



Lotta Lampola

Saavutettavuuden automaattitestausten työkaluvertailu

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Ohjelmistotuotanto

Insinöörityö

11.4.2024

Tiivistelmä

Tekijä: Lotta Lampola
Otsikko: Saavutettavuuden automaattitestauksen työkaluverailu
Sivumäärä: 35 sivua
Aika: 11.04.2024

Tutkinto: Insinööri (AMK)
Tutkinto-ohjelma: Tieto- ja viestintätekniikka
Ammatillinen pääaine: Ohjelmistotuotanto
Ohjaajat: Lehtori Vesa Ollikainen
Lehtori Jussi Alhorinne

Opinnäytetyössä tutkittiin saavutettavuutta, sen huomiointia osana sovelluskehitystä sekä automaattisen testauksen työkaluja saavutettavuuden arvioimiseksi. Työssä valittiin tietyt saavutettavuuden onnistumiskriteerit ja etsittiin näitä vastaavat saavutettavuuspuutteet avoimesta verkosta. Tämän jälkeen valittiin kolme testityökalua, joiden saavutettavuusanalyysien havaintoja ja käytettävyyttä verrattiin toisiinsa nähden.

Opinnäytetyön tulokset osoittivat, että kaikki kolme työkalua kykenivät tunnistamaan yksinkertaisia saavutettavuusongelmia, kuten puuttuvia tekstivastineita ja kontrasti-ongelmia. Kuitenkin nämä työkalut eivät havainneet kaikkia valittuja saavutettavuuspuutteita, erityisesti niitä jotka liittyivät näppäimistön käyttöön ja sivujen skaalautuvuuteen.

Työn loppupäätelmänä todetaan, että manuaalinen testaus ja saavutettavuuden kokonaisarviointi on tärkeä osa saavutettavuuden kehitystä. Koneelliset analyysit voivat olla hyödyllisiä ja tehostaa sovelluskehitystyötä, mutta ne eivät korvaa manuaalista arviointia.

Avainsanat: automaatio, ohjelmistokehitys, saavutettavuus

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

Abstract

Author: Lotta Lampola
Title: Comparison of Automated Accessibility Testing Tools
Number of Pages: 35 pages
Date: 11 April 2024

Degree: Bachelor of Engineering
Degree Programme: Information Technology
Professional Major: Software Engineering
Supervisors: Ollikainen Vesa, Senior Lecturer
Alhorinne Jussi, Senior Lecturer

The thesis investigates accessibility, its consideration as part of application development, and automated testing tools for evaluating accessibility. Specific accessibility success criteria were chosen, and corresponding accessibility deficiencies were searched for from the open web. Subsequently, three testing tools were selected, and their accessibility analysis findings and usability were compared.

The results showed that all three tools were able to identify simple accessibility issues, such as missing text alternatives and contrast problems. However, the tools did not detect all selected accessibility deficiencies, particularly those related to keyboard usage and page scalability.

In conclusion, manual testing and comprehensive assessment of accessibility are essential parts of accessibility development. Automated analyses can be useful and streamline application development, but they do not replace manual evaluation.

Keywords: Accessibility, automation, software development

Sisällys

Lyhenteet

1	Johdanto	1
2	Saavutettavuus digimaailmassa	2
2.1	WCAG ja sen neljä pääperiaatetta	3
2.2	Saavutettavuusohjeista digipalveluiden kehittäjille	4
3	Saavutettavuuden automaattitestausta	7
3.1	Testauskriteerit ja vertailuun valitut työkalut	8
3.2	Automaattitestaustyökalujen vertailu	12
3.3	Testitulokset ja päätelmät	27
4	Yhteenveto	32
	Lähteet	34

Lyhenteet

DOM: *Document object model*. Dokumenttiobjektimalli. DOM-puu on hierarkkinen puumallin mukainen esitys web-sivuston rakenteesta, joka koostuu eri HTML-objekteista.

HTML: *Hypertext markup language*. Standardoitu merkintäkieli verkkosivustojen rakenteen ja sisällön määrittämiseksi.

WCAG: *Web Content Accessibility Guidelines*. Maailmanlaajuinen ja kansainvälinen saatavuusohjeistus.

1 Johdanto

Digitaalinen saavutettavuus on keskeinen näkökulma nykyaikaisessa teknologiaympäristössä, jonka tavoitteena on varmistaa, että kaikki käyttäjät voivat hyödyntää verkkopalveluita ja digitaalisia sovelluksia ilman rajoituksia tai esteitä. Saavutettavuuden merkitys korostuu entisestään, kun otetaan huomioon, että digitaalinen kanssakäyminen ja palveluiden käyttö ovat olennainen osa päivittäistä elämäämme. Erityisesti julkisen sektorin on oltava edelläkävijä tässä kehityksessä, jotta varmistetaan kaikkien kansalaisten yhdenvertainen mahdollisuus käyttää tarjottuja palveluita.

Tämän opinnäytetyön keskiössä on saavutettavuuden automaattitestityökalujen vertailu ja arviointi. Opinnäytetyön tavoitteena oli selvittää, millä tavoin ja missä määrin automaattitestityökalut ja saavutettavuuden analyysityökalut voivat olla hyödyksi verkkopalveluiden kehityksessä. Tarkoituksena oli myös selvittää, onko testityökalujen käytettävyydessä, havainnoissa ja raportointitavoissa suuria eroavaisuuksia ja mitä yhtäläisyyksiä eri palveluntarjoajin työkaluilla on. Työssä tarkastellaan näitä automaattitestaustyökaluja erityisesti näkövammaisten ja liikuntarajoitteisten käyttäjäryhmien näkökulmasta, jotka saattavat kohdata erityisiä haasteita digitaalisten palveluiden käytössä.

Raportin toisessa luvussa esitellään saavutettavuus käsitteenä, WCAG-ohjeisto ja sen merkitys saavutettavuustyössä sekä yleisiä hyviä käytänteitä, joita verkkopalveluiden kehittäjien tulisi hyödyntää. Huomioimme myös WCAG-saavutettavuusohjeiston neljä pääperiaatetta, jotka muodostavat perustan verkkosisältöjen ja sovellusten saavutettavuudelle.

Työn kolmannessa luvussa esitellään saavutettavuuden alueet ja kriteerit, joita vasten työkaluvertailu toteutettiin. Tämän jälkeen raportissa kuvataan itse vertailutyö ja vertailun tulokset.

Työn tavoitteena on tarjota kattava katsaus erilaisiin saavutettavuuden automaattitestyökaluihin sekä niiden soveltuvuuteen eri käyttötilanteissa. Toivon, että tämä opinnäytetyö voisi tarjota näkökulmia saavutettavuuden parantamiseksi digitaalisessa ympäristössä, mikä edistää näin yhdenvertaisuutta ja inklusiivisuutta verkossa.

2 Saavutettavuus digimaailmassa

Saavutettavuus tarkoittaa erilaisten ihmisten toimintarajoitteiden huomioon ottamista digitaalisessa ympäristössä. Verkkopalveluiden saavutettavuuden tarkoituksena on, ettei palvelun tekninen toteutus sulje mitään käyttäjäryhmää ulos ja estä palvelun käyttöä keneltäkään. (1, s. 11) Saavutettavuutta voidaan verrata fyysisen maailman esteettömyyteen. Ilman saavutettavuutta on monenlaisilla ihmisryhmillä suuria haasteita käyttää digitaalisia palveluita, tai palvelun käyttö voi estyä kokonaan. (1, s. 15; 2, s. 2.) Asioita ja ominaisuuksia, joita saavutettavuudella pyritään huomioimaan, ovat mm.

- näkövammatt
- fyysiset ja motoriset rajoitteet
- kuurous ja kuulovammatt
- kehitysvammatt
- muistihäiriöt ja -sairaudet
- luku- ja oppimisvaikeudet
- keskittymishäiriöt
- muut kognitiiviset rajoitteet
- heikko kielitaito
- tilapäiset haasteet, kuten meluista ympäristö, kirkas auringonpaiste, stressi tai kipsissä oleva käsi.

Listaus on laaja, ja pelkästään Suomessa on arvioitu jopa miljoonalla kansalaisella olevan vaikeuksia käyttää digipalveluita (3). Saavutettavuus tuottaa myös hyötyjä kaikille käyttäjille. Saavutettavuuden yleisiä etuja on mm.

riippumattomuus päätelaitteesta, selaimesta ja fonttikoosta sekä hakukoneiden parempi indeksointi. (1, s. 14-16; 4.)

Vuonna 2022 84 % EU:n kansalaisista ja jopa 96 % nuorista ilmoittivat käyttävänsä internetiä päivittäin (5; 6). Iso osa työstämme, oppimisestamme ja muusta asioinnistamme toteutuu digitaalisesti. Täten Euroopan unionin parlamentti ja neuvosto säätivät vuonna 2016 direktiivin, jonka tavoite on tehdä julkisen sektorin verkkopalveluista ja mobiilisovelluksista paremmin saavutettavia, sekä yhdenmukaistaa vaihtelevia standardeja. Suomessa on säädetty myös kansallinen digipalvelulaki. (7; 1, s. 17.)

Tässä työssä pyritään keskittymään saavutettavuuteen näkövammaisten ja -rajoitteisten sekä liikuntarajoitteisten, joilla ei ole mahdollisuutta käyttää esim. hiirtä tai touchpadia, näkökulmasta. Kuntaliiton mukaan Suomen väestöön kuuluu noin 10 000 sokeaa, 50–60 000 näkövammaista ja 70 000 heikkonäköistä ihmistä. Liikuntarajoitteisia kansalaisia voi olla jopa 220 000 ihmistä. (3.)

2.1 WCAG ja sen neljä pääperiaatetta

Kansainvälinen tietoverkkojen standardeja ja suosituksia kehittävä ja ylläpitämä World Wide Web Consortiumin (W3C) ylläpitää saavutettavuusohjeistusta Web Content Accessibility Guidelines (WCAG). WCAG-ohjeistuksia on n. 80, joille on laadittu onnistumiskriteerit, jotka on jaoteltu kolmeen eri tasoon: A-, AA- ja AAA, joista AAA on kaikkein korkein. EU:n saavutettavuuskriteeri velvoittaa julkisen sektorin digipalveluiden täyttävän A- ja AA-tason onnistumiskriteerit.

WCAG-saavutettavuusohjeistuksen suositukset ja onnistumiskriteeristö on rakennettu neljän pääperiaatteen ympärille. Lyhyt kuvaus neljästä periaatteesta on esitetty alla:

1. Havaittava. Kaikelle ei-tekstuaaliselle sisällölle tulee tarjota tekstivastine. Kaikelle aikasidonnaiselle medialle tulee tarjota vaihtoehtoinen vastine. Sisältö voidaan esittää eri tavoin informaatiota tai rakennetta menettämättä.

Taustasta erottuva etuala tulee lisätä helpottamaan käyttäjiä näkemään ja/tai kuulemaan sisältö.

2. Hallittava. Kaikki toiminnallisuus tulee olla käytettävissä näppäimistöltä. Käyttäjille tulee tarjota tarpeeksi aikaa lukea ja käyttää sisältöä. Sisällön ei tulisi olla sellaista, joka aiheuttaa sairaskohtauksia esim. epileptikoille. Käyttäjille tulee tarjota tapoja navigoida, etsiä sisältöä ja määrittää sijaintinsa palvelussa.
3. Ymmärrettävä. Tekstisisällön tulee olla luettavaa ja ymmärrettävää. Palvelun ulkoasu ja toiminta täytyy olla ennakoitavissa. Käyttäjien tulee pystyä välttämään sekä korjaamaan virhetilanteita.
4. Toimintavarma. Palveluiden tulee olla mahdollisimman yhteensopivia nykyisten sekä tulevien asiakasohjelmien kanssa, kuten avustavat teknologiat. (1, s. 23–24; 8.)

