

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
« ____ » _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**ІНТЕЛЕКТУАЛЬНА СИСТЕМА АНАЛІЗУ ДАНИХ LMS
MOODLE ДЛЯ МОНИТОРИНГУ НАВЧАЛЬНОГО
ПРОЦЕСУ В УНІВЕРСИТЕТІ**

Спеціальність 122 «Комп'ютерні науки»

122 – КРБ – 401з.22301006

Виконав студент 4-го курсу, групи 401з
_____ *О. Ю. Тараненко*
« 20 » червня 2024 р.

Керівник: канд. пед. наук, доцент
_____ *Н. М. Болюбаши*
« 20 » червня 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р. техн. наук, проф.
_____ Ю. П. Кондратенко
«____» _____ 2024 р.

З А В Д А Н Н Я

на виконання кваліфікаційної роботи

Видано студенту групи 401з ННПО Тараненку Олексію Юрійовичу.

1. Тема кваліфікаційної роботи «Інтелектуальна система аналізу даних LMS Moodle для моніторингу навчального процесу в університеті».

Керівник роботи Болюбаш Надія Миколаївна, канд. пед. наук, доцент.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «28» грудня 2024 р. № 271

2. Строк представлення кваліфікаційної роботи студентом «20» червня 2024 р.

3. Вхідні (початкові) дані до роботи: предметна сфера моніторингу навчання у ЗВО на базі LMS Moodle, методи та алгоритми Data Mining для аналізу навчального процесу, набір даних про мережеву діяльність користувачів Moodle і їхню взаємодію з системою.

Очікуваний результат: плагін LMS Moodle для моніторингу навчального процесу в університеті з вбудованими алгоритмами інтелектуального аналізу даних.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

– здійснення аналізу та дослідження теоретичних засад моніторингу навчального процесу в університеті на базі LMS Moodle;

– обґрунтування вибору технологій і інструментальних засобів розробки плагіну для аналізу даних Moodle;

– розробка, здійснення програмної реалізації та впровадження системи аналізу даних LMS Moodle для моніторингу навчального процесу в університеті.

5. Перелік графічного матеріалу: презентація, рисунки, таблиці.

6. Завдання до спеціальної частини: Охорона праці ІТ-фахівців при роботі з екранними пристроями.

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Григор'єва Л. І., професор кафедри екології	

Керівник роботи канд. пед. наук, доцент Болубаш Н. М.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Завдання прийнято до виконання Тараненко О. Ю.
(прізвище та ініціали)

_____ (підпис)

Дата видачі завдання « 14 » січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН виконання кваліфікаційної роботи

Тема: Інтелектуальна система аналізу даних LMS Moodle для моніторингу навчального процесу в університеті

№	Найменування роботи	Початок	Закінчення	Примітки
1	Визначення керівника і теми КРБ. Подання заяви на затвердження теми КРБ	10.11.2023	15.11.2023	Виконано
2	Отримання завдання на виконання КРБ	10.01.2024	15.01.2024	Виконано
3	Складання календарного плану	16.01.2024	29.01.2024	Виконано
4	Огляд літератури за темою дослідження. Аналіз предметної сфери навчання у ЗВО на базі LMS Moodle, дослідження підходів до моніторингу навчальної діяльності, методів та алгоритмів Data Mining	30.01.2024	17.02.2024	Виконано
5	Вибір технологій та інструментальних засобів розробки системи	18.02.2024	29.02.2024	Виконано
6	Створення дизайну, проєктування та програмна реалізація, тестування	1.03.2024	15.04.2024	Виконано
7	Робота над розділами фахової частини БКР	16.04.2024	31.04.2024	Виконано
8	Проходження переддипломної практики, збір та аналіз матеріалів, остаточне оформлення розділів фахової частини БКР	29.04.2024	12.05.2024	Виконано
9	Розробка спеціальної частини з охорони праці	13.05.2024	25.05.2024	Виконано
10	Обговорення отриманих результатів з керівником та попередній захист КРБ	27.05.2024	29.05.2024	Виконано
11	Корегування роботи за результатами попереднього захисту	30.05.2024	6.06.2024	Виконано
12	Другий попередній захист КРБ	10.06.2024	10.06.2024	Виконано
12	Остаточне оформлення пояснювальної записки та слайдів доповіді до захисту	11.06.2024	12.06.2024	Виконано
13	Подання рецензенту та рецензування КРБ	13.06.2024	13.06.2024	Виконано
14	Подання КРБ, її електронної копії та інших документів (відгуку, рецензії) до захисту	17.06.2024	21.06.2024	Виконано
15	Захист КРБ перед ЕК	28.06.2024	28.06.2024	Виконано

Розробив студент Тараненко О. Ю.
(прізвище та ініціали)

_____ (підпис)

Керівник канд. пед. наук, доцент Болюбаш Н. М.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

« 29 » 01 2024 р.

АНОТАЦІЯ

кваліфікаційної роботи студента групи 401з ЧНУ ім. Петра Могили

Тараненка Олексія Юрійовича

Тема: «Інтелектуальна система аналізу даних LMS Moodle для моніторингу навчального процесу в університеті»

Кваліфікаційна робота бакалавра спрямована на розробку та здійснення програмної реалізації системи – плагіну LMS Moodle, для моніторингу навчального процесу в університеті з вбудованими алгоритмами інтелектуального аналізу даних. Що є актуальним в умовах цифровізації сфери вищої освіти, яка супроводжується накопичення великих обсягів інформації стосовно освітніх процесів. Оскільки застосування методів та алгоритмів Data Mining дозволяє оперативно отримувати інформацію про хід навчального процесу та видобувати знання, які є критично важливими для прийняття рішень у освітній сфері.

Об'єкт роботи – процес аналізу та моніторингу підготовки фахівців у закладах вищої освіти на базі платформ електронного навчання.

Предмет роботи – програмні засоби, методи та алгоритми інтелектуального аналізу даних LMS Moodle для моніторингу навчального процесу.

Мета роботи – підвищення ефективності моніторингу навчального процесу в університеті шляхом розробки плагіну LMS-системи Moodle з вбудованими алгоритмами інтелектуального аналізу даних.

Структура кваліфікаційної роботи включає фахову та спеціальну частину з охорони праці. Фахова частина включає вступ, три розділи, висновки та додатки. У першому розділі розкрито теоретичні засади моніторингу навчального процесу в університеті на базі LMS Moodle. У другому розділі обґрунтовано вибір засобів розробки плагіну для аналізу даних Moodle. У третьому розділі описано моделювання, проектування, програмну реалізацію та тестування інтелектуальної системи для моніторингу навчального процесу в університеті.

Бакалаврська кваліфікаційна робота містить 82 сторінки (без додатків), 37 рисунків, 3 таблиці, 40 джерел та 3 додатки.

Ключові слова: електронне навчання, система LMS Moodle, аналіз освітніх даних, алгоритми кластерного аналізу, моніторинг навчального процесу.

ABSTRACT

**for bachelor's qualification work
of a student of 401z group at Petro Mohyla Black Sea National University**

Taranenko Oleksii

Theme: «LMS Moodle intelligent data analysis system for monitoring the educational process at the university»

The bachelor's qualification work is aimed at the development and implementation of the software implementation of the system - the LMS Moodle plugin, for monitoring the educational process at the university with built-in algorithms of intelligent data analysis. What is relevant in the conditions of digitization of the sphere of higher education, which is accompanied by the accumulation of large volumes of information about educational processes. Since the application of Data Mining methods and algorithms allows you to quickly obtain information about the course of the educational process and extract knowledge that is critically important for decision-making in the educational field.

Object of work – the process of analyzing and monitoring the training of specialists in higher education institutions based on e-learning platforms.

Subject of work – software tools, methods and algorithms for intellectual analysis of LMS Moodle data for monitoring the educational process.

The purpose of this work is to increasing the effectiveness of monitoring the educational process at the university by developing a plugin for the LMS Moodle system with built-in algorithms for Data Mining.

The structure of the bachelor's work includes a professional and special part on labor protection. The professional part includes an introduction, three chapters, conclusions and appendices. The first chapter reveals the theoretical principles of monitoring the educational process at the university based on LMS Moodle. The second section substantiates the choice of tools for the development of a Moodle data analysis plugin. The third chapter describe the modeling, design, software implementation and testing of an intelligent system for monitoring the educational process at the university.

The bachelor qualification work contains 82 pages (without appendices), 37 figures, 3 tables, 40 sources and 3 appendices.

Keywords: e-learning, LMS Moodle system, educational data analysis, cluster analysis algorithms, monitoring of the educational process.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП.....	5
1 ТЕОРЕТИЧНІ ЗАСАДИ МОНІТОРИНГУ НАВЧАЛЬНОГО ПРОЦЕСУ В СИСТЕМІ LMS MOODLE.....	7
1.1 Система управління навчанням Moodle	7
1.2 Засоби контролю та моніторингу Moodle.....	12
1.3 Методи та алгоритми інтелектуального аналізу освітніх даних.....	21
1.4 Постановка задачі.....	27
Висновки до розділу 1	29
2 ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТАЛЬНІ ЗАСОБИ РОЗРОБКИ ПЛАГІНУ ДЛЯ АНАЛІЗУ ДАНИХ MOODLE.....	32
2.1 Середовище розробки PhpStorm.....	32
2.2 Гіпертекстовий препроцесор PHP	33
2.3 Мова програмування JavaScript.....	35
2.4 HTML та каскадні таблиці стилів CSS	38
2.5 Мова структурованих запитів SQL	41
2.6 Бібліотеки для розробки плагіна Moodle.....	42
Висновки до розділу 2	45
3 ПРОЄКТУВАННЯ, ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ВПРОВАДЖЕННЯ СИСТЕМИ МОНІТОРИНГУ НАВЧАЛЬНОГО ПРОЦЕСУ	46
3.1 Архітектура системи у відкритих кодах Moodle	46
3.2 Проєктування розробки та структури плагіну	54
3.3 Програмна реалізація.....	57
3.4 Інтерфейс системи аналізу та моніторингу навчального процесу в університеті.....	66
Висновки до розділу 3	76
ВИСНОВКИ.....	77

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	80
ДОДАТОК А Функція реалізації алгоритмів кластеризації	83
ДОДАТОК Б Функція візуалізації результатів аналізу.....	86
ДОДАТОК В Область налаштувань кластерного аналізу для адміністратора.....	93
ДОДАТОК Г Акт про впровадження системи	94

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних

ЗВО – заклад вищої освіти

API – Application Programming Interface

ECTS – European Credit Transfer and Accumulation System

HTML – Hyper Text Markup Language

JS – Java Script

LMS – Learning Management System

MOODLE – Modular Object-Oriented Dynamic Learning Environment

PHP – Hypertext PreProcessor

RDMS – Relational Database Management System

ВСТУП

Актуальність. Цифрова трансформація системи освіти в цілому та професійної освіти зокрема супроводжується кардинальними змінами в освітніх технологіях, впровадженням інноваційних форм, методів та засобів навчання на базі сучасних освітніх технологічних платформ, Web-технологій, хмарних сервісів. Це обумовлює необхідність впровадження у практику вищих навчальних закладів моніторингу, який здатен регулярно відслідковувати якість професійної підготовки на базі можливостей потенціалу сучасних освітніх цифрових ресурсів та платформ. Перспективним напрямком підвищення ефективності контролю та моніторингу навчального процесу в університеті є розробка плагіну Moodle, здатного здійснювати аналіз, спрямований на отримання оперативної динамічної інформації про стан навчального процесу у середовищі електронного навчання за допомогою методів та алгоритмів Data Mining.

Найбільшу популярність для організації навчального процесу у закладах вищої освіти України отримала LMS-система Moodle, яка має засоби для моніторингу навчального процесу на рівні окремого навчального курсу та системи в цілому. Проте за допомогою стандартних інструментальних засобів Moodle неможливо отримувати інформацію про зведену успішність та якість електронного навчання на рівні окремого викладача чи навчального підрозділу закладу вищої освіти, що потребує подальшого опрацювання. Недостатньо дослідженими є способи реалізації у середовищі Moodle алгоритмів інтелектуального аналізу даних із метою виявлення закономірностей і їх використання для прийняття рішень у освітній сфері.

Це обумовило **мету роботи**, яка полягає у підвищенні ефективності моніторингу навчального процесу в університеті шляхом розробки плагіну LMS-системи Moodle з вбудованими алгоритмами інтелектуального аналізу даних.

Відповідно до поставленої мети було сформульовано **завдання**:

- здійснити аналіз та дослідити теоретичні засади моніторингу навчального процесу у закладах вищої освіти на базі LMS Moodle;
- обґрунтувати вибір технологій та інструментальних засобів розробки плагіну для аналізу даних Moodle;
- розробити, здійснити програмну реалізацію та впровадження системи аналізу даних LMS Moodle для моніторингу навчального процесу в університеті.

Об'єкт роботи – процес аналізу та моніторингу підготовки фахівців у закладах вищої освіти на базі платформ електронного навчання.

Предмет роботи – програмні засоби, методи та алгоритми інтелектуального аналізу даних LMS Moodle для моніторингу навчального процесу.

Методологічною основою дослідження є загальнонаукові, статистично-аналітичні методи та методи інтелектуального аналізу даних, які дозволили комплексно вивчити предмет та об'єкт дослідження, дослідити розвиток науково-методичних засад, напрямів та шляхів підвищення ефективності моніторингу навчального процесу у вищих навчальних закладах.

Практичне значення отриманих результатів полягає в тому, що розроблений плагін було впроваджено у LMS-систему Moodle університету. Застосування розробленої системи дозволило суттєво поліпшити моніторинг на рівні різних структурних підрозділів університету шляхом підвищення оперативності отримання інформації про хід навчального процесу на базі електронної платформи Moodle.

Структура кваліфікаційної роботи. Відповідно до мети, завдань і предмета дослідження, бакалаврська робота містить основну та спеціальну частини. Основна частина роботи складається із вступу, трьох розділів, висновку, списку використаних джерел та 3 додатків. Загальний обсяг роботи – 119 сторінок, із них основного тексту основної частини – 94 сторінки, спеціальної – 25 сторінок. Кількість використаних джерел – 40.

1 ТЕОРЕТИЧНІ ЗАСАДИ МОНІТОРИНГУ НАВЧАЛЬНОГО ПРОЦЕСУ В СИСТЕМІ LMS MOODLE

1.1 Система управління навчанням Moodle

Цифрова трансформація освітніх установ супроводжується широким впровадженням дистанційних і змішаних форм навчання. Широке розповсюдження електронного навчання та його використання у навчальних закладах сумісно з традиційними формами навчання здійснюється на базі чисельних електронних платформ, серед яких можна виділити систему Moodle (англ. Modular Object-Oriented Dynamic Learning Environment) [1].

Moodle є безкоштовною, відкритою (англ. Open Source) системою управління навчанням (англ. Learning Management System, LMS). Вона реалізує філософію «педагогіки соціального конструктивізму» та орієнтована насамперед на організацію взаємодії між викладачем та студентами. Цей підхід передбачає зміну функції викладача: замість джерела знань він перетворюється в "центр впливу". Викладач повинен знайти індивідуальний контакт із кожним студентом, адаптуючись під його освітні потреби, направляти дискусії й спільну діяльність таким чином, щоб колективно досягти цілей навчання. Платформа підходить для реалізації дистанційних, змішаних форм навчання, що перетворює її на додаток до традиційних очних форм навчання.

Moodle отримала найбільшу популярність у закладах вищої освіти для розміщення навчального контенту та організації навчального процесу. На базі навчальних закладів здійснюється формування цифрових навчальних середовищ з опорою на LMS-платформу Moodle та хмарні технології. Цифрова технологія навчання базується на представленні навчального контенту та результатів навчальної діяльності у цифровій формі зі зміною парадигми спілкування шляхом надання можливості для розподіленої у просторі або у просторі та часі взаємодії викладачів зі студентами.

Moodle має широкий набір функціональності, притаманний електронним платформам LMS-систем навчання. Її функціонал дозволяє реалізовувати багато видів навчальної діяльності під час вивчення дисциплін у закладах вищої освіти, що надає можливість максимально наблизити проведення дистанційних занять до традиційних форм шляхом надання суб'єктам навчального процесу можливості онлайн-спілкування та розподіленого у просторі й часі доступу до навчального контенту і результатів навчальної діяльності студентів [2].

Функції адміністрування в LMS охоплюють кілька базових областей. Управління студентами містить у собі завдання реєстрації користувачів і контролю доступу до системи й навчального контенту, організації слухачів у певні групи, для формування курсів, складання звітності, керування викладацькими й аудиторними ресурсами.

Система Moodle має інструментальні засоби для додавання до курсу навчального матеріалу у будь-якому форматі, організації дистанційних занять, контролю та оцінювання результатів навчання у рамках окремого навчального курсу. Основою навчального простору Moodle є дистанційні курси, призначені для вивчення навчальних дисциплін, які включають набір викладачів, студентів і навчальних матеріалів та забезпечують інтерактивну взаємодію суб'єктів навчального процесу з різними правами та ролями, обумовленими їх місцем у навчальному процесі, постійний контакт між ними на протязі всього періоду навчання, контроль отримуваних знань і накопичення інформації про процес навчання. До основних функціональних структурних елементів електронного курсу відносяться:

- початкова сторінка курсу: перша сторінка – містить *область даних*, яку викладач наповнює навчальним контентом (рис. 1.1);
- секція: фрагмент навчального курсу, розміщений у області даних, який має свою назву і включає один або декілька елементів курсу;

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система аналізу даних LMS–системи Moodle для моніторингу навчального процесу в університеті

- модуль діяльності (англ. Activity Modules): елемент курсу, який містить навчальний матеріал, є базою для створення навчальних засобів, організації електронного навчання й відображається на сторінці курсу у вигляді посилань;
- файлова система призначена для збереження електронних файлів різних форматів з навчальним контентом;
- банк тестових завдань: структурована по категоріям тестова база, використовується на різних етапах контролю й моніторингу при вивченні навчальної дисципліни;
- журнал оцінок: відображає отримані студентами бали за виконання навчальних завдань;
- блок: прямокутна область у правій або лівій частині сторінки, призначена для виконання певної роботи з курсом – перегляду календаря подій, новин та анонсів подій;
- комунікаційні засоби: електронна пошта, служба повідомлень Moodle, форуми, блоги тощо.

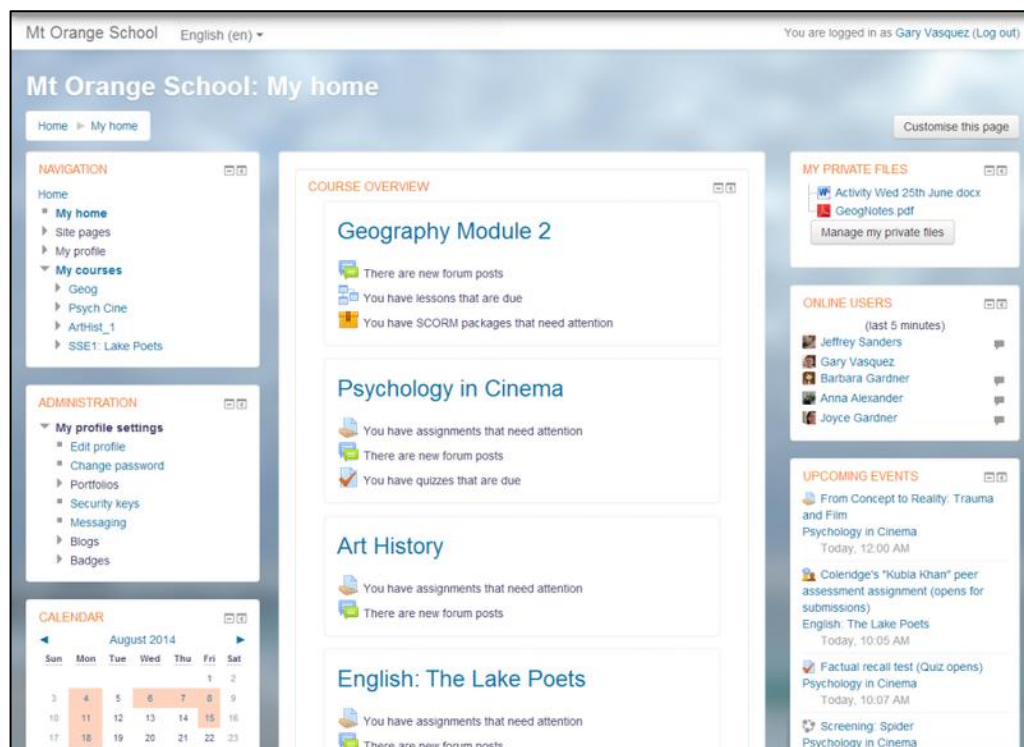


Рисунок 1.1 – Початкова сторінка LMS Moodle

На рисунку 1.2 показано основні зв'язки між функціональними структурними елементами електронного курсу в системі Moodle.



Рисунок 1.2 – Функціональна структура курсу Moodle

Наявність модулів діяльності Activity Modules Moodle дозволяє організувати навчальну роботу шляхом формування навчально-методичних матеріалів. Всі модулі умовно поділено на *Ресурси* та *Завдання*.

Ресурси дозволяють додавати до курсу навчальний матеріал, представлений практично у будь-якому цифровому форматі (текстовому, графічному, звуковому, відео, тощо) для оглядового ознайомлення з ним: гіпертекстові сторінки та посилання на них; посилання на файли, які можуть бути розміщені у файловій системі курсу Moodle, на диску Google Drive або у іншому хмарному сервісі.

Завдання є інтерактивними елементами курсу, які призначені для розміщення навчальної інформації та організації навчального процесу. Елементи, які найчастіше використовують, описано нижче.

1. Глосарій (англ. Glossary) – список означень основних понять дисципліни, обов'язкових для засвоєння.

2. Форум (англ. Forum) – застосовують для колективного спілкування та обговорення різних питань.

3. Завдання (англ. Assignment) – дозволяє створювати практичне завдання, яке потребує розв’язку в електронному вигляді та передбачає збір виконаних завдань і оцінювати їх викладачем.

4. Тест (англ. Quiz) – дозволяє створювати набори тестових завдань різних типів для здійснення різних видів контролю, проходження тесту оцінюється автоматично відповідно до зроблених налаштувань.

5. Урок (англ. Lesson) – дозволяє гнучко подавати навчальний матеріал разом з сукупністю завдань для його засвоєння та контролю на веб-сторінках, пов’язаних між собою посиланнями.

У рамках окремого навчального закладу Moodle дозволяє створювати структуровану базу дистанційних курсів із навчальними матеріалами по кожній дисципліні, проводити тестування й опитування, призначати навчальні завдання, перевіряти й оцінювати їх, а також формувати зведені звіти. Викладач має широкі можливості для оцінювання студентів, у тому числі в автоматичному режимі.

Використання хмарних освітніх сервісів Google, засобів для відеозв’язку у режимі онлайн, інтегрованих із системою управління навчанням Moodle, підвищує мобільність навчання, розширює можливості для асинхронної та синхронної дистанційної взаємодії викладача та студентів, дозволяє реалізовувати різні сценарії проведення онлайн занять, оптимізує обмін і зберігання інформації з навчальним контентом і результатами навчання та полегшує й розширює доступ суб’єктів навчального процесу до неї. Застосування засобів автоматизованого оцінювання Moodle полегшує й суттєво підвищує ефективність проведення поточного та підсумкового контролю і моніторингу.

Включення до структури дистанційних курсів Moodle занять-тренінгів сприяє підвищенню залученості студентів до роботи з цифровим навчальним контентом, дозволяє реалізовувати індивідуальні траєкторії навчання та супроводжується зростанням рівня засвоєння навчального матеріалу. Цифрова форма подачі навчального матеріалу у різних форматах: текстовому, гіпертекстовому, мультимедійному з фрагментами відео та аудіо, у вигляді

слайдів з елементами анімації сприяє підвищенню наочності навчального контенту, візуалізації навчальної інформації у рамках вивчення дисциплін.

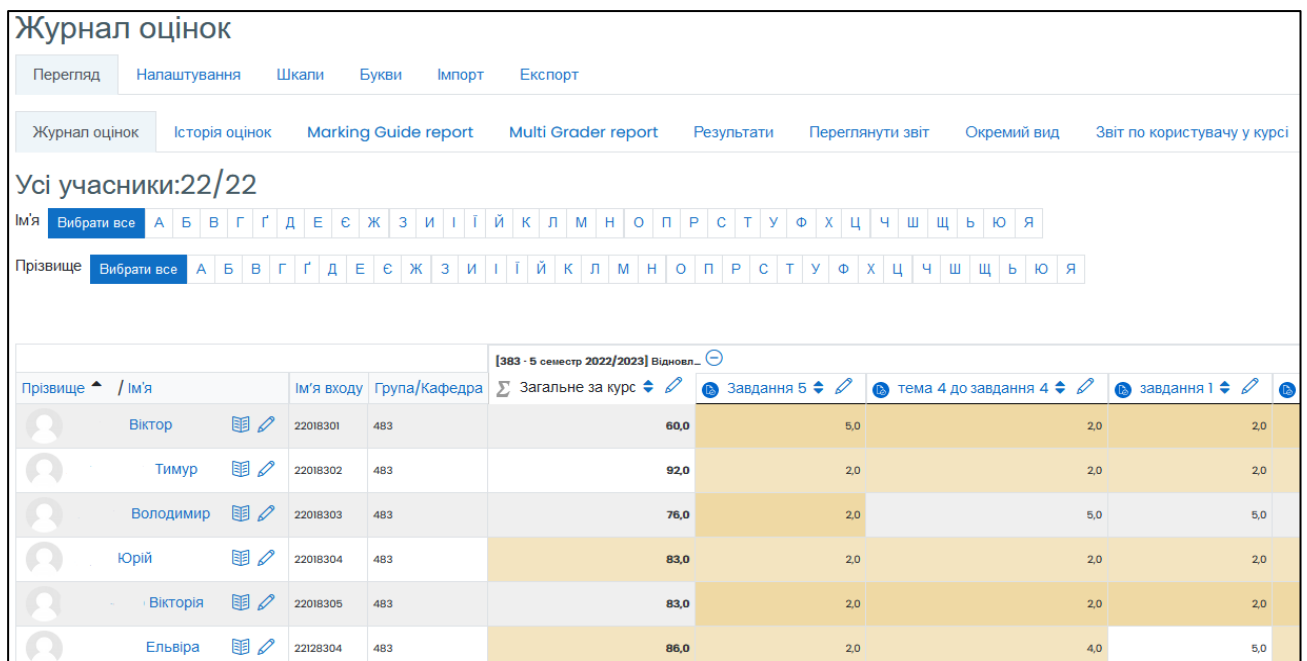
В цілому система Moodle у багатьох навчальних закладах використовується в якості фреймворка для власних програмних рішень. Вона є Web-застосунком типу систем управління навчанням, написаним на мові програмування PHP, використовує MySQL та мову програмування Perl, працює з MySQL, PostgreSQL, Microsoft SQL Server або Oracle. Moodle може працювати з об'єктами SCO, відповідає стандарту SCORM. Завдяки розвиненій модульній архітектурі, можливості Moodle можуть легко розширюватися сторонніми розроблювачами у разі потреби.

