

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-  
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ  
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

## КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Веб-орієнтована інформаційна система супроводу навчального  
процесу в центрах підвищення кваліфікації»  
на здобуття освітнього ступеня бакалавр  
за спеціальності 126 Інформаційні системи та технології  
(код, найменування спеціальності)  
освітньо-професійної програми Інформаційні системи та технології  
(назва)

*Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання на  
відповідне джерело*

\_\_\_\_\_  
(підпис)

Алла МЕЛЬНИЧЕНКО  
(ім'я, ПРІЗВИЩЕ здобувача)

Виконав: здобувач вищої освіти гр. ІСД-41

Алла МЕЛЬНИЧЕНКО  
(ім'я, ПРІЗВИЩЕ)

Керівник:  
к.т.н., доцент

Олег СЕНЬКОВ  
(ім'я, ПРІЗВИЩЕ)

Рецензент:

\_\_\_\_\_  
(ім'я, ПРІЗВИЩЕ)

Київ 2026

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**Навчально-науковий інститут інформаційних технологій**

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти Бакалавр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

**ЗАТВЕРДЖУЮ**

Завідувач кафедру ІСТ

\_\_\_\_\_ Каміла СТОРЧАК

«\_\_\_\_\_» \_\_\_\_\_ 2026 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Мельниченко Алла Вікторівна

---

*(прізвище, ім'я, по батькові здобувача)*

1. Тема кваліфікаційної роботи: Веб-орієнтована інформаційна система супроводу навчального процесу в центрах підвищення кваліфікації

керівник кваліфікаційної роботи Олег Сеньков, к.т.н., доцент

*(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)*

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «20» 02.2026р. №51

2. Строк подання кваліфікаційної роботи «01» червня 2026 р р.

3. Вихідні дані до кваліфікаційної роботи: Принципи побудови веб-орієнтованої інформаційної системи супроводу навчального процесу в центрах підвищення кваліфікації.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз сучасних систем для супроводу навчального процесу в центрах підвищення кваліфікації.

2.Формування вимог до веб-орієнтованої інформаційної системи супроводу навчального процесу.

3. Проектування архітектури програмного забезпечення системи.

4. Проектування бази даних інформаційної системи.

5. Розробка інтерфейсу користувача для забезпечення взаємодії з системою.

6. Інтеграція з системами управління базами даних для забезпечення ефективного доступу до інформації.

5. Перелік ілюстративного матеріалу: презентація

6. Дата видачі завдання «20» лютого 2026 р.

## КАЛЕНДАРНИЙ ПЛАН

| №<br>№<br>з/п | Назва етапів<br>кваліфікаційної роботи                            | Строк виконання<br>етапів роботи | Примітка |
|---------------|-------------------------------------------------------------------|----------------------------------|----------|
| 1             | Вступ, визначення мети, об'єкта та предмета дослідження           | 02.02–16.02.2026                 | Виконано |
| 2             | Аналіз предметної області та особливостей навчального процесу     | 17.02–02.03.2026                 | Виконано |
| 3             | Аналіз існуючих LMS та обґрунтування доцільності розробки системи | 03.03–16.03.2026                 | Виконано |
| 4             | Формування функціональних і нефункціональних вимог до системи     | 17.03–31.03.2026                 | Виконано |
| 5             | Проектування архітектури веб-орієнтованої інформаційної системи   | 01.04–14.04.2026                 | Виконано |
| 6             | Проектування бази даних, бізнес-процесів та користувацьких ролей  | 15.04–30.04.2026                 | Виконано |
| 7             | Оформлення роботи: вступ, висновки, реферат                       | 01.05–15.05.2026                 | Виконано |
| 8             | Розробка демонстраційних матеріалів                               | 16.05–01.06.2026                 | Виконано |

Здобувачка вищої освіти \_\_\_\_\_  
(підпис)

Алла Мельниченко  
(Ім'я, ПРІЗВИЩЕ)

Керівник  
кваліфікаційної роботи \_\_\_\_\_  
(підпис)

Олег СЕНЬКОВ  
(Ім'я, ПРІЗВИЩЕ)





## РЕФЕРАТ

Дипломна робота: 79 с., 35 рис., 35 джерел.

*Об'єктом дослідження* є процес організації та супроводу навчального процесу в центрах підвищення кваліфікації.

*Предметом дослідження* є методи, моделі та програмні засоби розробки веб-орієнтованих інформаційних систем для автоматизації освітньої діяльності.

*Метою дипломної роботи* є розробка веб-орієнтованої інформаційної системи супроводу навчального процесу в центрах підвищення кваліфікації, яка забезпечує автоматизацію основних процесів управління навчанням.

У роботі проведено аналіз предметної області та визначено особливості організації навчального процесу в центрах підвищення кваліфікації. Здійснено огляд існуючих систем управління навчанням та виявлено їхні обмеження щодо адаптації до короткострокових освітніх програм.

Сформовано функціональні та нефункціональні вимоги до інформаційної системи. Розроблено архітектуру системи на основі клієнт-серверного підходу, що забезпечує масштабованість та ефективність роботи. Спроектовано структуру бази даних із використанням реляційної моделі.

Реалізовано програмне забезпечення системи з використанням сучасних технологій веб-розробки, зокрема Django для серверної частини та React для клієнтської частини. Для зберігання даних використано PostgreSQL.

У системі реалізовано модулі управління користувачами, курсами, тестуванням та сертифікацією. Забезпечено взаємодію між компонентами через REST API.

Проведено тестування системи, результати якого підтвердили її працездатність, стабільність та відповідність поставленим вимогам. Система демонструє достатній рівень продуктивності та може бути використана у практичній діяльності центрів підвищення кваліфікації.

Практичне значення роботи полягає у можливості впровадження розробленої інформаційної системи для автоматизації навчального процесу, що дозволяє підвищити ефективність управління освітньою діяльністю та зменшити адміністративне навантаження.

**КЛЮЧОВІ СЛОВА:** інформаційна система, веб-орієнтована система, навчальний процес, система управління навчанням, база даних, REST API, автоматизація, тестування, сертифікація.

## ABSTRACT

The text part of the qualifying work for obtaining a master's degree: 79 pages, 35 figures, 35 sources.

The diploma thesis is devoted to the development of a web-based information system for supporting the educational process in professional development centers.

The object of the research is the process of organization and support of the educational process in professional development centers.

The subject of the research is methods, models, and software tools for developing web-based information systems for automating educational activities.

The aim of the thesis is to design and implement a web-oriented information system that ensures automation of key processes in managing the educational process.

The study analyzes the specifics of the educational process in professional development centers and identifies the limitations of existing Learning Management Systems in adapting to short-term training programs.

Based on the analysis, functional and non-functional requirements for the system were formulated. The system architecture was designed using a client-server approach, ensuring scalability and efficiency. A relational database structure was developed to provide data integrity and consistency.

The software system was implemented using modern web development technologies, including Django for the backend and React for the frontend. The database management system is based on PostgreSQL.

The system includes modules for user management, course management, testing, and certification. Interaction between components is implemented via REST API.

Testing of the developed system confirmed its functionality, stability, and compliance with the defined requirements. The system demonstrates sufficient performance and can be effectively applied in professional development centers.

The practical significance of the work lies in the possibility of implementing the developed system to improve the efficiency of educational process management and reduce administrative workload.

Keywords: information system, web-based system, learning process, learning management system, database, REST API, automation, testing, certification.

## ЗМІСТ

|                                                                           |           |
|---------------------------------------------------------------------------|-----------|
| ВСТУП.....                                                                | 11        |
| РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО СИСТЕМИ ....              | 14        |
| 1.1. Особливості навчального процесу в центрах підвищення кваліфікації .. | 14        |
| 1.2. Аналіз існуючих інформаційних систем та LMS .....                    | 17        |
| 1.3. Формування функціональних та нефункціональних вимог .....            | 23        |
| РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....        | 29        |
| 2.1. Загальна архітектура системи .....                                   | 29        |
| 2.2. Проєктування бази даних.....                                         | 37        |
| 2.3. Моделювання бізнес-процесів та користувацьких ролей .....            | 45        |
| РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ.....                           | 56        |
| 3.1. Вибір технологій та інструментів розробки.....                       | 56        |
| 3.2. Реалізація основних модулів.....                                     | 62        |
| 3.3. Тестування та оцінка ефективності .....                              | 73        |
| <b>ВИСНОВОК .....</b>                                                     | <b>84</b> |
| ПЕРЕЛІК ПОСИЛАНЬ .....                                                    | 87        |
| ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ .....                                            | 90        |

## ВСТУП

У сучасних умовах розвитку інформаційного суспільства та цифрової трансформації освіти особливої актуальності набуває впровадження веб-орієнтованих інформаційних систем для автоматизації освітньої діяльності. Активне використання дистанційних технологій навчання, електронних освітніх ресурсів та онлайн-комунікації суттєво змінює підходи до організації навчального процесу, зокрема у сфері підвищення кваліфікації та професійного розвитку фахівців.

Центри підвищення кваліфікації відіграють важливу роль у системі неперервної освіти, забезпечуючи професійне вдосконалення, перепідготовку та розвиток компетентностей слухачів. Специфікою діяльності таких центрів є короткостроковий характер освітніх програм, необхідність швидкого оновлення навчальних матеріалів, проведення дистанційного навчання, оперативного контролю результатів та автоматизованої сертифікації.

Традиційні підходи до організації навчального процесу часто супроводжуються значним адміністративним навантаженням, складністю управління навчальними ресурсами та недостатнім рівнем автоматизації. Використання універсальних систем управління навчанням не завжди забезпечує необхідний рівень адаптації до специфіки короткострокових програм підвищення кваліфікації, що зумовлює необхідність створення спеціалізованих веб-орієнтованих інформаційних систем.

Актуальність теми дипломної роботи обумовлена потребою у створенні сучасної інформаційної системи супроводу навчального процесу, яка забезпечує автоматизацію управління курсами, тестуванням, навчальними матеріалами та сертифікацією слухачів у центрах підвищення кваліфікації.

Об'єктом дослідження є процес організації та супроводу навчального процесу в центрах підвищення кваліфікації.

Предметом дослідження є методи, моделі та програмні засоби розробки веб-орієнтованих інформаційних систем для автоматизації освітньої діяльності.

Метою дипломної роботи є розробка веб-орієнтованої інформаційної системи супроводу навчального процесу в центрах підвищення кваліфікації, яка забезпечує автоматизацію основних процесів управління навчанням.

Для досягнення поставленої мети необхідно провести аналіз предметної області та особливостей організації навчального процесу в центрах підвищення кваліфікації, дослідити існуючі системи управління навчанням із визначенням їхніх переваг і недоліків, а також сформулювати функціональні та нефункціональні вимоги до інформаційної системи. Окрім цього, необхідно розробити архітектуру веб-орієнтованої інформаційної системи та спроектувати структуру бази даних. Практична частина дослідження передбачає реалізацію програмних модулів системи, забезпечення REST API-взаємодії між frontend- і backend-компонентами, а також впровадження механізмів автентифікації та контролю доступу. Завершальним етапом є проведення тестування програмного забезпечення та оцінювання ефективності функціонування системи.

У процесі розробки системи використано Next.js та React для реалізації клієнтської частини, Django і Django REST Framework для серверної частини, а також PostgreSQL для організації зберігання даних. Для взаємодії компонентів системи використовується REST API, а для реалізації автентифікації користувачів — JWT-механізми.

Практичне значення одержаних результатів полягає у можливості використання розробленої системи NexLearn для автоматизації навчального процесу в центрах підвищення кваліфікації. Використання системи сприяє підвищенню ефективності управління освітньою діяльністю, автоматизації процесів тестування та сертифікації, зменшенню адміністративного навантаження, забезпеченню централізованого

управління навчальними ресурсами та покращенню доступності дистанційного навчання.

Інформаційна система NexLearn реалізована на основі сучасної клієнт-серверної архітектури та забезпечує підтримку багаторольової моделі доступу, управління курсами, автоматизованого тестування і генерації сертифікатів. Крім того, система містить адаптивний вебінтерфейс і підтримує безпечну взаємодію між компонентами через REST API. Структурно дипломна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

У першому розділі проведено аналіз предметної області, досліджено особливості навчального процесу в центрах підвищення кваліфікації, виконано аналіз існуючих LMS та сформовано вимоги до системи.

У другому розділі виконано проєктування архітектури системи, структури бази даних, бізнес-процесів та користувацьких ролей.

У третьому розділі розглянуто практичну реалізацію програмного забезпечення, вибір технологій розробки, реалізацію основних модулів та результати тестування системи.

## **РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО СИСТЕМИ**

### **1.1. Особливості навчального процесу в центрах підвищення кваліфікації**

У сучасних умовах розвитку цифрових технологій та інформаційного суспільства система післядипломної освіти набуває особливого значення. Одним із ключових елементів такої системи є центри підвищення кваліфікації, діяльність яких спрямована на професійний розвиток, перепідготовку та вдосконалення компетентностей фахівців різних галузей.

Навчальний процес у центрах підвищення кваліфікації має низку специфічних особливостей, які відрізняють його від традиційної системи освіти. Насамперед це пов'язано з короткостроковим характером освітніх програм, необхідністю швидкого оновлення навчального контенту, високою динамічністю навчальних курсів та широким використанням дистанційних технологій навчання.

Основними особливостями навчального процесу в центрах підвищення кваліфікації є короткостроковість навчальних програм, гнучкість організації освітнього процесу, модульна структура курсів та орієнтація на формування практичних компетентностей. Значна увага приділяється дистанційній і змішаній формам навчання, оперативному оцінюванню результатів навчальної діяльності та автоматизованій сертифікації слухачів. На відміну від закладів вищої освіти, де реалізація освітніх програм триває декілька років, навчання у центрах підвищення кваліфікації може відбуватися протягом декількох днів або місяців, що зумовлює необхідність високого рівня організації освітнього процесу та автоматизації основних процедур управління навчанням.

Важливою особливістю діяльності таких центрів є значна кількість слухачів, які одночасно проходять навчання за різними програмами. У зв'язку з цим виникає потреба у централізованому управлінні навчальними матеріалами, автоматизованому

обліку слухачів, контролі проходження курсів, організації тестування, формуванні звітності та генерації сертифікатів.

У сучасних умовах значна частина освітнього процесу реалізується із використанням дистанційних технологій навчання, що забезпечує доступність освітніх послуг незалежно від місця перебування слухачів, сприяє використанню електронних освітніх ресурсів та автоматизації взаємодії між викладачами й слухачами. Дистанційне навчання передбачає застосування веборієнтованих інформаційних систем, які забезпечують управління курсами, доступ до навчальних матеріалів, проведення тестування, комунікацію між учасниками освітнього процесу та збереження результатів навчання.

Однією з ключових особливостей діяльності центрів підвищення кваліфікації є необхідність оперативного оновлення навчальних матеріалів відповідно до змін у професійній сфері. Це вимагає впровадження гнучкої системи управління контентом, підтримки електронних ресурсів, можливості швидкого редагування курсів та централізованого доступу до навчальних матеріалів.

Крім того, сучасний навчальний процес потребує автоматизованого контролю знань слухачів, що реалізується за допомогою онлайн-тестування, інтерактивних завдань, автоматизованого оцінювання та електронної сертифікації. Автоматизація тестування дозволяє зменшити навантаження на викладачів, підвищити об'єктивність оцінювання, прискорити обробку результатів і забезпечити оперативне формування сертифікатів.

У процесі організації навчального процесу важливе значення має інформаційна взаємодія між користувачами системи, основними категоріями яких є адміністратори, викладачі та слухачі. Кожна категорія користувачів має власні функції та рівень доступу до інформаційної системи.

Адміністратор відповідає за управління користувачами, налаштування системи, контроль функціонування платформи та управління ролями доступу.

Викладач здійснює створення курсів, завантаження навчальних матеріалів, організацію тестування та оцінювання результатів навчання.

Слухач виконує перегляд навчальних матеріалів, проходження тестування, взаємодію з навчальними курсами та отримання сертифікатів.

Табл. 1.1. Основні учасники навчального процесу

| <b>Категорія користувача</b> | <b>Основні функції</b>               |
|------------------------------|--------------------------------------|
| Адміністратор                | Управління системою та користувачами |
| Викладач                     | Організація навчального процесу      |
| Слухач                       | Проходження навчання                 |

У сучасних умовах важливим аспектом функціонування освітнього середовища є забезпечення інформаційної безпеки. Веборієнтовані системи повинні підтримувати механізми автентифікації користувачів, контролю доступу, захисту персональних даних, безпечної передачі інформації та протидії несанкціонованому доступу до ресурсів системи.

Важливим фактором ефективності інформаційної системи є також її масштабованість і адаптивність. Система має забезпечувати одночасну роботу значної кількості користувачів, підтримувати можливість швидкого оновлення функціоналу, інтеграцію із зовнішніми сервісами та надавати адаптивний вебінтерфейс для роботи на різних типах пристроїв.

З урахуванням особливостей діяльності центрів підвищення кваліфікації виникає потреба у створенні спеціалізованої веборієнтованої інформаційної системи, яка забезпечуватиме автоматизацію управління навчальним процесом, централізоване управління курсами, підтримку дистанційного навчання, автоматизоване тестування та генерацію сертифікатів. Крім того, така система повинна підтримувати

багаторольову модель доступу та забезпечувати безпечну взаємодію між користувачами.

## **1.2. Аналіз існуючих інформаційних систем та LMS**

У сучасних умовах цифровізації освіти значного поширення набули системи управління навчанням (Learning Management System, LMS), які забезпечують автоматизацію організації освітнього процесу, управління навчальними матеріалами, тестування та взаємодію між учасниками навчання.

Системи управління навчанням (LMS) широко використовуються у закладах вищої освіти, корпоративному навчанні, системах дистанційної освіти, центрах підвищення кваліфікації та на онлайн-освітніх платформах. Їх застосування забезпечує ефективну організацію та автоматизацію освітнього процесу в умовах сучасного цифрового середовища.

Основними функціями LMS є управління курсами, реєстрація користувачів і контроль доступу до ресурсів системи. Крім того, такі системи забезпечують організацію дистанційного навчання, проведення тестування, формування звітності та підтримку комунікації між учасниками освітнього процесу.

Найбільш поширеними LMS є: Moodle, Google Classroom, Canvas, Blackboard, Open edX.

Для визначення доцільності розробки власної інформаційної системи було проведено аналіз існуючих LMS та їх функціональних можливостей.

Табл. 1.2. Порівняння популярних LMS

| <b>LMS</b>       | <b>Переваги</b>                   | <b>Недоліки</b>            |
|------------------|-----------------------------------|----------------------------|
| Moodle           | Відкритий код, великий функціонал | Складність адміністрування |
| Google Classroom | Простота використання             | Обмежена кастомізація      |
| Canvas           | Сучасний інтерфейс                | Частина функцій платна     |
| Blackboard       | Потужний функціонал               | Висока вартість            |
| Open edX         | Масштабованість                   | Складність розгортання     |

Однією з найбільш відомих систем управління навчанням є Moodle. Система має відкритий програмний код і забезпечує широкі можливості для організації освітнього процесу, зокрема підтримує створення навчальних курсів, проведення тестування, управління користувачами, функціонування форумів та реалізацію системи оцінювання результатів навчання.