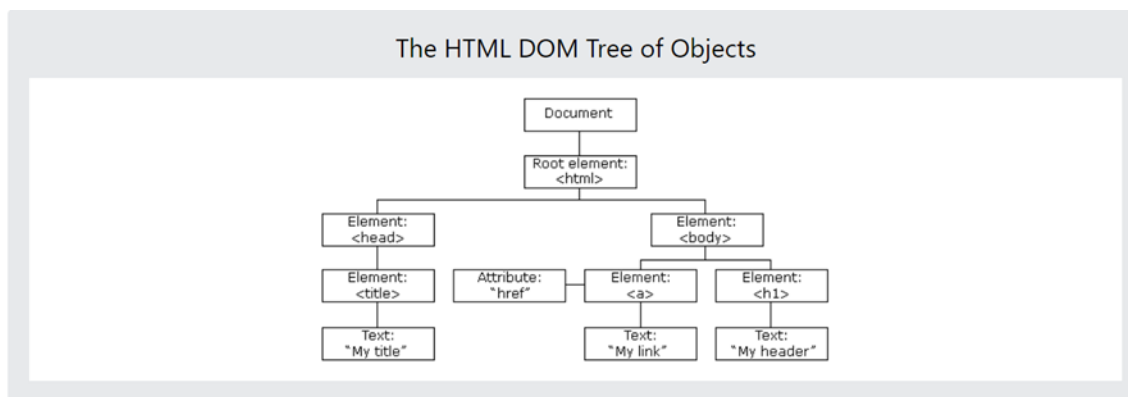
2.2 Saavutettavuusohjeista digipalveluiden kehittäjille

Heikkonäköisille ja näkövammaisille on useita tapoja helpottaa digipalveluiden käyttöä. Heikkonäköiset ja värisokeat voivat hyödyntää esimerkiksi käyttöliittymäelementtien suurentamista ja käänteisiä värejä. Näkövammaiset, etenkin sokeat ihmiset käyttävät digipalveluja ruudunlukijan avulla, ja hiiren sijaan liikkuvat si-vustolla näppäimistön avulla. Näppäimistöä käyttävät myös monet ihmiset, joilla on motorisia tai fyysisiä rajoitteita. Heille saavutettavuus tulisin huomioida niin näppäimistö- kuin hiirikäytössäkin. (1, s. 34-35; 3.)

On monia asioita ja tapoja, joilla verkkopalvelun voi toteuttaa saavutettavammaksi heikkonäköisille, värisokeille ja muille näkövammaisille sekä ihmisille, joilla on motorisia tai fyysisiä rajoitteita. Tässä luvussa esitellään saavutettavuusohjeita erityisesti näiden ryhmien näkökulmasta.

Lähdekoodin laatu ja semanttinen HTML

Saavutettava käyttöliittymä on helpompi toteuttaa, kun lähdekoodi noudattaa hyviä ohjelmointikäytäntöjä ja on laadukasta. Modernit käyttöliittymäkoodikirjastot, kuten React ja Angular mahdollistavat verkkosivun HTML DOM -puun (kuva 1) dynaamisen muokkaamisen ja ohjelmistokehityksessä kirjoitetaan enää harvoin staattista HTML-koodia. (9, s. 292; 10.)



Kuva 1. DOM-puu (11).

Saavutettavuuden kannalta on tärkeää huolehtia, että huolimatta modernien käyttöliittymäteknologioiden mahdollistamasta dynaamisesta sisällöstä verkkopalveluiden DOM-puu noudattaa semanttisen HTML:n periaatteita. Semanttinen HTML tarkoittaa, että elementit ovat loogisessa järjestyksessä ja valitut html-elementit tarjoavat tietoa sisällöstään. Ruudunlukijat lukevat verkkopalvelun html-koodia normaalissa lukujärjestyksessä ylhäältä alas ja oikealta vasemmalta. Täten on tärkeää, että html-koodin järjestys kuvaa myös graafisen käyttöliittymän esitysjärjestyttä. Ruudunlukijaa hyödyntävä käyttäjä saa myös paljon tietoa html-elementeistä, joten on tärkeää valita käytettävät elementit oikein, esimerkiksi `<p>`-tagit tekstielementteihin ja `<a>`-tagit linkeille jne. (12; 13.)

Skaalautuvuus

Palvelun sisältö pitäisi pystyä skaalaamaan vähintään 200 %:iin menettämättä sisältöä tai toiminnallisuutta. Yleinen suositus on, että sisällön voisi esittää vähintään leveyssuunnassa 320 CSS pikselissä ja korkeussuunnassa 256 CSS

pikselissä ilman, että sisällön tarkastelu ei vaadi vieritystä kahdessa suunnassa. Käyttöliittymän rakennetta suunniteltaessa ja toteutettaessa tulisi varmistaa, että elementit järjestyvät loogisesti skaalausta kasvattaessa. On käyttäjäystävällisempää asetella komponentit niin, että palvelussa on vain pystysuuntavieritys. (14.)

Kontrastit ja värit

Verkkopalveluiden sisältö kannattaa esittää suurikontrastisesti, mutta pelkkiä värejä ei tule käyttää tiedon ilmaisuun. Normaalikokoisen tekstin kontrastin tulee olla vähintään 4,5:1 ja ison tekstin vähintään 3:1. Kaikki käyttäjät eivät näe värejä tai ovat värisokeita, joten värejä ei tule yksinään käyttää toiminnallisuuden tai tekstin ilmaisuun. Tämä on hyvä muistaa etenkin notifikaatioita ja virheilmoituksia kehitettäessä. (1, s. 65; 14; 15.)

Kuvien, ikonien, videoiden ja muun median tekstivastineet

Verkkopalvelut sisältävät usein myös ei-tekstuaalista sisältöä, kuten esim. kuvia, infograafeja, ikoneita ja logoja. Kaikella ei-tekstuaalisella sisällöllä tulee olla tekstivastine, jotta myös ruudunlukijaa käyttävät ihmiset pääsevät käsiksi sisältöön. Tekstivastine/vaihtoehtoinen teksti ei ole sama kuin kuvateksti tai kuvan otsikko. Tekstivastine tarjotaan usein alt-attribuuttina html img -elementissä. Myös linkkeihin on hyvä lisätä vaihtoehtoinen teksti, joka kertoo käyttäjälle, minne linkki vie. (1, s. 62; 14; 15.)

Näppäimistöllä kohdistaminen

Verkkopalvelun kaikki tieto ja toiminnallisuus olisi hyvä olla saavutettavissa pelkkää näppäimistöä käyttäen. Näppäimistöä voidaan käyttää joko verkkopalvelun visuaalisen informaation tai ruudunlukuohjelman kanssa. Näkeville käyttäjille on olennaista, että näppäimistöllä liikkuessa aktiivinen elementti on selkeästi ja havaittavasti visuaalisesti korostettu. Ruudunlukijaa käyttäville käyttäjille on hyvä tarjota mahdollisuus sivuuttaa palvelun jokaisessa näkymässä toistuva sisältö, kuten esim. yläpalkin komponentit, kielivalinnat ja päävalikko,

käyttämällä tähän tarkoitukseen tarkoitettua näppäinoikotietä. Tällainen toiminnallisuus helpottaa käyttäjän navigointia ja tekee verkkopalvelun käytöstä sujuvampaa ja tehokkaampaa.

Standardinäppäimet sivustolla liikkumiseen ovat tab, esc, enter, välilyöntinäppäin sekä nuolinäppäimet. Kaikille näppäimistöllä liikkuville käyttäjille on tärkeää, että näppäimistön kohdistusjärjestys noudattaa samaa järjestystä kuin visuaalinen rakenne ja säilyttää palvelun merkityksen ja toimivuuden. On olennaista huolehtia, että myös komponentin tai elementin päältä tulee pystyä siirtymään kohdistus pois myös pelkän näppäimistön avulla. (1, s. 34-35 ja 70; 14.)

Motoristen rajoitteiden huomiointi

Motoriset ja muut fyysiset rajoitteet eivät välttämättä estä hiiren (tai kosketusnäytön) käyttöä, mutta on useita asioita, jota on silti hyvä huomioida käyttöliittymäsuunnittelussa. Esimerkiksi vapina käsissä voi lisätä haastetta käyttää liukuelementtejä sekä muita käyttöliittymäkomponentteja, jotka vaativat tarkkaa hiiren käyttöä. Asettelu on hyvä toteuttaa väljästi, jolloin erilaisten komponenttien väliin jää tilaa. Klikattavien ja napsautettavien komponenttien riittävä koko on hyvä huomioida, jotta mahdollisimman moni käyttäjä pystyisi kohdistamaan niihin hiirellä. Monelle käyttäjälle on myös suuri hyöty, jos käyttöliittymässä on mahdollista käyttää automaattista täyttöä esim. lomakkeissa. (14; 16.)

3 Saavutettavuuden automaattitestausta

Saavutettavuus on olennainen osa modernia sovelluskehitystä. Saavutettavuus vähentää erilaisten ihmisten ja käyttäjien eriarvoisuutta ja julkisen sektorin digipalvelut ovat laissa velvoitettuja huolehtimaan palveluiden saavutettavuudesta. Saavutettavien verkkopalveluiden tekninen toteutus usein edistää myös sivun hakukoneoptimointia. Parhaassa tapauksessa saavutettavuus lisää käyttäjiä kahdella tapaa: palvelu on saavutettava monenlaisille käyttäjille ja käyttötaivoille, ja verkkopalvelun näkyvyys hakukoneissa paranee. (1, s. 16-17.)

Sovelluskehityksessä testausautomaatio vähentää regressiota ja tehostaa työkentelyä. Saavutettavan verkkopalvelun kohderyhmät ja käyttötarpeet ovat hyvin moninaisia ja täten mahdollisimman saavutettavan palvelun suunnittelu ja toteutus on kompleksinen kokonaisuus (1, s. 14). Saavutettavuuden automaattitestauksella voidaan tehostaa kehitettävän verkkopalvelun saavutettavuuden arviointia. Automaattitestauksen avulla voidaan tunnistaa tekniseen toteutukseen liittyviä saavutettavuusongelmia jo varhaisessa vaiheessa, joka säästää resursseja.

3.1 Testauskriteerit ja vertailuun valitut työkalut

Alaluvussa 2.2 on nostettu heikkonäköisten, värisokeiden, näkövammaisten sekä motorisia tai fyysisiä rajoitteita omaavien käyttäjien kannalta olennaisia ohjeita ja hyviä käytänteitä. Testaustyökalujen testauskriteerit on rakennettu WCAG-kriteereistä, jotka liittyvät näihin aiheisiin.

Tietojärjestelmätieteen pro gradu -tutkielmassa Verkkopohjaisten karttojen saavutettavuushaasteet, Petra Jokisuu tarkastelee erityisryhmien saavutettavuushaasteita verkkopohjaisilla kartta-alustoilla. Tutkimuksessa havaittiin yleisimpiä käyttöhaasteita rajoiteryhmittäin. Jokisuun löydöksenä on, että motorisen rajoiteryhmän yleisimpiä haasteita olivat visuaaliset elementit/nimettömät painikkeet sekä epäselvät painikkeet. Näkörajoitteisen tutkimusryhmän yleisimmät haasteet olivat visuaalisen informaation lukeminen ja värit. Molemmille tutkimusryhmille yhteisiä yleisimpiä haasteita olivat sisällön runsas määrä ympäri käyttöliittymää sekä elementtien ja tiedon epäjärjestelmällinen asettelu. (16.)

Saavutettavuusohjeistuksiin (1; 7; 8; 14) ja Jokisuun tutkielman havaintoihin perustuen valittiin seitsemän saavutettavuuden onnistumiskriteeriä tarkasteltavaksi (taulukko 1). Valinta pyrittiin tekemään siten, että kyseisten kriteerien täyttäminen ratkaisisi yleisimmät ja kaikista haitallisimmat verkkopalvelun saavutettavuuspuutteet. Taulukon ensimmäisessä sarakkeessa on onnistumiskriteerin numerotunniste, toisessa sen vaatimusluokka (A, AA tai AAA) ja kolmannessa onnistumiskriteeri sanallisesti esitettynä.

Taulukko 1. Tarkasteluun valitut saavutettavuuskriteerit (8).

WCAG 2.1 onnistumiskriteeri	Luokka	Selite/Ehto
1.4.4	AA	Tekstin kokoa voidaan suurentaa vähintään 200 % ilman, että sisältöä tai toiminnallisuutta menetetään. Kriteeri ei koske selite- tai kuvatekstejä.
1.4.10	AA	Sisältö voidaan esittää ilman, että mitään tietoa tai toiminnallisuutta hävitetään, ainakin 320 X 256 CSS pikselin kokoisessa näytössä siten, ettei näkyvässä ole kahdensuuntaista vieritystä.
1.4.3	AA	Normaalikokoisen tekstin kontrastin tulee olla vähintään 4,5:1 ja ison tekstin vähintään 3:1.
1.4.1	A	Tiedon tai toiminnallisuuden esittämiseen, vastauksen kehottamiseen tai visuaalisen elementin erottamiseen ei tule käyttää pelkkää väriä ainoana indikaattorina.

