

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)
на тему:

«Вебплатформа дистанційного навчання»

Виконала студентка групи ПЗ-22-4
КОЛЬЦОВА Надія Олегівна

Керівник роботи:
САВІЦЬКИЙ Роман Святославович

Рецензент:
ЄФРЕМОВ Юрій Миколайович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення

Тетяна ВАКАЛЮК

«28» лютого 2026 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувачка вищої освіти: **КОЛЬЦОВА Надія Олегівна**

Керівник роботи: **САВІЦЬКИЙ Роман Святославович**

Тема роботи: **«Вебплатформа дистанційного навчання»,**

затверджена Наказом закладу вищої освіти від **«27» лютого 2026 р., №92**

Вихідні дані для роботи: **вимоги до організації дистанційного навчання, функціональні можливості освітніх вебплатформ, архітектурні рішення, інформаційні моделі та програмні засоби для створення системи керування навчальними курсами, матеріалами й результатами навчання.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Савіцький Р.С.	02.03.2026р.	02.03.2026р.
2	Савіцький Р.С.	06.04.2026р.	06.04.2026р.
3	Савіцький Р.С.	01.05.2026р.	04.05.2026р.

РЕФЕРАТ

Бакалаврська кваліфікаційна робота складається з програмного продукту: вебплатформи дистанційного навчання та пояснювальної записки. Пояснювальна записка містить 109 сторінок, з них 95 сторінок основного тексту, 47 рисунків, 23 таблиці, список із 31 використаного джерела та 1 додаток.

Метою роботи є проєктування та реалізація вебплатформи дистанційного навчання для створення, публікації, придбання та проходження відеокурсів із підтримкою рольового доступу, контролю прогресу та оцінювання результатів навчання.

У роботі застосовано методи системного та порівняльного аналізу предметної області, об'єктно-орієнтованого і компонентного проєктування, моделювання реляційної бази даних, розроблення вебзастосунків та функціонального тестування.

У результаті розроблено повностекову вебплатформу для студентів, викладачів та адміністраторів. Реалізовано керування курсами й навчальними матеріалами, оплату курсів, безоплатний і пільговий доступ, відстеження прогресу навчання, тестування, формування сертифікатів, систему відгуків, аналітику та технічну підтримку. Дані зберігаються в базі даних PostgreSQL із використанням Prisma ORM. Для автентифікації, оброблення платежів, роботи з відео та файлами інтегровано зовнішні сервіси.

Практичне значення роботи полягає в можливості використання платформи авторами онлайн-курсів, навчальними центрами та освітніми організаціями для проведення дистанційного навчання.

КЛЮЧОВІ СЛОВА: ВЕБПЛАТФОРМА, ВІДЕОКУРС, ДИСТАНЦІЙНЕ НАВЧАННЯ, СИСТЕМА КЕРУВАННЯ НАВЧАННЯМ, НАВЧАЛЬНИЙ ПРОГРЕС.

					ІПЗ.КР.Б-121-26-ІПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.	Кольцова НО.				Вебплатформа дистанційного навчання	Літ.	Арк.
Керівник	Савіцький Р.С.						3
						Аркуші	109
Рецензент	Єфремов Ю.М.					Житомирська політехніка, група ІПЗ-22-4	
Зав. каф.	Вакалюк Т.А.						

ABSTRACT

The Bachelor's thesis consists of a software product: a web-based distance learning platform and an explanatory note. The explanatory note comprises 109 pages, of which 95 pages are main text, 47 figures, 23 tables, a list of 31 references and 1 appendix.

The aim of the thesis is to design and implement a web-based distance learning platform for the creation, publication, purchase and completion of video courses, with support for role-based access, progress monitoring and assessment of learning outcomes.

The work employs methods of systematic and comparative analysis of the subject area, object-oriented and component-based design, relational database modelling, web application development and functional testing.

As a result, a full-stack web platform has been developed for students, lecturers and administrators. The platform implements management of courses and learning materials, course payments, free and discounted access, tracking of learning progress, testing, certificate generation, a feedback system, analytics and technical support. Data is stored in a using Prisma ORM. External services have been integrated for authentication, payment processing, and handling videos and files.

The practical significance of this work lies in the ability of online course creators, training centres and educational organisations to use the platform for delivering distance learning.

KEYWORDS: WEB PLATFORM, VIDEO COURSE, DISTANCE LEARNING, LEARNING MANAGEMENT SYSTEM, LEARNING PROGRESS.

		Кольцова Н.О.			ПІЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				
						4

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CDN – мережа доставлення вмісту

CSS – каскадні таблиці стилів

FK – зовнішній ключ

HTTP – протокол передавання гіпертексту

HTTPS – захищений протокол передавання гіпертексту

ISO– Міжнародна організація зі стандартизації

LMS – система керування навчанням

MIME – стандарт визначення типу вмісту файла

ORM – технологія об'єктно-реляційного відображення даних

PK – первинний ключ

SQL – мова структурованих запитів до реляційних баз даних

UML – уніфікована мова моделювання

URL – уніфікований показник інформаційного ресурсу

UUID – універсальний унікальний ідентифікатор

WCAG – настанови з доступності вебвмісту

ВПО – внутрішньо переміщена особа

СКБД – система керування базами даних

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				5

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ВЕБПЛАТФОРМИ ДИСТАНЦІЙНОГО НАВЧАННЯ	9
1.1. Постановка задачі	9
1.2. Аналіз аналогів програмного продукту	11
1.3. Вибір архітектури програмного забезпечення.....	15
1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	19
Висновки до першого розділу	22
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБПЛАТФОРМИ ДИСТАНЦІЙНОГО НАВЧАННЯ	24
2.1. Визначення варіантів використання та структури системи	24
2.2. Розробка структури бази даних системи	31
2.3. Проєктування та реалізація алгоритмів роботи системи.....	48
2.4. Реалізація функціональних модулів вебплатформи.....	56
Висновки до другого розділу.....	61
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБПЛАТФОРМОЮ ДИСТАНЦІЙНОГО НАВЧАННЯ.....	62
3.1. Порядок встановлення та налаштування параметрів системи.....	62
3.2. Структура інтерфейсу вебплатформи дистанційного навчання	66
3.3. Тестування роботи програмного продукту	86
Висновки до третього розділу	90
ВИСНОВКИ	91
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	93
ДОДАТКИ	96

ВСТУП

Актуальність теми. Розвиток вебтехнологій і поширення дистанційного навчання зумовлюють потребу в освітніх платформах, які забезпечують централізоване керування навчальними матеріалами, контроль результатів і взаємодію між учасниками освітнього процесу. Сучасні системи керування навчанням мають підтримувати створення структурованих курсів, перегляд відеоматеріалів, проходження тестів, збереження прогресу, розмежування прав доступу та адміністрування.

У межах кваліфікаційної роботи розроблено вебплатформу дистанційного навчання для студентів, викладачів та адміністраторів. Платформа забезпечує створення і публікацію відеокурсів, роботу з навчальними матеріалами, перевірку знань, оплату курсів, збереження прогресу, формування сертифікатів, оцінювання курсів і технічну підтримку. Також передбачено надання пільгового доступу внутрішньо переміщеним особам і ветеранам після підтвердження відповідного статусу.

Метою є проектування та реалізація вебплатформи дистанційного навчання для створення, публікації, придбання та проходження відеокурсів із підтримкою рольового доступу, контролю прогресу та оцінювання результатів навчання.

Для досягнення поставленої мети потрібно виконати такі **завдання**: проаналізувати предметну область і наявні освітні платформи; визначити вимоги до системи; обґрунтувати вибір архітектури та інструментальних засобів; спроектувати структуру програмних компонентів і реляційної бази даних; реалізувати керування курсами, навчальними матеріалами, перевіркою знань і доступом користувачів; забезпечити збереження прогресу та формування сертифікатів; провести функціональне тестування вебплатформи.

Об'єктом дослідження є процес організації дистанційного навчання за допомогою вебплатформи.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				
						7

Предметом дослідження є архітектурні рішення, інформаційні моделі, методи та програмні засоби проєктування і реалізації вебплатформи дистанційного навчання.

У роботі використано методи системного та порівняльного аналізу, об'єктно-орієнтованого і компонентного проєктування, моделювання реляційної бази даних та функціонального тестування. Інформаційну базу дослідження становлять наукові й навчальні джерела, результати аналізу освітніх платформ, офіційна документація використаних технологій і матеріали, отримані під час розроблення програмного продукту.

Практичне значення одержаних результатів полягає у створенні функціональної вебплатформи, яка може використовуватися авторами онлайн-курсів, навчальними центрами, громадськими організаціями та освітніми установами для організації комерційних і соціально орієнтованих програм дистанційного навчання.

Апробація результатів роботи. За результатами дослідження опубліковано одну наукову працю та подано до друку другу:

1. Кольцова Н.О., Савіцький Р.С. Підтримка інклюзивності: курс для реінтеграції ВПО/ветеранів. Тези VIII Всеукраїнської науково-практичної конференції «Комп'ютерні технології: інновації, проблеми, рішення», 2-3 грудня 2025 року. Житомир: «Житомирська політехніка», 2025. С. 358-369 [1];

2. Кольцова Н.О., Савіцький Р.С. Розробка LMS-платформи з використанням Next.js. Тези XIX Міжнародної науково-практичної конференції «Теорія і практика сучасної науки та освіти», 19-20 червня 2026 року. Львів: «Львівський науковий форум», 2026. [Подано до друку]

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				8

РОЗДІЛ 1. АНАЛІЗ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ВЕБПЛАТФОРМИ ДИСТАНЦІЙНОГО НАВЧАННЯ

1.1. Постановка задачі

Розвиток інформаційних технологій і поширення дистанційної форми навчання зумовлюють необхідність створення вебплатформ, які забезпечують централізоване керування освітнім контентом, доступ до навчальних матеріалів і контроль результатів навчання. Такі системи дають змогу користувачам навчатися незалежно від місця перебування, самостійно визначати темп опрацювання матеріалу та отримувати доступ до курсів із різних типів пристроїв.

Організація дистанційного навчання охоплює створення й публікацію курсів, розміщення відеоматеріалів, проведення тестування, збереження прогресу, керування доступом, приймання платежів і підтримку користувачів. Використання окремих сервісів для виконання цих процесів ускладнює адміністрування та не забезпечує цілісного навчального сценарію. Тому виникає потреба у створенні єдиної вебплатформи, яка об'єднуватиме основні засоби організації дистанційного навчання.

У межах кваліфікаційної роботи поставлено задачу розробити вебплатформу дистанційного навчання, призначену для створення, публікації, придбання та проходження відеокурсів. Платформа має підтримувати роботу студентів, викладачів та адміністраторів і надавати кожній категорії користувачів окремий набір функціональних можливостей відповідно до її ролі.

Для роботи з користувачами необхідно реалізувати реєстрацію, автентифікацію, керування профілем і розмежування прав доступу. Студент має отримувати доступ до каталогу курсів, виконувати пошук і фільтрацію, переглядати структуру та опис курсу, опрацьовувати навчальні матеріали, проходити тестування, відстежувати прогрес та отримувати сертифікати. Викладач повинен мати можливість створювати й редагувати курси, завантажувати відеоуроки, додавати текстові матеріали та тести, а також керувати публікацією контенту. Адміністратор має керувати користувачами,

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				
						9

ролями та категоріями курсів, опрацьовувати заявки на пільговий доступ і технічні звернення.

Важливою складовою є організація роботи з відеоматеріалами, що охоплює їх завантаження, оброблення та відтворення, а також збереження позиції перегляду. Для контролю навчання необхідно реалізувати механізм відстеження прогресу та автоматичну перевірку результатів тестування. Після виконання встановлених умов завершення курсу система повинна формувати електронний сертифікат.

Платформа має підтримувати платні та безоплатні курси. Для платних курсів необхідно реалізувати кошик і взаємодію з платіжним сервісом. Соціальна спрямованість програмного продукту передбачає надання пільгового доступу до навчального контенту внутрішньо переміщеним особам і ветеранам після подання заявки та підтвердження відповідного статусу адміністратором.

Для отримання зворотного зв'язку необхідно реалізувати систему оцінок і відгуків, а також аналітичні засоби для викладачів та адміністраторів. Це дасть змогу оцінювати навчальні курси, аналізувати прогрес студентів і контролювати основні показники функціонування платформи.

Поряд із функціональними задачами необхідно забезпечити безпечність і зручність використання програмного продукту. Інтерфейс має бути адаптивним для різних типів пристроїв, а доступ до захищених функцій повинен контролюватися відповідно до ролі користувача. Введені дані мають проходити перевірку перед збереженням та опрацюванням.

У межах кваліфікаційної роботи необхідно створити програмний комплекс, який автоматизує керування користувачами, створення та проходження відеокурсів, тестування, збереження прогресу, оплату й надання різних типів доступу, сертифікацію, оцінювання курсів, аналітику та технічну підтримку. Визначені задачі формують основу для подальшого аналізу аналогів, вибору архітектури та засобів реалізації вебплатформи.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				10

1.2. Аналіз аналогів програмного продукту

Аналіз аналогів є важливим етапом проектування вебплатформи дистанційного навчання, оскільки дає змогу визначити поширені підходи до організації онлайн-освіти, встановити типовий набір функцій програмних продуктів цього класу та виявити можливості для їх удосконалення. Для порівняння обрано українські платформи EdEra і Mate academy, а також міжнародну платформу Coursera. Аналіз виконано за такими характеристиками, як структура каталогу, організація навчального процесу, підтримка відеоматеріалів, тестування, відстеження прогресу, сертифікація та моделі доступу до контенту.

EdEra є українською освітньою платформою, яка пропонує онлайн-курси з різних тематичних напрямів [2]. Навчальні програми містять структуровані матеріали, а для окремих курсів передбачено оцінювання результатів та отримання сертифіката. Каталог організовано за тематичними напрямками, що спрощує пошук необхідного освітнього контенту. Зовнішній вигляд головної сторінки платформи EdEra наведено на рис. 1.1.

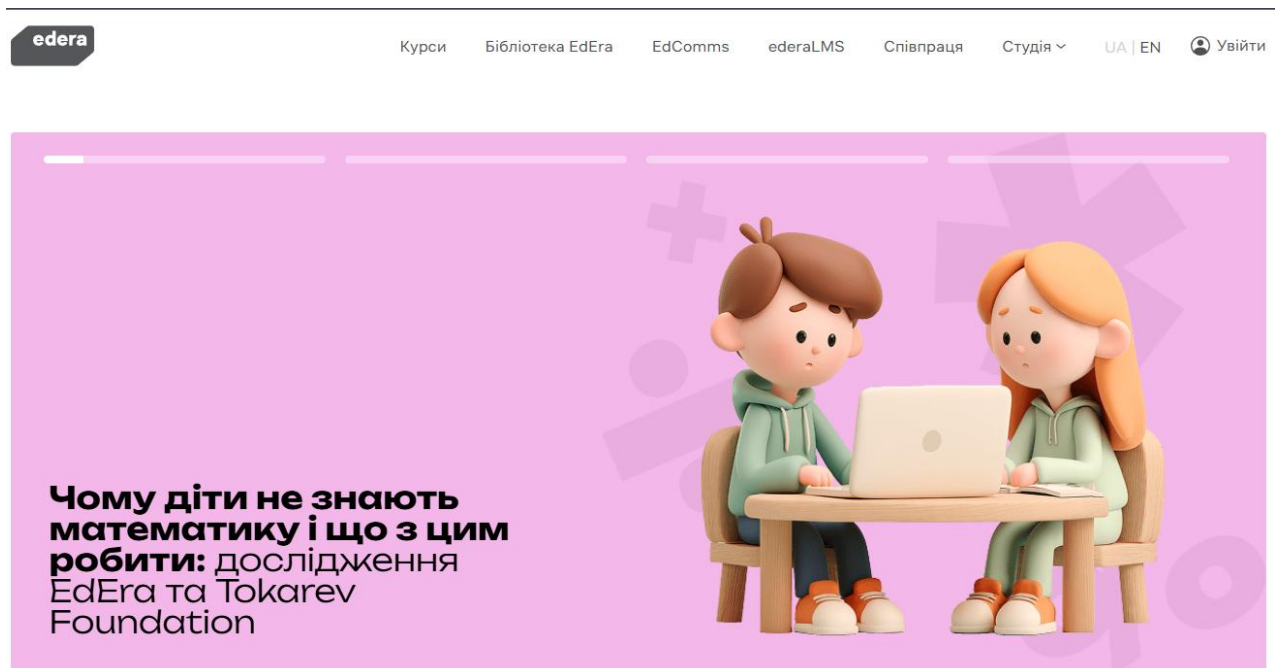


Рис. 1.1. Головна сторінка платформи EdEra

До переваг EdEra належать орієнтація на українську аудиторію, доступність значної частини навчальних матеріалів і зрозуміла структура

		Кольцова Н.О.			ПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				
						11

каталогу. Платформа використовується для поширення готових курсів і реалізації освітніх проєктів. Водночас у відкритій частині сервісу основну увагу зосереджено на проходженні підготовлених програм, а самостійне створення та адміністрування курсів незалежними викладачами не є його основною функцією.

Mate academy орієнтована переважно на професійну підготовку та працевлаштування у сфері інформаційних технологій [3]. Платформа пропонує структуровані навчальні програми, практичні завдання, супровід менторів, контроль проходження навчання та кар'єрну підтримку з гарантією працевлаштування. Залежно від обраного формату може застосовуватися попередня оплата або оплата після працевлаштування. Головну сторінку платформи Mate academy наведено на рис. 1.2.

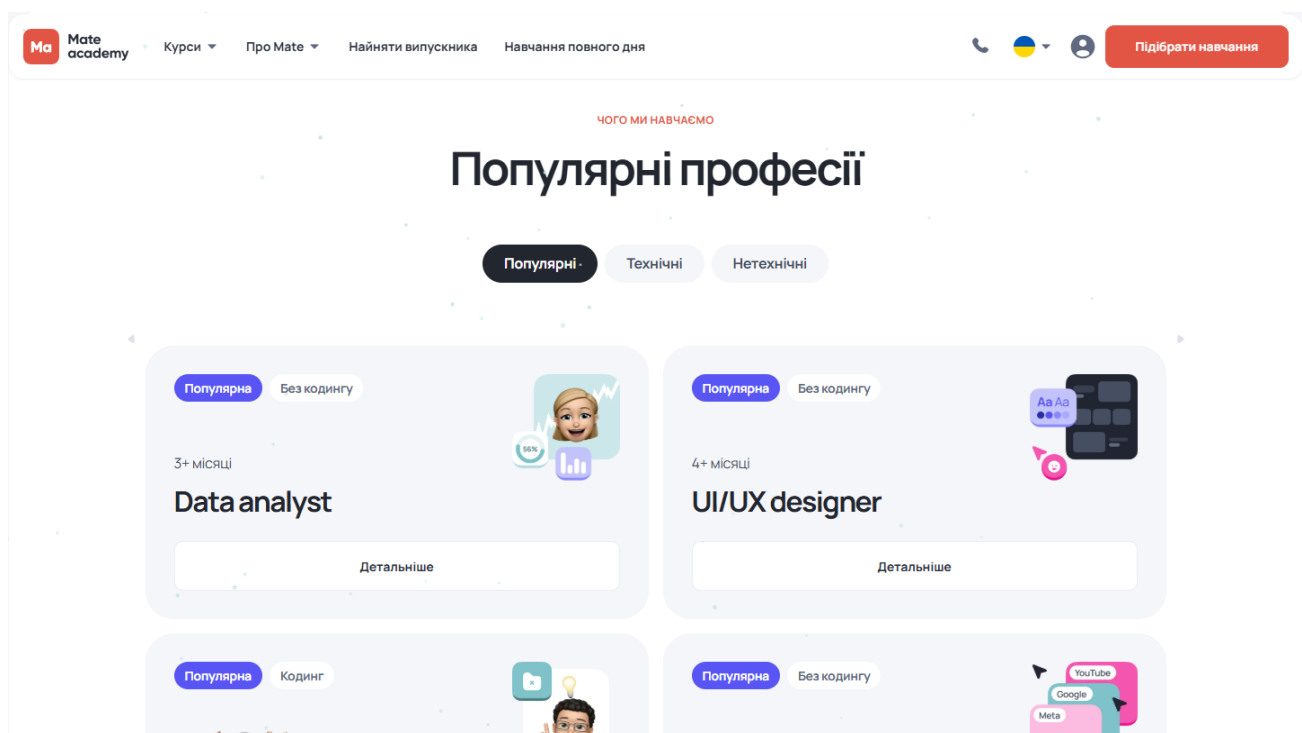


Рис. 1.2. Головна сторінка платформи Mate academy

Перевагами Mate academy є практична спрямованість, послідовна структура навчальних програм і зв'язок навчання з професійним розвитком. Водночас платформа зосереджена на власних освітніх програмах та внутрішній методиці. Можливість створення довільних курсів незалежними викладачами не є основою її моделі, тому сервіс не може розглядатися як універсальна система керування навчальними курсами.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				12

Coursera є міжнародною платформою дистанційного навчання, яка пропонує окремі курси, спеціалізації, професійні сертифікати та освітні програми від університетів і компаній [4]. Навчальний контент може містити відеолекції, текстові матеріали, тести, практичні завдання та підсумкове оцінювання. Після виконання встановлених вимог користувач може отримати електронний сертифікат, що підтверджує набуті знання та навички. Головну сторінку платформи Coursera наведено на рис. 1.3.

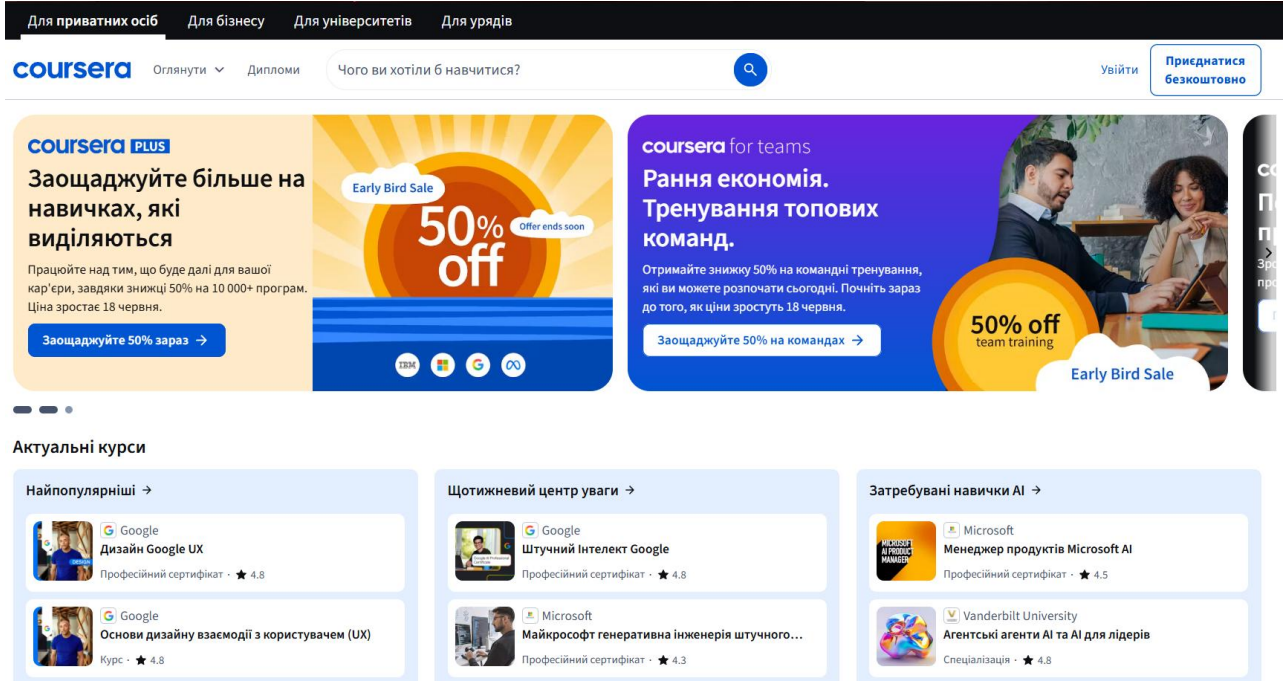


Рис. 1.3. Головна сторінка платформи Coursera

До переваг Coursera належать широкий вибір навчальних напрямів, співпраця з університетами й компаніями та підтримка сертифікації. Повний доступ до багатьох програм і сертифікатів надається на платній основі. Для окремих курсів можуть бути доступні безоплатні матеріали, пробний період або фінансова допомога. Публікація курсів здійснюється переважно через партнерські організації, тому платформа не орієнтована на самостійне розміщення курсів окремими викладачами.

Розглянуті платформи відрізняються моделями доступу до навчальних матеріалів і можливостями для авторів курсів. Їх порівняння дає змогу визначити функції, які доцільно передбачити під час розроблення вебплатформи дистанційного навчання.

Узагальнене порівняння розглянутих платформ наведено в табл. 1.1.

Таблиця 1.1

Порівняльна характеристика аналогів

Критерій	EdEra	Mate academy	Coursera	Коментар
Каталог онлайн-курсів	+	+	+	Каталог курсів наявний у всіх розглянутих платформах.
Відеоматеріали	+	+	+	Відеоматеріали використовуються в навчальних програмах усіх платформ.
Тестування	+/-	+	+	В EdEra наявність тестів залежить від конкретного курсу.
Відстеження прогресу	+	+	+	Усі платформи дають змогу контролювати проходження навчальної програми.
Електронні сертифікати	+/-	+	+	В EdEra сертифікат передбачено не для всіх курсів.
Безоплатний доступ	+	+/-	+/-	У Mate academy та Coursera умови залежать від програми або формату навчання.
Платний доступ	+/-	+	+	Для EdEra платний доступ не є основною моделлю роботи.
Самостійне створення курсів викладачами	—	—	+/-	У Coursera публікація курсів доступна переважно партнерським організаціям.
Аналітика для викладачів	—	+/-	+/-	Наявність аналітичних засобів залежить від ролі користувача та умов співпраці з платформою.
Пільговий доступ після верифікації	—	—	—	У відкритій частині платформ окремого механізму для ВПО та ветеранів не виявлено.

У таблиці використано такі позначення: «+» означає наявність функції; «—» означає її відсутність або відсутність підтвердженої інформації у відкритій

частині платформи; «+/-» означає часткову реалізацію функції або її доступність лише для окремих курсів, програм чи категорій користувачів.

Проведений аналіз показав, що розглянуті платформи використовують різні підходи до організації дистанційного навчання. EdEra орієнтована переважно на українську аудиторію, Mate academy поєднує професійну підготовку з кар'єрним супроводом, а Coursera пропонує широкий вибір програм від університетів і компаній.

Серед розглянутих платформ не виявлено рішення, яке одночасно підтримувало б самостійне створення курсів викладачами, рольове адміністрування, платний, безоплатний і пільговий доступ, аналітику та технічну підтримку. Тому під час розроблення вебплатформи передбачено ці можливості, зокрема надання пільгового доступу внутрішньо переміщеним особам і ветеранам після підтвердження відповідного статусу.

1.3. Вибір архітектури програмного забезпечення

Архітектура програмного забезпечення визначає структуру системи, розподіл відповідальності між її складовими та принципи взаємодії компонентів. Від обраного архітектурного рішення залежать складність розроблення, зручність супроводження, масштабованість і можливість подальшого розширення програмного продукту [5]. Для вебплатформи дистанційного навчання необхідно забезпечити роботу користувачів через браузер, централізоване зберігання даних, виконання серверної логіки та розмежування прав доступу. Система також повинна підтримувати роботу з відеоматеріалами, файлами, платежами та зовнішнім сервісом автентифікації. Під час проектування було розглянуто монолітний і мікросервісний підходи до організації програмного продукту.

Монолітний підхід передбачає об'єднання основних складових системи в одному застосунку. Він спрощує початкове розроблення, локальний запуск і розгортання, оскільки використовується спільна кодова база та не виникає потреби організовувати мережеву взаємодію між власними сервісами. Водночас

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				
						15

зі збільшенням системи може зростати зв'язність її складових, тому програмний код необхідно логічно розподіляти між окремими функціональними модулями.

Мікросервісний підхід передбачає поділ програмного продукту на незалежні сервіси, кожен із яких відповідає за окрему функцію та може розгортатися самостійно. Він забезпечує можливість окремого масштабування складових системи, але потребує складнішої інфраструктури, організації обміну даними, моніторингу та контролю узгодженості інформації. Для вебплатформи такий підхід є надмірним, оскільки ускладнює розроблення та розгортання без достатньої практичної потреби

Взаємодію користувачів із вебплатформою побудовано за клієнт-серверною моделлю, а сам програмний продукт реалізовано у формі модульного повностекового моноліту. Клієнтська частина відповідає за відображення інтерфейсу та опрацювання дій користувача. Серверна частина виконує прикладну логіку, перевіряє права доступу, взаємодіє з базою даних і зовнішніми системами. Основні складові розміщено в одному застосунку, проте програмний код логічно розподілено між модулями курсів, навчального прогресу, тестування, оплати, сертифікації, верифікації, аналітики та технічної підтримки.

Основою застосунку є фреймворк Next.js із маршрутизатором App Router, який дає змогу поєднати клієнтську та серверну частини в межах одного проєкту. Сторінки, компоненти інтерфейсу, серверні обробники та засоби отримання даних організовано в узгодженій файловій структурі. Це спрощує навігацію проєктом, повторне використання типів і компонентів та супроводження програмного коду.

На рівні подання використовуються серверні та клієнтські компоненти React. Серверні компоненти застосовуються для отримання даних, перевірки доступу та початкового формування сторінок. Клієнтські компоненти використовуються у випадках, коли необхідні локальний стан, оброблення подій та інтерактивна взаємодія, зокрема під час роботи з формами, відеопрогравачем, кошиком і тестовими завданнями. Такий розподіл дає змогу виконувати

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				16

захищені операції на сервері та забезпечувати динамічну взаємодію з користувачем.

Маршрути застосунку організовано відповідно до основних сценаріїв роботи системи. Окремо виділено сторінки автентифікації, панелі керування та проходження курсів. Динамічні сегменти маршрутів використовуються для відображення конкретного курсу, глави, тесту, сертифіката або звернення технічної підтримки. Для опрацювання запитів застосовуються обробники маршрутів Route Handlers, які виконують операції з даними та повертають результат клієнтській частині.

Перед виконанням захищених операцій серверна частина перевіряє автентифікацію користувача, його роль і право доступу до відповідного ресурсу. Рольова модель передбачає три категорії користувачів: студента, викладача та адміністратора. Перевірка виконується не лише під час відображення елементів інтерфейсу, а й безпосередньо в серверних обробниках, що запобігає виконанню недозволених операцій через прямі запити.

