Ohjelmistokehittäjän päiväkirja etätyössä

Ruichao Guan

Haaga-Helia ammattikorkeakoulu

Tradenomi tietojenkäsittelyn tutkinto

Amk-opinnäytetyö

2025

Tiivistelmä

Tekijä(t)
Ruichao Guan

Tutkinto
Tradenomi tietojenkäsittely

Raportin/Opinnäytetyön nimi
Ohjelmistokehittäjän päiväkirja etätyössä

Sivu- ja liitesivumäärä
54 + 0

Opinnäytetyö on päiväkirjamuotoinen työ, jossa dokumentoidaan kahdeksan viikon ajanjakso ohjelmistokehitysalan työtehtävissä DocToDoc-yrityksessä. Yritys vaati ohjelmistokehittäjiä projektiin, jossa tavoitteena oli vanhan sivuston uudistaminen moderneilla teknologioilla. Uudistuksen avulla pystyttiin parantamaan sivuston skaalautuvuutta ja suorituskykyä. Työskentely oli pääosin itsenäistä, mutta tiivis kommunikointi tiimin ja esimiehen kanssa oli tärkeää projektin etenemisen varmistamiseksi. Viikoittaisissa palavereissa seurattiin edistymistä ja raportoitiin työtehtävien etenemisestä esimiehelle.

Seurantajakso ajoittui aikavälille 6.1.-2.3.2025. Jakson aikana asetettiin päivittäiset tavoitteet ja kirjattiin ylös työn eteneminen sekä mahdolliset haasteet. Työtehtävät vaihtelivat viikoittain, ja niihin kuului muun muassa datan käsittelyä ja synkronointia, käyttöliittymäkomponenttien suunnittelua ja optimointia. Lisäksi työssä hyödynnettiin uusia työkaluja ja kirjastoja, koodin refaktorointia ja parannettiin käyttöliittymän käytettävyyttä. Monipuoliset työtehtävät tarjosivat kokemusta sekä käyttöliittymien että palvelinpuolen ratkaisujen kehittämisestä, mikä auttoi ymmärtämään ohjelmistokehityksen kokonaisprosessia. Harjoittelun aikana pystyi hyödyntämään aiempia taitoja, mutta samalla oli tärkeää myös opiskella projektissa käytettäviä teknologioita. Erityisesti Reactin ja TypeScriptin osaaminen komponenttien rakentamisessa sekä API-rajapintojen hyödyntäminen tiedonsiirrossa olivat keskeisiä osa-alueita. Näiden taitojen käyttäminen ja kehittäminen oli tärkeää oman ammattitaidon ja työtehtävien kannalta.

Jokaisen viikon päätteeksi laadittiin seurantaviikkoanalyysi, jossa pohdittiin viikon onnistumisia ja haasteita. Nämä pohdinnat keskittyivät osaamisen kehittymiseen erityisesti uusien teknologioiden oppimisen ja ammatillisten taitojen näkökulmasta. Lisäksi tarkasteltiin ajanhallinnan parantamista laaditun aikataulun ja säännöllisten taukojen avulla sekä itsensä johtamisen kehittämistä tasapainottamalla itseopiskelua ja työskentelyä.

Koko työskentelyn lopussa pohdittiin osaamisen kehittämistä harjoittelun jälkeen, mukaan lukien ohjelmointitaitojen ylläpito ja jatkokehittäminen keskeneräisten projektien jatkamisen kautta GitHubissa sekä avoimen lähdekoodin projekteihin osallistumalla. Osaamisen syventämisen arvioitiin tukevan myös työhaastatteluissa menestymistä, erityisesti teknisissä haastatteluissa, joissa korostuvat tekninen asiantuntemus, ongelmanratkaisukyky ja kyky perustella omia ratkaisujaan selkeästi.

Yhteenvetona opinnäytetyö korosti jatkuvan oppimisen ja oma-aloitteisuuden merkitystä ohjelmistokehitysalalla. Alalla tarvitaan kykyä omaksua uutta tietoa, soveltaa sitä käytännössä sekä oman osaamisen kehittämistä suunnitelmallisesti ja johdonmukaisesti. Nämä taidot ovat keskeisiä nopeasti kehittyvässä IT-alassa ja työtehtävissä, joissa sopeutumiskyky ja proaktiivinen asenne ovat avainasemassa ammatillisessa menestyksessä.

Asiasanat
Ohjelmointi, tietokanta, data, teknologiat, kommunikointi, ajanhallinta

Sisällys

1 Johdanto

Työharjoittelu alkoi jo joulukuun alussa, mutta itse seuranta-ajanjakso on aikaväliltä 6.1.2025 – 2.3.2025. Päiväkirjanmerkintöihin kirjaan, mitä olen tehnyt kunkin päivän ajan ja miten olen edistynyt tavoitteissani ammatillisen kehittymisen ja projektin etenemisen suhteen.

1.1 Kuvaus yrityksestä, työtehtävistä ja työkaluista

DocToDoc on terveydenhuoltoalan yritys, joka on erikoistunut tarkkoihin ja laadukkaisiin käännöksiin alan termeistä ja lauseista. Yrityksen tavoitteena on terveydenhuollon viestinnän edistäminen ja kielimuurien rikkominen. IT-tiimimme toimenkuvana on vanhan sivuston uudistaminen uusilla teknologioilla, joiden avulla sivuston skaalautuvuutta ja tehokkuutta voidaan parantaa. Työskentely yrityksessä tapahtuu kokonaan etänä, ja viikoittaisilla palavereilla pystyn varmistamaan, että olen ajan tasalla projektin etenemisestä ja omista tehtävistäni. DocToDocin virallinen työkieli on englanti, mutta käytän suomea IT-mentorin kanssa kahden kesken luontevuuden ja selkeyden vuoksi. IT-mentori on tiimimme johtaja, joka jakaa työtehtäviä tiimin sisällä ja jolta voi pyytää apua ja myös palautetta ratkaisuista. Lisäksi hän varmistaa, että tiimin työskentely on sujuvaa ja että tavoitteet saavutetaan ajallaan. IT-mentori on usein vuorovaikutuksessa muiden sidosryhmien kanssa muun muassa toimeksiantajan ja asiakasryhmien kanssa.

Tarvittava osaaminen työtehtävissä on JavaScriptin ja TypeScriptin ymmärtäminen, sillä ne ovat pääteknologiat työskentelyssä. Käytössä ovat myös Next.js ja React, jotka ovat molemmat JavaScript-pohjaisia työkaluja ja joiden osaaminen aloittamiseksi oli katsottu eduksi. Uskon, että osaaminen Reactilla ja JavaScriptillä tulee kehittymään huomattavasti ja sen kautta myös edesauttamaan minua omaksumaan Next.js työkalu, jota en ole aikaisemmin käyttänyt. Ohjelmointikielien lisäksi projektinhallinnan ja versiohallinnan taidot, tulevat kehittymään varsinkin GitHubin käyttö ja Gitin kanssa työskentely. Tietokannan hallinta- ja tiedonkäsittelytaidot vahvistuvat huomattavasti, sillä projektissamme käytetään NoSQL-tietokanta MongoDB:tä, joka on dokumenttipohjainen tietokanta. Projektissamme hyödynnetään myös Tailwind CSS -tyylikirjastoa, joka eroaa perinteisestä CSS:stä käyttämällä utility-first-lähestymistapaa. Se mahdollistaa tyylien määrittämisen suoraan HTML-elementeissä valmiilla luokilla, mikä nopeuttaa kehitystä ja helpottaa responsiivisuuden suunnittelua (Kramer 2024).

1.2 Ammatillisen kehittymisen tavoitteet ja rajaukset

Ammatillisen kehittymisen tavoitteenani on kasvaa ohjelmistokehityksen alan ammattilaiseksi, oppimalla uusia teknologioita ja syventämällä osaamistani. Ammatillisen kehittymisen lisäksi tavoitteenani on myös oppia käytäntöjä etätyöstä ja kehittää taitoja, jotka tukevat tehokasta ja sujuvaa

työskentelyä virtuaalisessa ympäristössä. Pohdin etätyössä ajanhallinnan lisäksi myös itseni johtamistani, kuten työskentelyrutiinien ja omien aikataulujen luomista sekä tasapainon löytämistä työn ja vapaa-ajan välillä. Alla olevassa taulukossa (taulukko 1) esitetään, missä kohdassa opinnäytetyössä käydään läpi kyseistä tavoitetta.

Taulukko 1. Peittomatriisi päiväkirjaopinnäytetyön sisällöstä

Oman ammatillisen kehittymisen tavoitteet	**Tietoperustan luku**	**Seurantaviikko**	**Oman ammatillisen kehityksen tulokset**
Uusien teknologioiden oppiminen ja osaamisen syventäminen	1.3	1, 2, 3, 4, 5, 6, 7, 8	3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8
Ajanhallinta etätyössä	1.3	1, 2, 5, 6, 8	3.1, 3.2, 3.3, 3.5
Itsensä johtamisen taidot	1.3	1, 2, 5, 6	3.1, 3.2, 3.3, 3.5

Opinnäytetyön rajaukseksi keskityn vain omiin tehtäviini, jotka vaihtelevat usein viikoittain, enkä tarkastele muiden tiimin jäsenten työskentelyä yksityiskohtaisesti. Tällä rajauksella pystyn syventymään omaan ammatilliseen kasvuun, selkeyttämään dokumentointiani ja reflektointiani. Rajauksen avulla voin myös tarkastella paremmin ajanhallintaani ja työskentelytapojani, jotka voivat muuttua projektin ja tehtävien aikana.

1.3 Kirjallisuus, resurssit ja keskeiset käsitteet

Opinnäytetyössä ja myös päivittäisessä työskentelyssä olen hyödyntänyt paljon eri opetusmateriaaleja, blogeja ja kirjoja. Pääkirjallisuutena käytän Vasan Subramanian (2019) Pro Mern Stack -kirjaa teknologioissa ja Lorenzo Barbierin (2022) Successful remote working succinctly -kirjaa ajanhallinnassa. Pääopetusmateriaalina käytän W3Schools-sivustoa, joka tarjoaa tarvittavien teknologioiden perustiedot ja syventävää oppimateriaalia. Alla olevassa taulukossa (taulukko 2) esitellään keskeisiä käsitteitä ja työskentelyn aikana nousevia termejä.

Taulukko 2. Keskeiset käsitteet

Käsite	**Määrittely**
API	Rajapinta, joka mahdollistaa kahden ohjelmiston välisen viestinnän. Helpottaa tietojen, ominaisuuksien ja toiminnallisuuksien vaihtoa. (Gillis, A. S., Lutkevich, B. , & Nolle, T. 2024)

Back-end	Verkkosivun takaosa, joka koostuu palvelimesta, sovelluksesta ja tietokannasta (Wales, M. 2020)
BSON	JSON:n binäärimuoto, joka on optimoitu tehostamaan nopeaa tiedonkäsittelyä MongoDB:n kaltaisissa tietokannoissa. (MongoDB 2024d)
CI/CD	Jatkuva integrointi ja jatkuva toimitus/jakelu, pyrkivät nopeuttamaan ohjelmiston elinkaarta (Redhat 2023)
CRUD	Create, Read, Update ja Delete ovat neljä perustoimintoa, joita käytetään tietojen käsittelyssä tietokannassa ja ohjelmistoissa. (Codecademy 2025)
CSS	Tyylikieli, joka määrittelee HTML-sivun ulkoasun, hyödyntäen esimerkiksi värejä, fontteja ja marginaaleja. (Gupta 2025)
Express.js	Node.js:lle tarkoitettu verkkopalvelinkehys, joka yksinkertaistaa palvelinsovellusten kehitystä (Subramanian 2019, 9)
Figma	Suunnittelutyökalu, joka mahdollistaa käyttöliittymien (UI) ja käyttäjäkokemusten (UX) suunnittelun ja prototyyppien luomisen. (Figma 2019)
Front-end	Verkkosivun etuosa, jonka kanssa käyttäjät ovat vuorovaikutuksessa (Wales, M. 2020)
Full-stack	Yhdistelmä front-endistä ja back-endistä. (Wales 2020)
Git	Versionhallinta järjestelmä (W3Schools s.a. a.)
Github	Lähdekoodin isännöintipalvelu, Microsoftin omistuksessa vuodesta 2018 lähtien (W3Schools s.a. a.)
Hajautus (hashing)	Hajautus on prosessi, jossa data muutetaan kiinteäksi määräksi tavuja, jotka esitetään numeroiden ja kirjaimien merkkijonona (Moskaleva, N. 2023)
HTML	Verkkosivujen rakennuskieli, joka määrittelee sivun sisällön ja rakenteen hyödyntäen eri elementtejä, kuten otsikoita, kappaleita ja kuvia. (Gupta 2025)

JavaScript	Ohjelmointikieli, joka mahdollistaa monimutkaisten ominaisuuksien toteutuksen verkkosivulla. Keskeinen työkalu web-kehityksessä (Mozilla s.a. a.)
Jest	JavaScriptin testauskehys, jota käytetään yksikkötestaukseen. (Mulders 2022)
JSON	JavaScript Object Notation on tiedon esitysmuoto, joka on helposti luettavissa ja kirjoitettavissa. Yleisesti käytetty tiedon siirtämiseen verkkosovelluksissa (MongoDB 2024d)
JWT	JSON Web Token on tunniste, jota käytetään tietojen välittämiseen ja käyttäjän tunnistamiseen palvelimen ja asiakkaan välillä (Ibrahim, M 2024)
MongoDB	Dokumenttipohjainen tietokanta, joka on suunniteltu moderniin sovelluskehitykseen ja pilveen (MongoDB 2024a)
Next.js	Joustava React-kehys, joka tarjoaa rakennuspalikoita full-stack websovellusten luomiseen (Vercel 2025)
Node.js	Ajoympäristö, joka suorittaa JavaScript-ohjelmia. (Subramanian 2019, 7)
NoSQL	Yleisnimitys tietokannoille, jotka eivät käytä perinteistä SQL-taulurakennetta. Tiedot tallennetaan eri muodossa esimerkiksi dokumentteina, avain-arvopareina tai graafeina. (Smallcombe 2024)
Pandas	Pythonin tietojenkäsittelypaketti taulukkotyyppisille tiedoille (Chugh 30.5.2023)
Python	Python on monipuolinen ohjelmointikieli, jota käytetään esimerkiksi verkkosivujen, tekoälyn, datan käsittelyn ja automaation tekemiseen. (Coursera 21.1.2025)
React	Facebookin kehittämä JavaScript-kehys, joka hyödyntää komponentteja ja niiden tiloja tietojen muutosten seuraamiseen sekä uudelleenrenderöintiin (Mariano, C. L. 2017)
Redux	JavaScript-kirjasto, jota käytetään Reactin kanssa sovellusten tilan hallintaan (Sharma, T 2023)

Regex	Työkalu tekstinkäsittelyssä, jonka avulla voidaan etsiä, muokata tai tarkistaa tekstiä tiettyjen sääntöjen perusteella. (s.a. d W3Schools)
re.IGNORECASE	Python Regexin asetus, jossa kirjainkoot ovat riippumattomia esimerkiksi tekstinkäsittelyssä tai etsimisessä. (s.a. d W3Schools)
SHA-256	Hajautusalgoritmi, jossa lopullisessa hajautetussa arvossa on aina 256 bittiä (Jena, K. B. 2024)
strip-funktio	strip() -metodi poistaa alku- ja loppupään tyhjät merkit (esimerkiksi välilyönnit ja rivinvaihdot) merkkijonosta. Voi myös määrittää poistettavat merkit erikseen. (s.a. e W3Schools)
SQL	Relaatiotietokantoihin perustuva ratkaisu, jossa tieto tallennetaan tauluihin selkeällä rakenteella ja ennalta määritellyllä kaavalla. (Smallcombe 2024)
Tailwind CSS	Kehys, joka yksinkertaistaa verkkosuunnittelua sisältäen yli 500 valmista tyyliä ja joita voi yhdistellä (Kramer, N. 2024)
Turbopack	Turbopack yhdistää projektin lähdekooditiedostot yhdeksi tai useammaksi tiedostoksi, mikä tekee kehityksestä nopeampaa Next.js-projekteissa. (Joodi 2024).
TypeScript	Vahvasti tyypitetty ohjelmointikieli, joka pohjautuu JavaScriptiin ja parantaa kehitystyökalujen toimivuutta sekä skaalautuvuutta (Microsoft s.a.)
Utility-first	Utility-first tarkoittaa lähestymistapaa, jossa käytetään pieniä, uudelleenkäytettäviä luokkia. Esimerkiksi Tailwind CSS:ssä tyyliluokat ovat yksittäisiä ja tarkkaan määriteltyjä ominaisuuksia (Kramer, N. 2024)
Vercel	Pilvialusta, joka on suunniteltu yksinkertaistamaan verkkosovellusten käyttöönottoprosessia (Sharma, M. 2024)

2 Lähtötilanteen kuvaus

2.1 Oman nykyisen työ analysointi

Oma kokemukseni ohjelmoinnista tulee koulusta, josta olen oppinut ja saanut pohjan ohjelmointikielistä, tietokannoista, ohjelmistoalan toimintatavoista sekä ohjelmistokehityksen elinkaaresta. Olen suorittanut useita projekteja osana kursseja, jotka ovat olleet hyvin monipuolisia, sisältäen esimerkiksi mobiilisovelluksia ja full-stack projekteja. Olen työstänyt projekteja itsenäisesti, mutta myös ryhmissä, samalla oppien tiimityöskentelytaitoja, kuten tehokasta viestintää, tehtävien jakamista ja versiohallinnan käyttöä. Vapaa-ajalla olen keskittynyt enemmän mobiilisovelluksiin sekä front-end teknologioihin erityisesti verkkosuunnitteluun ja käyttäjäystävällisten käyttöliittymien suunnitteluun (UI/UX).

Konkreettisiin työtehtäviini DocToDocissa kuuluu pääasiassa ohjelmointi sekä tietokannan parissa työskentely. Ohjelmoinnin kautta pystyn parantamaan käyttäjäkokemusta ja kehittämään sivuston toiminnallisuuksia. Tavoitteenani on luoda responsiivisia käyttöliittymiä, jotka täyttävät vaatimukset ja vastaavat asiakkaiden tarpeita. Haluan myös varmistaa, että käyttöliittymät vastaanottavat tietokannoista tiedot oikein ja että ne toimivat yhdessä moitteettomasti. Sen lisäksi pyrin dokumentoimaan koodia ja ratkaisujani kommenteilla ja viesteillä, joiden avulla muut tiimin jäsenet voivat helposti ymmärtää ja muokata tai tehdä tarvittaessa jatkoratkaisuja. Ohjelmointi pitää sisällään myös virheiden tunnistamisen ja korjaamisen, jonka avulla taataan sovelluksen toimivuus ja luotettavuus.

2.2 Sidosryhmien esittely

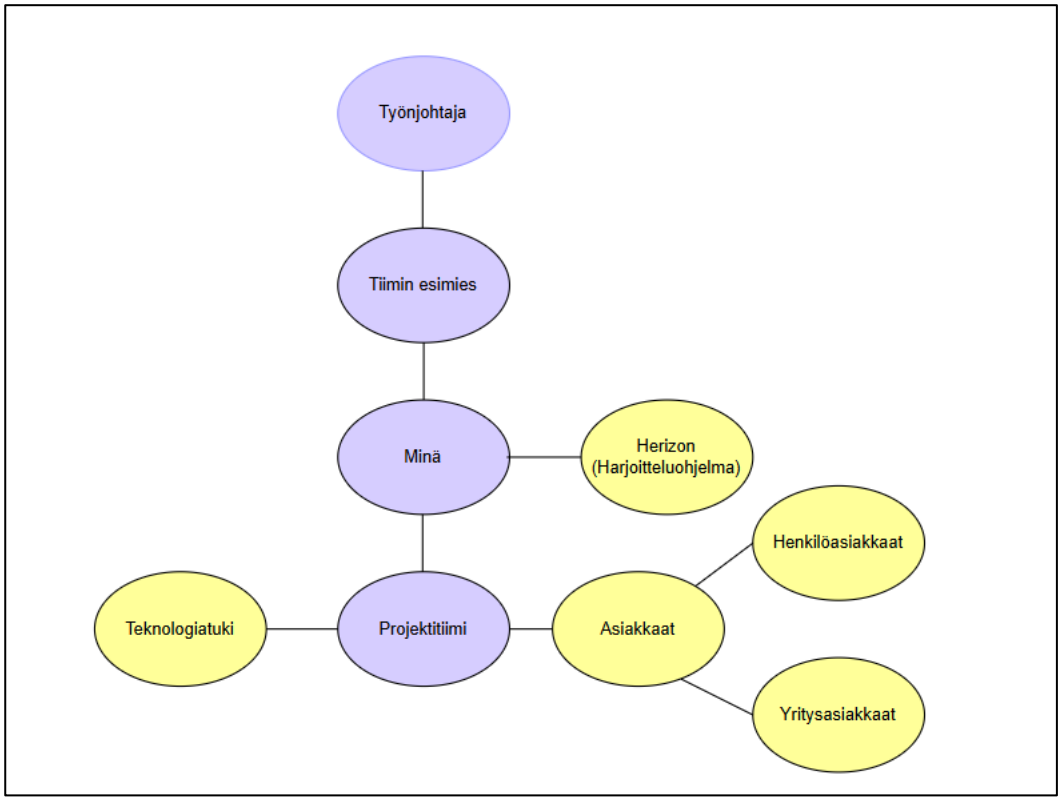
Projektin sidosryhmiin kuuluu projektitiimi, joka koostuu kuudesta junioriohjelmoijasta, kokeneemmasta ohjelmoijasta ja IT-mentorista, jolla on usean vuoden kokemus alalta. Tiimimme tehtävänä on kehittää ja toteuttaa sivusto, varmistaen sen toimivuus ja laatu. Tiimin jäsenet työskentelevät usein itsenäisesti, sillä jokaisella on vastuullaan erilaiset tehtävät. Osa vastaa esimerkiksi dokumentaatiosta ja testauksesta, kun taas toiset keskittyvät käyttäjähallintaan ja koodauksen. Opinnäytetyössä kuvaan tiimimme esimiestä IT-mentorina, joka jakaa tehtäviä ja on eniten vuorovaikutuksessa muiden sidosryhmien kanssa. IT-mentori tukee tiimin jäseniä tehtävissä ja vastaa projektin etenemisestä sekä määräaikaan valmistumisesta.

