

KARELIA-AMMATTIKORKEAKOULU
Tietojenkäsittelykoulutus

Jani Hyvönen

SAAVUTETTAVUUDEN HUOMIOIMINEN VERKKOSIVUJEN JA -
SOVELLUSTEN TOTEUTUKSESSA

Opinnäytetyö
joulukuu 2021



OPINNÄYTETYÖ
joulukuu 2021
Tietojenkäsittelykoulutus

Tikkarinne 9
80200 JOENSUU
+358 13 260 600 (vaihde)

Tekijä(t)
Jani Hyvönen

Nimeke
Saavutettavuuden huomioiminen verkkosivujen ja -sovellusten toteutuksessa

Toimeksiantaja
Karelia-ammattikorkeakoulu

Tiivistelmä

Tämän opinnäytetyön toimeksiantona oli tehdä saavutettavuusselvitys ”Pohjois-Karjalan ja Japanin Naganon metsäbiotalousyhteistyön kehittäminen 2020–2022” -hankkeen aikana tehdyille Virtual Showroom -verkkosivulle. Tavoitteena oli tuottaa tekstipohjainen saavutettavuusselvitys Suomen lain velvoittamien saavutettavuuskriteerien perusteella.

Työn alussa tutustuttiin erilaisiin vammoihin ja apulaitteisiin sekä Euroopan saavutettavuusdirektiivin ohjeistamaan verkkosisällön saavutettavuusohjeeseen (WCAG 2.1), jonka A- ja AA-tason onnistumiskriteereitä Suomen laki velvoittaa noudattamaan. Ennen saavutettavuusselvitystä tutustuttiin siihen, kuinka verkkopalveluiden kehittäjien tulisi huomioida saavutettavuus. Saavutettavuusselvityksessä sivuston saavutettavuutta tarkasteltiin siihen tarkoitetulla selainlisäosa Axella ja NVDA-ruudunlukijalla. Lisäksi saavutettavuutta selvitettiin manuaalisesti lähdekoodista. Virtual Showroom -verkkosivuston saavutettavuuden tilanne käytiin läpi sivuston jokaisesta mahdollisesta näkymästä.

Saavutettavuusselvitys saatiin tehtyä kaikkien tavoitteiden mukaisesti. Sivustolta löydettiin useita saavutettavuusongelmia. Sivuston näppäimistö- ja ruudunlukijanavigointi eivät toimineet lainkaan. Lisäksi mobiiliresponsiivisuus ei toiminut Suomen lain velvoittamien kriteerien mukaisesti.

Kieli
suomi

Sivuja 62

Asiasanat
saavutettavuus, semantiikka, HTML, WCAG



THESIS
December 2021
Degree Programme in Business Information Technology

Tikkarinne 9
80200 JOENSUU
FINLAND
+ 358 13 260 600 (switchboard)

Author (s)
Jani Hyvönen

Title
Mainstreaming Accessibility in the Implementation of Websites and Applications

Commissioned by
Karelia University of Applied Sciences

Abstract

The purpose of this thesis was to report the state of accessibility of the Virtual Showroom website for Karelia University of Applied Science's project "Forest Bioeconomy Development Co-operation 2020-2022 between North Karelia Prefecture in Finland and Nagano Prefecture in Japan". The purpose was to create a text-based state of accessibility report based on Finland's accessibility laws and criteria.

At the beginning of the thesis, research was done on different types of disabilities and assistive devices were introduced. During the familiarization of Europe's accessibility directive, it was found out that the success criteria of levels A and AA of the Web Content Accessibility Guidelines 2.1 are required by law in Finland. The theoretical background of the state of accessibility was based on the findings of how web developers should take accessibility into account during development. During the investigation of Virtual Showroom's state of accessibility, tools such as Axe and NVDA screen reader were used, and the source code was manually checked. The state of accessibility was reviewed for each possible view of the site.

The state of accessibility was successfully reported whilst fulfilling all the set goals. Many accessibility problems were found on the website, such as non-functional keyboard and screen reader navigation, in addition to non-functional mobile responsiveness.

Language
Finnish

Pages 62
Appendices 0
Pages of Appendices 0

Keywords
accessibility, semantics, HTML, WCAG

Sisältö

1	Johdanto	5
2	Verkkopalveluiden saavutettavuus.....	6
2.1	Mitä saavutettavuus on?	6
2.2	Mitä käytettävyys on?	7
2.3	Palveluiden käyttötavat vammaisilla	8
2.3.1	Saavutettavuus näkövammaisilla.....	8
2.3.2	Saavutettavuus kuulovammaisilla	11
2.3.3	Saavutettavuus liikuntavammaisilla	12
2.3.4	Saavutettavuus kognitiivisesti vammautuneilla	13
3	Verkkosisällön saavutettavuusohjeet (WCAG 2.1)	14
3.1	Mikä on WCAG?	14
3.2	WCAG 2.1:n ohjenuorat.....	15
4	Saavutettavuus kehittäjän näkökulmasta	17
4.1	HTML:n saavutettavuus	17
4.1.1	Semanttiset HTML-elementit	17
4.1.2	Näppäimistö navigointi.....	22
4.1.3	HTML:n hyvät käytännöt.....	23
4.1.4	Värit ja kontrasti	26
4.2	JavaScript ja CSS.....	27
4.3	WAI-ARIA	31
4.4	Mobiililaitteet ja saavutettavuus	33
5	Virtual Showroom -verkkosivuston saavutettavuuden selvitys.....	35
5.1	Virtual Showroom -verkkosivuston taustoitus	35
5.2	Testauksen työkalut.....	35
5.2.2	Saavutettavuustyökalujen vertailu toisiinsa	37
5.3	Virtual Showroomin saavutettavuuden selvitys.....	40
5.3.1	Saavutettavuuden selvityksen toteutustapa ja työkalut.....	40
5.3.2	Aloituskäyttö	41
5.3.3	Käytönopastus	43
5.3.4	Päänäkymä.....	44
5.3.5	Yritysnäkymä	47
5.3.6	Virtual Showroomin responsiivisuuden testaus.....	51
5.3.7	WCAG 2.1:n onnistumiskriteerit Virtual Showroomissa	55
5.3.8	Saavutettavuuden yhteenveto	56
6	Pohdinta.....	57
6.1	Saavutettavuudesta hyötyvät käyttäjät ja apulaitteet	57
6.2	Saavutettavuuden huomiointi kehittäjänä	57
6.3	Virtual Showroomin saavutettavuus selvityksen työkalut ja tulokset... ..	58
	Lähteet.....	60

Lyhenteet

CSS	Cascading Style Sheets on tyylittelyyn tarkoitettu kieli, joka määrittelee HTML:n tai XML:n ulkonäön ja esitystavan (Mozilla 2021a).
HTML	HTML eli HyperText Markup Language määrittää verkkosivujen rakenteen ja sisällön (Mozilla 2021b).
WAI	Web Accessibility Initiative kehittää standardeja ja materiaaleja helpottamaan saavutettavuuden ymmärtämistä ja implementointia (Web Accessibility Initiative 2021).
WAI-ARIA	Web Accessibility Initiativen kehittämä "Accessible Rich Internet Applications Suite" määrittelee tavan tehdä verkkosisällöstä ja -sovelluksista saavutettavamman erityisesti dynaamiseen ja kehittyneempiin käyttöliittymäkomponentteihin (Cooper 2020).
WCAG	Web Content Accessibility Guidelines eli verkkosisällön saavutettavuusohjeet määrittävät, kuinka verkkopalveluiden sisältö tehdään saavutettavaksi (WCAG 2.1 2018).

1 Johdanto

Koronapandemian seurauksena verkkopalveluiden tärkeys on korostunut ympäri maailmaa. Verkkopalveluiden käytön yleistymisen myötä on tärkeää saada palvelut käytettäviksi mahdollisimman monelle ihmiselle. Yli miljardilla ihmisellä on jonkinlainen vamma. Saavutettavat verkkosivut takaavat siis paremman käyttökokemuksen jopa 15 % koko maailman väestöstä. (World Health Organization 2020.)

Verkkopalveluiden saavutettavuus otetaan huomioon eri tavalla ympäri maailmaa. Euroopassa on perustettu saavutettavuusdirektiivi, joka neuvoo noudattamaan WCAG 2.1:n ohjeita (Euroopan komissio 2021). Yhdysvalloissa puolestaan noudatetaan WCAG:n edeltävän version 2.0:n mukaisia ohjeita (USA Government 2021).

Saavutettavuuden tärkeys nykypäivänä on jatkuvasti kasvavaa. Koronapandemian myötä suuri määrä arkisia asioita, kuten koulutus, työskentely, terveystalvelut ja monet muut ovat siirtyneet pääpiirteittäin internetin välityksellä toteutettaviksi. Mikäli nämä palvelut eivät olisi saavutettavia, olisi esimerkiksi koulunkäynti tai työskentely vaikeampaa suurelle määrälle ihmisiä ympäri maailmaa. Palveluiden saavutettavuus on reilua, ja sillä saavutetaan tasa-arvoinen palveluiden käyttö kaikille. (Henry 2021a.)

Saavutettavuus mahdollistetaan verkkopalveluiden oikeanlaisella suunnitellulla ja koodauksella. Tällä hetkellä monet sivustot eivät ole saavutettavia, jonka seurauksena sivustojen käyttö on joillekin vaikeaa tai jopa mahdotonta. Yhdistyneiden kansakuntien yleissopimus vammaisten henkilöiden oikeuksista määrittelee tieto- ja viestintäteknologioiden, kuten internetin käytön ihmisoikeutena. (Henry 2021a.)

Opinnäytetyön toimeksiantona oli tehdä saavutettavuus selvitys Karelian tietojenkäsittelyohjelman opiskelijoiden ja yhteistyökumppaneiden Japani-hankkeen

aikana kehittämään Virtual Showroom -verkkopalveluun. Opinnäytetyössä selvitettiin saavutettavuutta sekä käyttäjän että kehittäjän näkökulmasta, jonka jälkeen Virtual Showroomin saavutettavuutta tarkasteltiin käyttäen apuna erilaisia työkaluja sekä teoriaosuuden aikana opittuja parhaita käytäntöjä. Sivuston saavutettavuutta ei ole ennen tätä opinnäytetyötä tarkistettu perusteellisesti, ja tämän opinnäytetyön on tarkoituksena tuottaa toimeksiantajalle perusteellinen raportti saavutettavuuden tilanteesta.

Opinnäytetyön aiheen valintaan vaikuttivat monet eri asiat, ja tavoitteita opinnäytetyötä tehdessä oli useita. Opinnäytetyön tärkein tavoite oli saada mahdollisimman kattava teoriapohja, jonka avulla voidaan tuottaa tarkka raportti Virtual Showroom -ympäristön saavutettavuuden tilanteesta. Henkilökohtaisena tavoitteena verkkopalvelukehittäjänä oli saada hyvä yleiskäsitys saavutettavuudesta ja kuinka se voidaan ottaa huomioon mahdollisimman hyvin verkkopalveluiden kehityksessä. Tavoitteena oli saada hyvä teoriapohja käytännössä hyödynnettäville toimenpiteille ammattimaisessa verkkopalveluiden kehityksessä sekä realistinen käytännönläheinen kokemus saavutettavuuden tilanteen selvittämisestä jo olemassa olevaan verkkopalveluun.

2 Verkkopalveluiden saavutettavuus

2.1 Mitä saavutettavuus on?

Verkkosivujen ja -sovellusten saavutettavuudella tarkoitetaan verkkopalveluiden käytön mahdollistamista mahdollisimman monelle ihmiselle. Saavutettavat verkkosivut tarkoittavat sitä, että myös vammaiset ihmiset voivat käyttää niitä. Internetin käyttöä haittaaviin vammoihin kuuluvat auditiiviset, kognitiiviset, neurologiset, fyysiset, puhe- ja näkövammat. Saavutettavan verkkopalvelun tulee huomioida kaikki palvelun käyttöön vaikuttavat vammat. (Henry 2021a.)

Verkkopalveluiden saavutettavuuden takaamiseksi täytyy kaikkien verkkosivujen suunnittelussa, ohjelmoinnissa ja sisällöntuotannossa huomioida saavutettavuus. Verkkopalveluiden käyttöön on olemassa monia erilaisia avustavia laitteita, jotka mahdollistavat palvelun käytön apulaitetta tarvitseville. Apulaitteiden

ansioista esimerkiksi sokea tai neliraajahalvaantunut henkilö voi käyttää verkkopalvelua. Apulaitteet toimivat oikein vain niissä tapauksissa, kun verkkopalvelu on luotu saavutettavaksi. (Papunet 2021a.)

Saavutettavuus on hyödyksi myös ihmisille, joilla ei ole vammoja. Ihmisten vanhetessa suurimmalla osalla värien havaitseminen ja toisistaan erottaminen heikenee. Kontrastien huomioiminen verkkosivuja kehittäessä auttaa värisokeiden ihmisten lisäksi vanhoja ihmisiä. Saavutettavuudesta on apua ja hyötyä myös ihmisille, jotka ovat vammautuneet väliaikaisesti. Saavutettavat verkkopalvelut palvelevat näiden lisäksi eri alustoja käyttäviä henkilöitä, kuten esimerkiksi älypuhelimet, -kellot ja -televisiot kuuluvat verkkopalveluita käyttäviin alustoihin, joissa syötetapa on erilainen. Saavutettavat verkkosivustot ovat myös helpommin löydettävissä hakukoneiden, kuten Googlen avulla. (Henry 2021a.)

2.2 Mitä käytettävyys on?

Käytettävyys tarkoittaa verkkopalvelun helppokäyttöisyyttä. Käytettävyys kulkee saavutettavuuden kanssa käsi kädessä. Käytettävyystutkija Jakob Nielsenin mukaan käytettävyys koostuu seuraavista osa-alueista:

- opittavuus
- tehokkuus
- muistettavuus
- virheettömyys
- tyytyväisyys.

(Papunet 2021b.)

Käytettävyyttä voidaan arvioida testaamalla, kuinka nopeasti ja vaivattomasti käyttäjät suoriutuvat tavoitteistaan. Kun esimerkiksi verkkopankissa laskun maksaminen on vaivatonta ja nopeaa, verkkopankki on käytettävyydeltään hyvä. (Papunet 2021b.)

Pelkkä saavutettavuus ei kuitenkaan takaa, että vammaisten olisi helppo käyttää sivuja. Sivulla eksyminen voi olla todella helppoa. Liian pitkät sivut, jotka eivät tarjoa käyttäjälle nopeaa ja helppoa pääsyä sivuston pääsisältöön voivat turhauttaa ja väsyttää käyttäjää. Hyvien käytettävyysperiaatteiden noudattaminen

tukee myös sivun saavutettavuutta. Sivut tulisi pitää mahdollisimman yksinkertaisena, ja korostaa sivun tärkeitä elementtejä. Turhia graafisia elementtejä ja välkkyviä asioita kannattaa välttää. (Nielsen & Pernice 2021, 42.)

2.3 Palveluiden käyttötavat vammaisilla

2.3.1 Saavutettavuus näkövammaisilla

Maailman terveysjärjestön mukaan vähintään 2.2. miljardia ihmistä ovat näkövammaisia. Maailman terveysjärjestön mukaan noin puolessa tapauksista näkövammautuminen olisi voitu estää. Enemmistö näkövammoista on yli 50-vuotiailla ihmisillä. (World Health Organization 2021a.)

Visuaalisiin vammoihin kuuluvat sokeus, huononäköisyys ja värisokeus. Monet näistä vammoista kärsivistä käyttävät apunaan näytön suurentajia. Näytön suurentajia löytyy sekä fyysisinä laitteina että ohjelmistoina. Monissa selaimissa on valmiiksi sisäänrakennettu zoomaustoiminto, jolla verkkosivujen sisältöä voidaan suurentaa. (Mozilla 2021c.)

Osa käyttäjistä käyttää ruudunlukijoita, jotka lukevat verkkosivujen sisällön äänen käyttäjälleen. Ruudunlukijoita on olemassa sekä ilmaisia että ostettavia. Suosituille käyttöjärjestelmille kuten Windowsille, Mac OS:lle, Linuxille, Androidille ja Chrome OS:lle on olemassa omat käyttöjärjestelmäkohtaiset ruudunlukijansa. Osa ruudunlukijoista ovat käyttöjärjestelmään jo valmiiksi rakennettuja ohjelmistoja, ja osa erikseen hankittavia ohjelmistoja. Käyttöjärjestelmäkohtaiset ruudunlukijat pystyvät lukemaan tekstiä verkkopalveluista, mutta myös muista tietokoneen sovelluksista. Poikkeus käyttöjärjestelmäkohtaisiin ruudunlukijoihin on esimerkiksi ChromeVox, joka on Google Chrome -selaimen ilmaiseksi ladattava lisäosa. (Mozilla 2021c.)

Ruudunlukijoiden käytössä on monesti käytössä ruudunlukijan näppäinmääräte. Ruudunlukijan näppäinkomentojen yhteydessä käytetään siis myös näppäin-

määritettä, joka voi olla esimerkiksi Caps Lock -näppäin (taulukko 1). Näppäinmäärittteen käyttö estää näppäinkomentojen sekoittamisen muiden tietokoneen näppäinkomentojen kanssa. (Mozilla 2021d.)

Taulukoissa (Taulukko 1 ja taulukko 2) VO ja NVDA tarkoittavat kyseessä oleville ruudunlukijoille määritettyjä näppäinmäärittettä. Molemmista ruudunlukijoissa voidaan käyttää esimerkiksi Caps Lock -näppäintä näppäinmäärittteenä (Mozilla 2021d).

Näppäinkomento	Kuvaus
VO + nuolinäppäimet	Siirrä VO-kursoria ylös, oikealle, alas, vasemmalle
VO + välilyönti	Valitse/aktivoi VO-kursorin osoittama elementti.
Ctrl	(kun VO puhuu) Pysäytä/Jatka puheetta
VO + Z	Aloita viimeisin puhe alusta

Taulukko 1. Mac OS:n mukana tulevan VoiceOver-ruudunlukijan näppäinkomentoja (Mozilla 2021d).

Näppäinkomento	Kuvaus
Ylä- ja alanuoli, tai vaihtonäppäin + sarkain ja sarkain	Siirry edelliseen/seuraavaan sivun elementtiin ja lue se.
NVDA + ylänuoli	Lue tämänhetkinen rivi.
NVDA + alanuoli	Aloita lukemaan tämänhetkisestä kohdasta.
Välilyönti	(kun lomakkeen sisällä) Valitse valittu arvo
NVDA + Q	Sulje NVDA ruudunlukija pois päältä.