Рівень доступу до даних реалізовано за допомогою Prisma ORM, яка забезпечує взаємодію серверної частини з реляційною базою даних PostgreSQL. Структуру предметної області описано у схемі Prisma, на основі якої генерується типізований клієнт для виконання операцій із даними. Такий підхід забезпечує узгодженість між програмним кодом і моделями бази даних, підтримує зв'язки між сутностями, обмеження цілісності та транзакційне виконання операцій.

Окрему групу архітектурних складових утворюють інтеграції із зовнішніми сервісами. Clerk використовується для реєстрації, входу, керування обліковими записами та сесіями користувачів. Stripe забезпечує створення платіжних сесій і підтвердження придбання курсів. Mux застосовується для оброблення та відтворення відеоматеріалів, а UploadThing використовується для завантаження обкладинок, навчальних ресурсів і документів для верифікації. Застосування спеціалізованих платформ дає змогу не створювати власні механізми оброблення платежів, потокового відео та файлового зберігання.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				17

Взаємодія із зовнішніми системами здійснюється через програмні інтерфейси та вебхуки. Наприклад, після успішного завершення платежу Stripe надсилає серверній частині підписане повідомлення. Після його перевірки в базі даних створюється запис про доступ користувача до придбаного курсу. Службові ключі зовнішніх сервісів зберігаються у змінних середовища та не передаються до клієнтської частини.

Узагальнену архітектуру вебплатформи наведено на рис. 1.4.

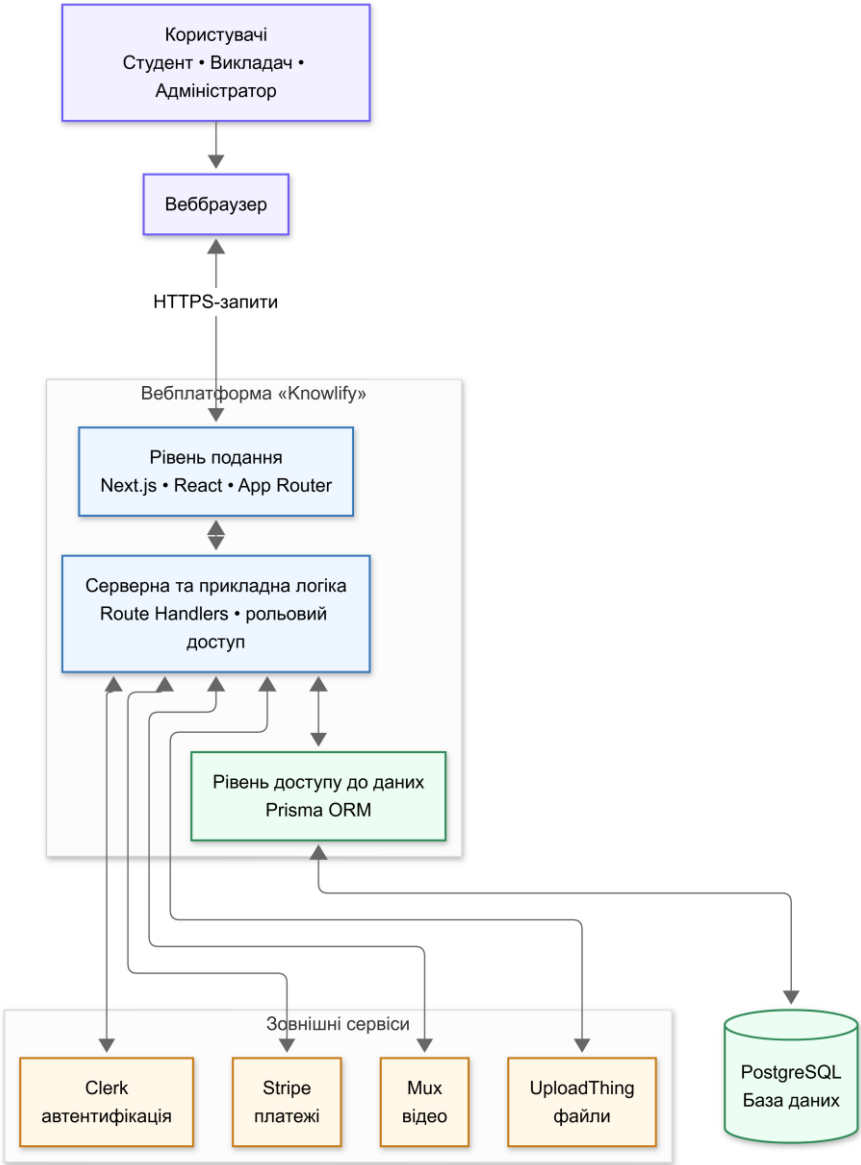


Рис. 1.4. Узагальнена архітектура вебплатформи

На схемі відображено взаємодію користувачів із платформою через веббраузер, рівень подання на основі Next.js і React, серверну та прикладну логіку, Prisma ORM, базу даних PostgreSQL і зовнішні сервіси. Зв'язки між

складовими показують передавання HTTPS запитів, виконання серверних операцій, звернення до бази даних та обмін інформацією з інтегрованими платформами.

Типовий процес взаємодії починається з дії користувача у веббраузері. Запит передається до серверного компонента або обробника маршруту, де перевіряються автентифікація, роль користувача та отримані дані. Після цього виконується прикладна операція, звернення до бази даних або зовнішнього сервісу, а результат повертається до інтерфейсу. Така послідовність забезпечує централізоване опрацювання даних і контроль доступу.

Отже, вебплатформу побудовано відповідно до клієнт-серверної моделі у формі модульного повностекового моноліту. Обрана архітектура поєднує простоту єдиного застосунку з логічним поділом його складових, забезпечує централізовану роботу з даними, серверну перевірку доступу та взаємодію зі спеціалізованими зовнішніми сервісами. Такий підхід відповідає функціональним вимогам платформи та дає змогу розширювати окремі модулі без упровадження складної мікросервісної інфраструктури.

1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Розроблення вебплатформи дистанційного навчання потребує засобів для створення адаптивного інтерфейсу, виконання серверної логіки, зберігання даних та інтеграції із зовнішніми сервісами. Під час вибору технологій враховувалися їхня відповідність обраній архітектурі, підтримка типізації, зручність супроводження та придатність до хмарного розгортання.

Основним фреймворком вебплатформи обрано Next.js [6]. Він дає змогу поєднати клієнтську та серверну частини в одному проєкті, використовувати файлову маршрутизацію, динамічні сторінки, серверні й клієнтські компоненти та обробники маршрутів. Така організація відповідає архітектурі модульного повностекового застосунку та спрощує взаємодію між інтерфейсом, серверною логікою і даними.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				
						19

Користувацький інтерфейс побудовано за допомогою React [7]. Компонентний підхід забезпечує повторне використання карток курсів, форм, таблиць, навігаційних елементів і засобів відображення навчального прогресу. Серверні компоненти застосовуються для отримання даних і початкового формування сторінок. Клієнтські компоненти використовуються для інтерактивних форм, тестування, кошика та відеопрогравача.

Основною мовою програмування обрано TypeScript [8]. Статична типізація дає змогу виявляти частину помилок до запуску застосунку та забезпечує узгодженість між компонентами, серверними обробниками й моделями даних. Це полегшує супроводження програмного коду та внесення змін до функціональних модулів.

Для оформлення інтерфейсу використано Tailwind CSS [9], компоненти shadcn/ui та бібліотеку піктограм Lucide React. Tailwind CSS забезпечує керування стилями й адаптивністю, а shadcn/ui надає основу для створення форм, кнопок, меню, таблиць і діалогових вікон. За допомогою цих засобів реалізовано єдиний стиль платформи та коректне відображення сторінок на різних пристроях.

Для окремих функцій клієнтської частини використано Zustand, TanStack Table, Axios і React Hot Toast. Zustand застосовується для зберігання стану кошика. TanStack Table використовується для побудови таблиць у викладацькій та адміністративній частинах. Axios забезпечує виконання HTTP запитів, а React Hot Toast використовується для відображення повідомлень про результати операцій.

Рівень доступу до даних реалізовано за допомогою Prisma ORM [10]. Моделі предметної області, зв'язки між ними та обмеження цілісності описуються у схемі Prisma, на основі якої створюється типізований клієнт. Це зменшує потребу в ручному формуванні SQL запитів і забезпечує узгодженість між структурою бази даних та серверним кодом.

Як систему керування базами даних обрано PostgreSQL [11]. Реляційна модель відповідає структурі платформи, у якій користувачі пов'язані з курсами,

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				20

главами, тестами, доступами, навчальним прогресом, сертифікатами та відгуками. PostgreSQL підтримує зовнішні ключі, транзакції, індекси й унікальні обмеження, необхідні для збереження цілісності даних. Базу даних розміщено у хмарному сервісі Neon, який надає засоби керованого розміщення PostgreSQL та віддаленого підключення до неї [12].

Для автентифікації користувачів використано Clerk, який забезпечує реєстрацію, вхід, керування обліковими записами та сесіями [13]. Ролі й додаткові відомості профілів зберігаються у власній базі даних, що дає змогу розмежувати можливості студента, викладача й адміністратора.

Оплату курсів реалізовано за допомогою Stripe [14]. Серверна частина створює платіжну сесію, після чого користувач переходить до захищеної платіжної сторінки. Інформація про результат операції надходить через вебхук. Після перевірки повідомлення в базі даних створюється запис про доступ до придбаного курсу.

Для роботи з відеоматеріалами обрано Mux [15]. Сервіс виконує завантаження, оброблення та потокове відтворення відео, що дає змогу не створювати власну інфраструктуру для перекодування великих файлів. У базі даних платформи зберігаються службові ідентифікатори, необхідні для зв'язування відеоматеріалів із відповідними главами.

Обкладинки курсів, навчальні матеріали й документи для верифікації завантажуються через UploadThing [16]. Файли зберігаються у зовнішньому сховищі, а в базі даних містяться їхні адреси та службові ідентифікатори. Такий підхід відповідає особливостям хмарного розгортання, за якого локальна файлова система застосунку не використовується для постійного зберігання матеріалів.

Розроблення програмного продукту виконувалося у Visual Studio Code. Для запуску проєкту й керування залежностями використовувалися Node.js і npm [17, 18]. Контроль версій здійснювався за допомогою Git [19], а віддалений репозиторій розміщено в GitLab. Використання окремих гілок і запитів на злиття

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				
						21

використовуються для перегляду й контролю змін перед їх додаванням до основної гілки проєкту [19, 20].

Для розгортання вебзастосунку передбачено використання Vercel [21]. Платформа підтримує застосунки Next.js, серверні функції, змінні середовища та автоматизоване складання проєкту. База даних при цьому працює в Neon, відеоматеріали обслуговуються Mux, а файли зберігаються через UploadThing. Така організація зменшує потребу у створенні та обслуговуванні власної серверної інфраструктури.

Отже, обрані інструментальні засоби відповідають архітектурі та функціональним вимогам вебплатформи. Next.js, React і TypeScript забезпечують реалізацію клієнтської та серверної частин. Prisma ORM і PostgreSQL використовуються для роботи з даними. Clerk, Stripe, Mux та UploadThing надають спеціалізовані засоби автентифікації, оплати, оброблення відео й зберігання файлів. Використання хмарних сервісів спрощує розгортання системи та зменшує потребу у власному апаратному забезпеченні

Висновки до першого розділу

У першому розділі проаналізовано предметну область дистанційного навчання та визначено основні завдання, які повинна виконувати вебплатформа. Система має забезпечувати керування користувачами й курсами, роботу з відеоматеріалами, тестування, збереження навчального прогресу, оплату курсів, надання різних типів доступу, формування сертифікатів, систему відгуків, аналітику та технічну підтримку.

Порівняння платформ EdEra, Mate academy і Coursera дало змогу визначити поширені функції систем дистанційного навчання та особливості розроблюваного програмного продукту. До них належать самостійне створення курсів викладачами, рольове розмежування доступу, підтримка платних, безоплатних і пільгових курсів, аналітичні засоби та централізоване адміністрування.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				22

Для реалізації вебплатформи обрано клієнт-серверну модель у формі модульного повностекового моноліту. Такий підхід забезпечує логічний поділ функціональних складових у межах єдиного застосунку, централізовану роботу з даними та взаємодію із зовнішніми сервісами. Основу технологічного стеку становлять Next.js, React і TypeScript. Для доступу до даних використовується Prisma ORM, а як СКБД обрано PostgreSQL. Сервіси Clerk, Stripe, Mux і UploadThing забезпечують автентифікацію, оброблення платежів, роботу з відеоматеріалами та зберігання файлів. Для розгортання вебзастосунку передбачено використання Vercel, а базу даних розміщено у хмарному сервісі Neon.

Обрані архітектурні та інструментальні рішення відповідають визначеним функціональним вимогам і дають змогу реалізувати вебплатформу без створення складної власної серверної інфраструктури. Визначені в першому розділі вимоги, архітектурні рішення та технології є основою для подальшого проєктування бази даних, програмних модулів і користувацьких сценаріїв системи.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				23

РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБПЛАТФОРМИ ДИСТАНЦІЙНОГО НАВЧАННЯ

2.1. Визначення варіантів використання та структури системи

Проектування вебплатформи дистанційного навчання передбачає визначення функціональних і нефункціональних вимог, категорій користувачів, доступних їм операцій та структури об'єктів предметної області. Вимоги до вебплатформи сформовано на основі результатів аналізу предметної області, розглянутих програмних аналогів і поставленого завдання на розроблення програмного продукту.

Платформа забезпечує основні етапи роботи з відеокурсами, до яких належать створення, редагування, публікація, пошук, придбання та проходження курсів. Додатково реалізовано роботу з відеоуроками й навчальними ресурсами, збереження прогресу, тестування, формування сертифікатів, оцінювання курсів, верифікацію користувачів, аналітику та технічну підтримку.

У системі передбачено три основні ролі: студент, викладач та адміністратор. Призначена роль визначає доступні користувачеві сторінки, дані й операції. Перевірка повноважень виконується як на рівні користувацького інтерфейсу, так і безпосередньо в серверних обробниках. Додатково контролюються право викладача на керування конкретним курсом і наявність у студента доступу до відповідних навчальних матеріалів.

Неавтентифікований користувач не має доступу до каталогу курсів та інших захищених сторінок платформи. Під час спроби відкрити захищений маршрут система перенаправляє його до форми входу. Після реєстрації або успішної автентифікації створюється чи оновлюється локальний профіль користувача, після чого відкриваються функціональні можливості відповідно до його ролі.

Студент може переглядати каталог опублікованих курсів, виконувати пошук і застосовувати передбачені інтерфейсом засоби фільтрування. На сторінці окремого курсу відображаються його назва, опис, програма, категорія, вартість, рейтинг і відгуки користувачів.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				24

Якщо окрему главу позначено як доступну для попереднього безоплатного перегляду, автентифікований студент може відкрити її без отримання повного доступу до курсу. У такому режимі навчальний прогрес і позиція відтворення відео не зберігаються, а проходження тесту до глави залишається недоступним.

Платний курс можна додати до кошика та оплатити через інтегровану платіжну систему. Доступ до безоплатного курсу відкривається без проведення платежу. Після підтвердження пільгового статусу студент може отримати доступ до обраного платного курсу на спеціальних умовах. Підтвердження статусу не означає автоматичного відкриття всіх курсів платформи.

Для проходження верифікації студент створює заявку, зазначає тип підтвердження, за потреби додає коментар і завантажує необхідні документи. Подання нової заявки обмежується, якщо попередня заявка ще перебуває на розгляді або статус користувача вже підтверджено. У профілі відображаються стан поточної заявки та відомості про попередні звернення.

Після отримання доступу до курсу студент може переглядати відеоуроки й додаткові навчальні матеріали, продовжувати відтворення відео зі збереженої позиції, позначати глави завершеними та контролювати загальний прогрес навчання. Для кожної пари користувача і глави формується один запис прогресу, що запобігає дублюванню даних.

Опубліковані тести до глав стають доступними після отримання повного доступу до курсу. Під час надсилання відповідей система перевіряє їхню повноту, обчислює результат, порівнює його з установленим прохідним балом і зберігає спробу разом із вибраними відповідями. Результати попередніх спроб залишаються доступними після повторного відкриття сторінки.

Підсумковий тест відкривається після завершення всіх опублікованих глав курсу. У разі успішного проходження тестування студент може сформувати сертифікат з унікальним кодом перевірки. Для однієї пари користувача і курсу створюється не більше одного сертифіката, який можна переглянути та завантажити.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				25

Користувач, який отримав платний, безоплатний або пільговий доступ до курсу, може оцінити його за п'ятибальною шкалою, додати текстовий коментар, відредагувати або видалити власний відгук. Для кожної пари користувача і курсу допускається лише один відгук. Автор курсу не може оцінювати власний навчальний матеріал.

Викладач може створювати, редагувати та видаляти власні курси. До основних параметрів курсу належать назва, опис, категорія, обкладинка, ціна та стан публікації. Окремої сутності для розділів у системі не передбачено, тому структура курсу формується безпосередньо з упорядкованих глав.

Для кожної глави викладач визначає назву, опис, позицію, стан публікації та можливість попереднього безоплатного перегляду. До глави можна додати відеофайл або зовнішнє відеопосилання, а також файли й посилання на додаткові навчальні ресурси. Глави можна створювати, редагувати, видаляти та переміщувати, змінюючи порядок їх відображення в курсі.

Публікація глави дозволяється після заповнення обов'язкових відомостей і додавання відеоматеріалу. Якщо після видалення або зняття з публікації глав у курсі не залишається жодної опублікованої глави, курс також знімається з публікації.

До кожної глави можна додати один тест із назвою, описом, прохідним балом, запитаннями та варіантами відповідей. Кожне запитання повинно містити щонайменше два варіанти й одну правильну відповідь. Окремо для курсу створюється підсумковий тест, проходження якого є однією з умов формування сертифіката.

Перед публікацією курсу система перевіряє наявність назви, опису, обкладинки, ціни, категорії та щонайменше однієї опублікованої глави. Обов'язковою умовою також є наявність опублікованого підсумкового тесту, який містить хоча б одне запитання. Кожне запитання повинно мати щонайменше два варіанти відповіді та одну правильну відповідь. Викладач може знімати з публікації курс, окремі глави й тести.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				
						26

Аналітичний модуль надає викладачеві відомості про кількість платних, безоплатних і пільгових доступів, орієнтовний дохід, середній прогрес студентів, кількість сформованих сертифікатів, рейтинг курсів та останні випадки надання доступу.

Адміністратору доступні загальні показники роботи платформи, зокрема відомості про користувачів, курси, надані доступи, заявки на верифікацію та кількість сформованих сертифікатів. Він може переглядати профілі користувачів, змінювати їхні ролі та контролювати дотримання правил доступу.

До адміністративних функцій належить керування категоріями курсів. Категорію можна створити, перейменувати або видалити. Видалення забороняється, якщо до відповідної категорії належить хоча б один курс. Адміністратор також може переглядати курси всіх викладачів і контролювати їхній стан.

Під час опрацювання заявки на верифікацію адміністратор переглядає відомості про користувача та завантажені документи, після чого підтверджує або відхиляє заявку й за потреби додає службовий коментар. Результат перевірки зберігається в заявці та відображається у профілі користувача.

Студент і викладач можуть створювати звернення до технічної підтримки, обирати категорію проблеми, зазначати тему та додавати повідомлення. Адміністратор переглядає всі звернення, відповідає на них і змінює їхній стан. Для звернень передбачено відкритий, опрацьований і закритий стани.

Наведені сценарії визначають функціональні вимоги до вебплатформи та є основою для формування варіантів її використання.

Нефункціональні вимоги визначають характеристики безпеки, надійності, продуктивності, доступності та зручності використання системи. Ці характеристики належать до основних складових моделі якості програмного продукту ISO/IEC 25010 [22].

Інтерфейс має адаптуватися до екранів комп'ютерів, планшетів і смартфонів. Навігація повинна підтримувати керування з клавіатури, швидкий перехід до основного вмісту та зменшення анімацій відповідно до налаштувань

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				
						27

операційної системи. Такі рішення відповідають рекомендаціям щодо доступності вебвмісту WCAG 2.2 [23]. Для процесів завантаження, відсутності результатів, втрати з'єднання та виникнення помилок передбачаються окремі стани інтерфейсу.

Захищені операції виконуються після серверної перевірки автентифікації, ролі користувача, належності ресурсу та наявності доступу до курсу. Вхідні дані перевіряються перед збереженням, а службові ключі зовнішніх сервісів зберігаються у змінних середовища. Серверна перевірка прав доступу зменшує ризик виконання несанкціонованих операцій, пов'язаних із порушенням контролю доступу [24].

Цілісність інформації забезпечується унікальними обмеженнями для доступу до курсу, прогресу в межах глави, відгуку та сертифіката. Для критичних операцій, що передбачають одночасне оновлення кількох пов'язаних записів, застосовуються транзакції. Під час видалення курсів, глав та інших пов'язаних об'єктів система опрацьовує залежні записи відповідно до налаштованих правил цілісності. Для операцій, пов'язаних із завантаженими матеріалами, передбачено також видалення відповідних файлів із зовнішнього сховища.

До часових характеристик системи встановлено вимогу, відповідно до якої за стабільного підключення відкриття основних сторінок, каталогу та результатів фільтрування має виконуватися не довше трьох секунд. Збереження текстових даних повинно завершуватися орієнтовно протягом двох секунд після надходження запиту. Тривалість платежів, завантаження файлів та оброблення відео залежить від швидкості мережі й роботи зовнішніх сервісів, тому інтерфейс має повідомляти користувача про стан виконання операції.

Діаграму варіантів використання вебплатформи наведено на рис. 2.1.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				
						28



Рис. 2.1. Діаграма варіантів використання вебплатформи дистанційного навчання

На діаграмі відображено розподіл основних функціональних можливостей між студентом, викладачем та адміністратором. Детальні умови надання доступу, проходження тестування, формування сертифіката й опрацювання заявок визначено в текстовому описі вимог.

Об'єктно-орієнтовану структуру вебплатформи сформовано на основі основних сутностей предметної області та зв'язків між ними. Для опису сутностей використано моделі Prisma ORM, а операції опрацювання даних реалізовано за допомогою серверних функцій і обробників маршрутів. На узагальненій діаграмі наведено ключові атрибути класів без технічних ідентифікаторів, зовнішніх ключів і службових полів.

Клас Profile представляє локальний профіль користувача та містить ім'я, обов'язкову адресу електронної пошти і роль. Його зв'язки з класами Course, Purchase, UserProgress, Review, Certificate, QuizAttempt, VerificationRequest і SupportTicket реалізовано через поле userId. Тому на діаграмі ці зв'язки позначено пунктирними лініями.

Клас Category призначений для тематичного групування курсів. Одна категорія може об'єднувати декілька курсів, проте належність курсу до категорії не є обов'язковою під час створення чернетки. Центральним класом моделі є

Course, який містить назву, опис, ціну та ознаку публікації. Курс пов'язаний з автором, категорією, главами, записами доступу, відгуками, сертифікатами і підсумковим тестом.

Клас Chapter описує окрему главу курсу та містить її назву, позицію, стан публікації й ознаку можливості безоплатного перегляду. Кожна глава належить одному курсу та може бути пов'язана з навчальними ресурсами, записами прогресу і тестом. Клас ChapterResource представляє файл або зовнішнє посилання, додане до відповідної глави.

Клас Purchase відображає факт надання користувачеві платного, безоплатного або пільгового доступу до курсу. Клас UserProgress зберігає відомості про завершення глави та позицію відтворення відео. Клас Review містить оцінку курсу і необов'язковий текстовий коментар. Клас Certificate зберігає дату видачі сертифіката та унікальний код його перевірки. Для кожної пари користувача і курсу може бути створено не більше одного запису доступу, одного відгуку та одного сертифіката.

Структуру тестування утворюють класи Quiz, QuizQuestion, QuizOption і QuizAttempt. Тест може належати окремій главі або курсу як підсумковий тест. Клас QuizQuestion містить текст запитання та його позицію, а клас QuizOption зберігає текст варіанта відповіді й ознаку його правильності. Клас QuizAttempt фіксує результат проходження тесту, ознаку успішності та дату створення спроби.

Процес верифікації представлено класами VerificationRequest і VerificationDocument. Заявка містить тип і стан перевірки, а пов'язані документи зберігають ім'я файла, адресу, розмір і тип даних. Роботу технічної підтримки описують класи SupportTicket і SupportMessage. Звернення містить тему та поточний стан, а повідомлення забезпечують обмін інформацією між користувачем і адміністратором.

Допоміжні класи Attachment, MuxData, StripeCustomer та QuizAttemptAnswer не включено до узагальненої діаграми. Їх розглянуто під час опису структури бази даних.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				30

Узагальнену діаграму класів вебплатформи наведено на рис. 2.2.

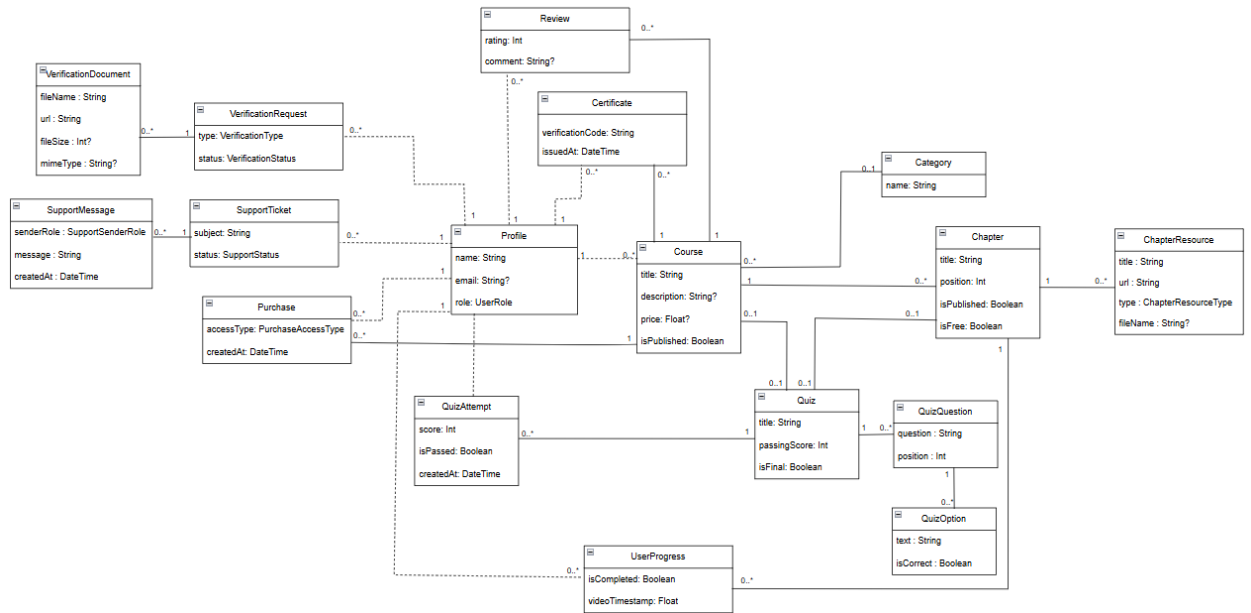


Рис. 2.2. Узагальнена діаграма класів вебплатформи дистанційного навчання

На діаграмі відображено ключові атрибути класів, кратності та зв'язки між сутностями. Суцільні лінії позначають прямі зв'язки між моделями, а пунктирні лінії відображають логічні зв'язки з класом Profile через поле userId.

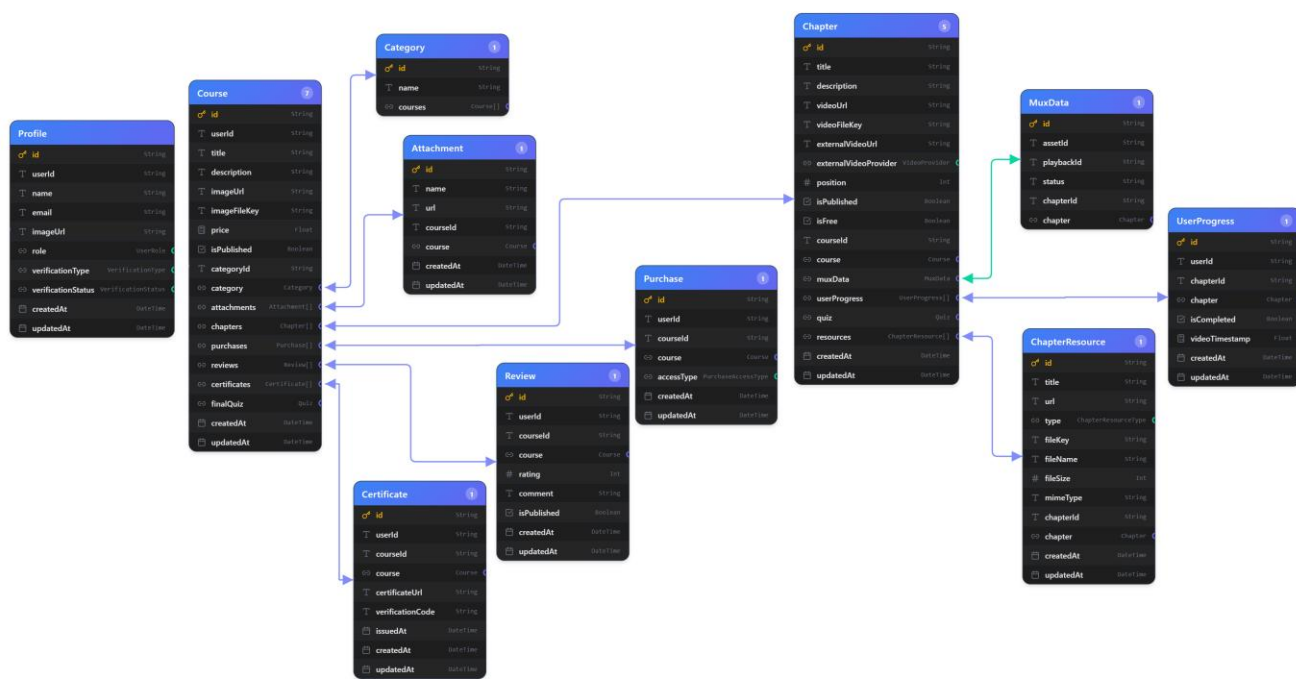
2.2. Розробка структури бази даних системи

Для зберігання даних вебплатформи використано реляційну систему керування базами даних PostgreSQL, яка підтримує зовнішні ключі, індекси, транзакції та обмеження цілісності [11]. Взаємодію серверної частини з базою даних реалізовано за допомогою Prisma ORM. Моделі описано у схемі Prisma, а операції з даними виконуються через типізований Prisma Client [10].

База даних містить відомості про профілі користувачів, курси, глави, навчальні матеріали, доступи, прогрес, результати тестування, відгуки, сертифікати, заявки на верифікацію та звернення до технічної підтримки. Зв'язки між таблицями й обмеження цілісності дають змогу уникнути зайвого дублювання та підтримувати узгодженість пов'язаних записів.

