

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 «Веб-технології»)

на тему:

**«Вебплатформа для організації та проведення навчальних
курсів»**

Виконав студент групи ВТ-21-1
ДЗЮБЧУК Данило Романович

Керівник роботи:
ФАНТ Микола Олександрович

Рецензент:
ВЛАСЕНКО Олег Віталійович

Житомир – 2025

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

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення

Тетяна Вакалюк

«14» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ДЗЮБЧУК Данило Романович**

Керівник роботи: **ФАНТ Микола Олександрович**

Тема роботи: **«Вебплатформа для організації та проведення навчальних курсів»**,

затверджена Наказом закладу вищої освіти від **«14» лютого 2025 р., №61с**

Вихідні дані для роботи: **використання Next.js для клієнтської частини та NestJS для серверної логіки для створення вебплатформи для організації та проведення навчальних курсів.**

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

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Фант М.О.	14.02.2025р.	14.02.2025р.
2	Фант М.О.	14.02.2025р.	14.02.2025р.
3	Фант М.О.	14.02.2025р.	14.02.2025р.

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра присвячена розробці «Вебплатформи для організації та проведення навчальних курсів».

Пояснювальна записка до кваліфікаційної роботи включає перелік умовних позначень, вступ, три розділи, висновки, список використаних джерел та додатки.

Текстова частина викладена на 62 сторінках друкованого тексту. Додатки займають 3 сторінки. Список використаних джерел містить 30 найменувань і займає 3 сторінки. У роботі подано 41 рисунків та 6 таблиць. Загальний обсяг документа – 68 сторінок.

У роботі було сформульовано технічні та функціональні вимоги до вебплатформи, проведено аналіз сучасних рішень у сфері освітніх систем, обґрунтовано доцільність створення власного рішення.

Розроблено багаторівневу архітектуру з використанням сучасного стеку технологій: Next.js, NestJS, MongoDB, WebSocket, Tailwind CSS, Socket.IO. Окрему увагу приділено реалізації системи навчальних матеріалів, досягнень, розкладу занять, файлів, ролей і механізмів доступу.

Виконано повну реалізацію бекенд-логіки, авторизації через JWT, реактивного інтерфейсу з типізованим управлінням станом, а також розгортання проекту на хмарних сервісах Vercel та Railway.

Система є повноцінним інструментом для проведення онлайн-курсів із підтримкою різних ролей користувачів (адміністратор, викладач, студент), керування контентом, комунікації та відстеження прогресу.

КЛЮЧОВІ СЛОВА: НАВЧАЛЬНА ПЛАТФОРМА, NEXT.JS, NESTJS, MONGODB, JWT, WYSIWYG, РОЛЬОВА МОДЕЛЬ, CRUD, WEB SOCKET, API.

					ІПЗ.КР.Б-121-25-ІПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	Вебплатформа для організації та проведення навчальних курсів	Літ.	Арк.	Аркуші
Розроб.		Д. ДЗЮБЧУК						
Керівник		М. ФАНТ					3	
н. контр.						Житомирська політехніка, група ВТ-21-1		
Зав. каф.								
					Пояснювальна записка			

ABSTRACT

The bachelor's qualification thesis is dedicated to the development of a web-based platform for organizing and conducting educational courses.

The explanatory report includes a list of abbreviations, an introduction, three chapters, conclusions, a list of references, and appendices.

The textual part spans 61 pages of printed content. The appendices cover 3 pages. The reference list includes 30 sources and takes up 3 pages. The work contains 41 figures and 6 tables, resulting in a total document length of 67 pages.

The thesis outlines the technical and functional requirements of the web platform, analyzes modern solutions in the field of educational systems, and substantiates the need for a custom-built system.

A multi-tier architecture was developed using a modern technology stack: Next.js, NestJS, MongoDB, WebSocket, Tailwind CSS, and Socket.IO. Special attention was given to implementing systems for educational content, achievements, lesson scheduling, file management, user roles, and access control mechanisms.

The backend logic was fully implemented, including JWT-based authentication, a reactive interface with typed state management, and cloud deployment via Vercel and Railway.

The resulting system is a comprehensive tool for conducting online courses, supporting multiple user roles (administrator, instructor, student), enabling effective content management, communication, and progress tracking.

KEYWORDS: EDUCATIONAL PLATFORM, NEXT.JS, NESTJS, MONGODB, JWT, WYSIWYG, ROLE-BASED ACCESS, CRUD, WEB SOCKET, API.

		Дзюбчук Д.Р.			ІІЗ.КР.Б-121-25-ІІЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

РЕФЕРАТ.....	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ВЕБПЛАТФОРМИ ДЛЯ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ НАВЧАЛЬНИХ КУРСІВ	10
1.1 Постановка задачі.....	10
1.2 Аналіз аналогів програмного продукту	11
1.3 Вибір архітектури застосунку	13
1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	15
Висновок до розділу 1.....	18
РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБПЛАТФОРМИ.....	20
2.1. Визначення варіантів використання і структури застосунку	20
2.2 Проектування бази даних системи	24
2.3. Проектування та реалізація алгоритмів роботи системи	28
2.4 Реалізація застосунку для вебплатформи для організації та проведення навчальних курсів.....	33
Висновки до розділу 2	35
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ЗАСТОСУНКОМ	37
3.1. Порядок встановлення та налаштування параметрів системи.	37
3.2 Структура інтерфейсу вебплатформи для організації та проведення навчальних курсів	40
3.3 Тестування роботи програмного комплексу	56
Висновки до розділу 3	58
ВИСНОВКИ	61

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	5
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 63

ДОДАТКИ..... 66

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

DB – база даних. Організована структура для зберігання, обробки та доступу до інформації.

CRUD – основні операції з даними в базі даних: створення (Create), читання (Read), оновлення (Update), видалення (Delete).

ORM – технологія, що дозволяє взаємодіяти з базою даних за допомогою об'єктно-орієнтованого підходу в програмуванні.

JWT – стандарт безпечної передачі підтвердження автентичності користувача між клієнтом та сервером у вигляді токена.

WYSIWYG – інтерфейс редагування контенту, в якому відображення відповідає кінцевому результату.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Актуальність теми. У цифрову епоху освітні процеси стрімко переходять у формат онлайн. Проте не всі наявні рішення відповідають сучасним вимогам інтерактивності, безпеки та зручності як для викладачів, так і для слухачів. Значна частина існуючих платформ має обмежену гнучкість, відсутність модульності або не враховує специфічні потреби організаторів курсів із практичним ухилом, а також залишає поза увагою аспекти управління результатами та прогресом студентів.

В умовах високого попиту на гнучкі інструменти для дистанційного навчання, розробка власної вебплатформи стає не лише технічно доцільною, але й суспільно важливою. Вона дозволяє створювати структуровані курси, проводити тестування, відстежувати досягнення, а також організовувати офлайн-активності із цифровим супроводом. Особливу увагу в реалізації було приділено адміністративному управлінню навчальним процесом, яке в більшості наявних рішень реалізоване поверхнево або надмірно складно.

Створення вебплатформи для організації та проведення навчальних курсів відповідає актуальним суспільним та технічним викликам. Вона покликана забезпечити зручність, ефективність, безпеку й повний контроль над навчальним процесом як для організаторів, так і для слухачів.

Метою цієї випускної кваліфікаційної роботи є створення зручної, безпечної та масштабованої вебплатформи для організації та проведення навчальних курсів. Такий програмний продукт покликаний забезпечити повний цикл навчання: від створення теоретичних матеріалів до контролю успішності. Розробка базується на практичному застосуванні фреймворку Next.js, типізованої мови TypeScript та бази даних MongoDB, що дозволяє поєднати сучасні підходи до frontend і backend-розробки в одному рішенні.

Задля досягнення поставленої мети у межах цієї роботи було поставлено низку практичних завдань. Зокрема, необхідно було:

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	8
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

- реалізувати повнофункціональну серверну частину, яка забезпечує автентифікацію користувачів, управління ролями доступу та взаємодію з базою даних;
- створити інтерактивний інтерфейс із використанням реактивного рендерингу, типізованого управління станом і забезпеченням швидкодії на стороні клієнта;
- розробити систему керування навчальними матеріалами, яка підтримує різні формати контенту (текст, відео, тести);
- впровадити механізми відстеження досягнень студентів та історії їхньої активності;
- реалізувати гнучке формування розкладу занять із можливістю керування з боку викладачів;
- забезпечити підтримку завантаження та зберігання навчальних файлів із відповідними правами доступу;

Очікуваним результатом роботи стане завершена та протестована платформа, яка не лише виконує заявлений функціонал, а й демонструє практичне втілення навичок, здобутих за час навчання. Цей проєкт дозволяє не просто перевірити технічні знання, а й відчувати себе розробником, який працює над реальним, корисним продуктом, здатним масштабуватись, змінюватись та ефективно розв'язувати прикладні задачі у сфері цифрової освіти.

Предметом дослідження є методи та інструменти створення сучасних веб-додатків для створення сучасних інтерактивних платформ навчання.

Об'єктом дослідження є комплексна реалізація вебплатформи для організації та проведення навчальних курсів.

Застосовані методи дослідження як системний аналіз, порівняльний аналіз, об'єктно-орієнтоване моделювання.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ВЕБПЛАТФОРМИ ДЛЯ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ НАВЧАЛЬНИХ КУРСІВ

1.1 Постановка задачі

Метою випускної роботи є розробка вебдодатку, що призначений для проведення навчальних курсів і забезпечує комплексну систему організації навчального процесу, управління користувачами, навчальними матеріалами та сертифікацією. Особливу увагу було приділено створенню зручного інтерфейсу, який відповідає сучасним вимогам інтуїтивності та простоти, що дозволяє користувачам легко орієнтуватися у платформі та ефективно взаємодіяти з усіма її функціями. Дизайн системи розроблений таким чином, щоб ключові сторінки, зокрема головна, особистий кабінет, система тестування, бібліотека матеріалів і адміністративна панель, були максимально зрозумілими і зручними для різних категорій користувачів.

Для створення клієнтської частини було обрано технологію Next.js разом з TypeScript, що забезпечує високу продуктивність, стабільність і масштабованість системи. Також для розробки інтерфейсних компонентів використана бібліотека ShadCN, яка поєднує в собі сучасний стиль оформлення з адаптивністю, дозволяючи підтримувати гарний вигляд на різних пристроях. Особливим інструментом для редагування навчальних матеріалів став TipTap – потужний і гнучкий редактор, який значно покращує комфорт авторів, одночасно обмежуючи витік контенту за межі платформи. Навігація і доступ до різних сторінок платформи налаштовані з урахуванням ролей користувачів, що гарантує правильний рівень доступу до платформи.

Для створення серверної частини було обрано технологію Nest.js, що забезпечує структуровану організацію API і високий рівень надійності системи. Для зберігання даних обрана база MongoDB, яка у поєднанні з Mongoose спрощує роботу з багатовимірними і складними структурами даних, а також гарантує масштабованість і швидкий доступ до інформації. Авторизація

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	10
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

користувачів реалізована за допомогою токенів, що забезпечує гнучкий контроль доступу і захист ресурсів платформи від несанкціонованого використання.

В результаті розробки була створена надійна платформа, яка не лише відповідає вимогам організації навчального процесу і контролю знань, але й демонструє практичну ефективність сучасних вебтехнологій у сфері цифрової освіти. Вона поєднує стабільність, гнучкість і готовність до подальшого розвитку, що є важливим для підтримки та масштабування проекту в майбутньому.

1.2 Аналіз аналогів програмного продукту

Moodle. Це одна з найпопулярніших і найдавніших систем дистанційного навчання з відкритим кодом. Вона добре підходить для масштабних освітніх установ, має широкі можливості для створення курсів, тестування, керування ролями і гнучке налаштування. Але при цьому її інтерфейс іноді виглядає застарілим і важким для користувачів, а також не підтримує сучасних реальних часів сповіщень і інтерактивності. Moodle вимагає досить потужної серверної інфраструктури та значних ресурсів на адміністрування. У порівнянні з моїм проектом, де пріоритет на простоту та сучасний UI, а також миттєве оновлення даних через вебсокети, Moodle може виглядати громіздкою і менш адаптованою для гнучких курсів із живим розкладом.

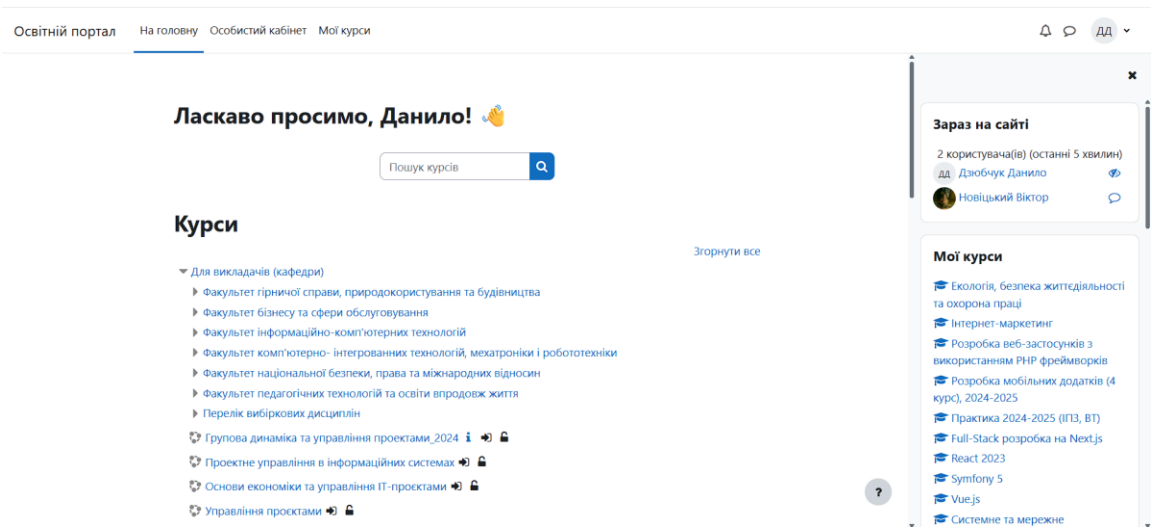


Рис. 1.1 – Moodle платформа на прикладі Освітнього Порталу ДУЖПІ

TalentLMS. Комерційна платформа з фокусом на швидкий старт і інтуїтивний дизайн. Вона має дуже зручний інтерфейс, вбудовані інструменти для створення курсів і звітності, а також підтримує інтеграції зі сторонніми сервісами. Водночас, TalentLMS орієнтована на більш стандартизовані сценарії, де реєстрація та доступ відбуваються автоматично, без особистої перевірки користувача адміністратором, що не підходить для моєї платформи, де важлива ручна верифікація і контроль доступу. Крім того, TalentLMS не має нативної системи сповіщень у реальному часі для інструкторів і студентів, що є важливим елементом у моєму рішенні.

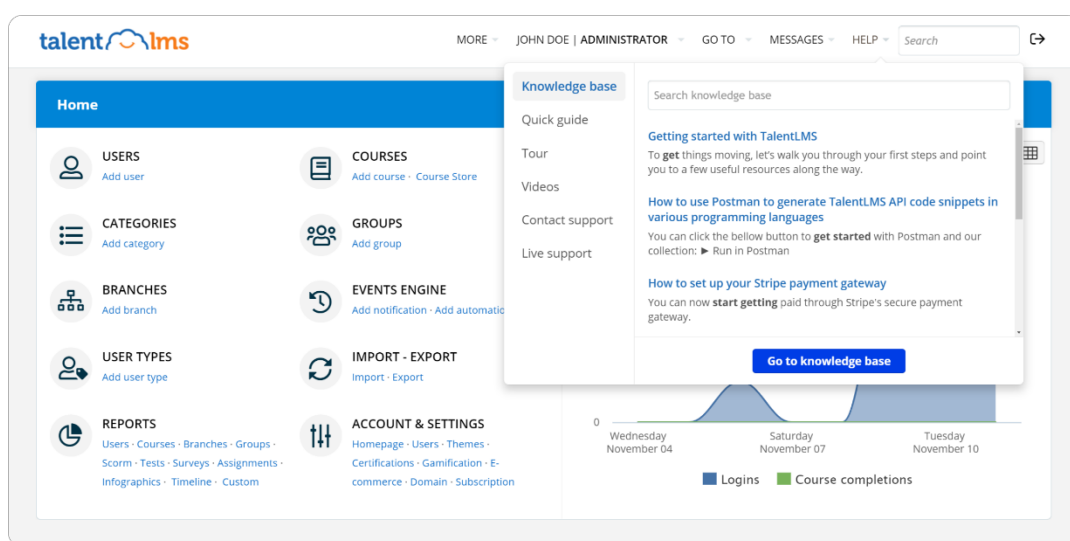


Рис. 1.2 – Приклад розгортки TalentLMS платформи

Google Classroom. Продукт великої компанії, орієнтований на освітні заклади та інтегрований з іншими сервісами Google. Його перевага – простота використання і зручність для вчителів та учнів, які вже працюють в екосистемі Google. Але ця платформа не дає глибокого контролю над контентом і не передбачає кастомізації під специфічні потреби окремих організацій. Також, Google Classroom не має складної системи управління розкладом із можливістю підписки на конкретні заняття, а весь процес реєстрації й доступу базується на акаунтах Google, що не дає можливості контролювати користувачів через особисту перевірку.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	12
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

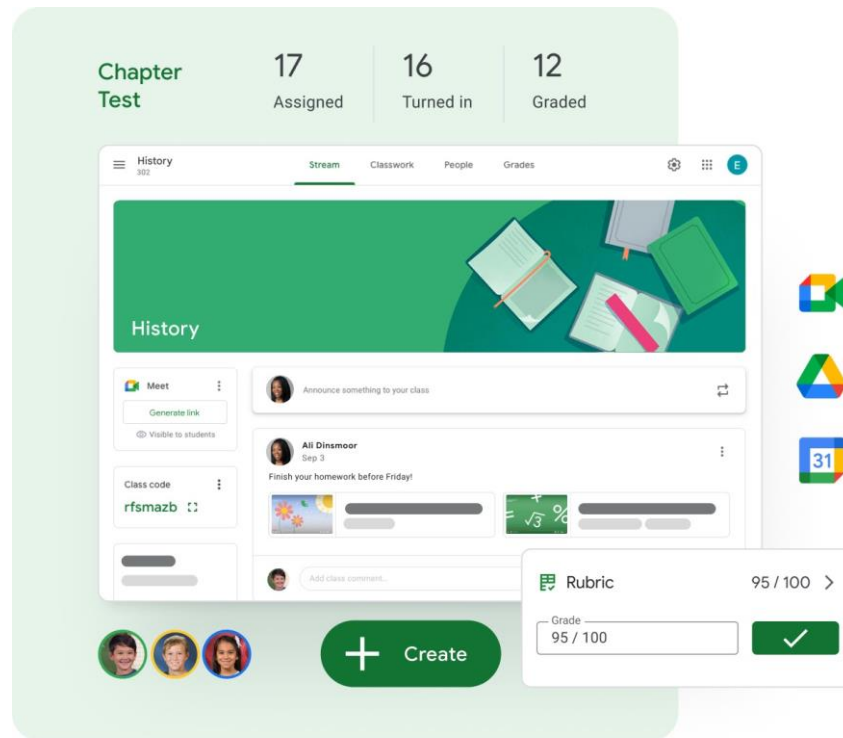


Рис. 1.3 – приклад створеної групи в Google Classroom

Таким чином, порівнюючи свій проєкт із цими трьома платформами, можна чітко показати, що він поєднує сучасний інтерфейс і інтерактивність (websocket-сповіщення), забезпечує гнучкий менеджмент розкладу із підпискою на заняття, а також унікальну систему ручної верифікації користувачів через адміністратора.

1.3 Вибір архітектури застосунку

Для реалізації вебплатформи було обрано сучасну багаторівневу архітектуру, яка дозволяє ефективно поєднати швидкодію, масштабованість і зручність розробки. Враховуючи вимоги до платформи – підтримка кількох ролей, система розкладів, досягнень, управління матеріалами та інші освітні функції – класична схема клієнт-сервер була доповнена новітніми підходами до інтерактивності та реального часу.

Основу архітектури становить фреймворк Next.js, який поєднує можливості серверного рендерингу (SSR) та статичної генерації (SSG) з класичним підходом до побудови SPA. Завдяки цьому застосунок демонструє хорошу швидкодію навіть за умов великого обсягу контенту. SSR забезпечує швидке завантаження сторінок та поліпшену SEO-оптимізацію, а можливість

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	13
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

гібридного рендерингу дозволяє розділити відповідальність між клієнтом і сервером.

Серверну частину було реалізовано за допомогою NestJS – прогресивного Node.js фреймворку, побудованого на TypeScript. NestJS дозволив реалізувати модульну структуру проєкту з розмежуванням відповідальностей між контролерами, сервісами, модулями та схемами.

