



VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

Ville Kurkela

DYNAAMISTEN PIENOISOHJELMIEN KEHI-  
TYS TEOLLISEN AUTOMAATION TIETOJEN  
VISUALISOINTIIN

Tekniikka  
2022

## TIIVISTELMÄ

Tekijä	Ville Kurkela
Opinnäytetyön nimi	Dynaamisten pienoishjelmien kehitys teollisen automaation tietojen visualisointiin
Vuosi	2022
Kieli	suomi
Sivumäärä	34
Ohjaaja	Mikael Jakas

---

Tämän opinnäytetyön tarkoituksena oli kehittää Noux Node Oy:n NouxCloud-ohjelmiston keräämän tiedon visualisointia. Kyseessä on teollisen esineiden internetin tiedonkeruu- ja valvontasovellus, jolla luetaan aikasarjatietoa.

Lähtötilanteessa sovelluksessa pystyi valitsemaan sovelluksen värin ja kojelautanäkymän pienoishjelmiin reunukset. Tavoitetaan sisällytettiin pienoishjelmien kohdalla taustakuva, taustaväri, kuvalinkki kuvaa esittävään pienoishjelmaan, päällekkäisyyden salliminen ja dynaaminen piilotus. Kojelautanäkymään vastaavasti staattinen leveys ja korkeus, taustakuva, taustaväri sekä zoomaus- ja vieritysmahdollisuus.

Kaikki tälle tutkimukselle asetetut tavoitteet saavutettiin. Kehitystehtävät saatiin valmiiksi ja jatkokehityskohteita saatiin pohdittua. Lisäksi tarkasteltiin Reactin ja CSS:n soveltuvuutta käyttöliittymien kehitykseen.

---

Avainsanat ohjelmistokehitys, teollinen esineiden internet, visuaalisuus, aikasarjat

## ABSTRACT

Author	Ville Kurkela
Title	Development of Dynamic Widget Layout to Visualize Industrial Automation Data
Year	2022
Language	Finnish
Pages	34
Name of Supervisor	Mikael Jakas

---

The purpose of this thesis was to further develop the data visualization of NouxCloud which was created by Noux Node Oy. NouxCloud is a data gathering and monitoring software used to read time series data.

Before the development implemented in the thesis, the user could choose color theme of the website and turn borders on and off in the dashboard settings. The goal for the thesis was set to include background image, background color, image link to image widget, allowing overlapping and dynamic hiding of the widgets to the widget settings and with the dashboard settings static width and height, background image, background color and possibility for zooming and panning of the dashboard.

All goals that were set for this thesis were met. All the development tasks were successfully finished, and further development targets were considered.

---

Keywords                      Software development, Industrial Internet of Things, edge computing and time series

# SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KUVALUETTELO

LYHENNELUETTELO

1	JOHDANTO.....	9
1.1	Sovelluksen kuvaus .....	9
1.2	Tutkimuksen tausta.....	10
1.3	Tutkimuksen kulku .....	12
2	SOVELLUKSEN NYKYTILA.....	13
2.1	Visuaalisuuteen liittyvät asetusvaihtoehdot kojelautanäkymällä ja pienoisohjelmilla .....	14
2.2	Käytetyt teknologiat selainkäyttöliittymässä.....	15
3	TAVOITETILA .....	17
3.1	Kehitystehtävien suunnittelu.....	17
3.1.1	Staattinen leveys ja korkeus kojelautanäkymälle .....	17
3.1.2	Taustakuva ja -väri pienoisohjelmaan ja kojelautanäkymään ....	17
3.1.3	Salli pienoisohjelmien päällekkäisyys .....	17
3.1.4	Zoomaus- ja vieritysmahdollisuus kojelautaan.....	18
3.1.5	Kuvalinkkien lisääminen pienoisohjelmaan .....	18
3.1.6	Dynaaminen pienoisohjelmien piilotus .....	18
3.2	Kehitystehtävien toteutus .....	18
4	TULOKSET .....	19
4.1	Staattinen leveys ja korkeus kojelautanäkymälle.....	20
4.2	Taustakuva ja -väri pienoisohjelmaan ja kojelautanäkymään.....	22
4.3	Salli pienoisohjelmien päällekkäisyys .....	23
4.4	Zoomaus- ja vieritysmahdollisuus kojelautanäkymään.....	25
4.5	Kuvalinkin lisääminen pienoisohjelmaan.....	26
4.6	Dynaaminen pienoisohjelmien piilotus .....	28
5	POHDINTA.....	31

5.1 Miksi tavoitetila saavutettiin .....	32
5.2 Reactin ja CSS:n käyttökelpoisuus .....	32
5.3 Esille tulleet jatkokehitysmahdollisuudet.....	33
LÄHTEET .....	34

## KUVALUETTELO

<b>Kuva 1.</b> Sovelluksen arkkitehtuuri. ....	10
<b>Kuva 2.</b> Kojelautanäkymä.....	13
<b>Kuva 3.</b> Kojelautakohtaiset asetukset lähtötilanteessa .....	14
<b>Kuva 4.</b> Pienoisohjelman asetukset lähtötilanteessa .....	15
<b>Kuva 5.</b> Kojelautanäkymä kehitystehtävien ollessa valmiina. ....	19
<b>Kuva 6.</b> Kojelautanäkymän asetukset kehitettynä. ....	20
<b>Kuva 7.</b> Staattisen näkymän asetukset. ....	21
<b>Kuva 8.</b> Staattinen leveys ja korkeus kojelautanäkymällä .....	21
<b>Kuva 9.</b> Yksittäisen pienoisohjelman ulkonäköasetukset .....	22
<b>Kuva 10.</b> Kojelautanäkymän taustan asetukset.....	23
<b>Kuva 11.</b> Päällekkäisyyden salliminen.....	24
<b>Kuva 12.</b> Kojelautanäkymä, jossa päällekkäisyys on sallittu.....	24
<b>Kuva 13.</b> Zoomausta ja vieritystä hallitseva liukukytkin kojelautanäkymässä. ....	25
<b>Kuva 14.</b> Kuvan 13 kytkimen sijainti kojelautanäkymässä. ....	25
<b>Kuva 15.</b> Vasemmalla kojelautanäkymä lähennettynä. Oikealla loitonnettuna. .	26
<b>Kuva 16.</b> Pienoisohjelman asetukset. ....	26
<b>Kuva 17.</b> Linkkinä toimiva NouxCloud-logo. ....	27
<b>Kuva 18.</b> Linkin kohteena oleva kojelautanäkymä. ....	27
<b>Kuva 19.</b> Pienoisohjelman näkyvyyden hallinta. ....	28
<b>Kuva 20.</b> Pienoisohjelma piilotettuna kojelautanäkymästä. ....	29
<b>Kuva 21.</b> Show hidden panels -asetuksen liukukytkin .....	29
<b>Kuva 22.</b> Show hidden panels -asetus päällä.....	30

## LYHENNELUETTELO

BSD	Berkeley Software Distribution. Unixin toinen päähaara.
CSS	Cascading Style Sheets. Rakenteisen dokumentin tyyliohje.
HTTP	Hypertext Transfer Protocol. Hypertekstin siirtoprotokolla.
HTTPS	Hypertext Transfer Protocol Secure. Hypertekstin siirtoprotokollan ja TLS/SSL-protokollan yhdistelmä.
InfluxQL	Influx Query Language. Kyselykieli
IIoT	Industrial Internet of Things. Teollinen esineiden internet.
IoT	Internet of Things. Esineiden internet.
IP	Internet Protocol. Tietoliikenneprotokolla.
I/O	Input/Output. Siirräntä.
MIT	Massachusetts Institute of Technology. Massachusettsin teknillinen korkeakoulu.
MQTT	MQ Telemetry Transport. Tiedonsiirtoprotokolla.
OASIS	Organization for the Advancement of Structured Information Standards. Voittoa tavoittelematon järjestö.
OPC UA	OPC Unified Architecture. Tiedonsiirtoprotokolla.
RFID	Radio Frequency Identification. Radiotaajuinen etätunnistus.
S7	Step 7. Tiedonsiirtoprotokolla.
SQL	Structured Query Language. Kyselykieli
SSL	Secure Sockets Layer. Salausprotokolla.

TCP	Transmission Control Protocol. Tietoliikenneprotokolla.
TLS	Transport Layer Security. Salausprotokolla.
UI	User Interface. Käyttöliittymä
UDP	User Datagram Protocol. Tietoliikenneprotokolla.
WLAN	Wireless Local Area Network. Langaton lähiverkko.