1.2 Засоби контролю та моніторингу Moodle

Моніторинг є механізмом, який полягає у зборі й аналізі інформації із широкого спектра критеріїв і показників, що дозволяє безперервно спостерігати за станом системи професійної підготовки фахівців у закладах освіти, виявляти сильні й слабкі місця та прогнозувати її розвиток. Дані, отримані під час моніторингу, є важливим інструментом для викладачів та керівників різних рівнів структурних підрозділів університету – завідуючих кафедрою, деканів факультету, директорів інститутів, ректорів та завідуючих навчальним відділом. Результати проведеного моніторингу допомагають ухвалювати обґрунтовані рішення щодо розвитку вдосконалення навчального процесу, кадрової політики та фінансового планування [3].

В цілому широкі можливості для моніторингу надає система Moodle як відкритий інтерактивний інформаційний освітній простір, що містить навчальну інформацію та інформацію з результатами навчальної діяльності, сукупність процедур і засобів оцінювання якості підготовки та моніторингу діяльності студентів у рамках окремого навчального курсу.

LMS Moodle має інструменти моніторингу навчального процесу, які дозволяють аналізувати процеси на рівні окремого електронного курсу або системи в цілому. Вона здійснює функції збору й зберігання інформації про отримані студентами оцінки – викладач може переглянути їх у Журналі оцінок (рис. 1.3).



Журнал оцінок

Перегляд Налаштування Шкали Букви Імпорт Експорт

Журнал оцінок Історія оцінок Marking Guide report Multi Grader report Результати Переглянути звіт Окремий вид Звіт по користувачу у курсі

Усі учасники: 22/22

Ім'я Вибрати все А Б В Г Г Д Е Є Ж З И І І Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ю Я

Прізвище Вибрати все А Б В Г Г Д Е Є Ж З И І І Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ь Ю Я

[383 - 5 семестр 2022/2023] Відновл...

Прізвище / Ім'я	Ім'я входу	Група/Кафедра	Σ Загальне за курс	Завдання 5	тема 4 до завдання 4	завдання 1
Віктор	22018301	483	60,0	5,0	2,0	2,0
Тимур	22018302	483	92,0	2,0	2,0	2,0
Володимир	22018303	483	76,0	2,0	5,0	5,0
Юрій	22018304	483	83,0	2,0	2,0	2,0
Вікторія	22018305	483	83,0	2,0	2,0	2,0
Ельвіра	22128304	483	86,0	2,0	4,0	5,0

Рисунок 1.3 – Журнал оцінок Moodle

Moodle дозволяє відслідковувати активність студентів та викладачів при роботі з матеріалами навчального курсу та переглядати список доступних йому курсів (рис. 1.4, рис. 1.5). Однак перегляд цих характеристик можливий у тільки у рамках окремого курсу. Отримати оперативні дані про результативність електронного навчання у сукупності груп на рівні окремого викладача, чи структурного підрозділу університету немає можливості. Недостатньо розвинений функціонал для аналізу ходу навчального процесу не дає можливості завідувачу кафедри, декану факультету чи проректору оперативно отримувати звітну інформацію та проводити моніторинг.

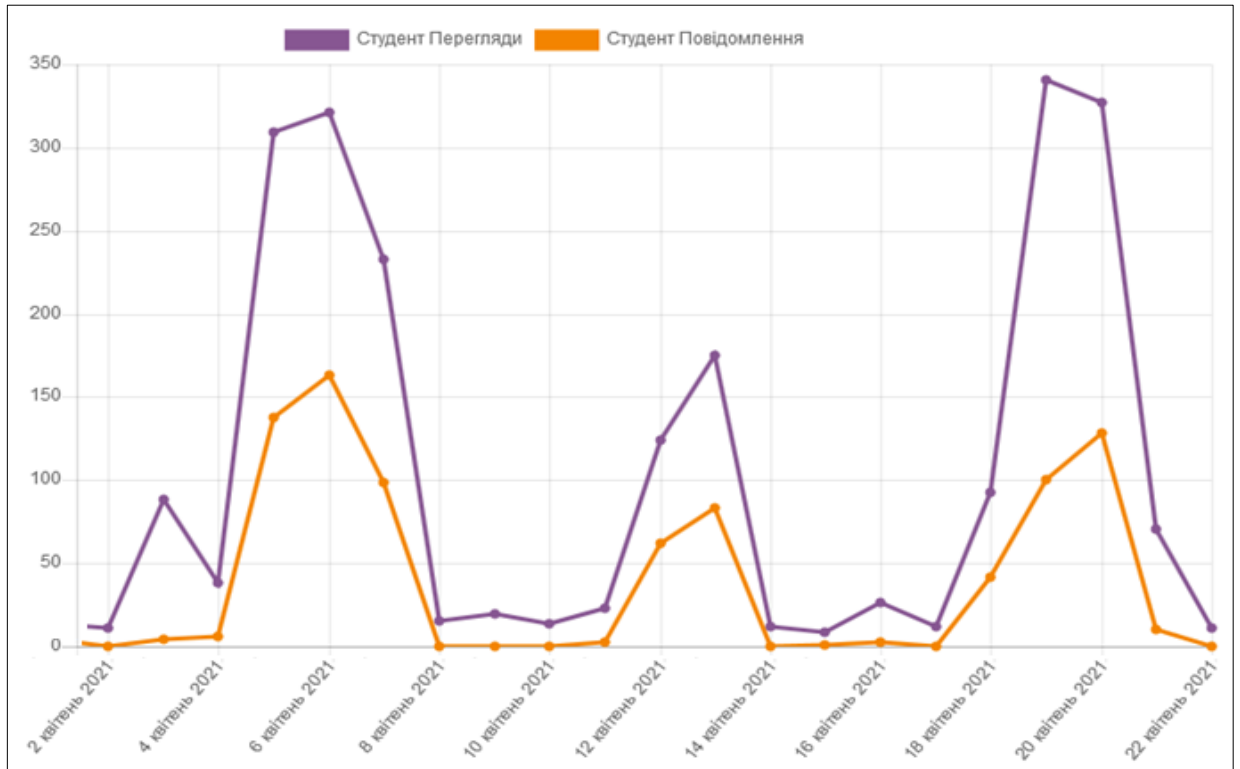


Рисунок 1.4 – Показники активності студентів до роботи з навчальним контентом курсу Moodle

Курси, де я викладаю

Назва курсу
[683м · 2023/2024 · ОЛ] Оглядові лекції
[286 · 4 семестр 2023/2024] Фітнес–технології у фізичному вихованні та спорту
[283 & 284 · 4 семестр 2023/2024] Фітнес–технології у фізичному вихованні та спорті
[583м · 10 семестр 2023/2024] Інтелектуальна власність у сфері фізичної культури і спорту
[583м · 10 семестр 2023/2024] Оздоровчі системи у фізичному вихованні та спорту
[583м · 10 семестр 2023/2024] Проектна діяльність у спорті
(286з · 2023/2024) Фітнес–технології у фізичному вихованні та спорті

Рисунок 1.5 – Список курсів викладача Moodle

Moodle дозволяє автоматично генерувати наступні звіти:

- звіти по логам користувачів (англ. Log report) у розрізі курсу й по кожному користувачеві (рис. 1.6) із вказівкою повного імені користувача, часу, курсу, чиненої ним дії, IP-адреси комп'ютера, з якого здійснювався вхід;
- звіт про діяльності (англ. Activity report) загальний по всіх елементах курсу і деталізований по кожному елементу зокрема;
- звіт про участь користувачів в елементі курсу (англ. Participation report).

Час	IP адреса	Повне ім'я	Дія	Інформація
Sun 27 January 2019, 21:15 PM	100.100.90.208	Онацький Віталій	user view	Віталій Онацький
Sun 27 January 2019, 20:07 PM	100.100.57.254	Соколюк Антон	resource view	Лабораторна робота №2 Методика проведення лекцій_Розробка КЛ
Sun 27 January 2019, 20:07 PM	100.100.57.254	Соколюк Антон	resource view	Лабораторна робота №2 Методика проведення лекцій_Розробка КЛ
Sun 27 January 2019, 20:07 PM	100.100.57.254	Соколюк Антон	course view	Методика викладання дисциплін з комп'ютерних технологій
Sat 26 January 2019, 09:16 AM	46.118.21.132	Чорновол Лена	assignment view	Зроблені завдання по Лабораторній роботі №2

Рисунок 1.6 – Звіт по логам користувачів курсу Moodle

Викладач має можливість переглядати звіти – їх аналіз дає можливість побачити, як часто той чи інший студент працював з матеріалами курсу. Однак зведену інформацію отримати не можливо у випадку, коли здійснюється аналіз роботи студентів сукупності курсів. Тому є потреба у розширенні функціональності системи.

Здійснений аналіз підходів до розширення можливостей системи Moodle дозволив виявити наступні:

- зміна ядра системи: дозволяє повністю підбудувати LMS Moodle для потреб окремого навчального закладу, але отримана нова система повинна бути відкритою відповідно до умов її розповсюдження;
- написання окремих плагінів системи: Moodle має розвинену модульну архітектуру, що дозволяє розширювати можливості системи стороннім розробникам;
- написання незалежного програмного забезпечення на тій же предметній основі: цей підхід прив'язаний до версії та серії Moodle, для якої він був реалізований;

– інтеграція стороннього програмного забезпечення засобами модулів реалізуючих інтерфейси: тут відсутня жорстка залежність від версії й серії LMS Moodle, проте використання стандартизованих протоколів (таких як SOAP) може привести до збільшення мережного трафіку й підвищення часових затримок, пов'язаних з обробкою й передачею даних;

– написання незалежного програмного забезпечення, яке здійснює аналіз логів Moodle: відсутня проблема, пов'язана з залежністю від версії й серії LMS Moodle та збільшенням мережевого трафіку при обробці даних.

Розвиток системи Moodle в основному відбувався у напрямку скорочення розміру ядра й переносу більшої кількості функцій у плагіни. Проаналізувавши усі можливі підходи до розширення функціоналу стосовно моніторингу навчального процесу у рамках окремого навчального закладу було прийнято рішення розробки плагіну Moodle із вбудованими алгоритмами інтелектуального аналізу даних.

Для формування зведеної успішності та якості електронного навчання було проаналізовано наявні модулі Moodle, які дозволяють формувати звіти з результатами навчальної діяльності та діями користувачів на курсі. Розглянемо їх можливості більш детально.

1. *Grade Distribution*. Цей модуль дозволяє вивести гістограму за заданим елементом курсу, який передбачає оцінювання, та визначеними границями балів оцінок, представленими по горизонтальній осі гістограми (рис. 1.7) [4]. Гістограму можна скачати у графічному та PDF форматі або роздрукувати. Дані, за якими побудовано гістограму, можна скачати у вигляді Excel документу, ods електронної таблиці та csv файлу. Мінуси даного модуля у тім, що на кожен курс потрібно заходити окремо і викликати цей модуль. Є тільки класифікація за балами, але немає класифікації за оцінками у національній шкалі та шкалі ECTS. Немає інших аналітик, наприклад активності студента.

2. *Multi Course Grader Report*. Модуль видає дерево активних курсів як на першій сторінці Moodle та дозволяє обрати будь-які курси в цьому дереві [5]. В

результаті отримуємо сукупність таблиць по кожному курсу окремо, якби на кожен курс зайти в журнал оцінок і скопіювати таблицю на цю сторінку. Тобто є повні таблиці оцінювання праці студентів по кожному курсу на одній сторінці (рис. 1.8). Модуль дозволяє скачати кожен таблицю у вигляді Excel документа, ods електронної таблиці, xml та txt файлу.

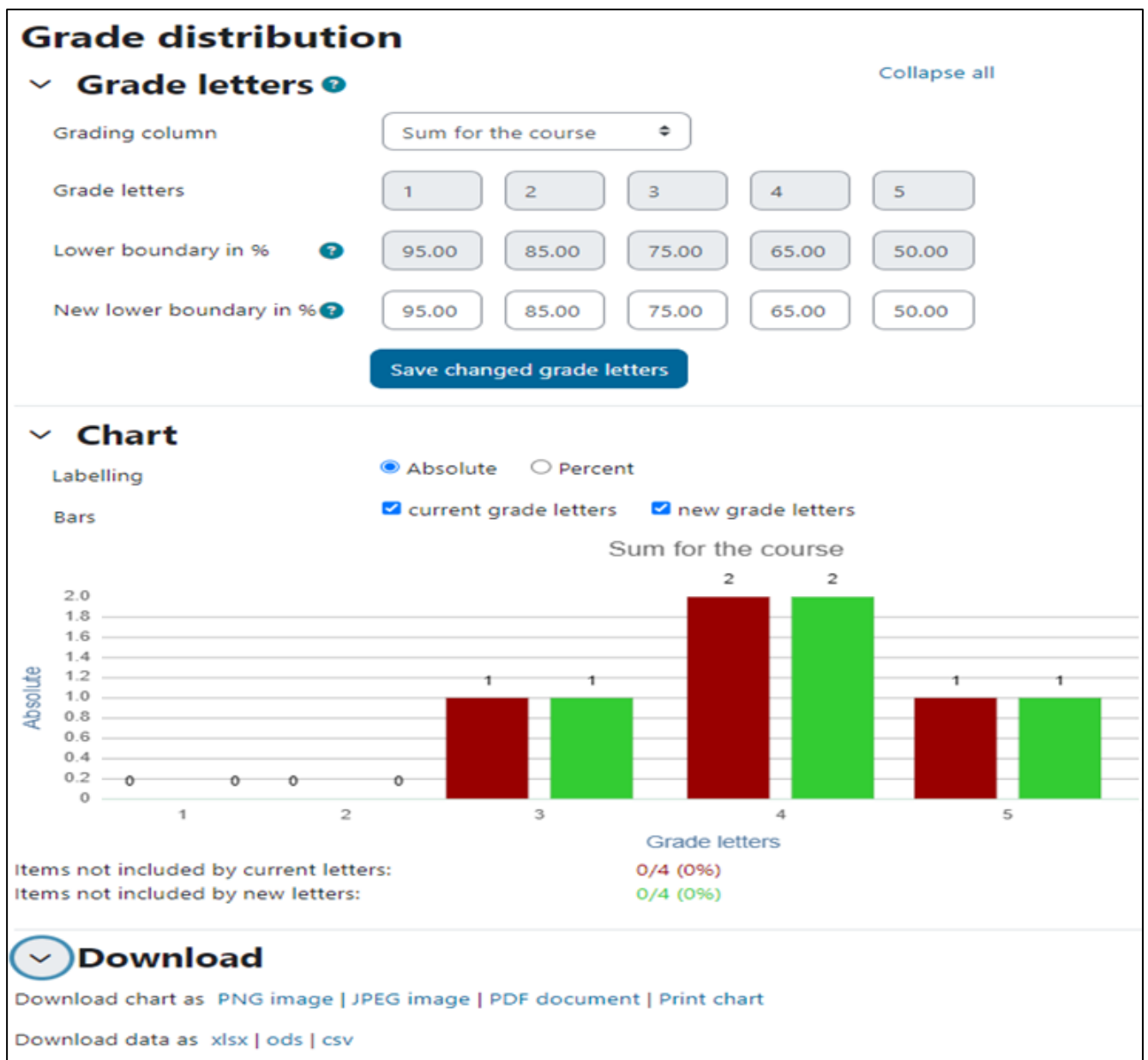


Рисунок 1.7 – Результат роботи модуля Grade Distribution

Розглянемо мінуси модуля Multi Course Grader Report. По перше, модуль

Інтелектуальна система аналізу даних LMS–системи Moodle для моніторингу навчального процесу в університеті розроблявся для версії 2.x, а сьогодні ми маємо версію 3.9.x, і вже скоро буде перехід до 4.x. Тому цей модуль вимагав змін для того, щоб з ним працювати. Немає перевірки доступу до курсів на період формування дерева курсів, тобто є можливість вибрати будь-який курс, але інформацію не буде відображено, якщо не має доступу користувача-викладача чи менеджера до нього. Немає ніяких графіків з аналізом отриманої інформації, відсутнє групування оцінок. Присутній великий обсяг зайвої інформації, хоча і на одній сторінці.

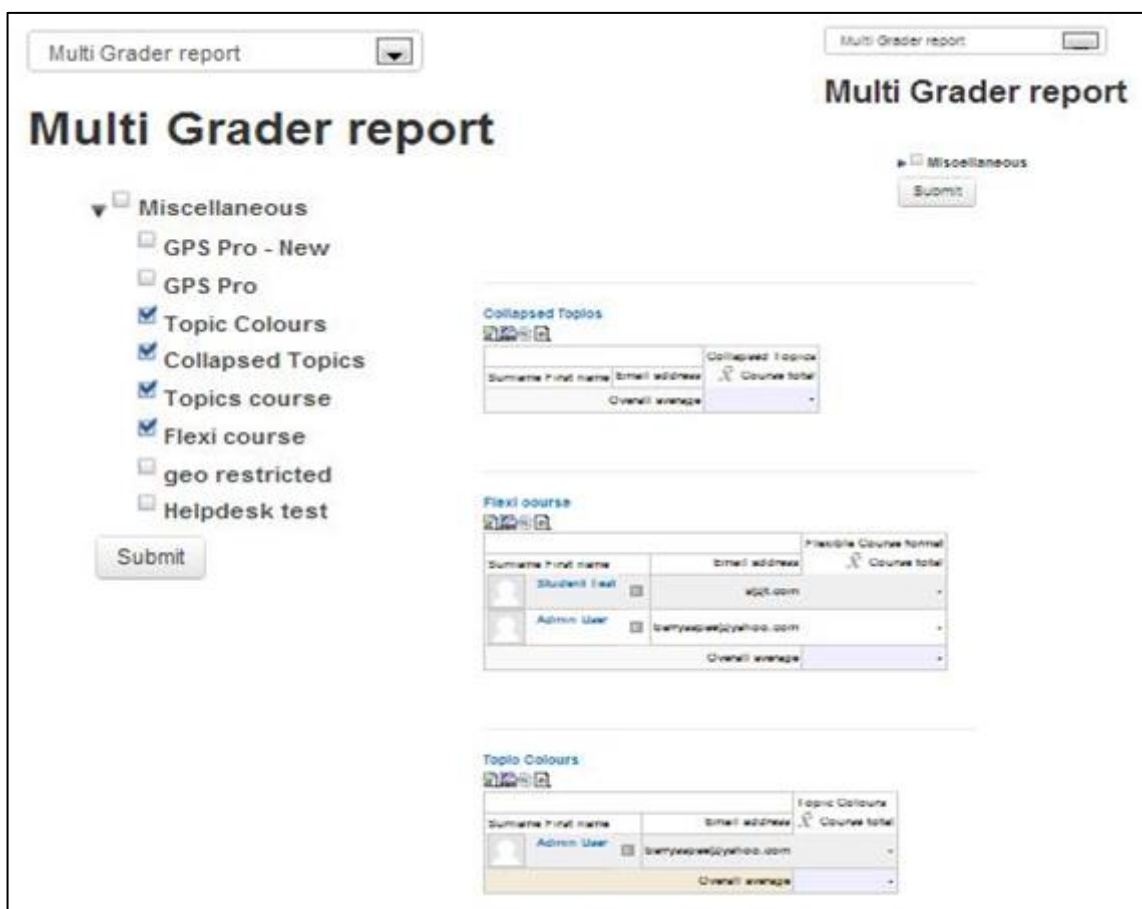


Рисунок 1.8 – Робота модуля Multi Course Grader Report

3. *Sitewide User Report*. Модуль генерує дерево курсів, з яких можна обрати потрібні (рис. 1.9) [6]. В результаті виводиться таблиця по кожному учаснику, кожного обраного курсу. З одного боку це дозволяє побачити інформацію одразу багатьох курсів, без потреби заходу на кожен курс окремо. Проте плюси на цьому

Інтелектуальна система аналізу даних LMS–системи Moodle для моніторингу навчального процесу в університеті закінчилися. З мінусів модуль просто видав купу розрізненої інформації, яка ніяким чином не проаналізована та не згрупована. Немає ніякої можливості зберегти отримані результати. Навіть скріншот сторінки не допоможе, бо якщо взяти навіть один курс і мати там наприклад 20 студентів, то отримаємо звіт на декілька сторінок. Також немає ніяких графіків, тобто візуального аналізу.

Рисунок 1.9 – Вигляд роботи модуля Sitewide User Report

4. *Rubrics Grading Report*. Модуль формує таблицю з результатами по кожному студенту курсу, в таблиці є коментарі, які додавав викладач коли оцінював роботи студента (рис. 1.10) [7]. Цей звіт дозволяє побачити, як змінювалася робота кожного зі студентів протягом всього курсу. Модуль дозволяє

Інтелектуальна система аналізу даних LMS–системи Moodle для моніторингу навчального процесу в університеті завантажити результат у вигляді csv файлу або Excel документу. Недоліки цього модуля такі ж як і у інших: звіт тільки по одному курсу. Тому потрібно на кожен курс заходити окремо, немає графіків, немає аналізу.

Student	Content (Max grade: 5)	Citing (Max grade: 5)	Readability (Max grade: 5)	Accessibility (Max grade: 5)	Feedback	Grade
Alan Andrews	Grade: 3 Level: Good quality Good, consistent, factual.	Grade: 4 Level: Excellent quality Correct format, good amount.	Grade: 3 Level: Good quality Reads very easily	Grade: 2 Level: Reasonable quality Ensure that top accessibility issues are avoided.	Overall good!	D (60.00 %)
Bruce Behan	Grade: 4 Level: Excellent quality Top quality, well done!	Grade: 3 Level: Good quality Correct format, good usage.	Grade: 4 Level: Excellent quality Reads very well.	Grade: 3 Level: Good quality Overall so so, check issues with link text.	Overall excellent!	C- (70.00 %)
Carl Clancy	Grade: 4 Level: Excellent quality Really well researched co	Grade: 3 Level: Good quality Correct format, all relevant.	Grade: 4 Level: Excellent quality Reads very comprehensively.	Grade: 3 Level: Good quality Passes basics, check for some link issues.	Overall, really great work!	C- (70.00 %)
Diana Drew	-	-	-	-	-	-
Summary	3.67	3.33	3.67	2.67		66.67

Рисунок 1.10 – Вигляд роботи модуля Rubrics Grading Report

5. *Grade History* (рис. 1.11). Цей модуль дозволяє переглядати інформацію по кожному викладачеві, який виставляв або змінював оцінки в курсі, коли і кому він це робив [8]. Проте відсутня можливість скачати результат, немає візуалізації у вигляді графіків.

Наприклад, який викладач був більш активний і перевіряв більшу кількість робіт, або частіше був на курсі та завантажував більше навчального контенту. Є просто історія зміни оцінок.

Date and time	Grade item	Username	Original grade	Revised grade	Email address	Surname	First name	Grader	Source	Overridden	Locked	Excluded from calculations
10/07/2014 06:03	Course total	admin			adam.silley@netpot.com.au	User	Admin		System	N	N	N
10/07/2014 06:03	Course total	admin		5.00000	adam.silley@netpot.com.au	User	Admin		Aggregation	N	N	N
10/07/2014 06:03	Course total	admin	5.00000	10.00000	adam.silley@netpot.com.au	User	Admin		Aggregation	N	N	N
10/07/2014 06:03	lesson	admin	5.00000	5.00000	adam.silley@netpot.com.au	User	Admin	User Admin	Gradebook	Y	N	N
10/07/2014 06:03	lesson	admin	5.00000	10.00000	adam.silley@netpot.com.au	User	Admin	User Admin	Gradebook	Y	N	N

Рисунок 1.11 – Вигляд роботи модуля Grade History

Таким чином, аналіз функціональних можливостей системи показав, що система не має вбудованих засобів для отримання зведеної інформації про результати навчання та хід навчального процесу у зручному для викладачів та керівників структурних підрозділів університету вигляді. У системі також відсутні засоби для реалізації алгоритмів інтелектуального аналізу навчального контенту. Це не дає можливості для проведення глибокого дескриптивного та предикативного аналізу даних про навчання, накопичених у системі. Що обумовлює необхідність розробки для вирішення цих задач інтелектуальної системи у вигляді логіну Moodle, який може бути інтегрованим у систему Moodle.

1.3 Методи та алгоритми інтелектуального аналізу освітніх даних

Використання у ЗВО освітніх інформаційних середовищ, сформованих на базі системи управління навчанням Moodle супроводжується накопиченням

Інтелектуальна система аналізу даних LMS-системи Moodle для моніторингу навчального процесу в університеті великих обсягів цифрового контенту стосовно освітнього процесу. Ріст обсягу освітніх даних сприяв появі нових напрямів у області інтелектуального аналізу даних – інтелектуального аналізу освітніх даних (англ. Data Mining Education) та навчальної аналітики (англ. Learning Analytics).

