

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Медіасистем та технологій
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розробка навчального web-додатку з інтегрованим цифровим помічником
(тема)

Виконав:

здобувач 4 року навчання,

групи ВПВПС-21-2



Богдан АКИМОВ

(власне ім'я, прізвище)

Спеціальність 186 Видавництво та поліграфія
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма

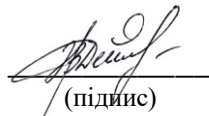
Видавничо-поліграфічна справа

(повна назва освітньої програми)

Керівник  доц. Олександр ЗЕЛЕНИЙ

(посада, власне ім'я, прізвище)

Допускається до захисту
Завідувач кафедри МСТ


(підпис)

Жанна ДЕЙНЕКО

(власне ім'я, прізвище)

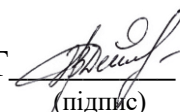
2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Медіасистем та технологій
Рівень вищої освіти перший (бакалаврський)
Спеціальність 186 Видавництво та поліграфія
Тип програми Освітньо-професійна
Освітня програма Видавничо-поліграфічна справа
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри МСТ



(підпис)

« 19 » травня 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

здобувачеві Акімову Богдану Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка навчального web-додатку з інтегрованим цифровим помічником

Затверджена наказом по університету від 19 травня 2025 р. № 385 Ст

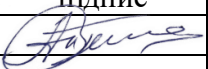
2. Термін подання здобувачем роботи до екзаменаційної комісії 11 червня 2025 р.

3. Вихідні дані до роботи: Національні та міжнародні стандарти оцінки якості додатків; проєкт на мові Java та JavaScript, реалізований на основі фреймворків React та Spring Boot в середовищі розробки Visual Studio Code, ОС Windows.

4. Перелік питань, що потрібно опрацювати в роботі
Вступ; Аналіз предметної області; Основні положення теорії електронного навчання; Системні вимоги; Опис архітектури системи; Системні вимоги; Вимоги до середовища розробки; Вимоги до зручності використання; Інтеграція цифрового помічника; Огляд реалізації e-learning платформи; Візуалізація структури бази даних; Виклики та подальший розвиток системи; Економічна частина; Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п. 5 включається до завдання за рішенням випускової кафедри)
Вступ; Мета та актуальність роботи; Аналіз предметної області; Мультимедійне навчання; Основні положення теорії електронного навчання; Системні вимоги; Опис архітектури системи; Огляд реалізації e-learning платформи; Вимоги до середовища розробки; Виклики та подальший розвиток системи; Економічна частина; Висновки.

6. Консультанти розділів роботи (п. 6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п. 1)

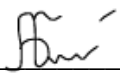
Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. Зелений О.П.		10.06.25
Економічна частина	ас. Легеза О.М.		10.06.25

КАЛЕНДАРНИЙ ПЛАН


№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу.	22.04	виконано
2	Аналіз предметної області.	24.04	виконано
3	Розробка постановки завдання.	27.04	виконано
4	Формулювання вимог.	01.05	виконано
5	Аналіз теоретичних відомостей.	05.05	виконано
6	Проектування програмної системи.	07.05	виконано
7	Формулювання висновків.	13.05	виконано
8	Написання пояснювальної записки.	16.05	виконано
9	Перевірка пояснювальної записки керівником.	19.05	виконано
10	Проходження нормоконтролю та оцінки від рецензента.	24.05	виконано
11	Економічна частина.	01.06	виконано
12	Захист кваліфікаційної роботи.	12.06	виконано

Дата видачі завдання 19 травня 2025 р.

Здобувач


(підпис)

Керівник роботи


(підпис)

доц. Олександр ЗЕЛЕНИЙ
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 75 с., 3 табл., 40 рис., 2 дод., 16 джерел.

ВЕБ-ДОДАТОК, E-LEARNING, ЦИФРОВИЙ ПОМІЧНИК,
НАВЧАЛЬНЕ СЕРЕДОВИЩЕ, ІНТЕРАКТИВНІСТЬ, REACT, СНАТВОТ.

Об'єкт дослідження – процес створення навчального веб-додатку з інтегрованим цифровим помічником для покращення якості засвоєння навчального матеріалу користувачами.

Мета роботи – розробка інтуїтивного навчального веб-додатку з вбудованим цифровим асистентом, що сприяє покращенню взаємодії користувача з платформою та підвищенню ефективності навчального процесу шляхом проходження вікторин, отримання зворотного зв'язку, підказок і рекомендацій у режимі реального часу.

Методи дослідження – використання мов програмування HTML, CSS, Java, JavaScript, бібліотеки React для реалізації інтерфейсу користувача, фреймворку Spring Boot для створення серверної частини, бази даних PostgreSQL, а також інтеграція цифрового помічника з використанням API та технологій платформи Botpress.

У результаті роботи проаналізовано потреби цільової аудиторії, розроблено архітектуру вебдодатку, реалізовано модуль створення та проходження вікторин, а також інтегровано інтелектуального помічника, який надає персоналізовану підтримку користувачу під час навчання.

Отримані результати можуть бути використані в системах формальної та неформальної освіти, у середовищах онлайн-курсів та платформах дистанційного навчання, сприяючи підвищенню зацікавленості та залучення студентів у навчальний процес.

ABSTRACT

Explanatory note: 75 p., 3 tab., 40 fig., 2 app., 16 sources.

WEB APPLICATION, E-LEARNING, DIGITAL ASSISTANT, LEARNING ENVIRONMENT, INTERACTIVITY, REACT, CHATBOT.

The object of the study is the process of creating an educational web application with an integrated digital assistant to improve the quality of learning material by users.

The purpose of the work is to develop an intuitive educational web application with a built-in digital assistant, which helps to improve user interaction with the platform and increase the efficiency of the learning process by taking quizzes, receiving feedback, tips and recommendations in real time.

Research methods are the use of programming languages HTML, CSS, Java, JavaScript, the React library to implement the user interface, the Spring Boot framework to create the server part, the PostgreSQL database, as well as the integration of the digital assistant using API and Botpress platform technologies.

As a result of the work, the needs of the target audience were analyzed, the architecture of the web application was developed, a module for creating and taking quizzes was implemented, and an intelligent assistant was integrated that provides personalized support to the user during training.

The results obtained can be used in formal and informal education systems, in online course environments and distance learning platforms, contributing to increasing student interest and engagement in the learning process.

ЗМІСТ

	С.
СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	8
ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Аналіз предметної галузі.....	11
1.2 Виявлення проблем та актуалізація рішень	12
1.3 Постановка задачі.....	14
2 ОСНОВНІ ПОЛОЖЕННЯ ТЕОРІЇ ЕЛЕКТРОННОГО НАВЧАННЯ	15
2.1 Мультимедійне навчання.....	15
2.2 Теоретичні підходи навчання	16
2.2.1 Саморегулювання	18
2.2.2 Теоретична основа.....	19
2.2.3 Застосування теорії навчання до електронного навчання.....	20
3 СИСТЕМНІ ВИМОГИ	21
3.1 Функціональні вимоги	21
3.2 Вимоги до даних.....	23
3.2.1 Структура бази даних.....	23
3.2.2 Забезпечення цілісності та безпеки даних	25
3.3 Вимоги користувача.....	26
3.4 Сценарії використання (Use Cases). UML-діаграми.....	28
3.5 Вимоги до середовища розробки.....	31
3.6 Вимоги до зручності використання (Usability requirements).....	32
4 ОПИС АРХІТЕКТУРИ СИСТЕМИ.....	35
4.1 Технології реалізації серверної частини (Back-End) та бази даних.....	38
4.1.1 Аутентифікація за допомогою JWT.....	39
4.1.2 Система зберігання даних – PostgreSQL	40
4.2 Технології реалізації клієнтської частини (Front-End).....	41
4.3 Використані інструменти розробки та середовище реалізації.....	42

