

Veera Annala

## **TYÖ JUNIOR SOFTWARE DEVELOPERINA**

## **TYÖ JUNIOR SOFTWARE DEVELOPERINA**

Veera Annala  
Opinnäytetyö  
Syksy 2021  
Tietojenkäsittelyn koulutusohjelma  
Oulun ammattikorkeakoulu

## TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma

---

Tekijä: Veera Annala

Opinnäytetyön nimi: Työ Junior Software Developerina

Työn ohjaaja: Minna Kamula

Työn valmistumislukukausi ja -vuosi: Syksy 2021

Sivumäärä: 46

---

Tämä opinnäytetyö on toteutettu päiväkirjamuotoisena. Siinä käydään läpi aloittelevan ohjelmoijan kymmenen viikon työtehtävät Liana Technologiesilla. Opinnäytetyössä kuvataan työn analyysi, perustietoja työpaikasta, sidosryhmät sekä vuorovaikutus työpaikalla. Päivittäin kirjataan ylös sen päivän aikana tehty työ ja viikottain kirjoitetaan analyysi viikon teemasta ja kuluneen viikon työtehtävistä. Päiväkirjaa on kirjoitettu 10 työviikkoa eli 50 työpäivää.

Opinnäytetyön tarkoituksena on seurata opiskelijan kehittymistä ohjelmoijana ja työntekijänä päivittäisten päiväkirjamerkintöjen avulla. Viikon lopuksi kirjoitetaan analyysi kuluneesta viikosta ja sen teemasta käyttäen lähteinä alan kirjallisuutta ja laadukkaita blogitekstejä.

Opinnäytetyön lopussa on pohdinta, jossa tarkastellaan opiskelijan kehitystä kymmenen viikon päiväkirjan kirjoittamisen aikana.

---

Asiasanat: Backend, ohjelmointi, tietokannat, PHP, PostgreSQL, päiväkirja

## ABSTRACT

Oulu University of Applied Sciences  
Degree Programme in Business Information Systems

---

Author: Veera Annala  
Title of thesis: Work as a Junior Developer  
Supervisor: Minna Kamula  
Term and year when the thesis was submitted: Fall 2021  
Number of pages: 46

---

This thesis is made in the form of a diary. It describes the work of a novice programmer at Liana Technologies in a ten-week period.

The purpose of the thesis is to monitor the student's development as a programmer and employee with the help of daily diary entries. In the end of the week an analysis is written based on diary entries and current week's theme.

At the end of the thesis, there is a reflection that looks at the student's development during the writing of a ten-week diary.

---

Keywords: Backend, programming, databases, PHP, PostgreSQL, diary

# SISÄLLYS

1	JOHDANTO .....	6
1.1	Yritys ja työympäristö .....	6
1.2	Käsitteitä.....	7
2	NYKYTILANTEEN KUVAUS .....	8
2.1	Oman nykytilanteen analyysi.....	8
2.2	Työpaikan sidosryhmät.....	9
2.3	Vuorovaikutustaidot työpaikalla .....	10
3	PÄIVÄKIRJASEURANTA .....	12
3.1	Viikko 1: 13.9.-17.9.2021 – jQuery .....	12
3.2	Viikko 2: 20.9.-24.9.2021 – PHP-ohjelmointi .....	15
3.3	Viikko 3: 27.9.-1.10.2021 – Google Analytics.....	18
3.4	Viikko 4: 4.10.-8.10.2021 – HTML-ohjelmointi.....	22
3.5	Viikko 5: 11.10.-15.10.2021 – PostgreSQL .....	25
3.6	Viikko 6: 18.10.-22.10.2021 – Ongelmanratkaisu / debugaus .....	28
3.7	Viikko 7: 25.10.-29.10.2021 – Git ja versionhallinta.....	31
3.8	Viikko 8: 1.11.-5.11.2021 – Huijarisyndrooma ja ammatillinen kasvu.....	34
3.9	Viikko 9: 8.11.-12.11.2021 – REST API.....	37
3.10	Viikko 10: 15.11.-19.11.2021 – Kommunikaatio etätyössä .....	40
4	POHDINTA .....	43
	LÄHTEET.....	45

# 1 JOHDANTO

Opinnäytetyö toteutetaan päiväkirjamuotoisena. Päiväkirjaa kirjoitetaan 10 viikon ajan päivittäin 13.9.2021 – 19.11.2021 välillä. Päiväkohtaisten kirjoitusten lisäksi joka viikko analysoidaan mennyttä viikkoa jonkun aiemmin määritellyn teeman pohjalta.

Päiväkirjamerkinnoissä tutustutaan päivittäiseen työhöni sisällönhallintajärjestelmän kehitystyön parissa. Työssä painottuu erityisesti PHP-ohjelmointi- ja ongelmanratkaisutaidot.

## 1.1 Yritys ja työympäristö

Liana Technologies on perustettu vuonna 2005 nimellä Koodiviidakko Oy. Se on Suomen suurin markkinoinnin ja viestinnän pilvipalveluja tuottava ohjelmistoyritys. Jo alkuun yrityksen visiona oli tuottaa markkinoinnin ja viestinnän työkaluja alan ammattilaisille. Lianan pääkonttori sijaitsee Oulussa, lisäksi toimipisteitä on Helsingissä, Tukholmassa, Pariisissa, Münchenissä, Dubaissa ja Hongkongissa. Lianan asiakkaita ovat markkinointia ja viestintää työkseen tekeviä ihmisiä. Lianan tavoitteena on tarjota heille parhaat mahdolliset työkalut. Liana@Cloud on kattava ohjelmistopaletti yrityksen viestintä- ja markkinointitiimeille. Liana@-työkalut on integroitu toisiinsa ja ne ovat räätälöitävissä asiakasyritysten markkinointitarpeiden mukaisiksi. Liana@Cloudiin kuuluu sähköposti-markkinointi, markkinoinnin automaatio, tiedotejakelu ja mediaseuranta, verkkosivut sekä verkkokauppa. Ohjelmistopaletit voidaan koota käyttäen seuraavia työkaluja:

- LianaCMS: alusta monenlaiselle sisällölle, esim. kuvat, blogit, videot, verkkolehdet
- LianaMailer: käyttäjäystävällinen uutiskirje-editori
- LianaAutomation: työkalu automatisoitujen sähköpostien lähettämiseen
- LianaCommerce: hakukoneoptimoitu verkkokauppa-alusta
- LianaMonitor: mediaseurantatyökalu julkaistuihin tiedotteisiin liittyvää seurantaa varten
- LianaPress: tehokas tiedotejakelupalvelu

## 1.2 Käsitteitä

Admin = Järjestelmänvalvoja. Esimerkiksi CMS:ssä on admin-puoli, jolla hallinnoidaan sivustoa.

Backend = Sivuston palvelinpuoli

Bugi = Virhe koodissa

Custom-moduuli = Yksittäiselle asiakkaalle tehty ratkaisu

Debuggaus = Virheiden etsiminen koodista

Frontend = Sivuston käyttäjälle näkyvä osio

GitLab = Verkkopalvelu, joka tarjoaa mm. versionhallintaa

Legacy-koodi = Vanhaa tai vanhentunutta koodia

Moduuli = Sivuston yksittäinen osio, esimerkiksi tuotelista, lomake

Slack = Erityisesti ohjelmistoyrityksille suunnattu pikaviestin- ja keskustelusovellus

Targetprocess = Projektinhallintatyökalu erityisesti ketteriin menetelmiin

Tiketti = TargetProcessiin tehty työmääräys

Upgrade Script = Koodi, jolla päivitetään esimerkiksi moduulien tarvitsemia tietokantatauluja

Visual Studio Code = Ohjelmointiin tarkoitettu tekstieditori

## 2 NYKYTILANTEEN KUVAUS

### 2.1 Oman nykytilanteen analyysi

Aloitin Liana Technologiesilla harjoittelijana toukokuussa 2021 ja varsinaisena työntekijänä olen ollut syyskuun 2021 alusta lähtien. Työskentelen CMS- eli sisällönhallintajärjestelmätiimissä, jonka vastuulla on julkaisujärjestelmän toimivuus. Työhöni kuuluu enimmäkseen bugien korjausta ja uusien ominaisuuksien kehittämistä sisällönhallintajärjestelmään. Lisäksi olen tehnyt pieniä custom-moduuleja asiakkaiden verkkosivustoille. Tehtävät annetaan niin sanottuina tiketteinä TargetProcessissa, jonne myös merkitään tehtävään käytetyt tunnit. Tikettejä voi tulla myynniltä (esim. asiakkaille tehtävät custom-moduulit), frontendin puolelta (esim. löydetyt bugit) tai muilta CMS-tiimiläisiltä (esim. uudet ominaisuudet, bugit). Kaikissa työtehtävissäni on olennaisena osaamisena PHP-ohjelmointitaito. Myös tietokantaosaaminen (PostgreSQL) on suuressa osassa työtehtäviä. Monissa tehtävissä tarvitaan lisäksi JavaScriptin ja HTML:n osaamista, ainakin perustaitoja. Lähes päivittäin tarvitsen myös Linuxin peruskäyttötaitoja. Päivittäin käytän myös GitLabia.

Bugien korjaamisen aloitan virheen toistamisella. Tutkin, miten kyseisen virheen saa toistettua omalla asennuksellani ja mietin, mistä se voisi johtua. Tämän jälkeen teen GitLabissa uuden branchin eli haaran, johon sitten siirryn Visual Studio Codessa. Koska koodikanta tuotteella on niin valtava, menee minulla vielä usein aikaa siihen, että löydän kannasta tarvittavat tiedostot ennen kuin pääsen varsinaisesti korjaamaan ongelmaa. Tähänkin saan toki apua, mutta oman oppimiseni kannalta olen todennut, että on parempi etsiä itse kuin kysyä aina muilta. Kun olen paikallistanut ongelman, kokeilen erilaisia tapoja korjata se ja testailen, mikä toimii parhaiten, ennen kuin annan tehtävän katselmoitavaksi. Näihin tehtäviin tarvitaan PHP-ohjelmointia ja yleensä myös tietokannan käyttöä.

Uusien ominaisuuksien kehittäminen alkaa tiketin user storyn pilkkomisella pienempiin tehtäviin, mikäli uusi ominaisuus on laaja. Jos tehtävä on pieni, on user story itsessään usein riittävä.

Tiketin perusteella lähdetään rakentamaan uusia ominaisuuksia. Bugikorjauksista poiketen uusia ominaisuuksia kehitettäessä tarvitaan usein laajempaa osaamista myös JavaScriptistä ja HTML:stä.

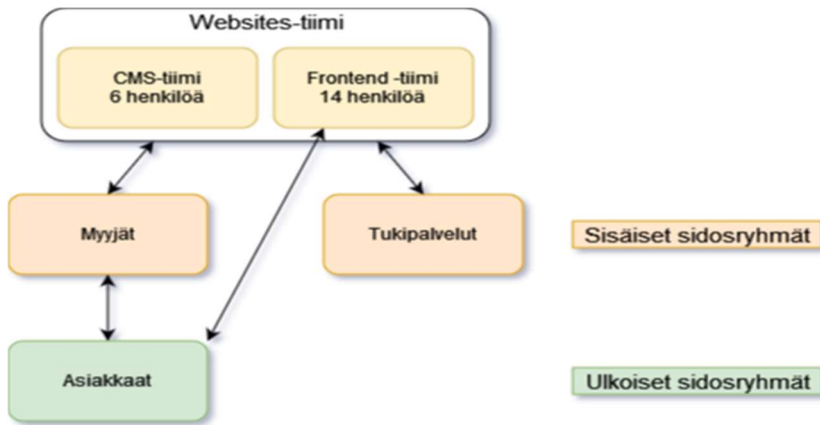


Taidoiltani olen vielä aloitteleva toimija. Tarvitsen ohjausta päästäkseni alkuun tehtävissä ja työskentelyni on vielä verrattain hidasta. Varsinaiset ohjelmointitaitoni ovat mielestäni perushyvät, mutta koska tuotteen koodikanta on niin suuri ja sitä on tehnyt lukuisat ohjelmoijat ennen minua, on välillä haastavaa ymmärtää koodin toimivuutta. Minulta ei onneksi vielä odoteta itsenäistä suoriutumista työtehtävissäni ja saan rauhassa opetella vielä harjoittelujakson jälkeenkin.

Olen vielä ohjelmointiurani alkuvaiheessa. Kehittymistä on vielä runsaasti edessä eikä se varmasti tällä alalla koskaan tule loppumaankaan. Olen utelias oppimaan uutta, enkä yleensä jätä käyttämättä tilaisuutta tutustua minulle tuntemattomiin asioihin. Tässä työtehtävässä suoriutumiseen minun tarvitsee oppia lisää Linuxin käyttöä. Tällä hetkellä osaan listata tiedostot ja siirtyä kansioista toiseen, mutta esimerkiksi uusien tiedostojen tekemiseen ja niiden siirtämiseen tarvitsen aina muistilappuja. Tähän minun täytyy panostaa, jotta työskentely olisi jatkossa sujuvampaa. Lisäksi tarvitsen varmuutta yleisesti ohjelmointiin sekä erityisesti JavaScriptin ja jQuery:n käyttöön. Suurin haasteeni lienee kuitenkin epätäydellisyyden sietämisessä. Minun on ollut aina vaikea näyttää kesken-eräistä koodia muille, koska siinä voi olla joku virhe. Vaadin liikaa täydellisyyttä itseltäni enkä tahdo hyväksyä virheitä. Tämä hidastaa monesti avun pyytämisen silloin, kun olen jumissa enkä tiedä mihin suuntaan jatkaisin työskentelyä. Tämä on suurin kehityshaasteeni seuraavien kuukausien aikana.