Через значну кількість таблиць фізичну структуру бази даних поділено на три частини. Перша містить таблиці навчального контенту, доступу до курсів і

Схему таблиц навчального контенту, доступу та результатів навчання наведено на рис. 2.3.



Основними таблицями цієї частини є «Course» і «Chapter». У таблиці «Course» зберігаються відомості про навчальний курс, його автора, категорію, вартість і стан публікації. З нею пов'язані таблиці «Category», «Attachment», «Chapter», «Purchase», «Review» і «Certificate».

Таблиця «Purchase» містить дані про платний, безоплатний або пільговий доступ користувача до курсу. Оцінки та коментарі до курсів зберігаються в таблиці «Review». Таблиця «Certificate» містить відомості про сертифікати, сформовані після завершення навчання.

Таблиця «Profile» зберігає локальні відомості про користувача, зокрема його ім'я, роль і стан верифікації. Таблиці «Course», «Purchase», «Review», «Certificate» і «UserProgress» містять поле `userId` з ідентифікатором користувача Clerk. Це поле не є зовнішнім ключем до таблиці «Profile», тому відповідні зв'язки на фізичній схемі не показано.

Схему таблиць, що використовуються під час створення та проходження тестів, наведено на рис. 2.4.

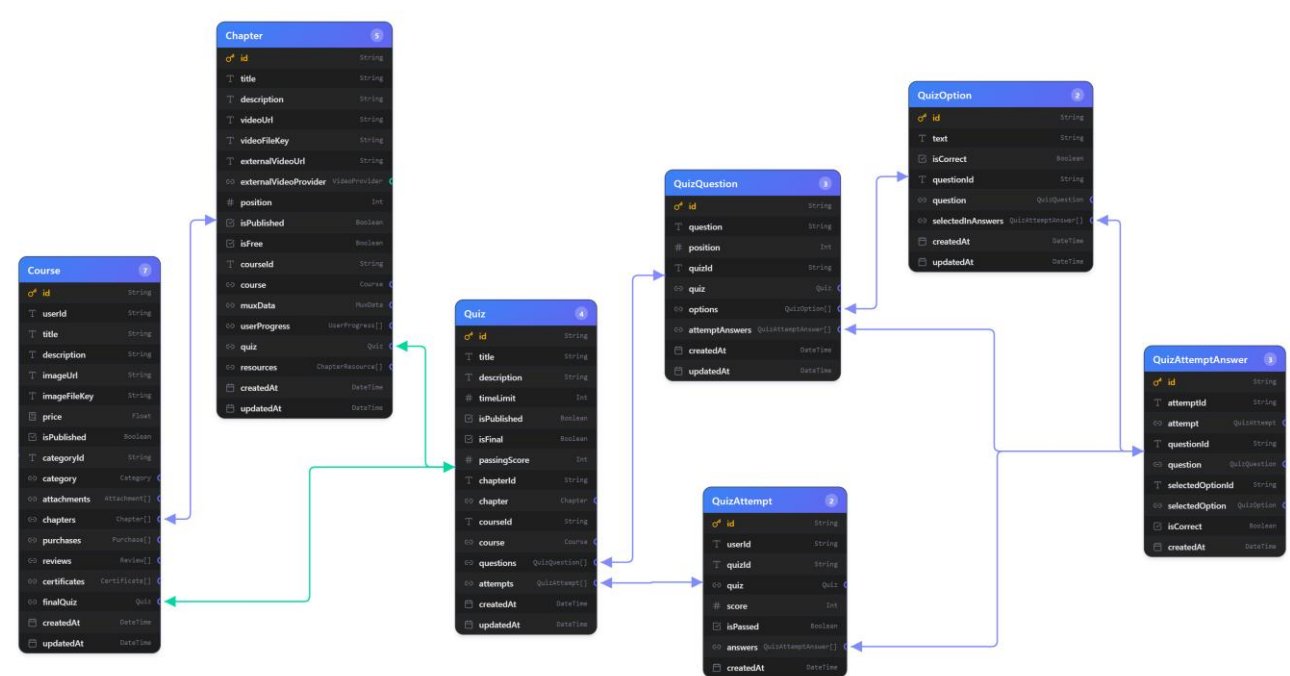


Рис. 2.4. Схема таблиц тестування вебплатформи

Тест може належати окремій главі або бути підсумковим тестом курсу. Основні відомості про тест зберігаються в таблиці «Quiz». Таблиця «QuizQuestion» містить запитання тесту, а таблиця «QuizOption» зберігає варіанти відповідей і ознаку правильного варіанта.

Результат проходження тесту зберігається в таблиці «QuizAttempt». Вибрані користувачем відповіді записуються в таблицю «QuizAttemptAnswer», яка пов'язана зі спробою тестування, запитанням і вибраним варіантом відповіді. Поле `userId` у таблиці «QuizAttempt» містить ідентифікатор користувача Clerk і не є зовнішнім ключем до таблиці «Profile».

Схему таблиць верифікації та технічної підтримки наведено на рис. 2.5.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				
						33

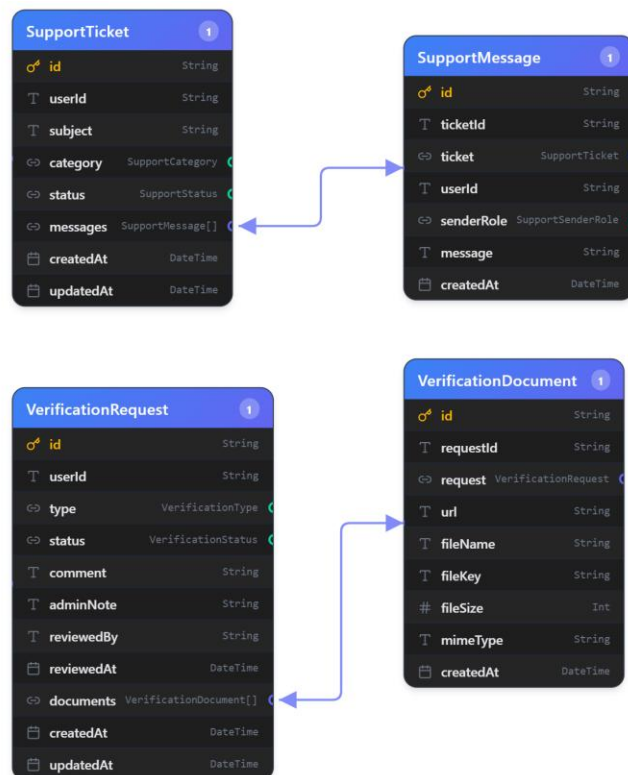


Рис. 2.5. Схема таблиц верифікації та технічної підтримки вебплатформи

Заявки на підтвердження статусу користувача зберігаються в таблиці «VerificationRequest». Для кожної заявки фіксуються тип верифікації, стан розгляду, коментар користувача, примітка адміністратора та дата прийняття рішення. Відомості про завантажені документи зберігаються в таблиці «VerificationDocument». Одна заявка може містити декілька документів.

Звернення користувачів до технічної підтримки зберігаються в таблиці «SupportTicket». Вона містить тему, категорію та поточний стан звернення. Повідомлення користувача й адміністратора записуються в таблицю «SupportMessage».

Таблиці «VerificationRequest», «SupportTicket» і «SupportMessage» містять поле userId з ідентифікатором користувача Clerk. Оскільки воно не є зовнішнім ключем до таблиці «Profile», зв'язок із профілем на фізичній схемі не відображено.

Первинні ключі використовуються для однозначного визначення записів, а зовнішні ключі встановлюють зв'язки між таблицями. Унікальні обмеження

запобігають створенню дубльованих записів, а індекси пришвидшують пошук і вибірку даних.

Опис полів таблиць бази даних наведено в табл. 2.1–2.21. Відомості про локальні профілі користувачів, їхні ролі та стан верифікації зберігаються в таблиці «Profile», структуру якої наведено в табл. 2.1.

Таблиця 2.1

Структура таблиці «Profile»

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор профілю у форматі UUID
userId	text	–	–	Унікальний ідентифікатор користувача в системі Clerk
name	text	–	–	Ім'я користувача
email	text	–	–	Електронна адреса користувача, може бути відсутньою
imageUrl	text	–	–	URL-адреса зображення профілю, може бути відсутньою
role	UserRole	–	–	Роль користувача в системі
verificationType	VerificationType	–	–	Тип верифікації користувача, може бути відсутнім
verificationStatus	VerificationStatus	–	–	Поточний стан верифікації
createdAt	timestamp(3)	–	–	Дата та час створення профілю
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення профілю

Поле `userId` має унікальне обмеження, що унеможливлює створення кількох локальних профілів для одного користувача Clerk.

Основні відомості про навчальні курси, їх авторів, категорії, вартість і стан публікації зберігаються в таблиці «Course». Структуру цієї таблиці наведено в табл. 2.2.

Таблиця 2.2

Структура таблиці «Course»

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор курсу у форматі UUID
userId	text	–	–	Ідентифікатор автора курсу в системі Clerk
title	text	–	–	Назва навчального курсу
description	text	–	–	Опис навчального курсу, може бути відсутнім
imageUrl	text	–	–	URL-адреса зображення курсу, може бути відсутньою
imageFileKey	text	–	–	Ключ зображення у файловому сховищі, може бути відсутнім
price	double precision	–	–	Вартість курсу, може бути відсутньою
isPublished	boolean	–	–	Ознака публікації курсу
categoryId	text	–	+	Ідентифікатор категорії курсу, може бути відсутнім
createdAt	timestamp(3)	–	–	Дата та час створення курсу
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення курсу

Файли, прикріплені до навчального курсу, зберігаються в таблиці «Attachment». Опис її полів наведено в табл. 2.3.

Таблиця 2.3

Структура таблиці «Attachment»

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор вкладення
name	text	–	–	Назва вложеного файла
url	text	–	–	URL-адреса вложеного файла
courseId	text	–	+	Ідентифікатор курсу
createdAt	timestamp(3)	–	–	Дата та час створення вкладення

продовж. табл. 2.3

1	2	3	4	5
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення вкладення

Категорії використовуються для тематичного групування навчальних курсів і подальшої фільтрації матеріалів у каталозі. Структуру таблиці «Category» наведено в табл. 2.4.

Таблиця 2.4

Структура таблиці «Category»

Назва	Тип даних	РК	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор категорії у форматі UUID
name	text	–	–	Унікальна назва категорії курсів

Глави є основними структурними елементами курсу та містять навчальний контент, відеоматеріали й параметри доступу. Структуру таблиці «Chapter» наведено в табл. 2.5.

Таблиця 2.5

Структура таблиці «Chapter»

Назва	Тип даних	РК	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор глави
title	text	–	–	Назва глави
description	text	–	–	Опис глави, може бути відсутнім
videoUrl	text	–	–	URL-адреса відеоматеріалу, може бути відсутньою
videoFileKey	text	–	–	Ключ відеофайла у файловому сховищі, може бути відсутнім
externalVideoUrl	text	–	–	URL-адреса зовнішнього відео, може бути відсутньою
externalVideoProvider	VideoProvider	–	–	Постачальник зовнішнього відео, може бути відсутнім

продовж. табл. 2.5

1	2	3	4	5
externalVideoUrl	text	–	–	URL-адреса зовнішнього відео, може бути відсутньою
externalVideoProvider	VideoProvider	–	–	Постачальник зовнішнього відео, може бути відсутнім
position	integer	–	–	Порядкова позиція глави в курсі
isPublished	boolean	–	–	Ознака публікації глави
isFree	boolean	–	–	Ознака безоплатного перегляду
courseId	text	–	+	Ідентифікатор курсу
createdAt	timestamp(3)	–	–	Дата та час створення глави
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення глави

Відомості про відеодані глави, що використовуються для відтворення матеріалів через сервіс Mux, зберігаються в окремій таблиці. Структуру таблиці «MuxData» наведено в табл. 2.6.

Таблиця 2.6

Структура таблиці «MuxData»

Назва	Тип даних	РК	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор запису відеоданих
assetId	text	–	–	Ідентифікатор відеокурсу в сервісі Mux
playbackId	text	–	–	Ідентифікатор відтворення відео, може бути відсутнім
status	text	–	–	Стан оброблення відео, може бути відсутнім
chapterId	text	–	+	Ідентифікатор глави

Поле chapterId є зовнішнім ключем до таблиці «Chapter» і має унікальне обмеження, що забезпечує зв'язок «один до одного» між главою та відеоданими Mux.

Додаткові навчальні ресурси до глав зберігаються окремо від відеоматеріалів і можуть містити файли або зовнішні посилання. Структуру таблиці «ChapterResource» наведено в табл. 2.7.

Таблиця 2.7

Структура таблиці «ChapterResource»

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор навчального ресурсу
title	text	–	–	Назва навчального ресурсу
url	text	–	–	URL-адреса файла або зовнішнього посилання
type	ChapterResourceType	–	–	Тип навчального ресурсу
fileKey	text	–	–	Ключ файла у файловому сховищі, може бути відсутнім
fileName	text	–	–	Назва файла, може бути відсутньою
fileSize	integer	–	–	Розмір файла у байтах, може бути відсутнім
mimeType	text	–	–	MIME-тип файла, може бути відсутнім
chapterId	text	–	+	Ідентифікатор глави
createdAt	timestamp(3)	–	–	Дата та час створення ресурсу
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення ресурсу

Для фіксації проходження студентом окремих глав і збереження позиції перегляду відео використовується таблиця прогресу користувача. Структуру таблиці «UserProgress» наведено в табл. 2.8.

Таблиця 2.8

Структура таблиці «UserProgress»

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор запису прогресу
userId	text	–	–	Ідентифікатор користувача в системі Clerk

продовж. табл. 2.8

1	2	3	4	5
chapterId	text	–	+	Ідентифікатор глави
isCompleted	boolean	–	–	Ознака завершення глави
videoTimestamp	double precision	–	–	Остання збережена позиція відтворення відео
createdAt	timestamp(3)	–	–	Дата та час створення запису
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення прогресу

Поєднання полів `userId` і `chapterId` має унікальне обмеження, що запобігає дублюванню записів прогресу для однієї глави.

Доступ користувачів до курсів після купівлі, безоплатного надання або підтвердження пільгового статусу зберігається в таблиці «Purchase». Структуру таблиці наведено в табл. 2.9.

Таблиця 2.9

Структура таблиці «Purchase»

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор запису доступу
userId	text	–	–	Ідентифікатор користувача в системі Clerk
courseId	text	–	+	Ідентифікатор курсу
accessType	PurchaseAccessType	–	–	Тип доступу до курсу
createdAt	timestamp(3)	–	–	Дата та час створення запису
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення запису
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення запису

Поєднання полів `userId` і `courseId` має унікальне обмеження, що унеможливорює створення кількох записів доступу одного користувача до одного курсу.

Заявки користувачів на підтвердження пільгового статусу зберігаються в таблиці «VerificationRequest». Структуру цієї таблиці наведено в табл. 2.10.

Таблиця 2.10

Структура таблиці «VerificationRequest»

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор заявки
userId	text	–	–	Ідентифікатор користувача в системі Clerk
type	VerificationType	–	–	Тип верифікації
status	VerificationStatus	–	–	Стан розгляду заявки
comment	text	–	–	Коментар користувача, може бути відсутнім
adminNote	text	–	–	Примітка адміністратора, може бути відсутньою
reviewedBy	text	–	–	Ідентифікатор адміністратора, який розглянув заявку, може бути відсутнім
reviewedAt	timestamp(3)	–	–	Дата та час розгляду заявки, може бути відсутньою.
createdAt	timestamp(3)	–	–	Дата та час створення заявки
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення заявки

Для зв'язку користувача з платіжною системою Stripe використовується окрема таблиця клієнтів платіжного сервісу. Структуру таблиці «StripeCustomer» наведено в табл. 2.11.

Таблиця 2.11

Структура таблиці «StripeCustomer»

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор запису
userId	text	–	–	Ідентифікатор користувача в системі Clerk
stripeCustomerId	text	–	–	Ідентифікатор клієнта в системі Stripe
createdAt	timestamp(3)	–	–	Дата та час створення запису
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення запису

Кольцова Н.О.

Савіцький Р.С.

ІПЗ.КР.Б-121-26-ПЗ

Арк.

41

продовж. табл. 2.11

1	2	3	4	5
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення запису

Поля `userId` і `stripeCustomerId` мають унікальні обмеження, що забезпечує однозначну відповідність між користувачем платформи та клієнтом у Stripe.

Документи, прикріплені до заявки на верифікацію, зберігаються окремо та пов'язуються з відповідною заявкою. Структуру таблиці «VerificationDocument» наведено в табл. 2.12.

Таблиця 2.12

Структура таблиці «VerificationDocument»

Назва	Тип даних	РК	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор документа
requestId	text	–	+	Ідентифікатор заявки на верифікацію
url	text	–	–	URL-адреса завантаженого документа
fileName	text	–	–	Назва файлу
fileKey	text	–	–	Ключ файлу у файловому сховищі, може бути відсутнім
fileSize	integer	–	–	Розмір файлу у байтах, може бути відсутнім
mimeType	text	–	–	MIME-тип файлу, може бути відсутнім
createdAt	timestamp(3)	–	–	Дата та час завантаження документа

Відгуки студентів використовуються для зберігання оцінок курсів і текстових коментарів до навчальних матеріалів. Структуру таблиці «Review» наведено в табл. 2.13.

Таблиця 2.13

Структура таблиці «Review»

Назва	Тип даних	РК	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор відгуку
userId	text	–	–	Ідентифікатор автора відгуку

продовж. табл. 2.13

1	2	3	4	5
courseId	text	–	+	Ідентифікатор курсу
rating	integer	–	–	Оцінка курсу
comment	text	–	–	Текст відгуку, може бути відсутнім
isPublished	boolean	–	–	Ознака публікації відгуку
createdAt	timestamp(3)	–	–	Дата та час створення відгуку
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення відгуку

Поєднання полів `userId` і `courseId` має унікальне обмеження, що дозволяє одному користувачу залишити лише один відгук до конкретного курсу. Після успішного завершення курсу система може створювати сертифікат для користувача. Структуру таблиці «Certificate» наведено в табл. 2.14.

Таблиця 2.14

Структура таблиці «Certificate»

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор сертифіката
userId	text	–	–	Ідентифікатор власника сертифіката
courseId	text	–	+	Ідентифікатор завершеного курсу
certificateUrl	text	–	–	URL-адреса файлу сертифіката, може бути відсутньою
verificationCode	text	–	–	Код перевірки справжності сертифіката
issuedAt	timestamp(3)	–	–	Дата та час видачі сертифіката
createdAt	timestamp(3)	–	–	Дата та час створення запису
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення запису

Поле `verificationCode` та поєднання полів `userId` і `courseId` мають унікальні обмеження, що забезпечує унікальність сертифіката та його перевірного коду.

Тести використовуються для перевірки знань студентів після проходження окремих глав або всього курсу. Структуру таблиці «Quiz» наведено в табл. 2.15.

Таблиця 2.15

Структура таблиці «Quiz»

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор тесту
title	text	–	–	Назва тесту
description	text	–	–	Опис тесту, може бути відсутнім
timeLimit	integer	–	–	Обмеження часу проходження у хвилинах, може бути відсутнім
isPublished	boolean	–	–	Ознака публікації тесту
isFinal	boolean	–	–	Ознака підсумкового тесту
passingScore	integer	–	–	Мінімальний прохідний бал
chapterId	text	–	+	Ідентифікатор глави, може бути відсутнім
courseId	text	–	+	Ідентифікатор курсу, може бути відсутнім
createdAt	timestamp(3)	–	–	Дата та час створення тесту
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення тесту

Поля chapterId і courseId можуть мати значення NULL та мають окремі унікальні обмеження. Належність тесту главі або курсу контролюється бізнес-логікою застосунку.

Запитання тестів зберігаються окремо від загальної інформації про тест і пов'язуються з ним за допомогою зовнішнього ключа. Структуру таблиці «QuizQuestion» наведено в табл. 2.16.

Таблиця 2.16

Структура таблиці «QuizQuestion»

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор запитання
question	text	–	–	Текст запитання
position	integer	–	–	Порядкова позиція запитання в тесті

продовж табл. 2.16

1	2	3	4	5
quizId	text	–	+	Ідентифікатор тесту
createdAt	timestamp(3)	–	–	Дата та час створення запитання
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення запитання

Для кожного запитання тесту передбачено набір варіантів відповідей, серед яких може бути позначено правильний варіант. Структуру таблиці «QuizOption» наведено в табл. 2.17.

Таблиця 2.17

Структура таблиці “QuizOption”

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор варіанта відповіді
text	text	–	–	Текст варіанта відповіді
isCorrect	boolean	–	–	Ознака правильності варіанта
questionId	text	–	+	Ідентифікатор запитання
createdAt	timestamp(3)	–	–	Дата та час створення варіанта
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення варіанта

Відповіді користувача під час проходження тесту зберігаються для фіксації результатів і подальшого перегляду спроби. Структуру таблиці «QuizAttemptAnswer» наведено в табл. 2.18.

Таблиця 2.18

Структура таблиці “QuizAttemptAnswer”

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор відповіді
attemptId	text	–	+	Ідентифікатор спроби тестування
questionId	text	–	+	Ідентифікатор запитання
selectedOption Id	text	–	+	Ідентифікатор вибраного варіанта відповіді, може бути відсутнім

продовж. табл. 2.18

1	2	3	4	5
isCorrect	boolean	–	–	Ознака правильності відповіді
createdAt	timestamp(3)	–	–	Дата та час створення запису

Поле selectedOptionId може мати значення NULL. Після видалення варіанта відповіді запис про результат тестування зберігається.

Загальний результат проходження тесту користувачем фіксується у вигляді окремої спроби тестування. Структуру таблиці «QuizAttempt» наведено в табл. 2.19.

Таблиця 2.19

Структура таблиці «QuizAttempt»

Назва	Тип даних	РК	ФК	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор спроби
userId	text	–	–	Ідентифікатор користувача в системі Clerk
quizId	text	–	+	Ідентифікатор тесту
score	integer	–	–	Отриманий результат тестування
isPassed	boolean	–	–	Ознака успішного проходження тесту
createdAt	timestamp(3)	–	–	Дата та час створення спроби

Звернення користувачів до технічної підтримки зберігаються в окремій таблиці та містять тему, категорію і поточний стан опрацювання. Структуру таблиці «SupportTicket» наведено в табл. 2.20.

Таблиця 2.20

Структура таблиці «SupportTicket»

Назва	Тип даних	РК	ФК	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор звернення
userId	text	–	–	Ідентифікатор автора звернення
subject	text	–	–	Тема звернення
category	SupportCategory	–	–	Категорія звернення
status	SupportStatus	–	–	Поточний стан звернення

продовж. табл. 2.20

1	2	3	4	5
createdAt	timestamp(3)	–	–	Дата та час створення звернення
updatedAt	timestamp(3)	–	–	Дата та час останнього оновлення звернення

Повідомлення в межах звернення до технічної підтримки зберігаються окремо та пов'язуються з відповідною заявкою. Структуру таблиці «SupportMessage» наведено в табл. 2.21.

Таблиця 2.21

Структура таблиці «SupportMessage»

Назва	Тип даних	PK	FK	Опис
1	2	3	4	5
id	text	+	–	Унікальний ідентифікатор повідомлення
ticketId	text	–	+	Ідентифікатор звернення
userId	text	–	–	Ідентифікатор автора повідомлення
senderRole	SupportSenderRole	–	–	Роль відправника повідомлення
message	text	–	–	Текст повідомлення
createdAt	timestamp(3)	–	–	Дата та час створення повідомлення

Створення, отримання, оновлення та видалення записів виконуються за допомогою Prisma Client. Зв'язки між таблицями реалізовано через зовнішні ключі. Для залежних записів, які не можуть існувати без батьківської сутності, налаштовано каскадне видалення. Унікальні обмеження, індекси й транзакції використовуються для підтримання узгодженості даних та ефективного виконання запитів [10, 11].

Індекси створено для полів, які використовуються під час пошуку, фільтрування та об'єднання даних. Поля userId зберігають ідентифікатори користувачів зовнішньої системи Clerk і не є зовнішніми ключами до таблиці Profile.

2.3. Проєктування та реалізація алгоритмів роботи системи

Алгоритми функціонування вебплатформи спроектовано на основі типових сценаріїв її використання, визначених у підрозділі 2.1. Вони відображають послідовність дій користувачів, перевірки, які виконує серверна частина, звернення до бази даних та взаємодію із зовнішніми сервісами.

Для графічного подання алгоритмів використано UML-діаграми активності та діаграму послідовності. Діаграми активності дають змогу показати порядок виконання операцій, розподіл відповідальності між учасниками процесу, умови переходів та альтернативні результати. Діаграма послідовності відображає обмін повідомленнями між користувачем, вебінтерфейсом, серверною частиною, платіжним сервісом і базою даних.

У підрозділі розглянуто алгоритми створення та публікації навчального курсу, подання й опрацювання заявки на верифікацію, проходження курсу та отримання сертифіката, а також оплату курсу і надання користувачеві доступу до навчального контенту.

Створення навчального курсу виконує користувач із роллю викладача. Перед створенням курсу серверна частина перевіряє наявність активної сесії та роль користувача. Під час редагування або публікації додатково перевіряється право користувача керувати відповідним курсом. Новий курс зберігається в базі даних як неопублікований запис, для якого поле `isPublished` має значення `false`. Завдяки цьому незавершений курс не відображається в загальному каталозі та доступний лише його автору й адміністратору.

Викладач заповнює назву й опис курсу, вибирає категорію, встановлює вартість і додає обкладинку. Після цього формується структура навчального контенту: створюються глави, до них додаються відеоматеріали, навчальні ресурси та, за потреби, тести. Для публікації курсу необхідно підготувати й опублікувати щонайменше одну главу та підсумковий тест.

Після натискання кнопки публікації серверна частина повторно перевіряє право користувача керувати курсом і визначає повноту його наповнення. Перевіряється наявність основних даних, обкладинки, категорії, вартості,

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				48

опублікованої глави та підсумкового тесту. Якщо виявлено недоліки, система формує повідомлення з переліком необхідних виправлень і повертає курс викладачеві для редагування.

Після внесення змін викладач повторно подає курс на публікацію. Якщо всі умови виконано, серверна частина змінює значення поля isPublished на true, зберігає зміни в базі даних, після чого курс стає доступним у загальному каталозі. Адміністративна модерація перед публікацією не використовується, оскільки готовність курсу визначається автоматичними перевірками системи.

Алгоритм створення та публікації навчального курсу наведено на рис. 2.6.

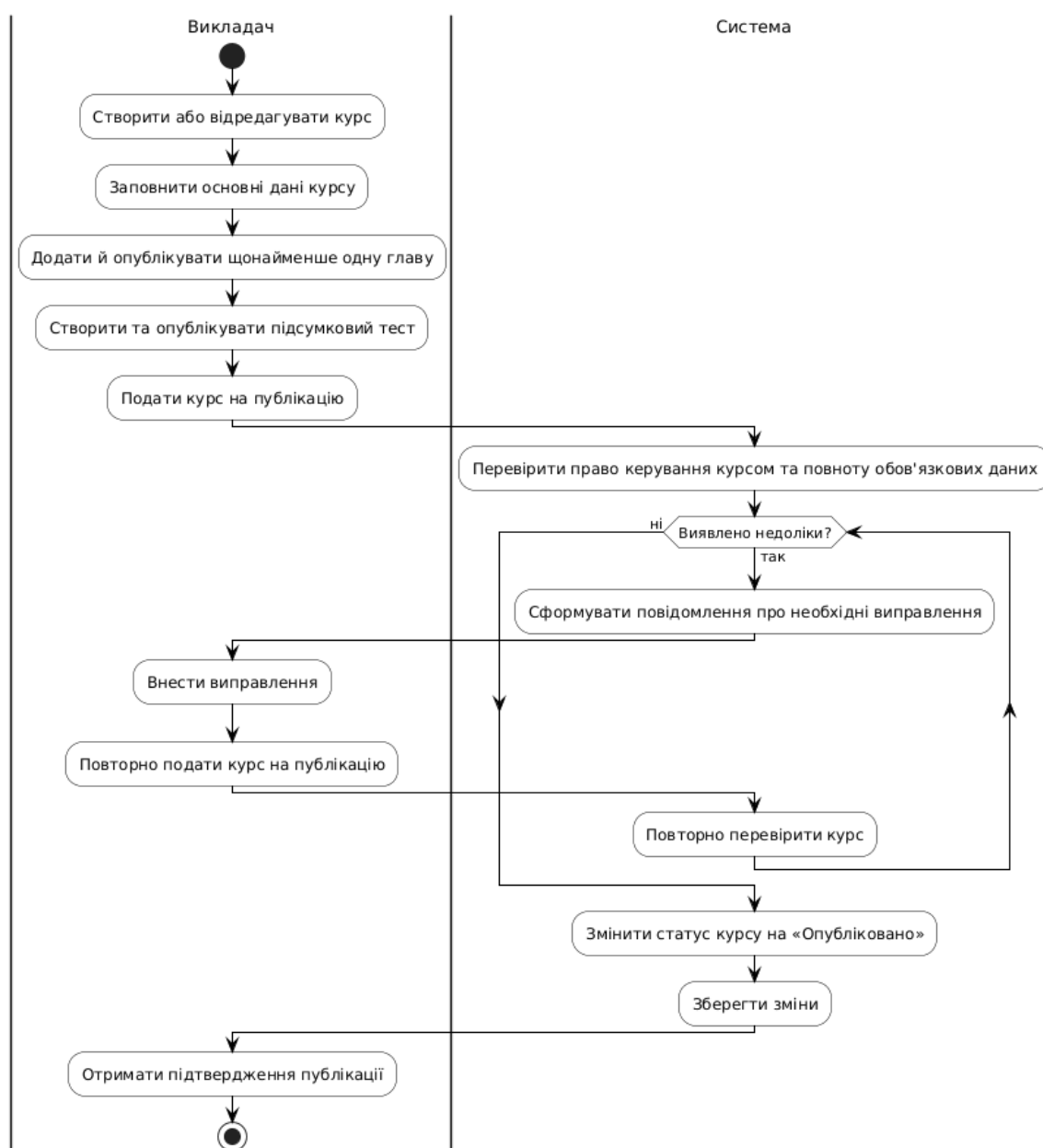


Рис. 2.6. Діаграма активності створення та публікації курсу

На діаграмі дії розподілено між двома доріжками: «Викладач» і «Система». Такий поділ демонструє, що викладач відповідає за формування навчального контенту, а система - за перевірку прав доступу, валідацію даних, зміну стану курсу та його відображення в каталозі.

Окремим процесом вебплатформи є верифікація користувачів, які можуть претендувати на пільговий доступ до навчальних курсів. Перед поданням заявки система перевіряє автентифікацію користувача та поточний стан його профілю. Користувач вибирає тип верифікації, заповнює необхідні відомості й додає підтвердні документи.

