



# Moodle-lisäosa opiskelijoiden aktiivisuuden seurantaan

Samuli Leikkola

OPINNÄYTETYÖ  
Kesäkuu 2020

Tieto- ja viestintäteknikan insinöörikoulutus  
Ohjelmistotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tieto- ja viestintätekniikan insinöörikoulutus  
Ohjelmistotekniikka

LEIKKOLA SAMULI:

Moodle-lisäosa opiskelijoiden aktiivisuuden seurantaan

Opinnäytetyö 23 sivua

Kesäkuu 2020

---

Opinnäytetyön tarkoituksena on tuottaa Tampereen Ammattikorkeakoululle Moodle-lisäosa, jonka avulla voidaan seurata opiskelijoiden aktiivisuutta. Lisäosan tarkoituksena on tuottaa TAMK:n henkilökunnalle visuaalinen työkalu, jolla voidaan tarkastella kunkin aloittavan opiskelijaryhmän opiskelijoiden aktiivisuutta Moodle alustan kautta. Lisäosa tutkii opiskelijoiden avaamien ja palauttamien tiedostojen määrää ja laskee näistä jokaiselle opiskelijalle viikkokohtaisen aktiivisuuden. Viikoittaisia aktiivisuuksia tarkastellaan kolmen viikon liukuvan keskiarvon avulla. Lasketut liukuvat keskiarvot visualisoidaan kuvaajaan, josta niitä on helppoa tarkastella. Visuaalinen esitys tarjoaa mahdollisuuden tarkastella opiskelijoiden aktiivisuuden muutoksia pitkällä aikavälillä. Kuvaajasta aktiivisuudessa tapahtuvat muutokset on helpompi huomata kuin listasta, jossa aktiivisuudet olisi esitetty numeerisina arvoina.

Lisäosan toinen tavoiteltu ominaisuus on automaattinen hälytysjärjestelmä, joka ilmoittaa henkilökunnalle oppilaista, joiden aktiivisuus on liian alhainen. Tällaiset opiskelijat ovat vaarassa keskeyttää opintonsa, ja heihin halutaan pystyä olemaan yhteydessä mahdollisimman tehokkaasti. Jos oppilaiden aktiivisuuden mahdolliset muutokset pystytään havaitsemaan nopeasti, pystytään niihin tällöin myös reagoimaan nopeasti. Oppilaille voidaan henkilökunnan toimesta tarjota tarvittaessa apua opintojen jatkon suunnittelun suhteen.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
ICT Engineering  
Software Engineering

SAMULI LEIKKOLA  
Moodle plugin for tracking student activity

Bachelor's thesis 23 pages  
June 2020

---

The aim of this thesis is to produce a Moodle plugin for tracking student activity. This plugin was made for Tampere University of Applied Sciences. The goal of the plugin is to create an easy visual tool for tracking the activity of each new student group through Moodle platform. The plugin tracks which documents students open and how many files they submit to Moodle. The plugin then uses these statistics and calculates each student's activity. Activity is examined using a moving average of three weeks. These moving averages are then visualized in a graph. When the activity data is visualized in a graph it is easy to analyze. Visual representation gives the opportunity to analyze the changes in student activities through a long period of time. It is easier to spot changes in activity when it is visualized in graphs rather than displayed in lists of numerical values.

Second main function of the plugin is an automated warning system that will inform the TAMK staff if a student has been very inactive in his or her studies. These kinds of students are at risk of dropping out. TAMK staff wants to have the possibility to contact these students as proficiently as possible. Best way to reach this is to have the ability to spot the changes in student activity as soon as possible. If the staff gets notified about a student that has been very inactive in their studies, they can offer them help in planning their studies and provide support.

---

Key words: Moodle, plugin, activity

## SISÄLLYS

1	JOHDANTO .....	6
2	TEORIA .....	7
	2.1 Moodle .....	7
	2.1.1 Moodle lisäosat .....	7
	2.2 Käytetyt kehitystyökalut .....	8
	2.2.1 Microsoft Visual Studio Code .....	8
	2.2.2 Notepad++ .....	8
	2.3 Käytetyt ohjelmointikielät .....	9
	2.3.1 PHP .....	9
	2.3.2 HTML .....	9
	2.3.3 SQL .....	9
3	SUUNNITTELU .....	10
	3.1 Toimintaympäristö .....	10
	3.2 Lisäosa ja sen tyyppi .....	10
	3.3 Lisäosan toiminta .....	10
	3.4 Tietoturva .....	11
4	TOTEUTUS .....	13
	4.1 Aloitus .....	13
	4.2 Kehitysympäristöjen asennus .....	13
	4.3 Moodleen ja sen ominaisuuksiin tutustuminen .....	14
	4.4 Moodlen valmiisiin lisäosiin tutustuminen .....	14
	4.5 Oman lisäosan tuottaminen .....	14
	4.5.1 Pohjatyö .....	15
	4.5.2 Työnjako .....	15
	4.5.3 Graafisten kuvaajien ohjelmointi .....	15
	4.5.4 Kuvaajien tuottaminen .....	16
5	POHDINTA .....	19
	5.1 Työn eteneminen .....	19
	5.2 Työn tavoitteiden täytyminen .....	19
	5.3 Mahdollinen jatkokehitys .....	20
	5.4 Yhteistyön toimiminen ja parityön hyödyt .....	21
	5.5 Yhteenveto .....	22
	LÄHTEET .....	23