Для зберігання даних обрано документоорієнтовану базу даних MongoDB, яка дозволяє гнучко працювати з вкладеними структурами та швидко масштабувати систему горизонтально. Через використання бібліотеки Mongoose реалізовано схеми з валідацією даних, зв'язками між моделями (наприклад, студент і його досягнення), а також обробку хук-подій (наприклад, створення токена після реєстрації).

Ключовим елементом архітектури є автентифікація з використанням JWT (JSON Web Token), яка дозволяє зберігати інформацію про користувача в токені без необхідності утримувати сесію на сервері. Токен передається у вигляді httpOnly cookie, що забезпечує захист від XSS-атак. Для контролю доступу реалізовані гварди (guards), які перевіряють роль користувача перед виконанням запиту. Це дозволяє гнучко реалізувати рольову модель (адміністратор, інструктор, студент) і забезпечити безпечний доступ до функціоналу.

Особливу увагу приділено інтерактивності. Для реалізації комунікації в реальному часі використано WebSocket через Socket.IO, що дозволяє відображати оновлення (наприклад, нові повідомлення або зміни в розкладі занять) без перезавантаження сторінки. Це критично важливо в контексті живих курсів, консультацій або практичних занять із зворотним зв'язком.

Інтерфейс платформи реалізовано з використанням Tailwind CSS, що дало змогу швидко розробити адаптивний дизайн з урахуванням зручності на різних пристроях. Управління станом застосунку здійснюється через кастомні хуки та контексти React, а також бібліотеку React Query – це дозволяє забезпечити реактивність інтерфейсу та синхронізацію даних між клієнтом і сервером.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	14
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Файли (презентації, додатки до занять, PDF, відео тощо) обробляються через окремий модуль FileModule, який керує завантаженням, зберіганням і наданням доступу до ресурсів. Це дозволяє централізовано керувати бібліотекою матеріалів та дотримуватись політик безпеки.

Усі модулі, такі як система досягнень, матеріали, тести, розклад, бібліотека, комунікації – реалізовані із дотриманням принципу single responsibility. Завдяки цьому платформа залишається легко розширюваною, а окремі її частини можуть бути оновлені або вдосконалені без впливу на решту системи.

Схематичне зображення архітектури платформи наведено на рисунку 1.4 – воно ілюструє взаємодію основних модулів системи: клієнтської частини, реалізованої на базі Next.js, серверної логіки у середовищі NestJS та бази даних MongoDB.

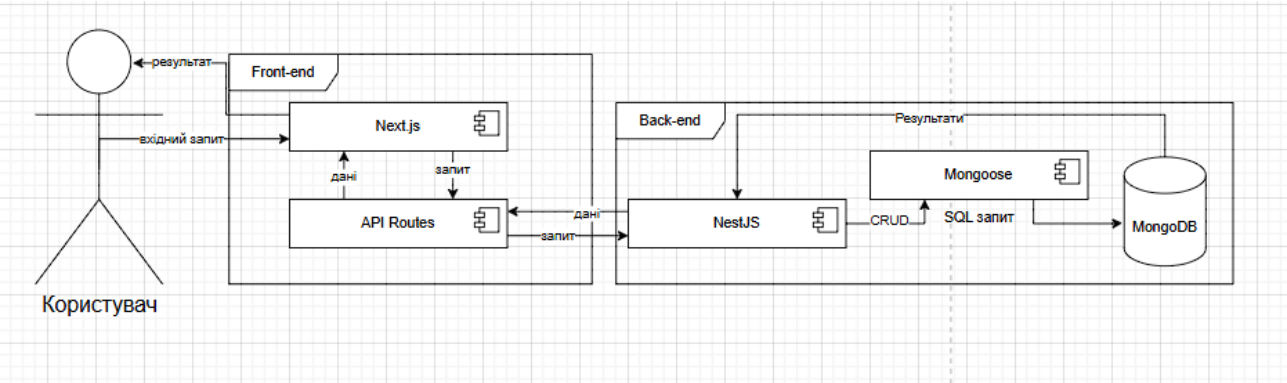


Рис. 1.4 – Архітектура застосунку

Уся ця інфраструктура розгорнута в хмарному середовищі: Vercel відповідає за фронтенд, Railway – за бекенд і базу даних. Такий підхід забезпечує автоматизовану CI/CD-інтеграцію, стабільність роботи та готовність до горизонтального масштабування при зростанні навантаження.

1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Для реалізації вебдодатку було обрано архітектурний підхід, який передбачає чітке розділення клієнтської та серверної частин із визначеними зонами відповідальності, що дозволяє ефективно керувати розробкою та

масштабуванням системи. Клієнтська частина побудована на основі Next.js із застосуванням TypeScript, що забезпечує типізацію коду, підвищує надійність та якість розробки, а також дозволяє створювати швидкий та зручний інтерфейс користувача. Використання Next.js дає змогу поєднувати серверний рендеринг з рендерингом на боці клієнта, що значно покращує швидкість завантаження сторінок, оптимізує SEO та підвищує загальну продуктивність платформи. Для створення інтерфейсних компонентів застосовується бібліотека ShadCN, яка забезпечує адаптивний дизайн і сучасний зовнішній вигляд, що сприяє підвищенню комфортності користування додатком на різних пристроях і дозволяє легко підтримувати стильову цілісність візуальної частини.

Побудована структура враховує ролі користувачів і забезпечує персоналізований доступ до функціоналу. Це дозволяє не лише підвищити безпеку, але й оптимізувати користувацький досвід, направляючи користувачів лише до необхідних для них розділів платформи. Логіка взаємодії реалізована через REST API, що є гнучким і масштабованим рішенням для побудови зв'язку між клієнтом і сервером.

Серверна частина побудована на базі Nest.js, прогресивного фреймворку для Node.js, який дозволяє ефективно організувати структуру API, інтегрувати складну бізнес-логіку та забезпечувати масштабованість системи. Завдяки модульній архітектурі Nest.js розробники можуть легко підтримувати й розширювати додаток, вносячи зміни в окремі компоненти без ризику порушення цілісності системи. Збереження даних відбувається у MongoDB – потужній NoSQL базі даних, що забезпечує гнучкість роботи зі складними структурами та високий рівень доступності інформації. Для зручності та безпеки взаємодії з базою застосовується ORM Mongoose, що спрощує моделювання даних і реалізацію валідованих операцій із колекціями.

Для управління доступом і автентифікації користувачів використовується система на основі токенів, яка дозволяє гнучко налаштовувати права доступу відповідно до ролей користувачів і гарантує безпеку персональних даних та інформації платформи. Цей підхід підвищує захищеність системи, мінімізує

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

ризика несанкціонованого доступу і забезпечує безперебійну роботу в умовах багатокористувацького середовища. З метою забезпечення реального часу у функціональності платформи (наприклад, повідомлення, зміни у розкладі, живе спілкування з інструкторами) впроваджено підтримку WebSocket-з'єднання на базі бібліотеки Socket.IO. Це дозволяє забезпечити двосторонню комунікацію між клієнтом і сервером без необхідності періодичних опитувань (polling), що значно знижує навантаження на серверну частину. Вебсокети активно використовуються для реалізації системи оповіщень, які інформують користувачів про нові заняття, оновлення матеріалів чи досягнень. Такий підхід також покращує інтерактивність і створює ефект «живої» платформи, що є особливо важливим для підтримки онлайн-навчання з елементами реального часу.

Для організації зручного і захищеного процесу розгортання та підтримки платформи обрано хмарні сервіси Vercel (для клієнтської частини) та Railway (для серверної логіки та бази даних). Ці сервіси надають автоматизований CI/CD, дозволяють швидко масштабувати середовище в разі підвищеного навантаження та забезпечують стабільність роботи додатку навіть під час оновлень. Завдяки вбудованим моніторинговим інструментам, система може вчасно реагувати на помилки, аналізувати логи та приймати рішення щодо оптимізації продуктивності. Обрана інфраструктура також дозволяє безперешкодно додавати нові сервіси, такі як зовнішня аналітика або сервіси перевірки тестів, що відкриває перспективу подальшого розвитку платформи.

Взаємодія між клієнтською та серверною частинами відбувається через REST API, розроблені відповідно до сучасних стандартів. При отриманні запиту сервер проводить верифікацію прав доступу користувача, виконує бізнес-логіку та формує відповідь, яка використовується клієнтом для оновлення інтерфейсу. Така архітектура сприяє прозорості роботи системи, дозволяє швидко виявляти та усувати помилки, а також легко інтегрувати нові функції. Вона також забезпечує можливість масштабування платформи в майбутньому, що важливо для підтримки розвитку і адаптації до нових вимог користувачів і бізнесу.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	17
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдяки розділенню обов'язків між клієнтом і сервером, а також використанню сучасних технологій, система має високу надійність, гнучкість та ефективність, що є ключовими факторами успіху освітнього вебдодатку.

Висновок до розділу 1

При розробці вебплатформи для організації та проведення навчальних курсів було прийнято рішення використати сучасний технологічний стек, який забезпечує ефективність, масштабованість і надійність системи. Клієнтська частина побудована на основі Next.js та TypeScript, що дозволяє поєднувати переваги серверного рендерингу з гнучкістю реактивних компонентів. Такий підхід гарантує швидке завантаження сторінок і добру SEO-оптимізацію, що важливо для залучення користувачів та покращення видимості платформи в пошукових системах.

Важливим аспектом є використання бібліотеки ShadCN для створення інтерфейсних компонентів. Вона надає широкий набір адаптивних та кастомізованих елементів, що значно прискорює розробку та забезпечує єдиний стиль у всьому додатку. Застосування цієї бібліотеки сприяє покращенню користувацького досвіду, адже компоненти коректно відображаються на різних пристроях і мають сучасний вигляд.

Організація навігації і маршрутизації враховує розмежування доступу відповідно до ролей користувачів, що забезпечує безпеку і персоналізований досвід. Система дозволяє студентам переглядати навчальні матеріали та проходити тести, тоді як інструктори отримують інструменти для управління курсами, матеріалами і контролю прогресу учнів. Адміністратори, у свою чергу, мають повний доступ до всіх функцій платформи для налаштування і підтримки системи.

Серверна частина реалізована на Nest.js, що забезпечує модульність і структурованість коду. Використання цього фреймворку дозволяє легко масштабувати додаток і інтегрувати нові функціональні можливості без значного переписування існуючого коду. Для роботи з базою даних обрано MongoDB.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

Авторизація користувачів базується на використанні токенів, що дозволяє гнучко контролювати доступ до різних частин платформи та захищати дані від несанкціонованого доступу. Такий підхід підтримує одночасну роботу багатьох користувачів, гарантує безпеку особистої інформації і полегшує інтеграцію з зовнішніми системами, якщо це буде необхідно у майбутньому.

Для створення та редагування навчальних матеріалів застосовано TipTap – потужний та гнучкий редактор, що дозволяє авторам зручно формувати контент різного типу, включаючи текст, зображення та вбудовані медіа. Використання TipTap мінімізує ризики втрати інформації і сприяє стандартизації форматів контенту, що підвищує якість навчальних курсів.

Окрім основної архітектури, платформа включає модуль WebSocket–комунікацій, реалізований через Socket.IO, який дозволяє підтримувати миттєві повідомлення між користувачами, оновлення статусів та взаємодію в реальному часі. Це є критично важливим для інтерактивних елементів, таких як онлайн–консультації, живі анонси занять та сповіщення. Така інтерактивність надає платформі перевагу над класичними статичними рішеннями у сфері онлайн–освіти.

Таким чином, технологічні рішення, що були обрані на етапі проектування, дозволили сформувати надійний і сучасний фундамент для повноцінної навчальної платформи. Завдяки використанню перевірених фреймворків, модульній архітектурі та гнучкій системі управління ролями і доступом, платформа може легко адаптуватися до потреб різних освітніх проєктів, розширюватись та інтегруватись з іншими сервісами без зміни основної логіки.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	19
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБПЛАТФОРМИ

2.1. Визначення варіантів використання і структури застосунку

Розроблена система являє собою вебплатформу для організації, проведення та управління навчальними курсами, що забезпечує взаємодію користувачів із різними ролями: студентів, інструкторів та адміністраторів. Для наочного представлення основних учасників та їх взаємодії із системою було створено діаграму варіантів використання (рисунок 2.1).



Рисунок 2.1 – Діаграма варіантів використання для неавторизованого користувача

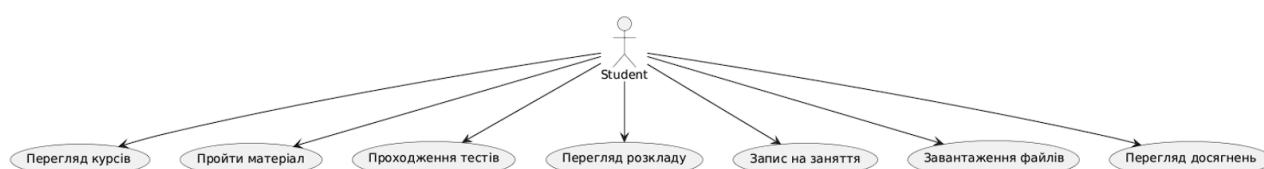


Рисунок 2.2 – Діаграма варіантів використання для студента

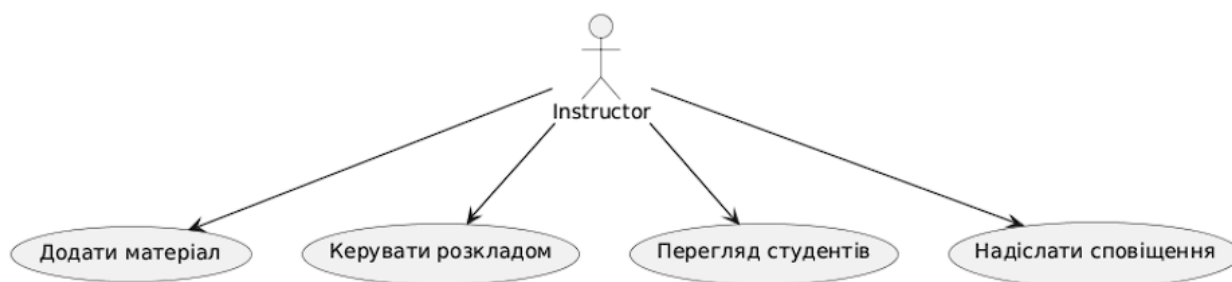


Рисунок 2.3 – Діаграма варіантів використання для інструктора

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	20
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

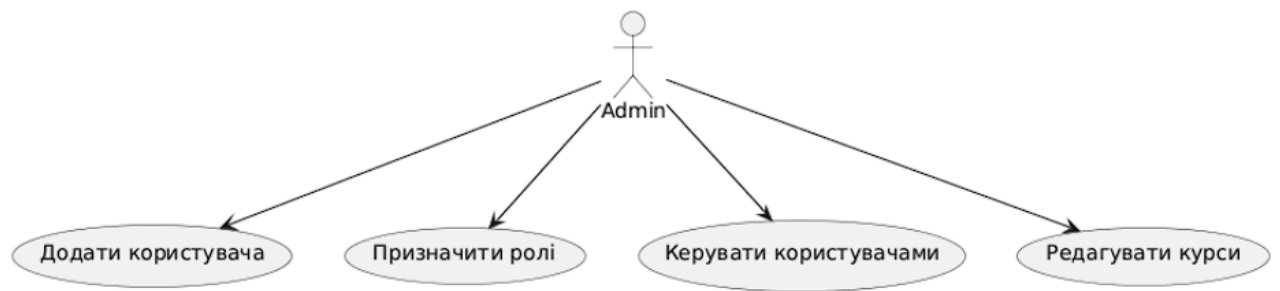


Рисунок 2.4 – Діаграма варіантів використання для адміністратора

Діаграма варіантів використання включає три основні типи акторів:

Студенти – основні користувачі платформи, які реєструються для проходження курсів, вивчення навчальних матеріалів та складання тестів з подальшим контролем власного прогресу.

Інструктори – користувачі з розширеними правами, відповідальні за створення, оновлення та управління курсами, формування навчальних матеріалів, а також оцінювання та підтримку студентів.

Адміністратори – користувачі з повним доступом, які відповідають за підтримку платформи, управління користувачами, контроль за якістю контенту та адміністрування системи.

Функціональні вимоги студентів полягають у забезпеченні можливості реєстрації за допомогою електронної пошти та паролю або через сторонні сервіси автентифікації, що гарантує швидкий і безпечний доступ до системи. Після авторизації користувачі отримують доступ до персонального кабінету, де можуть переглядати доступні курси, записуватися на них та відслідковувати власний прогрес. Платформа передбачає зручні інструменти для взаємодії з навчальними матеріалами в різних форматах: текстовому, відео та презентаційному, що сприяє ефективному засвоєнню знань. Для оцінювання рівня знань реалізовано систему тестування, яка підтримує множинні типи запитань, включно з множинним вибором, відкритими відповідями та завданнями на відповідність. Результати тестування зберігаються в особистому профілі студента. Окрім цього, студенти мають можливість ставити запитання, залишати відгуки та коментувати матеріали, що підвищує рівень інтерактивності навчання. Особисті дані студентів зберігаються відповідно до чинних вимог

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	21
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

захисту інформації; доступ до них мають лише самі користувачі та уповноважені адміністратори системи. Вебплатформа надає інструменти для перегляду навчальних матеріалів у різних форматах (текст, відео, презентації), що підтримує різні стилі навчання та сприяє кращому засвоєнню інформації.

Вимоги інструкторів передбачають наявність повного набору функціоналу для створення та редагування навчальних курсів. Інструктори можуть завантажувати навчальні матеріали, структурувати зміст курсу, формувати розклад занять і тестів. Система підтримує управління навчальними групами, що дозволяє відстежувати відвідуваність, оцінки та виконання завдань кожним студентом. Для аналітики реалізовано інструменти моніторингу ефективності навчального процесу: інструктор має змогу переглядати статистику успішності, коригувати зміст курсу та вчасно реагувати на труднощі учнів. Комунікація з користувачами здійснюється через внутрішню систему повідомлень і коментарів. Додатково передбачена можливість сертифікації студентів та формування відповідної документації згідно з освітніми стандартами.

Вимоги адміністраторів зосереджені на повному контролі над системою. Адміністратори мають змогу управляти користувачами – здійснювати їх реєстрацію, зміну ролей, блокування облікових записів. Також вони контролюють весь контент платформи та мають доступ до модерації: видалення неприйнятних матеріалів, забезпечення дотримання правил користування. Адміністративна панель надає змогу керувати налаштуваннями системи, включаючи параметри безпеки, резервне копіювання бази даних і моніторинг стабільності роботи. Адміністраторам доступна розширена статистика, яка охоплює кількість активних користувачів, завантажені матеріали, популярність курсів тощо. В системі реалізовано механізм аудиту дій усіх користувачів, що гарантує прозорість і підвищує рівень захищеності.

Бізнес-вимоги полягають у відповідності вебплатформи сучасним стандартам безпеки та конфіденційності, що передбачає захист персональних даних відповідно до чинного законодавства. Важливою умовою є наявність

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

інтуїтивно зрозумілого інтерфейсу, який забезпечує швидке освоєння функціоналу та мінімізує потребу у технічній підтримці. Крім того, система має бути сумісною з зовнішніми сервісами, зокрема для автентифікації, аналітики та інших інтеграцій. Особливу увагу приділено адаптивності дизайну, що гарантує коректне відображення інтерфейсу на пристроях з різною роздільною здатністю.

Функціональні характеристики системи включають захищену авторизацію користувачів та гнучку систему розмежування доступу на основі ролей. Застосування Next.js із серверним рендерингом і мови програмування TypeScript забезпечує високу продуктивність і масштабованість клієнтської частини. Для керування станом застосовано сучасні підходи централізованого збереження даних, що гарантує швидкий відгук інтерфейсу на дії користувача. Навігація оптимізована відповідно до ролі користувача, що забезпечує персоналізований і безпечний досвід. Вбудований WYSIWYG-редактор дозволяє створювати навчальні матеріали з мультимедійним контентом. Система тестування автоматизує процес оцінювання, підтримує різні типи завдань і зберігає історію результатів.

Нефункціональні вимоги визначають необхідність забезпечення часу відповіді на основні запити в межах 2 секунд при стабільності роботи не менше ніж 99,7%. Застосовано сучасні засоби захисту від атак типу SQL-ін'єкцій, XSS і CSRF, а також механізми шифрування даних при зберіганні й передачі. База даних автоматично резервується щодня, із можливістю відновлення за останні 30 днів. Архітектура системи орієнтована на горизонтальне масштабування, що дозволяє без суттєвих змін інфраструктури обслуговувати зростаючу кількість користувачів.

Архітектура системи реалізована за клієнт-серверною архітектурою з розподілом відповідальності між фронтендом і бекендом.

