Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

РОЗРОБКА ПЛАТФОРМИ ДЛЯ ОНЛАЙН-КУРСІВ З ВИКОРИСТАННЯМ REACT, DRUPAL ТА MYSQL

DEVELOPMENT OF A PLATFORM FOR ONLINE COURSES USING REACT, DRUPAL AND MYSQL

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконала: здобувач вищої освіти
групи ІПЗз-41
Бодун Софія Олександрівна

Керівник:
Повстяна Юлія Славомирівна

Кваліфікаційну роботу
допущено до захисту
«10» 06 2025 р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Наталія Миколаївна

_____
(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних та інформаційних технологій*
Кафедра *інженерії програмного забезпечення*
Ступінь вищої освіти *бакалавр*
Галузь знань: *12 «Інформаційні технології»*
Спеціальність: *121 «Інженерія програмного забезпечення»*
Освітня програма: «*Інженерія програмного забезпечення»*

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____
«20» 12 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Бодун Софії Олександрівні
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *«Розробка платформи для онлайн-курсів з використанням React, Drupal та MySQL».*
Керівник роботи: *к.т.н., доцент Повстяна Ю. С.*
затверджені наказом закладу вищої освіти від *«19» грудня 2024 р.* № *474/01-02*
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра *«10» червня 2025 р.*
3. Вихідні дані до роботи: *CMS Drupal для побудови серверної частини та управління контентом, Vite та бібліотека React для реалізації клієнтської частини, середовище розробки Docksal на основі Docker для локального розгортання проекту, система керування базами даних MySQL для збереження даних.*
4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): *кваліфікаційна робота передбачає розробку адаптивної платформи для онлайн-курсів із використанням React, Drupal і MySQL. Основні завдання: створення інтерфейсу, системи управління контентом, автентифікації, оцінювання знань.*
5. Перелік графічного матеріалу: *кількість рисунків: 15, кількість додатків: 6.*

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	*Повстяна Ю. С.*		
Специфікація вимог до розробленої системи	*Повстяна Ю. С.*		
Розробка об'єкта проектування	*Повстяна Ю. С.*		
Нормоконтроль	*Вознюк А. В.*		
Гарант ОП	*Ліщина Н. М.*		
Показник запозичень тексту	1,91 %		
Академічна доброчесність	*Повстяна Ю. С.*		

7. Дата видачі завдання: *«20» грудня 2024 р.*

№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 04.02.2025 р.	вик
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 01.03.2025 р.	вик
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 15.03.2025 р.	вик
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 29.03.2025 р.	вик
5	Практична реалізація об'єкта проектування та розробка бази даних	до 26.04.2025 р.	вик
6	Тестування та налагодження об'єкта проектування	до 03.05.2025 р.	вик
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру	до 10.06.2025 р.	вик

Здобувач вищої освіти _____ Софія БОДУН

Керівник кваліфікаційної роботи _____ Юлія ПОВСТЯНА

АНОТАЦІЯ

Бодун С. О. Розробка платформи для онлайн-курсів з використанням React, Drupal та MySQL. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності 121 «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота бакалавра складається з вступу, 3 розділів, висновків, списку використаних джерел, додатків.

У першому розділі здійснено аналіз предметної області, досліджено сучасні веб-технології для створення платформ для створення курсів для онлайн-навчання, а також сформульовано завдання кваліфікаційної роботи. В другому розділі розроблено архітектуру програмного забезпечення, визначено технічні вимоги, створено UML-діаграму. У третьому розділі реалізовано функціонал платформи із використанням React та серверної логіки на базі CMS Drupal, а також ефективність інтеграції баз даних за допомогою MySQL. У висновках узагальнено результати виконаної роботи, підтверджено досягнення поставлених завдань та наведено рекомендації щодо подальшого вдосконалення платформи.

Ключові слова: React, Drupal, Drupal Headless, CMS, LMS-системи, тренінги, курси, фільтрація, безпека, автентифікація.

ABSTRACT

Bodun S. Development of a Platform for Online Courses Using React, Drupal and MySQL. Manuscript. Bachelor's qualification work in the educational program "Software Engineering", specialty 121 Software Engineering. Lutsk National Technical University. Lutsk, 2025.

The bachelor's qualification work consists of an introduction, three chapters, conclusions, a list of references, and appendices.

The first chapter analyzes the subject area, explores modern web technologies for creating platforms for online learning courses, and formulates the tasks of the qualification work. The second section develops the software architecture, defines technical requirements, and creates a UML diagram. The third section implements the platform functionality using React and server logic based on CMS Drupal, as well as the efficiency of database integration using MySQL. The conclusions summarize the results of the work performed, confirm the achievement of the set tasks, and provide recommendations for further improvement of the platform.

Keywords: React, Drupal, Drupal Headless, CMS, LMS, training, courses, filtering, security, authentication.

ЗМІСТ

ВСТУП

Актуальність теми. На сьогоднішній день сучасні виклики в галузі освіти України вимагають відновлення нових підходів до організації навчального процесу. Унаслідок наслідків COVID-19 та повномасштабної війни велика кількість людей перейшла на дистанційне навчання. Зростання попиту на дистанційне навчання виявило низку проблем у функціонуванні існуючих платформ це є нестабільне інтернет-з'єднання, недостатня інтерактивність систем, а також технічна нерівність серед користувачів значно ускладнить ефективне засвоєння матеріалу.

Метою кваліфікаційної роботи бакалавра є розробка власної платформи для освітніх курсів у мережі для ефективної організації навчального процесу та високої інтерактивності процесу і доступності для користувачів у сучасних викликах.

Об'єктом кваліфікаційної роботи бакалавра є процес розробки веб-платформи для онлайн-навчання з використанням сучасних веб-технологій.

Предметом кваліфікаційної роботи бакалавра є програмні та архітектурні рішення, що використовуються для створення платформи, включаючи застосування Drupal для бекенду, React та Vite для реалізації інтерфейсу для користувачів, а також MySQL для збереження даних.

Завданням на кваліфікаційну роботу є реалізація низки завдань, спрямованих на реалізацію функціональних і технічних аспектів розробки, а саме:

– розробити інтерактивний інтерфейс на React для зручності користувачів (реєстрація, курси, тести, взаємодія);

– інтегрувати Drupal як CMS для управління навчальним контентом та контролю доступу;

– реалізувати безпечну систему автентифікація через MySQL з шифруванням даних;

– додати гейміфікацію (бали, лідерборди, нагороди) для підвищення мотивації студентів;

– забезпечити адаптивність платформи для роботи на слабких пристроях та при нестабільному інтернеті;

– створити систему оцінювання знань з автоматичною перевіркою тестів та зворотним зв’язком;

– протестувати та оптимізувати платформу для стабільної роботи в різних умовах.

РОЗДІЛ 1
АНАЛІЗ ПЛАТФОРМ ЕЛЕКТРОННОГО НАВЧАННЯ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1. Аналіз сучасного стану проблеми

Сучасні системи управління навчанням (Learning Management Systems, LMS) є ключовим інструментом для організації дистанційної освіти. Проте, як зазначають у методичних рекомендаціях, існуючі платформи мають низку обмежень, які загострилися в умовах пандемії COVID-19 та повномасштабної війни в Україні. Згідно з даними Центру технологій дистанційного навчання, основним викликом для української освіти є відсутність адаптованих рішень, які б враховували специфіку локальних потреб та технічні обмеження [1].

Аналіз міжнародних LMS:

1) Moodle (рис. 1.1) є однією з найпоширеніших відкритих LMS, широко використовуваною у вищій освіті. Її архітектура базується на моделі клієнт-сервер, що забезпечує гнучкість у налаштуванні. Однак, застарілий інтерфейс та висока складність адміністрування можуть обмежувати її використання в умовах обмежених ресурсів. Дослідження показують, що близько 60 % українських університетів використовують Moodle, але 75 % з них відзначають проблеми з інтуїтивністю інтерфейсу та недостатньою підтримкою інтерактивних функцій [2];

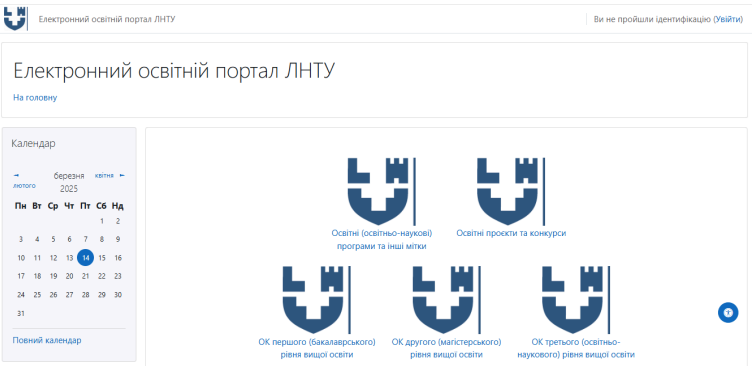


Рисунок 1.1 – Moodle в ЛНТУ [3]

2) Open edX (рис. 1.2). Open edX використовує мікросервісну архітектуру, що дозволяє ефективно масштабувати систему для задоволення потреб великих організацій та урядів. Хоча Open edX здатна підтримувати великі навантаження, її впровадження може вимагати значних обчислювальних ресурсів та технічної експертизи. Це може стати викликом для малих навчальних закладів з обмеженою інфраструктурою та ресурсами. Відсутність достатніх технічних можливостей може ускладнити розгортання та підтримку платформи.

В Україні платформа Open edX була успішно адаптована для проекту «Всеукраїнська школа онлайн», який надає доступ до освітніх матеріалів для учнів 5-11 класів. Цей проект демонструє можливість використання Open edX в українських умовах, хоча його реалізація потребувала належної технічної підтримки та ресурсів [4];

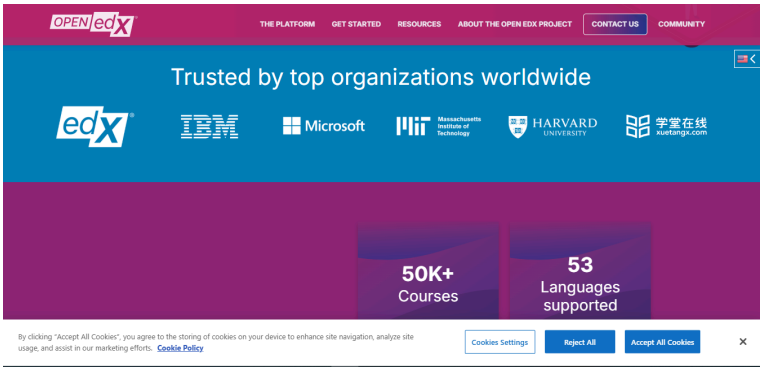


Рисунок 1.2 – Система Open edX [5]

3) Coursera та Udemy. Coursera є однією з провідних онлайн-освітніх платформ, яка пропонує понад 2000 курсів у більш ніж 180 спеціалізаціях, залучаючи близько 25 мільйонів слухачів [6]. Платформа співпрацює з понад 200 провідними університетами та компаніями для забезпечення онлайн-навчання по всьому світу.

Udemy (рис. 1.3) є найбільшою у світі навчальною платформою за кількістю онлайн-курсів, пропонуючи понад 17 000 курсів у різних галузях та категоріях, таких як Cloud Computing, Data Science, Development, IT Operations, Leadership & Management, Marketing та інші [7]. Для українських студентів

Udemy відкрила доступ до 7000 курсів зі своєї колекції, що входять до топ 3 % серед усього маркетплейсу [8].