## 1 JOHDANTO

Tutkimuksessa kehitetään pienoishjelmia tietojen visualisointiin. Pienoisohjelmien esittämät tiedot kerätään toimeksiantajayrityksen NouxCloud -ohjelmistolla. Kyseessä on selainkäyttöliittymällä varustettu teollisen esineiden internetin käyttöön tarkoitettu tiedonkeruu- ja valvontasovellus, jolla luetaan aikasarjatietoa. (Noux Node Oy)

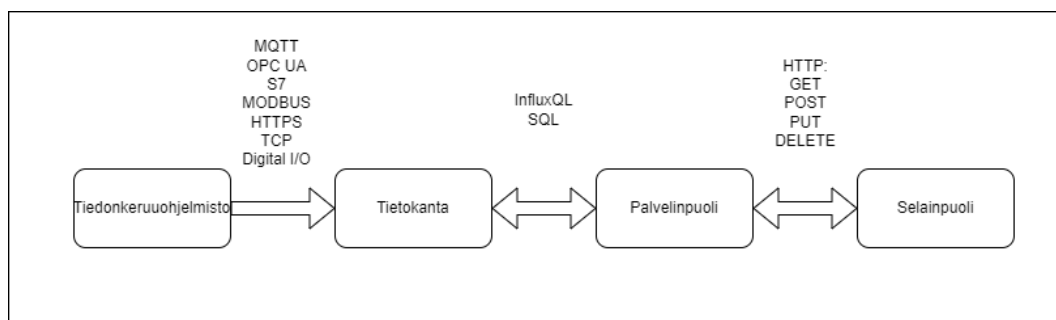
Reunasovellus toimii minkä tahansa Docker-virtualisointia tukevan laitteen tai käyttöjärjestelmän avulla. Anturidata voidaan sovelluksen kautta lähettää NouxCloudiin. Automaatiojärjestelmään tallennettua tietoa voidaan lukea tuetuilla kommunikaatioprotokollilla, joita ovat mm. MQTT, OPC UA, HTTPS, S7, Modbus TCP, Digital I/O. (Noux Node Oy)

Automaatiojärjestelmän parametrit voidaan lukea NouxCloudiin muuttujina. Näistä muuttujista voidaan rakentaa hierarkioita tiedon, laitteiden, laitosten ja sijaintien hallintaan. Muuttujista voidaan edelleen rakentaa kojetaulunäkymiä, kuten graafeja tai taulukoita. (Noux Node Oy)

Toimeksiantajana toimii ohjelmistoyhtiö Noux Node Oy. Osakeyhtiö on perustettu Suomessa vuonna 2017 ja kotipaikka on Espoo. Yrityksen visiona on luoda luokansa paras riippumaton ja helppokäyttöinen esineiden internet-alusta koneenrakentajille. (Noux Node Oy)

### 1.1 Sovelluksen kuvaus

Kyseessä on tavanomainen tietokannasta, palvelinpuolesta ja selainpuolesta koostuva sovellus. NouxCloudissa on lisäksi vielä erillinen tiedonkeruuohjelmisto, jolla tieto kerätään.



**Kuva 1.** Sovelluksen arkkitehtuuri.

Kuvassa 1 on esitelty sovelluksen toimintaperiaate. Tieto kerätään reunasovelluksella, joka on asennettu esimerkiksi tietokoneeseen, jossa on Docker-virtualisoinnin vaatimukset täyttävä käyttöjärjestelmä. Tämä tietokone on yhdistetty laitteeseen, josta tieto halutaan kerätä. Tässä vaiheessa tietoa voidaan myös suodattaa.

Tieto lähetetään tietokantaan esimerkiksi MQTT-protokollaa käyttäen. MQTT-protokollan määrittelyä hallitsee OASIS MQTT Technical Committee. (MQTT) OASIS-järjestön tavoitteena on avointen IT-alan standardien kehitys ja edistäminen. (OASIS Open)

Palvelinpuolelta kyselyt tiedon hakemiseksi tehdään InfluxQL- ja SQL-kyselykieliä käyttäen. InfluxQLia käytetään aikasarjatietojen hakuun ja SQLia käytetään reaaliaikaisiin tietokantaan tallennettujen tietojen hakuun.

Käyttöliittymästä kyselyt palvelinpuolelle tehdään http-pyyntöjä käyttäen. Tiedot esitetään käyttöliittymässä graafeissa. Graafit ovat itsessään pienoisohjelmia, joiden asetuksia käyttäjä pääsee muokkaamaan halutunlaisiksi käyttöliittymän avulla.

## 1.2 Tutkimuksen tausta

Teollisuuden toimijat integroivat uusia teknologioita valmistusprosesseihinsa, sisältäen esineiden internetin, pilvilaskennan ja -analytiikan sekä tekoäly- koneoppimisen. Tätä voidaan kutsua teollisuus 4.0:ksi tai neljänneksi teolliseksi vallanku-

moukiseksi. Älykkäät tehtaot sisältävät edistyneitä sensoreita, sulautettuja ohjelmistoja ja robotiikkaa, jotka keräävät ja analysoivat tietoa tehostaen päätöksentekoa. (IBM)

Uudet teknologiat luovat mahdollisuuden kehittää ohjelmistoja edellä mainitut tarpeet silmällä pitäen.

Esineiden internetissä tietoverkkoa käytetään esineiden ja laitteiden tiedonvälityskkeinona. Esine voi liittyä suoraan internetiin tietokoneena toimivan komponentin avulla, jolla on IP-osoite. Komponentti voi olla esimerkiksi anturi, RFID- tai WLAN-siru. Esineen ei välttämättä tarvitse kytkeytyä suoraan internetiin, vaan riittää, että esineellä on yksilöllinen tunniste, kuten postipaketin lähetystunnus. Etäluettavat sähkömittarit ovat esimerkki yleisestä esineiden internetin laitteesta. (Sanastokeskus. 2017)

Teollinen esineiden internet eli teollinen internet viittaa esineiden internetin käyttöön teollisuussektorilla ja teollisuuden sovelluksissa. Teollinen internet keskittyy koneidenväliseen kommunikaatioon, massadataan ja koneoppimiseen mahdollistaen näin tuotantolaitosten ja yritysten parantaa toimintojensa tuottavuutta ja luotettavuutta. Teollinen internet käsittää teollisuuden sovellukset käsittäen robotiikan ja terveydenhuollon laitteet sekä ohjelmiston määrittelemät tuotantomenetelmät. (Industrial Internet of Things (IIoT))

Tutkimuksen taustalla on tarve saada kojelautanäkymän nykyinen staattinen käyttöliittymä käyttäjän muokattavaksi omiin tarpeisiinsa sopivaksi. Kehitys tehdään markkina-aseman vakiinnuttaneeseen ohjelmaan ja kehityksestä on liiketoimintahyötyä yrityksen nykyisissä ja tulevaisissa asiakastapauksissa.

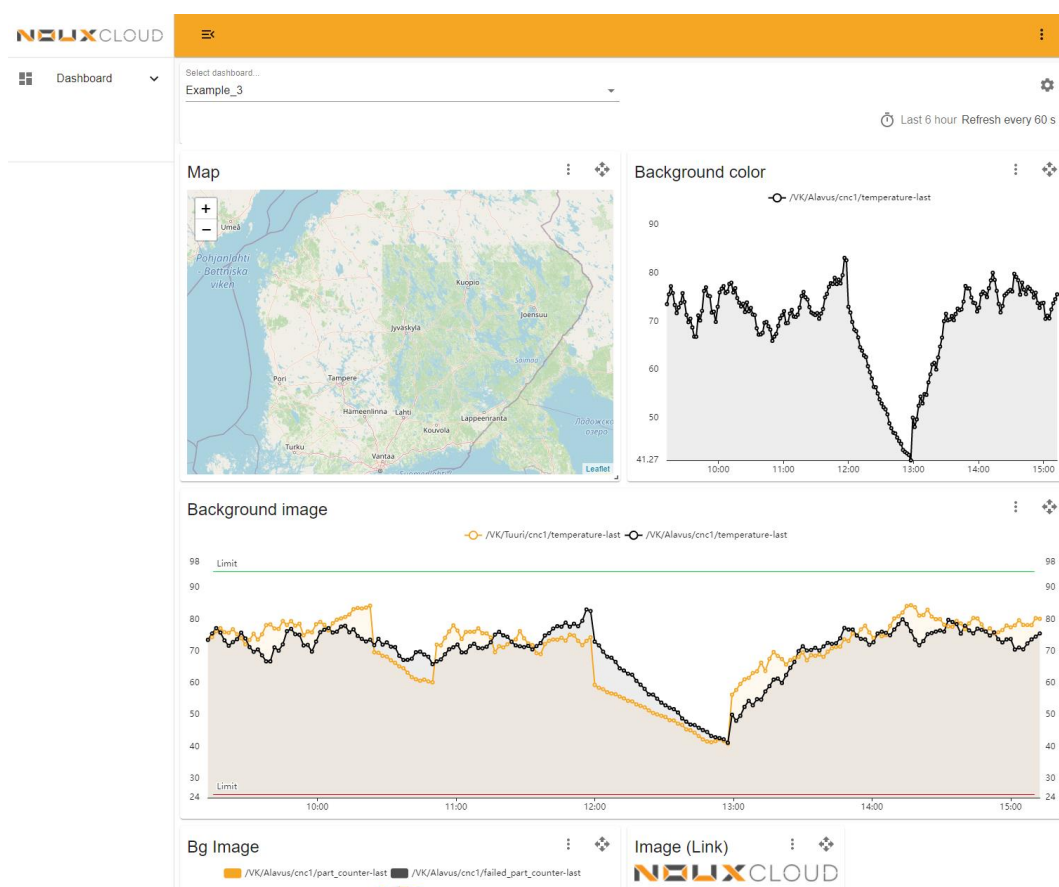
Kojelautanäkymän pienoishjelmat ovat graafeja ja taulukoita, joissa tietoa esitetään. Tutkimuksessa keskitytään sekä pienoishjelmien että kojelautanäkymän ulkoasuun ja muokattavuuden kehittämiseen.