Використання алгоритмів інтелектуального аналізу даних в LMS-системах дозволяє реалізовувати збір даних про дії студентів всередині системи, візуалізацію цих даних для викладача з метою відслідковування дій студентів та подальший аналіз отриманих даних для прогнозування успішності студента та поліпшення процесу навчання (рис. 1.12).

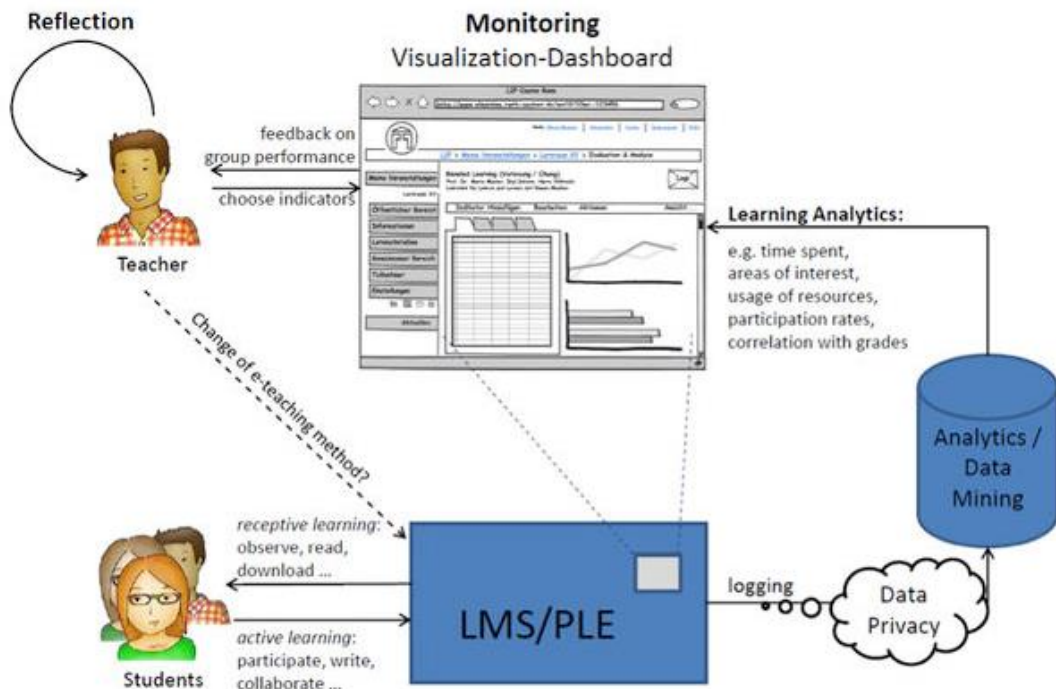


Рисунок 1.12 – Навчальна аналітика в LMS-системах

Learning Analytics та Data Mining Education широко використовують традиційні методи Data Mining: класифікацію, кластеризацію, пошук правил зв'язку, пошук послідовних шаблонів, Text Mining. Освітні дані мають складну структуру (ієрархічні, семантичні), що складає трудність для аналізу традиційними методами. Крім того алгоритми Data Mining мають справу з

даними, які не доходять до працівників закладів освіти – викладачів та керівництва.

Цифровий слід користувачів у системі Moodle, відображений у логах та базі даних, включає у себе наступні дані:

- кількість входів;
- час кожного входу;
- кількість звернень до кожного елементу курсу, який містить теоретичний матеріал та методичні рекомендації;
- кількість звернень до елементів курсу, які передбачають виконання практичних завдань;
- оцінки, отримані за кожне практичне завдання курсу, за яке передбачено отримання оцінки;
- підсумкова оцінка з дисципліни.

Для виявлення структури даних, які аналізуються, доцільно застосовувати алгоритми кластерного аналізу. При проведенні кластеризації здійснюється розбиття наявної множини об'єктів на підмножини таким чином, щоб елементи однієї підмножини істотно відрізнялися по деякому набору властивостей від елементів усіх інших підмножин.

У контексті задачі, яка розв'язується, в якості об'єктів виступають курси, які викладаються у певному семестрі певного навчального року для кожної групи студентів. Цей об'єкт представлений набором атрибутів, кожен з яких є певною характеристикою курсу у середовищі електронного навчання. У якості характеристик при проведенні кластерного аналізу було вирішено обрати наступні:

- активність студентів при вивченні навчального матеріалу курсу (загальна кількість переглядів матеріалів студентами поділена на кількість студентів у групі);
- активність студентів при виконанні поставлених практичних завдань (загальна кількість завантажень матеріалів студентами поділена на кількість

студентів у групі)

- кількість виконаних та оцінених практичних завдань (у % до загальної кількості таких завдань у курсі);
- активність викладача на курсі: (кількість переглядів матеріалів викладачем + кількість завантажень ним матеріалів поділена на кількість студентів у групі);
- кількість студентів, які не здали іспит/залік (у %);
- якість навчання – кількість студентів, які мають добре та відмінно (у %);
- успішність – середній бал з дисципліни (як зважене середнє отриманих студентами групи балів).

Розглянемо більш детально базові поняття та основні алгоритми кластерного аналізу даних. Основними етапами алгоритму кластеризації є такі:

- підготовка даних;
- вибір способу розрахунку міри близькості між об'єктами;
- вибір алгоритму: зумовлюється вихідними даними;
- виконання алгоритму;
- представлення результатів та інтерпретація.

Перший етап полягає у підготовці даних до кластерного аналізу [9]. Для представлення характеристик об'єктів до одного числового діапазону при проведенні кластерного аналізу здійснюється їх нормалізація. Існує багато підходів до нормалізації даних. Одним із них є лінійна min-max нормалізація:

$$x'_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \quad (1.1)$$

де x'_i – нормалізоване значення змінної x_i ;

x_{\max} , x_{\min} – максимальне та мінімальне значення змінної x ;

Дана формула розміщує усі значення в інтервалі [0, 1].

Характеристики, обрані для аналізу, є метриками, тому на наступному етапі кластерного аналізу для визначення міри близькості між об'єктами доцільно розраховувати міри несхожості – відстані.

Основні формули для розрахунку відстаней між об'єктами x_i і x_j представлено у таблиці 1.1.

Далі обирається алгоритм, часто доводиться використовувати кілька алгоритмів, комбінувати їх для більш точного результату.

Таблиця 1.1 – Міри відстаней між об'єктами для метричних шкал

Метрика	Формула
Відстань Евкліда	$d_E(x_i, x_j) = \sqrt{\sum_{t=1}^m (x_{it} - x_{jt})^2}$
Квадрат відстані Евкліда	$d_E^2(x_i, x_j) = \sum_{t=1}^m (x_{it} - x_{jt})^2$
Манхетенська відстань (відстань міських кварталів)	$d_H(x_i, x_j) = \sum_{t=1}^m x_{it} - x_{jt} $
Відстань Чебишева	$d_\infty(x_i, x_j) = \max_{1 \leq t \leq m} x_{it} - x_{jt} $

Реалізація обраного алгоритму дозволяє отримати результат у вигляді групування об'єктів за кластерами. Останній етап передбачає представлення отриманого групування у зручному для інтерпретації вигляді [10].

Існує багато алгоритмів кластерного аналізу. Серед алгоритмів кластеризації для вирішення поставленої задачі було обрано агломеративний ієрархічний [11] алгоритм та алгоритми k-means [12] і c-means [13], процедура яких є ітераційною. Опишемо кожен з цих алгоритмів більш детально.

Ієрархічний агломеративний алгоритм на початку кожен об'єкт розглядає як окремий кластер, після чого знаходить найближчі об'єкти та об'єднує їх у один

кластер. Ці етапи повторюються до тих пір, поки усі об'єкти набору даних не будуть об'єднані у один кластер. Для знаходження оптимальної кількості кластерів аналізують послідовність відстаней, на основі яких здійснювалося об'єднання алгоритмів у кластери на етапах ітерації алгоритму та знаходять етап, на якому відбувся перший різкий скачок зміни значень відстані. Оптимальні кількість кластерів буде рівна різниці між кількістю об'єктів у наборі даних, який аналізується, та номером етапу ітерації на якому відбувся цей скачок.

При об'єднанні кластерів на етапах ієрархічного алгоритму застосовують різні методи зв'язку, які є критеріями під час об'єднання кластерів. Найчастіше це метод найближчого сусіда – при об'єднанні кластерів за відстань між кластерами приймають відстань між найближчими об'єктами цих кластерів. Або метод дальнього сусіда – при об'єднанні кластерів за відстань між кластерами приймають відстань між найдальшими об'єктами цих кластерів.

Серед неієрархічних алгоритмів найбільш поширеними є чіткий алгоритм k-means та нечіткий алгоритм c-means. Ці алгоритми також є ітераційними, а оптимальність розбиття на кластери визначається шляхом оптимізації цільової функції, яка є внутрішньокластерною квадратичною похибкою розбиття і для алгоритму k-means має вигляд:

$$J = \sum_{j=1}^k \sum_{i=1}^{n_j} d^2(x_{ij}, c_j), \quad (1.2)$$

де c_j – j -й кластер, k – кількість кластерів,

x_{ij} – i -й об'єкт j -го кластера, n_j – кількість об'єктів у j -му кластері,

$d(x_{ij}, c_j)$ – відстань між i -м об'єктом j -го кластера та його центроїдом – центром ваги.

Центр ваги кластеру розраховують як точку у просторі ознак з координатами, що є середнім арифметичним усіх значень кожної ознаки об'єктів кластера.

Для виконання алгоритму k-means необхідно задати кількість кластерів, на які буде розбито сукупність об'єктів набору даних. На початковому етапі рандомно задаються центри кластерів, потім на кожному з етапів визначається відстань об'єктів до центрів та здійснюється перегрупування об'єктів до тих кластерів, центри яких є найближчими. Робота алгоритму припиняється, коли буде досягнуто мінімум цільової функції.

Нечіткий алгоритм кластеризації c-means є вдосконаленням алгоритму k-means, він визначає ймовірність віднесення об'єкту до кожного з кластерів, формуючи матрицю нечіткого розбиття. Цільова функція алгоритму c-means має вигляд:

$$J = \sum_{j=1}^k \sum_{i=1}^{n_j} u_{ji}^w d^2(x_{ij}, c_j), \quad (1.3)$$

де u_{ji} – елемент матриці нечіткого розбиття U , W – коефіцієнт нечіткості.

Застосування кластерного аналізу для обробки освітніх даних дає можливість розділити курси на кластери. Візуалізація цієї інформації дає уявлення про подальші стратегії при прийнятті освітніх рішень стосовно організації навчального процесу в університеті.

1.4 Постановка задачі

Розробка інтелектуальної системи для аналізу даних системи LMS Moodle, для моніторингу навчального процесу в університеті з вбудованими алгоритмами інтелектуального аналізу даних дозволяє оперативно отримувати інформацію про хід навчального процесу та видобувати знання, які є критично важливими для прийняття рішень у освітній сфері.

Об'єкт роботи – процес аналізу та моніторингу підготовки фахівців у закладах вищої освіти на базі платформ електронного навчання.

Предмет роботи – програмні засоби, методи та алгоритми інтелектуального аналізу даних LMS Moodle для моніторингу навчального процесу.

Мета роботи – підвищення ефективності моніторингу навчального процесу в університеті шляхом розробки плагіну LMS-системи Moodle з вбудованими алгоритмами інтелектуального аналізу даних.

Для досягнення поставленої мети було поставлено такі **завдання**:

1) здійснити аналіз та дослідити теоретичні засади моніторингу навчального процесу у закладах вищої освіти на базі LMS Moodle;

2) обґрунтувати вибір технологій та інструментальних засобів розробки плагіну для аналізу даних Moodle;

3) розробити, здійснити програмну реалізацію та впровадження системи аналізу даних LMS Moodle для моніторингу навчального процесу в університеті.

Сформуємо технічне завдання. Розроблювана інтелектуальна система повинна мати наступні функціональні можливості для аналізу та виведення зведеної інформації про успішність:

1) надавати можливість вибору груп із вказівкою навчального року, факультету, семестру, серед доступних для кожного викладача та керівника структурного підрозділу університету (кафедри, факультету, інституту, університету) груп і навчальних курсів, які відповідають його ролі у системі;

2) на основі вилученої із бази даних Moodle інформації забезпечувати можливість візуалізації результатів у зручному для користувача форматі для відібраних груп результатів поточного стану вивчення курсів студентами груп у вказаному семестрі та навчальному році у табличному вигляді з можливістю групування по групам та семестрам;

3) передбачати можливість експорту результатів у форматах XLSX, DOCX, PDF, CSV, XML, TXT.

4) передбачати графічну візуалізацію отриманих даних у вигляді діаграм:

- у національній шкалі для іспиту: не здано (менше 60 балів), задовільно (60-74 бали), добре (75-89), відмінно (90-100);
- у національній шкалі для заліку: не здано (менше 60 балів), зараховано (60-100 балів);
- у шкалі ECTS для іспиту та заліку: F (0-34 балів), FX (35-59 балів), E (60-66 балів), D (67-74 балів), C (75-81 балів), B (82-89 балів), A (90-100 балів);

5) при здійсненні кластерного аналізу передбачати можливість вибору алгоритму, характеристик груп, налаштування параметрів алгоритму та візуалізації його результатів із можливістю розширених налаштувань для адміністратора Moodle.

Етапи розробки: дослідження предметної сфери; розробка альфа версії; тестування та збір зауважень; додавання алгоритмів інтелектуального аналізу; реліз бета версії та її тестування; збір зауважень та пропозицій; виправлення помилок та всіх знайдених недоліків; реліз та впровадження фінальної версії.

Ризики проекту: хвороба виконавця; технічні проблеми; відсутність необхідної інформації; зміна у ТЗ через попередні причини.

Встановлені вимоги до проекту є основою для подальшого проектування та реалізації системи.

Висновки до розділу 1

В даному розділі здійснено аналіз системи управління навчанням Moodle. Виявлено, що функції адміністрування LMS-системи Moodle охоплюють управління користувачами з ролями (викладач, студент, адміністратор тощо), які визначають права доступу до навчального контенту, організації студентів у певні групи, формування курсів, складання звітності. Основою навчального простору Moodle є дистанційні курси, призначені для вивчення навчальних дисциплін, які включають навчальні матеріали, викладачів і студентів із забезпеченням взаємодії

між ними, контроль отримуваних знань і накопичення інформації про процес навчання. Moodle має засоби для додавання до курсу навчального матеріалу у будь-якому форматі, організації дистанційних занять, контролю та оцінювання результатів навчання. Платформа підходить для реалізації дистанційних, змішаних форм навчання й є додатком до традиційних очних форм навчання.

Установлено, що у навчальних закладах Moodle використовується як фреймворк для власних програмних рішень. Є безкоштовною Open Source-системою типу LMS, написана на PHP, використовує MySQL та Perl, працює з MySQL, PostgreSQL, Microsoft SQL Server або Oracle. Завдяки розвиненій модульній архітектурі, можливості Moodle можуть легко розширюватися сторонніми розроблювачами.

З'ясовано, що моніторинг є механізмом, який полягає у зборі й аналізі широкого спектра критеріїв і показників, аналіз яких дозволяє безперервно спостерігати за станом професійної підготовки фахівців у закладах освіти. LMS Moodle має інструменти моніторингу, які дозволяють аналізувати процеси на рівні окремого електронного курсу або системи в цілому: відслідковувати отримані студентами оцінки, завантажені та виконані завдання, активність студентів та викладачів на курсі. Однак зведену інформацію отримати не можливо у випадку, коли здійснюється аналіз роботи студентів сукупності курсів.

Здійснений аналіз додаткових плагінів, які можна встановити в Moodle (Grade Distribution, Multi Course Grader Report, Sitewide User Report, Rubrics Grading Report, Grade History), показав, що вони не надають можливості для отримання оперативних даних про результативність навчання у сукупності груп на рівні окремого викладача чи структурного підрозділу університету. Недостатньо розвинений функціонал для аналізу ходу навчального процесу не дає можливості завідувачу кафедри, декану факультету чи проректору оперативно отримувати звітну інформацію та проводити моніторинг. У системі також відсутні засоби для реалізації алгоритмів інтелектуального аналізу навчального контенту. Тому є потреба у розширенні функціональності системи.

Було проаналізовано можливі підходи до розширення можливостей Moodle: зміна ядра системи, інтеграція стороннього ПЗ, використання плагінів для моніторингу сторонніх розробників, написання власного плагіну. Та вирішено для розв'язання поставленої задачі зупинитися на розробці власного плагіну, адаптованого до потреб конкретного навчального закладу.

Для аналізу даних Moodle було обрано ієрархічний алгоритм кластерного аналізу та алгоритми k-means і c-means. У якості об'єктів виступають курси, які викладаються у певному семестрі певного навчального року для кожної групи студентів з атрибутами: активність студентів при вивченні навчального матеріалу, активність студентів при виконанні поставлених завдань, активність викладача, кількість студентів, які не здали іспит/залік, якість навчання, успішність.

До розроблюваного плагіну було сформовано специфікації вимог.

2 ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТАЛЬНІ ЗАСОБИ РОЗРОБКИ ПЛАГІНУ ДЛЯ АНАЛІЗУ ДАНИХ MOODLE

2.1 Середовище розробки PhpStorm

IDE PhpStorm – це потужне інтегроване середовище розробки від компанії JetBrains, призначене для розробки на PHP [14]. Воно надає широкий спектр інструментів і функцій для полегшення процесу написання, налагодження та тестування коду. Розглянемо основні можливості PhpStorm.

Редактор коду:

- підсвічування синтаксису: розпізнає синтаксис PHP, HTML, CSS, JavaScript та інших мов;
- автозаповнення коду: надає пропозиції для завершення коду, що значно прискорює процес написання;
- інспекція коду: автоматично виявляє помилки та попереджає про потенційні проблеми в коді;
- refactoring: інструменти для безпечного та ефективного рефакторингу коду;
- інтеграція з системами контролю версій;
- підтримує Git, SVN, Mercurial, Perforce та інші системи контролю версій;
- вбудовані інструменти для злиття, вирішення конфліктів і перегляду історії змін.

Налагодження та профілювання. Вбудований відладчик дозволяє встановлювати точки зупину (breakpoints), крокувати через код та переглядати значення змінних у реальному часі. Профілювальник: інструмент для вимірювання продуктивності коду та виявлення вузьких місць.

Інструменти для тестування: підтримка PHPUnit та інших фреймворків для автоматизованого тестування; інтеграція з інструментами для поведінкового тестування (BDD), такими як Behat.

Інтеграція з базами даних та SQL: вбудований клієнт для роботи з базами даних, що дозволяє виконувати SQL-запити, переглядати структуру таблиць і редагувати дані; підтримка різних СУБД, включаючи MySQL, PostgreSQL, SQLite та інші.

Інтеграція з фреймворками та платформами: підтримка популярних PHP-фреймворків, таких як Symfony, Laravel, Zend Framework, Yii; інструменти для роботи з JavaScript-фреймворками (React, Angular, Vue.js) і CSS-препроцесорами (Sass, Less).

Розширюваність: можливість встановлення плагінів для розширення функціональності IDE; власний плагін-менеджер для простого пошуку, встановлення та оновлення плагінів.

Інструменти для командної роботи: інтеграція з системами трекінгу завдань (JIRA, YouTrack); підтримка інструментів для безперервної інтеграції та доставки (CI/CD).

Таким чином, PhpStorm є потужним та багатофункціональним середовищем розробки, яке значно полегшує процес створення, налагодження та тестування PHP-застосунків. Завдяки широкому спектру інструментів та інтеграцій, PhpStorm є чудовим вибором для професійних розробників, які прагнуть підвищити продуктивність та якість своєї роботи.

2.2 Гіпертекстовий препроцесор PHP

PHP – Hypertext PreProcessor, попередня назва: Personal Home Page Tools, є скриптовою мовою програмування, яка була створена для генерації HTML-сторінок на стороні вебсервера [15]. PHP є однією з найпоширеніших мов, що використовуються у сфері веброзробок, разом із Java, .NET, JavaScript, Python, Ruby. PHP підтримується переважною більшістю хостинг-провайдерів і є проектом відкритого програмного забезпечення.

PHP інтерпретується вебсервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, тому що браузер отримує готовий html-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніхто не забороняє використовувати PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта.

Можна виділити наступні основні сфери, де використовується мова програмування PHP.

1. Створення веб-додатків та скриптів, які виконуватиме сервер. Найбільш поширена сфера застосування PHP, адже спочатку ця мова створювалася саме під веб. Для роботи PHP-скриптів необхідний сервер, CGI-додаток (парсер) та програмне забезпечення клієнта (браузер).

2. Створення сценаріїв командного рядка. Подібні міні-програми працюватимуть на ПК із будь-якою операційною системою, адже для їх виконання необхідний лише парсер. Зважаючи на те, що PHP відрізняється потужним інструментарієм для роботи з рядками, подібні сценарії зазвичай обробляють текстові дані.

3. Створення графічних інтерфейсів. У PHP є маса відгалужень, які дозволяють застосовувати цю мову програмування для різних завдань. І одне з таких відгалужень – PHP-GTK, використовується для створення застосунків із графічним інтерфейсом користувача.

Можна виділити такі переваги мови програмування PHP:

– зрозумілий та простий синтаксис – швидко освоїти PHP може навіть новачок у програмуванні. Код мови читається легко в будь-якій сфері використання (як при створенні невеликих скриптів, так і при написанні серйозних програмних продуктів);

– гнучкість та кросплатформеність – PHP може працювати на будь-яких сучасних операційних системах (MacOS, Windows, Linux), а скрипти, написані цією мовою, успішно функціонують різним серверним ПЗ (Apache, IIS, Nginx);

- масштабованість – використовуючи PHP, можна досягти високої продуктивності ПЗ, а програми для веб, написані цією мовою, можуть справлятися буквально з будь-яким трафіком;
- використання разом із HTML – додавши на HTML-сторінку код PHP, можна реалізувати будь-який контент, що динамічно змінюється. PHP-код не порушує структуру документа HTML;
- безліч докладних посібників в інтернеті у вільному доступі та покрокових інструкцій для навчання мови програмування PHP та роботи з ним, проєкт має і свій офіційний сайт;
- просте вирішення проблем, що виникли, оскільки у мережі є безліч різноманітних форумів, які присвячені PHP, величезна спільнота завжди може прийти на допомогу у разі виникнення будь-якої складної ситуації.

2.3 Мова програмування JavaScript

JavaScript (JS) – динамічна, об’єктно-орієнтована прототипна мова програмування [16]. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв вебсторінок, що надає можливість на боці клієнта взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки.

JavaScript класифікують як прототипну (підмножина об’єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам’яттю, прототипне наслідування, функції як об’єкти першого класу.

JavaScript – це мова програмування, яку розробники використовують для створення інтерактивних веб-сторінок. Функції JavaScript можуть покращити

зручність взаємодії користувача з веб-сайтом: від оновлення стрічки новин у соціальних мережах до відображення анімації та інтерактивних карт. JavaScript є мовою програмування розробки скриптів для виконання на стороні клієнта, що робить його однією з базових технологій у всесвітній мережі Інтернет. Наприклад, карусель зображення, що випадає по кліку меню і кольори елементів, що динамічно змінюються, на веб-сторінці, які ви бачите під час перегляду сторінок в Інтернеті, виконані за допомогою JavaScript.

Історично веб-сторінки були статичними, схожими на сторінки книги. Статична сторінка здебільшого відображала інформацію у фіксованому вигляді та не виконувала всього того, що ми зараз очікуємо від сучасного сайту. Мова JavaScript виникла як технологія на стороні браузера, що дозволяє зробити веб-програми динамічнішими. Використовуючи його, браузери могли реагувати на взаємодію з користувачем та змінювати розташування контенту на веб-сторінці.

У міру розвитку мови розробники JavaScript створили бібліотеки, фреймворки та практики програмування та почали використовувати його за межами веб-браузерів. Сьогодні JavaScript можна використовувати для розробки як на стороні клієнта, так і сервері.

Всі мови програмування працюють шляхом переведення синтаксису в машинний код, який виконує операційна система. JavaScript у широкому розумінні можна віднести до категорії скриптових або інтерпретованих мов. Код JavaScript інтерпретується, тобто безпосередньо перекладається код машинного мови двигком JavaScript. В інших мовах програмування компілятор обробляє весь код машинного на окремому етапі.

JavaScript – це комп'ютерна програма, яка виконує код JavaScript. Перші двигуни JavaScript були лише інтерпретаторами, але всі сучасні двигуни використовують для підвищення продуктивності JIT-компіляцію або компіляцію під час виконання.

JavaScript на стороні клієнта відноситься до того, як JavaScript працює у вашому браузері. У цьому випадку движок JavaScript знаходиться всередині коду браузера. Всі основні веб-браузери мають власні вбудовані движки JavaScript.

Розробники веб-застосунків пишуть код JavaScript з різними функціями, пов'язаними з різними подіями, такими як клацання миші або наведення курсору. Ці функції вносять зміни до HTML та CSS. Нижче наведено огляд того, як працює JavaScript на стороні клієнта:

- браузер завантажує веб-сторінку, коли ви її відвідуєте;
- у процесі завантаження браузер перетворює сторінку та всі її елементи, такі як кнопки, ярлики та поля, що випадають, в структуру даних, звану об'єктною моделлю документа (DOM);
- двигун JavaScript браузера перетворює код JavaScript на байткод. Цей код є посередником між синтаксисом JavaScript та машиною;
- різні події, такі як клацання миші по кнопці, викликають виконання зв'язаного блоку JavaScript. Потім двигун інтерпретує байткод і вносить зміни до DOM;
- браузер відображає новий DOM.

