

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «Магістр»

«Інтерактивна платформа онлайн-освіти з використанням
3D-моделей і доповненої реальності»

(тема кваліфікаційної роботи українською мовою)

**Interactive Online Education Platform Using 3D Models
and Augmented Reality**

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач денної форми навчання
спеціальності 122 Комп'ютерні науки

(код, назва спеціальності)

Освітня програма Комп'ютерні науки

(назва)

КРАВЧЕНКО Сергій Анатолійович

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

Керівник Мещеряков В.І., д.т.н., професор

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент канд. техн. наук, доцент кафедри інженерії ПЗ національного
університету "Одеська політехніка", Тройніна А.С.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:
Протокол засідання кафедри
Інформаційних технологій

№ від 2025 р.

Завідувачка кафедри

КАЗАКОВА Надія

(підпис)

(прізвище, ім'я)

Захищено на засіданні ЕК №
протокол № від 2025 р.

Оцінка / /
(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

СОКОЛОВ Артем

(підпис)

(прізвище, ім'я)

Одеса 2025

АНОТАЦІЯ

Магістерська кваліфікаційна робота присвячена розробці інтерактивної платформи онлайн-освіти з використанням 3D-моделей і технологій доповненої реальності. Актуальність дослідження зумовлена потребою модернізації освітнього процесу в умовах цифровізації, зростанням ролі дистанційного навчання та необхідністю підвищення наочності й практичної спрямованості навчальних матеріалів, особливо для технічних і природничих дисциплін.

У роботі виконано аналіз сучасних підходів до онлайн-освіти, існуючих платформ та інструментів з підтримкою 3D/AR/VR-технологій, а також визначено їхні переваги й обмеження. На основі проведеного аналізу сформульовано функціональні та нефункціональні вимоги до платформи, визначено ролі користувачів і сценарії їх взаємодії із системою.

Спроектовано архітектуру програмного забезпечення у вигляді багаторівневої клієнт–серверної системи, придатної для інтеграції 3D-моделей і механізмів доповненої реальності. Розроблено UML-діаграми варіантів використання, послідовності, станів, класів, а також ER-діаграму бази даних. База даних забезпечує зберігання користувачів, курсів, лабораторних робіт, AR-сцен, параметрів симуляцій та результатів навчання.

У межах роботи створено програмний прототип платформи, побудований на сучасному клієнт-серверному технологічному стеку. Проведено тестування основних функціональних модулів системи, що підтвердило коректність роботи прототипу та відповідність поставленим вимогам. Отримані результати свідчать про доцільність використання AR і 3D-моделей в онлайн-освіті для підвищення залученості студентів і якості засвоєння навчального матеріалу.

Ключові слова: онлайн-освіта, доповнена реальність, 3D-моделі, AR-лабораторні роботи, інтерактивне навчання, веб-платформа, клієнт-серверна архітектура, UML, база даних.

ABSTRACT

The master's qualification thesis is devoted to the development of an interactive online education platform using 3D models and augmented reality technologies. The relevance of the research is determined by the need to modernize the educational process in the context of digitalization, the growing role of distance learning, and the necessity to increase the visual clarity and practical orientation of educational materials, especially for technical and natural science disciplines.

The thesis analyzes modern approaches to online education, as well as existing platforms and tools that support 3D/AR/VR technologies, identifying their advantages and limitations. Based on this analysis, functional and non-functional requirements for the platform are formulated, and user roles along with their interaction scenarios within the system are defined.

A software architecture is designed in the form of a multi-tier client–server system suitable for the integration of 3D models and augmented reality mechanisms. UML diagrams, including use case, sequence, state, and class diagrams, as well as an ER diagram of the database, are developed. The database design ensures the storage of information about users, courses, laboratory works, AR scenes, simulation parameters, and learning outcomes.

Within the scope of the work, a software prototype of the platform is implemented using a modern client-server technology stack. Testing of the main functional modules of the system is carried out, confirming the correctness of the prototype's operation and its compliance with the defined requirements. The obtained results demonstrate the feasibility and effectiveness of using augmented reality and 3D models in online education to increase student engagement and improve the quality of learning outcomes.

Keywords: online education, augmented reality, 3D models, AR laboratory works, interactive learning, web platform, client–server architecture, UML, database.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	8
1.1 Аналіз сучасних підходів та інструментів онлайн-освіти	8
1.2 Інструменти для роботи з 3D-контентом в освіті	9
1.3 Методи побудови систем доповненої реальності	11
1.4 Аналіз платформ з підтримкою AR/VR-технологій	13
1.5 Постановка завдання.....	17
2 СПЕЦИФІКАЦІЯ ВИМОГ ДО ПЛАТФОРМИ	19
2.1 Функціональні вимоги	19
2.2 Нефункціональні вимоги.....	22
3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРАКТИВНОЇ ОСВІТНЬОЇ ПЛАТФОРМИ.....	24
3.1 Проєктування архітектури програмної системи	24
3.2 Проєктування поведінкових UML-діаграм системи	28
3.3 Проєктування бази даних інтерактивної платформи	33
3.4 Проєктування класів інтерактивної платформи	36
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРАКТИВНОЇ ПЛАТФОРМИ.....	39
4.1 Обґрунтування вибору технологій	39
4.2 Структура програмного прототипу	40
4.3 Реалізація функціональних модулів платформи	44
4.4 Приклад реалізації AR-лабораторних робіт з використанням 3D-моделей	47
4.5 Приклад використання та функціонування розробленої платформи	51

5	ТЕСТУВАННЯ ПРОГРАМНОГО ПРОТОТИПУ ІНТЕРАКТИВНОЇ ПЛАТФОРМИ	57
5.1	Набір тест-кейсів для ключових сценаріїв	60
5.2	Реалізація та приклади тестування програмних компонентів.....	61
	ВИСНОВКИ.....	67
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	69

ВСТУП

Стрімкий розвиток цифрових технологій, зокрема засобів тривимірної візуалізації та доповненої реальності (AR), докорінно змінює підходи до подачі та засвоєння навчального матеріалу. Традиційні системи дистанційного навчання вже не повною мірою відповідають сучасним вимогам інтерактивності, наочності та адаптивності, яких очікують користувачі. Особливо це стосується дисциплін, де критично важливими є просторові уявлення, практична взаємодія з об'єктами та моделювання процесів – інженерія, медицина, природничі науки, дизайн, архітектура тощо.

Доповнена реальність і 3D-моделі створюють можливості для підвищення ефективності навчання шляхом занурення студентів у віртуалізовані ситуації, які складно або неможливо відтворити в офлайн-середовищі [1]. Завдяки цьому формується глибше розуміння матеріалу та стійкі навички. Водночас на ринку онлайн-освіти досі бракує платформ, які комплексно поєднують інтерактивний контент, сучасні інструменти комунікації та зручні засоби створення й використання AR-об'єктів. Це визначає актуальність розроблення інтерактивної освітньої платформи, орієнтованої на практичне застосування новітніх технологій.

Об'єктом дослідження є процес організації та підтримки онлайн-навчання з використанням інтерактивних цифрових ресурсів.

Предметом дослідження є методи, технології та програмні засоби розроблення онлайн-платформ із підтримкою 3D-моделей і доповненої реальності.

Метою роботи є розроблення інтерактивної онлайн-платформи, що забезпечує підтримку 3D-моделей і технологій доповненої реальності для підвищення ефективності дистанційного навчання та створення умов для більш глибокої взаємодії користувачів із навчальним контентом.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати сучасні підходи та інструменти онлайн-освіти, а також існуючі платформи, що підтримують AR/VR-технології;
- визначити вимоги до функціональності платформи, структуру користувацьких сценаріїв та технічні обмеження;
- розробити архітектуру програмного забезпечення, придатну для інтеграції 3D-моделей і механізмів доповненої реальності;
- створити програмний прототип платформи, що забезпечує завантаження, відображення та взаємодію з тривимірними освітніми об'єктами;
- реалізувати функціональні модулі управління навчальними курсами, користувачами, тестуванням і комунікацією;
- протестувати платформу.

Кваліфікаційна робота містить 70 сторінок, з них 64 сторінок основного тексту, 14 рисунків, 3 таблиці, 25 джерел посилань.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Аналіз сучасних підходів та інструментів онлайн-освіти

Онлайн-освіта стала невід'ємною складовою освітнього простору, забезпечуючи доступ до навчальних ресурсів незалежно від місця перебування та графіка користувача. Основні сучасні підходи до організації онлайн-навчання включають наступні.

Масові відкриті онлайн-курси (МООС) – курси, що надаються широкій аудиторії через спеціалізовані платформи. Для них характерні відеолекції, тести, форуми, автоматизоване оцінювання [2].

Змішане навчання – поєднання традиційних аудиторних занять з онлайн-компонентами (електронні курси, інтерактивні завдання, віртуальні лабораторії) [3].

Мікронавчання – подання матеріалу у вигляді коротких, структурованих модулів, орієнтованих на швидке засвоєння конкретних компетентностей [4].

Компетентнісний підхід – фокус не тільки на передачі знань, а й на формуванні конкретних умінь та навичок, що відображається в структуруванні навчального контенту, системі оцінювання та зворотного зв'язку [5].

Основним програмним інструментом у сфері онлайн-освіти є системи управління навчанням (LMS) [6]. Такі системи забезпечують:

- створення та адміністрування курсів;
- управління користувачами та групами;
- проведення тестування, збору статистики та формування звітів;
- комунікацію між викладачами та студентами (форуми, чати, повідомлення).

Серед поширених рішень можна відзначити як традиційні LMS (Moodle, Blackboard, Canvas), так і хмарні освітні платформи (Google Classroom, Microsoft Teams for Education) та спеціалізовані платформи для онлайн-курсів.

Попри значні можливості, більшість таких платформ використовують переважно двовимірні засоби подання інформації (текст, відео, презентації). Інтеграція 3D- та AR-контенту в них обмежена або потребує додаткових плагінів та зовнішніх сервісів, що ускладнює впровадження подібних рішень на рівні окремого курсу чи навчального закладу.

1.2 Інструменти для роботи з 3D-контентом в освіті

У сучасному освітньому середовищі все більшого значення набувають інструменти, що забезпечують роботу з тривимірним контентом, особливо в навчальних дисциплінах, де необхідна просторова уява та наочне представлення складних об'єктів і процесів. До таких галузей належать інженерія, медицина, фізика, хімія, біологія, архітектура, дизайн, а також комп'ютерні науки. У межах відповідних курсів активно застосовуються:

- тривимірні моделі обладнання, механізмів, приладів і технічних систем;
- 3D-візуалізації анатомічних структур, молекул, фізичних та хімічних процесів;
- інтерактивні сцени, що відтворюють поведінку об'єктів у часі та просторі;
- віртуальні лабораторії та навчальні симулятори, які імітують реальні експерименти.

Для створення та обробки 3D-контенту використовуються як універсальні, так і спеціалізовані програмні засоби. До універсальних інструментів належать системи тривимірного моделювання, такі як Blender, які дозволяють створювати геометричні моделі, анімації та текстури з подальшим експортом у формати, придатні для веб- та AR-застосунків [7]. У технічній і інженерній освіті також застосовуються CAD-системи [8] (наприклад, Fusion 360), що забезпечують точне параметричне моделювання.

Для інтеграції 3D-контенту у веб-середовище поширеними є бібліотеки та фреймворки на основі WebGL, зокрема Three.js та Babylon.js, які дозволяють відображати інтерактивні тривимірні сцени безпосередньо в браузері. Такі інструменти відкривають можливості для створення інтерактивних демонстрацій, візуалізації лабораторних робіт і навчальних симуляцій без необхідності встановлення додаткового програмного забезпечення [9].

Окрему категорію становлять платформи з готовим 3D-контентом для освітніх цілей, зокрема mozaWeb, Lifelike, Evtika 3D, які пропонують бібліотеки моделей та інтерактивних сцен. Їхньою перевагою є швидкий доступ до візуальних матеріалів, однак такі платформи часто мають обмежені можливості адаптації контенту під конкретні навчальні сценарії та не завжди інтегруються з повноцінними системами управління навчанням [10].

Разом з тим, існують веб-сервіси та онлайн-репозиторії, що дозволяють завантажувати, переглядати та вбудовувати 3D-моделі у веб-сторінки. Попри зручність використання, подібні рішення здебільшого не орієнтовані на комплексну освітню діяльність, оскільки в них відсутні механізми структурування курсів, організації лабораторних робіт, оцінювання результатів і відстеження прогресу студентів. У таких випадках 3D-модель виступає ізольованим візуальним елементом і не інтегрується у цілісний навчальний сценарій.

Для розроблення інтерактивних AR/VR-сцен застосовуються спеціалізовані середовища просторового моделювання, наприклад Gravity Sketch, що дозволяє створювати тривимірні форми безпосередньо у віртуальному просторі. Це значно прискорює процес проектування моделей, призначених для наочного відтворення у доповненій реальності.

Окрему нішу посідають інструменти на основі штучного інтелекту, призначені для автоматизованої генерації тривимірних моделей. Одним із таких рішень є Meshy – платформа, що дає змогу отримувати 3D-об'єкти на основі текстових описів або двовимірних зображень. Використання подібних

AI-систем спрощує та здешевлює процес створення навчального контенту, дозволяючи швидко формувати об'єкти для візуалізації різних явищ або процесів, що особливо корисно для динамічних або міждисциплінарних курсів [11].

У сукупності ці інструменти формують широкий екосистемний спектр технологій для створення 3D-контенту, придатного для AR-додатків. Такий підхід дозволяє розробникам освітніх платформ підбирати оптимальні засоби залежно від предметної області, рівня деталізації моделі, апаратних можливостей користувачів та формату взаємодії з навчальним матеріалом. Завдяки цьому доповнена реальність стає не лише візуальним доповненням до курсу, а повноцінним інтерактивним інструментом, що сприяє глибшому розумінню складних об'єктів і процесів.

Таким чином, актуальною є потреба у створенні спеціалізованих освітніх платформ, у яких тривимірний контент та елементи доповненої реальності є органічною складовою навчального процесу. Такі платформи мають поєднувати 3D-візуалізацію з теоретичними матеріалами, інтерактивними завданнями, лабораторними роботами та системами контролю знань, забезпечуючи підвищення наочності, практичної спрямованості та ефективності онлайн-навчання.

1.3 Методи побудови систем доповненої реальності

Доповнена реальність (AR) – це технологія, що дозволяє накладати віртуальні об'єкти (3D-моделі, текст, графіку) на зображення реального світу в режимі реального часу. Для побудови AR-систем використовуються різні підходи.

Маркерна доповнена реальність. В основі лежить розпізнавання спеціальних маркерів (зображень, QR-кодів, патернів), які слугують прив'язкою для розміщення віртуального об'єкта [12].

Переваги:

- простота реалізації;
- стабільне позиціонування об'єктів відносно маркера;
- невисокі вимоги до обчислювальних ресурсів.

Недоліки:

- необхідність друку або відображення маркерів;
- обмеження щодо дизайну сцени (камера повинна бачити маркер).

Безмаркерна доповнена реальність. Використовує інформацію про навколишнє середовище (поверхні, об'єкти) з датчиків пристрою та алгоритмів комп'ютерного зору [13].

Сюди належать:

- AR на основі відстеження площин та глибини;
- AR на основі геолокації.

Переваги:

- природніша взаємодія користувача з віртуальними об'єктами;
- можливість прив'язки до реального простору без спеціальних маркерів.