### 1.3 Tutkimuksen kulku

Tässä luvussa esitellään toimeksiantaja ja heidän kehittämänsä sovellus. Lisäksi tarkastellaan tutkimuksen taustaa ja kulkua. Luvussa kaksi tarkastellaan tietojen visualisoinnissa käytettyjä teknologioita ja sovelluksen nykytilaa. Lisäksi käydään läpi, miten esiteltyjä teknologioita on hyödynnetty sovelluksen kehityksessä. Luku kolme käsittelee sovellukseen suunniteltuja muutoksia toteutustapoineen ja visualisoinnille asetettua tavoitetilaa. Luvussa neljä käydään läpi saavutetut tulokset ja saatiinko muutokset toteutettua halutunlaisina. Samalla tarkastellaan esille tulleita asioita ja huomioita visualisointiin liittyviä ohjelmointitehtäviä tehdessä. Tässä luvussa vertaillaan myös tavoitetilaa ja saavutettuja tuloksia. Viidennessä luvussa vertaillaan sovelluksen käytettävyyttä lähtötilanteeseen, tutkitaan käytettyjen teknologioiden sopivuutta tutkimuksen kohteena olevaan sovellukseen ja perehdytään tarkemmin tavoitetilan saavuttamiseen tai miksi tavoitteita mahdollisesti ei saavutettu.

## 2 SOVELLUKSEN NYKYTILA

Nykyisellään tiedot esitetään staattisesti pienoishjelmissä. Sovelluksen taustaväriin voi valita valkoisen tai tumman teeman välillä. Käyttäjä voi myös valita teemaväriin, toissijaisen väriin ja hälytysväriin sovelluksen asetuksissa. Pienoishjelmien järjestystä ja kokoa voi muuttaa käyttöliittymässä. Kojelautanäkymän asetuksista voidaan asettaa pienoishjelmille reunukset näkyviin. Käyttäjä voi luoda useita erilaisia kojelautanäkymiä käyttökohteen ja -tarpeen mukaan.



**Kuva 2.** Kojelautanäkymä.

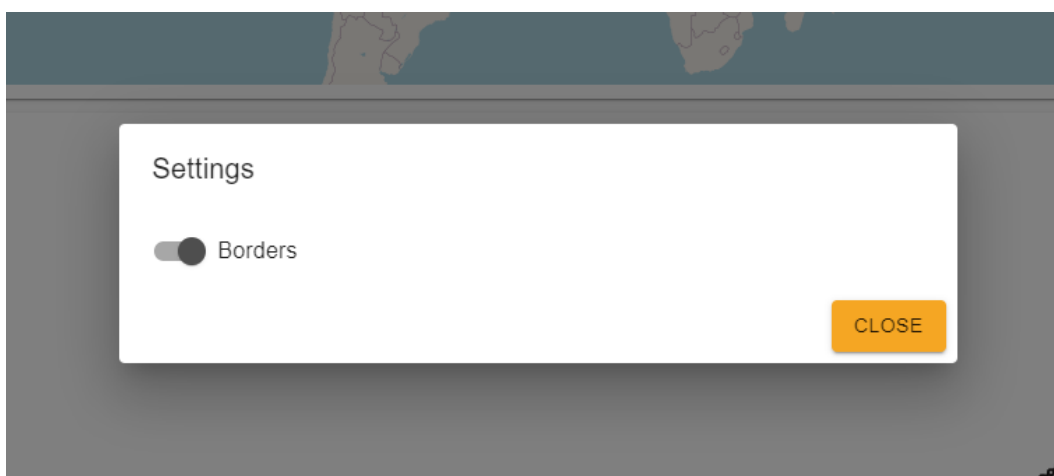
Kuvassa 2 näkyy kojelautanäkymä lähtötilanteessa. Näkymä on toteutettu ruudukkona, jossa jokaisen pienoishjelman kokoa voi muuttaa sopivaksi. Ruudukko on suunniteltu responsiiviseksi, joten kojelautanäkymää voi käyttää myös esimerkiksi

mobiililaitteilla. Näkymän pohjana on React grid layout -kirjasto, joka sisältää kehitystä helpottavia, valmiiksi rakennettuja työkaluja. React grid layout -kirjaston kehitystä hallitsee RGL. Kyseessä on X11-lisenssin omaava kirjasto.

X11-lisenssi tai MIT-lisenssi on avoin ohjelmistolisenssi. X11-lisenssi ei ole käyttäjänoikeuslisenssi, joten kyseisen kirjaston käyttö on sallittua esimerkiksi NouxCloudin kaltaisessa suljetun lähdekoodin ohjelmistossa. (GNU)

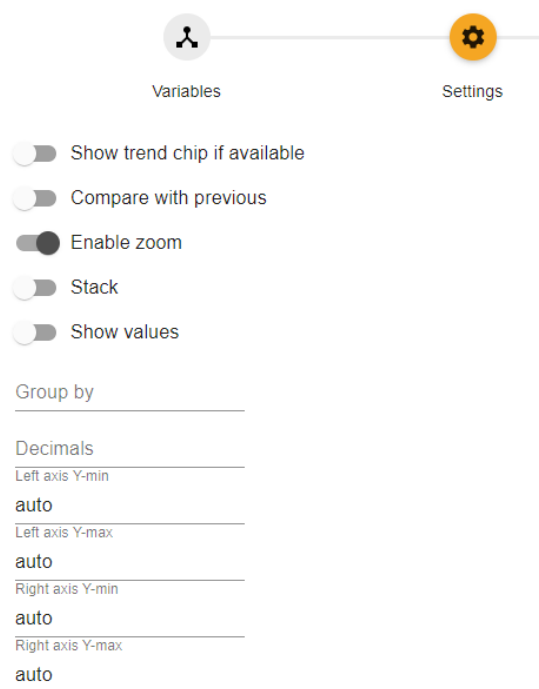
### 2.1 Visuaalisuuteen liittyvät asetusvaihtoehdot kojelautanäkymällä ja pienoishjelmissä

Datan visualisointiin liittyviä asetuksia ei ollut lähtötilanteessa juurikaan olemassa.



**Kuva 3.** Kojelautakohtaiset asetukset lähtötilanteessa

Kojelautanäkymässä asetuksena on ainoastaan pienoishjelmien rajat päälle tai pois, kuten kuvassa 3 on esitetty. Reuna-asetus luo vaikutelmaa kolmiulotteisuudesta ja tekee pienoishjelmista tietokortin näköisiä.



**Kuva 4.** Pienoisohjelman asetukset lähtötilanteessa

Kuvassa 4 näkyvät pienoisohjelman asetukset lähtötilanteessa. Osa toiminnallisuuksista riippuu käytetystä pienoisohjelmatyypistä. Esimerkiksi pylväskaavio sisältää zoomauksen haluttuun kohtaan tai jokaisen pylvään arvon näyttämisen. Esittävän tiedon näyttöväriä voi pienoisohjelmissa vaihtaa haluamakseen.

## 2.2 Käytetyt teknologiat selainkäyttöliittymässä

Tässä tutkimustyössä kehitetään ohjelmiston Nginx-alustalla toimivaa selainsovellusta. Kehitys kohdistetaan tietoa visualisoivaan kojelautanäkymään ja tässä näkyvässä käytettäviin pienoisohjelmiin. Selainsovellus on ohjelmoitu käyttäen React-kirjastoa.

Yritys päätyi Reactiin, koska se oli sovelluksen kehitystä aloittaessa laajimmin käytetty. Yrityksen edustajan mukaan React sopii hyvin myös laajempiin sovelluksiin, kun taas vastaavista kirjastoista esimerkiksi Vue sopii lähtökohtaisesti hieman pienempiin. (Berg 2022)

React on avoimen lähdekoodin JavaScript-kirjasto, jolla voidaan toteuttaa komponenttipohjaisia käyttöliittymiä. React-ohjelmat voidaan toteuttaa myös käyttäen TypeScriptiä.

Reactia käyttämällä on helppo luoda vuorovaikutteisia ja tehokkaita käyttöliittymiä. Reactissa käytetään tilakohtaisia näkymiä. Näkymän muuttuessa React hahmontaa näkymässä vain ne komponentit, joissa tieto muuttui. (React)

Stack Overflown vuoden 2020 kehittäjäkyselyn mukaan JavaScript on yleisimmin käytetty ohjelmointikieli. React on JavaScriptiä, joten tukea, vinkkejä ja ohjeita kehitykseen löytyy tarvittaessa internetistä runsaasti.