**LYHENTEET JA TERMIT**

TAMK	Tampereen ammattikorkeakoulu
Moodle	Oppilaitosten käyttämä verkossa toimiva oppimisympäristö
Tabula	Tampereen ammattikorkeakoulun käyttämä nimitys Moodlesta
PHP	Ohjelmointikieli, jota käytetään yleisesti verkkosivujen rakentamiseen. PHP on lyhenne sanoista Hypertext Preprocessor
HTML	Ohjelmointikieli, jota käytetään enimmäkseen verkkosivujen rakentamiseen. HTML on lyhenne sanoista Hyper Text Markup Language
SQL	Ohjelmointikieli, jota käytetään pääasiassa tietokantojen käsittelyyn. SQL on lyhenne sanoista Structured Query Language
Notepad++	Kehitysympäristö, jota käytettiin työn ohjelmointiosuuk- sissa
Microsoft Visual Code	Kehitysympäristö, jota käytettiin työn ohjelmointiosuuk- sissa.

## 1 JOHDANTO

Opinnäytetyön tarkoituksena oli tuottaa Tampereen ammattikorkeakoululle Moodle-lisäosa, jonka avulla koulun henkilökunta voi seurata opiskelijoiden aktiivisuutta. Lisäosan tarkoituksena on helpottaa koulun henkilökunnan toimintaa opiskelijoiden aktiivisuuden seurannassa. Aikaisemmin aktiivisuutta seurattiin Microsoftin Excel-taulukkolaskentasovelluksella. Kaikki tutkittava data jouduttiin lisäämään Exceliin manuaalisesti ja tämä aiheutti paljon ylimääräistä työtä. Opinnäytetyössä tuotetulla lisäosalla pyritään automatisoimaan aktiivisuuden seurantaprosessia ja näin ollen henkilökunnan työtaakkaa saadaan kevennettyä.

Opinnäytetyö on toteutettu yhteistyössä toisen opiskelijan kanssa. Yhteistyöhön päädyttiin molempien opiskelijoiden kiinnostuttua samasta aiheesta. Aleksi Majaran osuus työstä keskittyi työn tietokantaosuuteen ja tässä työssä keskityttiin opiskelijoiden aktiivisuuden kuvantamiseen.

## 2 TEORIA

Tämä kappale käsittelee tämän opinnäytetyön toteuttamiseen vaadittujen ympäristöjen ja teknologioiden teoriapuolta. Tässä kappaleessa kerrotaan myös mitä kehitysympäristöjä ja ohjelmointikieliä tässä työssä on käytetty.

### 2.1 Moodle

Moodle on avoimen lähdekoodin virtuaalinen oppimisolusta, joka tarjoaa oppilaitoksille ja opettajille mahdollisuuden luoda ja muokata kursseja ja aktiviteetteja joustavilla työkaluilla verkossa tapahtuvaa opiskelua varten (Moodle, 2020).

Moodlea käytetään Suomessa laajasti sekä korkeakouluissa että toisen asteen oppilaitoksissa.

Moodleen voidaan lisätä lisäosia, joiden avulla alusta on helposti muokattavissa yksittäisen oppilaitoksen, koulutusohjelman tai jopa kurssin tarpeisiin. Moodleen perusasennus tarjoaa kaikki tarvittavat työkalut kurssien luomiseen ja muokkaamiseen. Kurseille voidaan lisätä tiedostoja ja tehtäviä, joita opiskelijat voivat tarkastella, suorittaa ja palauttaa. Lisäosia asentamalla Moodleen voidaan lisätä monia ominaisuuksia, kuten tässä työssä toteutetun kaltaisia aktiivisuuden analysointiin tarkoitettuja työkaluja.

Tampereen Ammattikorkeakoulu käyttää Moodlesta versiota 3.5 ja alusta toimii nimellä Tabula. Työssä tuotettu lisäosa suunniteltiin toimimaan ensisijaisesti TAMK:n käyttämän Moodle-version kanssa. Työssä pyrittiin myös huomioimaan lisäosan toimivuus mahdollisen Moodleen versiopäivityksen jälkeen.

#### 2.1.1 Moodle lisäosat

Helpoin ja parhaiten hallittavissa oleva tapa lisätä toiminnallisuutta Moodleen ovat lisäosat (Moodle, 2020). Moodleen on saatavilla valtava määrä erilaisia ja

erityyppisiä lisäosia, kuten aktiviteetteja, lohkoja ja teemoja. Lisäosien avulla Moodleen voidaan lisätä toimintoja, joita perusasennukseen ei kuulu.

## **2.2 Käytetyt kehitystyökalut**

Moodlen lisäosia tuotetaan suurimmaksi osaksi PHP-ohjelmointikielellä. Tämän kaltaisten ohjelmien muokkaamiseen ja luomiseen on useita hyviä työkaluja, joista tämän projektin toteuttamiseen valittiin Microsoft Visual Studio Code ja Notepad++. Työtä tehdessä havaittiin, että näistä työkaluista Notepad++ oli huomattavasti tehokkaampi tässä käyttötarkoituksessa ja näin ollen Visual Studio Code hylättiin kehitystyökaluna lopulta kokonaan.

### **2.2.1 Microsoft Visual Studio Code**

Microsoft Visual Studio Code on Microsoftin tuottama kevyt mutta tehokas lähdekoodin editointiin tarkoitettu ohjelma, joka tukee useiden eri ohjelmointikielten kirjoittamista ja muokkaamista (Microsoft, 2020). Visual Studio Codella voidaan kirjoittaa kaikkia tässä opinnäytetyössä tarvittavia ohjelmointikieliä. Visual Studio Codea käytettiin tässä työssä Moodlen, ja sen lisäosien ohjelmakoodin tarkasteluun ja muokkaamiseen.