## **2.2 Työpaikan sidosryhmät**

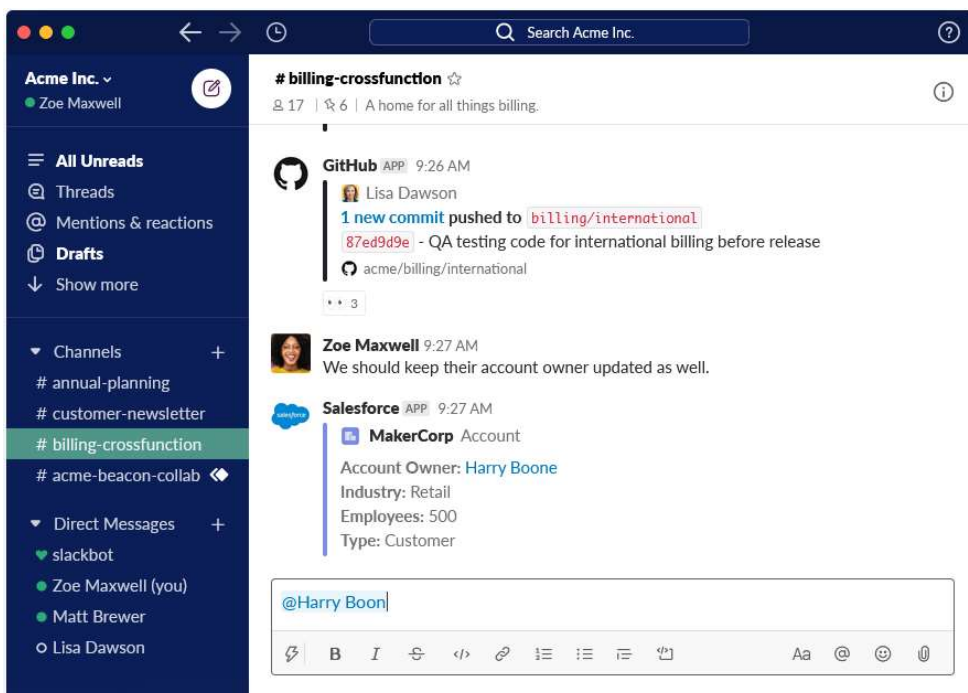
Työskentelen Websites-tiimissä, jonka alla työskentelee erikseen vielä CMS-tiimi, eli ohjelmistokehittäjien tiimi ja frontend-tiimi, joka tekee asiakkaille verkkosivuja. Minun oma tiimini on CMS-tiimi. Websites-tiimin kesken kommunikaatiota tapahtuu päivittäin, vaikka työskentelemmekin pääasiassa eri tehtävien parissa. Frontend-tiimin jäsenet ovat enemmän tekemisissä sekä myyjien että asiakkaiden kanssa, koska heillä työ on suoraan asiakkaille tehtävää. CMS-tiimillä on myös jonkin verran asiakkaille tehtäviä töitä, mutta niiden kehitys tapahtuu pääasiassa myyjien välityksellä, ellei kyseessä ole iso työ. Sisäisiin sidosryhmiin kuuluu lisäksi myös tukipalvelut, jonka työntekijät auttavat sekä asiakkaita että työntekijöitä ongelmien kohdatessa. Kuviossa 1 on esitetty sidosryhmät kaaviona.



Kuvio 1 Työpaikan sidosryhmät -kaavio

## 2.3 Vuorovaikutustaidot työpaikalla

Koronan edelleen hallitessa ihmisten arkea työskentelemme enimmäkseen kotitoimistoilla. Pääasiainen kommunikaatiokanavamme on Slack, missä on useita kanavia eri aihealueille (kuvio 2). Koko Websites-tiimillä on yhteinen kanava ja lisäksi backend-kehittäjillä on oma kanavansa, jossa käydään suurin osa meidän tiimiläisten keskusteluista. Lisäksi on olemassa LianaCMS:iin liittyen tukikanava, jossa apua kysyvät niin myynnin kuin markkinoinninkin puolen ihmiset. Tukikanavalla minun ei ole toistaiseksi tarvinnut osallistua keskusteluun. Olen siellä enemmänkin oppimassa toisten ongelmista. Ongelmiin vastailee enimmäkseen kokeneemmat kehittäjät.



Kuvio 2 Slackin esimerkinäkymä (Slack 2021)

Keskustelut Slackissä oman tiimin kesken ovat enimmäkseen juttelua työhön liittyvistä asioista. Kanavalla kysytään neuvoa toisilta, keskustellaan erilaisista vaihtoehtoista jonkun asian toteutukseen tai vain päivitellään jonkun ominaisuuden toimimattomuutta. On päiviä, jolloin kanava on hyvin hiljainen ja toisina päivinä sinne tulee paljonkin asiaa. Minulle itselleni kirjoittaminen on varsin luonteva kommunikointikeino. Saan hahmoteltua ajatukseni paljon ymmärrettävämpään muotoon kirjoitettuna kuin ääneen puhuttuna. Toki kirjoitetussa kommunikaatiossa on omat ongelmansa, kun ei kuule toisen ihmisen äänensävyä tai näe ilmeitä. On vaikea aistia viestin sävyä pelkästä tekstistä. Tämä on johtanut siihen, että oma persoonani ei varmasti pääse täysin näkyviin tiimissä pelätesäni toisten ymmärtävän huumorini väärin. Tilanne olisi varmasti toinen, jos olisin aloittanut jo ennen koronaa ja etätöitä sekä tuntisin työkaverini paremmin. Uskon tilanteen helpottavan, kun näemme toisiamme enemmän kasvokkain jatkossa. Olemme ottaneet tavoitteeksi nähdä perjantaisin toimistolla viikkopalaverin merkeissä. Tämä varmasti laajentaa kommunikointitapaa Slackinkin puolella monipuolisemmaksi myös minun osaltani.

Toistaiseksi minun ei ole tarvinnut olla vuorovaikutuksessa muiden kuin oman CMS-tiimini ja muiden työkavereiden kanssa. Joskus tulevaisuudessa voi olla mahdollista olla palaverissa myyjän ja asiakkaan kanssa, mutta toistaiseksi sellaista ei ole näköpiirissä. Minun vuorovaikutustaitoni ovat kuitenkin perushyvät, joten en koe asiakaspalvelutilanteita ongelmallisiksi.

## 3 PÄIVÄKIRJASEURANTA

### 3.1 Viikko 1: 13.9.-17.9.2021 – jQuery

#### **Maanantai 13.9.**

Päivän ohjelmassa on edellisellä viikolla alkaneen työtehtävän jatkamista. Kyseessä on erilaisten ominaisuuksien siirtämistä raahaamalla riviltä toiselle ja uuden sijainnin tallentamisesta tietokantaan. Päivän tavoitteena on ymmärtää jQuery-koodin toimintaa tuossa muutoksessa.

Päivä oli vähän rikkonainen johtuen siirtymisestä kotoa toimistolle kesken päivää. Huomaan tällaisen tapahtuman vaikuttavan omaan keskittymiskykyyn todella voimakkaasti sekä ennen että jälkeen siirtymän. Tästä häiriöstä huolimatta opin uutta jQuerystä. Sain taulukon rivit siirtymään raahaamalla, joskin niiden uuteen paikkaan asettaminen on vielä hieman haasteellista enkä saa tietoa vielä siirtymään tietokantaan asti. Huomenna jatkan saman ongelman parissa.

#### **Tiistai 14.9.**

Chromen DevTools on ollut minulle vähän vieras, enkä ole sitä kunnolla osannut käyttää. Tänään löysin siitä virheilmoituksia liittyen jQueryyn. Rivin raahaaminen ei edelleenkään toimi, mutta alan olla paremmin kärryillä siitä, missä vika piilee. Samantyyppinen ominaisuus on käytössä toisaalla CMS:ssä, joten olen katsonut siitä mallia tähän toisenlaiseen toteutukseen ja en vielä täysin ymmärrä sitä, mitä kaikkea siitä täytyy muuttaa, jotta se toimii tässä toiminnossa. Tänään olisi kannattanut pyytää apua jumitukseen, mutta päädyin edelleen yrittämään itse.

#### **Keskiviikko 15.9.**

Tänään tehtävässä tuli paljon edistystä. Taulukkorivit raahautuvat uusille paikoilleen "kahvasta" vetämällä ja tieto tallentuu myös tietokantaan. Löysin koodikannasta jQuery-toteutuksen, jossa tietoa muokataan niin, että käytössä oleva funktio saa tiedon siitä, onko rivi menossa ylös- vai alaspäin nuolinäppäimiä painettaessa. Tämän jälkeen löytyi myös toteutus sille, miten aiemmin on tehty tuo raahaamalla saadun uuden paikan paikkatieto funktiolle. Kun jQuery-koodi toimi, oli helppoa muuttaa PHP-koodi vastaamaan tiedon siirtymisestä tietokantaan.

## Torstai 16.9.

Eilisen läpimurron jälkeen tälle päivälle jäi enää pientä viilausta ulkoasuun. Kolumnit eivät osuneet aivan kohdalleen siirtoikonin lisäämisen jälkeen. Lisäksi kun siirtoikonista ottaa kiinni, rivi hyppää oikeaan reunaan, aivan kuin olisi otettu kiinni rivin vasemmasta päädystä. Nämä tuli korjattua aamupäivällä. (Kuvio 3.)

Ominaisuudet										
<span>+ Luo uusi ominaisuus</span> <span>+ Lisää olemassa oleva</span> <span>+ Tuo ominaisuudet</span>										
	Nimi	Tyyppi	Jaettu	Moniarvoinen	Mukana vienneissä	Vaadittu	Mukana hauissa	Kopioi kielestä	Järjestys	Poista
0040	uri-slug	Lyhyt teksti	Kyllä (1 muun kanssa)	Ei	Ei	Ei	Kyllä	Ei	+	↓
0053	lyhytnimi	Lyhyt teksti	Ei	Ei	Ei	Ei	Kyllä	Ei	+	↑
0024	Tuotehinta	Hinta	Ei	Ei	Ei	Ei	Kyllä	Ensisijainen kieli	+	↑
0025	tuotekuvaus	Teksti	Ei	Ei	Ei	Ei	Kyllä	Ensisijainen kieli	+	↑
0022	description	Lyhyt teksti	Ei	Ei	Ei	Kyllä	Kyllä	Ei	+	↑
0023	tuotokuva	Kuva	Ei	Ei	Ei	Ei	Kyllä	Ensisijainen kieli	+	↑
0054	toinen nimi	Lyhyt teksti	Ei	Ei	Ei	Ei	Kyllä	Ei	+	↑

Kuvio 3. Kuvakaappaus kategoriäeditorin järjestyksen muuttamisesta

Iltapäivällä tein korjauksia erääseen vanhempaan lisäominaisuuteen. Tuossa ominaisuudessa tietokantatauluun tulee yksi lisäkolumni, enkä ominaisuutta rakentaessa ollut huomannut tehdä upgrade scriptiä tietokannalle, jotta jo asennetut moduulit saavat uudet kolumnit tauluihinsa.

Loppupäivän suoritin Udemystä Linux-kurssia. Udeemy on verkossa toimiva kurssialusta, jossa on kurseja monista aihealueista kuten liiketoiminta, IT-ala, markkinointi, design ja stressinhallinta (Udeemy 2021).

## Perjantai 17.9.

Perjantaisin on aina tiimin viikkopalaveri, jossa käydään läpi jokaisen tiimin jäsenen kulunut viikko ja suunnitelmat seuraavalle viikolle. Lisäksi tapaamisessa käydään läpi mahdollisia ongelmia ja niiden ratkaisuvaihtoehtoja. Palaveri kestää yleensä n. tunnin. Tähän asti viikkopalaverit on pidetty etäpalavereina Google Meetin kautta, mutta viime viikolla sovimme tulevamme jatkossa perjantaisin toimistolle, jotta voimme tavata kasvoistusten, joten tällä viikolla näimme toimistolla. Mitään erityistä tehtävää minulle ei tälle päivää ollut, joten opiskelin Udemystä PostgreSQL-asioita.

## Viikkoanalyysi – jQuery

Tällä viikolla painin enimmäkseen jQueryn parissa. jQuery on JavaScript-kirjasto, jonka tarkoitus on helpottaa JavaScriptin käyttöä verkkosivuilla. jQueryn kehittäjä on John Resig. Ensimmäinen julkaisu tapahtui elokuussa 2006, minkä jälkeen suosittu kirjasto on jatkanut kasvuaan. Sen vaikutusta verkkosivujen kehittämiseen ei voida aliarvioida. jQuery on tehnyt JavaScriptistä saavutettavamman keskinkertaiselle kehittäjälle. Sen avulla on esimerkiksi helpompi valita elementtejä DOM-puusta CSS-selektoreiden avulla, suorittaa tehtäviä ja hallinnoida tapahtumia verkkosivulla vain yhdellä koodirivillä verrattuna tavalliseen JavaScriptiin, jolla koodia pitäisi kirjoittaa paljon pidemmästi. Verkkosivua katsoessa näkee suuren määrän elementtejä lisättyinä yhteen ja muotoiltuna käyttäjän näkemään muotoon. Jotta voisi päästä käsiksi noihin elementteihin poistaakseen, lisätäkseen ja muokatakseen niitä, tarvitaan rajapinta. DOM-puu määrittää miten dokumentissa olevat elementit välittävät tietoa toisilleen ja kuinka näihin elementteihin voidaan viitata. DOM mahdollistaa myös selaimen tapahtumien kiinnittämisen, kuten linkin klikkaamisen, lomakkeen lähettämisen tai sivun vierittämisen. (Franklin & Ferguson 2017, luku 2.)

Opintojen ensimmäisenä lukukautena opiskelin jQueryn perusteita ja silloin koin hallitsevani ne ihan kohtuullisen hyvin. Parin vuoden aikana asiat olivat kuitenkin painuneet aivoissani taka-alalle ja paljon piti muistella asioita Googlen avulla. Tällä viikolla tehdyssä työssä keskityin lähinnä jQueryn sortable-toimintoon ja sen toimintatapaan. Huomaan, että alkuviikkoon verrattuna jQuery tuntuu taas paljon tutummalta ja jatkossa sen pariin hyppääminen ei varmasti ole yhtä hankalaa ja aikaavievää.

Koodipuolen kehittymisen lisäksi opin käyttämään paremmin Chromen DevToolsia. DevTools on selaimen sisällytetty debugaus-työkalu, joka on korvaamaton työkalu JavaScript-koodin kirjoittajalle. Suosituin metodi on `console.log()`, joka tulostaa dataa konsoliin luettavaksi. (Franklin & Ferguson 2017, luku 1.) Tätä käytin itsekin melko paljon viikon aikana tutkiessani funktioiden toimintaa ja niistä tulevaa tietoa.

Viikon aikana ilmenneitä ongelmia oli enimmäkseen koodipuolella. Olen melko tottunut tiedonhakija ja oikeastaan kaikkiin ongelmiin löysinkin ratkaisun melko nopeasti joko Googlen avulla tai etsimällä vastaavia toimintoja muualta valmiista koodikannasta.

## 3.2 Viikko 2: 20.9.-24.9.2021 – PHP-ohjelmointi

### Maanantai 20.9.

Alkupäivän suoritin PostgreSQL-kurssia Udemysta. Kurssilla käytiin läpi alikyselyitä ja kytettyjä alikyselyitä. Alikyselyt itsessään ovat olleet minulle tuttuja, mutta kytettyjä alikyselyitä oli hyvä kerrata. Näitä aion kokeilla myös oman CMS-asennukseni parissa, kunhan sellaiseen on sopiva aika.