Клієнтська частина побудована на Next.js з використанням React-компонентів та TypeScript, що забезпечує швидке рендеринг сторінок і високу продуктивність. Центральний менеджмент стану здійснюється через Redux або

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	23
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

інший сучасний інструмент, що дозволяє підтримувати консистентність даних у всьому додатку.

Серверна частина розроблена з використанням Nest.js, що надає модульність, високий рівень організації коду та можливість масштабування. API реалізоване у вигляді REST або GraphQL, що забезпечує гнучкість інтеграції з клієнтською частиною.

База даних обрана на основі MongoDB з Mongoose як ODM, що дає змогу зручно працювати зі складними структурами даних, типами документів та відносинами між ними.

Для забезпечення безпеки застосовано **JWT-токени** для авторизації, які гарантують контроль доступу та захист персональних даних.

Резервування, моніторинг та логування виконуються через інтегровані сервіси хостингу, що забезпечують стабільність і швидке реагування на інциденти.

2.2 Проектування бази даних системи

Для зберігання інформації у застосунку було обрано MongoDB – NoSQL базу даних, яка працює з документами замість традиційних таблиць.

База даних складається з 5 основних колекцій, кожна з яких виконує певну функцію у системі:

Таблиця 2.1.

Призначення колекцій бази даних

Назва колекції	Призначення
users	Зберігання профілів користувачів, налаштувань та метаданих
achievements	Зберігання інформації про досягнення користувачів
schedules	Планування та збереження інформації про заняття і їх розклад
materials	Зберігання навчальних матеріалів (статті, відео, лекції)

files	Зберігання інформації про завантажені файли користувачами
-------	---

Структуру бази даних представлено на рисунку 2.5, де відображено 5 основних колекцій документів з відповідно визначеними зв'язками. Вся система продумана та оптимізована для безпечного зберігання різноманітних даних, що виникають у процесі взаємодії користувачів із застосунком, із розподілом інформації на колекції відповідно до типу даних, які необхідно зберігати.

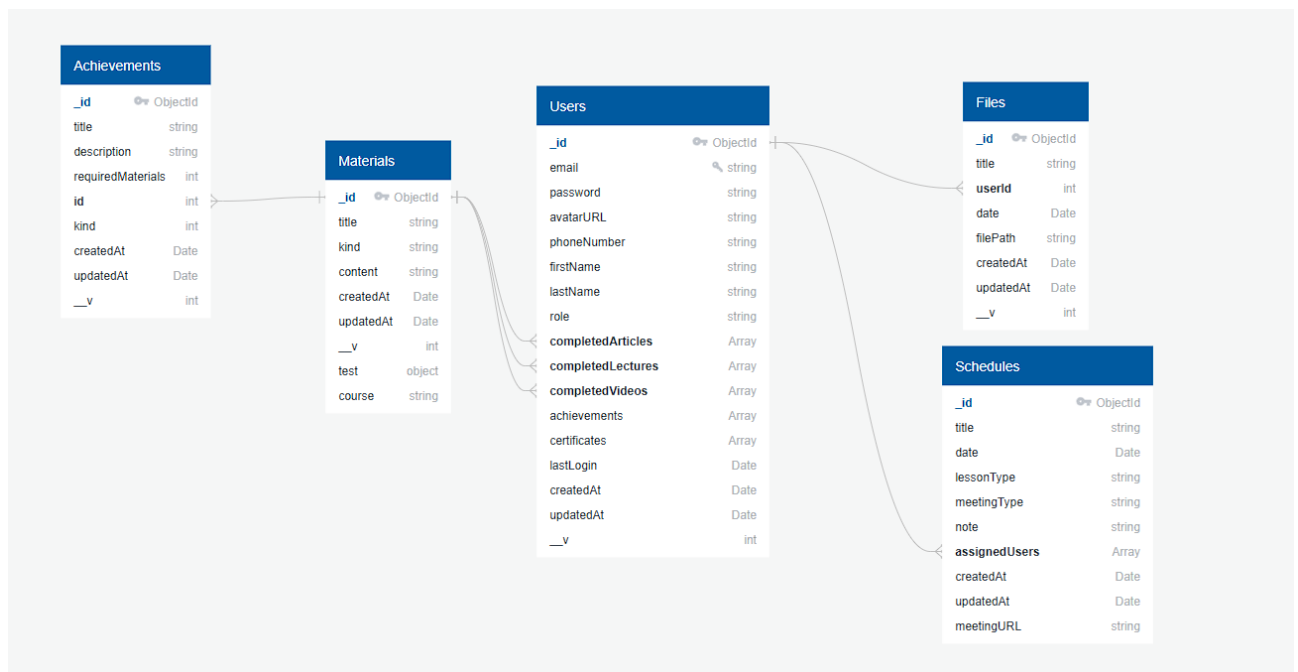


Рис. 2.5 – Діаграма бази даних

Таблиця 2.2

Колекція Users

Поле	Тип	Призначення
_id	string (ObjectId)	Унікальний ідентифікатор користувача
email	string	Електронна адреса
password	string	Хеш паролю
avatarURL	string	URL аватарки
phoneNumber	string	Номер телефону
firstName	string	Ім'я
lastName	string	Прізвище

Продовження таблиці 2.2

role	string	Роль користувача (наприклад, "user", "admin")
completedArticles	Array	Ідентифікатори прочитаних статей (Materials._id)
completedLectures	Array	Ідентифікатори переглянутих лекцій (Materials._id)
completedVideos	Array	Ідентифікатори переглянутих відео (Materials._id)
achievements	Array	Масив досягнень
certificates	Array	Масив сертифікатів
lastLogin	string (Date ISO)	Дата останнього входу
createdAt	string (Date ISO)	Дата створення запису
updatedAt	string (Date ISO)	Дата останнього оновлення
__v	number	Внутрішнє поле MongoDB

Колекція **Users** відповідає за збереження даних про користувача, його особисту інформацію, ролі та пройдені матеріали. Унікальний ідентифікатор `_id` використовується для формування зв'язків з іншими колекціями. Поля `email`, `password`, `firstName` та `lastName` містять інформацію для аутентифікації та персоналізації. Поле `role` визначає рівень доступу користувача. Масиви `completedArticles`, `completedLectures`, `completedVideos` містять посилання на пройдені матеріали, а поля `achievements` і `certificates` – на здобуті досягнення та сертифікати.

Таблиця 2.3

Колекція Files

Поле	Тип	Призначення
<code>_id</code>	string (ObjectId)	Унікальний ідентифікатор
<code>title</code>	string	Назва файлу
<code>userId</code>	string	Посилання на користувача (Users._id)
<code>date</code>	string (Date ISO)	Дата завантаження файлу
<code>filePath</code>	string	Шлях до файлу
<code>createdAt</code>	string (Date ISO)	Дата створення
<code>updatedAt</code>	string (Date ISO)	Дата оновлення
<code>__v</code>	number	Внутрішнє поле MongoDB

Колекція **Files** відповідає за зберігання інформації про файли, які завантажують користувачі. Документи містять унікальний `_id`, заголовок `title`,

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	26
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

посилання на власника через `userId`, дату завантаження `date`, шлях до файлу `filePath`, а також дату створення і оновлення запису.

Таблиця 2.4

Колекція **Materials**

Поле	Тип	Призначення
<code>_id</code>	string (ObjectId)	Унікальний ідентифікатор
<code>title</code>	string	Назва матеріалу
<code>kind</code>	string	Тип матеріалу (Article, Video, Lecture тощо)
<code>content</code>	string	Вміст матеріалу (текст, посилання і т.п.)
<code>test</code>	object	Дані тесту, якщо є
<code>course</code>	string	Назва курсу, до якого належить матеріал
<code>createdAt</code>	string (Date ISO)	Дата створення
<code>updatedAt</code>	string (Date ISO)	Дата оновлення
<code>__v</code>	number	Внутрішнє поле MongoDB

Колекція **Materials** зберігає навчальні матеріали різних типів, таких як статті, відео, лекції. Кожен документ має унікальний ідентифікатор `_id`, заголовок `title`, тип `kind`, основний зміст у полі `content`, а також інформацію про дату створення і останнього оновлення. Поле `test` містить пов’язані тести, а `course` – належність до певного курсу.

Таблиця 2.5

Колекція **Schedules**

Поле	Тип	Призначення
<code>_id</code>	string (ObjectId)	Унікальний ідентифікатор
<code>title</code>	string	Назва заняття
<code>date</code>	string (Date ISO)	Дата та час заняття
<code>lessonType</code>	string	Тип заняття (Lecture, Workshop тощо)
<code>meetingType</code>	string	Формат заняття (Online, Offline)
<code>note</code>	string	Додаткові примітки
<code>assignedUsers</code>	Array	Масив користувачів (Users._id), призначених на заняття
<code>meetingURL</code>	string	URL для онлайн зустрічі
<code>createdAt</code>	string (Date ISO)	Дата створення
<code>updatedAt</code>	string (Date ISO)	Дата оновлення

Колекція **Schedules** відповідає за організацію розкладу навчальних занять. Документи містять унікальний `_id`, заголовок `title`, дату `date`, тип уроку

lessonType (наприклад, лекція або практичне заняття), тип проведення meetingType (онлайн чи офлайн), а також URL для онлайн-зустрічей. Поле assignedUsers містить список користувачів, призначених на конкретне заняття.

Таблиця 2.6

Колекція Achievements

Поле	Тип	Призначення
_id	string (ObjectId)	Унікальний ідентифікатор
title	string	Назва досягнення
description	string	Опис досягнення
requiredMaterials	Array<{ id: string, kind: string }>	Масив обов'язкових матеріалів (id – посилання на Materials._id, kind – тип матеріалу)
createdAt	string (Date ISO)	Дата створення
updatedAt	string (Date ISO)	Дата оновлення
__v	number	Внутрішнє поле MongoDB

Колекція **Achievements** зберігає інформацію про досягнення користувачів. Кожне досягнення має унікальний _id, назву title, опис description та масив requiredMaterials – перелік матеріалів, які необхідно пройти для отримання досягнення, із зазначенням їхніх ідентифікаторів та типів. Поля createdAt і updatedAt відображають час створення і оновлення запису.

2.3. Проєктування та реалізація алгоритмів роботи системи

Вебплатформа для організації та проведення навчальних курсів – це складна екосистема, яка поєднує багато різних модулів і компонентів, але в ідеалі має працювати як єдине організоване ціле, де кожен елемент логічно впливає з іншого. Головна мета платформи – забезпечити не лише зручний доступ до навчальних матеріалів, а й ефективний контроль прогресу користувачів, управління правами доступу та інтерактивність.

Починаючи зі системи **автентифікації** – вона лежить в основі безпеки платформи. Тут не просто перевірка логіна і пароля, а комплексний механізм із використанням **JWT-токенів**, що передаються клієнту та перевіряються на кожному запиті. Використання токенів, а не сесійних куків, дає гнучкість у масштабуванні сервісу і дозволяє уникнути стану на сервері, що знижує навантаження. Важливо, що токени зберігаються в httpOnly cookie для захисту

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	28
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

від **XSS** атак, а також перевіряються через стратегії Passport.js – це гарантує, що кожен користувач справді має право виконувати дії, які він намагається здійснити.

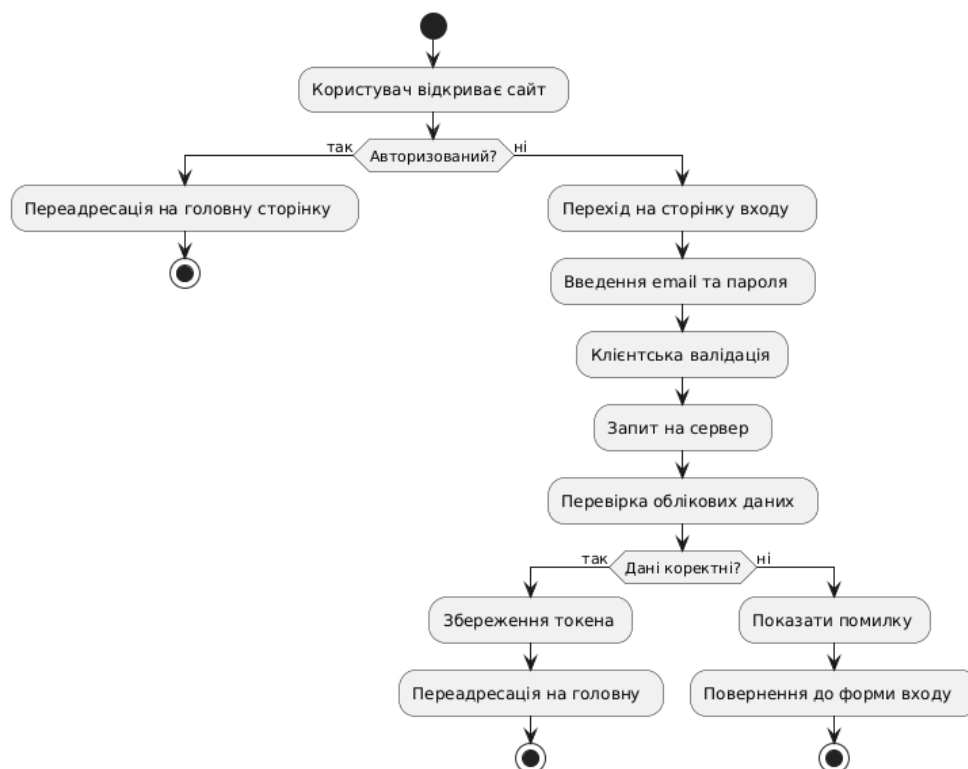


Рис. 2.6 – Діаграма діяльності реєстрації та авторизації користувача

У цьому контексті дуже важливим є застосування **гвардів** (guards), які діють як надійні ворота між користувачем і внутрішніми ресурсами платформи. Вони аналізують JWT, перевіряють роль користувача і дозволяють або блокують доступ до певних маршрутів. Наприклад, ролі Admin, Instructor і Student визначають, що саме користувач може бачити або редагувати. Завдяки цьому, критично важливі дії, як створення чи видалення досягнень або матеріалів, залишаються у руках уповноважених осіб. Це не middleware, бо middleware в NestJS зазвичай працює на рівні запиту загалом, тоді як guards працюють точково для захисту конкретних маршрутів, забезпечуючи більш гнучке і рольове управління доступом.

Самі **матеріали** у платформі є серцем навчального процесу – це статті, відео, тести, лекції, які розділені на категорії для зручності пошуку і структурованого вивчення. Система дозволяє інструкторам створювати нові матеріали, редагувати їх та прив’язувати до конкретних курсів чи досягнень. Це

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	29
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечує логічний зв'язок між тим, що користувач вивчає, і тим, що він повинен опанувати для успішного завершення курсу

Поряд із матеріалами існують **досягнення** (achievements) – це система мотивації і визнання, яка стимулює студентів до активної участі. Досягнення можуть бути простими, як завершення певного блоку матеріалів, або складними, наприклад, виконанням практичних завдань чи тестів. Вони зберігаються в базі і мають категорії, що дозволяє структурувати їх за темами або рівнями складності. Роль досягнень – створити відчуття прогресу і успіху, що значно підвищує залученість користувачів.

Щоб цей процес відбувався організовано, існує **розклад** (schedules) – модуль, який управляє датами проведення занять та інших активностей. Розклад дозволяє як студентам, так і інструкторам планувати свій час, відстежувати майбутні події і не пропускати важливі етапи навчання. Він також пов'язаний із матеріалами та досягненнями – наприклад, після завершення заняття відкривається доступ до відповідного матеріалу або тесту.

Усе це потребує інтерактивності, особливо коли йдеться про реальні тренінги чи консультації в режимі реального часу. Саме тому платформа використовує **WebSocket**, що дозволяє забезпечити двонаправлену комунікацію між клієнтом і сервером. Через WebSocket можна надсилати миттєві повідомлення, оновлення статусу занять, або навіть організовувати live-чати між студентами і викладачами. Це критично для підтримки живого зв'язку, який неможливо реалізувати через традиційні HTTP-запити.

Не менш важливим є модуль **бібліотеки** (library) – це місце, де зберігаються всі освітні ресурси, доступні для вільного перегляду або під час проходження курсів. Це дозволяє централізовано управляти великим обсягом інформації, роблячи її доступною і зручною для користувачів.

Окремо варто згадати про реалізацію розгортання платформи. Проєкт був опублікований на хмарних сервісах Vercel (для frontend-частини) та Railway (для backend-сервера), що дозволило забезпечити високу доступність, автоматичні оновлення та масштабованість без складної інфраструктури. Такий підхід

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	30
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволив зосередитись на розробці функціоналу, не витрачаючи час на ручне налаштування серверів. Інтеграція з Git дозволяє кожне оновлення автоматично доставляти на продакшн, а логування та моніторинг спрощують відстеження помилок і стабільність системи. Це підтверджує, що платформа не лише добре спроектована з погляду логіки роботи, але й ефективно впроваджена у реальних умовах.

Взаємодія між усіма цими модулями підтримується за рахунок чіткої архітектурної логіки, де ролі користувачів – це фундаментальний принцип контролю доступу. Кожен запит проходить перевірку через **систему гвардів**, а токени гарантують, що користувач є тим, за кого себе видає. Завдяки такому підходу, платформа забезпечує не лише функціональність, а й безпеку.

Отже, якщо підсумувати, твоя вебплатформа – це складний механізм, де кожен компонент виконує свою роль у єдиній системі: від автентифікації користувача з ролями, через захист доступу до ресурсів гвардами, організацію навчальних матеріалів та розкладу, мотивацію через систему досягнень, і до забезпечення інтерактивності за допомогою WebSocket. Всі ці елементи працюють разом, щоб навчальний процес був максимально ефективним, зручним і безпечним.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

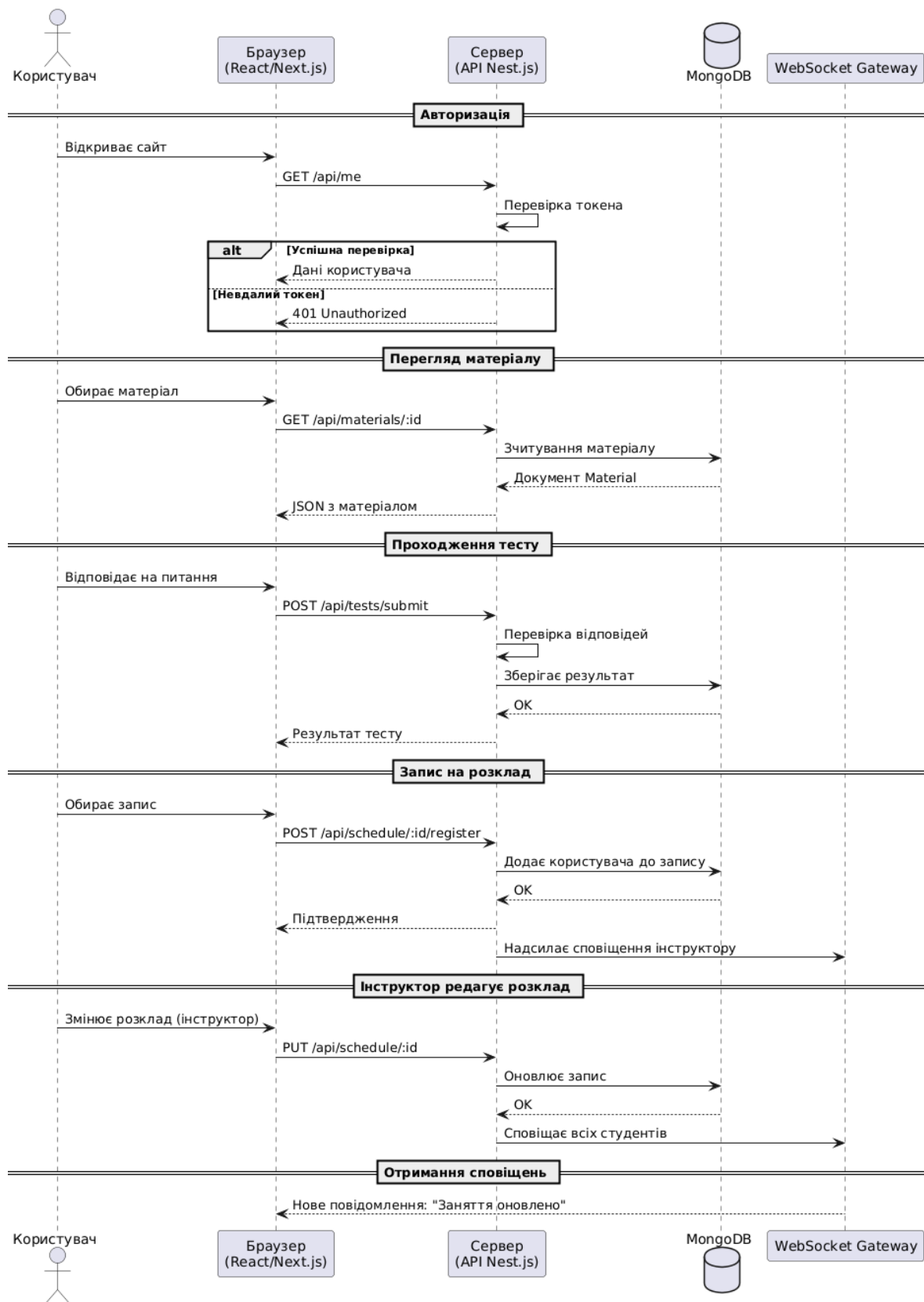


Рис. 2.7 – Діаграма послідовності вебплатформи

2.4 Реалізація застосунку для вебплатформи для організації та проведення навчальних курсів

Реалізація вебплатформи охоплює низку логічно пов'язаних підсистем, кожна з яких забезпечує конкретний аспект роботи з навчальними матеріалами, керуванням користувачами та їхнім навчальним прогресом. Архітектура платформи побудована з використанням підходу розділення відповідальностей між модулями, що підвищує масштабованість, підтримуваність і безпеку системи.