### **2.2.2 Notepad++**

Notepad++ on ilmainen lähdekoodin editointiohjelma, joka tukee useiden eri ohjelmointikielten kirjoittamista ja muokkaamista (What is Notepad++, 2020). Notepad++:a käytettiin työssä ohjelmointiosuuskien suorittamiseen. Notepad++ havaittiin työssä äärimmäisen käyttökelpoiseksi työkaluksi sen monipuolisuuden ja loistavan käytettävyyden vuoksi. Ohjelmalla pystyttiin kirjoittamaan kaikkia työssä tarvittavia ohjelmointikieliä. Notepad++ tukee sekä HTML, PHP, että SQL-ohjelmointikieliä.

Ohjelman käyttöä tuki myös sen käytettävyys. Moodlen paikalliselle palvelimelle asennettuja tiedostoja voidaan avata suoraan Notepadilla ja niitä voidaan muokata ja tallentaa erittäin nopeasti ja kätevästi.

## **2.3 Käytetyt ohjelmointikielet**

Tässä opinnäytetyössä tuotetun Moodle-lisäosan rakentamiseen tarvittiin useita eri ohjelmointikieliä. Eri ohjelmointikielillä rakennettiin lisäosan eri osia. Erilaisia ohjelmointikieliä käytetään esimerkiksi tietokantojen käsittelyyn ja tietojen muokkaamiseen ja visualisointiin.

### **2.3.1 PHP**

PHP on suosittu ja yleisesti käytetty ohjelmointikieli, joka sopii erityisesti webkehitykseen (The PHP Group, 2020). PHP mahdollistaa monien toimintojen ja sovellusten luomisen verkkosivuille. PHP-koodilla voidaan käsitellä ja muokata tietoja sekä luoda algoritmeja, joilla käsitellyt tiedot saadaan tulostettavaan muotoon.

### **2.3.2 HTML**

HTML on ohjelmointikieli, jolla suurin osa internetsivuista luodaan (Frank Moraes, 2020). HTML-kieli on käytössä varsinkin sivustojen visuaalisten elementtien tulostamisessa. HTML-koodilla voidaan tulostaa sivustoille esimerkiksi tekstiä ja kuvia.

### **2.3.3 SQL**

SQL on ohjelmointikieli, jolla voidaan ottaa yhteyttä tietokantoihin ja muokata niitä (w3schools, 2020). SQL-kyselyitä käytettiin Moodlen keräämien lokitietojen hakemiseen tietokannasta. SQL-kyselyillä saatuja tietoja voidaan muokata esimerkiksi PHP-koodilla ja tiedot voidaan palauttaa takaisin tietokantaan SQL-koodilla.

## **3 SUUNNITTELU**

### **3.1 Toimintaympäristö**

Tässä opinnäytetyössä toteutettu lisäosa tehtiin toimimaan Moodlen versiossa 3.5. Tämä on versio, jota TAMK tällä hetkellä käyttää ja siksi se valittiin tuotettavan lisäosan toimintaympäristöksi. Työn toimintaympäristöksi päätettiin asentaa Moodlen versio 3.5 toimimaan paikallisesti omalla tietokoneella pyörivälle palvelimelle. Moodle asennettiin paikallisesti, koska lisäosan kehitystä ja testausta ei voitu suorittaa TAMK:n omalla Moodle-alustalla toiminnallisista ja tietoturvaan liittyvistä syistä.

### **3.2 Lisäosa ja sen tyyppi**

Työssä tuotettavan lisäosan tyyppiä päädyttiin valitsemaan lohko. Lohkot ovat pieniä informaationäyttöjä tai työkaluja, joita voidaan liikuttaa ympäriinsä sivuilla (Moodle, 2020). Lohkoa päädyttiin käyttämään lisäosan tyyppinä, sillä lohkoissa on mahdollista käyttää HTML-koodia. HTML-koodin käyttö on tässä lisäosassa erittäin tärkeää, sillä se on paras mahdollinen tapa luoda tässä työssä tarvittavia kuvaajia.

### **3.3 Lisäosan toiminta**

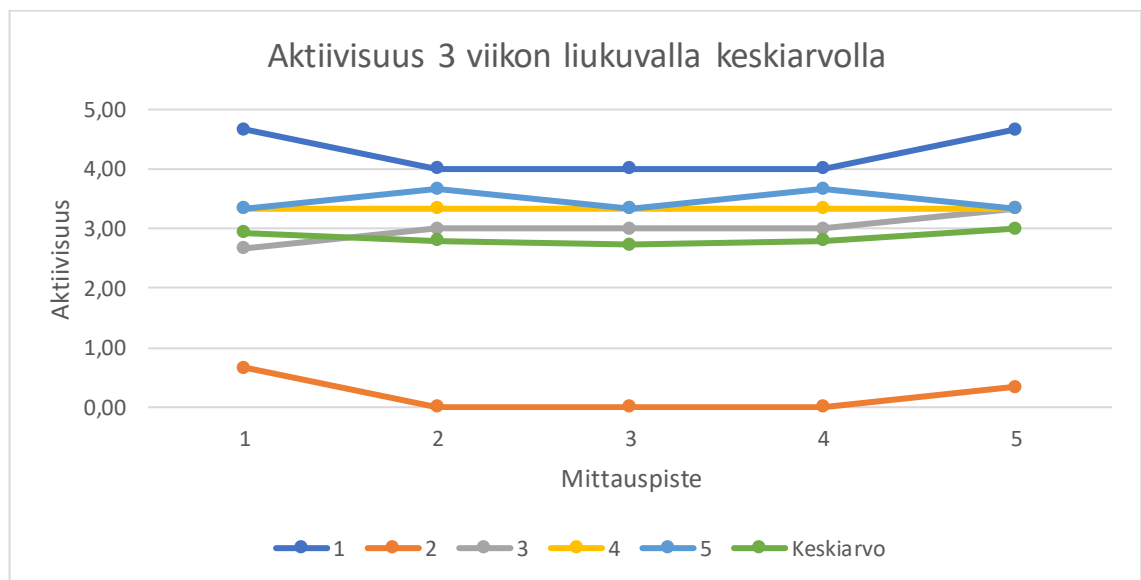
Lisäosan toiminta perustuu Moodleen keräämiin lokitietoihin. Moodle kerää lokitiedostoihin opiskelijoiden ja muiden käyttäjien tekemät toiminnot, kuten esimerkiksi materiaalien avaukset ja tehtävien palautukset.

