



Verkkosivustojen saavutettavuuden testaustyökalujen vertailu ja arviointi

Laura Heikkinen

Lin Lam

Haaga-Helia ammattikorkeakoulu

Tradenomi, tietojenkäsittely

Opinnäytetyö

2023

Tiivistelmä

Tekijät Laura Heikkinen, Lin Lam
Tutkinto Tradenomi, tietojenkäsittely
Raportin/Opinnäytetyön nimi Verkkosivustojen saavutettavuuden testaustyökalujen vertailu ja arviointi
Sivu- ja liitesivumäärä 72 + 29
<p>Saavutettavuus on yhä ajankohtaisempi teema digitaalisissa palveluissa. Saavutettavuuden avulla kaikki käyttäjät voivat hyödyntää digitaalisia palveluita yhdenvertaisesti, riippumatta esimerkiksi heidän kyvyistään tai käytetyistä laitteista. Myös lainsäädäntöä on viety suuntaan, jossa julkisia organisaatioita ja yrityksiä veloitetaan noudattamaan saavutettavuusvaatimuksia verkkosivustoilla. Nämä saavutettavuusvaatimukset liittyvät kansainväliseen Web Content Accessibility Guidelines 2.1 (WCAG) -ohjeistukseen, johon perustuen myös yleisimmät saavutettavuustyökalut tutkivat verkkosivustojen saavutettavuutta.</p> <p>Opinnäytetyössä tutkittiin joitakin yleisimpiä verkkosivustojen saavutettavuuden arviointiin käytettäviä, käyttäjälle maksutta saatavilla olevia testaustyökaluja ja vertailtiin niitä keskenään. Taroituksena oli saada vastaus kysymykseen, kuinka hyvin saavutettavuuden testaamiseen tarkoitetut automaattiset työkalut pystyvät tunnistamaan saavutettavuuteen liittyviä ongelmia verkkosivustolla.</p> <p>Työkalujen vertailun lisäksi työssä pyrittiin selvittämään, minkälaisia asioita liittyy saavutettavuuden käsitteeseen, sen arviointiin ja testaamiseen. Tietoperustassa pyrittiin käsittelemään kattavasti digitaalisten palveluiden saavutettavuutta, verkkosisällön saavutettavuusvaatimuksia (WCAG), käyttäjäkokemuksen ja saavutettavuuden suhdetta sekä saavutettavien verkkosivustojen toteutusta ja testaamista.</p> <p>Työkalujen vertailussa käytettiin tutkimusmenetelmänä vertailuanalyysiä. Vertailuun valitut työkalut valittiin tietyin valintakriteerein ja vertailuanalyysin prosessiaskeleita sovellettiin opinnäytetyöhön sopivaksi. Vertailuanalyysin prosessiaskelien perusteella suoritettiin valittujen työkalujen testaus ja arviointi. Työn tuloksena syntyi yhteenveto tutkimuksen kohteina olleista työkaluista.</p> <p>Opinnäytetyön tulokseksi saatiin, että kohteena olleet automaattiset testaustyökalut ottivat esiin ja painottivat testiraporteissaan erilaisia asioita, sekä raportoivat testien tuloksia erilaisin tavoin. Nämä työkalujen erilaiset tavat raportoida testiensä tuloksia vaikuttivat eniten tulkintaan siitä, miten hyvänä voitiin pitää kunkin työkalun kykyä arvioida saavutettavuutta. Tulosten perusteella muodostettiin johtopäätös, jossa Google Lighthouse nousi työkaluista parhaimmaksi verkkosivuston saavutettavuuden arvioinnissa.</p> <p>Tutkimuksen ja tulosten luotettavuuteen pyrittiin ottamaan kantaa kattavasti tämän työn lopussa, sekä esittämään ehdotuksia jatkokehittämistä varten. Jatkokehitysideaksi opinnäytetyön pohjalta nousi erityisesti testausytimiin perehtyminen tarkemmin, jolloin saavutettavuustyökalujen toiminnasta voitaisiin muodostaa entistä kattavampi käsitys.</p>
Asiasanat saavutettavuus, www-sivut, saavutettavuusvaatimukset, WCAG, vertailu, testaus

Sisällys

1	Johdanto	1
1.1	Keskeiset käsitteet	2
2	Saavutettavuus ja käyttäjäkokemus	4
2.1	Digitaalisten palveluiden saavutettavuus	4
2.2	Lainsäädäntö	5
2.3	Verkkosisällön saavutettavuusohjeet (WCAG)	6
2.4	Saavutettavuuden suhde käyttäjäkokemukseen	10
2.5	Käyttäjäkokemuksen suunnittelu	10
2.5.1	Käyttäjäkeskeinen suunnittelu	11
2.6	Hyvän käyttäjäkokemuksen muodostuminen	12
2.6.1	Morvillen hunajakennomalli	14
3	Saavutettavien verkkosivustojen toteutus	16
3.1	HTML	16
3.2	CSS ja JavaScript	18
3.3	Saavutettavan sivuston rakenne	19
3.4	Saavutettavuuden testaaminen	22
3.4.1	ACT-testaussäännöt	23
3.4.2	Manuaalinen testaus	24
3.4.3	Käyttäjätestaus	25
3.4.4	Automaattinen testaus	25
4	Saavutettavuustyökalujen testaus	29
4.1	Vertailuanalyysi tutkimusmenetelmänä	29
4.2	Tutkimuksen luotettavuuden varmistaminen	31
4.3	Vertailuun valitut testaustyökalut	32
4.4	Testaustyökalujen testaus	34
4.4.1	Web Accessibility Evaluation Tool - WAVE	35
4.4.2	Axe DevTools	39
4.4.3	Accessibility Insights for Web	42
4.4.4	Access Assistant	45
4.4.5	Google Lighthouse	48
4.4.6	Salvia	54
4.5	Vertailuanalyysin tulokset	60
5	Pohdinta	63
	Lähteet	66

Liitteet	73
Liite 1. WAVE-työkalun testaus	73
Liite 2. Axe DevTools -työkalun testaus	77
Liite 3. Accessibility Insights for Web -työkalun testaus	79
Liite 4. Access Assistant -työkalun testaus	84
Liite 5. Google Lighthouse -työkalun testaus	86
Liite 6. Salvia-työkalun testaus	93

1 Johdanto

Saavutettavuus on tärkeä tekijä nykyaikaisilla verkkosivustoilla, sillä yhä useammat ihmiset käyttävät internetiä erilaisilla laitteilla ja teknologioilla. Verkkosivustojen saavutettavuus tarkoittaa sitä, että sivustot on suunniteltu ja toteutettu siten, että ne ovat helppokäyttöisiä ja käytettävissä kaikille käyttäjille, myös niille, joilla on esimerkiksi näkö- tai kuulovamma tai jotka käyttävät apuvälineitä sivuston käyttöön. Verkkosivustojen saavutettavuus on myös tärkeä teema lainsäädännön kannalta, sillä monet maat ovat säätäneet lakeja ja asetuksia, jotka velvoittavat yrityksiä ja julkisia organisaatioita noudattamaan saavutettavuusvaatimuksia verkkosivustoillaan.

Saavutettavuuden merkitys korostuu erityisesti yritysten ja julkisen sektorin verkkosivustoilla, jotka ovat tärkeitä palvelukanavia yhteiskunnassamme. Saavutettavat verkkosivustot edistävät yhdenvertaisuutta ja tasa-arvoa ja mahdollistavat kaikkien ihmisten osallistumisen yhteiskuntaan ja sen palveluihin. Esimerkiksi Helsingin kaupunki aloitti vuonna 2019 digitalisaatio-ohjelman, jonka avulla parannetaan palveluja ja uudistetaan toimintoja monin tavoin. Ohjelman tavoitteena on tehdä digitalisaatiolla Helsingistä maailman toimivin kaupunki. Toimivuudella kaupunki haluaa hyödyntää digitalisaation tuomat mahdollisuudet, jolloin helpottuu kaupunkilaisten, yritysten, yhteisöjen, matkailijoiden, työntekijöiden ja päättäjien elämä. (Helsingin kaupunki 2022.)

Opinnäytetyömme tarkoituksena on tarkastella joitakin yleisimpiä verkkosivustojen saavutettavuuden arviointiin käytettäviä, käyttäjälle maksutta saatavilla olevia testaustyökaluja ja vertailla niitä keskenään. Tavoitteenamme on selvittää, miten nämä eri testaustyökalut auttavat sivustojen saavutettavuuden parantamisessa ja arvioimisessa, ja millaisia etuja työkaluilla on suhteessa toisiinsa. Pääkysymyksenämme onkin: kuinka hyvin saavutettavuuden testaamiseen tarkoitetut automaattiset työkalut pystyvät tunnistamaan saavutettavuuteen liittyviä ongelmia verkkosivustolla?

Jotta voisimme saada vastauksen pääkysymykseemme, käsittelemme tietoperustassamme myöskin sitä, mitä verkkosivujen saavutettavuudella tarkoitetaan, miten saavutettavuutta voidaan arvioida ja mitata, sekä mitä työkaluja on saatavilla saavutettavuuden testaamiseen. Käymme läpi opinnäytetyömme tietoperustassa myös saavutettavuuden merkitystä käyttäjäkokemuksessa ja käyttäjäkokemuksen suunnittelussa, sekä saavutettavien verkkosivustojen rakennetta. Työmme pyrkii lisäämään yleistä tietoisuutta saavutettavuudesta ja sen merkityksestä verkkosivustojen käyttäjäkokemuksessa, suunnittelussa ja kehittämisessä.

Käytämme opinnäytetyössämme sekä teoreettisia, että käytännön näkökulmia. Teoreettinen osa käsittelee saavutettavuutta ja sen merkitystä verkkosivujen suunnittelussa ja käytössä. Saavutettavuuteen liittyy läheisesti ja oleellisesti käyttäjäkokemus, joten keskitymme teoriapohjassamme paljon myös käyttäjäkokemukseen, sen suunnitteluun ja arviointiin. Käytännön osassa testaamme

erilaisia saavutettavuuden testaustyökaluja ja arvioimme niiden tehokkuutta suhteessa tietoperustaan. Lopuksi esitämme työssämme yhteenvedon siitä, millaisia ominaisuuksia yleisimmät saavutettavuuden testaustyökalut sisältävät, miten ne eroavat toisistaan ja miten ne ottavat huomioon saavutettavuuden tietoperustassamme esitettävien määritteiden osalta.

1.1 Keskeiset käsitteet

Opinnäytetyössämme toistuu usein monia käsitteitä tai termejä, kuten saavutettavuus, esteettömyys, käyttäjä, käytettävyys, käyttöliittymä ja käyttäjäkokemus. Nämä kaikki liittyvät olennaisesti saavutettaviin verkkosivustoihin, joten esittelemme nämä käsitteet tarkemmin.

Saavutettavuus

Saavutettavuus tarkoittaa, että kaikilla ihmisillä tulee olla yhtäläinen mahdollisuus käyttää ja hyödyntää erilaisia palveluita, tuotteita tai informaatiota riippumatta henkilön fyysisistä, kognitiivisista tai muista rajoitteista (Kehitysvammaliitto ry 2023a).

Esteettömyys

Esteettömyydellä viitataan pääsääntöisesti fyysiseen ympäristöön, kuten rakennuksiin, ulkoalueisiin ja julkisiin liikennevälineisiin. Esteettömyyden suunnittelussa otetaan huomioon mahdolliset esteet, jotka voivat haitata ihmisten liikkumista ja tuotteiden käyttöä. (Invalidiliitto ry s.a.; Terveiden ja hyvinvoinnin laitos 2022.)

Käyttäjä

Käyttäjällä tarkoitetaan henkilöä, joka on vuorovaikutuksessa järjestelmän, tuotteen tai palvelun kanssa (ISO 9241–210:2010).

Käyttöliittymä

Käyttöliittymä (engl. User Interface, UI), on ohjelman tai laitteen osa, jonka kautta käyttäjä seuraa ja ohjaa ohjelman tai laitteen toimintaa, ja saa siitä tietoa. Käyttöliittymä voi olla joko merkkipohjainen, tai graafinen. Graafisen käyttöliittymän kautta sovelluksella voidaan suorittaa toimintoja esimerkiksi erilaisia kuvakkeita painamalla. (Dumont 2021; Sanastokeskus ry s.a.)

Verkkosivustojen kohdalla käyttöliittymällä viitataan visuaalisten elementtien kokonaisuuteen, kuten esimerkiksi typografiaan, väreihin ja valikkopalkkeihin sekä muihin toiminnallisuuksiin, joissa ihmiset ovat vuorovaikutuksessa palvelun kanssa (Dumont 2021; Viljanen 22.2.2020).

Käyttäjäkokemus

Käyttäjäkokemus (engl. User Experience, UX) on moniulotteinen käsite, jolle ei ole olemassa vain yhtä määritelmää. Yleisesti käyttäjäkokemuksen katsotaan koostuvan monista eri elementeistä, kuten käytettävyydestä, toiminnallisuudesta, ulkoasusta, käyttäjän tarpeiden ja odotusten täyttämisestä ja käyttäjän tunnekokemuksesta. (Roto, Law, Vermeeren & Hoonhout 2011, 6.)

Yksi tunnetuimmista käyttäjäkokemuksen määritelmistä on ISO-standardin määritelmä, jonka mukaan käyttäjäkokemuksella tarkoitetaan kaikkia käyttäjän tekemiä havaintoja sekä reaktioita, jotka tulevat järjestelmän, tuotteen tai palvelun käytöstä tai ennakoidusta käytöstä ja käytön jälkeen (ISO 9241–210:2010 kohta 2.15).

Avustava teknologia

Avustava teknologia tarkoittaa verkkoympäristössä käytettyä ohjelmaa tai laitetta, jonka avulla toimintarajoitteiset henkilöt voivat käyttää digitaalista palvelua. Näitä ohjelmia voivat olla valmiiksi olevuksina selaimissa tai laitteissa olevat ohjelmat, tai tietokoneeseen asennettavat ohjelmat, kuten ruudunlukuoohjelma. (Aluehallintovirasto s.a.a.)

2 Saavutettavuus ja käyttäjäkokemus

Yhdistyneiden Kansakuntien vammaisten henkilöiden oikeuksia koskevassa yleissopimuksessa tunnustetaan, että on perusihmisoikeus saada yhdenvertainen pääsy esimerkiksi tiedottamiseen ja viestintään, muun muassa tieto- ja viestintäteknologiaan ja -järjestelmiin (W3C 2018a; Valtioneuvoston asetus vammaisten henkilöiden oikeuksista tehdyn yleissopimuksen ja sen valinnaisen pöytäkirjan voimaansaattamisesta sekä yleissopimuksen ja sen valinnaisen pöytäkirjan lainsäädännön alaan kuuluvien määräysten voimaansaattamisesta annetun lain voimaantulosta 398/2016). Siten myös verkko, ja siihen pohjautuvat palvelut ovat lähtökohtaisesti tarkoitettu kaikkien käyttäjien saataville. Tämä tarkoittaa sitä, että käyttäjän sijainnista, käyttämästä laitteesta tai ohjelmistosta, kielestä tai toimintakyvystä huolimatta verkon tulee olla kaikkien käytettävissä. (W3C 2018a.)

Suomessa yli miljoonalla henkilöllä on vaikeuksia käyttää digitaalisia palveluja, jos palvelujen saavutettavuutta ei ole huomioitu. Saavutettavuus on välttämätöntä ihmisille, joilla on jokin vamma, toimintarajoite, heikko suomen kielen taito tai he eivät ole tottuneet käyttämään digitaalisia palveluita. Esimerkiksi kehitysvammaisella tai maahanmuuttajataustaisella henkilöllä voi olla haasteita ymmärtää vaikeita termejä tai kuulovammainen henkilö tarvitsee videoihin tekstitykset. Myös ihmiset, joilla ei ole kuulorajoitteita voivat hyötyä videoiden tekstityksistä. (Aluehallintovirasto s.a.b; Kehitysvammaliitto ry 21.2.2023a.)

Saavutettavuus riippuu tilanteesta ja käyttäjästä, mutta saavutettavuudesta hyötyvät kaikki ihmiset. Panostamalla saavutettavuuteen voidaan lisätä yhdenvertaisuutta ja mahdollistaa itsenäinen asiointi digitaalisissa palveluissa yhä useammalle ihmiselle. (Aluehallintovirasto s.a.b; Kehitysvammaliitto ry 21.2.2023a.)

Esteettömyyttä ja saavutettavuutta käytetään usein rinnakkaisina termeinä, mutta ne tarkoittavat hieman eri asioita. Esteettömyys on yksi keino saavuttaa saavutettavuus. Esteettömyydellä voidaan taata se, että ympäristö tai tuote on mahdollisimman käyttäjäystävällinen, kun taas saavutettavuudella varmistetaan, että kaikki ihmiset voivat käyttää sitä yhdenvertaisesti. Esimerkiksi esteettömästi suunniteltu rakennus mahdollistaa ihmisten esteettömän liikkumisen, mutta saavutettavuus takaa sen, että myös esimerkiksi näkö- tai kuulovammaiset henkilöt voivat käyttää rakennuksen palveluita yhdenvertaisesti. (Terveystieteiden tutkimuskeskus 2022.)

2.1 Digitaalisten palveluiden saavutettavuus

Saavutettavuudella tarkoitetaan monien erilaisten ihmisten mahdollisuutta käyttää verkkosivuja ja mobiilisovelluksia mahdollisimman helposti. Saavutettavuudella huomioidaan ihmisten erilaisuudet

ja moninaisuudet verkkosivujen ja mobiilisovellusten suunnittelussa ja toteutuksessa. Erilaiset ihmiset voivat käyttää itsenäisesti digitaalisia palveluita, jos saavutettavuus on huomioitu hyvin. (Aluehallintovirasto s.a.c.)

Digitaalisen palvelun saavutettavuus jakautuu kolmeen osa-alueeseen. Näitä ovat tekninen toteutus, helppokäyttöisyys ja sisältöjen selkeys ja ymmärrettävyys. Teknisellä saavutettavuudella tarkoitetaan virheetöntä, loogista lähdekoodia sekä HTML-standardin noudattamista, WCAG-ohjeistuksen noudattamista ja palvelun toimivuuden takaamista erilaisilla laitteilla ja avustavilla teknologioilla, kuten ruudunlukuohjelmalla ja puheohjauksella. Käyttäjällä tulisi myös olla mahdollisuus muokata verkkosivuston sisällön ulkoasua tarpeidensa mukaisesti, esimerkiksi tekstin kokoa suurentamalla tai väriä vaihtamalla. (Aluehallintovirasto s.a.c; Celia 2022a.)

Helppokäyttöisyydellä tarkoitetaan digitaalisen palvelun helppoa hahmottamista, navigaatioiden selkeyttä ja käyttäjän mahdollisuutta löytää etsitty sisältö helposti. Navigaation ei pitäisi olla monitasoinen ja sivujen nimet pitäisivät olla kuvaavia ja selkeitä. Pääsisällön pitäisi myös erottua selkeästi sivun muista elementeistä ja halutun toiminnon suorittaminen olla palvelussa vaivatonta. (Aluehallintovirasto s.a.c.)

Ymmärrettävyydellä tarkoitetaan, että sisällössä on käytetty selkeää ja ymmärrettävää kieltä, ja selkokieltä. Tekstin pitäisi olla helppolukuista, helposti hahmotettavaa ja hyvin jäsenneltä lyhyiksi kappaleiksi ja luetteloiksi. Kuvaavia otsikoita pitäisi myös olla käytetty riittävästi. Ymmärrettävyys tarkoittaa myös monikanavaisuutta, eli sisältöjä tarjotaan eri muotoisina, kuten videoina, kuvina ja äänenä tekstin lisäksi. (Aluehallintovirasto s.a.c.)

2.2 Lainsäädäntö

Vuonna 2019 tuli voimaan laki digitaalisten palveluiden tarjoamisesta eli digipalvelulaki, jonka tavoitteena on saada saavutettavuus ymmärrettäväksi keskeisenä toiminnan periaatteena, joka pitäisi ottaa huomioon kaikessa verkkosisältöjen ja -palvelujen suunnittelussa, toteutuksessa ja ylläpidossa. Laki velvoittaa julkista sektoria noudattamaan saavutettavuusvaatimuksia ja samalla laki turvaa käyttäjien oikeuden hyvään digitaaliseen palveluun. Digipalvelulaki ei takaa täydellistä saavutettavuutta, mutta sitä noudattamalla voidaan parantaa verkkosivustolle pääsyä erilaisilla laitteilla. (Aluehallintovirasto 27.1.2020; Aluehallintovirasto s.a.d.)

Aluehallintoviraston (s.a.d) mukaan digipalvelulaki sisältää kolme vaatimusta, jotka verkkosivustojen ja mobiilisovellusten tulisi täyttää:

- Digitaalisen palvelun ja sen sisältöjen tulee täyttää saavutettavuusvaatimukset eli verkkosisällön saavutettavuusohjeistuksen (WCAG) A- ja AA-tason kriteerit.
- Saavutettavuusselosteessa täytyy olla saavutettavuusvaatimuksen noudattamisen arviointi sekä saavutettavuuden mahdolliset puutteet.
- Sähköinen palautekanava on saatavilla saavutettavuuspalautteen jättämiselle.

Digipalvelulaki ei takaa, että digitaalinen palvelu olisi kaikille saavutettava tai helppokäyttöinen. Lain noudattamisella voidaan kuitenkin parantaa verkkosivustojen ja mobiilisovellusten yhteensopivuutta eri laitteilla ja avustavilla teknologioilla. Digipalvelulain lisäksi on muita lakeja, jotka velvoittavat organisaatioita saavutettavaan toteutukseen, kuten perustus- ja yhdenvertaisuuslaki. (Aluehallintovirasto s.a.d; Aluehallintovirasto s.a.e.)

Esteettömyysdirektiivin toimeenpaneva lainsäädäntö, jossa on tehty muutoksia useaan eri lakiin kuten digipalvelulakiin, tuli voimaan 1.2.2023. Esteettömyysdirektiivin tavoitteena on parantaa esteettömien tuotteiden ja palvelujen saatavuutta, jolloin helpotetaan vammaisten henkilöiden itsestä elämää. Lain soveltaminen ja valvonta tulee voimaan kesäkuussa 2025. Valvonta jakaantuu eri viranomaisille, esimerkiksi Etelä-Suomen aluehallintovirasto vastaa digipalvelulain saavutettavuusvaatimusten noudattamisen valvonnasta, liikenne- ja viestintävirasto Traficom esteettömyys-tiedosta ja tuotteiden esteettömyyden markkinavalvonnasta ja sisäministeriö vastaa yleisen hätänumeron esteettömyyden valvonnasta. (Aluehallintovirasto 23.1.2023.)

Etelä-Suomen aluehallintovirasto toimii digipalvelulain valvontaviranomaisena, joka valvoo vaatimusten noudattamista. Valvontaviranomainen antaa ohjausta ja neuvontaa säädettyjen vaatimusten, velvollisuuksien ja oikeuksien toteuttamisesta sekä osallistuu kansainvälisten saavutettavuusohjeiden valmisteluun ja huolehtii saavutettavuusstandardien ja niissä viitattujen ohjeiden kääntämisestä suomen kielelle. Viranomaisen tehtävänä on myös raportoida valvonnan tuloksista Euroopan komissiolle saatavuusdirektiivin säädetyllä tavalla. (Laki digitaalisten palvelujen tarjoamisesta 306/2019, 4 luvun 12 §.)

2.3 Verkkosisällön saavutettavuusohjeet (WCAG)

Verkkosisällön saavutettavuusohjeet, WCAG (Web Content Accessibility Guidelines), ovat ohjeistus verkkosisältöjen saavutettavuudesta. Ensimmäinen versio WCAG verkkosisällön saavutettavuusohjeista julkaistiin vuonna 1999. Suomessa saavutettavuusvaatimuksissa edellytetään WCAG 2.1 -version noudattamista, joka hyväksyttiin kesäkuussa 2018. (Celia 2022a.) WCAG 2.1 -versio

sisältää suosituksia, joiden avulla verkkosisällön saavutettavuutta voidaan parantaa henkilöille, joilla on vammoja, kognitiivisia rajoitteita, toimintarajoitteita, oppimisen tai ymmärtämisen vaikeuksia tai neurologisia sairauksia. Ohjeet kattavat laajan joukon suosituksia, mutta ne eivät pysty huomioimaan kaikenlaisia vammoja ja rajoitteita. (W3C 2019, kohta 0.1.)

WCAG-ohjeistusta voivat käyttää erilaiset kohderyhmät, kuten verkkosuunnittelijat, sovelluskehittäjät, päättäjät, hankintavastaavat, opettajat ja opiskelijat. Ohjeistusta tarjotaan useilla eri tasoilla, jotta vaihteleviin tarpeisiin voidaan vastata. Ohjeistuksen eri tasoja ovat periaatteet, yleiset ohjeet, testattavat onnistumiskriteerit ja riittävät ja neuvoa-antavat tekniikat, ja yhdessä ne toimivat verkkosisällön saavutettavuuden oppaana. (W3C 2019, kohta 0.2.)

W3C (2019, kohta 0.2) mukaan WCAG-ohjeistuksen ylätasolla ovat neljä periaatetta (kuva 1), joita ovat havaittavuus, hallittavuus, ymmärrettävyys ja toimintavarmuus. Periaatteiden tason alapuolella ovat yleiset ohjeet, jotka sisältävät yhteensä 13 ohjetta ja ne määrittelevät tarkemmin periaatteiden puitteet ja yleiset tavoitteet (kuva 1).



Kuva 1. – WCAG 2.1 -ohjeistuksen ylätason neljä periaatetta. (W3C 2019)

1. *Havaittavuus (engl. perceivable)*: Verkkosivuston sisällön tulee olla helposti havaittavissa erilaisille käyttäjille. Esimerkiksi värit, kontrasti, fonttikoko ja muut visuaaliset elementit ovat helposti havaittavissa eivätkä näkymättömissä käyttäjille. (W3C 2022.)
 - Ohje 1.1 tekstivastineet: Ei-tekstuaaliselle sisällölle on tarjottava tekstivastineet niin, että sisältö on muutettavissa eri muotoihin (W3C 2019 kohta 1.1).
 - Ohje 1.2 aikasidonnainen media: Tarjottava vastine aikasidonnaiselle medialle, eli audio- tai videomedialle (W3C 2019 kohta 1.2).
 - Ohje 1.3 mukautettava: Sisällön tuottaminen niin, että sisältöä voi esittää eri tavoin menettämättä rakennetta (W3C 2019 kohta 1.3).
 - Ohje 1.4 erottuva: Sisällön näkemisen ja kuulemisen helpottaminen käyttäjille ja etu- alan erottaminen taustasta (W3C 2019 kohta 1.4).
2. *Hallittavuus (engl. operable)*: Käyttöliittymän eri komponentit ja navigaatio täytyy olla hallittavissa. Tämä tarkoittaa, että käyttäjät pystyvät hallinnoimaan rajapintaa ja sivustolla navigointi on helppoa. Sivustolla ei saisi olla vuorovaikutuksia, joita ei voi hallita. (W3C 2022.)
 - Ohje 2.1 käytettävissä näppäimistöltä: Kaikki toiminnallisuuden toteuttaminen niin, että ne ovat käytettävissä näppäimistöltä (W3C 2019 kohta 2.1).
 - Ohje 2.2 tarpeeksi aikaa: Annettava käyttäjille tarpeeksi aikaa lukea ja käyttää sisältöä (W3C 2019 kohta 2.2).
 - Ohje 2.3 sairauskohtaus: Vältä sisältöä, jotka tiedetään aiheuttavan sairauskohtauksia, esimerkiksi liikeanimaatiot ja välähdykset (W3C 2019 kohta 2.3).
 - Ohje 2.4 navigoitava: Käyttäjille on tarjottava tapa navigoida, etsiä sisältöä ja määrittää heidän sijaintinsa (W3C 2019 kohta 2.4).
 - Ohje 2.5 syötetavat: Toimintojen käyttämisen helpottaminen käyttäjille erilaisilla syötetavoilla näppäimistön lisäksi (W3C 2019 kohta 2.5).
3. *Ymmärrettävyys (engl. understandable)*: Tiedon ja käyttöliittymän toimintojen täytyy olla ymmärrettäviä käyttäjille. Sivustolla käytetty kieli ja erilaiset toiminnot, kuten lomakkeiden täyttäminen tulee olla selkeää ja ymmärrettävää. (W3C 2022.)

- Ohje 3.1 luettava: Tekstisisällön tulisi olla luettavaa ja ymmärrettävää (W3C 2019 kohta 3.1).
- Ohje 3.2 ennakoitava: Verkkosivuston ilmiäsu ja toiminta tulisi olla ennakoitavia (W3C 2019 kohta 3.2).
- Ohje 3.3 syötteen avustaminen: Käyttäjiä on autettava välttämään ja korjaamaan virheitä (W3C 2019 kohta 3.3).

4. *Toimintavarmuus (engl. robust)*: Sivuston täytyy olla teknisesti luotettava ja toimia yleisimmillä selaimilla ja laitteilla, mukaan lukien avustavilla teknologioilla, eli sivustojen tulee olla saavutettavissa eri laitteilla (W3C 2022).

- Ohje 4.1 yhteensopiva: Yhteensopivuuden maksimointi nykyisten ja tulevien ohjelmien, mukaan lukien avustavien teknologioiden kanssa (W3C 2019 kohta 4.1).

Ohjeiden avulla voidaan ymmärtää onnistumiskriteereitä ja soveltaa teknisiä tekniikoita paremmin. Jokaisen ohjeen alle on laadittu testattavia onnistumiskriteereitä, jotka kuvaavat tarkemmin, miten saavutettavuusohjeistusta noudatetaan. (Celia 2022a.)

WCAG-ohjeistuksen onnistumiskriteerit voidaan jakaa kolmeen eri tasoon: A-, AA- ja AAA –tasoihin. A-taso on vähimmäisvaatimus, joka tulisi täyttää, jotta verkkosivusto olisi saavutettava osalle käyttäjistä. AA-tason kriteerit edellyttävät A-tason kriteereiden täyttymistä ja sen lisäksi ne sisältävät lisävaatimuksia, joilla saavutettavuutta voidaan parantaa laajemmalle joukolle. AAA-tason kriteerit edellyttävät edellä mainittujen kriteereiden täyttymistä ja tällä tasolla voidaan parantaa saavutettavuutta yhä useammalle käyttäjille. (Aluehallintovirasto s.a.f.) Joitakin onnistumiskriteereitä käydään tarkemmin läpi kappaleessa 3.3.

WCAG-ohjeistuksella voidaan lähinnä varmistaa verkkosivuston tekninen saavutettavuus ja osa ohjeistuksen kriteereistä ovat tulkinnanvaraisia. Erilaiset kohderyhmät voivat olla eri mieltä kriteerien täyttymisestä. Esimerkiksi verkkosivustossa oleviin kuviin tulee sisällyttää teksti, joka kuvailee sivustolla olevaa kuvaa, mutta eri ihmisillä on erilainen käsitys siitä, minkälainen on riittävän kuvaava teksti kuvalle. Tämän takia saavutettavuuden lisäksi on hyvä ottaa huomioon kaikki käyttäjät hyvällä käytettävyydellä ja käytettävyyssuunnittelulla. (Kehitysvammaliitto ry 23.1.2023).

2.4 Saavutettavuuden suhde käyttäjäkokemukseen

Käyttäjäkokemus ja saavutettavuus ovat läheisesti kytköksissä toisiinsa. Hyvällä saavutettavuudella tarkoitetaan kaikkien ihmisten huomioon ottamista yhdenmukaisesti verkkosivustoja suunniteltaessa ja rakentaessa (Aluehallintovirasto s.a.c). Saavutettavuuden voidaankin sanoa auttavan varmistamaan, että kaikki käyttäjät voivat käyttää sujuvasti verkkosivustoa tai -palvelua. Hyvä käyttäjäkokemus puolestaan auttaa varmistamaan, että käyttäjät käyttävät palvelua mielellään, todennäköisemmin palaavat sen pariin uudelleen ja suosittelevat sitä myös muille. (Gevorkian 24.3.2020.)

Jos verkkosivusto tai -sovellus ei ole saavutettava, monet käyttäjät kohtaavat todennäköisesti erilaisia esteitä käyttäessään sitä, ja tämä johtaa todennäköisesti huonoon käyttäjäkokemukseen. Saavutettavuuden ja kokonaisvaltaisen hyvän käyttäjäkokemuksen varmistamiseksi verkkosivuston suunnittelussa tarvitaankin huolellista käyttäjäkeskeistä suunnittelua. Sen tarkoituksena on ottaa mahdollisimman kattavasti huomioon erilaisten käyttäjien tarpeet ja toimintarajoitteet, jotta sivusto on helppokäyttöinen ja saavutettava kaikille. (Gevorkian 24.3.2020.)

2.5 Käyttäjäkokemuksen suunnittelu

Käyttäjäkokemukseen sisältyy käyttäjän subjektiivisen kokemuksen lisäksi kokoelma suunnitteluprosesseja, joiden tarkoituksena on keskittyä asettamaan käyttäjät suunnittelun ja kehityksen keskipisteeseen ja varmistaa onnistunut käyttäjäkokemus. (Babich 2019.) Käyttäjäkokemuksen suunnittelussa esiintyy usein monia käsitteitä, kuten käyttäjäkeskeinen suunnittelu, käyttäjäkokemussuunnittelu ja käyttöliittymäsuunnittelu. Nämä kaikki liittyvät läheisesti toisiinsa, mutta niillä on erilaiset painopisteet.

Käyttäjäkokemussuunnittelu (engl. User Experience Design) keskittyy erityisesti käyttäjän kokemukseen tuotteen tai palvelun käytön aikana. Se sisältää käyttöliittymän suunnittelun, visuaalisen ilmeen suunnittelun, interaktiivisuuden suunnittelun ja käyttökokemuksen arvioinnin. Käyttäjäkokemussuunnittelun tavoitteena on luoda käyttäjälle mahdollisimman miellyttävä, tehokas ja helppokäyttöinen kokemus, joka vastaa käyttäjän tarpeita ja tavoitteita. (Marsh 2015, luku 1.1; Babich 2019.)

Käyttöliittymäsuunnittelu (eng. User Interface Design) keskittyy käyttäjän vuorovaikutukseen tuotteen tai palvelun kanssa. Käyttöliittymäsuunnittelun tavoitteena on suunnitella käyttöliittymä, joka on helppokäyttöinen, esteettisesti miellyttävä ja joka helpottaa käyttäjän vuorovaikutusta tuotteen tai palvelun kanssa. Käyttöliittymäsuunnittelussa suunnitellaan käyttöliittymän visuaalisia ja

toiminnallisia elementtejä saavutettavuuden ja käytettävyyden näkökulmasta, kuten värejä, navigaatioita ja typografioita. (Marsh 2015, luku 1.1; Babich 2019.)

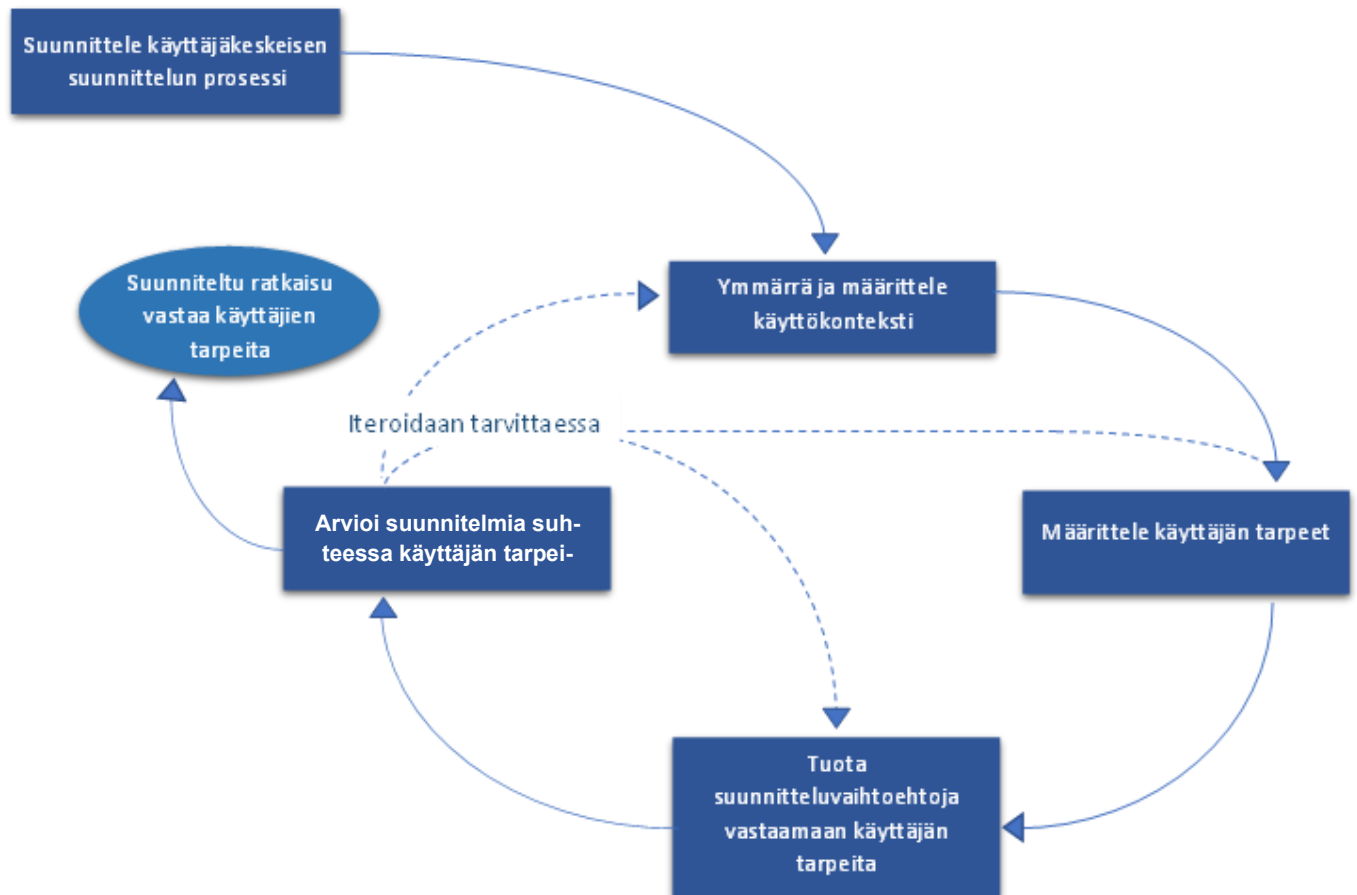
2.5.1 Käyttäjäkeskeinen suunnittelu

Käyttäjäkeskeinen suunnittelu (engl. User-Centered Design, UCD tai Human-Centered Design, HCD) on laaja käsite, joka kattaa koko suunnitteluprosessin alusta loppuun. Se sisältää käyttäjien tarpeiden ja ympäristön tunnistamisen, prototyyppien suunnittelun ja testaamisen, käyttöliittymän suunnittelun, käyttäjäpalautteen analysoinnin ja jatkuvan iteratiivisen kehittämisen. Käyttäjäkeskeisessä suunnittelussa käyttäjän tarpeet, tavoitteet ja käyttötilanteet otetaan huomioon tuotteen tai palvelun suunnittelussa, ja tavoitteena on saada tehtyä käyttäjille hyödyllinen ja helppokäyttöinen tuote. (Babich 2019; ISO 9241-210:2010 kohta 4.)

Käyttäjäkokemus, saavutettavuus ja esteettömyys liittyvät keskeisesti käyttäjäkeskeiseen suunnitteluun. ISO-standardin (9241–210:2010 kohta 3) mukaan käyttäjäkeskeisesti suunnitellut järjestelmät ovat käytettävyydeltään ja käyttäjäkokemukseltaan parempia erilaisille ihmisille, sillä ne lisäävät esteettömyyttä. Käyttäjäkeskeisen suunnittelun tulisi noudattaa seuraavia periaatteita (9241–210:2010 kohta 4):

- *Suunnittelu perustuu käyttäjien, tehtävien ja ympäristöjen selkeään ymmärtämiseen:* Kaikkien käyttäjäryhmien ja sidosryhmien tunnistaminen on tärkeää, sillä yksi keskeisimmistä syistä projektien epäonnistumisille on käyttäjien tarpeiden vajavainen ymmärrys.
- *Käyttäjät osallistetaan suunnittelun ja kehityksen jokaiseen vaiheeseen:* Käyttäjien avulla saadaan tietoa käyttötarpeista ja käyttötiheydestä.
- *Suunnittelua määrittää ja tarkentaa käyttäjäkeskeinen arviointi:* Käyttäjiltä saatava palaute on tärkeässä roolissa, sillä sen avulla voidaan minimoida riskejä siitä, että lopputuote ei vastaisi käyttäjien tarpeita.
- *Suunnitteluprosessi on iteratiivinen:* Heti kehityksen alkuvaiheessa ei ole mahdollista suunnitella valmista tai täydellistä järjestelmää.
- *Suunnittelu koskee koko käyttäjäkokemusta:* Kaikki käyttäjäkokemuksen määrittelyn mukaiset osa-alueet on huomioitava.
- *Suunnittelutiimiin kuuluu monialaisia taitoja ja näkökulmia:* Laaja osaaminen ja tiiminjäsenten välisessä yhteistyössä syntyvät ideat hyödyttävät koko projektia.

Suunnitteluprosessi sisältää usein erilaisia menetelmiä, kuten käyttäjätutkimuksia, käyttäjähaastatteluja, käyttäjätestauksia ja prototyyppien luontia, joiden avulla suunnittelijat voivat saada käyttäjäpalautteen ja parantaa tuotteen tai palvelun käyttökokemusta. (Marsh 2015, luku 1.1; Babich 2019.) Suunnitteluprosessi on iteratiivinen, eli prosessin vaiheita toistetaan kuten kuvasta 4 havaitaan, kunnes suunnittelun tulos ja käyttäjävaatimukset kohtaavat (ISO 9241-210:2010 kohta 6).



Kuva 2. – Käyttäjäkeskeisen suunnittelun prosessi (mukaillen ISO 9241-210:2010 kohta 6)

2.6 Hyvän käyttäjäkokemuksen muodostuminen

Hyvä käyttäjäkokemus muodostuu monesta eri tekijästä. Yleisesti ottaen voidaan sanoa, että se syntyy silloin, kun käyttäjän tarpeet, tavoitteet ja odotukset vastaavat tuotteen tai palvelun tarjoamaa toiminnallisuutta ja käyttöliittymää. Tällöin tuote tai palvelu on miellyttävä, tehokas, helppo käyttää ja arvoa tuottava. (Viljanen 22.2.2020.)

Hyvä käyttäjäkokemus voi vaikuttaa merkittävästi tuotteen tai palvelun käyttöasteeseen, käyttäjän tyytyväisyyteen, maineeseen ja yrityksen taloudellisiin tuloksiin. Hyvän käyttäjäkokemuksen

muodostumista voidaan edesauttaa käyttämällä käyttäjäkeskeistä suunnittelua, joka huomioi käyttäjien tarpeet, tavoitteet ja käyttötilanteet sekä tarjoaa helppokäyttöisen ja tehokkaan käyttöliittymän. (Babich 2019; Viljanen 22.2.2020).

Hyvään käyttäjäkokemukseen liittyy oleellisesti käytettävyys. Käytettävyyden ISO-standardissa määritellään käytettävyyden tarkoittavan vaikuttavuutta, tehokkuutta ja tyytyväisyyttä, joita järjestelmän, tuotteen tai palvelun määritellyt käyttäjät kokevat saavuttaessaan tavoitteensa määritetyssä ympäristössä (ISO 9241–11:2018, kohta 3.1). Käyttäjät kokevat todennäköisemmin tuotteen tai palvelun käytön miellyttävänä ja tehokkaana, mikäli he pystyvät helposti löytämään tarvitsemansa toiminnot ja käyttämään niitä ilman suurta ponnistelua (Viljanen 22.2.2020).

