

# Moodlen analytiikkalisäosan kehittäminen

Alexi Majara

OPINNÄYTETYÖ  
Kesäkuu 2020

Tieto- ja viestintäteknikka  
Ohjelmistotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tieto- ja viestintäteknikka  
Ohjelmistotekniikka

MAJARA, ALEKSI:  
Moodlen analytiikkalisäosan kehittäminen

Opinnäytetyö 32 sivua, joista liitteitä 1 sivu  
Kesäkuu 2020

---

Työn tavoitteena oli toteuttaa Moodlen lisäosa Tampereen ammattikorkeakoululle. Moodle on virtuaalinen oppimisympäristö, jota käytetään oppilaitoksissa ympäri maailmaa.

Lisäosan avulla halutaan löytää ennakoivia merkkejä opiskelijoiden vaarasta pudota opetuksesta. Sen toiminta perustuu opiskelijoiden jättämien lokitietojen tarkasteluun kursseilla.

Työssä on käsitelty lisäosan alustavan käyttöliittymän toteuttamista. Käyttöliittymä koostuu aloitusnäköymästä, muutamasta välilehdestä ja välilehtien toiminnallisuuksista. Toiminnallisuuksia ovat lokitietojen tarkastelu ja lokitietojen hakeminen nimen avulla.

Lisäosan tarkoituksena on helpottaa tutkintovastaavan, opettajatuutorin ja opinto-ohjaajan työtä tarkasteltaessa opiskelijoiden opiskeluaktiivisuutta. Lisäosaa voidaan hyödyntää tulevaisuudessa mahdollisesti muissakin oppilaitoksissa, joissa käytetään Moodlea.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
ICT Engineering  
Software Engineering

MAJARA, ALEKSI:  
Development of the Moodle analytics plug-in

Bachelor's thesis 32 pages, appendices 1 page  
June 2020

---

The goal of the work was to implement the Moodle plug-in for Tampere University of Applied Sciences. Moodle is a virtual learning environment for educational institutions around the world.

The plug-in is intended to find proactive signs of the students that are at risk of dropping out. Its operation is selected to view the log information left by the student in courses.

The implementation of the preliminary user interface of the plug-in has been discussed in the work. The interface consists of a home view, some tabs and functionalities. The functionalities include viewing log information and retrieving log information by name.

The purpose of the plug-in is to help e.g. head of degree program, teacher tutor and student councillor, to identify the students that are risk of dropping out of studies. In the future, it will be possible to use the plugin in other educational institutions that uses Moodle.

---

Key words: moodle, virtual learning environment, plug-in

## SISÄLLYS

1	JOHDANTO .....	6
2	TEORIA .....	7
	2.1 Kehitysympäristöt.....	7
	2.1.1 Moodle.....	7
	2.1.2 Peppi .....	8
	2.1.3 TAMK avoin data .....	8
	2.2 Kehitystyökalut.....	9
	2.2.1 PHP-ohjelmointikieli.....	9
	2.2.2 HTML-kieli .....	9
	2.2.3 Notepad++.....	10
	2.2.4 MySQL Workbench .....	10
3	SUUNNITTELU.....	11
	3.1 Sovelluksen toimintaympäristö.....	11
	3.1.1 Lisäosan tyyppi.....	11
	3.2 Sovelluksen toiminta .....	11
	3.2.1 Aktiivisuuden graafinen esitys .....	12
4	TOTEUTUS .....	14
	4.1 Sovelluksen näkymät .....	14
	4.1.1 Loki-välilehti.....	16
	4.1.2 Analysoi-välilehti.....	21
	4.1.3 Luokkatunnukset-välilehti .....	25
5	LOPPUTULOKSET JA JATKOKEHITYS .....	28
	5.1 Lopputulokset.....	28
	5.1.1 Lisäosan toimintaympäristö .....	28
	5.1.2 Lisäosan kehitysympäristöt ja työkalut .....	28
	5.1.3 Lisäosan toiminta.....	29
	5.2 Jatkokehitys .....	29
6	POHDINTA .....	30
	LÄHTEET.....	31
	LIITE .....	32

**ERITYISSANASTO**

tietokanta	tiedontallennuskomponentti tietotekniikassa
lähdekoodi	kooditeksti, jonka perusteella esimerkiksi verkkosivut rakennetaan
HTTPS-protokolla	salausprotokolla, jonka avulla internetissä data suojataan tiedonsiirron aikana
GET-menetelmä	menetelmä, jolla haetaan dataa esimerkiksi verkkosivulta
POST-menetelmä	menetelmä, jolla lähetetään dataa esimerkiksi verkkosivulle
kehitysympäristö	virtuaaliympäristö, jossa kehitettävä työ toimii
kehitystyökalu	ohjelmisto tai teknologia, jota tarvitaan työn kehittämiseen
rajapinta	väylä, jota pitkin kaksi ohjelmaa voi kommunikoida keskenään
SQL-kysely	tietokantakysely koodissa, jonka avulla haetaan vastauksia tietokannasta haluttujen kriteerien mukaisesti
HEX-värikoodi	värien merkintätapa koodikielissä, jonka avulla saadaan tarkasti haluttu värisävy
API-avain	henkilökohtainen tunnisteavain, jolla saa oikeudet tiettyyn palveluun

## 1 JOHDANTO

Opinnäytetyön tehtävänä oli ohjelmoida lisäosa Tampereen ammattikorkeakoulun Moodle-palveluun. Moodle on yleisesti käytössä oleva virtuaalinen oppimisjärjestelmä. Tampereen ammattikorkeakoulun lisäksi myös monet muut koulut Suomessa käyttävät Moodlea oppimisympäristönä. Moodlen lähdekoodi on kaikille vapaasti ladattavissa internetistä, mikä tekee palvelusta ilmaisen käytettäväksi.

Lisäosan tarkoituksena on helpottaa muun muassa tutkintovastaavan, opettaja-tuutorin ja opinto-ohjaajan työtä tarkasteltaessa oppilaiden opiskeluaktiivisuutta. Tarkastelun tavoitteena on löytää ne opiskelijat, jotka ovat putoamisvaarassa opinnoista. Tarkastelun ansiosta putoamisvaarassa oleville opiskelijoille on mahdollista kohdistaa tukitoimia.

Aiemmin aktiivisuuden tarkastelua tehtiin Microsoftin Excelillä, jonne tarkasteltava data piti manuaalisesti tuoda jälkeinpäin Moodle-kurssien lokitiedoista. Lisäosalla mahdollistetaan aktiivisuuden tarkasteleminen reaaliajassa, jotta voidaan reagoida liian vähäiseen aktiivisuuteen tarpeeksi ajoissa. Lisäksi lisäosalla pyritään vähentämään aktiivisuuden tarkasteluun käytettyjä työtunteja, jolloin opettajille jää enemmän aikaa tukea putoamisvaarassa olevia opiskelijoita.

Lisäosaa kehitettiin yhteistyössä toisen opiskelijan ja Tampereen ammattikorkeakoulun kanssa. Tässä työssä on toteutettu lisäosan alustava käyttöliittymä ja sen toiminnallisuuksia Moodlen tietokantatauluja hyväksikäyttäen.

## 2 TEORIA

Teoria osuudessa käydään läpi työssä oleellisesti käytettävät tai tarvittavat kehitysympäristöt ja niiden ympärillä toimivat kehitystyökalut.