Reactin kehitystä johtaa Metalla työskentelevä ryhmä. Lisäksi yhteisö osallistuu Reactin kehittämiseen. Yhteisön on mahdollista kehittää sekä Reactia itseään että omia kirjastoja, jotka tukevat Reactia. (React team)

Yhteisön rakentamista kirjastoista esimerkiksi Material UI -kirjasto sisältää paljon erilaisia valmiita komponentteja, joita voidaan käyttää käyttöliittymän rakentamiseen nopeasti. (Material UI)

Nginx on BSD-lisenssin avoimen lähdekoodin palvelinohjelmisto. Ohjelmisto toimii HTTP-palvelimena, käänteisenä välityspalvelimena, sähköpostin välityspalvelimena sekä TCP-/UDP-protokollien välityspalvelimena. (Nginx)



### **3 TAVOITETILA**

Opinnäytetyön tekemistä suunnitellessa käytiin toimeksiantajan tuoteomistajan kanssa läpi tehtävät ja tehtiin niihin tarkennuksia. Halutut kehittämiskohteet olivat tehtävinä toimeksiantajan projektinseurantaohjelmistossa. Toimeksiantaja pyysi myös tekemään omia kehitysehdotuksia nykyisiin tehtäviin sekä kertomaan mahdollisista uusista ideoista, joita saattaisi tulla esille tehtäviä tehdessä.

#### **3.1 Kehitystehtävien suunnittelu**

Tehtävien toteutusta suunniteltiin toimeksiantajan teknisen johtajan avustuksella. Käytännön suunnittelussa keskityttiin tutkimaan tarkemmin, miten tehtävissä esitetyt määrittelyt on mahdollista toteuttaa ja mihin osaan koodia ne on parasta sijoittaa.

##### **3.1.1 Staattinen leveys ja korkeus kojelautanäkymälle**

Kojelautanäkymän asetuksista voidaan määrittellä pikselikorkeus ja -leveys kojelaudalle. Asetuksesta saadaan hyötyä esimerkiksi erittäin leveitä näyttöjä sekä zoomaus- ja vieritysmahdollisuutta käytettäessä.

##### **3.1.2 Taustakuva ja -väri pienoishjelmaan ja kojelautanäkymään**

Pienoisohjelman ja kojelautanäkymän asetuksiin lisätään mahdollisuus asettaa taustakuva ja -väri. Oletusasetuksilla kuva sovitetaan pienoishjelman tai kojelaudan kokoon. Lisäasetuksina kuvasuhteen säilyttäminen ja kuvan toistomahdollisuus. Sekä väriä että kuvaa koskevana asetuksena lisäksi läpinäkyvyys.

##### **3.1.3 Salli pienoishjelmien päällekkäisyys**

Asetuksen ollessa päällä, kojelautanäkymä sallii pienoishjelmien asettamisen joko osittain tai kokonaan toistensa päälle. Samalla lisätään mahdollisuus poistaa ruudukko pois käytöstä, jolloin pienoishjelmat voi vapaasti sijoittaa näkymään.

Ruudukon poistaminen käytöstä tulee toteuttaa siten, että asetus toimii vain, kun päällekkäisyys on sallittu.

#### **3.1.4 Zoomaus- ja vieritysmahdollisuus kojelautaan**

Zoomaus- ja vieritysmahdollisuus toteutetaan suoraan kojelautanäkymään, jolloin hiiren rullalla pyörittelemällä voidaan lähentää ja loitontaa zoomaustasoa sekä vierittää näkymää hiiren vasemmalla napilla.

Asetuksen halutaan olevan oletuksena pois päältä, mikäli kojelautanäkymälle ei ole asetettu staattista leveyttä ja korkeutta.

#### **3.1.5 Kovalinkkien lisääminen pienoishjelmaan**

Pienoisohjelmaa, jossa on mahdollista esittää kuva, muokataan siten, että kuva voi sisältää linkin käyttäjän toiseen kojelautanäkymään. Ilman linkkiä näytetään pelkästään pienoishjelmassa oleva kuva.

#### **3.1.6 Dynaaminen pienoishjelmien piilotus**

Pienoisohjelma voidaan piilottaa ehtojen täyttyessä. Pienoisohjelman asetuksiin lisätään välilehti, jossa hallitaan piilottamisen määrittämiä.

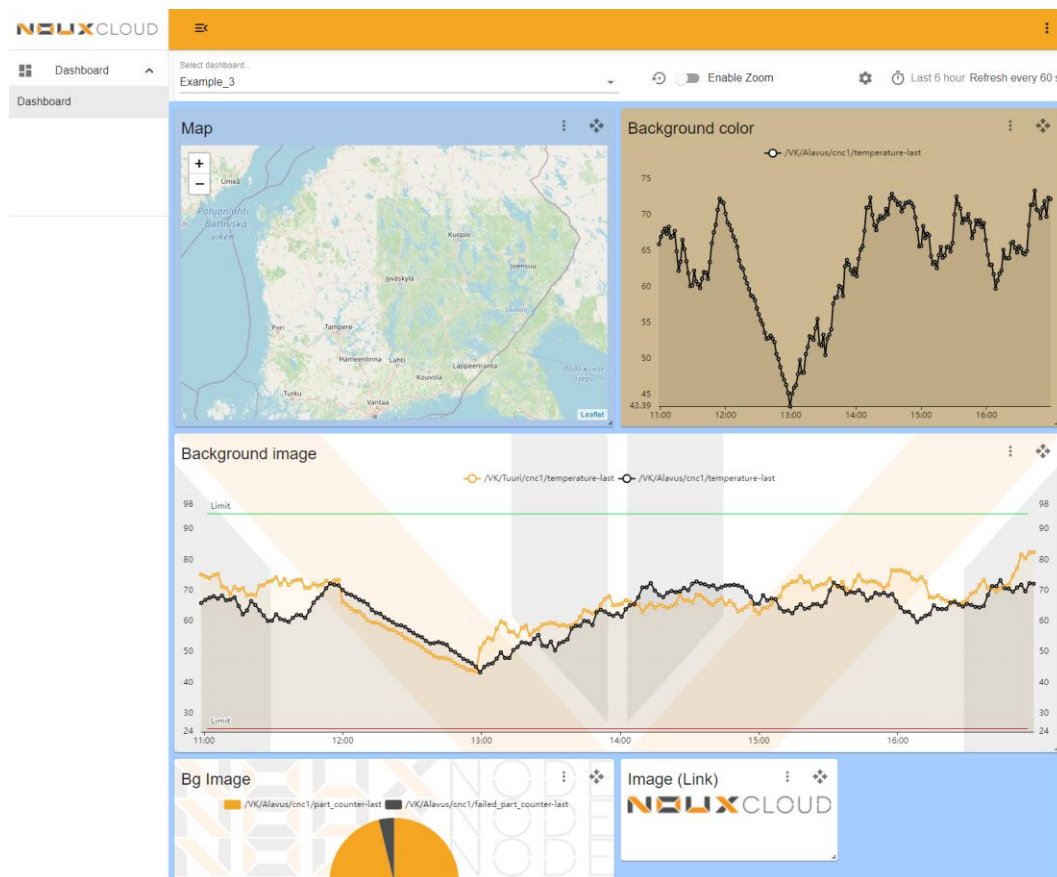
### **3.2 Kehitystehtävien toteutus**

Kehityskohteisiin liittyvät tehtävät toteutettiin tehtävä kerrallaan, mutta esimerkiksi taustakuvaa ja -väriä koskevat tehtävät esitellään tässä tutkimuksessa yhdistettynä toteutuksen samankaltaisuuden vuoksi. Jokainen tehtävä käytiin uudelleen läpi toimeksiantajan kanssa ennen toteutusta. Tämän jälkeen ohjelmointityö tehtiin itsenäisesti ja tarvittaessa ohjausta sai sekä esimieheltä että vanhemmalta ohjelmistokehittäjältä.

