

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

До захисту допущено:

В.о. завідувача кафедри

Михайло НОВОТАРСЬКИЙ

(підпис)

“__” _____ 2025 р.

Дипломний проєкт
на здобуття ступеня бакалавра

за освітньо-професійною програмою “Комп’ютерні системи та мережі”
спеціальності 123 “Комп’ютерна інженерія”

на тему: Програмно-апаратна платформа з аналітичного скорингу освітньо-професійних програм та освітніх компонент

Виконав: студент IV курсу, групи Ю-13

(шифр групи)

Козаренко Денис Олегович

(прізвище, ім’я, по батькові)

(підпис)

Керівник ас. Туганських О. А.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант (нормоконтроль) Міщенко Л. Д.

(назва розділу)

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

(підпис)

Київ – 2025 р.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп’ютерні системи та мережі”

спеціальності 123 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Михайло НОВОТАРСЬКИЙ

(підпис)

“ ____ ” _____ 2025 р.

ЗАВДАННЯ

на бакалаврський дипломний проєкт студента

Козаренка Дениса Олеговича

1. Тема проєкту Програмно-апаратна платформа з аналітичного скорингу освітньо-професійних програм та освітніх компонент

керівник проєкту ас. Туганських О. А.

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 05 березня 2025 року №2101-с

2. Термін здачі студентом закінченого проєкту 09.06.2025

3. Вихідні дані до проєкту технічна документація, теоретичні дані.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються):

Розділ 1. Аналіз існуючих підходів до реалізації аналітичного скорингу освітньо-професійних програм та освітніх компонент

Розділ 2. Розробка апаратно-програмної платформи з аналітичного скорингу освітньо-професійних програм та освітніх компонент

Розділ 3. Прикладні аспекти застосування платформи з аналітичного скорингу освітньо-професійних програм та освітніх компонент

5. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень)

Структурна схема, функціональна схема та принципова схема

6. Консультанта проєкту, з вказівкою розділів проєкту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Міщенко Л. Д.		

7. Дата видачі завдання «19» січня 2025 р.

Календарний план

№ п/п	Найменування етапів дипломного проєкту	Терміни виконання етапів проєкту	Примітки
1.	<i>Затвердження теми проєкту</i>	06.02.2025	
2.	<i>Вивчення та аналіз завдання</i>	06.02.2025-21.02.2025	
3.	<i>Розробка архітектури та загальної структури системи</i>	22.02.2025-14.03.2025	
4.	<i>Розробка структур окремих підсистем</i>	15.03.2025-23.03.2025	
5.	<i>Програмна реалізація системи</i>	24.03.2025-24.04.2025	
6.	<i>Оформлення пояснювальної записки</i>	25.04.2025-25.05.2025	
7.	<i>Захист програмного продукту</i>	26.05.2025	
8.	<i>Передзахист</i>	26.05.2025	
9.	<i>Захист</i>	16.06.2025	

Студент-дипломник _____ Денис КОЗАРЕНКО
(підпис)

Керівник проєкту _____ Олександр ТУГАНСЬКИХ
(підпис)

АНОТАЦІЯ

У дипломній роботі розроблено аналітичну платформу для збору, обробки та візуалізації даних освітнього скорингу, яка дозволяє виявляти взаємозв'язки між рівнем задоволеності студентів навчальним процесом та їхньою успішністю. Реалізовано систему аналітичного скорингу на основі клієнт-серверної архітектури із застосуванням стеку технологій React та Node.js. Інтерфейс користувача підтримує зручне введення відповідей через динамічні форми, а адміністративна панель дозволяє генерувати звіти та графіки (на основі Chart.js). Система має модульну структуру, дотримано принципів чистого коду та забезпечено можливість подальшого масштабування. Проведено тестування на тестовому наборі даних, результати якого підтверджують працездатність і практичну цінність платформи для закладів освіти.

Ключові слова: скоринг, освітня аналітика, візуалізація, аналіз даних

ANNOTATION

In this project for a Bachelor's Degree, was developed an analytical platform for collecting, processing and visualising educational scoring data, which allows identifying the relationship between the level of student satisfaction students with the educational process and their academic performance. An analytical scoring system based on client-server architecture was implemented using React and Node.js technology stack. The user interface supports the convenient input of answers through dynamic forms, and the administrative panel allows generating reports and graphs (based on Chart.js). The system has a modular structure, adheres to the principles of clean code, and the possibility of further scaling is ensured. Testing was carried out on a test data set, the results of which confirm the operability and practical value of the platform for educational institutions.

Keywords: scoring, educational analytics, visualisation, data analys

ОПИС АЛЬБОМУ
ДО ДИПЛОМНОГО ПРОЄКТУ

на тему: «Програмно-апаратна платформа з аналітичного скорингу освітньо-професійних програм та освітніх компонент»

Київ – 2025

№ рядка	Формат	Значення	Найменування	Кіл. листів	№ екземпляра	Примітка	
			Документація загальна				
			Знову розроблена				
1	A4	ІАЛЦ.4672 00.002 ТЗ	Програмно-апаратна платформа з аналітичного скорингу освітньо-професійних програм та освітніх компонент	3			
			Технічне завдання				
2	A4	ІАЛЦ.4672 00.003 ПЗ	Програмно-апаратна платформа з аналітичного скорингу освітньо-професійних програм та освітніх компонент	56			
			Пояснювальна записка				
3	A4	ІАЛЦ.4672 00.004 ДА	Програмно-апаратна платформа з аналітичного скорингу освітньо-професійних програм та освітніх компонент	1			
			Структура системи				
4	A4	ІАЛЦ.4672 00.005 ДБ	Програмно-апаратна платформа з аналітичного скорингу освітньо-професійних програм та освітніх компонент	1			
			Принципова схема				
5	A4	ІАЛЦ.4672 00.006 ДВ	Програмно-апаратна платформа з аналітичного скорингу освітньо-професійних програм та освітніх компонент	1			
			Діаграма класів				
6	A4	ІАЛЦ.4672 00.007 ДГ	Програмно-апаратна платформа з аналітичного скорингу освітньо-професійних програм та освітніх компонент	22			
			Текст програми				
Зм	Лист	№ докум.	Підп	Дата	ІАЛЦ.467200.001 ОА Програмно-апаратна платформа з аналітичного скорингу освітньо-професійних програм та освітніх компонент Опис альбому		
Розроб		Козаренко Д. О.					
Перев		Туганських О. А.					
Н. Контр		Міщенко Л. Д.					
Затвердив							
					Літ.	Аркуш	Аркушів
						1	1
					НТУУ “КПІ” ФІОТ ІО-13		

ТЕХНІЧНЕ ЗАВДАННЯ
ДО ДИПЛОМНОГО ПРОЄКТУ

на тему: «Програмно-апаратна платформа з аналітичного скорингу освітньо-
професійних програм та освітніх компонент»

Київ – 2025

ЗМІСТ

ЗМІСТ	1
1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ.....	2
3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ	2
4. ДЖЕРЕЛА РОЗРОБКИ	2
5. ТЕХНІЧНІ ВИМОГИ	2
5.1. Вимоги до розробленого продукту.....	2
5.2. Вимоги до програмного забезпечення	3
5.3. Вимоги до апаратної частини	3
6. ЕТАПИ РОЗРОБКИ	3

					ІАЛЦ.467200.002 ТЗ		
		№ докум.	Підпис	Дата			
Розробив	Козаренко Д. О.				Програмно-апаратна платформа з аналітичного скорингу освітньо-	Літ.	Аркуш
Перевірив	Туганських О.А						Аркушів
							1
Н. Контр.						НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІО-13	
Затвердив							
							3

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Аналітична платформа освітнього скорингу — це програмна система для збору, обробки та аналізу зворотного зв'язку студентів з метою оцінювання якості навчального процесу. Платформа застосовується в закладах вищої освіти для підтримки прийняття рішень щодо підвищення ефективності навчання та задоволеності студентів.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки служить завдання на виконання розробки програмно-апаратної платформи з аналітичного скорингу освітньо-професійних програм та освітніх компонент, затвердженою кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний інститут»

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою розробки є підвищення оперативності, достовірності та повноти оцінювання освітніх програми на основі аналізу статистики студентських відгуків.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки даного дипломного проекту є офіційні документації, публікації та статті в мережі Інтернет на дану тему, науково-технічна література.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробленого продукту

Розроблена система має виконувати такі вимоги:

- Проведення аналітичного скорингу отриманих оцінок.
- Виведення графічних візуалізацій (графіки, діаграми, прогрес-бари).
- Кросбраузерна підтримка та адаптивний інтерфейс (використання Bootstrap).
- Модульність та легка підтримка коду (структура проекту, відповідність код-стандартам).

					ІАЛЦ.467200.002 ТЗ	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

5.2. Вимоги до програмного забезпечення

- ОС Windows, Mac чи Linux.
- Visual Studio 2017 IDE версії або вище.

5.3. Вимоги до апаратної частини

- ЦП не менше ніж Intel® Core (TM) i3-2100T.
- ROM не менше ніж 64 ГБ.
- RAM не менше ніж 4 ГБ.

6. ЕТАПИ РОЗРОБКИ

Назва етапів виконання	Термін виконання
Затвердження теми роботи	01.02.2025
Вивчення та аналіз завдання	01.02.2025-21.02.2025
Розробка архітектури та загальної структури системи	22.02.2025-14.03.2025
Розробка структур окремих частин системи	15.03.2025-23.03.2025
Програмна реалізація системи	24.03.2025-24.04.2025
Виправлення помилок	25.04.2025-25.05.2025
Оформлення пояснювальної записки	26.05.2025

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОГО ПРОЄКТУ**

на тему: «Програмно-апаратна платформа з аналітичного скорингу освітньо-професійних програм та освітніх компонент»

Київ – 2025

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО РЕАЛІЗАЦІЇ АНАЛІТИЧНОГО СКОРИНГУ ОСВІТНЬО-ПРОФЕСІЙНИХ ПРОГРАМ ТА ОСВІТНІХ КОМПОНЕНТ.....	6
1.1 Аналіз існуючих технологічних рішень з реалізації аналітичного скорингу освітньо-професійних програм.....	6
1.2 Аналіз існуючих методологій до скорингу освітньо-професійних програм та компонент	11
1.3. Формалізація задачі та постановка часткових задач проєктування....	14
ВИСНОВКИ ДО РОЗДІЛУ 1.....	17
РОЗДІЛ 2 РОЗРОБКА АПАРАТНО-ПРОГРАМНОЇ ПЛАТФОРМИ З АНАЛІТИЧНОГО СКОРИНГУ ОСВІТНЬО-ПРОФЕСІЙНИХ ПРОГРАМ ОСВІТНІХ КОМПОНЕНТ.....	19
2.1 Розробка структури апаратної компоненти платформи з аналітичного скорингу освітньо-професійних програм.....	19
2.2 Розробка програмної компоненти платформи з аналітичного скорингу освітньо-професійних програм.....	21
2.2.1 Математична модель скорингу освітньо-професійних програм.....	21
2.2.2 Інженерія вимог	22
2.2.3 Архітектурне проєктування програмної компоненти	25
2.2.4 Розробка програмної компоненти платформи з скорингу освітньо- професійних програм	27
ВИСНОВКИ ДО РОЗДІЛУ 2.....	30
РОЗДІЛ 3 ПРИКЛАДНІ АСПЕКТИ ЗАСТОСУВАННЯ ПЛАТФОРМИ З АНАЛІТИЧНОГО СКОРИНГУ ОСВІТНЬО-ПРОФЕСІЙНИХ ПРОГРАМ ТА ОСВІТНІХ КОМПОНЕНТ.....	31

					ІАЛЦ.467200.003 ПЗ		
		№ докум.	Підпис	Дата			
Розробив	Козаренко Д. О.				Програмно-апаратна платформа з аналітичного скорингу освітньо-	Літ.	Аркуш
Перевірив	Туганських О.А						1
						Аркушів	55
Н. Контр.	Міщенко Л.Р.					НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІО-13	
Затвердив							

3.1 Демонстрація практичних можливостей розробленої системи	31
3.2 Результати тестування розробленої системи	47
3.3 Програмна документація користувача	48
ВИСНОВКИ ДО РОЗДІЛУ 3.....	52
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55

					ІАЛЦ.467200.003 ПЗ	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК СКОРОЧЕНЬ

LMS – Learning Management System

MUSA – Multicriteria Satisfaction Analysis

API – Application Programming Interface

LDA – Latent Dirichlet Allocation

NLP – Natural Language Processing

					ІАЛЦ.467200.003 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У процесі модернізації вищої освіти важливим завданням стає оцінювання якості освітніх програм і дисциплін. Одним з основних джерел інформації для вдосконалення навчального процесу є відгуки студентів і випускників. В університетах дедалі активніше впроваджуються внутрішні системи моніторингу якості, що базуються на регулярних опитуваннях студентів щодо змісту курсів, методики викладання та загальної задоволеності навчанням. Проте традиційні підходи до збору й аналізу таких даних — паперові анкети чи ручне опрацювання — є малоефективними через складність в обробці великого обсягу інформації та ризики втрати важливих нюансів.

Актуальність теми зумовлена збільшенням популярності в останні роки віддаленої (асинхронної) форми навчання, особливо в умовах воєнного стану. Така тенденція створює необхідність в підвищенні оперативності, достовірності та повноти оцінювання освітніх програм. Відгуки студентів є одним із основних джерел оцінювання якості, тому існує попит на нові платформи для їх збору та аналізу. Таким чином, розробка інтегрованої програмно-апаратної платформи для аналітичного скорингу освітніх програм є актуальним завданням, яке відповідає сучасним тенденціям та світовим викликам.

Для вирішення цієї задачі доцільно використовувати автоматизовану платформу, яка забезпечує швидкий збір, аналіз і візуалізацію зворотного зв'язку. Така система дає змогу не лише зекономити час, а й підвищити якість управлінських рішень у сфері освітнього планування. У цьому контексті запропоновано концепцію аналітичного скорингу, що передбачає кількісну багатокритеріальну оцінку ефективності освітніх програм на основі даних, отриманих з відгуків: ступінь задоволеності студентів, релевантність і складність курсів, вплив на успішність тощо.

					ІАЛЦ.467200.003 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

Цей підхід дозволяє виявити сильні й слабкі сторони програм, простежити причинно-наслідкові зв'язки між характеристиками дисциплін і результатами навчання, що особливо важливо в умовах реформування системи вищої освіти. Робота має на меті створити систему, здатну автоматично оцінювати якість освітніх програм на основі аналізу відгуків, надаючи керівництву університету достовірну й оперативну інформацію.

Мета дипломного проєкту – підвищення оперативності, достовірності та повноти оцінювання освітніх програм на основі аналізу статистики студентських відгуків. Іншими словами, робота спрямована на створення системи, що забезпечить швидкий та науково обґрунтований зворотний зв'язок про ефективність освітньо-професійних програм і окремих курсів.

Об'єкт дипломного проєкту – процес багатокритеріального скорингу (оцінювання) освітньо-професійних програм та освітніх компонентів (дисциплін).

Предмет дипломного проєкту – методи багатofакторного аналізу (скорингу) ефективності освітніх програм та способи представлення характеристик освітніх програм на основі зібраних відгуків.

					ІАЛЦ.467200.003 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1

АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО РЕАЛІЗАЦІЇ АНАЛІТИЧНОГО СКОРИНГУ ОСВІТНЬО-ПРОФЕСІЙНИХ ПРОГРАМ ТА ОСВІТНІХ КОМПОНЕНТ

1.1 Аналіз існуючих технологічних рішень з реалізації аналітичного скорингу освітньо-професійних програм

Для побудови власної платформи аналітичного скорингу необхідно вивчити наявні на сьогодні технологічні рішення та системи, що забезпечують збір та аналіз відгуків про освітні програми. У світовій практиці переважна більшість університетів регулярно проводять оцінювання курсів та викладання через опитування студентів (Student Course Evaluations). Такі опитування зазвичай містять ряд запитань, на які студенти відповідають за шкалою Лікерта (наприклад, від 1 до 5 – від «цілком незгоден» до «цілком згоден») або вибирають один з варіантів, що найбільш відповідає їх думці про певний аспект курсу. Наприклад, поширеними є твердження на кшталт: «Викладач чітко пояснював матеріал», «Курс був корисним і сучасним», «Матеріали курсу були добре організовані» – студент оцінює свій рівень згоди із цими твердженнями. Також часто включають загальну оцінку курсу та викладання, а інколи – відкрите запитання для коментарів. Результати таких опитувань застосовуються для оцінки ефективності викладачів і програм, планування змін у навчальних курсах, прийняття рішень щодо підвищення кваліфікації кадрів тощо. Дослідження показують, що анкетування студентів є найбільш розповсюдженим способом отримання зворотного зв'язку щодо якості викладання та дизайну курсів в закладах вищої освіти [6]. Іншими словами, студентські рейтинги викладання фактично слугують основним інструментом вимірювання якості з погляду здобувачів освіти.

