

QAutoEQUAL

Web-pohjaisen käyttöliittymän kehitys saavutettavuuden parantamiseksi

Sasa Mora Roca
Opinnäytetyö (AMK)
Syksy 2024
Tietojenkäsittelyn tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn tutkinto-ohjelma
Tradenomi

Tekijä: Sasa Mora Roca

Opinnäytetyön otsikko: QAutoEQUAL: Web-pohjaisen käyttöliittymän kehitys saavutettavuuden parantamiseksi

Työn ohjaaja: Minna Kamula

Syksy 2024

Sivumäärä: 36

Tämä toiminnallinen opinnäytetyö keskittyy QAutoEQUAL-työkalun käyttöliittymän kehittämiseen saavutettavuuden automaattista arviointia varten. Työkalun tavoitteena on helpottaa verkkosivustojen saavutettavuuden tarkastusta WCAG-standardien mukaisesti tarjoamalla käyttäjäystävällinen sivusto, joka mahdollistaa skannausprosessin seuraamisen, tulosten tarkastelun ja raporttien lataamisen käyttäjän valitsemasta verkkosivustosta.

Työn lähtökohtana olivat saavutettavuuden merkitys digitaalisten palvelujen tasa-arvon edistäjänä ja lainsäädännölliset vaatimukset. Suunnitteluprosessi noudatti käyttäjäkeskeisiä periaatteita ja iteratiivista kehitysmallia, jossa käyttöliittymän toiminnallisuuksia ja käytettävyyttä parannettiin jatkuvasti palautteen pohjalta. Teknologisina ratkaisuinä hyödynnettiin WordPress-alustaa ja HTMX-tekniologiaa, jotka mahdollistivat dynaamisen ja modulaarisen käyttöliittymän toteuttamisen.

Keskeisiä toteutuksia olivat syötteiden validointi, skannausprosessin tilan seuranta ja tulosten esittäminen HTMX-endpointien avulla. Käyttöliittymässä toteutettiin myös kaupallisen laajennettavuuden mahdollistavia ominaisuuksia, kuten WooCommerce-integraatio käyttäjähallintaan. Näiden avulla palveluun voidaan jatkossa liittää lisenssipohjaisia kaupallisia ominaisuuksia.

Työ osoittaa, että hyvin suunniteltu sekä helposti käytettävä käyttöliittymä voi merkittävästi tukea saavutettavuuden parantamista ja tarjota selkeää lisäarvoa käyttäjille. Projektin aikana kertynyt kokemus sekä saavutettavuusteknologioiden että kehitysmenetelmien soveltamisesta tarjoaa vahvan pohjan jatkokehitykselle ja tuleville haasteille ICT-alalla.

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Business Information Systems
Option of Bachelor of Business Administration

Author: Sasa Mora Roca

Title of thesis: QAutoEQUAL: Development of a Web-Based Interface to Improve Accessibility

Supervisor: Minna Kamula

Fall 2024

Pages: 36

This functional thesis focuses on developing the user interface for the QAutoEQUAL tool, designed for automated accessibility evaluation. The tool aims to simplify the inspection of website accessibility according to WCAG standards by providing a user-friendly platform that allows users to monitor the scanning process, review results, and download reports from their chosen website.

The foundation of this work lies in the significance of accessibility as a promoter of equality in digital services and the related legal requirements. The design process followed user-centered principles and an iterative development model, where the functionalities and usability of the interface were continuously improved based on feedback. Technological solutions included leveraging the WordPress platform and HTMX technology, which enabled the creation of a dynamic and modular user interface.

Key implementations included input validation, monitoring the scanning process, and presenting results through HTMX endpoints. The interface also incorporated features enabling commercial scalability, such as WooCommerce integration for user management. These features provide the foundation for future integration of license-based commercial functionalities.

This work demonstrates that a well-designed and user-friendly interface can significantly support accessibility improvement and deliver clear value to users. The experience gained during the project, as well as the application of accessibility technologies and development methods, provides a strong foundation for further development and future challenges in the ICT field.

SISÄLLYS

1	JOHDANTO	6
1.1	Tausta ja merkitys	6
1.2	Työn tavoitteet ja rajaus	6
1.3	Suunnitteluprosessi	6
2	SAAVUTETTAVUUDEN PERUSTEET	7
2.1	Saavutettavuuden määritelmä ja merkitys	7
2.2	Lainsäädännölliset vaatimukset ja standardit	8
2.3	Saavutettavuuden merkitys käyttäjäkokemukselle	11
3	NYKYISET SAAVUTETTAVUUSTEKNOLOGIAT	12
3.1	Käsin kirjoitettu vs. automaattinen testaus	12
3.2	Nykyisten teknologioiden rajoitukset	14
4	SUUNNITTELUPERIAATTEET JA KEHITYSMETODOLOGIA	15
4.1	Käyttäjakeskeinen suunnittelu (User-Center Design)	15
4.2	Iteratiivinen kehitysprosessi	15
5	SAAVUTETTAVUUSTEKNOLOGIAN KEHITYS	17
5.1	Vaatimusmäärittely	17
5.2	Arkkitehtuurin suunnittelu ja teknologiavalinnat	17
5.3	Prototyypin kehittäminen	18
5.4	Keskeiset koodiratkaisut ja HTMX-endpointit	21
5.4.1	Syötteiden käsittely ja validointi käyttöliittymässä	22
5.4.2	Skannauksen käynnistäminen: robot-start HTMX-endpoint..	24
5.4.3	Skannauksen tilaseuranta: robot-status HTMX-endpoint	27
5.4.4	Skannausten tulokset: robot-done HTMX-endpoint	29
5.5	Testaus ja iterointi	30
6	TEKNOLOGIAN SOVELTAMINEN KÄYTÄNNÖSSÄ	31
6.1	Kohdevalinta	31
6.2	Testausprosessi	31
6.3	Tulokset ja analyysi	32
6.4	Käytännön hyödyt ja haasteet	33
7	POHDINTA	34
	LÄHTEET	35

SANASTO

HTMX	JavaScript-kirjasto, joka lisää HTML-elementtien interaktiivisuutta ja mahdollistaa dynaamiset toiminnot, kuten http-pyyntöt ja sivun osittaisen päivityksen, yksinkertaistaen käyttöliittymän kehitystä.
Inklusiivinen	Huomioi erilaiset käyttäjät ja varmistaa yhdenvertaisen osallistumisen mahdollisuuden kaikille.
Kognitiivinen	Käytettävyyden osa-alue, joka liittyy ihmisen ajattelun ja tiedonkäsittelyn tarpeisiin. Saavutettavuudessa se tarkoittaa selkeää, ymmärrettävää ja helposti navigoitavaa sisältöä, joka tukee erilaisten käyttäjien kognitiivisia kykyjä.
MVP	<i>Minimum Viable Product</i> , tuotteen tai palvelun vähimmäisversio, joka sisältää olennaiset toiminnot käyttäjäpalautteen keräämiseksi ja testauksen mahdollistamiseksi.
Pollaus	Tilan tai tiedon säännöllinen tarkistaminen palvelimelta käyttöliittymässä, usein käyttäen määritettyä aikaväliä tietojen päivittämiseksi reaaliaikaisesti.
QAutoEQUAL	Työkalu, joka on suunniteltu arvioimaan ja parantamaan verkkosivustojen saavutettavuutta WCAG-standardien mukaisesti.
WebCrawling	Verkkosivustojen automaattinen selailu ja tietojen kerääminen niiden saavutettavuuden, rakenteen tai muun sisällön analysoimiseksi.
WCAG	<i>Web Content Accessibility Guidelines</i> , kansainväliset ohjeet verkkosisällön saavutettavuudelle, jotka sisältävät eri tasoisia saavutettavuusvaatimuksia.

1 JOHDANTO

1.1 Tausta ja merkitys

Tämä toiminnallinen opinnäytetyö keskittyy web-pohjaisen robottiohjauksen käyttöliittymän kehitykseen. Työ on osa laajempaa pyrkimystä parantaa digitaalisten palveluiden saavutettavuutta ja käytettävyyttä. Projektin kohteena on QAutoEQUAL, innovatiivinen työkalu, joka on suunniteltu arvioimaan ja parantamaan web-sivustojen saavutettavuutta WCAG-standardien mukaisesti. Projektin toimeksiantaja on Q-Factory, Suomen suurin riippumaton ohjelmistotestauksen ja laadunvarmistuksen toimittaja, joka edistää digitaalisten palveluiden toimintavarmuutta. (Q-Factory 2024.)

1.2 Työn tavoitteet ja rajaus

Opinnäytetyön tavoitteena on suunnitella käyttäjäystävällinen ja tehokas web-käyttöliittymä, jonka avulla erilaiset käyttäjät voivat helposti saada yksityiskohtaisia tietoja ja parannusehdotuksia sivustojensa saavutettavuuden osalta. Työn metodologia perustuu iteratiiviseen kehitysprosessiin, jossa yhdistyvät käyttäjäkeskeinen suunnittelu ja pilvipohjaisten teknologioiden hyödyntäminen.