### 2.1 Kehitysympäristöt

Lisäosa tehdään toimivaksi oppimisjärjestelmä Moodlea. Tampereen ammattikorkeakoulu käyttää Moodlea versiota 3.5, joten lisäosa kehitetään tukemaan lähtökohtaisesti tätä versiota. Tampereen ammattikorkeakoulu käyttää Moodlea nimellä Tabula.

#### 2.1.1 Moodle

Moodle eli "Modular Object-Oriented Dynamic Learning Environment" on virtuaalinen oppimisjärjestelmä, jonka lähdekoodi on ilmaiseksi ladattavissa internetistä. Opettajat voivat luoda Moodleen yksityisiä tai yleisiä kursseja, joihin oppilaat voivat liittyä. Moodlea voidaan käyttää pelkästään verkkokurssi alustana tai pelkästään lähiopetustuntien tukena sekä sitä voidaan käyttää näiden kahden yhdistelmänä. Yhdistelmässä voidaan toteuttaa osa kurssin suorituksista verkko-opintoina ja osa suorituksista voidaan toteuttaa lähiopetuksena. (Moodle, 2019.)

Moodle sisältää paljon vakio-ominaisuuksia, joiden avulla sitä on helppo muokata koulutusjärjestelmälle tai yksittäiselle kurssille sopivaksi. Moodleen on ladattavissa internetistä myös paljon lisäosia, jotka parantavat Moodleen käyttökokemusta. Moodlea on mahdollisuus käyttää esimerkiksi kalenteria, joka näyttää käyttäjälle kurssien tapaamiset, tapahtumat ja palautuksien takarajat. (Moodle, 2019.)

Tampereen ammattikorkeakoulun käyttämässä Moodlessa eli Tabulassa ei ole mahdollista peruskäyttäjän ladata lisäosia omalle käyttäjälleen. Tabulan ylläpitäjät huolehtivat järjestelmän sisällöstä ja sen valvomisesta. Ylläpitäjät hankkivat tarpeelliset työkalut ja lisäosat järjestelmään.

### **2.1.2 Peppi**

Peppi on opetuskokonaisuuteen suunniteltu järjestelmä, joka kattaa esimerkiksi opiskelijoiden arvioinnit ja opetussuunnitelmat. Järjestelmää ylläpitää ja kehittää Peppi-konsortio. Pepin omistavat Metropolia ja Tampereen ammattikorkeakoulu. Peppi voidaan liittää myös muihin järjestelmiin kuten työssä käytettyyn Moodleen. (Emilia Tietäväinen, 2018).

Tampereen ammattikorkeakoulun käyttämä Tabula eli Moodle hyödyntää Peppiä esimerkiksi opintojakson toteutuksen luomiseen. Pepin avulla saadaan siirrettyä vaivattomasti opintojakson nimi, koodi, opettajat ja hyväksytyt opiskelijat Tabulaan. (TAMK Tabula, 2020).

### **2.1.3 TAMK avoin data**

Tampereen ammattikorkeakoulun avoin data on mahdollista hakea julkisen palveluväylän rajapintojen kautta. Palveluväylä on avoin tiedonsiirtoreitti, jonka avulla voidaan siirtää tietoja palveluiden välillä vakioidulla tavalla. Rajapintojen käyttämiseen vaaditaan salausavain, joka pystytään hakemaan sähköisellä lomakkeella ylläpitäjältä. (TAMK avoin data.)

Avoimen datan palvelusta voidaan hakea muun muassa

- Opetussuunnitelmia
- Opintojaksoja
- Opintojaksojen toteutuksia



Avoimen datan tarjoamia rajapintoja kutsutaan HTTPS-protokollaa käyttämällä ja käyttötarkoituksen mukaan GET- tai POST-menetelmää käyttäen. (TAMK avoin data.)

## **2.2 Kehitystyökalut**

Tässä työssä kehitettävää Moodlen lisäosaa kehitetään lähinnä PHP-ohjelmointikielillä. PHP-kieltä voidaan kirjoittaa monella tekstinmuokkausohjelmalla, mutta tähän työhön valittiin Notepad++. Notepad++ valittiin sen nopeuden ja monipuolisuuden takia.

### **2.2.1 PHP-ohjelmointikieli**

PHP eli "HyperText Preprocessor" on ohjelmointiin soveltuva scriptikieli. PHP-ohjelmointikieltä käytetään eniten verkkosivujen luomiseen. PHP-kielen ansiosta verkkosivuille voidaan lisätä monenlaisia toimintoja. Sillä voidaan esimerkiksi tarkastella ja lähettää lomaketietoja. PHP-kielillä voidaan myös luoda monimutkaisia sovelluskokonaisuuksia, kuten esimerkiksi monenlaisia sisällönhallintajärjestelmiä. (Keski-Suomen verkkomedia, 2009.)

### **2.2.2 HTML-kieli**

HTML-kieli eli "HyperText Markup Language" on verkkosivujen esittämiseen kehitetty kieli. HTML-kieli ei ole varsinaisesti virallinen ohjelmointikieli, koska sillä ei voida tehdä tietokoneohjelmia, kuten esimerkiksi PHP- tai C-kielillä. HTML-kieli on tarkoitettu lähinnä www-kotisivujen tekemiseen ja sivuille, joissa tarvitaan paljon visuaalisia ominaisuuksia. (MVnet, 2006.)

### 2.2.3 Notepad++

Notepad++ on tekstinmuokkausohjelma, joka tukee monia ohjelmointikieliä. Notepad++ on ilmainen ohjelma ja se on vapaasti ladattavissa internetistä. Sitä voidaan käyttää Microsoftin Windows -käyttöjärjestelmissä. (Don Ho, 2005.)

Notepad++ perustuu tehokkaaseen Scintilla-muokkauskomponenttiin, joka mahdollistaa ohjelman tehokkaan käytön. Notepad++ on kirjoitettu C++ -ohjelmointikielillä. (Don Ho, 2005.)

Notepad++ yrittää vähentää prosessorin kuormitusta optimoimalla mahdollisimman monta rutiinia. Tällä pyritään vähentämään ohjelman käytöstä aiheutuvia hiilidioksidipäästöjä. (Don Ho, 2005.)

### 2.2.4 MySQL Workbench

MySQL Workbench on ohjelmisto, jolla voidaan tarkastella ja muokata tietokantoja. Se on yhtenäinen visuaalinen ohjelmisto tietokanta-arkkitehdeille, ohjelmistokehittäjille sekä niiden ylläpitäjille. MySQL Workbench on saatavilla Windowsille, Linuxille ja Mac OS X:lle. (MySQL, 2020.)

Tässä työssä MySQL Workbenchia käytetään Moodlen tietokantataulujen tarkastelemiseen. Workbenchin avulla pyritään helpottamaan lisäosalla tehtyjen sql-kyselyiden muodostamista.

## **3 SUUNNITTELU**

### **3.1 Sovelluksen toimintaympäristö**

Tässä työssä toteutettavan sovelluksen on tarkoitus toimia erottuvana lisäosana käyttöympäristöstä eli Moodlesta. Erottuvalla lisäosalla tarkoitetaan sitä, että lisäosaa käytettäessä kaikki käyttäjälle näkyvä toiminta tehdään erillisellä sivulla. Lisäosaa käytettäessä verkkoselain avaa uuden verkkosivun, jossa on Moodlesta erottuva ulkoasu. Lisäosa käyttää Moodlen tietokantaa erillisestä ulkoasusta ja sivusta huolimatta.