Lokitietojen perusteella lisäosa kykenee tarkastelemaan yksittäisen oppilaan aktiivisuutta Moodle-alustalla. Lisäosa seuraa oppilaan lokimerkintöjen määrää ja tutkii opiskelijan aktiivisuutta kolmen viikon liukuvan keskiarvon avulla. Kolmen viikon liukuva keskiarvo tarkoittaa sitä, että opiskelijan aktiivisuutta tarkastellaan kolmen viikon jaksoissa. Näiden kolmen viikon aktiivisuuksista lasketaan keskiarvo. Lisäosa antaa varoituksen, mikäli tämä keskiarvo on nollassa kolmessa

perättäisessä pisteessä. Jotta keskiarvo olisi nolla kolmessa perättäisessä pisteessä tulee opiskelijan aktiivisuuden olla nollassa viitenä viikkona peräkkäin.

Lisäosa piirtää keräämiensä tietojen perusteella kuvaajan, joka esittää opiskelijan aktiivisuuden visuaalisesti. Kuvaajasta on helppo tarkastella opiskelijan aktiivisuutta pitkältä ajalta ja nähdä, missä kohdassa opiskelijan aktiivisuudessa on tapahtunut muutoksia. Muutokset on helpompi havaita graafisesta kuvaajasta, kuin esimerkiksi listasta, jossa aktiivisuus esitettäisiin numeroina.

Alla olevassa kuviossa (Kuvio 1) on esimerkki siitä, miltä lisäosan kuvaajien olisi tarkoitus näyttää valmiissa tuotteessa. Kuviossa esitetään viiden eri opiskelijan aktiivisuutta tietyn ajanjakson ajalta. Kuvioista on nähtävissä, että opiskelijan numero kaksi aktiivisuus on ollut koko tarkasteltavan jakson ajan erittäin matalaa, tai jopa olematonta. Kuvioista voidaan nähdä, että opiskelijan numero kaksi aktiivisuus on ollut nollassa kolmessa perättäisessä mittauskohdassa, ja näin ollen hänen tilanteestaan tulisi lähettää ilmoitus koulun henkilökunnalle.



Kuvio 1 Viiden eri käyttäjän aktiivisuus samassa kuvaajassa

### 3.4 Tietoturva

Tietoturva ja palveluiden turvallisuus ovat erittäin tärkeitä asioita nykypäivän ohjelmistokehityksessä. Tässä työssä on pyritty huomioimaan näitä asioita mahdollisimman tehokkaasti.

Moodlen toiminta on jo asennettaessa melko turvallista ja opiskelijoista on mahdollista saada vain hyvin rajatusti tietoa Moodlen lokitietojen kautta. Vaikka lokitietoihin tallentuu kaikki opiskelijan toiminta alustalla, ei opiskelijasta kuitenkaan jää lokeihin mitään arkaluonteista tietoa. Esimerkiksi opiskelijan opiskelijanumero tai syntymäpäivä eivät ole saatavissa Moodlen lokitiedoista tai tietokannoista. Opiskelijanumeron sijaan Moodlen sisällä opiskelijalle annetaan automaattisesti oma id eli numeraalinen tunnus, josta hänen toimintansa voidaan tunnistaa.

## **4 TOTEUTUS**

### **4.1 Aloitus**

Työn valmistelu aloitettiin pitämällä palaveri työn tilaajan eli TAMK:n edustajan kanssa. Palaverissa käsiteltiin tilatun tuotteen ominaisuuksia ja toteutustapaa. Palaveriin osallistui työn ohjaajan ja työn tekijöiden lisäksi myös muita TAMK:n työntekijöitä.

Palaverissa sovittiin työn aikatauluista ja ohjauksesta. Työn ohjaukseen osallistui aktiivisesti opinnäytetyön ohjaajan lisäksi myös TAMK:n sisällä toimivan Smart Campus Innovation Labin työntekijä Jussi Hannunen. Hannusen kanssa pidettiin työn alkuvaiheessa lähes viikoittain palavereita, joissa seurattiin työn edistymistä ja keskusteltiin työn tavoitteista ja toteutuksesta.

### **4.2 Kehitysympäristöjen asennus**

Moodle voidaan asentaa toimimaan paikallisesti omalla tietokoneella pyörivälle palvelimelle. Tällaista paikallista asennusta on hyvä käyttää Moodlen ja sen lisäosien kehittämiseen ja testaamiseen ennen kuin tehtyjä muutoksia asennetaan toimivaan Moodle ympäristöön. Toimivalla ympäristöllä tarkoitetaan jonkin oppilaitoksen käytössä olevaa julkisesti verkossa toimivaa Moodlea, jonka tietokannoissa on oikeaa opiskelijoiden dataa.