Недоліки:

- підвищені вимоги до продуктивності пристрою;
- складніші алгоритми обробки.

Гібридні підходи. Поєднання маркерних і безмаркерних методів, використання декількох каналів (зоровий, геолокаційний, інерційні датчики) для більш стійкого відстеження та позиціонування об'єктів [14].

Для реалізації AR-сцен в освітніх платформах можуть застосовуватися:

- спеціалізовані фреймворки для мобільних пристроїв;
- кросплатформні рушії для 3D-графіки та AR/VR;
- веб-технології доповненої реальності (WebXR, бібліотеки поверх WebGL тощо), що дозволяють запускати AR через браузер без встановлення додаткових застосунків.

У контексті онлайн-освіти особливий інтерес становлять веборієнтовані рішення, які дають змогу запускати AR-сценарії без складної інсталяції, лише за допомогою камери та сучасного браузера.

Сучасні технології доповненої реальності тісно пов'язані з можливістю створення та інтеграції тривимірних об'єктів, що відтворюються у реальному просторі. Тому при побудові AR-систем важливим є не лише вибір алгоритмів позиціонування чи відстеження сцени, а й добір інструментів для формування та оптимізації 3D-контенту. Для освітніх платформ цей аспект має особливе значення, адже якість моделей безпосередньо впливає на наочність і ефективність навчального процесу.

1.4 Аналіз платформ з підтримкою AR/VR-технологій

На ринку присутні різні рішення, що поєднують освіту й AR/VR.

VR-платформи та віртуальні лабораторії, орієнтовані на повне занурення у віртуальне середовище. Вони потребують спеціалізованого обладнання (VR-окуляри, контролери), що не завжди доступне широкому колу користувачів.

Мобільні додатки з AR, розроблені для окремих навчальних курсів або дисциплін (наприклад, для вивчення анатомії, хімічних реакцій, інженерних конструкцій). Часто вони є автономними рішеннями і не інтегруються в LMS.

Онлайн-платформи з підтримкою 3D та AR, що дозволяють створювати й публікувати інтерактивні освітні сцени. Їхніми обмеженнями можуть бути:

- залежність від конкретного формату 3D-моделей;
- платна модель доступу або обмеження в безкоштовних версіях;
- відсутність гнучкої системи управління курсами та користувачами;
- складність інтеграції з існуючими освітніми системами.

Загалом, аналіз наявних платформ показує, що хоча окремі компоненти (3D-візуалізація, AR-сцени, тести, відео) представлені досить широко,

комплексні рішення, які б поєднували структуроване онлайн-навчання з глибокою інтеракцією з 3D-моделями та AR, залишаються обмеженими.

Розвиток технологій доповненої та віртуальної реальності привів до появи низки освітніх рішень, які пропонують готові віртуальні лабораторії, 3D-сцени або середовища для створення власних віртуальних просторів. Для обґрунтування необхідності розробки інтерактивної онлайн-платформи з використанням 3D-моделей і доповненої реальності доцільно проаналізувати наявні аналоги, оцінити їх сильні та слабкі сторони з точки зору дидактичних можливостей, відкритості та можливості розширення.

Проведемо аналіз існуючих платформ.

CoSpaces Edu (Delightex Edu) [15] є веб- та мобільною платформою, яка дозволяє учням і викладачам створювати власні інтерактивні 3D-сцени з подальшим переглядом у режимах AR та VR. Платформа підтримує конструювання сцен на основі готових об'єктів, анімацію за допомогою блокового або текстового програмування, а також перегляд проєктів у браузері та через мобільні додатки iOS/Android.

Основний акцент зроблено на творчості учнів, гейміфікації та розвитку навичок програмування, тоді як засоби побудови формалізованих наукових моделей (з параметрами, вимірюваннями, обробкою даних тощо) реалізовані обмежено і вимагають додаткового програмування зі сторони користувача.

Система Mozaik Education [16] пропонує екосистему mozaBook (для вчителя) та mozaWeb (онлайн-доступ для учнів), що містить велику медіатеку з понад тисячею 3D-сцен, відео, інтерактивних вправ і готових цифрових уроків. Учні можуть працювати з 3D-сценами, у тому числі переглядати їх у VR-режимі через спеціальний 3D-переглядач, а також виконувати інтерактивні завдання й домашню роботу онлайн.

Платформа орієнтована насамперед на шкільну освіту й споживання вже підготовленого контенту. Можливості створення власних 3D-моделей та складних AR-сцен для студентів є обмеженими: користувач більше «споживає» наявний контент, ніж конструює свої інтерактивні експерименти.

Платформа Labster [17] – спеціалізована платформа віртуальних лабораторій для вивчення природничих наук (біологія, хімія, фізика тощо), що надає бібліотеку 3D-симуляцій і VR-лабораторій для університетів та шкіл. Платформа позиціонується як засіб підвищення успішності студентів та розширення доступу до лабораторних робіт за рахунок імітації експериментів у безпечному віртуальному середовищі. Labster має розвинений готовий контент та інтеграції з поширеними LMS, однак орієнтований переважно на закриті, заздалегідь розроблені сценарії. Студенти не мають повноцінного середовища для самостійної розробки нових віртуальних лабораторних робіт з можливістю глибокої модифікації фізичних параметрів моделей.

Система ClassVR [18] представляє собою апаратно-програмний комплекс для навчальних закладів, що включає VR-гарнітури, бібліотеку навчального контенту та вчительський портал для керування заняттями. Рішення сфокусоване на створенні занурювального досвіду в класі, підтримує як VR, так і елементи AR/змішаної реальності та широко використовується у вищій освіті. Сильна сторона ClassVR – інтегрованість «під ключ» (обладнання плюс контент), проте суттєвим обмеженням є прив'язаність до специфічного обладнання, обмежені можливості розробки власних складних лабораторних моделей безпосередньо користувачами та неорієнтованість на веб-доступ із будь-яких пристроїв.

Ще одна система Mozilla Hubs [19] – веб-орієнтована, переважно VR-платформа для створення спільних 3D-просторів (віртуальні аудиторії, виставки, класи), яка працює без встановлення додаткового ПЗ безпосередньо в браузері. Використовуючи інструмент Spoke, користувачі можуть проєктувати власні простори, імпортувати 3D-моделі з інших редакторів (наприклад, Blender) та проводити віртуальні зустрічі, лекції чи семінари. Mozilla Hubs забезпечує хороші можливості для колаборативної взаємодії в реальному часі, але не має спеціалізованих засобів для побудови формалізованих освітніх експериментів (вимірювань, автоматизованої перевірки результатів, інтеграції з системами контролю знань тощо).

Для узагальнення проведеного аналізу наведемо порівняльну таблицю за ключовими критеріями, важливими для побудови інтерактивної платформи онлайн-освіти з підтримкою 3D-моделей та AR/VR (табл. 1.1).

Таблиця 1.1 – Порівняльна характеристика освітніх платформ з підтримкою AR/VR

Платформа	Веб-доступ без інсталяції	Підтримка AR	Підтримка VR	Вбудований конструктор 3D-сцен
CoSpaces Edu	+	+	+	+
mozaWeb	+	+ (3D/VR-сцени)	+	–
Labster	+	–	+	–
ClassVR	– (фокус на гарнітурах)	+	+	–
Mozilla Hubs	+	– (здебільшого VR)	+	+ (через Spoke)

Проведений аналіз показує, що:

- частина платформ (mozaWeb, Labster, ClassVR) надає, передусім, закриті бібліотеки готового контенту, орієнтовані на типові навчальні сценарії;
- платформи типу CoSpaces Edu та Mozilla Hubs забезпечують більш відкриті середовища для конструювання 3D-просторів, однак не пропонують повноцінного інструментарію для формалізованих наукових експериментів (задання фізичних параметрів, автоматизована оцінка, тісна інтеграція з онлайн-курсами та системами контролю знань);

– частина рішень прив’язана до спеціалізованого обладнання (ClassVR), що ускладнює масове впровадження в умовах змішаної чи дистанційної освіти та обмежує доступність для студентів, які працюють з дому.

Отже, існує ніша для веб-орієнтованої інтерактивної платформи, яка поєднує:

- можливість створення та використання параметризованих 3D-моделей та AR-сцен для лабораторних робіт;
- незалежність від спеціалізованих гарнітур (підтримка звичайних ПК і мобільних пристроїв);
- інтеграцію з онлайн-курсами (LMS) та засобами автоматизованого контролю знань;
- орієнтацію на конструювання власних експериментів студентами, а не лише на відтворення наперед заданих симуляцій.

Саме ці обмеження існуючих рішень обґрунтовують доцільність розробки запропонованої в роботі інтерактивної платформи онлайн-освіти з використанням 3D-моделей і доповненої реальності.

1.5 Постановка завдання

На основі проведеного аналізу можна сформулювати основні вимоги до розроблюваної платформи:

- забезпечення веб-доступу до навчальних курсів без необхідності встановлення спеціального програмного забезпечення;
- підтримка завантаження, зберігання та відображення 3D-моделей навчальних об’єктів;
- інтеграція AR-сцен з можливістю перегляду віртуальних об’єктів у реальному оточенні за допомогою камери пристрою;
- наявність механізмів організації навчального процесу: структурування курсів, модулів, уроків, завдань;

- підтримка реєстрації користувачів, розмежування ролей (адміністратор, викладач, студент);
- реалізація базових засобів оцінювання та відстеження прогресу;
- забезпечення адаптивного інтерфейсу для коректної роботи на настільних і мобільних пристроях;
- можливість подальшої інтеграції з існуючими LMS або сервісами (через API чи модулі).

Задача дослідження полягає у розробці концепції, архітектури і програмної реалізації інтерактивної платформи онлайн-освіти, що відповідає зазначеним вимогам та надає користувачам (викладачам і студентам) зручні засоби створення, перегляду і використання навчального контенту на основі 3D-моделей і доповненої реальності.

2 СПЕЦИФІКАЦІЯ ВИМОГ ДО ПЛАТФОРМИ

Розроблення інтерактивної платформи онлайн-освіти з використанням 3D-моделей та технологій доповненої реальності потребує формалізації вимог, що визначають її функціональні можливості, архітектурні особливості та експлуатаційні характеристики. Правильно сформульована специфікація вимог забезпечує узгодженість між очікуваннями користувачів, технічними можливостями системи та критеріями її подальшої оцінки.

Платформа має підтримувати інтерактивну взаємодію користувача з 3D-об'єктами та AR-сценами, забезпечувати доступ до навчальних матеріалів, виконання лабораторних робіт, фіксацію результатів, а також можливість розширення функціоналу. Система орієнтована на дистанційне та змішане навчання, що зумовлює високі вимоги до стабільності, продуктивності та зручності використання.

2.1 Функціональні вимоги

Платформа повинна забезпечувати:

- завантаження, зберігання та відтворення 3D-моделей різних форматів;
- інтеграцію AR-сцен на основі технологій WebAR або мобільних AR-SDK;
- можливість додавання текстових, графічних, відео- та інтерактивних матеріалів;
- підтримку структурованого подання навчальних модулів (лекції, лабораторні роботи, тести).

Платформа повинна забезпечувати:

- масштабування, обертання та переміщення 3D-об'єктів у режимі реального часу;
- активацію анімацій, прихованих елементів, супровідних пояснень;
- взаємодію з AR-об'єктами у фізичному просторі користувача;

– параметричне налаштування моделей (зміна величин, форм, експериментальних значень).

Платформа має підтримувати:

- реалізацію інтерактивних симуляцій (наприклад, експериментів з моментом інерції маятника Обербека чи інтерференцією Юнга);
- автоматизоване обчислення результатів на основі формул, заданих у методичних матеріалах;
- генерацію індивідуальних варіантів експериментів через змінні параметри;
- фіксацію результатів, їх перевірку та збереження у профілі студента.

Система має передбачати наступні ролі.

Студент – доступ до матеріалів, виконання лабораторних робіт, отримання зворотного зв'язку.

Викладач – створення та редагування курсів, аналіз результатів студентів.

Адміністратор – керування користувачами, контентом та конфігурацією системи.

Платформа повинна:

- формувати звіти про виконання лабораторних робіт;
- забезпечувати зберігання історії спроб;
- підтримувати експорт результатів у PDF/CSV;
- надавати викладачу зведену статистику щодо успішності студентів.

Модель взаємодії користувачів із системою наведено на рис. 2.1. Подана діаграма варіантів використання відображає основні функціональні можливості інтерактивної онлайн-платформи освіти з використанням 3D-моделей та технологій доповненої реальності (AR). Діаграма структурована відповідно до трьох основних ролей користувачів системи: Студент, Викладач та Адміністратор, кожна з яких має свій набір доступних сценаріїв взаємодії.

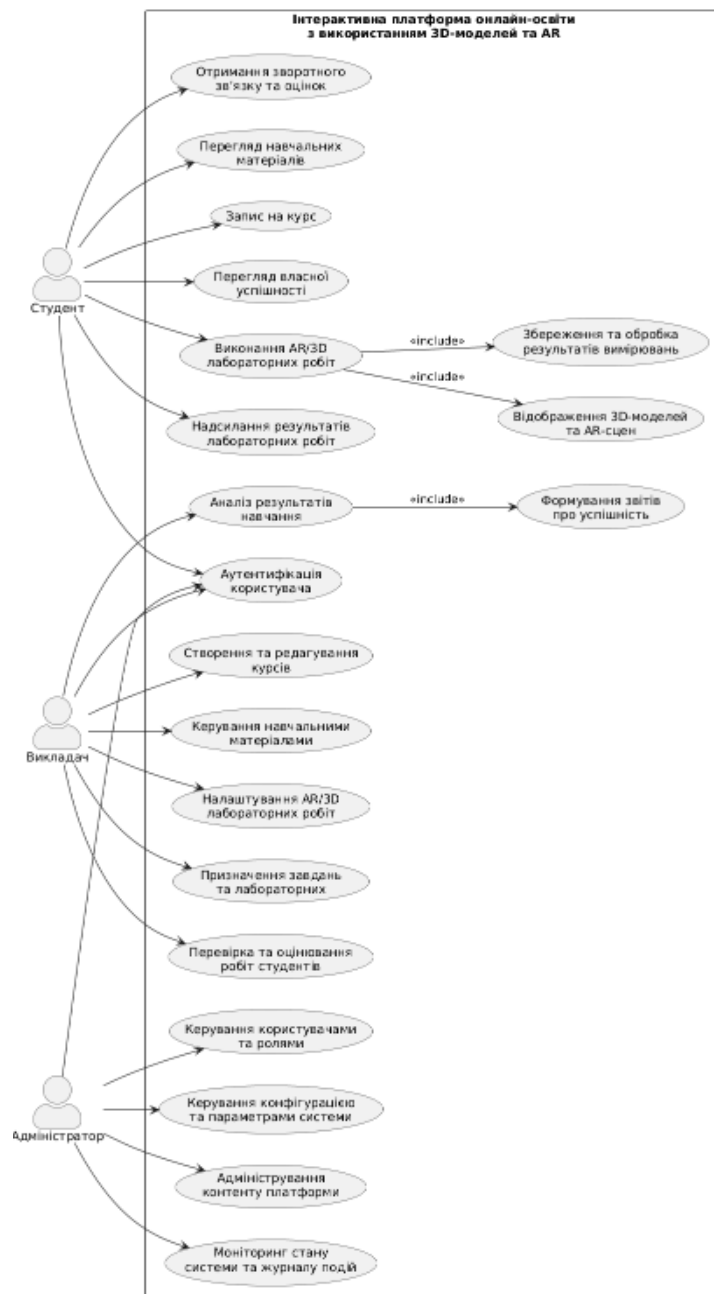


Рисунок 2.1 – Діаграма варіантів використання

Студент взаємодіє з платформою для отримання освітнього контенту, виконання лабораторних робіт у форматі AR/3D та відстеження власних навчальних результатів.

