

**БАКАЛАВРСЬКА РОБОТА**

**БР. ІІІ - 38.00.00.000 ІІЗ**

**Група ІІІ-21-2**

**Шеремета Антон**

**2025**

**Івано-Франківський національний технічний університет нафти і газу**

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

**Шеремета Антон Миколайович**

(прізвище, ім'я, по батькові)

УДК 004  
(індекс)

## **БАКАЛАВРСЬКА РОБОТА**

**Веб-застосунок для адміністрування інтелектуальних квестів**

(назва роботи)

**Інженерія програмного забезпечення**

(назва освітньої програми)

**121 - Інженерія програмного забезпечення**

(шифр і назва спеціальності)

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

Здобувач освітнього рівня Шеремета А.М.  
(підпис, ініціали та прізвище здобувача)

Науковий керівник Корнута Володимир Андрійович, к.т.н., доцент  
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту  
Завідувач кафедри

доц. Бандура В.В.  
(посада) (підпис) (дата) (ініціали та прізвище)

**Івано-Франківськ – 2025**

**Івано-Франківський національний технічний університет нафти і газу**

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітньо-кваліфікаційний рівень бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

**ЗАТВЕРДЖУЮ:**

Зав. кафедрою ІІЗ

доц.

В.В. Бандура

“     ”     2025 р.

## **ЗАВДАННЯ**

### **НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ**

**Шереметі Антону Миколайовичу**

(прізвище, ім'я, по-батькові)

**1. Тема проекту (роботи) “Веб-застосунок для адміністрування інтелектуальних квестів”**

керівник проекту (роботи) Корнута В.А., доцент

затвержені наказом закладу вищої освіти від “ 28 ” квітня 2025 р. № 264/7

**2. Строк подання студентом проекту (роботи) 10 червня 2025 р.**

**3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики**

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

1. Аналіз предметної області побудови веб-застосунків для навчання і тестування знань

2. Моделювання структури веб-застосунку адміністрування інтелектуальних квестів

3. Реалізація моделі сутність-зв'язок (ER Model)

4. Програмна реалізація веб-застосунку для адміністрування інтелектуальних квестів

5. Реалізація інтерфейсу користувача

**5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)**

1. Використання бібліотеки react-router-dom (рис. 1.1)

2. Використання пакету nodemon (рис. 1.2)

3. Використання Node.js та бібліотеки Express (рис. 1.3)

4. Використання Sequelize для роботи з базою даних MySQL (рис. 1.4)

5. Створення таблиці БД (рис. 1.5)

## 6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 28 квітня 2025 р.

Керівник \_\_\_\_\_

(підпис)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області побудови веб-застосунків для навчання і тестування знань	01.05.2025	виконано
2	Моделювання структури веб-застосунку адміністрування інтелектуальних квестів	23.05.2025	виконано
3	Реалізація моделі сутність-зв'язок (ER Model)	29.05.2025	виконано
4	Програмна реалізація веб-застосунку для адміністрування інтелектуальних квестів	04.06.2025	виконано
5	Реалізація інтерфейсу користувача	07.06.2025	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	10.06.2025	виконано

Студент – дипломник \_\_\_\_\_

(підпис)

Керівник роботи \_\_\_\_\_

(підпис)

## АНОТАЦІЯ

Бакалаврська робота містить 77 сторінок, 31 рисунок, список використаних джерел із 32 найменуваннями.

**Метою роботи** є створення веб-застосунку для адміністрування інтелектуальних квестів, який дозволяє організаторам (викладачам) створювати навчальні сценарії у формі квестів, забезпечувати контроль за їх проходженням, автоматизовано збирати статистику виконання та надавати зручні інтерфейси для всіх користувачів системи.

**Об'єкт дослідження:** процес організації та проведення навчальних інтелектуальних квестів і вікторин у віртуальному середовищі.

**Предмет дослідження:** методи та засоби розробки інтерактивного веб-застосунку для проведення та адміністрування освітніх інтелектуальних квестів

**Результати дослідження:** програмний засіб, що забезпечує створення, управління та оцінювання квестів із можливістю інтеграції в освітній процес.

В першому проведено аналіз предметної області, визначено функціональні вимоги та архітектуру системи, досліджено сучасні інструменти розробки.

В другому розділі запропоновано багаторівневу структуру взаємодії між користувачами (адміністратор, викладач, студент).

В третьому розділі реалізовано інтерфейс та моделі даних системи організації інтелектуальних квестів.

**Висновок:** представлено розробку веб-застосунку для адміністрування інтелектуальних квестів, орієнтованого на використання в навчальному процесі.

**КЛЮЧОВІ СЛОВА:** ВЕБ-ЗАСТОСУНОК, ІНТЕЛЕКТУАЛЬНІ КВЕСТИ, ГЕЙМІФІКАЦІЯ, НАВЧАЛЬНІ СИСТЕМИ, АДМІНІСТРУВАННЯ, REACT, NODE.JS, ТЕСТУВАННЯ ЗНАНЬ.

## ANNOTATION

The bachelor's thesis comprises 77 pages, 31 figures, and a list of 32 references.

**The purpose of the work** is to create a web application for administering intellectual quests, which allows organizers (teachers) to create educational scenarios in the form of quests, monitor their progress, automatically collect performance statistics, and provide convenient interfaces for all system users.

**The object of research** is the process of organizing and conducting educational intellectual quests and quizzes in a virtual environment.

**The subject of research** is the methods and tools for developing an interactive web application for conducting and administering educational intellectual quests.

**The results of the research** include a software tool that enables the creation, management, and evaluation of quests with the possibility of integration into the educational process.

The first chapter analyzes the subject area, defines functional requirements and system architecture, and explores modern development tools.

The second chapter proposes a multi-level interaction structure between users (administrator, teacher, student).

The third chapter implements the interface and data models for the intellectual quest organization system.

**Conclusion:** The development of a web application for administering intellectual quests, oriented for use in the educational process, is presented.

**KEYWORDS:** WEB APPLICATION, INTELLECTUAL QUESTS, GAMIFICATION, EDUCATIONAL SYSTEMS, ADMINISTRATION, REACT, NODE.JS, KNOWLEDGE TESTING.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....	8
ВСТУП .....	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПОБУДОВИ ІНТЕРАКТИВНИХ ВЕБ-ЗАСТОСУНКІВ ДЛЯ НАВЧАННЯ І ТЕСТУВАННЯ ЗНАНЬ .....	13
1.1. Концепція та архітектура веб-додатку для організації інтелектуальних квестів .....	13
1.1.1. Функціональні ролі та можливості.....	13
1.1.2. Забезпечення об'єктивності оцінювання.....	15
1.2. Актуальність розробки веб-додатку для організації інтелектуальних квестів .....	16
1.3. Системні вимоги до розробки веб-додатку.....	17
1.3.1. Апаратні вимоги .....	17
1.3.2. Програмні вимоги.....	18
1.4. Інструменти та технології розробки веб-додатку для інтелектуальних квестів .....	20
1.4.1. React .....	20
1.4.2. Node.js.....	21
1.4.3. Express.js.....	21
1.4.4. Sequelize .....	22
1.4.5. MySQL.....	23
1.4.6. PhpMyAdmin .....	24
1.4.7. XAMPP .....	25
1.5. Аналіз схожих існуючих навчальних платформ .....	26

					БР.ІІ – 38.00.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Шеремета А.			Веб-застосунок для адміністрування інтелектуальних квестів  <b>Пояснювальна записка</b>	Літ.	Арк.	Акрушіє
Перевір.		Корнута В.А.					6	
Реценз.						ІФНТУНГ ІІ-21-2		
Н. Контр.		Піх М.М.						
Затверд.		Бандура В.В.						

1.5.1. Системи управління навчанням (LMS).....	26
1.5.2. Платформи для створення та проведення онлайн-тестів/вікторин.	29
1.5.3. Платформи для проведення онлайн-курсів (MOOCs / Course Platforms) .....	32
1.5.4. Системи онлайн-прокторингу.....	35

## РОЗДІЛ 2. МОДЕЛЮВАННЯ СТРУКТУРИ ВЕБ-ЗАСТОСУНКУ

АДМІНІСТРУВАННЯ ІНТЕЛЕКТУАЛЬНИХ КВЕСТІВ ТА ВІКТОРИН ....	38
2.1. Представлення архітектури веб-застосунку .....	38
2.1.2. Роль викладача.....	40
2.1.3. Роль студента.....	40
2.2. Взаємодії користувача та системи за допомогою діаграми варіантів використання .....	41
2.3. Розробка діаграми активності .....	43
2.4. Реалізація моделі сутність-зв'язок (ER Model).....	45

## РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ

АДМІНІСТРУВАННЯ ІНТЕЛЕКТУАЛЬНИХ КВЕСТІВ .....	52
3.1. Функціональні можливості розроблюваного веб-застосунку.....	52
3.2. Початковий етап реалізації проекту .....	54
3.3. Модель бази даних.....	56
3.4. Реалізація інтерфейсу користувача.....	57
3.4.1. Інтерфейс адміністратора .....	59
3.4.2. Інтерфейс викладача .....	63
3.4.3. Інтерфейс студента.....	67

ВИСНОВКИ.....	72
---------------	----

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	74
---------------------------------------	----

## БІБЛІОГРАФІЧНА ДОВІДКА

					БР.ІІІ – 38.00.00.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ERD – Entity-Relationship Diagram (Діаграма сутність-зв'язок)

JWT – JSON Web Token (Веб-токен JSON)

LMS – Learning Management System (Система управління навчанням)

MOOC – Massive Open Online Course (Масовий відкритий онлайн-курс)

MySQL – My Structured Query Language

ORM – Object-Relational Mapping (Об'єктно-реляційне відображення)

RESTful API – Representational State Transfer Application Programming Interface (Інтерфейс прикладного програмування для передачі репрезентативного стану)

SaaS – Software as a Service (Програмне забезпечення як послуга)

SCSS – Sassy CSS (розширення CSS)

UML – Unified Modeling Language (Уніфікована мова моделювання)

UX – User Experience (Досвід користувача)

XAMPP – Cross-Platform (X), Apache, MySQL, PHP, Perl  
(кросплатформовий пакет веб-сервера)

					БР.ІП – 38.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

У сучасних умовах цифровізації освіти зростає потреба у використанні інтерактивних інструментів, які сприяють підвищенню мотивації студентів та ефективності навчального процесу. Одним із таких інструментів є інтелектуальні квести — форма навчальної діяльності, що поєднує елементи гри, тестування та проблемного навчання. Проте більшість існуючих платформ не забезпечують достатньої підтримки для організації, адміністрування та об'єктивного оцінювання таких навчальних сценаріїв.

### **Актуальність роботи**

Сучасні тенденції у сфері освіти все більше орієнтуються на використання цифрових технологій, що забезпечують інтерактивність, персоналізацію та гнучкість навчального процесу. Особливої популярності набувають гейміфіковані освітні підходи, зокрема інтелектуальні квести та вікторини, які підвищують мотивацію студентів до навчання. Проте, на сьогоднішній день бракує універсальних веб-застосунків, які б забезпечували повноцінне адміністрування, організацію та об'єктивне оцінювання результатів таких активностей. Це зумовлює необхідність створення спеціалізованих платформ, що відповідають сучасним технічним і педагогічним вимогам.

Розробка спеціалізованого веб-застосунку для адміністрування інтелектуальних квестів є актуальною як з технічної, так і з педагогічної точок зору. Такий застосунок має забезпечувати зручні інтерфейси для викладачів, студентів та адміністраторів, підтримку різноманітних типів завдань, гнучке налаштування сценаріїв проходження квестів і автоматизований збір результатів. Це дозволяє інтегрувати гейміфіковані підходи в традиційні освітні моделі та сприяти розвитку сучасного освітнього середовища.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтеграція інформаційно-комунікаційних технологій у сферу освіти сприяє трансформації традиційних підходів до навчання. Одним із перспективних напрямів є використання веб-застосунків для організації інтерактивної освітньої діяльності, зокрема інтелектуальних квестів, які поєднують навчальні завдання з елементами гри, дослідження та співпраці. Такі квести сприяють розвитку критичного мислення, аналітичних навичок та активному залученню студентів до процесу здобуття знань.

Попри популярність ігрових методів навчання, більшість існуючих освітніх платформ не адаптовані до реалізації повноцінних інтелектуальних квестів з можливістю гнучкого адміністрування та багаторівневої взаємодії. Відсутність комплексного рішення, яке б дозволяло ефективно управляти процесом створення, налаштування та оцінювання таких активностей, створює потребу в розробці спеціалізованого програмного засобу.

**Метою даної роботи** є створення веб-застосунку для адміністрування інтелектуальних квестів, який дозволяє організаторам (викладачам) створювати навчальні сценарії у формі квестів, забезпечувати контроль за їх проходженням, автоматизовано збирати статистику виконання та надавати зручні інтерфейси для всіх користувачів системи.

**Об'єкт дослідження** - процес організації та проведення навчальних інтелектуальних квестів і вікторин у віртуальному середовищі.

**Предмет дослідження** - методи та засоби розробки інтерактивного веб-застосунку для проведення та адміністрування освітніх інтелектуальних квестів.

#### **Завдання дослідження**

1. Провести аналіз предметної області та існуючих аналогів.
2. Визначити функціональні вимоги до веб-застосунку.
3. Спроекувати архітектуру та структуру системи.
4. Розробити модель бази даних та інтерфейси користувача.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

5. Реалізувати веб-застосунок із використанням сучасного стеку технологій (React, Node.js, Express.js, Sequelize, MySQL).

6. Забезпечити об'єктивність оцінювання в рамках функціоналу системи.

7. Протестувати працездатність та зручність використання застосунку.

У роботі проведено аналіз предметної області, визначено функціональні вимоги до системи, обрано сучасні інструменти розробки (React, Node.js, Express.js, Sequelize, MySQL), спроектовано архітектуру застосунку, розроблено моделі взаємодії користувачів, реалізовано повнофункціональний прототип та перевірено його працездатність у тестовому середовищі.

#### **Методи дослідження**

- Метод аналізу та синтезу;
- Моделювання (діаграми варіантів використання, діаграми активності, ER-моделі);
- Системний аналіз програмних рішень;
- Проектування архітектури інформаційних систем;
- Прототипування та розробка веб-додатків;

#### **Наукова новизна**

У роботі запропоновано та реалізовано архітектуру веб-застосунку, який поєднує гейміфікований формат навчання (інтелектуальні квести) із системою об'єктивного тестування знань. Особливістю розробки є інтеграція багаторівневого управління ролями користувачів (адміністратор, викладач, студент) та модуль оцінювання, який мінімізує вплив людського фактора.

#### **Практичне застосування**

Результати дослідження можуть бути використані навчальними закладами, викладачами та адміністраторами освітніх платформ для організації інтерактивного навчального процесу. Розроблений веб-застосунок

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

може слугувати інструментом підвищення залученості студентів, оптимізації перевірки знань і впровадження інноваційних форм навчання.

Бакалаврська робота містить 77 сторінок, 31 рисунок, 3 розділи список використаних джерел із 32 найменуванням.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
						12
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

# РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПОБУДОВИ ІНТЕРАКТИВНИХ ВЕБ-ЗАСТОСУНКІВ ДЛЯ НАВЧАННЯ І ТЕСТУВАННЯ ЗНАНЬ

## 1.1. Концепція та архітектура веб-додатку для організації інтелектуальних квестів

Цей документ описує архітектуру та функціональні можливості веб-додатку, призначеного для спрощення процесу створення, адміністрування та участі в інтелектуальних квестах. Запропонована система є комплексним рішенням, що оптимізує організацію освітніх та оцінювальних заходів, забезпечуючи при цьому високий рівень об'єктивності та достовірності результатів. Dodatok імплементує тридискретні ролі користувачів: Адміністратор, Викладач та Студент, кожна з яких наділена чітко визначеними функціональними можливостями та рівнями доступу, що відповідають їхнім специфічним потребам у рамках освітнього процесу.

### 1.1.1. Функціональні ролі та можливості

Роль адміністратора є центральною ланкою в ієрархії управління системою. Вона надає повний контроль над організаційною структурою та кадровим складом. Основні функції адміністратора включають:

- Управління відділами - створення, модифікація та видалення структурних підрозділів освітнього закладу.
- Управління курсами - адміністрування навчальних програм та дисциплін, що викладаються.
- Управління викладачами - реєстрація, редагування профілів та призначення викладачів до відповідних курсів та відділів.

Моніторинг системної активності - огляд загальної статистики використання системи та стану її компонентів.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Роль викладача є ключовою для розробки та проведення інтелектуальних квестів. Інтерфейс викладача розроблено таким чином, щоб максимально спростити процес підготовки та оцінювання. До його функцій належать:

1. Планування квестів.

Створення розкладу для майбутніх інтелектуальних квестів, визначення дати, часу та тривалості.

2. Розробка питань.

Створення та інтеграція різнотипних завдань, що можуть включати текстові питання, вибір варіантів, завдання на відповідність, питання з відкритою відповіддю тощо. Система може підтримувати мультимедійні елементи для підвищення інтерактивності квестів.

3. Управління студентами.

Формування груп студентів для участі у квестах, відстеження їх прогресу та надання доступу до результатів.

4. Налаштування параметрів квестів.

Встановлення критеріїв оцінювання, лімітів часу, кількості спроб та інших специфічних правил для кожного квесту.

Роль студента є основною для взаємодії з навчальним контентом та проходження оцінювання. Інтерфейс студента є інтуїтивно зрозумілим та забезпечує ефективну участь у квестах.

Основні можливості студента:

а) Проходження квестів.

Доступ до призначених інтелектуальних квестів та їх виконання у визначені терміни.

б) Перегляд результатів.

Ознайомлення з підсумковими балами та детальною інформацією щодо виконаних завдань.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

в) Відгуки та аналіз.

Можливість отримання зворотного зв'язку від викладача та аналізу власних помилок для покращення знань.

### *1.1.2. Забезпечення об'єктивності оцінювання*

Однією з критично важливих функцій даного веб-додатку є вбудована система запобігання недоброчесності під час онлайн-тестування. Цей механізм має на меті гарантувати чесність та достовірність оцінки знань студентів. Потенційні заходи можуть включати, але не обмежуватися:

- Рандомізація порядку питань та варіантів відповідей, що унеможлиблює обмін відповідями між студентами.

- Обмеження часу на відповідь, тобто встановлення жорстких часових рамок для кожного питання або квесту в цілому.

- Моніторинг активності браузера відстеженням перемикання між вікнами або спроб доступу до сторонніх ресурсів під час проходження квесту.

- Системи прокторингу - інтеграція з програмним забезпеченням для відеоспостереження або моніторингу екрану для віддаленого контролю над процесом тестування (опційно).

- Виявлення аномалій за допомогою використання алгоритмів, що аналізують поведінку користувачів та виявляють нетипові шаблони, що можуть свідчити про спроби обману.

Впровадження цих механізмів дозволяє забезпечити високу академічну доброчесність та сприяє формуванню об'єктивної оцінки знань, що є фундаментальним для ефективного освітнього процесу.

Цей веб-додаток є комплексним інструментом, що не тільки спрощує адміністрування та проходження інтелектуальних квестів, але й сприяє підвищенню якості освітнього процесу завдяки інтегрованим механізмам контролю та оцінювання.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

## 1.2. Актуальність розробки веб-додатку для організації інтелектуальних квестів

Пандемія COVID-19 виявила значні виклики для освітніх установ, змусивши їх оперативно перейти до формату дистанційного навчання. Однією з найгостріших проблем, що постала перед викладачами, була відсутність адекватних інструментів для ефективної та об'єктивної оцінки знань студентів в онлайн-середовищі. У відповідь на ці виклики був розроблений інноваційний веб-додаток, призначений для організації інтелектуальних квестів. Цей додаток покликаний задовольнити потреби освітніх закладів у ефективному управлінні навчальним процесом, охоплюючи адміністрування відділів, курсів, викладачів, студентів та, безпосередньо, інтелектуальних квестів.

Основне значення розробленого веб-додатку полягає в суттєвому спрощенні процесу управління інтелектуальними квестами для викладачів. Система надає інтуїтивно зрозумілий інтерфейс, що дозволяє викладачам безперешкодно створювати квести та формулювати питання, які точно відповідають змісту їхніх курсів.

Ключовою функціональною перевагою є система автоматичної оцінки відповідей. Ця інноваційна можливість звільняє викладачів від трудомісткого процесу ручної перевірки робіт студентів. Веб-додаток автоматично аналізує надані відповіді та обчислює підсумковий бал згідно з попередньо встановленими критеріями оцінювання для кожного квесту. Це не тільки значно економить час викладачів, дозволяючи їм зосередитися на інших аспектах освітнього процесу, але й мінімізує суб'єктивний фактор в оцінюванні, забезпечуючи її прозорість та об'єктивність.

Інтерфейс користувача даного проекту розроблений з акцентом на простоту та ефективність, що робить його доступним для будь-якого користувача, незалежно від рівня його технічної підготовки. Інтуїтивна

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

навігація та логічна структура забезпечують швидке освоєння функціоналу та безперебійну роботу.

Крім того, критично важливим аспектом веб-додатку є вбудована функція запобігання недоброчесності під час онлайн-тестування. Цей механізм має на меті гарантувати академічну доброчесність та достовірність отриманих результатів.

Таким чином, розроблюваний веб-додаток для інтелектуальних квестів є комплексним та багатофункціональним інструментом, що не лише вирішує актуальні проблеми дистанційної оцінки знань, але й сприяє підвищенню ефективності та прозорості всього навчального процесу в умовах сучасної освітньої парадигми.

### **1.3. Системні вимоги до розробки веб-додатку**

Для забезпечення оптимальної продуктивності та стабільного функціонування розробленого веб-додатку для інтелектуальних квестів, необхідно дотримуватися наступних системних вимог

#### *1.3.1. Апаратні вимоги*

##### **1. Оперативна пам'ять.**

Рекомендується мінімум 4 ГБ для безперебійної роботи додатку та операційної системи. Для розробки та тестування зі значними обсягами даних або кількома одночасними процесами може бути бажаним збільшення цього обсягу.

##### **2. Сховище.**

Необхідно щонайменше 64 ГБ вільного дискового простору для встановлення операційної системи, програмного забезпечення, бази даних та збереження файлів додатку.

##### **3. Жорсткий диск.**

					БР.ІП – 38.00.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Рекомендований обсяг 160 ГБ забезпечить достатній простір для довгострокового зберігання даних, резервних копій та системних логів.

#### 4. Процесор.

Процесор класу Intel i3 або еквівалентний забезпечить достатню обчислювальну потужність для виконання основних операцій додатку. Для розробки та великих навантажень може бути рекомендований більш потужний процесор.

#### 1.3.2. Програмні вимоги

##### 1. Операційна система.

Розробка та функціонування додатку оптимізовані для операційної системи Microsoft Windows. Сумісність з іншими операційними системами, такими як Linux або macOS, не гарантується без додаткових налаштувань.

##### 2. Інтегроване середовище розробки (IDE).

Для розробки та підтримки коду рекомендується використовувати Visual Studio Code, що забезпечує гнучке та функціональне середовище для роботи з JavaScript та пов'язаними технологіями.

##### 3. Мови програмування та фреймворки:

- React: Використовується для розробки інтерфейсу користувача (frontend) завдяки своїй компонентній архітектурі та ефективності. Пакунок JavaScript під назвою React рендерить користувацькі інтерфейси (UI). Користувацький інтерфейс складається з невеликих компонентів, таких як кнопки, текст та графіка

- Node.js: Застосовується для розробки серверної частини (backend) додатку, забезпечуючи асинхронну та масштабовану обробку запитів. Це крос-платформне середовище виконання JavaScript з відкритим вихідним кодом

- JavaScript: Основна мова програмування, що використовується як на клієнтській, так і на серверній стороні.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

- HTML: Стандартна мова розмітки для створення структури веб-сторінок.