Työnantaja on projektin omistaja ja tehtävänä on myös rahoitus. Suunnanantaminen palautteen ja ehdotusten avulla mahdollistaa työnantajan edistymisen valvonnan. Työnantaja asettaa tavoitteita ja vaatimuksia projektille sekä varmistaa niiden täyttymisen sovitussa budjetissa ja aikataulussa.

Lisäsin sidosryhmään myös Herizonin, jossa olin harjoitteluohjelmassa ennen työn aloittamista. Heidän tavoitteensa on varmistaa ammatillinen kehittyminen ja uralla edistyminen viikoittaisilla tapaamisilla sekä luentojen avulla. Raportoin viikoittain edistymistäni työnantajalle ja Herizonille, jotta he voivat seurata ammatillista kasvuani, mutta myös oman reflektoinnin tueksi.

Ulkopuolisena sidosryhmänä toimivat asiakkaat, jotka myös kokeilevat prototyyppiä ja antavat palautetta yhdessä työnantajan kanssa kehitysvaiheessa muun muassa käytettävyyden, helposti ymmärrettävyyden kannalta ja ennen kaikkea soveltuvuuden osalta heidän työtehtäviinsä. Asiakkaat voidaan jakaa henkilöasiakkaisiin ja myös isompiin lääkealan yritysasiakkaihin.

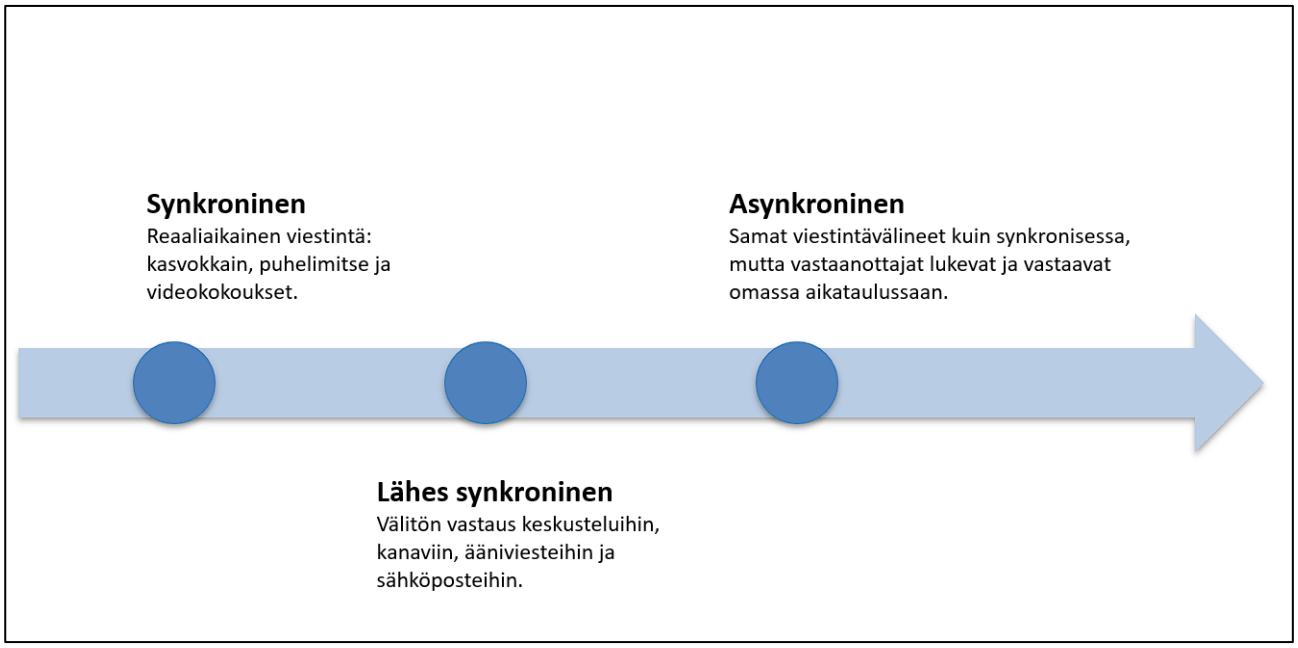
Teknologiatuki sidosryhmänä tarkoittaa asiakastukia eri teknologioissa kuten MongoDB:ssä ja Vercelissä. Nämä asiakastuet voivat olla joko tekoälyjä tai muita tarjoamia palveluita, kuten chatteja, blogeja ja keskusteluforumeja. Tämä teknologiatuki auttaa ongelmanratkaisussa, mutta myös auttaa virheiden havaitsemisissa. Ne tarjoavat esimerkiksi MongoDB:ssä vastauksia tietokantakyselyihin liittyviin ongelmiin, kun taas Vercelin tukipalvelut keskittyvät enemmän julkaisuprosessiin ja sovelluksen optimointiin. Alhaalla (kuva 1.) on havainnollistettu sidosryhmät ja niiden väliset yhteydet



Kuva 1. Sidosryhmät työharjoittelun aikana. Sidosryhmät ovat jaoteltu seuraavasti: violetilla sisäiset sidosryhmät ja keltaisella ulkoiset sidosryhmät.

2.3 Työpaikan vuorovaikutustilanteet

Etätyö tuo mukanaan haasteita, varsinkin tiimin sisäisessä kommunikoinnissa. Esimerkiksi palautteiden saaminen ja tapaamisten sopiminen vievät usein enemmän aikaa, sillä toisen osapuolen aikataulut ja saatavuus on ensin varmistettava. Ajanviennin ja vastausten odotusten aikana korostuu oma-aloitteisuus ja myös kyky priorisoida, mitä seuraavaksi tulisi tehdä, jotta pystyy edetä työskentelyssä sujuvasti ilman välitöntä palautetta. Tässä näkee asynkronisen ja synkronisen kommunikaation erot. Synkroninen kommunikaatio on tavanomainen tapa, jossa ihmiset tapaavat kasvotusten kokouksissa tai esimerkiksi puhelimitse. Se on nopein tapa tehdä yhteistyötä, mutta voi heikentää myös ihmisten tuottavuutta. (Barbier 2022, 52). Asynkronisessa kommunikaatiossa sen sijaan hyödynnetään esimerkiksi viestejä ja sähköposteja, jolloin ihmiset voivat lukea ja vastata omassa aikataulussaan. Tämä kommunikaatiotapa on käytössä työskentelyssämme. Asynkronisen kommunikaation kautta voidaan keskittyä paremmin ja olla tuottavampia (Barbier 2022, 52). Alla olevassa kuvassa (kuva 2) on havainnollistettu näiden kahden kommunikaation erot ja myös yhdistelmä, jossa molempien piirteitä on otettu lähes synkronisen tapaan. (Barbier 2022, 53)



Kuva 2. Synkroninen, asynkroninen ja lähes synkroninen (Barbier 2022, 53)

3 Seurantajakson raportointi viikkoanalyyseineen

Seurantajakso koostuu kahdeksasta viikosta, jossa asetetaan tavoitteita päivä päivältä. Päivän lopussa raportoin edistymisestäni ja arvioin, miten hyvin olin päässyt asetettuihin tavoitteisiin. Jokaisen seurantaviikon lopussa analysoin lähteitä apuna käyttäen kohtaamia ongelmia sekä edistystäni ammatillisen kehittymisen suhteen.

3.1 Seurantaviikko 1

Maanantai 6.1.2025

Päivän tavoitteenani on hienosäätää joulukuun alussa luotujen Admin-käyttäjän CRUD-toimintojen modaaleja visuaalisesti ja varmistaa niiden toimivuus eri syötteillä. Sen lisäksi pidän IT-mentorin kanssa tapaamisen, jossa hän antaa ohjeita ja tarvittavia resursseja seuraavaan tehtävään.

Sain seuraavaksi tehtäväkseni lisätä Excel-tiedoston data MongoDB:n tietokantaan. Tämä sisälsi muun muassa Excel-datan muuntamisen ensin JSON-muotoon, jonka jälkeen muunnettu data voidaan lisätä MongoDB:hen. Tässä tehtävässä on myös tärkeää seurata lisäysvaiheessa syntyneitä ongelmia ja tarkistaa, onko joitakin tietueita, joita ei pystytty lisäämään. Virheidenkäsittelyn avulla voidaan esimerkiksi tulostaa epäonnistuneet tietueet konsoliin, jotta voin analysoida syitä niiden epäonnistumisessa. Sain oikeudet ja pääsyn MongoDB tietokantaan, jotta pystyin kokeilemaan datan lisäystä tietokantaan ja tutustua syvällisemmin tietokannan toimintoihin.

Loin myös itselleni aikataulun, että milloin olen töissä päivän aikana ja kirjoitin ylös myös taukoajat. Tavoitteenani on noudattaa mahdollisimman tarkasti aikataulutustani, jossa päivän työskentely on jaettu kolmeen isoon osaan. Pyrin pitämään tauot mahdollisimman tehokkaina, jotta pystyn virkistymään työskentelyosien välissä mahdollisimman hyvin. Monet ihmiset viettävät taukonsa ruudun äärellä, jonka takia tauot eivät ole virkistäviä (Barbier 2022, 49–50). Taukojen aikana pyrin jaloittelemaan tai tekemään muita aktiviteettejä, jotka vievät minut pois ruudun ääreltä.

Tiistai 7.1.2025

Päivän tavoitteena on itseopiskella datan käsittelyä Python koodikielellä, josta minulla on vähemmän kokemusta. Haluan löytää Pythonin sisäisiä työkaluja ja kirjastoja, joita pystyn hyödyntämään datan muuntamisessa.

Päädyin käyttämään Pandas-kirjastoa, jonka avulla pystyy käsittelemään ja muuntamaan dataa. Kävin läpi kirjaston dokumentaatioita, josta löytyi muun muassa käyttöönotto ohjeet ja varmistin myös Pandas-kirjaston ajankohtaisuuden. Sen lisäksi harjoittelin Pandas-kirjaston käyttöönottoa

W3Schools opetusmateriaalien avulla. Pandas yksinkertaistaa datan käsittelyä ja analysointia Pythonilla, tehden datan kanssa työskentelystä nopeaa, joustavaa ja käyttäjäystävällistä (Chugh 2025). Näiden materiaalien avulla sain aluksi muunnettua Excel-tiedoston JSON-formaattiin, jonka jälkeen piti muuntaa JSON-datan rakenne samanlaiseksi, mitä tietokannassa on valmiiksi, jotta se olisi yhteensopiva tietokantajärjestelmän rakenteen ja tietokenttien kanssa. Tähän sisältyi muun muassa kenttien luontia, uudelleen nimeämistä ja tiedon suodattamista.

Keskiviikko 8.1.2025

Päivän tavoitteena on jatkaa datan muuntamista ja mahdollisesti saada tarkennusta sekä lisätietoja tehtävästä IT-mentorilta.

Lähetin IT-mentorille nykyisen edistyksen JSON-datan formatoinnista, ja sain siihen selvennystä. Huomasin, että Excel-datasta puuttui tietokenttiä, jotka olivat MongoDB:n tietokannassa, joten päätin jättää puuttuvat tietokentät tyhjiksi. Jätin tyhjiksi tietokentät, enkä kovakoodannut dataa, jotta pystymme erottamaan puuttuvat tiedot ja jatkokäsittely helpottuu myös sen avulla.

Siirryin myös käyttämään MongoDB Compass-työkalua, jonka avulla pystyn suorittamaan CRUD-operaatioita helpommin seuraamalla ohjeita dokumentaatiosta (MongoDB 2024b). Aikaisemmin käyttämäni MongoDB-atlas oli enemmän pilvipohjaiseen tietokantojen hallintaan, joka teki sen käytöstä hieman haastavampaa lisätoiminnallisuuksiensa takia. Sen takia siirryin MongoDB Compass-työkalun käyttöön sen yksinkertaisuuden ansioista, vaikka molemmilla pystyi tekemään perustoiminnot, joita tarvitsin työskentelyssä.

Torstai 9.1.2025

Tavoitteenani on löytää ratkaisu code-tietokentän luomiselle, jonka tarkoituksena on identifioida jokainen käännös id-kentän lisäksi. Ongelmana on, että projektissa on code:n generointitiedosto, joka on TypeScriptillä kirjoitettu, joten sen yhteensopivuus Pythonin kanssa saattaa olla ongelmallista.

IT-mentori mainitsi, että code:n generointitiedoston sijasta pystyin luomaan vastaavan käyttämällä hajautusta (hashing). Sain tarkennusta, että pystyin käyttämään tiedon hajautukseen SHA-256-algoritmia, minkä jälkeen aloin perehtyä tarkemmin ja tutkia sen materiaaleja. Tämä hajautus algoritmi mahdollisti yksinkertaisen ratkaisun ja testailin code-tietokentän luomista hajautuksella paikallisesti, sillä tietokannan pääsyssä ilmeni ongelmia.

Perjantai 10.1.2025

Tavoitteenani on saada valmiiksi hajautuksen logiikka ja päivittää MongoDB:n käännökset syöttämällä puuttuville code-tietokentille uniikit arvot. Sen lisäksi haluan varmistaa käännösdatan oikeamuotoisuuden puhdistamalla, esimerkiksi ylimääräiset isot kirjaimet ja välilyönnit.

Sain tiedon hajautuksen toimimaan, ja parantelin koodia selventämällä ja tiivistämällä. Loin hajautuslogiikan, joka luo coden käännöstekstin perusteella ja ottaa hajautuksesta ensimmäiset 8 merkkiä, jolloin jokaisella käännöksellä on eri code-arvot.

Jouduin testaamaan paljon käyttämällä tulostusfunktiota (print) tietyissä kohdissa, jotta pystyin tunnistamaan tiedon arvot ennen päätepistettä. Tutustuin myös Pythonin re-työkaluun, jonka avulla pystytään muokkaamaan, korvaamaan ja etsimään tiettyjä kirjaimia tai tekstejä. Re-työkalusta löytyi hyödyllisiä funktioita, kuten re.IGNORECASE, jonka avulla pystyin hakemaan dataa välittämättä pienistä ja isoista kirjaimista. Käytin myös Pythonin strip-funktiota, jolla pystyin pääsemään eroon käännöstekstien ylimääräisistä välilyönneistä ja rivinvaihdoista tekstin alussa sekä lopussa.

Seurantaviikkoanalyysi 1

Viikko kului nopeasti itseopiskelussa ja datan parsimisessa. Sain tehtäväni valmiiksi ja viikonlopun aikana lisäsin koodini avulla Excel-datan muunnettuna oikeaan muotoon DocToDocin tietokantaan MongoDB:ssä. Vaikka sain tehtäväni valmiiksi, jatkoin viikonloppuna myös itseopiskelua käytetyistä teknologioista kuten Pandas, Regex, SHA-256, sillä uskon tarvitsevani niitä myöhemmin tässä projektissa. Haastavin tilanne viikolla oli todennäköisesti parsimisen aloitus varsinkin Pythonilla, sillä kokemukseni Pythonilla oli hyvin vähäinen ylipäätänsä. Vaikka mainitsin tästä IT-mentorille, hän kuitenkin suositteli Pythonilla kokeilemista, sillä hänen kokemuksensa mukaan Pythonilta löytyi paljon kattavia työkaluja tehtävään. Opin kuitenkin datan muuntamisen ja integroinnin perusteet maanantaina ja tiistaina, jolloin sain luotua jonkinlaisen JSON-formaatin Excelistä. Huomasin myös, kuinka tärkeää on kysyä palautetta aktiivisesti muilta ihmisiltä, tässä tapauksessa IT-mentorilta. Palautteen avulla pystyin saavuttamaan halutun lopputuloksen ja kehittämään omia taitojani huomattavasti. Monesti pienet vinkit ja parannusehdotukset voivat tehdä ison eron ja ne myös antavat uuden näkökulman ongelmanratkaisussa.

Aikataulutuksen seuraaminen oli melko vaivatonta viikon aikana, koska olin luonut aikataulun yhteensopivaksi päivittäisen energiatason kanssa. Energiatasot vaihtelevat päivän aikana, joihin on monia eri tekijöitä ja jokaisella ihmisellä on omat mieltymykset (Barbier 2020, 50). Muokkasin kuitenkin sen verran, että iltapäivän työskentelyaika oli hieman lyhyempi, sillä huomasin energiatasoni olevan alhainen siihen aikaan. Yritin luoda päivittäistä rutiinia etätyöskentelyssä, jotta tottuisin paremmin siihen, että työskentelen kokonaan kotona ja hallitsen omaa aikaani paremmin. Itselleni oli

aluksi haastavaa erotella vapaa-aika ja työaika, sillä usein vapaa-ajalla saattaa herätä ideoita ja ratkaisuja, jotka edesauttavat projektin etenemisessä. Olen kuitenkin tehnyt päätöksen, että kirjoitan ratkaisun tai ajatuksen ylös ja jatkan sen työstämistä seuraavana työpäivänä. Tämän lähestymistavan ansiosta olen pystynyt pitämään selkeän eron työn ja vapaa-ajan välillä, mutta myös onnistunut pitämään työasiat suurimmaksi osaksi poissa mielestä vapaa-ajalla. Kotona oleminen koko ajan voi myös hermostuttaa, jolloin säännöllinen liikunta ja yksinkertaisesti käveleminen on olennaista hyvinvoinnin kannalta (Barbier 2020, 70). Olen huomannut, että tauon pitäminen erityisesti ulkona virkistää ihmistä ja auttaa jaksamaan paremmin loppupäivän niin henkisesti kuin myös fyysisesti.

Alla olevassa kuvassa (kuva 3.) esitetään kyselyn tulos, joka toteutettiin 24.8–15.9.2023 välisenä aikana taukojen merkityksestä. Kyselyssä kartoitettiin 10 333 työntekijän näkemyksiä Yhdysvalloissa, Australiassa, Ranskassa, Saksassa, Japanissa ja Isossa-Britanniassa. Kysely kohdistettiin toimistotyöntekijöille, jotka työskentelivät kokopäiväisesti (Slack Workforce Index 2023).

Kuva 3. Kyselyn tulos (Slack Workforce Index 2023)

3.2 Seurantaviikko 2

Maanantai 13.1.2025

Tavoitteenani on hienosäätää kielivalikkoa, koska huomasin viikonloppuna, että tietokannassa oli muutamia kieliä, jotka puuttuivat tästä. Haluan korjauksen jälkeen myös testata ja varmistaa, että kielen valittua myös näytettävät käännökset muuttuvat käyttäjän valinnan mukaan.

Sovin palaverin IT-mentorin kanssa huomenna tiistaille seuraavaa työtehtävää varten. Kielten lisääminen kielivalikkoon osoittautui yllättävän helpoksi, ja samalla tutkin MongoDB:ssä kielten rakennetta. Kielten rakenne koostui muun muassa maan koodin ISO 3166 -standardin mukaan, joka on kansainvälinen standardi maiden tunnistamiseen ja koodaukseen. Tämän tietokentän lisäksi kielten nimi sekä kielen nimi omalla kielellä olivat osa rakennetta.

Syvensin myös osaamistani Pandas-kirjastosta käyttämällä W3Schools-materiaaleja. Materiaaleissa oli selkeitä esimerkkejä, harjoituksia ja myös pieniä tietovisoja joka kappaleen lopussa, jonka avulla pystyy tarkistamaan kappaleen opettamat asiat. W3Schools-materiaali on ollut erittäin hyödyllinen eri teknologioiden ja kirjastojen oppimisessa ja auttanut syventämään osaamistani ylipäätänsä myös Python-koodikielestä.

Tiistai 14.1.2025

Tavoitteeni on tutustua Vercel-pilvipalvelualustaan, sillä keskustelimme ennen palaveria IT-mentorin kanssa alustasta, joka tarjoaa eri työkaluja ja palveluita demoja ja jatkuvaa integrointia varten (CI/CD). Vercel tarjoaa kehittäjille palveluita ja pohjia, jotka helpottavat web-pohjaisten sovelluksien toteutusta, ylläpitoa ja julkaisua (Ahmed 2023). Vercel on myös Next.js kehittäjä, ja koska käytämme Next.js:ää sovelluksessamme, uskon että yhteensopivuus on erinomainen ja sovelluksemme on helppo integroida Vercel-alustaa.

Palaverin jälkeen sain kaksi tehtävää tälle ja tulevalle viikolle. Tehtävääni kuuluu kaikkien null-datan muuttaminen ”coming soon”-tekstiksi sivuston kielillä. Sen lisäksi käännösten haussa pääkategorioiden kautta ilmeni puuttuvia tietoja. Näihin puuttuviin tietoihin pitäisi lisätä esimerkiksi ”testi” prioriteetti kielillä: suomeksi, englanniksi, venäjäksi ja arabiaksi. Tämän kautta pystytään parantamaan käyttäjäkokemusta, sillä käyttäjälle näytetään enemmän dataa ja käyttäjät saavat selkeää tietoa valitulla kielellään. Kirjoitin heti aluksi itselleni ylös, että mistä kategorioista puuttui millä kielellä käännöksiä kokonaan. Eniten puuttuvia käännöksiä pääkategorioista ilmeni arabian kielellä ja venäjän kielellä, joten keskityin niihin aluksi.