1.1.1	A	Kaikelle ei-tekstuaaliselle sisällölle tulee esittää tekstivastine.
2.1.3	AAA	Kaikki toiminnallisuus ja sisältö tulee olla saavutettavissa pelkällä näppäimistöllä, ilman aikarajoitteita tietyille näppäinvalinnoille.
2.1.2	A	Mikäli komponenttiin voidaan kohdistaa näppäimistöllä, myös poiskohdistus komponentista näppäimistöllä tulee olla mahdollinen. Mikäli poiskohdistus edellyttää muita kuin standardi poistumisnäppäinkomentoja, tulee käyttäjää ohjeistaa, miten poiskohdistus voidaan toteuttaa.

Testaustyökalujen valinta vertailuun

Saavutettavuuden testaukseen on kehitetty lukuisia työkaluja, joista Web Accessibility Initiative W3C:n verkkosivuilla on esitetty 166 työkalua. Palveluita löytyy useassa muodossa, muun muassa omalle työasemalle asennettavia ohjelmistoja, ohjelmistoympäristöihin asennettavia lisäosia, verkkopalveluita sekä selaimeen asennettavia laajennoksia. (17.)

Testaustyökalujen vertailuun valitsin kolme työkalua: Wave Evaluation Tool, Lighthouse ja AxeDevTools – Web accessibility Testing. Kaikki kolme valittua työkalua toimivat selainlaajennoksena.

Ennen työkaluvertailua työn aikana tutustuttiin valittujen työkalujen dokumentaatioon ja markkinalupauksiin. Wave Evaluation Tool on voittoa tavoittelemattoman WebAim:in kehittämä palvelu. WebAim-organisaatio on osa Utahin osavaltion yliopiston vammaistutkimuksen, -politiikan ja -käytännön instituuttia. Wave-laajennos ei vaadi analysoitavan sivuston datan siirtämistä käytettävän selaimen ulkopuolelle, eikä tietoa siirry Wave-palvelun tarjoajalle. Kaikki Waven keräämä tieto anonymisoidaan eli myöskään työkalua käyttävän laitteen tai käyttäjän tietoja ei tallenneta palveluntarjoajan rekistereihin. Tietosuojaselosteessaan WebAim myös lupaa, ettei käyttäjän tietoja koskaan myydä kolmansille osapuolille. Markkinoinnissaan Wave painottaa selkeästi helppokäyttöisyyttä, yksityisyyttä ja tietoturvaa. (18.)

Toisin kuin Wave, Dequen tarjoama ilmaisversio axe DevTools-työkalusta kerää yksilöitäviä tietoa käyttäjästänsä sekä sivustosta, jonka saavutettavuutta analysoidaan. Työkalusta on tarjolla myös maksullinen versio axe DevTools Pro, joka tarjoaa lisätoiminnallisuuksia. Tässä vertailussa käytettiin ainoastaan ilmaisversiota. Palvelun markkinointimateriaalissa vedotaan selkeästi Dequen kokemukseen ja suosioon saavutettavuuden kehityksessä. Palvelu kertoo käyttävänsä myös tekoälyä, joka tehostaa saavutettavuusarviointia ja lupaa tarjota kattavan analyysin muita työkaluja nopeammin. (19.)

Lighthouse on Googlen avoimen lähdekoodin ohjelmisto, jonka tarkoitus on tarjota ratkaisuja verkkosivujen laadun parantamiseen useasta eri näkökulmasta kuten saavutettavuus, suorituskyky sekä hakukoneoptimointi. Tässä opinnäytetyössä ja vertailussa käytettiin Lighthousen saavutettavuusanalyysityökalua. (20.)

Näiden kolmen tuotteen dokumentaation perusteella axe DevTools ja Wave ovat juuri saavutettavuuden arviointiin tarkoitettuja työkaluja, kun Lighthouse taas analysoi sivuston laatua laajemmin. Markkinointimateriaalin perusteella Dequen axe DevTools profiloituu ammattilaiskäyttöön tarkoitettuna työkaluna ja houkuttelee maksavia yritysasiakkaita, kun Wave tarjoaa ilmaista, helposti käytönotettavaa ja yksityisyyttä varjelevaa palvelua. Kaikkien kolmen palvelun

tarjoajat ovat suuria, teknologiaan ja sovelluskehitykseen erikoistuneita organisaatioita.

Valitsin Wave Evaluation Toolin, Lighthousein ja AxeDevToolsin saavutettavuustestaustyökalujen laajasta tarjonnasta, sillä ne kaikki nousivat usean kerran esiin etsiessäni tietoa suosituista ja hyväksi havaituista palveluista. Saavutettavuuden suhteen näiden työkalujen lupaukset ovat melko samankaltaisia, mutta kokonaisprofiilit ja kohderyhmät kuitenkin eroavat hieman.

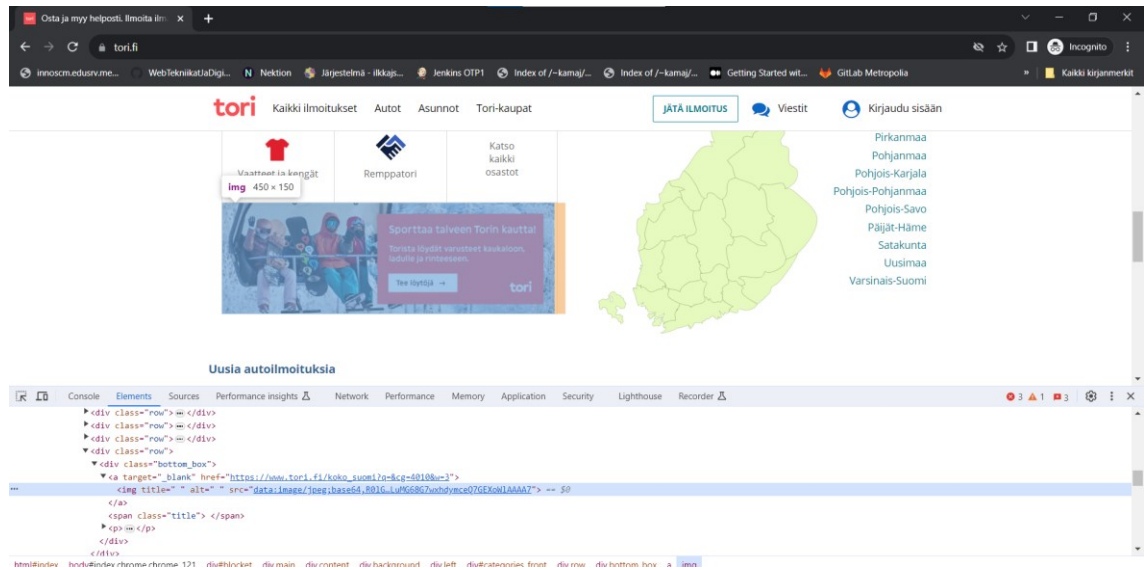
3.2 Automaattitestaustyökalujen vertailu

Tarkasteltavien saavutettavuuskriteerien ja vertailtavien analyysityökalujen valinnan jälkeen tutkielmaan tuli löytää kutakin WCAG-kriteeriä vastaava saavutettavuuspuute joltakin verkkosivustolta. Kun sopiva kohdesivu tai sen osa löytyi, suoritettiin automaattitestausta kaikilla kolmella työkalulla Wavella, axeDevToolsilla sekä Lighthouseella. Tässä luvussa esitetään testikohteiksi valitut, saavutettavuuspuutteita sisältävät verkkopalvelut ja kunkin automaattitestaustyökalun havainnot kohdesivulta.

Kriteeri 1.1.1: Kaikelle ei-tekstuaaliselle sisällölle tulee esittää tekstivastine (8).

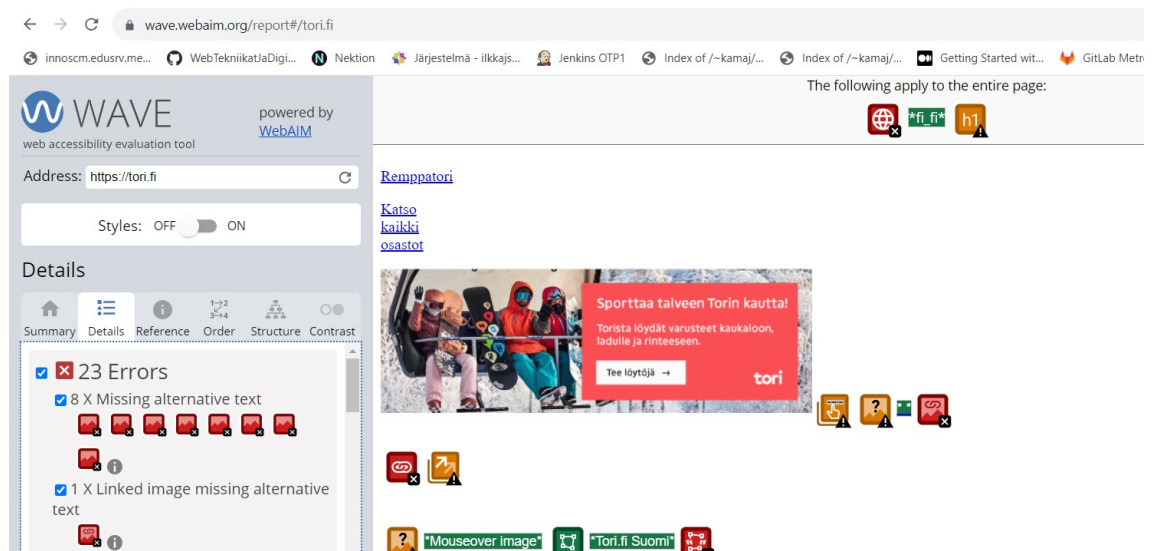
Hyvin yleinen ei-tekstuaalinen sisältö verkkosivustolla on kuvaelementit. Lähtökohtaisesti HTML img-elementti tulisi aina sisältää ominaisuustiedon alt- tai title-attribuuttina, joka kuvaa saavutettavasti kuvasisältöä ja sen tarkoitusta verkkosivustolla. Poikkeustilanteita on esimerkiksi, jos kuvalla ei ole varsinaista merkitystä saavutettavuusnäkökulmasta, vaan se toimii eksplisiittisesti koristeena, kuten esim. sivuston taustakuvana.

Sivuston tori.fi-etusivulta löytyi heti useampi kuvaelementti, josta puuttui tekstivastine (kuva 2).

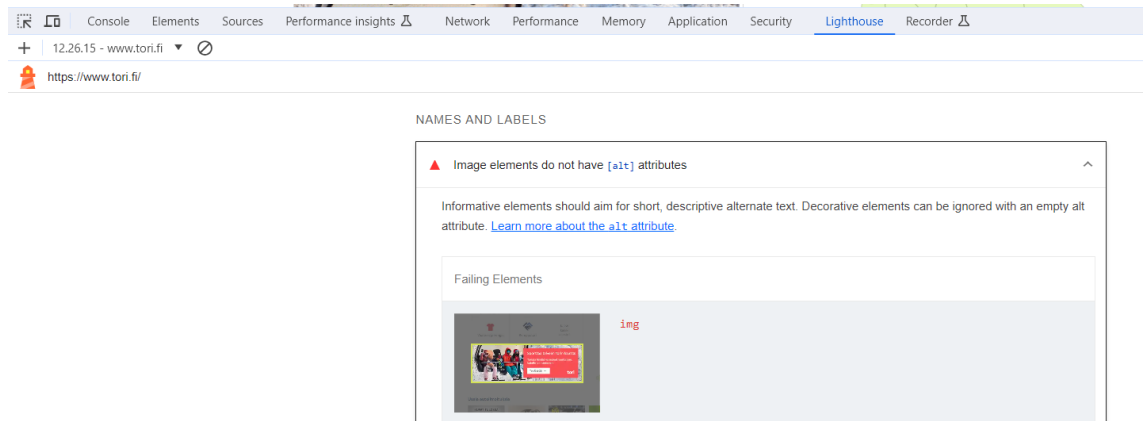


Kuva 2. Esimerkki kuvaelementistä, jolla ei tekstivastinetta.

