kerralla (ks. taulukko 1). Työkaluun syötetään projektien nimet ja verkko-osoitteet, josta analysoitavat sivustot löytyvät.

Taulukko 1. Esimerkki saavutettavuusprojektien massaluomiseen käytettävästä datasta.

site_name	site_url
Into-Digital	<a href="https://into-digital.fi/">https://into-digital.fi/</a>
JAMK	<a href="https://jamk.fi/">https://jamk.fi/</a>

Työkaluun syötetty data muutetaan ohjelmallisesti JSON-formaattiin. Tämän jälkeen data lähetetään AJAX:lla WordPressin Rest API:iin. Rest API:ssa data tallennetaan WordPressin tietokantaan. Kun projektit on lisätty WordPressiin, ne haetaan automaattisesti AJAX:lla ja tallennetaan Vue-ohjelman tableData-tauluun (ks. kuvio 2). Koska projektien listaamiseksi käytettävä taulukko on toteutettu Vue.js:llä, Vue tunnistaa automaattisesti muutoksen tableData-taulussa ja osaa päivittää sen käyttöliittymään ilman että käyttäjän tarvitsee erikseen päivittää sivua.

### **Yksittäisen projektin sivu**

Yksittäisen projektin sivu koostuu kolmesta osiosta: projektin tiedoista, sivuston analysointiin käytettävistä työkaluista ja Lighthouse-analyseista saaduista raporteista.

Projektin tiedoista näkee projektin nimen, statuksen, vastuuhenkilöt, verkko-osoitteen, sivuston sisällönhallintajärjestelmän, lisätiedot ja projektiin liittyvät liitetiedostot (ks. kuvio 4). Näiden tietojen tallentamista varten WordPressiin asennettiin ACF-lisäosa, joka mahdollistaa omien sisältökenttien lisäämisen WordPressin hallintapaneliin. Luomalla kaikille tiedoille oman kentän, tiedot pysyvät helposti luettavissa ja niihin pääsee helposti käsiksi ohjelmallisesti kentän nimellä. Tietojen näyttämiseksi käyttöliittymässä käytettiin ACF:n `acf_form`-funktioita, jolla halutut kentät voidaan tulostaa lomakkeena käyttöliittymään. Lomakkeen kentät sisältävät automaattisesti niihin tallennetut tiedot ja päivitettyt tiedot tallentuvat automaattisesti projektin artikkeliin, kun lomake tallennetaan.

## Into-Digital

Title \*

Into-Digital

### Projektin tiedot

#### Status

- Odottaa automaattisesti
- Automaattisesti tehty
- Asiakas kontaktoitu
- Odottaa laajaa raporttia
- Laaja raportti ja kustannusarvio tehty
- Odottaa korjauksia
- Muutokset työnalla
- Sivusto korjattu

#### Sivuston URL

Tätä osoitetta käytetään sisällönhallintajärjestelmän löytämiseen.

https://into-digital.fi

#### Sisällönhallinta järjestelmä

#### Myyntin vastuuhenkilö

Aapo Mäki

#### Tekninen vastuuhenkilö

Janne Hyryläinen

#### Lisätiedot


Haluatko kertoa jotain projektiin liittyviä huomioita. Tee se tähän.

 Lisää media

**B** *I* U **“** **”** **≡** **≡** **≡** **≡** **≡** **↶** **↷** **🔗** **✕**

### Tiedostot

Tähän voit ladata esimerkiksi asiakkaille meneviä tiedostoja, kuten ilmaiset analyysit.

Tiedosto	
1	 <b>into-digital-saavutettavuus-analyysi</b> Tiedoston nimi: into-digital-saavutettavuus-analyysi.docx Tiedoston koko: 17 KB

Lisää rivi

Päivitä

Poista projekti

Kuvio 4. Projektien tietojen näyttämiseen käytettävä ACF-lomake

Halutut ACF-kenttäryhmät pystytään näyttämään antamalla field\_groups-objektin arvoksi taulukon, joka sisältää kenttäryhmien uniikit id-arvot (ks. kuvio 5). Lomakkeelle

voidaan antaa myös muita asetuksia kuten lomakkeen lähettämäksi tarkoitettun painikkeen tekstin ja lähettämisen jälkeen saatavan vahvistus viestin. Käyttöliittymässä näytettävät arvot kuten submit\_value voidaan kääriä WordPressin \_\_() funktioon (ks. kuvio 5), joka mahdollistaa näiden arvojen kieliversioinnin esimerkiksi WordPress Multilingual -lisäosalla.

```
<?php
$settings = array(
    'field_groups' => array('group_5bc9c365dc23d'),
    'post_title' => true,
    'submit_value' => __('Päivitä', 'ID'),
    'updated_message' => __('Projekti päivitetty <span class="close-notification"></span>', 'ID'),
);
acf_form( $settings );
?>
```

### Kuvio 5. Acf\_form funktion käyttäminen

Yksittäisen projektin sivu sisältää myös saavutettavuusanalyysien tekemiseen käytettävät työkalut: sisällönhallintajärjestelmän tunnistaminen ja sivukartan hakutyökalun (ks. luku 3.3.3), sekä Lighthouse-analysaattorin (ks. luku 3.3.5).