Taulukko 2. Windowsille ladattavan NVDA ruudunlukijan näppäinkomentoja (Mozilla 2021d).

Verkkosivuilla tabulaattorilla liikkuminen tulisi tapahtua sivustorakenteen huomiioon ottaen loogisessa järjestyksessä. Se auttaa ihmisiä, jotka käyttävät ruudunlukijoita, -suurentajia tai motoriikkaan liittyviä apuvälineitä. Siitä on apua myös käyttäjille, jotka haluavat navigoida sivulla näppäimistön avulla. Linkkien väliin

kannattaisi jättää tarpeeksi tilaa, jotta niiden klikkaaminen olisi helpompaa. (Nielsen & Pernice 2021, 42.)

Linkkejä yhdelle sivulle ei kannata laittaa liikaa. Käyttäjä, joka ei tarvitse avustavaa teknologiaa sivun sisällön näkemiseen, voi tutkailla sisällön ja sivun rakenteen läpi nopeasti. Ruudunlukijan käytössä käyttäjän on kuunneltava sivuston sisältö läpi, johon voi kulua paljon aikaa. Nielsenin tutkimuksen mukaan osa käyttäjistä saattaa jopa kuunnella koko sivun läpi, eivätkä he klikkaa mitään ennen kuin uskovat sivun olevan lopussa. Nielsenin tutkimuksessa yksi käyttäjä kuunteli läpi 40 linkkiä, kunnes hän tajusi olevansa väärällä sivulla. Nielsen suosittelee, että yhdelle sivulle laitettaisiin enintään 20 linkkiä. (Nielsen & Pernice 2021, 69.)

Yleisimmillä mobiilialustoilla on nykyisin täysin toimivat ruudunlukijat. Ruudunlukijat toimivat mobiililaitteilla samantapaisesti kuin tietokoneellakin, mutta niitä käytetään kosketuseleillä eikä näppäimistöllä. Yleisimmät ruudunlukijat mobiilialustoilla ovat Android TalkBack sekä iOS:n VoiceOver. Käytettäessä esimerkiksi TalkBack-ruudunlukijaa muuttuvat normaalit puhelimen toiminnot hieman erilaisiksi. (Mozilla 2021e.)

Mobiililaitteen käytön esimerkkejä, kun käytetään Androidin TalkBack-ruudunlukijaa:

1. Kun käyttäjä klikkaa ruudulla olevaa sovellusta yhdesti, ruudunlukija lukee sovelluksen nimen ääneen.
2. Vasemmalle tai oikealle pyyhkäisy siirtyy sovellusten välillä, tai painikkeiden ja kontrollien, jos käyttäjä on ohjauspalkissa. TalkBack lukee kaikki vaihtoehdot ääneen.
3. Tuplaklikkaus avaa sovelluksen / valitsee vaihtoehdon.
4. Halutessaan voi käyttää "explore by touch" -tapaa. Sormen liikkuessa näytön päällä eri kohtiin laite lukee ääneen sovellukset ja elementit, joiden päällä sormi liikkuu. (Mozilla 2021e.)

Verkkopalveluiden käytön esimerkkejä mobiililaitteella, jossa käytetään Androidin TalkBack-ruudunlukijaa:

1. Sormen liikkuaessa näppäimistöllä lukee TalkBack kirjaimet käyttäjälle ääneen. Kun käyttäjä löytää halutun kirjaimen, voi näppäimistöstä päästää irti.
2. Vasemmalle tai oikealle pyyhkäisyllä voi siirtyä sivun eri elementteihin.
3. Sulava pyyhkäisy ylös oikealle avaa sisältövalikon.
4. Sisältövalikosta voi löytää esimerkiksi sivun otsikot.
5. Tuplaklikkaamalla ”sivun otsikot” -kohtaa, pyyhkäisy oikealle tai vasemmalle selaa nyt sivun eri otsikoita läpi. (Mozilla 2021e.)

2.3.2 Saavutettavuus kuulovammaisilla

Yli 5 % populaatiosta tai noin 432 miljoonaa aikuista ja 34 miljoonaa lasta ovat kuulovammaisia. On arvioitu, että vuoteen 2050 mennessä noin 700 miljoonaa ihmistä ovat kuulovammaisia. (World Health Organization 2021b.)

Kuurot ja heikkokuuloiset voivat tarvita tekstipohjaisen vaihtoehdon. Osa kuulovammaisista käyttää kuulolaitteita, mutta ne eivät ole kovin yleisiä. Kuulovammaisille saavutettavuus voidaan varmistaa kirjoittamalla tekstipohjaiset vaihtoehdot esimerkiksi mediapohjaiselle sisällölle. Kaiken äänipohjaisen informaation tulisi olla luettavissa ja videoissa täytyy olla tekstitykset. (Mozilla 2021c.)

Pelkät tekstitykset eivät kaikissa tapauksissa riitä. Tällöin on hyvä tarjota transkriptio. Transkriptiot ovat tekstipohjaisia versioita puheellisesta sekä puheettomasta sisällöstä. Transkriptiot ovat hyviä heikkokuuloisille ja kuuroille, jotta he voivat ymmärtää mitä tapahtuu esimerkiksi elokuvan kohtauksessa, jossa kuuluu ääniä mutta ei puhetta. Useimmat transkriptiot verkossa ovat HTML-muodossa. (Henry 2021b.)

Jotta verkkopalveluita voidaan käyttää tehokkaasti, kuulovammaiset tukeutuvat useasti transkriptioiden ja tekstitysten lisäksi mediasoittimiin, joissa on mahdollista muuttaa tekstin kokoa ja väriä, sekä mahdollisuus pysäyttää video ja säätää äänenvoimakkuutta. Osalle kuulovammaisista viittomakieli on heidän pääkielensä, ja he eivät lue tekstiä niin useasti. Sisällön tarjoaminen viittomakielellä ja selkokielellisesti kuvien, graafien ja taulukoiden avustamana voi auttaa sisällön ymmärtämisessä useaa ihmistä. (Abou-Zahra 2017.)

2.3.3 Saavutettavuus liikuntavammaisilla

Liikuntavammaisilla on liikkuvuuteen vaikuttavia vammoja, kuten esimerkiksi onnettomuuksissa, neurologisesti tai geneettisesti tulleet vammat. Vammat voivat vaikeuttaa käsien liikkuvuutta, minkä seurauksena hiiren tai näppäimistön käyttö voi hankaloitua. (Mozilla 2021c.)

Tietokonetta voi ohjata esimerkiksi silmän ja pään liikkeillä. Silmän liikkeillä ohjattava laite tunnistaa, mihin käyttäjä suuntaa katseensa. Avuksi on, että verkkosivujen painikkeet ovat tarpeeksi suuria. Klikkaus toimii niin, että käyttäjä katsoo kohdetta pidempään tai esimerkiksi ulkoisella valinnalla. Ulkoisessa valinnassa voidaan käyttää silmien räpäytykseen reagoivaa kytkintä. (Papunet 2021c.)

Muita apulaitteita ovat esimerkiksi:

- ergonomiset tai erikseen suunnitellut hiiret ja näppäimistöt
- pääosoitin ja suutikku
- näytöllä sijaitseva näppäimistö ja pallohiiri
- jalalla tai olkapäällä käytettävät kytkimet
- puhetunnistus, silmien liikkeen seuraaja ja muut lähestymistavat hands-free-tavat tietokoneen käyttöön. (Abou-Zahra 2017.)

Kuten näkövammaisille, myös liikuntavammaisille liian pienet välit linkkien välissä voivat olla epäkäytännöllisiä. Käyttäjä ei välttämättä osu kohteeseen tai saattaa klikata vahingossa viereistä linkkiä. (Nielsen & Pernice 2021, 69.)

Liikuntavammaiset voivat käyttää hiirtä tai hiirentapaisia laitteita ja näppäimistöä tai näppäimistön tapaisia laitteita verkkopalveluissa navigointiin. Palveluiden tulisi siis olla navigoitavissa pelkän näppäimistön avulla. Käyttäjille tulisi antaa tarpeeksi aikaa suorittaa palvelussa tehtäviä asioita. Lomakkeiden kenttien tulisi olla tarpeeksi isoja, jotta niitä on helpompi klikata. Lomakkeisiin syötettyjen arvojen korjaaminen kannattaa myös tehdä mahdollisimman helpoksi. Liikuntavammaisten lisäksi myös kognitiivisista- ja näkövammoista kärsivät henkilöt hyötyvät, kun nämä asiat huomioidaan. (Abou-Zahra 2017.)

2.3.4 Saavutettavuus kognitiivisesti vammautuneilla

Kognitiivisesta vajaatoiminnasta kärsiviin kuuluu ihmisiä laajalti. Heitä ovat esimerkiksi kehitysvammaiset, muistisairauksista kärsivät, ihmiset, joilla on mielisairaus sekä oppimisvaikeuden omaavat henkilöt. (Mozilla 2021c).

Heillä voi olla mitä tahansa hermostoon vaikuttavia ongelmia, jotka vaikuttavat kuuloon, liikkumiseen, näkemiseen, puhumiseen tai sisällön ymmärtämiseen. Kognitiiviset-, oppimis- tai neurologiset vammat eivät välttämättä vaikuta henkilön älykkyyteen. (Abou-Zahra 2017.)

Teknologia ja internet tarjoavat useita mahdollisuuksia ihmisille, joilla on kognitiivisia-, oppimis- tai neurologisia vammoja olla vuorovaikutuksessa sisällön kanssa tavoilla, jotka ovat heille hyödyllisempiä. Esimerkiksi käyttäjät voivat navigoida verkkosisältöä käyttäen erilaisia tekniikoita. He voivat käyttää sisältöä teksti-, audio- tai muissa muodoissa ja muuttaa sisällön esitystapaa yksilön tarpeiden mukaisesti. (Abou-Zahra 2017.)

Kognitiivisista vaikeuksista kärsivät ovat suurin saavutettavuudesta hyötyvistä käyttäjäryhmistä. Käyttäjäryhmän moninaisuuden takia on vaikea kattaa kaikkien tarpeet. Verkkopalvelun hyvä ja selkeä käytettävyys on saavutettavuuden kannalta tärkeää. Tekstin selkokieli on olennaista, sillä pelkästään Suomessa on arvioitu olevan noin 500 000 henkilöä, jotka sitä tarvitsevat. Esimerkiksi viranomaistekstien saatavuus selkokielenä on hyödyllistä. Tämä palvelee myös henkilöitä, joiden äidinkieli ei ole verkkopalvelussa käytettävä kieli. (Papunet 2021d.)

Käyttäjäryhmän moninaisuuden vuoksi monet apuvälineet ja saavutettavuusstandardit ovat käyttäjäryhmälle olennaisia. Osa käyttäjistä käyttää ohjelmistoja, jotka muuttavat puheen tekstiksi kuullakseen tekstisisällön, kun he samanaikaisesti lukevat sitä tai käyttävät tekstityksiä kuuntelemisen ohella. Jotkut puolestaan käyttävät työkaluja, jotka muuttavat tekstin kokoa, välistystä tai väriä lukemisen helpottamiseen. Osa käyttää kielioppia ja oikeinkirjoitusta avustavia työkaluja. (Abou-Zahra 2017.)

Saavutettavuuden parantamiseksi olisi hyvä huomioida selkeän yleiskielen käyttö. Monimutkaisten ja vaikeiden sanojen käytön voisi korvata synonyymeilla. Teksti tulee jakaa selkiin osiin. Apuna voi käyttää kuvia ja listoja. Verkkosivun elementtien yhtenäisenä pitäminen ja sivun rakenteen huomiointi on tärkeää. (Papunet 2021d.)

3 Verkkosisällön saavutettavuusohjeet (WCAG 2.1)

3.1 Mikä on WCAG?

Web Content Accessibility Guidelines eli verkkosisällön saavutettavuusohjeet ovat kehitetty W3C-prosessin avulla yhteistyössä alan yksilöiden ja organisaatioiden kanssa ympäri maailmaa, tarkoituksena tarjota yksi jaettu standardi verkkosisällön saavutettavuudelle, joka kattaa yksilöiden, organisaatioiden ja hallitusten tarpeet kansainvälisesti. WCAG-dokumentit selittävät, kuinka tehdä verkkosisällöstä saavutettavampaa ihmisille, joilla on vammoja. ”Verkkosisältö” yleisesti viittaa informaatioon verkkosivustossa tai -sovelluksessa. (Henry 2021c.)

Kuten aiemmin opinnäytetyössä mainittiin, Yhdysvaltojen hallituksen verkkosivujen mukaan Yhdysvalloilla on käytössään WCAG 2.0, joka on WCAG 2.1:tä edeltävä versio (USA Government 2021). Euroopassa puolestaan ohjenuorana saavutettavuuden kannalta käytetään nykyistä WCAG 2.1 -versiota (Euroopan komissio 2021). Suomessa laki velvoittaa noudattamaan WCAG 2.1:n ohjeistuksen A- ja AA-tason onnistumiskriteerejä (Aluehallintovirasto 2021a).

WCAG 2.1 sisältää laajalti suosituksia verkkopalveluiden saavutettavuuden takaamiseksi. Ohjenuorien seuraaminen tekee verkkopalveluista saavutettavimmat laajalle määrälle käyttäjiä, kuten näkövammaisille, kuulovammaisille, liikuntarajoittuneille, puhevammaisille, käyttäjille, joilla on valonherkkyyttä ja osittain käyttäjille, joilla on oppimisvaikeuksia tai kognitiivisia rajoituksia. (Henry 2021c.)

WCAG 2.1 on jatkettu versio edellisestä WCAG 2.0 -versiosta. Kaikki mitä löytyy versiosta 2.0, löytyy myös versiosta 2.1. WCAG 2.0 on julkaistu vuonna 2008, kun taas WCAG 2.1 on julkaistu vuonna 2018. WCAG 2.1 ei kuitenkaan suoraan korvaa aiempaa versiota, vaan kattaa samat asiat ja tarjoaa uusia vaihtoehtoja. W3C suosittelee käyttämään mieluummin versiota 2.1, jotta saavutettavuus voitaisiin maksimoida tulevaisuuden kannalta etukäteen. (Henry 2021c.)

WCAG 3 on työn alla, ja se on tällä hetkellä keskeneräinen luonnos. Sen on tarkoitettu kehittyvän W3C-standardiksi muutaman vuoden kuluessa, korvaamaan nykyisen 2.1-version. Ensimmäinen luonnos WCAG 3.0:sta julkaistiin 21.1.2021. WCAG 3 eroaa aiemmista versioista rakenteeltaan, vaatimusmalliltaan ja se on laajempi, eikä keskity ainoastaan verkkosisältöön. (Henry 2021d.)

WCAG 2.1 aloitettiin tavoitteena parantaa käytettävyyden opastusta kolmea merkittävää ryhmää varten: kognitiivisia tai oppimisvaikeuksia omaaville, heikonäköisille sekä vammaisille, jotka käyttävät mobiililaitteita. WCAG 2.1 on jatkettu versio 2.0:sta ja siihen on lisätty uusia onnistumiskriteerejä ja kuvaksia tukemaan kriteerejä. Tämä jatkettu versio takaa, että noudattamalla versiota 2.1, noudattaa automaattisesti myös versiota 2.0. On suositeltua käyttää uudempaa versiota. (WCAG 2.1 2018.)

3.2 WCAG 2.1:n ohjenuorat

Ohjenuoria on WCAG 2.1:ssä yhteensä 13. Noudattamalla ohjenuoria saadaan verkkopalveluista saavutettavimmat. Jokaista ohjenuoraa kohden, on olemassa testattavat onnistumiskriteerit, jotta WCAG:ta voidaan käyttää myös paikoissa, jossa käytännön testaaminen on välttämätöntä. Vaatimustasoja on olemassa kolme, jotta voidaan toimia erilaisten ryhmien tarpeiden mukaisesti. Vaatimustasot ovat A (matalin), AA ja AAA (korkein). (WCAG 2.1 2018.)

Jokaiselle ohjenuoralle ja vaatimuskriteerille on myös dokumentoitu erilaisia tekniikoita. Tekniikat ovat informatiivisia ja kuuluvat kahteen eri kategoriaan, niihin, jotka ovat kelpollisia täyttämään vaatimuskriteerit sekä niihin, jotka ovat tiedotta-

via ja hyviä käytäntöjä. Tiedottavat tekniikat menevät vaatimuskriteerejä pidemmälle. On huomioitava, että edes korkeimman vaatimustason implementointi ei silti tee verkkopalvelusta saavutettavaa kaikentyyppisille, asteisille, monivammaiselle tai kognitiivisia- ja oppimisvaikeuksia omaaville henkilöille. (WCAG 2.1 2018.)

Ohjenuorat jakautuvat neljään eri kategoriaan. Ohjenuorat sisältävät useita eri vaatimuksia ja vaatimustasoja, mutta niitä ei käydä tässä kohtaa läpi. Jokaisen kategorian alla on vielä lisäksi onnistumiskriteerit, joita tässä listauksessa ei mainittu. Suomessa lain velvoittamia onnistumiskriteereitä on yhteensä 49 kappaletta. WCAG 2.1:n verkkopalveluiden ohjenuorat jakautuvat seuraaviin kategorioihin:

1. Havaittava
 - 1.1 Tekstivastineet
 - 1.2 Aikasidonnainen media
 - 1.3 Mukautettava
 - 1.4 Erottuva
 2. Hallittava
 - 2.1 Käytettävissä näppäimistöllä
 - 2.2 Tarpeeksi aikaa
 - 2.3 Sairauskohtaukset
 - 2.4 Navigoitava
 - 2.5 Syötetävät
 3. Ymmärrettävä
 - 3.1 Luettava
 - 3.2 Ennakoitava
 - 3.3 Syötteen avustaminen
 4. Toimintavarma
 - 4.1 Yhteensopiva
- (Aluehallintovirasto 2021a.)

Aluehallintoviraston saavutettavuusvaatimukset.fi -verkkosivustolta löytyy tarkempi määrittely, jossa listataan mitä palveluja ja sisältöjä laki koskee, ja erillinen listaus palveluille ja sisällöille, joita laki ei koske. Laki koskee kuitenkin suurinta osaa verkkosivuja sekä niiden toimintojaan. ”Ennen 23.9.2019 arkistoituun verkkosivustoon tai sen arkistoituun sisältöön ei sovelleta saavutettavuusvaatimuksia. Kun verkkosivusto tai sen sisältö arkistoidaan 23.9.2019 tai sen jälkeen, tulee saavutettavuusvaatimuksia noudattaa”. (Aluehallintovirasto 2021b.)