- SCSS: Передпроцесор CSS, що розширює можливості стилізації та робить її більш модульною.

- База даних: Для зберігання та управління даними додатку використовується реляційна система управління базами даних MySQL.

- Об'єктно-реляційне відображення (ORM): Sequelize використовується для взаємодії з базою даних, спрощуючи операції з даними та забезпечуючи абстракцію від синтаксису SQL.

#### 4. Інструменти розробки та розгортання:

- XAMPP: Комплект програмного забезпечення, що включає Apache, MySQL та PHP, необхідний для локального розгортання та тестування.

- phpMyAdmin: Веб-інтерфейс для управління базами даних MySQL.

- npm (Node Package Manager): Менеджер пакетів для JavaScript, використовується для встановлення бібліотек та залежностей.

- prx: Інструмент для запуску виконуваних пакетів Node.js без їх глобальної установки.

- nodemon: Інструмент для автоматичного перезапуску Node.js додатку при виявленні змін у файлах.

#### 5. Веб-сервер.

Для розгортання та обслуговування додатку використовується Apache Tomcat, що забезпечує середовище для виконання Java-сервлетів та статичного контенту.

Дотримання цих системних вимог є критично важливим для забезпечення коректної роботи та максимальної ефективності веб-додатку для інтелектуальних квестів.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

## 1.4. Інструменти та технології розробки веб-додатку для інтелектуальних квестів

Розробка веб-додатку для інтелектуальних квестів базувалася на використанні сучасних та ефективних інструментів і технологій, що забезпечили його функціональність, масштабованість та зручність у використанні. Нижче наведено детальний опис ключових компонентів програмного стеку, що були задіяні у проекті.

### 1.4.1. React

React є декларативною, ефективною та гнучкою бібліотекою JavaScript, призначеною для розробки інтерфейсів користувача (UI). Вона базується на компонентній архітектурі, де кожен елемент інтерфейсу, такий як кнопки, текстові поля або графічні елементи, інкапсулюється у незалежні та повторно використовувані компоненти [1]. Цей підхід дозволяє декомпонувати складні інтерфейси на менші, керовані частини. Для даного проекту була використана версія 16.8.4 бібліотеки React. Ініціалізація розробки здійснювалася за допомогою прх (Node Package Executes), а необхідні бібліотеки та залежності встановлювалися за допомогою прм (Node Package Manager). Архітектура додатку реалізована як односторінковий веб-додаток (SPA) з використанням віртуального DOM (Document Object Model), що оптимізує оновлення інтерфейсу. Навігація між сторінками забезпечувалася за допомогою бібліотеки react-router-dom.

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

// Рендеринг основного компонента React у DOM-елемент з ідентифікатором 'root'.
ReactDOM.render(<App />, document.getElementById('root'));
```

Рисунок 1.1 – Використання бібліотеки react-router-dom

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

### 1.4.2. Node.js

Node.js є крос-платформним середовищем виконання JavaScript з відкритим вихідним кодом, що дозволяє виконувати JavaScript-код поза межами веб-браузера. Він використовує високопродуктивний движок JavaScript V8 від Google Chrome, що забезпечує значну швидкість виконання коду. Ключовою особливістю Node.js є його асинхронна, неблокуюча модель вводу-виводу, що дозволяє обробляти велику кількість одночасних запитів без створення окремого потоку для кожного з них. Це досягається завдяки набору асинхронних примітивів вводу-виводу, інтегрованих у стандартну бібліотеку Node.js, що запобігає блокуванню потоку виконання JavaScript під час операцій, таких як мережеві запити, доступ до баз даних або файлової системи [2]. Для розробки серверної частини веб-додатку була використана версія Node LTS 18.18.0. Для забезпечення безперервної розробки та автоматичного перезапуску сервера при змінах у вихідному коді використовувався пакет nodemon.

```
const http = require('http');

// Створення HTTP-сервера.
const server = http.createServer((req, res) => {
  res.statusCode = 200; // Встановлення статусу відповіді OK.
  res.setHeader('Content-Type', 'text/plain'); // Встановлення типу контенту.
  res.end('Hello World\n'); // Надсилання відповіді клієнту.
});

// Запуск сервера на порту 3000.
server.listen(3000, () => {
  console.log('Server running at http://localhost:3000/');
});
```

Рисунок 1.2 – Використання пакету nodemon

### 1.4.3. Express.js

Express.js є мінімалістичним та гнучким веб-фреймворком для Node.js, який надає надійні функції для розробки веб-додатків та API. Він дозволяє швидко створювати потужні API за допомогою різноманітних HTTP-методів

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

та механізмів проміжного програмного забезпечення (middleware). Express.js зберігає неблокуючі характеристики Node.js, одночасно надаючи основний рівень функціональності, необхідний для веб-розробки. У веб-додатку для інтелектуальних квестів Express.js використовувався для реалізації RESTful API, маршрутизації запитів та архітектури MVC (Model-View-Controller) на серверній стороні.

```
const express = require('express');
const app = express();

// Визначення маршруту для кореневого URL.
app.get('/', (req, res) => {
  res.send('Hello World!'); // Надсилання текстової відповіді.
});

// Запуск сервера Express на порту 3000.
app.listen(3000, () => {
  console.log('Server is running on port 3000');
});
```

Рисунок 1.3 – Використання Node.js та бібліотеки Express

#### 1.4.4. Sequelize

Sequelize — це сучасний об'єктно-реляційний мапер (ORM) для Node.js, який підтримує різні системи управління базами даних, включаючи MySQL, PostgreSQL, SQLite, MariaDB та SQL Server. Він дозволяє розробникам взаємодіяти з реляційними базами даних за допомогою об'єктно-орієнтованого підходу, абстрагуючись від прямого написання SQL-запитів. Sequelize підтримує такі важливі функції, як реплікація читання, відкладене та негайне завантаження, визначення відносин між моделями, а також транзакції. Він спрощує визначення моделей даних та надає можливість автоматичної синхронізації схеми бази даних. Крім того, Sequelize ефективно керує асоціаціями між моделями, значно спрощуючи роботу з комплексними структурами даних [4]. У даному проекті Sequelize використовувався у поєднанні з базою даних MySQL для проектування та реалізації

масштабованої структури даних, включаючи визначення моделей та їх відносин.

```
const { Sequelize, DataTypes } = require('sequelize');
// Ініціалізація екземпляра Sequelize для підключення до бази даних MySQL.
const sequelize = new Sequelize('database', 'username', 'password', {
  host: 'localhost',
  dialect: 'mysql' // Визначення діалекту бази даних.
});

// Визначення моделі 'User'.
const User = sequelize.define('User', {
  username: DataTypes.STRING, // Поле для імені користувача.
  email: DataTypes.STRING // Поле для електронної пошти.
});

(async () => {
  await sequelize.sync(); // Синхронізація моделі з базою даних (створення таблиці, якщо її немає).
  // Створення нового запису користувача.
  const user = await User.create({ username: 'john_doe', email: 'john@example.com' });
  console.log(user.toJSON()); // Виведення створеного користувача у форматі JSON.
})();
```

Рисунок 1.4 – Використання Sequelize для роботи з базою даних MySQL

#### 1.4.5. MySQL

MySQL є потужним, багатопотоковим, багатокористувацьким та високошвидкісним сервером баз даних, що реалізує стандартну мову структурованих запитів (SQL). Ця система управління реляційними базами даних (СУБД) розроблена для інтеграції у широко розповсюджене програмне забезпечення, а також для використання у високопродуктивних виробничих системах [5].

```
-- Створення таблиці 'users' для зберігання інформації про користувачів.
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY, -- Унікальний ідентифікатор користувача, автоінкремент.
  username VARCHAR(255) NOT NULL, -- Ім'я користувача, обов'язкове поле.
  email VARCHAR(255) NOT NULL -- Електронна пошта користувача, обов'язкове поле.
);