З появою інформаційних технологій процес збору та аналізу відгуків студентів перемістився в електронне середовище. Нині існує чимало

					ІАЛЦ.467200.003 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

спеціалізованих програмних продуктів і онлайн-платформ для проведення курс-оцінювальних опитувань. Деякі університети інтегрують опитування у свої системи керування навчанням (LMS, напр. Moodle, Canvas), інші використовують комерційні програмні комплекси для оцінювання курсів, такі як Watermark / EvaluationKIT, Explorance Blue, Qualtrics XM тощо. Подібні рішення дозволяють гнучко створювати опитувальники, розсилати їх студентам онлайн, автоматично агрегувати відповіді та формувати звіти з ключовими показниками. За допомогою таких систем адміністрація отримує «actionable insights», тобто корисні висновки для покращення навчального процесу.



Рисунок 1.1 – Explorance Blue

Наприклад, платформа Explorance Blue позиціонується як інструмент для «безперешкодного збирання та аналізу студентського зворотного зв'язку з метою покращення викладання та результатів навчання». Сучасні комерційні рішення зазвичай мають веб-інтерфейс адміністратора для керування опитуваннями і базою запитань, модуль для студентів (доступ до анкет, часто з

					ІАЛЦ.467200.003 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

мобільних пристроїв), та аналітичний модуль перегляду результатів (інтерактивні графіки, таблиці).

Окрім комерційних продуктів, існують також open-source (відкриті) системи оцінювання курсів, розроблені університетами під власні потреби. Наприклад, Hasso Plattner Institute (Німеччина) створив внутрішню веб-систему EvaP (Evaluation Platform) для опитування студентів та оцінки курсів [3]. Ця система, написана на Python / Django, надає зручні інтерфейси для студентів (заповнення анкет) і викладачів / адміністраторів (перегляд узагальнених результатів).

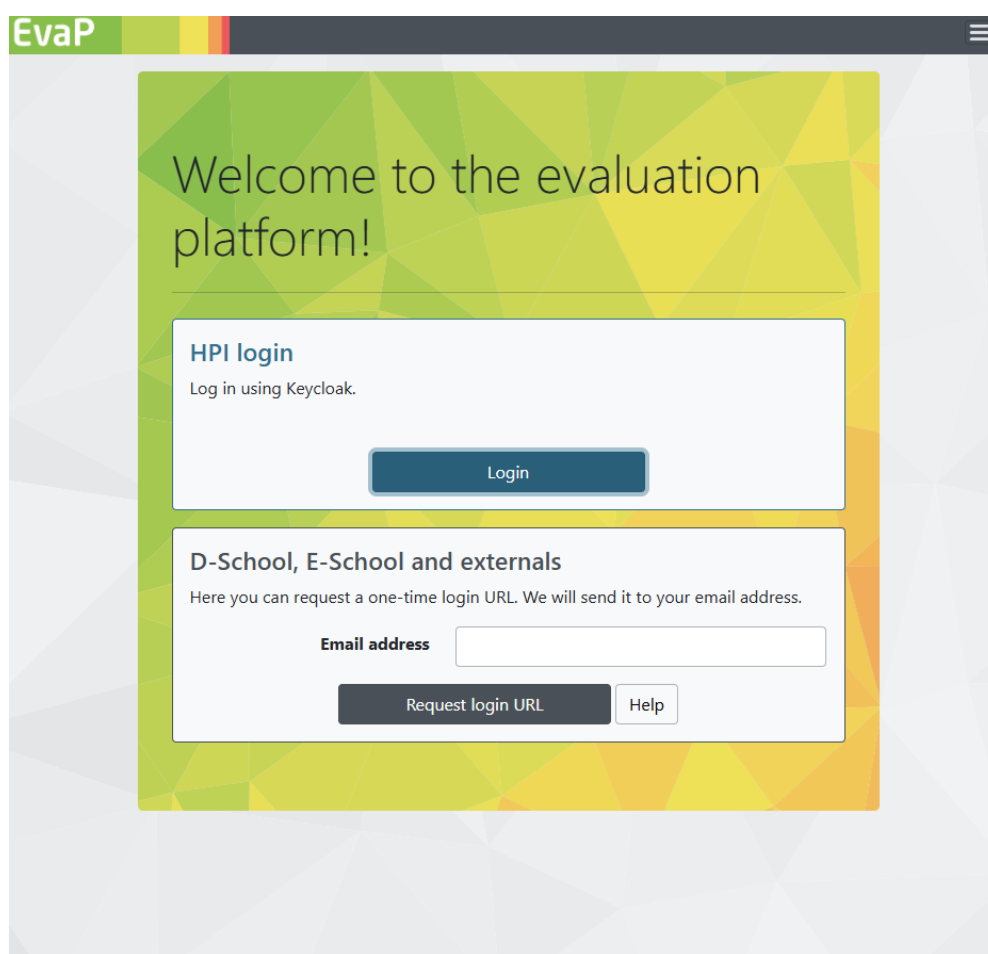


Рисунок 1.2 – EvaP (Evaluation Platform)

EvaP є сучасним та зрозумілим рішенням. Воно поєднує сучасний підхід до дизайну та інтуїтивно зрозумілий інтерфейс, що робить його зручним для кінцевих користувачів — як студентів, так і викладачів або адміністрації.

Система спеціалізується на організації процесів анкетування, збору відгуків та оцінювання якості освітніх курсів. Вона дозволяє автоматизувати та стандартизувати ці процеси, які зазвичай є складними, тривалими й потребують значних зусиль при ручній реалізації. Відкритий вихідний код таких проєктів доступний спільноті, що свідчить про активний інтерес академічних установ до впровадження технологій аналітики зворотного зв'язку [14].



Рисунок 1.3 – Qualtrics XM

Qualtrics XM, наприклад — це приклад саме комерційного хмарного рішення, тобто такого, що базується на моделі програмного забезпечення як послуги (SaaS). Користування платформою можливе лише при оформленні відповідної підписки: вона включає в себе підтримку, оновлення та широкий спектр можливостей для керування досвідом користувачів (Experience Management). Вона використовується не лише в освітньому середовищі, а й у бізнесі, маркетингових дослідженнях, UX-аналітиці, управлінні брендом тощо.

					ІАЛЦ.467200.003 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.1 – Порівняльна таблиця систем для оцінювання курсів

Назва платформи	Тип рішення	Основні функції	Інтерфейс користувача	Додаткові можливості	Особливості впровадження
Explorance Blue	Комерційне	Створення та проведення опитувань, збір зворотного зв'язку	Веб-інтерфейс для студентів та адміністраторів	Аналітика, інтерактивні графіки, звіти	Використовується в багатьох університетах
Qualtrics XM	Комерційне (SaaS)	Управління досвідом, проведення опитувань	Веб-інтерфейс, адаптація під мобільні пристрої	Потужна аналітика, автоматичне формування звітів	Хмарна платформа, доступ за підпискою
Watermark / EvaluationKIT	Комерційне	Курсова оцінка, створення та розсилка опитувальників	Інтуїтивний веб-інтерфейс	Інтеграція з LMS, звітність, KPI	Поширене в США, підтримка стандартів
EvaP	Open-source	Опитування, оцінка курсів, збір результатів	Зручний інтерфейс для студентів, адміністраторів	Відкритий код, кастомізація, інтеграція	Розробка HPI, Python/Django
LMS-інтегровані опитування (Moodle, Canvas тощо)	Вбудовані в LMS	Базове створення і проходження опитувань	В межах LMS	Обмежена аналітика, просте використання	Залежить від LMS, без додаткових витрат

Існуючі технологічні рішення для збору та аналітичного опрацювання студентських відгуків охоплюють як комерційні платформи, так і відкриті системи. Ключовими рисами сучасних реалізацій є веб-орієнтованість (онлайн-доступ для всіх учасників), підтримка анонімності та зручності проходження опитування для студентів, автоматичне обчислення узагальнених показників та гнучке представлення результатів (графіки, звіти). Це підтверджує доцільність розробки власної веб-платформи з архітектурою «клієнт–сервер», де фронтенд виступає клієнтом (браузер або мобільний пристрій студента), а бекенд на сервері обробляє дані і зберігає їх у базі даних. Такий підхід забезпечить масштабованість і інтеграцію системи в існуючу ІТ-інфраструктуру університету (як окремий модуль або в складі LMS [11]).

1.2 Аналіз існуючих методологій до скорингу освітньо-професійних програм та компонент

Багатокритеріальний аналіз є потужним інструментом для оцінювання складних систем, де загальна ефективність або задоволеність визначається сукупністю різних чинників. Однією з найбільш відомих і практично застосовуваних моделей у цій сфері є модель MUSA (Multi-criteria Satisfaction Analysis), яка знайшла широке застосування в дослідженнях рівня задоволеності споживачів і згодом була успішно адаптована до сфери освітніх послуг [8].

Метод MUSA ґрунтується на гіпотезі, що загальна задоволеність суб'єкта (у цьому випадку — студента) є функцією від ряду часткових критеріїв. До таких критеріїв у контексті освітньої діяльності можуть належати: зміст курсу, робота викладача, доступність навчальних матеріалів, організація занять, застосованість знань на практиці тощо. Суть методу полягає у визначенні вагових коефіцієнтів кожного критерію на основі опитувальних даних з подальшим розв'язанням задачі багатокритеріальної оптимізації. Таким чином, модель дозволяє не лише встановити рівень загальної задоволеності, а й провести структурний аналіз — тобто виявити, які з критеріїв найбільше впливають на загальну оцінку, а які можуть мати приховані резерви для покращення.

З технічної точки зору, MUSA є методом на основі порядкової регресії, де залежна змінна (загальна задоволеність) розглядається як порядкова шкала, а незалежні змінні — це оціночні характеристики окремих аспектів. За результатами аналізу формується профіль важливості кожного критерію, що дозволяє проводити рекомендаційний аналіз для адміністрації чи викладачів. Наприклад, якщо виявлено, що значна частина загальної оцінки формується на основі показників викладацької майстерності, це свідчить про

доцільність інвестування у підвищення кваліфікації викладачів або впровадження педагогічних тренінгів.

У практичному застосуванні MUSA довела свою ефективність. Так, наприклад, в одному з досліджень, проведених серед бізнес-студентів грецького університету, рівень загальної задоволеності було оцінено на рівні близько 86 %, при цьому найважливішими факторами виявилися якість викладацького складу та структура навчальної програми. Подібні дослідження дають змогу формувати обґрунтовану політику розвитку освітніх програм і вчасно реагувати на виклики в освітньому середовищі.

Окрему увагу у сучасних підходах до аналізу зворотного зв'язку приділяють неструктурованим даним, зокрема — відкритим коментарям студентів. Вони, на відміну від числових оцінок, несуть якісну інформацію, що може вказувати на глибинні причини позитивної або негативної оцінки. Коментарі містять вільну мову, іноді емоційно забарвлену, і можуть висвітлювати аспекти, не передбачені у формалізованій анкеті. Обробка таких даних є складнішим завданням і відноситься до галузі обробки природної мови — NLP.

Один із ключових напрямів у цьому контексті — аналіз тональності (Sentiment Analysis). Суть цього підходу полягає у визначенні емоційного забарвлення тексту (позитивного, нейтрального чи негативного), а також — у виявленні ключових об'єктів, дій чи аспектів, до яких це забарвлення відноситься. Завдяки застосуванню алгоритмів машинного навчання та лінгвістичних правил, sentiment analysis дозволяє автоматизовано оцінювати сотні й тисячі коментарів, що особливо актуально у випадку великомасштабних опитувань.

Емпіричні дослідження показують, що sentiment analysis є дієвим інструментом в освітньому контексті. Наприклад, у рамках аналізу відкритих відповідей студентів на платформах масових онлайн-курсів (МООС) було виявлено, що автоматичне розпізнавання настрою коментарів значно

					ІАЛЦ.467200.003 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

полегшує роботу викладачів та адміністраторів, дозволяючи оперативно виявляти найбільш болючі теми або позитивно сприйняті елементи курсу [5]. Якщо, наприклад, алгоритм виявляє часте згадування терміну «організація занять» у негативному контексті, це може слугувати сигналом для термінового перегляду розкладу або структури проведення занять.

Більш того, сучасні моделі глибинного навчання — зокрема нейронні мережі, моделі типу трансформерів (наприклад, BERT, RoBERTa тощо) — дозволяють не лише досягати високої точності класифікації, а й здійснювати автоматичне узагальнення відгуків. Такий підхід дозволяє формувати короткі, інформативні резюме на основі великої кількості коментарів, що надзвичайно корисно у процесі прийняття управлінських рішень.

Ще один перспективний напрям — тематичне моделювання, наприклад, метод LDA. Цей метод дозволяє розбивати великі обсяги тексту на групи (теми) за змістом, автоматично виділяючи ключові напрями фідбеку, такі як «розклад занять», «доступ до матеріалів», «якість зворотного зв'язку», «оцінювання» тощо. У поєднанні з аналізом тональності це відкриває можливість багатовимірного аналізу, в межах якого можна оцінити не лише загальне ставлення студентів, а й емоційний фон у розрізі кожної теми.

В рамках нашого проєкту основна увага приділяється обробці числових оцінок, отриманих внаслідок опитування студентів, що дозволяє провести кількісний аналіз загальної задоволеності. Таким чином, поєднання багатокритеріальної оптимізації (на основі числових даних) з інтелектуальним аналізом неструктурованої текстової інформації дозволяє формувати комплексний підхід до оцінювання задоволеності студентів. Такий підхід має не лише аналітичну, а й стратегічну цінність для закладів вищої освіти, які прагнуть до підвищення якості навчального процесу, оптимізації організаційних процесів та ефективного реагування на потреби студентської аудиторії.

					ІАЛЦ.467200.003 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

1.3. Формалізація задачі та постановка часткових задач проєктування

На основі аналізу існуючих технологічних та методологічних рішень у сфері оцінювання якості освітніх програм було обґрунтовано потребу у створенні цілісної, інтегрованої системи, що дозволяє автоматизувати процес збору, обробки та аналізу зворотного зв'язку від студентів. У контексті зростаючого значення освітньої аналітики в управлінні якістю освітнього процесу особливо актуальним постає завдання формалізації підходів до побудови такої системи, що передбачає чітке визначення її функціональних і нефункціональних вимог, математичних моделей обробки даних, а також структурних елементів архітектури.

Основними особливостями, які має мати платформа є:

1. Формування опитувальника з групами питань за категоріями (якість викладання, актуальність змісту, інтерактивність, складність курсу, загальна оцінка тощо),
2. Організацію анонімного опитування студентів через веб-інтерфейс із можливістю ідентифікації курсу та семестру,
3. Зберігання відповідей у базі даних із забезпеченням цілісності та конфіденційності даних,
4. Автоматизовану обробку результатів, як, наприклад, обчислення середніх значень.

Важливо зазначити, що система повинна підтримувати динамічне формування анкети, що дозволяє змінювати або доповнювати набір критеріїв без потреби переробки основного коду. Також у структурі анкети можуть бути запроваджені як кількісні запитання (шкала 1–5).

На основі вищезазначеного можна сформулювати основні часткові задачі проєктування, вирішення яких забезпечить реалізацію повнофункціональної платформи:

					ІАЛЦ.467200.003 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

Провести аналіз підходу до аналітичного скорингу. Визначити перелік критеріїв, за якими здійснюється оцінювання, обґрунтувати їх релевантність з точки зору впливу на якість навчання та визначити оцінки загального рівня задоволеності курсом.

Розробити структуру анкети — розробити формат анкетування з урахуванням UX-принципів та надання студентам зручного і зрозумілого інтерфейсу для заповнення опитувань, що не перевищує 5–10 хвилин за тривалістю.

Побудувати логічну та фізичну моделі бази даних — визначити структуру збереження відповідей, метаданих про курси, семестри, категорії питань та зв'язків між ними.

Впровадити аналітичний підхід до скорингу — реалізувати алгоритми обчислення середніх оцінок, розподілів, стандартних відхилень, кореляцій, а також можливих залежностей між показниками (наприклад, між задоволеністю і складністю курсу).

Розробити систему авторизації та ролей — забезпечити багаторівневий доступ до системи: для студентів (анонімне опитування), викладачів (перегляд агрегованих результатів), адміністраторів (створення курсів, генерація звітів). Функцію реєстрації можна відкинути — система буде працювати за прикладом Електронного кампуса КІІ або системи Moodle, де за реєстрацію та додавання студентів до системи відповідає адміністрація.