Saavutettavuusvaatimukset.fi -verkkosivustolta löytyy myös tarkempi määrittely siitä, ketä saavutettavuus koskee. Yleisesti ottaen laki velvoittaa viranomaisia,

julkisoikeudellisia laitoksia ja osaa järjestöistä. Osa yksityisestä sektorista, kuten pankit ja vakuutusyhtiöt ovat myös velvollisia noudattamaan saavutettavuusvaatimuksia. Huomioitavaa tämän opinnäytetyön kannalta on, että ammatikorkeakoulut kuuluvat lain piiriin, ja tulisi heidän noudattaa WCAG 2.1:ssä määriteltyjä saavutettavuusvaatimuksia. (Aluehallintovirasto 2021c.)

4 Saavutettavuus kehittäjän näkökulmasta

4.1 HTML:n saavutettavuus

4.1.1 Semanttiset HTML-elementit

Suuri osa verkkosivujen sisällöstä voidaan tehdä saavutettavaksi käyttämällä oikeita HTML-elementtejä. Saavutettavuuden takaamiseksi on käytettävä HTML:n semanttisia elementtejä. Tämä tarkoittaa oikeiden elementtien käyttöä, niille tarkoitetuissa tilanteissa. (Mozilla 2021f.)

Teoriassa CSS:n ja JavaScriptin avulla voi esimerkiksi painikkeena käyttää jotain muutakin kuin sille tarkoitettua elementtiä (kuva 1). <Button>-elementissä on kuitenkin valmiina sopiva tyyllittely sekä sisäänrakennettu näppäimistösaavutettavuus. Käyttäjät voivat navigoida painikkeiden välillä käyttämällä Tabulaattori-näppäintä ja aktivoida valintansa käyttämällä Enter-näppäintä. (Mozilla 2021f.)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Otsikko</title>
  </head>
  <body>
    <div>Play video</div>
    <button>Play video</button>
  </body>
</html>
```

Kuva 1. Esimerkki HTML-elementtien käytöstä.

Semanttisten elementtien käyttö on kehittäjän näkökulmasta ajallisesti yhtä nopeaa, kuin ei-semanttisten elementtien käyttö. Semanttisten elementtien käyttö myös helpottaa kehittäjän työtä, sillä semanttiset elementit tuovat mukanaan valmista toiminnallisuutta ja niitä on helpompi ymmärtää ja tulkita. Valmiin toiminnallisuuden esimerkkinä toimii esimerkiksi äsken mainittu näppäimistötoiminnallisuus. Semanttiset elementit ovat myös parempia mobiililaitteilla. Semanttinen HTML on kevyempää tiedostomuodossa ja helpompaa tehdä responsiiviseksi mobiililaitteille. Tämän lisäksi semanttiset elementit ovat parempia, kun halutaan parempi hakukonenäkyvyys. Hakukoneet löytävät semanttisia elementtejä käyttäviä verkkosivustoja paremmin. (Mozilla 2021f.)

Ruudunlukijan yksi parhaista tuista on hyvä sisältörakenne, esimerkiksi otsikot, kappaleet ja listat. Kuvassa 2 on esimerkki siitä, miltä erinomainen HTML-elementtien käyttö näyttää. Ruudunlukijan käydessä läpi sivuston sisältöä (kuva 2.), se lukee jokaisen otsikon ja kappaleen käyttäjälleen ääneen ja kertoo mikä kyseessä oleva elementti on. Ruudunlukija pysähtyy jokaisen elementin jälkeen ja antaa käyttäjän valita tahdin, jolla edetään. Monissa ruudunlukijoissa voi myös hypätä suoraan seuraavaan tai edelliseen otsikkoon. Useat ruudunlukijat antavat myös käyttäjän tuoda esiin listan kaikista sivuston otsikoista, jotta käyttäjä voi navigoida suoraan itselleen mieleiseen kappaleeseen. (Mozilla 2021f.)

```
<body>
  <h1>Otsikko</h1>

  <p>Ensimmäinen kappale.</p>

  <p>Toinen kappale.</p>

  <ol>
    <li>Listan 1. elementti</li>
    <li>Listan 2. elementti</li>
    <li>Listan 3. elementti</li>
  </ol>

  <h2>Aliotsikko</h2>

  <p>Aliotsikon alainen kappale</p>

  <h2>Toinen aliotsikko</h2>

  <p>Toinen aliotsikon alainen kappale.</p>
</body>
</html>
```

Kuva 2. Hyvän semantiikan esimerkki.

Ruudunlukijan läpi käydessä sivuston sisältöä (kuva 3), kokemus ei ole yhtä hyvä. Ruudunlukijalla ei ole mitään tietoa mikä sivuston sisällöstä on mitään, joten käyttäjä ei saa hyödyllistä tietoa sivuston sisällöstä. Koko sivusto on ruudunlukijalle vain yksi iso kappale, jonka se lukee putkeen ilman taukoja. Tämän lisäksi sivustoa on vaikeampi tyylitellä CSS-tyylitiedostoilla, tai muokata funktionaalisuutta JavaScriptillä. (Mozilla 2021f).

```

<body>
  <font size="7">My heading</font>
  <br><br>
  This is the first section of my document.
  <br><br>
  I'll add another paragraph here too.
  <br><br>
  1. Here is
  <br><br>
  2. a list for
  <br><br>
  3. you to read
  <br><br>
  <font size="5">My subheading</font>
  <br><br>
  This is the first subsection of my document. I'd love people to be able to find this content!
  <br><br>
  <font size="5">My 2nd subheading</font>
  <br><br>
  This is the second subsection of my content. I think is more interesting than the last one.
</body>

```

Kuva 3. Huonon semantiikan esimerkki.

Selkeä kielenkäyttö on myös saavutettavuuden kannalta hyödyksi monille. Yleisesti tulisi käyttää mahdollisimman selkeitä ja yksinkertaisia lauseita sekä termejä. Ruudunlukijan kannalta esimerkiksi lyhenteiden käyttöä tulisi välttää. Englanninkielisessä sivustossa kannattaa esimerkiksi kirjoittaa kuukauden nimet kokonaisuudessaan, kolmen ensimmäisen kirjaimen sijasta, kuten usein on tapana. Väliviivojen käyttöä kannattaa välttää tilanteissa, joissa se vain on mahdollista. Lyhenteet kuten HTML, kannattaa ilmaista vähintään muutaman kerran kokonaisuudessaan "Hypertext Markup Language":na. (Mozilla 2021f.)

Vuonna 2014 HTML5:n mukana tuli uusia semanttisia elementtejä, kuten <nav>, <section> ja <article> ja monia muita sivustorakennetta kuvaavia elementtejä. Uudet semanttiset elementit toimivat lähes samalla tavalla kuin kehittäjille entuudestaan tutut <div>-elementit. Uudet semanttiset elementit ovat parempia siitä, että jo pelkkä lähdekoodi kertoo mitä sivuston mikäkin kohta sisältää. Kehittäjän ja ruudunlukijan on helpompi lukea alempana olevaa lähdekoodia alkaen <header>-elementistä, kuin sen yläpuolella olevia sisäkkäisiä <div>-elementtejä (kuva 4). (freeCodeCamp 2019.)

```

<div id="header"></div>
<div class="section">
  <div class="article">
    <div class="figure">
      <img>
      <div class="figcaption"></div>
    </div>
  </div>
</div>
<div id="footer"></div>

<header></header>
<section>
  <article>
    <figure>
      <img>
      <figcaption>
      </figcaption>
    </figure>
  </article>
</section>
<footer></footer>

```

Kuva 4. Semanttisten vs. ei-semanttisen elementtien käyttö lähdekoodissa. (freeCodeCamp 2019.)

Hakukoneet ja apulaitteet, kuten ruudunlukija ymmärtävät paremmin sivustorakennetta, kun sivusto käyttää semanttisia elementtejä. Konventioista riippuen myös <div>-elementtien käyttö voi olla erilaista eri kehittäjillä. Esimerkiksi kuvassa 4 ylin rivi <div id="header"> voitaisiin tietyissä piireissä tai organisaatioissa kirjoittaa id:n sijasta HTML:n luokkamääritteellä <div class="header">. Vastaavia tapoja on useita, mutta semanttisia elementtejä, kuten <header>-elementtiä käyttämällä taataan parempi ymmärrys sekä kehittäjille että käyttäjille. (freeCodeCamp 2019.)

Ennen vanhaan HTML-sivustojen rakenteet on usein tehty HTML:n table-elementeillä (kuva 5). Tämä on ollut järkevämpää silloin, kun CSS-tuki oli vähäistä. Nykyisin ne ovat vain epäkäytännöllisiä ruudunlukijoille ja kehittäjille. <table>-elementit vaativat paljon enemmän koodia, jonka myötä niitä on vaikeampi ylläpitää. Ruudunlukijalla navigoidessa <table>-elementeillä rakennettua sivua se todennäköisesti kertoo käyttäjälle, että kyseessä on taulukko. Toinen huomioitava asia sivuston rakenteen kannalta on konventio, jossa käytetään pelkästään sisäkkäisiä <div>-elementtejä. Sisäkkäisten <div>-elementtien sijasta on järkevämpää käyttää tilanteeseen sopivia semanttisia elementtejä, kuten <nav>-

<footer>- ja <article>-elementtejä. Näiden semanttisten elementtien käyttö parantaa ruudunlukijoiden saavutettavuutta, jotka voivat antaa käyttäjälle parempaa tietoa navigoinnin suhteen. (Mozilla 2021f.)

```

<body>
  <table width="1200">
    <!-- main heading row -->
    <tr id="heading">
      <td colspan="6">
        <h1 align="center">Otsikko</h1>
      </td>
    </tr>
    <!-- nav menu row -->
    <tr id="nav" bgcolor="#ffffff">
      <td width="200">
        <a href="#" align="center">Navigointilinkki 1</a>
      </td>
      <td width="200">
        <a href="#" align="center">Navigointilinkki 2</a>
      </td>
      <td width="200">
        <a href="#" align="center">Navigointilinkki 3</a>
      </td>
      <td width="200">
        <a href="#" align="center">Navigointilinkki 4</a>
      </td>
      <td width="300">
        <form width="300">
          <input type="search" name="q" placeholder="Search query" width="300">
        </form>
      </td>
      <td width="100">
        <button width="100">Go!</button>
      </td>
    </tr>
  </table>

```

Kuva 5. Esimerkki <table>-elementeillä rakennetusta sivustosta.

Hyvän semanttisuuden ja visuaalisen rakenteen lisäksi sisällön tulisi olla lähdekoodiltaan loogisessa järjestyksessä. CSS:n avulla on mahdollista tehdä epäloogisessa järjestyksessä olevasta lähdekoodista visuaalisesti sivustorakenteeltaan järkevän näköinen sivusto. Ruudunlukijoille se ei kuitenkaan ole järkevää, sillä ne lukevat HTML-rakenteen käyttäjälleen ylhäältä alas. Jos esimerkiksi otsikko on sijoitettu lähdekoodissa ennen kappaleen tekstiä, ruudunlukija lukee kappaleen tekstin ennen otsikkoa. (Mozilla 2021f.)

4.1.2 Näppäimistönavigointi

Verkkoselaimet antavat oletusarvoisesti käyttäjän ohjata sivuja pelkällä näppäimistöllä. Kehittäjien pitää pääasiassa vain käyttää oikeita elementtejä, jotta näppäimistön käyttö toimii kuten pitää. Tabulaattorilla voidaan siirtyä seuraavaan elementtiin. Shift + tabulaattori puolestaan siirtyy edelliseen elementtiin. Tabulaattorilla navigoidessa esimerkiksi sivun linkkeihin tai painikkeisiin, voidaan linkit avata ja painikkeita painaa Enter-näppäimellä. (kuva 6). Enter-näppäimellä voidaan avata myös lomakkeessa oleva pudotusvalikko/<select>-elementti, valita nuolinäppäimillä sieltä arvo ja vahvistaa valinta enterillä. Kaikki tämä toiminnallisuus saadaan vaivattomasti käyttämällä oikeita elementtejä sivuston kehityksessä. Mikäli käyttäjät käyttävät esimerkiksi <div>-elementtiä <button>-elementin sijaan, joudutaan näppäimistökäytettävyys luoda JavaScriptillä uudestaan. (Mozilla 2021f.)

Links

This is a link to [Mozilla](#).

Another link, to the [Mozilla Developer Network](#).

Buttons

Form

Fill in your name:

Enter your age:

Choose your mood: ▼

Kuva 6. Näppäimistöllä käytettävissä oleva verkkosivu (Mozilla 2021f).

<div>-elementit eivät oletusarvoisesti sisällä näppäimistötoiminnallisuutta, sillä niihin ei yleensä ole syytä navigoida. Kuvassa 7 <div>-elementtejä käytetään <button>-elementtien korvikkeena. Tämä on huono käytäntö, mutta kuitenkin mahdollista. Attribuutilla "tabindex=0" voidaan lisätä mahdollisuus navigoida kyseiseen elementtiin näppäimistöllä. Tabindex on pääasiassa tarkoitettu siihen, että näppäimistöllä elementtien selausjärjestystä voidaan muuttaa. Tämä on

olennaista esimerkiksi tapauksissa, jolloin elementti ilmenee lähdekoodissa eri järjestyksessä, kuin käyttäjälle visuaalisesti sivulla. Tabindex ei kuitenkaan itsessään mahdollista Enterillä elementin kanssa vuorovaikuttamista, vaan pitää klikkaustoiminto lisätä JavaScriptillä. (Mozilla 2021f.)

```
<div data-message="This is from the first button" tabindex="0">Click me!</div>
<div data-message="This is from the second button" tabindex="0">Click me too!</div>
<div data-message="This is from the third button" tabindex="0">And me!</div>
```

Kuva 7. Näppäimistötoiminnallisuuden lisääminen.

4.1.3 HTML:n hyvät käytännöt

Verkkosivujen kehityksessä on tärkeää muistaa kirjoittaa linkkitekstit niin, että ne ovat ymmärrettävissä ilman kontekstia. Kuvassa 8 esimerkiksi ylempi linkkiteksti on siitä huono, että ruudunlukijat voivat luoda listan sivun linkeistä. Tässä tapauksessa ylemmällä tavalla tehtynä linkiksi tulisi lukemaan vain "click here". Alempi linkkiteksti "Find out more about whales" on paljon helpommin ymmärrettävissä ilman kontekstia. (Mozilla 2021f.)

```
<p>Whales are really awesome creatures. To find more out about whales, <a href="whales.html">click here</a>.</p>
<p>Whales are really awesome creatures. <a href="whales.html">Find out more about whales</a>.</p>
```

Kuva 8. Esimerkki linkkitekstin käytöstä.

Lomakkeiden kenttäotsikot ovat myös olennainen osa saavutettavuutta. Kenttäotsikoiden avulla käyttäjä saa tietoonsa, mitä kuhunkin syötekenttään tulee kirjoittaa. Kenttäotsikot kannattaa laittaa erikseen <label>-elementillä, jotta ruudunlukijat tietävät kenttäotsikon olevan nimenomaan tietylle kentälle tarkoitettu. Suurimmassa osassa selaimia klikkaamalla kenttäotsikkoa, fokusoituu lomakkeen kenttä, joka tekee siis klikattavasta kohteesta helpommin klikattavan ja suuremman. (Mozilla 2021f.)

Kuvan 9 ja kuvan 10 taulurakenne näyttää visuaalisesti ymmärrettävältä, mutta siinä on saavutettavuuden kannalta ongelmia. Ruudunlukija ei pysty yhdistämään rivejä tai sarakkeita yhtenäiseksi datajoukoksi. Tätä varten on käytettävä erikseen "table header" eli <th>-elementtejä. Table header eli <th>-elementeille voidaan määritellä ovatko ne otsikoita riveille vai sarakkeille käyttämällä "scope"

-attribuuttia. Attribuutin käyttö antaa ruudunlukijalle kokonaisia loogisia data-joukkoja. Myös elementit `<caption>` ja `<table summary="">` ovat saavutettavuuden kannalta samankaltaisia vaihtoehtoja. Ne toimivat vaihtoehtoisena tekstinä taulukolle, jonka ruudunlukija voi lukea käyttäjälle. (Mozilla 2021f.)

```

<table>
  <tr>
    <td>Nimi</td>
    <td>Ikä</td>
    <td>Sukupuoli</td>
  </tr>
  <tr>
    <td>Nimi1</td>
    <td>25</td>
    <td>Mies</td>
  </tr>
  <tr>
    <td>Nimi2</td>
    <td>35</td>
    <td>Nainen</td>
  </tr>
  <tr>
    <td>Nimi3</td>
    <td>45</td>
    <td>Nainen</td>
  </tr>
</table>

```

Kuva 9. Saavutettavuuden kannalta huono taulurakenne HTML:ssä.

Taulu

Nimi	Ikä	Sukupuoli
Nimi1	25	Mies
Nimi2	35	Nainen
Nimi3	45	Nainen

Kuva 10. Kuvan 9 taulurakenne verkkosivulla.

Ruudunlukijan lukiessa ylimmän (kuva 11) ``-kuvaelementin sisällön käyttäjälle, se lukee ääneen pelkän tiedostonimen "henkilöt.png, kuva". Tämä välittää käyttäjälle vain tiedon, että kuvassa on mahdollisesti henkilöitä. Usein kuitenkin tiedostonimet ovat automaattisesti generoituja esimerkiksi kameran toimesta, ja niistä ei ruudunlukijan käyttäjä saa minkäänlaista kontekstia siihen, mitä kuvassa tapahtuu. (Mozilla 2021f.)

Ruudunlukijan lukiessa toiseksi ylimmän (kuva 11) -kuvaelementin käyttäjälle, se lukee tiedostonimen lisäksi myös alt-attribuutin, jonka kehittäjä on siihen kirjoittanut. Tässä tapauksessa siis ”Kuvassa on henkilö, joka juo vettä. Hänen vieressään istuu toinen henkilö, joka lukee lehteä”. On huomioitavaa, että alt-tekstissä tulee kuvailla mahdollisimman tarkasti, mutta lyhyesti kuvan sisältöä. Kaikki ylimääräinen tulisi jättää alt-tekstistä pois. Ylimääräinen teksti kannattaa sijoittaa esimerkiksi tekstiksi kuvan läheisyyteen tai title-attribuuttiin, kuten kolmannessa -elementissä on tehty (kuva 10). (Mozilla 2021f.)

Neljännessä -elementissä (kuva 11) käytetään alt-tekstin sijasta <p>-elementtiä, jonka ”id”-arvo annetaan -elementille ”aria-labelledby”-attribuutin arvoksi. Ruudunlukija kohtelee tässä tapauksessa <p>-elementtiä kuvan alt-tekstinä. Tämä on erittäin hyödyllistä esimerkiksi tilanteissa, jolloin halutaan käyttää useammassa kuvassa samaa alt-tekstiä, koska se ei ole mahdollista alt-tekstillä. (Mozilla 2021f.)