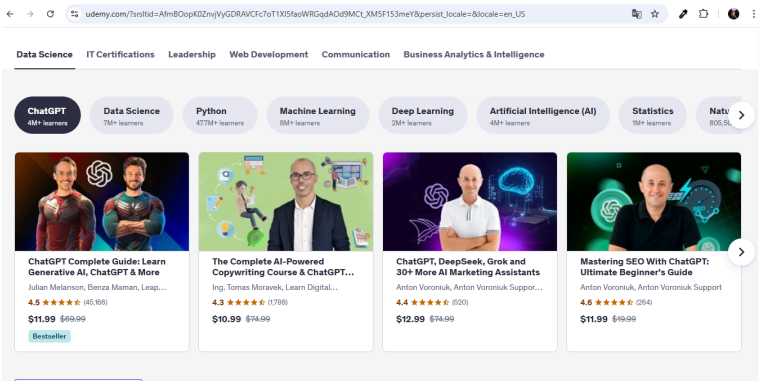


Рисунок 1.3 – Перелік навчальних курсів на Udemy [9]

Аналіз українських LMS:

1) Prometheus (рис. 1.4) є найбільшою платформою безкоштовної онлайн-освіти в Україні, яка надає доступ до якісних курсів від провідних університетів та організацій. Платформа пропонує широкий спектр курсів, включаючи відеолекції, практичні завдання, вебінари та форуми для спілкування з викладачами. Завдяки мобільним застосункам, навчатися можна будь-де і будь-коли. Однак, загальною проблемою масових онлайн-курсів є низький відсоток слухачів, які успішно завершують навчання [10]. Це може бути пов’язано з недостатньою мотивацією або браком часу у студентів;

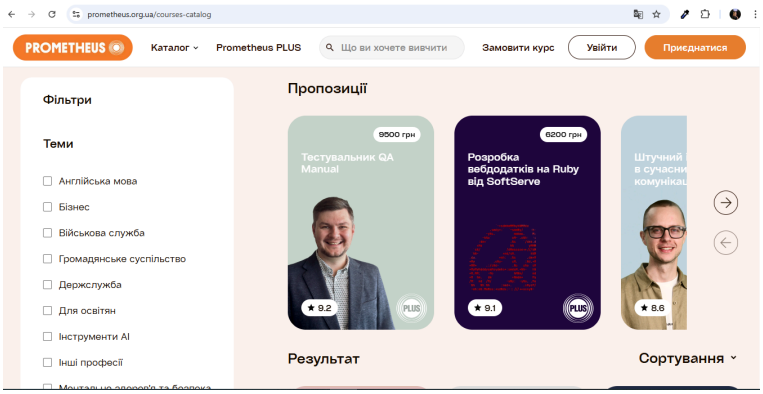


Рисунок 1.4 – Каталог курсів на Prometheus [11]

2) Всеукраїнська школа онлайн (ВШО) (рис. 1.5) – це ініціатива, створена під час пандемії для підтримки шкільного навчання в Україні. Основний фокус зроблено на загальноосвітніх дисциплінах, однак платформа має обмежений функціонал, що ускладнює її застосування для повноцінного дистанційного навчання в університетах.

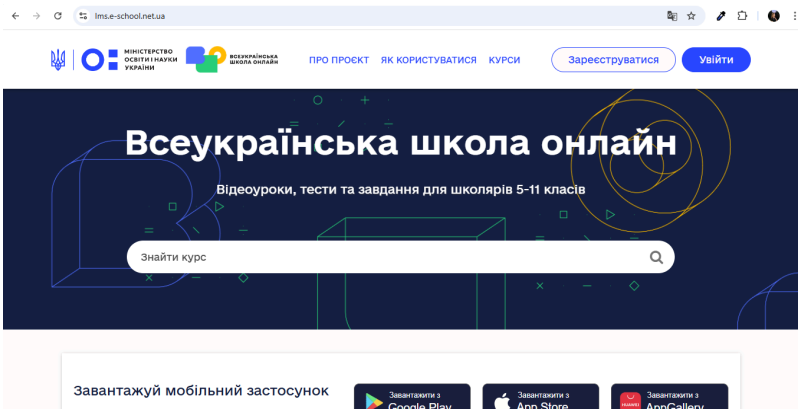


Рисунок 1.5 – Сайт Всеукраїнської школи онлайн [12]

Аналіз існуючих систем показує, що незважаючи на їхні переваги, вони мають суттєві обмеження, особливо в умовах воєнного часу та частих кризових ситуацій. Основні проблеми, які виявлено:

– складність налаштування та використання, багато іноземних платформ вимагають значних ресурсів для інтеграції та підтримки;

– обмежена функціональність українських платформ,які хоч і адаптовані під місцеві освітні потреби, не можуть повністю замінити комплексні LMS;

– безпекові ризики такі як загроза кібератак і втрати персональних даних є особливо актуальною в умовах війни.

Зважаючи на всі наведені фактори, виникає об’єктивна необхідність у розробці нової LMS, яка відповідатиме викликам сьогодення. Вона має враховувати технічні обмеження, забезпечувати безперебійний доступ до навчального контенту, бути зручною для використання та захищеною від зовнішніх загроз. Це дозволить створити ефективне та адаптивне освітнє

середовище, яке відповідатиме потребам українських студентів і викладачів в умовах воєнного стану та подальшого відновлення країни.

1.2. Постановка завдання на кваліфікаційну роботу бакалавра

Метою моєї кваліфікаційної роботи є розробка адаптованої платформи для онлайн-курсів, що використовує сучасні веб-технології React, Drupal та MySQL, яка б відповідала потребам онлайн-освіти в умовах війни в Україні. У зв’язку з високими вимогами до якості та безпеки таких платформ, а також зважаючи на специфічні виклики, що виникають під час дистанційного навчання, моїм завданням є вирішення таких ключових питань:

– розробка інтерактивного інтерфейсу користувача на основі React, що забезпечить зручність та доступність платформи для всіх учасників навчального процесу, зокрема для студентів та викладачів. Інтерфейс повинен включати функціональність для реєстрації, перегляду курсів, проходження тестів та отримання результатів, а також можливість взаємодії між користувачами;

– інтеграція Drupal як системи керування контентом (CMS) для ефективного управління навчальними матеріалами та курсами на платформі. Це дозволить адміністраторам додавати та редагувати матеріали, а також здійснювати контроль за доступом до різних розділів платформи, забезпечуючи таким чином зручне керування навчальним процесом;

– розробка системи реєстрації та автентифікації користувачів з використанням MySQL, що дозволить безпечно зберігати дані студентів та викладачів, а також забезпечити чітке управління правами доступу для різних категорій користувачів (студенти, викладачі, адміністратори). Це включає в себе також забезпечення безпеки даних користувачів через використання сучасних методів шифрування;

– впровадження елементів гейміфікації для підвищення мотивації студентів. Зокрема, планується реалізувати систему балів, лідербордів, а також віртуальні нагороди за успішно виконані завдання та активність на платформі.

Це допоможе збільшити рівень залученості студентів до навчального процесу та зробити навчання більш цікавим та інтерактивним;

– створення адаптивної платформи, що буде оптимізована для роботи на різних пристроях, зокрема на старих комп’ютерах та мобільних телефонах. Враховуючи проблеми з доступом до стабільного інтернет-з’єднання в Україні під час війни, також важливо забезпечити низькі вимоги до технічного обладнання;

– розробка системи для оцінювання знань студентів, яка включатиме можливості для автоматичного оцінювання тестів, виконаних завдань, а також надання зворотного зв’язку викладачами. Це дозволяє об’єктивно оцінювати рівень засвоєння матеріалу та забезпечить ефективний контроль за процесом навчання;

– тестування та оптимізація розробленої платформи, що включатиме перевірку її працездатності на різних пристроях, оцінку продуктивності, швидкості завантаження та виявлення можливих помилок. Це дозволить забезпечити стабільну роботу платформи навіть в умовах непередбачених технічних проблем.

Розробка такої платформи допоможе вирішити багато проблем сучасного онлайн-навчання, таких як недостатня інтерактивність, відсутність єдиної системи для навчання, проблеми з безпекою та доступністю.

Висновки до розділу 1

У першому розділі проведено аналіз сучасних платформ електронного навчання (LMS), що використовуються в Україні та світі. Аналіз українських рішень (Prometheus, ВШО) виявив їхню орієнтацію на доступність, проте обмежену функціональність для повноцінного дистанційного навчання у ЗВО.

РОЗДІЛ 2
СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1. Аналіз та визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Процес розробки програмного забезпечення починається з аналізу вимог, що передбачає виявлення функціональних потреб системи та очікувань її користувачів. Цей етап критично важливий, оскільки дозволяє сформувати цілісне уявлення про те, що саме повинна робити система, незалежно від технічних деталей реалізації. У результаті аналізу зменшуються ризики нерозуміння між замовником і розробником, що напряму впливає на якість кінцевого продукту.

Для чіткої візуалізації вимог системи та сценаріїв її використання, було виконано первинне проектування системи, яке включає створення UML-діаграм таких, як діаграма класів та діаграма прецедентів [13].

Діаграма класів (рис. 2.1) відображає структуру програмної системи – її основні сутності, атрибути, методи та взаємозв’язки між ними [14].

Вона демонструє статичну модель об’єктів, що існують у межах навчальної платформи. У структурі навчальної системи я виділила такі основні класи, як User, Course, Modules, Quiz, Certificate та інші. Клас User представляє користувача системи та має атрибути, такі як ім’я, прізвище, електронна пошта, пароль, а також роль у системі (адміністратор, викладач або студент). Користувачам надаються різні повноваження залежно від призначеної ролі. Наприклад, студенти можуть проходити курси, здавати завдання та переглядати результати, тоді як викладачі можуть створювати та редагувати курси. Адміністратор має перевагу, оскільки керує всіма користувачами та навчальними матеріалами.

Клас Course пов’язаний із класом User (викладачем), який створив цей курс. Кожен курс складається з одного або кількох модулів, які у свою чергу можуть містити навчальні матеріали, завдання чи тести. Після проходження

курсу студент може отримати сертифікат, який підтверджує здобуті знання. Така структура дозволяє чітко організувати навчальний процес і забезпечити логічну побудову платформи.

Значну увагу також приділено можливості комунікації між учасниками навчального процесу, адже підтримка соціальної взаємодії підвищує залученість і ефективність навчання. Для цього до структури були додані класи Forum і Topic. Кожен курс має свій форум, у якому створюються топіки для обговорень. Клас Topic пов’язаний із автором, містить дату створення та зміст повідомлення. Завдяки цій моделі платформа перетворюється не лише на інструмент для індивідуального навчання, а й на середовище для колективної взаємодії.

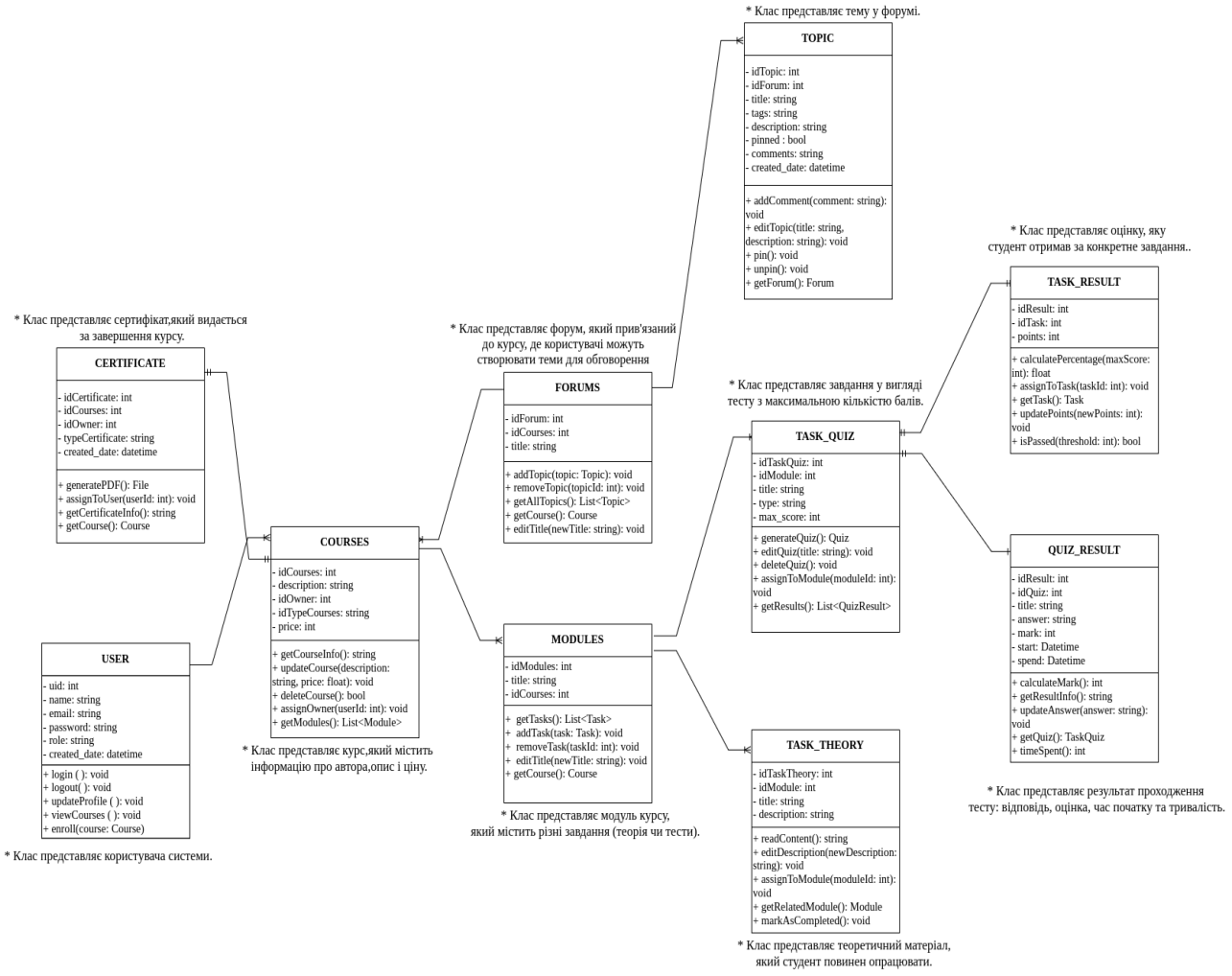


Рисунок 2.1 – Діаграма класів

Діаграма прецедентів (рис. 2.2) дозволяє відобразити систему з точки зору взаємодії користувачів із нею [15]. Для побудови цієї діаграми було виокремлено групу осіб, які взаємодіють із системою, – так званих акторів. Далі було ідентифіковано варіанти використання системи (прецеденти) та для кожного з них розроблено відповідний сценарій. Кожен актор має свій набір доступних дій, що відповідають його функціональності в межах системи.

