

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:

**«Онлайн-платформа проходження курсів»**

Виконав студент групи ПЗкх-20-1  
КОВБАН Роман Олександрович

Керівник роботи:  
ЛОБАНЧИКОВА Надія Миколаївна

Рецензент:  
ВЛАСЕНКО Олег Васильович

Житомир – 2024

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

**«ЗАТВЕРДЖУЮ»**

Завідувач кафедри інженерії  
програмного забезпечення

Тетяна ВАКАЛЮК

«26» лютого 2024 р.

**ЗАВДАННЯ**  
**на кваліфікаційну роботу**

Здобувач вищої освіти: **КОВБАН Роман Олександрович**

Керівник роботи: **ЛОБАНЧИКОВА Надія Миколаївна**

Тема роботи: **«Онлайн-платформа проходження курсів»**,

затверджена Наказом закладу вищої освіти від **«23» лютого 2024 р., №74с**

Вихідні дані для роботи: створення та реалізація онлайн-платформи проходження курсів, яка використовує сучасні вебтехнології та API для забезпечення безперебійного доступу до великої кількості курсів.

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

| Розділ | Консультант       | Завдання видав | Завдання прийняв |
|--------|-------------------|----------------|------------------|
| 1      | Надія ЛОБАНЧИКОВА | 28.02.2024р    | 28.02.2024р      |
| 2      | Надія ЛОБАНЧИКОВА | 26.03.2024р    | 26.03.2024р      |
| 3      | Надія ЛОБАНЧИКОВА | 22.04.2024р    | 22.04.2024р      |

## РЕФЕРАТ

Випускна кваліфікаційна робота складається з програмної онлайн-платформи для надання послуг у навчанні за допомогою проходження курсів та пояснювальної записки. Пояснювальна записка до кваліфікаційної роботи містить 55 сторінок, 46 ілюстрацій, 6 таблиць, 3 додатки та 22 літературних джерел. Загальний обсяг роботи – 72 сторінки.

Метою випускної кваліфікаційної роботи є проектування архітектури та розробка програмного продукту – онлайн-платформи для проходження курсів, яка забезпечує можливість створення та вивчення навчальних матеріалів для забезпечення доступу до якісної освіти та навчання для широкої аудиторії.

У роботі було описано основні цілі та задачі системи, визначено вимоги, які повинні бути дотримані при створенні додатку. Було проаналізовано існуючі аналоги програмних продуктів для отримання сильних та слабких сторін. При проектуванні онлайн-платформи було створено гнучку та масштабовану архітектуру для подальшого розвитку системи. З готовою моделлю структури проекту було розроблено базу даних, визначено основні алгоритми роботи додатку, описано виконання програмної частини створення додатку.

**КЛЮЧОВІ СЛОВА:** ПЛАТФОРМА, ОНЛАЙН-КУРСИ, НАВЧАННЯ.

|           |      |                  |        |      |                                                                |                                              |         |
|-----------|------|------------------|--------|------|----------------------------------------------------------------|----------------------------------------------|---------|
|           |      |                  |        |      | ІПЗ.КР.Б-121-24-ПЗ                                             |                                              |         |
|           |      |                  |        |      |                                                                |                                              |         |
| Змн.      | Арк. | № докум.         | Підпис | Дата |                                                                |                                              |         |
| Розроб.   |      | Ковбан Р.О.      |        |      | Онлайн-платформа<br>проходження курсів<br>Пояснювальна записка | Літ.                                         | Арк.    |
| Керівник  |      | Лобанчикова Н.М. |        |      |                                                                |                                              | Аркушів |
| Рецензент |      | Власенко О.В.    |        |      |                                                                | 3                                            | 72      |
| Н. контр. |      | Тичина О.П.      |        |      |                                                                | Житомирська політехніка,<br>група ІПЗкх-20-1 |         |
| Зав. каф. |      | Вакалюк Т.А.     |        |      |                                                                |                                              |         |

## ABSTRACT

The final qualification work consists of a web-based software platform for providing services in education through the completion of courses and an explanatory note. The explanatory note to the qualification work contains 55 pages, 46 illustrations, 6 tables, 3 appendices and 22 references. The total volume of the work is 72 pages.

The purpose of the final qualification work is to design the architecture and develop a software product - an educational platform for online learning that provides the ability to create and study educational materials to ensure access to quality education and training for a wide audience.

The paper describes the main goals and objectives of the system under development, identifies the requirements that must be met when creating the application. Existing analogues of software products were analyzed to identify their strengths and weaknesses. This helped us determine what basic functions and features should be implemented in our product. When designing the web platform, we created a flexible and scalable architecture for further development of the system. With the finished model of the project structure, a database was developed, the main algorithms of the application were determined, and the implementation of the program part of the application creation was described.

KEYWORDS: PLATFORM, ONLINE COURSES, TRAINING.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 4    |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

## ЗМІСТ

|                                                                                               |           |
|-----------------------------------------------------------------------------------------------|-----------|
| <b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....</b>                                                        | <b>6</b>  |
| <b>ВСТУП.....</b>                                                                             | <b>7</b>  |
| <b>РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЙ СТВОРЕННЯ ОНЛАЙН-ПЛАТФОРМ.....</b>                    | <b>9</b>  |
| 1.1. Постановка задачі розробки.....                                                          | 9         |
| 1.2. Порівняння сучасних аналогічних продуктів .....                                          | 10        |
| 1.3. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення ..... | 15        |
| Висновки до першого розділу.....                                                              | 21        |
| <b>РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПЛАТФОРМИ .....</b>                                     | <b>23</b> |
| 2.1. Визначення варіантів використання та об'єктно-орієнтованої структури .....               | 23        |
| 2.2. Розробка структури бази даних системи .....                                              | 27        |
| 2.3. Проектування та реалізація алгоритмів роботи системи .....                               | 31        |
| 2.4. Реалізація функціоналу онлайн-платформи проходження курсів ..                            | 35        |
| Висновки до другого розділу .....                                                             | 37        |
| <b>РОЗДІЛ 3. ІНСТРУКЦІЯ КОРИСТУВАННЯ ТА РОЗГОРТАННЯ СИСТЕМИ. РЕЗУЛЬТАТИ ТЕСТУВАННЯ.....</b>   | <b>39</b> |
| 3.1. Особливості щодо встановлення та розгортання .....                                       | 39        |
| 3.2. Інтерфейс та порядок роботи .....                                                        | 40        |
| 3.3. Проведення та результати тестування платформи .....                                      | 56        |
| Висновки до третього розділу .....                                                            | 58        |
| <b>ВИСНОВКИ .....</b>                                                                         | <b>60</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>                                                       | <b>62</b> |
| <b>ДОДАТКИ.....</b>                                                                           | <b>64</b> |

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 5    |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – (Application Programing Interface), інтерфейс програмного забезпечення.

JWT – (JSON Web Token), відкритий стандарт для безпечної передачі даних.

UI – (User Interface), інтерфейс користувача.

UML – (Unified Modeling Language), уніфікована мова моделювання.

БД – база даних.

ОС – операційна система.

ПЗ – програмне забезпечення.

СУБД – система управління базами даних.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 6    |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

## ВСТУП

**Актуальність теми.** Навчання є невід’ємною частиною життя кожної людини. Завдяки доступу до Інтернету та технологічному прогресу, навчання онлайн стає все більш популярним у сучасному світі.

Багато людей шукають зручні та гнучкі способи отримання нових знань та навичок, без необхідності фізичної присутності в навчальних закладах. Он-лайн-навчання дозволяє здобувати освіту незалежно від місця перебування та доступності ресурсів. Крім того, освітні платформи надають навчальним закладам можливість залучати студентів з усього світу, розширюючи їхній вплив на глобальному рівні.

Однією з ключових переваг онлайн-навчання є постійний доступ до актуальної інформації. Навчальні матеріали можуть швидко оновлюватися відповідно до нових тенденцій, технологій та вимог ринку праці, що дає студентам змогу отримувати сучасні знання та навички.

Також важливо зазначити, що такі освітні платформи є корисними і для викладачів, надаючи їм інструменти для створення, редагування та організації навчального матеріалу. Викладачі можуть використовувати різноманітні методи подачі матеріалу, забезпечуючи якісне навчання для своїх студентів. Це також дозволяє викладачам експериментувати з новими методами викладання, покращуючи навчальний процес.

Розробка таких навчальних освітніх платформ дає можливість створити продукт з необмеженим потенціалом для розвитку та вдосконалення, інтегруючи новітні технології для підвищення якості освітнього процесу.

**Метою випускної кваліфікаційної роботи** є проектування архітектури та розробка програмного продукту – онлайн-платформи проходження курсів, яка забезпечує можливість створення та вивчення навчальних матеріалів для забезпечення доступу до якісної освіти та навчання для широкої аудиторії. Для досягнення цієї мети потрібно провести аналіз існуючих аналогічних

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 7    |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

платформ, розробити алгоритми та архітектуру системи, що забезпечать її ефективну роботу.

**Встановлена мета обумовлює наступні завдання:**

- Аналіз функціональних вимог до продукту.
- Розробка моделі для визначення структури та функцій платформи.
- Аналіз існуючих аналогів та вибір засобів реалізації.
- Розробка алгоритму роботи системи.
- Проектування структури БД.
- Вибір архітектури розробки онлайн-платформи
- Розробка структури та програмна реалізація.

**Об'єктом дослідження** є технологія створення онлайн-платформи для проходження курсів, яка включає основні аспекти додатку з метою створення оптимального середовища для освіти.

**Предметом** є алгоритми, структури та методи розробки онлайн-платформи проходження курсів.

Створена платформа забезпечує якісне надання доступу до освіти в онлайн форматі. Розроблений продукт сприяє популяризації принципів постійного вдосконалення шляхом навчання. Він може використовуватись у різних галузях, які включають у себе етапи навчання.

Випускна кваліфікаційна робота складається з вступу, 3-х розділів висновків, списку використаних джерел із 22 найменувань (2 стор.), 3-х додатків, 6-ти таблиць та 46 ілюстрацій по тексту. Повний обсяг роботи становить 72 сторінки, з них 55 сторінок основного тексту.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 8    |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

# РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЙ СТВОРЕННЯ ОНЛАЙН-ПЛАТФОРМ

## 1.1. Постановка задачі розробки

Оскільки попит на онлайн навчання постійно зростає, цей проєкт має велике практичне значення. Він спрямований на розробку онлайн-платформи проходження курсів, щоб забезпечити доступ до якісного навчання через можливість проходження відео-курсу.

Платформа дозволить користувачам обирати свою роль у системі, що визначатиме доступний їм функціонал. Проєкт надає можливість створювати та редагувати курси і матеріали, а також динамічно керувати доступом інших користувачів до цих матеріалів. Система також дає можливість купівлі курсів для проходження платних курсів.

Кожен відео-курс може бути оцінений за допомогою коментарів, що відображатимуть думки користувачів. Робота зі списками та матеріалами включає розширену систему фільтрів, що полегшує пошук та організацію контенту.

Основними етапами розробки платформи є:

1. Розробка дизайну вебсайту:
  - а) створення загальної концепції дизайну додатку;
  - б) створення UI компонентів;
  - в) розробка дизайну сторінок.
2. Побудова архітектури клієнтської частини додатку:
  - а) розробка клієнтської логіки додатку;
  - б) інтеграція з API серверної частини;
  - в) забезпечення адаптивності дизайну для різних пристроїв.
3. Побудова архітектури серверної частини додатку:
  - а) створення серверної логіки для обробки запитів користувачів;
  - б) реалізація функцій додавання, видалення та редагування контенту;
  - в) забезпечення зберігання даних у базі даних.
4. Розробка системи авторизації та ролей користувачів:

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 9    |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

- a) створення механізму реєстрації та входу користувачів;
- b) розподіл ролей (користувач, адміністратор);
- c) управління доступом до різних функцій залежно від ролі користувача.

5. Розробка основного функціоналу:

- a) Створення можливостей для додавання, редагування та управління курсами, уроками.
- b) Розробка системи купівлі курсів.
- c) Створення віддаленої бази даних для зберігання інформації;
- d) Організація спілкування серверної частини з клієнтською;
- e) Інтеграція можливості коментування та оцінювання курсу.
- f) Інтеграція можливості задавати питання під курсами та давати відповідь на запитання.

6. Інтеграція зовнішніх сервісів:

- a) Підключення платіжних систем та процесу стягування валют.
- b) Інтеграція з системами авторизації, наприклад OAuth.
- c) Інтеграція хмарного сховища для збереження файлів курсів, наприклад Cloudinary та VdoCipher.

7. Тестування додатку:

- a) функціональне тестування додатку;
- b) тестування графічного інтерфейсу;
- c) забезпечення стабільності роботи під високим навантаженням.

1.2. Порівняння сучасних аналогічних продуктів

Актуальність теми забезпечує широкий вибір вже існуючих аналогів онлайн платформ для навчання. У цьому розділі буде проведено аналіз, визначено переваги та недоліки таких ресурсів: ITVDN, Udemu, EdEra.

ITDVN – онлайн-ресурс, спеціалізований на наданні навчальних матеріалів зі сфери інформаційних технологій (рис.1.1). Ресурс надає велику кількість відеоуроків та курсів, які охоплюють технологічні теми.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 10   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

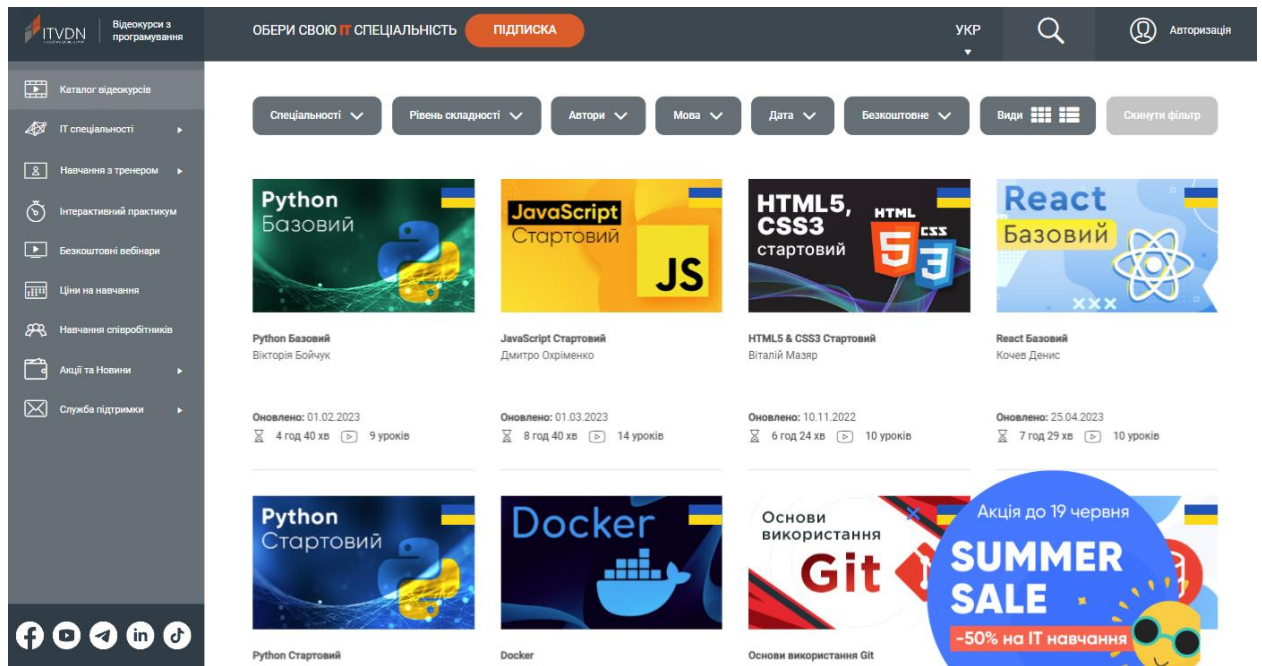


Рис. 1.1. Список курсів ITDVN

#### Переваги:

- Широкий вибір курсів у галузі інформаційних технологій.
- Підбір професійних фахівців для забезпечення високої якості викладання.
- Можливість отримання сертифікатів після успішного завершення курсів.
- Доступність курсів для всіх рівнів користувачів, від початківців до досвідчених розробників.

#### Недоліки:

- Відсутність індивідуального підходу, оскільки всі курси доступні тільки у відеоформаті.
- Обмежена тематика курсів: наявність великої кількості матеріалу, але відсутність специфічних тем.
- Складний процес отримання дозволу на викладання.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 11   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

Udemy – платформа, що спеціалізується на наданні навчальних курсів з різних галузей (рис.1.2). Ресурс охоплює широкий спектр тем та має велику кількість користувачів. Веб-сайт пропонує студентам можливість гнучкого навчання, дозволяючи вивчати матеріал у власному темпі.

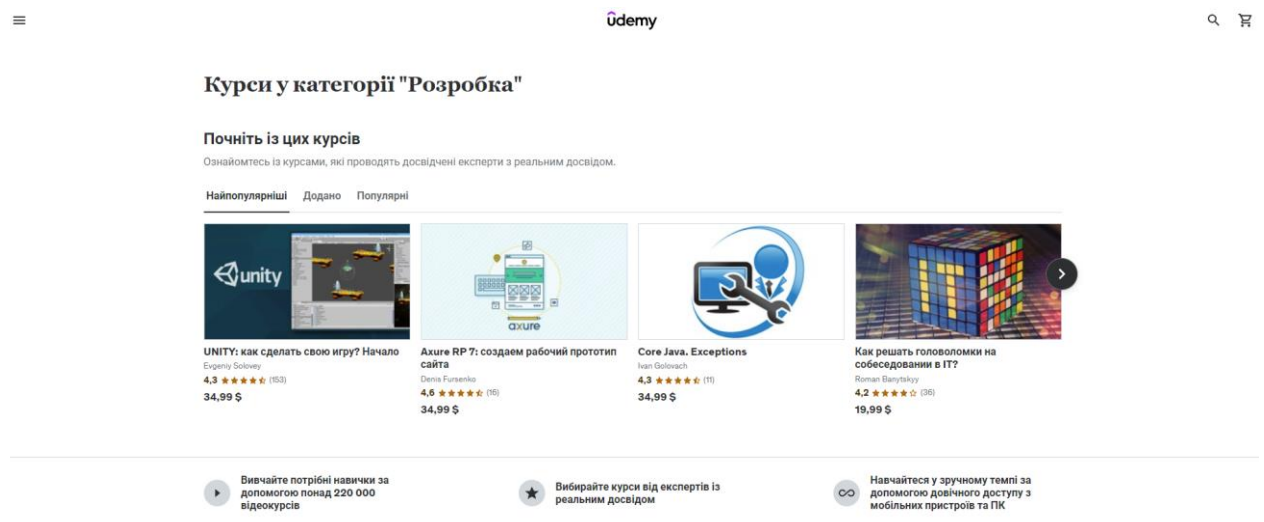


Рис. 1.2. Список курсів Udemy

Переваги:

- Широкий асортимент тем та курсів з різних галузей.
- Різноманітні формати викладання матеріалу: відеоуроки, текстові матеріали, практичні завдання..
- Можливість навчатися у власному темпі.

Недоліки:

- Якість викладання курсів може змінюватися залежно від конкретного викладача та курсу.
- Відсутність прямої комунікації між студентами та викладачами.

EdEra – освітній ресурс, який забезпечує доступ до онлайн-курсів з різних галузей знань (рис. 1.3). Ресурс забезпечує можливість проходити як безкоштовні, так і платні курси, розроблені особисто підібраними адміністрацією викладачами та спеціалістами. Система надає матеріали у вигляді інтерактивних підручників, освітніх спецпроектів та інтегрує сучасні

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 12   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

освітні рішення у навчальний процес. Веб-сайт базується саме на розвитку освіти в Україні.