Табл. 1.3. Основні характеристики Moodle

| <b>Характеристика</b> | <b>Опис</b>                             |
|-----------------------|-----------------------------------------|
| Тип системи           | Open Source LMS                         |
| Технології            | PHP, MySQL/PostgreSQL                   |
| Переваги              | Гнучкість, велика спільнота             |
| Недоліки              | Складний інтерфейс, високе навантаження |

Основними перевагами Moodle є відкритий програмний код, наявність великої кількості додаткових модулів, підтримка дистанційного навчання та можливість гнучкого налаштування курсів відповідно до потреб освітнього процесу. Водночас система характеризується певними недоліками, серед яких складність

адміністрування, перевантажений інтерфейс, значні вимоги до серверних ресурсів і недостатня адаптованість до специфіки короткострокових курсів.

Популярною платформою для організації дистанційного навчання є також Google Classroom, яка інтегрується із сервісами Google та забезпечує можливості управління навчальними завданнями, комунікації між учасниками освітнього процесу, обміну файлами й організації навчальних груп.

Табл. 1.4. Основні характеристики Google Classroom

| Характеристика | Опис                  |
|----------------|-----------------------|
| Тип системи    | Cloud LMS             |
| Інтеграція     | Google Workspace      |
| Переваги       | Простота використання |
| Недоліки       | Обмежений функціонал  |

Перевагами Google Classroom є простота використання, швидке налаштування освітнього середовища, інтеграція із сервісом Google Drive та підтримка дистанційної роботи користувачів. Водночас система має певні обмеження, зокрема недостатні можливості кастомізації, відсутність розвинених механізмів тестування та обмежену підтримку складної бізнес-логіки.

Сучасною системою управління навчанням є також Canvas, яка реалізована на основі хмарної архітектури та підтримує REST API. Система орієнтована на організацію дистанційного навчання й забезпечує підтримку модульної структури курсів, проведення тестування, аналітики освітнього процесу та мобільного доступу до навчальних ресурсів.

Табл. 1.5. Основні характеристики Canvas

| Характеристика | Опис                   |
|----------------|------------------------|
| Тип системи    | Cloud LMS              |
| Технології     | Ruby on Rails          |
| Переваги       | Сучасний інтерфейс     |
| Недоліки       | Частина функцій платна |

Основними перевагами Canvas є сучасний інтерфейс користувача, підтримка REST API, адаптивний дизайн та зручність використання в умовах дистанційного навчання. Водночас система має певні недоліки, серед яких комерційна модель використання, залежність від хмарної інфраструктури та складність локального розгортання й адміністрування. Blackboard є однією з найбільш функціональних корпоративних LMS, яка використовується переважно у великих освітніх організаціях.

Табл. 1.6. Основні характеристики Blackboard

| Характеристика | Опис                |
|----------------|---------------------|
| Тип системи    | Enterprise LMS      |
| Переваги       | Потужний функціонал |
| Недоліки       | Висока вартість     |

Система забезпечує підтримку управління курсами, аналітики освітнього процесу, проведення тестування, ведення електронного журналу та інтеграції із зовнішніми сервісами. Водночас до основних недоліків Blackboard належать висока вартість використання, складність упровадження та значні вимоги до програмно-апаратної інфраструктури.

Open edX є платформою для масових онлайн-курсів та дистанційного навчання.

Табл. 1.7. Основні характеристики Open edX

| Характеристика | Опис                       |
|----------------|----------------------------|
| Тип системи    | Open Source LMS            |
| Переваги       | Масштабованість            |
| Недоліки       | Складність адміністрування |

Перевагами Open edX є підтримка великої кількості користувачів, масштабованість системи, можливість роботи з відеоконтентом та наявність відкритого програмного коду. Водночас система характеризується складністю розгортання й адміністрування, а також високими вимогами до серверної інфраструктури.

Проведений аналіз показав, що більшість існуючих систем управління навчанням орієнтовані переважно на довготривале навчання, універсальні освітні процеси та великі освітні організації. Натомість для центрів підвищення кваліфікації особливого значення набувають можливість швидкого створення курсів, простота адміністрування, автоматизована сертифікація, підтримка короткострокових програм, адаптивний вебінтерфейс і гнучке управління ролями користувачів.

Табл. 1.8. Порівняння LMS із потребами центрів підвищення кваліфікації

| <b>Критерій</b>          | <b>Існуючі LMS</b>     | <b>Потреби центрів підвищення кваліфікації</b> |
|--------------------------|------------------------|------------------------------------------------|
| Короткострокові курси    | Частково підтримуються | Повна підтримка                                |
| Автоматична сертифікація | Обмежено               | Необхідна                                      |
| Простота використання    | Середня                | Висока                                         |
| Гнучкість кастомізації   | Обмежена               | Висока                                         |
| REST API                 | Не у всіх системах     | Необхідний                                     |
| Адаптивний інтерфейс     | Частково               | Обов'язковий                                   |

Аналіз існуючих систем управління навчанням показав, що універсальні LMS характеризуються широким функціоналом, проте водночас є надлишково складними, не повністю адаптованими до організації короткострокових курсів, потребують значних програмно-апаратних ресурсів і мають певні обмеження щодо реалізації спеціалізованої бізнес-логіки.

У зв'язку з цим виникає потреба у створенні спеціалізованої веборієнтованої інформаційної системи, яка враховуватиме особливості діяльності центрів підвищення кваліфікації, забезпечуватиме автоматизацію навчального процесу, підтримку дистанційного навчання та автоматизоване тестування. Крім того, така система повинна реалізовувати механізми генерації сертифікатів, а також забезпечувати масштабованість і безпечність функціонування.

Таким чином, проведений аналіз існуючих LMS підтверджує доцільність розробки власної веб-орієнтованої інформаційної системи NexLearn для супроводу навчального процесу в центрах підвищення кваліфікації.

### **1.3. Формування функціональних та нефункціональних вимог**

Одним із важливих етапів розробки веб-орієнтованої інформаційної системи є формування функціональних та нефункціональних вимог. Саме вимоги визначають основні характеристики системи, її функціональні можливості, рівень продуктивності, безпечності та зручності використання.

Формування вимог до інформаційної системи NexLearn здійснювалося на основі аналізу предметної області, особливостей діяльності центрів підвищення кваліфікації, дослідження існуючих систем управління навчанням, потреб учасників освітнього процесу та сучасних вимог до веборієнтованих інформаційних систем.

У процесі аналізу було визначено основні категорії користувачів системи, до яких належать адміністратор, викладач і слухач. Кожна категорія користувачів має власний набір функціональних можливостей та відповідний рівень доступу до інформаційних ресурсів системи.

До основних функціональних вимог інформаційної системи належать підтримка реєстрації та авторизації користувачів, управління ролями доступу, створення та редагування курсів, управління навчальними матеріалами, організація тестування, автоматизована сертифікація, управління користувачами, підтримка REST API, збереження результатів навчання та формування звітності.

Однією з ключових вимог є реалізація багаторольової моделі доступу, яка повинна забезпечувати розмежування прав доступу, контроль дій користувачів, захищену автентифікацію та ефективне управління ролями в системі.

Табл. 1.9. Основні ролі користувачів системи

| <b>Роль</b>   | <b>Основні функції</b>               |
|---------------|--------------------------------------|
| Адміністратор | Управління системою та користувачами |
| Викладач      | Створення курсів і тестів            |
| Слухач        | Проходження навчання                 |

Важливою функціональною вимогою інформаційної системи є підтримка управління курсами. Система повинна забезпечувати можливість створення курсів, редагування інформації, структурування навчальних матеріалів, реєстрації слухачів та контролю проходження навчальних програм.

Для організації дистанційного навчання система має підтримувати завантаження файлів, зберігання навчальних матеріалів, доступ до електронних ресурсів і онлайн-взаємодію між користувачами освітнього середовища.

Табл. 1.10. Функціональні вимоги до модуля курсів

| <b>Вимога</b>           | <b>Опис</b>                 |
|-------------------------|-----------------------------|
| Створення курсу         | Формування нового курсу     |
| Редагування             | Оновлення інформації        |
| Завантаження матеріалів | Додавання навчальних файлів |
| Реєстрація слухачів     | Запис на курс               |
| Контроль проходження    | Відстеження прогресу        |

Одним із важливих компонентів інформаційної системи є модуль тестування, який повинен забезпечувати створення тестів, формування запитань, автоматичне оцінювання результатів, збереження отриманих даних та обмеження кількості спроб проходження тестування.

Автоматизація процесу тестування сприяє підвищенню об'єктивності оцінювання знань, зменшенню навантаження на викладачів, прискоренню перевірки результатів навчання та автоматизації процедури сертифікації слухачів.

Табл. 1.11. Вимоги до модуля тестування

| Вимога                 | Опис                            |
|------------------------|---------------------------------|
| Створення тестів       | Формування тестових завдань     |
| Автоматичне оцінювання | Підрахунок результатів          |
| Обмеження спроб        | Контроль повторного проходження |
| Збереження результатів | Фіксація оцінок                 |
| Сертифікація           | Генерація сертифікатів          |

Система повинна підтримувати автоматизовану генерацію сертифікатів після успішного завершення навчального курсу. Сертифікат має містити прізвище, ім'я та по батькові слухача, назву курсу, дату завершення навчання, унікальний номер документа та інформацію про результати навчання.

Для забезпечення взаємодії між frontend- і backend-компонентами система повинна використовувати REST API, що дозволяє забезпечити масштабованість програмного рішення, підтримку SPA-архітектури, спрощення інтеграції компонентів та незалежний розвиток frontend- і backend-частин системи.

Окрім функціональних характеристик, важливе значення мають нефункціональні вимоги, які визначають якість функціонування інформаційної системи. До основних нефункціональних вимог належать продуктивність, масштабованість, безпечність, адаптивність, надійність та зручність використання системи.

Табл. 1.12. Нефункціональні вимоги до системи

| Вимога                 | Характеристика                           |
|------------------------|------------------------------------------|
| Продуктивність         | Швидка обробка запитів                   |
| Масштабованість        | Підтримка великої кількості користувачів |
| Безпечність            | Захист даних та доступу                  |
| Адаптивність           | Робота на різних пристроях               |
| Надійність             | Стабільне функціонування                 |
| Зручність використання | Інтуїтивний інтерфейс                    |

Система повинна забезпечувати високу продуктивність за умови одночасної роботи значної кількості користувачів. Для досягнення цього необхідно реалізувати оптимізацію запитів до бази даних, використання механізмів кешування та ефективну взаємодію між окремими компонентами системи.

Важливою вимогою є також масштабованість архітектури, яка має забезпечувати можливість додавання нових модулів, розширення функціональних можливостей, підтримку збільшення кількості користувачів та інтеграцію із зовнішніми сервісами.

Особливу увагу необхідно приділити інформаційній безпеці системи. Інформаційна система повинна підтримувати механізми автентифікації користувачів, захисту персональних даних, безпечної взаємодії через REST API, контролю доступу та шифрування даних.

Табл. 1.13. Основні вимоги до безпеки системи

| Вимога             | Призначення             |
|--------------------|-------------------------|
| JWT-автентифікація | Захищений доступ        |
| HTTPS              | Безпечна передача даних |
| Контроль ролей     | Розмежування доступу    |
| Захист API         | Безпечна взаємодія      |
| Захист БД          | Цілісність даних        |

Система повинна підтримувати адаптивний вебінтерфейс, який забезпечуватиме коректну роботу на персональних комп'ютерах, планшетах і мобільних пристроях. Інтерфейс користувача має бути інтуїтивно зрозумілим, зручним, сучасним, адаптивним та простим у використанні, що сприятиме ефективній взаємодії користувачів із функціональними можливостями системи.

Для забезпечення надійності функціонування інформаційна система повинна підтримувати механізми резервного копіювання даних, журналювання подій, обробку помилок і контроль стабільності роботи програмного забезпечення.

У результаті проведеного аналізу сформовано основні вимоги до веб-орієнтованої інформаційної системи NexLearn, які стали основою для подальшого проектування архітектури системи, структури бази даних та реалізації програмного забезпечення.

Таким чином, сформовані функціональні та нефункціональні вимоги визначають основні характеристики системи та забезпечують реалізацію ефективної веб-орієнтованої інформаційної системи супроводу навчального процесу в центрах підвищення кваліфікації.

## **РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ**

### **2.1. Загальна архітектура системи**

Після проведення аналізу предметної області та формування вимог до інформаційної системи наступним етапом дипломної роботи є проєктування архітектури веб-орієнтованої інформаційної системи супроводу навчального процесу NexLearn.

Архітектура інформаційної системи визначає структуру програмного забезпечення, принципи взаємодії між окремими компонентами, механізми обробки даних, способи інтеграції модулів, а також особливості масштабування та супроводу системи.

Під час проєктування інформаційної системи було обрано клієнт-серверну архітектуру, яка забезпечує розподіл логіки між frontend- і backend-компонентами, незалежний розвиток окремих частин системи, масштабованість, підтримку REST API та зручність супроводу програмного забезпечення.

Основними компонентами системи NexLearn є клієнтська частина (frontend), серверна частина (backend), база даних, REST API, модуль автентифікації та система зберігання файлів.

Табл. 2.1. Основні компоненти системи

| Компонент      | Призначення                      |
|----------------|----------------------------------|
| Frontend       | Користувацький інтерфейс         |
| Backend        | Бізнес-логіка системи            |
| REST API       | Взаємодія між компонентами       |
| PostgreSQL     | Зберігання даних                 |
| Authentication | Контроль доступу                 |
| File Storage   | Зберігання навчальних матеріалів |

Загальну архітектуру системи наведено на рисунку 2.1.

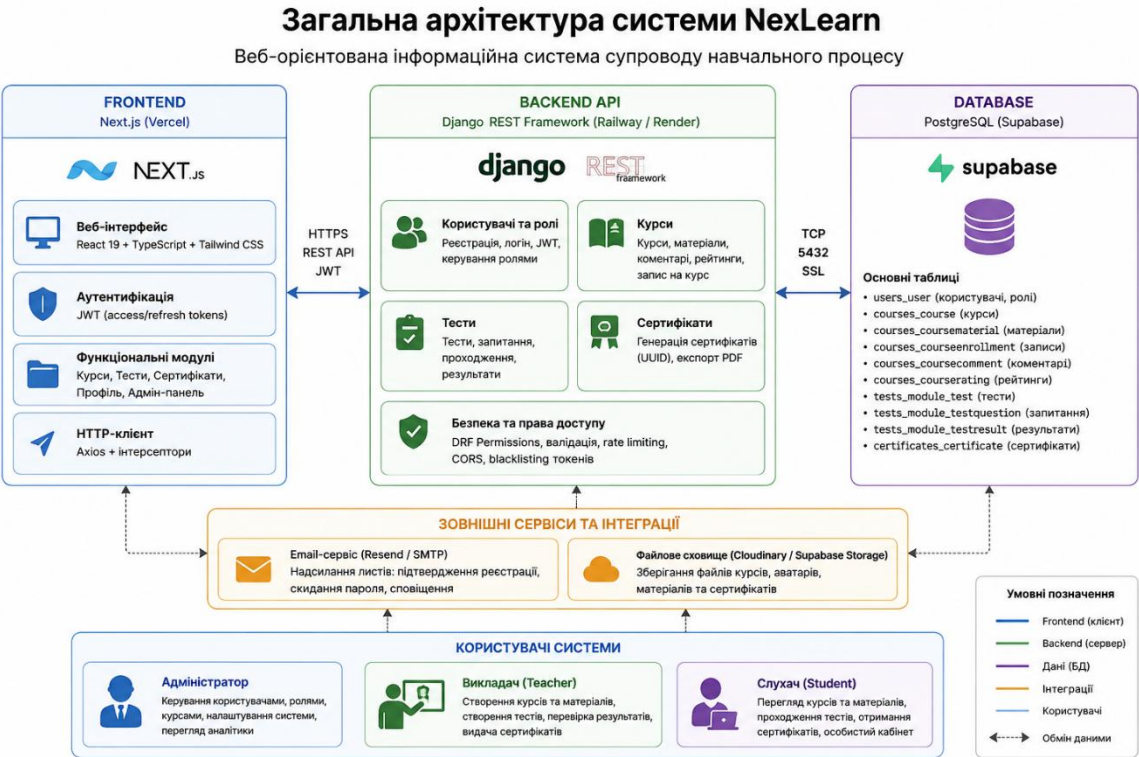


Рис. 2.1 Загальна архітектура системи NexLearn

Для реалізації клієнтської частини інформаційної системи використано Next.js та React. Такий підхід дозволяє реалізувати SPA-архітектуру, забезпечити швидку взаємодію з користувачем, створити адаптивний вебінтерфейс і підтримувати серверний рендеринг сторінок.

Frontend-частина системи відповідає за відображення інтерфейсу користувача, організацію взаємодії із системою, обробку подій, формування HTTP-запитів до API та відображення результатів роботи програмного забезпечення.

До основних компонентів frontend-частини належать dashboard, сторінки курсів, система навігації, модулі тестування, сторінки сертифікації та адміністративна панель системи.

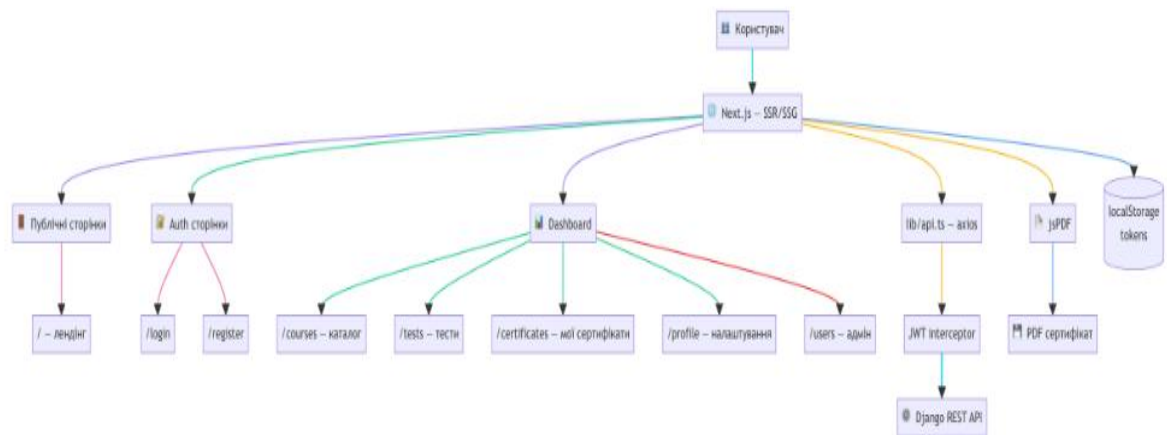


Рис. 2.2 Структура frontend частини системи

Серверна частина інформаційної системи реалізована із використанням Django та Django REST Framework. Backend-компонент забезпечує реалізацію бізнес-логіки системи, обробку HTTP-запитів, взаємодію з базою даних, автентифікацію користувачів, генерацію сертифікатів і обмін даними із frontend-частиною системи.

До основних функцій backend-частини належать управління користувачами, управління курсами, організація тестування, формування сертифікатів, підтримка REST API-взаємодії та контроль доступу до інформаційних ресурсів системи.

Табл. 2.2 – Основні функції backend частини

| Функція            | Призначення              |
|--------------------|--------------------------|
| User Management    | Управління користувачами |
| Course Management  | Управління курсами       |
| Testing Module     | Організація тестування   |
| Certificate Module | Генерація сертифікатів   |
| REST API           | Обробка API-запитів      |

Архітектуру backend частини наведено на рисунку 2.3.