Kehitystyökaluna käytettiin Microsoftin Visual Studio Codea. Visual Studio Code on Windowsilla, MacOS:lla ja Linuxilla toimiva lähdekoodimuokkain, joka sisältää tuen JavaScriptille, TypeScriptille ja Node.js:lle. (Visual Studio Code)

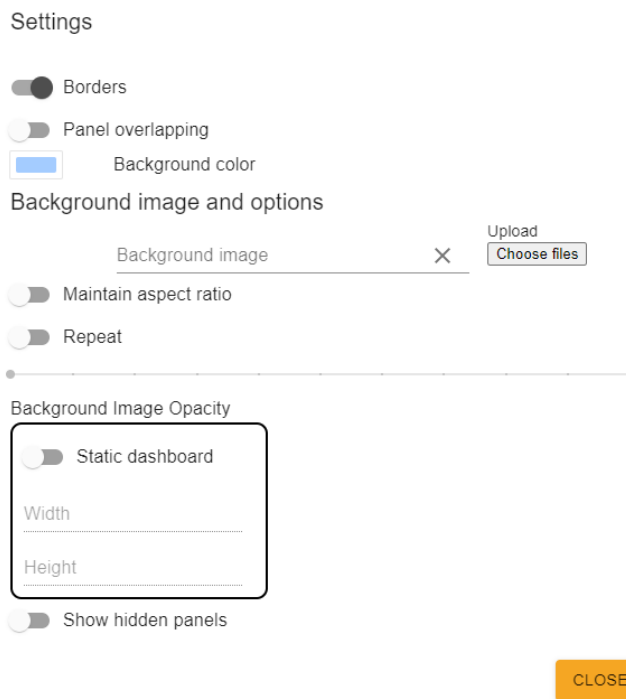
## 4 TULOKSET

Kojelautanäkymän ulkoasu muuttui huomattavasti ja uudet ominaisuudet mahdollistavat entistä yksilöllisemmin muotoillut näkymät. Osa toteutustavoista jätettiin vapaasti mietittäväksi jo suunnitteluvaiheessa.



**Kuva 5.** Kojelautanäkymä kehitystehtävien ollessa valmiina.

Kojelaudan pienoishjelmia saa nyt korostettua taustakuvilla ja -väreillä. Kuvassa 5 näkyy esimerkkipojelauta, jossa on asetettu pienoishjelmille taustakuva, taustaväri sekä itse näkymälle sininen taustaväri. Zoomaus- ja vieritysominaisuus sijoitettiin näkymän ylälaitaan ja se on oletuksena pois päältä.

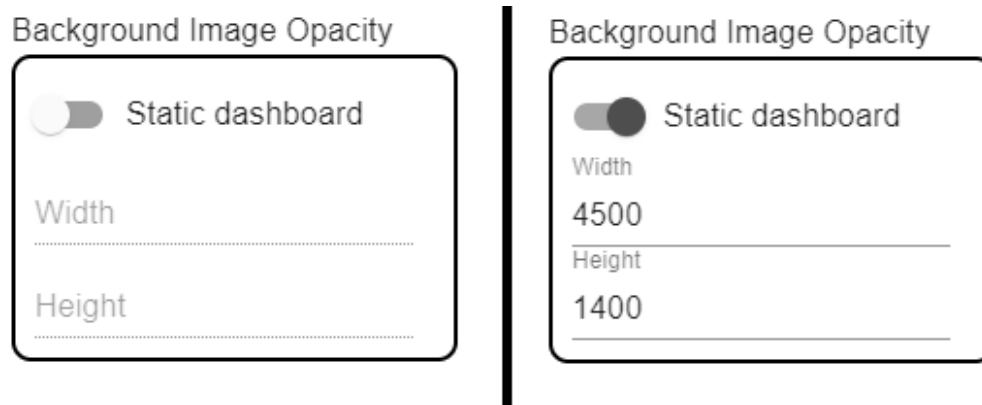


**Kuva 6.** Kojelautanäkymän asetukset kehitettynä.

Kuvassa 6 on kojelautanäkymän rataskuvakkeesta aukeava asetusvalikko. Valikko sai lisäsisältöä taustan, pienoishjelmien päällekkäisyysasetuksen ja staattisen näkymän asetuksen muodossa. Piilotettujen pienoishjelmien näyttäminen lisättiin myös tänne.

#### 4.1 Staattinen leveys ja korkeus kojelautanäkymälle

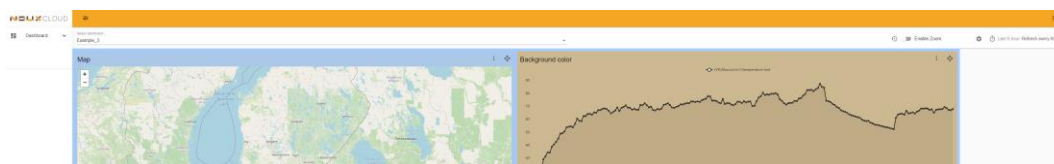
Käytettävyyden parantamiseksi staattisen korkeuden ja leveyden asetukset kehystettiin omalla reunuksellaan.



**Kuva 7.** Staattisen näkymän asetukset.

Kuvassa 7 näkyy asetus poissa päältä sekä päälle kytkettynä pituuden ja leveyden arvoilla täytettynä. Asetus laitetaan päälle liukukytimestä, jonka jälkeen leveys ja korkeus voidaan syöttää. Mikäli jompikumpi kentistä jätetään tyhjäksi, käytetään selainikkunan mittaa.

Kojelaudan leveys vaihtuu välittömästi, kun arvo syötetään ja tekstikentästä poistutaan. Tämä helpottaa sopivien arvojen hakemista kokeilemalla, mikäli näytön tarkkuus ei ole tiedossa. Mikäli käyttäjä ottaa staattisen näkymän myöhemmin pois päältä, arvot jäävät talteen. Staattisten arvojen käytössä hyödynnetään CSS-kielen korkeus- ja leveysominaisuuksia.



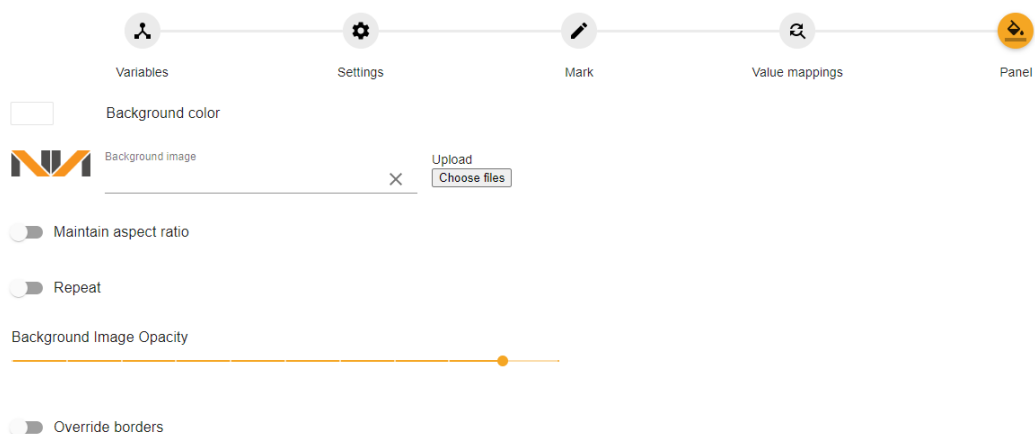
**Kuva 8.** Staattinen leveys ja korkeus kojelautanäkymällä

Kuvassa 8 näkyy esimerkki kojelaudasta, jolle on syötetty kuvassa 7 esitetyt arvot. Kuva on otettu erittäin leveältä näytöltä. Leveillä näytöillä voidaan esittää tieto pidemmiltäkin aikaväleiltä luettavassa muodossa.

## 4.2 Taustakuva ja -väri pienoisojelmaan ja kojelautanäkymään

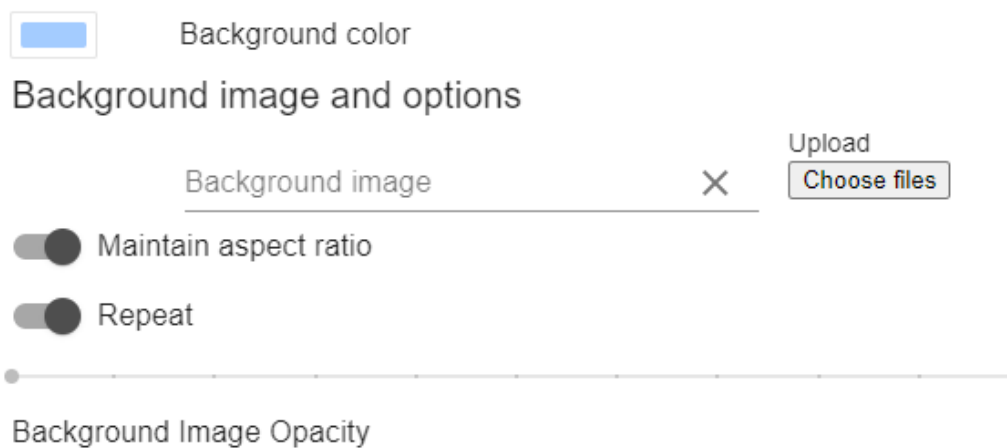
Kaikki taustakuvan ja taustaväriin muokkaamisen asetukset on toteutettu CSS-kielellä. CSS sisältää väriin, kuvaan, kuvasuhteeseen ja kuvan toistoon liittyvät ominaisuudet suoraan.

Taustakuvan läpinäkyvyydelle ei suoraan löydy ominaisuutta, joten siihen käytettiin CSS:n taustakuvaominaisuudesta löytyvää liukuväritaustan käytön mahdollistavaa arvoa. Ominaisuudelle syötettiin arvoksi sivuston taustaväriin arvo, joka sai kuvan näyttämään läpinäkyvältä.