Lounaan jälkeen aloitin uuden tehtävän parissa. Edellinen ”drag and drop” -tehtävä onnistui niin hyvin, että minulle osoitettiin uusi vastaava tehtävä. Tämä on hyvin samantapainen tehtävä kuin edellinenkin ja se tulikin pääosin valmiiksi jo parin tunnin työn jälkeen. Sekä tähän että samalla edelliseen tehtävään ajattelin kuitenkin tehdä pienen ulkoasuparannuksen. Kun siirrettäviä rivejä on vain yksi, eivät siirtonuolet ole näkyvissä, mutta siirtoikoni on. Tuon ominaisuuden toiminnallisuus on tehty jQueryllä, mutta en vielä ole löytänyt oikeaa tiedostoa. Se jää huomisen työpäivän tehtäväksi.

### Tiistai 21.9.

Eilisen ”drag and drop” -tehtävän viimeistelyä. Pienellä avustuksella löysin oikean tiedoston, johon muutaman rivin lisääminen poisti siirtoikonin näkyvistä, kun rivejä on vain yksi. Lisäksi minulle opetettiin, miten JavaScript-tiedosto pakataan pienemmäksi tuotantoasennuksia varten. Meidän omat testiasennuksemme käyttävät pakkaamattomia tiedostoja, jotka sitten muutosten jälkeen ajetaan terminaalissa skriptillä pienemmiksi. Tämän jälkeen tein vielä yhden ”drag and drop” -ominaisuuden. Tämä alkaa tuntua jo helpolta, kun tietää mitä pitää muuttaa mistäkin ja tähän työhön menikin enää vain alle kaksi tuntia. Seuraavaksi työksi sainkin vähän erilaisen työn. Suurimmassa osassa CMS:n moduuleita on mahdollisuus kopioida toisen samantyyppisen moduulin asetukset ja ulkoasu, mutta tuotehakulomakemoduulista se puuttuu. Sain nyt tehtäväksi lisätä tuon kopiointimahdollisuuden. Tämä vaatii vähän muiden moduulien koodien läpikäyntiä, jotta näen, miten kopiointi on toteutettu. Työssä tarvitsen enimmäkseen PHP-ohjelmointia. Työ alkoi kuten muutkin tehtävät: loin Gitlabissa uuden branchin eli haaran, jotta voin turvallisesti muokata koodia sotkematta mitään toimivaa. Paljon muuta en tähän tehtävään vielä saanut tehtyä.

## **Keskiviikko 22.9.**

Päivän tavoitteena on selvittää, miten asetusten kopiointi on toteutettu muissa moduuleissa, jotta voin muokata myös tuotehakulomakemoduuliin kopiointin. Arvioin tähän kuluvan lähes koko päivän, koska kyseessä olevat moduulit ovat minulle uusia, enkä ole näiden toimintaan aiemmin perehtynyt.

Ongelma ratkesikin odotettua nopeammin tällä kertaa, vaikka tuotehakulomakemoduuli oli rakennettu toisen moduulin päälle, joka taas oli rakennettu toisen luokan päälle. Tehtävän suoritusnopeuteen vaikutti se, että olen oppinut käyttämään hyväkseni Visual Studio Coden hakua kaikista tiedostoista. Olen huomannut, että parhaiten löytää vastaavat toiminnot, kun hakee funktionimellä muita toteutuksia. Päivä alkoi tänään vähän tahmeasti, kun edessä oleva tehtävä tuntui aika vaikealta, mutta loppui onnistumisen kokemukseen. Saan pikkuhiljaa itselleni uskoa siihen, että minäkin voin oikeasti olla hyvä koodari.

## **Torstai 23.9.**

Eilen ajattelin, että ongelma ratkesi niin helposti, että eihän se voi olla tottakaan. Eikä se ollutkaan. Tänään uutta ominaisuutta testaillessa huomasin, että osa moduulin asetuksista ei kopioitu. Etsin syytä ensin koodista, kunnes huomasin katsoa tietokannan taulut ja sieltä löysin syyn ongelmaan. Kategoria-asetukset olivatkin eri taulussa kuin muut asetukset ja siitä tauluryhmästä ei ole koodattu hakua tuohon valmiiseen kopiointitoimintoon. En vielä keksinyt tapaa saada sitä siististi koodattua. Iltapäivästä iskikin melkoinen väsymys legacy-koodiin ja siirryin viimeiseksi tunniksi Udemyn PostgreSQL-kurssin pariin. Kurssilla käsiteltiin edelleen kytkettyjä alikyselyitä sekä DISTINCT-määrettä.

## **Perjantai 24.9.**

Tänään oli jälleen viikkopalaveri, jossa käsiteltiin mm. ensi viikon CMS-päivitystä. Muun ajan jatkoin kategoriaoingelman ratkaisemista. Huomasin, että kategorioita käytetään kyllä muidenkin moduulien kopiointissa, mutta tässä moduulissa tarvitaan lisäksi vielä kategorian ominaisuudet eli propertyt. Niiden kopioimiseen minun täytyy kirjoittaa uutta koodia, jota aloittelin tänään. Kokonaan uuden koodin kirjoittaminen tuntui hankalalta, oli vaikea päästä alkuun. Päätin lopulta, että teen muutamia kokeiluja välittämättä siitä toimivatko ne ja pääsinkin hyvin alkuun hommassa, mutta



valmiiksi se ei vieläkään tullut. Valmiina ollut koodi on monimutkaista ja polveilevaa ja siihen on vaikea päästä kärryille, joten uuden koodin kirjoittaminen sen rinnalle on hankalaa.

## **Viikkoanalyysi – PHP-ohjelmointi**

Tällä viikolla minulla oli aika hajanaisia tehtäviä. Alkuvuikosta opiskelin tietokanta-asioita ja tein jQueryllä drag and drop -ominaisuuksia, kun taas loppuviikon tehtäviin tarvittiin enemmänkin PHP-ohjelmointia. Oma PHP-osaamiseni on vakaalla pohjalla, mutta opittavaa on vielä paljon. Tuotteen koodikanta on peräisin monen vuoden takaa ja sen kirjoitustapa on erilaista kuin mihin opiskeluprojekteissa totuin, joten sen lukeminen on ollut haastavaa ja vienyt aikaa. Lisäksi se on paljon monimutkaisempaa kuin olin osannut kuvitellaakaan. Tiedostoja ja luokkia on paljon ja luokkia on rakennettu toisten luokkien päälle. Tämä tekee koodista monimutkaista ja ongelmien syyn löytämisen välillä hankalaksi. Tämänkin viikon tehtävässä tuotehakulomake on rakennettu kontaktihakulomakkeen päälle, koska tuotteet ja kontaktit ovat koodissa samantyyppisiä kokonaisuuksia. Tuotehakulomakkeen kloonaus ei toiminut, koska verkkosivusto yritti etsiä sen moduulin nimeen liitettyä ulkoasupohjaa, jota ei ollut olemassa, koska se on rakennettu kontaktihakulomakkeen päälle.

Koska tämän viikon suurimmat haasteet keskittyivät PHP-ohjelmointiin, päätin ottaa sen viikkoanalyysin teemaksi. PHP on yksinkertainen mutta tehokas kieli, joka on suunniteltu HTML-sisällön luomiseen. Se on myös hyvin joustava ohjelmointikieli eikä sen käyttö esimerkiksi rajoitu vain HTML-tai muiden tekstitiedostojen tulostamiseen vaan sillä voidaan luoda mikä tahansa asiakirjamuoto. PHP:ssä on sisäänrakennettu tuki PDF-tiedostojen sekä GIF-, JPEG- ja PNG-kuvien luomiseen. Yksi PHP:n merkittävimmistä ominaisuuksista on kuitenkin sen laaja tuki tietokannoille. PHP tukee kaikkia tärkeimpiä tietokantoja, kuten MySQL, PostgreSQL ja Oracle sekä jopa joitain NoSQL-tyyppisiä tietokantoja kuten MongoDB. PHP tarjoaa myös laajan koodikirjaston, joka suorittaa yleisiä tehtäviä tietojen keräämisestä tietokannasta virheiden käsittelyyn. (Tatroe & MacIntyre, 2020. Luku 1.)

PHP tukee myös olio-ohjelmointia (object-oriented programming, OOP). Olio-ohjelmointi tunnustaa tiedon ja siihen liittyvän koodin välisen perusyhteyden ja antaa ohjelmoijan suunnitella ja toteuttaa ohjelman sen yhteyden ympärille. Esimerkiksi ilmoitustaulujärjestelmällä on useita käyttäjiä, jotka kirjoittavat viestejä viestiketjuihin. Objekti koodin ja datan yhdistelmänä on modulaarinen yksikkö

sovellusten kehittämiseen ja koodin uudelleenkäyttöön. Oliot ovat myös yhteydessä toisiinsa: käyttäjäolio tietää, mitkä viestit se on kirjoittanut ja viesti tietää, mihin viestiketjuun se kuuluu. Viestillä voi myös olla koodi uuden viestin kirjoittamiseen, vastaamisen jo olemassa olevaan viestiin ja viestien näyttämiseen. (Tatroe & MacIntyre, 2020. Luku 6.) Liana CMS:ssä on käytetty myös olio-ohjelmointia, mm. tuotteet ja kontaktit ovat olioita, joilla on omat funktionsa ja ominaisuutensa.

Viikon alkupuolella opin tuottamaan jQueryä sujuvammin ja nopeammin. Tässä auttoi useampi samantyyppinen työtehtävä. PHP-ohjelmoinnissa en koe juurikaan kehittyneeni, vaikka pieniä ahaa-elämyksiä tulikin koettua lähinnä tuon CMS:n koodikannan parissa. Kehitys lienee sen verran hidasta, ettei sitä viikon aikana ehdi huomata.

### 3.3 Viikko 3: 27.9.-1.10.2021 – Google Analytics

#### **Maanantai 27.9.**

Jatkoin viimeviikkoisen ongelman parissa. Kategorioiden kopioimiseen käytettiin viittä eri funktiota, joissa jokaisessa tietoa muokattiin. Tämän vuoksi oli hankala saada tuotehakumoduulin tieto samaan muotoon, missä se tallennetaan, koska se poikkeaa muista moduuleista. Useiden epäonnistuneiden kokeilujen jälkeen löysin oikean tavan. Käytin apuna konsoliin tulostamista, joka ei mielestäni PHP:llä ole yhtä helppoa kuin JavaScriptillä. Teinkin niin, että tulostin PHP:llä JavaScriptiä, joka puolestaan antoi tietoa konsoliin: `echo "<script> console.log(" . json_encode($props) . ")</script>";`

Kun sain sekä kategoriat että kategorian ominaisuudet tallennettua myös kopioituun moduuliin, muokkasin ulkoasua niin, että kopioidun moduulin asetuksia ei pysty käyttäjä muokkaamaan, koska ne tulevat toisesta moduulista. Tämä vaati hieman jQueryllä säätämistä, mutta se meni jo helpommin kuin viime viikolla.

#### **Tiistai 28.9.**

Olen ottanut tavaksi testata aamusta tekemisiäni, mikäli jokin tehtävä on tullut valmiiksi edellisenä iltapäivänä. Tänään siis testasin eilen valmistunutta moduulin asetusten, kategorioiden, ominaisuuksien ja ulkoasun kopiointia. Vaikutti toimivalta, joten siirryin seuraavan työtehtävän pariin. LianaCMS käyttää Googlen analytiikkatoimintoja ja admin-etusivulla näkyy analytiikan tuomat tiedot.

Tässä on ollut sellainen ongelma, että mikäli analytiikkakytkökset jostain syystä katkeavat esimerkiksi Google-tilin salasanan vaihdon takia, ei LianaCMS:n ohjauspaneelissa näy siitä minkäänlaista ilmoitusta, vaan näyttää siltä kuin analytiikka toimisi, vaikka dataa ei tule. Tähän on tarkoitus tehdä jonkinlainen ilmoitus, että käyttäjä huomaa tarkistaa kytkennän. Löysin ulkoasusta nopeasti ne paikat, joihin virheilmoitus laitetaan. Enemmän ongelmia tuottaa se, miten saan ulkoasutemplatelle tuotua tiedon siitä, onko virheitä olemassa. Se jää vielä huomisen tutkittavaksi, kun ei tänään vielä selvinnyt.

### **Keskiviikko 29.9.**

Jatkoin eilen aloitettua analytiikkaongelmatehtävää. Tuntui, että tehtävä ei edennyt ollenkaan, vaikka työskentelin sen parissa monta tuntia. Todellisuudessa selvitin monta eri toimimatonta tapaa hoitaa asia. Tämäkin on kai jonkinlaista edistymistä, vaikka se ei aina siltä tunnukaan. Iltapäivästä etenemätön työ alkoi puuduttaa ja totesin, että parempi jättää ongelma hautumaan takaraivoon ja tehdä jotain muuta, joten opiskelin Udemystä lisää PostgreSQL:ää. Siinä käytiin läpi vielä vähän erilaisia hakuehtoja ja asennettiin omalle koneelle PostgreSQL, kun tähän asti on käytetty verkossa olevaan palvelua.

### **Torstai 30.9.**

Jatkoin samaa tehtävää edelleen, eikä varsinaista edistymistä tullut tänäänkään näkyville. Tutkin itse tehtävän lisäksi myös Google Analyticsin toimintaa, jotta ymmärtäisin paremmin, miten se toimii. Kovin paljoa viisaammaksi en tainnut tulla. Analytiikkaongelman lisäksi valmistauduin huomiiseen kehityskeskusteluun miettimällä aiheita, joista haluaisin keskustella. Iltapäivästä katsoin myös Udemystä PostgreSQL-kurssia. Aiheena oli tänään erilaiset tietotyypit; erityisesti numerotyypit käytiin tarkasti läpi. Uutta tietoa tuli minulle erilaisista päivämäärä- ja intervalli tietotyypeistä. Lisäksi kurssilla aloitettiin validointiasioita.