Sisällönhallintajärjestelmän tunnistamiseksi käytetään What CMS -palvelua, johon voidaan lähettää sivuston verkko-osoite ja What CMS palauttaa käytetyn alustan nimen. Jos tunnistettu sisällönhallintajärjestelmä on WordPress, sivukartan hakutyökalua voidaan käyttää Yoast! -lisäosan luomien sivukarttojen etsimiseksi. Jos nämä sivukartat löytyvät, järjestelmä palauttaa automaattisesti Lighthouse-työkalulle joitain sivustolta löytyviä osoitteita analysoitavaksi. Tällä prosessilla voidaan välttää paljon aikaa vievä osoitteiden manuaalinen kerääminen.

Lighthouse-työkalun käyttöliittymä on toteutettu käyttäen Vue.js (ks. kuvio 6). Käyttöliittymän tekstikenttiin syötetään analysoitavat verkko-osoitteet. Uusia kenttiä voidaan lisätä tarvittaessa painalla Lisää osoite -painiketta. Kun kaikki halutut osoitteet on syötetty, kenttien yläpuolelta pitää valita suoritettavat testit. Tämän jälkeen kaikki sivut voidaan analysoida painamalla Auditori-painiketta tai vaihtoehtoisesti yksittäinen osoite voidaan analysoida painamalla tekstikentän vieressä olevaa painiketta. Jos käyttäjä ei ole valinnut vähintään yhtä testiä, hänelle näytetään virheilmoitus (ks. kuvio 7). Jos käyttäjä on valinnut jonkin analyysin, hänelle näytetään ilmoitus analyysin alkamisesta ja toinen ilmoitus, kun analyysi on valmis. Kaikki käyttöliittymän ilmoituk-

set on toteutettu Elements.io Vue.js-kirjastolla. Kun analyysi on valmis, käyttäjälle lähetetään myös sähköposti-ilmoitus, joka sisältää linkin projektisivulle. Analysoinnin aloittaminen tallentaa kaikki analysoitavat osoitteet järjestelmään.

## Testit

### Sisällönhallintajärjestelmä

Sisällönhallintajärjestelmän tunnistamiseen käytetään projektin tiedoissa annettua osoitetta.

**Tunnista sisällönhallinta**

### Sivukartta

Toimii tällä hetkellä vain WP:ssä, jossa on Yoast!

**Etsi sivukarttaa**

### Lighthouse

- Accessibility
- Performance
- Progressive web app
- Best practices
- SEO

### Sivuston osoitteet

1.
2.

**Lisää osoite**

**Auditoi**



\*Auditoinnin päätyttyä saat siitä ilmoituksen sähköpostiisi.

Kuvio 6. Saavutettavuus analyysien tekemiseen käytettävät työkalut



Kuvio 7. Saavutettavuustyökalun ilmoitukset

Auditoi-painikkeen vieressä on osoitteiden kopioimiseen tarkoitettu painike, jota painamalla kaikki osoitteet voidaan kopioida tietokoneen leikepöydälle. Kaikille painikkeille, joilla ei ole tekstiä, on toteutettu työkaluvinkki Element.io Vue.js-kirjastolla. Vinkki näytetään, kun käyttäjä vie hiiren painikkeen päälle.

Kun verkkosivustolle on tehty analyysit Lighthouse-analysaattorilla, niiden sivukohtaiset raportit listataan linkkeinä taulukkoon (ks. kuvio 8). Koska Lighthouse-analysaattori tallentaa raportit JSON-muotoisina, linkit ohjaavat raporttien lukemiseen tarkoitettuun työkaluun (ks. luku 3.3.6). Taulukosta löytyy myös painikkeet linkin vierestä raporttien poistamiseksi.

Lighthouse raportit	Poista
<a href="#">.json</a>	✘
<a href="#">https-into-digital-fi-.json</a>	✘
<a href="#">https-into-digital-fi-palvelut-.json</a>	✘
<a href="#">https-into-digital-fi-palvelut-verkkosivut-.json</a>	✘

Kuvio 8. Taulukko saavutettavuusanalyysien listaamiseksi

### 3.3.3 Analysoitavan sivuston sisällönhallintajärjestelmän tunnistaminen

Analysoitavan sivuston sisällönhallintajärjestelmän tunnistaminen on jokaisen saavutettavuus projektin kannalta tärkeää. Kun järjestelmä on tunnistettu, asiakkaalle pystytään antamaan tarkempi arvio mahdollisten korjaustöiden toteuttamiselle. Mitä

paremmin käytetty järjestelmä tunnetaan, sitä nopeammin puutteet voidaan korjata, koska sivuston tekniikkaan tutustuminen vie vähemmän aikaa.

