

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

## Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБЛЕННЯ ПЛАГІНІВ ДЛЯ ЦИФРОВОЇ

СИСТЕМИ НАВЧАННЯ НА БАЗІ MOODLE

DEVELOPMENT OF THE PLUGINS FOR A MOODLE-

BASED DIGITAL EDUCATION SYSTEM

Виконав: студент(ка) 4 курсу, групи КНТ-128

Спеціальності 121 Інженерія програмного  
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Синицин С.С.

(прізвище та ініціали)

Керівник Табунщик Г. В.

(прізвище та ініціали)

Рецензент Шитікова О.В.

(прізвище та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Національний університет «Запорізька політехніка»**  
 (повне найменування закладу вищої освіти)

Інститут, факультет ФКНТ  
 Кафедра програмних засобів  
 Ступінь вищої освіти бакалавр  
 Спеціальність 121 Інженерія програмного забезпечення  
 (код і найменування)  
 Освітня програма (спеціалізація) Інженерія програмного забезпечення  
 (назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ПЗ, д.т.н, проф.  
С.О. Субботін  
 “      ”        20   року

**З А В Д А Н Н Я**

**НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)**

Синицина Семена Сергійовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розроблення плагінів для цифрової системи навчання на базі Moodle Development of the Plugins for a Moodle-Based Digital Education System

керівник проєкту (роботи) Табунщик Галина Володимирівна, к.т.н., професор,  
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “27” квітня 2022 року № 102

2. Строк подання студентом проєкту (роботи) 09 червня 2022 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Проекування програмного забезпечення. 3. Основні рішення по реалізації компонентів системи. 4. Опис програмного додатку. 5. Керівництво програміста. 6. Експлуатація оператором програми. 8. Додатки.  
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
Слайди презентації

## 6. Консультанти розділів проєкту (роботи)

| Розділ              | Прізвище, ініціали та посада консультанта | Підпис, дата   |                           |
|---------------------|-------------------------------------------|----------------|---------------------------|
|                     |                                           | завдання видав | прийняв виконане завдання |
| 1-6 Основна частина | Табунщик Г.В., професор                   |                |                           |
| Нормоконтроль       | Белова А.В., асистент                     |                |                           |
|                     |                                           |                |                           |
|                     |                                           |                |                           |

7. Дата видачі завдання “25”квітня2022року.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проєкту (роботи)                                  | Строк виконання етапів проєкту (роботи) | Примітка     |
|-------|---------------------------------------------------------------------------|-----------------------------------------|--------------|
| 1     | Постановка завдання роботи.                                               | 1 тиждень                               | Завдання, ТЗ |
| 2     | Аналіз предметної області.                                                | 2 тиждень                               | Розділ 1     |
| 3     | Проектування програмного забезпечення                                     | 3тиждень                                | Розділ 2     |
| 3     | Розробка та удосконалення методів, моделей й алгоритмів вирішення задачі. | 4тиждень                                | Розділ 3     |
| 4     | Розробка архітектури програми.                                            | 5 тиждень                               | Розділ 4     |
| 5     | Розробка програми.                                                        | 6 тиждень                               | Розділ 5     |
| 6     | Тестування та експериментальне дослідження програмного забезпечення.      | 7 тиждень                               | Розділ 6     |
| 7     | Оформлення пояснювальної записки та документів до неї.                    | 8-9 тижні                               | Додатки      |
| 8     | Нормоконтроль та рецензування.                                            | 10 тиждень                              |              |
| 9     | Захист роботи.                                                            | 11 тиждень                              |              |

Студент(ка)

Синицин С.С.  
( підпис ) (прізвище та ініціали)

Керівник проєкту (роботи)

Табунщик Г.В.  
( підпис ) (прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:  
141 с., 11 табл., 56 рис., 2 дод., 37 джерел.

ПЛАГІН, СИСТЕМА ДИСТАНЦІЙНОГО НАВЧАННЯ, LMS,  
РЕЦЕНЗУВАННЯ, MOODLE, PHP.

Об'єкт дослідження – способи розробки плагінів для LMS Moodle.

Предмет дослідження – системи дистанційного навчання, плагіни для Moodle.

Мета роботи – дослідження процесу створення плагіну на LMS Moodle, та збільшення якості навчання за допомогою дистанційного навчання процесу через створення плагіну для рецензування курсів.

Матеріали, методи та технічні засоби – мова програмування php, запити до бази даних на мові sql, відображення результатів за допомогою html та scss.

Результати. Досліджено розробку плагінів для Moodle на мові програмування PHP, розроблено плагіни для рецензування курсу у LMS Moodle.

Висновки. Створені плагіни, що вирішують поставлені задачі.

Галузь використання – рецензування курсів у системі дистанційного навчання Moodle.

## **ABSTRACT**

Explanatory note to the diploma qualifying work of the bachelor: 141 pages, 11 tables, 56 figures, 2 appendices, 37 sources.

PLUGIN, DISTANCE LEARNING SYSTEM, LMS, REVIEW, MOODLE, PHP.

Object of study – Methods of developing plugins for LMS Moodle.

Subject of research – distance learning systems, plugins for Moodle.

The aim of the work is to study the process of creating a plugin on LMS Moodle, and increase the quality of learning through distance learning through the creation of a plugin for reviewing courses.

Materials, methods and technical means – php programming language, sql database queries, displaying results using html and scss.

Results. The development of plug-ins for Moodle in the PHP programming language has been studied, plug-ins for reviewing the course in LMS Moodle have been developed.

Conclusions. Plugins have been created to solve the set tasks.

Scope – reviewing courses in the distance learning system Moodle.

## ЗМІСТ

|                                                                       |    |
|-----------------------------------------------------------------------|----|
|                                                                       | С. |
| Перелік скорочень та умовних позначок .....                           | 9  |
| Вступ.....                                                            | 10 |
| 1 Аналіз проблеми та постановка завдань дослідження.....              | 12 |
| 1.1 Поняття системи дистанційного навчання.....                       | 12 |
| 1.2 Основні показники платформ онлайн навчання.....                   | 13 |
| 1.2.1 Загальні показники платформ онлайн навчання .....               | 13 |
| 1.2.2 Специфікація SCROM.....                                         | 16 |
| 1.2.3 Специфікація xAPI.....                                          | 18 |
| 1.2.4 Порівняння специфікацій: SCROM та xAPI .....                    | 20 |
| 1.3 Огляд популярних систем дистанційного навчання .....              | 22 |
| 1.3.1 Classroom .....                                                 | 22 |
| 1.3.2 Moodle .....                                                    | 25 |
| 1.3.3 Absorb.....                                                     | 27 |
| 1.3.4 Docebo .....                                                    | 29 |
| 1.3.5 TalentLMS .....                                                 | 31 |
| 1.3.6 Порівняння систем дистанційного навчання.....                   | 33 |
| 1.4 Висновки по розділу 1 .....                                       | 34 |
| 2 Проектування програмного забезпечення.....                          | 34 |
| 2.1 Розгортання та вимоги системи дистанційного навчання Moodle ..... | 35 |
| 2.1.1 Основні вимоги до Moodle .....                                  | 35 |
| 2.1.1.1 Вимоги до апаратного забезпечення.....                        | 35 |
| 2.1.1.2 Вимоги до програмного забезпечення.....                       | 36 |
| 2.1.2 Налаштування серверу .....                                      | 37 |
| 2.1.3 Завантаження та копіювання файлів на місце .....                | 38 |
| 2.1.4 Створення пустої бази даних.....                                | 39 |
| 2.1.5 Створення каталогу даних (moodledata).....                      | 39 |
| 2.1.6 Початок встановлення Moodle .....                               | 40 |
| 2.1.7 Остаточна конфігурація .....                                    | 40 |

|         |                                                                                                 |    |
|---------|-------------------------------------------------------------------------------------------------|----|
| 2.2     | Вибір мови програмування та технологій для реалізації backend-у програмного забезпечення .....  | 41 |
| 2.2.1   | PHP .....                                                                                       | 41 |
| 2.2.2   | NodeJS .....                                                                                    | 42 |
| 2.2.3   | Порівняльні таблиці Мов/Технологій .....                                                        | 43 |
| 2.3     | Вибір мови програмування та технологій для реалізації frontend-у програмного забезпечення ..... | 46 |
| 2.3.1   | Мови програмування для Frontend-у .....                                                         | 46 |
| 2.3.1.1 | JavaScript .....                                                                                | 46 |
| 2.3.1.2 | TypeScript .....                                                                                | 47 |
| 2.3.1.3 | CoffeeScript .....                                                                              | 47 |
| 2.3.1.4 | Порівняння мов програмування для Frontend-у .....                                               | 48 |
| 2.3.2   | Мови розмітки, опису, та препроцесори зовнішнього вигляду для Frontend-у .....                  | 50 |
| 2.3.2.1 | HTML .....                                                                                      | 50 |
| 2.3.2.2 | CSS .....                                                                                       | 51 |
| 2.3.2.3 | SASS .....                                                                                      | 51 |
| 2.3.2.4 | SCSS .....                                                                                      | 51 |
| 2.3.2.5 | Порівняння препроцесорів SCSS та SASS .....                                                     | 52 |
| 2.4     | Модель розробки та основні принципи розроблення та інтегрування плагінів .....                  | 52 |
| 2.4.1   | Основні принципи розробки плагінів .....                                                        | 53 |
| 2.4.2   | Встановлення плагінів у систему Moodle .....                                                    | 58 |
| 2.5     | Підсумки розділу 2 .....                                                                        | 60 |
| 3       | Основні рішення щодо реалізації компонентів системи .....                                       | 61 |
| 3.1     | Алгоритм рецензування курсу .....                                                               | 62 |
| 3.2     | Основні модулі розробленого додатка .....                                                       | 63 |
| 3.2.1   | Плагін-блок .....                                                                               | 64 |
| 3.2.2   | Плагін-звіт .....                                                                               | 66 |
| 3.3     | Підсумки розділу 3 .....                                                                        | 67 |

|       |                                                                 |     |
|-------|-----------------------------------------------------------------|-----|
| 4     | Опис програмного додатку .....                                  | 68  |
| 4.1   | Функціональні можливості розробленої системи рецензування ..... | 69  |
| 4.1.1 | Плагін-блок.....                                                | 69  |
| 4.1.2 | Плагін-звіт .....                                               | 70  |
| 4.2   | Додавання плагінів до системи .....                             | 72  |
| 4.3   | Призначення та умови застосування програмного забезпечення..... | 76  |
| 4.3.1 | Призначення програмного забезпечення .....                      | 76  |
| 4.3.2 | Умови застосування програмного забезпечення.....                | 77  |
| 4.3.3 | Підсумки за розділом 4 .....                                    | 78  |
| 5     | Керівництво програміста .....                                   | 78  |
| 5.1   | Характеристики програми .....                                   | 79  |
| 5.2   | Звернення до плагіна .....                                      | 79  |
| 5.3   | База даних програми.....                                        | 86  |
| 5.4   | Алгоритми програми .....                                        | 89  |
| 5.5   | Вхідні та вихідні дані .....                                    | 94  |
| 5.6   | Повідомлення, обробка помилкових ситуацій .....                 | 95  |
| 5.7   | Висновок за розділом 5 .....                                    | 97  |
| 6     | Експлуатація оператором програми .....                          | 97  |
| 6.1   | Призначення програми.....                                       | 98  |
| 6.2   | Умови використання програми .....                               | 98  |
| 6.3   | Використання програми.....                                      | 98  |
| 6.3.1 | Використання програми для ролі "Адміністратор" .....            | 98  |
| 6.3.2 | Використання програми для ролі "Рецензент" .....                | 101 |
| 6.4   | Повідомлення оператору.....                                     | 103 |
| 6.5   | Підсумки по розділу 6 .....                                     | 103 |
|       | Висновки .....                                                  | 103 |
|       | Перелік джерел та посилання.....                                | 105 |
|       | ДодатокАТекст програми.....                                     | 109 |
|       | ДодатокБСлайди презентації .....                                | 134 |

## ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

|       |                                            |
|-------|--------------------------------------------|
| API   | – Application Programming Interface;       |
| CSS   | – Cascading Style Sheets;                  |
| LMS   | – Learning Management System;              |
| SCORM | – Sharable Content Object Reference Model; |
| SCSS  | – Sassy Cascading Style Sheets;            |
| SASS  | – Syntactically Awesome Style Sheets;      |
| HTML  | – HyperText Markup Language;               |
| SQL   | – Structured Query Language;               |
| xAPI  | – The Experience API.                      |

## ВСТУП

Актуальність теми. Пандемія COVID-19 призвела до загальної соціальної ізоляції. Але все ж таки, завдяки технологіям навчання може продовжуватися навіть у цей скрутний час.

COVID-19 з'явився у 2019 році у Китаї, і незабаром поширився на всі країни. Оскільки це високо інфекційне захворювання має відносно високий рівень смертності, воно природним чином спричинило зростання страху серед людей, оскільки занепокоєння з приводу COVID-19 було засноване на контакті з людьми, які могли бути заражені цим захворюванням.

У відповідь на кризу, викликану COVID-19, уряди у всьому світі прийняли державну політику, що включає соціальне дистанціювання, ізоляцію та самокарантин, що призвело до безпрецедентних економічних та психо-соціальних наслідків у всьому світі. У той час як мільйони людей у всьому світі залишаються у своїх будинках, щоб запобігти розповсюдженню коронавірусу, їхні засоби до існування були обмежені, а у випадку студентів – їх доступ до освіти. Однак у міру того, як країни закриваються, розвиток інформаційних технологій дозволяє знайти можливі альтернативи. Кардинальні зміни, спричинені розвитком інформаційних технологій у всіх аспектах життя, особливо з урахуванням їхньої участі у вищій освіті, мають вирішальне значення для обговорення під час пандемії COVID-19. Технології завжди допомагали покращити найпростіші завдання, такі як покращення традиційного процесу навчання. Технологія, що лежить в основі електронного навчання, дозволила продовжити навчання під час блокування. Ця технологія називається системою управління навчання (LMS – Learning Management System). LMS визначається як мережна технологія, розроблена для покращення процесу навчання шляхом його належного планування, застосування та оцінки в освітніх закладах. Використання LMS у навчальному процесі допомагає полегшити електронне навчання, оскільки воно надає навчальні матеріали без обмежень за часом або місцем, дозволяючи учням та вчителям взаємодіяти

через інтернет та полегшуючи обмін інформацією, пов'язаною з курсом. Це свідчить про те, що з використанням цієї технології під час пандемії COVID-19 є необхідністю продовження процесу навчання [2].

Мета та завдання дослідження. Метою роботи є дослідження процесу навчання за допомогою LMS, та збільшення якості навчання за допомогою дистанційного навчання процесу через створення плагіну для рецензування курсів.

Виконавши наступні завдання буде досягнута поставлена мета:

- дослідити та порівняти системи дистанційного навчання;
- дослідити та порівняти реалізації навчальних платформ;
- дослідити типи плагінів для Moodle;
- дослідження проектування плагіну для Moodle;
- розробити плагін для рецензування курсів у системі Moodle;
- виконати тестування розробленого плагіну.

Об'єкт дослідження – Способи розробки плагінів для Moodle.

Предмет дослідження – системи дистанційного навчання, плагіни для Moodle.

Методи дослідження. Для розробки плагіну для Moodle було використано сучасні технології розроблення веб застосунків. Плагін було написано за допомогою таких мов програмування та препроцесорів: PHP, JavaScript, SCSS.

Наукова новизна роботи полягає у тому, що виконано дослідження новітньої технології – системи дистанційного навчання, та розроблено програмну модифікацію до модульної системи дистанційного навчання Moodle.

Практична цінність результатів роботи полягає у тому, цей плагін дозволяє робити рецензування курсу, що покращує якість викладання матеріалу.

# **1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ**

## **1.1 Поняття системи дистанційного навчання**

Система управління навчанням (LMS) – це програмне забезпечення або вебтехнологія, яка використовується для планування, впровадження та оцінки конкретного процесу навчання [3]. Вона використовується для практик електронного навчання і, у своїй найпоширенішій формі, складається з двох елементів: сервера, який виконує базову функціональність, та інтерфейсу користувача, яким керують викладачі, студенти та адміністратори.

Як правило, система управління навчанням надає викладачеві можливість створювати й передавати контент, контролювати участь студентів та оцінювати результати діяльності учнів. Система управління навчанням також може надати студентам можливість використовувати інтерактивні функції, такі як ланцюжкові дискусії, відеоконференції та дискусійні форуми.

LMS часто використовуються підприємствами будь-якого розміру, національними урядовими установами, місцевими органами влади, традиційними навчальними закладами та закладами на базі онлайн або електронного навчання. Системи можуть покращити традиційні методи навчання, а також заощадити час і гроші організації. Ефективна система дозволить інструкторам та адміністраторам ефективно керувати такими елементами, як реєстрація користувачів, вміст, календарі, доступ користувачів, комунікація, сертифікати та сповіщення.

Група розширеного дистанційного навчання, спонсорована Міністерством оборони Сполучених Штатів, створила набір специфікацій під назвою Shareable Content Object Reference Model (SCORM), щоб заохочувати стандартизацію систем управління навчанням [3].

Системи управління навчанням надають користувачам різноманітні переваги, незалежно від типу організації, яка їх використовує. Наприклад, LMS може заощадити час і гроші організації. Замість того, щоб змусити учнів виділяти час для подорожей і проходження уроків або тренінгів, LMS

дозволяють користувачам виконувати курсову роботу в зручний для них час. Крім того, витрати можна скоротити, усунувши потребу в інструкторах, днях навчання, навчальних матеріалах, витратах на відрядження та найму місця.

Інші переваги систем управління навчанням включають:

- можливість контролювати прогрес і продуктивність користувачів;
- підвищена доступність електронного навчання без географічних обмежень;
- можливість персоналізувати онлайн-навчання та досвід навчання;
- можливість легко та ефективно оновлювати модулі та заходи дистанційного навчання;
- можливість легко забезпечити ефективне розповсюдження навчальних і навчальних матеріалів онлайн;
- використання автоматизації, яка дозволяє користувачам забути про втомливі, повторювані завдання, такі як реєстрація користувачів і розповсюдження сертифікатів, і зосередитися на більш важливих справах.

Нарешті, централізоване навчання є ще однією важливою перевагою LMS. Це дозволяє організації безпечно організовувати та зберігати всі великі дані в одному місці. Це дозволяє викладачам та адміністраторам легше оновлювати й підтримувати навчальні матеріали. Це також допомагає проводити ефективне навчання, яке є послідовним у всій організації. Крім того, більшість LMS включають розширені функції шифрування, які допомагають гарантувати безпеку даних і вмісту[3].

## **1.2 Основні показники платформ онлайн навчання**

### **1.2.1 Загальні показники платформ онлайн навчання**

Систему управління навчанням можна розглядати як велике сховище, яке дозволяє користувачам зберігати та відстежувати інформацію в одному місці. Будь-який користувач із безпечним логіном та паролем може отримати доступ до системи та її онлайн-ресурсів навчання. Або, якщо система

розміщена самостійно, користувач повинен або встановити програмне забезпечення на свій жорсткий диск, або отримати до нього доступ через сервер своєї компанії[3].

Деякі загальні показники:

- адаптивний дизайн – користувачі повинні мати можливість отримати доступ до LMS з будь-якого типу пристрою, який вони вибирають, будь то настільний комп'ютер, ноутбук, планшет або смартфон. LMS має автоматично відображати версію, яка найкраще підходить для пристрою, вибраного користувачем. Крім того, LMS також має дозволяти користувачам завантажувати вміст, щоб він був доступним в автономному режимі[3];

- зручний інтерфейс – користувацький інтерфейс (UI – User Interface) повинен дозволити учням легко орієнтуватися на платформі LMS. Інтерфейс користувача також повинен відповідати можливостям і цілям як користувача, так і організації. Неінтуїтивно зрозумілий інтерфейс користувача може збити з пантелику або відриватикористувачів і зробить LMS неефективною [3];

- звіти та аналітика – це включає інструменти оцінювання дистанційного навчання. Інструктори та адміністратори повинні мати можливість переглядати та відстежувати свої онлайн-тренінги, щоб визначити, чи ефективні вони, чи потребують коригування. Це можна застосувати до груп учнів та окремих осіб[3];

- управління курсами та каталогами – LMS містить усі курси електронного навчання та відповідний зміст курсів. Адміністратори та викладачі повинні мати можливість створювати ці каталоги та курси та керувати ними, щоб забезпечити більш цілеспрямований досвід навчання [3];

- сумісність та інтеграція вмісту– Контент, створений та збережений у LMS, має бути запакований відповідно до стандартів взаємодії, включаючи SCORM [3];

- послуги підтримки – різні постачальники LMS пропонують різні рівні підтримки. Багато з них пропонують онлайн-дошки обговорень, де

користувачі можуть спілкуватися та допомагати один одному. Додаткові послуги підтримки, наприклад виділений безкоштовний номер служби, доступні за додаткову плату [3];

- підтримка сертифікації та відповідності. Ця функція є важливою для систем, які використовуються для онлайн-навчання та сертифікації щодо відповідності. Інструктори та адміністратори повинні бути в змозі оцінити набір навичок особи та виявити будь-які прогалини в їхній продуктивності. Ця функція також дозволить використовувати записи LMS під час аудиту [3];

- можливості соціального навчання. Багато LMS почали включати інструменти соціальних мереж у свою платформу. Це дозволяє користувачам взаємодіяти зі своїми однолітками, співпрацювати та ділитися своїм досвідом навчання [3];

- гейміфікація – деякі LMS включають ігрову механіку або вбудовані функції гейміфікації, які дозволяють інструкторам і адміністраторам створювати курси з додатковою мотивацією та залученням. Це може допомогти студентам, які потребують додаткового стимулу для завершення курсу, можливо, у вигляді таблиць лідерів, балів та значків [3];

- автоматизація. Системи керування навчанням повинні дозволяти адміністраторам автоматизувати повторювані та виснажливі завдання. Приклади включають групування користувачів, нову групу користувачів, деактивацію користувачів та реєстрацію груп [3];

- локалізація – для LMS важливо включати багатомовні функції підтримки, щоб на зміст навчання та навчання не впливали мовні бар'єри. Деякі LMS інтегрують функції геолокації, які дозволяють їм автоматично представляти відповідну версію курсу відразу після доступу [3];

- штучний інтелект– штучний інтелект може допомогти LMS створювати персоналізований досвід навчання для користувачів, надаючи формати курсів, які відповідають їхнім потребам, і пропонуючи теми, які користувачі можуть виявити цікавими на основі курсів, які вони вже закінчили [3].

### 1.2.2 Специфікація SCORM

SCORM, що розшифровується як довідкова модель об'єкта спільного вмісту, являє собою набір технічних стандартів для програмних продуктів дистанційного навчання. SCORM розповідає програмістам, як писати свій код, щоб він міг «добре грати» з іншим програмним забезпеченням для електронного навчання. Це де-факто галузевий стандарт для сумісності електронного навчання. Зокрема, SCORM регулює взаємодію між контентом онлайн-навчання та системами управління навчанням (LMS). SCORM не говорить про дизайн навчання чи будь-які інші педагогічні проблеми – це суто технічний стандарт [4].

SCORM 2004 представив складну ідею під назвою послідовність, яка являє собою набір правил, які визначають порядок, у якому учень може відчувати об'єкти вмісту. Простіше кажучи, вони обмежують учня фіксованим набором шляхів через навчальний матеріал, дозволяють учню «закладати» свій прогрес під час перерв і гарантують прийнятність результатів тестів, отриманих учнем. У стандарті використовується XML, і він базується на результатах роботи AICC, IEEE LTSC та Ariadne [5].

Технічний огляд стандарту SCORM:

SCORM вказує, що вміст має:

- бути запакованим у ZIP-файл;
- описуватися у файлі XML;
- спілкуватися за допомогою JavaScript;
- послідовність використання правил у XML.

SCORM складається з трьох підспецифікацій:

- розділ «Content Packaging» визначає, як вміст має бути упакований та описаний. В основному він заснований на XML;
- у розділі «Run-Time» вказується, як має бути запущений контент і як він взаємодіє з LMS. В основному він заснований на ECMAScript;

– розділ «Sequencing» визначає, як учень може переміщатися між частинами курсу. Він визначається набором правил і атрибутів, написаних у XML.

Упаковка вмісту. SCORM Content Aggregation Model (CAM) вказує, що вміст має бути упаковано в автономний каталог або файл ZIP. Ця доставка називається файлом обміну пакетами (PIF). PIF завжди повинен містити файл XML з ім'ям `imsmanifest.xml` («файл маніфесту») у корені. Файл маніфесту містить всю інформацію, необхідну LMS для доставки вмісту. Маніфест розділяє курс на одну або кілька частин, які називаються SCO. SCO можна об'єднати в структуру дерева, яка представляє курс, відому як «дерево діяльності». Маніфест містить XML-подання дерева діяльності, інформацію про те, як запустити кожну SCO та (за бажанням) метадані, які описують курс та його частини.

Середовище виконання. У специфікації часу виконання зазначено, що LMS має запускати вміст у веб-браузері або в новому вікні, або у наборі фреймів. LMS може одночасно запускати лише одну SCO. Весь вміст має бути доступним через Інтернет, і він завжди запускається у веб-браузері. Після запуску вмісту він використовує чітко визначений алгоритм для пошуку API ECMAScript (JavaScript), який надається LMS. Цей API має функції, які дозволяють обмінюватися даними з LMS. Модель даних СМІ надає список елементів даних(словник), які можна записати та прочитати з LMS. Деякі приклади елементів моделі даних включають статус SCO (завершено, пройдено, не успішно тощо), оцінку, яку отримав учень, закладку для відстеження місцезнаходження учня та загальну кількість часу, який учень провів у SCO.

Послідовність і навігація. Специфікація послідовності дозволяє автору вмісту керувати тим, як учню дозволяється переміщатися між SCO і як дані прогресу згортаються до рівня курсу. Правила послідовності представлені XML у маніфесті курсу. Секвенування працює на основі моделі відстеження, яка

тісно збігається з даними СМІ, які повідомляються SCO під час виконання.

Правила послідовності дозволяють автору вмісту робити такі дії, як:

- визначити, які навігаційні елементи керування LMS має надати користувачеві (кнопки «попередній/наступний», навігаційний зміст тощо);
- вказати, що певні дії мають бути виконані раніше інших (передумови);
- зобити так, щоб деякі частини курсу враховувалися більше, ніж інші, для остаточного статусу або оцінки (створення необов'язкових розділів або надання зважування питань);
- випадково обрати різну підмножину доступних SCO, які будуть доставлені при кожній новій спробі (наприклад, щоб увімкнути тестовий банкінг);
- повернути користувача до неосвоєного навчального матеріалу (виправлення).

Як видно, SCROM є складним стандартом повної підтримки. Час, необхідний для досягнення відповідності SCORM, зазвичай вимірюється «роками розробника»[6].

### 1.2.3 Специфікація xAPI

Experience API (або xAPI) – це нова специфікація для технології навчання, яка дає змогу збирати дані про широкий спектр досвіду, який має людина (онлайн та оффлайн). Цей API збирає дані в узгодженому форматі про діяльність особи або групи з багатьох технологій. Дуже різні системи можуть безпечно спілкуватися, фіксуючи та передаючи цей потік діяльності за допомогою простого словника xAPI.

Як працює xAPI:

- люди вчаться на взаємодії з іншими людьми, на контенті тощо. Ці дії можуть відбуватися в будь-якому місці і сигналізувати про подію, на якій може статися навчання;

- коли діяльність потрібно записати, програма надсилає захищені заяви у формі «Назва, дія, об'єкт» або «Я зробив це» до магазину записів для навчання (LRS);
- вмагазини записів навчання записують усі зроблені заяви. LRS може ділитися цими заявами з іншими LRS. LRS може існувати самостійно або всередині LMS.

#### Свободи Experience API:

- свобода висловлювань: структура «висловлень» із використанням іменників, дієслів та предметів дозволяє записувати практично будь-яку діяльність;
- свобода історії: Experience API дозволяє LRS спілкуватися один з одним. LRS можуть обмінюватися даними та стенограмами один з одним, і ваш досвід може переслідувати вас від однієї LRS (або організації) до іншої. Учні навіть можуть мати власні «шафки для персональних даних» з особистою навчальною інформацією всередині них;
- свобода пристрою: будь-який увімкнений пристрій може надсилати заяви API Experience (мобільні телефони, моделювання, ігри, манекен CPR, список можна продовжувати). Постійне мережеве з'єднання не потрібне – це нормально;
- свобода робочого процесу: відстеження навчальних подій не повинно починатися або закінчуватися в LMS, воно може починатися де б учень і на будь-якому пристрої, який він вирішив використовувати. Вміст не прив'язаний до LMS.

Специфікація SCROM є складними та є обмеження, але Experience API простий і гнучкий. Це знімає багато старих обмежень. Мобільне навчання, моделювання, віртуальні світи, серйозні ігри, діяльність у реальному світі, експериментальне навчання, соціальне навчання, оффлайн-навчання та спільне навчання – це лише деякі з речей, які тепер можна розпізнати та добре передати за допомогою Experience API [7].

### 1.2.4 Порівняння специфікацій: SCORM та xAPI

SCORM є найбільш поширеним стандартом електронного навчання. Якщо LMS відповідає SCORM, то він може відтворювати будь-який вміст SCORM, і, навпаки, будь-який вміст SCORM можна відтворювати в будь-якому LMS, що відповідає SCORM. Така сумісність робить усіх щасливими та заощаджує багато часу та грошей.

SCORM добре працює, але насправді він не відображає повної картини електронного навчання.

Навчання відбувається скрізь, а не лише на традиційних курсах SCORM у традиційних LMS. API Experience дозволяє записувати будь-який досвід навчання, де б і як би це не відбувалося. API Experience дає можливість побачити всю картину.

Отже, які відмінності між SCORM і Experience API? Наведена нижче таблиця 1.1 досить добре підсумовує це [8].

Таблиця 1.1 – Порівняння SCORM та xAPI [8]

| Функція                         | SCORM | xAPI |
|---------------------------------|-------|------|
| 1                               | 2     | 3    |
| Завершення треку                | Так   | Так  |
| Відстежування часу              | Так   | Так  |
| Відстеження пройдено/не вдалося | Так   | Так  |
| Повідомлення про один бал       | Так   | Так  |
| Повідомлення про декілька балів | Ні    | Так  |
| Детальні результати тесту       | Ні    | Так  |
| LMS не потрібно                 | Ні    | Так  |
| Інтернет-браузер не потрібен    | Ні    | Так  |

Продовження таблиці 1.1

| 1                                                   | 2  | 3   |
|-----------------------------------------------------|----|-----|
| Повний контроль над своїм вмістом                   | Ні | Так |
| Немає междоменних обмежень                          | Ні | Так |
| Використовування мобільні додатки для навчання      | Ні | Так |
| Перехід платформи (тобто з комп'ютера на мобільний) | Ні | Так |
| Відстежування серйозних ігор                        | Ні | Так |
| Симуляції треків                                    | Ні | Так |
| Відстежування неформального навчання                | Ні | Так |
| Відстежування реальної продуктивності               | Ні | Так |
| Відстежування навчання в автономному режимі         | Ні | Так |
| Відстежування інтерактивного навчання               | Ні | Так |
| Відстежування адаптивного навчання                  | Ні | Так |
| Відстежування змішаного навчання                    | Ні | Так |
| Відстежування довготривалого навчання               | Ні | Так |
| Відстежування командного навчання                   | Ні | Так |

### 1.3 Огляд популярних систем дистанційного навчання

#### 1.3.1 Classroom

Google Classroom – це безкоштовна платформа змішаного навчання, розроблена Google для навчальних закладів, яка спрямована на спрощення створення, розподілу та оцінювання завдань. Основна мета Google Classroom – оптимізувати процес обміну файлами між викладачами та учнями. Станом на 2021 рік Google Classroom користується приблизно 150 мільйонами користувачів [9]. Інтерфейс вебсайту наведено нижче у рисунку 1.1.