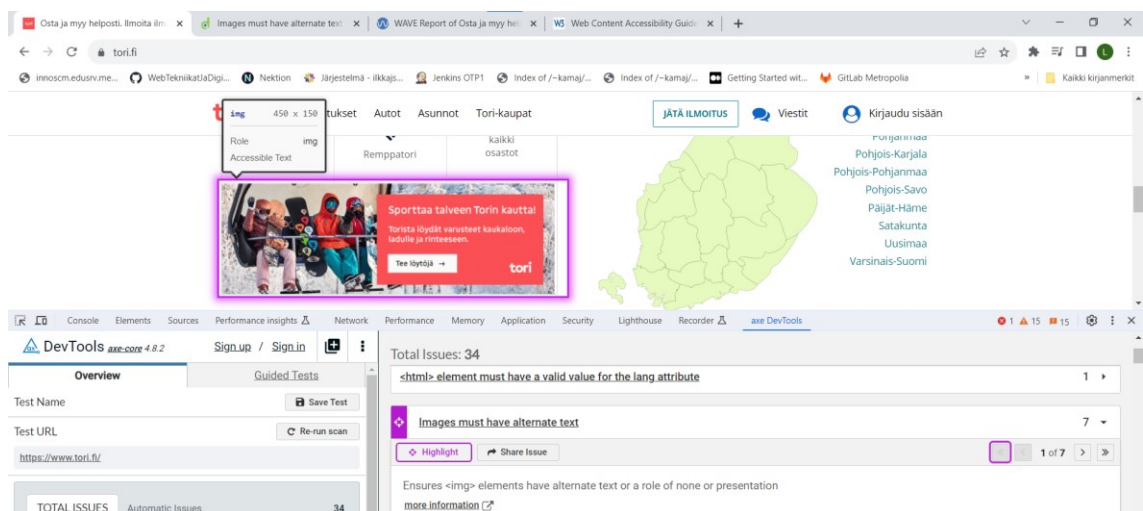
Kaikki kolme saavutettavuuden analysointityökalua löysivät saavutettavuuspuutteen. Kuvista 3–5 voidaan nähdä kunkin automaattitestyökalun havainnot kuvasta, josta puuttuu tekstivastine.



Kuva 3. Testin Wave-tulos



Kuva 4. Testin Lighthouse-tulos



Kuva 5. Testin axe DevTools-tulos

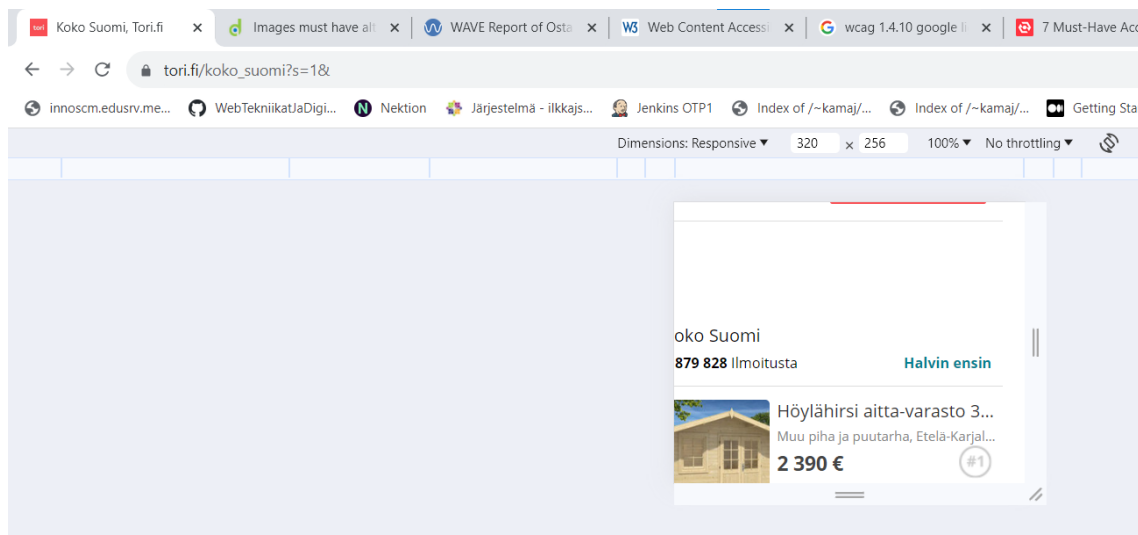
Kriteeri 1.4.10: Sisältö voidaan esittää ilman, että mitään tietoa tai toiminnallisuutta hävitetään, ainakin 320 X 256 CSS pikselin kokoisessa näytössä siten, ettei näkymässä ole kahdensuuntaista vieritystä (8).

Sivuston tori.fi näkymä ”Kaikki ilmoitukset” (url

https://www.tori.fi/koko_suomi?s=1&) ei täyttänyt saavutettavuuskriteeriä

1.4.10. Kokeilemalla Chromen device-näkymässä ruudunkokoa 320px x 256px

sivulle tuli kaksisuuntainen vieritys. Kuvassa 6 voidaan nähdä, miltä sivusto näytti halutussa ruutukoossa.

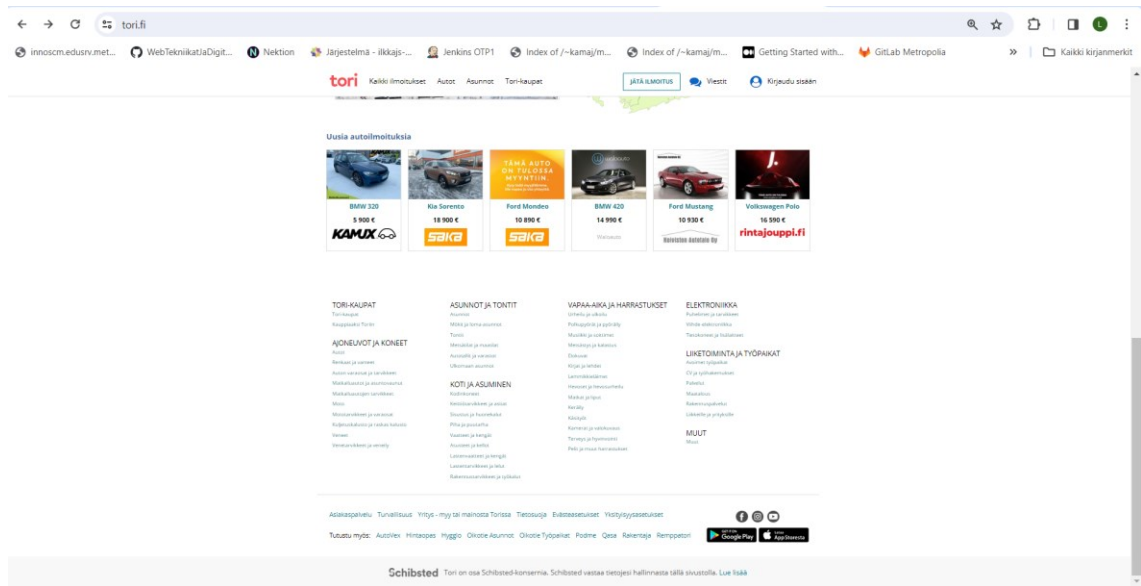


Kuva 6. Tori.fi-sivuston Kaikki ilmoitukset -näkyminen koossa 320x256 pikseliä.

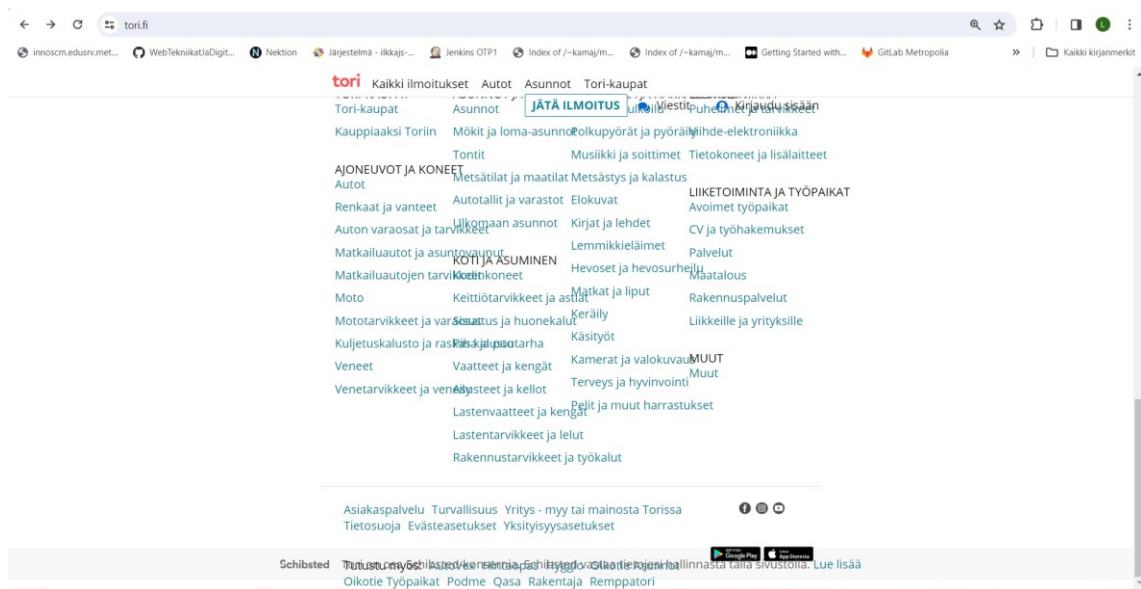
Yksikään vertailtavista testityökaluista ei havainnut saavutettavuuspuutetta.

Kriteeri 1.4.4: Tekstin kokoa voidaan suurentaa vähintään 200 % ilman, että sisältöä tai toiminnallisuutta menetetään. Kriteeri ei koske selite- tai kuvatekstejä. (8.)

Sivuston tori.fi etusivulla sekä näkyvässä "Kaikki ilmoitukset" osassa toiminnallisuudesta menetettiin, kun tekstin kokoa suurennettiin 200 %. Alla olevissa kuvaesimerkeissä (kuvat 7-10) esitetään sivuston alaosa ensin oletuskokoisella tekstillä ja toisessa kuvauksessa itse selaimen zoomaus on asetettu 50 %:iin ja tekstin suurennus 200%:iin.



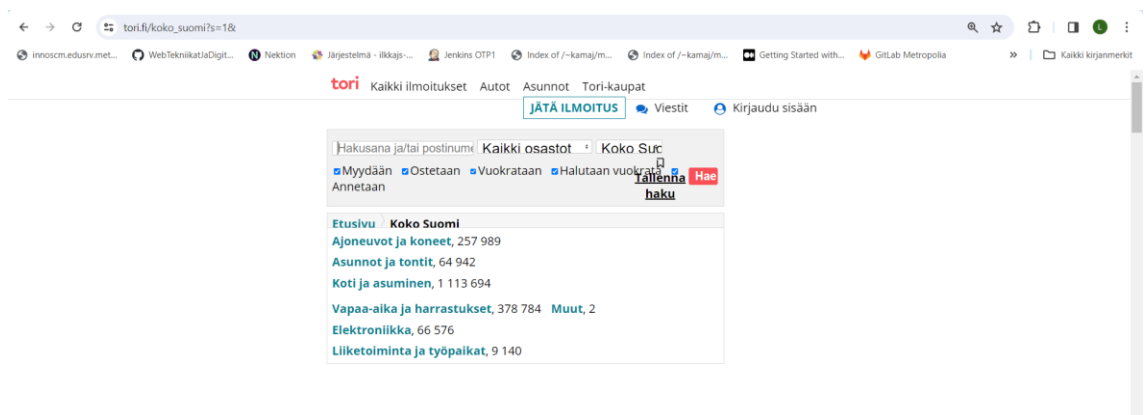
Kuva 7. Tori.fi-sivuston etusivun sivukartta normaalikokoisena.



Kuva 8. Tori.fi-sivuston sivukartta, kun tekstin zoomaus asetettu 200 prosenttiin.



Kuva 9. Osoitteen https://www.tori.fi/koko_suomi?s=1& ulkoasu normaalikokoisena.

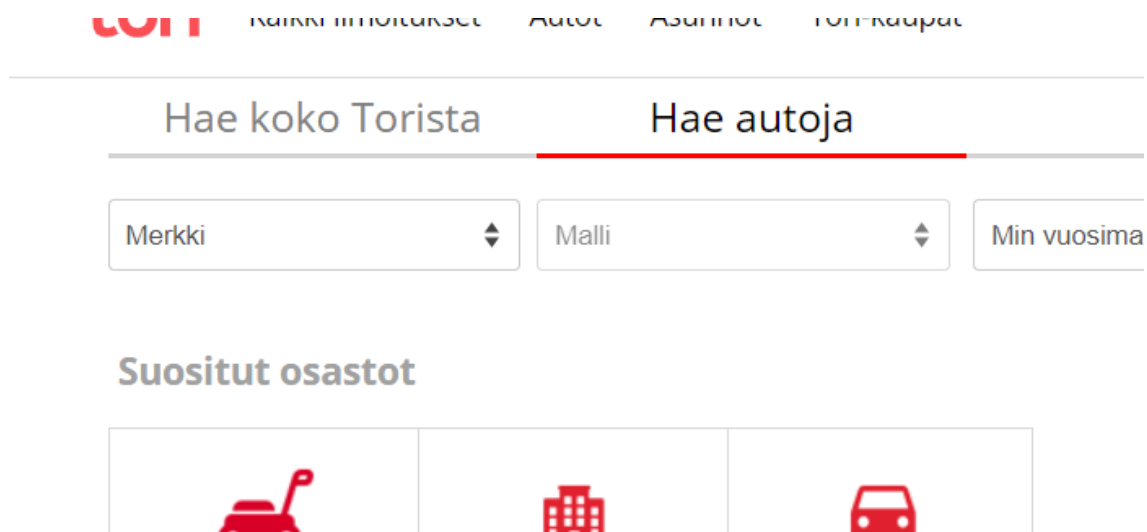


Kuva 10. Osoitteen https://www.tori.fi/koko_suomi?s=1& ulkoasu, kun tekstin koko suurennettu 200 prosenttiin.