Jacob Nielsen (2012) jakaa käytettävyyden viiteen komponenttiin, joiden perusteella käytettävyyden laatua voidaan arvioida. Näitä ovat opittavuus, tehokkuus, muistettavuus, virhesietoisuus ja tyytyväisyys.

- *Opittavuus (engl. learnability)*: Kuinka helppoa käyttäjän on suorittaa haluamansa tehtävät sivustolla tai palvelussa ensimmäisellä kerralla.
- *Tehokkuus (engl. efficiency)*: Kuinka nopeasti tehtävä voidaan suorittaa, kun palvelu tai verkkosivusto on jo entuudestaan käyttäjälle tuttu.
- *Muistettavuus (engl. memorability)*: Kuinka nopeasti käyttäjä pystyy jälleen toimimaan tehokkaasti, kun hän palaa palveluun tai sivustolle pidemmän ajan jälkeen.
- *Virhesietoisuus (engl. errors)*: Kuinka paljon käyttäjät tekevät sivustolla tai palvelua käyttäessään virheitä, minkälaisia virheet ovat ja miten käyttäjät selviytyvät niistä.
- *Tyytyväisyys (engl. satisfaction)*: Kuinka tyytyväinen käyttäjä on palveluun tai sivuston käyttöön ja käytettävyyteen.

(Nielsen 2012; Viljanen 22.2.2020.)

Myös käyttäjäkokemuksen arviointiin on olemassa malleja, joita käyttämällä voidaan edesauttaa hyvän käyttäjäkokemuksen muodostumista. Yksi tunnetuimmista on Peter Morvillen (2004) luoma hunajakennomalli (engl. User Experience Honeycomb), jolla hahmotellaan hyvän käyttäjäkokemuksen peruseräitä.

2.6.1 Morvillen hunajakennomalli

Morvillen hunajakennomalli koostuu seitsemästä osa-alueesta (kuva 3), joita ovat hyödyllisyys, käytettävyys, haluttavuus, löydettävyys, saavutettavuus, arvokkuus sekä luotettavuus. Yhdessä nämä osa-alueet muodostavat hyvän käyttäjäkokemuksen. (Morville 2004; Viljanen 22.2.2020.)



Kuva 3. – Hunajakennomalli (mukaillen Morville 2004; Viljanen 22.2.2020)

- *Hyödyllinen (engl. useful)*: Määritetyn kohderyhmän tulee saada tuotteesta hyötyä, ja tuotteen tulee tarjota tarkoitus käyttäjilleen. Tuotteen tulee myös auttaa saavuttamaan tavoitteensa loogisella tavalla. Hyödyn ja tarkoituksen tulee näkyä selkeästi myös käyttäjälle. Mikäli tuotteesta saatava hyöty ei ilmene selkeästi käyttäjälle, käyttäjä ei todennäköisesti pala enää tuotteen pariin. (Morville 2004; Viljanen 22.2.2020.)
- *Käytettävä (engl. usable)*: Tuotteen tulee olla käytettävissä ja helppokäyttöinen. Tällöin käyttäjä kykenee saavuttamaan tavoitteensa ja onnistuu tehtävien suorittamisessa vaivattomasti ja tehokkaasti. (Morville 2004; Nielsen 2012.)
- *Haluttava (engl. desirable)*: Tuotteen haluttavuus muodostuu ilmeestä, identiteetistä, brändistä, esteettisyydestä ja subjektiivisesta kokemuksesta. Tuotteen tulee olla mielenkiintoinen ja visuaalisesti miellyttävä käyttäjille ja sitä tulee olla helppo ja mukava käyttää.

Esimerkiksi intuitiivinen ja käyttäjää johdatteleva käyttöliittymä auttaa käyttäjää toimimaan sivustolla ja vaikuttaa siten haluttavuuteen. (Rae 2020; Viljanen 22.2.2020.)

- *Löydettävä (engl. findable)*: Esimerkiksi sivustolta tulee löytyä haluttu tieto helposti ja navigointi toteutuu loogisesti läpi sivuston. Jokaiselta sivuston yksittäiseltä sivulta tulisi olla helppoa siirtyä eteenpäin, sillä useimmat käyttäjät saapuvat sivustoille yleensä jonkin muun, kuin etusivun kautta. Käyttäjän eksyminen digitaalisessa alustassa on isossa roolissa huonossa käyttäjäkokemuksessa. (Rae 2020; Viljanen 22.2.2020.)
- *Saavutettava (engl. accessible)*: Kaikkien käyttäjien tulee voida kokea sama käyttökokemus tuotteen käytöstä ja tuotteen tulee olla kaikkien käytettävissä rajoitteista riippumatta. (Morville 2004; Viljanen 22.2.2020.)
- *Luotettava (engl. credible)*: Rakentuu sisäisistä ja ulkoisista tekijöistä. Esimerkiksi sivuston visuaalisesti miellyttävä käyttöliittymä ja navigoinnin loogisuus ja helppous ovat avaintekijöitä luotettavuudessa. Luotettava sivusto on käyttäjälle myöskin selkeä, asianmukainen, moderni ja helppo käyttää, sekä lukea. Sisältö on ajantasaista, tarkistettua eikä sisällä esimerkiksi merkittäviä kielioppivirheitä. (Rae 2020; Viljanen 22.2.2020.)
- *Arvokas (engl. valuable)*: Kokemuksen tulee tuoda arvoa käyttäjälleen. Esimerkiksi sivusto voi olla miellyttävä, käyttökelpoinen tai haluttu, mutta jos sillä ei ole luontaista arvoa, käyttäjällä ei ole syytä palata verkkosivustolle. (Rae 2020.) Hunajakennomallin kaikkiin osa-alueisiin panostaminen saa käyttäjän todennäköisemmin kokemaan sivuston tai palvelun arvokkaana (Viljanen 22.2.2020).

3 Saavutettavien verkkosivustojen toteutus

Tässä vaiheessa tiedämme jo, että saavutettava verkkosivusto on suunniteltu siten, että se on helppo käyttää ja helposti ymmärrettävissä kaikille käyttäjille. Käyttöliittymä- ja käyttäjäkokemuksen suunnittelun lisäksi saavutettavia verkkosivustoja toteuttaessa tulisi huomioida myös sivuston tekninen saavutettavuus ja sen vastaaminen saavutettavuusvaatimuksia ja –standardeja. Tämä vaatii huomiota muun muassa sivuston selkeään asetteluun ja rakenteeseen, sisällön määrään, lukemisjärjestyksen loogisuuteen, navigointiin, painikkeisiin, tekstiin, fonttiin, kontrastiin, kuviin, lomakkeisiin, automaattisesti käynnistyviin mediasisältöihin ja hakutoimintoihin. (Kehitysvammaliitto ry 21.2.2023b.) Nämä liittyvät läheisesti digitaalisen palvelun saavutettavuuden osa-alueisiin; helppokäyttöisyyteen ja ymmärrettävyyteen.

Verkkosivuston helppokäyttöisyyden ja ymmärrettävyyden lisäksi yllä mainittujen tekijöiden teknisesti virheetön toteutus on tärkeä saavutettavuuden osa-alue (Robbins 2018, luku 1). Kun verkkosivuston tekninen toteutus on mahdollisimman virheetön, toimii se todennäköisimmin oikein myös erilaisilla avustavilla teknologioilla ja on riippumaton käyttäjän käyttämästä laitteesta. (Aluehallintovirasto s.a.c.)

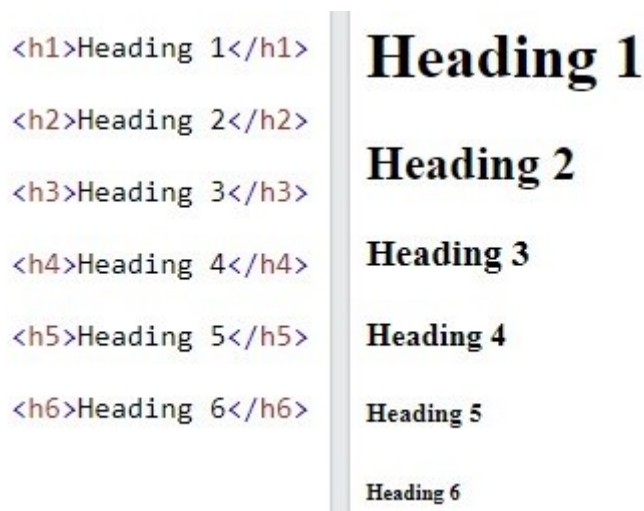
Yleisiä verkkosivuston rakentamiseen käytettyjä teknologioita ovat HTML, CSS ja JavaScript. HTML:n avulla lisätään sivustolle rakennetta ja sisältöä, CSS:n avulla tyyliä ja JavaScriptia käyttämällä saadaan vuorovaikutusta verkkosivustoon. (Robbins 2018, luku 1.) Näiden teknologioiden käytössä ja sivuston käyttöliittymärakenteen toteutuksessa on kuitenkin erilaisia saavutettavuuden kannalta oleellisia tekijöitä, joten käymme näitä seuraavaksi läpi tarkemmin.

3.1 HTML

HTML (Hypertext Markup Language) on merkintäkieli, joka kuvailee erilaisia verkkosivuston rakenteita ja sisältöä. HTML on standardisoitu ja W3C-konsortio ylläpitää ja kehittää standardia. HTML-standardin noudattaminen kuuluu tekniseen saavutettavuuteen. HTML perustuu tunnisteisiin eli tageihin, joiden avulla voidaan määritellä erilaisia elementtejä verkkosivustolle, kuten otsikoita, kappaleita, linkkejä, kuvia ja listoja. Merkinnoilla voidaan osoittaa verkkosivuston taustalla olevaa rakennetta. HTML:n käyttö täytyy olla standardin mukaista ja virheetöntä, jotta sisältö näkyy oikein eri selaimilla ja laitteilla, sekä toimii avustavilla teknologioilla. (Robbins 2018, luku 1; Aluehallintovirasto s.a.a.)

Semanttisella HTML-merkintäkielellä voidaan parantaa verkkosivustojen saavutettavuutta. HTML-elementit välittävät selaimelle ja avustaville teknologioille, mitä verkkosivusto sisältää. Tällöin esimerkiksi ruudunlukuohjelmat puhuvat sivuston sisällön ääneen. (W3Schools s.a.)

Semanttisessa HTML-kielessä otsikoinnit oikeilla elementeillä ovat tärkeitä, koska ne ilmaisevat sivun rakennetta ja merkitsevät otsikoita. Otsikkoelementtien, kuten `<h1>`, `<h2>`, `<h3>` jne. avulla käyttäjä pystyy hahmottamaan sivun sisällön ja hierarkian. (W3Schools s.a.) Oikein merkityt ja johdonmukaiset otsikot toimivat myös paremmin ruudunlukulaitteissa, sillä tällöin käyttäjä pystyy hypäämään kätevämmiin otsikosta seuraavaan (Viljanen 1.9.2019). Lisäksi hakukoneet käyttävät sivun rakennetta ja otsikoita ymmärtääkseen sen sisältöä ja indeksoidakseen sen hakutuloksissa (W3Schools s.a.). Alla esimerkki otsikkoelementtien käytöstä kuvassa 4.



Kuva 4. – Esimerkki eri otsikkoelementeistä

HTML-elementtiin lisättävä `lang`-attribuutti `<html lang="fi">` määrittää sivun pääkielen ja kertoo selaimelle sekä hakukoneille, mitä kieltä sivu käyttää. (W3Schools s.a.) Se auttaa myös avustavia teknologioita, kuten ruudunlukusovelluksia ja kielityökaluja ymmärtämään sivun kielen. Lisäksi jos käyttäjä lukee sivua, joka on kirjoitettu eri kielellä kuin hänen käyttöjärjestelmänsä kieli, selain voi tarjota käännöstyökaluja tai kysyä käyttäjältä haluaako hän vaihtaa kieltä. (Bailey 23.10.2018.)

Elementit, kuten `<div>` ja `` ovat vain asettelua varten. Ne ovat semanttisesti merkityksettömiä eivätkä ne välitä mitään selaimelle. Vastaavasti `<button>` -elementti on semanttisesti merkityksellinen, sillä kyseessä on selvästi painike (kuva 5). Esimerkiksi `<div>` tai `` -elementtejä ei saisi käyttää painikkeena, jos on olemassa `<button>` -elementti painikkeelle. Tärkeimpiä semanttisia HTML-elementtejä verkkosivuston rakenteessa ovat `<header>`, `<nav>`, `<main>`, `<article>`, `<aside>` ja `<footer>`. (Gilbert 2019, luku 2.)

KOODI:	<code><button>Esimerkki</button></code>	<code><div>Esimerkki</div></code>
TULOSTAA:	<input type="button" value="Esimerkki"/>	Esimerkki

Kuva 5. – Esimerkki semanttisesta `<button>` ja ei-semanttisesta `<div>` -elementeistä

WAI-ARIA (Web Accessibility Initiative – Accessible Rich Internet Applications) paremmin tunnettu ARIA on W3C:n kehittämä standardi, jolla pyritään parantamaan toiminnallisuuksien saavutettavuutta ruudunlukuohjelmien käyttäjille verkkosivustoissa. ARIA täydentää HTML-merkintäkielen standardia. (Aluehallintovirasto s.a.a.) ARIA on suunniteltu tarjoamaan joukko ylimääräisiä semantiikkoja isäntäkielille, kuten HTML-merkintäkielelle. ARIA:n roolit, tyypit ja ominaisuudet lisäävät semanttista tietoa, jos isäntäkielen elementiltä puuttuu kyseinen semantiikka. Tästä esimerkkinä kuvassa 6 näkyvä `<div>` -elementti, jolle on annettu rooli ”button”.

```
<div role="button">Esimerkki</div>
```

Kuva 6. – Esimerkki ARIA:n käytöstä `<div>` -elementissä

Isäntäkielet voivat kehittyä ajan myötä ja tarjota uusia natiiviominaisuuksia, jotka vastaavat ARIA-ominaisuuksia. Tällöin ARIA:n käyttö on tarpeetonta isäntäkielen kanssa, kuten esimerkki kuvassa 6, olisi parempi hyödyntää HTML:n `<button>` -elementtiä. ARIA on tarkoitettu parantamaan natiivisemantiikkaa tai muuttamaan semantiikka eri tarkoitukseen kuin alkuperäisessä tarkoituksessa. (W3C 2023 kohta 8.4.)

3.2 CSS ja JavaScript

Verkkosivusto on mahdollista rakentaa vain käyttämällä HTML-merkintäkieltä, mutta lisäämällä myös CSS (Cascading Style Sheets) voidaan muokata verkkosivujen tyyliä ja ulkoasua. CSS on tyylikieli, joka kuvailee, miten sisältö tulisi näyttää. Tyylikielellä voidaan määrittää verkkosivuston värit, fontit, taustakuvat, sisennykset, marginaalit ja animaatiot. (Robbins 2018, luku 1.) W3C (s.a.a) mukaan CSS mahdollistaa myös verkkosivuston responsiivisuuden, eli verkkosivuston muuttumisen eri laitteiden näyttökokoihin.

JavaScript on ohjelmointikieli, joka lisää vuorovaikutusta ja käyttäytymistä verkkosivustolla. JavaScriptin avulla voidaan antaa palautetta välittömästi ilman sivuston päivittämistä. Esimerkkinä välittömästä palautteesta on palaute lomakkeiden virheellisesti täytetyistä kentistä. JavaScriptillä

voidaan tarkistaa lomakekentät, lisätä viesti virheellisiin kenttiin ja muokata niiden tyyliä. Lyhyesti sanottuna JavaScript auttaa luomaan responsiivisen käyttöliittymän, joka parantaa käyttäjäkokemusta ja tarjoaa dynaamista toimintaa ilman, että sivua pitäisi ladata uudelleen. (Robbins 2018, luku 3.)

3.3 Saavutettavan sivuston rakenne

Selkeät ja yhdenmukaiset sivupohjat verkkosivuston jokaisessa osassa ovat tärkeitä saavutettavien ja käytettävien verkkosivustojen suunnittelussa. Sivupohjien osien on erotuttava toisistaan selkeästi ja niiden tulee olla tunnistettavissa. Sivuston olennainen sisältö täytyy erottua selkeästi ja tärkeimmät asiat tulisi esittää heti sivun alussa. (Kehitysvammaliitto ry 16.3.2023.) Selkeällä rakenteella ja asettelulla voidaan vaikuttaa sisällön määrään. Sivuilla ei saisi olla liian suuri määrä sisältöä, sillä se voi häiritä käyttäjän tarkkaavaisuutta ja henkilöitä, joilla on lukihäiriö. (Gilbert 2019, luku 3.)

Sivustojen suunnittelussa tulisi myös ottaa huomioon näppäimistöselaamisen mahdollistaminen, eli mahdollisuus käyttää sivustoa näppäimistön kautta, joka on myös osa WCAG-ohjeistusta. Näppäimistökäyttäjät selaavat verkkosivustoja tavallisesti käyttäen sarkain-, eli TAB-näppäintä, Enter-näppäintä sekä nuolinäppäimiä. Sarkainnäppäimellä selataan tavallisesti verkkosivuston toiminnallisia kohteita, kuten linkkejä, painikkeita ja lomakkeita. Enter-näppäimellä käyttäjä tyypillisesti suorittaa toimintoja samalla tavalla, kuten hiirtä käyttäessä vasenta painiketta klikkaamalla, ja nuolinäppäimillä vierittää sivunäkymää alas ja ylös. (Kehitysvammaliitto ry 1.2.2023.)

Tekstisisältö

Tekstien riittävällä suuruudella voidaan parantaa luettavuutta heikkonäköisille ja käyttäjille, joilla on lukihäiriö. Jotkin käyttäjät pitävät suuresta tekstikoosta, kun taas osa lukee mieluummin pienempää tekstiä. Tämän takia sivustolla on suositeltavaa olla toiminto, jolla voidaan suurentaa tai pienentää tekstin kokoa. (Kehitysvammaliitto ry 24.1.2023a.) WCAG-ohjeistuksen ohjeen 1.4 kriteerin mukaan tekstin kokoa tulisi voida mukauttaa ilman avustavaa teknologiaa 200 prosenttiin asti ilman, että sisältö katoaa (W3C 2019 kohta 1.4.4). Selkeällä fontilla voidaan vaikuttaa ymmärrettävyyteen ja usein käytetty Arial-fontti on hyvin luettava useille käyttäjälle (Kehitysvammaliitto ry 24.1.2023a).

Heikkonäköisillä käyttäjillä voi olla vaikeuksia lukea tekstiä matalakontrastisilla taustoilla. Suurempi kontrastisuhde tekstin ja taustan välillä auttaa heikkonäköisiä lukemaan ilman kontrastia

parantavaa avustavaa teknologiaa. Myös muut käyttäjät kuin heikkonäköiset saavat mukavan lukukokemuksen paremmalla värikontrastilla. (Gordon 2022.) W3C (2019 kohta 1.4.3) mukaan WCAG-ohjeistuksen ohjeessa 1.4 tekstien kontrastisuhde tulisi olla vähintään 4,5:1, paitsi isokokoiselle tekstille kontrastisuhde tulisi olla vähintään 3:1. Esimerkki tekstin ja taustavärien välisistä erilaisista kontrasteista esitetään kuvassa 7.



Kuva 7. – Tekstin ja taustan välinen kontrasti (mukaillen Steenhout 20.9.2017)

Visuaalinen sisältö

Interaktiiviset elementit, kuten linkit ja painikkeet täytyy olla tunnistettavia. Linkkien tulisi erottua normaalista tekstistä ja näyttää selkeästi, että ne ovat klikattavia. Yleinen tapa erottaa linkit muusta sisällöstä on alleviivata ja merkitä ne sinisellä värillä. Alleviivaus on hyödyllinen, jos käyttäjä ei erota värejä. Lisäksi linkkien tekstin tulee kuvata selkeästi, mihin käyttäjä siirtyy linkkiä painamalla. Yleistä tekstiä, kuten “paina tästä” tai “lue lisää” tulee välttää. (Kehitysvammaliitto ry 26.1.2023a.)

Kuvissa on oltava ALT (alternative) -attribuutti, joka kuvailee kuvan sisältöä. Tämä on hyödyllinen ominaisuus erityisesti heikkonäköisille tai näkövammaisille käyttäjille, jotka käyttävät ruudunlukuohjelmaa verkkosivustoissa. Alt-tekstistä käyttäjät saavat jonkinlaisen käsityksen, mitä kuvassa on. (Gordon 2022.) Jos kuvalla ei ole alt-tekstiä, niin ruudunlukuohjelma yleensä joko hyppää sen yli tai lukee kuvan tiedostonimen, mikä ei ole ideaalia käyttäjille. Alt-teksti on myös hyödyllinen tilanteissa, jossa kuva ei jostain syystä lataudu, sillä tällöin kuvaava teksti kuvasta näkyy käyttäjille, kuten kuvassa 8 havainnollistetaan. (Latva 2023.)



Kuva 8. – Esimerkki kuvasta ja alt-tekstistä, joka näkyy kuvan tilalla tarvittaessa

Saavutettavuudessa verkkosivustoilla ei suositella käytettävän itsestään käynnistyviä mediasisältöjä, ja jos niitä on, WCAG-ohjeistuksen ohjeen 2.2 mukaan niiden tulee olla helposti pysäytettävissä. Automaattisesti käynnistyvällä mediasisällöllä tarkoitetaan mediasisältöjen, kuten videoiden tai musiikin käynnistymistä itsestään ilman, että käyttäjä hallitsee käynnistymistä. Automaattinen mediasisällön käynnistys saattaa olla vaarallista, sillä se voi mahdollisesti aiheuttaa käyttäjällä paniikkikohtauksen tai ahdistusta, kun ääni kuuluu yhtäkkiä voimakkaasti. Ne saattavat myöskin vaikeuttaa sivuston käyttöä avustavilla teknologioilla, kuten ruudunlukuohjelmilla. (Gilbert 2019, luku 3.)

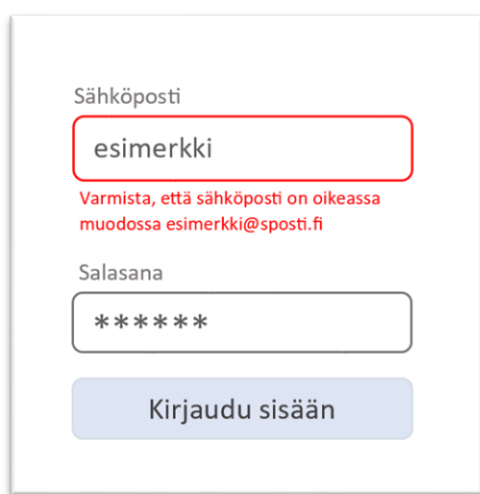
Lomakkeet

Lomakkeiden pitää olla ymmärrettäviä ja saavutettavia. Niiden tulisi olla mahdollisimman yksinkertaisia ja lyhyitä siten, että vain pakolliset tarvittavat tiedot kysytään. Lomakkeen täyttöön tulee tarjota selkeät ohjeet, joista käyttäjälle selviää, miten tiedot tulee syöttää lomakkeeseen ja kuinka lomake täytetään oikealla tavalla. (Kehitysvammaliitto ry 24.1.2023.) Lomakkeiden kenttien tulisi myös sijaita ruudulla allekkain ja olla ryhmiteltynä aihealueittain, esimerkiksi yhteystietoja käsittelevässä lomakkeessa nimitiedot omana osanaan ja osoitetiedot omanaan. Semanttisessa HTML-kielessä vastaava ryhmittely voidaan toteuttaa esimerkiksi <fieldset> ja <legend> -elementtejä käyttämällä. (Abou-Zahra, Eggert, Elton 27.7.2019.)

Jokaisella lomake-elementillä tulisi olla nimilappu (engl. label) -elementti, josta käy ilmi, mihin lomakekenttään kirjoitetaan ja mitä tietoa siihen tulee syöttää. Nimilapussa on myös hyvä mainita tai

merkitä, jos kenttä on pakollinen. Ruudunlukuohjelmat pystyvät lukemaan ääneen nimilaput käyttäjälle. Käyttäjälle olisi myös hyvä kertoa, missä muodossa tietoa tulee kirjoittaa lomakkeeseen, eli onko kyseessä teksti- vai numeromuodossa syötettävä tieto ja tuleeko desimaalit erottaa pilkulla, vai pisteellä. (Kehitysvammaliitto ry 24.1.2023.)

Lomakkeen täytön aikana käyttäjän tulisi saada palautetta, jos lomakkeessa on täytetty kenttiä virheellisesti, tai kun lomake on lähetetty onnistuneesti. Virheellisesti täytetyistä kentistä saatava palaute tulisi olla selkeä ja kertoa, mitkä kentät ovat virheellisiä ja mahdollisesti ohjeistaa virheiden korjaamiseksi (kuva 9). Lomakkeen täyttäminen ja lähettäminen pelkkää näppäimistöä käyttäen tulisi olla myös mahdollista. (Kehitysvammaliitto ry 24.1.2023b.)



The image shows a login form with the following elements:

- A label "Sähköposti" (Email) above a text input field.
- The text input field contains the word "esimerkki" (example) and is outlined with a red border, indicating an error.
- Below the email field, a red error message reads: "Varmista, että sähköposti on oikeassa muodossa esimerkki@sposti.fi" (Ensure that the email is in the correct format).
- A label "Salasana" (Password) above a password input field.
- The password input field contains six asterisks "*****".
- A blue button labeled "Kirjaudu sisään" (Log in) is positioned below the password field.

Kuva 9. – Esimerkki lomakkeesta ja virhepalautteesta.

3.4 Saavutettavuuden testaaminen

Saavutettavuuden testaamisella voidaan varmistaa, että verkkosivusto noudattaa saavutettavuusvaatimuksia. Verkkosivuston saavutettavuuden testaamiseen ja arviointiin on useita menetelmiä. Jokaisella menetelmällä on omat vahvuutensa ja heikkoutensa, joten sivustoja kannattaa testata eri menetelmillä. Saavutettavuuden testaamista suositellaan kaikissa kehitysvaiheissa, eli suunnittelu- ja koodausvaiheessa, sekä verkkosivuston julkaisun jälkeen. (Kehitysvammaliitto ry 3.4.2023.) Verkkosivustojen saavutettavuutta voidaan testata manuaalisesti, avustavilla teknologioilla ja automaattitestauksella (Viljanen 1.9.2019). Verkkosivustojen saavutettavuuden testaaminen ja arviointi perustuu WCAG 2.0 ja sitä uudempien versioiden mukaisiin A-, AA- ja AAA-tason vaatimuksiin.

Ennen saavutettavuuden testaamista on hyvä suunnitella saavutettavuuden testaamisen vaiheet ja asiat, joita tullaan testaamisessa huomioimaan. Ensiksi määritellään saavutettavuuden testaamisen tavoite ja tarkoitus, kuten mitä testaamisella halutaan saavuttaa, ja aiotaanko testauksen jälkeen esittää saavutettavuusongelmien korjausehdotukset. Seuraavaksi määritellään menetelmät, eli ketkä testaavat sivustoa, mitä saavutettavuuden ohjeistuksia tai standardeja hyödynnetään, testataanko teknisen saavutettavuuden lisäksi myös helppokäyttöisyyttä ja ymmärrettävyyttä, ja miten tuloksia raportoidaan. Näiden jälkeen määritellään testaamisen ja arvioinnin kohde, eli verkkosivusto, jota tullaan testaamaan ja arvioimaan ja sitten arvioinnin toteutuksen ajankohta. Lopuksi raportoidaan selkeästi ja ymmärrettävästi, täyttääkö verkkosivusto asetetut tavoitteet, missä kohdissa ilmenee saavutettavuusongelmia ja miten ongelmat voidaan korjata. (Kehitysvammaliitto ry 26.1.2023b.)

3.4.1 ACT-testaussäännöt

Vuonna 2017 käynnistetyssä Web Accessibility Initiative - Advanced Decision Support Tools for Scalable Web Accessibility Assessments (WAI-Tools) -projektissa kehitettiin ACT-testaussäännöt (engl. Accessibility Conformance Testing rules) WCAG:n saavutettavuusvaatimusten systemaattiseen testaamiseen. Projektin tavoitteena oli muun muassa yhdenmukaisen tarkkuuden varmistaminen automaattisessa, puoliautomaattisessa ja manuaalisessa testauksessa, sekä saavutettavuusvaatimusten tulkintojen selkeyttäminen. (W3C s.a.b.) Yhtenä ACT-testaussääntöjen tavoitteena on, että saavutettavuusvaatimusten tulkinnassa ei tapahtuisi jatkossa merkittäviä eroavaisuuksia eri testaajien, organisaatioiden ja maiden välillä (W3C s.a.c).

ACT-testaussäännöt voidaan jakaa kahteen sääntötyyppiin: atomisiin- ja yhdistelmäsääntöihin. Atomiset säännöt kuvaavat, kuinka yksittäistä toteutusta tulee testata. Yksi atominen sääntö sisältää tarkan määritelmän siitä, mitä elementtejä ja osia sisällöstä testataan ja milloin testauskohteiden katsotaan täyttävän säännönmukaiset vaatimukset tai vastaavasti epäonnistuvan vaatimusten täytössä. Atomiset säännöt tulee pitää nimensä mukaisesti pieninä ja hyvin tarkkoina. Ne testaavat yksittäistä ominaisuutta tai ehtoa, eivätkä ne ole riippuvaisia muiden sääntöjen testien tuloksista. (Abou-Zahra, Fiers, Kraft & Mueller 31.10.2019.)

Yhdistelmäsäännöt kuvaavat, miten useamman atomisen säännön testituloksesta muodostetaan yksi lopputulos jokaiselle testauskohteelle. Yhdistelmäsääntö voi sisältää useita osaehtoja, joista jokainen tulee testata erikseen atomisilla säännöillä. (Abou-Zahra ym. 31.10.2019.)

WAI-Tools-projektin rahoittajana toimi Euroopan Unioni ja siihen osallistuivat kansainvälinen W3C, hollantilainen saavutettavuusauditoiteja tekevä Accessibility Foundation sekä Portugalin ja Norjan

valvontaviranomaiset. Myös saavutettavuuden testaamiseen ohjelmistoja kehittäneet tahot, kuten muun muassa Deque Systems, IBM Accessibility, Level Access, Siteimprove ja Lissabonin yliopisto osallistuivat projektiin ja sisällyttivät projektista syntyneet ACT-testaussäännöt omiin ohjelmistoihinsa. Näitä ohjelmistoja olivat esimerkiksi testausytimet Axe-Core (Deque Systems), Equal Access Accessibility Checker (IBM Accessibility), Access Engine (Level Access), Alfa (Siteimprove) ja QualWeb-core (Lissabonin yliopisto). (Konsi 16.1.2023a; W3C s.a.d.)

3.4.2 Manuaalinen testaus

Saavutettavuuden manuaalinen testaus tarkoittaa prosessia, jossa esimerkiksi saavutettavuuden asiantuntijat tai testaajat tarkistavat manuaalisesti verkkosivuston tai -sovelluksen saavutettavuuden tason. Tämä tarkoittaa, että testaajat käyttävät sivustoa tai sovellusta eri laitteilla, eri käyttöjärjestelmissä ja eri selaimissa sekä käyttävät avustavia teknologioita, simuloidakseen käyttäjien kokemuksia. (UsableNet 2.2.2023.)

Manuaalinen testaus on tärkeä osa saavutettavuuden varmistamista, sillä se mahdollistaa monimutkaisempien saavutettavuusongelmien tunnistamisen, jotka eivät välttämättä ole havaittavissa automaattisilla testaustyökaluilla. Manuaalinen testaus mahdollistaa myös yksityiskohtaisemman tarkastelun sivuston toimivuudesta ja käytettävyydestä eri tilanteissa, kuten esimerkiksi, kun käyttäjä käyttää sivustoa avustavien teknologioiden avulla. (UsableNet 2.2.2023.)

Manuaalisesti voidaan testata verkkosivuston tekstin mukautuminen, visuaalisuus ja pelkän näppäimistön käyttö. Manuaalisesti voidaan myös kokeilla, pystyykö verkkosivustossa pienentämään ja suurentamaan tekstiä 200 prosenttiin asti ilman, että sisältö hajoaa. Tekstiä voidaan myös lukea useaan kertaan ja tarkistaa mahdolliset kielioppivirheet. (Viljanen 1.9.2019.)

Ottamalla kuvat pois käytöstä selaimesta, voidaan visuaalisuutta testata manuaalisesti. Tällä tavoin voidaan tarkistaa, ovatko alt-tekstit ymmärrettäviä ja johdonmukaisia. Toinen keino testata visuaalisuutta on poistaa mahdolliset tyylitiedostot käytöstä, jolloin nähdään, toimiiko verkkosivusto oikein ilman tyyliä. (Viljanen 1.9.2019.)

Testaamalla manuaalisesti ilman hiirtä, eli näppäimistötestaamisella nähdään, toimiiko verkkosivusto pelkästään näppäimistöä käyttämällä. Sivustoa pitäisi pystyä selaamaan ilman hiirtä ja tämän testaaminen onnistuu parhaiten manuaalisesti. (Viljanen 1.9.2019.) Samalla testauksessa näkee, pystyykö käyttäjä havaitsemaan, mikä käyttöliittymän elementti on kohdistettuna milloinkin (Näkövammaisten liitto ry 2019).

Saavutettavuutta tulisi testata manuaalisesti myös eri selaimilla ja selainversioilla, sillä niissä saat-
taa olla erilaisia saavutettavuuteen liittyviä ongelmia. Sivustoilla käytetyt ratkaisut eivät välttämättä
toimi kaikilla selaimilla ja selainversioilla. Yleensä selaimien ja selainversioiden välillä on eroja vi-
suaalisessa ilmeessä, jolloin avustavan teknologian käyttö sivustolla saattaa toimia eri tavoin eri
selaimilla. Yleisimpiä selaimia, joilla kannattaa testata ovat Google Chrome, Safari, Mozilla Firefox
ja Microsoft Edge. (Kehitysvammaliitto ry 31.1.2023a.)

Eri selaimilla testaamisen lisäksi on myös hyvä testata verkkosivustoa eri päätelaitteilla. Eri pääte-
laitteita voivat olla esimerkiksi Windows-tietokoneet, Mac-tietokoneet, Android-älypuhelimet ja
iPhone-älypuhelimet. Verkkosivustot näyttävät erilaisilta eri päätelaitteilla. Myös laitteiden käyttöjär-
jestelmät ja niiden versiot voivat vaikuttaa sivuston käytettävyyteen. Testaamalla verkkosivustoa
eri päätelaitteilla voidaan varmistua siitä, että käyttäjän käyttämä päätelaite ei vaikuta verkkosivus-
ton saavutettavuuteen. (Kehitysvammaliitto ry 31.1.2023a.)

3.4.3 Käyttäjätestaus

Käyttäjätestaus on yksi käyttäjäkeskeisen suunnittelun menetelmistä. Käyttäjätestauksessa voi-
daan havaita saavutettavuuden ja käytettävyyden ongelmia. Käyttäjätestaukseen kannattaa osal-
listaa monipuolisesti erilaisia käyttäjiä, kuten toimintarajoitteisia, ikääntyneitä ja vieraskielisiä käyt-
täjiä, jotta voidaan varmistaa verkkosivuston saavutettavuus kaikille käyttäjille, eikä vain tietyille
käyttäjryhmille. (Kehitysvammaliitto ry 31.1.2023b.)

Käyttäjät suorittavat testauksessa yleensä verkkosivustolla tehtäviä, kuten verkkokauppasivustolla
tuotteen tilaamista ja maksamista. Testaajat voivat kokeilla sivuston käyttöä myös erilaisilla avusta-
villa teknologioilla tai pelkällä näppäimistöllä, jotta voidaan varmistaa saavutettavuus myös tällai-
silla käyttötavoilla. (Kehitysvammaliitto ry 31.1.2023b.)

Käyttäjätestauksessa täytyy muistaa, että tarkoituksena on verkkosivuston tai palvelun testaami-
nen, ei käyttäjän testaaminen. Jos käyttäjä ei saa suoritettua jotakin tehtävää verkkosivustolla, se
tarkoittaa, että sivustoa täytyy vielä parantaa. (Kehitysvammaliitto ry 31.1.2023b.)

3.4.4 Automaattinen testaus

Verkosta löytyy runsaasti erilaisia saavutettavuuden testaamiseen tarkoitettuja automaattisia työ-
kaluja, joilla voidaan varmistaa ja testata verkkosivuston saavutettavuus (Viljanen 1.9.2019). Kehi-
tysvammaliitto ry:n (31.1.2023a) mukaan testaustyökalujen avulla voidaan parantaa sivustojen

teknistä saavutettavuutta. Testaamiseen on olemassa erilaisia työkaluja, esimerkiksi jo toteutusvaiheessa käytettäviä koodin saavutettavuuden tarkistukseen käytettäviä lisäosia, selaimeen asennettavia selainlaajennuksia, automaattisia tarkistustyökaluja ja kontrastin tarkistamisen työkaluja.

Nämä työkalut tarkistavat sivuston sisällön ja rakenteen ja vertaavat sitä saavutettavuuden kriteereihin, kuten WCAG-standardin mukaisiin ominaisuuksiin. (Harvard University s.a.; Kehitysvammaliitto 31.1.2023a.) Lisäksi joillakin työkaluilla voidaan simuloida erilaisia toimintarajoitteita, kuten näköön, ymmärtämiseen, keskittymiseen tai motoriikkaan liittyviä ongelmia (Celia 2022b).

Automaattiset testaustyökalut ovat hyvä lisä sivustojen saavutettavuuden testaamiseen, sillä ne ovat nopeita ja tehokkaita (Harvard University s.a.). Niiden avulla voidaan tarkistaa esimerkiksi HTML-lähdekoodista saavutettavuuden kannalta kriittiset virheet, ja ne voivat myös havaita monia muita saavutettavuusongelmia, kuten puuttuvia linkkitekstien tai kuvien tekstivastineita, väärin käytettyjä värikontrasteja ja muita sellaisia ongelmia, joita manuaalisella testaamisella voi olla vaikea havaita. (Kehitysvammaliitto ry 31.1.2023a.)

Toisaalta automaattinen testaus ei ole täydellinen ratkaisu yksinään saavutettavuuden arviointiin, sillä automaattitestaustyökalut eivät pysty arvioimaan palvelun tai sivuston käytettävyyttä tai helpokäyttöisyyttä. Ne eivät myöskään kykene välttämättä havaitsemaan kaikkia saavutettavuuden ongelmia, ja ne saattavat antaa vääriä virheitä tai vääriä positiivisia tuloksia. (Harvard University s.a.)

Automaattiset työkalut pohjautuvat useimmiten testausytimiin. Testausytimet eivät ole käyttövalmiita testaustyökaluja, vaan ne ovat ohjelmistokirjastoja, joita voidaan käyttää käyttöliittymän kautta. Nämä kirjastot mahdollistavat teknisten kutsujen lähettämisen verkkosisällön saavutettavuuden arviointia varten. Kun kutsu lähetetään, ohjelmisto palauttaa koneluettavan teknisen raportin, joka sisältää testauksen tulokset. Nämä kirjastot ovat käytännössä komponentteja, joita erilaiset toimijat voivat integroida omiin järjestelmiinsä ja sovittaa omiin käyttötarkoituksiinsa sopiviksi. (Konsi 16.1.2023a.)

Saavutettavuuden testaamiseen tarkoitettuja automaattisia työkaluja voidaan kategorisoida niiden ominaisuuksien mukaan esimerkiksi seuraavasti: kattavaan testaukseen tarkoitetut työkalut, koodin ja sisällön tiettyjen osien tarkistamiseen tarkoitetut työkalut ja värikontrastin tarkistukseen käytettävät työkalut. Kattavaan testaukseen tarkoitetut työkalut yhdistävät useamman eri työkalun verkkosivuston erilaisten ongelmien tarkistamiseen. (Harvard University s.a.)

Kattavaan testaamiseen tarkoitettuja, useita eri ominaisuuksia arvioivia työkaluja ovat muun muassa:

- *Web Accessibility Evaluation Tool - WAVE*: WebAIMin kehittämä yksi vanhimmista saavutettavuuden testaukseen käytettävistä työkaluista. Työkalua voidaan käyttää syöttämällä tarkistettavan verkkosivuston osoite tai asentamalla selaimen lisäosa. Selainlisäosa on saatavilla Google Chromelle, Mozilla Firefoxille ja Microsoft Edgelle. (Harvard University s.a.; Viljanen 1.9.2019.)
- *axe DevTools*: Dequen kehittämä, Axe-Core –testausyttimeen pohjautuva työkalu. Viljasen (1.9.2019) mukaan yksi hyödyllisimmistä automaattisen testauksen työkaluista. Saatavilla lisäosana Google Chromelle, Microsoft Edgelle, sekä rajoitetuin ominaisuuksin Mozilla Firefoxille ja käytön tulee tapahtua ”Tarkista” (engl. Inspect) -tilassa (Harvard University s.a.; Deque Docs s.a.).
- *Accessibility Insights for Web*: Microsoftin luoma lisäosa Google Chrome ja Microsoft Edge –selaimille. Sisältää joitakin automaattisia tarkistuksia, jotka pohjautuvat Dequen Axe-Core-testausyttimeen ja oppaan järjestelmälliseen manuaaliseen testaukseen. (Harvard University s.a.; Viljanen 1.9.2019).
- *Access Assistant*: Level Accessin kehittämä työkalu, joka tarkistaa esimerkiksi otsikot, listat, raamitukset, kuvat ja taulukot ja tarjoaa korjausehdotuksia ongelmatilanteille. Työkalulla saadaan myös tyylitiedostot nopeasti pois käytöstä, jolloin voidaan tarkistaa, toimiiko sivusto ilman tyylejä. (Viljanen 1.9.2019.) Pohjautuu Level Accessin Access Engine –testausyttimeen. Selainlisäosa saatavilla Google Chromelle, Mozilla Firefoxille ja Microsoft Edgelle. (Level Access 31.3.2023.)
- *Google Lighthouse*: Chrome-selaimen sisäänrakennettu avoimen lähdekoodin työkalu, joka pohjautuu Dequen Axe-Core-testausyttimeen. Työkalulla voidaan testata verkkosivuston toimintaa ja saada raportteja sivuston käyttökokemukseen, tehokkuuteen ja suositeltuihin käytäntöihin liittyvistä asioista. Vaihtoehtoisesti se voidaan myös asentaa omalle tietokoneelle ja ajaa komentorivin kautta. Selaimesta sitä käytetään ”Tarkista” -tilassa Lighthouse-välilehdeltä, jolta voidaan valita, mitä testejä sivustolle halutaan ajaa. (Savolainen 14.3.2018.)
- *Salvia*: Suomen aluehallintoviraston vuonna 2023 julkaisema testaustyökalu saavutettavuuden testaamiseen, joka perustuu QualWeb-core –testausyttimeen. Ohjelmisto on saatavilla avoimena lähdekoodina saavutettavuusvalvonnan GitHub-repositoriosta. Ohjelmisto voidaan ottaa käyttöön kloonaamalla avoin lähdekoodi git-versionhallintaohjelman avulla omalle tietokoneelle, tai zip-tiedosto lataamalla. Salvia hyödyntää toiminnassaan Docker-

ohjelmistoratkaisua, joka virtualisoi Salvian suorittamisessa tarvittavat käyttöjärjestelmät ja muut riippuvuudet. (Konsi 16.1.2023b.)

Koodin ja sisällön tiettyjen osien tarkistamiseen käytettyjä yleisiä työkaluja ovat muun muassa:

- *HeadingsMap*: Jorge Rumorson kehittämä työkalu HTML-otsikkojen tasojen tarkistamiseen. Selainlisäosa saatavilla Mozilla Firefoxille ja Google Chromelle. (Harvard University s.a.; Viljanen 1.9.2019.)
- *Nu HTML Checker*: W3C:n työkalu HTML-koodin oikeaoppisuuden tarkistamiseen. Käytetään heidän sivujensa kautta syöttämällä tarkistettavan verkkosivuston osoite. (Harvard University s.a.)

