

**SAVONIA**

ammattikorkeakoulu

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO  
TEKNIKAN JA LIIKENTEEN ALA

# RAKENNUSHANKKEEN YHTEENVE- TODOKUMENTTI 10PRO-JÄRJES- TELMÄÄN

TEKIJÄ:

Oskari Rökkänen

Koulutusala Tekniikan ja liikenteen ala	
Tutkinto-ohjelma Tietotekniikan tutkinto-ohjelma	
Työn tekijä Oskari Röppänen	
Työn nimi Rakennushankkeen yhteenvedodokumentti 10Pro järjestelmään	
Päiväys 4.4.2022	Sivumäärä/Liitteet 24
Toimeksiantaja/Yhteistyökumppani(t) Weego Software	
Tiivistelmä <p>Projektin tavoitteena oli tehdä eräänlainen yhteenvedodokumentti 10pro järjestelmään, jota eri rakennusyri-tykset käyttävät avuksi työmailla. 10pro on web-sovellus, joka tarjoaa edistykselliset työnjohtamisen sähköi-set työkalut. Yhteenvedodokumentin tavoitteena oli muokata 10pro hallintajärjestelmään tulevaa dokumen-taatiota hankkeen vaatimusten mukaisesti ja niin että se on työmaan vastuuhenkilöiden käytettävissä. Li-säksi vastuulliset eri osa-alueiden työnjohtajat pystyisivät omien rooliensa mukaan kuittaa-maan tehtäviä valmiiksi.</p> <p>Koodieditorina toimi Visual Studio Code ja ohjelmointikielinä toimivat PHP, JavaScript, CSS ja HTML. Projek-tiin kuului myös tietokantataulujen suunnittelu ja luonti kyseiseen yhteenvedodokumenttiin. Opinnäytetyö tehtiin ja toteutettiin Weego Softwaren toimistolla ja välineillä. Valmis projekti tarjoaa tilaajalle tarkastuslis-tan, jonne voi työmaakohtaisesti tallettaa erilaisia dokumentteja kuten liitetiedostot, käyttäjät, työnjohtajat, työvaiheet, työvaiheiden kuittaukset ja muut oleelliset asiat mitä kyseisellä työmaalla halutaan dokumen-toida.</p> <p>Tuloksena saatiin toimiva yhteenvedodokumentti, joka menee myöhemmin tuotantoon ja liitetään osaksi 10Pro järjestelmää. Kaikki halutut toiminnot saatiin tehtyä ja yhteenvedodokumentista saatiin hyvin dynaa-minen ja helppokäyttöinen. Yhteenvedodokumentin vienti tuotantoon tapahtuu myöhemmin, koska raken-nusalan yrittäjien on katsottava yhteenvedodokumentti läpi ja he antavat lopullisen palautteen onko yhteen-vedodokumentissa kaikki mitä he tarvitsevat. Yhteenvedodokumentti kuitenkin tarjoaa jo nyt paljon eri toi-mintoja, joka helpottaa rakennushankkeen dokumentointia ja koko työmaan etenemisen seurantaa.</p>	
Avainsanat Web-ohjelmointi, PHP, JavaScript, html, CSS, jQuery	

Field of Study Technology, Communication and Transport	
Degree Programme Degree Programme in Information Technology	
Author(s) Oskari Röppänen	
Title of Thesis Construction project summary document for 10Pro system	
Date 4.4.2022	Pages/Appendices 24
Client Organisation /Partners Weego Software	
<p><b>Abstract</b></p> <p>The aim of the thesis was to make a summary document for the 10pro system that is used by different construction companies to help on construction sites. 10pro is a web application that provides advanced electronic tools for work management. The aim of the summary document was to modify the documentation used in the 10pro management system in accordance with the requirements of the client organization, Weego Software. The goal was also to make the system available to the persons in charge of the site. In addition, the responsible supervisors in the various areas would be able to sign their area specific tasks completed.</p> <p>The code editor was Visual Studio Code, and the programming languages were PHP, JavaScript, CSS and HTML. The project also included the design and creation of database tables for the summary document. The thesis was planned and implemented using the tools and premises of Weego Software. The completed project provides the client with a checklist where various documents can be stored for each site. The documents may include attachments, information about users, site managers, various phases of work, phases signed as completed and other essentials that are to be documented at the site.</p> <p>The result was a workable summary document. All the desired functions were included and the summary document was made very dynamic and easy to use. The summary document will later go into production and be integrated into the 10Pro system. Before that, construction entrepreneurs must look through the summary document and give the final feedback on it. However, the summary document already offers many different functions that make it easier to document the construction project and monitor the progress of the entire site.</p>	
<p><b>Keywords</b> Web-programming, php, JavaScript, HTML, CSS, jQuery</p>	

## ESIPUHE

Opinnäytetyöksi toivoin mielenkiintoista aihetta ja sen löytämiseen menikin pidempi tovi osasyynä tämänhetkinen koronatilanne. Aihe kuitenkin löytyi ja tästä isot kiitokset Weego Softwarelle niin aiheesta, kokemuksesta, laitteista sekä työtiloista. Weego Software tarjosi loistavan kokemuksen tietotekniikan alan töistä ja Web-sovelluksen kehittämisestä isommassa mittakaavassa.

Henkilökohtaiset kiitokset Pasi Ollikaiselle joustavuudesta ja aiheesta sekä kiitokset myös Antti Koskiselle vinkeistä ja avusta projektin aikana.

Opinnäytetyö tarjosi loistavan kuvan ohjelmistokehittäjän töistä ja tarjosi massiivisen tietopaketin web-ohjelmoinnista sekä kokonaisuuden rakentamisesta. Joka päivä oppi jotain uutta mikä teki tästä työstä mielenkiintoisen ja tähän pätee sanonta mitä koulussa kuuli ”koodarina uuden oppiminen ei koskaan lopu”.