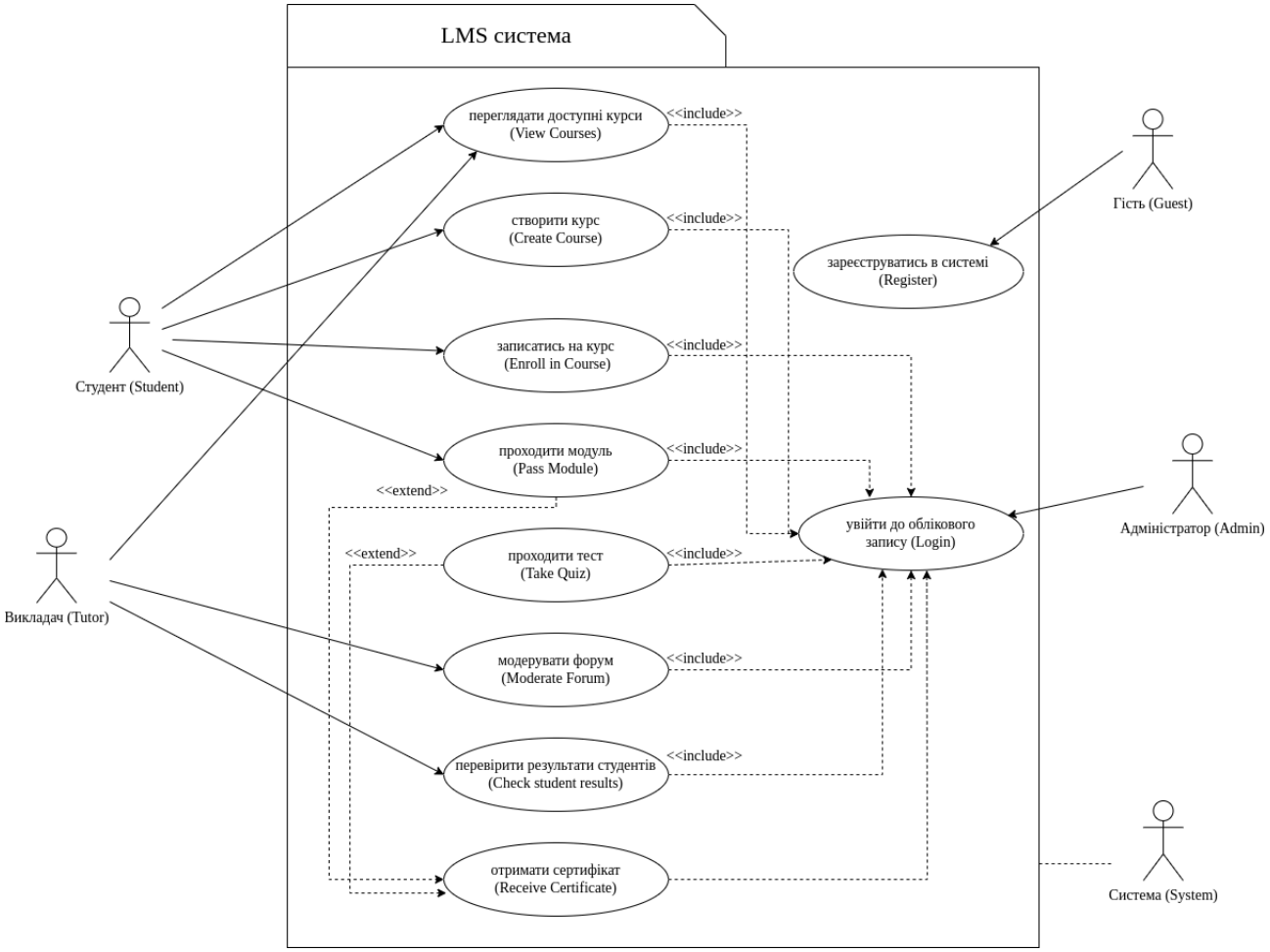


Рисунок 2.2 – Діаграма прецедентів

Студент має змогу:

– реєструватися та переглядати доступні курси;

– проходити модулі, виконувати тести;

– отримувати сертифікати;

– брати участь у форумних обговореннях.

Викладач має ширші можливості:

– формування навчального контенту, а саме він може створювати і редагувати курси, наповнювати їх модулями, завданнями та тестами;

– модерація обговорень у форумах, що належать до його курсів.

Адміністратор керує:

– обліковими записами користувачів;

– загальною статистикою проходження курсів.

Діаграма прецедентів дозволяє чітко окреслити межі відповідальності кожного типу користувача, наочно продемонструвати функціональність платформи та сформулювати вимоги до реалізації. Це значно полегшує подальше проектування системи, її реалізацію та тестування, оскільки всі можливі сценарії використання визначені ще на ранньому етапі розробки.

2.2. Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Проектування архітектури освітньої онлайн-платформи потребує комплексного підходу з урахуванням функціональних вимог, потенційного масштабування та особливостей взаємодії користувачів різних ролей з системою. Вибір технологій здійснювався на основі таких критеріїв:

– продуктивність при обробці значних обсягів даних та одночасній роботі багатьох користувачів;

– безпека збереження персональних даних та навчальних матеріалів;

– масштабованість для подальшого розширення функціональності;

– підтримка сучасних стандартів веброзробки;

– можливість реалізації інтерактивних навчальних елементів.

Після аналізу доступних рішень було обрано такі основні технології, серед яких Drupal як backend-платформу з урахуванням необхідності побудови складної системи. Drupal – це потужна система управління контентом (CMS), яка дозволяє створювати складні вебдодатки з великою кількістю

структурованих даних, ролей і типів користувачів [16]. Його архітектура дозволяє використовувати Drupal не тільки як класичну CMS, а й як headless-систему – тобто бекенд, який лише надає API для обміну даними з фронтендом. Після аналізу альтернатив (WordPress, Strapi, Joomla) було обрано Drupal через такі переваги:

– розвинена система типів контенту, що дозволяє моделювати різноманітні сутності освітньої системи (курси, уроки, тести, сертифікати);

– гнучка система ролей та прав доступу для диференціації можливостей студентів, викладачів та адміністраторів;

– модуль JSON, який автоматично генерує RESTful API-ендпоінти для безпечної взаємодії з React-інтерфейсом [17];

– вбудовані механізми кешування та оптимізації для забезпечення стабільної роботи під навантаженням.

Альтернативи, які розглядалися:

– WordPress – також потужний CMS, але менш гнучкий у налаштуванні складних типів контенту та прав доступу. Його можливості в побудові кастомних API обмежені;

– Strapi – headless CMS, який забезпечує гнучкість, однак не має такої розвиненої екосистеми і менше підтримується для великих проектів порівняно з Drupal;

– Joomla також є популярною CMS, однак вона має складну структуру і не має такої інтеграції з сучасними фронтенд-технологіями, як Drupal.

З огляду на складність і вимоги до платформи, Drupal був вибраний як найбільш підходящий інструмент для задоволення всіх технічних потреб.

Для реалізації клієнтської частини було обрано React.js – це популярна JavaScript-бібліотека для побудови інтерфейсів користувача, розроблена компанією Meta [18]. React дозволяє створювати веб-застосунки на основі компонентної архітектури, де кожен елемент інтерфейсу – незалежний блок із власною логікою. React працює з віртуальним DOM, що забезпечує високу продуктивність навіть при частих оновленнях інтерфейсу. Цей підхід особливо

ефективний у випадках, коли інтерфейс динамічний, адаптивний і взаємодіє з великою кількістю даних у реальному часі. У порівнянні з Angular та Vue.js, React було вибрано завдяки таким перевагам:

– компонентно-орієнтована структура дозволяє розробити ієрархічну модель UI, де атомарні компоненти (кнопки, поля вводу) формують базовий рівень інтерфейсу. Композитні компоненти (картки курсів, навігаційні панелі) об’єднують атомарні елементи; контейнерні компоненти управляють даними та станом, використовуючи хуки (useState, useEffect, useContext);

– концепція JSX дозволяє декларативно описувати інтерфейс, поєднуючи HTML-структуру з логікою компонентів, що підвищує читабельність коду та продуктивність розробки;

– односпрямований потік даних (Flux-архітектура) забезпечує передбачуваність поведінки інтерфейсу та полегшує відлагодження, що критично важливо при розробці складних форм для створення навчальних матеріалів.

Альтернативи:

– Angular фреймворк, який також підтримує компонентний підхід, але важчий у порівнянні з React через велику кількість вбудованих функцій та складнішу настройку;

– Vue.js ще один фреймворк, схожий на React за компонентною структурою, однак менш популярний серед великих компаній і має менше підтримки при інтеграції з іншими технологіями.

Після порівняння всіх трьох технологій, React був вибраний через свою легкість у використанні, продуктивність і можливості для масштабування інтерфейсу.

Для реалізації системи збереження та обробки даних було обрано MySQL – це одна з найпоширеніших реляційних систем управління базами даних (СУБД) [19]. Вона забезпечує надійне зберігання структурованих даних, підтримує транзакції, індексацію, обмеження цілісності та масштабування. MySQL є відкритим програмним забезпеченням, активно підтримується

спільнотою та має відмінну інтеграцію з більшістю сучасних вебфреймворків. Основні переваги MySQL для проекту:

– висока швидкість обробки запитів, що критично при одночасній роботі багатьох користувачів;

– надійна підтримка транзакцій для забезпечення цілісності даних при збереженні результатів навчання;

– ефективна система індексування для швидкого пошуку та фільтрації курсів і навчальних матеріалів;

– масштабованість та можливість оптимізації при зростанні бази користувачів.

Альтернативи, які розглядалися:

– PostgreSQL ще одна популярна реляційна СУБД, яка підтримує більше функцій для складних запитів, але може бути важчою для налаштування та менш швидкою при великій кількості простих запитів;

– MongoDB нереляційна СУБД, яка краще підходить для NoSQL-даних, але менш ефективна для обробки транзакцій та структурованих даних, що потребують складних зв’язків.

MySQL була обрана через її високу продуктивність та сумісність із Drupal, а також її здатність ефективно працювати з великими обсягами даних.

Обрана архітектура побудована за принципом headless CMS, що дозволяє чітко розмежувати обробку даних (на стороні Drupal) та відображення (на стороні React) [20]. Це дає такі переваги:

– гнучкість у розробці інтерфейсу без прив’язки до специфічної системи управління контентом;

– можливість багато платформеного розширення (мобільні застосунки, окремі SPA для різних ролей);

– краща безпека, оскільки frontend не має прямого доступу до бази даних;

– швидка розробка та тестування, завдяки незалежності частин системи.

У цьому розділі було сформовано архітектурне бачення платформи, визначено функціональність, яку вона повинна реалізовувати, та обґрунтовано

вибір інструментів і технологій. Обрана структура дозволяє створити надійну, зручну та масштабовану систему онлайн-навчання з широкими можливостями для студентів, викладачів і адміністраторів. Застосовані підходи та інструменти відповідають сучасним вимогам до веб розробки та забезпечують ефективну реалізацію поставлених завдань.

Висновки до розділу 2

У другому розділі було проведено детальний аналіз вимог до розроблюваної системи електронного навчання та обґрунтовано вибір технологічного стеку. На основі дослідження функціональних потреб спроектовано архітектуру платформи з використанням UML-діаграм. Діаграма класів дозволила визначити ключові сутності системи та їх взаємозв’язки, а діаграма прецедентів наочно відобразила сценарії взаємодії користувачів з платформою.

На основі аналізу вимог розроблено архітектуру платформи з використанням Drupal (бекенд), React.js (фронтенд) та MySQL (СУБД), що забезпечує продуктивність, безпеку та масштабованість. Запропонований headless-підхід розділяє бекенд і фронтенд, підвищуючи безпеку та спрощуючи розробку. Вибір технологій підтверджено порівняльним аналізом, що гарантує ефективність рішення для освітньої платформи, зокрема в умовах обмежених технічних ресурсів.