-- Вставка тестових даних у таблицю 'users'.
INSERT INTO users (username, email) VALUES ('john_doe', 'john@example.com');
```

Рисунок 1.5 – Створення таблиці БД

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

У веб-додатку для інтелектуальних квестів MySQL була обрана як основна база даних завдяки її надійності та здатності забезпечувати цілісність даних. MySQL повністю відповідає властивостям ACID (атомарність, узгодженість, ізольованість та стійкість), що гарантує збереження узгодженості даних у базі даних навіть у випадку системних збоїв.

#### 1.4.6. PhpMyAdmin

PhpMyAdmin — це програмний інструмент з відкритим вихідним кодом, розроблений на мові PHP, призначений для дистанційного адміністрування баз даних MySQL через веб-інтерфейс. PhpMyAdmin підтримує широкий спектр операцій MySQL, дозволяючи користувачам швидко виконувати будь-які SQL-запити, а також керувати базами даних, таблицями, стовпцями, відносинами, індексами, користувачами та їхніми дозволами за допомогою інтуїтивно зрозумілого графічного інтерфейсу [6]. У контексті веб-додатку для інтелектуальних квестів phpMyAdmin використовувався для оперативного перегляду та верифікації структури даних, таблиць та їх відносин (рис. 1.6).

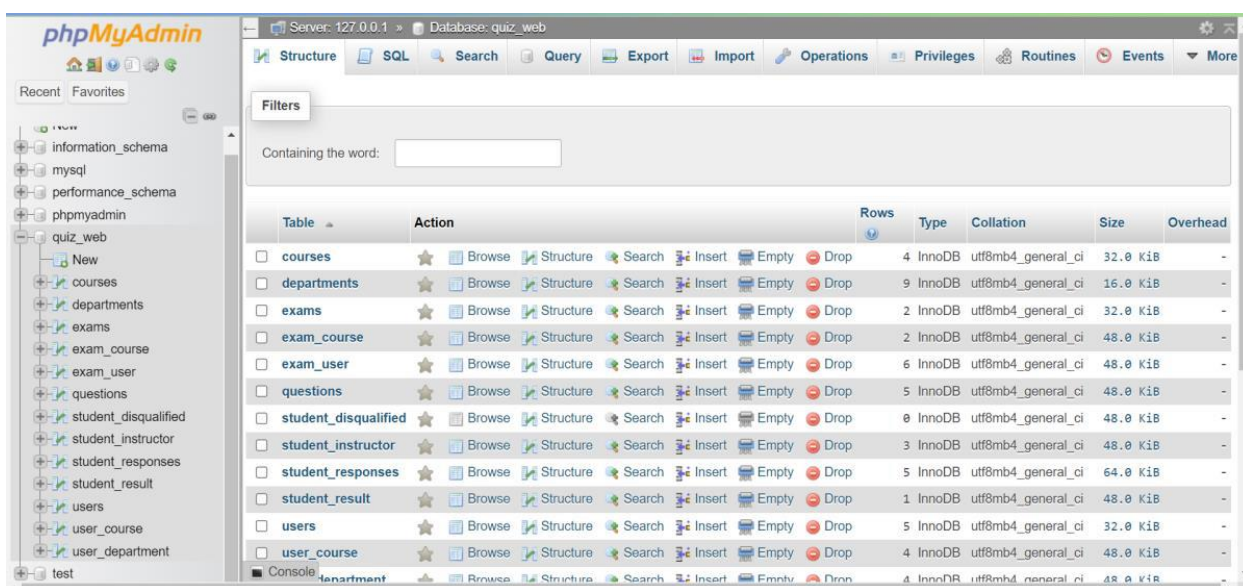


Рисунок 1.6 – Вигляд бази даних в PhpMyAdmin

Функціональність PhpMyAdmin підтримується модульною структурою, яка включає підтримку різних версій MySQL та MariaDB, а також інтеграцію з вебсервером Apache та PHP-оточенням. Інструмент характеризується інтуїтивно зрозумілим графічним інтерфейсом, що полегшує виконання складних адміністративних завдань без необхідності прямого використання командного рядка. Окрім того, PhpMyAdmin підтримує багатомовність (понад 72 мов) та адаптацію до різних операційних систем, що сприяє його глобальному використанню.

#### 1.4.7. XAMPP

XAMPP є вільно розповсюджуваним дистрибутивом Apache, який включає MariaDB (форк MySQL), PHP та Perl. Це програмне забезпечення з відкритим вихідним кодом розроблено з акцентом на простоту встановлення та використання, що робить його ідеальним для локального розгортання веб-додатків та тестування.

У його назві зашифровані основні складові: Apache — веб-сервер, MySQL або MariaDB — база даних, PHP та Perl — мови програмування. Усе це поєднано в одному пакеті, який легко інсталується на Windows, Linux або macOS. Після встановлення користувач отримує просту панель керування, де можна запускати або зупиняти сервер, працювати з базами даних через зручний інтерфейс phpMyAdmin, тестувати свої PHP-скрипти та експериментувати без ризику для "живого" сайту.

Переваги XAMPP полягають в простому встановленні — все входить в один інсталятор, зручності у тестуванні PHP-додатків локально, без потреби в окремому хостингу та підходить для навчання, розробки та тестування веб-проектів.

XAMPP особливо корисний у навчанні та на етапах розробки. Він дозволяє створювати локальні копії сайтів, перевіряти роботу скриптів і баз даних, перш ніж публікувати їх в Інтернеті. Хоча XAMPP не рекомендовано

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

використовувати як сервер для реального хостингу через спрощену систему безпеки, для локального використання він майже ідеальний.

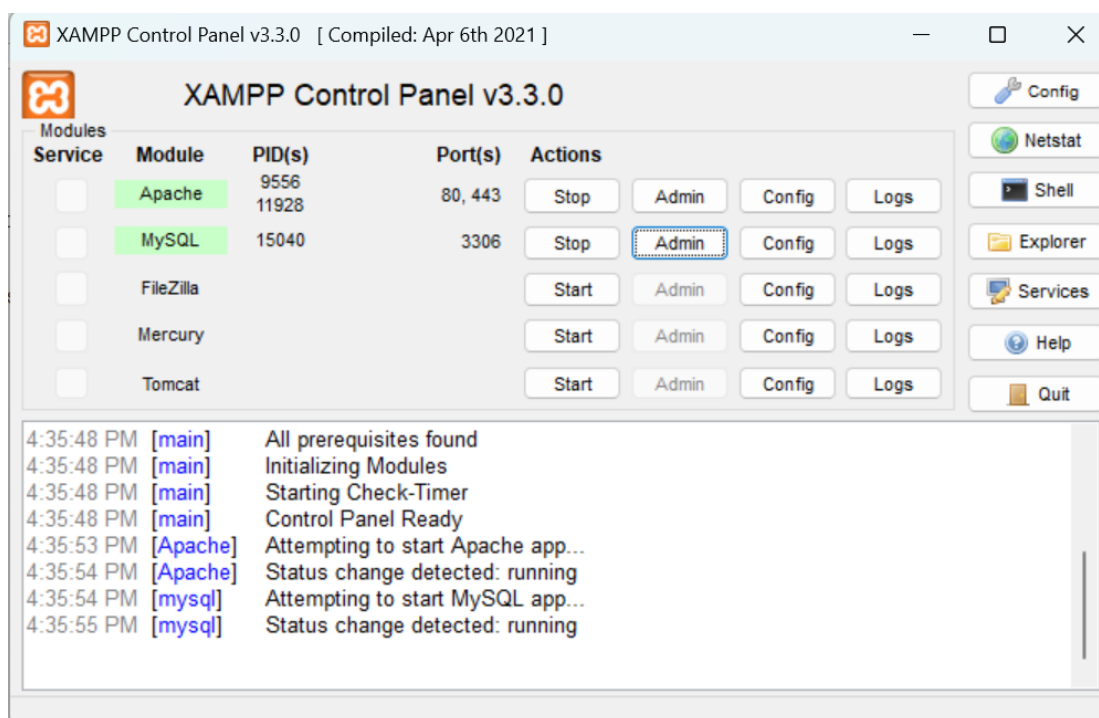


Рисунок 1.7 – Дистрибутив ХАМРР

У даному проекті ХАМРР використовувався для локального запуску модулів MySQL та Apache, що забезпечувало необхідне середовище для розробки та тестування серверної та баз даних частин додатку (рис. 1.7).

## 1.5. Аналіз схожих існуючих навчальних платформ

### 1.5.1. Системи управління навчанням (LMS)

LMS - це програмні системи для адміністрування, документування, відстеження, звітування та доставки освітніх курсів або навчальних програм. Вони пропонують централізовану платформу для всіх аспектів навчання.

Moodle - одна з найпопулярніших open-source LMS. Має потужний модуль для тестування (Quizzes), який дозволяє створювати різні типи питань, налаштовувати часові обмеження, перемішування питань/відповідей.

Підтримує ролі викладача, студента, адміністратора. Функціонал зворотного зв'язку та оцінки є дуже розвиненим. Moodle також може інтегруватися з прокторинговими рішеннями сторонніх розробників.

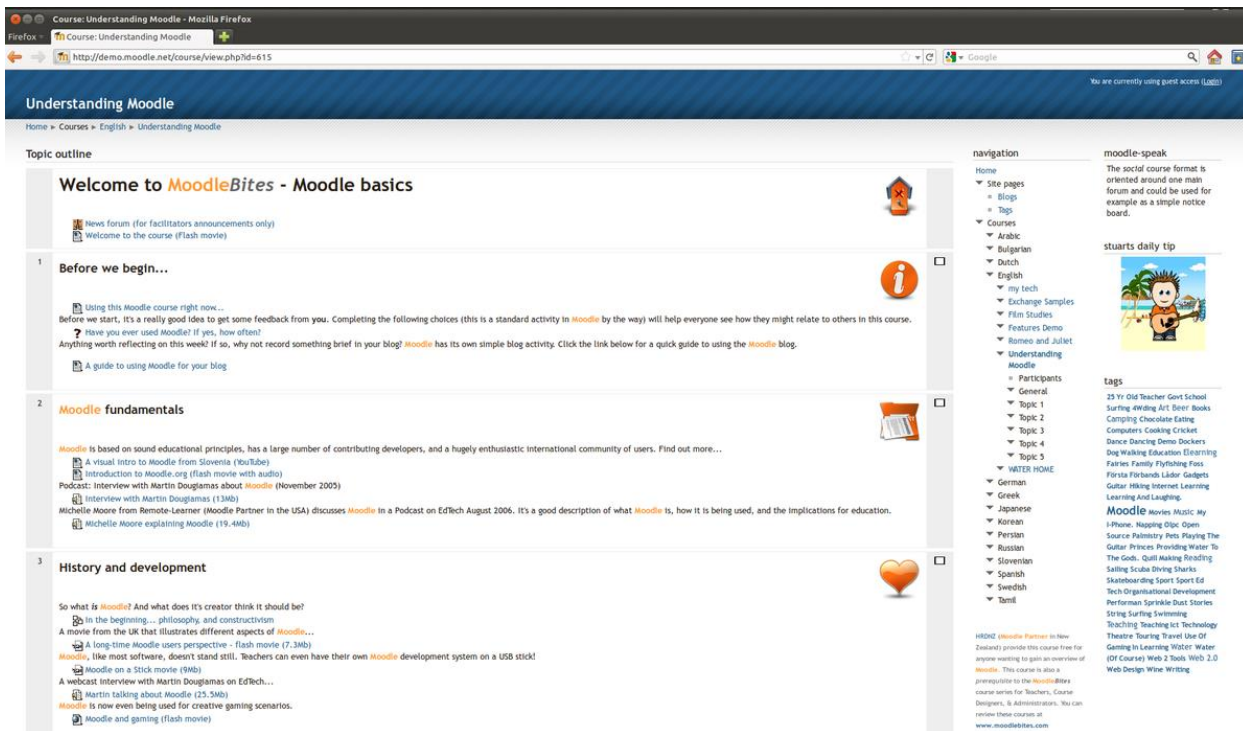


Рисунок 1.8 – LMS Moodle

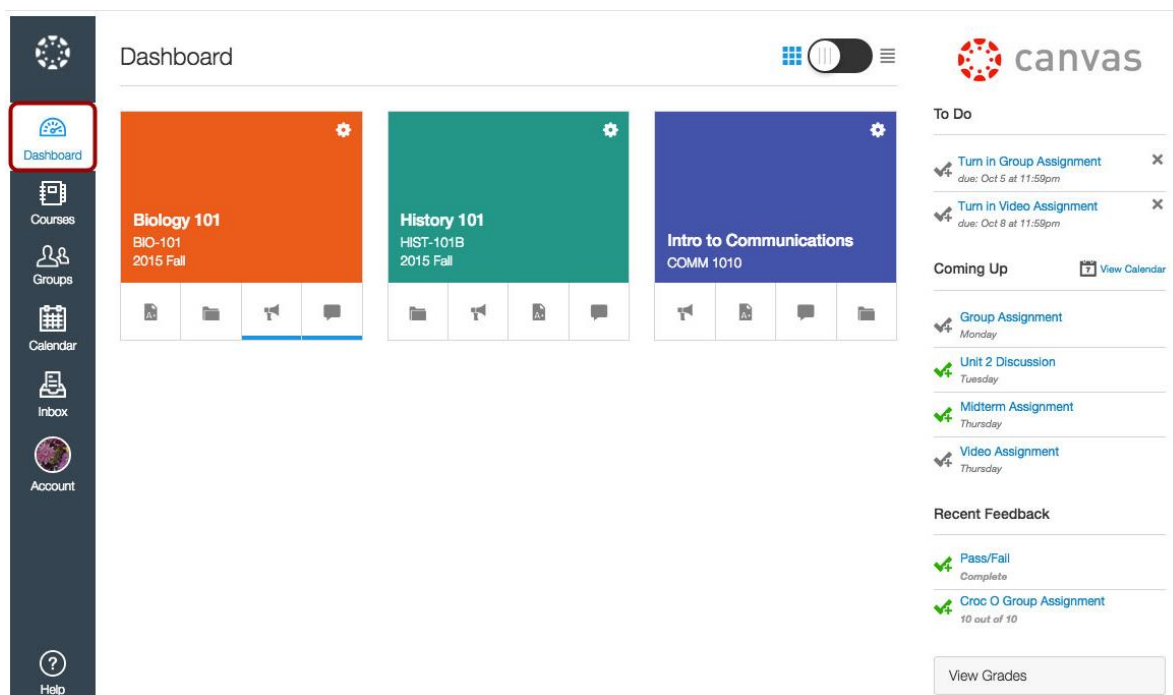


Рисунок 1.9 – Інтерфейс LMS Canvas

Canvas (рис. 1.9) - комерційна LMS, відома своїм інтуїтивно зрозумілим інтерфейсом та гнучкістю. Пропонує широкі можливості для створення тестів, автоматичної оцінки, управління групами студентів та інтеграції з зовнішніми інструментами (LTI), включаючи прокторингові сервіси.

Blackboard Learn - ще одна велика комерційна LMS. Надає інструменти для створення тестів, журналів оцінок, дискусійних форумів та співпраці. Має власні функції відстеження активності студентів та інтегрується з системами прокторингу.

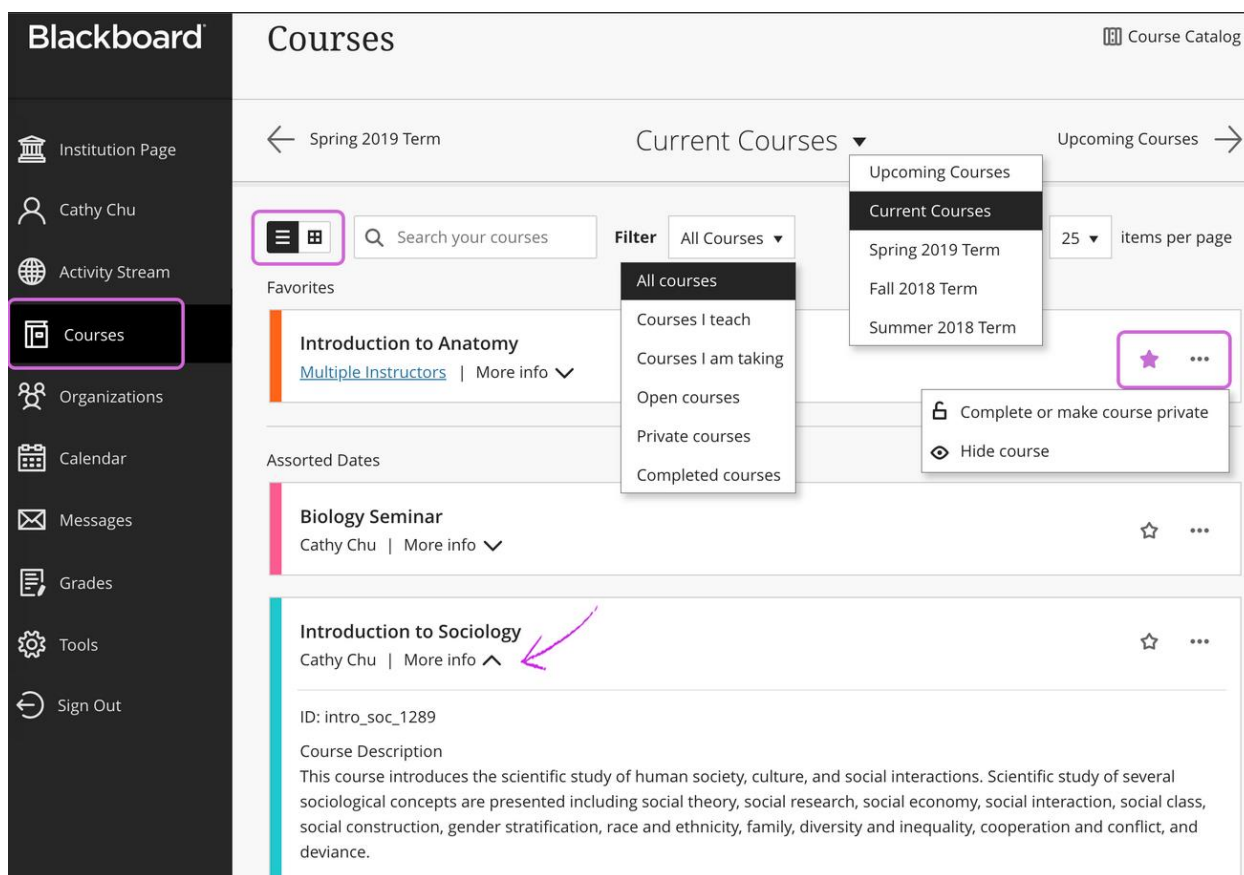


Рисунок 1.10 – Інтерфейс LMS Blackboard Learn

Google Classroom - простіше рішення, орієнтоване на співпрацю та організацію завдань. Має функціонал для створення тестів через Google Forms, але можливості контролю та прокторингу є більш обмеженими порівняно з повноцінними LMS.

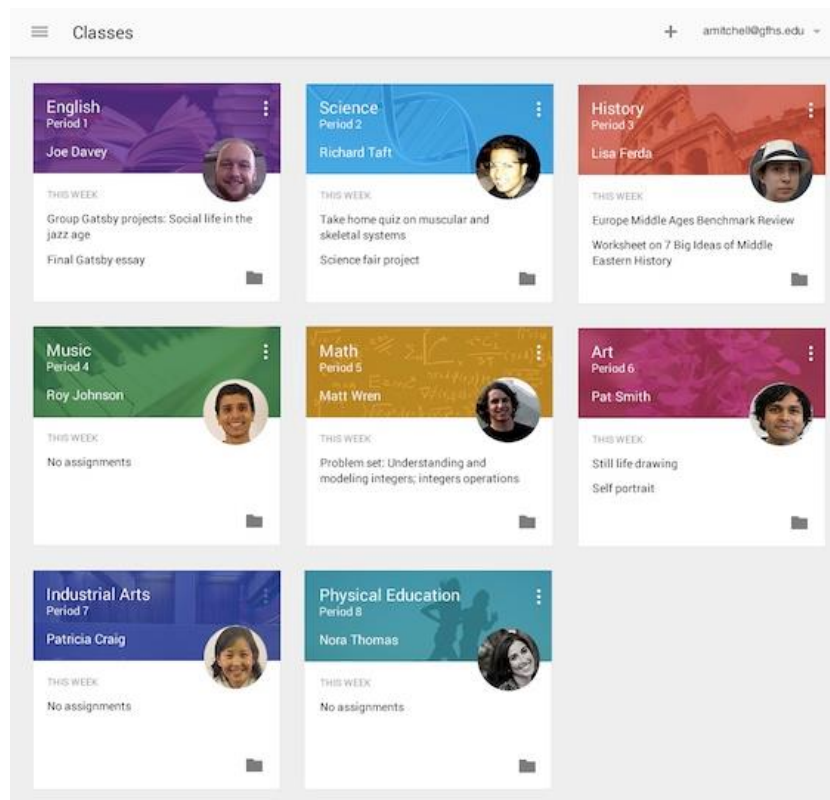


Рисунок 1.11 - Google Classroom

Порівняння з пропонованим веб-застосунком.

Спільні риси: Управління користувачами, створення/проходження тестів, автоматична оцінка, підтримка online навчання.

Відмінності: Описуваний додаток, сфокусований переважно на інтелектуальних квестах/тестуванні, тоді як LMS є ширшими платформами, що охоплюють також управління курсами, комунікацію, календар, файлообмін тощо. Вбудований механізм відстеження екрану в описі є більш специфічним анти-читінг рішенням, яке в більшості LMS реалізується через інтеграцію.

### *1.5.2. Платформи для створення та проведення онлайн-тестів/вікторин*

Ці платформи спеціалізуються на створенні, розповсюдженні та аналізі результатів тестів та вікторин. Вони можуть бути як частиною LMS, так і окремими інструментами.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

Kahoot! - дуже популярна платформа для створення інтерактивних вікторин у ігровій формі. Фокусується на залученні та змагальності, але можливості прокторингу та глибокої оцінки знань тут обмежені.

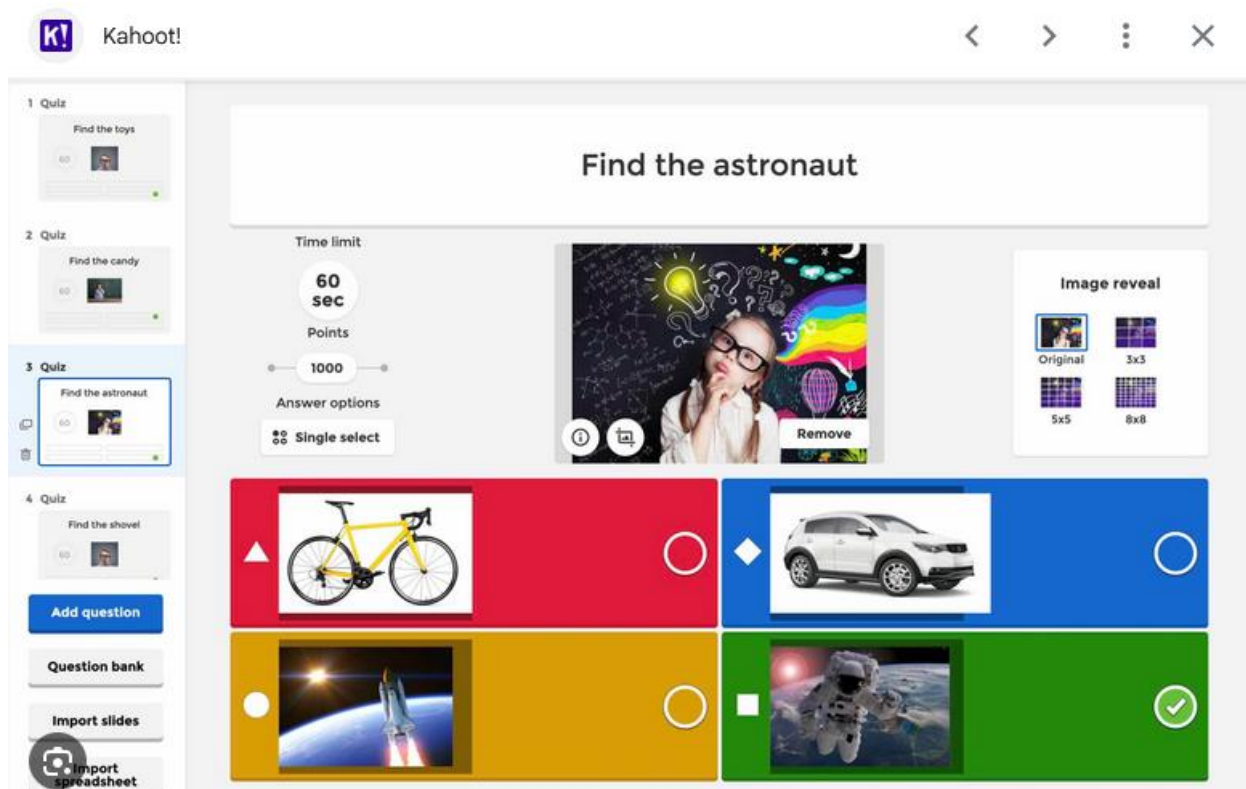


Рисунок 1.12 – Платформа Kahoot!

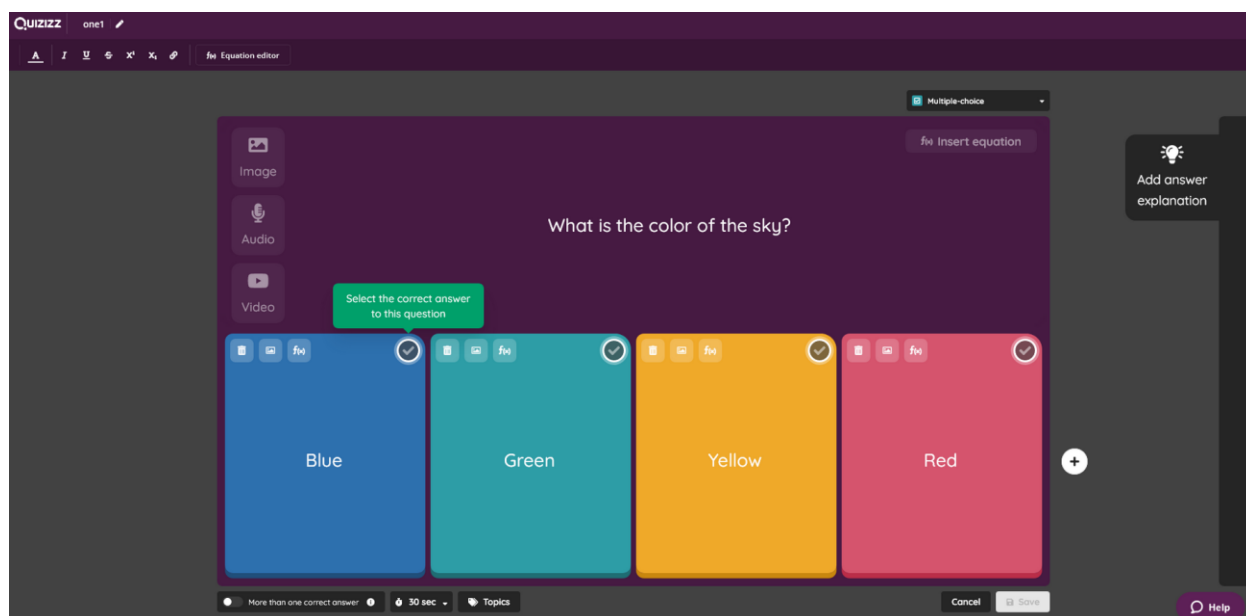


Рисунок 1.13 – Платформа Quizizz

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Quizizz - схожий на Kahoot!, пропонує вікторини з елементами гейміфікації, інтерактивні завдання. Також має обмежені можливості для запобігання списуванню.

ProProfs Quiz Maker дозволяє створювати професійні тести, іспити та вікторини з розширеними можливостями налаштування, включаючи часові обмеження, рандомізацію питань, а також певні функції безпеки, хоча повний прокторинг зазвичай потребує платної версії або інтеграції.

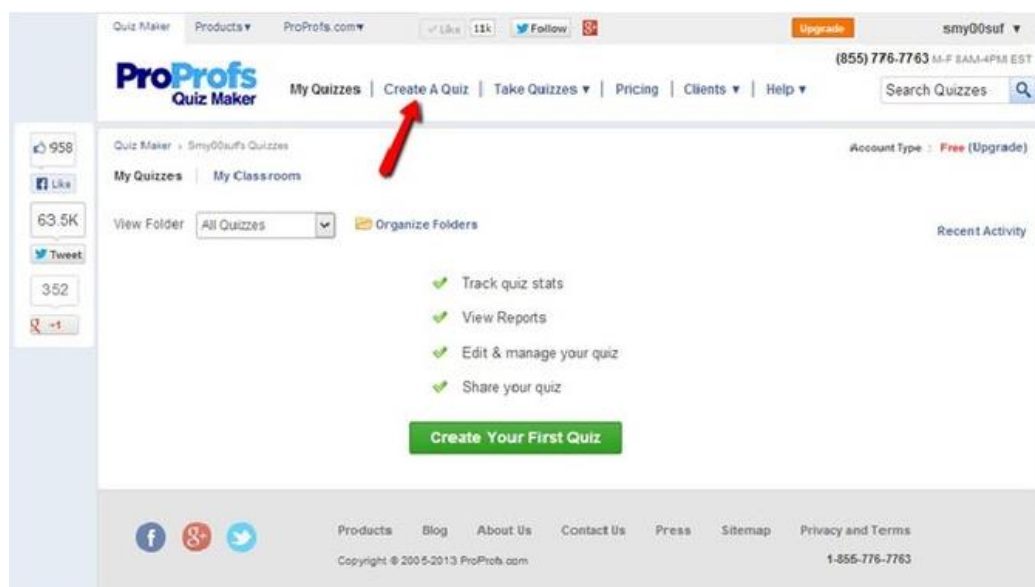


Рисунок 1.14 – Платформа ProProfs Quiz Maker

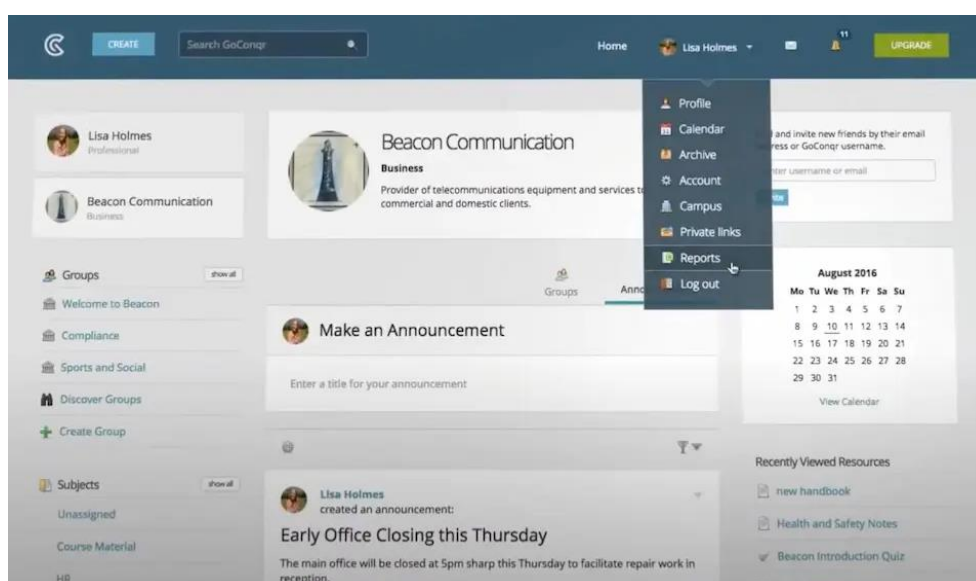


Рисунок 1.15 – Інструмент GoConqr

									Арк.
									31
Змн.	Арк.	№ докум.	Підпис	Дата					

GoConqr (Quiz Maker) пропонує інструменти для створення тестів, флеш-карток, розумових карт. Має базові функції тестування.

Порівняння з пропонованим веб-застосунком.

Спільні риси: Основний функціонал - створення та проходження тестів, автоматична оцінка.

Відмінності: Багато з цих платформ менш сфокусовані на суворому академічному прокторингу порівняно з описуваним додатком. Вони часто більше орієнтовані на інтерактивність та гейміфікацію, а не на боротьбу зі списуванням. Пропонований додаток має більш чітко виражену функцію протидії академічній недоброчесності.

### 1.5.3. Платформи для проведення онлайн-курсів (MOOCs / Course Platforms)

Ці платформи пропонують онлайн-курси, часто з інтегрованими елементами оцінювання.

Coursera / edX - пропонують курси від університетів та організацій. Мають вбудовані системи тестування, але прокторинг часто реалізується через сторонні сервіси (наприклад, для сертифікованих іспитів) і не завжди є частиною базового функціоналу для всіх завдань.

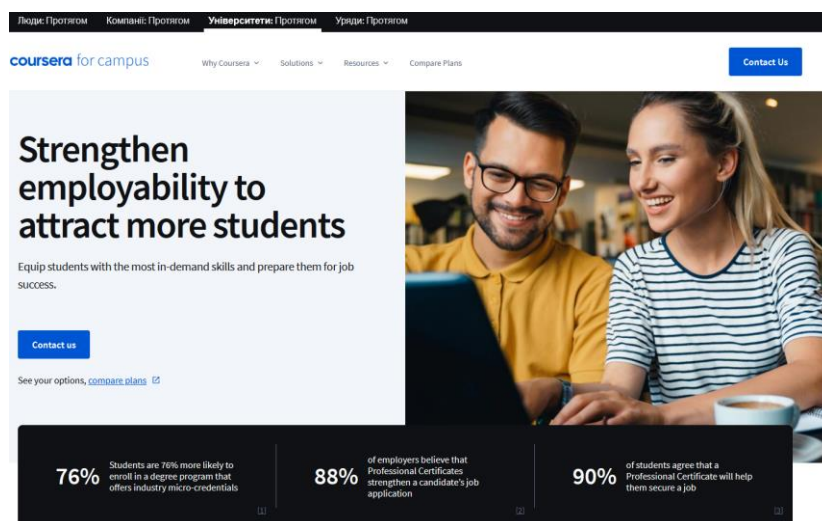


Рисунок 1.16 – Платформа Coursera

									Арк.
									32
Змн.	Арк.	№ докум.	Підпис	Дата					

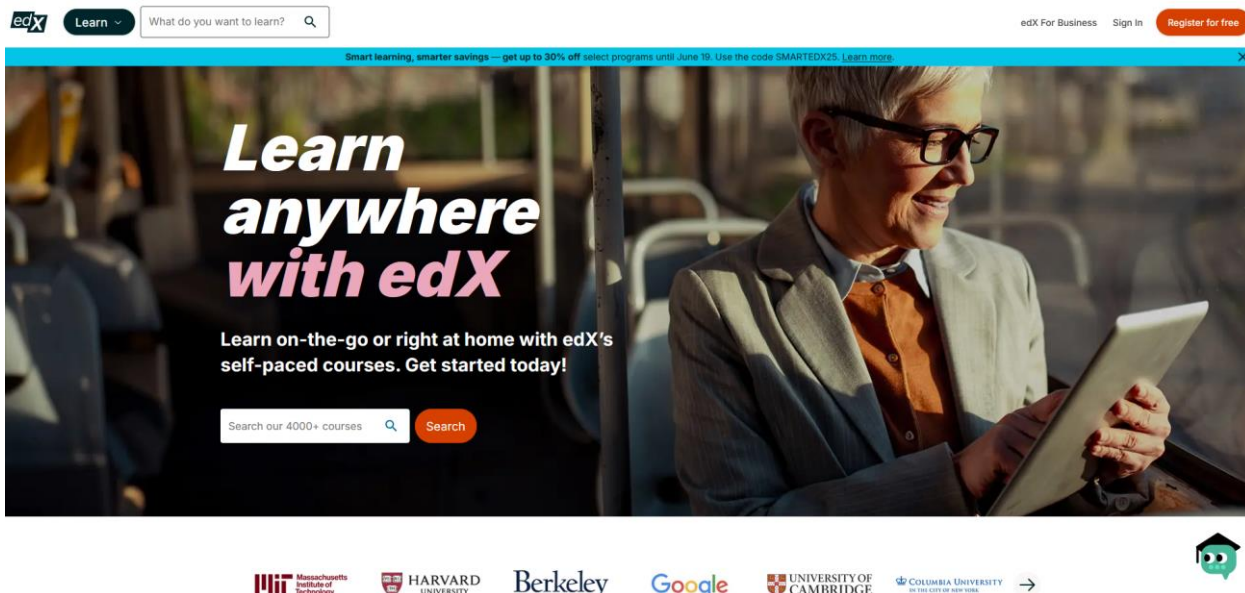


Рисунок 1.17 – Платформа edX

Udemy / Skillshare - платформи для доступу до широкого спектру курсів. Функції тестування та оцінки можуть бути присутні, але фокус на прокторингу та запобіганні шахрайству не є першочерговим.

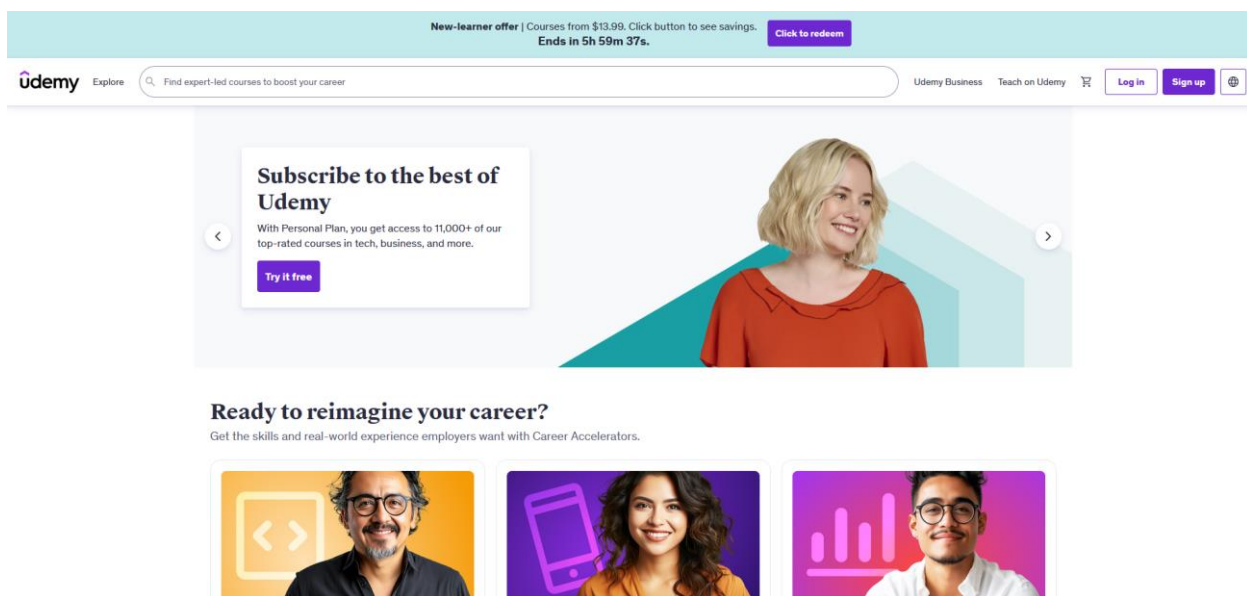


Рисунок 1.18 – Платформа Udemy

Skillshare — це онлайн-платформа для навчання, яка пропонує тисячі курсів із творчих, професійних та особистих навичок. Заснована в 2010 році,

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

базується в Нью-Йорку. Користувачі отримують доступ до відеоуроків, які ведуть експерти та практикуючі фахівці в таких галузях, як дизайн, ілюстрація, фотографія, письменство, маркетинг, програмування та підприємництво. Курси зосереджені на практичному застосуванні: студенти виконують проекти, отримують зворотний зв'язок і можуть ділитися результатами з спільнотою.

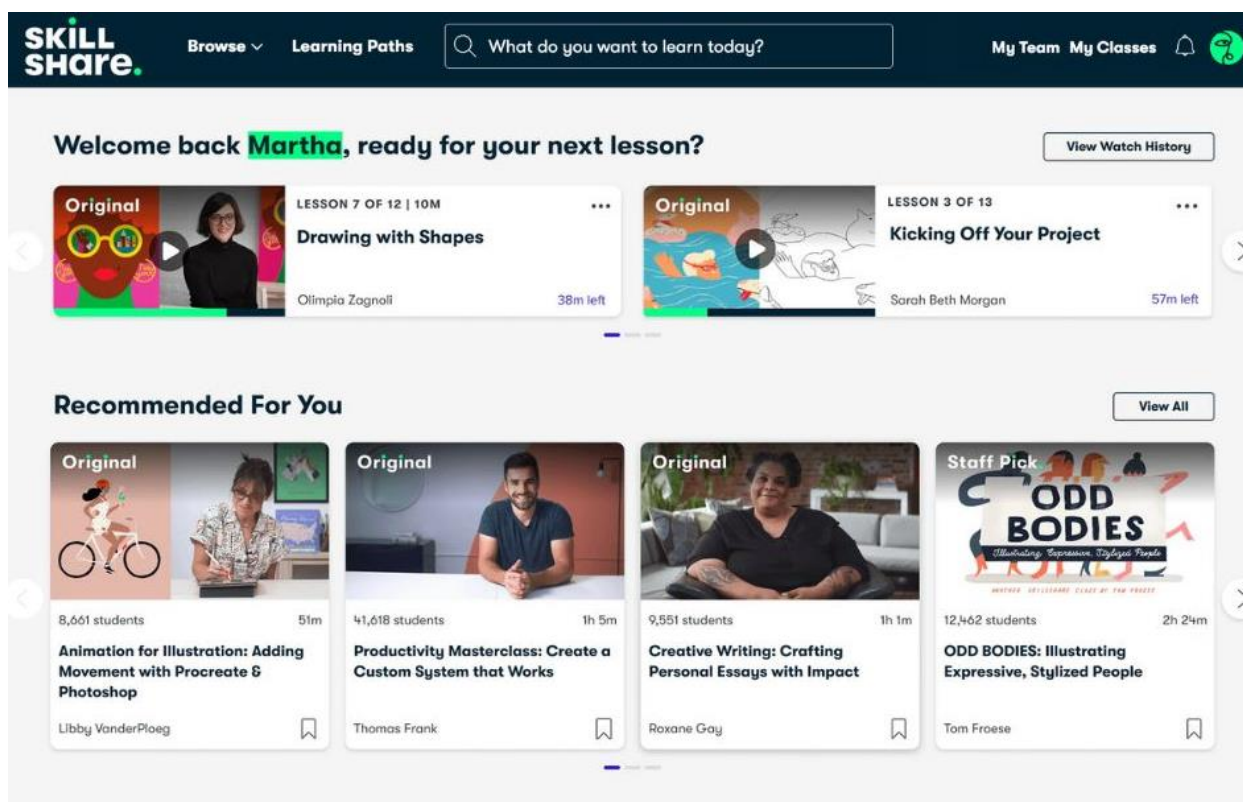


Рисунок 1.19 - Онлайн-платформа Skillshare

Порівняння з пропонованим веб-застосунком.

Спільні риси: Підтримка дистанційного навчання, можливість проведення оцінювання.

Відмінності: Описуваний додаток є більш спеціалізованим на "квестах" та аспектах прокторингу, тоді як МООС-платформи надають повний курс з відео-лекціями, матеріалами, форумами тощо, а тестування є лише одним з компонентів.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

#### 1.5.4. Системи онлайн-прокторингу

Це окремі сервіси або програмні рішення, які використовуються для моніторингу студентів під час онлайн-іспитів, щоб запобігти списуванню.

ProctorU - одна з найвідоміших. Використовує штучний інтелект та живих прокторів для моніторингу за допомогою веб-камери, мікрофона та відстеження екрану.

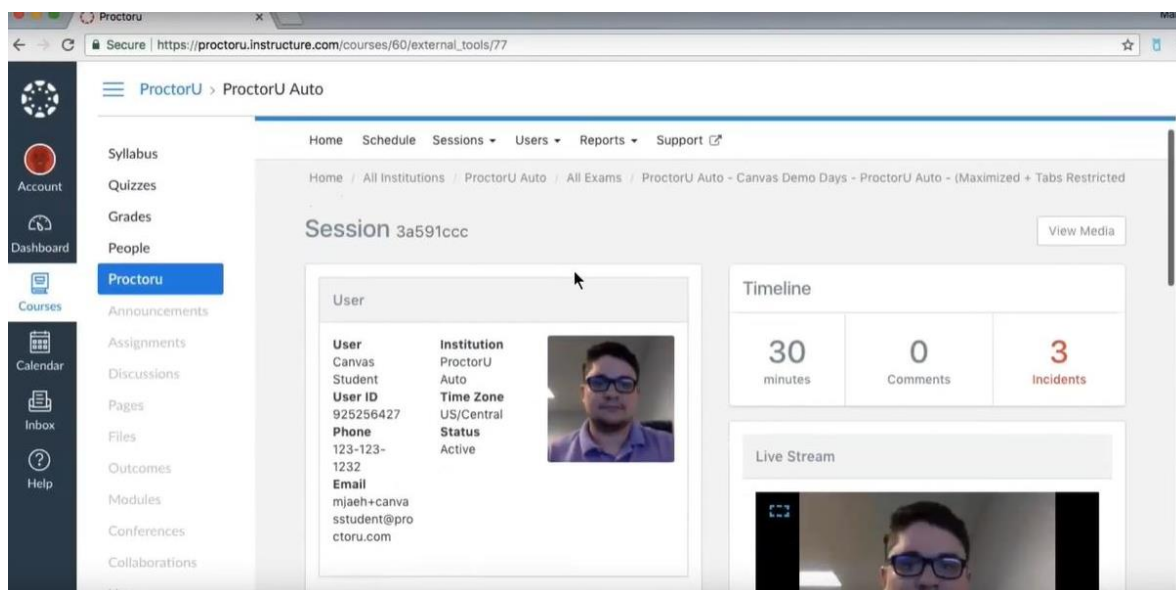


Рисунок 1.20 – ProctorU

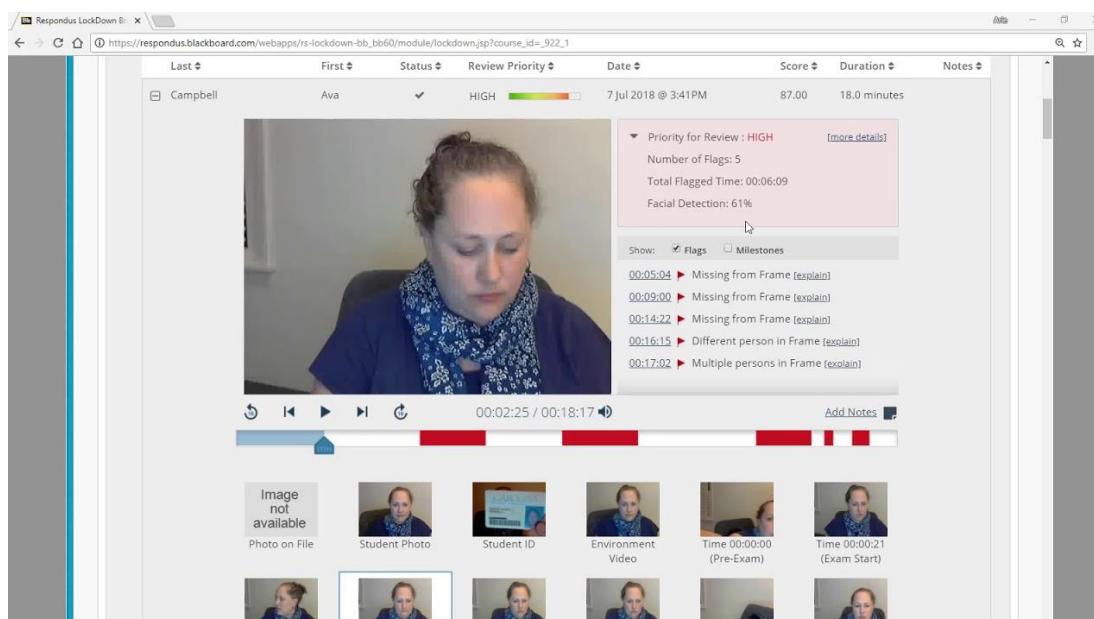


Рисунок 1.21 - Respondus Monitor

Змн.	Арк.	№ докум.	Підпис	Дата

Respondus Monitor часто інтегрується з LMS. Вимагає спеціального браузера та використовує веб-камеру для запису сесії тестування, аналізуючи поведінку студента.

Honorlock поєднує ШІ та опціональний живий прокторинг. Використовує розширення для браузера для моніторингу екрану та поведінки.

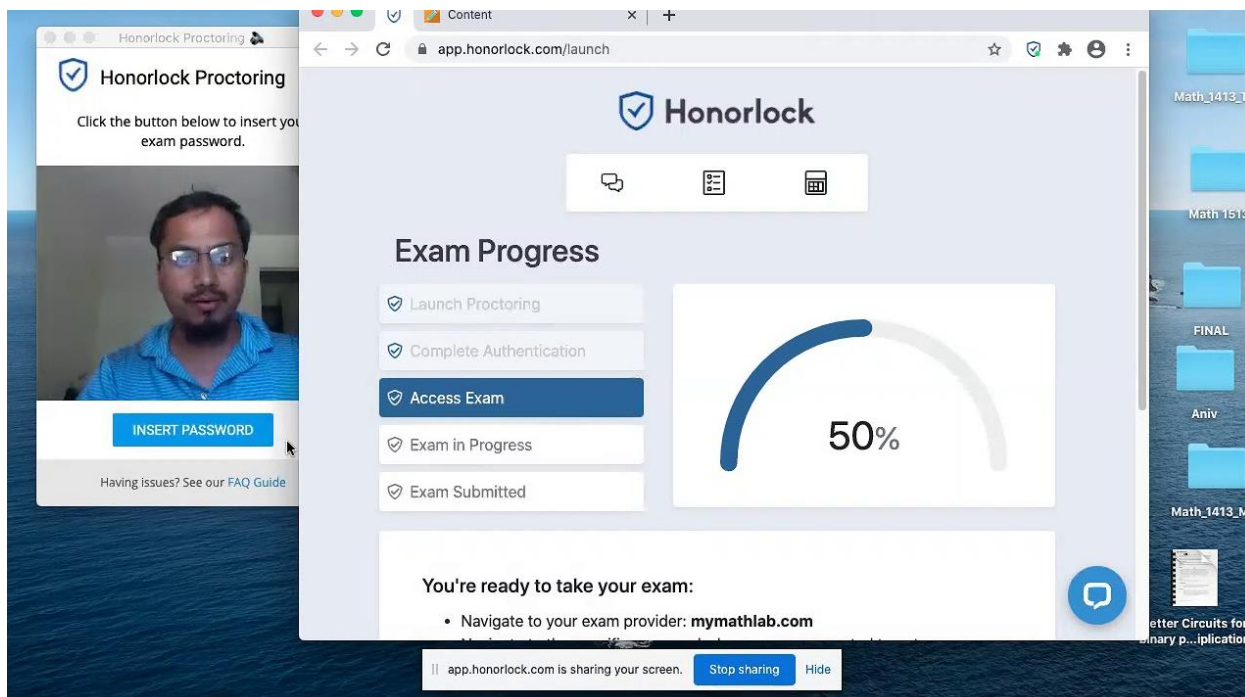


Рисунок 1.22 – Платформа Honorlock

Порівняння з пропонованим веб-застосунком.

Спільні риси: Описуваний додаток описує "відстеження екрану тесту" та "відстеження рухів студентів", що є ключовими функціями прокторингових систем. "Автоматична дискваліфікація студента у випадку зміни активної вкладки браузера" є вбудованим механізмом прокторингу.

Відмінності: Згадані прокторингові системи є окремими, часто платними сервісами, які інтегруються в навчальні платформи. Описуваний додаток, схоже, планує або вже має вбудовані подібні механізми, що є значною перевагою з точки зору інтеграції та вартості.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

Описуваний веб-додаток для інтелектуальних квестів має амбітний набір функцій, який позиціонує його між спеціалізованими інструментами для створення вікторин та повноцінними LMS. Його сильні сторони, полягають у:

1. Акценті на анти-читінг механізмах.

Вбудований контроль часу, відстеження екрану та автоматична дискваліфікація є ключовими відмінностями від багатьох базових платформ для вікторин.

2. Зручному інтерфейсі.

Використання React Datta Template та фокус на UX вказує на зручність використання.

3. Комплексній ролевій моделі.

Наявність адміністратора, викладача та студента забезпечує повний життєвий цикл управління навчальним процесом тестування.