Värikontrastien tarkistamiseen käytettyjä yleisiä työkaluja ovat:

- *Contrast Checker*: WebAIMin kehittämä työkalu, jolla voidaan tarkistaa tummuuskontrasteja (Harvard University s.a.; Kehitysvammaliitto ry 7.2.2023).
- *Contrast Ratio*: Lea Veroun kehittämä työkalu pohjautuen WCAG:n mukaisiin kontrastikriteereihin. Työkalu tarkistaa, mille WCAG:n tasolle käytetty kontrasti on riittävä. (Harvard University s.a.; Kehitysvammaliitto ry 7.2.2023.)

Sisällön ja rakenteen tarkistamisen lisäksi saavutettavuutta voidaan arvioida selaimiin saatavilla simulaattoreilla, joilla voidaan havainnollistaa toimintarajoitteisen käyttäjän käyttökokemusta (Celia 2022b). Tällaisia simulointiin tarkoitettuja ilmaisia selainlaajennuksia ovat muun muassa:

- *Web Disability Simulator*: Metamatrixin luoma selainlisäosa Google Chromelle erilaisten toimintarajoitteiden simuloimiseen. Lisäosalla voidaan simuloida esimerkiksi värisokeutta, lukihäiriötä ja näköongelmia. (Celia 2022b; Viljanen 1.9.2019.)
- *Silktide Toolbar Disability Simulator*: Google Chromeen saatavilla oleva selainlisäosa, jolla voidaan simuloida erilaisia näköön liittyviä ongelmia, kuten lukihäiriötä, myopiaa ja värisokeutta (Celia 2022b).
- *NoCoffee Vision Simulator*: Mozilla Firefoxille ja Google Chromelle saatavilla oleva selainlisäosa, jolla voidaan simuloida erilaisia värien näkemiseen ja näköön liittyviä ongelmia (Celia 2022b).

4 Saavutettavuustyökalujen testaus

Opinnäytetyömme tarkoituksena on vertailla saavutettavuuden arviointiin ja varmistamiseen tarkoitettuja, maksuttomia automaattisia testaustyökaluja toisiinsa. Tutkimus toteutetaan laadullisena tutkimuksena ja tutkimusmenetelmänä käytämme vertailuanalyysiä.

Valitsimme vertailuanalyysin opinnäytetyömme tutkimusmenetelmäksi, sillä halusimme tutkia ja vertailla saavutettavuuden testaukseen tarkoitettuja automaattisia testaustyökaluja toisiinsa. Vertailuanalyysiä käyttämällä pystymme tarkastelemaan testaustyökalujen ominaisuuksia, toiminnallisuuksia ja suorituskkyä objektiivisesti ja systemaattisesti. Tulosten avulla voimme muodostaa yhteenvetoon eri testaustyökalujen vahvuuksista ja heikkouksista. Tätä yhteenvetoa, eli opinnäytetyömme tuloksia, voidaan hyödyntää sopivan testaustyökalun valinnassa verkkosivuston saavutettavuutta testattaessa ja arvioidessa.

4.1 Vertailuanalyysi tutkimusmenetelmänä

Vertailuanalyysi (engl. benchmarking) on tutkimusmenetelmä, jossa verrataan omaa toimintaa toiseen kohteeseen. Vertailukohde on usein paras käytäntö tai menestyvät organisaatiot. Hyvä vertailukohde voi myös mahdollisesti löytyä toiselta alalta. Menetelmällä pyritään tunnistamaan parhaita käytänteitä muista organisaatioista ja soveltamaan niitä omaan toimintaan, jolloin tuotetaan jotakin uutta omaan toimintaan. Tarkoituksena on tunnistaa oman toiminnan heikkouksia ja tavoitella niiden kehittämistä. Vertailuanalyysi on lyhyesti toisilta oppimista ja oman toiminnan kyseenalaistamista. (Ojasalo, Moilanen & Ritalahti 2015, 43.)

Useimmiten vertailuanalyysissä tutkitaan menestyviä organisaatioita, joilta halutaan oppia menestymisen syitä ja mahdollisia hyväksi havaittuja tapoja toimia. Vertailukohteena voivat olla esimerkiksi eri toimialojen organisaatiot, oman yrityksen toiset osastot, kilpailijat ja standardit. Vertailuanalyysi vaatii aluksi kohteen, jossa itse kaipaa kehittämistä, esimerkiksi laadun, tuottavuuden, toimintaprosessien ja työtapojen kehittämistä. Tämän jälkeen etsitään kehittämiskohteelle vertailukumppanit, jotka onnistuvat kyseisessä asiassa paremmin, ja kerätään tietoa heidän onnistumisestaan. Tietoa vertailukohteiden parhaista käytännöistä voidaan hakea erilaisista artikkeleista, kirjoista, tunnusluvuista, mittareista, uutisista, yritysten kotisivuista tai patenttisivustosta. (Ojasalo ym. 2015, 43–44; 186.)

Tiedon hankinnan ja havainnoinnin jälkeen on tärkeää tunnistaa, mistä voidaan oppia uutta, mitä voidaan soveltaa suoraan omaan toimintaan ja mikä omassa toiminnassa vaatii ratkaisun löytämistä. Kaikki vertailukohteiden hyvät käytännöt eivät mahdollisesti ole sovellettavissa suoraan

omaan toimintaan, jos organisaatioiden kulttuurierot ovat suuret. Vertailuanalyysillä voidaan edistää yrityksen kilpailuasemaa, sillä oman toiminnan taso selkiytyy ja tietoisuus kilpailijoiden vahvuuksista ja heikkouksista kiihdyttää kehittämishalukkuutta. Vertailuanalyysissä voi yleensä ilmetä uutta tietoa, jota ei osata ennakoida. (Ojasalo ym. 2015, 186.)

Tuomisen (2016, 8–10) mukaan vertailuanalyysillä on olemassa prosessiaskaaleet, jotka ovat seuraavat:

1. *Kohteen määrittäminen*: Määritellään oman toiminnan menestymisen kannalta tärkeä kehittämiskohde ja mitataan sen suorituskky.
2. *Vertailukohteen etsiminen*: Tavoitteena on etsiä vertailukohteita, jotka ovat parempia tai menestyviä omassa kehittämiskohteessa. Valinta tapahtuu yleensä vertaamalla valittua suorituskkyä ja määrittelemällä menestyksen syitä.
3. *Oman prosessin mittaaminen ja oppiminen*: Mitataan oman prosessin suorituskkyä ja kuvataan toimintatapoja. Suorituskkyä haittaaviin tekijöihin kiinnitetään erityisesti huomiota tässä askeleessa.
4. *Vertailukohteiden prosessien oppiminen*: Mitataan vertailukohteiden prosessien suorituskkyä ja kuvataan toimintatapoja. Kiinnitetään huomiota suorituskkyyen edistäviin tekijöihin. Tässä askeleessa vertaillaan, määritetään suorituskkyerojen syyt ja valitaan parhaat prosessit ja toimintatavat.
5. *Tavoitteiden asettaminen*: Asetetaan kehitystavoitteet, johon tähdätään välittömästi, lyhyellä tähtäyksellä tai pitkällä tähtäyksellä.
6. *Soveltaminen ja käyttöönotto*: Sovelletaan löydettyjä toimintatapoja ja suorituskkyä edistäviä tekijöitä omaan prosessiin. Tässä askeleessa suunnitellaan muutoksia, laaditaan toteutamisajataulu ja sen toteuttaminen.
7. *Vakiinnuttaminen ja uudelleen kehittäminen*: Määritetään pitkän aikavälin tavoite, jota kehitetään edelleen ja aloitetaan uusiksi vertailuanalyysin prosessiaskaaleet. Askeleen tavoitteena on jatkuvalla mittauksella varmistaa asetettujen tavoitteiden saavuttamista ja parhaan mitatun suorituksen yllättämistä.

Vertailuanalyysin tavoitteena on löytää paras menetelmä ja toimintatapa oman organisaation suorituskkyyen kehittämiseksi. Opinnäytetyössämme käytämme vertailuanalyysiä tutkimusmenetelmänä, mutta tavoitteena ei ole oman organisaation kehittäminen, vaan testaustyökalujen vertailu ja

parhaan saavutettavuuden testaustyökalun löytäminen vertailemistamme testaustyökaluista. Vertailuanalyysin prosessiaskelita ei voi suoraan soveltaa tällaiseen tutkimukseen kuin meillä, joten päädyimme soveltamaan vertailuanalyysin prosessiaskelita opinnäytetyössämme seuraavasti:

1. *Testattavien testityökalujen valinta:* Vastaa vertailuanalyysin 1. prosessiaskelta. Määritellään tietyn kategorian testaustyökaluja, joiden suorituskkyä mitataan.
2. *Testitapausten määrittäminen:* Vastaa vertailuanalyysin prosessiaskelta 2., jossa etsitään valitun kategorian testaustyökaluja, joita vertaillaan toisiinsa. Tässä käydään läpi työkalujen ominaisuuksia, suorituskkyä ja mahdollisia menestyksen syitä.
3. *Testien suorittaminen:* Vastaa prosessiaskelia 3.–4., joissa opitaan näiden työkalujen käyttöä ja kuvaillaan niiden toimintatapoja.
4. *Tulosten vertailuanalyysi:* Vastaa osittain 4. prosessiaskelta. Vertaillaan työkalujen ominaisuuksia ja määritellään niiden suorituskkyerojen syitä. Jätämme vertailuanalyysin prosessiaskeliet 5.–6. välistä, sillä emme ole kehittämässä omaa toimintaa, vaan meidän tapauksessamme vertailemme eri työkaluja ja etsimme parasta testaustyökalua.
5. *Johtopäätösten tekeminen:* Vastaa osittain prosessiaskelietta 7, jossa sovelletaan ja tehdään johtopäätöksiä työkalujen ominaisuuksista, miten ne eroavat ja minkälaisiin tarpeisiin testaustyökalut soveltuvat. Lopuksi valitaan paras testaustyökalu, mutta ei aloiteta prosessiaskelietä uusiksi.

Aluksi siis valitsemme valintakriteeristön täyttävät vertailtavat testityökalut, eli määrittelemme vertailukohteet. Esittelemme valintakriteerit, ja niiden pohjalta valitut testaustyökalut kappaleessa 4.3. Tämän jälkeen määrittelemme testitapausten, eli valitsemme testauksessa käytettävän verkkosivuston, jota käytetään jokaisessa työkalussa.

Testitapausten määrittämisen jälkeen suoritamme testauksen kaikilla valituilla työkaluilla ja kirjaamme tulokset kappaleeseen 4.4. Analysoimme testeistä saatuja tuloksia kappaleessa 4.5., jossa vertailemme testaustyökaluja toisiinsa. Vertailuanalyysin avulla muodostamme johtopäätökset ja arviomme siitä, mikä testaustyökalu tutkii verkkosivun saavutettavuutta kattavimmin.

4.2 Tutkimuksen luotettavuuden varmistaminen

Tutkimuksissa virheiden syntymistä pyritään välttämään, mutta tulosten pätevyys ja luotettavuus voivat silti vaihdella. Siksi tutkimuksen luotettavuutta on arvioitava huolellisesti, ja tämä voidaan

tehdä käyttäen monia erilaisia mittaus- ja tutkimustapoja. (Hirsjärvi, Remes & Sajavaara 2014, 231.)

Tärkeitä käsitteitä tutkimuksen luotettavuuden arvioimisessa ovat validiteetti ja reliabiliteetti. Validiteetti tarkoittaa sitä, että tutkimus mittaa todella sitä, mitä sen on tarkoitus mitata, kun taas reliabiliteetti tarkoittaa sitä, että tutkimuksen tulokset ovat toistettavissa ja eivät perustu sattumaan. Tutkimusmenetelmien tarkka valinta ja oikeanlainen käyttö ovat avainasemassa näiden tekijöiden varmistamisessa. (Hirsjärvi ym. 2014, 231–232.)

Laadullisen tutkimuksen luotettavuuden ydinasioita ovat tarkat kuvaukset henkilöistä, paikoista ja tapahtumista. Tutkijoiden tulee myös antaa tarkat selostukset tutkimuksen toteutuksesta kaikissa sen vaiheissa. Tutkijan reflektointi on myös tärkeää, sillä se auttaa tunnistamaan tutkijan omat ennakkooletukset ja asenteet, jotka voivat vaikuttaa tutkimuksen tuloksiin. Tutkijan tulee olla tietoinen omasta roolistaan tutkimuksessa ja siitä, miten se voi vaikuttaa tutkimukseen. (Hirsjärvi ym. 2014, 232.)

Luotettavuuden parantamiseksi on tärkeää kuvata aineiston tuottamisen olosuhteet tarkasti, selkeästi ja totuudenmukaisesti. Esimerkiksi haastattelu- ja havainnointitutkimuksessa tulee kertoa tarkasti, missä ja miten aineisto kerättiin, miten kauan se kesti, sekä mahdolliset häiriötekijät ja virhetulkinnat. Tutkimustulosten tulkinnassa on myös tärkeää osoittaa, millä perusteella tutkija tekee tulkintansa ja mihin hän perustaa päätelmänsä. (Hirsjärvi ym. 2014, 232–233.)

4.3 Vertailuun valitut testaustyökalut

Vertailtavien työkalujen valinta perustui ennalta määritettyihin valintakriteereihin. Vertailuun valittujen työkalujen valintakriteereitä olivat:

- Työkalu suorittaa testin tai testit automaattisesti
- Työkalun tulee testata useampaa kuin vain yhtä verkkosivuston osaa tai osa-aluetta
- Työkalun tulee olla saatavilla maksutta
- Työkalun tulee olla saatavilla selainlaajennuksena
- tai
- Työkalun tulee olla käytettävissä suoraan palveluntarjoajan verkkosivuston kautta URL syöttämällä

tai

— Työkalu on ladattavissa omalle tietokoneelle

Valittujen työkalujen tulee kaikkien olla maksuttomia ja soveltua kappaleessa 3.4.4. kerrottuun kattavaan testaamiseen. Kattavaan testaukseen tarkoitetut työkalut tutkivat useampia osa-alueita samalla kertaa, kuten koodia, sisällön tiettyjä osia ja värikontrastia.

Valintakriteerien perusteella tutkimme, mitkä tietoperustassamme esiteltyt työkalut vastasivat asetettuja kriteereitä. Valitsimme tietoperustastamme tutkimukseemme kuusi työkalua, joita olivat kaikki tietoperustassamme esiteltyt kattavaan testaukseen tarkoitetut työkalut. Tutkimukseemme valitut saavutettavuustestauksen työkalut olivat:

1. Web Accessibility Evaluation Tool – WAVE
2. Axe DevTools
3. Accessibility Insights for Web
4. Access Assistant
5. Google Lighthouse
6. Salvia

Tutkimuksessa on tarkoitus testata vain itse työkaluja ja niiden toimintaa, jossa käytettävä verkkosivusto toimii vain välineenä. Tästä syystä päätettiin, että käytettäväksi valittavalle verkkosivustolle asetetaan ainoastaan seuraava valintakriteeri:

– Verkkosivun tulee soveltua valituilla työkaluilla testattavaksi

Asetetulla valintakriteerillä rajattiin pois one-page- tai single-page –mallilla toteutetut verkkosivut, joita kaikilla testiin valituilla työkaluilla ei pystyttäisi luotettavasti testaamaan.

Valitsimme Helsingin Kaupungin verkkosivun (<https://www.hel.fi/fi>) käytettäväksi työkalujen testauksessa. Valittu verkkosivu vastasi asetettua valintakriteeriä. Helsingin kaupungin verkkosivu valikoitui tutkimukseemme myös ajankohtaisuutensa puolesta (vuonna 2019 käynnistetty digitalisatio-ohjelma), johon johdannossa viittasimme.

4.4 Testaustyökalujen testaus

Suoritimme kaikkien testaustyökalujen testaamisen Google Chrome –selaimella, jota kaikki työkalut tukivat, ja tällä tavoin pystyimme vähentämään selainkohtaisten erojen aiheuttamia mahdollisia virheitä työkalujen toiminnassa. Testaukset suoritettiin tietokoneilla. Mobiililaitteita ei käytetty työkalujen testaamisessa.

Työkalujen testaaminen suoritettiin 8.5.2023 ja 9.5.2023. Testien kulku ja tulokset dokumentoitiin kappaleisiin 4.4.1–4.4.6 sekä sanallisesti, että kuvilla havainnollistaen. Testien tarkka kulku esitetään opinnäytetyön liitteissä 1–6.

Työkalut otettiin käyttöön joko asentamalla selainlisäosa omaan Chrome-selaimeen, tai navigoimalla testaustyökalun omalle verkkosivulle, ja syöttämällä siellä testattavan verkkosivuston osoite. Selainlisäosaa käytettiin käynnistämällä työkalu selaimen ”Laajennukset” -valikosta tai kehittäjätyökalun ”Tarkista” -tilassa, kun oli ensin navigoitu selaimella halutulle testattavalle verkkosivustolle, eli tässä tapauksessa Helsingin Kaupungin sivuille. ”Tarkista” -tila saatiin Chrome-selaimessa auki painamalla F12-näppäintä, tai klikkaamalla hiiren oikeaa painiketta verkkosivulla ja valitsemalla avautuvasta valikosta ”Tarkista”.

Poikkeuksena yllä mainituista käyttöönottotavoista on tutkimuksessamme mukana oleva Salvia-saavutettavuustestaustyökalu, joka ladattiin omalle tietokoneelle aluehallintoviraston GitHub-repositoriosta. Salvian käyttöönotto vaati erilaisia asennuksia, muun muassa Docker-työkalun asennuksen erikseen ja oli muita tutkimuksessamme mukana olevia työkaluja monimutkaisempi käyttöönotossaan. Salvian ja Dockerin asennusten jälkeen Salvia käynnistettiin selaimessa osoitteessa <https://localhost:3000>.

Localhost tarkoittaa tietokoneen omaa verkko-osoitetta, joka viittaa itse tietokoneeseen. Kehittäjät käyttävät localhostia verkkosivustojen testaamiseen tai kehittämiseen paikallisesti omalla tietokoneella. Localhost on turvallinen ja yksityinen testaus- ja kehitysympäristö, joka ei lähetä mitään ulkopuoliselle verkolle. (Juviler 12.7.2022.)

4.4.1 Web Accessibility Evaluation Tool - WAVE

Web Accessibility Evaluation Tool, paremmin tunnettuna WAVE on voittoa tavoittelemattoman WebAIMin kehittämä työkalu saavutettavuuden testaamiseen, joka auttaa tunnistamaan saavutettavuusongelmia automaattisesti. WAVE perustuu WCAG-ohjeistuksiin. (WebAIM – Web Accessibility in Mind s.a.)

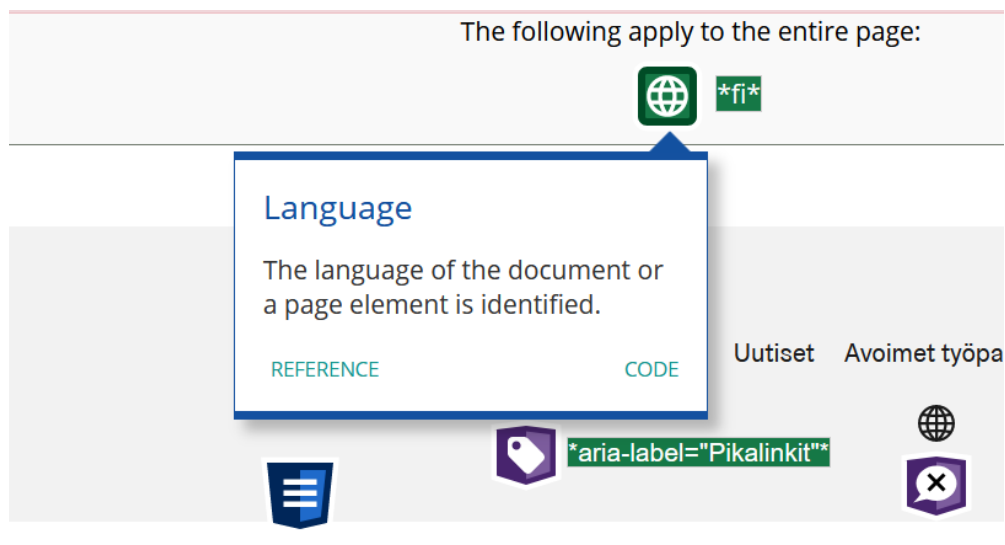
WAVE:n käyttöönotto on helppoa, sillä se vaatii vain verkkosivuston osoitteen. Jos kuitenkin esimerkiksi kehittäjä haluavat käyttää työkalua paikallisen, dynaamisen tai salaisen sivuston testaamiseen, niin WAVE on myös saatavilla selainlaajennuksena Google Chromeen, Mozilla Firefoxiin ja Microsoft Edgeen tätä varten. Selainlaajennus turvaa yksityisyyden ja suojatun saavutettavuusraportoinnin täysin. (WebAIM – Web Accessibility in Mind s.a.)

WAVE korostaa testausraportissaan ikoneilla tietoja ja huomioita, jotka ovat tärkeitä saavutettavuuden arvioinnissa. Työkalu näyttää tiivistelmän havaituista saavutettavuuden ongelmista, verkkosivuston tekstien kontrastivirheistä, varoituksista, toiminnallisuuksista, rakenteellisista elementeistä ja ARIA:sta. Saavutettavuuden ongelmat ja tekstien kontrastivirheet viittaavat siihen, että ne eivät täytä WCAG:n mukaisia saavutettavuuskriteereitä. Käyttäjä joutuu kuitenkin itse lopulta arvioimaan, kuinka työkalun esiin nostamat varoitukset ja huomiot vaikuttavat saavutettavuuteen. Toiminnallisuudet viittaavat elementteihin, jotka parantavat saavutettavuutta ja joita on hyödynnetty oikein. Rakenteelliset elementit ja ARIA:n käytön määrä näkyvät tiivistelmässä sekä verkkosivustolla ikoneina. (WebAIM – Web Accessibility in Mind 9.4.2020, 2:05–5:05 min.)

Testaaminen

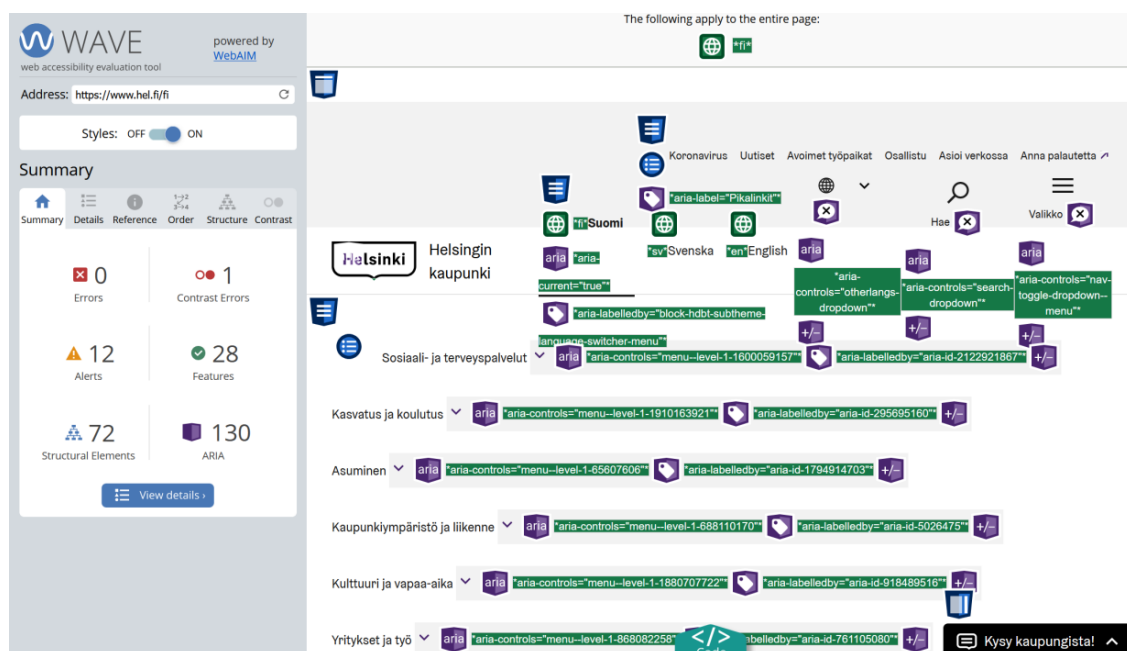
WAVE-työkalua käytettäessä navigoitiin ensiksi työkalun verkkosivulle (<https://wave.webaim.org/>), jossa sivulla näkyvään syöttökenttään syötettiin testattavan verkkosivuston osoite. Sivun oikealla puolella nähtiin testattava verkkosivusto, jossa työkalu esitti erilaisia saavutettavuuteen liittyviä ikoneita. Vasemmassa laidassa nähtiin paneeli, jossa työkalu esitteli ja kuvasi sivustolta löytämiään huomioita.

Punaisilla ikoneilla korostettiin saavutettavuusvirheitä, jotka tulisi huomioida ja mahdollisesti korjata. Vihreillä ikoneilla viitattiin saavutettavuuden toiminnallisuuksiin, jotka paransivat saavutettavuutta ja olivat toteutettu oikein. Työkalun esittämiä ikoneita pystyi klikkaamaan, jolloin näkyville saatiin pieni ikkuna, jossa kerrottiin enemmän ikonien merkityksestä. Esimerkkinä kuvassa 10 klickatusta ikonista, jossa kerrotaan, että lang-attribuutin olemassaolo oli tunnistettu sivustolla.



Kuva 10. – Näkymä klikatusta ikonista

Sivun vasemmalla puolella näkyi heti testin ajamisen jälkeen tiivistelmä verkkosivuston saavutettavuudesta (kuva 11). Työkalu raportoi testattavassa verkkosivussa olleen nolla (0) saavutettavuuden ongelmaa, yksi (1) kontrastivirhe, 12 varoitusta, 28 toiminnallisuutta, 72 rakenteellista elementtiä ja 130 ARIA roolia, tyyppiä tai ominaisuutta. Tiivistelmä (engl. summary)-välilehden vierestä päästiin navigoimaan muihin välilehtiin, kuten yksityiskohtiin (engl. details), suositukseen (engl. reference), järjestykseen (engl. order), rakenteeseen (engl. structure) ja kontrastiin (engl. contrast). Tiivistelmän yläpuolella nähtiin on/off-kytkin, jolla voitiin ottaa tyylit pois päältä, jolloin sivuston rakenteen pääsi näkemään ilman tyyli tiedostoa.



Kuva 11. – WAVE-työkalun alku näkymä

Yksityiskohdat-välilehdellä esitettiin tarkemmin havaitut kategoriat, eli saavutettavuusongelmat, kontrastivirheet, varoitukset, toiminnallisuudet, rakenteelliset elementit ja ARIA. Työkalulla pystyi myös katsomaan verkkosivuston lähdekoodin ja havaittujen virheiden kohdat lähdekoodissa. Havaittuja kontrastivirheitä työkalu löysi yhden, mutta se sijaitti verkkosivulta piilotetussa elementissä, joka ei ollut ollenkaan näkyvässä käyttäjille. Tämä kontrastivirhe ilmeni, kun tyylitiedosto otettiin työkalun avulla pois käytöstä ja klikattiin lähdekoodi esiin sivuston alapuolella olevaan `</> Code` -kohtaan. Lähdekoodista nähtiin, että tämä piilotettu elementti oli otsikko `<h2>` -elementti ja sen luokka-attribuutiksi oli annettu "visually-hidden", jolla elementti oli piilotettuna. Tämän voidaan olettaa olleen tarkoituksellista, joten virhettä ei välttämättä tarvitse ottaa huomioon.

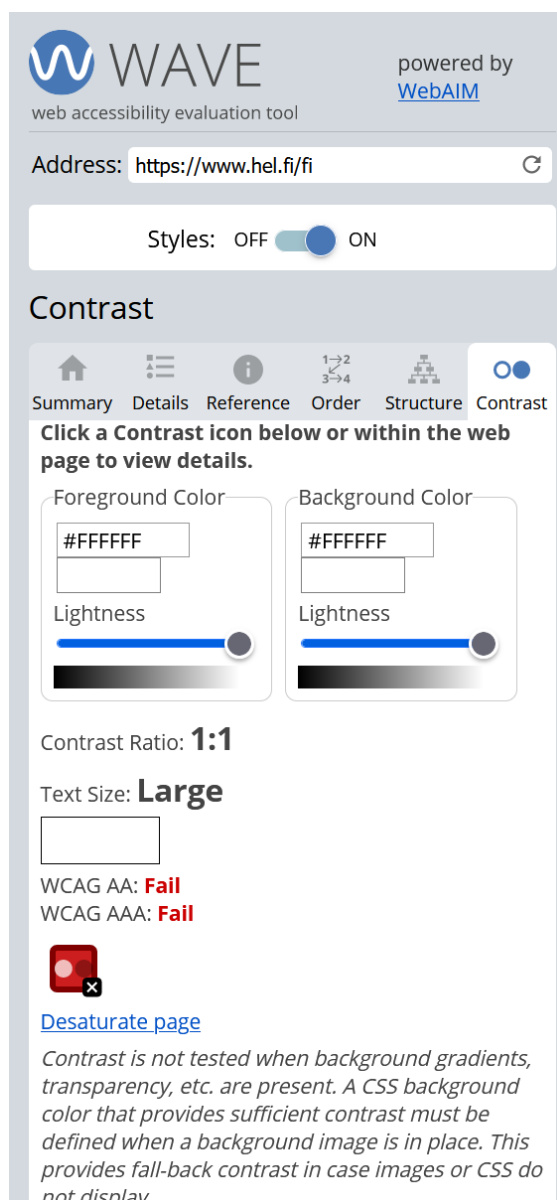
Suositus-välilehdeltä saatiin lisätietoa valitusta huomiosta, jossa selitettiin, mitä kyseinen saavutettavuuden huomio tarkoittaa, miksi sillä on väliä ja mitä tälle huomiolle tulisi tehdä. Välilehden lopussa viitattiin vielä WCAG-ohjeistuksen A- tai AA-tasoon.

Varoituksia työkalu havaitsi 12 kappaletta, joista yksi liittyi `<noscript>` -elementin käyttöön ja loput alleviivattuun tekstiin. Alleviivatut tekstit tarkoittavat yleensä linkkejä, joten varoitus huomauttaa tästä, jotta ihminen tarkistaisi varoituksen. Toiminnallisuuksia oli 28 kappaletta ja näitä kuvattiin raportissa vihreillä ikoneilla, eli ne paransivat verkkosivun saavutettavuutta. Nämä 28 parantavaa saavutettavuuden toiminnallisuutta oli ryhmitelty neljään kategoriaan, joista viisi (5) oli alt-tekstejä, kolme (3) tyhjiä alt-tekstejä, kaksi (2) lomakkeen nimilappua (engl. form label) ja 18 kappaletta kieliin liittyviä tekijöitä, kuten kielen tunnistamista tai lang-attribuutin käyttöä.

Järjestys-välilehdellä listattiin navigointielementtejä järjestyksessä. Järjestyksessä tuotiin esille yhteensä 83 navigointielementtiä, joita olivat link tai button -elementit. Elementtien roolien ja järjestyksen lisäksi järjestyksessä nähtiin joidenkin elementtien kohdalla, ettei kyseisillä elementeillä ole saatavilla olevaa nimeä. Kyseiset elementit olivat klikattavia ikoneita, joilla ei ollut nimiä.

Rakenteellisia elementtejä testatulla verkkosivulla työkalu havaitsi 72 kappaletta, ja ne näkyivät rakennevälilehdellä. Näillä tarkoitettiin rakenteellisten elementtien käyttöä, kuten tässä tapauksessa otsikko- (h1, h2, h3), listaus, ylätunniste, hakutoiminto, navigointi, alatunniste ja sivuhuomautus -elementtejä. 130 ARIA:an liittyvää tekijää tarkoittivat ARIA roolien, tyyppien ja ominaisuuksien käyttöjen määrä.

Viimeisenä välilehtenä esitettiin kontrasti, josta saatiin esille tarkemmat tiedot kontrastivirheen kontrastisuhteesta, kun kontrastivirheen ikonia painettiin (kuva 12). Välilehden liukusäätimistä päästiin säätämään kyseisen kontrastivirheen sisältävän elementin tekstin tai taustan väriä. Säädintä käyttämällä nähtiin, mitkä väreistä läpäisi WCAG-ohjeistuksen AA- tai AAA-tasot.



Kuva 12. – Kontrastivälilehden näkymä

Yhteenvedona WAVE-työkalu korosti ikoneilla saavutettavuuteen liittyviä huomioita testattavalla verkkosivulla. Työkalu löysi saavutettavuusongelmia liittyen kontrastiin, sekä huomioitavia varoituksia, joista käyttäjän täytyi itse päätellä, vaativatko niissä esiin nostetut asiat korjaamista. Työkalu antoi myös ehdotuksia, miten sen löytämiä, saavutettavuuteen liittyviä ongelmia voisi korjata. WAVE-työkalu näytti saavutettavuusongelmien lisäksi myös sellaiset toiminnallisuudet, rakenteet ja ARIA käyttöjen määrät, jotka paransivat verkkosivun saavutettavuutta. Loppujen lopuksi jäi kuitenkin käyttäjän vastuulle arvioida, olivatko työkalun korostamat saavutettavuutta parantavat kohdat oikeasti saavutettavia.

4.4.2 Axe DevTools

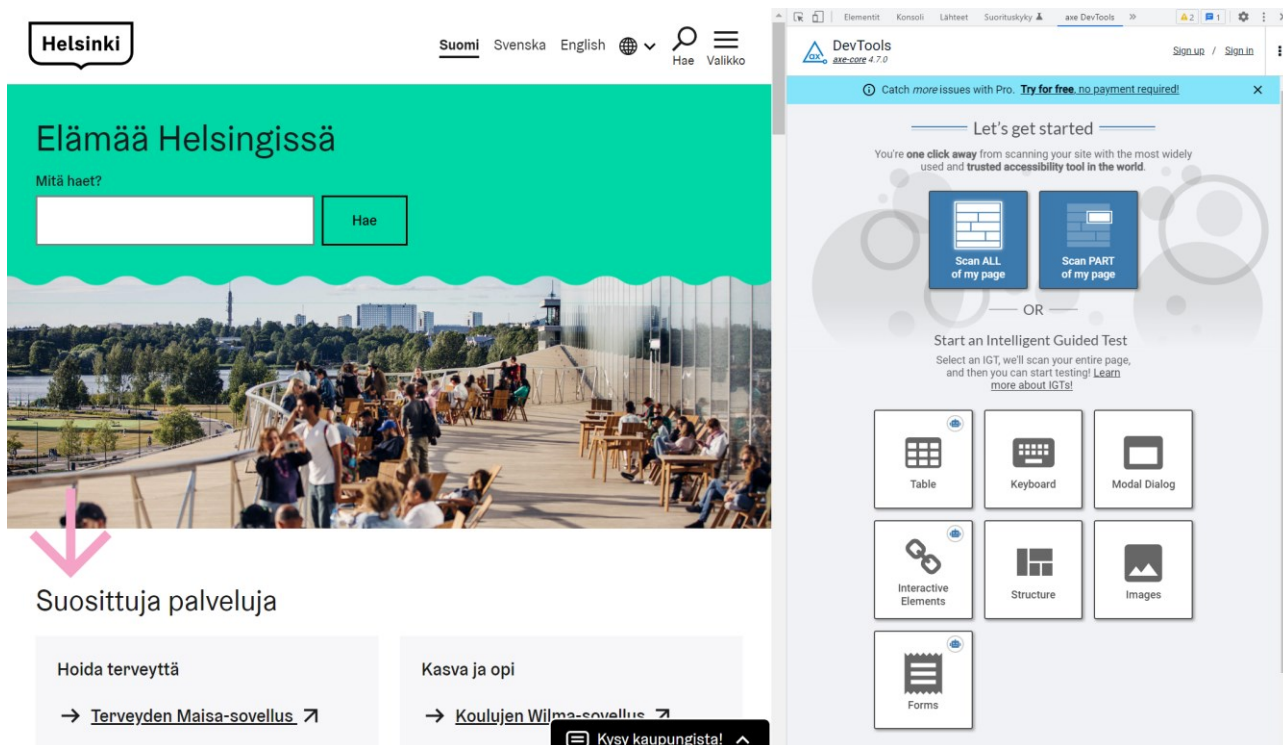
Axe DevTools on Dequen kehittämä ja ylläpitämä ilmainen selainlisäosa Google Chromelle ja Microsoft Edgelle. Saatavilla ovat myöskin rajoitettu versio Mozilla Firefoxille ja maksulliset, kattavammat Pro ja Enterprise –versiot, joita voidaan hyödyntää laajemmin ja monipuolisemmin, esimerkiksi jo koodinkirjoitusvaiheessa saavutettavuusongelmien tunnistamiseen. (Deque s.a.)

Deque (s.a.) mukaan Axe DevTools on nopea, kevyt ja tehokas testaustyökalu, joka pohjautuu Dequen luomaan, ACT-testaussäännöt sisältävään Axe-Core –testausytimeen. Axe DevToolsilla voidaan testata koko näkyvää verkkosivua, tai jotakin sivun osaa. Työkalu perustuu WCAG 2.1. - ohjeisiin, ja työkalun ilmainen selainlisäosa löytää ja tunnistaa itsestään noin 57 % saavutettavuusongelmista. Dequen mukaan Axe DevTools tunnistaa saavutettavuusongelmia lähes kolminkertaisesti enemmän, kuin muut vastaavat automaattiset työkalut. Deque lupaa työkalun esittelyssä myös, että Axe DevTools ei anna vääriä positiivisia tuloksia, eli vääriä virheitä ja on tässä suhteessa ainoa laatuaan. (Deque s.a.)

Testaaminen

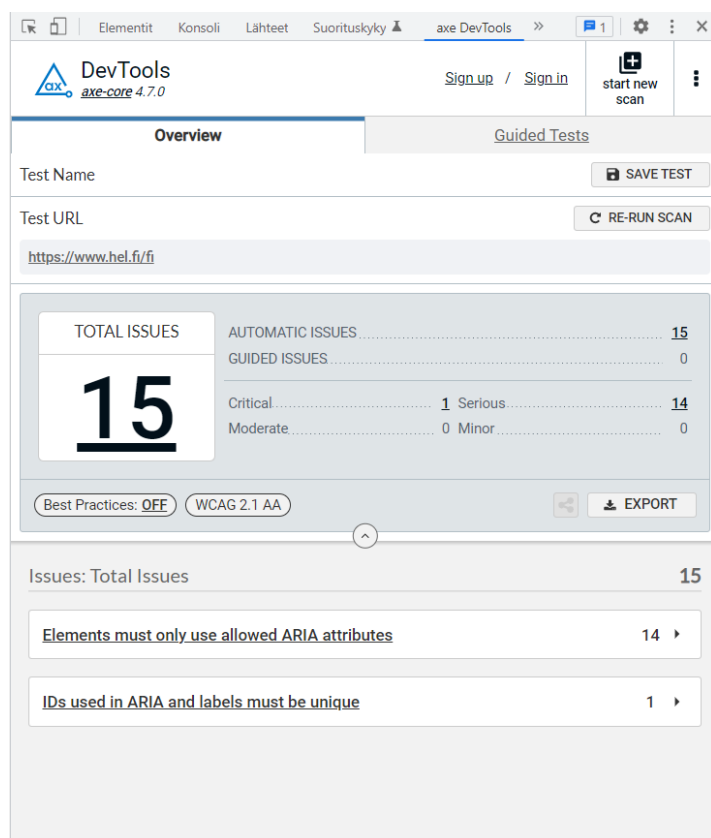
Axe Devtools-työkalu avattiin selaimen ”Tarkista”-tilasta, jossa kehittäjätyökalusta valittiin axe Devtools -välilehti. Ennen työkalun käyttöönottoa tuli valita käyttäjärooli ja hyväksyä käyttöehdot. Valittavissa olevia rooleja olivat testaaja, kehittäjä, suunnittelija, saavutettavuuden asiantuntija, johtaja tai muu rooli. Testasimme eri rooleja, mutta roolilla ei ollut merkitystä testaustuloksiin tai työkalun käyttöön, joten päädyimme valitsemaan rooliksi muu rooli.

Työkalun alunäkymässä (kuva 13) pääsi heti valitsemaan, tutkiiko työkalu verkkosivuston kokonaan vai vain joitakin osin. Näiden alla on myös muita älykkäästi ohjattuja testauksia (engl. guided tests), mutta näiden käyttämiseen tarvittiin axe DevTools Pro -versio, joka oli maksullinen. Tämän työkalun tutkimiseen valitsimme vaihtoehdon, jossa verkkosivusto testataan kokonaan.



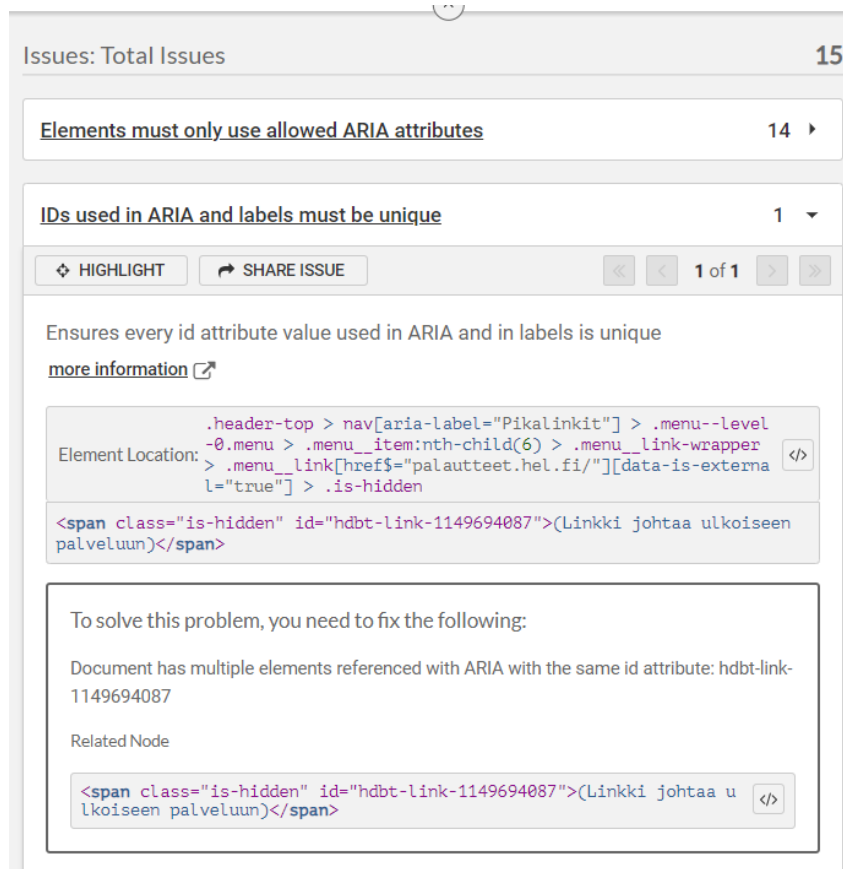
Kuva 13. – Axe DevTools alkunäkymä

Testin ajamisen jälkeen työkalu näytti raportin testituloksista. Testissä havaitut ongelmat oli jaettu kriittisiin, vakaviin, kohtuullisiin ja vähäisiin ongelmiksi. Työkalusta nähtiin myös, että se noudatti WCAG 2.1 -ohjeistuksen AA-tasoa. Axe DevTools toi koko verkkosivuston testauksessa esiin yhteensä 15 saavutettavuusongelmaa (kuva 14). Yksi (1) ongelmista oli kategorisoitu kriittiseksi ja 14 muuta vakaviksi ongelmiksi, jotka liittyivät erityisesti ARIA:n käyttöön.



Kuva 14. – Tulokset axe DevToolsin kokonaan verkkosivuston skannaamisesta

Työkalu jakoi löytämänsä saavutettavuusongelmat kategorioittain ja kategoriaa klikkaamalla saatiin avattua tarkempi kuvaus ongelmasta. Korosta (engl. highlight) -nappia painamalla työkalu korosti verkkosivulla ongelman, joka on kyseessä. Työkalun löytämä kriittinen ongelma verkkosivulla johtui useassa kohdassa toistuvasta samasta ARIA:n id-attribuutista (kuva 15). Vakaviksi ongelmiksi lisättiin ARIA-attribuuttien käyttö, jolla tarkoitettiin ARIA-attribuutin käyttöä -elementissä ilman roolia.