JavaScript на стороні сервера відноситься до мови кодування в логіці внутрішнього сервера. У цьому випадку движок JavaScript знаходиться безпосередньо на сервері. Функція JavaScript на стороні сервера може звертатися до бази даних, виконувати різні логічні операції та реагувати на різні події, які запускає операційна система сервера. Основна перевага скриптингу на стороні сервера полягає в тому, що ви можете значно налаштувати відповідь сайту відповідно до ваших вимог, прав доступу та інформаційних запитів сайту.

Сторона клієнта та сторона сервера. Слово динамічний описує як клієнтський, і серверний JavaScript. Динамічна поведінка – це здатність оновлювати відображення веб-сторінки для створення нового контенту в міру необхідності. Різниця між JavaScript на стороні клієнта та на стороні сервера полягає у генеруванні нового контенту. Код на стороні сервера динамічно генерує

новий контент, використовуючи логіку програми та змінюючи дані з бази даних. JavaScript на стороні клієнта, з іншого боку, динамічно генерує новий контент усередині браузера, використовуючи логіку інтерфейсу користувача і змінюючи контент веб-сторінки, яка вже знаходиться на стороні клієнта. Сенса трохи відрізняється в цих двох контекстах, але вони пов'язані, і обидва підходи працюють разом для покращення користувацького досвіду.

Інша різниця між двома видами використання JavaScript, крім реалізації динамічних функцій, полягає в ресурсах, до яких код може отримати доступ. На стороні клієнта браузер контролює середовище JavaScript. Код може отримати доступ тільки до тих ресурсів, які дозволяє браузер. Наприклад, він не може записати вміст на жорсткий диск, якщо ви не натиснете на кнопку завантаження. З іншого боку, функції на стороні сервера можуть отримати доступ до всіх ресурсів серверної машини за необхідності.

Бібліотеки JavaScript – це колекції попередньо написаних фрагментів коду, які веб-розробники можуть повторно використовувати для виконання стандартних функцій JavaScript. Код бібліотеки JavaScript підключається до іншого коду проекту за необхідності, розробники можуть використовувати для поліпшення його функціональності.

2.4 HTML та каскадні таблиці стилів CSS

HTML (англ. HyperText Markup Language) стандартизована мова розмітки документів для перегляду вебсторінок у браузері [17]. Браузери отримують HTML документ від сервера за протоколами HTTP/HTTPS або відкривають з локального диска, далі інтерпретують код в інтерфейс, який відобразатиметься на екрані монітора.

Елементи HTML є будівельними блоками сторінок HTML. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовану сторінку. HTML надає засоби для

створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML окреслені тегами, написаними з використанням кутових дужок. Теги на кшталт `` чи `<input />` безпосередньо виводять вміст на сторінку. Інші теги, такі як `<p>`, оточують текст і надають інформацію про нього, а також можуть включати інші теги як піделементи. Браузери не показують теги HTML, але використовують їх для інтерпретації вмісту сторінки.

HTML впроваджує засоби для:

- створення структурованого документа шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- отримання інформації зі Всесвітньої мережі через гіперпосилання;
- створення інтерактивних форм;
- включення зображень, звуку, відео, та інших об'єктів до тексту.

В HTML можна вбудовувати програми, написані на скриптових мовах, наприклад JavaScript, які впливають на поведінку та вміст вебсторінок. Включення CSS визначає вигляд і компоновання вмісту.

CSS (англ. Cascading Style Sheets) є спеціальною мовою стилю сторінок, що використовується для опису їхнього зовнішнього вигляду [18]. Самі ж сторінки написані мовами розмітки даних. CSS є основною технологією всесвітньої павутини, поряд із HTML та JavaScript. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів. Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадне або блокове верстання) прийшло на заміну табличному верстанню вебсторінок. Головна перевага блокового верстання — розділення вмісту сторінки (даних) та його візуальної презентації.

CSS використовується авторами та відвідувачами вебсторінок, щоб визначити кольори, шрифти, верстання та інші аспекти вигляду сторінки. Одна з головних переваг — можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS).

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори тощо. CSS також дозволяє адаптувати контент до різних апаратних та програмних платформ.

Один і той самий HTML або XML документ може бути відображений по-різному залежно від використаного CSS. Стили для відображення сторінки можуть бути:

1) стилі автора (інформація надана автором сторінки):

- зовнішні таблиці стилів (англ. *stylesheet*), найчастіше окремий файл або файли *.css*;
- внутрішні таблиці стилів, включені як частина документу або блоку;
- стилі для окремого елемента;

2) стилі користувача: локальний *.css*-файл, вказаний користувачем для використання на сторінках і вказаний в налаштуваннях браузера (наприклад Opera);

3) стилі переглядача (браузера): стандартний стиль переглядача, наприклад стандартні стилі для елементів, визначені браузером, використовуються коли немає інформації про стиль елемента або вона неповна.

Стандарт CSS визначає порядок та діапазон застосування стилів, тобто, в якій послідовності і для яких елементів застосовуються стилі. Таким чином,

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система аналізу даних LMS–системи Moodle для моніторингу навчального процесу в університеті

використовується принцип каскадності, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена загальнішими стилями.

2.5 Мова структурованих запитів SQL

SQL (англ. Structured query language) – декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних [19]. Сама по собі SQL не є ані системою керування базами даних, ані окремим програмним продуктом. На відміну від дійсних мов програмування (C або Pascal), SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати як інструкції для керування даними. Окрім цього, стандарт SQL містить функції для визначення зміни, перевірки та захисту даних.

SQL – це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також керування базами даних. Багато баз даних підтримує SQL з розширеннями до стандартної мови. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення і вилучення даних за допомогою використання системи керування і адміністративних функцій. SQL також включає CLI (англ. Call Level Interface) для доступу і керування базами даних дистанційно.

Основу бази даних SQL Server утворює сервер або ядро бази даних. Ядро бази даних відповідає за обробку запитів, які надходять від клієнтів, і передачу відповідних результатів клієнтським компонентам.

Перша версія SQL була розроблена на початку 1970-х років у IBM. Ця версія мала назву SEQUEL і була призначена для обробки та пошуку даних, що містилися в реляційній базі даних IBM, System R. Мова SQL надалі була стандартизована Американськими Держстандартами (ANSI) в 1986. На початку SQL була запланована як мова запитів і управління даними, а пізніші модифікації SQL створені продавцями системи управління базами даних, які додали

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система аналізу даних LMS–системи Moodle для моніторингу навчального процесу в університеті

процедурні конструкції, control-of-flow команд і темпоральні розширення мов. З випуском стандарту SQL:1999 такі розширення були формально запозичені як частина мови SQL через Persistent Stored Modules (SQL/PSM).

Критика SQL включає відсутність крос-платформенності, невідповідну обробку відсутніх даних (дивіться Null (SQL)). Часто це неоднозначна граматика і семантика мови.

2.6 Бібліотеки для розробки плагіна Moodle

Використання різноманітних бібліотек є ключовим для забезпечення аналізу та моніторингу даних системи Moodle з високою точністю та швидкістю. Одними з найбільш важливих інструментів у цьому процесі є бібліотеки Rubix ML, JavaScript – jQuery, jQuery CheckTree, Apache ECharts, ECharts-GL, dat.GUI.

Rubix ML – це бібліотека машинного навчання для PHP, яка надає широкий набір інструментів для побудови, тренування і використання моделей машинного навчання [20]. Вона розроблена для легкого інтегрування в проекти на PHP та має інтуїтивно зрозумілий API. Основні можливості Rubix ML:

- підтримка різних типів моделей;
- попередня обробка даних;
- побудова конвеєрів (pipelines);
- тюнінг гіперпараметрів;
- оцінка моделей;
- підтримка багатопоточності.

Rubix ML є потужним інструментом для реалізації машинного навчання на PHP. Завдяки його інтуїтивному API, підтримці різних алгоритмів і зручним інструментам для обробки даних, Rubix ML може бути корисним для розробників, які працюють у сфері машинного навчання та прагнуть інтегрувати ці технології у свої PHP-проекти.

jQuery – популярна JavaScript-бібліотека з відкритим кодом [21]. Вона була представлена у січні 2006 року у BarCamp NYC Джоном Ресігом (John Resig). Згідно з дослідженнями організації W3Techs, jQuery використовується понад половиною від мільйона найвідвідуваніших сайтів. jQuery є найпопулярнішою бібліотекою JavaScript, яка посилено використовується на сьогоднішній день. jQuery є вільним програмним забезпеченням під ліцензією MIT.

Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій, і розробки AJAX-застосунків. jQuery також надає можливості для розробників, для створення плагінів у верхній частині бібліотеки JavaScript. Використовуючи ці об'єкти, розробники можуть створювати абстракції для низькорівневої взаємодії та створювати анімацію для ефектів високого рівня. Це сприяє створенню потужних і динамічних вебсторінок.

jQuery CheckTree – це плагін jQuery, який бере вкладений список і перетворює його на ієрархічне дерево, яке можна згортати, розгортати та перевіряти [22]. Це дозволяє додавати прапорці біля кожного елемента списку в списку, щоб розгортати/згортати його дочірні елементи. Чудово підходить для швидкого створення списку підпунктів, які показують зв'язок між ними. Плагін jQuery наразі доступний для використання в усіх особистих або комерційних проектах за ліцензіями MIT і GPL. Це означає, що ви можете вибрати ліцензію, яка найкраще підходить для вашого проекту, і використовувати її відповідно.

Apache ECharts – потужна, інтерактивна бібліотека для візуалізації даних, яка є частиною проектів Apache [23]. Вона надає широкий набір інструментів для створення графіків, діаграм і карт, що дозволяє легко інтегрувати візуалізації в веб-додатки. ECharts написана на JavaScript і підтримує різні платформи та браузери. Її основні можливості:

- різноманітні типи графіків: лінійні, стовпчасті, кругові діаграми, теплові карти, географічні карти та інші;
- інтерактивність: масштабування, прокрутка, вибір елементів, спливаючі

підказки;

- адаптивність: підтримка різних пристроїв і автоматична адаптація до розмірів контейнера;
- гнучке налаштування: налаштування зовнішнього вигляду, підтримка тем і стилів;
- робота з великими даними: оптимізація для великих обсягів даних з використанням WebGL.

Apache ECharts — це багатofункціональна бібліотека для візуалізації даних, яка дозволяє створювати різноманітні та інтерактивні графіки з мінімальними зусиллями. Завдяки підтримці різних типів візуалізацій, інтерактивності та адаптивності, ECharts є чудовим інструментом для розробників, які бажають інтегрувати потужні засоби візуалізації у свої веб-додатки.

ECharts-GL – плагін для бібліотеки Apache ECharts, який додає можливості тривимірної (3D) візуалізації. Він дозволяє створювати інтерактивні 3D-графіки та діаграми, що розширює функціональність стандартної бібліотеки ECharts [24]. Основні можливості echarts-gl:

- 3D-графіки: підтримка різних типів 3D-графіків, таких як розсіювання, поверхні, бари та лінії;
- географічні карти: 3D-географічні карти для більш деталізованої візуалізації геопросторових даних;
- системи координат: підтримка 3D-систем координат для точного позиціонування даних у тривимірному просторі;
- покращена інтерактивність: додавання нових інструментів для масштабування, обертання та панорамування 3D-графіків;
- WebGL: використання технології WebGL для рендерингу, що забезпечує високу продуктивність і гладку анімацію.

Dat.GUI – легкий JavaScript-плагін для створення графічного інтерфейсу управління, який дозволяє розробникам легко додавати інтерактивні контролери для зміни параметрів та властивостей додатків у реальному часі [25]. Це корисний

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система аналізу даних LMS–системи Moodle для моніторингу навчального процесу в університеті

інструмент для налагодження та тестування, оскільки він надає можливість швидко змінювати значення і спостерігати за їх впливом безпосередньо в браузері.

DendrogramViewer – набір бібліотек для відображення дендрограм за спеціально сформованим деревом у форматі json [26]. Підтримується зміна кольорів, відображення міток, зміна розміру, поділ на кластери та збереження малюнку. Плагін був використаний для показу дендрограми після обробки даних за алгоритмом ієрархічного аналізу.

TableExport – плагін до jQuery, який може експортувати таблиці у різних форматах, таких як XLS(Excel 2003), XLSX(Excel 2007), DOC(Word), PDF, CSV, XML, TXT та інші [27]. Використовується для експорту зібраних даних.

Висновки до розділу 2

У другому розділі було розглянуто та обґрунтовано вибір технологій та інструментальних засобів розробки системи аналізу даних системи Moodle, з метою проведення оперативного та динамічного моніторингу навчального процесу в університеті. Для створення плагіну до LSM Moodle у якості основної мови програмування обрано PHP у середовищі розробки PhpStorm. Для додання інтерактиву та динамічності вебсторінок застосовано мову JavaScript. Оформлення та відображення даних для користувача відбувається за допомогою HTML та CSS. Для роботи з базою даних PostgreSQL було застосовано вбудовані засоби Moodle та прямі запити до БД на мові SQL. Візуалізація результатів проведеного аналізу здійснювалася з використанням бібліотек Apache Echarts та DendrogramViewer. Для додавання 3D графіки застосовано плагін ECharts-GL, а для додавання до графіків панелі управління використано бібліотеку dat.GUI. Для експорту даних із результатами зведеного аналізу про якість навчання та хід навчального процесу використано плагін jQuery TableExport.

3 ПРОЄКТУВАННЯ, ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ВПРОВАДЖЕННЯ СИСТЕМИ МОНІТОРИНГУ НАВЧАЛЬНОГО ПРОЦЕСУ

3.1 Архітектура системи у відкритих кодах Moodle

Moodle є вебзастосунком із відкритим вихідним кодом, її код є у відкритому доступі на системі Github. Вона написана на мові програмування PHP, дані зберігаються у SQL базі даних (наприклад PostgreSQL, MySQL, MSSQL) та у файловій системі (завантажені та згенеровані файли). Система має ядро та набір базових плагінів, що дозволяє легко її розширювати та змінювати за допомогою розширень – плагінів та модулів [28].

Moodle має наступні основні частини:

1) *код*, звичайно розташований у такій директорії, як `/var/www/moodle` або `~/htdocs/moodle`, доступ на запис у цю директорію не повинен надаватися веб-серверу;

2) *база даних* – реляційна RDMS (relational database management system), Moodle додає префікс до всіх імен таблиць, тому він може ділити базу даних з іншими додатками;

3) *директорія moodledata*, у якій Moodle зберігає файли, що завантажуються і генеруються, вона повинна бути доступна для запису з боку веб-серверу, з метою безпеки повинна перебувати поза кореневою директорією веб-сервера.

Конфігураційна інформація для всіх трьох складових частин зберігається у файлі з назвою `config.php` у кореневій директорії установки Moodle з ім'ям `moodle`. Як приклад вміст файлів цієї директорії представлено на рисунку 3.1. В Moodle реалізована класична трьох шарова архітектура Model (DB)-Library-View (`index.php`), представлена на рисунку 3.2. Розробник може будувати свої шари, підключати свої бібліотеки. Відсутня строга архітектура залежностей та зв'язків. Папка `moodledata` призначена для збереження динамічних файлів Moodle (картинки, кеші, завантажені файли тощо).

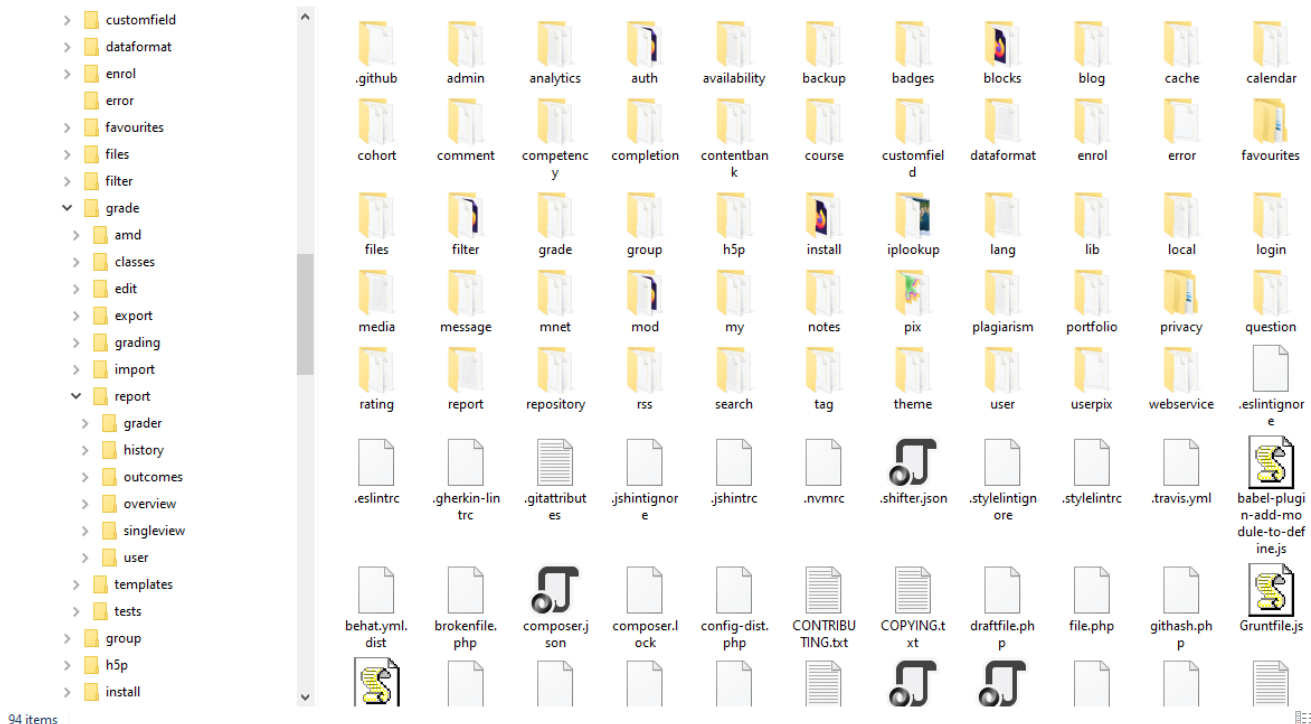


Рисунок 3.1 – Список файлів і папок в Moodle

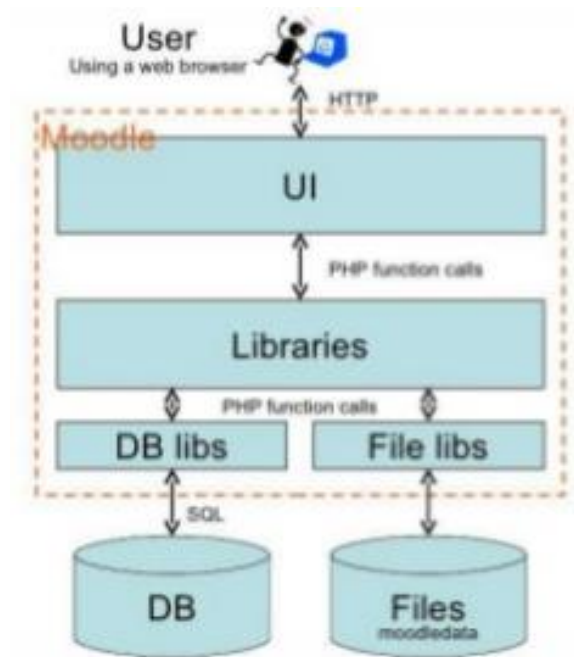


Рисунок 3.2 – Архітектура Moodle

Користувачі взаємодіють із системою Moodle за допомогою своїх веб браузерів. Процес взаємодії полягає у відправленні відповідей на HTTP-запити.

Важливим аспектом архітектури Moodle є *простір імен URL* і спосіб доставки даних URL різним сценаріям. У цьому випадку Moodle використовує стандартний підхід мови PHP. Наприклад, рядок URL для перегляду головної сторінки курсу буде мати такий вигляд:

`.../course/view.php?id=123` , де

123 – унікальний ідентифікатор курсу в базі даних.

Рядок URL для перегляду дискусії форуму буде виглядати аналогічно:

`.../mod/forum/discuss.php?id=456789`.

Тобто, ці сценарії `course/view.php` або `mod/forum/discuss.php` будуть обробляти ці запити. Розробнику для розуміння того, як Moodle обробляє певний запит, варто розглянути рядок URL і почати читати код у ньому. Ці рядки URL постійні, вони не змінюються у випадку перейменування курсу або у тому випадку, коли модератор переміщує дискусію в інший форум (ця властивість рядків URL є корисною). Альтернативним підходом, що може здатися корисним, є *єдина точка входу*:

`.../index.php/[додаткова-інформація-для-перетворення-запиту-в-унікальний]`.

Після цього окремий сценарій `index.php` буде обробляти запити певним чином. Цей підхід додає рівень абстракції.

Як система з відкритим вихідним кодом, Moodle побудовано на основі великої кількості розширень – плагінів, які працюють разом з ядром системи. Цей підхід вдалий, тому що він дозволяє змінювати й розширювати можливості Moodle різними способами. Однак, виконання значних змін коду може привести до великих проблем при оновленні системи, навіть у випадку використання зручної системи контролю версій.

Дозволяючи робити так багато змін і додавати стільки нових функцій, скільки можливо за умови їхньої реалізації у рамках самостійних розширень, взаємодіючих з ядром Moodle за допомогою описаного API, можна полегшити завдання по зміні можливостей Moodle користувачами відповідно до їх потреб, а

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система аналізу даних LMS–системи Moodle для моніторингу навчального процесу в університеті

також по поширенню внесених змін, при цьому зберігаючи можливість відновлення ядра системи Moodle.

Строго типізовані розширення передбачають те, що доведеться розробляти різні типи розширень і використовувати різні API залежно від того, який тип функцій необхідно реалізувати. Існує повний список таких типів розширень додатка Moodle (https://docs.moodle.org/dev/Plugin_types) деякі з них наведені у таблиці 3.1.

Таблиця 3.1 – Деякі типи розширень / плагінів

Plugin type	Component name (Frankenstyle)	Moodle path
activity modules (модулі діяльності)	mod	/mod
quiz reports (звіти з опитування)	quiz	/mod/quiz/report
blocks (блоки бокової панелі)	block	/blocks
question types (тип питання)	qtype	/question/type
admin tools (панель адміністратора)	tool	/admin/tool

Кожне розширення Moodle – плагін, має свою папку – точку входу і набір обов’язкових файлів. Точка входу – це клас, який наслідується від базового класу потрібного типу плагіна. Цей клас може переписувати методи батьківського класу [29].

Плагін має *тип* і *назву*, які спільно формують складену назву розширення, відому, як *Frankenstyle*. Тип розширення і його назва задають шлях до директорії

розширення. Тип розширення задає префікс, а ім'я директорії є назвою розширення. Нижче наведено у таблиці 3.2 кілька прикладів.

Таблиця 3.2 – Приклади розширень

Тип розширення	Назва розширення	Frankenstyle	Директорія
mod (модуль діяльності)	forum	mod_forum	mod/forum
block (блок бокової панелі)	navigation	block_navigation	blocks/navigation
qtype (тип питання)	shortanswer	qtype_shortanswer	question/type/shortanswer
quiz (звіт з тестування)	statistics	quiz_statistics	mod/quiz/report/statistics

Директорія Moodle /grade містить папки: export, import, report, які є папками плагінів, що відповідають за експорт, імпорт та отримання звітів із журналу оцінок (рис. 3.3).

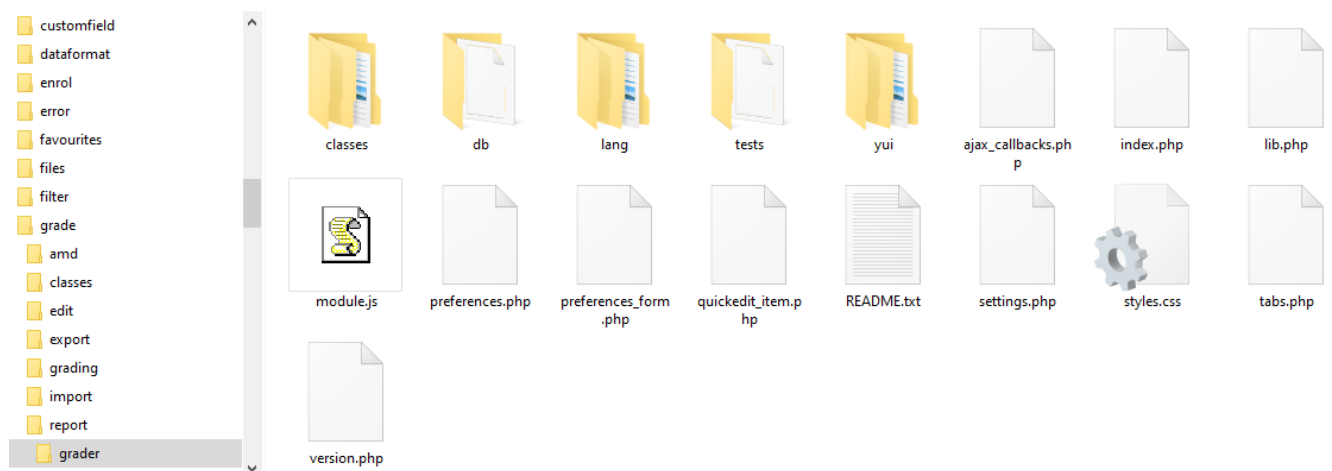


Рисунок 3.3 – Список файлів і папок в директорії Moodle grade

У представленому плагіні є папки `classes` (зберігають спеціальні класи для цього плагіна), `db` (використовується для встановлення прав доступу), `lang` (зберігаються переклади використаних текстів), `tests` (автотести для перевірки праці плагіна після змін як в ньому так і в системі), `ui` (це вже застаріла система, в ній використані `js` скрипти та `css` стилі), також набір базових файлів це – `index.php` (основний файл), `lib.php` (набір функцій для роботи), `settings.php` (зберігаються налаштування для адміністратора), `style.css` (базові таблиці стилів).