Побудувати інтерфейс аналітики та візуалізації — реалізувати панель для відображення звітів у форматі графіків, діаграм.

Забезпечити масштабованості і розширюваності системи — архітектура повинна передбачати можливість підключення додаткових модулів (наприклад, NLP для аналізу відкритих коментарів) або переходу до масштабованої бази даних (PostgreSQL, MongoDB).

					ІАЛЦ.467200.003 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

Виконання цих задач забезпечить якісний та сучасний застосунок, який буде відповідати вимогам сучасності та надасть позитивний відгук зі сторони користувачів.

					ІАЛЦ.467200.003 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 1

Проведене в межах розділу дослідження засвідчило, що на сучасному етапі розвитку вищої освіти ключову роль у забезпеченні її якості відіграє організований та систематичний аналіз зворотного зв'язку студентів. Зокрема, було виявлено, що міжнародна практика активно застосовує цифрові системи оцінювання освітніх курсів, які базуються на використанні анкетування та обчислення агрегованих показників задоволеності. Ці системи — як комерційні (Explorance Blue, Watermark), так і open-source (EvaP) — дозволяють не лише отримувати великі обсяги даних у зручному форматі, але й інтегрувати їх в інформаційні потоки закладів освіти, забезпечуючи тим самим прозорість, обґрунтованість управлінських рішень та підзвітність.

Аналіз методологічних підходів показав, що найбільш поширеними є методи описової статистики, які дозволяють визначати базові характеристики розподілу оцінок (середнє значення, медіана, мода), а також кореляційний аналіз для виявлення взаємозв'язків між критеріями задоволеності та іншими чинниками (наприклад, складністю курсу або його практичною цінністю). Водночас інтенсивно розвивається підхід до багатокритеріальної оцінки задоволеності, зокрема метод MUSA, який дозволяє враховувати вагу кожного критерію у формуванні загального індексу якості. Додатково перспективним напрямом виявлено аналіз відкритих текстових відповідей за допомогою методів обробки природної мови (NLP), включно з аналізом тональності та тематичним моделюванням.

Таким чином, розділ не лише окреслив актуальні технологічні та аналітичні інструменти для реалізації скорингу, але й підтвердив доцільність створення вітчизняної інтегрованої системи, яка б дозволила адаптувати кращі практики до специфіки національного освітнього середовища. Отримані результати дали змогу чітко сформулювати цілі та завдання проєкту, визначити

					ІАЛЦ.467200.003 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

функціональні вимоги до системи та закласти методичну основу для розробки власної платформи оцінювання якості курсів.

					ІАЛЦ.467200.003 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2

РОЗРОБКА АПАРАТНО-ПРОГРАМНОЇ ПЛАТФОРМИ З АНАЛІТИЧНОГО СКОРИНГУ ОСВІТНЬО-ПРОФЕСІЙНИХ ПРОГРАМ ОСВІТНІХ КОМПОНЕНТ

2.1 Розробка структури апаратної компоненти платформи з аналітичного скорингу освітньо-професійних програм

Основу функціонування платформи становить клієнт-серверна архітектура, яка є однією з найпоширеніших моделей у сучасних інформаційних системах. В межах цієї архітектурної моделі ключову роль відіграє центральний сервер (або серверна частина програмного забезпечення), який виконує завдання координації, обробки й зберігання інформації. Цей сервер взаємодіє з численними клієнтськими пристроями, що можуть бути представлені як веб-браузерами на персональних комп'ютерах, так і мобільними застосунками на смартфонах чи планшетах, якими користуються як студенти, так і адміністративний персонал.

Сервер виконує кілька важливих функцій: він приймає дані, зібрані під час анкетування, обробляє їх відповідно до визначених алгоритмів, зберігає результати у базі даних та формує аналітичні звіти або візуалізації на їх основі. У свою чергу, клієнтська частина платформи забезпечує зручний інтерфейс користувача, що дозволяє вводити дані, переглядати результати та виконувати інші інтерактивні дії, пов'язані з опитуваннями й аналітикою [11].

До основних структурних компонентів системи належать:

Сервер додатка — це може бути як виділений фізичний сервер, розміщений на базі університету, так і віртуальний сервер у хмарному середовищі. Він виконує роль веб-сервера, що відповідає за прийом запитів, виконання бізнес-логіки, а також взаємодію з базою даних. За потреби, така архітектура може бути масштабована шляхом додавання нових серверів або

					ІАЛЦ.467200.003 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

переходу до кластерного рішення. Розгортання системи можливе у локальному середовищі університету або ж у хмарних платформах, при цьому обов'язково дотримуючись вимог щодо захисту персональних даних і національного законодавства в сфері інформаційної безпеки [4].

Клієнтські пристрої — до них належать ноутбуки, настільні комп'ютери, смартфони, планшети, за допомогою яких користувачі можуть підключатися до системи. Платформа оптимізована для коректного відображення на різних типах екранів, що дозволяє використовувати її на будь-якому пристрої без необхідності встановлення додаткового програмного забезпечення. Інтерфейс є адаптивним, тобто автоматично підлаштовується під розмір і формат екрана користувача.

Мережеве з'єднання — система підтримує як роботу в локальній мережі університету, так і віддалений доступ через інтернет. Це дозволяє користувачам проходити анкетування або переглядати результати з будь-якого місця. Для гарантування безпеки даних та недопущення несанкціонованого доступу, передбачено використання захищеного протоколу HTTPS, який шифрує передані дані.

Процес обміну інформацією між клієнтом і сервером відбувається наступним чином: користувач через клієнтський інтерфейс формує запит, який надсилається через мережу, проходить через маршрутизатор та потрапляє на серверну частину застосунку. Після обробки сервер може звернутися до бази даних для отримання або збереження необхідної інформації. В результаті, відповідь повертається до клієнтського пристрою, де відображається у зрозумілому для користувача вигляді.

Завдяки такій структурі забезпечується стабільність, гнучкість і масштабованість системи, що дозволяє їй ефективно функціонувати в умовах великої кількості користувачів, зберігаючи високу швидкість обробки запитів і надійність збереження даних.

Структурна схема апаратної компоненти представлена на рис. 2.1.

					ІАЛЦ.467200.003 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

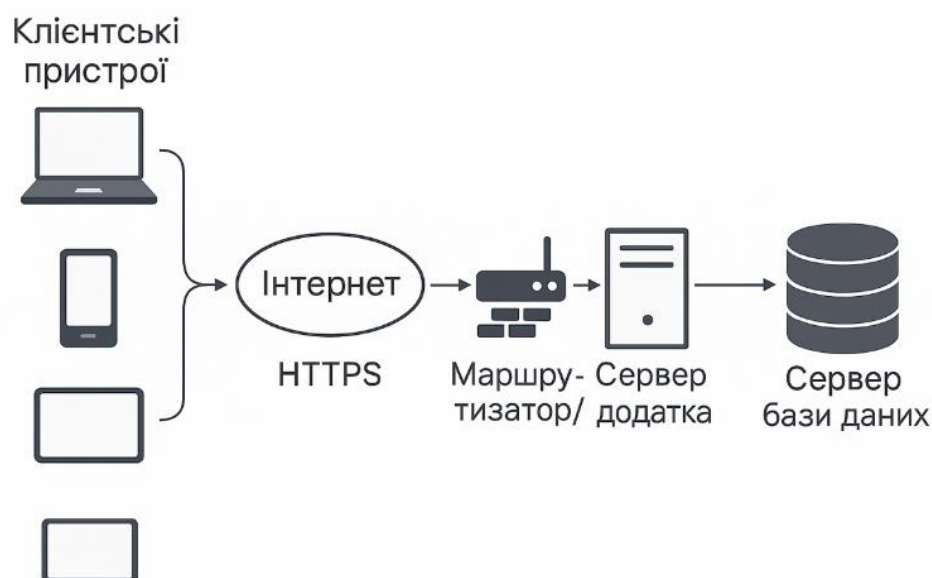


Рисунок 2.1 – Структурна схема апаратної компоненти

Підсумовуючи: запити надсилаються від клієнтів, проходять через маршрутизатор до серверу застосунку. Далі до серверу бази даних, звідки виконується зворотній зв'язок з результатами запитів клієнтам.

2.2 Розробка програмної компоненти платформи з аналітичного скорингу освітньо-професійних програм

2.2.1 Математична модель скорингу освітньо-професійних програм

Набір критеріїв (факторів) оцінювання $F_1, F_2, \dots, F_m$.

Точний перелік і кількість критеріїв визначається на етапі розробки анкети. Згідно з вимогами анкети містить питання, згруповані за критеріями, наприклад:

F_1 – ясність викладання матеріалу,

F_2 – актуальність та сучасність курсу,

F_3 – наявність інтерактивності у навчанні,

F_4 – корисність курсу для майбутньої професії,

F_5 – складність курсу тощо [13].

Для кожного з курсів розраховуємо середній рівень задоволеності за формулою.

$$S_i = \frac{1}{n_i} \sum_{k=1}^{n_i} a_k \quad (2.1)$$

Де S_i – середній рівень задоволеності курсом,

a_k – відповідь k-го студента для курсу,

n_i – кількість відповідей.

У випадку відсутності відповідей на анкету, $S_i = null$.

Така модель забезпечує всі потрібні величини для аналітичного звіту по курсу (рівні задоволеності, розподіли оцінок, залежності між характеристиками курсу та рівнями задоволеності / оцінок) [6; 10].

2.2.2 Інженерія вимог

У процесі розробки аналітичної платформи особливу увагу було приділено формалізації функціональних і нефункціональних вимог до системи. Ці вимоги окреслюють не лише обсяг функціональності, але й визначають якісні характеристики, які система має задовольняти, аби бути придатною до практичного використання в освітньому середовищі.

З функціональної точки зору, першочерговим завданням системи є забезпечення гнучкого механізму створення і керування анкетами для опитування студентів. Адміністратор має змогу формувати опитувальники, які складаються з набору запитань, згрупованих за ключовими критеріями оцінювання освітнього процесу. Система дозволяє визначати типи запитань (наприклад, числові шкали типу Лікєрта, відкриті коментарі, демографічні поля тощо), задавати їхню послідовність, а також структурувати блоки таким чином, щоб підвищити логічність та зручність заповнення.

Після створення анкети система повинна забезпечувати процедуру проведення опитування, що ґрунтується на принципах добровільності та

					ІАЛЦ.467200.003 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

анонімності. Студент може взяти участь в оцінюванні, авторизувавшись через персоналізоване посилання або через загальний обліковий запис. Система гарантує, що кожен студент надає відповіді лише один раз на конкретний курс у межах визначеного семестру, не зберігаючи при цьому персональних ідентифікаційних даних, що дозволяє зберегти приватність та етичність дослідження.

Усі надані відповіді автоматично зберігаються в базі даних, при цьому кожна відповідь зв'язується з унікальним ідентифікатором курсу та часовим інтервалом (семестром), але не з конкретною особою. Завдяки цьому можливо здійснювати подальший аналіз без порушення конфіденційності. Така модель даних дозволяє проводити позовжній аналіз динаміки оцінювання для окремих курсів або програм.

Особливе значення має модуль аналітики, який здійснює обробку зібраних оцінок. Система автоматично обчислює середні значення, визначає розподіли відповідей, зводить агреговані результати у форматі, придатному для порівняння. За потреби, можливе обчислення додаткових показників, таких як коефіцієнт варіації або кореляція між критеріями (наприклад, зв'язок між складністю курсу та загальною задоволеністю, або між практичною цінністю та актуальністю тематики). Також передбачена опціональна можливість інтеграції даних з систем обліку успішності для аналізу впливу параметрів курсу на академічні результати.

Крім того, система реалізує функціональність керування довідковою інформацією, яка включає список курсів, навчальних програм, семестрів, груп студентів і викладачів. Цим компонентом керує адміністратор через окремий модуль, що дозволяє актуалізувати навчальну структуру та коректно пов'язувати відгуки з відповідними навчальними елементами.

Не менш важливою є система керування доступом, яка забезпечує обмеження прав користувачів залежно від їхніх ролей. Наприклад, викладач має доступ лише до агрегованих результатів по своєму курсу без розшифрування

					ІАЛЦ.467200.003 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

окремих відповідей, тоді як адміністрація може формувати зведені звіти по кафедрах чи програмах. Публікація результатів дозволена лише після завершення періоду збору відповідей, що унеможлиблює вплив опитувань на поточний навчальний процес.

Щодо продуктивності, система має відповідати вимогам масштабованості, тобто коректно функціонувати в умовах зростання кількості користувачів і оброблюваних даних. Зокрема, має забезпечувати підтримку щонайменше 10 освітніх програм, 100 курсів і декількох тисяч студентів без істотного зниження швидкодії або доступності сервісу.

Окрім функціональних характеристик, були визначені й нефункціональні вимоги, що стосуються зручності, ефективності та якості взаємодії з системою. Насамперед, користувацький інтерфейс має бути інтуїтивно зрозумілим, забезпечуючи можливість проходження опитування не більше ніж за 5–10 хвилин. Система має оперативно реагувати на дії користувача, зокрема обробка запитів повинна відбуватись у реальному часі, а формування аналітичного звіту — тривати не більше кількох секунд.

Особливе значення надається анонімності, яка досягається як на рівні архітектури бази даних (відсутність зберігання ПІБ або IP-адрес), так і за рахунок логіки контролю доступу. Анонімність є критично важливою умовою для формування чесних і щирих відповідей студентів.

Важливим аспектом також є сумісність системи з іншими сервісами, зокрема можливість імпорту існуючих даних про курси та користувачів з LMS або експорту звітів у стандартні формати (CSV, Excel, PDF) для подальшої роботи з ними в офлайн-режимі.

Система повинна бути надійною, тобто мати функції транзакційного збереження, регулярного резервного копіювання та механізми автоматичного перезапуску у випадку збоїв (через Docker, systemd або інші сучасні засоби розгортання).

					ІАЛЦ.467200.003 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

Важливою вимогою для підтримки подальшої розробки є модульність архітектури, що передбачає дотримання принципів SOLID, використання шаблонів проєктування та забезпечення незалежності між основними функціональними блоками.

Одним із визначальних чинників є документованість, що включає коментування коду, опис REST API, наявність інструкцій для кінцевого користувача та адміністратора. Це підвищує довговічність рішення, забезпечує легке залучення нових розробників або інтеграцію з іншими освітніми сервісами в межах цифрової екосистеми університету.

2.2.3 Архітектурне проєктування програмної компоненти

У межах реалізації програмної системи було спроектовано двошарову архітектуру, яка складається з клієнтської частини (front-end) та серверної частини (back-end), що взаємодіють між собою через RESTful API. Такий підхід забезпечує гнучкість, модульність, розділення обов'язків і можливість масштабування в майбутньому. Серверна частина системи реалізована за допомогою Node.js та Express.js. Зберігання інформації здійснюється у легкій файловій JSON-базі даних, що розміщується у структурі проєкту в окремому файлі. Така реалізація є оптимальною для прототипів та навчальних інформаційних систем із невеликою кількістю користувачів.

Взаємодія з базою даних реалізується через набір REST API-ендпоїнтів. Вони дозволяють здійснювати автентифікацію користувачів, керувати курсами та здійснювати отримання аналітичної інформації.

Клієнтський шар реалізований з використанням бібліотеки React.js. Це дозволяє побудувати інтерактивний односторінковий застосунок (SPA), де оновлення вмісту відбувається динамічно без повного перезавантаження сторінки. Структура компонентів поділена за функціональними зонами: логін, домашня сторінка, список курсів, панель адміністратора тощо.

Основні принципи реалізації клієнтської частини:

					ІАЛЦ.467200.003 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

Контекстна модель (Context API) — застосовується для глобального зберігання авторизаційної інформації (AuthContext) та даних курсів і відповідей (CourseContext).

Маршрутизація (React Router) — забезпечує логіку переходу між сторінками з урахуванням ролі користувача (адміністратор або студент).

Авторизація — після входу користувача токен зберігається у localStorage і автоматично додається до всіх API-запитів.

Формування анкет і надсилання оцінок — здійснюється через форму з варіантами відповідей, які надсилаються на сервер у вигляді масиву об'єктів.

Рольовий доступ реалізовано безпосередньо у клієнтському коді — доступ до адміністративного функціоналу надається лише користувачам з відповідною роллю. Таким чином, функціонал системи динамічно адаптується до контексту користувача.

Кожен запит із клієнта до сервера здійснюється через HTTP-запити з токеном авторизації у заголовку. Сервер перевіряє автентичність запиту й надає відповідь у форматі JSON. Така взаємодія гарантує ізольованість логіки бізнес-рівня від візуального представлення даних, що відповідає принципам сучасної веб-архітектури.