## SISÄLTÖ

1	JOHDANTO .....	9
1.1	Aihe .....	9
1.2	Tilaaaja .....	9
2	PROJEKTIN ALOITUS JA SUUNNITTELU .....	10
2.1	Projektin aloitus .....	10
2.2	Suunnittelu .....	11
2.3	Aikataulu ja ajankäyttö .....	11
2.4	Projektin vaiheet .....	12
2.5	Kehitysympäristö ja sen luominen .....	13
2.6	Riskit ja niiden hallinta .....	13
3	PROJEKTIN TOTEUTUS .....	14
3.1	Yhteenvetodokumentin tietokantataulujen rakentaminen .....	14
3.2	Yhteenvetodokumentin muodostaminen ja muotoilu .....	15
3.3	Yhteenvetodokumentin toiminnallisuus ja dynaamisuus .....	16
3.4	Yhteenvetodokumentin muita toimintoja .....	19
4	PROJEKTIN TULOS .....	21
5	YHTEENVETO JA POHDINTA .....	22
6	JATKOKEHITYS .....	23
7	LÄHDELUETTELO.....	24

## KUVALUETTELO

Kuva 1. Prototyyppi.....	10
Kuva 2. Prototyyppi.....	11
Kuva 3 Taulurakenne hahmotelma .....	14
Kuva 4. Taulun luontilause.....	14
Kuva 5. HTML:n liittäminen PHP:n avulla .....	15
Kuva 6. Miltä kuvan 4 koodi näyttää käyttäjälle CSS määrityksen jälkeen .....	15
Kuva 7. Yhteenvedodokumentti .....	16
Kuva 8. Rakennushankkeeseen ryhtyvät .....	17
Kuva 9. Rakennushankkeeseen ryhtyvien lisäys JavaScript.....	17
Kuva 10. Rakennushankkeeseen ryhtyvien lisäys PHP .....	18
Kuva 11. Front End(asiakaspuoli) Back End(palvelinpuoli).....	18
Kuva 12. Työnjohtajaosio .....	19
Kuva 13. Valintalista työmaan henkilöstöstä .....	19
Kuva 14. Työlistan toiminta.....	19
Kuva 15. Työtehtävän merkitseminen valmiiksi .....	20
Kuva 16. Liitteet osio.....	20

## TAULUKKOLUETTELO

Taulukko 1. Alustava aikataulu .....	12
--------------------------------------	----

## KÄYTETTY SANASTO

- PHP – (lyhenne sanoista PHP: Hypertext Preprocessor) on ohjelmointikieli, jota käytetään web-palvelinympäristöissä dynaamisten web-sivujen luonnissa. Komentosarjakieli, jossa ohjelmakoodi tulkitaan vasta ohjelman suoritusvaiheessa.
- JavaScript – on komentosarja- tai ohjelmointikieli, jonka avulla voit toteuttaa monimutkaisia ominaisuuksia verkkosivuilla. JavaScriptin tärkein sovellus on mahdollisuus lisätä verkkosivuille dynaamista toiminnallisuutta. **\*\*liite\*\*** [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)**\*\***
- HTML – on lyhenne sanoista Hyper Text Markup Language, jota käytetään verkkosivujen ja verkkosovellusten luomiseen. On avoimesti standardoitu merkintäkieli, jolla voidaan kuvata hyperlinkkejä sisältävää tekstiä eli hypertekstiä **\*\*liite\*\***<https://www.javatpoint.com/what-is-html> **\*\***
- CSS – eli Cascading Style Sheets on erityisesti verkkosivuille kehitetty tyylisivu. Kuvaa kuinka HTML-elementit näytetään näytöllä. Voi hallita useiden verkkosivujen asettelua kerralla eli säästää paljon työaika **\*\*linkki\*\***[https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)**\*\***
- jQuery – on nopea, pieni, monipuolinen Javascript-kirjasto. Sen syntaksi on pyritty tekemään helposti ymmärrettäväksi mikä tekee siitä suosittua. **\*\*liite\*\***<https://jquery.com>**\*\***
- Front-End – on verkkosivuston graafisen käyttöliittymän kehittämistä HTML:n, CSS:n, ja JavaScriptin avulla, jotta käyttäjät voivat tarkastella verkkosivua ja olla vuorovaikutuksessa sen kanssa
- Back-End – viittaa sovelluksen tai ohjelman koodin osiin, jotka mahdollistavat sen toiminnan ja joita käyttäjä ei voi käyttää. Suurin osa tiedoista ja käyttösyntaksista tallennetaan ja niitä käytetään tietokonejärjestelmän takaosassa. Yleensä koodi koostuu yhdestä tai useammasta ohjelmointikielestä. **\*\*liite\*\***<https://whatis.techtarget.com/definition/front-end>**\*\***
- MySQL – on suosituin avoimen lähdekoodin SQL-tietokannan hallintajärjestelmä. Oraclen Corporation kehittämä, jakama ja tukema



## 1 JOHDANTO

### 1.1 Aihe

Opinnäytetyöni aiheena on lisätä 10Pro järjestelmään uusi ominaisuus rakennusvalvonnan vaatimista hankkeen yhteenvetodokumentaatiosta. Työn toteutus pitää sisällään 10Pro hallintaportaalien tulevan dokumentaation muokkaamisen hankkeen vaatimusten mukaisesti, joka on työmaan vastuuhenkilöiden käytössä. Vastuuhenkilöt pystyvät omien rooliensa mukaisesti kuittaamaan tehtäviä/vaiheita valmiiksi. Uusi ominaisuus on käytännössä tarkastuslista, jonne kerätään kaikki dokumentaatiot kuten tehtävät, vaiheet, liitteet, hankkeen tiedot, työnjohtajat, hankkeeseen ryhtyvät. Tarkastuslista on työmaakohtainen, jokaisella työmaalla on oma ID, joka näyttää tietyn työmaan tarkastuslistan ja sinne tallennetut tiedot ja dokumentit.

Toteutettava kokonaisuus koostuu www-sivuston päivittämisestä (HTML, CSS, JavaScript, jQuery), tietokannan suunnittelusta ja muokkaamisesta (MySQL) sekä palvelinpään ohjelmointi vaadittujen toimintojen ja dokumenttien osalta (PHP).

Tarkastuslista edesauttaa työmaan vaiheiden seuraamista ja dokumenttien säilyttämistä sekä koko työmaan ylläpitämistä. Vastuuhenkilöiden on helppo seurata tehtävien sekä vaiheiden valmistumista ja liitteet ja yhteystiedot ovat helposti saatavilla.

