

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра інформатики

Курсова робота

освітній ступінь – бакалавр

на тему: **«СТВОРЕННЯ ПРОГРАМНИХ ЗАСОБІВ ПІДТРИМКИ
КОЛАБОРАТИВНОГО ДИСТАНЦІЙНОГО НАВЧАННЯ»**

Виконав: студент 3-го року
навчання,

Освітньої програми «Інженерія
програмного забезпечення», 121

Шевченко Михайло Григорович

Керівник Бублик В. В.

кандидат фіз.-мат. наук, доцент

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1: МОЖЛИВОСТІ КОЛАБОРАТИВНОГО НАВЧАННЯ В СИСТЕМІ MOODLE. ОЦІНЮВАННЯ ФОРУМУ. НЕДОЛІКИ МОДУЛЯ ВЗАЄМНОГО РЕЦЕНЗУВАННЯ	6
1.1 Опис можливостей системи Moodle	6
1.2 Опис модуля форумів	6
1.3 Огляд можливостей оцінювання форуму	7
1.4 Постановка задачі про оцінювання форуму	9
1.5 Опис модулю взаємного рецензування і його недоліків.....	10
1.6 Постановка задачі про вдосконалення модулю рецензування.....	10
РОЗДІЛ 2: ПРОГРАМНІ МЕТОДИ ТА СПОСОБИ РОЗРОБКИ ПРОГРАМНИХ ЗАСОБІВ MOODLE	11
2.1 Початок роботи з Moodle. Інсталяція.....	11
2.2 Використані види планів Moodle.....	11
2.2.1 Activity plugin.....	12
2.2.2 Local plugin.....	12
2.3 Moodle API	13
2.3.1 Data manipulation API та Data Definition Api.....	13
2.3.2 Navigation API.....	14
2.3.3 File API	15
2.3.4 Callbacks API.....	15
2.3.5 Templates	16
2.3.6 WebServices.....	17
2.3.7 Moodle AJAX	17
РОЗДІЛ 3: ОПИС РОЗРОБКИ ЗАПРОПОНОВАНИХ ЕЛЕМЕНТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	19
3.1.1 Плагін для оцінювання повідомлень на форумі.....	19
3.1.1 Способи реалізації плагіну.....	19
3.1.2 Створення бази даних та веб-сервісу.....	20
3.1.3 Додавання вкладки в меню навігації форуму	22
3.1.4 Створення сторінки перегляду дискусій	24
3.1.5 Створення сторінки перегляду повідомлень дискусії.....	26
3.2 Виправлення недоліків модуля взаємного рецензування.....	30
3.2.1 Створення відображення групових робіт студентів.....	30

3.2.2 Створення відображення всіх рецензій на групову роботу	32
3.2.3 Створення оберненого копіювання рецензентів.....	34
3.2.4 Інші недоліки	36
ВИСНОВКИ.....	38
ВИКОРИСТАНІ ДЖЕРЕЛА	39

ВСТУП

Електронне навчання є гарним способом взаємодії студентів з викладачами. Воно може бути або дистанційним або бути чудово поєднаним з офлайн навчанням. В роботі [1] розповідається про шлях дистанційного навчання від поштових листів до сучасного електронного навчання. Автори розглядають найрозповсюдженіші LMS (Learning Management System). Звернено увагу на досвід роботи з системою Moodle на факультеті інформатики і розглядаються можливі способи її покращення.

Одним із таких способів є колаборативне навчання. Це доволі складний вид навчання, що об'єднує студентів спільною діяльністю. Воно може включати в себе наукові семінари, роботу в групах або взаємне рецензування. Хоча Moodle підтримує засоби розбиття студентів на групи, але не реалізовує можливості колаборативного навчання в повній мірі. Завдяки відкритому коду і модульності системи є можливість доповнити її новими можливостями.

Науковий семінар – це вид діяльності на якому кожен студент представляє доповідь, а студенти зазначені викладачем роблять на неї рецензію. Кожен студент отримує оцінку за доповідь і за кожен власну рецензію. Найкращим стандартним модулем системи Moodle, який підходить для такого виду діяльності є модуль форумів. Завдяки форуму семінар стає доступним всім зацікавленим і кожен може взяти участь в обговоренні. Проблемою у використанні форуму для наукового семінару є не можливість оцінювати окремо доповідь і рецензії.

Також на Факультеті Інформатики НаУКМА [2] був створений модуль взаємного рецензування для забезпечення колаборативного навчання. На жаль, він не підтримує навчання в групах в повній мірі, що обмежує його функціонал.

Виходячи з описаного вище, *за мету даної роботи* були поставлені завдання розроблення, доповнення та вдосконалення існуючих модулів в системі Moodle.

Перше завдання полягало в тому, щоб розробити новий плагін оцінювання форуму. Він не повинен змінювати стандартний код форуму, а лише доповнювати його новими можливостями завдяки модульності системи Moodle. Також за допомогою нього повинна бути можливість перегляду оцінки на повідомлення форуму.

Друге завдання полягало в тому, щоб вирішити існуючі проблеми групового проходження модуля взаємного рецензування. Також зробити його більш зручним у використанні.

Мета роботи зумовила наступне **наукове завдання**:

1. Ознайомитись із системою Moodle та її можливостями для виконання наукових семінарів та проведення колаборативного навчання у ній.
2. Виконати аналіз наявних проблем існуючого оцінювання форуму та модуля рецензування з перспективи викладача і студента.
3. Охарактеризувати можливості та способи розробки плагінів та покращення існуючих модулів у системі Moodle.
4. Розробити новий функціонал та покращити існуючий за допомогою програмних засобів.
5. Сформулювати висновки і надати рекомендації для подальшого покращення створених модулів.

Робота складається з трьох розділів.

У першому розділі опрацьована інформація про систему Moodle і її можливості колаборативного навчання. Проаналізовано стандартні можливості оцінювання модулю форумів, виявлено їх недоліки і запропонований варіант їх вирішення. Також в ньому розглядаються проблеми модуля рецензування для роботи в групах.

В другому розділі проаналізовані можливості розробки плагінів для системи Moodle, які будуть використовуватися в реалізації. Також вивчено типи модулів Moodle.

Третій розділ присвячений покроковому опису розробки запропонованих елементів програмного забезпечення.

РОЗДІЛ 1: МОЖЛИВОСТІ КОЛАБОРАТИВНОГО НАВЧАННЯ В СИСТЕМІ MOODLE. ОЦІНЮВАННЯ ФОРУМУ. НЕДОЛІКИ МОДУЛЯ ВЗАЄМНОГО РЕЦЕНЗУВАННЯ

1.1 Опис можливостей системи Moodle

Система Moodle [3] – це безкоштовна навчальна платформа для дистанційного навчання. Вона створена для ефективної взаємодії між викладачами та студентами. Система надає різні можливості такі як: створення курсів, форумів, завдань, оцінювання, тестування і багато іншого.

Moodle – це акронім від Modular Object-Oriented Dynamic Learning Environment, що означає модульне об'єктно-орієнтоване динамічне навчальне середовище. Згідно цього терміну Moodle підтримує модульність, динамічність і об'єктно-орієнтованість. Відповідно до цього і відкритого коду самої системи, розробники можуть створювати свої власні модулі та плагіни для розширення можливостей Moodle.

Для забезпечення колаборативного навчання система пропонує такі можливості:

1. Розділення студентів на групи.
2. Розділення груп на групування, для кращого керування навчальним процесом.
3. Модулі активності, де студенти можуть здавати роботи в групах, а не самотійно. Наприклад, модуль завдань.
4. Форум для обговорення та обміну думками.

1.2 Опис модуля форумів

Форум (forum) [4] – це модуль активності на якому студенти і викладачі можуть обмінюватися ідеями і матеріалами. Форум може бути використаним для інформування студентів, питання-відповідь між викладачами та студентами або в навчальних цілях. Головною перевагою форуму є те, що всі матеріали на ньому доступні кожному, на відмінну від інших стандартних модулів активності, де роботу студента бачить лише викладач.

Для того, щоб створити повідомлення на форумі треба спочатку створити дискусію, яка в свою чергу і є першим повідомленням цієї дискусії. В контексті наукового семінару це – доповідь студента. Інші користувачі можуть відповідати на це повідомлення. Це і є рецензії.

Так як доповідь і рецензії повинні бути оціненими, то форум повинен надавати таку можливість викладачам.

1.3 Огляд можливостей оцінювання форуму

Для оцінювання активності студентів на форумі, Moodle надає дві можливості.

Перша – це оцінювання активності студентів на форумі вцілому. Цю можливість можна ввімкнути в налаштуваннях власне самого форуму. За допомогою неї викладач може оцінювати роботу студентів на форумі. Кожен студент отримує лише одну оцінку за всі свої повідомлення на форумі. Недоліком такого оцінювання є те, що викладач не може оцінювати різні повідомлення студента щодо певного питання різними оцінками. Якщо використовувати форум як науковий семінар, то студенти не зможуть отримати окремо оцінку за доповідь і окремо за рецензію.

The image shows a Moodle configuration interface for forum grading. It features a section titled 'Whole forum grading' with a dropdown arrow. Below this, there is a 'Grade' label with a blue question mark icon. To the right, there is a rounded rectangular box containing two settings: 'Type' with a dropdown menu currently set to 'Point', and 'Maximum grade' with a text input field containing the number '100'.