Архітектура програмного забезпечення організована на наступні компоненти (рисунк 2.8). До складу платформи входять наступні функціональні групи: система аутентифікації та авторизації, модуль керування навчальними матеріалами, система розкладу занять, модуль досягнень, бібліотека файлів, розмежування прав доступу через ролі, захист маршрутів за допомогою гвардів, обмін подіями через WebSocket, та централізована ініціалізація залежностей. Кожен компонент відіграє важливу роль у підтримці цілісного навчального процесу.

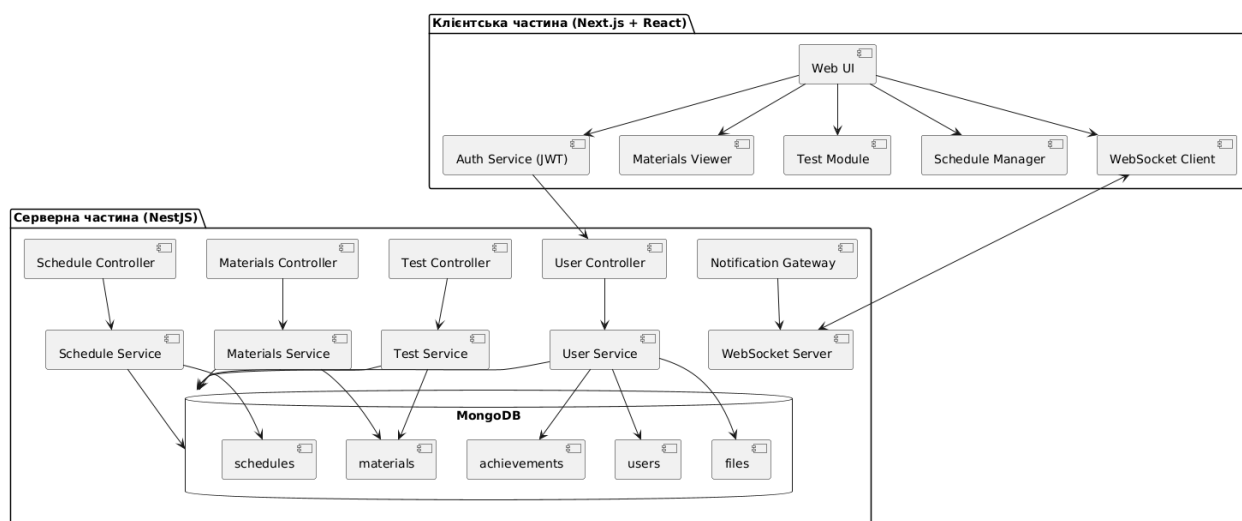


Рис. 2.8 – Діаграма компонентів системи

Підсистема конфігурації та ініціалізації. Центральний Реалізація вебплатформи передбачає побудову масштабованої, безпечної та логічно структурованої архітектури, що охоплює управління навчальними матеріалами, розкладом занять, досягненнями, ролями користувачів, завантаженням файлів і

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	33
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

сповіщеннями через WebSocket. Основний стек складається з NestJS на бекенді, MongoDB як бази даних, а також React на фронтенді. Розглянемо деякі з найважливіших складових реалізації, акцентуючи увагу на фрагментах коду, що потребують пояснення.

Система управління навчальними матеріалами реалізована з використанням MongoDB–дискримінаторів. Вона дозволяє створювати матеріали різного типу – відео, статті, лекції – і забезпечує їхню типізовану обробку. Збереження нового матеріалу в залежності від його типу наведено в додатку А.

Тут ми динамічно обираємо дискримінатор схеми за `dto.kind` і створюємо матеріал відповідного типу. Також після створення відправляється повідомлення через WebSocket, яке дає змогу оперативно повідомляти користувачів про нові матеріали без потреби у частому опитуванні сервера.

Окремої уваги заслуговує також реалізація візуалізації та підрахунку статистики щодо проведених та запланованих занять у місячному календарі. Фронтенд платформи дозволяє швидко орієнтуватися у кількості онлайн, офлайн та пройдених занять відносно обраного місяця. Частина логіки, що здійснює підрахунок записів в попередніх місяцях наведено в додатку Б.

Функція приймає масив розкладів та визначає для кожного, чи він відбувся до поточного моменту, і чи потрапляє до місяців перед обраним. Це дозволяє формувати звітність та динамічні показники в інтерфейсі, які швидко оновлюються без потреби перезавантаження сторінки.

Таке розділення логіки підрахунку на до та після місяця, а також індикатори для подій у минулому, надає користувачам зручну навігацію календарем розкладів. Візуальні компоненти платформи на кшталт **LeftCountBar** і **RightCountBar** працюють на основі цієї логіки.

Таким чином, архітектура додатку дозволяє ефективно оперувати як із створенням та обробкою матеріалів, так і з їхнім візуальним представленням у розкладі. Це значно покращує користувацький досвід та надає адміністраторам зручні інструменти для аналізу й планування освітнього процесу.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	34
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу 2

У другому розділі було здійснено комплексну реалізацію основних підсистем вебплатформи для організації та проведення навчальних курсів. Результати роботи охоплюють як бекендову архітектуру на базі NestJS з MongoDB, так і фронтендову частину, реалізовану засобами React. Окрема увага приділялася логіці доступу, зберігання та взаємодії з навчальними матеріалами, розкладом занять, досягненнями, а також підтримці інтерактивної взаємодії через WebSocket. Роботу над цим етапом можна вважати ключовою, оскільки вона безпосередньо забезпечує функціональність, яку кінцевий користувач взаємодіє щодня.

Однією з найбільш фундаментальних частин проекту стало моделювання системи матеріалів, яка підтримує різні типи навчального контенту – відео, статті, лекції. Її реалізація за допомогою MongoDB-дискримінаторів дозволила зберігати всі типи в одній колекції, водночас забезпечуючи гнучке та безпечне опрацювання типізованих документів. Це значно полегшило як створення, так і подальшу обробку даних. Логіка контролера матеріалів відображає грамотне використання патернів проектування (зокрема Strategy через перемикання по kind) і показує, як мінімізувати дублікацію коду завдяки динамічній генерації моделей.

Ще одним прикладом грамотно організованої логіки є статистичний облік занять. Місячний розклад, який динамічно відображає заняття у календарі, був посилений логікою підрахунку кількості онлайн-, офлайн- та пройдених занять до й після поточного місяця. Це рішення дозволяє не лише зручно представляти дані у візуальній формі, але й надає функціональність для подальшого використання у звітах та дашбордах. Підрахунок базується на точній перевірці дат, а також порівнянні з обраним місяцем і поточним моментом часу, що гарантує точність навіть при великій кількості даних.

На відміну від класичного використання middleware для валідації доступу або фільтрації, у платформі зроблено акцент на декларативне керування правами через декоратори @Roles у поєднанні з JwtAuthGuard та RolesGuard. Такий

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	35
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

підхід забезпечує кращу масштабованість, гнучкість і зрозумілу читаємість, дозволяючи чітко бачити вимоги до ролей у кожному методі контролера.

Окрема перевага архітектури полягає у впровадженні WebSocket-шлюзу (NotificationGateway), який повідомляє користувачів про зміни у реальному часі. Це надає сучасну взаємодію в стилі "live updates", коли нові навчальні матеріали, оновлення розкладу або досягнення з'являються в інтерфейсі користувача одразу після їх додавання адміністратором. Така функціональність раніше вимагала або ручного оновлення сторінки, або реалізації складних інтервалів опитування сервера, що значно збільшує навантаження. Натомість використання WebSocket виявилось оптимальним рішенням як з погляду UX, так і з технічної точки зору.

У межах розділу було детально реалізовано взаємодію всіх компонентів: від логіки реєстрації, автентифікації та ролей користувачів до організації розкладу та контенту, який цей розклад представляє. Робота з MongoDB була організована через репозиторний підхід, із чітким розділенням сервісів та контролерів, що спрощує підтримку та потенційне масштабування системи в майбутньому.

Таким чином, реалізація вебплатформи була виконана з урахуванням сучасних архітектурних принципів, із фокусом на масштабованість, типізованість та безпеку. Обраний технологічний стек, поєднаний із грамотними рішеннями в архітектурі та коді, дозволив створити гнучку і потужну платформу, здатну ефективно задовольняти як навчальні, так і адміністративні потреби цільової аудиторії. Результати, отримані на цьому етапі, формують міцний фундамент для майбутньої розробки мобільної версії системи, а також подальшого розширення її функціональності.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ЗАСТОСУНКОМ

3.1. Порядок встановлення та налаштування параметрів системи.

Для розгортання вебплатформи було обрано сучасну архітектуру клієнт–сервер з поділом на два основні середовища – **фронтенд (React)** та **бекенд (NestJS)**. Обидві частини працюють як незалежні сервіси, що дає змогу масштабувати додаток за необхідності та ізолювати окремі його частини. Також це дозволяє забезпечити зручність розгортання, безпеку й простоту оновлення без порушення основного циклу роботи.

- **Vercel** для хостингу клієнтської частини;
- **Railway** для хостингу серверної частини та бази даних.

Для локального запуску система потребує наявності встановленого:

- **Node.js** (версія ≥ 18);
- менеджера пакетів **npm**;
- **MongoDB** або відповідного хмарного екземпляра;
- **.env** файлів із змінними середовища.

Змінні оточення (.env) є важливим інструментом конфігурації системи. Вони використовуються як у фронтенді (наприклад, для API–ендпоінтів чи ключів WebSocket, JWT), так і на бекенді (для JWT секретів, підключення до бази даних, шляху до публічних файлів тощо).

Розгортання на Vercel

Фронтенд додатку, реалізований на базі Next.js, було розгорнуто на Vercel – це дозволило автоматизувати процес публікації: кожен push у головну гілку GitHub–репозиторію запускає процес побудови та деплою. Система інтегрується з GitHub, тому не потребує ручних оновлень. Vercel автоматично налаштовує HTTPS, CDN, та серверний рендеринг. UI–інтерфейс Vercel дозволяє керувати середовищами (preview, production). Підтримка fallback–поведінки, коли сервер недоступний, дозволяє уникати збоїв.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	37
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

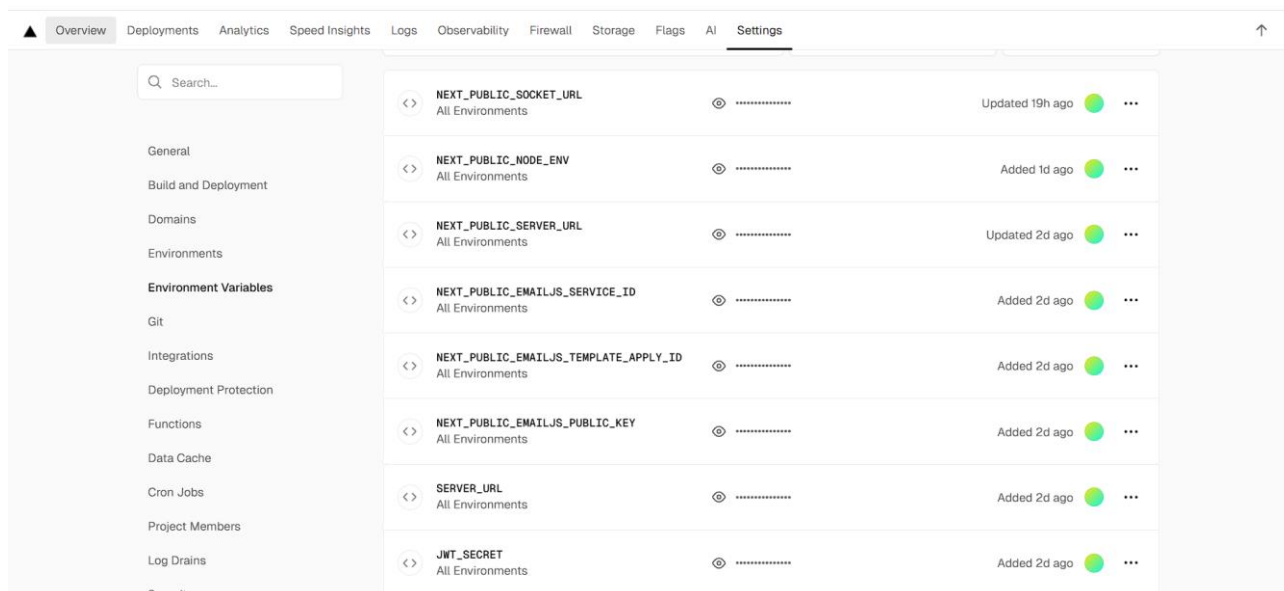


Рис. 3.1 – Конфігурація змінних оточення у Vercel Dashboard.

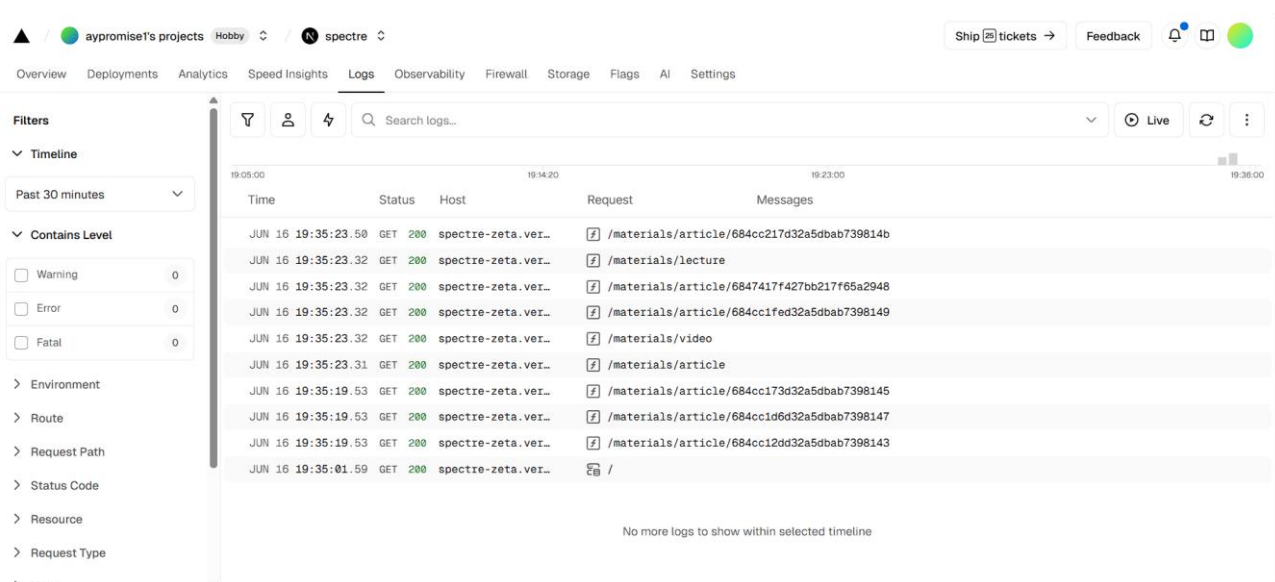


Рис. 3.2 – Перегляд логів в Vercel.

Розгортання на Railway

Серверна частина проєкту, реалізована на NestJS, розгортається на Railway. Railway дозволяє легко створити проєкт, налаштувати змінні середовища та базу даних MongoDB у кілька кліків. Додатково Railway надає: зручну CLI-утиліту для локальної розробки та деплою; можливість переглядати логування в реальному часі; REST API для моніторингу.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	38
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Railway дозволяє окремо створити сервіси типу mongo, а потім просто підключити URI до Node-серверу. Завдяки цьому, серверне оточення стало незалежним, із централізованим зберіганням змінних.

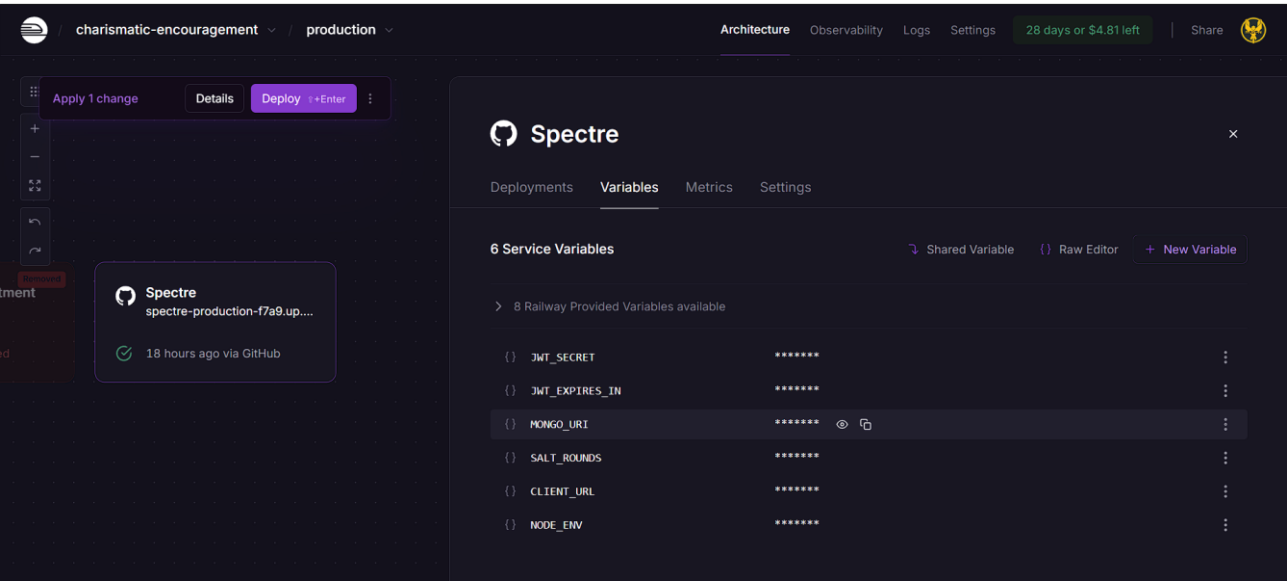


Рис. 3.3 – Налаштування MongoDB сервісу на Railway.

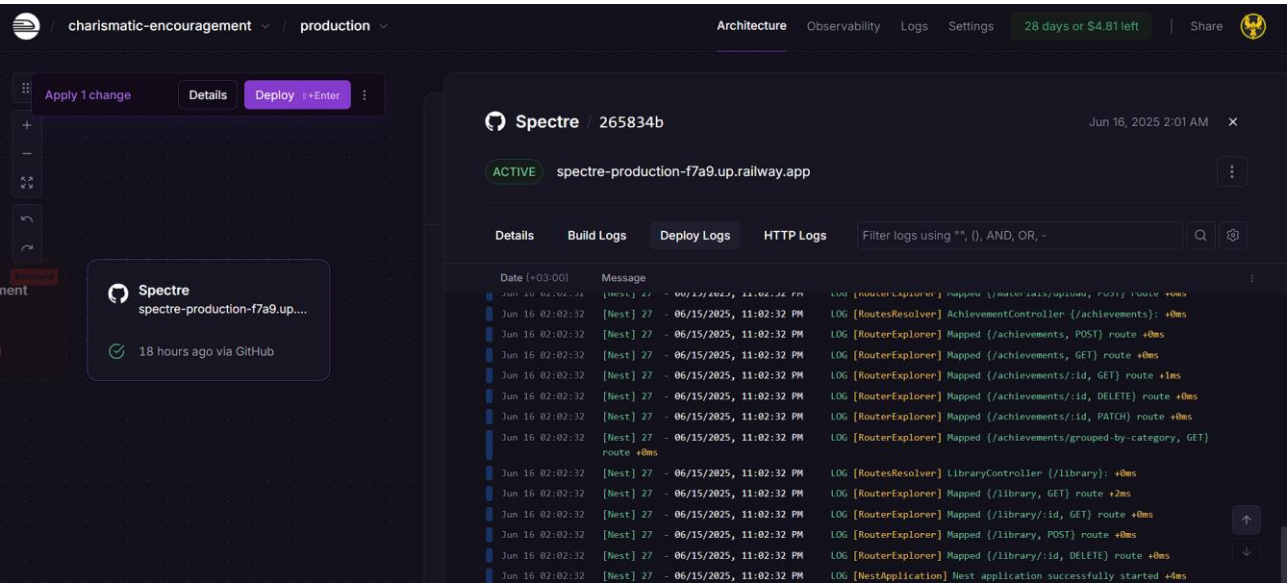


Рис. 3.4 – Перегляд логів сервера в Railway.