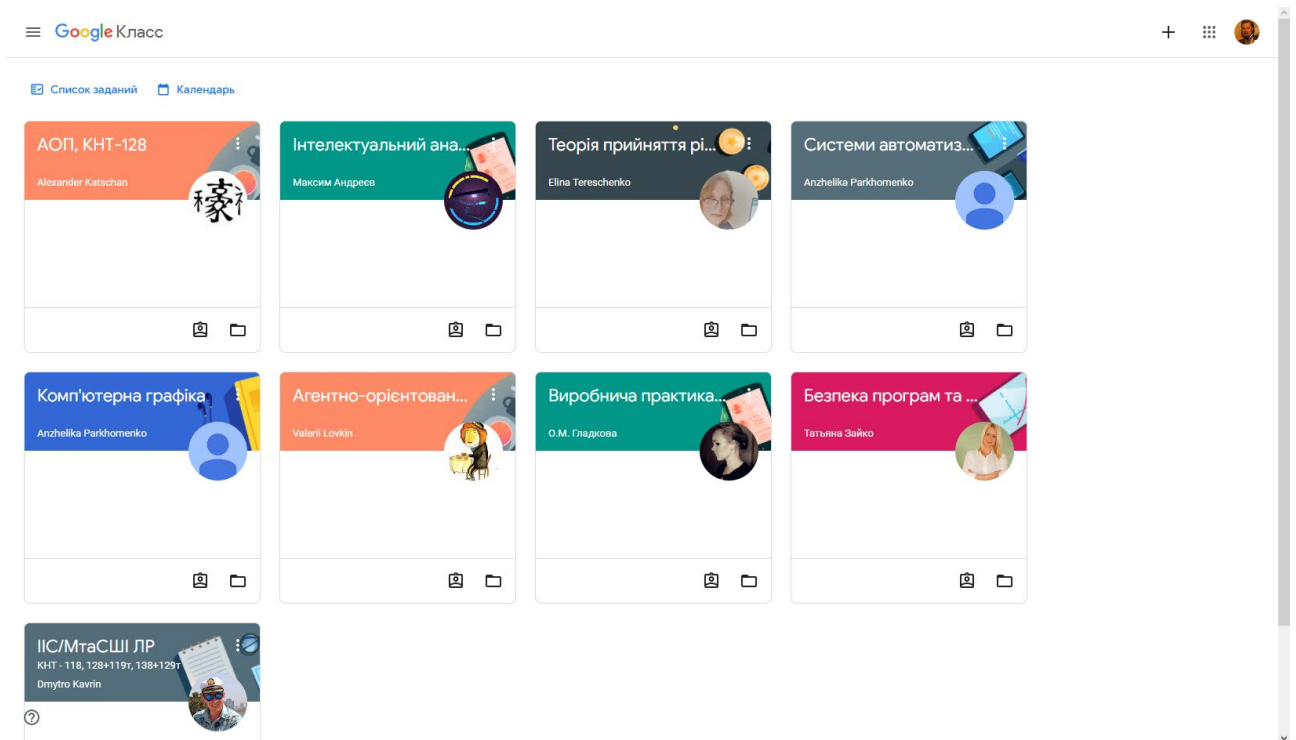


Рисунок 1.1 – Скріншот Classroom

Google Classroom інтегрує різноманітні вебзастосунки Google для освіти у єдину платформу для керування спілкуванням учнів і викладачів. Студентів можна запросити приєднатися до класу через приватний «код класу» або автоматично імпортувати з домену школи. Викладачі можуть створювати, розповсюджувати та позначати завдання в межах домену Google. Кожен клас створює окрему папку на Google Диску відповідного користувача, куди учень

може подати роботу для оцінки вчителю. Завдання та терміни виконання додаються до Календаря Google, де кожне завдання може належати до категорії або теми. Вчителі можуть стежити за прогресом кожного учня, переглядаючи історію переглядів документа, а після оцінки вчителі можуть повертати роботу разом із коментарями та оцінками [9].

Ключові особливості цієї платформи:

- завдання зберігаються та оцінюються в наборі програм Google для підвищення продуктивності, які дозволяють співпрацювати між викладачем і студентом або між учнями. Замість того, щоб надавати вчителю доступ до документів, які знаходяться на Диску Google учня, файли розміщуються на Диску учня, а потім надсилаються для оцінки. Учителі в Google Classroom мають можливість створювати завдання в різних шаблонах і форматах з різними параметрами доступності: «учень може переглядати файл», «студент може редагувати файл» або «зробити копію для кожного учня». Ці завдання можна надіслати для оцінки та відгуку вчителя. Студенти також можуть додати до завдання додаткові документи зі свого Диска. Встановлені завдання відображаються в списку «Справи» [9];

- Google Classroom підтримує багато різних схем оцінювання. Викладачі мають можливість прикріпити файли до завдання, які студенти можуть переглядати, редагувати або отримати окрему копію. Учні можуть створювати файли, а потім прикріплювати їх до завдання, якщо копію файлу створив не вчитель. Вчителі мають можливість контролювати прогрес кожного учня у виконанні завдання, де вони можуть коментувати та редагувати. Здані завдання можуть бути оцінені вчителем та повернуті з коментарями, щоб учень міг переглянути завдання та повернутись назад. Після здачі завдання може бути відредагований лише вчителем, якщо вчитель не поверне завдання назад [9];

- вчителі можуть публікувати оголошення в потоці класу, які можуть коментувати учні, забезпечуючи двостороннє спілкування між викладачем та учнями. Студенти також можуть публікувати повідомлення в потоці класу, але це не буде таким пріоритетом, як оголошення вчителя, і їх можна модерувати.

Щоб поділитися вмістом, до оголошень і публікацій можна прикріпити багато типів медіа з продуктів Google, як-от відео YouTube та файли Google Диска. Gmail також надає варіанти електронної пошти, щоб викладачі могли надсилати електронні листи одному або кільком учням в інтерфейсі Google Classroom. Доступ до Classroom можна отримати в Інтернеті або за допомогою мобільних додатків Classroom для Android та iOS [9];

- звіт про оригінальність дає змогу викладачам та студентам побачити частини та розділи поданої роботи, які містять точні чи схожі формулювання з іншими джерелами. Для студентів він виділяє вихідні матеріали та позначає відсутні цитати, щоб допомогти студентам покращити їх написання. Викладачі також можуть переглядати звіт про оригінальність, що дозволяє їм перевірити академічну чесність поданої студентом роботи. У G Suite for Education (безкоштовно) викладачі можуть увімкнути звіт про оригінальність для 3 завдань, але мають обмежене хмарне сховище. Це обмеження скасовано для G Suite Enterprise for Education (платний) [9];

- Classroom дозволяє викладачам архівувати курси в кінці семестру або року. Коли курс архівується, він видаляється з домашньої сторінки та розміщується в області заархівованих класів, щоб допомогти вчителям упорядкувати свої поточні заняття. Коли курс заархівовано, викладачі та студенти можуть переглядати його, але не зможуть вносити в нього жодних змін, доки його не відновлять [9];

- мобільна програма Google Classroom доступна для мобільних пристроїв. Вона дозволяє користувачам фотографувати та прикріплювати фото до своїх завдань, обмінюватися файлами і підтримує оффлайн-доступ, полегшуючи доступ для користувачів [9].

Переваги цієї платформи:

- безкоштовна система керування навчанням [11];
- дозволяє працювати безпосередньо в середовищі Google Workspace, створюючи документи, організовувати класну роботу та проводити онлайн-лекції за допомогою Google Meet. Можна додати матеріал до своєї програми,

використовуючи готові шаблони для завдань, вікторин та уроків. Також можна повторно використовувати або перепрофілювати матеріал, який створили раніше [11];

– інтегрується безпосередньо з Gmail, Календарем Google і Google Диском. Це означає, що можна за потреби витягати з Google Презентацій, Документів і Таблиць. Також можна запланувати уроки, тести чи робочий час прямо у своєму календарі. Також можна створювати власні рубрики, відстежувати призначені та передані дані та навіть використовувати перевірку на плагіат [11].

### **1.3.2 Moodle**

Moodle – це безкоштовна система управління навчанням (LMS) з відкритим вихідним кодом, написана на PHP і поширюється під загальною суспільною ліцензією GNU. Розроблений на основі педагогічних принципів, Moodle використовується для змішаного навчання, дистанційної освіти, перевернутих класів та інших проектів електронного навчання в школах, університетах, на робочих місцях та інших секторах [10]. Інтерфейс програми наведено нижче у рисунку 1.2.

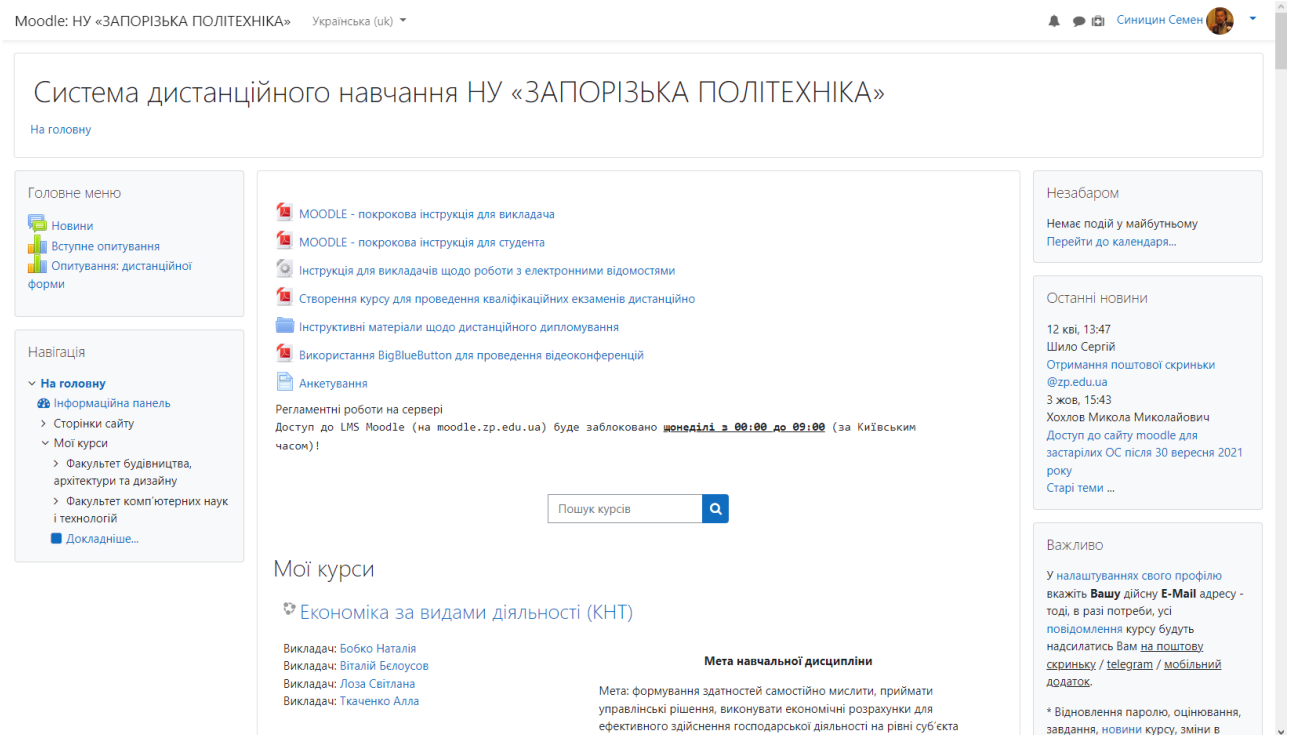


Рисунок 1.2 – Скріншот Moodle

Завдяки налаштувальним функціям керування Moodle використовується для створення приватних вебсайтів з онлайн-курсами для викладачів і тренерів для досягнення цілей навчання. Moodle дозволяє розширювати та адаптувати навчальні середовища за допомогою плагінів, створених спільнотою [10].

Ця навчальна платформа, використовується для розширення й переміщення існуючих навчальних середовищ в Інтернет. Як інструмент електронного навчання, Moodle розробив ряд функцій, які зараз вважаються стандартними для систем управління навчанням, таких як календар і зошит. Moodle – це віртуальне навчальне середовище, яке може використовуватися в багатьох типах середовищ, таких як освіта, навчання, і розвитку та в бізнес-середовищах.

Moodle підтримує 2 типа користувацьких модифікацій:

— плагіни – це гнучкий набір інструментів, що дозволяє користувачам Moodle розширювати можливості сайту. Moodle має сотні доступних плагінів, які розширюють функції основної функціональності Moodle. Кожен плагін

зберігається в каталозі плагінів Moodle. Станом на жовтень 2019 року для Moodle було доступно 1619 плагінів із понад 380 400 останніх завантажень;

- графічні теми для Moodle можна встановити, щоб змінити вигляд і функціональність сайту Moodle або окремого курсу.

Багато тем Moodle, засновані на адаптивному вебдизайні, дозволяють використовувати Moodle на мобільних пристроях. Також мобільний додаток Moodle доступний у Google Play, App Store, F-Droid та Windows Phone Store [10].

Ключові особливості цієї платформи:

- є велика спільнота розробників, які забезпечують надходження нових функцій і виправлень безпеки у вигляді регулярних оновлень [11];
- можна легко налаштувати роль користувача або процес навчання відповідно до моделі навчання [11];
- для забезпечення абсолютної безпеки moodle пропонує можливість розгорнути програмне забезпечення LMS на приватному сервері [11].

Переваги цієї платформи:

- підтримує необмежену кількість користувачів [11];
- має персоналізовану інформаційну панель і настроювану функцію звітування про дані [11];
- для додаткових функцій LMS він забезпечує інтеграцію з великою кількістю плагінів і доповнень [11];
- пропонує зручний і зрозумілий інтерфейс користувача [11];
- підтримує як Android, так і Windows [11].

### **1.3.3 Absorb**

Absorb LMS – це відзначена нагородами система управління навчанням із розумними адміністративними функціями, високоякісними інструментами залучення учнів і навіть варіантами електронної комерції для продажу

спеціально створених курсів онлайн [12]. Інтерфейс програми наведено нижче у рисунку 1.3.

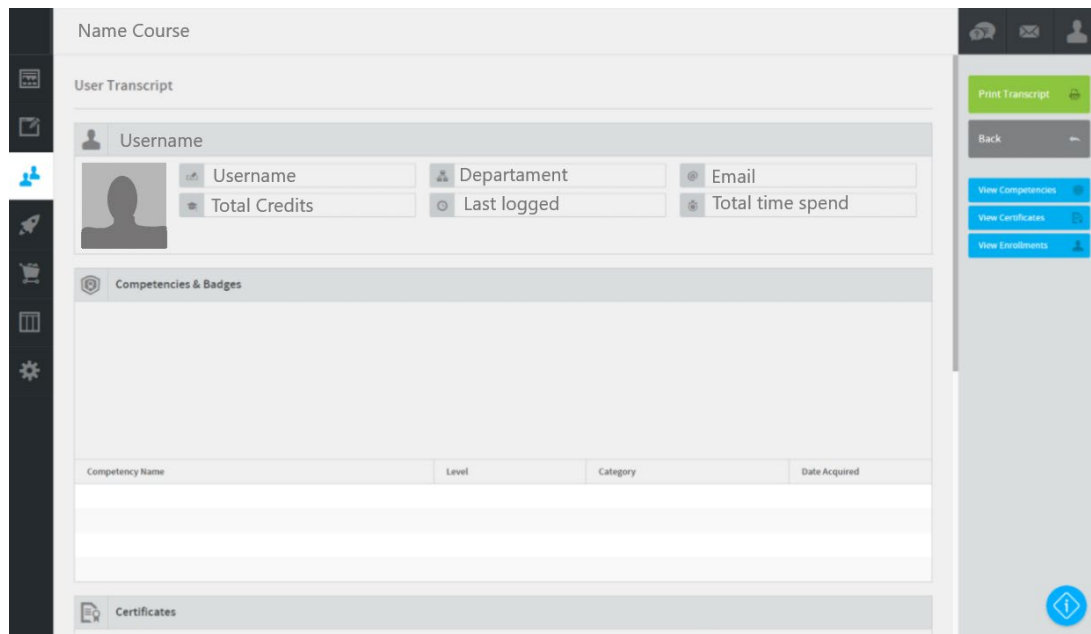


Рисунок 1.3 – Скріншот Absorb

Absorb LMS дає організаціям можливість проводити цікаве навчання співробітників, яке відповідає потребам і стандартам кожного учня, надаючи їм навички, необхідні для того, щоб залишатися сучасними та актуальними в сучасному швидко розвивається світі. Його сучасний дизайн включає інтуїтивно зрозумілу інформаційну панель з підтримкою кількох мов і повністю відповідає стандартам WCAG 2.0, що робить вашу онлайн-програму розвитку співробітників настільки ж інклюзивною, як і ефективною.

Absorb LMS прискорює роботу нових працівників, а потім використовує потужний штучний інтелект, щоб налаштувати досвід навчання для кожного співробітника, передбачаючи їхні потреби та підключаючи їх до відповідного навчання.

Absorb LMS інтегрується з провідними в галузі бізнес-рішеннями, такими як Zoom, ADP, і також Microsoft Teams, і надає доступ до тисяч готових до використання курсів через партнерів бібліотек вмісту, таких як Skillsoft, BizLibrary, тощо [12].

Ключові особливості цієї платформи:

- розумне адміністрування дозволяє легко керувати користувачами, курсами та вмістом[11];
- розширені можливості звітності та аналітики відстежують прогрес і продуктивність співробітників з часом [11];
- комплексні функції керування курсами дозволяють швидко й легко створювати, керувати та передавати навчальний контент [11];
- інструменти залучення учнів підтримують інтерес і мотивацію учнів[11];
- надійні функції електронної комерції Absorb дають змогу продавати навчальний контент і відстежувати продажі в режимі реального часу [11];
- досвід для навчання є гладким, інтуїтивно зрозумілим і простим у використанні [11].

Переваги цієї платформи:

- легке адміністрування [11];
- автоматизація для масштабованості адміністратора [11];
- сучасний досвід користувача [11];
- комплексні характеристики [11];
- звітність та аналітика, які можна діяти [11];
- попередньо вбудовані та інтеграції API [11];
- внутрішнє обслуговування клієнтів [11].

#### **1.3.4 Docebo**

Docebo – це програмне забезпечення, що спеціалізується на системі управління навчанням. Docebo було засновано в 2005 році. Його продукт Docebo Learn LMS сумісний з довідковою моделлю об'єкта спільного доступу (SCORM) 1.2 і 2004, а також з консервною банкою. Програма працює як програмне забезпечення, розміщене в хмарі, як сервісна платформа, а також

сумісна зі сторонніми розробниками. Наразі він працює версії 7.5. DoceboLMS доступний більш ніж 40 мовами [13]. Інтерфейс програми наведено нижче у рисунку 1.4.

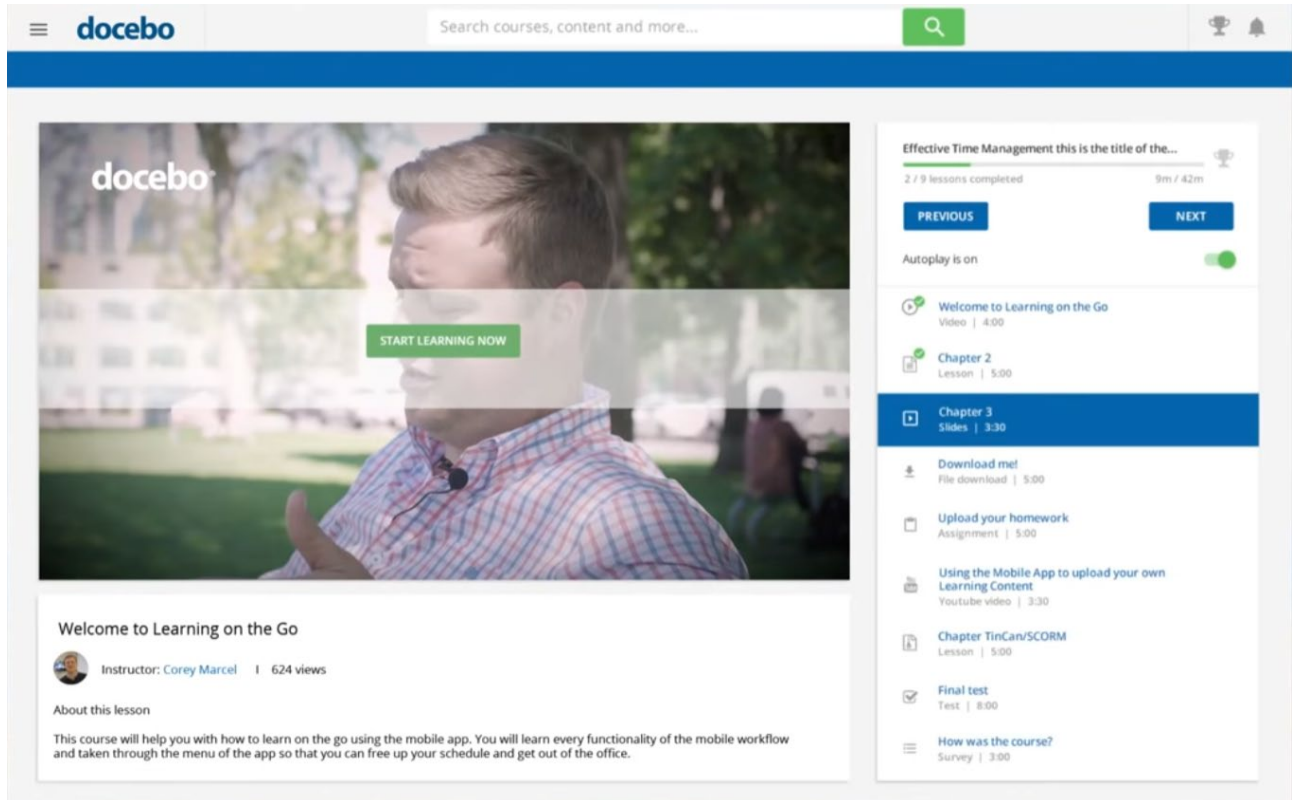


Рисунок 1.4 – Скріншот Docebo

У жовтні 2018 року Docebo випустила на навчальну платформу свої алгоритми штучного інтелекту для навчання, які працюють на основі комбінації машинного навчання, глибокого навчання та обробки природної мови, щоб створити автоматизований і цілісний підхід до навчання (формальний LMS, досвідчений та соціальний). що стимулює зростання, ефективність організації та прибуток [13].

Ключові особливості цієї платформи:

- можна створювати нові курси, додавати користувачів і запускати їх лише кількома кліками [11];
- Cmi5 – ця специфікація на основі xAPI сприяє ефективному обміну даними eLearning [11];

- можна легко організовувати файли, ділитися ними з користувачами, повторно використовувати їх для навчальних модулів [11];
- можна оцінювати прогрес учнів за допомогою кількох типів тестів і аналізувати результати тестів за допомогою різних звітів з даними [11].

Переваги цієї платформи:

- можна легко відстежувати та вимірювати вплив вашої навчальної програми за допомогою налаштованої інформаційної панелі та звітів, які містять лише необхідні дані [11];
- Досебо пропонує простий у використанні інтерфейс користувача, який дозволяє учням переглядати та купувати навчальні модулі/каталоги вмісту [11];
- підвищена зацікавленість користувачів, вводячи цікаві елементи, такі як значки, бали та нагороди [11].

### **1.3.5 TalentLMS**

TalentLMS – є гнучкою системою управління навчанням, дозволяє створювати інстинктивне, компактне, безпроблемне та просте навчання. Ця онлайн платформа, яка доступна і забезпечує велику масштабованість та вільну потужність [14]. Інтерфейс програми наведено нижче у рисунку 1.5.

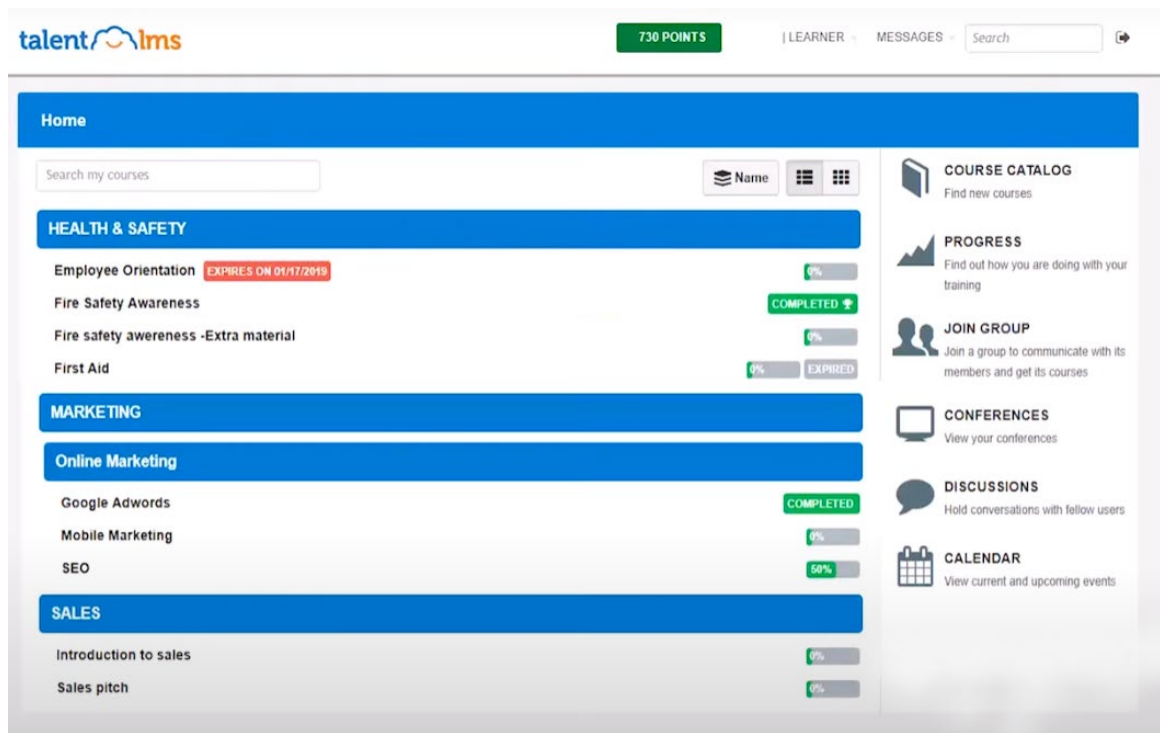


Рисунок 1.5 – СкріншотTalentLMS

TalentLMS реалізує потужні засоби розробки програм, які адаптуються до різних платформ онлайн-навчання [13].

Ключові особливості цієї платформи:

- можна створювати нові курси, додавати користувачів і запускати їх лише кількома кліками [11];
- Smi5 – ця специфікація на основі xAPI сприяє ефективному обміну даними eLearning [11];
- можна легко організовувати файли, ділитися ними з користувачами, повторно використовувати їх для навчальних модулів [11];
- можна оцінювати прогрес учнів за допомогою кількох типів тестів і аналізувати результати тестів за допомогою різних звітів з даними [11].

Переваги цієї платформи:

- автоматизація, яка дозволяє створювати навчальні програми для учнів з обмеженими шляхами навчання [11];
- забезпечує поєднання електронного навчання та навчання під керівництвом інструктора [11];

- підтримує календар, особисті повідомлення та дискусійні форуми [11];
- дозволяє монетизувати матеріали курсу, продаючи їх через PayPal або Stripe [11];
- дозволяє керувати своїм навчальним середовищем, розділяючи їх на філії, кожна філія буде діяти як окрема сутність зі своїми користувачами, курсами та темою [11];
- повний контроль управління контентом і створення контенту [11];
- є доступ до навчальних матеріалів курсу, створення курсів та керування курсами [11].

### 1.3.6 Порівняння систем дистанційного навчання

Для порівняння засобів реалізації систем дистанційного навчання була побудована таблиця 1.2:

Таблиця 1.2 – Порівняння засобів реалізації LMS

| LMS       | Засоби встановлення | Система поширення                                                                      |
|-----------|---------------------|----------------------------------------------------------------------------------------|
| 1         | 2                   | 3                                                                                      |
| Classroom | Розміщення у хмарі  | Безкоштовно                                                                            |
| Moodle    | Розміщення у хмарі  | Безкоштовно                                                                            |
|           | Локально            |                                                                                        |
|           | Відкритий API       |                                                                                        |
| Absorb    | Локально            | Платно(Залежно від кількості аудиторії)[15]                                            |
|           | Хмарний хостинг     |                                                                                        |
| Docebo    | Відкритий API       | Платно(Залежно від кількості аудиторії)[16]                                            |
|           | Розміщення у хмарі  |                                                                                        |
| TalentLMS | Відкритий API       | Безкоштовно(максимум 5 користувачів)<br>абоПлатно(від 69 до 459 доларів за місяць)[17] |
|           | Розміщення у хмарі  |                                                                                        |

## 1.4 Висновки по розділу 1

У розділі було розглянуто поняття системи дистанційного навчання. Розглянуто показники платформ навчання. Було розглянуто та порівняно у вигляді таблиці специфікації SCROM та xAPI. Було проведено огляд популярних систем дистанційного навчання та порівняно функції та засоби реалізації у вигляді таблиць.

Серед розглянутих LMS були:

- Classroom;
- Moodle;
- Absorb;
- Docedo;
- TalentLMS.

Для написання плагіну було обрано систему дистанційного навчання Moodle. Бо ця система є безкоштовною, та вже стоїть як основний навчальний портал університету. Розробка цього плагіну допоможе університету покращити процес навчання.

## **2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

### **2.1 Розгортання та вимоги системи дистанційного навчання Moodle**

Moodle в основному розробляється в Linux за допомогою Apache, PostgreSQL/MySQL/MariaDB і PHP (іноді відомий як платформа LAMP). Зазвичай так само запускається Moodle, хоча є й інші варіанти, якщо дотримуються вимоги до програмного забезпечення випуску.

Для встановлення Moodle на сервер Windows, починаючи з php5.5 також потрібно інсталиувати Visual C++ Redistributable для Visual Studio 2012 [18].

#### **2.1.1 Основні вимоги до Moodle**

##### **2.1.1.1 Вимоги до апаратного забезпечення**

Основні вимоги до апаратного забезпечення:

- місце на диску: 200 МБ займає Moodle, плюс місце для контенту, який буде розміщено на платформі. 5 ГБ – реалістичний мінімум;
- процесор: 1 ГГц (хв), рекомендовано двоядерний 2 ГГц або більше;
- пам'ять: 512 МБ (хв), рекомендовано від одного гігабайту.

Усі вище зазначені вимоги будуть відрізнятися залежно від типу використання так комбінацій апаратного та програмного забезпечення – зайняті сайти цілком можуть вимагати додаткових ресурсів [18].

Найшвидша та найефективніша зміна, яку можна зробити для підвищення продуктивності, – це збільшити обсяг оперативної пам'яті на веб-сервері – отримати якомога більше (наприклад, 4 ГБ або більше). Збільшення первинної пам'яті зменшить необхідність переміщення процесів на диск і дозволить серверу працювати з більшою кількістю користувачів.

Декілька порад до налаштування апаратної частини:

- а) краща продуктивність досягається завдяки отриманню найкращих можливостей процесора, тобто двоядерних(або міцніше) процесорів. Сучасний BIOS повинен дозволяти увімкнути гіперпотоковість;

б) використання жорстких дисків SCSI замість дисків SATA. Диски SATA збільшать використання ЦП системою, тоді як диски SCSI мають власні вбудовані процесори і стають власними, коли є кілька дисків;

в) використовувати дискову систему RAID. Хоча існує багато різних конфігурацій RAID, які можна створити, загалом найкраще працює наступне:

- 1) встановлення апаратного RAID-контролера (якщо можна);
- 2) операційна система та підмінний диск на одному наборі дисків, налаштованих як RAID-1;
- 3) Moodle, веб-сервер і сервер баз даних на іншому наборі дисків, налаштованих як RAID-5;

г) якщо область «moodleddata» буде знаходитися на відносно повільному сховищі (наприклад, монтувати NFS на пристрій NAS), виникнуть проблеми з продуктивністю з конфігурацією кешу за замовчуванням (який записує в це сховище). Рекомендується обрати альтернативу Redis. Використання GlusterFS / OCFS2 / GFS2 на пристрої SAN і Fibre Channel може підвищити продуктивність;

д) використовувати гігабітний Ethernet для покращення затримки та пропускної здатності. Це особливо важливо, коли веб-сервер і сервер бази даних розділені на різних хостах;

е) можна підвищити продуктивність, збільшивши використання буферів і дескрипторів передачі/приймання і вивантаживши обчислення контрольної суми TCP на карту замість ОС [19].

#### **2.1.1.2 Вимоги до програмного забезпечення**

Вимоги до програмного забезпечення. Рекомендовано підтримувати все програмне забезпечення та операційні системи в актуальному стані.

Це лише мінімальні підтримувані версії програмного забезпечення для Moodle:

- версія Moodle: Moodle 3.6 або новішої версії;

- версія PHP: Мінімальною підтримуваною версією є – PHP 7.3.0;
- рекомендується розширення PHP sodium;
- рекомендується розширення PHP exif;
- для встановлення PHP 7.x рекомендовано значення параметра `max_input_vars >= 5000`. Це вимога для встановлення PHP 8.x.

Нижче наведена таблиця з серверами баз даних, що підтримуються Moodle. У таблиці присутні нормери мінімальних версій, що підтримуються, але рекомендується використовувати саме останню версію будь-якої з наведеної бази даних [20].

Таблиця 2.1 – Базы даних для Moodle

| База даних           | Мінімальна версія              | Рекомендована версія |
|----------------------|--------------------------------|----------------------|
| PostgreSQL           | 10 (збільшено з Moodle 3.11)   | Остання              |
| MySQL                | 5.7                            | Остання              |
| MariaDB              | 10.2.29                        | Остання              |
| Microsoft SQL Server | 2017 (збільшено з Moodle 3.10) | Остання              |
| Oracle Database      | 11.2                           | Остання              |

### 2.1.2 Налаштування серверу

Залежно від потреб сервер можна встановити і використовувати на будь-якій машині: від віртуальних, або власних серверних кластерів до лише персонального комп'ютера. Існує багато можливостей для встановлення основного серверного програмного забезпечення [18]:

- встановлення AMP;
- IIS;
- Nginx;
- Apache.

### 2.1.3 Завантаження та копіювання файлів на місце

Хоча зараз є кілька місць, де можна отримати код Moodle, рекомендовано отримувати Moodle лише з moodle.org. Є два варіанти:

- завантажити необхідну версію з <http://moodle.org/downloads> розпакувати;
- витягнути код із репозиторію Git (рекомендується для розробників, а також робить оновлення дуже простим):  
\$ git clone -b MOODLE\_{{Version3}}\_STABLE git://git.moodle.org/moodle.git

Будь-яке з перерахованого вище має привести до каталогу під назвою moodle, який містить ряд файлів і каталогів.

Зазвичай можна розмістити весь каталог в каталозі документів вебсервера, і в цьому випадку сайт буде розміщено за адресою <http://yourwebserver.com/moodle>, або можна скопіювати весь вміст прямо в основний каталог документів вебсервера, у цьому випадку сайт буде просто <http://yourwebserver.com>.

Якщо завантажити Moodle на свій локальний комп'ютер, а потім завантажити його на розміщений вебсайт – зазвичай краще завантажити стиснений файл Moodle, а потім розпакувати його на розміщеному вебсайті. Якщо розпакувати Moodle на локальному комп'ютері, оскільки Moodle складається з понад 25 000 файлів, спроба завантажити понад 25 000 файлів за допомогою FTP-клієнта або «диспетчера файлів» хоста іноді може пропустити файл і викликати помилки.