5 ОГЛЯД РЕАЛІЗАЦІЇ E-LERNING ПЛАТФОРМИ	46
5.1 Функціонування системи з погляду користувача та адміністратора.....	46
5.1.1 Опис інтерфейсу користувача: робота з вікториною	47
5.1.2 Опис інтерфейсу адміністратора (admin)	54
5.2 Інтеграція цифрового помічника (чат-бота)	60
5.3 Візуалізація структури бази даних та заповнення тестовими даними ..	62
6 ВИКЛИКИ ТА ПОДАЛЬШИЙ РОЗВИТОК СИСТЕМИ.....	65
6.1 Масштабування даних та продуктивність PostgreSQL.....	65
6.2 Складність у синхронізації Front-End та Back-End.....	65
6.3 Проблеми з авторизацією через JWT	66
6.4 Захист персональних даних користувачів.....	66
6.5 Напрями подальшого розвитку системи	66
7 ЕКОНОМІЧНА ЧАСТИНА	68
7.1 Характеристика об'єкта економічного аналізу	68
7.2 Визначення витрат на розробку.....	69
ВИСНОВКИ	72
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	74
ДОДАТОК А Лістинг SQL-команди додавання даних про користувачів	76
ДОДАТОК Б Лістинг SQL-команди додавання питань вікторини	77

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

UI – User Interface – користувацький інтерфейс;

UX – User Experience – досвід взаємодії користувача з додатком;

E-Learning – Electronic Learning – електронне навчання, система навчання, за допомогою інформаційних, електронних технологій;

API – Application Programming Interface – інтерфейс прикладного програмування, що дозволяє програмам взаємодіяти між собою;

SPA – Single Page Application – односторінковий веб-додаток, який динамічно оновлює вміст без перезавантаження сторінки;

JWT – JSON Web Token – токен автентифікації у форматі JSON, що використовується для безпечної передачі інформації;

React – JavaScript-бібліотека для створення інтерактивних користувацьких інтерфейсів;

PostgreSQL – система керування реляційними базами даних;

SQL – Structured Query Language – мова програмування для роботи з реляційними базами даних;

Spring Boot – фреймворк для розробки серверної частини додатків на мові Java;

Botpress – платформа з відкритим кодом для створення чат-ботів та цифрових помічників;

CRUD – операції створення (Create), читання (Read), оновлення (Update) та видалення (Delete) даних у базі;

JSON – JavaScript Object Notation – текстовий формат для обміну даними між клієнтом і сервером;

HTTPS – HyperText Transfer Protocol Secure – захищений протокол передачі даних між клієнтом і сервером;

СУБД – Система Управління Базами Даних.

ВСТУП

Швидкий розвиток цифрових технологій кардинально трансформує всі сфери людської діяльності, зокрема освіти. Традиційні форми навчання дедалі частіше доповнюються або замінюються сучасними інтерактивними підходами, що орієнтовані на потреби користувача, його залученість та ефективність засвоєння знань. Дослідження показують, що гра є ключовим аспектом розвитку учнів, адже одним з найскладніших завдань в процесі навчання є підтримка залученості студента. Особливу роль у цьому процесі відіграє веб-розробка, яка дозволяє створювати зручні, адаптивні та доступні освітні інструменти. Потреба в доступі до інформації в режимі реального часу призвела до появи та подальшої інтеграції цифрових помічників у веб-додатки. Цифрові помічники, зокрема чат-боти, спрямовані на покращення та забезпечення персоналізованого користувацького досвіду, дають змогу підвищити якість навчання за рахунок постійного діалогу з користувачем, відповіді на запитання та надання контекстної допомоги. В умовах великої кількості інформації та високої динаміки навчального процесу потреба у таких рішеннях є вкрай актуальною. Ідея поєднання функціонального веб-додатку з можливістю взаємодії з інтелектуальним помічником є перспективним напрямом у сфері електронного навчання (з англ. «Electronic Learning», E-learning).

У межах цієї роботи розроблено платформу електронного навчання, веб-додаток з інтегрованим цифровим помічником, орієнтований на користувачів, які бажають перевірити або поглибити знання у сфері комп'ютерних наук. Dodatok realizovano z vikoristanniam suchasnoho steuku tehnologiy: React dla stvorennia interfejsu korystuvacha (UI), Spring Boot dla pobudovi servernoi logiki, PostgreSQL yak sistemi upravlinnia bazami danih, a takozh Botpress dla rozrobki i vbudovuвання чат-бота.

Ключовою особливістю є інтерактивна вікторина – ігрова форма навчання, яка активізує увагу користувача, стимулює мотивацію та сприяє кращому запам'ятовуванню навчального матеріалу. Вікторина містить питання з галузі інформаційних технологій і передбачає оцінювання результатів, формування рейтингу, а також генерацію рекомендацій з боку цифрового асистента. Крім того, цифровий помічник має здатність обробляти запити у довільній формі, генеруючи персоналізовані відповіді на основі історії взаємодії користувача із системою. Особливою перевагою є можливість спілкування з асистентом у режимі реального часу, що дозволяє отримувати негайні пояснення, уточнення або додаткову інформацію щодо будь-яких питань, які можуть виникнути під час проходження вікторини.

Розробка охоплює не лише технічні аспекти побудови клієнт-серверної архітектури, але й питання безпеки, зокрема реалізацію аутентифікації користувачів за допомогою JWT-токенів, розмежування ролей (адміністратор/користувач), збереження даних у зашифрованому вигляді, а також виведення статистики прогресу навчання.

Практична цінність створеного рішення полягає в його універсальності: веб-додаток може бути застосований як у формальній освіті (в школах і ВНЗ), так і в неформальному навчанні для самоосвіти або підвищення кваліфікації фахівців. Завдяки своїй адаптивності, модульності та простоті у використанні, платформа відкриває широкі можливості для масштабування та подальшого функціонального розширення (наприклад, багатомовність, гейміфікація, інтеграція з іншими освітніми сервісами).

Таким чином, робота спрямована на дослідження та практичну реалізацію сучасного підходу до організації освітнього процесу із застосуванням веб-технологій і штучного інтелекту. Отримані результати можуть бути використані як основа для створення нових цифрових навчальних платформ, що відповідають викликам сучасного інформаційного суспільства.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної галузі

Сфера електронного навчання (з англ. «Electronic Learning», E-learning) стрімко розвивається в умовах цифрової трансформації освітніх процесів. Веб-застосунки з освітнім контентом стали невіддільною частиною як формальної, так і неформальної освіти. Вони дозволяють створювати гнучкі, доступні та персоналізовані умови для навчання незалежно від місця перебування користувача. Особливої актуальності набуває поєднання таких застосунків із технологіями штучного інтелекту (ШІ), що дозволяє істотно підвищити ефективність взаємодії між користувачем та навчальним середовищем [1-3, 12].

Одним із перспективних напрямів розвитку освітніх платформ є інтеграція цифрових помічників – інтелектуальних систем, здатних надавати підтримку в реальному часі, консультувати відповідно до рівня знань користувача та сприяти формуванню індивідуальної траєкторії навчання.

Розробка навчального веб-додатку з інтегрованим цифровим помічником поєднує функціональні можливості повноцінної платформи для надання знань та систему інтелектуального супроводу користувача [4, 13]. Основу такого застосунку становить взаємодія фронтенду, бекенду, бази даних та API-зв'язків, що разом утворюють багаторівневу архітектуру.