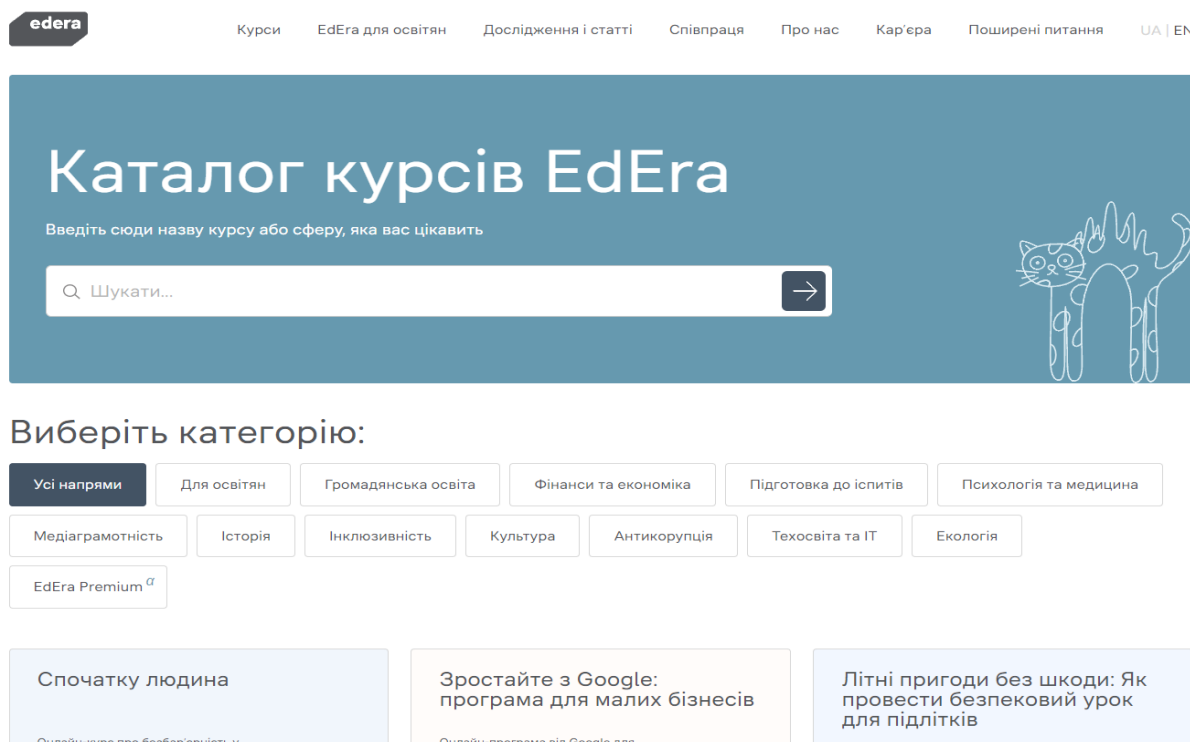


Рис. 1.3. Список курсів EdEra

#### Переваги:

- Різноманітний вибір курсів.
- Ефективна система оцінювання, що дозволяє відстежувати прогрес студентів.
- Велика кількість інтерактивних елементів, таких як вебінари, практичні завдання, онлайн-тести.
- Часті оновлення матеріалів забезпечують їх актуальність і відповідність сучасним тенденціям та вимогам ринку праці.

#### Недоліки:

- Відсутність активної взаємодії з викладачами, при якій студенти отримують оцінки без можливості спілкування.
- Відсутність офіційної акредитації.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 13   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

- Висока вартість курсів, що робить їх недоступними для всіх студентів.

Проаналізувавши існуючі аналоги освітніх платформ для навчання, можна виділити основні критерії та провести відносну оцінку платформ, створивши таблицю (табл. 1.1).

Таблиця 1.1

Порівняльна таблиця аналогів освітніх платформ

|                                    | ITDVN | Udemy | EdEra |
|------------------------------------|-------|-------|-------|
| Зручний інтерфейс                  | +     | +     | +     |
| Можливість отримати роль викладача | -     | -     | +     |
| Доступна ціна                      | +     | +     | -     |
| Вибір власного графіку             | +     | +     | +     |
| Зв'язок студента та викладача      | -     | +     | +     |
| Широкий вибір курсів               | -     | +     | +     |
| Вибір різних методів навчання      | -     | +     | +     |
| Якість викладання                  | +     | -     | -     |

Визначивши основні особливості обраних ресурсів, описавши їх функціонал, сильні та слабкі сторони, є можливість проаналізувати їх у порівнянні один з одним та отримати аспекти, на які варто звернути увагу, ідеї, які варто реалізувати у своєму програмного додатку, які варто покращити або виправити. За допомогою такого підходу, можна бути впевненим, що продукт задовольняє потреби користувачів, має свої сильні сторони та готовий давати конкуренцію вже працюючим системам. В результаті розроблений продукт гарантує користувачам доступ до онлайн-платформи, що надаватиме можливості проходження навчальних матеріалів.

### 1.3. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Онлайн-платформа буде побудована на клієнт-серверній архітектурі. Серверна частина буде представлена у вигляді API та буде написана з використанням фреймворку Express Js. Для клієнтської частини буде використано фреймворк Next Js, який побудований на основі бібліотеки ReactJS. Як сховище буде використовуватись база даних MongoDB та Redis, а для взаємодії з базою даною буде використовуватись Mongoose. Для збереження медіа файлів використано Cloudinary та VdoCipher для шифрування та захисту відео.

Express.js – це мінімалістичний та гнучкий фреймворк для веб-додатків на Node.js, який надає безліч потужних функцій для створення веб-додатків і API. Маючи в своєму розпорядженні безліч допоміжних HTTP-методів та проміжних обробників, створювати надійні API можна легко і швидко. Express забезпечує тонкий прошарок базової функціональності для веб-застосунків, що не спотворює звичну та зручну функціональність Node.js.

Переваги Express.js:

- *Простота у використанні.* Express.js має простий і інтуїтивно зрозумілий API, що дозволяє швидко розпочати роботу навіть новачкам у Node.js.
- *Гнучкість.* Завдяки мідлварам та модульній структурі, Express.js дозволяє налаштовувати додаток під конкретні вимоги та легко інтегрувати сторонні бібліотеки.
- *Висока продуктивність.* Express.js побудований на основі швидкого та асинхронного двигуна Node.js, що забезпечує високу продуктивність і можливість обробляти великий обсяг запитів.
- *Широке співтовариство.* Велике та активне співтовариство розробників забезпечує наявність безлічі навчальних матеріалів, прикладів, плагінів та розширень для Express.js.

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 15   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

- *Сумісність з іншими технологіями.* Express.js легко інтегрується з іншими технологіями та базами даних, такими як MongoDB, PostgreSQL, Redis тощо.

Переваги Express.js для онлайн-платформи проходження курсів:

- *Швидка розробка.* Express.js дозволяє швидко створити прототип вебсайту та реалізувати основні функції, такі як реєстрація користувачів, авторизація, додавання та перегляд курсів.
- *Масштабованість.* Завдяки асинхронній природі Node.js та гнучкості Express.js, можна легко масштабувати додаток для обробки великої кількості одночасних користувачів.
- *Обробка мультимедійних даних.* Express.js дозволяє легко інтегруватися з CDN та серверами для обслуговування мультимедійного контенту, що є важливим для платформи проходження курсів.
- *Підтримка RESTful API.* Express.js спрощує створення RESTful API, що дозволяє легко інтегрувати фронтенд з бекендом та забезпечити взаємодію між компонентами системи.
- *Безпека.* За допомогою мідлварів можна легко додати різні рівні безпеки, такі як захист від CSRF, захист маршрутів та обробку автентифікації.
- *Можливість розширення.* Велика кількість доступних плагінів та модулів дозволяє розширювати функціональність платформи без необхідності писати багато коду з нуля.

Next.js – це потужний фреймворк для створення веб-додатків на базі React, розроблений командою Vercel. Next.js надає широкий набір інструментів для побудови сучасних веб-додатків, включаючи серверний рендеринг, статичну генерацію сторінок, маршрутизацію та багато інших корисних функцій.

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                    | 16   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

### Переваги Next.js:

- *SEO-оптимізація.* Завдяки серверному рендерингу та статичній генерації сторінок, Next.js покращує індексацію сайтів пошуковими системами.
- *Висока продуктивність.* Статична генерація та оптимізація зображень значно покращують швидкість завантаження сторінок, що забезпечує кращий досвід користувачів.
- *Простота розробки.* Автоматична маршрутизація, підтримка CSS та TypeScript, а також API Routes роблять процес розробки значно простішим та швидшим.
- *Масштабованість.* Next.js легко масштабується від простих статичних сайтів до складних динамічних додатків з великою кількістю користувачів.
- *Гнучкість.* Next.js підтримує різні методи рендерингу (SSR, SSG, ISR), що дозволяє вибрати оптимальний підхід для конкретних завдань.

### Переваги Next.js для освітньої платформи онлайн навчання:

- *Поліпшена SEO-оптимізація.* Серверний рендеринг дозволяє індексувати сторінки курсів, покращуючи видимість у пошукових системах та залучаючи більше трафіку.
- *Швидка завантажувальність.* Статична генерація сторінок курсів забезпечує швидке завантаження контенту для користувачів, що покращує загальний досвід перегляду курсів.
- *Динамічне оновлення контенту.* Інкрементальна статична генерація дозволяє динамічно оновлювати сторінки курсів без необхідності повного перезавантаження сайту.
- *Легка інтеграція з базами даних та API.* API Routes в Next.js дозволяють легко створювати серверні функції для роботи з базами даних та зовнішніми API, що є важливим для зберігання та відображення інформації про фільми.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 17   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

- *Можливість використання TypeScript.* Підтримка TypeScript дозволяє створювати більш надійний та стабільний код, зменшуючи кількість потенційних помилок.
- *Оптимізація зображень та мультимедіа.* Вбудована оптимізація зображень покращує завантаження великих зображень курсів та інших мультимедійних файлів, що є ключовим для онлайн-платформи проходження курсів.

MongoDB – це потужна, відкрита система керування базами даних, що спеціалізується на зберіганні неструктурованих даних у форматі документів. Вона здобула популярність завдяки своїй гнучкості, масштабованості та високій продуктивності. MongoDB належить до категорії NoSQL баз даних, що дозволяє зберігати дані у вигляді JSON-подібних документів.

#### Переваги PostgreSQL:

- *Безкоштовність та відкритий код.* PostgreSQL є повністю безкоштовним і відкритим програмним забезпеченням з ліцензією PostgreSQL, яка дозволяє використовувати, модифікувати та поширювати її без будь-яких обмежень.
- *Надійність та стабільність.* PostgreSQL відомий своєю стабільністю та надійністю. Він успішно використовується у великих корпоративних середовищах, де надійність є критично важливою
- *Гнучкість та розширюваність.* Можливість додавання нових функцій, типів даних та індексів робить PostgreSQL надзвичайно гнучким інструментом для вирішення різноманітних завдань.
- *Потужний механізм транзакцій.* Підтримка ACID-транзакцій забезпечує консистентність даних та їх цілісність навіть у складних сценаріях.
- *Широка підтримка спільноти.* Велика спільнота розробників та користувачів забезпечує постійну підтримку та розвиток PostgreSQL, а також доступ до великої кількості ресурсів та документації.

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                    | 18   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

### Переваги MongoDB:

- *Безкоштовність та відкритий код:* MongoDB є безкоштовним програмним забезпеченням з відкритим кодом, що дозволяє використовувати, змінювати та поширювати його без обмежень.
- *Гнучкість даних:* Документна модель MongoDB дозволяє зберігати різноманітні та складні дані, надаючи гнучкість у структурі даних і можливість легко їх змінювати.
- *Висока продуктивність:* Завдяки горизонтальному масштабуванню та підтримці шардінгу, MongoDB може ефективно обробляти великі обсяги даних та високі навантаження.
- *Масштабованість:* MongoDB підтримує горизонтальне масштабування, дозволяючи додавати нові сервери для обробки збільшених обсягів даних та підвищення продуктивності.
- *Інтеграція з іншими інструментами:* Платформа легко інтегрується з різними мовами програмування та інструментами, що робить її універсальним рішенням для розробників.
- *Глобальна підтримка спільноти:* Величезна спільнота розробників та користувачів MongoDB забезпечує постійну підтримку та розвиток, а також доступ до великої кількості ресурсів та документації.

### Переваги MongoDB для онлайн-платформи проходження курсів:

- *Ефективне зберігання метаданих:* MongoDB дозволяє ефективно зберігати різноманітні дані про курси, матеріали, користувачів у форматі документів, що сприяє зручному доступу та управлінню інформацією.
- *Гнучкий пошук:* Вбудовані можливості повнотекстового пошуку дозволяють швидко знаходити інформацію про курси та матеріали за різними критеріями, що сприяє полегшенню навігації та зручності використання платформи користувачами.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 19   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

- *Підтримка різних типів даних:* MongoDB підтримує роботу з масивами, вкладеними документами та іншими складними структурами даних, що дозволяє зберігати та обробляти різноманітні навчальні матеріали, включаючи відео, тексти та інші формати.
- *Висока доступність:* Завдяки можливості реплікації та автоматичного фейловеру, MongoDB забезпечує стабільну роботу платформи з низькими ризиками втрати даних та збоїв в роботі, що є важливим для неперервної доступності освітніх ресурсів.
- *Масштабованість та продуктивність:* Можливість горизонтального масштабування дозволяє MongoDB ефективно обробляти великі обсяги даних та високі навантаження, що забезпечує швидку відповідь платформи на запити користувачів та оптимізує її продуктивність.
- *Аналіз даних:* За допомогою вбудованих агрегаційних фреймворків MongoDB можна проводити складний аналіз навчальних даних, включаючи вивчення активності користувачів, оцінювання ефективності курсів та їх популярність.
- *Безпека:* MongoDB надає потужні засоби контролю доступу, шифрування та управління правами користувачів, що забезпечує високий рівень захисту конфіденційної інформації на освітній платформі.

Mongoose – це моделюючий об'єктно-документний (Object-Document Mapping) інструмент для MongoDB, який спрощує взаємодію з базою даних MongoDB у веб-додатках. Mongoose дозволяє розробникам описувати схеми документів у форматі JavaScript, забезпечуючи зручний інтерфейс для створення, зберігання, оновлення та операцій з даними.

Переваги Mongoose:

- *Простота використання:* Інтуїтивно зрозуміле API Mongoose спрощує розробку шляхом дозволу визначати схеми документів та виконувати операції з базою даних MongoDB у звичний для JavaScript спосіб.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 20   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

- *Валідація даних:* Mongoose надає можливість валідувати дані перед збереженням у базі, що зменшує ймовірність помилок і забезпечує консистентність даних.
- *Зручна робота з реляціями:* Завдяки можливості вбудовувати документи та визначати посилання між колекціями, Mongoose дозволяє зручно працювати з складними структурами даних та взаємозв'язками.
- *Розширення функціональності:* Mongoose підтримує плагіни, що дозволяє розширювати функціональність та використовувати додаткові модулі для роботи з даними.

Переваги Mongoose для онлайн-платформи проходження курсів:

- *Простота і швидкість розробки:* Використання Mongoose дозволяє швидко створювати та керувати моделями даних для курсів та інших атрибутів платформи.
- *Валідація та типізація:* Можливість використовувати схеми Mongoose дозволяє валідувати дані та використовувати типізацію для забезпечення правильності та безпеки операцій з даними.
- *Масштабованість:* Mongoose дозволяє легко масштабувати додаток та адаптувати схему даних до зростаючих потреб проєкту, забезпечуючи високу продуктивність та ефективність.

Висновки до першого розділу

У першому розділі випускної кваліфікаційної роботи визначено основні вимоги до програмного продукту та чітко окреслено задачі, які повинні бути виконані при розробці додатку. Визначено мету та особливості розробки онлайн-платформи проходження курсів.

Проведено розширений аналіз аналогічних рішень реалізації поставленої задачі. У процесі обрано такі веб-сервіси: ITDVN, Udemu, EdEra. В результаті проведеного аналізу та порівнянь, визначено основні характеристики систем, сильні та слабкі сторони. Сформовано бачення

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 21   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

конкурентоспроможного рішення для поставленої задачі з урахуванням особливостей аналогів для реалізації переваг та виправлення, покращення недоліків.

Для передачі даних було обрано клієнт-серверну архітектурну модель RESTful API. Дані рішення дозволять ефективно розробити онлайн-платформу, забезпечивши стабільну структуру з можливістю розширення.

Після обрання архітектури було проведено дослідницьку роботу інструментів розробки та здійснено обґрунтований вибір середовища та технологій, які будуть використані про роботі з проєктом. Обрано такі технології: Express.js, Next.js, MongoDB, Mongoose.

Отримано чітко сформоване бачення онлайн-платформи проходження курсів, визначено основні вимоги, інструменти та структуру проєкту для його успішної реалізації.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 22   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

## РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПЛАТФОРМИ

### 2.1. Визначення варіантів використання та об'єктно-орієнтованої структури

У цьому розділі представлено аналіз вимог до онлайн-платформи проходження курсів з акцентом на виявленні функціональних та нефункціональних вимог. Дана процедура дозволяє визначити конкретні функції та можливості, які повинен мати додаток для забезпечення потреб користувачів. Також буде описано варіанти використання системи для вимог до освітньої платформи для різних ролей клієнтів системи. З метою отримати бачення структури системи буде сформовано діаграму сутностей, їх структури та методи. Визначення вимог відбувається на базі досліджень потреб та бажань користувачів щодо роботи з подібними додатками. Метою даних досліджень є визначення основних функцій для забезпечення виконання потреб користувачів створюваної системи.

#### **Бізнес вимоги:**

- основні цілі: проєкт створюється з метою забезпечення можливостей проходження курсів та опрацювання навчальних матеріалів, а також можливості їх створення. В освітній платформі передбачені можливості пошуку курсів, купівлі курсів, функціонал відгуків до курсів,
- представлення проєкту: проєкт буде реалізовано у вигляді веб-сайту, що реалізує поставлені цілі,
- розширення: проєкт повинен створюватися з розумінням подальшого розвитку,
- оплата: можливість придбати курс.

#### **Функціональні вимоги**

- авторизація користувачів у системі: в проєкті повинна бути представлена реєстрації користувача,

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 23   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

- можливість збереження інформації: система повинна зберігати інформацію та завантаженні файли використовуючи хмарні технології,
- можливість створювати та управляти курсами: можливість створювати курси, публікувати їх, наповнювати матеріалом,
- можливість безпечно проводити фінансові операції: система повинна коректно обробляти запити, що пов'язані із фінансами,

### **Нефункціональні вимоги:**

#### **А. Сприйняття:**

- а) Час потрібний для вивчення інструментів роботи з платформою для звичайних користувачів – 15-25 хв, а для досвідчених – не більше 10 хв.

#### **В. Надійність:**

- а) Доступність – час, потрібний для обслуговування платформи не повинен перевищувати 10% від загального часу роботи.
- б) Середній час безперервної роботи - 61 робочий день.
- с) Максимальна норма помилок та дефектів в роботі платформи – 1 помилка на 1000 запитів користувача.

#### **С. Продуктивність:**

- а) Система повинна підтримувати мінімум 500 одночасно працюючих користувачів, пов'язаних з спільною базою даних.

#### **Д. Можливість експлуатації:**

- а) Масштабування – система повинна мати можливість збільшувати потужності зі збільшенням користувачів таким чином, щоб це не завадило працювати платформі у тому ж ритмі.
- б) Оновлення версій – оновлення повинно відбуватися в час найменшого навантаження на платформу із зупинкою роботи сайту при завантаженні оновлення.

#### **Е. Захист даних:**

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 24   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

- а) Всі запити, що стосуються авторизованих користувачів повинні відбуватися з обов'язковим захешованим токеном.

**Часові характеристики реакції системи:**

- час відгуку: обробка запитів та надання відповіді повинні відбуватися з не більше 5 с,
- час відновлення: час відновлення системи до працюючого стану за умови, що не було критичних помилок, не повинен перевищувати 1 год,
- час доставки файлів: час, за який система зберігає файли великого розміру повинен бути в рамках 3-15 с в залежності від розміру файлу.
- час реакцій на події та завантаження сторінки: події роботи користувача на сторінці, а також їх завантаження повинні відбуватися моментально для користувача.

Часові характеристики визначаються за допомогою виконання навантажувального тестування, вивчення стандартів роботи веб-сайтів. Дані показники можуть змінюватися в залежності від швидкості інтернет з'єднання користувача.

Відповідно до визначених вимог та особливостей можливе будування діаграми варіантів використання (Use Cases Diagram) для онлайн-платформи проходження курсів.

Діаграма варіантів використання описує функціональне призначення системи, ті функції, які вона повинна виконувати. Вона створюється з метою визначити загальні межі і предметну область системи, сформулювати загальні вимоги до функціональної поведінки, розробити початкову концептуальну модель проєкту, її подальшої деталізації у формі логічних та фізичних моделей, визначити відношення між акторами і прецедентами у системі.

В результаті було отримано діаграму використання (рис. 2.1).