Компонентами серверного застосунку є:

Модуль опитувань – створення, редагування та публікація анкет для адміністратора та отримання анкети й збереження відповідей для студента.

Модуль аутентифікації та безпеки – керування входом користувачів (опціонально через ролі або токени). У прототипі допускається спрощений вхід без реєстрації.

Аналітичний модуль – обчислення статистики: середні оцінки, розподіли, кореляції.

Модуль керування довідниками – API для редагування списків програм, курсів, питань [6].

API-Endpoints:

					ІАЛЦ.467200.003 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

POST /api/login – логін користувача
 GET /api/courses – отримати список всіх курсів
 POST /api/courses – створити новий курс
 GET /api/responses – отримати всі відповіді
 POST /api/responses – зберегти відповіді користувача на курс
 DELETE /api/responses – видалити відповіді конкретного користувача для конкретного курсу

Сутності бази даних:

Course – навчальна дисципліна

User – користувач

Response – відповідь користувача на нкету

Створена модель забезпечує гнучкість створення опитувань, де запитання можуть задавати окремо для кожного опитування. Архітектура програмної компоненти реалізована за класичною моделлю "клієнт-сервер". Це забезпечує прозору логіку, розширюваність та можливість переходу до повноцінної СКБД (наприклад, PostgreSQL чи MongoDB) без необхідності кардинальних змін у структурі застосунку. Весь стек технологій був обраний з урахуванням легкості розгортання, підтримки відкритих стандартів та актуальності в сучасному веб-розробленні.

2.2.4 Розробка програмної компоненти платформи з скорингу освітньо-професійних програм

При створенні серверного шару було реалізовано базову файлову структуру з окремими сутностями, які логічно відповідають основним моделям домену: Course, User та Response. Вони зберігаються у вигляді масивів у структурованому файлі db.json. Кожна сутність має чітко визначені атрибути: наприклад, об'єкти Response включають дані про користувача, курс, запитання, оцінку та дату заповнення.

Обробку запитів реалізовано через Express.js у вигляді REST-контролерів, які реагують на запити до маршрутів /api/courses, /api/responses та

/api/login. Так, контролер авторизації (POST /api/login) перевіряє автентичність користувача й у разі успіху повертає JWT-токен разом із базовою інформацією про роль та курс навчання.

Обробка збереження відповідей реалізована в межах маршруту POST /api/responses: сервер приймає масив відповідей, зчитує файл бази даних, додає нові записи та зберігає оновлення. Перед записом перевіряється, чи користувач раніше вже подавав відповіді на цей курс, і в разі виявлення — попередні відповіді видаляються, що дозволяє повторне проходження анкети без дублювання даних. У маршруті GET /api/responses реалізовано повернення всіх відповідей — ці дані використовуються для побудови адміністративних звітів.

Клієнтський шар системи реалізовано як React-додаток зі структурованою системою компонентів. Ключовим елементом є контекст авторизації AuthContext, який зберігає інформацію про поточного користувача та автоматично додає токен до всіх HTTP-запитів. Аналогічно, CourseContext керує курсами та відповідями студентів, реалізуючи логіку взаємодії з API.

Компонент CourseEvaluator.js відповідає за відображення анкети для обраного курсу. Питання анкети жорстко закодовані у вигляді масиву, а відповіді обираються через елементи <select> для кожного запитання. Після заповнення форми дані збираються в масив об'єктів і надсилаються на сервер за допомогою HTTP POST. У компоненті AdminPanel.js адміністратор може створювати нові курси, а також переглядати дані відповіді у табличному вигляді.

Візуальне компонування сторінок реалізовано з використанням Bootstrap 5: застосовується grid-система, кнопки, картки, індикатори та інші компоненти. У домашній сторінці (HomePage.js) відображаються ролі користувача, привітання та навігаційні кнопки, а також додано кнопку виходу з системи (LogoutButton), яка очищує контекст і перенаправляє користувача на головну сторінку.

					ІАЛЦ.467200.003 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

Проект відповідає принципам чистої архітектури: логіка поділена на контексти, компоненти мають єдину відповідальність, а імена функцій та змінних є самодокументованими (наприклад, `addCourse`, `submitSurvey`). Завдяки застосуванню моделі з токенами автентифікації клієнтська та серверна частина чітко відокремлені, що дозволяє легко масштабувати систему або переносити її до повноцінної бази даних (наприклад, MongoDB чи PostgreSQL).

На поточному етапі реалізовано функціональний прототип програмної частини освітньої аналітичної платформи. Усі компоненти взаємодіють узгоджено, підтверджуючи відповідність функціональним вимогам та базовим принципам архітектурного проектування сучасних вебзастосунків.

Наразі реалізовано базову версію програмної частини платформи. Наведено аргументовані пояснення та здійснено оцінку відповідності вимогам.

					ІАЛЦ.467200.003 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 2

У другому розділі було розроблено повноцінну архітектурну модель апаратно-програмної платформи, яка покликана забезпечити повний цикл аналітичного скорингу освітніх компонент — від збору первинних оцінок до генерації комплексних звітів і візуалізації даних. Побудована система базується на принципах модульності, масштабованості та клієнт-серверної взаємодії, що дозволяє гнучко інтегрувати її як у локальні, так і у віддалені ІТ-інфраструктури закладів вищої освіти.

У межах архітектурного проектування запропоновано логічно розмежовану структуру, де фронтенд реалізовано засобами React.js для забезпечення динамічного інтерфейсу користувача, а бекенд — на базі Node.js та Express.js з використанням REST API для забезпечення обробки запитів, авторизації користувачів та збереження відповідей у JSON-базі. Така реалізація не лише відповідає сучасним тенденціям веб-розробки, а й дозволяє в подальшому легко перейти на промислові СКБД, не змінюючи логіку застосунку.

Важливо, що в рамках платформи було реалізовано аналітичний підхід до скорингу оцінок, яка дозволяє отримувати індикативні показники якості курсів на основі узагальнення думок студентів за заданими критеріями. Ключовими особливостями системи стали підтримка анонімності, захищений доступ, модуль управління анкетами, автоматичний підрахунок середніх оцінок, а також можливість виведення звітів з інтерактивною візуалізацією.

Були визначені чіткі вимоги щодо розробки платформи з аналітичного скорингу освітньо-професійних програм та освітніх компонент.

					ІАЛЦ.467200.003 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3

ПРИКЛАДНІ АСПЕКТИ ЗАСТОСУВАННЯ ПЛАТФОРМИ З АНАЛІТИЧНОГО СКОРИНГУ ОСВІТНЬО-ПРОФЕСІЙНИХ ПРОГРАМ ТА ОСВІТНІХ КОМПОНЕНТ

3.1 Демонстрація практичних можливостей розробленої системи

У цьому підрозділі продемонструємо, як працює створена платформа на практиці, на прикладі оцінювання умовної освітньої програми. Ми розглянемо процес проведення опитування студентів, а також одержання і інтерпретацію аналітичних звітів, що генерує система. Для демонстрації використано тестові дані, що моделюють реальні відгуки студентів.

Налаштування сценарію: Припустимо, у весняному семестрі 2025 року на освітній програмі “Комп’ютерна інженерія” проводиться оцінювання наступних курсів: “Алгоритми та структури даних”, “Комп’ютерна теорія”, “Психологія розробника”, “Програмування мікропроцесорних систем”, “Інженерія соціального побуту”, “Теорія прийняття рішень”, “Системна розробка програмного забезпечення”, “Машинне навчання та штучний інтелект”, “Кібербезпека та захист даних”, “Розробка мобільних додатків”, “Комп’ютерна графіка та візуалізація”, “Хмарні технології та DevOps”, “Бази даних та аналіз великих даних”, “Розробка ігор та віртуальна реальність”, “Блокчейн та криптографія”. Група складається з 50 студентів. Для простоти продемонструємо результат по одному курсу (“Алгоритми та структури даних”). Анкета містила 5 груп питань (відповідно до 5 критеріїв оцінки курсу, згаданих раніше). Кожен критерій охоплював 2-3 конкретних питання. Таким чином, загалом близько 12 питань, кожне з оцінкою від 1 до 5.

Інтерфейс студента (заповнення анкети): Студент заходить у систему через веб-браузер за унікальним посиланням. Далі студень бачить привітання зі сторони сервісу, а також, прохання про авторизацію. Авторизація у сервісі працює схожим чином на мудл КПП, де вже існують логіни та паролі, які вписані

					ІАЛЦ.467200.003 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

в систему. Як тільки студент отримує від університету дані для авторизації – він має можливість оцінювати дисципліни з курсу.

Ласкаво просимо на освітню платформу!

Будь ласка, [увійдіть](#), щоб продовжити.

Рисунок 3.1 – Привітання та перехід до авторизації

Студент натискає на посилання, що веде до авторизації та бачить перед собою стандартне вікно входу в системи, яке містить у собі заголовок, а також два текстових поля, куди треба вписати поштову скриньку (Email) студента та його пароль.

Вхід до системи

Email:

Пароль:

Увійти

Рисунок 3.2 – Вікно входу до системи

					ІАЛЦ.467200.003 ПЗ	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		

У випадку некоректно введених даних для авторизації, яких немає у системі студент отримає помилку.

Вхід до системи

Email:

Пароль:

Невірний email або пароль

Увійти

Рисунок 3.3 – Невдалий вхід до системи

Вхід до системи

Email:

Пароль:

Увійти

Рисунок 3.4 – Використання коректних даних для авторизації

					ІАЛЦ.467200.003 ПЗ	Арк.
						33
Зм.	Арк.	№ докум.	Підпис	Дата		

У випадку коректно введених даних для авторизації, студент потрапляє на наступну сторінку, де бачить привітання та повідомлення про можливість оцінювання дисципліни своєї освітньої програми. Додатково є кнопка для переходу до оцінювання.



Освітня Платформа Аналітики

Оцінюйте та аналізуйте освітні програми університету з легкістю!

Вітаємо, студент 3-го курсу!

Ви можете оцінити дисципліни своєї освітньої програми.

[Переглянути курси для оцінювання](#)

Рисунок 3.5 – Головна сторінка студента

При переході до оцінювання студент має можливість побачити всі доступні курси для оцінювання.

Поруч з назвою кожного з курсів можна побачити назву освітньо-навчальної програми, номер курсу та перехід до оцінювання конкретного курсу.

На даному етапі цілю є забезпечити зручність та комфорт користувача при вході в застосунок, адже за тривалий час навчання студента список доступних курсів для оцінювання може значно розширитися.

Всі доступні курси розташовані у вигляді простого списку, не перевантажуючи студента. Це зображено на Рисунку 3.6 – Доступні курси для оцінювання.

Доступні курси для оцінювання

Алгоритми та структури даних (Комп'ютерні науки, 3 курс)	Оцінити
Комп'ютерна теорія (Комп'ютерні науки, 3 курс)	Оцінити
Психологія розробника (Комп'ютерні науки, 3 курс)	Оцінити
Програмування мікропроцесорних систем (Комп'ютерні науки, 3 курс)	Оцінити
Інженерія соціального побуту (Комп'ютерні науки, 3 курс)	Оцінити
Теорія прийняття рішень (Комп'ютерні науки, 3 курс)	Оцінити
Системна розробка програмного забезпечення (Комп'ютерні науки, 3 курс)	Оцінити
Машинне навчання та штучний інтелект (Комп'ютерні науки, 3 курс)	Оцінити
Кібербезпека та захист даних (Комп'ютерні науки, 3 курс)	Оцінити
Розробка мобільних додатків (Комп'ютерні науки, 3 курс)	Оцінити
Комп'ютерна графіка та візуалізація (Комп'ютерні науки, 3 курс)	Оцінити
Хмарні технології та DevOps (Комп'ютерні науки, 3 курс)	Оцінити
Бази даних та аналіз великих даних (Комп'ютерні науки, 3 курс)	Оцінити
Розробка ігор та віртуальна реальність (Комп'ютерні науки, 3 курс)	Оцінити
Блокчейн та криптографія (Комп'ютерні науки, 3 курс)	Оцінити

Рисунок 3.6 – Доступні курси для оцінювання

При переході до анкети оцінювання курсу, наприклад, «Алгоритми та структури даних» студент бачить на екрані заголовки курсу. “Оцініть, будь ласка, наведені аспекти курсу. Ваші відповіді анонімні та будуть використані для покращення якості навчання”.

Анкета оцінювання курсу: Алгоритми та структури даних

Освітня програма: Комп'ютерні науки, 3 курс

Оцініть, будь ласка, наведені аспекти курсу. Ваші відповіді анонімні та будуть використані для покращення якості навчання.

1. Викладач зрозуміло пояснював теоретичний матеріал. (1-5)

Оберіть оцінку

2. Викладач доступно відповідав на запитання студентів.(1-5)

Оберіть оцінку

3. Тематика курсу є сучасною і відповідає вимогам галузі. (1-5)

Оберіть оцінку

4. Курс був корисним для розуміння інших дисциплін. (1-5)

Оберіть оцінку

5. Курс містив інтерактивні елементи в навчанні. (1-5)

Оберіть оцінку

6. Під час занять використовувались практичні приклади, демонстрації або інтерактивні завдання. (1–5)

Оберіть оцінку

7. Я був(ла) залучений(на) до обговорень або активностей на заняттях. (1–5)

Оберіть оцінку

8. Навчальне навантаження (домашні завдання, проекти) було помірним і посильним. (1–5)

Оберіть оцінку

Рисунок 3.7 – Анкета оцінювання курсу

Надання відповіді для запитань з шкалою від 1 до 5 побудовано у вигляді випадального списку, як зображено на Рисунку 3.8 – Формат надання відповіді.

					ІАЛЦ.467200.003 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

1. Викладач зрозуміло пояснював теоретичний матеріал. (1-5)

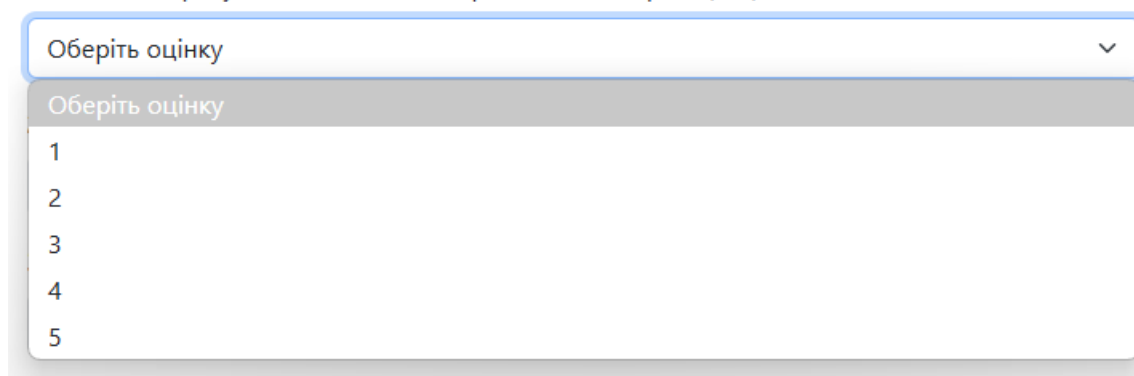


Рисунок 3.8 – Формат надання відповіді

Нижче представлена форма з питаннями, згрупованими за категоріями.

Наприклад:

Якість викладання курсу:

1. Викладач зрозуміло пояснював теоретичний матеріал. (1–5)

2. Викладач доступно відповідав на запитання студентів. (1–5)

Актуальність курсу:

3. Тематика курсу є сучасною і відповідає вимогам галузі. (1–5)

4. Курс був корисним для розуміння інших дисциплін. (1–5)

Інтерактивність курсу:

5. Курс містив інтерактивні елементи в навчання. (1-5)

6. Під час занять використовувались практичні приклади, демонстрації або інтерактивні завдання. (1–5)

7. Я був(ла) залучений(на) до обговорень або активностей на заняттях. (1–5)

Складність та навантаження курсу:

8. Навчальне навантаження (домашні завдання, проєкти) було помірним і посильним. (1–5)

9. Курс виявився для мене складним. (1–5) (1 - “зовсім не складним”, 5 – “надзвичайно складним”).