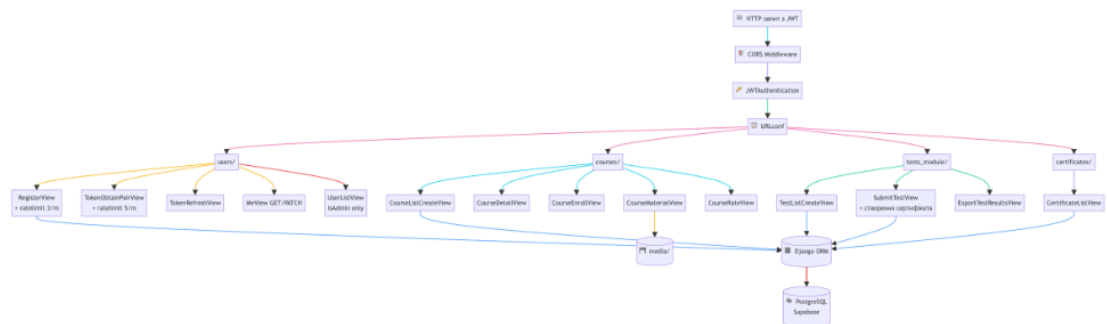


Рис. 2.3. Архітектура backend частини системи

Для взаємодії між frontend- і backend-компонентами використовується REST API. Такий підхід дозволяє розділити клієнтську та серверну логіку, забезпечити

масштабованість системи, реалізувати незалежну розробку окремих компонентів і підтримувати інтеграцію із зовнішніми сервісами.

Обмін даними між компонентами здійснюється у форматі JSON через HTTP-запити типів GET, POST, PUT та DELETE.

Табл. 2.3 Основні HTTP-методи REST API

| Метод  | Призначення          |
|--------|----------------------|
| GET    | Отримання даних      |
| POST   | Створення записів    |
| PUT    | Оновлення інформації |
| DELETE | Видалення записів    |

Схему REST API-взаємодії наведено на рисунку 2.4.



## Рис. 2.4 Взаємодія компонентів через REST API

Для зберігання даних у системі використовується PostgreSQL. Використання цієї системи керування базами даних забезпечує підтримку реляційної моделі, цілісність даних, транзакційність, можливість виконання складних SQL-запитів та масштабованість інформаційної системи.

База даних містить інформацію про користувачів, навчальні курси, навчальні матеріали, тести, результати тестування, сертифікати та ролі користувачів системи.

Табл. 2.4. Основні сутності бази даних

| Сутність    | Призначення           |
|-------------|-----------------------|
| User        | Дані користувачів     |
| Course      | Навчальні курси       |
| Material    | Навчальні матеріали   |
| Test        | Тести                 |
| Result      | Результати тестування |
| Certificate | Сертифікати           |

Для забезпечення безпечної взаємодії користувачів із системою реалізовано модуль автентифікації та авторизації.

Система підтримує:

- JWT-автентифікацію;
- role-based access control;
- refresh token;
- permission classes;
- контроль доступу до API.

Табл 2.5 Основні механізми безпеки системи

| Механізм           | Призначення                 |
|--------------------|-----------------------------|
| JWT                | Автентифікація користувачів |
| HTTPS              | Захищена передача даних     |
| RBAC               | Контроль ролей              |
| Permission Classes | Захист API                  |
| CORS               | Контроль доступу            |

Схему автентифікації наведено на рисунку 2.5.

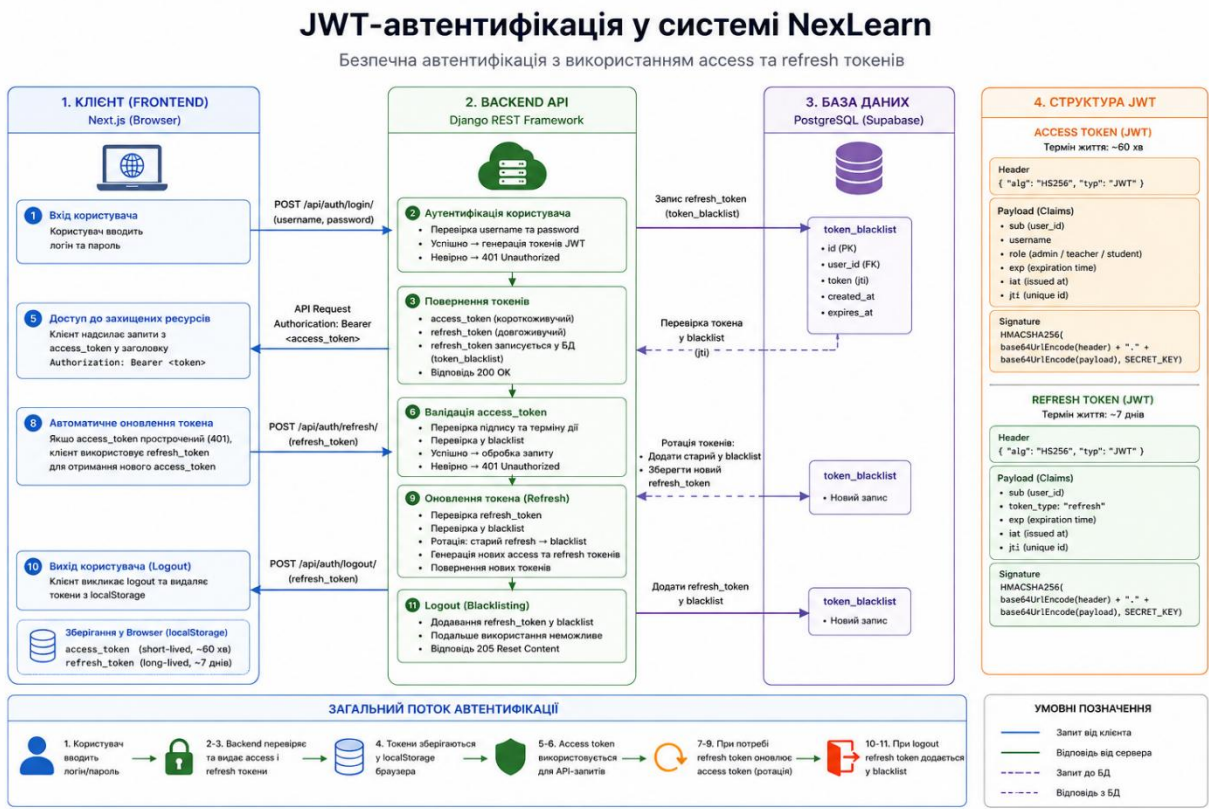


Рис. 2.5 JWT-автентифікація у системі

Однією з важливих вимог до інформаційної системи є підтримка адаптивного вебінтерфейсу, який повинен забезпечувати коректну роботу на персональних комп'ютерах, планшетах і мобільних пристроях. Для реалізації цієї вимоги використовується responsive design та компонентний підхід бібліотеки React.

Архітектура системи також передбачає масштабованість, модульність, можливість розширення функціоналу, інтеграцію із зовнішніми сервісами та підтримку cloud deployment, що забезпечує гнучкість і перспективність подальшого розвитку інформаційної системи.

Для розгортання системи можуть використовуватися:

- Vercel;
- Railway;
- Render;
- Supabase.

Табл. 2.6. Переваги обраної архітектури

| Перевага        | Характеристика               |
|-----------------|------------------------------|
| Масштабованість | Підтримка розвитку системи   |
| Модульність     | Незалежність компонентів     |
| Безпечність     | Захист даних та API          |
| Адаптивність    | Робота на різних пристроях   |
| REST API        | Гнучка взаємодія компонентів |

У результаті проєктування архітектури системи сформовано сучасну клієнт-серверну структуру веборієнтованої інформаційної системи NexLearn, яка забезпечує автоматизацію навчального процесу, підтримку дистанційного навчання, масштабованість, безпечність, адаптивність та ефективну взаємодію між компонентами системи.

## 2.2. Проєктування бази даних

Одним із ключових етапів розробки веб-орієнтованої інформаційної системи є проєктування бази даних. Від структури бази даних залежить ефективність зберігання інформації, швидкість доступу до даних, цілісність інформаційних ресурсів та можливість масштабування системи.

У процесі проєктування інформаційної системи NexLearn було використано реляційний підхід до організації даних із застосуванням PostgreSQL. Вибір реляційної моделі обумовлений необхідністю забезпечення цілісності даних, підтримки складних зв'язків між сутностями, транзакційності, масштабованості та ефективного виконання SQL-запитів.

База даних системи забезпечує зберігання інформації про користувачів, навчальні курси, навчальні матеріали, тести, результати тестування, сертифікати, рейтинги та коментарі користувачів.

Табл. 2.7 Основні сутності бази даних

| Сутність         | Призначення                    |
|------------------|--------------------------------|
| User             | Дані користувачів системи      |
| Course           | Інформація про навчальні курси |
| CourseMaterial   | Матеріали курсів               |
| CourseEnrollment | Реєстрація слухачів на курси   |
| CourseComment    | Коментарі до курсів            |
| CourseRating     | Оцінювання курсів              |
| Test             | Тести курсів                   |
| TestQuestion     | Тестові запитання              |
| TestResult       | Результати тестування          |
| Certificate      | Сертифікати слухачів           |

У процесі аналізу предметної області було визначено основні інформаційні потоки системи, до яких належать реєстрація користувачів, створення курсів, запис слухачів на курси, проходження тестування, формування сертифікатів та взаємодія з навчальними матеріалами.

Структуру інформаційних потоків системи наведено на рисунку 2.6.

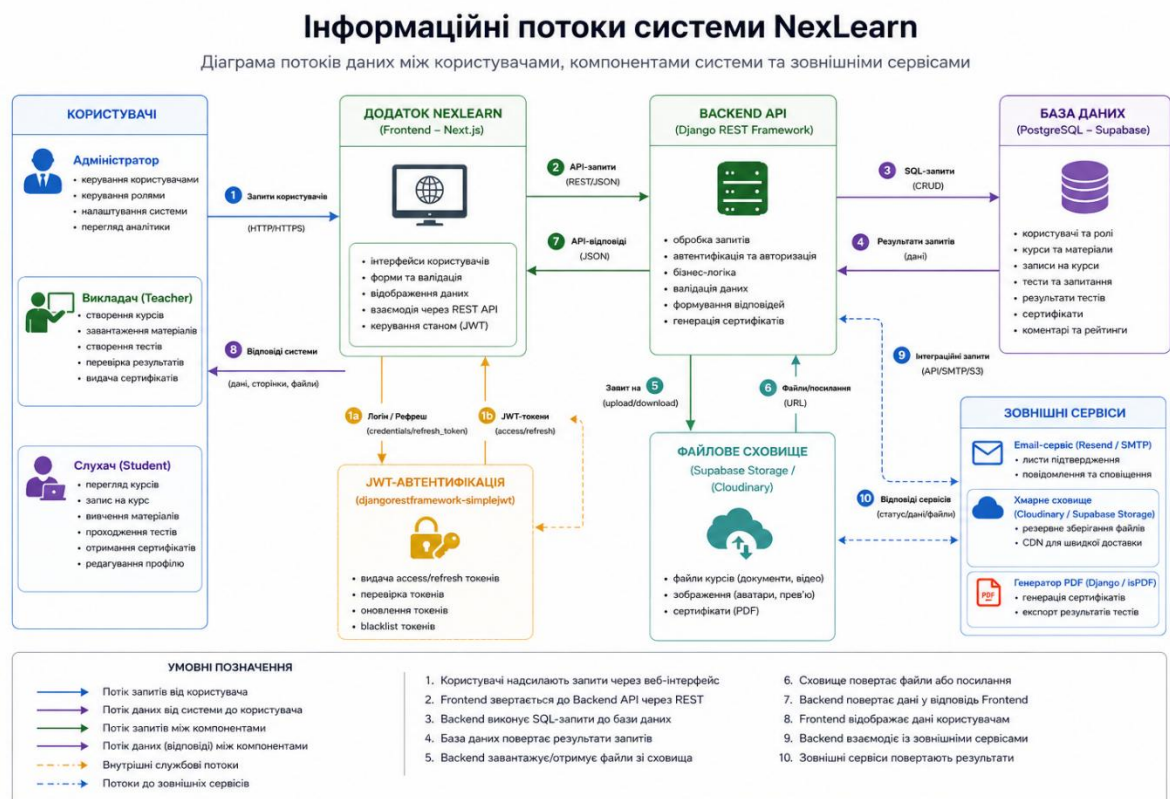


Рис. 2.6 Інформаційні потоки системи NexLearn

Основною сутністю інформаційної системи є користувач (User). У моделі користувача зберігаються логін, пароль, електронна пошта, роль, контактні дані та інформація профілю.

У системі реалізовано кастомну модель користувача Django із підтримкою ролей admin, teacher та student.

Залежно від ролі користувач може створювати курси, проходити навчання, складати тести та отримувати сертифікати.

Структуру сутності User наведено на рисунку 2.7.

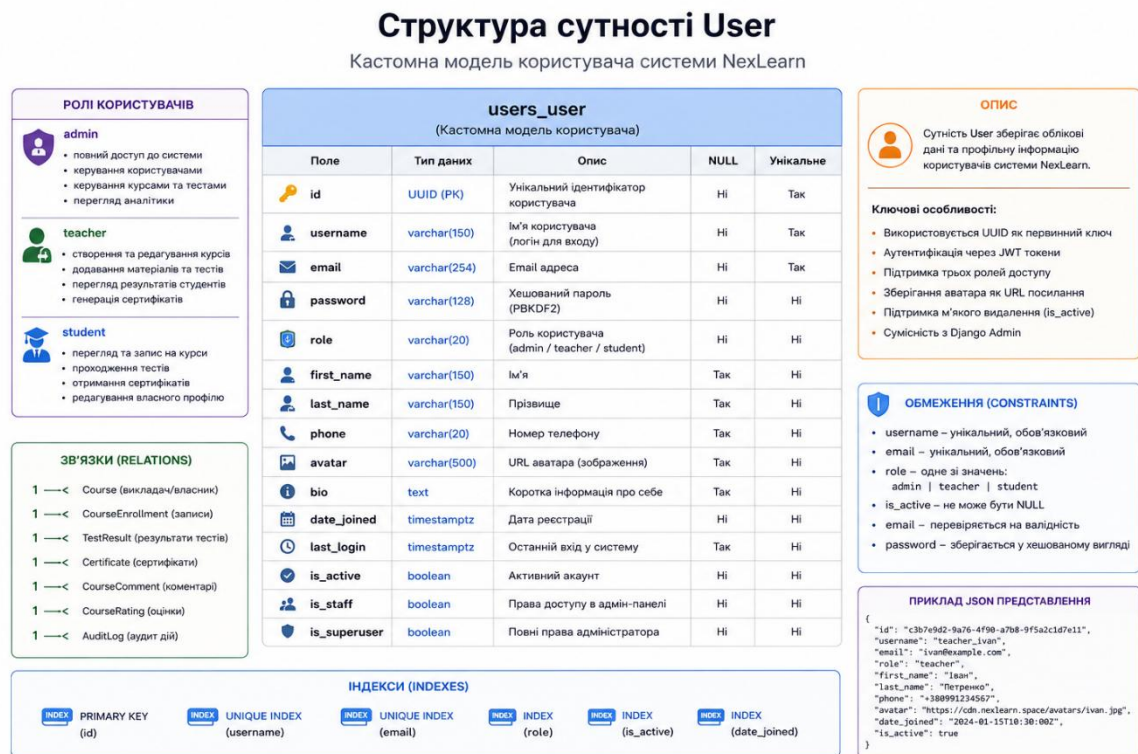


Рис 2.7 Структура сутності User

Сутність Course використовується для зберігання інформації про навчальні курси. Для кожного курсу зберігаються:

- назва;
- опис;
- статус;
- автор курсу;
- дата створення;
- рейтинг;
- навчальні матеріали.

Курс пов'язаний із викладачем через зв'язок one-to-many. Один викладач може створювати декілька курсів.

Слухачі взаємодіють із курсами через проміжну сутність CourseEnrollment, яка реалізує зв'язок many-to-many між користувачами та курсами.

Табл. 2.8. Основні зв'язки між сутностями

| Сутність 1 | Тип зв'язку  | Сутність 2     |
|------------|--------------|----------------|
| User       | one-to-many  | Course         |
| User       | many-to-many | Course         |
| Course     | one-to-many  | CourseMaterial |
| Course     | one-to-many  | Test           |
| Test       | one-to-many  | TestQuestion   |
| Test       | one-to-many  | TestResult     |
| Course     | one-to-many  | Certificate    |

Концептуальну модель бази даних наведено на рисунку 2.8.

Повна схема моделей з полями, типами і зв'язками між таблицями PostgreSQL (Supabase).

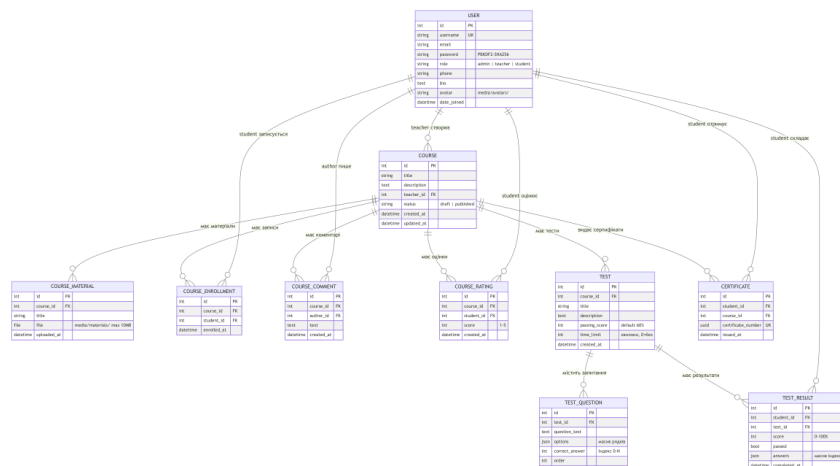


Рис. 2.8 ER-модель бази даних системи

Навчальні матеріали зберігаються у сутності CourseMaterial, для якої визначаються назва матеріалу, файл, відповідний курс та дата завантаження. Система підтримує завантаження PDF-файлів, DOCX-документів, ZIP-архівів і мультимедійних матеріалів.

Для організації тестування використовується сутність Test, яка містить назву тесту, опис, прохідний бал, часові обмеження та зв'язок із відповідним курсом. Кожен тест включає набір тестових запитань, що зберігаються у сутності TestQuestion. Для кожного запитання визначаються текст запитання, варіанти відповідей, правильна відповідь та порядок відображення у системі.

Структуру модуля тестування наведено на рисунку 2.9.