Sisällönhallintajärjestelmän tunnistamiseksi voi käyttää What CMS -palvelua. Palvelu tarjoaa oman käyttöliittymän, mutta siihen voi syöttää vain yhden osoitteen kerrallaan. Tämän takia käyttöliittymän käyttäminen vie liikaa aikaa, jos projekteja on useita. What CMS tarjoaa kehittäjille myös ns. freemium-ohjelmointirajapinnan eli rajapinnan käyttäminen on tiettyyn pisteeseen asti ilmaista, mutta maksamalla palvelun käyttämisestä, sitä voidaan käyttää isompien sivustomäärien analysointiin ja nopeammin. Saavutettavuustyökalun käyttämiseksi voidaan kuitenkin käyttää ilmaista rajapintaa. Ilmainen rajapinta sallii 1000 pyyntöä kuukaudessa ja yhden pyynnön kerran kymmenessä sekunnissa. Koska yksi sivusto toimii tyypillisesti yhden sisällönhallintajärjestelmän päällä, rajapintaan tarvitsee lähettää vain yksi osoite jokaiselta sivustolta. Täten kuukaudessa voi analysoida 1000 sivustoa.

Rajapintapyynnön lähettämiseksi voidaan käyttää PHP:n curl-funktioita, jolla lähetetään rajapintapyyntö haluttuun verkko-osoitteeseen (ks. kuvio 9). Rajapinnan käyttämiseksi verkko-osoitteeseen on annettava key-parametriksi What CMS -palvelusta saatu API-avain ja url-parametriksi analysoitavan sivuston verkko-osoite.

```

//Detect content managment system
function ID_cms_audit($request){
    $projectID = $request->get_body();

    $projectInfo = get_field('project_info', $projectID);
    $siteURL = $projectInfo['site-base-url'];

    if(!$siteURL){
        echo 'no-site-url';
    }

    $API_KEY = 'XXX';
    $url = "https://whatcms.org/APIEndpoint/Detect?key=" . $API_KEY . "&url=" . $siteURL;

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $result = json_decode(curl_exec($ch));

    if($result->result->code == 201){
        return 'cms-not-detected';
    }else if($result->result->code == 120){
        return 'too-many-requests';
    }else if($result->result->code == 202){
        return 'url-unavailable';
    }else if($result->result->code == 111){
        return 'invalid-url';
    }else if($result->result->code == 121){
        return 'quota-exceeded';
    }else if($result->result->code == 200){
        $cms = $result->result->name;
        $values = array('cms' => $cms);
        update_field('project_info', $values, $projectID);

        return 'success';
    }
}

```

### Kuvio 9. What CMS rajapintapyyntö

Rajapintapyynnön lähettämisen jälkeen, rajapinta palauttaa käyttäjälle JSON muotoisen objektin. Jos analysoitavan sivun sisällönhallintajärjestelmä tunnistettiin onnistuneesti, se palautetaan käyttäjälle. Sisällönhallintajärjestelmän lisäksi käyttäjälle palautetaan ongelmatilanteiden analysoimiseksi kolme numeroinen koodi. Tämän koodin avulla käyttäjä pystyy tarvittaessa selvittämään, miksi rajapintapyyntö epäonnistui. Saavutettavuustyökalussa näiden koodien perusteella käyttäjälle näytetään käyttöliittymässä asianmukainen ilmoitus.

Saavutettavuustyökaluun voi tarvittaessa lisätä kerralla kymmeniä uusia projekteja (ks. luku 3.3.2). Näille sivustoille suoritetaan automaattisesti sisällönhallintajärjestelmän tunnistaminen. Kun käyttäjä lisää projektit järjestelmään, ne lisätään järjestelmässä olevaan tauluun. Tämän jälkeen WordPressin cron (ajastettu funktio suoritin), suorittaa testit työkalun tavallisten käyttöaikojen ulkopuolella, jotta työkalun suorituskyky ei laske sen käytön aikana. Koska cron suorittaa testit työkalun taustalla ja sen takia se ei pysty näyttämään käyttöliittymässä palautetta testeihin liittyen,

suoritetuista testeistä tallennetaan lokitiedosto. Lokitiedostosta näkee testin suoritusaikakohta, mahdollinen virhekoodi ja testattavan sivuston sisällönhallintajärjestelmä.

### 3.3.4 Yoast sitemap-tiedoston hakeminen

Jos WordPress-sisällönhallintajärjestelmän päälle rakennetulle sivustolle on asennettu Yoast SEO -lisäosa, se luo sivustolle automaattisesti sitemap.xml sivukarttatiedostoja (ks. kuvio 10). Sitemap-tiedostoja käytetään verkkosivuston sivujen osoiterakenteen esittämiseksi. Saavutettavuustyökalulla voi etsiä Yoastin luomia sivukarttoja, jolloin työkalu pystyy automaattisesti keräämään niistä sivustoon liittyviä verkko-osoitteita analysoitavaksi.

## XML Sitemap

Generated by **YoastSEO**, this is an XML Sitemap, meant for consumption by search engines.

You can find more information about XML sitemaps on [sitemaps.org](https://sitemaps.org).

This XML Sitemap Index file contains 10 sitemaps.