Molemmissa esimerkkitapauksissa voidaan havaita kuvista sekä testaamalla, että osa klikattavista elementeistä on päällekkäin, eikä täten kaikkiin klikattaviin kohteisiin pysty kohdistamaan hiirellä. Mikään työkaluista ei havainnut tätä puutetta.

Kriteeri 1.4.3: Normaalikokoisen tekstin kontrastin tulee olla vähintään 4,5:1 ja ison tekstin vähintään 3:1 (8).

Tori.fi-sivuston hakulomakkeessa oli saatavuuspuutteita, jotka koskivat vähimmäiskontrastia. Alla olevassa Kuvassa 11 teksti ”Hae koko Torista” sekä ”Suositut osastot” olivat molemmat kontrastia 2.58:1.

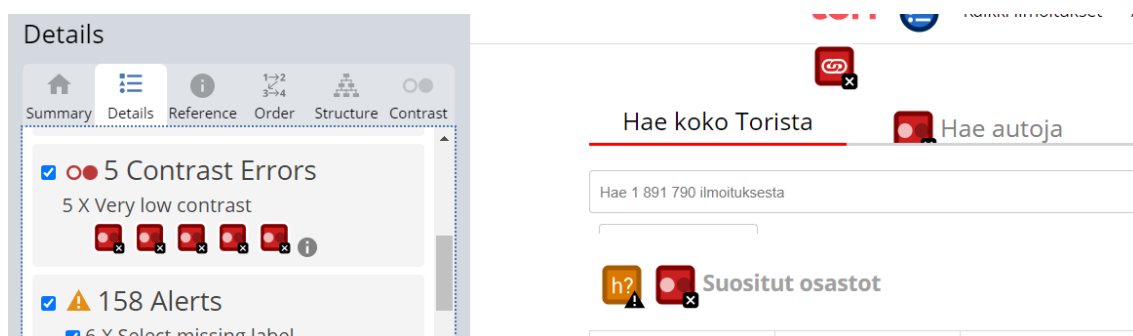


Kuva 11. Esimerkkikuva Tori.fi-sivuston tekstien kontrastista.

Kaikki työkalut havaitsivat puutteet tekstien kontrastissa. Tätä saavutettavuuspuutetta vasten testatessa huomasi, että kunkin testityökalun käytettävyydessä on selkeitä eroja. Välillä koin haastavaksi löytää yksilöllisen elementin, jota työkalun havainto koski. Kaikista helpoimmaksi tietyn puutteellisen elementin havainnoinnin koin axe DevTools -työkalulla.

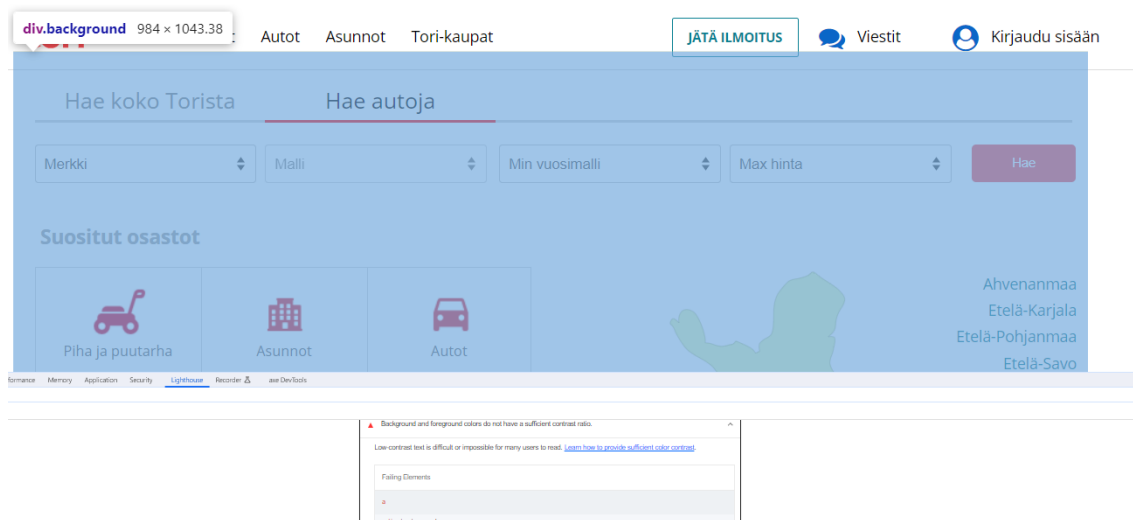
Waven raportti manipuloi analysoimansa sivuston sisältöä lisäämällä puutemerkintöjä suoraan sivustolle (kuva 12). Nämä kuvakkeet ajoittain ns. rikkoivat

sivuston asettelu, jolloin tuloksien lukeminen oli välillä haastavaa.



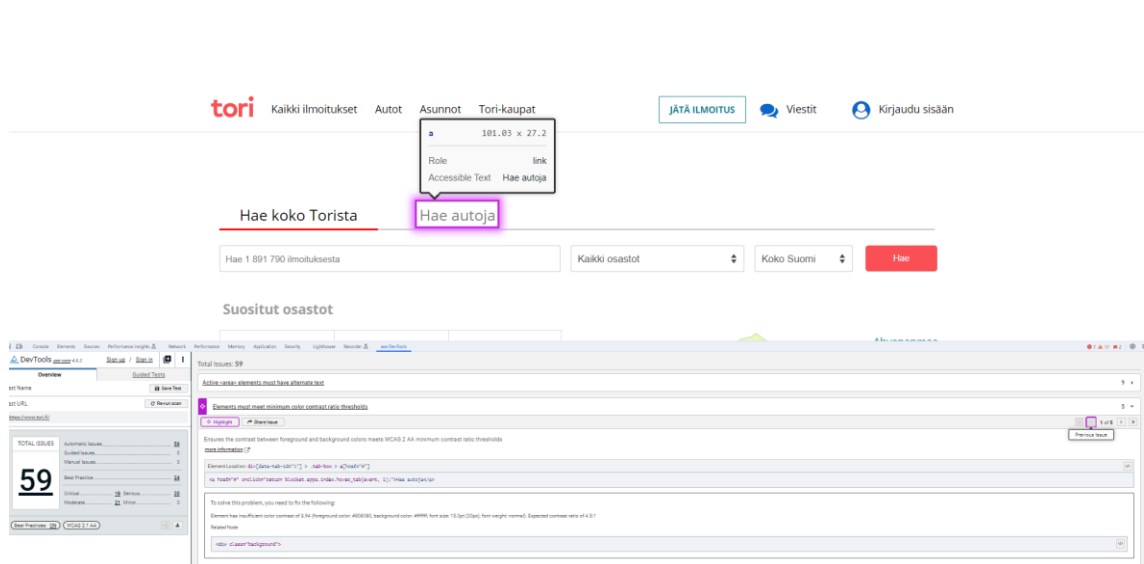
Kuva 12. Esimerkkikuva Wave-työkalun havaintojen esitystavasta.

Google Lighthouse taas ei osoittanut visuaalisesti, eikä elementin yksilöllisellä tunnisteella tarkkaa paikkaa, jossa puute esiintyi. Työkalu osoitti pelkästään ylempään tason komponentin, jota vasten alikomponentin kontrastipuute esiintyi. Kuvassa 13 on esimerkki, jossa a-elementin värikontrasti on puutteellinen suhteessa sen yläkomponenttiin div-elementtiin. Hiiren viemällä raportin kohtaan Lighthouse maalasi kyseessä olevan div-elementin. Tekstuaalinen raportti osoitti, että kyseessä on a-elementti. Työkalu ei kuitenkaan osoittanut yksilöivästi, mikä a-elementti on kyseessä.



Kuva 2. Google Lighthousen visuaalinen ja tekstuaalinen tapa esittää saavutettavuuspuute.

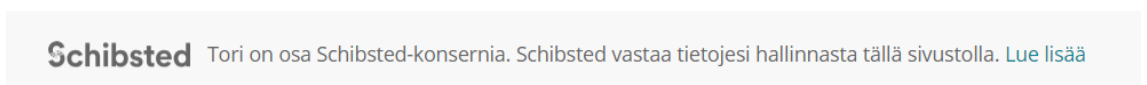
Oman kokemuksen mukaan Axe DevToolsien tapa esittää saavutettavuushavaintoja ja -puutteita sisältävä komponentti oli kaikkein selkein. Ensin raportti esittää tekstuaalisesti, että sivusto sisältää elementtejä, joiden kontrasti on liian vähäinen. Raportin Highlight-nappia klikkaamalla työkalu rajaa selkeästi lilalla värillä alueen, jossa saatavuuspuute esiintyy. Lisäksi kyseisen elementin tiedot esitetään korostettuna. Axe DevToolsin esitystapa on esitetty kuvassa 14.



Kuva 3. Axe DevTools-raportin tapa paikantaa tietty elementti, jossa kontrastipuute on havaittu.

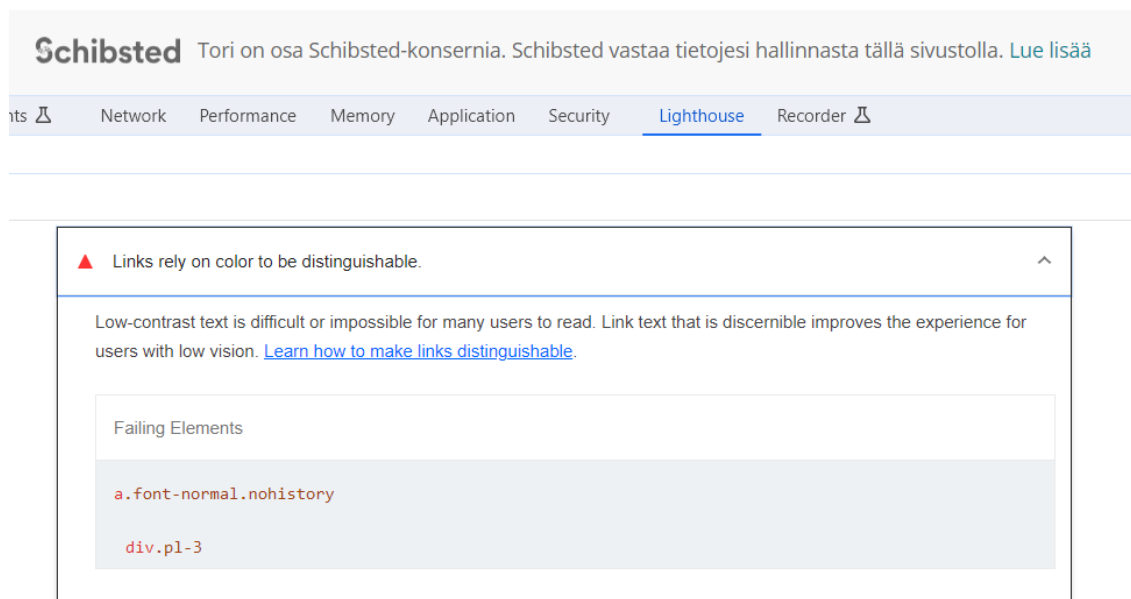
Kriteeri 1.4.1: Tiedon tai toiminnallisuuden esittämiseen, vastauksen kehittämiseen tai visuaalisen elementin erottamiseen ei tule käyttää pelkkää väriä ainoana indikaattorina (8).

Tori.fi-sivuston footer-osiosta (kuva 15) löytyi hyvä esimerkki esittämään puute tässä saavutettavuuskriteerissä. Alla olevassa kuvassa 15 on ensin sivuston omistavan konsernin logo, sitten normaalia leipätekstiä ja viimeisenä Lue lisää -linkki.