### Kuva 9. Yksittäisen pienoisohjelman ulkonäköasetukset

Pienohjelmakohtaiset taustakuvan ja -väriin asetukset päätettiin siirtää asetuksetvälilehdeltä omaan erilliseen välilehteen, kuten kuvassa 9 näkyy. Kaikki ulkoasuun liittyvät asetukset ovat näin samassa paikassa, mikä parantaa käytettävyyttä.



**Kuva 10.** Kojelautanäkymän taustan asetukset

Oletusasetuksilla taustakuva täyttää pienoishjelman tai kojelaudan tilan. Kuvasuhteen säilyttävällä asetuksella taustalle asetetun kuvan alkuperäinen kuvasuhde säilyy. Pienoishjelman tai kojelaudan kokoa pienempää kuvaa käytettäessä, kuva venytetään taustalle sopivaksi tai kuvasuhteen säilytyksen ja toistoasetuksen ollessa päällä, toistetaan taustakuvaa niin monta kertaa kuin on tarpeen taustan täyttämiseksi.

Sekä kojelautanäkymän että pienoishjelmien asetuksissa voi käyttää yhtä aikaa taustaväriä ja taustakuvaa. Kuvassa 10 on esimerkkinä taustaväriksi asetettu sininen. Taustaväri asetetaan värivalitsimesta, joka aukeaa Background color -asetuksen vieressä olevaa ruutua napsauttamalla. Kaikki muut asetukset liittyvät taustakuvaan. Kuvan voi ladata omalta laitteeltaan tai syöttämällä linkin kuvaan Background image and options -kohtaan.

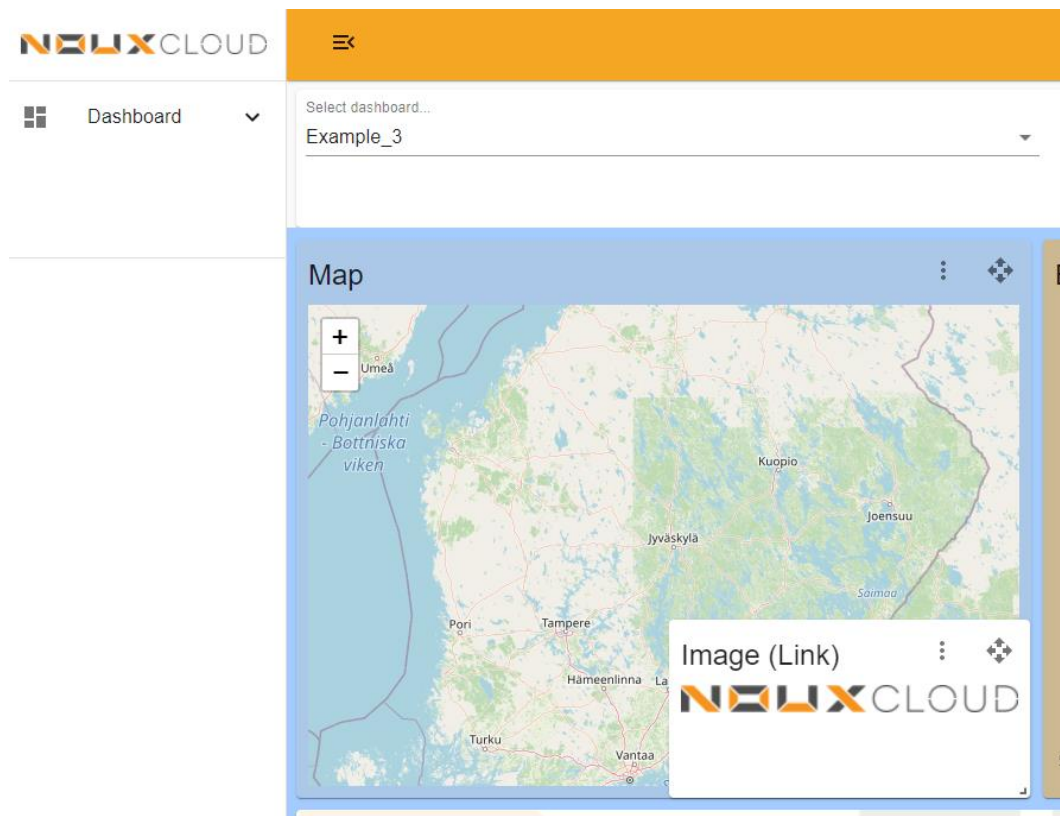
### 4.3 Salli pienoishjelmien päällekkäisyys

Päällekkäisyysasetus on käytössä olevassa React grid layout -kirjastossa valmiina oleva ominaisuus.

Panel overlapping

**Kuva 11.** Pällekkäisyyden salliminen.

Kuvassa 11 on kojelautanäkymän asetuksissa oleva liukukytin, jolla ominaisuus laitetaan päälle.



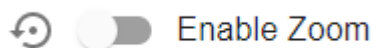
**Kuva 12.** Kojelautanäkymä, jossa päällekkäisyys on sallittu.

Asetuksen ollessa päällä, pienoishjelmia voi asettaa joko kokonaan tai osittain toistensa päälle. Näin saadaan esimerkiksi toisiinsa liittyvät pienoishjelmat osittain päällekkäin näyttötilan säästämiseksi. Kuvassa 12 on esimerkki, jossa on kaksi erikokoista pienoishjelmaa asetettu päällekkäin.



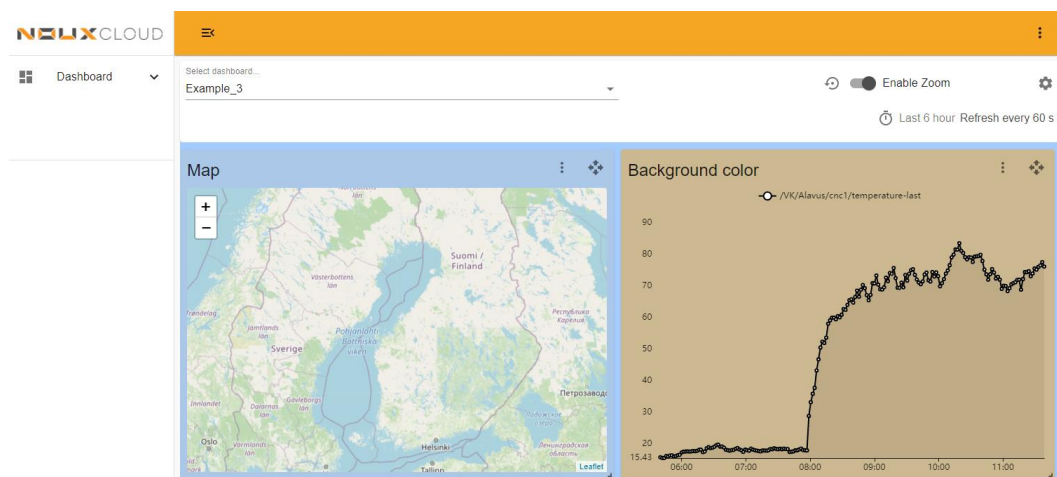
#### 4.4 Zoomaus- ja vieritysmahdollisuus kojelautanäkymään

Zoomaus- ja vieritysmahdollisuuden toteuttamiseen käytettiin React map interaction -kirjastoa, jonka kehitystä hallinnoi Strateos. Kirjasto on lisensoitu MIT-lisenssillä.



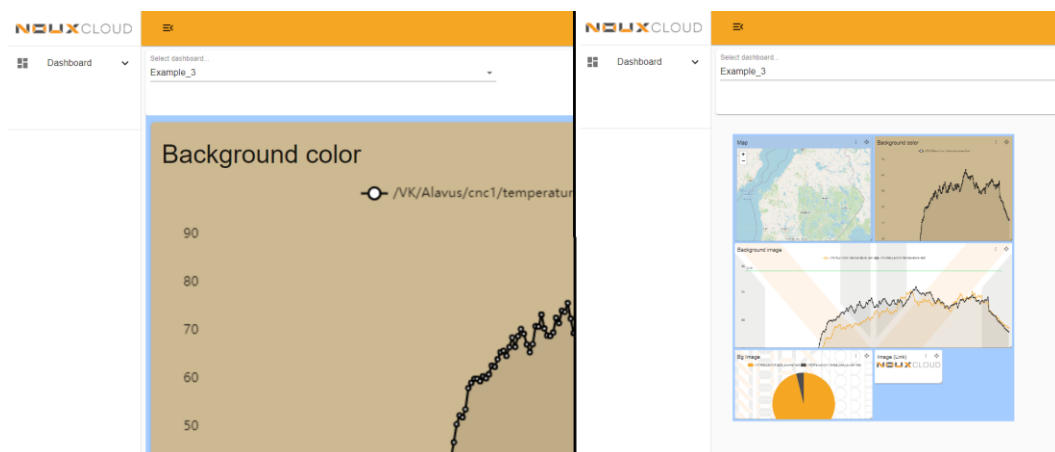
**Kuva 13.** Zoomausta ja vieritystä hallitseva liukukytkin kojelautanäkymässä.