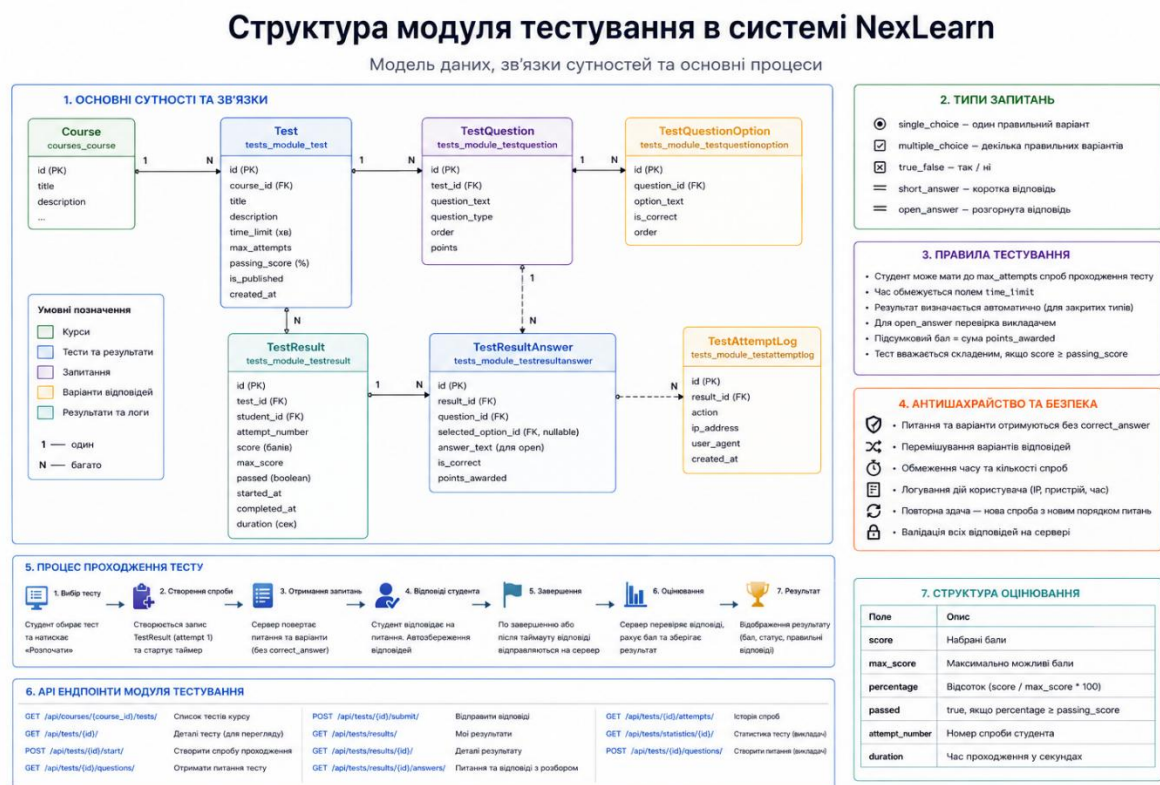


Рис. 2.9 Структура модуля тестування

Результати проходження тестів зберігаються у сутності TestResult. Для кожного результату фіксуються дані про слухача, тест, набраний бал, статус проходження, кількість спроб та дату проходження тестування.

Для автоматизації процесу сертифікації використовується сутність Certificate, яка містить інформацію про слухача, курс, унікальний UUID-номер сертифіката та дату його видачі.

Сертифікат створюється автоматично після успішного проходження тесту.

Структуру сутності Certificate наведено на рисунку 2.10.

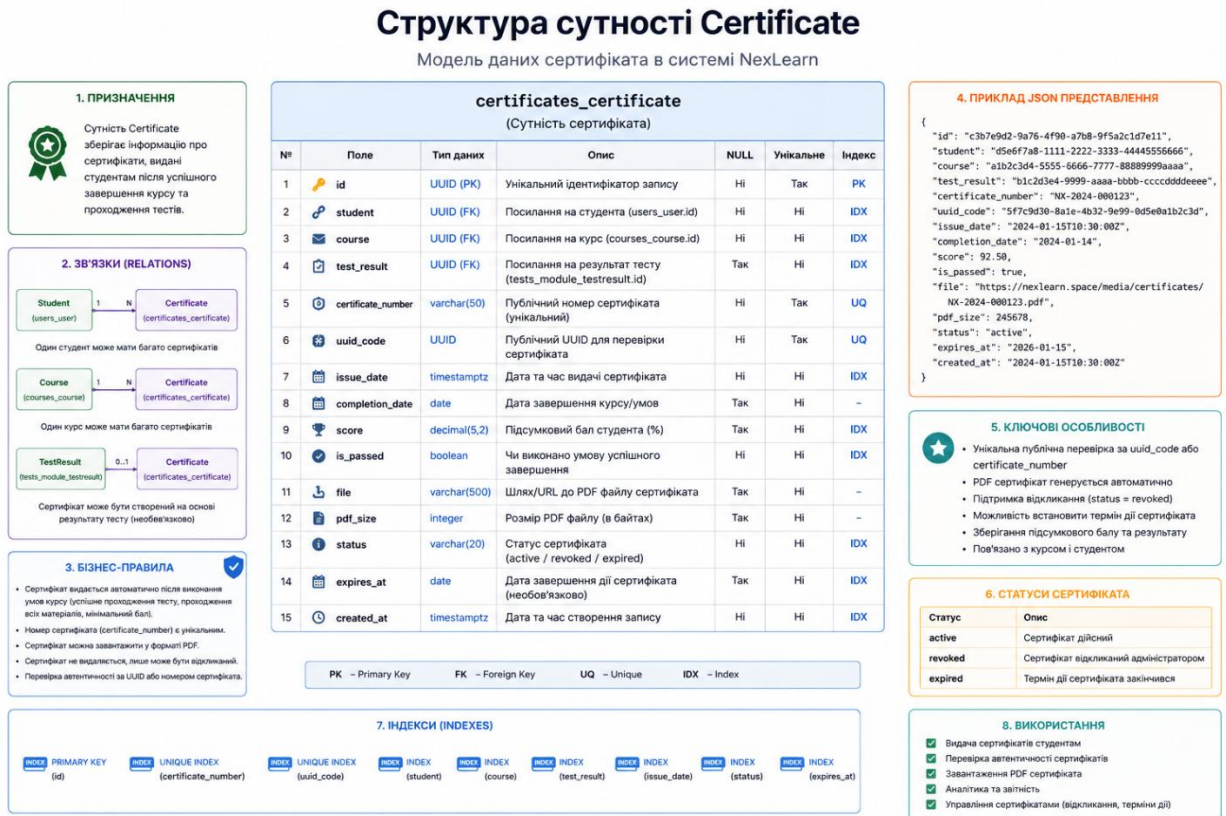


Рис. 2.10 Структура сутності Certificate

Для забезпечення цілісності даних у базі даних використовуються механізми primary key, foreign key, unique constraints, каскадні зв'язки та засоби ORM фреймворку Django.

У процесі проєктування було виконано нормалізацію бази даних, що дозволило усунути дублювання інформації, забезпечити логічну цілісність даних і підвищити ефективність обробки SQL-запитів.

Табл. 2.9 Основні механізми забезпечення цілісності даних

| Механізм          | Призначення                     |
|-------------------|---------------------------------|
| Primary Key       | Унікальна ідентифікація записів |
| Foreign Key       | Зв'язок між таблицями           |
| Unique Constraint | Уникнення дублювання            |
| ORM Django        | Контроль цілісності даних       |
| Validation        | Перевірка коректності значень   |

Для оптимізації продуктивності інформаційної системи використовуються механізми QuerySet annotations, select\_related, prefetch\_related, індексація таблиць та кешування ORM-запитів. Застосування цих підходів дозволяє підвищити ефективність обробки даних і зменшити навантаження на базу даних.

Важливою перевагою використання PostgreSQL є підтримка транзакційності, конкурентного доступу до даних, виконання складних SQL-запитів, масштабування системи та реалізації механізмів резервного копіювання. Таблиця 2.10 – Переваги використання PostgreSQL

| Перевага        | Характеристика                   |
|-----------------|----------------------------------|
| Транзакційність | Підтримка ACID                   |
| Масштабованість | Робота з великими обсягами даних |
| Безпечність     | Захист та контроль доступу       |
| Продуктивність  | Оптимізація SQL-запитів          |
| Надійність      | Механізми резервного копіювання  |

Таким чином, проєктування бази даних системи NexLearn виконано з урахуванням особливостей навчального процесу в центрах підвищення кваліфікації та сучасних вимог до веб-орієнтованих інформаційних систем.

Розроблена структура бази даних забезпечує ефективне зберігання інформації, підтримку складних взаємозв'язків між сутностями, масштабованість інформаційної системи, цілісність даних та реалізацію бізнес-логіки навчального процесу.

### **2.3. Моделювання бізнес-процесів та користувацьких ролей**

Одним із важливих етапів проєктування веб-орієнтованої інформаційної системи є моделювання бізнес-процесів та користувацьких ролей. Побудова моделей дозволяє формалізувати логіку роботи системи, визначити механізми взаємодії користувачів та забезпечити коректну реалізацію функціональних можливостей програмного забезпечення.

У системі NexLearn бізнес-процеси охоплюють реєстрацію користувачів, авторизацію, створення курсів, проходження навчання, тестування, сертифікацію та адміністрування системи. Основою функціонування платформи є багаторольова модель доступу, яка визначає права користувачів відповідно до їхніх функцій у навчальному процесі.

У платформі NexLearn реалізовано три основні ролі користувачів: адміністратор, викладач і слухач.

Табл. 2.11 Основні ролі користувачів системи

| <b>Роль</b>   | <b>Основні функції</b>                           |
|---------------|--------------------------------------------------|
| Адміністратор | Управління користувачами, ролями та системою     |
| Викладач      | Створення курсів, тестів і навчальних матеріалів |
| Слухач        | Проходження навчання та тестування               |

Адміністратор виконує функції керування інформаційною системою та має доступ до управління користувачами, зміни ролей, модерації контенту, контролю функціонування системи й перегляду статистичних даних.

Викладач відповідає за створення курсів, завантаження навчальних матеріалів, формування тестів, перевірку результатів навчання та видачу сертифікатів слухачам.

Слухач здійснює реєстрацію в системі, перегляд навчальних курсів, проходження тестування, отримання сертифікатів і взаємодію з навчальними матеріалами.

Рольову модель доступу в системі наведено на рисунку 2.11.



Рис. 2.11. Рольова модель доступу системи NexLearn

Контроль доступу реалізовано за допомогою role-based access control та permission classes Django REST Framework. Для кожного API-ендпоінта визначаються дозволені ролі користувачів.

У системі реалізовано декілька основних бізнес-процесів, які забезпечують повний цикл супроводу навчального процесу.

Одним із базових процесів інформаційної системи є реєстрація та авторизація користувачів. Після створення облікового запису система перевіряє коректність введених даних, виконує хешування пароля, генерує JWT-токени та надає користувачу доступ до функціональних можливостей системи.

Схему процесу авторизації наведено на рисунку 2.12.



Рис. 2.12. Схема процесу авторизації користувача

Після авторизації користувач отримує доступ до dashboard, функціонал якого залежить від ролі користувача.

Одним із ключових бізнес-процесів системи є створення та управління навчальними курсами. Викладач має можливість створювати курси, редагувати інформацію про них, завантажувати навчальні матеріали, формувати тести та керувати слухачами, які зареєстровані на відповідний курс.

Система забезпечує CRUD-операції для управління курсами через REST API.

Табл. 2.12 Основні бізнес-процеси системи

| Бізнес-процес           | Опис                          |
|-------------------------|-------------------------------|
| Реєстрація              | Створення облікового запису   |
| Авторизація             | JWT-автентифікація            |
| Управління курсами      | CRUD-операції для курсів      |
| Завантаження матеріалів | Додавання навчальних ресурсів |
| Тестування              | Проходження тестів            |
| Сертифікація            | Генерація сертифікатів        |

Процес управління курсами наведено на рисунку 2.13.

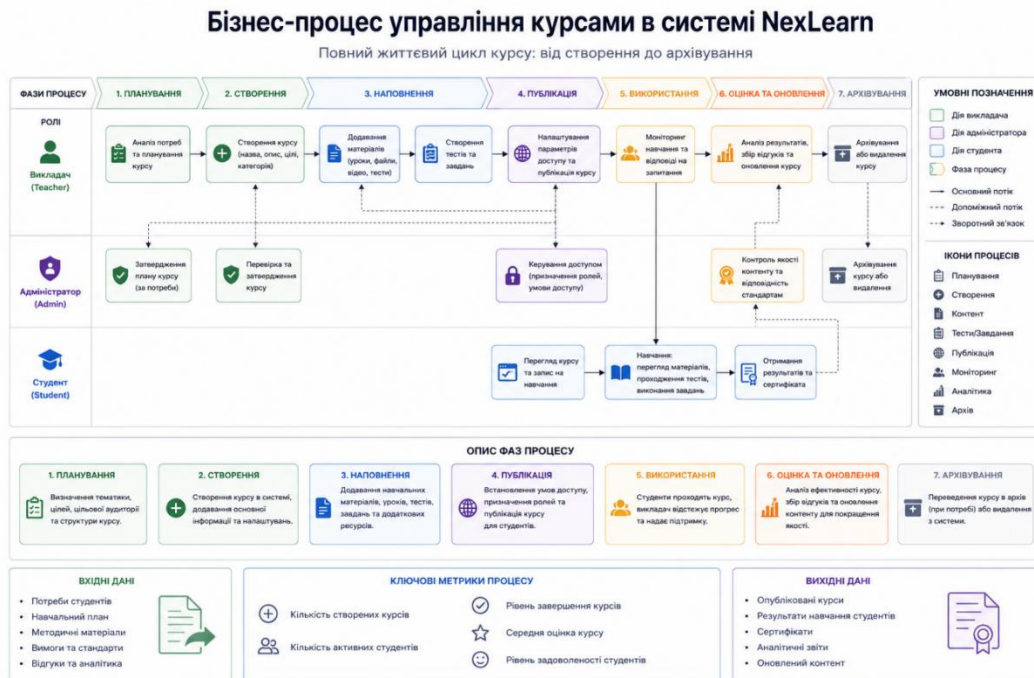


Рис. 2.13 Бізнес-процес управління курсами

Важливим елементом інформаційної системи є модуль тестування, бізнес-процес якого охоплює створення тесту, формування тестових запитань, проходження тестування слухачем, автоматичний підрахунок результатів, збереження отриманих даних та формування сертифіката після успішного завершення курсу.

У системі реалізовано механізм автоматичної перевірки результатів та обмеження кількості спроб проходження тесту.

Схему процесу тестування наведено на рисунку 2.14.

## Бізнес-процес тестування в системі NexLearn

Повний цикл: від створення тесту до отримання результатів

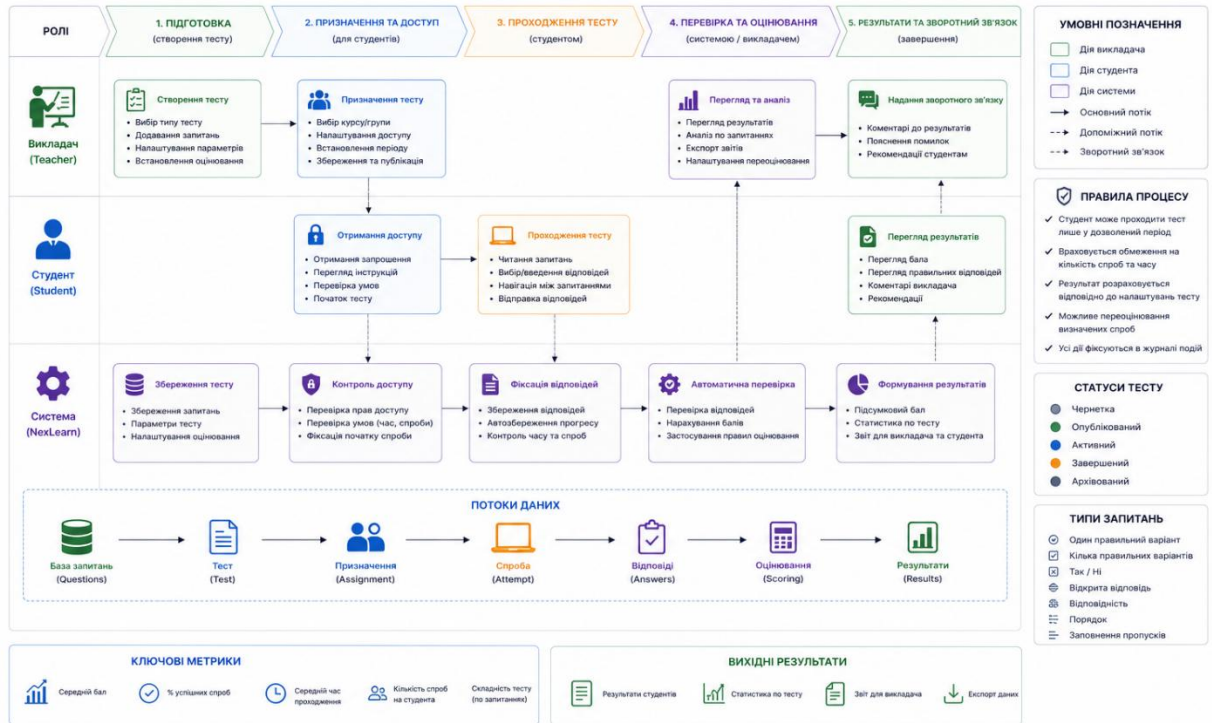


Рис. 2.14 Бізнес-процес тестування

Після успішного проходження тестування система автоматично генерує сертифікат слухача. У процесі сертифікації виконується перевірка результатів тестування, створення UUID сертифіката, формування PDF-документа та збереження сертифіката у базі даних.

Процес сертифікації наведено на рисунку 2.15.

## Бізнес-процес сертифікації в системі NexLearn

Повний цикл: від виконання вимог курсу до видачі сертифіката

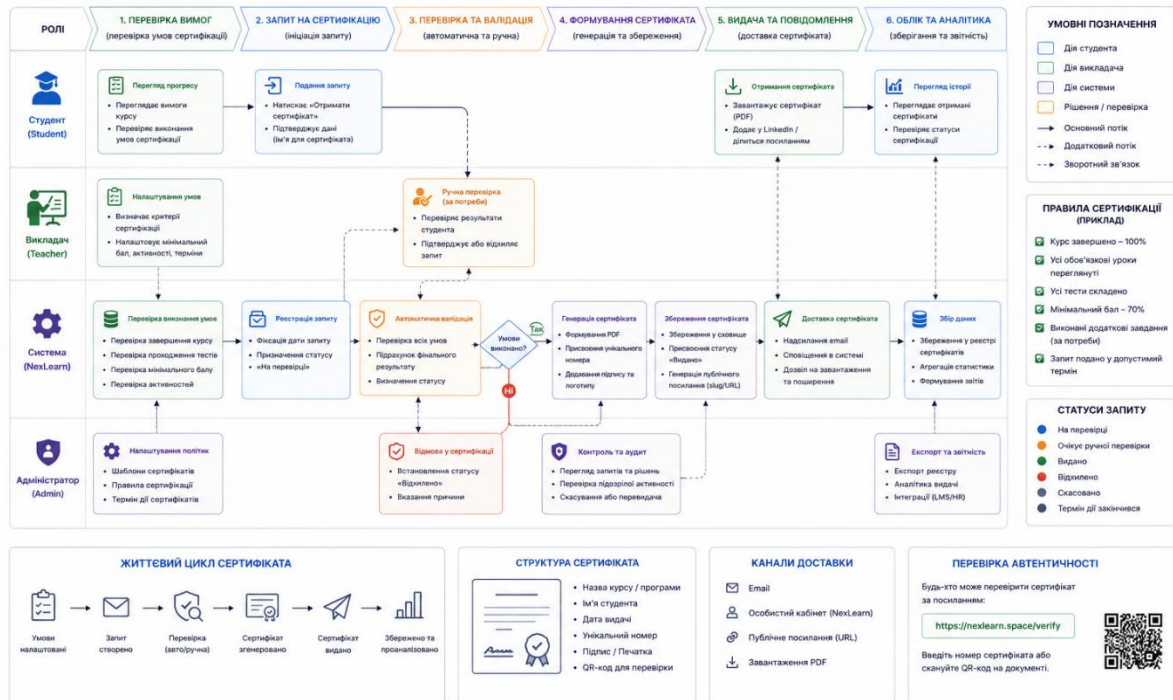


Рис. 2.15 Бізнес-процес сертифікації

Для формалізації взаємодії користувачів із системою використовуються UML-діаграми. Однією з основних UML-моделей є use-case діаграма, яка відображає взаємодію користувачів із функціоналом системи.