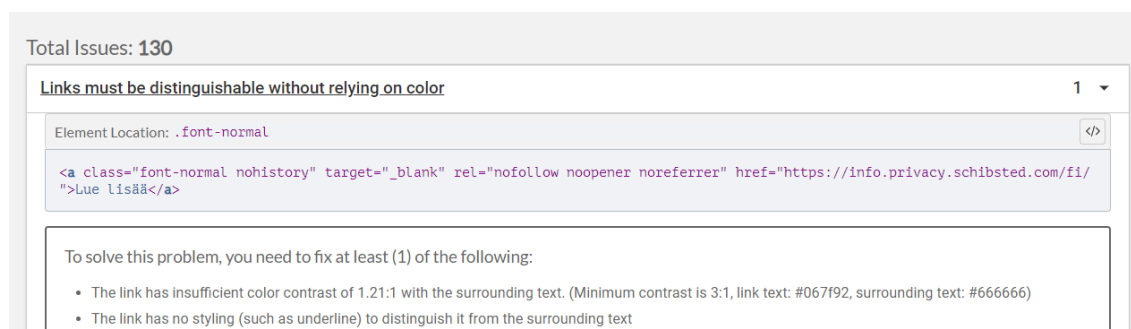


Kuva 4. Tori.fi-verkkosivuston footer.

WCAG-ohjeistuksen mukaan leipätekstin seassa olevat linkit tulee korostaa tunnistettavaksi esimerkiksi alleiviivauksella, eikä pelkkä väri riitä kuvaamaan selkeästi, että osa tekstistä on klikattava linkki. Kolmesta työkalusta Google Lighthouse sekä Axe DevTools löysivät kyseisen saavutettavuuspuutteen. Kuvista 16 ja 17 nähdään, että molemmat työkalut ilmoittivat selkeästi havainneensa puutteen linkin esitystavassa.



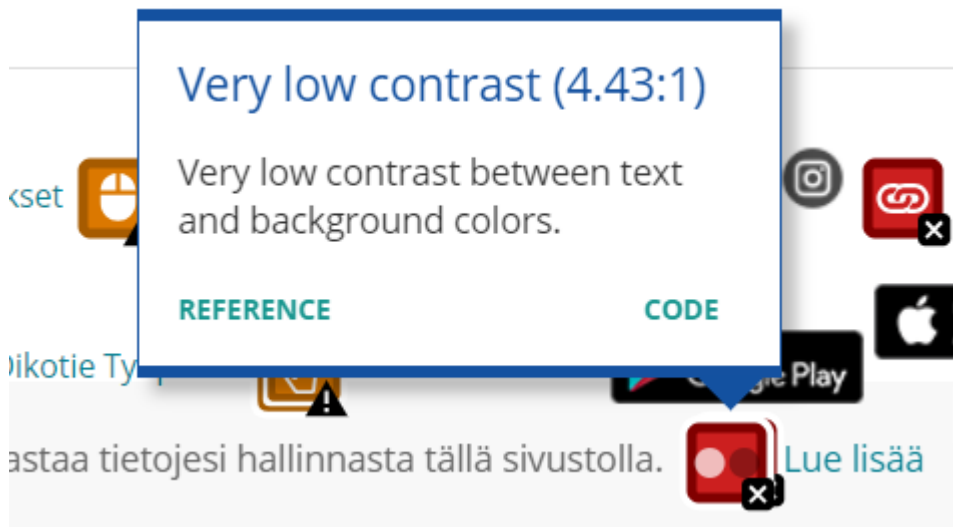
Kuva 5. Google Lighthousen raportti kriteerin 1.4.1 puutteesta.



Kuva 6. Axe DevTools-työkalun havainto kriteerin 1.4.1 puutteesta.

Toisin kuin Lighthouse ja Axe DevTools, Wave ei selkeästi ilmaissut linkkiin liittyvää puutetta. Työkalu varoitti, että tekstin ja taustan välinen kontrasti on liian vähäinen. Kuvassa 18 on esitetty Waven varoitus, joka liittyy saavutet-

tavuuskriteeriin 1.4.3. Wave ei siis löytänyt kyseistä puutetta, jonka mukaan linkki tulisi esittää muutenkin erottavasti kuin värin avulla.

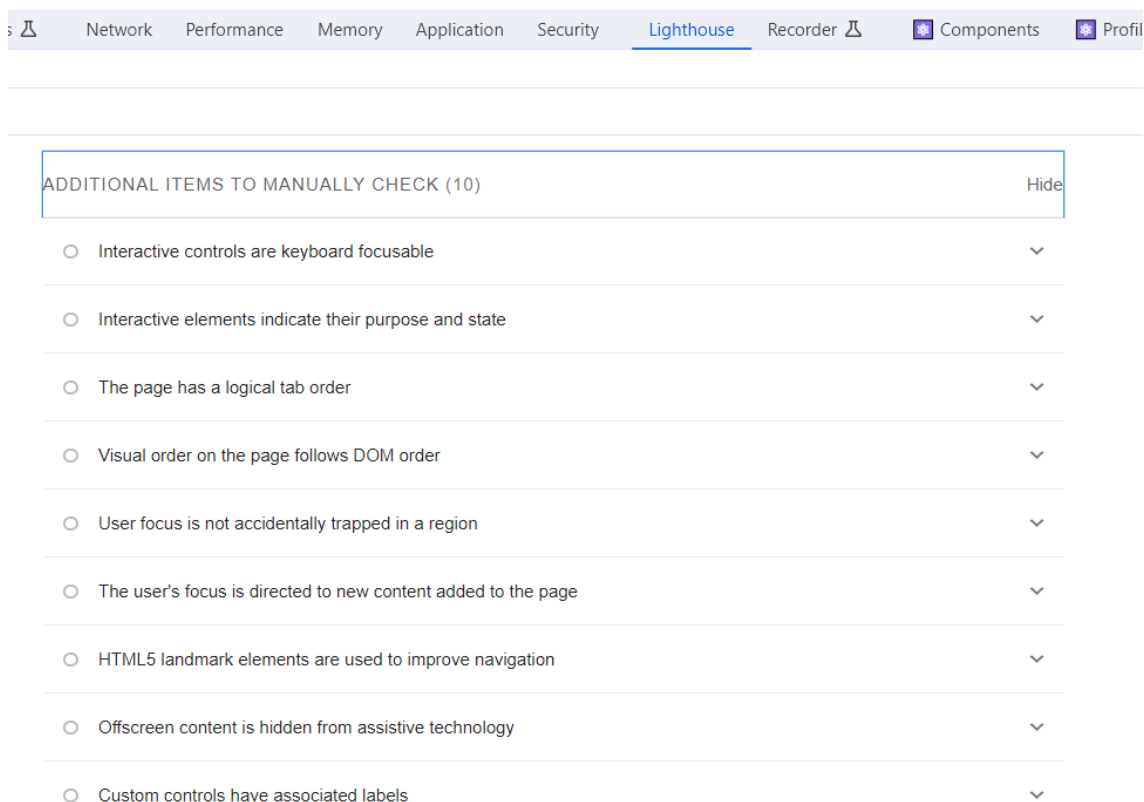


Kuva 7. Wave-työkalun varoitus liittyen linkkiin.

Kriteeri 2.1.2: Mikäli komponenttiin voidaan kohdistaa näppäimistöllä, myös poiskohdistus komponentista näppäimistöllä tulee olla mahdollinen. Mikäli poiskohdistus edellyttää muita kuin standardi poistumisnäppäinkomentoja, tulee käyttäjää ohjeistaa, miten poiskohdistus voidaan toteuttaa. (8.)

Google Chromen sovelluskehittäjä Rob Dodson kertoo, että erityisesti automaattista täyttöä tarjoavat syötekentät ovat usein paikkoja, joissa näppäimistökohdistus voi jäädä jumiin (21). Täten koetin etsiä syötekenttää, joka ei täyttäisi WCAG-kriteeriä 2.1.2.

Huomasin myös, että Google Lighthouse-työkalu listasi jokaisella analyysikerällä tietyt asiat, jotka vaativat manuaalista tarkistusta. Kuvassa 19 on esitettyinä Lighthousen generoima listaus.



Kuva 8. Lighthouse-työkalun listaus asioista, jotka vaativat manuaalista tarkistusta.

Yllä olevasta kuvasta 19 voidaan nähdä, että useampi kohta liittyy näppäimistöllä navigointiin, ja täten voidaan epäillä, ettei Lighthouse-työkalu tarjoa toiminnallisuutta, jolla näppäimistönavigointiin liittyvät puutteet löytyisivät. Lighthouse-työkalun listauksessa asioista, jotka tulisi manuaalisesti tutkia, on mainittu mm:

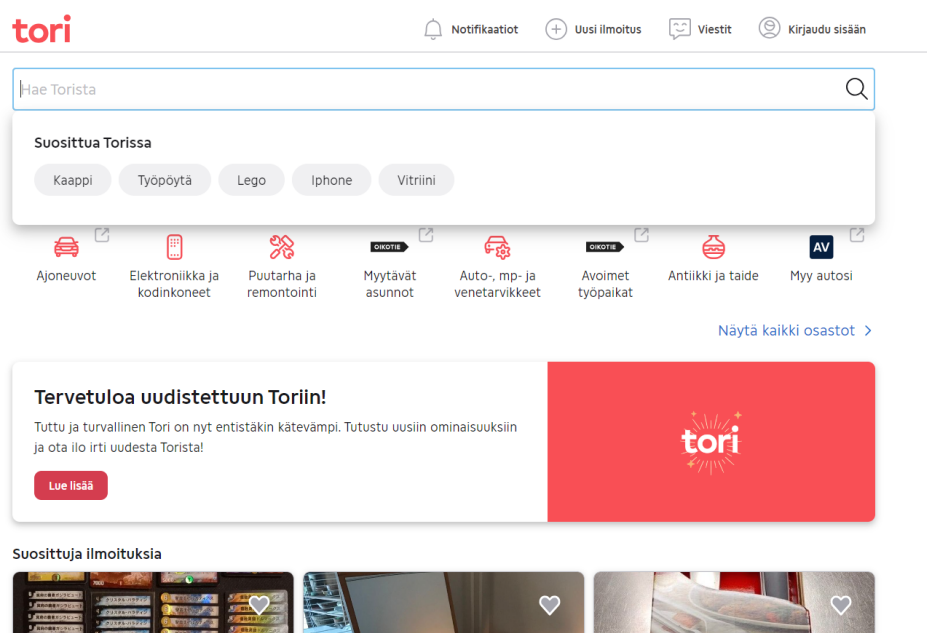
- Interaktiivisiin komponentteihin voidaan kohdistaa näppäimistöä käyttäen.
- Interaktiiviset elementit sisältävät tiedon niiden käyttötarkoituksesta sekä tilasta ja ne tulee selkeästi olla tunnistettavissa ei-toiminnallisista komponenteista.
- Sivun sarkainjärjestys on looginen eli sarkainjärjestys noudattaa visuaalista esitystapaa.
- Visuaalinen järjestys noudattaa DOM-puun järjestystä.
- Käyttäjän kohdistus ei vahingossa jää jumiin jonkin alueelle sivustolla.

- Mikäli sivulle tulee uutta sisältöä, käyttäjän kohdistus tulisi ohjata uuteen sisältöön.
- HTML5 mukaisia merkkipaaluelementtejä (landmark elements), kuten esim. <main> ja <nav>, käytetään parantamaan navigointia avustavia teknologioita käyttäen.

Vaikka Lighthouse ei tarjoa kaiken kattavaa analyysia, joka arvioisi myös näppäimistö navigointia, se kuitenkin tarjoaa listauksen asioista, jotka ovat tärkeä huomioida ja analysoida manuaalisesti.

Opinnäytetyöprosessini aikana tori.fi-sivusto koki uudistuksen, ja jatkoin beta.tori.fi-sivuston tutkimista. Pyrin löytämään kohdan, joka ei täyttäisi kriteeriä 2.1.2, jotta voisin jatkaa automaattitestaustyökalujen vertailua.

Beta.tori.fi etusivulla on heti yläpalkin jälkeen hakukenttä. Kun kohdistuksen vienkenttään, aukeaa alapuolelle modaali, joka tarjoaa suosittuja hakusanoja. Yläpalkki ja hakukenttä on esitetty kuvassa 20.