РОЗДІЛ 3
РОЗРОБКА ПЛАТФОРМИ ОНЛАЙН-НАВЧАННЯ

3.1. Практична реалізація об’єкта проектування

Платформа онлайн-навчання реалізована за принципом розділення клієнтської та серверної логіки, де бекенд працює на CMS Drupal 10, а фронтенд на React (версія 18.2.0). Для розробки фронтенду використовується сучасний інструмент збірки Vite [21]. Він значно пришвидшує процес розробки завдяки мінімальній початковій компіляції та підтримці гарячого перезапуску компонентів (HMR) [22]. Це особливо зручно при частому тестуванні змін інтерфейсу, адже дозволяє миттєво бачити оновлення без повного перезавантаження сторінки. У лістингу 3.1 наведено приклад файлу vite.config.js, в якому здійснюється базове налаштування плагінів і поведінки збірки для React-проекту. Цей файл конфігурації імпортує основні плагіни React, що забезпечує роботу з JSX і миттєве оновлення UI, а також ESLint і Stylelint – для автоматичної перевірки та корекції JavaScript і стилів, що допомагає підтримувати код у чистому та стандартизованому вигляді без додаткових зусиль.

Лістинг 3.1 – Код конфігурацій для Vite для налаштування плагінів та поведінки Vite під час збірки

```
import { defineConfig } from 'vite';
import react from '@vitejs/plugin-react';
import eslint from 'vite-plugin-eslint';
import stylelint from 'vite-plugin-stylelint';

// https://vitejs.dev/config/
export default defineConfig({
 plugins: [react(), eslint({ fix: true }), stylelint({ fix: true })],
});
```

кінець лістингу 3.1

Фронтенд застосунок побудований на основі компонентної архітектури з чітким розділенням логіки та візуального представлення. Структура проекту організована за принципом модульності та повторного використання коду (рис. 3.1).

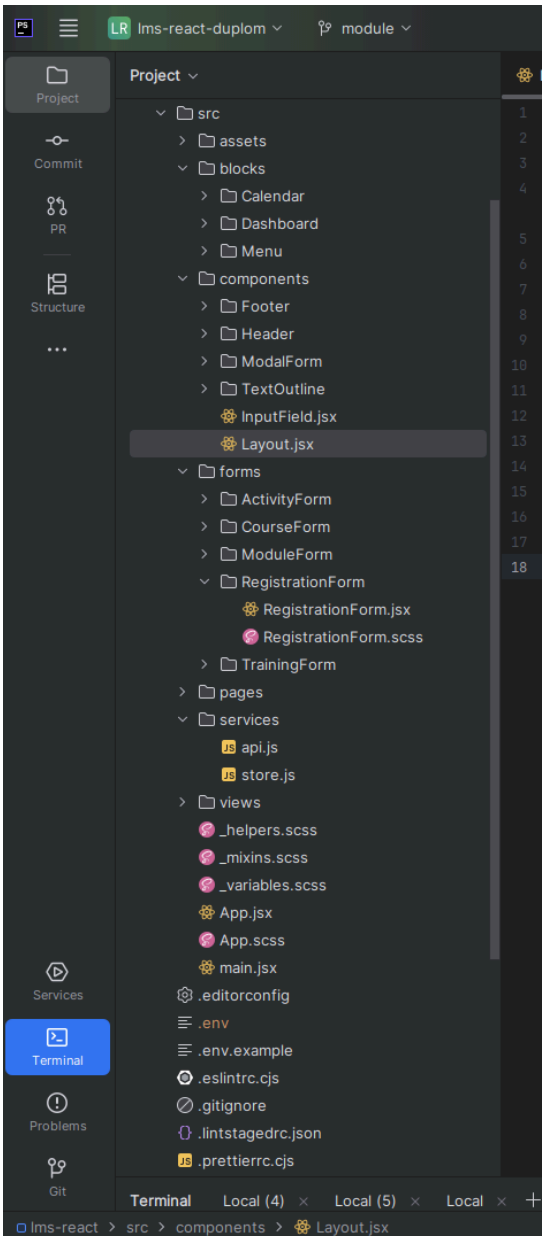


Рисунок 3.1 – Структура React проекту

Основні директорії:

– blocks – містить функціональні блоки інтерфейсу, як-от Calendar, Dashboard, Menu;

– components – багаторазові атомарні компоненти, зокрема Footer, ModalForm, TextOutline, InputField, а також загальний шаблон компонування Layout.jsx;

– forms – папка для виведення форм створення окремих сутностей;

– pages – сторінки, що відповідають за маршрути на сайті HomePage, LoginPage, RegisterPage, TrainingsFullPage, ProfilePage та інші;

– services – службові модулі для взаємодії з API (api.js) та глобальне управління станом через Redux (store.js);

– views – компоненти, що відображають списки,як-от список тренінгів.

У React не було використано axios або fetch напряму. Замість цього робота з API відбувалась через Redux Toolkit Query [23]. Було створено файл api.jsx, де сконфігуровано всі необхідні ендпоінти бекенду, зокрема для отримання й надсилання форми тренінгу. У додатку А наведено приклад цього файлу. У компонентах застосовано згенеровані хуки, такі як useCreateTrainingMutation – приклад використання наведений у додатку Б, це код першого кроку форми створення тренінгів. Це дозволило уникнути дублювання коду та забезпечити автоматичну обробку помилок, станів завантаження і відправки.

Кожна сторінка складається з окремих блоків та форм, що дозволяє легко масштабувати проект і адаптувати інтерфейс до нових функціональних вимог. Основу сторінок становить компонент Layout.jsx, який відповідає за загальну структуру застосунку – він огортає контент основними елементами інтерфейсу, такими як хедер (Header), футер (Footer) та головна область сторінки. Реалізація цього компонента наведена у лістингу 3.2. Layout приймає дочірні елементи як children, що дозволяє повторно використовувати його для різних сторінок без дублювання верстки. Такий підхід забезпечує єдиний каркас інтерфейсу й консистентність в усьому застосунку. Компоненти загалом реалізовані як функціональні, з використанням хуків React (useState, useEffect, useContext тощо), що сприяє ефективному керуванню локальним і глобальним станом.

Лістинг 3.2 – Код основного компонента Layout.jsx

```
import PropTypes from 'prop-types';
import Header from './Header/Header.jsx';
import Footer from './Footer/Footer.jsx';

export default function Layout({children}) {
  return (
    <>
      <Header/>
      <main className="container">{children}</main>
      <Footer/>
    </>
  );
}

Layout.propTypes = {
  children: PropTypes.node.isRequired,
};
```

кінець лістингу 3.2

Однією з базових сторінок застосунку є сторінка авторизації (рис. 3.2), реалізована в компоненті LoginPage.jsx, що розміщений у директорії pages.

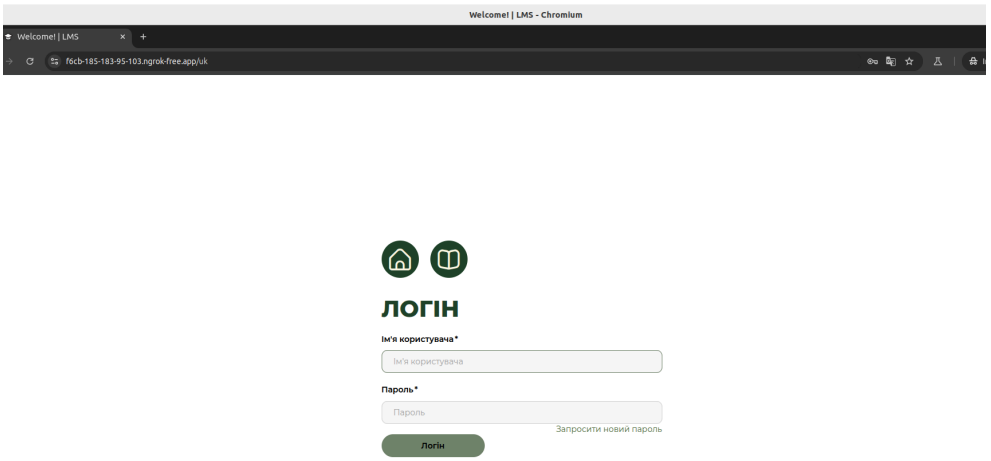


Рисунок 3.2 – Інтерфейс сторінки логінації

Її завдання – забезпечити зручний та інтуїтивно зрозумілий інтерфейс для входу користувача до системи. Компонент LoginPage побудований із використанням React-хуків useState та useMutation з бібліотеки

@reduxjs/toolkit/query, що відповідає за обробку запитів до API. Завдяки цій інтеграції здійснюється надсилання логін-даних до відповідного ендпоінту бекенду на Drupal 10. При успішному запиті зберігається токен авторизації, який у подальшому використовується для доступу до захищених маршрутів системи. Реалізація компонента LoginPage.jsx наведена в додатку В.

Серверна частина платформи реалізована на основі CMS Drupal 10, з використанням концепції Headless Drupal. Це означає, що Drupal відповідає лише за збереження, обробку та видачу даних через API, не опікуючись рендерингом інтерфейсу. У такій архітектурі фронтенд (React) та бекенд (Drupal) функціонують незалежно, що забезпечує високу гнучкість, масштабованість та можливість повторного використання API у сторонніх системах. Для реалізації взаємодії з фронтендом використано стандартний модуль RESTful Web Services, який спрощує налаштування ендпоінтів через інтерфейс адміністратора. Основними типами доступу до даних є [24]:

– GET для отримання ресурсів;

– POST для створення нових сутностей (наприклад, тренінгів);

– PATCH для оновлення наявних сутностей;

– DELETE для їх видалення.

Для автентифікації REST-запитів використовується модуль Simple OAuth, який забезпечує обмін токеном доступу для авторизованих запитів [25]. Фронтенд, зберігаючи токен після логіна, може здійснювати захищені запити до Drupal API. У лістингу 3.3 наведений приклад отримання списку ентіті тренінгів для відображення на сайті (рис. 3.3).

Лістинг 3.3 – API відповідь зі списком тренінгів

```
{
  "filters": {
    "search": "",
      "status": ["Завершений", "У процесі", "Розпочато", "Зданий", "Не
вийшло"],
    "category": null
  },
  "trainings": [
```

```
    {
      "id": 37,
      "title": "Професійна етика та комунікація",
      "category": "Командна робота та співпраця",
          "description": "Формування етичних норм поведінки на роботі, розвиток навичок ефективної комунікації в команді.",
      "status": "Завершений",
      "image_url": "/sites/default/large/ethics_communication.png",
      "resultAvailable": true,
      "actions": [
        {
          "label": "Переглянути результат",
          "type": "link",
          "url": "/trainings/37/result"
        }
      ]
    }
  ]
}
```


кінець лістингу 3.3

На основі отриманої відповіді формується інтерфейс списку тренінгів на сайті, де кожна сутність виводиться у вигляді окремого блоку з відповідною інформацією.

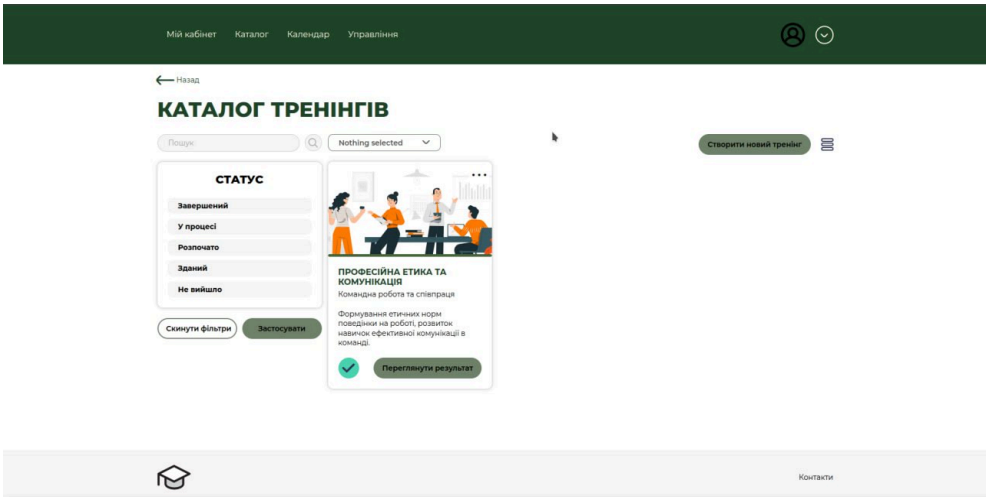


Рисунок 3.3 – Інтерфейс списку тренінгів на сайті

Бекенд платформи реалізовано на мові програмування PHP версії 8.1.32 відповідно до стандартів Drupal API. Архітектура коду побудована на

об’єктно-орієнтованому підході, що дозволяє забезпечити модульність, повторне використання компонентів та легкість у супроводі. Основними інструментами розробки виступають базові класи Drupal, зокрема ControllerBase для побудови контролерів, FormBase для створення форм, а також механізм кастомних сутностей (Entity API), що дозволяє зручно моделювати та керувати доменними об’єктами системи [26]. Загальна структура бекенду організована у вигляді кількох окремих модулів (рис. 3.4), кожен із яких відповідає за певну частину логіки платформи:

– my_custom_learning_path – модуль для створення та керування тренінгами;

– my_custom_course – модуль для управління навчальними курсами;

– my_custom_modules – модуль для управління навчальними модулями;

– my_custom_activity – модуль для управління вправами.