Рис. 1.3.1 Налаштування для оцінювання модулю форумів вцілому

Іншим методом оцінювання на форумі є виставлення рейтингу на повідомлення студентів. Цю можливість також можна ввімкнути у налаштуваннях форуму. Цей метод дозволяє виставляти рейтинг на кожне повідомлення студента. В кінці ці результати рейтингу сумуються або знаходять його середнє значення. Хоча оцінюються окремі повідомлення студента, а не вся

його активність як в попередньому методі, в кінці він все рівно отримує лише одну оцінку. Цей метод не надає розуміння того скільки студент заробив балів за ту чи іншу активність, бо в журналі оцінок є лише фінальний рейтинг за всі повідомлення студента на форумі. Тож кожен студент отримує загальну оцінку (рейтинг) за всі повідомлення.

▼ **Ratings**

Roles with permission to rate ? Manager, Teacher, Non-editing teacher


Aggregate type ? Average of ratings ▾

Scale ?

Type ▾


Maximum grade

Рис. 1.3.2 Налаштування для для виставлення рейтингу на повідомлення

 **Topic1**
by user1 user1 - Wednesday, 15 May 2024, 5:45 PM

Message1

Average of ratings: 10 (1)

 **Re: Topic1**
by user2 user2 - Wednesday, 15 May 2024, 5:46 PM

Message2

Average of ratings: -

Рис. 1.3.3 Приклад виставлення рейтингу на повідомлення

Також до мінусів попередніх методів оцінювання форуму можна віднести те, що вони не дають змогу залишити відгук викладача на роботу студента.

1.4 Постановка задачі про оцінювання форуму

Тож ні один зі стандартних методів оцінювання форуму не відповідає в повній мірі вимогам наукового семінару. Тому було запропоновано створити додатковий плагін для оцінювання форуму, який поєднає форум і модуль завдань.

Модуль завдань (assign) [5] – це стандартний модуль, який надає можливість студентам завантажувати свої роботи на сайт. Викладачі можуть оцінювати роботи студентів, залишати відгуки з зазначенням помилок і рекомендацій, а також встановлювати терміни здачі завдання, що видно у календарі.

Поєднавши форум і модуль завдань, можна буде оцінювати кожне повідомлення на форумі окремо. Кожному повідомленню можна буде задати завдання, якому воно належить, і оцінити його за допомогою стандартного вікна оцінювання завдання. Такий метод дозволить оцінювати доповіді і рецензії окремо і кожен студент матиме різні оцінки за доповідь і рецензії. Також перевагою такого оцінювання є те, що викладач може залишати відгуки на роботи студентів та виставляти терміни здачі робіт.

Цей плагін не повинен змінювати стандартний код форуму, а лише доповнювати його, щоб код форуму міг легко оновлюватися. Також він повинен надавати можливості для перегляду оцінок за повідомленнями.

1.5 Опис модулю взаємного рецензування і його недоліків

Модуль взаємного рецензування – це плагін, який дозволяє рецензентам бачити роботи авторів й залишати на них рецензії в новій активності.

Для того, щоб використати цей модуль для організації навчання треба створити мінімум дві активності. Перша активність для здачі робіт (на ній рецензенти бачать роботи авторів), інша для здачі рецензій (на ній автори бачать рецензії на їхні роботи). Так як навчальна діяльність може складатися з багатьох етапів, таких активностей може бути більше. Для призначення авторам рецензентів використовується графічний інтерфейс. Зрозуміло, що пари автори-рецензенти повинні бути змінні місцями (оберненні) в активності здачі рецензій для того, щоб автори бачили рецензії на них.

Часто проекти, які рецензуються виконуються в командах, тому дуже важливо, щоб модуль рецензування підтримував такі можливості, а саме рецензенту бачити всі роботи команди та команді бачити рецензії на свої роботи.

1.6 Постановка задачі про вдосконалення модулю рецензування

Можна сформулювати дві основні задачі щодо покращення модулю взаємного рецензування.

Перша – якщо студенти здають роботи в групах, то рецензенти повинні бачити роботи груп студентів, яких вони рецензують. Друга задача полягає в тому, щоб якщо студенти здають свої рецензії на командні проекти, то учасники команди повинні бачити всі рецензії на їхній проект.

Також важливо зазначити, що пари автор-рецензент призначаються не автоматично, а викладачем. Так як для активності здачі рецензії вони повинні бути оберненими, це створює труднощі для викладача. Тому треба доповнити функціонал розподілення пар автор-рецензент оберненим копіюванням цих пар з цільової активності здачі робіт.

РОЗДІЛ 2: ПРОГРАМНІ МЕТОДИ ТА СПОСОБИ РОЗРОБКИ ПРОГРАМНИХ ЗАСОБІВ MOODLE

2.1 Початок роботи з Moodle. Інсталяція

Для того, щоб розпочати розробляти нові модулі або змінювати код системи Moodle, треба спочатку її інсталювати. Щоб встановити Moodle [6] на свій комп'ютер спочатку треба встановити один з веб-серверів на вибір. Це може бути Apache, Nginx, OpenLiteSpeed. Також треба мати встановлену реляційну базу даних. Moodle підтримує запити для PostgreSQL, MySQL, MariaDB. Також треба встановити PHP відповідної версії для системи Moodle

Варто зазначити, що Moodle розробляється в основному для Linux. Це не є причиною, що його не можна встановити на Windows, але через це можуть бути проблеми зі встановленням, сумісністю та коректною роботою.

Для полегшеного встановлення на Windows існує інсталяційний пакет XAMPP [7]. Він автоматично встановлює веб-сервер Apache, базу даних MySQL, PHP та Moodle відповідних версій і все налаштоване для швидкого початку роботи.

Для виконання роботи система Moodle була встановлена за офіційною інструкцією [8] на віртуальну машину з Ubuntu [9]. Було встановлено Moodle версії 4.3.2+ і PHP версії 8.2.

Для полегшення розробки також було встановлене додаткове програмне забезпечення phpMyAdmin [10] для управління базами даних та PHPStorm [11] для редагування коду.

2.2 Використані види планів Moodle

Плагін [12] (англ. plugin) – це додатковий модуль системи Moodle для розширення функціоналу системи. Він може додавати нові активності, способи задачі та оцінювання робіт або просто змінити зовнішній вигляд сайту.

Нижче проаналізовані типи плагінів з якими велася розробка в цій роботі.

2.2.1 Activity plugin

Activity plugin [13] (з англ. плагін активностей) – це основний вид плагінів для впровадження навчання на курсі. Завдяки ньому викладачі можуть створювати різні інтерактивні активності та слідкувати за прогресом студентів. Це один із найважливіших видів плагінів, бо завдяки ньому студенти можуть проходити різні завдання, а викладачі оцінювати їхню роботу.

Прикладами таких плагінів є форум (forum), модуль завдань (assign), модуль взаємного рецензування (review) та багато інших. Всі плагіни активності знаходяться в директорії /mod.

2.2.2 Local plugin

Local plugin [14] (з англ. локальний плагін) – це вид плагінів, які використовуються тоді коли інший тип не підходить для реалізації поставлених завдань. Якщо поставлені задачі можна виконати за допомогою інших видів плагінів, то треба використовувати їх.

Ось деякі варіанти використання локальних плагінів:

1. Реагування на події та дії користувачів.
2. Визначення і створення нових налаштувань адміністратора.
3. Визначення нових веб-сервісів і спеціальних зовнішніх служб.
4. Розширення навігації новими меню.
5. Створення нових додаткових баз даних.

2.3 Moodle API

Moodle надає великий список різних API [15] (англ. Application Programming Interface) для керування різними системами, розробки плагінів та інтеграції своїх власних можливостей. Розробка жодного плагіну не може обійтися без їх використання.

В цьому підрозділі буде розглянуто лише ті, які використовувалися при виконанні роботи.

2.3.1 Data manipulation API та Data Definition Api

Data manipulation API та Data Definition API – це інтерфейси для керування базами даних в системі Moodle.

Data manipulation API [16] - призначений для виконання різних запитів до бази даних. Розробники повинні використовувати виключно ці функції для того, щоб отримати або змінити вміст бази даних, оскільки ці функції забезпечують високий рівень абстракції та гарантують, що маніпуляції базою даних будуть працювати з будь-якою реляційною системою управління базою даних.

Доступ до бази даних відбувається за допомогою глобального об'єкта `$DB` який знаходиться в конфігураційному файлі `config.php`.

Data Definition API [17] – призначений для зміни структури бази даних (додавання нових колонок в таблицю, нових таблиць, редагування таблиць і так далі). Ці функції забезпечують коректне редагування таблиць у будь-якій реляційній системі управління базами даних. Використання цих функцій суворо обмежено процесами оновлення плагінів Moodle, і їх не можна використовувати в інших частинах Moodle.

Дуже рекомендовано використовувати XMLDB Editor [18] (графічний інтерфейс для редагування XML-файлів) для генерації PHP коду.

2.3.2 Navigatian API

Navigatian API [19] – це інтерфейс, який надає широкі можливості для керування навігацією курсу, активності або навігацією налаштування. Завдяки ним можна додавати нові елементи навігації або редагувати існуючі.