Викладач відповідає за організацію навчального процесу, створення навчального контенту й оцінювання студентських робіт.

Адміністратор відповідає за підтримку системи, управління доступами та технічними параметрами.

2.2 Нефункціональні вимоги

Для забезпечення стабільної, ефективної та зручної роботи інтерактивної платформи онлайн-освіти, окрім функціональних можливостей, необхідно також визначити нефункціональні вимоги, які регламентують якісні характеристики системи, умови її експлуатації та обмеження, пов'язані з продуктивністю, безпекою, надійністю, масштабованістю та зручністю використання.

Продуктивність:

- завантаження 3D-моделей має здійснюватися не більше ніж за 3-5 секунд за умови середньої складності моделей;
- AR-сцена повинна відображатися зі швидкістю не менше ніж 30 FPS на підтримуваних пристроях;
- система має забезпечувати роботу одночасно не менш ніж 100 активних користувачів.

Надійність:

- безперервна робота сервера – не менше 99% часу;
- збереження користувацьких даних при можливих збоях;
- відновлення сесії користувача після перезавантаження.

Масштабованість:

- можливість додавання нових модулів без зміни основної архітектури;
- підтримка розширення бібліотеки 3D-моделей та інтеграції із зовнішніми сервісами.

Безпека:

- автентифікація користувачів та безпечне зберігання персональних даних;
- захист від несанкціонованого доступу до навчальних матеріалів;
- передача даних через захищені протоколи (HTTPS).

Юзабіліті:

- інтуїтивний інтерфейс, адаптований під мобільні та десктопні пристрої;

- наявність навчальних підказок;

- локалізація інтерфейсу (мінімум українська + англійська).

Технічні обмеження:

- веб-платформа має працювати в сучасних браузерях (Chrome, Firefox, Safari, Edge) без встановлення додаткового ПЗ;

- AR-функціональність може вимагати камер з підтримкою WebRTC та сучасних мобільних пристроїв;

- максимальний розмір 3D-моделі – до 20 МБ;

- застосування форматів, оптимізованих для Web-рендерингу (GLTF/GLB);

- інтеграція з хмарним сервером для зберігання контенту.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРАКТИВНОЇ ОСВІТНЬОЇ ПЛАТФОРМИ

3.1 Проєктування архітектури програмної системи

Архітектура програмного забезпечення інтерактивної платформи онлайн-освіти має забезпечувати:

- підтримку ролей «Студент», «Викладач», «Адміністратор» із чітко розмежованими правами доступу;
- інтеграцію 3D-моделей та AR-сцен у навчальний контент (лабораторні роботи, візуалізації, тренажери);
- кросплатформеність (веб-браузер, мобільні пристрої, AR-орієнтовані пристрої) без дублювання бізнес-логіки;
- масштабованість при збільшенні кількості курсів, студентів та AR-об'єктів;
- розширюваність, тобто можливість підключення нових типів лабораторних робіт, додаткових AR-модулів, аналітичних сервісів;
- інформаційну безпеку: автентифікація, авторизація, журналювання дій, захист навчальних даних та результатів вимірювань;
- інтеграцію з репозиторіями 3D-ресурсів (колекції моделей, зовнішні каталоги), а також з інструментами створення контенту.

На основі цих вимог обирається архітектурний стиль та визначаються основні підсистеми платформи.

З огляду на розподіленість користувачів, необхідність роботи в мережі Інтернет та інтеграцію з зовнішніми сервісами доцільно обрати багаторівневу клієнт-серверну [20] архітектуру з чітким поділом на:

- рівень представлення (Front-end) – веб-інтерфейс платформи та AR-клієнт, що взаємодіє з користувачем;
- рівень прикладної логіки – реалізація бізнес-процесів онлайн-навчання, керування курсами, лабораторіями, ролями та доступом;

- рівень даних (Data Layer) – сховище користувачів, курсів, результатів тестування, а також файлове сховище 3D-моделей та пов'язаних ресурсів;
- інтеграційний рівень (Integration / AR Engine Layer) – модулі, що забезпечують рендеринг 3D-об'єктів, обробку просторових даних та взаємодію з AR-фреймворками.

Запропонована архітектура програмного забезпечення реалізована у вигляді багаторівневої клієнт-серверної системи, що забезпечує розмежування відповідальностей між клієнтським рівнем, рівнем прикладної логіки, рівнем даних та рівнем 3D/AR-рушія (рис. 3.1).

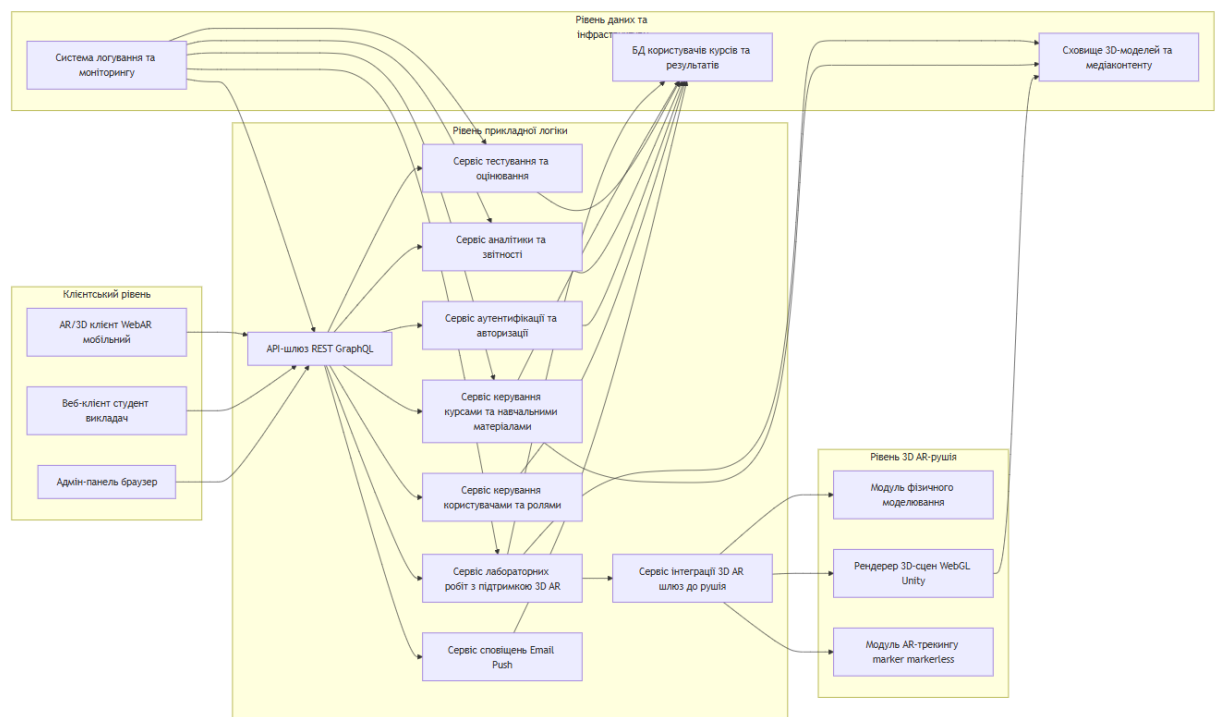


Рисунок 3.1 – Архітектура інтерактивної платформи

Такий підхід забезпечує слабке зв'язування компонентів, спрощує супровід та дозволяє незалежно масштабувати фронтенд, бекенд і підсистему AR-візуалізації.

Опишемо структуру програмної системи.

У межах обраного стилю виділяються такі основні підсистеми.

Підсистема керування користувачами та ролями:

- реєстрація та автентифікація користувачів;
- розподіл на ролі (студент, викладач, адміністратор);
- керування правами доступу до курсів, лабораторій та адміністративних функцій.

Підсистема управління навчальним контентом:

- створення, редагування та публікація курсів;
- організація контенту у вигляді модулів, лекцій, лабораторних робіт;
- прив'язка 3D-моделей та AR-сцен до конкретних навчальних завдань.

Підсистема проведення лабораторних робіт:

- сценарії виконання віртуальних та AR-лабораторій;
- засоби налаштування параметрів експерименту;
- збір, валідація й збереження результатів вимірювань;
- передача результатів у модуль оцінювання.

AR/3D-підсистема:

- завантаження 3D-моделей із репозиторію;
- підготовка сцен, налаштування камер та освітлення;
- інтеграція з AR-фреймворками для відображення моделей у реальному оточенні;
- синхронізація дій користувача в AR-інтерфейсі з логікою курсу (фіксація кроків, значень параметрів).

Підсистема тестування та аналітики:

- формування тестових завдань, контрольних робіт;
- автоматизоване оцінювання результатів, формування звітів;
- візуалізація прогресу студентів для викладача.

Адміністративна підсистема:

- керування конфігурацією платформи (набір доступних курсів, AR-модулів, мов інтерфейсу);
- моніторинг стану системи та журналювання подій;
- резервне копіювання та відновлення даних.

Щоб забезпечити коректну роботу AR-лабораторій, архітектура передбачає окремий модуль інтеграції 3D-ресурсів, який виконує:

- зберігання 3D-моделей, текстур та пов'язаних метаданих у спеціалізованому репозиторії;
- конвертацію моделей у формати, оптимізовані для веб-та мобільного рендерингу;
- опис освітніх сценаріїв у вигляді конфігурацій (які параметри модель має, що користувач може змінювати, які формули застосовуються для обчислень, як формується візуальний результат);
- обмін даними між AR-клієнтом і сервером через REST/GraphQL API чи WebSocket-канали (передавання параметрів експерименту, проміжних та підсумкових результатів).

На стороні клієнта AR-модуль має можливість:

- розпізнавати маркери чи поверхні в реальному середовищі;
- розміщувати й масштабувати 3D-моделі відповідно до вимог лабораторної роботи;
- фіксувати дії користувача (зміна параметрів, запуск експерименту, зчитування значень) й передавати їх у підсистему оцінювання.

Передбачена архітектура дозволяє:

- легко додавати нові лабораторні роботи за рахунок опису їх параметрів та формул на рівні конфігурацій без переробки ядра системи;
- переносити AR-клієнт на інші платформи за умови збереження інтерфейсу взаємодії з сервером;
- масштабувати бекенд (горизонтально або через виділення мікросервісів для найнавантажениших підсистем – тестування, обробки результатів, рендерингу моделей);
- підтримувати необхідні нефункціональні вимоги: продуктивність, надійність, модульність, інформаційну безпеку.

Таким чином, розроблена архітектура є базою для подальшої реалізації інтерактивної платформи онлайн-освіти, що забезпечує органічну інтеграцію 3D-моделей і механізмів доповненої реальності у навчальний процес.

3.2 Проєктування поведінкових UML-діаграм системи

У процесі проєктування інтерактивної платформи онлайн-освіти з підтримкою 3D-моделей та механізмів доповненої реальності важливим етапом є моделювання динамічної взаємодії між компонентами системи. Для моделювання динамічної поведінки системи та взаємодії її компонентів у процесі виконання AR-лабораторних робіт було використано UML-діаграми послідовності та станів.

Діаграми послідовності відіграють ключову роль у формалізації алгоритмів виконання лабораторних робіт, відображаючи:

- часову послідовність викликів між компонентами;
- залежності між сервісами та їхніми відповідальностями;
- життєвий цикл AR-сесії;
- точки інтеграції з 3D-рушієм та фізичним моделюванням;
- механізми обробки та збереження результатів.

Для AR-лабораторних робіт важливою особливістю є значна кількість взаємодій між рівнем клієнта та серверною частиною, а також необхідність синхронізації даних між модулем візуалізації (WebGL/Unity) та сервісами платформи. Саме тому діаграма послідовності використовується для деталізації складних сценаріїв, які включають наступні.

Ініціалізацію AR-лабораторії – автентифікація користувача, отримання конфігураційних даних, завантаження 3D-моделей та ресурсів.

Формування AR-сцени – підготовка моделі, застосування фізичних параметрів, ініціалізація AR-трекінгу.

Взаємодію студента з AR-лабораторною роботою – зміна параметрів симуляції, запуск обчислень, перегляд візуальних результатів.

Збір та збереження результатів експерименту – відправлення вимірювань на сервер, оцінювання та формування зворотного зв'язку.

На діаграмі (рис. 3.2) представлено типовий сценарій взаємодії студента із системою під час виконання AR-лабораторної роботи.

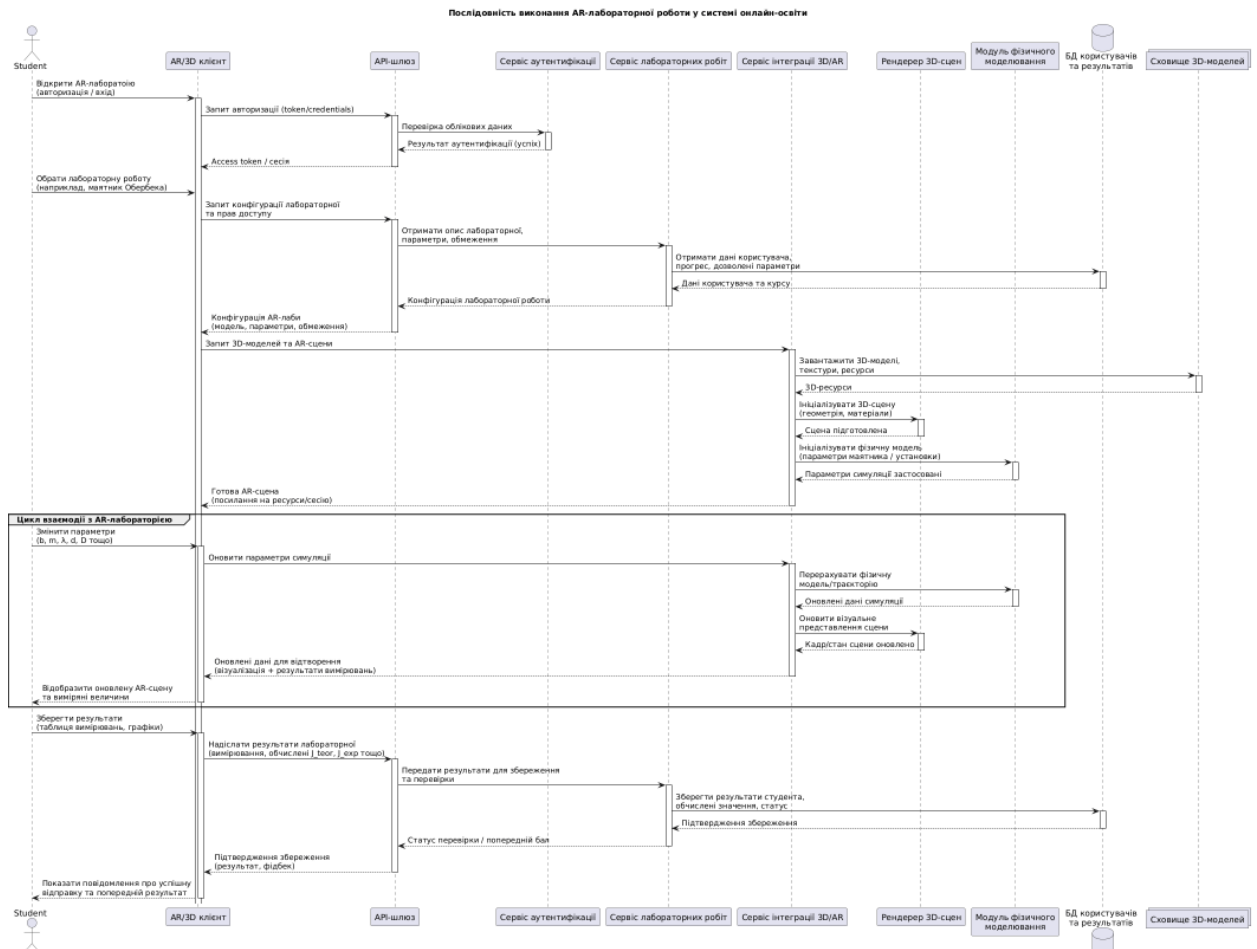


Рисунок 3.2 – Діаграми послідовності виконання AR-лабораторної роботи

Послідовність охоплює такі ключові етапи.