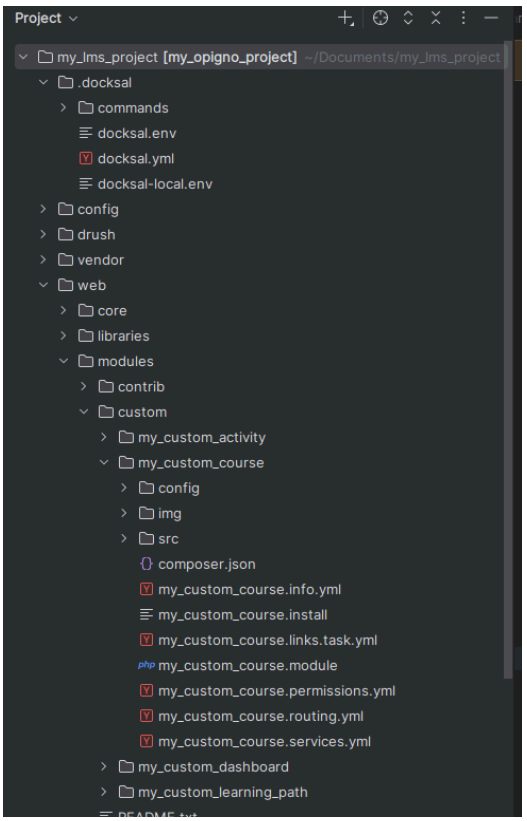


Рисунок 3.4 – Структура проекта на Drupal

Для кожної з ключових сутностей – тренінгів, курсів, модулів і вправ були створені кастомні Entity-класи, які реалізують повний CRUD-функціонал та підтримують серіалізацію у форматах JSON/REST. Завдяки цьому як бекенд, так і фронтенд можуть уніфіковано взаємодіяти з даними через REST API.

Модуль my_custom_activity є кастомним Drupal-модулем, що реалізує функціонал для створення, збереження та виведення вправ (activity) у рамках навчальної платформи. Нижче подано опис основних файлів та директорій модуля:

– ActivityTypes/ – містить типи активностей та їхню специфічну логіку;

– config/ – зберігає експортовані конфігурації в форматі YAML (наприклад, типи сутностей, форми, дозволи тощо);

– h5plib/ – містить бібліотеки для інтеграції H5P-контенту;

– json-schema/ – JSON-схеми для валідації даних;

– src/ – основна PHP-логіка модуля, включаючи класи сутностей, контролери, сервіси;

– my_custom_activity.info.yml – описує мета-інформацію про модуль (назва, залежності, версія);

– my_custom_activity.install – містить хук hook_install() для створення необхідних таблиць або налаштувань при встановленні/видаленні модуля;

– my_custom_activity.routing.yml – маршрутизація для API-запитів типу GET/POST/PATCH/DELETE;

– my_custom_activity.permissions.yml – доступ до операцій редагування, перегляду та видалення вправ;

– my_custom_activity.services.yml – допоміжні сервіси для обробки логіки.

У додатку Г представлено уривок коду з файлу LMSActivity.php, у якому реалізується повноцінна кастомна сутність для вправ (activity), що є частиною навчального процесу.

Реалізація онлайн-платформи навчання засвідчила ефективність використання сучасного технологічного стеку – React і Vite для фронтенду та Drupal 10 у ролі headless-бекенду. Завдяки компонентній архітектурі,

використанню Redux Toolkit Query для взаємодії з API, а також модульній організації коду вдалося досягти високої зручності в розробці. Розділення логіки клієнтської та серверної частин забезпечує гнучкість системи та полегшує її подальший розвиток.

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

У процесі розробки онлайн-платформи для навчання особливу увагу було приділено етапу тестування та налагодження системи. Це дозволило виявити і усунути помилки на ранніх етапах розробки, підвищити стабільність роботи та забезпечити якісну взаємодію між клієнтською та серверною частинами.

Модульне тестування проводилося для перевірки окремих компонентів, реалізованих на React. Зокрема, тестуванню підлягали сторінки входу, реєстрації, перегляду курсів, що дозволило впевнитися в коректності реалізації їхньої логіки та відсутності критичних помилок у поведінці інтерфейсу.

Інтеграційне тестування здійснювалося з метою перевірки коректної взаємодії між клієнтською частиною додатку та серверною логікою, реалізованою в Drupal 10. Особливу увагу приділено перевірці роботи з Redux RTK Query – важливого інструмента для обміну даними між компонентами та сервером. Було протестовано обробку користувацьких даних, курсів, а також функціональність, пов'язану з розкладом та іншими API-запитами.

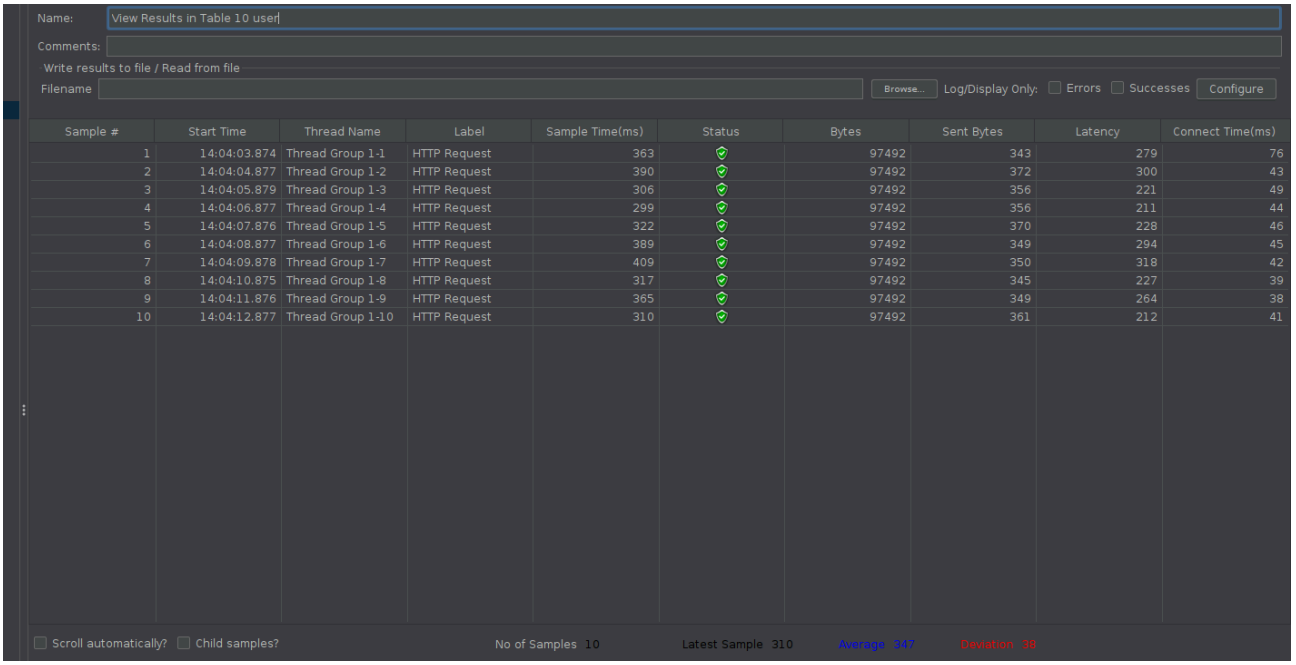
Навантажувальне тестування для перевірки продуктивності системи було проведено серію навантажувальних тестів з використанням JMeter. Особливістю підходу стало застосування плагіна «Random CSV Data Set Config», який дозволив імітувати різноманітні сценарії роботи користувачів шляхом випадкового вибору тестових даних з підготовлених наборів. Тестова конфігурація включала три етапи перевірки:

– тестування з групою 10 користувачів показало середній час відповіді 120 мс;

– при збільшенні навантаження до 30 одночасних користувачів час відгуку зріс до 210 мс;

– максимальне навантаження у 50 користувачів демонструє стабільний час відповіді на рівні 290 мс.

Як видно з наведених скріншотів, система демонструє передбачувану динаміку зростання часу відгуку при збільшенні навантаження (рис. 3.5).



Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	14:04:03.874	Thread Group 1-1	HTTP Request	363		97492	343	279	76
2	14:04:04.877	Thread Group 1-2	HTTP Request	390		97492	372	300	43
3	14:04:05.879	Thread Group 1-3	HTTP Request	306		97492	356	221	49
4	14:04:06.877	Thread Group 1-4	HTTP Request	299		97492	356	211	44
5	14:04:07.876	Thread Group 1-5	HTTP Request	322		97492	370	228	46
6	14:04:08.877	Thread Group 1-6	HTTP Request	389		97492	349	294	45
7	14:04:09.878	Thread Group 1-7	HTTP Request	409		97492	350	318	42
8	14:04:10.875	Thread Group 1-8	HTTP Request	317		97492	345	227	39
9	14:04:11.876	Thread Group 1-9	HTTP Request	365		97492	349	264	38
10	14:04:12.877	Thread Group 1-10	HTTP Request	310		97492	361	212	41

Рисунок 3.5 – Тестування на групу з 10 користувачів

Особливо варто відзначити результати з 50 користувачами (рис. 3.6), де система зберегла стабільність показників. Це підтверджує ефективність обраної архітектури та оптимізацію процесів.

Функціональне тестування включає перевірку основних сценаріїв взаємодії користувача із системою: авторизація, перегляд курсів, запис на курс, проходження уроків тощо. Таблицю з тест-кейсами наведено у додатку Д.

Тестування проводилось вручну, за допомогою браузерних DevTools та інструментів консолі для виявлення JavaScript-помилок. У процесі роботи також активно використовувалися системи логування у фронтенді та бекенді для відстеження помилок HTTP-запитів.

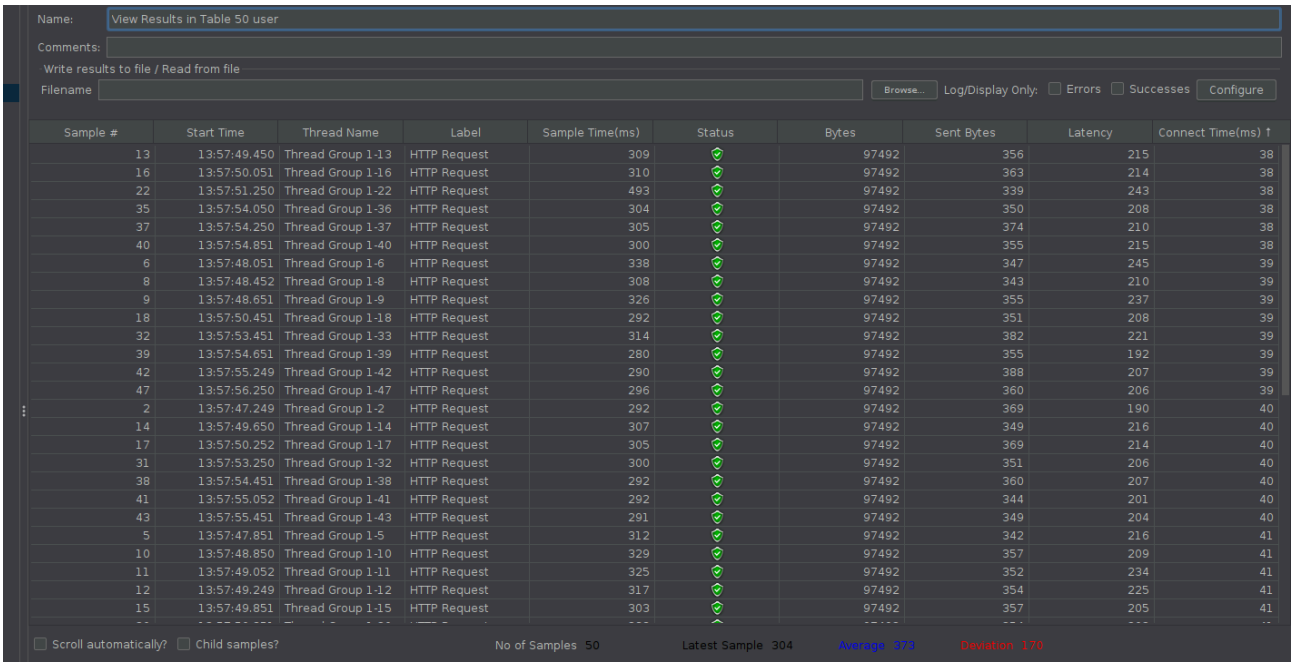


Рисунок 3.6 – Тестування на групу з 50 користувачів

У процесі налагодження було усунуто низку проблем:

– некоректна обробка помилок при неправильному введенні логіна/пароля;

– виправлення помилки відображення компонентів при оновленні сторінки без втрати стану (використання persistedReducer);

– усунення проблеми з некоректним рендерингом адаптивної сітки на мобільних пристроях;

– налагодження авторизації та збереження токенів за допомогою localStorage.