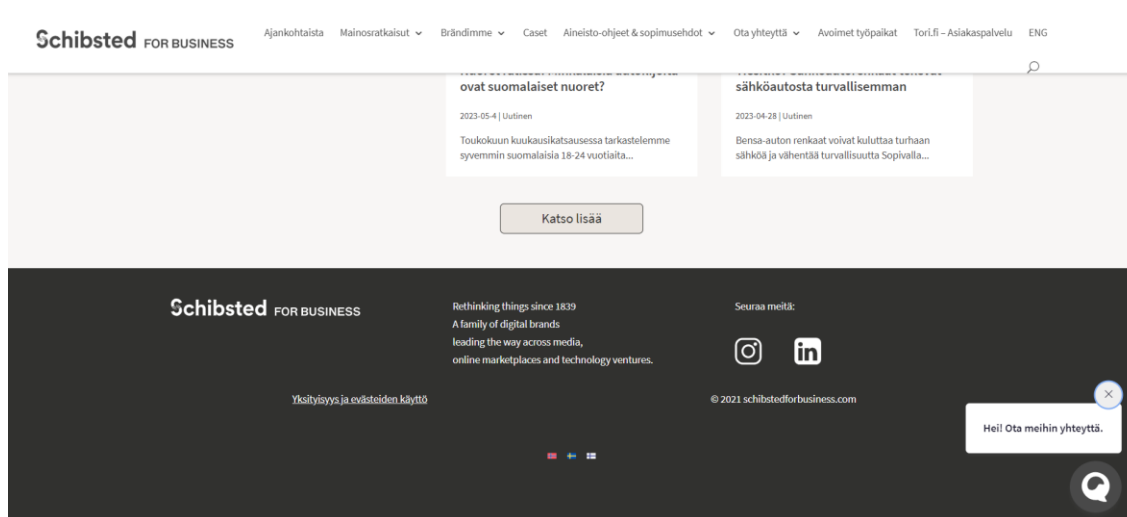
Kuva 9. Uudistetun Tori-palvelun etusivu ja sen hakukenttä.

Kokeillessani näppäimistöllä navigointia ensimmäisenä huomasin, ettei modaalin avautumiseen voi vaikuttaa eikä sitä saa enää suljettua, kun se on

avautunut. Näppäimistöllä sivulla liikkuesssa modaali jäi auki, eli se ei sulkeutunut automaattisesti, kun hakukentästä siirsi kohdistuksen muualle.

Oletuksena hakukentästä eteenpäin liikuttaessa kohdistus ei mennyt modaalin hakusanalinkkielelementteihin. Nuolinäppäimistöllä alaspäin-näppäintä klikkaamalla modaalin sisältöön pääsi kuitenkin kohdistamaan. Modaalin sisällön kohdistuksesta ei päässyt enää takaisin sivuston pääsisältöön. Kokeilin perinteistä takaisinmenomenetelmää, joka toimii pääsisällön kanssa, eli shift + tab sekä esc-näppäintä. Kumpikaan näistä ei vienyt näppäimistöllä navigoitaessa takaisin sivun muun sisällön pariin. Modaalin sisällöstä ei päässyt pois myöskään tab-näppäimellä eteenpäin siirtymällä. Sekä shift+tab että tab molemmat laukaisivat haun ja siirsivät käyttäjän siis kokonaan pois etusivulta. Ainoa tapa millä modaalista pääsi pois takaisin hakukentän kohdistukseen, oli nuolinäppäin ylöspäin. Tämän jälkeen yrittäessä jatkaa pois hakukentästä, sekä tab (eteenpäin) että shift+tab (taaksepäin) laukaisivat seuraavana järjestyksessä olevan toiminnallisuutta sisältävät elementin. Eli tämän jälkeen kohdistusnäppäimet eivät enää vain kohdistaneet, vaan toimivat Enter- ja välilyönti-näppäimien tavoin toiminnallisuuden laukaisevana valintana. Täten hakukenttä ei täyttänyt saavuuskriteeriä 2.1.2 ja päätin kokeilla, löytävätkö saavutettavuutta arvioivat testityökalut tätä puutetta.

Sivustolta löytyi myös toinen selkeä kohta, jossa kriteeri 2.1.2. ei täyty. Sivun chatbottiin pystyi kohdistamaan, mutta sen jälkeen kohdistus jäi jumiin chatbottiin eikä sieltä päässyt enää takaisin sivun pääsisältöön. Kuvassa 21 on havaittavissa tilanne, kun kohdistus on osoitettu chatbotin sulje-painikkeeseen. Päätin vielä katsoa 2.1.3 lisäksi, löytäisivätkö testitulokset tätä puutetta.

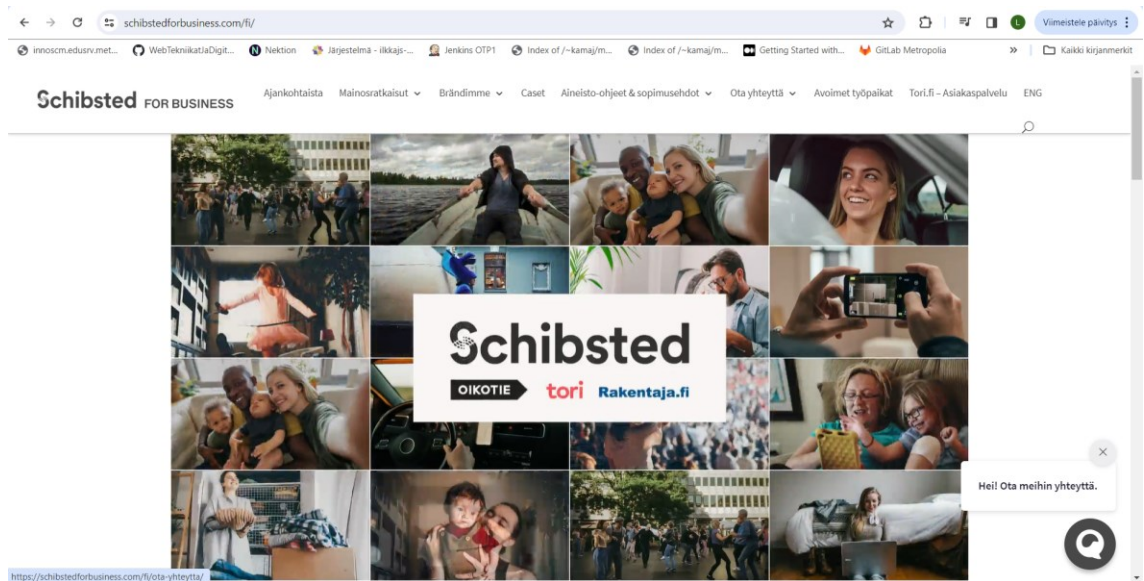


Kuva 21. Schibsted for business -sivuston chatbot.

Mikään kolmesta työkalusta ei löytänyt saavutettavuuspuutetta.

Kriteeri 2.1.3: Kaikki toiminnallisuus ja sisältö tulee olla saavutettavissa pelkällä näppäimistöllä, ilman aikarajoitteita tietyille näppäinvalinnoille (8).

Tori.fi-palvelun omistavan yrityksen Schibstedin yritysasiakkaille tarkoitettu sivusto <https://schibstedforbusiness.com/fi/> sisälsi useampia saavutettavuuspuutteita. Navigoidessa näppäimistöllä kohdistusta ei välttämättä esitetty visuaalisesti käyttöliittymässä lainkaan. Seurasin navigoinnin kohdistuksen tilannetta sivun vasemmassa alaosassa näkyvässä osiossa, joka esittää kohdistettuna olevan elementin sisältävän linkin. Päävalikon alavetovalikoihin ei päässyt näppäimistöä käyttäen kohdistamaan lainkaan. Kuvassa 22 sivuston etusivu, kun kohdistus on päävalikon kuudennessa kohdassa ”Ota yhteyttä”.



Kuva 22. Schibstedin yritysasiakkaiden etusivu.

Mikään testityökaluista ei löytänyt näitä puutteita.

3.3 Testitulokset ja päätelmät

Taulukossa 2 on esitetty alaluvussa 3.2 esitellyn testityökaluvertailun tulokset. Kullekin riville työkalukohtainen tulos on tekstin lisäksi värikoodattu: solu on vihreä, mikäli työkalu löysi ko. saavutettavuuspuutteen, ja punainen, jos työkalu ei havainnut puutetta.

Taulukko 2. Vertailun tulokset saavutettavuuskriteereittäin.

Kriteeri	Wave	Google Light-house	Axe DevTools
1.1.1 Kaikelle ei-tekstuaaliselle sisällölle tulee esittää tekstivastine.	Puute havaittu	Puute havaittu	Puute havaittu

<p>1.4.10 Sisältö voidaan esittää ilman, että mitään tietoa tai toiminnallisuutta hävitetään, ainakin 320 X 256 CSS pikselin kokoisessa näytössä siten, ettei näkymässä ole kahdensuuntaista vieritystä.</p>	Puutetta ei havaittu	Puutetta ei havaittu	Puutetta ei havaittu
<p>1.4.4 Tekstin kokoa voidaan suurentaa vähintään 200 % ilman, että sisältöä tai toiminnallisuutta menetetään. Kriteeri ei koske selite- tai kuvatekstejä.</p>	Puutetta ei havaittu	Puutetta ei havaittu	Puutetta ei havaittu
<p>1.4.3 Normaalikokoisen tekstin kontrastin tulee olla vähintään 4,5:1 ja ison tekstin vähintään 3:1.</p>	Puute havaittu	Puute havaittu	Puute havaittu
<p>1.4.1. Tiedon tai toiminnallisuuden esittämiseen, vastauksen kehittämiseen tai visuaalisen elementin erottamiseen ei tule käyttää pelkkää väriä ainoana indikaattorina.</p>	Puutetta ei havaittu	Puute havaittu	Puute havaittu
<p>2.1.2 Mikäli komponenttiin voidaan kohdistaa näppäimistöllä, myös poiskohdistus komponentista näppäimistöllä tulee olla mahdollinen. Mikäli poiskohdistus edellyttää muita kuin standardi poistumisnäppäinkomentoja, tulee käyttäjää</p>	Puutetta ei havaittu	Puutetta ei havaittu	Puutetta ei havaittu

ohjeistaa miten poiskohdistus voidaan toteuttaa.			
2.1.3 Kaikki toiminnallisuus ja sisältö tulee olla saavutettavissa pelkällä näppäimistöllä, ilman aikarajoitteita tietyille näppäinvalinnoille.	Puutetta ei havaittu	Puutetta ei havaittu	Puutetta ei havaittu

Testattuani kolmea selaimen laajennoksena toimivaa saavutettavuuden analyysityökalua, koin, että kaikki työkalut olivat helppokäyttöisiä. Mielestäni axe DevToolsin tapa esittää havainnot oli kaikista selkein. Yksilölliset elementit ja niihin liittyvät saavutettavuuspuutteet analysoitavassa sivussa oli esitetty hyvin selkeästi ja ne oli helppo yksilöidä. Axe DevTools korosti kyseessä olevan komponentin suoraan kohdesivustolla ja korostuksen viereen nostettiin esiin aina html-elementin tyyppi ja tunnisteet. Raportissa oli myös esitetty kyseisen elementin koko html-koodi kaikkine attribuutteineen.

Wave-työkalun esitystapa taas manipuloi kohdesivuston tyylittelyä tavalla, joka useassa tapauksessa muutti sivun jäsentelyä ja rakennetta, joka teki raportin tulkitsemisesta ajoittain epäselvää. Kaupallisiin vertailukumppaneihin Waven ulkoasuun fontit ja ikonit olivat ehkä hieman vanhanaikaisempia. Moderni ulkoasu voi vahvistaa käyttäjälle mielikuvaa nykyaikaisista teknologioista ja ammattimaisuudesta.

Lighthouse ei yksilöinyt välttämättä ollenkaan kohde-elementtiä, vaan raportoi vaan ylätasen komponentin, josta puutteellinen elementti löytyi. Axe DevTools oli ainoa, joka selkeästi yksilöi elementin, josta saavutettavuus-puutteita havaittiin.

Kaikkien kolmen vertailussa olleiden työkalujen testitulokset olivat hyvin samankaltaisia. Ainoastaan kriteeriä 1.4.1 vasten testatessa oli eroavaisuuksia: Wave oli ainoa työkalu, joka ei havainnut puutetta. Tätä poikkeusta lukuun ottamatta yksinkertaiset, yhtä elementtiä koskevat puutteet, työkalut löytävät parhaiten. Näitä on esimerkiksi tekstivastineiden ja vaihtoehtoisten kuvausten puute kuvissa, napeissa ja linkeissä, kontrastiongelmat ja sivuston html-rakenteen semanttisuus.

Näppäimistöllä navigointiin ja kohdistamiseen liittyviä ongelmia nämä työkalut eivät havainneet. Myös tekstin ja koko sivuston koon muutoksiin ja skaalautuvuuteen liittyviä puutteita työkalut eivät analysoineet. Työssä ei pystytty selvittämään, kuinka tarkkoja ja tehokkaita maksulliset automaattitestaustyökalut ovat ja olisivatko ne suoriutuneet vertailusta paremmin.