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 25   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

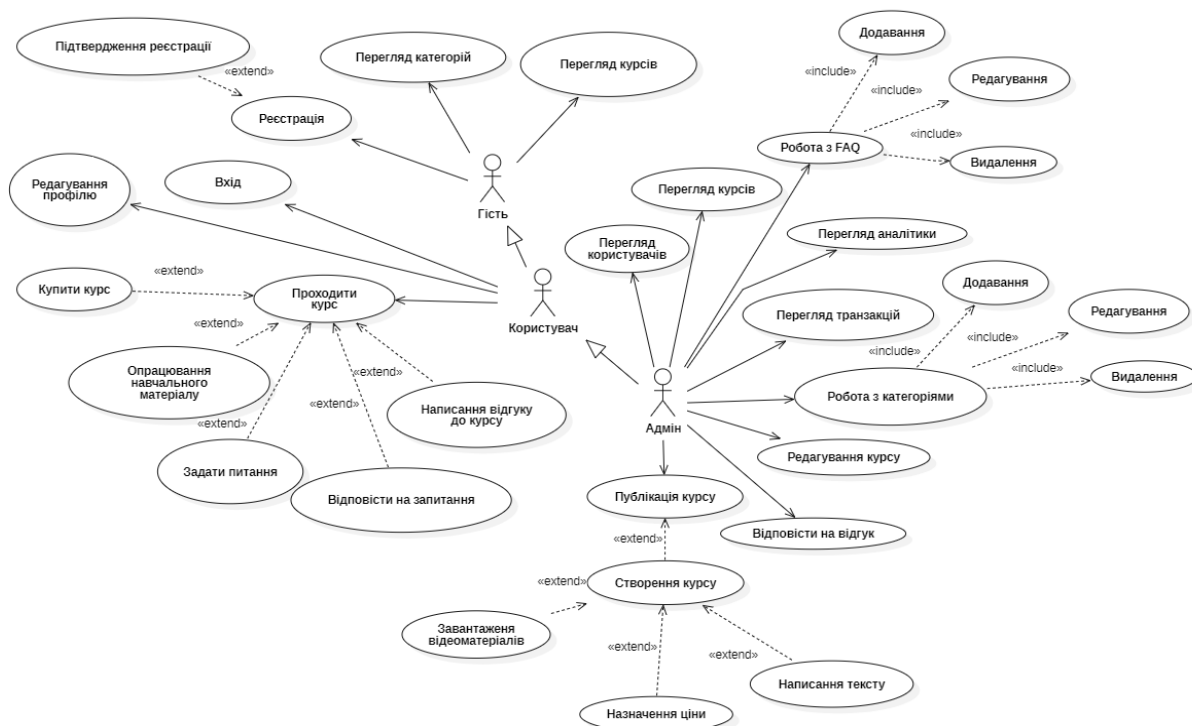


Рис. 2.1. Діаграма варіантів використання

#### Гість:

- перегляд курсів: можливість перегляду списку курсів, його опис.
- перегляд категорій: можливість перегляду категорій та фільтрації курсів по ним.
- можливість реєструватися.

#### Користувач:

- можливість входу: можливість вийти у систему.
- можливість працювати з обліковим записом: функціонал редагування даних профілю.
- можливість переглядати доступні курси: можливість переглядати безкоштовні курси та курси, які користувач купив.
- проходити курс: можливість проходження доступних курсів.
- купити курс: можливість купити курс онлайн.
- задати питання: можливість задати питання під доступним курсом.

- відповісти на запитання: можливість відповісти комусь на його запитання під курсом.
- написання відгуку: можливість написати відгук та оцінити доступний курс.

#### **Адміністратор:**

- можливість роботи з курсами: функціонал створення, редагування, публікації курсів, створення, редагування, публікації уроків та матеріалу до них.
- можливість переглядати всі курси.
- можливість переглядати всіх зареєстрованих користувачів.
- можливість переглядати всі транзакції.
- можливість відповісти на відгук до курсу.
- можливість управління категоріями: створення, редагування, видалення категорій.
- можливість управління FAQ: створення, редагування, видалення FAQ.
- можливість редагування банера.

На наступному етапі проведемо розробку структури бази даних.

### **2.2. Розробка структури бази даних системи**

Розробка структури бази даних є важливою складовою при розробці програмного додатку, так як саме вона забезпечує зберігання та організацію даних, їх цілісність. Добре спроектована БД може забезпечити швидкий та ефективний доступ до потрібних даних.

Правильно розроблена структура дозволяє зручно впроваджувати нові функції та розширювати функціональність системи. Вона також полегшує підтримку в обслуговуванні системи, забезпечуючи зручність виконання змін та операцій обслуговування бази даних.

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                    | 27   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

Для забезпечення всіх вимог було обрано нереляційний тип баз даних та СУБД MongoDB. Дана система надає функціонал, використання якого забезпечить всі поставлені вимоги щодо управління даними.

В результаті було розроблено структуру бази даних, яка відповідає всім поставленим задачам та вимогам (рис. 2.2).

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div>courses</div> <div> <div>_id</div> <div>objectId</div> <div>NN</div> </div> <div> <div>name</div> <div>string</div> <div>NN</div> </div> <div> <div>description</div> <div>string</div> <div>NN</div> </div> <div> <div>categories</div> <div>string</div> <div>NN</div> </div> <div> <div>price</div> <div>double</div> <div>NN</div> </div> <div> <div>estimatedPrice</div> <div>any</div> <div></div> </div> <div> <div>thumbnail {}</div> <div>object</div> <div>NN</div> </div> <div> <div>tags</div> <div>string</div> <div>NN</div> </div> <div> <div>level</div> <div>string</div> <div>NN</div> </div> <div> <div>demoUrl</div> <div>string</div> <div>NN</div> </div> <div> <div>benefits[{}]</div> <div>object</div> <div>NN</div> </div> <div> <div>prerequisites[{}]</div> <div>object</div> <div>NN</div> </div> <div> <div>courseData[{}]</div> <div>object</div> <div>NN</div> </div> <div> <div>ratings</div> <div>double</div> <div>NN</div> </div> <div> <div>purchased</div> <div>double</div> <div>NN</div> </div> <div> <div>reviews[{}]</div> <div>object</div> <div>NN</div> </div> <div> <div>lessons</div> <div>object</div> <div>NN</div> </div> <div> <div>certificates</div> <div>object</div> <div>NN</div> </div> <div> <div>feedbacks</div> <div>object</div> <div>NN</div> </div> <div> <div>createdAt</div> <div>date</div> <div>NN</div> </div> <div> <div>updatedAt</div> <div>date</div> <div>NN</div> </div> <div> <div>_v</div> <div>double</div> <div>NN</div> </div> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Рис. 2.2. Розроблена структура бази даних освітньої платформи

Кожну сутність було спроектовано окремо для забезпечення коректної роботи кожного елемента.

Для збереження даних про користувача була створена таблиця “users” (табл. 2.1):

Таблиця 2.1

Структура таблиці «users»

| Поле  | Тип               | Призначення            |
|-------|-------------------|------------------------|
| _id   | objectId (string) | Ідентифікатор сутності |
| name  | string            | Ім'я                   |
| email | string            | Пошта користувача      |

## Продовження таблиці 2.1

| Поле       | Тип    | Призначення               |
|------------|--------|---------------------------|
| password   | string | Пароль                    |
| role       | string | Роль                      |
| isVerified | bool   | Статус користувача        |
| courses    | object | Курси користувача         |
| createdAt  | date   | Дата створення            |
| updatedAt  | date   | Дата останнього оновлення |
| avatar     | object | Фото користувача          |

Для опису сутності курсу, його даних було розроблено таблицю «courses» (табл.2.2):

Таблиця 2.2

## Структура таблиці «courses»

| Поле           | Тип               | Призначення                |
|----------------|-------------------|----------------------------|
| _id            | objectId (string) | Ідентифікатор сутності     |
| name           | string            | Назва курсу                |
| description    | string            | Опис курсу                 |
| categories     | string            | Категорії курсу            |
| price          | number            | Ціна курсу                 |
| estimatedPrice | number            | Оцінкова ціна курсу        |
| thumbnail      | object            | Зображення мініатюри курсу |
| tags           | string            | Теги курсу                 |
| level          | string            | Рівень курсу               |
| demoUrl        | string            | URL демонстраційного відео |
| benefits       | array             | Переваги курсу             |
| prerequisites  | array             | Вимоги до курсу            |
| courseData     | array             | Дані курсу                 |
| ratings        | number            | Рейтинг курсу              |
| purchased      | number            | Кількість покупок курсу    |
| reviews        | array             | Відгуки про курс           |

## Продовження таблиці 2.2

| Поле         | Тип   | Призначення                     |
|--------------|-------|---------------------------------|
| lessons      | array | Уроки курсу                     |
| certificates | array | Сертифікати курсу               |
| feedbacks    | array | Зворотний зв'язок щодо курсу    |
| createdAt    | date  | Дата створення курсу            |
| updatedAt    | date  | Дата останнього оновлення курсу |

Для контролю доступу користувача до курсів було створено сутність замовлень, які описані у таблиці «orders» (табл.2.3):

Таблиця 2.3

## Структура таблиці «orders»

| Поле         | Тип               | Призначення               |
|--------------|-------------------|---------------------------|
| _id          | objectId (string) | Ідентифікатор замовлення  |
| courseId     | string            | Ідентифікатор курсу       |
| userId       | string            | Ідентифікатор користувача |
| payment_info | object            | Інформація про оплату     |
| createdAt    | date              | Дата створення            |
| updatedAt    | date              | Дата останнього оновлення |

Для можливості адміністратору отримувати сповіщення про активні дії користувача, було створено сутність сповіщень, які описані в таблиці «notifications» (табл.2.4).

Таблиця 2.4

## Структура таблиці «notifications»

| Поле      | Тип               | Призначення               |
|-----------|-------------------|---------------------------|
| _id       | objectId (string) | Ідентифікатор сутності    |
| title     | string            | Заголовок повідомлення    |
| message   | string            | Текст повідомлення        |
| status    | string            | Статус повідомлення       |
| createdAt | date              | Дата створення            |
| updatedAt | date              | Дата останнього оновлення |

Для можливості зміни категорій, банера, часто заданих питань і відгуків на сайті, було створено сутність макету, які описані у таблиці «layouts», (табл.2.5).

Таблиця 2.5

Структура таблиці «layouts»

| Поле       | Тип               | Призначення            |
|------------|-------------------|------------------------|
| _id        | objectId (string) | Ідентифікатор сутності |
| type       | string            | Тип макета             |
| faq        | array             | Поширені запитання     |
| categories | array             | Список категорій       |
| banner     | object            | Банер                  |
| reviews    | array             | Відгуки                |

В результаті розробки бази даних було отримано структуру, яка дозволить ефективно управляти даними, потенціал масштабування та розвитку. Спроектowana база даних має надзвичайно важливу роль при розробці програмного продукту.

### 2.3. Проектування та реалізація алгоритмів роботи системи

Проектування алгоритмів системи дозволяє створювати логічні послідовності дій і процесів, які система повинна виконувати для досягнення своїх функціональних цілей. Під час проектування були створені діаграми активності та послідовності, які наочно показують дії, прийняття рішень та взаємодію між об'єктами під час роботи з системою.

Діаграма активності реєстрації та входу в систему (рис. 2.3) наочно демонструє процес реєстрації та отримання ролі користувача. Під час реєстрації користувач повинен вводити коректні дані для успішного проходження алгоритму.

Після реєстрації користувач отримує на електронну пошту лист з кодом для активації акаунта. Цей крок необхідний для забезпечення того, що

системою користуватимуться лише реальні люди та для запобігання атакам з боку злоумисників. Ввівши код, акаунт активується та перенаправляє користувача за форму, де він вже вводить дані для входу на онлайн-платформу.

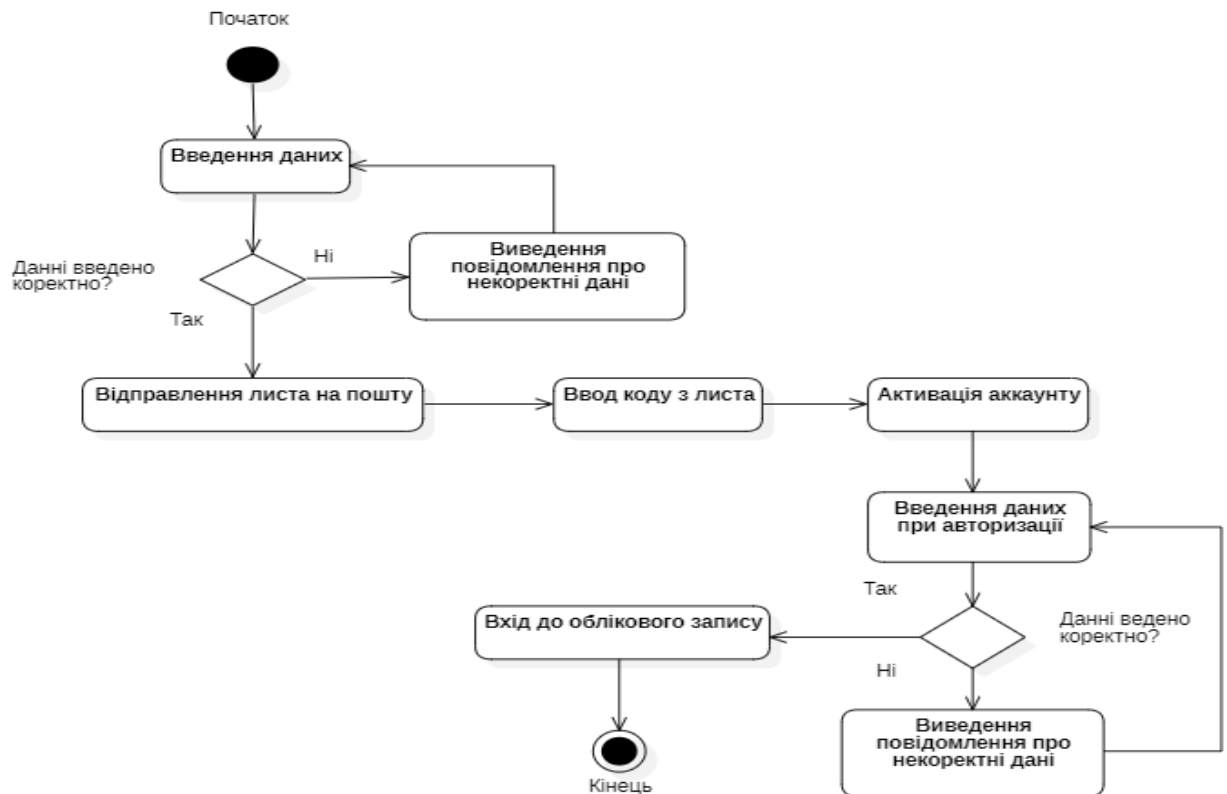


Рис. 2.3. Діаграма реєстрація та авторизації

Ввівши код активації, акаунт активується, процес відбувається за допомогою генерації коду, який дійсний 5 хвилин, що забезпечує те, що саме той користувач, що зареєструвався, отримає можливість активувати акаунт. При авторизації також відбувається перевірка на правильно введені дані.

Основним об'єктом системи є курс. Цикл його життя складається з багатьох етапів та включає в себе участь кожної ролі, прописаної у системі. Діаграма активності циклу життя курсу (рис. 2.4) демонструє основні етапи публікації та вивчення матеріалів.

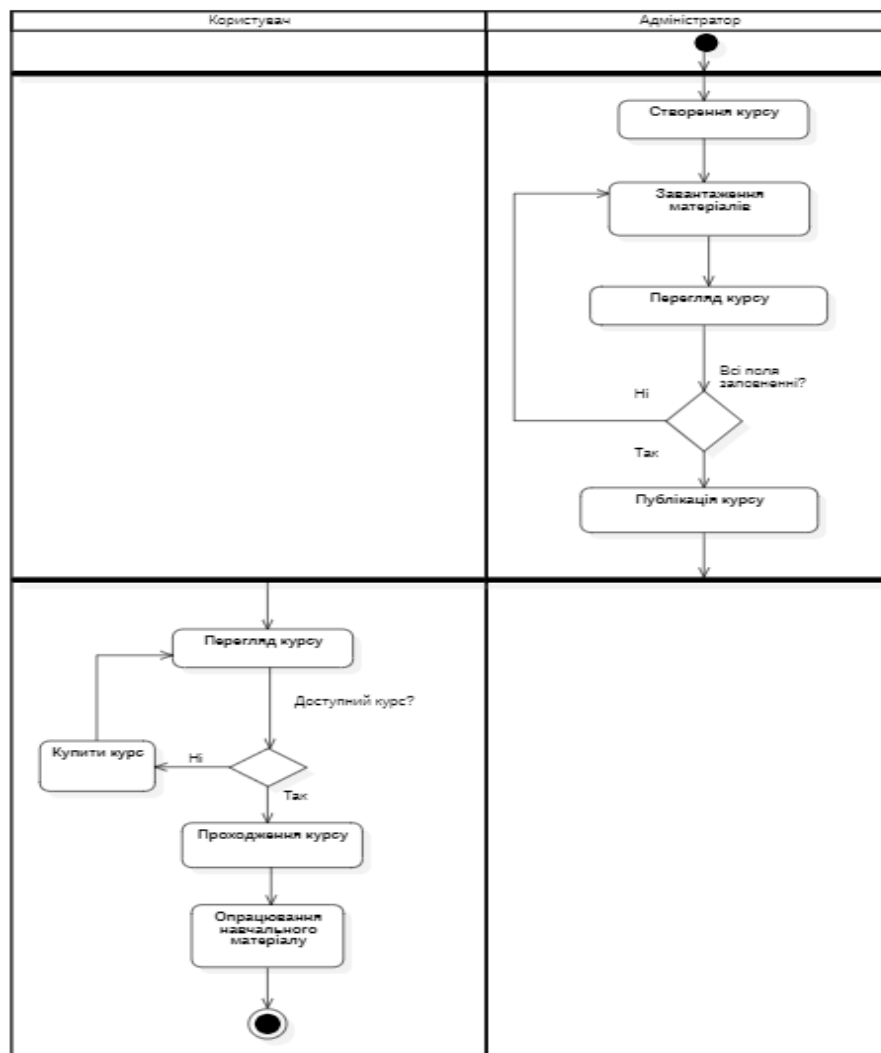


Рис. 2.4. Діаграма активності циклу життя курсу

На діаграмі зображено процес створення та редагування курсу. Під час завантаження матеріалів адміністратор створює уроки та додає до них інформацію різних типів. Адміністратор перед публікацією переглядає як виглядає сторінка курсу і публікує його.

З боку користувача він може переглядати деталі курсу. Для доступу до матеріалів уроків користувач має зареєструватися на курс або купити його. Також є функціонал для відгуків про курс у вигляді коментарів, які можуть залишати користувачі.

Ця діаграма в цілому описує функціональні особливості, пов'язані з курсами та їх зв'язками.

Для купівлі курсу повинна бути достатня сума на банківській карті. Користувач повинен натиснути на кнопку купити курс, після чого відкриється форма створення транзакції за допомогою Stripe, для введення даних карти. Після успішної оплати курс буде доданий до доступних курсів користувача. Послідовність процесу купівлі курсу ілюструється на діаграмі активності (рис. 2.5).

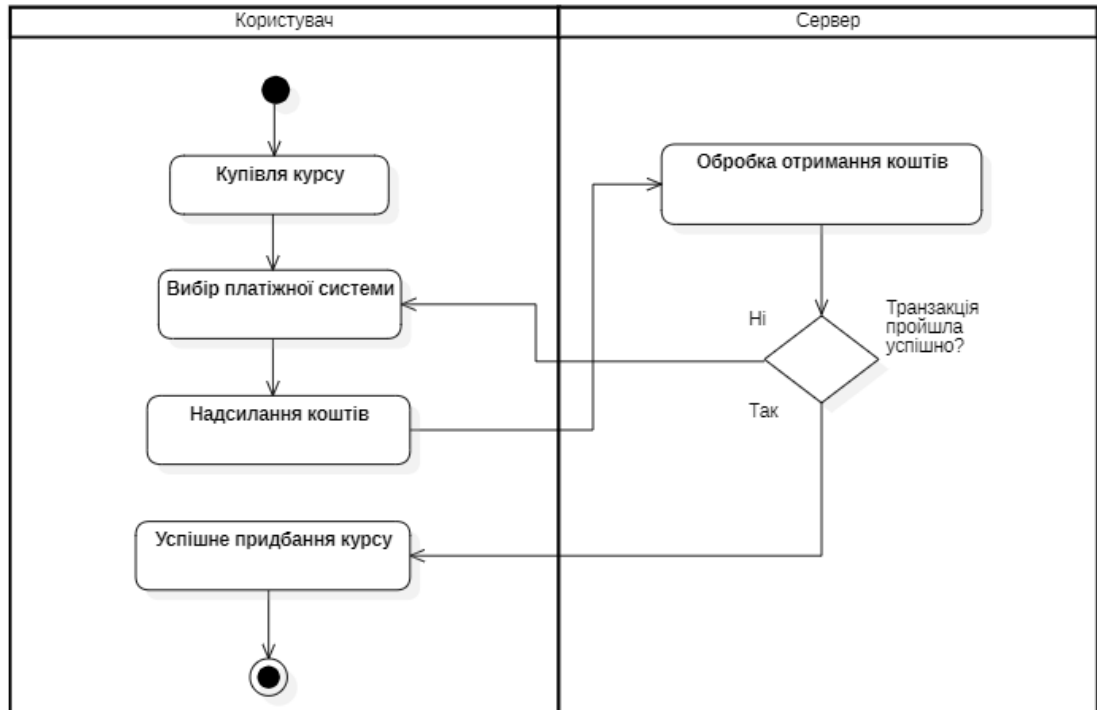


Рис. 2.5. Діаграма активності купівлі курсу

Діаграми демонструють наочне представлення алгоритмів дій купівлі курсу. Створення таких діаграм забезпечує чітку розуміння структури роботи проєкту, продумати різні сценарії та прорахувати всі можливі ситуації.