3.3 Розробка бази даних

У процесі створення онлайн-платформи для навчання особливу увагу було приділено проектуванню та реалізації бази даних, яка забезпечує зберігання, обробку й ефективну взаємодію між основними сутностями системи. У проекті використовується MariaDB версії 10.11.11 – сучасний, сумісний із MySQL рушій, що забезпечує високу продуктивність, стабільність і надійність. Для

перегляду та ручної роботи з базою даних використовується phpMyAdmin,який має зручний вебінтерфейс для адміністрування MySQL/MariaDB (рис. 3.7).

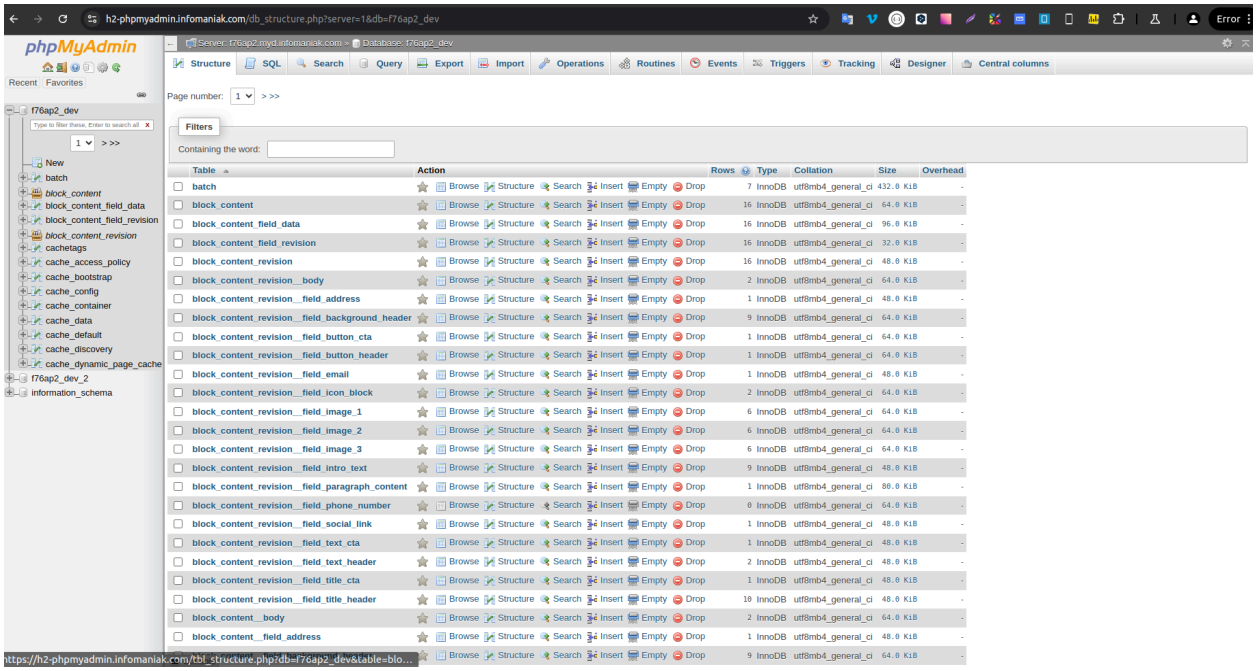


Рисунок 3.7 – Перегляд таблиць з проекту у phpMyAdmin

У Drupal 10 (як і у Drupal 8/9) дуже багато речей зберігаються у вигляді конфігурацій, описаних у YAML-файлах, приклад файлу наведено у лістингу 3.4. Це декларативний спосіб описати:

– типи сутностей (Entity types) – наприклад, користувачі, курси, тренінги;

– поля сутностей (Fields) – текстові поля, дати, числа, посилання, довільні дані;

– форми введення (Form displays) – які поля, в якому порядку і як відображаються у формах;

– перегляди (View displays) – як дані відображаються у списках, сторінках;

– правила доступу (Permissions) – хто що може робити з певними сутностями.

Лістинг 3.4 – Конфігураційний файл до поля опису курсу

```
langcode: ukr
status: true
dependencies:
  module:
    - group
    - text
id: group.field_course_description
field_name: field_course_description
entity_type: group
type: text_long
settings: {  }
module: text
locked: false
cardinality: 1
translatable: true
indexes: {  }
persist_with_no_fields: false
custom_storage: false
```

кінець лістингу 3.4

Конфігураційні файли YAML покривають більшість потреб щодо опису структури даних, але іноді потрібно створити спеціальні таблиці або виконати додаткові дії на рівні БД, які Drupal не підтримує «з коробки» через Entity API.

Ось тут використовуються .install файли, які є частиною кастомного модуля і містять PHP-код із функціями, які викликаються під час:

– встановлення модуля (hook_install());

– оновлення модуля (hook_update_N());

– видалення модуля (hook_uninstall()).

У цих функціях можна:

– створювати таблиці вручну через DB API Drupal (db_create_table(), Schema API);

– робити складні маніпуляції з даними;

– виконувати міграції або оновлення структури бази даних.

Приклад функції hook_schema() файлу my_custom_modules.install з мого власного модуля наведений у додатку Е.

Загалом, ретельно спроектована та гнучко реалізована база даних є фундаментом ефективної роботи онлайн-платформи. Поєднання можливостей

системи управління базами даних MariaDB, гнучкості YAML-конфігурацій Drupal та розширюваності через .install-файли дає змогу не лише чітко структурувати інформацію, а й адаптувати систему до специфічних вимог освітнього процесу. Такий підхід забезпечує як стабільність роботи, так і готовність до масштабування та подальшого розвитку платформи.

3.4 Захист інформаційно-комп’ютерної системи

У платформі особливу увагу приділяємо безпеці, оскільки тут зберігаються персональні дані студентів, їхні оцінки, історія проходження курсів та інші важливі відомості. Наприклад, ми налаштували регулярне резервне копіювання бази даних Drupal, а також інтегрували систему логування безпеки. Завдяки цьому, ми отримуємо дії всіх користувачів (рис. 3.8).

user	05/19/2025 - 18:39	Session closed for student1.	Student 1 Surname
user	05/19/2025 - 18:36	Session opened for student1.	Student 1 Surname
user	05/19/2025 - 18:35	Session closed for sonya.	Sonya Bodun
user	05/19/2025 - 10:26	Session opened for sonya.	Sonya Bodun
group	05/19/2025 - 10:10	The group Мій перший тренінг has been deleted.	Sonya Bodun
group	05/19/2025 - 10:10	The group Мій другий тренінг has been deleted.	Sonya Bodun

Рисунок 3.8 – Логування активностей на сайті

В нашій системі ми встановили і налаштували модуль Simple OAuth для Drupal. Після того, як користувач проходить процедуру входу, сервер генерує токен доступу. Цей токен фронтенд, написаний на React, зберігає у локальному сховищі браузера. Наприклад, коли користувач запитує список своїх курсів через REST API, токен автоматично додається до заголовку Authorization: Bearer <token>. Сервер перевіряє цей токен перед тим, як надати доступ до даних. Якщо токен є недійсним, відсутнім або його термін дії минув, сервер ідентифікує запит як такий, що надійшов від неавторизованого користувача, і автоматично відхиляє його з кодом помилки 401 (Unauthorized).

Паролі користувачів у Drupal в базі зберігаються у вигляді хешів (хешованих значень) із застосуванням алгоритму bcrypt. Це означає, що реальний пароль не зберігається, а лише його захищена хеш-версія, що унеможливлює відновлення оригінального пароля навіть у разі доступу до бази даних.

Таким чином, комплекс цих заходів допомагає забезпечити надійний захист платформи і зробити її безпечною для користувачів.

3.5 Розробка документації для програмного продукту

Документація є невід'ємною частиною будь-якого програмного продукту, оскільки вона допомагає користувачам та адміністраторам ефективно працювати з системою, а також полегшує подальше обслуговування і розвиток проекту.

Для стабільної та коректної роботи розробленої онлайн-платформи для навчання рекомендовано використовувати сервер із наступними параметрами:

– операційна система: Linux (рекомендовано Ubuntu 20.04 або новіша версія);

– веб-сервер: Apache 2.4+ або Nginx 1.18+;

– PHP: версія 8.1 або вище, з увімкненими розширеннями, необхідними для Drupal 10;

– база даних: MariaDB 10.3+ або MySQL 5.7+;

– пам'ять: мінімум 2 ГБ ОЗП, рекомендовано 4 ГБ і більше;

– інтернет-з'єднання: стабільне, з підтримкою HTTPS.

Такі параметри забезпечують достатній рівень продуктивності та безпеки при роботі з платформою. Встановлення програмного продукту передбачає кілька основних кроків:

– підготовка серверного оточення: інсталяція операційної системи, веб-сервера, PHP, бази даних;

– завантаження та розпакування коду платформи на сервері;

– налаштування підключення до бази даних через конфігураційний файл Drupal (settings.php);

– встановлення Drupal та імпорт початкових налаштувань за допомогою інсталяційного скрипту наведений у лістингу 3.5.

Лістинг 3.5 – Інсталяційний скрипт для розгортання Drupal проекта

```
#!/bin/bash
DB_NAME="f76ap2_dev"
DB_USER="f76ap2_dev"
DB_PASS="P7m1HyOb g70C!$"
DB_HOST="f76ap2.myd.infomaniak.com"
PROJECT_DIR="/var/www/my-lms-project"
DB_DUMP="/var/www/drupal-project/db/db_dump.sql"
echo "Крок 1: Переходимо в директорію проекту"
cd $PROJECT_DIR || { echo "Не вдалося перейти в директорію $PROJECT_DIR"; exit 1; }
echo "Крок 2: Встановлення залежностей через composer"
composer install || { echo "Помилка встановлення залежностей"; exit 1; }
echo "Крок 3: Імпорт бази даних"
mysql -u $DB_USER -p$DB_PASS $DB_NAME < $DB_DUMP || { echo "Помилка імпорту бази даних"; exit 1; }
echo "Крок 4: Імпорт конфігурації Drupal"
drush config-import --yes || { echo "Помилка імпорту конфігурацій"; exit 1; }
echo "Крок 5: Очищення кешу"
drush cache-rebuild || { echo "Помилка очищення кешу"; exit 1; }
```

кінець лістингу 3.5

Для запуску фронтенд частини необхідно запустити інсталяційний скрипт для швидкого запуску React проекта, приклад мого скрипту наведено у лістингу 3.6.

Лістинг 3.6 – Скрипт для запуску React проекту

```
#!/bin/bash
echo "Активуємо Node.js версії 18 через nvm"
nvm use v18 || { echo "Node.js версія 18 не встановлена. Встановіть її через nvm"; exit 1; }

PROJECT_DIR="/var/www/lms-react"
```

```
cd $PROJECT_DIR || { echo "Не вдалося перейти в директорію $PROJECT_DIR"; exit 1; }

echo "Встановлюємо залежності через npm"
npm install || { echo "Помилка встановлення npm залежностей"; exit 1; }

npm run build
```

кінець лістингу 3.6

У результаті практичної реалізації платформи онлайн-навчання було успішно поєднано сучасні технології фронтенду React з надійним та гнучким бекендом на основі Drupal 10 у режимі headless CMS. Використання компонентної архітектури, Redux Toolkit Query для роботи з API та модульної структури як на фронтенді, так і на бекенді забезпечило високу масштабованість, зручність підтримки та розвиток системи. Розділення клієнтської і серверної логіки дозволяє ефективно адаптувати платформу під подальші функціональні вимоги та інтеграції, що підтверджує обраний підхід як ефективний і перспективний для створення сучасних веб-застосунків у сфері онлайн-освіти.

Висновки до розділу 3

У третьому розділі було реалізовано платформу онлайн-навчання з використанням сучасного технологічного стеку, фронтенд на React з Vite для швидкої розробки, бекенд на Drupal 10 у headless-режимі та база даних MySQL. Архітектура системи передбачає чітке розділення клієнтської та серверної частин, що забезпечує гнучкість і масштабованість. Для взаємодії між компонентами використано Redux Toolkit Query, що спростило роботу з API та обробку станів.

Тестування включало модульну, інтеграційну та функціональну перевірки, що дозволило виявити та усунути критичні помилки. Система безпеки реалізована через OAuth2 автентифікацію, хешування паролів та логування подій. Документація містить вимоги до серверного середовища та інструкції з

розгортання. Отримані результати підтверджують ефективність обраного підходу для створення сучасних освітніх платформ.