Основними сценаріями використання інформаційної системи є авторизація користувачів, перегляд навчальних курсів, створення курсів, проходження тестування, отримання сертифікатів та управління користувачами системи.

Use-case модель системи наведено на рисунку 2.16.

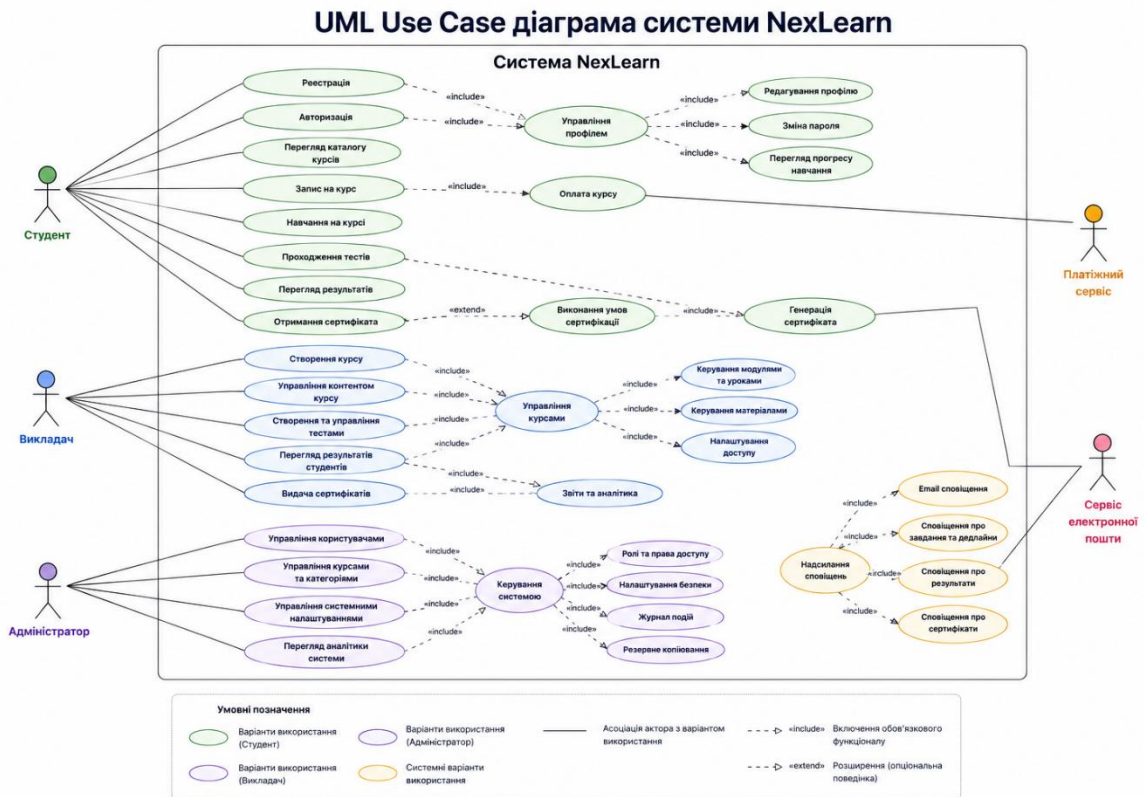


Рис. 2.16 UML use-case діаграма системи

Для моделювання послідовності взаємодії компонентів використовуються sequence diagrams, які демонструють порядок обробки запитів між frontend, backend та базою даних.

Прикладом є сценарій проходження тестування:

- Користувач відкриває тест.
- Frontend надсилає API-запит.
- Backend отримує тест із бази даних.
- Користувач надсилає відповіді.
- Backend обчислює результат.
- Результат зберігається у PostgreSQL.
- Система формує сертифікат.

Sequence-діаграму процесу тестування наведено на рисунку 2.17.

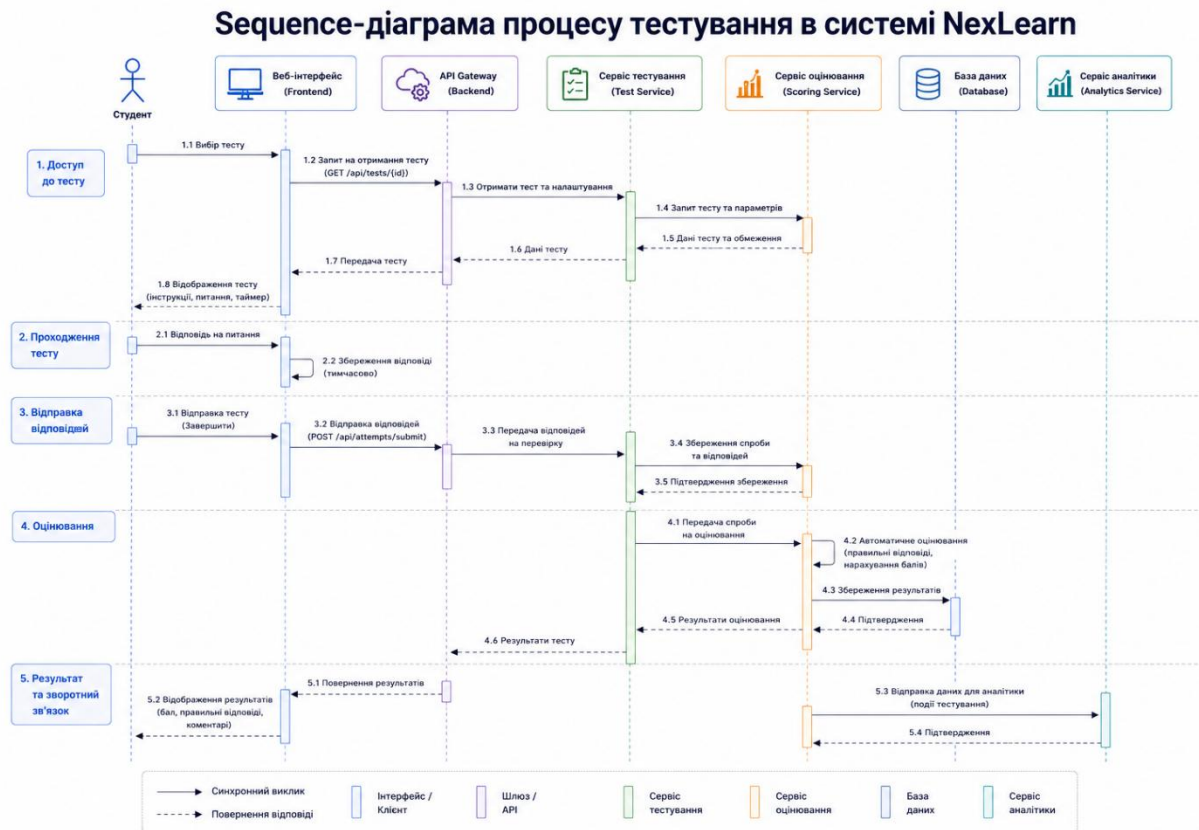


Рис. 2.17 Sequence-діаграма процесу тестування

Важливим аспектом моделювання є побудова activity diagrams, які описують послідовність виконання дій у системі.

Activity-діаграма проходження навчання наведена на рисунку 2.18.

Activity-діаграма процесу навчання в системі NexLearn

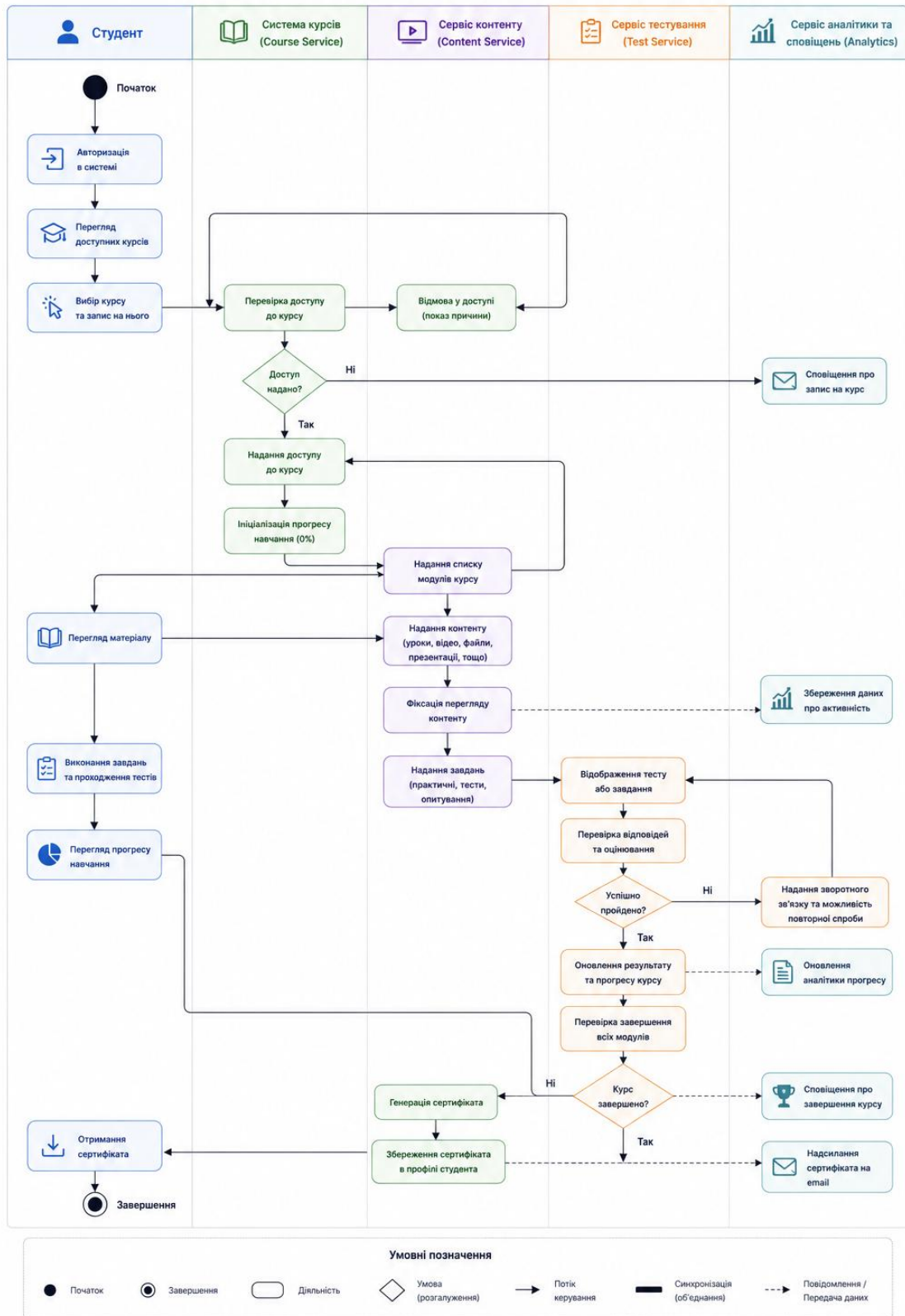


Рис. 2.18 Activity-діаграма процесу навчання

У результаті моделювання бізнес-процесів було визначено логіку взаємодії користувачів, структуру доступу до інформаційних ресурсів, основні сценарії роботи системи, порядок обробки даних та механізми автоматизації навчального процесу.

Використання UML-моделювання дозволило формалізувати архітектуру інформаційної системи, спростити проєктування її компонентів, забезпечити структурованість програмного забезпечення та підвищити якість реалізації системи.

Таким чином, моделювання бізнес-процесів та користувацьких ролей дозволило сформулювати логічну основу функціонування веб-орієнтованої інформаційної системи NexLearn та забезпечити коректну реалізацію процесів супроводу навчального процесу в центрах підвищення кваліфікації.

## **РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ**

### **3.1. Вибір технологій та інструментів розробки**

Одним із важливих етапів реалізації веб-орієнтованої інформаційної системи є вибір технологій та інструментів розробки. Від правильності вибору програмних засобів залежить продуктивність системи, масштабованість, безпечність, зручність супроводу та можливість подальшого розвитку програмного забезпечення.

Під час розробки інформаційної системи NexLearn було обрано сучасний стек вебтехнологій, який забезпечує модульність архітектури, підтримку REST API, ефективну взаємодію між frontend- і backend-компонентами, адаптивність інтерфейсу, масштабованість і безпечність функціонування системи.

Для реалізації клієнтської частини системи використано React та Next.js. Використання React обумовлено широкими можливостями побудови компонентного інтерфейсу, підтримкою SPA-архітектури та високою продуктивністю рендерингу.

Основними перевагами React є компонентний підхід до побудови інтерфейсу, можливість повторного використання компонентів, ефективне оновлення DOM, підтримка hooks, висока швидкодія інтерфейсу та наявність великої екосистеми бібліотек.

Next.js було обрано як framework для React завдяки підтримці SSR і SSG, вбудованій маршрутизації, оптимізації продуктивності, підтримці API routes та інтеграції із сучасними deployment-платформами.

Табл. 3.1 Технології frontend частини системи

| Технологія   | Призначення                       |
|--------------|-----------------------------------|
| React        | Побудова компонентного інтерфейсу |
| Next.js      | SPA-архітектура та SSR            |
| TypeScript   | Типізація JavaScript-коду         |
| Tailwind CSS | Стилізація інтерфейсу             |
| Axios        | HTTP-взаємодія з API              |

Архітектуру frontend частини системи наведено на рисунку 3.1.

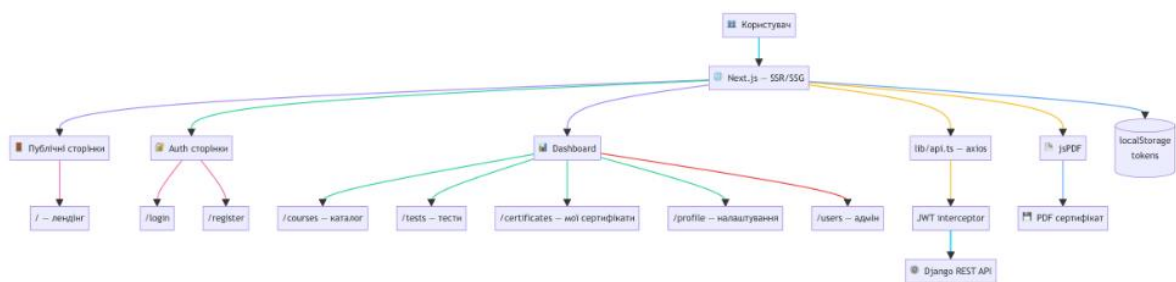


Рис. 3.1 Архітектура frontend частини системи

Для забезпечення типізації програмного коду використовується TypeScript, який дозволяє зменшити кількість помилок, забезпечити контроль типів даних, покращити підтримуваність коду та підвищити якість розробки програмного забезпечення.

Для стилізації інтерфейсу застосовується Tailwind CSS, який забезпечує адаптивність інтерфейсу, швидке створення UI-компонентів, мінімізацію CSS-коду та підтримку responsive design.

Для взаємодії frontend-частини із backend-компонентом використовується бібліотека Axios, що забезпечує виконання HTTP-запитів, підтримку interceptor-механізмів, автоматичне оновлення JWT-токенів та централізовану обробку помилок.

Для реалізації серверної частини системи використано Django та Django REST Framework. Вибір Django обумовлений високим рівнем безпеки, підтримкою ORM, наявністю вбудованої системи автентифікації, підтримкою REST API, швидкістю розробки та масштабованістю.

Django REST Framework використовується для створення REST API, серіалізації даних, реалізації permission classes, побудови API-ендпоінтів та обробки HTTP-запитів.

Табл. 3.2 Технології backend частини системи

| Технологія            | Призначення                 |
|-----------------------|-----------------------------|
| Django                | Реалізація backend логіки   |
| Django REST Framework | Створення REST API          |
| SimpleJWT             | JWT-автентифікація          |
| django-ratelimit      | Обмеження кількості запитів |
| Gunicorn              | WSGI-сервер                 |

Структуру backend частини системи наведено на рисунку 3.2.

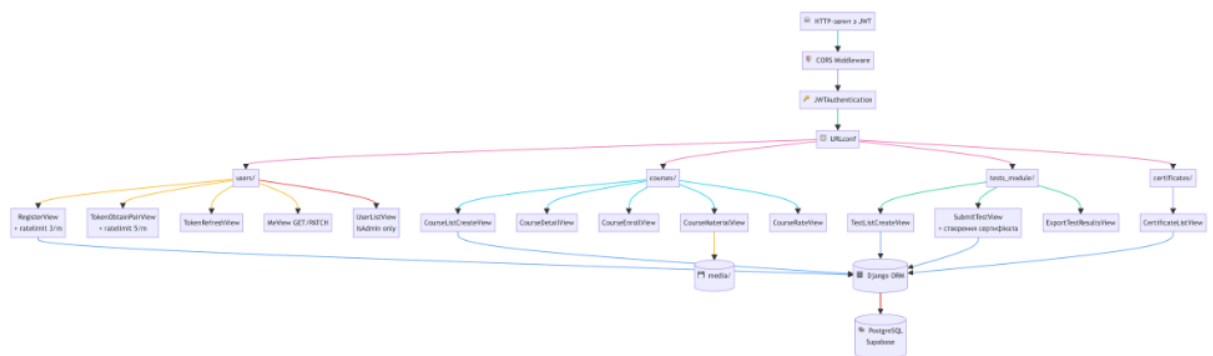


Рис. 3.2 Архітектура backend частини системи

Схему JWT-взаємодії наведено на рисунку 3.3.



Рис. 3.3 JWT-автентифікація в системі

Для зберігання даних у системі використовується PostgreSQL, яка забезпечує підтримку реляційної моделі даних, транзакційність, масштабованість, можливість виконання складних SQL-запитів та високу продуктивність обробки інформації.

Табл. 3.3 Переваги використання PostgreSQL

| Перевага            | Характеристика                   |
|---------------------|----------------------------------|
| ACID-транзакції     | Забезпечення цілісності даних    |
| Масштабованість     | Робота з великими обсягами даних |
| Продуктивність      | Оптимізація SQL-запитів          |
| Безпечність         | Контроль доступу та шифрування   |
| Інтеграція з Django | ORM-підтримка                    |

Для розгортання інформаційної системи використовуються сучасні cloud-сервіси, зокрема Vercel, Railway, Supabase та GitHub.

Frontend-частина системи розгортається на платформі Vercel, що забезпечує автоматичний deployment, CDN-доставку контенту, підтримку Next.js та високу швидкодію вебзастосунку.

Backend-компонент розгортається на платформах Railway або Render із підтримкою Gunicorn, environment variables, PostgreSQL та механізмів CI/CD.

Табл. 3.4 Засоби розгортання системи

| Інструмент       | Призначення              |
|------------------|--------------------------|
| Vercel           | Хостинг frontend         |
| Railway / Render | Хостинг backend          |
| Supabase         | PostgreSQL та storage    |
| GitHub           | Контроль версій та CI/CD |

Інфраструктуру розгортання системи наведено на рисунку 3.4.



Рис. 3.4 Інфраструктура розгортання системи

Для забезпечення якості програмного забезпечення використовуються:

- ESLint;
- Prettier;
- Django Debug Toolbar;
- logging;
- GitHub Actions.