1. Ініціалізація та автентифікація.

Студент ініціює AR-лабораторію через клієнтський застосунок (мобільний або WebAR). AR-клієнт надсилає запит автентифікації до API-шлюзу, який передає дані до сервісу автентифікації. Після успішної перевірки користувач отримує токен доступу, що дає змогу виконувати подальші запити.

2. Отримання та підготовка лабораторної роботи.

Користувач обирає потрібну лабораторну роботу; клієнт надсилає запит до сервісу лабораторних робіт. Сервіс повертає:

- список параметрів симуляції;
- вимоги до виконання;
- структуру AR-сцени;
- посилання на 3D-моделі та текстури.

Після цього AR-клієнт звертається до сервісу інтеграції 3D/AR, який взаємодіє з:

- модулем фізичного моделювання;
- рендерером 3D-сцен;
- AR-трекінгом.

На цьому етапі готуються геометричні ресурси, застосовуються початкові фізичні параметри, формується AR-сцена для подальшого виконання лабораторної роботи.

3. Виконання лабораторного експерименту.

Студент змінює входові параметри (наприклад, масу, відстань, довжину хвилі тощо). AR-клієнт відсилає змінені дані назад до сервісів:

- нові параметри симуляції передаються фізичному модулю;
- оновлена 3D-сцена формується рендерером;
- AR-трекінг синхронізує положення об'єктів у фізичному просторі.

Користувач отримує візуалізацію результатів у реальному часі.

4. Передача та збереження результатів.

Після завершення лабораторної роботи студент надсилає результати вимірювань на сервер. Сервіс лабораторних робіт:

- зберігає результати у базі даних,
- передає їх сервісу аналітики для подальшої обробки,
- надсилає студенту підтвердження та зворотний зв'язок.

Діаграма станів AR-лабораторної роботи, наведена на рис. 3.3, відображає життєвий цикл лабораторної роботи від етапу створення до архівування.

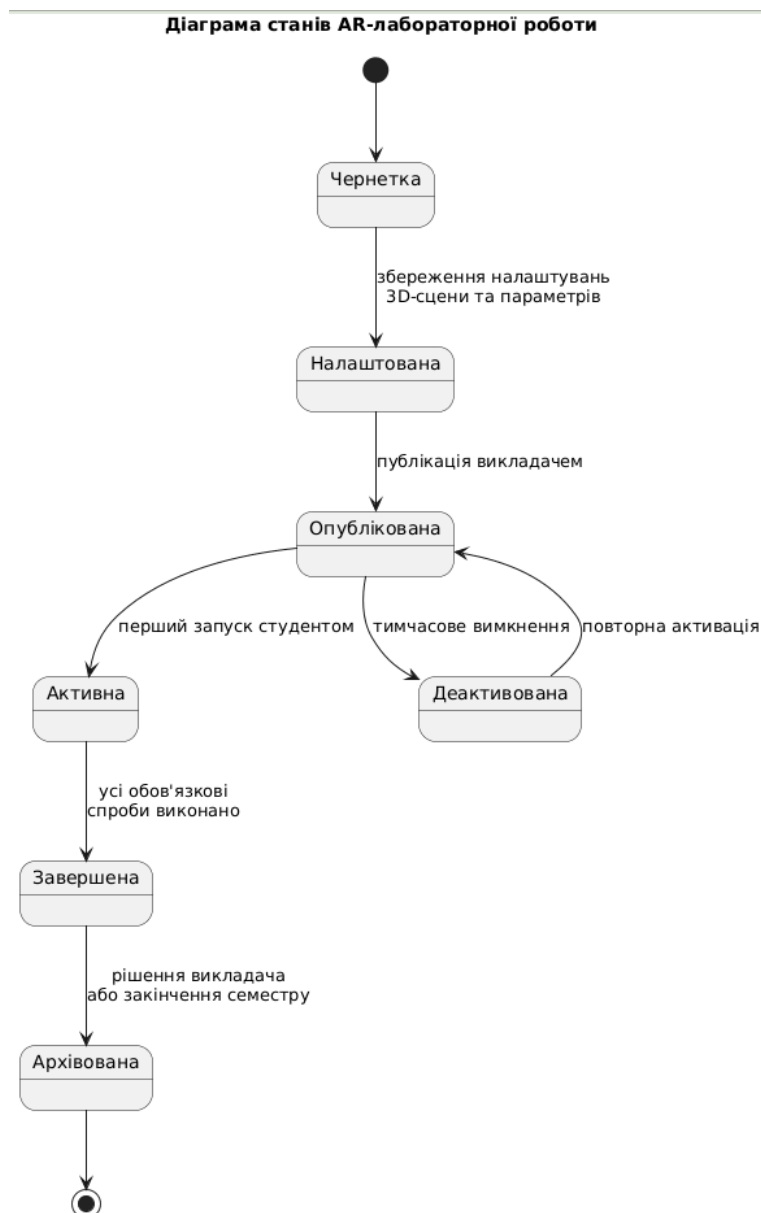


Рисунок 3.3 – Діаграма станів AR-лабораторної роботи

Діаграма відображає послідовну зміну станів лабораторної роботи від моменту її створення викладачем до завершення та архівації.

Початковим станом є «Чернетка», у якому лабораторна робота створюється викладачем і ще не доступна студентам. На цьому етапі відбувається первинне формування структури лабораторної роботи, підбір 3D-моделей, налаштування AR-сцени та визначення основних параметрів симуляції.

Після збереження налаштувань 3D-сцени та параметрів лабораторна робота переходить у стан «Налаштована». У цьому стані вона повністю

підготовлена з технічної та методичної точки зору, однак все ще недоступна для виконання студентами.

Наступним етапом є «Опублікована», який настає після публікації лабораторної роботи викладачем. У цьому стані лабораторна робота стає доступною в межах відповідного курсу, але ще не є активною для виконання.

Після першого запуску лабораторної роботи студентом вона переходить у стан «Активна». У цьому стані студенти можуть виконувати AR-лабораторну роботу, взаємодіяти з тривимірними об'єктами, змінювати параметри симуляції та отримувати результати експериментів. Стан «Активна» є ключовим з точки зору навчального процесу, оскільки саме на цьому етапі відбувається практичне засвоєння навчального матеріалу.

У разі необхідності лабораторна робота може бути тимчасово переведена у стан «Деактивована» (наприклад, для внесення змін, усунення помилок або на час перерви в навчальному процесі). Після повторної активації вона знову повертається до стану «Опублікована».

Після виконання студентом усіх обов'язкових спроб і завершення лабораторної роботи вона переходить у стан «Завершена». У цьому стані результати фіксуються в системі, а викладач може здійснити перевірку та оцінювання виконаних завдань.

Завершальним етапом життєвого циклу є стан «Архівована», який настає за рішенням викладача або після закінчення семестру. Архівована лабораторна робота зберігається в системі для історії та аналітики, але недоступна для повторного виконання студентами.

Таким чином, діаграма станів дозволяє наочно відобразити логіку керування AR-лабораторними роботами, забезпечує контроль їх життєвого циклу та підтримує гнучке адміністрування навчального процесу в інтерактивній платформі онлайн-освіти.

3.3 Проєктування бази даних інтерактивної платформи

База даних інтерактивної платформи онлайн-освіти з використанням 3D-моделей і технологій доповненої реальності проєктувалась з урахуванням таких основних вимог:

- підтримка різних ролей користувачів (студент, викладач, адміністратор);
- зберігання навчальних курсів, матеріалів і лабораторних робіт;
- інтеграція AR/3D-контенту та параметрів інтерактивних сцен;
- фіксація спроб виконання лабораторних робіт і результатів навчання;
- забезпечення масштабованості та цілісності даних.

Для цього була обрана реляційна модель даних, що дозволяє формалізувати зв'язки між сутностями, мінімізувати дублювання інформації та забезпечити підтримку складних запитів аналітики.

Було спроектована ER-діаграма (рис. 3.4).

Сутність USER містить облікові записи всіх користувачів платформи та включає ідентифікаційні дані, електронну адресу, хеш пароля, персональну інформацію та статус активності.

Сутність ROLE визначає типи ролей у системі (студент, викладач, адміністратор). Зв'язок між користувачами та ролями реалізований через проміжну таблицю USER_ROLE, що дозволяє одному користувачу мати декілька ролей, забезпечуючи гнучку модель керування доступом.

Сутність COURSE описує навчальні курси, включаючи код курсу, назву, опис, викладача та статус публікації.

Запис студентів на курси реалізується через сутність ENROLLMENT, яка фіксує факт реєстрації користувача на курс, його поточний статус та дату зарахування.

Навчальні матеріали зберігаються в сутності COURSE_MATERIAL, яка підтримує різні типи контенту (текстові матеріали, відео, посилання, 3D-

ресурси). Ознака AR-орієнтованості дозволяє ідентифікувати матеріали, що використовуються у доповненій реальності.

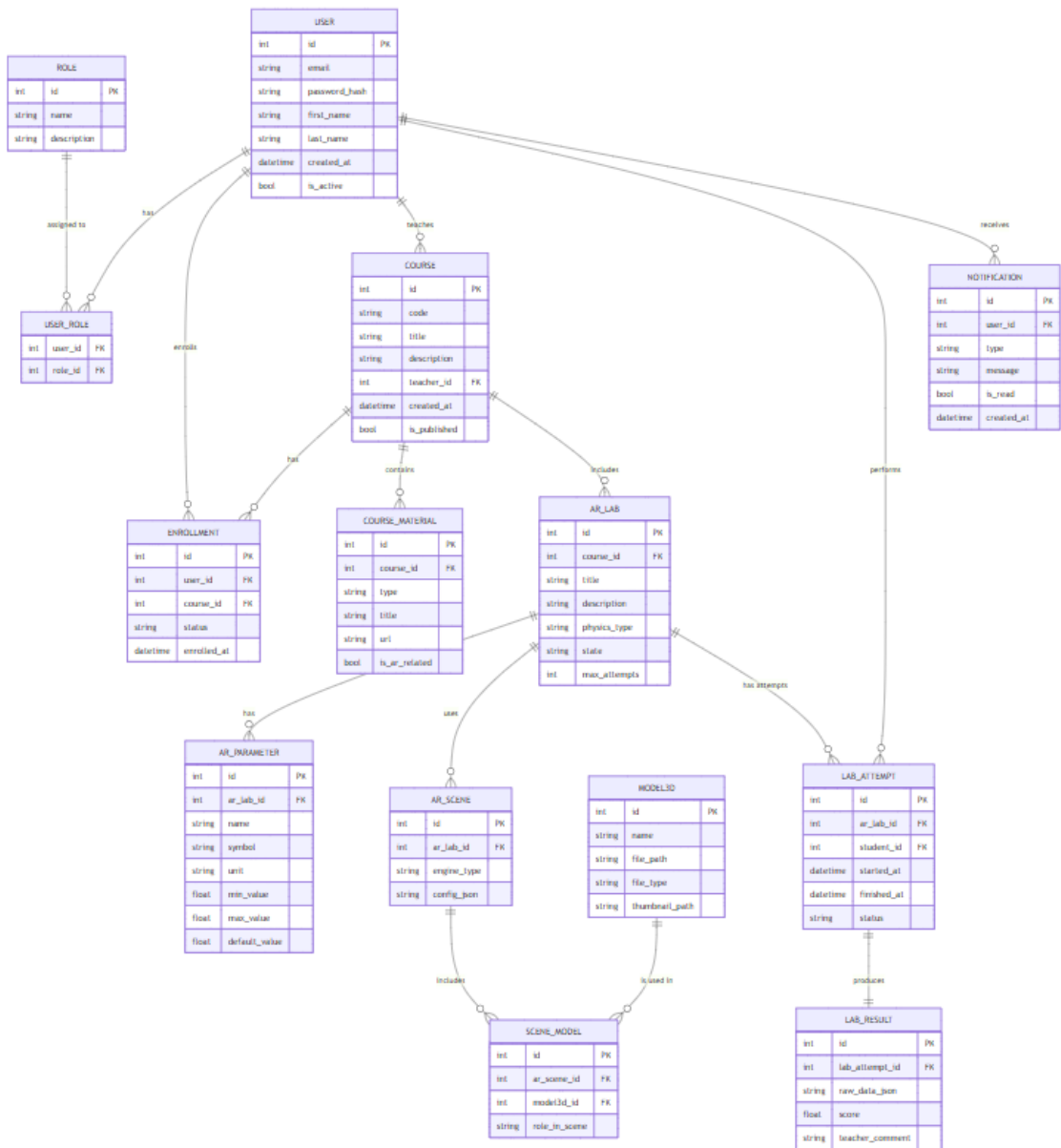


Рисунок 3.4 – ER-діаграма бази даних платформи

AR-лабораторні роботи та інтерактивні сцени.

Сутність AR_LAB є ключовою для реалізації лабораторних робіт у доповненій реальності. Вона містить опис лабораторної роботи, тип фізичної

моделі, поточний стан (чернетка, опублікована, активна тощо) та обмеження на кількість спроб.

Кожна лабораторна робота може містити набір параметрів, які змінюються користувачем під час симуляції. Ці параметри описані у сутності AR_PARAMETER, що зберігає діапазони допустимих значень і значення за замовчуванням.

Візуальна частина AR-лабораторії реалізується через сутність AR_SCENE, яка містить конфігурацію сцени у форматі JSON та тип використовуваного 3D/AR-рушія.

Сутність MODEL_3D використовується для зберігання метаданих 3D-моделей, включаючи формат файлу, шляхи до ресурсів і мініатюр.

Зв'язок між сценами та 3D-моделями реалізується через таблицю SCENE_MODEL, що дозволяє використовувати одну модель у кількох сценах та визначати її роль у конкретній AR-сцені.

Сутність LAB_ATTEMPT фіксує кожну спробу виконання AR-лабораторної роботи студентом, включаючи час початку і завершення, а також поточний статус.

Результати виконання зберігаються у сутності LAB_RESULT, яка містить числову оцінку, експериментальні дані та коментарі викладача. Така структура забезпечує прозорий механізм оцінювання та подальшого аналізу навчальних досягнень.

Сутність NOTIFICATION використовується для реалізації системи зворотного зв'язку та інформування користувачів про результати, зміни статусів лабораторних робіт або курсів.

Спроектowana база даних забезпечує:

- цілісне зберігання навчального та AR/3D-контенту;
- підтримку складних сценаріїв взаємодії користувачів з лабораторними роботами;
- можливість масштабування платформи та розширення функціональності;

- основу для аналітики успішності та адаптивного навчання.