React map interaction -kirjasto käyttää CSS-kielen skaalaus- ja siirto-ominaisuutta zoomattavan ja vieritettävän sisällön esittämiseen. Zoomausta ja vieritystä hallitaan samasta, kuvassa 13 näkyvästä, liukukytkimestä. Liukukytkimen vieressä on zoomaustason ja vierityssijainnin palautusnäppäin, jota painamalla kojelauta palautuu näkymän oletussijaintiin vasempaan ylänurkkaan.



**Kuva 14.** Kuvan 13 kytkimen sijainti kojelautanäkymässä.

Kuvassa 14 on asetettu zoomaus ja vieritys oletusarvoihin. Asetus on kytketty päälle. Zoomaus tapahtuu hiiren rullasta pyörittämällä tai sormieleillä, mikäli käytössä on kosketusnäyttö. Vieritys tapahtuu tarttumalla hiirellä kojelautaan ja raa haamalla tai kosketusnäytöllä sormieleellä.

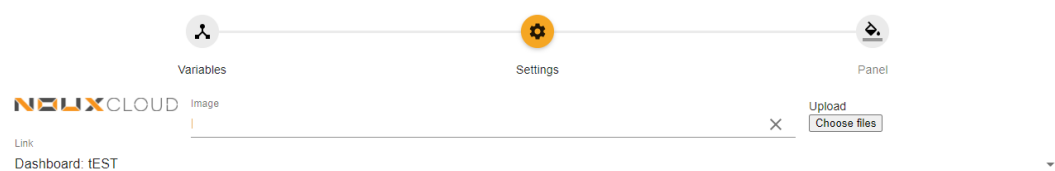


**Kuva 15.** Vasemmalla kojelautanäkymä lähennettynä. Oikealla loitonnettuna.

Loitonnuksella voidaan tarvittaessa esittää pienemmän tarkkuuden omaavilla näytöillä suurempia kojelautanäkymiä yhdellä sivulla ilman vieritystarvetta tai lähentää näkymä haluttuun pienoisohjelman, kuten kuvassa 15 on esitetty. Asetuksen poistaminen käytöstä, kun kojelautanäkymä on avattuna, palauttaa zoomauksen ja vierityksen oletusasetukset.

#### 4.5 Kvalinkin lisääminen pienoisohjelman

Kuvaa esittävän pienoisohjelman asetuksiin lisättiin mahdollisuus lisätä pudotusvalikosta käyttäjän kojelautanäkymistä haluttu näkymä linkin kohteeksi.



**Kuva 16.** Pienisohjelman asetukset.

Uusi pudotusvalikko näkyy kuvassa 16 alimmaisena. Mikäli linkkiä ei lisätä, näkyy pienohjelmassa ainoastaan asetettu kuva ilman linkkiä.



**Kuva 17.** Linkkinä toimiva NouxCloud-logo.

Pienisohjelmassa olevaa kuvaa klikkaamalla päästään haluttuun kojelautaan. Kuvassa 17 on esimerkki kuvalinkkinä toimivasta pienisohjelmasta.

Time	/VK/Tuuri/cnc1/temperature-last
2022-05-09 14:15:00	84.23390681359218
2022-05-09 14:13:12	82.19753742413624
2022-05-09 14:11:24	81.02941601462365
2022-05-09 14:09:36	80.34988431691195
2022-05-09 14:07:48	78.00423360669426

Rows per page: 5 1-5 of 200

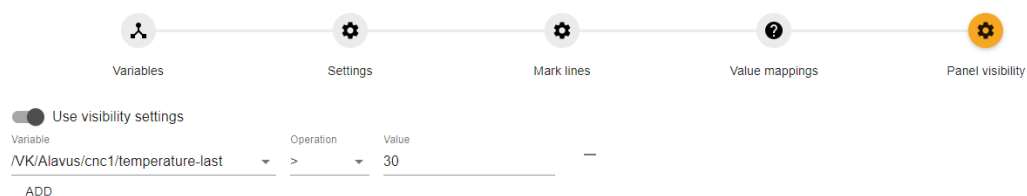
**Kuva 18.** Linkin kohteena oleva kojelautanäkymä.

Kuva 18 esittää kojelautanäkymää, johon on siirrytty kuvassa 17 näkyvän pienisohjelman kautta. Käyttötapausesimerkinä käyttäjällä voi olla tehtaassa yksittäiselle koneelle tai linjastolle luotu kojelautanäkymä. Mikäli samanlaisia koneita tai linjastoja on tehtaassa useita, voidaan jokaiselle luoda oma kojelautanäkymä. Tämän

jälkeen voidaan yhteen kojelautanäkymään lisätä kovalinkkinä toimiva pienoishjelma, josta päästään siirtymään kunkin linjaston kojelautanäkymään. Tämä nopeuttaa näkymien välillä siirtymistä.

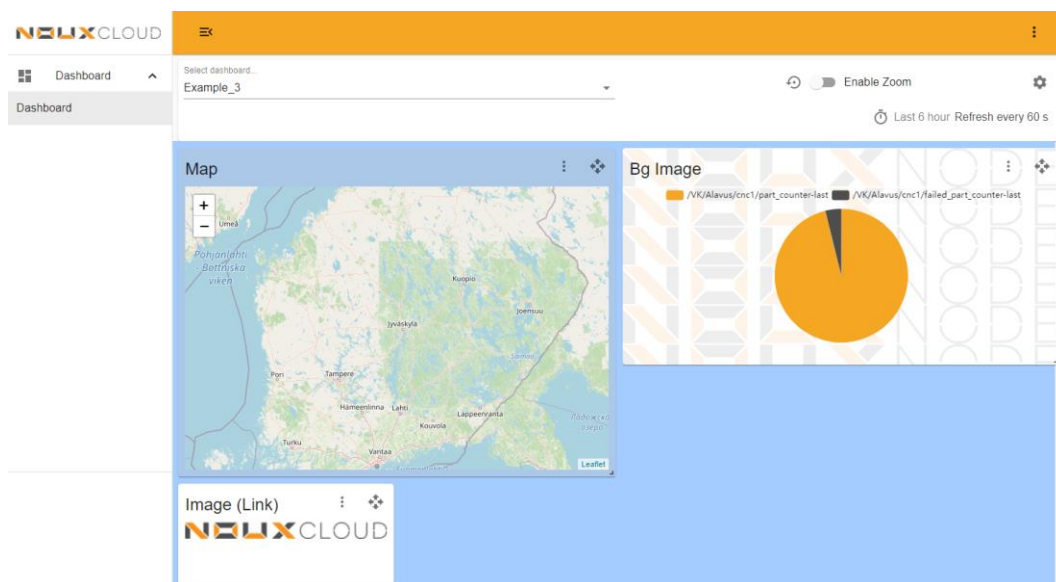
#### 4.6 Dynaaminen pienoishjelmien piilotus

Pienoisohjelma piilotetaan, kun esitettyjen muuttujien arvot eivät osu raja-arvojen sisään tai muuttuja puuttuu kokonaan.



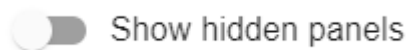
**Kuva 19.** Pienoisohjelman näkyvyyden hallinta.

Pienoisohjelman asetuksiin tehtiin uusi välilehti, josta hallitaan paneelin näkyvyysasetuksia. Kuvassa 19 näkyy asetusvälilehti. Liukukytimestä asetus laitetaan päälle, jonka jälkeen painamalla Add-nappia haetaan kyseisen pienoishjelman muuttujat. Pudotusvalikosta valitaan haluttu muuttuja, ehto ja muuttujan arvo, jolla pienoishjelma piilotetaan. Näkyvyysasetuksia voidaan lisätä useita, jolloin saadaan pienoishjelma näkyviin vain, jos muuttuja osuu tiettyjen arvojen väliin.



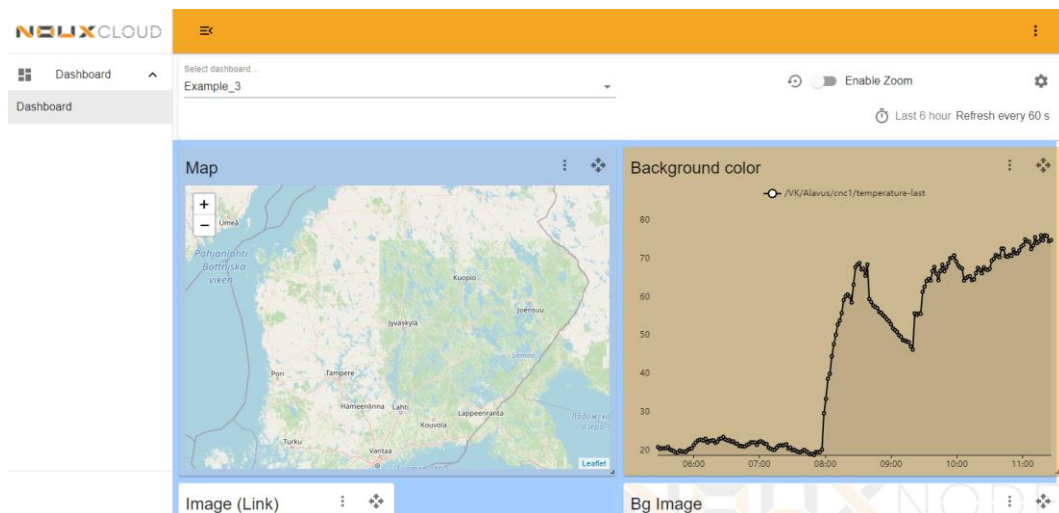
**Kuva 20.** Pienisohjelma piilotettuna kojelautanäkymästä.