Після надсилання заявки серверна частина перевіряє повноту введених даних і наявність документів. Якщо інформація неповна, користувач отримує повідомлення про необхідність доповнення заявки. За умови успішної перевірки створюється запис `VerificationRequest` зі станом `PENDING`, а відомості про завантажені файли зберігаються в пов'язаних записах `VerificationDocument`.

Подана заявка стає доступною адміністратору. Під час її розгляду адміністратор перевіряє дані користувача та підтвердні документи, після чого може схвалити або відхилити заявку. Рішення супроводжується збереженням службової примітки та відомостей про особу, яка виконала перевірку.

У разі схвалення стан заявки змінюється на `APPROVED`, а відповідний стан верифікації записується в профілі користувача. У разі відхилення встановлюється стан `REJECTED`. Оновлення заявки та профілю виконується в межах транзакції, щоб не допустити розбіжності між станами пов'язаних записів. Транзакційне виконання операцій підтримується засобами Prisma ORM [10].

Користувач може повторно відкрити сторінку верифікації та переглянути результат розгляду. Підтвердження статусу саме по собі не відкриває всі курси автоматично, а лише дає змогу оформити пільговий доступ до вибраного навчального курсу.

Алгоритм подання та опрацювання заявки на верифікацію наведено на рис. 2.7.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				50

запис `UserProgress`. Якщо студент уже переглядав відео, відтворення продовжується з останньої збереженої позиції.

Під час навчання система періодично зберігає поточну позицію відтворення відео. Для кожної пари користувача і глави створюється один запис прогресу, який оновлюється під час подальшого перегляду. Після опрацювання матеріалів студент позначає главу як завершену, а система оновлює поле `isCompleted` і перераховує загальний прогрес курсу.

Якщо до глави додано тест, студент відповідає на всі запитання й надсилає результат на перевірку. Серверна частина визначає правильність вибраних варіантів, обчислює відсотковий результат і порівнює його з установленим прохідним балом. Відомості про проходження зберігаються в таблицях `QuizAttempt` і `QuizAttemptAnswer`, завдяки чому результат не втрачається після перезавантаження сторінки.

Якщо тест не складено, студент може повторити спробу. Після завершення всіх опублікованих глав стає доступним підсумковий тест курсу. Його перевірка виконується за аналогічним алгоритмом.

Після успішного складання підсумкового тесту система перевіряє завершення всіх необхідних глав і наявність успішної спроби тестування. Якщо умови виконано, створюється запис `Certificate` з унікальним кодом перевірки. Якщо сертифікат уже сформовано, система повертає наявний запис і не створює дублікату.

Студент може переглянути та завантажити сформований сертифікат. Його створення підтверджує завершення навчального курсу й успішне проходження підсумкового контролю.

Алгоритм проходження курсу та отримання сертифіката наведено на рис. 2.8.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				
						52

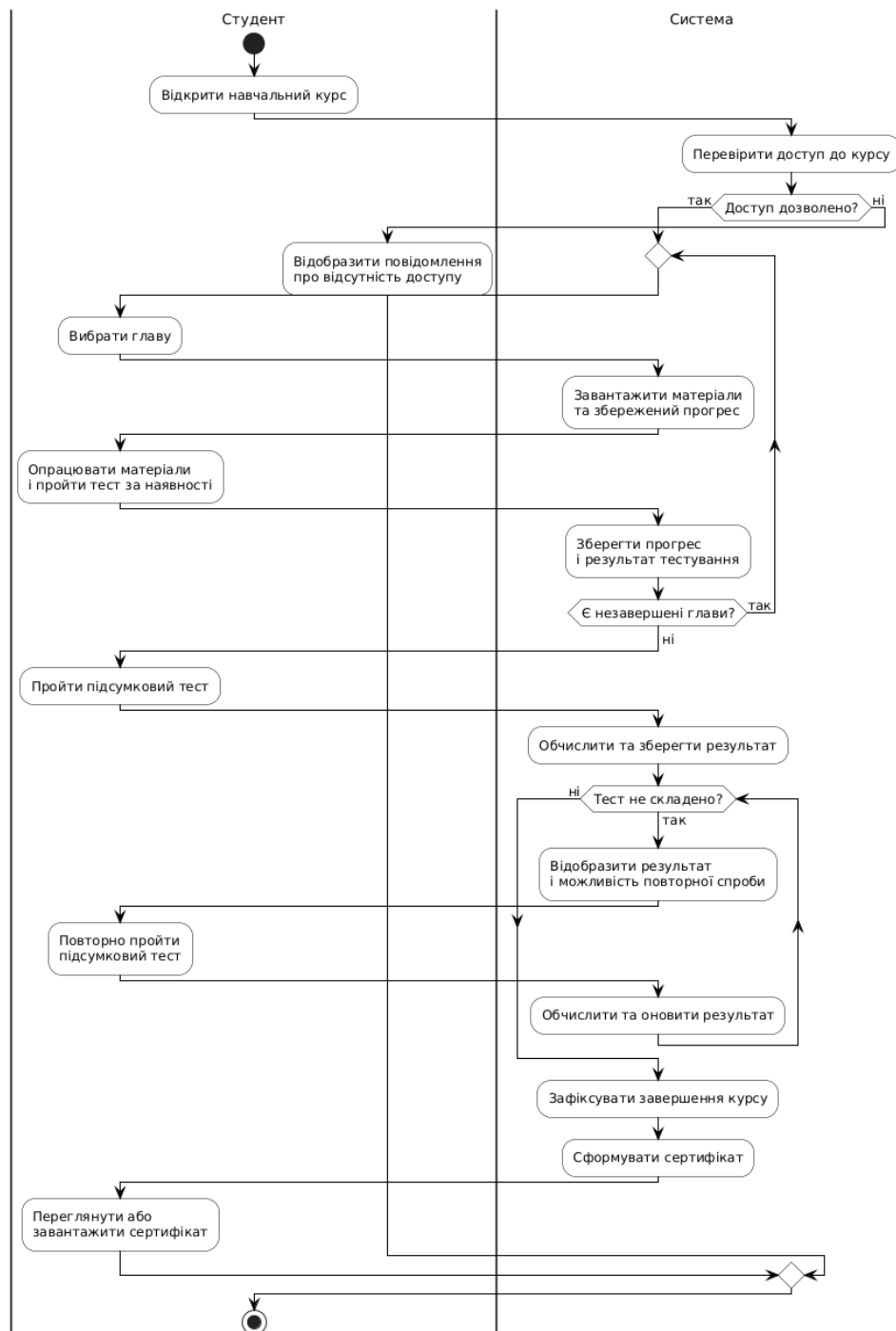


Рис. 2.8. Діаграма активності проходження курсу та отримання сертифіката

На діаграмі показано перевірку доступу, завантаження матеріалів, збереження прогресу, проходження тестування, можливість повторної спроби та формування сертифіката. Циклічність процесу відображає послідовне проходження студентом усіх глав навчального курсу.

Для платних курсів доступ надається після успішного завершення оплати через Stripe Checkout. Stripe Checkout забезпечує створення захищеної платіжної сесії, а остаточне підтвердження результату оплати виконується за допомогою підписаного повідомлення вебхука [14]. Процес починається з вибору студентом команди придбання курсу. Вебінтерфейс передає серверній частині запит на оформлення доступу, після чого система перевіряє, чи не існує вже запис Purchase для цього користувача й курсу.

Якщо доступ був наданий раніше, система не створює нову платіжну сесію, а повертає підтвердження наявності придбаного курсу. Студент одразу може перейти до навчання, а така перевірка запобігає повторній оплаті одного й того самого курсу. Якщо запис доступу відсутній, серверна частина отримує наявний або створює новий запис StripeCustomer, пов'язаний із користувачем, формує платіжну сесію та передає вебінтерфейсу адресу сторінки оплати

Успішне повернення користувача зі сторінки Stripe саме по собі не використовується як підтвердження оплати. Остаточне рішення приймається після отримання серверною частиною повідомлення вебхука Stripe. Система перевіряє цифровий підпис повідомлення, стан платежу, ідентифікатор користувача та перелік придбаних курсів [14].

Після успішної перевірки в базі даних створюється запис Purchase з типом доступу PAID. Унікальне обмеження для полів userId і courseId не допускає дублювання доступу в разі повторного надходження webhook.

Після повернення на сторінку курсу вебінтерфейс повторно запитує стан доступу. Серверна частина перевіряє наявність запису Purchase і, якщо його знайдено, дозволяє розпочати навчання. Якщо платіж скасовано або відхилено, запис доступу не створюється, а користувач отримує повідомлення про незавершену оплату.

Взаємодію компонентів під час оплати курсу та надання доступу наведено на рис. 2.9.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				
						54

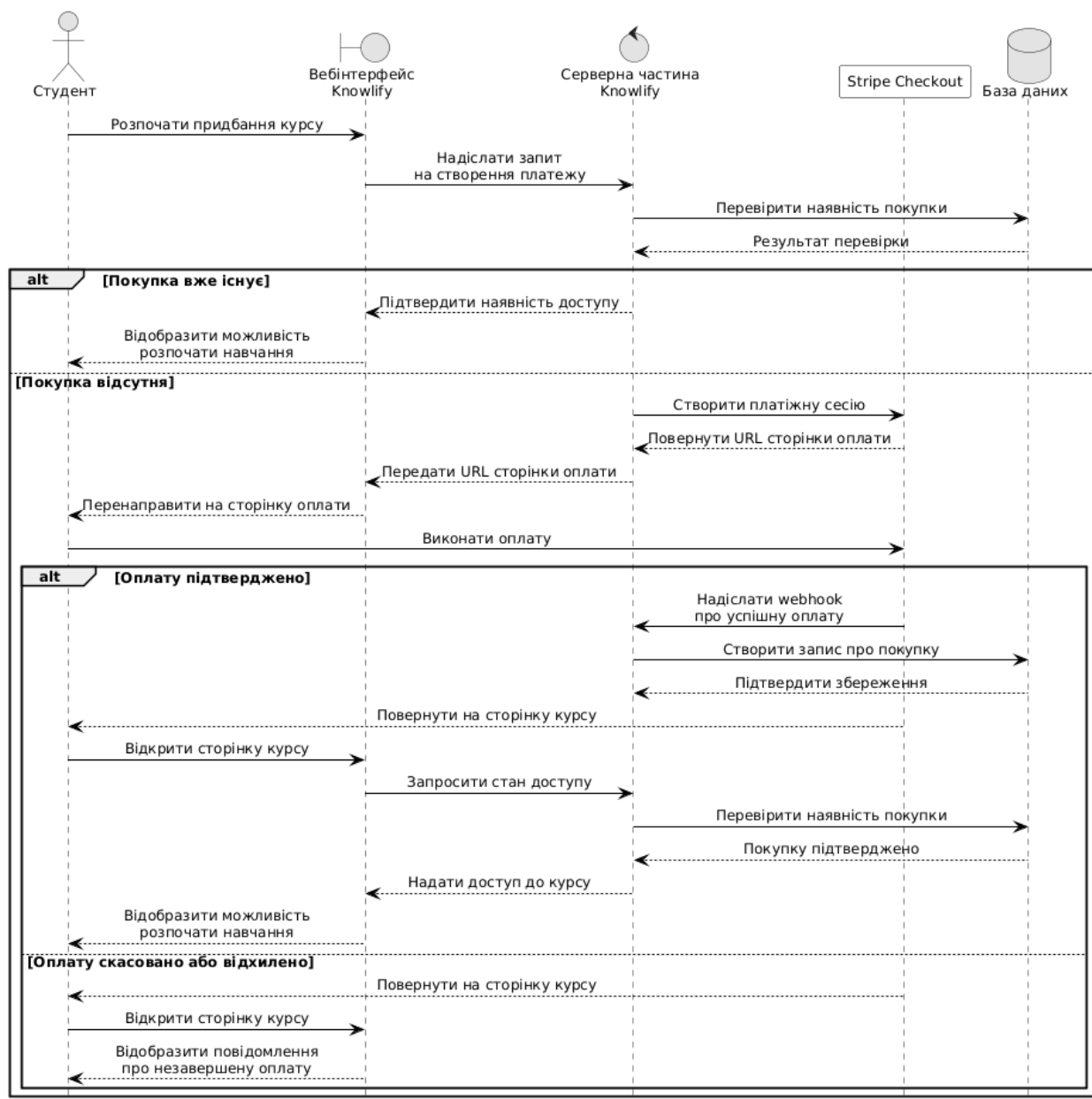


Рис. 2.9. Діаграма послідовності оплати курсу та надання доступу

На діаграмі показано взаємодію студента з вебінтерфейсом платформи, серверною частиною, сервісом Stripe Checkout і базою даних. Відображено випадки, коли доступ до курсу вже надано, коли користувачеві потрібно перейти до оплати, а також результати успішного або скасованого платежу.

Розроблені алгоритми охоплюють основні процеси вебплатформи: створення та публікацію курсів, подання і розгляд заявок на верифікацію, проходження навчання, збереження прогресу, тестування, формування сертифікатів та оплату курсів. UML-діаграми відображають послідовність дій користувачів і системи, основні перевірки та можливі результати виконання

операцій. Визначені алгоритми стали основою для реалізації функціональних модулів вебплатформи.

2.4. Реалізація функціональних модулів вебплатформи

Вебплатформу реалізовано як модульний повностековий вебзастосунок на основі Next.js із використанням маршрутизатора App Router. В одному проєкті поєднано сторінки, компоненти інтерфейсу, серверні обробники та засоби роботи з даними [6]

Користувачський інтерфейс побудовано за допомогою серверних і клієнтських компонентів React. Серверні компоненти отримують дані та формують початковий вміст сторінок, а клієнтські забезпечують роботу інтерактивних форм, кошика, відеопрогравача й тестових завдань [7].

Оброблення запитів виконується за допомогою серверних функцій та обробників маршрутів Next.js. Перед виконанням захищеної операції сервер перевіряє активну сесію, роль користувача, право доступу до відповідного ресурсу та коректність отриманих даних. Робота з базою даних PostgreSQL здійснюється через Prisma Client [10, 11].

Функціональні можливості вебплатформи розподілено між модулями автентифікації, керування навчальним контентом, доступу до курсів, збереження прогресу, тестування, оплати, сертифікації, верифікації, аналітики та технічної підтримки. Кожен модуль відповідає за окрему групу операцій і використовує спільні засоби перевірки користувачів та роботи з базою даних.

Для окремих функцій підключено зовнішні сервіси. Clerk забезпечує автентифікацію та керування сесіями, а Stripe використовується для створення платіжних сесій і підтвердження оплати [13,14]. Мух виконує оброблення та відтворення відеоматеріалів, а UploadThing забезпечує завантаження файлів [15, 16].

Узагальнену взаємодію програмних компонентів наведено на рис. 2.10.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				
						56

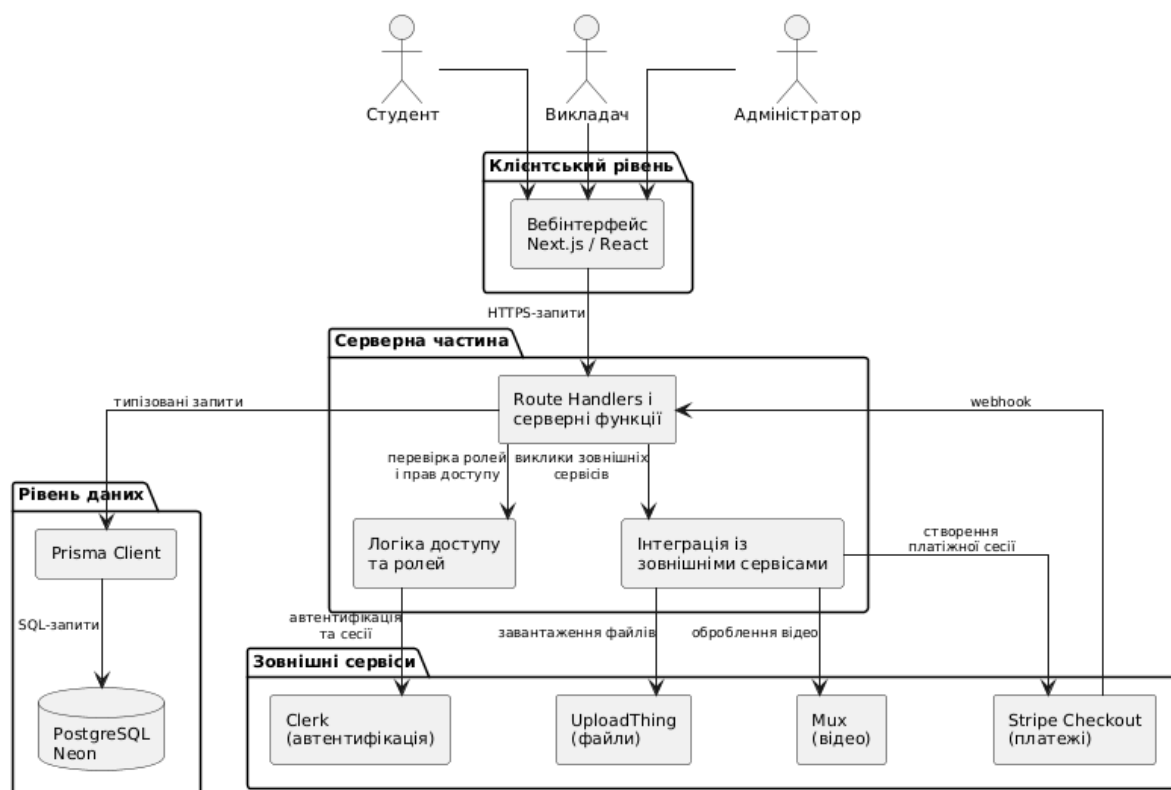


Рис. 2.10. Діаграма компонентів вебплатформи дистанційного навчання

На діаграмі показано взаємодію вебінтерфейсу, серверної частини, Prisma Client, бази даних PostgreSQL та зовнішніх сервісів. Запити користувачів надходять до серверних функцій і обробників маршрутів. Після перевірки даних сервер звертається до бази даних або відповідного зовнішнього сервісу та повертає результат до інтерфейсу.

Автентифікацію користувачів реалізовано за допомогою Clerk. Сервіс забезпечує реєстрацію, вхід, керування обліковими записами та підтримання користувацьких сесій [13]. Локальна модель Profile містить ім'я, електронну адресу, роль і стан верифікації користувача. Після успішного входу відомості Clerk синхронізуються з локальним профілем за полем `userId`.

За відсутності активної сесії сервер повертає код 401, а за недостатніх прав доступу код 403. Ці значення відповідають стандартним кодам стану HTTP [25].

Фрагмент серверної перевірки ролі користувача подано нижче.

```

export const requireApiRole = async (
  allowedRoles: readonly AppRole[]
) => {
  const profile = await getCurrentProfile();

```

```

    if (!profile) {
        return {
            profile: null,
            response: new NextResponse("Не авторизовано", { status: 401 }),
        };
    }

    if (!isAllowedRole(profile.role, allowedRoles)) {
        return {
            profile: null,
            response: new NextResponse("Недостатньо прав доступу", {
                status: 403,
            }),
        };
    }

    return {
        profile,
        response: null,
    };
};

```

Користувачі з ролями TEACHER і ADMIN мають доступ до викладацьких функцій. Адміністративні операції дозволено лише користувачам із роллю ADMIN. Під час редагування курсу, глави або тесту додатково перевіряється авторство відповідного ресурсу. Повний лістинг синхронізації профілю та перевірки ролі користувача наведено в Додатку А (лістинг А.1).

Модуль керування навчальним контентом забезпечує створення, редагування, упорядкування, публікацію та видалення курсів і глав. Новий курс зберігається як чернетка зі значенням `isPublished = false`. Викладач додає назву, опис, категорію, ціну, обкладинку, глави, навчальні ресурси та підсумковий тест.

Матеріали до глав зберігаються в моделі `ChapterResource`. Файли й документи завантажуються через `UploadThing` [16]. Для оброблення та відтворення відео використовується `Mux`, а отримані ідентифікатори відеокурсу й відтворення зберігаються в моделі `MuxData` [15].

Перед публікацією система перевіряє повноту даних курсу, наявність опублікованих глав і коректність структури підсумкового тесту. Якщо встановлені вимоги виконано, значення поля `isPublished` змінюється на `true`, після чого курс стає доступним у каталозі. Повний лістинг перевірки готовності та публікації навчального курсу наведено в Додатку А (лістинг А.2).

Модуль оплати підтримує платний, безоплатний і пільговий способи отримання доступу до курсу. Відомості про наданий доступ зберігаються в моделі Purchase зі значенням PAID, FREE_COURSE або FREE_VERIFIED.

Для платних курсів сервер створює платіжну сесію Stripe Checkout. Підтвердження платежу виконується після отримання серверного повідомлення вебхука. Обробник перевіряє цифровий підпис події та стан платежу відповідно до механізму перевірки вебхуків Stripe [14]. Фрагмент перевірки повідомлення Stripe подано нижче.

```
const body = await req.text();
const signature = (await headers()).get("stripe-signature");

const event = stripe.webhooks.constructEvent(
  body,
  signature!,
  process.env.STRIPE_WEBHOOK_SECRET!
);

if (
  event.type === "checkout.session.completed" ||
  event.type === "checkout.session.async_payment_succeeded"
) {
  const session =
    event.data.object as Stripe.Checkout.Session;

  await processCompletedSession(session);
}
```

Функція processCompletedSession перевіряє стан оплати, отримує відомості про користувача й придбані курси та створює записи Purchase. Унікальне обмеження для полів userId і courseId запобігає повторному наданню доступу.

Загальний прогрес курсу обчислюється на основі кількості завершених опублікованих глав. Окреме поле для загального відсотка не використовується, тому отримане значення відповідає поточним записам прогресу. Повний лістинг збереження прогресу проходження навчальної глави наведено в Додатку А (лістинг А.4).

Модуль тестування підтримує тести до глав і підсумковий тест курсу. Серверна частина перевіряє повноту відповідей і коректність отриманих даних. Результат визначається у відсотках та порівнюється з установленим прохідним балом.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				
						59

Відомості про спробу зберігаються в моделі QuizAttempt, а вибрані відповіді в моделі QuizAttemptAnswer. Завдяки цьому результат тестування не втрачається після повторного відкриття сторінки. Підсумковий тест стає доступним після завершення всіх опублікованих глав курсу. Повний лістинг обчислення та збереження результату тестування наведено в Додатку А (лістинг А.5).

Після успішного складання підсумкового тесту система перевіряє завершення всіх опублікованих глав і створює сертифікат з унікальним кодом перевірки. Складене унікальне обмеження для полів userId і courseId запобігає повторному створенню сертифіката для одного користувача та курсу.

Модуль верифікації забезпечує подання та опрацювання заявок на підтвердження статусу користувача. Заявка зберігається в моделі VerificationRequest, а підтвердні документи в моделі VerificationDocument. Створення заявки та зміна стану профілю виконуються в межах транзакції Prisma. Під час прийняття рішення адміністратором узгоджено оновлюються заявка і профіль користувача. Повний лістинг транзакційного створення заявки на верифікацію наведено в Додатку А (лістинг А.6).

Технічну підтримку реалізовано за допомогою моделей SupportTicket і SupportMessage. Користувач створює звернення та обмінюється повідомленнями з адміністратором. Стан звернення змінюється відповідно до дій учасників і може набувати значень OPEN, ANSWERED або CLOSED.

Аналітичний модуль формує показники на основі даних про курси, доступи, прогрес і відгуки. Для викладача обчислюються кількість студентів, середній прогрес, рейтинг і розрахунковий дохід. Адміністративна панель відображає узагальнені відомості про користувачів, курси, заявки на верифікацію та звернення до підтримки.

Клієнтську частину реалізовано з використанням React і Tailwind CSS [7, 9]. Компоненти shadcn/ui застосовуються для побудови елементів інтерфейсу [26], Lucide React для відображення піктограм [27], Zustand для збереження стану кошика [28], TanStack Table для табличного подання

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				60

даних [29], Axios для виконання HTTP-запитів [30], а React Hot Toast для повідомлень про результати операцій [31].

Інтерфейс підтримує адаптивне відображення, клавіатурне керування, стани завантаження та повідомлення про помилки. Під час реалізації елементів навігації та керування враховано рекомендації WCAG 2.2 щодо доступності вебінтерфейсів [23].

Функціональні модулі працюють у межах одного застосунку та використовують спільну базу даних. Серверна частина перевіряє права доступу, обробляє запити та взаємодіє з PostgreSQL через Prisma Client. Зовнішні сервіси забезпечують автентифікацію, оплату курсів, роботу з відеоматеріалами та завантаження файлів, а узгодженість даних підтримується транзакціями й унікальними обмеженнями.

Висновки до другого розділу

У другому розділі виконано проектування вебплатформи дистанційного навчання. Визначено функціональні й нефункціональні вимоги, ролі користувачів та доступні їм операції. На їх основі розроблено діаграму варіантів використання й узагальнену діаграму класів системи.

Спроектовано реляційну базу даних PostgreSQL та визначено основні сутності, їхні атрибути, зв'язки й обмеження цілісності. Структуру даних описано у схемі Prisma. Для запобігання дублюванню записів і забезпечення узгодженості даних передбачено унікальні обмеження та транзакції.

За допомогою UML-діаграм описано алгоритми створення і публікації курсів, верифікації користувачів, проходження навчання, збереження прогресу, тестування, формування сертифікатів та оплати курсів.

Систему реалізовано як модульний повностековий вебзастосунок на основі Next.js із використанням Prisma Client для роботи з базою даних. Автентифікацію, оплату, роботу з відео та файлами забезпечують Clerk, Stripe, Mux і UploadThing.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				
						61

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБПЛАТФОРМОЮ ДИСТАНЦІЙНОГО НАВЧАННЯ

3.1. Порядок встановлення та налаштування параметрів системи

Встановлення вебплатформи передбачає підготовку програмного середовища, отримання вихідного коду, встановлення залежностей, створення структури бази даних та підключення зовнішніх сервісів. Систему реалізовано як повностековий вебзастосунок на основі Next.js із маршрутизатором App Router, тому клієнтські компоненти, серверна логіка та обробники маршрутів розміщені в одному програмному проєкті [6].

Для локальної роботи необхідні Node.js, npm, Git, сучасний веббраузер і підключення до мережі Інтернет. Node.js забезпечує виконання серверної частини застосунку, а npm використовується для встановлення залежностей і запуску команд проєкту [17, 18]. Git застосовується для отримання вихідного коду та роботи зі змінами [19].

Вихідний код зберігається у віддаленому репозиторії GitLab. Після створення локальної копії встановлюються залежності, визначені у файлі package.json. До них належать Next.js, React, Prisma ORM, засоби побудови інтерфейсу та пакети для взаємодії із зовнішніми сервісами. Файл package-lock.json фіксує версії пакетів і дає змогу відтворити однакове програмне середовище.

Основні параметри системи зберігаються у файлі .env. Він містить адресу вебзастосунку, рядок підключення до бази даних і ключі зовнішніх сервісів. Під час локального запуску використовується адреса локального сервера, а після розгортання зазначається постійна адреса вебплатформи.

Файл .env виключено із системи контролю версій, оскільки він містить конфіденційні відомості. Параметри клієнтської частини відокремлено від службових значень, які використовуються лише серверними обробниками. Завдяки цьому рядок підключення до бази даних і секретні ключі не передаються до браузера користувача.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				62

Для зберігання інформації використовується PostgreSQL у хмарному сервісі Neon [11, 12]. Це усуває потребу встановлювати окремий сервер бази даних на локальному комп'ютері та забезпечує віддалений доступ до сховища. Рядок підключення отримується в панелі керування Neon і додається до змінних середовища.

Структуру таблиць, зв'язків, індексів та обмежень цілісності описано у файлі `prisma/schema.prisma`. Взаємодія серверної частини з базою даних виконується через Prisma Client [10]. Після налаштування підключення генерується Prisma Client, а до бази даних застосовуються підготовлені міграції. У результаті створюються таблиці профілів, курсів, глав, навчальних матеріалів, доступів, прогресу, тестування, сертифікатів, верифікації та технічної підтримки.

Prisma ORM забезпечує виконання операцій із базою даних за допомогою типізованих запитів і підтримує узгодженість між моделями та програмним кодом. Для перегляду таблиць і записів використовується Prisma Studio.

Автентифікацію користувачів забезпечує Clerk [13]. Для підключення сервісу створюється застосунок у панелі Clerk, після чого публічний і секретний ключі додаються до файла змінних середовища. Також задаються адреси сторінок входу, реєстрації та маршрути перенаправлення після автентифікації.

Під час першого входу відомості користувача синхронізуються з локальною таблицею профілів. Новому профілю призначається роль STUDENT. У системі також передбачено ролі TEACHER і ADMIN. Роль першого адміністратора встановлюється через Prisma Studio, після чого керування ролями виконується через адміністративний інтерфейс.

Clerk відповідає за облікові записи та користувацькі сесії, тому паролі не зберігаються у власній базі даних вебплатформи. Локальний профіль містить роль, стан верифікації та інші відомості, потрібні для роботи системи.

Для оплати курсів використовується Stripe Checkout [14]. У змінних середовища зберігаються ключі для створення платіжних сесій і перевірки повідомлень вебхука. Після вибору платного курсу серверна частина формує

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				63

платіжну сесію, а користувач переходить на захищену сторінку Stripe. Оплата підтверджується після отримання повідомлення від платіжного сервісу.

Під час локальної роботи Stripe використовується в тестовому режимі, а його події перенаправляються до локального обробника вебхука. Після розгортання у Stripe вказується адреса обробника у виробничому середовищі. Це дає змогу перевірити оплату та надання доступу до курсу без реальних фінансових операцій.

Оброблення відеоматеріалів виконується сервісом Mux [15]. Він забезпечує підготовку та потокове відтворення відео. У базі даних зберігаються службові ідентифікатори, необхідні для відображення матеріалу у відповідній главі. Використання Mux усуває потребу у створенні власного механізму перекодування великих відеофайлів.

Завантаження обкладинок курсів, навчальних матеріалів і документів верифікації забезпечує UploadThing [16]. Файли зберігаються у зовнішньому сховищі, а в PostgreSQL записуються їхні адреси, назви та службові ключі. Така організація відповідає умовам хмарного розгортання, оскільки файлова система Vercel не використовується для постійного зберігання матеріалів.

Після налаштування бази даних і зовнішніх сервісів вебплатформа запускається локально. Під час першого запуску перевіряються сторінки входу та реєстрації, створення профілю, отримання даних із PostgreSQL і доступність основних маршрутів. Окремо перевіряються завантаження файлів, відтворення відео та створення тестової платіжної сесії.