Keskiviikko 15.1.2025

Tavoitteeni on ensin korvata null-data ”tulossa pian” -tekstiksi jokaisella kielellä. Uskon tehtävän olevan melko nopea, joten todennäköisesti ehdin myös aloittaa datan täydennystä jokaisen kategorian alle.

Aloitin päivän käyttämällä MongoDB-kyselyä, jossa suodatin kaikki null-tietotyyppiset datat sekä tietueet tietokannasta, joilla oli puuttuvia käännöksiä. Tämän avulla pystyn tarkistamaan koodini toimivuuden. Pystyin päivittämään kaikki tietokannan tiedot käyttämällä MongoDB:n update_many tai update_one komentoa insert-lausekkeen sijasta. Testasin ratkaisun toimivuuden useaan otteeseen paikallisesti ennen sen päivitystä DocToDocin tietokantaan. Sain päivitettyä datat ilman mitään virheitä, joten seuraavaksi keskityin pääkategorioiden täydentämiseen. Datan täydentäminen oli erittäin tärkeää, sillä huomasin sovelluksessa olevan monta kategoriayhdistelmää, joista ei löytynyt yhtään käännöstä. Tämä vaikuttaa negatiivisesti demoon, koska se antaa vaikutelman datan puutteellisuudesta, mikä voi luoda epäluotettavuuden tunnetta ja heikkoa käytettävyyttä.

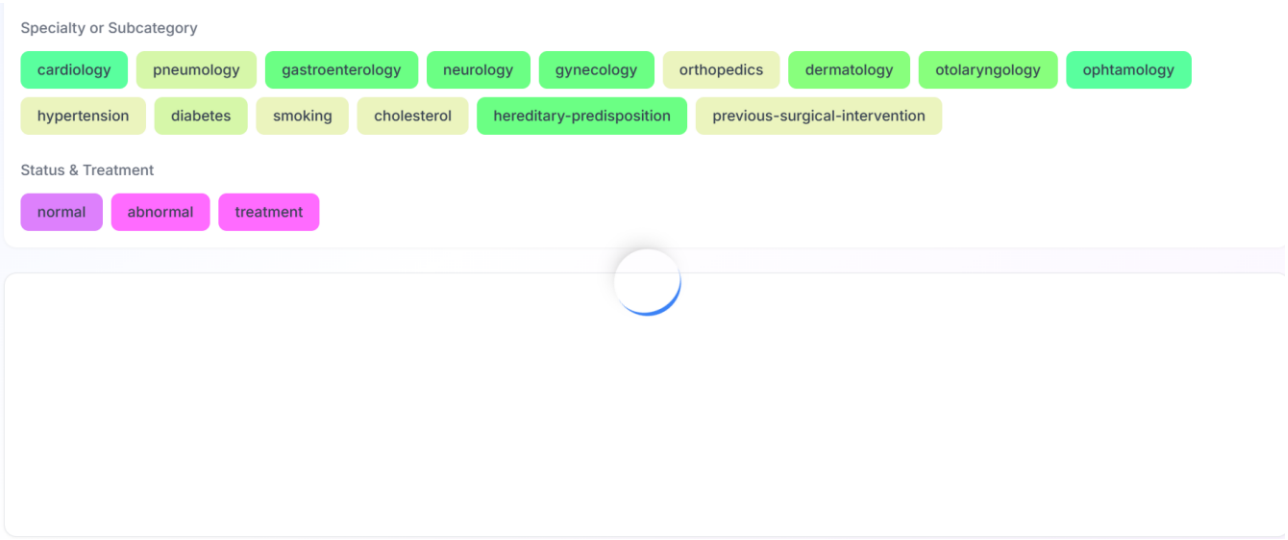
Otin myös pieneksi lisätehtäväkseni latausanimaation tekemisen. Latausanimaatiolla tarkoitetaan graafista elementtiä, joka osoittaa, että tietoja haetaan taustalla. Tällä pienellä animaatiolla voidaan parantaa käyttäjäkokemusta, sillä sen avulla käyttäjälle annetaan eräänlainen visuaalinen merkki, että sovellus on aktiivinen ja tietoja haetaan taustalla. (Kernaghan 2023)

Torstai 16.1.2025

Tavoitteenani on jatkaa datan täydennystä eri kategorioihin ja samalla myös toteuttaa valmiiksi mainitsemani latausanimaatio. Latausanimaatiota luodessa on tärkeä pitää mielessä sen sopivuus muiden elementtien kanssa, jotta kokonaisuus pysyy yhtenäisenä.

Minulla oli toimintavapaus tyylin suhteen, joten yritin luoda sopivanlaisen pyörän, ottaen huomioon sivuston taustan ja elementit. Sain toteutettua pyörän, jolle muotoilin tyylejä käyttämällä Tailwind CSS-tyylikirjastoa, josta löytyi muun muassa animate-spin toiminto. Tyylejä oli paljon erilaisia ja halusin luoda hieman erottuvamman version. Sen lisäksi, että se pyörii yksinkertaisella renkaalla, lisäsin siihen varjostusta ja tein sen koosta terävämmän. Käytin myös aikaa Tailwind CSS-materiaalien perehtymiseen ja latasin koodieditorilleni laajennuksen, jotta se pystyy ehdottamaan ja näyttämään tyylejä, mutta myös auttamaan virheiden havaitsemisessa tyylikoodissa. Alla on kuva (kuva 4.) tästä latausanimaatiosta, jonka toteutin. Latausanimaation näyttämiseksi käytin setTimeout-funktiota, jossa pystyin pidentämään latausaikaa määrittämällä viiveen millisekunteina. Tämä siis mahdollistaa animaation näkyvyyden pidempään käyttäjälle, vaikka datat olisikin haettu nopeammin. Normaalisti datan hakemisessa kestää noin 1–2 sekuntia, joten latausanimaatio on tärkeä lisä

elementti. Sen lisäksi se kertoo käyttäjälle, että sivusto ei ole jäätynyt ja antaa visuaalisen palautteen datan hakemisen etenemisestä.



Kuva 4. käännöksien hakemisessa näytetään lautasanimaatio ruudun keskellä.

Perjantai 17.1.2025

Tavoitteenani on viimeistellä datan täydennys ja luoda yhdistämispyyntö (pull request) latausanimaatiosta. Mahdollisesti aion löytää keinoja nopeuttaa datan täydennystä, jotta saan sen kokonaan valmiiksi tällä viikolla ja varmistan myös testaamalla, että kaikissa kategorioissa on käännöksiä.

Viimeistelin tietokanta täydennystä, jotta jokaisen kategorian sisällä olisi vähintään yksi käännös. Nämä käännökset olivat vain testi nimisiä tai joissa luki, että käännös oli tulossa pian, eli väliaikaisia ratkaisuja. Keksin lopulta tavan nopeuttaa tätä käännösten lisäysprosessia, minkä ansiosta sain työn valmiiksi iltapäivällä. Lisäsin ensin käännöksiä yhdellä kielellä kaikkiin kategorioihin, jonka jälkeen pystyn MongoDB-kyselyn (query) avulla lisäämään muiden kielten käännökset kerralla, suodattamalla vain kategoriat, joilla oli vain yksi käännös. Tein yhdistämispyynnön, jossa pyysin muutosteni tarkistamista ja yhdistämistä main-haaraan, sekä mentorilta palautetta. Yhdistämispyynnön aikana huomasin aiheutuneen pieniä ongelmia yhdistämisprosessin aikana. Ongelmana oli, että main-haara ei ollut ajan tasalla, mikä aiheutti konflikteja sen aikana. Lopulta poistin yhdistämispyynnön kokonaan ja tein sen uusiksi oikeilla versioilla.

Käytin myös aikaa tutustumiseen DocToDocin Vercel-alustaan, josta näkyi muun muassa suorituskyky ja aikaisemmat käyttöönotot (deployments), joita Vercel pystyi ottamaan suoraan Github-julkaisuista. Käytössämme oli Vercel Pro, joka mahdollistaa useiden tiimin jäsenten lisäämisen projektiin ja myös enemmän resursseja erityisesti tiedonsiirrossa

Seurantaviikkoanalyysi 2

Toisella viikolla tehtäväni olivat suurimmaksi osaksi data-analysoinnin parissa. Tiimimme ryhmässä tiedotettiin myös, että tulevalle torstaille pitäisi saada demoversio sovelluksesta valmiiksi. Tämän vuoksi keskityimme, että jokainen tiimin jäsen sai kriittisimmät tehtävät ajoissa valmiiksi. Viikonlopun aikana vedin main-haaraan kaikki uudet päivitykset, sillä perjantaina tehtiin paljon muutoksia. IT-mentorini antoi myös uuden tehtävän ennen torstain demoversiota. Tein viikon aikana suunnitelman tehtävästä, jossa minun piti luoda sivu, jonka sisällä käyttäjän merkitsemät suosikkikäännökset näkyvät. Suunnittelin muun muassa ajankäyttöä, ja kuinka paljon aikaa on käytettävissä ennen torstaita. Harkitsin vielä, ehdinkö luoda sivun kovakoodatuilla datoilla vai oikeilla käännöksillä.

Ajanhallinta toisella viikolla oli helpompaa kuin ensimmäisellä viikolla, sillä tehtäviä oli vähemmän ja aikaa kului enemmän itseopiskeluun. Kiinnitin myös huomiota työskentely ympäristööni kotona, jotta se olisi mahdollisimman ideaalinen ja tuottavuutta tukeva. Barbier (2022, 21) mainitsee, että etätyöskentelyssä on tärkeää löytää työskentelylle omistettu tila, olipa se kotona tai erillinen etätyötila, joka auttaa myös työn ja vapaa-ajan erottamista. Vaikka työskentelyni tapahtuu usein yksin kotona, niin haluan kokeilla työtiloja esimerkiksi kirjastoissa ja nähdä miten se vaikuttaa työskentelyyni ulkoisessa ympäristössä. Uskon, että ulkoisessa ympäristössä työskenteleminen tuo varmasti virkistystä, mikä voi tuoda uusia näkökulmia työskentelyyn ja nostaa motivaatiota, kun on muiden ympäröimänä.

MongoDB:n CRUD-toimintoihin perehtymisen ansiosta sain päivitettyä datan viikon aikana melko nopeasti, mikä jätti enemmän aikaa itseopiskeluun. Käytin myös aikaa projektin rakenteen tutustumiseen syvällisemmin varsinkin .next kansion sisälle, joka on automaattisesti luotu rakennusprosessin aikana. Vaikka kansion sisältö tuntui melko sekavalta, sain kuitenkin yleiskäsityksen, että se sisältää tietoa muun muassa rakennusprosessista, välimuistitiedosta ja reitityksestä. Yleisesti ei tarvitse muuttaa kansion tietoja (Shah 2024). Tutustuin myös Husky-työkaluun, joka oli projektissa valmiina, kun tulin mukaan. Husky-työkalu varmistaa Git commit-viestien pätevyyden ja auttaa ylläpitämään koodin laatua testaamalla (Joseph 2023). Husky muun muassa varmistaa, etteivät esimerkiksi arkaluontoiset tiedostot kuten .env tiedostot, pääse julkisesti GitHubiin.

3.3 Seurantaviikko 3

Maanantai 20.1.2025

Päivän tavoitteena on luoda suosikit-sivu, joka perustuu aluksi kovakoodattuun dataan. On myös tärkeä pitää sivuston ulkoasu yhtenäisenä muun muassa taustan ja aiemman luodun profiilit-sivun kanssa. Tyylien avulla on myös varmistettava, että teksti pysyy helposti luettavana ja, että sivusto on jatkokehitettävissä, varsinkin kun kyseessä on kovakoodatulla datalla luotu sivu. Sivun luomisessa pystyn soveltamaan osaamistani TypeScriptillä ja lisäämään tyylejä Tailwind CSS-kirjastoa käyttäen.

Pääsin päivän tavoitteisiin luomalla onnistuneesti suosikit-sivun, joka sisälsi kovakoodattua dataa ja oli tyyliltään yhtenäinen profiilit-sivun kanssa. Navigoinnin toteutus oli myös hyvin yksinkertainen, mistä nappia painamalla käyttäjälle aukeaa näkymä, jossa kaikki suosikkikäännökset ovat. Käännöksen poistotoiminnossa ilmeni ongelmia, sillä haluamani toimintoni tarkoitus oli poistaa käännös vain suosikeista, eikä koko tietokannasta. Testauksen jälkeen sain kuitenkin pulman ratkottua ja julkaisin illalla valmiin osion GitHubiin omaan haaraan.

Tiistai 21.1.2025

Tavoitteenani tälle päivälle on enemmän ajanhallintani suunnittelemista ja eilisen GitHub haaran yhdistäminen main-haaraan. Päätin myös lopulta jättää suosikit-sivun datat sellaisenaan torstain demolle, jonka jälkeen minulla olisi enemmän aikaa tehdä jatkoratkaisu sille.

Varmistin tyylien yhtenäisyyden sekä poiston toimivuuden ja tein yhdistämispyynnön main-haaraan, jossa pyysin myös palautetta IT-mentorilta. Suunnittelin omaa aikatauluani työn suhteen yksityiskohtaisemmaksi, että milloin pidän lyhyet tauot näiden kolmen ison osan sisällä. Päätin ajoittaa tasaisesti 15 minuutin tauot, jonka avulla yritän virkistäytyä ja levähtää ennen kuin jatkan tehtävääni. Koen kuitenkin, että lyhyiden taukojen aikana helposti jää katsomaan vain puhelimen ruutua, mikä ei välttämättä ole ideaalista, jonka takia olen yrittänyt löytää muita keinoja käyttää nämä lyhyet tauot hyödyllisemmin.

Keskiviikko 22.1.2025

Aloitin päivän palaverilla, jonka aikana minulle jaettiin kaksi pienempää tehtävää. Tavoitteenani on saada ne valmiiksi samana päivänä, jotta ehdin vielä lisätä ne demoversioon huomiseksi. Ensimmäinen tehtävä on sivuston logon vaihtaminen uuteen versioon ja toinen on luoda historia-sivu samalla tyylillä kuin aiempi suosikit-sivu. Logo oli valmiina kuvana, ja sen lisäyksessä mietin, että lisäänkö kuvan PNG-tiedostona vai SVG-tiedostona. Palaverissa jokainen kävi läpi omia tehtäviään

ja sen, että ehtiikö lisäämään oman osansa lopulliseen demoversioon. Kerroin saaneen suosikit-sivun valmiiksi, joka tarvitsi vain IT-mentorin tarkistuksen ja hyväksymisen GitHubissa. Palaverin aikana korostettiin testauksen merkitystä ja päätettiin lisätä GitHubin projektinhallintaan ”testing”-välilehti, josta näkee, mitkä tehtävät tarvitsevat vielä testausta. Usein helposti unohtuu, että vaikka koodi toimii omalla koneella moitteettomasti, se ei välttämättä toimi toivotulla tavalla toisella koneella. Sen takia on tärkeää, että testausta tehdään monipuolisesti eri laitteilla ja selaimilla, jotta voidaan havaita nämä ongelmat.

Sain logon lisättyä hyvin nopeasti projektin rakenteeseen, jossa toin sen aloitus- ja pääsivulle korvaten vanhan logon uudella. Päädyin tallentamaan logon PNG-tiedostona, sillä SVG-version kanssa ilmeni ongelmia sovelluksessa ja PNG oli yhteensopivampi Next.js:n image-työkalun kanssa. Opin logon lisäyksessä leikkaamaan taustan PNG-tiedostoista pois käyttämällä remove.bg-ohjelmaa ja tein siitä jälleen kerran yhdistämispyynnön main-haaraan.

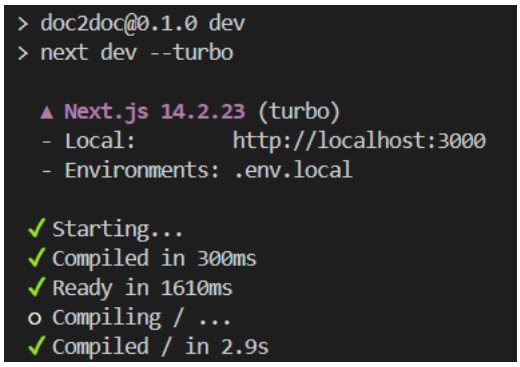
Tämän jälkeen tein toisen haaran historia-sivulle, jotta pystyn aloittamaan historia-sivun toteutusta samalla tavalla kuin suosikit-sivua. Sain sivun valmiiksi nopeasti, sillä se oli melkein identtinen suosikit-sivun kanssa, paitsi että pienet ikonit muuttuivat. Tämän muutoksen yhteydessä lisäsin reitityksen sivuston nimeen, jota klikkaamalla käyttäjä pääsee tarkastelemaan omaa profiiliaan. Tämän reitityksen koodin pystyin ottamaan suoraan aikaisemmista toteutuksista, josta ikonia klikkaamalla käyttäjä ohjataan profiili-sivulle.

Torstai 23.1.2025

Tavoitteenani on parantaa suorituskykyä, sillä olin huomannut viime aikoina projektin käynnistys localhost:3000 on ollut hyvin hidasta, ja joskus se ei käynnistynyt ollenkaan. Ratkon ja tunnistan hitaan käynnistyksen syyt, ja mahdollisesti löydän keinon sen parantamiseksi.

Aloitin tarkistamalla projektin riippuvuudet ja mahdolliset virheilmoitukset konsolissa sekä terminaalissa. Lisäksi tarkastelin koodin optimointiin eri mahdollisuuksia ja työkaluja, jotka voisivat parantaa projektin käynnistysnopeutta. Löysin Turbopack-työkalun, joka tarjoaa merkittävästi nopeampaa suorituskykyä kehityksen aikana ja suurissa projekteissa (Joodi 2024). Turbopackin käyttöönotto on myös hyvin helppoa, koska projektin package.json tiedostoon tarvitsi vain lisätä ”next dev --turbo”-skripti, jolloin käyttämällä normaalia käynnistystoimintoa turbo aktivoituu. Alla on kuvakaappauksia (kuva 5. ja kuva 6.) normaalin suorituskomennon ja turbon suorituskomennon eroja.

Kuva 5. normaalin "next dev" -skriptillä ajamisesta.



Kuva 6. "next dev --turbo" -skriptillä ajamisesta

Suurimpana erona turbossa on, että käynnistyksen suoritukseen kuluu millisekunteja verrattua normaaliin. Vaikka nämä sekunnit saattavat tuntua pieniltä eroilta, projektin kasvaessa nämä erot ilmenevät laajemmin, sillä suuret koodit ja useammat tiedostot tekevät rakennus- ja latausajoista pitkiä.

IT-mentori myös jakoi palautetta meille tiimin WhatsApp-kanavalle. Sovellustamme oli kokeiltu Lääkäripäivillä Messukeskuksessa, ja se sai sieltä erittäin positiivisen arvion. Tämä palaute rohkaisi erityisesti itseäni, että työn tulos on ollut merkittävä ja myös palkitsevaa. Oli inspiroivaa kuulla, että sovellus oli toiminut hyvin käytännössä ja saanut hyvää palautetta oikeilta käyttäjiltä.

Perjantai 24.1.2025

Sain tehtäväkseni kielivalikon uudistamisen, joka hakee kielet MongoDB:n languages-kokoelmasta. Tehtävä tuntui laajemmalta, joten kysyin neuvoa ja toteutukseen ohjeita IT-mentorilta. Sain myös selville, että nykyinen kielivalikko on kovakoodattuna projektiin, jonka takia sen joustavuus on ollut heikkoa ja kielen lisäys tietokannassa ei ollut integroitunut sivustoon. Tavoitteena on saada tehtävä suurimmaksi osaksi valmiiksi tänään ja viikonlopun jälkeen hioa visuaalisuutta, jonka jälkeen julkaisen valmiin komponentin maanantaina.

Vedin muutokset main-haarasta omaan haaraan, jotta saisin uusimman version haltuuni, kun työskentelen languages-haarassani, jonka juuri loin. Yritin luoda kielen hakemisen tietokannasta samalla tavalla, kuin kategorioiden ja käännöksien haun tietokannasta. Sain haettua kielet luomalla Next.js reitityksen avulla API-rajapinnan api/languages, joka palauttaa kielten tiedot JSON-muodossa front-endiä varten. Korvasin kovakoodatut kielet tällä API:n vastauksella, jonka jälkeen yritin parantaa valikon visuaalisuutta lisäämällä kielten lippuikoneita mukaan pudotusvalikkoon (dropdown). Lippujen ikonien lisäys tuotti paljon ongelmia, ja lopulta jouduin luomaan mukautetun pudotusvalikon. Tämä mukautettu pudotusvalikko mahdollisti lippujen ja kielennäyttämisen yhdessä valikossa, mikä ei ollut mahdollista aikaisemmassa valikossa. Lisäksi se paransi käyttökokemusta selkeällä visuaalisuudella.

Seurantaviikkoanalyysi 3

Viikon aikana oli paljon pienempiä tehtäviä, jotka olivat suurimmaksi osaksi front-endiin liittyviä. Näiden tehtävien avulla pystyttiin varmistamaan demon käyttäjäystävällisyys ja parantamaan visuaalisuutta ikoneilla ja logolla. Uskon, että vaikka kovakoodatut osat ovat vain väliaikaisia ratkaisuja, niillä oli merkittävä vaikutus asiakastyytyväisyydessä torstain demossa. Kovakoodaus, jota kutsutaan englanniksi nimellä hardcoding tarkoittaa siis tiettyjen tietojen upottamista suoraan sovellukseen sen sijaan, että ne olisivat helposti muokattavissa (Knell 2023). Esimerkkinä kovakoodauksessa oli viikolla tällä viikolla historia- ja suosikit-sivu, joissa luki vain ”testi” tai ”tulossa pian”, oikean tiedon sijaan. Tämä ratkaisu oli kuitenkin yksinkertainen ja nopea toteuttaa, varsinkin ajankäytön näkökulmasta, jossa priorisoitiin sivustojen nappien toimiminen ja niiden tietojen näyttämistä käyttäjälle.