У системі Moodle дані зберігаються у файловій системі Moodle та у SQL базі даних. У файловій системі зберігаються завантажені викладачами матеріали до курсів та завантажені студентами файли у більшості випадків з результатами виконання завдань у різних форматах. Інформація про тестування, яке в оцінюється автоматично, налаштування, оцінки, чати, логи та інша інформація зберігається у базі даних. при розробці проєкту це була СУБД PostgreSQL (рис. 3.4).

На рисунку 3.4 в лівій частині бачимо список таблиць, яких в розглянутій версії 3.9.9 більш ніж чотири сотні, серед яких, наприклад, `mdl_config` – використовується для збереження налаштувань, а `mdl_course` – зберігає курси, в правій частині рисунку ми якраз і бачимо список курсів у цій таблиці. Є окремі таблиці для логів, статистики, користувачів, різноманітних модулів для навчання.

Для отримання даних, наприклад списку курсів, не обов'язково звертатися до БД з використанням SQL-запитів. В Moodle майже будь яку інформацію можна отримати за допомогою `core` класів з папки бібліотек `lib`, не взаємодіючи з базою даних. На рисунку 3.5 є такі файли як `coursecatlib.php` – відповідає за роботу з курсами та категоріями, `grouplib.php` – відповідає за роботу с групами, `outputlib.php` – відповідає за вивід інформації на екран, та багато інших. Кожен плагін також має файл `lib.php` в якому є функції та методи для його роботи, але їх може використовувати не тільки сам плагін, а і вкладені у нього плагіни. Наприклад плагін `grade` має методи для роботи з оцінками, тому його плагіни, які знаходяться рівнем нижче, наприклад `report`, можуть це використовувати і йому треба реалізувати менше функцій для його роботи.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of database objects, with 'mdl_course' selected. The main window shows a SQL query: `SELECT * FROM public.mdl_course ORDER BY id ASC LIMIT 100`. Below the query, a table of results is displayed with columns: id, category, sortorder, fullname, and shortname. The table contains 20 rows of course data.

id	category	sortorder	fullname	shortname
1	1	0	MOODLE3 ЧНУ імені Петра Могили	MOODLE3 ЧНУ
2	8614	1703	Курс для Sergio	Курс для S
3	8636	1703	Тестовий курс 2022	Тестовий к
4	8991	1849	[1201 · I семестр 2019/2020] Філософія	[1201 · I sei
5	8992	1849	[1201 · I семестр 2019/2020] Українська мова (за професійним спрямуванням)	[1201 · I sei
6	8993	1849	[1201 · I семестр 2019/2020] Медична хімія	[1201 · I sei
7	8994	1849	[1201 · I семестр 2019/2020] Медична та біологічна фізика	[1201 · I sei
8	8995	1849	[1201 · I семестр 2019/2020] Медична біологія	[1201 · I sei
9	8996	1849	[1201 · I семестр 2019/2020] Латинська мова та медична термінологія	[1201 · I sei
10	8997	1849	[1201 · I семестр 2019/2020] Історія України та української культури	[1201 · I sei
11	8998	1849	[1201 · I семестр 2019/2020] Іноземна мова	[1201 · I sei
12	8999	1849	[1201 · I семестр 2019/2020] Анатомія людини	[1201 · I sei
13	9000	2158	[2201 · III семестр 2019/2020] Фізіологія	[2201 · III ci
14	9001	2158	[2201 · III семестр 2019/2020] Охорона праці в галузі	[2201 · III ci
15	9002	2158	[2201 · III семестр 2019/2020] Європейський стандарт комп'ютерної грамотності	[2201 · III ci
16	9003	2158	[2201 · III семестр 2019/2020] Гістологія, цитологія та ембріологія	[2201 · III ci
17	9004	2158	[2201 · III семестр 2019/2020] Біологічна та біоорганічна хімія	[2201 · III ci
18	9005	2158	[2201 · III семестр 2019/2020] Англійська мова як іноземна	[2201 · III ci
19	9006	2158	[2201 · III семестр 2019/2020] Англійська мова (за професійним спрямуванням)	[2201 · III ci
20	9007	2158	[2201 · III семестр 2019/2020] Анатомія людини	[2201 · III ci

Рисунок 3.4 – Витяг інформації з БД PostgreSQL за допомогою pgAdmin 4

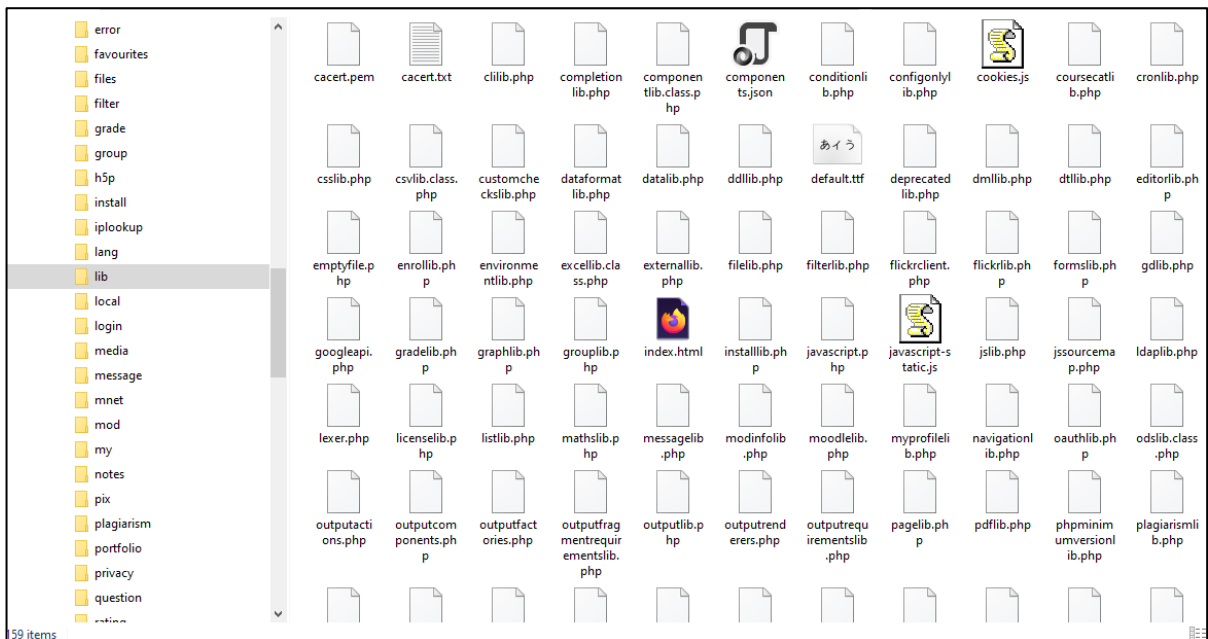


Рисунок 3.5 – Лістинг файлів папки lib

З точки зору розробника в Moodle необхідно враховувати *систему прав доступу*, яка дозволяє адміністраторові здійснювати контроль над тим, хто й що має право робити у системі. Вона є ускладненою. Користувачі можуть мати різні права в різних місцях, які називаються контекстами (англ. contexts). Наприклад, користувач може бути викладачем на одному курсі й студентом на іншому й, таким чином, мати різні права в кожному з місць.

Контексти Moodle формують ієрархію, дуже схожу на ієрархію директорій у файловій системі. При створенні плагіну ця ієрархія повинна бути врахована. На верхньому рівні перебуває контекст системи (англ. System context). У системному контексті перебуває деяка кількість контекстів для різних категорій, які створюються для організації курсів. Вони можуть бути вкладеними, тому категорія може містити інші категорії. Контексти категорій (англ. Category contexts) можуть містити контексти курсів (англ. Course contexts). Нарешті, кожна діяльність у рамках курсу має свій контекст модуля (англ. Module context) (рис. 3.6).

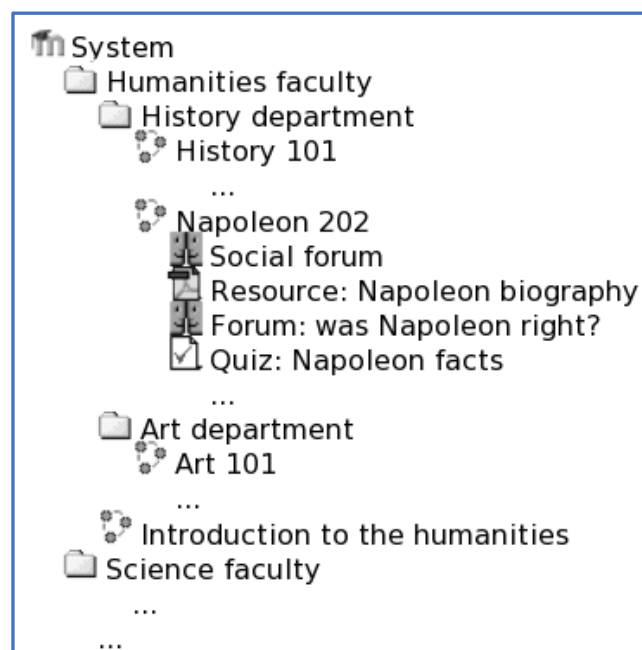


Рисунок 3.6 – Ієрархія контекстів Moodle

Система дозволів Moodle дозволяє адміністраторам здійснювати дуже гнучке керування системою. Адміністратор може:

- описувати бажані ролі з різними дозволами для кожної з можливостей;
- змінювати задані значення у рамках ролі в підконтекстах;
- ставити у відповідність користувачам у різних контекстах різні ролі.

3.2 Проєктування розробки та структури плагіну

Основні етапи розробки додаткового плагіна Moodle є наступними:

- вибір типу плагіна відповідно до поставленої задачі;
- проєктування інтерфейсу;
- планування архітектури коду, вивчення базового класу;
- заповнення базової структури модуля (`version.php`) найменування;
- вхід у «точку входу» – файл `php`, з іменем, що відповідає типу плагіна;
- програмування;
- тестування та налагоджування;
- впровадження розробленого плагіну в систему Moodle.

Першим кроком потрібно вирішити, яким буде доступ до плагіну у викладачів та керівників структурних підрозділів університету. При створенні плагіну для моніторингу навчального процесу в університеті плагін було вирішено розмістити у директорії Moodle `/grade/report/`. Оскільки створюється система аналізу даних з метою моніторингу навчального процесу, де головним фактором є оцінки, які відображаються у журналі оцінок (рис 3.6).

Журнал оцінок є у кожному курсі Moodle, він має дітей плагінів які відображають звіти по оцінкам, історії, результати окремих студентів, тому тут доцільно розмістити плагін, який видає звіти для моніторингу. Кожен, хто має права доступу до журналу оцінок, зможе використовувати плагін, а це викладачі, завідувачі кафедр, декани, проректори відповідно до їх ролей та прав доступу.

Програмна реалізація застосунку починається з проєктування інтуїтивно зрозумілого дружнього інтерфейсу користувача (user friendly). Спроєктуємо загальний вигляд інтерфейсу [30].

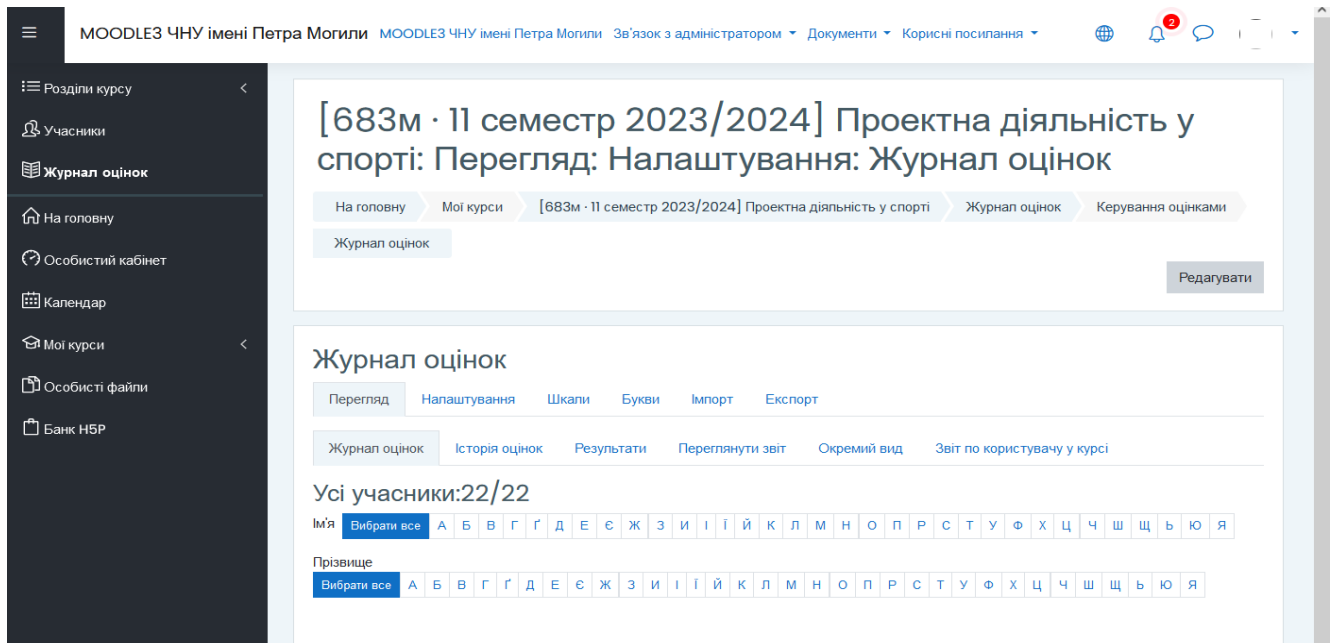


Рисунок 3.6 – Вигляд журналу оцінок

Щоб відібрати необхідні для аналізу дані користувач повинен виконати наступні кроки:

- обрати у ієрархії контекстів Moodle форму навчання (очну чи заочну) та певний семестр (семестри) з доступних користувачеві відповідно до його прав;
- обрати групу (групи) та курс (курси) у відібраних семестрах з доступних користувачеві відповідно до його прав;
- налаштувати групування по групам та семестрам для зручного перегляду зведеної інформації про хід навчання.

У журналі оцінок необхідно розмістити вкладку, при виборі якої відкриється сторінка, на якій користувач матиме можливість обрати вказані вище налаштування для відбору курсів. За результатами цих кроків для обраних курсів будуть відібрані необхідні для аналізу дані, які необхідно вивести у зручному для користувача зведеному вигляді.

Зведена інформація про успішність повинна бути виведена на наступній сторінці у вигляді таблиць, де інформація про наявні бали з кожного курсу буде виведена для кожної групи у вигляді однієї таблиці (при виборі групування за групами) або у вигляді сукупності таблиць для кожного курсу. Біля таблиць повинні бути в наявності засоби для експорту сформованих даних як для кожного курсу окремо, так і для усіх відібраних курсів разом у різних форматах: XLSX, DOCX, PDF, CSV, XML, TXT. На цій сторінці необхідно також передбачити перехід до наступної сторінки з візуальним виведенням інформації про успішність груп у кожному курсі у вигляді діаграм з відображенням оцінок у національній шкалі та шкалі ECTS.

Далі необхідно передбачити можливість переходу до сторінки з налаштуваннями кластерного аналізу та виведенням його результатів. Для усіх користувачів необхідно передбачити можливості по налаштуванню параметрів алгоритмів кластерного аналізу:

- вибір алгоритму кластеризації, який буде встановлено по замовчуванню із наступних: алгоритм ієрархічного кластерного аналізу, k-means і c-means;
- вибір формули, за якою буде розраховуватися близькість між об'єктами: (Euclidean, Canberra, Chebyshev, Manhattan, Cosine Similarity);
- для алгоритмів k-means та c-means: кількості кластерів, методу розрахунку початкових центрів кластерів, максимальну кількість епох (етапів ітерацій), мінімальну зміну при досягненні якої алгоритм вважається завершеним;
- для ієрархічного алгоритму: спосіб визначення відстані між кластерами при їх об'єднанні – найближчого сусіда, найдальшого сусіда та середня зважена.

Для адміністратора та досвідчених користувачів необхідно передбачити розширені можливості по налаштуванню параметрів алгоритмів кластерного аналізу, які передбачають виведення дебагу з додатковою інформацією по роботі алгоритму: значеннями нормалізованих ознак, координатами центроїдів, отриманими значеннями на кожному етапі ітерації.

На цій сторінці після вибору налаштувань необхідно розмістити кнопку для початку аналізу, за результатами якого на сторінці буде виведена інформація про проведений кластерний аналіз у вигляді таблиці з вказівкою кластеру, до якого попав кожен об'єкт (курс) та таблиці з середніми значеннями ознак кожного кластера.

Інформація про кластеризацію повинна бути також виведена у графічному вигляді. Для алгоритмів k-means і c-means доцільно реалізувати виведення графіку 3D із можливістю вибору осей, у яких будуть відображені утворені кластери. Оскільки аналіз проводиться у просторі 7 ознак. Для ієрархічного алгоритму необхідно передбачити виведення дендрограми.

Окремо для інтерфейсу необхідно розробити сторінку налаштувань для адміністратора, на якій він зможе вибрати алгоритм та його налаштування за замовчуванням, які і будуть використовуватися для всіх користувачів, а також вказати досвідчених користувачів, які мають змогу робити налаштування для себе самостійно.

3.3 Програмна реалізація

Вебзастосунок інтелектуальної системи було створено у середовищі розробки PhpStorm, рівень мови PHP використано 7.4 [31], тому що у такій версії працює в даний час Moodle на сервері в університеті, де плагін було впроваджено.

Назва створеного плагіну – `gradereport_viewsumminforesults`, повний шлях – `grade/report/viewsumminforesults`. Папка плагіну `viewsumminforesults` знаходиться у каталозі Moodle `grade/report/` і має наступну структуру папок та файлів (рис. 3.7). У файлі `index.php` основна логіка збереження даних та рух між сторінками. `Lib.php` має в собі весь набір функцій з маніпуляції та відображенням даних, ці функції викликаються з головного файлу. `Settings.php` – це налаштування адміністратора, `style.css` таблиці стилів, а `version.php` містить версію та назву плагіну.

Серед каталогів ми маємо `amd` – це модульне налаштування, завантаження `js` скриптів у форматі AMD [32].

Каталог db містить налаштування базового доступу до модуля, каталог js це каталог зі всіма js модулями та плагінами, каталог lang містить мовні файли, а каталог vendor додаткові PHP бібліотеки [33].

Name	Date modified	Type	Size
.idea	03.06.2024 21:24	File folder	
amd	29.05.2024 3:19	File folder	
db	15.05.2024 2:38	File folder	
js	01.06.2024 15:10	File folder	
lang	08.05.2024 3:56	File folder	
vendor	18.05.2024 4:28	File folder	
index.php	01.06.2024 1:49	PHP File	21 KB
lib.php	02.06.2024 21:16	PHP File	96 KB
settings.php	28.05.2024 20:22	PHP File	6 KB
styles.css	20.05.2024 2:00	Cascading Style S...	1 KB
version.php	02.06.2024 21:16	PHP File	2 KB

Рисунок 3.7 – Структура папок розробленого плагіну

Застосунок починає свою роботу зі списку курсів які доступні користувачу, тож використаємо core бібліотеку coursecatlib.php і викликаємо функцію get із номером категорії 0 (рис. 3.8), яка поверне базовий список категорій, в нашому випадку це навчальні роки. Наступним кроком будемо рекурсивно викликати до кожної категорії цю функцію для отримання категорій, а також функцію get_courses для отримання списку курсів в категорії (рис. 3.9).

```
function gradereport_viewsuminforesults_get_category_categories($categoryid = 0) {
    $listcategories = array();
    if ($categories = coursecat::get($categoryid)->get_children()) { // Print all the children recursively
        foreach ($categories as $cat) {
            $categoryinfo = gradereport_viewsuminforesults_get_category($cat);
            if (count($categoryinfo) == 0) {
                continue;
            }
            $listcategories[$cat->id] = $categoryinfo;
        }
    }
    return $listcategories;
}
```

Рисунок 3.8 – Виклик функції списку категорій

Функція `get_courses` має налаштування сортування, та які колонки нам треба, це задається за допомогою SQL мови [34]. Отримані курси перевіряємо чи є до них доступ у користувача за допомогою функції `has_capability`, це універсальна core функція яка перевіряє наявність різноманітних дозволів до різних модулів. За допомогою вищеназваних функцій ми формуємо список доступних користувачу курсів. Результат роботи ми зберігаємо користувачу у сесію (рис. 3.10).

```
function gradereport_viewsumminforesults_get_category_courses($category) {
    $listcourses = array();

    $courses = get_courses($category->id, 'c.sortorder ASC', 'c.id,c.sortorder,c.visible,c.fullname,c.shortname');

    if ($courses) {
        foreach ($courses as $course) {
            if (has_capability('gradereport/grader:view', context_course::instance($course->id))) {
                $listcourses[$course->id] = format_string($course->fullname);
            }
        }
    }

    return $listcourses;
}
```

Рисунок 3.9 – Виклик функції списку доступних курсів

```
if ($step == 1) {
    $SESSION->viewsumminforesults["categorytree"] = gradereport_viewsumminforesults_get_category_categories();
    $SESSION->viewsumminforesults["courses"] = array();
    $SESSION->viewsumminforesults["categories"] = array();
    gradereport_viewsumminforesults_list_courses_and_categories($SESSION->viewsumminforesults["categorytree"],
        $SESSION->viewsumminforesults["categories"], $SESSION->viewsumminforesults["courses"]);
    echo gradereport_viewsumminforesults_print_category_categories($SESSION->viewsumminforesults["categorytree"]);
}
```

Рисунок 3.10 – Збереження результату у сесію

Також у сесію зберігаємо подальшу обробку і списки тільки категорій і тільки курсів. Це дозволить по перше не завантажувати постійно списки при кожному кроці і при виборі курсів швидко перевірити чи є такі у списку.

За вибраними курсами треба сформувати список груп, тобто список груп студентів які є у вибраних курсах (рис. 3.11). Нажаль, Moodle не має вбудованої можливості отримання списку студентів за списком курсів, але це можна зробити

якщо зробити виклик у базу даних за допомогою функції `get_records_sql`, яка верне результат як список `stdClass` об'єктів [35]. Цей список обробляється додатковими функціями які формують список студентів і їх курсів, та список груп.

```
$sql = "SELECT je.enrolid,u.id,u.username,u.firstname,u.lastname,u.city,je.courseid FROM {user} u
JOIN (SELECT eu3_u.id,ej3_e.courseid,ej3_ue.id as enrolid FROM {user} eu3_u
JOIN {user_enrolments} ej3_ue ON ej3_ue.userid = eu3_u.id JOIN {enrol} ej3_e ON (ej3_e.id = ej3_ue.enrolid)
WHERE ej3_e.courseid in (".implode(", ", $SESSION->viewsuminforesults["selectedcourses"]).")
AND ej3_e.roleid=5 AND ej3_ue.status = 0 AND ej3_e.status = 0 AND eu3_u.deleted = 0) je ON je.id = u.id
AND u.deleted = 0 AND u.suspended = 0 ORDER BY u.lastname ASC, u.firstname ASC, u.idnumber, u.email";
$SESSION->viewsuminforesults["users"] = $DB->get_records_sql($sql);

$SESSION->viewsuminforesults["users"] = gradereport_viewsuminforesults_list_users($SESSION->viewsuminforesults["users"]);
$SESSION->viewsuminforesults["groups"] = gradereport_viewsuminforesults_list_groups($SESSION->viewsuminforesults["users"]);
echo gradereport_viewsuminforesults_print_groups($SESSION->viewsuminforesults["groups"]);
```

Рисунок 3.11 – Вибірка з бази даних списку студентів

Наступним кроком було створено сторінки з вибору налаштувань. На цьому закінчено перший крок в якому було зібрано дані від користувача з яких курсів, груп, та яким чином групувати отриману інформацію.

Переходимо до наступного кроку і відобразимо першу частину – таблиці даних із курсами, студентами та їх оцінками. Уже є інформація по студентам і курсам, але не має оцінок. Запросимо ці дані у бази даних (рис. 3.12) [36].

```
$sql = "SELECT g.id,gi.courseid,g.userid,g.finalgrade FROM {grade_items} gi, {grade_grades} g WHERE g.itemid = gi.id AND
gi.itemtype = 'course' AND gi.courseid in (".implode(", ", $SESSION->viewsuminforesults["selectedcourses"]).");
$grades = $DB->get_records_sql($sql);
```

Рисунок 3.12 – Сформований запит оцінок до бази даних

Дані отримуємо таким же чином, як і дані по студентам, і на основі цього формуємо таблиці з результатами. Але це не все, ще було заплановано додати експорт таблиць у файли різних форматів, такі як XLSX, DOC, PDF та інші. Для цього було використано плагін для jQuery, який вже вбудований в Moodle, `tableExport` (рис. 3.13).

```
function gradereport_viewsumminforesults_print_data($formatteddata) {
    $stdout = '';

    $iconxls = '';
    $iconword = '';
    $iconpdf = '';
    $iconcsv = '';
    $iconxml = '';
    $icontxt = '';

    $stdout .= '
<script type="text/javascript">
    function DoOnXlsxHyperlink($cell, row, col, href, content, hyperlink) {
        return content;
    }
</script>
';

    $stdout .= print_collapsible_region_start('', 'exportalltables', 'Експортувати все', '', true, true);
    $stdout .= '<a href="#" onClick="$($\'table[id]='\''tabletoexport_\'\'')\'.tableExport({type: \'excel\', tfootSelector: \'\'', mso: {fileFormat:
    xlsx: { onHyperlink: DoOnXlsxHyperlink, formatId: {numbers: 49}}}); return false;" title="Excel (XLSX)">'. $iconxls. '</a>';
    $stdout .= ' <a href="#" onClick="$($\'table[id]='\''tabletoexport_\'\'')\'.tableExport({type: \'doc\', tfootSelector: \'\'', mso: {pageOrientati
    $stdout .= ' <a href="#" onClick="$($\'table[id]='\''tabletoexport_\'\'')\'.tableExport({type: \'pdf\', tfootSelector: \'\'',
    pdfmake: {enabled:true, docDefinition: {pageOrientation: \'landscape\'}, widths: \'auto\'}); return false;" title="PDF">'. $iconpdf
    $stdout .= print_collapsible_region_end(true);
```

Рисунок 3.13 – Додавання кнопок експорту даних таблиць

На цьому кроці плагін просто не працював, jQuery успішно завантажувався, плагін також успішно додавався, але при натисканні на кнопку експорту цього плагіна вже не знаходило.