Kuva 15. – Lisätietoa yksittäisestä saavutettavuusongelmasta

Ongelmien kuvauksen alapuolella esitettiin lyhyt ehdotus ongelman ratkaisemiseen. Alleviivattua ”more information” -tekstiä klikkaamalla avautui Deque University:n sivu selaimen uudelle välilehdelle, jossa kerrottiin yksityiskohtaisemmin kyseisestä ongelmasta.

Yhteenvetona Axe DevTools havaitsi Helsingin kaupungin etusivulta pääosin ARIA:an liittyviä saavutettavuusongelmia. Osa työkalun toiminnoista oli maksullisia ja esimerkiksi testauksen tuloksien lataaminen, ja saavutettavuusongelmien jakaminen eteenpäin vaati käyttäjän kirjautumista palveluun. Siitä huolimatta ilmainen axe DevTools-versio antoi jokaisesta ongelmasta lisätietoja ja mahdollisen ratkaisun kyseisen saavutettavuusongelman korjaamiseen. Tuloksista pääsi myös yksityiskohtaisempiin saavutettavuusongelmien tietoihin ja työkalussa myös mainittiin, minkä WCAG-ohjeistuksen tasoa työkalu noudatti testeissään.

4.4.3 Accessibility Insights for Web

Accessibility Insights for Web on Microsoftin kehittämä työkalu verkkosivuston saavutettavuusongelmien havaitsemiseen, joka on saatavilla selainlisäosana Google Chromelle ja Microsoft Edgelle.

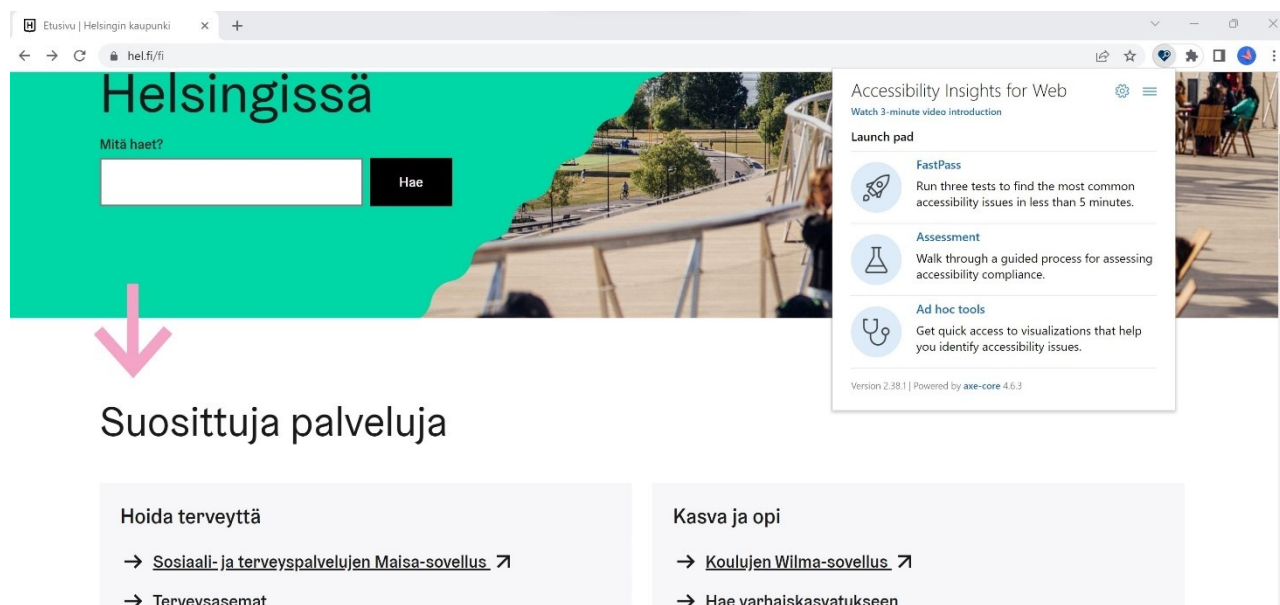
Sitä voidaan käyttää Windows, Linux ja MacOS -käyttöjärjestelmillä. Työkalun automaattiset testit perustuvat Dequen Axe-Core –testausytimeen. (Microsoft s.a.)

Accessibility Insights for Web -työkalua voidaan käyttää kahdella tavalla. Työkalun FastPass –ominaisuus auttaa testaajia ja kehittäjiä nopealla kaksivaiheisella prosessilla tunnistamaan suurimpia ja yleisimpiä saavutettavuusongelmia verkkosivustoilla. FastPass –prosessissa työkalu tarkistaa automaattisesti noin 50 saavutettavuusvaatimusta ja ilmoittaa selkeästi visuaalisen havainnollistamisen kautta esimerkiksi ongelmat, jotka vaikeuttavat näppäimistöselaamista. (Microsoft s.a.)

Toinen työkalun ominaisuus, Assessment, auttaa Microsoftin mukaan ketä tahansa, joka osaa HTML-merkintäkieltä tarkistamaan, että verkkosivusto tai verkkosovellus on yhteensopiva verkkosisällön saavutettavuusohjeistuksen (WCAG) 2.1. AA-tason kriteerien kanssa. Assessment –ominaisuus sisältää automaattiset tarkistukset noin viidenkymmenen saavutettavuusvaatimuksen testaamiseen ja kattavat ohjeet manuaalista testaamista varten. Manuaalisten testien osalta Assessment tarjoaa visuaalista apua ja tunnistaa testitapauksia. (Microsoft s.a.)

Testaaminen

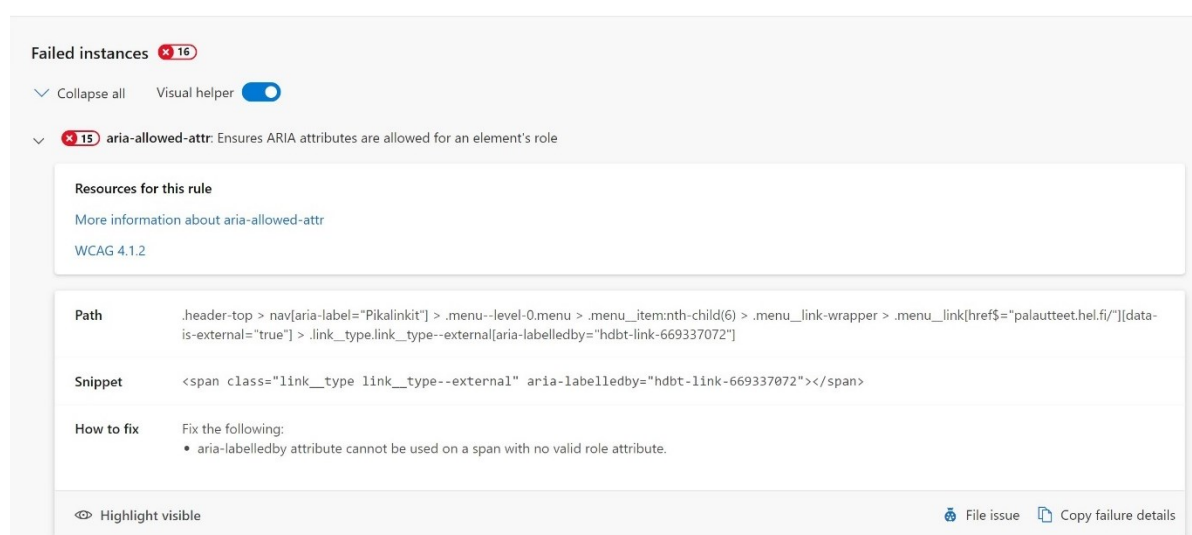
Accessibility Insights for Web -selainlaajennus käynnistettiin Helsingin kaupungin sivuilla selaimen ”Laajennukset”-valikosta, jolloin selaimen yläosaan oikealle avautui työkalun paneeli. Paneelista voitiin valita testin suoritustavaksi joko FastPass, Assessment tai Ad hoc tools -ominaisuus (kuva 16).



Kuva 16. – Accessibility Insights for Web -työkalun ominaisuuden valinta.

Valitsimme Accessibility Insights for Web -työkalun testaamiseen FastPass-tavan, sillä halusimme testata vain automaattitarkistusominaisuuden. Kun testin suoritustapa oli valittu, aukesi työkalu uuteen selainikkunaan. FastPass-tarkistuksessa oli kolme askelta, joita olivat automaattiset tarkistukset ja kaksi muuta liittyen manuaaliseen testaamiseen. Ensimmäinen askel, eli automaattinen tarkistus oli suoritettu heti uuden selainikkunan avautuessa, ja näkyvillä oli listaus epäonnistuneista tapauksista (engl. failed instances), joita löytyi yhteensä 16 kappaletta. Nämä hylätyt testit olivat eroteltu kahteen kategoriaan.

Jokainen työkalun automaattisesti löytämä saavutettavuusongelma oli omana osionaan, jossa työkalu kertoi ensimmäisenä kyseessä olevan elementin polun (engl. path), seuraavaksi kyseisen osan lähdekoodista kohdassa "Snippet", elementin relatiivisen polun ja lopuksi keinon tai keinot, joilla ongelma voitiin korjata (kuva 17). Osion alareunasta klikkaamalla "Highlight visible" -kohtaa työkalun visuaalinen avustaja (eng. visual helper) korosti punaisella huutomerkki-ikonilla testatussa verkkosivustossa kohdan, johon virhe liittyi.

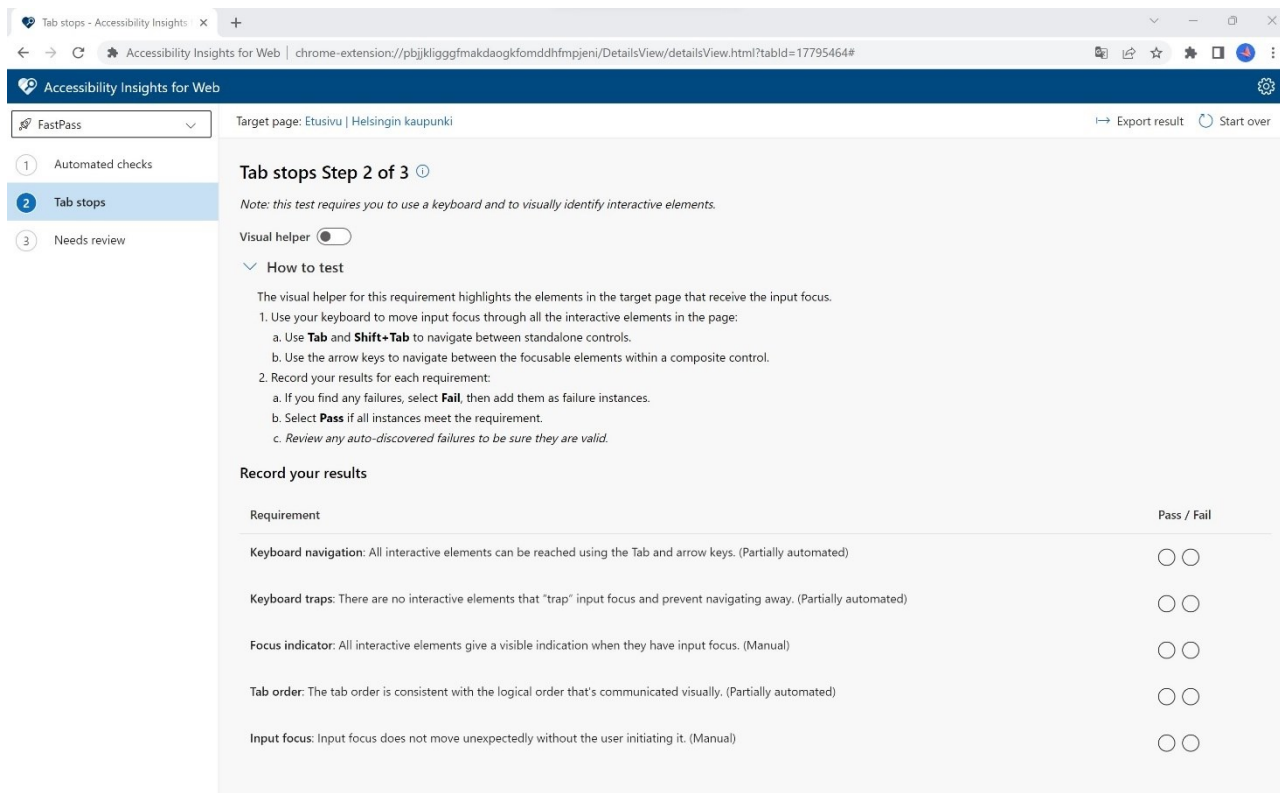


Kuva 17. – Näkymä työkalun testituloksista

Ensimmäisen kategorian 15 hylättyä testiä liittyivät ARIA-attribuutteihin. Kategorian alussa kerrottiin suora linkki kyseiseen WCAG 2.1. kriteeriin, johon ongelmat liittyivät. Linkitettyinä näkyi myös Accessibility Insights:n oma verkkosivu, jolla aria-allowed-attr –ongelmasta ja sen korjaamisesta kerrottiin tarkemmin. Toinen kategoria, josta löytyi yksi (1) hylätty testitapaus, oli id-attribuuttien arvojen toistuminen eri elementeillä. Virhe liittyi raportin mukaan useisiin ``-elementteihin, joilla ei ollut yksilöllisiä id-arvoja.

Automaattisen tarkistuksen lisäksi FastPass-työkalulla tulisi edetä myös askeleisiin 2 ja 3, joissa käyttäjä tarkistaa manuaalisesti mahdollisia saavutettavuusongelmia. Kohdassa 2 käyttäjän tulee

testata näppäimistötestauksen avulla interaktiiviset elementit, jotka työkalun visuaalinen avustaja näyttää verkkosivulla (kuva 18). Kohdassa 3 käyttäjä suorittaa kontrastintarkistuksia manuaalisesti, tai Accessibility Insights-työkalun toisia versioita käyttämällä.



Kuva 18. – Näkymä testaustyökalun toisen askeleen manuaalisesta testauksesta

Yhteenvedona Accessibility Insights for Web-työkalu löysi Helsingin kaupungin etusivulta pääasiassa ARIA:an liittyviä saavutettavuusongelmia. Työkalu antoi jokaisen ongelman kohdalla kuvauksen kyseisestä ongelmasta sekä kohdan, jossa ongelma oli ja korjausehdotuksen. Testituloksista päästiin tarvittaessa näkemään ongelmaa koskevat perustelut ja WCAG-saavutettavuusvaatimukset, jotka liittyivät työkalun havaitsemiin ongelmakohtiin.

4.4.4 Access Assistant

Access Assistant on Level Accessin luoma työkalu, joka on saatavilla maksuttomana selainlaajenuksena Google Chromelle, Mozilla Firefoxille ja Microsoft Edgelle. Työkalun avulla voidaan testata manuaalisesti, sekä automaattisesti helppokäyttöisiä testejä suoraan selaimessa. Testaustyökalu perustuu Level Accessin Access Engine –testausytimeen, joka sisältää laajan joukon JavaScript-pohjaisia automatisoituja testejä. (Level Access 31.3.2023.)

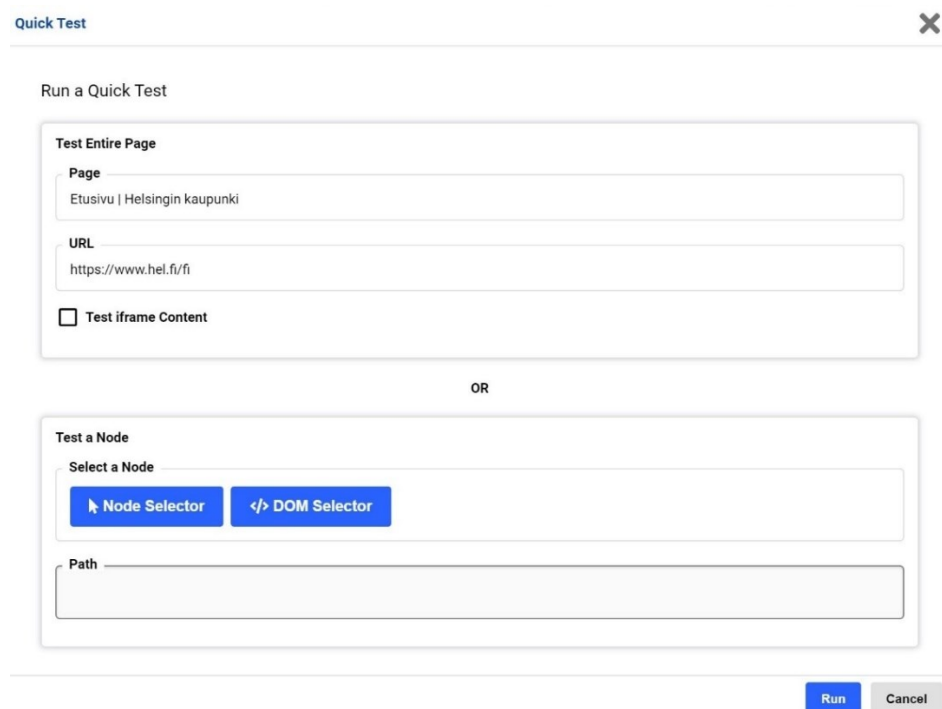
Access Engine –testausydin sisältää yli 250 täysin automatisoitua testiä. Työkalulla voidaan tutkia saavutettavuuden osa-alueita tätäkin enemmän puoliautomaattisesti, ja manuaalisesti testaamalla. Työkalu tunnistaa verkkosivuston tai –sovelluksen yleisimmät ja suurimmat saavutettavuusongelmat ja tarjoaa ohjeita niiden korjaamiseksi. (Access Assistant s.a.)

Level Accessin mukaan Access Assistantin käyttö on nopeaa. Testaustulokset tallentuvat automaattisesti Level Accessin Accessibility Management Platform (AMP) -alustalle, josta niitä voidaan tarkastella seuranta- ja hallintaa varten. (Level Access 31.3.2023.)

Testaaminen

Access Assistant käynnistettiin Helsingin kaupungin sivuilla selaimen ”Laajennukset”-valikosta. Tällöin työkalu avautui omaan ikkunaan ja avautuvasta näkymästä voitiin valita joko Quick Test automaattista testaamista, tai Preview Modes-testaus manuaalista testaamista varten. Valitsimme Quick Testin -vaihtoehdon automaattista testaamista varten.

Quick Test näkymässä testattavaksi haluttavan verkkosivuston URL oli jo valmiiksi syötettynä, joten testin suorittamiseen voitiin siirtyä Run-nappia painamalla. Testin tulosten kannalta ei ollut merkitystä, valittiinko käyttöliittymässä tarjottu ”Test iframe Content” -valinta käytettäväksi testissä, vaiko ei (kuva 19).



Kuva 19. – Näkymä Quick Test -ikkunasta ennen testin suorittamista

Testin suorittamisen jälkeen Access Assistant listasi havaitut saavutettavuusongelmat allekkain. Ongelmat olivat eriteltynä omiksi osioikseen ja niiden otsikkoina toimivat kuvaukset havaitusta ongelmasta tai siihen ehdotetusta korjauksesta. Otsikoita voitiin klikata, ja tällöin tarkempi selitys tai kuvaus ongelmasta avautui näkyville (kuva 20).

Quick Test Results

Quick Test Results

Page

Etusivu | Helsingin kaupunki

URL

https://www.hel.fi/fi

Provide valid, concise, and meaningful alternative text for image buttons

Tag	Description	Actions
<button/>	<button value="Submit" type="submit" class="hds-button hds-button--primary hel-fi-search__submit-button">Hae</button>	

Ensure heading level matches the heading's visual importance/level

Tag	Description	Actions
<article/>	<article class="node--type-landing-page node--view-mode-full" role="article"><div class="paragraph-content components components--upper"><div class="compo...	

Ensure all active elements receive keyboard focus or can be activated with the keyboard

Tag	Description	Actions
<a/>		

Ensure markup documents contain well-formed elements

Kuva 20. – Näkymä testin tuloksista

Havaituista ongelmista näytettiin käyttäjälle esimerkit suoraan lähdekoodista, taulukoituna osioiden Tag, Description ja Actions -alle (kuva 20). Tag-sarakkeessa näytettiin kyseinen elementti, jota ongelma koski. Description-kohdassa annettiin esimerkki suoraan lähdekoodista, ja Actions-sarakkeen alla nähtiin sininen silmäikoni. Silmäikonia klikkaamalla saatiin auki tarkempi selitys kyseisestä tapauksesta.

Ensimmäisenä saavutettavuusongelmana Access Assistant listasi alt-tekstin puuttumisen <button>-elementiltä. Toisena ongelmana mainittiin, että <article />-elementin otsikon tulee vastata sen visuaalista tärkeyttä tai tasoa. Kolmantena havaittuna ongelmana listataan, että kaikki aktiiviset

elementit tulisi olla myös näppäimistöllä aktivoitavissa tai tarkennettavissa. Neljäs havaittu ongelma on, että merkinäkielen elementtien oikeaoppisuus ja muotoilu tulisi tarkistaa.

Viidentenä Access Assistant listaa useiden elementtien ARIA roolien, tyyppien ja ominaisuuksien oikeaoppisuuden tarkistamisen. ARIA-ominaisuuksien osalta työkalu ei testattavan sivuston kohdalla erotellut, oliko kyseessä pelkästään tarkistettava asia, vaiko saavutettavuuden kannalta huomattu konkreettinen ongelma tai virhe. Esimerkiksi useiden ``-elementtien osalta työkalu raportoi tarkemmassa kuvauksessa, että elementille oli annettu `aria-label`-attribuutti, mutta ei kertonut käyttäjälle, onko tämä saavutettavuusvirhe, vai pelkkä huomio.

Yhteenvedona Access Assistant nosti Helsingin kaupungin etusivulla esiin saavutettavuusongelmia liittyen muun muassa väriin tai puuttuviin otsikotasoisiin, saavutettavuuden kannalta haastaviin tai puuttuviin `alt`-teksteihin, sekä elementtien väriin ARIA rooleihin, tyypeihin ja ominaisuuksiin. Työkalu kertoi jokaisen ongelman kohdalla tarkemman kuvauksen havaitsemastaan tai epäilemättään saavutettavuusongelmasta, mutta ei kertonut suoraan, mitä toimenpiteitä käyttäjän oli tehtävä ongelman korjaamiseksi.

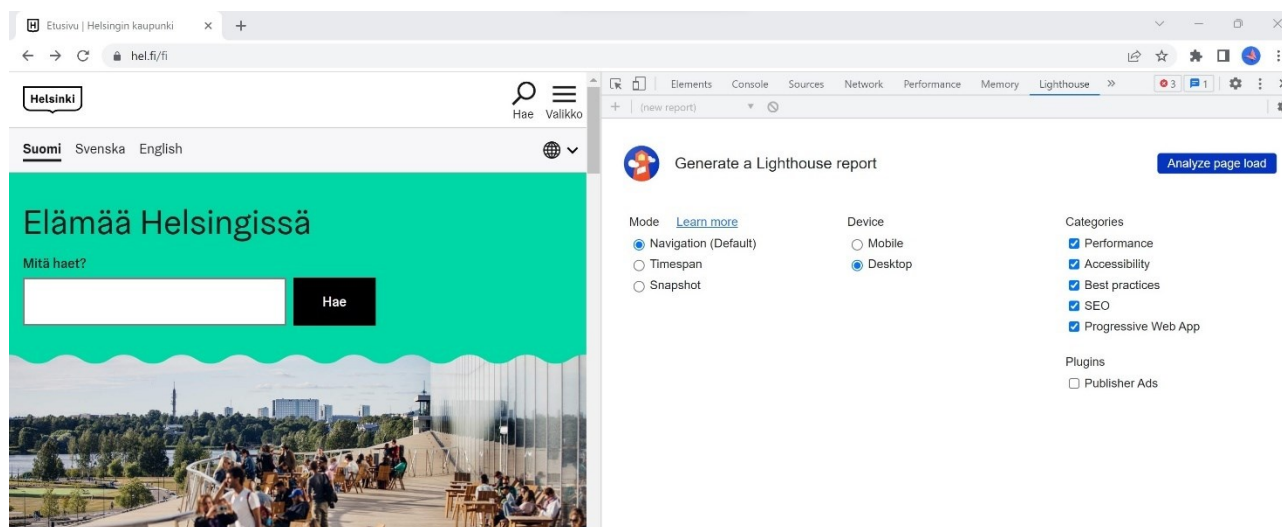
4.4.5 Google Lighthouse

Google Lighthouse on Google Chrome -selaimen sisäänrakennettu avoimen lähdekoodin työkalu, jolla voidaan testata verkkosivuston toimintaa. Lighthouse on saatavilla suoraan Chrome-selaimessa "Tarkista" -tilassa, eli Chromen kehittäjätyökalussa Lighthouse-välilehdeltä. Lighthousea on myös mahdollista käyttää komentorivin kautta, PageSpeed Insights -verkkosivuston käyttöliittymän kautta tai selainlaajennuksena Google Chrome -selaimessa. Selainlaajennusta ei suositella käytettäväksi ilman erityistä syytä, sillä Chromeen kehittäjätyökalun sisältämän Lighthousen avulla voidaan testata myös paikallisia ja todennettuja sivustoja, kun taas selainlaajennuksella ei voida. (Google Developers s.a.)

Lighthouse -työkalulla voidaan tarkistaa mitä tahansa verkkosivustoja, kuten julkisia ja todennettuja sivustoja. Työkalu auttaa tarkistamaan eri kategorioita verkkosivustojen suorituskyvyn, saavutettavuuden, parhaiden käytäntöjen noudattamisen, hakukoneoptimoinnin (SEO) ja progressiivisen verkkosovelluksen (PWA). Lighthousen käyttöönoton kerrotaan olevan helppoa. Google Chrome -selaimessa siirrytään verkkosivustoon, jota halutaan tarkistaa, ja avataan Chromen kehittäjätyökalun Lighthouse-välilehti. Välilehdessä voidaan valita, mitä kategorioita tarkistetaan ja lopuksi klikataan "Analyze page load"-nappia, jolloin saadaan yksityiskohtainen raportti halutuista kategorioista, kuten verkkosivuston saavutettavuudesta tai suorituskyvystä. (Google Developers s.a.)

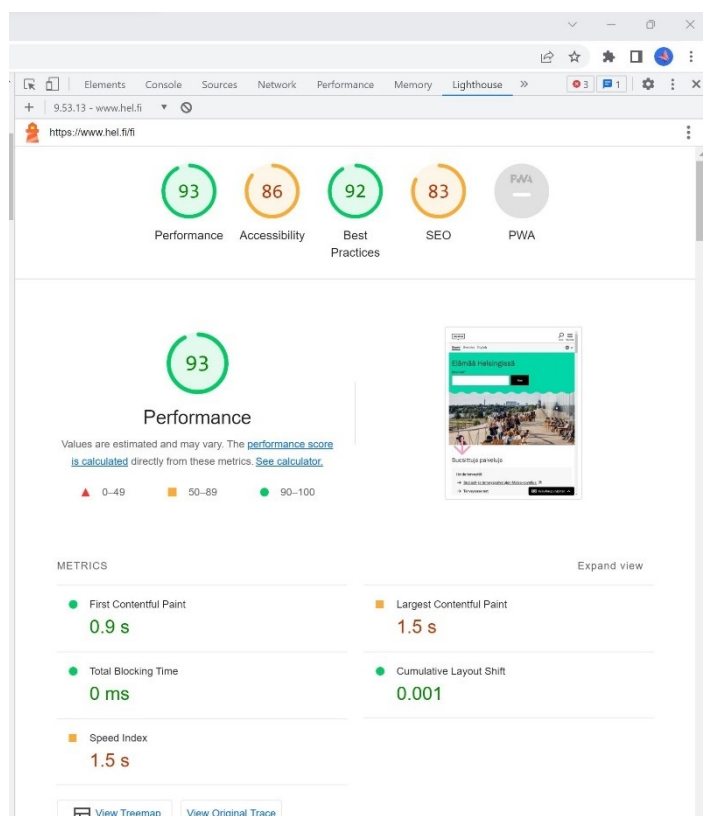
Testaaminen

Lighthouse-työkalu käynnistettiin Helsingin kaupungin etusivulla F12-nappia painamalla, jolloin selaimen kehittäjätyökalut avautuivat näkyville. Kehittäjätyökaluista navigoitiin “Lighthouse”-välilehdelle, jonka alkunäkymästä päästiin valitsemaan testin suoritustavat. Valitsimme testin suoritettavaksi “Navigation”-tilassa, joka oli oletusasetuksena työkalussa. Navigation-tilassa Lighthouse analysoi yhden sivun kerrallaan, verkkosivun kyvykkyyden toimia progressiivisena verkkosovelluksena (PWA) ja sivun saavutettavuuden. Testi käynnistettiin käyttöliittymän oikeassa ylä laidassa olevasta sinisestä napista “Analyze page load”, joka nähdään kuvassa 21.



Kuva 21. – Alkunäkymä ennen testin suorittamista

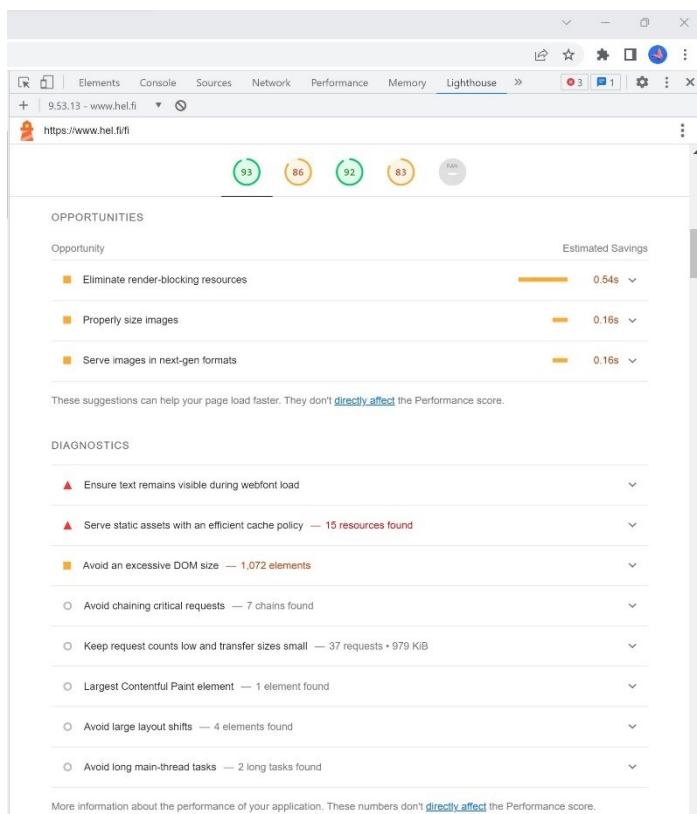
Kun Lighthouse on suorittanut verkkosivun testaamisen, saadaan näkyville testin tulokset sisältävä yhteenvetoraportti. Lighthouse pisteyttää testattavan verkkosivun 0–100 asteikolla, ja sisällyttää raporttiin punaisella heikot ominaisuudet, oranssilla parannusta vaativat ominaisuudet ja vihreällä hyvät ominaisuudet (kuva 22). Värikoodaus jatkuu läpi koko raportin visuaalisesti havainnollistaen testaustuloksia. Testattava verkkosivu saa testaushetkellä kokonaissuorituskykypisteitä 93, saavutettavuuspisteitä 86, parhaista käytännöistä 92 pistettä ja hakukoneoptimoinnista (SEO) 83 pistettä. Progressiivisen verkkosovelluksen (PWA) pisteitä ei ole laskettu.



Kuva 22. – Näkymä suorituskykytestituloksesta

Ensimmäisenä osiona testausraportissa on suorituskyky (engl. performance) -osio. Tässä kohdassa Lighthouse ilmoittaa testaamansa verkkosivun kokonaissuorituskyvyn, joka on muodostettu erilaisten mittarien perusteella. Nämä suorituskykyyn vaikuttavat mittarit mittaavat sitä, kuinka helpokäyttöinen ja saavutettava verkkosivusto on käyttäjille. Testitulokseen ja suorituskykypisteisiin vaikuttavat mittarit Lighthouse erittelee omana “Metrics”-osionaan, josta käyttäjälle on saatavilla lisätietoa kunkin mittarin mittaamista asioista.

Mittareiden jälkeen raportissa listataan työkalun ehdottamia lisäohjeita “Opportunities” ja “Diagnostics” -osioissa, joita käyttäjä voi tutkia. Näissä kohdassa listatut korjausehdotukset parantaisivat Lighousen mukaan verkkosivun latausnopeutta, mutta käyttäjän tulee tarkistaa ne manuaalisesti. Opportunities-kohdassa Lighthouse listaa kolme parannusta vaativaa, oranssilla värikoodattua asiaa. Näitä olivat renderöinnin estävien resurssien poistaminen, kuvakokojen oikeellisuuden tarkistaminen ja kuvaformaattien tarkistaminen.

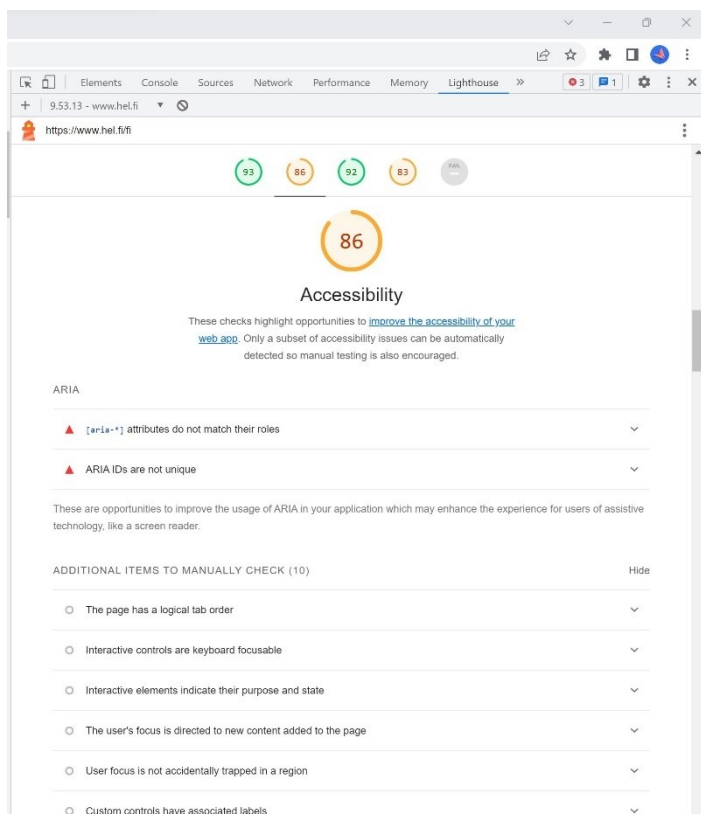


Kuva 23. – Näkymä ehdotetuista lisäohjeista

Diagnostics-osiossa Lighthouse listasi useampia asioita (kuva 23). Heikoksi ilmoitettuja punaisella värikoodattuja tekijöitä olivat tekstin näkyvissä pysymisen varmistaminen verkkofontin latauksen aikana ja välimuistikäytäntöjen tehostaminen. Parannusta vaativaksi asiaksi Lighthouse listasi liian suurten DOM-kokojen välttämisen, jotka liittyivät useaan verkkosivulta löytyneeseen elementtiin. Diagnostics-osiossa listattiin suorituskykyyn mahdollisesti vaikuttavia tekijöitä näiden lisäksi vielä viisi (5) kappaletta, mutta näiden vaikuttavuutta työkalu ei erikseen määritellyt värikoodailla.

Jokaisen tekijän kohdalla sekä Opportunities- ja Diagnostics-osioissa Lighthouse tarjosi lisätietoja elementeistä, joita ehdotus koski sekä selityksen, minkälainen korjaus tai ratkaisu parantaisi asiaan liittyen verkkosivun latausnopeutta. Lisätiedoissa tarjottiin myös linkki Googlen omaan Chrome Developers-dokumentaatioon, jossa kyseessä olevasta tekijästä tarjottiin lisätietoa.

Suorituskykyosion jälkeen Lighthouse näyttää raportissa saavutettavuuteen vaikuttavat tekijät (kuva 24). Ensimmäisenä näytetään virheet tai ongelmat saavutettavuudessa, jotka korjaamalla saavutettavuutta voidaan parantaa. Lighthouse löysi verkkosivulta kaksi (2) punaisella värikoodattua ongelmaa ARIA-attribuuttien ja -roolien yhdenmukaisuuksissa, sekä ARIA:n mukaisissa yksilöintitunnisteissa (ID), jotka eivät ole yksilöllisiä. Sekä yksilöintitunnisteita, että ARIA-attribuutteja ja -rooleja koskevat virheilmoitukset liittyivät sivulta löytyviin ``-elementteihin.



Kuva 24. – Näkymä työkalun saavutettavuusraportista

Automaattisesti tunnistettujen saavutettavuusongelmien listauksen alla on osio, jossa työkalu on listannut osia ja ominaisuuksia, jotka käyttäjän tulisi testata manuaalisesti. Näitä Lighthouse on löytänyt verkkosivulta yhteensä 10 kappaletta:

1. Sivun Tab-järjestys on looginen
2. Interaktiiviset säätimet ovat tarkennettavissa näppäimistöllä
3. Interaktiiviset elementit osoittavat niiden tarkoituksen ja tilan
4. Käyttäjän huomio kohdistuu sivulle lisättyyn uuteen sisältöön
5. Käyttäjän huomio ei vahingossa juutu tietylle alueelle
6. Mukautetuissa säätimissä on tunnisteet
7. Mukautetuissa säätimissä on ARIA-roolit
8. Sivun visuaalinen järjestys on DOM-järjestyksen mukainen
9. Näkymätön sisältö on piilotettu avustavilta teknologioilta

10. HTML5 mukaisia maamerkkielementtejä käytetään navigoinnin parantamiseen

Myös tässä osiossa jokaisen kohdan pystyi klikkaamaan auki, jolloin työkalu tarjosi kohdasta lisätietoja sekä linkin Chrome Developers-dokumentaatioon tarkempaa infoa varten. Saavutettavuusosiossa työkalu kertoi hyväksytysti suoritettuja testejä olleen yhteensä 21 kappaletta, joita olivat:

1. `<body>` -elementissä ei ole `[aria-hidden="true"]` -attribuuttia
2. ARIA-attribuutit, jotka edellyttävät ARIA-roolitusta, ovat oikein
3. ARIA-roolit ovat oikein
4. ARIA-attribuutit sisältävät oikeanlaiset arvot
5. ARIA-attribuutit ovat oikein eivätkä sisällä kirjoitusvirheitä
6. Button-elementeillä on saavutettavat nimet
7. Kuva-elementeillä on alt-tekstit
8. Lomake-elementeillä on nimilaput
9. `<meta name="viewport">` -elementissä ei käytetä `[user-scalable="no"]` -attribuuttia ja `[maximum-scale]` -attribuutti ei ole pienempi kuin 5
10. Elementit, jotka sisältävät `[aria-hidden="true"]` -attribuutin, eivät sisällä tarkennettavia jälkeläisiä
11. Tausta- ja etualan väreillä on riittävä kontrastisuhde
12. Dokumentilla on `<title>`-elementti
13. Aktiivisten ja tarkennettavissa olevien elementtien id-attribuutit ovat yksilöllisiä
14. `<html>`-elementillä on lang-attribuutti
15. `<html>`-elementillä on oikeanlainen arvo lang-attribuutissa
16. Linkeillä on havaittavat nimet
17. Listat sisältävät vain ``-elementtejä ja scriptejä tukevia elementtejä (`<script>` ja `<template>`)

18. Lista-elementit () ovat joko , tai <menu>-elementin sisällä

19. Minkään elementin [tabindex]-arvo ei ole suurempi kuin 0

20. Lang-attribuuteilla on oikeanlaiset arvot

21. Otsikko-elementit esiintyvät peräkkäin, laskevassa järjestyksessä

Sellaisia testejä, jotka eivät tässä kategoriassa soveltuneet verkkosivun testaamiseen oli yhteensä myös 21 kappaletta. Nämä testit löytyvät liitteestä 5 kuvista 57–58.

Raportin lopussa Lighthouse listaa samaan tapaan “Best Practices”-osiossa kyseiseen osa-alueeseen vaikuttaneet tekijät. Punaisella löydettyjä tekijöitä on yksi (1) kappale, joka Lighthousen mukaan on konsoliin lokiin tallentunut selaimen virheviesti. Lokiin tallentuneita virheviestejä on yhteensä kolme ja nämä Lighthousen mukaan ilmaisevat, että verkkosivulla tai selaimessa on ratkaisemattomia ongelmia. Tässä osiossa läpimenneitä testejä oli yhteensä 12 kappaletta ja verkkosivun testaamiseen soveltumattomia testejä yksi (1) kappale.

Hakukoneoptimoinnin (SEO) -osiossa Lighthouse ilmoittaa punaisella kaksi (2) kohtaa. Näitä olivat metakuvauksen puuttuminen sekä indeksoimattomat linkit. Manuaalisesti tarkistettavaksi listattiin strukturoidun datan oikeellisuuden tarkistaminen. Hyväksytysti läpäistyjä testejä oli yhteensä 10 kappaletta ja soveltumattomia testejä kaksi (2) kappaletta. Hakukoneoptimoinnin ja verkkosovelluksen progressiivisuuden (PWA) -osioiden tulokset löytyvät liitteestä 5 kuvista 64–65.

Yhteenvedona Google Lighthouse havaitsi Helsingin kaupungin etusivulla saavutettavuusongelmia, jotka liittyivät ARIA rooleihin, tyyppeihin ja ominaisuuksiin. Työkalu listasi erilaisia verkkosivuston sisältöjä, joita tulisi manuaalisesti testata. Lighthouse tarjosi lisätietoja näistä saavutettavuushuomioista ja esitti myös, mitkä testit olivat hyväksytysti läpäisty. Google Lighthouse tarjosi saavutettavuuden lisäksi raportteja suorituskyvystä, parhaista käytännöistä, progressiivisuudesta ja hakukoneoptimoinnista. Näiden raporttien avulla saatiin kattavampi kuva verkkosivuston tilasta ja mahdollisista parannuskohteista.

4.4.6 Salvia

Aluehallintoviraston kehittämä ja julkaisema julkinen Salvia-ohjelmisto on verkkosivustojen saavutettavuuden automaattisen testauksen työkalu. Julkinen Salvia-ohjelmisto on lisensoitu avoimen lähdekoodin lisenssillä ja ladattavissa GitHub-palvelusta. Salvia hyödyntää taustalla saavutettavuuden testaamiseen QualWeb-core -nimistä testausydintä, joka sisältää 72 ACT-testaussääntöä.