Ретельний аналіз існуючих методів та інструментів веб-розробки показує доцільність використання бібліотеки React мови JavaScript для фронтенд частини, яка дозволяє створювати адаптивний інтерфейс користувача, зосереджений на зручності UX/UI. Застосування компонентного підходу та віртуальної об'єктної моделі документа (з англ. «Document Object Model», DOM) сприяє ефективному оновленню даних і швидкодії веб-додатку [1, 3-7].

Для реалізації бекенд частини, що відповідає за логіку додатку, обробку запитів користувача та взаємодію з цифровим помічником, раціональним є використання Java-фреймворку Spring Boot який забезпечує стабільність, безпечне з'єднання, просту інтеграцію з базами даних (через PostgreSQL), а також підтримку масштабованості системи [5, 14-16].

Інноваційною особливістю побудови e-learning web-застосунку є інтеграція технологій та систем штучного інтелекту [1, 7, 13]. Одним з найпоширеніших виступає цифровий помічник у вигляді чат-боту, який виступає допоміжним інструментом для користувача під час його взаємодії з застосунком, консультуючи його та надаючи корисну інформацію для вирішення поставлених йому питань. Крім того, цифровий помічник має змогу приймати будь-яку форму питань, генеруючи персоналізовані відповіді на основі історії користувача. Особливістю цифрового помічника є можливість спілкування в режимі реального часу, що дозволяє користувачам запитувати роз'яснення або додаткову інформацію з будь-яких питань, які можуть виникати під час роботи.

У ході проведення порівняльного аналізу існуючих методів та платформ побудови цифрових помічників було виявлено, що ефективним кроком для досягнення поставленої мети роботи є розробка цифрового помічника за допомогою платформи Botpress, яка підтримує обробку природної мови (з англ. «Natural Language Processing», NLP) та побудову багатопланових систем сценаріїв діалогу з користувачем [8, 9]. Таким чином, побудований помічник буде мати змогу надавати консультації, пояснення до питань вікторини та генерувати відповіді на основі контексту, створюючи індивідуальний досвід навчання.

1.2 Виявлення проблем та актуалізація рішень

Незважаючи на стрімкий розвиток e-learning систем, існують проблеми, які обмежують ефективність онлайн-навчання [10]. Однією з них є брак

персоналізації – користувачі часто отримують універсальний контент без врахування своїх індивідуальних потреб, знань і темпу навчання. Також недостатньо реалізованою є інтерактивна підтримка у процесі навчання, коли користувач може задати уточнююче запитання або отримати пояснення в реальному часі.

Рішенням зазначених проблем є використання ігрових механізмів для залучення користувачів та підвищення мотивації, таких як вікторина (з англ. «quiz») – ефективний та найпростіший метод гейміфікації, який впроваджує ігрові механізми для залучення користувачів та підвищення мотивації, та можуть бути використані для навчання, тестування знань, а також для розваг та змагань. Іншими методами підвищення зацікавленості можуть виступати системи рейтингів, балів, тощо.

Проблематику недостатньої персоналізації при взаємодії з навчальними платформами гарно вирішують завдяки інтеграції цифрових помічників, що забезпечують персоналізовану взаємодію за допомогою методів ШІ. Такий помічник у складі навчального веб-додатку дозволяє:

- адаптувати відповіді до рівня знань користувача;
- надавати зворотний зв'язок у режимі реального часу;
- зменшити когнітивне навантаження за рахунок пояснень та підказок.

Невід'ємною складовою побудови будь-яких веб-додатків та освітніх платформ, які обробляють чутливу інформацію, є захист персональних даних. В цьому випадку доцільним є розробка і використання JWT-аутентифікації (з англ. «JSON Web Token»), що забезпечує безпечне підключення користувачів різних рівнів та захист персональних даних, шляхом створення сервером так званих «токенів», що підписуються секретним ключем і передаються «клієнту», який надалі використовує цей токен для підтвердження своєї особи.

1.3 Постановка задачі

Після аналізу предметної області та виявлення недоліків наявних рішень, стає очевидною доцільність створення навчального веб-додатку з інтерфейсом вікторини та інтегрованим цифровим помічником у вигляді чат-боту. Тобто платформа онлайн-навчання, що:

- підтримує ролі адміністратора (вчителя) та звичайного користувача (учня) з відповідними правами доступу;
- дозволяє реєструвати користувачів, відстежувати їх прогрес і результати;
- надає питання з галузі інформаційних технологій у тестовій формі;
- інтегрує ШІ-помічника на базі Botpress для забезпечення інтелектуальної підтримки користувача в реальному часі;
- ґрунтується на використанні сучасного технологічного стеку, зокрема React, Spring Boot та PostgreSQL;
- реалізує чітке розмежування фронтенд- та бекенд-частин для забезпечення масштабованості та гнучкості системи.

Таким чином, розробка даного веб-додатку має на меті не лише надати освітній контент, а й забезпечити якісну підтримку користувача, сприяючи підвищенню ефективності онлайн-навчання та індивідуалізації освітнього процесу.

2 ОСНОВНІ ПОЛОЖЕННЯ ТЕОРІЇ ЕЛЕКТРОННОГО НАВЧАННЯ

В теперішній час онлайн-спілкування кардинально змінює спосіб взаємодії і навчання викладачів та учнів. Методи та технології електронного навчання (з англ. «E-Learning») широко використовується в навчальних закладах різних рівнів як доповнення до традиційного очного навчання в аудиторіях, або як основний інструмент взаємодії у закладах безперервної та дистанційної освіти.

Теорія електронного навчання описує принципи когнітивної науки ефективного мультимедійного навчання з використанням електронних освітніх технологій [1, 3, 12].

У сучасних цифрових освітніх платформах активно впроваджуються принципи мультимедійного дизайну навчання, що сприяє підвищенню ефективності онлайн-освіти. Зокрема, інноваційні системи електронного навчання пропонують збалансовану комбінацію візуального та текстового контенту, забезпечуючи структуровану подачу матеріалу та можливість навчання у власному темпі. Такий підхід особливо ефективний у віртуальних навчальних середовищах (ВНС), де якісно розроблені мультимедійні компоненти здатні не лише відтворювати, а й перевершувати традиційну класну взаємодію. Інтерактивні елементи – зокрема вікторини, візуальні підказки та анімації – сприяють зниженню когнітивного навантаження та покращенню засвоєння знань [1].

2.1 Мультимедійне навчання

Мультимедійне навчання – це використання візуальних і аудіо навчальних матеріалів, зокрема відео, комп'ютерних засобів та інших інформаційних технологій. Теорія мультимедійного навчання зосереджується на принципах ефективного застосування мультимедіа в освітньому процесі,

акцентуючи увагу на одночасному використанні зорового та слухового каналів для обробки інформації.

Слуховий канал відповідає за сприйняття аудіо інформації, тоді як зоровий – за візуальну [15]. При цьому зоровий канал має меншу пропускну здатність порівняно зі слуховим. Якщо інформація подається одночасно через обидва канали, рівень засвоєння знань зростає. Водночас надмірне навантаження на ці канали призводить до неефективної обробки інформації та унеможливорює її закріплення в довготривалій пам'яті.

Метою мультимедійного навчання є створення умов, за яких викладач може одночасно активізувати як візуальні, так і аудіальні канали сприйняття, що сприяє кращому навчанню та засвоєнню матеріалу [14-16].