ВИСНОВКИ

У межах даної кваліфікаційної роботи мною було розроблено сучасний інтерактивний інтерфейс з використанням React, який забезпечує зручну навігацію, реєстрацію користувачів, перегляд курсів, проходження тестів та отримання результатів. Інтуїтивно зрозумілий UI/UX дизайн значно підвищив комфорт роботи з платформою для всіх категорій користувачів.

Для ефективного управління навчальним контентом було інтегровано систему Drupal, що надало адміністраторам зручні інструменти для публікації та редагування матеріалів, а також контроль доступу до різних розділів платформи. Це значно спростило процес адміністрування навчальних курсів.

Реалізовано надійну систему реєстрації та авторизації з використанням MySQL, яка забезпечує безпечне зберігання даних користувачів за допомогою сучасних методів шифрування. Система також передбачає чіткий розподіл прав доступу між студентами, викладачами та адміністраторами.

Для підвищення мотивації студентів було впроваджено систему гейміфікації, що включає механізми нарахування балів за виконання завдань з курсів. Оцінки реалізовані, як поле в кастомному модулі my_custom_activity в Друпалі, який відстежує максимальний бал за завдання та вираховує кількість отриманих балів. Завдяки цьому створено систему оцінювання, яка мотивує студентів бути активнішими.

Платформа була спеціально оптимізована для роботи на різних пристроях, включаючи застарілі комп’ютери та мобільні телефони. Це забезпечило доступність навчання навіть за умов обмежених технічних можливостей користувачів.

Розроблено ефективну систему оцінювання, яка включає автоматичну перевірку тестів, можливість оцінки виконаних завдань викладачами. Це дозволило об’єктивно оцінювати рівень знань студентів.

Після розробки було проведено тестування, яке включало як функціональну, так і навантажувальну перевірку платформи. Функціональне

тестування охоплювало основні сценарії взаємодії користувача з платформою, зокрема авторизацію, створення курсів. Це дозволило впевнитись в коректності роботи інтерфейсу та логіки платформи. Навантажувальне тестування проводилося за допомогою JMeter і підтвердило стабільність роботи системи за умов значного навантаження. Результати тестування засвідчили стабільну роботу платформи за умов поступового збільшення навантаження при одночасному підключенні 10 користувачів середній час відповіді становив 120 мс, при 30 – зростав до 210 мс, а за максимального навантаження у 50 користувачів не перевищував 290 мс. Така передбачувана і контрольована динаміка свідчить про ефективну побудову платформи та впровадження кешування на рівні API-запитів.

Результатом моєї роботи стала повноцінна платформа електронного навчання, що відповідає сучасним вимогам та враховує специфіку української освіти в кризових умовах. Розроблений продукт готовий до впровадження в навчальних закладах та має значний потенціал для подальшого розвитку. Усі поставлені завдання виконано, що підтверджує досягнення мети дослідження.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Освіта в умовах воєнного стану. Виклики, розвиток, повоєнні перспективи. URL: https://mon.gov.ua/static-objects/mon/sites/ (дата звернення: 26.02.2025 р.).

2. Moodle до, під час і після пандемії COVID-19: використання студентами бакалаврату та магістратури. URL: https://openedu.kubg.edu.ua/journal/index.php/openedu/article/ (дата звернення: 26.02.2025 р.).

3. ЛНТУ. URL: https://lntu.edu.ua/uk (дата звернення: 26.02.2025 р.).

4. Дистанційне навчання в Україні за допомогою платформи Open edX. URL: https://openedx.org/uk/blog/a-success-story-for-distance-learning (дата звернення: 27.02.2025 р.).

5. Сайт Open edX. URL: https://openedx.org/ (дата звернення: 26.02.2025 р.).

6. Перелік платформ для вдосконалення своїх навичок і для саморозвитку. URL: https://nung.edu.ua/event/perelik-platform (дата звернення: 27.02.2025 р.).

7. Використання навчальних платформ для неформальної освіти. URL: https://content.hneu.edu.ua/s/2HyNYjAiy (дата звернення: 28.02.2025 р.).

8. Цифрові платформи у вищій освіті. URL: https://mon.gov.ua/osvita-2/tsifrova-transformatsiya-osviti (дата звернення: 28.02.2025 р.).

9. Сайт Udemy. URL: https://www.udemy.com/ (дата звернення: 28.02.2025 р.).

10. Онлайн-навчання на платформі «Prometheus». Нові реалії української вищої освіти. URL: https://mala.storinka.org (дата звернення: 28.02.2025 р.).

11. Сайт Prometheus. URL: https://prometheus.org.ua/ (дата звернення: 28.02.2025 р.).

12. Сайт Всеукраїнської онлайн школи .URL:https://lms.e-school.net.ua/ (дата звернення: 28.02.2025 р.).

13. Діаграми UML для моделювання процесів і архітектури. URL: https://evergreens.com.ua/uml-diagrams.html (дата звернення: 10.03.2025 р.).

14. Діаграма класів UML. URL: https://www.mindonmap.com/uk/ (дата звернення: 10.03.2025 р.).

15. Діаграма прецедентів. URL: https://dou.ua/forums/topic/40575/ (дата звернення: 10.03.2025 р.).

16. Документація по Drupal. URL: https://new.drupal.org/drupal-cms (дата звернення: 12.03.2025 р.).

17. Створення API-запитів використовуючи Drupal JSON Endpoints. URL: https://www.drupal.org/docs/core (дата звернення: 12.03.2025 р.).

18. Документація створення сайту за допомогою React. URL: https://legacy.reactjs.org/ (дата звернення: 12.03.2025 р.)

19. Використання MySQL. URL: https://www.drupal.org/docs/contributed (дата звернення: 12.03.2025 р.).

20. Документація по використанню Drupal Headless. URL: https://wishdesk.com/blog/headless (дата звернення: 12.03.2025 р.).

21. How To Set Up a React Project with Vite for Fast Development. URL: https://www.digitalocean.com/community (дата звернення: 05.05.2025 р.).

22. Документація HMR API Vite. URL: https://vite.dev/guide/api-hmr (дата звернення: 05.05.2025 р.).

23. Створення Redux Toolkit Queries у React. URL: https://medium.com/@r.sipchenko/rtk-query (дата звернення: 05.05.2025 р.).

24. Документація з використанням REST API. URL: https://www.drupal.org/docs/drupal-apis/restful (дата звернення: 05.05.2025 р.).

25. Підключення Simple OAuth (OAuth2) та OpenID Connect. URL: https://www.drupal.org/project/simple_oauth (дата звернення: 15.05.2025 р.).

26. Сутності контенту та поля в Drupal. URL: https://www.drupal.org/docs/user_guide/planning (дата звернення: 15.05.2025 р.).

ДОДАТКИ

ДОДАТОК А

Демонстрація коду конфігурації Redux Toolkit Query для взаємодії фронтенду на React із бекендом LMS через REST API

```
import {createApi, fetchBaseQuery} from '@reduxjs/toolkit/query/react';

const apiUrl = `${import.meta.env.VITE_BACKEND_URL}/`;

export const lmsApi = createApi({
  reducerPath: 'lmsApi',
  baseQuery: fetchBaseQuery({
    baseUrl: apiUrl,
  }),

  endpoints: (builder) => ({
    Trainings: builder.query({
      query: () => 'trainings',
    }),

    TrainingsPage: builder.query({
      query: ({page}) => `trainings?page=${page}`,
    }),

    TrainingDetails: builder.query({
      query: ({id}) => `trainings/${id}`,
    }),

    CreateTraining: builder.mutation({
      query: (trainingData) => ({
        url: 'trainings',
        method: 'POST',
        body: trainingData,
      }),
    }),

    UpdateTraining: builder.mutation({
      query: ({id, ...patch}) => ({
        url: `trainings/${id}`,
        method: 'PUT',
        body: patch,
      }),
    }),

    Certificates: builder.query({
      query: () => 'certificates',
    }),

    CreateCertificate: builder.mutation({
```

```
      query: (certificateData) => ({
        url: 'certificates',
        method: 'POST',
        body: certificateData,
      }),
    }),

    Login: builder.mutation({
      query: (credentials) => ({
        url: 'auth/login',
        method: 'POST',
        body: credentials,
      }),
    }),

    RegistrationFormSubmit: builder.mutation({
      query: (formData) => ({
        url: 'registration/submit',
        method: 'POST',
        body: formData,
      }),
    }),
  }),
});

export const {
  useTrainingsQuery,
  useTrainingsPageQuery,
  useTrainingDetailsQuery,
  useCreateTrainingMutation,
  useUpdateTrainingMutation,
  useCertificatesQuery,
  useCreateCertificateMutation,
  useLoginMutation,
  useRegistrationFormSubmitMutation,
} = lmsApi;
```

ДОДАТОК Б

Форма першого кроку створення тренінгу.

```
import { useState } from 'react';
import { useForm, Controller } from 'react-hook-form';
import TextOutline from '../TextOutline/TextOutline';
import InputField from '../InputField';
import './TrainingStepOne.scss';
import { useCreateTrainingMutation } from '../../services/api';

export default function TrainingStepOne({ onNext }) {
  const { control, handleSubmit, setValue, formState: { errors } } =
useForm({
    defaultValues: {
      title: '',
      description: '',
      training_type: 'public',
      category: '',
      certificate_url: '',
      photo: null,
    },
  });

  const [createTraining, { isLoading, isError, error }] =
useCreateTrainingMutation();
  const [submitError, setSubmitError] = useState(null);

  const onSubmit = async (data) => {
    try {
          await createTraining(data).unwrap();
      if(onNext) onNext(data);
    } catch (err) {
      setSubmitError('Error');
      console.error(err);
    }
  };

  return (
    <div className="training-step-one">
      <TextOutline description="Тренінг - це програма, що складається з
різноманітних курсів, модулів та заходів, які супроводжують студентів протягом
усього процесу навчання." />
      <form onSubmit={handleSubmit(onSubmit)}>
        <InputField
          control={control}
          name="title"
          rules={{ required: 'Будь ласка, введіть назву тренінгу' }}
          error={errors.title}
          label="Назва"
```

```
        />

        <Controller
          control={control}
          name="description"
          rules={{ required: 'Вкажіть опис тренінгу' }}
          render={({ field }) => (
            <div className="form-group">
              <label>Опис тренінгу</label>
              <textarea
                {...field}
                rows={4}
                className={errors.description ? 'error' : ''}
              />
              {errors.description && <p
classNamе="error-message">{errors.description.message}</p>}
            </div>
          )}
        />

        <Controller
          control={control}
          name="training_type"
          render={({ field }) => (
            <div className="form-group">
              <label>Тип тренінгу</label>
              <select {...field} defaultValue="public">
                <option value="public">
                  Публічний
                </option>
                <option value="semi_private">
                  Напівприватний
                </option>
                <option value="private">
                  Приватний
                </option>
              </select>
            </div>
          )}
        />

        <InputField
          control={control}
          name="category"
          rules={{ required: 'Оберіть категорію' }}
          error={errors.category}
          label="Категорія"
          type="select"
          options={[
            { label: 'IT', value: 'it' },
            { label: 'Менеджмент', value: 'management' },
```

```
            { label: 'Мови', value: 'languages' },
            { label: 'Інше', value: 'other' },
          ]}
        />

      <Controller
         control={control}
         name="photo"
         render={({ field }) => (
           <div className="form-group">
             <label>Фото</label>
             <input
               type="file"
               accept="image/*"
               onChange={(e) => field.onChange(e.target.files[0])}
             />
           </div>
         )}
        />

  {submitError && <p className="error-message">{submitError}</p>}

       <div className="submit-button">
         <button type="submit" disabled={isLoading}>
           {isLoading ? 'Завантаження...'}
         </button>
       </div>
      </form>
    </div>
  );
}
```

ДОДАТОК В

Реалізація логін сторінки через React

```
import {useState} from 'react';
import {useLoginMutation} from '../../services/api';
import {Link} from 'react-router-dom';
import './LoginPage.scss';
import homeIcon from '../../assets/icons/home.svg';
import bookIcon from '../../assets/icons/book.svg';

export default function LoginPage() {
  const [formData, setFormData] = useState({username: '', password: ''});
  const [login, {isLoading, error}] = useLoginMutation();

  const handleChange = (e) => {
    const {name, value} = e.target;
    setFormData((prev) => ({...prev, [name]: value}));
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      const res = await login(formData).unwrap();
      localStorage.setItem('token', res.token);
      window.location.href = '/';
    } catch (e) {
      console.error('Login failed:', e);
    }
  };

  return (
    <div className="login-wrapper">
      <form className="login-card" onSubmit={handleSubmit}>
        <div className="login-icons">
          <img src={homeIcon} alt="home"/>
          <img src={bookIcon} alt="book"/>
        </div>

        <h2 className="login-title">Логін</h2>

        <label htmlFor="username">Ім'я користувача*</label>
        <input
          type="text"
          id="username"
          name="username"
          placeholder="Ім'я користувача"
          value={formData.username}
          onChange={handleChange}
          required
```

```
        />

        <label htmlFor="password">Пароль*</label>
        <input
          type="password"
          id="password"
          name="password"
          placeholder="Пароль"
          value={formData.password}
          onChange={handleChange}
          required
        />

        <div className="login-links">
          <Link to="/request-password">Запросити новий пароль</Link>
        </div>

        {error && <p className="error-msg">Невірні дані. Спробуйте ще
раз.</p>}

        <button type="submit" disabled={isLoading}>
          {isLoading ? 'Завантаження...' : 'Логін'}
        </button>

        <div className="create-account-link">
          <Link to="/register">Створити новий акаунт</Link>
        </div>
      </form>
    </div>
  );
}
```