Загальна оцінка курсу:

10. Я отримав(ла) нові навички після проходження цього курсу (1-5)

					ІАЛЦ.467200.003 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

11. Загалом я задоволений(на) якістю цього курсу. (1–5)

12. Я б рекомендував(ла) цей курс іншим студентам. (1–5)

Оцінювання курсу охоплює кілька ключових аспектів, що дозволяють комплексно зрозуміти його якість та ефективність. Зокрема, враховується, наскільки викладач зрозуміло і доступно подає теоретичний матеріал та відповідає на запитання студентів, що відображає загальний рівень викладання. Окремо оцінюється актуальність курсу — його відповідність сучасним вимогам галузі й користь для опанування інших дисциплін. Важливою складовою є інтерактивність: аналізується наявність практичних завдань, прикладів і залучення студентів до обговорень, що сприяє активному навчанню. Не менш важливим є й баланс між складністю курсу та навчальним навантаженням — чи є завдання посилюючими і наскільки загалом курс був складним для студентів. Завершує оцінювання загальне враження: чи отримали студенти нові навички, чи задоволені якістю курсу і чи рекомендували б його іншим. Такий підхід дозволяє отримати цілісну картину навчального досвіду.

Визначення питань в такому форматі полегшує сприйняття студентом та підкреслює ключові моменти, що стосуються проведення опитувань задоволеності курсами, а групування за категоріями такими як якість викладання курсу, актуальність, інтерактивність та складність курсу чітко чітко описує ціль кожного з запитань дізнатися справжній настрій студентів після проходження курсу.

Після заповнення всіх відповідей на запитання, студент має можливість відправити свої відповіді. У випадку незаповнення всіх місць для надання відповіді, система вкаже на порожні колонки.

					ІАЛЦ.467200.003 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

3. Тематика курсу є сучасною і відповідає вимогам галузі. (1-5)

4

4. Курс був корисним для розуміння інших дисциплін. (1-5)

5

5. Курс містив інтерактивні елементи в навчанні. (1-5)

4

6. Під час занять використовувались практичні приклади, демонстрації або інтерактивні завдання. (1–5)

5

7. Я був(ла) залучений(на) до обговорень або активностей на заняттях. (1–5)

4

8. Навчальне навантаження (домашні завдання, проекти) було помірним і посильним. (1–5)

5

9. Курс виявився для мене складним. (1 - "зовсім не складним", 5 – "надзвичайно складним").

5

10. Я отримав(ла) нові навички після проходження цього курсу. (1-5)

4

11. Загалом я задоволений(на) якістю цього курсу. (1–5)

4

12. Я б рекомендував(ла) цей курс іншим студентам. (1–5)

5

Відправити відповіді

Рисунок 3.9 – Заповнена анкета оцінювання курсу

Після надсилання заповненої анкети система підтвердить надсилання даних та подякує студенту за участь у покращенні якості навчання університету, це зображено на Рисунку 3.10 – Підтвердження надсилання даних. Додатково, студент перенаправиться назад на сторінку з курсами.

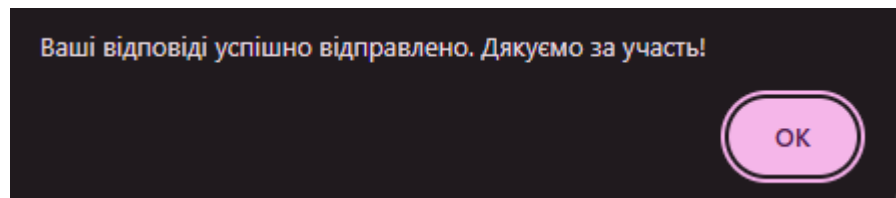


Рисунок 3.10 – Підтвердження надсилання даних

Тепер перейдемо до того, що може зробити адміністратор.

Вхід до системи

Email:

Пароль:

Увійти

Рисунок 3.11 – Вхід у якості адміністратора

При виконанні входу у якості адміністратора та введенні коректних даних до нього, змінюється привітання на головній сторінці та з'являється перехід до панелі адміністратора.

Вітаємо, Адміністратор!

Перейдіть до панелі адміністратора, щоб керувати курсами.

Відкрити панель адміністратора

Рисунок 3.12 – Головна сторінка адміністратора

					ІАЛЦ.467200.003 ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

Панель адміністратора передбачає у собі список всіх наявних курсів, подібно до студенту, проте, замість кнопки з оцінюванням курсу адміністратор має можливість видалити курс. Додатково, передбачається можливість для створення нового курсу.

Панель адміністратора

+ Створити новий курс	
Алгоритми та структури даних (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Комп'ютерна теорія (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Психологія розробника (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Програмування мікропроцесорних систем (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Інженерія соціального побуту (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Теорія прийняття рішень (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Системна розробка програмного забезпечення (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Машинне навчання та штучний інтелект (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Кібербезпека та захист даних (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Розробка мобільних додатків (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Комп'ютерна графіка та візуалізація (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Хмарні технології та DevOps (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Бази даних та аналіз великих даних (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Розробка ігор та віртуальна реальність (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити
Блокчейн та криптографія (Комп'ютерні науки, 3 курс)	Переглянути звіт Видалити

Рисунок 3.13 – Панель адміністратора

Створення нового курсу передбачає у собі заголовок, визначення освітньо-навчальної програми, курсу навчання, назви курсу та всіх питань, що будуть знаходитись в анкеті.

Створення нового курсу

Освітня програма:

Курс (рік навчання):

Назва курсу (дисципліни):

Питання анкети (кожне з нового рядка):

Створити курс

Рисунок 3.14 – Створення нового курсу

Для тестового прикладу створюється новий курс з освітньою програмою «Комп’ютерна інженерія», курс навчання - 4. Назвою курсу є «Програмне забезпечення для аналітичних платформ», що поєднує в собі анкети, опитування та збір статистики. Конкретно в цьому прикладі створимо 12 різноманітних питань, які стосуються доступного пояснення матеріалу, сучасності даного курсу, відповідності вимогам галузі, наявності інтерактивних елементів та достатньої кількості прикладів, складності курсу та інших питань, що стосуються курсу. Більша кількість питань допоможе краще побачити загальну картину задоволеності курсу студентами, актуальності матеріалу, доступності його викладання.

					ІАЛЦ.467200.003 ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

Створення нового курсу

Освітня програма:

Комп'ютерна інженерія

Курс (рік навчання):

4 курс

Назва курсу (дисципліни):

Програмне забезпечення для аналітичних платформ

Питання анкети (кожне з нового рядка):

1. Викладач зрозуміло пояснював теоретичний матеріал. (1-5)
2. Викладач доступно відповідав на запитання студентів. (1-5)
3. Тематика курсу є сучасною і відповідає вимогам галузі. (1-5)
4. Курс був корисним для розуміння інших дисциплін. (1-5)
5. Курс містив інтерактивні елементи в навчанні. (1-5)
6. Під час занять використовувались практичні приклади, демонстрації або інтерактивні завдання. (1-5)
7. Я був(ла) залучений(на) до обговорень або активностей на заняттях. (1-5)
8. Навчальне навантаження (домашні завдання, проекти) було помірним і посилює. (1-5)
9. Курс виявився для мене складним. (1 - "зовсім не складним", 5 – "надзвичайно складним").
10. Я отримав(ла) нові навички після проходження цього курсу. (1-5)
11. Загалом я задоволений(на) якістю цього курсу. (1-5)
12. Я б рекомендував(ла) цей курс іншим студентам. (1-5)



Створити курс

Рисунок 3.15 – Створення нового курсу

Після заповнення всіх необхідних даних для створення нового курсу «Програмне забезпечення для аналітичних платформ» та підтвердження створення курсу цей курс відобразиться як студентам, так і в панелі адміністратора.

Програмне забезпечення для аналітичних платформ (Комп'ютерна інженерія, 1 курс)

Видалити

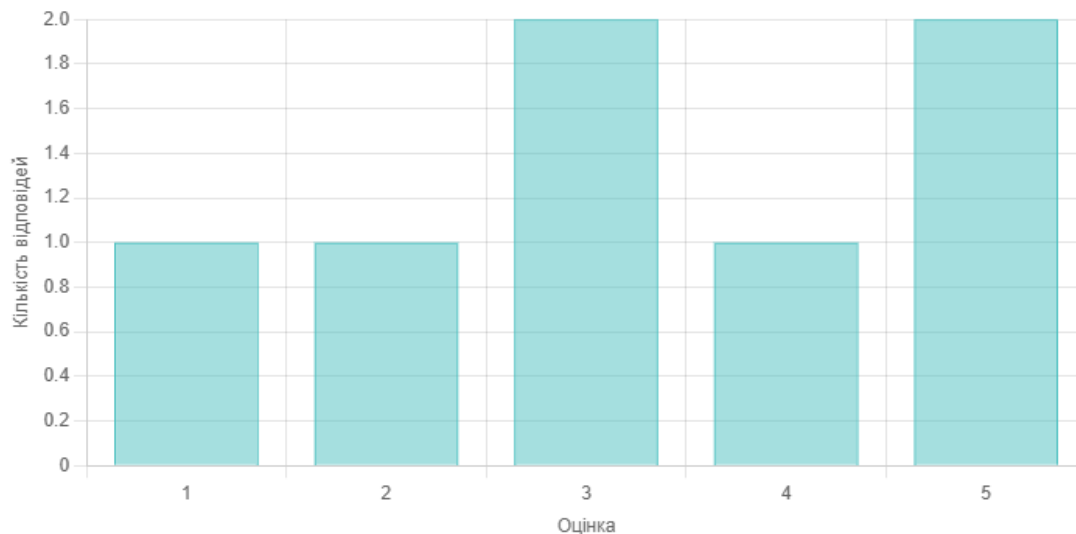
Рисунок 3.16 – Створений курс

Адміністратор має можливість створювати аналітику для кожного з курсів для того, щоб отримати результати та мати можливість оприлюднити висновки оцінювання. Далі, ці результати можуть передаватися до викладачів з ціллю покращити якість викладання курсу або відзнаки найкращих викладачів за прекрасну роботу. Адміністратор має можливість переглянути звіт та отримати результати кожного з питань окремо.

					ІАЛЦ.467200.003 ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

Статистика відповідей

Алгоритми та структури даних: 1. Викладач зрозуміло пояснював теоретичний матеріал. (1-5)



Алгоритми та структури даних: 2. Викладач доступно відповідав на запитання студентів. (1-5)

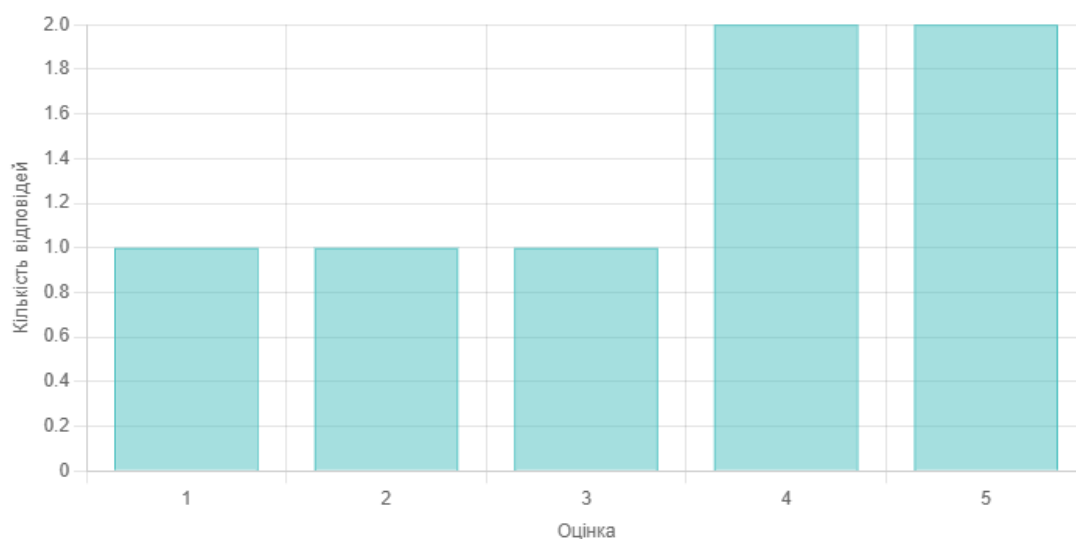


Рисунок 3.17 – Аналітика для кожного з запитань курсу

Адміністратор має змогу отримати загальний результат задоволеності курсом. Він підраховується в залежності від всіх відповідей учасників опитування та дає змогу легко порівняти, який з курсів показав кращий результат. Для розрахунку використовується описана в математичній моделі формула знаходження середньої задоволеності курсом.

Загальний рівень задоволеності курсами



Рисунок 3.18 – Загальний рівень задоволеності курсами

Платформа має перевірку повторного голосування студента, щоб не портити статистику курсу. У випадку, якщо студент вже надіслав заповнені анкету опитування, то на місці кнопки для проходження буде надпис «Вже пройдено».

Алгоритми та структури даних (Комп'ютерні науки, 3 курс)

Вже пройдено

Рисунок 3.19 – Перевірка на повторне голосування

Кожен студент має лише одну спробу для заповнення анкети, щоб не створювати дисбаланс на статистику та зробити процес чесним та прозорим. Після заповнення анкети на місці кнопки «Оцінити» буде надпис «Вже пройдено», як зображено на рисунку 3.20 – Пройдений студентом курс.

Таким чином, забезпечено реалізацію всього однієї спроби для кожного курсу, для якого студент має можливість пройти опитування з задоволеності курсом.

Вже пройдено

Рисунок 3.20 – Пройдений студентом курс

Враховуючи, що основною цілю платформи є не тільки збір даних про настрої студентів після проходження курсів, а й анонімність, реалізована наступна логіка. Кожен студент має лише одну спробу для проходження опитування, проте, у випадку допущення помилок при проходженні і заповненні студент повинен мати можливість очистити спробу. Для цього в панелі адміністратора є можливість вписати пошту студента, яку він сам може надати для видалення спроби. Це проілюстровано на рисунку 3.21 – Керування спробами користувачів.

Керування спробами користувачів

Введіть email користувача:

Отримати оцінені курси

Цей користувач ще не оцінював жоден курс.

Рисунок 3.21 – Керування спробами користувачів

Далі, адміністратор може ввести надану студентом пошту та побачити всі спроби студента. Важливим моментом є те, що викладач або адміністратор не бачать які відповіді залишив студент, лише наявність спроби. Це забезпечує повну анонімність студентів.

					ІАЛЦ.467200.003 ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

Керування спробами користувачів

Введіть email користувача:

d@gmail.com

Отримати оцінені курси

213123

Скинути спробу

Рисунок 3.22 – Керування спробами студента з поштою d@gmail.com

3.2 Результати тестування розробленої системи

Важливою частиною роботи є перевірка правильності та надійності розробленої платформи для аналітичного скорингу освітньо-професійних програм.

В ході тестування платформи акцентовано увагу на коректній роботі платформи, відсутність наявних проблем при роботі користувачів (студентів та адміністраторів).

У випадку надання некоректних даних, наприклад, незаповнення частини з обов'язкових полів для оцінювання система автоматично вказує на наявність проблем при заповненні. Після виправлення всіх помилок користувачем, система повторно виконує перевірку та приймає у випадку коректності даних.

У випадку введення некоректних даних для авторизації у систему платформи, користувач отримує повідомлення проблем при авторизації, вказуючи на потенційну помилку у даних для авторизації. Після виправлення помилок система пропустить користувача, якщо будуть введені правильні дані.

					ІАЛЦ.467200.003 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.1 – Проведені тести для платформи з аналітичного скорингу

Тип тестування	Проведено	Коментар
Юніт-тести	+	Перевірка окремих функцій, зокрема <code>handleSubmit</code> (авторизація)
Інтеграційні тести	+	Перевірка взаємодії модулів: логін, перенаправлення, рольові права
Системне тестування	+	Тестування усієї платформи на відповідність вимогам
Тестування інтерфейсу користувача (UI)	+	Перевірка повідомлень про помилки, обробка форм, зручність взаємодії
Тестування обробки помилок	+	Некоректні поля, неправильний логін, повідомлення про помилки
Тестування ролей користувача	+	Розмежування прав між студентами та адміністраторами
Безпекове тестування	+	Проведено безпекові тести, що стосуються JWT токенів

Є наявність прав користувачів, які розділяють їх на студентів та адміністраторів. Всі описані вимоги для платформи виконані в повному обсязі та дозволяють студентам авторизуватися, обирати курси для оцінювання, відповідати на запитання та підтверджувати відповіді. Для адміністратора є повна можливість створення нових анкет для опитування, видалення поточних анкет, отримання результатів опитування. Всі надані студентами відповіді є повністю анонімними та не репресуються з боку платформи.

3.3 Програмна документація користувача

Завершальною частиною проекту є підготовка документації – як для кінцевих користувачів (адміністраторів, викладачів, студентів), так і для розробників, які будуть підтримувати або розвивати систему. Нижче наведено ключові положення інструкцій користувача, а також коротко – для адміністратора системи (див. рисунок 3.7).