2.2 Теоретичні підходи навчання

Ефективна педагогічна практика завжди ґрунтується на певній теорії навчання. Водночас, єдиних універсальних стандартів для електронного навчання (e-learning) досі не сформовано. Це зумовлено різноманіттям стилів навчання й викладання, широкими можливостями використання технологій, а також динамічним розвитком самих освітніх технологій. У зв'язку з цим, при проєктуванні програм електронного навчання враховуються різні педагогічні підходи та теоретичні основи. А саме:

– «соціально-конструктивістський підхід» – особливо добре реалізується за допомогою форумів, блогів, вікі та онлайн-інструментів для спільної роботи. Цей підхід базується на співпраці й залучає ширше коло учасників до створення навчального контенту, включаючи самих студентів. Наприклад, фонд One Laptop Per Child впроваджував саме конструктивістську модель навчання;

– «модель діалогу Лорілланд» є особливо актуальною в умовах електронного навчання, як і п'ятиетапна модель Гілли Салмон, яка пропонує поетапне використання дискусійних форумів як педагогічного інструменту;

- «когнітивний підхід» – зосереджений на розумінні когнітивних процесів, що лежать в основі навчання, та механізмів функціонування мозку [10];
- «емоційний підхід» акцентує увагу на таких аспектах навчання, як мотивація, зацікавленість, емоційне залучення та задоволення від навчального процесу;
- «поведінковий підхід» орієнтується на розвиток практичних навичок та поведінкових результатів. При такому підході властиве використання рольових ігор, симуляцій та перенесення знань у реальні (зокрема професійні) ситуації;
- «контекстуальний підхід» підкреслює значення соціального та фізичного оточення як стимулів до навчання, включаючи взаємодію з іншими, колективне відкриття знань та підтримку серед однолітків;
- «нейтральна до формату» модель навчання передбачає поєднання онлайн- і офлайн-форм у межах одного освітнього середовища, сприяючи міжформатній взаємодії та колективному інтелекту.

Багато теоретиків (наприклад, Мейс і де Фрейтас) вважають, що саме взаємодія між студентом і викладачем, а також між самими студентами в онлайн-середовищі найбільше сприяє ефективному навчанню. Теорія Паска також підтверджує цю думку, адже в її основі лежить ідея, що знання формується через діалог, який допомагає зробити його явним. Це особливо актуально в межах віртуального навчального середовища (з англ. «Virtual Learning Environment», VLE).

П'ятиетапна модель електронного навчання Гіллі Салмон значно вплинула на формування онлайн-курсів і форумів. Згідно з цією моделлю перше, що потрібно забезпечити, – це доступ користувача та його здатність користуватись технологією. Далі студент формує онлайн-ідентичність і налагоджує соціальні зв'язки. На третьому етапі відбувається обмін інформацією, пов'язаною з курсом. Наступний рівень – спільна взаємодія студентів у межах навчального контексту. І, нарешті, студенти починають шукати додаткові джерела поза межами системи для поглиблення знань.

Упродовж усіх етапів викладач виконує роль фасилітатора або модератора навчання, підтримуючи процес та спрямовуючи його [11].

Однак модель Салмон піддається певній критиці. По-перше, вона розроблена в контексті дистанційних курсів Відкритого університету й важко переноситься в інші середовища. По-друге, вона не охоплює всю різноманітність підходів до навчання в умовах комп'ютерно-опосередкованої комунікації (з англ. «Computer-Mediated Communication», СМС).

2.2.1 Саморегулювання

Саморегульоване навчання охоплює низку концепцій, що відіграють ключову роль у процесі засвоєння знань і мають особливе значення в контексті електронної освіти. Зиммерман, Б. Дж., у своєму дослідженні «Саморегульоване навчання: від викладання до саморефлексивної практики» стверджує, що для формування навичок саморегуляції навчальні курси мають створювати умови, за яких студенти зможуть самостійно практикувати навчальні стратегії та розвивати відповідні навички.

Саморегуляція навчання тісно пов'язана з соціальним оточенням студента, зокрема впливом батьків і вчителів. Так, Стайнберг (1996) встановив, що учні з високими навчальними досягненнями зазвичай мають батьків із високими очікуваннями, які активно контролюють та підтримують освітню діяльність своїх дітей.

В академічному середовищі саморегульовані студенти, як правило, самостійно визначають навчальні цілі, постійно відстежують власний прогрес і коригують дії для досягнення запланованих результатів. За словами Дейла Х. Шунка студентам необхідно регулювати не лише власні дії, а й пов'язані з досягненнями переконання, наміри та емоційні стани. Крім того, розвиток академічної саморегуляції сприяє формуванню впевненості у власних силах, що є критично важливим для успішного проходження курсів електронного навчання.

2.2.2 Теоретична основа

Наукова література з електронного навчання формує своєрідну екосистему понять, що пов'язана з використанням комп'ютерних технологій у навчальному процесі. Згідно з результатами бібліометричного аналізу, найбільш поширеними концептами є:

- комп'ютерно-асистоване навчання (з англ. «computer-assisted instruction», CAI);
- комп'ютерно-підтримуване навчання (з англ. «computer-assisted learning», CAL);
- освіта на базі комп'ютерних технологій (з англ. «computer-based education», CBE);
- e-learning;
- системи управління навчанням (з англ. «learning management systems», LMS);
- самостійне навчання (з англ. «self-directed learning», SDL);
- масові відкриті онлайн-курси (з англ. «massive open online courses», MOOC).

Усі ці терміни мають спільну рису – інтеграцію елементів навчання та комп'ютерних технологій. Винятком є лише концепт SDL, що походить із психологічної теорії і не обов'язково передбачає застосування цифрових засобів [9]. Зазначені концепції ще потребують подальшого глибокого наукового опрацювання, особливо в контексті порівняння з новими формами освітньої взаємодії, такими як MOOCs. Сьогодні електронне навчання все частіше асоціюється не лише з використанням технологій у класі, а й із масовим поширенням контенту та створенням глобальних освітніх середовищ, доступних усім користувачам Інтернету.

У рамках сучасних досліджень e-learning переважно розглядається крізь призму трьох ключових вимірів: користувачі, технології та освітні послуги.

2.2.3 Застосування теорії навчання до електронного навчання

Як зазначалося на початку цього розділу, дискусія щодо переваг віртуального чи фізичного навчального середовища наразі не має однозначного вирішення. Насамперед, ефективність освітнього середовища значною мірою залежить від характеру навчального матеріалу, який вивчається. Крім того, відмінності у результатах між фізичним і віртуальним навчанням часто пояснюються через призму різних теоретичних підходів вже постфактум.

Однак дослідження показали, що коли фізичне та віртуальне середовища побудовані з урахуванням однакових навчальних принципів – таких як фізичне залучення, когнітивне навантаження, втілене кодування, схемна втіленість та концептуальна виразність – різниця в успішності студентів не залежить від типу середовища (фізичного чи віртуального), а визначається тим, наскільки добре це середовище підтримує відповідну теорію навчання.

Ці висновки свідчать про те, що за умови якісного проєктування віртуальних навчальних середовищ та їх здатності відтворювати ключові елементи фізичного середовища, до них можуть бути застосовані ті самі теоретичні моделі, які раніше використовувалися для традиційних форм навчання.

Більш того, правильно спроектовані віртуальні середовища можуть стати економічно ефективним рішенням з погляду організації освітнього процесу, часу, витраченого на налаштування, і можливості багаторазового використання. Завдяки відносно низьким витратам, студенти можуть виконувати складні аналітичні завдання без необхідності використовувати дорогі лабораторні матеріали.

Багато дослідників також наголошують, що поєднання віртуального та фізичного навчання з урахуванням переваг кожного формату може забезпечити ще вищу ефективність освітнього процесу.