Перед розгортанням виконується перевірка типів, статичний аналіз програмного коду та створення виробничої збірки. TypeScript дає змогу виявити невідповідності між компонентами, серверними обробниками й моделями даних. ESLint використовується для контролю якості програмного коду, а успішне створення збірки підтверджує готовність проєкту до хмарного розміщення.

Для розгортання вебплатформи використовується Vercel [21]. Платформа підтримує Next.js, автоматично встановлює залежності та створює виробничу

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				64

збірку. Репозиторій GitLab підключається до проєкту Vercel, а конфіденційні параметри додаються до налаштувань середовища.

У Vercel задаються змінні середовища для підключення до Neon, Clerk, Stripe, Mux і UploadThing. Після отримання постійної адреси вебплатформи вона додається до налаштувань перенаправлення Clerk і параметрів вебхука Stripe. Після зміни параметрів виконується повторне розгортання застосунку.

Структуру розгортання та взаємодію компонентів наведено на рис. 3.1.

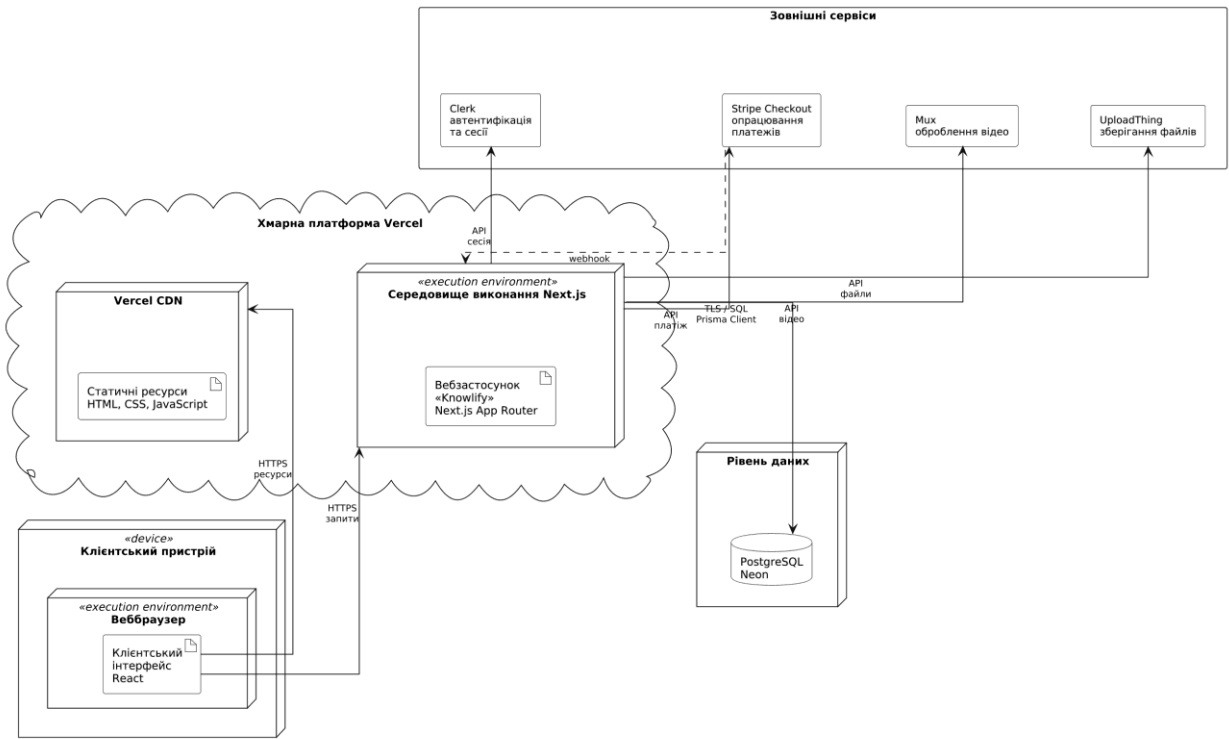


Рис. 3.1. Діаграма розгортання вебплатформи дистанційного навчання

На діаграмі відображено клієнтський пристрій, хмарне середовище Vercel, базу даних і зовнішні сервіси. Користувач працює із системою через веббраузер. Статичні ресурси передаються через Vercel CDN, а запити опрацьовуються вебзастосунком Next.js у серверному середовищі Vercel.

Серверна частина взаємодіє з PostgreSQL у сервісі Neon через Prisma Client. Clerk забезпечує автентифікацію користувачів, Stripe Checkout опрацьовує платежі, Mux використовується для роботи з відеоматеріалами, а UploadThing зберігає завантажені файли. Обмін даними між компонентами виконується через захищені мережеві з’єднання.

Після розгортання перевіряються реєстрація та вхід користувачів, отримання даних із бази, рольове розмежування доступу, завантаження матеріалів, відтворення відео та створення платіжної сесії. Така перевірка дає змогу виявити відсутні змінні середовища або неправильно вказані адреси зовнішніх сервісів.

Встановлення та налаштування вебплатформи охоплюють підготовку програмного середовища, встановлення залежностей, створення файлу змінних середовища, підготовку бази даних і підключення зовнішніх сервісів. Після завершення налаштування система може працювати локально або бути розгорнута у Vercel.

3.2. Структура інтерфейсу вебплатформи дистанційного навчання

Користувацький інтерфейс вебплатформи побудовано з урахуванням рольового розмежування доступу. Після автентифікації склад навігаційного меню та доступні функції визначаються роллю користувача. У системі передбачено інтерфейси студента, викладача й адміністратора.

Основними складовими інтерфейсу є бічна навігаційна панель, верхня панель та основна робоча область. Бічне меню забезпечує перехід між функціональними розділами, а робоча область використовується для відображення каталогу, навчальних матеріалів, форм, аналітичних даних і засобів керування. Адаптивна структура дає змогу працювати з платформою на комп'ютерах, планшетах і смартфонах

Для студента робота з платформою починається зі сторінки реєстрації. Створення облікового запису виконується через сервіс Clerk за допомогою електронної адреси й пароля або підтримуваного зовнішнього провайдера. Для зареєстрованих користувачів на сторінці передбачено перехід до форми входу. Сторінку автентифікації користувача наведено на рис. 3.2.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				
						66

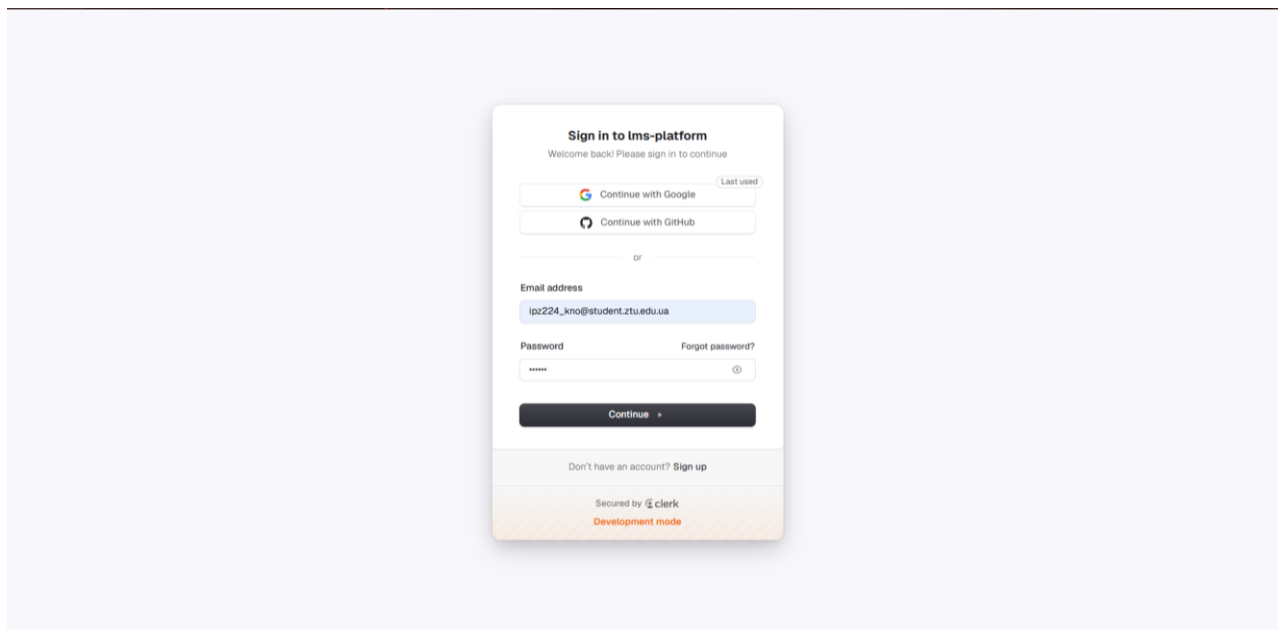


Рис. 3.2. Сторінка автентифікації користувача

Основним засобом пошуку навчального контенту є каталог курсів. Студент може виконувати пошук за назвою, вибирати категорію та застосовувати фільтри за типом доступу, станом проходження й наявністю додаткових матеріалів. Курси відображаються у вигляді карток з обкладинкою, назвою, вартістю та відомостями про навчальний прогрес. Головну сторінку каталогу та засоби фільтрації наведено на рис. 3.3.

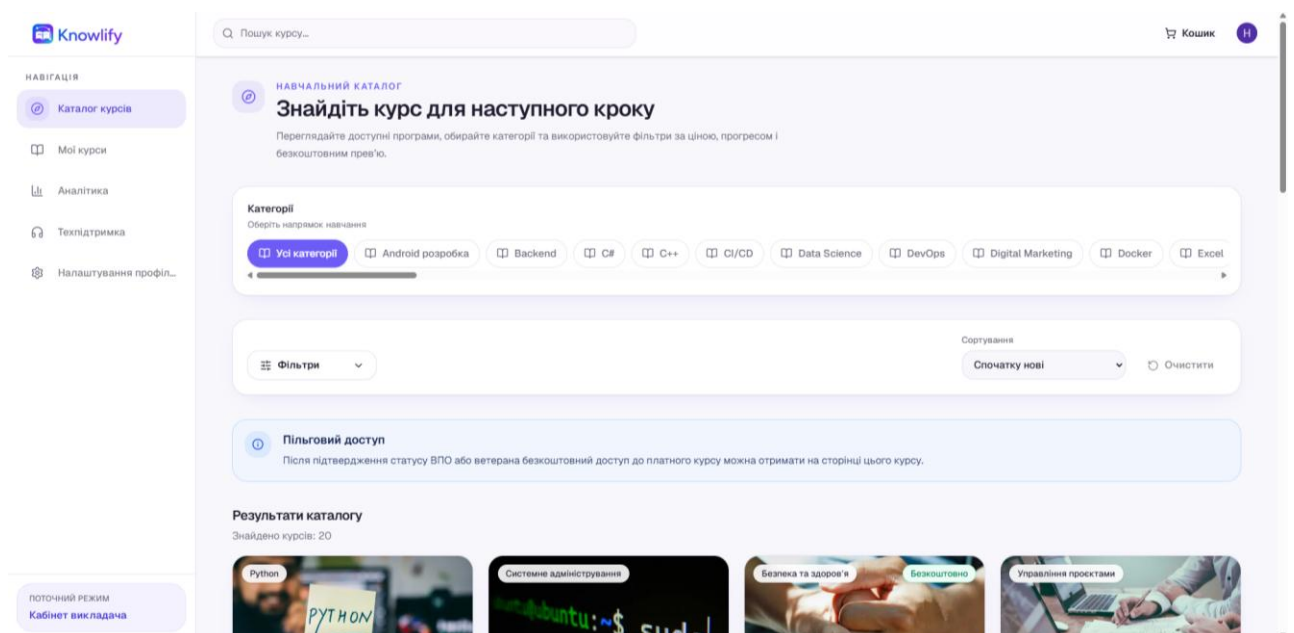


Рис. 3.3. Головна сторінка каталогу і фільтрація навчальних курсів

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				67

Після вибору курсу відкривається сторінка з його назвою, описом, категорією, структурою та кількістю навчальних матеріалів. Для безоплатного курсу студент може одразу оформити доступ. Для платного курсу передбачено придбання, а окрема глава може бути доступною для попереднього перегляду. Сторінку перегляду навчального курсу наведено на рис. 3.4.

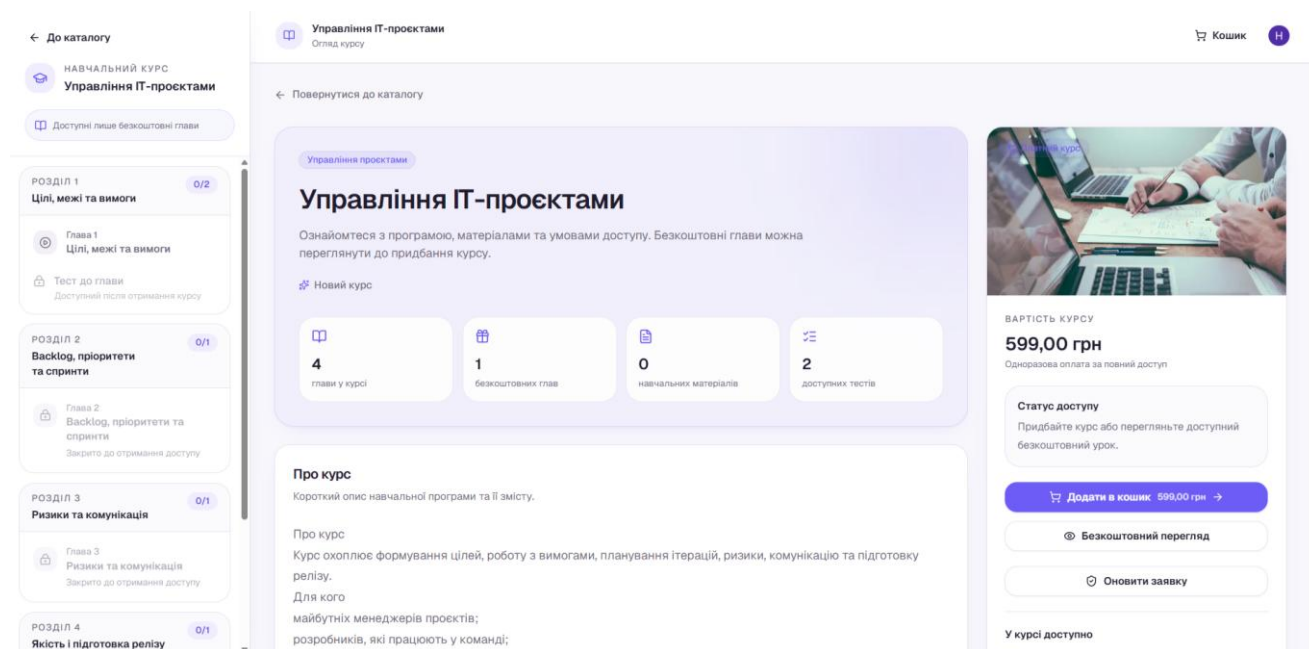


Рис. 3.4. Сторінка перегляду навчального курсу

На сторінці курсу студент бачить доступні способи його проходження. Залежно від вартості курсу та стану верифікації користувач може отримати безоплатний, платний або пільговий доступ. Після оформлення доступу курс з'являється серед доступних для навчання.

Якщо студент обирає платний курс, він може додати його до кошика для подальшого оформлення доступу. У кошику відображаються вибрані платні курси, їхня вартість і загальна сума замовлення. Користувач може видалити окрему позицію, очистити кошик, повернутися до каталогу або перейти до оплати. На сторінці також наведено інформацію про пільговий доступ для користувачів із підтвердженим статусом. Інтерфейс кошика та оформлення доступу до платного курсу наведено на рис. 3.5.

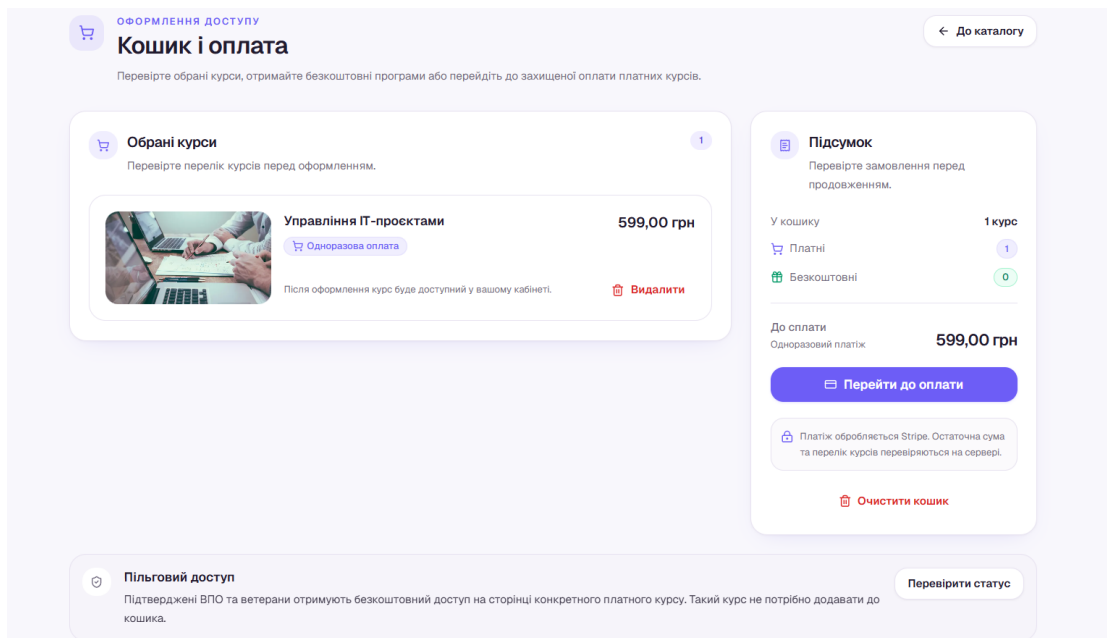


Рис. 3.5. Інтерфейс кошика та оформлення доступу до платного курсу

Після переходу до оплати система створює захищену платіжну сесію Stripe Checkout. На сторінці відображаються назва курсу, сума платежу, контактна інформація та форма введення платіжних реквізитів. Для перевірки роботи платіжного механізму використано тестовий режим Stripe із тестовими платіжними даними. Сторінку оплати через Stripe Checkout наведено на рис. 3.6.

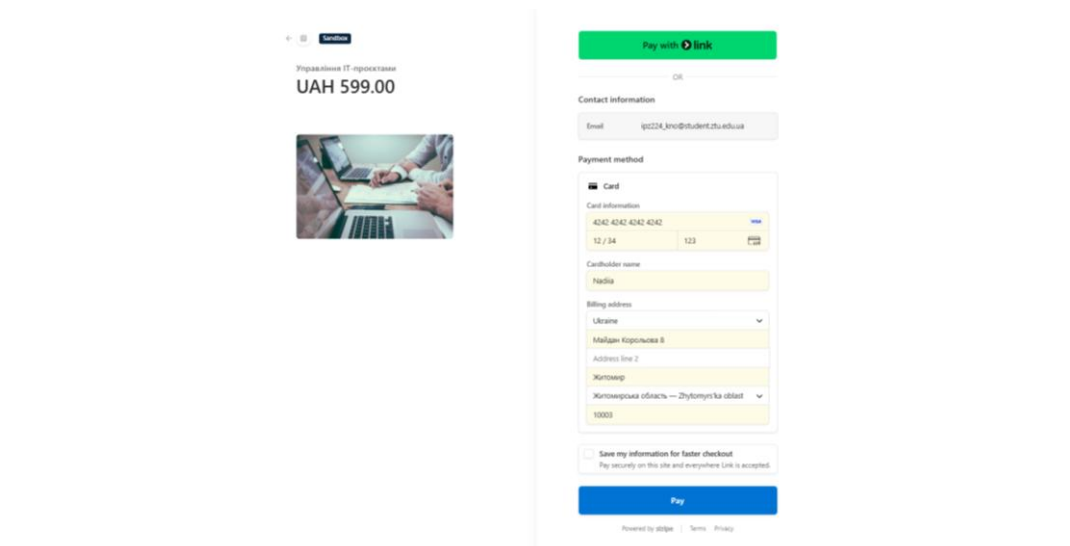


Рис. 3.6. Сторінка оплати курсу через Stripe Checkout

Після успішного завершення платежу користувач повертається до вебплатформи. На сторінці кошика відображається повідомлення про підтвердження оплати, а придбаний курс стає доступним для проходження.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				69

Оплачена позиція автоматично видаляється з кошика. Підтвердження успішної оплати та надання доступу наведено на рис. 3.7.

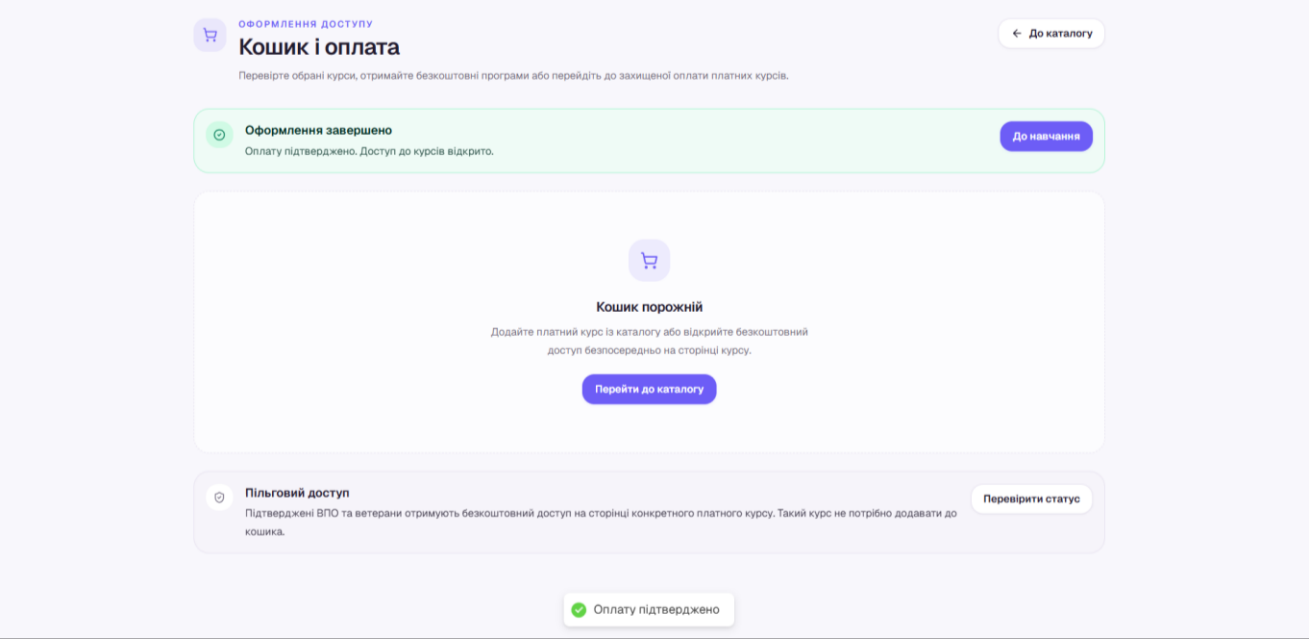


Рис. 3.7. Підтвердження успішної оплати та надання доступу до курсу

Після проходження курсу студент може залишити оцінку та текстовий відгук. На сторінці відображаються середній рейтинг, розподіл оцінок і відгуки користувачів. Інтерфейс перегляду рейтингу та відгуків наведено на рис. 3.8.

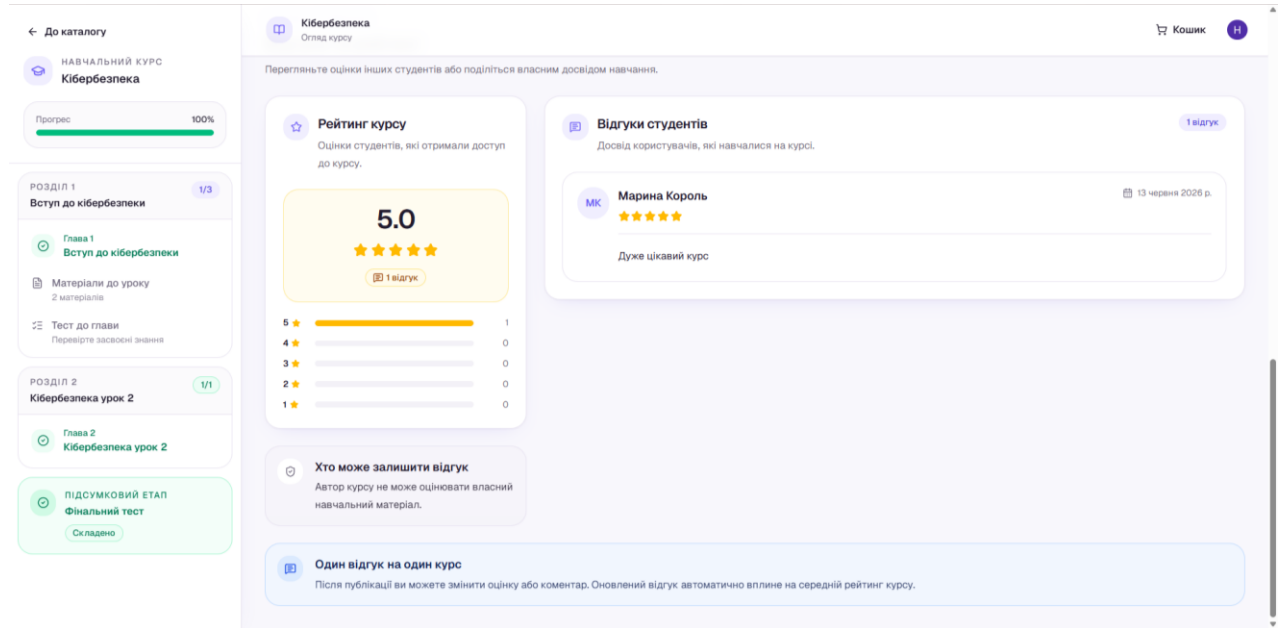


Рис. 3.8. Інтерфейс перегляду рейтингу та відгуків про навчальний курс

Основний навчальний процес відбувається на сторінці проходження курсу. У бічній панелі наведено перелік розділів і глав, а в робочій області розміщено

відеопрогравач. Також відображається загальний прогрес проходження. Система зберігає позицію перегляду відео та стан завершення глав. Інтерфейс навчання наведено на рис. 3.9.

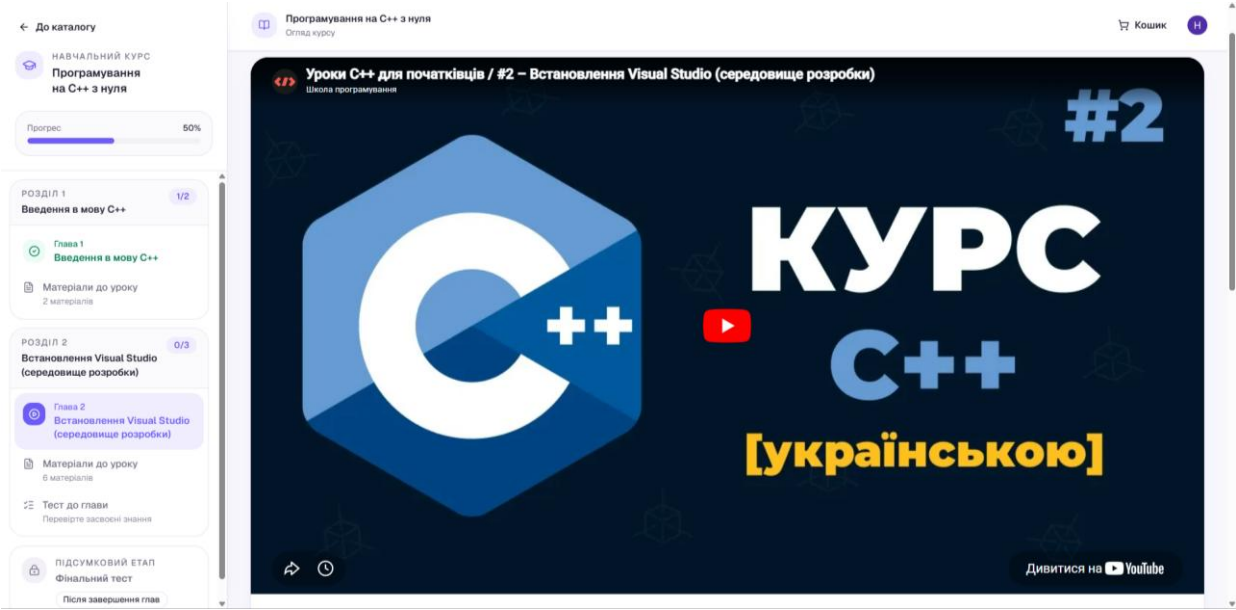


Рис. 3.9. Інтерфейс проходження навчального курсу

Після опрацювання матеріалів студент може пройти тест до глави. Система обчислює відсотковий результат, кількість правильних відповідей і порівнює отримане значення з прохідним балом. Результат зберігається та залишається доступним після повторного відкриття сторінки. Результат проходження тесту до глави наведено на рис. 3.10.

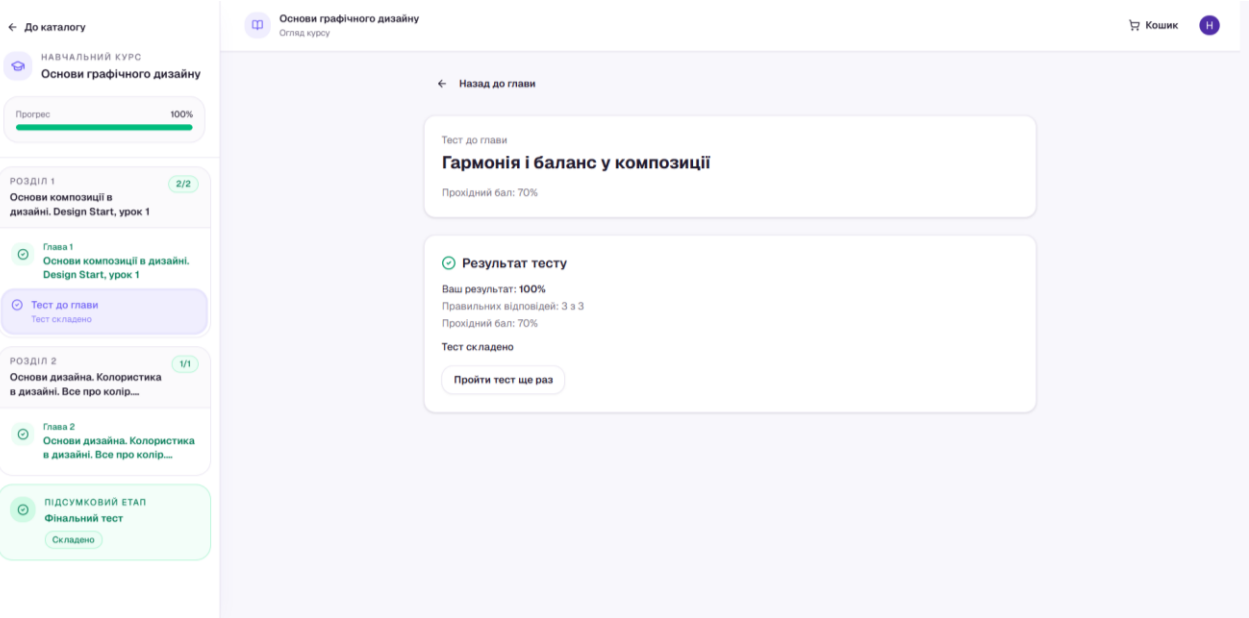


Рис. 3.10. Результат проходження тесту до глави

Після завершення всіх опублікованих глав студент отримує доступ до підсумкового тесту. На сторінці відображаються кількість пройдених глав, кількість запитань і встановлений прохідний бал. Після надсилання відповідей система обчислює результат, порівнює його з прохідним значенням і зберігає спробу проходження. Якщо тест складено успішно, студент бачить отриманий відсоток, кількість правильних відповідей і повідомлення про успішне завершення підсумкового контролю. Після цього стає доступною функція формування сертифіката. Результат проходження підсумкового тесту наведено на рис. 3.11.