1.3 Suunnitteluprosessi

Projektin kehitysprosessi on iteratiivinen, ja se sisältää useita prototyypin suunnittelu- ja kehitysvaiheita. Tässä vaiheessa keskitytään suunnitelmien ja prototyyppien luomiseen, jotka myöhemmin testataan ja kehitetään edelleen käyttäjäpalautteen avulla. Käyttäjäpalautteen kerääminen ja tarkempi analyysi tapahtuvat myöhemmässä vaiheessa, kun prototyyppi on valmis ja BETA-tason sivusto on käytettävissä.

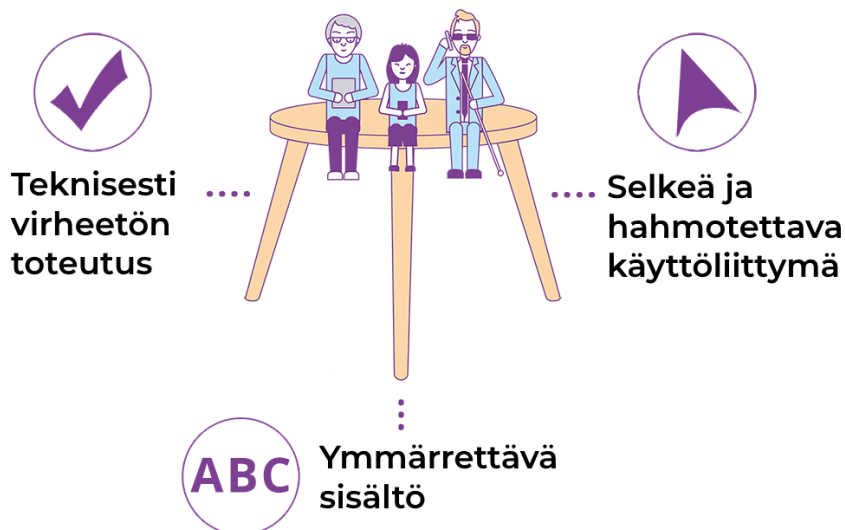
2 SAAVUTETTAVUUDEN PERUSTEET

2.1 Saavutettavuuden määritelmä ja merkitys

Saavutettavuus tarkoittaa sitä, että mahdollisimman moni erilainen ihminen voi käyttää verkkosivuja mahdollisimman helposti. Saavutettavuus on ihmisten erilaisuuden ja moninaisuuden huomiointia verkkosivujen suunnittelussa ja toteutuksessa.

Saavutettavuus liittyy usein digitaalisiin palveluihin, kuten verkkosivuihin. Se tarkoittaa sitä, että digitaaliset palvelut ovat esteettömiä ja helposti käytettävissä kaikille ihmisille heidän erilaisista tarpeistaan huolimatta. Tämä lisää yhdenvertaisuutta digitaalisessa yhteiskunnassa.

Kun saavutettavuus toteutetaan hyvin, eri taustaiset ja kykyiset ihmiset pystyvät käyttämään digitaalisia palveluja itsenäisesti. Tämä parantaa erityisesti rajoitteisten henkilöiden itsenäisyyttä ja osallistumismahdollisuuksia yhteiskunnassa, mahdollistaen heille täysipainoisen osallistumisen digitaaliseen maailmaan. (Saavutettavuusvaatimukset.fi 2024.)



Kuva 1. Saavutettavuuden perusteet: teknisesti virheetön toteutus, ymmärrettävä sisältö ja selkeä käyttöliittymä. (Saavutettavuusvaatimukset.fi 2024.)

2.2 Lainsäädännölliset vaatimukset ja standardit

”Laki digitaalisten palvelujen tarjoamisesta, eli digipalvelulaki (306/2019), asettaa saavutettavuusvaatimuksia julkiselle sektorille sekä osalle yksityisen ja kolmannen sektorin organisaatioista. Tämä laki velvoittaa näitä toimijoita varmistamaan, että heidän digitaaliset palvelunsa, kuten verkkosivut, ovat kaikkien saavutettavissa”. (Laki digitaalisten palvelujen tarjoamisesta 306/2019.)

Kolme keskeistä vaatimusta

Digipalvelulaki sisältää kolme päävaatimusta, jotka kaikkien lain piiriin kuuluvien digitaalisten palvelujen tulee täyttää:

1. Tekniset saavutettavuusvaatimukset: Digitaalisten palvelujen ja niiden sisältöjen tulee täyttää eurooppalaisen standardin EN 301 549 määrittelemät tekniset vaatimukset. Tämä standardi viittaa WCAG-ohjeistukseen ja kattaa laajemmin saavutettavuusvaatimuksia. (Saavutettavuusvaatimukset.fi, Digipalvelulain vaatimukset 2024.)
2. Saavutettavuusseloste: Digitaalisen palvelun tulee sisältää saavutettavuusseloste, joka kertoo käyttäjille palvelun saavutettavuuden tilasta. Selosteessa tulee arvioida palvelun saavutettavuus ja tuoda esiin mahdolliset puutteet. Kuvassa 2 on esimerkki QAutoEQUAL-työkalun tuottamasta saavutettavuusselosteesta, joka havainnollistaa, kuinka palvelun saavutettavuutta voidaan arvioida ja esittää käyttäjälle. (Saavutettavuusvaatimukset.fi, Digipalvelulain vaatimukset 2024.)
3. Palaute: Käyttäjille tulee tarjota mahdollisuus antaa palautetta saavutettavuudesta. Palvelussa on oltava kanava palautteelle, ja siihen tulee vastata 14 päivän kuluessa. Joillekin palveluille on asetettu myös lisävaatimuksia ja -velvollisuuksia, joista löytyy lisätietoja digipalvelulain muutoksia käsittelevältä sivulta. (Saavutettavuusvaatimukset.fi, Digipalvelulain vaatimukset 2024.)

Saavutettavuusseloste:

Tämä saavutettavuusseloste koskee THIS IS TRIAL TEST SCAN -palvelua ([THIS IS TRIAL TEST SCAN](#)).

Lain mukaan julkisten verkkopalvelujen on oltava saavutettavia, eli kaikilla tulee olla tasavertaiset mahdollisuudet käyttää verkkopalveluja (Laki 306/2019).

Tässä selosteessa kerrotaan, mitä puutteita sivuston [THIS IS TRIAL TEST SCAN](#) saavutettavuudessa on, ja miten voit antaa meille palautetta palvelun saavutettavuudesta.

Digipalvelun saavutettavuuden tila:

- Palvelu täyttää saavutettavuusvaatimukset osittain

Tämän saavutettavuusselosteen tiedot perustuvat jatkuvaan arviointiin käyttäen Qautoequal-saavutettavuustyökalua. Testaustyökalu auttaa tunnistamaan verkkosivuston mahdolliset saavutettavuuspuutteet, mutta sen tulokset voivat vaihdella käyttöympäristön ja testausparametrien mukaan. Pyrimme aktiivisesti seuraamaan ja päivittämään saavutettavuustietojamme varmistaaksemme, että palvelumme on mahdollisimman saavutettava kaikille käyttäjille.

Verkkopalvelussa olevat keskeiset puutteet on kuvattu jäljempänä tässä selosteessa.

Tällä verkkosivustolla on havaittu seuraavat saavutettavuuspuutteet 09/09/2024

- Sivulla on painikkeita, joilla ei ole saavutettavaa nimeä (WCAG 4.1.2)
- Sivulla on linkkejä, joilla ei ole saavutettavaa nimeä (WCAG 2.4.4, 2.4.9, 4.1.2)
- Komponentin toteutuksessa on puutteita: rakenteet eivät ole roolimäärittelyn mukaisia (WCAG 1.3.1)
- Uppoiteella ei ole saavutettavaa nimeä (WCAG 4.1.2)
- Elementin saavutettava nimi ei vastaa sen näkyvää tekstiä (WCAG 2.5.3)
- Tekstin ja taustan välinen kontrasti ei ole riittävä (WCAG 1.4.3)
- Elementti, jonka sisältö on koristeellista, sisältää kohdistettavia elementtejä (WCAG 4.1.2)

Huomasitko muita puutteita saavutettavuudessa?

Pyrimme jatkuvasti parantamaan verkkopalvelun saavutettavuutta. Jos huomaat sivustolla saavutettavuusongelmia, voit lähettää palautetta meille palautetoiminnon kautta. Otamme mielellämme vastaan myös muita huomioita ja parannusehdotuksia sivuston saavutettavuudesta. Vastamme palautteeseen 14 päivän sisällä.

Anna saavutettavuuspalautetta :

[THIS IS TRIAL TEST SCAN / THIS IS TRIAL TEST SCAN](#)

Saavutettavuuden valvonta Etelä-Suomen aluehallintovirasto valvoo saavutettavuusvaatimusten toteutumista. Jos et ole tyytyväinen meiltä saamaasi vastaukseen tai et saa vastausta lainkaan kahden viikon aikana, voit antaa palautteen Etelä-Suomen aluehallintovirastoon. Etelä-Suomen aluehallintoviraston sivulla kerrotaan tarkasti, miten valituksen voi tehdä ja miten asia käsitellään.