3 СИСТЕМНІ ВИМОГИ

3.1 Функціональні вимоги

Для забезпечення повноцінного функціонування навчального веб-додатку було визначено низку ключових функціональних вимог, які охоплюють основні процеси взаємодії користувачів із системою. Ці вимоги сформовані на основі аналізу потреб цільової аудиторії (учнів і вчителів), а також принципів безпеки, зручності користування та інтеграції компонентів. Нижче наведено основні функціональні блоки, що реалізуються у системі.

Управління користувачами: реєстрація та авторизація. Важливим аспектом стало забезпечення можливості створення окремих облікових записів для учнів і вчителів/адміністраторів з різними функціональними можливостями. Також необхідно було передбачити перевірку введених даних, наприклад, наявність символу «@» в електронній адресі. Безпека збереження паролів була забезпечена шляхом їх хешування з використанням відповідних механізмів, зокрема алгоритму BCrypt.

Управління користувачами: зв'язок між вчителями та учнями. Вчителі повинні мати можливість переглядати та керувати списком учнів, що з ними пов'язані. Учням, у свою чергу, надано право доступу до необхідних вікторин для проходження. Наявність таких зв'язків є критично важливою, оскільки без них учні отримували б доступ до всіх тестів на платформі, незалежно від того, ким вони були створені.

Управління вікторинами: створення та призначення тестів. Система повинна забезпечувати адміністраторам можливість створення вікторин з мінімальною кількістю в п'ять питань та чотирма варіантами відповідей для кожного.

Управління вікторинами: участь у реальному часі. Учні повинні мати можливість проходити вікторини, використовуючи вхід до свого облікового

запису. Після завершення тесту результати мають бути зафіксовані в базі даних PostgreSQL. Кожна відповідь учня фіксується в реальному часі разом із її коректністю [5].

Статистика та зворотний зв'язок: миттєвий аналіз результатів. Після проходження вікторини статистика має автоматично оновлюватися й відображатися. Підраховується кількість правильних відповідей, а результати кожного учня зберігаються в системі.

Статистика та зворотний зв'язок: перегляд результатів. Повинен бути доступний окремий інтерфейс, який дозволяє вчителям/адміністраторам переглядати результати вікторин учнів. Учень також може ознайомитися з загальним результатом у кожному тесті.

Інтеграція та взаємодія компонентів системи. Для ефективного функціонування платформи було необхідно забезпечити безперебійну взаємодію між фронтендом (React) і бекендом (Java Spring Boot) за допомогою RESTful API. Крім того, дані повинні надійно передаватися з інтерфейсу користувача до бази даних через Spring Data JPA у зв'язці з PostgreSQL [2].

Цілісність даних і безпека. Однією з ключових вимог проекту була забезпеченість цілісності даних. Для цього було обрано нормалізовану структуру бази даних, яка охоплює користувацькі записи, вікторини та статистику. Також реалізовано контроль доступу на основі ролей (RBAC), що гарантує, що користувачі з роллю вчителя або учня мають доступ лише до відповідного функціоналу згідно з їхніми правами.

Інтеграція інтелектуального цифрового помічника. У систему інтегровано цифрового помічника на базі платформи Botpress, який здійснює взаємодію з користувачем у форматі діалогу. Помічник може:

- відповідати на запитання користувача під час проходження вікторини;
- пояснювати окремі поняття;
- надавати персоналізовану підтримку відповідно до етапу навчання.

Взаємодія з ботом є інтегрованою в інтерфейс вікторини (рис. 3.1).

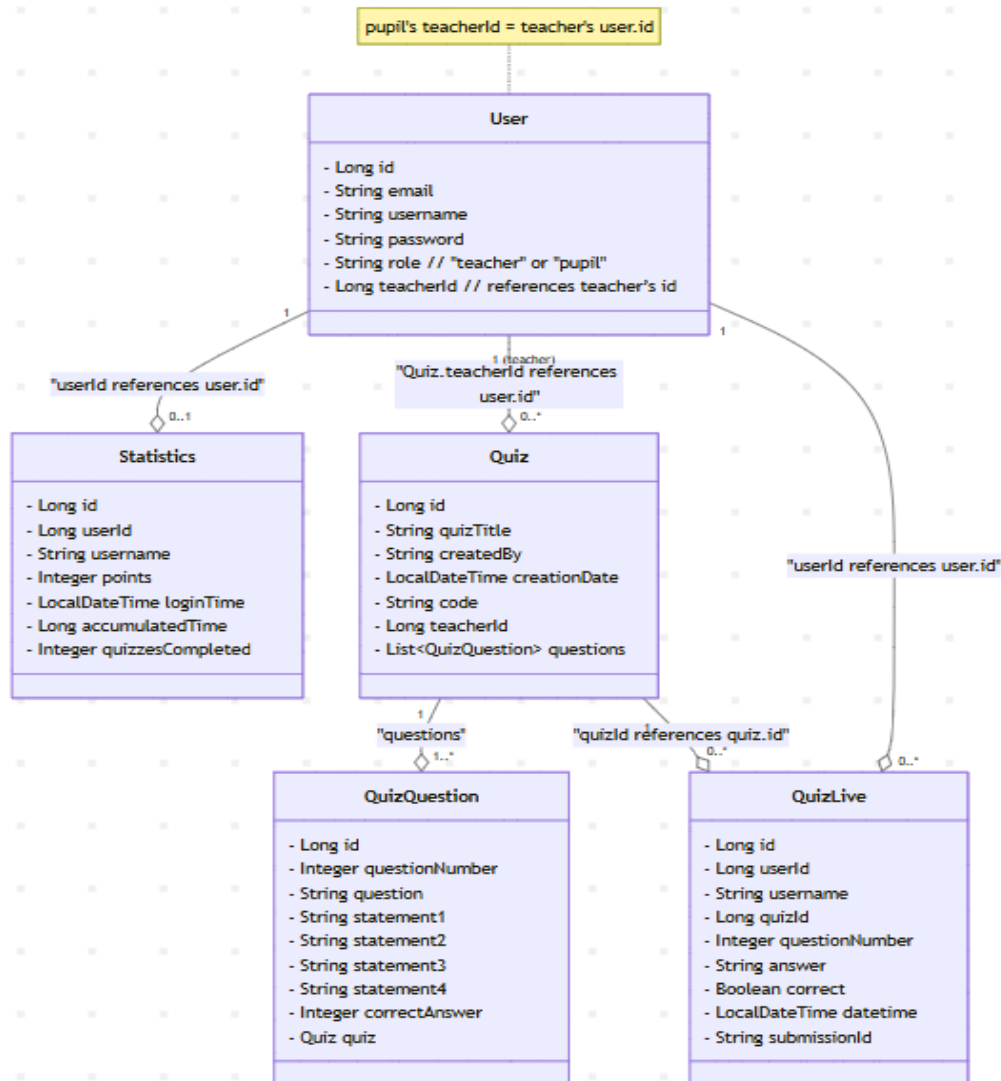


Рисунок 3.1 – Діаграма класів, створена за допомогою редактора Mermaid Live Editor

3.2 Вимоги до даних

3.2.1 Структура бази даних

У процесі розробки навчальної платформи було визначено чіткі вимоги до структури бази даних для забезпечення збереження, обробки та цілісності даних користувачів, вікторин і результатів. Нижче описано основні таблиці, їх призначення та ключові поля.

Таблиця користувачів (users).

Призначення: зберігання профілів учнів та вчителів. Основні поля:

- id – первинний ключ, що однозначно ідентифікує кожного користувача;
- email – унікальна адреса електронної пошти користувача;
- name – відображуване ім'я користувача;
- password – пароль, що зберігається у вигляді хешу з використанням алгоритму (наприклад, BCrypt) для відповідності вимогам безпеки;
- role – вказує роль користувача (учень або вчитель);
- user_score – поточний результат набраних балів користувача.