### **Perjantai 1.10.**

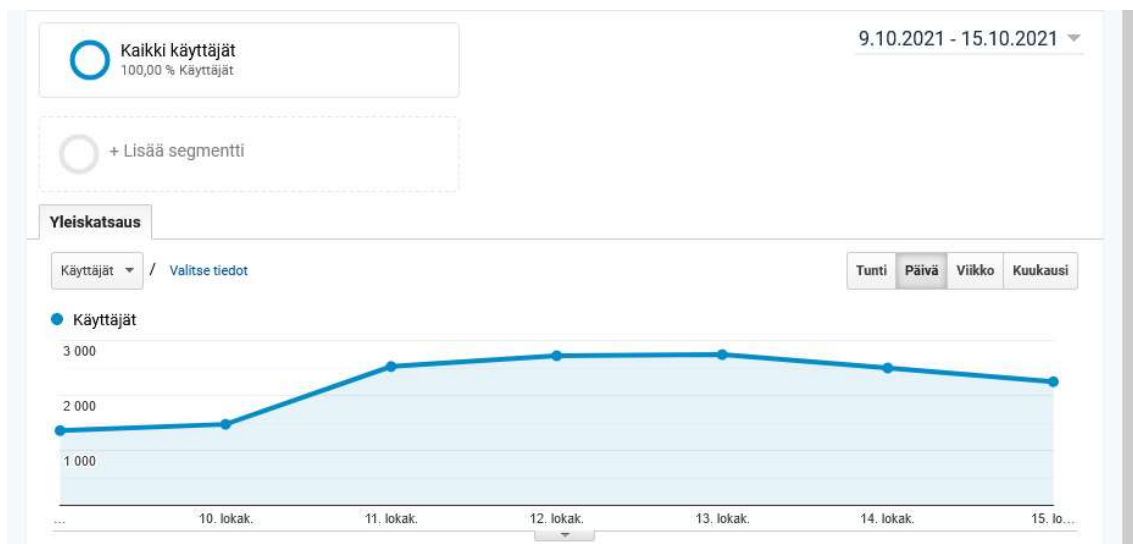
Perjantai, eli toimisto- ja viikkopalaveripäivä. Tänään oli myös ensimmäinen kehityskeskusteluni ja sain oikein hyvää palautetta sekä yleisestä työskentelystä että erityisesti hyvästä koodista. Työtehtävistä jatkoin samaa Googlen analytiikkaongelmaa. Tänään tapahtui asioissa sikäli edistymistä,

että tajusin viimein tehdä tietokannan analytiikka-asetustauluun uuden kolumnin virhetiedolle. Tiedon saa sieltä nyt haettua ja tuotua tiedon templatelle, mutta edelleen painin tiedon tallentamisen kanssa. Analytiikkatoiminnot on jaettu useisiin eri luokkiin ja funktio, jossa virheitä haetaan, sijaitsee luokassa, jossa asetustauluun ei kosketa lainkaan. Minun täytyy siis sinne rakentaa tietokantayhteys alusta asti. Siihen menee aikaa, jotta saan sen yhteneväiseksi muun koodin kanssa.

## Viikkoanalyysi – Google Analytics

Tämä viikko kului pääasiassa analytiikkaongelman parissa. Välillä tuntui, että työ ei etene ollenkaan ja välillä se otti selkeitä edistysaskeleita. Google Analytics on ollut minulle itselleni varsin vieras työkalu, mutta sen tuntemisesta on varmasti jatkossakin apua vastaavien tehtävien parissa.

Google Analytics on verkkosivujen analytiikkatyökalu, joka tarjoaa kattavat digitaaliset työkalut, joiden avulla voi ymmärtää verkkosivuston käyttäjiä ja arvioida sivustoa sekä sen sisältöä ja tuotteita. Analyticsin avulla voi nähdä, kuinka paljon verkkosivuilla on kävijöitä, millä sivuilla käyttäjät viettävät eniten aikaa ja millä sivuilta he poistuvat sivustolta. Nämä ovat tärkeitä tietoja verkkosivuja kehitettäessä. (Google Marketing Platform 2021.) (Kuviot 4 ja 5.)



Kuvio 4 Google Analytics Demo-tilin kävijämäärät (Google Analytics 2021).

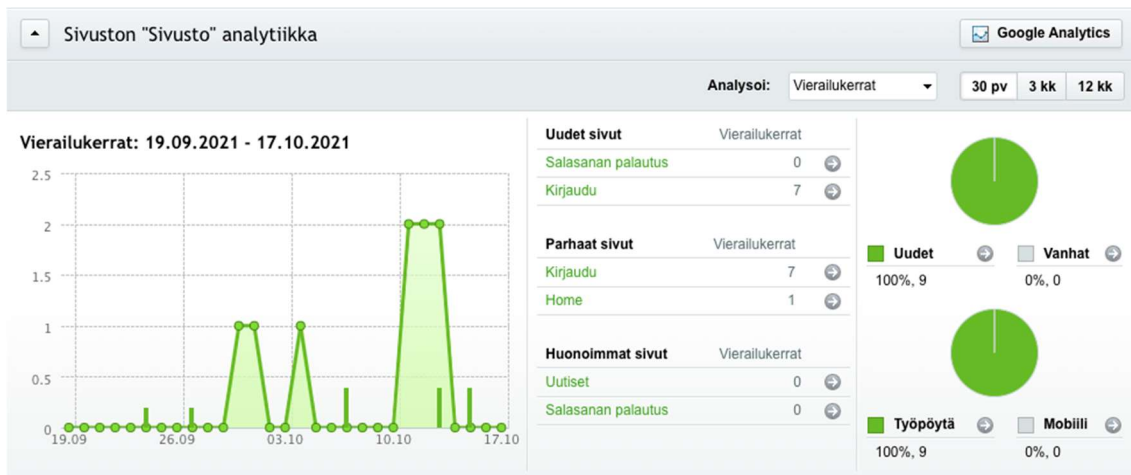
Sivu	Sivun katselut	Sivun arvo
/home	13 267	1,90 \$
/basket.html	5 636	23,15 \$
/google+redesign/apparel/mens/quickview	5 302	11,38 \$
/store.html	4 496	12,27 \$
/google+redesign/apparel/mens	3 903	6,68 \$
/store.html/quickview	2 843	11,86 \$
/signin.html	2 611	20,88 \$
/google+redesign/new	2 606	6,96 \$
/google+redesign/new/quickview	2 073	13,58 \$
/google+redesign/apparel	2 042	3,58 \$

7 viime päivän aikana ▼ [SIVUT-RAPORTTI >](#)

Kuvio 5 Suosituimmat sivut Demo-tilillä (Google Analytics 2021)

Google Analyticsin etusivulla voi tehdä nopean katsauksen käyttäjiin ja alasivuilta löytää yksityiskohtaisempaa tietoa. Sieltä löytää hyvin tarkkoja tietoja sivuston kävijöistä, kuten esimerkiksi vierailijoiden demografisia tietoja (ikä, sukupuoli) sekä millä laitteella ja selaimella kävijät käyvät sivustolla. Tarkastelujaksoa voi muokata ja eri ajanjaksoja vertailla toisiinsa. Tämä on hyödyllistä etenkin, jos sivustoon on tehty rakenteellisia tai sisältömuutoksia. Käyttäytymisvälilehdeltä löytää tietoja siitä, mikä on ollut kävijöiden aloitussivu, miten kävijät toimivat verkkosivustolla sekä tiedon mikäli kävijä poistuu heti ensimmäiseltä vierailemaltaan sivulta. (Orpana 2017.)

LianaCMS:n tulee Google Analyticsiltä tiedot kävijämääristä sekä siitä, mitkä sivut ovat olleet suosituimpia. Näistä tiedoista rakennetaan kaaviot admin-näkymän etusivulle (kuvio 6). Lisäksi tiedon suosituimmista sivuista saa liitettyä verkkosivuille haluttuun paikkaan omalla moduulillaan.



Kuvio 6 Kuvakaappaus sivuston analytiikkatiedoista

Tämä viikko oli hyvin turhauttava, koska monesti tuntui, ettei työ edennyt juuri ollenkaan. Viikon suurin oppi oli mahdollisesti se, että vaikka työ ei tunnu edistyvän, niin siltikin on löytänyt monta toimimatonta ratkaisua ongelmaan, eikä niitä siis tarvitse enää kokeilla uudelleen. Epäilin itseäni ja taitojani moneen kertaan viikon aikana. Olen huomannut, että kun käyn epävarmaksi, työnteko hidastuu ajatellessani, etten osaa mitään. Tästä syntyy ikävä kierre, ellen saa sitä ajoissa katkaistua. Onneksi voin tauottaa varsinaisten tehtävien tekemistä opiskelemalla välillä muuta ja jatkaa taas seuraavana päivänä paremmalla mielellä. Tällä kertaa perjantain kehityskeskustelun kehut tulivat aivan oikeaan hetkeen ja pääsin taas jaloilleni epävarmuuteni kanssa. Palautetta saan onneksi muutenkin usein katselmoinnin yhteydessä. Se auttaa usein selättämään epäilykseni.

### 3.4 Viikko 4: 4.10.-8.10.2021 – HTML-ohjelmointi

#### Maanantai 4.10.

Tämänkin päivän ohjelmassa oli analytiikkatietojen ongelmien korjausta. Perjantaina jäin ongelmassani siihen vaiheeseen, että olisi pitänyt tallentaa virhetiedot tietokantaan. Aamupäivän tutkin miten asiat on koodissa tehty ja päädyin siihen, että tallennan tiedon muuttujaan, jonka sitten haen toisesta luokasta ja tallennan siellä tietokantaan. Analytiikkatiedot haetaan vain kerran vuorokaudessa enkä vielä saanut selville, miten tämän voi kiertää, joten lopullista testausta en päässyt tekemään vielä. Se jäänee huomisen ongelmaksi.

## **Tiistai 5.10.**

Eilen työt lopettaessa pääsin siihen vaiheeseen, jossa virheiden pitäisi tulla näkyville halutusti etusivulle ja analytiikka-asetussivulle. Aamulla sivuja aukoessa huomasinkin, että homma toimii! Viilasinkin vielä koodia paremmaksi sekä lisäsin etusivun varoitusikoniin linkin asetussivulle. Olin jo lopettelemassa työtä, kun huomasin, että uusien analytiikka-asetusten tallentaminen ei poista varoitusmerkintää. Tämän ongelman korjaamiseen sainkin kulumaan loppupäivän. Työ lienee huomenna valmis, kunhan vielä kertaalleen testailen kaiken toimivaksi.

## **Keskiviikko 6.10.**

Analytiikkatehtävä tuli testailtua toimivaksi heti aamusta. Totesin vain, että eilinen työ tallentamisen kanssa oli turhaa, koska ei varoitusmerkintä saa poistua pelkästään Tallenna-painiketta painamalla, vaan vanhat analytiikka-asetukset pitää poistaa ja lisätä uudet, jotta virheilmoitus saa poistua. Ja kun asetukset poistaa, poistuu myös virhetieto tietokannasta.

Seuraava tehtävä onkin pieni bugikorjaus. Sivustolle voi tallentaa uudelleenohjauksia, eli käyttäjän syöttämästä osoitteesta siirrytään ennalta määriteltyyn toiseen osoitteeseen. Tässä on ollut sellainen ongelma, että järjestelmään ei ole voinut tallentaa sellaista osoitetta, jossa on iso kirjain, kun järjestelmä korjaa isot kirjaimet aina pieniksi. Tämäkin toiminto on tehty monen mutkan kautta, joten toiminnon alkulähde pitää paikallistaa ja miettiä, miten se kierretään ilman vaikutusta kaikkiin muihin tuota funktiota käyttäviin toimintoihin.

## **Torstai 7.10.**

Eilen aloitettu bugikorjaus tuli hoidettua loppuun iltapäivästä. Funktiota, jolla validoidaan linkit, käytetään suoraan vain aivan muutamassa paikassa, joten poistin sieltä isojen kirjainten poiston ja siirsin sen niihin funktioihin, joissa sitä tarvitaan.

Seuraavaksi aloitin salasananpalauttamismoduulin muokkaamisen. Tässä täytyisi muokata moduulia niin, että sivustolta lähtevää salasananvaihtosähköpostia voisi muokata myös HTML:llä. Tällä hetkellä sähköpostia voi muokata ainoastaan tekstien osalta. Tässä voi ottaa mallia lomakkeiden sähköposteista, mutta lomakekoodit ovat niin monimutkaisia ja sekavia, että sieltä kaiveleminen vie aikansa.

## **Perjantai 8.10.**

Tänään olin perjantaista huolimatta kotitoimistolla. Viikkopalaveri hoitui siis etänä. Palaverissa puhuttiin mm. testien tekemisestä, jota minulle ei vielä ollut opetettu. Palaverin jälkeen eräs tiimiläinen näytti minulle, miten testit ajetaan. Myös niiden logiikasta oli puhetta. Käytännössä testeihin kirjoitetaan, mikä on haluttu tulos ja testataan malliesimerkillä, että tulos on toivottu.

Viikkopalaverin ja testien lisäksi jatkoin eilen aloittamaani salasananpalautusmoduulin muok-  
kausta. Etenemistä ei vielä suuresti tullut.

## **Viikkoanalyysi – HTML-ohjelmointi**

Vaikka olenkin tekemisissä pääasiassa PHP- ja tietokantaohjelmoinnin parissa, on kuitenkin tärkeä osata myös HTML-ohjelmointia ymmärtääkseen sivustojen toimintaa. Itselleni on HTML:n perus-  
asiat tuttuja, mutta myönnän, että syvällisempi tutustuminen HTML:ään ei olisi pahitteeksi.

HTML tulee sanoista HyperText Markup Language. Siitä on ollut viisi versiota verkon perustamisen jälkeen ja kielen kehitystä valvoo World Wide Web Consortium (W3C) -niminen organisaatio. Tällä hetkellä uusin versio on nimeltään HTML5, edellinen tunnettiin nimellä HTML 4.01 vuodelta 1999. Nimestäkin voi päätellä HTML:n olevan merkintäkieli (markup language). Se saattaa kuulostaa monimutkaiselta, mutta jos ajattelee tekstinkäsittelyohjelman tyylejä, niin huomaa HTML:n toimivan hyvin samalla tavalla. Tekstissä voi käyttää esimerkiksi otsikkotyylä, jossa on isompi fontti kuin muussa tekstissä. Kappaleet erotellaan sekä tekstissä että HTML:ssä. Tämän merkitsemisen tarkoitus on tarjota asiakirjan lukijalle helposti ymmärrettävä rakenne, kuten myös verkkosivun käyttäjälle helppo rakenne näytöllä. Tekstinkäsittelyohjelmista poiketen HTML:ssä ei ole tyylejä vaan tageja eli tunnisteita. Tunnisteilla merkitään dokumentin elementtirakenne, jonka osia ovat mm. otsikko, kappale, taulukko ja lista. Verkkoselaimet osaavat sitten esittää dokumentin samantyyppisesti kuin tekstinkäsittelyohjelma - pääotsikot suuremmalla fontilla, kappaleet eriteltynä ja listassa luettelomerkit rivin alussa. HTML:n kirjoittamiseen ei tarvitse välttämättä mitään erityistä koodieditoria, vaan tavallinen tekstinkäsittelyohjelma riittää, kunhan dokumentin tallentaa .html- tai .htm-päätteellä. (Larsen 2013. Luku 1.)