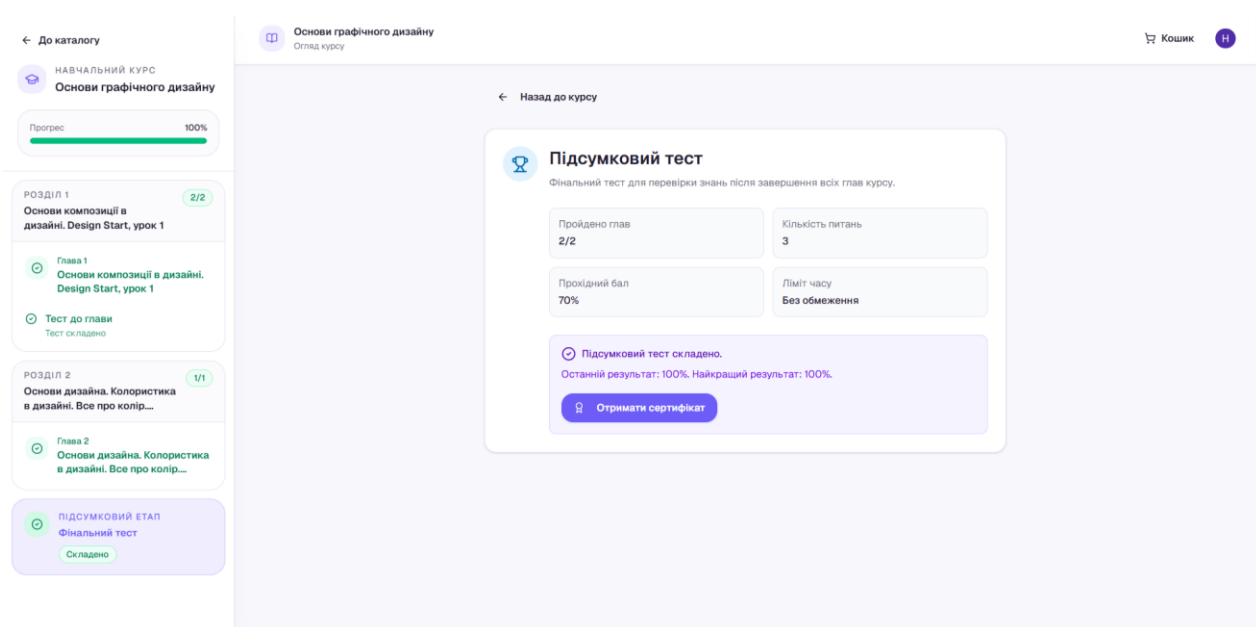


Рис. 3.11. Результат проходження підсумкового тесту

Наведена сторінка підтверджує, що студент виконав усі умови завершення курсу. Збереження результату в базі даних дає змогу повторно переглянути його після оновлення сторінки без необхідності проходити тест ще раз.

Сертифікат формується після завершення навчальних матеріалів і успішного проходження підсумкового контролю. Він містить ім'я студента, назву навчального курсу, дату видачі, ідентифікатор документа та унікальний код перевірки. Окремо відображається інформація про підтвердження сертифіката системою та короткий опис пройденого курсу. Сертифікат про завершення навчального курсу наведено на рис. 3.12.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				
						72

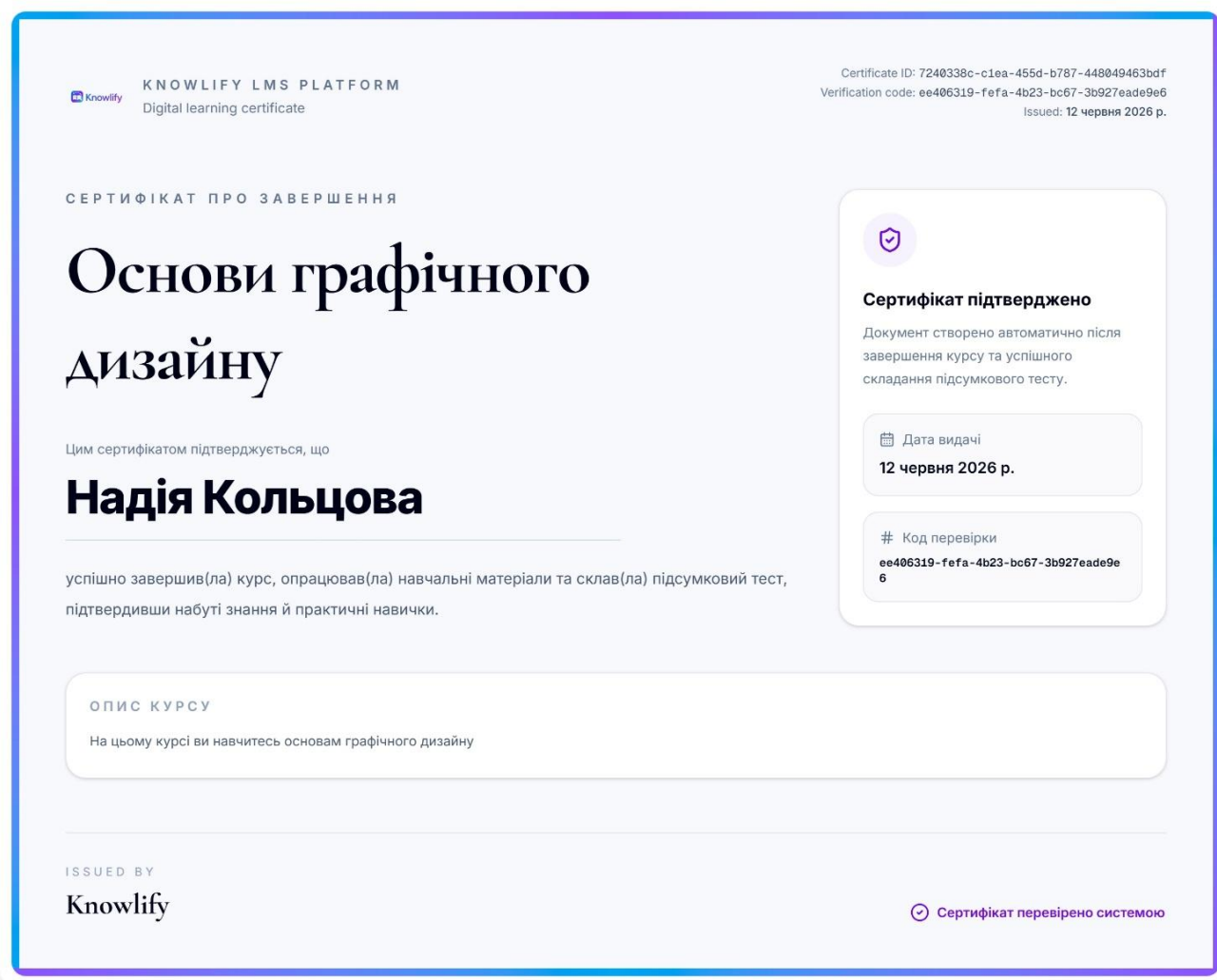


Рис. 3.12. Сертифікат про завершення навчального курсу

Унікальний код дає змогу перевірити дійсність сертифіката. Якщо документ уже створено, система повертає його без повторного формування.

Для отримання пільгового доступу до платного курсу користувач може подати заявку на підтвердження статусу ВПО або ветерана. На сторінці відображаються тип і стан заявки, завантажені документи та рішення адміністратора.

У разі схвалення підтверджений статус зберігається в профілі користувача. Якщо заявку відхилено, користувач може переглянути коментар адміністратора та подати її повторно. Сторінку підтвердження статусу користувача наведено на рис. 3.13.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				73

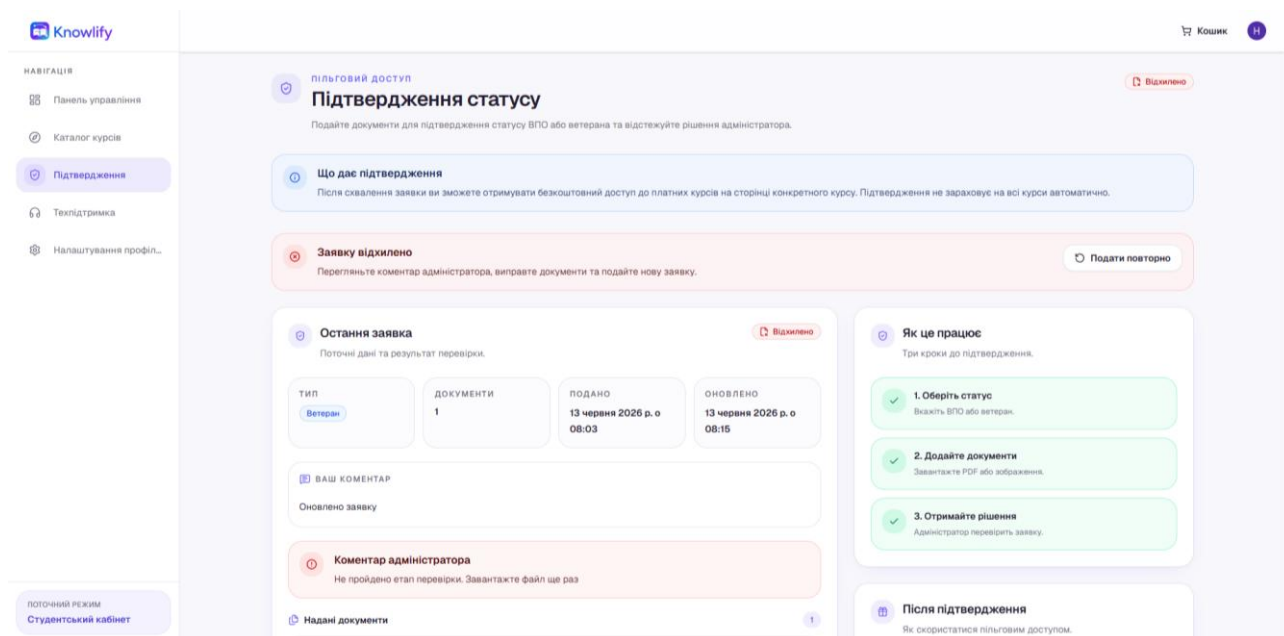


Рис. 3.13. Сторінка підтвердження статусу ВПО або ветерана

Під час створення заявки користувач вибирає тип статусу, завантажує підтвердні документи та за потреби додає коментар. Форму подання заявки наведено на рис. 3.14.

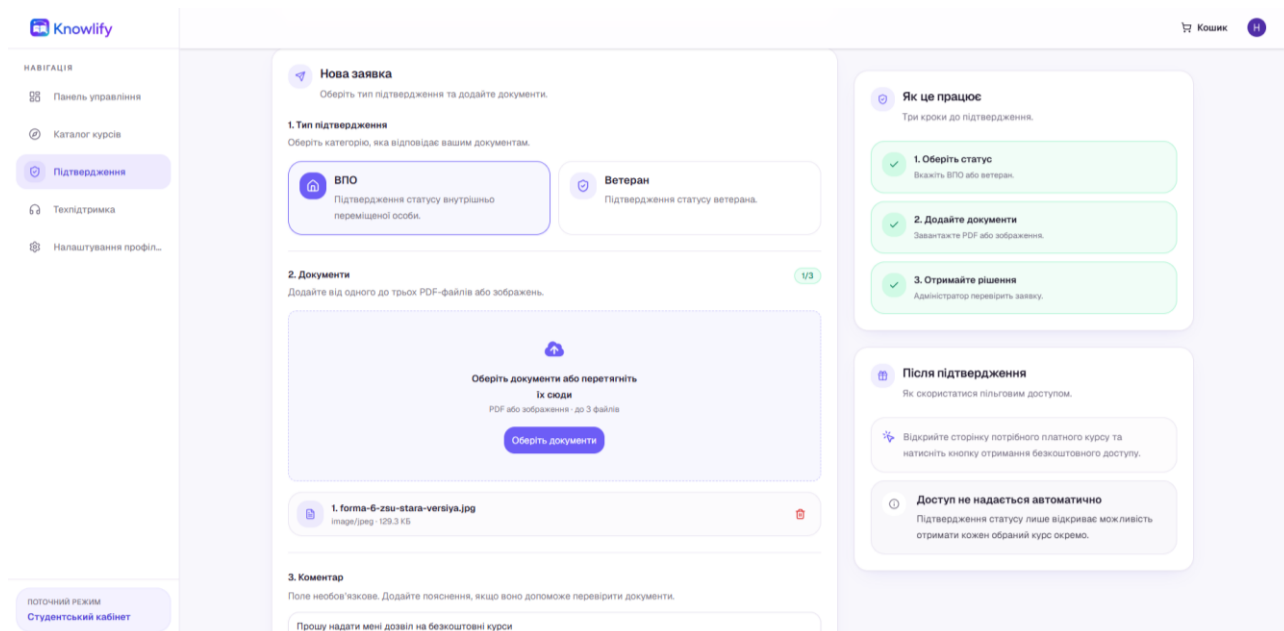


Рис. 3.14. Форма подання заявки на підтвердження статусу для ВПО та ветеранів

Для вирішення технічних та організаційних питань передбачено модуль підтримки. Студент може створити нове звернення, вибрати категорію проблеми та переглядати стан його опрацювання. На сторінці також наведено кількість

		Кольцова Н.О.			ПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				74

відкритих, опрацьовуваних і закритих звернень. Сторінку технічної підтримки наведено на рис. 3.15.

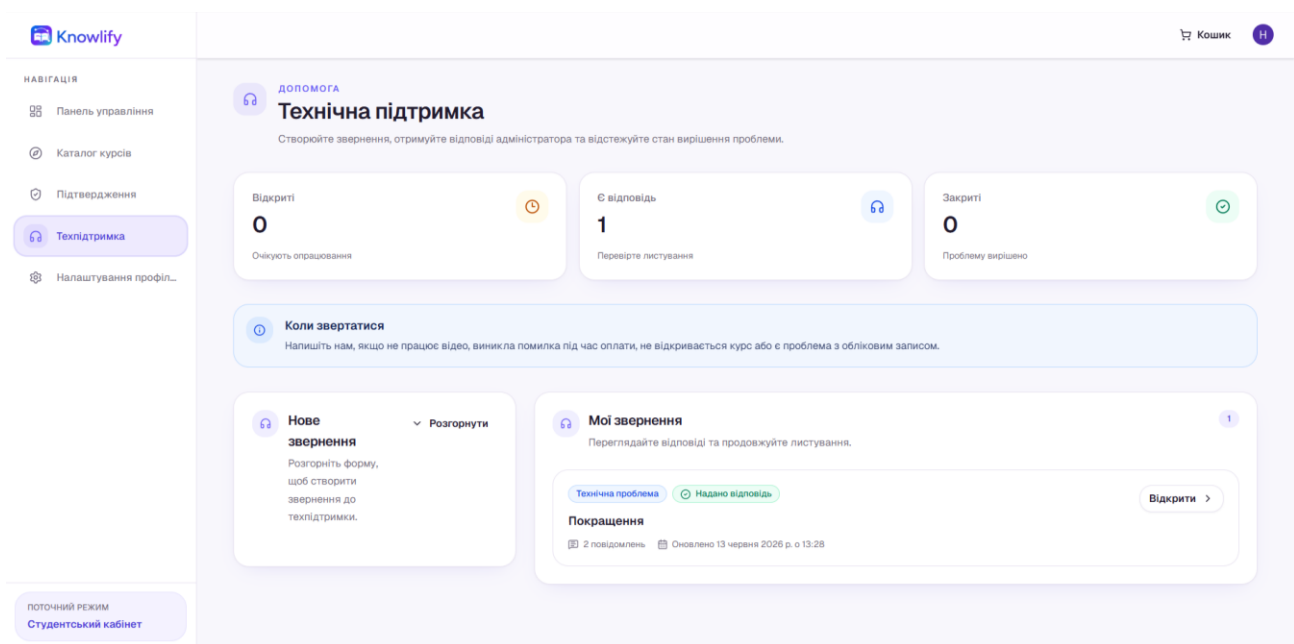


Рис. 3.15. Сторінка технічної підтримки користувача

Після відкриття звернення відображаються його категорія, стан, дата створення та історія повідомлень. Користувач може надсилати нові повідомлення та закрити звернення після вирішення проблеми. Інтерфейс перегляду звернення наведено на рис. 3.16.

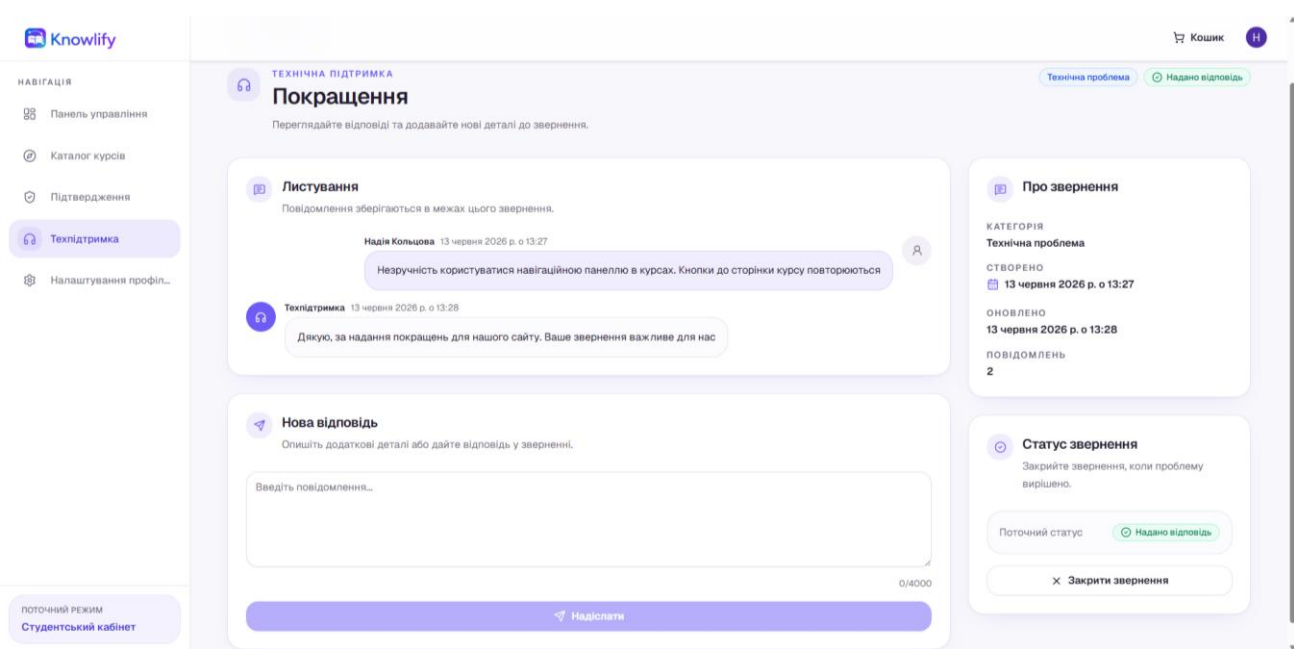


Рис. 3.16. Інтерфейс перегляду звернення до технічної підтримки

У налаштуваннях профілю користувач може переглядати роль, стан верифікації, кількість отриманих доступів і сертифікатів, а також змінювати ім'я, прізвище та фотографію. Облікові дані синхронізуються із сервісом Clerk. Сторінку налаштування профілю наведено на рис. 3.17.

Рис. 3.17. Сторінка налаштування профілю користувача

Для користувача з роллю викладача інтерфейс платформи розширюється засобами створення, редагування та публікації навчальних курсів, керування їхнім вмістом і перегляду аналітичних показників. Після призначення відповідної ролі в навігаційному меню з'являються розділи «Мої курси» та «Аналітика».

На сторінці «Мої курси» відображаються загальна кількість створених курсів, кількість опублікованих записів, чернеток і наданих доступів. Таблиця містить назви курсів, категорії, вартість, стан публікації та дату останнього оновлення. Також передбачено пошук і фільтрацію. Сторінку керування навчальними курсами викладача наведено на рис. 3.18.

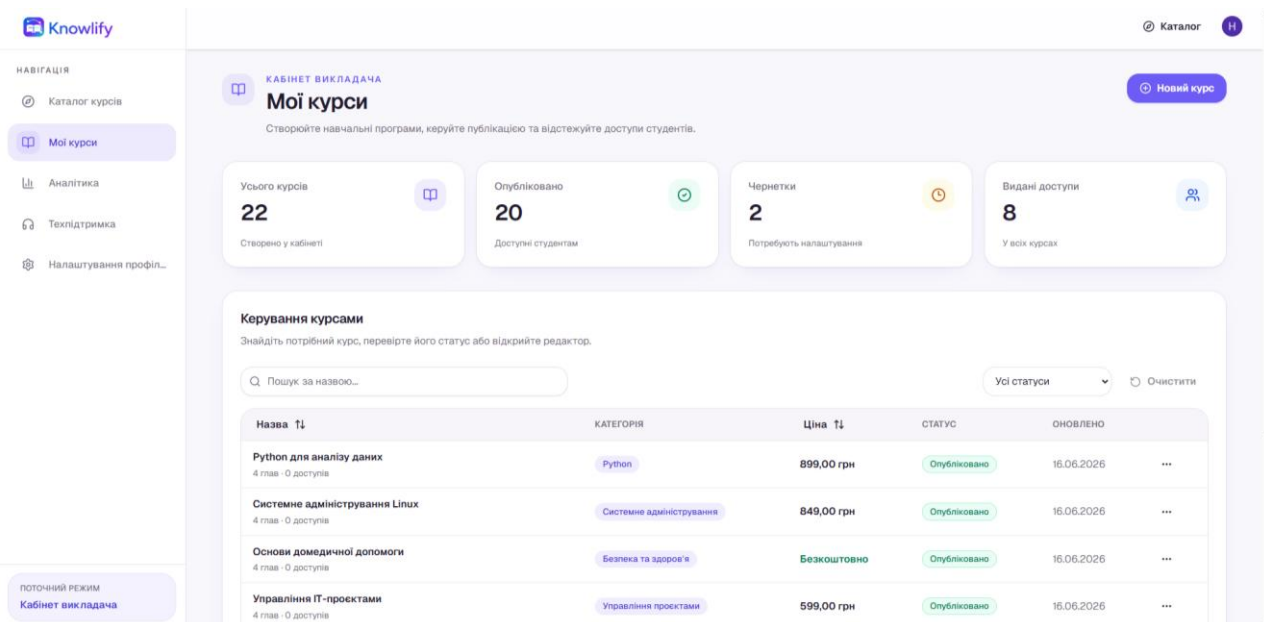


Рис. 3.18. Сторінка керування навчальними курсами викладача

У редакторі курсу викладач налаштовує назву, опис, обкладинку, категорію та вартість. Блок готовності показує виконання обов’язкових умов, необхідних для публікації, що зменшує ймовірність появи в каталозі неповністю заповнених курсів. Після заповнення всіх відомостей курс можна опублікувати, переглянути або зняти з публікації. Інтерфейс налаштування та публікації навчального курсу наведено на рис. 3.19.

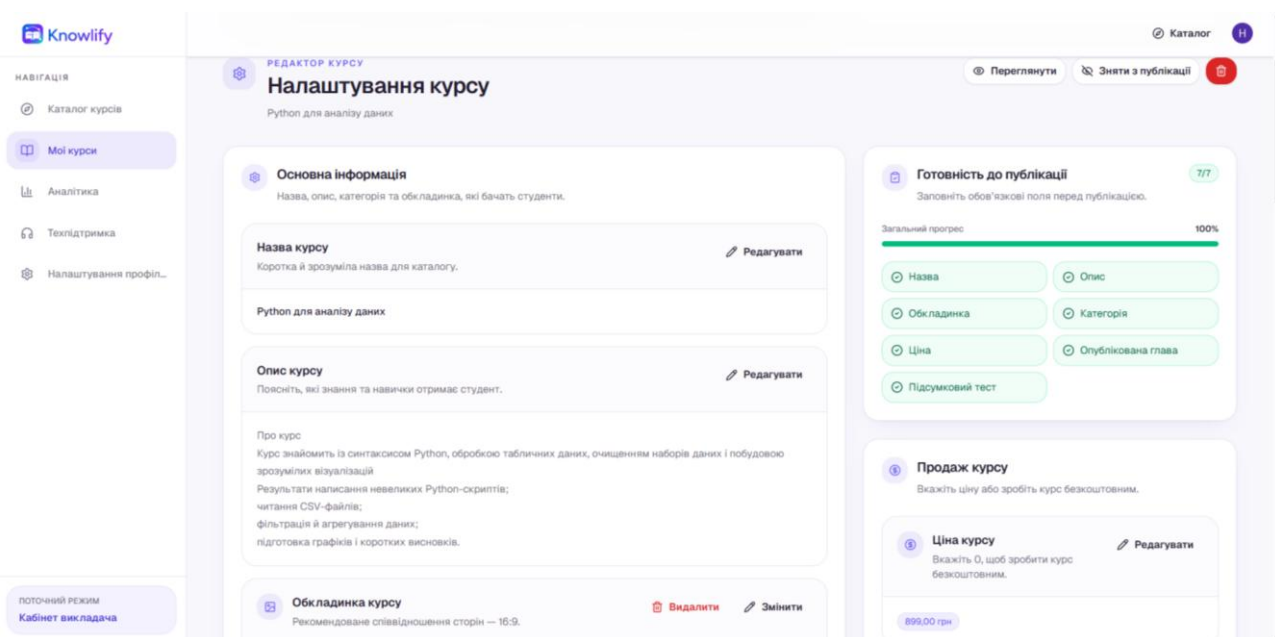


Рис. 3.19. Інтерфейс налаштування та публікації навчального курсу

У редакторі глави викладач зазначає назву й опис, додає відео, навчальні ресурси та тестові завдання. На сторінці відображаються позиція глави, тип відеоматеріалу, кількість доданих ресурсів і стан готовності до публікації. Навчальні матеріали можна додавати у вигляді файлів або посилань на зовнішні джерела. Такий підхід дає змогу поступово наповнювати курс матеріалами й публікувати лише готові до використання глави. Інтерфейс налаштування глави навчального курсу наведено на рис. 3.20.

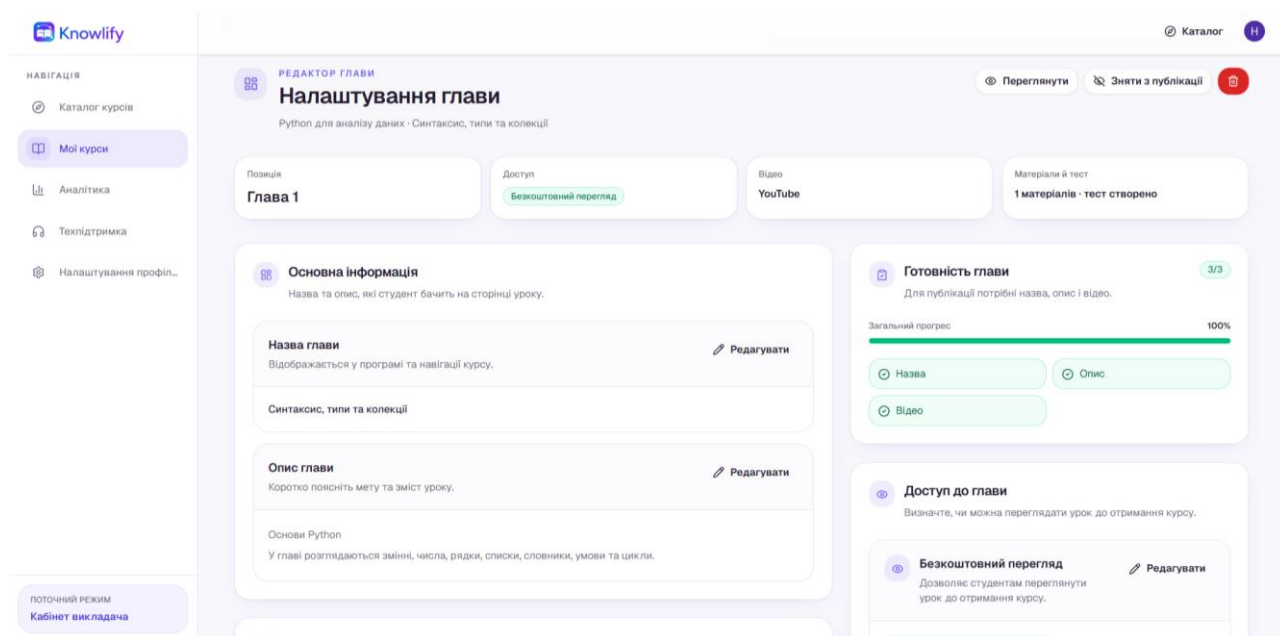


Рис. 3.20. Інтерфейс налаштування глави навчального курсу

До глави можна додати або замінити відеоматеріал і перевірити його відтворення безпосередньо в редакторі. Окремо налаштовується можливість безоплатного попереднього перегляду глави користувачами, які ще не отримали повного доступу до курсу. Після налаштування відео та параметрів доступу викладач може підготувати главу до публікації в межах курсу. Інтерфейс додавання відеоматеріалу та налаштування доступу до глави наведено на рис. 3.21.