Треба зауважити що спочатку плагін завантажувався простим додаванням тегу *script src*, а jQuery завантажувався вбудованим викликом функції *\$PAGE->requires->jquery()*, але виявилось що цей підхід вже застарілий і jQuery вдруге завантажувався через систему модулів AMD і виконуване до цього завантаження затиралось. Тож було вирішено переробити всю роботу завантаження js скриптів за модульною системою AMD (рис. 3.14).

За цією системою створюємо файл зі списком залежностей, таких як jQuery та конфігом нашого модуля в якому вказані всі наші скрипти (рис. 3.15) [37].

У функції формування таблиць, після виведення всіх таблиць викликаємо модуль таким чином:

```
$PAGE->requires->js_call_amd('gradereport_viewsumminforesults/tableexport', 'init')
```

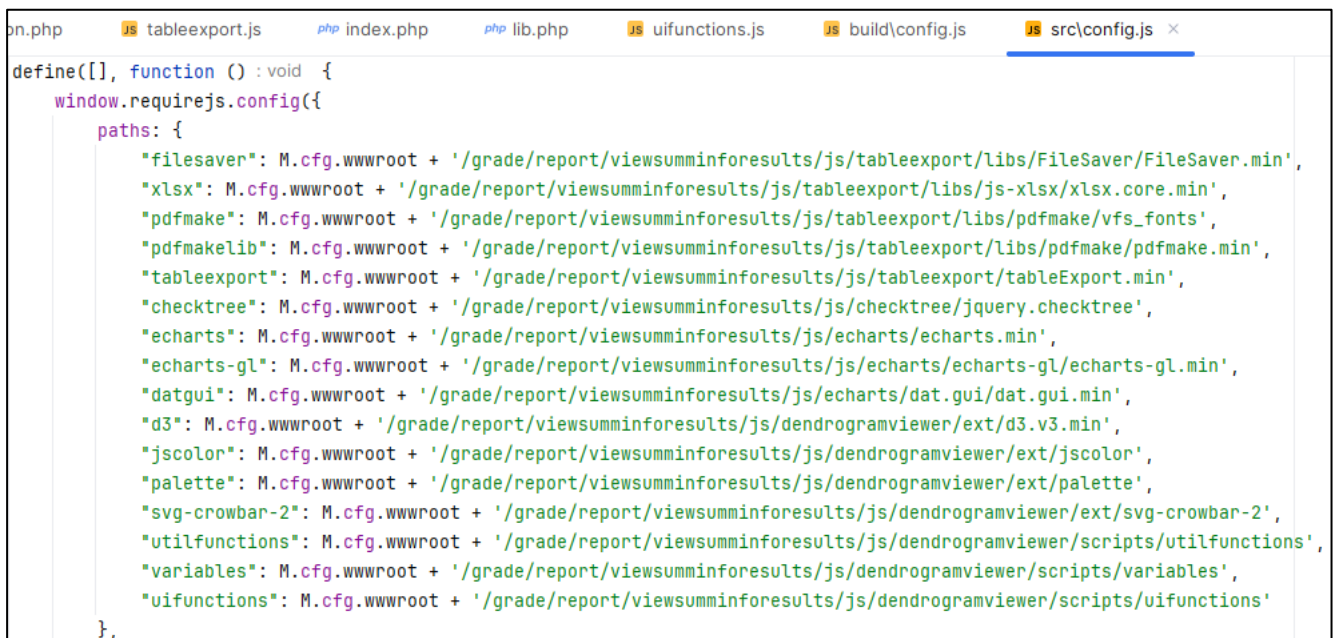
Завдяки цьому необхідні js скрипти завантажуються у тому порядку якому треба і все відмінно працює. З цієї сторінки маємо кнопки на результати графіків виведення зведеної інформації про успішність та результатів кластерного аналізу.



```

1  define(['jquery', 'jqueryui', 'gradereport_viewssuminforesults/config'], function ($, jquery) : {init: function(): void} {
2      Show usages
3      function initialize() : void {
4          require(['filesaver']);
5          require(['xlsx']);
6          require(['pdfmake/lib']);
7          require(['pdfmake']);
8          require(['tableexport']);
9      }
10
11     return {
12         init: function () : void {
13             initialize();
14         }
15     };
  
```

Рисунок 3.14 – Завантаження плагіна tableExport за системою AMD



```

define([], function () : void {
  window.requirejs.config({
    paths: {
      "filesaver": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/tableexport/libs/FileSaver/FileSaver.min',
      "xlsx": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/tableexport/libs/js-xlsx/xlsx.core.min',
      "pdfmake": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/tableexport/libs/pdfmake/vfs_fonts',
      "pdfmake/lib": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/tableexport/libs/pdfmake/pdfmake.min',
      "tableexport": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/tableexport/tableExport.min',
      "checktree": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/checktree/jquery.checktree',
      "echarts": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/echarts/echarts.min',
      "echarts-gl": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/echarts/echarts-gl/echarts-gl.min',
      "datgui": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/echarts/dat.gui/dat.gui.min',
      "d3": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/dendrogramviewer/ext/d3.v3.min',
      "jscolor": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/dendrogramviewer/ext/jscolor',
      "palette": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/dendrogramviewer/ext/palette',
      "svg-crowbar-2": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/dendrogramviewer/ext/svg-crowbar-2',
      "utilfunctions": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/dendrogramviewer/scripts/utilfunctions',
      "variables": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/dendrogramviewer/scripts/variables',
      "uifunctions": M.cfg.wwwroot + '/grade/report/viewssuminforesults/js/dendrogramviewer/scripts/uifunctions'
    }
  },
  
```

Рисунок 3.15 – Частина файлу налаштувань скриптів

Для діаграм успішності за національною системою ми маємо всі дані, тож просто формуємо графіки за допомогою js бібліотеки Apache Echarts (рис. 3.16).

На цьому кроці виникла проблема, яка полягала в тому, що для відображення кожного графіка потрібно створювати об'єкт *echarts* та задавати йому параметри, але ж завантаження js скриптів за системою AMD відбувається вже після завантаження всієї сторінки. Тому була використана можливість кастомних івентів за допомогою *dispatchEvent* після того, як було завантажено бібліотеку *echarts* [38]. А код створення об'єктів та задачі даних було обернуто в *addEventListener*, який слухав потрібний івент та запускав виконання як прийшов час [39].

```
function gradereport_viewsminforesults_print_course_graph($group, $courseid, $coursename, $caption, $grades) {
    "tooltip" => [{"trigger" => 'item', "formatter" => '{b}: {c} ({d}%)'},
    "legend" => [{"top" => 25, "orient" => 'horizontal', "left" => 'center'}, "series" => [{"name" => ' ', "type" => '
    "data" => [{"value" => 1, "name" => ' '}], "emphasis" => [{"itemStyle" => [{"shadowBlur" => 10, "shadowOffsetX"

    $char_key_exam = 'chart_exam_'.$courseid.'_'.$group;
    $char_key_test = 'chart_test_'.$courseid.'_'.$group;
    $stdout .= '<tr><td colspan="2">
        Відкорегуйте вид підсумкової атестації - залік чи іспит (у разі необхідності)<br>
        <button type="button" class="btn btn-primary" onClick="showTestExamGraph('.$char_key_test.', \''.$char_k
        <button type="button" class="btn btn-primary" onClick="showTestExamGraph('.$char_key_exam.', \''.$char_k
        <tr><td><div id=".$char_key_exam.'" style="height:450px; min-width:450px; display: block"></div>
        <div id=".$char_key_test.'" style="height:450px; min-width:450px; display: none"></div></td></tr>';
    $char_key_ects = 'chart_ects_'.$courseid.'_'.$group;
    $stdout .= '<td><div id=".$char_key_ects.'" style="height:450px; min-width:450px"></div></td></tr>';
    $stdout .= '</tbody></table>';
    $stdout .= '<script>
var '.$char_key_exam.';
var '.$char_key_test.';
var '.$char_key_ects.';
document.addEventListener("echartsbaseloaded", (e) => {
';

    $chart_option["title"]["text"] = $data_name["ukraine"];
    $chart_option["series"]["name"] = $data_name["ukraine"];
    $chart_option["series"]["data"] = array();
```

Рисунок 3.16 – Частина коду формування графіків

Для здійснення кластерного аналізу мало відібраної інформації про успішність, необхідно ще додати ознаки активності студентів та викладачів. Тож робимо додаткові запити до бази даних (рис. 3.17) [36].

```

$ssql = "SELECT id, courseid, roleid, stat1, stat2, timeend FROM {stats_monthly} WHERE stattype='activity'
        AND courseid in (*.implode(",", $SESSION->viewsuminforesults["selectedcourses"]).*) order by id";
$activity_monthly = $DB->get_records_sql($ssql);
$ssql = "SELECT id, courseid, roleid, stat1, stat2, timeend FROM {stats_daily} WHERE stattype='activity'
        AND courseid in (*.implode(",", $SESSION->viewsuminforesults["selectedcourses"]).*) order by id";
$activity_daily = $DB->get_records_sql($ssql);

$ssql = "SELECT g.id,gi.courseid,g.userid,g.finalgrade,gi.iteminstance FROM mdl_grade_items gi, mdl_grade_grades g WHERE g.itemid = gi.id
        AND g.timemodified IS NOT NULL AND gi.itemmodule in ('assign','lesson','quiz') AND gi.courseid in
        (*.implode(",", $SESSION->viewsuminforesults["selectedcourses"]).*) order by id";
$activity_task = $DB->get_records_sql($ssql);
  
```

Рисунок 3.17 – Запити до бази даних для отримання інформації про активність користувачів

Бачимо що активність було вибрано з таблиць *stats_monthly* та *stats_daily*, перша має інформацію тільки по минулим місяцям, а друга по дням, але записи старші ніж 3 місяці видаляються, тож для повної картини беремо місячні дані та додаємо дані поточного місяця з щоденної таблиці. Розраховуємо необхідні ознаки для кожного курсу (рис. 3.19).

```

$curitem["data1val"] = [$course["activities"][5]["view"], $course["studentcount"]];
$curitem["data1"] = round($course["activities"][5]["view"] / $course["studentcount"], 1);
$curitem["datacorrelation1"] = $curitem["data1"];

$curitem["data2val"] = [$course["activities"][5]["post"], $course["studentcount"]];
$curitem["data2"] = round($course["activities"][5]["post"] / $course["studentcount"], 1);
$curitem["datacorrelation2"] = $curitem["data2"];

$taskgradesum = 0;
foreach($course["taskgrade"][$group] as $iteminstancecount) {
    $taskgradesum += round($iteminstancecount / $students_in_group, 4);
}
$curitem["data3val"] = [$taskgradesum, $course["taskcount"]];
$curitem["data3"] = round($taskgradesum / $course["taskcount"], 4);
$curitem["datacorrelation3"] = $curitem["data3"];

if (!isset($course["activities"][3])) {
    $course["activities"][3] = array("view" => 0, "post" => 0);
}
$curitem["data5val"] = [$course["activities"][3]["view"], $course["activities"][3]["post"], $course["studentcount"]];
$curitem["data5"] = round(($course["activities"][3]["view"] + $course["activities"][3]["post"]) / $course["studentcount"], 1);
$curitem["datacorrelation5"] = $curitem["data5"];
  
```

Рисунок 3.19 – Розрахунок ознак активності користувачів на курсі

Розраховані дані представлені у різних числових діапазонах, тому для ознак, які не є відсотковими, робимо min-max нормалізацію (рис. 3.20).


```

function gradereport_viewsumminforesults_calc_min_max(&$data, $value) {
    if ($data["min"] > $value) {
        $data["min"] = $value;
    }
    if ($data["max"] < $value) {
        $data["max"] = $value;
    }
}
}
1 usage
function gradereport_viewsumminforesults_prepare_cluster_data_normalize(&$itemsforcluster) {
    $itemt = $itemsforcluster[key($itemsforcluster)];
    $data1 = ["min" => $itemt["data1"], "max" => $itemt["data1"]];
    $data2 = ["min" => $itemt["data2"], "max" => $itemt["data2"]];
    $data5 = ["min" => $itemt["data5"], "max" => $itemt["data5"]];
    $datacorrelation1 = ["min" => $itemt["datacorrelation1"], "max" => $itemt["datacorrelation1"]];
    $datacorrelation2 = ["min" => $itemt["datacorrelation2"], "max" => $itemt["datacorrelation2"]];
    $datacorrelation5 = ["min" => $itemt["datacorrelation5"], "max" => $itemt["datacorrelation5"]];

    foreach ($itemsforcluster as $itemkey => $item) {
        gradereport_viewsumminforesults_calc_min_max($data1, $item["data1"]);
        gradereport_viewsumminforesults_calc_min_max($data2, $item["data2"]);
        gradereport_viewsumminforesults_calc_min_max($data5, $item["data5"]);
        gradereport_viewsumminforesults_calc_min_max($datacorrelation1, $item["datacorrelation1"]);
        gradereport_viewsumminforesults_calc_min_max($datacorrelation2, $item["datacorrelation2"]);
        gradereport_viewsumminforesults_calc_min_max($datacorrelation5, $item["datacorrelation5"]);
    }
}

```

Рисунок 3.20 – Код нормалізації ознак

На цьому етапі виконано всі кроки для проведення кластеризації даних за обраним алгоритмом.

Тож виконуємо функцію кластеризації (додаток А), в яку передаємо нормалізовані дані, тип алгоритму, метод зв'язку, кількість кластерів, міру близькості, методи початкового прорахунку центроїдів, кількість епох або ітерацій, мінімальну зміну цільової функції для завершення алгоритму (рис. 3.21).

Також вказуємо, які ознаки було обрано для проведення аналізу. У результаті отримуємо список кластерів, список об'єктів у кожному кластері та список ознак кожного центроїда для алгоритмів k-means та c-means.

Кластеризацію було виконано за допомогою бібліотеки Rubix ML, але вона мала тільки k-means та c-means кластеризації, тож кластеризації ієрархічного аналізу було частково взято з іншої бібліотеки php-nlp-tools та адаптовано для роботи з Rubix ML [40].

```

switch($clusterconfig->clusteralgo) {
  case 'kmeans':
    $clusteralgo = new Rubix\ML\Clusterers\KMeans($clusterconfig->clusteramount, 128, $clusterconfig->maxepochs,
      $clusterconfig->minshift, 10, $distancealgo, $centroidselect);
    break;
  case 'cmeans':
    $clusteralgo = new Rubix\ML\Clusterers\FuzzyCMeans($clusterconfig->clusteramount, 1.2, $clusterconfig->maxepochs,
      $clusterconfig->minshift, $distancealgo, $centroidselect);
    break;
  case 'hierarchical':
    $clusteralgo = new Rubix\ML\Clusterers\Hierarchical($linkagemethod, $distancealgo);
    break;
  default:
    $clusteralgo = new Rubix\ML\Clusterers\KMeans($clusterconfig->clusteramount, 128, $clusterconfig->maxepochs,
      $clusterconfig->minshift, 10, $distancealgo, $centroidselect);
}

$clusteralgo->setLogger(new Rubix\ML\Loggers\Screen());
ob_start();
$clusteralgo->train($dataset);
$logdata = ob_get_contents();
ob_end_clean();

$dataset = new Rubix\ML\Datasets\Unlabeled($samples);
$predictions = $clusteralgo->predict($dataset);

$result["sizes"] = array_fill(0, count($clusteralgo->centroids()), 0);
if ($clusterconfig->clusteralgo == 'hierarchical') {

```

Рисунок 3.21 – Код реалізації алгоритму кластерного аналізу

З отриманих даних по кластеризації було прораховано середні значення ознак за кожним кластером і виведено у табличному вигляді. Також виведено трьох вимірний графік із можливістю вибору ознак або дендрограму в залежності від алгоритму кластеризації. Функції візуалізації результатів аналізу наведено у додатку Б.

Всі задані кроки розробки було опрацьовано, тож на цьому розробку плагіну було завершено. У процесі розробки здійснювалося тестування системи та корегування результатів роботи плагіну із виправленням виявлених помилок.

3.4 Інтерфейс системи аналізу та моніторингу навчального процесу в університеті

Для можливості користування плагіном викладачеві або керівнику відповідного структурного підрозділу університету необхідно перейти до

Кафедра інтелектуальних інформаційних систем
 Інтелектуальна система аналізу даних LMS–системи Moodle для моніторингу навчального процесу в університеті
 журналу оцінок у доступних для нього курсах та обрати вкладку «Звіт про успішність у відібраних курсах».

Буде відкрита ієрархія контекстів системи з доступними для користувача формами навчання, навчальними роками, семестрами, факультетами, спеціальностями кафедр та курсами (рис 3.22).

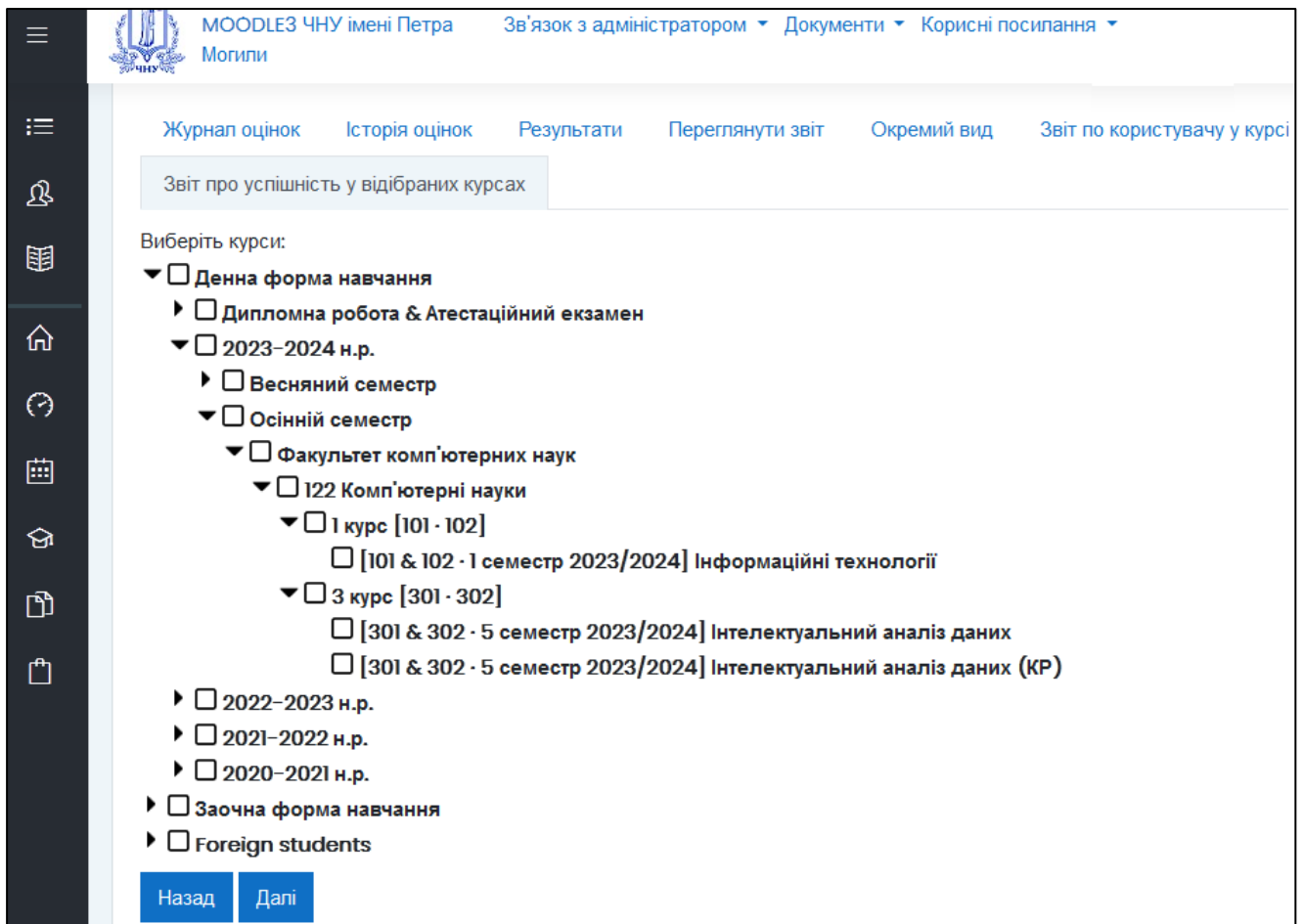


Рисунок 3.22 – Сторінка плагіну для вибору курсів

Після здійснення відбору необхідно натиснути кнопку «Далі» для переходу до наступної сторінки, на якій потрібно відібрати групи для аналізу і знову натиснути кнопку «Далі» для переходу до вибору способу групування (рис. 3.23).

The image shows two identical Moodle interface panels. Each panel has a title 'Звіт про успішність у відібраних курсах'. The left panel is for group selection, with a dropdown 'Виберіть групи:' and a list of checkboxes for 'Всі групи', '101', '102', '301', and '302'. The right panel is for grouping options, with a dropdown 'Виберіть налаштування:' and radio buttons for 'Не групувати', 'Групувати за групами', and 'Групувати за групами та семестрами'. Both panels have 'Назад' and 'Далі' buttons at the bottom.

Рисунок 3.23 – Сторінки вибору груп та типу групування

Натискаючи кнопку «Далі» на сторінці вибору способу групування здійснюється перехід до сторінки, яка містить таблиці з результатами успішності у відібраних курсах по кожній з відібраних груп (рис. 3.24).

The image shows a Moodle interface with a navigation bar at the top containing buttons: 'Назад', 'На початок', 'Відобразити графіки', and 'Зведена інформація про якість навчання'. Below the navigation bar is a link 'Експортувати все' with a dropdown arrow. Underneath are icons for various file formats. The main content is titled 'Група 301' and contains a table with student performance data.

#	Прізвище Ім'я	Ім'я входу	Група	[301 & 302 - 5 семестр 2023/2024] Інтелектуальний аналіз даних	[301 & 302 - 5 семестр 2023/2024] Інтелектуальний аналіз даних (КР)
1	Альсахбані Валід	21810326	301	0	80
2	Бакунов Віталій	22110101	301	8.9	65
3	Бойко Марія	22110103	301	60.4	91
4	Бугайова Юлія	22110104	301	94.8	100
5	Житомирська Анна	22110106	301	22.4	0

Рисунок 3.24 – Сторінка таблиць оцінок

Якщо було відібрано курси минулих семестрів, то буде картина, яка відображатиме наявні борги та успішність бал, отриманий кожним студентом групи з кожної дисципліни. Якщо відібрано дисципліни поточного семестру, то буде відображено поточний стан навчання з кожного курсу. Це дає можливість

оперативно у зручному вигляді мати інформацію про поточний стан навчання у відібраних курсах. Натиснувши список, що розгортається, «Експортувати все», буде надана можливість для експорту усіх таблиць. Також кожену таблицю можна експортувати окремо, натиснувши безпосередньо над таблицею кнопку з піктограмою, яка відповідає необхідному формату.

Натискаючи кнопку «Відобразити графіки», буде здійснено перехід до наступної сторінки, де для кожного курсу кожної обраної групи будуть відображені діаграми з успішністю у національній шкалі та шкалі ECTS (рис. 3.25).

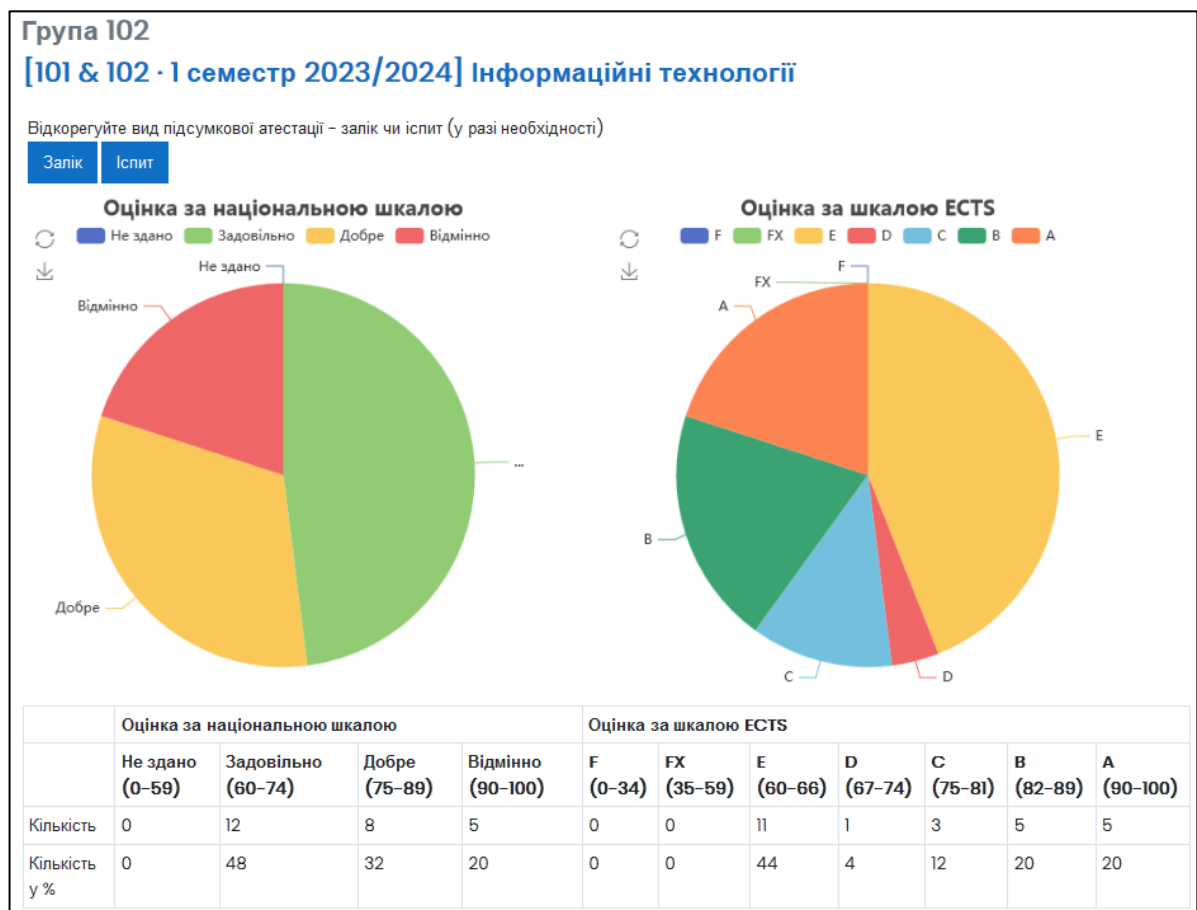


Рисунок 3.25 – Діаграми з успішністю у національній шкалі та шкалі ECTS

Необхідно відмітити, що у Moodle не фіксується вид підсумкового оцінювання з дисципліни – іспит чи залік. Тому на сторінці з діаграмами є можливість відкорегувати це, натиснувши кнопку «Залік». По замовчуванню

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система аналізу даних LMS–системи Moodle для моніторингу навчального процесу в університеті

діаграма відображає результат у національній шкалі для іспиту. Візуалізація результатів у вигляді діаграм є досить корисною для моніторингу стану навчання та полегшує роботу завідуючих кафедр, деканів, проректорів і викладачів.