Захист файлів Moodle: важливо, щоб користувачі вебсервера не могли записувати файли. Наприклад, на Unix/Linux (як root):

```
chown -R root /path/to/moodle
```

```
chmod -R 0755 /path/to/moodle
```

(файли належать адміністратору/суперкористувачу і доступні для запису лише ним – доступні для читання всіма іншими).

На сайтах тестування/розробника можна зробити файли доступними для запису, щоб використовувати вбудований інсталятор плагінів. Для живих сайтів це не рекомендується (принаймні, поверніться до більш безпечних налаштувань, якщо це зробити) [18].

#### **2.1.4 Створення пустої бази даних**

Далі створення нової порожньої бази даних для інсталяції. Потрібно знайти та записати таку інформацію для використання на останньому етапі встановлення:

- `dbhost` – ім'я хоста сервера бази даних. Ймовірно, `localhost`, якщо база даних і веб-сервер є однією машиною, інакше ім'я сервера бази даних;
- `dbname` – ім'я бази даних;
- `dbpass` – пароль для вищевказаного користувача.

Якщо сайт розміщено, слід знайти веб-сторінку адміністрування баз даних як частину панелі керування [18].

#### **2.1.5 Створення каталогу даних (moodledata)**

Moodle потрібен каталог для зберігання всіх його файлів. Веб-сервер повинен мати можливість запису в цей каталог. У великих системах потрібно враховувати, скільки вільного місця планується використовувати, виділяючи цей каталог.

Через стандартний спосіб кешування даних Moodle у можуть виникнути серйозні проблеми з продуктивністю, якщо використовувати відносно повільне сховище (наприклад, NFS) для цього каталогу.

Цей каталог не повинен бути доступним безпосередньо через Інтернет. Це була б серйозна діра в безпеці. Не можна розміщати його у своєму кореневому веб-сайті або в каталозі програмних файлів Moodle. Moodle не встановлюється.

Якщо використовувати розміщений сайт і немає іншого вибору, окрім як розмістити 'moodledata' в каталозі, доступному для інтернету. Каталог 'moodledata' можна буде захистити якщо створити файл під назвою .htaccess з такими рядками [18]:

- для Apache версією 2.2: "order deny,allow", "deny from all" [18];
- для Apache версією 2.4: "require all denied" [18].

### **2.1.6 Початок встановлення Moodle**

Після виконання попередніх етапів треба запустити програму встановлення, щоб створити таблиці бази даних та налаштувати новий сайт. Рекомендований метод – використовувати інсталятор командного рядка.

Найкраще запускати командний рядок як веб-користувач системи.

Приклад використання командного рядка [18]:

```
chown/path/moodle
```

```
cd /path/moodle/admin/cli
```

```
sudo -u /path/usr/bin/php install.php
```

```
chown -R path/moodle
```

chown дозволяють скрипту записати новий файл config.php. Додаткову інформацію про параметри можна знайти за допомогою [18]:

```
php install.php --help
```

### **2.1.7 Остаточна конфігурація**

На екранах адміністрування сайту Moodle є ряд опцій. Ось деякі з важливіших:

- Administration > Site administration > Server > Email > Outgoing mail configuration: налаштування свого smtp-серверу та аутентифікацію, якщо потрібно. Також можна встановити електронну адресу погеру на цій сторінці;

- Administration > Site administration > Server > Server > Support contact. Встановить свою контактну електронну адресу служби підтримки;
- Administration > Site administration > Server > System paths: встановити шляхи до двійкових файлів du, dot та aspell;
- Administration > Site administration > Server > HTTP: якщо перебувати за брандмауером, може знадобитися вказати облікові дані проксі в розділі «Веб-проксі»;
- Administration > Site administration > Location > Update timezones: потрібно запустити це, щоб переконатися, що інформація про часовий пояс актуальна.

Залишилися задачі:

- налаштувати Cron: фонові завдання Moodle виконуються за допомогою сценарію, який можна налаштувати на виконання в певний час дня. Це відомо як сценарій cron;
- збільшення максимального розміру завантаження;
- перевірити роботу пошти : з Site administration > Server > Test outgoing mail configuration скористатися посиланням, щоб надіслати собі тестовий електронний лист.

На цьому остаточна конфігурація закінчена[18].

## **2.2 Вибір мови програмування та технологій для реалізації backend-у програмного забезпечення**

### **2.2.1 PHP**

PHP (Hypertext Preprocessor) – це мова сценаріїв, вбудована в HTML. Більша частина його синтаксису запозичена з Java, Perl та C з додаванням кількох унікальних функцій, характерних для PHP. Мета мови – дозволити веб-розробникам швидко писати динамічно створювані сторінки.

Оскільки PHP є мовою сценаріїв на стороні сервера, вся обробка мови виконується на сервері. Тому клієнти отримують лише вихід. Це підвищує

безпеку. Також дуже важливою функцією PHP є забезпечення взаємодії з базою даних. На відміну від інших мов сценаріїв, PHP може обмінюватися будь-якою інформацією без клопоту [21].

Використання PHP у веброзробці може підвищити загальну швидкість сайту. PHP має власний простір пам'яті, тому це робить його одним із кращих варіантів для вебдизайну та розробки.

Ця мова програмування надає можливості та гнучкість створювати різні функції та включати якомога більше функцій у вебсайт.

Більшість розробників плагінів для Moodle використовують саме мову PHP. Оскільки основна система також написана на PHP і це спрощує її модифікацію та адаптацію.

### **2.2.2 NodeJS**

Node.js – це середовище виконання Javascript з відкритим вихідним кодом і бібліотека для виконання та запуску веб-програм поза браузером клієнта, засноване на V8, розробленому для веб-браузерів.

Node.js використовується для розробки та запуску веб-додатків із інтенсивним використанням операційної системи, таких як сайти потокового відео, односторінкові програми або програми для онлайн-чату. Це ідеальне середовище для додатків із великим обсягом даних, оскільки використовує однопотокową асинхронну модель, керовану подіями.

Але в цьому випадку основна частина документації Moodle відображає рішення для мови PHP. Тому вибір Node.js як мови для розробки плагіна для LMS Moodle є поганим рішенням [22].

### 2.2.3 Порівняльні таблиці Мов/Технологій

Порівняємо недоліки та переваги мов програмування у вигляді таблиці 2.2.

Таблиця 2.2 – Переваги та недоліки мов програмування

| Мова програмування | Переваги                                                                                                                                      | Недоліки                                                                                                                   |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| 1                  | 2                                                                                                                                             | 3                                                                                                                          |
| PHP                | PHP підтримує архітектуру модель-погляд-контролер і спрощує розробку та обслуговування вебдодатків [23].                                      | Оскільки окремі структури можуть відрізнятися з точки зору сили спільноти, їй бракує належної та швидкої підтримки [23].   |
|                    | Фреймворки PHP надають фрагменти коду, функції та інструменти, щоб допомогти розробникам прискорити розробку користувацьких вебдодатків [23]. | Замість PHP програміст повинен вивчити PHP-фреймворк, на що потрібен час і зусилля [23].                                   |
|                    | Більшість його фреймворків дозволяють програмістам використовувати широкий спектр реляційних баз даних [23].                                  | Оскільки він надає багато потужних інструментів і функцій, він часто впливає на швидкість і продуктивність вебсайтів [23]. |
|                    | Це дозволяє автоматизувати звичайні завдання веброзробки, заощаджуючи час і енергію [23].                                                     | Розробники позбавлені можливості змінювати основну поведінку фреймворків [23]                                              |
|                    | Ця мова програмування використовується як основна мова системи Moodle.                                                                        |                                                                                                                            |

Продовження таблиці 2.2

| 1      | 2                                                                                                                                              | 3                                                                                                                                                                     |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NodeJS | Він забезпечує легку масштабованість для сучасних додатків і використовується такими великими брендами, як PayPal, Uber і Netflix [23].        | Розробникам часто потрібно змінювати код доступу для синхронізації з найновішою версією API NodeJS [23].                                                              |
|        | Дає миттєві результати завдяки потужним додаткам [23].                                                                                         | Якщо додати значне робоче навантаження на основі ЦП у його цикл подій, продуктивність падає, оскільки він використовує всю потужність ЦП для виконання завдання [23]. |
|        | Розробники можуть писати коди на серверній і передній панелі, оскільки NodeJS надає їм можливість створювати код на стороні сервера [23].      | NodeJS користується великим попитом, але має мало досвідчених розробників [23].                                                                                       |
|        | Можна отримати підвищену швидкість розробки завдяки тому, що він легкий, що означає, що час виходу додатків на ринок значно скорочується [23]. | Багато з його бібліотек і реєстрів NPM є неповними або погано розробленими, що призводить до неефективного моніторингу розробки веб-додатків [23].                    |
|        | Його модуль кешування допомагає зменшити повторне виконання коду та зменшити навантаження на завдання [23].                                    |                                                                                                                                                                       |

Також порівняємо властивості мов програмування у вигляді таблиці 2.3.

Таблиця 2.3– Порівняння властивостей мов програмування

| Мова програмування | Особливості                                                                                                                                           | Популярні програми | Підтримка Moodle |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|------------------|
| 1                  | 2                                                                                                                                                     | 3                  | 4                |
| PHP                | Ви можете виконувати PHP-скрипти швидше, ніж сценарії, написані такими мовами, як ASP і JSP [23].                                                     | Wikipedia          | Так              |
|                    | Пропонує зручне кодування для програмістів із зрозумілим синтаксисом [23].                                                                            | MailChimp          |                  |
|                    | Він доступний безкоштовно. PHP є відкритим вихідним кодом.[23]                                                                                        | iStockPhoto        |                  |
|                    | Він не залежить від платформи і підтримує операційні системи Windows, Mac, UNIX і LINUX[23].                                                          |                    |                  |
|                    | Ця мова підтримує ODBC, SQLite, MySQL та інші основні бази даних [23].                                                                                |                    |                  |
| NodeJS             | Виконання коду в NodeJS відбувається швидко, оскільки він створений на движку JavaScript V8 Google Chrome [23].                                       | PayPal             | Так              |
|                    | NodeJS розгортається за ліцензією MIT [23].                                                                                                           | eBay               |                  |
|                    | Його програми не буферизують інформацію, і вони виводять дані лише частинами [23].                                                                    | Godaddy            |                  |
|                    | Він використовує однопотокову програму, яка також може запропонувати обслуговування більшої кількості запитів порівняно зі звичайними серверами [23]. |                    |                  |
|                    | Сервер на базі NodeJS не чекає повернення даних від API, усі його API асинхронні [23].                                                                |                    |                  |

## **2.3 Вибір мови програмування та технологій для реалізації frontend-у програмного забезпечення**

### **2.3.1 Мови програмування для Frontend-у**

#### **2.3.1.1 JavaScript**

JavaScript, часто скорочено JS, є мовою програмування, яка є основним засобом динамічного змінення веб-сторінки. Понад 97% веб-сайтів використовують JavaScript на стороні користувача для динамізації html-сторінок. Усі сучасні веб-браузери мають інтерпретацію цієї мови.

Ця мова високого рівня, відповідає стандарту ECMAScript. Він має динамічний введення тексту, орієнтацію на об'єкт на основі прототипу та першокласні функції. Це багато парадигмальний стандарт, який підтримує керовані події, функціональні та імперативний стилі програмування.

У стандарті ECMAScript немає жодних засобів введення/виводу (I/O), таких як мережа, зберігання чи графічні засоби.

Спочатку JavaScript був створений для того, щоб «оживити веб-документи». Набори кодів, написані цією мовою, називаються скриптами. Він може запускатися безпосередньо, коли сторінка завантажується безпосередньо в HTML-код. Команди виконуються безпосередньо без необхідності компіляції та підготовки.

JavaScript має приставку Java, але при цьому не має жодного відношення до цієї мови програмування. Синтаксисом і відповідними стандартними бібліотеками обидві мови суттєво відрізняються [24].

### 2.3.1.2 TypeScript

TypeScript є мовою програмування, розробленою та підтримуваною Microsoft. Це суворий синтаксичний наднабір JavaScript додає до мови не обов'язкове статичне введення. Він розроблений для полегшення сприйняття великих додатків на JavaScript. На виході компілятор створює той самий JavaScript, який потім виконується браузером [25].

TypeScript розвивається як opensource-проект і, як і багато проектів, хоститься на гітхабі.

Він є типізованою надмножиною JavaScript, а це означає, що будь-яка програма на JS є програмою на TypeScript.

TypeScript має незвичайне відношення до JavaScript. TypeScript пропонує всі функції JavaScript, а також додатковий шар поверх них: систему типів TypeScript. Наприклад, JavaScript надає мовні примітиви, такі як рядок і число, але він не перевіряє, чи їх постійно призначали.

Це означає, що існуючий робочий код JavaScript також є кодом TypeScript. Основна перевага TypeScript полягає в тому, що він може виділяти неочікувану поведінку у вашому коді, знижуючи ймовірність помилок.

За допомогою TypeScript ви можна створювати складні типи, комбінуючи прості.

Один із основних принципів TypeScript полягає в тому, що перевірка типу фокусується на формі, яку мають значення. Це іноді називають «качиним введенням» або «структурним типізацією».

### 2.3.1.3 CoffeeScript

CoffeeScript є це мовою програмування, яка компілюється на JavaScript. Вона додає синтаксичний цукор, натхненний Ruby, Python і Haskell, щоб

підвищити стислість і читабельність JavaScript. Спеціальні додаткові функції включають розуміння списку та призначення деструктуризації.

Підтримка CoffeeScript включена в Ruby on Rails версії 3.1 і Play Framework [26].

Золоте правило CoffeeScript: «Це просто JavaScript». Код компілюється один до одного в еквівалентний JS, і під час виконання немає інтерпретації. Можна використовувати будь-яку наявну бібліотеку JavaScript без проблем із CoffeeScript (і навпаки). Скомпільований результат є читабельним, добре надрукованим і має тенденцію працювати так само швидко або швидше, ніж еквівалент рукописного JavaScript [27].

#### 2.3.1.4 Порівняння мов програмування для Frontend-y

Порівнюємо переваги та недоліки мов програмування у вигляді таблиці 2.4.

Таблиця 2.4 – Порівняння мов програмування для Frontend-y

| Мова програмування | Переваги                                                                                                                                  | Недоліки                                                                                                                                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1                  | 2                                                                                                                                         | 3                                                                                                                                                                                                                                                      |
| JavaScript         | JavaScript порівняно простий у навчанні та виконанні. Будучи на стороні користувача, зменшується запит на головний сервер веб-сайту [28]. | Захист на стороні користувача іноді занадто складний для управління в Javascript. Код виконується на пристрої клієнта, тому його можна використовувати для зловмисних намірів. Це причина, чому деякі люди вважають за краще вимкнути Javascript [28]. |

Продовження таблиці 2.4

| 1          | 2                                                                                                                                                                                                                                                             | 3                                                                                                                                                                                                                 |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| JavaScript | JavaScript застосовується скрізь в Інтернеті. Джерела для вивчення JavaScript різноманітні [28].                                                                                                                                                              | JavaScript рідко визначається окремо різними браузерами. У той час як коди на стороні сервера завжди забезпечують ідентичний вихід, коди на стороні клієнта можуть бути трохи нерегулярними [28].                 |
|            | JavaScript точно працює з іншими мовами програмування і може застосовуватися у величезній різноманітності програм. На відміну від інших мов сценаріїв, таких як PHP, JavaScript можна вбудувати в будь-яку веб-сторінку, незважаючи на розширення файлу [28]. |                                                                                                                                                                                                                   |
| TypeScript | Основною перевагою Typescript є можливість комбінувати статичні типи з JS [28].                                                                                                                                                                               | Ця мова перевіряє типи лише шляхом компіляції. Зрештою, усе компілюється у простий JavaScript, який цього не робить. Це говорить про те, що все ще можливо знайти деякі недоліки, які компілятор не отримав [28]. |
|            | Typescript максимально надійно поєднує традиційне з новим. Можна застосувати додаткові функції з ES7, а компілятор TypeScript перетворить їх на стандарт ES5 [28].                                                                                            | Він може дати фальшиве відчуття безпеки [28].                                                                                                                                                                     |
|            | Вона простіше, ніж Javascript [28].                                                                                                                                                                                                                           |                                                                                                                                                                                                                   |

Продовження таблиці 2.4

| 1            | 2                                                                                                                                                                                                                                                                            | 3                                                                                                                                               |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| CoffeeScript | Код Coffeescript приблизно на 55% менше, ніж Javascript для еквівалентної функціональності. Це менше випадків, коли програмістам доводиться вводити та перевіряти [28].                                                                                                      | Хоча досить швидко оволодіти функціями CoffeeScript, у порівнянні з JS є багато обманених витрат часу, так як іноді його складно зібрати [28].  |
|              | Coffeescript використовує більш обмежений синтаксис, і тому, можливо, простіше керувати у великому проекті. Це також дозволяє використовувати більше синтаксичних заходів, і, таким чином, різні програмісти в одному проекті розробляють свій код набагато однорідним [28]. | CoffeeScript став дуже успішним за короткий проміжок часу, але сумнівно, що він коли-небудь матиме таке широке суспільство, як JavaScript [28]. |
|              |                                                                                                                                                                                                                                                                              | Синтаксичні помилки, які дає компілятор CoffeeScript, часто є проблематичними [28].                                                             |
|              |                                                                                                                                                                                                                                                                              | Деяка логіка розділу не працює належним чином, потрібно вивчити основний код JavaScript, щоб зрозуміти повністю таємничий код [28].             |

### 2.3.2 Мови розмітки, опису, та препроцесори зовнішнього вигляду для Frontend-у

#### 2.3.2.1 HTML

HTML – "мова розмітки гіпертексту", яка є основним статичним форматом відображення інформації для веб сайтів. Завжди використовується з CSS, яка дозволяє покращувати відображення інформації. Для динамізації цього формату використовують JavaScript. А для змінення контенту використовують технології Backend та Frontend, наприклад мову програмування php [29].

### 2.3.2.2 CSS

Каскадні таблиці стилів (CSS) – мова опису зовнішнього вигляду документа HTML. Є єдиним способом змінювати вигляд таблиць HTML, застосовувати анімацію.

Розроблено воно для підвищення якості сприйняття веб-сайтів, покращення їхнього зовнішнього вигляду.

Дозволяє відобразити практично будь-який задум дизайнера.

Також крім HTML дозволяє описувати зовнішній вигляд таких файлів як .XML, .XHTML, .XUL а також .SVG [30].

### 2.3.2.3 SASS

SASS – препроцесор css, який дозволяє використовувати вкладені правила, міксини, змінні, функції, успадкування та багато іншого. Сам він ніде не запускається, але його можна компілювати у файл формату CSS. Синтаксисом він нагадує python, він залежить від табуляції. Тому більшість програмістів все ж таки віддають перевагу SCSS [31].

### 2.3.2.4 SCSS

SCSS – це препроцесор, це означає просто програму, яка обробляє вхідні дані і виробляє вихід, а потім цей вихід буде використовуватися в іншій програмі. Sass має різні синтаксиси, тепер є два популярних, які називаються SCSS і SASS, тому ми маємо SCSS, що означає Scssy CSS, і це набір синтаксису CSS3, а SCSS більше схожий на посередника між SCSS і CSS, ніж у нас є фактичний Scss код, тому Scss буде просто повноцінним файлом SCSS, який буде скомпільовано в CSS [32].

### 2.3.2.5 Порівняння препроцесорів SCSS та SASS

Порівнюємо особливості та властивості препроцесорів за допомогою таблиці 2.5.

Таблиця 2.5 – Порівняння препроцесорів SCSS та SASS

| Властивості порівняння | SCSS                                                                                                                      | SASS                                                                                      |
|------------------------|---------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Визначення             | Sassy Cascaded Style Sheets                                                                                               | Syntactically Awesome Style Sheets                                                        |
| Використання           | Використовується, коли немає вимог або критеріїв щодо використовуваного синтаксису коду [33].                             | Використовується, коли для розробки потрібен оригінальний синтаксис [33].                 |
| Інтеграція             | Можна інтегрувати з будь-яким пакетом або проектом, оскільки він є доповненням до CSS, який містить усі функції CSS [33]. | Можна інтегрувати разом з будь-яким проектом, оскільки він підтримує всі версії CSS [33]. |
| Платформа              | Підтримує будь-які операційні системи або платформи [33].                                                                 | Підтримує будь-які операційні системи або платформи [33].                                 |
| Синтаксис              | Подібний до CSS (використовує фігурні дужки та крапку з комою)[34]                                                        | Використовує строгий відступ (як наприклад у Python) [34].                                |
| Підтримка .css         | Будь-який дійсний CSS є дійсним SCSS [34].                                                                                | Код CSS не можна використовувати як SASS [34].                                            |
| Розширення             | Розширення .scss                                                                                                          | Розширення .sass                                                                          |

## 2.4 Модель розробки та основні принципи розроблення та інтегрування плагінів

### 2.4.1 Основні принципи розробки плагінів

М в Moodle означає модульний. Найпростіший і найдоступніший спосіб додати нові функції до Moodle – це створити один із цих типів плагінів [35].

Плагіни дозволяють додавати додаткові функції та функції до Moodle, наприклад, нові дії, нові типи питань вікторини, нові звіти, інтеграцію з іншими системами та багато іншого [37].

Плагіни пишуть на основі коду, що вказаний у директорії moodle. Можна скопіювати файли на шляху відповідно до типу плагіна, який планується розробити. Далі вони модифікуються, запаковуються у .zip архів. Після цього їх можна встановити у Moodle.

Список доступних типів плагінів Moodle у вигляді таблиці 2.6.

Таблиця 2.6 – Список доступних типів плагінів для Moodle

| Тип плагіна                 | Назва компонента | Шлях Moodle            | Опис                                                                                                         | Версії Moodle |
|-----------------------------|------------------|------------------------|--------------------------------------------------------------------------------------------------------------|---------------|
| 1                           | 2                | 3                      | 4                                                                                                            | 5             |
| Модулі діяльності           | mod              | /mod                   | Модулі активності є важливими типами плагінів у Moodle, оскільки вони забезпечують діяльність у курсах [35]. | 1.0+          |
| Антивірусні плагіни         | antivirus        | /lib/antivirus         | Плагіни антивірусного сканера [35].                                                                          | 3.1+          |
| Плагіни для подання завдань | assignsubmission | /mod/assign/submission | Різні форми подання завдань [35].                                                                            | 2.3+          |
| Плагіни зворотного зв'язку  | assignfeedback   | /mod/assign/feedback   | Різні форми зворотного зв'язку з завданнями [35].                                                            | 2.3+          |
| Книжкові інструменти        | booktool         | /mod/book/tool         | Невеликі інформаційні дисплеї або інструменти, які можна переміщати по сторінках [35].                       | 2.1+          |
| Спеціальні поля курсу       | customfield      | /customfield/field     | Спеціальні типи полів, що використовуються е. г. в полях спеціального курсу [35].                            | 3.7+          |
| Поля бази даних             | datafield        | /mod/data/field        | Різні типи даних, які можна додати до модуля активності бази даних [35].                                     | 1.6+          |

Продовження таблиці 2.6

| 1                                 | 2             | 3                           | 4                                                                                                                                                 | 5    |
|-----------------------------------|---------------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|------|
| Попередні налаштування бази даних | datapreset    | /mod/data/preset            | Попередньо визначені шаблони для модуля активності бази даних [35].                                                                               | 1.6+ |
| Конвертери файлів                 | fileconverter | /files/converter            | Дозволити перетворення між різними типами файлів, надісланих користувачем. Наприклад, з .doc в PDF [35].                                          | 3.2+ |
| Послуги LTI                       | ltiservice    | /mod/lti/service            | Дозволяє реалізовувати послуги LTI, як описано в специфікації IMS LTI [35].                                                                       | 2.8+ |
| Бекенди машинного навчання        | mlbackend     | /lib/mlbackend              | Процесори прогнозування для API аналітики [35].                                                                                                   | 3.4+ |
| Звіти форуму                      | forumreport   | /mod/forum/report           | Відображати різні звіти в діяльності форуму [35].                                                                                                 | 3.8+ |
| Звіти вікторини                   | quiz          | /mod/quiz/report            | Відображайте та аналізуйте результати вікторини або просто підключайте різну поведінку в модуль вікторини [35].                                   | 1.1+ |
| Правила доступу до вікторини      | quizaccess    | /mod/quiz/accessrule        | Додайте умови до того, коли чи де можна спробувати пройти тести, наприклад, лише з деяких IP-адрес, або учень повинен спочатку ввести пароль [35] | 2.2+ |
| SCORM повідомлення                | scormreport   | /mod/scorm/report           | Аналіз повідомлень SCORM [35].                                                                                                                    | 2.2+ |
| Блоки                             | block         | /blocks                     | Невеликі інформаційні дисплеї або інструменти, які можна переміщати по сторінках [35].                                                            | 2.0+ |
| Типи запитань                     | qtype         | /question/type              | Різні типи запитань (наприклад, з кількома варіантами, перетягуванням), які можна використовувати у вікторинах та інших видах діяльності [35].    | 1.6+ |
| Питання поведінки                 | qbehaviour    | /question/behaviour         | Контролюйте, як студент взаємодіє із запитаннями під час спроби [35].                                                                             | 2.1+ |
| Формати імпорту/експорту запитань | qformat       | /question/format            | Імпорт і експорт визначень запитань до/з банку запитань [35].                                                                                     | 1.6+ |
| Текстові фільтри                  | filter        | /filter                     | Автоматично перетворювати, виділяти та транслювати текст, опублікований в Moodle [35].                                                            | 1.4+ |
| Редактори                         | editor        | /lib/editor                 | Альтернативні текстові редактори для редагування вмісту [35]                                                                                      | 2.0+ |
| Плагіни редактора Atto            | atto          | /lib/editor/atto/plugins    | Додаткова функціональність для текстового редактора Atto [35].                                                                                    | 2.7+ |
| Плагіни редактора TinyMCE         | tinymce       | /lib/editor/tinymce/plugins | Додаткова функціональність для текстового редактора TinyMCE [35].                                                                                 | 2.4+ |
| Плагіни реєстрації                | enrol         | /enrol                      | Способи контролю зарахування на курси [35].                                                                                                       | 2.0+ |
| Плагіни аутентифікації            | auth          | /auth                       | Дозволяє підключатися до зовнішніх джерел аутентифікації [35].                                                                                    | 2.0+ |
| Інструменти адміністратора        | tool          | /admin/tool                 | Надає допоміжні скрипти, корисні для різних завдань адміністрування та обслуговування сайту [35].                                                 | 2.2+ |

Продовження таблиці 2.6

|                                  |              |                         |                                                                                                                    |      |
|----------------------------------|--------------|-------------------------|--------------------------------------------------------------------------------------------------------------------|------|
| Магазини деревини                | logstore     | /admin/tool/log/store   | Сховище журналів подій [35]                                                                                        | 2.7+ |
| Умови доступності                | availability | /availability/condition | Умови обмеження доступу користувачів до діяльності та розділів [35]                                                | 2.7+ |
| Типи календарів                  | calendartype | /calendar/type          | Визначає, як дати відображаються в Moodle [35]                                                                     | 2.6+ |
| Повідомлення споживачам          | message      | /message/output         | Представляти різні цілі, куди можна надсилати повідомлення та сповіщення (електронна пошта, sms, jabber, ...) [35] | 2.0+ |
| Формати курсу                    | format       | /course/format          | Різні способи розміщення завдань і блоків у курсі [35]                                                             | 1.3+ |
| Формати даних                    | dataformat   | /dataformat             | Формати для експорту та завантаження даних [35]                                                                    | 3.1+ |
| Поля профілю користувача         | profilefield | /user/profile/field     | Додайте нові типи даних до профілів користувачів [35]                                                              | 1.9+ |
| Звіти                            | report       | /report                 | Забезпечує корисні перегляди даних на сайті Moodle для адміністраторів і викладачів [35]                           | 2.2+ |
| Звіти про курси                  | coursereport | /course/report          | Звіти про діяльність у рамках курсу [35]                                                                           | 2.1+ |
| Експорт оцінок                   | gradeexport  | /grade/export           | Експортувати оцінки в різних форматах [35]                                                                         | 1.9+ |
| Імпорт оцінок                    | gradeimport  | /grade/import           | Імпорт оцінок у різних форматах [35]                                                                               | 1.9+ |
| Звіти про оцінки                 | gradereport  | /grade/report           | Відображати/редагувати оцінки в різних макетах і звітах [35]                                                       | 1.9+ |
| Розширені методи оцінювання      | gradingform  | /grade/grading/form     | Інтерфейси для фактичного виконання оцінювання в модулях діяльності (наприклад, Рубрики)[35]                       | 2.2+ |
| Протоколи веб-сервісів           | webservice   | /webservice             | Визначте нові протоколи для зв'язку веб-сервісів (такі як SOAP, XML-RPC, JSON, REST ...) [35]                      | 2.0+ |
| Плагіни репозиторію              | repository   | /repository             | Підключайтеся до зовнішніх джерел файлів для використання в Moodle [35]                                            | 2.0+ |
| Плагіни портфоліо                | portfolio    | /portfolio              | Підключіть зовнішні служби портфоліо як місця призначення для зберігання вмісту Moodle [35]                        | 1.9+ |
| Пошукові системи                 | search       | /search/engine          | Сервери пошукової системи для індексації вмісту Moodle [35]                                                        | 3.1+ |
| Медіаплеєри                      | media        | /media/player           | Підключаються медіаплеєри [35]                                                                                     | 3.2+ |
| Плагіни для перевірки на плагіат | plagiarism   | /plagiarism             | Визначте зовнішні служби для обробки надісланих файлів і вмісту [35]                                               | 2.0+ |
| Менеджер кеша                    | cachestore   | /cache/stores           | Сховище кешу [35]                                                                                                  | 2.4+ |
| Блокування кешу                  | cachelock    | /cache/locks            | Реалізація блокування кешу [35]                                                                                    | 2.4+ |
| Теми                             | theme        | /theme                  | Змініть вигляд Moodle, змінивши HTML і CSS [35]                                                                    | 2.0+ |

Продовження таблиці 2.6

| 1                              | 2           | 3                        | 4                                                                                                                 | 5         |
|--------------------------------|-------------|--------------------------|-------------------------------------------------------------------------------------------------------------------|-----------|
| Локальні плагіни               | local       | /local                   | Загальні плагіни для локальних налаштувань [35]                                                                   | 2.0+      |
| Застарілі типи призначення     | assignment  | /mod/assignment/type     | Різні форми завдань, які оцінюються вчителями [35]                                                                | 1.x - 2.2 |
| Застарілі звіти адміністратора | report      | /admin/report            | Забезпечує корисні перегляди даних на сайті Moodle лише для адміністраторів [35]                                  | 2.1+      |
| Типи вмісту банку вмісту       | contenttype | /contentbank/contenttype | Типи вмісту для завантаження, створення або редагування в банк вмісту та використання на всьому сайті Moodle [35] | 3.9+      |
| Бібліотеки H5P                 | h5plib      | /h5p/h5plib              | Тип плагіна для окремих версій інтеграційної бібліотеки H5P [35]                                                  | 3.9+      |
| Плагіни банку запитань         | qbank       | /question/bank           | Тип плагіна для розширення функціональності банку запитань [35]                                                   | 4.0+      |

Нижче наведено список файлів, які однаково працюють у всіх типах плагінів:

- `version.php` – Тут визначаються метадані плагіна (наприклад, номер версії, залежності або рівень зрілості) [36];
- `lang/en/pluginname_pluginname.php` – Англійські рядки для плагіна визначаються тут. Повинен бути присутній принаймні `$string['pluginname']`. Існує виняток для модулів Activity, їх очікуване ім'я файлу – це просто `pluginname.php` (без префікса `mod_`) [36];
- `lib.php` – Тут визначено інтерфейс між ядром Moodle і плагіном для більшості типів плагінів. Очікуваний вміст файлу залежить від конкретного типу плагіна. Ядро Moodle часто (але не завжди) завантажує всі файли `lib.php` зазначених типів плагінів. З міркувань продуктивності настійно рекомендовано, щоб цей файл був якомога меншим і просто впровадити в нього необхідний код. Вся внутрішня логіка плагіна має бути реалізована в автоматично завантажених класах [36];
- `locallib.php` – Використання цього файлу більше не рекомендується, і не буде дозволено в основному коді. Замість того, щоб створювати глобальну функцію в глобальному просторі імен у файлі `locallib.php`, слід використовувати автоматично завантажені класи, які знаходяться в каталозі `classes/`. У ядрі Moodle не буде дозволено додавати нові модифікації до цього файлу [36];

- `db/install.xml` – Тут визначається схема бази даних плагінів (таблиці, поля, індекси та ключі). Завжди потрібно використовувати вбудований редактор XML DB для створення цього файлу [36];
- `db/upgrade.php` – Тут визначаються етапи оновлення (наприклад, зміни схеми бази даних та інші речі, які мають відбутися під час оновлення плагіна). Вбудований редактор XMLDB можна використовувати для створення коду для зміни схеми бази даних [36];
- `db/access.php` – Дадає функції, щоб можна було визначити, що дозволено робити поточному користувачеві. Це також дозволяє модулям розширювати Moodle новими можливостями [36];
- `db/install.php` – Дозволяє виконувати PHP-код відразу після встановлення схеми бази даних плагіна [36];
- `db/uninstall.php` – Дозволяє виконувати PHP-код до того, як таблиці бази даних і дані плагіна будуть видалені під час видалення плагіна [36];
- `db/events.php` – Тут визначаються обробники подій (підписки). У ньому перераховані всі події, які ваш плагін хоче спостерігати та отримувати сповіщення [36];
- `db/messages.php` – Дозвольте оголосити плагін постачальником повідомлень [36];
- `db/services.php` – Тут описано зовнішні функції та вебсервіси, які надає ваш плагін [36];
- `db/renamedclasses.php` – Деталі класів, які були перейменовані, щоб відповідати автозавантаженню [36];
- `classes/` – Автоматичного завантажувач класів, наданого Moodle [36];
- `cli/` – У коді Moodle передбачено, що всі сценарії CLI розміщуються в папці `cli/` всередині каталогу плагінів [36];
- `settings.php` – Описує конфігурацію адміністрування плагіна [36];
- `amd/` – Модулі Javascript плагіна, написані у форматі AMD або ES6. (AMD, підтримується з Moodle 2.9, ES6 з Moodle 3.8) [36];

- `yui/` – Модулі YUI плагіна (рекомендований спосіб включення Javascript в Moodle 2.8 і старіші) [36];
- `jquery/` – Модулі jQuery плагіна (Moodle 2.8 і старіші) [36];
- `styles.css` – CSS плагіна зберігається тут [36];
- `pix/icon.svg` – Значок плагіна. Модулі активності мають забезпечувати кольорові піктограми розміром 24 x 24 пікселі. Усі інші файли плагінів мають бути представлені монохромним значком розміром 16 x 16 пікселів. Повинна бути резервна версія значка `pix/icon.png` для застарілих браузерів [36];
- `thirdpartylibs.xml` – У файлі має бути перераховано всі сторонні бібліотеки, які входять до комплекту плагіна [36];
- `readme_moodle.txt` – Файл повинен містити детальні інструкції щодо імпорту сторонніх бібліотек до Moodle. Це має нарахувати: "Завантажити URL-адреси" та "Інструкції по збірці" [36];
- `environment.xml` – Плагін може оголошувати свої додаткові вимоги до середовища за допомогою цього файлу – наприклад, обов'язкова наявність деяких необов'язкових розширень PHP [36];
- `README` – Файл `README.md` або `README.txt` повинен містити корисну інформацію про плагін. В ідеалі він повинен діяти як автономна версія всієї інформації, яка доступна на сторінці плагіна в каталозі плагінів [36];
- `CHANGES` – Якщо файл `CHANGES.md` (або `CHANGES.txt`, `CHANGES.html` або просто `CHANGES`) знайдено під час завантаження нової версії плагіна в каталог `Plugins`, тоді вміст цього файлу використовується для попереднього заповнення поля приміток до випуску для нова версія [36].