Опис алгоритмів роботи системи дозволяє виділити ключові кроки, умови, цикли та залежності розробляемого додатку. Ефективність та безперебійність системи напряму залежить від правильно розроблених алгоритмів та кроків її роботи.

## 2.4. Реалізація функціоналу онлайн-платформи проходження курсів

Розробка онлайн-платформи складається з реалізації ключових блоків функціоналу системи. У даному розділі будуть продемонстровані програмні рішення для виконання ключових задач із реалізації проєкту.

Для реалізації збереження файлів курсів було обрано використати хмарне сховище Cloudinary та VdoCipher. Такий підхід захищає від втрати у випадку проблем із сервером. Хмарні технології дозволяють легко збільшувати обсяг зберігання даних за потребою.

Розглянемо основні етапи створення фільму на освітній платформі. Існує чотири етапи. Для цього адміністратор повинен зайти в адмін панель та заповнити інформацію про курс.

Перший етап - інформація про курс. Адміністратор заповнює поля, що містять інформацію про назву курсу, опис, вартість, орієнтовну ціну, теги, категорія, рівень курсу, демонстративне відео, зображення курсу.

Другий етап – параметри курсу. Адміністратор вказує переваги та вимоги, щоб успішно пройти курс.

Третій етап – зміст курсу. Адміністратор заповнює назву розділу, назву відео, URL-адрес відео, а саме вказує ідентифікатор раніше завантаженого відео на платформу VdoCipher, тривалість відео, опис відео, посилання.

Четвертий етап – перегляд курсу. Адміністратор переглядає раніше заповнену інформацію і якщо все гаразд, публікує курс.

Після публікації, курс стає доступним на освітній платформі для його проходження.

```
export const uploadCourse = asyncErrorHandler(  
  async (req: Request, res: Response, next: NextFunction) => {  
    try {  
      const data = req.body;  
      const thumbnail = data.thumbnail;  
      if (thumbnail) {  
        const myCloud = await cloudinary.v2.uploader.upload(thumbnail, {  
          folder: 'courses',
```

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                    | 35   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

```

});
data.thumbnail = {
  public_id: myCloud.public_id,
  url: myCloud.secure_url,
};
}
createCourse(data, res, next);
} catch (error: any) {
  return next(new ErrorHandler(error.message, 500));
}
}
);

```

Після створення адміністратором курсу, йде генерація ОТР токена, для авторизації відтворення відео.

```

export const generateVideoUrl = asyncErrorHandler(
  async (req: Request, res: Response, next: NextFunction) => {
    try {
      const { videoid } = req.body;
      const response = await axios.post(
        `https://dev.vdocipher.com/api/videos/${videoid}/otp`,
        { ttl: 300 },
        {
          headers: {
            Accept: 'application/json',
            'Content-Type': 'application/json',
            Authorization: `Apisecret ${process.env.VDOCIPHER_API_SECRET}`,
          },
        },
      );
      res.json(response.data);
    } catch (error: any) {
      return next(new ErrorHandler(error.message, 400));
    }
  }
);

```

Для роботи із зовнішнім сервісом Stripe було створено метод інформацією про замовлення.

```

export const newPayment = asyncErrorHandler(
  async (req: Request, res: Response, next: NextFunction) => {

```

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 36   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

```

try {
  const myPayment = await stripe.paymentIntents.create({
    amount: req.body.amount,
    currency: 'UAH',
    metadata: {
      company: 'E-Learning',
    },
    automatic_payment_methods: {
      enabled: true,
    },
  });
  res.status(201).json({
    success: true,
    client_secret: myPayment.client_secret,
  });
} catch (error: any) {
  return next(new ErrorHandler(error.message, 500));
}
}
);

```

## Висновки до другого розділу

У другому розділі кваліфікаційної роботи було спроектовано та реалізовано функціональну структуру проєкту.

Спочатку було проведено аналіз раніше поставлених задач та визначено основні функціональні та нефункціональні вимоги до додатку та його зв'язку з користувачами. Було проведено аналіз ролей для користувачів та визначено основні ролі: Гість, Користувач, Адміністратор. Згідно до цих даних було реалізовано діаграму використання, яка демонструвала опис зв'язку користувачів з системою.

Далі було приділено увагу створенню бази даних, адже це є одним із найголовніших етапів розробки освітньої платформи. Використовуючи СУБД MongoDB було спроектовано та реалізовано БД, яка відповідає всім поставленим вимогам та забезпечує ефективну роботу з даними. Також було

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                    | 37   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

описано кожен сутність, її значення та роль у системі, зв'язки та залежності між об'єктами.

Після пророблених дій було проведено роботу по визначенню основних алгоритмів роботи системи. Було створено діаграми активності. Вони наочно показують функціональну частину взаємодії продукту з користувачем. В результаті було реалізовано функціональну частину додатку згідно до раніше поставлених завдань, вимог та спроектованої архітектури.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 38   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

## РОЗДІЛ 3. ІНСТРУКЦІЯ КОРИСТУВАННЯ ТА РОЗГОРТАННЯ СИСТЕМИ. РЕЗУЛЬТАТИ ТЕСТУВАННЯ

### 3.1. Особливості щодо встановлення та розгортання

Для встановлення додатку та успішної роботи з ним потрібно, щоб система відповідала вимогам та покроково виконати дії із налаштування та встановлення.

#### Системні вимоги до середовища розгортання:

- для оптимальної роботи серверу потрібно обрати потужний процесор з великою кількістю ядер. Мінімальне значення – 4.
- Для зберігання файлових даних потрібно мати достатньо місця на сервері.
- Сервер повинен мати стабільне підключення до Інтернет ресурсів.
- Обсяг оперативної пам'яті не повинен бути менше 8 ГБ.

#### Порядок розгортання додатку:

- Повинна бути встановлена Unix подібна операційна система, бажано Ubuntu останньої версії (Ubuntu 24.04 LTS).
- Потрібно встановити Node.js.
- Далі потрібно завантажити файли проєкту з репозиторію, налаштувати ключі для поштового сервісу SMTP, гугл авторизації OAuth.
- Потрібно створити акаунт на Cloudinary, створити файлове сховище та додати ключі до конфігураційного файлу.
- Потрібно створити акаунт на VdoCipher, створити файлове сховище та додати ключі до конфігураційного файлу.
- Потрібно створити акаунт на Stripe, створити там проєкт та отримати ключі, які потрібно занести до конфігураційного файлу.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 39   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

Після виконання всіх дій в результаті буде отримано готовий до експлуатації програмний продукт. Для перевірки потрібно в консолі перейти в папку «client» та ввести команду «npm run dev», після цього перейти в консолі у папку «server» та ввести команду «npm run dev» та переконатися чи працюють всі компоненти стабільно.

### 3.2. Інтерфейс та порядок роботи

У цьому розділі буде продемонстровано інтерфейс і роботу з системою. До кожної дії буде надано опис і додано відповідну фото-демонстрацію. При відкритті веб-сайту на головній сторінці (рис. 3.1) відображається банер з текстом, кнопка для переходу до курсів та список із чотирьох останніх доданих курсів.

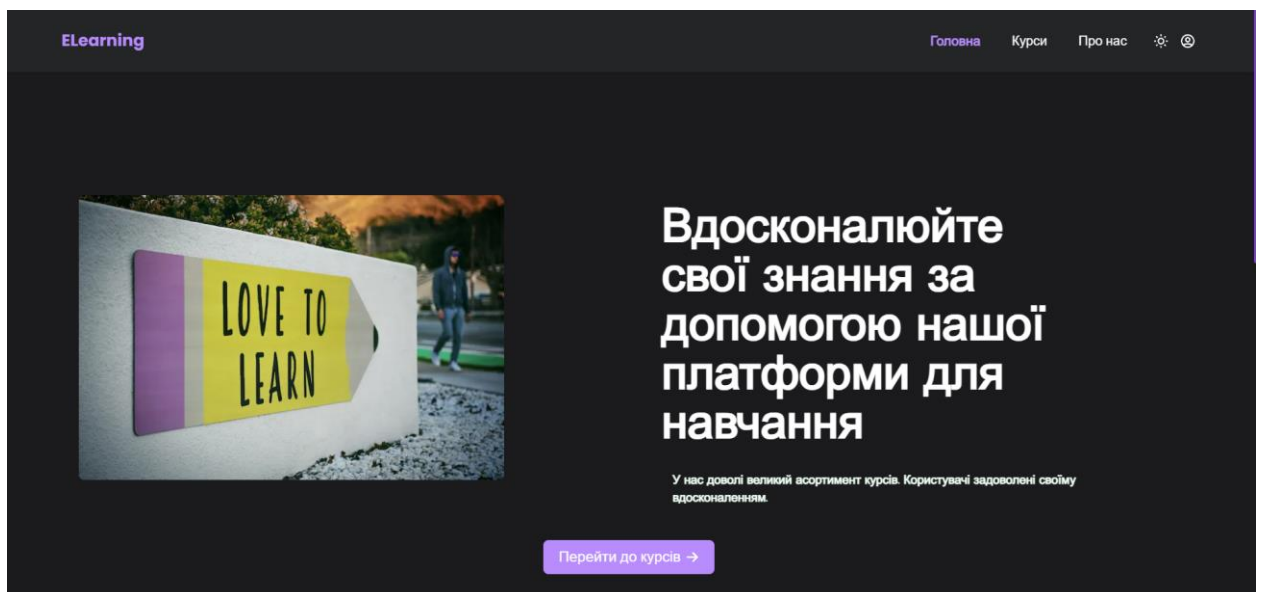


Рис. 3.1. Головна сторінка

Користувач має змогу переглянути деталі курсів натиснувши на відповідний курс (рис. 3.2). В деталях можна побачити основну інформацію, уроки, рейтинг та відгуки.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 40   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

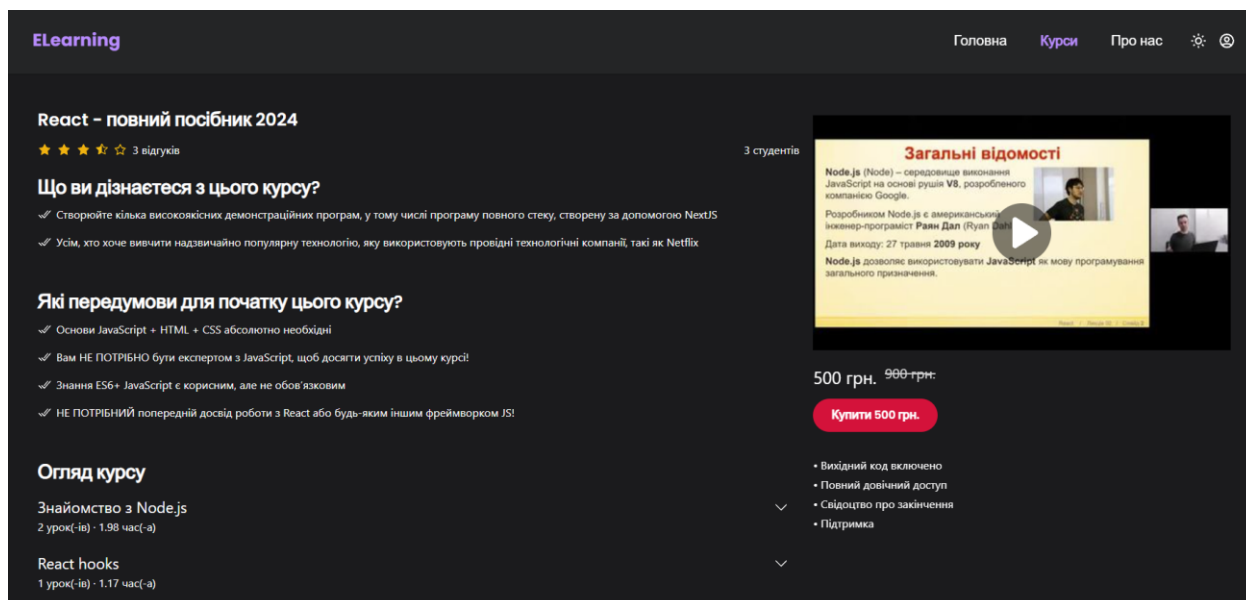


Рис. 3.2. Перегляд курсу

В модальному вікні реєстрації можна побачити форму для введення даних. Вона складається з таких полів: ім'я та прізвище, пошта, пароль (рис. 3.3).

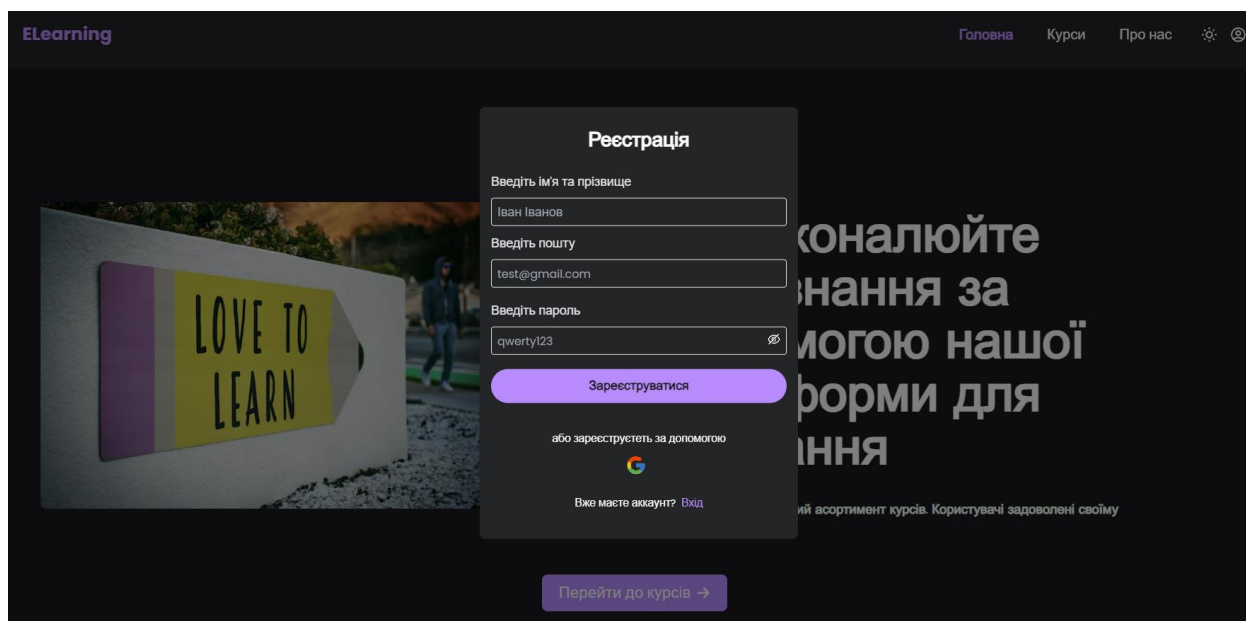


Рис. 3.3. Модальне вікно реєстрації

Після реєстрації користувач отримає повідомлення на вказану пошту (рис. 3.4). У повідомленні буде код для активації облікового запису. Без активації профілю авторизуватися неможливо.

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 41   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

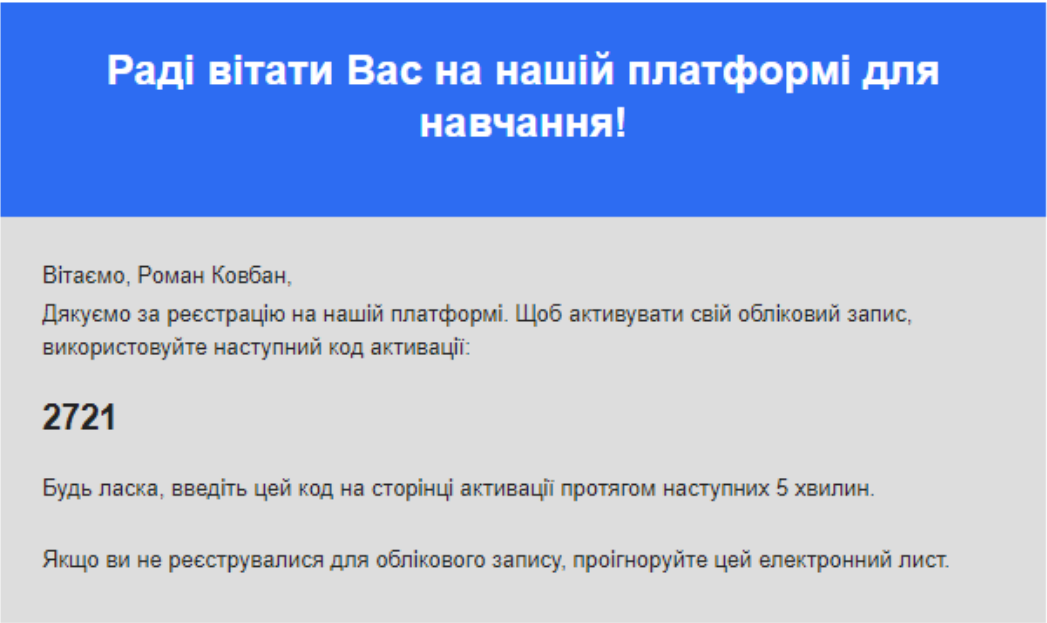


Рис. 3.4. Лист на пошті

Після отримання коду на пошту, вводимо його у форму для активації акаунту (рис. 3.5).

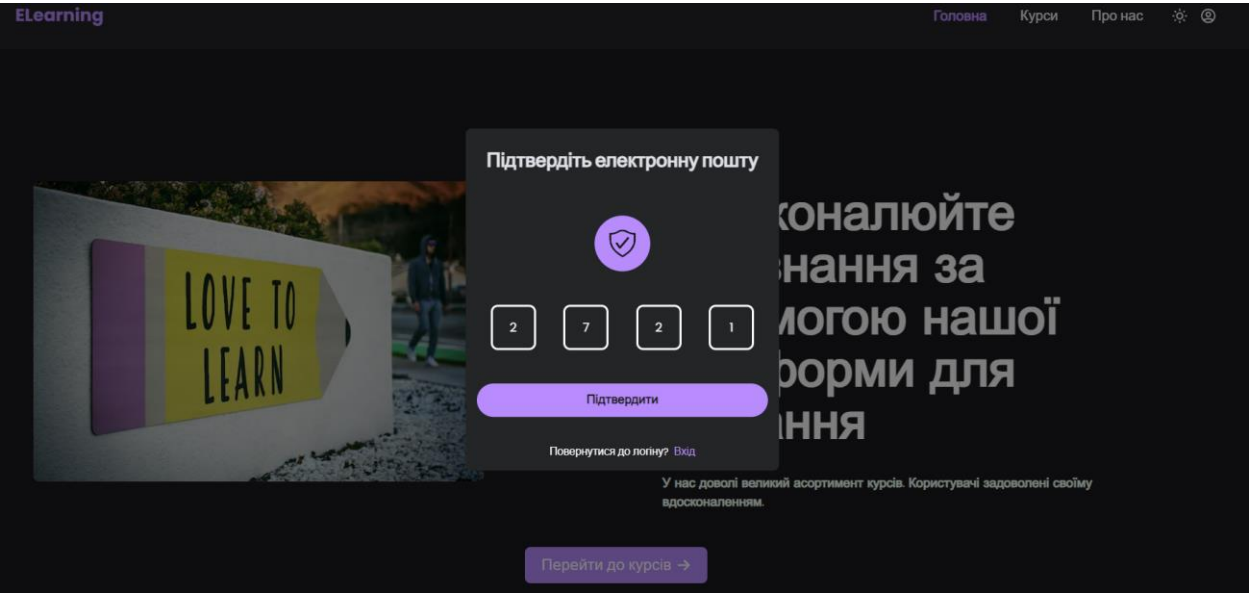


Рис. 3.5. Модальне вікно для вводу коду

Після активації акаунту можна авторизуватися до системи ввівши у форму авторизації свої дані (рис. 3.6).