Elementit koostuvat yleensä kahdesta tunnisteesta: alku- ja sulkutunnisteesta. Sulkutunnisteen tunnistaa vinoviivasta, jollaista ei alkutunnisteessa ole. Attribuutit antavat lisätietoja elementin sisällöstä. Ne näkyvät elementin avaustunnisteessa ja koostuvat kahdesta osasta: nimi ja arvo, jotka on erotettu yhtäläisyysmerkillä. Attribuutin nimi osoittaa, millaisia lisätietoja annat elementin sisällöstä. Arvo puolestaan on attribuutin tieto tai asetus. Se tulee laittaa lainausmerkkeihin. Eri attribuuteilla voi olla eri arvoja. (Duckett 2011. Luku 1.)

Tällä viikolla en huomannut suurta edistymistä osaamisessani. Luulen, että kehittyminen on sen verran hidasta, että sitä ei viikossa ehdi nähdä. Sain kuitenkin kaksi tehtävää valmiiksi asti ja kolmannenkin alulle.

### **3.5 Viikko 5: 11.10.-15.10.2021 – PostgreSQL**

#### **Maanantai 11.10.**

Tänään oli lyhyempi päivä työhöntulotarkastuksen takia. Jatkoin viime viikolla aloittamaani salasanpalautusmoduulia. Yleensä kirjoittelen itselleni käsin muistiinpanoja perjantaina maanantaita varten, mutta viime viikolla se unohtui tehdä. Alkuun pääseminen oli siis hieman hankalaa tänään. Sain kuitenkin kaiveltua koodista esille sen, missä ja suunnilleen myös miten lomakkeilla tehdään HTML-muokkaus sähköposteihin, joten huomenna on tästä helppo jatkaa.

Lisäksi tein muutoksia viime viikolla valmistuneeseen uudelleenohjaustehtävään. Huomasin, että testejä ajaessa tulee virheilmoitus isoista kirjaimista. Varmistin tiimikaverilta, voinko muokata testejä niin, että virheilmoitusta ei tule, vai pitääkö tuo tehtävä tehdä jollain toisella tapaa. Sain varmistuksen omalle toteutukselleni ja poistin yhden rivin testeistä.

#### **Tiistai 12.10**

Jatkoin edelleen edellistä tehtävää. Tänään tein muokkauksia tietokantaan HTML-sähköpostia varten. Lisäsin kentät HTML-viesteille sekä asetuksen, josta näkee, käytetäänkö HTML-muotoisia sähköposteja. Myöhemmin luin tiketin vielä tarkemmin ja totesin, että tarkoituksena ei ollut tehdä erillisiä kenttiä HTML-viestille vaan ne kirjoitetaan samaan kenttään kuin pelkkä teksti. Lisäksi

muokkasin moduulin ulkoasua. Pääsin tallentamaan uudet asetukset tietokantaan, mutta jostain syystä en saa tuota asetusta toimimaan. Vaikka tietokannassa on HTML-sähköpostiasetuksessa arvona epätosi, ohjelmisto olettaa sen olevan tosi ja lähettää HTML-muotoista viestiä. Samoin ulkoasussa tietokannan arvoa en ole saanut oikein ulos, vaan sekin näyttää aina siltä kuin käytössä olisi HTML-muotoilut.

### **Keskiviikko 13.10**

Tänäänkin jatkoin salasananpalauttamismoduulin sähköpostien HTML-muokkauksen mahdollistamista. Löysin eiliseen tosi-/epätosiongelmaan ratkaisun: php-koodiin piti lisätä tieto siitä, että kyseinen arvo on totuusarvo. Tällä pienellä muokkauksella koko ominaisuus lähti toimimaan. Tein vielä pieniä viilauksia ja ajelin testejä, sekä tein muutamille eri kieliversioille (suomi, ruotsi, englanti) käännökset.

Loppupäivän käytin PostgreSQL-kurssiin Udemystä. Minun täytyi myös asentaa Postgres.app uudelleen koneelleni, koska edellisellä kerralla jokin oli ilmeisesti mennyt vikaan enkä saanut sitä tänään toimimaan. Tällä kertaa tein asetusten määrittämisen tarkemmin, joten nyt sen pitäisi toimia.

### **Torstai 14.10**

Tämän päivän suurinta antia oli epätäydellisyyden sietäminen. Uudelleenohjaustehtävän katselmoinnissa todettiin, että se on varmistuksesta huolimatta väärin tehty ja tein sen uudelleen. Onneksi homma oli nopea, nyt kun tiesin mitä kaikkea sieltä pitää muuttaa. Yhteen drag and drop -tehtävään tuli myös pieni korjaus ja yhdestä tehtävästä oli unohtunut upgrade script, joten ne tein myös. Raskas päivä ihmiselle, jolla on vaikeuksia sietää omia virheitä!

Korjausten lisäksi aloittelin uutta tehtävää, jossa korjataan toimimaton property uutisiin ja tapahtumiin. En ole tähän propertyyn tutustunut aiemmin ollenkaan, joten aloitin tehtävän tutkimalla, miten se toimii.

### **Perjantai 15.10**

Jatkoin tänään eilen aloitettua tehtävää toimimattoman propertyn parissa. Kyseinen property on nimi-arvo -pari. Sitä ei jostain syystä saada näkymään sivustolla uutisten ja tapahtumien kohdalla,

mutta esimerkiksi tuotteissa se toimii. Tämä on taas sellainen ominaisuus, joka hyödyntää monia luokkia ja virheen metsästäminen on kuin etsisi neulaa heinäsuovasta. Löysin paikan, josta sen saa rikki tuotteiden kohdalla, mutta en saa vastaavaa toimivaa toiminnallisuutta aikaiseksi uutisten ja tapahtumien puolella. Tiimin odotukset tämän toiminnallisuuden korjaamiseksi kuuluivat olevan hyvin pienet, joten ei ihme, että tuntuu hankalalta. Tutkin kuitenkin lisää ensi viikolla!

## **Viikkoanalyysi – PostgreSQL**

Tälläkin viikolla tein paljon tietokantoihin liittyviä asioita. Opinnoissani olin tutustunut MySQL:n käyttöön, eikä PostgreSQL:n peruskäyttö poikkea siitä kovinkaan paljoa. Molemmat ovat relaatiotietokantoja (relational database management system RBDS). Relaatiotietokannassa on tauluja, joiden tiedot linkittyvät toisiinsa yhteisten tietoarvojen eli avainten kautta. Jokaisen taulu koostuu riveistä ja sarakkeista, joihin tieto tallennetaan. Ensimmäisen kerran relaatiotietokantaa kuvaili E. F. Codd vuonna 1970 Communications of the ACM -lehdessä artikkelissaan "A Relational Model of Data for Large Shared Data Banks". Codd sovelsi matemaattisia teorioita, erityisesti joukko-oppia tietojen hallintaan sekä laati luettelon kriteereistä, jotka tietokannan on käytettävä, jotta niitä voidaan pitää relaatiotietokantoina. Näitä periaatteita käytetään edelleen tietokannan relaatio-ominaisuuksien validointiin: tietokanta, joka ei täytä näitä sääntöjä, ei ole täysin suhteellinen. (Hsu, Kline & Obe 2022, luku 1.)

PostgreSQL on laajasti tunnettu avoimen lähdekoodin relaatiotietokanta. Se on täysin kehitetty avoimen lähdekoodin maailmassa, mikä tarkoittaa sitä, että projekti on täysin ei-kaupallinen, eikä siitä ole vastuussa mikään yksittäinen yksikkö. PostgreSQL kuuluu kaikille ja kuka tahansa voi osallistua sen kehittämiseen. Koska PostgreSQL:n takana on suuri yhteisö minkään yksittäisen yrityksen sijasta, on mahdollista uskoa sen jatkavan kehittymistään pitkään ilman pelkoa yritysten konkurssista tai muista vaikeuksista. PostgreSQL:n tärkeimpiä ominaisuuksia ovat vakaus, skaalautuvuus ja turvallisuus. Se on kykenevä suoriutumaan valtavista tietomääristä, lähes määrättömän suurista. Jotta pystyy toimimaan suurten tietokantojen kanssa, on käyttäjän kuitenkin ymmärrettävä PostgreSQL:ää ja sen ominaisuuksia. PostgreSQL on yhteensopiva lähes kaikkien käyttöjärjestelmien kanssa. Laajan muokkausmekanisminsa ansiosta se voidaan mukauttaa erittäin hyvin erilaisille alustoille. (Ferrari & Pirozzi 2020, luku "Introduction to PostgreSQL".)

Oma kehittymiseni tietokantaohjelmoinnissa on tasaisen varmaa. Olen kiinnostunut tietokannoista ja niiden logiikka tuntuu minulle järkevältä. Vaikka PostgreSQL oli minulle ennen harjoittelun alkamista vieras, on MySQL-tuntemuksestani ollut paljon hyötyä, sillä perustason toiminta on hyvin samanlaista. Tietokantahaut ja tietojen muokkaaminen toimii pitkälti samalla tavalla. Tunnistan kuitenkin vielä paljon kehittymistä. Esimerkiksi sarakkeen lisäämiseen tai poistamiseen käytetyn komennon joudun edelleen tarkistamaan, koska en muista sitä ulkoa. Työnteko nopeutuisi, jos tällaisia perusasioita muistaisi. Toisaalta annan itselleni tässä asiassa armoa ja ennemmin tarkistan omista muistiinpanoistani kuin stressaan muistamattomuuttani. Olen kuitenkin vielä aika tuore ohjelmoija.

### **3.6 Viikko 6: 18.10.-22.10.2021 – Ongelmanratkaisu / debuggaus**

#### **Maanantai 18.10.**

Viime viikolla aloitetun propertykorjauksen jatkamista. Tutkin eri luokissa tapahtuvaa tiedon käsittelyä ja tarkastelin niitä Chromen konsolissa. Selvisi, että nimi-arvo –pari tulostuu eri tavoin tuotteissa ja tapahtumissa, mutta en vielä keksinyt syytä sille, miksi näin tapahtuu. Tapahtumien ja uutisten propertyt käsitellään eri tavalla kuin esimerkiksi tuotteiden ja täytyy lähteä kaivelemaan niitä eroavaisuuksia jatkossa tarkemmin.

#### **Tiistai 19.10.**

Tämä päivä oli todellista ongelmanratkaisua. Löysin tähän etsimääni bugiin liittyen toisenkin propertybugin. Taulukkoa ei myöskään saada tulostettua tapahtumille tai uutisille, aivan kuten ei nimi-arvo –pariakaan. Kävin tänäänkin läpi eri funktioiden tulostelua ja pääsin luullakseni jo melko lähelle ongelman juurisyytä. Uskoakseni se on kuin onkin siellä, mistä sen alkuunkin epäilin olevan, mutta en saanut toimivaksi, eli news-luokan propertyfunktiossa.

#### **Keskiviikko 20.10.**

Ratkaisu lienee jo aika lähellä. Tein tänään tietokantahakuja miettien, miten saisin kaiken tarvittavan tiedon ulos tietokannasta rikkomatta mitään. Sain lopulta toimivan tietokantahaun aikaiseksi. Koodia pitää vielä muokata, että tiedot saa tulostettua oikein. Tällä hetkellä tiedot tulostuvat, mutta

sinne tulostuu myös ylimääräistä tietoa (kuvio 7). Testitapauksessani propertyn nimi on "Nimi" ja arvo on "Arvo" ja niiden molempien pitäisi tulostua kerran. Tulostuksen short\_text on propertyn tyyppi.

# 14.10.2021

## Tapahtuma

### Teksti

Nimi: Arvo

Arvo: Arvo

Nimi: Nimi

Arvo: Nimi

Nimi: short\_text

Arvo: short\_text

*Kuvio 7. Kuvakaappaus kokeilusivustoltani.*

### **Torstai 21.10.**

Jatkoin saman ongelman parissa. Ei suuria läpimurtoja. Kokeilin erilaisia tapoja saada tieto oikeaan muotoon samaan tapaan kuin tuotteissa, mutta se johtaa jostain syystä aina virheisiin. Alkaa tuntua siltä, että koko koodi pitäisi kirjoittaa uusiksi, mutta en luota siihen, että saisin siihen upotettua kaiken tarpeellisen. Viikon taistelunkin jälkeen koodi on minulle vielä osittain lukukelvotonta kaik-  
kine yksikirjaimisine muuttujineen.

### **Perjantai 22.10.**

Perjantai, eli toimisto- ja viikkopalaveripäivä. Tänään oli toimistolla meidän tiimimme osalta hiljaista, vain kolme paikalla. Viikkopalaveri pidettiin tavan mukaisesti osittain etänä.

Jatkoin saman ongelman selvittelyä. Luulin jo olevani lähellä ratkaisua, mutta en olekaan. Selvitte-  
lin edelleen funktioiden kulkua luokasta toiseen. Tähän toimintoon vaikuttaa ainakin neljä eri luok-  
kaa. Tiimin kokeneempi jäsen alkoi selvittää ongelmaa myös, koska minä en siihen yksin taida  
pystyä. Katsomme ensi viikolla, selviääkö tämä ollenkaan.

## **Viikkoanalyysi – Ongelmanratkaisu / debuggaus**

Tämän viikon teemaksi valitsin ongelmanratkaisun ja debuggauksen eli virheenkorjauksen, koska  
arvasin tämän viikon tehtävässä sitä tarvittavan.

Virheenkorjaus on prosessi, jossa havaitaan ja poistetaan olemassa olevia ja mahdollisia virheitä  
(bugeja) ohjelmistokoodissa, jotka voivat aiheuttaa sen ohjelmiston odottamattoman käyttäytymi-  
sen tai kaatumisen. Ohjelmiston virheellisen toiminnan estämiseksi tehdään virheenkorjausta viko-  
jen löytämiseen ja korjaamiseen. Kun eri alijärjestelmät tai moduulit on liitetty tiukasti toisiinsa, vir-  
heenkorjaus vaikeutuu, koska kaikki muutokset yhdessä moduulissa voivat aiheuttaa lisää virheitä  
toisessa. Joskus ohjelman vianetsintä vie enemmän aikaa kuin sen koodaaminen. (The Economic  
Times 2021.)

Virheenkorjausta varten käyttäjän on aloitettava ongelmasta, eristettävä ongelman lähdekoodi ja  
korjattava se. Kun vika on korjattu, ohjelmisto on käyttövalmis. Virheenkorjaustyökaluja, joita kut-  
sutaan myös debuggereiksi, käytetään tunnistamaan koodausvirheet eri kehitysvaiheissa. Niitä  
käytetään toistamaan olosuhteet, joissa virhe on tapahtunut, sitten tutkittava ohjelman tila tuolloin  
ja löydettävä syy. Ohjelmoijat voivat jäljittää ohjelman suoritusta vaiheittain arvioimalla muuttujien  
arvon ja pysäyttää suorituksen aina, kun se on tarpeen muuttujien arvon saamiseksi tai ohjelman  
muuttujien nollaamiseksi. Jotkin ohjelmointikielipaketit tarjoavat virheenkorjaimen, joka tarkistaa  
koodin virheiden varalta, kun sitä kirjoitetaan ajon aikana. (The Economic Times 2021.)