#### **3.1.1 Lisäosan tyyppi**

Moodleen on mahdollista ladata erityyppisiä lisäosia kuten raportteja, lokikirjastoja ja teemoja (Moodle, 2017). Tähän työhön valittiin käytettäväksi lisäosatyyppejä nimeltä lohko. Lohkotyyppinen lisäosa valittiin, koska se mahdollistaa HTML-ohjelmointikielen käyttämisen lisäosassa. HTML-kielellä voidaan tehdä verkkosivuille muun muassa parempia tilastollisesti tarkasteltavia diagrammeja kuin PHP-kielellä.

### **3.2 Sovelluksen toiminta**

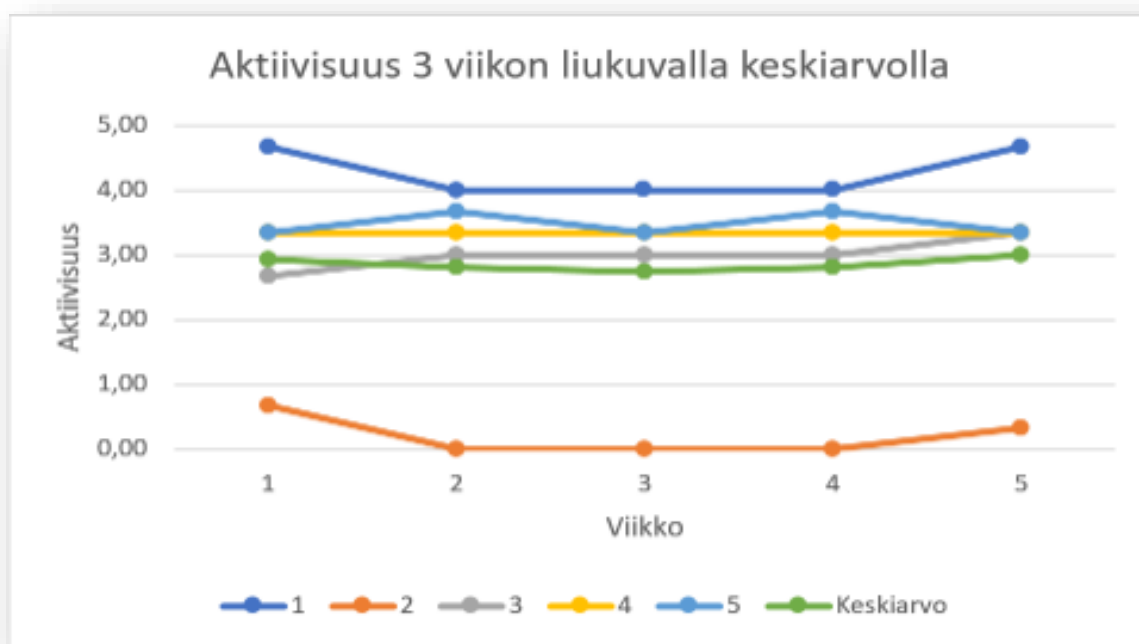
Sovelluksen toiminta perustuu opiskelijoiden jättämiin lokitietoihin Moodlen sisällä. Käyttäjältä jää lokitietoja Moodlen tietokantaa kaikesta hänen suorittamasta toiminnasta. Tässä työssä keskitytään käyttäjän jättämiin lokimerkintöihin kursseilla. Lokimerkintöjä jää esimerkiksi aina silloin, kun käyttäjä avaa kurssilla linkkejä, tiedostoja tai palauttaa tiedostoja. Käyttäjä on tässä tapauksessa opiskelija.

Käyttäjän jättämien lokitietojen perusteella sovellus tutkii käyttäjän aktiivisuutta Moodle-käyttäjänä. Mikäli käyttäjältä ei tallennu lokitietoja tietynä ajanjaksona, sovellus huomaa sen ja pyrkii huomauttamaan siitä käyttäjää tai käyttäjästä vastuussa olevaa henkilöä.

Lokitietojen perusteella sovellus piirtää verkkosivulle diagrammin, josta voidaan ennakoida käyttäjän liian vähäinen aktiivisuus Moodlessa. Sovelluksella pyritään mahdollistamaan käyttäjän varoittaminen tässä vaiheessa esimerkiksi lähettämällä hänelle sähköpostiviesti havaitusta tilanteesta.

### 3.2.1 Aktiivisuuden graafinen esitys

Sovelluksen on tarkoitus näyttää haluttujen käyttäjien aktiivisuus Moodlessa yksinkertaisena viivadiagrammina. Sovelluksessa esiintyvä viivadiagrammi on muodostettu alla olevaan Excel-diagrammiin (kuvio 1) perustuen. Excel-diagrammissa on käytetty viiden käyttäjän aktiivisuutta viiden viikon ajan Moodlessa. Diagrammissa käyttäjät on merkattu erivärisillä viivoilla. Vaaka-akselilla oleva viikko tarkoittaa normaalia kalenteriviikkoa.



KUVIO 1. Viiden Moodle-käyttäjän aktiivisuudet kolmen viikon liukuvalla keskiarvolla

Excel-diagrammissa pystyakselilla oleva aktiivisuus on laskettu arvosanojen avulla, mutta Moodle-sovelluksessa tarkoitus on tarkastella vain aktiivisuutta. Eli

käyttäjä on aktiivinen tai ei aktiivinen. Yksikin tapahtuma, jonka käyttäjä tekee Moodlessa tarkoittaa aktiivisuutta. Tämä mahdollistaa myös sen, että opiskelija voi saada arvosanaksi nollan esimerkiksi tentistä, mutta silti sovellus näyttää opiskelijan olevan aktiivinen.

Kolmen viikon liukuvalla keskiarvolla tarkoitetaan sitä, että käyttäjän aktiivisuutta tarkastellaan aina kolmen viikon jaksoissa ja lasketaan niiden keskiarvo. Jos käyttäjän liukuva keskiarvo kolmen peräkkäisen viikon ajalta on nolla, käyttäjä on ei aktiivinen. Käyttäjä leimataan aktiiviseksi heti, kun hän tekee jollain kurssilla yhden tapahtuman kolmen viikon aikana, esimerkiksi hän avaa linkin tai palauttaa tiedoston. Kolmen viikon liukuvaa keskiarvoa käytetään siksi, että sillä pyritään minimoimaan lukukausien aikana olevien lomien tai sairastumisien vaikutukset aktiivisuuteen.

Koko järjestelmän käyttäjiä, jotka tässä tapauksessa ovat opiskelijoita ei voida laittaa samaan graafiseen esitykseen, koska oppilaita saattaa olla tuhansia. Sovelluksen viivadiagrammissa on tarkoitus näyttää TAMK:n avoimesta datasta löytyvien luokkatunnusten avulla ryhmittäin oppilaiden aktiivisuudet. Luokkatunnuksen alla on yleensä noin 30 oppilasta, jolloin diagrammi pysyy luettavana ja sieltä voidaan poimia ei aktiiviset oppilaat.

Tampereen ammattikorkeakoulun käyttämä Moodle ei tarjoa mahdollisuutta ryhmitellä suoraan Moodlen sisällä oppilaita luokkatunnusten perusteella, vaan oppilaat ryhmitellään Peppi-palvelun avulla. Eli TAMK:n avoimen datan palvelusta on tarkoitus hakea luokkatunnus ja luokkatunnuksen avulla tarkoitus hakea Peppistä kyseisen luokkatunnuksen alla olevat oppilaat ja tuoda ne Moodleen.