Työtä varten asennettiin TAMK:n käyttämä Moodle-versio. Moodle on avoimen lähdekoodin ohjelma, joten sen lataaminen ja asentaminen on helppoa ja ilmaista. Moodle asennettiin toimimaan paikallisesti omalla työasemalla pyörivälle palvelimelle työn tekemistä ja testaamista varten.

### **4.3 Moodleen ja sen ominaisuuksiin tutustuminen**

Kun haluttu Moodlen versio saatiin asennettua, alettiin tutustua sen ominaisuuksiin ja rakenteeseen. Kummallakaan työn tekijöistä ei ollut aikaisempaa kokemusta Moodlen lisäosien kehittämisestä, joten suurin osa työssä tarvittavista taidoista jouduttiin hankkimaan työn tekemisen ohessa. Nopeasti huomattiinkin, että Moodle ja sen tietorakenne ovat melko monimutkaisia ja niiden selvittämiseen tulisi kulumaan huomattavasti aikaa. Moodle tarjoaa sivuillaan hyvin laajan ja kattavan dokumentaation, jota tutkimalla on mahdollista selvittää suurin osa vastaan tulevista kysymyksistä.

### **4.4 Moodlen valmiisiin lisäosiin tutustuminen**

Moodleen on vapaasti ladattavissa valtava määrä valmiita lisäosia, joiden kehittäminen perustuu avoimeen lähdekoodiin ja työn alkuvaiheessa keskusteltiinkin paljon siitä, tulisiko työn pohjana käyttää jo olemassa olevia lisäosia. Lopulta päädyttiin ottamaan työn pohjaksi Analytics Graph -niminen lisäosa, jota lähdettiin tutkimaan aluksi Microsoft Visual Code -ohjelmalla. Valittua lisäosaa tutkittaessa huomattiin, että se sopii erinomaisesti työn pohjaksi. Analytics Graph on lisäosa, joka tuottaa Moodlen keräämistä lokitiedoista erilaisia taulukoita ja kuvia.

Analytics Graph -lisäosa asennettiin Moodlen paikalliselle palvelimelle ja sitä alettiin testata itse tuotetulla testidatalla. Testidataa tuotettiin luomalla Moodleen testikursseja. Näille kursseille lisättiin resursseja, joita avaamalla ja tarkastelemalla saatiin tuotettua aktiivisuusdataa.

### **4.5 Oman lisäosan tuottaminen**

Tässä osiossa kuvataan Moodle-lisäosan rakentamista. Prosessi käydään läpi pohjatöistä valmiiseen lisäosaan asti.

### 4.5.1 Pohjatyö

Oman lisäosan pohjana käytettiin jo aikaisemmin tässä tekstissä mainittua Analytics Graph -lisäosaa, jota lähdettiin muokkaamaan työlle annettujen vaatimusten mukaiseksi. Lisäosan ohjelmointi aloitettiin Microsoftin Visual Code ohjelmalla. Työn edetessä havaittiin, että Visual Code ei tarjonnut kaikkia työssä vaadittavia toimintoja ja uudeksi kehitystyökaluksi päädyttiin valitsemaan Notepad++. Notepad++ tarjosi huomattavasti joustavamman ja monipuolisemman työkalun, jolla pystyttiin muokkaamaan ja tutkimaan Moodleen asennettuja tiedostoja nopeasti ja tehokkaasti. Notepad++ kykenee tallentamaan tiedostoihin tehdyt muutokset suoraan Moodlen palvelimelle ja näin säästyy huomattavan paljon aikaa. Ilman Notepad++:a tehdyt muutokset olisi jouduttu asentamaan manuaalisesti Moodlen tietokantaan ja tämä olisi aiheuttanut valtavan määrän turhaa työtä ja hidastanut huomattavasti työskentelyä.

### 4.5.2 Työnjako

Kuten johdannossa mainittiin, tällä opinnäytetyöllä oli kaksi tekijää. Työn tietokantaa käsittelevä osuus kuului Aleksi Majaralle ja tietokantaosuudesta kerrotaan tarkemmin hänen opinnäytetyössään. Tässä työssä keskitytään opiskelijoiden aktiivisuusdatan visuaaliseen esittämiseen kuvaajien avulla.

### 4.5.3 Graafisten kuvaajien ohjelmointi

Yksi lisäosan tärkeimmistä ominaisuuksista oli opiskelijoiden aktiivisuuden tarkasteleminen graafisessa muodossa. Tämä tarkoittaa käytännössä kuvaajia, joista voidaan nähdä yksittäisten opiskelijoiden aktiivisuus pidemmällä aikavälillä. Moodlen lisäosien ohjelmoinnissa käytetään pääasiassa HTML, PHP ja SQL-ohjelmointikieliä.

HTML-kieltä käytetään lisäosan frontendin ohjelmoimiseen. Frontendillä tarkoitetaan lisäosan käyttäjälle näkyvää osaa eli käyttöliittymää. HTML-koodilla tulostetaan lisäosan loppukäyttäjälle näkyvät kuvaajat.

PHP-kieltä käytetään lisäosan backendin ohjelmoimiseen. Backendillä tarkoitetaan lisäosan sisäistä toimintaa, kuten algoritmeja ja varsinaista toiminnallisuutta. SQL-kieltä käytetään tietokantaan kohdistuvien kyselyjen tekemiseen ja tietojen hakemiseen tietokannasta.