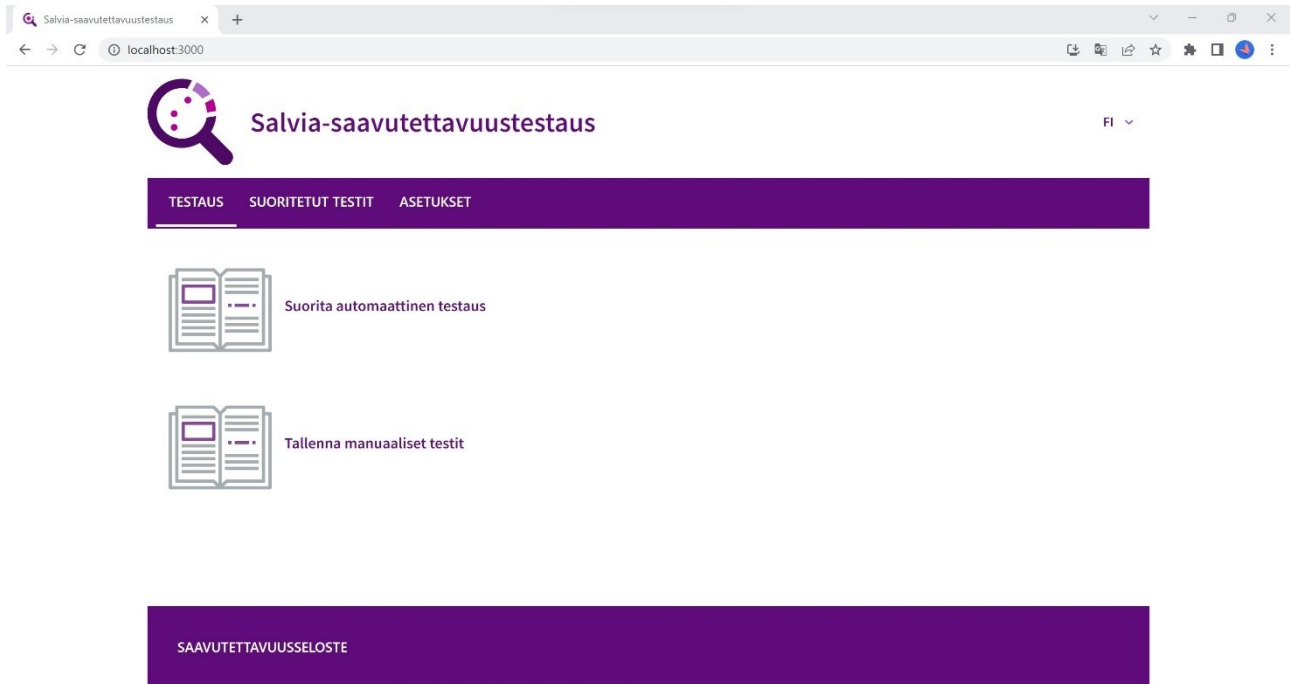
Salvia käyttää niistä kuitenkin vain 47, ja ne liittyvät WCAG-ohjeistuksen A- ja AA- tasojen vaatimukseen. (Konsi 16.1.2023b.)

Salvian osalta Konsi (16.1.2023b) kertoo, että osa ACT-testeistä voidaan suorittaa automaattisesti, ja osa taas ainakin toistaiseksi vaatii erillistä käyttäjän arviointia, manuaalista testaamista. Salvia pystyy tarkastamaan ACT-testeistä automaattisesti esimerkiksi sen, että html-elementti sisältää lang-attribuutin tai mittaamaan käytetyn tekstin kontrastin. Toisaalta Konsi (16.1.2023b) muistuttaa, että ACT-testien läpäisy ei vielä ole tae siitä, etteikö sivustolla voisi olla muita saavutettavuusongelmia.

Salvian lähdekoodi löytyy GitHub-palvelusta ja sen voi ladata omalle tietokoneelle kloonamalla repositorion git-versionhallintaohjelmalla tai lataamalla se zip-tiedostona. Salvian käyttöönottoon tarvitaan Docker, joka on ohjelmistoratkaisu, jolla voidaan virtualisoida ohjelmiston suorittamiseen tarvittavat käyttöjärjestelmät ja niiden riippuvuudet. Ladatussa Salvian lähdekoodissa on valmiiksi Docker-työkalun tarvitsemat osat, mutta Docker on ladattava omalle koneelle vielä erikseen. Kun Dockerin kontit ovat kunnossa, niin Salviaa voidaan käyttää millä tahansa selaimella osoitteessa <http://localhost:3000/>. (Konsi 16.1.2023b.)

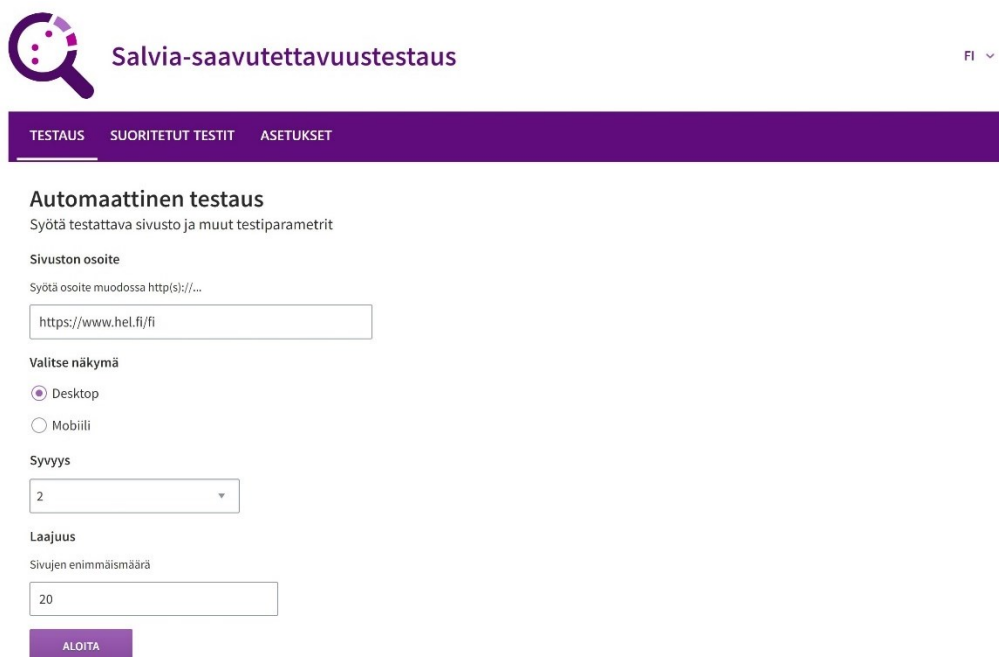
Testaaminen

Salvia-ohjelmisto otettiin käyttöön lataamalla lähdekoodin zip-tiedosto aluehallintoviraston julkisesta GitHub-repositoriosta, osoitteesta <https://github.com/accessibility-fi/salvia>. Ladatussa Salvian lähdekoodissa oli valmiiksi Docker-työkalun tarvitsemat osat, mutta Docker tuli asentaa erikseen itse omalle tietokoneelle. Meiltä Docker löytyi jo ennestään, joten etenimme GitHub-repositoriosta löytyvien ohjeiden mukaan Salvian asennukseen. Asennusten jälkeen Salvia käynnistettiin Google Chrome -selaimessa osoitteessa <https://localhost:3000>.



Kuva 25. – Salvian alkunäkymä

Käynnistämisen jälkeen Salvian alkunäytössä (kuva 25) nähdään valittuna oleva Testaus-välilehti, jolta löytyvät vaihtoehdot automaattisen testin suorittamiseen ja manuaalisten testien tallentamiseen. Automaattisessa testauksessa Salvia kysyy käyttäjältä sivuston osoitteen, näkymän, jolla testi suoritetaan, testin syvyyden ja testin laajuuden (kuva 26). Sivuston osoitteeksi annoimme <https://hel.fi> ja näkymäksi valitsimme "Desktop" käyttämämme laitteen mukaisesti. Syvyyden pidimme oletusasetuksella 2 ja laajuudeksi (sivujen enimmäismäärä) jätimme myöskin oletuksena olleen 20. Valintojen jälkeen "Aloita"-nappia painamalla etenimme käyttöliittymässä seuraavaan vaiheeseen.



Salvia-saavutettavuustestaus FI

TESTAUS SUORITETUT TESTIT ASETUKSET

Automaattinen testaus

Syötä testattava sivusto ja muut testiparametrit

Sivuston osoite

Syötä osoite muodossa http(s)://...

Valitse näkymä

☒ Desktop

☐ Mobiliili

Syvyys

Laajuus

Sivujen enimmäismäärä

ALOITA

Kuva 26. – Näkymä ennen automaattisen testin suorittamista

Seuraavaksi Salvia etsi aluksi annetun verkkosivuston sivuja, jotka se ilmoitti etsinnän jälkeen listana. Käyttäjän tuli valita listasta, mitä verkkosivuston sivuja testataan. Valitsimme listasta ainoastaan sivun <https://hel.fi/>, jotta vain etusivua testattaisiin. Valinnan jälkeen automaattinen testi käynnistettiin "Aloita"-nappia painamalla.

Salvia ilmoitti testin suorittamisen jälkeen testin onnistumisesta ja siitä, että tuloksista muodostettu raportti löytyy käyttöliittymän välilehdeltä "Suoritetut testit". Suoritetuista testeistä nähtiin testin suorittamisen päivämäärä ja kellonaika, sekä ikonien avulla muodostettu yhteenveto testin tuloksista. Testin tulokset tarjottiin käyttäjälle json-, json (PDF)- ja PDF-muodossa. Valitsimme ladattavan raportin muodoksi PDF. Raportti löytyy kokonaisuudessaan tämän työn liitteestä 6.

Salvian muodostama raportti testin tuloksista oli saatavilla käyttäjän valinnan mukaan suomen-, ruotsin- ja englannin kielellä, ja raportissa oli sivuja yhteensä seitsemän. Raportin ensimmäisellä sivulla ilmoitettiin perustiedot testistä, eli testattava verkkosivu, testin suorittamisen päivämäärä ja kellonaika, sekä testissä käytetyn näyttöpäätelaitteen resoluutio. Lisäksi kerrottiin lyhyt yhteenveto Salviasta, kuten sen perustuvuus QualWeb-testausytimeen ja A- ja AA-tason WCAG 2.1. saavutettavuusvaatimuksiin.

Raportin toisella sivulla tarjottiin testituloksista yhteenveto. Alkuun omana osionaan listattiin tutkitut verkkosivut ja niiden määrä. Seuraavassa osiossa listattiin verkkosivulle suoritetut testit. Suoritetut testit oli kategorisoitu neljään osioon; löydettyihin saavutettavuuspuutteisiin, mahdollisiin

saavutettavuuspuutteisiin, joita ei pystytty automaattisesti testaamaan, hyväksytysti läpäistyihin testeihin sekä testeihin, jotka eivät soveltuneet verkkosivulta löytyneiden sisältöjen testaamiseen.

Suoritetut testit olivat jokaisessa kategoriassa listattuna järjestyksessä niiden ACT-testaussääntö-numeron mukaan. ACT-numeron jälkeen kerrottiin kuvaus kyseisestä ACT-testaussäännöstä ja linkki kyseiseen sääntöön ACT-Rules Communityn sivuille.

Salvia löysi Helsingin kaupungin etusivulta yhden (1) saavutettavuusongelman, seitsemän (7) käyttäjän manuaalista tarkistamista vaativaa kohtaa, 15 hyväksytysti läpäistyä kohtaa sekä 41 kappaletta sellaista ACT-testaussääntöä, jota ei voitu soveltaa verkkosivun testaamisessa. Löydetty saavutettavuusongelma oli QW-ACT-R18 ("id"-attribuutin arvo on yksilöllinen) -säännön poikkeama: useammalla kuin yhdellä elementillä on sama id-attribuutin arvo. Näitä olivat raportin mukaan kaksi -elementtiä.

Manuaalisesti testattaviksi jäivät seuraavien seitsemän ACT-testaussäännön toteutuminen:

1. QW-ACT-R63, Dokumentilla on maamerkki ei-toistuvalla sisällölle
2. QW-ACT-R64, Dokumentilla on otsikko ei-toistuvalla sisällölle
3. QW-ACT-R73, Toistuvan sisällön lohko on kutistettava
4. QW-ACT-R74, Dokumentilla on mekanismi kohdistuksen siirtämiseksi ei-toistuvaan sisältöön
5. QW-ACT-R62, Järjestyksessä kohdistettaviin sisältöihin kuuluvalla elementillä on näkyvä kohdistus
6. QW-ACT-R41, Virheilmoitus selittää virheen lomakekentän syötteessä.
7. QW-ACT-R75, Ohita toistuvan sisällön lohkot

Hyväksytysti läpäistyjä ACT-testaussääntöjen mukaisia testejä olivat:

1. QW-ACT-R1, HTML-sivulla on sivuotsikko
2. QW-ACT-R2, Html-elementissä on lang-attribuutti
3. QW-ACT-R5, Html-elementin lang-attribuutin kelpoisuus

4. QW-ACT-R11, Painikkeella on saavutettava nimi
5. QW-ACT-R12, Linkillä on saavutettava nimi
6. QW-ACT-R13, Elementillä, jolle on määritelty "aria-hidden"-attribuutti, ei ole kohdistettavaa sisältöä
7. QW-ACT-R14, "Meta viewport" ei estä zoomausta
8. QW-ACT-R22, Body-elementin alielementin lang-attribuutin arvo on sallittu
9. QW-ACT-R28, Elementillä, jolle on määritelty role-attribuutti, on annettu myös vaaditut tilat ja ominaisuudet
10. QW-ACT-R16, Lomake-elementillä on saavutettava nimi
11. QW-ACT-R20, Role-attribuutilla on sallittu arvo
12. QW-ACT-R17, Kuvalla on saavutettava nimi
13. QW-ACT-R35, Otsikolla on saavutettava nimi
14. QW-ACT-R37, Teksti täyttää kontrastin vähimmäisvaatimukset
15. QW-ACT-R65, Elementeillä, jotka sisältävät vain visualisoivia alielementtejä, ei ole kohdistettavia sisältöjä

Salvian listaamat testit, jotka eivät soveltuneet sivulta löytyneiden sisältöjen testaamiseen oli yhteensä 41 kappaletta. Nämä ACT-testaussääntöjen mukaiset testit löytyvät liitteestä 6, kuvista 72–74.

Yhteenvedona Salvia nosti Helsingin kaupungin etusivulla esiin yhden saavutettavuusongelman liittyen toistuvaan id-attribuuttiin eri elementeillä. Työkalu esitti selkeästi kaikki suoritettavat saavutettavuustestit, kuten hyväksytysti läpäisseet testit, manuaalisesti tarkistettavat kohdat ja ACT-säännöt, joita ei voitu soveltaa sivun testaamiseen. Työkalu listasi selkeästi kaikki kohdat testiraportissa, mutta saavutettavuusongelman kohdalla se ei esittänyt parannusehdotusta.

4.5 Vertailuanalyysin tulokset

Jokainen tutkimuksessamme mukana ollut työkalu perustui ACT-testaussäännöt sisältävään testausytimeen. Tällä perusteella teimme oletuksen, että WCAG 2.1 mukaisten saavutettavuusvaatimusten osalta työkalut arvioivat verkkosivun saavutettavuutta yhtä hyvin. Kun oletetaan työkalujen arvioivan verkkosivun saavutettavuutta perustuen samoihin kriteereihin, seuraavaksi oleellisimmaksi koettiin tarkastella sitä, miten tarkasti, hyvin ja kattavasti työkalut saattavat käyttäjän tietoon verkkosivulta löytämänsä saavutettavuuteen vaikuttavat ominaisuudet.

Testausraportista olisi käytävä ilmi saavutettavuusvirheiden lisäksi myöskin ominaisuudet, jotka verkkosivulla on toteutettu onnistuneesti ja saavutettavasti, sekä kaikki testit, joita työkalulla voidaan suorittaa, jotta työkalujen kyvystä arvioida saavutettavuutta voidaan saada luotettava käsitys. Työkaluja testattaessa huomattiinkin, että ACT-testaussäännöt sisältävistä testausytimistä huolimatta työkalut ottivat kantaa hyvin erilaisilla tavoilla testatun verkkosivun saavutettavuuteen liittyviin tekijöihin ja löysivät myös osittain toisistaan poikkeavia saavutettavuusvirheitä.

Ilman työkalujen sisältämien testausytimien sisältöjen tarkempaa tutkimusta kattavaa tulosten raportointitapaa voidaan pitää ainoana tapana, josta käyttäjälle voi käydä selväksi, mitä testejä työkalu on suorittanut tai yrittänyt suorittaa verkkosivulla. Esimerkiksi ainoastaan Salvia ja Google Lighthouse sisällyttivät testausraportteihinsa kaikki testit, jotka työkalu suoritti tai yritti suorittaa verkkosivulla. Ne kategorisoivat ja ilmoittivat erillisinä osioina epäonnistuneiden testien lisäksi sellaiset testit, jotka verkkosivu oli läpäissyt hyväksytysti, manuaalisesti testattavat testit ja testit, joita verkkosivun testaamiseen ei voitu soveltaa.

Axe DevTools ja Access Assistant raportoivat ainoastaan epäonnistuneet testit tai löytämänsä saavutettavuusvirheet. Näiden työkalujen tarjoamasta testausraportista tai testin tuloksista ei käynyt selväksi, mitä testejä verkkosivu oli läpäissyt hyväksytysti ja sisälsikö työkalu sellaisia testejä, joita se ei voinut verkkosivun testaamisessa soveltaa. Access Assistant tosin sisälsi myöskin ohjattuja, manuaalisesti suoritettavia testejä, jotka käyttäjä olisi voinut suorittaa tarvittaessa. Axe DevToolsista testaamamme versio ei sisältänyt ohjattuja manuaalisia testejä ollenkaan.

Myöskään Accessibility Insights for Web ei ottanut kantaa hyväksytysti suorittamiinsa testeihin tai testeihin, joita se ei voinut automaattisesti soveltaa verkkosivun testaamiseen. Se kuitenkin sisälsi ohjattuja manuaalisia testejä tarkan ohjeistuksen kanssa, joita suorittamalla käyttäjä olisi voinut muodostaa kattavamman kuvan verkkosivun saavutettavuudesta.

WAVE kertoi testit, jotka se on suorittanut hyväksytysti verkkosivulla. Tämän lisäksi se kertoi, mitä testejä tulisi suorittaa manuaalisesti saavutettavuuden varmistamiseksi. WAVE ei kuitenkaan

tarjonnut testausraportissaan käyttäjälle tietoa kaikista testeistä, joita se oli suorittanut tai yrittänyt suorittaa verkkosivulla.

Työkalujen vertailusta on koottu yhteenveto taulukkoon (taulukko 1). Taulukossa on yllä mainituista tekijöistä johtuen päätetty pitää pääpaino työkalujen sisältämissä testausmenetelmissä sekä tavoissa raportoida testien tuloksia. Selkeyden vuoksi taulukossa esitetään myöskin jokaisen työkalun sisältämä testausydin ja tapa, jolla työkalua käytettiin.

Taulukon ensimmäisellä vaakarivillä esitellään tutkimuksemme kohteena olleet työkalut. Taulukon ensimmäisellä pystyrivillä esitellään allekkain ominaisuudet, jotka työkalu joko sisältää tai ei sisällä, poislukien testausydin ja käyttötapa, jotka kerrotaan erikseen.

Taulukon 1 perusteella voidaan tehdä johtopäätös, että tutkimuksessamme mukana olleista työkaluista parhaiten verkkosivuston saavutettavuutta pystyi arvioimaan Google Lighthouse. Se esitti testin tulokset raportissaan selkeästi kategorisoituna hyväksytysti läpäistyihin testeihin, epäonnistuneisiin testeihin, kohteeseen soveltumattomiin testeihin ja manuaalisesti testattaviin testeihin. Raportin perusteella pystyttiin muodostamaan kattava käsitys siitä, miten ja kuinka laajasti työkalu kykeni arvioimaan testaamansa verkkosivun saavutettavuutta.

Taulukko 1. – Yhteenvedo vertailuanalyysin tuloksista

	WAVE	Axe DevTools	Accessibility Insights for Web	Access Assistant	Google Lighthouse	Salvia
Testausydin	WAVE Testing Engine	Axe-Core	Axe-Core	Access Engine	Axe-Core	QualWeb-core
Käyttötapa	Selain	Selainlaajennus	Selainlaajennus	Selainlaajennus	Chrome-selaimen kehittäjätyökalu	Ladattava sovellus
Sisältää automaattisia testejä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä
Sisältää ohjattuja manuaalisia testejä	Kyllä	Ei	Kyllä	Kyllä	Kyllä	Kyllä
Sisältää visuaalisen avustajan	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Ei
Ilmoittaa hyväksytysti läpäistyt testit	Kyllä	Ei	Ei	Ei	Kyllä	Kyllä
Ilmoittaa epäonnistuneet testit	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä
Ilmoittaa manuaalisesti testattavat testit	Kyllä	Ei	Kyllä	Ei	Kyllä	Kyllä
Ilmoittaa testit, joita ei voitu soveltaa verkkosivun testaamiseen	Ei	Ei	Ei	Ei	Kyllä	Kyllä
Viittaus testituloksessa ohjeistukseen tai standardiin	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä
Esittää korjausehdotuksen	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Ei

5 Pohdinta

Opinnäytetyömme tarkoituksena oli selvittää vertailuanalyysin avulla, kuinka hyvin saavutettavuuden testaamiseen tarkoitetut automaattiset työkalut pystyvät tunnistamaan saavutettavuuteen liittyviä ongelmia verkkosivustolla. Asetimme tavoitteeksemme myös yleisen tietoisuuden lisäämisen saavutettavuudesta ja sen merkityksestä verkkosivustojen käyttäjäkokemuksessa, suunnittelussa ja kehittämisessä.

Tietoperustassamme todettiin automaattisten testaustyökalujen olevan nopeita ja tehokkaita joidenkin yleisimpien saavutettavuusongelmien testaamiseen. Selvitimme, että automaattisia työkaluja käyttämällä pystytään havaitsemaan yksityiskohtaisia ongelmia, joita manuaalisella testaamisella voi olla vaikeampi havaita. Toisaalta myöskin todettiin, että automaattisilla työkaluilla on myös heikkouksia, kuten mahdollisuus virheellisiin tuloksiin ja kykenemättömyys arvioida verkkosivun helppokäyttöisyyttä kokonaisuutena.

Valitsimme tutkimukseemme mukaan viisi yleistä saavutettavuuden automaattiseen testaamiseen tarkoitettua työkalua, sekä aluehallintoviraston, vuoden 2023 alussa julkaiseman oman saavutettavuuden testausohjelmiston Salvian. Koimme Salvian ottamisen osaksi tutkimusta äärimmäisen mielenkiintoiseksi ja ajankohtaiseksi, sillä julkisesti jaossa oleva ja kaikille saatavilla oleva Salvia-ohjelmisto vastasi pitkälti samaa Salvian versiota, kuin mitä aluehallintovirasto kertoi käyttävänsä itse saavutettavuuden valvonnassa. Salvian vertaileminen oli mielestämme mielenkiintoista myös siksi, että sen käyttöönotto- ja käyttötapa poikkesi melko huomattavasti muista tutkimuksessamme mukana olleista työkaluista.

Muut työkalut olivat käyttöönotoiltaan ja käyttötavoiltaan suoraviivaisempia. Käyttöönotoissa ilmeni kuitenkin joitakin haasteita, kuten epäonnistuneita asennuksia ja haasteita saada jokin työkalu toimimaan. Nämä ongelmat saatiin kuitenkin selätettyä yhteistyöllä, ja työkalujen testaus saatiin suoritettua tutkimuksessa määritetyllä tavalla.

Jokainen tutkimuksessamme mukana ollut työkalu perustui ACT-testaussäännöt sisältävään testausytimeen, jonka perusteella asetimme oletuksen, että työkalut tutkivat verkkosivun saavutettavuutta perustuen samoihin kriteereihin. Tästä johtuen päätettiin työkalujen vertailussa keskittyä siihen, miten kattavasti ja millä tavoin ne raportoivat tuloksia testaamastaan verkkosivusta, kun oletettiin, että ne arvioivat sitä perustuen samoihin kriteereihin.

Tutkimuksessamme ilmeni, että kohteenamme olleet automaattiset testaustyökalut ottivat esiin ja painottivat testiraporteissaan erilaisia asioita, sekä raportoivat testien tuloksia erilaisin tavoin. Nämä työkalujen erilaiset tavat raportoida testiensä tuloksia vaikuttivat eniten tulkintaan siitä, miten

hyvänä voitiin pitää kunkin työkalun kykyä arvioida saavutettavuutta. Vertailuanalyysin yhteenvedossa (taulukko 1) päätettiin pitää mukana selkeyden vuoksi työkalujen testausytimet ja käyttötavat, vaikka nämä tekijät eivät suoraan vaikuttaneet vertailun tuloksiin. Tarkoituksena oli, että myös taulukosta yksin voisi saada selkeän kokonaiskuvan siitä, mikä työkalu tarjoaa eniten tietoa käyttäjälleen testatun verkkosivun saavutettavuudesta.

On otettava huomioon, että tutkimuksemme ei käsittänyt testausytimien sisältöihin perehtymistä nykyistä syvällisemmin, joten testausytimien sisältöjen samankaltaisuus jäi vain olettamukseksi. Jatkokehitysideana voisikin olla testausytimien tarkempi tutkimus, jossa tutkittaisiin syvällisemmin testausytimien sisältöjä ja vertailtaisiin, miten eri testausytimet eroavat toisistaan. Muuta tutkimuksemme kannalta huomioitavaa on myös, että testeissä käytetyn verkkosivun valinnalla on merkitystä työkalujen suorittamien testien tuloksiin. Tämä tutkimus voitaisiinkin toistaa jollakin toisella verkkosivulla, esimerkiksi sellaisella, joka on tarkoituksella saavuttamaton. Tällä tavoin voitaisiin mahdollisesti arvioida työkalujen kykyä tutkia saavutettavuutta vieläkin tarkemmin.

Kokonaisuudessaan koemme saavuttaneemme opinnäytetyölle asettamamme tavoitteet onnistuneesti ja vastanneemme tutkimuskysymykseemme. Opinnäytetyömme tuloksia voidaan hyödyntää saavutettavuuden testaamisessa silloin, kun halutaan arvioida, mitä automaattista työkalua kannattaa käyttää. Parhaimmillaan työmme lisää myös yleistä tietoisuutta saavutettavuudesta ja sen merkityksestä verkkosivustojen käyttäjäkokemuksessa, suunnittelussa ja kehittämisessä.

Tutkimuksen validiteetin varmistamiseen on pyritty kiinnittämään huomioita läpi koko opinnäytetyöprosessin. Aiheen valinta ja ajankohtaisuus on pyritty esittelemään selkeästi, ja tietoperustassamme on pyritty keskittymään tutkimuskysymykseemme kannalta oleellisiin teemoihin. Lähdeaineistoksi on pyritty valitsemaan mahdollisimman luotettavia lähteitä ja lähteitä on käytetty monipuolisesti. Käytetyt termit ja käsitteet on pyritty selittämään auki mahdollisimman yksitulkintaisella tavalla.

Toisaalta on otettava huomioon, että lähdeaineistossa ja itse tutkimuksessamme on käytetty paljon vieraskielistä lähdemateriaalia. Aineiston tulkinnassa ja suomennoksissa on pyritty mahdollisimman tarkkoihin ja oikeanlaisiin käännöksiin, mutta satunnaisten virhetulkintojen mahdollisuutta ei voida täysin poissulkea. Tekstissä käytettyihin suomennoksiin on kuitenkin pyritty lisäämään alkuperäinen termi mahdollisten tulkintavirheiden varalta.

Tutkimuksemme reliabiliteetti on pyritty varmistamaan dokumentoimalla tutkimuksen kulku ja tulokset mahdollisimman tarkasti. Tutkimuksen kohteiden valinnassa on noudatettu ennalta määritettyjä valintakriteereitä, sekä tutkimusmenetelmän valinta ja soveltaminen tutkimuksessamme on pyritty kuvaamaan ja perustelemaan mahdollisimman tarkasti. Vertailuanalyysin prosessiaskeleita on

kuitenkin jouduttu soveltamaan tutkimukseemme sopivaksi, joten valitsemamme tavan soveltuvuus tutkimukseemme voidaan kyseenalaistaa.

Työkalujen testaamisen vaiheet on dokumentoitu sekä sanallisesti, että kuvia käyttäen mahdollisimman yksityiskohtaisesti, vaihe vaiheelta. Tarkalla dokumentoinnilla olemme pyrkineet varmistamaan, että tutkimuksemme vaiheet ovat toistettavissa. Toisaalta tässäkin on otettava huomioon, että tutkimuksemme kohteina olleiden työkalujen, sekä testissä käytetyn verkkosivun mahdolliset versiomuutokset tai päivitykset vaikuttavat todennäköisesti saataviin tuloksiin, jos tutkimus toistetaan. Myöskin käyttäjästä johtuvat virheet työkalujen toiminnassa tai toiminnan havainnoinnissa on otettava mahdollisuutena huomioon.

Lähteet

Abou-Zahra, S., Eggert, E., Elton, B. 27.7.2019. Forms tutorial. W3C Web Accessibility Initiative (WAI). Luettavissa: <https://www.w3.org/WAI/tutorials/forms/>. Luettu: 12.4.2023.

Abou-Zahra, S., Fiers, W., Kraft, M., Mueller, M. 31.10.2019. Accessibility Conformance Testing (ACT) Rules Format 1.0. W3C Recommendation. Luettavissa: <https://www.w3.org/TR/act-rules-format-1.0/>. Luettu: 24.4.2023.

Access Assistant s.a. Chrome web store. Luettavissa: <https://chrome.google.com/webstore/detail/access-assistant/ojiighldhdmahfdnhfdebnpmmbiemdfm>. Luettu: 28.4.2023.

Aluehallintovirasto 23.1.2023. Esteettömyyssäntely tuo uusia digitaalisia palveluita saavutettavuusvaatimusten piiriin. Luettavissa: <https://www.saavutettavuusvaatimukset.fi/esteettomyysaantely-tuo-uusia-digitaalisia-palveluita-saavutettavuusvaatimusten-piiriin/>. Luettu: 21.3.2023.

Aluehallintovirasto 27.1.2020. Viranomaisten uusilta verkkosivuilta puuttuu lain vaatimia saavutettavuustietoja. Luettavissa: <https://avi.fi/tiedote/-/tiedote/69907647>. Luettu: 5.3.2023.

Aluehallintovirasto s.a.a. Sanastoa ja termejä. Luettavissa: <https://www.saavutettavuusvaatimukset.fi/yleista-saavutettavuudesta/sanastoa-ja-termeja/>. Luettu: 24.3.2023.

Aluehallintovirasto s.a.b. Kenelle saavutettavuus on tärkeää? Luettavissa: <https://www.saavutettavuusvaatimukset.fi/yleista-saavutettavuudesta/kenelle-saavutettavuus-on-tarkeaa/#yli-miljoona-suomalaista-tarvitsee-saavutettavampia-palveluja>. Luettu: 21.3.2023.

Aluehallintovirasto s.a.c. Yleistä saavutettavuudesta. Luettavissa: <https://www.saavutettavuusvaatimukset.fi/yleista-saavutettavuudesta/#saavutettavuus-on-verkkomaailman-esteettomytta>. Luettu: 5.3.2023.

Aluehallintovirasto s.a.d. Digipalvelulain vaatimukset. Luettavissa: <https://www.saavutettavuusvaatimukset.fi/digipalvelulain-vaatimukset/>. Luettu: 13.3.2023.

Aluehallintovirasto s.a.e. Muita lakeja. Luettavissa: <https://www.saavutettavuusvaatimukset.fi/digipalvelulain-vaatimukset/muita-lakeja/>. Luettu: 13.3.2023.

Aluehallintovirasto s.a.f. Tietoa WCAG-ohjeistuksesta. Luettavissa: <https://www.saavutettavuusvaatimukset.fi/digipalvelulain-vaatimukset/tietoa-wcag-kriteereista/>. Luettu: 24.3.2023.

- Babich, N. 2019. User Centered Design Principles & Methods. Luettavissa: <https://xd.adobe.com/ideas/principles/human-computer-interaction/user-centered-design/>. Luettu: 05.03.2023.
- Bailey, E. 23.10.2018. Quick Tip: Use the “lang” Attribute for Better Accessibility. Envato Tuts+. Luettavissa: <https://webdesign.tutsplus.com/tutorials/use-the-lang-attribute-for-better-accessibility--cms-31961>. Luettu: 6.4.2023.
- Celia 2022a. WCAG. Luettavissa: <https://www.saavutettavasti.fi/verkkosisaltojen-saavutettavuus/wcag/>. Luettu: 5.3.2023.
- Celia 2022b. Hyödyllisiä sivustoja ja työkaluja. Luettavissa: <https://www.saavutettavasti.fi/verkkosisaltojen-saavutettavuus/linkkejä-ja-tyokaluja/>. Luettu: 13.4.2023.
- Deque Docs, s.a. Axe DevTools Extension v4.0.0. Deque Systems. Luettavissa: <https://docs.deque.com/devtools-html/4.0.0/en/devtools-extension>. Luettu: 28.4.2023.
- Deque s.a. Axe DevTools. Luettavissa: <https://www.deque.com/axe/devtools/>. Luettu: 24.4.2023.
- Dumont, S. 2021. The difference between UX and UI, according to designer and Maze CEO, Jonathan Widawski. Maze. Luettavissa: <https://maze.co/blog/ui-vs-ux/>. Luettu: 5.3.2023.
- Gevorkian, D. 24.3.2020. All you need to know about web accessibility and UX. Devmio Blog. Blogi. Luettavissa: <https://devm.io/ux/web-accessibility-ux-170115>. Luettu 6.4.2023.
- Gilbert, R. 2019. Inclusive Design for a Digital World: Designing with Accessibility in Mind. Apress. New York. E-kirja. Luettu: 30.3.2023.
- Google Developers s.a. Lighthouse overview. Luettavissa: <https://developer.chrome.com/docs/lighthouse/overview/>. Luettu: 24.4.2023.
- Gordon, K. 2022. 5 Visual Treatments that Improve Accessibility. Nielsen Norman Group. Luettavissa: <https://www.nngroup.com/articles/visual-treatments-accessibility/>. Luettu: 3.4.2023.
- Harvard University s.a. Automated Tools for Testing Accessibility. Digital Accessibility. Luettavissa: <https://accessibility.huit.harvard.edu/auto-tools-testing>. Luettu: 13.4.2023.
- Helsingin kaupunki 2022. Digitalisaatiolla teemme Helsingistä maailman toimivimman kaupungin. Luettavissa: <https://digi.hel.fi/esittely/>. Luettu: 5.3.2023.
- Hirsjärvi, S., Remes, P. & Sajavaara, P. 2014. Tutki ja kirjoita. 19. painos. Kustannusosakeyhtiö Tammi. Helsinki.

Invalidiliitto ry s.a. Saavutettavuus. Luettavissa: <https://www.invalidiliitto.fi/esteettomyys/saavutettavuus>. Luettu: 24.3.2023.

ISO 9241-11:2018. the International Organization for Standardization. Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts. Luettavissa: <https://kebs.isolutions.iso.org/obp/ui#iso:std:iso:9241:-11:ed-2:v1:en>. Luettu: 20.3.2023.

ISO 9241-210:2010. the International Organization for Standardization. Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems. Luettavissa: <https://richardcornish.s3.amazonaws.com/static/pdfs/iso-9241-210.pdf>. Luettu: 23.3.2023.

Juviler, J. 12.7.2022. What is Localhost? (And How Can You Use It?). HubSpot. Luettavissa: <https://blog.hubspot.com/website/what-is-localhost>. Luettu: 8.5.2023.

Kehitysvammaliitto ry 1.2.2023. Näppäimistöselaaminen. Luettavissa: <https://papunet.net/saavutettavuus/miksi-saavutettava/avustavat-teknologiat/nappaimistoselaaminen/>. Luettu: 12.4.2023.

Kehitysvammaliitto ry 16.3.2023. Suunnittele selkeät sivupohjat. Luettavissa: <https://papunet.net/saavutettavuus/ohjeita-ja-oppaita/saavutettavien-verkkosivujen-suunnitteluopas/suunnittele-selkeat-sivupohjat/>. Luettu: 25.3.2023.

Kehitysvammaliitto ry 21.2.2023a. Kuka hyötyy saavutettavuudesta? Luettavissa: <https://papunet.net/saavutettavuus/miksi-saavutettava/kuka-hyotyy-saavutettavuudesta/>. Luettu: 21.3.2023.

Kehitysvammaliitto ry 21.2.2023b. Saavutettavien verkkosivujen suunnitteluopas. Luettavissa: <https://papunet.net/saavutettavuus/ohjeita-ja-oppaita/saavutettavien-verkkosivujen-suunnitteluopas/>. Luettu: 30.3.2023.

Kehitysvammaliitto ry 23.1.2023. Huomioitavaa WCAG:sta. Luettavissa: <https://papunet.net/saavutettavuus/ohjeita-ja-oppaita/teknisen-saavutettavuuden-ohjeet-wcag/huomioitavaa-wcagsta/>. Luettu: 24.3.2023.

Kehitysvammaliitto ry 24.1.2023a. Tekstin koko ja kirjasintyyppi. Luettavissa: <https://papunet.net/saavutettavuus/ohjeita-ja-oppaita/saavutettavien-verkkosivujen-suunnitteluopas/tekstin-koko-ja-kirjasintyyppi/>. Luettu: 30.3.2023.

Kehitysvammaliitto ry 24.1.2023b. Suunnittele selkeitä lomakkeita. Luettavissa: <https://papunet.net/saavutettavuus/ohjeita-ja-oppaita/saavutettavien-verkkosivujen-suunnitteluopas/suunnittele-selkeita-lomakkeita/>. Luettu: 30.3.2023.

Kehitysvammaliitto ry 26.1.2023a. Tee sivuston interaktiivisista elementeistä tunnistettavia. Luettavissa: <https://papunet.net/saavutettavuus/ohjeita-ja-oppaita/saavutettavien-verkkosivujen-suunnitteluopas/tee-sivuston-interaktiivisista-elementeista-tunnistettavia/>. Luettu: 30.3.2023.

Kehitysvammaliitto ry 26.1.2023b. Saavutettavuuden arvioinnin suunnitelma. Luettavissa: <https://papunet.net/saavutettavuus/ohjeita-ja-oppaita/saavutettavuuden-arviointi/saavutettavuuden-arvioinnin-suunnitelma/>. Luettu: 16.5.2023.

Kehitysvammaliitto ry 3.4.2023. Saavutettavuuden arviointi. Luettavissa: <https://papunet.net/saavutettavuus/ohjeita-ja-oppaita/saavutettavuuden-arviointi/>. Luettu: 5.4.2023.

Kehitysvammaliitto ry 31.1.2023a. Saavutettavuuden testaaminen itse. Luettavissa: <https://papunet.net/saavutettavuus/ohjeita-ja-oppaita/saavutettavuuden-arviointi/saavutettavuuden-testaaminen-itse/>. Luettu: 5.4.2023.

Kehitysvammaliitto ry 31.1.2023b. Käyttäjätestaaminen. Luettavissa: <https://papunet.net/saavutettavuus/ohjeita-ja-oppaita/saavutettavuuden-arviointi/kayttajatestaaminen/>. Luettu: 6.4.2023.

Kehitysvammaliitto ry 7.2.2023. Työkaluja saavutettavuusarvioinnin tueksi. Luettavissa: <https://papunet.net/saavutettavuus/ohjeita-ja-oppaita/saavutettavuuden-arviointi/tyokaluja-saavutettavuusarvioinnin-tukena/>. Luettu: 13.4.2023.

Komsi, E. 16.1.2023a. Salvia-ohjelmistosta ratkaisu saavutettavuusvalvonnalle. Aluehallintovirasto. Luettavissa: <https://www.saavutettavuusvaatimukset.fi/salvia-ohjelmistosta-ratkaisu-saavutettavuusvalvonnalle/>. Luettu 24.4.2023.

Komsi, E. 16.1.2023b. Aluehallintovirasto on julkaissut testausohjelmansa lähdekoodin. Aluehallintovirasto. Luettavissa: <https://www.saavutettavuusvaatimukset.fi/aluehallintovirasto-on-julkaissut-testausohjelmansa-lahdekoodin/>. Luettu: 26.4.2023.

Laki digitaalisten palvelujen tarjoamisesta 306/2019.

Latva, T. 2023. Devaajan saavutettavuustärpit. Web-ostajan opas. Blogi. Luettavissa: <https://web-ostajanopas.fi/2023/02/07/devaajan-saavutettavuustarpit/>. Luettu: 3.4.2023.

Level Access 31.3.2023. What is Access Assistant? Level Access Product Support. Luettavissa: <https://support.levelaccess.com/hc/en-us/articles/115001757886-What-is-Access-Assistant->. Luettu: 28.4.2023.

Marsh, J. 2015. UX for Beginners. O'Reilly Media, Inc. Sebastopol. E-kirja. Luettu: 5.3.2023.

Microsoft s.a. Accessibility Insights. Accessibility Insights for Web. Luettavissa: <https://accessibilityinsights.io/docs/web/overview/>. Luettu: 24.4.2023.

Morville, P. 2004. User Experience Design. Luettavissa: http://semanticstudios.com/user_experience_design/. Luettu: 5.3.2023.

Näkövammaisten liitto ry 2019. Saavutettavuuden testaaminen. Luettavissa: <https://www.nkl.fi/fi/verkkosivujen-saavutettavuus>. Luettu: 3.4.2023.

Nielsen, J. 3.1.2012. Usability 101: Introduction to Usability. Luettavissa: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>. Luettu: 20.3.2023.

Ojasalo, K., Moilanen, T. & Ritalahti, J. 2015. Kehittämistyön menetelmät: Uudenlaista osaamista liiketoimintaan. 3.–4. painos. Sanoma Pro Oy. Helsinki. E-kirja. Luettu: 14.4.2023.

Rae, M. 2020. What Makes a Good User Experience? Luettavissa: <https://xd.adobe.com/ideas/principles/web-design/what-makes-good-ux/>. Luettu: 5.3.2023.

Robbins, J. 2018. Learning Web Design, 5th edition. O'Reilly Media, Inc. Sebastopol. E-kirja. Luettu: 26.3.2023.

Rosenzweig, E. 2015. Successful User Experience: Strategies and Roadmaps. Morgan Kaufmann. Burlington. E-kirja. Luettu: 5.3.2023.

Roto, V., Law, E., Vermeeren, A., Hoonhout, J. 2011. User Experience White Paper – Bringing clarity to the concept of user experience. Result from Dagstuhl Seminar on Demarcating User Experience, September 15-18, 2010. Luettavissa: <https://experience.aalto.fi/wp-content/uploads/2020/04/UX-WhitePaper.pdf>. Luettu: 24.3.2023.

Sanastokeskus ry s.a. Käyttöliittymä. Luettavissa: <https://termi-pankki.fi/tepa/fi/haku/k%C3%A4ytt%C3%B6liittym%C3%A4>. Luettu: 20.3.2023.

Savolainen, K. 14.3.2018. Google Lighthouse Audit - uusi tapa tutkia verkkosivuston toimivuutta. Seravo. Blogi. Luettavissa: <https://seravo.com/fi/google-lighthouse-audit-uusi-tapa-tutkia-verkkosivuston-toimivuutta/>. Luettu 13.4.2023.

Steenhout, N. 20.9.2017. When branding colours conflict with colour contrast requirements. Part Of a Whole. Blogi. Luettavissa: <https://incl.ca/when-branding-colours-conflict-with-colour-contrast-requirements/>. Luettu 3.4.2023.

Terveiden ja hyvinvoinnin laitos 2022. Esteettömyys ja saavutettavuus. Vammaispalvelujen käsikirja. Luettavissa: <https://thl.fi/fi/web/vammaispalvelujen-kasikirja/vammaisuus-yhteiskunnassa/esteettomyys-ja-saavutettavuus>. Luettu: 24.3.2023.

Tuominen, K. 2016. Benchmarking-käsikirja. Oy Benchmarking Ltd. Turku.

UsableNet 2.2.2023. What is Manual Website Accessibility Testing? (And Why It's Important). Blogi. Luettavissa: <https://blog.usablenet.com/quick-guide-to-manual-accessibility-testing-and-why-its-important>. Luettu 6.4.2023.

Valtioneuvoston asetus vammaisten henkilöiden oikeuksista tehdyn yleissopimuksen ja sen valinnaisen pöytäkirjan voimaansaattamisesta sekä yleissopimuksen ja sen valinnaisen pöytäkirjan lainsäädännön alaan kuuluvien määräysten voimaansaattamisesta annetun lain voimaantulosta 398/2016. Annettu 31.5.2016.

Viljanen, V. 1.9.2019. Saavutettavuus verkossa. Valkohattu. Luettavissa: <https://valkohattu.fi/artikkeli/saavutettavuus-verkossa>. Luettu: 3.4.2023.

Viljanen, V. 22.2.2020. Käytettävyys ja käyttökokemus. Valkohattu. Luettavissa: <https://valkohattu.fi/artikkeli/kayttokokemus>. Luettu: 20.3.2023.