Kuvassa 20 Background color -pienoisohjelma on piilotettu, koska muuttujan arvo oli tarkasteluaikavälillä yli 30. Alapuolella oleva pienisohjelma siirtyy piilotetun tilalle, mikäli se leveydeltään mahtuu tyhjäksi jääneeseen tilaan.



**Kuva 21.** Show hidden panels -asetuksen liukukytin

Piilotetut paneelit voi pakottaa näkyviin kääntämällä kuvassa 21 näkyvän Show hidden panels -asetuksen päälle kojelautanäkymän asetuksista. Näin päästään muokkaamaan paneelin asetuksia, vaikka se olisi oikeasti piilotettu juuri muokkaushetkellä.



**Kuva 22.** Show hidden panels -asetus päällä.

Kuvassa 22 Show hidden panels -asetus on päällä, jolloin myös piilotettu pienoishjelma näytetään. Pienoishjelma menee takaisin omalle paikalleen ja sen tilalla ollut siirtyä alaspäin automaattisesti. Piilotushetken pienoishjelmien sijainnit on tallennettu kojelautanäkymän asetuksiin, josta ne haetaan tarvittaessa.

## 5 POHDINTA

Työlle asetettu tavoitetilä saavutettiin. Kaikki kehitystehtävät saatiin tehtyä ja uudet sekä päivitetyt ominaisuudet toimivat testeissä määrittelyn vaatimalla tavalla. Osalle muutoksia ilmeni tai oli jo olemassa jatkokehitysmahdollisuuksia.

Tavoitteena oli myös kehittää omaa suunnitteluosaamista ja ideointia. Tässä onnistuttiin saadun tuen ansiosta. Kyky havaita jo suunnitellessa tietyn toteutustavan tai idean vaikutuksia muihin ohjelmiston osiin kehittyi opinnäytetyöprosessin edetessä.

Käytettävyys parani, koska nyt käyttäjä voi korostaa haluamansa pienoishjelmat sopivalla värillä tai kuvalla.

Dynaaminen pienoishjelmien piilotus selkeyttää kojelautaa tapauksissa, joissa pienoishjelmia on suuri määrä. Käyttäjä voi halutessaan määrittää millä arvoilla pienoishjelmia ei näytetä, jolloin saadaan haluttuja tietoja esittävät ohjelmat paremmin esille.

Käyttäjä voi lisätä kuvapienoishjelmiin linkin haluamaansa kojelautaan näkymien välillä siirtymisen nopeuttamiseksi.

Staattinen korkeus ja leveys lisää käytettävyttä todella leveillä näytöillä, koska tietoa voidaan tarkastella näkymää vierittämättä esimerkiksi viikon jaksolta kerrallaan.

Zoomaus- ja vieritysmahdollisuus mahdollistaa suurten kojetaulunäkymien tarkastelun pienemmillä ja pienemmän tarkkuuden näytöillä. Toiminto tukee siten myös staattista näkymää. Näkymä voidaan pitää suurena, vaikka osalla käyttäjistä olisi käytössään esimerkiksi tabletti tai matkapuhelin ja käyttäjät voivat tarvittaessa vierittää tai zoomata näkymää.

### 5.1 Miksi tavoitetila saavutettiin

Tehtävänannot olivat selkeitä ja suunnitteluvaiheessa tehtävät käytiin yksityiskohdaisesti läpi. Ohjelmointitehtäviin annettiin keskittyä rauhassa tehtävä kerrallaan ja apua sai aina tarvittaessa joko vanhemmalta ohjelmistokehittäjältä tai esimieheltä. Saatu tuki oli myös oikeanlaista ja jätti tilaa oman osaamisen kehittämiseksi.

### 5.2 Reactin ja CSS:n käyttökelpoisuus

Reactin avulla käyttöliittymien tekeminen on nopeaa. Hyvin suunniteltuja komponentteja voidaan kierrättää eli käyttää tarvittaessa muuallakin ohjelmassa. Kuten jo aiemmin mainittiin, React on laajasti käytetty, joten valmiita kirjastoja on runsaasti saatavilla. Esimerkiksi tässä sovelluksessa käytetty React grid layout -kirjasto.

Reactilla saadaan tehtyä nopeasti toimivia käyttöliittymiä. Käyttäjän näkymän päivittyessä päivitetään vain ne komponentit, joissa tila muuttui. Tällöin ei tarvitse jonkin tiedon muuttuessa päivittää koko sivua, mikä nopeuttaa käyttöliittymän toimintaa ja vähentää tietoliikennettä.

Esimerkkinä jokaisen pienohjelman päivittyessä päivitetään ainoastaan muuttujan arvoa esittävä kuvaaja ilman sivun uudelleenlataamista.

CSS toimii Reactin kanssa hyvin, koska tyyliohjeita voidaan kirjoittaa erilliseen tyyli-tiedostoon tai suoraan koodiin JavaScript olioina. Tutkimuksen kohteena olevassa sovelluksessa tyyliohjeita on kirjoitettu kummallakin tavalla. Useita erilaisia tyylejä tarvitsevat komponentit on helpompi kirjoittaa erillisiin tyyli-tiedostoihin. Muutamaa erilaista CSS:n ominaisuutta tarvitsevissa komponenteissa tyyliohjeet on nopeampi kirjoittaa suoraan koodin joukkoon.



### 5.3 Esille tulleet jatkokehitysmahdollisuudet

Kojelautanäkymän asetusvalikkoa voisi kehittää tekemällä valikosta luettavaman, esimerkiksi jakamalla asetukset ryhmittäin ulkoasuun vaikuttaviin ja käyttöön vaikuttaviin. Tämän toteuttaminen vaatii jonkin verran käyttöliittymäsuunnittelua.

Pienisohjelmien portaaton siirtely, päällekkäisyyden sallimisen ollessa päällä, parantaisi päällekkäisyyden sallimisen käytettävyyttä. React grid layout -kirjastossa on tätä tukevia ominaisuuksia, mutta käyttöönotto vaatii jatkosuunnittelua.

Samoin kuin kojelautanäkymän asetuksia, pienisohjelman asetusten ulkoasua voisi suunnitella luettavammaksi. Tämä tuli esille jo tutkimusraportin kirjoittamisen aikana ja näkyy esimerkiksi kuvassa 9, jossa taustakuvan, -värin ja reunusten asetukset ovat omalla välilehdellään.

Taustakuvien hallintaan voisi lisätä ominaisuuksia. Esimerkiksi erilaisia keskityksiä kuville tai marginaalien lisääminen niille. Taustaväriin taas voisi lisätä liukuvärjäyksen.

## LÄHTEET

Berg, Elias 2022. Suullinen tiedonanto 4.3.2022.

GNU license list. GNU. Viitattu 11.5.2022. <https://www.gnu.org/licenses/license-list.html#X11License>

Industrial Internet of Things (IIoT). Trend Micro. Viitattu 4.3.2022. <https://www.trendmicro.com/vinfo/us/security/definition/industrial-internet-of-things-iiot>

Material UI. Material UI. Viitattu 3.3.2022. <https://mui.com/>

MQTT. MQTT. Viitattu 5.5.2022. <https://mqtt.org/mqtt-specification/>

Nginx. Nginx. Viitattu 4.3.2022. <http://nginx.org/en/>

NouxCloud features. Noux Node Oy. Viitattu 3.3.2022. <https://www.nouxnode.fi/features.html>

OASIS. Organization for the Advancement of Structured Information Standards. Viitattu 5.5.2022. <https://www.oasis-open.org/org/>

React. ReactJs. Viitattu 3.3.2022. <https://reactjs.org/>

React Team. ReactJs. Viitattu 3.3.2022. <https://reactjs.org/community/team.html>

Tietotekniikan termitalkoot. 2017. Sanastokeskus. Viitattu 4.3.2022. [http://www.tsk.fi/tsk/termitalkoot/fi/hakemistot-267.html?page=get\\_id&id=ID335&vocabulary\\_code=TSKTT](http://www.tsk.fi/tsk/termitalkoot/fi/hakemistot-267.html?page=get_id&id=ID335&vocabulary_code=TSKTT)

Visual Studio Code. Microsoft Visual Studio Code. Viitattu 6.5.2022. <https://code.visualstudio.com/docs>.

What is Industry 4.0. IBM. Viitattu 3.3.2022. <https://www.ibm.com/topics/industry-4-0>