Railway також підтримує деплой через GitHub – пуш до main чи production гілки автоматично оновлює систему. Цей процес можна налаштувати вручну, або використати шаблон CI/CD:

Загальна схема розгортання

Загалом архітектура побудована як класичний децентралізований клієнт-серверний підхід. Frontend працює на Vercel, робить запити до Railway backend.

The diagram illustrates the system architecture for the Next.js SPA application. It is organized into several main components and their interactions:

- Клієнтський пристрій (Client Device):** Contains the **Браузер (Next.js SPA) (Browser)** and the **Веб-сервер (Web Server)**.
 - The **Браузер (Next.js SPA)** includes the **Інтерфейс користувача (User Interface)** and the **WebSocket клієнт (WebSocket client)**.
 - The **Веб-сервер** contains the **Next.js Frontend** with **SSR / CSR сторінки (SSR / CSR pages)**.
- Сервер застосунку (Application Server):** Contains the **NextJS Backend** with **JWT Auth**, **WebSocket Gateway**, and **REST API**.
- База даних (Database):** Contains **MongoDB Atlas** and **MongoDB**.
 - MongoDB Atlas** is connected to the **WebSocket Gateway** and **REST API** via **Моніторинг змін (Change Monitoring)**.
 - MongoDB** contains collections: **users**, **materials**, **schedules**, **achievements**, and **files**.
- Хмарне сховище (Cloud Storage):** Contains **Cloud Storage (наприклад, AWS S3 / Railway FS) (e.g., AWS S3 / Railway FS)** and **Зберігання файлів (File Storage)**.

Interactions:

- The **Браузер (Next.js SPA)** sends **Реальний час (Real-time)** data to the **Сервер застосунку**.
- The **Браузер (Next.js SPA)** sends **HTTP-запити (HTTP requests)** to the **Сервер застосунку**.
- The **Сервер застосунку** sends **Завантаження/Отримання файлів (File upload/download)** to the **Cloud Storage**.

3.2 Структура інтерфейсу вебплатформи для організації та проведення навчальних курсів

Перш ніж розпочати роботу з системою, користувач має пройти автентифікацію. Для цього реалізовано сторінку входу, доступну за маршрутом «*/sign-in*». Сторінка має адаптивну форму з валідацією введених даних, а також систему оповіщень у випадку помилок (наприклад, неправильна пошта або пароль).

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ІЗ	40
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Увійдіть в свій акаунт

Email* Невірний формат пошти

фів

Password* Це поле не може бути пустим [Забули свій пароль?](#)

Увійти

Бажаєте приєднатися до **SPECTRE**? [Заповніть форму](#)

Рис. 3.2 – Сторінка авторизації з прикладом повідомлення про помилку

Увійдіть в свій акаунт

Email*

example@gmail.com

Password* [Забули свій пароль?](#)

Увійти

Бажаєте приєднатися до **SPECTRE**? [Заповніть форму](#)

! Введено невірні дані

Рис. 3.3 – Сторінка авторизації з прикладом повідомлення про відсутність користувача з такими даними

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	41
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

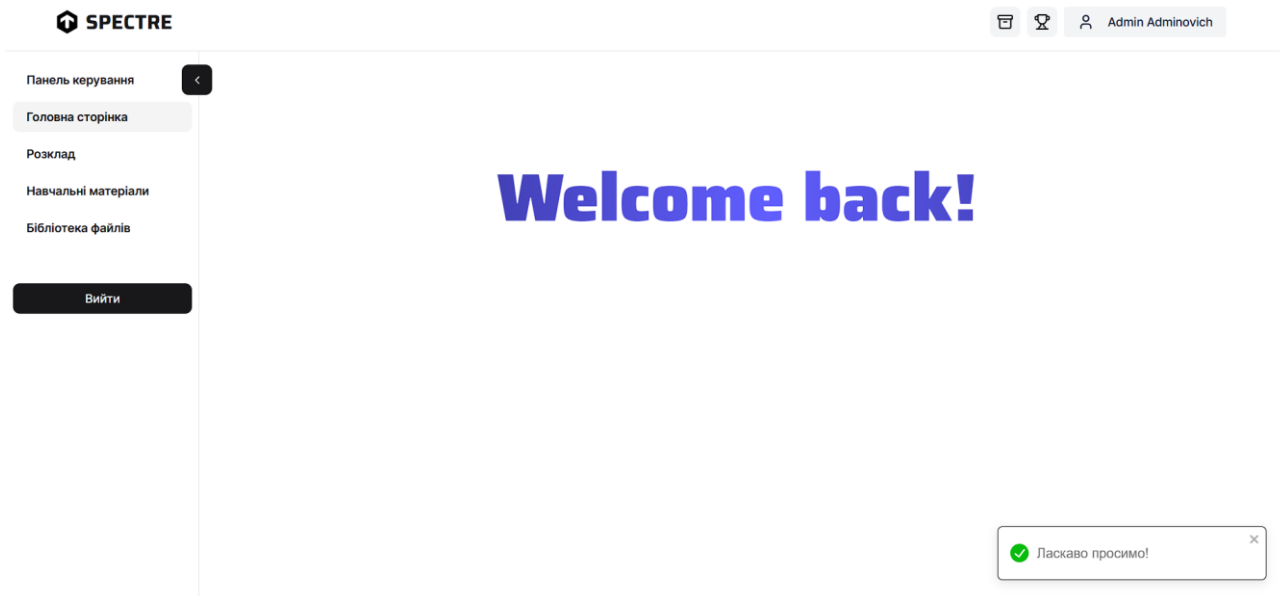


Рис. 3.4 – Приклад успішної авторизації

Після входу користувач потрапляє на головну сторінку «/», яка для авторизованих відображає вітання та навігаційні блоки. Для неавторизованих доступна форма подання заявки на створення користувача. Цей екран дозволяє розпочати взаємодію з системою, подати заявку на реєстрацію або повернутися до авторизації.

SPECTRE

Увійти

Приєднуйся

Подати заявку

Ім'я*

Прізвище*

Пошта*

example@mail.com

Номер телефону*

(38) _ _ _ _ _

Мета*

Яка ваша мета?

Send

Рис. 3.5 – Головна сторінка для неавторизованих користувачів

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	42
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

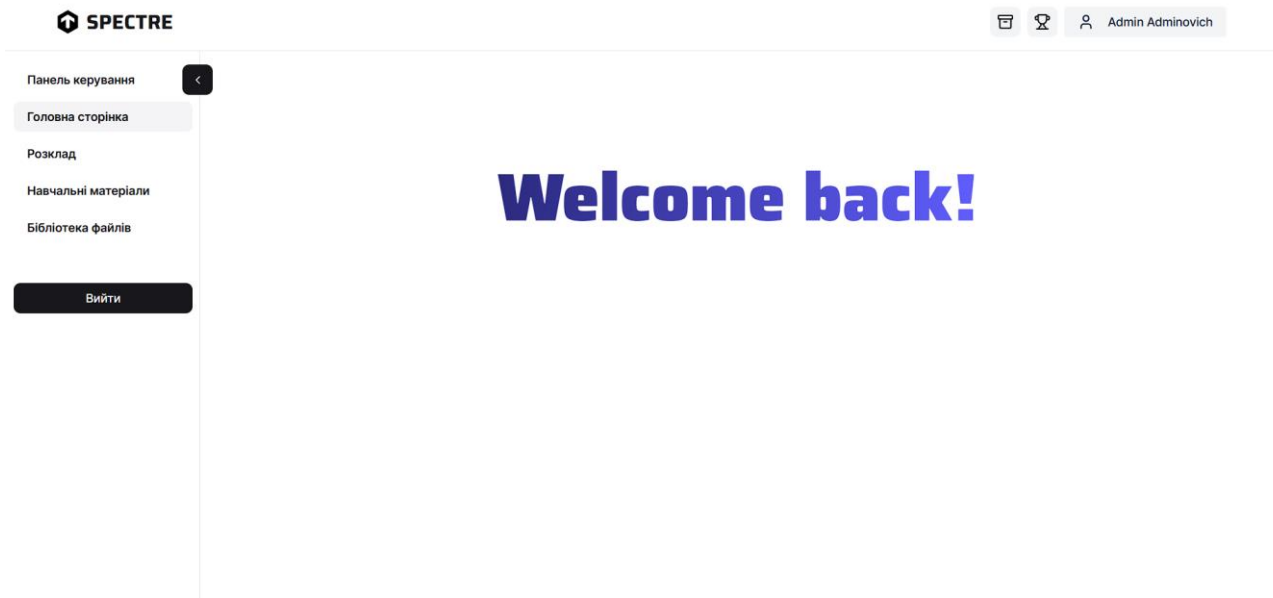


Рис. 3.6 – Головна сторінка для авторизованих користувачів

Маршрут «/schedule» відображає розклад усіх доступних занять у вигляді календаря. Користувач може бачити заплановані заняття, а також переглянути деталі кожного, натиснувши на відповідну подію. У вікні, що з’являється, можна записатися або скасувати участь. Всі дії мають зворотний зв’язок через сповіщення.

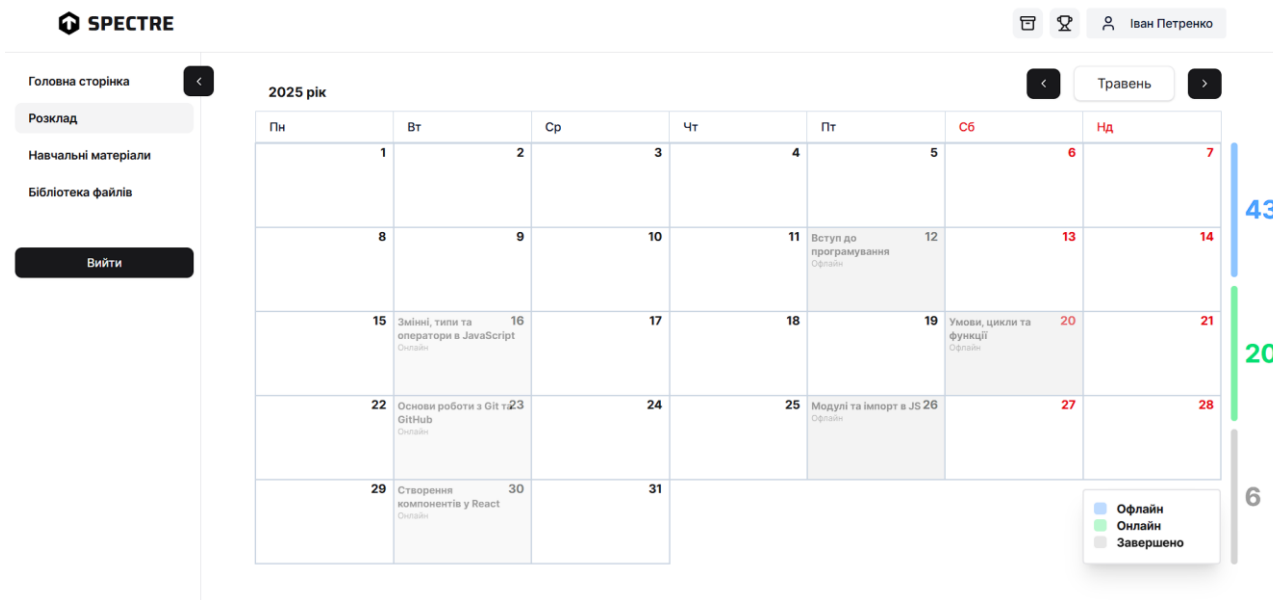


Рис. 3.7 – Сторінка розкладу з календарем

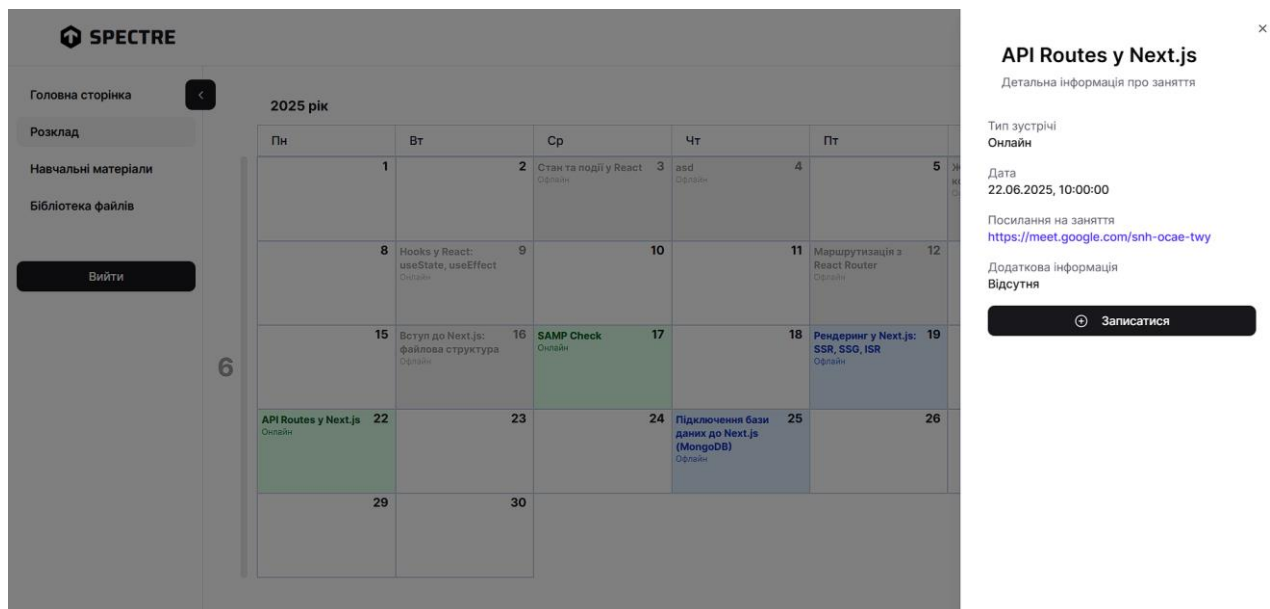


Рис. 3.8 – Модальне вікно окремого заняття, де ми не записані

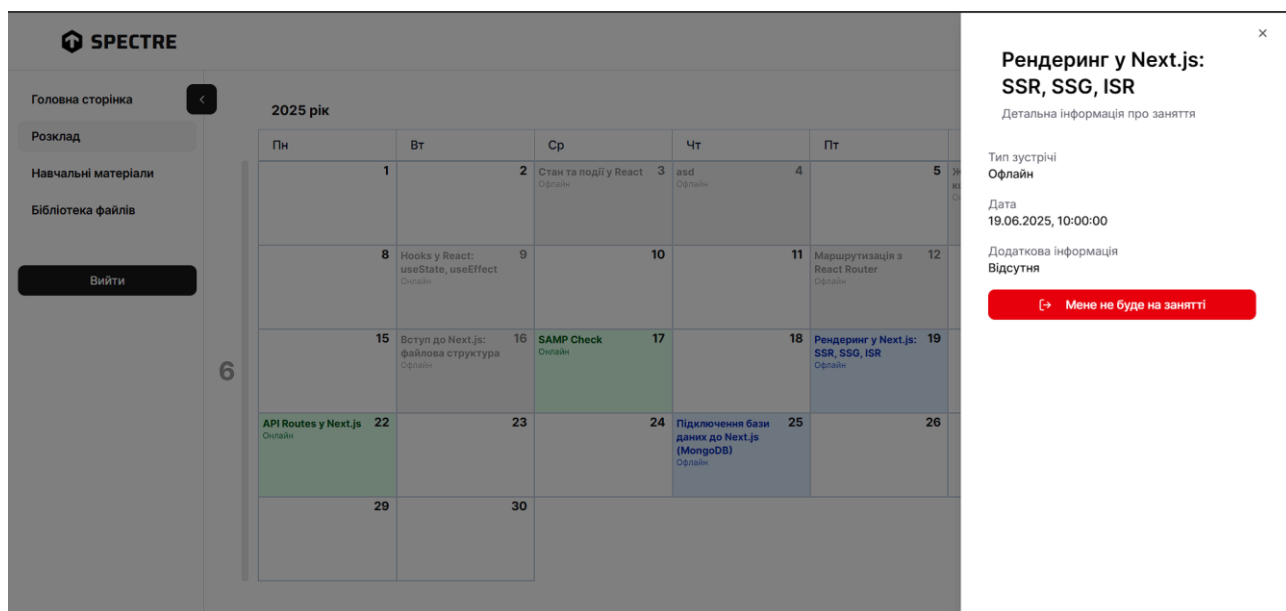


Рис. 3.9 – Модальне вікно окремого заняття, де ми записані

На маршруті «*/materials*» розміщено всі навчальні матеріали, доступні користувачеві. Вони згруповані за категоріями, можна виконувати фільтрацію за курсами. Візуально кожен матеріал представлений у вигляді картки з короткою інформацією.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	44
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

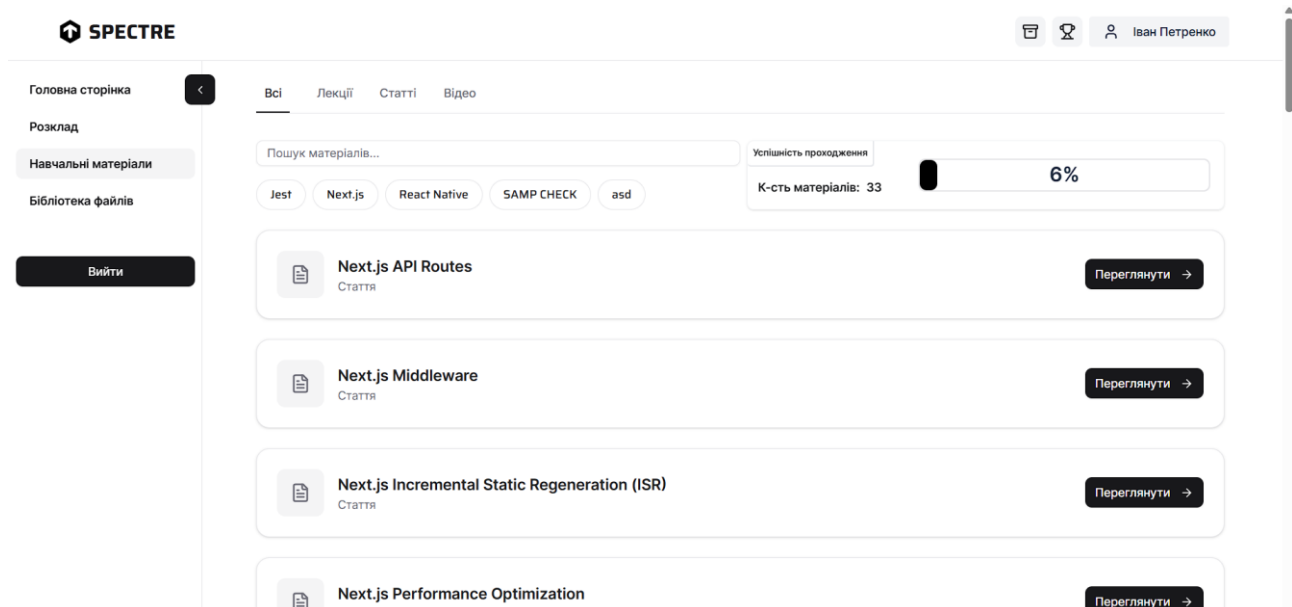


Рис. 3.10 – Сторінка навчальних матеріалів

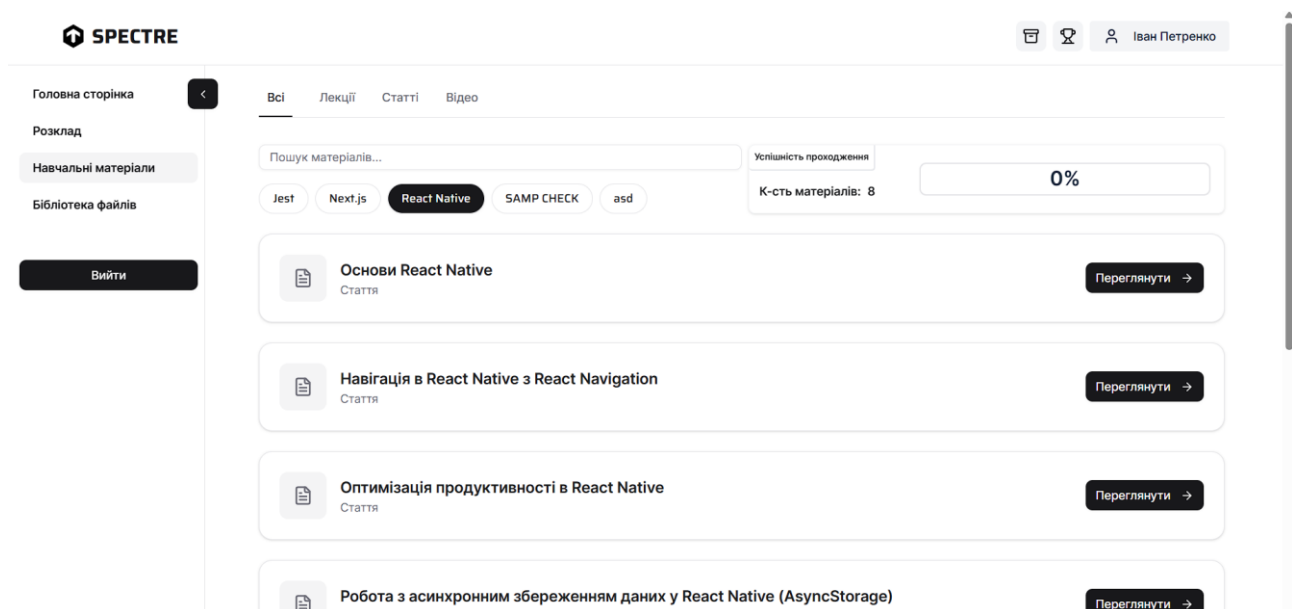


Рис. 3.11 – Сторінка навчальних матеріалів з фільтрацією по окремому курсу

На «*/materials/lecture*» відображаються виключно лекції. Це дозволяє швидко зосередитися на теоретичному матеріалі. Усі записи мають однаковий формат представлення, що спрощує навігацію.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	45
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