Запропонована ER-модель є придатною для реалізації у сучасних СУБД та відповідає вимогам до інформаційних систем інтерактивної онлайн-освіти з використанням доповненої реальності.

3.4 Проєктування класів інтерактивної платформи

Проєктування класів інтерактивної платформи онлайн-освіти з використанням 3D-моделей та доповненої реальності виконувалося з урахуванням принципів об'єктно-орієнтованого програмування, зокрема:

- інкапсуляції – приховування внутрішньої реалізації;
- абстракції – виділення суттєвих характеристик предметної області;
- наслідування – повторне використання спільної логіки;
- поліморфізму – уніфікована взаємодія з різними типами об'єктів [21].

Модель класів відображає логічну структуру платформи та безпосередньо пов'язана з раніше спроектованою ER-діаграмою бази даних і архітектурою прикладної логіки (рис. 3.5).

Клас User представляє користувача системи. Використовується для всіх ролей (студент, викладач, адміністратор).

Основні атрибути:

- id – унікальний ідентифікатор;
- email, passwordHash – облікові дані;
- firstName, lastName – персональні дані;
- status – стан облікового запису.

Основні методи:

- authenticate() – автентифікація користувача;
- logout() – завершення сесії.

Клас Role визначає роль користувача в системі (Student, Teacher, Admin). Реалізує рольову модель доступу (RBAC).

Клас Course описує навчальний курс, який містить навчальні матеріали та AR-лабораторні роботи.

Методи:

- publish() – публікація курсу;
- archive() – архівація курсу.

Клас ARLab – ключовий клас системи, що представляє лабораторну роботу з підтримкою AR/3D.

Особливості:

- пов’язаний з курсом;
- має обмеження на кількість спроб;
- керує станами (чернетка, активна, завершена).

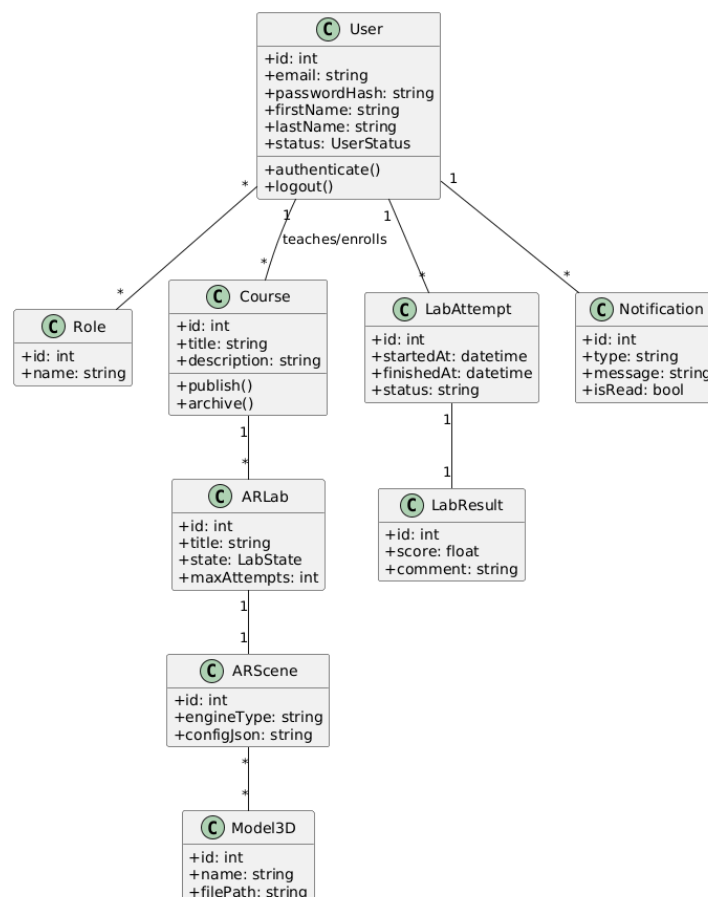


Рисунок 3.5 – Діаграма класів платформи

Клас ARScene відповідає за AR-сцену лабораторної роботи.

Функції:

- збереження конфігурації сцени;
- ініціалізація 3D-оточення.

Клас Model3D описує 3D-модель, яка використовується в AR-сценах.

Клас LabAttempt фіксує спробу виконання лабораторної роботи студентом.

Містить:

- часові мітки;
- статус виконання;
- зв'язок із результатами.

Клас LabResult містить підсумкові результати виконання лабораторної роботи.

Клас Notification використовується для реалізації механізму зворотного зв'язку та сповіщень.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРАКТИВНОЇ ПЛАТФОРМИ

4.1 Обґрунтування вибору технологій

Метою програмної реалізації є створення прототипу інтерактивної платформи онлайн-освіти, який забезпечує:

- завантаження, зберігання та відображення 3D-об'єктів освітнього призначення;
- взаємодію користувача з 3D-сценою (обертання, масштабування, переміщення, вибір об'єктів, зміна параметрів);
- реалізацію функціональних модулів керування курсами, користувачами, тестуванням і комунікацією.

React доцільний для побудови SPA-інтерфейсу з компонентною структурою, повторним використанням UI-елементів та зручним керуванням станом. Це критично для кабінетів студента/викладача та адмін-панелі [22].

Three.js забезпечує відображення та інтерактивність 3D-контенту у браузері на основі WebGL, що відповідає вимозі доступності без складного встановлення ПЗ [23].

NestJS (Node.js) обрано як серверний фреймворк із модульною архітектурою, підтримкою DI (Dependency Injection), валідації DTO, guards/roles, зручним структуруванням доменної логіки та API [24].

PostgreSQL використано як надійну реляційну СУБД для зберігання навчальних даних (користувачі, курси, матеріали, спроби, результати), забезпечуючи цілісність через ключі та обмеження [25].

Для 3D-ресурсів застосовується файлове сховище, у БД зберігаються метадані (шлях, формат, прев'ю, зв'язки з AR-лабами).

Таким чином, прототип побудовано на стеку: React, Three.js, NestJS, PostgreSQL.

4.2 Структура програмного прототипу

Прототип реалізовано як клієнт-серверну систему.

Front-end (React):

- модуль автентифікації/авторизації (JWT);
- кабінет студента (курси, матеріали, AR/3D-лаби, результати);
- кабінет викладача (створення курсів/лаб, аналіз успішності);
- адмін-панель (користувачі, ролі, контент, журнали подій);
- 3D-переглядач (Three.js) як окремий компонент.

Back-end (NestJS):

- модулі: Auth, Users, Courses, Materials, Labs, Attempts/Results, Tests, Notifications;

- REST API для SPA;
- інтеграція з PostgreSQL через ORM.

Database (PostgreSQL):

- сутності: User/Role/UserRole, Course, Enrollment, CourseMaterial, ARLab, ARScene, Model3D, LabAttempt, LabResult, Notification тощо.

Завантаження 3D-ресурсів.

Функціонально прототип підтримує завантаження моделей викладачем або адміністратором. На сервері файл зберігається у сховищі, а в БД фіксуються метадані (назва, формат, шлях, прив'язка до лабораторної).

Фрагмент програмної реалізації контролера завантаження 3D-моделей наведено у лістингу 4.1.

Лістинг 4.1 – Фрагмент коду контролера завантаження 3D-моделей

```
// models.controller.ts
import { Controller, Post, UseInterceptors, UploadedFile,
Body } from '@nestjs/common';
import { FileInterceptor } from '@nestjs/platform-express';
@Controller('models')
export class ModelsController {
```

```

    constructor(private readonly modelsService: ModelsService)
    {}

    @Post('upload')
    @UseInterceptors(FileInterceptor('file'))
    async uploadModel(
        @UploadedFile() file: Express.Multer.File,
        @Body('name') name: string,
    ) {
        // file.path -> шлях у файловому сховищі
        return this.modelsService.create({
            name,
            filePath: file.path,
            fileType: file.mimetype,
        });
    }
}

```

Для забезпечення інтерактивної роботи з тривимірними освітніми об'єктами на клієнтській стороні платформи реалізовано компонент 3D-переглядача з використанням бібліотеки Three.js, інтегрованої у середовище React. Даний компонент відповідає за ініціалізацію та відображення 3D-сцени безпосередньо у веб-браузері користувача.

Функціональність компонента включає:

- створення та налаштування сцени, камери та джерел освітлення;
- завантаження тривимірних моделей у форматах GLB/GLTF, що є стандартними для веб- та AR-застосунків;
- підключення механізмів керування камерою (OrbitControls), які дозволяють користувачу обертати, масштабувати та переміщувати сцену;
- реалізацію базової взаємодії з об'єктами сцени, зокрема вибір 3D-об'єктів за допомогою кліку.

Використання підходу React та Three.js забезпечує модульність клієнтського коду, спрощує інтеграцію 3D-візуалізацій з іншими

компонентами інтерфейсу платформи та створює основу для подальшого розширення функціональності, зокрема підтримки AR-сцен через WebXR.

У лістингу 4.2 наведено реалізацію клієнтського модуля відображення та взаємодії з тривимірними моделями в інтерактивній онлайн-освітній платформі.

Лістинг 4.2 – Фрагмент клієнтського коду відображення 3D-сцени з використанням React та Three.js

```
// ModelViewer.jsx
import { useEffect, useRef } from "react";
import * as THREE from "three";
import { OrbitControls } from
"three/examples/jsm/controls/OrbitControls.js";
import { GLTFLoader } from
"three/examples/jsm/loaders/GLTFLoader.js";
export function ModelViewer({ modelUrl }) {
  const mountRef = useRef(null);
  useEffect(() => {
    const mount = mountRef.current;
    const scene = new THREE.Scene();
    const camera = new THREE.PerspectiveCamera(
      60,
      mount.clientWidth / mount.clientHeight,
      0.1,
      1000
    );
    camera.position.set(0, 1.2, 2.5);
    const renderer = new THREE.WebGLRenderer({ antialias:
true });
    renderer.setSize(mount.clientWidth, mount.clientHeight);
    mount.appendChild(renderer.domElement);
    // світло
    scene.add(new THREE.AmbientLight(0xffffff, 0.6));
    const dir = new THREE.DirectionalLight(0xffffff, 0.8);
```

```

    dir.position.set(2, 4, 2);
    scene.add(dir);
    // керування камерою
    const controls = new OrbitControls(camera,
renderer.domElement);
    controls.enableDamping = true;
    // завантаження GLTF/GLB
    const loader = new GLTFLoader();
    loader.load(modelUrl, (gltf) => {
        const obj = gltf.scene;
        obj.name = "edu-model";
        scene.add(obj);
    });
    // взаємодія: вибір об'єкта
    const raycaster = new THREE.Raycaster();
    const mouse = new THREE.Vector2();
    const onClick = (e) => {
        const rect =
renderer.domElement.getBoundingClientRect();
        mouse.x = ((e.clientX - rect.left) / rect.width) * 2 -
1;
        mouse.y = -(((e.clientY - rect.top) / rect.height) * 2 -
1);
        raycaster.setFromCamera(mouse, camera);
        const hits =
raycaster.intersectObjects(scene.children, true);
        if (hits.length) {
            const picked = hits[0].object;
            picked.material = picked.material?.clone?.() ??
picked.material;
            if (picked.material) picked.material.wireframe =
true;
        }
    };
    renderer.domElement.addEventListener("click", onClick);
    let raf = 0;
    const animate = () => {
        controls.update();
        renderer.render(scene, camera);
    };

```

```

        raf = requestAnimationFrame(animate);
    };
    animate();
    return () => {
        cancelAnimationFrame(raf);
        renderer.domElement.removeEventListener("click",
onClick);
        mount.removeChild(renderer.domElement);
        renderer.dispose();
    };
    }, [modelUrl]);
    return <div ref={mountRef} style={{ width: "100%", height:
"480px" }} />;
}

```

Таким чином виконано ключову вимогу прототипу: 3D-контент завантажується та відображається в браузері, а також підтримується базова інтерактивність, необхідна для навчальних сценаріїв (огляд, вибір, модифікація параметрів).

4.3 Реалізація функціональних модулів платформи

Модуль керування користувачами та ролями. У NestJS реалізовано RBAC-підхід (role-based access control): Студент, Викладач, Адміністратор. Доступ до ендпоїнтів контролюється guards та декораторами ролей (лістинг 4.3).

Лістинг 4.3 – Обмеження доступу до створення курсу

```

// courses.controller.ts
import { Controller, Post, Body, UseGuards } from
'@nestjs/common';
@Controller('courses')
export class CoursesController {

```

```

        constructor(private readonly coursesService:
CoursesService) {}

    @Post()
    @UseGuards(JwtAuthGuard, RolesGuard)
    @Roles('TEACHER', 'ADMIN')
    create(@Body() dto: CreateCourseDto) {
        return this.coursesService.create(dto);
    }
}

```

Модуль керування курсами та матеріалами (лістинг 4.4).

Функції:

- створення/редагування курсу (викладач/адмін);
- публікація курсу;
- додавання навчальних матеріалів (текст/посилання/файли/3D-пов'язані матеріали);
- запис студента на курс та перегляд прогресу.

Лістинг 4.4 – Отримання списку курсів

```

// api/courses.js
export async function fetchCourses(token) {
    const res = await fetch("/api/courses", {
        headers: { Authorization: `Bearer ${token}` }
    });
    if (!res.ok) throw new Error("Не вдалося завантажити
курси");
    return res.json();
}

```

Модуль AR/3D-лабораторних робіт та результатів.

Реалізовано сценарій:

- студент відкриває лабораторну, отримує конфігурацію сцени та параметри;

- виконує взаємодію/симуляцію у 3D-переглядачі;
- надсилає результати;
- система зберігає спробу та формує результат для перегляду викладачем (лістинг 4.5).

Лістинг 4.5 – Збереження спроби

```
// attempts.controller.ts
import { Controller, Post, Body, UseGuards } from
 '@nestjs/common';

@Controller('attempts')
export class AttemptsController {
  constructor(private readonly attemptsService:
 AttemptsService) {}

  @Post()
  @UseGuards(JwtAuthGuard)
  createAttempt(@Body() dto: CreateAttemptDto) {
    // dto: { arLabId, studentId, rawDataJson, status }
    return this.attemptsService.create(dto);
  }
}
```

Модуль комунікації та сповіщень. Для прототипу реалізовано внутрішні повідомлення платформи (лістинг 4.6):

- сповіщення про публікацію курсу/лаби;
- повідомлення про перевірку роботи/оновлення оцінки;
- системні повідомлення.

Лістинг 4.6 – Створення сповіщення

```
// notifications.service.ts
async create(userId: number, type: string, message: string)
{
```

```

return this.repo.save({
    userId,
    type,
    message,
    isRead: false,
    createdAt: new Date(),
});
}

```

У межах розділу реалізовано програмний прототип платформи, який:

- забезпечує завантаження 3D-ресурсів, їх зберігання та відображення у браузері;
- підтримує інтерактивну взаємодію з 3D-сценою (керування камерою, вибір об'єктів, базова модифікація);
- містить ключові модулі платформи: керування курсами, користувачами та ролями, тестування, комунікація/сповіщення, а також реєстрацію спроб і результатів лабораторних робіт.

4.4 Приклад реалізації AR-лабораторних робіт з використанням 3D-моделей

У межах магістерської кваліфікаційної роботи розглянуто приклади AR-лабораторних робіт з фізики, які використовуються для демонстрації можливостей розробленої інтерактивної онлайн-платформи щодо інтеграції тривимірних моделей, математичних залежностей та параметризованих симуляцій.