4. Потенціалі для SaaS.

Перехід на SaaS-модель вказує на прагнення до масштабованості та доступності.

Цей додаток може знайти свою нішу як спеціалізована платформа для онлайн-оцінювання з посиленими заходами безпеки, що може бути особливо актуальним для освітніх закладів, які шукають ефективне рішення для проведення іспитів та тестів у дистанційному форматі, але не потребують повного функціоналу LMS.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

## РОЗДІЛ 2. МОДЕЛЮВАННЯ СТРУКТУРИ ВЕБ-ЗАСТОСУНКУ АДМІНІСТРУВАННЯ ІНТЕЛЕКТУАЛЬНИХ КВЕСТІВ ТА ВІКТОРИН

### 2.1. Представлення архітектури веб-застосунку

UML, або Unified Modeling Language, є стандартною візуалізацією для проектування програмного забезпечення. Загальна архітектура веб-застосунку показана на рисунку 3.1.

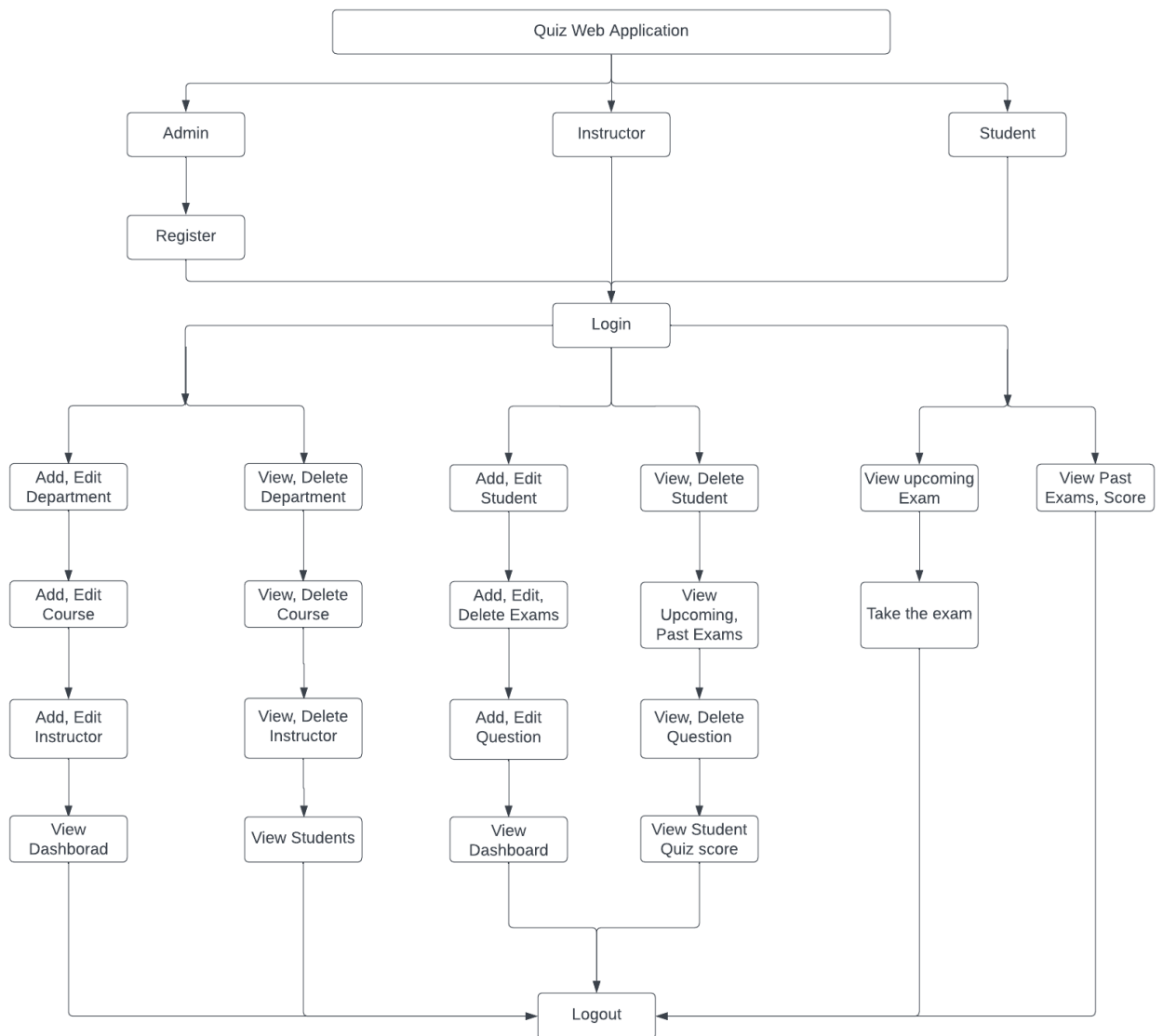


Рисунок 2.1 - Архітектура веб-застосунку для інтелектуальних квестів

На рисунку 2.1 представлено архітектуру веб-застосунку для інтелектуальних квестів, що базується на трьох основних ролях користувачів: Адміністратор, Викладач та Студент. Кожна роль має чітко визначений набір функціональних можливостей та взаємодіє з системою через централізований механізм аутентифікації.

Веб-додаток є кореневим елементом архітектури. Від нього відходять три основні гілки, що представляють різні ролі користувачів.

### 2.1.1. Роль адміністратора

1. Реєстрація (Register). Адміністратор має можливість реєструвати нових користувачів у системі.

2. Управління відділами:

- Додати, Редагувати відділ (Add, Edit Department) - Функціонал для створення та зміни інформації про відділи.

- Переглянути, Видалити відділ (View, Delete Department) - Можливість переглядати існуючі відділи та видаляти їх.

3. Управління курсами:

- Додати, Редагувати курс (Add, Edit Course) - Функціонал для створення та зміни інформації про навчальні курси.

- Переглянути, Видалити курс (View, Delete Course): Можливість переглядати існуючі курси та видаляти їх.

4. Управління викладачами:

- Додати, Редагувати викладача (Add, Edit Instructor) - Функціонал для реєстрації та зміни інформації про викладачів.

- Переглянути, Видалити викладача (View, Delete Instructor): Можливість переглядати існуючих викладачів та видаляти їх.

5. Перегляд інформаційної панелі (View Dashboard) - Доступ до загальної статистики та системної інформації.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

6. Перегляд студентів (View Students): Можливість переглядати список зареєстрованих студентів.

### 2.1.2. Роль викладача

1. Вхід (Login): Викладач входить у систему після реєстрації.

2. Управління студентами:

- Додати, Редагувати студента (Add, Edit Student) - Функціонал для додавання та зміни інформації про студентів (призначення до курсів/груп).

- Переглянути, Видалити студента (View, Delete Student): Можливість переглядати та видаляти записи студентів.

3. Управління іспитами (квестами):

- Додати, Редагувати, Видалити іспити (Add, Edit, Delete Exams) - Функціонал для створення, зміни та видалення інтелектуальних квестів.

- Переглянути майбутні, минулі іспити (View Upcoming, Past Exams) - Можливість переглядати заплановані та вже проведені квести.

4. Управління питаннями:

- Додати, Редагувати питання (Add, Edit Question) - Функціонал для створення та зміни питань до квестів.

- Переглянути, Видалити питання (View, Delete Question) - Можливість переглядати та видаляти існуючі питання.

5. Перегляд інформаційної панелі (View Dashboard) - Доступ до статистики, пов'язаної з його курсами та квестами.

6. Переглянути оцінки студентів за квести (View Student Quiz score): Можливість переглядати результати студентів.

### 2.1.3. Роль студента

1. Вхід (Login) - Студент входить у систему.

2. Переглянути майбутній іспит (View upcoming Exam) - Доступ до запланованих інтелектуальних квестів.

					БР.ІІІ – 38.00.00.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Пройти іспит (Take the exam) - Можливість безпосереднього проходження квесту.

4. Переглянути минулі іспити, оцінки (View Past Exams, Score): Доступ до історії пройдених квестів та їхніх результатів.

Всі ролі (крім первинної реєстрації адміністратора) проходять через централізовану систему входу. Після завершення роботи всі ролі мають можливість вийти з системи.

Ця архітектура забезпечує чітке розмежування відповідальності між різними типами користувачів, оптимізуючи робочі процеси в рамках освітнього середовища.

## **2.2. Взаємодії користувача та системи за допомогою діаграми варіантів використання**

Для детального аналізу функціональних вимог та відображення взаємодії між користувачами та розробленим веб-додатком для інтелектуальних квестів доцільно використовувати діаграму варіантів використання (Use Case Diagram). Цей інструмент моделювання, що є частиною Уніфікованої мови моделювання (UML), дозволяє візуалізувати функціональність системи з точки зору її зовнішніх користувачів (акторів) [8].

На рисунку 2.2 представлена діаграма варіантів використання, що ілюструє ключові взаємодії трьох основних ролей (акторів) у веб-додатку для інтелектуальних квестів: адміністратора, викладача та студента. Кожен варіант використання відображає певну функцію або послугу, яку система надає актору.

Розглянемо основні взаємодії за ролями.

1. Адміністратор:

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

- Включає реєстрацію нових користувачів, а також додавання, редагування та видалення облікових записів викладачів та студентів.
- Охоплює додавання, редагування та видалення відділів та курсів.
- Доступ до інформаційної панелі та перегляд списків студентів для загального контролю.

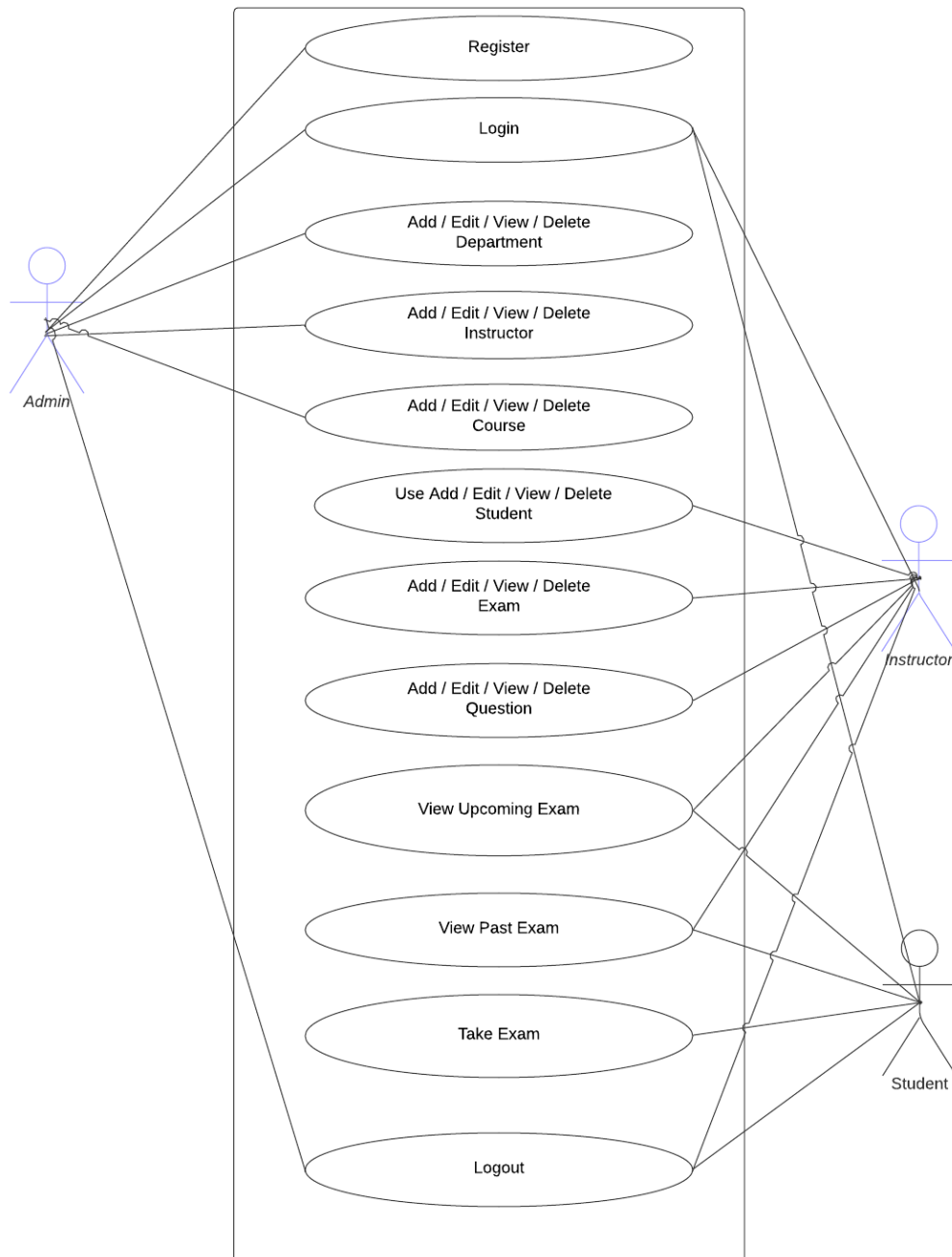


Рисунок 2.2 - Діаграма варіантів використання

## 2. Викладач:

- Додавання, редагування та видалення студентів, пов'язаних з їхніми курсами.
- Створення, редагування, видалення та планування квестів.
- Додавання, редагування та видалення питань до квестів.
- Доступ до інформаційної панелі та перегляд оцінок студентів за квести.

## 3. Студент:

- Перегляд майбутніх квестів та їх безпосереднє проходження.
- Доступ до своїх минулих квестів та отриманих оцінок.

Центральним елементом взаємодії для всіх ролей є процес аутентифікації (Вхід), який передусь доступу до специфічних функціональних можливостей. Завершення роботи з системою здійснюється через процес виходу (Вихід).

### 2.3. Розробка діаграми активності

Для детального моделювання динамічної поведінки системи та візуалізації послідовності дій у веб-додатку для інтелектуальних квестів використовуються діаграми активності (Activity Diagrams). Ці діаграми є графічними представленнями робочих процесів, що демонструють покрокові дії та активності, а також потоки керування між ними. Вони дозволяють відобразити логіку виконання операцій, паралельні процеси, розгалуження та об'єднання потоків. На рисунку 2.3 продемонстровано всі ключові активності та дії, що відбуваються у веб-додатку для інтелектуальних квестів, надаючи повне уявлення про його функціональні потоки.

Діаграма активності починається з початкового вузла, що символізує старт процесу. Система перевіряє, чи користувач зареєстрований

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

(Registered?). Якщо відповідь "Ні", користувач переходить до реєстрації (Register). Якщо відповідь "Так", користувач переходить до входу (Login).

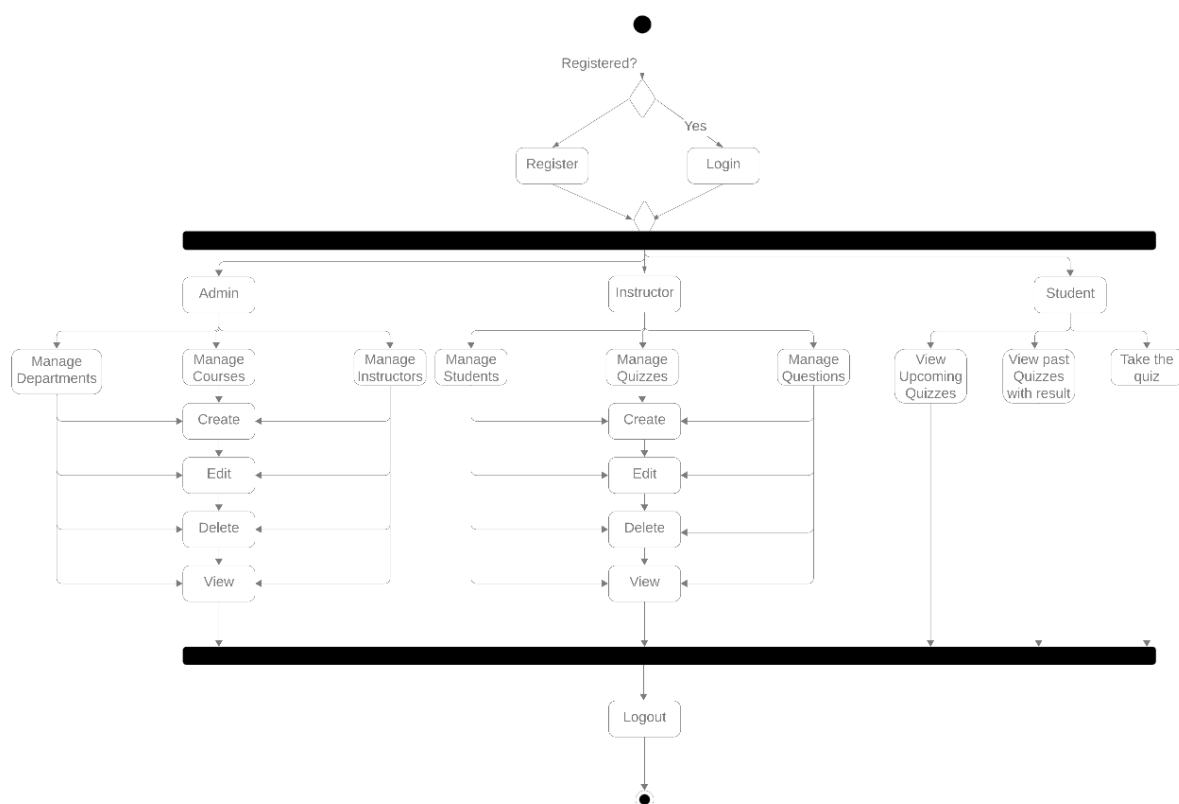


Рисунок 2.3 - Діаграма активності веб-додатку для інтелектуальних квестів

Після реєстрації/входу потік керування розгалужується на три паралельні гілки, що відповідають трьом основним ролям користувачів: адміністратор, викладач, студент. Кожна роль має свій набір доступних активностей.

а) Адміністратор:

- Керування відділами (Manage Departments): Включає активності Створити (Create), Редагувати (Edit), Видалити (Delete), Переглянути (View).
- Керування курсами (Manage Courses): Аналогічно, включає Створити, Редагувати, Видалити, Переглянути.
- Керування викладачами (Manage Instructors): Включає Створити, Редагувати, Видалити, Переглянути.

- Керування студентами (Manage Students): Включає Створити, Редагувати, Видалити, Переглянути.

б) Викладач:

- Керування квестами (Manage Quizzes): Включає активності Створити, Редагувати, Видалити, Переглянути.

- Керування питаннями (Manage Questions): Включає Створити, Редагувати, Видалити, Переглянути.

- Перегляд майбутніх квестів (View Upcoming Quizzes): Дозволяє переглядати заплановані квести.

- Перегляд минулих квестів з результатами (View past Quizzes with result): Дозволяє переглядати завершені квести та їхні оцінки.

в) Студент:

- Перегляд майбутніх квестів (View Upcoming Quizzes): Дозволяє студенту бачити доступні квести.

- Пройти квест (Take the Quiz): Активність безпосереднього проходження інтелектуального квесту.

- Перегляд минулих квестів з результатами (View past Quizzes with result): Доступ до історії своїх квестів та отриманих оцінок.

З усіх гілок активностей потік керування сходиться до активності Вихід (Logout), що завершує сесію користувача. Діаграма завершується кінцевим вузлом, що позначає завершення всіх процесів.

#### 2.4. Реалізація моделі сутність-зв'язок (ER Model)

Для проектування структури бази даних веб-додатку використовується модель сутність-зв'язок (Entity-Relationship Model, ER Model). Ця модель є фундаментальним інструментом для концептуального проектування баз даних, що дозволяє візуально представити основні сутності системи та

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

зв'язки між ними. У контексті реляційних баз даних, сутності зазвичай відповідають таблицям, а їхні атрибути – стовпцям цих таблиць.

ER-модель відіграє ключову роль у забезпеченні логічної структури даних, що є основою для ефективного зберігання, отримання та маніпулювання інформацією в системі. Вона дозволяє розробникам чітко визначити, які дані будуть зберігатися, як вони будуть організовані та як вони будуть взаємодіяти між собою.

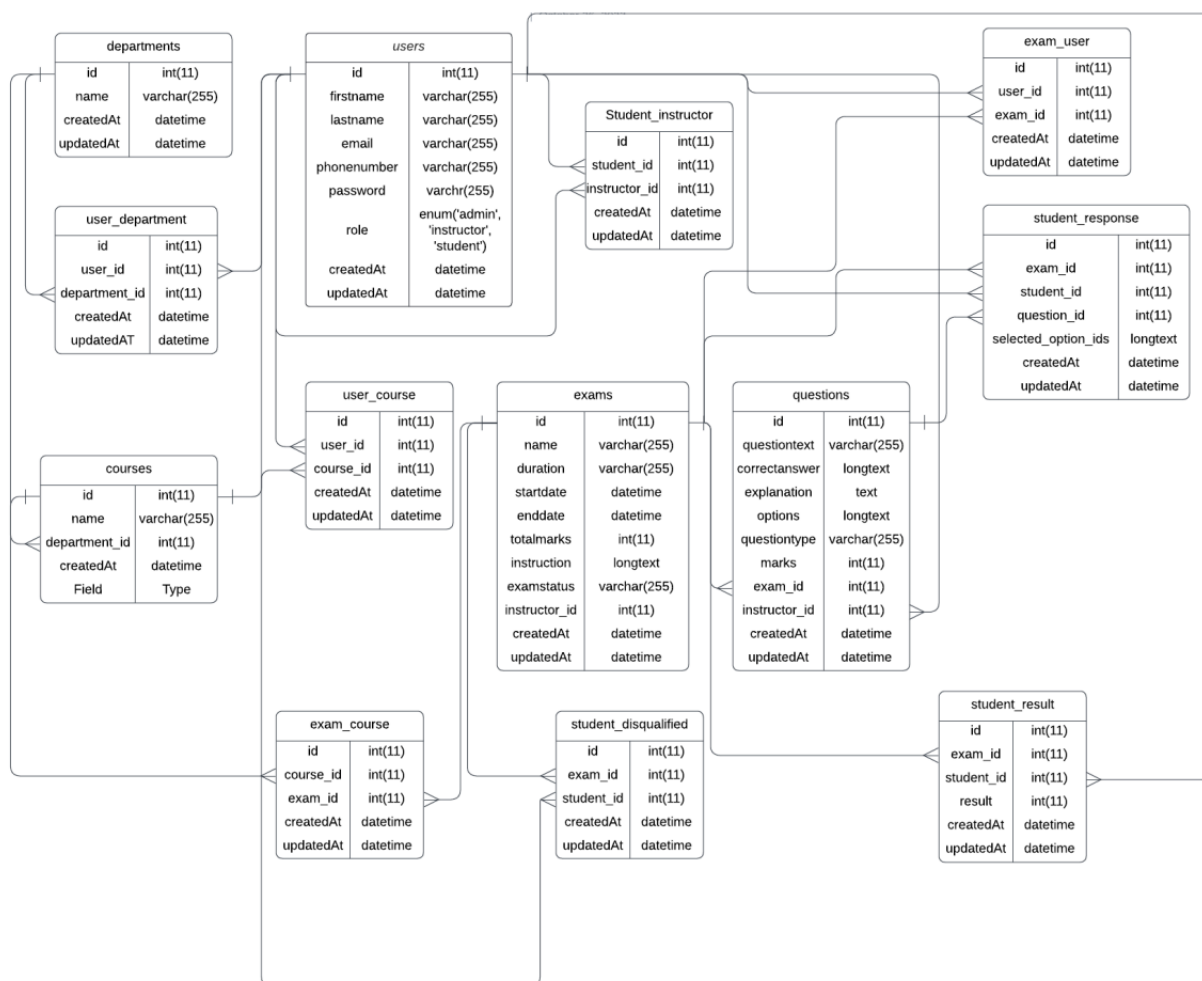


Рисунок 2.4 – ER діаграма бази даних

На рисунку 2.4 ілюструються відносини між кожною сутністю в базі даних веб-застосунку для інтелектуальних квестів. Це представлення дозволяє візуалізувати структуру даних, включаючи ключові поля, атрибути

та типи зв'язків (наприклад, "один-до-одного", "один-до-багатьох", "багато-до-багатьох") між таблицями, такими як користувачі, відділи, курси, квести, питання та результати.

Діаграма сутність-зв'язок (ERD) для веб-застосунку інтелектуальних квестів складається з наступних основних сутностей (таблиць) та їхніх взаємозв'язків:

#### 1. departments

Атрибути: id (PK, int), name (varchar), createdAt (datetime), updatedAt (datetime).

Описує структурні підрозділи освітнього закладу.

#### 2. users

Атрибути: id (PK, int), firstname (varchar), lastname (varchar), email (varchar), phonenumber (varchar), password (varchar), role (enum: 'admin', 'instructor', 'student'), createdAt (datetime), updatedAt (datetime).

Центральна сутність для управління всіма користувачами системи з різними ролями.

#### 3. user\_department

Атрибути: id (PK, int), user\_id (FK, int), department\_id (FK, int), createdAt (datetime), updatedAt (datetime).

Зв'язуюча таблиця для зв'язку "багато-до-багатьох" між users та departments, що дозволяє користувачам належати до кількох відділів або відділам мати багато користувачів.

#### 4. courses

Атрибути: id (PK, int), name (varchar), department\_id (FK, int), createdAt (datetime), updatedAt (datetime).

Описує навчальні курси, кожен з яких належить до певного відділу.

#### 5. user\_course

					БР.ІП – 38.00.00.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

Атрибути: id (PK, int), user\_id (FK, int), course\_id (FK, int), createdAt (datetime), updatedAt (datetime).

Зв'язуюча таблиця для зв'язку "багато-до-багатьох" між users та courses, що дозволяє користувачам (студентам або викладачам) бути пов'язаними з кількома курсами.