Etelä-Suomen aluehallintovirasto
Saavutettavuuden valvonnan yksikkö
[Saavutettavuusvaatimukset.fi](https://saavutettavuusvaatimukset.fi)
saavutettavuus@avi.fi
p. 0295 016 000 (vaihde)

Lain taustalla vaikuttavat tekijät

"Digipalvelulaki astui voimaan Suomessa 1.4.2019 ja sen 3 a lukua koskevat muutokset 1.2.2023. Lain taustalla ovat Euroopan unionin saavutettavuus- ja esteettömyysdirektiivit sekä YK yleissopimus rajoitteisten henkilöiden oikeuksista, joka korostaa heidän osallisuuttaan ja mahdollisuuttaan itsenäiseen asiointiin myös digitaalisessa ympäristössä. Saavutettavuus on yksi sopimuksen keskeisistä periaatteista". (Laki digitaalisten palvelujen tarjoamisesta 306/2019.)

Saavutettavuuden laajempi merkitys

Lain tavoitteena on parantaa jokaisen mahdollisuuksia käyttää yhdenvertaisesti digitaalisia palveluja. Saavutettavuus tulee ymmärtää keskeisenä toiminnan periaatteena, joka tulisi ottaa huomioon kaikessa verkkosisältöjen ja -palvelujen suunnittelussa, toteutuksessa ja ylläpidossa. Etelä-Suomen aluehallintoviraston saavutettavuuden valvonnan yksikkö valvoo digipalvelulain noudattamista. (Saavutettavuuden lait ja standardit – Saavutettavuusvaatimukset.fi 2024.)

Saavutettavuus digipalvelulain vaatimuksia laajempänä kokonaisuutena

Lain saavutettavuusvaatimusten noudattaminen ei takaa, että digipalvelu olisi kaikille täysin saavutettava tai helppokäyttöinen. Noudattamalla digipalvelulain saavutettavuusvaatimuksia voidaan kuitenkin varmistaa, että verkkosivuja voidaan käyttää erilaisilla laitteilla ja avustavilla teknologioilla. Saavutettavuuden huomioon ottaminen edellyttää myös selkeää ja ymmärrettävää kieltä sekä palvelun helppokäyttöisyyttä. (Saavutettavuusvaatimukset.fi 2024.)

Eurooppalainen standardi EN 301 549

Digipalvelulain tekniset saavutettavuusvaatimukset on määritelty yhdenmukaisesti eurooppalaisessa EN 301 549 -standardissa. Standardi viittaa WCAG-ohjeistukseen, mutta se kattaa saavutettavuusvaatimukset laajemmin. Verkkosivustojen tulee noudattaa WCAG-ohjeistuksen A- ja AA-tason kriteereitä. (European Union 2021; W3C 2018; Saavutettavuusvaatimukset.fi 2024.)

2.3 Saavutettavuuden merkitys käyttäjäkokemukselle

Saavutettavuus on keskeinen tekijä käyttäjäkokemuksen parantamisessa, sillä se vaikuttaa suoraan siihen, kuinka käyttäjät vuorovaikuttavat verkkosivujen tai digitaalisten palvelujen kanssa. Saavutettavuus ei koske vain käyttäjiä, joilla on erilaisia vammoja tai rajoitteita, vaan myös kaikkia käyttäjiä, jotka saattavat kohdata tilapäisiä esteitä, kuten heikentynyttä verkkoyhteyttä, kirkasta auringonvaloa tai laitteistojen rajoitteita. Tämä tekee saavutettavuudesta olennaisen osan koko käyttäjäkokemuksen parantamista. (Saavutettavuusvaatimukset.fi 2024.)

Hyvin toteutetut verkkosivustot saavutettavuuden näkökannalta tarjoavat käyttäjille selkeän, loogisen ja miellyttävän käyttökokemuksen, mikä vähentää käyttäjien turhautumista ja virheiden syntymistä. Saavutettavuudella varmistetaan, että eri käyttäjäryhmät voivat käyttää sivustoa helposti. Tämä saavutetaan esimerkiksi tarjoamalla kuville tekstivastineet, selkeästi jäsenneily navigaatio, tekstitykset videoille ja lomakkeet, joita on helppo täyttää. Näiden avulla käyttäjät voivat suoriutua haluamistaan toiminnoista helpommin ja ilman esteitä. (Saavutettavuusvaatimukset.fi 2024.)

Lisäksi saavutettavuus edistää käyttäjien sitoutumista ja asiakastyytyväisyyttä, koska se osoittaa organisaation arvojen, kuten yhdenvertaisuuden ja inklusiivisuuden, huomioimista. Inklusiivisten verkkosivujen ansiosta käyttäjät kokevat, että heidän tarpeensa otetaan huomioon, mikä luo positiivisen kuvan organisaatiosta ja parantaa asiakasuskollisuutta. Saavutettavat verkkopalvelut houkuttelevat myös laajempaa käyttäjäkuntaa, mikä voi lisätä liikennettä ja laajentaa palvelun yleisöä. (Saavutettavuusvaatimukset.fi 2024.)

Selkeät otsikkorakenteet, kuvien alt-tekstit ja sisällön looginen jäsentely auttavat hakukoneita ymmärtämään verkkosivujen sisältöä paremmin. Tämän ansiosta sivuston näkyvyys hakutuloksissa voi parantua. Parempi hakukonesijoitus voi puolestaan ohjata oikeita käyttäjäryhmiä sivustolle, mikä lisää kävijöiden määrää ja auttaa tavoittamaan halutut käyttäjät tehokkaammin. (Saavutettavuusvaatimukset.fi 2024.)

3 NYKYISET SAAVUTETTAVUUSTEKNOLOGIAT

3.1 Käsin kirjoitettu vs. automaattinen testaus

Saavutettavuuden testaamisessa käytetään sekä käsin kirjoitettua testausta että automaattisia työkaluja. Molemmilla menetelmillä on omat vahvuutensa ja heikkoutensa, ja ne täydentävät toisiaan saavutettavuuden arvioinnissa. (Itpedia 2017.)

Käsin kirjoitettu testaus

Käsin kirjoitettu, eli manuaalinen testaus, suoritetaan yleensä ihmisen toimesta, ja se perustuu tiettyihin ennalta asetettuihin kriteereihin. Manuaalisessa testauksessa testaaja arvioi esimerkiksi verkkosivun navigoitavuutta ilman hiiren käyttöä, ruudunlukijan toimivuutta sekä tekstien selkeyttä ja ymmärrettävyyttä. Tämä testausmuoto edellyttää, että testaaja tuntee saavutettavuusvaatimukset hyvin ja pystyy havaitsemaan hienovaraisia ongelmia, joita automaattiset työkalut eivät välttämättä tunnista. (Itpedia 2017.)

Manuaalisella testauksella voidaan arvioida erityisesti WCAG 2.1 -ohjeistuksen A- ja AA-tason kriteerien täyttymistä, kuten sisällön ymmärrettävyyttä, käyttöliittymän loogisuutta ja kognitiivista saavutettavuutta. Tällainen arviointi on erityisen hyödyllinen, kun halutaan varmistaa, että sivuston käyttäjäkokemus on sujuva ja intuitiivinen kaikille käyttäjille. (W3C 2018.)

Automaattinen testaus

Automaattinen testaus perustuu ohjelmistotyökaluihin, kuten QAutoEQUAL, jotka analysoivat verkkosivujen koodin ja sisällön tarkistaakseen niiden teknisen saavutettavuuden WCAG-ohjeistuksen mukaisesti. Työkalut voivat nopeasti skannata suuren määrän verkkosivuja ja löytää yleisiä saavutettavuusvirheitä, kuten puuttuvia alt-tekstejä, virheellisesti käytettyjä otsikkotasoja sekä muita HTML-rakenteen ongelmia. Automaattinen testaus on erityisen tehokas, kun

testausprosessit ovat laajoja tai sisältävät paljon toistuvia tehtäviä, jotka tekisivät manuaalisesta testauksesta työlästä.

Kuvassa 3 on esimerkki QAutoEQUAL-työkalulla tehdystä saavutettavuustestauksen loppuraportista, jossa on havaittu useita WCAG-ohjeistuksen mukaisia virheitä. Tällaiset raportit tarjoavat yksityiskohtaisen analyysin teknisistä saavutettavuusongelmista ja auttavat kehittäjiä korjaamaan puutteet.