Зокрема, як демонстраційні сценарії обрано лабораторну роботу з визначення моменту інерції маятника Обербека та дослідження явища інтерференції світла від двох когерентних джерел за методом Юнга. Вибір зазначених робіт зумовлений їхньою наочністю, наявністю формалізованих

математичних моделей та можливістю інтерактивної візуалізації фізичних процесів.

Розроблена платформа забезпечує можливість задання користувачем параметрів моделі у визначених діапазонах, виконання необхідних обчислень у реальному часі та відображення результатів у вигляді інтерактивних 3D- і AR-сцен. Зміна параметрів безпосередньо впливає на поведінку моделі, що дозволяє студенту досліджувати вплив фізичних величин на результат експерименту.

Для лабораторної роботи з маятником Обербека реалізовано параметризовану 3D-модель установки, яка дає змогу змінювати положення вантажів та масу підвісного тіла, а також візуалізувати процес обертального руху. Платформа підтримує обчислення теоретичних і експериментальних значень моменту інерції з подальшим збереженням результатів у базі даних.

У випадку дослідження інтерференції світла за методом Юнга платформа дозволяє змінювати довжину хвилі, відстань між щілинами та відстань до екрана, що супроводжується динамічним формуванням інтерференційної картини. Візуалізація результатів сприяє кращому розумінню хвильових явищ і закономірностей інтерференції.

Таким чином, наведені приклади підтверджують можливість використання розробленої платформи для створення інтерактивних AR-лабораторних робіт, які поєднують теоретичні знання, математичне моделювання та наочну візуалізацію, підвищуючи ефективність і практичну спрямованість онлайн-навчання.

Проектування тривимірних елементів здійснювалося з використанням програмного середовища Blender. Дане середовище широко застосовується для створення тривимірних моделей, анімації, візуальних ефектів, інтерактивних застосунків, а також проєктів у сфері віртуальної та доповненої реальності. Використання Blender у межах цієї роботи зумовлене його функціональністю, кросплатформеністю та підтримкою сучасних форматів 3D-моделей, придатних для інтеграції у веб- та AR-середовища.

У процесі реалізації прототипу платформи було розроблено низку навчальних 3D-об'єктів, необхідних для відтворення AR-лабораторних робіт з фізики. Зокрема, створено модель джерела світла, модель світлової перешкоди з двома прорізами для демонстрації явища інтерференції, а також модель маятника Обербека з елементами вимірювання часу. Розроблені об'єкти використовуються як складові інтерактивних 3D- та AR-сцен і забезпечують наочну візуалізацію фізичних процесів у межах навчального сценарію платформи (рис. 4.1).

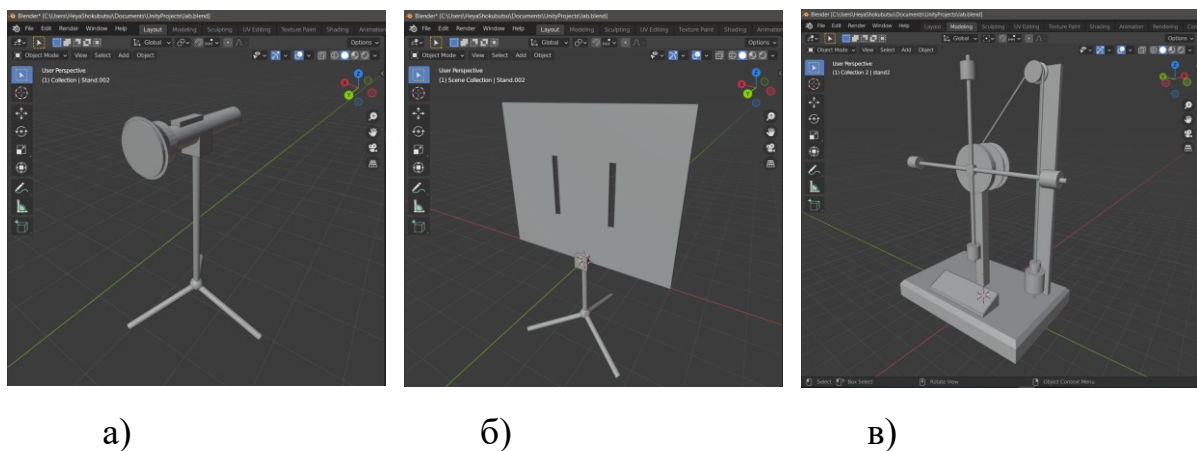
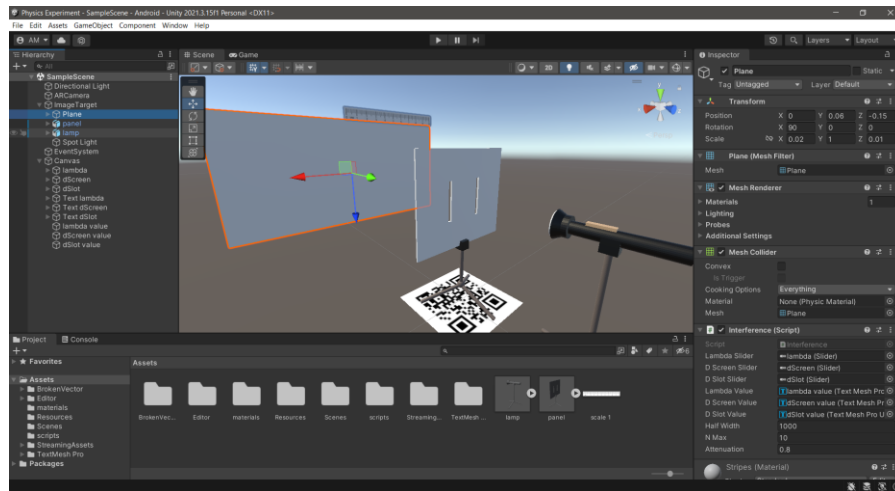


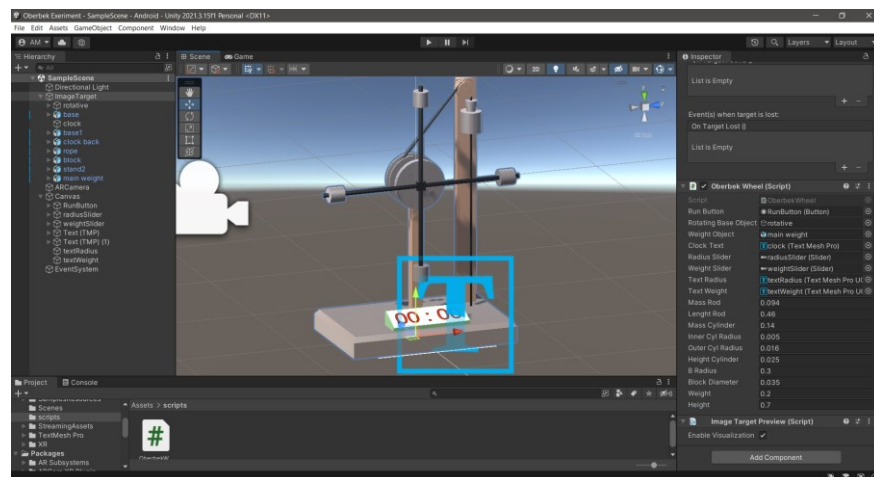
Рисунок 4.1 – Тривимірні моделі, розроблені у середовищі Blender: модель джерела світла (а), модель оптичної перешкоди з двома щілинами (б), а також модель маятника Обербека з віртуальним секундоміром (в)

Реалізацію компонентів доповненої реальності виконано з використанням ігрового рушія Unity, який є одним із найбільш поширених середовищ для розробки інтерактивних 3D- та AR-застосунків і забезпечує підтримку технологій відстеження графічних маркерів та просторової прив'язки об'єктів. У межах розробленого прототипу було створено AR-сцени для візуалізації фізичних процесів: експеримент з інтерференції світла за методом Юнга, представлений на рисунку 4.2а, а також модель маятника Обербека, показану на рисунку 4.2б. Зазначені сцени демонструють можливості платформи щодо поєднання тривимірних моделей із механізмами

доповненої реальності та інтерактивною взаємодією користувача з навчальним контентом.



a)



б)

Рисунок 4.2 – Дослід інтерференції світла за методом Юнга (а) та маятник Обербека (б), реалізовані в середовищі Unity

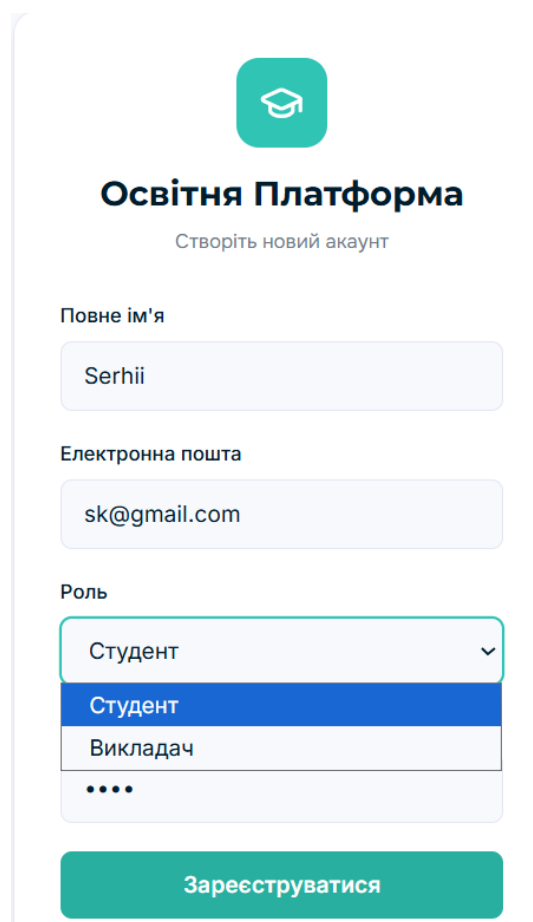
Таким чином, реалізовані AR-сцени в середовищі Unity демонструють можливості платформи щодо інтеграції 3D-моделей і доповненої реальності для наочного та інтерактивного виконання навчальних лабораторних робіт.

4.5 Приклад використання та функціонування розробленої платформи

Розроблена інтерактивна онлайн-платформа призначена для підтримки освітнього процесу з використанням тривимірних моделей і технологій доповненої реальності. Платформа реалізує рольову модель доступу та забезпечує окремі користувацькі інтерфейси для студентів, викладачів і адміністраторів системи.

Робота з платформою починається з процесу автентифікації.

Якщо у користувача немає облікового запису, необхідно спочатку зареєструватися (рис. 4.3). Під час реєстрації користувач обирає роль «Студент» або «Викладач», що визначає подальший функціональний доступ до системи.



The image shows a registration form for the 'Освітня Платформа' (Educational Platform). At the top is a green square icon with a white graduation cap. Below it, the title 'Освітня Платформа' is displayed in bold, followed by the subtitle 'Створіть новий акаунт' (Create a new account). The form contains three input fields: 'Повне ім'я' (Full name) with the value 'Serhii', 'Електронна пошта' (Email) with the value 'sk@gmail.com', and 'Роль' (Role) which is a dropdown menu currently showing 'Студент' (Student). The dropdown menu is open, showing 'Студент' and 'Викладач' (Lecturer) as options, with 'Студент' highlighted in blue. Below the dropdown is a button labeled 'Зареєструватися' (Register).

Рисунок 4.3 – Реєстрація користувача

Після реєстрації користувачі мають можливість увійти до системи за допомогою електронної пошти та пароля або створити новий обліковий запис (рис. 4.4).

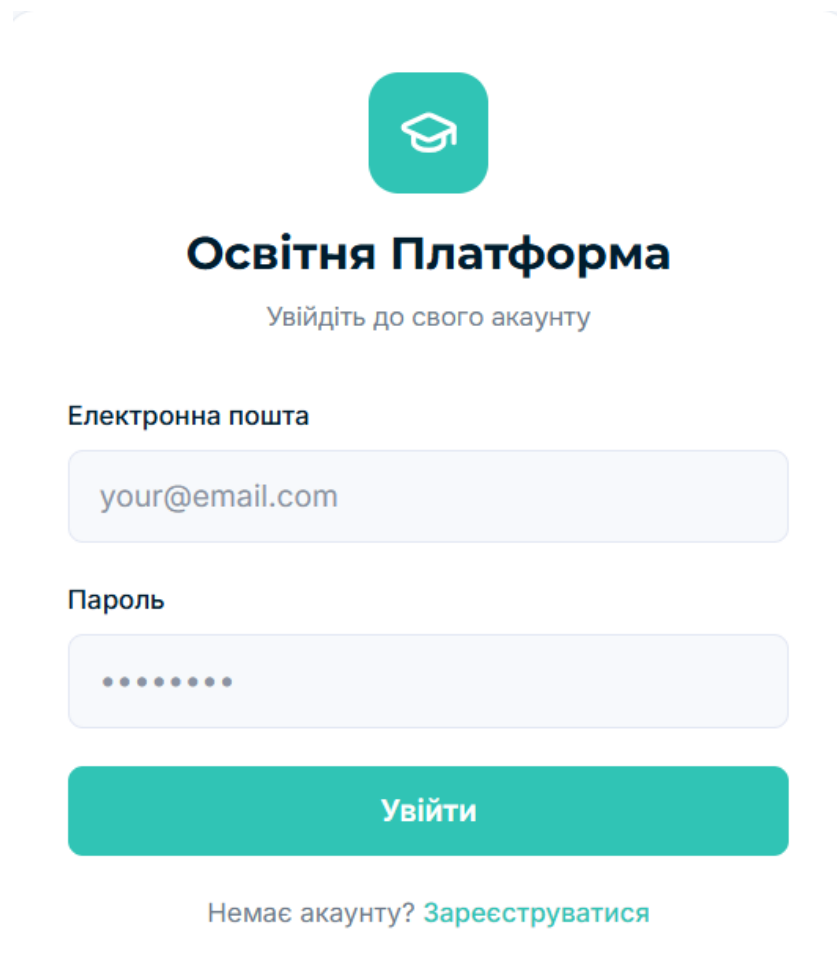


Рисунок 4.4 – Автентифікація користувача

Механізм автентифікації реалізований з використанням захищених сесій і токенів доступу, що забезпечує безпеку персональних даних та контроль доступу до навчального контенту.

Після успішного входу студент потрапляє до головної панелі користувача, яка відображає узагальнену інформацію про навчальну активність, зокрема (рис. 4.5):

- кількість доступних та активних курсів;
- кількість лабораторних робіт;

- середній бал успішності;
- статус виконання тестів і практичних завдань.

Студент має можливість:

- переглядати доступні курси та записуватися на них;
- відкривати навчальні матеріали курсу;
- виконувати лабораторні роботи, у тому числі з використанням 3D-моделей і AR-сцен;
- змінювати параметри лабораторної роботи в межах, визначених викладачем;
- переглядати результати виконання та отримувати зворотний зв'язок.