Vertailun perusteella koneelliset testityökalut ovat tehokkaita havaitsemaan yksinkertaisia saavutettavuusongelmia, kuten tekstivastineiden puutteita kuvissa ja kontrastiongelmia. Tämä johtunee siitä, että nämä puutteet voidaan tunnistaa suhteellisen suoraviivaisesti tarkastelemalla verkkosivuston rakennetta ja sisältöä lähdekoodista.

Sen sijaan näppäimistöllä liikkumiseen ja skaalautuvuuteen liittyvät ongelmat ovat teknisesti monimutkaisempia, ja ne voivat vaatia syvällisempää analyysia. Esimerkiksi sivun skaalautuvuus kokonaisuutena ja tekstin suurennus voivat vaatia monimutkaisten CSS-tyyliin tai JavaScript-toiminnallisuuksien oikein toteuttamista. Tämän myötä tällaisten dynaamisten kokonaisuuksien kaikkien mahdollisten eri tilojen analyysi voi olla teknisesti haastavampi toteuttaa. Ihmisen käsittämä sivuston looginen järjestys ja rakenne on myös aina riippuvainen kontekstista, eikä tätä voida arvioida koneellisesti.

Myös laaja web-ohjelmistokehityksen kehys- ja kirjastotarjonta voi lisätä saavutettavuuden arvioinnin haastavuutta. Monimutkaiset teknologiakirjastot ja kehysratkaisut voivat lisätä monimutkaisuutta sivun rakenteeseen ja toimintaan, mikä

voi heikentää automaattitestityökalujen kykyä arvioida saavutettavuutta koneellisesti.

Esimerkiksi suosittu kirjasto kuten Angular, React ja Next.js voivat dynaamisesti muuttaa sivun rakennetta ja sisältöä käyttäjän toimintojen mukaan. Tämä voi aiheuttaa haasteita automaattisille testityökaluille, jotka eivät välttämättä pysty seuraamaan ja ymmärtämään näitä muutoksia kattavasti. Lisäksi, jos sivusto käyttää erityisiä käyttöliittymäkomponentteja tai räätälöityjä elementtejä, kuten Material-UI (MUI) tai muut kirjastot, nämä voivat tarjota saavutettavuuteen liittyviä ominaisuuksia, jotka saattavat olla vaikeita arvioida automaattisesti ilman syvällistä ymmärrystä käytetyistä komponenteista.

Tämä johtuu siitä, että saavutettavuuden varmistaminen näppäimistöllä navigoinnissa ja skaalautuvuudessa vaatii usein tarkempaa tietoa sivuston interaktiivisista toiminnoista ja visuaalisista asetuksista, jota automaattiset analyysityökalut eivät välttämättä pysty tunnistamaan. Lisäksi nämä ongelmat voivat olla hyvin kontekstiriippuvaisia ja vaatia manuaalista tarkastelua ja testausta varmistukseen täydellisen saavutettavuuden.

Verraten luvussa 3.1 esitettyihin kunkin työkalun profiileihin, vertailun tuloksissa yllättää se, ettei Lighthouse hävinnyt vertailussa saavutettavuusarviointiin erikoistuneille palveluille axe DevTools ja Wave. Kaupallisten toimijoiden, Dequen ja Googlen resurssit kaupallisiin toimijoihin myös näkyivät käyttäjäystävällisyydessä ja ulkoasussa.

Verkkopalveluiden käyttöliittymäkehityksessä tulee olla useita tapoja tehdä palvelusta saavutettava niin heikkonäköisille, värisokeille, muuten näkörajoitteisille, avusteisia teknologioita käyttäville henkilöille kuin näppäimistöllä navigoiville käyttäjille. Sovelluskehittäjien, palvelumuotoilijoiden ja käyttöliittymäsuunnittelijoiden on tärkeä tuntee käsite saavutettavuus, sen mittarit ja hyvät käytännöt. Täten saavutettavuus huomioidaan jo suunnitteluvaiheessa sekä varhain toteutusvaiheessa. Saavutettavuutta arvioidessa ja kehitettäessä voi automaattitestityökalut olla hyvä lisä, mutta ne eivät ole yksinään riittäviä arviointiin. Käyt-

töliittymän robottitestauksella voidaan myös tehokkaasti ehkäistä regressiota ja esimerkiksi varmistaa teksivastineiden ja vaihtoehtoisten tekstien esiintyminen. Manuaalitestaus eri käyttäjäryhmien näkökulmasta on kuitenkin aina hyvin olennainen osa saavutettavuuden arviointia, eikä tällä hetkellä tarjolla olevat automaattitesti- ja analyysityökalut voi korvata sitä.

4 Yhteenveto

Sovelluskehityksessä saavutettavuus on keskeinen osa, joka vähentää käyttäjien eriarvoisuutta ja parantaa digipalveluiden laatua. EU-jäsenvaltioissa saavutettavuusdirektiivi velvoittaa julkisen sektorin huolehtimaan digipalveluidensa saavutettavuudesta. Sovelluskehittäjien, palvelumuotoilijoiden ja käyttöliittymäsuunnittelijoiden tulee ymmärtää saavutettavuuden käsitteet, mittaristot ja hyvät käytännöt varmistaakseen kehittämiensä palveluiden saavutettavuuden.

Saavutettavuuden automaattitestauksella pyritään tehostamaan verkkopalveluiden saavutettavuuden arviointia ja tunnistamaan teknisiä toteutukseen liittyviä haasteita varhaisessa vaiheessa. Opinnäytetyön tavoitteena oli tutkia kolmea erilaista automaattitestauksen työkalua ja näiden kykyä havaita saavutettavuuspuutteita. Opinnäytetyöprosessin aikana tutustuttiin WCAG-kriteereihin ja useiden lähteiden esittämiin yleisiin saavutettavuuspuutteisiin, jotka ohjasivat seitsemän keskeisen onnistumiskriteerin valintaan. Työssä etsittiin valittuja WCAG-onnistumiskriteereitä vastaava saavutettavuuspuute, jota vasten automaattitestaustyökaluja vertailtiin.

Kolme valittua testaustyökalua olivat Wave Evaluation Tool, Lighthouse ja Axe-DevTools, jotka toimivat selainlaajennoksina. Työkalujen vertailussa kävi ilmi, että ne olivat kaikki helppokäyttöisiä, mutta Axe DevTools erottui selkeänä ja tehokkaana vaihtoehtona. Testitulokset olivat yhteneväisiä suurimmassa osassa saavutettavuuskriteereistä, mutta tietyissä kohdissa työkalujen välillä oli eroja. Esimerkiksi tekstin ja sivuston koon muutoksiin sekä näppäimistön käyttöön liittyviä puutteita työkalut eivät analysoineet.

Vaikka automaattitestityökalut voivat olla hyödyllisiä saavutettavuuden arvioinnissa, manuaalitestaus eri käyttäjäryhmien näkökulmasta on edelleen välttämätöntä. Yksinkertaisten, teknisen toteutuksen saavutettavuuspuutteiden havainnointiin automaattitestaustyökalut voivat lisätä nopeutta ja tehokkuutta. Saavutettavuus on kuitenkin niin laaja ja kompleksinen käsite, ettei koneellinen testaus tämänhetkisin työkaluilla korvaa ihmisen tekemää saavutettavuusarviointia.

Lähteet

- 1 Selovuo, Kari. 2019. Saavutettavuusopas. Seloit.
- 2 Kenelle saavutettavuus on tärkeää? Verkkoaineisto. Etelä-Suomen aluehallintovirasto. <<https://www.saavutettavuusvaatimukset.fi/yleista-saavutettavuudesta/kenelle-saavutettavuus-on-tarkeaa/>>. Luettu 12.12.2023.
- 3 Digital society statistics at regional level. 2023. Verkkoaineisto. Eurostat. <https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Digital_society_statistics_at_regional_level&oldid=610768#Internet_users/>. Päivitetty 11.03.2024. Luettu 12.12.2023.
- 4 96% of young people in the EU uses the internet daily. 2023. Verkkoaineisto. Eurostat. <<https://ec.europa.eu/eurostat/web/products-eurostat-news/w/ddn-20230714-1/>>. Päivitetty 11.7.2023. Luettu 12.12.2023.
- 5 Alinikula, Petteri; Harlund, Tony; Lindroth, Matias & Tamminen Terhi. 2017. Kuntien saavutettavuusopas – Liite 1: Arvio vammaisten henkilöiden määrästä Suomessa ja heidän erityistarpeistaan. Verkkoaineisto. Kuntaliitto. <<https://www.kuntaliitto.fi/julkaisut/saavutettavuusopas/liite-1#/>> Luettu 12.12.2023.
- 6 Euroopan parlamentin ja neuvoston direktiivi julkisen sektorin elinten verkkosivustojen ja mobiilisovellusten saavutettavuudesta. 2016. Direktiivi 2016/2102/EU. Verkkoaineisto. Euroopan union virallinen lehti 2.12.2016. <<https://eur-lex.europa.eu/legal-content/FI/TXT/HTML/?uri=CELEX:32016L2102&from=FI/>>. Luettu 14.12.2023.
- 7 Ollikainen, Vesa. 2021. Käyttäjäkeskeinen suunnittelu-kurssi – Saavutettavuus-luentokalvot.
- 8 Web Content Accessibility Guidelines (WCAG) 2.1. 2023. Verkkoaineisto. The World Wide Web Consortium (W3C). <<https://www.w3.org/TR/WCAG21/>>. Luettu 14.12.2023.
- 9 Pickering, Heydon. 2016. Inclusive Design Patterns – Coding Accessibility Into Web Design. Smashing Magazine.
- 10 Kumar, Ashutosh. 8.3.2021. How Modern Web Applications Are Made Today – How frameworks and technologies go into building a modern-day web app. Verkkoaineisto. Geek Culture, Medium. <<https://medium.com/geekculture/how-modern-web-applications-are-made-today-514ff5fc8506/>>. Luettu 18.12.2023.

- 11 JavaScript HTML DOM. Verkkoaineisto. W3Schools. <https://www.w3schools.com/js/js_htmlDOM.asp/>. Luettu 18.12.2023.
- 12 Alinikula, Petteri; Harlund, Tony; Lindroth, Matias & Tamminen Terhi. 2017. Kuntien saavutettavuusopas. Verkkoaineisto. Kuntaliitto. <<https://www.kuntaliitto.fi/julkaisut/saavutettavuusopas/11-muut-vaalimukset/>>. Luettu 20.12.2023.
- 13 Full Stack Open. Verkkoaineisto. Helsingin Yliopisto; Houston Inc. <https://fullstackopen.com/osa1/reactin_alkeet/>. Luettu 18.12.2023.
- 14 Ohjeita suunnittelun tueksi. Verkkoaineisto. Etelä-Suomen aluehallintovirasto. <<https://www.saavutettavuusvaatimukset.fi/yleista-saavutettavuudesta/ohjeita-suunnittelun-tueksi/>>. Luettu 12.12.2023.
- 15 Kuva ja ääni. Verkkoaineisto. Saavutettavuuskirjasto Celia. <<https://www.saavutettavasti.fi/kuva-ja-aani/>>. Päivitetty 20.12.2023. Luettu 20.12.2023.
- 16 Jokisuu, Petra. 2022. Verkkopohjaisten karttojen saavutettavuushaasteet. Pro gradu-tutkielma. Jyväskylän Yliopisto. Theseus-tietokanta.
- 17 Web Accessibility Evaluation Tools List. Verkkoaineisto. The World Wide Web Consortium (W3C). <<https://www.w3.org/WAI/test-evaluate/tools/list/>>. Luettu 15.01.2024.
- 18 WAVE Web Accessibility Evaluation Tools. Verkkoaineisto. WebAIM; Institute for Disability Research, Policy & Practice, Utah State University. <<https://wave.webaim.org/>>. Luettu 02.02.2024.
- 19 axe Devtools ®: Digital accessibility testing tools dev teams love. Verkkoaineisto. Deque. <<https://www.deque.com/axe/devtools/>>. Luettu 02.02.2024.
- 20 Lighthouse. Verkkoaineisto. Google. <<https://developer.chrome.com/docs/lighthouse/>>. Luettu 02.02.2024.
- 21 Dodson, Rob. 2017. How to review accessibility. Verkkoaineisto. Web.dev. <https://web.dev/articles/how-to-review?utm_source=lighthouse&utm_medium=devtools/>. Luettu 02.02.2024.