Versio: 3.6.5
QAutomate:
<https://qautomate.fi>
Powered by:
Qualweb@0.7.72

QAutoEQUAL
Saavutettavuustestauksen tulokset

Testitapaus: Crawl Through Site
Ajettu: 2024-09-09 09:34:39
Saavutettavuusseloste

Reportin tyyppi: Sivun mukaan WCAG-koodi:

Vaativuustaso: Kaikki Tasot: A 7 AA 1 Vakavuus: Kaikki Vakavuus: 9 218 6
Virheellisiä elementtejä: 233

53 #1 233 virheellistä elementtiä A 7 AA 1

Mistä kutsuttu: Crawl Through Site
Osoitteelle:

> Kuvakaappaus (virheelliset elementit merkitty punaisella)

Rikotut säännöt:

- > **QW-ACT-R11: Painikkeella tulee olla saavutettava nimi** | 1 virheellinen elementti | A
- > **QW-ACT-R12: Linkillä tulee olla saavutettava nimi** | 4 virheellistä elementtiä | A
- > **QW-ACT-R38: Elementillä tulee olla ARIA:n vaatimat alielementit** | 50 virheellistä elementtiä | A
- > **QW-ACT-R19: `iframe`-elementillä tulee olla saavutettava nimi** | 1 virheellinen elementti | A
- > **QW-ACT-R30: Elementissä näkyvän tekstin tulee olla osa sen saavutettavaa nimeä** | 4 virheellistä elementtiä | A
- > **QW-ACT-R37: Tekstin tulee täyttää kontrastin vähimmäisvaatimukset** | 167 virheellistä elementtiä | AA
- > **QW-ACT-R65: Elementissä, jonka sisältö on koristeellista, ei ole kohdistettavia alielementtejä** | 6 virheellistä elementtiä | A

Kuva 3. QAutoEQUAL-työkalun tuottama raportti. (kuvakaappaus QAutoEQUAL, 9.9.2024)

3.2 Nykyisten teknologioiden rajoitukset

Vaikka automaattisen testauksen työkalut ovat erittäin tehokkaita teknisten saavutettavuusongelmien havaitsemisessa, ne eivät kykene arvioimaan täysin käyttäjäkokemukseen liittyviä saavutettavuuden osa-alueita, kuten sisällön ymmärrettävyyttä, loogista navigointia tai käyttäjäystävällisyyttä. Esimerkiksi kognitiiviset haasteet ja selkeän vuorovaikutuksen varmistaminen vaativat ihmisen tekemää laadullista arviointia. (Inside UX Design 2024.) Automaattiset työkalut, kuten QAutoEQUAL, keskittyvät erityisesti koodipohjaisiin ongelmiin ja WCAG 2.1 -standardin teknisten vaatimusten täyttymiseen, mutta eivät pysty arvioimaan käyttäjäkokemuksen vivahteita.

Automaattinen testaus täydentää tehokkaasti manuaalista testausta, sillä se havaitsee nopeasti yleisiä teknisiä virheitä, kuten puuttuvia alt-tekstejä tai otsikkotasoissa olevia virheitä. Parhaan tuloksen saavuttamiseksi on kuitenkin suositeltavaa yhdistää automaattinen ja manuaalinen testaus, jotta saavutettavuus voidaan varmistaa kokonaisvaltaisesti niin teknisten kuin käytettävyyteen liittyvien kriteerien osalta. Tämä lähestymistapa tarjoaa kattavamman näkymän saavutettavuuden tilasta, ja sitä suositellaan etenkin laajoissa verkkosivustoissa ja monimutkaisissa käyttöliittymissä, joissa sekä tekniset että kokemukselliset seikat ovat tärkeitä. (Inside UX Design 2024.)

4 SUUNNITTELUPERIAATTEET JA KEHITYSMETODOLOGIA

Tässä osiossa käsitellään keskeiset periaatteet, jotka ohjaavat web-käyttöliittymän kehittämistä QAutoEQUAL-työkalulle. Käyttöliittymän suunnittelu ja toteutus noudattavat käyttäjäkeskeisen suunnittelun, iteratiivisen kehitysprosessin ja saatavuusstandardien integroinnin periaatteita. Seuraavassa käydään läpi, miten nämä periaatteet on sovellettu käyttöliittymän kehitysprosessissa.

4.1 Käyttäjakeskeinen suunnittelu (User-Center Design)

Käyttäjakeskeisessä suunnittelussa otetaan huomioon, miten erilaiset käyttäjät vuorovaikuttavat käyttöliittymän kanssa ja miten heidän tarpeensa huomioidaan suunnitteluprosessissa. (Mainostoimistoluma.fi 2024.) QAutoEQUAL-sivuston web-käyttöliittymän kehittämisessä on tunnistettu kaksi keskeistä käyttäjäryhmää: uudet käyttäjät ja olemassa olevat käyttäjät. Käyttäjätarinoiden avulla pyrittiin varmistamaan, että käyttöliittymä tukee molempien käyttäjäryhmien erityistarpeita ja mahdollistaa sujuvan navigoinnin sekä teknisten työkalujen käytön ilman ohjelmointiosaamista.

4.2 Iteratiivinen kehitysprosessi

QAutoEQUAL-työkalun web-käyttöliittymän kehityksessä on sovellettu iteratiivista kehitysprosessia. Tämä tarkoittaa, että käyttöliittymää on kehitetty vaiheittain, ja suunnittelu- ja toteutusvaiheita on parannettu saatujen palautteiden perusteella. Vaikka varsinaista käyttäjäpalautetta ei vielä ole saatavilla, kehitysprosessia on ohjannut tiivis yhteistyö työnohjaajan kanssa, joka toimii myös QAutoEQUAL-työkalun kehittäjänä. Hänen asiantuntemuksensa ja ehdotuksensa ovat auttaneet ohjaamaan iteratiivisia kehitysvaiheita.

Iteratiivinen kehitysprosessi on jatkuva sykli, joka alkaa analysointivaiheesta ja etenee suunnittelun, kehityksen ja testauksen kautta. (Kuva 4.) Palautteen ja havaintojen perusteella tehdään tarvittavat muutokset, ja prosessi toistuu, kunnes käyttöliittymä saavuttaa halutun käytettävyyden ja toiminnallisuuden tason – tässä tapauksessa BETA-tason sivuston. Tämä prosessi varmistaa, että käyttöliittymä vastaa sekä teknisiä vaatimuksia että käyttäjäystävällisyyden tarpeita. (htmlburger.com 2024.)



Kuva 4. Iteratiivinen kehitysprosessi (omatekemä)

"Iteratiivinen kehittäminen tarkoittaa, että palvelua kehitetään vaiheittain ja jatkuvasti parannetaan saatujen palautteiden perusteella". (Virnex, 2024).

Tässä vaiheessa kehitysprosessin tavoitteena on varmistaa, että käyttöliittymä tukee erilaisia käyttäjäryhmiä ja että se täyttää saavutettavuuden ja käytettävyyden kriteerit. Vaikka käyttäjätestausta ei ole vielä tehty, kehitystyötä on ohjannut jatkuva arviointi ja parannukset työnohjaajan antaman palautteen perusteella.

5 SAAVUTETTAVUUSTEKNOLOGIAN KEHITYS

5.1 Vaatimusmäärittely

Käyttöliittymän vaatimukset määriteltiin käyttäjätarinoiden avulla, joissa huomioitiin kaksi pääkäyttäjärühmää:

- **Uudet käyttäjät:** Rekisteröityvät sivuston käyttäjiksi ensimmäistä kertaa.
- **Olemassa olevat käyttäjät:** Käyttäjät, jotka ovat jo luoneet tilin ja kirjautuvat sisään jatkotoimenpiteitä varten.

Vaatimukset jaotellaan perustoimintoihin:

- Käyttäjän rekisteröityminen ja kirjautuminen
- Testityökalun käyttö ja analysointi
- Tulosten selkeä esittäminen käyttäjälle
- Palautteen kerääminen työkalun jatkokehittämistä varten

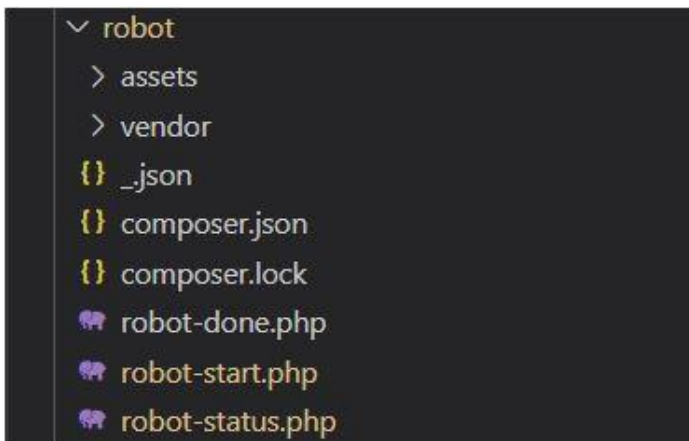
Tavoitteena oli kehittää käyttöliittymä, joka mahdollistaa saavutettavuuden tarkastelun helposti ilman ohjelmointiosaamista ja on vaivaton käyttää molemmille käyttäjärühmille.

5.2 Arkkitehtuurin suunnittelu ja teknologiavalinnat

Arkkitehtuuri rakentuu WordPress-alustalle, jossa käytettiin Hutcode Oy:n kehittämää kustomoitu teemaa. Tämä teema noudattaa WCAG 2.1 -ohjeistuksen A- ja AA-tason saavutettavuusvaatimuksia. (W3C 2018.) Valitut teknologiat tukevat saavutettavuuden toteutusta ja varmistavat sivuston rakenteen yhtenäisyyden.