### 1.2 Tilaaja

Opinnäytetyön tilaajana toimii Weego Software. Weego Software on 2010 vuonna perustettu kuopiolainen ketterä ohjelmistotuottaja, joka on profiloitunut korkean osaamisen kautta luottotoimijaksi terveydenhuollon ja operaattorien liiketoiminnan alueella. Henkilöstöä yrityksellä on tällä hetkellä 14. Weego keskittyy ohjelmistojen suunnitteluun ja valmistukseen. Yrityksellä ei ole omia tuotteita vaan se keskittyy ohjelmistojen alihankintaan ja konsultaatioon. Isoimpina asiakkaina Elisa, New Icon sekä iso joukko rakennusteollisuuden toimijoita. Juuri 10Pro järjestelmä on suunniteltu rakennusteollisuus yritysten käyttöön.

## 2 PROJEKTIN ALOITUS JA SUUNNITTELU

### 2.1 Projektin aloitus

Projekti alkoi palaverilla tilaajan kanssa, jossa suunniteltiin tilaajan kanssa millaisen uuden ominaisuuden he haluaisivat 10Pro järjestelmään. Kyseisestä ominaisuudesta oli alustava prototyyppi missä ei ollut minkäänlaista toiminnallisuutta, ja ulkoasukaan ei ollut yhteensopiva osaksi 10Pro järjestelmää. Prototyypin avulla kuitenkin sai osviittaa mitä kyseisen ominaisuuden haluttiin sisältävän. Palaverissa käytiin myös hieman läpi minkälainen taulurakenne kannattaisi uudelle ominaisuudelle rakentaa, jotta se olisi hyvä osana isompaa kokonaisuutta.

Uuden ominaisuuden haluttiin olevan selkeä ja helppokäyttöinen käyttäjille. Ominaisuus on siis eräänlainen tarkastuslista jonne työmaan dokumentit, henkilöt, työtehtävien kuittaukset ja liitteet voidaan tallentaa. Eli käytännössä kaikki tarvittava tieto olisi helposti saatavilla ja nähtävillä yhdessä paikassa.

Web-ohjelmoinnista itselläni on hieman aikaisempaa kokemusta koulun puolelta sekä työharjoittelusta. Työharjoittelu kuitenkin perustui Laravel Frameworkkiin kun taas kyseisessä projektissa Frameworkkina toimi JQuery. Projekti oli myös paljon isompi kokonaisuus kuin mitä itse olen koskaan tehnyt ja sisälsi paljon asioita, joita piti ottaa huomioon. Projektin alku menikin tutustuesssa 10Pro järjestelmään ja itse koodiin. Tämä veikin paljon aikaa, jotta näki kuinka 10Pro oli rakennettu ja näki hieman ohjelmointiratkaisuja, joita pystyisi sitten hyödyntämään omassa projektissa. Koodialustaksi valitsin Visual Studio Coden, koska se oli ennestään tuttu ja mielestäni helppokäyttöinen.

**Lupatunnus**

**Rakennuspaikka**

Kaupunginosa/kylä	Kortteli/tilan nimi	Tontin nro/ tilan RN:o
<input type="text"/>	<input type="text"/>	<input type="text"/>

**Osoite**

**Tallenna tiedot**

**Rakennushankkeeseen ryhtyvät**

<input type="text" value="Nimi"/>	<input type="text" value="Puhelinnumero"/>	<b>Lisää</b>
-----------------------------------	--	--------------

**Tallenna**

**Työnjohtajat**

Rooli: <input type="text" value="Vtj"/>	<input type="text" value="Nimi"/>	<input type="text" value="Puhelinnumero"/>	<b>Lisää</b>
---	-----------------------------------	--	--------------

**Tallenna**

Kuva 1. Prototyyppi

Rakentaminen ja rakennustyömaa	
Rakennuslupa ⚠️	<a href="#">Lisää</a>
Aikataulu ⚠️	<a href="#">Lisää</a>
Asiat (siisteys, puhtaus, mittaukset, olosuhteet, havainnot, työturvallisuus) ⚠️	<a href="#">Lisää</a>
Runko ja kantavat rakenteet sekä liittymät ⚠️	<a href="#">Lisää</a>
Alueet sekä liittymät ⚠️	<a href="#">Lisää</a>
Alapohjat sekä liittymät ⚠️	<a href="#">Lisää</a>
Välipohjat sekä liittymät ⚠️	<a href="#">Lisää</a>
Yläpohjat sekä liittymät ⚠️	<a href="#">Lisää</a>
Talotekniikka sekä liittymät ⚠️	<a href="#">Lisää</a>

Kuva 2. Prototyyppi

## 2.2 Suunnittelu

Suunnittelun alussa apuna toimi juuri edellä mainittu prototyyppi, joka auttoi paljon projektin hahmottamisessa. Isona osana suunnittelua oli tietokannan taulurakenne niin, että se olisi yhteensopiva 10Pro:ssa käytettyihin tauluihin ja linkityksiä taulujen kesken voisi käyttää. Suunnittelun apuna toimi myös itse 10Pro järjestelmä, josta sai hyvän kuvan minkälaisen uudesta ominaisuudesta tekisi visuaalisesti. Suunnittelun tukena toimi myös Weegon työntekijä Antti Koskinen, joka antoi hyviä ohjeita mistä kannattaa aloittaa ja mitä tekniikoita he ovat käyttäneet 10Pro järjestelmässä. Suunnitelman valmistuttua kävimme vielä tilaajan kanssa läpi projektin, jonka jälkeen työ alkoi rakentamalla kehitysympäristö.

## 2.3 Aikataulu ja ajankäyttö

Projektin aikataulu oli melko avoin mutta alustava aikataulutus oli, että työn alkaessa marraskuun loppupuolella käytettäisiin ensimmäiset pari viikkoa suunnitteluun, 10Pro järjestelmään tutustumiseen sekä kehitysympäristön rakentamiseen. Joulukuu olisi taas kehitystyötä ja ihan puhdasta koodausta. Tavoitteena oli, että joulukuussa saisi jo jonkinlaisen version uudesta ominaisuudesta testaukseen. Tammikuu taas olisi tarkoitettu projektin viimeistelyyn, virheiden korjaukseen ja liittämisen isompaan kokonaisuuteen. Varavaihtoehtona oli myös, että jos projekti hieman viivästyy, helmikuu on vielä aikaa saada projekti valmiiksi ja tuotantoon.