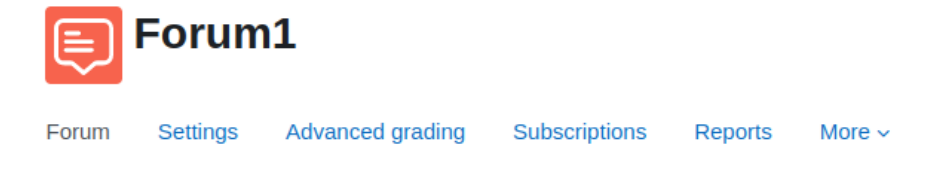


Рис. 2.3.1 Приклад навігації форуму

Дуже важливо розуміти, що таке навігація в Moodle. Навігація – дуже важливий елемент Moodle, тому що вона дозволяє впорядкувати структуру сайту Moodle. Навігація доступна через об'єкт сторінки `$PAGE`, для якого можна встановити заголовок сторінки, назву, будь-які вимоги JavaScript тощо. Навігація використовує інформацію, яку містить `$PAGE`, для створення структури навігації для сайту. Блоки навігації або параметрів є інтерпретаціями навігаційної структури, яку створює Moodle.

Navigatian Api надає широкі можливості для керування, зміни, редагування, видалення, та додавання нових елементів навігації. Доступ до цієї навігаційної структури можна отримати за допомогою трьох змінних:

1. `$PAGE->navigation` – основна структура навігації. Вона містить елементи, які дозволяють користувачеві переходити на інші доступні сторінки.
2. `$PAGE->settingsnav` – дерево навігації налаштувань містить елементи, які дозволяють розробнику редагувати налаштування, це може бути налаштування курсу, активності або інших елементів Moodle.
3. `$PAGE->navbar` – спеціальна структура для навігації по сторінках.

Доступ до основної структури навігації можна отримати через `$PAGE->navigation`. Навігація та налаштування є контекстними, оскільки вони стосуються сторінки, яку переглядає користувач. Тобто структура та вигляд навігації напряму залежить від сторінки. Для керування цими властивостями

сторінки використовуються наступні глобальні змінні, які знаходяться в об'єкті \$PAGE:

1. \$PAGE->context — це контекст Moodle, який одразу окреслює характер та вигляд сторінки, яку переглядає користувач.
2. \$PAGE->course — це курс, який переглядає користувач.
3. \$PAGE->cm — екземпляр модуля курсу.
4. \$PAGE->url — посилання сторінки, яку переглядає користувач. Цей елемент використовується для встановлення активного елемента навігації.

2.3.3 File API

File API [20] – використовується для керування, обробки файлів, які треба завантажити або зберігати в системі Moodle. Файли зберігаються в спеціальних файлових зонах, які визначаються contextid, itemid та модулем для якого він належить. Наприклад mod_assign.

Важливо зазначити, що кожний плагін повинен використовувати тільки його файлові зони. Для доступу до файлових зон інших модулів повинні використовуватися їхні власні API.

Доступ до сховища файлів відбувається за допомогою глобальної функції get_file_storage().

Найчастіше File Api використовується для створення посилань для завантаження файлів, обробкою запитів на завантаження файлів та обробкою запитів на віддачу файлів.

2.3.4 Callbacks API

Moodle Callbacks [21] (з англ. зворотні виклики) – це список функцій, які викликаються при певних подіях та діях користувача. Завдяки цьому механізму, можна викликати певний код або функції в Moodle для зміни поведінки системи Moodle або доповнення її новим функціоналом. Прикладами можуть слугувати

вітальні листи після реєстрації або оновлення даних користувача у випадку зміни ролі.

Основні події, для яких можна налаштовувати зворотні виклики, включають:

1. Реєстрація користувача: Викликається після того, як був зареєстрований новий користувач.
2. Вхід користувача: Викликається при кожному вході користувача на сайт Moodle.
3. Вихід користувача: Викликається при кожному виході користувача з системи.
4. Створення курсу або модуля: Викликається при створенні нового курсу або модуля.
5. Зміна ролі користувача: Викликається при зміні ролі користувача на курсі або сайті.
6. Розширення навігації: Викликається при створенні навігаційної структури сторінки Moodle.

Завдяки зворотнім викликам можна автоматизувати рутинні дії або створювати власні плагіни, які реагують на певні події для створення індивідуального вигляду сторінки або доповнення її новим функціоналом.

2.3.5 Templates

Moodle використовує шаблонізатор Mustache [22] для генерування HTML сторінок згідно шаблону і вхідних даних. Mustache шаблон – це звичайний текстовий файл, який може включати html, JavaScript та Mustache теги, які замінюються вхідними даними при генерування сторінки. Генерування сторінки може відбуватися як на стороні сервера за допомогою PHP або за допомогою JavaScript на стороні клієнта.

Також Moodle надає список різних допоміжних значень для доступу до основних налаштувань та функцій Moodle. Такими допоміжними тегами є `globals.config`, `str`, `quote` і багато інших.

Використання шаблонів дозволяє легко генерувати об'ємні сторінки та налаштувати їх за допомогою тем.

2.3.6 WebServices

Moodle WebServices [23] – це механізм, який дозволяє визначати свої власні сервіси для обслуговування HTTP запитів у файлах PHP.

Завдяки цим сервісам можна отримати доступ до системи Moodle через зовнішні інструменти, такі як мобільні додатки, веб-сайти, додатки сторонніх розробників тощо.

Для того, щоб визначити свій сервіс є офіційна інструкція [24]. Спочатку треба визначити сервіс у файлі `db/services.php` в директорії свого плагіну. Потім треба реалізувати сервіс в окремому PHP файлі, визначивши три методи:

1. `execute(...)` – виконує запит сервісу.
2. `execute_parameters()`- повертає параметри і їх типи, які може обробити сервіс.
3. `execute_return()`- повертає структуру відповіді запиту.

Moodle зазвичай використовує ці сервіси для інтеграції з Ajax та Moodle Mobile App.

2.3.7 Moodle AJAX

Moodle AJAX [25] – це найкращий спосіб написання взаємодій AJAX та відправки HTTP запитів на сервер. Всі потрібні інструменти для використання Moodle AJAX знаходяться у JavaScript модулі `core/ajax`. Вони напямую викликають визначені веб-сервіси за допомогою Web Service API.

Основні переваги використання Moode AJAX:

1. Жоден новий сценарій AJAX не потребує перевірки на авторизацію та вразливості безпеки.
2. Кілька запитів можна об'єднати в один http запит.
3. Суворі перевірки типів для всіх вхідних параметрів і типів повернення даних.

РОЗДІЛ 3: ОПИС РОЗРОБКИ ЗАПРОПОНОВАНИХ ЕЛЕМЕНТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1.1 Плагін для оцінювання повідомлень на форумі

3.1.1 Способи реалізації плагіну

Для того, щоб реалізувати плагін, який буде з'єднувати оцінювання повідомлень на форумі з активністю завдань є декілька способів.

Перший спосіб – це змінити та доповнити код стандартного форуму новим функціоналом. Недоліком такого методу є те, що форум є одним із основних та найбільш використовуваних модулів і зміна його коду призведе до неможливості підтримувати подальші його оновлення.

Другий спосіб – це доповнити функціонал стандартного модулю форуму за допомогою підвантаження JavaScript коду, що доповнить повідомлення на форумі можливістю оцінювання та додаванням нового налаштування до форуму. Недоліком такого методу є те, що додавання нових можливостей до існуючих плагінів за допомогою JavaScript унеможливорює використання на ці блоки спеціальних тем і це може призвести до конфлікту інтерфейсів в подальшому (після оновлення форуму).

Третій спосіб – це створити новий плагін на основі вже стандартного форуму і доповнити його новим функціоналом. Хоча здається, що це найлегший спосіб, насправді це доволі тяжкий процес, так як виникає проблема конфлікту з вже існуючим форумом. Також цей модуль не зможе отримувати подальші оновлення стандартного форуму.

Четвертий спосіб – це створити локальний плагін, який приєднає новий функціонал за допомогою створення нового блоку меню в навігації налаштування форуму. На цій вкладці буде відбуватися оцінювання повідомлень. Завдяки тому, що цей плагін не змінює код форуму і його вигляд, то не виникне ніяких конфліктів системи і форум зможе підтримувати подальші оновлення.

Для реалізації плагіну в даній роботі був вибраний четвертий варіант.

3.1.2 Створення бази даних та веб-сервісу

Для того, щоб створити локальний плагін, треба додати директорію майбутнього плагіну в папку /local. Також в цю директорію треба додати обов'язковий для всіх плагінів файл version.php.

```
defined( constant_name: 'MOODLE_INTERNAL') || die();

$plugin->version      = 2023120904;
$plugin->requires     = 2023100400;
$plugin->component    = 'local_gradingforum';
```

Рис. 3.1.2.1 Вміст файлу version.php

Зі списком обов'язкових файлів для різних плагінів можна ознайомитися на офіційному ресурсі [26].