ДОДАТОК Г

Демонстрація функції, яка описує структуру базових полів для кастомної сутності “Activity”

```
public static function baseFieldDefinitions(EntityTypeInterface
$entity_type) {
      $fields = parent::baseFieldDefinitions($entity_type);
       $fields['uid'] = BaseFieldDefinition::create('entity_reference')
         ->setLabel(t('Authored by'))
         ->setDescription(t('The user ID of author of the Activity
entity.'))
         ->setRevisionable(TRUE)
         ->setSetting('target_type', 'user')
         ->setSetting('handler', 'default')
         ->setTranslatable(TRUE)
         ->setDisplayOptions('view', [
           'label' => 'hidden',
           'type' => 'author',
           'weight' => 0,
         ])
         ->setDisplayOptions('form', [
           'type' => 'entity_reference_autocomplete',
           'weight' => 5,
           'settings' => [
             'match_operator' => 'CONTAINS',
             'size' => '60',
             'autocomplete_type' => 'tags',
             'placeholder' => '',
           ],
         ])
         ->setDisplayConfigurable('form', TRUE)
         ->setDisplayConfigurable('view', TRUE);


       $fields['name'] = BaseFieldDefinition::create('string')
         ->setLabel(t('Name'))
         ->setDescription(t('The name of the Activity entity plus.'))
         ->setRequired(TRUE)
         ->setRevisionable(TRUE)
         ->setTranslatable(TRUE)
         ->setSettings([
           'max_length' => 60,
           'text_processing' => 0,
         ])
         ->setDefaultValue('')
         ->setDisplayOptions('view', [
           'label' => 'above',
           'type' => 'string',
```

```
        'weight' => -4,
      ])
      ->setDisplayOptions('form', [
        'type' => 'string_textfield',
        'weight' => -4,
      ])
      ->setDisplayConfigurable('form', TRUE)
      ->setDisplayConfigurable('view', TRUE);



    $fields['usage_activity']= BaseFieldDefinition::create('list_string')
      ->setLabel(t('Usage of activity'))
      ->setRevisionable(TRUE)
      ->setTranslatable(TRUE)
      ->setDefaultValue('local')
      ->setRequired(TRUE)
      ->setSetting('allowed_values', $options)
      ->setDisplayOptions('form', [
        'type' => 'options_buttons',
        'weight' => 4,
      ]);


     $fields['status'] = BaseFieldDefinition::create('boolean')
      ->setLabel(t('Publishing status'))
      ->setDescription(t('A boolean indicating whether the Activity is
published.'))
      ->setRevisionable(TRUE)
      ->setTranslatable(TRUE)
      ->setDefaultValue(TRUE);


     $fields['created'] = BaseFieldDefinition::create('created')
      ->setLabel(t('Authored on'))
      ->setDescription(t('The time that the Module was created.'))
      ->setRevisionable(TRUE)
      ->setTranslatable(TRUE)
      ->setDisplayOptions('view', [
        'label' => 'hidden',
        'type' => 'timestamp',
        'weight' => 0,
      ])
      ->setDisplayOptions('form', [
        'type' => 'datetime_timestamp',
        'weight' => 10,
      ])
      ->setDisplayConfigurable('form', TRUE);
```

```
    $fields['changed'] = BaseFieldDefinition::create('changed')
      ->setLabel(t('Changed'))
      ->setDescription(t('The time that the Module was last edited.'))
      ->setRevisionable(TRUE)
      ->setTranslatable(TRUE);


    return $fields;
  }
```

ДОДАТОК Д

Набір тест-кейсів для розроблюваної платформи.

№	Назва тесту	Роль користувача	Очікуваний результат	Фактичний результат
1	Логінація	Всі ролі	Користувач вводить власний емейл та пароль, його редіректить на головну сторінку	Користувач вводить власний емейл та пароль, його редіректить на головну сторінку
2	Перенаправлення на сторінку створення першого кроку тренінгу	Викладач	На сторінці каталогу всіх тренінгів викладач натискає «Створити новий тренінг», після чого його перекидає на перший крок створення тренінгу	На сторінці каталогу всіх тренінгів викладач натискає «Створити новий тренінг», після чого його перекидає на перший крок створення тренінгу
3	Створення курсів	Викладач	На другому кроці створення тренінгу користувач натискає «Додати перший айтем», після чого бачить кнопку «Створити новий курс». Після створення курс з’являється в навігаційній панелі тренінгу	На другому кроці створення тренінгу користувач натискає «Додати перший айтем», після чого бачить кнопку «Створити новий курс». Після створення курс з’являється в навігаційній панелі тренінгу
4	Створення модуля у курсі	Викладач	На третьому кроці створення тренінгу користувач відкриває курс, натискає «Додати перший айтем» і бачить кнопку «Створити новий модуль». Створений модуль відображається в навігаційній панелі з кількістю модулів	На третьому кроці створення тренінгу користувач відкриває курс, натискає «Додати перший айтем» і бачить кнопку «Створити новий модуль». Створений модуль відображається в навігаційній панелі з кількістю модулів

№	Назва тесту	Роль користувача	Очікуваний результат	Фактичний результат
5	Створення навчального матеріалу	Викладач	На четвертому кроці створення тренінгу викладач бачить список курсів з вкладеними модулями. Для користувача відображається кнопка «Додати активність», після відображається список активностей	На четвертому кроці створення тренінгу викладач бачить список курсів з вкладеними модулями. Для користувача відображається кнопка «Додати активність», після відображається список активностей
6	Додавання студентів до тренінгу	Викладач	На останньому кроці тренінгу викладач натискає «Додати учасників», обирає їх із пошуку, після чого з’являється кнопка «Опублікувати»	На останньому кроці тренінгу викладач натискає «Додати учасників», обирає їх із пошуку, після чого з’являється кнопка «Опублікувати»
7	Долучення до тренінгу	Студент	На сторінці каталогу всіх тренінгів студент бачить список доступних тренінгів. Студент натискає «Більше» біля тренінгу, переходить на його сторінку, де бачить кнопку «Записатись». Після її натискання з’являється модальне вікно підтвердження, а далі список курсів	На сторінці каталогу всіх тренінгів студент бачить список доступних тренінгів. Студент натискає «Більше» біля тренінгу, переходить на його сторінку, де бачить кнопку «Записатись». Після її натискання з’являється модальне вікно підтвердження, а далі список курсів
8	Проходження матеріалів курсу	Студент	Користувач, який записаний на курс, переходить до вправ і може переглядати відео, читати текстові матеріали та проходити тести	Користувач, який записаний на курс, переходить до вправ і може переглядати відео, читати текстові матеріали та проходити тести

№	Назва тесту	Роль користувача	Очікуваний результат	Фактичний результат
9	Помилка студента в автоматичному тесті	Студент	Після неправильної відповіді студент бачить повідомлення про помилку та правильну відповідь, бал не зараховується	Після неправильної відповіді студент бачить повідомлення про помилку та правильну відповідь, бал не зараховується
10	Коректна відповідь на тест	Студент	Студент правильно відповів на автоматичний тест, бал зарахувався	Студент правильно відповів на автоматичний тест, бал зарахувався
11	Достатня кількість балів для завершення курсу	Студент	Студент набирає мінімально необхідну кількість балів — курс позначається як завершений	Студент набирає мінімально необхідну кількість балів — курс позначається як завершений
12	Недостатня кількість балів для завершення	Студент	Студент не набирає мінімальну кількість балів – курс не завершується, поруч з назвою модуля з’являється іконка «Restart»	Студент не набирає мінімальну кількість балів – курс не завершується, поруч з назвою модуля з’являється іконка «Restart»

ДОДАТОК Е

Код функції my_custom_module_schema() у файлі .install

```
 /**
 * Implements hook_schema().
 */
function my_custom_module_schema() {
  // Create the my_custom module relationship table.
  $schema['my_custom_module_relationship'] = [
    'description' => 'Table storing what activities belong to what
modules',
    'fields' => [
      'omr_id' => [
        'type' => 'serial',
        'size' => 'normal',
        'unsigned' => TRUE,
        'not null' => TRUE,
        'description' => 'The primary identifier of this relationship.',
      ],
      'omr_pid' => [
        'type' => 'int',
        'size' => 'normal',
        'unsigned' => TRUE,
        'not null' => FALSE,
        'default' => NULL,
        'description' => 'The parent relationship of this relationship.',
      ],
      'parent_id' => [
        'type' => 'int',
        'unsigned' => TRUE,
        'not null' => TRUE,
        'description' => 'The Module that this activity belongs to.',
      ],
      'parent_vid' => [
        'type' => 'int',
        'unsigned' => TRUE,
        'not null' => TRUE,
        'description' => 'The Module version that this activity belongs
to.',
      ],
      'child_id' => [
        'type' => 'int',
        'unsigned' => TRUE,
        'not null' => TRUE,
        'description' => 'The Activity ID.',
      ],
      'child_vid' => ["
        'type' => 'int',
        'unsigned' => TRUE,
```

```
        'not null' => TRUE,
        'description' => 'The Activity version ID.',
      ],
      'activity_status' => [
        'type' => 'int',
        'size' => 'tiny',
        'unsigned' => TRUE,
        'not null' => TRUE,
        'default' => 1,
        'description' => 'The status of the Activity in this Module.
0=random, 1=always',
      ],
      'weight' => [
        'type' => 'int',
        'not null' => TRUE,
        'default' => 0,
        'description' => 'The weight of this Activity in the Module.',
      ],
      'max_score' => [
        'type' => 'int',
        'not null' => TRUE,
        'default' => 0,
        'description' => 'The max score of the Activity in this Module.',
      ],
      'auto_update_max_score' => [
        'type' => 'int',
        'size' => 'tiny',
        'not null' => TRUE,
        'default' => 0,
        'description' => 'Boolean indicating whether updates to the
Activity will update the max score of the Activity in the Module.',
      ],
      'group_id' => [
        'type' => 'int',
        'size' => 'normal',
        'description' => "Training ID",
      ],
    ],
    'primary key' => ['omr_id'],
    'unique keys' => [
      'parent_child' => [
        'parent_id',
        'parent_vid',
        'child_id',
        'child_vid',
      ],
    ],
    'indexes' => [
      'parent_vid' => ['parent_vid'],
      'child_vid' => ['child_vid'],
      'parent_id' => ['parent_id'],
```

```
    'child_id' => ['child_id'],
    'group_id' => ['group_id'],
  ],
];

$schema['my_custom_module_result_options'] = [
  'description' => 'Table storing result options for module.',
  'fields' => [
    'option_id' => [
      'type' => 'serial',
      'size' => 'normal',
      'unsigned' => TRUE,
      'not null' => TRUE,
      'description' => 'The primary identifier for the range.',
    ],
    'module_id' => [
      'type' => 'int',
      'unsigned' => TRUE,
      'not null' => TRUE,
      'description' => 'Module identifier.',
    ],
    'module_vid' => [
      'type' => 'int',
      'unsigned' => TRUE,
      'not null' => TRUE,
      'description' => 'Module revision identifier.',
    ],
    'option_start' => [
      'type' => 'int',
      'unsigned' => TRUE,
      'default' => 0,
      'description' => 'Score range low value.',
    ],
    'option_end' => [
      'type' => 'int',
      'unsigned' => TRUE,
      'default' => 0,
      'description' => 'Score range high value.',
    ],
    'option_name' => [
      'type' => 'varchar',
      'length' => 255,
      'not null' => TRUE,
      'description' => 'The name of this range.',
    ],
    'option_summary' => [
      'type' => 'text',
      'description' => 'The text to show when this range is met.',
    ],
    'option_summary_format' => [
      'type' => 'varchar',
```

```
        'length' => 255,
        'not null' => TRUE,
        'description' => 'Text format of the range text.',
      ],
    ],
    'primary key' => ['option_id'],
    'indexes' => [
      'module_id' => ['module_id', 'module_vid'],
    ],
  ];

  return $schema;
}
```