Інструкція для студента (користувача опитування)

Доступ до анкети: студент отримує повідомлення (на електронну пошту або в навчальному кабінеті) зі посиланням на опитування або інформацію про

те, де його знайти (наприклад, в кінці семестру на сайті кафедри). Перейшовши за посиланням, студент бачить список курсів, за якими потрібно дати відгук (звичайно, тих, які він прослухав у цьому семестрі). Треба обрати курс і натиснути “Розпочати опитування”.

Заповнення анкети: на сторінці опитування студент послідовно відповідає на всі запитання. Потрібно оцінювати щиро і об’єктивно, пам’ятаючи, що опитування анонімне – викладачі не побачать, хто як відповів, а результати будуть використовуватись лише в узагальненому вигляді. Якщо якесь запитання незрозуміле – можна запитати координатора (але зазвичай формулювання прості).

Завершення: після натискання кнопки “Відправити” слід дочекатися повідомлення про успішне збереження. Якщо раптом є обов’язкове питання без відповіді, система підсвітить його червоним – потрібно заповнити і повторно відправити. На одну анкету відведено близько 10 хвилин; якщо не вкладатися – можна зберегти частково (наша система наразі не підтримує чернетки, тож краще виділити час і зробити за раз).

Конфіденційність: ні ім’я студента, ні будь-які персональні дані не прив’язуються до відповідей. Якщо якісь питання студент не хоче розголошувати навіть анонімно (наприклад, писати негативний коментар) – це його право, але адміністрація гарантує, що репресій за критику не буде, навпаки – цінуються конструктивні відгуки.

Інструкція для адміністратора системи

Додавання навчальних даних: Перед початком семестру або опитування адміністратор через інтерфейс вводить назви освітніх програм, перелік курсів, прізвища викладачів, якщо потрібно.

Створення анкети: через інтерфейс адміністратор формує список запитань. Можна змінювати порядок запитань. Визначити шкалу (в нашому випадку скрізь 1–5, тому за замовчуванням виставлено).

					ІАЛЦ.467200.003 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

Запуск опитування: коли настає час адміністратор активує опитування – натискає “Відкрити опитування”. Це робить анкету доступною для студентів. Далі адміністратор може розіслати повідомлення студентам або попросити кураторів груп довести інформацію. Статус опитування відображається – скільки відповідей надійшло.

Закриття опитування: по завершенні адміністратор закриває форму. В цей момент студенти більше не можуть надсилати відповіді.

Розповсюдження результатів: адміністратор може передати статистику викладачам, обговорити на засіданні кафедри. В деяких випадках узагальнену інформацію можна донести до студентів (наприклад, “ми почули ваші відгуки, ось що збираємося змінити...” – це підвищить довіру до опитувань).

Технічне обслуговування: Адміністратор (або ІТ-відділ) також має стежити за самою системою: робити бекапи бази (у нас реалізовано щоденний автоматичний бекап), слідкувати за оновленнями безпеки (Node.js, React). Якщо потрібно – додавати нових користувачів (якщо є логін) або змінювати налаштування сервера (наприклад, серверна адреса, порт – це описано в Readme системи).

Документація розробника: В рамках дипломної роботи підготовлено окремий технічний документ (у додатках), який містить:

Опис архітектури (діаграми, які наведено в розділі 2).

Інструкції по розгортанню проекту: які потрібні залежності (Node.js 16+, React) які команди виконати для запуску (наприклад: `npm install` для бекенду, `ng build` для фронтенду, налаштувати файл конфігурації `.env` із параметрами БД, потім `npm start` і `ng serve`).

Опис можливих подальших поліпшень (з точки зору коду): наприклад, рефакторинг деяких частин, додавання кешування результатів опитування щоб не рахувати заново кожного разу.

Підсумково про документацію: наявність зрозумілої документації – один із критеріїв якості програмного продукту. У нашому випадку, керуючись

					ІАЛЦ.467200.003 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

методичними вимогами до дипломної роботи, ми оформили пояснювальну записку (цю роботу) та доповнили її графічними матеріалами і текстовими інструкціями користувача. Це забезпечує можливість передати проект іншому фахівцю для підтримки або розгортання в експлуатацію без необхідності глибоко розбиратися в коді з нуля.

Розроблена платформа може бути корисною не лише в нашому університеті. За наявності документації її можна адаптувати під інші кафедри чи навчальні заклади. Конфігурація опитувань доволі гнучка, а код – модульний. Отже, дипломний проект має перспективу розвитку у вигляді реального інструменту забезпечення якості освіти.

					ІАЛЦ.467200.003 ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 3

У третьому розділі дипломного проєкту «Програмно-апаратна платформа з аналітичного скорингу освітньо-професійних програм та освітніх компонент» продемонстровано практичні аспекти застосування розробленої системи, її функціональні можливості, а також результати тестування.

Демонстрація роботи платформи:

Розроблена система забезпечує повний цикл аналітичного скорингу: від збору анонімних відгуків студентів через зручний веб-інтерфейс до обробки даних та генерації звітів.

Інтерфейс для студентів дозволяє легко авторизуватися, обирати курси для оцінювання, заповнювати анкети з питаннями за шкалою 1–5 та отримувати підтвердження надсилання відповідей.

Панель адміністратора надає можливості створення, редагування та видалення курсів, а також перегляду аналітичних звітів. Система підтримує інтуїтивне управління та адаптивний дизайн на основі Bootstrap.

Продемонстровано приклад оцінювання курсу «Алгоритми та структури даних» на основі тестових даних, що моделюють реальні відгуки 47 студентів, з питаннями, згрупованими за критеріями (якість викладання, актуальність, інтерактивність, складність, загальна оцінка).

Документація користувача:

Підготовлено детальні інструкції для студентів (як заповнювати анкети) та адміністраторів (як створювати курси, запускати опитування, генерувати звіти та забезпечувати технічне обслуговування).

Документація для розробників включає опис архітектури, структури бази даних, інструкції з розгортання (залежності, конфігурація) та рекомендації щодо подальшого вдосконалення системи.

Наявність документації забезпечує легкість підтримки, масштабування та адаптації платформи для інших навчальних закладів.

					ІАЛЦ.467200.003 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

Результати тестування:

Тестування підтвердило коректну роботу платформи: система обробляє запити в реальному часі, забезпечує анонімність відповідей, правильно валідує введені дані та видає повідомлення про помилки (наприклад, при незаповненні обов'язкових полів чи некоректній авторизації).

Ролі користувачів (студенти, адміністратори) чітко розмежовані, що гарантує безпечний доступ до функціоналу.

Усі функціональні вимоги, включаючи створення анкет, обробку даних, генерацію звітів і забезпечення анонімності, виконані в повному обсязі.

Практична цінність:

Платформа є гнучким і модульним рішенням, яке може бути адаптоване для інших кафедр чи університетів.

Реалізована система сприяє підвищенню якості освітнього процесу шляхом автоматизації збору та аналізу зворотного зв'язку, що дозволяє виявляти слабкі сторони курсів і приймати обґрунтовані управлінські рішення.

Таким чином, розроблена платформа відповідає поставленим цілям, забезпечує зручність використання, надійність і потенціал для подальшого розвитку, що робить її цінним інструментом для забезпечення якості освіти.

					ІАЛЦ.467200.003 ПЗ	Арк.
						53
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В результаті дипломного проєкту було розроблено веб-додаток для аналітичного скорингу освітньо-професійних програм та освітніх компонент. Завдяки фреймворку React було реалізовано повноцінну модель платформи, де наявні чітке розмежування прав користувачів, можливість проведення аналітичних опитувань та отримання звіту і статистики щодо кожного з опитувань.

По технічному завданню по пунктах реалізовано проведення аналітичного скорингу отриманих оцінок. Система дає можливість надіслати відповідь на анкету, створену адміністратором та отримати детальні графіки, що описують всі необхідні дані.

Забезпечено виведення графічних візуалізацій, таких як графіки у зручному та привабливому дизайні з коректною обробкою даних та відображенням їх для користувача.

Реалізовано кросбраузерну підтримку застосунку та адаптивний інтерфейс, що дозволяє переглядати платформу при різних розмірах екрану та на девайсах різних моделей, таких як Apple, Samsung, та інші.

Втілено модульність та легку підтримку коду. Структура проєкту є зрозумілою, ділиться на такі папки як frontend та backend, зміст цих папок є структурованим та зрозумілим для розробника. Код є чистим та відповідає визначеним вимогам написання коду. Платформа має властивість масштабування та може легко бути покращена з додаванням нових інструментів, таких як перехід на потужнішу базу даних, тощо.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бахрушин В. Є. Методи аналізу даних. – Запоріжжя, 2011. – С. 268.
2. Василенко О. А., Сенча І. А. Математично-статистичні методи аналізу у прикладних дослідженнях. – Одеса, 2011 – С. 166.
3. Караєва Н. В. WEB- та мобільні технології в економіці [Електронний ресурс] // НТУУ «КПІ ім. Ігоря Сікорського» – Режим доступу до ресурсу: <https://ela.kpi.ua/server/api/core/bitstreams/7b3d0f26-a3fe-4e1b-a1f6-97a76c43dc5a/content> (дата звернення: 02.05.2025)
4. Ланде Д. В., Субач І. Ю., Бояринова Ю. Є. Основи теорії і практики інтелектуального аналізу даних у сфері кібербезпеки. [Електронний ресурс] // НТУУ «КПІ ім. Ігоря Сікорського» – Режим доступу до ресурсу: <https://ela.kpi.ua/server/api/core/bitstreams/7b3d0f26-a3fe-4e1b-a1f6-97a76c43dc5a/content> (дата звернення: 03.05.2025)
5. Марченко О. О., Россада Т. В. Актуальні проблеми Data Mining. [Електронний ресурс] // НТУУ «КПІ ім. Ігоря Сікорського» – Режим доступу до ресурсу: http://csc.knu.ua/media/filer_public/38/03/3803002b-e068-4a08-8a6c-a4edc183892a/datamining20170917.pdf (дата звернення: 03.05.2025)
6. Міністерство освіти і науки України. Національна рамка кваліфікацій. – Київ, 2021 – С. 100.
7. Banks A., Porcello E. Learning React: modern patterns for developing React apps (2nd ed.) – Sebastopol, 2020 – P. 310.
8. Bishop C. M. Pattern Recognition and Machine Learning. – Springer, 2006 – P. 738.
9. Dietrich D., Heller B., Yang B. Data Science & Big Data Analytics: discovering, analyzing, visualizing and presenting data – Indianapolis, 2015 – P. 432.
10. Field A. Discovering Statistics Using IBM SPSS Statistics (5th ed.). – Los Angeles, 2018 – P. 1104.

					ІАЛЦ.467200.003 ПЗ	Арк.
						55
Зм.	Арк.	№ докум.	Підпис	Дата		

11. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: elements of reusable object-oriented software. – Boston, 1994 – P. 395.
12. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: data mining, inference, and prediction (2nd ed.). – New York, 2009 – P. 745.
13. Likert R. Technique for the Measurement of Attitudes. – 1932 – P. 55.
14. Mandel T. The Elements of User Interface Design. – New York, 2005 – P. 400.

					ІАЛЦ.467200.003 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

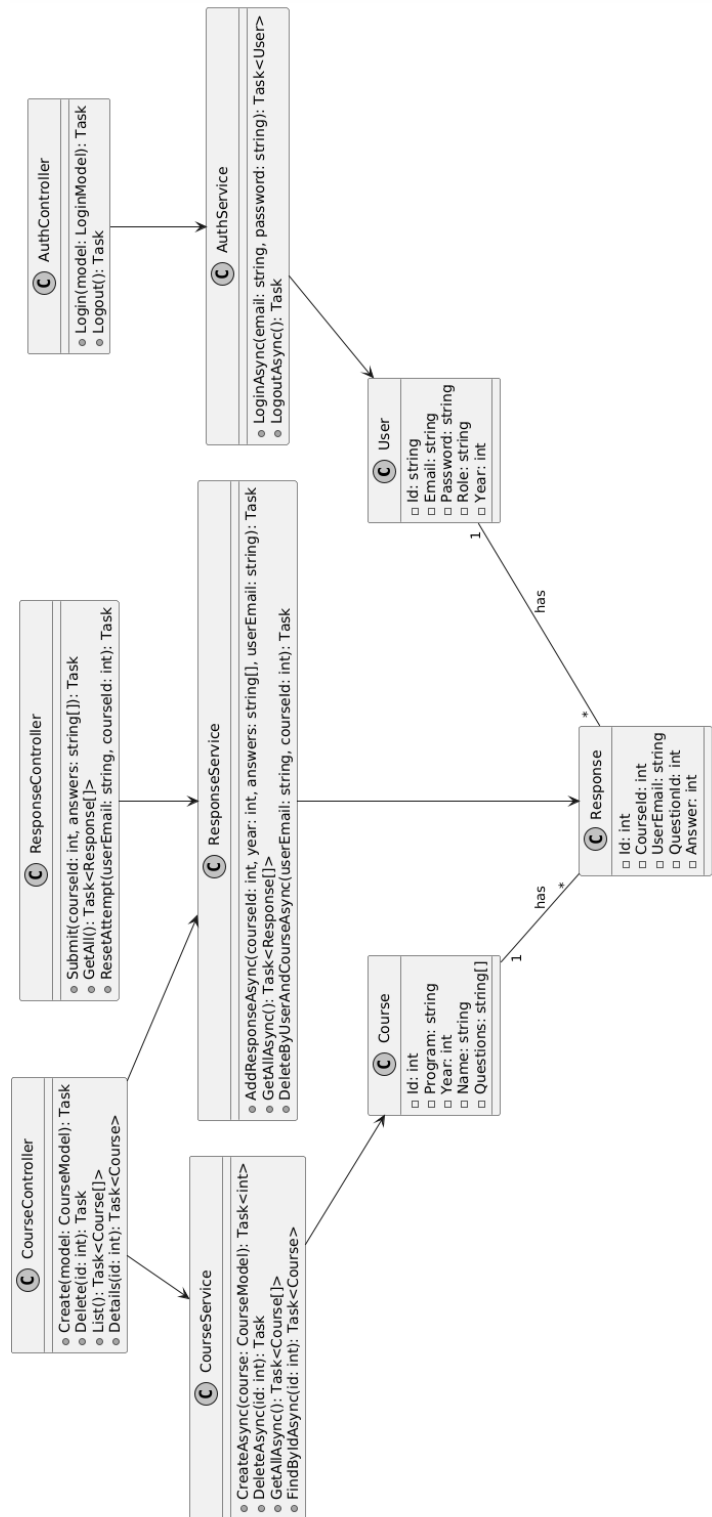
ДОДАТОК А

Програмно-апаратна платформа з аналітичного скорингу
освітньо-професійних програм та освітніх компонент

Структура системи
ІАЛЦ.467200.004 ДА

Аркушів 1

Київ 2025 р.