Для початку треба створити таблицю бази даних, яка буде містити в собі ідентифікатори повідомлень та ідентифікатори активностей завдання. Вона буде використана для того, щоб ідентифікувати повідомлення, що пов'язані з активністю задачі робіт.

```
<TABLE NAME="local_gradingforum" COMMENT="This table saves information about post and assign which are linked">
  <FIELDS>
    <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" SEQUENCE="true"/>
    <FIELD NAME="postid" TYPE="int" LENGTH="10" NOTNULL="true" DEFAULT="0" SEQUENCE="false"/>
    <FIELD NAME="assignid" TYPE="int" LENGTH="10" NOTNULL="true" DEFAULT="0" SEQUENCE="false"/>
  </FIELDS>
  <KEYS>
    <KEY NAME="primary" TYPE="primary" FIELDS="id" COMMENT="The unique id for post-assign link."/>
  </KEYS>
</TABLE>
```

Рис. 3.1.2.2 Таблиця бази даних, яка пов'язує повідомлення і завдання

Також створено файл services.php для визначення веб-сервісу. Цей сервіс буде відповідати за те, щоб прив'язати якесь повідомлення до певної активності задачі робіт. Він буде повертати інформацію чи вже оцінена активність для студента, якому належить дане повідомлення, оцінку та максимальну можливу оцінку за роботу.

```

$functions = [
    'local_gradingforum_set_post_assign' => array(
        'classname' => 'local_gradingforum_external',
        'methodname' => 'set_post_assign',
        'classpath' => 'local/gradingforum/external/externallib.php',
        'description' => 'links post with assign',
        'type' => 'write',
        'ajax' => true,
        'loginrequired' => true,
        'services' => array(MOODLE_OFFICIAL_MOBILE_SERVICE)
    ),
];

```

Рис. 3.1.2.3 Вміст файлу services.php

Для написання веб-сервісу створено файл externallib.php. В ньому визначено клас local_gradingforum_external, який міститиме три методи для нашого сервісу:

1. set_post_assign_parameters – повертає поля, які приймає на вхід наш сервіс.

```

public static function set_post_assign_parameters() {
    return new external_function_parameters([
        'postid' => new external_value( type: PARAM_INT, desc: 'id of post'),
        'assignid' => new external_value( type: PARAM_INT, desc: 'id of assign'),
    ]);
}

```

Рис. 3.1.2.4 Метод set_post_assign_parameters

2. set_post_assign_returns – повертає поля, які повертає наш веб-сервіс.

```

public static function set_post_assign_returns() {
    new external_single_structure([
        'isgraded' => new external_value( type: PARAM_BOOL, desc: 'group record id'),
        'grade' => new external_value( type: PARAM_INT, desc: 'group record id'),
        'maxgrade' => new external_value( type: PARAM_INT, desc: 'id of course'),
    ]);
}

```

Рис. 3.1.2.5 Метод set_post_assign_returns

3. `set_post_assign` – метод, який обробляє запит. Завдяки запитам до бази даних, використовуючи Data manipulation API, опрацьовує запит та повертає результат.

```

public static function set_post_assign($postid, $assignid) {
    global $CFG, $DB;
    $params = self::validate_parameters(
        self::set_post_assign_parameters(),
        ['postid' => $postid, 'assignid' => $assignid]);
    $transaction = $DB->start_delegated_transaction();
    $postassign = $DB->get_record( table: 'local_gradingforum', array('postid' => $params['postid']));
    $res = new stdClass();
    $res->isgraded = false;
    $res->grade = 0;
    $res->maxgrade = 0;
    if(!$postassign) {
        $DB->insert_record( table: 'local_gradingforum', array('postid' => $params['postid'], 'assignid' => $params['assignid']));
    } else if($params['assignid'] < 0) {
        $DB->delete_records( table: 'local_gradingforum', array('postid' => $params['postid']));
        $transaction->allow_commit();
        return $res;
    }
    else {
        $DB->update_record( table: 'local_gradingforum', array('id' => $postassign->id, 'postid' => $params['postid'], 'assignid' => $params['assignid']));
    }
    if($params['assignid'] < 0) {
        $DB->delete_records( table: 'local_gradingforum', array('postid' => $params['postid']));
        $transaction->allow_commit();
        return $res;
    }
    $assign = $DB->get_record( table: 'assign', array('id' => $params['assignid']));
    $post = $DB->get_record( table: 'forum_posts', array('id' => $params['postid']));
    if($post && $assign) {
        $assign_grade = $DB->get_record( table: 'assign_grades', array('assignment' => $assign->id, 'userid' => $post->userid));
        if($assign_grade && $assign_grade->grade >= 0) {
            $res->isgraded = true;
            $res->grade = (float)$assign_grade->grade;
            $res->maxgrade = (float)$assign->grade;
        }
    }
    $transaction->allow_commit();
    return $res;
}

```

Рис. 3.1.2.6 Метод set_post_assign

3.1.3 Додавання вкладки в меню навігації форуму

Для додавання нового меню в навігацію налаштування форуму треба використати зворотній виклик. Для нашої цілі підходить callback `extend_settings_navigation`. Moodle викликає всі функції, які закінчуються на `extend_settings_navigation` під час кожного створення сторінки. Тому створюємо функцію `local_gradingforum_extend_settings_navigation` в файлі `lib.php`, яка і буде додавати нове меню. `lib.php` – це файл, який використовується як «міст» між стандартним кодом Moodle і кодами плагінів.

Спочатку перевіряємо за допомогою стандартної функції `has_capability` чи має користувач можливість переглядати оцінки студентів на курсі (перевіряємо чи це викладач).

```
if(!has_capability(capability: 'gradereport/grader:view', $context)){
    return false;
}
```

Рис. 3.1.3.1 Перевірка на доступ до плагіну

Також перевіряємо чи ця сторінка стосується модулю форум, а не якогось іншого плагіну.

```
$cm = $PAGE->cm;
if (!$cm) {
    return false;
}
if ($cm->modname != 'forum') {
    return false;
}
```

Рис. 3.1.3.2 Перевірка чи сторінка стосується форуму

За допомогою Navigation API створюємо новий блок навігації, який буде посилатися на файл `view.php`, який знаходиться в директорії нашого плагіну. Додаємо цей блок в кінець навігації.

```
if ($settingnode = $settingsnav->find('modulesettings', navigation_node::TYPE_SETTING)) {
    $str = 'Grading forum';
    $url = new moodle_url(url: '/local/gradingforum/view.php', array('id' => $PAGE->cm->id));
    $node = navigation_node::create(
        $str,
        $url,
        type: navigation_node::NODETYPE_LEAF,
        shorttext: 'gradingforum',
        key: 'gradingforum',
        new pix_icon(pix: 't/addcontact', $str)
    );
    if ($PAGE->url->compare($url, matchtype: URL_MATCH_BASE)) {
        $node->make_active();
    }
    $settingnode->add_node($node);
}
```

Рис. 3.1.3.3 Додавання нового блоку навігації

Після цього повинне з'явитися нове меню навігації у форуму з посиланням на нашу сторінку.

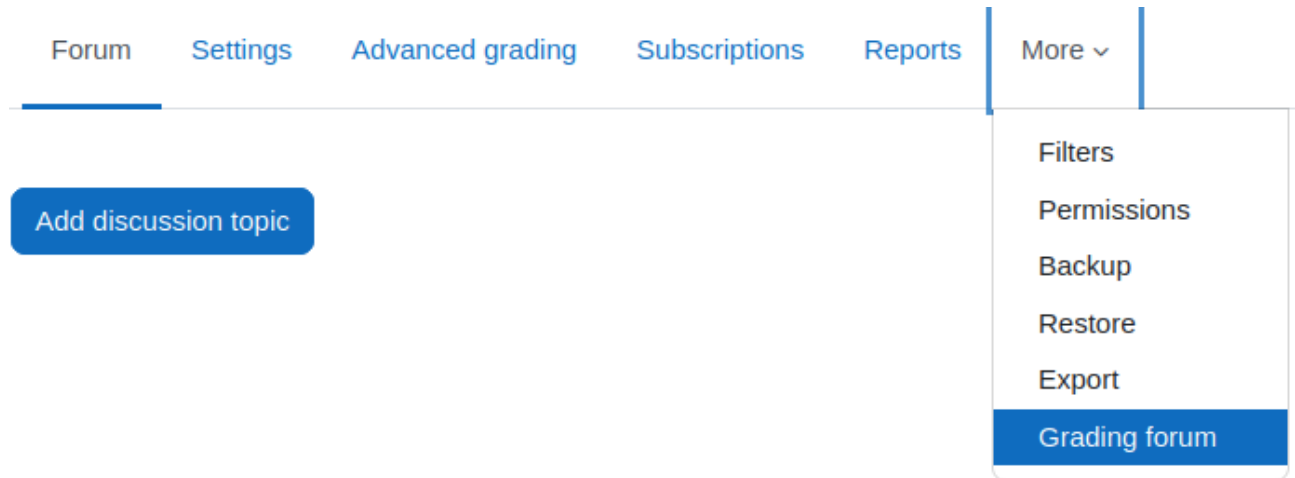


Рис. 3.1.3.4 Новий блок навігації в налаштуваннях форуму

3.1.4 Створення сторінки перегляду дискусій

На головній сторінці нашого плагіну буде список дискусій форуму з посиланнями на сторінки, де можна буде оцінювати повідомлення. Для створення сторінки дискусій за основу було взято файл `discussion_list.php` зі стандартного модуля форум. Цей клас відповідає за відображення списку дискусій. Також було взято шаблон `discussion.mustache` і відредаговано, щоб він відповідав нашим цілям за представлення списку дискусій в html форматі.