Таблиці вікторин та питань (quiz_questions).

Призначення: зберігання метаданих про створені вікторини та окремих питань. Основні поля:

- id – зовнішній ключ, що пов'язує питання з відповідною;
- question_text – текст питання;
- answer_option_1 – перший варіант відповіді;
- answer_option_2 – другий варіант відповіді;
- answer_option_3 – третій варіант відповіді;
- answer_option_4 – четвертий варіант відповіді;
- correct_answer – номер правильної відповіді (наприклад, варіант 2 або 3).

Таблиця статистики (scoreboard).

Призначення: зберігання загальних показників успішності учнів і метрик використання системи. Основні поля:

- Statistics ID – унікальний ідентифікатор запису статистики;
- User ID – ID учня, до якого належить запис;
- Quizzes Completed – кількість завершених вікторин;
- Points Earned – загальна кількість набраних балів (наприклад, за правильні відповіді);
- Session Duration – тривалість активних сесій користувача, з урахуванням повторних входів.

3.2.2 Забезпечення цілісності та безпеки даних

Референційна цілісність – зовнішні ключі. Для забезпечення узгодженості даних таблиці містять зовнішні ключі, що посиляються на первинні ключі відповідних таблиць. Це дозволяє автоматично підтримувати зв'язки між таблицями, запобігаючи некоректним оновленням або видаленню даних. Було встановлено обов'язковість (NOT NULL) і унікальність для таких полів, як електронна пошта чи код вікторини, щоб уникнути дублювання записів і забезпечити однозначну ідентифікацію.

Нормалізація. Базу даних спроектовано згідно з принципами нормалізації – дані розподілено між логічно розділеними таблицями («users», «quiz_questions», тощо) з метою усунення надмірності, мінімізації дублювання інформації та збереження структурної цілісності. Це дозволяє оптимізувати структуру бази даних, спрощуючи її супровід і знижуючи ймовірність помилок у даних.

Індекси. Було створено індекси для колонок, до яких найчастіше надсилаються запити, таких як email, quiz_code або user_id. Використання індексів значно покращує продуктивність запитів до бази даних, особливо при виконанні операцій пошуку, сортування чи фільтрації великих обсягів інформації.

Резервне копіювання та відновлення. Регулярне створення резервних копій за допомогою таких інструментів, як pgAdmin 4, або автоматизованих сценаріїв, є необхідною умовою для забезпечення збереження даних у разі втрат або системних збоїв. Відновлення даних з резервної копії дозволяє мінімізувати простоя системи та уникнути втрати критично важливої інформації.

Моніторинг безпеки. Крім основних механізмів захисту, базу даних оснащено налаштуванням ролей і прав доступу. Це дозволяє контролювати рівень доступу до даних для кожного користувача та запобігати несанкціонованим змінам або витоку інформації.

На рисунку 3.2 представлено ER-діаграму, яка демонструє структуру бази даних PostgreSQL та взаємозв'язки між основними таблицями.

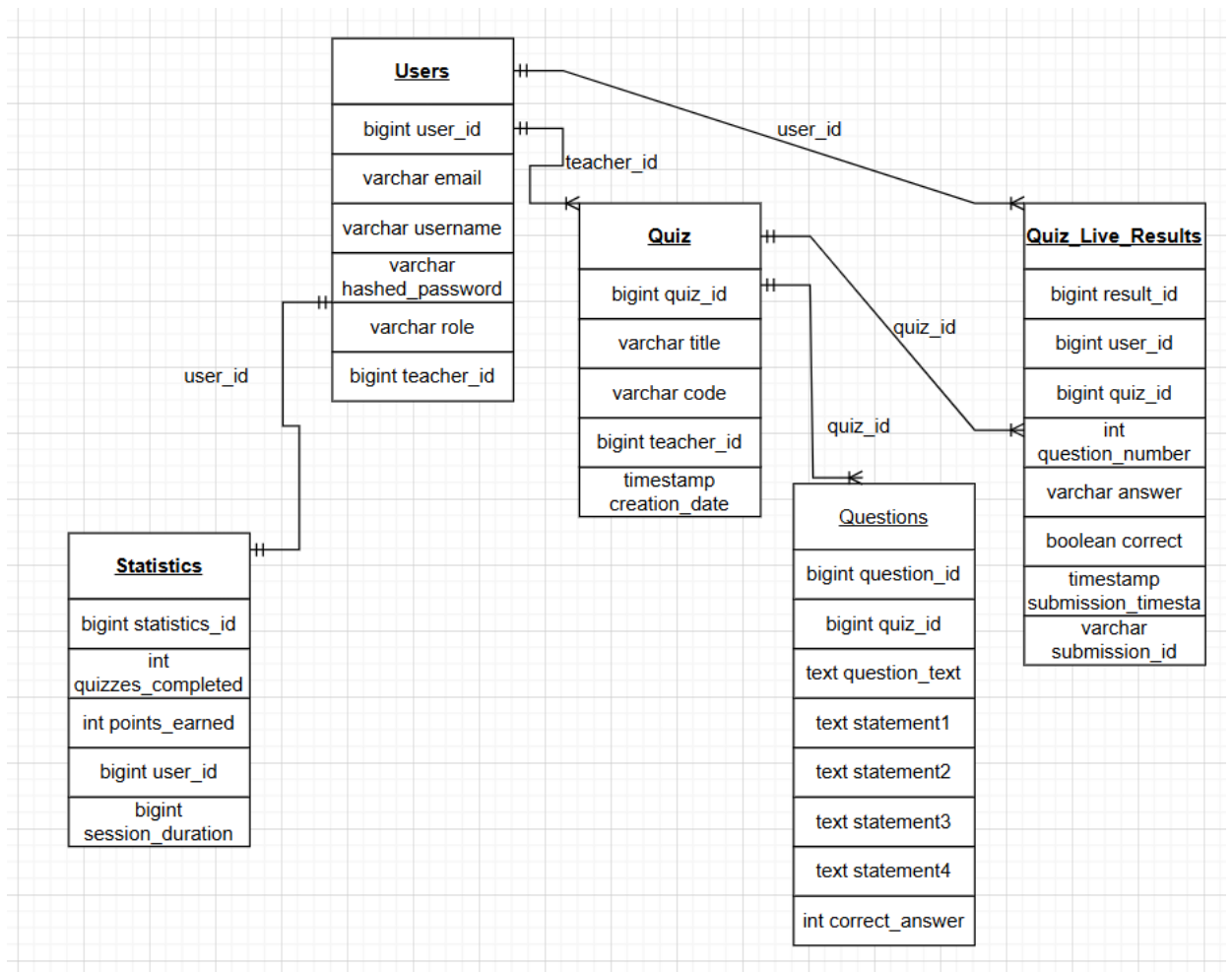


Рисунок 3.2 – ER-діаграма, що демонструє таблиці та зв'язки бази даних PostgreSQL

3.3 Вимоги користувача

Розробка навчальної платформи передбачала створення адаптивного та інтуїтивно зрозумілого інтерфейсу, орієнтованого як на учнів, так і на вчителів. Кожна роль мала власні функціональні та візуальні вимоги, враховані при проектуванні системи.

Для учнівської аудиторії було критично важливо забезпечити простий, але водночас візуально привабливий та інтуїтивний інтерфейс, що сприяє

створенню захопливого навчального досвіду. Зважаючи на молодший вік цільової групи, головна увага приділялася зручності користування.