W3C 2018a. Accessibility. Luettavissa: <https://www.w3.org/standards/webdesign/accessibility>. Luettu: 7.3.2023.

W3C 2019. Web Content Accessibility Guidelines (WCAG) 2.1. Virallinen suomenkielinen käännös. Luettavissa: <https://www.w3.org/Translations/WCAG21-fi/#background-on-wcag-2>. Luettu: 21.3.2023.

W3C 2022. Introduction to Understanding WCAG. Luettavissa: <https://www.w3.org/WAI/WCAG21/Understanding/intro#understanding-the-four-principles-of-accessibility>. Luettu: 24.3.2023.

W3C 2023. Accessible Rich Internet Applications (WAI-ARIA) 1.2. Luettavissa: <https://www.w3.org/TR/wai-aria-1.2/>. Luettu: 3.4.2023.

W3C s.a.a. HTML & CSS. Luettavissa: <https://www.w3.org/standards/webdesign/htmlcss.html>. Luettu: 26.3.2023.

W3C s.a.b. WAI-Tools. Luettavissa: <https://www.w3.org/WAI/about/projects/wai-tools/>. Luettu: 24.4.2023.

W3C s.a.c. About Us. ACT-Rules Community. Luettavissa: <https://act-rules.github.io/pages/about/>.
Luettu: 24.4.2023.

W3C s.a.d. ACT-implementations. ACT-Rules Community. Luettavissa: <https://act-rules.github.io/pages/implementations/overview/>. Luettu: 24.4.2023.

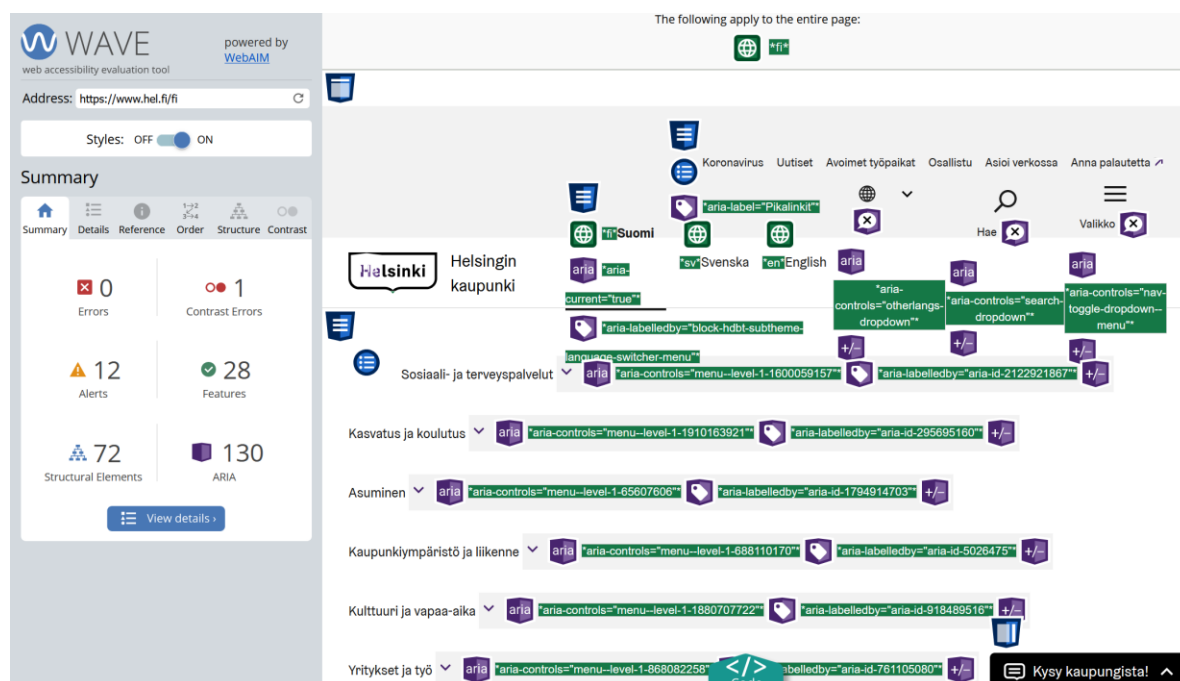
W3Schools s.a. HTML Accessibility. Luettavissa: https://www.w3schools.com/html/html_accessibility.asp. Luettu: 6.4.2023.

WebAIM – Web Accessibility in Mind 9.4.2020. Introduction to WAVE. Video. Katsottavissa: <https://www.youtube.com/watch?v=ITUDiTgAZY0>. Katsottu: 24.4.2023.

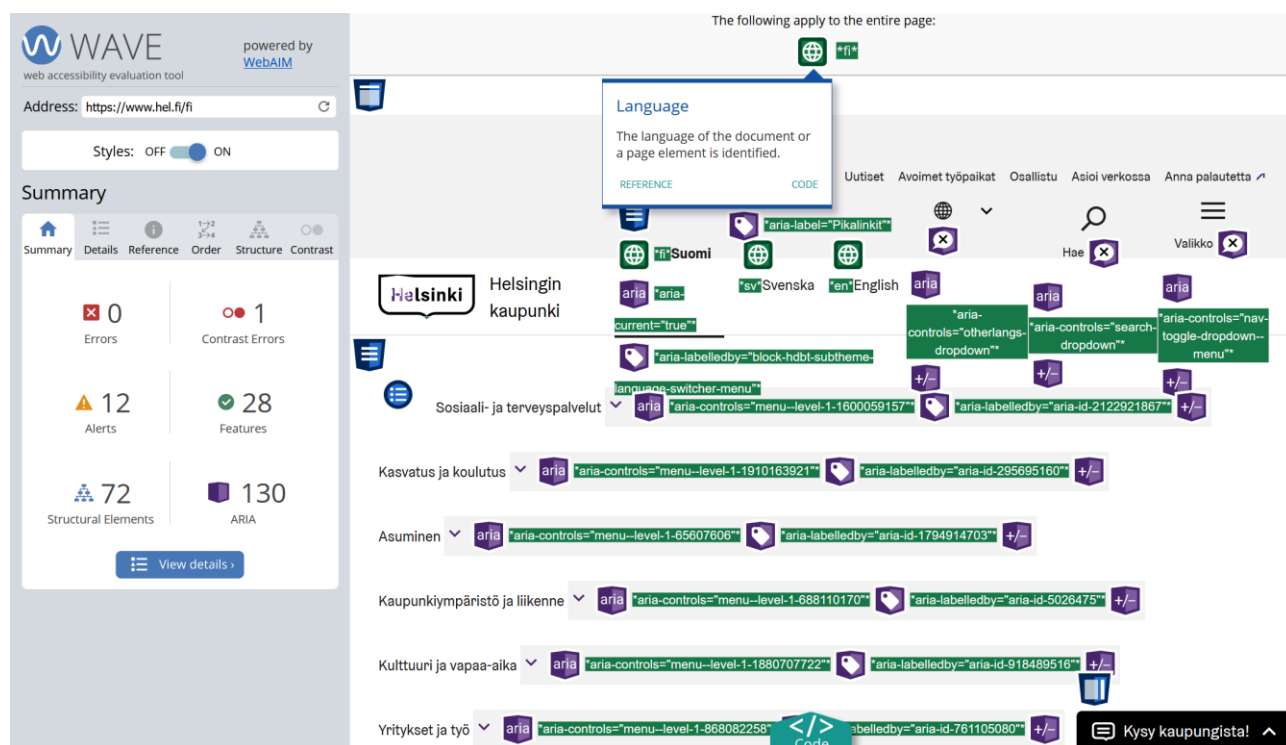
WebAIM – Web Accessibility in Mind s.a. WAVE Browser Extensions. Luettavissa: <https://wave.webaim.org/extension/>. Luettu: 24.4.2023.

Liitteet

Liite 1. WAVE-työkalun testaus



Kuva 27. – WAVE alkunäkymä



Kuva 28. – Näkymä klikatusta ikonista

WAVE powered by WebAIM
web accessibility evaluation tool

Address: <https://www.hel.fi/fi>

Styles: OFF ON

Details

Summary Details Reference Order Structure Contrast

1 Contrast Errors
1 X Very low contrast

12 Alerts
1 X Noscript element
11 X Underlined text

28 Features
5 X Alternative text
3 X Null or empty alternative text
2 X Form label

The following apply to the entire page:

aria-labelledby="block-hdht-subtheme-footertopnavigationsecond-menu"

Helsinki-kehystunnus **aria** **role="img"** **aria-labelledby="hdht-icon-1335707341"**

- © Helsingin kaupunki 2023
- Saavutettavuusseloste
- Tietosuoja
- Tietoa Hel.fistä

Takaisin ylös **aria** **role="img"**

Kysy kaupungista!

```

<script>
  </script>
<aside id="chat-leijuke-wrapper">
  <h2 class="visually-hidden">
    Chat
  </h2>
  <button id="chat-leijuke-genesys_neuvonta" class="chat-leijuke">
    <span class="hel-icon hel-icon--speechbubble-text"></span>
    <span>
      Kysy kaupungista!
    </span>
  </button>

```

Kuva 29. – Näkymä lähdekoodista ja tyylitiedoston poistamisesta

WAVE powered by WebAIM
web accessibility evaluation tool

Address: <https://www.hel.fi/fi>

Styles: OFF ON

Reference

Summary Details Reference Order Structure Contrast

Contrast Errors
Very low contrast

What It Means
Very low contrast between text and background colors.

Why It Matters
Adequate contrast of text is necessary for all users, especially users with low vision.

What To Do
Increase the contrast between the foreground (text) color and the background color. Large text (larger than 18 point or 14 point bold) does not require as much contrast as smaller text.

The Algorithm... in English
Text is present that has a contrast ratio less than 4.5:1, or large text (larger than 18 point or 14 point bold) has a contrast ratio less than 3:1. WCAG requires that page elements have both foreground AND background colors defined (or inherited) that provide sufficient contrast. When text is presented over a background image, the text must have a background color defined (typically in CSS) that provides adequate text contrast when the background image is disabled or unavailable. WAVE does not identify contrast issues in text with CSS.

The following apply to the entire page:

aria-labelledby="block-hdht-subtheme-footertopnavigationsecond-menu"

Helsinki-kehystunnus **aria** **role="img"** **aria-labelledby="hdht-icon-1335707341"**

- © Helsingin kaupunki 2023
- Saavutettavuusseloste
- Tietosuoja
- Tietoa Hel.fistä

Takaisin ylös **aria** **role="img"**


Kysy kaupungista!

```

<script>
  </script>
<aside id="chat-leijuke-wrapper">
  <h2 class="visually-hidden">
    Chat
  </h2>
  <button id="chat-leijuke-genesys_neuvonta" class="chat-leijuke">
    <span class="hel-icon hel-icon--speechbubble-text"></span>
    <span>
      Kysy kaupungista!
    </span>
  </button>

```

Kuva 30. – Näkymä suositusvälilehdestä



powered by WebAIM

Address: <https://www.hel.fi/fi>

Styles: OFF ☐ ON ☐

Order

Summary Details Reference Order Structure Contrast

[Refresh Navigation Order](#)

[Show WAVE icons](#)

Order, role, and accessible name (what is read by a screen reader) for all navigable page elements are listed. Elements that do not have a function should not be listed.

- Link: Hyppää pääsisältöön
- Link: Koronavirus
- Link: Uutiset
- Link: Avoimet työpaikat
- Link: Osallistu
- Link: Asioi verkossa
- Link: Anna palautetta(Linkki johtaa ulkoiseen palveluun)
- Link: Helsinki-kehystunnusHelsingin kaupunki
- Link: Suomi
- Link: Svenska
- Link: English
- Link: Sulje tietoa muilla kielillä valikko

(Linkki johtaa ulkoiseen palveluun) (Linkki aloittaa puhelun) (Linkki avaa oletussähköpostiohjelman) [Hyppää pääsisältöön](#)

- [Koronavirus](#)
- [Uutiset](#)
- [Avoimet työpaikat](#)
- [Osallistu](#)
- [Asioi verkossa](#)
- [Anna palautetta\(Linkki johtaa ulkoiseen palveluun\)](#)

[Helsinki-kehystunnusHelsingin kaupunki](#)

Valitse sivuston kieli

[Suomi](#) [Svenska](#) [English](#) [Deutsch](#) [Français](#) [Russian](#) [Ukrainian](#) [Arabic](#) [Estonian](#) [Persian, farsi](#) [Español](#) [Somali](#) [Davvisámegiella](#) [Chinese, simplified](#)

[Sulje tietoa muilla kielillä valikko](#) [Sulje tietoa muilla kielillä valikko](#)

Tietoa muilla kielillä

- Tietoa muilla kielillä
- [Suomi](#) (Linkki johtaa ulkoiseen palveluun)
- [Svenska](#) (Linkki johtaa ulkoiseen palveluun)
- [English](#) (Linkki johtaa ulkoiseen palveluun)
- [Deutsch](#) (Linkki johtaa ulkoiseen palveluun)
- [Français](#) (Linkki johtaa ulkoiseen palveluun)
- [Russian](#) (Linkki johtaa ulkoiseen palveluun)
- [Ukrainian](#) (Linkki johtaa ulkoiseen palveluun)
- [Arabic](#) (Linkki johtaa ulkoiseen palveluun)
- [Estonian](#) (Linkki johtaa ulkoiseen palveluun)
- [Persian, farsi](#) (Linkki johtaa ulkoiseen palveluun)
- [Español](#) (Linkki johtaa ulkoiseen palveluun)
- [Somali](#) (Linkki johtaa ulkoiseen palveluun)
- [Davvisámegiella](#) (Linkki johtaa ulkoiseen palveluun)
- [Chinese, simplified](#) (Linkki johtaa ulkoiseen palveluun)

[Sulje tietoa muilla kielillä valikko](#) [Sulje tietoa muilla kielillä valikko](#)

Hae Helsingin kaupungin sivuilta

Mitä haet?

[Sulje](#) [Sulje](#) [Sulje](#) [Sulje](#)

[Tapahtumia](#) (Linkki johtaa ulkoiseen palveluun)


Kaupungin ajankohtaisia tapahtumia voit hakea Tapahtumat.hel.fi-sivustolla.

```

<script>
</script>
<aside id="chat-leijuke-wrapper">
  <h2 class="visually-hidden">
    Chat
  </h2>
  <button id="chat-leijuke-genesys_neuvonta" class="chat-leijuke">
    <span class="hel-icon hel-icon--speechbubble-text"></span>
    <span>
      Kysy kaupungista!
    </span>
  </button>

```

Kuva 31. – Näkymä järjestysvälilehdestä



powered by WebAIM

Address: <https://www.hel.fi/fi>

Styles: OFF ☐ ON ☐

Structure

Summary Details Reference Order Structure Contrast

Header

Navigation

Navigation

Navigation

Search

Navigation

Navigation

Navigation

The following apply to the entire page:

[Hyppää pääsisältöön](#)

[Koronavirus](#)

[Uutiset](#)

[Avoimet työpaikat](#)

[Osallistu](#)

[Asioi verkossa](#)

[Anna palautetta](#) [Anna palautetta](#) (Linkki johtaa ulkoiseen palveluun)

[Helsinki-kehystunnus](#) [Helsinki-kehystunnus](#) (Linkki johtaa ulkoiseen palveluun)

Valitse sivuston kieli


[Suomi](#) [Svenska](#) [English](#) [Deutsch](#) [Français](#) [Russian](#) [Ukrainian](#) [Arabic](#) [Estonian](#) [Persian, farsi](#) [Español](#) [Somali](#) [Davvisámegiella](#) [Chinese, simplified](#)

```

<script>
</script>
<aside id="chat-leijuke-wrapper">
  <h2 class="visually-hidden">
    Chat
  </h2>
  <button id="chat-leijuke-genesys_neuvonta" class="chat-leijuke">
    <span class="hel-icon hel-icon--speechbubble-text"></span>
    <span>
      Kysy kaupungista!
    </span>
  </button>

```

Kuva 32. – Näkymä rakennevälilehdestä



powered by [WebAIM](#)

Address: <https://www.hel.fi/fi>

Styles: OFF ☐ ON ☐

Contrast

Summary Details Reference Order Structure Contrast

Click a Contrast icon below or within the web page to view details.

Foreground Color: #FFFFFF

Lightness:

Background Color: #FFFFFF

Lightness:

Contrast Ratio: **1:1**

Text Size: **Large**

Normal Text:


AA: **Fail**

AAA: **Fail**

Large Text:


AA: **Fail**

AAA: **Fail**



 [Desaturate page](#)


Contrast is not tested when background gradients, transparency, etc. are present. A CSS background color that provides sufficient contrast must be


The following apply to the entire page:


 [fi](#)


(Linkki johtaa ulkoiseen palveluun) (Linkki aloittaa puhelun) (Linkki avaa oletussähköpostiohjelman) [Hyppää pääsisältöön](#)


 









 [Koronavirus](#)








- [Uutiset](#)
- [Avoimet työpaikat](#)
- [Osallistu](#)
- [Asioi verkossa](#)
- [Anna palautetta](#)  [aria-labelledby="hdbt-link-669337072">\(Linkki johtaa ulkoiseen palveluun\)](#)

 [aria-label="Pikalinkit"](#)

[Helsinki-kehystunnus](#) [aria](#) [role="img"](#)  [aria-labelledby="hdbt-icon-609970993">Helsingin kaupunki](#)


 Valitse sivuston kieli


 [fi](#) [Suomi](#)  [aria-current="true"](#)  [sv](#) [Svenska](#)  [en](#) [English](#)  [de](#) [Deutsch](#)  [fr](#) [Français](#)  [ru](#) [Russian](#)  [uk](#) [Ukrainian](#)

 [ar](#) [Arabic](#)  [et](#) [Estonian](#)  [fa](#) [Persian, farsi](#)  [es](#) [Español](#)  [so](#) [Somali](#)  [so](#) [Davvisámegiella](#)  [zh-hans](#) [Chinese,](#)

`quot;";}`

`</script>`

 `<aside id="chat-leijuke-wrapper">`

 `<h2 class="visually-hidden">`

`Chat`

`</h2>`

`<button id="chat-leijuke-genesys_neuvonta" class="chat-leijuke">`

``

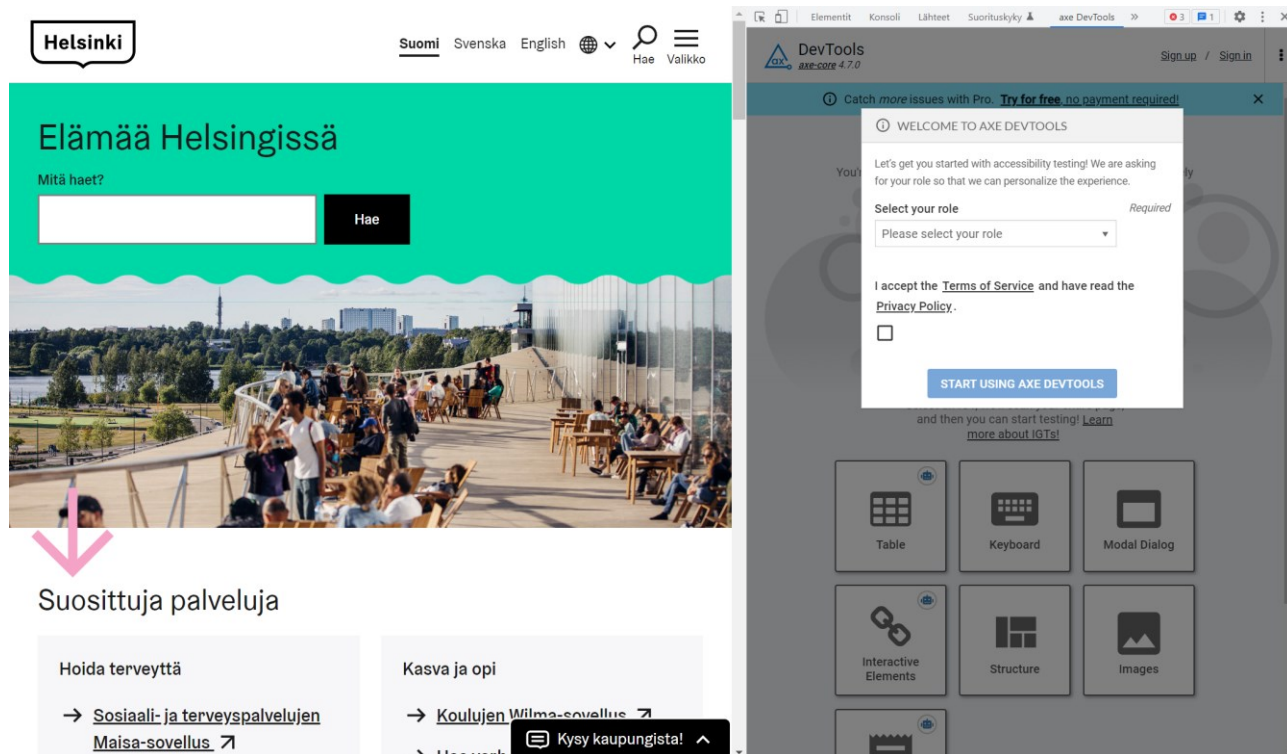
``

`Kysy kaupungista!`

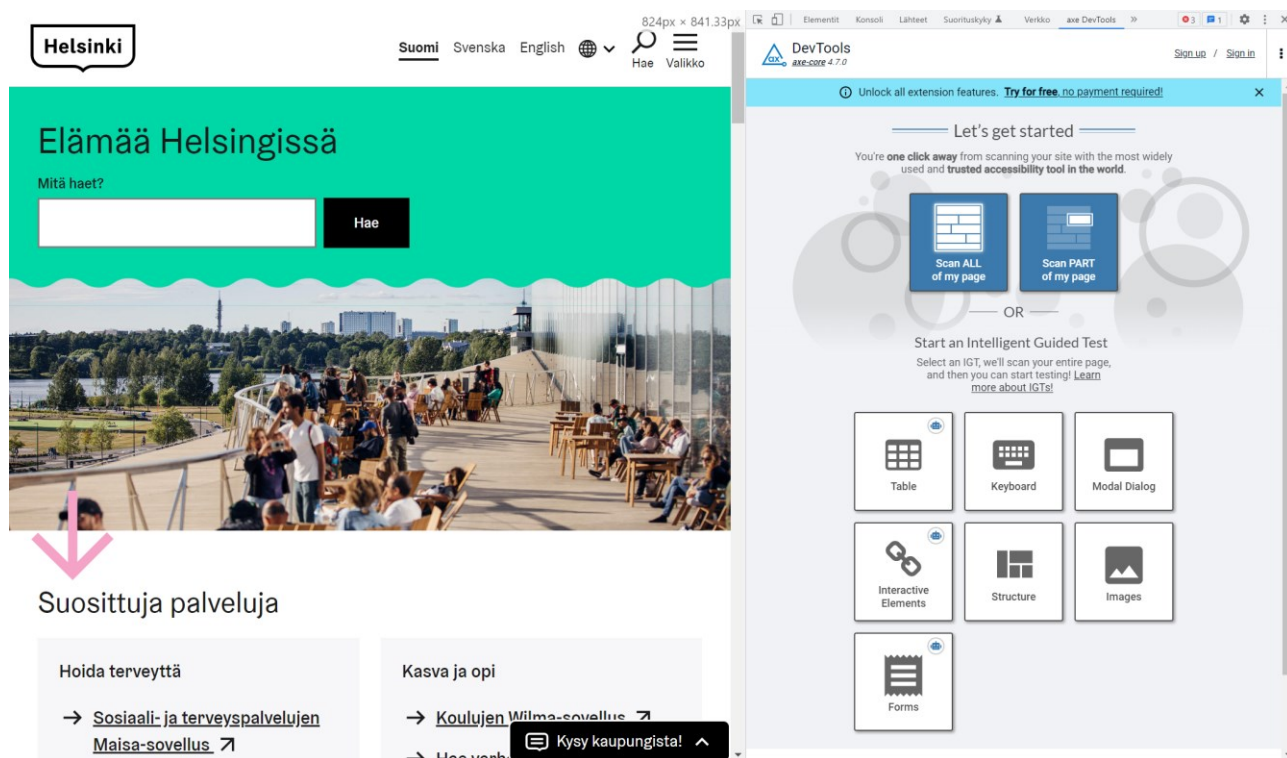
``

Kuva 33. – Näkymä kontrastiväliohdestä

Liite 2. Axe DevTools -työkalun testaus



Kuva 34. – Näkymä roolin valinnasta ennen Axe DevToolsin käyttöönottoa



Kuva 35. – Axe DevTools alkunäkymä

The image shows a screenshot of the Helsinki website (helsinki.fi) and the Chrome DevTools Accessibility DevTools panel. The website has a green header with the Helsinki logo and navigation links for Suomi, Svenska, and English. Below the header is a large image of a park with people sitting on a bench. A pink arrow points to the 'Suosittuja palveluja' (Popular services) section, which lists 'Hoida terveyttä' (Take care of your health) and 'Kasva ja opi' (Grow and learn). The DevTools panel shows the 'Overview' tab with a total of 15 issues. The issues are categorized as 15 Automatic, 0 Guided, 1 Critical, 14 Serious, and 0 Moderate. The specific issue highlighted is 'IDs used in ARIA and labels must be unique'.

Helsinki

Suomi Svenska English Hae Valikko

Elämää Helsingissä

Mitä haet?

Hae

Suosittuja palveluja

Hoida terveyttä

→ [Sosiaali- ja terveyspalvelujen Maisa-sovellus](#)

Kasva ja opi

→ [Koulujen Wilma-sovellus](#)

→ [Hae varh...](#)

Kysy kaupungista! ^

DevTools

axe-core 4.7.0

Sign up / Sign in start new scan

Overview Guided Tests

Test Name

Test URL <https://www.hel.fi/fi>

SAVE TEST RE-RUN SCAN

TOTAL ISSUES 15

AUTOMATIC ISSUES 15

GUIDED ISSUES 0

Critical 1 Serious 14

Moderate 0 Minor 0

Best Practices: OFF WCAG 2.1 AA EXPORT

Issues: Total Issues 15

IDs used in ARIA and labels must be unique 14

IDs used in ARIA and labels must be unique 1

Kuva 36. – Kokonaan verkkosivuston skannaamisen testitulokset

This image is similar to the previous one, but it shows a detailed view of the 'IDs used in ARIA and labels must be unique' issue in the DevTools panel. The issue description states: 'Ensures every id attribute value used in ARIA and in labels is unique'. The related node is shown as: `(Linkki johtaa ulkoiseen palveluun)`. The document has multiple elements referenced with ARIA with the same id attribute: hdbt-link-669337072. The related node is shown as: `(Linkki johtaa ulkoiseen palveluun)`. The issue was found automatically, is critical, and was found on 9.5.2023 at 1:22 PM.

Helsinki

Suomi Svenska English Hae Valikko

Elämää Helsingissä

Mitä haet?

Hae

Suosittuja palveluja

Hoida terveyttä

→ [Sosiaali- ja terveyspalvelujen Maisa-sovellus](#)

Kasva ja opi

→ [Koulujen Wilma-sovellus](#)

→ [Hae varh...](#)

Kysy kaupungista! ^

DevTools

axe-core 4.7.0

Sign up / Sign in start new scan

Overview Guided Tests

Test Name

Test URL <https://www.hel.fi/fi>

SAVE TEST RE-RUN SCAN

TOTAL ISSUES 15

AUTOMATIC ISSUES 15

GUIDED ISSUES 0

Critical 1 Serious 14

Moderate 0 Minor 0

Best Practices: OFF WCAG 2.1 AA EXPORT

Issues: Total Issues 15

IDs used in ARIA and labels must be unique 1

HIGHLIGHT SHARE ISSUE 1 of 1

Ensures every id attribute value used in ARIA and in labels is unique

more information

Element Location: `item:nth-child(6) > .menu__link-wapper > .menu__link[href="#palautteet-he...l.fi/"] [data-is-external="true"] > .is-hidden`

`(Linkki johtaa ulkoiseen palveluun)`

To solve this problem, you need to fix the following:

Document has multiple elements referenced with ARIA with the same id attribute: hdbt-link-669337072

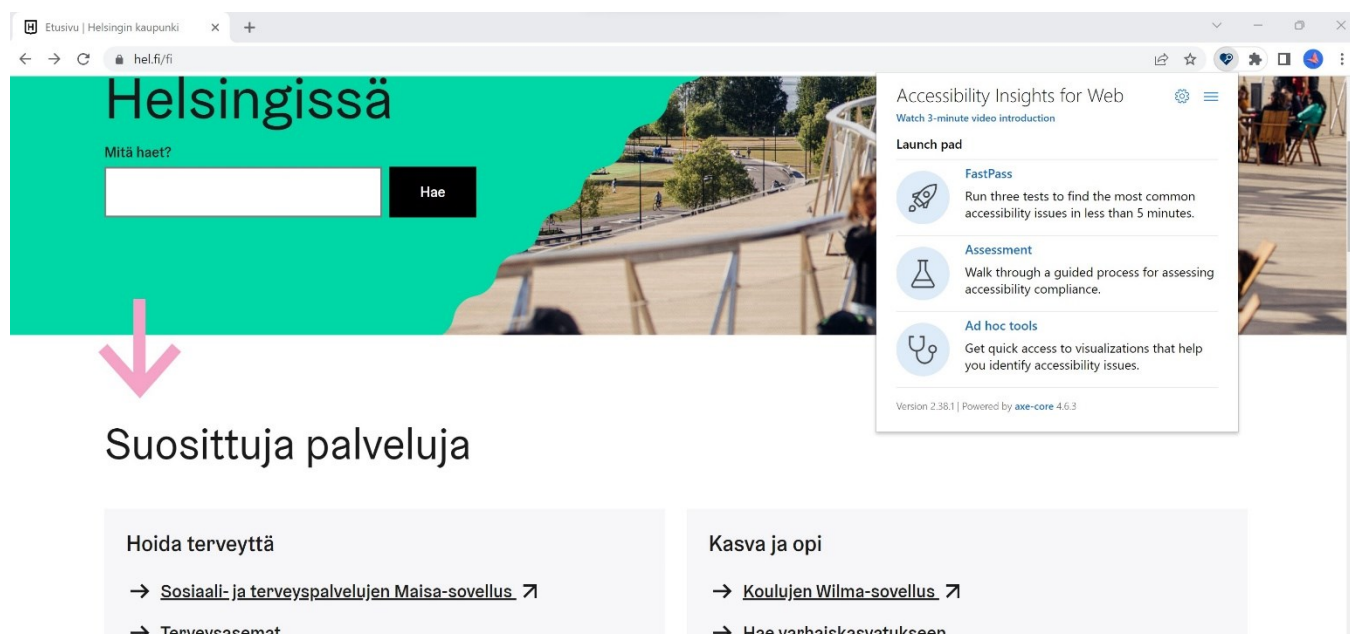
Related Node

`(Linkki johtaa ulkoiseen palveluun)`

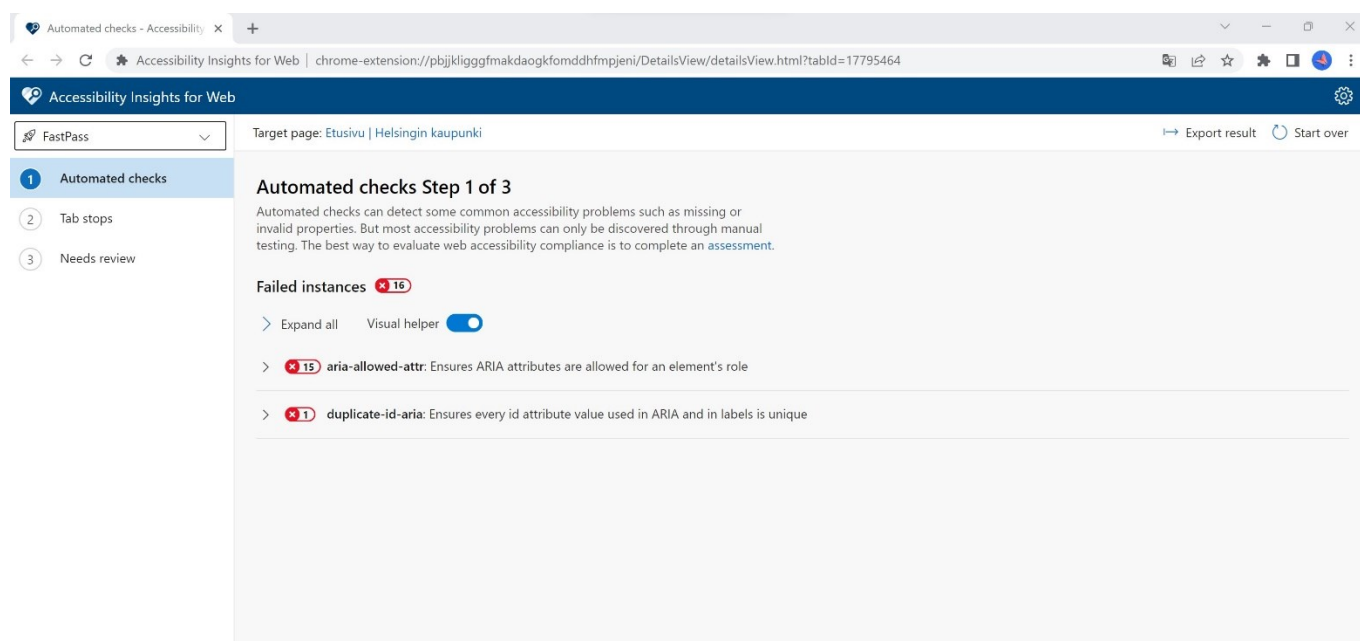
Found: Automatically Impact: critical cat.parsing wcag2a wcag411 Found on: 9.5.2023 at 1:22 PM

Kuva 37. – Lisätietoa yksittäisestä saavutettavuusongelmasta

Liite 3. Accessibility Insights for Web -työkalun testaus



Kuva 38. – Accessibility Insights for Web -työkalun valinta



Kuva 39. – Näkymä työkalun testituloksista

Automated checks - Accessibility x +

Accessibility Insights for Web | chrome-extension://pbjklgggfmakdaogkfomddhfmjpeni/DetailsView/detailsView.html?tabId=17795464

Accessibility Insights for Web

FastPass Target page: Etusivu | Helsingin kaupunki Export result Start over

1 Automated checks

2 Tab stops

3 Needs review

Failed instances **16**

▼ Collapse all Visual helper ☒

▼ **15** aria-allowed-attr: Ensures ARIA attributes are allowed for an element's role

Resources for this rule

[More information about aria-allowed-attr](#)

WCAG 4.1.2

Path .header-top > nav[aria-label="Pikalinkit"] > .menu--level-0.menu > .menu__item:nth-child(6) > .menu__link-wrapper > .menu__link[href\$="palautteet.hel.fi/"]([data-is-external="true"]) > .link__type.link_type--external[aria-labelledby="hdbt-link-669337072"]

Snippet

How to fix Fix the following:

- aria-labelledby attribute cannot be used on a span with no valid role attribute.

Highlight visible File issue Copy failure details

Path span[aria-labelledby="hdbt-link-1230112606"]

Snippet

How to fix Fix the following:

- aria-labelledby attribute cannot be used on a span with no valid role attribute.

Highlight visible File issue Copy failure details

Kuva 40. – Listattu lisätietoa aria-allowed-attr -saavutettavuusongelmaan liittyen

Automated checks - Accessibility x +

Accessibility Insights for Web | chrome-extension://pbjklgggfmakdaogkfomddhfmjpeni/DetailsView/detailsView.html?tabId=17795464

Accessibility Insights for Web

FastPass Target page: Etusivu | Helsingin kaupunki Export result Start over

1 Automated checks

2 Tab stops

3 Needs review

Path span[aria-labelledby="hdbt-link-122554316"]

Snippet

How to fix Fix the following:

- aria-labelledby attribute cannot be used on a span with no valid role attribute.

Highlight visible File issue Copy failure details

Path span[aria-labelledby="hdbt-link-1156825206"]

Snippet

How to fix Fix the following:

- aria-labelledby attribute cannot be used on a span with no valid role attribute.

Highlight visible File issue Copy failure details

Path span[aria-labelledby="hdbt-link-204617189"]

Snippet

How to fix Fix the following:

- aria-labelledby attribute cannot be used on a span with no valid role attribute.

Highlight visible File issue Copy failure details

Kuva 41. – Listattu lisätietoa aria-allowed-attr -saavutettavuusongelmaan liittyen

The screenshot shows the 'Automated checks' tab in the Accessibility Insights for Web extension. The target page is 'Etusivu | Helsingin kaupunki'. Three failed checks are listed, all related to the rule 'aria-labelledby attribute cannot be used on a span with no valid role attribute'.

Path	Snippet	How to fix
span[aria-labelledby="hdbt-link-219337737"]		Fix the following: • aria-labelledby attribute cannot be used on a span with no valid role attribute.
span[aria-labelledby="hdbt-link-1288818423"]		Fix the following: • aria-labelledby attribute cannot be used on a span with no valid role attribute.
span[aria-labelledby="hdbt-link-115797966"]		Fix the following: • aria-labelledby attribute cannot be used on a span with no valid role attribute.

Each entry includes a 'Highlight visible' button and links for 'File issue' and 'Copy failure details'.

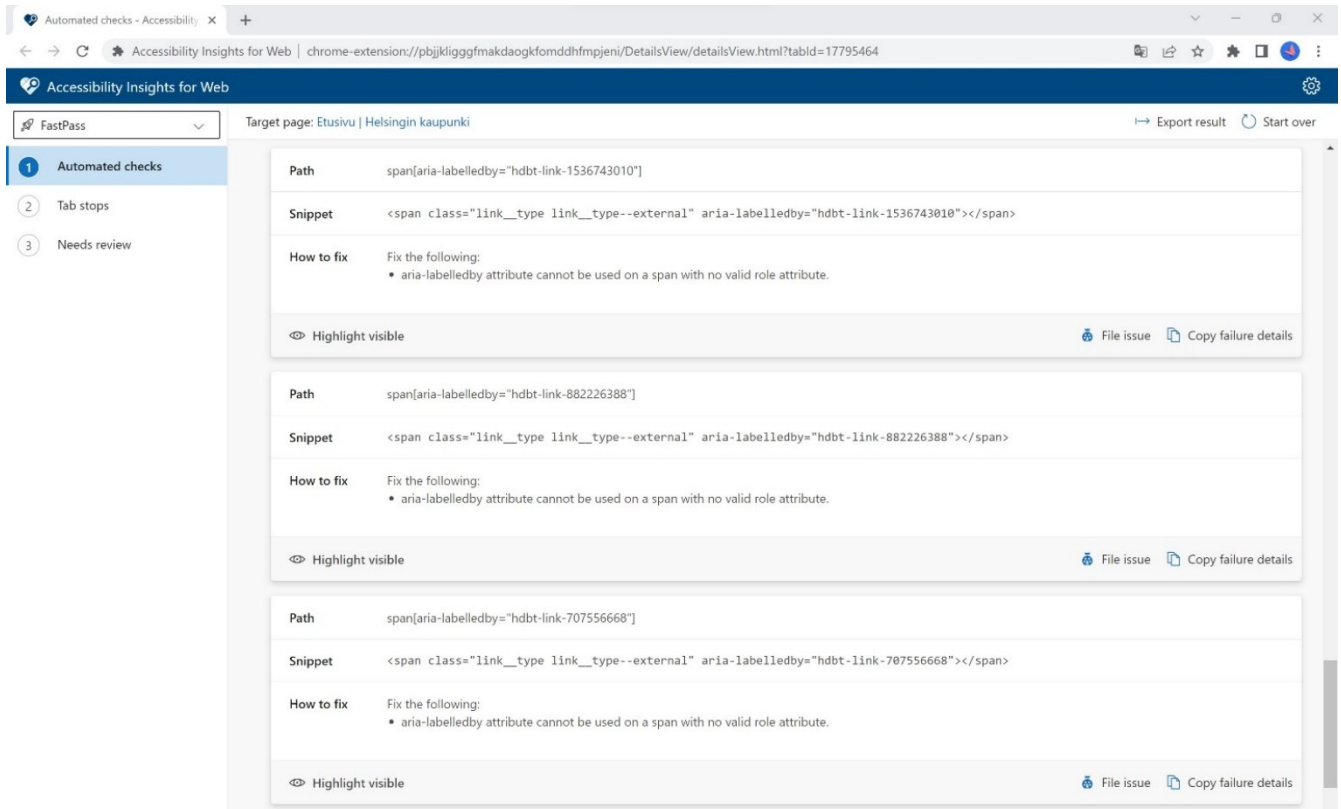
Kuva 42. – Listattu lisätietoa aria-allowed-attr -saavutettavuusongelmaan liittyen

The screenshot shows the 'Automated checks' tab in the Accessibility Insights for Web extension. The target page is 'Etusivu | Helsingin kaupunki'. Three failed checks are listed, all related to the rule 'aria-labelledby attribute cannot be used on a span with no valid role attribute'.

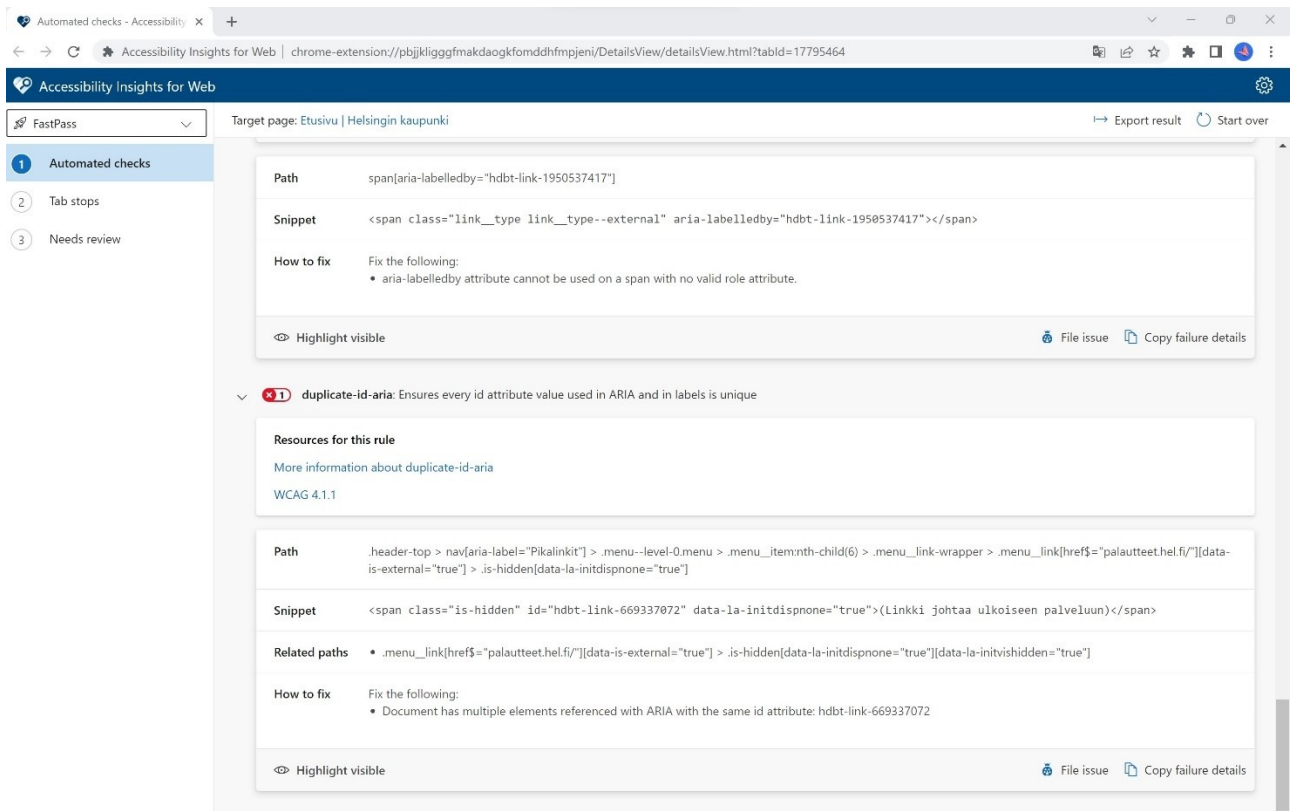
Path	Snippet	How to fix
span[aria-labelledby="hdbt-link-2069748504"]		Fix the following: • aria-labelledby attribute cannot be used on a span with no valid role attribute.
span[aria-labelledby="hdbt-link-1049101908"]		Fix the following: • aria-labelledby attribute cannot be used on a span with no valid role attribute.
span[aria-labelledby="hdbt-link-1866153743"]		Fix the following: • aria-labelledby attribute cannot be used on a span with no valid role attribute.