## 4 TOTEUTUS

Tässä opinnäytetyössä toteutettavaa sovellusta lähdettiin tekemään jo olemassa olevan sovelluksen päälle. Jo olemassa olevan sovelluksen toimintaperiaate on hyvin lähellä samaa kuin tässä työssä tehdyn sovelluksen, mutta se toimii vain yhden Moodle-kurssin sisäisesti. Työssä tehdyn sovelluksen toiminnassa haluttiin tarkastella käyttäjien kaikkia valittuja kursseja kokonaisuutena. Tämän vuoksi valmis sovellus ei soveltunut toimeksiantajan käyttötarkoitukseen.

### 4.1 Sovelluksen näkymät

Sovellusta lähdettiin tekemään siten, että sitä olisi helppo jatkokehittää. Päädettiin luomaan HTML-kieltä hyväksi käyttäen sovellukselle navigointivalikko (kuva 1), joka sijaitsee sivujen yläpaneelissa. Navigointivalikolla voidaan siirtyä sovelluksessa, miten halutaan eli valikossa ei tarvitse siirtyä loogisessa järjestyksessä. Valikossa voidaan esimerkiksi hypätä suoraan aloitusnäköymästä analysointinäköymään.



KUVA 1. Lisäosan päänäkymä, jossa näkyvissä navigointivalikko

Alla olevassa näyttökuvassa (kuva 2) on lisäosan päänäkymän lähdekoodi. Lähdekoodi on toteutettu HTML-kielellä. Koodi sisältää navigointivalikon toteuttamisen ja kaksi yksinkertaista tekstin tulostusta, joissa tervetuloivotus ja ilmoitus beta-versiosta.

Koodi alkaa navigointivalikon toteuttamisella. Toteutus alkaa valikon lähtökohdan määrittämisellä, joka tässä tapauksessa on nolla, koska valikon halutaan alkavan heti sivun alusta. Navigointivalikon ulkoasun graafiset muodot ja värit toteutetaan riveillä 25–38. Navigointivalikon taustan muodostamista ohjailaan muun muassa valikon tyyllillä, reunuksilla, taustanvärillä ja sijainnilla. Taustan HEX-värikoodi "#333" tarkoittaa hieman haalistunutta mustaa väriä.

Taustan luomisen jälkeen toteutetaan valikon painikkeiden sisällön grafiikka, jossa ovat muun muassa painikkeiden väri, sijainti ja reunukset. Rivillä 40 esiintyvät "display" tarkoittaa valikon painikkeiden tyyliä. Tässä tapauksessa painikkeet on eroteltu toisistaan näkymättömillä pystyviivoilla. Riveillä 46–50 määritellään "Aloitus"-painikkeen tyyli. Aloituspainikkeen ollessa oranssi ollaan lisäosan aloitusnäkyssä. Aloituspainikkeen tarkoitus oli muuttua mustaksi, kun ollaan muissa näkymissä, mutta tässä lisäosaversiossa ei saatu sitä toimimaan.

Navigointivalikon jälkeen määritellään sivun runko. Riveillä 55–61 määritellään tiedostot, joihin valikon painikkeista päädytään. Esimerkiksi "Loki"-painikkeesta siirrytään tiedostoon, jossa haetaan ja näytetään lokidataa.

Lopuksi tulostetaan rivillä 63 tervetuloivotus ja rivillä 64 ilmoitus beta-versiosta.

```

20
21 <!DOCTYPE html>
22 <html>
23 <head>
24 <style>
25 body {margin:0;}
26 ul {
27   list-style-type: none;
28   margin: 0;
29   padding: 0;
30   overflow: hidden;
31   background-color: #333;
32   position: fixed;
33   top: 0;
34   width: 100%;
35 }
36 li {
37   float: left;
38 }
39 li a {
40   display: block;
41   color: white;
42   text-align: center;
43   padding: 14px 16px;
44   text-decoration: none;
45 }
46 li a:hover:not(.active) {
47   background-color: #111;
48 }
49 .active {
50   background-color: #ff9900;
51 }
52 </style>
53 </head>
54 <body>
55 <ul>
56 <li><a class="active" href="main.php">Aloitus</a></li>
57 <li><a href="lokitesti.php">Loki</a></li>
58 <li><a href="analysoi.php">Analysoi</a></li>
59 <li><a href="opiskelijat.php">Opiskelijat</a></li>
60 <li><a href="test2.php">Luokkatunnukset</a></li>
61 </ul>
62 <div style="padding:20px;margin-top:30px;background-color: white;height:1500px;">
63 <h1>Tervetuloa analysoimaan oppilaiden koulunkäyntiä</h1>
64 <h2>Tämä on beta versio</h2>
65 </div>
66 </body>
67 </html>

```

KUVA 2. Navigointivalikon ja päänäkymän lähdekoodi

#### 4.1.1 Loki-välilehti

Ensimmäisenä ominaisuutena sovelluksessa ei niinkään ole varsinainen analysointityökalu, vaan lokitietojen tarkastustyökalu. Lokinäkömystä (kuva 3) nähdään viimeaikaiset käyttäjien tekemät toiminnallisuudet listattuna. Listasta nähdään lokimerkinnän id-numero, tapahtuman nimi, käyttäjän id ja aikaleima. Käyttäjän id:n avulla tiedetään lokimerkinnän tekijä. Tapahtuman nimestä saadaan suuntaa-antavaa informaatiota, mitä käyttäjä on tehnyt.



Lokidataa on siivottu sen verran, että siinä näkyy vain lokitiedot, jotka käyttäjät ovat tehneet kurssien sisällä. Tämän ansiosta saadaan tarkemmin näkymään käyttäjien aktiivisuudet kursseilla.

Aloitus			
Loki			
Analysoi			
Opiskelijat			
Luokkatunnukset			
<b>LOKIDATAA TAULUSTA LOG STANDARD LOG</b>			
<b>ID</b>	<b>Tapahtuma</b>	<b>Käyttäjän ID</b>	<b>Aikaleima</b>
51415	\mod_assign\event\submission_status_viewed	10003	14-05-2020
51399	\mod_assign\event\submission_status_viewed	2	11-05-2020
51398	\mod_assign\event\submission_status_viewed	2	11-05-2020
51392	\assignsubmission_comments\event\comment_created	2	22-04-2020
51391	\mod_assign\event\submission_status_viewed	2	22-04-2020
51390	\mod_forum\event\discussion_viewed	2	22-04-2020
51389	\mod_forum\event\course_module_viewed	2	22-04-2020
51387	\mod_assign\event\submission_status_viewed	2	22-04-2020
51386	\mod_assign\event\assessable_submitted	2	22-04-2020
51383	\assignsubmission_file\event\submission_updated	2	22-04-2020
51382	\assignsubmission_file\event\assessable_uploaded	2	22-04-2020
51381	\assignsubmission_onlinetext\event\submission_updated	2	22-04-2020
51380	\assignsubmission_onlinetext\event\assessable_uploaded	2	22-04-2020
51379	\mod_assign\event\submission_form_viewed	2	22-04-2020
51378	\mod_assign\event\submission_form_viewed	2	22-04-2020
51377	\mod_assign\event\submission_status_viewed	2	22-04-2020
51376	\mod_assign\event\submission_status_viewed	2	22-04-2020
51375	\mod_assign\event\submission_status_viewed	2	22-04-2020

KUVA 3. Lokitapahtumia taulusta ”standard log”

Lokitapahtumat haetaan PHP-koodilla (kuva 4), jossa ensimmäiseksi muodostetaan yhteys Moodlen tietokantaan. Tietokantaan voidaan jatkossa yhdistää Moodlen omaa konfigurointi tiedostoa hyväksikäyttäen, mutta tässä työssä on käytetty manuaalista tietokantaan yhdistämistä, testauksen helpottamiseksi. Ohjelma palauttaa käyttäjälle ilmoituksen, jos yhteyttä ei voida muodostaa tietokantaan.