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 42   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

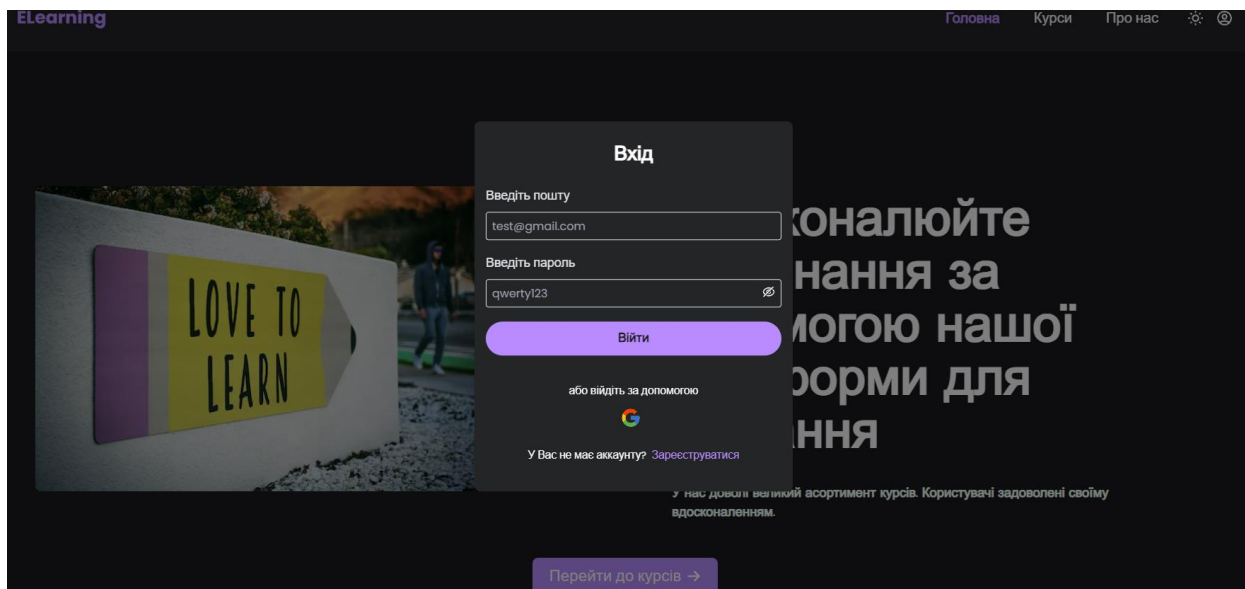


Рис. 3.6. Авторизація

Увійшовши до акаунту, можна змінити облікові дані за потреби (рис. 3.7). Дана сторінка складається із чотирьох табів для користувача і з п'яти табів для адміністратора: мій профіль, змінити пароль, мої курси, панель адміністратора (тільки для адміністратора) та вийти.

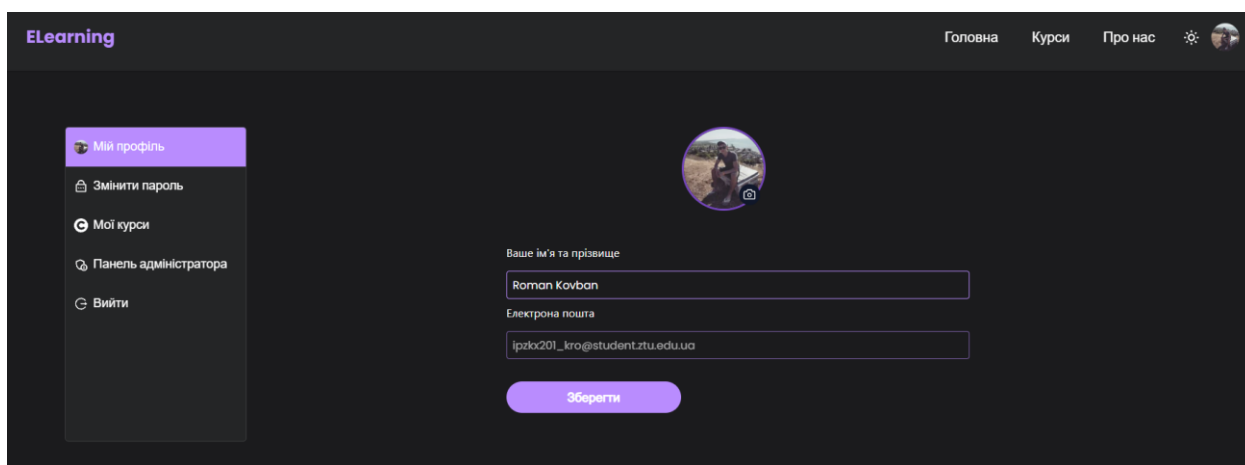


Рис. 3.7. Сторінка профілю

Авторизувавшись з ролі Адміністратор можна перейти до адміністративної панелі та до пункту створити курс (рис. 3.8). На даній сторінці буде форма, в якій буде можливість ввести всі необхідні поля в декілька етапів. Перший етап - інформація про курс. Адміністратор заповнює поля, що містять інформацію про назву курсу, опис, вартість, орієнтовну ціну, теги, категорія, рівень курсу, демонстративне відео, зображення курсу (рис.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 43   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

3.9). Другий етап – параметри курсу. Адміністратор вказує переваги та вимоги, щоб успішно пройти курс (рис. 3.10). Третій етап – зміст курсу. Адміністратор заповнює назву розділу, назву відео, URL-адрес відео, а саме вказує ідентифікатор раніше завантаженого відео на платформу VdoCipher, тривалість відео, опис відео, посилання (рис. 3.11). Четвертий етап – перегляд курсу. Адміністратор переглядає раніше заповнену інформацію і якщо все гаразд, публікує курс (рис. 3.12).

Назва курсу: ReactJS для початківців

Опис курсу: Опис курсу, як технології...

Вартість курсу: 200

Орієнтовна ціна (необов'язково): 200

Теги курсу: React, Redux

Категорія: Вибірть категорію

Рівень курсу: Початковий/Середній/Професійний

Демо URL: eef74fd

Далі

Рис. 3.8. Сторінка створення

Назва курсу: React - повний посібник 2024

Опис курсу: Ласкаво просимо до "React - повний посібник". Цей курс навчить вас React.js у практичний спосіб, використовуючи всі новітні шаблони та найкращі практики, які вам потрібні. Ви дізнаєтеся про всі основні основи, а також розширені концепції та пов'язані теми, які допоможуть вам стати розробником React.js. Це величезний курс, оскільки він справді охоплює ВСЕ, що вам потрібно знати та навчитися, щоб стати розробником React.js!

Вартість курсу: 500

Орієнтовна ціна (необов'язково): 900

Теги курсу: React

Категорія: ReactJS

Рівень курсу: Початковий

Демо URL: 5e7ff32265364f83925edff64ec3c035

Рис. 3.9. Заповнення інформації про курс

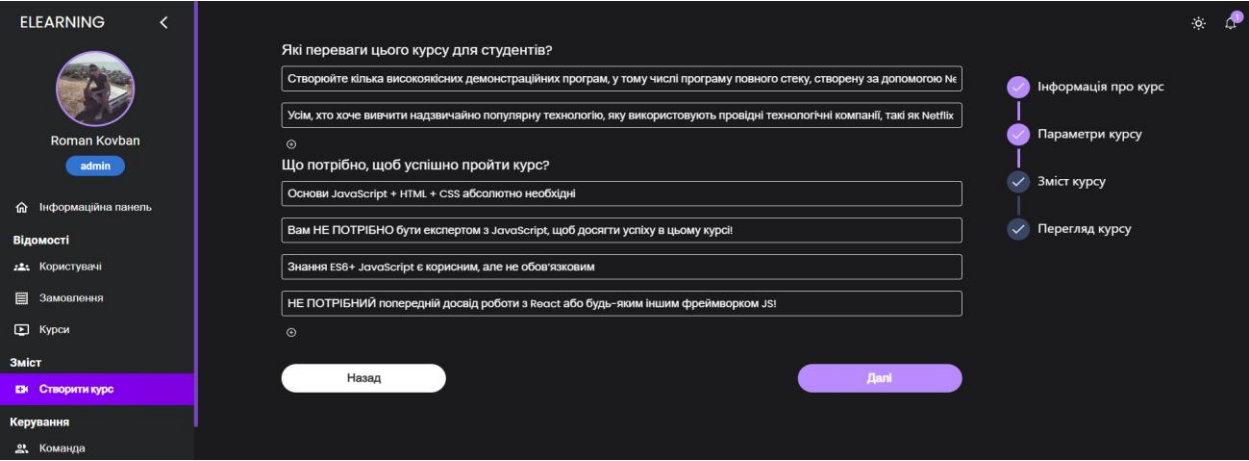


Рис. 3.10. Параметри курсу

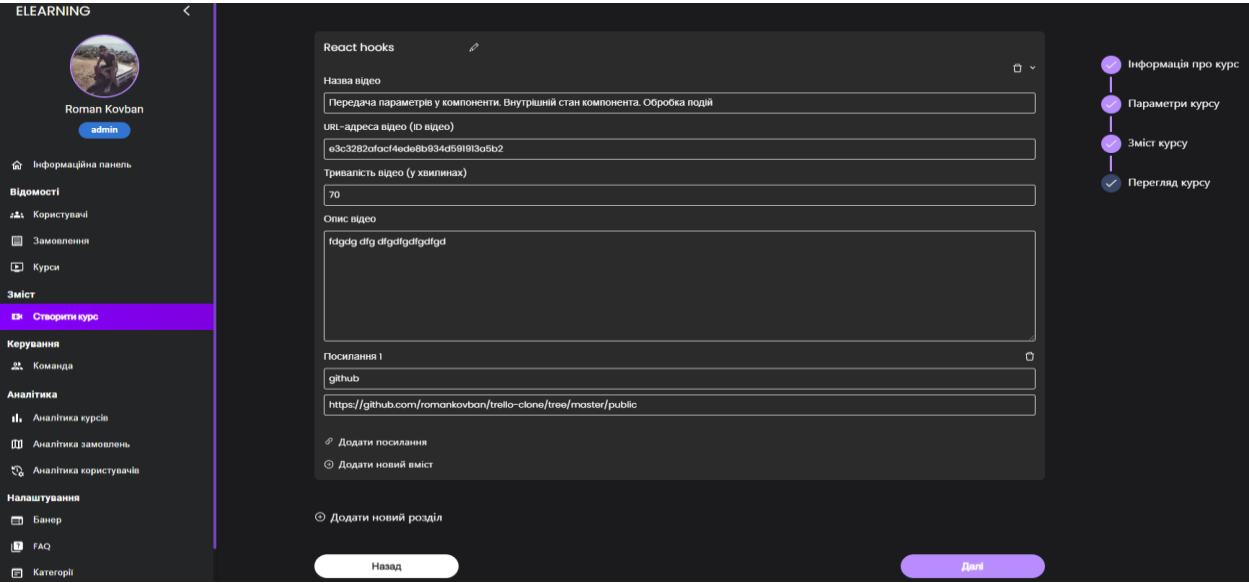


Рис. 3.11. Зміст курсу

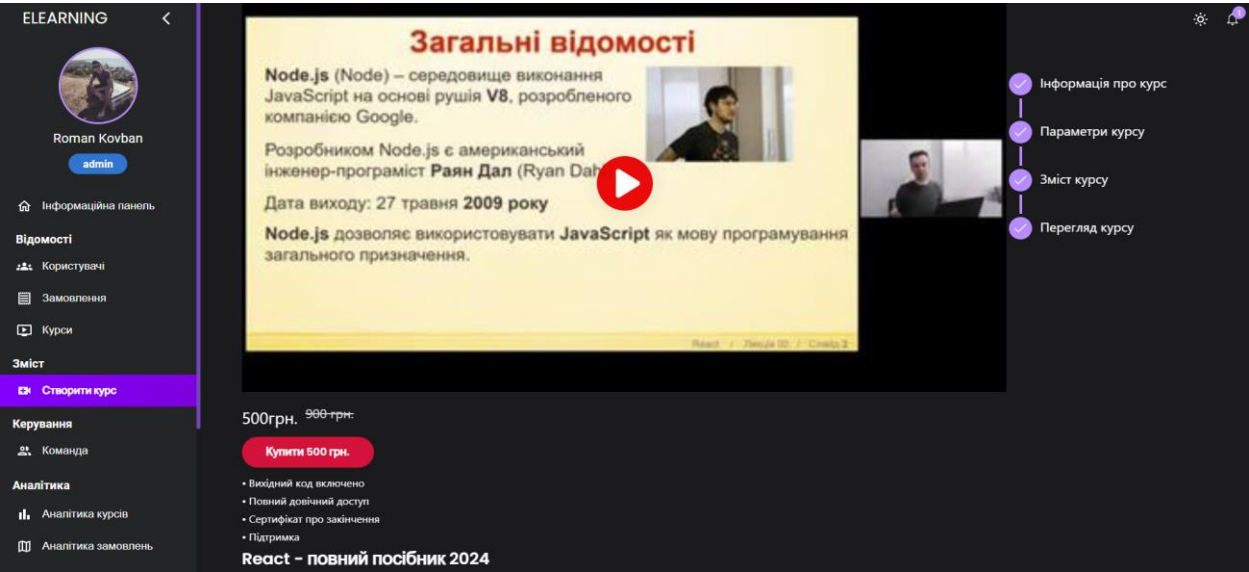
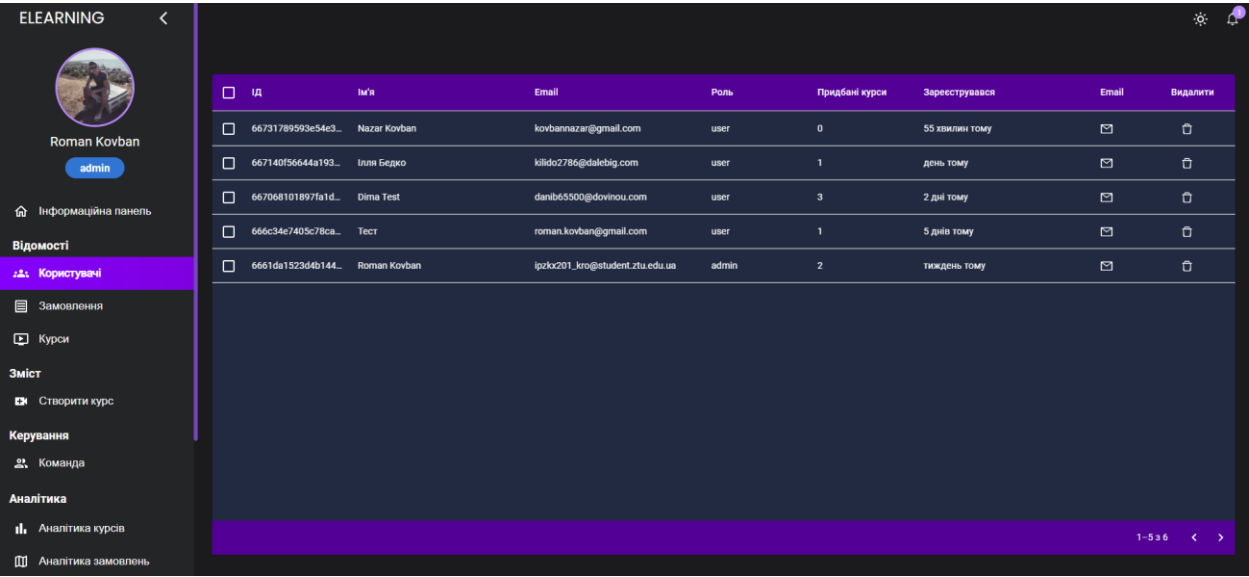


Рис. 3.12. Перегляд курсу перед публікацією

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 45   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

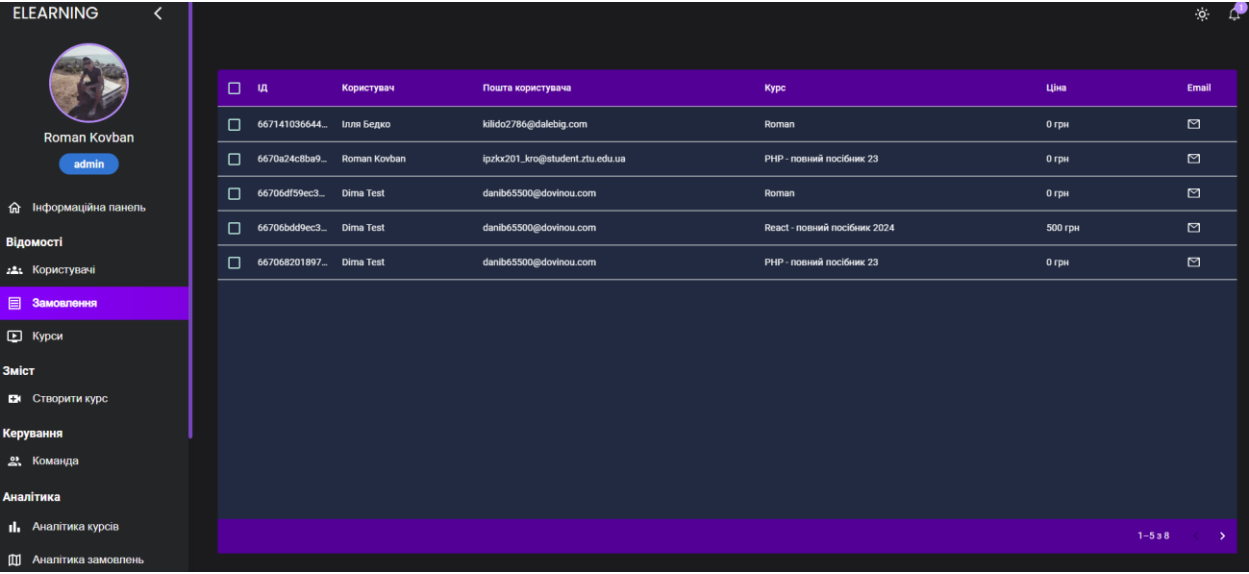
Зі сторони адміністратора є доступ до списку користувачів з інформацією про них (рис. 3.13).



| <input type="checkbox"/> | ID                  | Ім'я         | Email                            | Роль  | Придбані курси | Зареєструвався | Email | Видалити |
|--------------------------|---------------------|--------------|----------------------------------|-------|----------------|----------------|-------|----------|
| <input type="checkbox"/> | 66731789593e54e3... | Nazar Kovban | kovbannazar@gmail.com            | user  | 0              | 55 хвилин тому |       |          |
| <input type="checkbox"/> | 667140f56644a193... | Ілля Бедко   | ilido2786@dalebig.com            | user  | 1              | день тому      |       |          |
| <input type="checkbox"/> | 667068101897fa1d... | Dima Test    | danib65500@dovinou.com           | user  | 3              | 2 дні тому     |       |          |
| <input type="checkbox"/> | 666c34e7405c78ca... | Тест         | roman.kovban@gmail.com           | user  | 1              | 5 днів тому    |       |          |
| <input type="checkbox"/> | 6661da1523d4b144... | Roman Kovban | lpzxx201_kro@student.zftu.edu.ua | admin | 2              | тиждень тому   |       |          |

Рис. 3.13. Список користувачів

Адміністратор також має можливість переглядати список замовлень (рис. 3.14).



| <input type="checkbox"/> | ID              | Користувач   | Пошта користувача                | Курс                         | Ціна    | Email |
|--------------------------|-----------------|--------------|----------------------------------|------------------------------|---------|-------|
| <input type="checkbox"/> | 667141036644... | Ілля Бедко   | ilido2786@dalebig.com            | Roman                        | 0 грн   |       |
| <input type="checkbox"/> | 6670a24c8ba9... | Roman Kovban | lpzxx201_kro@student.zftu.edu.ua | PHP - повний посібник 23     | 0 грн   |       |
| <input type="checkbox"/> | 66706df59ec3... | Dima Test    | danib65500@dovinou.com           | Roman                        | 0 грн   |       |
| <input type="checkbox"/> | 66706bdf9ec3... | Dima Test    | danib65500@dovinou.com           | React - повний посібник 2024 | 500 грн |       |
| <input type="checkbox"/> | 667068201897... | Dima Test    | danib65500@dovinou.com           | PHP - повний посібник 23     | 0 грн   |       |

Рис. 3.14. Список замовлень

Також адміністратор може додавати та редагувати створені курси (рис. 3.15).



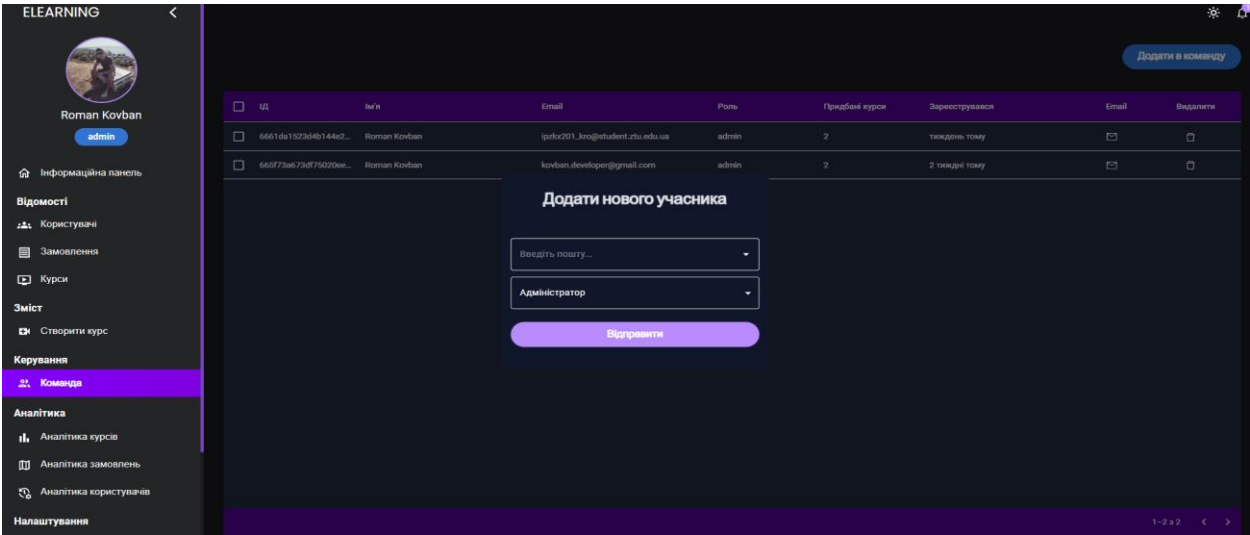


Рис. 3.17. Додавання користувача до команди

Адміністратор може переглядати аналітику додавання курсів (рис. 3.18), аналітику замовлень (рис. 3.19) та аналітику реєстрації користувачів (рис. 3.20).