Натискаючи кнопку «Зведений аналіз якості навчання у відібраних курсах», буде здійснено перехід до фінальної сторінки, де здійснюється налаштування та виведення результатів кластерного аналізу (рис. 3.26).

Зведений аналіз якості навчання у відібраних курсах

Назад
На початок
Відобразити графіки

Налаштування аналізу

	Активність студента				Активність викладача (кількість переглядів і завантажень на 1 студента)	Якість навчання	Успішність
	Середня кількість переглядів	Середня кількість завантажень	Кількість оцінених практичних завдань, %	Кількість студентів, які не здали іспит/залік			
Оберіть показники для проведення аналізу	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Додаткові налаштування аналізу](#) ▶

Провести аналіз

[Показники якості вивчення курсів](#) ▶

Результати проведеного аналізу

Середні значення характеристик курсів у виявлених групах – кластерах

Номер кластеру	Активність студента				Активність викладача (кількість переглядів і завантажень на 1 студента)	Якість навчання	Успішність	Кількість курсів
	Середня кількість переглядів	Середня кількість завантажень	Кількість оцінених практичних завдань, %	Кількість студентів, які не здали іспит/залік				
1	782.2	239.2	74.66%	15.74%	131.3	57.45%	67.87	4
2	85.6	11.1	40.4%	19.22%	18.6	92.74%	72.85	2

Рисунок 3.26 – Сторінка з налаштуванням та виведенням результатів кластерного аналізу

У верхній частині цієї сторінки міститься таблиця з ознаками, які можуть бути відібрані для аналізу. По замовчуванню відібрані усі ознаки. Але у разі потреби, можна частину ознак вилучити. По замовчуванню кластерний аналіз здійснюється за алгоритмом k-means та кількістю кластерів рівною 2. У нижній

Кафедра інтелектуальних інформаційних систем
Інтелектуальна система аналізу даних LMS–системи Moodle для моніторингу навчального процесу в університеті

частині цієї сторінки виведена таблиця, яка містить середні значення ознак кожного кластеру та кількість об’єктів, які до нього потрапили.

Натиснувши над цією таблицею кнопку «Показники якості вивчення курсів», на сторінці буде розкрита область із таблицею, яка показує значення ознак усіх відібраних курсів та номер кластеру, до якого курс потрапив (рис. 3.27).

Показники якості вивчення курсів ▾										
#	Номер кластеру	Дисципліна	Група	Активність студента				Активність викладача (кількість переглядів і завантажень на 1 студента)	Якість навчання	Успішність
				Середня кількість переглядів	Середня кількість завантажень	Кількість оцінених практичних завдань, %	Кількість студентів, які не здали іспит/залік			
1	1	[101 & 102 - 1 семестр 2023/2024] Інформаційні технології	102	877.9	284.6	83.2%	0%	143.7	52%	74.35
2	1	[101 & 102 - 1 семестр 2023/2024] Інформаційні технології	101	877.9	284.6	88.45%	3.7%	143.7	61.54%	77.57
3	1	[301 & 302 - 5 семестр 2023/2024] Інтелектуальний аналіз даних	301	686.5	193.8	62.28%	3.33%	118.9	56.25%	56.87
4	1	[301 & 302 - 5 семестр 2023/2024] Інтелектуальний аналіз даних	302	686.5	193.8	64.72%	5.93%	118.9	60%	62.69
5	2	[301 & 302 - 5 семестр 2023/2024] Інтелектуальний аналіз даних (КР)	301	85.6	11.1	83.75%	2.5%	18.6	90.48%	78.58
6	2	[301 & 302 - 5 семестр 2023/2024] Інтелектуальний аналіз даних (КР)	302	85.6	11.1	87.04%	5.93%	18.6	95%	67.11

Рисунок 3.27 – Таблиця показників якості вивчення курсів

У нижній частині сторінки під таблицею із середніми значеннями характеристик курсів у відібраних кластерах відображено графік результатів кластерного аналізу за алгоритмом по замовчуванню – k-means. Оскільки аналіз відбувається у просторі ознак, більшому за 3 (сім характеристик), у користувача є можливість для перегляду результатів обирати характеристики, у просторі яких буде відображено результат (рис. 3.28).

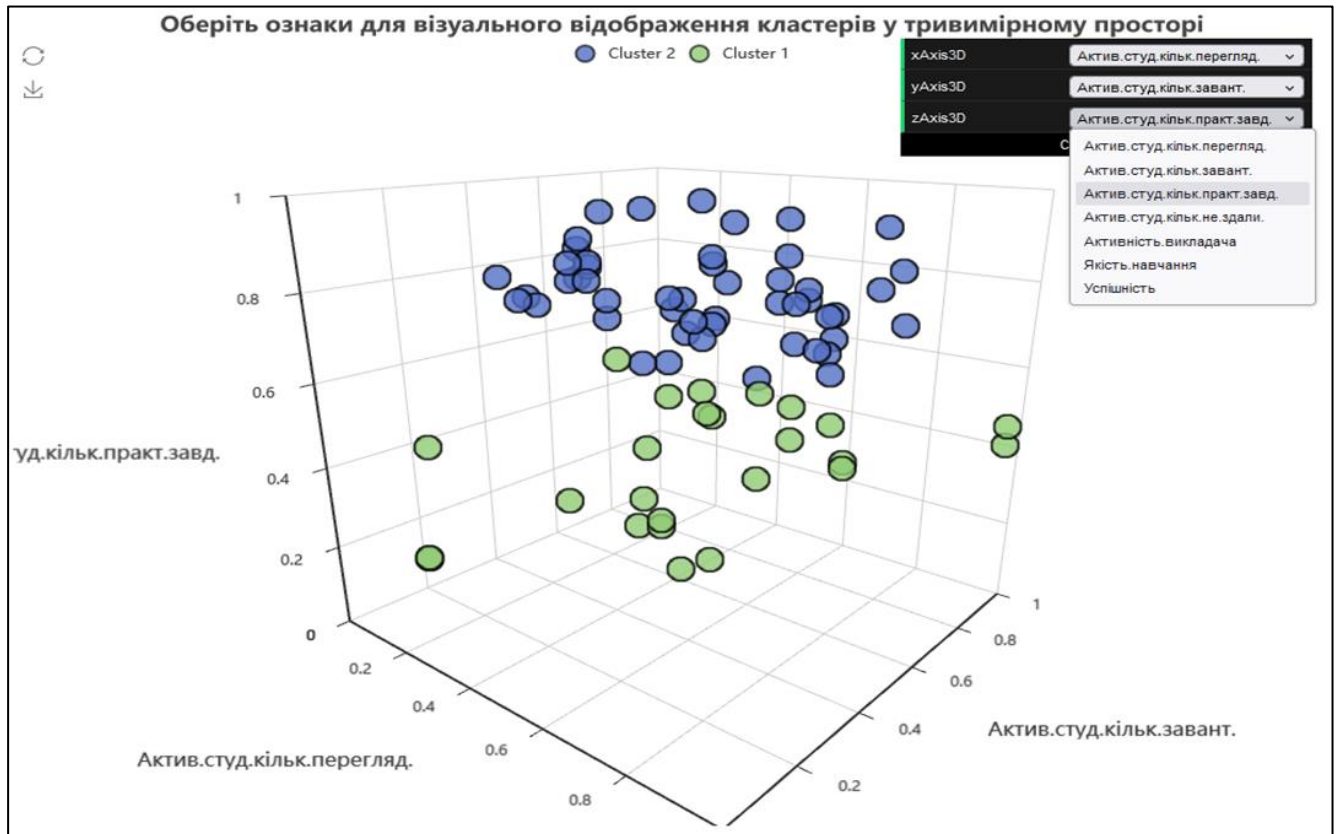


Рисунок 3.28 – Графічне відображення результатів аналізу за алгоритмами k-means та c-means

При здійсненні аналізу за алгоритмом c-means візуалізація результатів буде представлена у вигляді такого ж графіку. Якщо ознак буде обрано дві, то графік буде двовимірний. На рисунку 3.29 показано результати кластерного аналізу для тієї ж групи курсів, що й на рисунку 3.28, але у випадку вибору 3-х ознак та алгоритму ієрархічного кластерного аналізу. У цьому випадку буде виведена дендрограма з пропозицією визначити оптимальну кількість кластерів та перезапустити процедуру аналізу. По замовчуванню було 2 кластери, але проведений аналіз дозволив установити, що оптимальним є 3 кластери. Здійснивши аналіз характеристик курсів, які потрапили до кожного кластеру, ми можемо здійснити інтерпретацію результатів.

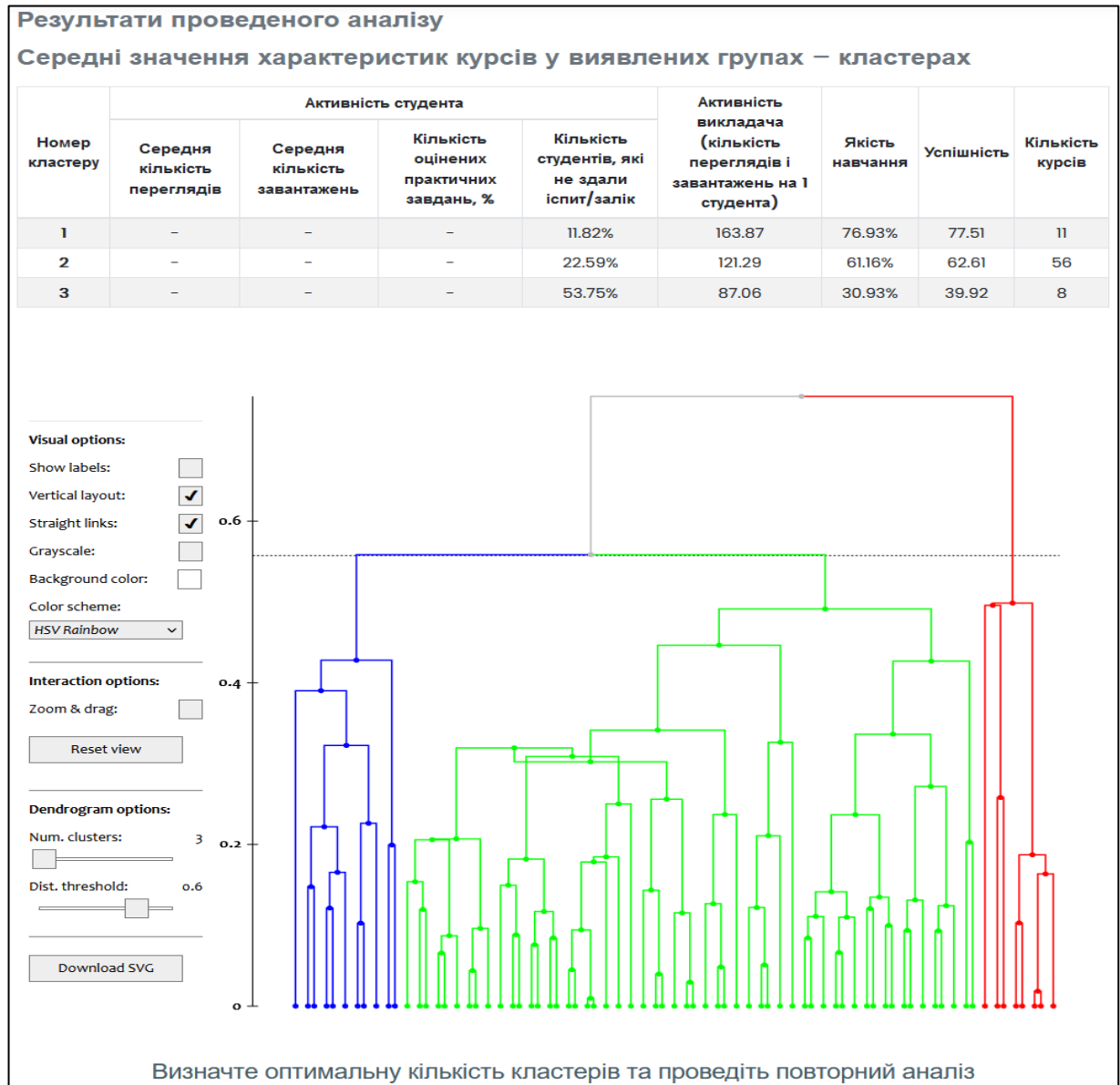


Рисунок 3.29 – Результат ієрархічного кластерного аналізу у вигляді дендрограми

Користувач має можливість змінити алгоритм для проведення кластерного аналізу та самостійно здійснити його налаштування. Для цього необхідно натиснути кнопку, яка розміщена у верхній частині сторінки, «Додаткові налаштування аналізу». На сторінці під кнопкою буде розкрита область для налаштувань (рис. 3.30) Досвідчений користувач буде мати більше можливостей для налаштування (рис. 3.31). Найбільше можливостей для налаштування кластерного аналізу має адміністратор (додаток В).

Додаткові налаштування аналізу ▾

Кількість кластерів

Провести аналіз

Рисунок 3.30 – Область додаткових налаштувань для звичайного користувача

Додаткові налаштування аналізу ▾

Алгоритм кластерного аналізу

Кількість кластерів

Метод зв'язку

Міра близькості

Розрахунок початкових центрів кластерів

Максимальна кількість епох

Мінімальна зміна при якій продовжується кластерування

Відобразити дебаг

Провести аналіз

Додаткові налаштування аналізу ▾

Алгоритм кластерного аналізу

Кількість кластерів

Метод зв'язку

Міра близькості

Розрахунок початкових центрів кластерів

Максимальна кількість епох

Мінімальна зміна при якій продовжується кластерування

Відобразити дебаг

Провести аналіз

а) алгоритм k-means

б) ієрархічний алгоритм

Рисунок 3.31 – Область додаткових налаштувань для адміністратора та досвідченого користувача

У адміністратора та досвідчених користувачів є можливість налаштувати виведення дебагу. Це дає можливість побачити розрахунки значень характеристик курсів, що було використано при тесуванні системи, нормалізовані значення ознак, інформацію про реалізований алгоритм та етапах ітерації та координати центроїдів (рис. 3.32 – рис. 3.35).

Показники якості вивчення курсів										
#	Номер кластеру	Дисципліна	Група	Активність студента				Активність викладача (кількість переглядів і завантажень на 1 студента)	Якість навчання	Успішніс
				Середня кількість переглядів	Середня кількість завантажень	Кількість оцінених практичних завдань, %	Кількість студентів, які не здали іспит/залік			
1	1	[101 & 102 - 1 семестр 2023/2024] Інформаційні технології	102	877.9 (45651 / 52)	284.6 (14797 / 52)	83.2% (2080 / 25)	0% (0 / 25)	143.7 ((4445 + 3029) / 52)	52% (13 / (25 - 0))	74.35 (1858.7 / 25)

Рисунок 3.32 – Відображення розрахунків значень характеристик курсів

#	Номер кластеру	Дисципліна	Група	Активність студента				Активність викладача (кількість переглядів і завантажень на 1 студента)	Якість навчання	Успішність
				Середня кількість переглядів	Середня кількість завантажень	Кількість оцінених практичних завдань, %	Кількість студентів, які не здали іспит/залік			
1	1	[101 & 102 - 1 семестр 2023/2024] Інформаційні технології	102	0.782	0.681	0.832	0	0.8274	0.52	0.7858

Рисунок 3.33 – Відображення нормалізованих значень характеристик курсів

Інформація про реалізований алгоритм										
[2024-06-06 21:21:40] INFO: Training K Means (k: 2, batch size: 128, epochs: 1000, min change: 1.0E-8, window: 10, kernel: Euclidean)										
[2024-06-06 21:21:40] INFO: Epoch: 1, Inertia: 0.45341130934752, Loss Change: 4INF										
[2024-06-06 21:21:40] INFO: Epoch: 2, Inertia: 0.44374171058919, Loss Change: 40.0096695987583323										
[2024-06-06 21:21:40] INFO: Epoch: 3, Inertia: 0.44166441880594, Loss Change: 40.0020772917832526										
[2024-06-06 21:21:40] INFO: Epoch: 4, Inertia: 0.44094759865236, Loss Change: 40.0007168201535761										
[2024-06-06 21:21:40] INFO: Epoch: 5, Inertia: 0.44064604110805, Loss Change: 40.00030155754430894										
[2024-06-06 21:21:40] INFO: Epoch: 6, Inertia: 0.44051442314239, Loss Change: 40.00013161796566707										
[2024-06-06 21:21:40] INFO: Epoch: 7, Inertia: 0.4404627580405, Loss Change: 45.1665101882592E-5										
[2024-06-06 21:21:40] INFO: Epoch: 8, Inertia: 0.44045088663798, Loss Change: 41.1871402518515E-5										
[2024-06-06 21:21:40] INFO: Epoch: 9, Inertia: 0.44045874259174, Loss Change: 47.8559537535772E-6										
[2024-06-06 21:21:40] INFO: Epoch: 10, Inertia: 0.44047567914188, Loss Change: 41.6936550140512E-5										
[2024-06-06 21:21:40] INFO: Epoch: 11, Inertia: 0.43968329913329, Loss Change: 40.00079238000859178										

Рисунок 3.34 – Інформація про реалізований алгоритм

Значення характеристик центроїдів у кластерах									
Номер кластеру	Активність студента				Активність викладача (кількість переглядів і завантажень на 1 студента)	Якість навчання	Успішність	Кількість курсів	
	Середня кількість переглядів	Середня кількість завантажень	Кількість оцінених практичних завдань, %	Кількість студентів, які не здали іспит/залік					
1	0.89	0.84	0.75	0.16	0.91	0.57	0.68	4	
2	0.1	0.04	0.4	0.19	0.13	0.93	0.73	2	

Рисунок 3.35 – Координати центроїдів

Розроблена система була протестована та впроваджена у систему Moodle ЧНУ ім. П. Могили для використання з метою аналізу та моніторингу успішності й якості навчання в університеті (додаток Г).

Висновки до розділу 3

У даному розділі було описано розробку, програмну реалізацію та впровадження системи аналізу даних LMS Moodle. Розроблений плагін дозволяє здійснювати відбір курсів та груп для аналізу, доступних користувачеві відповідно до ролей у системі. Для відібраних курсів плагін здійснює відбір даних для формування зведених звітів про успішність у табличному вигляді та у вигляді діаграм. У системі передбачено також здійснення кластерного аналізу для виявлення внутрішньої структури курсів у навчальному закладі стосовно якості організації навчального процесу з можливістю налаштувань алгоритмів та візуалізацією отриманих результатів. Застосування кластерного аналізу для обробки освітніх даних є підґрунтям при прийнятті освітніх рішень стосовно організації навчального процесу в університеті.

ВИСНОВКИ

Проведене дослідження дозволяє зробити наступні висновки.

Виявлено, що функції адміністрування LMS-системи Moodle охоплюють управління користувачами з ролями (викладач, студент, адміністратор тощо), які визначають права доступу до навчального контенту, організації студентів у певні групи, формування курсів, складання звітності. Основою навчального простору Moodle є дистанційні курси, призначені для вивчення навчальних дисциплін, які включають навчальні матеріали, викладачів і студентів із забезпеченням взаємодії між ними, контроль отримуваних знань і накопичення інформації про процес навчання. Moodle має засоби для додавання до курсу навчального матеріалу у будь-якому форматі, організації дистанційних занять, контролю та оцінювання результатів навчання. Платформа підходить для реалізації дистанційних, змішаних форм навчання й є додатком до традиційних очних форм навчання.

Установлено, що у навчальних закладах Moodle використовується як фреймворк для власних програмних рішень. Є безкоштовною Open Source-системою типу LMS, написана на PHP, використовує MySQL та Perl, працює з MySQL, PostgreSQL, Microsoft SQL Server або Oracle. Завдяки розвиненій модульній архітектурі, можливості Moodle можуть легко розширюватися сторонніми розроблювачами.

З'ясовано, що моніторинг є механізмом, який полягає у зборі й аналізі широкого спектра критеріїв і показників, аналіз яких дозволяє безперервно спостерігати за станом професійної підготовки фахівців у закладах освіти. LMS Moodle має інструменти моніторингу, які дозволяють аналізувати процеси на рівні окремого електронного курсу або системи в цілому: відслідковувати отримані студентами оцінки, завантажені та виконані завдання, активність студентів та викладачів на курсі. Однак зведену інформацію отримати не можливо у випадку, коли здійснюється аналіз роботи студентів сукупності курсів.

Здійснений аналіз додаткових плагінів, які можна встановити в Moodle (Grade Distribution, Multi Course Grader Report, Sitewide User Report, Rubrics Grading Report, Grade History), показав, що вони не надають можливості для отримання оперативних даних про результативність навчання у сукупності груп на рівні окремого викладача чи структурного підрозділу університету. Недостатньо розвинений функціонал для аналізу ходу навчального процесу не дає можливості завідувачу кафедри, декану факультету чи проректору оперативно отримувати звітну інформацію та проводити моніторинг. У системі також відсутні засоби для реалізації алгоритмів інтелектуального аналізу навчального контенту. Тому є потреба у розширенні функціональності системи.

Було проаналізовано можливі підходи до розширення можливостей Moodle: зміна ядра системи, інтеграція стороннього ПЗ, використання плагінів для моніторингу сторонніх розробників, написання власного плагіну. Та вирішено для розв'язання поставленої задачі зупинитися на розробці власного плагіну, адаптованого до потреб конкретного навчального закладу.

Для аналізу даних Moodle було обрано ієрархічний алгоритм кластерного аналізу та алгоритми k-means і c-means. Для аналізу даних Moodle було обрано ієрархічний алгоритм кластерного аналізу та алгоритми k-means і c-means. У якості об'єктів виступають курси, які викладаються у певному семестрі певного навчального року для кожної групи студентів з атрибутами: активність студентів при вивченні навчального матеріалу, активність студентів при виконанні поставлених завдань, активність викладача, кількість студентів, які не здали іспит/залік, якість навчання, успішність.

Для створення плагіну до LMS Moodle у якості основної мови програмування обрано PHP у середовищі розробки PhpStorm. Для додання інтерактиву та динамічності вебсторінок застосовано мову JavaScript. Оформлення та відображення даних для користувача відбувається за допомогою HTML та CSS. Для роботи з базою даних PostgreSQL було застосовано вбудовані засоби Moodle та прямі запити до БД на мові SQL. Візуалізація результатів

проведеного аналізу здійснювалася з використанням бібліотек Apache Echarts та DendrogramViewer. Для додавання 3D графіки застосовано плагін ECharts-GL, а для додавання до графіків панелі управління використано бібліотеку dat.GUI. Для експорту даних із результатами зведеного аналізу про якість навчання та хід навчального процесу використано плагін jQuery TableExport.

Здійснено розробку, програмну реалізацію та впровадження системи аналізу даних в LMS Moodle університету з метою проведення оперативного моніторингу якості навчання. Розроблений плагін дозволяє здійснювати відбір курсів та груп для аналізу, доступних користувачеві відповідно до ролей у системі. Для відібраних курсів плагін здійснює відбір даних для формування зведених звітів про успішність у табличному вигляді та у вигляді діаграм. У системі передбачено також здійснення кластерного аналізу для виявлення внутрішньої структури курсів у навчальному закладі стосовно якості організації навчального процесу з можливістю налаштувань алгоритмів та візуалізацією отриманих результатів. Застосування кластерного аналізу для обробки освітніх даних є підґрунтям при прийнятті освітніх рішень стосовно організації навчального процесу в університеті.