Sitemap	Last Modified
<a href="https://into-digital.fi/post-sitemap.xml">https://into-digital.fi/post-sitemap.xml</a>	2019-03-20 15:47 +02:00
<a href="https://into-digital.fi/page-sitemap.xml">https://into-digital.fi/page-sitemap.xml</a>	2019-03-26 15:33 +02:00
<a href="https://into-digital.fi/reference-sitemap.xml">https://into-digital.fi/reference-sitemap.xml</a>	2019-03-06 13:18 +02:00
<a href="https://into-digital.fi/contact-info-sitemap.xml">https://into-digital.fi/contact-info-sitemap.xml</a>	2019-02-14 16:58 +02:00
<a href="https://into-digital.fi/category-sitemap.xml">https://into-digital.fi/category-sitemap.xml</a>	2019-03-20 15:47 +02:00
<a href="https://into-digital.fi/post_tag-sitemap.xml">https://into-digital.fi/post_tag-sitemap.xml</a>	2018-10-26 10:44 +03:00

### Kuvio 10. Yoast! -lisäosan luoma sitemap-tiedosto

Analysoitavan sivuston toteutustavasta ja koosta riippuen sivustolla voi olla satoja sivuja ja artikkeleita, jonka takia kaikkien sivujen osoitteita ei kannata analysoida, koska suurin osa niistä voi käyttää samaa sivupohjaa. Satojen sivujen analysoiminen on myös todella paljon aikaa vievä prosessi. Tämän takia saavutettavuus hakee kaikista sivukartoista vain kolme ensimmäistä sivua. Kolme sivua jokaisesta sivukartasta mahdollistaa tarpeeksi laajan otannan, jotta erilaisia sivupohjia on mahdollisimman monta erilaista mutta analysoitavien sivujen kokonaismäärä pitää sivujen analysointiin käytettävän ajan mahdollisimman lyhyenä.

Kun käyttäjä painaa käyttöliittymässä Etsi sivukarttaa -painiketta, työkalun JavaScript tiedostossa suoritetaan AJAX-funktio, joka lähettää pyynnön WordPressin Rest API:lle, joka yrittää etsiä sivuston sivukarttoja. Pyyntö sivukarttojen hakemiseen lähetetään JavaScriptillä, jotta mahdollisesti löydettyt osoitteet voidaan lisätä dynaamisesti työkalun käyttöliittymään. Jos sivustolla on paljon sivuja, sivukarttojen hakeminen voi viedä pitkään. Tämän takia Rest API:ssa voidaan suorittaa funktio `set_time_limit(0)` (ks. kuvio 11), joka pitää yhteyden JavaScriptiin yllä niin pitkään kuin sivukarttojen hakemin vaatii. Jos tätä funktiota ei suoriteta, yhteys katkeaa kolmenkymmenen sekunnin jälkeen.

Yoast luo sivustolle useita sivukarttoja ja kerää niiden osoitteet yhteen juuritiedostoon (`sitemap_index.xml`). Sivukartta voidaan helposti hakea PHP:n `simplexml_load_file`-funktioilla, joka palauttaa käyttäjälle XML-objektin, joka sisältää juuritiedostoon linkitetty sivukarttojen osoitteet (ks. kuvio 11). Tämän jälkeen sama voidaan tehdä palautetun objektin jokaiselle osoitteelle. Näistä saaduista objekteista palautetaan työkalulle kolme ensimmäistä osoitetta.

```
set_time_limit(0);
$projectURL = $request->get_body();
$listOfPages = array();
$url = $projectURL . '/sitemap_index.xml';

if(!simplexml_load_file($url)){
    return 'Sitemap not found';
}

$sitemap_json = simplexml_load_file($url);
```

Kuvio 11. Sivukarttojen juuritiedoston hakeminen

Jos saavutettavuustyökalu löytää sivukartan, kaikki palautetut osoitteet lisätään automaattisesti Lighthouse-työkalun osoitekenttiin.

### 3.3.5 Lighthouse API

Työkalua suunniteltaessa todettiin, että lopulliselle palvelimelle ei voida asentaa sivustojen analysointiin käytettävää Googlen tarjoamaan Lighthouse-työkalua. Tämän rajoitteen aiheutti se, että palvelimelle ei voitu asentaa itse Node Package Managerin

avulla tarvittavia paketteja ja Lighthousen vaatimaa Google Chrome -selainta. Tämän takia sivustojen analysointi päätettiin toteuttaa erillisellä virtuaalipalvelimella, johon pystyttiin asentamaan kaikki tarvittavat paketit ja ohjelmistot. Mahdollisia palvelinvaihtoehtoja olivat Digital Oceanin dropletit ja Amazonin AWS S3 palvelimet, joista valittiin Digital Oceanin droplet-ympäristöjen aiemman käyttökokemuksen takia. Digital Ocean tarjoaa useita erilaisia valmiiksi konfiguroituja droplet-ympäristöjä. Työkalua varten valittiin valmiiksi asennettu Ubuntu 18.04 -ympäristö, johon asennettiin Apache 2 -palvelinympäristö.