Завдяки опису цих файлів можна написати свій плагін.

## 2.4.2 Встановлення плагінів у систему Moodle

Щоб встановити плагін, його вихідний код необхідно розмістити (розгорнути) у відповідному місці в каталозі встановлення Moodle і відвідати головну сторінку адміністрування Адміністрування > Адміністрування сайту > Сповіщення. BLMSMoodle є лише три способи встановити плагіни.

Першим способом є встановлення разом із Moodle. Другим – встановлення через каталог плагінів Moodle. Третім – встановлення плагіна у форматі zip. Веб-сервер повинен мати доступ для запису до каталогу типу плагіна, куди має бути встановлений новий плагін, щоб використовувати будь-який із цих методів.

Крім того, плагін можна розгорнути вручну на сервері [37].

Встановлення за допомогою каталогу плагінів у системі LMSMoodle:

- уролі адміністратора перейти до Адміністрування сайту > Плагіни > Встановити плагіни;
- натиснути кнопку «Встановити плагіни з каталогу плагінів Moodle»;
- обрати свою поточну версію Moodle, потім знайти плагін за допомогою кнопки «Встановити», натиснути кнопку «Встановити», а потім натиснути «Продовжити»;
- підтвердити запит на встановлення;
- перевірити звіт про перевірку плагіна.

Встановлення через завантажений ZIP-файл:

- перейти до каталогу плагінів Moodle, обрати поточну версію Moodle, потім обрати плагін за допомогою кнопки «Завантажити» та завантажити ZIP-файл;
- увійти на свій сайт Moodle як адміністратор і перейти до Адміністрування > Адміністрування сайту > Плагіни > Встановити плагіни;

- завантажити ZIP-файл. Буде запропоновано додати додаткові відомості (у розділі Показати більше), лише якщо плагін не виявляється автоматично;
- якщо цільовий каталог недоступний для запису, буде виведено попередження;
- перевірити звіт про перевірку плагіна.

Встановлення вручну на сервері.

Якщо не можливо розгорнути код плагіна через вебінтерфейс адміністрування, доведеться скопіювати його в файлову систему сервера вручну. Встановіть правильне місце в кодовому дереві Moodle для типу плагіна [37].

## 2.5 Підсумки розділу 2

Розглянуто способи розгорнення системи Moodle та вимоги до системи Moodle. Також були розглянуті та зіставлені мови програмування для backend-у та frontend-у у табличній формі. Описані такі технології, як HTML, CSS. Також розглянуті та порівняні у вигляді таблиці препроцесори: SCSS, SASS.

Розглянуто основні принципи розробки плагінів, список доступних типів плагіну, список однакових файлів у всіх файлів, спосіб інсталяції плагінів у систему Moodle.

Для backend-а плагіна, що розробляється, була обрана мова PHP у зв'язку з тим, що ця мова використовується Moodle за замовчуванням.

Для frontend-а плагіна, що розробляється, була обрана мова програмування JavaScript у зв'язку з тим, що ця мова є відносно простою в реалізації та навчанні. Також по суті він є оригіналом, оскільки TypeScript та CoffeeScript у результаті компілюються у JavaScript файл.

Серед мов опису зовнішнього вигляду було обрано препроцесор SCSS. Тому що за синтаксисом він дуже нагадує CSS, але при цьому значно розширює функціонал використання. SASS не був обраний через незручність. Щоб

спростити сприйняття він не використовує дужки, натомість він використовує відступи для відмінності між вкладеними класами (як це зроблено в Python). Таке рішення є не звичним для більшості програмістів, хто писав на CSS.

### 3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

#### 3.1 Алгоритм рецензування курсу

Рецензування курсу потрібно для визначення, чи можна використовувати даний курс для навчання студентів. Відбувається це так.

Крок 1. Створюється курс.

Крок 2. Далі створений курс редагується.

Крок 3. Після редагування курсу необхідно поставити `block_course_review_vote`.

Крок 4. Далі на необхідний курс додаються користувачі, які цей курс рецензуватимуть.

Крок 5. Далі користувачі залишають свою рецензію (можуть залишити лише оцінку за шкалою від -2 балів до 2, а можуть також написати свою замітку з приводу курсу (якщо в налаштуваннях адміністратор дозволив їх залишати));

Крок 6. Далі буде 2 варіанти:

- якщо адміністратор вирішує, що курс пройшов рецензування – він видаляє рецензентів з курсу і видаляє блок з курсу. Після цього зараховує на курс студентів;

- якщо адміністратор вирішує, що курс не пройшов рецензування – то він очищає дані рецензування з цього курсу в `course_review` звіті та повертається до кроку 5.

Цей алгоритм можна представити як блок схему (рисунок 3.1).

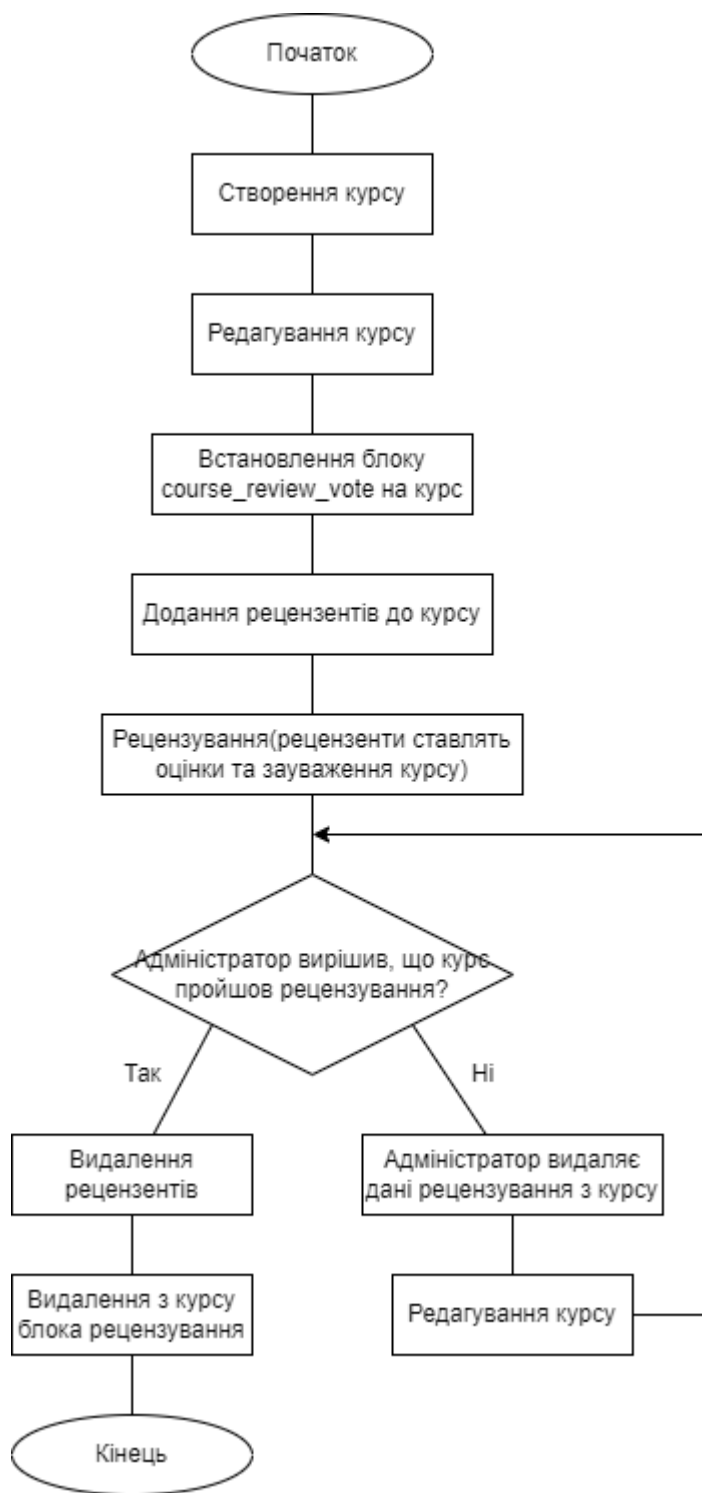


Рисунок 3.1 – Діаграма алгоритму рецензування курсу

### 3.2 Основні модулі розробленого додатка

Система рецензування складається з плагіна-блоку, та плагіна-звіту. Мета плагіна-блоку – збирати рецензії. Мета плагіна-звіту – виводити інформацію для адміністратора за курсами, а також надавати способи

адміністрування рецензій. Для рецензування необхідно встановити обидва плагіни в послідовності: `block_course_review_vote` потім `report_course_review`.

### 3.2.1 Плагін-блок

Плагін-блок має таку файлову архітектуру (рисунок 3.2).

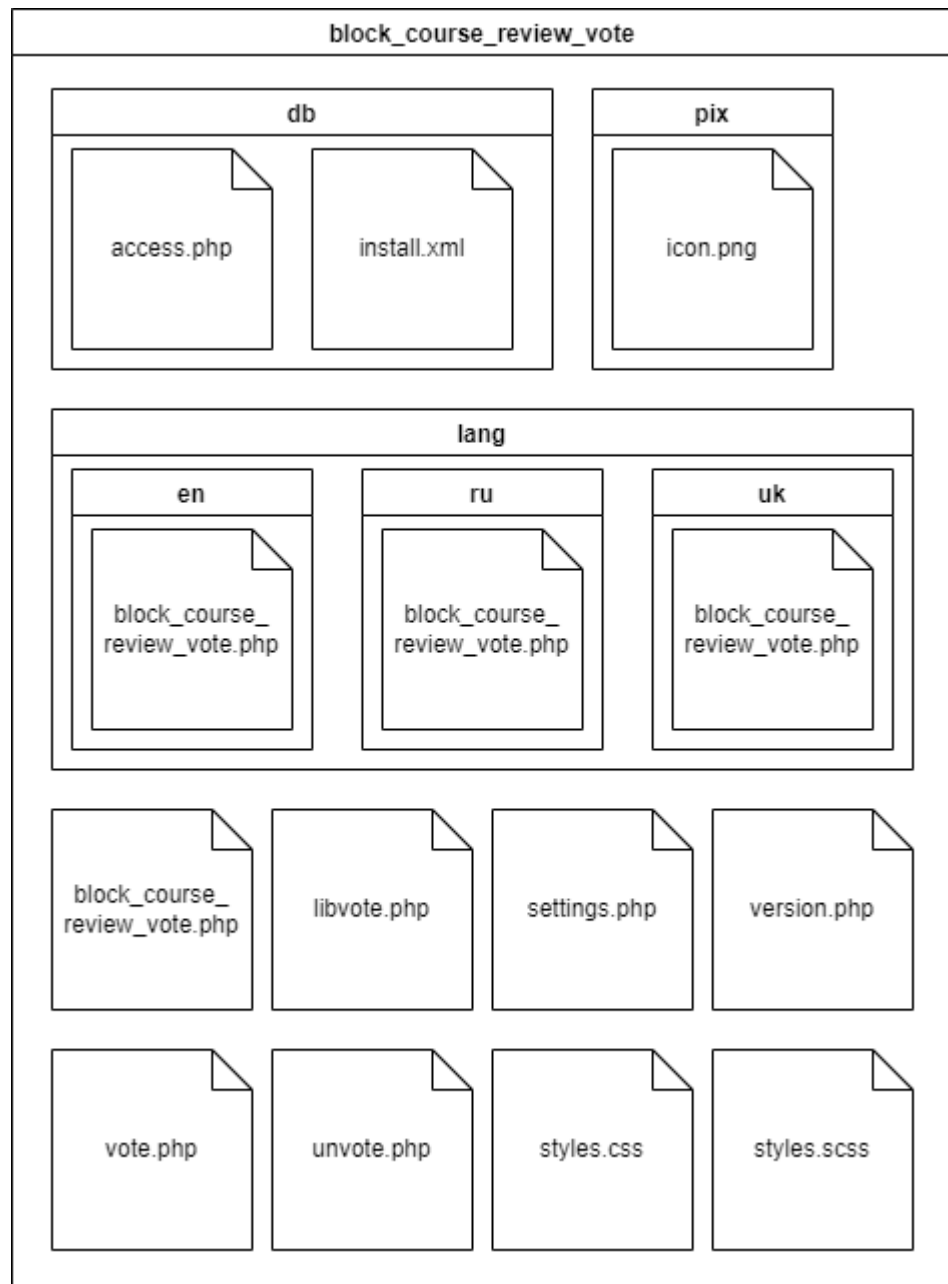


Рисунок 3.2 – Діаграма файлової архітектури плагіну-блоку

Детальний опис файлів:

- "db\access.php" – містить у собі налаштування щодо ролей користувачів. У цьому випадку дозволи для адміністратора;
- "db\install.xml" – містить налаштування створення таблиці, в якій будуть зберігатися рецензії;
- "lang\en\block\_course\_review\_vote" – містить рядки для локалізації блоку "block\_course\_review\_vote" на мову "en"(англійська);
- "lang\ru\block\_course\_review\_vote" – містить рядки для локалізації блоку "block\_course\_review\_vote" на мову "ru"(російський);
- "lang\uk\block\_course\_review\_vote" – містить рядки для локалізації блоку "block\_course\_review\_vote" на мову "uk"(українська);
- "pix\icon.png" – є іконкою плагін;
- "block\_course\_review\_vote.php" – основний файл плагіна, містить у собі те – що відображатиме бло;
- "libvote.php" – додатковий файл, здебільшого містить у собі функції відображення контенту, викликані в "block\_course\_review\_vote.php";
- "vote.php" – файл, який дозволяє виконувати додавання до бази даних рецензії;
- "unvote.php" – файл, який дозволяє переводити в тип "Видалені" певну рецензі;
- "settings.php" – файл, який містить конфігурацію плагіна. У ньому міститься лише дозвіл залишати нотатки (яка за замовчуванням увімкнена);
- "styles.scss" – файл препроцесора SCSS, який містить стилі для плагін;
- "styles.css" – конвертований файл опису зовнішнього вигляду із SCSS, в якому містяться стилі для плагіна-блок;
- "version.php" – подробиці про плагін (версія, назва тощо).

### 3.2.2 Плагін-звіт

Плагін-звіт має таку файлову архітектуру (рисунок 3.3).

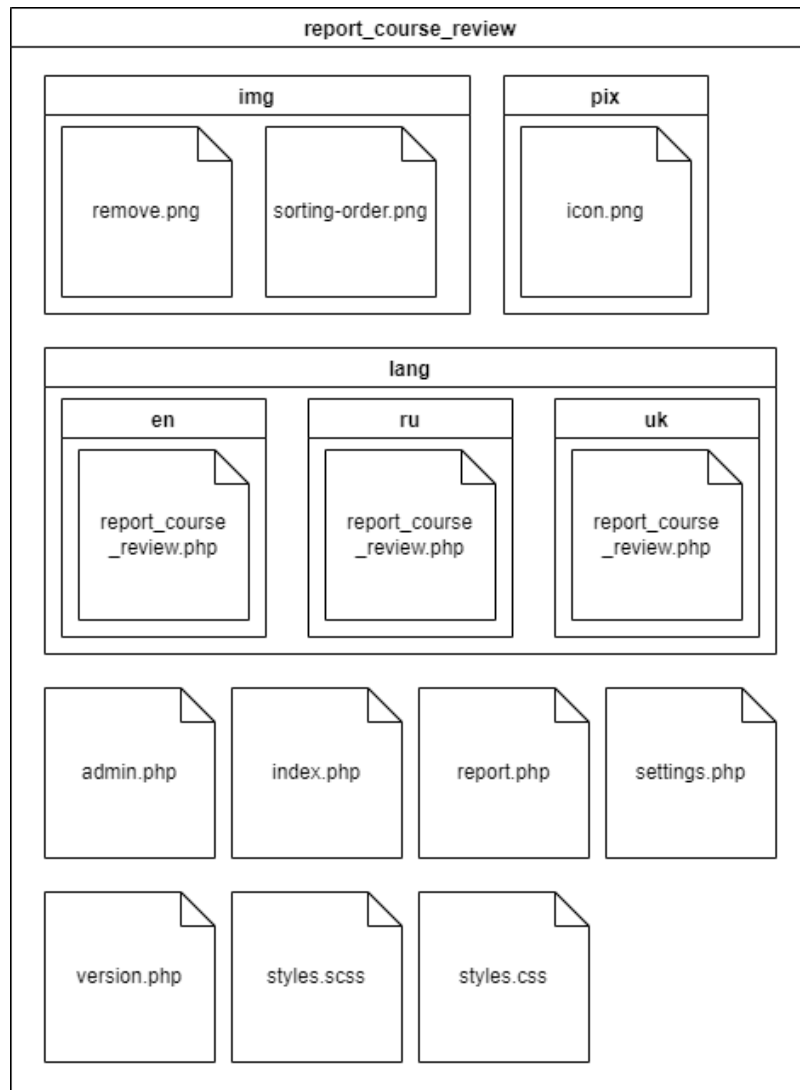


Рисунок 3.3 – Діаграма файлової архітектури плагіну-звіту

Детальний опис файлів:

- "img\remove.png" – зображення, яке використовується для видалення рецензій у таблиці огляду курсу;
- "img\sorting-order.png" – зображення відображається, по якій колонці сортується таблиця;
- "lang\en\report\_course\_review" – містить рядки для локалізації звіту "report\_course\_review" на мову "en"(англійська;

- "lang\ru\report\_course\_review" – містить рядки для локалізації звіту "report\_course\_review" на мову "ru"(російська);
- "lang\uk\report\_course\_review" – містить рядки для локалізації звіту "report\_course\_review" на мову "uk"(українська);
- "pix\icon.png" – є іконкою плагіна;
- "admin.php" – файл для дій адміністратора. Дозволяє видаляти рецензії, видаляти всі нотатки, видаляти всі існуючі рецензії, видаляти всі видалені рецензії назавжди, видаляти всі рецензії;
- "index.php" – у цьому файлі відображаються основна сторінка керування адміністратором;
- "report.php" – файл, що відображає звіт залежно від типу звіту, та сортування;
- "settings.php" – файл, який містить конфігурацію плагіна. У ньому міститься дозвіл використовувати інструменти адміністратора;
- "styles.scss" – файл препроцесора SCSS, який містить стилі для плагіна;
- "styles.css" – конвертований файл опису зовнішнього вигляду із SCSS, в якому містяться стилі для плагіна-звіту;
- "version.php" – подробиці про плагін (версія, необхідні плагіни, назва тощо).

### 3.3 Підсумки розділу 3

Для реалізації системи рецензування було два доповнення з архітектурною версією для адаптації в Moodle. Кожен додаток містить стилі (styles.css), налаштування (settings.php), файли властивостей додатка (version.php), іконку додатка (icon.png), 3 каталоги взаємодії інтерфейсу для англійської, російської, української мов.

Плагін-блок містить також файли для доступу до даних, і створює базу даних. Також він містить у собі файли відображення блоку, бібліотеки функцій, файл реакцію видалення рецензії і файл реакцію видалення рецензії.

Плагін-звіт містить додаткові 2 зображення для застосування в зображенні, файл реакції на дію адміністратора, файл відображення головного меню адміністратора, файл для складання звіту для адміністратора.

## 4 ОПИС ПРОГРАМНОГО ДОДАТКУ

### 4.1 Функціональні можливості розробленої системи рецензування

#### 4.1.1 Плагін-блок

Плагін має 2 прецеденти системи:

- адміністратор/Менеджер;
- рецензент (будь-який інший користувач).

Побудована діаграма прецедентів (рисунок 4.1).

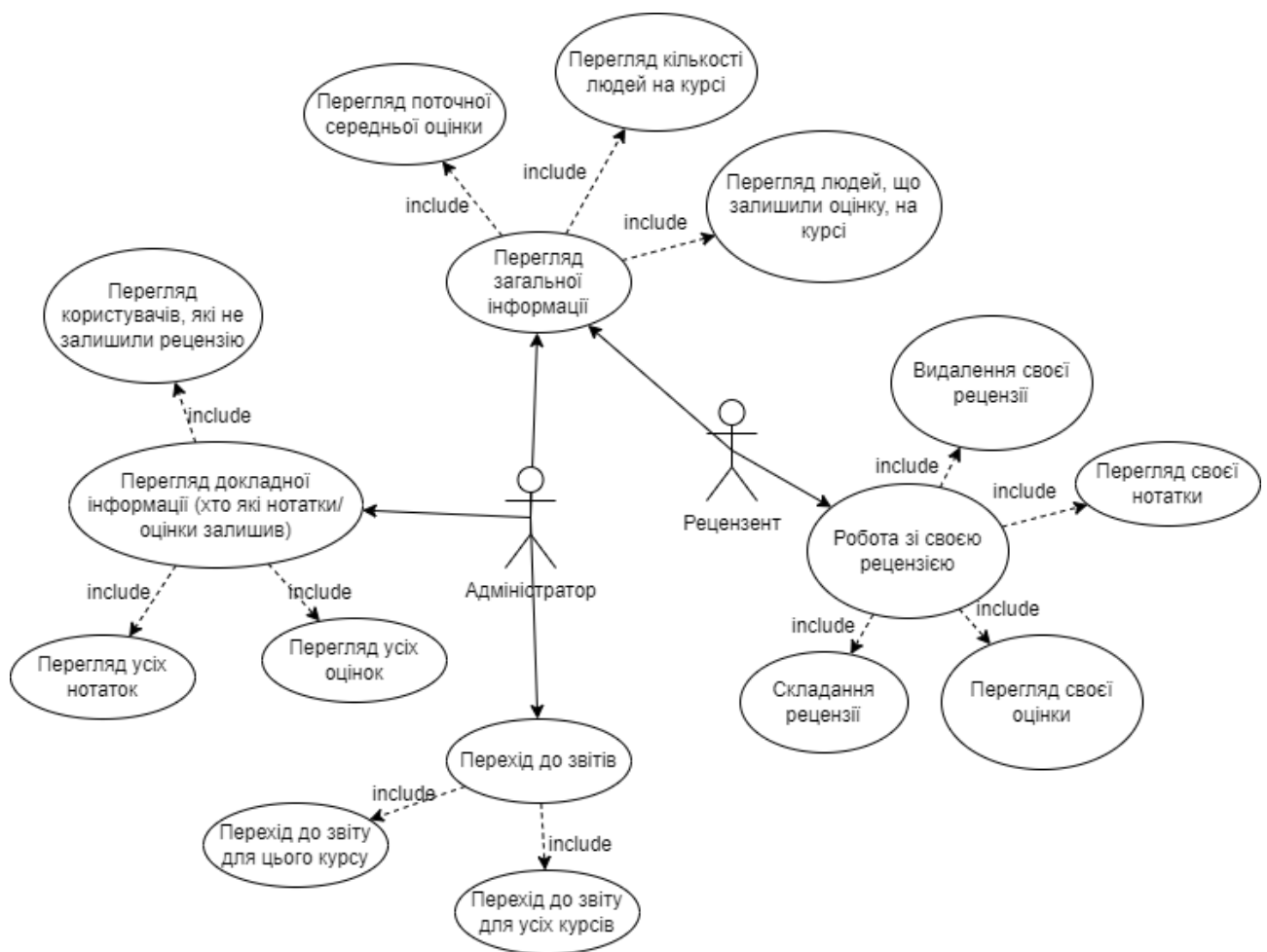


Рисунок 4.1 – Діаграма прецедентів системи для плагіну-блоку

Функції для ролі "рецензентів":

- редаговане поле для залишення нотатки (якщо це дозволено адміністратором);

- виставлення оцінки: -2, -1, 0, 1, 2. при натисканні на яку відправляється рецензія (разом із заміткою, якщо вона написана);
- перегляд поставленої їм оцінки;
- перегляд залишеної ним заміткою (якщо вона є);
- пропонується видалити рецензію (якщо вона написана).

Функції для ролі "адміністратора":

- перегляд користувачів, які не поставили рецензію;
- перегляд оцінок, та користувачів, яких їх поставили;
- перегляд нотаток, та користувачів, яких їх написали;
- перехід по облікових записах користувачів;
- перехід до докладного звіту за цей курс (необхідно, щоб стояв другий плагін);
- перехід до всіх функцій адміністратора (необхідно, щоб стояв другий плагін).

Функції для всіх користувачів:

- відображення поточної середньої оцінки;
- відображення кількості користувачів, що залишили рецензію;
- відображення кількості користувачів на курсі.

#### **4.1.2 Плагін-звіт**

Плагін має лише одного прецеденту системи:

- Адміністратор/Менеджер.

Побудована діаграма прецедентів рис. 4.2.

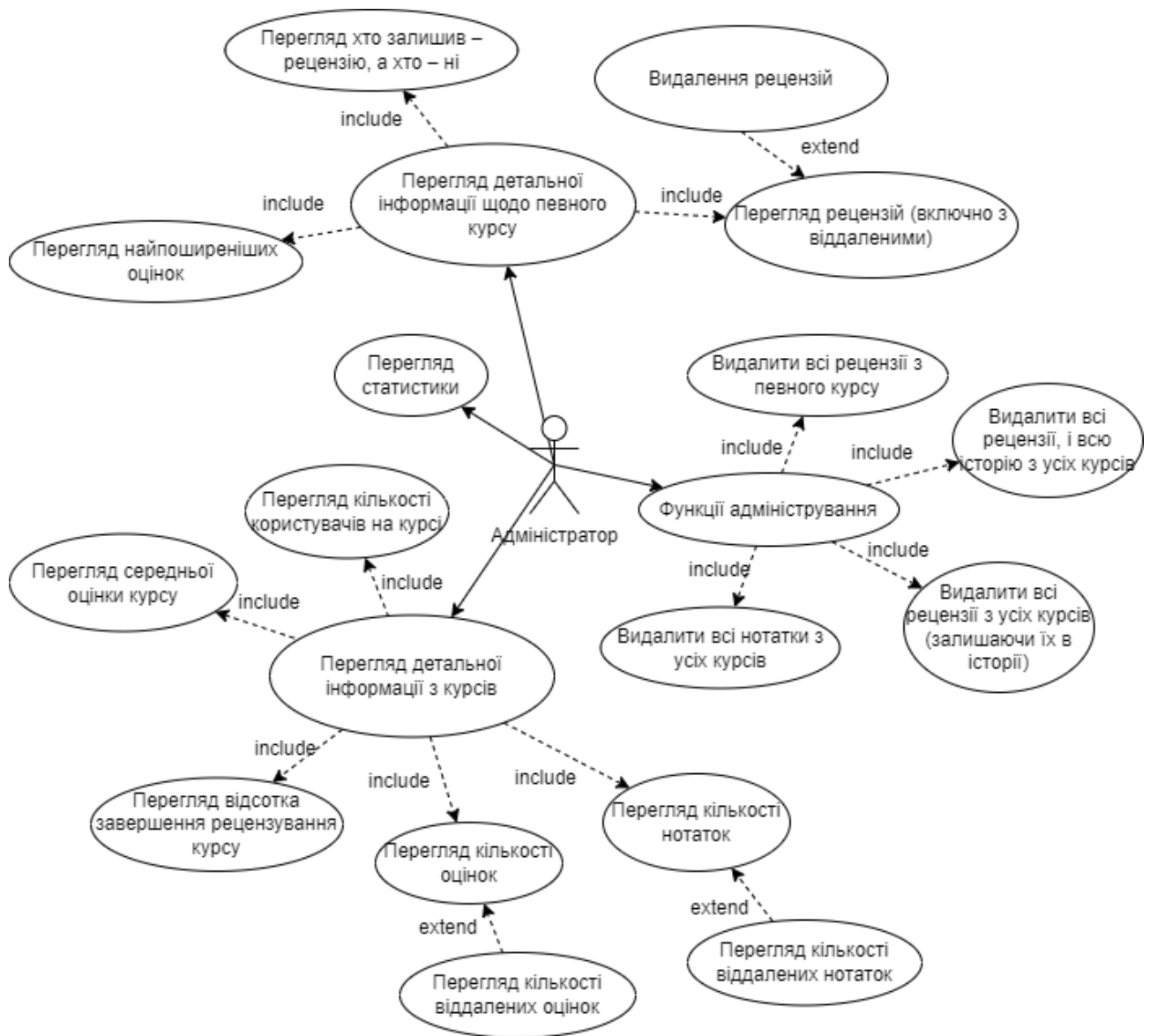


Рисунок 4.2 – Діаграма прецедентів системи для плагіну-звіту

Функції для ролі "адміністратора":

а) перегляд детальної інформації з курсів. Включає:

- 1) перегляд кількості користувачів на курсі;
- 2) перегляд відсотка завершення рецензування курсу;
- 3) перегляд середньої оцінки курсу;
- 4) перегляд кількості оцінок. Розширює:
  - перегляд кількості віддалених оцінок;
- 5) перегляд кількості нотаток. Розширює:
  - перегляд кількості віддалених нотаток;

б) перегляд детальної інформації щодо певного курсу. Включає:

- 1) перегляд найпоширеніших оцінок;
- 2) перегляд хто залишив – рецензію, а хто – ні;
- 3) перегляд рецензій (включно з віддаленими). Розширює:
  - видалення рецензій;

в) функції адміністрування. Включає:

- 1) видалити всі рецензії з певного курсу;
- 2) видалити всі рецензії, і всю історію з усіх курсів;
- 3) видалити всі нотатки з усіх курсів;
- 4) видалити всі рецензії з усіх курсів (залишаючи їх в історії);

г) перегляд статистики.

## 4.2 Додавання плагінів до системи

Послідовність дій для додавання плагінів:

- запускаємо Moodle від адміністратора;
- відкриваємо "Site administration" (рисунок 4.3);
- переходимо у вкладку "Plugins" (рисунок 4.4);
- переходимо в "Install plugins" (рисунок 4.4);
- завантажуюємо архів із плагіном (рисунок 4.5);
- натискаємо "Continue" (рисунок 4.6);
- натискаємо "Continue" (рисунок 4.7);
- натискаємо "Upgrade Moodle databse now" (рисунок 4.8);
- натискаємо "Continue" (рисунок 4.9);
- плагін встановлений.