#### 6. Student\_instructor

Атрибути: id (PK, int), student\_id (FK, int), instructor\_id (FK, int), createdAt (datetime), updatedAt (datetime).

Зв'язує студентів з їхніми викладачами, ймовірно, для відстеження призначень або взаємодії.

#### 7. exams

Атрибути: id (PK, int), name (varchar), duration (varchar), startdate (datetime), enddate (datetime), totalmarks (int), instruction (longtext), examstatus (varchar), instructor\_id (FK, int), createdAt (datetime), updatedAt (datetime).

Описує самі інтелектуальні квести, включаючи їхні параметри та статус. Кожен квест пов'язаний з певним викладачем.

#### 8. exam\_course

Атрибути: id (PK, int), course\_id (FK, int), exam\_id (FK, int), createdAt (datetime), updatedAt (datetime).

Зв'язуюча таблиця для зв'язку "багато-до-багатьох" між exams та courses, що дозволяє одному квесту бути пов'язаним з кількома курсами, або курсу мати кілька квестів.

#### 9. questions

Атрибути: id (PK, int), questiontext (varchar), correctanswer (longtext), explanationtext (longtext), options (longtext), questiontype (varchar), marks (int), exam\_id (FK, int), instructor\_id (FK, int), createdAt (datetime), updatedAt (datetime).

Описує питання, що використовуються в квестах. Кожне питання належить до певного квесту та створене певним викладачем.

					БР.ІІІ – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

#### 10. exam\_user

Атрибути: id (PK, int), user\_id (FK, int), exam\_id (FK, int), createdAt (datetime), updatedAt (datetime).

Зв'язуюча таблиця для зв'язку "багато-до-багатьох" між users та exams, що вказує, які користувачі (студенти) призначені для проходження певних квестів.

#### 11. student\_response

Атрибути: id (PK, int), exam\_id (FK, int), student\_id (FK, int), question\_id (FK, int), selected\_option\_ids (longtext), createdAt (datetime), updatedAt (datetime).

Зберігає відповіді студентів на конкретні питання в рамках певного квесту.

#### 12. student\_disqualified

Атрибути: id (PK, int), exam\_id (FK, int), student\_id (FK, int), createdAt (datetime), updatedAt (datetime).

Записує інформацію про студентів, які були дискваліфіковані з певного квесту.

#### 13. student\_result

Атрибути: id (PK, int), exam\_id (FK, int), student\_id (FK, int), result (int), createdAt (datetime), updatedAt (datetime).

Зберігає підсумкові результати студентів за кожен пройдений квест.

Ця ER-діаграма надає вичерпне уявлення про логічну структуру бази даних, демонструючи, як різні сутності пов'язані між собою для підтримки функціональності веб-застосунку інтелектуальних квестів.

Враховуючи контекст розробки веб-додатку для інтелектуальних квестів, важливість розробки ER-діаграми (діаграми сутність-зв'язок) є фундаментальною та багатоаспектною.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

ER-діаграма є потужним інструментом для візуального представлення концептуальної моделі даних системи. Вона дозволяє розробникам та стейкхолдерам чітко зрозуміти, які ключові сутності (наприклад, "Користувачі", "Квести", "Питання", "Відділи") існують у системі, які атрибути їх характеризують (наприклад, ім'я користувача, тривалість квесту, текст питання) та як ці сутності взаємодіють між собою. Це особливо важливо для складної системи з трьома різними ролями користувачів (адміністратор, викладач, студент) та їх специфічними взаємодіями з даними.

ER-діаграма слугує безпосередньою основою для логічного та фізичного проектування реляційної бази даних. Кожна сутність на діаграмі зазвичай трансформується в таблицю бази даних, а атрибути – у стовпці. Зв'язки між сутностями (один-до-одного, один-до-багатьох, багато-до-багатьох) визначають потребу у зовнішніх ключах, проміжних таблицях та обмеженнях цілісності, що є критично важливим для забезпечення коректності зберігання даних. У контексті пропонованого додатку (quiz\_web\_departments, quiz\_web\_users, quiz\_web\_exams тощо), ERD є дорожньою картою для створення цих таблиць та їх зв'язків.

Чітко визначені зв'язки та обмеження на ER-діаграмі допомагають запобігти аномаліям даних (наприклад, видалення курсу, до якого все ще прикріплені студенти без відповідної обробки). Це забезпечує цілісність та узгодженість інформації в базі даних, що є життєво важливим для коректної роботи системи оцінювання та обліку навчального прогресу.

ER-діаграма виступає як універсальна мова для комунікації між технічною командою (розробники, архітектори баз даних) та нетехнічними стейкхолдерами (викладачі, адміністратори). Вона дозволяє обговорювати структуру даних, виявляти потенційні проблеми або невідповідності вимогам на ранніх етапах проектування, до початку імплементації, коли зміни є найбільш затратними.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

Детальна ER-діаграма є ключовим компонентом системної документації. Вона спрощує розуміння структури бази даних для нових членів команди, полегшує процеси підтримки, усунення несправностей та майбутніх модифікацій. Без ERD, розуміння складної схеми даних, що містить багато таблиць та зв'язків, було б значно ускладнене.

Правильно спроектована ER-діаграма допомагає уникнути надмірності даних та оптимізувати структуру для ефективного виконання запитів. Це безпосередньо впливає на продуктивність веб-додатку, особливо при великій кількості користувачів та операцій, таких як завантаження списку студентів, питань до квесту або результатів.

Чітка та логічна структура бази даних, розроблена на основі ERD, є запорукою масштабованості системи. Вона дозволяє легко додавати нові функції, сутності та розширювати обсяги даних без необхідності значних рефакторингів базової структури. Це особливо актуально, якщо в майбутньому планується перехід на хмарні сервіси та модель SaaS.

Підсумовуючи, розробка ER-діаграми для веб-додатку інтелектуальних квестів є невід'ємним етапом проектування, який забезпечує створення міцної, логічно обґрунтованої, підтримуваної та масштабованої основи для всієї системи.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

## РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ ДЛЯ АДМІНІСТРУВАННЯ ІНТЕЛЕКТУАЛЬНИХ КВЕСТІВ

### 3.1. Функціональні можливості розроблюваного веб-застосунку

Розроблений веб-застосунок для інтелектуальних квестів та вікторин є багатокористувацькою платформою, що функціонує на основі трьох дискретних ролей: адміністратор, викладач та студент. Кожна роль наділена суворо визначеними дозволами та функціональними можливостями, що забезпечує структурований та безпечний доступ до платформи.

Роль адміністратора охоплює комплексне управління освітнім закладом. Це включає адміністрування відділів, курсів та викладачів, що дозволяє підтримувати актуальну організаційну структуру та кадровий склад.

Роль викладача зосереджена на організації навчального процесу. Викладачі відповідають за управління студентами, розробку та проведення інтелектуальних квестів, а також формування питань, що відповідають їхнім курсам та відділам.

Роль студента призначена для безпосередньої взаємодії з навчальним контентом. Студенти мають можливість переглядати майбутні та минулі квести, ознайомлюватися з результатами пройдених квестів, а також брати участь у запланованих інтелектуальних квестах.

Розглянемо функціональні можливості розроблюваного веб-застосунку для різних ролей.

#### 1. Функції для адміністратора

Для забезпечення безпечного та ефективного доступу адміністратора до системи реалізовано компонент входу, що використовує механізм аутентифікації на основі JSON Web Token (JWT). Це гарантує, що лише авторизовані адміністратори можуть отримати доступ до функцій управління. Основні функціональні можливості адміністратора включають:

					БР.ІІІ – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

- Повний CRUD (Create, Read, Update, Delete) функціонал для відділів, що дозволяє створювати нові відділи, редагувати існуючі, переглядати їхній список та видаляти застарілі записи.

- Аналогічний CRUD функціонал для курсів, що забезпечує гнучке адміністрування навчальних програм.

- Можливість додавання, редагування, перегляду та видалення облікових записів викладачів.

- Доступ до інформації про студентів, включаючи їхні відповідні курси та відділи.

Ці функції дозволяють адміністратору ефективно керувати різними модулями освітнього закладу, забезпечуючи безперебійне функціонування платформи.

## 2. Функції для викладача

Доступ викладачів до системи також забезпечується через компонент входу, який верифікує їхні облікові дані, що зберігаються в базі даних. Функціонал для викладачів включає:

- Можливість додавання, редагування, перегляду та видалення студентів, пов'язаних з їхніми курсами.

- Повний CRUD функціонал для квестів, що дозволяє викладачам створювати, змінювати, переглядати та видаляти квести.

- Можливість додавання, редагування, перегляду та видалення питань, що використовуються в квестах.

- Викладачі можуть переглядати детальні результати кожного квесту, включаючи оцінки студентів та їхні відповіді.

## 3. Функції для студента

Студенти отримують доступ до системи через компонент входу, де їхні облікові дані також верифікуються. Основні функції для студентів:

- Студенти можуть бачити список запланованих інтелектуальних квестів.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

- Перегляд минулих квестів та оцінок тобто доступ до історії пройдених квестів та детальних результатів.

- Можливість брати участь у квестах з дотриманням встановлених часових обмежень.

Однією з ключових функцій веб-додатку, спрямованою на забезпечення академічної доброчесності, є автоматична дискваліфікація студента у випадку зміни активної вкладки браузера під час проходження квесту. При виявленні такої дії система автоматично перенаправляє студента на сторінку входу, а його поточний квест вважається недійсним. Крім того, для зручності та запобігання втраті даних, якщо студент проходить квест і час вичерпується, його відповіді автоматично надсилаються на сервер. Це усуває необхідність ручного підтвердження відправки та забезпечує збереження прогресу студента.

### 3.2. Початковий етап реалізації проекту

Початковий етап реалізації проекту включав налаштування середовища розробки та розгортання основних компонентів системи. Спочатку було встановлено Node.js версії v18.18.0 з офіційного веб-сайту, що забезпечило необхідне середовище виконання JavaScript на стороні сервера. Для ініціалізації та налаштування фронтенд-частини на базі React було використано команду `npm create-react-app quiz_web` у командному рядку. Паралельно з цим, було налаштовано компоненти Apache, MySQL та PHP в рамках інсталяції XAMPP, що забезпечило локальне середовище для розробки та тестування бази даних та веб-сервера.

Для розробки користувацького інтерфейсу (UI) були створені окремі компоненти та таблиці стилів з використанням HTML та SCSS. Навігація в односторінковому додатку реалізована за допомогою React Router Dom, а

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

управління станом та бізнес-логіка фронтенду – за допомогою Redux та бібліотек React Bootstrap для створення зручного та адаптивного інтерфейсу.

На стороні бекенду, використовуючи Node.js та фреймворк Express.js, було розроблено RESTful API. Ці API призначені для виконання різноманітних операцій, таких як управління відділами, курсами та автентифікацією користувачів. Маршрутизація API була організована у відповідних файлах для забезпечення модульності та зручності підтримки. Особлива увага була приділена правильній обробці авторизації та аутентифікації на рівні API для забезпечення безпеки даних.

З'єднання між бекенд-сервером та базою даних було встановлено за допомогою драйверів Sequelize, що дозволило ефективно виконувати операції CRUD (Створення, Читання, Оновлення та Видалення) з даними веб-додатку для інтелектуальних квестів (рис. 3.1).