<b>Aikataulu (kk)</b>	marraskuu	joulukuu	tammikuu	helmikuu
<b>Työ</b>	-Alustava suunnitelma  -Kehitysympäristö valmis  -Tietokantasuunnitelma	-Tietokanta valmis  -Kehitystyö, koodaus ja kokonaisuuden rakentaminen  -Ensimmäinen testiversio valmis	-Kokonaisuuden vieminen loppuun  -Bugien korjaus  -Testausta kehitysympäristössä	-Viimeistely  -Liittäminen isompaan kokonaisuuteen

Taulukko 1. Alustava aikataulu

Aikataulu oli melko joustava juuri sen takia, koska muita opintoja ei enää ollut jäljellä. Eli opinnäytetyöhön pystyi keskittymään sataprosenttisesti. Tämä helpotti ajankäyttöä niin että projektia pystyi tekemään täyspäiväisesti. Viikkokohtainen ajankäyttö oli käytännössä kuin normaali työviikko noin 8 tuntia maanantaista perjantaihin. Tämä teki työstä järjestelmällistä ja helpotti työn etenemistä. Täyspäiväisesti opinnäytetyön tekeminen toi paljon joustavuutta aikatauluun, kun työtä tehdessä törmäsi ongelmatilanteisiin. Vapaa-aikaa kuitenkin on oltava, jotta työ edistyisi hyvällä tahdilla. Uudet tekijät löytävät työelämän vaatimuksissa ja oppilaitoksensa tarjonnassa suuren kuilun, joka täytyisi jotenkin kuroa. Silti ohjelmointi ei voi olla elämän ainut sisältö (Suoranta, 2020). Aikataulua auttoi myös se, että opinnäytetyötä pystyi tekemään Weegon toimistolla, jossa oli aina apu saatavilla. Tästä iso kiitos Weegolle.

## 2.4 Projektin vaiheet

Projektin vaiheina toimivat suunnittelu, johon kuului tietokantataulujen suunnittelu, uuden ominaisuuden suunnittelu visuaalisesti sekä toiminnallisesti. Suunnittelun kuului myös tutustuminen 10Pro järjestelmään, jotta näki hieman, miten muut ominaisuudet oli tehty järjestelmään ja ylipäätään tutustui hieman web-ohjelmointiin.

Seuraava vaihe oli itse työn toteutus, johon oli varattu eniten aikaan. Tähän kuului itse uuden ominaisuuden luonti ja kehitys. Kehityksessä täytyi ottaa huomioon se, että ominaisuus olisi käytännöllinen 10Pro järjestelmään. Toteutuksen aikana pystyi kokeilemaan eri ratkaisuja, miten uusi ominaisuus toteutetaan ja keskustelemaan tilaajan kanssa mikä näistä ratkaisuista oli paras. Työn toteutusta helpotti hyvä suunnitelma, tutustuminen 10Pro järjestelmään sekä jatkuva kommunikointi tilaajan kanssa. Kommunikaatio auttoi siinä, että ei tehnyt niin sanottua turhaa työtä vaan ensin esitti

oman ideansa. Tämän jälkeen keskusteltiin tilaajan kanssa kuinka erilaiset ratkaisut uudessa ominaisuudessa kannattaisi toteuttaa.

Viimeinen vaihe oli uuden ominaisuuden testaus ja mahdollisten virheiden korjaus. Testauksen yhteydessä saattoi myös hahmottaa uusia ideoita mitä muuttaa uudessa ominaisuudessa. Käytännössä kun kaikki ominaisuuden kohdat näkivät osana isompaa kokonaisuutta, saattoi hyvin huomata kuinka esimerkiksi jokin uuden ominaisuuden kohta kannattaisi muuttaa erilaiseksi. Testaus toimi siis myös eräänlaisena kehitysalustana, jossa versiota päivitettiin käytettävämmäksi.

## 2.5 Kehitysympäristö ja sen luominen

Kehitysympäristö luominen tehtiin Weegon tarjoamalle laitteelle ja heillä oli hyvä ohjeistus, kuinka se luodaan. Käyttöjärjestelmällä ei ollut väliä mutta kyseisellä laitteella se oli Ubuntu. Kehitysympäristön asennus alkoi asentamalla komponentit apache2, MySQL ja PHP. Tämän jälkeen tehtiin käyttäjä ja asennettiin 10Pro. 10Pro haettiin versionhallinnasta ja ajettiin konfiguraatioscripti. Sitten tuotiin tietokannan sisältö tuotannosta. Tietokantana käytettiin testitietokantaa, joka vastaa rakenteeltaan oikeaa tietokantaa, jota käytetään 10Pro:ssa. Tämän jälkeen kehitysympäristöön ajettiin vielä muut tarvittavat sofit ja kehitysympäristö oli valmis käytettäväksi. Kehitysympäristö toimi erittäin hyvin ja täytti kaikki kriteerit, että uuden ominaisuuden kehitys onnistui.