Viikon suurimpana oppimisena oli Next.js API-reitityksen luonti, joka on vahvasti riippuvainen projektin kansiorakenteesta. API:n luonti tapahtuu siis kansioiden välillä, jossa esimerkiksi omassa tapauksessani kielten API oli muodostettu api-kansion alle languages-kansio, jonka sisällä on routes.ts. Tiedoston nimen on oltava route, jotta Next.js tunnistaa sen automaattisesti olevan osaksi reititysjärjestelmää (Jith 2024). Näiden kansioiden ja tiedostojen nimien perusteella muodostuu URL-polku api/languages, josta saadaan kielten tiedot tietokannasta. Testasin toteutuksen jälkeen varmistamalla koodin toimivuuden lisäämällä MongoDB:n tietokantaan uuden kielen. Lisäyksen jälkeen tarkistin, että kieli on näkyvillä api/languages URL-polussa ja sitä kautta myös sivuston pudotusvalikossa. Lopulta julkaisin viikonlopun aikana tehtäväni edistymisen omaan haaraan, jotta IT-mentorini pystyy myös tarkastella muutoksiani. Suunnittelin vielä viikonloppuna, miten pystyn hiomaan ratkaisuani ja refaktoroida koodiani. Erityisesti pohdin pudotusvalikon siirtämistä omaan tiedostoon, jotta koodin rakenne olisi helpommin ylläpidettävä ja pudotusvalikko mahdollisimman uudelleenkäytettävä.

Haluan myös kiinnittää enemmän huomiota jatkossa suorituskykyyn ja sen optimointiin varsinkin myöhemmässä vaiheessa, kun projekti suurenee. Tämä tarkoittaa esimerkiksi tutkimista, missä kohdissa koodia kuluu aikaa tietojen käsittelyyn ja mahdollisesti löytää tehokkaampia ratkaisuja näille esimerkiksi hyödyntämällä välimuistia. Välimuisti voi olla myös erittäin hyödyllinen suorituskyvyn parantamisessa, koska se tallentaa käytettyjä tietoja väliaikaisesti, mikä vähentää tarvetta tehdä pyyntöjä samoja tietoja varten uudelleen (Arora 2024).

3.4 Seurantaviikko 4

Maanantai 27.1.2025

Tavoitteenani on viimeistellä kielivalikko ja yhdistää viimeistelty versioni yhdessä main-haaraan. Huomasin myös tiedostoissani olevan paljon vanhaan kielivalikkoon liittyviä koodinpätkiä, joten koodista piti poistaa nämä ylimääräiset osiot, jotka saattavat sekoittaa jatkokehitystä. Haluan myös käyttää aikaa itseopiskeluun MERN-stackiin, jossa Subramanian (2019) käy läpi kirjassaan yksityiskohtaisesti, jokaista MERN-stackin komponenttia.

Siirsin kielten pudotusvalikon omaan komponenttiin, jonka jälkeen pystyin helposti tuoda sen tarvittaviin tiedostoihin muun muassa muokkaus- ja lisäysmodaaleihin, jossa kyseistä valikkoa tarvittiin. Tämä lyhensi koodia merkittävästi, ja poistin ylimääräisiä koodiosia tiedostoista. Tein iltapäivällä yhdistämispyynnön main-haaraan, jotta pystyisin yhdistämään muutokset ja pyysin IT-mentorilta arviointia ja jonkinlaista palautetta koodista.

Samalla perehdyin iltapäivällä MERN-stackiin, joka on teknologia kokonaisuus koostuen seuraavista teknologioista: MongoDB, Express.js, React.js ja Node.js. Näistä teknologioista huomaa JavaScriptin merkittävän roolin, sillä kaikki neljä perustuvat JavaScriptiin. Subramanian (2019) mainitsee, että aloittelijana ei kannata luoda suoraan kokonaan projektia MERNillä vaan tutustua myös muihin teknologiapaketteihin ja valita se, joka tuntuu tutuimmalta. Kertasin ja kävin läpi kirjan tehtäviä, joista löytyi lopussa vastaukset niihin. Tämä auttoi minua syventämään ymmärrystäni MERN-stackista ja sen komponenteista.

Tiistai 28.1.2025

Tavoitteenani on viime viikon demon jälkeen saada suosikit-sivu kokonaan valmiiksi, jotta sen toiminnot ovat valideja ja visualisuus miellyttävä käyttäjälle. Uskon tehtävän olevan hieman laajempi, joten tavoitteena on aloittaa työstämistä tänään ja pitää myös main-haara ajankohtaisena, kun IT-mentori mainitsi päivittävänsä sitä myös. Lähtökohtana koodissa on, että kovakoodatun datan sijasta on käytetty Math.random-funktiota satunnaisten käännöksien näyttämiseen sivulla.

Tämä piti poistaa sekä korvata oikealla logiikalla, ja kysyin IT-mentorilta, että miten kannattaisi säilyttää näitä käyttäjän merkitsemiä suosikkikäännöksiä. Pitäisikö esimerkiksi tietokantaan tehdä uusi erillinen kokoelma näille tai jotain muuta vastaavaa. Hän ehdotti, että helpompaa olisi luoda olemassa olevaan käyttäjät-kokoelmaan favorites-taulukko, josta pystyy erottelemaan jokaisen käyttäjän valitsemat suosikit.

Loin itselleni MongoDB:n Compass-työkalun avulla ensin omalle käyttäjälleni tyhjän favorites-taulukon. Käyttäjätietojen kanssa pitää olla hyvin varovainen, koska niiden muokkaaminen voi helposti vaikuttaa järjestelmän toimintaan ja pääsyyn. Sain myös yksilöityä käännöksien id-kentän, jonka avulla pystyn erottelemaan käännöksiä toisistaan.

Keskiviikko 29.1.2025

Tavoitteena tälle päivälle on saada käyttäjät-kokoelman sisällä olevan favorites-taulukon testidata näkyviin sivuston käyttäjäliittymässä, joka vaatii todennäköisesti jonkinlaisen API-rajapinnan luonnin ja hyödyntämisen, josta olen saanut jo aikaisemmin kokemusta. Sain myös palautetta aikaisemmasta kielivalikosta IT-mentorilta, joten päivän tavoitteen lisäksi tuli myös kielivalikon hiomista ja sen korjaamista.

Loin API-polun onnistuneesti, josta näkyy käyttäjän suosikkikäännökset tietokannassa. Samalla periaatteella kuin aikaisempi toteutus, tein polun api/user/favorites alle, josta näkyy tällä hetkellä käyttäjän oman suosikkikäännökset, jotka olen vain lisännyt tietokantaan favorites-taulukon sisälle. Julkaisin tämän edistyksen ensin GitHubiin, ja päätin jatkaa kielivalikon korjausta iltapäivällä.

Varmistin IT-mentorilta tarkemmin korjattavat asiat, jonka jälkeen sain selvyyttä korjattaviin asioihin. Loin jokaiselle kielelle active-tietokentän ja svgIcon-tietokentän. Active-tietokentän tarkoituksena oli suodattaa kieliä, joita oli enemmän kuin haluttu tietokannasta, että mitkä aktiiviset kielet ovat valittavissa sivustolla. SvgIcon-tietokentän tarkoitus oli sen sijaan näyttää käyttäjälle kyseisen lipun SVG-versio kuva. IT-mentori mainitsi, että lippukuvakkeen pystyi suoraan kopioida tietokantaan tekstimuodossa, jotka löytyivät node-paketista react-world-flags.

IT-mentori muistutti myös kiinnittämään huomiota datan siirtymiseen ja uudelleenrenderöintiin, jotka vaikuttavat sivuston suorituskykyyn suuresti. Sain vinkkejä ja myös opetusmateriaaleja, joita kertaamalla pystyn parantamaan sovelluksen tehokkuutta. Erityisesti opin tehokkaan datansiirron, kuten map-funktion käytön, joka mahdollistaa datan käsittelyn ilman suorituskyvyn heikentymistä. Tämä vähentää myös turhia uudelleenrenderöintejä, mikä nopeuttaa sivun latausaikaa ja parantaa käyttäjäkokemusta

Torstai 30.1.202 5

Päivän tavoitteena on saada valmiiksi tämä kielivalikko, josta löytyi korjattavia asioita. Aikataulutuksena oli, että saisin tälle päivälle tehtävän lähes valmiiksi ja tehdä perjantaina pienet hienosäädöt, jonka jälkeen julkaisen GitHubiin ja yhdistän main-haaraan muutokseni.

Löysin netistä node paketin React-svg, joka helpottaa SVG:n ja Reactin yhteensopivuutta. Tutustuin React-svg:n dokumentaatioihin, sen käyttöesimerkkeihin ja propseihin. Latasin paketin npm install-komennolla, josta sain tarvittavat riippuvuudet paketille. Kielivalikon korjauksessa oli myös tärkeää varmistaa ja testata jatkuvasti, että muut komponentit toimivat normaalisti uuden version kanssa. Varmistin koodin laadun poistamalla käyttämättömiä lausekkeita ja refaktoroinnilla. Kiinnitin huomiota myös virheiden hallintaan esimerkiksi datan hakemisessa tietokannasta. Lisäsin muun muassa try-catch-lohkoja API-kutsujen ympärille, jotta mahdolliset virheet pysyisivät hallinnassa. Sain edettyä kielivalikon siihen vaiheeseen, että julkaisin sen GitHubiin omaan haaraani ja pyysin IT-mentorilta arviointia koodista.

Perjantai 31.1.2025

Tarkastelin eilistä kielivalikon koodia, ja huomasin sen yhteydessä, että Admin-käyttäjän muokkaustoiminnossa oli epäjohdonmukaisuuksia. Osa käännöksistä päivittyi muokkauksia tehdessä, mutta jostain syystä uudempien käännöksien muokkaukset eivät näkyneet käyttöliittymässä. Tavoitteeni on ratkoa tämä ongelma ja selvittää sen syy, mistä se johtuu.

Konsolilokituksen avulla sain selville, että muokkauksien tallentamisvaiheessa ilmeni virheitä konsoliin, jonka takia muutokset eivät heijastuneet sivustoon. Konsolilokituksessa tuli ”Casterror”-niminen virheilmoitus, joka on itselleni jokseenkin tuttu virheilmoitus. Casterror tarkoittaa siis muunnoksen epäonnistumista, ja tässä tapauksessa tyhjän merkkijonon muuntaminen boolean-arvoksi tuotti tämän virheen, sillä boolean vastaanottaa arvoksi vain ”true” tai ”false” (Mongoose s.a.). Sain ratkottua tämän ongelman toteuttamalla koodiin varmistuksen, että tietokentät ovat oikeassa muodossa ja toteutin niille asianmukaisen käsittelyn.

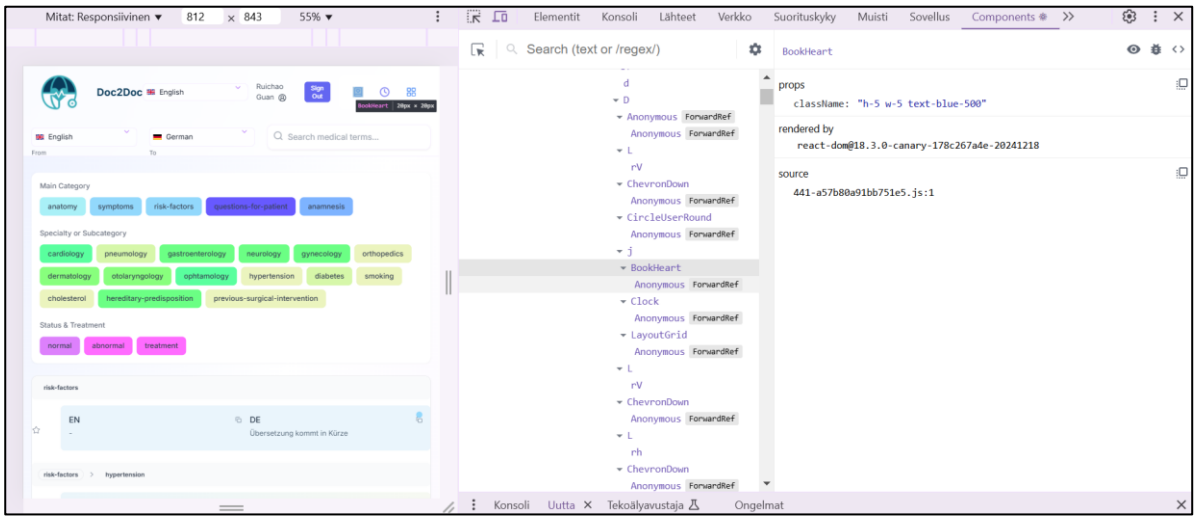
Ehdotin myös IT-mentorille, että tietokannassa oli pientä epäyhteensopivuutta tietokentän nimen kanssa, joten muokkasin myös tietokantaa Python-koodikielellä. IT-mentori painotti kokeilua ensin paikallisesti, jotta pystyin varmistamaan muutoksen toimivuuden ennen varsinaista sovelluksen tietokantaa. Sain luotua Pythonilla tarvittavan tietokentän nimen muutoksen, ja testasin paikallisesti monesti, että se toimi toivotulla tavalla. Ennen kuin päivitin datat varsinaisessa tietokannassa, varmuuskopioin nykyiset datat MongoDB:n export-komennolla JSON-tiedostoksi, jotta voin tarvittaessa palauttaa alkuperäiseen dataan, jos ongelmia ilmenee.

Seurantaviikkoanalyysi 4

Tällä viikolla tekemistä oli enemmän ja sain myös paljon palautetta sekä ehdotuksia IT-mentorilta, joiden korjaaminen vei suurimman osan ajan viikosta. Isoin haaste tällä viikolla oli kielivalikon korjaaminen. Kielivalikon korjaus vaati paljon muutoksia niin tietokantarakenteeseen kuin käyttöliittymään, joka vei paljon aikaa. Opin tärkeyden pilkkoa tehtäviä pienemmiksi, sillä suurempien kokonaisuuksien hahmottaminen helpottui ja pystyin keskittymään pienempiin osa-alueisiin kerrallaan.

Viikon suurimpiin oppimisen teemoihin kuului suorituskyvyn optimointi ja virheiden sekä ongelmien havaitseminen ja ratkaiseminen. Erityisesti suorituskyvyn parannuksessa koin, että lopputuloksen hahmotus oli hankalaa, sillä sovelluksen latausajat eivät olleet mitenkään liian suuria. Subramanian (2019, 79) mainitsee kirjassaan esimerkeillä, miten suorituskyvyn optimoinnissa suositellaan käyttämään funktiokomponentteja luokkakomponenttien sijasta. Syynä tähän on, että funktiokomponentit ovat helpommin luettavia ja vievät vähemmän tilaa luokkakomponentteihin verrattuna. Ayebola (2024) vertailee tarkemmin funktiokomponentteja ja luokkakomponentteja Reactissa ja pohtii niiden eroja ja käyttötarpeita. Luokkakomponenteilla on muun muassa pääsy eri elinkaari metodeihin sekä instanssimetodeihin, joita käytetään muun muassa komponentin datan hakemiseen ja tapahtumien tarkempaan käsittelyyn. Funktiokomponentti sen sijaan vastaanottaa arvoja (props), joiden avulla tiedon siirtymistä voidaan hallita ja edistää edellä mainittujen ominaisuuksien lisäksi.

Virheiden tunnistaminen ja korjaaminen vei jokseenkin paljon aikaa myös, sillä itse työkalut niihin olivat itselleni hieman vieraita. Tutustuin Subramanian (2019, 228) suosittelemaan React Development tools-laajennukseen, jossa pystyy tarkastelemaan React-komponentteja hierarkkisessa järjestyksessä. Sen lisäksi tämä työkalu on helposti käytettävissä, sillä latauksen jälkeen, kehittäjätyökaluihin ilmestyy React-välilehti, jonka avulla voi analysoida tätä komponenttipuuta, tarkastella komponenttien tilaa ja propseja, mutta ennen kaikkea tunnistaa virheet tehokkaammin. Alla on julkaistun sovelluksemme React Developer tools-näkymä (kuva 7). Käytin aikaa työkalun dokumentaatioihin perehtymiseen ja huomasin, että sama työkalu oli saataville mobiilisovelluksille React Native Devtools-nimellä. Työkalu on myös hyvin ajan tasalla, sillä viime päivitys oli Google Chrome kaupan mukaan 29.1.2025. Kokeilin tutoriaalia seuraamalla työkalun suurimpia toimintoja demon avulla.



Kuva 7. Julkaistun sovelluksen React Developer Tools-näkymä

Tällä viikolla korostui myös proaktiivisuus ja oma-aloitteisuus, kun tällä viikolla emme pitäneet tapaamista, vaan kaikki työskentelivät itsenäisesti. Monesti on käynyt, että olen joutunut odottamaan esimerkiksi palautetta, joten yleensä testailen sivustoa löytääkseni virheitä ja parannuskeinoja. Samalla pyrin arvioimaan käyttäjäkokemusta käyttäjän näkökulmasta: onko sivuston ulkoasu miellyttävä, navigointi selkeä ja onko sivusto kokonaisuudessa responsiivinen sekä toimiva eri selaimilla ilman häiriöitä.

3.5 Seurantaviikko 5

Maanantai 3.2.2025

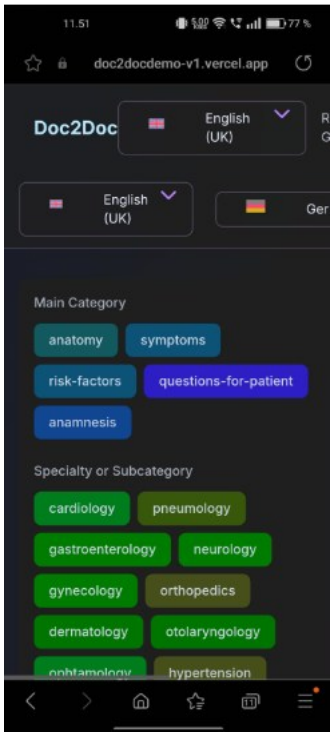
Palaan kielivalikon korjauksen ja viimeistelyn jälkeen takaisin suosikit-sivun työstämiseen. Sain myös iltapäivällä IT-mentorilta tehtäväkseni tuoda kaikki olemassa olevat käännökset englanniksi, suomeksi ja italiaksi helposti luettavaan muotoon, jotta lääkäri pystyisi tarkistamaan niiden oikean muodon ja varmistamaan, että kaikki tiedot ovat oikein ja ymmärrettäviä. Tavoitteeni tälle päivälle on muuntaa MongoDB:n suomi-, englanti- ja italiankieliset käännökset CSV-muotoon ja siitä Exceliin, jonka pystyn lähettämään sähköpostilla lääkärille tarkistettavaksi. Loput ajastani aion käyttää suosikit-sivuun ja mahdollisesti myös testaamaan sivustoa puhelimellani tarkistaakseni responsiivisuutta ja näkymiä.

Kertasin, mihin vaiheeseen jäin viime viikon alussa ja yhdistin ajankohtaisimman main-haaran omaan haaraani, jotta pystyn välttymään mahdollisista yhdistämiskonflikteista. Muutin lopulta favorites-taulukon logiikkaa siten, että se tallentaa vain käännösten id-tietokentät. Tämän avulla taulu-

kon sisällä on pelkästään id-dataa, mikä tekee tietorakenteesta helpommin luettavamman. Sen jälkeen, kun varmistin id-kentän olevan validi, toteutin API logiikan, josta haetaan kyseisen id:n käännökset ja jotka näytetään käyttäjälle.

Loin MongoDB Compass-työkalulla kyselyn, joka suodattaa käännökset vain englanniksi, suomeksi ja italiaksi. Tämän avulla pystyin tuomaan CSV-version kaikista käännöksistä vain näistä kolmesta kielistä. Opin muuntamaan CSV-tiedostoista Excel-tiedostoja ja tallentamaan ne tarvittaessa eri formaateissa kuten JSON, PDF, XML:inä.

Kokeilin myös illalla sovellusta puhelimella, että miltä se näyttää ja huomasin sen responsiivisuudessa olevan parannettavaa. Alhaalla on kuva (kuva 8.) miltä sovellus näytti puhelimellani.



Kuva 8. Puhelimella näkymä sovelluksesta.

Kuvasta 8 näkyy, että vain kategoriaosio on säilynyt responsiivisena, mutta navigointipalkki ja käännökset näkyivät osittain leikattuina. Tämä viittaa siihen, että sivun tyyleissä on ollut haasteita, mutta uskon, että on helposti korjattavissa Tailwind CSS:n avulla. Sen lisäksi voidaan ottaa mallia kategoriaosiosta ja soveltaa sitä kautta muihin komponentteihin

Tiistai 4.2.2025

Päivän tavoitteena on jatkaa suosikit-sivua ja myös perehtyä syvemmin Tailwind CSS:n dokumentaatioon, jonka avulla pystyisin korjata sivuston responsiivisuutta. Valitsin myös seuraavan tehtävän, jonka tulen todennäköisesti aloittamaan ensi viikolla, koska arvelen saavani suosikit-sivun kokonaan valmiiksi tällä viikolla. Valitsin seuraavaksi tehtäväkseni luoda statistiikkanäkymän käyttäjälle, jossa muun muassa kerrotaan, kuinka monta kertaa käyttäjä on kirjautunut sivustolle, kuinka kauan käyttäjä oli sivustolla ja kuinka monta hakua käyttäjä on tehnyt. Uskon, että näiden tilastojen keräämiseen tarvitaan jonkinlainen event listener ja event handler-funktio, joka kuuntelee ja seuraa käyttäjän tekemiä muutoksia, kuten nappien painalluksia ja reagoi niihin päivittämällä tässä tapauksessa tilastotietoja.