### **Apache 2 asentaminen ja konfigurointi**

Ennen Apache 2 palvelimen asentamista on suositeltavaa hakea tuoreimmat päivitykset Ubuntun paketinhallintaan komennolla `sudo apt update`, jotta asennetuista ohjelmistoista saadaan niiden uusimmat versiot sulavan toiminnan takaamiseksi. Tämän jälkeen Apache voidaan asentaa komennolla `sudo apt install apache2`. Ennen kuin Apachea voidaan käyttää, palomuurista pitää sallia verkon ulkopuoliset pyynnöt komennolla `sudo ufw allow 'Apache'`. Tämän jälkeen palvelimeen voi ottaa yhteyttä, menemällä selaimella palvelimen IP-osoitteeseen.

Saavutettavuusanalyysien rajapinta päätettiin toteuttaa PHP-ohjelmointikielellä, joka mahdollistaa palvelimen ulkopuolelta tulevien pyyntöjen vastaanottamisen, sekä komentorivikomentojen suorittamisen palvelimella. Apache 2 ei oletuksena sisällä PHP-moduulia, joten se pitää asentaa erikseen.

Apachen PHP-moduuli asennettiin komennolla `sudo apt install php libapache2-mod-php php-mysql`. PHP-moduuli toimii jo tässä vaiheessa, mutta Apache etsii palvelimelta HTML-tiedostot ennen PHP-tiedostoja. Lataus järjestystä voi vaihtaa Apachen `dir.conf`-tiedostossa, jota voi muokata komennolla `sudo nano /etc/apache2/mods-enabled/dir.conf`. Kun Apachen konfigurointiin liittyviä tiedostoja muokataan, Apache tulee käynnistää uudelleen. Uudelleen käynnistäminen voidaan suorittaa komennolla `sudo systemctl restart apache2`.

## Headless Chrome

Saavutettavuusanalyysit suoritetaan Googlen Lighthouse-työkalulla, joka löytyy myös kaikista Google Chrome -työpöytäselaimista. Koska työkalu toimii palvelinympäristössä, jossa ei ole visuaalista käyttöliittymää, Google Chromea pitää käyttää komentorivin kautta headless-tilassa (headless Chrome).

Headless Chrome voidaan ladata Googlen repositoriosta komennolla `wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb`. Ladatau paketti voidaan asentaa komennolla `sudo dpkg -i google-chrome*.deb`, jonka jälkeen asennusta voi testata komennolla `google-chrome-stable --headless --disable-gpu --dump-dom https://www.chromestatus.com/`. Tämä komento hakee sivuston lähdekoodin ja tulostaa sen komentoriville. Komennon `--headless`-attribuutti kertoo, että Chromea tulee ajaa headless-tilassa, jolloin Chromen käyttöliittymää ei avata. Tämä attribuutti on erityisen tärkeä palvelinympäristössä, koska käyttöliittymää ei tueta.

## Google Lighthouse asentaminen

Saavutettavuusanalyysien suorittamista varten Google Lighthousesta on asennettava versio, jota voidaan ajaa komentorivillä. Google tarjoaa Lighthousesta komentoriviversion NPM-paketinhallintajärjestelmän (Node Package Manager) kautta. Lighthousen asentaminen vaatii, että ympäristössä, jossa sitä käytetään, on asennettu vähintään Node.js 10.13 tai uudempi. Jos Lighthousea halutaan käyttää palvelimella missä tahansa kansiossa, se pitää asentaa globaalisti komennolla `npm install -g lighthouse`. Kun palvelimelle on asennettu headless Chrome ja Lighthouse, Lighthousea voi testata komentorivikomennolla `lighthouse [URL] --chrome-flags="--headless"`.

## Rajapinta

Kun kaikki rajapinnan vaatimat ohjelmistot on asennettu, palvelimelle asennettu PHP mahdollistaa rajapintapyyntöjen vastaanottamisen, komentojen suorittamisen ja datan palauttamisen työkalun käyttöliittymään. Google Chromen työpöytäversion Lighthouse mahdollistaa vain yhden verkko-osoitteen analysoinnin kerralla, tämän takia se ei sovellu suurien osoitemäärien analysointiin. Tämä sama rajoite on myös komentoriviversiossa, mutta tämä pystytään kiertämään ajamalla komento monta

kertaa ja jokaisella kerralla komennolle annetaan automaattisesti uusi osoite. Rajapintaa lähetetään tämän takia objekti, joka sisältää kaikki analysoitavat osoitteet ja Lighousen-rajoittamiseksi tai rajoitteiden poistamiseksi vaadittavat parametrit (ks. kuvio 12). Rajapinnassa syötetyt osoitteet iteroidaan läpi (ks. kuvio 13) ja jokaiselle suoritetaan annetut analyysit.

```
let auditData = {
  userID: 1,
  projectID: 1581,
  audits:{
    cms:{
      cms_audit:false
    },
    lighthouse:{
      lighthouse_accessibility:true,
      lighthouse_performance:true,
      lighthouse_pwa:true,
      lighthouse_best_practices:true,
      lighthouse_seo:true
    }
  },
  single:false,
  url_list:[
    {
      pageUrl:"https://into-digital.fi/"
    },
    {
      pageUrl:"https://into-digital.fi/referenssit/"
    }
  ]
}
```