## 2.6 Riskit ja niiden hallinta

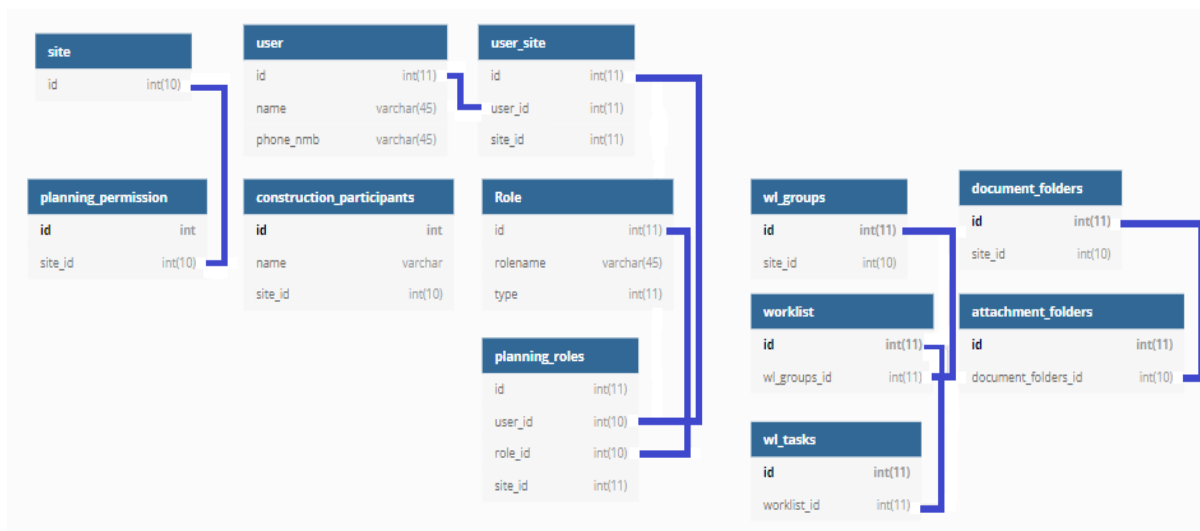
Suunnittelun ohessa pohdittiin mahdollisia riskejä ja kuinka näiltä vältetään. Kehitysympäristö poistaa paljon riskejä, koska siellä testatessa uutta ominaisuutta pystyy hyvin karsimaan isot riskitekijät. Tietokantataulujen rakentaminen uuteen ominaisuuteen niin, että ne ovat yhteensopivat osaksi 10Pro järjestelmää on riski koska itselläni ei ollut aiempaa kokemusta aiheesta. Riskitekijäksi myös osoittautui 10Pro järjestelmässä käytetyt ohjelmointiratkaisut. Jatkoa ajatellen on hyvä, että uuden ominaisuuden ohjelmointiratkaisut vastaavat 10Pro järjestelmän ratkaisuja, joten tätä riskiä käsitellään siten että käydään Weegon työntekijöiden kanssa läpi 10Pro järjestelmän ratkaisuja ja sovelletaan niitä uuteen ominaisuuteen.

### 3 PROJEKTIN TOTEUTUS

#### 3.1 Yhteenvedodokumentin tietokantataulujen rakentaminen

Uuden ominaisuuden eli yhteenvedodokumentin valmistelu alkoi tekemällä siihen vaadittavat tietokantataulut MySQL:ään. Ja MySQL:ää tarvitaan useastakin syystä. Se toteuttaa jäsennellyn kyselykielen (SQL), jota voidaan käyttää tietokannan tietojen lisäämiseen, poistamiseen ja muokkaamiseen. Komentoja, kuten ADD, DROP, INSERT, UPDATE, voidaan käyttää MySQL:n kanssa. PHP-koodi voidaan liittää MySQL-komentoihin tietokantaan pääsemiseksi, jotta se voi tuottaa osia tai koko verkkosivun tietokantaan tallennetuista tiedoista (Education-wiki, 2022).

Suunniteltaessa tuli ilmi, että tauluja tarvitaan kuusi. Alla oleva kuva näyttää hieman minkälaisia linkityksiä taulut tarvitsivat, jotta kaikki toimii niin yhteenvedodokumentissa kuin myös osana itse 10Pro järjestelmää.



Kuva 3 Taulurakenne hahmotelma

Kaikille tauluille tehdään oma luontilause, joka rakentaa taulun tietokantaan. Taulut luodaan MySQL:n komentorivityökalun avulla. Työkalun avulla voi suorittaa SQL-kyselyitä ja tutkia niiden tuloksia (Laaksonen, 2009).

```

planning_permission | CREATE TABLE `planning_permission` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `site_id` int(10) unsigned NOT NULL,
  `authorization_code` varchar(45) DEFAULT NULL,
  `city` varchar(45) DEFAULT NULL,
  `block_nmb` varchar(45) DEFAULT NULL,
  `site_nmb` varchar(45) DEFAULT NULL,
  `address` varchar(45) DEFAULT NULL,
  `created_ts` datetime NOT NULL,
  `last_mod_by` varchar(45) DEFAULT NULL,
  `status` varchar(45) DEFAULT 'active',
  `last_mod_ts` datetime DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `site_id` (`site_id`),
  CONSTRAINT `planning_permission_ibfk_1` FOREIGN KEY (`site_id`) REFERENCES `site` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=51 DEFAULT CHARSET=latin1 |
  
```

Kuva 4. Taulun luontilause

### 3.2 Yhteenvedodokumentin muodostaminen ja muotoilu

Yhteenvedodokumentin muodostaminen tehdään käyttämällä HTML elementtejä halutulla tavalla, jotta halutut kentät, otsikot ja kaikki muu tarpeellinen on nähtävissä käyttäjälle. Prototyypistä on helppo hahmottaa mitä kaikkia elementtejä tarvitaan ja missä järjestyksessä. Dokumentin muotoilu tehdään CSS:n avulla. Käytännössä HTML:ssä luodaan vain elementit ja CSS:ssä määritellään kaikki milta kyseiset elementit näyttävät ja rakentuvat verkkosivulle. HTML-tiedostosta pystytään viittamaan CSS-tiedostoon ja tämän avulla saadaan oikeat muotoilut verkkosivulle. Isoimpana etuna erillisistä tyylitiedostoista käytettäessä on se, että HTML-tiedostot pysyvät selkeinä ja muutkin HTML-tiedostot voivat käyttää kyseistä tyylitiedostoa. Tällä tavoin laajemman verkkosivuston muotoilut voidaan laittaa yhteen tyylitiedostoon. (Commons, 2019).