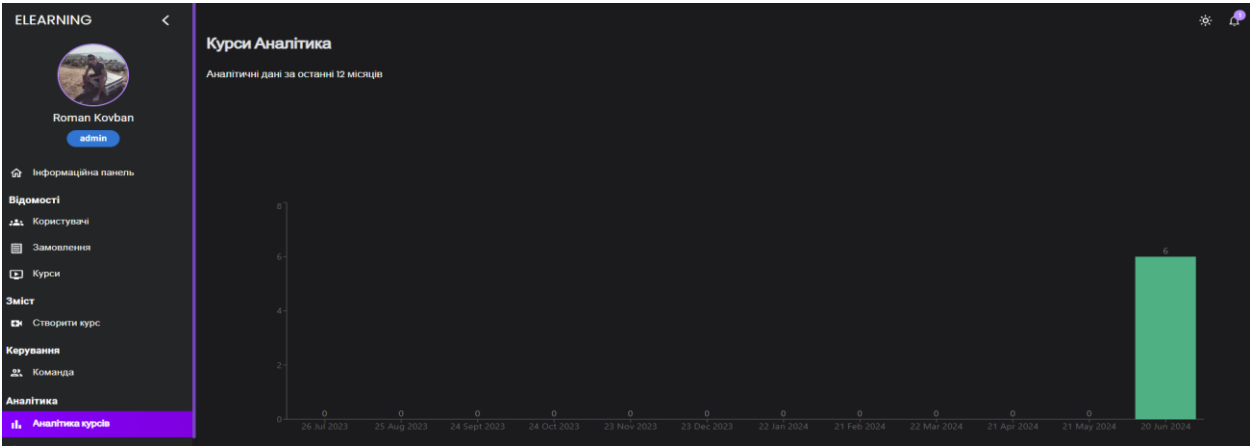


Рис. 3.18. Аналітика додавання курсів

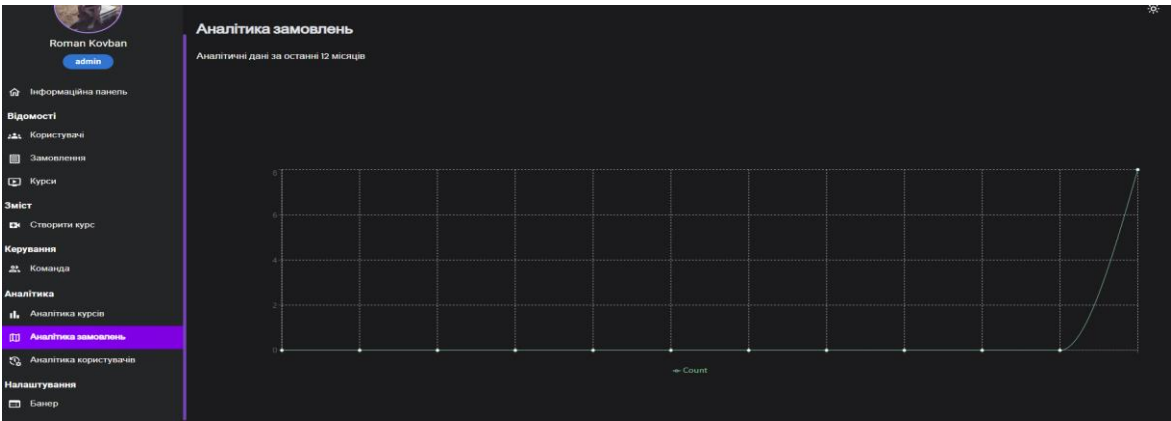


Рис. 3.19. Аналітика замовлень

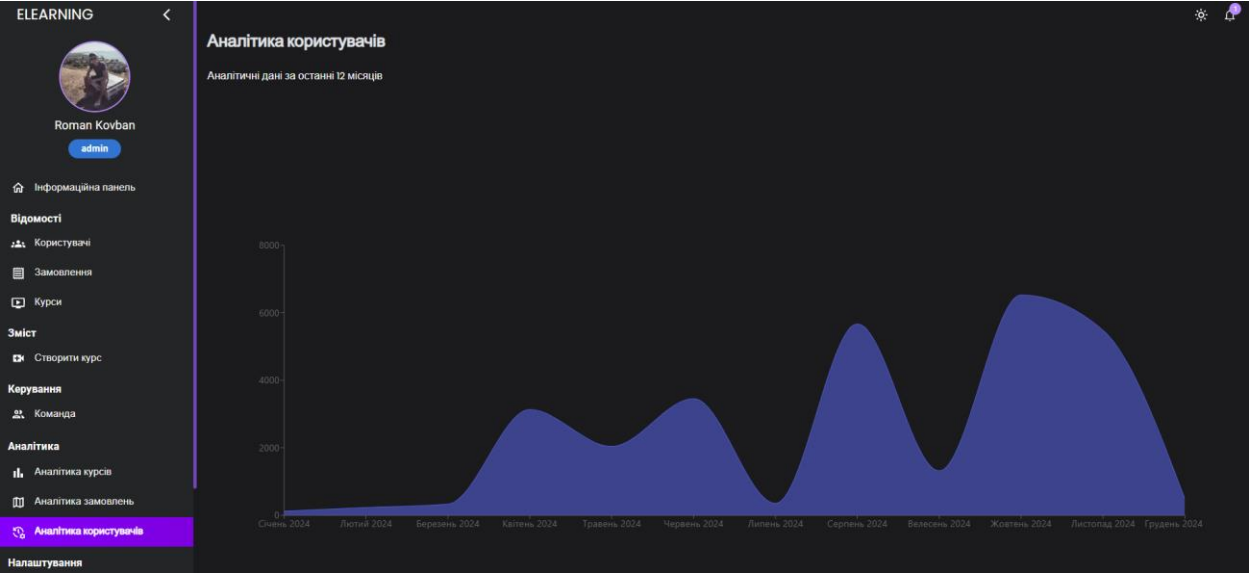


Рис. 3.20. Аналітика реєстрації користувачів

Для адміністратора є можливість змінювати банер (рис. 3.21), FAQ (рис. 3.22), категорії (рис. 3.23) через адміністративну панель.

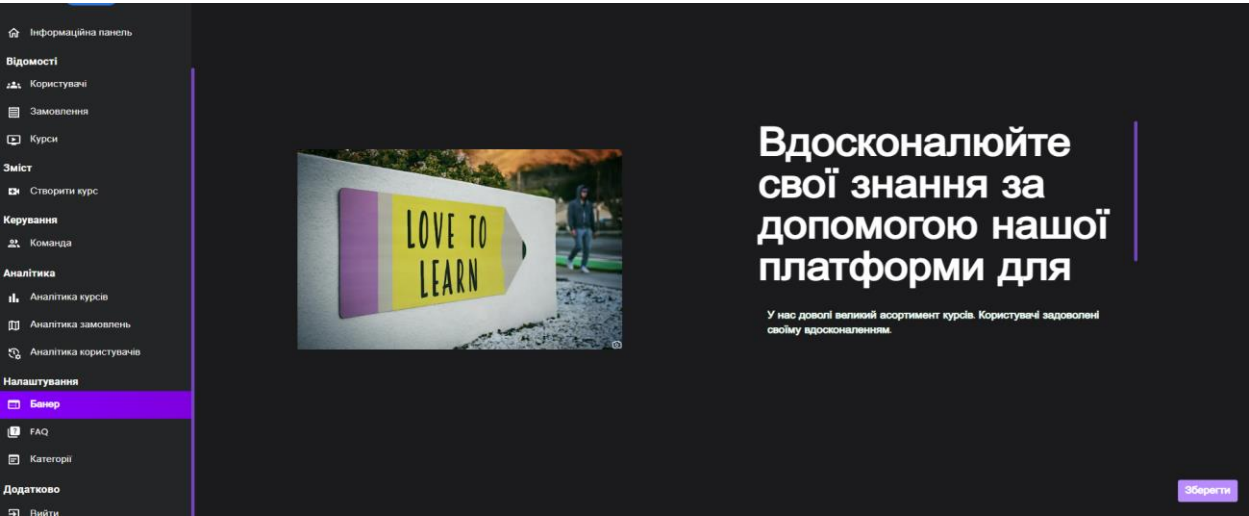


Рис. 3.21. Редагування банера

Чим можна здійснити оплату?

Оплату можна здійснити за допомогою карт Visa, MasterCard

Як зареєструватися на курс?

Перейти на курс та натиснути зареєструватися на курс або придбати курс, якщо він платний

Рис. 3.22. Редагування FAQ

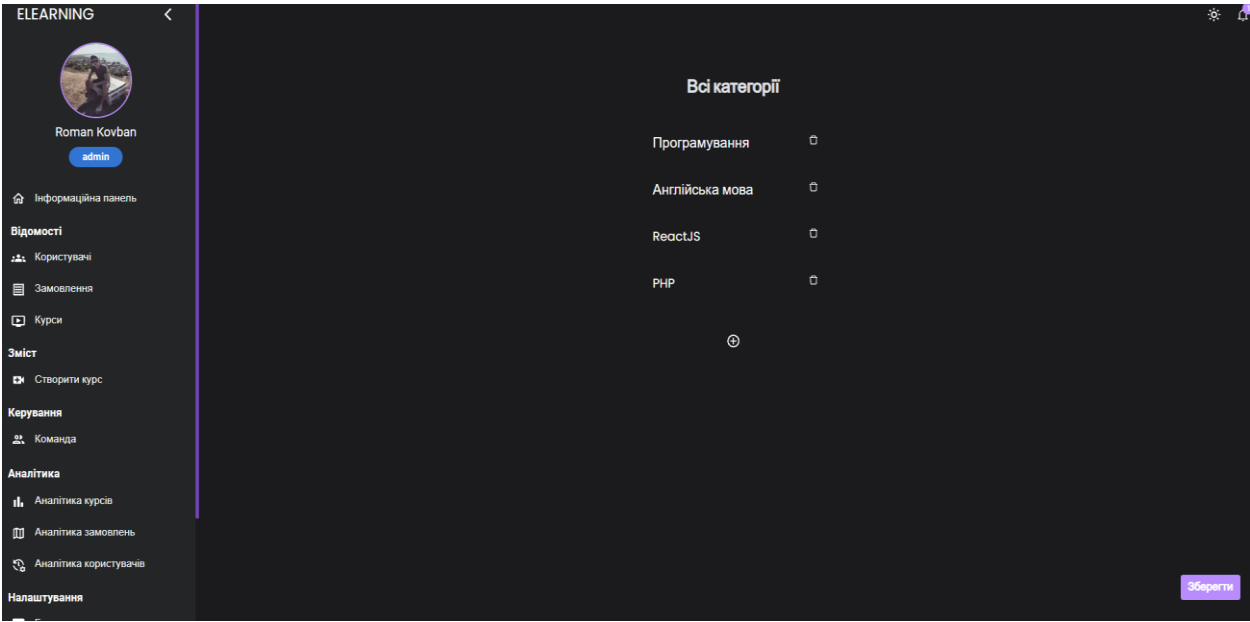


Рис. 3.23. Редагування категорій

Адміністратор може переглянути всю аналітику перейшовши на інформаційну панель (рис. 3.24).

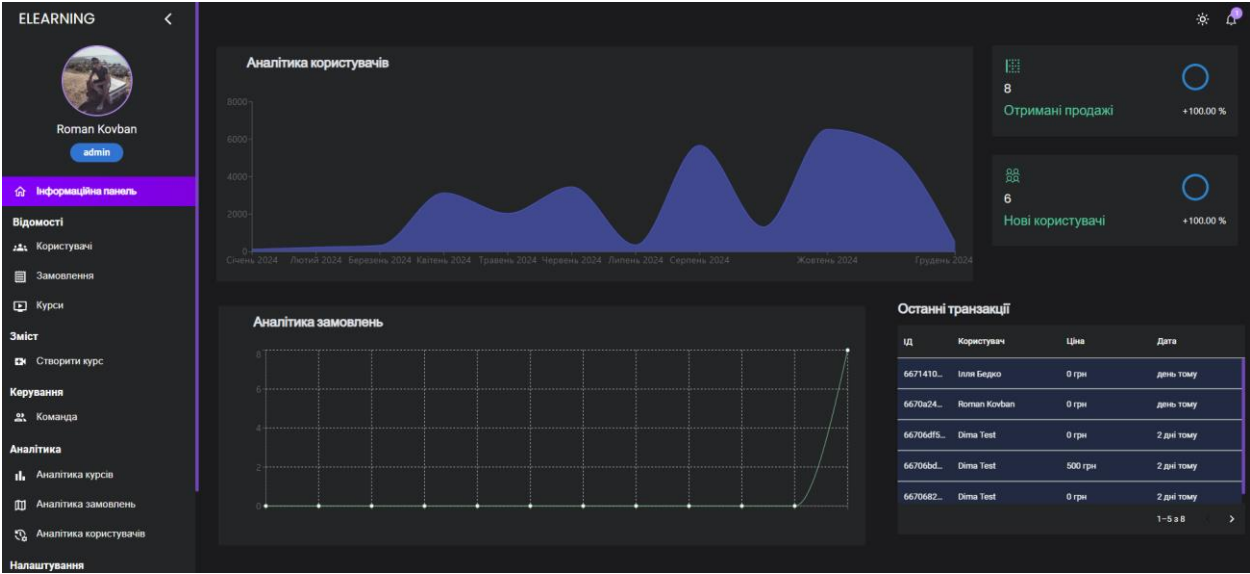


Рис. 3.24. Головна сторінка адміністративної панелі

Адміністратор може отримувати сповіщення якщо користувача залишив відгук чи здійснив замовлення (рис. 3.25).

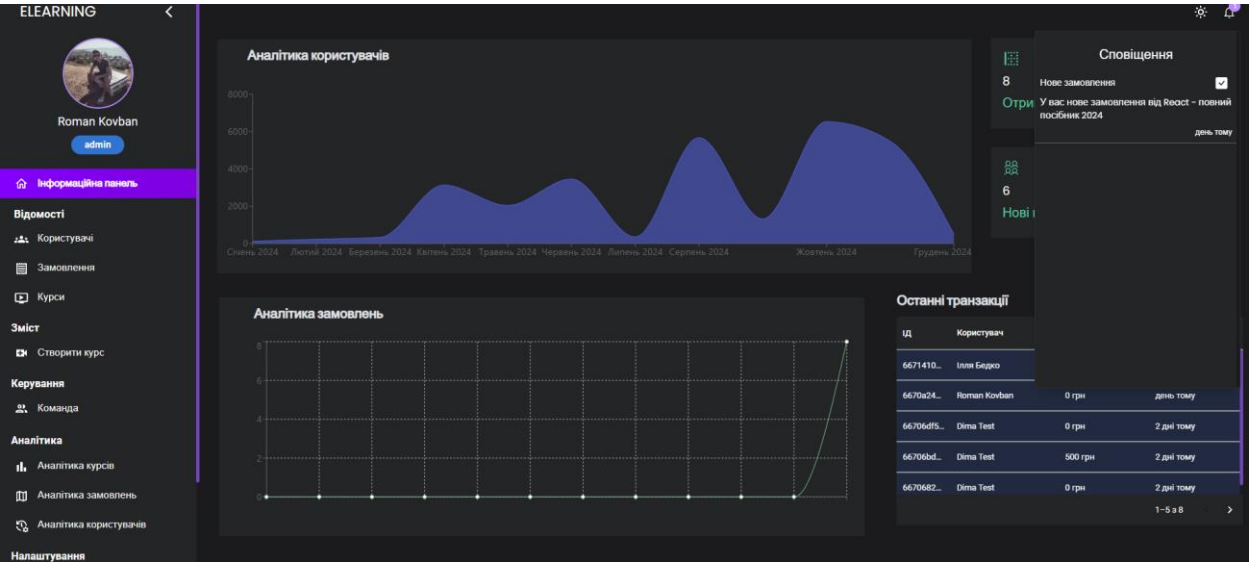


Рис. 3.25. Сповіщення для адміністратора

Користувач може купувати курси. Для цього йому потрібно перейти на курс і натиснути на кнопку «Купити» та заповнити форму даними про банківську карту (рис. 3.26).

Рис. 3.26. Форма для купівлі курсу

Користувачі можуть залишати відгуки до курсів (рис. 3.27). Вони відображаються на сторінці курсу.

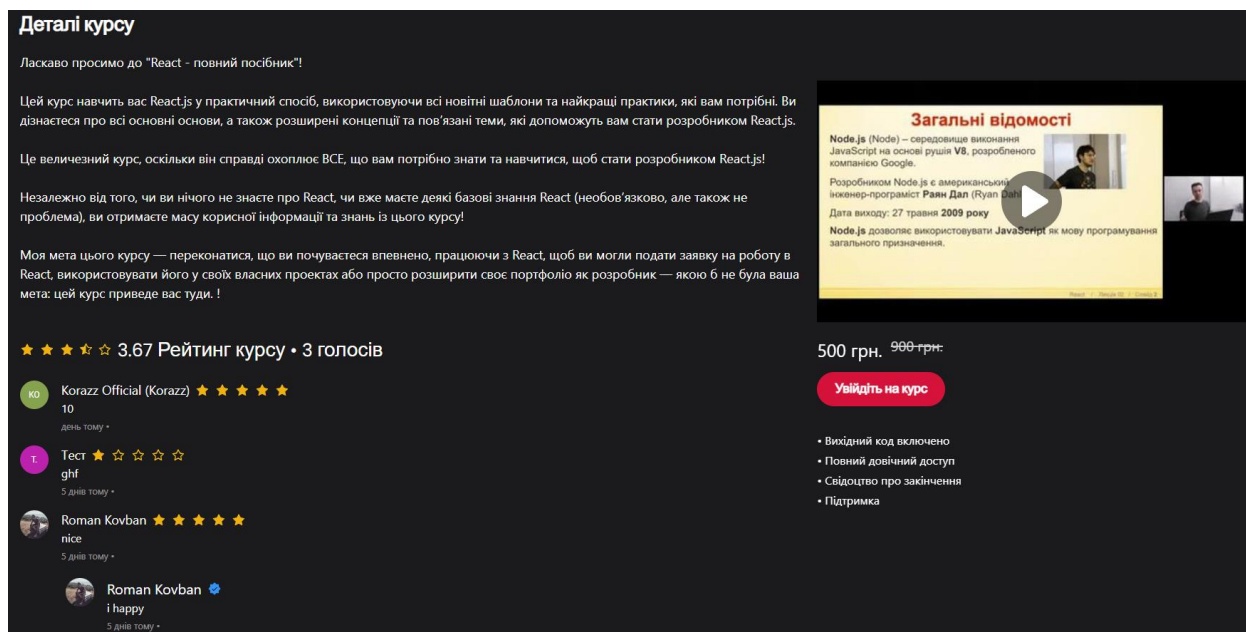


Рис. 3.27. Відгуки до курсу

Під час проходження курсу, користувачі можуть задавати питання та відповідати на запитання до уроків (рис. 3.28).

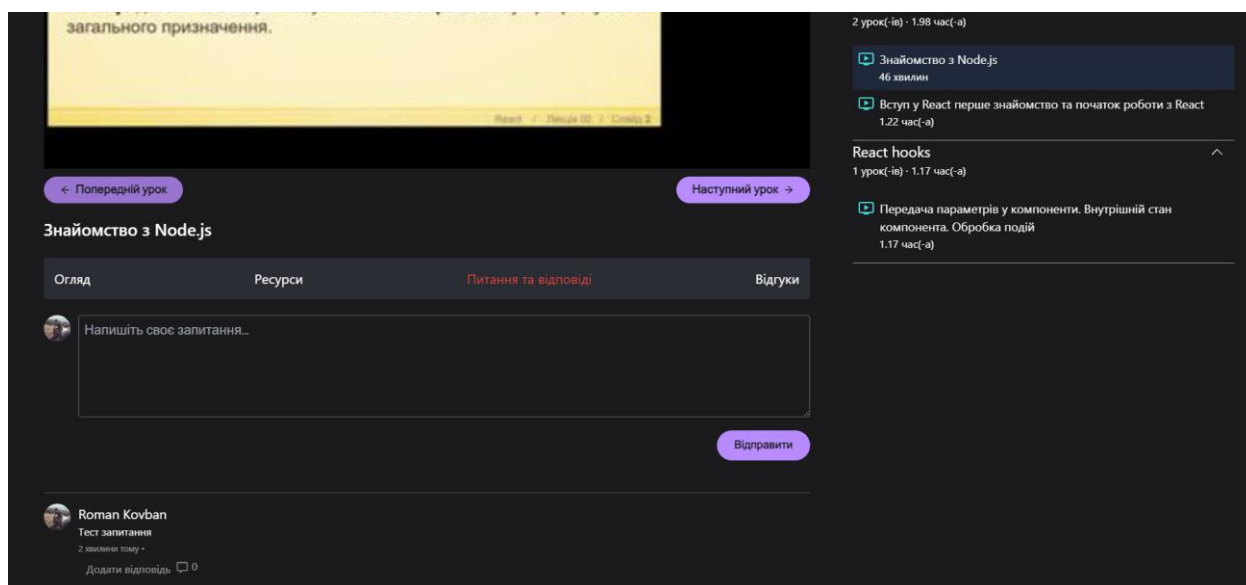


Рис. 3.28. Запитання та відповіді до уроку

На сторінці проходження курсу, користувач може перемикатися між уроками (рис. 3.29).