В класі `discussion_list` треба змінити посилання на відображення повідомлень за дискусіями на майбутню сторінку.

```
foreach ($exportedposts['summaries'] as &$summary) {
    $summary->discussion->urls['view'] =
        "$CFG->wwwroot/local/gradingforum/discuss.php?d={$summary->discussion->id}";
}
```

Рис. 3.1.4.1 Додавання посилань на сторінку відображення повідомлень за дискусіями

В файлі `view.php` виконуємо стандартну функцію `require_login`, щоб перевірити чи авторизований користувач а також, щоб пов'язати контекст

форуму з сторінкою. Після її виклику вигляд сторінки буде подібним до стартової сторінки стандартного форуму.

```
$cm = get_coursemodule_from_id(modulename: '', $cmid, courseid: 0, sectionnum: false, strictness: MUST_EXIST);
$course = $DB->get_record(table: 'course', array('id'=>$cm->course), fields: '*', strictness: MUST_EXIST);
require_login($course, autologinguest: false, $cm);
```


Рис. 3.1.4.2 Виклик функції `require_login`

Після цього створюємо нашу сторінку за допомогою об'єкта класу `discussion_list`.

```
echo $OUTPUT->header();
$assigns = $DB->get_records(table: 'assign', array('course'=>$course->id));
if(!$assigns) {
    echo "Add assignments to course to link them with posts and grade";
}
else {
    echo $discussionsrenderer->render(
        $USER,
        $coursemodule,
        $groupid,
        $sortorder,
        $pageno,
        $pagesize,
        displaymode: null,
        enablediscussioncreation: false);
}
echo $OUTPUT->footer();
```





Рис. 3.1.4.3 Створення сторінки

Course1 / Forum1 / Grading forum



Forum1

Forum Settings Advanced grading Subscriptions Reports More ▾

Discussion	Started by	Last post	Replies
Topic1	 user1 user1 15 May 2024	 user2 user2 15 May 2024	1
Topic2	 user2 user2 15 May 2024	 user2 user2 15 May 2024	0

3.1.5 Створення сторінки перегляду повідомлень дискусії

Для створення сторінки перегляду повідомлень дискусії, за основу було використано файл `discussion.php` і `forum_discussion.mustache` зі стандартного модуля форум. Редагуємо файл `discussion.php`, щоб він використовував шаблон `forum_discussion.mustache` з нашого плагіну. Також створено функцію, яка буде створювати посилання на попередню дискусію і наступну, щоб вони посилалися на файли нашого модулю.

```
private function get_neighbour_links_html() : string {
    $forum = $this->forum;
    $coursemodule = $forum->get_course_module_record();
    $neighbours = forum_get_discussion_neighbours($coursemodule, $this->discussionrecord, $this->forumrecord);
    $prev = $neighbours['prev'];
    $next = $neighbours['next'];
    $html = '';
    if ($prev || $next) {
        $html .= html_writer::start_tag( 'tagname: 'div', array('class' => 'discussion-nav clearfix'));
        $html .= html_writer::start_tag( 'tagname: 'ul');
        if ($prev) {
            $url = new moodle_url( url: '/local/gradingforum/discuss.php', array('d' => $prev->id));
            $html .= html_writer::start_tag( 'tagname: 'li', array('class' => 'prev-discussion'));
            $html .= html_writer::link($url, text: $this->renderer->arrow() . ' ' . format_string($prev->name),
                array('aria-label' => get_string(
                    identifier: 'prevdiscussiona',
                    component: 'mod_forum', format_string($prev->name)),
                    'class' => 'btn btn-link'));
            $html .= html_writer::end_tag( 'tagname: 'li');
        }
        if ($next) {
            $url = new moodle_url( url: '/local/gradingforum/discuss.php', array('d' => $next->id));
            $html .= html_writer::start_tag( 'tagname: 'li', array('class' => 'next-discussion'));
            $html .= html_writer::link($url, text: format_string($next->name) . ' ' . $this->renderer->arrow(),
                array('aria-label' => get_string( identifier: 'nextdiscussiona',
                    component: 'mod_forum', format_string($next->name)),
                    'class' => 'btn btn-link'));
            $html .= html_writer::end_tag( 'tagname: 'li');
        }
        $html .= html_writer::end_tag( 'tagname: 'ul');
        $html .= html_writer::end_tag( 'tagname: 'div');
    }
    return $html;
}
```

Для відображення повідомлень дискусії використовуються файли, які були створені на основі файлів з модуля форум:

1. `posts.php` – файл, який виконує попередню обробку списку повідомлень і використовує шаблонізатор `forum_discussion_posts.mustache` для їх відображення.

2. `forum_discussion_posts.mustache` – шаблонізатор, який створює html представлення списку повідомлень.
3. `forum_discussion_post.mustache` – шаблонізатор, який створює html представлення одного повідомлення.

```

{{#posts}}
  {{> local_gradingforum/forum_discussion_post }}
{{/posts}}

```

Рис. 3.1.5.1 Файл `forum_discussion_posts.mustache`

В класі `posts`, який знаходиться в файлі `posts.php`, до кожного повідомлення додаємо список всіх наявних активностей завдання для того, щоб викладач міг вибирати, якому завданню належить повідомлення. Також було додано інформацію про оцінку цього завдання для даного повідомлення для її відображення.

```

$assigns = $this->get_all_assigns_in_course($forums);
foreach ($assigns as &$assign) {
    $cm = get_coursemodule_from_instance( modulename: 'assign', $assign->id, $assign->course);
    $assign->cmid = $cm->id;
}
foreach ($exportedposts['posts'] as &$post) {
    $post->readonly = true;
    $post->assigns = $assigns;
    $post->graded = false;
    $post->grade = 0;
    $post->maxgrade = 0;
    $post->assignid = 0;
    $postassign = $DB->get_record( table: 'local_gradingforum', array('postid' => $post->id));
    if($postassign) {
        $post->assignid = $postassign->assignid;
        $assign_grade = $DB->get_record( table: 'assign_grades', array('assignment' => $postassign->assignid,
            'userid' => $post->author->id));
        if($assign_grade && $assign_grade->grade >= 0) {
            $post->isgraded = true;
            $post->grade = (float)$assign_grade->grade;
            $post->maxgrade = (float)$assign->grade;
        }
    }
}
}

```

Рис. 3.1.5.2 Додавання інформації про завдання до повідомлення

До шаблону `forum_discussion_post.mustache` додали блок, який буде відповідати за список завдань та відображення оцінки. Крім того цей блок

містить кнопку оцінювання повідомлення, яка спрямовує викладача відразу на оцінювання завдання студента, якому належить дане повідомлення.

```

<div class="d-flex">
  <form class="form-inline my-auto">
    <label for="grading-select-for-post{{id}}">Assignment</label>
    <select id="grading-select-for-post{{id}}" class="custom-select singleselect">
      <option value="-1" assignid="-1" selected>
        None
      </option>
      {{#assigns}}
        <option value="{{cmid}}" assignid="{{id}}">
          {{name}}
        </option>
      {{/assigns}}
    </select>
  </form>
  <button id="grading-button-for-post{{id}}" class="btn btn-primary mx-2">Grade</button>
  {{#isgraded}}
    <span id="grading-span-for-post{{id}}" class="my-auto">{{grade}} / {{maxgrade}}</span>
  {{/isgraded}}
  {{^isgraded}}
    <span id="grading-span-for-post{{id}}" class="my-auto">Not graded</span>
  {{/isgraded}}
</div>

```

Рис. 3.1.5.3 Шаблон для відображення списку завдань, кнопки оцінювання та оцінки

Також для того, щоб повідомлення було пов'язане з певним завдання і ці дані додавалися в базу даних додали JavaScript код, який відправляє запити веб-сервіс за допомогою Moodle AJAX.

```

grade_button.addEventListener("click", () => {
  let assignid = post_select.options[post_select.selectedIndex].getAttribute("assignid");
  if(assignid < 0) {
    return;
  }
  let promises = ajax.call([
    { methodname: 'local_gradingforum_set_post_assign', args: { postid: {{id}}, assignid: assignid } }
  ]);
  let val = post_select.value;
  let url = `{{config.wwwroot}}/mod/assign/view.php?id=${val}&action=grader&userid={{author.id}}`;
  window.location.href = url;
});

```

Рис. 3.1.5.4 Код кнопки оцінювання, яка відправляє запити на сервер

У файлі `discuss.php` за допомогою Navigation API додали елементи навігації, щоб можна було переходити на сторінку перегляду всіх дискусій.

```
$forumnode = $PAGE->navigation->find($cm->id, type: navigation_node::TYPE_ACTIVITY);
if (empty($forumnode)) {
    $forumnode = $PAGE->navbar;
} else {
    $forumnode->make_active();
}
$gradingforumurl = new moodle_url(url: '/local/gradingforum/view.php', array('id' => $cm->id));
$node = $forumnode->add(format_string(string: "Grading forum"), $gradingforumurl);
$node->make_active();
$node = $node->add(format_string($discussion->get_name()), $discussionviewurl);
```

Рис. 3.1.5.5 Код додавання навігації

Після цього відображуємо сторінку за допомогою об'єкта класу `posts`, попередньо створивши його.