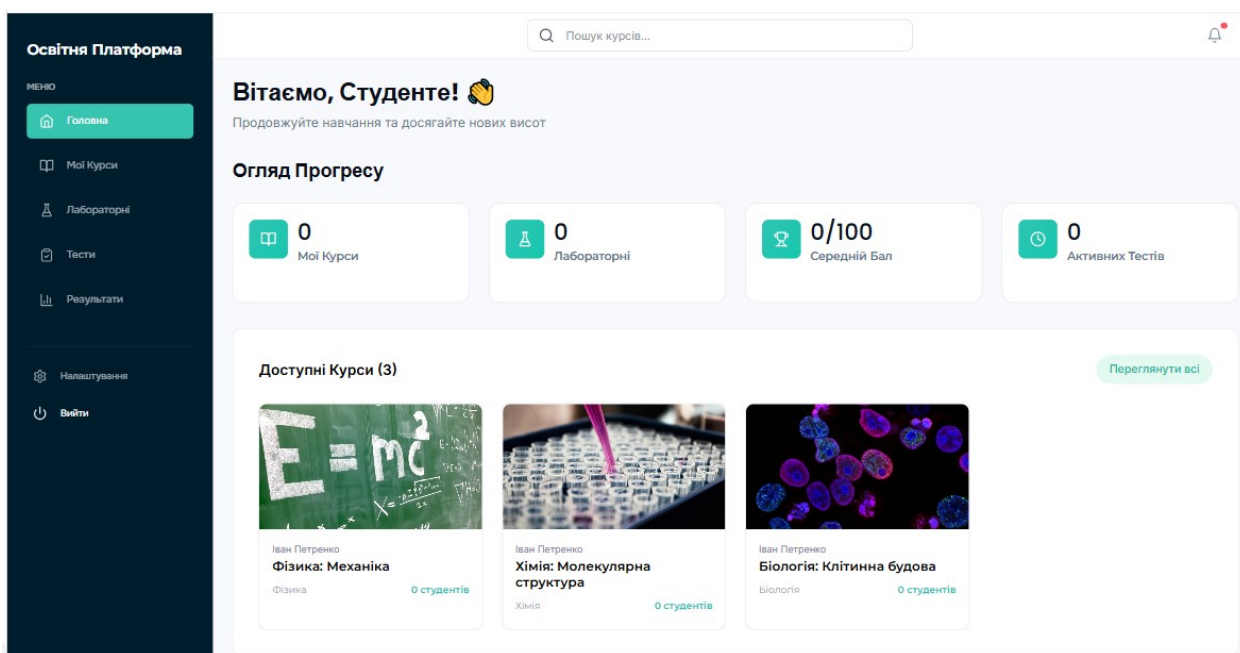


Рисунок 4.5 – Інтерфейс кабінета студента

Окремий розділ платформи присвячений лабораторним роботам, де відображається їх поточний стан (не розпочато, в процесі, завершено) та отримані оцінки (рис. 4.6).

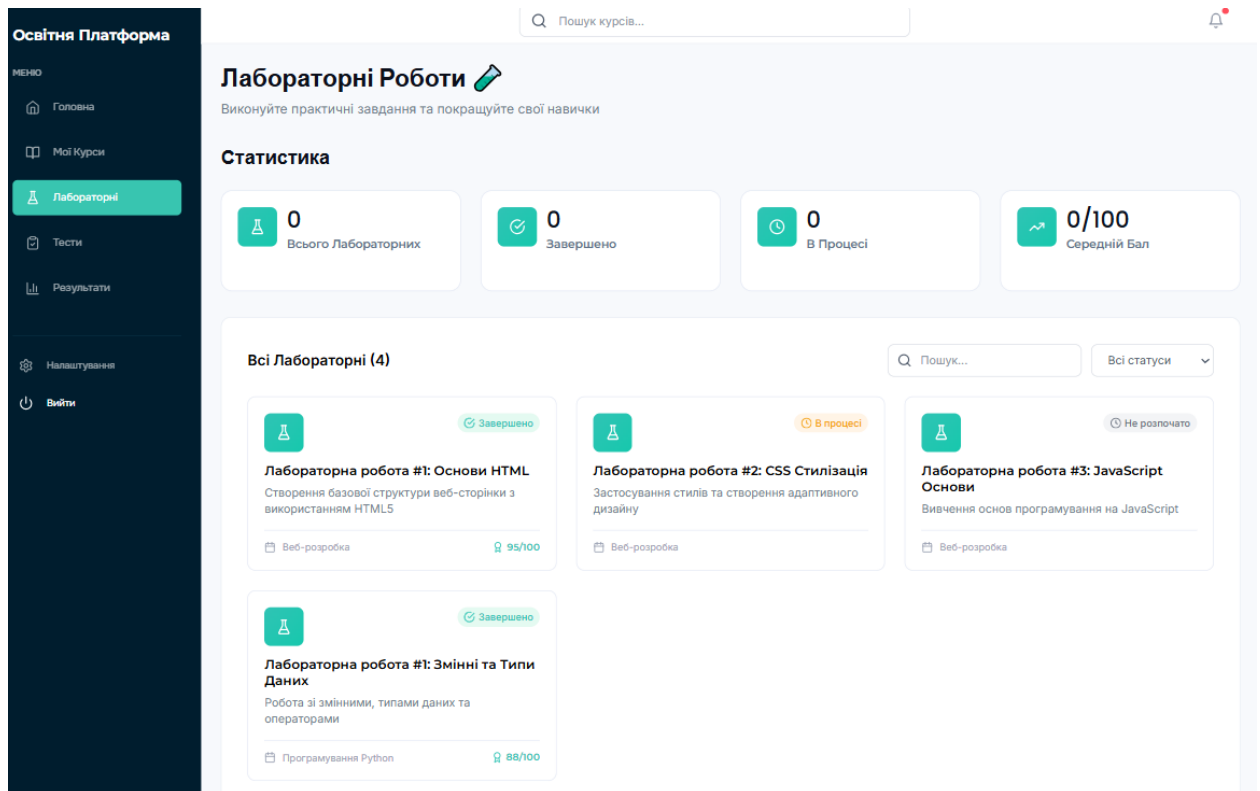


Рисунок 4.6 – Інтерфейс модуля виконання лабораторних робіт

Під час виконання лабораторної роботи студент взаємодіє з інтерактивною 3D-сценою, яка відображається у веббраузері з використанням технологій WebGL та Three.js. Платформа дозволяє:

- завантажувати тривимірні моделі;
- відображати фізичні процеси та експериментальні установки;
- змінювати параметри моделювання в реальному часі;
- візуалізувати результати експерименту.

Отримані дані зберігаються в системі та можуть бути використані для подальшого аналізу й оцінювання.

Викладач після входу до системи отримує доступ до спеціалізованої панелі керування (рис. 4.7). Основними функціями викладача є:

- створення та редагування навчальних курсів;
- додавання лабораторних робіт із підтримкою AR/3D;
- налаштування параметрів лабораторних експериментів;

- призначення завдань студентам;
- перегляд результатів виконання робіт і статистики успішності;
- виставлення оцінок і надання коментарів.

Інтерфейс викладача орієнтований на зручне керування навчальним процесом та аналіз навчальних досягнень студентів у режимі реального часу.

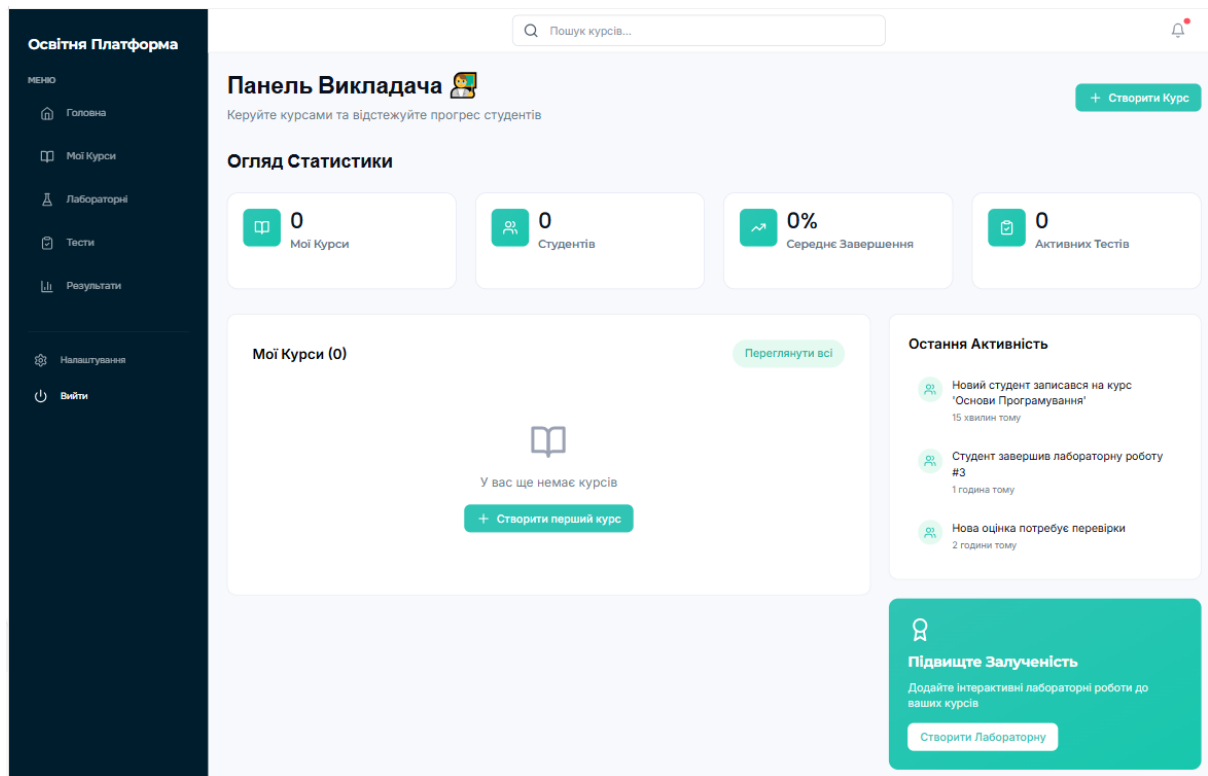


Рисунок 4.7 – Інтерфейс кабінета викладача

Адміністратор платформи відповідає за загальне керування системою. До його функцій належать:

- управління користувачами та ролями;
- адміністрування навчального контенту;
- налаштування параметрів системи;
- моніторинг активності та журналювання подій;
- забезпечення стабільної роботи платформи.

Платформа передбачає безпечне завершення сеансу користувача шляхом виходу з системи (рис. 4.8). Після завершення роботи всі сесійні дані очищуються, що запобігає несанкціонованому доступу до облікового запису.

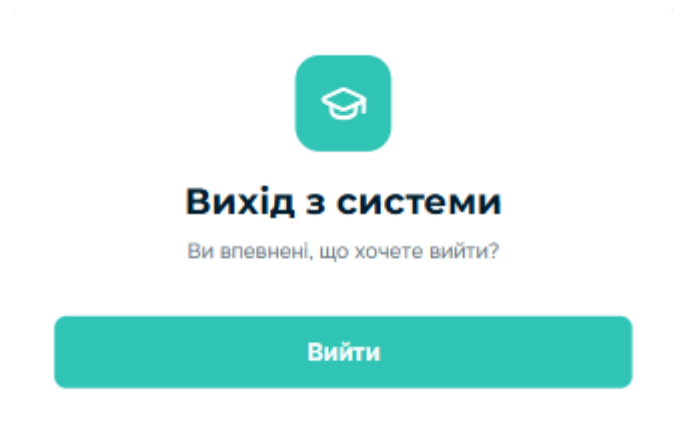


Рисунок 4.8 – Вихід з системи

Описаний механізм роботи платформи демонструє цілісний підхід до організації онлайн-освіти з використанням 3D-моделей і технологій доповненої реальності. Реалізовані користувацькі сценарії забезпечують інтерактивність, наочність та підвищення ефективності навчального процесу.

5 ТЕСТУВАННЯ ПРОГРАМНОГО ПРОТОТИПУ ІНТЕРАКТИВНОЇ ПЛАТФОРМИ

Тестування у межах магістерської роботи спрямоване на підтвердження працездатності прототипу платформи, коректності реалізованих функціональних модулів (курси, користувачі, тестування, комунікація) та стабільності інтеграції 3D-візуалізації у користувацькі сценарії.

Основні завдання тестування:

- перевірити правильність бізнес-логіки серверної частини та API-контрактів;
- підтвердити коректність доступу за ролями (студент, викладач, адміністратор);
- перевірити цілісність даних у PostgreSQL (зв'язки, каскадні дії, унікальність);
- протестувати сценарії роботи користувача у веб-інтерфейсі (React);
- оцінити стабільність відображення 3D-сцен та взаємодії з 3D-об'єктами;
- перевірити нефункціональні аспекти: продуктивність, надійність, безпека базового рівня.

Тестування програмного прототипу інтерактивної онлайн-платформи проводилось з метою перевірки коректності реалізації функціональних можливостей, надійності взаємодії компонентів системи та відповідності програмного забезпечення сформульованим вимогам. У процесі дослідження застосовано багаторівневий підхід до тестування, що охоплює модульний, інтеграційний та системний рівні, а також окрему перевірку 3D/AR-функціональності.

Модульне тестування спрямоване на перевірку окремих функцій, класів та компонентів системи в ізоляції від зовнішніх залежностей.

Для серверної частини платформи тестуванню підлягали:

- сервіси бізнес-логіки (керування курсами, лабораторними роботами, користувачами, аутентифікацією);

- допоміжні модулі та утилітарні функції;
- механізми валідації вхідних даних.

Для клієнтської частини перевірялись:

- окремі UI-компоненти (форми, сторінки, модальні вікна);
- логіка керування станом застосунку;
- допоміжні функції обробки подій та даних.

Інтеграційне тестування проводилось з метою перевірки коректності взаємодії між модулями системи. Зокрема, перевірялись:

- взаємодія серверних сервісів з базою даних PostgreSQL через ORM;
- коректність обробки HTTP-запитів REST API;
- механізми авторизації та розмежування доступу на основі ролей користувачів;

- узгодженість роботи клієнтської та серверної частин платформи.

Системне тестування орієнтоване на перевірку ключових користувацьких сценаріїв «від інтерфейсу до бази даних». У межах цього рівня було протестовано такі сценарії:

- реєстрація та автентифікація користувачів;
- доступ до навчальних курсів відповідно до ролі;
- проходження тестування студентом;
- створення та публікація навчальних матеріалів викладачем;
- завантаження та відображення 3D-моделей у лабораторних роботах;
- отримання повідомлень у внутрішній системі комунікації.

Окремий піднапрямок тестування було присвячено оцінюванню якості роботи модуля тривимірної та доповненої реальності. У рамках цієї перевірки аналізувались:

- коректність завантаження та відображення моделей у форматах GLB/GLTF;
- стабільність рендерингу та час завантаження на типових пристроях;

- реакція інтерфейсу на взаємодію користувача (обертання, масштабування, вибір об'єктів);
- узгодженість візуалізації з параметрами симуляцій.

Вхідні критерії тестування:

- клієнтська та серверна частини зібрані без помилок;
- розгорнуте тестове середовище з доступною базою даних;
- створені тестові облікові записи для основних ролей користувачів.

Критерії завершення тестування:

- успішне виконання всіх критичних сценаріїв системного тестування;
- проходження модульних та інтеграційних тестів для ключових компонентів;
- відсутність критичних дефектів, що блокують базову функціональність платформи (автентифікація, доступ до курсів, перегляд 3D-контенту, збереження результатів навчання).