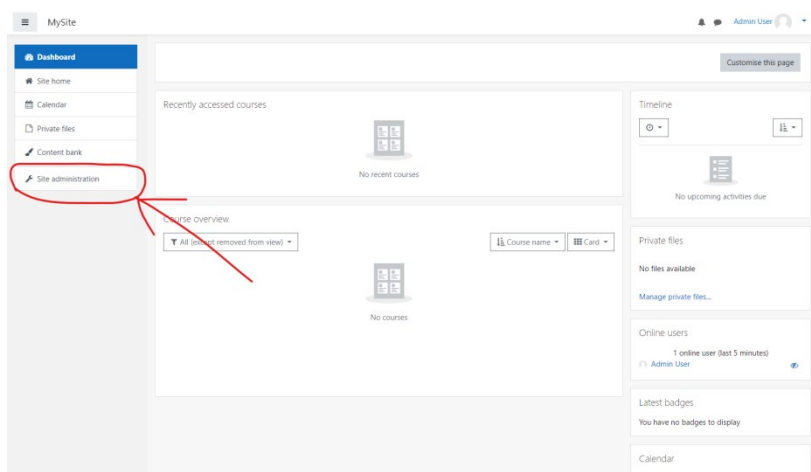


Рисунок 4.3 – Відкриття "Site administration" у Moodle

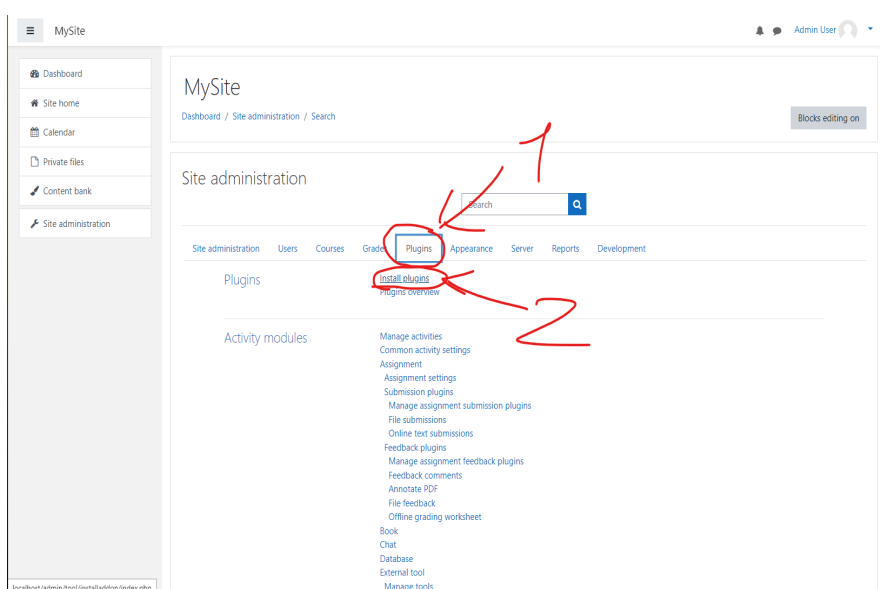


Рисунок 4.4 – Відкриття меню "Install plugins" у Moodle

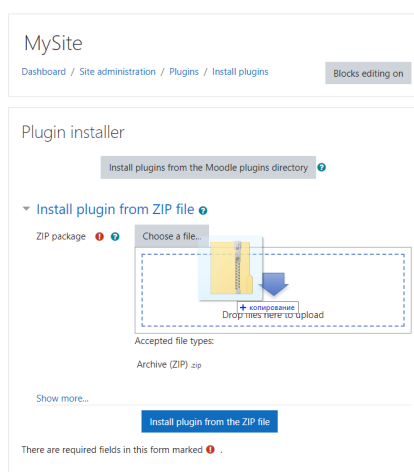


Рисунок 4.5 – Завантаження плагіну

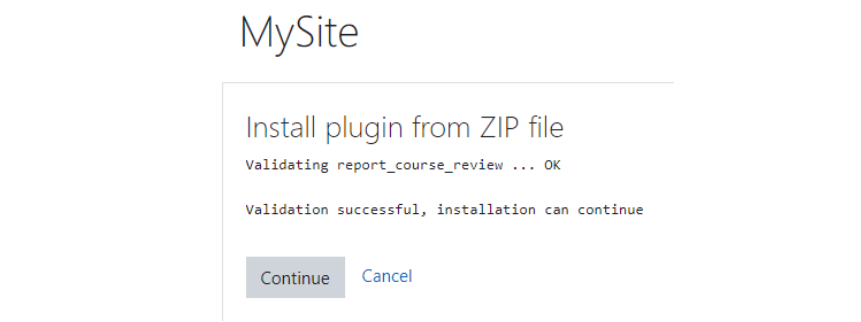


Рисунок 4.6 – Встановлення плагіну

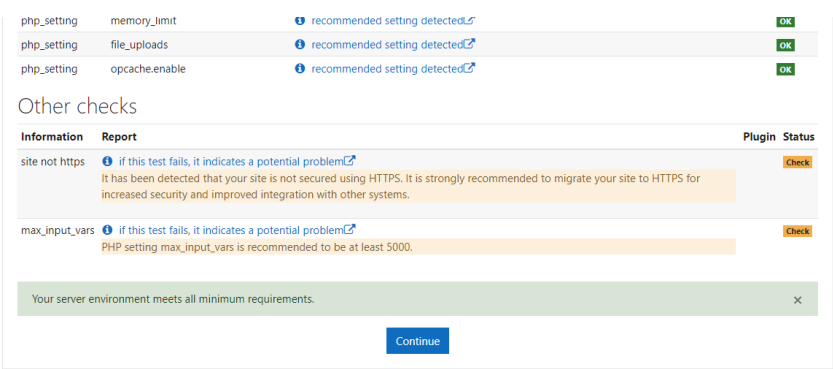


Рисунок 4.7 – Перевірка плагіну

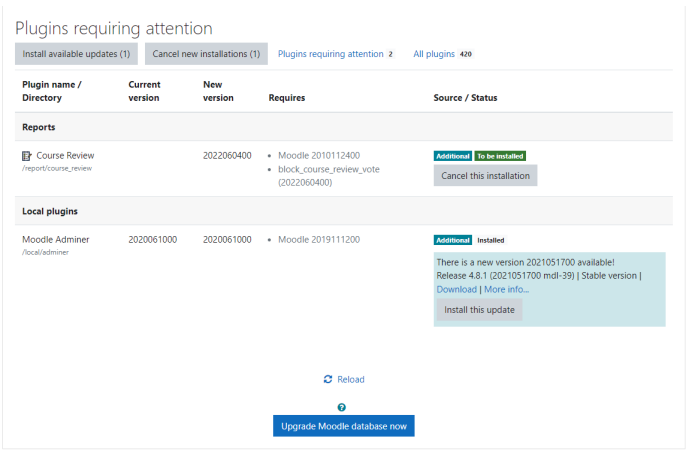


Рисунок 4.8 – Оновлення бази даних

Upgrading to new version

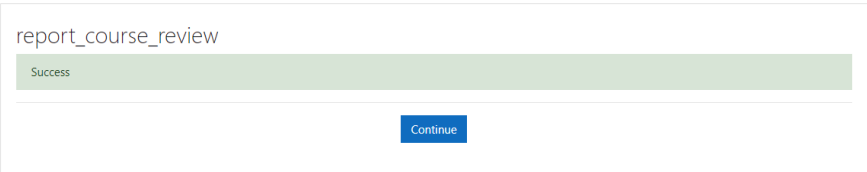


Рисунок 4.9 – Результат успішного встановлення

Проробимо ці дії для двох плагінів у такій послідовності:

- `block_course_review_vote`;
- `report_course_review`.

Далі для того щоб поставити блок на курс необхідно:

- зайти курс;
- натиснути на "Turn editing on" (рисунок 4.10);
- натиснути на "Add a block" (рисунок 4.11);
- вибрати блок "Course Review – Voting" (рисунок 4.12);
- блок доданий.

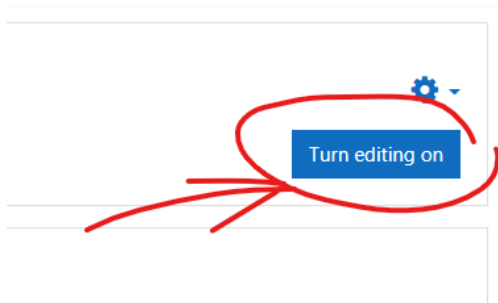


Рисунок 4.10 – Відкриття режиму редагування

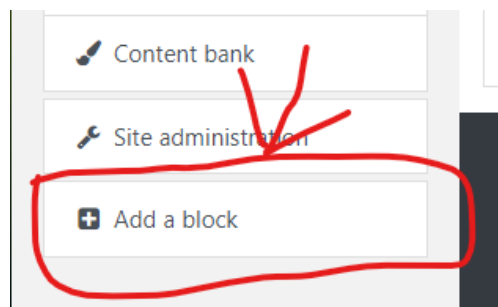


Рисунок 4.11 – Додавання блоку

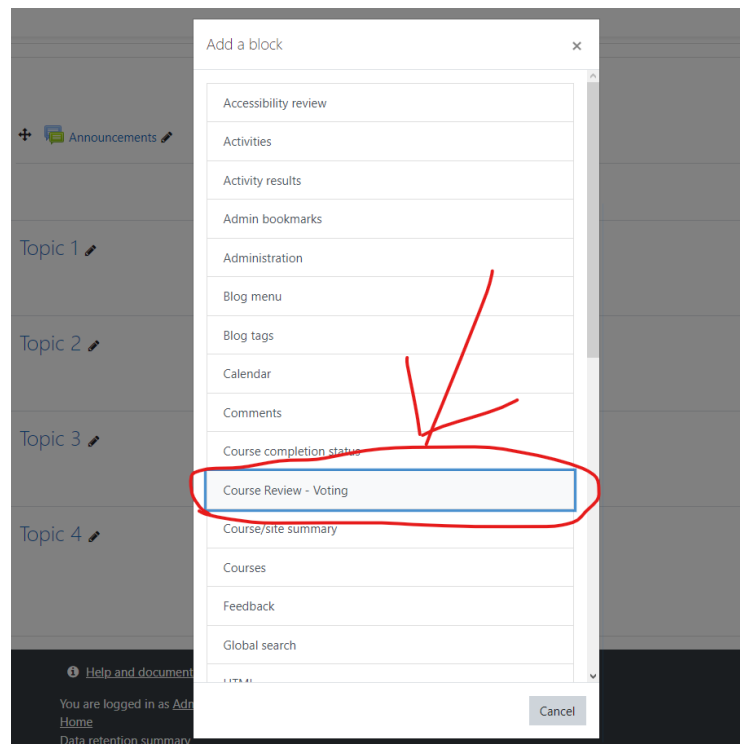


Рисунок 4.12 – Обирання блоку

### 4.3 Призначення та умови застосування програмного забезпечення

#### 4.3.1 Призначення програмного забезпечення

Цільовою аудиторією є адміністратор системи дистанційного навчання Moodle, викладачі та рецензенти.

Система рецензування курсів потрібна для забезпечення покращення якості навчання студентів. Рецензування курсів дозволить не залежати від суб'єктивності. Курс буде вважатися об'єктивно якісним, коли його схвалить без упередженості, спричиненої розумним суб'єктом.

Дане програмне забезпечення надасть можливість:

- залишати рецензії;
- переглядати рецензії;
- також має багато додаткових функцій для забезпечення зручного адміністрування рецензій.

#### 4.3.2 Умови застосування програмного забезпечення

Цей програмний додаток буде корисним в умовах університетів та шкіл.

Працює воно за умови наявності інтернету, середніх системних вимог, а також настроєної LMS Moodle, яка знаходиться на сервері.

Можна виділити основні мінімальні вимоги для персонального комп'ютера, який використовуватиметься користувачами:

а) платформа:

- 1) Windows: 7, 8, 8.1, 10;
- 2) Linux: будь-який сучасний дистрибутив;
- 3) Mac OS: OS X Yosemite 10.10 або вище;

б) розрядність:

- 1) x86 (32-bit);
- 2) x64 (64-bit);

в) відеоадаптер:

- 1) nVidia;
- 2) Intel;
- 3) Amd;
- 4) ATI;

г) пам'ять: 20 гб;

д) оперативна пам'ять: 2 гб;

е) контролер:

- 1) клавіатура;
- 2) миша;

є) стабільне інтернет з'єднання;

ж) роздільна здатність екрана 800x600.

Мінімальною необхідною версією Moodle для плагіна є: 2010112400.

### 4.3.3 Підсумки за розділом 4

Були розглянуті прецеденти системи та доступні їм функції. Для плагіна-блоку це: адміністратор та рецензент.

Роль – "Рецензент" може залишати і змінювати свою оцінку, також переглядати загальні дані (кількість користувачів на курсі, кількість тих, хто проголосував, середня оцінка курсу) і подивитися оцінку і замітку, що залишилася.

Роль – "Адміністратор" може переглядати хто – які оцінки, і нотатки поставив. також адміністратор може подивитися тих, хто ще не поставив оцінку.

Для плагіна-звіту є лише один прецедент – Адміністратор. Він може адмініструвати рецензування (видаляючи ті чи інші дані), переглядати детальну статистику всіх курсів (на яких на даний момент стоїть блок рецензування) і переглядати детальний огляд певного курсу.

Було також розглянуто призначення та умови застосування програмного забезпечення. Коротко це використання всередині університетів і шкіл для підвищення якості освіти студентів. А мінімальними вимогами для сервера є сервер, який може потягнути LMS Moodle, і на якому стоїть версія Moodle 2010112400. Для стандартного користувача буде достатньо простого комп'ютера, який може запустити нормальний браузер.

## **5 КЕРІВНИЦТВО ПРОГРАМІСТА**

### **5.1 Характеристики програми**

Програмна система являє собою 2 плагіни, один з яких – плагін-блок інший – плагін-звіт.

Плагіни написані для системи дистанційного навчання Moodle та виконують функцію рецензування курсів.

Готові установки файли знаходяться у файлах розширення .zip.

Займають вони у сжатому вигляді:

- block\_course\_review\_vote.zip – 17,9 КБ (18390 байт);
- report\_course\_review.zip 20,9 КБ (21472 байт).

У несжатому вигляді вони займають:

- block\_course\_review\_vote – 40,9 КБ (41893 байт);
- report\_course\_review – 58,3 КБ (59800 байт).

У більшості файлів плагіни містять код мови PHP, 1 файл .css для стилів, і зображення типу .png. Також у плагіні-блоці є файл розширення .xml у ньому зберігатиметься розмітка таблиці для бази даних.

В основному плагін написаний мовою програмування PHP з періодичним використанням декларативної мови програмування – SQL який використовується для взаємодії з базою даних Moodle. Файли .php генерують сторінку, що відображається для користувача.

Для використання системи рецензування необхідно встановити в LMS Moodle.

### **5.2 Звернення до плагіна**

Інтерфейсом плагіна є html сторінка сайту Moodle.

Інтерфейс блоку доступний за його наявності на сторінці курсу.

Інтерфейс звітів знаходиться у відповідному розділі адміністрування "Reports", де знаходиться у відповідній назві – пункті "Course Review".

Інші способи переходу між інтерфейсами плагінів будуть представлені на діаграмі переходу між графічними інтерфейсами (рисунок 5.1).

На рисунку 5.1 відображено переходи між інтерфейсами для ролі рецензентів.



Рисунок 5.1 – Діаграма переходів між інтерфейсами для рецензентів

На рисунку 5.2 відображені переходи між інтерфейсами для ролі адміністраторів.

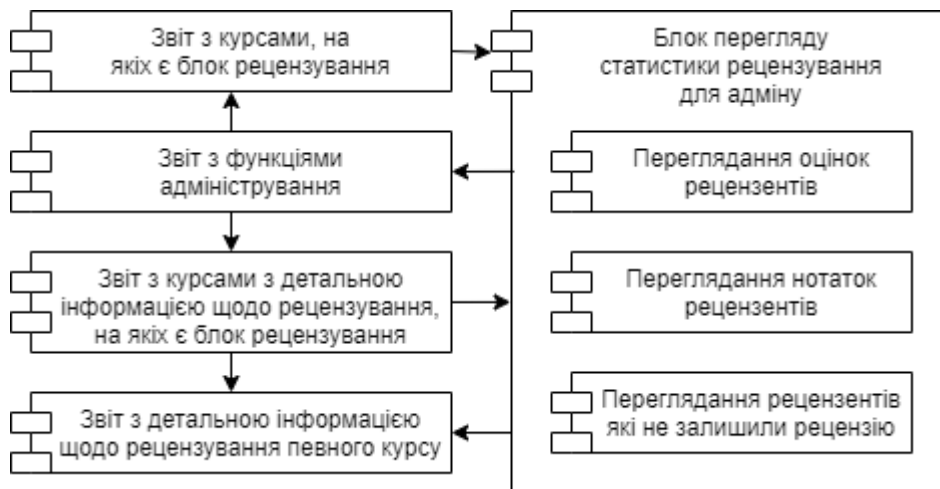


Рисунок 5.2 – Діаграма переходів між інтерфейсами для адміністратора

Наведемо приклади інтерфейсів для участі рецензентів.

На рисунку 5.3 відображено приклад інтерфейсу блоку залишення рецензії для ролі "Рецензент".

Course Review - Voting

**Vote for this course**

You can write a review about the course (optional):

Click on a rate to vote for this course. Choose from:

-2
-1
0
1
2

Voted 2 out of 3 users, this course has an average rate:

1.5

Рисунок 5.3 – Блок залишення рецензії для ролі "Рецензент"

На рисунку 5.4 відображено приклад інтерфейсу блоку перегляду рецензії з нотаткою для ролі "Рецензент".

Course Review - Voting

You have already voted for this course. Your vote was:

2

You noted: "Really not bad. I would like a little more tasks, so that the students would suffer, but it's not bad anyway"

[\(Want to remove vote?\)](#)

Voted 1 out of 3 users, this course has an average rate:

2

Рисунок 5.4 – Блок перегляду рецензії з нотаткою для ролі "Рецензент"

На рисунку 5.5 відображено приклад інтерфейсу блоку перегляду рецензії без нотатки для ролі "Рецензент".

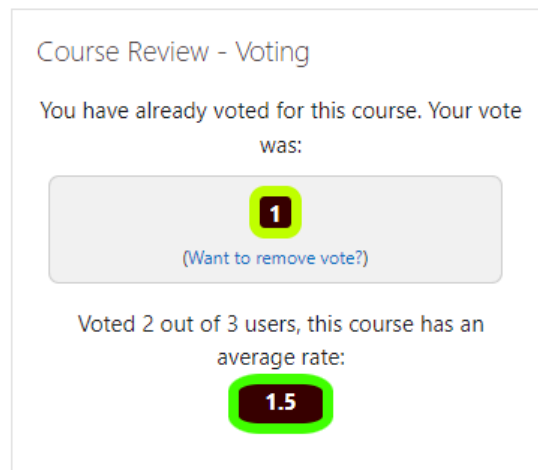


Рисунок 5.5 – Блок перегляду рецензії без нотатки для ролі "Рецензент"

Додамо приклади інтерфейсів для участі адміністраторів.

На рисунку 5.6 відображено приклад інтерфейсу блоку перегляду результатів рецензування з незаповненими рецензіями для ролі "Адміністратор".

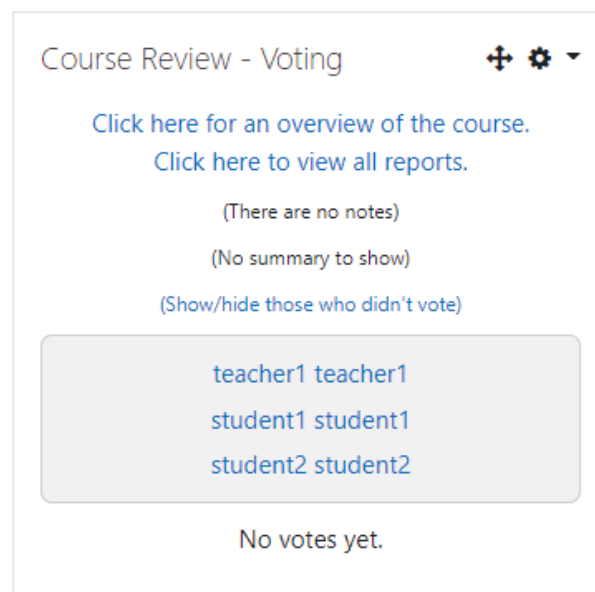


Рисунок 5.6 – Блок перегляду результатів рецензування з незаповненими рецензіями для ролі "Адміністратор"

На рисунку 5.7 відображено приклад інтерфейсу блоку перегляду результатів рецензування з деякими заповненими рецензіями для ролі "Адміністратор".

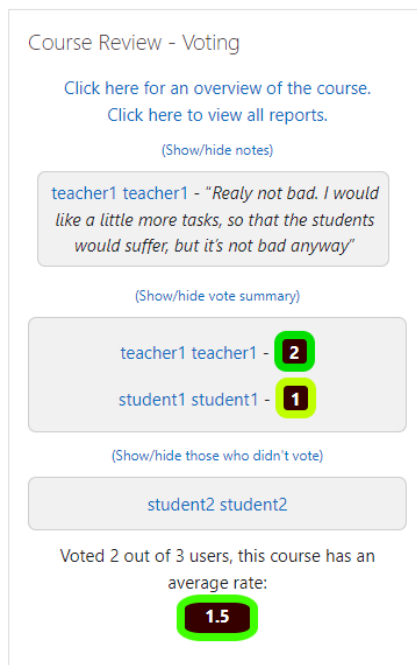


Рисунок 5.7 – перегляду результатів рецензування з деякими заповненими рецензіями для ролі "Адміністратор"

На рисунку 5.8 відображено приклад інтерфейсу звіту з функціями адміністрування для ролі "Адміністратор".

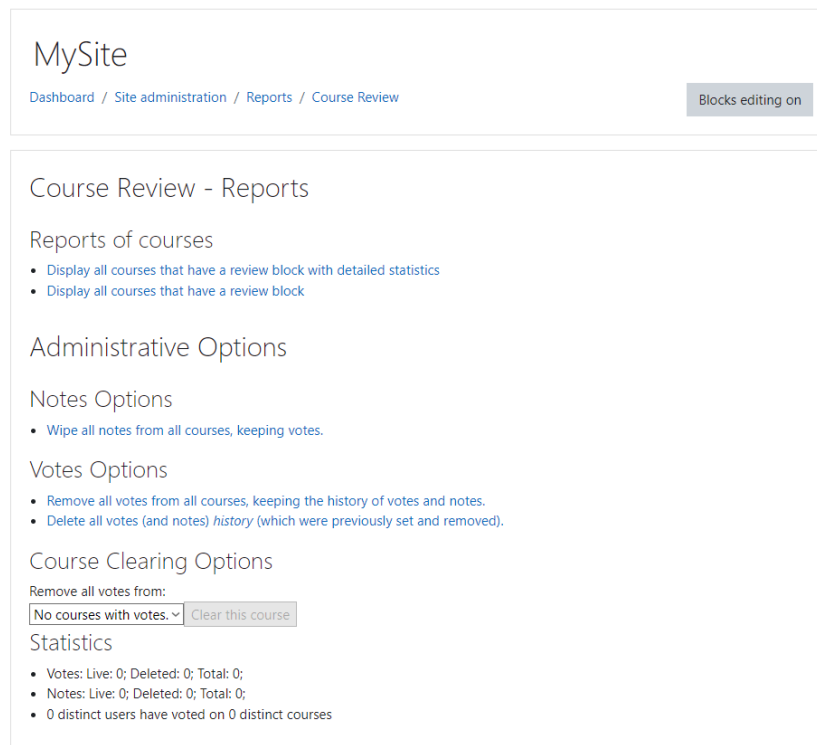


Рисунок 5.8 – Звіт з функціями адміністрування для ролі "Адміністратор"

На рисунку 5.9 відображено приклад інтерфейсу звіту зі списком курсів, на яких стоїть блок рецензування для ролі "Адміністратор".

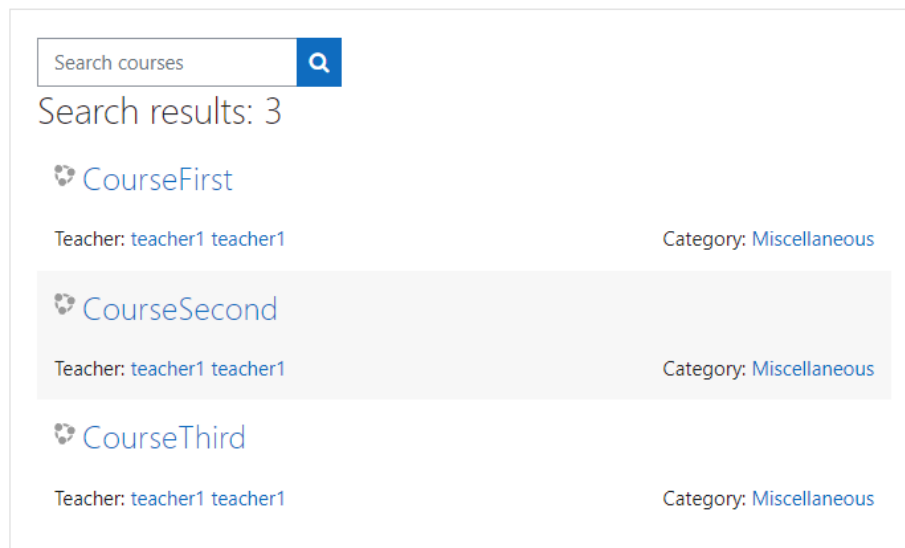


Рисунок 5.9 – Звіт зі списком курсів, на яких стоїть блок рецензування для ролі "Адміністратор"

На рисунку 5.10 відображено приклад інтерфейсу звіту зі списком курсів, на яких стоїть блок рецензування з детальною інформацією для ролі "Адміністратор".

MySite

Dashboard / Site administration / Reports / Course Review

Simple List of all courses with votes (as of 2022-06-06 14:45:06)

This image:  indicates the column which is sorted.

| Pos | Course       | Average course rate | Users | Votes (Deleted) | Notes (Deleted) | Percentage of voters |
|-----|--------------|---------------------|-------|-----------------|-----------------|----------------------|
| 1   | CourseFirst  | 1.50                | 3     | 2 (1 deleted)   | 1               | 67%                  |
| 2   | CourseSecond | 1.00                | 2     | 1               | 0               | 50%                  |
| 3   | CourseThird  | 0.00                | 3     | 0               | 0               | 0%                   |

[Report Index.](#)

Рисунок 5.10 – Звіт зі списком курсів, на яких стоїть блок рецензування з детальною інформацією для ролі "Адміністратор"

На рисунку 5.11 відображено приклад інтерфейсу звіту з детальною інформацією за певним курсом для ролі "Адміністратор".

Votes and Notes for CourseFirst (CourseFirst) (as of 2022-06-06 14:46:38)

Summary (Live votes only)

| Vote | Votes Cast | Percentage |
|------|------------|------------|
| 2    | 1          | 50.0%      |
| 1    | 1          | 50.0%      |

This course's score is 1.50

Who voted

| Name              | Voted |
|-------------------|-------|
| teacher1 teacher1 | +     |
| student1 student1 | +     |
| student2 student2 | -     |

Detail

Greyed out text is a deleted vote.

| Pos | Date                                                       | Name              | Vote | Note                                                                                                           | Delete?                                                                               |
|-----|------------------------------------------------------------|-------------------|------|----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| 1   | Added: 2022-06-06 14:24:11                                 | student1 student1 | 1    | -                                                                                                              |  |
| 2   | Added: 2022-06-06 14:22:18                                 | teacher1 teacher1 | 2    | "Really not bad. I would like a little more tasks, so that the students would suffer, but it's not bad anyway" |  |
| 3   | Added: 2022-06-06 14:20:50<br>Deleted: 2022-06-06 14:20:52 | teacher1 teacher1 | 1    | -                                                                                                              |                                                                                       |

[Report Index.](#)

Рисунок 5.11 – Звіт з детальною інформацією за певним курсом для ролі "Адміністратор"

### 5.3 База даних програми

Всі записи та зчитування даних виконуються за допомогою мови програмування SQL.

Для запису рецензій було розроблено та створено таку таблицю для бази даних (таблиця 5.1):

Таблиця 5.1 – Таблиця даних для збереження рецензій

| Назва поля      | Тип поля | Довжина поля | Опис поля                                                                                     |
|-----------------|----------|--------------|-----------------------------------------------------------------------------------------------|
| id              | int      | 10           | Ключове поле таблиці.                                                                         |
| user_id         | int      | 10           | Прив'язка до ідентифікатора користувача.                                                      |
| course_id       | int      | 10           | Прив'язка до ідентифікатора курсу.                                                            |
| vote            | int      | 1            | Оцінка.                                                                                       |
| date_added      | int      | 10           | Дата додавання.                                                                               |
| date_modified   | int      | 10           | Дата зміни.                                                                                   |
| note            | char     | 1024         | Нотатка(необов'язкове поле).                                                                  |
| deleted         | int      | 1            | Чи видалена замітка (0 = не видалена, 1 = видалена).                                          |
| deleted_user_id | int      | 10           | Ідентифікатор того, ким видалено нотатку (може видалити сам рецензент, а може адміністратор). |

Для взаємодії плагіну із системою Moodle використовувалися такі вбудовані таблиці:

- course;
- context;
- role\_assignments;
- user;
- role;
- block\_instances.

Ось їх схема взаємозв'язку на рисунку 5.12.

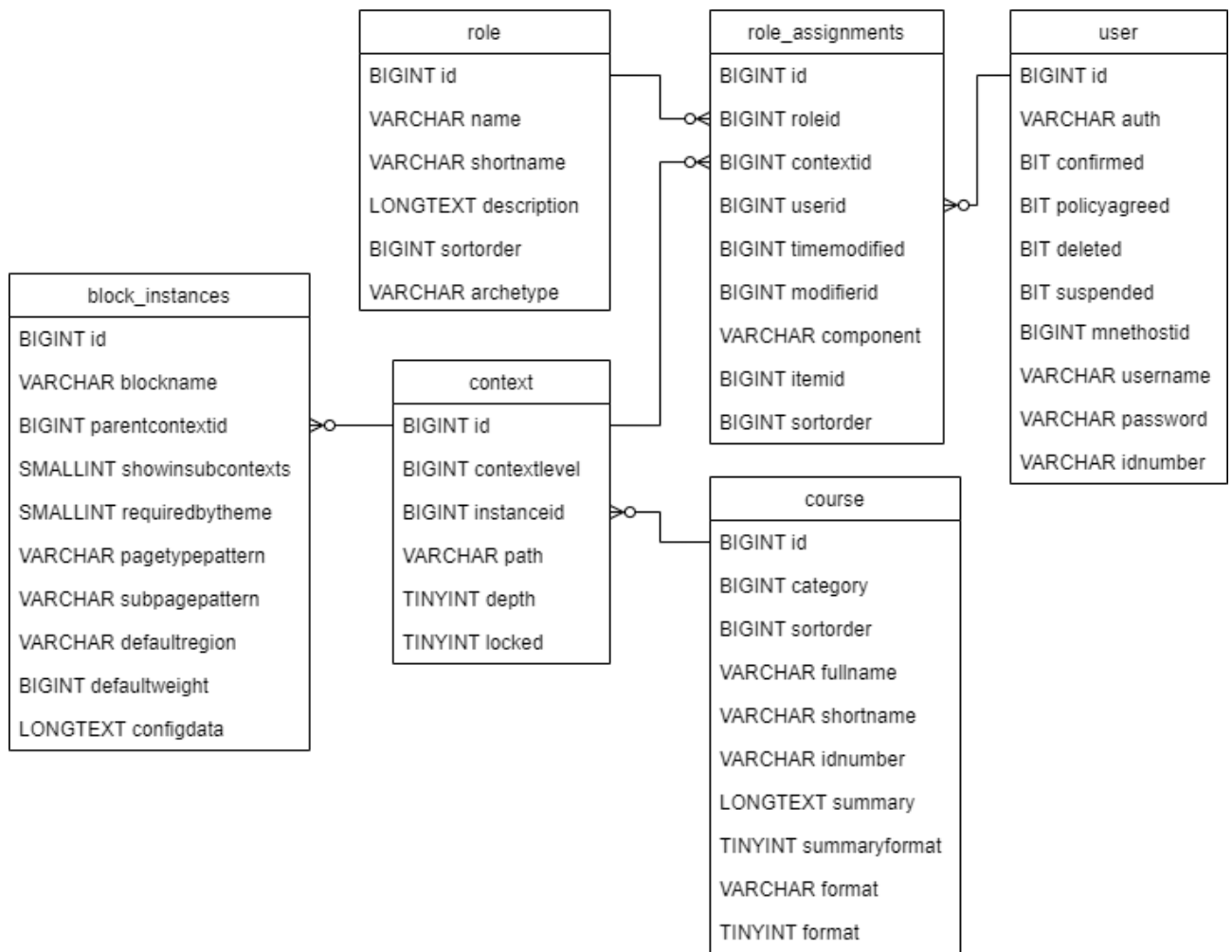


Рисунок 5.12 – Діаграма зв'язків використовуваних у плагіні вбудованих таблиць бази даних LMS Moodle

Таблиця `block_instances` використовується для виявлення курсів, на яких зараз стоїть блок `block_course_review_vote`, знаходиться це за допомогою такого запиту sql:

```

$contain = $DB->get_records_sql("
SELECT course.id
FROM course course
JOIN context ON course.id = context.instanceid
JOIN block_instances ON block_instances.parentcontextid = context.id
WHERE course.id = ".$this->course_id."
AND block_instances.blockname = 'course_review_vote'");
$this->containBlock = $contain ? true : false;
  
```

Таблиці `context`, `role_assignments`, `user`, `role` використовується для виявлення підрахунку кількості користувачів на курсі, це за допомогою такого запиту `sql`.

```
$count = $DB->get_records_sql("
SELECT COUNT(course.id, user.id)
FROM course
JOIN context ON course.id = context.instanceid
JOIN role_assignments} ra ON role_assignments.contextid = context.id
JOIN user ON user.id = role_assignments.userid
JOIN role ON role.id = role_assignments.roleid
WHERE course.id = ".$this->course_id.";");
```

Створена таблиця `block_course_review_vote` використовується для багатьох простих маніпуляцій, ось приклади деяких з них.