Jatkoin eilisen koodin korjausta, josta huomasin id kentän menneen sekaisin MongoDB:n _id:n kanssa. Päätin vaihtaa id-kentän tallentamisen sijasta käännöksen code-kentän, joka on myös yksilöivä ja uniikki, luotu hajautuksen avulla. Sain näytettyä oikeat käännökset sivulla ja integroitua painikkeen yhteensopivuuden, jossa tähti-ikonia painamalla kyseinen käännös tallentuu tietokantaan, ja tämän kautta sain näytettyä sen käyttäjälle suosikit-sivulla. Mainitsin IT-mentorille edistymiseni ja, että sivu oli melkein valmis. Sivusto vaati vielä pientä hienosäätöä tyylien kanssa, jotta näkymä olisi samanlainen kuin muissa. Julkaisin edistykseni omaan haaraan ja samalla toin uusimmat muutokset main-haarasta yhdistämällä sen omaan haaraani. Haaran yhdistys oli sulava ja vältyin konflikteilta. Tämän avulla voin edetä omassa haarassani ajantasaisella ympäristöllä, joka sisältää myös muiden tiimin jäsenten muutokset.

Käytin iltapäivän responsiivisuuden tutkimiseen Tailwind CSS-tyylikirjastolla. Chauhan (2024) kertoo Tailwind CSS:n oletus katkaisupisteet (breakpoints) olevan seuraavat: sm (640px), md (768px), lg (1024px), xl (1280px) ja 2xl (1536px). Nämä katkaisupisteet korvaavat perinteisen CSS:n media queryt, ja näitä voidaan yhdistää eri luokkien kanssa responssiivisen kokonaisuuden toteuttamiseksi.

Keskiviikko 5.2.2025

Päivän tavoitteeksi asetin itselleni suosikit-sivun viimeistelyn ja todennäköisesti julkaisen valmiin version huomenna GitHubiin. Ennen GitHubiin julkaisua haluan myös varmistaa, että kaikki olemassa olevat toiminnot ovat vielä ehjiä ja toimivat samalla tavalla kuin aikaisemmin. Aion myös iltapäivällä käyttää aikaa itseopiskeluun komponenttien suunnittelusta ja niiden välisestä kommunikaatiosta

Sain suosikit-sivun lähes valmiiksi, jossa loin tyylin yhtenäiseksi muiden sivujen kanssa. Jatkan huomenna datan siirtymisen hienosäädössä suosikit-sivulla, jossa datan uudelleenhaku on keskeisessä roolissa. Uudelleen haun tarkoitus on pitää data synkronisena, jossa käännöksen merkkaus suosikkeihin näkyy käyttäjälle ilman viivettä. Ongelmana oli näyttää käyttäjälle mitkä käännökset oli valmiiksi merkitty suosikiksi, jolloin tähti-ikoni näyttäytyy hieman erilaisena. Uskon, että ongelma sijaitsee toggle-funktiossa. Yritin löytää virheen, joka vaikutti virheelliseen toimintaan.

Jatkoin loppupäivän itseopiskelua erityisesti komponenttien suunnittelusta. Subramanian (2019, 81–83) käy läpi tilan (state) ja ominaisuuksien (props) erot, komponenttien hierarkiaa ja komponenttien välistä kommunikaatiota. Isona erona tilan ja ominaisuuksien välillä on muun muassa, että ominaisuudet ovat muuttumattomia ja komponentti voi vain lukea ominaisuuksia. Sen sijaan tilaa pystyy muuttamaan this.setState() -metodilla, jonka avulla pystytään päivittämään komponentin tilaa. Subramanian (2019, 82) muistuttaa, että komponentin on noudatettava Single Responsibility-periaatetta, joka tarkoittaa, että jokainen komponentti on vastuussa vain yhdestä asiasta. Tämän periaatteen avulla komponentti on selkeä ja helpommin ymmärrettävä, jonka avulla esimerkiksi virheiden hallinta on myös helpompaa.

Torstai 6.2.2025

Päivän tavoitteena on selventää koodia jakamalla erilliseen tiedostoon ja saada tietovirtaa synkronisemmaksi. Projektissa oli valmiiksi konteksti kansio, joten tarkastelin ensin aikaisempia kontekstin käyttöjä muissa tiedostoissa. Aion myös itseopiskella enemmän synkronisuudesta ja siihen liittyvistä työkaluista, kuten React Contextista ja async-funktioista.

Pääsin päivän tavoitteisiin kohdistamalla suosikkien API-kutsut, tilat ja ominaisuudet omaan kontekstitiedostoon, jotta voin hallita näitä kaikkia samassa tiedostossa. Kontekstin käyttö mahdollisti myös synkronisuuden, jonka kanssa oli pieniä ongelmia aiemmin. Esimerkiksi data ei päivittynyt heti suosikit-sivulla, vaan sivu piti sulkea ja avata uudelleen, jotta näki uudet suosikit. Aiemmin tämä oli haasteellista, koska suosikkien tila ei aina päivittynyt oikein, mikä johti virheelliseen toimintaan. Siirsin kontekstitiedostooni myös async/await-käsittelyn, joka varmistaa tilan päivityksen vasta sen jälkeen, kun palvelimen vastaus on saatu API:sta. Async/await vähentää myös tarpeettomia uudelleenrenderöintejä ja varmistaa, että käyttäjän tekemät muutokset näkyvät välittömästi käyttöliittymässä datan hakemisen jälkeen.

Perjantai 7.2.2025

Päivän tavoitteena on suunnitella seuraavan viikon tehtävää. Todennäköisesti on suunniteltava tietokannan rakennetta, jotta käyttäjän eri toimintaa pystytään seurata ja tallentaa näytettäviin tilastoihin. Lähetin kuvan kokonaan valmiista suosikit-sivusta IT-mentorille, jonka jälkeen tein yhdistämispyynnön main-haaraan.

Käyttäjän aktiviteetin näyttäminen tilaston suhteen IT-mentori mainitsi, että pitää pohtia, mitä kaikkia tietoja näytetään käyttäjälle ja mitkä muut analyyttiset tiedot näytetään vain yrityksellemme. Tämä tarkoittaa, että on tärkeää tehdä selvä ero näitä näkymiä tehdessä. On myös olennaista visualisoida, miltä lopputulos näyttää ja minkälaisena tilastot esitetään käyttäjälle. Huomasin profiilitsivun sisällä olevan kovakoodattuja arvoja, joka sisälsi muun muassa tiedot siitä, milloin käyttäjä oli viimeksi kirjautunut sisälle, suosikkikäännösten määrän ja montako käännöstä on hakenut. Uskon, että nämä tulevat tilastot halutaan näyttää käyttäjän profiilisivulla, mutta on myös pohdittava tilan määrää. Esimerkiksi tilastoissa pylväät vievät paljon tilaa sivulta, joka ei välttämättä ole paras vaihtoehto käyttäjäkokemuksen kannalta. Tilankäytön vuoksi mietin, olisiko järkevämpää siirtää tilastot kokonaan erilliselle sivulle, jolloin käytettävissä on enemmän tilaa ja sivu keskittyisi pelkästään tilastoihin ja analytiikkadataan.

Seurantaviikkoanalyysi 5

Viikko sujui sulavasti suosikit-sivustoa työstäen. Kävin kirjastossa muutaman kerran viikon aikana vaihtelun vuoksi kotityöympäristöön, jossa pystyin itseopiskella teknologioita ja edistämään projektia. Sain koko suosikit-sivun valmiiksi tällä viikolla, joka oli kokonaistavoitteeni ja, jotta pystyn siirtyä seuraavaan tehtävään uuden viikon alkaessa. Ajanhallinnan suunnittelu kotona ja kirjastossa auttoi minua saavuttamaan pienempiä tavoitteitani viikon aikana, ja olin hyvin tarkka ajankäytön suhteen itseopiskelun ja työn välillä. Erityisesti uusien konseptien ja työkalujen opettelu vaati aikaa sekä keskittymistä, joten yritin tasapainottaa näitä kahta, jotta pystyisin syventymään uusiin aiheisiin ilman, että työtehtävän edistyminen hidastuisi merkittävästi. Kirjasto työympäristönä motivoi ja auttoi keskittymään suuresti, jonka avulla pystyin hallitsemaan ajankäyttöä paremmin kuin kotona. Rauhallinen ja hiljainen ilmapiiri sekä vähäiset häiriötekijät paransivat tehokkuuttani, mikä edisti minua keskittymään syvällisemmin tehtäviin. Kirjastossa käynti auttoi myös itseäni erottamaan vapaa-aika työajasta, jonka kanssa aiemmin oli ollut ongelmia. Alhaalla (kuva 9.) on Shrestha (2008, 101) kenttätutkimuksen tulos, jossa kysyttiin Iso-Britannian oppilaiden motiivia kirjastoissa käymisessä.

Opiskelijoiden kirjastossa käynnin syyt

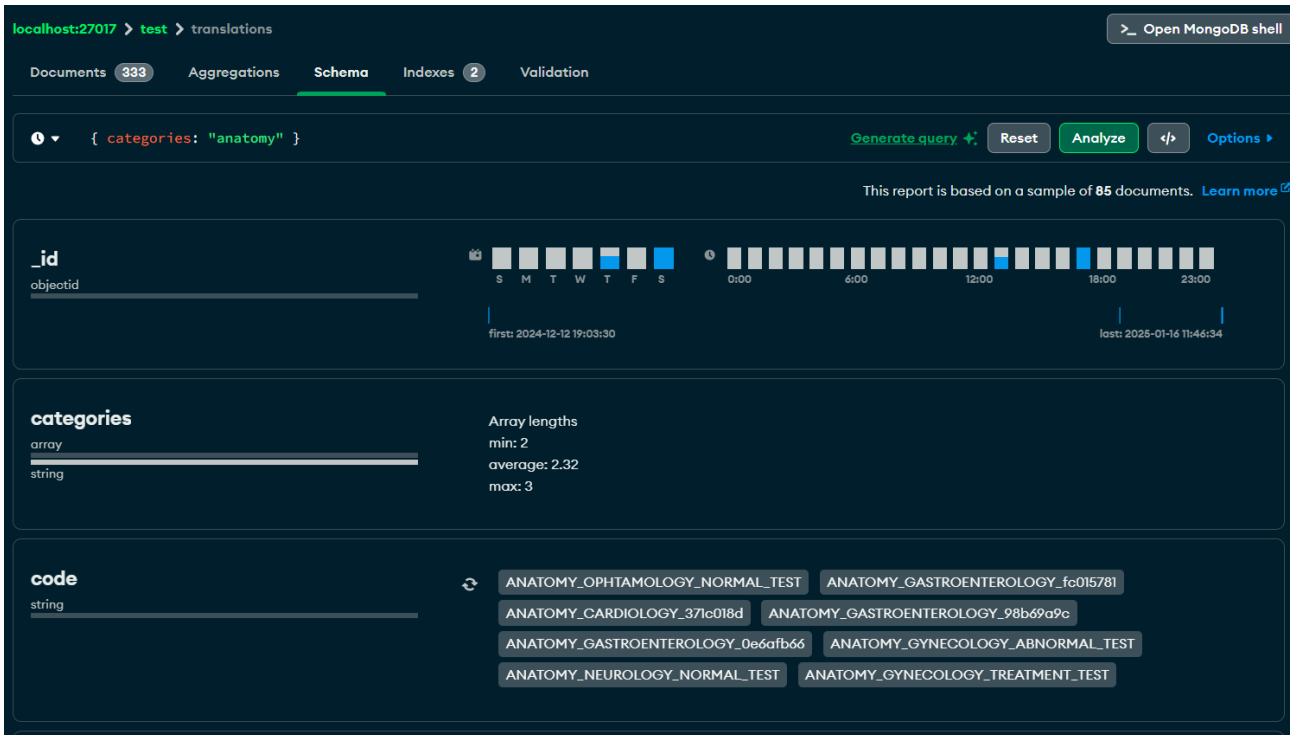
Oppilaitoksen nimi	Rauhallinen ympäristö		Opiskelum ateriaalin saatavuus		Ei mitään erityistä		Yhteensä	
	Lkm	%	Lkm	%	Lkm	%	Lkm	%
K. U.	14	41	18	53	2	8	34	100
T.C	8	19	27	66	6	15	41	100
NCCS	10	20	26	30	16	30	52	100
Yhteensä	32	25	71	56	24	19	127	100

Lähde: Field Survey

Kuva 9. Shresthan laatima kenttätutkimuksen tulos vuodelta 2008.

Tulosten mukaan suurin osa opiskelijoista käy kirjastossa opiskelumateriaalien saatavuuden takia, mutta myös sen rauhallisen ympäristön vuoksi. Lisäksi tutkimuksen kolmantena kolumnina ilmeni, että osa opiskelijoista ei kokenut erityistä syytä kirjastossa käymiselle vaan todennäköisesti käyttää sitä muun muassa satunnaisesti tai tapaamispaikkana muiden aktiviteettien ohella (Shrestha 2008).

Huomasin, että kirjaston julkista Wi-Fi-verkkoa käyttäessä projektini ei käynnistynyt localhostissa, johtuen todennäköisesti julkisen Wi-Fi-verkon palomuurista, joka rajoitti yhteyksiä ja esti mahdollisesti portteja, joita localhost tarvitsi koodin ajamiseen. Localhost on tietokoneen oma verkko-osoite (127.0.0.1), jonka avulla se voi kommunikoida itsensä kanssa (Chris 2022). Localhost on avain ohjelmistokehityksessä, sillä sen avulla voidaan testata ja etsiä virheitä sovelluksessa turvallisesti ilman internet-yhteyttä tai ulkoista palvelinta. Sen lisäksi localhost on muun muassa mahdollistanut MongoDB:n muutosten ajamisen paikallisesti ennen kuin olin tehnyt muutoksia varsinaiseen tietokantaan, mikä on vähentänyt virheiden riskiä ja parantanut päivitysten integrointia. Olin kokeillut paikallisesti MongoDB:ssä datan tuontia, skeemamuutoksia ja indeksien optimointia, jotta tietokannan suorituskyky pysyisi mahdollisimman tehokkaana. Tutustuin viikonaikana MongoDB Compass-työkaluun syvemmin, josta huomasin pystyvän analysoivan indeksejä, tietokenttiä ja tietokannan rakenteita tarkemmin. Alhaalla (kuva 10.) on demonstroitu, miten schema-välilehdestä voi tarkastella muun muassa otettua datanäytettä, jossa esitetään keskiarvoja ja tietokenttien luontiajankohtia.



Kuva 10. MongoDB Compass -näkymä paikallisen tietokannan rakenteen analysoinnista.

Kokeilin myös Mongosh-työkalua, joka on komentorivityökalu MongoDB:n hallintaan ja kyselyiden suorittamiseen (MongoDB 2024c). Uskon, että Mongosh-työkalun oppimiseen kuluu kuitenkin enemmän aikaa, sillä komentojen opettelu saattaa olla aluksi haastavaa, erityisesti itselleni, koska en ole tottunut työskentelemään komentorivillä. Kävin kuitenkin pikaisesti läpi Beugnetin (2022) esittelemät peruskomennot, jotka auttavat ymmärtämään Mongosh-työkalun toimintaa.

Syvensin myös osaamistani React Contextista itseopiskelemalla viikon aikana. React Contextin avulla pystytään jakamaan tietoja globaalisti. Subramanian (2019, 497) demonstroi kontekstin käyttöä esimerkeillään, joissa hänen sovelluksensa eri osat käyttivät tätä yhteistä tilaa ja tämän avulla ominaisuuksia ei tarvitse viedä useiden komponenttien läpi. Subramanian mukaan (2019, 12) React Context on erittäin hyödyllinen funktio, mutta suuremmissa ja monimutkaisemmissa projekteissa tyypillisempi työkalu on Redux-kirjasto. Tutustuin myös Redux-kirjaston dokumentaatioihin, sillä olen huomannut, että Redux on hyvin suosittu osaaminen nykyisessä työmarkkinassa, erityisesti front-end- ja React-ohjelmoijan tehtävissä. Reduxin ja React Contextin eroista Sharma (2023) painottaa React Contextin helppoutta ja yksinkertaisuutta, mutta Reduxin tarjoama laajempi ja skaalautuvampi ratkaisu on usein parempi suuremmissa sovelluksissa. Tärkeänä on myös huomata, että React Context toimii luontaisesti vain Reactin kanssa, kun taas Redux on joustava ja riippumaton kirjasto, jota voidaan käyttää yhdessä muiden front-end teknologioiden kanssa.

Suurimpana ongelmana viikon aikana oli datan synkronisointi eri komponenttien välillä, joka aiheutti viiveitä datan päivittymisessä sivulla. Hyödynsin muun muassa Reactin useEffect-työkalua ja Context-funktiota synkronoinnin hallitsemiseksi. UseEffect on työkalu Reactissa, jonka avulla suoritetaan ns. sivuvaikutuksia komponentin elinkaaren aikana esimerkiksi käynnistyessä, datan tallentamisen aikana tai komponentin päivittyessä (W3Schools s.a. b.). Käytin useEffectiä muun muassa kontekstissa, jossa API-kutsun haku tapahtui sen sisällä. Tämä mahdollisti datan päivittymisen vain tarvittaessa ja oikein synkronoituna, mikä paransi sovelluksen kokonaissuorituskykyä.

Sen lisäksi toisena isona ongelmana oli tietokannan _id-kentän erottelu projektissa, joka johti tiedostojen ja moduulien väliseen epäselvyyteen, sillä osa tiedostosta tunnisti vain id-kentän ja osa vain _id kentän. Subramanian (2019, 138) selventää, että _id on varattu kenttänimi ja MongoDB on sen automaattisesti luonut jokaiselle dokumentille. _id-kenttä on myös automaattisesti indeksoitu, joka nopeuttaa tiedonhaku prosessia _id-kentän perusteella tietokannassa. Ratkaisin tämän sekaannuksen code-tietokentän avulla, joka oli generoitu käännöksen perusteella, joka loi jokaisesta code-tietokentästä yksilöivän tunnisteen.

3.6 **Seurantaviikko 6**

Maanantai 10.2.2025

Päivän tavoitteena on kokeilla analytiikkadatan keräämistä ja näyttää ensin profiilit-sivun kovakoodatun datan tilalle käyttäjän suosikkikäännösten määrän ja milloin käyttäjä on viimeksi kirjautunut sisään. Sen lisäksi tavoitteena on tehdä enemmän itseopiskelua aiheesta ja tutkia projektissa olemassa olevia tiedostoa, jotka liittyvät kirjautumiseen ja käyttäjätietojen hallintaan.

Aloitin päivän tutkimalla käyttäjän kirjautumista ja tunnistautumista projektissa. Projektissamme oli NextAuth- kirjasto tunnistautumista varten, räätälöity Next.js projekteille ja, joka tarjoaa kirjautumisvaihtoehtoja muun muassa Google-, Facebook- ja sähköpostitunnuksilla (Collins, I 2025). NextAuth-kirjasto piti sisällään JWT-tunnisteen, jota en ole aikaisemmin käyttänyt, joten opiskelin tarkemmin aiheesta ja sain lisättyä lopulta lastLogin-tietokentän käyttäjän JWT:n sisälle. Tämä mahdollisti, että koodissa pystyttiin tarkistamaan helposti, milloin käyttäjä viimeksi oli kirjautunut sisään, eikä erillistä tietokantakyselyä tarvinnut luoda. Esimerkiksi jos käyttäjällä oli lastLogin-tietokenttä valmiina ja hän kirjautui uudelleen, loin logiikan, joka päivitti kentän uusimmalla päivämäärällä ja korvasi vanhan, jotta sivustossa näkyi aina viimeisin aikamääre. Formatoin lopuksi päivämäärän käyttämällä toLocaleString-funktiota, joka formatoi päivämäärän käyttäjän paikallisten asetusten mukaisesti. Suosikkikäännöksien määrän näyttäminen oli sen sijaan hyvin helppoa, koska se oli tutumpi ja siihen olin luonut viime viikolla API:n, joka oli käytössä React Contextin sisällä.

Tiistai 11.2.2025

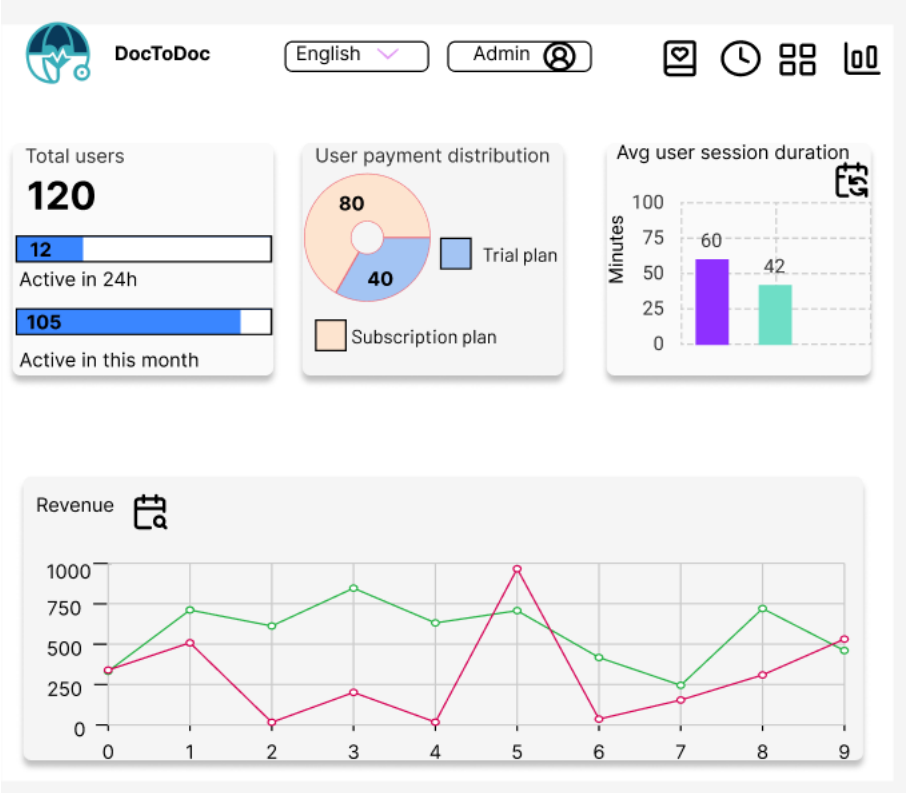
Päivän tavoitteena on suunnitella analytiikkasivua tarkemmin, käyttämällä Figma-työkalua käyttöliittymän luomiseen (Figma, 2019). Kokemukseni Figma-työkalulla on vähäinen, joten haluan syventää osaamistani tällä suunnittelutyökalulla. Haluisin myös kysyä lisätietoja IT-mentorilta ja tarkennusta tehtävään.