```




<p id="henkilöt-label">Kuvassa on henkilö, joka juo vettä. Hänen vieressään istuu toinen henkilö, joka lukee lehteä.</p>
```

Kuva 11. Esimerkki kuvien saavutettavuudesta.

Yhtenä vaihtoehtona on myös attribuutti ”longdesc”. Longdesciä käytetään, kun halutaan antaa tekstille pidempi selitys kuin alt-tekstissä. Longdescillä voidaan myös viitata esimerkiksi toiseen html-dokumenttiin. Monet ruudunlukijat eivät kuitenkaan tue longdesc-attribuuttia, eikä sitä kannata tämän takia käyttää. (Mozilla 2021f.)

On myös olemassa tilanteita, jolloin kuvia käytetään sivustolla esimerkiksi sivun taustakuvana tai muuna koristeena, joka ei liity sivuston tarjoamaan sisältöön mitenkään. Kyseisissä tilanteissa voi olla parempi jättää alt-teksti kokonaan pois. Vaihtoehtoisesti koristeellisille kuvaelementeille voidaan myös antaa attribuutiksi role=”presentation”, joka estää ruudunlukijaa lukemasta alt-tekstiä.

Mozilla suosittelee, että vain koristeelliseen tarkoitukseen olevat kuvat kannattaisi sisällyttää pelkän CSS:n avulla, jos vain mahdollista. (Mozilla 2021f.)

HTML:ssä on jo valmiiksi olemassa kontrollit video- ja audiotiedostoille. Ne näyttävät hieman erilaisilta riippuen käytettävästä selaimesta. Valmiiksi toimintoina tulee play/pause -nappula, kelaustoiminto, äänenvoimakkuuden säätö ja kokonäytön tila. Suurimassa osassa selaimia ohjauspainikkeet eivät ole näppäimistöllä saavutettavia. Joissain selaimissa osittain, mutta ei silti riittävästi. Selaimet tyyllittelevät kontrollit valmiiksi, eikä niiden tyyllittelyä ole mahdollista muuttaa. Kehittäjät voivat kuitenkin tehdä omat audio- ja videokontrollinsa HTML:llä ja JavaScriptillä. Tämä on pienen vaivannäön takana, mutta esimerkiksi Mozilla tarjoaa siihen erinomaiset ohjeet. (Mozilla 2021g.)

4.1.4 Värit ja kontrasti

Linkit eroavat oletusarvoisesti HTML:ssä väriltään ja koristelultaan muusta tekstistä. HTML:n oletustyyliellä varustetut linkit ovat alleviivattuja ja sinisiä, jotka muuttuvat väriltään violetiksi, kun käyttäjä on klikannut linkkiä. Linkeissä on myös valmiiksi fokusta indikoiva ympyrä, kun käyttäjä navigoi näppäimistöllä linkkiin (kuva 12). (Mozilla 2021f.)



Kuva 12. Linkki, johon on navigoitu näppäimistöllä.

WCAG 2.1:ssä on ohjeistus erilaisten käyttöliittymäkomponenttien vaadittaviin kontrastisuhteisiin. Kontrastisuhteet ovat määritelty AA- ja AAA-tason vaatimuksina (taulukko 3). (Mozilla 2021h.)

Sisällön tyyppi	Minimi suhde (AA-taso)	Paranneltu suhde (AAA-taso)
Leipäteksti	4.5: 1	7: 1
Suuremman skaalan teksti (120–150% suurempi kuin leipä teksti)	3: 1	4.5: 1
Aktiiviset käyttöliittymä-komponentit ja graafiset objektit, kuten ikonit ja kaaviot	3: 1	Ei määritelty

Taulukko 3. WCAG:n ohjeistamat värikontrastit. (Mozilla 2021h.)

Pelkkä väriero muuhun tekstiin ei riitä, vaan kuten kaikessa muussakin tekstissä linkkitekstin värin tulee olla selvästi erotettavissa sivun taustaväristä. WCAG:n määrittelemät kontrastisuhteet eivät kuitenkaan liity koristeellisiin elementteihin. (Mozilla 2021h.)

Ideaalisissa tapauksissa sivuston väreiksi kannattaa valita AAA-tason kontrastivaatimukseen sopivat värit. Huonon kontrastisuhteen tekstiä on vaikeampi lukea (kuva 13). Internetistä löytyy esimerkiksi useita sivustoja, joilla voi testata oman HTML-dokumentin kontrastien saavutettavuutta. Esimerkiksi WebAIMilla on hyvä sivusto tätä varten. (Mozilla 2021h.)



Kuva 13. Hyvän ja huonon kontrastin esimerkit.

4.2 JavaScript ja CSS

JavaScript ja CSS eivät ole saavutettavuuden kannalta yhtä merkittävässä roolissa kuin HTML, mutta voivat silti vaikuttaa saavutettavuuteen joko negatiivi-

sesti tai positiivisesti riippuen niiden käyttötavoista. Parhaiden käytäntöjen noudattaminen takaa sen, että CSS tai JavaScript ei vahingossakaan pilaa sivun saavutettavuutta. (Mozilla 2021i.)

CSS:llä voidaan tehdä muutoksia kaikkiin HTML-elementteihin, mutta niitä ei kuitenkaan kannata muuttaa liikaa. Esimerkiksi fonttikoot, välistykset jne. kannattaa pitää järkevinä, jotta tekstiä on helppo lukea. Otsikoiden tulee erottua leipätekstistä selvästi. Tekstin ja taustan kontrastin tulee olla WCAG:n ohjeiden mukainen. Tyypillisesti leipätekstiä ja otsikoita voitaisiin CSS:llä muuttaa kuten kuvassa 14. (Mozilla 2021i.)

```
h1 {
  font-size: 5rem;
}

p,
li {
  line-height: 1.5;
  font-size: 1.6rem;
}
```

Kuva 14. Esimerkki leipätekstin ja otsikoiden CSS-tyylittelyistä.

HTML-kappaleen luvussa 4.1.4 Värit ja kontrasti käytiin läpi linkkien oletusarvoisia tyylejä. Linkkien tyyllittelyssä kannattaa muistaa oletusarvoisten tyylien mukaisesti hyvät käytännöt ja kertoa käyttäjälle linkkien tila. Linkkien tyylien tulisi muuttua esimerkiksi, kun käyttäjä laittaa hiirenosoittimen linkin päälle, tai jos linkissä on jo vierailtu. Fokusta indikoivaa kehikkoa ei myöskään kannata ottaa valmiista tyyllittelyistä pois, sillä se helpottaa näppäimistö navigointia sivulla. (Mozilla 2021i.)

Lomakkeiden tyyllittelyssä yleensä olennaisimpia muutoksia ovat elementtien koot ja tekstikenttien ryhmittäminen järkevästi. Kuten linkeissä, lomakkeiden syötekenttien tulisi kertoa käyttäjälle, kun ne ovat fokuksen alaisena. Syötekenttien oletustyyliessä on jo tilan mukainen tyyllittely valmiina. Sitä voi muokata halutessaan, mutta sitä ei kannata poistaa kokonaan. (Mozilla 2021i.)

Taulukoiden tyylittelyissä yleensä sovitetaan taulukko sopimaan sivun muihin elementteihin väreiltään ja tyyleiltään. Taulukoiden tyylittelyssä kannattaa esimerkiksi rivien ja sarakkeiden otsikkotekstit saada selkeästi erottumaan muusta tekstistä. Joka toisen rivin taustan pieni väriero auttaa myös hyvin erottamaan rivit toisistaan (kuva 15). (Mozilla 2021i.)

Tuote	Hinta	Varastossa
T-paita	10€	27 kpl
Farkut	40€	14 kpl
Hattu	20€	35 kpl

Kuva 15. Esimerkki taulukon CSS-tyylittelystä.

Usein sivun elementtejä halutaan piilottaa käyttäjältä, jotta tietyt sisällöt ilmaantuvat vasta esimerkiksi klikatessa jotain sivulla. Sivulla voi olla esimerkiksi välilehtiä, joita klikkaamalla piilotetaan vanha sisältö ja saadaan uusi sisältö näkyviin (kuva 16). Esimerkissä välilehdet laitetaan päällekkäin, mutta ruudunlukijat eivät välitä tästä. Ruudunlukijoille merkityksellistä on lähdekoodin järjestys. Tässä tapauksessa siis CSS-sääntö "position: absolute" toimii loistavasti. Kyseistä CSS-sääntöä pidetään yhtenä parhaista tavoista sisällön piilottamiseen, koska se ei ole haitaksi ruudunlukijoille. Huonoja käytäntöjä puolestaan ovat CSS-säännöt "visibility: hidden" ja "display: none", sillä ne piilottavat sisällön ruudunlukijoilta. Poikkeuksena se, jos haluaa piilottaa sisällön ruudunlukijalta. (Mozilla 2021i.)



Kuva 16. Esimerkki sisällön piilottamisesta verkkosivulla.

JavaScriptilläkin on mahdollista vaikuttaa saavutettavuuteen positiivisesti tai negatiivisesti. Aiemmin tässä opinnäytetyössä puhuttiin esimerkiksi näppäimistötoiminnallisuuden luonnista. Saavutettavuuden parantaminen JavaScriptillä on tietenkin hyvä asia, mutta esimerkiksi näppäimistötoiminnallisuutta ei aina ole tarpeen luoda uudestaan, jos vain käyttää oikeita HTML-elementtejä, joista se jo valmiiksi löytyy. JavaScriptillä saavutettavuuteen vaikuttaminen voidaan minimoida käyttämällä semanttisia HTML-elementtejä niille kuuluvissa tilanteissa. (Mozilla 2021i.)

JavaScriptissä halutaan usein reagoida käyttäjän syötteisiin verkkosivulla, esimerkiksi hiirenklikkaukseen tietyssä HTML-elementissä. Termi näille on "event handler". Monet event handlerit eivät ole saavutettavia, kuten esimerkiksi "mouseover", "mouseout" ja "dblclick". "Mouseover" -tapahtuma reagoi siihen, kun käyttäjä siirtää hiiren tietyn elementin päälle. "Mouseout" puolestaan siihen, kun hiiri viedään tietyn elementin päältä pois. "Dblick" tarkoittaa sitä, kun käyttäjä tuplaklikkaa elementtiä. Mikään kyseisistä tapahtumista ei reagoi esimerkiksi näppäimistöllä tehtyihin painalluksiin. Tapahtumat "mouseover" ja

”mouseout” voidaan korvata ”focus” ja ”blur” -tapahtumilla, jotka reagoivat siihen onko elementti fokuksen alaisena. (Mozilla 2021i.)

4.3 WAI-ARIA

WAI-ARIA eli Web Accessibility Initiativen luoma ”Accessible Rich Internet Applications Suite” on avustava työkalu sivustojen saavutettavuuden luonnissa. WAI-ARIAssa tulee uusia avustavia attribuutteja, joita voidaan käyttää HTML-elementeissä saavutettavuuden parantamiseksi. Pääasiassa se auttaa varsinkin dynaamisen sisällön, sekä monimutkaisempien käyttöliittymäelementtien saavutettavuuden takaamisessa. Ilman WAI-ARIAa osa verkkosivujen toiminnallisuudesta jäisi saavuttamattomaksi osalle käyttäjiä, erityisesti ruudunlukijakäyttäjille ja käyttäjille, jotka eivät voi käyttää hiirtä. (Cooper 2020.)

WAI-ARIA:n tuki on hieman monimutkainen asia, sillä käyttöjärjestelmiä, verkkoselaimia, ruudunlukijoita ja näiden kaikkien yhdistelmiä on paljon. Tämän lisäksi WAI-ARIAssa on erittäin paljon ominaisuuksia, ja esimerkiksi jotkut ominaisuudet saattavat toimia tietyissä ruudunlukijoissa tai selaimissa ja toiset taas ei. Yleisesti selaintuki on todella hyvä, mutta tuki ruudunlukijoille ei ole vielä selainten tasolla. (Mozilla 2021j.)

Monimutkaisempien käyttöliittymäelementtien luonnissa voi olla käytössä esimerkiksi HTML-elementtejä ja dynaamista JavaScriptiä, joka voi tehdä saavutettavuuden implementoinnista hankalaa. WAI-ARIA voi nimenomaan auttaa kyseisissä tilanteissa. Verkkosivut ovat ajan kanssa muuttuneet dynaamisimmiksi ja monimutkaisemmiksi, joka on tuonut mukanaan uusia haasteita ja ongelmia saavutettavuuden kannalta. (Mozilla 2021j.)

HTML5:n mukana tuli uusia monimutkaisempia elementtejä kuten kalentereita, joista käyttäjä voi valita päivämäärän ja mahdollisesti ajan. Selaimet eivät tue näitä hyvin ja niitä on vaikea tyylitellä, sillä selaimet pakottavat elementteihin omia tyylejensä. Tämän seurauksena kehittäjät monesti käyttävät kolmannen osapuolen JavaScript-kirjastoja, joissa toteutukset useasti ovat sisäkkäisillä <div>-elementeillä luotuja. Visuaalisesti ne toimivat hyvin, mutta ruudunlukijat eivät ymmärrä niitä. (Mozilla 2021j.)

WAI-ARIAssa on rooliattribuutteja, jotka toimivat HTML5:n semanttisten rakenteellisten elementtien tyyllisesti (kuva 17). Rooleja on monia erilaisia, joita ei välttämättä löydy HTML5:stä, kuten "search" ja "banner". (Mozilla 2021j.)

```
<div role="navigation"></div>  
<nav></nav>
```

Kuva 17. WAI-ARIAn role-attribuutti.

ARIA-roolit tuovat siis elementteihin semanttista tarkoitusta, jotta ruudunlukijat ja muut apulaitteet voisivat ymmärtää sivuston rakennetta paremmin. ARIA-rooleja voidaan käyttää kuvaamaan elementtejä, joita HTML:ssä ei ole valmiina tai joita selaimet eivät vielä täysin tue. Semanttisilla elementeillä on jo valmiiksi oma roolinsa, mutta ei-semanttisilla elementeillä kuten <div>-elementillä ei. Ruudunlukijat lukevat ARIA-roolit käyttäjälleen ääneen. Esimerkiksi käyttäessä järjestämätöntä listaa, ja asettamalla sen rooliksi "tabpanel", lukee ruudunlukija sen käyttäjälleen välilehtipaneelina. Kehittäjän vastuulle jää roolien oikeinkäyttö, jos esimerkiksi roolina on välilehtipaneeli, mutta välilehtiä ei ole on saavutettavuuteen vaikuttettu vain negatiivisesti. (Mozilla 2021k.)

Elementeille voidaan antaa myös erilaisia arvoja. Esimerkiksi lomakkeelle voidaan antaa arvo `aria-required="true"`, joka määrittää lomakkeen täyttämisen pakolliseksi käyttäjälle. WAI-ARIAlla voidaan myös asettaa elementeille tiloja, kuten "disabled". Ruudunlukijalle voidaan ilmoittaa `aria-disabled="true"` -arvolla, että esimerkiksi lomakkeen kenttä on poissa käytöstä. (Mozilla 2021j.)

WAI-ARIAa tulisi käyttää ainoastaan silloin, kun on tarve. Natiivien ja semanttisten HTML-elementtien käyttö on suositeltua. Niillä voidaan minimoida WAI-ARIAn tarve. WAI-ARIAa on hyvä käyttää esimerkiksi elementtien roolien merkkäamiseen, kuten haku, välilehtijoukko, välilehti jne. Dynaamisten muutosten kanssa on hyvä käyttää `aria-live` attribuuttia. Ruudunlukijoiden on vaikea pysyä jatkuvasti muuttuvien elementtien perässä, jolloin `aria-live` ilmoittaa ruudunlukijalle muutoksesta. (Mozilla 2021j.)

4.4 Mobiililaitteet ja saavutettavuus

Perficientin tutkimuksessa selvisi, että globaalisti vuonna 2020 kaikista verkkosivujen vierailuista 68.1% oli mobiililaitteita, 28.9% tietokoneita ja 3.1% tabletteja. Tietokonekäyttäjät viettivät silti vuonna 2020 sivuilla vierailuista ajasta 46.4%, vaikka käyttäjiä on vähemmän. (Enge 2021.)

Mobiililaitteiden saavutettavuus suosituimmilla mobiilialustoilla eli iOS:llä ja Androidilla on todella kattava. Modernit mobiililaitteiden selaimet tukevat nykyään myös hyvin WAI-ARIAa. Verkkosivujen saavutettavuus mobiililaitteikäyttäjille voidaan varmistaa noudattamalla hyviä käytäntöjä kehityksessä ja suunnittelussa. (Mozilla 2021e.)

Muutamia poikkeuksia, jotka tulisi ottaa enemmän huomioon mobiililaitteiden käytöstä ovat ohjausmekanismit, käyttäjän syöte ja responsiivisuus. Ohjausmekanismeissa kannattaa varmistaa, että käyttöliittymän elementit toimivat samalla tavalla kosketusnäytöillä kuin näppäimistöillä ja hiirelläkin. (Mozilla 2021e.)

Verkkosivujen responsiivisuus on eri päätelaitteiden kannalta olennaista, jotta käyttö on mahdollisimman sujuvaa ja vaivatonta. Responsiivinen verkkosivuskaalautuu päätelaitteen näytön tai ikkunan koon ja resoluution mukaan, ja näyttää sivuston sisällön laitteelle sopivalla tavalla. (Mozilla 2021e.)

Olennaista responsiivisessa suunnittelussa on sivustorakenne ja sisällön esitystapa. Tietokoneella on yleensä isommat näytöt ja resoluutiot, joten monta elementtiä voivat olla vierekkäin rivissä, mutta pienellä puhelimen näytöllä ne eivät välttämättä mahdu vierekkäin. Hyvä responsiivinen sivu laittaisi esimerkiksi tietyssä näyttökoossa rivissä olevat elementit allekkain, jotta sisällön esitystapa on loogisempaa. Responsiivisuuden yleisimmät toteutustavat ovat media queryt, viewport ja flexbox. (Mozilla 2021e.)

Media queryt antavat kehittäjän määrittellä erilaisia CSS-sääntöjä kehittäjän valitsemien ehtojen perusteella. Media queryja voidaan käyttää esimerkiksi sivustorakenteen muuttamiseen mobiililaitteilla määrittämällä muutoksien tulevan voimaan, kun ruudunleveys on 800 pikseliä vähemmän (kuva 18). (Mozilla 2021I.)