Табл. 3.5 Додаткові інструменти розробки

| Інструмент           | Призначення            |
|----------------------|------------------------|
| ESLint               | Аналіз JavaScript-коду |
| Prettier             | Форматування коду      |
| GitHub Actions       | Автоматизація CI/CD    |
| Django Debug Toolbar | Аналіз SQL-запитів     |
| Logging              | Моніторинг помилок     |

Таким чином, обраний стек технологій забезпечує реалізацію сучасної SPA-архітектури, масштабованість інформаційної системи, підтримку REST API, високий рівень безпеки, ефективну взаємодію між компонентами, а також адаптивність і високу продуктивність веборієнтованої інформаційної системи NexLearn.

### 3.2. Реалізація основних модулів

Після завершення етапу проєктування архітектури системи та вибору технологій було виконано практичну реалізацію веб-орієнтованої інформаційної системи NexLearn. У процесі розробки реалізовано основні функціональні модулі, які забезпечують автоматизацію супроводу навчального процесу в центрах підвищення кваліфікації.

Програмне забезпечення реалізовано на основі клієнт-серверної архітектури із використанням:

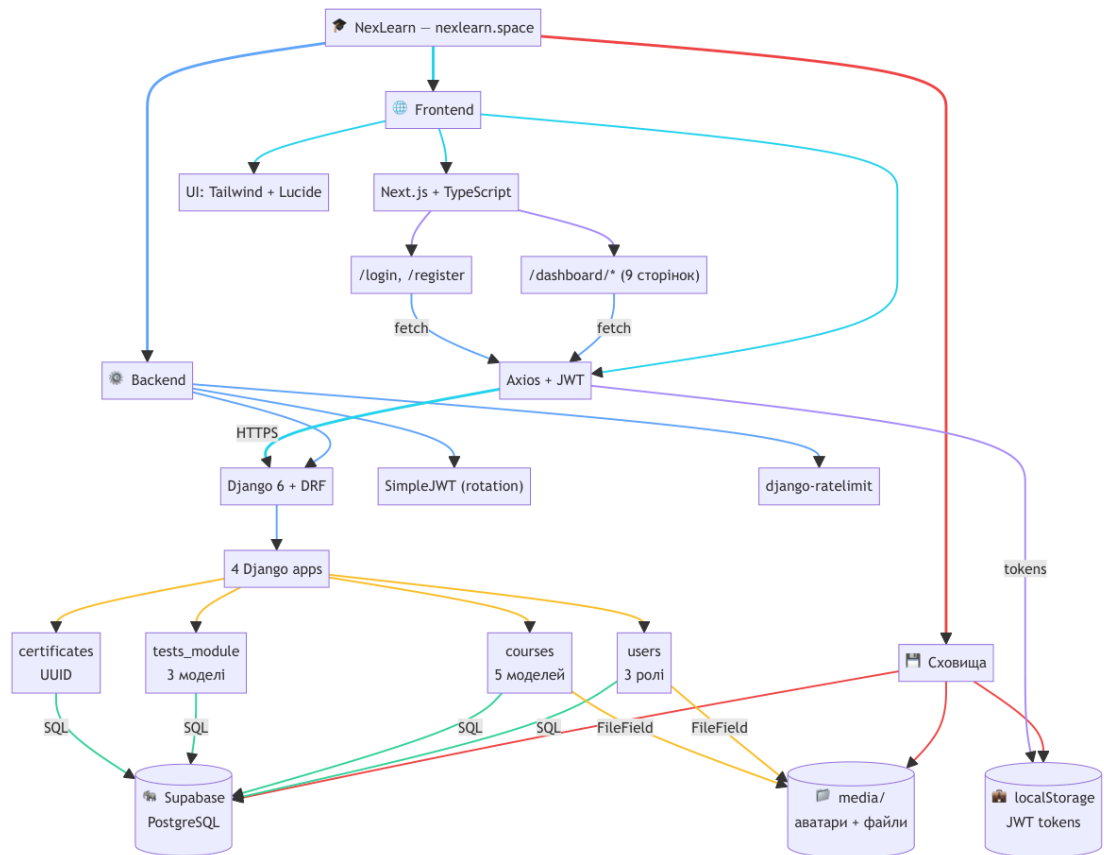
- Next.js та React для frontend частини;
- Django і Django REST Framework для backend;
- PostgreSQL для зберігання даних.
- У системі реалізовано такі основні модулі:
- модуль автентифікації та авторизації;

- модуль управління користувачами;
- модуль управління курсами;
- модуль навчальних матеріалів;
- модуль тестування;
- модуль сертифікації;
- REST API модуль.

Табл. 3.6. Основні модулі системи NexLearn

| <b>Модуль</b>  | <b>Призначення</b>               |
|----------------|----------------------------------|
| Authentication | Авторизація та контроль доступу  |
| Users          | Управління користувачами         |
| Courses        | Управління курсами               |
| Materials      | Робота з навчальними матеріалами |
| Tests          | Організація тестування           |
| Certificates   | Генерація сертифікатів           |
| REST API       | Взаємодія frontend та backend    |

Загальну структуру програмних модулів наведено на рисунку 3.5.



Рису. 3.5 Архітектура програмних модулів системи

Одним із ключових компонентів системи є модуль автентифікації та авторизації. У системі реалізовано JWT-автентифікацію із використанням access та refresh token.

Модуль автентифікації забезпечує реєстрацію користувачів, вхід до системи, оновлення JWT-токенів, завершення користувацької сесії та контроль доступу до API.

Для реалізації механізмів автентифікації використано вбудовану систему Django Authentication, бібліотеку SimpleJWT, permission classes та middleware-механізми, що забезпечують безпечну взаємодію користувачів із системою.

Схему роботи модуля автентифікації наведено на рисунку 3.6.

## Робота модуля автентифікації в системі NexLearn

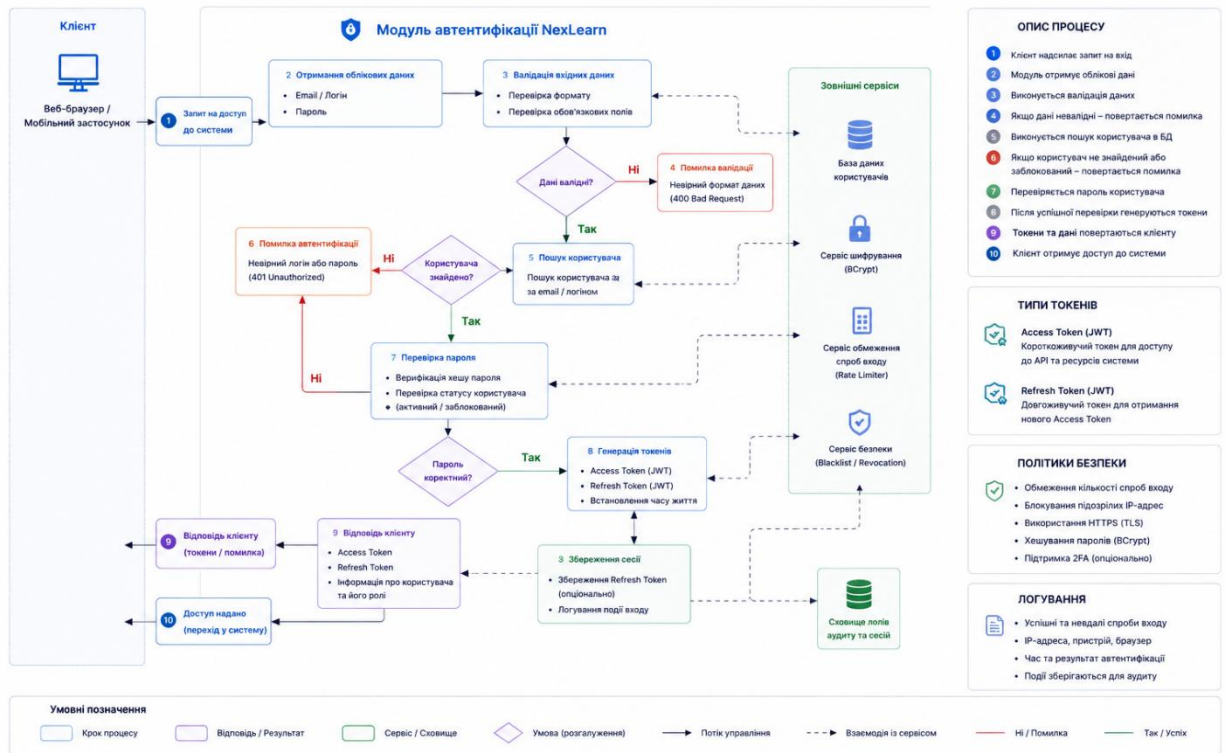


Рис. 3.6 Робота модуля автентифікації

У системі реалізовано багаторольову модель доступу, відповідно до якої користувачі поділяються на адміністраторів, викладачів і слухачів. Контроль доступу реалізовано із застосуванням role-based access control та permission classes фреймворку Django REST Framework.

Модуль управління користувачами забезпечує створення користувачів, редагування профілю, зміну ролей, керування правами доступу та перегляд інформації про користувачів системи.

Табл. 3.7 Функції модуля Users

| Функція            | Призначення             |
|--------------------|-------------------------|
| Registration       | Реєстрація користувачів |
| Authentication     | Авторизація             |
| Profile Management | Управління профілем     |
| Role Management    | Управління ролями       |
| Access Control     | Контроль доступу        |

Одним із центральних компонентів інформаційної системи є модуль управління курсами, який дозволяє створювати курси, редагувати інформацію про них, додавати навчальні матеріали, керувати слухачами та організовувати процес тестування.

Frontend частина модуля курсів реалізована у вигляді SPA-інтерфейсу з використанням React-компонентів.

Структуру модуля курсів наведено на рисунку 3.7.

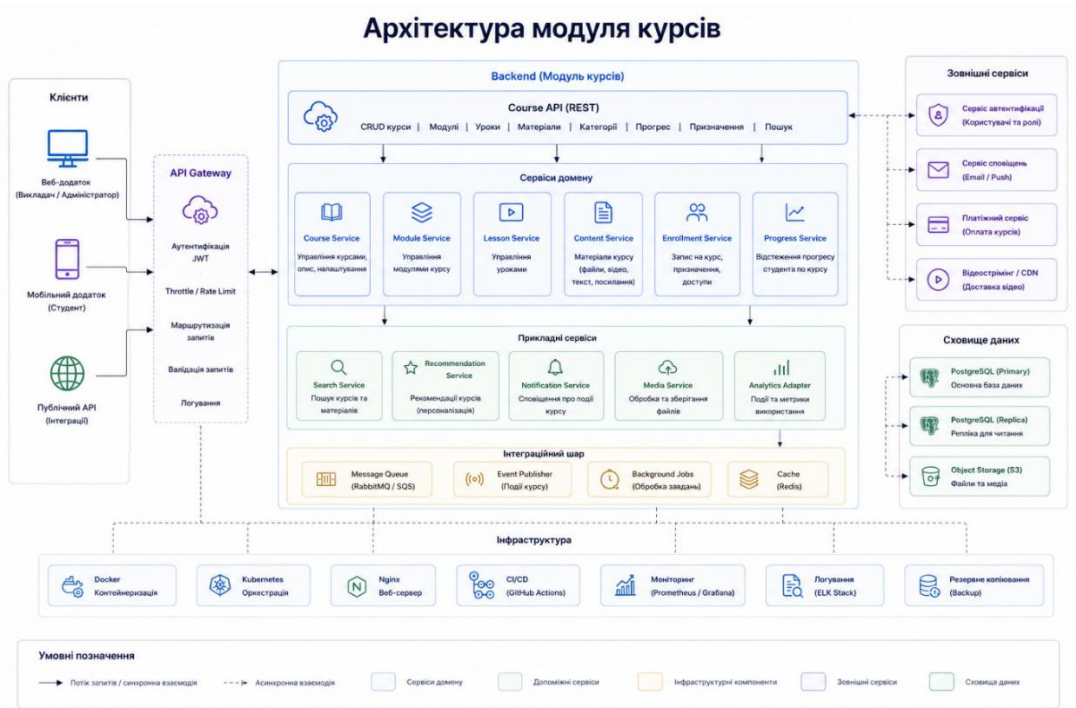


Рис. 3.7 Архітектура модуля курсів

Для організації навчального контенту реалізовано модуль навчальних матеріалів. Система підтримує:

завантаження PDF-файлів;

DOCX-документів;

ZIP-архівів;

мультимедійних матеріалів.

Усі файли зберігаються у file storage та пов'язані із відповідними курсами через PostgreSQL.

Табл. 3.8 Підтримувані типи навчальних матеріалів

| Тип файлу | Призначення             |
|-----------|-------------------------|
| PDF       | Методичні матеріали     |
| DOCX      | Документи та інструкції |
| ZIP       | Архіви матеріалів       |
| Image     | Графічні ресурси        |
| Video     | Відеоматеріали          |

Одним із найважливіших компонентів інформаційної системи є модуль тестування, у межах якого реалізовано створення тестів, формування тестових запитань, автоматичне оцінювання результатів, контроль кількості спроб проходження тестування та збереження отриманих результатів.

Модуль тестування взаємодіє з модулем управління курсами, системою сертифікації, PostgreSQL та REST API, що забезпечує цілісність і автоматизацію процесу контролю знань.

Схему роботи модуля тестування наведено на рисунку 3.8.

## Робота модуля тестування в системі NexLearn

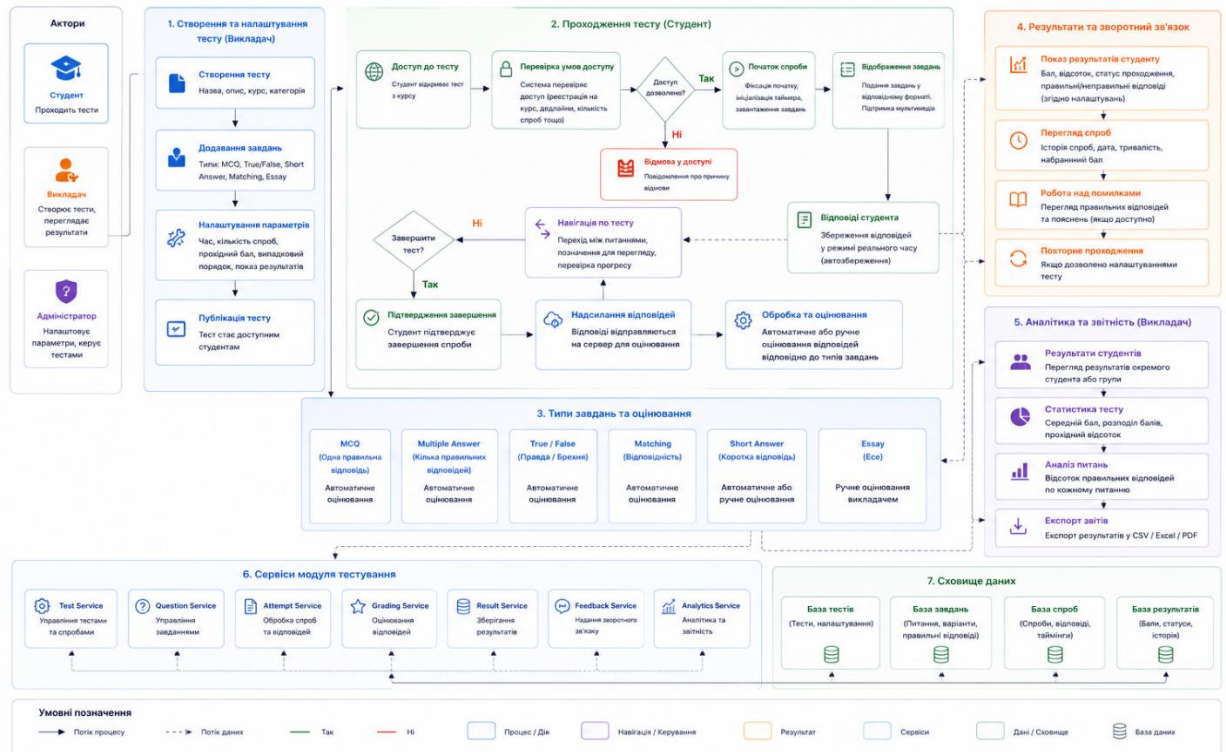


Рис. 3.8 Робота модуля тестування

Після завершення тестування система автоматично виконує перевірку результатів та визначає статус проходження тесту, кількість набраних балів і можливість генерації сертифіката для слухача.

Для автоматизації процесу видачі сертифікатів реалізовано окремий модуль сертифікації, який забезпечує генерацію UUID, створення PDF-сертифікатів, збереження документів та перевірку їхньої унікальності.

Табл. 3.9 Основні функції модуля сертифікації

| Функція         | Призначення                   |
|-----------------|-------------------------------|
| UUID Generation | Формування унікального номера |
| PDF Generation  | Створення сертифіката         |
| Validation      | Перевірка валідності          |
| Storage         | Збереження документа          |

Процес генерації сертифіката наведено на рисунку 3.9.



Рис. 3.9 Генерація сертифікатів у системі

Для взаємодії frontend та backend реалізовано REST API. У системі створено API-ендпоінти для:

- користувачів;

- курсів;
- тестів;
- результатів;
- сертифікатів.

Обмін даними між компонентами здійснюється у форматі JSON через HTTP-запити.

Табл. 3.10 – Основні REST API ендпоінти

| Endpoint           | Призначення |
|--------------------|-------------|
| /api/auth/         | Авторизація |
| /api/users/        | Користувачі |
| /api/courses/      | Курси       |
| /api/tests/        | Тести       |
| /api/results/      | Результати  |
| /api/certificates/ | Сертифікати |

Схему REST API наведено на рисунку 3.10.

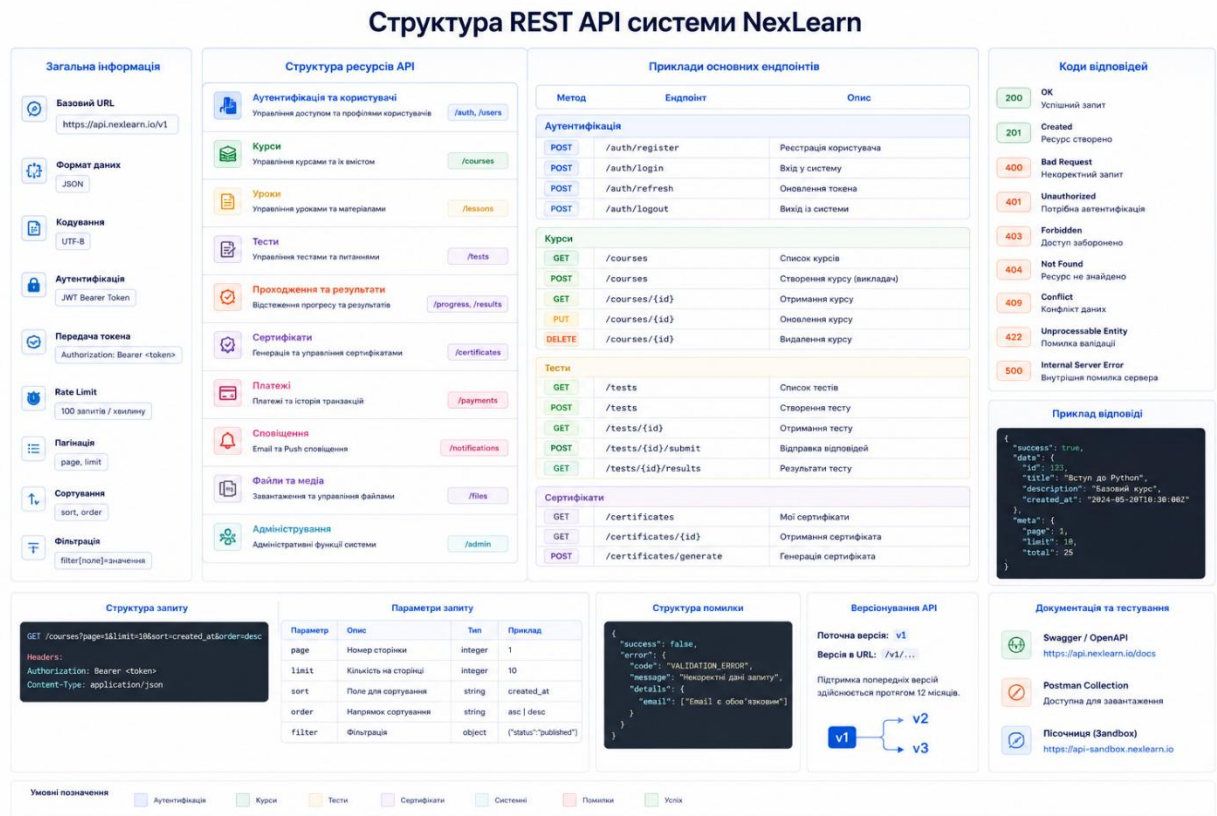


Рис. 3.10 Структура REST API системи

У процесі реалізації системи особливу увагу приділено інформаційній безпеці.