- **Alusta:** WordPress
- **Teemapohja:** Hutcode Oy:n kustomoitu saavutettavuusteema
- **Käyttäjähallinta:** WordPressin käyttäjähallinta yhdistettynä WooCommerce liisäosaan, mikä mahdollistaa tulevaisuudessa kaupalliset ominaisuudet.

- **Modulaarinen rakenne:** Koodirakenne on toteutettu modulaarisesti, ja saavutettavuusskannaukseen liittyvät toiminnot on keskitetty omaan robot-kansioon, joka on osa WordPress-teeman rakenteita (Kuva 5.) Tämä kansio sisältää yksittäisiä tiedostoja eri HTMX-endpointien (kuten robot-start, robot-status ja robot-done) toteutukseen sekä tukifunktioita ja käyttäjän syötteiden käsittelyyn liittyviä tiedostoja. Modulaarinen lähestymistapa parantaa koodin ylläpidettävyyttä ja selkeyttä, ja mahdollistaa uusien toimintojen helpon lisäämisen projektin laajentuessa.



Kuva 5 Robot-kansio ja sen sisältämät oleelliset endpointit (omatekemä, Visual Studio Code)

Tämä arkkitehtuuri tukee jatkokehitystä ja skaalautuvuutta, jolloin uusia toimintoja on mahdollista lisätä ilman suurempia rakenteellisia muutoksia.

5.3 Prototyypin kehittäminen

Kehitysprosessin aluksi luotiin verkkosivustosta MVP-versio, joka sisälsi työkalun perustoiminnot ja mahdollisti verkkosivustojen saavutettavuuden arvioinnin WCAG-standardien mukaisesti. MVP-versiota kehitettiin vaiheittain, ja sitä paranneltiin palautteen ja testausten perusteella.

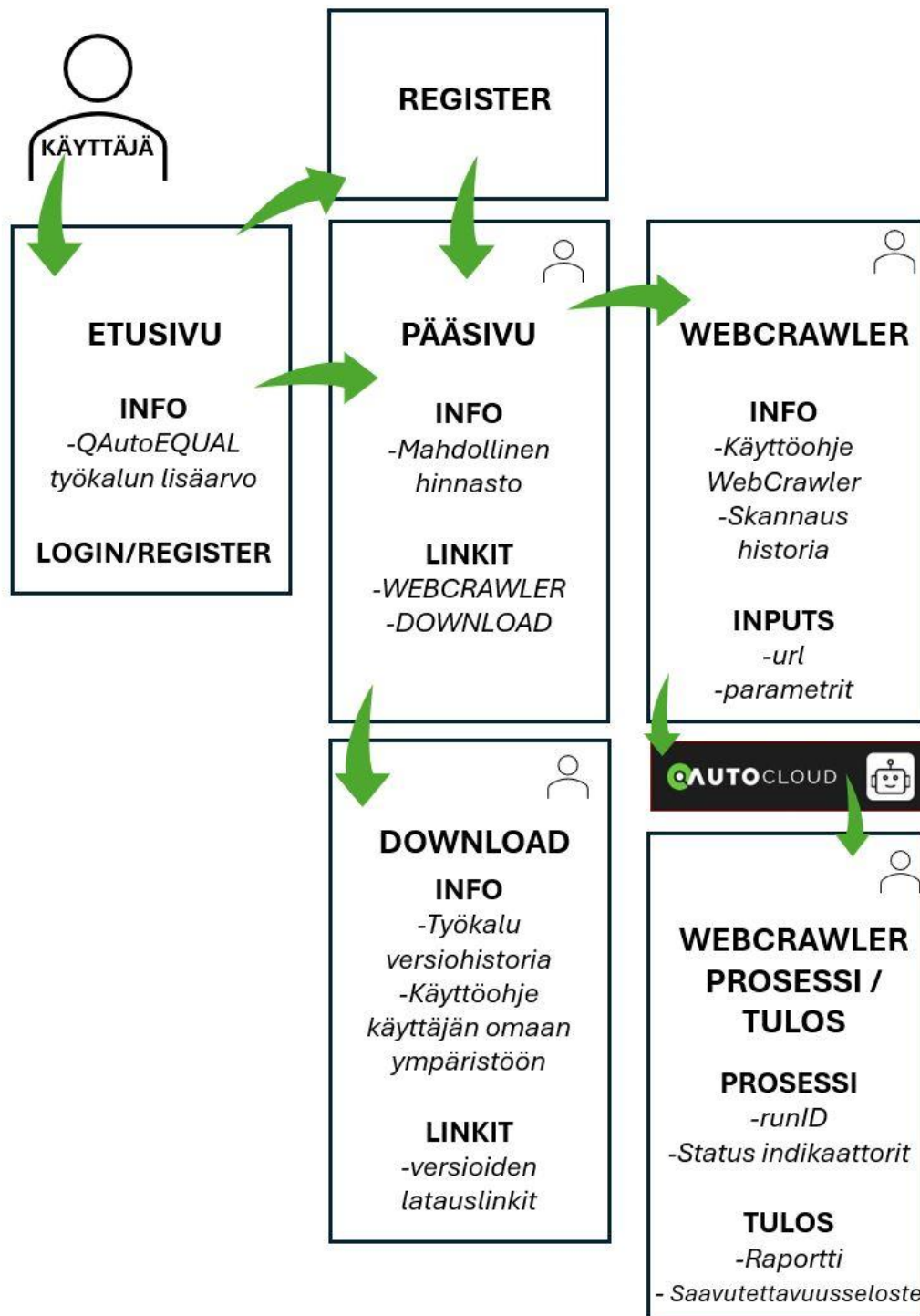
MVP QAutoEQUAL – Verkkosivuston arkkitehtuuri ja keskeiset toiminnot (Kuva6)

Keskeiset ominaisuudet:

- **Käyttäjärekisteröityminen ja kirjautuminen:** Perustoiminto, joka mahdollistaa käyttäjätilin luonnin ja sisäänkirjautumisen.
- **Saavutettavuustestien suorittaminen (WebCrawler-toiminto):** WebCrawler on toiminto, joka mahdollistaa verkkosivustojen saavutettavuuden analysoinnin WCAG 2.1 -standardin mukaisesti. Käyttäjät voivat syöttää sivuston URL-osoitteen ja asettaa testausparametreja, jonka jälkeen WebCrawler skannaa sivuston automaattisesti ja arvioi saavutettavuuden. WebCrawler-prosessi on suunniteltu tarjoamaan käyttäjille syvällistä analytiikkaa saavutettavuusongelmista, kuten puuttuvista alt-teksteistä ja navigointiongelmista.
- **Testitulosten selkeä esittäminen:** Testitulokset esitetään helposti ymmärrettävässä muodossa. Saavutettavuusseloste sekä raportti skannatusta sivustosta.
- **Palautteen kerääminen:** Mahdollisuus kerätä käyttäjäpalautetta jatkokehitystä varten.

Tämä MVP -versio verkkosivustosta toimi pohjana kehityksessä ja varmisti, että käyttöliittymä täyttää saavutettavuuden perusvaatimukset ja on helposti ymmärrettävissä. (Kuva 6.)

MVP QAutoEQUAL verkkosivusto



Kuva 6. MVP QAutoEQUAL - verkkosivuston arkkitehtuuri ja keskeiset toiminnot (omatekemä)

MVP QAutoEQUAL - verkkosivuston arkkitehtuuri ja keskeiset toiminnot (Kuva6)

- **Etusivu:** Etusivu tarjoaa käyttäjille tietoa QAutoEQUAL -työkalun arvolupauksesta ja tarjoaa rekisteröitymis- ja kirjautumisvaihtoehdot.
- **Register-sivu:** Tällä sivulla uudet käyttäjät voivat rekisteröityä QAutoEQUAL-työkalun käyttäjiksi. Rekisteröitymisen yhteydessä käyttäjä luo tilin, joka mahdollistaa työkalun ominaisuuksien käytön, kuten saavutettavuustestauksen suorittamisen ja tulosraporttien tarkastelun.
- **Pääsivu:** Sisään kirjautuville käyttäjille avautuva sivu, jolta löytyy pääsy WebCrawler -sivulle, lataussivulle ja mahdollisesti hinnoittelutiedot.
- **Download-sivu:** Täältä käyttäjät voivat ladata QAutoEQUAL ohjelmiston omaan ympäristöönsä. Työkalun versiohistoriat sekä käyttöohjeet.
- **WebCrawler-sivu:** Tällä sivulla käyttäjä voi asettaa skannausasetuksia ja käynnistää testausprosessin. Käyttäjä syöttää testattavan sivuston osoitteen ja muut parametrit.
- **Tulos-sivu (WebCrawler-prosessi):** Skannauksen tulokset näytetään selkeästi. Käyttäjä voi tarkastella skannauksen etenemistä ja lopputulosta. Tulossivulta löytyy myös saavutettavuusseloste ja muut ladattavat raportit aiemmista testeistä.