Kuvio 12. Esimerkki rajapintaan lähetettävästä objektista

```
foreach($urlList as $url){
  array_push($object, shell_exec('lighthouse ' . $url->pageUrl . ' --chrome-flags="--no-sandbox --headless --disable-gpu" --disable-device-emulation --throttling-method=provided ' . $auditsToRunString . ' --output="json"'));
}
```

Kuvio 13. Datan iterointi ja analyysien suorittaminen

Kun yksittäinen verkko-osoite on analysoitu, Lighousen palauttama JSON-data tallennetaan taulukkoon, joka lähetetään takaisin käyttöliittymän palvelimelle, jossa se tallennetaan JSON-tiedostoiksi (ks. kuvio 14), jotta se voidaan säilöä ja analyseja ei tarvitse aina suorittaa uudelleen.

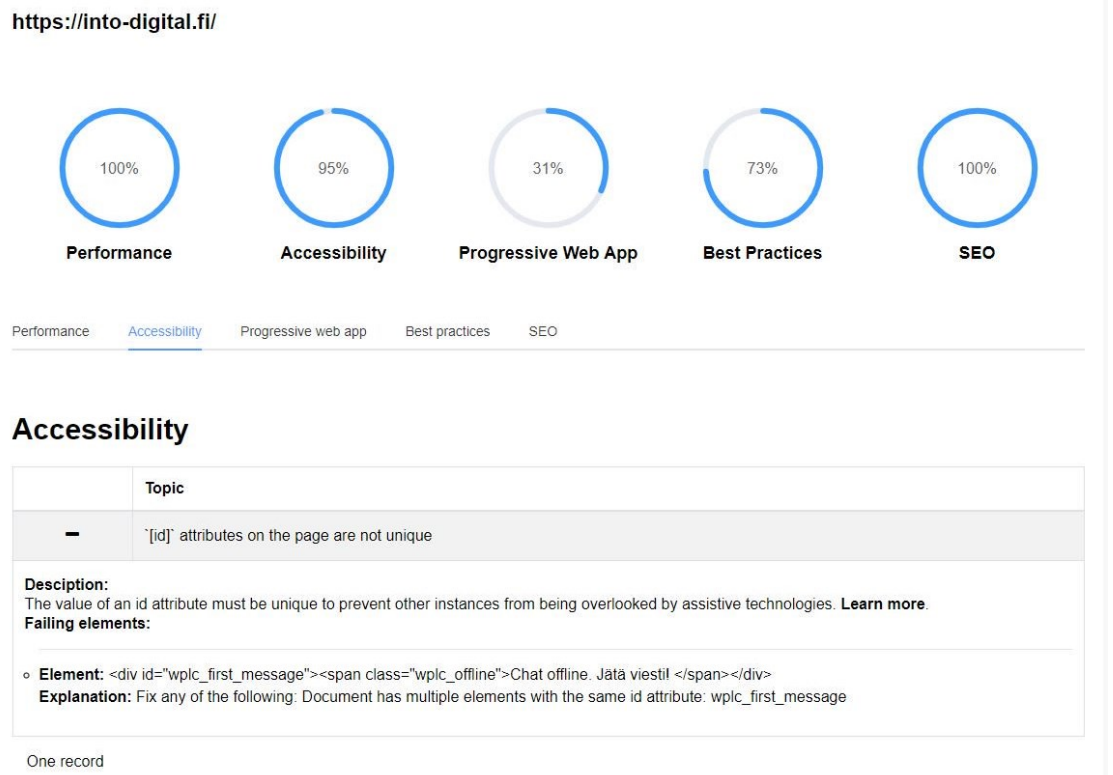
```
$myfile = fopen($uploads['basedir'] . '/lighthouse-reports/' . $projectID . '/' . $fileName . ".json", "w") or die("Unable to open file!");
fwrite($myfile, json_encode($encodedData, JSON_PRETTY_PRINT));
fclose($myfile);
```

Kuvio 14. JSON-datan tallentaminen tiedostoon

### 3.3.6 Lighthouse datan näyttäminen

Lighthouse-testi palauttaa saavutettavuustyökälulle tulokset JSON-formaatissa, jonka takia tulokset pitää muuttaa luettavaan muotoon. Google näiden JSON-tiedostojen

lukemiseen oman työkalun, mutta tiedostot täytyy ladata jokaisella katselukerralla työkaluun erikseen. Tämän takia JSON-tiedostojen lukemiseksi päätettiin toteuttaa oma käyttöliittymä (ks. kuvio 15). Kun käyttäjä painaa raportin linkkiä yksittäisen projektin sivulla, saavutettavuustyökalu lataa JSON-tiedoston AJAX:lla ja näyttää sen käyttöliittymässä. Raporttien lukemiseen käytettävä käyttöliittymä on toteutettu Element.io kirjastolla.