У системі реалізовано:

- JWT-автентифікацію;
- HTTPS;
- permission classes;
- rate-limit;
- ORM-захист від SQL Injection;
- захист від XSS.

Табл. 3.11. Реалізовані механізми безпеки

| Механізм       | Призначення              |
|----------------|--------------------------|
| JWT            | Автентифікація           |
| HTTPS          | Захист передачі даних    |
| RBAC           | Контроль доступу         |
| Rate-limit     | Захист від brute-force   |
| ORM Protection | Захист від SQL Injection |

Інтерфейс інформаційної системи реалізовано із використанням responsive design, що забезпечує коректну роботу вебзастосунку на персональних комп'ютерах, планшетах і мобільних пристроях.

Для оптимізації продуктивності системи використовуються механізми SSR та SSG у Next.js, ORM-оптимізація, select\_related, prefetch\_related та кешування запитів, що дозволяє підвищити швидкодію та ефективність обробки даних.

Табл. 3.12 – Механізми оптимізації продуктивності

| Механізм         | Призначення             |
|------------------|-------------------------|
| SSR              | Серверний рендеринг     |
| SSG              | Статична генерація      |
| ORM Optimization | Оптимізація SQL-запитів |
| Caching          | Зменшення навантаження  |
| Lazy Loading     | Оптимізація frontend    |

Таким чином, у процесі реалізації системи NexLearn було створено повноцінну веборієнтовану інформаційну систему, яка забезпечує автоматизацію навчального процесу, підтримку дистанційного навчання, тестування та сертифікацію користувачів, багаторольову модель доступу, безпечну REST API-взаємодію,

адаптивний вебінтерфейс, а також масштабованість і ефективність функціонування системи.

### 3.3. Тестування та оцінка ефективності

Після завершення реалізації основних модулів веб-орієнтованої інформаційної системи NexLearn було проведено комплексне тестування програмного забезпечення. Основною метою тестування є перевірка працездатності системи, відповідності функціональним вимогам, оцінка продуктивності, стабільності та безпечності програмного забезпечення.

У процесі тестування перевірялися коректність роботи функціональних модулів, взаємодія між frontend- і backend-компонентами, стабільність REST API, коректність функціонування бази даних, рольова модель доступу, механізми безпеки та продуктивність інформаційної системи.

Тестування системи проводилося поетапно й охоплювало функціональне тестування, інтеграційне тестування, API-тестування, тестування безпеки, тестування продуктивності та перевірку користувацького інтерфейсу.

Табл. 3.13 Основні види тестування системи

| Вид тестування     | Призначення                 |
|--------------------|-----------------------------|
| Функціональне      | Перевірка функцій системи   |
| Інтеграційне       | Перевірка взаємодії модулів |
| API-тестування     | Перевірка REST API          |
| Тестування безпеки | Перевірка захисту системи   |
| Навантажувальне    | Оцінка продуктивності       |
| UI-тестування      | Перевірка інтерфейсу        |

Схему процесу тестування наведено на рисунку 3.11.



Рис. 3.11 Загальна схема процесу тестування системи

На етапі функціонального тестування перевірялася коректність роботи основних модулів інформаційної системи, зокрема модулів реєстрації користувачів, авторизації, управління курсами, тестування, сертифікації та управління ролями користувачів.

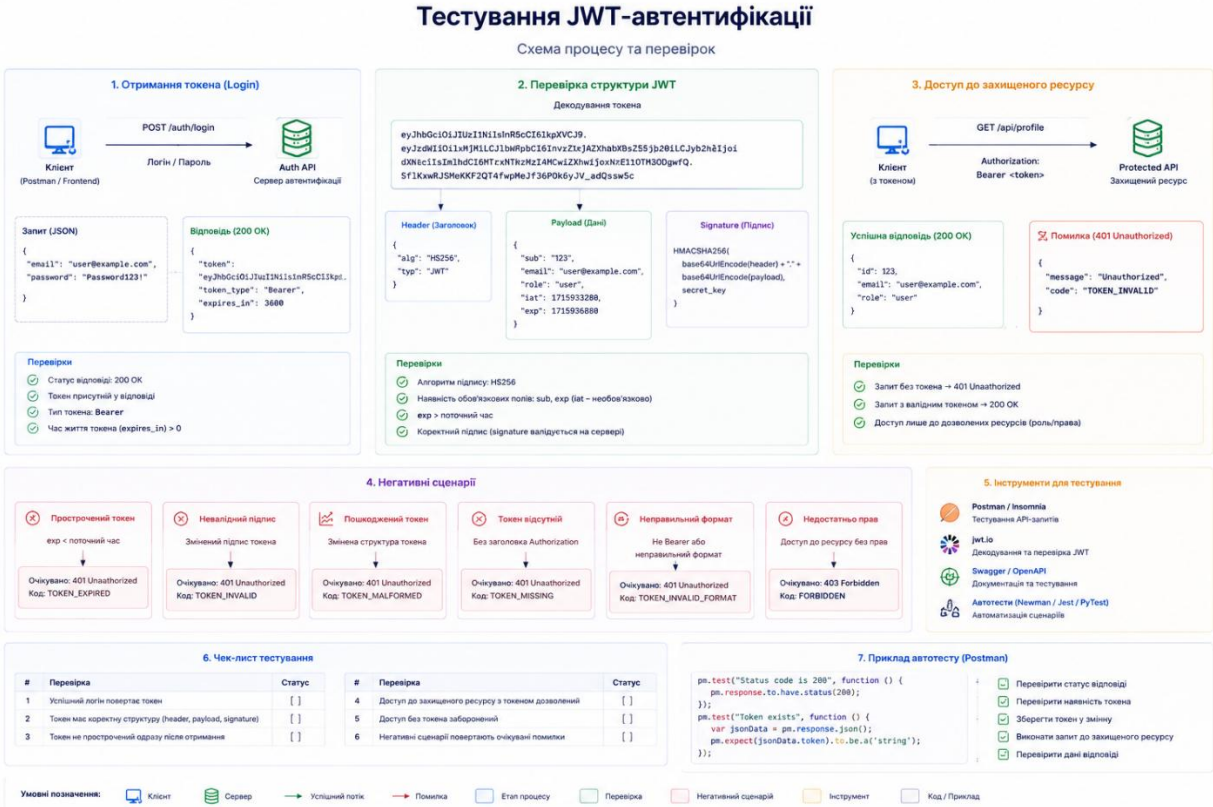
Для кожного функціонального модуля виконувалася перевірка правильності виконання операцій, обробки помилок, коректності збереження даних і відповідності отриманих результатів очікуваним значенням.

### Табл. 3.14 Результати функціонального тестування

| Модуль         | Перевірка                 | Результат |
|----------------|---------------------------|-----------|
| Authentication | Реєстрація та авторизація | Успішно   |
| Courses        | CRUD-операції курсів      | Успішно   |
| Tests          | Проходження тестів        | Успішно   |
| Certificates   | Генерація сертифікатів    | Успішно   |
| Users          | Role-based access         | Успішно   |

Особливу увагу було приділено тестуванню модуля автентифікації. У процесі тестування перевірялися механізми генерації JWT-токенів, оновлення access token, завершення користувацької сесії, обмеження доступу до API та захисту авторизованих маршрутів системи.

Схему тестування JWT-автентифікації наведено на рисунку 3.12.



### Рис. 3.12 Тестування JWT-автентифікації

Під час інтеграційного тестування перевірялася взаємодія frontend та backend, REST API та бази даних, модулів тестування та сертифікації, компонентів системи авторизації.

Тестування REST API проводилося шляхом надсилення HTTP-запитів до основних API-ендпоінтів системи:

/api/auth/;

/api/users/;

/api/courses/;

/api/tests/;

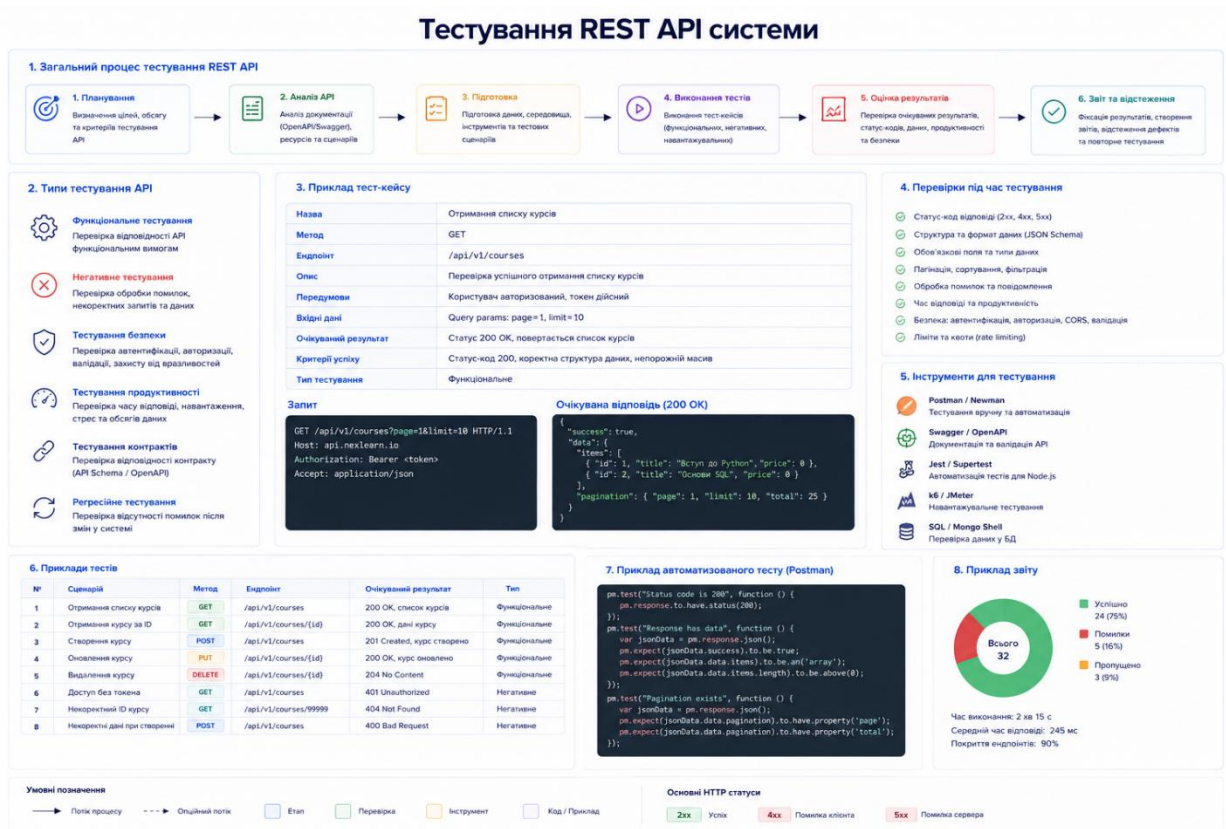
/api/certificates/.

У процесі тестування перевірялися, статуси HTTP-відповідей, правильність JSON-відповідей, permission classes, коректність обробки помилок, валідність JWT-токенів.

Табл. 3.15 Результати тестування REST API

| Endpoint          | Тип запиту | Результат |
|-------------------|------------|-----------|
| /api/auth/login   | POST       | Успішно   |
| /api/courses      | GET        | Успішно   |
| /api/courses      | POST       | Успішно   |
| /api/tests        | GET        | Успішно   |
| /api/certificates | GET        | Успішно   |

Схему тестування REST API наведено на рисунку 3.13.



Процес тестування сертифікації наведено на рисунку 3.14.



Табл. 3.17 Оцінка продуктивності системи

| Показник                   | Результат |
|----------------------------|-----------|
| Середній час відповіді API | < 300 мс  |
| Час авторизації            | < 200 мс  |
| Генерація сертифіката      | < 2 с     |
| Завантаження dashboard     | < 1 с     |

Схему оцінювання продуктивності наведено на рисунку 3.15.



Рис. 3.15 Оцінка продуктивності системи

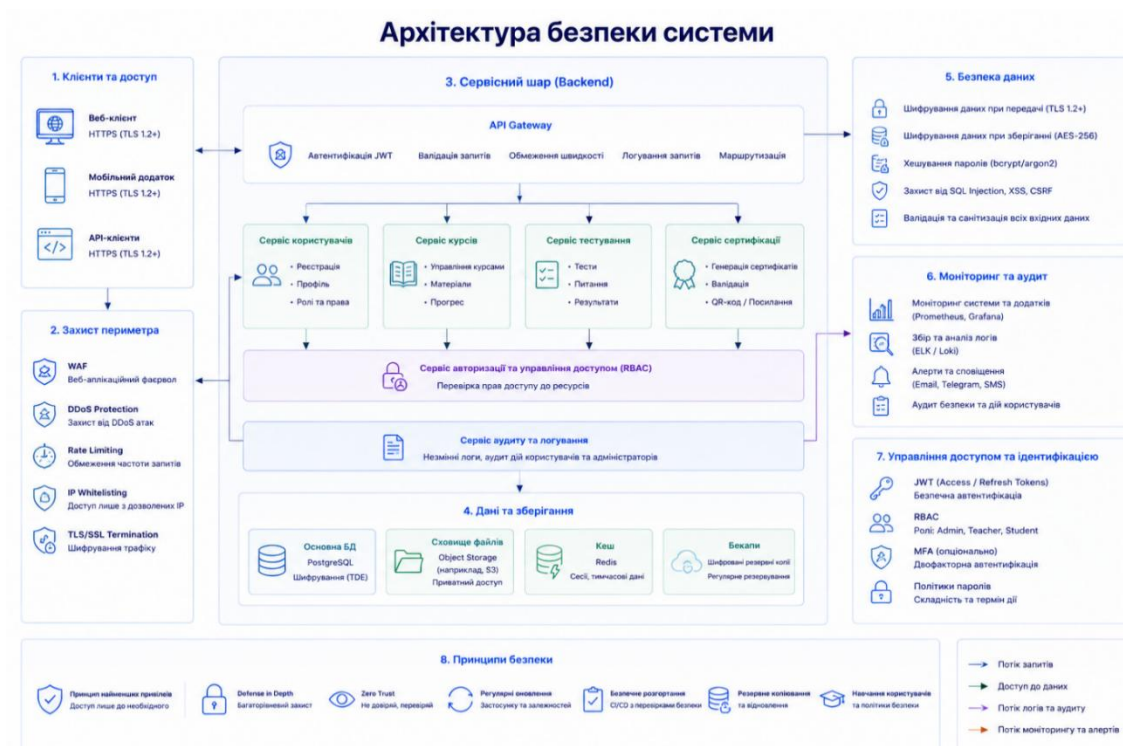
У процесі тестування безпеки перевірялися механізми захисту JWT-токенів, permission classes, налаштування CORS, використання HTTPS, захист від SQL Injection і XSS-атак, а також механізми rate-limit.

У системі реалізовано JWT Authentication, HTTPS, CSP, HSTS, ORM-захист та використання бібліотеки django-ratelimit, що забезпечує підвищений рівень безпеки веборієнтованої інформаційної системи.

Табл. 3.18 Результати тестування безпеки

| Механізм захисту         | Результат |
|--------------------------|-----------|
| JWT Authentication       | Успішно   |
| Role-based Access        | Успішно   |
| HTTPS                    | Успішно   |
| SQL Injection Protection | Успішно   |
| XSS Protection           | Успішно   |
| Rate-limit               | Успішно   |

Архітектуру безпеки системи наведено на рисунку 3.16.



### Рис. 3.16 Архітектура безпеки системи

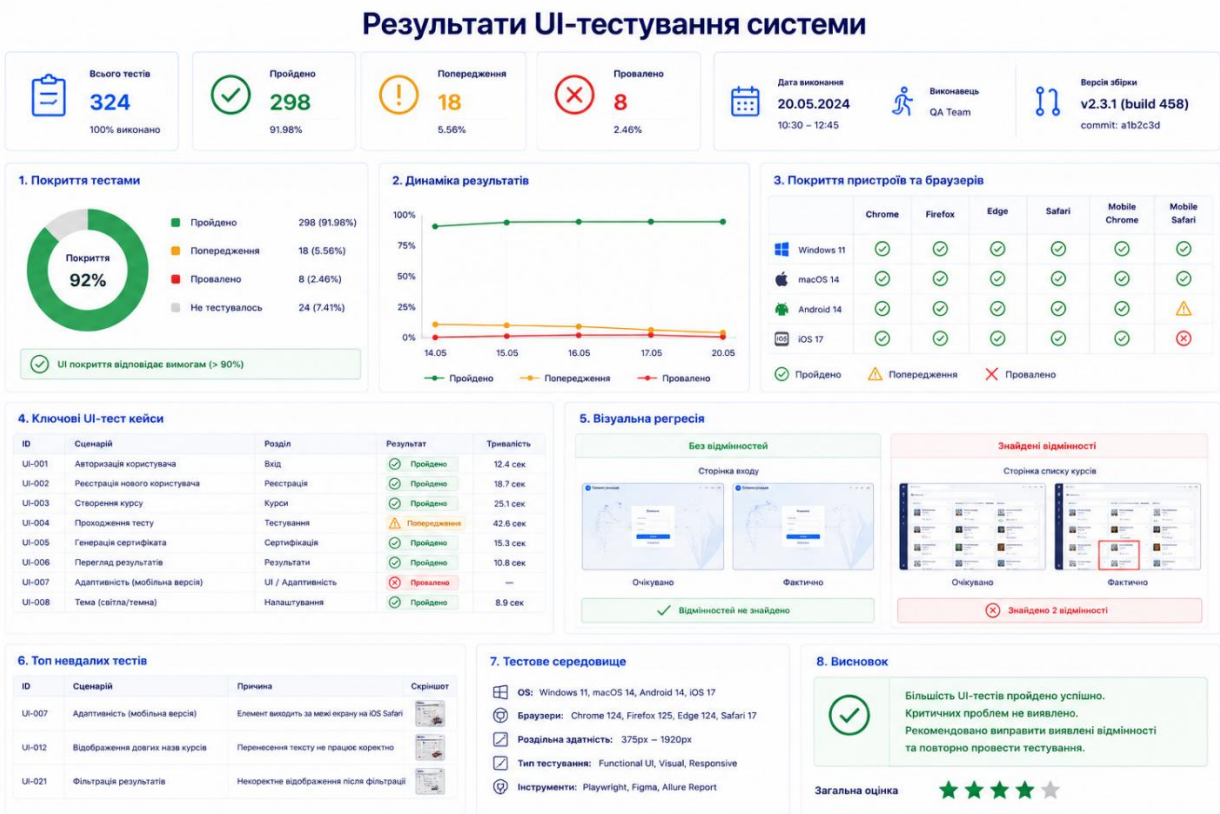
Для оцінювання зручності використання інформаційної системи було проведено UI-тестування, у межах якого оцінювалися адаптивність інтерфейсу, зручність навігації, швидкість доступу до функціональних можливостей системи та стабільність роботи frontend-компонента.