```

.flex-container {
  display: flex;
  flex-direction: row;
}

@media (max-width: 800px) {
  .flex-container {
    flex-direction: column;
  }
}

```

Kuva 18. Flexbox ja media query.

Monilla verkkosivuilla näytön zoomaaminen mobiililaitteilla on otettu pois käytöstä. Näin ei kannattaisi kuitenkaan tehdä, sillä se vie pois sivuston saavutettavuutta. Monet ihmiset tarvitsevat zoomausta, jotta he voivat nähdä sisällön paremmin. Joissain tilanteissa zoomaus voi kuitenkin rikkoa käyttöliittymän.

Tällaisissa tapauksissa on mahdollista ottaa zoomaus pois käytöstä, mutta tulee käyttäjälle tarjota esimerkiksi mahdollisuus suurentaa fonttia tavalla, joka ei riko käyttöliittymää. (Mozilla 2021e.)

Pienemmille näytöille kehittäessä kannattaa huomioon ottaa myös menut. Yleinen tapa on laittaa sivun yläoikealle menua indikoiva kuvake, jota painamalla koko menu aukeaa. Menua kehittäessä täytyy muistaa tarkistaa, että menu aukeaa erilaisilla päätelaitteilla oikein. On myös muistettava piilottaa menussa oleva sisältö, kun menu ei ole auki niin, että ruudunlukijakäyttäjät otetaan huomioon. (Mozilla 2021e.)

Käyttäjän syöte kannattaa huomioida kehittäessä responsiivisia sivuja. Eri päätelaitteilla kirjoittaminen on vaikeampaa kuin perinteisellä näppäimistöllä. Tämän takia tekstisyöte kannattaa minimoida esimerkiksi pudotusvalikolla, josta käyttäjä voi valita yleisistä vaihtoehdoista yhden tai vaihtoehtoisesti valita arvon ”Muu”, ja kirjoittaa vastauksensa. (Mozilla 2021e.)

Huomioon tulee ottaa myös HTML5:n erilaiset syötetyypit. Jos käyttäjän halutaan antavan esimerkiksi puhelinnumeron, voidaan päätelaitteelle tarjota tähän sopiva virtuaalinen näppäimistö, joka näyttää vain numerot ja merkit, jotka ovat oleellisia puhelinnumerolle. Myös päivämäärän ja ajanvalinnalle voidaan näyttää esimerkiksi kalenteri, josta käyttäjä voi valita päivämäärän ja tämän lisäksi sopiva aikavalinta. (Mozilla 2021e.)

5 Virtual Showroom -verkkosivuston saavutettavuuden selvitys

5.1 Virtual Showroom -verkkosivuston taustoitus

Karelia-ammattikorkeakoulu toteuttaa yhdessä Pohjois-Karjalan maakuntaliiton, Luonnonvarakeskus LUKE:n, Itä-Suomen yliopiston, Pohjois-Karjalan koulutuskuntayhtymä Riverian ja Business Joensuu Oy:n kanssa 2-vuotisen 'Pohjois-Karjalan ja Japanin Naganon metsäbiotalousyhteistyön kehittäminen 2020–2022' -hankkeen ajalla 1.10.2020-30.9.2022. (Karelia-ammattikorkeakoulu 2021.)

Hankkeen päätoteuttajan, Karelia-ammattikorkeakoulun yhtenä toimintatavoitteena hankkeessa on virtuaalisen showroom-ympäristön rakentaminen Pohjois-Karjalan metsäbiotalousalan ja sen toimijoiden esittäytymistä ja yhteydenpitoa varten kohdealue Japanin kumppanitahojen kanssa. Virtual Showroomin ympäristön suunnittelu ja rakentaminen toteutettiin Karelia-ammattikorkeakoulun tietojenkäsittelyn koulutusohjelman projektityönä. (Karelia-ammattikorkeakoulu 2021.)

Tämän opinnäytetyön toimeksiantona oli tehdä saavutettavuusselvitys Virtual Showroom -ympäristön tämänhetkisestä tilanteesta, jota opiskelijat ovat kehittäneet hankkeen aikana aktiivisesti. Opinnäytetyön ajaksi saatiin pääsy sivuston lähdekoodiin ja julkinen URL-osoite, josta sivuston sen hetkistä saavutettavuuden tilannetta päästiin selvittämään.

5.2 Testauksen työkalut

Verkkopalveluiden saavutettavuuden testaamiseen on luotu monia työkaluja. Yleisiä testaustapoja ovat esimerkiksi sivustot kuten WAVE eli Web Accessibility Evaluation Tool, jonne syötetään testattavan verkkosivun URL-osoite ja esimerkiksi Dequen luoma selaimen lisäosa "axe". Axe toimii selaimen devtools -osiossa, jotka ovat kehittäjille tarkoitettuja työkaluja (Deque 2021). Devtoolsit voidaan avata esimerkiksi Chromessa painamalla F12-näppäintä. Myös WAVE:n voi ladata selaimen lisäosaksi (WAVE 2021).

WAVE:n sivuille Virtual Showroomin URL-osoitteen syöttö antoi paljon vähemmän tietoa Virtual Showroomin saavutettavuuden tilanteesta, kuin WAVE:n selainlisäosa. Virtual Showroomin saavutettavuutta testataan tässä opinnäytetyössä selaimen ladattavalla lisäosalla, sillä selainlisäosat osaavat antaa tarkempia tuloksia kuin pelkän URL-osoitteen syöttäminen Virtual Showroom -sivuston luonteen vuoksi. Sivusto on yksittäinen HTML-dokumentti, jossa on useita eri näkymiä, mutta ei alisivuja eri URL-osoitteella varustettuna. Tämän vuoksi pelkän URL-osoitteen syöttäminen ei anna kattavaa raporttia saavutettavuuden tilanteesta, sillä se ei tarkastele sivuston kaikkia eri näkymiä. URL-osoitteen syöttöön staattiset sivut toimivat paremmin, sillä niissä sisältö ei juurikaan muutu käytön aikana.

Tässä opinnäytetyössä keskitytään saavutettavuuden testaukseen saavutettavuuden evaluointi selainlisäosalla, NVDA-ruudunlukijalla ja manuaalisesti rinnastamalla sivustoa WCAG 2.1:n lainvelvoittamiin tason A- ja AA-kriteereihin. Tässä opinnäytetyössä ei käydä läpi tason AAA-kriteereitä, sillä ne eivät ole Suomessa lain velvoittamia. Manuaalisessa tarkastuksessa käytetään apuna WebAIM:in kontrastisuhteen tarkasteluun luotua työkalua. WebAIM:in kontrastityökalulle annetaan taustan ja esimerkiksi tekstin väri, joiden kontrastisuhteen työkalu kertoo. Sen lisäksi työkalu ilmoittaa onko kontrastisuhte WCAG 2.1:n mukaan hyväksyttävä (kuva 19).

Contrast Checker

[Home](#) > [Resources](#) > Contrast Checker

Foreground Color: #606060
Background Color: #CECECE

Lightness

Contrast Ratio: **3.99:1**

[permalink](#)

Normal Text

WCAG AA: **Fail**
WCAG AAA: **Fail**

Large Text

WCAG AA: **Pass**
WCAG AAA: **Fail**

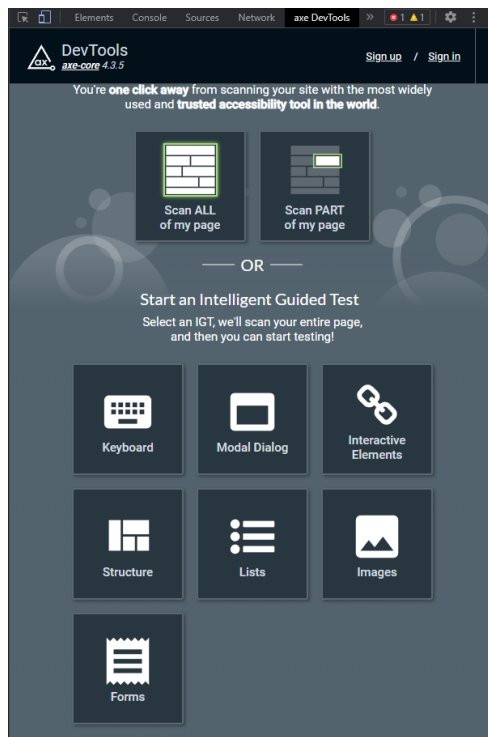
Graphical Objects and User Interface Components

WCAG AA: **Pass**

Kuva 19. WebAIM:in kontrastityökalu. (WebAIM 2021.)

5.2.2 Saavutettavuustyökalujen vertailu toisiinsa

Dequen luoma Axe -selainlisäosa on yksi suosituista testaustyökaluista, jota tässä opinnäytetyössä tullaan hyödyntämään. Axe:n ladattua sen löytää selaimen kehittäjätyökaluista, ja aloitusnäkyä antaa kehittäjälle erilaisia vaihtoehtoja. Ilmaisversiossa kehittyneemmät vaihtoehdot eivät ole käytössä, minkä vuoksi opinnäytetyössä hyödynnettiin ”Scan all of my page” -vaihtoehtoa (kuva 20).

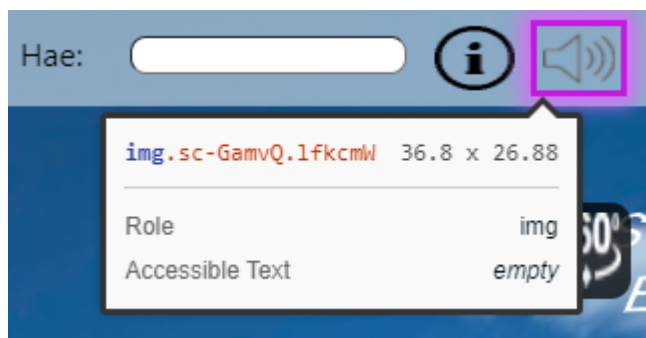


Kuva 20. Axe:n aloitusnäkyä Chromen kehittäjätyökaluissa.

Axen tuottama raportti ilmoittaa kategorioittain löytämänsä virheet ja listaa ne allekkain (kuva 21). Axe ilmoittaa myös WCAG 2.1:n onnistumiskriteerin, johon virhe liittyy kuten kuvassa 21 kohdassa ”Issue Tags” näkyy. Jokainen sivuston virheellinen elementti voidaan käydä läpi yksitellen, sekä korostaa elementti sivulla visuaalisesti halutessaan painamalla highlight-painiketta. (kuva 22).

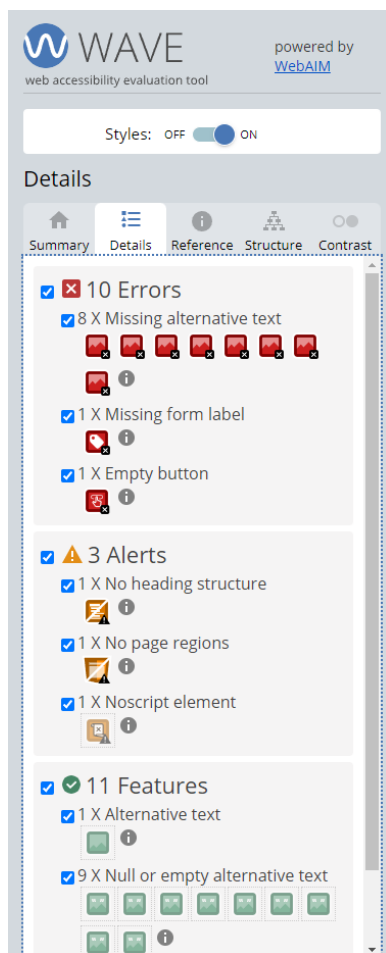


Kuva 21. Axen tuottama raportti sivustonäkymän skannauksen jälkeen.



Kuva 22. Axen visuaalisesti korostama elementti, josta puuttuu alt-teksti.

Myös WAVE tuottaa samantapaisen raportin kuin axe sivustonäkymän saavutettavuuden tilanteesta (kuva 23.). WAVE korostaa automaattisesti sivuston kaikki elementit, joista se löytää ongelmia (kuva 24). WAVE ei tässä testauksessa antanut yhtä kattavaa raporttia kuin axe. WAVE ei esimerkiksi ilmoittanut ollenkaan kontrastivirheistä, joita sivulta löytyy. Axe puolestaan huomasi kontrastivirheet, sekä ilmoitti muutamasta ylimääräisestäkin, jotka axen mukaan saattavat olla ongelmia. Testaustyökaluilla ilmenee olevan vaikeuksia kontrastisuhteiden kanssa, kun taustalla on esimerkiksi kuva. Tässä tapauksessa axen kontrastivirheistä vähintään yksi piti paikkaansa, joka on parempi kuin WAVEN tuottama raportti, joka ei ilmoittanut yhdestäkään.



Kuva 23. WAVE:n tuottama raportti sivustonäkymän skannauksen jälkeen.



Kuva 24. WAVE:n visuaalisesti korostamat elementit, joista löytyy saavutettavuuden kannalta virheitä.

WAVE toimii lähes samalla tavalla kuin axe. Yhtenä erona on se, että aloitusnäkyä ei ole, vaan WAVE skannaa sivuston sen hetkisen näkymän, kun käyttäjä avaa WAVE-selainlisäosan selaimessaan. Tämä on epäkäytännöllisempi testauksessa axeen verrattuna, sillä sivuston eri osan skannaukseen joutuu WAVE:n sammuttamaan kokonaan, ja sen jälkeen käynnistämään uudelleen. Axe:lla puolestaan voidaan painaa vain ”re-run automatic scan”, joka suorittaa sen hetkiselle sivustonäkymälle uuden skannauksen.

Kokonaisuudessaan axe antoi sivustonäkymästä 31 mahdollista virhettä, kun WAVE ilmoitti löytäneensä vain 18. Osa axen tuottamista virheistä ovat merkattu tarvitsevansa ihmisen tarkastusta eli mahdollisesti virheellisiksi elementeiksi, ja osa virheistä oli merkattu parhaiksi käytännöiksi, jotka eivät ole lain velvoittamia. Tämä on kuitenkin sivustoa kehittäessä huomioitava asia, sillä sivustosta saadaan kaiken kaikkiaan saavutettavuudeltaan parempi käyttämällä axen tuottamaa raporttia kuin WAVEN, ainakin Virtual Showroomin osalta tämän testauksen perusteella. Tämän vertailun perusteella tämän opinnäytetyön yhteydessä päädyttiin käyttämään axea saavutettavuus selvityksen yhteydessä WAVE:n sijasta.

5.3 Virtual Showroomin saavutettavuuden selvitys

5.3.1 Saavutettavuuden selvityksen toteutustapa ja työkalut

Saavutettavuuden selvitys opinnäytetyössä tehtiin vaiheittain, aloittaen selainliisäosa axella, jonka jälkeen sivu käytiin läpi NVDA-ruudunlukijalla. Lopuksi sivustolle tehtiin vielä manuaalinen tarkistus. Virtual Showroomissa on useita eri näkymiä, joihin käyttäjä voi navigoida. Opinnäytetyössä käytiin läpi kaikki sivuston näkymät ja ilmoitettiin niistä löytyneet ongelmat. Näkymät käytiin sivuston käyttöön nähden loogisessa järjestyksessä läpi. Aloitusruudusta käytönopastukseen, jonka jälkeen siirryttiin sivuston päänäkymään, josta löytyy klikattavia palloja. Kukin palloista sisältää eri yrityksen tietoja, jotka voidaan avata palloa klikkaamalla. Klikkauksen jälkeen aukeaa 360-näkymään ikkuna, josta löytää klikatun yrityksen tiedot. Jokaisesta näkymästä tarkasteltiin mahdollisimman kattavasti työkalujen avulla saavutettavuuden tilanne, ja raportoitiin tilanne opinnäytetyöhön tekstimuodossa.

Saavutettavuuden selvityksen jälkeen selvityksessä löydettyjä tuloksia rinnastettiin WCAG 2.1:n onnistumiskriteereihin, ja käytiin selvityksessä löydetty ongelmakohdat kriteereittäin läpi. Tässä opinnäytetyössä ei käydä läpi jo onnistumiskriteerit läpäiseviä sivuston kohtia tai onnistumiskriteereitä, jotka eivät ole sovellettavissa Virtual Showroom -sivustoon Saavutettavuus selvityksen tarkoi-