```
const { Sequelize } = require('sequelize');
// Ініціалізація екземпляра Sequelize для підключення до бази даних.
// 'quiz_web' - назва бази даних, 'username' - ім'я користувача, 'password' - пароль.
const sequelize = new Sequelize('quiz_web', 'username', 'password', {
  host: 'localhost', // Хост бази даних.
  dialect: 'mysql' // Визначення діалекту бази даних (MySQL).
});

// Перевірка з'єднання з базою даних.
sequelize.authenticate()
  .then(() => {
    console.log('Connection has been established successfully.');// Повідомлення про успішне з'єднання.
  })
  .catch(err => {
    console.error('Unable to connect to the database:', err); // Повідомлення про помилку з'єднання.
  });
```

Рисунок 3.1 - З'єднання між бекенд-сервером та базою даних

Для встановлення зв'язку між фронтендом та бекенд-API використовувався модуль "HTTP". Зокрема, для розробки сервісів RESTful API, що дозволяють виконувати асинхронні дії в межах викликів API, було застосовано бібліотеку Axios.

```

const axios = require('axios');

// Виконання GET-запиту до API бекенду.
axios.get('http://localhost:3000/api/data')
  .then(response => {
    console.log(response.data); // Виведення отриманих даних.
  })
  .catch(error => {
    console.error('Error fetching data:', error); // Обробка помилок запиту.
  });

```

Рисунок 3.2 - Встановлення зв'язку між фронтендом та бекенд-API

Крім того, для локального розгортання та тестування веб-додатку, а також для забезпечення доступу до бази даних, було налаштовано сервери Apache та MySQL у програмі XAMPP.

### 3.3. Модель бази даних

Дизайн бази даних є критично важливим для забезпечення цілісності, ефективності та масштабованості системи. Приклад визначення таблиць departments та courses з їхніми атрибутами та зовнішнім ключем подано на рисунку 3.3.

```

-- Створення таблиці 'departments' для зберігання інформації про відділи.
CREATE TABLE departments (
  id INT AUTO_INCREMENT PRIMARY KEY, -- Первинний ключ, автоінкремент.
  name VARCHAR(255) NOT NULL        -- Назва відділу, обов'язкове поле.
);