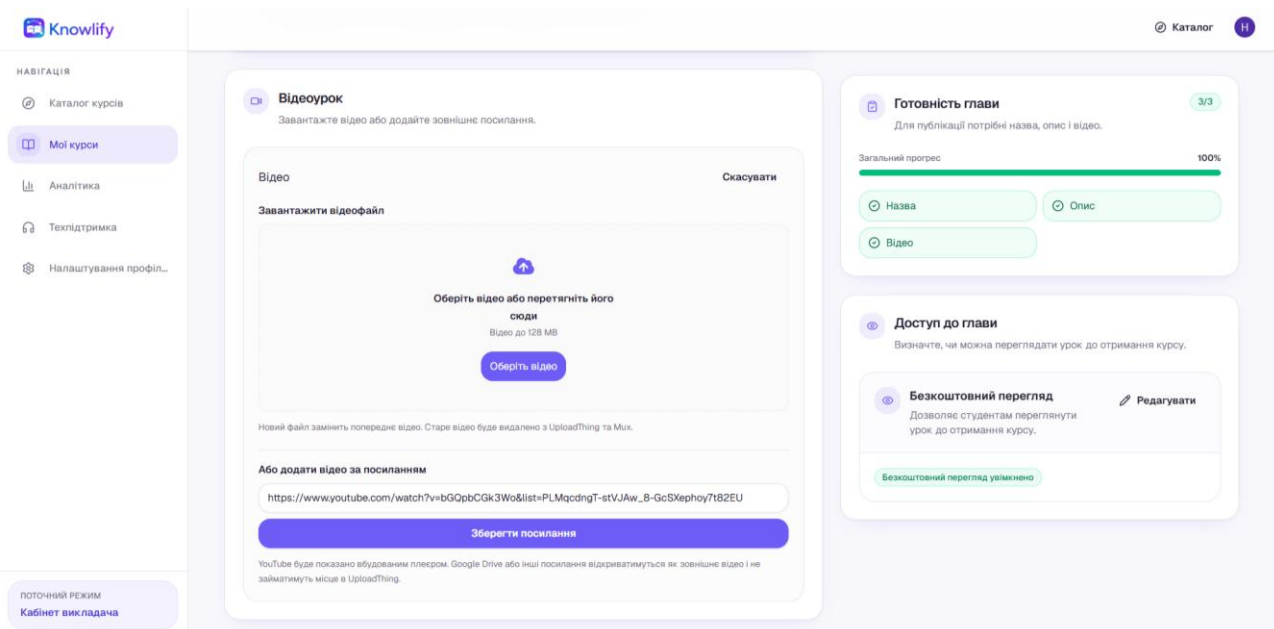


Рис. 3.21. Налаштування відеоматеріалу та доступу до глави

Для підсумкової перевірки знань викладач створює тест курсу, зазначає його назву й опис, установлює прохідний бал, обмеження часу та допустиму кількість спроб. До тесту додаються запитання з варіантами відповідей, серед яких визначається одна правильна відповідь. Інтерфейс налаштування підсумкового тесту навчального курсу наведено на рис. 3.22.

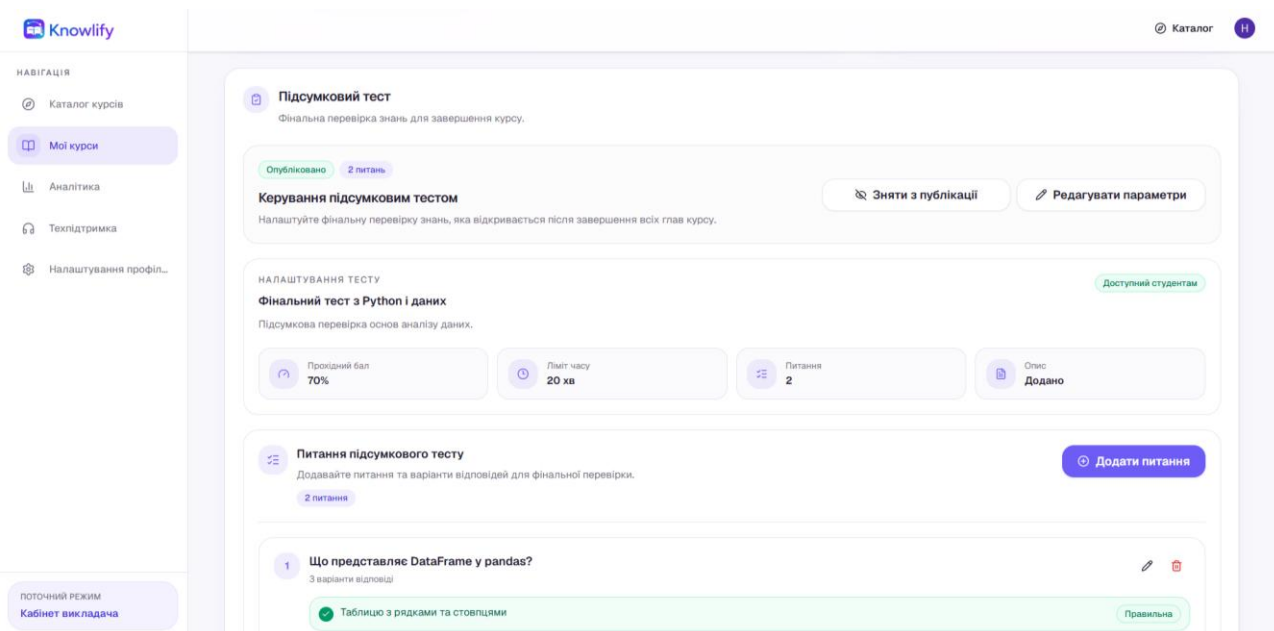


Рис. 3.22. Інтерфейс налаштування підсумкового тесту навчального курсу

У розділі аналітики відображаються розрахований дохід, кількість студентів, середній прогрес проходження, середній рейтинг і кількість

		Кольцова Н.О.			ПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				79

опублікованих курсів. Діаграма показує динаміку доходів та наданих доступів за вибраний період. Додатково наведено співвідношення платних і безоплатних доступів, перелік останніх студентів та таблицю ефективності курсів. Для кожного курсу відображаються кількість доступів, середній прогрес, рейтинг, кількість завершень і отриманий дохід. Ці показники дають змогу порівнювати курси та визначати матеріали, які потребують оновлення. Основні аналітичні показники курсів викладача наведено на рис. 3.23.

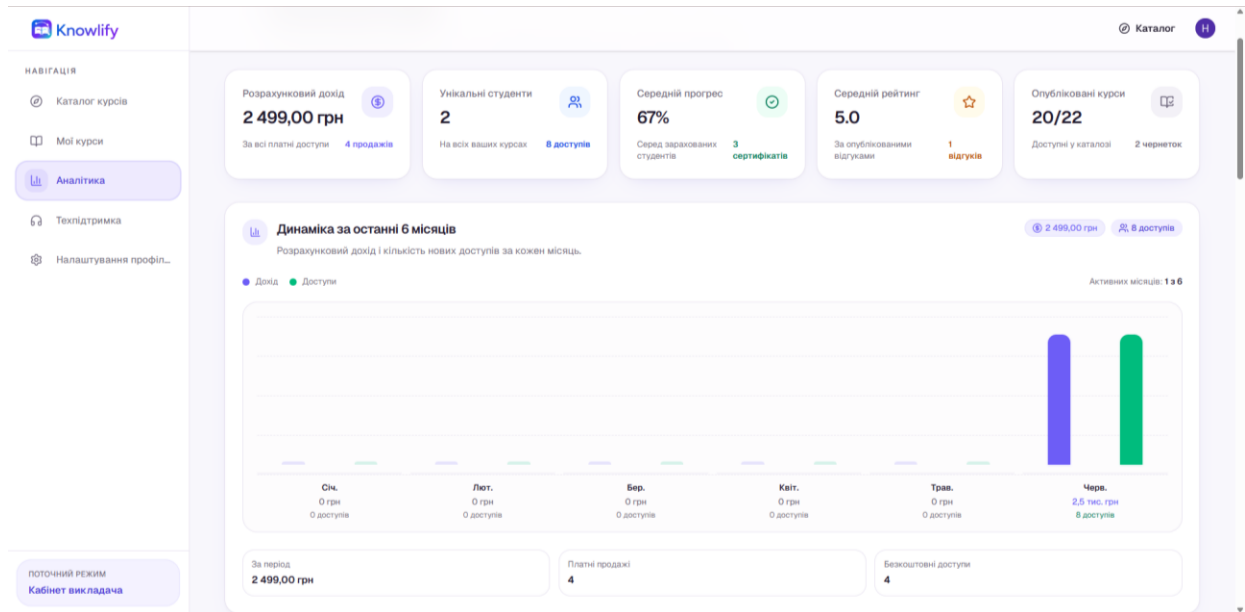


Рис. 3.23. Основні аналітичні показники курсів викладача

Інтерфейс викладача забезпечує створення й налаштування курсів, формування структури глав, додавання відео та навчальних ресурсів, підготовку тестів і публікацію матеріалів. Аналітичний розділ дає змогу контролювати доступи, активність студентів, прогрес проходження, рейтинг і фінансові показники курсів.

Для користувача з роллю адміністратора інтерфейс призначений для централізованого керування користувачами, курсами, категоріями, заявками на підтвердження статусу та зверненнями технічної підтримки. На головній панелі відображаються дохід, кількість користувачів, курсів, наданих доступів і заявок, які потребують розгляду. Додаткові блоки показують розподіл користувачів за ролями, стан курсів, типи доступу та результати опрацювання заявок. Головну панель керування адміністратора наведено на рис. 3.24.

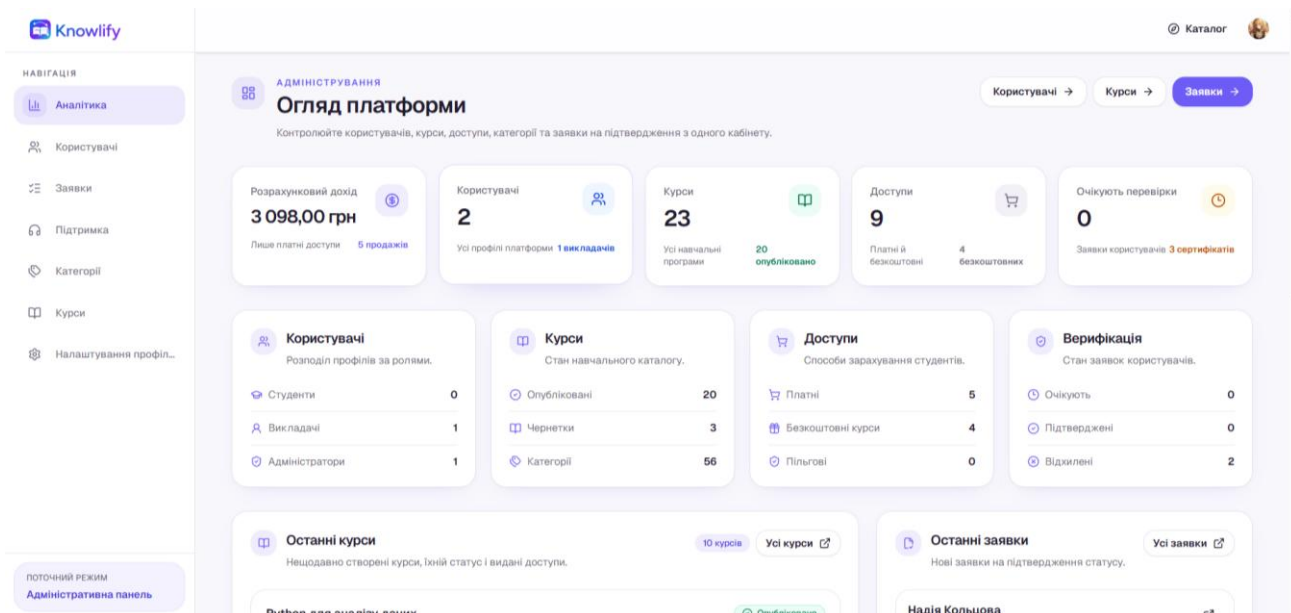


Рис. 3.24. Головна панель керування адміністратора

У розділі керування користувачами адміністратор може переглядати зареєстровані профілі, виконувати пошук за ім'ям або електронною адресою та фільтрувати записи за роллю і станом верифікації. У верхній частині сторінки відображаються загальна кількість користувачів і розподіл за ролями. Для вибраного профілю адміністратор може переглянути докладні відомості та змінити роль відповідно до повноважень користувача. Інтерфейс керування користувачами та їхніми ролями наведено на рис. 3.25.

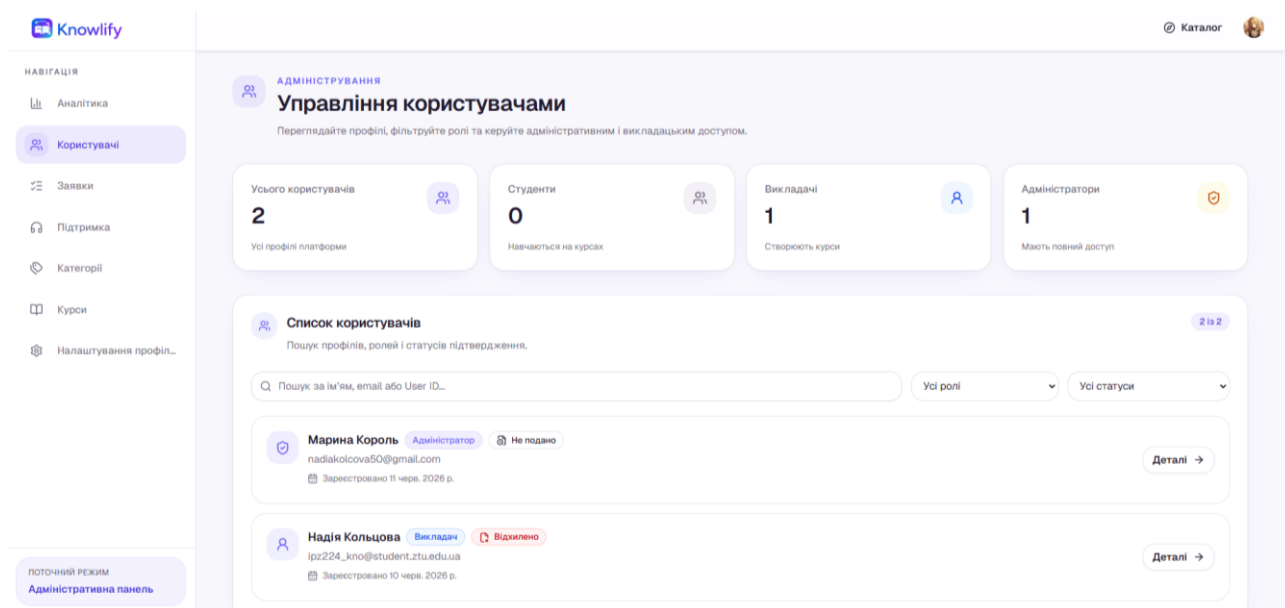


Рис. 3.25. Інтерфейс керування користувачами та їхніми ролями

Окремий розділ призначений для опрацювання заявок на підтвердження статусу ВПО або ветерана. На сторінці відображаються загальна кількість заявок і кількість записів, що очікують розгляду, були підтверджені або відхилені. Адміністратор може виконувати пошук та фільтрувати заявки за їхнім станом і типом. Кожний запис містить ім'я користувача, електронну адресу, заявлений статус, кількість доданих документів, дату подання та поточний стан. Сторінку перегляду заявок на підтвердження статусу наведено на рис. 3.26.

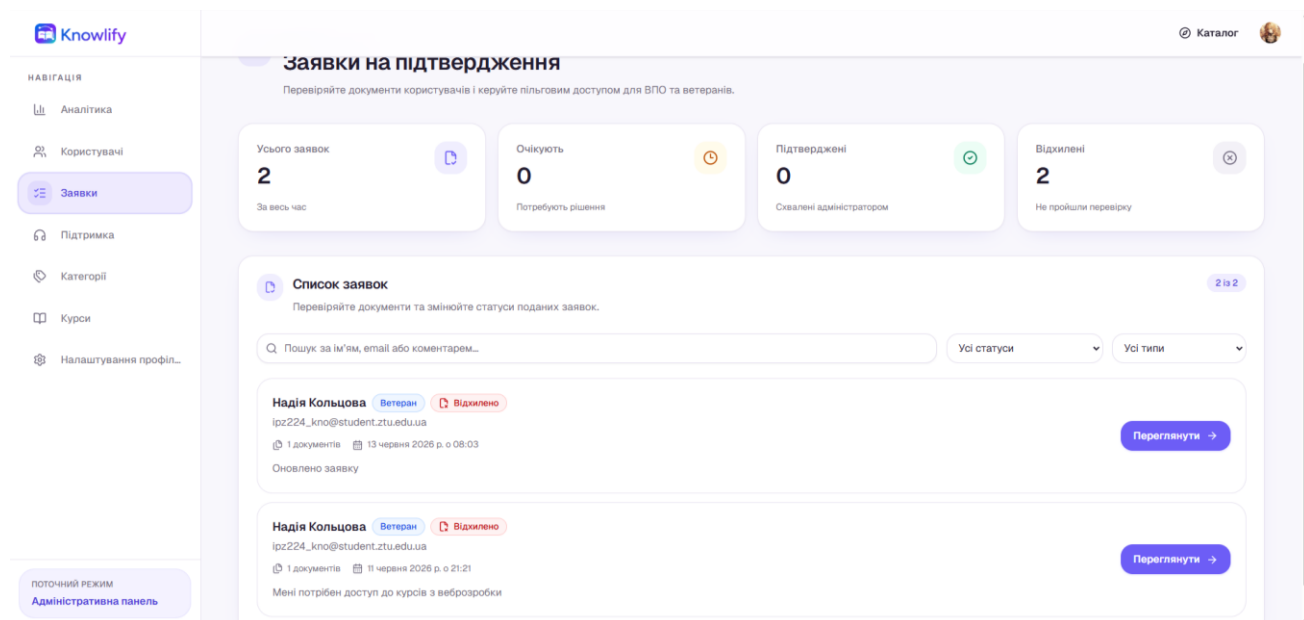


Рис. 3.26. Сторінка перегляду заявок на підтвердження статусу

Після відкриття заявки адміністратор отримує доступ до відомостей профілю, коментаря користувача, завантажених документів і даних про подання. На сторінці відображаються тип заявки, дати її створення й оновлення та поточний стан верифікації профілю. Після перевірки документів адміністратор може підтвердити або відхилити заявку. У разі відхилення зазначається причина рішення, яку користувач зможе переглянути у своєму профілі.

Після збереження рішення стан заявки та відомості про верифікацію профілю оновлюються одночасно. Підтверджений статус дає користувачеві можливість оформлювати пільговий доступ до окремих платних курсів, але не відкриває їх автоматично. Інтерфейс розгляду заявки наведено на рис. 3.27.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				82

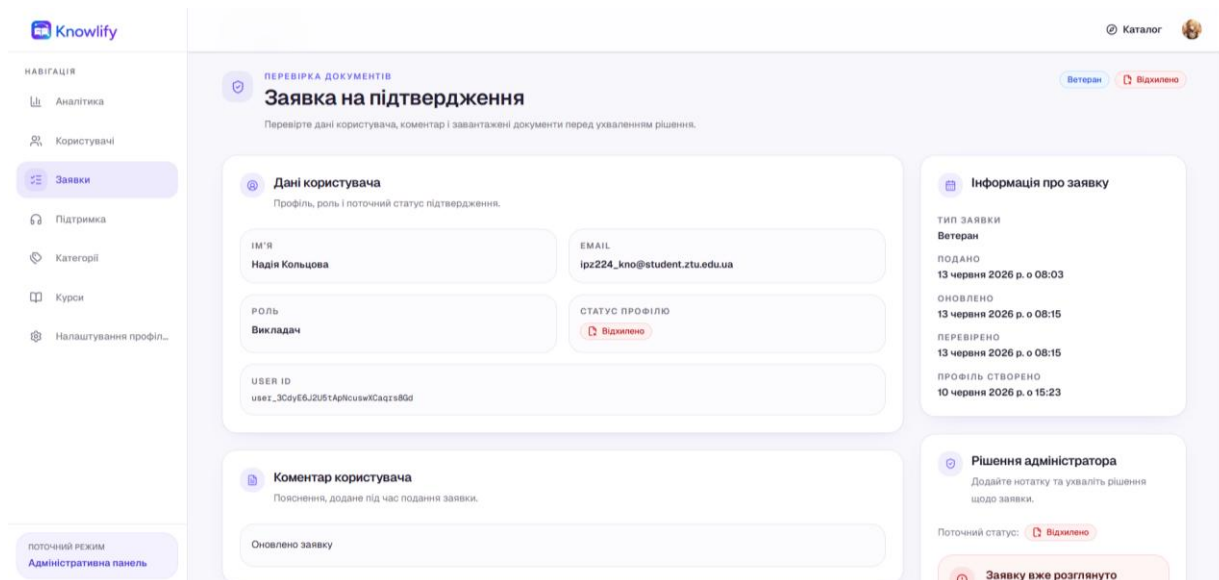


Рис. 3.27. Інтерфейс розгляду заявки на підтвердження статусу користувача

Адміністративний модуль технічної підтримки використовується для опрацювання звернень студентів і викладачів. На сторінці відображаються загальна кількість звернень та їх розподіл за станами: відкриті, опрацьовувані й закриті. Адміністратор може знаходити звернення за темою, ім'ям або електронною адресою користувача, а також застосовувати фільтрацію за категорією та станом. Після відкриття звернення доступна історія листування, форма відповіді та засоби зміни стану. Сторінку опрацювання звернень технічної підтримки наведено на рис. 3.28.

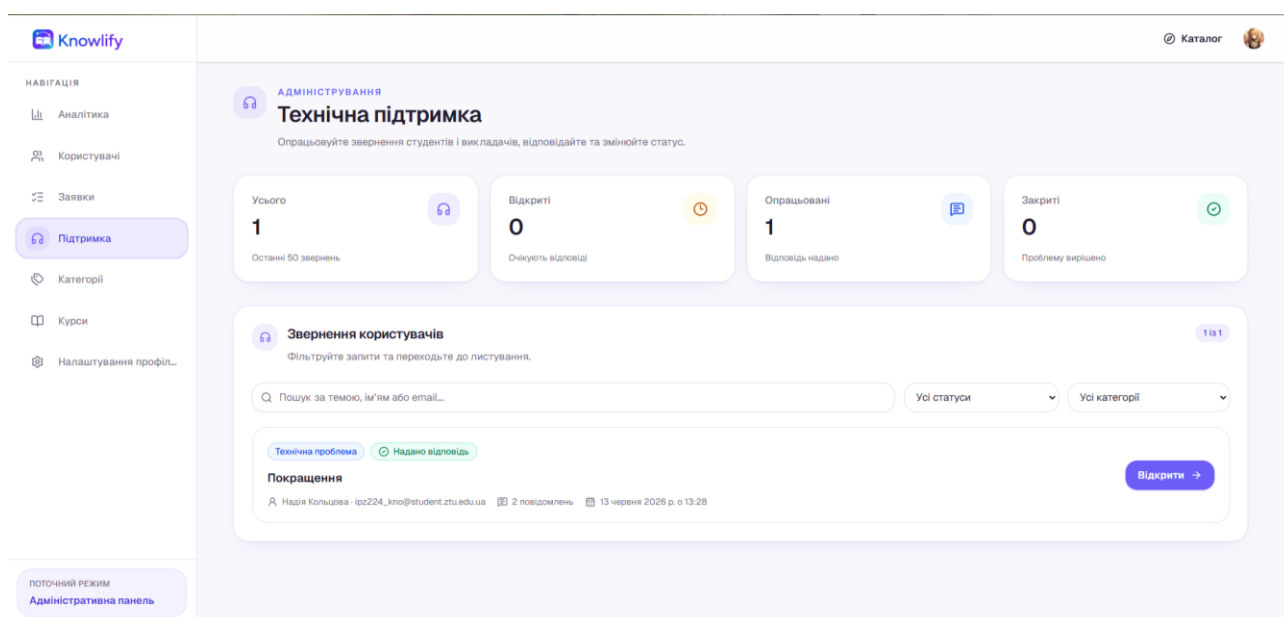


Рис. 3.28. Сторінка опрацювання звернень технічної підтримки

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				83

Для впорядкування каталогу адміністратор може створювати, перейменовувати та видаляти невикористані категорії. На сторінці відображаються кількість категорій і пов'язаних із ними курсів. Перед видаленням система перевіряє, чи використовується категорія в каталозі. Сторінку керування категоріями наведено на рис. 3.29.

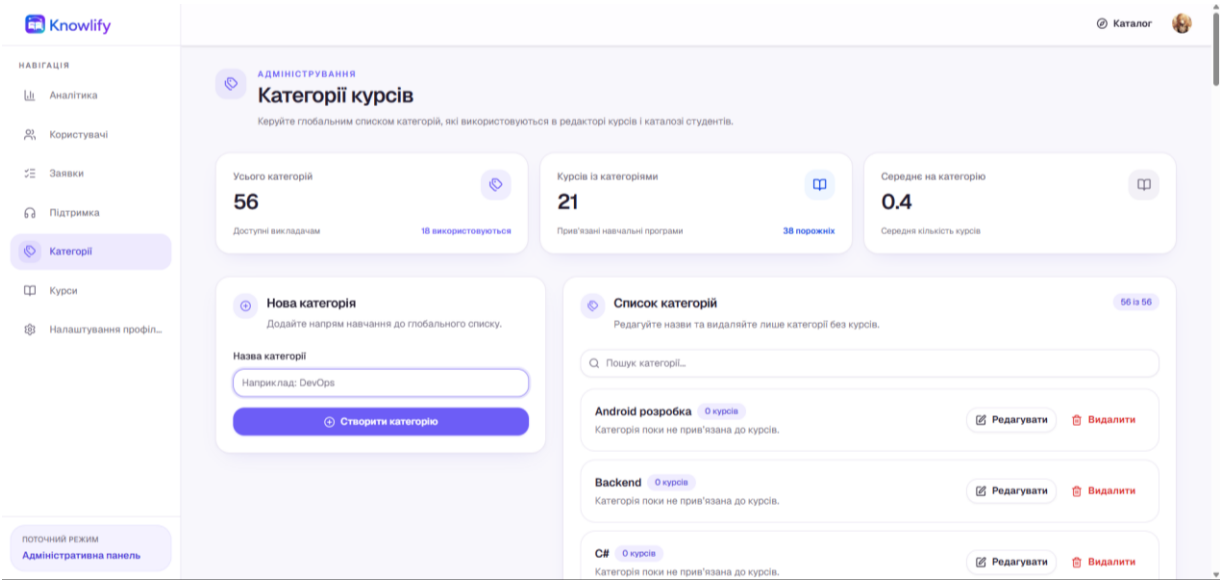


Рис. 3.29. Інтерфейс керування категоріями навчальних курсів

Адміністратор також може переглядати курси всіх викладачів. На сторінці наведено загальну кількість курсів, опублікованих програм, чернеток і наданих доступів. Пошук і фільтри дають змогу знайти курс за назвою, категорією або станом публікації. Сторінку керування курсами наведено на рис. 3.30.

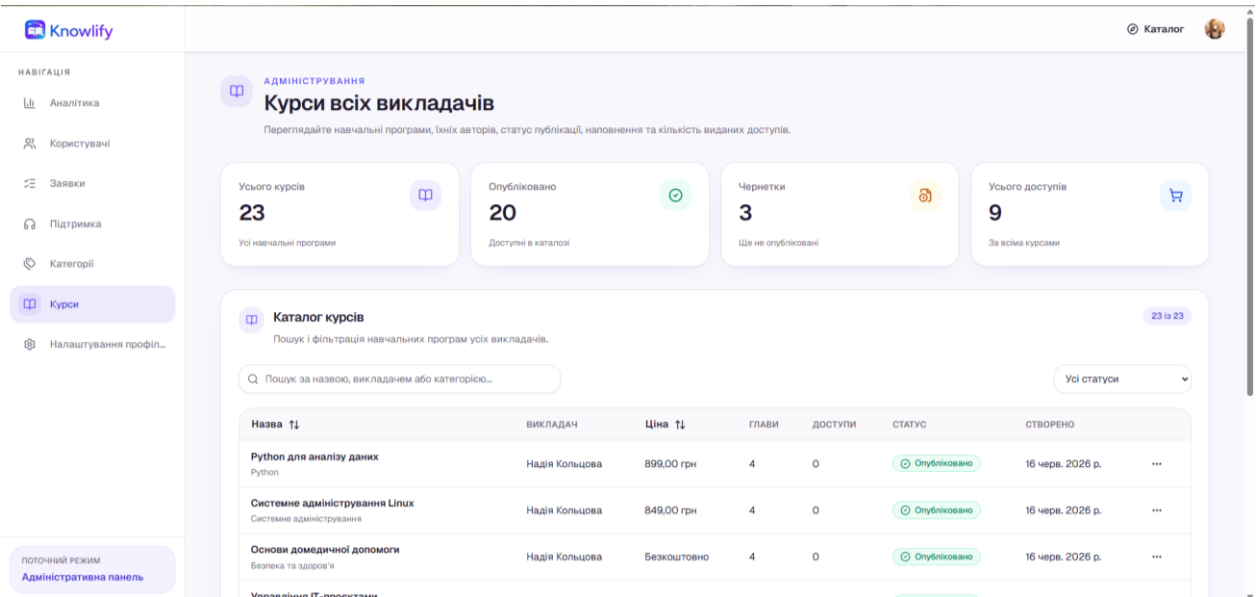


Рис. 3.30. Сторінка керування курсами всіх викладачів

Адміністративний інтерфейс дає змогу керувати користувачами й ролями, розглядати заявки на підтвердження статусу, опрацьовувати звернення технічної підтримки, підтримувати перелік категорій і контролювати курси викладачів. Пошук, фільтри та статистичні показники спрощують роботу з даними.

Інтерфейс вебплатформи адаптовано до пристроїв із різними розмірами екрана. У мобільній версії бічна навігаційна панель замінюється компактним меню, а статистичні блоки й елементи керування розміщуються вертикально. Відображення кабінету викладача на мобільному пристрої наведено на рис. 3.31.