tuksena on löytää mahdolliset ongelmakohdat ja parannusehdotukset, jotta sivuston saavutettavuutta voidaan parantaa tulevaisuudessa, mikäli ongelmia löytyy. Esimerkiksi kaikkien lain velvoittamien 49 onnistumiskriteerin listaaminen ja niiden läpikäyminen Virtual Showroomin osalta ei ole olennaista, vaan ainoastaan ongelmakohtien korjaaminen riittää.

5.3.2 Aloitusnäky

Virtual Showroomin ensimmäinen näkymä sisältää ”Tervetuloa”-tekstin, joka kertoo käyttäjän tutustuvan Pohjois-Karjalaan ja sen metsäbiotalouteen. Aloitusnäkyä löytyy myös sivuston ja hankkeen mukana olleita toimijoita. Näkyästä voidaan siirtyä eteenpäin ”Aloita tästä” -painikkeella.

Ensimmäiseksi ongelmaksi axle ilmoittaa sivuston aloitusruudun kuvaelementeistä puuttuvan alt-tekstin. Aloitusruudussa kaikki sisältö lukuun ottamatta ”Aloita tästä” -painiketta on toteutettu -elementeillä, joissa ei ole alt-tekstiä (kuva 25). Testatessa aloitusnäkyä NVDA-ruudunlukijalla, ruudunlukija ei lue aloitusruudussa olevia tekstejä ääneen, koska alt-teksti puuttuu.



Kuva 25. Kuvaelementeillä toteutettu aloitusruutu.

Tämän lisäksi WCAG 2.1:n onnistumiskriteeri ”1.4.5 Tekstiä esittävät kuvat” sanoo, että ”Jos käytetty teknologia voi tuottaa visuaalisen esityksen, informaation välittämiseen käytetään enemmän tekstiä kuin tekstiä esittäviä kuvia” lukuun ottamatta tilanteita, jossa tekstiä esittävä kuva voidaan muuttaa käyttäjän vaatimusten mukaisesti tai tekstin esitystapa on olennainen välitettävän informaation kannalta (WCAG 2.1). Tästä syystä olisi parempi korvata kuvaelementit HTML-tekstillä, ja tyyllillä ne CSS-tyyleillä. Sivuston alareunassa oleviin logoihin tulisi lisätä alt-tekstit, jotta ruudunlukija voi ottaa ne huomioon. Yksi hyvä vaihtoehto olisi esimerkiksi korvata aloitusruudun ”Tervetuloa Pohjois-Karjalaan”-teksti <h1>-elementillä, jotta ruudunlukijalla navigointi sivustolla olisi helpompaa.

Manuaalisesti aloitusruudusta löydettiin, että kontrastisuhteet eivät ole tarpeeksi hyvät. Aloitusruudun teksti ”Olet tutustumassa maakunnan metsäbiotalouden Showroom -sivustolle” ja tekstin takana oleva sininen taivas laitettiin WebAIM:in kontrastinvertailutyökalulle, joka ilmoitti kontrastisuhteen olevan 1.53:1 (kuva 26). Isokokoisien tekstien kontrastisuhteen tulisi olla WCAG:n 1.4.3 Kontrasti - vaatimuskriteerin mukaan vähintään 3:1. Isokokoiseksi tekstiksi luokitellaan vähintään 18 pisteen tai lihavoituna 14 pisteen fonttikoko. Logoille ja brändeille ei ole kontrastivaatimusta. (WCAG 2.1.)

Contrast Ratio
1.53:1

[permalink](#)

Normal Text

WCAG AA: **Fail**
WCAG AAA: **Fail**

The five boxing wizards jump quickly.

Large Text

WCAG AA: **Fail**
WCAG AAA: **Fail**

The five boxing wizards jump quickly.

Graphical Objects and User Interface Components

WCAG AA: **Fail**

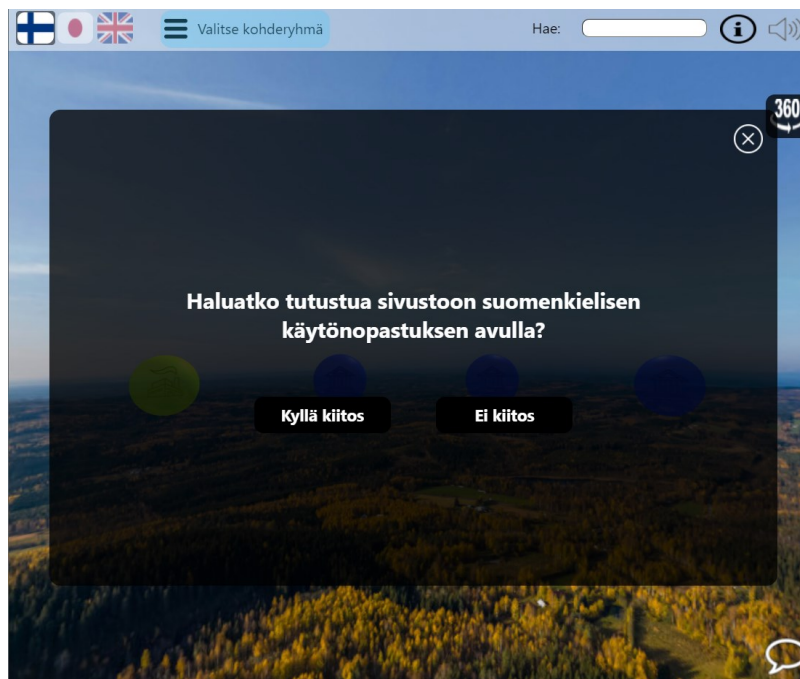
Text Input

Kuva 26. Virtual Showroomin aloitusruudun tekstin kontrastisuhte. (WebAIM 2021.)

5.3.3 Käytönopastus

Aloituskäytön jälkeen tulee käyttäjälle mahdollisuus tutustua sivustoon käytönopastuksen avulla. Käytönopastus on mahdollista ohittaa klikkaamalla ”Ei kiitos” -painiketta tai suorittaa klikkaamalla ”Kyllä kiitos” -painiketta. Käytönopastuksessa käyttäjälle kerrotaan mitä sivuston eri painikkeet tekevät ja kuinka sivustoa käytetään.

Käytönopastuksen manuaalisessa tarkistuksessa selvisi, että painikkeet, joilla käyttäjä etenee käytönopastuksessa ovat toteutettu <div>-elementeillä. Käytönopastuksen modaalin yläkulmassa oleva raksi on toteutettu -elementinä. (kuva 27). Nämä kannattaisi korvata <button>-elementeillä, sillä ne ovat semanttisia ja niissä on näppäimistötoiminnallisuus valmiina. Sivustolla käytettyihin <div>- ja -painikkeisiin ei ole rakennettu näppäimistötoiminnallisuutta.



Kuva 27. Virtual Showroomin käytönopastus.

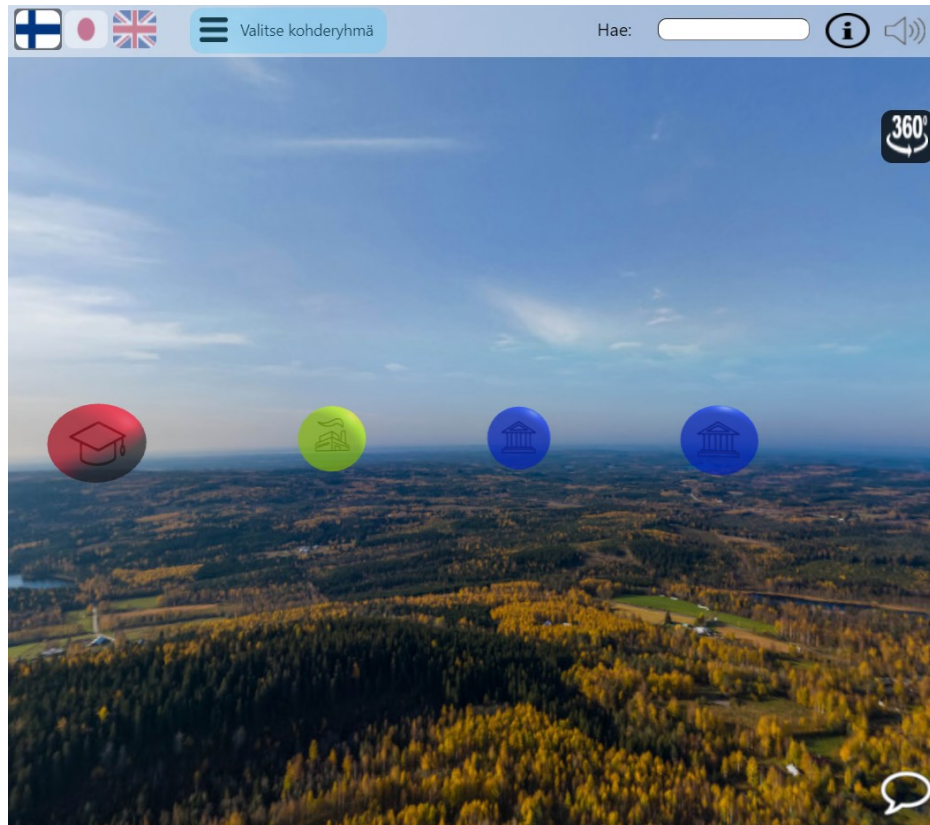
Manuaalisen tarkastuksen aikana kokeiltiin vaihtaa <div>- ja -painikkeet <button>-elementeiksi, jonka jälkeen käytiin käytönopastus ruudunlukijalla läpi. Ruudunlukija tulkitse vastaa <button>-elementeiksi korjauksen jälkeen, että kyseessä olevat elementit ovat painikkeita, ja oli tätä myöten käyttökokemus ruudunlukijalla miellyttävämpi. Myös näppäimistö navigointi alkoi toimimaan <button>-elementeillä.

5.3.4 Päänäkymä

Käytönopastuksen jälkeen aukeaa sivuston päänäkymä. Päänäkymän yläpalkista löytyy kielivalinta, kohderyhmän valinta, hakupalkki, infopainike, josta aukeaa käytönopastus ja taustamusiikin hiljentäminen. Päänäkymä on 360-asteinen kuva, jota käyttäjä voi hiiren vasenta painiketta pohjassa pitämällä ja raahaamalla hiirtä käänneellä haluamaansa suuntaan. 360-asteista kuvaa voidaan kääntää myös sormella mobiililaitteilla. 360-asteisen kuvan päällä on eri värisiä palloja, joita klikkaamalla voidaan nähdä eri yritysten ja organisaatioiden tietoja. Päänäkymässä on myös chat-painike, jota klikkaamalla voidaan avata chat-toiminto.

Axe ilmoittaa, että sivuston kielen tulisi olla ohjelmallisesti selvitettävissä. Sivustolta löytyvän kielivalinnan lisäksi tulisi HTML-dokumentista itsestään muuttaa lang-attribuuttia kielivalinnan mukaan. Tällä hetkellä oletuskielenä on suomi, mutta arvona HTML-dokumentissa on "en". WCAG 2.1:n mukaan jokaisen verkkosivun oletusarvoisen luonnollisen kielen tulisi olla ohjelmallisesti selvitettävissä. (WCAG 2.1 2018.) Tämä pätee koko sivustoon, ei pelkästään päänäkymään.

Axella löydettiin, että sivuston yläoikealla sijaitsevalla hakupalkilla ei ole ohjelmallisesti määritettävissä olevaa kentän otsikkoa, eli <label>-elementtiä (kuva 28). Visuaalisesti "Hae"-teksti voidaan mieltää kentän otsikoksi, mutta se on toteutettu <div>-elementillä, joka ei ole hyvä käytäntö.



Kuva 28. Virtual Showroomin päänäkymä.

WCAG 2.1:n onnistumiskriteerin ”4.1.2 Nimi, rooli, arvo” mukaan:

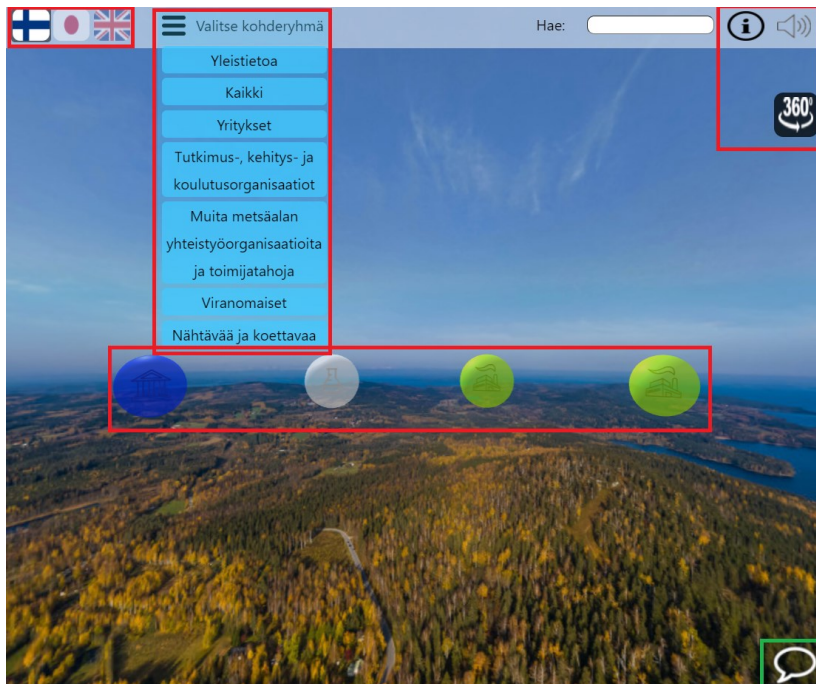
”Kaikkien käyttöliittymäkomponenttien (mm. lomake-elementit, linkit ja skriptien luomat komponentit) nimi ja rooli voidaan selvittää ohjelmallisesti. Tilat, ominaisuudet ja arvot, jotka käyttäjä voi asettaa, voidaan myös asettaa ohjelmallisesti ja tieto näiden muutoksista on käyttäjäagenttien, mukaan lukien avustavien teknologioiden, saatavissa” (WCAG 2.1 2018).

Tässä tapauksessa korjaus voidaan tehdä esimerkiksi <label>-elementin oikeaoppisella käytöllä <div>-elementin sijasta.

WCAG 2.1:n onnistumiskriteerin 2.1.1 Näppäimistö mukaan ”Kaikki sisällön toiminnallisuus on hallittavissa näppäimistörajapinnan välityksellä ilman vaatimusta yksittäisten näppäinpainallusten erityisestä ajoittamisesta, paitsi kun taustalla oleva toiminnallisuus vaatii syötettä, joka riippuu käyttäjän liikkeiden reitistä eikä vain päätepisteistä”. Manuaalisessa tarkistuksessa testattiin sivuston näppäimistösaavutettavuutta ja todettiin, että tällä hetkellä Virtual Showroomissa voidaan näppäimistön avulla navigoida vain sivuston yläoikealla sijaitsevaan ”Hae”-syötekenttään, ja oikealla alhaalla sijaitsevaan chatti-ikoniin. Chatti-ikoniin navigoidessa voidaan painaa Enteriä, joka avaa chatin. Ongelmana chatti-ikoniin navigoidessa on se, että onnistumiskriteeri ”2.4.7 Näkyvä kohdis-

tus” epäonnistuu. Onnistumiskriteerin mukaan ”Kaikilla näppäimistöltä käytettävillä käyttöliittymillä on käyttötila, jossa näppäimistön kohdistuksen ilmaisin on näkyvässä”. Tämä tarkoittaa sitä, että ikonin tulisi visuaalisesti muuttua indikoidakseen, että siihen on navigoitu. Yleinen indikaattori tästä on esimerkiksi aiemmin opinnäytetyössä mainittu elementin ympärillä sijaitseva kehikko. (WCAG 2.1 2018.)

Manuaalisessa tarkistuksessa todettiin, että suurin osa sivuston painikkeista on tehty jollain muulla, kuin <button>-elementillä. Kuvassa 29 olevat painikkeet, jotka eivät ole <button>-elementtejä on ympyröity punaisella. Sivustolla sijaitsevat kielenvalinta-painikkeet ovat -elementtejä, ”Valitse kohderyhmä” ja sen sisäiset painikkeet ovat toteutettu <div>-elementeillä, info-, ääni- ja 360-painikkeet ovat -elementtejä sekä sivustolla olevat pallot, joita klikatessa aukeaa eri yritysnäkymä ovat <div>-elementtejä (kuva 29). Chat-painike on toteutettu <button>-elementtinä, kuten kaikki muutkin painikkeet kannattaisi toteuttaa. <button>-elementeillä on näppäimistötoiminnallisuus valmiina entuudestaan. Tämän vuoksi ainut elementti, johon näppäimistöllä voidaan navigoida lomakekentän lisäksi, on chat-painike.



Kuva 29. Virtual Showroomin painikkeet, joista punaisella värillä ovat korostettuna muut, kuin <button>-elementit.

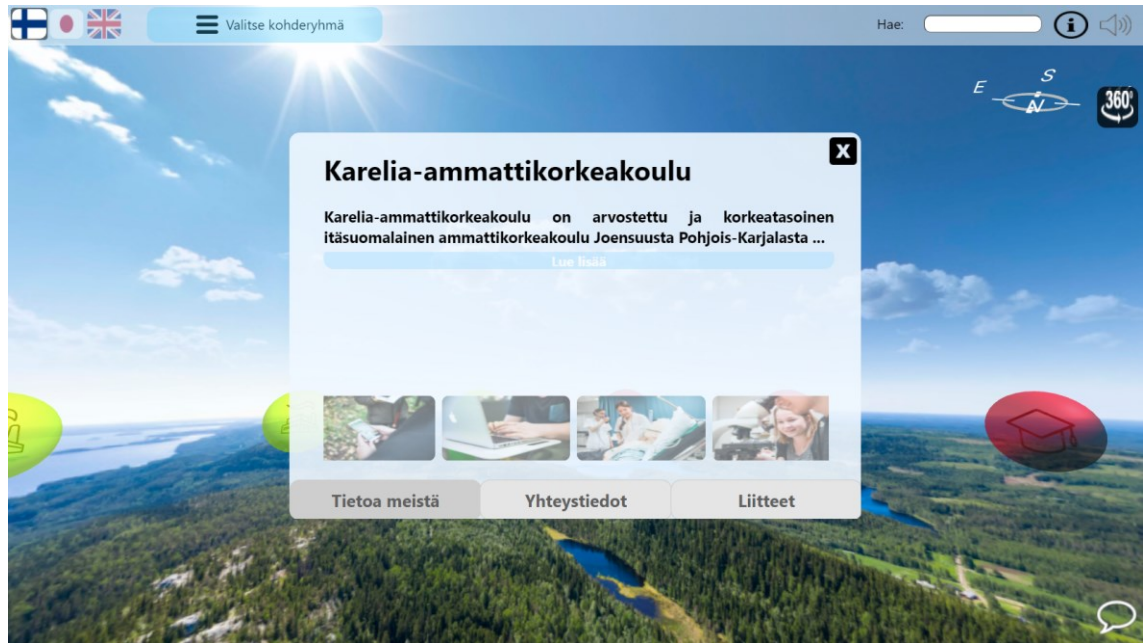
Kuten tässä opinnäytetyössä todettiin luvussa ”4.1 HTML”, parempi vaihtoehto olisi käyttää semanttisia elementtejä niille tarkoitetuissa tilanteissa, eli tässä tapauksessa HTML:n <button>-elementtiä jokaisen sivuston painikkeen kohdalla <button>-elementissä on valmiiksi näppäimistösaavutettavuus, eikä sitä tarvitse lähteä luomaan uudestaan, kuten <div>- ja -elementteihin tulisi. Myös ruudunlukija osaa ilmoittaa kyseessä olevan painikkeen, kun se navigoi <button>-elementin kohdalle.

5.3.5 Yritysnäkymä

Päänäkymässä olevia palloja klikkaamalla saadaan erilaisten yritysten tietoja tarkasteltua yritysnäkymässä. Yritysnäkymässä kerrotaan yritysten ja organisaatioiden nimi, kuvaus, yhteystiedot, mahdolliset liitteet sekä kuvat. Yritysnäkymässä voi navigoida yritysnäkymäikkunassa sijaitsevien välilehtien avulla.

Yritysnäkymästä axe ilmoittaa kontrastivirheistä ja alt-tekstin puutteista. Alt-tekstin puutteesta axe kertoo myös yritysnäkymän raksin kohdalla (kuva 30). Automaattiset saavutettavuustyökalut luottavat siis kehittäjiin semanttisuuden kanssa, eivätkä esimerkiksi tässä tapauksessa huomaa, että kuvaa käytetään painikkeena. Tässäkin tilanteessa tulisi kuvaelementti korvata <button>-elementillä.

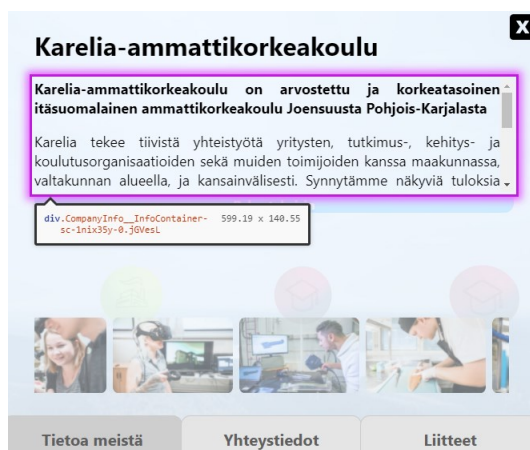
Kontrastivirheenä axe ilmoitti jokaisen tekstin, joka yritysnäkymästä löytyy. Tämä ei kuitenkaan pidä paikkaansa, vaan työkaluilla on ongelmia kontrastien erottamisessa, kun taustana on esimerkiksi kuva ja CSS:n opacity eli elementin läpinäkyvyys -arvoa on säädetty. Kontrastivirhe pitää paikkaansa yrityskuvauksen alla olevassa vaalean sinisessä ”Lue lisää” -painikkeessa, jossa on valkoinen teksti (kuva 30). Tämä kontrastisuhde ei ole hyväksyttävä WCAG 2.1:n kontrastisuhteiden mukaisesti. Kontrastisuhde paranee, kun hiiri viedään elementin päälle, ja silloinkin kontrastisuhde on vain 1.43:1.



Kuva 30. Virtual Showroomin yritysnäkymä.

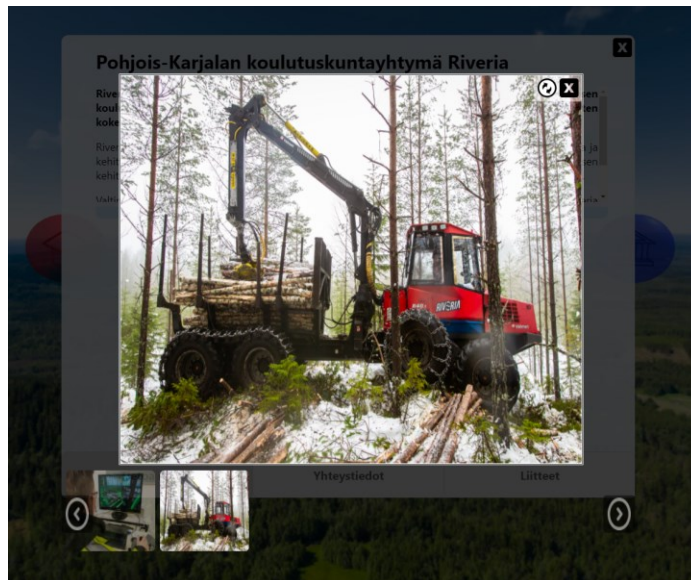