Opinnäytetyössä HTML ratkaisu on hieman erilainen. Erillistä HTML-tiedostoa ei ole vaan HTML-elementit luodaan JavaScriptin sisällä lennosta ja sitten liitetään PHP:n avulla sivun sisältöön. HTML syntaksi on täysin samanlaista kuin normaalissa HTML-tiedostossa. JavaScriptiin luodaan vain muutuja, jonka sisään lisätään HTML-elementtejä ja tämä muuttuja liitetään haluttuun sisältöön.

```

var docdata = ('');
docdata += '<div class="content" data-view="work-list">';
docdata += '<div id="work-wrapper">';
docdata += '<ul class="work-list">';
docdata += '<div class="title">Lupatunnus</div>';
docdata += '<div class="inp"><input id="lupatunnus" type="text"></div>';
docdata += '<div class="title" style="padding-bottom:1em;">Rakennuspaikka</div>';
docdata += '<div class="inp" style="margin-left:1%;width:90%;">';
docdata += '<div class="title" style="width:32%;">Kaupunginosa/kylä</div>';
docdata += '<div class="title" style="width:32%;">Kortteli/tilan nimi</div>';
docdata += '<div class="title" style="width:32%;">Tontin nro/ tilan RN:o</div>';
docdata += '<div class="inp" style="width:32%;"><input id="kaupunginosa" type="text"></div>';
docdata += '<div class="inp" style="width:32%;"><input id="kortteli" type="text"></div>';
docdata += '<div class="inp" style="width:32%;"><input id="tontti" type="text"></div>';
docdata += '<div class="title">Osoite</div>';
docdata += '<div class="inp"><input id="osoite" type="text" style="width:50%;"></div>';
docdata += '</div>';
docdata += '<br>';
docdata += '<div><input type="button" name="save" style="background-color:#00838E; float:right; margin-top:1em; padding:0.5em;" class="btn btn-primary" value="Tallenna tiedot" id="butsave"></div><br>';
docdata += '<div><input type="button" name="update" style="background-color:#00838E; float:right; margin-top:1em; padding:0.5em;" class="btn btn-primary" value="Päivitä tiedot" id="butupdate"></div>';

$('#content').empty().append(docdata);

```

Kuva 5. HTML:n liittäminen PHP:n avulla

**Lupatunnus\***

**Rakennuspaikka**

Kaupunginosa/kylä*	Kortteli/tilan nimi*	Tontin nro/ tilan RN:o*
<input style="width: 95%;" type="text"/>	<input style="width: 95%;" type="text"/>	<input style="width: 95%;" type="text"/>

**Osoite\***

Kuva 6. Miltä kuvan 4 koodi näyttää käyttäjälle CSS määrytyksien jälkeen

Koko projektin muodostaminen helpottuu, kun saa yhden osan valmiiksi muotoilun osalta koska sitten voi käyttää samoja muotoiluja tulevissa HTML-elementeissä. Kuten kuva 5 näyttää hyvälle käyttäjällä on helppo rakentaa puuttuvat elementit ja muotoilla ne samankaltaisiksi. CSS:n puolella projektissa huomioitiin 10Pro järjestelmän muotoilua ja ulkoasua, jotta uusi ominaisuus näyttäisi olevan osa tätä kokonaisuutta.

Kuva 7. Yhteenvedodokumentti

### 3.3 Yhteenvedodokumentin toiminnallisuus ja dynaamisuus

Yhteenvedodokumentin toiminnallisuus ja dynaamisuus toteutettiin PHP:n ja JavaScriptin avulla. Ei ole reilua verrata PHP:tä JavaScriptiin, koska niillä molemmilla on eri tarkoitukset verkkosivustojen kehittämiseen. PHP on palvelinpuolen ohjelmointia ja JavaScript taas on asiakaspuolen ohjelmointia. Dynaamisin verkkosivusto syntyy yhdistämällä asiakas- ja palvelinpuolen toimintoja. Jos PHP on sivellin kuvan maalaamiseen, JavaScript on sen maaliväri (Jackson, 2021).

JavaScript hoitaa käyttäjän interaktion dokumentin kanssa. Esimerkkeinä kaikki nappuloiden toiminnot, jotka löytyvät kyseistä yhteenvedodokumentista. JavaScriptin toiminnot sidotaan HTML-elementtien ID:hen. Jokaisella HTML-elementillä, jota tarvitaan tallennukseen tai käyttöön täytyy löytyä ID, jotta JavaScript komento tietää mistä elementistä on kyse.



Rakennushankkeeseen ryhtyvät

Kuva 8. Rakennushankkeeseen ryhtyvät

Kyseisessä kuvassa HTML-elementeillä nimi, puhelinnumero ja lisää nappulalla on oma ID, joiden avulla niitä voidaan käsitellä JavaScriptissä. Tässä kohtaa mukaan tulee PHP, joka on mukana Lisää nappulan JavaScriptissä niin sanottuna toimintona.

```

$(document).on('click', '#construction_participantadd', function(e){

    if (!$('#participant_name').val()) {
        message.show('Nimikenttä ei voi olla tyhjä', 'error', 5000);
        return;
    }
    if (!$('#participant_phone').val()) {
        message.show('Anna puhelinnumero', 'error', 5000);
        return;
    }

    var data = {
        'action': 'addparticipant',
        'name': $('#participant_name').val(),
        'phone_nmb': $('#participant_phone').val(),
        'id': selectedSiteId
    };

    $.post('php/project_document.php', data, function(response) {

        $('#participant_name').val('');
        $('#participant_phone').val('');

        var succeed = (response,true);

        if(succeed){
            message.show("Lisäys onnistui!");
            getDocumentinfo();
        }
        else {
            message.show("Virhe kokeile uudelleen!");
        }
    });
});