Порахувати всі існуючі оцінки з курсу.

```
SELECT COUNT(vote)
FROM ".PREFIX.TBL_REVIEW."
WHERE deleted = '0'
AND ".PREFIX.TBL_REVIEW.".course_id = ".PREFIX."course.id
GROUP BY course_id;
```

Порахувати всі віддалені оцінки з курсу.

```
SELECT COUNT(vote)
FROM ".PREFIX.TBL_REVIEW."
WHERE deleted = '1'
AND ".PREFIX.TBL_REVIEW.".course_id = ".PREFIX."course.id
GROUP BY course_id;
```

## 5.4 Алгоритми програми

Представимо алгоритм виведення інтерфейсу плагіна-блоку на рисунку 5.13:

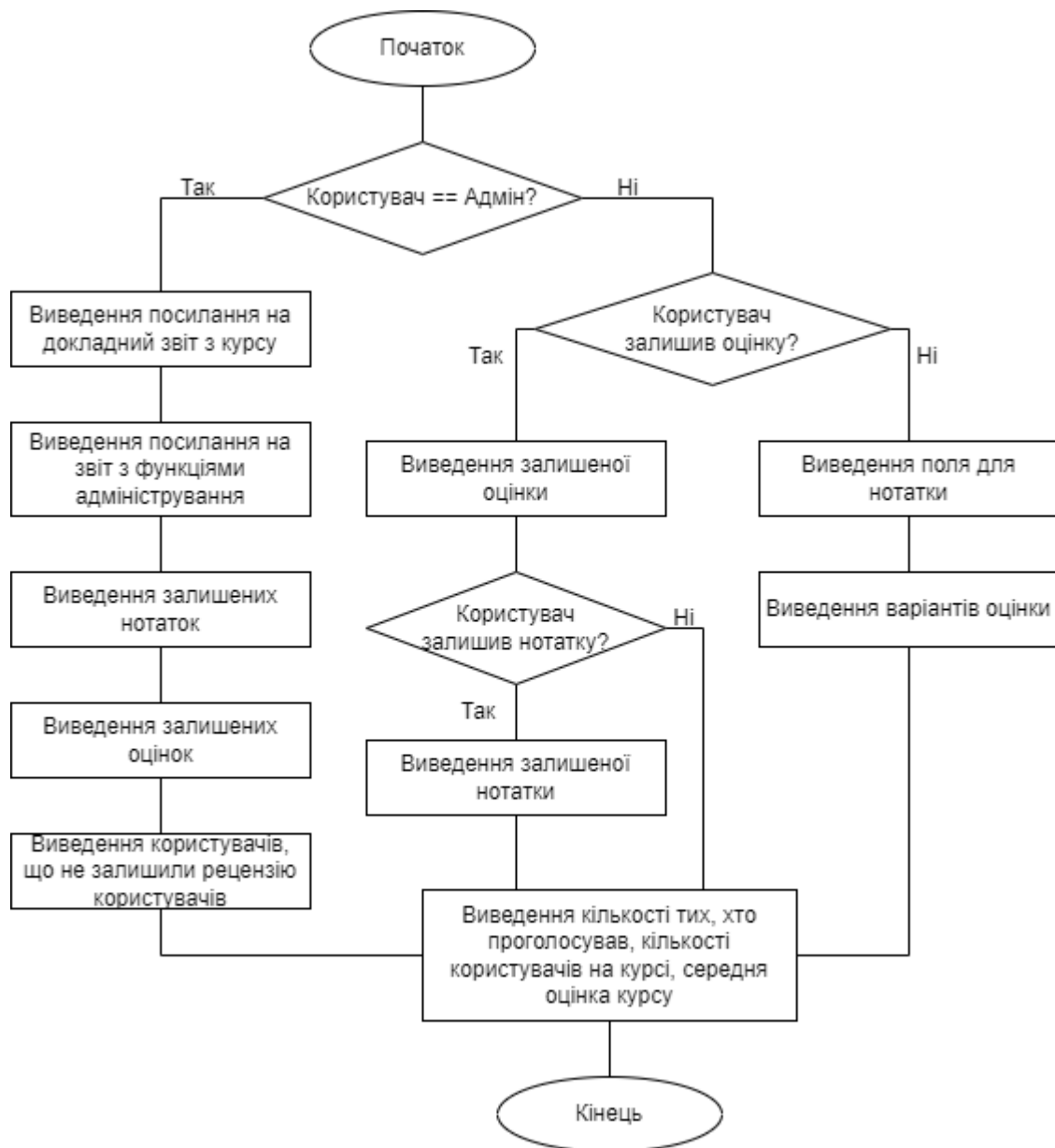


Рисунок 5.13 – Алгоритм виведення на сторінку плагіна-блоку

Представимо алгоритми обробників подій для ролі "Рецензент".

При натисканні на виставлення оцінок (рисунок 5.14).

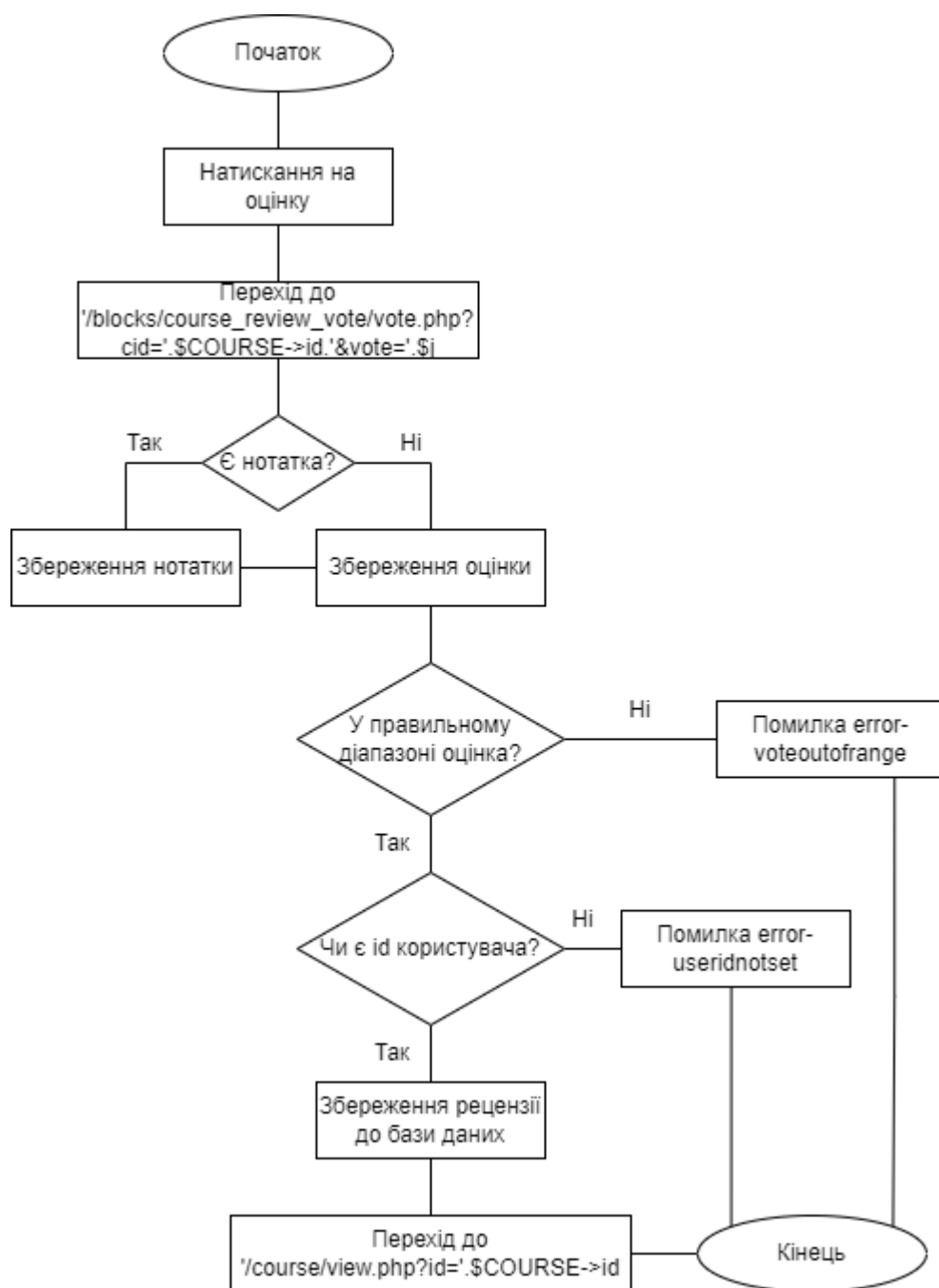


Рисунок 5.14 – Діаграма збереження рецензій

При натисканні на видалення рецензії (рисунок 5.15).

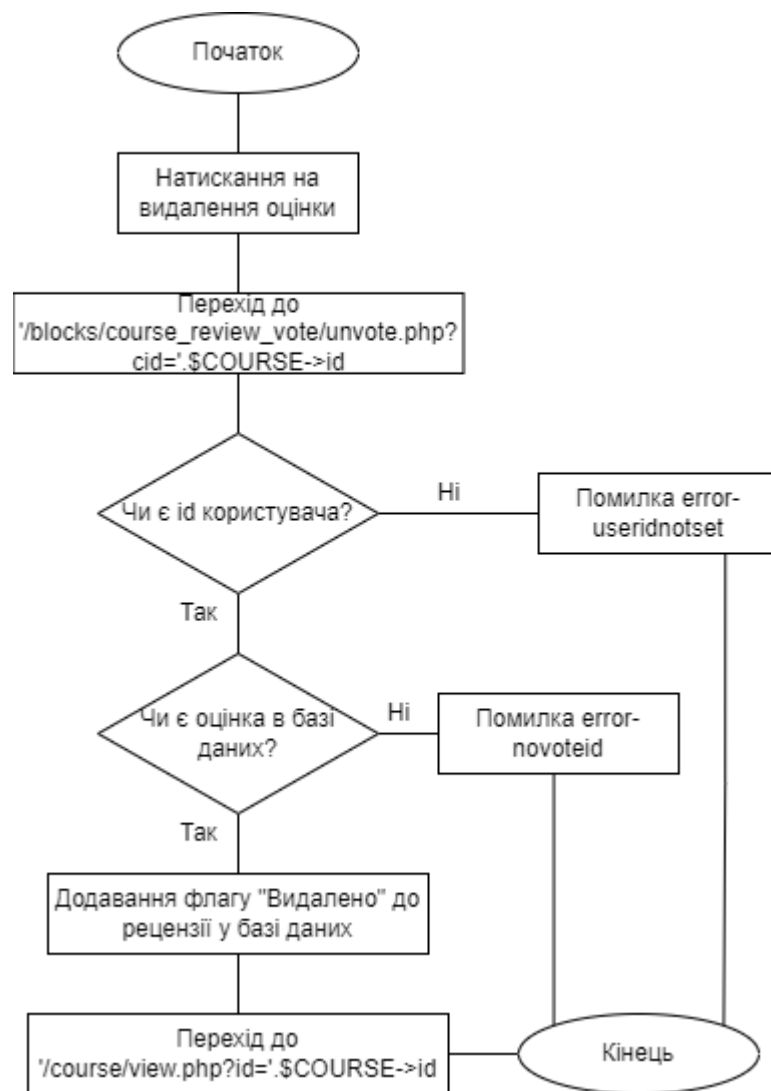


Рисунок 5.15 – Діаграма видалення рецензій

Обробки на інтерфейсі ролі "Адміністратора" набагато простіші – вони просто переходять за посиланнями.

При натисканні на "Click here for an overview of the course." – пересилається на адресу `/report/course_review/report.php?q=v&c='.$COURSE->id.'&s=c'`.

При натисканні на "Click here to view all reports." – пересилається на адресу `/report/course_review/index.php`.

При натисканні на будь-яке з "Show/hide" ховаються або відображаються результати по тій чи іншій темі.

При натисканні на користувачів надсилається на адресу цього користувача.

Для плагіна-звіту немає жодних алгоритмів, які потребують пояснення. Вони просто по черзі виводяться інформація з бази даних. Тільки можна докладніше пояснити складання звітів (рисунк 5.16).

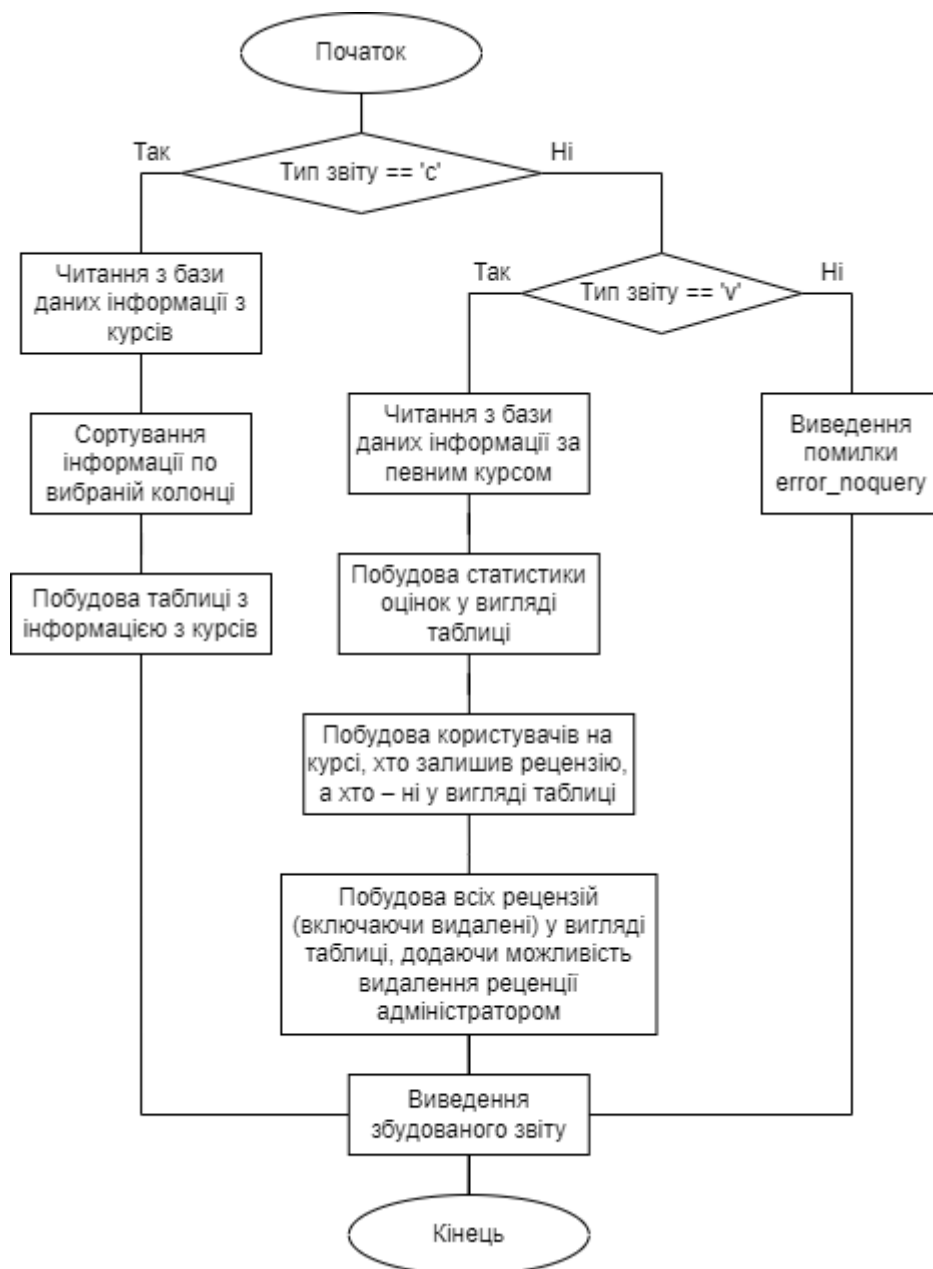


Рисунок 5.16 – Діаграма побудови звіту

## 5.5 Вхідні та вихідні дані

Вхідні дані від ролі "Рецензент" до системи:

- залишення оцінки курсу;
- залишення нотатки курсу (опціонально);
- видалення рецензії курсу.

Вихідні дані від системи до ролі "Рецензент":

- перегляд своєї оцінки курсу;
- перегляд своєї замітки курсу;
- перегляд кількості людей на курсі;
- перегляд тих, хто залишив рецензію на курсі;
- перегляд середньої оцінки курсу.

Вхідні дані від ролі "Адміністратор" до системи:

- видалення всіх рецензій;
- видалення всіх нотаток;
- видалення історії рецензій;
- видалення історії рецензій курсу;
- видалення певної рецензії курсу.

Вихідні дані від системи до ролі "Адміністратор":

- перегляд курсів, на яких стоїть блок рецензування;
- перегляд курсів, на яких стоїть блок рецензування з детальною інформацією;
- перегляд тих, хто залишив оцінку на курсі;
- перегляд тих, хто залишив нотатки на курсі;
- перегляд статистики оцінок курсу;
- перегляд віддалених рецензій.

## 5.6 Повідомлення, обробка помилкових ситуацій

Таблиця помилок, можливих під час використання плагіна-блоку (таблиця 5.2):

Таблиця 5.2 – Таблиця помилок для плагіну-блоку

| Тип помилки          | Текст помилки                                                                         | Опис помилки                                                                                                 |
|----------------------|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| error-voteoutofrange | Vote cast is out of range.<br>Stop trying to hack the review system.                  | Оцінка виходить за межі допустимого діапазону.<br>Винамагаєтесь зламати систему оцінювання.                  |
| error-courseidnotset | Course ID not set, for some reason.                                                   | Ідентифікатор курсу не встановлено з якоїсь причини.                                                         |
| error-useridnotset   | User ID not set, for some reason.                                                     | Ідентифікатор користувача не встановлено з якоїсь причини.                                                   |
| error-dbinsert       | Vote not recorded into the database for some reason.                                  | Голосування не занесено до бази даних з якоїсь причини.                                                      |
| error-dbupdate       | Vote not updated in the database for some reason.                                     | Голосування не оновлюється в базі даних з якоїсь причини.                                                    |
| error-novoteid       | Could not get ID for the vote for some reason.                                        | Не вдалося отримати ID для голосування з якоїсь причини.                                                     |
| error-iefixfail      | The code which fixes IE code has failed. Please use a better web browser to continue. | Код, який адаптується під ІЕ, зазнав невдачі. Будь ласка, скористайтесь новим веб-браузером, щоб продовжити. |

Таблиця помилок, можливих під час використання плагіна-звіту (таблиця 5.3).

Таблиця 5.2 – Таблиця помилок для плагіну-звіту

| Тип помилки      | Текст помилки                                                                                                                                  | Опис помилки                                                                   |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| error_admin      | Sorry, you need to be a site Admin to see the Course Reports.                                                                                  | Ви повинні бути адміністратором сайту, щоб переглядати звіти про курси.        |
| error_query      | Sorry, the query could not be composed properly and cannot be run.                                                                             | Запит не може бути правильно складений і не може бути запущений.               |
| error_noquery    | Sorry, the URL is malformed and an appropriate report could not be generated.                                                                  | URL-адреса має неправильний формат, і відповідний звіт не може бути створений. |
| error_notmysql   | Sorry, you need to be using MySQL for this to work.                                                                                            | Немає нотаток для видалення.                                                   |
| error_nonotes    | There are no live notes to remove.                                                                                                             | Немає оцінок для видалення.                                                    |
| error_novotes    | There are no live votes to remove.                                                                                                             | Немає рецензій для видалення.                                                  |
| error_noanyvotes | There are no votes to remove.                                                                                                                  | Немає рецензій на видалення.                                                   |
| error_noblock    | Sorry, couldn't get the block ID for the 'course_review_vote\' block. Have you installed it? If so, please report this error to the developer. | не вдалося отримати ідентифікатор блоку для блоку 'course_review_vote'.        |
| error_dbupdate   | Database not updated for some reason.                                                                                                          | База даних не оновлюється з якоїсь причини.                                    |

## 5.7 Висновок за розділом 5

Програма написана здебільшого на php, періодично використовуються запити SQL для роботи з базою даних.

Для роботи з базою даних була створена таблиця `block_course_review_vote`, яка містить дані про рецензії.

Також використовувалися такі вбудовані таблиці бази даних:

- `course`;
- `context`;
- `role_assignments`;
- `user`;
- `role`;
- `block_instances`.

Робота алгоритмів полягає у переходах на різні адреси php-файлів, які обробляють події.

Вхідні та вихідні дані містять в основному способи адміністрування, та перегляду рецензій для роду "Адміністратор". А для ролі "Рецензент" доступне лише залишення рецензії, перегляд своєї рецензії та її видалення.

Кожен плагін містить у собі різні обробники винятків, помилок.

## **6 ЕКСПЛУАТАЦІЯ ОПЕРАТОРОМ ПРОГРАМИ**

### **6.1 Призначення програми**

Загалом програма призначена для рецензування курсів у LMS Moodle. Вбудованих засобів у Moodle для такого функціоналу немає тому було вирішено написати додаток (Плагін). Було написано два плагіни: плагін-блок, і плагін-звіт. Плагін-блок дозволяє писати та надсилати рецензії. А плагін-звіт дозволяє переглядати статистику та детальну інформацію з рецензування курсів.

### **6.2 Умови використання програми**

Для сервера необхідні мінімальні системні вимоги Moodle, також для роботи плагіна Moodle повинен бути не старішим ніж 2010112400 (Moodle 2.0).

Для віддаленого використання сервера достатньо використовувати будь-який статичний або портативний пристрій. Головне, щоб він підтримував браузер, і мав доступ в інтернет.

### **6.3 Використання програми**

#### **6.3.1 Використання програми для ролі "Адміністратор"**

Адміністратор має широкі можливості для адміністрування.

На плагіні-блоці він може переглядати оцінки рецензентами, нотатки написані рецензентами, також можна побачити, які рецензенти ще не залишили рецензію. Також адміністратор може перейти на докладний звіт з цього курсу, і має посилання функції адміністрування рецензіями. Приклад інтерфейсу наведено рисунку 6.1:

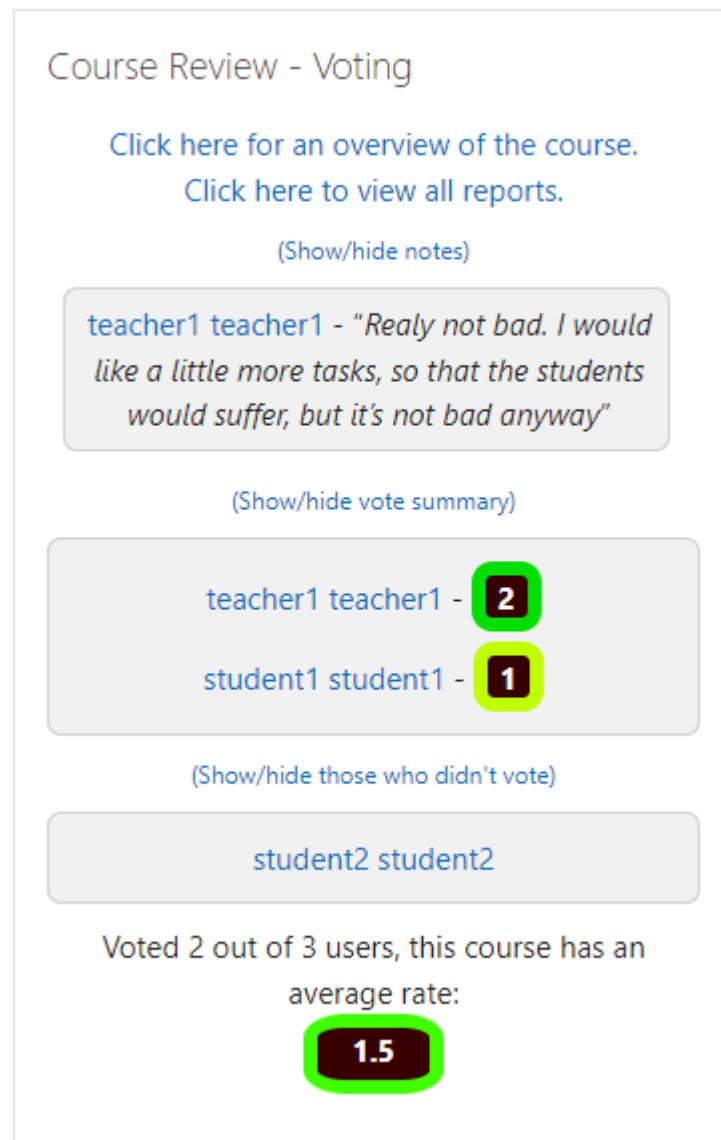


Рисунок 6.1 – Інтерфейс плагіну-блоку для адміністратора

На плагіні-звіті адміністратор може знайти багато корисних функцій.

Серед них:

- відображення всіх курсів, на яких стоїть блок рецензування;
- відображення всіх курсів, на яких стоїть блок рецензування у вигляді таблиці, з детальною інформацією щодо кожного курсу, та можливістю сортування за стовпчиками;
- видалення всіх нотаток з усіх курсів (при цьому зберігаючи оцінки);
- видалення всіх рецензій з усіх курсів (при цьому зберігаючи історію);

- видалення всіх рецензій, та історію з усіх курсів;
- видалення всіх рецензій із певного курсу.

Приклад такого інтерфейсу можна спостерігати на рисунку 6.2:

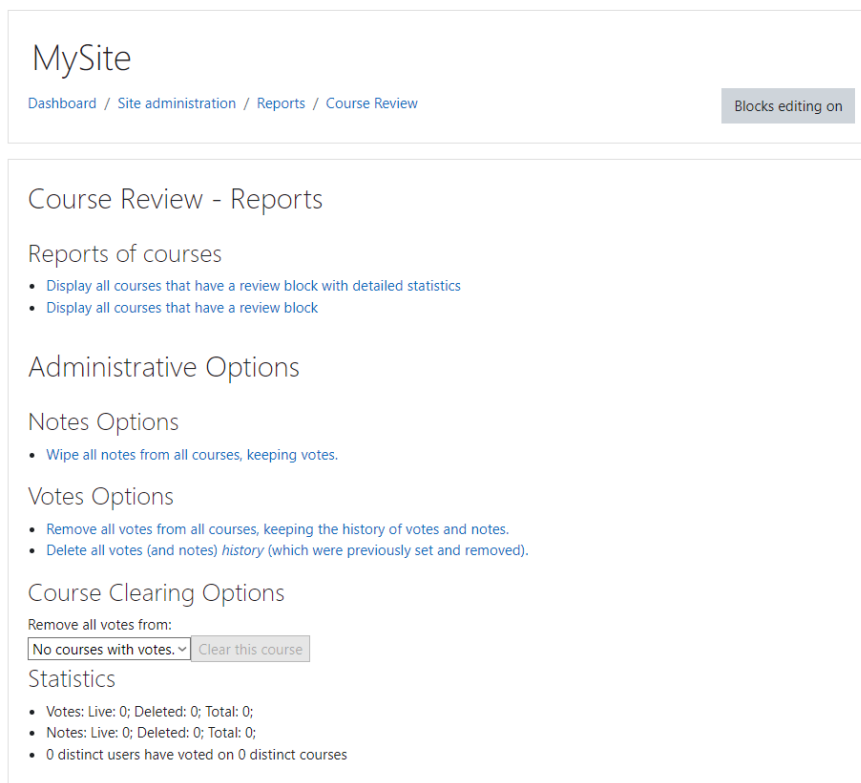


Рисунок 6.2 – Інтерфейс функцій адміністрування

Також можна навести зовнішній вигляд таблиці з даними за курсами, на яких знаходиться блок рецензування (рисунок 6.3).

MySite

Dashboard / Site administration / Reports / Course Review

### Simple List of all courses with votes (as of 2022-06-06 14:45:06)

This image:  indicates the column which is sorted.

| Pos | Course       | Average course rate | Users | Votes (Deleted) | Notes (Deleted) | Percentage of voters |
|-----|--------------|---------------------|-------|-----------------|-----------------|----------------------|
| 1   | CourseFirst  | 1.50                | 3     | 2 (1 deleted)   | 1               | 67%                  |
| 2   | CourseSecond | 1.00                | 2     | 1               | 0               | 50%                  |
| 3   | CourseThird  | 0.00                | 3     | 0               | 0               | 0%                   |

[Report Index.](#)

Рисунок 6.3 – Детальні дані по курсам, на яких є блок рецензування

## Докладний звіт про рецензування для певного курсу (рисунок 6.4).



Рисунок 6.4 – Детальні дані по курсу

### 6.3.2 Використання програми для ролі "Рецензент"

Для ролі "Рецензент" все набагато простіше. Йому доступне залишення рецензії, яку можна спостерігати на рисунку 6.5, додавання нотатки є опціональною дією і не обов'язково.

Course Review - Voting

**Vote for this course**

You can write a review about the course (optional):

Click on a rate to vote for this course. Choose from:

-2
-1
0
1
2

Voted 2 out of 3 users, this course has an average rate:

1.5

Рисунок 6.5 – Залишення своєї рецензії

Також йому доступно перегляд своєї рецензії, та перегляд сумарної статистики за курсом (кількість користувачів на курсі, кількість тих, хто залишив рецензію та середня оцінка курсу) (рисунок 6.6). Він може свою рецензію видалити і залишити іншу.

Course Review - Voting

You have already voted for this course. Your vote was:

2

You noted: "Really not bad. I would like a little more tasks, so that the students would suffer, but it's not bad anyway"

[\(Want to remove vote?\)](#)

Voted 1 out of 3 users, this course has an average rate:

2

Рисунок 6.6 – Перегляд своєї рецензії

## **6.4 Повідомлення оператору**

Повідомлення оператору можна переглянути в розділі "5.6 Повідомлення, обробка помилкових ситуацій" у даному розділі знаходиться 2 таблиці з обробками помилок, які можуть пощастити сісти оператору.

## **6.5 Підсумки по розділу 6**

Програма призначена для рецензування курсів, має 2 ролі: "Адміністратор" та "Рецензент". Першому доступний широкий функціонал для адміністрування та перегляду статистики рецензування курсів. А другому є лише залишення рецензії, видалення своєї рецензії, перегляд своєї рецензії.

## ВИСНОВКИ

У процесі написання дипломної роботи в першому розділі було проаналізовано:

- основні показники системи дистанційного навчання;
- специфікації розробки LMS;
- було розглянуто та порівняно основні 5 систем дистанційного навчання (Classroom, moodle, Absorb, Docebo, TalentLMS).

У другому розділі проаналізовано способи розгорнення та вимоги системи дистанційного навчання. Також порівняно мови програмування та технології для написання backend та frontend. В результаті чого було обрано наступний стек: PHP + JavaScript + SCSS. Також було розглянуто принципи розробки та встановлення плагінів у Moodle.

У третьому розділі розглянуто:

- алгоритм рецензування курсів;
- модулі розробленого додатка (серед яких два плагіни: плагін-блок і плагін-звіт);
- також розглянуто файлове налаштування плагінів.

У четвертому розділі розглянуто функціональні можливості розробленої системи рецензування. Визначено функції ролей, яких дві: "Адміністратор" та "Рецензент". Розглянуто встановлення плагінів, та встановлення блоку рецензування на курс. Було розглянуто призначення та умови застосування програмного забезпечення.

У п'ятому розділі розглянуто характеристики програми, спосіб звернення до плагінів. Крім того, було розглянуто написану таблицю для бази даних, і використовувані вбудовані таблиці бази даних Moodle. Були представлені алгоритми програми. Визначено вхідні та вихідні дані для обох ролей. Написано таблицю з описом можливих помилок під час використання плагіну для кожного з плагінів.

У шостому розділі було розглянуто експлуатацію програми оператором, призначення та умову використання програми, доступні функції для обох ролей.

На основі досліджень, проведених при написанні дипломної роботи було розроблено 2 плагіни мовою PHP. Також всередині обгортки php були написані запити до бази даних мовою sql, html сторінки з функціоналом, та невеликі скрипти на javascript. Для отримання повного функціоналу плагінів необхідно встановити їх у Moodle у послідовності: плагін-блок, плагін-звіт. Плагіни підтримуються у системі Moodle з версії 2.0.

Практична цінність роботи становить можливість проводити рецензування курсів, що допоможе підвищити якість навчання студентів.

## ПЕРЕЛІК ДЖЕРЕЛ ТА ПОСИЛАННЯ

1. Learning Management System [Електрон. ресурс]. – Режим доступу: [https://en.wikipedia.org/wiki/Learning\\_management\\_system](https://en.wikipedia.org/wiki/Learning_management_system).
2. Social Isolation and Acceptance of the Learning Management System (LMS) in the time of COVID-19 Pandemic [Електрон. ресурс] / Syed A. Raza. – Режим доступу: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7509242/>.
3. learning management system [Електрон. ресурс]. – Режим доступу: <https://www.techtarget.com/searchcio/definition/learning-management-system>.
4. one minute scorm overview [Електрон. ресурс]. – Режим доступу: <https://scorm.com/scorm-explained/one-minute-scorm-overview/>.
5. Sharable Content Object Reference Model [Електрон. ресурс]. – Режим доступу: [https://en.wikipedia.org/wiki/Sharable\\_Content\\_Object\\_Reference\\_Model](https://en.wikipedia.org/wiki/Sharable_Content_Object_Reference_Model).
6. technical scorm [Електрон. ресурс]. – Режим доступу: <https://scorm.com/scorm-explained/technical-scorm/>.
7. overview харі [Електрон. ресурс]. – Режим доступу: <https://харі.com/overview/>.
8. scorm vs the experience арі харі [Електрон. ресурс]. – Режим доступу: <https://харі.com/scorm-vs-the-experience-арі-харі/>.
9. Google Classroom [Електрон. ресурс]. – Режим доступу: [https://en.wikipedia.org/wiki/Google\\_Classroom](https://en.wikipedia.org/wiki/Google_Classroom).
10. Moodle [Електрон. ресурс]. – Режим доступу: <https://en.wikipedia.org/wiki/Moodle>.
11. learning management system [Електрон. ресурс]. – Режим доступу: <https://www.adamenfroy.com/learning-management-system>.
12. best learning management systems [Електрон. ресурс]. – Режим доступу: <https://thedigitalprojectmanager.com/tools/best-learning-management-systems/>.

13. Docebo [Электрон. ресурс]. – Режим доступа:  
<https://en.wikipedia.org/wiki/Docebo>.

14. Recommended Reading [Электрон. ресурс]. – Режим доступа:  
[https://www.softwaretestinghelp.com/learning-management-system/#Recommended\\_Reading](https://www.softwaretestinghelp.com/learning-management-system/#Recommended_Reading).

15. get pricing [Электрон. ресурс]. – Режим доступа:  
<https://www.absorblms.com/get-pricing>.

16. pricing [Электрон. ресурс]. – Режим доступа:  
<https://www.docebo.com/pricing/>.

17. prices [Электрон. ресурс]. – Режим доступа:  
<https://www.talentlms.com/prices>.