”Lue lisää” -painikkeella aukeaa vieritettävä kuvaus yrityksestä, jonka axe korostaa, sillä se ei ole vieritettävissä näppäimistöllä (kuva 31). Tämä oli ensimmäinen kerta, kun axe ilmoitti näppäimistösaavutettavuudesta Virtual Showroomissa. Myös ”Yhteystiedot”-välilehdeltä löytyy vieritettävä osio, josta axe myös ilmoittaa, että näppäimistösaavutettavuus ei toimi.

Yritysnäkymän alhaalla olevat ”Tietoa meistä”, ”Yhteystiedot” ja ”Liitteet” painikkeet ovat <div>-elementtejä, joiden kuuluisi olla <button>-elementtejä (kuva 31). Axen mukaan kontrastisuhteet ovat niissä liian vähäiset. Tekstikoko on kuitenkin kyseisen elementin CSS-säännöissä määrätty olevan minimissään 20px, ja WebAIM:in kontrastityökalun mukaan kontrastisuhde 3.99:1 on isolle tekstille riittävä.



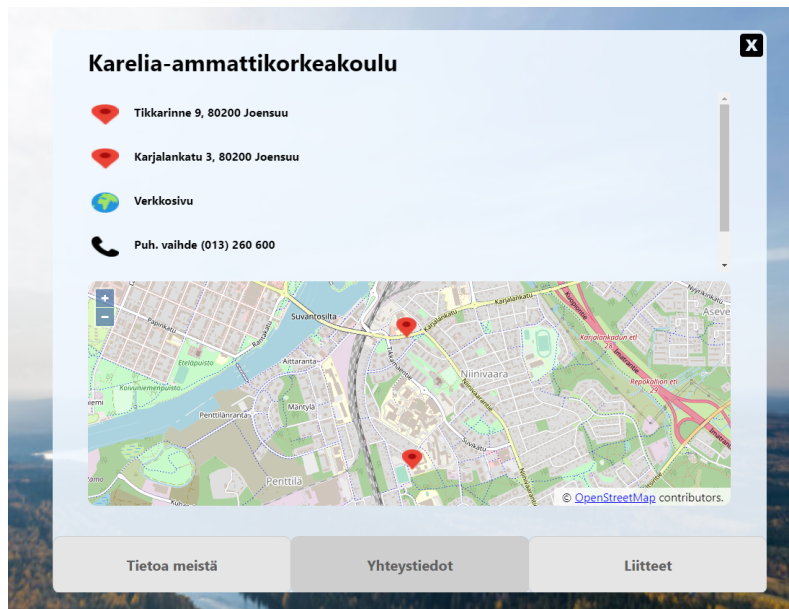
Kuva 31. Vieritettävä kuvaus yrityksestä.

Yritysnäkymän kuvat voidaan klickata auki suuremmaksi ja selata niitä kuvagalleriatyylisesti klikkaamalla alhaalla olevia pieniä kuvia (kuva 32). Axella löydettiin muutama virhe, jotka jatkavat samaa kaavaa kuin aiemmat Virtual Showroomin virheet. Kaikki gallerian painikkeet ovat tehty -elementeillä, nämä tulisi korvata <button>-elementeillä. Kuvilla ei ole alt-tekstiä, jonka vuoksi ruudunlukijasaavutettavuus ei ole riittävä.



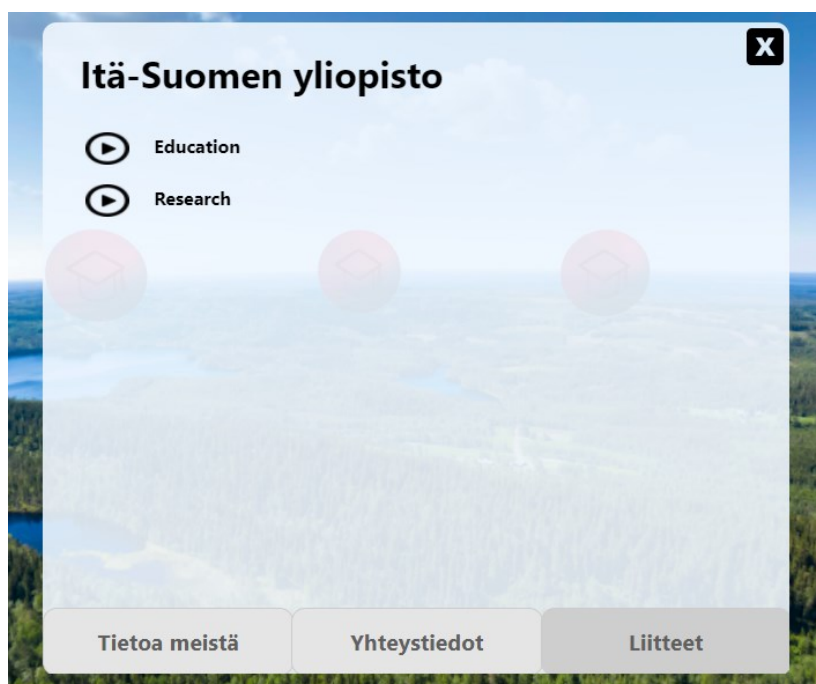
Kuva 32. Yritysnäkymän kuvien katselu.

Yhteystiedot-välilehdestä axe löytää ongelmaksi listattujen yhteystietojen ikonit, jotka edeltävät osoitteita, ”Verkkosivu”-painiketta ja puhelinnumeroa (kuva 33). Kuvaelementeillä ei ole alt-tekstiä. Ruudunlukijatoiminnallisuutta on vaikea testata, sillä yritys näkymä ei ole näppäimistöllä navigoitavissa. Tällä hetkellä ruudunlukija ei voi lukea kuvakkeiden sisältöä ääneen, ja olisi järkevää ilmoittaa ruudunlukijakäyttäjälle ennen jokaisen yhteystiedon lukemista, millainen yhteystieto on kyseessä. Yksi vaihtoehto on esimerkiksi sisällyttää osoitetiedot ja puhelinnumero HTML:n <address>-elementin sisälle, ja kirjoittaa osoitetiedot <a>-elementteihin käyttäen href-attribuuttia. Myös verkkosivulinkin kannattaisi olla <a>-linkkielementillä toteutettu, jotta ruudunlukija osaa ilmoittaa kyseessä olevan elementin olevan linkki.



Kuva 33. Virtual Showroomin yritysnäkymän ”Yhteystiedot”-välilehti.

Liitteet-välilehdeltä tarkastelun tapauksessa löytyy kaksi linkkiä, joista axe ei ilmoita mitään (kuva 34). Axe tai ruudunlukija eivät kumpikaan tiedä kyseessä olevan linkkejä, kun niitä ei ole toteutettu `<a>`-elementeillä. Nämä ovat toteutettu ``-elementeillä, mutta saavutettavuuden takaamiseksi ne kannattaisi olla `<a>`-elementtejä, jotta ruudunlukija voi kertoa käyttäjälleen niiden olevan linkkejä.



Kuva 34. Yritysnäkymän liitteet -välilehti.

5.3.6 Virtual Showroomin responsiivisuuden testaus

WCAG 2.1:n onnistumiskriteerin 1.4.10 Responsiivisuus mukaan ”Sisältö voidaan esittää ilman sisällön tai toiminnallisuuden menettämistä ja ilman kahdensuuntaista vierittämistä, kun pystysuuntaan vieritettävän sisällön leveys on 320 CSS-pikseliä ja vaakasuuntaan vieritettävän sisällön korkeus on 256 CSS-pikseliä”. Huomautuksena onnistumiskriteerissä kerrotaan, että ”320 CSS-pikseliä vastaa 1280 CSS-pikselin levyistä selainikkunaa, joka on zoomattu 400% kokoiseksi. Vaakasuuntaan vieritettävässä sisällössä (esim. pystysuuntainen kirjoitus), 256 CSS-pikseliä vastaa 1024 px korkeaa selainikkunaa, joka on zoomattu 400% kokoiseksi”. (WCAG 2.1 2018.)

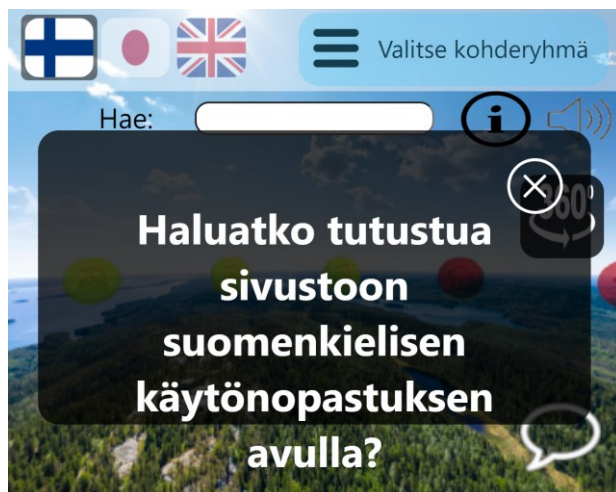
Tämän responsiivisuuden onnistumiskriteerin pohjalta toteutettiin sivuston responsiivisuuden testaus manuaalisesti ilman työkaluja 1280 CSS-pikselin levyisellä ja 1024 CSS-pikselin korkuisella selainikkunalla selaimen zoomauksen ollessa 400%. Esimerkkejä WCAG 2.1:stä tai tarkempaa selitystä sisällön tai toiminnallisuuden menettämisestä ei ole, joka jätti testauksen jollain osin testajan tulkinnan varaiseksi.

Aloituskäytännön logoista puuttuu Karelian logo kokonaan, ja ”Aloita tästä” -painike on logojen päällä (kuva 35). Käyttämällä opinnäytetyössä aiemmin mainittua CSS:n media query -sääntöä, voitaisiin tietyssä ikkunakoossa siirtää ”Aloita tästä” -painiketta ylemmäs ja pienentää logoja, jotta ne kaikki mahtuvat ruudulle samanaikaisesti näkyviin.



Kuva 35. Aloitusnäytön responsiivisuuden testaus.

Käytönopastuksesta löydettiin käytettävyyttä rikkovia ongelmia, sillä käytönopastusruudusta puuttuu painikkeet ”Kyllä kiitos” ja ”Ei kiitos”, joilla käytönopastuksessa edetään. Puuttuvan toiminnallisuuden lisäksi suuri osa sivuston painikkeista ovat käytönopastusruudun takana osittain piilossa (kuva 36).



Kuva 36. Käytönopastuksen responsiivisuuden testaus.

Päänäkymän pudotusvalikosta puuttuu osa vaihtoehdoista, ja chat-painike peittää osan yhden kohderyhmän nimestä (kuva 37). Sisällön menetyksellä rajoittaa tässä tapauksessa kohderyhmien valintaa huomattavasti.



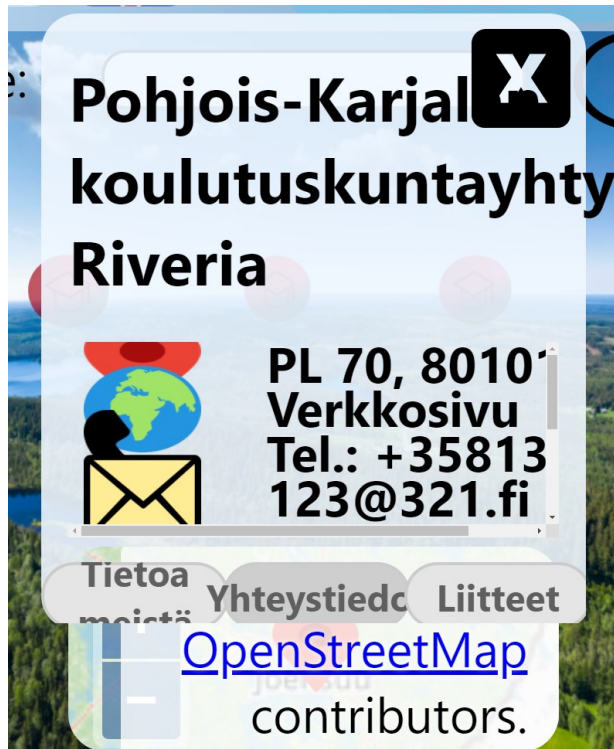
Kuva 37. Päänäkymän responsiivisuuden testaus.

Yritysnäkymä ei ole tällä ikkunakoolla käytettävissä lainkaan. Yläoikealla sijaitseva raksi peittää yrityksen nimen kirjaimista osan, otsikkoteksti valuu yritysnäkymän reunojen yli, kuvat ovat "Lue lisää" -painikkeen päällä, kuvien päällä on osa yrityskuvauksen tekstistä, ja painikkeiden tekstit eivät näy kokonaan (kuva 38).



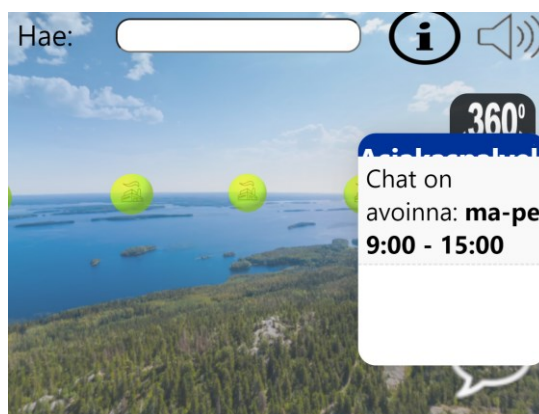
Kuva 38. Yritysnäkymän responsiivisuuden testaus.

"Yhteystiedot"-välilehdellä olevat yhteystiedot ovat vaikeasti erotettavissa toisistaan, ja niitä on vaikea klikata (kuva 39). Yritysnäkymän kaikki välilehdet sisältävät otsikon ja painikkeiden kannalta samat ongelmat. Myös liitteet-välilehdeltä löytyi samaiset ongelmat.



Kuva 39. Yritysnäkymän ”Yhteystiedot” -välilehden responsiivisuuden testaus.

Myös chat-ikkunasta katoaa kokonaisuudessaan ”Asiakaspalvelu”-teksti, joka normaalisti näkyy otsikkotekstinä sinisellä taustalla chat-ikkunan yläosassa, tekstikenttää johon chatin viestit kirjoitetaan ja ”Lähetä”-painike, jolla viestit lähetetään (kuva 40).



Kuva 40. Chat-ikkunan responsiivisuuden testaus.

Responsiivisuuden testauksessa kokeiltiin myös Chrome-selaimen kehittäjän työkaluista löytyvien erilaisten mobiililaitteiden näkymillä responsiivisuutta mm. tableteilla ja joillain puhelimilla, joissa ikkunankoko ei ole yhtä pieni kuin WCAG 2.1 vaatii. Sivusto toimi hieman suuremman näyttökoon omaavilla mobiililaitteilla selkeämmin ja paremmin, eikä toiminnallisuutta menetetty yhtä paljoa. Sivustoa

ei kuitenkaan voida sanoa vielä tarpeeksi responsiiviseksi, sillä WCAG 2.1:n ohjeistamassa ikkunakoossa sivusto menettää suuren osan sisältöä ja toiminnallisuutta.

5.3.7 WCAG 2.1:n onnistumiskriteerit Virtual Showroomissa

Testauksen jälkeen WCAG 2.1:n onnistumiskriteerit käytiin läpi rinnastaen niitä Virtual Showroom -sivuston testauksessa löydettyihin ongelmiin. Tässä opinnäytetyössä käytiin läpi vain A- ja AA-tason kriteerit, jotka ovat Suomessa lainvelvoittamia. Onnistumiskriteerit, joita tämän opinnäytetyön selvityksen mukaan Virtual Showroom ei tällä hetkellä läpäise, listattiin allekkain. Listauksesta rajattiin pois onnistumiskriteerit, jotka eivät ole sovellettavissa Virtual Showroom -sivustoon ja onnistumiskriteerit, joita vastaavia ongelmia ei löydetty.

Huomioitavaa onnistumiskriteereihin rinnastaessa on, että käytännön esimerkkejä onnistumiskriteereistä ei vielä saavutettavuusvaatimukset.fi sivustolla tai virallisessa verkkosisällön saavutettavuusohjeissa (WCAG 2.1) ole olemassa, ja tämä listaus perustuu täysin opinnäytetyön tekijän onnistumiskriteerien tulkintaan. Tulkinnanvaraiseksi jääneet kriteerit rajattiin tästä listauksesta pois, jotta listaus voitaisiin pitää mahdollisimman totuudenmukaisena. Onnistumiskriteerit, joita Virtual Showroom ei tämän saavutettavuus selvityksen mukaan noudata ovat:

- "1.1.1 Ei-tekstuaalinen sisältö. Kaikki käyttäjälle esitettävä ei-tekstuaalinen sisältö on varustettu saman tarpeen täyttävällä tekstivastineella."
- "1.3.1 Informaatio ja suhteet. Esitystavassa välittyvät informaatio, rakenne ja suhteet voidaan selvittää ohjelmallisesti tai ne ovat saatavilla tekstinä"
- "1.4.3 Kontrasti (minimi). Tekstin ja tekstiä esittävien kuvien visuaalisen esitystavan kontrastisuhde on vähintään 4,5:1."
- "1.4.5 Tekstiä esittävät kuvat. Jos käytetty teknologia voi tuottaa visuaalisen esityksen informaation välittämiseen käytetään enemmän tekstiä kuin tekstiä esittäviä kuvia."
- "1.4.10 Responsiivisuus. Sisältö voidaan esittää ilman sisällön tai toiminnallisuuden menettämistä ja ilman kahdensuuntaista vierittämistä."
- "2.1.1 Näppäimistö. Kaikki sisällön toiminnallisuus on hallittavissa näppäimistörajapinnan välityksellä."

- ”2.4.7 Näkyvä kohdistus. Kaikilla näppäimistöä käytettävillä käyttöliittymillä on käyttötila, jossa näppäimistön kohdistuksen ilmaisun on näkyvissä”
- ”3.1.1 Sivun kieli. Jokaisen verkkosivun oletusarvoinen luonnollinen kieli voidaan selvittää ohjelmallisesti”.
- ”3.1.2 Osien kieli. Sisällön jokaisen tekstikatkelman tai ilmaisun luonnollinen kieli voidaan selvittää ohjelmallisesti”.
- ”3.2.2 Nimilaput tai ohjeet. Kun sisältö edellyttää käyttäjän syötettä, tarjolla on nimilappuja tai ohjeita”.
- ”4.1.2 Nimi, rooli, arvo. Kaikkien käyttöliittymäkomponenttien (mm. lomake-elementit, linkit ja skriptien luomat komponentit) nimi ja rooli voidaan selvittää ohjelmallisesti”. (WCAG 2.1 2018).

5.3.8 Saavutettavuuden yhteenveto

Sivustolta löydettiin paljon samaa kaavaa noudattavia ongelmia. Suurin osa virheistä voidaan korjata käyttämällä HTML:n semanttisia elementtejä, eli tässä tapauksessa esimerkiksi painikkeiden muuttaminen <button>-elementeiksi olisi yksi suurimmista muutoksista.

Kaikkiin sivustolla esiintyviin kuviin kannattaa lisätä alt-tekstit, lukuun ottamatta ikoneina käytettäviä kuvakkeita, jos varmistetaan ruudunlukijan välittävän yhtä kattavan tiedon käyttäjälleen kuin kuvakkeet tekevät visuaalisesti. Tämä voidaan tehdä esimerkiksi aiemmin mainitulla <address>-elementin oikeaoppisella käytöllä. Sivuston kontrastisuhteet kannattaa korjata, sekä noudattaa yhdenmukaisuutta selkeyden ja käytettävyyden vuoksi. Näppäimistö navigointi sivulla parantuu semanttisten elementtien käytön myötä paljon jo itsestään, mutta siihen kannattaa kiinnittää erityisesti huomiota ja varmistaa, että se toimii onnistumiskriteerien vaatimalla tavalla. Responsiivisuuden parantaminen on hyvä toteuttaa CSS:n media queryjen avulla, ja käyttää ohella esimerkiksi aiemmin mainittua flexboxia elementtien asetteluun.