Для реалізації процесу тестування використовувався типовий набір інструментів, що відповідає обраному технологічному стеку:

- Back-end (NestJS): фреймворк Jest для модульних тестів, Supertest для тестування HTTP-запитів, окрема тестова база даних PostgreSQL;
- Front-end (React): React Testing Library та Jest для перевірки компонентів інтерфейсу;
- End-to-End тестування: Playwright або Cypress для перевірки користувацьких сценаріїв у браузері;
- контроль бази даних: механізми міграцій та наповнення тестовими даними;
- 3D-модуль: базові автоматизовані смоук-тести та ручна валідація якості візуалізації.

5.1 Набір тест-кейсів для ключових сценаріїв

Тестування програмного прототипу інтерактивної онлайн-платформи є невід’ємним етапом її розробки та спрямоване на перевірку коректності реалізації функціональних і нефункціональних вимог, а також на забезпечення стабільності та надійності роботи системи. У межах роботи застосовано багаторівневий підхід до тестування, що охоплює модульне, інтеграційне та системне тестування, а також перевірку якості роботи 3D/AR-модулів.

Для оцінювання працездатності платформи було сформовано набір тест-кейсів, що відповідають основним користувацьким сценаріям та ролям системи (студент, викладач, адміністратор). Тест-кейси охоплюють процеси автентифікації, доступу до навчальних матеріалів, виконання лабораторних робіт, тестування знань та роботи з тривимірним контентом.

У таблиці 5.1 наведено приклади тестових сценаріїв, які дозволяють перевірити відповідність реалізованого прототипу визначеним вимогам платформи.

Оскільки розроблена платформа поєднує функції системи управління навчанням та інтерактивного середовища з підтримкою 3D і доповненої реальності, особливу увагу під час тестування приділено перевірці ключових користувацьких сценаріїв. Тест-кейси формуються на основі функціональних вимог та типових дій користувачів із різними ролями (студент, викладач, адміністратор). Основною метою такого тестування є підтвердження коректної роботи основних модулів системи в реальних умовах експлуатації. Кожен тест-кейс описує початкові умови, послідовність дій користувача та очікуваний результат. Це дозволяє систематизувати процес перевірки, забезпечити відтворюваність результатів і виявити потенційні помилки на ранніх етапах впровадження платформи.

Таблиця 5.1 – Приклади системних тест-кейсів

ID	Сценарій	Роль	Кроки	Очікуваний результат
ТС-01	Вхід у систему	Усі	Ввести валідні дані	Отримано токен/сесію, відкрито кабінет
ТС-02	Контроль доступу до адмін-панелі	Студент	Перейти /admin	403/редірект, доступ заборонено
ТС-03	Створення курсу	Викладач	Створити курс, зберегти	Курс з'являється у списку, запис у БД
ТС-04	Запис на курс	Студент	Відкрити курс, приєднатись	Створено зв'язок enrollment, доступ надано
ТС-05	Проходження тесту	Студент	Відкрити тест, відповісти, відправити	Збережено результат, відображено бал
ТС-06	Завантаження 3D-моделі	Викладач/Адмін	Завантажити GLB, прикріпити до лабораторної	Файл у сховищі, метадані у БД
ТС-07	Перегляд 3D-лаби	Студент	Відкрити лабораторну, завантажити сцену	Модель відображається, доступні взаємодії
ТС-08	Повідомлення студенту	Викладач	Надіслати повідомлення	Запис у таблиці messages, відображення в UI

5.2 Реалізація та приклади тестування програмних компонентів

З метою перевірки коректності роботи програмного прототипу платформи було реалізовано набір тестів різних рівнів, що охоплюють

серверну, клієнтську та 3D-частини системи. Тестування виконувалося з використанням стандартних інструментів, рекомендованих для обраного технологічного стеку (NestJS, React, Three.js).

Інтеграційне тестування серверної частини спрямоване на перевірку взаємодії HTTP-контролерів із бізнес-логікою та конфігурацією застосунку. У лістингу 5.1 наведено приклад E2E-тесту контролера курсів, який перевіряє коректність обробки запиту до REST-ендпойнта /courses.

Лістинг 5.1 – Інтеграційний тест контролера керування курсами

```
// test/courses.e2e-spec.ts
import { Test } from '@nestjs/testing';
import { INestApplication } from '@nestjs/common';
import * as request from 'supertest';
import { AppModule } from '../src/app.module';

describe('Courses (e2e)', () => {
  let app: INestApplication;

  beforeEach(async () => {
    const moduleRef = await Test.createTestingModule({
      imports: [AppModule],
    }).compile();

    app = moduleRef.createNestApplication();
    await app.init();
  });

  afterEach(async () => {
    await app.close();
  });

  it('GET /courses should return array', async () => {
    const res = await request(app.getHttpServer())
      .get('/courses')
```

```

    .expect(200);

    expect(Array.isArray(res.body)).toBe(true);
  });
});

```

Даний тест підтверджує, що сервер коректно ініціалізується, приймає HTTP-запити та повертає дані у відповідному форматі.

Модульне тестування клієнтського компонента. Для клієнтської частини платформи реалізовано модульні тести React-компонентів із використанням бібліотеки React Testing Library. У лістингу 5.2 наведено приклад тесту компонента форми входу, який перевіряє відображення елементів інтерфейсу та виклик обробника події після натискання кнопки.

Лістинг 5.2 – Модульний тест компонента автентифікації користувача

```

// src/components/LoginForm.test.tsx
import { render, screen, fireEvent } from '@testing-
library/react';
import LoginForm from './LoginForm';

test('renders login form and calls submit', async () => {
  const onSubmit = jest.fn();
  render(<LoginForm onSubmit={onSubmit} />);

  fireEvent.change(screen.getByLabelText(/email/i), {
    target: { value: 'a@b.com' }
  });

  fireEvent.change(screen.getByLabelText(/password/i), {
    target: { value: '123456' }
  });

  fireEvent.click(screen.getByRole('button', { name:
/login/i }));
  expect(onSubmit).toHaveBeenCalledTimes(1);

```

```
});
```

Тест забезпечує перевірку базової логіки взаємодії користувача з формою входу.

Для перевірки повного користувацького сценарію «від інтерфейсу до сервера» застосовано E2E-тестування за допомогою інструменту Playwright. У лістингу 5.3 наведено приклад тесту, який імітує дії користувача під час входу до системи.

Лістинг 5.3 – E2E-тест сценарію входу користувача

```
// e2e/login.spec.ts
import { test, expect } from '@playwright/test';

test('user can log in', async ({ page }) => {
  await page.goto('http://localhost:3000/login');
  await
    page.fill('input[name="email"]',
'student@test.com');
  await page.fill('input[name="password"]', 'password');
  await page.click('button[type="submit"]');

  await expect(page).toHaveURL(/dashboard/);
});
```

Даний тест підтверджує коректність роботи маршрутизації, автентифікації та клієнтсько-серверної взаємодії.

Smoke-тестування завантаження 3D-моделі. Для 3D-модуля платформи доцільно застосовувати спрощені перевірки (smoke tests), які дозволяють впевнитися у базовій працездатності механізму завантаження та додавання моделей до сцени. У лістингу 5.4 наведено приклад логічного тесту додавання об'єкта до сцени Three.js.

Лістинг 5.4 – Smoke-тест завантаження 3D-моделі у сцену Three.js

```
// src/three/loadModel.test.ts
import { Scene } from 'three';
function addMockModelToScene(scene: Scene) {
  const obj = { name: 'modelRoot' } as any;
  scene.add(obj);
  return obj;
}
test('model is added to scene', () => {
  const scene = new Scene();
  const obj = addMockModelToScene(scene);
  const found = scene.children.find(
    x => (x as any).name === 'modelRoot'
  );
  expect(found).toBeDefined();
  expect(obj).toBe(found);
});
```

Такий підхід дозволяє перевірити логіку інтеграції 3D-моделей без виконання повної візуальної валідації, що є доцільним на етапі прототипування.

У межах прототипування найбільш імовірні категорії дефектів:

- помилки контролю доступу (неправильні ролі/гварди, витік доступу до адмін-функцій);
- неконсистентність даних (видалення курсу без каскадних правил, “осиротілі” записи);
- проблеми 3D (довге завантаження GLB, “важкі” моделі, падіння FPS на слабких пристроях);
- помилки UI-логіки (стани завантаження, обробка помилок API, валідація форм).

У процесі тестування програмного прототипу платформи фіксувалися виявлені дефекти з метою їх подальшого аналізу, усунення та перевірки

повторної появи. Для систематизації результатів тестування використовується журнал дефектів (табл. 5.2).

Таблиця 5.2 – Журнал дефектів програмного прототипу платформи

ID дефекту	Модуль системи	Опис дефекту	Кроки відтворення	Очікуваний результат	Фактичний результат	Пріоритет	Статус
DF-01	Аутентифікація	Помилка при вході з некоректним email	Ввести email без «@», натиснути «Увійти»	Відображається повідомлення про помилку	Вхід не блокується	High	Виправлено
DF-02	AR-лабораторія	3D-модель не завантажується з першої спроби	Відкрити AR-лабу, завантажити модель	Модель відображається в сцені	Потрібно повторне оновлення	Medium	У роботі
DF-03	Курси	Некоректне відображення списку курсів	Увійти як студент, перейти до курсів	Відображається список доступних курсів	Список порожній	High	Виправлено
DF-04	Тестування	Результат тесту не зберігається	Пройти тест і натиснути «Завершити»	Результат зберігається в БД	Дані не записуються	Critical	Виправлено
DF-05	3D-переглядач	Камера некоректно масштабується	Прокрутити колесо миші	Плавне масштабування	Різке наближення	Low	Відкладено

ВИСНОВКИ

У межах магістерської кваліфікаційної роботи було розроблено та досліджено інтерактивну платформу онлайн-освіти з використанням 3D-моделей і технологій доповненої реальності, що відповідає сучасним тенденціям цифрової освіти.

Було проаналізовано предметну область онлайн-освіти, зокрема сучасні підходи до дистанційного навчання, використання інтерактивних засобів візуалізації та AR/VR-технологій у навчальному процесі. Проведений аналіз існуючих освітніх платформ показав їх обмежені можливості щодо глибокої інтерактивної взаємодії з навчальними об'єктами та недостатню інтеграцію доповненої реальності у повноцінні навчальні сценарії, що підтверджує актуальність обраної теми.

У роботі було сформульовано функціональні та нефункціональні вимоги до платформи, визначено ролі користувачів (студент, викладач, адміністратор), а також спроектовано користувацькі сценарії взаємодії із системою. На основі цих вимог розроблено багаторівневу клієнт–серверну архітектуру, придатну для інтеграції 3D-моделей та AR-механізмів, що забезпечує масштабованість, модульність і можливість подальшого розширення функціоналу.

З використанням UML-нотації було виконано проектування ключових аспектів системи, зокрема діаграм варіантів використання, послідовності, станів, класів та ER-діаграми бази даних. Це дозволило формалізувати логіку роботи платформи, життєвий цикл AR-лабораторних робіт та структуру зберігання навчальних даних і результатів користувачів.

У практичній частині роботи реалізовано програмний прототип платформи з використанням сучасного технологічного стеку. Прототип забезпечує:

- реєстрацію та автентифікацію користувачів;
- керування навчальними курсами та лабораторними роботами;

- виконання інтерактивних AR/3D-лабораторій;
- збереження та аналіз результатів навчання;
- зручні кабінети для студентів і викладачів.

Також було розглянуто питання тестування програмного забезпечення, визначено основні види тестів (модульне, інтеграційне, функціональне та користувацьке), що підтверджують коректність роботи основних модулів платформи.

Отримані результати свідчать, що поставлена мета роботи досягнута, а всі визначені завдання виконані в повному обсязі. Розроблена платформа може бути використана як основа для впровадження інтерактивних AR-лабораторій у закладах вищої освіти та має практичну цінність для підвищення якості навчального процесу.

Подальший розвиток роботи може передбачати:

- розширення набору AR-лабораторних робіт;
- підтримку повноцінних VR-сценаріїв;
- інтеграцію адаптивних та AI-орієнтованих механізмів навчання;
- оптимізацію продуктивності для мобільних пристроїв.

Таким чином, виконана магістерська робота є завершеним дослідженням і практичною реалізацією сучасної інтерактивної освітньої платформи з використанням 3D та доповненої реальності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Azuma R. A Survey of Augmented Reality // Presence: Teleoperators and Virtual Environments, 2020. Vol. 29. P. 1-38.
2. Milgram P., Kishino F. A taxonomy of mixed reality visual displays // IEICE Transactions on Information and Systems, 2021. Vol. E104-D. P. 1321-1329.
3. Bonk C. J., Graham C. R. The Handbook of Blended Learning. 2nd ed. San Francisco : Wiley, 2022. 624 p.
4. Hug T. Microlearning: Emerging Concepts, Practices and Technologies. – London : Routledge, 2021. 310 p.
5. Mulder M. Competence-based education and training // Journal of Education and Work, 2020. Vol. 33(3). P. 247-264.
6. Dahlstrom E., Brooks D. C. Online Learning and LMS Trends. EDUCAUSE Review, 2023.
7. Blender Foundation. Blender Documentation, 2024. URL: <https://docs.blender.org> (дата звернення: 05.09.2025).
8. Autodesk. Fusion 360 Documentation, 2024. URL: <https://help.autodesk.com> (дата звернення: 10.09.2025).
9. Parisi T. WebGL: Up and Running. – 2nd ed. – Sebastopol : O'Reilly Media, 2020. – 260 p.
10. Dirksen J. Learn Three.js. – 5th ed. – Birmingham : Packt Publishing, 2023. 520 p.
11. Babylon.js Team. Babylon.js Documentation, 2024. URL: <https://doc.babylonjs.com> (дата звернення: 11.09.2025).
12. Meshy AI. AI-powered 3D Model Generation. – 2024. – URL: <https://www.meshy.ai> (дата звернення: 13.09.2025).
13. Billinghurst M., Clark A., Lee G. A survey of augmented reality // Foundations and Trends in Human–Computer Interaction, 2021. Vol. 8(2-3). P. 73-272.

14. Zhou F., Duh H. B., Billinghurst M. Trends in augmented reality tracking // IEEE Computer Graphics and Applications, 2020. Vol. 40(2). P. 47-56.
15. Delightex GmbH. CoSpaces Edu Documentation. URL: <https://cospaces.io/edu> (дата звернення: 01.10.2025).
16. Mozaik Education. mozaWeb Platform Overview. URL: <https://www.mozaweb.com> (дата звернення: 01.10.2025).
17. Labster ApS. Virtual Labs for Science Education. URL: <https://www.labster.com> (дата звернення: 01.10.2025).
18. ClassVR Ltd. Immersive Learning with VR/AR. URL: <https://www.classvr.com> (дата звернення: 01.10.2025).
19. Mozilla. Mozilla Hubs Documentation. URL: <https://hubs.mozilla.com> (дата звернення: 11.10.2025).
20. Tanenbaum A. S., van Steen M. Distributed Systems: Principles and Paradigms. 3rd ed. Pearson, 2021. 740 p.
21. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston : Addison-Wesley, 2020.
22. Banks A., Porcello E. Learning React. – 3rd ed. – Sebastopol : O'Reilly Media, 2023. 350 p.
23. Mozilla Developer Network. WebGL API Reference. URL: <https://developer.mozilla.org/webgl> (дата звернення: 12.10.2025).
24. NestJS Team. NestJS Documentation. URL: <https://docs.nestjs.com> (дата звернення: 13.10.2025).
25. PostgreSQL Global Development Group. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs> (дата звернення: 01.11.2025).