					ІАЛЦ.467200.004 ДА			
		№ докум.	Підпис	Дата				
Розробив	Козаренко Д. О.				Програмно-апаратна платформа з аналітичного скорингу освітньо-		Літ.	Аркуш
Перевірів	Туганських О.А							1
							НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІО-13	
Н. Контр.	Міщенко Л.Р.							
Затвердив								

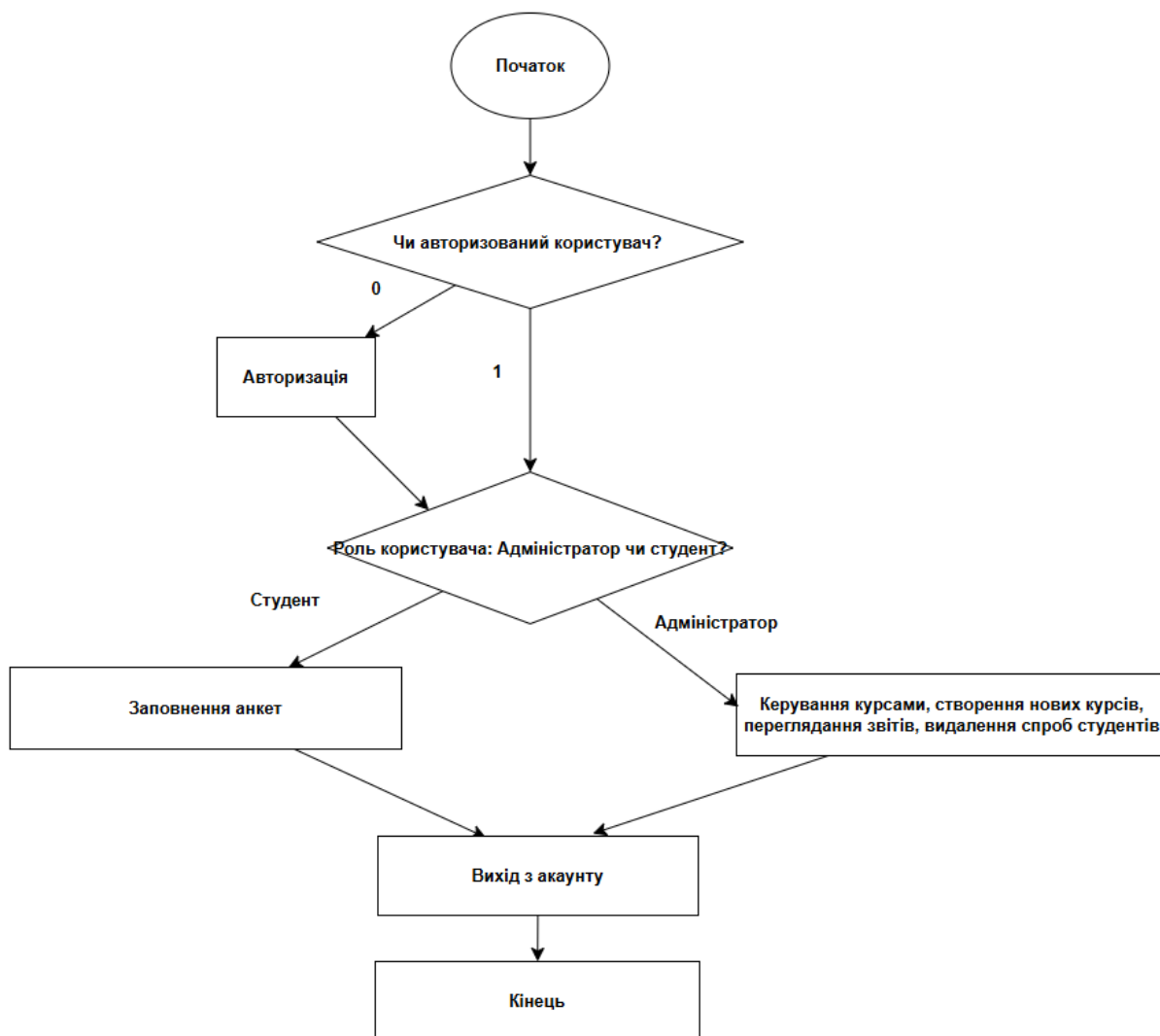
ДОДАТОК Б

Програмно-апаратна платформа з аналітичного скорингу
освітньо-професійних програм та освітніх компонент

Принципова схема системи
ІАЛЦ.467200.005 ДБ

Аркушів 1

Київ 2025 р



					ІАЛЦ.467200.005 ДБ						
		№ докум.	Підпис	Дата							
Розробив		Козаренко Д. О.			Програмно-апаратна платформа з аналітичного скорингу Принципова схема			Літ.	Аркуш	Аркушів	
Перевірив		Туганських О.А								1	1
								НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІО-13			
Н. Контр.		Міщенко Л. Д.									
Затвердив											

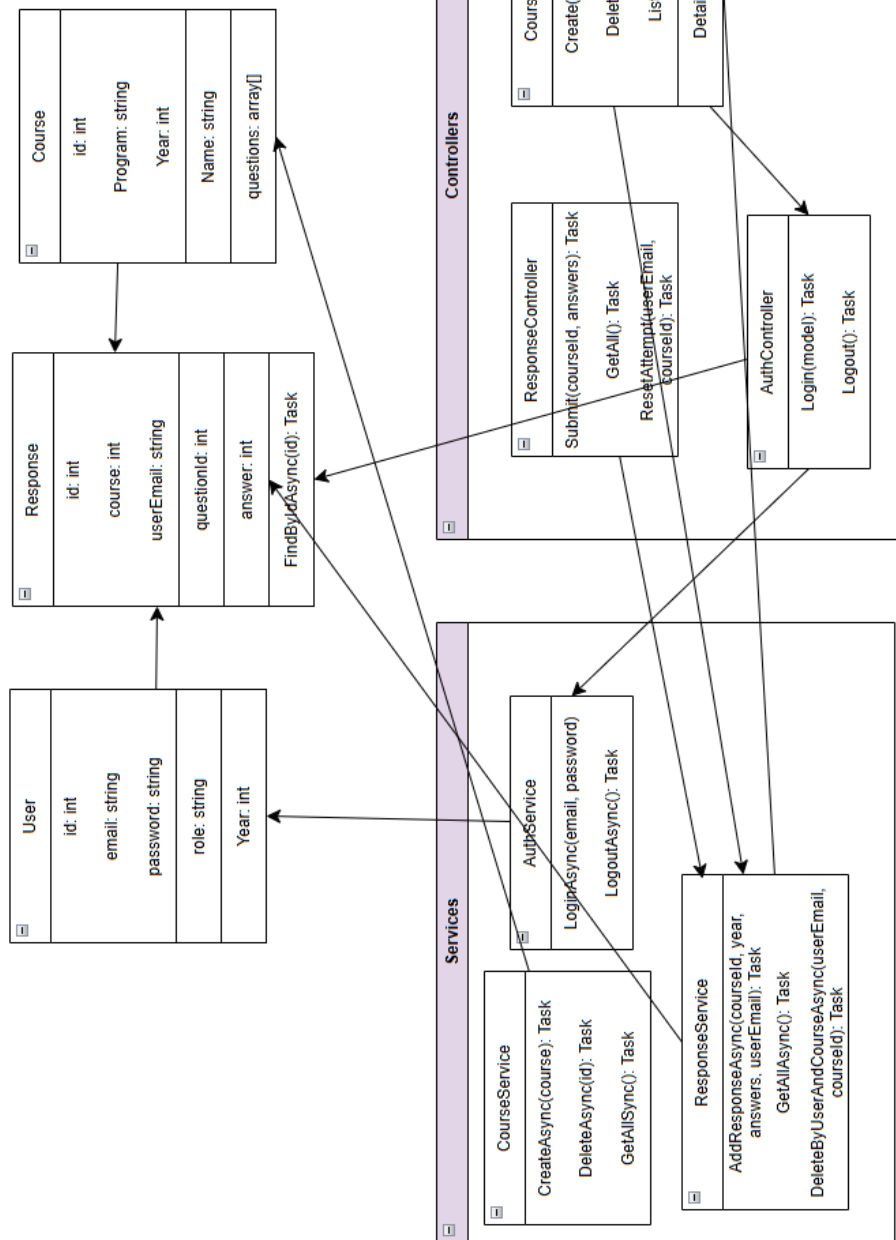
ДОДАТОК В

Програмно-апаратна платформа з аналітичного скорингу
освітньо-професійних програм та освітніх компонент

Діаграма класів
ІАЛЦ.467200.006 ДВ

Аркушів 1

Київ 2025 р



					ІАЛЦ.467200.006 ДВ		
		№ докум.	Підпис	Дата			
Розробив		Козаренко Д. О.			Програмно-апаратна платформа з аналітичного скорингу Діаграма класів	Літ.	Аркуш
Перевірив		Туганських О.А				1	1
Н. Контр.		Міщенко Л. Д.				НТУУ КПІ ім. Ігоря Сікорського, ФІОТ, ІО-13	
Затвердив`							

ДОДАТОК Г

Програмно-апаратна платформа з аналітичного скорингу
освітньо-професійних програм та освітніх компонент

ТЕКСТ ПРОГРАМИ
ІАЛЦ.467200.007 ДГ

Аркушів 22

Київ 2025 р

```

import { useContext, useState } from 'react';
import { AuthContext } from '../context/AuthContext';
import { CourseContext } from '../context/CourseContext';
import { Navigate, Link } from 'react-router-dom';
import { Bar } from 'react-chartjs-2';
import 'chart.js/auto';
import 'bootstrap/dist/css/bootstrap.min.css';

const AdminPanel = () => {
  const { user } = useContext(AuthContext);
  const { courses, deleteCourse, responses } = useContext(CourseContext);
  const [expandedCourses, setExpandedCourses] = useState({});
  const [manageEmail, setManageEmail] = useState('');
  const [userEvaluatedCourses, setUserEvaluatedCourses] = useState([]);

  if (!user) return <Navigate to="/login" replace />;
  if (user.role !== 'admin') return <Navigate to="/" replace />;

  const toggleCourseReport = (courseId) => {
    setExpandedCourses(prev => ({
      ...prev,
      [courseId]: !prev[courseId]
    }));
  };

  const fetchUserEvaluatedCourses = () => {
    const evaluatedCourseIds = [...new Set(
      responses.filter(res => res.userEmail === manageEmail).map(res =>
res.courseId)
    )];

    const evaluatedCourses = evaluatedCourseIds.map(courseId => ({
      id: courseId,
      name: courses.find(c => c.id === courseId)?.name || 'Невідомий
курс'
    }));
  };

```

					ІАЛЦ.467200.007 ДГ								
		№ докум.	Підпис	Дата									
Розробив		Козаренко Д. О.			Програмно-апаратна платформа з аналітичного скорингу освітньо-			Літ.		Аркуш		Аркушів	
Перевірив		Туганських О.А								1		55	
								НТУУ КПІ ім. Ігоря					
Н. Контр.		Міщенко Л.Р.						Сікорського, ФІОТ, ІО-13					
Затвердив													

```

        method: 'DELETE',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ userEmail: manageEmail, courseId })
    });

    const result = await res.json();
    if (result.success) {
        setUserEvaluatedCourses(prev => prev.filter(c => c.id !==
courseId));
        alert(`Спроба для курсу видалена (${result.removed}
відповідей).`);
    } else {
        alert("Помилка під час видалення спроби.");
    }
};

// Обчислення середніх оцінок
const courseAverages = {};
courses.forEach(course => {
    const courseResponses = responses.filter(res => res.courseId ===
course.id);
    if (courseResponses.length > 0) {
        const allAnswers = courseResponses.map(res =>
Number(res.answer));
        const sum = allAnswers.reduce((acc, val) => acc + val, 0);
        courseAverages[course.id] = sum / allAnswers.length;
    } else {
        courseAverages[course.id] = null;
    }
});

const sortedCourses = courses
    .map(course => ({
        id: course.id,
        name: course.name,
        average: courseAverages[course.id]
    }))
    .filter(course => course.average !== null)
    .sort((a, b) => b.average - a.average);

```

					ІАЛЦ.467200.007 ДГ	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

```

const satisfactionChartData = {
  labels: sortedCourses.map(course => course.name),
  datasets: [{
    label: 'Середній рівень задоволеності',
    data: sortedCourses.map(course => course.average),
    backgroundColor: 'rgba(153, 102, 255, 0.5)',
    borderColor: 'rgba(153, 102, 255, 1)',
    borderWidth: 1
  }]
};

// Частотний аналіз
const freqByCourse = {};
responses.forEach(res => {
  const { courseId, questionId, answer } = res;

  if (!freqByCourse[courseId]) freqByCourse[courseId] = {};
  if (!freqByCourse[courseId][questionId]) {
    freqByCourse[courseId][questionId] = {
      questionText: courses.find(c => c.id ===
courseId)?.questions[questionId] || `Питання ${questionId + 1}`,
      answers: {}
    };
  }
  freqByCourse[courseId][questionId].answers[answer] =
(freqByCourse[courseId][questionId].answers[answer] || 0) + 1;
});

return (
  <div className="container mt-4">
    <h2>Панель адміністратора</h2>
    <Link to="/admin/create" className="btn btn-primary mb-3">+
Створити новий курс</Link>

    {courses.map(course => (
      <div key={course.id} className="list-group-item border mb-
3 p-3 rounded shadow-sm">
        <div className="d-flex justify-content-between align-
items-center mb-2">
          <div>
            <strong>{course.name}</strong>
            <em>({course.program}, {course.year} курс)</em>
          </div>

```

					ІАЛЦ.467200.007 ДГ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

```

<div className="d-flex gap-2">
  <button
    className="btn btn-primary btn-sm"
    onClick={() =>
toggleCourseReport(course.id)}
  >
    {expandedCourses[course.id] ? 'Приховати
звіт' : 'Переглянути звіт'}
  </button>
  <button
    className="btn btn-danger btn-sm"
    onClick={() => deleteCourse(course.id)}
  >
    Видалити
  </button>
</div>
</div>

{expandedCourses[course.id] && (
  <div>
    {(freqByCourse[course.id] &&
Object.entries(freqByCourse[course.id]).map(([questionId, { questionText,
answers }]) => {
      const labels =
Object.keys(answers).sort((a, b) => Number(a) - Number(b));
      const counts = labels.map(label =>
answers[label]);

      return (
        <div key={questionId} className="mb-
4">
          <h6>{questionText}</h6>
          <Bar
            data={{
              labels,
              datasets: [{
                label: 'Кількість
відповідей',
                data: counts,
                backgroundColor:
'rgba(75, 192, 192, 0.5)',
                borderColor: 'rgba(75,
192, 192, 1)',

```

					ІАЛЦ.467200.007 ДГ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

```

borderWidth: 1
    ]]
  }}
  options={{
    scales: {
      y: { beginAtZero:
true, title: { display: true, text: 'Кількість відповідей' } },
      x: { title: { display:
true, text: 'Оцінка' } }
    },
    plugins: { legend: {
display: false } }
  }}
  />
</div>

);
))) || <p>Немає відповідей для цього
курсу.</p>
</div>
  )}
</div>
)})

<h3 className="mt-5">Загальний рівень задоволеності
курсами</h3>
{sortedCourses.length > 0 ? (
  <div className="mb-5">
    <Bar
      data={satisfactionChartData}
      options={{
        scales: {
          y: { beginAtZero: true, max: 5, title: {
display: true, text: 'Середній бал' } },
          x: { title: { display: true, text: 'Курси'
} }
        },
        plugins: { legend: { display: false } }
      }}
    />
  </div>
) : (
  <p>Немає даних для відображення рівня задоволеності.</p>
)}

```

					ІАЛЦ.467200.007 ДГ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

```

<h3 className="mt-5">Керування спробами користувачів</h3>
<div className="mb-3">
  <label className="form-label">Введіть email
користувача:</label>
  <input
    type="email"
    className="form-control"
    value={manageEmail}
    onChange={ (e) => setManageEmail (e.target.value) }
  />
  <button className="btn btn-primary mt-2"
onClick={fetchUserEvaluatedCourses}>
    Отримати оцінені курси
  </button>
</div>

{userEvaluatedCourses.length > 0 ? (
  <ul className="list-group">
    {userEvaluatedCourses.map(course => (
      <li key={course.id} className="list-group-item d-
flex justify-content-between align-items-center">
        <span>{course.name}</span>
        <button className="btn btn-warning btn-sm"
onClick={ () => resetUserAttempt (course.id) }>
          Скинути спробу (тимчасово не працює)
        </button>
      </li>
    ))}
  </ul>
) : (
  <p>Цей користувач ще не оцінював жоден курс.</p>
)}
</div>

);

};

export default AdminPanel;

import { useState, useContext } from 'react';
import { AuthContext } from '../context/AuthContext';
import { CourseContext } from '../context/CourseContext';
import { Navigate, useNavigate } from 'react-router-dom';

```

					ІАЛЦ.467200.007 ДГ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		


```

import 'bootstrap/dist/css/bootstrap.min.css';

const CourseCreator = () => {
  const { user } = useContext(AuthContext);
  const { addCourse } = useContext(CourseContext);
  const navigate = useNavigate();
  const [program, setProgram] = useState('');
  const [year, setYear] = useState('1');
  const [name, setName] = useState('');
  const [questionsText, setQuestionsText] = useState('');

  if (!user) return <Navigate to="/login" replace />;
  if (user.role !== 'admin') return <Navigate to="/" replace />;

  const handleSubmit = (e) => {
    e.preventDefault();
    const questions = questionsText.split('\n').filter(q => q.trim()
    !== '');
    const newCourse = {
      program,
      year: Number(year),
      name,
      questions
    };
    addCourse(newCourse);
    navigate('/admin');
  };

  return (
    <div className="container mt-4">
      <h2>Створення нового курсу</h2>
      <form onSubmit={handleSubmit}>
        <div className="mb-3">
          <label className="form-label">Освітня
програма:</label>
          <input
            type="text"
            className="form-control"
            value={program}
            onChange={(e) => setProgram(e.target.value)}
            required
          />
        </div>