Основні принципи реалізації інтерфейсу для учнів:

- використання великих кнопок і чітких написів;
- застосування візуальних підказок для кращої навігації;
- миттєве відображення результатів після проходження вікторини;
- наявність елементів гейміфікації:
- можливість обирати та розблоковувати аватари;
- перегляд кількості завершених вікторин;
- підрахунок накопиченого часу входу в систему;
- відображення набраних балів за правильні відповіді.

Крім того, учні повинні мати змогу від'єднатися від поточного вчителя та повторно асоціюватися з новим (наприклад, у разі зміни класу або навчального закладу).

Для вчителів була необхідною наявність повноцінного інструментарію для керування вікторинами та учнями. Простота у використанні також була ключовим критерієм – особливо в аспектах створення, редагування та призначення вікторин учням.

Основні вимоги до інтерфейсу вчителя:

- зручна форма створення вікторин із покроковим заповненням;
- генерація унікального коду для кожної вікторини з метою безпечного доступу;
- можливість перегляду детальних результатів вікторин;
- правильність відповідей окремих учнів;
- середній бал усіх учасників конкретної вікторини;
- керування асоціаціями з учнями:
- від'єднання учнів, які більше не належать до класу;
- додавання нових учнів до свого класу.

Платформа була спроектована з урахуванням принципів адаптивного дизайну. Інтерфейс автоматично підлаштовується під розміри екрану пристрою,

з якого здійснюється доступ (наприклад, планшет або смартфон). Було також реалізовано сенсорні елементи керування для полегшення взаємодії на мобільних пристроях.

З метою забезпечення безпеки користувацьких даних система використовує сучасні механізми автентифікації та розмежування прав доступу:

- для збереження паролів використовується алгоритм хешування BCrypt, що відповідає вимогам безпеки;

- контроль доступу реалізовано за допомогою RBAC (розмежування доступу на основі ролей), що гарантує різний рівень привілеїв для вчителів і учнів;

- вчителі мають доступ до адміністративного функціоналу, який недоступний учням.

Хоча було розглянуто можливість впровадження автентифікації на основі токенів (JWT), на цьому етапі проєкту було вирішено обмежитися вже реалізованими механізмами, які повністю задовольняють вимоги функціональності та безпеки.

3.4 Сценарії використання (Use Cases). UML-діаграми

У цьому розділі представлено варіанти використання функціональних можливостей розробленого навчального вебзастосунку з інтегрованим цифровим помічником. Для візуалізації діаграм варіантів використання було використано програмне забезпечення Microsoft Visio Professional 2021.

Для планування варіантів використання вебзастосунку на етапі автентифікації та реєстрації користувача, було побудовано UML-діаграму, яка ілюструє процес автентифікації користувача (вхід до системи, login), та реєстрації нового користувача (додавання до бази даних, register). Як показано на рисунку 3.3, у випадку введення неправильних даних під час авторизації, система повинна виводити відповідне повідомлення про помилку. Якщо ж дані вірні – користувач успішно ідентифікується, і йому відкривається доступ до

функціоналу вебзастосунку. При реєстрації нового користувача система повинна підтверджувати успішність операції через інформаційне повідомлення про збереження даних.

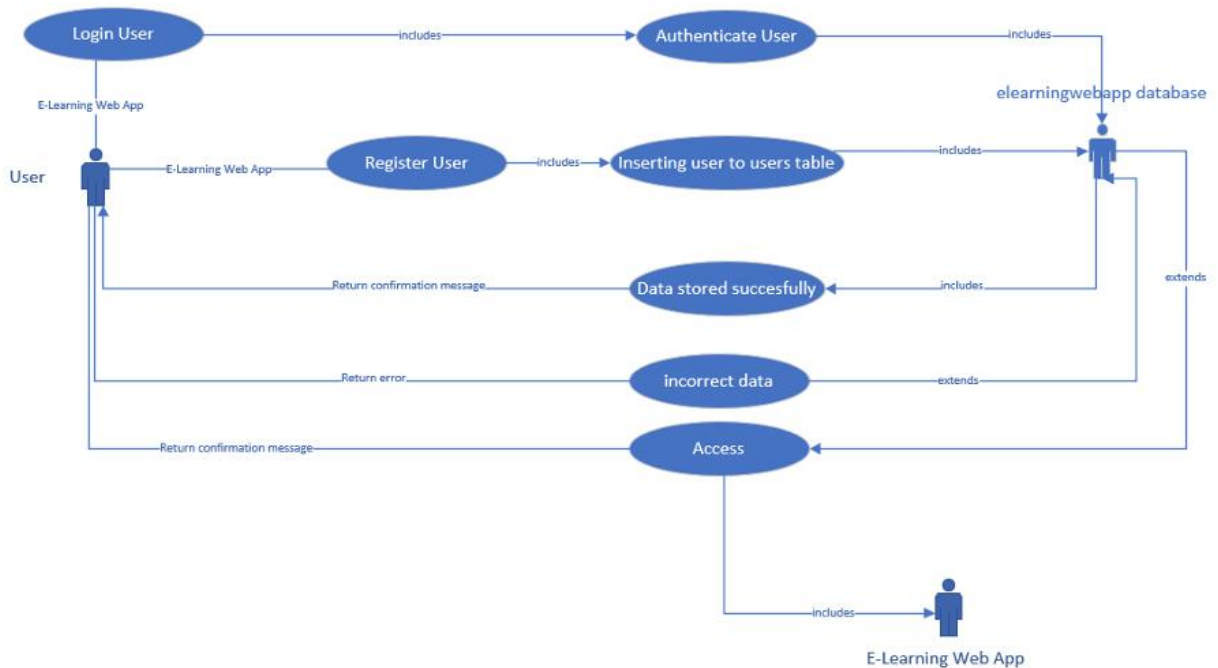


Рисунок 3.3 – UML-діаграма, що описує процес ідентифікації користувача та введення даних про нового користувача у базу даних

Окремо проілюстровано варіант використання (з англ. «use case») для адміністратора (admin), користувача з розширеними правами.

Як показано на рисунку 3.4, після успішної аутентифікації в системі, адміністратор отримує доступ до двох основних функцій:

- керування користувачами – перегляд та видалення облікових записів при необхідності, з підтвердженням дії.

- керування вікторинами – можливість додавати нові запитання, редагувати існуючі або видаляти їх, при цьому система повідомляє про успішність виконання відповідної дії.

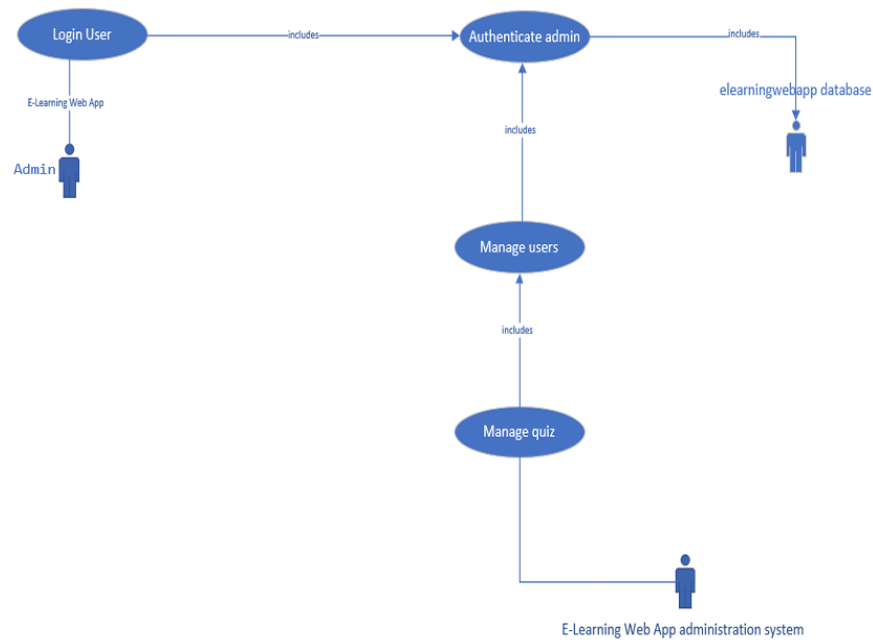


Рисунок 3.4 – UML-діаграма, що описує процес авторизації та функціоналу адміністратора (admin)

В свою чергу звичайний користувач, після входу до системи, має доступ до таких функцій (рисунок 3.5), як:

- перегляд персонального профілю та оновлення особистих даних в реальному часі;
- проходження вікторини, під час якої активується інтегрований цифровий помічник для надання підтримки;
- після завершення вікторини – відображення персонального результату та рейтингу всіх користувачів відповідно до кількості правильних відповідей.