Virheenkorjausprosessi menee näin:

1. Toista ongelma.
2. Kuvaile vikaa. Yritä saada mahdollisimman paljon palautetta käyttäjältä saadaksesi tarkan  
syy.
3. Ota ohjelmasta kuvakaappaus, kun vika tulee näkyviin. Yritä saada kaikki ohjelman muut-  
tujien arvot ja tilat sillä hetkellä.

4. Analysoi tilannekuva tilan ja toiminnan perusteella. Yritä sen perusteella löytää vian syy.
5. Korjaa olemassa oleva vika, mutta tarkista myös, ettei uusia virheitä ilmene.

(The Economic Times 2021.)

Koen, että oma ongelmanratkaisukyky on aina ollut melko hyvä. Olen luova ihminen ja osaan usein ajatella ns. laatikon ulkopuolelta. Tämä taitoni on yksi syy, miksi päädyin ohjelmointialalle aikuisiällä. Ongelmanratkaisu- ja debuggauskirjallisuutta luettuani olen kuitenkin todennut, että on paljon asioita vielä opittavana nimenomaan ohjelmointiin liittyen.

Toimivana ratkaisuna ongelmanratkaisussa minulla on ollut ns. kumiankkametodi. Siinä ongelma selitetään jollekin toiselle ihmiselle, jonka odotetaan nyökkäilevän ja kuuntelevan, tai vaikkapa kumiankalle. Kun selittää ongelmaa toiselle, joutuu kertomaan asioita, joita itse pitää itsestään selvinä, kun pyörittää koodia itsekseen. Kun sanoo asiat ääneen, saattaa yllättäen saada asioihin uuden näkökulman ja ongelmat ratkeavat. (Thomas & Hunt 2019. Luku 20.) Minulla ei ole kotitöimistöllä työkavereita eikä kumiankkaa, mutta sen sijaan minulla on hyvin osallistuva koiranpentu, jolle asioitani olen selittänyt ja se on kuunnellut tarkkaavaisena. Muutama ongelma ratkesikin tämän avulla!

### **3.7 Viikko 7: 25.10.-29.10.2021 – Git ja versionhallinta**

#### **Maanantai 25.10.**

Tänään oli vähän lyhyempi päivä työhöntulotarkastukseen kuuluvan lääkärintarkastuksen takia. Jatkoin nimi-arvoparin tutkimista. Alan epäillä, etten saa tätä toimimaan mitenkään. En vain ymmärrä, miten tämä on tehty! Suurimman osan ajasta koitan selvittää, mitä mikäkin osa koodista tekee, mutta en oikein saa niitä auki. Tämä on tehty jotenkin aivan toisella tapaa vaikeasti kuin muut tähän asti tutkimani koodin alueet. En tänään vielä saanut apua toiselta työntekijältä, mutta huomenna toivottavasti asia selkeytyy.

## **Tiistai 26.10.**

Aamupäivästä katselin vielä nimi-arvopariongelmaa, enkä päässyt eteenpäin. Työkaverilta sain apuja, mutta hän ei vielä päässyt ongelmassa sen pidemmälle kuin minäkään. Iltapäivästä vaihdoin gitin branchin takaisin aiemmin tehtyyn analytiikkaongelmaan, jossa CMS ei saa tietoa analytiikkakytköksen katkeamisesta ja näyttää päällisin puolin olevan kunnossa, mutta ei saa tietoa Googelta. Se vaati korjauksia ja parannuksia, jotka sainkin melko lailla valmiiksi tänään.

## **Keskiviikko 27.10.**

Aloitin päivän viimeistelemällä analytiikkakytkösongelman ratkaisun. Siihen ei kauaa enää mennyt-kään. Sen jälkeen jatkoin nimi-arvoparin tutkimista, kunnes päädyimme siihen, että siihen ei kannata käyttää enempää aikaa, koska ei tunnu selviävän. Tämän jälkeen aloitin uuden tehtävän, jossa täytyy tehdä backendiin tuki monen tiedoston yhtäaikaiselle lähettämiseksi lomakkeelta. Aloitin tutkimalla, miten se on yleensä tehty ja miten tiedostojen käsittely on tällä hetkellä backendissä toteutettu. Lisäsin admin-näkymään valinnan, sallitaanko useamman tiedoston lähetykset sekä tietokantaan sarakkeen tätä tietoa varten.

## **Torstai 28.10.**

Tänään onnistuin saamaan eilen aloitetun työn niin jumiin, että tiedoston lähettäminen ei onnistunut enää ollenkaan. Koska tein sen virheen, että en ollut kokeillut tätä ominaisuutta ennen muutosten tekemistä koodiin, en ollut varma, miten se toimi aiemmin. Vaihdoin siis gitistä branchin takaisin masteriin ja kokeilin, eikä mitään ongelmaa ollut. Yritin käyttää gitin diff -toimintoa, jonka pitäisi näyttää branchien erot, mutta koska en saanut sitä järkevässä ajassa toimimaan, päätin hylätä muutokset ja aloittaa alusta. Pääsinkin hyvin nopeasti samaan tilanteeseen kuin eilen ja hoksasin jopa sen syyn, miksi tiedoston lähetys ei onnistunut. Kokeilin muokata tiedostojen tallentamista, mutta koska lomakemoduulin koodi on niin monimutkaista, on siellä varmasti useampi kohta, jota pitää muokata päästäkseen toivottuun lopputulokseen.

## **Perjantai 29.10.**

Viikkopalaveripäivä. Keskustelimme mm. nimi-arvopropertyn bugista ja siitä, mikä siinä oli vialla. Totesimme, että vaatii paljon uudelleenkirjoittamista, joten jätetään hautumaan. Palaverin lisäksi



jatkoin usean tiedoston lähettämisen mahdollistamista. Työ oli käytännössä selvillä, missä muodossa tiedot arrayssa tulevat ja niiden uudelleenmuokkaamista. Arrayt, joiden sisällä on useita sisäkkäisiä arrayita, on minusta tosi hankalia käsitellä ja joudunkin aina kokeilemaan moneen kertaan, että saan niistä halutun tiedon ulos. Sain kuitenkin tallentamisen toimimaan, tieto tiedostosta menee tietokantaan ja itse tiedosto on tallessa oikeassa kansiossa. Ensi viikolle jää vielä pienten ulkoasuasioiden korjaamista ja testaamista sekä mahdollisten ongelmien korjaamiset.

## **Viikkoanalyysi – Git ja versionhallinta**

Päätin ottaa tämän viikon analyysiaiheeksi Gitin käytön, koska se on olennainen osa päivittäistä tai vähintäänkin viikottaista tekemistäni.

Versionhallinta tarkoittaa useiden projektin versioiden hallintaa. Version hallitsemiseksi jokaista projektin tiedostoihin tehtyä muutosta (lisäys, muokkaus tai poisto) on seurattava. Versionhallinta tallentaa jokaisen tiedostoon tehdyn muutoksen ja antaa tavan kumota tai peruuttaa nuo muutokset. Tehokkaaseen versionhallintaan tarvitaan hyvä versionhallintajärjestelmä (Version Control System, VCS). Sen avulla voi siirtyä muutosten välillä ja palata nopeasti ja helposti edelliseen versioon, mikäli jokin menee pieleen. Yksi versionhallinnan käytön suurimmista eduista on ryhmätyö. Kun useampi kuin yksi henkilö osallistuu projektiin, muutosten seurannasta tulee sekavaa ja todennäköisyys tehdä muutoksia samaan tiedostoon, kasvaa. Versionhallinnan avulla useat ihmiset voivat tehdä töitä samassa projektissa omassa versionhallintahaarassaan ja yhdistää muutokset pääprojektiin vasta, kun he ja muut tiimin jäsenet ovat tyytyväisiä työhön. (Tsitoara 2019. Luku 1.)

On olemassa lukuisia eri versionhallintajärjestelmiä, mutta kaikista suosituin niistä on Git. Gitin takana on sama ihminen kuin Linuxilla, Linus Torvalds. Hän toteutti Gitin vastatakseen omiin tarpeisiinsa ja julkaisi sen maailmalle heinäkuussa 2005. Gitin tavoitteena oli olla nopea ja tehokas ja tässä tavoitteessaan se onnistui. Git käyttää hajautettua järjestelmää, jossa jokaisella on oma arkisto. Gitille ei välttämättä tarvita keskuspalvelintä, mutta ryhmässä työskennellessä paikka koodin jakamiseen tarvitaan. (Liberty & Galloway 2021. Luku 1.)

Haarojen (branchien) käyttäminen on tärkeää Gitin kanssa työskentelyssä. Gitissä on tallessa päähaara, master, jossa on julkaisussa oleva koodi. Aina kun koodi lisätään päähaaraan, se katselmoidaan, jotta päähaara pysyy mahdollisimman puhtaana. Kun koodin kanssa työskennellään, tehdään uusi haara, nk. ominaisuushaara (feature branch). Tämä luo kopion koodista, joka on tällä

hetkellä päähaarassa. Nyt ominaisuushaarassa voi työskennellä vaikuttamatta päähaaraan ollenkaan. Kun työ on valmis, kaikki toimii ja on katselmoitu, voi ominaisuus haaran yhdistää päähaaraan. (Liberty & Galloway 2021. Luku 3, Branches.)

Gitiä käytetään yleisimmin tietokoneen komentokehoteen kautta. Kaikki komennot alkavat sanalla "git". Jokaisen ohjelmoijan on hyvä osata muutama peruskomento, kuten git init, git clone, git checkout, git add, git commit, git push ja git pull. Hyödyllisiä ovat myös git diff, git stash, git status ja git log. Git init alustaa Gitin käyttöä varten. Git clone kopioi koodin repositoriosta omalle koneelle. Git checkout vaihtaa koodin haarasta toiseen. Git add lisää koodin tallennettavaksi ja git commit tallentaa muutokset eräänlaiseen tarkistuspisteeseen. Git push vie koodin repositorioon ja git pull tuo sen sieltä omalle koneelle. Git diff näyttää muutokset oman ja jonkin toisen haaran välillä. Git stash tallentaa työn väliaikaisesti, jos esimerkiksi haluaa siirtyä haarasta toiseen. Git status kertoo missä haarassa työskennellään, onko haara ajan tasalla ja onko mitään siirretty tallennettavaksi tai vietäväksi repositorioon. Git log kertoo koko commit-historian. (Koushik 2020.)

Olen itse opetellut käyttämään Gitiä jo opiskeluaikoina ja sen peruskäyttö sujuuikin suhteellisen varmasti. Jännitän kyllä joka kerta koodia pushatessani, että meneehän se oikeaan haaraan enkä sekoita mitään. Tämä liittyyneen kuitenkin enemmän omaan epävarmuuteeni koodarina kuin varsinaiseen Gitin käytön haasteellisuuteen. Olen opetellut joitain uusia komentoja, mutta pääasiassa käytän noita aiemmin listattuja peruskomentoja. Git diff ja git stash on tullut opeteltua tällä viikolla, mutta niiden käyttöön tarvitsen vielä muistilappuja.

### **3.8 Viikko 8: 1.11.-5.11.2021 – Huijarisyndrooma ja ammatillinen kasvu**

#### **Maanantai 1.11.**

Jatkoin perjantaina lähes valmistunutta usean tiedoston lähettämisen mahdollistamista. Tein erilaisia testejä, lähettelin liian suuria tiedostoja ym. Kokeilin siis kaikin mahdollisin keinoin saada moduulin rikottua. Lisäksi tein muutaman viilauksen moduulin asetusten ulkoasuun. Tämä oli mukava tehtävä, ei liian vaikea, mutta ei liian helppokään tehtävä. Muutama viimeksi tehty tehtävä on palautunut takaisin korjattavaksi ja luulen, että tässä ei niin käy. Kasvattaa itseluottamusta, kun onnistuu.

## **Tiistai 2.11.**

Tänään korjailin taas kerran analytiikkakytkösongelman korjailua. Toivon, että sain viimeiset korjaukset tehtyä ja homma toimisi toivotusti. Viimeiset pari viikkoa olen taas kamppailut itseluottamusongelmien kanssa, joten onnistuminen tulisi tarpeeseen. Jatkoin myös Udemystä PostgreSQL-kurssia, jossa käytiin läpi erilaisia validointitapoja sekä tietokannan struktuuria. Lisäksi jatkettiin erilaisten tietokantojen luomista ja hyviä tapoja tehdä tietokantoja. Päivän päätteeksi aloitin vielä uuden tehtävän tutkimisen. Tässä täytyy tehdä menumoduulille suodatus tunnisteiden eli asiasanojen perusteella.

## **Keskiviikko 3.11.**

Jatkoin eilistä tehtävää menumoduulin parissa. Tutkin tuote- ja menumoduulien eroja suodatuksen kannalta ja totesinkin ne melko eri tavoilla tehdyiksi. Menumoduulissa ei tällä hetkellä ole oikeastaan minkäänlaista suodatusta. Tietokantaan täytyi myös tehdä uusi taulu, johon tallennetaan suodatuksessa käytettävät tunnisteet. Itse taulun luomisen olisin osannut tehdä ilman netistä löytyviä neuvoja, mutta en muistanut miten tauluun asetetaan viittaukset toisiin tauluihin, joita tässä tapauksessa oli kolme. Sain taulun luotua ja viittauksetkin menemään oikein.

## **Torstai 4.11.**

Aamulla sain viestiä työkaverilta, että on hyväksynyt analytiikkakytkösongelman korjaukset. Mikä helpotus! Ehkä pääsen tälläkin kertaa tästä itseluottamuskuopasta ylös. Jatkoin lisäksi menumoduulin tutkimista. Tein menumoduuliin tunnisteiden tallennusta ja hakua, mutta en saanut niitä toimimaan. Tutkiessani ongelmaa, huomasin että tietokannassa taulun omistajana on veera.annala eikä www-data kuten muilla tauluilla ja epäilen tämän olevan ongelman takana. En saanut omistajaa vaihdettua enkä vielä keksinyt muutakaan ratkaisua, mutta jatkan taas huomenna.

## **Perjantai 5.11.**

Perinteinen viikkopalaveri- ja toimistopäivä. Viikkopalaverissa käytiin taas pikaisesti läpi mitä kukin on tehnyt. Minulla valmistui tällä viikolla lomakemoduulin useamman tiedoston lähetys ja siitä puhuttiin muutama sananen. Palaverin jälkeen jatkoin menumoduulin suodatusta tunnisteilla. Sain työkaverilta apua tietokannan taulun omistajan vaihdokseen ja nyt tunnisteiden tallennus ja haku

onnistuu moitteetta. Tuotteissa on toteutettu suodatus kaikin osin eri tavalla kuin menussa, joten joudun miettimään, miten tämä tehdään loppuun. Menussa sivut tulevat cachem kautta arrayna, jossa ei ole listattuna tunnisteita. Tämä tieto täytyy jotenkin saada mukaan, että suodatus toimii.