```

Kuva 9. Rakennushankkeeseen ryhtyvien lisäys JavaScript

Aihetunnisteella eli hashtagilla merkatut ovat juuri Kuva 7 kenttien ID:tä, joita käytetään datan viemisessä PHP:lle. Koko yhteenvedodokumentti rakentuu samalla tavalla. Käyttäjä täyttää kenttiä ja painaa haluamaansa nappulaa, joka kutsuu JavaScriptissä olevaa toimintoa. JavaScript on sidoksissa PHP:hen, joka tekee toiminnon mitä kutsutaan kyseisessä JavaScriptissä. Kuten Kuva 8 ('action': 'addparticipant') on toiminto, joka sitten suoritetaan PHP:n puolella. PHP puoli on taas yhteydessä tietokantaan. PHP:ssä tehdään se toiminto mitä käyttäjä painoi JavaScriptin puolella. Esimerkkinä juuri kuvan seitsemän ja kahdeksan hankkeeseen ryhtyvien lisäys, jossa PHP:ssä lisätään tietokantaan käyttäjän kenttiin syöttämät tiedot.

```

<?php
else if (isset($_POST['action']) && $_POST['action'] === 'addparticipant') {

    $siteId = $_POST['id'];
    $name = $_POST['name'];
    $phone = $_POST['phone_nmb'];
    $created_ts = date("Y-m-d H:i:s");

    $query_insert_participant = "INSERT INTO construction_participants (site_id, name, phone_nmb, created_ts, last_mod_by) VALUES ('$siteId', '$name', '$phone', '$created_ts', '$userId')";
    if ($con->query($query_insert_participant) == FALSE) {

        alert("perskule");
        $json['message'] = 'jotain vikaa';
    } else {
        echo "INSERTTI TOIMII ";
    }

    $query_update_participant = "UPDATE construction_participants SET name = '$name' WHERE phone_nmb='$phone' AND created_ts='$created_ts'";

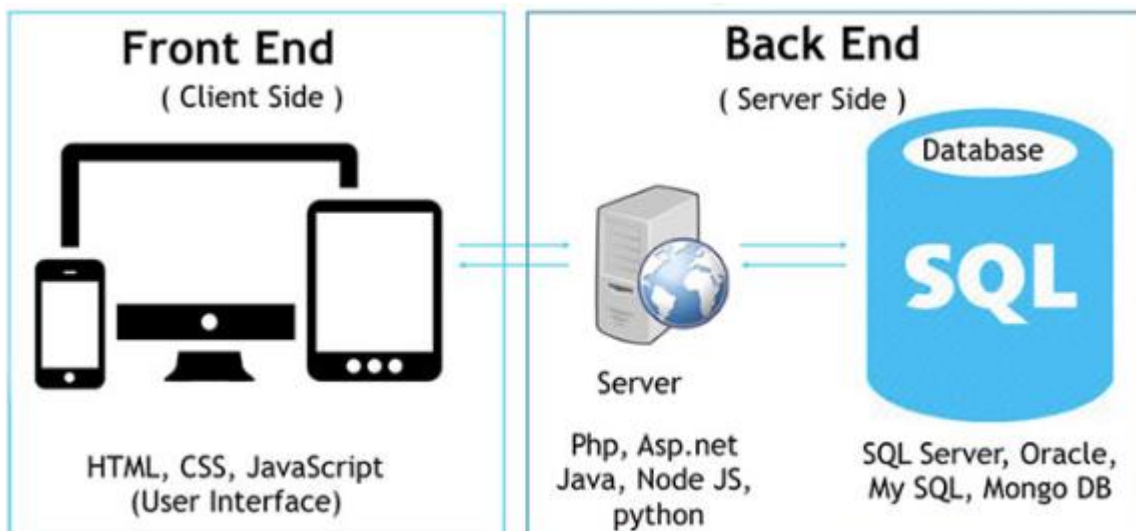
    if ($con->query($query_update_participant) == TRUE) {

        echo "UPDATEKIN TOIMII";
    }else {
        echo "Error: " . $sql . "<br> . $con->error;
        echo "ERRORRRRRR";
    }
}

```

Kuva 10. Rakennushankkeeseen ryhtyvien lisäys PHP

Yhteenvedodokumentin toiminnallisuus ja dynaamisuus rakennettiin käytännössä samalla tavalla kuin esimerkki kuvissa 8–9 mutta tietysti koodin sisältö muuttuu sen mukaan mitä asiakas- ja palvelinpuolella haluttiin tapahtuvan milläkin toiminnolla.



Kuva 11. Front End(asiakaspuoli) Back End(palvelinpuoli)

### 3.4 Yhteenvedodokumentin muita toimintoja

Työnjohtajien lisääminen tapahtuu samalla tavalla kuin aikaisemmin esimerkkinä käytetty rakennushankkeeseen ryhtyvien lisääminen. Isoimpana erona on kuitenkin, että kirjoituskenttien tilalla on valintalistat, jonka alta löytyy tietyn työmaan henkilöt listattuna aakkosjärjestykseen sekä toisesta kentästä roolit.

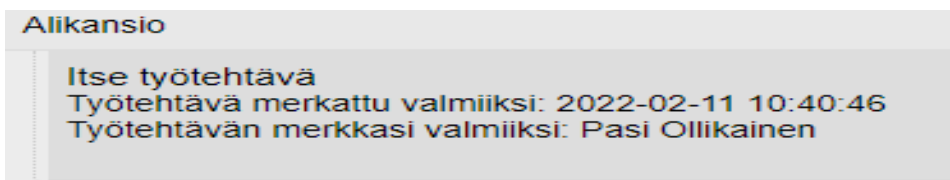
Kuva 12. Työnjohtajaosio

Kuva 13. Valintalista työmaan henkilöstöstä

Työtehtävien kuittaus toteutettiin niin että aluksi luodaan kansio, jonka alle luodaan alikansio ja alikansion alle itse työtehtävä. Tämä helpottaa käyttäjää seuraamaan tehtäviä, kun tehtävät ovat tiettyjen kansioiden alla.

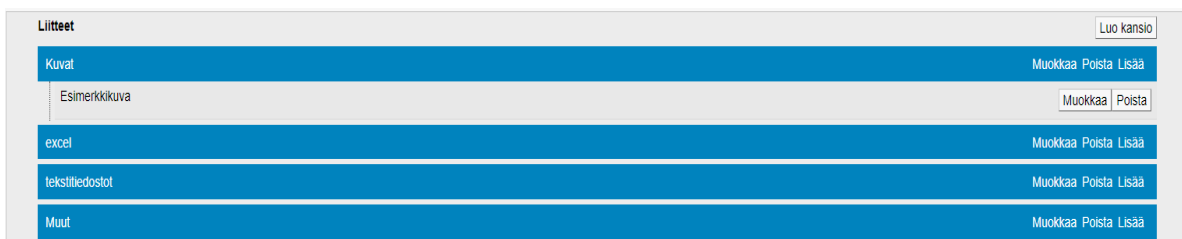
Kuva 14. Työlistan toiminta

Kun luo työtehtävän se näyttää, että sitä ei ole merkattu valmiiksi. Kun työtehtävän merkkää valmiiksi se ottaa ylös käyttäjän, joka merkkasi tehtävän valmiiksi ja ajan siltä hetkeltä.



Kuva 15. Työtehtävän merkitseminen valmiiksi

Liitteiden lisääminen on tehty toteutettu samankaltaisesti kuin työlistat. Erona että on vain pääkansio ja sen alle pystyy lisäämään haluamansa liitteen. Pääkansioina esimerkiksi kuvat, excel-tiedostot, tekstitiedostot ja muut. Näin kaikki työmaan tarvitsevat liitteet ovat yhdessä paikkaa.