Riviltä 68 alkaen määritetään sql-kysely tietokannalle, jossa haetaan lokidataa taulusta ”standard log”. Rivillä 68 valitaan taulusta haettavaksi id, tapahtuman nimi käyttäjän id, kontekstitaso ja aikaleima. Kontekstitasolla tarkoitetaan

Moodlen osaa, johon lokimerkintä tehdään. Kurssikohtaiset lokimerkinnät on määritelty numerolla 70. Rivillä 69 muunnetaan aikaleima tunnistettavaan muotoon, jossa on päivä, kuukausi ja vuosi. Lokidatan hakeminen järjestyksessä uusin ensin, määritellään rivillä 72 ja se rajoitetaan maksimissaan tuhanteen hakutulokseen rivillä 73.

Riviltä 76 alkaen tietokannasta saatu lokidata laitetaan yksinkertaiseen taulukkoon, joka tulostetaan käyttäjälle näkyviin. Käyttäjälle ilmoitetaan, jos lokidataa ei löytynyt tietokannasta. Lopuksi yhteys tietokantaan suljetaan.

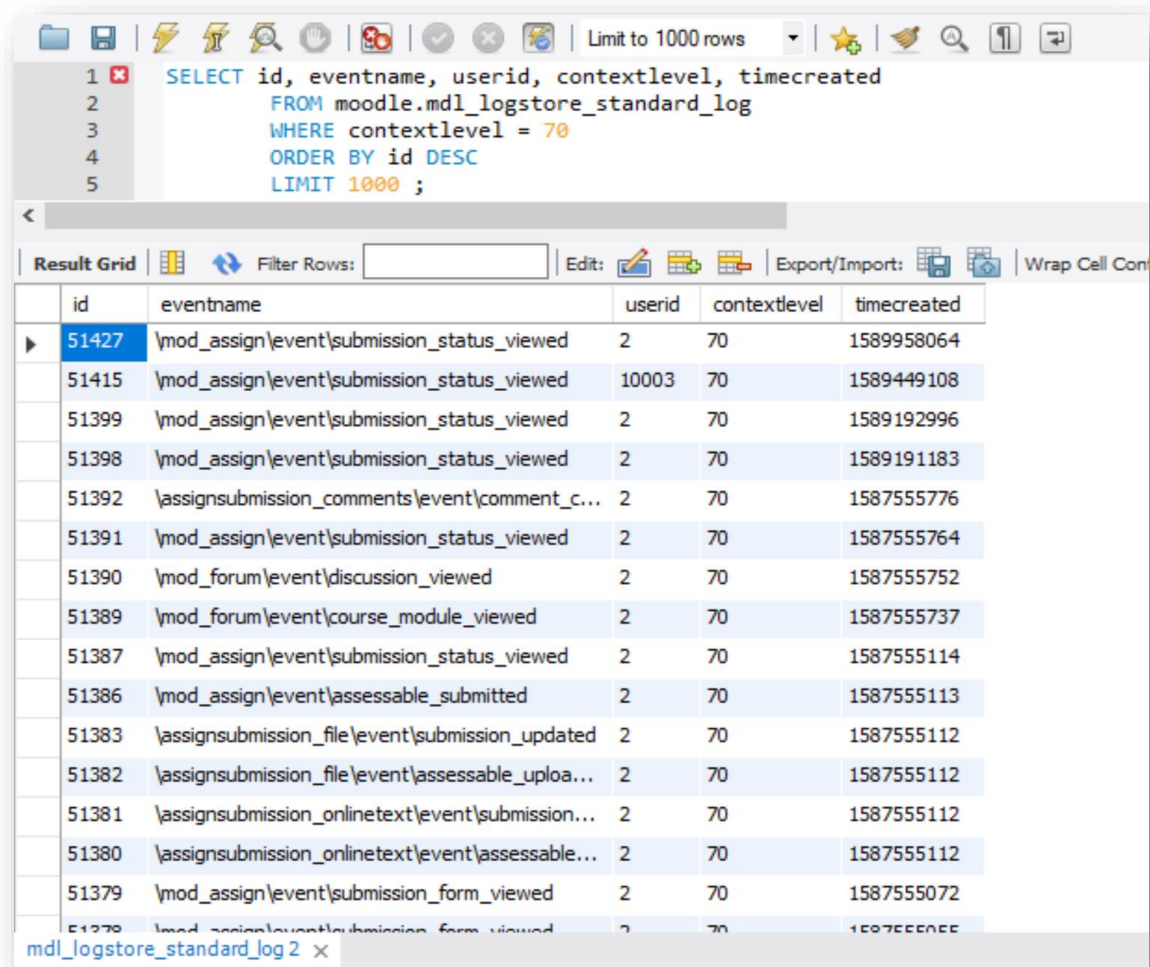
```

53 <?php
54
55
56 $servername = "localhost";
57 $username = "root";
58 $password = "";
59 $dbname = "moodle";
60
61 // Create connection
62 $conn = new mysqli($servername, $username, $password, $dbname);
63 // Check connection
64 if ($conn->connect_error) {
65     die("Connection failed: " . $conn->connect_error);
66 }
67
68 $sql = "SELECT id, eventname, userid, contextlevel, timecreated,
69         FROM_UNIXTIME( timecreated, '%d-%m-%Y' ) AS timecreated
70         FROM mdl_logstore_standard_log s
71         WHERE contextlevel = 70
72         ORDER BY id DESC
73         LIMIT 1000 ";
74 $result = $conn->query($sql);
75
76 if ($result->num_rows > 0) {
77     echo "<table><tr><th>ID</th><th>Tapahtuma</th><th>Käyttäjän ID</th><th>Aikaleima</th></tr>";
78     // output data of each row
79     while($row = $result->fetch_assoc()) {
80         echo "<tr><td>".$row["id"]."</td>
81             <td>".$row["eventname"]."</td>
82             <td>".$row["userid"]."</td>
83             <td>".$row["timecreated"]."</td></tr>";
84     }
85     echo "</table>";
86 } else {
87     echo "0 results";
88 }
89 $conn->close();
90 ?>

```

KUVA 4. PHP-koodi lokitietojen hakemiseen ja näyttämiseen

Edellä mainittua sql-kyselyä voidaan tarkastella myös MySQL-ohjelmalla (kuva 5). MySQL-ohjelmalla tehty kysely antaa samat lokitiedot kuin kuvassa 3. MySQL-ohjelmistolla haetussa tietokantataulussa nähdään, että kaikki haun tulokset sisältävät saman kontekstitason 70 ja aikaleima on hankalasti luettavissa.