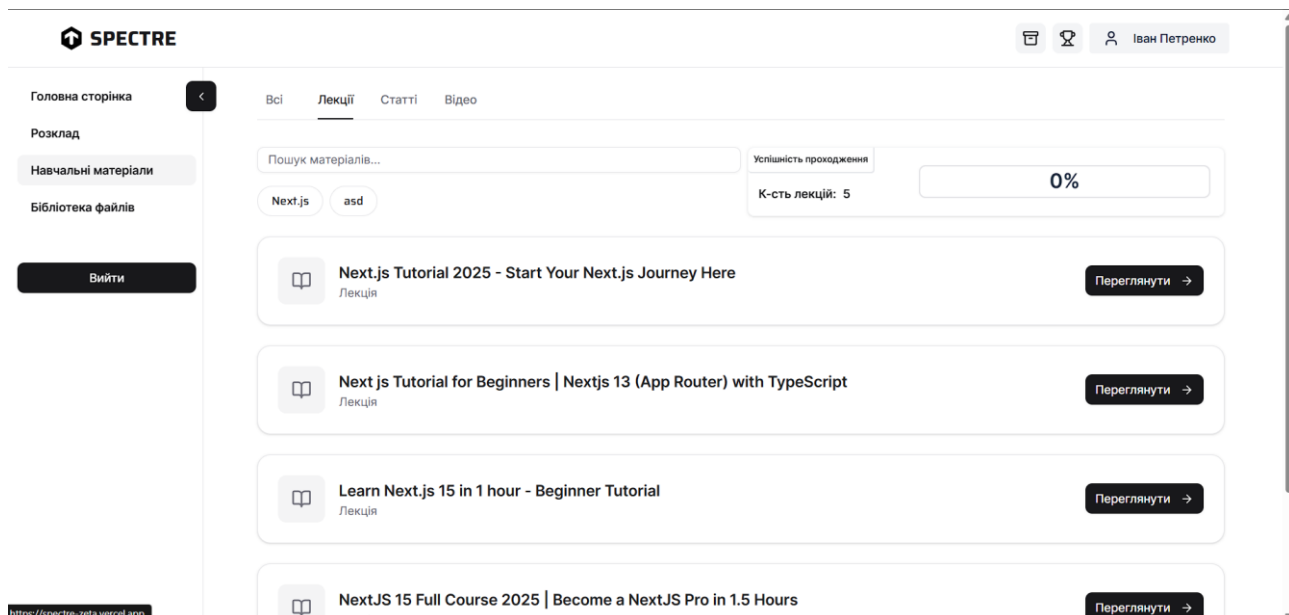


Рис. 3.12 – Сторінка з лекціями

Маршрут «`/materials/lecture/:id`» відкриває конкретну лекцію. У нижній частині розміщено інтерактивний тест. Успішне проходження супроводжується зеленою модальною, яка інформує про завершення матеріалу. Невдача викликає відповідне повідомлення з можливістю повтору.

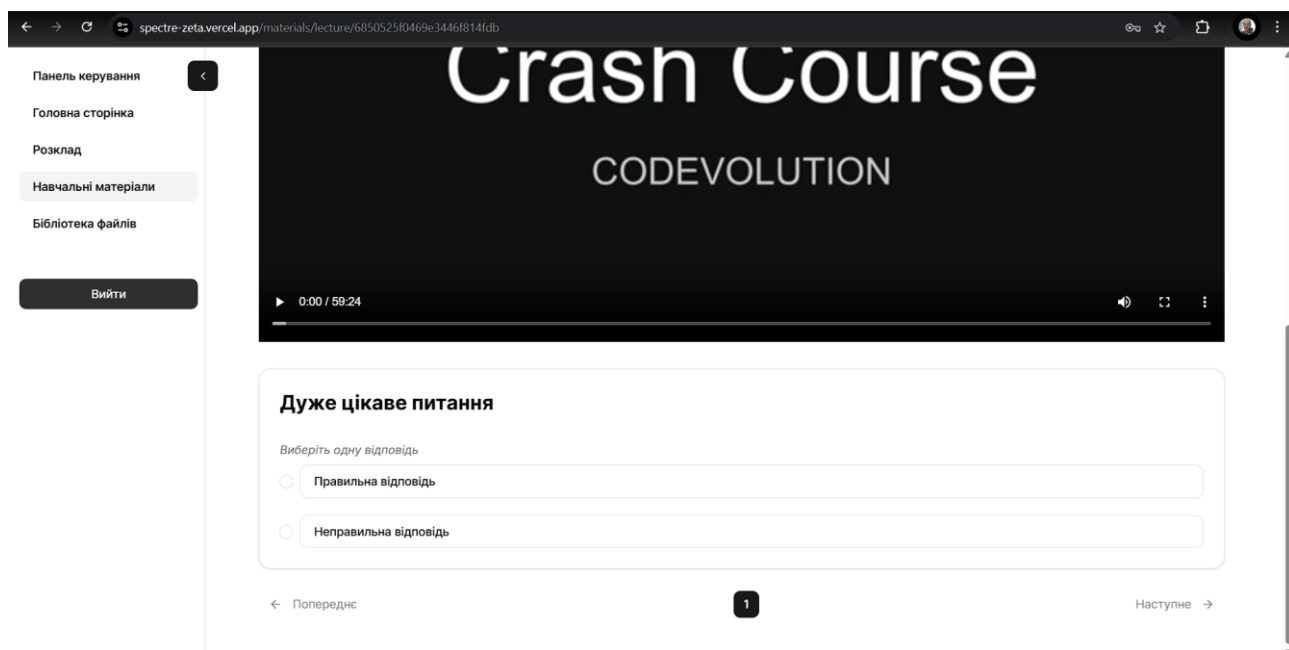


Рис. 3.13 – Сторінка лекції з інтерактивним тестом

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	46
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

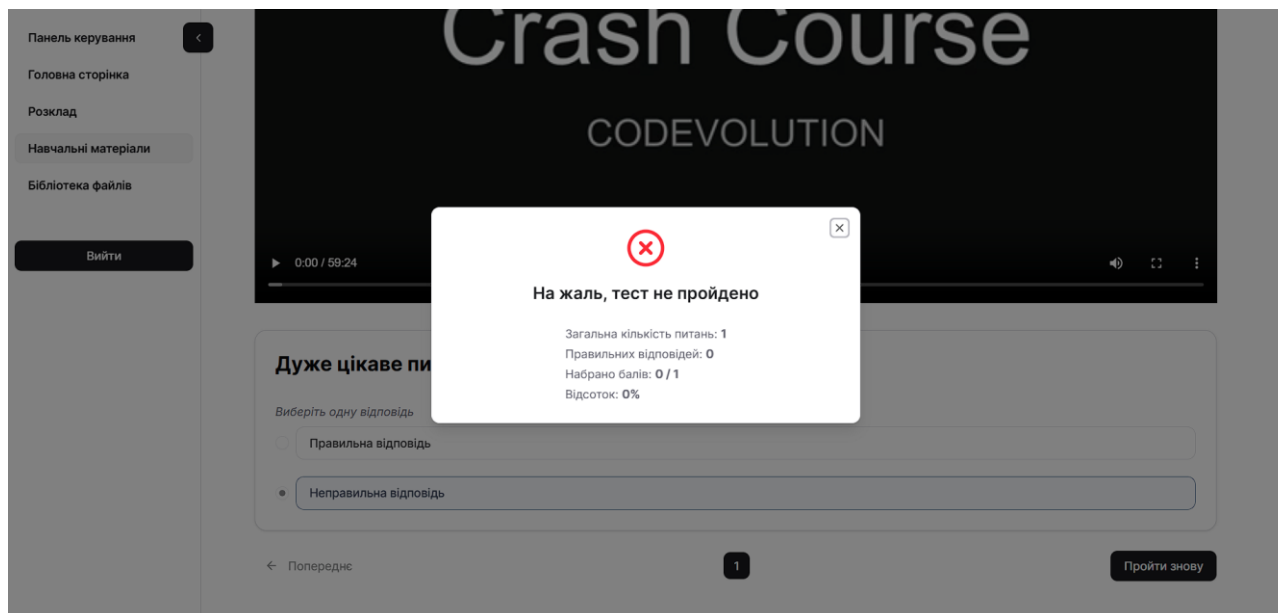


Рис. 3.14 – Реакція на провалений тест

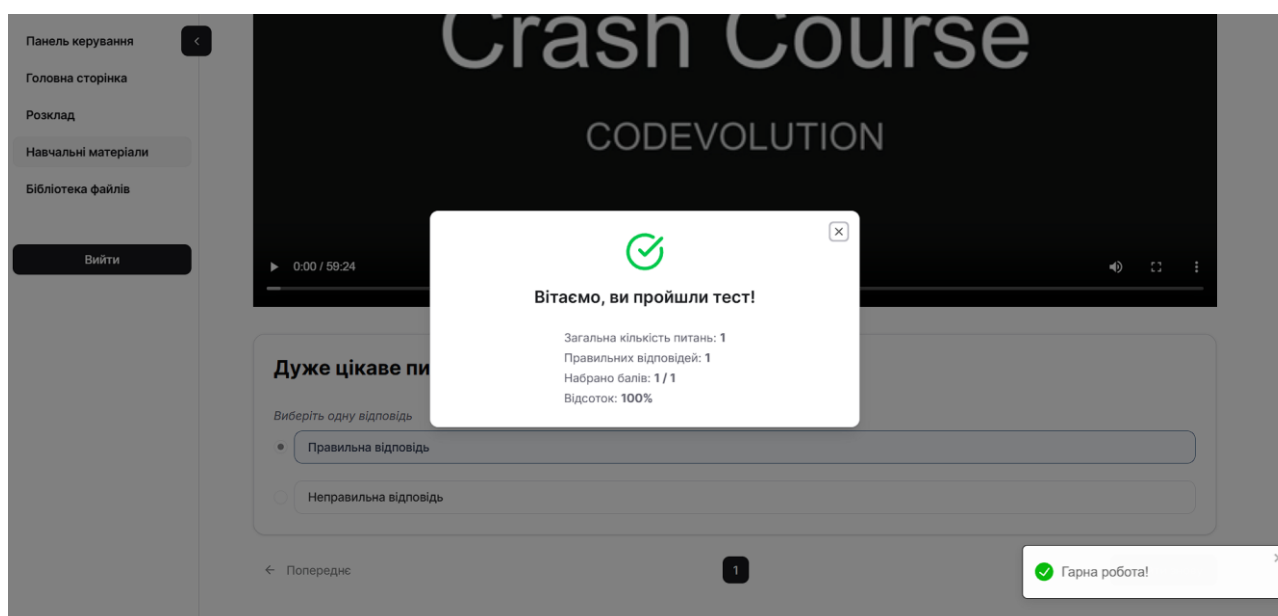


Рис. 3.15 – Реакція на успішно пройдений тест

Сторінка «[/materials/article/:id](#)» відображає статтю з освітнім змістом, після якої також іде тест для перевірки засвоєного матеріалу. Після проходження результат обробляється аналогічно до лекцій.

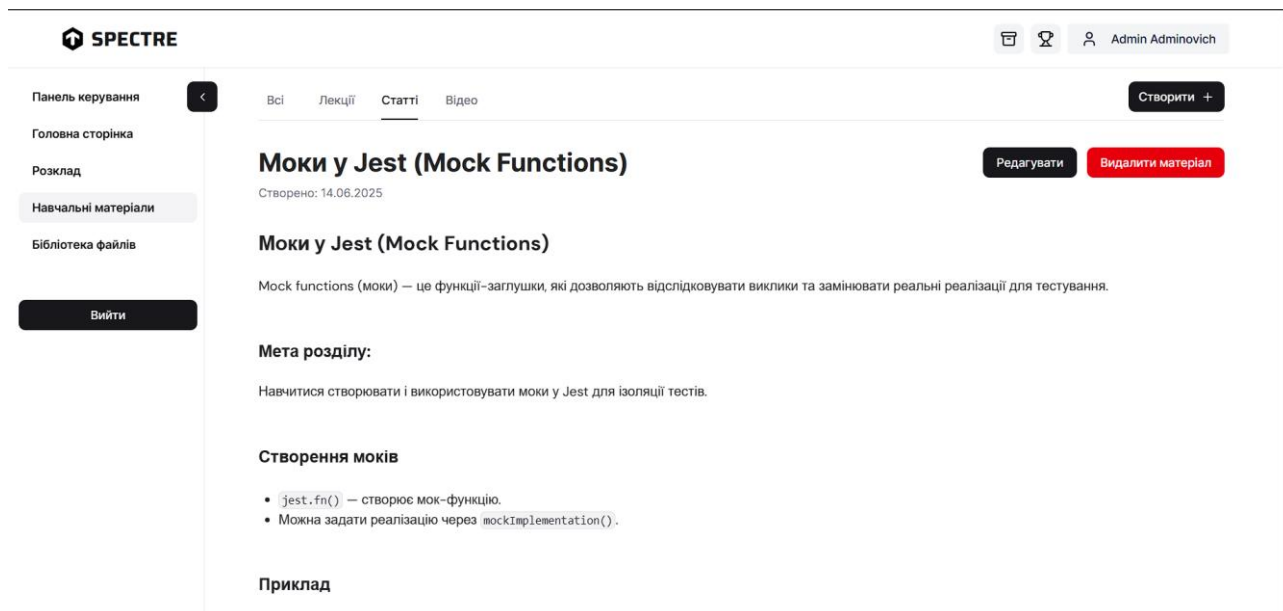


Рис. 3.16 – Сторінка статті з тестом

Сторінка «*/materials/video/:id*» має мінімалістичний вигляд — відеопрогравач та кнопка завершення перегляду. Наявність лише однієї дії обумовлена тим, що тестування для відео не передбачено.

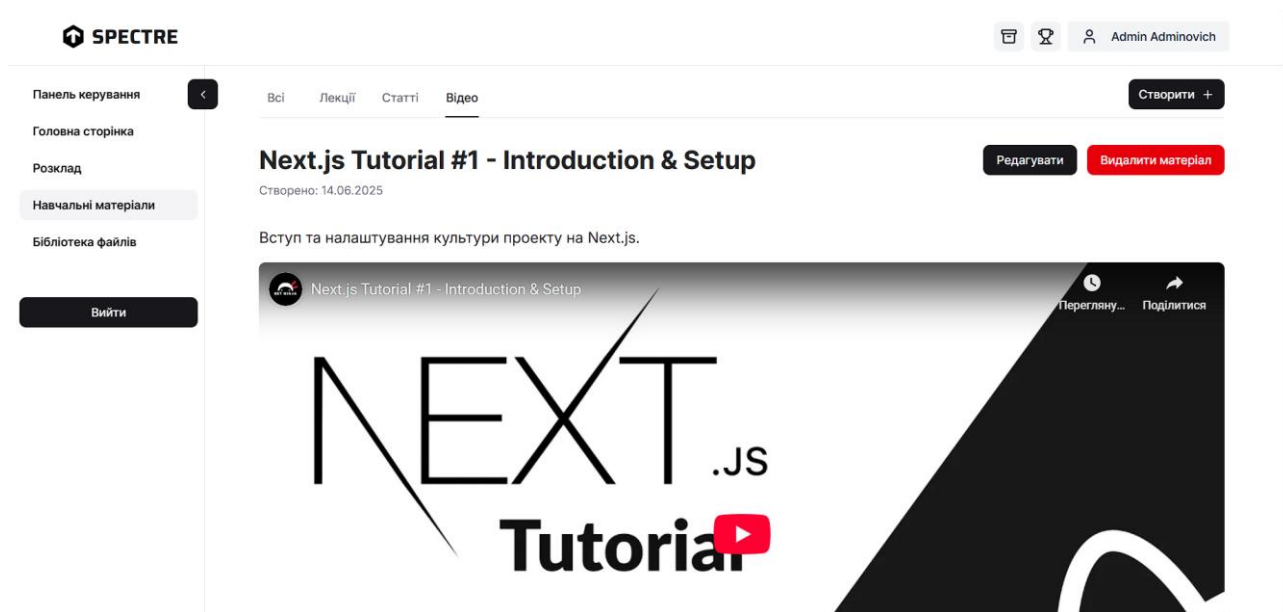


Рис. 3.17 – Сторінка відеоматеріалу

Маршрут «*/library*» дозволяє користувачам бачити всі доступні завантаження. Користувач може додавати файли, видаляти лише свої, або завантажувати будь-які. В інтерфейсі чітко розділено файли за типами та джерелами.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	48
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

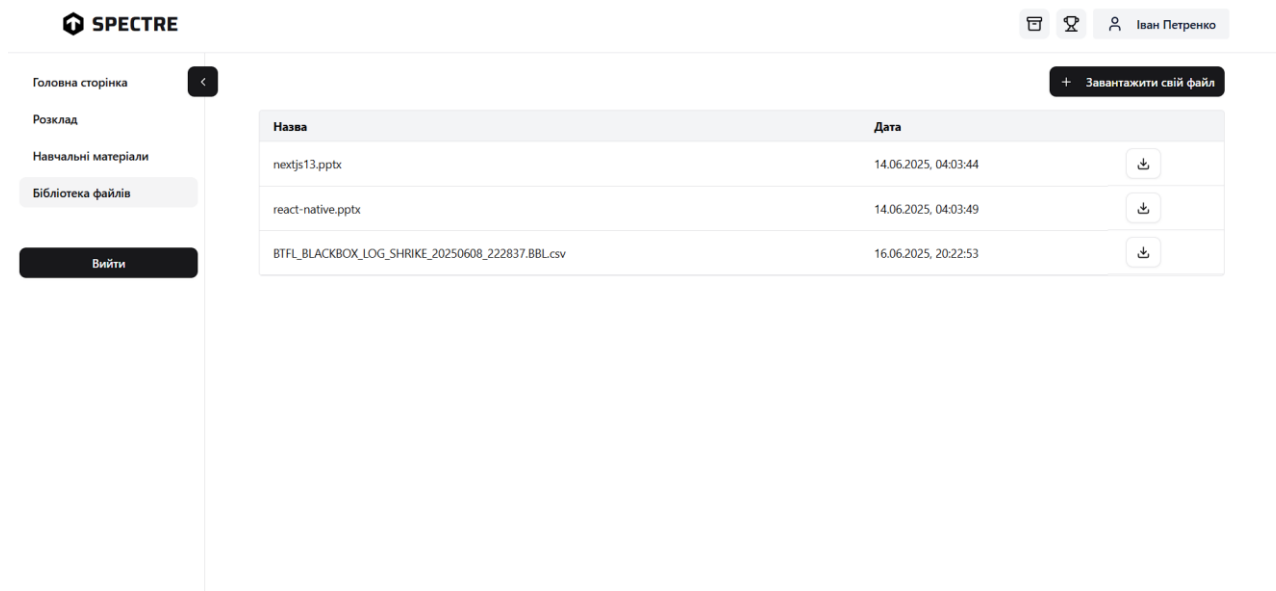


Рис. 3.18 – Сторінка завантажених файлів із можливістю керування

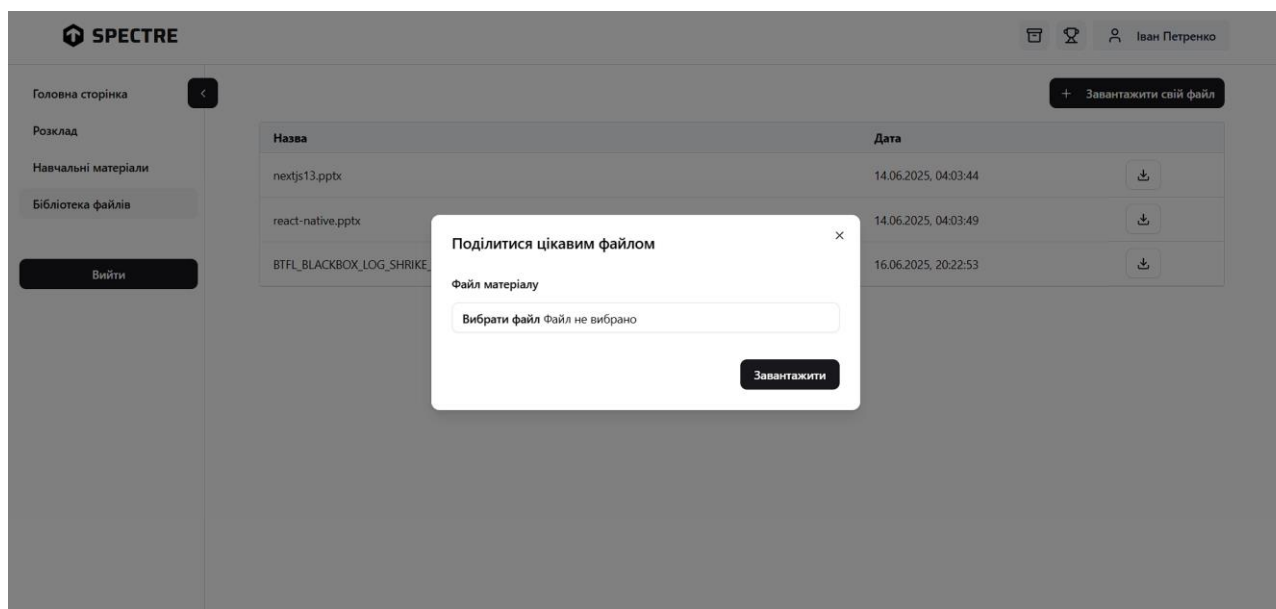


Рис. 3.19 – Модальне вікно завантаження файлу

На сторінці «/achievements» представлено досягнення користувача: завершені матеріали, успішно пройдені тести, загальний прогрес. Інтерфейс організований у вигляді категорій.