## **Viikkoanalyysi – Huijarisyndrooma ja ammatillinen kasvu**

Valitsin viikkoanalyysin aiheeksi huijarisyndrooman ja sen merkityksen ammatilliselle kasvulle, koska tunnistan itsessäni piirteitä tällaisesta käyttäytymisestä ja haluan oppia, miten siitä pääsee eroon pystyäkseen kasvamaan ammattilaisena taitojeni tarjoamalle tasolle. Lisäksi tällä viikolla olen korjailut aiemmassa tehtävässä tulleita virheitä ja se vaikuttaa minulla aina koko päivän suoriutumiseen negatiivisesti.

Huijarisyndrooma on tunne siitä, ettei usko osaavansa oikeasti huolimatta onnistumisista. Siitä kärsivä pelkää muiden huomaavan, ettei olekaan hyvä, että vain huijaa osaavansa ja on tullut valituksi tehtävänsä sattumalta tai epähuomiossa. Huijarisyndrooma ei ole oikeasti huijaamista, valehtelua tai petosta, vaan se on voimakas sisäinen kokemus omasta osaamattomuudesta, vaikka ympärillä toiset ovat täysin päinvastaista mieltä. "Huijauksen" paljastuminen saattaa aiheuttaa voimakasta ahdistusta ja masennustakin. Huijarisyndrooma on samaa sukua kuin perfektionismi, itsekriittisyys, vaativuus itseä kohtaan sekä suorittamispakko. Kaikkien taustalla on sama epävarmuus omasta minuudestaan ja kelpaamisesta yhteiskuntaan ja työelämään sellaisena kuin on, sekä niihin kaikkiin liittyy monesti työuupumusta, ahdistus- ja masennusoireita sekä paniikkikohtauksia. (Ekman 2017. Luku 2.)

Ammatillinen identiteetti rakentuu työn tekemisen ja siitä saatujen kokemusten yhteisestä vuorovaikutuksesta. Ihminen mittailee itseään ja ominaisuuksiaan työn ominaisuuksiin jo opiskeluaikoina ja sen mukaan määrittää omaa ammatti-identiteettiään. Kun huijarisyndroomasta kärsivä astuu työelämään, hänen minäkuvassaan on sinnikäs itsensä epäilyn elementti, joka vaikeuttaa identiteetin rakentumista. Ammatti-identiteetti perustuu näin ajatukseen omasta osaamattomuudesta ja itsensä epäilemisestä. Tällä tavoin muodostunut identiteetti pitää yllä ahdistusta ja altistaa masennukselle. Syndroomasta kärsivälle ihmiselle omien taitojen määrittely on vaikeaa, eikä hänellä ole luottamusta omiin kykyihinsä. Hänen ammatti-identiteettinsä on hauras ja epävakaa, ja työelämän jatkuvat muutosvaatimukset voivat haitata sen vakiinnuttamista. (Ekman 2017. Luku 6, Huijarin ammatti-identiteetti.)

Yksilön urakehitys ja uran suunnittelu vaatii itsearvostusta, tavoitteita ja eteenpäin pyrkimistä. Työelämässä jokaisen odotetaan muuntuvan ja oppivan uusia taitoja ja huijarisyndroomasta kärsivällä on jo etukäteen epäily siitä, ettei hän oikeasti pysty siihen. Hänelle on tyypillistä yhtäaikainen epäonnistumisen ja onnistumisen pelko, jotka ovat esteenä urakehitykselle. Oman uran ja työelämän suunnittelu on usein hankalaa ja puuttuu kokonaan. Yleisimmin syndroomasta kärsivät jäävät paikoilleen, jotta eivät epäonnistu missään uudessa. (Ekman 2017. Luku 6, Asenne omaan osaamiseen ja urakehitykseen.)

Itse tunnistan tämän kaiken omassa elämässäni, sekä työ- että henkilökohtaisessa elämässäni. Vaadin itseltäni täydellisiä suorituksia ja pelkään päivittäin, milloin toiset huomaavat minun huijanneen osaamiseni. Tämä kuormittaa jaksamistani ja usein päivän päätteeksi olenkin aika väsynyt. Lisäksi toisten huomaamat virheet koodissani on aina pieni maailmanloppu. Entisissä töissäni sain usein negatiivista palautetta varsin aggressiivisella otteella ja tämäkin osaltaan vaikuttaa virheiden sietokykyyni. Olen kuitenkin nyt tietoisesti pyrkinyt ajattelemaan, että olen vasta urani alussa eikä minun tarvitse olla vielä täysi ammattilainen. Työpaikalla sitä onneksi ei edes odoteta, vaan saan tehdä tehtäväni omaan hitaaseen tahtiini asioita opetellen. Tämä lempeä suhtautuminen aloittelemaan ohjelmoijaan on ollut yksi parhaita asioita tässä työpaikassa ja auttaa myös omaan suhtautumiseen osaamisestani. Huomaan, että minulle annetaan vähä vähältä vaativampia työtehtäviä, joista suoriudun itsenäisesti, joskin hitaasti. Pikkuhiljaa omakin luottamus osaamiseen toivottavasti kasvaa ja ammattilaisen identiteetti pääsee vahvistumaan.

### **3.9 Viikko 9: 8.11.-12.11.2021 – REST API**

#### **Maanantai 8.11.**

Menumoduulin suodatuksen jatkamista. Tutkiskelin, miten suodatuksen saisi helpoiten toimimaan ja kokeilin muutamaa tapaa. Luulen, että kokeilin tätä niin, että haen tietokannasta niiden sivujen id-numerot, joilla on tuo haluttu suodatus ja vertaan sitä sitten jo olemassa olevaan listaan sivuista. Tätä jo koitinkin rakentaa tänään, mutta en saanut tietokantakutsua toimimaan koodissa, vaikka suoraan tietokannasta hakiessa sainkin halutun tiedon ulos.

## **Tiistai 9.11.**

Menumoduulin suodatus lähti tänään toimimaan. Sain tietokantahaun toimimaan toivotusti ja suodatuksenkin pelaamaan. Huomasin eilisen ongelman olevan siinä, että id-numerot olivat array-muodossa, joten käytin tänään php:n implode-funktiota saadakseni ne erotelluiksi pilkuilla. Sain myös testattua suodatuksen toiminnan sekä omilla käyttötesteillä että CMS:n testiajolla. Lisäksi tein käännökset eri kieliversioille.

## **Keskiviikko 10.11.**

Minulla on alkamassa uusi työtehtävä, jossa kopioidaan tietoja asennukselta toiselle REST API:n avulla. Tässä täytyy tehdä uusi moduuli, jonka asentamalla ja konfiguroimalla tiedot kopioidaan ja ylläpidetään molemmissa asennuksissa. Olen aiemmin tutustunut jo Lianan REST API:n, mutta kävin tänään uudelleen läpi sen toimintaa ja käyttöä sekä tutustuin muihin vastaavanlaisiin ratkaisuihin. Huomenna pääsen aloittamaan tehtävän kunnolla, kun saan siihen lopullisen perehdyttämisen. Lisäksi opiskelin Udemysta PostgreSQL-asioita.

## **Torstai 11.11.**

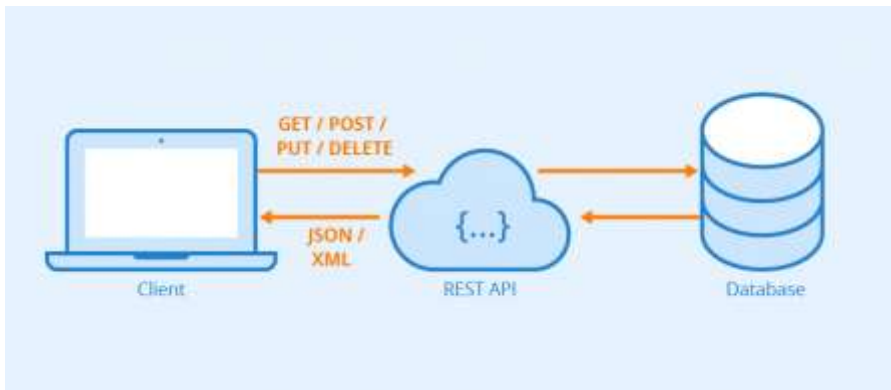
Sain lyhyen perehdytyksen uuteen tehtävään ja aloittelin tekemään sitä. Siinä tuntuu olevan vähän haastetta, kun en ole tällaisia koskaan aiemmin tehnyt. Vaikuttaa mielenkiintoiselta ja toivon pääseväni tähän huomenna kunnolla sisään. Tämä päivä meni vielä tutustellessa REST API:n ja sen käyttöön sekä yleisesti tuon tulevan moduulin ominaisuuksiin. Muistelin myös vähän JSONin kirjoittamista.

## **Perjantai 12.11.**

Tänään oli taas toimistopäivä. Käytiin vielä tarkemmin läpi uuden moduulin speksejä ja sen tekemistä. Luonnostelin itselleni, missä järjestyksessä alan tehdä hommia. Tarkoituksena on siis hakea tiedot nykyiseltä asennukselta, muuttaa ne REST API:n sopiviksi ja tallentaa toiselle asennukselle, joka voi olla myös toisella palvelimella. Työ kuulosti varsin selkeältä, mutta olikin ajattelemaani monimutkaisempi. Minulla alkaa kuitenkin olla hyvä käsitys siitä, mistä aloittaa ja miten työtä lähtee jatkamaan.

## Viikkoanalyysi – REST API

REST API on lyhenne sanoista Representational State Transfer ja Application Programming Interface. Ne ovat keskenään kaksi eri asiaa, joista yhdessä muodostuu REST API. (Bin Uzayr 2016. Luku 1.)



Kuvio 8. REST API kuvana. (Seobility.)

API tarkoittaa sovellusohjelmointirajapintaa. Sen avulla voi muodostaa yhteyden kahden erityyppisen ohjelmiston välille. Esimerkkinä voi käyttää tietokoneen USB-porttia, johon voi liittää käytännössä minkä tahansa tyyppisen laitteen, joka käyttää USB:tä. Aivan kuten USB-portti helpottaa tiedonsiirtoa kahden fyysisen laitteen välillä, API helpottaa tiedonsiirtoa kahden erityyppisen ohjelmiston välillä. (Bin Uzayr 2016. Luku 1.)

REST puolestaan on HTTP-protokollaan perustuva arkkitehtuurimalli ohjelmistorajapintojen toteuttamiseen. REST on kevyt ja pystyy käsittelemään suuria määriä toimintaa helposti. Sovelluksia, jotka käyttävät RESTiä, sanotaan RESTful-sovelluksiksi. RESTin taustalla on ajatus, että monimutkaisten verkkopalveluiden sijaan yhteyksien luomiseen käytetään yksinkertaista HTTP-protokollaa. Kaikki RESTful-sovellukset käyttävät http-pyyntöjä käsitellessään CRUD-toimintoja, eli luodessaan (Create), lukiessaan (Read), päivittäessään (Update) ja poistessaan (Delete) tietoja. REST on myön täysin alustariippumaton. Sitä voidaan käyttää esimerkiksi tilanteissa, joissa palvelin toimii Linuxilla ja asiakkaan kone Windowsilla. Koska REST on standardipohjainen ja kielestä riippumaton, REST-pyyntö sisältää kaiken tiedon, jota saatetaan tarvita sen suorittamiseen tai loppuun saattamiseen. Melkein kaikki suurimmat verkkopalvelut kuten Google, Facebook ja Twitter käyttävät RESTiä. (Bin Uzayr 2016. Luku 1.)

Kun käyttää REST APIA, törmää väistämättä myös termiin JSON. JSON on lyhenne sanoista JavaScript Object Notation. Se on JavaScriptiin perustuva tiedonsiirtomuoto. JSON on ihmisten helppo ymmärtää ja kirjoittaa ja koneiden helppo jäsentää ja luoda. JSON on tekstimuoto, joka on täysin kieliriippumaton, mutta siinä on käytäntöjä, jotka ovat tuttuja C-kieliperheen ohjelmoijille, mukaan lukien C, C++, C#, Java, JavaScript, Perl, PHP, Python ja monet muut. Nämä ominaisuudet tekevät JSONista ihanteellisen tiedonsiirtokielen. (json.org.)

Itselleni tutustuminen REST API:n käyttöön on ollut mieluista. Olen käyttänyt vastaavia jo opiskeluissa, mutta en varsinaisesti ymmärtänyt mitä ne ovat ja miten ne toimivat. API:n käyttö on ollut helppo oppia, mutta itselleni oli vaikeampaa hahmottaa sen toiminta taustalla. Tutkittuani ja opiskeltuani REST API:n toimintaa, on minulle avautunut aivan toisella tavalla niiden käyttömahdollisuudet.

### **3.10 Viikko 10: 15.11.-19.11.2021 – Kommunikaatio etätyössä**

#### **Maanantai 15.11.**

Jatkoin viime viikolla aloittamaani moduulin tekemistä. Päätin aloittaa siitä, että saan yksittäisen entiteetin tallennuksen onnistumaan halutusti, koska sen luokan metodia tarvitaan myös koko kategorian tallennukseen. Kokeilin, miten alkuperäisen tallennettavan entiteetin tiedot saadaan helpoimmin tallennettavaksi toiselle entiteetille. Vielä en ottanut tietoja siihen JSON-tiedostosta, se jää huomiseksi. Mietin jo kuitenkin valmiiksi, millä tavalla lähden sitä toiminnallisuutta toteuttamaan.

#### **Tiistai 16.11.**

Tänään pyörittelin REST APIa ja listasin kaikkien eri entiteettien tallennusmuodot ja mietin, miten saan kontaktien tallennuksen tuohon oikeaan muotoon. Moduulia on toivottu asiakkaan toimesta nimenomaan kontaktien kopiointiin, joten toteutan tuon kontaktien tallennuksen ensin, mutta moduuli tehdään niin, että siihen on helppo lisätä muutkin entiteettityypit (tuote, uutinen, tapahtuma). Sain loppupäivästä tiedot oikeaan muotoon, vain propertyt puuttuvat listasta.



## **Keskiviikko 17.11.**

Tämä päivä meni entiteettien propertyjen parissa. Moduulin asetustiedostoon laitetaan tieto siitä, minkä kategorian entiteettejä halutaan kopioida. Jokaisella kategorialla on omanlaisensa propertyt, joita voi olla paljonkin. Koitin tänään miettiä järkevintä tapaa järjestää ne kopiointia varten. Tiedot kopioitavista entiteeteistä ovat monimutkaisessa JSON-muodossa, ja tiedon saaminen JSON-tiedostosta vaatii pientä aivojumbppaa.