The screenshot shows a MySQL query editor window. The query is as follows:

```

1 SELECT id, eventname, userid, contextlevel, timecreated
2 FROM moodle.mdl_logstore_standard_log
3 WHERE contextlevel = 70
4 ORDER BY id DESC
5 LIMIT 1000 ;

```

The results are displayed in a table grid with the following columns: id, eventname, userid, contextlevel, and timecreated. The first row is highlighted in blue.

id	eventname	userid	contextlevel	timecreated
51427	\mod_assign\event\submission_status_viewed	2	70	1589958064
51415	\mod_assign\event\submission_status_viewed	10003	70	1589449108
51399	\mod_assign\event\submission_status_viewed	2	70	1589192996
51398	\mod_assign\event\submission_status_viewed	2	70	1589191183
51392	\assignsubmission_comments\event\comment_c...	2	70	1587555776
51391	\mod_assign\event\submission_status_viewed	2	70	1587555764
51390	\mod_forum\event\discussion_viewed	2	70	1587555752
51389	\mod_forum\event\course_module_viewed	2	70	1587555737
51387	\mod_assign\event\submission_status_viewed	2	70	1587555114
51386	\mod_assign\event\assessable_submitted	2	70	1587555113
51383	\assignsubmission_file\event\submission_updated	2	70	1587555112
51382	\assignsubmission_file\event\assessable_uploa...	2	70	1587555112
51381	\assignsubmission_onlinetext\event\submission...	2	70	1587555112
51380	\assignsubmission_onlinetext\event\assessable...	2	70	1587555112
51379	\mod_assign\event\submission_form_viewed	2	70	1587555072
51378	\mod_assign\event\submission_form_viewed	2	70	1587555055

KUVA 5. MySQL-ohjelmalla tehty sql-kysely

MySQL-ohjelmalla voidaan myös hakea koko "standard log"-tietokantataulun sisältö (kuvat 6 ja 7). Koko tietokantataulu sisältää liikaa dataa lisäosaa varten, joten sen takia sitä on muokattu kuvan 5 mukaiseksi.

id	eventname	component	action	target	objecttable	objectid	crud	edulevel	contextid	contextlevel	contextinstanceid	userid
1	\core\event\user_loggedin	core	loggedin	user	user	2	r	0	1	10	0	2
2	\core\event\user_password_updated	core	updated	user_password	NULL	NULL	u	0	5	30	2	2
3	\core\event\user_updated	core	updated	user	user	2	u	0	5	30	2	2
4	\core\event\config_log_created	core	created	config_log	config_log	543	c	0	1	10	0	2
5	\core\event\config_log_created	core	created	config_log	config_log	544	c	0	1	10	0	2
6	\core\event\config_log_created	core	created	config_log	config_log	545	c	0	1	10	0	2
7	\core\event\config_log_created	core	created	config_log	config_log	546	c	0	1	10	0	2
8	\core\event\config_log_created	core	created	config_log	config_log	547	c	0	1	10	0	2
9	\core\event\config_log_created	core	created	config_log	config_log	548	c	0	1	10	0	2
10	\core\event\config_log_created	core	created	config_log	config_log	549	c	0	1	10	0	2
11	\core\event\config_log_created	core	created	config_log	config_log	550	c	0	1	10	0	2
12	\core\event\config_log_created	core	created	config_log	config_log	551	c	0	1	10	0	2
13	\core\event\config_log_created	core	created	config_log	config_log	552	c	0	1	10	0	2
14	\core\event\config_log_created	core	created	config_log	config_log	553	c	0	1	10	0	2
15	\core\event\config_log_created	core	created	config_log	config_log	554	c	0	1	10	0	2

KUVA 6. "Standard log"-tietokantataulun sisältö (osa 1)

Kokonainen "standard log"-tietokantataulu sisältää muun muassa toiminnan, toiminnan kohteen ja kurssi-id:n.

crud	edulevel	contextid	contextlevel	contextinstanceid	userid	courseid	relateduserid	anonymous	other	timecreated	origin
r	0	1	10	0	2	0	NULL	0	a:1:{s:8:"username";s:5:"admin"};	1581933922	web
u	0	5	30	2	2	0	2	0	a:1:{s:14:"forgottenreset";b:0};	1581934243	web
u	0	5	30	2	2	0	2	0	N;	1581934243	web
c	0	1	10	0	2	0	NULL	0	a:4:{s:4:"name";s:17:"notloggedinroleid";s:8:"o...	1581934254	web
c	0	1	10	0	2	0	NULL	0	a:4:{s:4:"name";s:11:"guestroleid";s:8:"oldvalu...	1581934254	web
c	0	1	10	0	2	0	NULL	0	a:4:{s:4:"name";s:17:"defaultuserroleid";s:8:"o...	1581934254	web
c	0	1	10	0	2	0	NULL	0	a:4:{s:4:"name";s:16:"creatornewroleid";s:8:"o...	1581934254	web
c	0	1	10	0	2	0	NULL	0	a:4:{s:4:"name";s:17:"restorernewroleid";s:8:"o...	1581934254	web
c	0	1	10	0	2	0	NULL	0	a:4:{s:4:"name";s:9:"exportlog";s:8:"oldvalue";...	1581934254	web
c	0	1	10	0	2	0	NULL	0	a:4:{s:4:"name";s:28:"contactdataprotectionof...	1581934254	web
c	0	1	10	0	2	0	NULL	0	a:4:{s:4:"name";s:25:"automaticdeletionreques...	1581934254	web
c	0	1	10	0	2	0	NULL	0	a:4:{s:4:"name";s:20:"privacyrequestexpiry";s:...	1581934254	web
c	0	1	10	0	2	0	NULL	0	a:4:{s:4:"name";s:33:"requirealldatesforus...	1581934254	web
c	0	1	10	0	2	0	NULL	0	a:4:{s:4:"name";s:24:"showdataretentionsum...	1581934254	web
c	0	1	10	0	2	0	NULL	0	a:4:{s:4:"name";s:17:"sitepolicyhandler";s:8:"ol...	1581934254	web

KUVA 7. "Standard log"-tietokantataulun sisältö (osa 2)

### 4.1.2 Analysoi-välilehti

Seuraavassa sovelluksen näkymässä (kuva 8) päästään itse analysointityökalun pariin. Näkymä löytyy navigointivalikosta kohdasta "Analysoi". Siirryttäessä analysointinäkymälle sovellus kysyy halutun opiskelijan etunimeä ja sukunimeä. Haluttu opiskelija on se opiskelija, jonka aktiivisuutta halutaan tarkastella kyseisellä hetkellä.



Aloitus Loki Analysoi Opiskelijat Luokkatunnukset

## Hae opiskelijan aktiivisuus nimellä

Etunimi:

Sukunimi:

Hae

KUVA 8. Opiskelijan haku nimen perusteella

Halutun opiskelijan kysely on toteutettu yksinkertaisella GET-metodilla HTML-kielellä (kuva 9). Lisäosan käyttäjän on tarkoitus syöttää edellisessä kuvassa näkyviin tekstikenttiin oppilaan nimet sekä painaa "Hae" painiketta, jolloin ohjelma (kuva 10) tulostaa, sillä nimellä löydytetyt opiskelijat käyttäjälle näkyville (kuva 11).

```

69
70     <h2>Hae opiskelijan aktiivisuus nimellä</h2>
71
72     <form action="opiskelijat.php" method="GET">
73     Etunimi: <input type="value" name="etunimi"><br>
74     Sukunimi: <input type="value" name="sukunimi"><br>
75     <input type="submit" value="Hae">
76     </form>

```

KUVA 9. Lähdekoodi opiskelijan nimen kysymiseen.

Ohjelman rivillä 86 ja 87 syötetyistä nimistä tehdään syötetyn datan mukaan muuttuvat parametrit, joiden avulla rivillä 90 muodostetaan yksinkertainen sql-kysely opiskelijan hakemiseksi tietokannasta. Tietokannasta löydetyt tiedot listataan taulukkoon riveillä 93–98 ja tulostetaan ne lisäosan käyttäjälle rivillä 99.