18. Installing Moodle [Электрон. ресурс]. – Режим доступа:  
[https://docs.moodle.org/400/en/Installing\\_Moodle](https://docs.moodle.org/400/en/Installing_Moodle).

19. Performance recommendations [Электрон. ресурс]. – Режим доступа:  
[https://docs.moodle.org/400/en/Performance\\_recommendations](https://docs.moodle.org/400/en/Performance_recommendations).

20. Moodle 4.0 release notes [Электрон. ресурс]. – Режим доступа:  
[https://docs.moodle.org/dev/Moodle\\_4.0\\_release\\_notes](https://docs.moodle.org/dev/Moodle_4.0_release_notes).

21. faq.general [Электрон. ресурс]. – Режим доступа:  
<https://www.php.net/manual/en/faq.general.php>.

22. what-is-nodejs [Электрон. ресурс]. – Режим доступа:  
<https://www.educative.io/blog/what-is-nodejs>

23. backend languages [Электрон. ресурс]. – Режим доступа:  
<https://blog.back4app.com/backend-languages/>.

24. JavaScript [Электрон. ресурс]. – Режим доступа:  
<https://en.wikipedia.org/wiki/JavaScript>.

25. TypeScript [Электрон. ресурс]. – Режим доступа:  
<https://en.wikipedia.org/wiki/TypeScript>.

26. CoffeeScript [Электрон. ресурс]. – Режим доступа:  
<https://en.wikipedia.org/wiki/CoffeeScript>.

27. introduction [Электрон. ресурс]. – Режим доступа: <https://coffeescript.org/#introduction>.

28. javascript vs coffeescript vs typescript [Электрон. ресурс]. – Режим доступа: <https://bytescout.com/blog/2016/07/javascript-vs-coffeescript-vs-typescript.html>.

29. HTML [Электрон. ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/HTML>.

30. CSS [Электрон. ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/CSS>.

31. documentation [Электрон. ресурс]. – Режим доступа: <https://sass-lang.com/documentation>.

32. how to use scss in html [Электрон. ресурс]. – Режим доступа: <https://w3codemasters.in/how-to-use-scss-in-html/>.

33. sass-vs-scss [Электрон. ресурс]. – Режим доступа: <https://www.educba.com/sass-vs-scss/>.

34. the definitive guide to scss [Электрон. ресурс]. – Режим доступа: <https://blog.logrocket.com/the-definitive-guide-to-scss/>.

35. Plugin types [Электрон. ресурс]. – Режим доступа: [https://docs.moodle.org/dev/Plugin\\_types](https://docs.moodle.org/dev/Plugin_types).

36. Plugin files [Электрон. ресурс]. – Режим доступа: [https://docs.moodle.org/dev/Plugin\\_files](https://docs.moodle.org/dev/Plugin_files).

37. Installing plugins [Электрон. ресурс]. – Режим доступа: [https://docs.moodle.org/310/en/Installing\\_plugins](https://docs.moodle.org/310/en/Installing_plugins).

**ДОДАТОКА**  
**Текст програми**

```
//block_corse_review_vote.php

class block_course_review_vote extends block_base {
    public function init() {
        $this->title = get_string('pluginname', 'block_course_review_vote');
    }

    public function instance_allow_multiple() {
        return false;
    }

    public function has_config() {
        return true;
    }

    public function applicable_formats() {
        return array('course-view' => true);
    }

    public function get_content() {
        global $CFG, $COURSE, $USER;

        $content = "\n";

        $pathtoblock = $CFG->wwwroot.'/blocks/course_review_vote/';
        require_once($CFG->dirroot.'/blocks/course_review_vote/libvote.php');
        if (has_capability('block/course_review_vote:admin',
            get_context_instance(CONTEXT_COURSE, $COURSE->id))) {
            if (DEBUG) {
                $content .= '<p style="color:#f00;font-weight:bold;text-align:center;">CAPABILITY IS ADMIN</p>'. "\n";
            }

            $content .= "\n". '<div class="center">';
            $content .= '<a href="'. $CFG->wwwroot.'/report/course_review/report.php?q=v&c='.$COURSE->id.'&s=c">'.
                get_string('admin-reportcourse',
                    'block_course_review_vote'). '</a><br />'. "\n";
            $content .= '<a href="'. $CFG->wwwroot.'/report/course_review/index.php">'.
                get_string('admin-reportall',
                    'block_course_review_vote'). '</a>'. "\n";
            $content .= '</div>';

            $content .= get_notes($COURSE->id, true);
            $content .= get_votes_summary($COURSE->id, true);

            $content .= get_didnot_voted_summary($COURSE->id, true);
        } else {
            if (DEBUG) {
                $content .= '<p style="color:#f00;font-weight:bold;text-align:center;">CAPABILITY IS VOTE</p>'. "\n";
            }
        }
    }
}
```



```

        $content .= '<input type="submit" class="rate rate-button
r'.$j.'" name="vote'.$j.'" value="'.$j.'"/>'.$j.'"';
    }
    $content .= '</div></form>'.$j.'"';
}

}

    $content                                     .=                '<div
class="center">'.get_course_score_average($COURSE->id). '</div>';
    $this->content                                = new stdClass;
    $this->content->text                          = $content;
    $this->content->footer                        = '';
    return $this->content;
}
}
//libvote.php

function get_users_of_course($course) {
    global $DB;
    $enrolled = $DB->get_records_sql("
SELECT course.id, user.id, user.lastname, user.firstname

FROM course
JOIN context ON course.id = context.instanceid
JOIN role_assignments ON role_assignments.contextid = context.id
JOIN user ON user.id = role_assignments.userid
JOIN role ON role.id = role_assignments.roleid

WHERE course.id = ".$course);
    return $enrolled;
}

function get_course_score_average($course) {
    global $COURSE, $DB;
    $res = $DB->get_records_select('block_course_review_vote', 'deleted = \'0\'
and course_id = \''.$course.'\'', array('id, vote'));
    if ($res) {
        $votes_tot = 0;
        foreach ($res as $row) {
            $votes_tot += $row->vote;
        }
    }

    $votes_no = $DB->count_records('block_course_review_vote', array(
'course_id'=>$course,
'deleted'=>0
));
    if ($votes_no) {
        $votes_avg = $votes_tot / $votes_no;
    }
}

```

```

    $votes_avg = substr($votes_avg, 0, 3);
    $enrolled = get_users_of_course($course);
    $count = 0;
    foreach($enrolled as $i){
        $count++;
    }
    $content = get_string('scoreavg1', 'block_course_review_vote');
    $content .= $votes_no.get_string('scoreavg2',
'block_course_review_vote').$count.get_string('scoreavg3', 'block_course_review_vote');
    //Расчёт подцветки средней оценки(красный - жёлтый - зелёный)
    $colorR = 0;
    $colorG = 0;
    $colorB = 0;
    if($votes_avg < 0){
        $colorG = ($votes_avg +2)/2*255;
        $colorR = 255;
    }
    else {
        $colorG = 255;
        $colorR = (2 -$votes_avg)/2*255;
    }
    $content .= '<div class="rate" style="border: 7px solid rgba('.$colorR.',
'.$colorG.', '.$colorB.')"; width: 70px; height: 40px;">'.$votes_avg.</div>';
    } else {
        if (has_capability('block/course_review_vote:admin',
get_context_instance(CONTEXT_COURSE, $COURSE->id)) ) {
            $content = get_string('scoreavgaltnonstudent',
'block_course_review_vote');
        } else {
            $content = get_string('scoreavgalt', 'block_course_review_vote');
        }
    }
    return $content;
}

function has_voted($user, $course) {
    global $DB;
    return $DB->record_exists('block_course_review_vote', array(
        'user_id'=>$user,
        'course_id'=>$course,
        'deleted'=>0
    ));
}

function get_vote($user, $course) {
    global $DB;
    $res = $DB->get_record('block_course_review_vote', array('user_id'=>$user,
'course_id'=>$course, 'deleted'=>0), 'vote');
    return $res->vote;
}

```

```

    }

    function get_vote_id($user, $course) {
        global $DB;
        $res = $DB->get_record('block_course_review_vote', array('user_id'=>$user,
'course_id'=>$course, 'deleted'=>0), 'id');
        return $res->id;
    }

    function get_vote_note($user, $course) {
        global $DB;
        $res = $DB->get_record('block_course_review_vote', array('user_id'=>$user,
'course_id'=>$course, 'deleted'=>0), 'note');
        if (isset($res->note)) {
            return $res->note;
        } else {
            return false;
        }
    }

    function get_notes($cid, $deleted = false) {
        global $DB;
        if ($deleted == false) {
            $del = "deleted = '0' AND ";
        } else {
            $del = '';
        }
        $res = $DB->get_records_select('block_course_review_vote', $del.'course_id =
\'\'.$cid.\'\' AND note <> \'\'\'',
            array('id ASC', 'id, note, deleted'));
        if ($res) {
            $content = '<div class="center smaller">

<a
href="javascript:hideshow(document.getElementById(\'course_review_vote_feedback\'))">'.
            get_string('note_get', 'block_course_review_vote').'</a>

</div>
<script type="text/javascript">
    //<![CDATA[
        function hideshow(which){
            if (!document.getElementById)
                return
            if (which.style.display=="block")
                which.style.display="none"
            else
                which.style.display="block"
        }
    //]]>

</script>

<div id="course_review_vote_feedback" class="bgborder" style="display:
none;"><ul>'. "\n";

```

```

        foreach ($res as $row) {
            if (!empty($row->note)) {
                if (!$row->deleted) {
                    $data = $DB->get_record('user', array('id'=>$row->user_id));
                    $content .= '<li><a href="'. $CFG->wwwroot.'/user/view.php?id='.$row->user_id.'&course='.$cid.'">'. $data->firstname.''. $data->lastname.'</a> - &ldquo;<em>'. $row->note.'</em>&rdquo;';
                    $content .= '</li>'. "\n";
                }
            }
        }
        $content .= '</ul></div>'. "\n";
        return $content;
    } else {
        return '<div class="center smaller">'. get_string('note_none', 'block_course_review_vote'). '</div>';
    }
}

function get_votes_summary($cid, $deleted = false) {
    global $CFG, $DB;
    $res = $DB->get_records_select('block_course_review_vote', 'course_id = \''.$cid.'\' AND deleted = 0', array('id ASC', 'id, vote'));
    if ($res) {
        $content = '<div class="center smaller">

        <a
href="javascript:hideshow(document.getElementById(\'course_review_vote_summary\'))">'.
            get_string('admin-votesummary', 'block_course_review_vote'). '</a>

        </div>

        <script type="text/javascript">
            function hideshow(which){
                if (!document.getElementById)
                    return
                if (which.style.display=="block")
                    which.style.display="none"
                else
                    which.style.display="block"
            }
        </script>

        <div id="course_review_vote_summary" class="center slightborder" style="display: none;">'. "\n";

        foreach ($res as $row) {
            $data = $DB->get_record('user', array('id'=>$row->user_id));
            $content .= '<a href="'. $CFG->wwwroot.'/user/view.php?id='.$row->user_id.'&course='.$cid.'">
                <div style="margin: 5px;">'. $data->firstname.''. $data->lastname.'</a> - <div
                class="rate r'.$row->vote.
                '" style="display: inline-block;">'. $row->vote.'</div></div>'. "\n";
        }
    }
}

```

```

    }
    $content .= '</div>'. "\n";
    return $content;
} else {
    return '<div class="center smaller">'.get_string('admin-novotesummary',
'block_course_review_vote').'</div>';
}
}

function get_didnot_voted_summary($cid, $deleted = false) {
global $CFG, $DB;

$enrolled = get_users_of_course($cid);
$res = $DB->get_records_select('block_course_review_vote', 'course_id =
\''.$cid.'\' AND deleted = 0', array('id ASC', 'id, vote'));
if (count($enrolled) != count($res)) {
    $content = '<div class="center smaller">

<a
href="javascript:hideshow(document.getElementById(\'coursereview_didnot_voted_summary\'))
">'.

        get_string('admin-didnotvotedummary',
'block_course_review_vote').'</a>
</div>
<script type="text/javascript">
    function hideshow(which){
        if (!document.getElementById)
            return
        if (which.style.display=="block")
            which.style.display="none"
        else
            which.style.display="block"
    }
</script>
<div id="coursereview_didnot_voted_summary" class="center slightborder"
style="display: none;">'. "\n";

        $status = true;
        foreach($enrolled as $i){
            foreach($res as $j){
                if($j->user_id == $i->id) {
                    $status = false;
                    break;
                }
            }
        }
        if($status){
            $data = $DB->get_record('user', array('id'=>$i->id));
            $content .= '<a href="'. $CFG->wwwroot.'/user/view.php?id='.$i->id.'&course='.$cid.'">

```

```

        <div                style="margin:                5px;">'.$data->firstname.''.$data-
>lastname.'</div></a>'.$"\n";
    }
    $status = true;
}
$content .= '</div>'.$"\n";
return $content;
} else {
    return '<div                class="center                smaller">'.$get_string('admin-
nodidnotvotedummary', 'block_course_review_vote').'</div>';
}
}
//vote.php
if (!$course = $DB->get_record('course', array('id'=>required_param('cid',
PARAM_INT)))) {
    print_error(get_string('error-courseidnotset', 'block_course_review_vote'));
}
require_login($course);
if (get_config('course_review_vote', 'note') == true) {
    if (optional_param('vote-2', '', PARAM_INT)) {
        $vote = -2;
    } else if (optional_param('vote-1', '', PARAM_INT)) {
        $vote = -1;
    } else if (optional_param('vote0', '', PARAM_INT) === 0) {
        $vote = 0;
    } else if (optional_param('vote1', '', PARAM_INT)) {
        $vote = 1;
    } else if (optional_param('vote2', '', PARAM_INT)) {
        $vote = 2;
    } else {
        print_error(get_string('error-iefixfail', 'block_course_review_vote'));
    }
} else {
    $vote = required_param('vote', PARAM_INT);
}
$note = optional_param('note', '', PARAM_NOTAGS);
if ($vote < -2 || $vote > 2) {
    print_error(get_string('error-voteoutofrange', 'block_course_review_vote'));
}
if (!$USER->id) {
    print_error(get_string('error-useridnotset', 'block_course_review_vote'));
}
$now = time();
class Vote{
    public $user_id;
    public $course_id;
    public $vote;

```

```

        public $date_added;
        public $date_modified;
        public $note;
    }

    $dbinsert = new Vote();
    $dbinsert->user_id      = $USER->id;
    $dbinsert->course_id    = $COURSE->id;
    $dbinsert->vote         = $vote;
    $dbinsert->date_added   = $now;
    $dbinsert->date_modified = $now;
    $dbinsert->note         = $note;
    if (!$DB->insert_record('block_course_review_vote', $dbinsert)) {
        print_error(get_string('error-dbinsert', 'block_course_review_vote'));
    } else {
        redirect($CFG->wwwroot.'/course/view.php?id='.$COURSE->id);
    }
    //unvote.php
    if (!$course = $DB->get_record('course', array('id'=>required_param('cid',
PARAM_INT)))) {
        print_error(get_string('error-courseidnotset', 'block_course_review_vote'));
    }
    require_login($course);
    require_once($CFG->dirroot.'/blocks/course_review_vote/libvote.php');
    if (!$USER->id) {
        print_error(get_string('error-useridnotset', 'block_course_review_vote'));
    }
    if (!$vote_id = get_vote_id($USER->id, $course->id)) {
        print_error(get_string('error-novoteid', 'block_course_review_vote'));
    }
    class UnVote{
        public $id;
        public $date_modified;
        public $deleted;
        public $deleted_user_id;
    }
    $dbupdate = new UnVote();
    $dbupdate->id          = $vote_id;
    $dbupdate->date_modified = time();
    $dbupdate->deleted      = 1;
    $dbupdate->deleted_user_id = $USER->id;
    if (!$DB->update_record('block_course_review_vote', $dbupdate)) {
        print_error(get_string('error-dbupdate', 'block_course_review_vote'));
    } else {
        redirect($CFG->wwwroot.'/course/view.php?id='.$course->id);
    }
    //admin.php
    $qid = required_param('q', PARAM_ALPHA);

```

```

class Vote{
    public $id;
    public $date_modified;
    public $note;
}

class UnVote{
    public $id;
    public $date_modified;
    public $deleted;
    public $deleted_user_id;
}

if (strtolower($qid) == 'delvote') {
    $vid    = required_param('v', PARAM_INT);
    $course = optional_param('c', '', PARAM_INT);
    $user    = optional_param('u', '', PARAM_INT);
    $dbupdate = new UnVote();
    $dbupdate->id                = $vid;
    $dbupdate->date_modified     = time();
    $dbupdate->deleted           = 1;
    $dbupdate->deleted_user_id   = $USER->id;
    if (!$DB->update_record('block_course_review_vote', $dbupdate)) {
        print_error(get_string('error_dbupdate', 'report_course_review'));
    } else {
        if (isset($user) && !empty($user)) {
            redirect($CFG->
>wwwroot.'/report/course_review/report.php?q=v&u='.$user);
        } else if (isset($course) && !empty($course)) {
            redirect($CFG->
>wwwroot.'/report/course_review/report.php?q=v&c='.$course);
        } else {
            redirect($CFG->wwwroot.'/report/course_review/');
        }
    }
} else if (strtolower($qid) == 'noteswipe') {
    if ($res = $DB->get_records_select('block_course_review_vote', 'deleted =
0 AND note <> \'' . array(''), '', 'id', '', '')) {
        foreach ($res as $row) {
            $now = time();
            $dbupdate = new Vote();
            $dbupdate->id                = $row->id;
            $dbupdate->date_modified     = $now;
            $dbupdate->note              = '';
            if (!$DB->update_record('block_course_review_vote',
$dbupdate)) {
                print_error(get_string('error_dbupdate',
'report_course_review'));
            }
        }
    }
}

```

```

    }
    redirect($CFG->wwwroot.'/report/course_review/');
} else {
    print_error(get_string('error_nonotes', 'report_course_review'));
}
} else if (strtolower($qid) == 'voteremove') {
    if ($res = $DB->get_records('block_course_review_vote', array('deleted'=>0),
'', 'id', '', '')) {
        foreach ($res as $row) {
            $now = time();
            $dbupdate = new UnVote();
            $dbupdate->id = $row->id;
            $dbupdate->date_modified = $now;
            $dbupdate->deleted = 1;
            $dbupdate->deleted_user_id = $USER->id;
            if (!$DB->update_record('block_course_review_vote',
$dbupdate)) {
                print_error(get_string('error_dbupdate',
'report_course_review'));
            }
        }
        redirect($CFG->wwwroot.'/report/course_review/');
    } else {
        print_error(get_string('error_novotes', 'report_course_review'));
    }
} else if (strtolower($qid) == 'votedelete') {
    if ($DB->delete_records('block_course_review_vote',
array('deleted'=>1))) {
        redirect($CFG->wwwroot.'/report/course_review/');
    }
} else if (strtolower($qid) == 'wipcoursevotes') {
    $wid = required_param('w', PARAM_INT);
    if (!$DB->delete_records('block_course_review_vote',
array('course_id'=>$wid))) {
        print_error(get_string('error_noanyvotes', 'report_course_review'));
    } else {
        redirect($CFG->wwwroot.'/report/course_review/');
    }
}
}
//index.php
require_login();
require_capability('moodle/site:viewreports',
get_context_instance(CONTEXT_SYSTEM));
admin_externalpage_setup('report_course_review', '', null, '',
array('pagelayout'=>'report'));
$content = '<h3>'.get_string('coursewith', 'report_course_review').'</h3>'. "\n";
$content .= '<ul>'. "\n";

```

```

$content .= '<li><a href="report.php?q=c&l=0&s=a">'
    .get_string('listvote_statistic', 'report_course_review').'</a></li>'. "\n";
    if ($res = $DB->get_record('block', array('name'=>'course_review_vote'))) {
        $vid = $res->id;
    } else {
        print_error(get_string('error_noblock', 'report_course_review'));
        die();
    }
    $content
        .=
        '<li><a
                                href="'. $CFG-
>wwwroot.'/course/search.php?blocklist='.$vid.'&sesskey='.$USER->sesskey.'">'
        .get_string('listvote', 'report_course_review').'</a></li>'. "\n";
    $content .= "</ul>\n";
    echo $OUTPUT->header();
    echo
        $OUTPUT->heading(format_string(get_string('course_reviewtoctitle',
'report_course_review')));
    echo $OUTPUT->box_start('generalbox boxwidthwide boxaligncenter');
    echo $content;
    echo $OUTPUT->box_end();
    $content_admin
        .=
        '<h3>'.get_string('admin_notestitle',
'report_course_review').'</h3>'. "\n";
    $content_admin .= "<ul>\n";
    $content_admin .= '<li><a href="admin.php?q=noteswipe">'.
        get_string('admin_removeallnotes', 'report_course_review').'</a></li>\n';
    $content_admin .= "</ul>\n";
    $content_admin
        .=
        '<h3>'.get_string('admin_votestitle',
'report_course_review').'</h3>'. "\n";
    $content_admin .= "<ul>\n";
    $content_admin .= '<li><a href="admin.php?q=voteremove">'.
        get_string('admin_removeallvotes', 'report_course_review').'</a></li>\n';
    $content_admin .= '<li><a href="admin.php?q=votedelete">'.
        get_string('admin_deleteallvotehistory',
'report_course_review').'</a></li>\n';
    $content_admin .= "</ul>\n";
    $content_admin
        .=
        '<h3>'.get_string('admin_courseclearingtitle',
'report_course_review').'</h3>'. "\n";
    $content_admin .= 'Remove all votes from: ';
    $content_admin
        .=
        '<form
                                name="wipecourse"
                                method="get"
action="admin.php"><select name="w">';
    $res = $DB->get_records_sql('SELECT DISTINCT course_id FROM '. $CFG-
>prefix.'block_course_review_vote ORDER BY course_id ASC');
    if ($res) {
        foreach ($res as $row) {
            $res2 = $DB->get_record_select('course', 'id = '.$row->course_id,
array('fullname'));
            if ($res2) {
                $content_admin
                    .=
                    '<option
                                value="'. $row->course_id.'">['.$row-
>course_id.'] '.$res2->fullname.'</option>'. "<br />\n";
            }
        }
    }

```

```

    }
}
$content_admin .= '</select>';
$content_admin .= '<input type="hidden" name="q" value="wipcoursevotes" />';
$content_admin .= '<input type="submit" value="Clear this course" />';
$content_admin .= '</form>';
} else {
    $content_admin .= '<option disabled="disabled" selected="selected">No courses
with votes.</option>';
    $content_admin .= '<input disabled="disabled" type="submit" value="Clear this
course" />';
    $content_admin .= '</select></form>';
}
$content_admin .= '<h3>'.get_string('stats',
'report_course_review').'</h3>'. "\n";
$content_admin .= '<ul>'. "\n";
$vlive = $DB->get_record_sql("SELECT COUNT(id) AS cid FROM ".
    $CFG->prefix."block_course_review_vote WHERE deleted = '0'");
$content_admin .= '<li>'.get_string('debugvotes', 'report_course_review').
    get_string('debuglive', 'report_course_review').$vlive->cid;
$vdel = $DB->get_record_sql("SELECT COUNT(id) AS cid FROM ".
    $CFG->prefix."block_course_review_vote WHERE deleted = '1'");
$content_admin .= '; '.get_string('debugdeleted', 'report_course_review').$vdel-
>cid;
$content_admin .= '; '.get_string('debugtotal', 'report_course_review').($vlive-
>cid + $vdel->cid).'</li>'. "\n";
$nlive = $DB->get_record_sql("SELECT COUNT(id) AS cid FROM ".
    $CFG->prefix."block_course_review_vote WHERE deleted = '0' AND note <>'");
$content_admin .= '<li>'.get_string('debugnotes', 'report_course_review').
    get_string('debuglive', 'report_course_review').$nlive->cid;
$ndel = $DB->get_record_sql("SELECT COUNT(id) AS cid FROM ".
    $CFG->prefix."block_course_review_vote WHERE deleted = '1' AND note <>'");
$content_admin .= '; '.get_string('debugdeleted', 'report_course_review').$ndel-
>cid;
$content_admin .= '; '.get_string('debugtotal', 'report_course_review').($nlive-
>cid + $ndel->cid).'</li>'. "\n";
$uno = $DB->get_record_sql("SELECT COUNT(DISTINCT user_id) AS cid FROM ".$CFG->
prefix."block_course_review_vote");
$cno = $DB->get_record_sql("SELECT COUNT(DISTINCT course_id) AS cid FROM ".$CFG->
prefix."block_course_review_vote");
$content_admin .= '<li>'. $uno->cid.get_string('debugusers',
'report_course_review').$cno->cid.
    get_string('debugcourses', 'report_course_review').'</li>'. "\n";
echo $OUTPUT->heading(format_string(get_string('admin_toc',
'report_course_review')));
echo $OUTPUT->box_start('generalbox boxwidthwide boxaligncenter');
echo $content_admin;

```

```

echo $OUTPUT->box_end();
echo $OUTPUT->footer();
//report.php
$qid = required_param('q', PARAM_ALPHA);
$now_fmt = "Y-m-d H:i:s";
$now = date($now_fmt, time());
$position = 0;
define('PREFIX', $CFG->prefix);
define('TBL_REVIEW', 'block_course_review_vote');
define('PATH_COURSE', $CFG->wwwroot.'/course/view.php?id=');
define('PATH_USER', $CFG->wwwroot.'/user/view.php?id=');
define('PATH_REPORT', $CFG->wwwroot.'/report/course_review');
define('SORT_IMG', PATH_REPORT.'/img/sorting-order.png');
define('DEL_IMG', PATH_REPORT.'/img/remove.png');
function get_course_shortcode($id) {
    global $DB;
    $res = $DB->get_record('course', array('id'=>$id));
    return '<a href="'.PATH_COURSE.$id.'">'.$res->shortcode.'</a>';
}
class Course {
    public $course_id;
    public $shortcode;
    public $fullname;
    public $votecount;
    public $votecountdeleted;
    public $notecount;
    public $notecountdeleted;
    public $voteavg;
    public $usersofcourse;
    public $percent;
    public $containBlock;
    function __construct($course_id, $shortcode,
        $fullname,$votecount,$votecountdeleted,$notecount,$notecountdeleted,$voteavg) {
        $this->course_id = $course_id;
        $this->shortcode = $shortcode;
        $this->fullname = $fullname;
        $this->votecount = $votecount;
        $this->votecountdeleted = $votecountdeleted;
        $this->notecount = $notecount;
        $this->notecountdeleted = $notecountdeleted;
        $this->voteavg = $voteavg;
        $this->getUsersOfCourse();
        $this->getPercent();
        $this->getContainBlock();
    }
    function getUsersOfCourse(){
        global $DB;

```

```

        $enrolled = $DB->get_records_sql("
SELECT course.id, user.id, user.lastname, user.firstname

FROM course
JOIN context ON course.id = context.instanceid
JOIN role_assignments ON role_assignments.contextid = context.id
JOIN user ON user.id = role_assignments.userid
JOIN role ON role.id = role_assignments.roleid

WHERE course.id = ".$this->course_id.";");
$count = 0;
foreach($enrolled as $i){
    $count++;
}
$this->usersofcourse = $count;
}
function getPercent(){
    $this->percent = round($this->votecount/$this->usersofcourse*100);
}
function getContainBlock(){
    global $DB;
    $contain = $DB->get_records_sql("
        SELECT course.id

FROM course
JOIN context ON course.id = context.instanceid
JOIN block_instances ON block_instances.parentcontextid = context.id

WHERE course.id = ".$this->course_id."
AND block_instances.blockname = 'course_review_vote'");
    $this->containBlock = $contain ? true : false;
}
}
function usersSort( $a, $b ) {
    return $b->usersofcourse - $a->usersofcourse;
}
function percentSort( $a, $b ) {
    return $b->percent - $a->percent;
}
$content = '';
$content_heading = '';
$content .= '<link rel="stylesheet" type="text/css" href="styles.css" />'. "\n";
$content .= "\n". '<div id="report_course_review">'. "\n";
if (strtolower($qid) == 'c') {
    $sort = required_param('s', PARAM_ALPHA);
    $course = optional_param('c', '', PARAM_INT);
    if (is_numeric($course) && $course > 0) {

```

```

        define('SCV', true);
    } else {
        define('SCV', false);
    }

    $query = "SELECT course_id, shortname, fullname, (
                SELECT COUNT(vote)
                FROM ".PREFIX.TBL_REVIEW."
                WHERE deleted = '0'
                AND ".PREFIX.TBL_REVIEW.".course_id = ".PREFIX."course.id
                GROUP BY course_id
            ) AS votecount, (
                SELECT COUNT(vote)
                FROM ".PREFIX.TBL_REVIEW."
                WHERE deleted = '1'
                AND ".PREFIX.TBL_REVIEW.".course_id = ".PREFIX."course.id
                GROUP BY course_id
            ) AS votecountdeleted, (
                SELECT COUNT(note)
                FROM ".PREFIX.TBL_REVIEW."
                WHERE deleted = '0'
                AND note <>' '
                AND ".PREFIX.TBL_REVIEW.".course_id = ".PREFIX."course.id
                GROUP BY course_id
            ) AS notecount, (
                SELECT COUNT(note)
                FROM ".PREFIX.TBL_REVIEW."
                WHERE deleted = '1'
                AND note <>' '
                AND ".PREFIX.TBL_REVIEW.".course_id = ".PREFIX."course.id
                GROUP BY course_id
            ) AS notecountdeleted, (
                SELECT AVG(vote)
                FROM ".PREFIX.TBL_REVIEW."
                WHERE deleted = '0'
                AND ".PREFIX.TBL_REVIEW.".course_id = ".PREFIX."course.id
                GROUP BY course_id
            ) AS voteavg
            FROM ".PREFIX.TBL_REVIEW.", ".PREFIX."course
            WHERE ".PREFIX.TBL_REVIEW.".course_id = ".PREFIX."course.id ";

    if (SCV) {
        $query .= "AND ".PREFIX."course.id = '". $course. "'";
    }

    $query .= "GROUP BY ".PREFIX."course.id, course_id, shortname, fullname ";

    if (SCV) {
        $title = get_string('coursereport_single', 'report_course_review');
    } else if ($sort == 's') {
        $query .= "ORDER BY voteavg DESC, votecount DESC";
    }

```

```

        $title = get_string('coursereport_highscore', 'report_course_review');
    } else if ($sort == 'v') {
        $query .= "ORDER BY votecount DESC, voteavg DESC";
        $title = get_string('coursereport_mostvoted', 'report_course_review');
    } else if ($sort == 'd') {
        $query .= "ORDER BY votecountdeleted DESC, votecount DESC";
        $title = get_string('coursereport_mostdeleted', 'report_course_review');
    } else if ($sort == 'n') {
        $query .= "ORDER BY notecount DESC, voteavg DESC";
        $title = get_string('coursereport_mostnoted', 'report_course_review');
    } else if ($sort == 'a') {
        $query .= "ORDER BY shortname ASC, voteavg DESC";
        $title = get_string('coursereport_list', 'report_course_review');
    }
    $query .= ";";

    $data = $DB->get_records_sql($query);
    $res = array();
    foreach($data as $element){
        array_push($res, new Course(
            $element->course_id,
            $element->shortname,
            $element->fullname,
            $element->votecount,
            $element->votecountdeleted,
            $element->notecount,
            $element->notecountdeleted,
            $element->voteavg
        ));
    }

    $courses = $DB->get_records_sql("
    SELECT course.id, course.shortname, course.fullname

    FROM course
    JOIN context ON c.id = context.instanceid
    JOIN block_instances ON block_instances.parentcontextid = context.id

    WHERE block_instances.blockname = 'course_review_vote'");
    $contain = false;
    foreach($courses as $i){
        foreach($res as $j){
            if($i->id == $j->course_id){
                $contain = true;
                break;
            }
        }
    }
    if(!$contain){

```

```

        array_push($res, new Course(
            $i->id,
            $i->shortname,
            $i->fullname,
            0, 0, 0, 0, 0
        ));
    }
    $contain = false;
}

if (!$res) {
    $content .= '<p>'.get_string('noresults',
'report_course_review').'</p>'. "\n";
    die($content);
}

if ($sort == 'u') {
    usort($res, 'usersSort');
    $title = get_string('coursereport_usersofcourse',
'report_course_review');
}

else if ($sort == 'p') {
    usort($res, 'percentSort');
    $title = get_string('coursereport_percent', 'report_course_review');
}

$content_heading = $OUTPUT->heading($title.' ('.get_string('asof',
'report_course_review').$now.'))';
$content .= '<p>'.get_string('sortimg1', 'report_course_review').'.
    get_string('sortimg2', 'report_course_review').'</p>'. "\n";
    $content .= '<table>'. "\n".<tr>'. "\n";
        if (SCV == false) {
            $content .= '<th>'.get_string('position',
'report_course_review').'</th>'. "\n";
        }
        if ($sort == 'a') {
            $content .= '<th>'.get_string('course', 'report_course_review').</th>'. "\n";
        } else {
            $content .= '<th><a href="report.php?q=c&s=a">'.get_string('course',
'report_course_review').'</a></th>'. "\n";
        }
        if ($sort == 's') {
            $content .= '<th>'.get_string('score', 'report_course_review').</th>'. "\n";
        } else {
            $content .= '<th><a href="report.php?q=c&s=s">'.get_string('score',
'report_course_review').'</a></th>'. "\n";
        }
    }
}