IT-mentori painotti aluksi, että voin keskittyä dataan, mitä voi näyttää käyttäjälle, kuten eiliset suosikkien määrät ja viimeisin kirjautuminen. Havainnoin eri toimintoja sivulla, josta voisi kerätä dataa. Päätin lopulta lisätä toiminnon, joka tallentaa, kuinka monta kertaa käyttäjä on käyttänyt kopiointitoimintoa ja hakupalkkia, jotta sen käytön yleisyyttä voidaan analysoida. Pohdin myös, että Admin-käyttäjälle eli yritykselle voisi luoda analytiikkanäkymän sivulle, kun taas normaalilla käyttäjällä ei ole tätä näkymää. Tämän avulla pystyy havainnoimaan analytiikkadatan rajat, että mitkä näytetään kaikille käyttäjille ja mitkä vain Admin-käyttäjälle ilman sekaannuksia. Opin myös käyttämään Figmassa pienoisohjelmia (widgets & plugins), joissa oli valmiina esimerkiksi diagrammeja ja kaavioita datan näyttämiseksi. Loin ensin sivuston yhteisen pohjan, jossa oli muun muassa yrityksen logo, nimi ja muut painikkeet yläosassa. Sen jälkeen aloin luomaan diagrammeja, jossa käytin valmiita pohjia ympyräkaaviolle ja diagrammille.

Keskiviikko 12.2.2025

Päivän tavoitteenani on viimeistellä Figmalla käyttöliittymäsuunnitelma, jonka jälkeen pystyn kysyä IT-mentorilta palautetta. Käyttöliittymänsuunnittelun jälkeen jatkan peruskäyttäjän tietojen näyttämistä, jossa luon jonkinlaisen API-polun analytiikkadatalle tietokannasta.

Sain käyttöliittymäsuunnittelun valmiiksi, jossa oli neljä eri diagrammia esimerkkinä. Todennäköisesti saan myöhemmin tarkennuksen yrityksille kerättävistä datoista, joten suunnittelin vain sivun muotoilua ja elementtien sijoitusta. Alla olevassa kuvassa (kuva 11.) on valmis suunnitelma käyttöliittymästä. Halusin tällä kertaa keskittyä Figmalla suunnitteluun ensin, sillä aikaisemmissa tehtävissä olin aloittanut koodauksen ilman suurempaa suunnittelua, jonka vuoksi loppuvaiheessa oli tullut paljon epäselvyyksiä eri elementtien ja komponenttien suhteen.



Kuva 11. Admin-käyttäjän suunniteltu analytiikkakäyttöliittymä.

Kalenteri-ikonien avulla käyttäjä voi valita päivämäärän tai aikavälin, jonka väliltä dataa näytetään. Tämä elementti antaa interaktiivisuutta, mutta parantaa myös kokemusta, sillä käyttäjä pystyy itse suodattamaan miltä aikaväliltä haluaa nähdä tietoja.

Jatkoin iltapäivällä käyttäjien näkymää profiilit-sivulla, jossa sain esiin käyttäjän hakupalkin käytön määrän ja myös kopioitujen käännöksien määrän. Datat haettiin oikein tietokannasta, sillä olin aikaisemmin tehnyt samanlaisia tietokannan datanhakuja, joten osasin varmistaa tietojen olevan relevantteja ja että ne päivittyivät oikein. Hakupalkin käytössä huomasin sen toimivuuden olevan hiukan epäjohdonmukainen, sillä hakupalkissa ei ollut mitään nappeja. Sen sijaan se suoritti haun joka näppäimen painalluksen jälkeen, joka oli uudenlainen hakupalkin toiminta itselle.

Torstai 13.2.2025

Päivän tavoitteena on ehdottaa muokkausta hakupalkista IT-mentorille, jotta sen toimivuus olisi käyttäjäystävällisempi, ja suunnitella hakupalkin optimointia. Sen lisäksi on luotava uusi haara näille analytiikkatiedostoille, jotta pystyn kokeilla ja työskennellä paikallisesti ilman vaikutusta main-haaraan.

Loin uuden haaran GitHubiin, jossa käytin Git stash ja stash pop-komentoa koodien siirtämiseksi omaan haaraan. Samalla vedin kaikki muutokset main-haarasta, jossa oli yhdistetty aikaisemmat

luodut muutokseni. Tällä hetkellä käyttäjälle näytettiin oikein tietokannasta viimeisin kirjautuminen, suosikkikäännösten määrä, kopioitujen käännöksien määrä. Hakupalkin funktio teki hakemisesta erittäin dynaamisen, mutta myös raskaamman sivustolle. Hakupalkissa oli käytössä Reactin useMemo (W3Schools s.a. c), joka oli itselle vieraampi, joten itseopiskelin enemmän näitä Reactin sisäänrakennettuja funktioita.

Perjantai 14.2.2025

Tavoitteenani on julkaista (push) GitHubiin omaan haaraani etenemiseni ja jatkaa hakupalkin korjausta sekä tutustua hakupalkin toimintaan syvemmin.

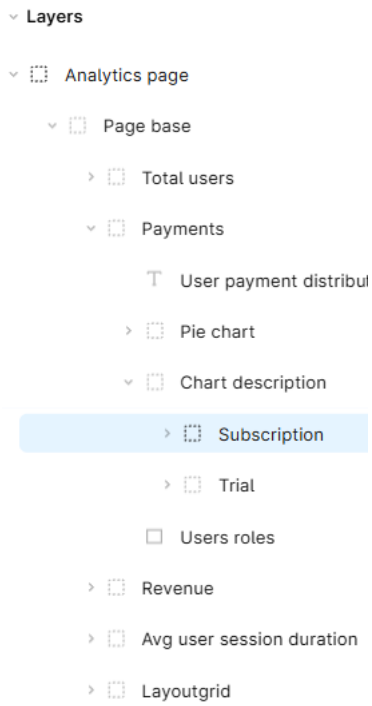
Sain julkaistua oman haaran GitHubiin ja samalla poistin aikaisempia haaroja, joiden muutokset oli jo yhdistetty main-haaraan. Huomasin, että minulla oli useita haaroja, joita en käyttänyt, sillä ne olivat yhdistetty main-haaraan jo, joten poistin ylimääräiset, jotta projektimme GitHub pysyisi selkeänä. Huomasin omassa vanhassa GitHub-haarassani olevan pienen toiminnon, joka oli unohtunut main-haaran lisäykseen, joten lisäsin tämän samalla nykyiseen koodiin samalla. Varmistin myös koodin toimivuuden npm run build -komennolla, jota IT-mentori muistutti tiimin ryhmässä. Npm run build -komennolla pystymme tarkistamaan koodin toimivuuden tuotantoympäristössä ja korjata esille tulleita virheitä. Tämä komento auttaa identifioimaan virheitä, sillä usein vaikka kehitysympäristössä ei ilmenisi ongelmia, tuotantoympäristössä niitä voi esiintyä. Näin pystymme varmistamaan sujuvan jatkuvan integraation Verceliin projektin main-haarasta.

Sain myös IT-mentorilta ohjeita hakupalkin toteuttamiseen ja sen optimointiin, jotta sen käytön seuranta pysyisi yksinkertaisena ja minimaalisena. IT-mentori muun muassa pyysi kiinnittämään huomiota debounce-funktioon, jonka avulla haku käynnistetään automaattisesti tietyn ajan kuluessa, kun käyttäjä on lopettanut kirjoittamisen. Lisäsin viivettä korkeammaksi, sillä liian alhainen viive oli aiheuttanut ongelmia ja heikensi käyttäjäystävällisyyttä. Käytin loppupäivän Lodash debounce-funktion tutkimiseen, joka mahdollistaa tämän toiminnon suorituksen.

Seurantaviikkoanalyysi 6

Tällä viikolla käytin paljon aikaa suunnitteluun varsinkin käyttöliittymään, jossa hyödynsin Figma-työkalua. Työkalussa kiinnitin paljon huomiota sen kerroksiin, jonka kautta eri elementit ovat helposti muokattavissa. Sen lisäksi elementtien yhdistäminen ryhmiin helpottaa käyttöliittymän suunnittelua, mutta myös uudelleenkäyttöä, jos toisissa sivuissa olisi samoja elementtejä. Käytin myös auto layout-ominaisuutta Figmassa, joka auttoi komponenttien sijoituksessa ja muotoilussa (Figma, 2021). Alhaalla (kuva 12.) on esitetty alkuviikolla luomani analytiikkasivun kerrokset. Näistä kerroksista huomaa, miten olin jakanut jokaisen diagrammin omaan ryhmään ja niiden sisällä ryhmitin

tekstin ja tekstiä kuvailevan diagrammin yhteen. Tämä tarkka ryhmitys ja järjestely auttaa rakenteen selkeydessä ja loogisuudessa.



Kuva 12. Admin käyttäjän analytiikkasivun kerrokset Figmassa.

Itseopiskelin myös viikon aikana pääosin kaksi aihetta: käyttäjän tunnistaminen projektissa ja Reactin sisäänrakennetut funktiot eli hookit. Itseopiskelin tunnistautumista alkuviikosta, sillä ensinnäkin se oli hyvin vieras itselleni, mutta tarvitsin myös enemmän ymmärrystä siitä, kun halusin seurata käyttäjien kirjautumista. Syy miksi en aikaisemmin ole perehtynyt käyttäjien tunnistautumiseen projektin aikana oli, koska toinen tiimin jäsenistä oli vastuussa näistä asioista ja käyttäjien maksujärjestelyistä. Perehdyin tunnistautumisessa NextAuth-kirjaston dokumentaatioon (2025), jossa mainittiin muun muassa käyttöönotosta, että projekti rakenne näille oli oltava api/auth[…nextauth], jonka avulla sisällytetään reittikäsittelyn dynaamisuus ja, jonka sisällä määritellään globaalit NextAuth-konfiguraatiot. Hyödynsin viikon aikana muun muassa NextAuth-kirjaston sisäistä useSession-funktiota, jonka avulla voidaan tarkistaa, onko käyttäjä kirjautunut sisään. Uskon tarvitsevan NextAuth-funktioita jatkossa, jos haluan seurata muun muassa käyttäjien sovelluksen käyttöä tarkemmin, esimerkiksi kuinka pitkiä keskimäärin käyttäjän sessiot sivustolla on. Syvensin ymmärrystä NextAuth-kirjaston lisäksi myös JWT-tunnisteen osalta. Subramanian (2019, 485) painottaa JWT:n skaalautuvuutta, jossa istunnontiedot on tallennettu itse tunnisteeseen. Tämän avulla JWT ei ole sidottu palvelimen muistiin, ja tunnistetiedot säilyvät myös palvelimen uudelleenkäynnistymisen yhteydessä, joka vähentää ongelmia istuntojen hallinnassa.

Reactin sisäänrakennetuista funktioista, joista yleisimmät olivat minulle tuttuja muun muassa useEffect, useState ja useContext. W3Schools-sivustolla (s.a. b) oli selitetty tärkeimmät piirteet näistä, jossa painotettiin kolmea sääntöä: sisäänrakennettuja funktioita eli hookeja voi käyttää vain Reactin funktiokomponenteissa, näitä on kutsuttava vain komponentin ylimmällä tasolla. Hookit eivät voi olla ehdollisia, eli ei voi kutsua esimerkiksi if-lausekkeiden sisällä. Tutustuin myös viikon aikana useMemo-funktioon, joka on myös yksi näistä sisäänrakennetuista funktioista. Sen avulla voidaan optimoida suorituskykyä ja W3Schoolsin (s.a. c) mukaan se hyödyntää välimuistissa säilytettyjä arvoja, jonka avulla vältytään ylimääräisillä renderöinneillä. UseMemoa käytettiin koodissa hakupalkin yhteydessä, jossa Lodash debounce-funktio oli yhdistetty siihen ja mikä varmisti optimaalisen suorituskyvyn.

Huomasin tällä viikolla kirjastossa, että olin aamulla tehokkaimmillani, mutta iltapäivisin tehokkuuteni heikkeni, jonka takia itseopiskelu sekä työtehtäviin keskittyminen oli vaikeampaa. Tämän vuoksi yritin joka päivä priorisoida tärkeimmät asiat aamulla tai aloittaa isoa tehtävää aamulla, jotta iltapäivällä voisin keskittyä kevyempiin tehtäviin ja pitää enemmän taukoja tarvittaessa. Pyrin kirjastossa myös liikkumaan taukojen aikana muun muassa tutustumalla kirjaston eri resursseihin, jotta pystyn virkistymään, mutta myös ylläpitämään keskittymiskykyä loppupäivän.

3.7 Seurantaviikko 7

Maanantai 17.2.2025

Tavoitteenani tälle päivälle on jatkaa käyttäjän analytiikkasivun työstämistä, jossa yritän saada näkyville enemmän käyttäjätietoja, kuten käyttöaika sivustossa ja käyttäjän luontiajankohta. Lisäksi haluan varmistaa ratkaisujen selkeyden ja helposti ymmärrettävyyden niin käyttöliittymässä kuin myös koodissa.

Sain käyttäjän luontipäivämäärän näytettyä, joka oli tallennettu tietokannassa subscription-tietokentän alle. Yritin muotoilla päivämäärän yhtenäiseksi last login -tietokentän kanssa käyttämällä toLocaleString -funktiota. Käyttäjän käyttöajan näyttäminen oli kuitenkin hyvin haastavaa, sillä lopetusajan määrittely oli vaikeaa, vaikka aloitusajankohdan sai suoraan talteen käyttäjän kirjautuessa sisään.

IT-mentori antoi myös positiivista palautetta analytiikkasivusta sekä Figma-suunnitelmasta ja mainitsi, että hän haluaisi jäädyttää ensimmäisen version. JetBrainsin (s.a.) sivustolla kerrotaan, että koodin jäädyttäminen tarkoittaa kehitysprosessin vaihetta, jossa uusien koodimuutosten lisääminen

ei ole sallittua, paitsi kriittisten bugien korjaamiseksi. Koodin jäädytyksen päätarkoituksena on vakauttaa ohjelmisto ennen seuraavia isoja askeleita esimerkiksi tuotantoon vientiä. Jäädytyksen tarkoitus projektissamme on, että pystymme sen jälkeen siirtymään seuraavaan versioon, samalla kun ensimmäinen jäädytetty versio tallennetaan.

Tiistai 18.2.2025

Päivän tavoitteena on jatkaa eilistä tehtävää, jossa lisään käyttäjälle näytettävää dataa näkyviin. Todennäköisesti on näytettävä muuta dataa kuin käyttäjän käyttöaika, sillä uskon, että yksityiskohtainen analyysi käyttöajasta on yritykselle olennaisempaa, joten siirrän sen myöhemmin yrityksen analytiikkanäkymään.

Korjasin aluksi käyttäjän käännösten kopioinnin eri näkymissä, sillä huomasin, että kopiointitoiminto eri näkymissä toimi epäjohdonmukaisesti. Tämä aiheutti sen, että kopiotoiminnon käyttö eri näkymissä ei vaikuttanut käyttäjän analytiikkasivun kopiointimäärään. Korjasin ongelman lähettämällä tiedot analytiikkakokoelmaan käyttämällä fetch-kutsua, johon lisäsin virheenkäsittelyn try-catch-lohkoilla, jotta kutsun yhteydessä mahdolliset virheet käsitellään oikein.

Päätin lopulta näyttää käyttäjän kirjautumisten määrät, joka oli helppo toteuttaa tarkistamalla kirjautumisen yhteydessä, jos kyseisellä käyttäjällä on valmiina analytiikkakokoelma. Jos sitä ei ole, käyttäjälle luodaan analytiikkakokoelma tietokantaan, joka alustaa kopiointi ja hakupalkin määrän nollaksi sekä kirjautumisen määrän yhteen. Sen sijaan, jos käyttäjällä on valmiina analytiikkakokoelma, niin käyttäjän id-kentän perusteella voidaan löytää tiedot ja lisätä kirjautumisen määrää yhdellä. Julkaisin ratkaisun GitHubiin omaan haaraani ja varmistin myös npm run build-komennolla toimivuuden tuotantoympäristössä.

Keskiviikko 19.2.2025

Tavoitteenani tälle päivälle on hienosäätää kielivalikon tyyliä, sillä huomasin eilen, että kielivalikko on jäänyt joidenkin elementtien taakse ja sumentui modaalien sisällä. Aion loppupäivän itseopiskella Reactin kirjastoja, joilla aion toteuttaa Admin-käyttäjän analytiikkasivun ensin kovakoodatulla datalla.

Kielivalikon virheellinen toiminta johtui liian alhaisesta z-indeksistä. Z-indeksi on CSS-ominaisuus, joka määrittää elementtien päällekkäisyyden järjestystä (Dhokai 10.4.2024). Nostin z-indeksin valikoille kaikkialla maksimille, jotta ne erottuvat taustasta, eivätkä modaalien elementit tai muiden komponenttien päällekkäisyydet häiritse käyttäjää valittaessa valikon vaihtoehtoja.

Löysin Recharts-kirjaston, joka sisälsi paljon eri diagrammi- ja kaaviovaihtoehtoja. Laajuuden lisäksi Recharts-kirjastossa oli paljon esimerkkejä kaavioiden käytöstä, joiden rakenne näytti selkeältä koodissa ja myös itse kaavio visuaalisuus oli helposti luettava. Tarkistin myös kirjaston ajankohtaisuuden GitHubin kautta, jossa huomasin, että kirjastoa on päivitetty aktiivisesti melkein päivittäin. Koen tämän tärkeäksi, sille se osoittaa kirjaston jatkuvaa kehitystä sekä virheiden korjaukset kirjastossa antavat luotettavuutta ja yhteensopivuutta uusimpien React työkalujen kanssa. Stack overflow:n (Stack Overflow s.a.) avoimissa keskusteluissa Recharts-kirjasto on myös ollut hyvin esillä, jossa ongelmatilanteisiin on jaettu vinkkejä ja ratkaisuja.

Torstai 20.2.2025

Tavoitteenani on aloittaa Admin-käyttäjän analytiikkasivun kehittäminen luomalla reititys sivulle. Kokeilen myös eri kaavioiden käyttöä sivustolla, ja korjailen kaavioiden koon sopivaksi, jotta sivulla näyttää useita eri kaavioita. Sain myös IT-mentorilta tarkennuksia, ennen kuin aloitin sivun työstämistä, jotta voisin varmistaa sen rakenteen ja sisällön reitityksen vastaavan odotuksia.

Sain luotua kovakoodatulla datalla kolme kaaviota Recharts-kirjastolla: alue-, pylväs- ja viivakaavion. Kaavioiden koodirakenteet olivat hyvin samalaisia, joten pystyin hyvin yksinkertaisesti luoda kaavioita yhtä toisensa jälkeen. Säädin kaavioiden koon käyttämällä Recharts-kirjaston ResponsiveContainer-komponenttia, jonka avulla koot ovat mukautettuja dynaamisesti käytettävässä tilassa. Sen lisäksi lisäsin kaavioihin lisäelementtejä muun muassa CartesianGrid-taustaverkon ja ToolTip-työkaluvihjeen, jotta käyttäjä voi tarkastella yksittäisiä tietoja tarkemmin. Yritin myös pitää analytiikkatiedoston mahdollisimman helposti luettavana, jotta jatkokehityksessä kovakoodatun datan korvaaminen olisi sulavaa ja vaivatonta. Käytin iltapäivän kirjaston komponenttien itseopiskeluun sekä tyylien muotoiluun ja elementtien korostukseen kokeilemalla eri lisäominaisuuksia.

Perjantai 21.2.2025

Tavoitteenani on julkaista edistymiseni GitHubiin omaan haaraani ja mahdollisesti miettiä ensi viikolle monipuolisuutta analytiikkasivulle.