Oikean oppilaan kysely id:n perusteella toimii samanlaisella GET-metodilla, kun aiemmin esiintynyt opiskelijan nimen kysely. Kysely id:n perusteella on toteutettu riveillä 105–109.

```

83 if ($conn->connect_error) {
84     die("Connection failed: " . $conn->connect_error);
85 }
86 $query_enimi = $_GET['etunimi'];
87 $query_snimi = $_GET['sukunimi'];
88
89 $sql = "SELECT id, firstname, lastname FROM mdl_user
90     WHERE deleted = 0 and suspended = 0 and firstname = '$query_enimi' and lastname = '$query_snimi'";
91 $result = $conn->query($sql);
92
93 if ($result->num_rows > 0) {
94     echo "<table><tr><th>ID</th><th>Opiskelijan nimi</th></tr>";
95     // output data of each row
96     while($row = $result->fetch_assoc()) {
97         echo "<tr><td>".$row["id"]."</td><td>".$row["firstname"]." ".$row["lastname"]."</td></tr>";
98     }
99     echo "</table>";
100 } else {
101     echo "0 results";
102 }
103 $conn->close();
104 ?>
105 <h2>Valitse oikea opiskelija</h2>
106 <form action="lokitestiGrafiikka.php" method="GET">
107 Anna opiskelijan id: <input type="value" name="id">
108
109 <input type="submit" value="Näytä aktiivisuus">

```

KUVA 10. Haetun opiskelijan tulostaminen ja valinta id:n perusteella

Opiskelijan nimet tulee kirjoittaa oikein niille varattuihin tekstinsyöttölaatikoihin, jotta hakeminen onnistuu. Sovelluksessa on otettu huomioon, että Moodle-järjestelmissä voi olla monta saman nimistä henkilöä. Tästä syystä sovellus listaa kaikki kyseisellä nimellä löytyvät henkilöt ensin näkyville, josta voidaan valita oikea henkilö id-numeron perusteella. Oikean opiskelijan löydyttyä listasta, haetaan hänen id-numeroaan hyväksikäyttäen kyseisen opiskelijan lokitiedot kursseilla (kuva 12).



ID	Opiskelijan nimi
10003	Matti Meikkinen

**Valitse oikea opiskelija**

Anna opiskelijan id:

KUVA 11. Opiskelijan valinta id:n perusteella

ID	Tapahtuma	Käyttäjän ID	Aikaleima
51415	\mod_assign\event\submission_status_viewed	10003	14-05-2020
23936	\mod_assign\event\submission_status_viewed	10003	03-03-2020
23935	\mod_assign\event\assessable_submitted	10003	03-03-2020
23932	\core\event\course_module_completion_updated	10003	03-03-2020
23931	\assignsubmission_file\event\submission_created	10003	03-03-2020
23930	\assignsubmission_file\event\assessable_uploaded	10003	03-03-2020
23929	\assignsubmission_onlinetext\event\submission_created	10003	03-03-2020
23928	\assignsubmission_onlinetext\event\assessable_uploaded	10003	03-03-2020
23927	\mod_assign\event\submission_form_viewed	10003	03-03-2020
23926	\mod_assign\event\submission_status_viewed	10003	03-03-2020

KUVA 12. Haetun opiskelijan lokidataa

Jos järjestelmästä löytyy haetulla nimellä enemmän kuin yksi opiskelija (kuva 13), pitää selvittää halutun opiskelijan id-numero toisella tavalla. Aiemmin mainitusta Moodlen tietokantataulusta "standard log" ei ole mahdollista saada käyttäjistä muuta tietoa kuin käyttäjän id. Sen sijaan opiskelijan nimen avulla voidaan Moodlen tietokantataulusta "user" löytää muita tunnistuskeinoja kuten opiskelijoiden toisia nimiä tai sähköpostiosoitteita. Tässä työssä on oletettu, että samannimiä opiskelijoita ei ole. Mahdollisesti jatkokehitysmielessä "user" -taulusta olisi hyvä tuoda enemmän tunnistettavia tietoja näkyville kuin vain nimi ja id.



Aloitus	Loki	Analysoi	Opiskelijat	Luokkatunnukset
ID	Opiskelijan nimi			
13	Lena Becker			
133	Lena Becker			
265	Lena Becker			
672	Lena Becker			
1049	Lena Becker			

KUVA 13. Monta samannimistä käyttäjää eri id-numerolla

#### 4.1.3 Luokkatunnukset-välilehti

Luokkatunnukset-välilehti (kuva 14) on tehty tällä hetkellä vain jatkokehitysmielessä tähän sovellukseen. Ajatuksena oli hakea luokkatunnuksien perusteella ryhmittäin opiskelijat Moodle-järjestelmästä, mutta jälkeinpäin havaittiin, että tämä vaatisi paljon ulkopuolisen tukea. Tukea tarvitaan Peppi-palvelusta luokkatunnuksilla opiskelijaryhmien hakemiseen. Peppi-palvelun data ei ole avointa kuten TAMK:n avoimen datan palvelu, joten hakeminen Pepistä hylättiin tältä erää.

Aloitus		
Loki		
Analysoi		
Opiskelijat		
Luokkatunnukset		
<b>Luokkatunnukset</b>		
Tunnus	Nimi	Koulutusohjelman johtaja
16i260	Tieto- ja viestintäteknikka	Opettaja
15i260	Tieto- ja viestintäteknikka	Opettaja
<pre>{ "studentGroups" }bool(true)</pre>		

KUVA 14. Esimerkki-näkymä luokkatunnuksista

Alla olevalla ohjelmalla (kuva 15) voidaan hakea Tampereen ammattikorkeakoulun avoimen datan palvelusta luokkatunnuksien listaa. Koodin alussa ohjelmalle kerrotaan osoite, josta luokkatunnukset haetaan, API-avain avoimen datan palveluun ja haettava aineisto.

Riviltä 93 alkaen aloitetaan yhteyden luominen palveluun. Yhteyden luominen alkaa yhdistämällä palvelun verkko-osoitteeseen. Yhdistämisen jälkeen tunnistautaan palveluun rivillä 94 ja määritetään haluttu kysely rivillä 95. Seuraavalla rivillä otetaan tulokset vastaan ja sen jälkeen suljetaan yhteys.

```
87 <?php
88 $url = 'https://opendata.tamk.fi/ri/realization';
89 $apiKey='[REDACTED]';
90 $query= "{
91     \"studentGroups\"
92 }";
93 $session = curl_init($url); // set up session
94 curl_setopt($session, CURLOPT_USERPWD, $apiKey); // authentication
95 curl_setopt($session, CURLOPT_POSTFIELDS, $query); // add query entries
96 $response = curl_exec($session); // execute and get response
97 curl_close($session); // close session
98 var_dump($response);
99 ?>
100
```

KUVA 15. Esimerkki PHP-koodista, jolla opiskelijaryhmiä voidaan hakea TAMK:n avoimesta datasta

## **5 LOPPUTULOKSET JA JATKOKEHITYS**

### **5.1 Lopputulokset**

Tässä työssä saatiin valmiiksi Moodle-lisäosaan käyttöliittymä, joka sisältää yhden henkilön lokitietojen tarkastelemisen, esimerkin luokkatunnusten hakemisesta avoimen datan palvelusta ja kaikkien Moodle-käyttäjien lokitietojen tarkastelun. Kaikissa lokitietojen tarkastelussa on käytetty suodatinta, joka karsii kaikista lokitiedoista näkyville vain kurssikohtaiset lokitiedot.