## **Torstai 18.11.**

Tänään muokkasin funktioiden toimintaa, yleisesti missä tapahtuu mitään. Propertyt tulevat nyt myös tietokannasta, mutta vielä pitää saada ne JSON-muotoon oikeilla id-numeroilla kopioitavan entiteetin id-numeroiden sijaan. Tässä tapahtuu asioita niin monessa tasossa ja moneen suuntaan, että välillä aivot tuntuvat räjähtävän. Olen ottanut tavaksi piirtää asioita paperille, jotta hahmotan niitä paremmin. Usein myös käsin kirjoittaminen jäsentää ajatuksia paremmin kuin asioiden laittaminen tietokoneen ruudulle ylös. Tänäänkin on tullut tätä hyödynnettyä.

## **Perjantai 19.11.**

Jatkoin tänään propertyjen säätämistä. Päivä oli hyvin katkonainen viikkopalaverista ja henkilökohtaisesta menosta johtuen, joten edistyminen oli hidasta. Uuden moduulin lisäksi korjailin myös erästä vanhempaa tehtävää, jossa huomattiin katselmoinnin yhteydessä muutama pieni korjattava kohta. Lisäksi oli WordPress-koulutus ja tiimin virkistysiltapäivä. Oli mukava tutustua niihinkin tiimiläisiin, jotka eivät ole aiemmin osuneet samaan aikaan toimistolle. Myös lähimpien työkavereiden kanssa tuli juteltua enemmän muistakin kuin työhön liittyvistä asioista.

## **Viikkoanalyysi – Kommunikaatio etätyössä**

Päätin viimeisen viikkoanalyysin kirjoittaa kommunikaatiosta etätyöaikana. Toimistollamme on siirrytty hybridimalliin, jossa toimistolla ollaan joinain päivinä viikosta ja muuten etänä. Yksi tiimiläinen on päivittäin toimistolla ja yksi aina etänä, mutta me muut kuljemme toimistolla vähintään perjantaisin.

Yhtäkkiä pandemian tuoma työntekijöiden hajaantuminen koteihinsa töihin on tuonut mukanaan haasteita työyhteisön ja tiimien yhteenkuuluvuuden ylläpitämiseen. Yhteenkuuluvuus on tärkeä inhimillinen perustarve, joka on yhteydessä sekä hyvinvointiin että motivaatioon. Se on kaksisuuntaista: ihminen kokee olevansa merkityksellinen osa ryhmää ja myös ryhmä on merkityksellinen hänelle. Ympäristö, joka tulee yhteenkuuluvuutta, antaa kokemuksen välittämisestä ja joukkoon kuulumisesta. Tarpeet yhteisöllisyyteen vaihtelevat, mutta jokainen tarvitsee riittävän kokemuksen yhteenkuuluvuudesta. (Sjöblom & Mäkikangas 2021.)

Etätyössä keskeistä osaa näyttelee verkkotapaamiset. Kasvokkain tapahtuvaan kohtaamiseen verrattuna verkkotapaamisissa on vaikeampi seurata puheenvuoroja ja pysyä keskusteluissa mukana. Kynnys pitää puheenvuoro voi myös nousta korkealle. Vuorovaikutus toimii paremmin, jos kaikki pitävät kameran päällä vähintään oman puheenvuoron aikana. Hybridikokouksiin tuo haasteita tasapuolinen osallistaminen. (Laivola & Marstio 2021.)

Oman kokemukseni mukaan asiapitoinen tieto kulkee hyvin myös etänä, mutta sellainen toisiin tutustumisen ja muista kuin työasioista jutustelu on kyllä hyvin minimaalista. Olemme viettäneet toimistolla nyt noin kymmenen työpäivää ja niinä päivinä olen tutustunut tiimikavereihin paremmin kuin aikaisempina kolmena kuukautena yhteensä. Uuden työntekijän integroituminen työyhteisöön vaatii suuria ponnisteluja etäaikana sekä itse työntekijältä että muilta tiimiläisiltä. Tiedän itse, että olisin voinut alusta asti osallistua keskusteluihin rohkeammin, mutta toisaalta olen perusluonteeltani vähän hitaasti lämpenevä, joten se ei ole minulle luontaista. Tiedän ja luotan, että ajan kuluessa alan uskaltaa olla enemmän oma itseni ja kommunikointi sitä myöten käy helpommaksi niin etänä kuin kasvotustenkin.

## 4 POHDINTA

Aloittaessani päiväkirjan kirjoittamista, kuvasin olevani taidoiltani aloitteleva toimija, mitä koen edelleen olevani. Huomaan kuitenkin kehittymistä siinä, että jos kymmenen viikkoa sitten uuden tehtävän saatuani tarvitsin siihen tarkan ohjeistuksen päästäkseni alulle, niin nyt tehtävät osoitetaan minulle ilman sen suurempia opastuksia. Tuotteen koodikanta on selkiytynyt mielessäni ja löydän sieltä ne tiedostot, joita minun täytyy muokata tehtävien suorittamiseksi. Tekemiseen on tullut lisää varmuutta, vaikka itseluottamukseni työn suhteen välillä onkin matala. Olen edelleen varmasti verrattain hidas ohjelmoija, mutta onnekseni saan kehittyä rauhassa eikä minulta odoteta liikoja. Saan myös itse vaikuttaa jonkun verran siihen, kuinka haastavia tehtäviä otan vastaan.

Epävarmuuteni ohjelmoijana on välillä vaikuttanut työn tekemiseen, koska inhoan ja pelkään virheitä. Olen kuitenkin ollut melko rohkea ottamaan vastaan hankalammaltakin kuulostavia työtehtäviä, vaikka en olekaan valmiiksi tiennyt, miten se tehdään. Epävarmuuttani varmasti kompensoi uteliaisuuteni ja haluni oppia jatkuvasti uutta. Alkutilanteessa kerroin, että virheidensietokyvyttömyyteni esti minua usein kysymästä neuvoa ongelmatilanteissa. Edelleen minulla on suuri kynnys kysyä neuvoa Slackin ryhmissä, koska silloin osaamattomuuteni jää sinne ikuisiksi ajoiksi näkyville. Tiedän, että tällainen ajattelu on omalle oppimiselleni ja työntekemiselleni vahingollista ja pyrin siitä jatkuvasti pääsemään eroon. Kasvotusten pyydän kyllä apua herkemmin, vaikka kynnys siihenkin on varsinkin aiemmin ollut suuri. Virheidensietokykyäni on myös kasvanut huomattavasti, että toisetkaan eivät ole täydellisiä ja muidenkin koodissa on virheitä. Jokainen tekee virheitä ja niitä pääsee joskus myös tuotantoon siitä huolimatta, että jokainen muutos katselmoidaan kahden ohjelmoijan toimesta. Saamani palaute on kuitenkin ollut ainoastaan positiivista, joten oma tunteeni osaamattomuudestani on ainoastaan minun omassa päässäni. Olenkin yrittänyt opetella katsomaan tilannetta muiden silmin: olen aivan aloittelija, ensimmäisessä ohjelmointialan työssäni, joten ei minulta voi odottaa täydellisyyttä. Hyvinä päivinä huomaan, että pärjään olosuhteisiin nähden erinomaisesti, mutta huonoina päivinä on vaikea pyrkiä näkemään itsessäni hyviä puolia.

Syksyllä ottamamme tapa nähdä perjantaisin toimistolla on lisännyt varmuuttani vuorovaikutuksessa toisten kanssa. Uskallan näyttää omaa persoonaani enemmän ja tuoda tiimiin minun tapani olla ja toimia. Aiemmin, kun en ollut tavannut kuin esimieheni työharjoittelun alkaessa toukokuussa, toin itseäni hyvin vähän esille esimerkiksi viikkopalaverissa. Nyt, kun viikkopalaverit on pidetty jo kolmen kuukauden ajan toimistolla, on paljon helpompi kertoa omia mielipiteitä asioista sen sijaan,

että vain kertoisi omat viikon asiansa. Olen yleensäkin hieman hidas ottamaan omaa paikkaani jo valmiissa ryhmässä ja tämä etäaika on kyllä hankaloittanut sitä entisestään.

Tein tämän päiväkirjaseurannan aikana kymmentä eri tehtävää, joista kahdeksan tuli valmiiksi. Eniten aikaa käytin nimi-arvopariprojektin korjaamiseen, vaikka sitä ei sitten lopulta edes tehty loppuun. Kymmenes aloitettu tehtävä on vielä kesken. Tehtävät olivat keskenään mukavan erilaisia. Osassa tarvittiin enemmän jQueryä, osassa PHP-ohjelmointia ja joissain enemmän tietokannan kanssa toimimista. Suurin osa asioista on ollut jollain tasolla tuttuja, mutta niiden osaaminen on ollut pintapuolista. Uutena asiana tässä työssä minulle on tullut PostgreSQL, joka tosin on hyvin samankaltainen jo aiemmin opiskelemani MySQL:n kanssa. Myös Gitin käyttö on muuttunut varmemmaksi ja osaan käyttää siitä enemmän, kuin pelkkiä peruskäyttötoimintoja. Ehkä vierain asia minulle tässä työssä on ollut Linuxin käyttö. Kaikki palvelimet pyörivät Linuxilla, joten sen käyttö on syytä osata. Minun ei ole vielä toistaiseksi tarvinnut kovin paljoa käyttää muuta kuin meidän omaa kehityspalvelinympäristöämme, jossa osaan käyttää perustoimintoja, kuten kansioista toiseen siirtymisen ja tietokannan avaamisen. Olen kuitenkin oppinut Linuxin toimintaperiaatetta pikkuhiljaa enemmän ja enemmän ja uskon, että siitä on hyötyä enemmän kuin yksittäisten komentojen ulkoa muistamisesta. Kun ymmärtää, miten komennot rakennetaan, on helpompi muistaa erillisiä komentojakin.

Päiväkirjan ja viikkoanalyysien kirjoittaminen on ollut siinä mielessä valaisevaa, että on itsekin helpompaa nähdä oma kehitys, kun lähtötilannekin on kirjoitettuna ylös. Mikäli tätä päiväkirjaa ei olisi, uskoisin todennäköisesti, että kehitystä ei ole tässä ajassa tullut juuri lainkaan. Luulen, että tämän opinnäytetyön suurin merkitys on omassa henkilökohtaisessa kasvussani ammattilaiseksi. Se on valanut minuun uskoa siihen, että minäkin osaan, vaikka välillä tuskailen ja taistelen ongelmien parissa. Voin jatkossa palata tähän päiväkirjaan ja verrata sen hetkistä osaamista tämän hetken tasoon ja toivottavasti todeta, että kehitystä on tapahtunut runsaasti.

## LÄHTEET

Bin Uzayr, Sufyan 2016. Learning WordPress REST API. Birmingham: Packt Publishing. Hakupäivä 21.11.2021. O'Reilly. Vaatii käyttöoikeuden.

Duckett, Jon 2011. HTML & CSS: Design and Build Websites. Hoboken: Wiley. Hakupäivä 10.10.2021. O'Reilly. Vaatii käyttöoikeuden.

Ekman, Tiina 2017. Huijarisyndrooma. Helsinki: Minerva. Hakupäivä 6.11.2021. Bookbeat. Vaatii käyttöoikeuden.

Ferrari, Luca & Pirozzi, Enrico 2020. Learn PostgreSQL. Birmingham: Packt Publishing. Hakupäivä 17.10.2021. O'Reilly. Vaatii käyttöoikeuden.

Franklin, Jack & Ferguson Russ 2017. Beginning jQuery: From the Basics of jQuery to Writing your Own Plug-in. Second edition. New York City: Apress. Hakupäivä 17.9.2021. O'Reilly. Vaatii käyttöoikeuden.

Google Analytics 2021. Hakupäivä 16.10.2021. <https://analytics.google.com/analytics/>.

Google Marketing Platform 2021. Hakupäivä 5.10.2021. <https://marketingplatform.google.com/about/analytics/>.

json.org. Introducing JSON. Hakupäivä 21.11.2021. <https://www.json.org/>.

Kline, Kevin, Obe, Regina O. & Hsu, Leo S. 2022. SQL in a Nutshell, 4th Edition (Early release). Sebastopol: O'Reilly Media, Inc. Hakupäivä 17.10.2021. O'Reilly. Vaatii käyttöoikeuden.

Koushik, Vikash 2020. 13 Git Commands Every Developer Must Know. Hakupäivä 30.10.2021. <https://zepel.io/blog/13-git-commands/>.

Laivola, Taina & Marstio, Tuija 2021. Etättyö vaatii yhteisiä pelisääntöjä. Hakupäivä 22.11.2021. <https://esignals.fi/pro/2021/05/11/etatyo-vaatii-yhteisia-pelisaantoja/>.

Larsen, Rob 2013. Beginning HTML and CSS. Birmingham: Wrox. Hakupäivä 10.10.2021. O'Reilly. Vaatii käyttöoikeuden.

Libert, Jesse & Galloway, Jon 2021. Git for Programmers. Birmingham: Packt Publishing. Hakupäivä 30.10.2021. O'Reilly. Vaatii käyttöoikeuden.

Orpana, Mari 2017. Google Analytics – mitä sillä tekee? Hakupäivä 5.10.2021. <https://www.kupli.fi/google-analytics-mita-silla-tekee/>.

Seobility. REST API. Hakupäivä 21.22.2021. [https://www.seobility.net/en/wiki/REST\\_API](https://www.seobility.net/en/wiki/REST_API).

Sjöblom, Kirsi & Mäkikangas, Anne 2021. Rapauttiko etätyö yhteisöllisyyden? Hakupäivä 22.11.2021. <https://www.tuni.fi/alustalehti/2021/05/26/rapauttiko-etatyo-yhteisollisyyden/>

Tatroe, Kevin & MacIntyre Peter 2020. Programming PHP. Sebastopol: O'Reilly Media, Inc. Hakupäivä 26.9.2021. O'Reilly. Vaatii käyttöoikeuden.

The Economic Times 2021. Definition of 'Debugging'. Hakupäivä 25.10.2021. <https://economictimes.indiatimes.com/definition/debugging>.

Thomas, David & Hunt, Andrew 2019. The Pragmatic Programmer: your journey to mastery, 20th Anniversary Edition, 2nd Edition. Boston: Addison-Wesley Professional. Hakupäivä 25.10.2021. O'Reilly. Vaatii käyttöoikeuden.

Tsitoara, Mariot 2019. Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer. New York City: Apress. Hakupäivä 30.10.2021. O'Reilly. Vaatii käyttöoikeuden.

Udemy. Hakupäivä 24.10.2021. <https://www.udemy.com/>. Vaatii käyttöoikeuden.