-- Створення таблиці 'courses' для зберігання інформації про курси.
CREATE TABLE courses (
  id INT AUTO_INCREMENT PRIMARY KEY, -- Первинний ключ, автоінкремент.
  name VARCHAR(255) NOT NULL,        -- Назва курсу, обов'язкове поле.
  department_id INT,                 -- Зовнішній ключ, посилається на 'departments'.
  FOREIGN KEY (department_id) REFERENCES departments(id) -- Визначення зовнішнього ключа.
);

```

Рисунок 3.3 – Створення таблиць БД

Рисунок 3.3 демонструє загальну структуру бази даних, включаючи всі таблиці, їхні відповідні атрибути, обмеження (такі як первинні та зовнішні ключі) та встановлені відносини між ними.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

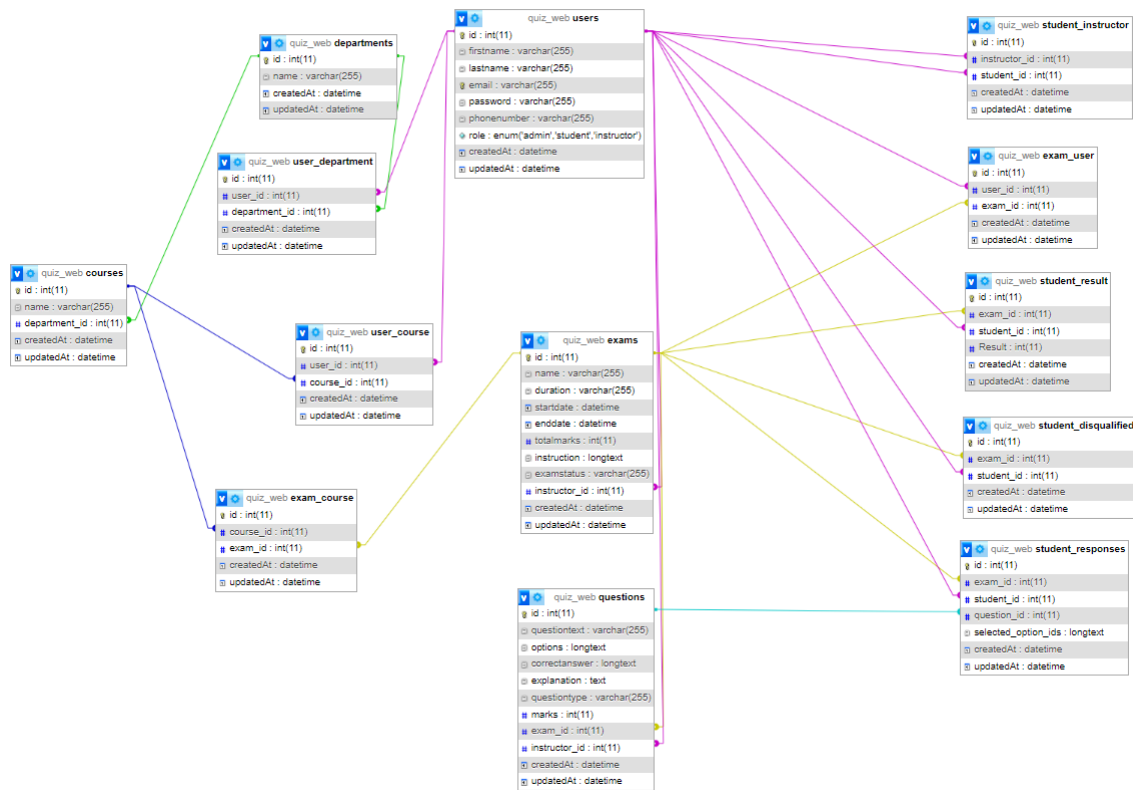


Рисунок 3.4 – Модель бази даних

Рисунок 3.4 ілюструє детальну схему реляційної бази даних для веб-додатку інтелектуальних квестів, відображаючи всі таблиці (сутності), їхні атрибути (стовпці) та зв'язки між ними. Ця схема є розширеною версією ER-моделі, показаної на Малюнку 6, і включає префікс `quiz_web_` для назв таблиць, що є типовим для IT проєктів.

### 3.4. Реалізація інтерфейсу користувача

Розроблений користувацький інтерфейс (UI) веб-додатку для інтелектуальних квестів спроектований з метою забезпечення інтуїтивно зрозумілої взаємодії та ефективного управління функціональними можливостями для всіх категорій користувачів. Для досягнення високого рівня візуальної привабливості та ефективності інтерфейсу було використано шаблон React Datta [9].

Змн.	Арк.	№ докум.	Підпис	Дата

Доступ до веб-застосунку здійснюється через сторінку реєстрації або входу. Сторінка реєстрації адміністратора доступна за адресою "localhost:3000/register". На рисунку 3.5 представлено інтерфейс для реєстрації адміністратора.

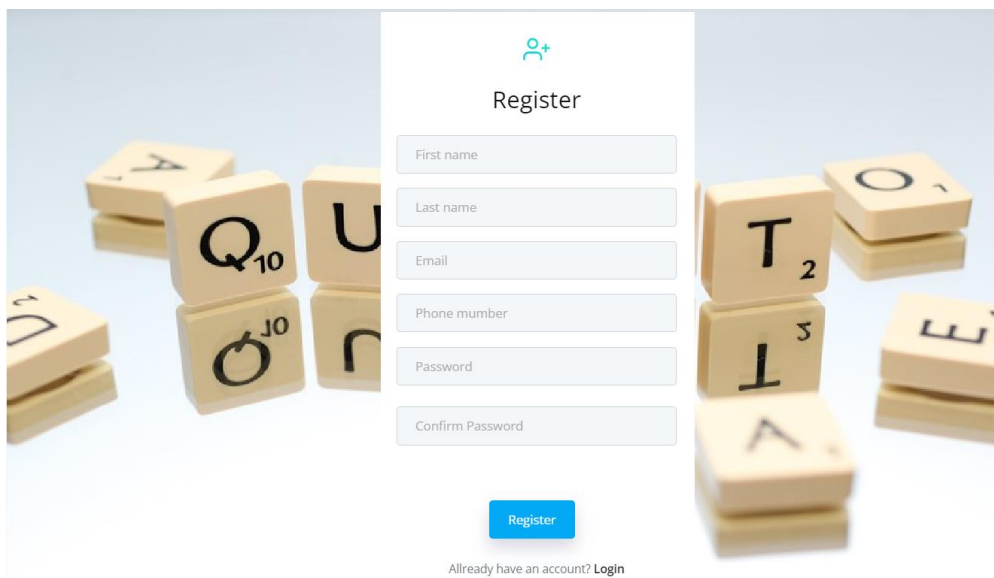


Рисунок 3.5 - Сторінка реєстрації адміністратора

На рисунку 3.6 зображено уніфіковану сторінку входу, призначену для адміністраторів, викладачів та студентів.

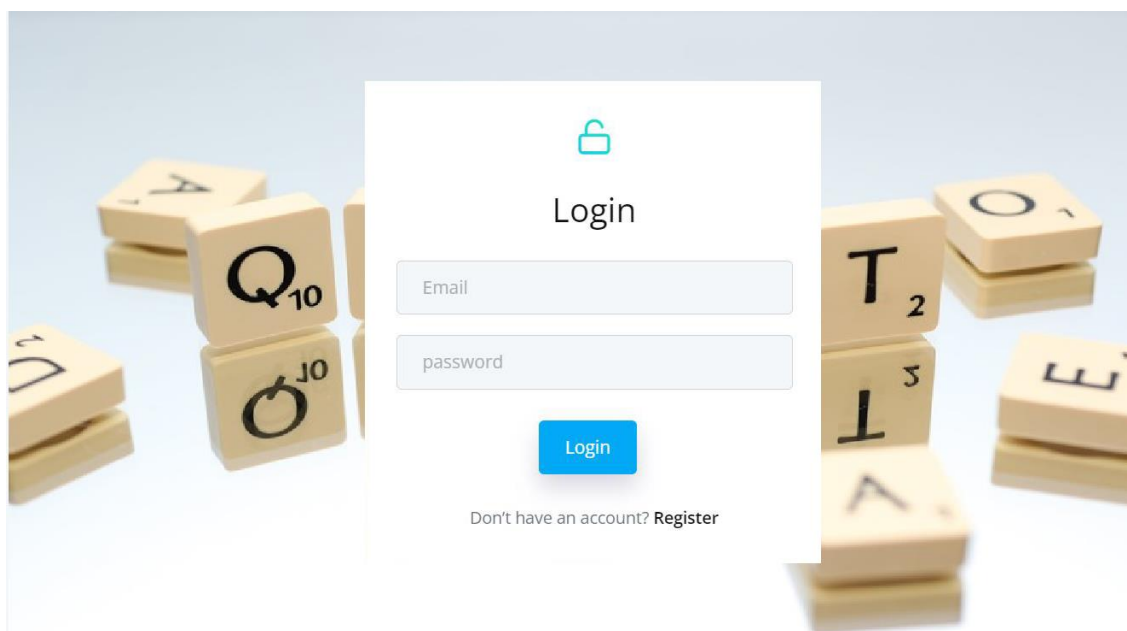


Рисунок 3.6 - Сторінка входу

У випадку введення користувачем некоректних облікових даних (неправильний email або пароль), система відображає повідомлення про помилку аутентифікації, як показано на рисунку 3.7.

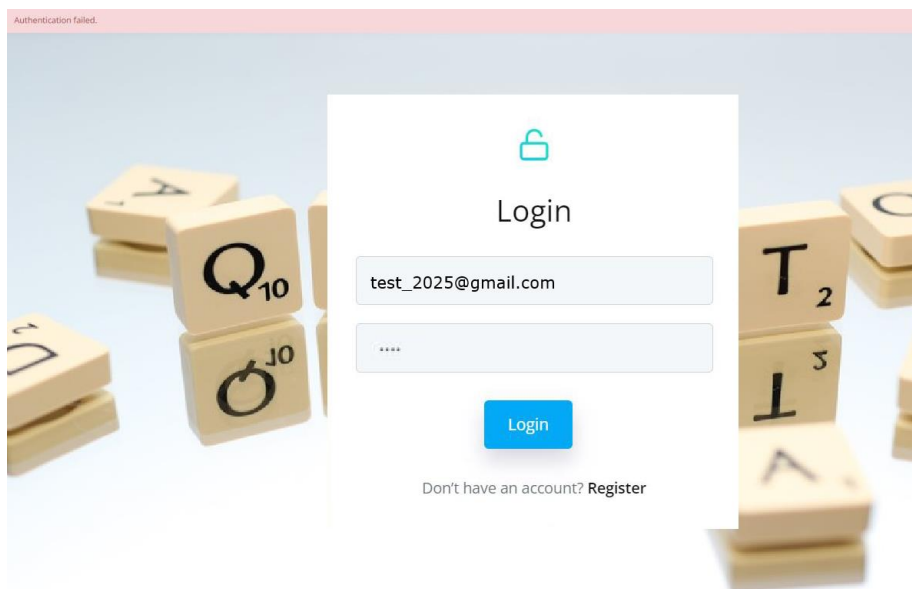


Рисунок 3.7 - Сторінка входу (помилка аутентифікації)

### 3.4.1. Інтерфейс адміністратора

Після успішної аутентифікації адміністратор автоматично перенаправляється на Панель управління (Dashboard) (рисунок 3.8). Ця панель є центральним вузлом для управління різними аспектами системи.

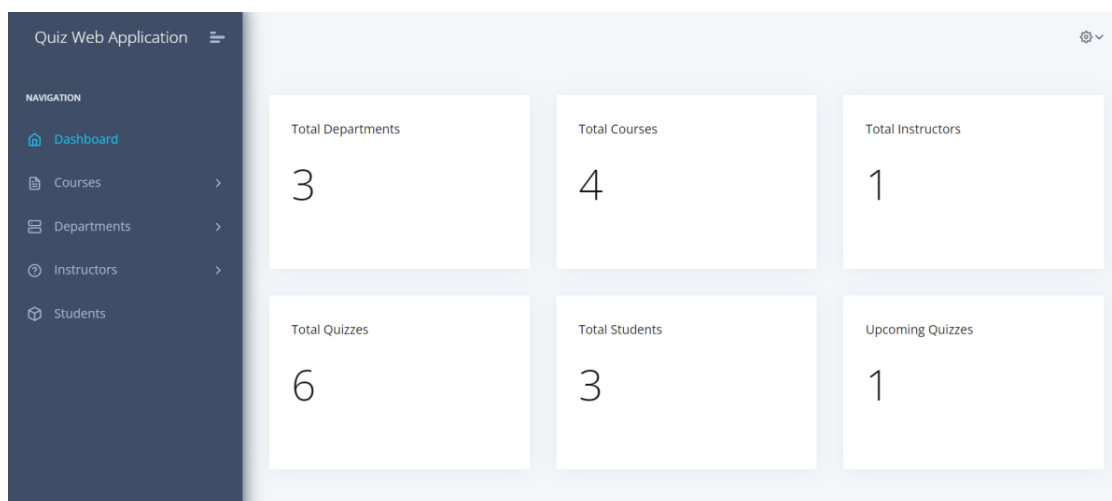


Рисунок 3.8 - Панель адміністратора

На рисунку 3.8 динамічна бічна панель забезпечує легкий доступ до модулів управління курсами, відділами, викладачами та студентами.

Адміністратор може переглядати та видаляти існуючі курси (рисунок 3.9), а також створювати та редагувати інформацію про курси (рисунок 3.10).

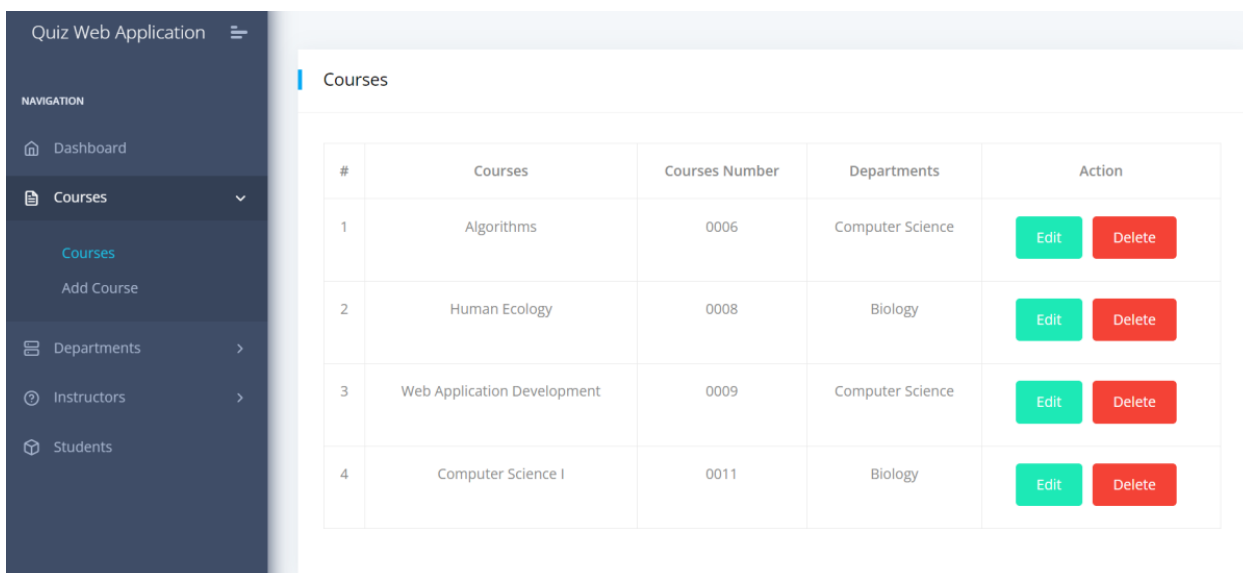


Рисунок 3.9 - Перегляд / Видалення курсів

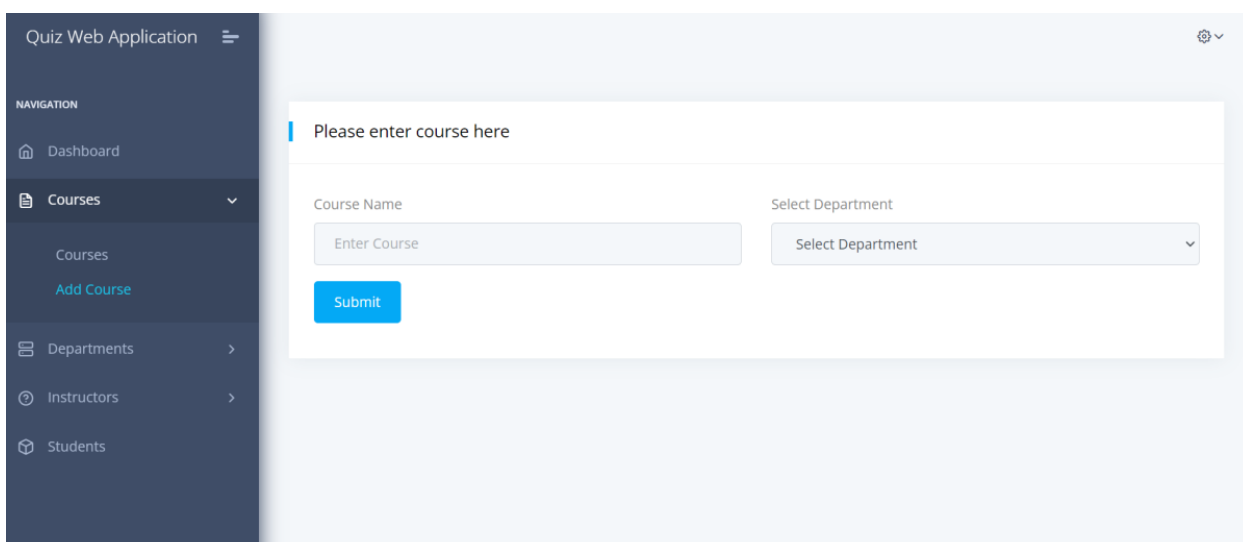


Рисунок 3.10 - Додавання / Редагування курсів

Аналогічно, адміністратор має можливість переглядати та видаляти відділи (рисунок 3.11), а також створювати та редагувати їх (рисунок 3.12).

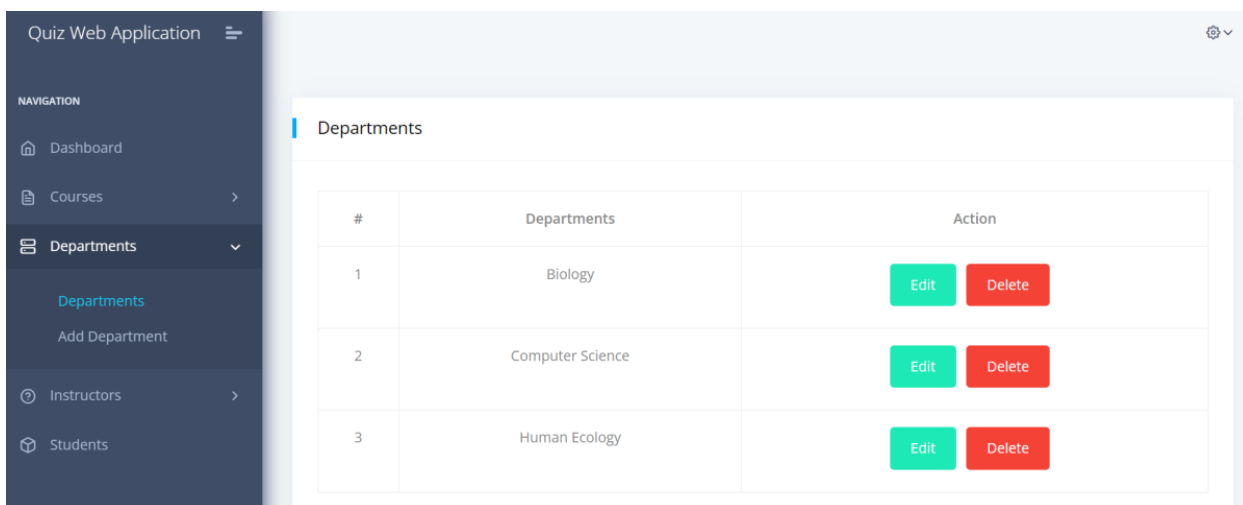


Рисунок 3.11 - Перегляд / Видалення відділів

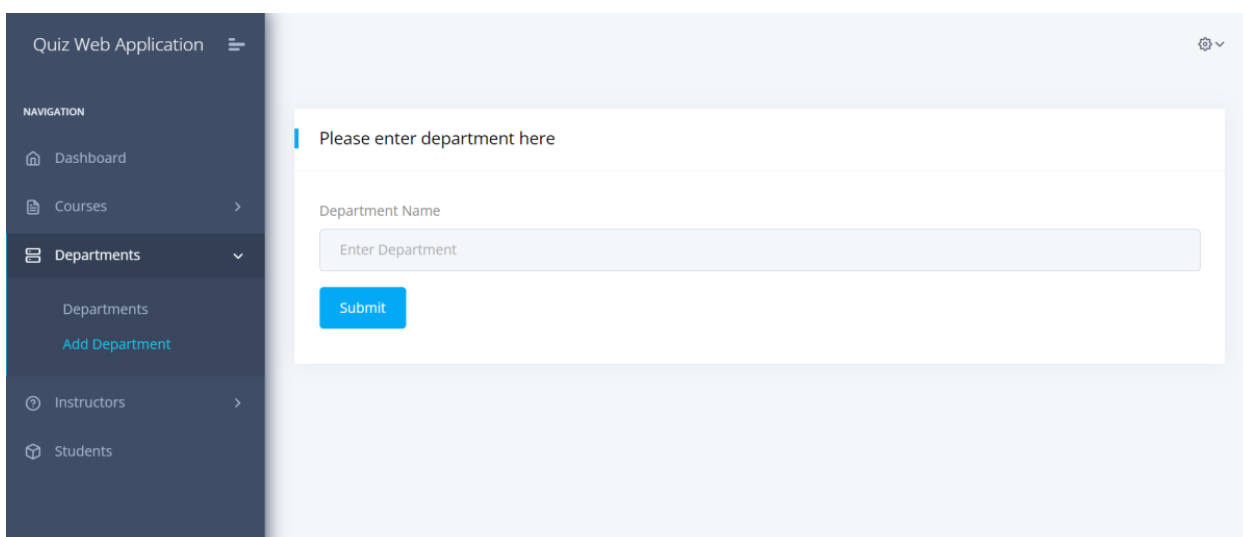


Рисунок 3.12 - Додавання / Редагування відділів

Також адміністратор може здійснювати управління викладачами, а саме він може переглядати та видаляти інформацію про викладачів, а саме їхні курси, відділи (рисунок 3.13), а також додавати та редагувати їхні дані (рисунок 3.14).

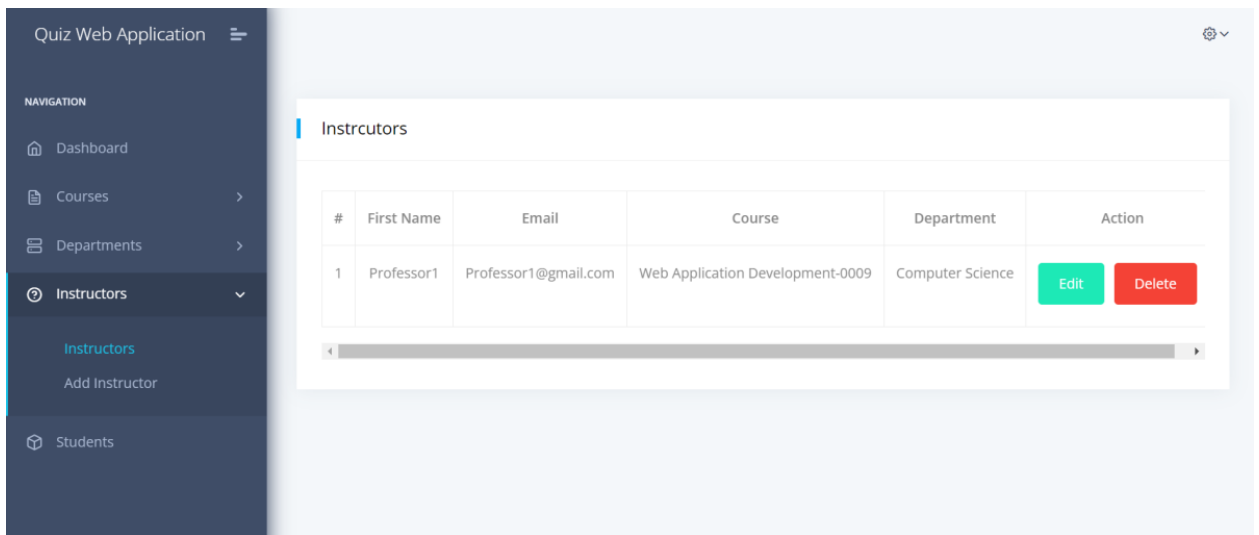


Рисунок 3.13 - Перегляд / Видалення викладачів

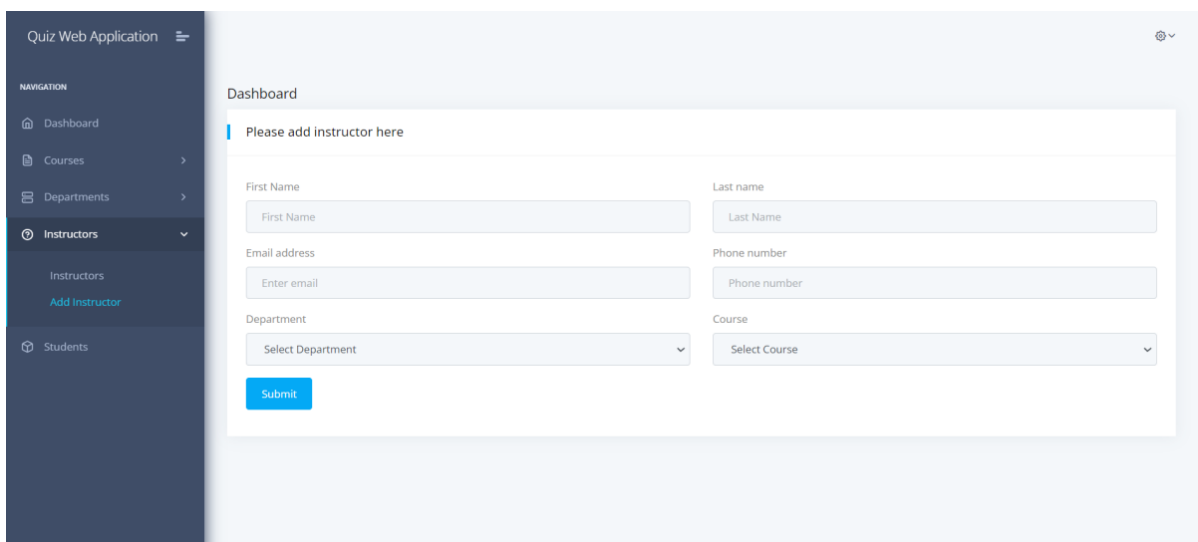


Рисунок 3.14 - Додавання / Редагування викладачів

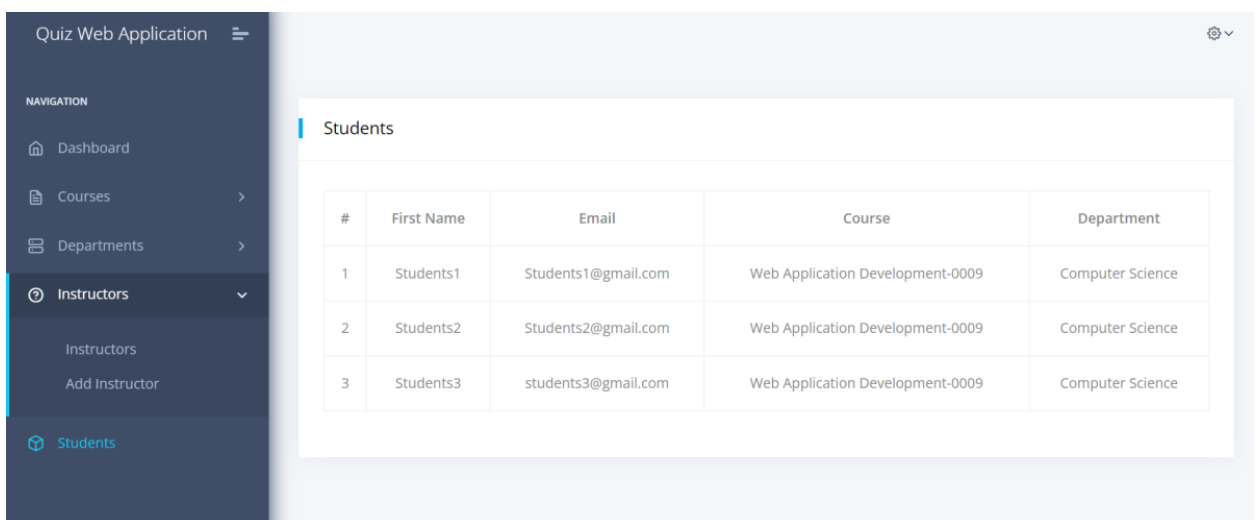


Рисунок 3.15 - Перегляд студентів

Також адміністратор має можливість перегляду студентів, тобто доступ до повного списку всіх зареєстрованих студентів (рисунок 3.15). Функціонал для безпечного виходу з веб-додатку показано на рисунку 3.16.

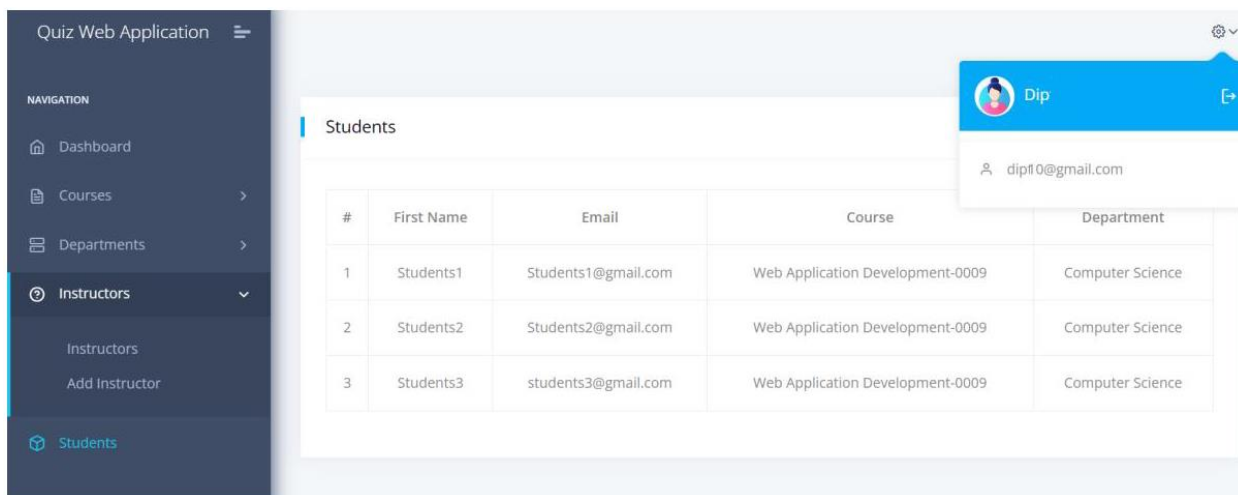


Рисунок 3.16 - Компонент виходу

### 3.4.2. Інтерфейс викладача

Після успішної аутентифікації викладач перенаправляється на власну панель управління (Dashboard) (рисунок 3.17). Динамічна бічна панель надає доступ до модулів управління студентами, інтелектуальними квестами та питаннями.

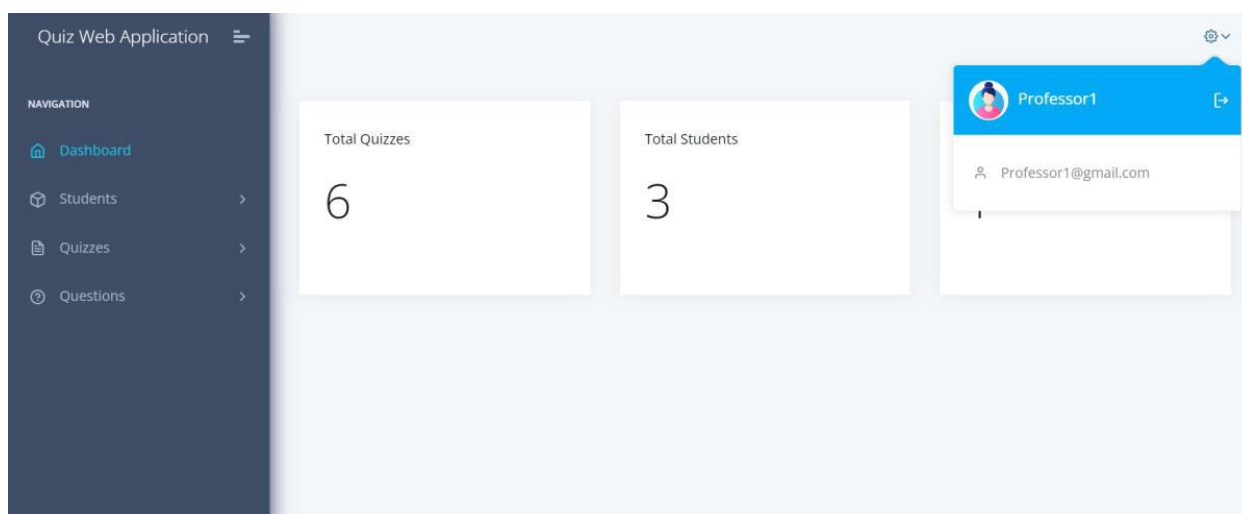


Рисунок 3.17 - Панель викладача

Викладач може створювати та редагувати записи студентів (рисунок 3.18), а також переглядати та видаляти їх (рисунок 3.19).

Quiz Web Application

NAVIGATION

- Dashboard
- Students
  - Students
  - Add Student
- Exams
- Questions

Please add student here

First Name:

Last name:

Email address:

Phone number:

Рисунок 3.18 - Додавання / Редагування студента

Quiz Web Application

NAVIGATION

- Dashboard
- Students
  - Students
  - Add Student
- Quizzes
- Questions

Students

#	First Name	Last Name	Email	Phone Number	Action
1	Students1	Students1	Students1@gmail.com		<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Students2	Students2	Students2@gmail.com		<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Students3	Students3	students3@gmail.com		<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Рисунок 3.19 - Перегляд / Видалення студента

Управління інтелектуальними квестами включає створення та редагування квестів (рисунок 3.20), перегляд минулих квестів (рисунок 3.21), аналіз результатів студентів за кожен квест (рисунок 3.22), а також перегляд та видалення майбутніх квестів (рисунок 3.23).

Quiz Web Application

NAVIGATION

- Dashboard
- Students
- Quizzes
  - Upcoming Quizzes
  - Past Quizzes
  - Create Quiz
- Questions

Please create quiz here

Quiz Name: Exam Name

Quiz Duration: Enter exam duration

Start Date: Start Date

End Date: End Date

Total Marks: 0

Submit

Рисунок 3.20 - Додавання / Редагування інтелектуального квесту

Quiz Web Application

NAVIGATION

- Dashboard
- Students
- Quizzes
  - Upcoming Quizzes
  - Past Quizzes
  - Create Quiz
- Questions

Past Quizzes

#	Quiz Name	Quiz Duration	Start Date	End Date	Total Marks	Action
1	Exam1	60	05-20-2025	05-21-2025	30	View Result Evaluate
2	Exam2	15	10-17-2025	10-18-2025	10	View Result Evaluate
3	Exam 3	15	05-26-2025	05-25-2025	15	View Result Evaluate
4	Exam 4	15	10-26-2025	10-27-2025	15	View Result Evaluate

Рисунок 3.21 - Перегляд минулих / Оцінка інтелектуального квесту

Quiz Web Application

NAVIGATION

- Dashboard
- Students
- Quizzes
  - Upcoming Quizzes
  - Past Quizzes
  - Create Quiz
- Questions

Student Result

#	Name	Email	Result
1	Students1 Students1	Students1@gmail.com	7

Рисунок 3.22 - Перегляд результату студента

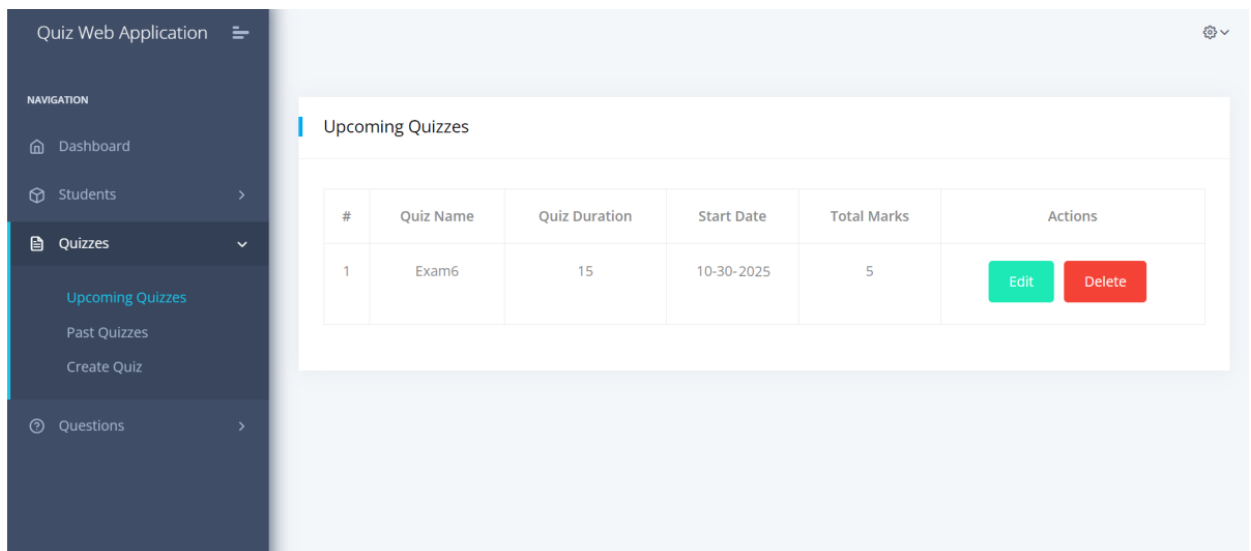


Рисунок 3.23 - Перегляд / Видалення майбутнього інтелектуального квесту

В ролі “викладач” користувач також може створювати та редагувати питання для квестів (рисунок 3.24), а також переглядати та видаляти їх (рисунок 3.25).

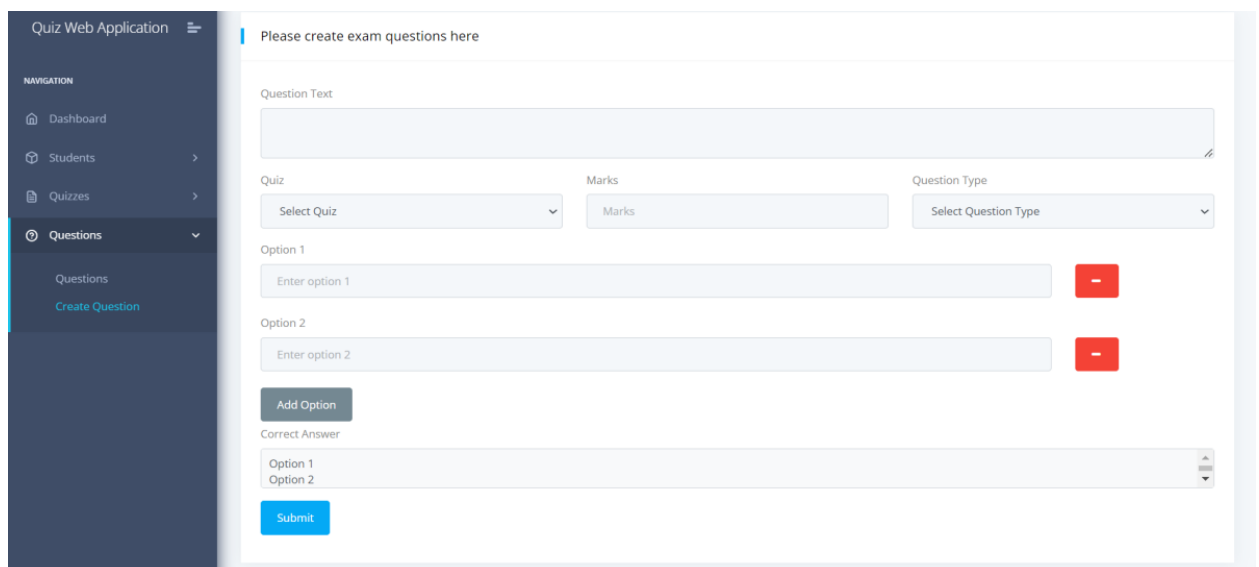


Рисунок 3.25 - Додавання / Редагування питання

Для додавання питання необхідно заповнити поле: "Question Text" - велике текстове поле для введення основного тексту питання. Вибір квесту: "Quiz" - випадаючий список "Select Quiz" для вибору, до якого квесту

належить це питання. Бали за питання: "Marks" - поле для введення кількості балів за це питання. Тип питання: "Question Type" - випадаючий список "Select Question Type" для вибору типу питання (наприклад, множинний вибір, відкрита відповідь тощо). Варіанти відповідей: "Option 1", "Option 2" - текстові поля для введення варіантів відповідей.

Правильна відповідь: "Correct Answer" - випадаючий список, який дозволяє вибрати правильний варіант(и) серед доданих опцій.

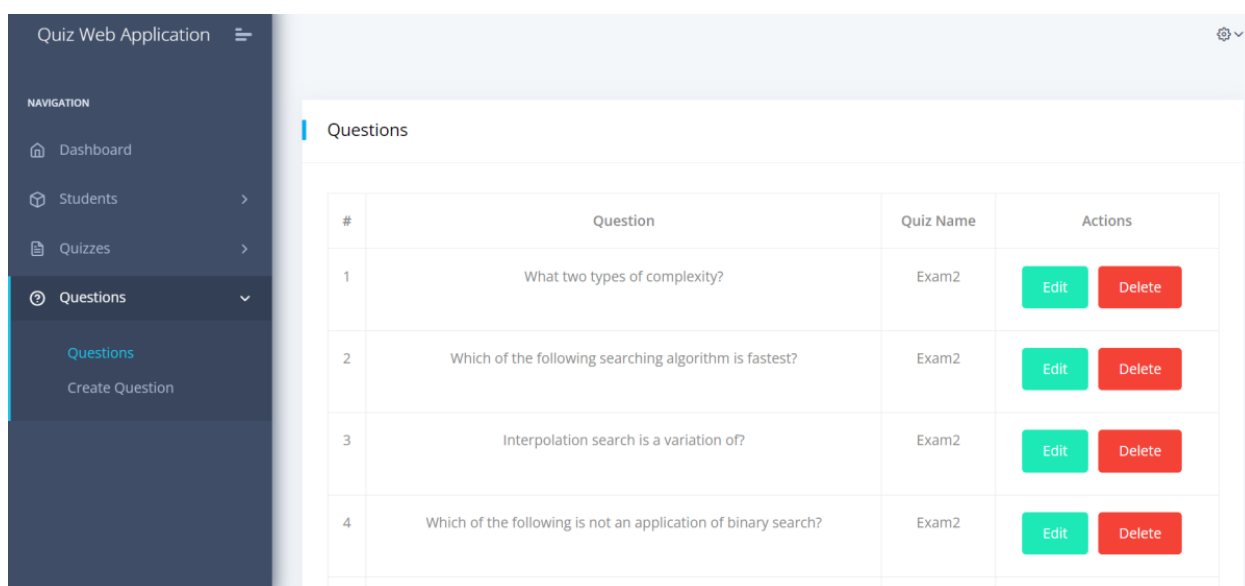


Рисунок 3.26 - Перегляд / Видалення питання

### 3.4.3. Інтерфейс студента

Після успішного входу студент автоматично перенаправляється на власну панель управління (Dashboard) (рисунок 3.27).

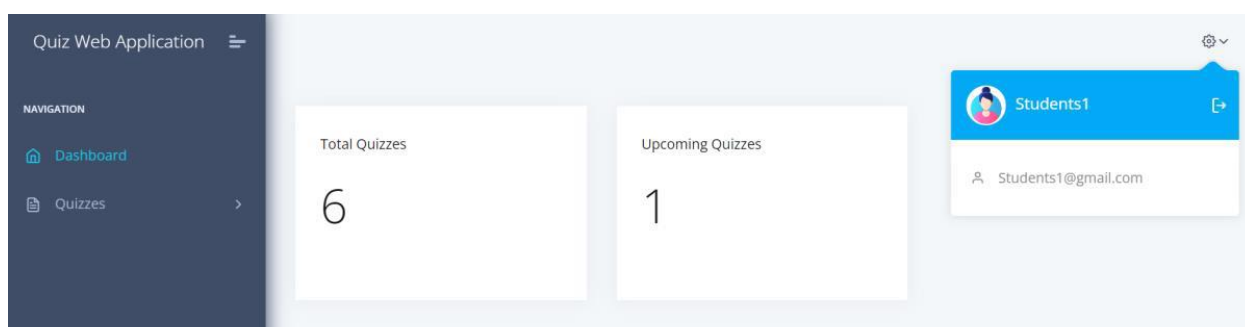


Рисунок 3.27 – Панель управління студента

Динамічна бічна панель надає доступ до екранів майбутніх квестів, минулих квестів та функціоналу для проходження квестів.

Студенти можуть переглядати список запланованих інтелектуальних квестів (рисунок 3.28).

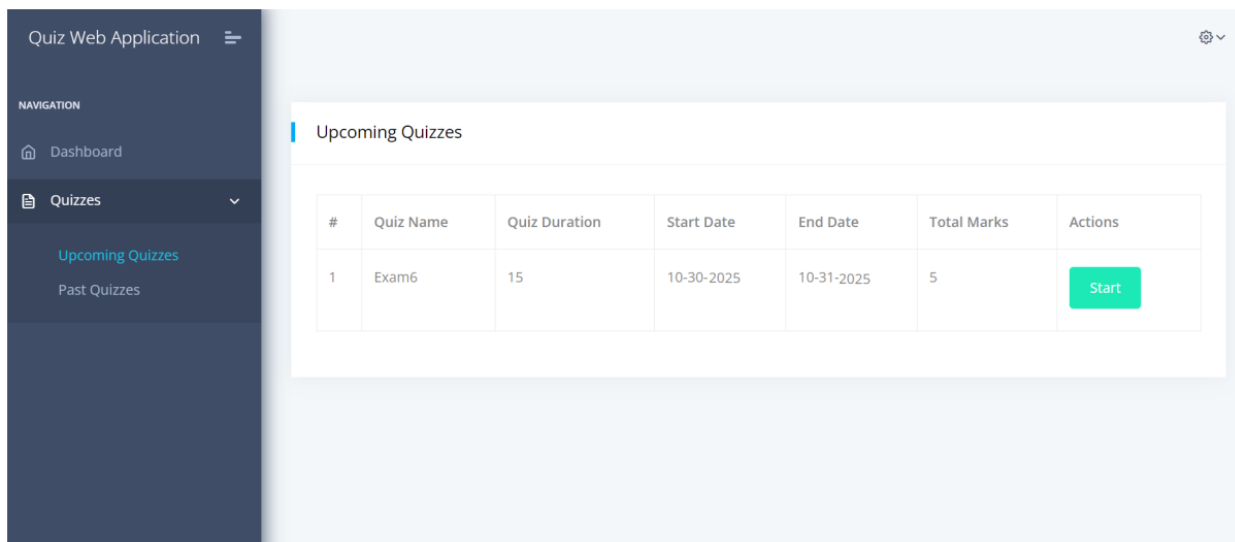


Рисунок 3.28 - Перегляд майбутнього інтелектуального квесту

Доступ до історії пройдених квестів та їхніх результатів (рисунок 3.29).

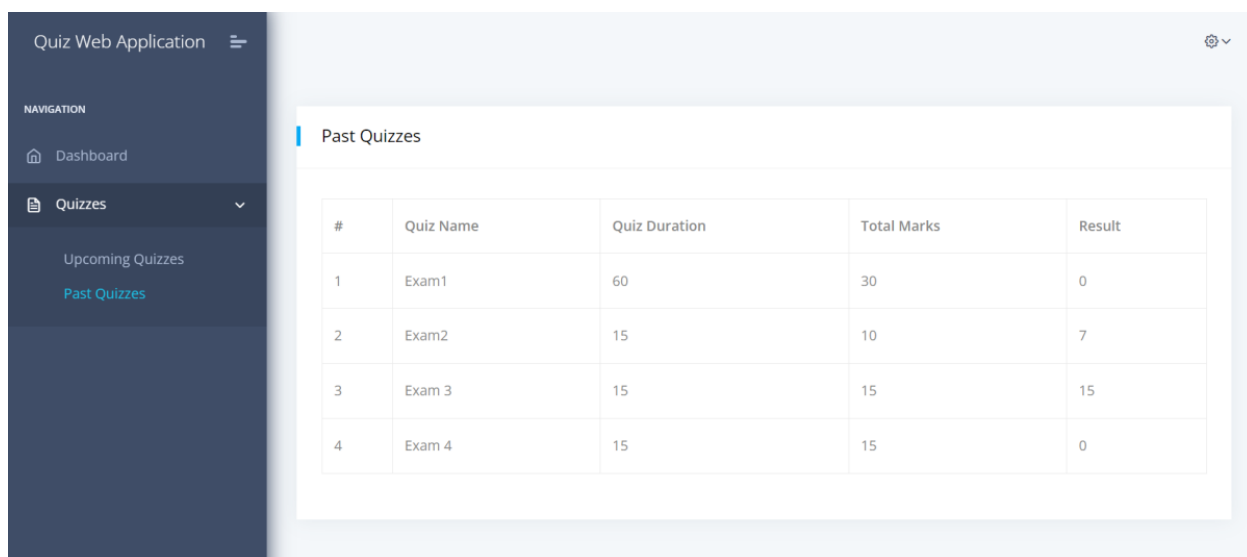


Рисунок 3.29 - Перегляд минулих інтелектуальних квестів

Проходження квесту або можливість безпосередньо брати участь у квесті подано на рисунку 3.30.

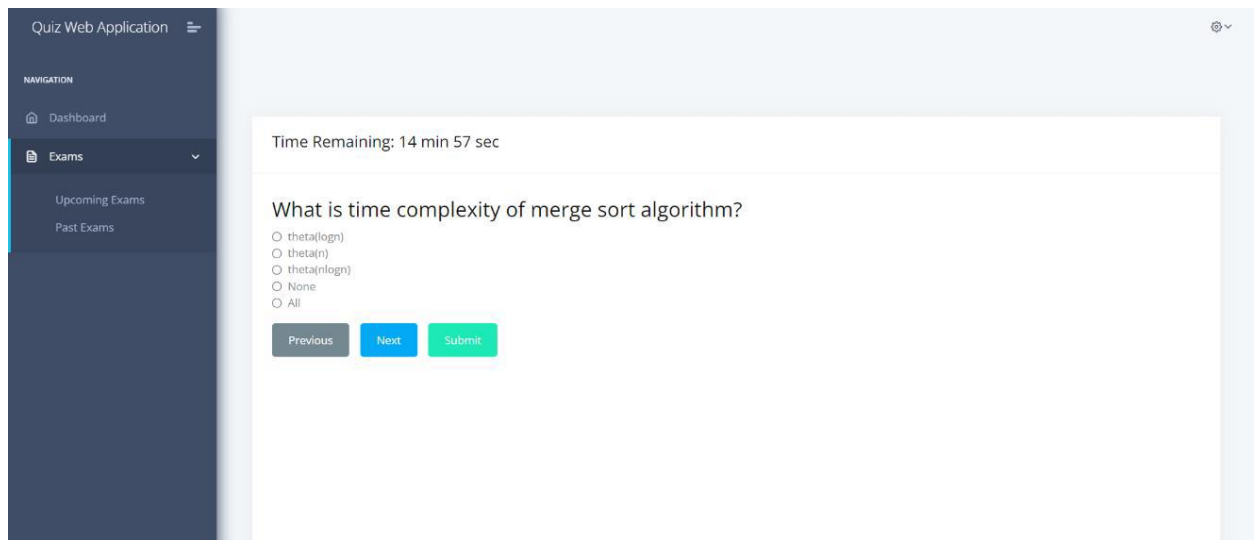


Рисунок 3.30 - Проходження інтелектуального квесту

На рисунку 3.31 показано екран, що з'являється після успішного надсилання відповідей студентом.

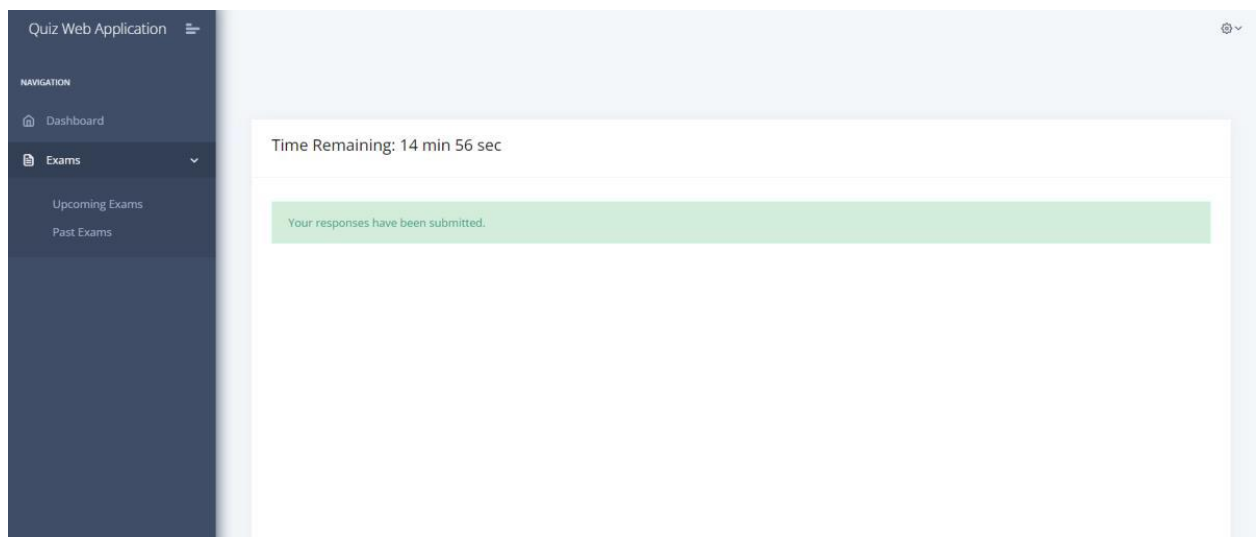


Рисунок 3.31 - Після надсилання інтелектуального квесту

Потужна та сумісна технологія бекенду веб-додатку надає викладачам розширені можливості для ефективного створення та конфігурування

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

інтелектуальних квестів. Суттєвою перевагою для викладачів є інтегрована функція автоматичної оцінки, що забезпечує високу точність та об'єктивність у процесі верифікації знань.

Веб-додаток для інтелектуальних квестів ефективно нівелює ризики академічної недоброчесності під час проведення дистанційних атестацій. Це досягається шляхом імплементації стратегічних контрольних механізмів, таких як обмеження часу на виконання завдань та моніторинг активності екрану під час тестування, що сприяє забезпеченню справедливої та об'єктивної оцінки знань студентів. Крім того, адаптивний та ергономічний користувацький інтерфейс гарантує зручність та доступність системи для студентської аудиторії.