#### 4.5.4 Kuvaajien tuottaminen

Kuvaajien tuottaminen on monivaiheinen prosessi, jossa tarvitaan useita eri ohjelmointikieliä. Prosessi sisältää tietokannan käsittelyä, datan muokkaamista ja visualisointia.

Moodle tallentaa kaiken käyttäjien suorittaman toiminnan tietokannasta löytyviin lokitiedostoihin. Kuvaajien rakentaminen aloitetaan opiskelijoiden aktiivisuutta kuvaavan datan hakemisella tietokannasta. Haku suoritetaan SQL-kyselyillä. SQL-kyselyt ovat SQL-ohjelmointikielellä kirjoitettuja funktioita, jotka hakevat haluttua dataa tietokannasta ja palauttavat saadun datan kyselyn tehneen ohjelman käyttöön. Kuvassa 1 esitellään SQL-kysely, jolla Moodlen tietokannasta haetaan opiskelijan aktiivisuus tietoja. Kuvassa näkyvällä koodirivillä, jolta löytyy komento SELECT, haetaan haluttuja tietoja, kuten opiskelijan tunnusta ja suoritetun aktiviteetin nimeä ja ajankohtaa. Koodissa näkyvä mdl\_logstore\_standard\_log on Moodlen tietokanta, josta tietoja haetaan.

```
$query_id = $_GET['id'];
$sql = "SELECT id, eventname, userid, contextlevel, timecreated,
        FROM_UNIXTIME( timecreated, '%d-%m-%Y' ) AS timecreated
        FROM mdl_logstore_standard_log
        WHERE contextlevel = 70 AND ( userid LIKE '%" . $query_id . "%' )
        ORDER BY id DESC
        LIMIT 10 ";
$result = $conn->query($sql);
```

*Kuva 1 SQL-kysely*

Kyselyjen tuloksena saatua dataa käsitellään PHP-kielellä kirjoitetuilla komennoilla. Lokitiedoista saatua dataa muokataan ja siitä haetaan tarvittavat tiedot. Haetusta datasta löydetään esimerkiksi opiskelijan suorittamat aktiviteetit ja niiden ajankohdat. Nämä ovat tietoja, joita käytetään opiskelijan aktiivisuutta esittävien kuvaajien tuottamisessa. Kuvan 2 koodissa tarkastellaan Moodlessa suoritettavia aktiviteetteja. Koodissa tutkitaan, onko joku opiskelija suorittanut kyseistä

aktiviteettia. Jos joku opiskelija on suorittanut haetun aktiviteetin, ohjelma hakee aktiviteetin suorittaneen opiskelijan tiedot kuvassa 3 näkyvällä ohjelmakoodilla.

```
foreach ($result as $tuple) {
    if ($resourceid == 0) { /* First time in loop -> get topic and content name */
        $numberofresourcesintopic[$tuple->section] = 1;
        $statistics[$counter]['topico'] = $tuple->section;
        $statistics[$counter]['tipo'] = $tuple->tipo;
        if ($tuple->tipo == 'activequiz') {
            $statistics[$counter]['material'] = $tuple->activequiz;
        } else if ($tuple->tipo == 'assign') {
            $statistics[$counter]['material'] = $tuple->assign;
        } else if ($tuple->tipo == 'attendance') {
            $statistics[$counter]['material'] = $tuple->attendance;
        } else if ($tuple->tipo == 'bigbluebuttonbn') {
            $statistics[$counter]['material'] = $tuple->bigbluebuttonbn;
        } else if ($tuple->tipo == 'booking') {
            $statistics[$counter]['material'] = $tuple->booking;
        }
    }
}
```

Kuva 2 Moodle aktiviteettien tarkastelu

```
if ($tuple->userid) { /* If a user accessed -> get name */
    $statistics[$counter]['studentswithaccess'][] = array('userid' => $tuple->userid,
        'nome' => $tuple->firstname." ".$tuple->lastname, 'email' => $tuple->email);
    $numberofaccesses++;
}
```

Kuva 3 Aktiviteetteja suorittaneiden oppilaiden tarkastelu

Kuvassa 4 esitetyssä ohjelmakoodissa tutkitaan opiskelijan aktiivisuutta. Kuvassa näkyvä ohjelmakoodi etsii opiskelijoita, jotka eivät ole käyneet Moodlessa suorittamassa tehtäviä.


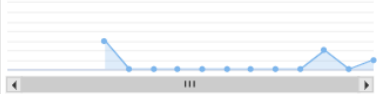
```
/* Get the students that have no access */
$maxnumberofweeks = 0;
foreach ($resultado as $tuple) {
    $arrayofaccess[] = array('userid' => $tuple->userid,
        'nome' => $tuple->firstname.' '.$tuple->lastname,
        'email' => $tuple->email);
    if ($tuple->week > $maxnumberofweeks) {
        $maxnumberofweeks = $tuple->week;
    }
}
```

Kuva 4 Opiskelijan aktiivisuuden tarkastelu

Datan visualisointi suoritetaan HTML-kielellä kirjoitetuilla funktioilla, joilla luodaan kuvaajia, joihin opiskelijan aktiivisuutta kuvaavat tiedot tulostetaan. Näistä kuvaajista lisäosan loppukäyttäjän on helppo tutkia opiskelijan aktiivisuutta pidemmällä aikavälillä ja havaita mahdolliset muutokset opiskelijan toiminnassa.