Each entry includes a 'Highlight visible' button and links for 'File issue' and 'Copy failure details'.

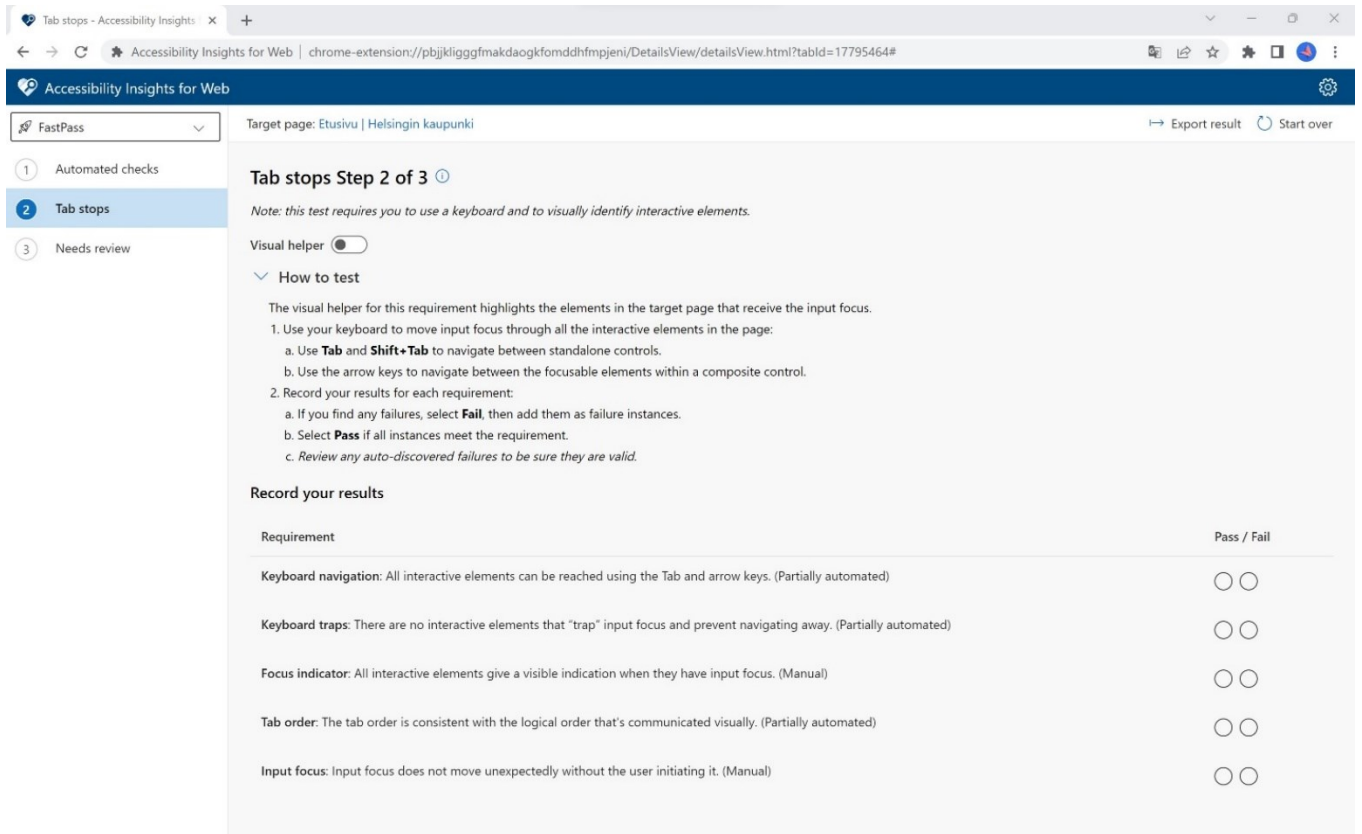
Kuva 43. – Listattu lisätietoa aria-allowed-attr -saavutettavuusongelmaan liittyen



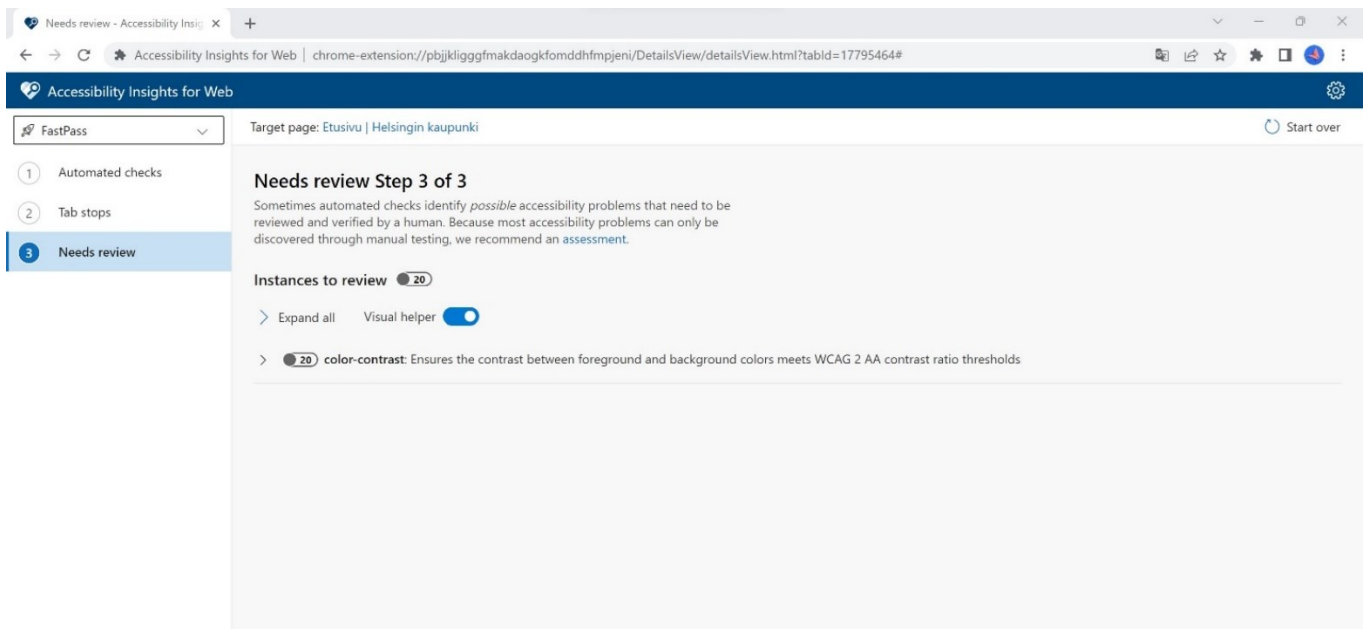
Kuva 44. – Listattu lisätietoa aria-allowed-attr -saavutettavuusongelmaan liittyen



Kuva 45. – Lisätietoa toistuvasta id-attribuutista eri elementeillä

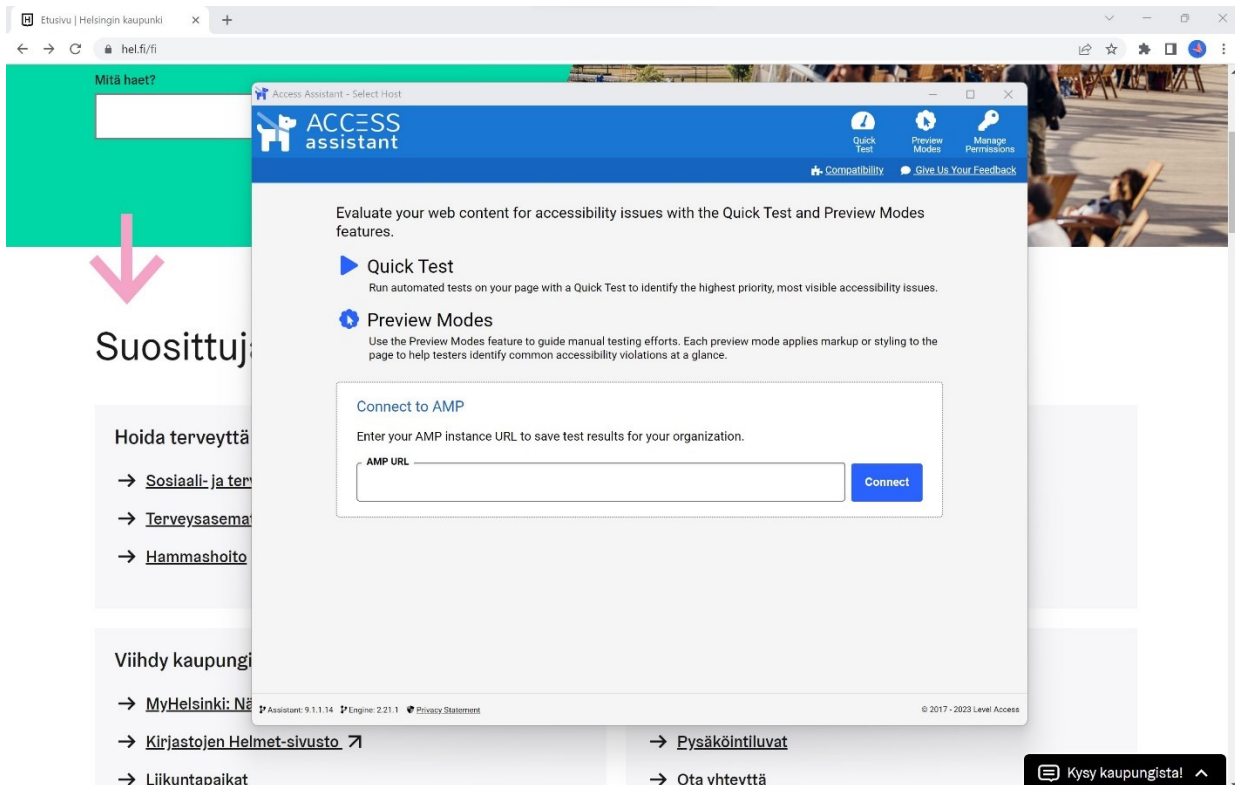


Kuva 46. – Näkymä työkalun toisesta askeleesta liittyen interaktiivisen elementtien läpikäyntiin

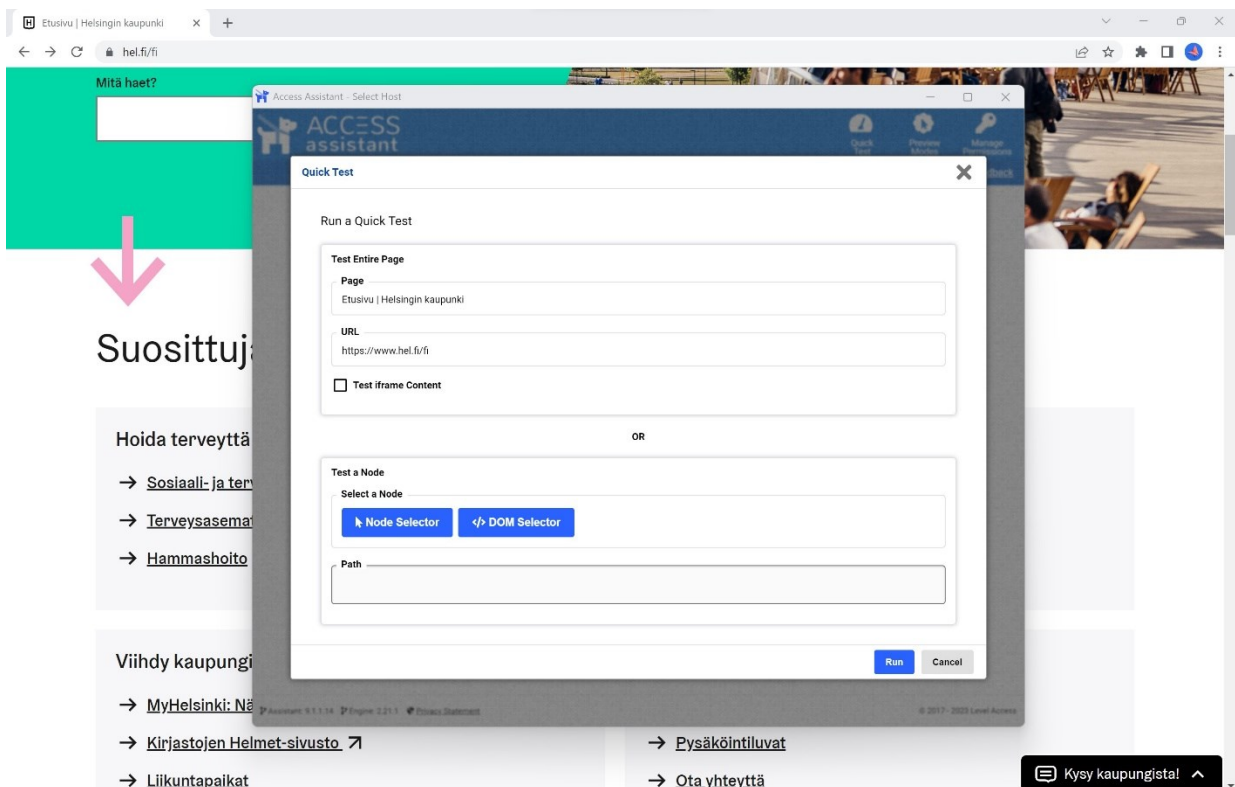


Kuva 47. – Näkymä työkalun kolmannesta askeleesta liittyen manuaaliseen tarkistukseen

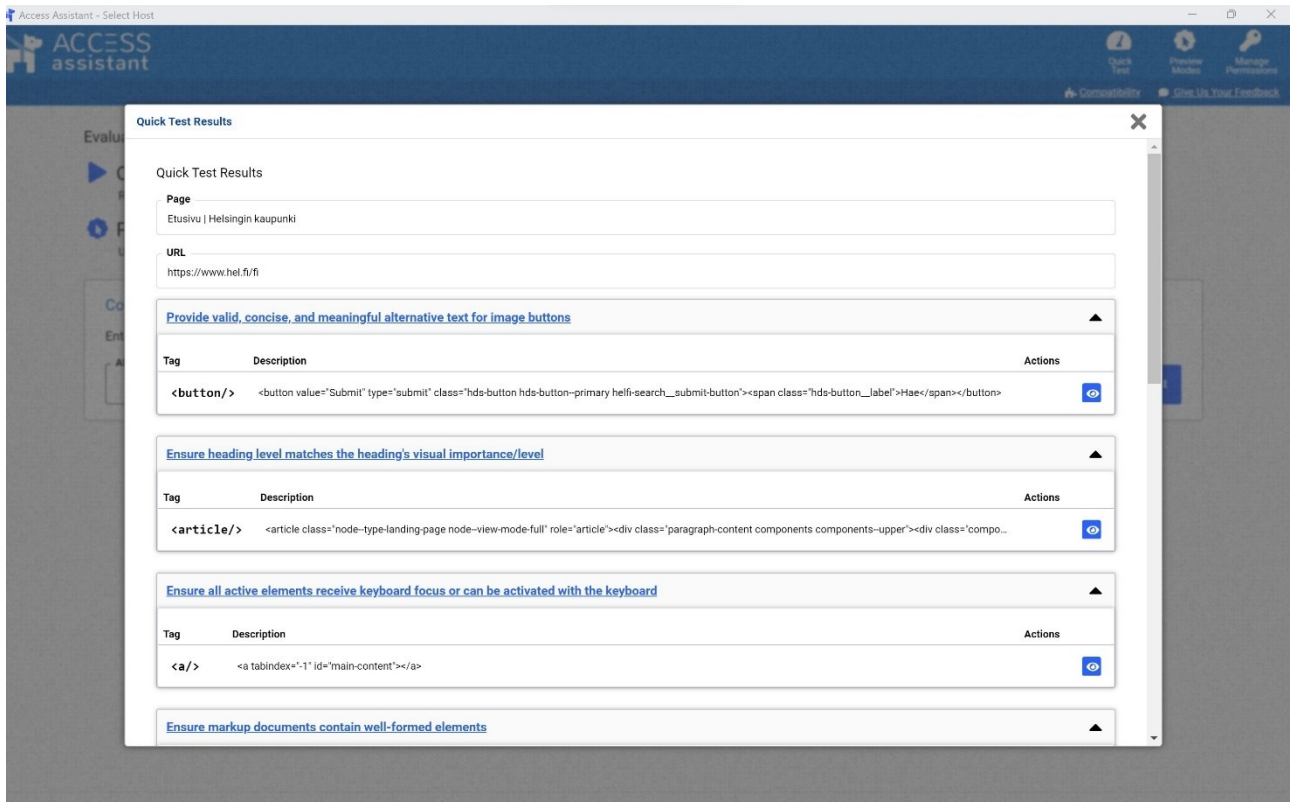
Liite 4. Access Assistant -työkalun testaus



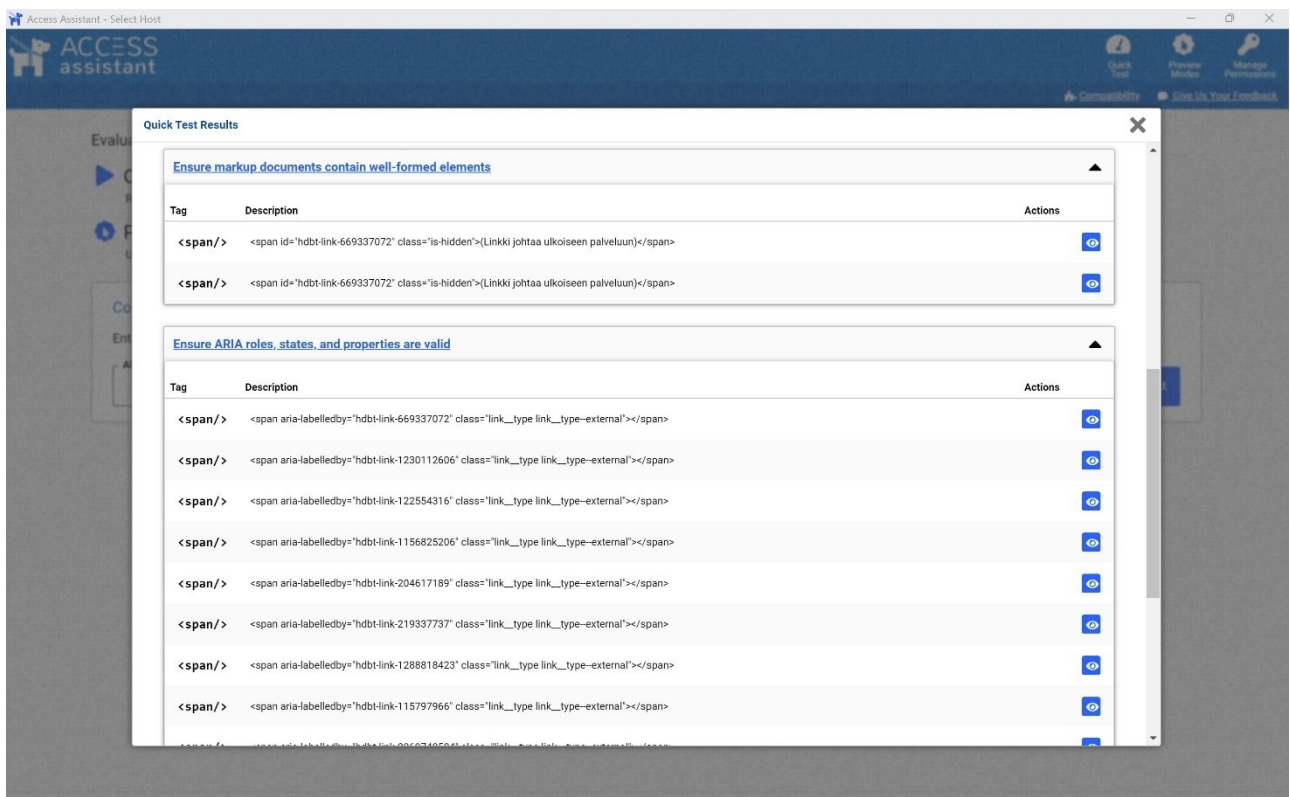
Kuva 48. – Access Assistantin alkunäkymä



Kuva 49. – Näkymä Quick Test ikkunasta

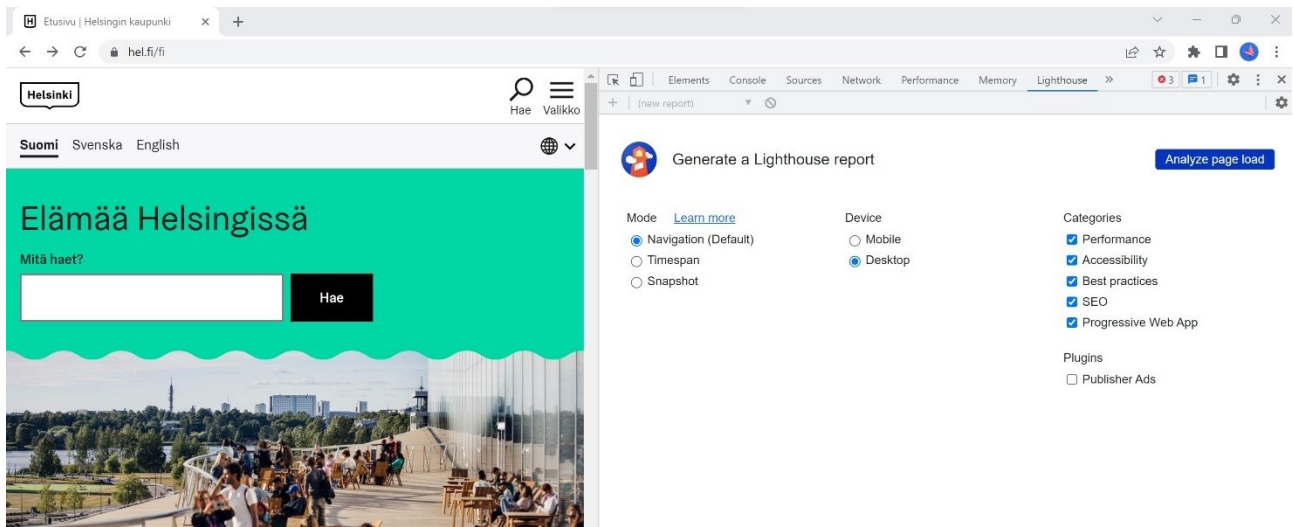


Kuva 50. – Näkymä testiraportista

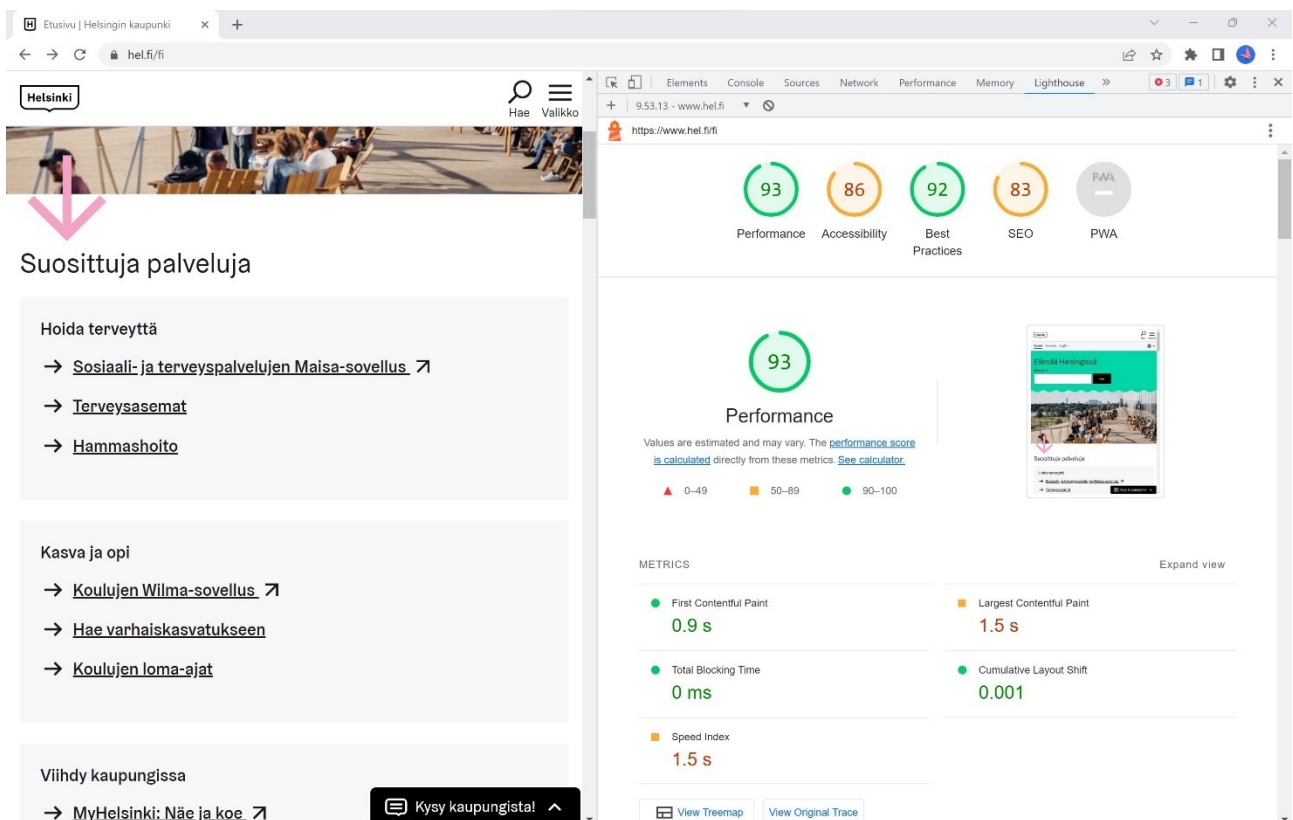


Kuva 51. – Näkymä testiraportista

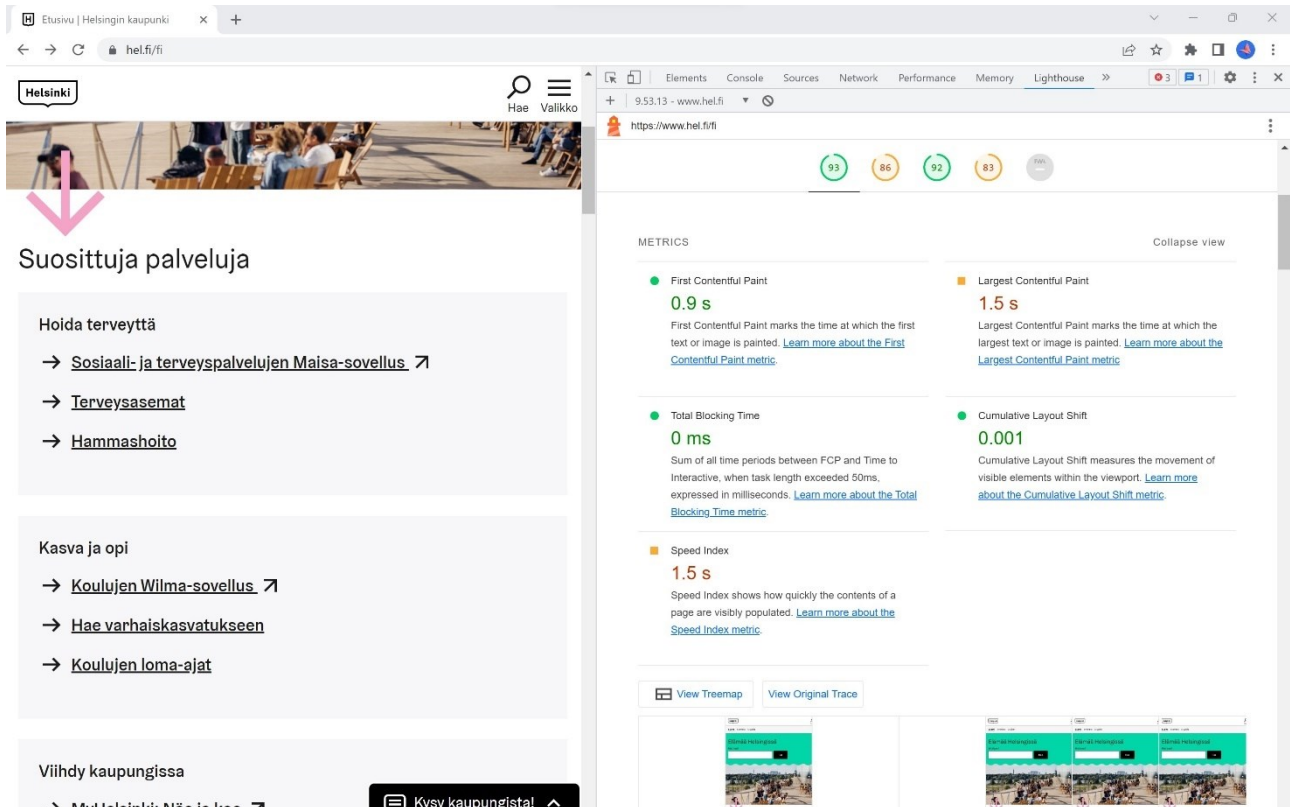
Liite 5. Google Lighthouse -työkalun testaus



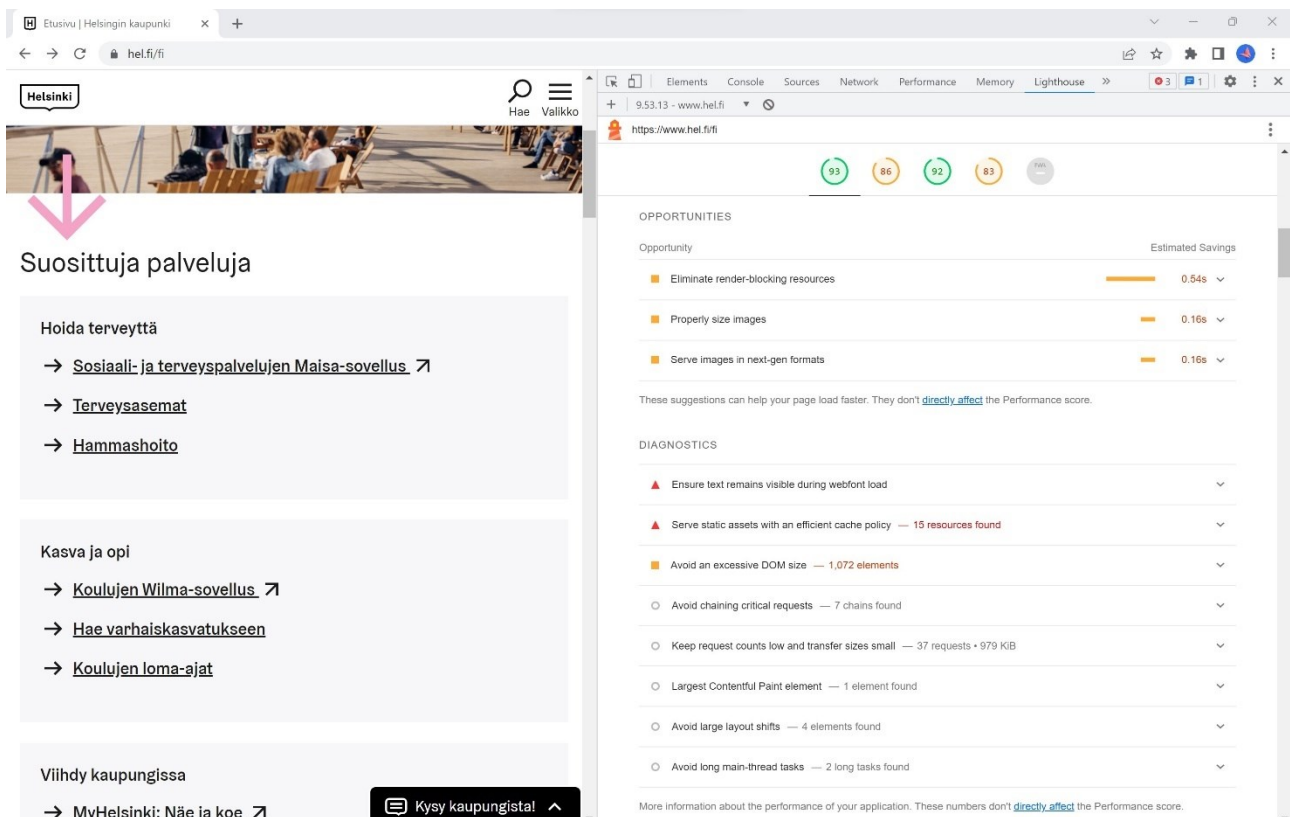
Kuva 52. – Alkunäkymä ennen testin suorittamista



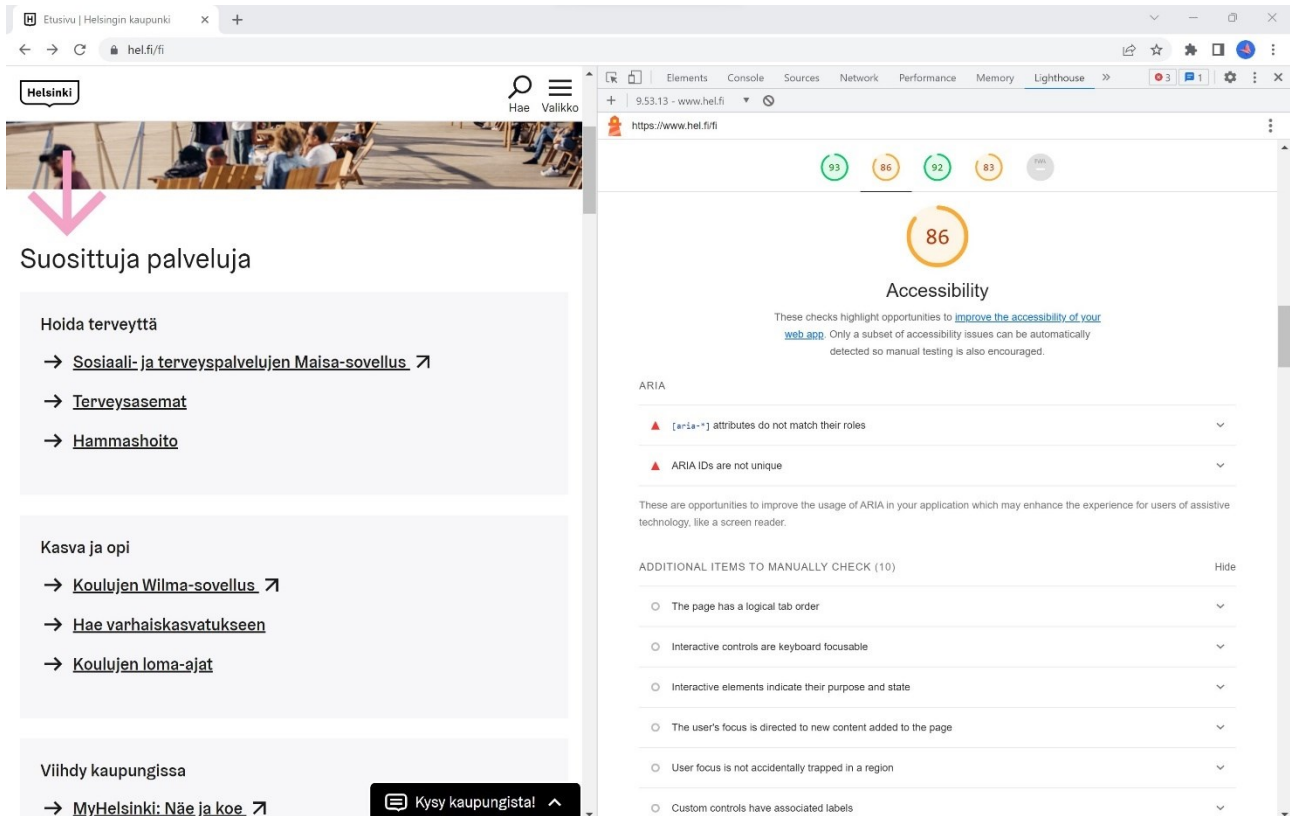
Kuva 53. – Näkymä testituloksista



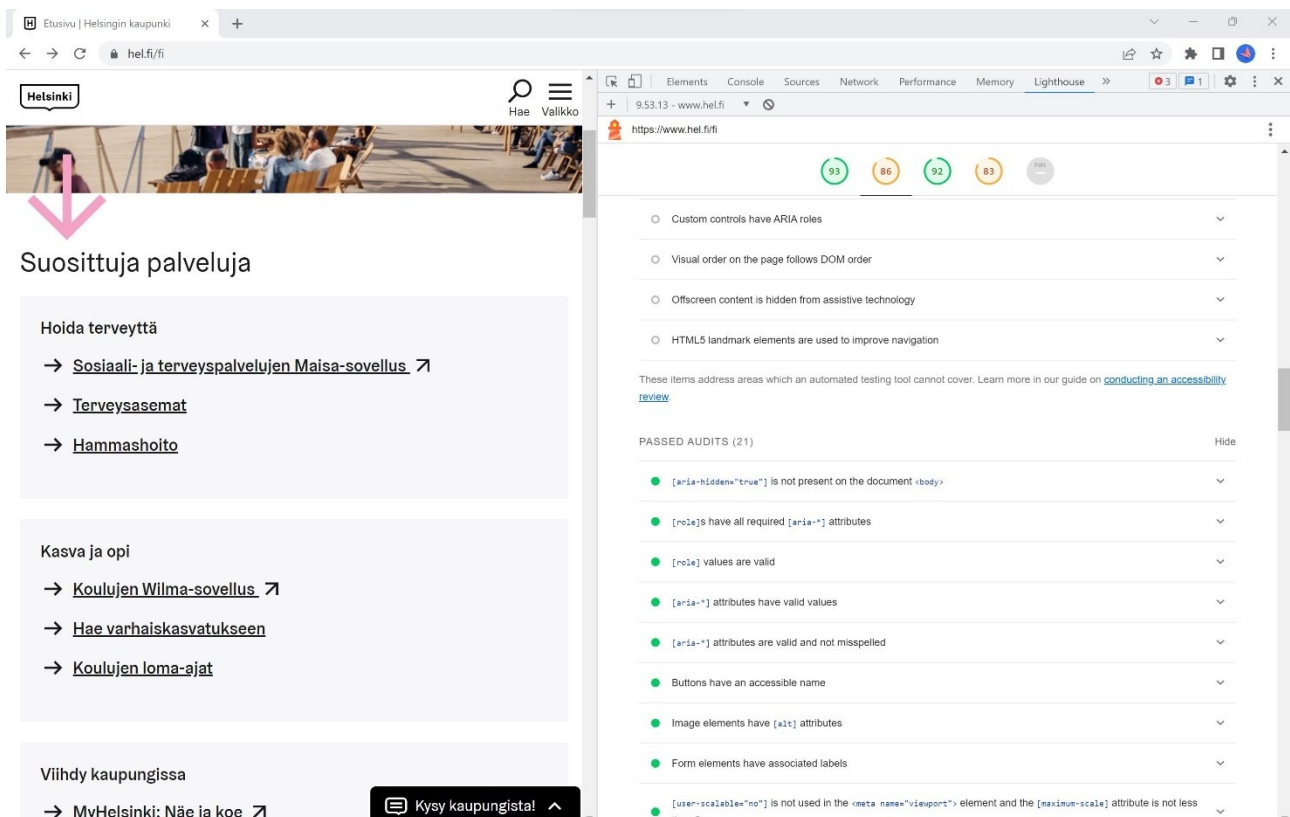
Kuva 54. – Näkymä tarkemmasta testitulokseen ja suorituskypisteisiin vaikuttavista mittareista



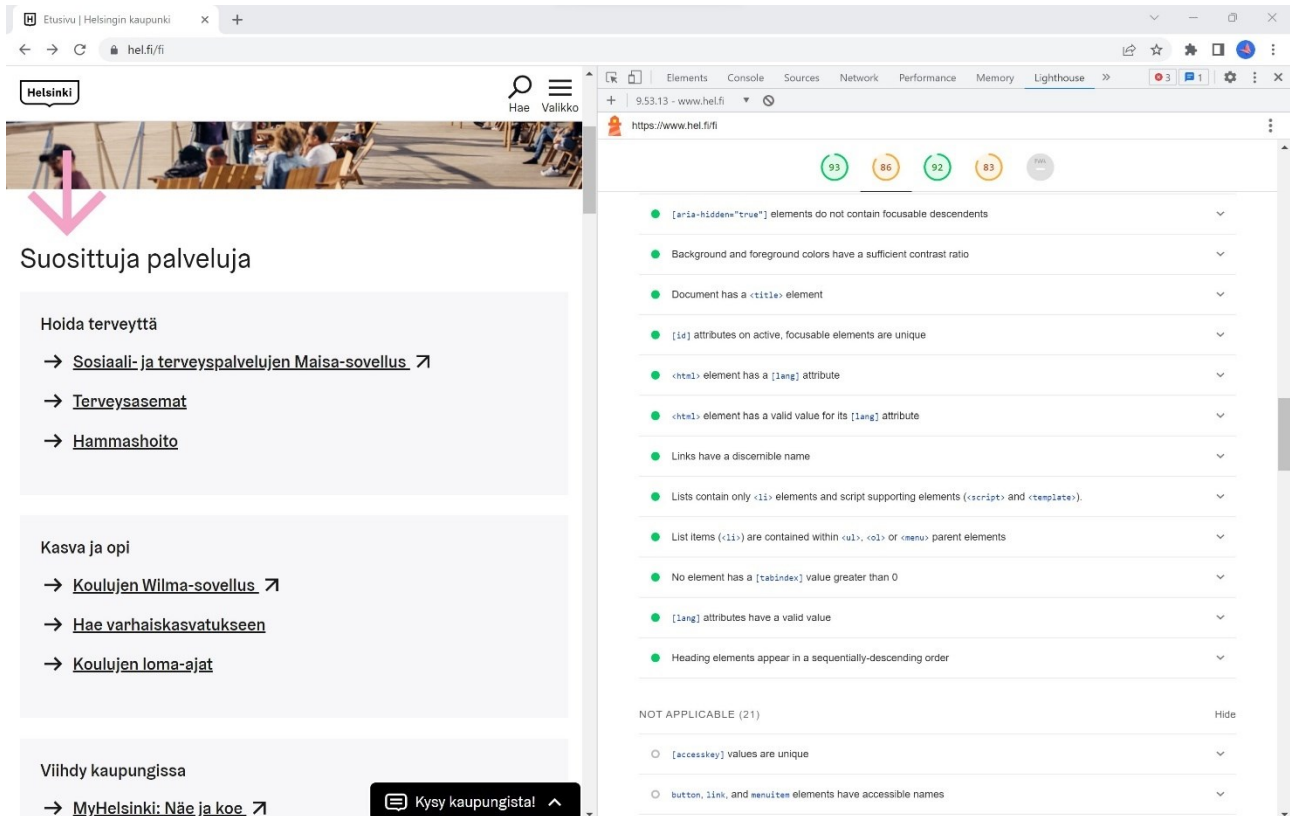
Kuva 55. – Lisätietoa ja ehdotuksia löytämistä ongelmista