The screenshot shows a Moodle forum interface. At the top, there is a breadcrumb trail: [Course1](#) / [Forum1](#) / [Grading forum](#) / [Topic1](#). Below this is the forum header with a red speech bubble icon and the text "Forum1". A navigation menu includes "Forum", "Settings", "Advanced grading", "Subscriptions", "Reports", and "More v".

The main content area shows a breadcrumb trail: [Topic2](#). Below this is a message card for "Topic1" by user1 user1 - Wednesday, 15 May 2024, 5:45 PM. The message is "Message1". Below the message, there is an "Assignment" dropdown menu set to "Assignment1" and a "Grade" button showing "68 / 100".

Below the first message card is a second message card for "Re: Topic1" by user2 user2 - Wednesday, 15 May 2024, 5:46 PM. The message is "Message2". Below the message, there is an "Assignment" dropdown menu set to "None" and a "Grade" button showing "Not graded". The dropdown menu is open, showing options: "None", "Assignment1", and "Assignment2".

Рис. 3.1.5.6 Приклад оцінювання, вибору завдання, перегляду оцінки на повідомлення

Для використання цього плагіну спочатку треба створити мінімум одне завдання на курсі. Після цього вкладка з оцінюванням форуму з'явиться на всіх

форумах і можна буде оцінювати повідомлення студентів. Для оцінювання повідомлення треба вибрати потрібне завдання з спадного списку на повідомленні і натиснути кнопку Grade (з англ. оцінити), яка відкриє стандартне меню оцінювання завдання. Після цього повідомлення зв'яжеться з вибраним завданням та буде відображати оцінку студента.

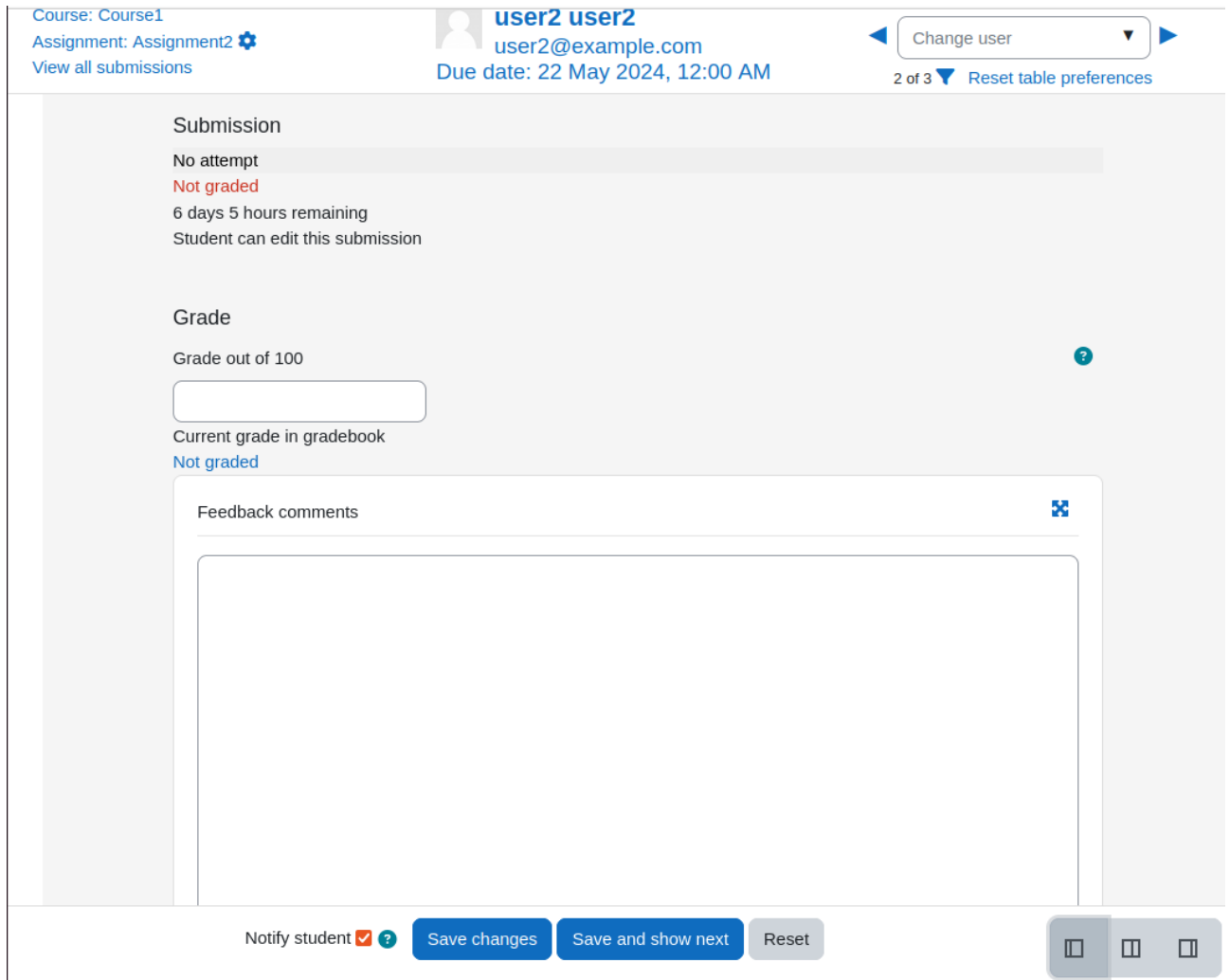


Рис. 3.1.5.7 Приклад оцінювання, вибору завдання, перегляду оцінки на повідомлення

3.2 Виправлення недоліків модуля взаємного рецензування

3.2.1 Створення відображення групових робіт студентів

Для того, щоб виправити відображення задачі робіт для груп модуля взаємного рецензування, спочатку треба знайти функцію, яка відповідає за відображення файлів, які завантажив автор. Після дослідження коду модуля

взаємного рецензування та його зневадження (англ. debugging) була знайдена функція `render_review_submission_status` у файлі `renderer.php`. Вона відповідає за відображення власне роботи студента та роботи автора, яку йому потрібно прорецензувати.

Після аналізу коду функції, а також таблиці `review_submission` бази даних (таблиця, яка зберігає дані про здані роботи) було виявлено, що для активності взаємного рецензування, що налаштована для роботи в групах, файли які завантажує група на оцінювання ідентифікується ідентифікатором групи, а не ідентифікаторами студентів, які входять в неї. Виявилось, що модуль взаємного рецензування не обробляє дану ситуацію, бо в базі даних зберігаються пари автор-рецензент, а не група-рецензент. Так як функція завжди намагається отримати список файлів за ідентифікатором автора, вона отримує пустий список і виводить, що зданих робіт немає.

Для вирішення цієї проблеми було досліджено базу даних, та складено запит до неї для отримання групи автора, якій він належить. Також треба враховувати, що групи можуть бути об'єднанні в групування.

```

if($currentreview->teamsubmissiongroupingid) {
    $sql = 'SELECT g.id, g.name
           FROM {groupings_groups} gg
           INNER JOIN {groups} g ON gg.groupid = g.id
           INNER JOIN {groups_members} gm ON g.id = gm.groupid
           WHERE gg.groupingid = :grouping_id AND gm.userid = :user_id';
    $targetgroup = $DB->get_record_sql($sql, [
        "grouping_id" => $currentreview->teamsubmissiongroupingid,
        "user_id" => $reviewerauthor->authorid
    ]);
}
else {
    $sql = 'SELECT g.id, g.name
           FROM {groups} g
           INNER JOIN {groups_members} gm ON g.id = gm.groupid
           WHERE g.courseid = :course_id AND gm.userid = :user_id';
    $targetgroup = $DB->get_record_sql($sql, [
        "course_id" => $courseid,
        "user_id" => $reviewerauthor->authorid
    ]);
}

```

Рис. 3.2.1.1 Запит для отримання групи автора, якій він належить

Після цього треба отримати файли, які завантажила група авторів за допомогою File API.

```
$fs = get_file_storage();
$files = $fs->get_area_files(
    $status->context->id,
    component: 'reviewsubmission_file',
    filearea: 'submission_files',
    $targetsubmission->id);
```

Рис. 3.2.1.2 Отримання файлів за групою автора

Потім виводимо ці файли на екран для рецензента.

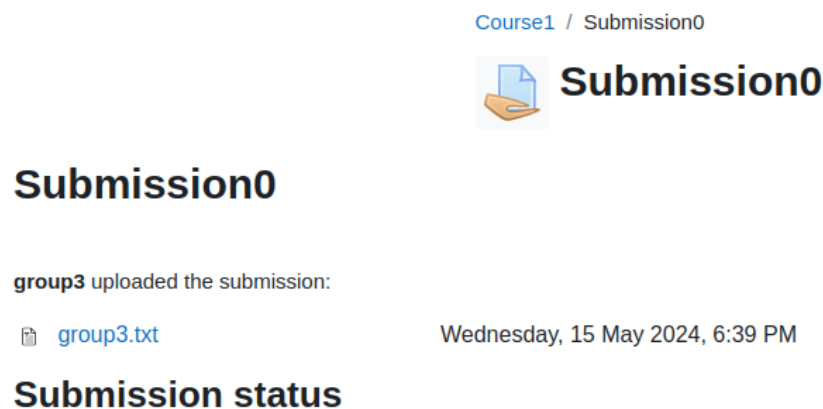


Рис. 3.2.1.3 Приклад виводу роботи групи автора для рецензента

3.2.2 Створення відображення всіх рецензій на групову роботу

Для того, щоб кожен автор бачив всі рецензії на роботу їхньої групи використаємо поле `targetreviewid` з таблиці `review` бази даних. Було перевірено і досліджено, що це поле не несе ніякої функціонального значення. Це поле буде відповідати за вказання активності групової задачі робіт, щоб ідентифікувати рецензентів роботи. Його можна вказати на графічному інтерфейсі призначення рецензентів.