Opiskelijan aktiivisuutta esitellään kuvassa 5. Kuvasta nähdään opiskelijan aktiivisuus sekä numeroina että visuaalisessa muodossa. Kuvassa voidaan nähdä

opiskelijan aktiivisuus viikoittain. Opiskelijan lokidatasta etsitään jokaisena päivänä aktiviteetteja, ja mikäli aktiviteetti löydetään, merkitään tämä päivä aktiiviseksi. Aktiivisten päivien määrä lasketaan viikoittain ja tämä lukema esitetään kuvaajassa. Mikäli viikolta ei löydy yhtäkään aktiviteettia, opiskelijan aktiivisuus näytetään kuvaajassa nollana.

Opiskelijaa	Kurssiosumia	Käyntipäiviä	Viikoittainen käyntipäivien lukumäärä (Viikkojen lukumäärä: 12)	Avattuja materiaaleja
Samuli Leikkola 	26	6		0

Kuva 5 Opiskelijan aktiivisuus

## 5 POHDINTA

### 5.1 Työn eteneminen

Työn tekeminen osoittautui haastavammaksi kuin etukäteen osattiin odottaa. Kummallakaan työn tekemiseen osallistuneista opiskelijoista ei ollut aikaisempaa kokemusta Moodlen kanssa työskentelystä, joten työn alkuvaiheessa kohdattiin jonkin verran haasteita. Ensimmäiset viikot kuluivat aiheen opiskelun ja Moodlen toimintaan ja tietorakenteeseen tutustuttaessa. Moodlen haastava ja melko monimutkainen tietokanta ja rakenne tuottivat jatkuvasti jonkin verran ongelmia. Jälkikäteen ajateltuna työn alkuvaiheessa olisi tullut pyrkiä hankkimaan enemmän tietoa Moodlen käyttämästä tietorakenteesta ja Moodlen toiminnasta. Intensiivisemmän tutkimisen sijaan päädyttiin osaamista syventämään työn edetessä ja tämä aiheutti useita katkoja työn etenemisessä.

Varsinaisissa ohjelmointiosuuksissa vaadittavien HTML, PHP ja SQL-ohjelmointikielten osaamista löytyi jonkin verran jo etukäteen, joten ohjelmointiosuuksissa työskentely oli tehokkaampaa kuin tietokantaosuuksissa. Toki myös ohjelmointiosuuksissa törmättiin ongelmiin. Vaikeuksia tuottivat varsinkin työn alkuvaiheessa Moodlen ja sen lisäosien melko vaikeaselkoinen ohjelmakoodi, jota oli hankalaa muokata tehokkaasti työn alussa käytetyllä Visual Studio Code-ohjelmalla. Koodin muokkaaminen tehostui huomattavasti, kun työssä siirryttiin käyttämään Notepad++-editoria.

### 5.2 Työn tavoitteiden täytyminen

Kaikkia työlle asetettuja tavoitteita ei saatu parhaista yrityksistä huolimatta täytettyä. Uuteen asiaan tutustumiseen kului yllättävän paljon aikaa ja näin ollen jo alkujaan hieman tiukka aikataulu osoittautui hieman liian tiukaksi. Työssä tuotettuun lisäosaan saatiin kuitenkin sisällytettyä paljon haluttuja ominaisuuksia ja toiminnallisuutta.

Valmiiseen lisäosaan saatiin tuotettua tietokannasta tiedon hakuun pystyvät funktiot, joiden avulla pystytään käsittelemään opiskelijoiden aktiivisuutta kuvaavaa dataa. Tätä dataa pystytään myös tulostamaan loppukäyttäjän nähtäväksi sekä numeroina että myös visuaalisemmassa muodossa kuvaajina.

Lisäosasta jäi myös pois joitain tavoiteltuja ominaisuuksia. Tavoitelluista toiminnallisuuksista pois jouduttiin jättämään automaattinen ilmoitus opiskelijan pitkään alhaisena pysyneestä aktiivisuudesta ja myös opiskelijoiden tarkastelu opiskelijaryhmittäin jäi toteuttamatta. Ongelmaksi osoittautui opiskelijoiden ryhmätunnusten puuttuminen Moodlen tietokannasta. Opiskelijoiden tarkastelu ryhmittäin olisi vaatinut huomattavan paljon lisää selvitystyötä ja tutkimista, sekä myös TAMK:n avoimen datan portaalista saatavan datan käyttöä ja yhdistelemistä Moodleen tarjoamiin lokitietoihin. Tämä ominaisuus päädyttiin lopulta rajaamaan opinnäytetyön ulkopuolelle yhdessä ohjaajan kanssa.

### **5.3 Mahdollinen jatkokehitys**

Koska tässä työssä ei saatu toteutettua kaikkia lisäosaan haluttuja ominaisuuksia, on tätä projektia mahdollista vielä jatkokehittää. Lisäosaan tulisi vielä lisätä ainakin automaattinen hälytys opiskelijan jatkuvasta alhaisesta aktiivisuudesta ja myös muita lisäosan ominaisuuksia olisi mahdollista parannella ja tehostaa. Tämän työn jatkokehittäminen saattaisi olla hyvä opinnäytetyön aihe jollekin opiskelijalle.