```

					ІАЛЦ.467200.007 ДГ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

```

<div className="mb-3">
    <label className="form-label">Курс (рік
навчання):</label>
    <select
        className="form-select"
        value={year}
        onChange={ (e) => setYear(e.target.value) }
        required
    >
        <option value="1">1 курс</option>
        <option value="2">2 курс</option>
        <option value="3">3 курс</option>
        <option value="4">4 курс</option>
    </select>
</div>
<div className="mb-3">
    <label className="form-label">Назва курсу
(дисципліни):</label>
    <input
        type="text"
        className="form-control"
        value={name}
        onChange={ (e) => setName(e.target.value) }
        required
    />
</div>
<div className="mb-3">
    <label className="form-label">Питання анкети (кожне з
нового рядка):</label>
    <textarea
        className="form-control"
        rows="5"
        value={questionsText}
        onChange={ (e) => setQuestionsText(e.target.value) }
        placeholder="Введіть питання, кожне з нового
рядка"
        required
    />
</div>
<button type="submit" className="btn btn-success">Створити
курс</button>
</form>
</div>

```

					ІАЛЦ.467200.007 ДГ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    );
};

export default CourseCreator;

import { useContext, useEffect, useState } from 'react';
import { useParams, Navigate, useNavigate } from 'react-router-dom';
import { AuthContext } from '../context/AuthContext';
import { CourseContext } from '../context/CourseContext';
import 'bootstrap/dist/css/bootstrap.min.css';

const CourseEvaluator = () => {
    const { user } = useContext(AuthContext);
    const { courses, responses, addResponse } = useContext(CourseContext);
    const { courseId } = useParams();
    const navigate = useNavigate();
    const [answers, setAnswers] = useState([]);
    const [loading, setLoading] = useState(true);

    const course = courses.find(c => c.id === Number(courseId));

    useEffect(() => {
        if (course && answers.length === 0) {
            setAnswers(course.questions.map(() => ""));
        }
        setLoading(false);
    }, [course]);

    if (!user) return <Navigate to="/login" replace />;
    if (user.role !== 'student') return <Navigate to="/" replace />;
    if (!course) return <div>Курс не знайдено.</div>;
    if (course.year !== user.year) return <div>Ви не можете оцінювати цей курс.</div>;

    const hasEvaluated = responses.some(res =>
        res.userEmail === user.email && res.courseId === Number(courseId)
    );

    if (loading) return <div>Завантаження...</div>;
    if (hasEvaluated) return <div>Ви вже оцінили цей курс.</div>;

    const handleAnswerChange = (index, value) => {
        const newAnswers = [...answers];

```

					ІАЛЦ.467200.007 ДГ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        newAnswers[index] = value;
        setAnswers(newAnswers);
    };

    const handleSubmit = async (event) => {
        event.preventDefault();
        await addResponse(course.id, user.year, answers, user.email);
        alert("Ваші відповіді успішно відправлено. Дякуємо за участь!");
        navigate('/courses');
    };

    return (
        <div className="container mt-4">
            <h2>Анкета оцінювання курсу: {course.name}</h2>
            <p><em>Освітня програма: {course.program}, {course.year}
курс</em></p>
            <form onSubmit={handleSubmit}>
                {course.questions.map((q, idx) => (
                    <div className="mb-3" key={idx}>
                        <label className="form-label">{q}</label>
                        <select
                            className="form-select"
                            value={answers[idx] || ''}
                            onChange={(e) => handleAnswerChange(idx,
e.target.value)}
                            required
                        >
                            <option value="" disabled>Оберіть
оцінку</option>
                            {[1, 2, 3, 4, 5].map(score => (
                                <option key={score}
value={score}>{score}</option>
                            ))}
                        </select>
                    </div>
                ))}
                <button type="submit" className="btn btn-
success">Відправити відповіді</button>
            </form>
        </div>
    );
};

```

					ІАЛЦ.467200.007 ДГ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

```

export default CourseEvaluator;

import { useContext } from 'react';
import { AuthContext } from '../context/AuthContext';
import { CourseContext } from '../context/CourseContext';
import { Navigate, Link } from 'react-router-dom';

const CourseList = () => {
  const { user } = useContext(AuthContext);
  const { courses, responses } = useContext(CourseContext);

  if (!user) return <Navigate to="/login" replace />;
  if (user.role !== 'student') return <Navigate to="/" replace />;

  const availableCourses = courses.filter(course => course.year ===
user.year);

  const hasVoted = (courseId) =>
    responses.some(res => res.userEmail === user.email && res.courseId
=== courseId);

  return (
    <div className="container mt-4">
      <h2>Доступні курси для оцінювання</h2>
      {availableCourses.length > 0 ? (
        <ul className="list-group">
          {availableCourses.map(course => (
            <li key={course.id} className="list-group-item d-
flex justify-content-between align-items-center">
              <div>
                <strong>{course.name}</strong>
                <em>({course.program}, {course.year} курс)</em>
              </div>
              {hasVoted(course.id) ? (
                <span className="text-success">Вже
пройдено</span>
              ) : (
                <Link to={` /courses/${course.id}` }
className="btn btn-success btn-sm">
                  Оцінити
                </Link>
              )}
            </li>
          )}
        </ul>
      ) : null}
    </div>
  );
}

```

					ІАЛЦ.467200.007 ДГ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        </li>
      )}}
    </ul>
  ) : (
    <p>Немає курсів для {user.year}-го курсу.</p>
  )}
</div>
);

};

export default CourseList;

import { useContext } from 'react';
import { AuthContext } from '../context/AuthContext';
import { Link } from 'react-router-dom';
import educationAnalyticsImage from '../images/education_analytics.jpg';
import LogoutButton from './LogoutButton'; //  Додаємо компонент

const HomePage = () => {
  const { user } = useContext(AuthContext);

  return (
    <div className="container mt-5">
      <div className="d-flex justify-content-end mb-3">
        {user && <LogoutButton />} { /*  Кнопка "Вийти" */ }
      </div>

      {!user && (
        <div>
          <h2>Ласкаво просимо на освітню платформу!</h2>
          <p>Будь ласка, <Link to="/login">увійдіть</Link>, щоб
продовжити.</p>
        </div>
      )}

      {user && user.role === 'student' && (
        <div className="mb-4">
          <h2>Вітаємо, студент {user.year}-го курсу!</h2>
          <p>Ви можете оцінити дисципліни своєї освітньої
програми.</p>

          <Link to="/courses" className="btn btn-
primary">Переглянути курси для оцінювання</Link>

```

					ІАЛЦ.467200.007 ДГ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        </div>
    )}

    {user && user.role === 'admin' && (
        <div className="mb-4">
            <h2>Вітаємо, Адміністратор!</h2>
            <p>Перейдіть до панелі адміністратора, щоб керувати
курсами.</p>
            <Link to="/admin" className="btn btn-primary">Відкрити
панель адміністратора</Link>
        </div>
    )}

    <div className="card mb-5 border-0 shadow-sm">
        <img
            src={educationAnalyticsImage}
            className="card-img-top"
            alt="Аналітика освітніх програм"
            style={{maxHeight: '600px', objectFit: 'cover'}}
        />
        <div className="card-body text-center">
            <h1 className="card-title display-4 fw-bold">Освітня
Платформа Аналітики</h1>
            <p className="card-text lead">
                Оцінюйте та аналізуйте освітні програми
університету з легкістю!
            </p>
        </div>
    </div>
    </div>
    </div>
    );
};

export default HomePage;

import { useState, useContext, useEffect } from 'react';
import { AuthContext } from '../context/AuthContext';
import { useNavigate } from 'react-router-dom';

const LoginPage = () => {
    const { user, login } = useContext(AuthContext);
    const navigate = useNavigate();

```

					ІАЛЦ.467200.007 ДГ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

```

const [email, setEmail] = useState('');
const [password, setPassword] = useState('');
const [error, setError] = useState(null);

useEffect(() => {
  if (user) {
    navigate('/');
  }
}, [user, navigate]);

const handleSubmit = async (e) => {
  e.preventDefault();
  const success = await login(email, password);
  if (success) {
    const storedUser = JSON.parse(localStorage.getItem('user'));
    navigate(storedUser?.role === 'admin' ? '/admin' :
'/courses');
  } else {
    setError('Невірний email або пароль');
  }
};

return (
  <div className="container mt-5" style={{ maxWidth: '400px' }}>
    <h2>Вхід до системи</h2>
    <form onSubmit={handleSubmit}>
      <div className="mb-3">
        <label className="form-label">Email:</label>
        <input
          type="email"
          className="form-control"
          value={email}
          onChange={(e) => setEmail(e.target.value)}
          required
        />
      </div>
      <div className="mb-3">
        <label className="form-label">Пароль:</label>
        <input
          type="password"
          className="form-control"
          value={password}

```

					ІАЛЦ.467200.007 ДГ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		


```

        onChange={ (e) => setPassword(e.target.value) }
        required
      />
    </div>
    {error && <div className="text-danger mb-3">{error}</div>}
    <button type="submit" className="btn btn-
primary">Увійти</button>
  </form>
</div>
);
};

export default LoginPage;

import { useContext } from 'react';
import { AuthContext } from '../context/AuthContext';
import { useNavigate } from 'react-router-dom';

const LogoutButton = () => {
  const { logout } = useContext(AuthContext);
  const navigate = useNavigate();

  const handleLogout = () => {
    logout();
    navigate('/login');
  };

  return (
    <button className="btn btn-outline-danger btn-sm"
onClick={handleLogout}>
      Вийти
    </button>
  );
};

export default LogoutButton;

import { createContext, useState } from 'react';

export const AuthContext = createContext(null);

export const AuthProvider = ({ children }) => {
  const [user, setUser] = useState(null);

```

					ІАЛЦ.467200.007 ДГ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

```

const login = async (email, password) => {
  try {
    const res = await fetch('http://localhost:5000/api/login', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({ email, password })
    });

    if (!res.ok) return false;

    const { token, user } = await res.json();
    setUser(user);
    localStorage.setItem('token', token);
    return true;
  } catch {
    return false;
  }
};

const logout = () => {
  setUser(null);
  localStorage.removeItem('token');
};

return (
  <AuthContext.Provider value={{ user, login, logout }}>
    {children}
  </AuthContext.Provider>
);
};

import { createContext, useState, useEffect } from 'react';

export const CourseContext = createContext(null);

export const CourseProvider = ({ children }) => {
  const [courses, setCourses] = useState([]);
  const [responses, setResponses] = useState([]);

  const getToken = () => localStorage.getItem('token');

  // Завантаження даних при старті

```

					ІАЛЦ.467200.007 ДГ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

```

useEffect(() => {
    fetchCourses();
    fetchResponses();
}, []);

const fetchCourses = async () => {
    try {
        const res = await fetch('http://localhost:5000/api/courses', {
            headers: { 'Authorization': `Bearer ${getToken()}` }
        });
        if (res.ok) {
            const data = await res.json();
            setCourses(data);
        }
    } catch (err) {
        console.error('Помилка при отриманні курсів:', err);
    }
};

const fetchResponses = async () => {
    try {
        const res = await fetch('http://localhost:5000/api/responses',
{
            headers: { 'Authorization': `Bearer ${getToken()}` }
        });
        if (res.ok) {
            const data = await res.json();
            setResponses(data);
        }
    } catch (err) {
        console.error('Помилка при отриманні відповідей:', err);
    }
};

const addCourse = async (course) => {
    try {
        const res = await fetch('http://localhost:5000/api/courses', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
                'Authorization': `Bearer ${getToken()}`
            },
            body: JSON.stringify(course)
        });
    }
};

```

					ІАЛЦ.467200.007 ДГ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    });
    if (res.ok) {
        const newCourse = await res.json();
        setCourses(prev => [...prev, newCourse]);
    }
} catch (err) {
    console.error('Помилка при додаванні курсу:', err);
}

};

const deleteCourse = (courseId) => {
    setCourses(prev => prev.filter(c => c.id !== courseId));
    setResponses(prev => prev.filter(r => r.courseId !== courseId));
    // ⚠ Тут можна додати запит до бекенду, якщо реалізуєш DELETE
    /api/courses/:id
};

const addResponse = async (courseId, year, answers, userEmail) => {
    const responseTime = new Date().toISOString();
    const formattedAnswers = answers.map((answer, index) => ({
        userEmail,
        courseId,
        questionId: index,
        answer,
        year,
        date: responseTime
    })));

    try {
        const res = await fetch('http://localhost:5000/api/responses',
        {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
                'Authorization': `Bearer ${getToken()}`
            },
            body: JSON.stringify(formattedAnswers)
        });

        if (res.ok) {
            setResponses(prev => [...prev, ...formattedAnswers]);
        }
    } catch (err) {

```

					ІАЛЦ.467200.007 ДГ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

```

        console.error('Помилка при надсиланні відповіді:', err);
    }
};

return (
    <CourseContext.Provider value={{
        courses, responses, addCourse, deleteCourse, addResponse
    }}>
        {children}
    </CourseContext.Provider>
);
};

import { BrowserRouter, Routes, Route, Navigate } from 'react-router-dom';
import LoginPage from './components/LoginPage';
import HomePage from './components/HomePage';
import CourseList from './components/CourseList';
import CourseEvaluator from './components/CourseEvaluator';
import AdminPanel from './components/AdminPanel';
import CourseCreator from './components/CourseCreator';

function App() {
    return (
        <BrowserRouter>
            <Routes>
                <Route path="/login" element={<LoginPage />} />
                <Route path="/" element={<HomePage />} />
                <Route path="/courses" element={<CourseList />} />
                <Route path="/courses/:courseId" element={<CourseEvaluator />} />
            </Routes>
            <Route path="/admin" element={<AdminPanel />} />
            <Route path="/admin/create" element={<CourseCreator />} />
            <Route path="*" element={<Navigate to="/" replace />} />
        </BrowserRouter>
    );
}

export default App;

import React from 'react';

```

					ІАЛЦ.467200.007 ДГ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

```

import ReactDOM from 'react-dom/client';
import App from './App';
import { AuthProvider } from './context/AuthContext';
import { CourseProvider } from './context/CourseContext';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <AuthProvider>
    <CourseProvider>
      <App />
    </CourseProvider>
  </AuthProvider>
);

const express = require('express');
const fs = require('fs');
const cors = require('cors');
const bodyParser = require('body-parser');
const jwt = require('jsonwebtoken');

const app = express();
const PORT = 5000;
const SECRET_KEY = 'super_secret_key_123';

app.use(cors());
app.use(bodyParser.json());

const DB_PATH = './db.json';

const USERS = [
  { email: 'a@gmail.com', password: '44332211', role: 'admin' },
  { email: 'd@gmail.com', password: '55443322', role: 'student', year: 3
}
];

const readDB = () => JSON.parse(fs.readFileSync(DB_PATH, 'utf-8'));
const writeDB = (data) => fs.writeFileSync(DB_PATH, JSON.stringify(data,
null, 2));

// Middleware для авторизації
const authMiddleware = (req, res, next) => {
  const authHeader = req.headers['authorization'];
  if (!authHeader) return res.status(401).json({ error: 'Token missing'

```

					ІАЛЦ.467200.007 ДГ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

```

});

const token = authHeader.split(' ')[1];
jwt.verify(token, SECRET_KEY, (err, user) => {
  if (err) return res.status(403).json({ error: 'Invalid token' });
  req.user = user;
  next();
});
});

// Логін
app.post('/api/login', (req, res) => {
  const { email, password } = req.body;
  const user = USERS.find(u => u.email === email && u.password ===
password);
  if (!user) return res.status(401).json({ error: 'Invalid credentials'
});

  const token = jwt.sign(user, SECRET_KEY, { expiresIn: '2h' });
  res.json({ token, user });
});

// Курсова логіка
app.get('/api/courses', authMiddleware, (req, res) => {
  const db = readDB();
  res.json(db.courses);
});

app.post('/api/courses', authMiddleware, (req, res) => {
  if (req.user.role !== 'admin') return res.status(403).json({ error:
'Access denied' });
  const db = readDB();
  const newCourse = { id: Date.now(), ...req.body };
  db.courses.push(newCourse);
  writeDB(db);
  res.status(201).json(newCourse);
});

// Відповіді
app.get('/api/responses', authMiddleware, (req, res) => {
  const db = readDB();
  res.json(db.responses);
});

```

					ІАЛЦ.467200.007 ДГ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

```

app.post('/api/responses', authMiddleware, (req, res) => {
  const db = readDB();
  db.responses.push(...req.body);
  writeDB(db);
  res.status(201).json({ success: true });
});

app.delete('/api/responses', authMiddleware, (req, res) => {
  const { userEmail, courseId } = req.body;
  if (req.user.role !== 'admin') return res.status(403).json({ error:
'Access denied' });

  const db = readDB();
  const originalLength = db.responses.length;
  db.responses = db.responses.filter(res =>
    !(res.userEmail === userEmail && res.courseId === courseId)
  );
  writeDB(db);
  res.json({ success: true, removed: originalLength -
db.responses.length });
});

app.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});

```

					ІАЛЦ.467200.007 ДГ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		