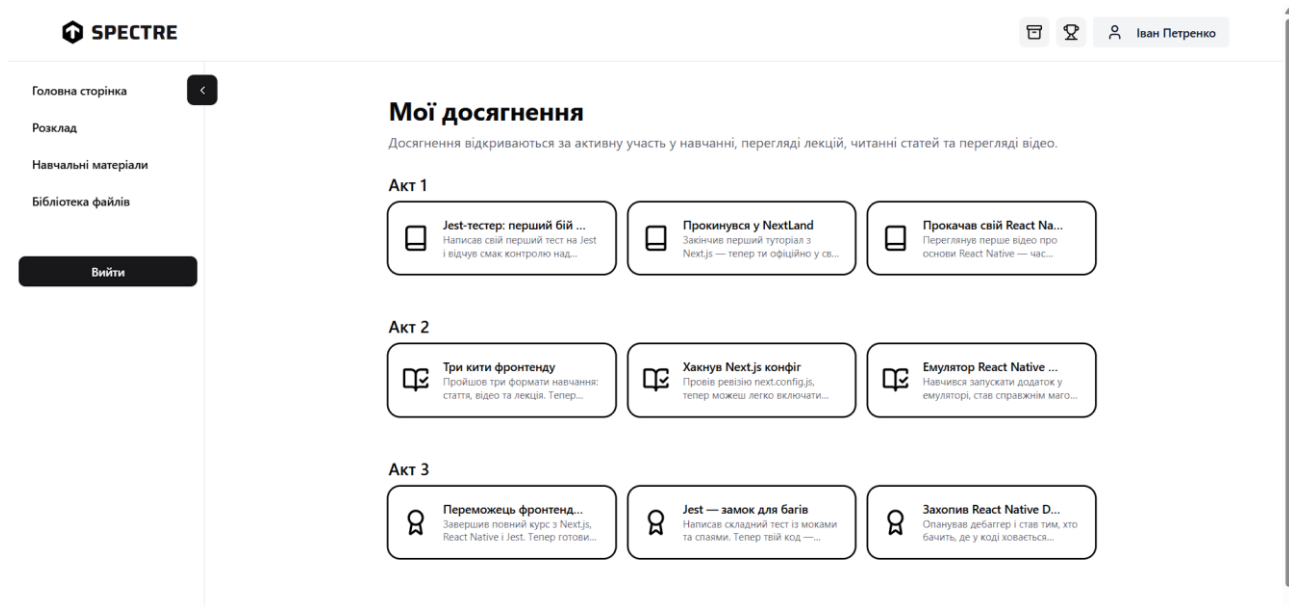


Рис. 3.20 – Екран досягнень користувача

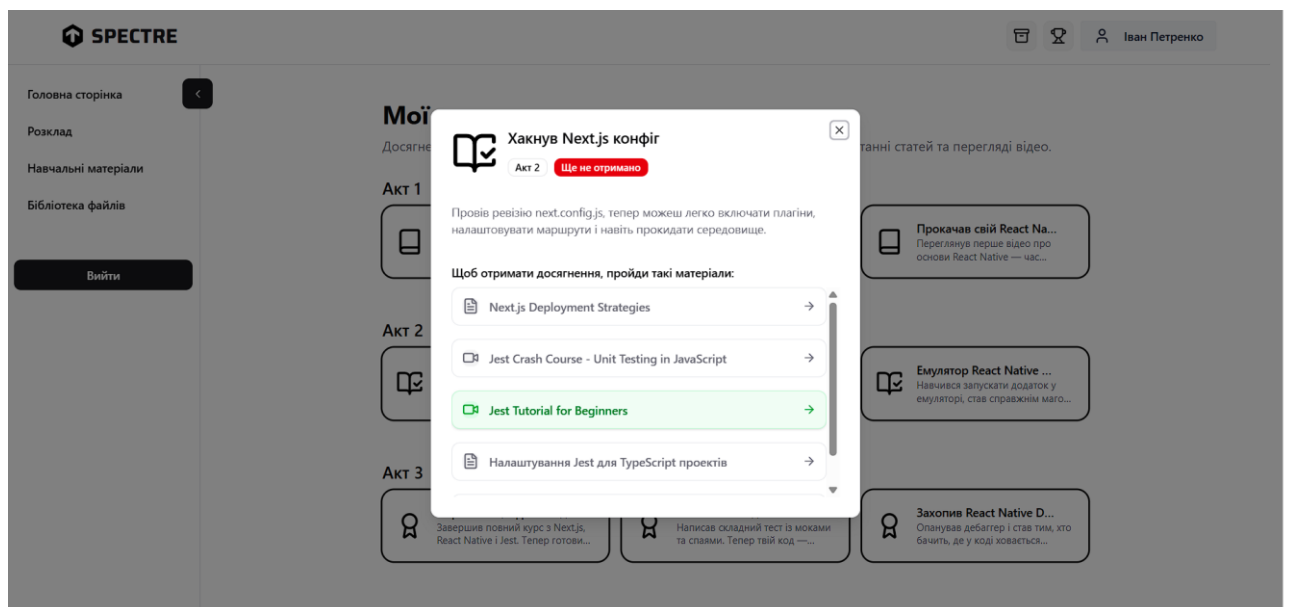


Рис. 3.21 – Модальне вікно досягнення

Інструктор бачить відмінності на вже знайомих маршрутах:

На сторінці «/schedule» реалізовано повний функціонал керування розкладом занять. Можливість створення нового заняття позначена іконкою плюса «+», що відображається у порожній клітинці відповідного дня в календарі. Натискання на цю іконку відкриває модальне вікно з формою для додавання нового запису. Редагування вже існуючого заняття здійснюється шляхом натискання на відповідний запис у календарі, після чого відкривається модальне вікно з попередньо заповненими полями, доступними для змін. Видалення

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	50
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

заняття також можливе з цього модального вікна – у ньому передбачено окрему кнопку для цього. Усі дії супроводжуються інтуїтивно зрозумілими кнопками з відповідними підписами або іконками.

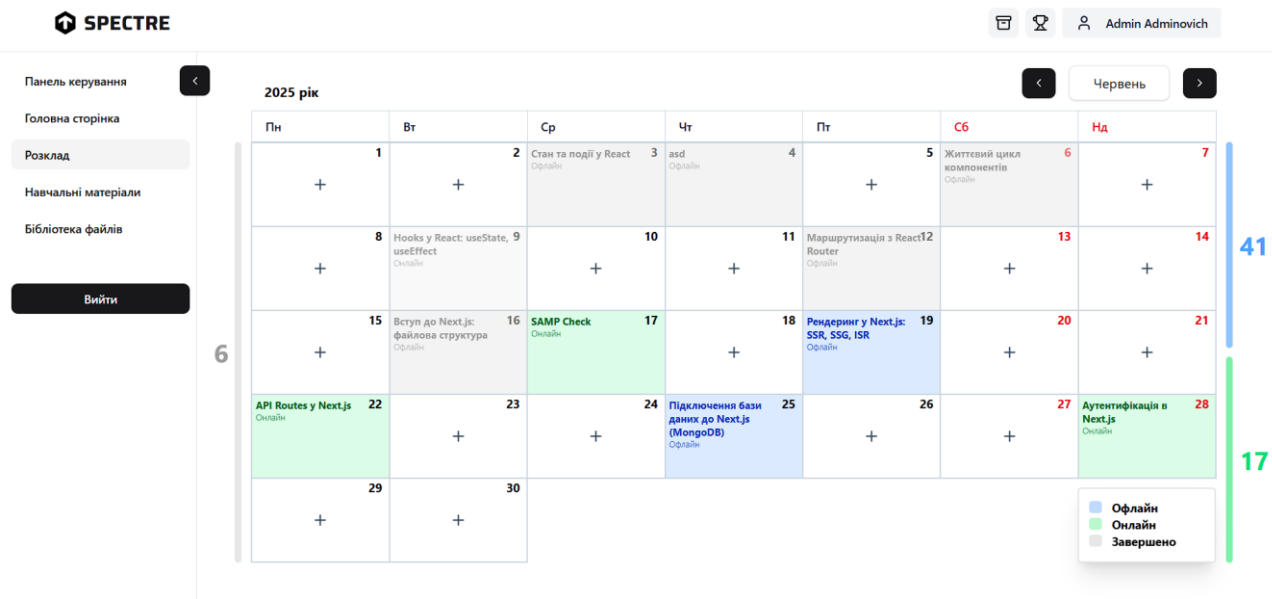


Рис. 3.22 – Сторінка розкладу для інструктора з кнопками керування

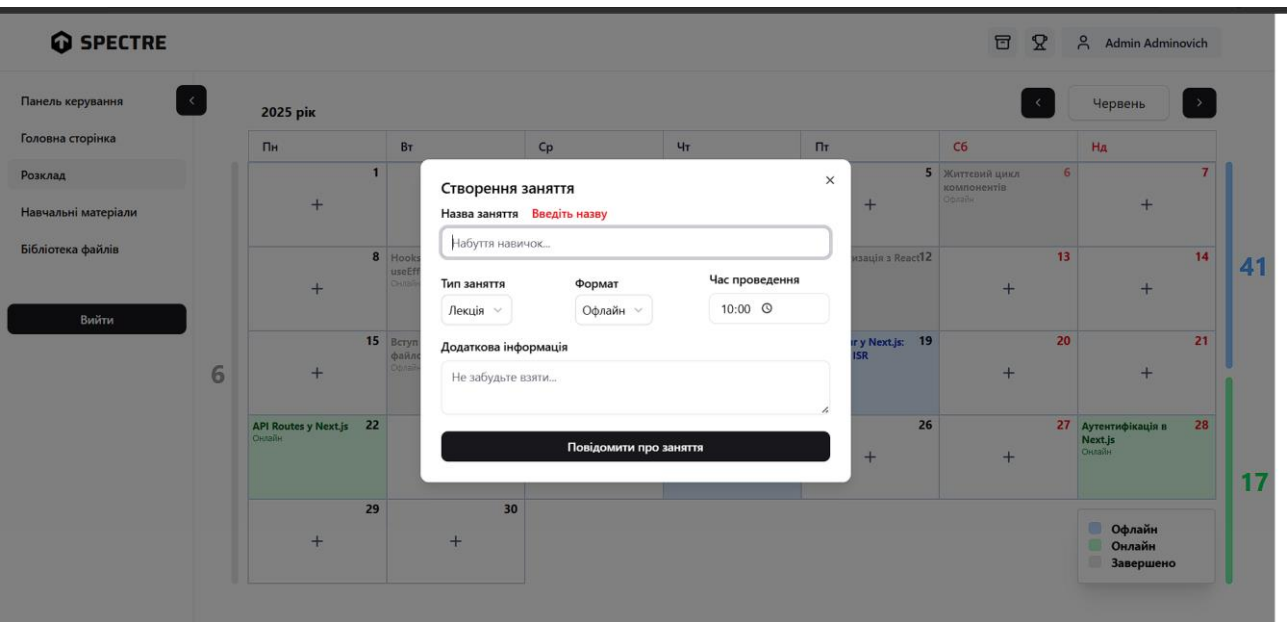


Рис. 3.23 – Модальне вікно створення запису

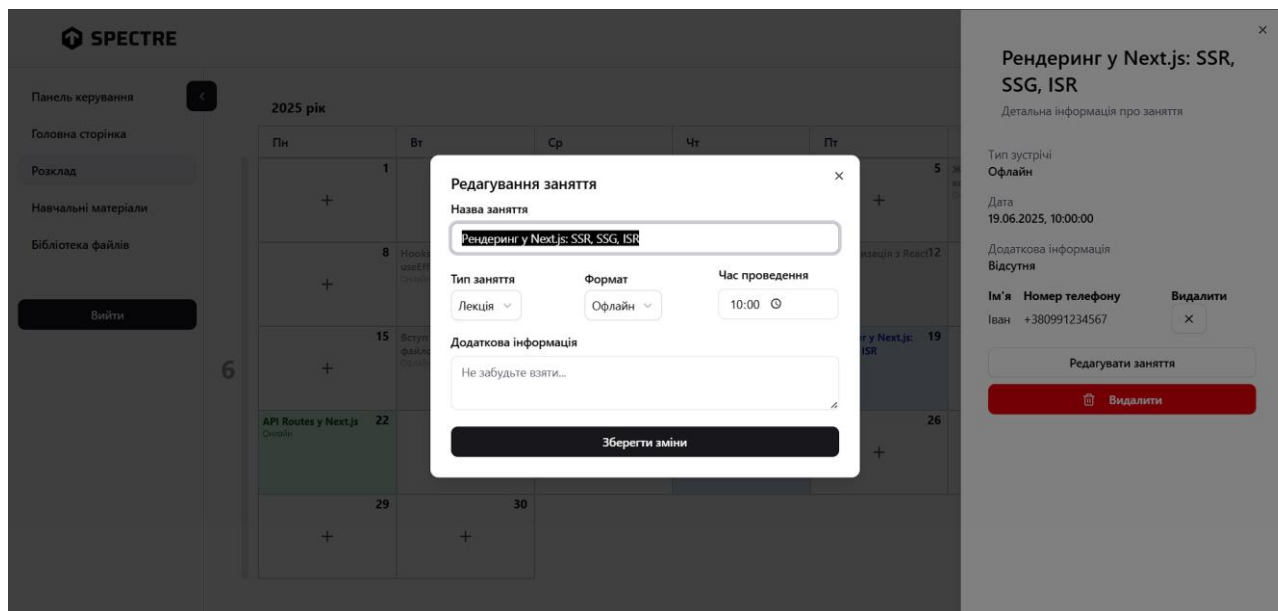


Рис. 3.23 – Модальне вікно редагування запису

На «/materials» – доступні кнопки редагування, створення й видалення.

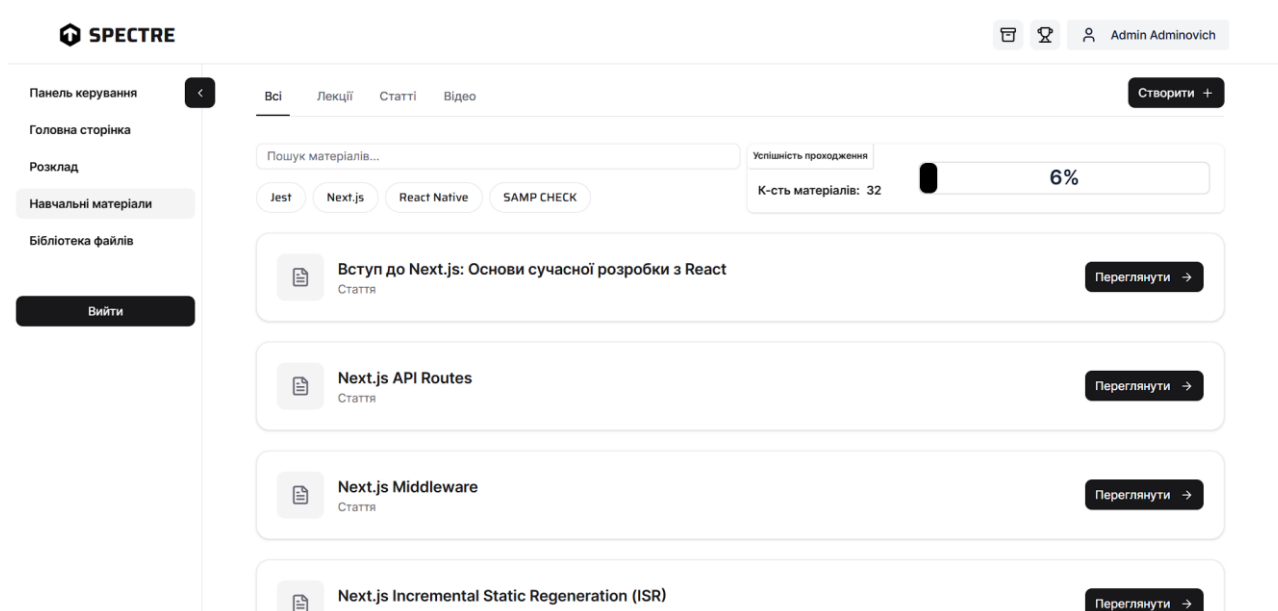


Рис. 3.24 – Інтерфейс перегляду матеріалів для інструктора

«/materials/create» – сторінка з окремими формами для кожного типу матеріалу: лекція, стаття, відео.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	52
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.25 – Сторінка створення навчального матеріалу

Сторінка «[/materials/lecture/:id/edit](#)» надає можливість редагування навчального матеріалу з урахуванням його типу – лекція, стаття або відео. Залежно від обраного типу, форма автоматично підлаштовується під відповідні вимоги.

Під час редагування обов’язковим є вибір навчального курсу, до якого належить матеріал, а також вказання заголовку та короткого опису – ці поля слугують базовою ідентифікацією контенту для студентів. Якщо редагується відеоматеріал, потрібно надати валідне посилання на відео (наприклад, YouTube або іншу платформу). Для лекцій – необхідне завантаження файлу у форматі PDF або DOCX. Якщо йдеться про статтю, обов’язковим є заповнення поля з текстовим WYSIWYG–контентом.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	53
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.26 – Сторінка для редагування навчального матеріалу

Адміністратори мають окремі маршрути для повного управління платформою:

На сторінці «*/dashboard*» розміщено таблицю з повним переліком користувачів платформи. У цій таблиці відображаються ключові дані про кожного користувача, зокрема ім'я, прізвище, електронна пошта, роль та інші параметри. Для кожного запису доступні дії редагування або повного видалення, що дозволяє адміністраторам оперативно оновлювати інформацію або очищувати систему від неактуальних облікових записів. Над таблицею розміщено кнопку створення нового користувача, яка веде на відповідну форму створення (сторінка «*/dashboard/create-user*»). Окрім того, інтерфейс передбачає перегляд надісланих заявок на реєстрацію та загальні метрики щодо активності користувачів.