Kuva 16. Liitteet osio

#### 4 PROJEKTIN TULOS

Kaiken kaikkiaan projekti onnistui hyvin ja kaikki halutut ominaisuudet ja toiminallisuudet tavoitettiin. Visuaalisesti opinnäytetyö onnistui myös hyvin ja se näyttää olevan juuri osa 10Pro järjestelmää. Kaikki mitä projektisuunnitelmassa tuli esille saatiin toteutettua. Tietysti erilaisia kompromisseja tehtiin työn ohessa, jotta toiminnallisuudesta saatiin mahdollisimman helppokäyttöinen rakennustyömaan vastuuhenkilöille. Vaikka kaikki halutut asiat saatiin tehtyä, on projektissa mahdollisuus jatkokehitykselle, joista enemmän jatkokehitysideoissa.

Visuaalinen puoli näyttää hyvälle käyttäjälle aivan niin kuin haluttiin. 10Pro järjestelmässä oli tietynlainen värimaailma ja erilaisten ominaisuuksien haluttiin olevan samankaltaisia kuin muissa järjestelmän ominaisuuksissa. Tässä onnistuttiin hyvin kuten myös toiminnallisuuden puolella. Vastuuhenkilöt pystyvät täyttämään rakennustyömaan tiedot omiin kenttiinsä ja muokkaamaan niitä. Hankkeeseen ryhtyviä voidaan lisätä sekä juuri kyseisen työmaan johtajien listaus ja roolitus toimii myös hyvin. Lisäksi työtehtävien listaaminen ja kuittaus onnistui. Viimeinen haluttu ominaisuus projektiin eli yhteenvedodokumenttiin oli liitteiden lataus, joka saatiin myös toimimaan.

## 5 YHTEENVETO JA POHDINTA

Kokonaisuutena opinnäytetyö onnistui hyvin ja antoi paljon oppeja ohjelmistokehittäjän työstä. Koko projekti meni oikeastaan niin kuin aikataulussa oli suunniteltu. Myöskin alustava suunnitelma ja mitä aloituspalaverissa tilaajan kanssa keskusteltiin, meni oikein mallikkaasti. Eli kokonaisuutena kaikki meni hyvin mutta isoimpina ongelmina oli mielestäni tutustuminen 10Pro järjestelmään ja siihen minkälaisia tekniikoita ja ratkaisuja he ovat käyttäneet tehdessään järjestelmää. Ylipäätään oma tietämys web-ohjelmoinnista ei ollut hyvinkään laaja, joten sen opettelu ja tutkiminen vei odotettua pitempään. Mitä pidemmälle opinnäytetyö eteni sitä nopeammin, alkoi ratkaisuja ongelmiin löytymään.

Koko opinnäytetyö tarjosi valtavan määrän oppia niin itse ohjelmoinnista kuin myös isomman projektin tekemisestä suunnittelusta lopputuotteeseen. Sai aivan uudenlaisen kuvan siitä mitä kaikkea projekti vaatii ja miten paljon asioita täytyy ottaa huomioon. Opini myös hahmottamaan paremmin asiakaspuolen ja palvelinpuolen koodia. Opini myös miten iso merkitys sosiaalisilla taidoilla on tällä alalla. Esimerkkinä tästä, kun sain osan työstä tehtyä, katsoimme sen läpi tilaajan kanssa ja pohdimme yhdessä, onko ratkaisu toimiva.

Kokonaisuudessaan opinnäytetyö tarjoaa loistavat lähtökohdat työelämään, koska se vastasi käytännössä täysipäiväistä työtä. Se tarjosi myös tekijälle hienon kokemuksen ja tietopaketin web-ohjelmoinnista ja kuinka yrityksessä toimitaan. Valmistumisen jälkeen tästä on erittäin hyvä lähteä jatkamaan kohti tulevaa työelämää. Opinnäytetyöstä myös iso apu työnhakuun. Mikään tutkinto ei yksinään riitä ohjelmointitöitä haikvalle. Käytännön osaamisesta tulee olla näyttöjä (Sullström, 2020).

## 6 JATKOKEHITYS

Selkeitä jatkokehitysideoita ei ollut koska kaikki halutut toiminnot saatiin toteutettua. Tässä kuitenkin hyvä ottaa huomioon, että jatkokehitysideoita saattaa tulla paljonkin rakennustyömailta, kun uusi ominaisuus otetaan käyttöön. Tilaajalta tuli selkeä listaus mitä projektilta halutaan ja siihen tavoitteeseen päästiin. Pitää muistaa, että rakennustyömaat muuttuvat ja saattavat haluta paljon uusia ja erilaisia ominaisuuksia yhteenvetodokumenttiin. Tässä isona plussana versionhallinta, jonka avulla pystytään nämä uudet ominaisuudet rakentamaan niin että vanha yhteenvetodokumentti on toiminnassa siihen asti, kunnes uudet jatkokehitysideat on toteutettu ja testattu. Yhteenvetodokumentti on myös rakennettu niin että sinne on helppo lisätä uusia ominaisuuksia, jos ja kun niille tulee tarvetta työmaille.

## 7 LÄHDELUETTELO

Commons, C. (2019). *kirjat.it*. Haettu 3. 2. 2022 osoitteesta [https://kirjat.it.jyu.fi/linkki/nettisivut\\_ohjelmointi/css.html](https://kirjat.it.jyu.fi/linkki/nettisivut_ohjelmointi/css.html)

Education-wiki. (2022). *Education-wiki*. Haettu 2. 2. 2022 osoitteesta <https://education-wiki.com/5876083-what-is-mysql>

Jackson, P. (25. 12. 2021). *Guru99*. Haettu 3. 2. 2022 osoitteesta <https://www.guru99.com/php-vs-javascript.html>

Laaksonen, A. (2009). *Ohjelmointiputka*. Haettu 3. 2. 2022 osoitteesta <https://www.ohjelmointiputka.net/opaat/opas.php?tunnus=mysqlphpl1>

Sullström, H. (18. 2. 2020). *Yle.fi*. Haettu 10. 2. 2022 osoitteesta <https://yle.fi/uutiset/3-11215988>

Suoranta, T. (5. 3. 2020). *Koodarikuiskaaja*. Haettu 10. 2. 2022 osoitteesta <https://koodarikuiskaaja.fi/blogi/vieraskyna-ohjelmointi-ei-voi-olla-elaman-ainut-sisalto/>