Поставлені завдання було виконано у повному обсязі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Moodle. Wikipedia: вебсайт. URL: <https://uk.wikipedia.org/wiki/Moodle> (дата звернення 09.05.2024).
2. Анисимов А.М. Работа в системе дистанционного обучения Moodle. Харьков, ХНАГХ, 2009. 292 с.
3. Ю.В. Триус, І.В. Герасименко, В.М. Франчук. Система електронного навчання ВНЗ на базі MOODLE: Методичний. Черкаси, 2012. 220 с.
4. Grade distribution. Moodle: вебсайт. URL: https://moodle.org/plugins/gradereport_gradedist (дата звернення 08.05.2024).
5. Multi Course Grader report. Moodle: вебсайт. URL: https://moodle.org/plugins/gradereport_multigrader (дата звернення 08.05.2024).
6. Sitewide User report. Moodle: вебсайт. URL: https://moodle.org/plugins/gradereport_sitewideuser (дата звернення 09.05.2024).
7. Rubrics Grading Report. Moodle: вебсайт. URL: https://moodle.org/plugins/gradereport_rubrics (дата звернення 09.05.2024).
8. Grade history. Moodle: вебсайт. URL: https://moodle.org/plugins/gradereport_history (дата звернення 10.05.2024).
9. Болюбаш Н. Інтелектуальний аналіз даних : навч. посіб. / Н. Болюбаш. – Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2023. – 320 с.
10. David Sculley. Proceedings of the 19th international conference on World wide web, 2010. 1178 p.
11. Joe H. Ward Jr. "Hierarchical Grouping to Optimize an Objective Function" Journal of the American Statistical Association, Volume 58, 1963. 244p.
12. Arthur, David & Vassilvitskii, Sergei. K-Means++: The Advantages of Careful Seeding. Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms, 2007.
13. J.C. Bezdek, R. Ehrlich and W. Full, "FCM: The fuzzy c-means clustering algorithm" Comput. Geosci., vol.10, May 1984, 203 p.

14. PhpStorm. Wikipedia: вебсайт. URL: <https://uk.wikipedia.org/wiki/PhpStorm> (дата звернення 25.05.2024).
15. PHP. Wikipedia: вебсайт. URL: <https://uk.wikipedia.org/wiki/PHP> (дата звернення 25.05.2024).
16. JavaScript. Wikipedia: вебсайт. URL: <https://uk.wikipedia.org/wiki/JavaScript> (дата звернення 25.05.2024).
17. HTML. Wikipedia: вебсайт. URL: <https://uk.wikipedia.org/wiki/HTML> (дата звернення 25.05.2024).
18. CSS. Wikipedia: вебсайт. URL: <https://uk.wikipedia.org/wiki/CSS> (дата звернення 25.05.2024).
19. SQL. Wikipedia: вебсайт. URL: <https://uk.wikipedia.org/wiki/SQL> (дата звернення 25.05.2024).
20. Rubix ML. Rubix ML: вебсайт. URL: <https://rubixml.com/> (дата звернення 25.05.2024).
21. jQuery. Wikipedia: вебсайт. URL: <https://uk.wikipedia.org/wiki/JQuery> (дата звернення 25.05.2024).
22. jQuery CheckTree. jQueryScript: вебсайт. URL: <https://www.jqueryscript.net/other/collapsible-checkable-hierarchical-tree.html> (дата звернення 25.05.2024).
23. Apache ECharts. Apache ECharts: вебсайт. URL: <https://echarts.apache.org/en/index.html> (дата звернення 25.05.2024).
24. ECharts-GL. Github: вебсайт. URL: <https://github.com/ecomfe/echarts-gl> (дата звернення 25.05.2024).
25. dat.GUI. Github: вебсайт. URL: <https://github.com/dataarts/dat.gui> (дата звернення 25.05.2024).
26. DendrogramViewer. Github: вебсайт. URL: <https://github.com/pedrodb/DendrogramViewer> (дата звернення 25.05.2024).
27. tableExport. Github: вебсайт. URL: <https://github.com/hhurz/tableExport.jquery.plugin> (дата звернення 25.05.2024).

28. Ian Wild. Moodle 3.x Developer's Guide. Packt Publishing, 2017. 368p.
29. Benjamin Ellis. Introduction to Moodle 3.9+ Plugin Development. Mukudu Publishing (Mukudu Ltd), 2021. 323 p.
30. Jon Duckett. HTML and CSS: Design and Build Websites. John Wiley & Sons, 2011. 490p.
31. David Sklar, Adam Trachtenberg. PHP Cookbook: Solutions & Examples for PHP Programmers. O'Reilly Media, 2014. 817 p.
32. Addy Osmani. Learning JavaScript Design Patterns. O'Reilly Media, 2012. 254 p.
33. MG Martin. PHP: The Complete Guide for Beginners, Intermediate and Advanced Detailed Approach To Master PHP Programming. Independently published, 2019. 238 p.
34. Anthony DeBarros. Practical SQL, 2nd Edition: A Beginner's Guide to Storytelling with Data. No Starch Press, 2022. 464 p.
35. Riaz Ahmed. Full Stack Web Development For Beginners: Learn Ecommerce Web Development Using HTML5, CSS3, Bootstrap, JavaScript, MySQL, and PHP. Independently published, 2021. 392 p.
36. Alan Beaulieu. Learning SQL: Generate, Manipulate, and Retrieve Data. O'Reilly Media, 2020. 377 p.
37. David Flanagan. Javascript: The Definitive Guide: Master the World's Most-Used Programming Language. O'Reilly Media, 2020. 704 p.
38. Wang Da Wei. ECharts data visualization: entry. actual combat and advanced. Machinery Industry Press, 2020. 208p.
39. Laurence Lars Svekis, Maaike Van Putten, Rob Percival. JavaScript from Beginner to Professional: Learn JavaScript quickly by building fun, interactive, and dynamic web apps, games, and pages. Packt Publishing, 2021. 546 p.
40. php-nlp-tools. Github: вебсайт. URL: <https://github.com/angeloskath/php-nlp-tools> (дата звернення 25.05.2024).

ДОДАТОК А

Функція реалізації алгоритмів кластеризації

```
function gradereport_viewsumminforesults_make_clusters($clusterconfig, &$itemsforcluster) {
    global $CFG;
    require_once($CFG->dirroot . '/grade/report/viewsumminforesults/vendor/autoload.php');

    $samples = array();
    $labels = array();
    foreach ($itemsforcluster as $itemkey => $item) {
        $itemdata = array();
        foreach ($clusterconfig->clusterdataselection as $dataid) {
            $itemdata[] = $item["normalizeddata".$dataid];
        }
        $samples[] = $itemdata;
        $labels[] = $itemkey;
    }

    // $dataset = new Rubix\ML\Datasets\Labeled($samples, $labels);
    $dataset = new Rubix\ML\Datasets\Unlabeled($samples);

    switch($clusterconfig->linkagemethod) {
        case 'single':
            $linkagemethod = new Rubix\ML\Clusterers\MergeStrategies\SingleLink();
            break;
        case 'complete':
            $linkagemethod = new Rubix\ML\Clusterers\MergeStrategies\CompleteLink();
            break;
        case 'selected':
            $linkagemethod = new Rubix\ML\Clusterers\MergeStrategies\GroupAverage();
            break;
        default:
            $linkagemethod = new Rubix\ML\Clusterers\MergeStrategies\GroupAverage();
    }

    switch($clusterconfig->distancealgo) {
        case 'euclidean':
            $distancealgo = new Rubix\ML\Kernels\Distance\Euclidean();
            break;
        case 'canberra':
            $distancealgo = new Rubix\ML\Kernels\Distance\Canberra();
            break;
        case 'diagonal':
            $distancealgo = new Rubix\ML\Kernels\Distance\Diagonal();
            break;
        case 'manhattan':
            $distancealgo = new Rubix\ML\Kernels\Distance\Manhattan();
            break;
        case 'cosine':
```



```

    $distancealgo = new Rubix\ML\Kernels\Distance\Cosine();
    break;
  default:
    $distancealgo = new Rubix\ML\Kernels\Distance\Euclidean();
  }

  switch($clusterconfig->centroidselect) {
    case 'random':
      $centroidselect = new Rubix\ML\Clusterers\Seeders\Random();
      break;
    case 'plusplus':
      $centroidselect = new Rubix\ML\Clusterers\Seeders\PlusPlus($distancealgo);
      break;
    default:
      $centroidselect = new Rubix\ML\Clusterers\Seeders\Random();
  }

  switch($clusterconfig->clusteralgo) {
    case 'kmeans':
      $clusteralgo = new Rubix\ML\Clusterers\KMeans($clusterconfig->clusteramount, 128,
      $clusterconfig->maxepochs,
      $clusterconfig->minshift, 10, $distancealgo, $centroidselect);
      break;
    case 'cmeans':
      $clusteralgo = new Rubix\ML\Clusterers\FuzzyCMeans($clusterconfig->clusteramount, 1.2,
      $clusterconfig->maxepochs,
      $clusterconfig->minshift, $distancealgo, $centroidselect);
      break;
    case 'hierarchical':
      $clusteralgo = new Rubix\ML\Clusterers\Hierarchical($linkagemethod, $distancealgo);
      break;
    default:
      $clusteralgo = new Rubix\ML\Clusterers\KMeans($clusterconfig->clusteramount, 128,
      $clusterconfig->maxepochs,
      $clusterconfig->minshift, 10, $distancealgo, $centroidselect);
  }

  $clusteralgo->setLogger(new Rubix\ML\Loggers\Screen());
  ob_start();
  $clusteralgo->train($dataset);
  $logdata = ob_get_contents();
  ob_end_clean();

  $dataset = new Rubix\ML\Datasets\Unlabeled($samples);
  $predictions = $clusteralgo->predict($dataset);

  $result["sizes"] = array_fill(0, count($clusteralgo->centroids()), 0);
  if ($clusterconfig->clusteralgo == 'hierarchical') {
    $result["hierarchical"] = $predictions;
    $result["centroids"] = $clusteralgo->centroids($clusterconfig->clusteramount);
  }

```

```

$result["sizes"] = array_fill(0, count($result["centroids"]), 0);
$i = 0;
$predictions = array();
foreach ($result["centroids"] as $centroid) {
    if (is_array($centroid)) {
        $items = explode(";", $centroid[3]);
        foreach ($items as $item) {
            $predictions[$item] = $i;
        }
    } else {
        $predictions[$centroid] = $i;
    }
    $i++;
}
$centroidcount = count($result["centroids"]);
$result["centroids"] = array();
for ($i = 0; $i < $centroidcount; $i++) {
    $result["centroids"][$i] = array_fill(0, count($clusterconfig->clusterdataselection), 0);
}

} else {
    $result["centroids"] = $clusteralgo->centroids();
}
$i = 0;
foreach ($itemsforcluster as $itemkey => $item) {
    $itemsforcluster[$itemkey]["cluster"] = $predictions[$i] + 1;
    $result["sizes"][$predictions[$i]]++;
    $i++;
}

$result["logdata"] = $logdata;
return $result;
}

```

ДОДАТОК Б

Функція візуалізації результатів аналізу

```
function gradereport_viewsumminforesults_print_cluster_graph($clusterconfig, $itemsforcluster,
$clusterresult) {
    global $PAGE;

    $stdout = "";

    $labels =
["name","Актив.студ.кільк.перегляд.,","Актив.студ.кільк.завант.,","Актив.студ.кільк.практ.завд.,"
"Актив.студ.кільк.не.здали.,","Активність.викладача","Якість.навчання","Успішність"];
    $selectedlabels = array("name");
    foreach ($clusterconfig->clusterdataselection as $dataid) {
        $selectedlabels[] = $labels[$dataid];
    }
    $samples = array();
    foreach ($itemsforcluster as $itemkey => $item) {
        $itemdata = array();
        $itemdata[] = $item["coursename"];
        foreach ($clusterconfig->clusterdataselection as $dataid) {
            $itemdata[] = $item["normalizeddata".$dataid];
        }
        $samples[$item["cluster"]][] = $itemdata;
    }

    if ($clusterconfig->clusteralgo != 'hierarchical') {
        $stdout .= '<div id="clusterGraphAndControl" style="position: relative;">
<div id="clusterGraphId" style="height:700px; min-width:700px"></div>
<div class="dg ac" style="position: absolute; right: 0px; top: 25px; z-index: 1000;"></div>
</div>';

        if (count($clusterconfig->clusterdataselection) > 2) {
            $stdout .= '
<script>
document.addEventListener("echarts3loaded", (e) => {
var clusterGraphApp = {};

var clusterGraphChartDom = document.getElementById("clusterGraphId");
var clusterGraphChart = echarts.init(clusterGraphChartDom);

var clusterGraphSchema = [];
    foreach ($selectedlabels as $lid => $lname) {
        $stdout .= '{ name: "' . $lname . '", index: ' . $lid . ' },';
    }
    $stdout .= '
];
var clusterGraphFieldIndices = clusterGraphSchema.reduce(function (obj, item) {
    obj[item.name] = item.index;
```

```

return obj;
}, {});
var clusterGraphFieldNames = clusterGraphSchema.map(function (item) {
  return item.name;
});
clusterGraphFieldNames = clusterGraphFieldNames.slice(1, clusterGraphFieldNames.length);
  foreach ($samples as $clusterid => $item) {
    $stdout .= 'var clusterGraphDataCluster' . $clusterid . ' = ' . json_encode($item) . ' ';
  }

  $stdout .= '
var clusterGraphConfig = (clusterGraphApp.config = {
  xAxis3D: "" . $selectedlabels[1] . "",
  yAxis3D: "" . $selectedlabels[2] . "",
  zAxis3D: "" . $selectedlabels[3] . "",
  onChange: function () {
    if (clusterGraphDataCluster1) {
      clusterGraphChart.setOption({
        xAxis3D: { name: clusterGraphConfig.xAxis3D },
        yAxis3D: { name: clusterGraphConfig.yAxis3D },
        zAxis3D: { name: clusterGraphConfig.zAxis3D },
        series: [;
          foreach ($samples as $clusterid => $item) {
            $stdout .= '{
              dimensions: [ clusterGraphConfig.xAxis3D, clusterGraphConfig.yAxis3D,
clusterGraphConfig.zAxis3D ],
              data: clusterGraphDataCluster' . $clusterid . '.map(function (item, idx) {
                return [
                  item[clusterGraphFieldIndices[clusterGraphConfig.xAxis3D]],
                  item[clusterGraphFieldIndices[clusterGraphConfig.yAxis3D]],
                  item[clusterGraphFieldIndices[clusterGraphConfig.zAxis3D]],
                  item[clusterGraphFieldIndices["name"]]
                ];
              })
            },';
          }
          $stdout .= '
        ]
      });
    }
  }
});
clusterGraphApp.configParameters = {};
["xAxis3D", "yAxis3D", "zAxis3D"].forEach(function (fieldName) {
  clusterGraphApp.configParameters[fieldName] = { options: clusterGraphFieldNames };
});

clusterGraphChart.setOption({
  title: { text: "Оберіть ознаки для візуального відображення кластерів у тривимірному просторі", left: "center" },

```

```

legend: { top: 25, orient: "horizontal", left: "center" },
tooltip: {},
toolbox: { left: "left", top: 25, orient: "vertical", feature: { restore: {}, saveAsImage: {} } },
xAxis3D: { name: clusterGraphConfig.xAxis3D, type: "value" },
yAxis3D: { name: clusterGraphConfig.yAxis3D, type: "value" },
zAxis3D: { name: clusterGraphConfig.zAxis3D, type: "value" },
grid3D: {},
series: [;
  foreach ($samples as $clusterid => $item) {
    $stdout .= '{
      name: "Cluster ' . $clusterid . '",
      type: "scatter3D",
      dimensions: [ clusterGraphConfig.xAxis3D, clusterGraphConfig.yAxis3D,
clusterGraphConfig.zAxis3D ],
      data: clusterGraphDataCluster' . $clusterid . '.map(function (item, idx) {
        return [
          item[clusterGraphFieldIndices[clusterGraphConfig.xAxis3D]],
          item[clusterGraphFieldIndices[clusterGraphConfig.yAxis3D]],
          item[clusterGraphFieldIndices[clusterGraphConfig.zAxis3D]],
          item[clusterGraphFieldIndices["name"]]
        ];
      }),
      symbolSize: 20,
      itemStyle: { borderWidth: 1, borderColor: "rgba(0,0,0,0.8)" },
      emphasis: { itemStyle: { color: "#fff" } }
    },';
  }

  $stdout .= '
});

var datgui = new dat.GUI({ autoPlace: false, width: 310});
var dom_dat = document.getElementsByClassName("dg ac")[0];
dom_dat.appendChild(datgui.domElement);
datgui.domElement.className += " a";

for (let key of Object.keys(clusterGraphApp.configParameters)) {
  const param = clusterGraphApp.configParameters[key];
  datgui.add(clusterGraphApp.config, key,
param.options).onChange(clusterGraphApp.config.onChange);
}
});
</script>
';
  $PAGE->requires->js_call_amd('gradereport_viewssumminforesults/echarts3d', 'init');
} else {
  $stdout .= '
<script>
document.addEventListener("echartsbaseloaded", (e) => {
var clusterGraphChartDom = document.getElementById("clusterGraphId");

```

```

var clusterGraphChart = echarts.init(clusterGraphChartDom);

    foreach ($samples as $clusterid => $item) {
        $stdout .= 'var clusterGraphDataCluster' . $clusterid . ' = ' . json_encode($item) . ' ';
    }

    $stdout .= '
clusterGraphChart.setOption({
    title: { text: "", left: "center" },
    legend: { top: 25, orient: "horizontal", left: "center" },
    toolbox: { left: "left", top: 25, orient: "vertical", feature: { restore: {}}, saveAsImage: {}, dataZoom:
{} } },
    tooltip: { position: "top", formatter: "{a}:<br>- {c}<br>- {b}" },
    xAxis: { name: "" . $selectedlabels[1] . "", type: "value" },
    yAxis: { name: "" . $selectedlabels[2] . "", type: "value" },
    series: [;
        foreach ($samples as $clusterid => $item) {
            $stdout .= '
            {
            name:"Cluster ' . $clusterid . '",
            symbolSize: 20,
            data: clusterGraphDataCluster' . $clusterid . '.map(function (item, idx) {
                return { name: item[0], value: [item[1], item[2]]};
            }
            },
            type: "scatter"
            },';
        }
        $stdout .= '
    ]
});
});
</script>
';
    }
} else {
    $clusterGraphData = array();
    gradereport_viewssumminforesults_calculate_hierarchical_graph_data($clusterGraphData,
$clusterresult["hierarchical"][0]);

    $stdout .= '<div id="clusterGraphAndControl" style="position: relative;">
<table id="clustergraph-ui-table" style="width: 1000px; height: 800px; margin-left: auto; margin-
right: auto;">
<tr>
<td id="options-column">
<hr class="options-box" />

<!-- **** Visual options **** -->
<p class="options-title">Visual options:</p>
<div class="checkbox-box">
    Show labels:

```

```

<div style="float: right">
  <input id="labels-chkbox" type="checkbox" unchecked onclick="onShowLabels()">
  <label for="labels-chkbox"></label>
</div>
</div>
<div class="chkbox-box">
  Vertical layout:
  <div style="float: right">
    <input id="vert-layout-chkbox" type="checkbox" checked onclick="onVertLayout()">
    <label for="vert-layout-chkbox"></label>
  </div>
</div>
<div class="chkbox-box">
  Straight links:
  <div style="float: right">
    <input id="straight-chkbox" type="checkbox" checked onclick="onStraightLinks()">
    <label for="straight-chkbox"></label>
  </div>
</div>
<div class="chkbox-box">
  Grayscale:
  <div style="float: right">
    <input id="grayscale-chkbox" type="checkbox" onclick="onGrayscale()">
    <label for="grayscale-chkbox"></label>
  </div>
</div>
<div class="chkbox-box">
  Background color:
  <div style="float: right">
    <input id="color-picker" type="hidden" onchange="onColorSchemeChanged()"
value="none">
    <input id="pick-color-btn" type="button" class="jscolor { valueElement: \'color-
picker\'}">
    <label for="pick-color-btn"></label>
  </div>
</div>
<div class="options-box">
  Color scheme:
  <select id="color-scheme-select" class="options-select"
onchange="onColorSchemeChanged();"></select>
</div>

<hr class="options-line"/>

<!-- **** Interaction options **** -->
<p class="options-title">Interaction options:</p>
<div class="chkbox-box">
  Zoom & drag:
  <div style="float: right">
    <input id="zoom-chkbox" type="checkbox" onclick="update()">

```

```

    <label for="zoom-chkbox"></label>
  </div>
</div>
  <input class="options-box" id="center-button" type="button" value="Reset view"
onclick="onResetDragZoom()"/>

  <hr class="options-line"/>

  <!-- **** Dendrogram options **** -->
  <p class="options-title">Dendrogram options:</p>
  <div class="options-box">
    Num. clusters:
    <div class="options-box-value" id="num-clusters-slider-value">0</div>
  </div>
  <input class="options-box" id="num-clusters-slider" type="range"
    oninput="onNumClustersChanged(this.value)"
  onchange="onNumClustersChanged(this.value)"/>

  <div class="options-box">
    Dist. threshold:
    <div class="options-box-value" id="threshold-slider-value">0</div>
  </div>
  <input class="options-box" id="threshold-slider" type="range"
    oninput="onThresholdChanged(this.value)"
  onchange="onThresholdChanged(this.value)"/>

  <!-- **** Buttons **** -->
  <hr class="options-line"/>
  <input class="options-box" id="save-button" type="button" value="Download SVG"
onclick="saveImage(\`innerSvg\`)/>

  </td>
  <td id="container-column">
    <div id="container"></div>
  </td>
</tr>
</table>
<h4 style="text-align: center;">Визначте оптимальну кількість кластерів та проведіть повторний
аналіз</h4>
</div>;

  $stdout .= '
<script>
document.addEventListener("dendrogramviewerloaded", (e) => {
  var clusterGraphData = '.json_encode($clusterGraphData).';
  //initializes all the D3 elements
  window.initUI();
  //const clusterJsonObj = JSON.parse(clusterGraphData);
  window.readData(clusterGraphData);
  document.querySelector("[id=num-clusters-slider]").value = '.$clusterconfig->clusteramount.';

```



```
document.querySelector("[id=num-clusters-slider]").dispatchEvent(new Event("input"))
});
</script>
    ;
    $PAGE->requires->js_call_amd('gradereport_viewsumminforesults/dendrogramviewer', 'init');
}

return $stdout;
}
```

ДОДАТОК В

Область налаштувань кластерного аналізу для адміністратора

MOODLE3 ЧНУ імені Петра Могили

Blocks editing on

Home
Site administration
Grades
Report settings
Summarized information of results

Summarized information of results

Алгоритм кластерного аналізу
gradereport_viewsumminforesults | clusteralgo

Кількість кластерів
gradereport_viewsumminforesults | clusteramount

Мінімальна кількість кластерів
gradereport_viewsumminforesults | minclusteramount

Максимальна кількість кластерів
gradereport_viewsumminforesults | maxclusteramount

Метод зв'язку
gradereport_viewsumminforesults | linkagemethod

Міра близькості
gradereport_viewsumminforesults | distancealgo

Розрахунок початкових центрів кластерів
gradereport_viewsumminforesults | centroidselect

Максимальна кількість епох
gradereport_viewsumminforesults | maxepochs

Точність кластерування
gradereport_viewsumminforesults | minshift

Показники для кластерування
gradereport_viewsumminforesults | clusterdataselection

Досвідчені викладачі
gradereport_viewsumminforesults | advancedusers

K-means Default: K-means

Яким алгоритмом буде виконуватись кластеризація

2 Default: 2

На яку кількість кластерів буде розбито дані

2 Default: 2

Яку мінімальну кількість кластерів можуть вибирати викладачі

10 Default: 10

Яку максимальну кількість кластерів можуть вибирати викладачі

Weighted Average Linkage Default: Weighted Average Linkage

Яким методом зв'язуємо кластери при ієрархічному кластерному аналізі

Euclidean Default: Euclidean

Якою метрикою будемо рахувати відстань

Random Default: Random

Яким чином будуть відбиратись початкові центроїди

1000 Default: 1000

Максимальна кількість епох після якої кластерування припиниться навіть якщо не завершиться

1e-8 Default: 1e-8

Мінімальна зміна при якій продовжується кластерування

Активність студента / Середня кількість переглядів

Активність студента / Середня кількість завантажень

Активність студента / Кількість оцінених практичних завдань

Активність студента / Кількість студентів, які не здали іспит/залік

Активність викладача / кількість переглядів і завантажень на 1 студента

Кількість оцінених практичних завдань

Кількість студентів, які не здали іспит/залік

Default: Активність студента / Середня кількість переглядів, Активність студента / Середня кількість завантажень, Активність студента / Кількість оцінених практичних завдань, Активність студента / Кількість студентів, які не здали іспит/залік, Активність викладача / кількість переглядів і завантажень на 1 студента, Кількість оцінених практичних завдань, Кількість студентів, які не здали іспит/залік

Оберіть показники для проведення аналізу

11043 Default: Empty

Список через кому айді юзерів які можуть смостійно обирати налаштування для себе

Save changes

ДОДАТОК Г

Акт про впровадження системи

№ _____ від 11.06. 2024 р.

АКТ

**про впровадження результатів наукового дослідження здобувача 4 курсу,
групи 401з спеціальності 122 «Комп'ютерні науки» Чорноморського
національного університету ім. П. Могили
Тараненко Олексія Юрійовича**

Цей акт засвідчує те, що у Чорноморському національному університеті ім. Петра Могили (на базі лабораторії сучасних інтелектуальних систем і технологій) було впроваджено інтелектуальну систему аналізу даних LMS Moodle для моніторингу навчального процесу:

- розроблено плагін із використанням алгоритмів Data Mining, який розширює функціональність системи Moodle шляхом надання можливості окремому викладачеві та працівникам структурних підрозділів університету (кафедри, деканату, навчально-методичного відділу) оперативно отримувати звітну інформацію та проводити моніторинг успішності й якості навчання з можливістю відбору сукупності курсів та груп відповідно до ролей користувача у системі;
- плагін впроваджено до системи Moodle, на базі якої здійснюється організація навчального процесу в університеті, розгорнуто необхідні елементи серверної частини, виконано налагодження та тестування системи.

Декан
Факультету комп'ютерних наук



Бойко А.П.