WCAG 2.1:n onnistumiskriteerit, joita Virtual Showroom -sivusto ei tämän saavutettavuusselvityksen perusteella vielä noudata korjaantuvat suurin osa itsestään, kun semanttiset elementit otetaan sivustolla käyttöön. Tämä saavutettavuusselvitys on korostanut erittäin paljon sitä, kuinka tärkeitä semanttiset

elementit verkkopalveluiden kehitystyössä ovat, ja kuinka monilta virheiltä olisi voitu välttyä, kun semanttisia elementtejä olisi käytetty projektin alusta asti.

6 Pohdinta

6.1 Saavutettavuudesta hyötyvät käyttäjät ja apulaitteet

Opinnäytetyön alussa saavutettavuus kokonaisuudessaan oli vielä täysin vieras aihe, ja suurena yllätyksenä oli, että koko maailman väestöstä yhteensä 15% kärsivät erilaisista vammoista. Saavutettavien verkkopalveluiden tärkeys korostui tätä myöten jo ensimmäistä lähdettä lukiessa. Apulaitteita osoittautui olevan olemassa eri tarpeisiin todella kattavasti, ja esimerkiksi ruudunlukijan käyttö oli helposti opittavissa.

Saavutettavuuden huomiointi verkkopalveluiden kehityksessä vaatii paljon tietoa ja ajatustyötä, jotta kaikki käyttäjäryhmät voidaan ottaa huomioon. Tämän vuoksi olisi tietojenkäsittelyalan opinnoissa hyvä opettaa enemmän saavutettavuudesta. Web-kehitykseen liittyvässä opetuksessa olisi tärkeää käyttää semanttisia HTML-elementtejä ja HTML:n parhaita käytäntöjä, jotta alaa opiskelevat kehittäjät osaisivat jo työelämään mennessään kehittää saavutettavampia verkkopalveluita.

6.2 Saavutettavuuden huomiointi kehittäjänä

Valmistuvana tulevana web-kehittäjänä opinnäytetyön aikana tuli opittua paljon erilaisia asioita, joita tulee hyödyntämään työelämässä. Monet asiat olivat uusia, mutta esimerkiksi semanttisten HTML-elementtien käyttö oli erittäin loogista ja asian pystyi sisäistämään suhteellisen nopeasti. Jatkossa verkkopalveluita kehittäessä tulee huomioimaan opinnäytetyön aikana opittuja parhaita käytäntöjä ja semanttisten elementtien käyttöä paljon paremmin.

WCAG 2.1:n ohjenuorien ansiosta monet aiemmin askarruttavat asiat, kuten milloin verkkosivu voidaan laskea tarpeeksi responsiiviseksi saivat ohjenuorista ja onnistumiskriteereistä suoran linjauksen, jota työelämässä ja omissa projekteissa voidaan jatkossa hyödyntää. WCAG:n maailmanlaajuinen käyttö tuo verkkosivustoille ympäri maailmaa tietynlaiset standardit, eikä saavutettavuus jää pelkästään kehittäjän tulkinnanvaraiseksi. Standardien yleistymisen myötä voidaan verkkosivustojen olettaa noudattavan samoja standardeja, jonka myötä verkkopalveluiden pitäisi olla yhdenmukaisesti saavutettavia ja käyttökokemukseltaan samankaltaisia. Tärkeänä onnistumiskriteereistä erityisesti korostui saatavilla oleva virallinen suomennos. On tärkeää saada lain velvoittamat kriteerit omalla äidinkielellään, jotta niiden noudattaminen on selvempää ja helpompaa, kun otetaan huomioon vaihteleva englannin kielen osaaminen.

6.3 Virtual Showroomin saavutettavuus selvityksen työkalut ja tulokset

Saavutettavuus selvityksen aikana saatu kokemus saavutettavuus työkalujen käytöstä oli positiivinen, ja axe-selainlisäosa osoittautui erittäin hyödylliseksi. Axen käyttäminen oli helppoa ja axen tuottama raportti oli järjestelty loogisesti. Positiivisena yllätyksenä oli axen leimaamat WCAG:n onnistumiskriteerit, joita kyseinen ongelma koski. Erityisesti mahdollisuus korostaa elementtejä, joista virheitä löydettiin, oli erinomainen. NVDA-ruudunlukijan käyttöä ei realistisesti päästy Virtual Showroomissa testaamaan, sillä näppäimistö navigointia sivulla ei opinnäytetyönaikana toiminut. Ruudunlukijan käyttö oli kuitenkin suoraviivaista ja loogista, kun sen käyttöä testattiin näppäimistön avulla muilla sivustoilla.

Virtual Showroomin saavutettavuus selvitys oli työkalujen ansiosta nopeampaa, kun axe osasi ilmoittaa monesta sivustolta löytyvästä ongelmasta automaattisesti. Kehittäjien on järkevää käyttää työkaluja kuten axe, sillä se kertoo olennaisia lain velvoittamia parannusehdotuksia sivustolle, mutta myöskin sellaisia parannusehdotuksia, jotka eivät ole välttämättä lain velvoittamia. Noudattamalla axen ilmoittamia parhaita käytäntöjä, saadaan sivustosta saavutettavampi kokonaisuudessaan. Ilman työkaluja saattaisi myös inhimillisesti joitain asioida jäädä huomioimatta, joten työkalujen käyttö on senkin vuoksi suositeltavaa.

Virtual Showroomin saavutettavuuden tilanne ennen saavutettavuusselvitystä oli osittain oletettavissa, sillä semanttisten HTML-elementtien käyttö ei ole vielä opiskeluvaiheessa opiskelijoilta odotettua. Saavutettavuuden tilanteen parantaminen ei käytännöllisessä mielessä kuulunut opinnäytetyöhön, mutta käytännönratkaisut Virtual Showroomista löydettyihin ongelmiin tulivat opinnäytetyötä tehdessä kuitenkin opittua. Semanttisten HTML-elementtien käyttö oli yksi tärkeimmistä asioista, jonka tärkeys opinnäytetyötä tehdessä opittiin.

Opinnäytetyön kaikissa tavoitteissa onnistuttiin hyvin. Opinnäytetyö oli kokonaisvaltaisesti erittäin hyödyllinen ja työn aikana opittuja asioita tullaan hyödyntämään tulevaisuudessa työelämässä.

Lähteet

- Abou-Zahra, S. 2017. Diverse Abilities and Barriers.
<https://www.w3.org/WAI/people-use-web/abilities-barriers/>.
 6.10.2021.
- Aluehallintovirasto. 2021a. WCAG 2.1: lain vaatimukset.
<https://www.saavutettavuusvaatimukset.fi/digipalvelulain-vaatimukset/wcag-2-1/>. 9.11.2021.
- Aluehallintovirasto. 2021b. Mitä palveluja ja sisältöjä laki koskee?
<https://www.saavutettavuusvaatimukset.fi/digipalvelulain-vaatimukset/mita-palveluja-ja-sisaltoja-laki-koskee/>. 27.11.2021.
- Aluehallintovirasto. 2021c. Soveltamisala: kuulummeko lain piiriin?
<https://www.saavutettavuusvaatimukset.fi/digipalvelulain-vaatimukset/soveltamisala-kuulummeko-lain-piiriin/>. 27.11.2021.
- Cooper, M. 2020. WAI-ARIA Overview.
<https://www.w3.org/WAI/standards-guidelines/aria/>. 1.11.2021.
- Deque. Axe – The Standard in Accessibility Testing.
<https://www.deque.com/axe/>. 30.11.2021.
- Enge, E. 2021. Mobile vs. Desktop Usage in 2020.
<https://www.perficient.com/insights/research-hub/mobile-vs-desktop-usage>. 22.10.2021.
- European Commission. 2021. Web Accessibility.
<https://digital-strategy.ec.europa.eu/en/policies/web-accessibility>.
 24.8.2021.
- FreeCodeCamp. 2019. Semantic HTML5 Elements Explained.
<https://www.freecodecamp.org/news/semantic-html5-elements/>.
 1.11.2021.
- Henry, S. 2021a. Introduction to Web Accessibility.
<https://www.w3.org/WAI/fundamentals/accessibility-intro/#what>
- Henry, S. 2021b. Transcripts.
<https://www.w3.org/WAI/media/av/transcripts/>. 6.10.2021.
- Henry, S. 2021c. Web Content Accessibility Guidelines (WCAG) Overview.
<https://www.w3.org/WAI/standards-guidelines/wcag/>. 5.9.2021.
- Henry, S. 2021d. WCAG 3 Introduction.
<https://www.w3.org/WAI/standards-guidelines/wcag/wcag3-intro/>.
 6.9.2021.
- Karelia-ammattikorkeakoulu. 2021. Virtuaalisen Showroom -ympäristötoteutuksen suunnittelu ja rakentaminen. Pohjois-Karjalan ja Japanin Naganon metsäbiotalousyhteistyön kehittäminen 2020–2022 – hanke. Toimeksianto ICT-toimeksiantoprojekti-opintojaksolla. 24.11.2021.
- Nielsen, J & Pernice, K. 2021. Usability Guidelines for Accessible Web Design
https://media.nngroup.com/media/reports/free/Usability_Guidelines_for_Accessible_Web_Design.pdf. 4.10.2021
- Mozilla. 2021a. CSS: Cascading Style Sheets.
<https://developer.mozilla.org/en-US/docs/Web/CSS>. 20.9.2021.
- Mozilla. 2021b. HTML: HyperText Markup Language.
<https://developer.mozilla.org/en-US/docs/Web/HTML>. 20.9.2021
- Mozilla. 2021c. What is accessibility?
https://developer.mozilla.org/en-US/docs/Learn/Accessibility/What_is_accessibility. 27.8.2021.

- Mozilla. 2021d. Handling common accessibility problems.
https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Cross_browser_testing/Accessibility. 27.8.2021.
- Mozilla 2021e. Mobile accessibility.
<https://developer.mozilla.org/en-US/docs/Learn/Accessibility/Mobile>. 22.10.2021
- Mozilla. 2021f. HTML: A good basis for accessibility
<https://developer.mozilla.org/en-US/docs/Learn/Accessibility/HTML>. 8.9.2021.
- Mozilla. 2021g. Accessible multimedia.
<https://developer.mozilla.org/en-US/docs/Learn/Accessibility/Multimedia>. 18.10.2021.
- Mozilla. 2021h. Color contrast.
https://developer.mozilla.org/en-US/docs/Web/Accessibility/Understanding_WCAG/Perceivable/Color_contrast. 20.10.2021.
- Mozilla. 2021i. CSS and JavaScript accessibility best practices.
https://developer.mozilla.org/en-US/docs/Learn/Accessibility/CSS_and_JavaScript. 28.10.2021.
- Mozilla. 2021j. WAI-ARIA basics.
https://developer.mozilla.org/en-US/docs/Learn/Accessibility/WAI-ARIA_basics. 1.11.2021.
- Mozilla. 2021k. WAI-ARIA Roles.
<https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles>. 2.11.2021.
- Mozilla. 2021l. Media queries.
https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries. 4.11.2021.
- Papunet. 2021a. Mitä on saavutettavuus?
<https://papunet.net/saavutettavuus/mita-on-saavutettavuus>. 25.8.2021.
- Papunet. 2021b. Mitä on käytettävyys?
<https://papunet.net/saavutettavuus/mita-on-kaytettavyys>. 26.8.2021.
- Papunet. 2021c. Silmän liikkeillä ohjattava kohdistin.
<https://papunet.net/saavutettavuus/silman-liikkeilla-ohjattava-kohdistin>. 2.9.2021.
- Papunet. 2021d. Kognitiiviset ja kielelliset vaikeudet.
<https://papunet.net/saavutettavuus/kognitiiviset-ja-kielelliset-vaikeudet>. 3.9.2021.
- USA Government. 2021. Accessibility Policy.
<https://www.usa.gov/accessibility>. 24.8.2021.
- WAVE. WAVE Web Accessibility Evaluation Tool.
<https://wave.webaim.org/>. 30.11.2021.
- Web Accessibility Initiative. 2021. Making the Web Accessible.
<https://www.w3.org/WAI/>. 20.9.2021.
- WebAIM. 2021. Contrast Checker.
<https://webaim.org/resources/contrastchecker/>. 22.11.2021.
- World Health Organization. 2020. Disability and health.
<https://www.who.int/news-room/fact-sheets/detail/disability-and-health>. 24.8.2021
- World Health Organization. 2021a. Blindness and vision impairment.

<https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>. 30.8.2021.

World Health Organization. 2021b Deafness and hearing loss.

<https://www.who.int/en/news-room/fact-sheets/detail/deafness-and-hearing-loss>. 31.8.2021.

W3. 2018. Web Content Accessibility Guidelines (WCAG)

<https://www.w3.org/TR/WCAG21/>. 5.9.2021.