5.4 Keskeiset koodiratkaisut ja HTMX-endpointit

Tässä osiossa esitellään QAutoEQUAL-sivuston keskeisiä koodiratkaisuja, jotka tukevat saavutettavuuden tarkistuksen automaatiota ja käyttöliittymän toiminnallisuutta. Toteutus perustuu HTMX-teknologiaan, joka on integroitu WordPress-alustalle. Tämä mahdollistaa dynaamisten ja reaktiivisten elementtien käytön käyttöliittymässä ilman ylimääräisiä JavaScript-kirjastoja. HTMX-endpointien avulla voidaan hallita sujuvasti eri toimintoja, kuten skannausprosessin käynnistämistä, tilatietojen seuraamista ja tulosten esittämistä. Ratkaisut on toteutettu modulaarisesti keskittämällä kaikki saavutettavuusskannaukseen liittyvät toiminnot yhteen robot-kansioon, mikä parantaa koodin ylläpidettävyyttä ja laajennettavuutta.

5.4.1 Syötteiden käsittely ja validointi käyttöliittymässä

QAutoEQUAL-sivustolla käyttäjän syöttämät parametrit, kuten skannattava URL-osoite, sivumäärä ja kieli, otetaan vastaan HTML-lomakkeella. Lomake sisältää kenttiä, joihin käyttäjä voi syöttää skannauksen asetuksia. Näiden kenttien syötteet validoidaan ennen niiden lähettämistä HTMX-pisteelle, joka käynnistää skannausprosessin.

Lomakkeen rakenne

Käyttöliittymän lomake tarjoaa kentät URL-osoitteelle, sivumäärälle, skannauskielelle ja muille valinnaisille asetuksille. Esimerkkinä seuraavassa koodissa on lomake, jonka avulla käyttäjä voi määrittää skannauksen asetukset. (Kuva 7.)

```
<form id="robotLomake" action="<?php echo site_url('/htmx/robot-start'); ?>" method="post">
  <div class="form-group">
    <label for="url">Give the website URL which you want to scan:</label>
    <input type="text" id="url" name="url" required placeholder="https://example.com" value="<?php
    echo htmlspecialchars($url); ?>">
  </div>
  <div class="form-row">
    <div class="form-group">
      <label for="pages">Pages:</label>
      <input type="number" id="pages" name="pages" placeholder="5">
    </div>
    <div class="form-group">
      <label for="depth">Scan depth:</label>
      <input type="number" id="depth" name="depth" placeholder="1">
    </div>
    <div class="form-group">
      <label for="lang">Language:</label>
      <select id="lang" name="lang" required>
        <option value="en">English</option>
        <option value="fi">Finnish</option>
      </select>
    </div>
  </div>
  <div class="form-group">
    <label>Create Statement:</label>
    <input type="radio" id="create_statement_true" name="create_statement" value="true" checked>
    <label for="create_statement_true">Yes</label>
    <input type="radio" id="create_statement_false" name="create_statement" value="false">
    <label for="create_statement_false">No</label>
  </div>
  <button type="submit">ACCESSIBILITY SCAN</button>
</form>
```

Kuva 7. Käyttäjäparametrien määrittelylomake skannausta varten (omatekemä, Visual Studio Code)

5.4.2 Skannauksen käynnistäminen: robot-start HTMX-endpoint

Käyttöliittymän lomake lähettää käyttäjän syötteet robot-start-endpointille, jossa ne käsitellään ja validoidaan ennen skannausprosessin käynnistämistä. Ensimmäinen tarkistus tapahtuu PHP kautta JSON-muodossa saapuville syötteille. Tämä validointi varmistaa, että kaikki tarvittavat parametrit ovat paikallaan ja oikeassa muodossa. (Kuva 9.)

```
$raw_input = file_get_contents('php://input');
$input_parameters = json_decode($raw_input, true);

if (json_last_error() !== JSON_ERROR_NONE) {
    echo json_encode(['success' => false, 'error' => 'Invalid JSON input: ' . json_last_error_msg(),
        'input' => $raw_input]);
    exit();
}

if (is_array($input_parameters)) {
    foreach ($input_parameters as $input_param) {
        foreach ($parameters as $param) {
            if ($param->name === $input_param['name']) {
                $param->value = $input_param['value'];
                $_SESSION[$param->name] = $param->value;
                break;
            }
        }
    }
} else {
    echo json_encode(['success' => false, 'error' => 'Invalid input parameters']);
    exit();
}
```

Kuva 9. PHP-validointikoodi, joka varmistaa JSON-muodossa saapuvien syötteiden oikeellisuuden ennen skannauksen käynnistämistä (omatekemä, Visual Studio Code)

RobotAPI-luokka ja skannauksen käynnistäminen

RobotAPI-luokka huolehtii yhteydestä QAutoEQUAL-työkaluun ja määrittelee *robot-start*-metodin, joka lähettää skannauksen käynnistävän API-pyyntö. (Kuva 10.)

```
class RobotAPI {
    private $client;
    private $headers;
    private $base_url;
    private $token;

    public function __construct($baseUrl, $token) {
        $this->base_url = $baseUrl;
        $this->token = $token;
        $this->client = new Client([
            'base_uri' => $baseUrl,
            'timeout' => 60,
            'verify' => false
        ]);
        $this->headers = [
            'X-robot-token' => $token,
            'Content-Type' => 'application/json',
            'Accept-Encoding' => ''
        ];
    }

    public function startRobot($parameters, $useQueue = false) {
        $jsonData = json_encode($parameters, JSON_UNESCAPED_SLASHES);

        try {
            $query = $useQueue ? ['use_queue' => $useQueue] : [];

            $response = $this->client->request('POST', '/robot/start', [
                'headers' => $this->headers,
                'body' => $jsonData,
                'query' => $query,
                'verify' => false
            ]);

            $statusCode = $response->getStatusCode();
            $response_body = $response->getBody()->getContents();
            $response_data = json_decode($response_body, true);

            if ($statusCode == 200) {
                return $response_data;
            } else {
                throw new Exception("Request failed with status code $statusCode, Error message: " .
                    $response_data['Error']);
            }
        } catch (RequestException $e) {
            throw new Exception("Request failed: " . $e->getMessage());
        }
    }
}
```

Kuva 10. RobotAPI-luokan koodiesimerkki, joka vastaa skannausprosessin käynnistämisestä QAutoEQUAL-työkalun avulla käyttäen *robot-start-HTML*-endpointia (omatekemä, Visual Studio Code)

HTMX Endpoint: Skannauksen käynnistäminen

HTMX-piste *robot-start* luodaan WordPressin kautta, ja se mahdollistaa skannauksen aloittamisen käyttöliittymässä. Piste käsittelee JSON-muotoiset parametrit, jotka käyttäjä on syöttänyt, ja aloittaa skannausprosessin. (Kuva 11.)

```
add_filter('hutcode/htmx/endpoints', function ($endpoints) {
    return array_merge($endpoints, [
        [
            'slug'      => 'robot-start',
            'template' => 'qautoequal/robot/start',
        ],
    ],
    );
});
```

Kuva 11. HTMX-endpoint, joka käsittelee käyttäjän syöttämät parametrit ja aloittaa skannauksen (omatekemä, Visual Studio Code)

Käyttäjän parametrien käsittely ja skannauksen aloitus

Koodi ottaa vastaan käyttäjän syöttämät parametrit, validoi ne ja käynnistää skannauksen niiden perusteella. (Kuva 12.)

```
$parameters = [
    new Parameter('url', ''),
    new Parameter('depth', ''),
    new Parameter('lang', '')
];
$input_parameters = json_decode(file_get_contents('php://input'), true);
foreach ($input_parameters as $input_param) {
    foreach ($parameters as $param) {
        if ($param->name === $input_param['name']) {
            $param->value = $input_param['value'];
            break;
        }
    }
}
$robotAPI = new RobotAPI($base_url, $token);
$response = $robotAPI->startRobot($parameters);
```

Kuva 12. Koodiesimerkki käyttäjän parametrien käsittelystä skannauksen käynnistämiseksi (omatekemä, Visual Studio Code)

5.4.3 Skannauksen tilaseuranta: robot-status HTMX-endpoint

robot-status HTMX-endpoint käsittelee skannausprosessin tilan seurannan ja päivittää sen reaaliajassa käyttöliittymään. Kun käyttäjä aloittaa skannauksen *robot-start*-endpointin kautta, *robot-status*-endpoint pollaa skannauksen tilan palvelimelta ja päivittää sen käyttöliittymään. Tämä endpoint mahdollistaa sen, että käyttäjä näkee skannauksen etenemisen ja tilatiedot reaaliajassa, mikä parantaa käyttäjäkokemusta.