```



```

        $content      .=      '      ('.$row->votecountdeleted.get_string('deleted',
'report_course_review').')';
    }
    $content .= '</a></td>'. cant . "\n";
    $content .= '<td><a href="report.php?q=v&c='.$row->course_id.'">';
    if (!$row->notecount > 0) {
        $content .= '0'. cant . "\n";
    } else {
        $content .= $row->notecount;
    }
    if ($row->notecountdeleted <> 0) {
        $content      .=      '      ('.$row->notecountdeleted.get_string('deleted',
'report_course_review').')';
    }
    $content .= '</a></td>'. cant . "\n";
    $content      .=      '<td><a      href="report.php?q=v&c='.$row-
>course_id.'">';
    $content .= $row->percent.'%';
    $content .= '</a></td>'. cant . "\n";
    $content .= '</tr>'. cant . "\n";
}

$content .= '</table>'. cant . "\n";
} else if (strtolower($qid) == 'v') {
    if ($user = optional_param('u', '', PARAM_INT)) {
        define('TYPE', 'user');
        $data = $DB->get_record('user', array('id'=>$user));
        $content_heading      =      $OUTPUT->heading(get_string('vnreport_title',
'report_course_review').$data->firstname.''.
        $data->lastname.'      ('.get_string('asof',
'report_course_review').$now.'')');
    } else if ($course = optional_param('c', '', PARAM_INT)) {
        define('TYPE', 'course');
        $data = $DB->get_record('course', array('id'=>$course));
        $content_heading      =      $OUTPUT->heading(get_string('vnreport_title',
'report_course_review').$data->fullname.' ('.
        $data->shortname.'')      ('.get_string('asof',
'report_course_review').$now.'')');
    } else {
        print_error(get_string('vnreport_errortype', 'report_course_review'));
    }

    $query = "SELECT vote, COUNT(vote) AS votecount, (
        SELECT COUNT(vote)
        FROM ".PREFIX.TBL_REVIEW."";
    if (TYPE == 'user') {
        $query .= "WHERE ".PREFIX.TBL_REVIEW.".user_id = '".$user."'";
    } else if (TYPE == 'course') {

```

```

$query = "WHERE ".PREFIX.TBL_REVIEW.".course_id = '". $course."'";
}
$query = "AND deleted = 0 )
        AS total
        FROM ".PREFIX.TBL_REVIEW."
        WHERE deleted = 0 ";
if (TYPE == 'user') {
    $query = "AND ".PREFIX.TBL_REVIEW.".user_id = '". $user."'";
} else if (TYPE == 'course') {
    $query = "AND ".PREFIX.TBL_REVIEW.".course_id = '". $course."'";
}
$query = "GROUP BY vote ORDER BY vote DESC;";
$res = $DB->get_records_sql($query);
if ($res) {
    $content .= "<h3>".get_string('vnreport_summary',
'report_course_review')."</h3>". "\n";
    $content .= "<table>". "\n";
    $content .= "<tr><th>".get_string('vnreport_vote',
'report_course_review')."</th><th>".
        get_string('vnreport_votecast', 'report_course_review')."</th><th>".
        get_string('vnreport_percentage',
'report_course_review')."</th></tr>". "\n";
    $score = 0;
    $chart = array();
    foreach ($res as $row) {
        $content .= "<tr><td><div class='rate' r'".$row->vote.'">".$row-
>vote.</div></td><td>".$row->votecount.
            "</td><td>".number_format(($row->votecount/$row->total)*100,
1).'%</td></tr>". "\n";
        $score += ($row->vote * $row->votecount)/$row->total;
        $chart[$row->vote] = $row->votecount;
    }
    $content .= "</table>". "\n";
    $content .= "<p>".get_string('vnreport_coursescore',
'report_course_review').number_format($score, 2)."</p>". "\n";
}

if(!$thoseWhoVoted = $DB->get_records_select('block_course_review_vote',
'deleted = \'0\' and course_id = \''. $course.'\'', array('id, vote')))$thoseWhoVoted =
false;

$content .= "<h3>".get_string('vnreport_whovoted',
'report_course_review')."</h3>". "\n";
$enrolled = $DB->get_records_sql("
SELECT course.id, user.id, user.lastname, user.firstname

FROM course
JOIN context ON course.id = context.instanceid

```

```

JOIN role_assignments ON role_assignments.contextid = context.id
JOIN user ON user.id = role_assignments.userid
JOIN role ON role.id = role_assignments.roleid

WHERE course.id = ".$course);
$content .= '<table><tbody>
<tr><th>'.get_string('name',
'report_course_review').</th><th>'.get_string('voted',
'report_course_review').</th></tr>';
$status = true;
foreach($enrolled as $i){
    if($thoseWhoVoted)
        foreach($thoseWhoVoted as $j){
            if($j->user_id == $i->id) {
                $status = false;
                break;
            }
        }
    $data = $DB->get_record('user', array('id'=>$i->id));
    $content .= '<tr><td><a href="'. $CFG->wwwroot.'/user/view.php?id='.$i-
>id.'&course='.$course.'">'
        . $data->firstname.' '.$data->lastname.'</a></td><td>'.($status ? "-" :
"+").</td></tr>'. "\n";
    $status = true;
}
$content .= '</tbody></table>'. "\n";
$query = "SELECT ".PREFIX.TBL_REVIEW.".id AS vote_id, firstname,
lastname, shortname, fullname, user_id,
        course_id, vote, date_added, date_modified, note,
".PREFIX.TBL_REVIEW.".deleted
        FROM ".PREFIX.TBL_REVIEW.", ".PREFIX."course, ".PREFIX."user
        WHERE ".PREFIX.TBL_REVIEW.".course_id = ".PREFIX."course.id
        AND ".PREFIX.TBL_REVIEW.".user_id = ".PREFIX."user.id ";
if (TYPE == 'user') {
    $query .= "AND ".PREFIX.TBL_REVIEW.".user_id = '". $user."'";
} else if (TYPE == 'course') {
    $query .= "AND ".PREFIX.TBL_REVIEW.".course_id = '". $course."'";
} else {
    print_error(get_string('error_query', 'report_course_review'));
}
$query .= "ORDER BY date_added DESC";
$res = $DB->get_records_sql($query);
if (!$res) {
    $content .= '<p>'.get_string('novotesnotes',
'report_course_review').</p>'. "\n";
} else {

```

```

        $content                .=                '<h3>'.get_string('vnreport_detail',
'report_course_review').</h3>'. "\n";
        $content                .=                '<p>'.get_string('vnreport_greytext',
'report_course_review').</p>'. "\n";
        $content .= '<table>'. "\n".<tr>'. "\n";
        $content                .=                '<th>'.get_string('position',
'report_course_review').</th>'. "\n";
        $content                .=                '<th>'.get_string('date',      'report_course_review').</th>'. "\n";
        $content                .=                '<th>'.get_string('name',
'report_course_review').</th>'. "\n";
        $content                .=                '<th>'.get_string('vnreport_vote',
'report_course_review').</th>'. "\n";
        $content                .=                '<th>'.get_string('vnreport_note',
'report_course_review').</th>'. "\n";
        $content                .=                '<th      class="delete">'.get_string('vnreport_del',
'report_course_review').</th>'. "\n";
        $content .= '</tr>'. "\n";
        foreach ($res as $row) {

            if ($row->deleted == 1) {
                $content .= '<tr class="deleted">'. "\n";
            } else {
                $content .= '<tr>'. "\n";
            }
            $content .= '<td>'.++$position.</td>'. "\n";
            $content                .=                '<td>'.get_string('vnreport_added',
'report_course_review').
                date($now_fmt, $row->date_added);
            if ($row->deleted == 1) {
                $content                .=                '<br
class="deleted">'.get_string('vnreport_deleted', 'report_course_review').
                date($now_fmt, $row->date_modified).</span>';
            }
            $content .= '</td>'. "\n";
            if (TYPE == 'course') {
                $content                .=                '<td><a      href="'.PATH_USER.$row->user_id.'">'. $row-
>firstname.''.
                $row->lastname.</a></td>'. "\n";
            } else if (TYPE == 'user') {
                $content                .=                '<td><a      href="'.PATH_COURSE.$row-
>course_id.'">'. $row->fullname.' ('.
                $row->shortname.')</a></td>'. "\n";
            }
            $content                .=                '<td><div      class="rate      r'. $row->vote.'">'. $row-
>vote.</div></td>'. "\n";
            if (!empty($row->note)) {

```

```

        $content .= '<td>&ldquo;'.$row->note.'&rdquo;</td>'. "\n";
    } else {
        $content .= '<td></td>'. "\n";
    }
    if ($row->deleted == 0) {
        if (TYPE == 'user') {
            $content .= '<td><a href="admin.php?q=delvote&v='.$row->vote_id.'&u='.$user.'">';
        } else if (TYPE == 'course') {
            $content .= '<td><a href="admin.php?q=delvote&v='.$row->vote_id.'&c='.$course.'">';
        }
        $content .= '</a></td>'. "\n";
    } else {
        $content .= '<td>&nbsp;</td>'. "\n";
    }
    $content .= '</tr>'. "\n";
}
$content .= '</table>'. "\n";
} else {
    $content .= get_string('error_noquery', 'report_course_review'). "\n";
}
$content .= '<p><a href="index.php">'.get_string('backtolisting', 'report_course_review').</a></p>'. "\n";
$content .= '</div>'. "\n";
echo $OUTPUT->header();
echo $content_heading;
echo $OUTPUT->box_start('generalbox boxwidthwide boxaligncenter');
echo $content;
echo $OUTPUT->box_end();
echo $OUTPUT->footer();

```

**ДОДАТОКБ**  
**Слайди презентації**

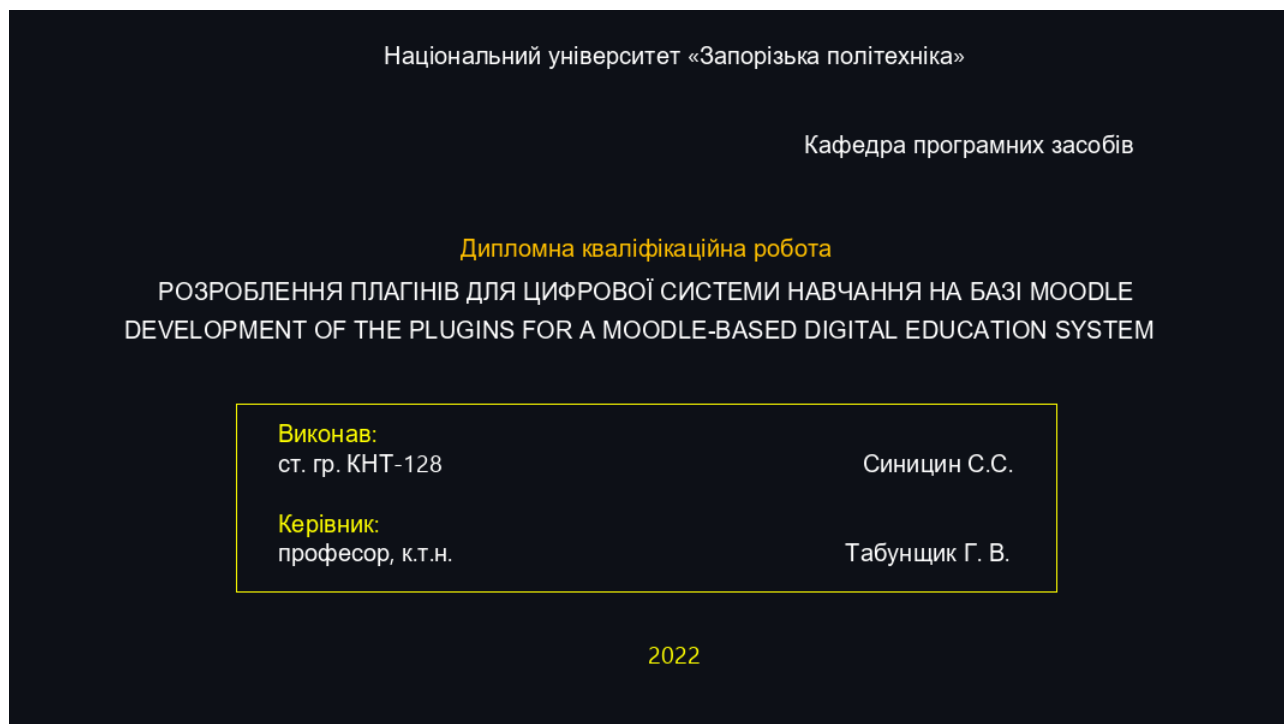


Рисунок Б.1 – Слайд №1

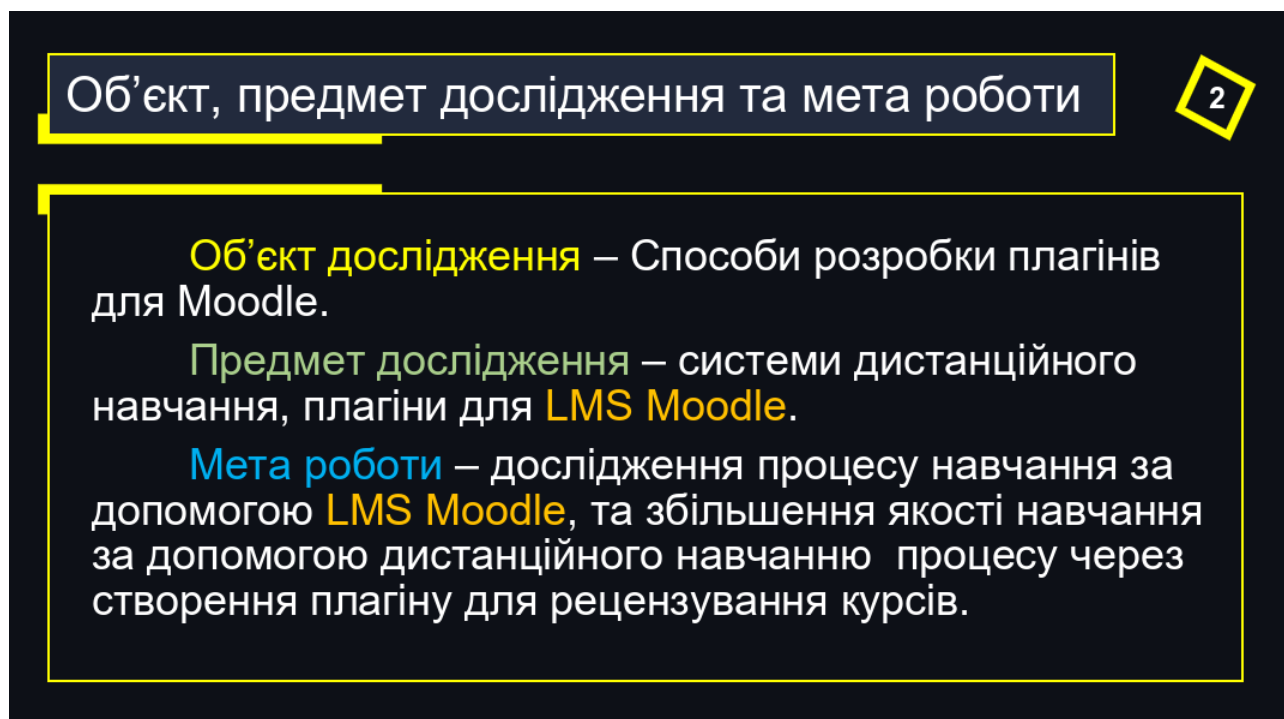


Рисунок Б.2 – Слайд №2

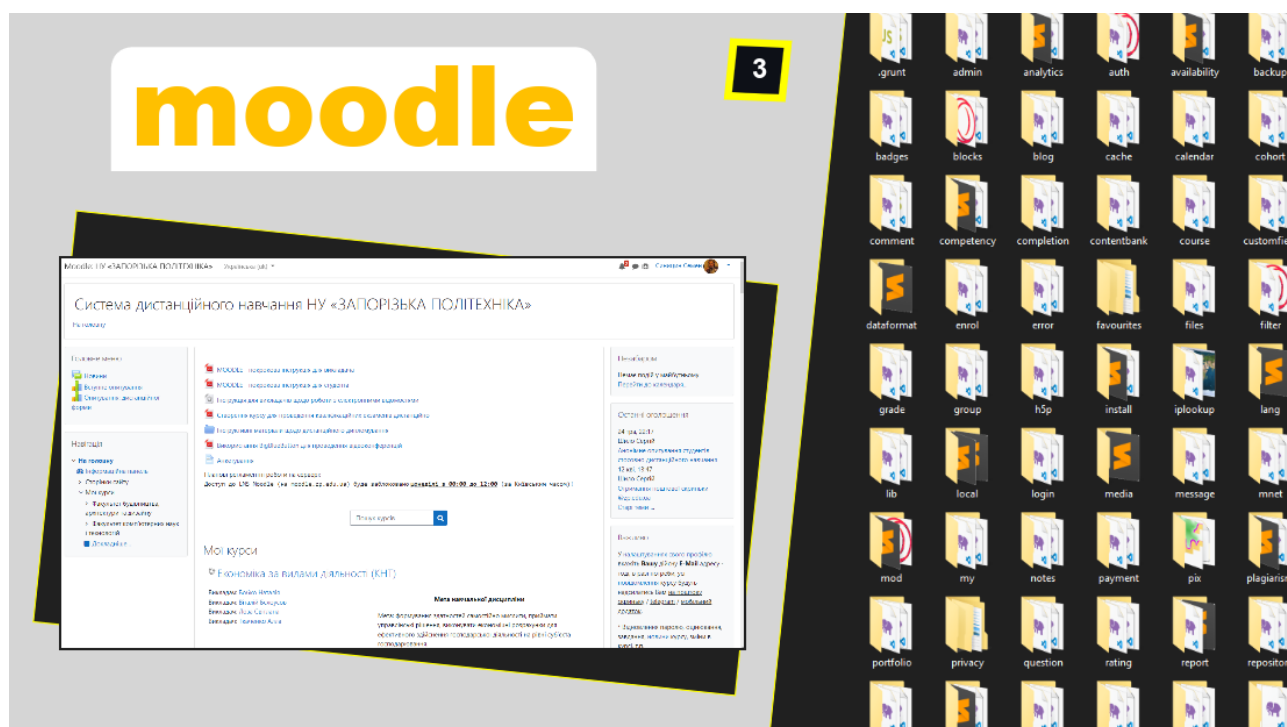


Рисунок Б.3 – Слайд №3

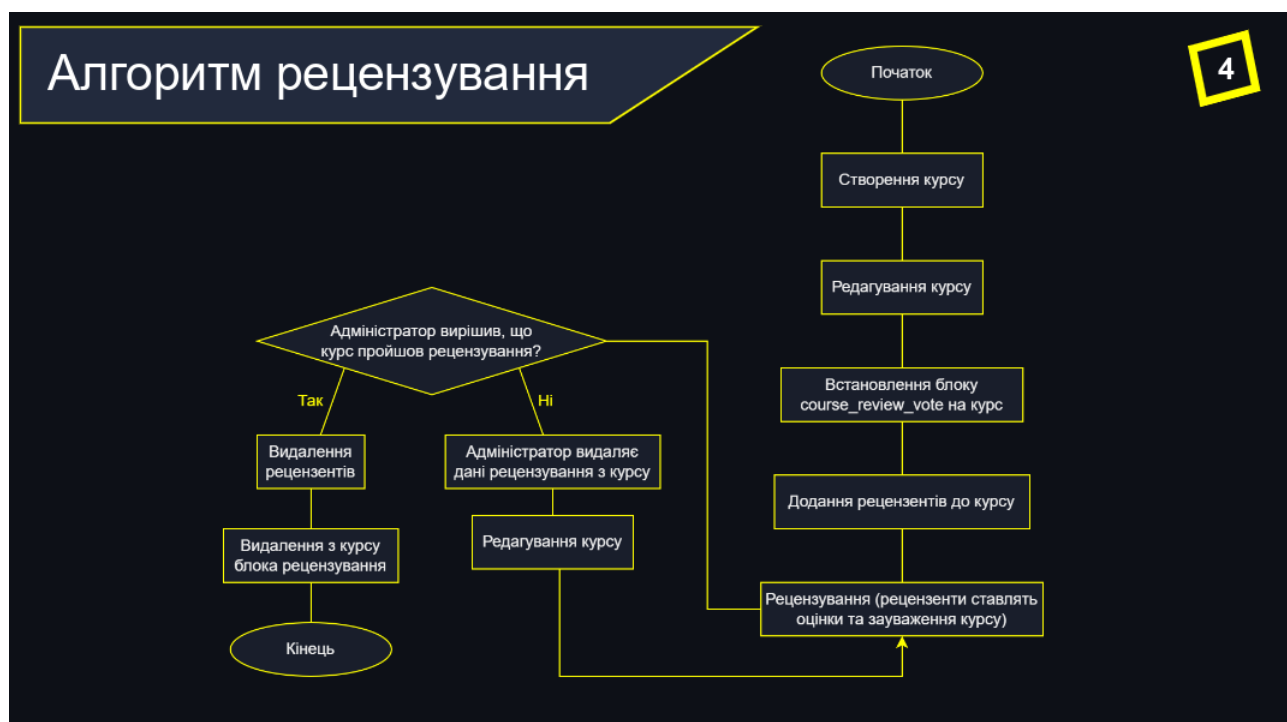


Рисунок Б.4 – Слайд №4

## Використання плагіна-блока

5

Course Review - Voting

**Vote for this course**

You can write a review about the course (optional):

Click on a rate to vote for this course. Choose from:

-2
-1
0
1
2

Voted 2 out of 3 users, this course has an average rate:

1.5

Course Review - Voting

You have already voted for this course. Your vote was:

2

You noted: "Really not bad. I would like a little more tasks, so that the students would suffer, but it's not bad anyway"

[\(Want to remove vote?\)](#)

Voted 1 out of 3 users, this course has an average rate:

2

Course Review - Voting

[Click here for an overview of the course.](#)  
[Click here to view all reports.](#)

[\(Show/hide notes\)](#)

teacher1 teacher1 - "Really not bad. I would like a little more tasks, so that the students would suffer, but it's not bad anyway"

[\(Show/hide vote summary\)](#)

teacher1 teacher1 - 2

student1 student1 - 1

[\(Show/hide those who didn't vote\)](#)

student2 student2

Voted 2 out of 3 users, this course has an average rate:

1.5

Рисунок Б.5 – Слайд №5

## Використання плагіна-звіту

6

Votes and Notes for CourseFirst (CourseFirst) (as of 2022-06-06 14:46:38)

Summary (Live votes only)

| Vote | Votes Cast | Percentage |
|------|------------|------------|
| -2   | 1          | 50.0%      |
| 1    | 1          | 50.0%      |

This course's votes at 1.50

Who voted

| Name              | Voted |
|-------------------|-------|
| teacher1 teacher1 | 1     |
| student1 student1 | 1     |
| student2 student2 | -     |

Detail

deleted our test is a deleted vote.

| Pos | Date [↑]                                                     | Name              | Vote | Note                                                                                                           | Delete! |
|-----|--------------------------------------------------------------|-------------------|------|----------------------------------------------------------------------------------------------------------------|---------|
| 1   | Added: 2022-06-06 14:28:11                                   | student1 student1 | 1    | -                                                                                                              | ✖       |
| 2   | Added: 2022-06-06 14:28:18                                   | teacher1 teacher1 | 2    | "Really not bad. I would like a little more tasks, so that the students would suffer, but it's not bad anyway" | ✖       |
| 3   | Added: 2022-06-06 14:28:50<br>(deleted: 2022-06-06 14:29:12) | teacher1 teacher1 | 0    | -                                                                                                              | ✖       |

[Report Index.](#)

Simple List of all courses with votes (as of 2022-06-06 14:45:06)

This image: [↑] indicates the column which is sorted.

| Pos | Course [↑]   | Average course rate | Users | Votes (Deleted) | Notes (Deleted) | Percentage of voters |
|-----|--------------|---------------------|-------|-----------------|-----------------|----------------------|
| 1   | CourseFirst  | 1.50                | 3     | 2 (1 deleted)   | 1               | 67%                  |
| 2   | CourseSecond | 1.00                | 2     | 1               | 0               | 50%                  |
| 3   | CourseThird  | 0.00                | 3     | 0               | 0               | 0%                   |

[Report Index.](#)

Course Review - Reports

Reports of courses

- Display all courses that have a review block with detailed statistics
- Display all courses that have a review block

Administrative Options

Notes Options

- Wipe all notes from all courses, keeping votes.

Votes Options

- Remove all votes from all courses, keeping the history of votes and notes.
- Delete all votes (and notes) *history* (which were previously set and removed).

Course Clearing Options

Remove all votes from:

No courses with votes. [Clear this course](#)

Statistics

- Votes: Live: 0; Deleted: 0; Total: 0;
- Notes: Live: 0; Deleted: 0; Total: 0;
- 0 distinct users have voted on 0 distinct courses

Рисунок Б.6 – Слайд №6

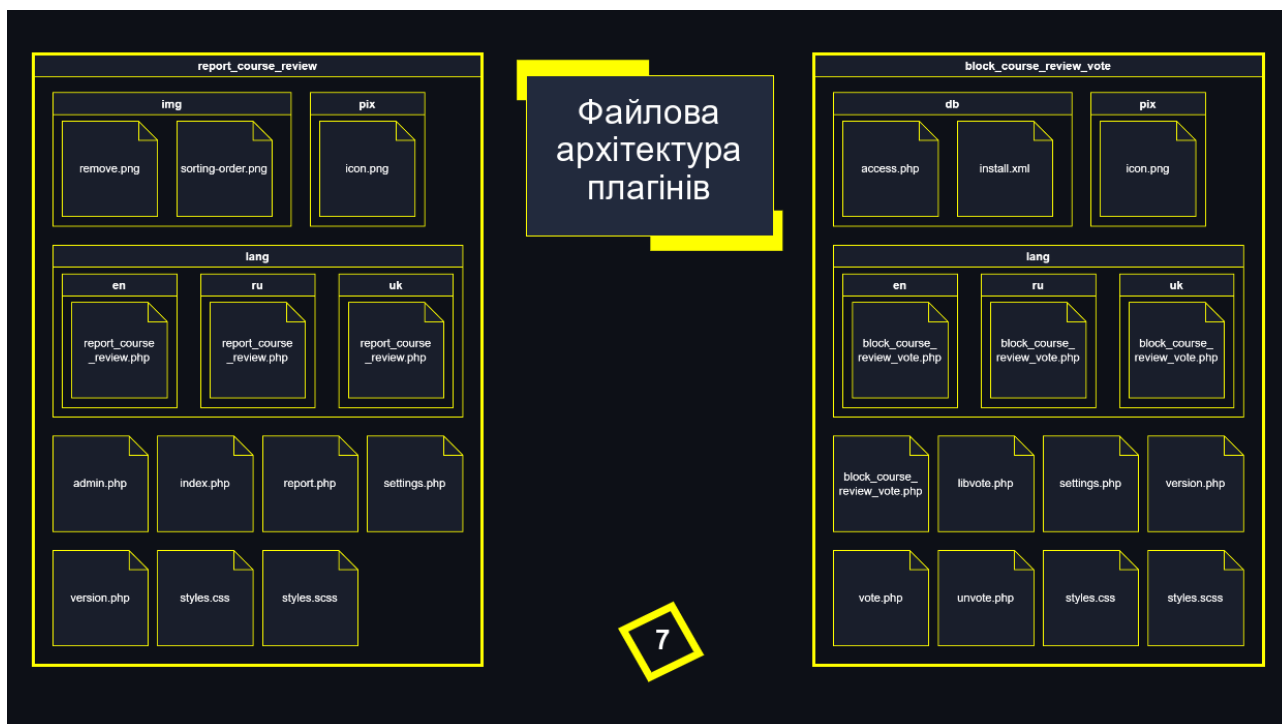


Рисунок Б.7 – Слайд №7

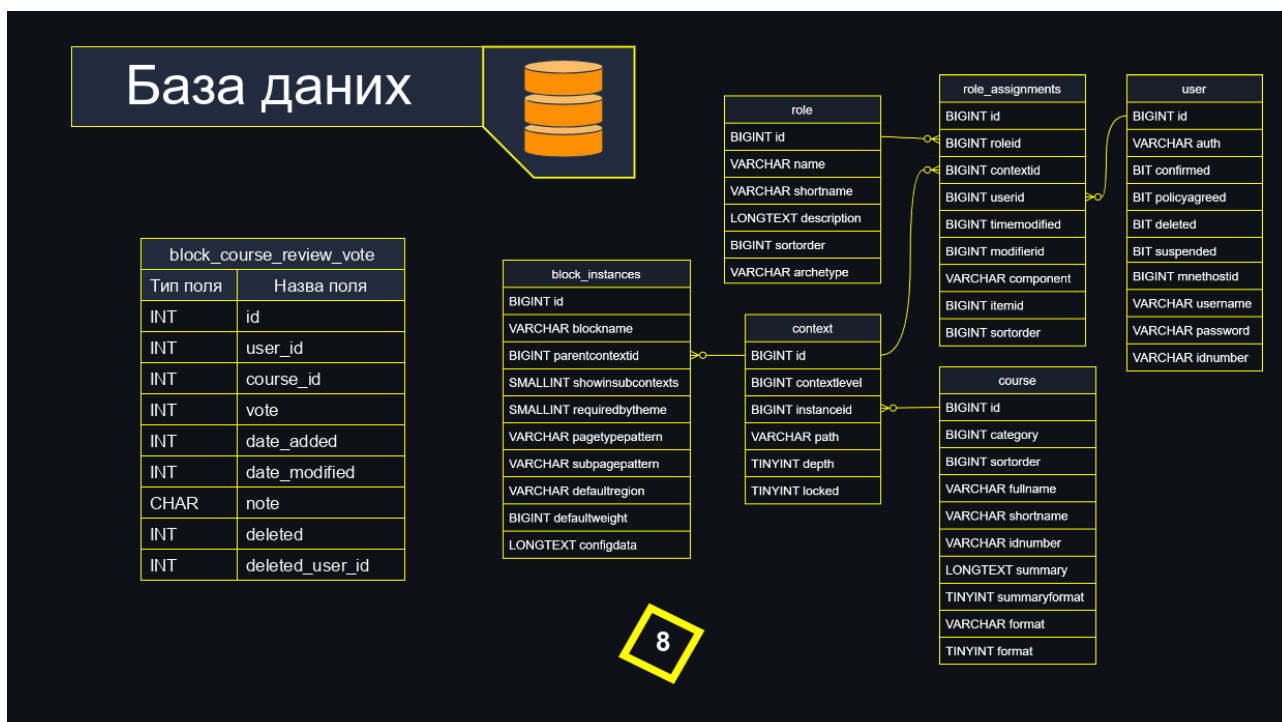


Рисунок Б.8 – Слайд №8



Рисунок Б.9 – Слайд №9



Рисунок Б.10 – Слайд №10

## Обробка помилок для плагіну-блоку

11

| Тип помилки                 | Текст помилки                                                     | Опис помилки                                                                              |
|-----------------------------|-------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| <b>error-voteoutofrange</b> | Vote cast is out of range. Stop trying to hack the review system. | Оцінка виходить за межі допустимого діапазону. Ви намагаєтесь зламати систему оцінювання. |
| <b>error-courseidnotset</b> | Course ID not set, for some reason.                               | Ідентифікатор курсу не встановлено з якоїсь причини.                                      |
| <b>error-useridnotset</b>   | User ID not set, for some reason.                                 | Ідентифікатор користувача не встановлено з якоїсь причини.                                |
| <b>error-dbinsert</b>       | Vote not recorded into the database for some reason.              | Рецензування не занесено до бази даних з якоїсь причини.                                  |
| <b>error-dbupdate</b>       | Vote not updated in the database for some reason.                 | Рецензування не оновлюється в базі даних з якоїсь причини.                                |
| <b>error-novoteid</b>       | Could not get ID for the vote for some reason.                    | Не вдалося отримати ID для рецензування з якоїсь причини.                                 |

Рисунок Б.11 – Слайд №11

## Обробка помилок для плагіну-звіту

12

| Тип помилки             | Текст помилки                                                                                                                                 | Опис помилки                                                                   |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| <b>error_admin</b>      | Sorry, you need to be a site Admin to see the Course Reports.                                                                                 | Ви повинні бути адміністратором сайту, щоб переглядати звіти про курси.        |
| <b>error_query</b>      | Sorry, the query could not be composed properly and cannot be run.                                                                            | Запит до бази даних не може бути правильно складений і не може бути запущений. |
| <b>error_noquery</b>    | Sorry, the URL is malformed and an appropriate report could not be generated.                                                                 | URL-адреса має неправильний формат, і відповідний звіт не може бути створений. |
| <b>error_nonotes</b>    | There are no live notes to remove.                                                                                                            | Немає нотаток для видалення.                                                   |
| <b>error_novotes</b>    | There are no live votes to remove.                                                                                                            | Немає оцінок для видалення.                                                    |
| <b>error_noanyvotes</b> | There are no votes to remove.                                                                                                                 | Немає рецензій на видалення.                                                   |
| <b>error_noblock</b>    | Sorry, couldn't get the block ID for the 'course_review_vote' block. Have you installed it? If so, please report this error to the developer. | не вдалося отримати ідентифікатор блоку для блоку 'block_course_review_vote'.  |
| <b>error_dbupdate</b>   | Database not updated for some reason.                                                                                                         | База даних не оновлюється з якоїсь причини.                                    |

Рисунок Б.12 – Слайд №12

## Висновки

13

Було розглянуто способи створення плагінів у **LMS Moodle**.

Написано систему рецензування курсів у **LMS Moodle** мовою **PHP**, що складаються з 2 плагінів. Також всередині обгортки php були написані запити до бази даних мовою **SQL**, **HTML** сторінки з функціоналом, та невеликі скрипти на **JavaScript**.

Рисунок Б.13 – Слайд №13

Дякую за увагу!

14

### Список слайдів:

1. Титульний лист;
2. Об'єкт, предмет дослідження та мета роботи;
3. Moodle;
4. Алгоритм рецензування;
5. Використання плагіна-блоку;
6. Використання плагіна-звіту;
7. Файлова архітектура плагінів;
8. База даних;
9. Прецеденти системи для плагіну-блоку;
10. Прецеденти системи для плагіну-звіту;
11. Обробка помилок для плагіну-блоку;
12. Обробка помилок для плагіну-звіту;
13. Висновки.

Рисунок Б.14 – Слайд №14