Kuvio 15. Lighthouse-raporttien datan visualisointi

### 3.4 Testaaminen

Saavutettavuustyökalua testattaessa testattiin sivustojen saavutettavuuden analysointia yksittäisellä sivustolla, sekä suurien sivustomäärien syöttämistä järjestelmään yhdellä kerralla, jonka yhteydessä näihin sivustoihin suoritettiin testi sisällönhallinta-järjestelmän selvittämiseksi.

Yksittäistä sivustoa analysoidessa vertailtiin työkalun palauttamia tuloksia Google Chrome -selaimen Lighthouse-työkalun tuloksiin ja testattiin sivuston verkko-osoitteiden noutamista sivukartoista, sekä erityyppisiä verkko-osoitteita, jotka saattavat sisältää erilaisia parametreja. Komentorivin kautta suoritettu analyysi palautti samat

virheilmoitukset kuin selain, mutta näiden lisäksi tuloksien yhteyteen on liitetty tarkempi selvitys, miten puute voidaan korjata. Tämä selvitys on selainversiossa piilotettu erilliselle verkkosivulle.

Saavutettavuustyökalulla voidaan hakea verkko-osoitteita WordPress-sivustoilta, joihin on asennettu Yoast SEO -lisäosa, joka luo sivustosta sivukarttoja. Tällä työkalulla sivukarttoja hakemalla, sivuston analysointi onnistuu ilman että käyttäjän tarvitsee erikseen kerätä sivustolta eri sivujen verkko-osoitteita. Sivustojen rakenteet kuitenkin poikkeavat todella usein toisistaan, riippuen sivuston kehittäjän työtavoista.

Osalla sivustoista jokainen sivu saattaa olla sivupohjaltaan erilainen, ja osalla sivustoista saattaa olla satoja sivuja mutta suurin osa sivuista käyttää samaa sivupohjaa. Suurien poikkeamien takia ei ole yksinkertaista sääntöä, jonka mukaan sivukartoista voidaan hakea osoitteita analysoitavaksi, jotta saataisiin mahdollisimman tarkkaa dataa sivupohjien saavutettavuuspuutteista ilman että kaikki sivuston sivut pitää analysoida. Jos kaikki sivut analysoidaan, analysoinnin vaatima aika pitenee huomattavasti ja palautetun datan käsittely ja lukeminen vie liian paljon aikaa. Tämän takia sivustoja analysoidessa käyttäjän on helpompaa itse kerätä sivustolta osoitteita, joissa on käytetty eri sivupohjia.

Saavutettavuustyökalua testattaessa huomattiin, että verkko-osoitteet, jotka sisältävät GET-parametreja, eivät palauta saavutettavuustyökalulle dataa. Tämän virheen aiheutui, kun verkko-osoitteista muodostettuihin tiedostonimiin oli jäänyt merkkejä, joita ei tueta tiedostonimissä.

Saavutettavuustyökalua testattaessa työkaluun syötettiin kerralla satojen eri verkkosivustojen etusivun verkko-osoite, ja niiden perusteella testattiin, millä sisällönhallintajärjestelmällä sivusto on toteutettu. Palautettujen sisällönhallintajärjestelmien perusteella voidaan muun muassa etsiä potentiaalisia asiakkaita toimeksiantajan osamisaalueelta. Tämä testi onnistui moitteettomasti.

## 4 Toteutuksen analysointi

### 4.1 Toteutuksen hyödyllisyys

Saavutettavuustyökalua on käytetty vuoden 2019 aikana noin 150 verkkosivuston analysointiin, ja osasta analysoiduista sivustoista on syntynyt uusia asiakassuhteita. Työkalu on helpottanut sivustojen analysointia, sillä kokonaisen sivuston käyttöliittymä pystytään analysoimaan minuuteissa. Analysoiduista sivustoista jää myös merkintä ja tiedot suoritetuista analyysistä, jolloin mahdollisten korjauksien jälkeen on mahdollista verrata, miten sivustoa on pystytty kehittämään. Vanhoista projekteista on myös mahdollista löytää, kuka sivustoa on korjannut, jolloin tarvittaessa uusi työntekijä pystyy selvittämään tarvittavia lisätietoja edelliseltä sivuston parissa töitä tehneeltä.

### 4.2 Jatkokehitysmahdollisuudet

Saavutettavuustyökalussa on selkeitä parantamista vaativia osa-alueita. Projektien listaus -sivulla olisi hyödyllistä pystyä muokkaamaan projektin perustietoja, kuten vastuuhenkilöitä ja projektin statusta. Listaussivun filttäreitä voisi myös kehittää, sillä tällä hetkellä sisällönhallintajärjestelmien filteri listaa vain 5 tavanomaisinta sisällönhallintajärjestelmää. Tämän sijaan tähän voisi listata dynaamisesti kaikki järjestelmästä löytyvät sisällönhallintajärjestelmät, joita sivustoilla on käytetty.

Yksittäisen projektin sivulla vanhat raportit ylikirjoitetaan, jos sama osoite analysoidaan uudestaan. Tällöin analysoitua dataa saattaa kadota, jos sitä ei ole otettu talteen ennen analyysiä.