Tilaseuranta ja pollaus

Skannauksen tilan seuraaminen on toteutettu *startProgressPolling* -funktiolla, joka kutsuu *robot-status*-endpointia määritellyllä frekvenssillä. Tämä funktio tekee API-kutsun ja vastaanottaa skannauksen tilatiedot, kuten **FAILURE**, **UNSTABLE**, **QUEUE**, **RUNNING** ja **SUCCESS**. Näiden tilojen perusteella käyttöliittymä päivittyy ja näyttää käyttäjälle skannauksen tilanteen. (Kuva 13.)

```
function startProgressPolling(runId, pages, url, depth) {
  const statusUrl = `${window.location.origin}/htmx/robot-status?run_id=${runId}`;
  let doneTriggered = false;
  let unstableCount = 0;

  const updateProgress = () => {
    fetch(statusUrl)
      .then(response => response.text())
      .then(html => {
        document.getElementById('status').innerHTML = html;

        if (html.includes('COMPLETED') || html.includes('The scan encountered an error, contact our support.')) {
          clearInterval(pollingInterval);
          if (!doneTriggered) {
            doneTriggered = true;
            fetch(`${window.location.origin}/htmx/robot-done?run_id=${runId}&pages=${pages}&url=${url}&depth=${depth}`)
              .then(response => response.text())
              .then(doneHtml => {
                const statusContainer = document.getElementById('status');
                const resultsDiv = document.createElement('div');
                resultsDiv.innerHTML = doneHtml;
                statusContainer.appendChild(resultsDiv);
              });
          }
        } else if (html.includes('Server too unstable at the moment. Please check your URL input. If this problem occurs again, contact our support')) {
          unstableCount++;
          if (unstableCount >= 3) {
            clearInterval(pollingInterval);
            document.getElementById('status').innerHTML = 'Server too unstable at the moment. Please check your URL input. If this problem occurs again dont hesitate to contact our support';
          }
        } else {
          unstableCount = 0;
        }
      });
  };
}
```

Kuva 13. *startProgressPolling*-funktion koodiesimerkki, joka seuraa skannauksen tilaa ja päivittää sen käyttöliittymään dynaamisesti (omatekemä, Visual Studio Code)

Tilojen käsittely ja näyttäminen käyttöliittymässä

robot-status HTMX-endpoint hakee tilatiedot QAutoEQUAL-työkalulta käyttäen *run_id* -parametria, joka yksilöi skannauksen. Endpoint palauttaa tilatiedot HTML-muodossa, ja visuaaliset elementit, kuten spinner ja status-text, näyttävät skannauksen tilan käyttäjälle. Aikatiedot, kuten jonoon asettaminen, skannauksen käynnistyminen ja päättyminen, tarjoavat kattavan kuvan prosessin etenemisestä. (Kuva 14.)

```
add_action('qautoequal/robot/status', function () {
    if (!isset($_GET['run_id']) || empty($_GET['run_id']) || !is_numeric($_GET['run_id'])) {
        echo "Invalid run_id";
        return;
    }

    $run_id = $_GET['run_id'];

    $api_url = "";
    $token = '';
    $headers = [
        "X-robot-token: $token"
    ];

    $context = stream_context_create([
        'http' => [
            'method' => 'GET',
            'header' => $headers
        ]
    ]);

    $response = file_get_contents($api_url, false, $context);
    if ($response === FALSE) {
        die("Error occurred");
    }

    $data = json_decode($response, true);
    if (json_last_error() !== JSON_ERROR_NONE || empty($data)) {
        echo "Error parsing API response";
        return;
    }

    $status = $data[0]['status'];
    $_SESSION['scan_status'] = $status;
    $queue_timestamp = isset($data[0]['queue_timestamp']) ? $data[0]['queue_timestamp'] : null;
    $running_timestamp = isset($data[0]['running_timestamp']) ? $data[0]['running_timestamp'] : null;
    $end_timestamp = isset($data[0]['end_timestamp']) ? $data[0]['end_timestamp'] : null;

    $status_text = '';
    $spinner_class = '';

    if ($status == 'FAILURE') {
        $status_text = 'The scan encountered an error, contact our support.';
        $spinner_class = 'failed';
    } elseif ($status == 'UNSTABLE') {
        $status_text = 'Server too unstable at the moment. Please try again. If this problem occurs again, contact our support.';
        $spinner_class = 'unstable';
    } else {
        switch ($status) {
            case 'QUEUE':
                $status_text = 'QUEUE';
                $spinner_class = '-queue';
                break;
            case 'RUNNING':
                $status_text = 'RUNNING';
                $spinner_class = '-running';
                break;
            case 'SUCCESS':
                $status_text = 'COMPLETED';
                $spinner_class = '-completed';
                break;
        }
    }
}
```

Kuva 14. Koodiesimerkki *robot-status* HTMX-endpointista, joka hakee skannauksen tilan ja päivittää sen käyttöliittymään (omatekemä, Visual Studio Code)

5.4.4 Skannausten tulokset: robot-done HTMX-endpoint

Skannauksen päätyttyä *robot-done*-endpoint aktivoituu ja tarjoaa käyttöliittymässä loppuraportin käyttäjälle. Endpoint noutaa skannauksen tulokset ja esittää niistä yhteenvetolinkin, jonka kautta käyttäjä voi tarkastella raporttia ja jatkaa tarvittaessa uudella skannauksella.

Modulaarisuus ja laajennettavuus

QAutoEQUAL-sivuston koodirakenne on suunniteltu modulaariseksi, mikä mahdollistaa yksittäisten toimintojen erillisen kehittämisen ja ylläpidon. Jokainen keskeinen toiminto, kuten skannauksen käynnistäminen (*robot-start*), tilan seuranta (*robot-status*) ja lopputulosten esittäminen (*robot-done*), on toteutettu omana HTMX-endpointina. Tämä modulaarinen rakenne tukee koodin selkeyttä, vähentää päällekkäisyyksiä ja helpottaa eri komponenttien ylläpitoa.

Modulaarisuuden ansiosta ratkaisua on helppo laajentaa tulevaisuudessa. Esimerkiksi uudet endpointit tai ominaisuudet voidaan lisätä ilman merkittäviä muutoksia olemassa olevaan rakenteeseen. Tämä lähestymistapa mahdollistaa myös koodin uudelleenkäytön, koska kukin komponentti on itsenäinen, mutta yhteensopiva muiden osien kanssa.

Laajennettavuuden näkökulmasta käytetyt teknologiat, kuten HTMX ja WordPress-alusta, tukevat ratkaisun kasvua. Uudet toiminnallisuudet, kuten lisäparametrit skannausprosessille tai käyttöliittymän visuaaliset päivitykset, voidaan lisätä koodiin ilman, että se vaikuttaa merkittävästi jo olemassa olevaan koodiin tai käyttöliittymän rakenteeseen.

Tämä rakenne on erityisen hyödyllinen pitkällä aikavälillä, koska modulaarinen ja laajennettavissa oleva koodi helpottaa skaalautuvien ominaisuuksien toteutusta ja mahdollistaa ratkaisun joustavan kehittämisen käyttäjien tarpeiden ja palautteen mukaisesti.

5.5 Testaus ja iterointi

Kehityksen aikana MVP:tä testattiin iteratiivisesti ja parannettiin yhteistyössä työnohjaajan kanssa. Jokaisen kehityssyklin aikana prototyyppiä muokattiin, kunnes se täytti asetetut vaatimukset ja tavoitteet.

- **Tekninen testaus:** Jokainen kehitysvaihe arvioitiin teknisesti, jotta varmistettiin suorituskyky ja yhteensopivuus.
- **Käytettävyystestaus:** Käytettävyystestaus suoritettiin arvioimalla käyttöliittymän navigointia, ymmärrettävyyttä ja esteettömyyttä.
- **Palautte:** Palautteen perusteella MVP tehtiin muokkauksia, jotka tukivat sen saavutettavuutta ja käytettävyyttä.

Näin projekti kehittyi vaiheittain lopulliseen versioonsa, joka on teknisesti saavutettava ja käytettävyydeltään riittävä BETA-tason julkaisuun.

6 TEKNOLOGIAN SOVELTAMINEN KÄYTÄNNÖSSÄ

6.1 Kohdevalinta

Kehitysvaiheessa käytettiin skannaukseen pääosin yksinkertaisia ja perusverkosivustoja. Tämä valinta johtui siitä, että QAutoEQUAL-työkalun kehitys ja raportointiprosessi sekä skannaukseen liittyvät yksityiskohdat olivat toisen tiimin vastuulla, eikä skannausraporttien tarkka sisältö ollut tässä vaiheessa olennaista käyttöliittymän suunnittelun ja toteutuksen kannalta.

Kohteiden valinnassa keskityttiin sivustoihin, jotka tarjosivat riittävän monipuolisia elementtejä testatakseen käyttöliittymän toimivuutta ja interaktiivisuutta, erityisesti HTMX-teknologiaa hyödyntäen. Näin varmistettiin, että käyttöliittymän toiminnallisuudet vastasivat projektin tavoitteita.

6.2 Testausprosessi

Käyttöliittymän testausprosessi keskittyi sen toiminnallisuuksien ja käyttäjäystävällisyyden varmistamiseen. Tavoitteena oli varmistaa, että käyttöliittymä tukee saumattomasti skannausprosessin käynnistämistä, statuksen seuraamista ja lopputulosten tarkastelua.