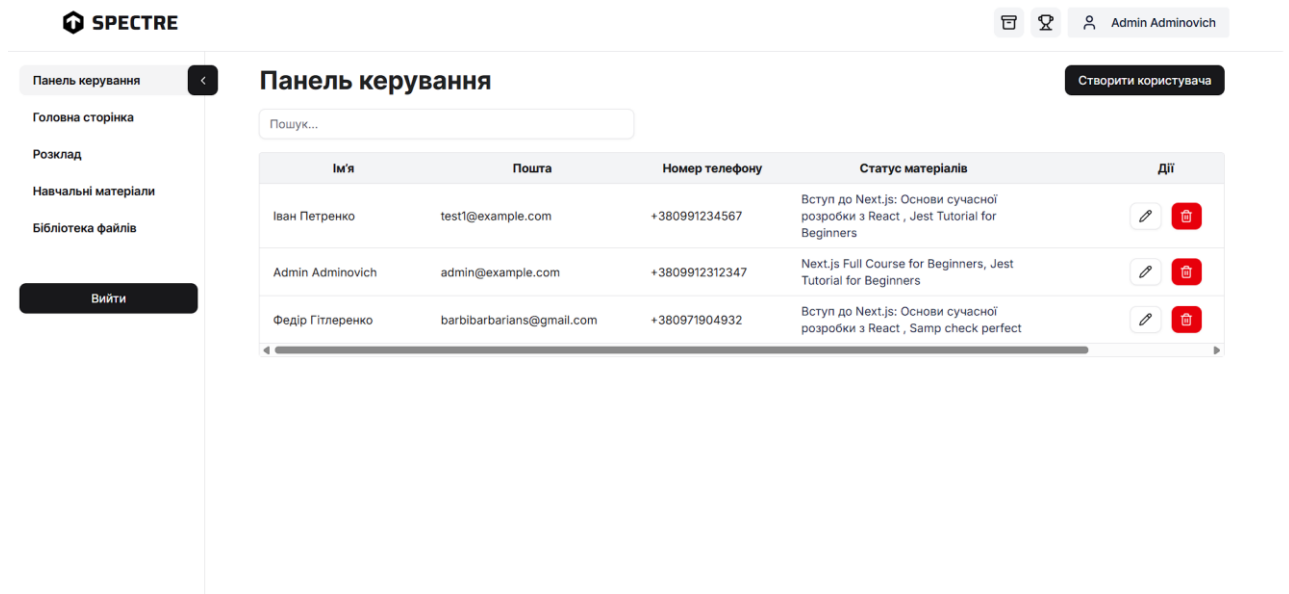


Рис. 3.27 – Адміністративна панель користувачів

На сторінці «*/dashboard/create-user*» реалізовано інтерфейс створення нового користувача із визначенням його ролі та базових облікових параметрів. Форма складається з полів для введення імені, прізвища, номера телефону та пароля. Для зручності передбачена можливість автоматичної генерації пароля, що дозволяє швидко створювати облікові записи без потреби вручну вигадувати надійні комбінації символів. У полі вибору ролі доступні лише дві опції: Інструктор або Студент, що обумовлено обмеженням функціоналу даної сторінки лише для створення користувачів, які безпосередньо беруть участь у навчальному процесі. Роль адміністратора може бути призначена лише через спеціальні внутрішні механізми керування системою доступу.

The screenshot shows the 'Створення користувача' (Create User) form in the SPECTRE system. The sidebar is identical to the previous screenshot. The form contains input fields for 'Email', 'Ім'я' (Name), 'Прізвище' (Surname), and 'Номер телефону' (Phone Number), each marked with a red asterisk. There is a 'Пароль' (Password) field with a 'Згенерувати' (Generate) button next to it. A 'Роль' (Role) dropdown menu is set to 'Студент'. At the bottom is a 'Створити користувача' (Create User) button.

Рис. 3.28 – Форма створення нового користувача

Крім того, у правому верхньому куті навігації є кнопка відкриття сповіщень. Вона показує інформацію про нещодавно створені або змінені матеріали, розклад тощо.

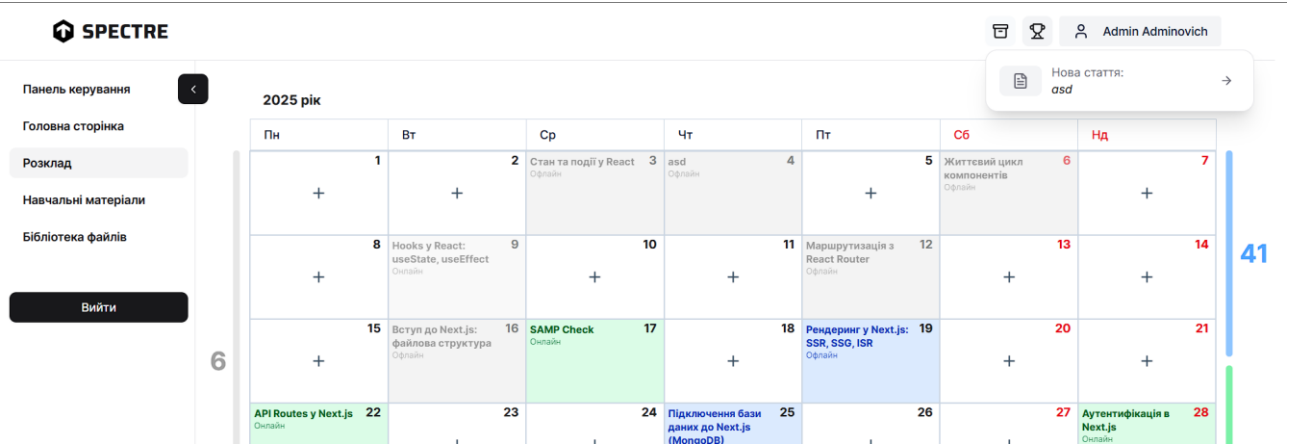


Рис. 3.29 – Панель сповіщень

Усі зображення створено на основі працюючого вебзастосунку та відображають реальний інтерфейс, доступний користувачам згідно з їх роллю та правами доступу.

3.3 Тестування роботи програмного комплексу

Тестування мобільного застосунку є критично важливим етапом розробки для забезпечення стабільної роботи та високої якості користувацького досвіду. Усі ключові функціональні сценарії були перевірені вручну шляхом проходження основного користувацького шляху.

Функціональне тестування

Тест 1: Авторизація користувача через електронну пошту

Передумови: користувач неавторизований;

Кроки: перехід на сторінку входу /sign-in; введення електронної пошти test@example.com та пароля StrongPass123, натискання кнопки входу;

Очікуваний результат: користувач успішно авторизується, з'являється повідомлення про успішний вхід, редирект на головну сторінку;

Фактичний результат: успішний вхід, сповіщення в кутку, відображення головної сторінки.

Тест 2: Виведення розкладу занять

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	56
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Передумови: авторизований студент, наявні заплановані заняття;

Кроки: перехід на сторінку /schedule;

Очікуваний результат: відображення розкладу на поточний місяць з підрахунками занять до/після місяця, можливість перегляду деталей;

Фактичний результат: розклад завантажується, статистика працює, модалка відкривається при натисканні.

Тест 3: Запис на заняття

Передумови: користувач на сторінці розкладу, є доступне майбутнє заняття;

Кроки: натискання на картку заняття, відкриття модалки, натискання кнопки "Записатися";

Очікуваний результат: користувач додається до заняття, сповіщення в кутку;

Фактичний результат: запис успішний, модалка оновлює статус.

Тест 4: Перегляд матеріалу та проходження тесту

Передумови: користувач має доступ до матеріалів;

Кроки: перехід на /materials/lecture/:id, проходження тесту та натискання кнопки завершення;

Очікуваний результат: після успішного проходження – зелена модалка, при провалі – інша;

Фактичний результат: обидва варіанти працюють коректно згідно з умовами.

Тест 5: Завантаження файлу у бібліотеку

Передумови: користувач має доступ до сторінки /library;

Кроки: обрати файл, натиснути кнопку "Завантажити", почекати завершення;

Очікуваний результат: файл з'являється у списку, стає доступним для завантаження;

Фактичний результат: файл збережено, доступний для завантаження.

Тест 6: Виведення досягнень

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	57
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Передумови: користувач відкриває сторінку /achievements;

Кроки: перехід на сторінку, очікування завантаження;

Очікуваний результат: список досягнень виводиться, позначаються завершені;

Фактичний результат: список відповідає даним користувача, інтерфейс відображається коректно.

Тест 7: Рольова авторизація

Передумови: користувач має роль інструктора або адміністратора;

Кроки: перехід на /materials/create, спроба створити новий матеріал;

Очікуваний результат: доступ до форми створення, перевірка полів;

Фактичний результат: сторінка доступна, створення матеріалу працює.

Тест 8: Надсилання заявки на створення облікового запису

Передумови: користувач неавторизований, на головній сторінці;

Кроки: заповнення форми заявки, натискання кнопки "Надіслати";

Очікуваний результат: форма очищається, з'являється повідомлення "Заявку надіслано";

Фактичний результат: форма працює, заявка надсилається успішно.

Тест 9: Робота сповіщень

Передумови: відбувається створення матеріалу або заняття (інструктор);

Кроки: створення нового ресурсу через відповідну форму;

Очікуваний результат: у правому верхньому кутку з'являється повідомлення;

Фактичний результат: сповіщення працюють для створення, редагування, видалення.

Висновки до розділу 3

В третьому розділі було зосереджено увагу на практичному використанні створеної вебплатформи, що охоплює всі аспекти взаємодії кінцевого користувача з системою. Було сформульовано мінімальні технічні вимоги до пристроїв користувачів та серверного середовища, необхідні для коректної роботи додатку. У розділі також представлено порядок розгортання системи,

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	58
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

включаючи використання хмарних платформ Vercel (для клієнтської частини) та Railway (для серверної логіки та бази даних). Такий вибір забезпечив легкість масштабування, гнучке тестування та безперервне розгортання оновлень із використанням CI/CD-підходу.

У візуальному аспекті платформи було реалізовано сучасний інтерфейс користувача, який динамічно адаптується до ролі: студент, інструктор або адміністратор. Навігаційна структура спроектована таким чином, щоб кожна роль мала лише той набір доступу, що відповідає її повноваженням, а також швидкий доступ до відповідних функцій. Користувачам без авторизації доступна лише заявка на створення акаунта, тоді як студентам надається функціонал перегляду розкладу, навчальних матеріалів, проходження тестів, перегляду досягнень і використання бібліотеки.

Усі ключові функції платформи – перегляд та керування розкладом, робота з навчальними матеріалами (лекції, статті, відео), проходження тестування, система досягнень, бібліотека файлів, а також панелі адміністратора та інструктора – були реалізовані з урахуванням принципів зручності, доступності та інформаційної безпеки. Особливу увагу приділено мікроінтеракціям – сповіщенням про дії (наприклад, успішне збереження, помилки доступу), динамічним модальним вікнам для запису на заняття, та адаптивному інтерфейсу.

У підрозділі тестування було проведено дев'ять ключових функціональних тестів, які охоплюють основні сценарії використання платформи. Всі перевірки підтвердили стабільність роботи як клієнтської, так і серверної частин. Система коректно реагує на типові дії користувача, успішно обробляє запити, а також інформує про результат кожної взаємодії.

Серед тестованих функцій – авторизація, навігація сторінками, запис/відписки на заняття, створення та проходження навчальних матеріалів із тестуванням, використання бібліотеки файлів, робота досягнень та адміністрування. Усі тести були виконані на активному продакшн-деплойменті,

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	59
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

що дозволило впевнено підтвердити відповідність функціоналу вимогам до проєкту.

Таким чином, результати третього розділу демонструють, що вебплатформа досягла високого рівня готовності до впровадження в навчальні процеси. Вона забезпечує не лише базові сценарії організації дистанційного чи гібридного навчання, а й містить розвинену систему зворотного зв'язку, гнучке керування контентом, безперервне вдосконалення на основі ролей користувачів та автоматизовану систему досягнень. Все це формує сучасний, зручний та інформативний інструмент для ефективної організації освітнього процесу на різних рівнях – від індивідуального навчання до групових курсів з централізованим керуванням.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		60

ВИСНОВКИ

В результаті виконання випускної кваліфікаційної роботи було спроектовано, реалізовано та протестовано вебплатформу для організації та проведення навчальних курсів, яка забезпечує повний цикл взаємодії між студентами, інструкторами та адміністраторами освітнього процесу. Протягом усіх етапів реалізації проєкту було використано сучасний технологічний стек, який включає Next.js, NestJS, MongoDB, TypeScript та WebSocket, що дозволило створити масштабовану, продуктивну і зручну в експлуатації систему.

У першому розділі було обґрунтовано вибір архітектурних рішень і технологій, а також проведено аналіз аналогічних рішень на ринку. Це дало змогу виділити ключові переваги платформи: підтримку ролей, динамічну маршрутизацію, гнучку систему керування контентом, можливість створення різних типів навчальних матеріалів і підтримку багаторівневого доступу. Розглянуто принципи використання ShadCN як бібліотеки UI-компонентів, що забезпечило швидку розробку візуально привабливих і адаптивних інтерфейсів. Важливе місце відведено моделі розподілу прав доступу, що реалізована через декоратори `@Roles` у NestJS – це надало системі чітку і гнучку авторизаційну модель.

У другому розділі була реалізована логіка взаємодії між усіма ключовими модулями: автентифікація, управління розкладом занять, навчальними матеріалами, досягненнями та файлами, сповіщення через WebSocket. Вся бізнес-логіка організована відповідно до принципів модульності та розділення відповідальностей. Базу даних було спроектовано з урахуванням ієрархії документів і типів контенту, а використання MongoDB-дискримінаторів забезпечило елегантну реалізацію збереження гетерогенних матеріалів в одній колекції. Важливою частиною також стала інтеграція системи повідомлень у реальному часі через WebSocket-шлюз, що покращило користувацький досвід і знизило навантаження на серверну інфраструктуру.

Третій розділ був присвячений практичному демонструванню функціоналу та тестуванню платформи. Було сформовано вимоги до середовища

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	61
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

розгортання та описано процес деплою на Vercel (frontend) і Railway (backend). Докладно продемонстровано інтерфейс користувача для всіх ролей, включаючи сценарії взаємодії зі сторінками матеріалів, розкладу, бібліотеки, досягнень і адміністративної панелі. Проведене функціональне тестування підтвердило стабільну та коректну роботу всіх підсистем. Протестовано автентифікацію, роботу з навчальними матеріалами, взаємодію з розкладом, проходження тестів, взаємодію з файлами, зміну ролей і навігацію. Результати тестування засвідчили, що система готова до використання в реальному навчальному середовищі.

Загалом, розроблена платформа повністю відповідає початково поставленим вимогам і має чітку архітектуру, зручний інтерфейс та розширювану модульну структуру. Її можна легко адаптувати до потреб різних навчальних закладів або організацій, додавши нові функції – мобільний застосунок, інтеграцію з BankID, підсистему аналітики прогресу користувачів або внутрішні комунікаційні інструменти. Запропонована архітектура також передбачає подальше масштабування за допомогою мікросервісного підходу або використання хмарних serverless-технологій у майбутньому.

Результати кваліфікаційної роботи можуть стати основою для реального впровадження вебплатформи в освітні програми або використання її як основи для створення сертифікованих систем дистанційного навчання, що відповідають сучасним державним і міжнародним стандартам.

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. React. React Documentation [Електронний ресурс]. – Режим доступу: <https://react.dev/docs/getting-started>
2. Next.js. Next.js Documentation [Електронний ресурс]. – Режим доступу: <https://nextjs.org/docs>
3. NestJS. NestJS Official Documentation [Електронний ресурс]. – Режим доступу: <https://docs.nestjs.com/>
4. MongoDB. MongoDB Manual: Data Modeling and Schema Design [Електронний ресурс]. – Режим доступу: <https://www.mongodb.com/docs/manual/core/data-modeling-introduction/>
5. Mongoose. Mongoose Official Documentation [Електронний ресурс]. – Режим доступу: <https://mongoosejs.com/docs/guide.html>
6. Tailwind CSS. Tailwind CSS Documentation [Електронний ресурс]. – Режим доступу: <https://tailwindcss.com/docs>
7. Shadcn UI. Shadcn UI Documentation [Електронний ресурс]. – Режим доступу: <https://ui.shadcn.com/docs>
8. Formik. Formik – React Forms Library [Електронний ресурс]. – Режим доступу: <https://formik.org/docs/overview>
9. Yup. Yup Validation Library [Електронний ресурс]. – Режим доступу: <https://github.com/jquense/yup>
10. JWT.io. JSON Web Token Introduction and Documentation [Електронний ресурс]. – Режим доступу: <https://jwt.io/introduction>
11. Passport.js. Passport Authentication Middleware [Електронний ресурс]. – Режим доступу: <http://www.passportjs.org/docs/>
12. Socket.IO. Socket.IO Documentation [Електронний ресурс]. – Режим доступу: <https://socket.io/docs/v4/>
13. React Query. React Query Documentation [Електронний ресурс]. – Режим доступу: <https://tanstack.com/query/latest>
14. Radix UI. Radix UI Primitives for React [Електронний ресурс]. – Режим доступу: <https://www.radix-ui.com/>

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ІЗ	63
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

- 15.Vercel. Deploy Next.js on Vercel [Електронний ресурс]. – Режим доступу: <https://vercel.com/docs/concepts/deployments/overview>
- 16.Railway. Railway.app Documentation for Backend Deployment [Електронний ресурс]. – Режим доступу: <https://docs.railway.app/>
- 17.Google. Web Vitals – User Experience Metrics [Електронний ресурс]. – Режим доступу: <https://web.dev/vitals/>
- 18.MDN Web Docs. Accessibility Guide for Web Applications [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Learn/Accessibility>
- 19.Tiptap. Rich Text Editor Framework Documentation [Електронний ресурс]. – Режим доступу: <https://tiptap.dev/>
- 20.TypeScript. TypeScript Handbook [Електронний ресурс]. – Режим доступу: <https://www.typescriptlang.org/docs/handbook/intro.html>
- 21.Jose. JavaScript Object Signing and Encryption (JOSE) Specification [Електронний ресурс]. – Режим доступу: <https://datatracker.ietf.org/wg/jose/about/>
- 22.Jsonwebtoken. JSON Web Token (JWT) Library for Node.js [Електронний ресурс]. – Режим доступу: <https://github.com/auth0/node-jwebtoken>
- 23.React Toastify. React Notification Library [Електронний ресурс]. – Режим доступу: <https://fkhadra.github.io/react-toastify/introduction>
- 24.Lucide React. Lucide Icons for React [Електронний ресурс]. – Режим доступу: <https://lucide.dev/docs/react/>
- 25.ESLint. ESLint JavaScript Linter Documentation [Електронний ресурс]. – Режим доступу: <https://eslint.org/docs/latest/>
- 26.Prettier. Code Formatter Documentation [Електронний ресурс]. – Режим доступу: <https://prettier.io/docs/en/index.html>
- 27.Jest. Jest Testing Framework Documentation [Електронний ресурс]. – Режим доступу: <https://jestjs.io/docs/getting-started>
- 28.Class Variance Authority. Utility for Managing CSS Variants [Електронний ресурс]. – Режим доступу: <https://cva.style/>

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ІПЗ	64
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		

- 29.Multer. Node.js Middleware for Handling multipart/form-data [Електронний ресурс]. – Режим доступу: <https://github.com/expressjs/multer>
- 30.Bcryptjs. Password Hashing Library for Node.js [Електронний ресурс]. – Режим доступу: <https://github.com/dcodeIO/bcrypt.js>

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ІЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		65

ДОДАТКИ

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		66

Лістинг фрагменту коду відповідального за створення навчального матеріалу:

Додаток А

```
async create(dto: CreateMaterialDto): Promise<Material> {  
  
    let createdMaterial: MaterialUnion;  
  
    switch (dto.kind) {  
  
        case MaterialType.Article:  
  
        case MaterialType.Lecture:  
  
        case MaterialType.Video:  
  
            createdMaterial =  
  
            await this.materialModel.discriminators[dto.kind].create(dto);  
  
            break;  
  
        default:  
  
            throw new BadRequestException('Invalid material kind');  
  
    }  
  
    this.notificationGateway.sendNewNotification(  
  
        createdMaterial,  
  
        'material',  
  
        'create',  
  
    );  
  
    return createdMaterial;  
  
}
```

		Дзюбчук Д.Р.			ІІЗ.КР.Б-121-25-ІІЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		67

Лістинг фрагменту коду відповідального за розрахунок кількості записів в попередньому місяці:

```
function countLessonsBeforeSelectedMonth(
  schedules: Schedule[],
  selectedYear: number,
  selectedMonth: number,
  now: Date
) {
  const targetDateBefore = new Date(selectedYear, selectedMonth, 1);

  let offline = 0;
  let online = 0;
  let passed = 0;

  for (const s of schedules) {
    const lessonDate = new Date(s.date);

    if (isBeforeYearMonth(lessonDate, targetDateBefore)) {
      if (isBeforeDay(lessonDate, now)) {
        passed++;
      } else {
        if (s.meetingType === "Offline") offline++;
        if (s.meetingType === "Online") online++;
      }
    }
  }

  return { offline, online, passed };
}
```

		Дзюбчук Д.Р.			ІПЗ.КР.Б-121-25-ПЗ	
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		68