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 52   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

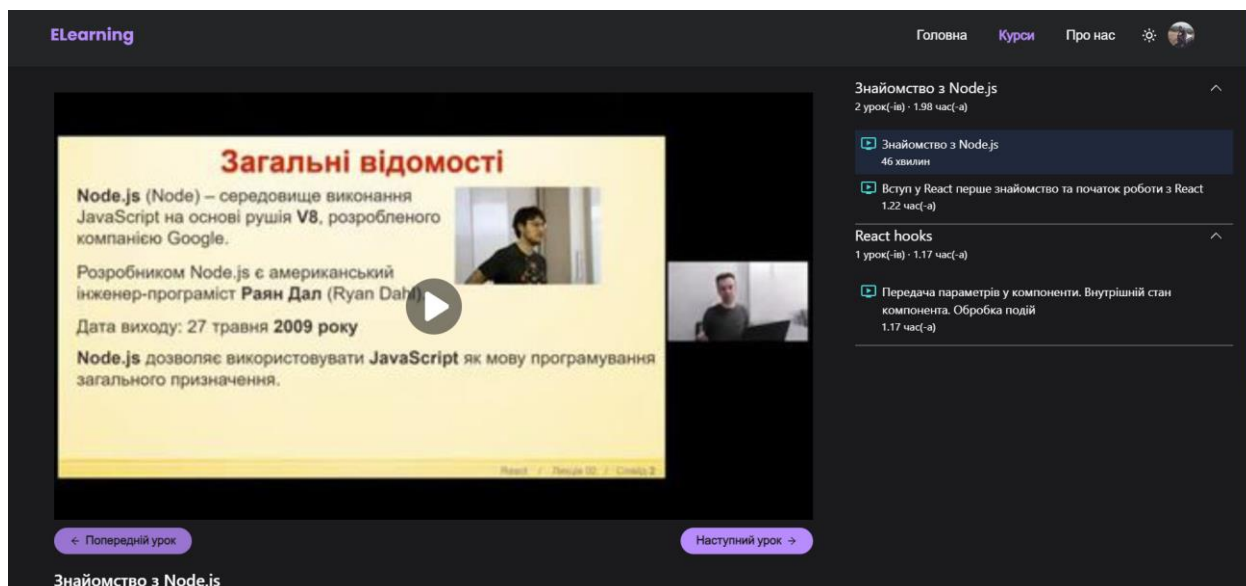


Рис. 3.29. Сторінка проходження курсу

Також на сторінці проходження курсу, для кожного уроку присутній огляд з детальною інформацією про урок (рис. 3.30).

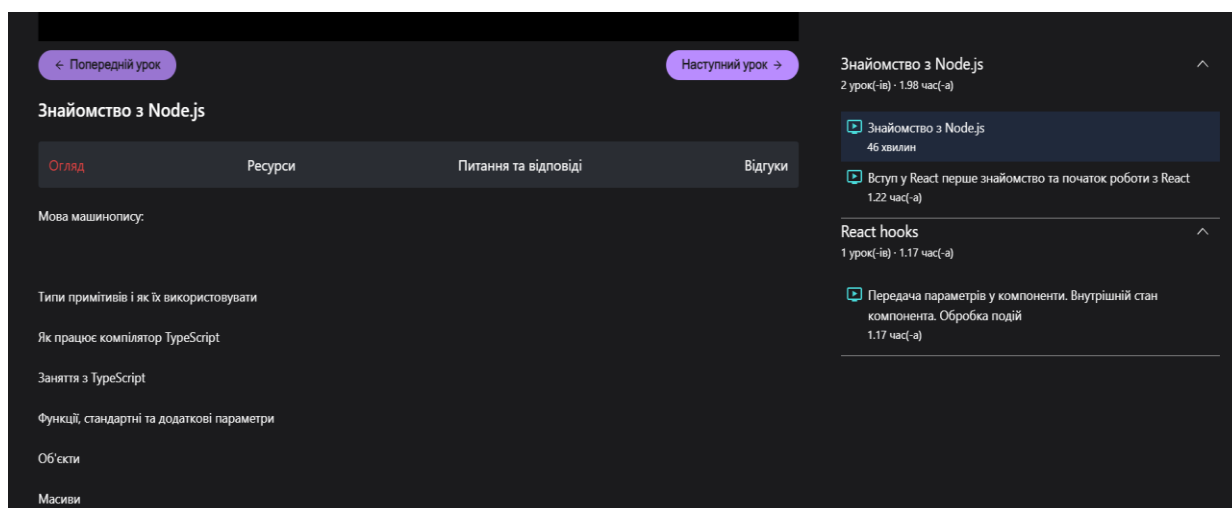


Рис. 3.30. Огляд до уроку

До кожного уроку є додаткові ресурси по пройденому матеріалу або для його закріплення (рис. 3.31).

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 53   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

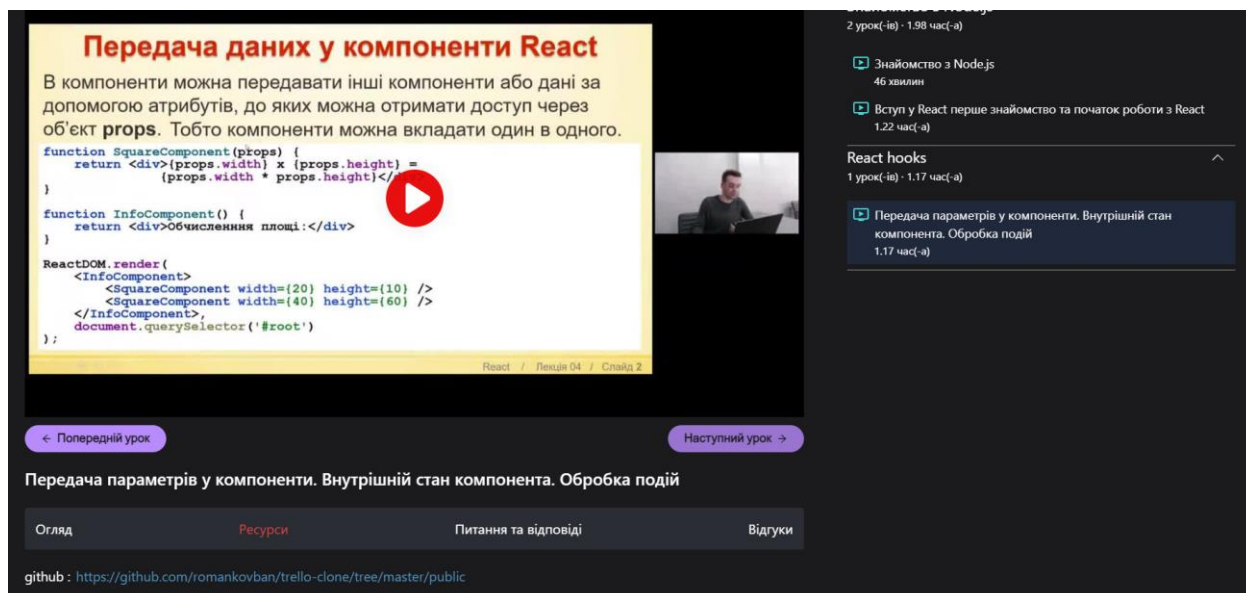


Рис. 3.31. Ресурси до уроку

На сторінці проходження курсу є можливість оцінити та написати відгук про курс (рис. 3.32).

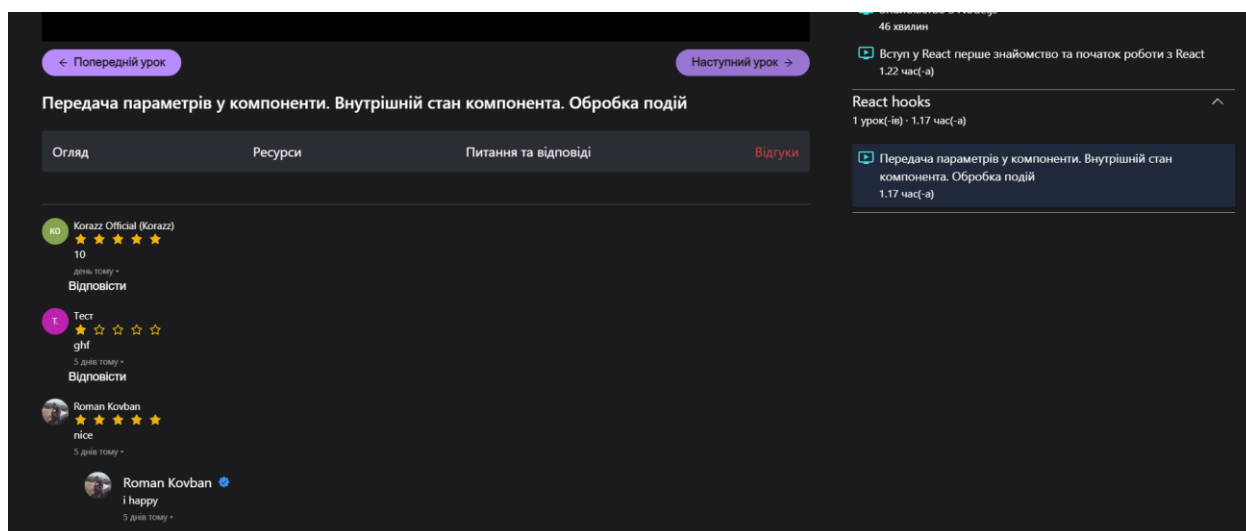


Рис. 3.32. Ресурси до уроку

Користувач може перейти на сторінку курсів, де він може робити фільтрацію за допомоги категорії та пошуку (рис. 3.33).

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 54   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

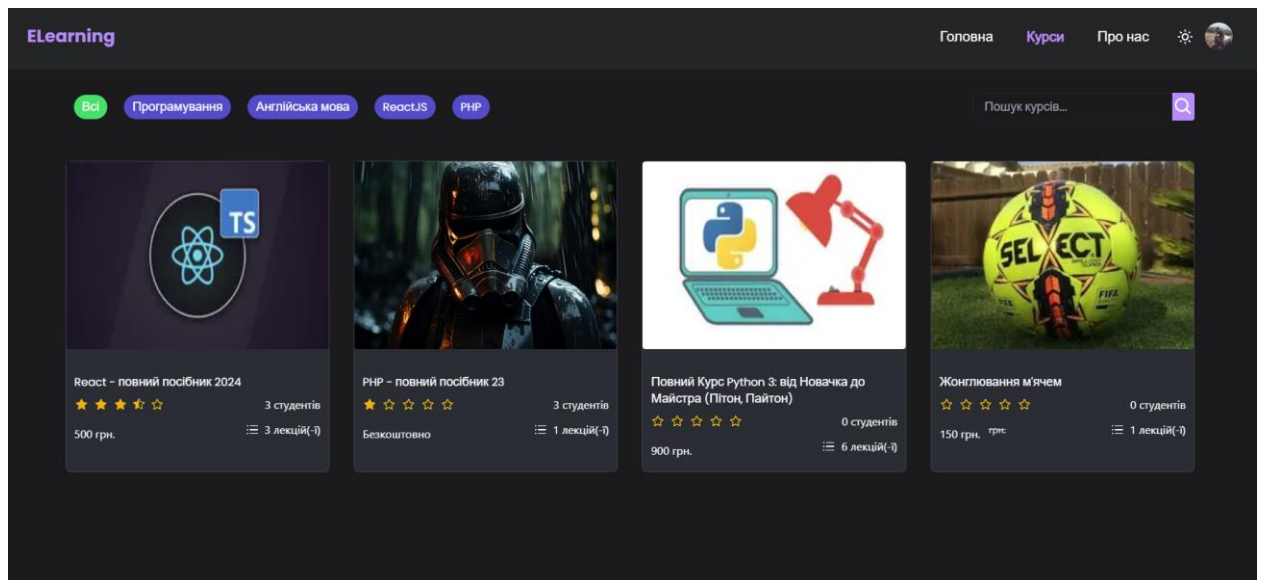


Рис. 3.33. Сторінка курсів

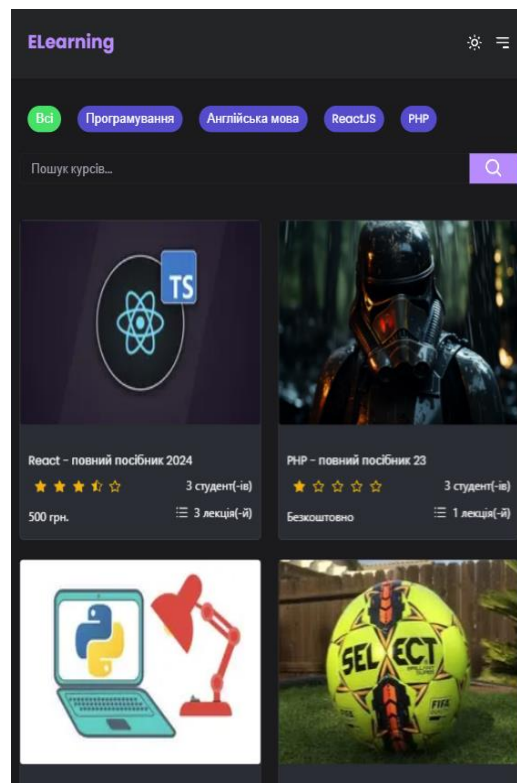


Рис. 3.34. Сторінка курсів на телефоні

У цьому підрозділі ми розглянули інтерфейс і робочий процес онлайн-платформи проходження курсів. Описано сторінки та модулі системи для розуміння, як користувачі взаємодіють із системою.

### 3.3. Проведення та результати тестування платформи

Тестування роботи платформи відбувається впродовж розробки модулів системи. Для перевірки всіх аспектів розробленого додатку проводиться тестування як функціональне так і не функціональне. При розробці освітньої платформи було враховано особливості та всі можливі ситуації для забезпечення безвідмовної роботи системи та створення захищеної системи. Під час реалізації було впроваджено перевірки як на серверній частині, так і на стороні клієнта для зменшення можливостей відмови системи.

При спробі створити курс без вказання переваг та що потрібно знати, можна побачити відповідне повідомлення (рис. 3.35).

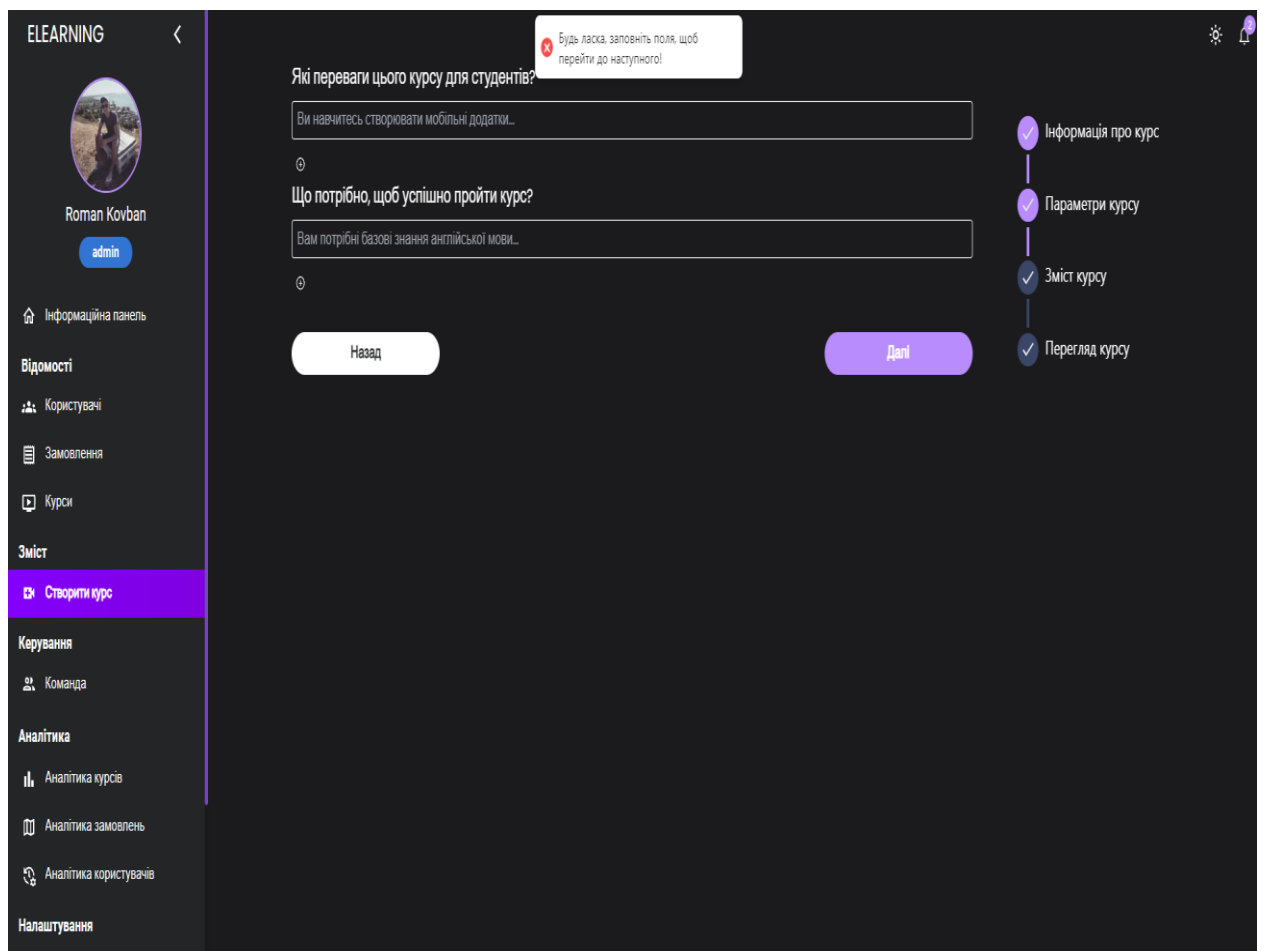


Рис. 3.35. Тестування створення курсу

При спробі активації акаунту за допомоги невірного коду (рис. 3.36).

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 56   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

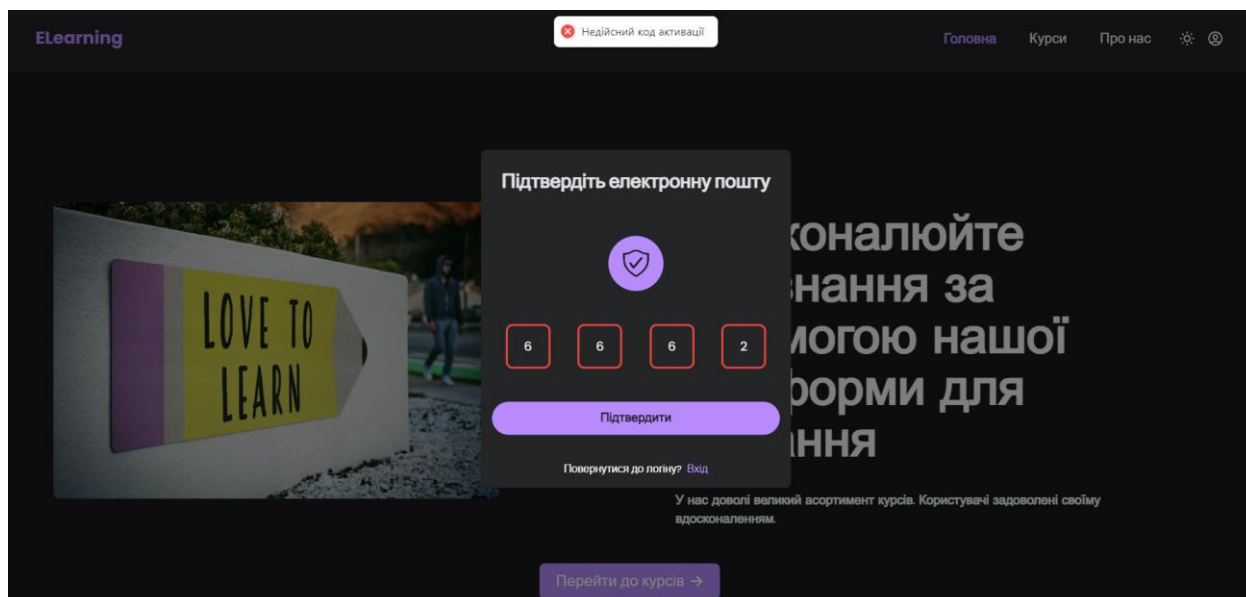


Рис. 3.36. Тестування активації акаунту

Тестування купівлі курсу з невірними значеннями (рис. 3.37). На рисунку зображена спроба купити курс з невірними значеннями.