Yhtenä merkittävimmistä jatkokehitysideoista olisi Moodlen tietokannan ja TAMK:n avoimen datan yhdisteleminen. Avoimesta datasta tulisi löytää opiskelijoiden ryhmätunnukset, joiden avulla opiskelijat voitaisiin jakaa omiin ryhmiinsä Moodlen tietokannassa. Jos avoimesta datasta saataisiin haettua opiskelijoiden tunnukset, voitaisiin näiden avulla luoda Moodlen tietokantaan oma taulu jokaiselle opiskelijaryhmälle. Tämä taulu sisältäisi kyseisen opiskelijaryhmän opiskelijat ja tätä taulua hyväksi käyttäen voitaisiin tarkastella opiskelijoiden aktiivisuutta heidän omaan opiskelijaryhmäänsä verrattuna.

Toinen mahdollinen tapa opiskelijoiden opiskelijaryhmän selvittämiseen olisi opiskelijan kurssien tarkastelu. Kun Moodleen luodaan kurssi, on sen tietoihin

mahdollista lisätä opiskelijaryhmä, jolle kurssi on tarkoitettu. Näin ollen tarkastelemalla jokaisen opiskelijan kursseja ja niille asetettuja ryhmätunnuksia voitaisiin tarkasteltavalle opiskelijalle löytää ryhmätunnus myös käyttämällä vain Moodlen tarjoamaa dataa. Tähän menetelmään liittyy kuitenkin omat haasteensa, joiden takia tämä ajatus lopulta tässä työssä hylättiin. Tämän toimintatavan luotettavuus ei ole kovinkaan suuri, koska opiskelijoiden on mahdollista suorittaa opintojaan vapaasti omassa tahdissaan. Näin ollen on mahdollista ja jopa todennäköistä, että opiskelija osallistuu jonkin toisen ryhmän kursseille ja tämän seurauksena hänet yhdistettäisiin useisiin eri opiskelijaryhmiin. Tämän tavan toisena suurena haittana on sen kuormittavuus. Jos jokaisen TAMK:n opiskelijan kohdalla joudutaisiin käymään läpi kaikki hänen kurssinsa vain ryhmätunnuksen selvittämiseksi, saatettaisiin päätyä tilanteeseen, jossa lisäosan toiminta olisi erittäin hidasta ja pahimmassa tapauksessa koko Moodlen toiminta saattaisi hidastua merkittävästi.

#### **5.4 Yhteistyön toimiminen ja parityön hyödyt**

Opinnäytetyötä tehtäessä yhteistyö toimi mallikkaasti ja ilman suurempia ongelmia. Toki kevään aikana maailmalla vaikuttanut COVID-19 pandemia aiheutti joi-takin ongelmia. Pandemian aiheuttamien varotoimien takia yhteistyö täytyi suorittaa erinäisten etäyhteyksien kautta. Työnteon kannalta olisi saattanut olla tehokkaampaa, jos töitä olisi pystytty tekemään yhdessä esimerkiksi koulun ti-loissa.

Kaiken kaikkiaan työparin kanssa toteutettu opinnäytetyö vaikuttaa varsin järke-vältä ja varteenotettavalta vaihtoehdolta. Useamman opiskelijan yhteistyönä to-teutettu opinnäytetyö tarjoaa mahdollisuuksia, joita yksittäisen opiskelijan tekemä työ ei mahdollista. Pareittain tehty työ mahdollistaa laajemman työn tekemisen ja tarjoaa opiskelijoille huomattavasti laajemmat mahdollisuudet valita työnsä aihe. Kun tekijöitä on kaksi, voidaan opiskelijoiden vahvuuksia ja kiinnostuksen koh-teita yhdistämällä tuottaa töitä, joita kumpikaan työparin jäsenistä ei yksin olisi kyennyt saamaan valmiiksi.

Myös työn rytmittäminen ja ajallaan valmiiksi saaminen on helpompaa yhteistyössä toisen opiskelijan kanssa. Opiskelijat työskentelevät todennäköisesti tehokkaammin ja aktiivisemmin opinnäytetyönsä parissa, kun he ovat oman valmistumisensa lisäksi osittain vastuussa myös toisen opiskelijan valmistumisesta.

## **5.5 Yhteenveto**

Yhteenvetona tätä projektia voidaan pitää melko onnistuneena. Kun huomioidaan kehittäjien kokemustaso ja melko tiukka aikataulu sekä maailmalla työn tekemisen aikana vallinnut poikkeustila, voidaan olla melko tyytyväisiä työn lopputulokseen.

Projektin aikana opittiin huomattavasti uusia asioita ja tekniikoita, ja näin ollen opinnäytetyön suorittamisesta saatiin hyötyä niin työn tilaajalle kuin toteuttajallekin. Moodlen kehittämisen parissa työskentely tutustutti alustan omien rakenteiden ja tietokantojen lisäksi myös syvemmin varsinkin HTML- ja PHP-ohjelmointikieliin, josta tulee varmuudella olemaan hyötyä tulevalla työuralla.

## LÄHTEET

What is Notepad++. 2020. Luettu 20.4.2020

<https://notepad-plus-plus.org/>

Microsoft. 2020. Visual Studio Code. Getting Started. Luettu 15.4.2020

<https://code.visualstudio.com/docs>

The PHP Group. 2020. Luettu 25.4.2020

<https://www.php.net/>

Moodle. 2020. About us. Luettu 25.4.2020

<https://moodle.com/about/>

Frank Moraes. 2020. What is HTML. Luettu 25.4.2020

[https://html.com/#What\\_is\\_HTML](https://html.com/#What_is_HTML)

w3schools. 2020. Introduction to SQL. Luettu 10.4.2020

[https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp)

Moodle. 2020. Plugin types. Luettu 20.4.2020

[https://docs.moodle.org/dev/Plugin\\_types](https://docs.moodle.org/dev/Plugin_types)