Тестування системи виконувалося у браузерях Google Chrome, Mozilla Firefox та Microsoft Edge, а також на мобільних пристроях і планшетах.

Табл. 3.19 Результати UI-тестування

| Критерій           | Результат |
|--------------------|-----------|
| Адаптивність       | Успішно   |
| Навігація          | Успішно   |
| Responsive Design  | Успішно   |
| Робота у браузерях | Успішно   |

Інтерфейс тестування системи наведено на рисунку 3.17.



- підтримує безпечну JWT-автентифікацію;
- забезпечує автоматизацію тестування та сертифікації;
- має достатній рівень продуктивності;
- підтримує адаптивний вебінтерфейс.

Проведене тестування підтвердило відповідність системи сформованим функціональним та нефункціональним вимогам, що дозволяє рекомендувати систему NexLearn для використання у центрах підвищення кваліфікації з метою автоматизації навчального процесу.

## ВИСНОВОК

У дипломній роботі вирішено актуальне завдання розробки веб-орієнтованої інформаційної системи супроводу навчального процесу в центрах підвищення кваліфікації. У процесі дослідження було виконано аналіз предметної області, спроектовано архітектуру системи, реалізовано програмне забезпечення та проведено тестування функціональних можливостей системи NexLearn.

У результаті аналізу предметної області встановлено, що сучасні центри підвищення кваліфікації потребують ефективних інструментів автоматизації освітньої діяльності, зокрема засобів управління курсами, організації дистанційного навчання, автоматизованого тестування, сертифікації слухачів та централізованого управління навчальними ресурсами.

Проведений аналіз існуючих систем управління навчанням показав, що універсальні LMS мають широкий функціонал, однак не повною мірою враховують специфіку короткострокових програм підвищення кваліфікації. Основними недоліками наявних рішень є надлишкова складність, недостатня адаптація до локальних потреб, обмежені можливості кастомізації та складність реалізації спеціалізованої бізнес-логіки.

На основі проведеного аналізу було сформовано функціональні та нефункціональні вимоги до веб-орієнтованої інформаційної системи супроводу навчального процесу.

У межах дипломної роботи спроектовано архітектуру системи NexLearn на основі клієнт-серверного підходу з використанням Next.js та React для frontend-частини, Django і Django REST Framework для backend-компонента, а також PostgreSQL для організації зберігання даних.

У процесі проєктування бази даних визначено основні сутності системи та реалізовано зв'язки між ними. Виконано нормалізацію структури бази даних, що

дозволило забезпечити цілісність інформації, ефективність зберігання даних, підтримку масштабованості системи та оптимізацію обробки запитів.

У системі реалізовано основні функціональні модулі, серед яких модуль автентифікації та авторизації, модуль управління користувачами, модуль управління курсами, модуль тестування, модуль сертифікації та REST API-модуль взаємодії компонентів.

Для забезпечення безпечної роботи системи реалізовано JWT-автентифікацію, role-based access control, HTTPS, CORS, CSP, HSTS, а також механізми захисту від SQL Injection і XSS-атак.

Реалізована система підтримує багаторольову модель доступу, автоматизоване тестування, генерацію сертифікатів, управління навчальними матеріалами, адаптивний вебінтерфейс та REST API-взаємодію між frontend- і backend-компонентами.

У процесі тестування перевірено функціональність програмних модулів, стабільність роботи REST API, коректність JWT-автентифікації, продуктивність системи, безпечність програмного забезпечення та адаптивність інтерфейсу.

Результати тестування підтвердили працездатність системи, стабільність взаємодії компонентів, відповідність функціональним і нефункціональним вимогам, достатній рівень продуктивності та коректність реалізації механізмів безпеки.

Практичне значення отриманих результатів полягає у можливості використання системи NexLearn для автоматизації навчального процесу в центрах підвищення кваліфікації. Використання системи дозволяє підвищити ефективність управління освітньою діяльністю, зменшити адміністративне навантаження, автоматизувати процес тестування та сертифікації, забезпечити централізоване управління освітніми ресурсами й покращити організацію дистанційного навчання.

Отже, поставлену мету дипломної роботи досягнуто, а всі визначені задачі виконано. Розроблена веб-орієнтована інформаційна система NexLearn може бути

використана у практичній діяльності центрів підвищення кваліфікації та має перспективи подальшого розвитку й розширення функціональних можливостей.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Django Software Foundation. Django Documentation [Електронний ресурс]. URL: <https://docs.djangoproject.com/>
2. Encode OSS Ltd. Django REST Framework Documentation [Електронний ресурс]. URL: <https://www.django-rest-framework.org/>
3. Meta Open Source. React Documentation [Електронний ресурс]. URL: <https://react.dev/>
4. Vercel. Next.js Documentation [Електронний ресурс]. URL: <https://nextjs.org/docs>
5. PostgreSQL Global Development Group. PostgreSQL Documentation [Електронний ресурс]. URL: <https://www.postgresql.org/docs/>
6. Auth0. Introduction to JSON Web Tokens [Електронний ресурс]. URL: <https://jwt.io/introduction/>
7. Axios. Axios Documentation [Електронний ресурс]. URL: <https://axios-http.com/docs/intro>
8. Tailwind Labs. Tailwind CSS Documentation [Електронний ресурс]. URL: <https://tailwindcss.com/docs>
9. Microsoft. TypeScript Documentation [Електронний ресурс]. URL: <https://www.typescriptlang.org/docs/>
10. Supabase. Supabase Documentation [Електронний ресурс]. URL: <https://supabase.com/docs>
11. Vercel. Vercel Documentation [Електронний ресурс]. URL: <https://vercel.com/docs>
12. Railway. Railway Documentation [Електронний ресурс]. URL: <https://docs.railway.com/>
13. GitHub. GitHub Docs [Електронний ресурс]. URL: <https://docs.github.com/>
14. Биков В. Ю. Технології хмарних обчислень у професійній освіті. Інформаційні технології і засоби навчання. 2019. Т. 70, № 2. С. 1–15.

15. Морзе Н. В., Глазунова О. Г. Сучасні інформаційні технології у професійній освіті. Київ : Видавництво КНУ, 2020. 320 с.
16. Жалдак М. І., Рамський Ю. С., Рафальська М. В. Інформаційні системи та технології в освіті. Київ : НПУ імені М. П. Драгоманова, 2018. 275 с.
17. Козяр М. М., Пархуць Л. Т. Проєктування інформаційних систем : навч. посіб. Львів : Львівська політехніка, 2021. 348 с.
18. Гужва В. М. Інформаційні системи і технології на підприємствах : навч. посіб. Київ : КНЕУ, 2019. 400 с.
19. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. New York : McGraw-Hill, 2020. 992 p.
20. Sommerville I. Software Engineering. 10th ed. Boston : Pearson, 2016. 816 p.
21. Fowler M. Patterns of Enterprise Application Architecture. Boston : Addison-Wesley, 2003. 533 p.
22. Moodle. Moodle Documentation [Електронний ресурс]. URL: <https://docs.moodle.org/>
23. Date C. J. An Introduction to Database Systems. 8th ed. Boston : Pearson, 2003. 1024 p.
24. Elmasri R., Navathe S. Fundamentals of Database Systems. 7th ed. Boston : Pearson, 2017. 1272 p.
25. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures : doctoral dissertation. Irvine : University of California, 2000.
26. Richards M., Ford N. Fundamentals of Software Architecture. Sebastopol : O'Reilly Media, 2020. 432 p.
27. Google. Google Classroom Help [Електронний ресурс]. URL: <https://support.google.com/edu/classroom/>
28. Instructure. Canvas LMS Documentation [Електронний ресурс]. URL: <https://developerdocs.instructure.com/services/canvas>

29. Anthology. Blackboard Learn Help [Электронный ресурс]. URL: <https://help.blackboard.com/Learn>
30. Open edX. Open edX Documentation [Электронный ресурс]. URL: <https://docs.openedx.org/en/latest/>
31. OWASP Foundation. OWASP Top 10 Web Application Security Risks. 2021 [Электронный ресурс]. URL: <https://owasp.org/www-project-top-ten/>
32. Mozilla Developer Network. HTTP Documentation [Электронный ресурс]. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP>
33. Mozilla Developer Network. REST API Concepts [Электронный ресурс]. URL: <https://developer.mozilla.org/en-US/docs/Glossary/REST>
34. ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Geneva : International Organization for Standardization, 2011.
35. Object Management Group. OMG Unified Modeling Language Specification. Version 2.5.1 [Электронный ресурс]. URL: <https://www.omg.org/spec/UML/2.5.1/>

# ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

Державний університет інформаційно-комунікаційних  
технологій

Кафедра Інформаційних систем та технологій

## Кваліфікаційна робота на тему:

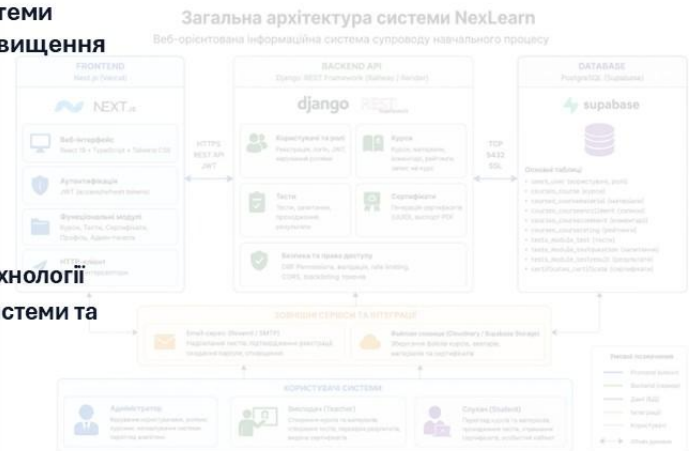
Розробка веб-орієнтованої інформаційної системи  
супроводу навчального процесу у центрах підвищення  
кваліфікації

на здобуття освітнього ступеня бакалавр  
зі спеціальності 126 Інформаційні системи та технології  
освітньо-професійної програми Інформаційні системи та  
технології

ВИКОНАЛА: АЛЛА МЕЛЬНИЧЕНКО

СТУД. ГР. ІСД-42

КЕРІВНИК: ОЛЕГ СЕНЬКОВ



## Актуальність роботи

Проблема цифровізації навчального процесу

Центри підвищення кваліфікації потребують швидкого управління короткостроковими курсами, слухачами, тестуванням і сертифікацією. Універсальні LMS часто є надлишково складними або недостатньо гнучкими, тому актуальною є розробка спеціалізованої веб-системи, адаптованої до потреб таких центрів.

Адміністративне навантаження  
ручний облік, розрізнені таблиці,  
повільна сертифікація

Короткострокові програми  
швидке створення курсів,  
оновлення матеріалів, контроль  
прогресу

Дистанційне навчання  
адаптивний вебінтерфейс,  
онлайн-тести, рольовий доступ

## Мета, об'єкт і предмет дослідження

Мета: розробити веб-орієнтовану інформаційну систему NexLearn для автоматизації супроводу навчального процесу.

Об'єкт: процес організації та супроводу навчання у центрах підвищення кваліфікації.

Предмет: методи, моделі й програмні засоби створення веб-орієнтованих інформаційних систем.

### Основні завдання

- 1) проаналізувати LMS та потреби центрів підвищення кваліфікації;
- 2) сформувати функціональні й нефункціональні вимоги;
- 3) спроектувати архітектуру, базу даних та ролі користувачів;
- 4) реалізувати frontend, backend, REST API та модулі системи;
- 5) провести тестування і оцінити працездатність рішення.

Мельниченко А.В. • 2

## Аналіз існуючих LMS

Порівняння універсальних платформ із потребами центрів

| Платформа        | Переваги                          | Обмеження                  |
|------------------|-----------------------------------|----------------------------|
| Moodle           | відкритий код, великий функціонал | складність адміністрування |
| Google Classroom | простота використання             | обмежена кастомізація      |
| Canvas           | сучасний інтерфейс, REST API      | частина функцій платна     |
| Blackboard       | потужні корпоративні можливості   | висока вартість            |
| Open edX         | масштабованість                   | складність розгортання     |

**Отже, для короткострокових програм важливі просте адміністрування, автоматична сертифікація, швидке створення курсів, REST API та адаптивний інтерфейс.**

Мельниченко А.В. • 3

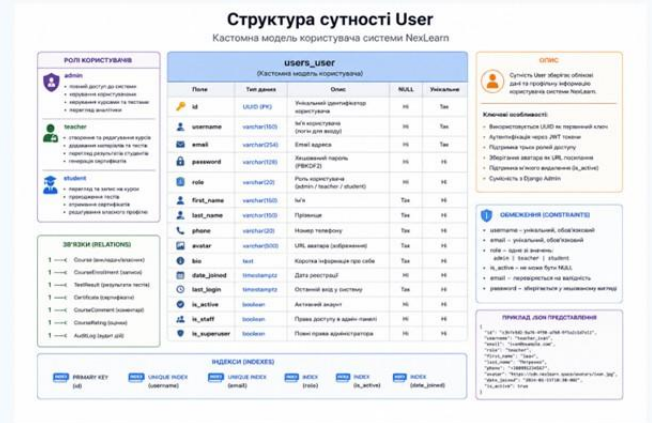
# Вимоги до системи NexLearn

## Функціональні вимоги

- реєстрація та авторизація користувачів;
- управління ролями, курсами та матеріалами;
- створення тестів і автоматичне оцінювання;
- генерація сертифікатів;
- REST API-взаємодія між модулями.

## Нефункціональні вимоги

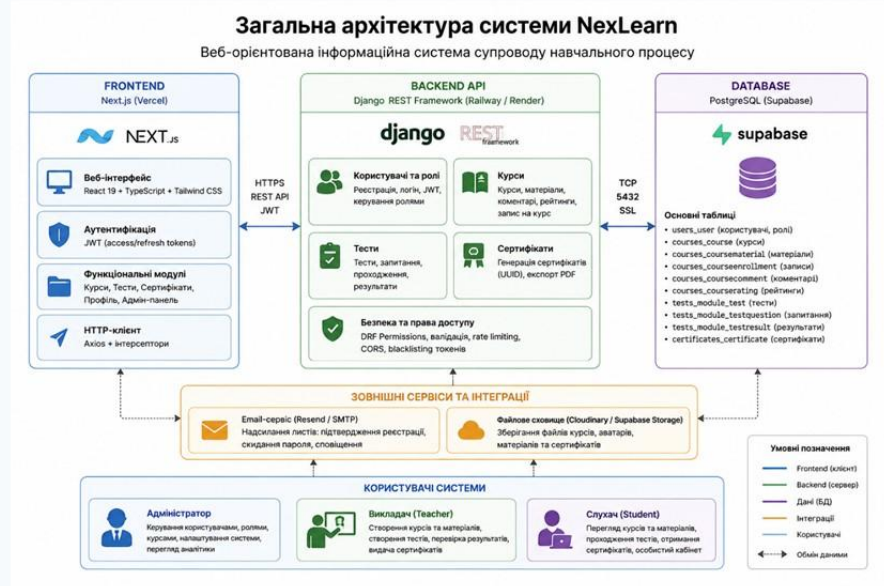
- продуктивність і масштабованість;
- захист доступу та персональних даних;
- адаптивність інтерфейсу;
- надійність, журналювання та резервування.



Мельниченко А.В. • 4

# Загальна архітектура системи NexLearn

Клієнт-серверна модель із REST API та реляційною БД



Мельниченко А.В. • 5

# Frontend та Backend частини системи

1. Frontend — дерево



Frontend: Next.js, React, TypeScript, Tailwind CSS;  
адаптивний інтерфейс, сторінки курсів, тестування,  
сертифікація, адмін-панель.

2. Backend — дерево



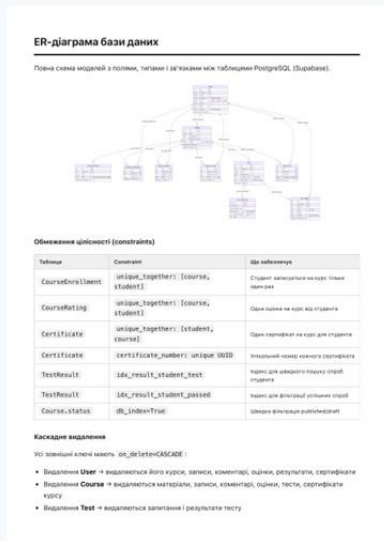
Backend: Django та Django REST Framework; бізнес-логіка,  
автентифікація, управління курсами, тестами,  
сертифікатами та API.

## REST API та безпека доступу



Компоненти взаємодіють через HTTP-запити у форматі JSON. Для захищеного доступу використовується JWT, контроль ролей, HTTPS і захист API.

# Проектування бази даних



База даних побудована на PostgreSQL та містить сутності користувачів, курсів, матеріалів, записів на курси, тестів, результатів і сертифікатів.

# Користувацькі ролі та бізнес-процеси



## Реалізація основних модулів



Модуль курсів

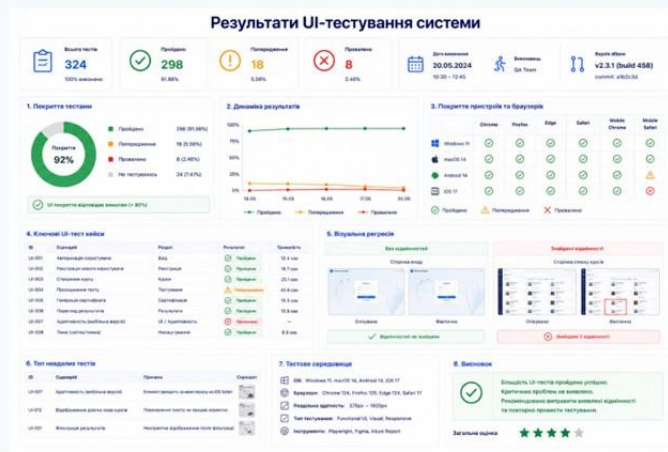


Модуль тестування



Генерація сертифікатів

## Тестування та результати



## Основні результати

- реалізовано веб-систему NexLearn для супроводу навчального процесу;
- підтримано ролі адміністратора, викладача та слухача;
- реалізовано управління курсами, тестуванням і сертифікатами;
- забезпечено REST API, JWT-автентифікацію та роботу з PostgreSQL;
- тестування підтвердило працездатність і відповідність базовим вимогам.

Практичне значення: система може бути використана для зменшення адміністративного навантаження та підвищення ефективності організації навчання.

---

## Апробація роботи

Дана робота пройшла апробацію на таких конференціях:

VII Науково-технічна конференція «Сучасний стан та перспективи розвитку IoT»

Всеукраїнська науково-технічна конференція «Технології цифрового розвитку: програмні системи та децентралізація»

**Дякую за увагу!**