Таким чином, веб-додаток для інтелектуальних квестів є цінним інструментом для дистанційного навчання та онлайн-освіти, усуваючи географічні обмеження. Застосування передових технологій дозволяє викладачам динамічно взаємодіяти зі студентами, сприяючи формуванню більш інтерактивного та залучаючого освітнього середовища.

На поточному етапі реалізації функціонал веб-застосунку для інтелектуальних квестів обмежується відображенням лише підсумкових результатів студентів. Проте, з метою подальшого вдосконалення системи, передбачається розробка механізму, який дозволить студентам переглядати власні відповіді після оприлюднення результатів, а також отримувати деталізований зворотний зв'язок від викладача щодо некоректних відповідей. Імплементація такої функціональності сприятиме ідентифікації студентами власних прогалин у знаннях та подальшому поглибленню їх розуміння навчального матеріалу. Крім того, перспективним напрямком розвитку є переведення веб-додатку на хмарну інфраструктуру, що дозволить його функціонування як веб-порталу на основі моделі "Програмне забезпечення як послуга" (SaaS).

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

Ще одним суттєвим напрямком майбутнього вдосконалення веб-застосунку для інтелектуальних квестів є розширення та імплементація додаткових протоколів безпеки, спрямованих на мінімізацію можливостей академічної недоброчесності під час проведення дистанційних атестацій. Одним із можливих рішень є впровадження систем моніторингу, які відстежуватимуть рухи або інші патерни поведінки студентів під час проходження квестів та ідентифікуватимуть будь-яку підозрілу активність. Реалізація таких заходів забезпечить цілісність та об'єктивність процесу оцінювання, створюючи більш безпечне та контрольоване середовище для тестування студентських знань.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

## ВИСНОВКИ

У межах виконання дипломної роботи було досліджено, спроектовано та реалізовано веб-застосунок для адміністрування інтелектуальних квестів, орієнтований на використання в освітньому середовищі з метою інтерактивного навчання та об'єктивного оцінювання знань.

На першому етапі було здійснено аналіз предметної області, що охоплює сучасні підходи до побудови інтерактивних навчальних веб-систем. Визначено основні вимоги до функціоналу, архітектури, апаратного та програмного забезпечення, а також обґрунтовано доцільність розробки спеціалізованого інструменту для організації інтелектуальних квестів. Проведено огляд актуальних технологій розробки, таких як React, Node.js, Express.js, Sequelize та MySQL, які забезпечують гнучкість, масштабованість та ефективність реалізації застосунку.

Порівняльний аналіз існуючих навчальних платформ (LMS, онлайн-тестування, MOOCs, систем онлайн-прокторингу) дозволив виявити обмеження наявних рішень у контексті реалізації адаптивних ігрових механік, характерних для інтелектуальних квестів. Це підтвердило потребу у створенні власного веб-застосунку, який поєднує функції адміністрування, контролю, навчання та гейміфікації.

На основі результатів моделювання було спроектовано архітектуру системи, побудовано діаграми варіантів використання та активності, а також реалізовано модель "сутність-зв'язок" для бази даних. Це забезпечило формалізацію взаємодій між користувачами різних ролей (адміністратор, викладач, студент) та внутрішніми компонентами системи.

У практичній частині реалізовано повнофункціональний веб-застосунок із модульною структурою. Інтерфейс користувача адаптований до різних ролей, що дозволяє ефективно керувати створенням, проведенням та аналізом інтелектуальних квестів. Застосунок забезпечує реєстрацію та

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

автентифікацію користувачів, створення завдань, формування квестів, збір результатів та формування звітів.

Таким чином, виконана робота продемонструвала можливість розробки ефективного інструменту для інтерактивного навчання, що може бути використаний як у формальній, так і в неформальній освітній діяльності. Розроблений веб-застосунок відповідає сучасним вимогам до адаптивності, масштабованості та функціональності, а також має потенціал до подальшого розвитку, зокрема інтеграції зі сторонніми освітніми платформами, розширення механізмів аналітики та впровадження адаптивного навчання на основі даних.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Smith, J. D., & Johnson, L. M. (2022). Modern Web UI Development with React and Component-Based Architectures. Academic Press.
2. Chen, S., & Li, H. (2021). "Designing User-Centric Educational Web Applications: A UX Perspective." Journal of Educational Technology & Society, 24(3), 155-168.
3. Brown, A. B. (2020). Responsive Web Design for E-Learning Platforms. Springer.
4. Wang, X., & Singh, R. (2023). "Evaluating the Effectiveness of UI Templates in Rapid Web Application Development." Proceedings of the International Conference on Web Information Systems Engineering (WISE), 287-296.
5. Davis, P. K. (2019). "The Role of Intuitive Interfaces in Enhancing Student Engagement in Online Quizzing Systems." Computers & Education, 140, 103621.
6. Miller, C. L., & White, K. R. (2022). "Accessibility and Usability in Educational Software: Guidelines and Best Practices." International Journal of Human-Computer Studies, 162, 102789.
7. Garcia, F., & Lopez, M. (2021). Building Scalable Frontends with Redux: A Comprehensive Guide. O'Reilly Media.
8. Thompson, E. G. (2020). "User Authentication Flows: Design Principles for Security and Usability." IEEE Transactions on Software Engineering, 46(11), 2320-2334.
9. Patel, R., & Sharma, S. (2023). RESTful API Design and Implementation with Node.js and Express.js. CRC Press.
10. Kim, Y. S. (2021). "Leveraging JWT for Secure API Authentication in Modern Web Applications." Journal of Network and Computer Applications, 178, 102958.

					БР.ІІІ – 38.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		74

11. Johnson, R. L., & Williams, K. A. (2022). Database Design and Management for Educational Systems. Elsevier.
12. Nguyen, M. T. (2020). "Scalable Database Architectures Using ORMs like Sequelize for Node.js Applications." Proceedings of the ACM Symposium on Applied Computing (SAC), 1201-1208.
13. Lee, D., & Park, S. (2021). "Performance Optimization of Node.js Backend Services for High-Traffic Web Platforms." International Journal of Web Information Systems, 17(2), 154-169.
14. Clark, S. A., & Babbage, A. N. (2019). Fundamentals of MySQL Database Management for Web Applications. Springer.
15. Green, B. D. (2022). "Asynchronous API Communication: Best Practices with Axios in Modern Web Development." Journal of Web Engineering, 21(5), 654-670.
16. Rodriguez, J. M. (2020). "Microservices Architecture for Educational Platforms: A Node.js Perspective." IEEE Conference on e-Learning, e-Business, Enterprise Information Systems, & e-Government, 1-6.
17. Al-Samarraie, H., & Al-Amoudi, A. (2021). "The Effectiveness of Online Quizzing in Enhancing Student Learning and Engagement: A Systematic Review." Education and Information Technologies, 26(3), 3291-3315.
18. Gaskill, D. B. (2022). Designing and Implementing Automated Grading Systems for Online Assessments. Routledge.
19. Adams, R. P. (2020). "The Role of E-Learning Platforms in Facilitating Remote Education: Challenges and Opportunities." British Journal of Educational Technology, 51(4), 1059-1073.
20. Kaur, A., & Sharma, V. (2023). "Interactive Quizzing Tools: A Comparative Analysis of Features and Pedagogical Implications." International Journal of Smart Education and Urban Society, 4(1), 45-62.
21. Wang, J., & Wu, L. (2019). Assessment in Online Learning: Principles and Practices. Stylus Publishing.

					БР.ІІІ – 38.00.00.000 ІІЗ	Арк. 75
Змн.	Арк.	№ докум.	Підпис	Дата		

22. Peterson, K. A., & Bell, L. R. (2021). "Feedback Mechanisms in Digital Learning Environments: Enhancing Student Self-Regulation." *Computers & Education*, 174, 104321.
23. Cross, S. T. (2022). "The Impact of E-Learning Technologies on Global Educational Accessibility." *Journal of Open, Flexible and Distance Learning*, 27(1), 1-17.
24. Evans, M. R. (2020). "Student Engagement in Digital Assessment: The Influence of Interface Design and Feedback." *Australasian Journal of Educational Technology*, 36(5), 1-15.
25. Dawson, P., & Sutherland-Smith, W. (2021). *Assessment and Academic Integrity in the Era of Online Learning*. Routledge.
26. Balaji, P. (2022). "Proctoring Technologies for Online Examinations: A Review of Current Practices and Future Trends." *Journal of Educational Technology Systems*, 50(4), 481-496.
27. Jones, T. L. (2020). "Strategies for Detecting and Preventing Cheating in Online Assessments: A Comprehensive Framework." *Innovations in Education and Teaching International*, 57(5), 580-591.
28. Lim, H. Y., & Soh, J. T. (2023). "Behavioral Biometrics for Enhanced Security in Online Proctoring Systems." *IEEE Transactions on Learning Technologies*, 16(1), 98-111.
29. Smith, K. (2021). "Maintaining Academic Integrity in High-Stakes Online Exams: A Case Study Approach." *Journal of Academic Integrity*, 20(1), 1-18.
30. Chen, L., & Guo, Y. (2019). "Real-time Cheating Detection in Online Examinations Using Machine Learning and Webcam Monitoring." *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT)*, 321-325.

31. Turner, R. J. (2022). "Ethical Considerations in the Deployment of AI-Powered Proctoring Systems." *Ethics and Information Technology*, 24(2), 241-255.
32. White, A. C. (2020). "The Effectiveness of Time Limits as an Anti-Cheating Measure in Online Quizzes." *Assessment & Evaluation in Higher Education*, 45(7), 1014-1025.

					БР.ІП – 38.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

## БІБЛІОГРАФІЧНА ДОВІДКА

**Тема дипломної роботи:** “Веб-застосунок для адміністрування інтелектуальних квестів”

Обсяг пояснювальної записки: 77 аркушів.

Дата закінчення роботи: 10 червня 2025 р.

Підпис студента \_\_\_\_\_