#### **5.1.1 Lisäosan toimintaympäristö**

Moodle-lisäosan toimintaympäristön toteutus onnistui suunnitelmien mukaan kyseisessä ympäristössä. Tarkoituksena oli päästä testaamaan sovellusta Tampereen ammattikorkeakoulun Tabula-ympäristössä, mutta useiden syiden kuten virusepidemian vuoksi sitä ei päästy suorittamaan.

#### **5.1.2 Lisäosan kehitysympäristöt ja työkalut**

Lisäosan toteuttamisessa käytetyt kehitysympäristöt ja työkalut soveltuivat mallikkaasti työn tekemiseen siltä osin, kun työtä saatiin valmiiksi. Sovelluksen toteuttamiseen voidaan käyttää muitakin ympäristöjä ja työkaluja yhtä onnistuneesti, mutta kyseisten ympäristöjen ja työkalujen valintaa ei haluttu enää vaihtaa työn aloituksen jälkeen.

Tampereen ammattikorkeakoulun avoimen datan palvelua ja Peppi-palvelua ei työssä pystytty hyödyntämään, koska esteeksi muodostui Peppi-palvelun rajapintojen vaikea käyttäminen opiskelijoille. Avoimen datan tarjoamaa luokkatunnusten listaa ei pystytty hyödyntämään, koska Tabulassa ei ole määritelty erikseen luokkatunnuksia, vaan ne tulevat Pepistä.

### 5.1.3 Lisäosan toiminta

Toteutuneen työn toiminnallisuuksista saatiin valmiiksi helppokäyttöinen ja yksinkertainen käyttöliittymä, jossa päästään tarkastelemaan kaikkien käyttäjien lokitietoja ja yhden käyttäjän lokitietoja. Lisäksi käyttöliittymässä on alustava näkymä luokkatunnuksista, joiden tarkoitus olisi tulla avoimen datan palvelusta.

Yhden käyttäjän lokitietojen hakeminen tapahtuu käyttäjän nimen avulla. Eli sovelluksen käyttäjä syöttää halutun Moodle-käyttäjän nimen tekstikenttään, löytää sen avulla Moodle-käyttäjän id-numeron ja hakee sen avulla halutun käyttäjän lokitiedot.

## 5.2 Jatkokehitys

Tehdyn työn jatkokehittäminen on välttämätöntä, sillä tähän mennessä tehty sovellus ei anna analyyttisessä mielessä mitään havainnollistavaa tietoa käyttäjän aktiivisuudesta. Käyttäjän aktiivisuus voidaan nähdä, kun käyttäjä on luonut lokimerkintöjä viime viikkoina. Lokimerkinnät näkyvät sovelluksessa listana, josta voidaan jossain määrin todeta käyttäjän aktiivisuus viime viikkoina.

Ensimmäinen ja tärkein jatkokehityskohde on graafisen kuvaajan lisääminen lokitietojen tarkastelemiseksi. Graafisesta kuvaajasta nähdään helposti ilman lokitieto-listauksia, onko käyttäjä aktiivinen vai ei. Lisäosan helppokäyttöisyyden takaamiseksi luokkatunnusten avulla opiskelijoiden ryhmittely ja lisääminen graafiseen kuvaajaan tulisi saada valmiiksi.

## 6 POHDINTA

Tämän työn kehittäminen on ollut mielenkiintoista, mutta samalla myös erittäin työlästä ja aikaa vievää. Työ olisi saanut hienon päätöksen, jos työtä olisi päästy kokeilemaan yhdessä graafisen kuvaajan ja Tampereen ammattikorkeakoulun Tabula-järjestelmän kanssa, mutta tällä kertaa se oli mahdotonta.

Kehitysmielessä Moodlesta tekee haasteellisen sen vaikeakäyttöinen tietokanta. Tietokanta koostuu liian monesta eri taulusta, joiden halutunlainen käyttäminen vaatii paljon toiminnallisuuksia kehiteltävältä sovellukselta.

Tässä työssä suunniteltu sovelluskokonaisuus antaa koulunkäyntiaktiivisuuden tutkimiseen helpotusta virtuaalisella puolella. Sovelluksen toiminta parhaassa tapauksessa ei tarvitsisi ollenkaan henkilöä tutkimaan aktiivisuuksia. Tässä tapauksessa sovelluksen pitäisi huomata oppilaan nolleen pudonnut aktiivisuus ja ilmoittaa siitä joko suoraan oppilaalle esimerkiksi sähköpostilla tai ilmoittaa tapauksesta opettajalle.

Sovelluksen toiminnan luotettavuudesta huolimatta pitää kuitenkin muistaa, että sovellus on suuntaa antava ja jokainen tapaus on käsiteltävä tapauskohtaisesti. Pelkästään sovelluksen avulla ei voida tehdä johtopäätöstä opiskelijan aktiivisuudesta opiskeluun.

Sovelluksella on tarkoitus ennakoida, ettei kukaan putoaisi opiskeluvauhdista. Sovellusta voidaan käyttää myös tilanteen selvittämiseen, kun opiskelija ei enää halua opiskella, mutta ei ole ilmoittanut siitä henkilökunnalle.

Sovelluksessa käytettäviä tietoja ei pääse näkemään muut kuin sovelluksen käyttäjät, jotka ovat koulun henkilökuntaa. Sovelluksen käyttöönotosta tulee ilmoittaa opiskelijoille etukäteen.

## LÄHTEET

Moodle. 2019. Ominaisuudet. Luettu 20.3.2020.

<https://docs.moodle.org/38/en/Features>

MySQL. 2020. Workbench. Luettu 24.3.2020. <https://www.mysql.com/products/workbench/>

<https://www.mysql.com/products/workbench/>

Emilia Tietäväinen. 2018. Mikä on Peppi? Luettu 13.4.2020.

<https://wiki.eduuni.fi/pages/viewpage.action?pageId=75748480>

TAMK Tabula. 2020. Opettajan ohjeet. Luettu 10.4.2020. <https://tabula.tamk.fi/mod/book/tool/print/index.php?id=1088590>

<https://tabula.tamk.fi/mod/book/tool/print/index.php?id=1088590>

TAMK avoin data. Rajapinnat. Luettu 20.3.2020. <http://avoindata.tamk.fi/rajapinnat/>

<http://avoindata.tamk.fi/rajapinnat/>

Keski-Suomen Verkkomedia. 2009. PHP-perusteet. Luettu 20.3.2020.

<http://www.php-perusteet.com/>

MVnet. 2006. HTML-kieli. Luettu 15.4.2020. [http://www.mvnet.fi/index.php?osio=Kotisivun\\_teko&sivu=HTML-kieli](http://www.mvnet.fi/index.php?osio=Kotisivun_teko&sivu=HTML-kieli)

[http://www.mvnet.fi/index.php?osio=Kotisivun\\_teko&sivu=HTML-kieli](http://www.mvnet.fi/index.php?osio=Kotisivun_teko&sivu=HTML-kieli)

Don Ho. 2005. Notepad++. Luettu 21.3.2020. <https://notepad-plus-plus.org/>

Moodle. 2017. Analytics plugins. Luettu 8.4.2020.

[https://docs.moodle.org/35/en/Analytics\\_plugins](https://docs.moodle.org/35/en/Analytics_plugins)

## LIITE

Liite 1. Excel-taulukko: Viiden Moodle-käyttäjän aktiivisuudet kolmen viikon liukuvalla keskiarvolla