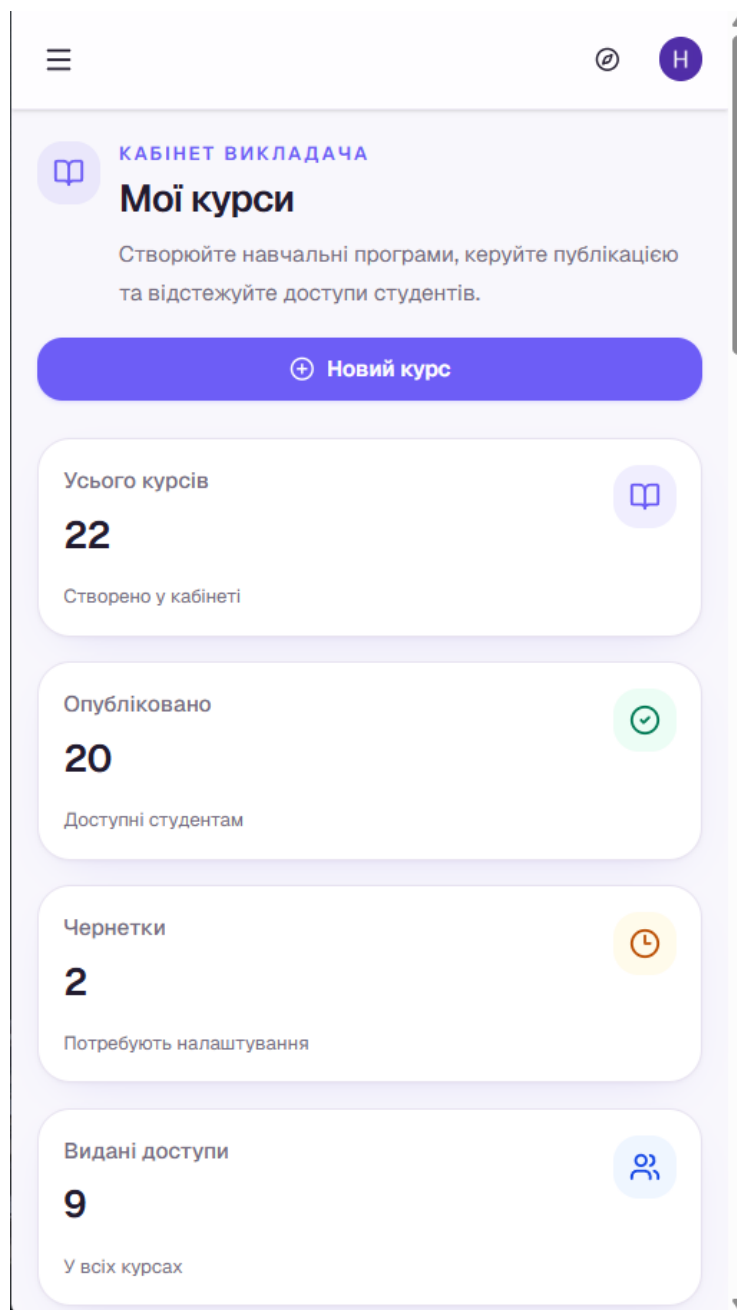


Рис. 3.31. Відображення кабінету викладача на мобільному пристрої

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				85

Рольове розмежування та адаптивне компонування забезпечують зручну роботу користувачів на комп'ютерах, планшетах і смартфонах.

3.3. Тестування роботи програмного продукту

Метою тестування розробленої вебплатформи є перевірка відповідності реалізованих функцій визначеним вимогам, правильності оброблення введених даних, дотримання рольового розмежування доступу та стабільності взаємодії із зовнішніми сервісами.

Основним методом було функціональне тестування за принципом чорної скриньки. Перевірка проводилася через користувацький інтерфейс без аналізу внутрішньої реалізації окремих програмних модулів. Для кожного тестового випадку визначено послідовність дій, очікуваний і фактичний результати. Під час тестування враховувалися як успішні сценарії виконання операцій, так і ситуації з помилковим або неповним введенням даних. Результати перевірки фіксувалися для подальшого порівняння з очікуваною поведінкою системи.

Під час тестування використовувалися облікові записи з ролями студента, викладача й адміністратора. Перевірялися реєстрація та автентифікація, пошук навчальних курсів, отримання доступу до них, виконання тестової оплати, збереження навчального прогресу, проходження тестів, формування сертифіката, створення та публікація навчального контенту, опрацювання заявок і звернень технічної підтримки.

Перевірки виконувалися послідовно, від реєстрації та входу до повного проходження курсу й отримання сертифіката, з особливою увагою до збереження даних після оновлення сторінки та рольового розмежування доступу.

Окремо перевірено правильність відображення інтерфейсу на комп'ютері, планшеті та смартфоні. Для контролю якості програмного коду й готовності застосунку до розгортання виконано статичний аналіз за допомогою ESLint та створено виробничу збірку вебплатформи.

Результати функціонального тестування вебплатформи наведено в табл. 3.1.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				
						86

Таблиця 3.1

Результати функціонального тестування вебплатформи

№	Тестовий сценарій	Послідовність дій	Очікуваний результат	Фактичний результат	Статус
1	Реєстрація користувача	Заповнити форму коректними даними	Обліковий запис створено	Обліковий запис створено, доступ надано	Пройдено
2	Перевірка обов'язкових полів	Надіслати незаповнену форму	Форму не надіслано, показано помилку	Поля позначено, операцію заблоковано	Пройдено
3	Пошук навчального курсу	Увести частину назви курсу	Відображено відповідні курси	Каталог оновлено за запитом	Пройдено
4	Отримання доступу до безоплатного курсу	Відкрити безоплатний курс і оформити доступ	Створено доступ до курсу	Курс додано до доступних програм	Пройдено
5	Оплата платного курсу	Додати курс до кошика та виконати тестову оплату	Курс стає доступним після оплати	Платіж підтверджено, доступ надано	Пройдено
6	Захист платного матеріалу	Відкрити платну главу без доступу	Матеріал недоступний	Доступ до глави заблоковано	Пройдено
7	Збереження позиції відео	Переглянути відео та оновити сторінку	Позицію перегляду збережено	Відтворення відновлено з останньої позиції	Пройдено
8	Проходження тесту до глави	Виконати тест і надіслати результат	Результат обчислено та збережено	Результат відображено користувачеві	Пройдено
9	Збереження результату тесту	Оновити сторінку після тестування	Результат залишається доступним	Результат не втрачено	Пройдено

Кольцова Н.О.

Савіцький Р.С.

ІПЗ.КР.Б-121-26-ПЗ

Арк.

87

№	Тестовий сценарій	Послідовність дій	Очікуваний результат	Фактичний результат	Статус
10	Формування сертифіката	Завершити курс і скласти тест	Сертифікат сформовано	Сертифікат створено та відображено	Пройдено
11	Публікація незаповненого курсу	Спробувати опублікувати курс	Публікацію заблоковано	Курс залишився чернеткою	Пройдено
12	Публікація готового курсу	Заповнити всі обов'язкові дані	Курс опубліковано	Курс додано до каталогу	Пройдено
13	Рольове розмежування доступу	Відкрити захищені сторінки різних ролей	Доступ обмежено відповідно до ролі	Несанкціонований доступ заблоковано	Пройдено
14	Зміна ролі користувача	Змінити роль профілю	Роль оновлено	Нову роль збережено	Пройдено
15	Розгляд заявки на верифікацію	Підтвердити або відхилити заявку	Стан заявки оновлено	Рішення збережено	Пройдено
16	Опрацювання звернення підтримки	Надіслати відповідь і змінити стан	Дані звернення оновлено	Відповідь і стан збережено	Пройдено
17	Перевірка адаптивності інтерфейсу	Відкрити основні сторінки на комп'ютері, планшеті та смартфоні	Інтерфейс коректно пристосовується до ширини екрана	Елементи сторінок відображаються коректно, основні функції доступні	Пройдено

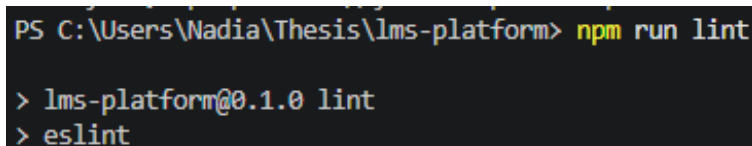
Під час перевірки рольового розмежування встановлено, що склад доступних сторінок і операцій залежить від ролі користувача. Студент не має доступу до засобів створення курсів та адміністративних розділів, викладач може керувати власними курсами, а адміністративні функції доступні лише користувачу з роллю адміністратора.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				88

Перевірка збереження навчального прогресу показала, що після повторного відкриття глави відновлюється остання позиція перегляду відео, а результати тестів до глав і підсумкового тесту не втрачаються після оновлення сторінки.

Адаптивність інтерфейсу перевірено в мобільному, планшетному та настільному режимах. За зміни ширини екрана елементи інтерфейсу пристосовуються до доступного простору, а основні функції вебплатформи залишаються доступними.

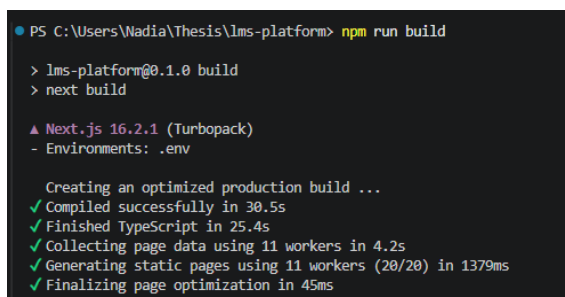
Для перевірки якості програмного коду виконано статичний аналіз за допомогою ESLint. Команда `npm run lint` завершилася без повідомлень про помилки або попередження. Результат статичного аналізу програмного коду наведено на рис. 3.32.



```
PS C:\Users\Wadia\Thesis\lms-platform> npm run lint
> lms-platform@0.1.0 lint
> eslint
```

Рис. 3.32. Результат статичного аналізу програмного коду

Для перевірки готовності програмного продукту до розгортання виконано команду `npm run build`. Під час створення виробничої збірки успішно завершено компіляцію програмного коду, перевірку типів TypeScript, формування статичних сторінок та оптимізацію застосунку. Помилки, які перешкоджали б створенню збірки, не виявлено. Результат створення виробничої збірки наведено на рис. 3.33.



```
PS C:\Users\Wadia\Thesis\lms-platform> npm run build
> lms-platform@0.1.0 build
> next build

▲ Next.js 16.2.1 (Turbopack)
- Environments: .env

Creating an optimized production build ...
✓ Compiled successfully in 30.5s
✓ Finished TypeScript in 25.4s
✓ Collecting page data using 11 workers in 4.2s
✓ Generating static pages using 11 workers (20/20) in 1379ms
✓ Finalizing page optimization in 45ms
```

Рис. 3.33. Результат створення виробничої збірки вебплатформи

Проведене тестування підтвердило коректну роботу основних модулів вебплатформи. Реалізовані сценарії відповідають визначеним вимогам, рольове

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				89

розмежування доступу працює належним чином, результати навчання зберігаються, а інтерфейс адаптується до різних розмірів екрана.

Висновки до третього розділу

У третьому розділі описано розгортання, структуру інтерфейсу та порядок роботи з вебплатформою дистанційного навчання. Розглянуто взаємодію клієнтської й серверної частин застосунку, бази даних та інтегрованих зовнішніх сервісів. Показано, що вебплатформа забезпечує централізоване опрацювання запитів, серверну перевірку прав доступу та збереження інформації в реляційній базі даних.

Описано інтерфейси основних ролей системи. Студент отримує засоби пошуку, придбання та проходження курсів, збереження навчального прогресу, виконання тестів, формування сертифікатів, подання заявок на пільговий доступ і звернень до технічної підтримки. Викладач створює й налаштовує курси, глави, відеоматеріали, навчальні ресурси та тести, публікує навчальний контент і переглядає аналітичні показники. Адміністратор керує користувачами, ролями, категоріями, курсами, заявками на підтвердження статусу та зверненнями технічної підтримки. Також перевірено адаптивне відображення інтерфейсу на комп'ютерах, планшетах і смартфонах.

Для перевірки програмного продукту проведено функціональне тестування основних користувацьких сценаріїв, рольового розмежування доступу, збереження прогресу, результатів тестування, роботи платіжного механізму та адаптивності інтерфейсу. Статичний аналіз за допомогою ESLint завершився без повідомлень про помилки, а виробничу збірку вебплатформи успішно сформовано командою `npm run build`.

Результати третього розділу підтверджують працездатність основних функціональних модулів, коректність рольового доступу, адаптивність інтерфейсу та готовність вебплатформи до розгортання й практичного використання.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				90

ВИСНОВКИ

Під час виконання бакалаврської кваліфікаційної роботи було досягнуто поставленої мети та розроблено вебплатформу дистанційного навчання. Система призначена для створення, публікації, придбання та проходження відеокурсів і підтримує роботу користувачів із ролями студента, викладача та адміністратора.

У процесі дослідження проаналізовано предметну область дистанційного навчання та наявні програмні аналоги. На основі отриманих результатів визначено функціональні й нефункціональні вимоги до системи, обґрунтовано вибір архітектури, технологій і зовнішніх сервісів. Спроектовано основні алгоритми роботи, об'єктно-орієнтовану структуру та реляційну базу даних.

Для студента реалізовано реєстрацію й автентифікацію, каталог курсів, пошук і фільтрацію, платний, безоплатний та пільговий доступ, кошик і оплату, перегляд відеоуроків і навчальних матеріалів, збереження прогресу, проходження тестів, формування сертифікатів, оцінювання курсів і технічну підтримку.

Викладач може створювати, редагувати та публікувати курси, формувати їх структуру, додавати глави, відеоматеріали, навчальні ресурси й тестові завдання. Також реалізовано перегляд основних аналітичних показників курсів. Адміністратору надано засоби керування користувачами, ролями, категоріями та курсами, розгляду заявок на підтвердження статусу й опрацювання звернень технічної підтримки.

Вебплатформу реалізовано з використанням Next.js, React, TypeScript, PostgreSQL і Prisma ORM. Автентифікацію організовано за допомогою Clerk, оплату курсів реалізовано через Stripe Checkout, роботу з відеоматеріалами через Mux, а завантаження файлів через UploadThing. Інтерфейс адаптовано до роботи на комп'ютерах, планшетах і смартфонах.

Функціональне тестування підтвердило коректність основних сценаріїв роботи системи, дотримання рольового доступу, збереження навчального прогресу та результатів тестування. Перевірка програмного коду за допомогою ESLint і створення виробничої збірки завершилися успішно. Отже, визначені

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				91

завдання виконано, а реалізований програмний продукт відповідає сформованим вимогам.

Особливістю розробленої вебплатформи є поєднання платного й безоплатного доступу до навчальних курсів із можливістю надання пільгового доступу користувачам із підтвердженим статусом. Розроблене рішення може використовуватися навчальними центрами, освітніми організаціями та авторами онлайн-курсів для розміщення навчального контенту й контролю результатів навчання.

Практична цінність роботи полягає у створенні готової основи для організації дистанційного навчання, керування освітнім контентом і контролю результатів студентів. Подальше вдосконалення системи може передбачати розширення аналітики, додавання персональних рекомендацій курсів, автоматизованих сповіщень, нових форматів навчальних завдань і додаткових засобів тестування програмного продукту.

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				92

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кольцова Н. О., Савіцький Р. С. Підтримка інклюзивності: курс для реінтеграції ВПО/ветеранів. Комп'ютерні технології: інновації, проблеми, рішення : тези доповідей VIII Всеукраїнської науково-технічної конференції, м. Житомир, 2–3 грудня 2025 р. Житомир : Державний університет «Житомирська політехніка», 2025. С. 358–359. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2026/01/povnyj-tekst-1.pdf> (дата звернення: 14.06.2026).
2. EdEra: українська студія онлайн-освіти. URL: <https://ed-era.com/> (дата звернення: 14.06.2026).
3. Mate academy : онлайн-курси IT-професій. URL: <https://mate.academy/> (дата звернення: 14.06.2026).
4. Coursera: Online Courses, Professional Certificates and Degrees. URL: <https://www.coursera.org/> (дата звернення: 14.06.2026).
5. Bass L., Clements P., Kazman R. Software Architecture in Practice. 4th ed. Boston : Addison-Wesley Professional, 2021. 464 p.
6. Next.js Documentation. URL: <https://nextjs.org/docs> (дата звернення: 14.06.2026).
7. React Documentation. URL: <https://react.dev/> (дата звернення: 14.06.2026).
8. TypeScript Documentation. URL: <https://www.typescriptlang.org/docs/> (дата звернення: 14.06.2026).
9. Tailwind CSS Documentation. URL: <https://tailwindcss.com/docs> (дата звернення: 14.06.2026).
10. Prisma ORM Documentation. URL: <https://www.prisma.io/docs/orm> (дата звернення: 14.06.2026).
11. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/> (дата звернення: 14.06.2026).
12. Neon Documentation. URL: <https://neon.com/docs/introduction> (дата звернення: 14.06.2026).

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				93

13. Clerk Documentation. URL: <https://clerk.com/docs> (дата звернення: 14.06.2026).
14. Stripe Documentation. URL: <https://docs.stripe.com/> (дата звернення: 14.06.2026).
15. Mux Documentation. URL: <https://www.mux.com/docs> (дата звернення: 14.06.2026).
16. UploadThing Documentation. URL: <https://docs.uploadthing.com/> (дата звернення: 14.06.2026).
17. Node.js Documentation. URL: <https://nodejs.org/docs/latest/api/> (дата звернення: 14.06.2026).
18. npm Documentation. URL: <https://docs.npmjs.com/> (дата звернення: 14.06.2026).
19. Git Documentation. URL: <https://git-scm.com/doc> (дата звернення: 14.06.2026).
20. GitLab Documentation. Merge Requests. URL: https://docs.gitlab.com/user/project/merge_requests/ (дата звернення: 14.06.2026).
21. Vercel Documentation. URL: <https://vercel.com/docs> (дата звернення: 14.06.2026).
22. ISO/IEC 25010:2023. Systems and software engineering. Systems and software Quality Requirements and Evaluation. Product quality model. Geneva: International Organization for Standardization, 2023. URL: <https://www.iso.org/standard/78176.html> (дата звернення: 14.06.2026).
23. Web Content Accessibility Guidelines 2.2. W3C Recommendation. URL: <https://www.w3.org/TR/WCAG22/> (дата звернення: 14.06.2026).
24. OWASP Top Ten Web Application Security Risks. URL: <https://owasp.org/www-project-top-ten/> (дата звернення: 14.06.2026).
25. MDN Web Docs. HTTP response status codes. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Status> (дата звернення: 14.06.2026).

		Кольцова Н.О.			ІІЗ.КР.Б-121-26-ІІЗ	Арк.
		Савіцький Р.С.				94

26. shadcn/ui Documentation. URL: <https://ui.shadcn.com/docs> (дата звернення: 14.06.2026).
27. Lucide for React. URL: <https://lucide.dev/guide/react> (дата звернення: 14.06.2026).
28. Zustand Documentation. URL: <https://zustand.docs.pmnd.rs/> (дата звернення: 14.06.2026).
29. TanStack Table Documentation. URL: <https://tanstack.com/table/latest/docs/introduction> (дата звернення: 14.06.2026).
30. Axios Documentation. URL: <https://axios-http.com/docs/intro> (дата звернення: 14.06.2026).
31. React Hot Toast Documentation. URL: <https://react-hot-toast.com/docs> (дата звернення: 14.06.2026).

ДОДАТКИ

		Кольцова Н.О.			ІІЗ.КР.Б-121-26-ІІЗ	Арк.
		Савіцький Р.С.				96

Лістинг вихідного коду

Лістинг А.1. Синхронізація профілю та перевірка ролі користувача

Файли lib/auth/get-current-profile.ts і lib/auth/require-role.ts.

```

export const getCurrentProfile = async () => {
  const { userId } = await auth();

  if (!userId) {
    return null;
  }

  const user = await currentUser();

  const email =
    user?.emailAddresses?.[0]?.emailAddress ?? null;

  const name =
    user?.fullName ||
    user?.firstName ||
    user?.username ||
    email ||
    "Користувач";

  const profile = await db.profile.upsert({
    where: {
      userId,
    },
    update: {
      name,
      email,
      imageUrl: user?.imageUrl ?? null,
    },
    create: {
      userId,
      name,
      email,
      imageUrl: user?.imageUrl ?? null,
    },
  });

  return profile;
};

export const requireApiRole = async (
  allowedRoles: readonly AppRole[]
) => {
  const profile = await getCurrentProfile();
  if (!profile) {

```

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				97
	.					

```

    return {
      profile: null,
      response: new NextResponse(
        "Не авторизовано",
        { status: 401 }
      ),
    };
  }

  if (!isAllowedRole(profile.role, allowedRoles)) {
    return {
      profile: null,
      response: new NextResponse(
        "Недостатньо прав доступу",
        { status: 403 }
      ),
    };
  }

  return {
    profile,
    response: null,
  };
};

```

Лістинг А.2. Перевірка готовності та публікація навчального курсу

Файл `app/api/courses/[courseId]/publish/route.ts`.

```

export async function PATCH(
  _req: Request,
  {
    params,
  }: {
    params: Promise<{
      courseId: string;
    }>;
  }
) {
  const { profile, response } =
    await requireApiRole(TEACHER_ROLES);

  if (response) {
    return response;
  }

  if (!profile) {
    return new NextResponse(
      "Не авторизовано",
      { status: 401 }
    );
  }

  const { courseId } = await params;

```

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				98

```

const course = await db.course.findUnique({
  where: {
    id: courseId,
  },
  include: {
    chapters: {
      where: {
        isPublished: true,
      },
    },
    finalQuiz: {
      include: {
        questions: {
          include: {
            options: true,
          },
        },
      },
    },
  },
});

if (!course) {
  return new NextResponse(
    "Курс не знайдено",
    { status: 404 }
  );
}

const isOwner =
  course.userId === profile.userId;

const isAdmin =
  profile.role === "ADMIN";

if (!isOwner && !isAdmin) {
  return new NextResponse(
    "Доступ заборонено",
    { status: 403 }
  );
}

const hasPublishedChapter =
  course.chapters.length > 0;

const hasPublishedFinalQuiz =
  !!course.finalQuiz?.isFinal &&
  course.finalQuiz.isPublished &&
  course.finalQuiz.questions.length > 0;

if (!hasPublishedFinalQuiz) {
  return new NextResponse(
    "Для публікації курсу потрібно " +

```

		Кольцова Н.О.			ІІЗ.КР.Б-121-26-ІІЗ	Арк.
		Савіцький Р.С.				99

```

        "створити та опублікувати підсумковий тест",
        { status: 400 }
    );
}

const invalidQuestion =
    course.finalQuiz.questions.find(
        (question) => {
            const correctOptions =
                question.options.filter(
                    (option) => option.isCorrect
                );

            return (
                question.options.length < 2 ||
                correctOptions.length !== 1
            );
        }
    );

if (invalidQuestion) {
    return new NextResponse(
        "Структура підсумкового тесту є некоректною",
        { status: 400 }
    );
}

const requiredFields = [
    course.title,
    course.description,
    course.imageUrl,
    course.price !== null &&
        course.price !== undefined,
    course.categoryId,
    hasPublishedChapter,
    hasPublishedFinalQuiz,
];

if (!requiredFields.every(Boolean)) {
    return new NextResponse(
        "Заповніть усі обов'язкові поля",
        { status: 400 }
    );
}

const publishedCourse =
    await db.course.update({
        where: {
            id: courseId,
        },
        data: {
            isPublished: true,
        },
    });

```

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				100

```
});

return NextResponse.json(publishedCourse);
}
Лістинг А.3. Перевірка події Stripe та надання доступу до курсів
```

Файли app/api/webhook/route.ts і lib/checkout.ts.

```
const processCompletedSession = async (
  session: Stripe.Checkout.Session
) => {
  const isPaid =
    session.payment_status === "paid" ||
    session.payment_status ===
      "no_payment_required";

  if (!isPaid) {
    return;
  }

  const userId = session.metadata?.userId;

  if (!userId || !stripe) {
    return;
  }

  const courseIds =
    await getCheckoutCourseIds(
      stripe,
      session
    );

  if (courseIds.length === 0) {
    return;
  }

  await grantPaidCourseAccess({
    userId,
    courseIds,
  });
};

export async function POST(req: Request) {
  if (
    !stripe ||
    !process.env.STRIPE_WEBHOOK_SECRET
  ) {
    return new NextResponse(
      "Stripe не налаштований",
      { status: 503 }
    );
  }
  const body = await req.text();
```

		Кольцова Н.О.			ПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				101

```

const headersList = await headers();

const signature =
  headersList.get("stripe-signature");

if (!signature) {
  return new NextResponse(
    "Stripe signature відсутній",
    { status: 400 }
  );
}

let event: Stripe.Event;

try {
  event =
    stripe.webhooks.constructEvent(
      body,
      signature,
      process.env.STRIPE_WEBHOOK_SECRET
    );
} catch {
  return new NextResponse(
    "Webhook signature verification failed",
    { status: 400 }
  );
}

if (
  event.type ===
    "checkout.session.completed" ||
  event.type ===
    "checkout.session.async_payment_succeeded"
) {
  const session =
    event.data
      .object as Stripe.Checkout.Session;

  await processCompletedSession(session);
}

return new NextResponse(null, {
  status: 200,
});
}

export const grantPaidCourseAccess = async ({
  userId,
  courseIds,
}): {
  userId: string;
  courseIds: string[];
} => {

```

		Кольцова Н.О.			ПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				102

```

const uniqueCourseIds = Array.from(
  new Set(
    courseIds
      .map((courseId) => courseId.trim())
      .filter(Boolean)
  )
);

const existingCourses =
  await db.course.findMany({
    where: {
      id: {
        in: uniqueCourseIds,
      },
    },
    select: {
      id: true,
    },
  });

const validCourseIds =
  existingCourses.map(
    (course) => course.id
  );

await db.purchase.createMany({
  data: validCourseIds.map(
    (courseId) => ({
      userId,
      courseId,
      accessType: "PAID",
    })
  ),
  skipDuplicates: true,
});

return validCourseIds;
};

```

Лістинг А.4. Збереження прогресу проходження навчальної глави

Файл app/api/courses/[courseId]/chapters/[chapterId]/progress/route.ts.

```

export async function PATCH(
  req: Request,
  {
    params,
  }: {
    params: Promise<{
      courseId: string;
      chapterId: string;
    }>;
  }
) {
  const { userId } = await auth();

```

```

const { courseId, chapterId } =
  await params;

const values = await req.json();

if (!userId) {
  return new NextResponse(
    "Не авторизовано",
    { status: 401 }
  );
}

const chapter =
  await db.chapter.findFirst({
    where: {
      id: chapterId,
      courseId,
      isPublished: true,
    },
    select: {
      id: true,
    },
  });

if (!chapter) {
  return new NextResponse(
    "Главу не знайдено",
    { status: 404 }
  );
}

const purchase =
  await db.purchase.findUnique({
    where: {
      userId_courseId: {
        userId,
        courseId,
      },
    },
    select: {
      id: true,
    },
  });

if (!purchase) {
  return new NextResponse(
    "Щоб зберігати прогрес, " +
    "спочатку отримайте доступ до курсу",
    { status: 403 }
  );
}

```

		Кольцова Н.О.			ПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				104

```

const data: {
  isCompleted?: boolean;
  videoTimestamp?: number;
} = {};

if (
  typeof values.isCompleted === "boolean"
) {
  data.isCompleted =
    values.isCompleted;
}

if (
  typeof values.videoTimestamp ===
    "number" &&
  Number.isFinite(
    values.videoTimestamp
  ) &&
  values.videoTimestamp >= 0
) {
  data.videoTimestamp =
    Math.floor(values.videoTimestamp);
}

if (Object.keys(data).length === 0) {
  return new NextResponse(
    "Немає даних для оновлення",
    { status: 400 }
  );
}

const userProgress =
  await db.userProgress.upsert({
    where: {
      userId_chapterId: {
        userId,
        chapterId,
      },
    },
    update: data,
    create: {
      userId,
      chapterId,
      isCompleted:
        data.isCompleted ?? false,
      videoTimestamp:
        data.videoTimestamp ?? 0,
    },
  });

return NextResponse.json(userProgress);
}

```

Лістинг А.5. Обчислення та збереження результату тестування

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				105

Файл app/api/courses/[courseId]/chapters/[chapterId]/quiz/attempts/route.ts.

```
if (
  answers.length !==
  quiz.questions.length
) {
  return new NextResponse(
    "Дайте відповідь на всі питання",
    { status: 400 }
  );
}

const answerMap = new Map(
  answers.map((answer) => [
    answer.questionId,
    answer.selectedOptionId,
  ])
);

if (
  answerMap.size !==
  quiz.questions.length
) {
  return new NextResponse(
    "Дайте відповідь на всі питання",
    { status: 400 }
  );
}

let correctAnswersCount = 0;

const attemptAnswers: {
  questionId: string;
  selectedOptionId: string;
  isCorrect: boolean;
}[] = [];

for (const question of quiz.questions) {
  const selectedOptionId =
    answerMap.get(question.id);

  if (!selectedOptionId) {
    return new NextResponse(
      "Дайте відповідь на всі питання",
      { status: 400 }
    );
  }

  const selectedOption =
    question.options.find(
      (option) =>
        option.id === selectedOptionId
    );
}
```

		Кольцова Н.О.			ІПЗ.КР.Б-121-26-ІЗ	Арк.
		Савіцький Р.С.				106

```

    if (!selectedOption) {
        return new NextResponse(
            "Некоректна відповідь у тесті",
            { status: 400 }
        );
    }

    const isCorrect =
        selectedOption.isCorrect;

    if (isCorrect) {
        correctAnswersCount += 1;
    }

    attemptAnswers.push({
        questionId: question.id,
        selectedOptionId,
        isCorrect,
    });
}

const score = Math.round(
    (
        correctAnswersCount /
        quiz.questions.length
    ) * 100
);

const isPassed =
    score >= quiz.passingScore;
const attempt =
    await db.quizAttempt.create({
        data: {
            userId,
            quizId: quiz.id,
            score,
            isPassed,
            answers: {
                create: attemptAnswers,
            },
        },
    });

return NextResponse.json({
    attemptId: attempt.id,
    score,
    isPassed,
    correctAnswersCount,
    totalQuestions:
        quiz.questions.length,
    passingScore: quiz.passingScore,
});

```

		Кольцова Н.О.			ІІЗ.КР.Б-121-26-ІІЗ	Арк.
		Савіцький Р.С.				107

Лістинг А.6. Транзакційне створення заявки на верифікацію

Файл `app/api/verification/route.ts`.

```
const pendingRequest =
  await db.verificationRequest.findFirst({
    where: {
      userId: profile.userId,
      status: "PENDING",
    },
  });

if (pendingRequest) {
  return new NextResponse(
    "У вас уже є заявка на перевірку",
    { status: 400 }
  );
}

const verificationRequest =
  await db.$transaction(
    async (tx) => {
      const request =
        await tx.verificationRequest.create({
          data: {
            userId: profile.userId,
            type,
            status: "PENDING",
            comment,
            documents: {
              create:
                uniqueDocuments.map(
                  (document) => ({
                    url: document.url,
                    fileName:
                      document.fileName,
                    fileKey:
                      document.fileKey,
                    fileSize:
                      document.fileSize,
                    mimeType:
                      document.mimeType,
                  })
                ),
            },
          },
          include: {
            documents: true,
          },
        });

      await tx.profile.update({
        where: {
          userId: profile.userId,
```

		Кольцова Н.О.			ПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				108

```

        },
        data: {
            verificationType: type,
            verificationStatus:
                "PENDING",
        },
    });

    return request;
}

);

return NextResponse.json(
    verificationRequest
);

```

		Кольцова Н.О.			ПЗ.КР.Б-121-26-ПЗ	Арк.
		Савіцький Р.С.				109