Työkalussa on mahdollisuus hakea sivustojen sivujen verkko-osoitteita automaattisesti sivukartoista. Tämä ominaisuus toimii kuitenkin vain tietyillä sivustoilla, jotka on toteutettu WordPressillä ja niihin on asennettu Yoast SEO -lisäosa. Mahdollisten sivustotyyppien kirjoa voi mahdollisesti kasvattaa tutkimalla yleisimpiä ratkaisuja sivukarttojen luomiseen. Sivujen hakemiseen käytettyä algoritmia voidaan myös kehittää, jolloin tätä ominaisuutta voitaisiin käyttää useammin, sillä nykyisessä versiossa ei voida olla varmoja saadaanko sivustolta dataa tarpeeksi laajasti. Mahdollisimman

laajan data keräämiseksi, työkalussa voitaisiin ottaa kuvia sivuista, verrata niitä tekoälyllä toisiinsa ja valita sivut, jotka eivät muistuta toisiaan.

## 5 Yhteenveto ja pohdinta

Saavutettavuustyökalulle asetetut tavoitteet saavutettiin. Työkalulla pystytään seuraamaan verkkosivujen saavutettavuuteen liittyvien projektien edistymistä, suorittamaan käyttöliittymien saavutettavuusanalyseja ja säilömään projekteihin liittyvää dataa mahdollista myöhempää käyttöä varten. Työkalua on käytetty vuoden 2019 aikana yli sadan verkkosivuston analysointiin ja sitä käytettäessä on huomattu, että kaikkia työkalun toimintoja ei ole juurikaan käytetty. Esimerkiksi sivuston verkko-osoitteiden noutamiseen käytettävä toiminto toimii vain muutamissa sivustoissa, jonka takia verkko-osoitteiden noutaminen on helpompaa manuaalisesti.

Saavutettavuustyökalun kehittämisen suurimmat haasteet muodostuivat palvelinrajotteista, joiden takia työkalu päätettiin toteuttaa hajautetusti useammalle palvelimelle. Hajautettu ympäristö vaikuttaa kuitenkin negatiivisesti työkalun suorituskykyyn, joka näkyy hieman pidempänä analysointiaikana yhtä verkko-osoitetta kohden. Tulevaisuudessa työkalun keskittäminen yhdelle palvelimelle on jatkon kannalta järkevin ratkaisu, sillä palvelimien hallintaan kuluva aika ja käyttökustannukset pienenevät.

Saavutettavuustyökalun käyttöaste sen käyttöönoton jälkeen on kasvanut huomattavasti vuoden 2019 loppua kohden. Verkkopalveluiden saavutettavuudesta eivät ole kiinnostuneita pelkästään julkisenhallinnon tahot, joita saavutettavuus direktiivi koskee, vaan myös yksityiset yritykset ovat osoittaneet kiinnostustaan. Erityisesti yksityisen sektorin mielenkiinto näkyy huomattavana kasvuna saavutettavuuteen liittyvissä yhteydenotoissa ja siihen liittyvien projektien toteuttamisessa.

## Lähteet

Campbell, A., Cooper, M., Kirkpatrick, A. & O Connor, J. 2018. Web Content Accessibility Guidelines (WCAG) 2.1. Viitattu 20.11.2019.

<https://www.w3.org/TR/WCAG21/> .

Contrast and Color Accessibility. 2018. Opas WebAIM sivustolla. Viitattu 20.11.2019.

<https://webaim.org/articles/contrast/> .

Digipalveluiden saavutettavuus-opas. 2018. Opas Eficoden sivustolla. Viitattu

2.1.2019. [https://www.eficode.com/hubfs/documents/eficode-uxstudiosaavutettavuusopas\\_V1.2.pdf](https://www.eficode.com/hubfs/documents/eficode-uxstudiosaavutettavuusopas_V1.2.pdf) .

EN 2016/2102. Euroopan parlamentin ja neuvoston julkaisema

saavutettavuusdirektiivi. Viitattu 20.11.2019. <https://eur-lex.europa.eu/eli/dir/2016/2102/oj> .

Saavutettavuus. N.d. Artikkelit valtiovarainministeriön sivustolla. Viitattu 7.1.2019.

<https://vm.fi/saavutettavuusdirektiivi> .

Saavutettavuusdirektiivi edistää yhdenvertaisuutta. N.d. Digitoimisto Poutapilven ylläpitämä sivusto saavutettavuudesta. Viitattu 19.11.2019.

<https://saavutettavuusdirektiivi.fi/>.

Saavutettavuuslaki on täällä tänään! 23.9.2019. Artikkelit digitoimisto Poutapilven ylläpitämällä saavutettavasti.fi -sivustolla. Viitattu 19.11.2019.

<https://www.saavutettavasti.fi/saavutettavuuslaki-on-taalla-tanaan/> .

WordPress market share. N.d. Artikkelit Kinstan sivustolla. Viitattu 19.11.2019.

<https://kinsta.com/wordpress-market-share/> .