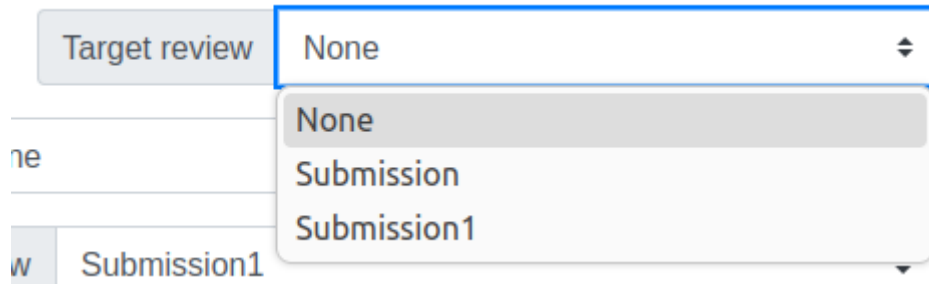


Рис. 3.2.2.1 Вибір цільової активності здачі робіт

Для того, щоб отримати всіх рецензентів для команди складемо запит за аналогією як було описано вище. В цьому запиті було використано поле `targetreviewid` для того, щоб знайти в яких групах здавали роботи рецензенти на етапі здачі робіт.

```

if($targetreview->teamsubmissiongroupingid) {
    $sql = 'SELECT g.id
            FROM {groupings_groups} gg
            INNER JOIN {groups} g ON gg.groupid = g.id
            INNER JOIN {groups_members} gm ON g.id = gm.groupid
            WHERE gg.groupingid = :grouping_id AND gm.userid = :user_id';
    $targetgroup = $DB->get_record_sql($sql, [
        "grouping_id" => $targetreview->teamsubmissiongroupingid,
        "user_id" => $USER->id
    ]);
}
else {
    $sql = 'SELECT g.id, g.name
            FROM {groups} g
            INNER JOIN {groups_members} gm ON g.id = gm.groupid
            WHERE g.courseid = :course_id AND gm.userid = :user_id';
    $targetgroup = $DB->get_record_sql($sql, [
        "course_id" => $courseid,
        "user_id" => $USER->id
    ]);
}

```

Рис. 3.2.2.2 Запиті для знаходження в яких групах здавали роботи рецензенти на етапі здачі робіт

Потім отримали всіх учасників цієї групи також за допомогою запити.

```


$sql = 'SELECT gm.userid
        FROM {groups_members} gm
        WHERE gm.groupid = :group_id';
$users = $DB->get_records_sql($sql, [
    "group_id" => $targetgroup->id
]);

```

Рис. 3.2.2.3 Запит для отримали всіх учасників групи


Виводимо на екран їхні рецензії.

Course1 / Submission1

 **Submission1**


Submission1

user2 user2 uploaded the submission:

 [user2.txt](#) Wednesday, 15 May 2024, 6:51 PM

user4 user4 has not uploaded submission yet.

user1 user1 uploaded the submission:

 [user1.txt](#) Wednesday, 15 May 2024, 6:50 PM

Submission status

Рис. 3.2.2.4 Приклад виводу всіх рецензій на групову роботу

3.2.3 Створення оберненого копіювання рецензентів

Для того, щоб використати модуль взаємного рецензування треба створити мінімум дві активності. Одна для здачі робіт, де рецензенти будуть бачити роботи авторів, і інша для здачі рецензій, де автори будуть бачити рецензії на них. Для цього треба, щоб пари автор-рецензент були оберненими на етапі здачі рецензій. Так як дуже важко правильно й без помилок перенести ці пари, вирішено додати можливість оберненого копіювання рецензентів з обраної активності для здачі робіт.

Спочатку треба знайти функцію, яка відповідає за відображення сторінки призначення рецензентів. Для цього бачимо, що в посиланні цієї сторінки є такий

параметр: `action=appointreviewers`. У файлі `locallib.php` у функції `view` при такому параметрі викликається функція `view_appoint_reviewers_page`. Це і є функція, яка відповідає за відображення сторінки для призначення рецензентів.

```
} else if ($action == 'appointreviewers') {
    $o .= $this->view_appoint_reviewers_page($args);
```

Рис. 3.2.3.1 Виклик функції `view_appoint_reviewers_page` у файлі `locallib.php`

Додаємо в неї html блок для того, щоб вибирати з якої активності треба обернено скопіювати рецензентів.

```
<div class="row mt-3">
  <div class="input-group px-3">
    <div class="input-group-prepend">
      <label class="input-group-text" for="copyReviewReverse">Copy reverse from other review</label>
    </div>
    <select class="custom-select" id="copyReviewReverse" onChange="onChangeReviewCopyReverse()">
      <option selected value="none">None</option>
    </select>
  </div>
</div>
```

Рис. 3.2.3.2 html код для відображення блоку вибору цільової активності для оберненого копіювання рецензентів

Також додаємо функцію оберненого копіювання в файл `util.js`.

```
function onChangeReviewCopyReverse():void {
  const pairs :Map<any, any> = new Map();
  const copyId = COPY_REVIEW_REVERSE.val();
  RESET_BUTTON.trigger("click");
  if (copyId === 'none') {
    RESULT_TEXT.text("Fields was cleared.");
    return;
  }
  Object.values(reviewer_authors).filter(i => i.review === copyId).map(i => pairs.set(i.reviewerid, i.authorid));
  pairs.forEach( callbackfn: (key, value):void => {
    assignAuthor(key, value);
  });
  renderColumns(getCurrentRootNode());
  RESULT_TEXT.text(`Reverse copied from ${Object.values(allReviews).find(i => i.id === copyId).name} review.`);
}
```

Рис. 3.2.3.3 Функція оберненого копіювання

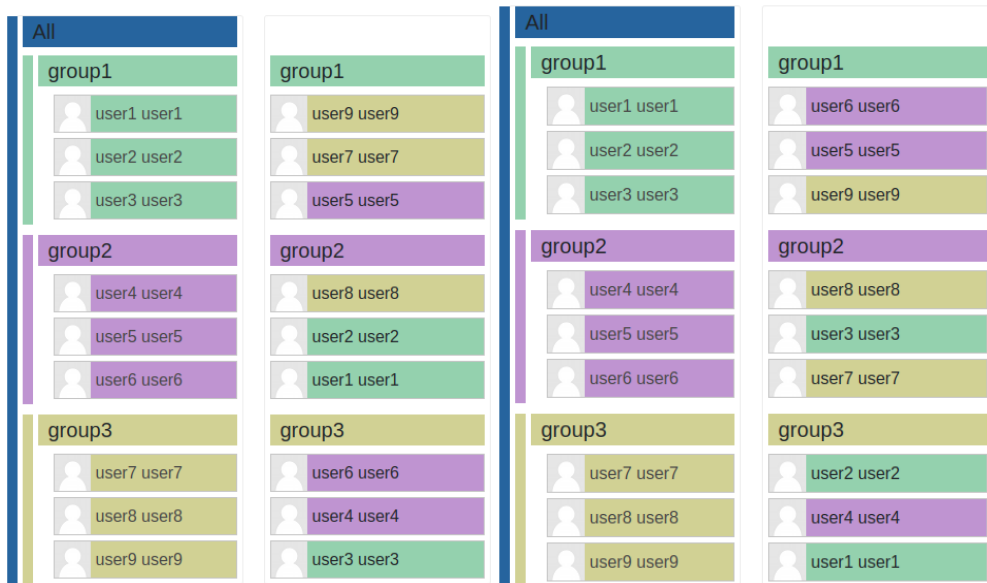


Рис. 3.2.3.4 Приклад оберненого копіювання рецензентів. Зліва активність з, якої рецензенти копіювалися. Справа активність з обернено скопійованими рецензентами

3.2.4 Інші недоліки

Під час користування модулем взаємного рецензування були виявлені суттєві недоліки, які погіршували досвід роботи з ним.

Перший недолік – це неправильне копіювання пар автор-рецензент з цільової активності, якщо в ній деякі пари пусті.