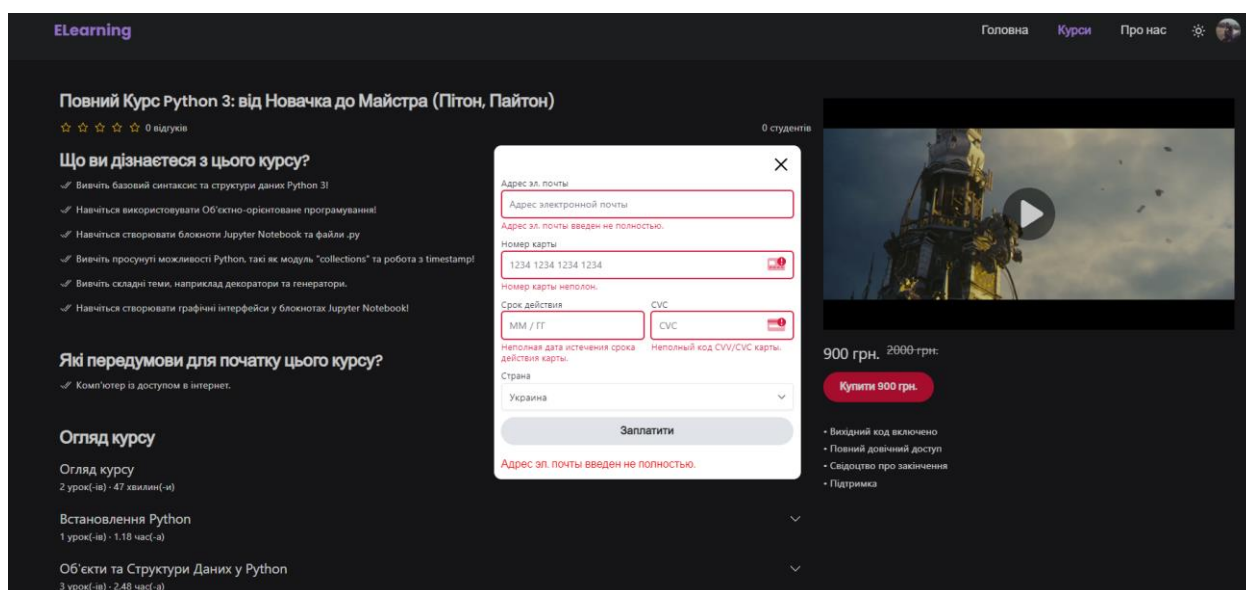


Рис. 3.37. Спроба купити курс

Тестування введення невірних даних під час авторизації (рис. 3.38). На рисунку зображена спроба ввести невірні дані у форму авторизації.

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 57   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

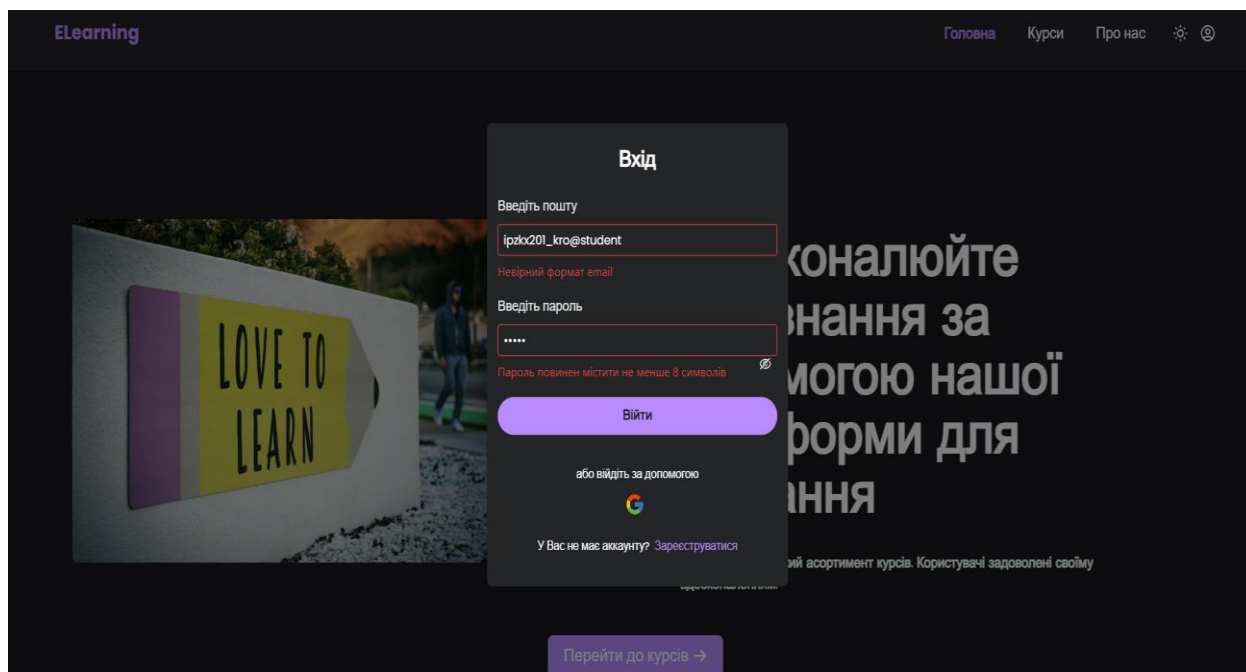


Рис. 3.38. Тестування введення невірних даних

У цьому підрозділі ми провели тестування програмного додатку згідно з нормами розробки програмного забезпечення. Також продемонстровано результати тестування.

### Висновки до третього розділу

Третій розділ містив налаштування середовища та вимоги для розгортання програмного додатку. Було продемонстровано інтерфейс веб-сайту та проведено тестування системи.

Визначено оптимальні вимоги до системи для забезпечення стабільної роботи продукту. Надано інструкції для встановлення додатку та його розгортання на сервері. Кроки розроблені для зручності користувачів, щоб додаток можна було використовувати різними користувачами.

Далі продемонстровано структуру інтерфейсу онлайн-платформи проходження курсів. Описано сторінки та модулі системи для розуміння, як користувачі взаємодіють із системою. У результаті отримано зручний та адаптивний інтерфейс, який дозволяє ефективно використовувати функціонал системи без надмірної кількості дій.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 58   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

Проведено тестування програмного додатку згідно з нормами розробки програмного забезпечення. Також продемонстровано результати тестування, зокрема оброблення помилок, які можуть виникнути в системі.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 59   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

## ВИСНОВКИ

Під час виконання випускної кваліфікаційної роботи було проведено аналіз, розробку та реалізацію онлайн-платформи проходження курсів. Проєкт складався з трьох основних розділів, кожен з яких відповідав етапам розробки додатку.

У першому розділі проведено повний аналіз теми, досліджено можливі рішення та проаналізовано інформаційні технології. Метою розділу було визначення чіткої задачі для проєкту, опис вимог та завдань, які мають бути виконані. Також проведено аналіз існуючих аналогічних рішень, визначено їх особливості, переваги та недоліки, а також підведено підсумки для формування цілей проєкту. Один із головних аспектів розділу – аналіз поширених архітектурних моделей розробки, їх особливості та вибір найбільш підходящої для проєкту з урахуванням можливості розширення, зручності розробки, підтримки та ефективності використання.

Другий розділ присвячений проєктуванню та реалізації програмного продукту. Визначено всі функціональні та нефункціональні вимоги до системи. Для наочного бачення створено діаграми використання, а також описані основні алгоритми роботи системи за допомогою діаграм активності. Важливою частиною розділу була реалізація функціональних рішень та розробка додатку, при якій враховано всі вимоги та попередньо визначені завдання.

Третій розділ детально описує процес встановлення програмного додатку, його розгортання та демонстрацію інтерфейсу з тестуванням. Визначено вимоги до системи, на якій має бути розгорнутий проєкт, та надано покрокову інструкцію з встановлення необхідних компонентів, налаштування конфігураційних файлів та розгортання проєкту. В розділі було продемонстровано спроектований інтерфейс, який був розроблений для зручності та інтуїтивно зрозумілою роботи користувачів. Також було проведено ретельне комплексне тестування проєкту, результат якого було продемонстровано у розділі.

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 60   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

Результатом виконання проєкту став завершений програмний продукт – онлайн-платформа для проходження курсів. Реалізований проєкт може бути використаний у сфері онлайн-навчання та здатний скласти конкуренцію на ринку. Особливістю проєкту є його масштабованість, архітектура розроблена з можливістю розширення, додавання нового без шкоди вже створеному.

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                    | 61   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Next.js [Електронний ресурс] - Режим доступу до ресурсу: <https://nextjs.org/docs>
2. Next.js Component Image [Електронний ресурс] - Режим доступу до ресурсу: <https://nextjs.org/docs/app/api-reference/components/image>
3. next.config.js [Електронний ресурс] - Режим доступу до ресурсу: <https://nextjs.org/docs/app/api-reference/next-config-js>
4. useSearchParams [Електронний ресурс] - Режим доступу до ресурсу: <https://nextjs.org/docs/app/api-reference/functions/use-search-params>
5. Pages and Layouts [Електронний ресурс] - Режим доступу до ресурсу: <https://nextjs.org/docs/app/building-your-application/routing/pages-and-layouts>
6. Express.js [Електронний ресурс] - Режим доступу до ресурсу: <https://expressjs.com/>
7. Redux Toolkit [Електронний ресурс] - Режим доступу до ресурсу: <https://redux-toolkit.js.org/tutorials/typescript>
8. RTK Query [Електронний ресурс] - Режим доступу до ресурсу: <https://redux-toolkit.js.org/tutorials/rtk-query>
9. Cloudinary [Електронний ресурс] - Режим доступу до ресурсу: <https://cloudinary.com/>
10. VdoCipher [Електронний ресурс] - Режим доступу до ресурсу: <https://www.vdocipher.com>
11. Next js with Axios Data Fetching [Електронний ресурс] - Режим доступу до ресурсу: <https://dev.to/syedmuhammadaliraza/next-js-with-axios-data-fetching-435a>
12. ITDVN [Електронний ресурс] - Режим доступу до ресурсу: <https://itvdn.com/ua>
13. Udemу [Електронний ресурс] - Режим доступу до ресурсу: <https://www.udemy.com/>

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 62   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |

14. EdEra [Електронний ресурс] - Режим доступу до ресурсу: <https://ed-era.com/>
15. Tailwind CSS Docs [Електронний ресурс] - Режим доступу до ресурсу: <https://tailwindcss.com/docs/installation>
16. Material UI Docs [Електронний ресурс] - Режим доступу до ресурсу: <https://mui.com/material-ui/getting-started/installation/>
17. Mongoose Docs [Електронний ресурс] - Режим доступу до ресурсу: <https://mongoosejs.com/>
18. Redis [Електронний ресурс] - Режим доступу до ресурсу: <https://redis.io/docs/latest/>
19. Upstash [Електронний ресурс] - Режим доступу до ресурсу: <https://upstash.com/docs/redis/overall/getstarted>
20. Stripe [Електронний ресурс] - Режим доступу до ресурсу: <https://stripe.com/>
21. JWT [Електронний ресурс] - Режим доступу до ресурсу: <https://jwt.io/>
22. MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/>

|      |      |                  |        |      |                    |      |
|------|------|------------------|--------|------|--------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІПЗ.КР.Б-121-24-ІЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                    | 63   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                    |      |

# ДОДАТКИ

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 64   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

Реалізація контролера, що відповідає за замовлення.

Лістинг order.controller.ts:

```
import 'dotenv/config';
import { NextFunction, Request, Response } from 'express';
import { asyncErrorHandler } from '../middleware/async-error-handler';
import { ErrorHandler } from '../utils/error-handler';
import { IOrder } from '../models/order.model';
import userModel from '../models/user.model';
import CourseModel, { ICourse } from '../models/course.model';
import path from 'path';
import ejs from 'ejs';
import sendMail from '../utils/send-mail';
import NotificationModel from '../models/notification.model';
import { getAllOrdersService, newOrder } from '../services/order.service';
import { redis } from '../utils/redis';
const stripe = require('stripe')(`${process.env.STRIPE_SECRET_KEY}`);

export const createOrder = asyncErrorHandler(
  async (req: Request, res: Response, next: NextFunction) => {
    try {
      const { courseId, payment_info } = req.body as IOrder;
      if (payment_info) {
        if ('id' in payment_info) {
          const paymentIntentId = payment_info.id;
          const paymentIntent = await stripe.paymentIntents.retrieve(
            paymentIntentId
          );
          if (paymentIntent.status !== 'succeeded') {
            return next(new ErrorHandler('Помилка при оплаті!', 400));
          }
        }
      }
      const user = await userModel.findById(req.user?._id);
      const courseExistInUser = user?.courses.some(
        (course: any) => course._id.toString() === courseId);
      if (courseExistInUser) {
        return next(new ErrorHandler('Ви вже придбали цей курс', 400));
      }
    }
  }
);
```

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 65   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

```

const course: ICourse | null = await CourseModel.findById(courseId);
if (!course) {
    return next(new ErrorHandler('Курс не знайдено', 404));
}
const data: any = {
    courseId: course._id,
    userId: user?._id,
    payment_info,
};
const mailData = {
    order: {
        _id: course._id.toString().slice(0, 6),
        name: course.name,
        price: course.price,
        date: new Date().toLocaleDateString('uk-UA', {
            year: 'numeric',
            month: 'long',
            day: 'numeric',
        }),
    },
};
const html = await ejs.renderFile(
    path.join(__dirname, '../mails/order-confirmation.ejs'),
    { order: mailData }
);
try {
    if (user) {
        await sendMail({
            email: user.email,
            subject: 'Підтвердження замовлення',
            template: 'order-confirmation.ejs',
            data: mailData,
        });
    }
} catch (error: any) {
    return next(new ErrorHandler(error.message, 500));
}
user?.courses.push(course?._id);
await redis.set(req.user?._id, JSON.stringify(user));
await user?.save();
await NotificationModel.create({

```

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 66   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

```

        user: user?._id,
        title: 'Нове замовлення',
        message: `У вас нове замовлення від ${course?.name}`,
    });
    course.purchased = course.purchased + 1;
    await course.save();
    newOrder(data, res, next);
} catch (error: any) {
    return next(new ErrorHandler(error.message, 500));
}
}
);
export const createOrderFree = asyncErrorHandler(
    async (req: Request, res: Response, next: NextFunction) => {
        try {
            const { courseId } = req.body as IOrder;
            const user = await userModel.findById(req.user?._id);
            const courseExistInUser = user?.courses.some(
                (course: any) => course._id.toString() === courseId
            );
            if (courseExistInUser) {
                return next(new ErrorHandler('Ви вже придбали цей курс', 400));
            }
            const course: ICourse | null = await CourseModel.findById(courseId);
            if (!course) {
                return next(new ErrorHandler('Курс не знайдено', 404));
            }
            const data: any = {
                courseId: course._id,
                userId: user?._id,
                payment_info: null,
            };
            const mailData = {
                order: {
                    _id: course._id.toString().slice(0, 6),
                    name: course.name,
                    price: 0,
                    date: new Date().toLocaleDateString('uk-UA', {
                        year: 'numeric',
                        month: 'long',
                        day: 'numeric',
                    }),
                },
            };

```

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 67   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

```

    }},
  },
};
const html = await ejs.renderFile(
  path.join(__dirname, '../mails/order-confirmation.ejs'),
  { order: mailData }
);
try {
  if (user) {
    await sendMail({
      email: user.email,
      subject: 'Підтвердження замовлення',
      template: 'order-confirmation.ejs',
      data: mailData,
    });
  }
} catch (error: any) {
  return next(new ErrorHandler(error.message, 500));
}
user?.courses.push(course?._id);
await redis.set(req.user?._id, JSON.stringify(user));
await user?.save();
await NotificationModel.create({
  user: user?._id,
  title: 'Нове замовлення',
  message: `У вас нове замовлення від ${course?.name}`,
});
course.purchased = course.purchased + 1;
await course.save();
newOrder(data, res, next);
} catch (error: any) {
  return next(new ErrorHandler(error.message, 500));
}
}
);

export const getAllOrders = asyncErrorHandler(
  async (req: Request, res: Response, next: NextFunction) => {
    try {
      getAllOrdersService(res);
    } catch (error: any) {

```

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 68   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

```

        return next(new ErrorHandler(error.message, 500));
    }
}
);

export const sendStripePublishableKey = asyncErrorHandler(
    async (req: Request, res: Response) => {
        res.status(200).json({
            publishablekey: `${process.env.STRIPE_PUBLISHABLE_KEY}`,
        });
    }
);

export const newPayment = asyncErrorHandler(
    async (req: Request, res: Response, next: NextFunction) => {
        try {
            const myPayment = await stripe.paymentIntents.create({
                amount: req.body.amount,
                currency: 'UAH',
                metadata: {
                    company: 'E-Learning',
                },
                automatic_payment_methods: {
                    enabled: true,
                },
            });
            res.status(201).json({
                success: true,
                client_secret: myPayment.client_secret,
            });
        } catch (error: any) {
            return next(new ErrorHandler(error.message, 500));
        }
    }
);

```

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 69   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

## Реалізація контролера, що відповідає за сповіщення

Лістинг notification.controller.ts:

```

import { NextFunction, Request, Response } from 'express';
import cron from 'node-cron';
import NotificationModel from '../models/notification.model';
import { asyncErrorHandler } from '../middleware/async-error-handler';
import { ErrorHandler } from '../utils/error-handler';

export const getNotifications = asyncErrorHandler(
  async (req: Request, res: Response, next: NextFunction) => {
    try {
      const notifications = await NotificationModel.find().sort({
        createdAt: -1,
      });

      res.status(201).json({
        success: true,
        notifications,
      });
    } catch (error: any) {
      return next(new ErrorHandler(error.message, 500));
    }
  }
);

export const updateNotification = asyncErrorHandler(
  async (req: Request, res: Response, next: NextFunction) => {
    try {
      const notification = await NotificationModel.findById(req.params.id);
      if (!notification) {
        return next(new ErrorHandler('Сповіщення не знайдено', 404));
      } else {
        notification.status
          ? (notification.status = 'read')
          : notification?.status;
      }
      await notification.save();
      const notifications = await NotificationModel.find().sort({
        createdAt: -1,
      });
    }
  }
);

```

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 70   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

```

    res.status(201).json({
      success: true,
      notifications,
    });
  } catch (error: any) {
    return next(new ErrorHandler(error.message, 500));
  }
}
);

cron.schedule('0 0 0 * * *', async () => {
  const thirtyDaysAgo = new Date(Date.now() - 30 * 24 * 60 * 60 * 1000);
  await NotificationModel.deleteMany({
    status: 'read',
    createdAt: { $lt: thirtyDaysAgo },
  });
  console.log('Видалено прочитані сповіщення');
});

```

|      |      |                  |        |      |                   |      |
|------|------|------------------|--------|------|-------------------|------|
|      |      | Ковбан Р.О.      |        |      | ПЗ.КР.Б-121-24-ПЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                   | 71   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                   |      |

Реалізація функції, що відповідає за відправку повідомлень на пошту

Лістинг send-mail.ts:

```
import 'dotenv/config';
import nodemailer, { Transporter } from 'nodemailer';
import ejs from 'ejs';
import path from 'path';

interface EmailOptions {
  email: string;
  subject: string;
  template: string;
  data: {
    [key: string]: any;
  };
}

const sendMail = async (options: EmailOptions): Promise<void> => {
  const transporter: Transporter = nodemailer.createTransport({
    host: process.env.SMTP_HOST,
    port: parseInt(process.env.SMTP_PORT || '587'),
    service: process.env.SMTP_SERVICE,
    auth: {
      user: process.env.SMTP_MAIL,
      pass: process.env.SMTP_PASSWORD,
    },
  });

  const { email, subject, template, data } = options;
  const templatePath = path.join(__dirname, '../mails', template);

  const html: string = await ejs.renderFile(templatePath, data);
  const mailOptions = {
    from: process.env.SMTP_MAIL,
    to: email,
    subject,
    html,
  };

  await transporter.sendMail(mailOptions);
};

export default sendMail;
```

|      |      |                  |        |      |                     |      |
|------|------|------------------|--------|------|---------------------|------|
|      |      | Ковбан Р.О.      |        |      | ІІЗ.КР.Б-121-24-ІІЗ | Арк. |
|      |      | Лобанчикова Н.М. |        |      |                     | 72   |
| Змн. | Арк. | № докум.         | Підпис | Дата |                     |      |