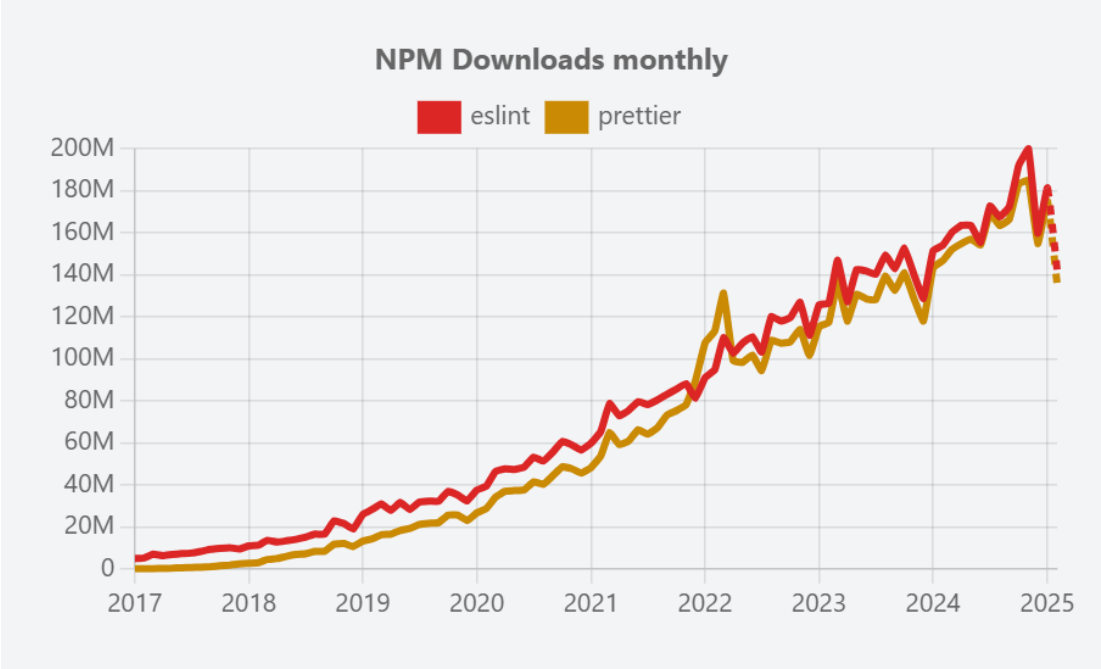
Sain julkaistua onnistuneesti omaan analytiikka GitHub haaraan edistymiseni. Sen lisäksi käytin git fetch origin main -komentoa terminaalissa, jotta saan main-haarasta ajankohtaisimman version, jonka aikana tuli pieniä konflikteja. Ongelmana oli Prettier-ohjelmaan liittyviä korjauksia, joka on koodin formatointityökalu. Suoritin komennon npm run prettier:fix, jotta pystyn havaitsemaan tarkemmin virhekohdat koodissa, jotka estävät main-haarasta tuonnin omalle haaralle. Konfliktien korjauksen jälkeen julkaisin yhdistetyt tiedostot ja uudet tiedostot omaan haaraani onnistuneesti. Suoritin vielä npm run build -komennon varmistaakseni koodin toimivuuden myös tuotantoympäristössä.

Suunnittelin analytiikkasivua pidemmälle iltapäivällä, että voisinko hyödyntää Rechartis-kirjaston hajontakaaviota (scatter chart) näyttääkseni dataa kaavion avulla. Kirjastossa oli paljon kaavioita, joita ei ollut Figman pienoistyökaluissa, joten halusin hyödyntää näitä projektissamme. Aion viikonloppuna miettiä valmiiksi, minkälaisia dataa pystytään hajontakaaviossa näyttämään Admin-käyttäjälle.

Seurantaviikkoanalyysi 7

Viikon yksi tärkeimpänä oppimisaiheena oli Recharts-kirjastoon (Recharts Group s.a.) tutustuminen ja sen kaavioiden sisältämien elementtien kokeilu ja toteutus. Kirjaston selkeyden ja esimerkkien avulla sain viikon aikana aloitettua analytiikkasivut sekä käyttäjälle että erillisen näkymän Admin-käyttäjälle. Aikaa kului paljon myös virheiden ja muutosten korjaamiseen oman haaran julkaisuvaiheessa, kun suoritin npm run build -komennon, sekä yhdistämiskonfliktien ratkaisemiseen. Koodien yhdistämiset olivat olleet aikaisemmin itselleni hyvin vieraita, ja pyrin aina varmistamaan yhdistysvaiheen olevan virheetön. Yhdistämisvaiheessa on tärkeää ja oleellista käydä läpi kaikki muutokset käyttäen Git merge -editoria, joka näyttää omat muutokset ja tulevat muutokset yhdistämisvaiheessa. Vaikka pyrin välttämään yhdistämiskonflikteja, koin kuitenkin oppivani paljon konfliktien kautta, miten niitä ratkotaan tehokkaasti ilman sekaannuksia. Subramanian (2019, 189) mainitsee parhaita käytäntöjä koodauksessa, jotta välttyy monilta virheiltä. Itselleni näistä käytännöistä uskon, että konsolin ylimääräisten viestin poistamiseen tarvitsen enemmän huomiota. Vaikka kehitysvaiheessa konsoliviestit ovat usein hyödyllisiä eivätkä ne haittaa koodin toimivuutta, tuotantoon viedessä on tärkeää poistaa ne, sillä ne kirjaavat yleensä dataa tai muuta datan rakenteen sisältöä. Yhdistämisvaiheessa virheiden käsittelyssä auttoivat paljon Prettier ja ESLint, jotka ovat molemmat hyvin yleisiä työkaluja projekteissa (Viktorsson 18.1.2025). Nämä työkalut olivat laajennuksia Visual Studio Code koodieditorissa, ja ne auttoivat automaattisesti ylläpitämään helposti luettavuutta ja yhtenäisyyttä. Prettier-työkalu auttaa koodin jäsentelyssä ja käytin sitä usein muotoilussa käyttämällä koodissa Format Document with Prettier -painiketta. Subramanian (2019, 187) kertoo, että ESLint on koodintarkistaja, joka analysoi koodia ja auttaa havaitsemaan mahdollisia virheitä, mikä edistää kokonaisuudessa koodin luettavuutta.

Tutustuin viikon loppupuolella tarkemmin ESLinttiin projektissamme ja tutustuin sen konfiguraatioihin projektissa. Konfiguraatioon oli lisätty muun muassa eslint-config-next ja typescript-eslint/parser, joka varmistaa yhteensopivuuden Next.js:n ja TypeScriptin kanssa, sillä ESLint toimii luonnostaan vain JavaScriptin kanssa. Tilastokuva (kuva 13.) havainnollistaa ESlintin ja Prettierin suosion vuodesta 2017 lähtien. Yhteiskäyttö erottuu hyvin selkeästi tilastosta, sillä molempien graafiset viivat seuraavat toisiaan hyvin tiiviisti.



Kuva 13. Latausten määrä, datan lähteenä npm-sivusto (NPM, s.a.).

Tilastokuvasta voidaan päätellä, että nämä kaksi työkalua ovat vakiinnuttaneet asemansa JavaScript-kehityksessä ja niiden yhteiskäyttö on yleistä kehittäjien keskuudessa. ESLintin dokumentaatioista löytyy myös uusimmat versiot sekä ohjeet siitä, miten sääntöjä voidaan säätää, kuten niiden tiukkuutta, ja miten kolmannen osapuolen sääntöjä voidaan lisätä ESLintin toimintaan. Tämä mukautettavuus ja joustavuus ovat myös merkittäviä syitä suosioon, sillä ne mahdollistavat sääntöjen hienosäädön omien tarpeiden mukaan.

3.8 Seurantaviikko 8

Maanantai 24.2.2025

Päivän tavoitteena on jatkaa Admin-käyttäjän analytiikkasivua, jossa lisään hajontakaavion Recharts-kirjastosta kovakoodatun datan kanssa, joka visualisoi kellon ajat päiviltä, mihin kellonaikaan on eniten kirjautumisia. Tämän avulla voidaan varmistaa tulevaisuudessa suorituskyvyn tasaisuus kaikkina aikoina ja havaita myöhemmin tietokannan datan avulla kiireisimmät ajat päivän sisällä.

Päivän lopussa sain luotua hajontakaavion, joka näyttää kovakoodattua dataa. Samalla siirsin Revenue & Profits, kaavion alas, jolloin pystyin laajentamaan sitä. Toteutukseni seurasi suuresti Figma-suunnitelmaa, joten sivuston kokonaisrakennetta oli helppo toteuttaa. Hajontakaavion toteutuksessa tuli pieniä ongelmia sen luettavuuden kannalta, sillä sen rakenne poikkesi hiukan muista kaavioista, muun muassa hajontakaavion yksittäisen datan tarkastelu.

Tiistai 25.2.2025

Tavoitteenani tälle päivälle on hienosäätää analytiikkasivua ja tutustua syvemmin Recharts-kirjaston monimutkaisempiin yhdistelmäkaavioihin sekä kaavioiden mukauttamiseen. Aion myös kiinnittää huomiota GitHubin haaroihin, jotta pysyn tietoisena haarojen muutoksista.

Huomasin main-haaran muuttuneen, ja siihen oli lisätty palautteenantonappi sivuston kulmaan. Kävin myös tarkastelemassa projektin Vercel-sivustoa, jossa main-haaran muutos oli näkyvillä sovelluksessa. Samalla Vercelissä kiinnitin myös huomiota Next.js API:en reitteihin, jossa oli muun muassa näytetty virheiden määrät reitissä, muistinkäyttömäärä ja API-reitin kutsumäärä. Sain tuotua uusimmat muutokset haaraan yhdistämisellä, joka tapahtui tällä kertaa ilman konflikteja.

Recharts-kirjaston yhdistelmäkaavioissa oli muun muassa vaihtoehtona yhdistää aiemmat luomani kaavion elementit yhteen, jolloin alue (area)-, pylväs (bar)- ja viivakaavio (linechart) olisivat yhdessä, mutta tämä vaihtoehto saattaisi olla liian tiivistetty ja todennäköisesti myös epäselkeä. Kiinnitin huomiota enemmän kaavioiden väriin, jotta elementit pysyvät helposti luettavina ja erotettavina.

Keskiviikko 26.2.2025

Tavoitteenani on tänään oman haaran koodini refaktorointi ja tehdä siitä paremmin luettava sekä ylläpidettävä. Refaktoroinnilla varmistan, että koodin toimivuus pysyy ennallaan, mutta parannan sen rakennetta, joka helpottaa jatkokehitystä ja virheiden havaitsemista. Haluan myös itseopiskella yksikkötestausta ja testityökaluja, sillä uskon sen olevan keskeinen rooli ylipäätänsä ohjelmistokehityksessä ja laadukkaan ohjelmiston varmistamisessa.

Sain lyhennettyä koodia vähentämällä toistuvuutta useEffectin sisällä käyttäjän profiilisivulla, jossa yhdistin osan useEffecteistä yhteen ja samalla poistin konsolilokituksia. Huomasin myös, että konsolissa oli varoituksia siirtyessä analytiikkasivulle, joten lähdin korjaamaan näitä. Varoitukset johtuivat mukautetusta hajontakaaviosta, jossa halusin muokata yksittäisten elementtien kokoa, jotta numeroiden vaihtelevuus korostuisi paremmin. Sain korjattua virheen ja varoitukset varmistamalla, että ominaisuuksien arvot olivat oikeissa paikoissa ja noudattivat Recharts-sääntöjä.

Löysin myös Jestin, JavaScript pohjaisen testityökalun, joka on hyvin suosittu ja myös asennettu valmiiksi projektiimme. Perehdyin Jestin materiaaleihin ja yleisesti testausaiheeseen, sillä kokemukseni ohjelmistontestauksesta on vielä vähäinen. Todennäköisesti huomenna pystyn luomaan yksinkertaisia testejä projektiimme varmistaakseni koodin toimivuuden odotetulla tavalla ja myös virheiden havaitsemisen.

Torstai 27.2.2025

Päivän tavoitteenani on luoda yhdistämispyyntö (pull request), jonka jälkeen aion aloittaa testikoodin luonnin. Tutkin myös projektin aikaisempia testitiedostoja, jossa on testattu tilauksen perumista Jestillä (Mulders, M. 25.3.2022). Testien suoritus vaatii myös esimerkkidatan luomista, ja on varmistettava testidatan yhteensopivuus käännöksen tyypin kanssa, joka sisältää vähintään pakolliset kentät.

Sain luotua hyvin yksinkertaisen testin, jossa testataan suosikkikäännöksen poistoa suosikeista. Tämän testin tekeminen koitui yllättävän vaikeaksi, sillä syntaksi oli hyvin erilainen ja se sisälsi myös toisen testikirjaston lisäpaketin. Nämä funktiot, kuten fireEvent ja screen, simuloivat napin löytymistä ja painamista. Ne kuuluivat yhdessä dom testing library -kirjastoon, mutta toimi yhdessä Jestin kanssa. Testin pystyin käynnistämään suorittamalla npm test -komennon, jonka jälkeen testin etenemistä pystyi seuraamaan terminaalin lokituksessa. Lokituksessa näkyi suorituskykyyn liittyviä tietoja, kuten testin suoritusaika sekä kuinka monta testiä meni läpi ja mitkä eivät. Opin myös iltapäivällä erottamaan eri napit toisistaan, jos halusin esimerkiksi kokeilla kopiointitoimintoa suosikit-sivulla. Annoin napille yksilöllisen nimen, johon pystyin viittaamaan testitiedostossa screen-funktion avulla, jotta testi löytää oikean napin testattavaksi.

Perjantai 28.8.2025

Tavoitteenani on suunnitella monimutkaisempia testejä projektin pääsivulle, jotta pystyn toteuttamaan osan ensi viikolla. Eilinen yhdistämispyyntö oli myös hyväksytty main-haaraan, joten luon myös oman haaran näille testaustiedostoille ja aion siirtää aikaisemmin luodut testitiedostot sinne.

Lähdin luomaan omaa haaraani GitHubissa hakemalla ensin uusimman version main-haarasta, josta pystyn luomaan oman haaran ajankohtaisimmasta main-haarasta. Luomisen jälkeen pystyin siirtämään aikaisemmat testitiedostoni oikeaan haaraan käyttämällä git stash- ja git pop -komentoa, jonka avulla pystyn säilyttämään muutokseni paikallisesti ja ottamaan ne käyttöön uudessa haarassa.

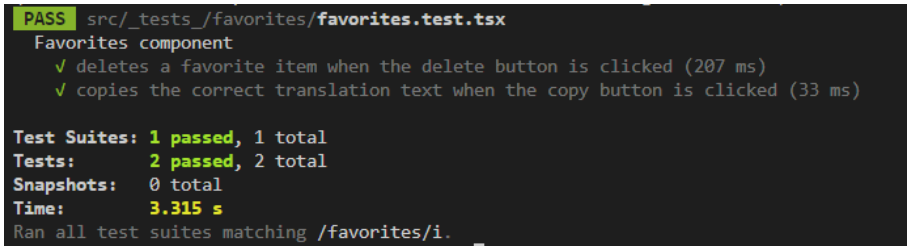
Suunnittelin CRUD-toimintojen testausta Jestillä pääsivulle, joka on monimutkaisempaa, sillä se on yksi yläkomponenteista komponenttihierarkiassa ja vaikuttaa useisiin alakomponentteihin. Esimerkkidatan luominen tähän yläkomponenttiin monimutkaistuu myös, sillä on huomioitava alikomponenttien riippuvuudet ja datan rakenteet. CRUD-toimintojen testauksen tavoitteena on varmistaa, että käyttäjän suorittamat toiminnot toimivat aina odotetusti, ilman odottamattomia virheitä. Tämä tarkoittaa esimerkiksi, että testauksen on katettava eri skenaarioita, kuten onnistuneet toiminnot, toiminnon peruuttaminen esimerkiksi lisäyksessä ja muokkauksessa sekä mahdollisten virhetilanteiden käsittely, muun muassa tyhjissä syötteissä.

Seurantaviikkoanalyysi 8

Viikon suurimpana teemana oli koodin testaus, joka vei paljon aikaa itseopiskelussa, sillä tämä osa-alue oli itselleni hyvin vieras. Testien kirjoittamisen alkuvaihe tuntui haastavalta sen syntaksin eroavaisuuden ja uusien funktioiden vuoksi. Onneksi projektissa oli aikaisemmin tehty yksikkötestejä Jest-testityökalulla, mikä helpotti merkittävästi aloitustani. Jestin dokumentaatio oli hyvin kattava, ja siinä avattiin testin kirjoittamisen perusrakenteet (OpenJS Foundation s.a.). Vaikka Jestin dokumentaatio oli hyvin kattava, esimerkkitestit sivustolla olivat liian yksinkertaisia, ja niitä oli vaikea soveltaa projektiini. Testien liiallinen yksinkertaisuus sivustolla johti siihen, että sovelsin muita materiaaleja, kuten YouTube-videoiden käyttö opastukseen. PedroTech (2022, 12–21 min.) demonstroi videollaan eri konsepteja, joiden avulla sain syvällisempää selitystä. Videolla käsiteltiin muun muassa nappien painamista ja oikeiden nappien löytämistä testin aikana, mitä sovelsin lopulta omassa testissäni viikon aikana. Näiden konseptien oppimisen avulla pystyin luomaan oikeanlaisen simuloinnin käyttäjän toiminnoista.

Avainfunktioina Jestissä on Matcherit, joiden avulla voidaan vertailla testattavan koodin palauttamaa arvoa odotettuun arvoon. Mock-funktiot ovat myös erittäin hyödyllisiä yksikkötesteissä, koska ne mahdollistavat logiikan testaamisen ilman, että ulkoiset riippuvuudet vaikuttavat (Mulders 2022). Omassa testitiedostossani oli muun muassa fetch-kutsut API:sta, suosikkikohteen poistaminen sekä tekstin kopiointi muokattu mock-funktioiksi. Tämän avulla testit pystyttiin suorittamaan eristetysti ja keskittymään vain komponentin sisäiseen toimintaan. Mulders (2022) painottaa Jestin eristettävyyden tärkeyttä, sillä Jestin rinnakkaissuorituksen avulla pystytään varmistamaan, että testit eivät vaikuta toisiinsa. Tämä parantaa testitulosten luotettavuutta ja vähentää myös virheiden mahdollisuutta.

Jest kuvakaappaus (kuva 14.) havainnollistaa testien kulun ja näyttää testien määrän, kuinka monta testiä meni läpi ja kokonaisuuden suoritusaika. Tämä selkeä raportointi testin kulusta auttaa kehittäjiä virheiden tunnistuksessa ja niiden korjaamisessa, jotta testit onnistuvat ja toiminnallisuus säilyy oikeana.



Kuva 14. Suosikit-sivun yksinkertaisten testien lokitus terminaalissa.

Työharjoitteluni myös läheni loppua, joten pohdin ajankäytön kannalta, mitä tehtäviä ehdin ottaa viimeiselle viikolle. Tärkeänä on myös varmistaa ratkaisujeni jatkokehitettävyys ja pitää tiimini ajan tasalla tekemisestäni, jotta he voivat jatkaa työtäni sujuvasti oman harjoitteluni jälkeen. Mainitsin IT-mentorilleni, että aion keskittyä harjoittelun loppuvaiheessa testaamiseen, jotta voin varmistaa omien ratkaisujeni toimivuuden ja luotettavuuden. Koko testausaihe on hyvin laaja, keskityn tarkemmin vain yksikkötestaukseen, enkä perehdy tarkemmin esimerkiksi käytettävyys- tai suorituskyvyn testaukseen ajan takia. Yksikkötestaus nimensä mukaan varmistaa yksittäisten komponenttien ja funktioiden toimivuuden ennen niiden yhdistämistä isoihin kokonaisuuksiin (Smartbear 2025). Ensiviikon monimutkaisempien testien luonnissa tulen todennäköesti lataamaan enemmän Jestin kanssa yhteensopivia kirjastoja, kuten aikaisemmin ladattu dom testing library. Tämän lisäksi löysin myös user-event-kirjaston testing libraryn sisältä, ja uskon tarvitsevani sitä ensi viikon testien kirjoittamisessa.

4 Pohdinta

Opinnäytetyötä kirjoittaessa olen pystynyt reflektoimaan kasvuani ammatillisesti, mutta myös päiväkirjan kirjoittajan näkökulmasta. Itse opinnäytetyön kirjoittaminen on ollut sujuvaa, ja olen pystynyt pysymään aikataulussani hyvin tiukasti. Uran ensimmäisenä IT-alan työpaikkana oppimiskäyrä on ollut hyvin jyrkkä, mutta samalla myös palkitsevaa, sillä pystyin soveltamaan ja omaksumaan moderneja ohjelmistokehityksen teknologioita.

Seurantaviikkoanalyysini pitivät sisällään viikon edistymisen projektissa, keskeisimmät oppimiskokemukseni sekä mahdolliset kohtaamani ongelmat tai haasteet. IT-mentorini tuki ja ohjaus harjoittelun aikana ovat olleet korvaamattomia. Hän on antanut ehdotuksia teknologioihin liittyen ja edistänyt myös ammatillista kasvuani. Hänen kokemuksenansa ja asiantuntemuksensa ovat olleet merkittävä apu koko projektin ajan, ja olen myös saanut häneltä arvokasta palautetta työskentelystäni ja osaamisestani.

4.1 Ammatillisen kehittymisen ja kirjoittamisen tulokset

Osaamiseni kehittyi huomattavasti teknologioissa, sillä projektissamme oli monia itselleni uusia teknologioita tai sellaisia, joista minulla oli vain pinnallinen kokemus koulusta tai vapaa-ajalla. Kuten johdannossa mainitsin, omaksuin Next.js työkalun hyvin nopeasti hyödyntäen React- ja JavaScript-osaamistani. Harjoittelun aikana loin muun muassa Next.js:n avulla API-polkuja sekä opin erottamaan tarkemmin palvelin- ja selainkomponentit. Näiden erottamiseksi Next.js:ssä käytetään ”use client” -sisäänrakennettua direktiiviä, joka määrittää komponentin suorituksen vain selaimen puolella asiakkaalle, eikä palvelinpuolella (Meet 9.1.2025). Tämä mahdollistaa Reactin hookkien käytön, jotka toimivat vain selaimessa ja ilman ”use client” direktiiviä, koodin suoritus epäonnistuu.