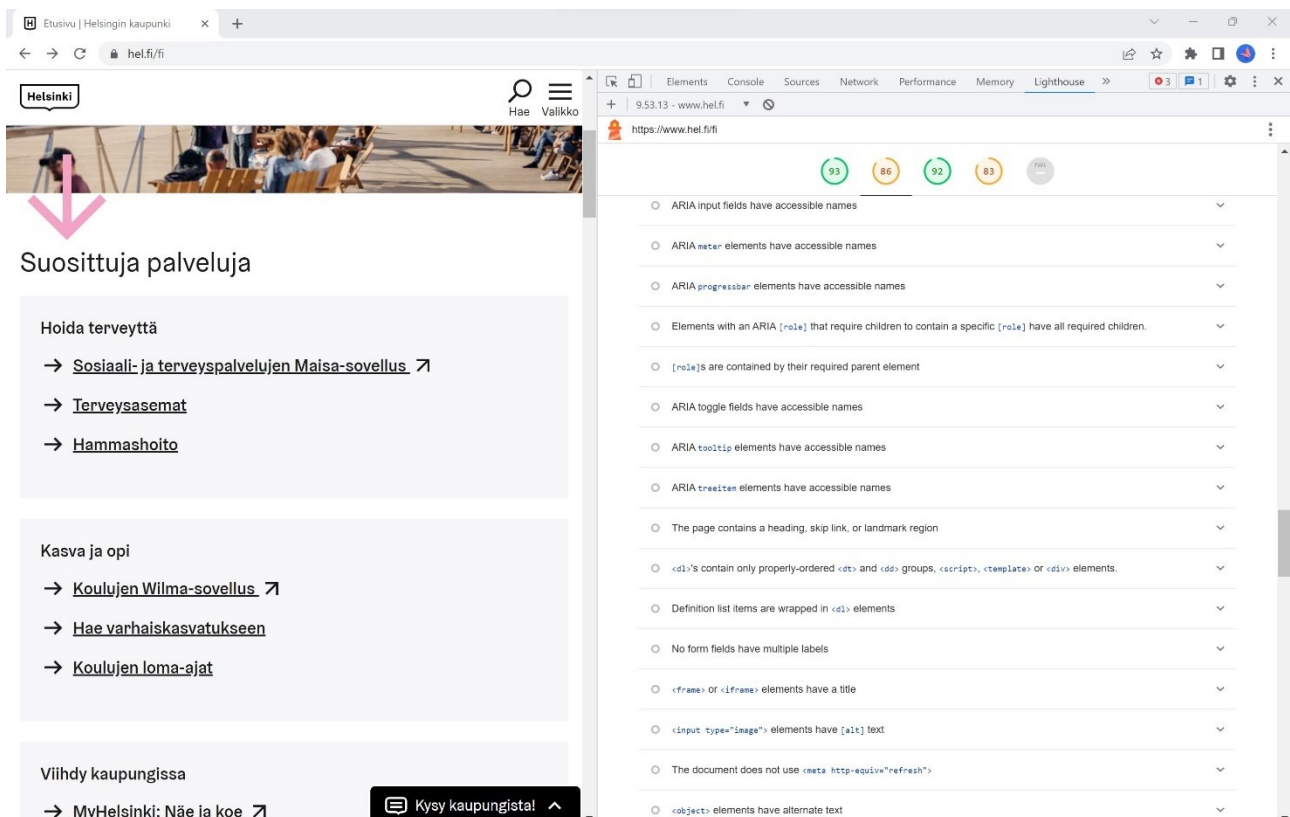
Kuva 56. – Raportti saavutettavuusongelmista ja manuaalisesti tarkistettavista testeistä



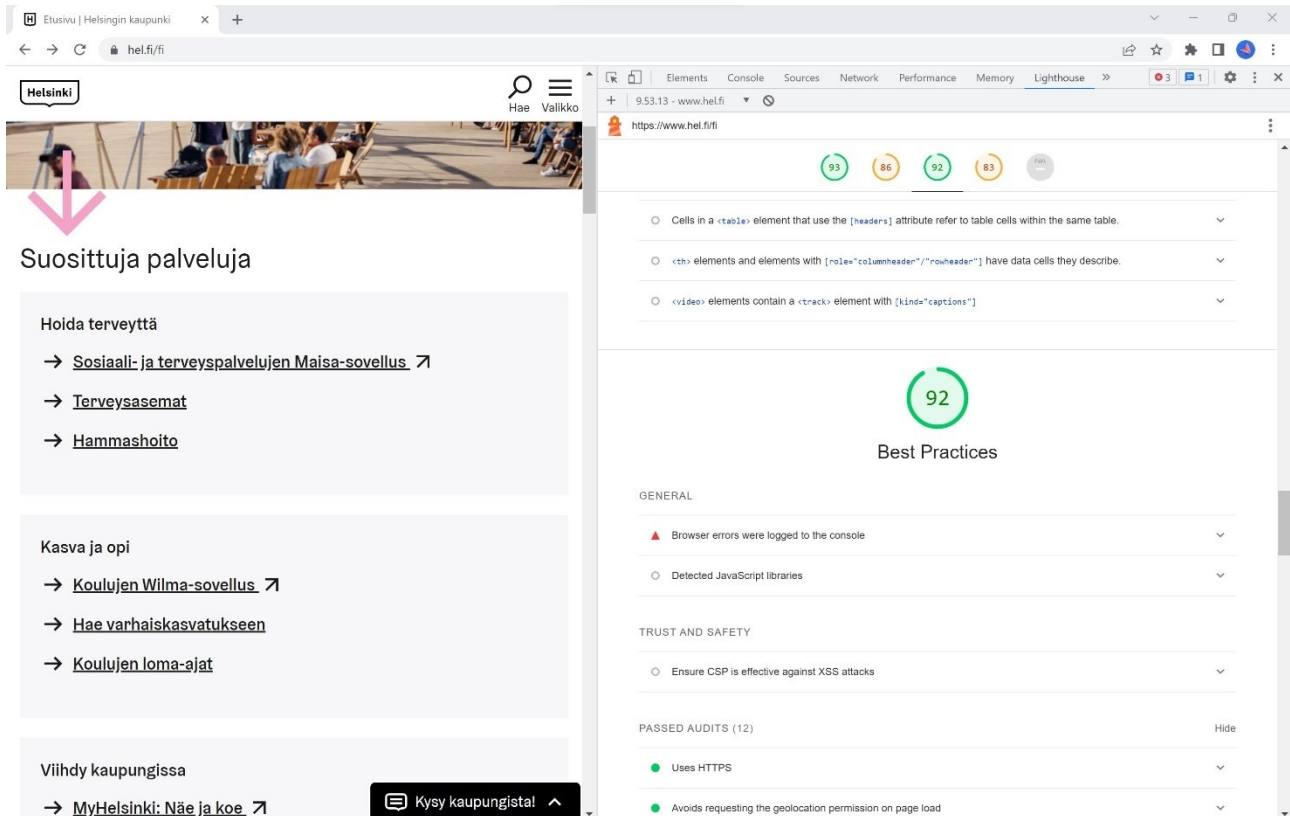
Kuva 57. – Näkymä hyväksytysti läpäistyistä saavutettavuustesteistä



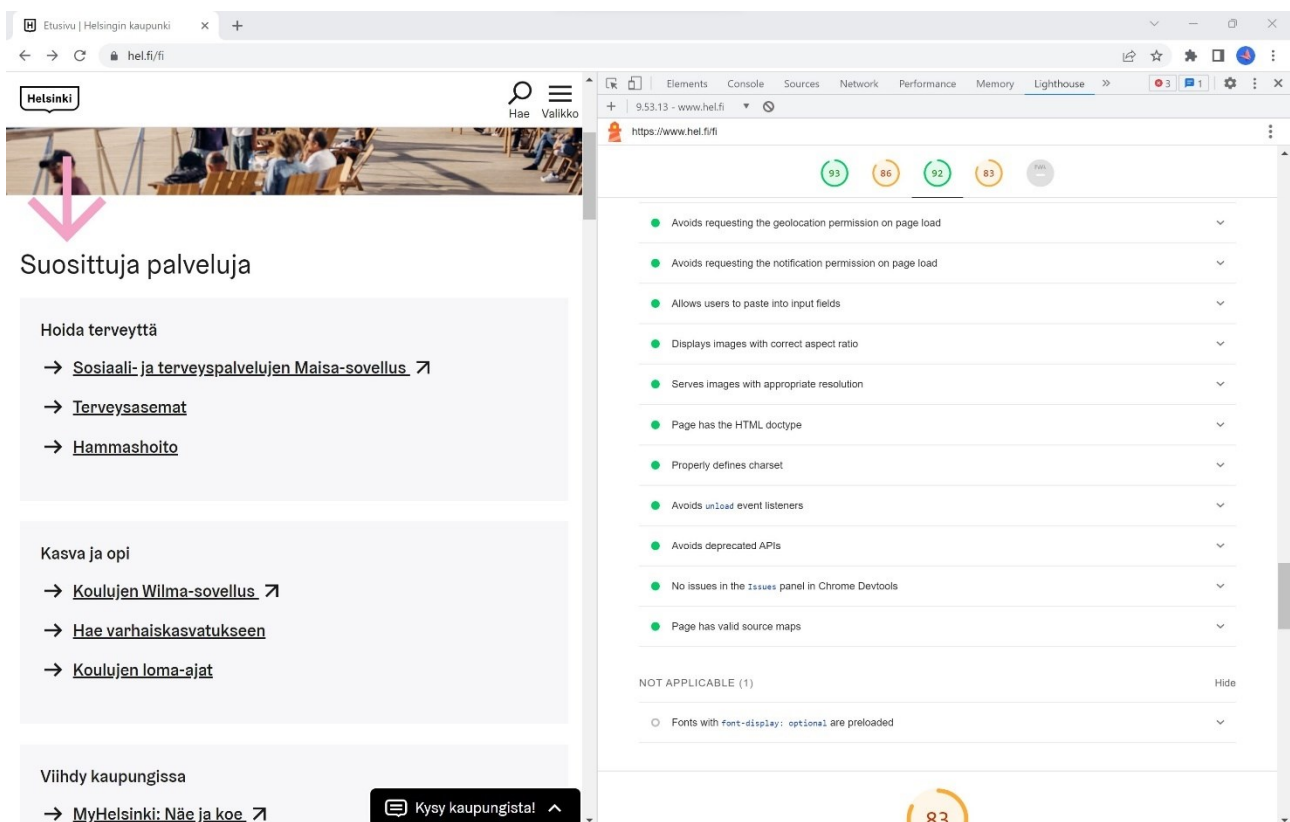
Kuva 58. – Näkymä hyväksytysti läpäisseistä testeistä ja ei-soveltuvista testeistä



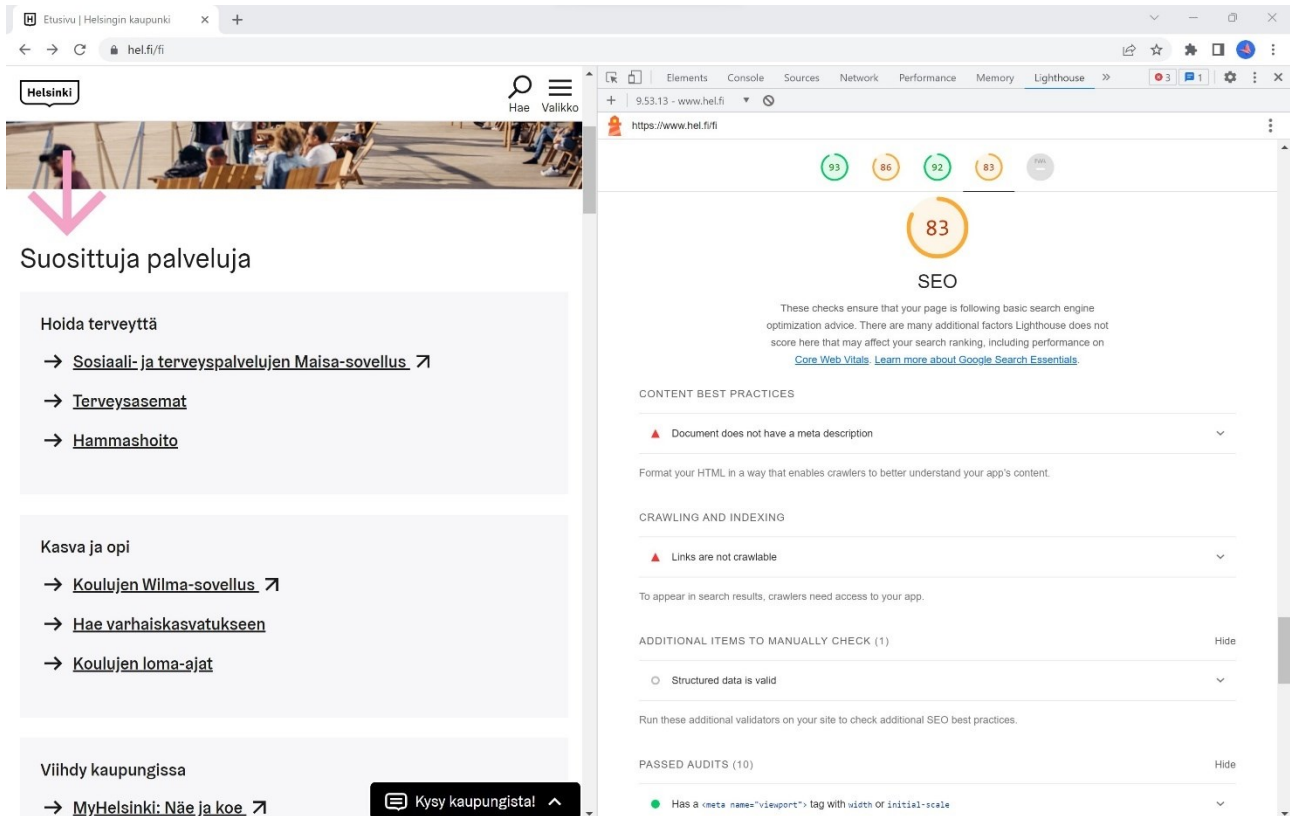
Kuva 59. – Lisää verkkosivustolla olevia ei-soveltuvia testejä



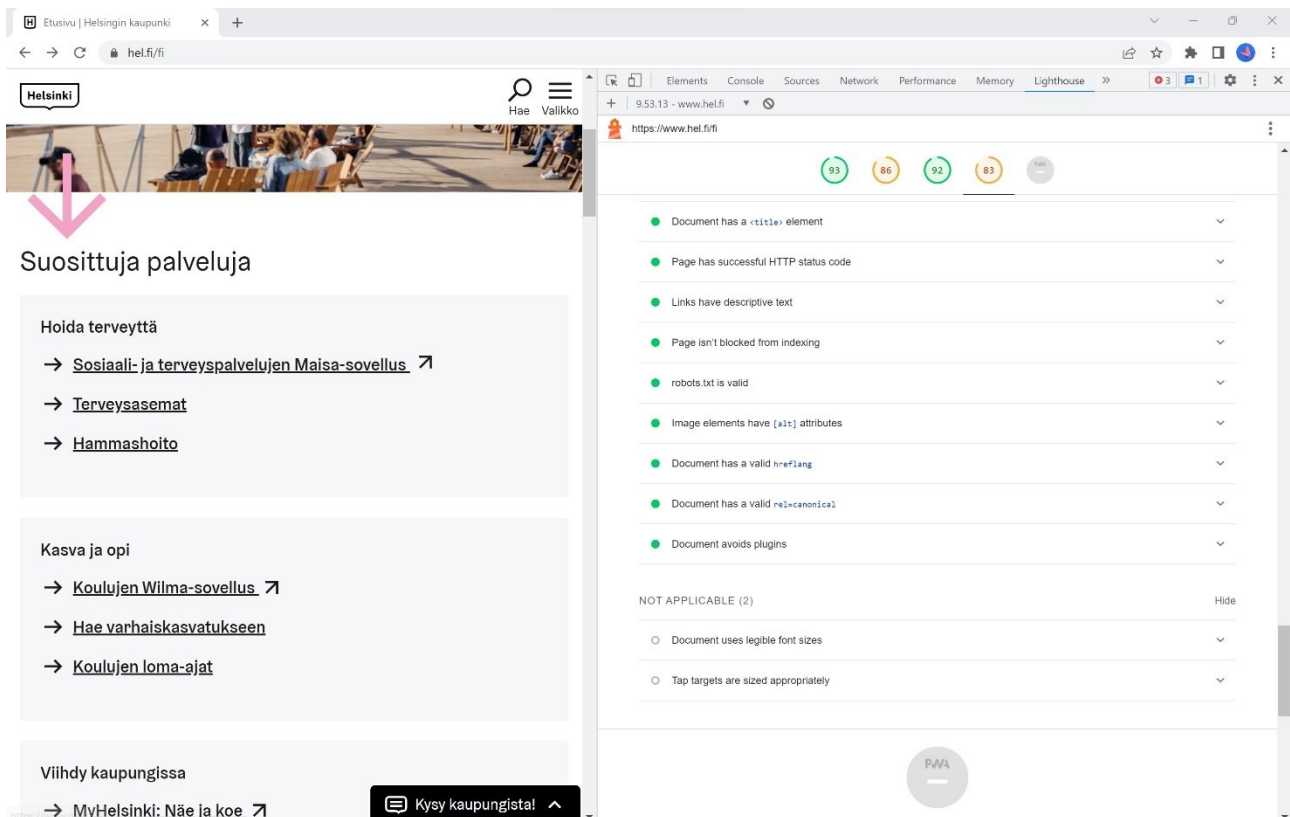
Kuva 60. – Testitulokset ”Best Practices”-osiosta



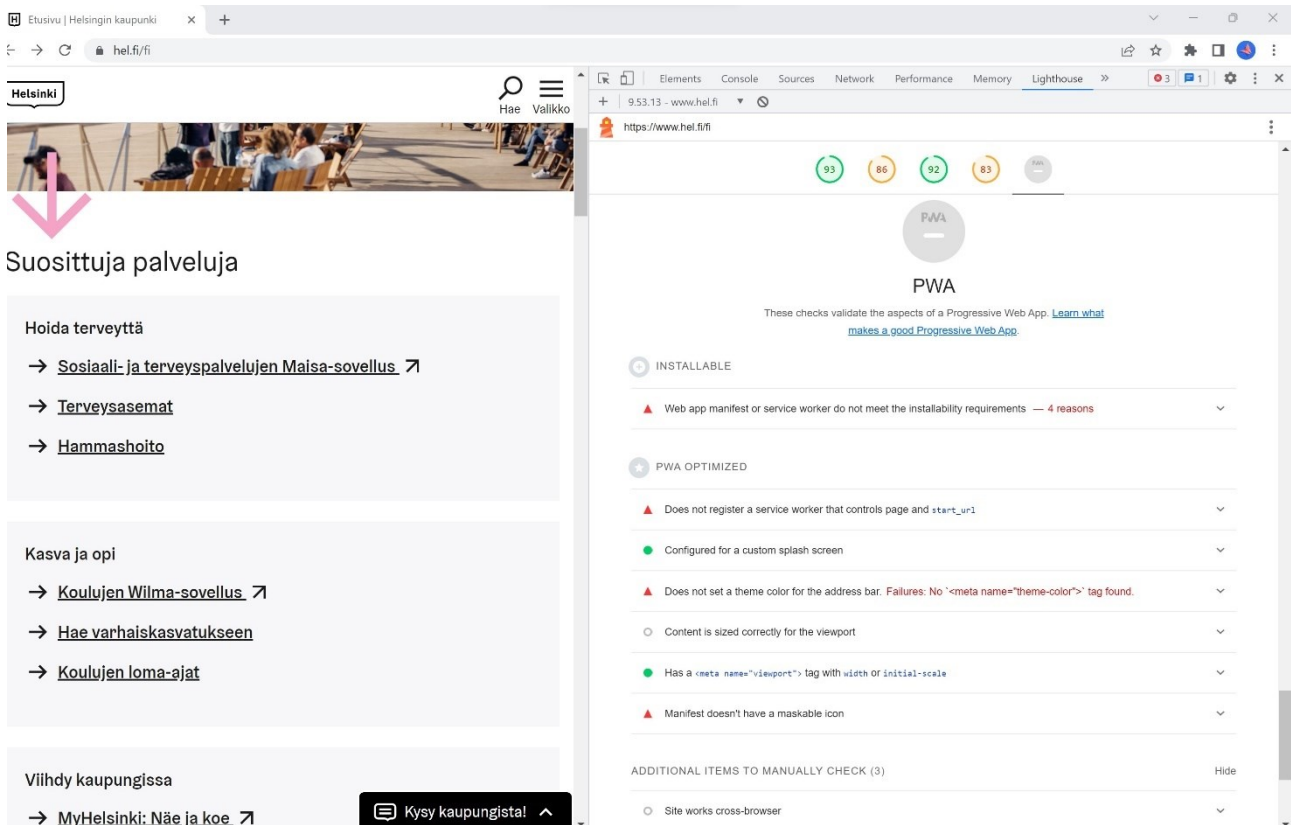
Kuva 61. – ”Best Practices” -osion hyväksytysti läpäisseet testit



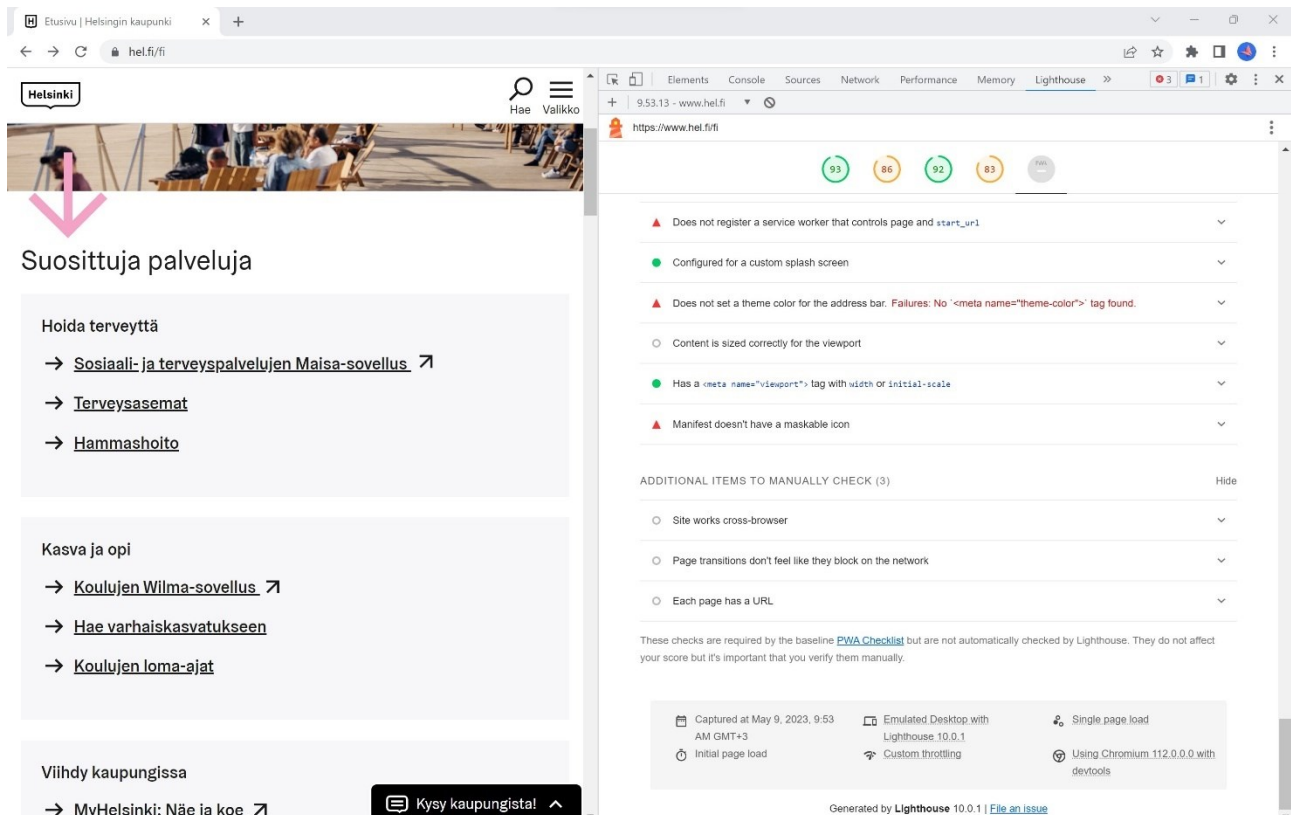
Kuva 62. – Hakukoneoptimoinnin testitulokset



Kuva 63. – Hyväksytysti läpäisseet hakukoneoptimoinnin testitulokset

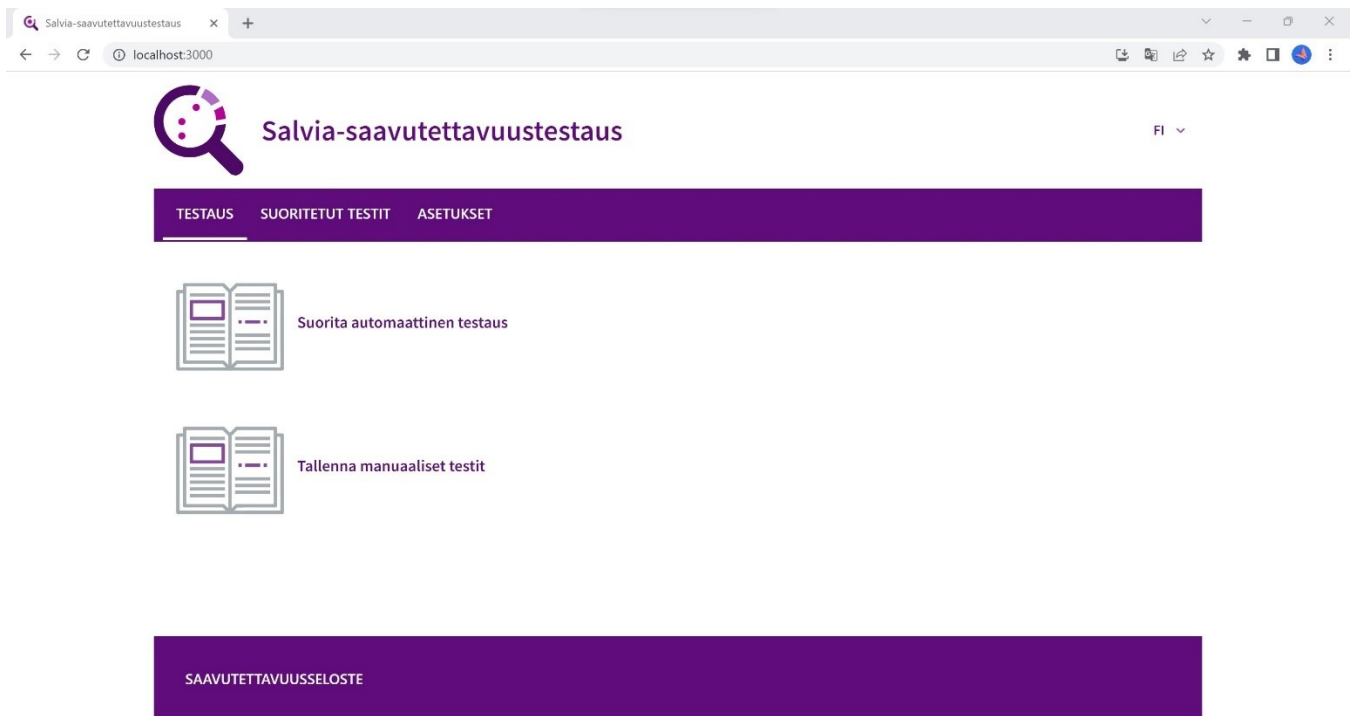


Kuva 64. – Progressiivisuuden (PWA) testitulokset

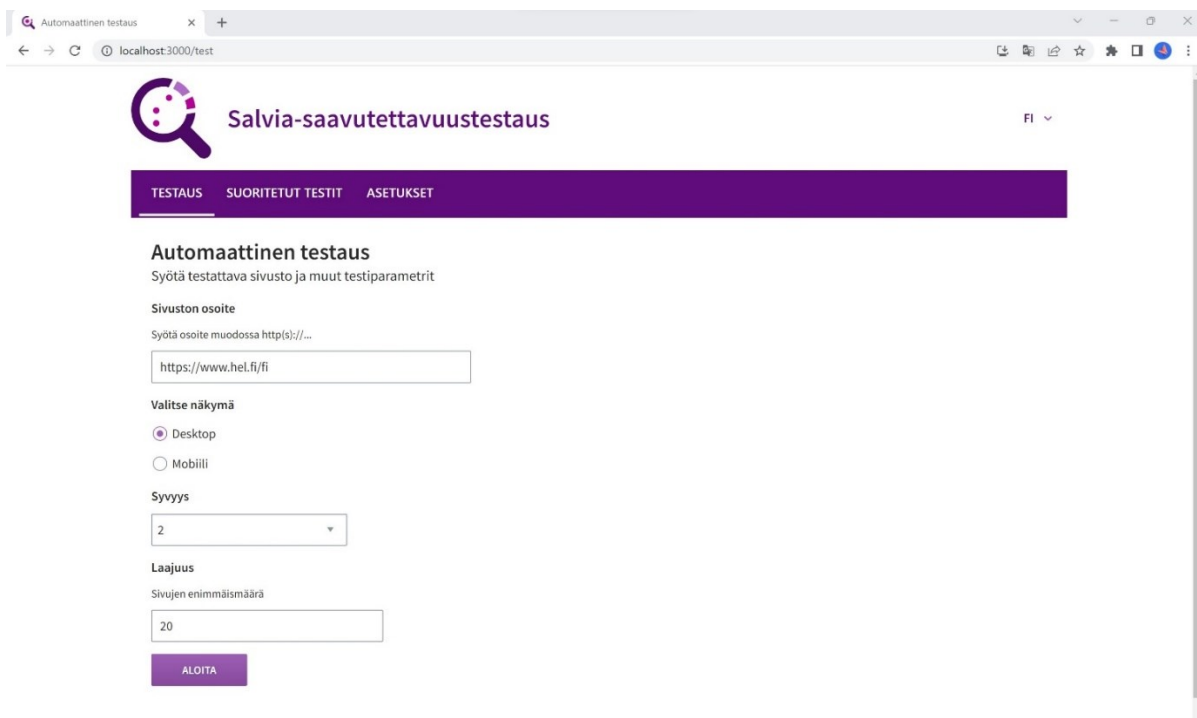


Kuva 65. – Lisää progressiivisuuden testituloksista

Liite 6. Salvia-työkalun testaus



Kuva 66. – Salvian alkunäkymä



Kuva 67. – Näkymä ennen automaattisen testin suorittamista

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/completed-tests'. The application header features a magnifying glass icon and the text 'Salvia-saavutettavuustestaus'. A navigation bar contains three tabs: 'TESTAUS', 'SUORITETUT TESTIT' (which is active), and 'ASETUKSET'. Below the navigation bar, the section title 'Suoritetut saavutettavuustestit (1)' is displayed. A search prompt 'Hae testejä verkkosivuston tunnuksella' is followed by a search input field containing the text 'Hae'. The main content area shows a list of completed tests, with the first entry for 'hel.fi/fi'. This entry includes a timestamp '08.05.2023 15.03', a desktop icon, and statistics: '1' document icon, '14' checkmarks, '6' question marks, and '1' cross. To the right of these statistics are three buttons: 'JSON', 'JSON (PDF)', and 'PDF'.

Testin nimi	Aika	Laajuus	Oikeat vastaukset	Väärät vastaukset	Kysymykset	Vaiheet	Export
hel.fi/fi	08.05.2023 15.03	1	14	6	1		JSON, JSON (PDF), PDF

Kuva 68. – Näkymä suoritetusta saavutettavuustestistä



Tarkastusraportti

Tulostettu: 08.05.2023

Saavutettavuustarkastuksen raportti, yksinkertaistettu valvonta

Tarkastettu palvelu: <http://hel.fi>, 08.05.2023 15:50

Testattu selainikkunan koolla: 1920px X 1080px

Tarkastus suoritettiin Salvia-testausohjelmistolla (<https://github.com/accessibility-fi/salvia>). Ohjelmisto perustuu avoimen lähdekoodin QualWeb core -testausohjelmistoon (<https://github.com/qualweb/core>). Tarkastuksessa testattiin sivuston saavutettavuutta käyttäen WCAG 2.1:n A- ja AA-tason kriteereihin liitettyjä automaattisia ACT-testaussääntöjä (<https://github.com/qualweb/act-rules>).

Testitulosten yhteenveto

Tarkastetut sivut

Tarkastuksessa testattiin seuraavat verkkosivut (1 kpl):

- <http://hel.fi/>

Suoritettut testit

Saavutettavuuspuutteita löytyi seuraavilla testeillä (1 kpl):

- QW-ACT-R18, `Id`-attribuutin arvo on yksilöllinen (<https://act-rules.github.io/rules/3ea0c8>)

Saavutettavuutta tai sen puutetta ei voitu automatisoidusti todentaa seuraavilla testeillä (7 kpl):

- QW-ACT-R63, Dokumentilla on maamerkki ei-toistuvalla sisällölle (<https://act-rules.github.io/rules/b40fd1>)
- QW-ACT-R64, Dokumentilla on otsikko ei-toistuvalla sisällölle (<https://act-rules.github.io/rules/047fe0>)
- QW-ACT-R73, Toistuvan sisällön lohko on kutistettava (<https://act-rules.github.io/rules/3e12e1>)
- QW-ACT-R74, Dokumentilla on mekanismi kohdistuksen siirtämiseksi ei-toistuvaan sisältöön (<https://act-rules.github.io/rules/ye5d6e>)
- QW-ACT-R62, Järjestyksessä kohdistettaviin sisältöihin kuuluvalla elementillä on näkyvä kohdistus (<https://act-rules.github.io/rules/oj04fd>)
- QW-ACT-R41, Virheilmoitus selittää virheen lomakekentän syötteessä. (<https://act-rules.github.io/rules/36b590>)
- QW-ACT-R75, Ohita toistuvan sisällön lohkot (<https://act-rules.github.io/rules/cf77f2>)

Saavutettavuuspuutteita ei löytynyt seuraavilla testeillä (15 kpl):

- QW-ACT-R1, HTML-sivulla on sivuotsikko (<https://act-rules.github.io/rules/2779a5>)
- QW-ACT-R2, `Html` -elementissä on `lang` -attribuutti (<https://act-rules.github.io/rules/b5c3f8>)
- QW-ACT-R5, `Html` -elementin `lang` -attribuutin kelpoisuus (<https://act-rules.github.io/rules/bf051a>)
- QW-ACT-R11, Painikkeella on saavutettava nimi (<https://act-rules.github.io/rules/97a4e1>)
- QW-ACT-R12, Linkillä on saavutettava nimi (<https://act-rules.github.io/rules/c487ae>)
- QW-ACT-R13, Elementillä, jolle on määritelty `aria-hidden` -attribuutti, ei ole kohdistettavaa sisältöä (<https://act-rules.github.io/rules/6cfa84>)
- QW-ACT-R14, `Meta viewport` ei estä zoomausta (<https://act-rules.github.io/rules/b4f0c3>)
- QW-ACT-R22, `Body` -elementin alielementin `lang` -attribuutin arvo on sallittu. (<https://act-rules.github.io/rules/de46e4>)
- QW-ACT-R28, Elementillä, jolle on määritelty `role` -attribuutti, on annettu myös vaaditut tilat ja ominaisuudet (<https://act-rules.github.io/rules/4e8ab6>)
- QW-ACT-R16, Lomake-elementillä on saavutettava nimi (<https://act-rules.github.io/rules/e086e5>)
- QW-ACT-R20, `Role` -attribuutilla on sallittu arvo (<https://act-rules.github.io/rules/674b10>)
- QW-ACT-R17, Kuvalla on saavutettava nimi (<https://act-rules.github.io/rules/23a2a8>)
- QW-ACT-R35, Otsikolla on saavutettava nimi (<https://act-rules.github.io/rules/ffd0e9>)
- QW-ACT-R37, teksti täyttää kontrastin vähimmäisvaatimukset (<https://act-rules.github.io/rules/afw4f7>)
- QW-ACT-R65, Elementeillä, jotka sisältävät vain visualisoivia alielementtejä, ei ole kohdistettavia sisältöjä (<https://act-rules.github.io/rules/307n5z>)

Testejä, jotka eivät soveltuneet sivuilta löytyneisiin sisältöihin oli (41 kpl):

- QW-ACT-R4, `Meta`-elementin uudelleenlatauksessa ei viivettä (<https://act-rules.github.io/rules/bc659a>)
- QW-ACT-R43, Vieritettävä elementti on käytettävissä näppäimistöllä (<https://act-rules.github.io/rules/0ssw9k>)
- QW-ACT-R3, HTML-elementin `lang` ja `xml:lang` vastaavat toisiaan (<https://act-rules.github.io/rules/5b7ae0>)
- QW-ACT-R6, Kuvaa esittävällä painikkeella on saavutettava nimi (<https://act-rules.github.io/rules/59796f>)
- QW-ACT-R21, `Svg`-elementillä, jolla on eksplisiittinen rooli, on saavutettava nimi (<https://act-rules.github.io/rules/7d6734>)
- QW-ACT-R38, ARIA:n vaatimat omistetut elementit (<https://act-rules.github.io/rules/bc4a75>)
- QW-ACT-R10, `Iframe`-elementeillä, joilla on samat saavutettavat nimet, on myös sama tarkoitus (<https://act-rules.github.io/rules/4b1c6c>)
- QW-ACT-R44, Samassa yhteydessä sijaitsevilla linkeillä, joilla on sama saavutettava nimi, on sama käyttötarkoitus. (<https://act-rules.github.io/rules/fd3a94>)
- QW-ACT-R33, ARIA:n vaatima rooli kontekstissa (<https://act-rules.github.io/rules/ff89c9>)
- QW-ACT-R19, `Iframe`-elementillä on saavutettava nimi (<https://act-rules.github.io/rules/cae760>)
- QW-ACT-R58, Äänitallenteelle on transkripti (<https://act-rules.github.io/rules/2eb176>)
- QW-ACT-R59, Äänitallenne on tekstin mediavastine (<https://act-rules.github.io/rules/afb423>)
- QW-ACT-R30, Elementin näkyvä nimi on osa sen saavutettavaa nimeä (<https://act-rules.github.io/rules/2ee8b8>)
- QW-ACT-R51, Videon näkyvä sisältö on tekstin mediavastine (<https://act-rules.github.io/rules/fd26cf>)
- QW-ACT-R52, Ääntä sisältämättömän videon videosisällölle on kuvailutulkauksen tekstitysraila (<https://act-rules.github.io/rules/ac7dc6>)

- QW-ACT-R53, Vain kuvaa sisältävälle videolle on tarjolla transkripti (<https://act-rules.github.io/rules/ee13b5>)
- QW-ACT-R54, Vain kuvaa sisältävälle videotallenteelle on tarjolla vaihtoehto äänitallenteena (<https://act-rules.github.io/rules/d7ba54>)
- QW-ACT-R55, Videon näkyvälle sisällölle on tarjolla kuvailutulkkaus (<https://act-rules.github.io/rules/1ea59c>)
- QW-ACT-R56, Video on tekstin mediavastine (<https://act-rules.github.io/rules/ab4d13>)
- QW-ACT-R57, Videon näkyvälle sisällölle on kuvailutulkkauksen tekstitysraita (<https://act-rules.github.io/rules/f196ce>)
- QW-ACT-R60, Videon äänisisällölle on tekstitys (<https://act-rules.github.io/rules/f51b46>)
- QW-ACT-R61, Video elementin ääni- ja kuvasisällölle on olemassa transkripti (<https://act-rules.github.io/rules/1a02b0>)
- QW-ACT-R49, Ääni- tai videosisältö, joka käynnistyy automaattisesti, ei sisällä ääntä, joka kestää yli kolme (3) sekuntia (<https://act-rules.github.io/rules/aaa1bf>)
- QW-ACT-R50, Ääni- tai videosisällöllä, joka käynnistyy automaattisesti, on ohjausvälineet. (<https://act-rules.github.io/rules/4c31df>)
- QW-ACT-R24, `Autocomplete`-attribuutilla on sallittu arvo (<https://act-rules.github.io/rules/73f2c2>)
- QW-ACT-R36, Solun `headers`-attribuutti viittaa soluun samassa taulukkoelementissä. (<https://act-rules.github.io/rules/a25f45>)
- QW-ACT-R39, Kaikki taulukon otsikkosolut kohdistuvat sisältösoluihin (<https://act-rules.github.io/rules/d0f69e>)
- QW-ACT-R42, `Object`-elementillä on saavutettava nimi, joka ei ole tyhjä. (<https://act-rules.github.io/rules/8fc3b6>)
- QW-ACT-R67, Tyylimäärittelyissä kirjainten välistystä ei ole määritetty `!important`-säännöllä (<https://act-rules.github.io/rules/24afc2>)
- QW-ACT-R68, Tyylimäärittelyissä rivin korkeutta ei ole määritetty `!important`-säännöllä (<https://act-rules.github.io/rules/78fd32>)
- QW-ACT-R69, Tyylimäärittelyissä sanojen välistystä ei ole määritetty `!important`-säännöllä (<https://act-rules.github.io/rules/9e45ec>)

- QW-ACT-R66, Valikon osalla on ei-tyhjä saavutettava nimi (<https://act-rules.github.io/rules/m6b1q3>)
- QW-ACT-R7, Sivun asentoa ei ole rajoitettu CSS:n `transform`-säännöllä (<https://act-rules.github.io/rules/b33eff>)
- QW-ACT-R70, `iframe`, jolla on negatiivinen `tabindex`-attribuutti, ei sisällä interaktiivisia elementtejä (<https://act-rules.github.io/rules/akn7bn>)
- QW-ACT-R15, Ääni- tai videosisältö ei sisällä ääntä, joka käynnistyy automaattisesti (<https://act-rules.github.io/rules/80f0bf>)
- QW-ACT-R23, Videon näkyvälle sisällölle on saavutettava vaihtoehto (<https://act-rules.github.io/rules/c5a4ea>)
- QW-ACT-R26, Videon äänisisällölle on saavutettava vaihtoehto (<https://act-rules.github.io/rules/eac66b>)
- QW-ACT-R29, Äänielementin sisältö on saatavissa myös tekstinä (<https://act-rules.github.io/rules/e7aa44>)
- QW-ACT-R31, Äänettömän videon näkyvälle sisällölle on saavutettava vaihtoehto (<https://act-rules.github.io/rules/c3232f>)
- QW-ACT-R32, Videon näkyvälle sisällölle on saavutettava vaihtoehto (<https://act-rules.github.io/rules/1ec09b>)
- QW-ACT-R40, CSS:n `overflow`-määrittäminen ei leikkaa zoomattua tekstiä. (<https://act-rules.github.io/rules/59br37>)

Havaitut saavutettavuuspuutteet

QW-ACT-R18 – `id`-attribuutin arvo on yksilöllinen

Tämä sääntö tarkastaa, että jokaisen samalla sivulla esiintyvällä `id`-attribuutilla on yksilöllinen arvo.

- <https://act-rules.github.io/rules/3ea0c8>
- WCAG 2.1, Taso A: 4.1.1 (<https://www.w3.org/Translations/WCAG21-fi/#parsing>)

SIVU: <http://hel.fi/>

ELEMENTTI:

```
html > body:nth-child(2) > div:nth-child(5) > div:nth-child(4) > header:nth-child(1) >
div:nth-child(1) > nav:nth-child(1) > ul:nth-child(1) > li:nth-child(6) > span:nth-
child(1) > a:nth-child(1) > span:nth-child(3)
```

```
<span class="is-hidden" id="hdbt-link-669337072">(Linkki johtaa ulkoiseen palveluun)</
span>
```

ELEMENTTI:

```
html > body:nth-child(2) > div:nth-child(5) > div:nth-child(4) > header:nth-child(1) >
div:nth-child(2) > nav:nth-child(7) > div:nth-child(2) > div:nth-child(1) > div:nth-
child(1) > div:nth-child(3) > nav:nth-child(1) > ul:nth-child(1) > li:nth-child(6) >
span:nth-child(1) > a:nth-child(1) > span:nth-child(3)
```

```
<span class="is-hidden" id="hdbt-link-669337072">(Linkki johtaa ulkoiseen palveluun)</
span>
```

TULOS: Usealla elementillä on sama `id`-attribuutin arvo.