Testauksen pääpaino oli seuraavissa osa-alueissa:

- **Navigointi ja käytettävyys:** Käyttäjäpolut ja HTMX-pohjaiset toiminnot, kuten lomakkeiden toiminta ja sivujen osittainen päivitys, testattiin kattavasti. Näin varmistettiin, että käyttöliittymä pysyi responsiivisena ja selkeänä eri käyttötilanteissa.
- **Interaktiiviset elementit:** Lomakekentät ja valinnat, kuten kieliasetukset ja skannaussyvyys, testattiin virheellisen syötteen estämiseksi. Lisäksi painikkeet ja ponnahdusikkunat arvioitiin käyttäjäkokemuksen sujuvuuden takaamiseksi.

- **Reaaliaikaiset päivitykset:** Skannausprosessin tilan seuraaminen ja dynaamiset päivitykset käyttöliittymässä arvioitiin simuloimalla eri tiloja esimerkiksi **QUEUE, RUNNING, FAILURE** *startProgressPolling* -funktion avulla.

6.3 Tulokset ja analyysi

Käyttöliittymän toteutus vastasi sille asetettuja tavoitteita, ja testauksen perusteella se osoittautui toimivaksi ja käyttäjäystävälliseksi. Keskeiset tulokset liittyivät käyttöliittymän responsiivisuuteen, virheiden estämiseen ja interaktiivisiin ominaisuuksiin:

- **Saumaton skannausprosessi:** Käyttöliittymä mahdollisti skannausprosessin sujuvan aloituksen ja statuksen reaaliaikaisen seuraamisen, mikä täytti projektin asettamat tekniset vaatimukset.
- **Käyttäjäystävällisyys:** Lomakkeet, valinnat ja dynaamiset päivitykset todettiin intuitiivisiksi, ja virheiden estomekanismit, kuten lomakekenttien validointi, toimivat odotetusti.
- **Reaaliaikainen tilatieto:** Skannausprosessin tilojen selkeä visualisointi ja käyttöliittymän dynaaminen päivittyminen paransivat käyttäjäkokemusta ja tukivat työkalun tehokkuutta.

Vaikka käyttöliittymä itsessään ei käsitellyt saavutettavuustestausten tarkkoja tuloksia, se tarjosi tarvittavat toiminnot näiden tulosten hallintaan ja tarkasteluun. Testitulosten perusteella käyttöliittymän toiminnallisuuksia pidettiin riittävän valmiina projektin BETA-version julkaisuun.

6.4 Käytännön hyödyt ja haasteet

Kehitysprosessin aikana QAutoEQUAL-käyttöliittymän toteutuksessa havaittiin sekä merkittäviä hyötyjä että haasteita.

Hyödyt:

- **HTMX-tekniikan tehokkuus:** HTMX mahdollisti dynaamisten toimintojen toteutuksen ilman raskaita JavaScript-kirjastoja, mikä yksinkertaisti kehitystä ja paransi suorituskykyä.
- **Modulaarinen rakenne:** Koodin modulaarinen toteutus helpotti ylläpitoa ja mahdollisti uusien toimintojen lisäämisen ilman merkittäviä rakenteellisia muutoksia.
- **Käyttäjakeskeinen suunnittelu:** Palautteen keruu ja iteratiivinen kehitys johtivat käyttöliittymään, joka vastasi käyttäjien tarpeita ja oli intuitiivinen käyttää.
- **Helppo käyttäjähallinta WordPressin kautta:** WordPress-alustan käyttö mahdollisti käyttäjien tilien tehokkaan hallinnan ja yksinkertaisti käyttäjähallintaprosessia.

Haasteet:

- **Integraatio WordPress-alustaan:** WordPressin rajapinnat ja mukautettu teema aiheuttivat joitakin rajoitteita, jotka vaativat luovia ratkaisuja.
- **Roolinjako kehitystiimien välillä:** Käyttöliittymän ja QAutoEQUAL-työkalun teknisten osien vastuunjako korosti kommunikaation ja dokumentaation merkitystä.
- **Validoinnin ja datan hallinnan monimutkaisuus:** Käyttäjäsyytöiden validointi ja prosessien hallinta edellyttivät tarkkaa suunnittelua ja testausta.

7 POHDINTA

Tämän opinnäytetyön tavoitteena oli kehittää QAutoEQUAL-työkalun käyttöliittymä, joka tukee saavutettavuuden automaattista arviointia ja parantaa käyttäjäkokemusta tarjoamalla selkeän ja helppokäyttöisen tavan seurata skannausprosessin vaiheita. Työ saavutti tavoitteensa hyvin. Käyttöliittymä toteutettiin modulaarisesti ja hyödyntäen WordPress-alustaa ja HTMX:ää, mikä mahdollisti projektin etenemisen tehokkaasti. Käyttäjäkokemukseen panostaminen näkyy reaaliaikaisena tilatietona ja yksinkertaisena pääsynä skannauksen tuloksiin.

Projektin aikana ilmeni haasteita, kuten yhteensopivuuden varmistaminen WordPressin kanssa ja HTMX-tekniikan integroiminen. Myös aikataulusuunnittelu vaati tarkkaa suunnittelua erityisesti iteratiivisen testauksen ja käyttäjäkokemuksen optimoinnin yhteydessä. Haasteista huolimatta projektin aikana kertynyt kokemus ja opit vahvistivat omaa osaamistani niin teknisesti kuin projektinhallinnallisestikin.

Työn lopputulos tarjoaa QAutoEQUAL-työkalulle lisäarvoa, mutta kehitykselle on vielä paljon mahdollisuuksia. Jatkokehityksessä voisi laajentaa työkalun ominaisuuksia lisäämällä uusia skannausparametreja. WooCommerce-integraation mahdollistama kytkös tarjoaa pohjan kaupallisten ominaisuuksien, kuten lisenssien hallinnan, kehittämiseksi.

Henkilökohtaisesti tämä projekti oli erittäin opettavainen. Se tarjosi mahdollisuuden soveltaa teknistä osaamistani käytännössä, kehittää ongelmanratkaisukykyäni ja syventää ymmärrystäni saavutettavuuden merkityksestä verkkopalveluissa. Työn aikana vahvistin myös taitojani työskentelyssä palautteen perusteella, mikä on olennainen osa ohjelmistokehityksen iteratiivista prosessia. Kaiken kaikkiaan tämä opinnäytetyö tarjosi arvokasta kokemusta, joka tukee ammatillista kehittymistäni ja antaa hyvät valmiudet tulevaisuuden haasteisiin ICT-alalla.

LÄHTEET

European Union. 2021. Accessibility requirements for ICT products and services (EN 301 549 V3.2.1.) Luettavissa: https://www.etsi.org/deliver/etsi_en/301500_301599/301549/03.02.01_60/en_301549v030201p.pdf. Luettu: 10.9.2024.

htmlburger.com. 2024. Iterative Design Process Explained. Luettavissa: <https://htmlburger.com/blog/iterative-design-process-explained/>. Luettu: 10.9.2024.

Inside UX Design. 2024. Accessibility Testing Tools and Methods for UX Designers. Luettavissa: <https://insideuxdesign.com/accessibility-testing-tools-and-methods-for-ux-designers/>. Luettu: 11.10.2024.

Itpedia 2017. Wat is handmatig testen? Luettavissa: <https://fi.itpedia.nl/2017/10/11/wat-is-handmatig-testen/>. Luettu: 28.10.2024.

Laki digitaalisten palvelujen tarjoamisesta 306/2019. Luettavissa: <https://www.finlex.fi/fi/laki/alkup/2019/20190306>. Luettu: 10.9.2024.

Mainostoimistoluma.fi. 2024. Käyttäjakeskeinen suunnittelu nettisivukehityksessä: mitä se on? Luettavissa: <https://mainostoimistoluma.fi/blogi/kayttajakeskeinen-suunnittelu-nettisivukehityksessa-mita-se-on/>. Luettu: 28.10.2024.

Q-Factory 2024. Etusivu. Luettavissa: <https://q-factory.fi/>. Luettu: 10.9.2024.

Saavutettavuusvaatimukset.fi. 2024. Digipalvelulain vaatimukset. Luettavissa: <https://www.saavutettavuusvaatimukset.fi/digipalvelulain-vaatimukset/>. Luettu: 10.9.2024.

Saavutettavuusvaatimukset.fi. 2024. Yleistä saavutettavuudesta. Luettavissa: <https://www.saavutettavuusvaatimukset.fi/yleista-saavutettavuudesta/>. Luettu: 28.10.2024.

Virnex 2024. Miten palvelumuotoilu voi parantaa asiakaskokemusta? Luettavissa: <https://virnex.fi/miten-palvelumuotoilu-voi-parantaa-asiakaskokemusta/>.

Luettu: 9.10.2024.

W3C. 2018. Web Content Accessibility Guidelines (WCAG) 2.1. Luettavissa:

<https://www.w3.org/TR/WCAG21/>. Luettu: 10.9.2024.