Koen myös, että TypeScript- ja React-osaamiseni ovat pysyneet melko samalla tasolla, sillä pidin niitä vahvuuksinani ohjelmistokehityksessä jo ennen harjoittelun alkua. Olin aikaisemmissa projekteissani hyödyntänyt muun muassa React Contextia, joten sen käyttöönotto oli sujuvaa ja rakenteeltaan muistutti hyvin paljon aiempia projektejani. Yhdistin Reactissa myös ohjelmistotestauksen ja Recharts-kirjaston käytön, mikä antoi minulle mahdollisuuden syventää osaamistani sekä yksikkötestauksessa että datan visualisoinnissa. Testauksessa keskityin erityisesti komponenttien eristettyyn testaukseen, jotta pystyin varmistamaan komponentin yksittäisten funktioiden toimivuuden ja luotettavuuden. Recharts-kirjaston avulla pystyin luomaan monipuolisia kaavioita ja taulukoita eri tietomäärillä ja esittämään ne myös käyttäjäystävällisellä tavalla.

Yllättäen itselleni, pääsin myös käsittelemään tietokantadataa Python-ohjelmointikielellä ja sen Pandas-kirjastolla. Opin muuntamaan dataa Excel-tiedostosta JSON-muotoon ja syöttämään tietueita MongoDB:n tietokantaan. Tämän tehtävän myötä ymmärsin suurten tehtäväkokonaisuuksien pilkkomisen tärkeyden. IT-mentorini aluksi pyysi vain datan parsimista MongoDB:n tietokantaan, mutta prosessi osoittautui monivaiheiseksi. Ensin muunsin datan JSON-muotoon ja varmistin, että JSON-muodossa käännöksien rakenteet vastasivat olemassa olevia tietueita. Kaikkien näiden vaiheiden aikana oli tärkeää käsitellä mahdollisia virheitä, jotka ilmenivät datan käsittelyssä. Esimerkiksi tyhjien ja null-arvojen salliminen oli otettava huomioon, sillä ne voivat olla kriittisiä ja aiheuttaa ongelmia syöttämisessä. Testasin syöttöprosessia runsaasti paikallisessa tietokannassa (localhost) varmistaakseni, että lopullinen siirto varsinaiseen tietokantaan sujuisi mahdollisimman virheettömästi.

Syvensin osaamistani NoSQL-tietokantoista ja vertailin toimintaperiaatetta perinteisiin SQL-tietokantoihin. NoSQL-tietokannoissa suurimpana erona on, että ne eivät perustu relaatiomalliin, vaan tiedon tallennus on joustavaa ilman ennalta määriteltyjä skeemoja. Tämä tarkoittaa, että tietueiden rakenne ja kentät saattavat poiketa toisistaan, joka mahdollistaa dynaamisuuden ja nopean sovelluskehityksen (Smallcombe 15.2.2024). MongoDB:n dokumenttipohjainen tiedon varastointi oli itselleni helposti ymmärrettävissä ja opin MongoDB:n kautta myös JSON:n ja BSON:n erot, sillä MongoDB:ssä dataa säilytetään sisäisesti BSON muodossa. Alla olevassa taulukossa (taulukko 3) kuvaillaan nämä erot tarkemmin.

	JSON	BSON
Koodimuoto	UTF-8	Binääri
Tietotyypin tuki	Merkkijonot, booleanit, numerot, taulukot, oliot, null-arvot	Merkkijonot, booleanit, numerot (kokonaisluvut, liukuluvut, pitkät, decimal128), taulukot, null-arvot, päivämäärät, binääridata
Luettavuus	Ihmisille ja koneille luettavaa	Vain koneille luettava

Taulukko 3. JSON ja BSON erot havainnollistettu (MongoDB 2024d).

Etätyöskentelijänä opin teknillisten osaamisten lisäksi priorisoimaan ja aikatauluttamaan tehtävien tekoa ja itseopiskelua. Koska työskentelyni oli hyvin itsenäistä, opin ottamaan vastuuta päätöksenteosta ja kysymään aktiivisesti palautetta IT-mentorilta varmistaakseni, että etenin tehtävissä toivotulla tavalla. Koin, että monissa tilanteissa minulla oli melko vapaat kädet esimerkiksi koodin tuottamisessa, että millä tavoilla luon komponentin, kunhan lopputuloksen visuaalisuus oli yhtenäinen

sivuston muiden komponenttien kanssa. Ajan myötä uskoin saaneeni luottamusta IT-mentorilta ja pystyin osoittamaan aktiivisuuteni, jolloin sain vastuulleni monimutkaisempia ja vaativampia tehtäviä projektin edetessä. Taukojen merkitys korostui myös itsenäisessä työskentelyssä, sillä huomasin selkeitä eroja tuottavuudessani silloin, kun pidin taukoja epäsäännöllisesti. Tämän vuoksi loin päivittäisen aikataulun, johon sisällytin säännöllisiä taukoja, mikä auttoi keskittymisen ja tehokkuuden ylläpidossa. Tämän seurauksena työn laatu parani, mutta myös jaksaminen pitkien itseopiskelu- ja työpäiväjaksojen jälkeen oli tasaista.

Opinnäytetyötä kirjoittaessa opin dokumentoimaan jokaisen työpäivän tapahtumia, mutta myös olemaan lähdekriittinen lähteitä etsiessä työni tueksi. Esimerkiksi Google Scholarissa huomioin lähteiden helposti ymmärrettävyyden ja julkaisuajankohdan. Pyrin myös vastuullisuuteen kiinnittämällä erityisesti huomiota lisensseihin ja muihin käyttölupiin kuvissa ja taulukoissa. Pääkirjallisuudessani ja monissa muissa kirjoissa oli ”All rights reserved” -merkintä, mikä tarkoittaa, että kirjan kuvia ja taulukoita ei voida käyttää ilman tekijän lupaa. Joissain oli sen sijaan ”Fair to use” -maininta, kuten W3Schoolsissa, joka tarkoittaa sisällön käytön sallimisen rajoitetusti. Mozilla Developer Networkin (MDN web docs) sisältö on lisensoitu CC-BY-SA v2.5:llä tai uudemmalla, joka tarkoittaa, että sisältöä voi vapaasti käyttää, jakaa, muokata ja hyödyntää myös kaupallisesti, kunhan materiaalin alkuperäinen lähde ja tekijät ovat mainittuja (Mozilla s.a. b). Opinnäytetyötä kirjoittaessa, syvensin myös tietämystäni IT-alan sanastosta, esimerkiksi kirjoitusten aikana termien selityksessä ja taustojen avaamisessa. Tämä vahvisti teknisten käsitteiden ymmärtämistä sekä kykyä selittää termit lukijalle, jolla ei ole välttämättä taustaa ohjelmistokehitys alalta.

4.2 Opinnäytetyöni merkitys ja tulevaisuuden suunta

Opinnäytetyöni aikana huomasin, että tehokkuuteni vaihtelee selkeästi päivän aikana. Koen olevani tehokkaimmillani aamupäivällä, jolloin pyrin priorisoimaan päivän tavoitteiden saavuttamista. Yleensä iltapäivällä tehokkuuteni laskee asteittain, jolloin pyrin viimeistelemään päivittäisten tavoitteen saavuttamisen ja aloittamaan myös itseopiskelun teknologioista, joita tulen tarvitsemaan seuraavina päivinä. Uskon, että tästä oivalluksesta tulee olemaan paljon hyötyä tulevaisuuden etätyössä ja itseohjautuvissa työympäristöissä, joissa oma työskentelyrytmi on itse päätettävissä. Teknologioiden oppimista ja hankittua tietämystä voin hyödyntää myös työhaastatteluissa, joissa osaamistani voidaan haastaa esimerkiksi teknisissä haastatteluissa, joissa perehtyminen eri teknologioihin, järjestelmien toiminnan kokonaisymmärrys ja ongelmanratkaisutaidot ovat pääosana. Tällaisissa tilanteissa voin tuoda esiin konkreettisia esimerkkejä opinnäytetyöni aikana kohtaamisista haasteista, ja miten pystyin ratkaista ne. Tämä avulla pystyn osoittamaan motivaationi kehittyä alan ammattilaisena ja myös kykyä soveltaa oppimaani käytännön tilanteissa.

IT-ala kehittyy jatkuvasti, joten uskon, että taitojeni jatkuvaa kehitystä on myös harkittava ja suunniteltava tulevaisuudessa. IT-mentori mainitsi myös, että vaikka hänellä itsellään oli yli viiden vuoden kokemus alalta, hän joutuu usein kuitenkin turvautumaan hakukoneisiin ja myös tekoälyn ohjeistukseen tiedon etsimisessä. Tämä kertoo siitä, miten nopeasti teknologiat muuttuvat ja kuinka tärkeää on pysyä ajan tasalla. Tavoitteenani tulevaisuudessa on säännöllinen koodaus, esimerkiksi kehittämällä keskeneräisiä projektejani GitHubissa sekä osallistua avoimen lähdekoodin projekteihin ja niiden kehittämiseen. Avoimen lähdekoodin projekteissa pyrin ensin löytämään sopivan projektin kiinnittämällä huomiota projektin teknologioihin ja GitHubin issue-välilehteen, josta löytyy mahdolliset ongelmat ja tehtävät projektissa. Yleensä nämä ongelmat ovat merkitty tunnisteilla, joista näkee tehtävän tyypin, että onko se dokumentaatioon liittyvä, virheidenkorjausta tai muu sivuston parannusehdotus.

4.3 Yhteenveto

Kaiken kaikkiaan opinnäytetyön suoritus oli yllättäen luontevaa työn ohella ja pystyin sen avulla havainnoimaan vahvuuksiani ja heikkouksiani. Sen lisäksi pystyin myös reflektoimaan ajankäyttöä uusien teknologioiden oppimisessa. Huomasin, kuinka tärkeää on suunnitella oppimista ennakoivasti ja varata riittävästi aikaa syventymiselle. Uskon palaavani tulevaisuudessa tietoperustassa käsiteltyihin materiaaleihin, erityisesti kohdatessani uusia haasteita tai työskennellessäni vastaavien ratkaisujen parissa. Näin pystyn varmistamaan, että osaamiseni pysyy ajan tasalla ja vastaa alan vaatimuksia.

Opinnäytetyö ei ollut pelkästään kirjoitelma itselleni, vaan myös konkreettinen askel ammatillisessa kasvussa. Se opetti minulle pitkäjänteisyyttä, oman työskentelyn arviointia sekä jatkuvan oppimisen tärkeyttä. Nämä ovat asioita, joita haluan pitää mukana myös tulevissa työtehtävissäni ja urani aikana.

Lähteet

Abbas, M. 31.8.2023. React Testing Library: Understanding act() and when to use it. Luettavissa: https://medium.com/@AbbasPlusPlus/react-testing-library-understanding-act-and-when-to-use-it-301bd06fd1bc. Luettu 3.3.2025.

Ahmed, A. 30.7.2023. What is Vercel and Why You Should Use It? Luettavissa: https://www.get-fishtank.com/insights/what-is-vercel. Luettu 9.4.2025.

Arora, P. 13.3.2024. Basics of caching. Luettavissa: https://dev.to/paras594/basics-of-caching-139l. Luettu 27.1.2025.

Ayebola, J. 16.4.2024. Function components vs class components in React – with examples. Luettavissa: https://www.freecodecamp.org/news/function-component-vs-class-component-in-react/. Luettu 31.1.2025.

Barbieri, L. 2022. Successful remote working succinctly. Luettavissa: https://www.syncfusion.com/succinctly-free-ebooks/successful-remote-working-succinctly. Luettu: 21.2.2025.

Beugnet, M. 31.1.2022. MongoDB cheat sheet. Luettavissa: https://www.mongodb.com/developer/products/mongodb/cheat-sheet/. Luettu 7.2.2025.

Chauhan, H. 14.8.2024. Mastering responsive design with Tailwind CSS: tips and tricks. Luettavissa: https://dev.to/hitesh_developer/mastering-responsive-design-with-tailwind-css-tips-and-tricks-1f39. Luettu 4.2.2025.

Chris, K. 29.6.2022. What is localhost? Localhost IP address explained. Luettavissa: https://www.freecodecamp.org/news/what-is-localhost/. Luettu 7.2.2025.

Chugh, V. 30.5.2023. Python Pandas tutorial: The ultimate guide for beginners. Luettavissa: https://www.datacamp.com/tutorial/pandas?dc_referrer=https%3A%2F%2Fwww.google.com%2F. Luettu: 20.1.2025.

Codecademy 2025. What is CRUD? Luettavissa: https://www.codecademy.com/article/what-is-crud. Luettu: 20.1.2025.

Collins, I. 2025. NextAuth.js. Luettavissa: https://next-auth.js.org/. Luettu: 19.2.2025.

Coursera 21.1.2025. What is Python Used For? A Beginner's Guide. Luettavissa: https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python. Luettu 11.4.2025.

Dhokai, C. 10.4.2024. Beyond the Basics: Mastering z-index in CSS. Luettavissa: https://dev.to/chintanonweb/beyond-the-basics-mastering-z-index-in-css-42fp. Luettu: 9.4.2025.

Figma, 2019. What is Figma? Luettavissa: https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma. Luettu 9.4.2025.

Figma, 2021. Add auto layout to a design. Luettavissa: https://help.figma.com/hc/en-us/articles/5731482952599-Add-auto-layout-to-a-design. Luettu: 30.4.2025.

Gillis, A. S., Lutkevich, B., & Nolle, T. 2024. What is an API (application programming interface)? Luettavissa: https://www.techtarget.com/searchapparchitecture/definition/application-program-interface-API. Luettu: 25.1.2025

Gupta, A. 26.1.2025. HTML vs. CSS: The Best Guide to Understand the Difference. Luettavissa: https://www.simplilearn.com/tutorials/html-tutorial/html-vs-css. Luettu: 25.4.2025.

Ibrahim, M. 2.8.2024. What is JWT? Understanding JSON Web Tokens. Luettavissa: https://supertokens.com/blog/what-is-jwt#what-is-a-jwt. Luettu 10.2.2025.

Jena, K. B. 17.11.2024. A definitive guide to learn the SHA-256 (Secure Hash Algorithms) Luettavissa: https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm. Luettu: 20.1.2025.

Jetbrains s.a. What is a code freeze? Luettavissa: https://www.jetbrains.com/teamcity/ci-cd-guide/faq/code-freeze/. Luettu: 17.2.2025.

Jith, S. 3.10.2024. Next.js Route Handlers in TypeScript: Comprehensive Guide. Luettavissa: https://medium.com/@1shyam2shyam/next-js-route-handlers-in-typescript-comprehensive-guide-ee3c9ea773b8. Luettu: 9.4.2025.

Joodi, 8.12.2024. Turbopack in Next.js: the future of development bundling. Luettavissa: https://dev.to/joodi/turbopack-in-nextjs-the-future-of-development-bundling-lhd. Luettu: 23.1.2025

Joseph, S. 14.7.2023. Linting Git Commit Messages Using Husky: A Guide to Cleaner Version Control. Luettavissa: https://medium.com/@sijokoothur/linting-git-commit-messages-using-husky-a-guide-to-cleaner-version-control-2ac37605433a. Luettu 9.4.2025.

Kernaghan, C. 14.12.2023. Loading spinners: Their purpose and alternatives for better UX. Luettavissa: https://blog.logrocket.com/ux-design/loading-spinners-purpose-alternatives/. Luettu 9.4.2025

Knell, J. 7.11.2023. What is hard coding. Luettavissa: https://www.bloomtech.com/article/what-is-hard-coding. Luettu: 25.1.2025.

Kramer, N. 31.3.2024. Tailwind CSS basics for beginners. Luettavissa: https://daily.dev/blog/tailwind-css-basics-for-beginners. Luettu: 21.1.2025.

Mariano, C. L. 2017. Benchmarking JavaScript Frameworks. Luettavissa: https://arrow.tudublin.ie/cgi/viewcontent.cgi?article=1100&context=scschcomdis. Luettu: 20.1.2025.

MDN Web Docs Mozilla s.a. a. What is JavaScript? Luettavissa: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/What_is_JavaScript#a_high-level_definition. Luettu: 20.1.2025.

MDN Web Docs Mozilla s.a. b. Attribution and contribution licensing. Luettavissa: https://developer.mozilla.org/en-US/docs/MDN/Writing_guidelines/Attrib_copyright_license. Luettu: 14.4.2025.

Meet 9.1.2025. What is 'use client' in Next.js and When Should You Use It? Luettavissa: https://meetpan1048.medium.com/what-is-use-client-in-next-js-and-when-should-you-use-it-96d5de250bc4. Luettu: 10.4.2025.

Microsoft s.a. TypeScript is JavaScript with syntax for types. Luettavissa: https://www.typescriptlang.org/. Luettu: 20.1.2025.

MongoDB 2024a. MongoDB basics. Luettavissa: https://www.mongodb.com/resources/products/fundamentals/basics. Luettu: 20.1.2025.

MongoDB 2024b. MongoDB CRUD operations. Luettavissa: https://www.mongodb.com/docs/manual/crud/. Luettu: 22.1.2025.

MongoDB 2024c. Welcome to MongoDB Shell (mongosh). Luettavissa: https://www.mongodb.com/docs/mongodb-shell/. Luettu: 9.4.2025

MongoDB 2024d. JSON and BSON. Luettavissa: https://www.mongodb.com/resources/basics/json-and-bson. Luettu 10.4.2025.

Mongoose s.a. Error.CastError. Luettavissa: https://mongoosejs.com/docs/5.x/docs/api/error.html#error_Error-CastError. Luettu 30.4.2025.

Moskaleva, N. 13.11.2023. What is hashing? Luettavissa: https://www.criipto.com/blog/what-is-hashing. Luettu: 20.1.2025.

Mulders, M. 25.3.2022. Jest testing: A helpful, introductory tutorial. Luettavissa: https://www.testim.io/blog/jest-testing-a-helpful-introductory-tutorial/. Luettu: 3.3.2025

OpenJS Foundation s.a. Jest Documentation. Luettavissa: https://jestjs.io/. Luettu: 3.3.2025.

PedroTech 16.6.2022. Testing In React Tutorial – Jest and React Testing Library. Video. Katsottavissa: https://www.youtube.com/watch?v=JBSUgDxICg8. Katsottu 3.3.2025.

RedHat 12.12.2023. What is CI/CD? Luettavissa: https://www.redhat.com/en/topics/devops/what-is-ci-cd. Luettu: 20.1.2025.

Recharts Group s.a. Recharts. Luettavissa: https://recharts.org/en-US/. Luettu: 25.2.2025

Shah, S. 28.5.2024. Next.js folder structure simplified: a comprehensive overview. Luettavissa: https://www.dhiwise.com/post/nextjs-folder-structure-simplified-a-comprehensive-overview. Luettu: 22.1.2025.

Sharma, M. 11.2.2024. Demystifying Vercel: How it works and how to create Vercel. Luettavissa: https://medium.com/@mayank2803sharma/demystifying-vercel-how-it-works-and-how-to-create-vercel-763586070478. Luettu: 20.1.2025.

Sharma, T. 5.6.2023. Redux vs Context-API. Luettavissa: https://medium.com/@tanyas022000/redux-vs-context-api-45e5bae61df2. Luettu 6.2.2025.

Shrestha, N. 2008. A study on students use of library resources and self-efficiancy. Luettavissa: https://core.ac.uk/download/pdf/290487934.pdf. Luettu 12.2.2025

Slack, Workforcelab 2023. Workforce Index. Luettavissa: https://d34u8crftukxnk.cloudfront.net/slackpress/prod/sites/6/Workforce-Lab-Winter-2023-Workforce-Index.pdf. Luettu: 21.1.2025.

Smallcombe, M. 15.2.2024. SQL vs NoSQL: 5 Critical Differences. Luettavissa: https://www.integrate.io/blog/the-sql-vs-nosql-difference/. Luettu: 10.4.2025.

Smartbear 2025, What Is Unit Testing? Luettavissa: https://smartbear.com/learn/automated-testing/what-is-unit-testing/. Luettu: 9.4.2025.

Stackoverflow s.a. Home. Luettavissa: https://stackoverflow.com/. Luettu: 30.4.2025.

Subramanian, V. 2019. Pro Mern Stack. 2. painos. Luettavissa: https://edu.anarcho-copy.org/Programming%20Languages/Node/Pro%20MERN%20Stack,%202nd%20Edition.pdf. Luettu: 20.1.2025.

Tailwind Labs 2025. Animation. Luettavissa: https://tailwindcss.com/docs/animation. Luettu: 16.1.2025.

Vercel 2025. React Foundations About React and Next.js. Luettavissa: https://nextjs.org/learn/react-foundations/what-is-react-and-nextjs. Luettu: 20.1.2025.

Viktorsson, R. 18.1.2025. Setting Up EsLint and Prettier for a TypeScript Project. https://medium.com/@robinviktorsson/setting-up-eslint-and-prettier-for-a-typescript-project-aa2434417b8f. Luettu 9.4.2025.

Wales, M. 8.12.2020. 3 Web Dev Career Decoded: Front-End vs Back-End vs Full Stack. Luettavissa: https://www.udacity.com/blog/2020/12/front-end-vs-back-end-vs-full-stack-web-developers.html. Luettu: 20.1.2025.

W3Schools s.a. a. Git and Github Introduction. Luettavissa: https://www.w3schools.com/git/git_intro.asp?remote=github. Luettu: 20.1.2025.

W3Schools s.a. b. React Hooks. Luettavissa: https://www.w3schools.com/react/react_hooks.asp. Luettu: 17.2.2025.

W3Schools s.a. c. React memo. Luettavissa: https://www.w3schools.com/react/react_memo.asp. Luettu: 19.2.2025

W3Schools s.a. d. RegEx in Python. Luettavissa: https://www.w3schools.com/python/python_regex.asp. Luettu 25.4.2025

W3Schools s.a. e. Python String strip() Method. Luettavissa: https://www.w3schools.com/python/ref_string_strip.asp. Luettu: 30.4.2025.