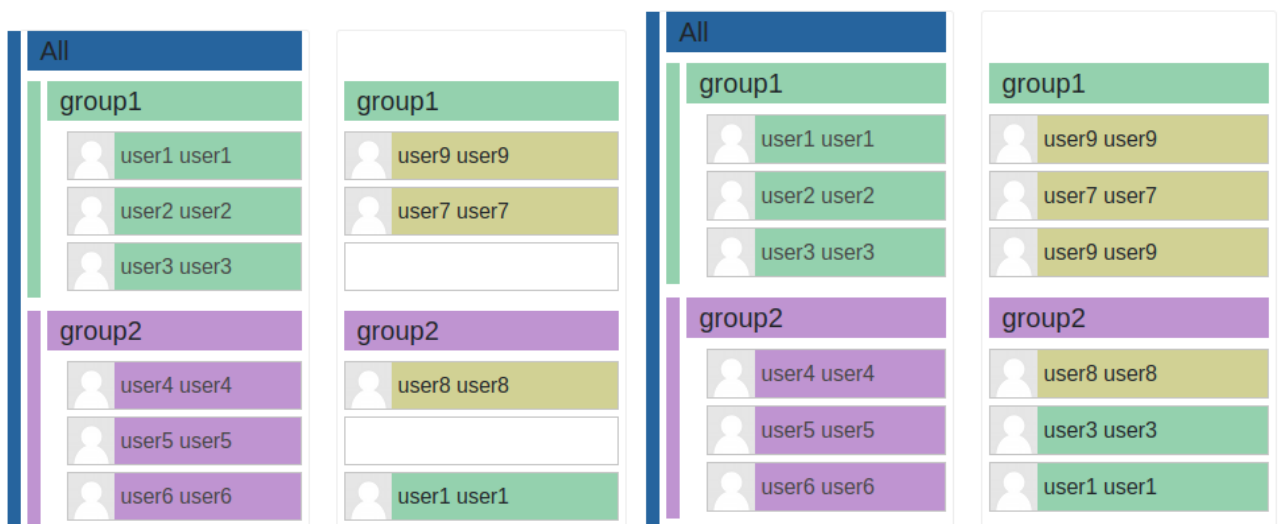


Рис. 3.2.4.1 Приклад некоректного копіювання рецензентів. Зліва активність з, якої рецензенти копіювалися. Справа активність зі скопійованими рецензентами

Для виправлення цієї помилки знаходимо функцію, яка копіює рецензентів з цільової активності у фалі util.js. Після її зневадження було виявлено, що вона не очищує попередні пари автор-рецензент, а залишає їх. Тому було зроблено, щоб вона спочатку очищувала всіх рецензентів, а потім їх копіювала.

```
function onChangeReviewCopy() : void {
  const pairs : Map<any, any> = new Map();
  const copyId : ... = COPY_REVIEW.val();
  RESET_BUTTON.trigger("click");
  if (copyId === 'none') {
    RESULT_TEXT.text("Fields was cleared.");
    return;
  }
  Object.values(reviewers_authors).filter(i => i.review === copyId).map(i => pairs.set(iReviewerid, i.authorid));
  console.log(pairs);
  pairs.forEach( callbackfn: (key, value) : void => {
    assignAuthor(value, key);
  })
  renderColumns(getCurrentRootNode());
  RESULT_TEXT.text(`Copied from ${Object.values(allReviews).find(i => i.id === copyId).name} review.`);
}
```

Рис. 3.2.4.2 Виправлений код функції для копіювання рецензентів

Також під час користування модулем взаємного рецензування, виявилось, що викладач призначає не рецензентів авторам, а авторів – рецензентам. Це викликало непорозуміння у викладачів і студентів під час використання даного модуля. Тому було додано написи над кожною колонкою для того, щоб уникнути непорозумінь в майбутньому.

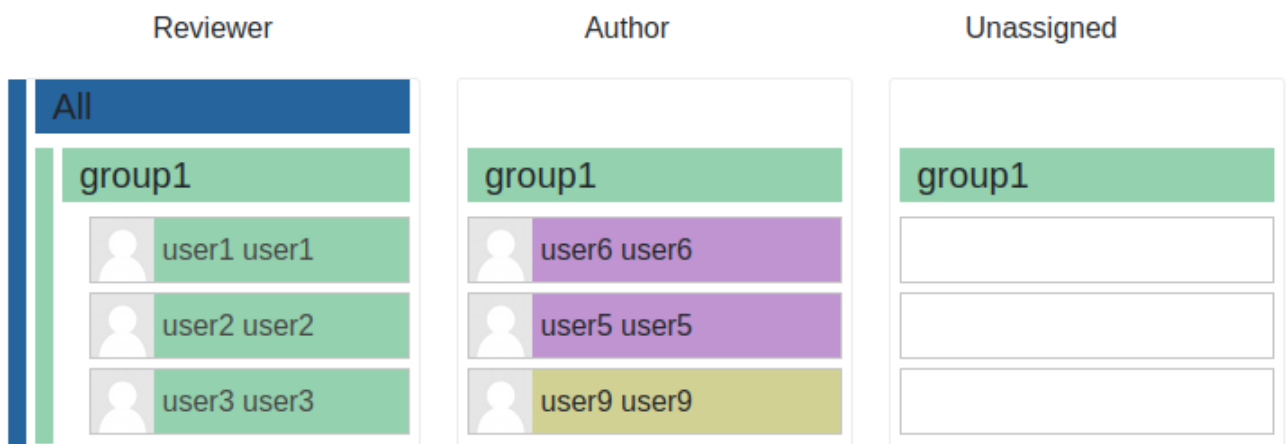


Рис. 3.2.4.3 Приклад надписів над колонками

ВИСНОВКИ

Було розглянуто методи і способи реалізації плагінів для системи Moodle.

У роботі розглянуто можливості та способи оцінювання форуму і виявленні їх недоліки. Було запропоновано та реалізовано можливість оцінювати форум, завдяки поєднанню модулю форуму та модулю завдань, не змінюючи оригінального коду Moodle. Реалізовано можливість переглядати оцінки викладачам відповідно до повідомлень на форумі.

Проаналізовано можливості колаборативного навчання за допомогою модулю взаємного рецензування і виявленні його недоліки. Запропоновано і реалізовано можливість перегляду робіт для груп і їх рецензентів. Реалізовано можливість оберненого копіювання рецензентів з інших екземплярів активності взаємного рецензування. Також були виправлені деякі помилки, які псували взаємодію з модулем рецензування.

Подальші вдосконалення передбачаються в напрямку додання можливості перегляду повідомлень при одночасному їх оцінюванні та покращення збору статистики проходження курсу за різними спеціальностями, командами та групами.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Бублик В. В. Шляхами дистанційної освіти та електронного навчання / Бублик В. В. // Наукові записки НаУКМА. Комп'ютерні науки. - 2018. - Т. 1. - С. 4-9. — URL: <https://ekmair.ukma.edu.ua/handle/123456789/14639>
2. Бублик В. В. Розвиток колаборативних навчальних середовищ / Бублик В. В., Дроздович Н. Ю. // Наукові записки НаУКМА. - 2012. - Т. 138: Комп'ютерні науки. - С. 76-79. – URL: <https://ekmair.ukma.edu.ua/handle/123456789/1919>
3. Moodle [Електронний ресурс] - <https://moodle.org>
4. Forum Activity [Електронний ресурс] - https://docs.moodle.org/404/en/Forum_activity
5. Assignment Activity [Електронний ресурс] - https://docs.moodle.org/404/en/Assignment_activity
6. Installing Moodle [Електронний ресурс] - https://docs.moodle.org/404/en/Installing_Moodle
7. XAMPP [Електронний ресурс] - https://docs.moodle.org/402/en/Complete_install_packages_for_Windows
8. Step-by-step Installation Guide for Ubuntu [Електронний ресурс] - https://docs.moodle.org/404/en/Step-by-step_Installation_Guide_for_Ubuntu
9. Ubuntu [Електронний ресурс] - <https://ubuntu.com>
- 10.phpMyAdmin [Електронний ресурс] - <https://www.phpmyadmin.net>
- 11.PhpStorm [Електронний ресурс] - <https://www.jetbrains.com/phpstorm/>
- 12.Moodle Plugin Types [Електронний ресурс] - <https://moodledev.io/docs/4.4/apis/pluginatypes>
- 13.Activity modules [Електронний ресурс] - <https://moodledev.io/docs/4.4/apis/pluginatypes/mod>
- 14.Local plugins [Електронний ресурс] - <https://moodledev.io/docs/4.4/apis/pluginatypes/local>
- 15.Moodle APIs [Електронний ресурс] - <https://moodledev.io/docs/4.4/apis>

- 16.Data manipulation API [Электронный ресурс] -
<https://moodledev.io/docs/4.4/apis/core/dml>
- 17.Data definition API [Электронный ресурс] -
<https://moodledev.io/docs/4.4/apis/core/dml/ddl>
- 18.XMLDB Editor [Электронный ресурс] -
<https://moodledev.io/general/development/tools/xmlldb>
- 19.Navigation API [Электронный ресурс] -
<https://moodledev.io/docs/4.4/apis/core/navigation>
- 20.File API [Электронный ресурс] -
<https://moodledev.io/docs/4.4/apis/subsystems/files>
- 21.Moodle Callbacks [Электронный ресурс] -
<https://docs.moodle.org/dev/Callbacks>
- 22.Moodle Templates [Электронный ресурс] -
<https://moodledev.io/docs/4.4/guides/templates>
- 23.Moodle External Services [Электронный ресурс] -
<https://moodledev.io/docs/4.4/apis/subsystems/external>
- 24.Writing a new service [Электронный ресурс] -
<https://moodledev.io/docs/4.4/apis/subsystems/external/writing-a-service>
- 25.Moodle AJAX [Электронный ресурс] -
<https://moodledev.io/docs/4.4/guides/javascript/ajax>
- 26.Moodle Common Files [Электронный ресурс] -
<https://moodledev.io/docs/4.4/apis/commonfiles>