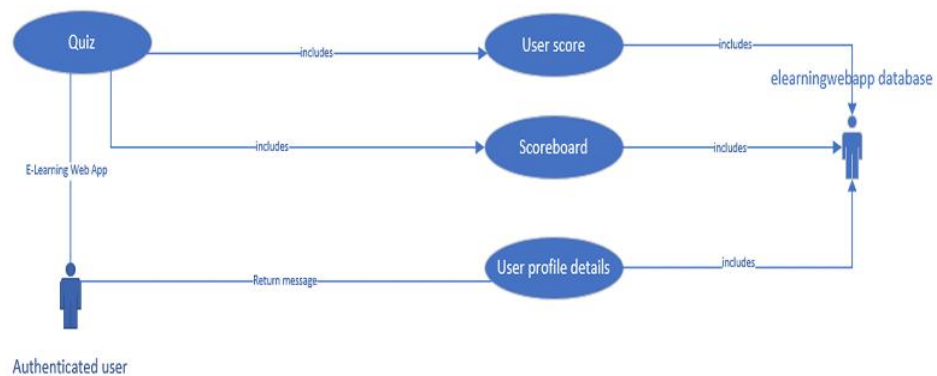


Рисунок 3.5 – UML-діаграма, що описує випадок використання можливостей автентифікованого користувача в додатку

3.5 Вимоги до середовища розробки

Для реалізації навчальної платформи було використано набір сучасних інструментів і середовищ, які забезпечили ефективну організацію процесу розробки, тестування та зберігання проєкту. У цьому розділі подано опис ключових компонентів, що використовувалися на етапах локальної розробки та продакшн-розгортання.

1. Середовище розробки – операційна система та інструменти.

Основна частина розробки здійснювалася в операційній системі Windows із використанням Visual Studio Code як інтегрованого середовища розробки (IDE). Для виконання команд застосовувалися термінал Visual Studio Code та Windows Command Prompt, що забезпечувало зручну взаємодію з інструментами розробки.

2. Керування залежностями.

Для бекенду використовувалася система керування залежностями Maven, яка відповідає за пакування, підключення бібліотек та збірку проєкту на базі Spring Boot [3]. Для фронтенду було застосовано npm (Node Package Manager) у поєднанні з Create React App. Це забезпечило керування залежностями React-додатку, локальну розробку, запуск скриптів та збірку клієнтської частини [6].

3. Локальне керування базою даних.

У процесі розробки система бази даних PostgreSQL була розгорнута локально. Для адміністрування та виконання SQL-запитів застосовувався графічний інструмент pgAdmin 4, що дозволяв ефективно працювати зі структурою таблиць, запитами, перевіркою зв'язків та змінами у схемі бази даних.

4. Середовище продакшн-розгортання.

Готовий проєкт був розміщений у відкритому репозиторії на GitHub. Репозиторій містить:

- повний код клієнтської та серверної частин;

- конфігураційні та скриптові файли для збірки;
- дампи бази даних PostgreSQL, який дозволяє повністю відтворити систему [5].

Хоча розгортання в хмарному середовищі (наприклад, AWS або Heroku) розглядалося, розміщення на GitHub було достатнім для демонстрації повного проєкту з документацією та кодом.

5. Середовище тестування – локальне тестування.

Бекенд проходив юніт- та інтеграційне тестування за допомогою JUnit та Spring Boot Test, що є частиною типових інструментів у Maven.

Фронтенд тестувався за допомогою React Testing Library та Jest, що входять до комплекту Create React App. Це дозволило перевірити коректність відображення компонентів і взаємодію з інтерфейсом.

6. Середовище тестування – перевірка бази даних.

У процесі розробки активно використовувався pgAdmin 4 для тестування бази даних:

- виконання SQL-запитів для перевірки цілісності даних;
- контроль зв'язків між таблицями;
- перевірка актуальності змін у схемі БД;
- верифікація збереження відповідей, результатів та облікових записів.

3.6 Вимоги до зручності використання (Usability requirements)

У процесі розробки платформи велика увага приділялася створенню зручного інтерфейсу, що забезпечує доступність для різних категорій користувачів, узгодженість між сторінками та високу продуктивність. Нижче описано ключові аспекти, пов'язані з дизайном користувацького інтерфейсу, адаптивністю, навігацією та швидкодією системи.

Інтерфейс користувача – інтуїтивний дизайн. Інтерфейс користувача було спроектовано з акцентом на простоту й зрозумілість. Головною метою було забезпечити, щоб як учні, так і вчителі могли безперешкодно орієнтуватися в

платформі. Усі візуальні елементи (кольори, іконки, компоненти) були підібрані з урахуванням принципів UX-дизайну, що сприяють покращенню сприйняття інформації та загального досвіду взаємодії з системою.

Інтерфейс користувача – узгодженість дизайну. Однією з ключових вимог було забезпечення візуальної та функціональної узгодженості між усіма сторінками платформи. Це включає:

- використання однакових елементів оформлення (шапка, меню, кольорова палітра);
- уніфіковані шрифти та розміщення елементів;
- наявність постійного заголовка з логотипом на кожній сторінці;
- стандартний футер з корисними посиланнями.

Адаптивне компонування. Платформа спроектована з урахуванням адаптивного дизайну, що забезпечує коректне відображення на пристроях з різними розмірами екранів – від настільних ПК до планшетів і смартфонів. Компоненти інтерфейсу є зручними для сенсорної взаємодії: кнопки та посилання мають оптимальні розміри, що полегшує їх використання на мобільних пристроях.

Навігація та елементи доступності. Система забезпечує зручну навігацію за рахунок чітко видимих меню, кнопок і посилань, які дозволяють переходити між сторінками вперед і назад без плутанини. Користувачі отримують миттєвий візуальний зворотний зв'язок – наприклад, підсвічування кнопок при наведенні та повідомлення про помилки. Крім того:

- використовується висококонтрастний текст для покращення читабельності;
- структура сайту побудована на основі семантичної HTML-розмітки, що забезпечує сумісність із засобами допоміжного доступу (екранні читалки тощо).

Продуктивність системи. Час завантаження. Особливу увагу було приділено оптимізації фронтенд-ресурсів (зображення, JavaScript, CSS), щоб забезпечити швидке завантаження сторінок. Архітектура додатку, включно з

бекенд-ендпойнтами, була реалізована таким чином, щоб ефективно обробляти запити та мінімізувати час відповіді сервера.

Миттєвий зворотний зв'язок у реальному часі. Система забезпечує негайне надання результатів після завершення вікторини користувачем. Це дозволяє користувачам одразу побачити власні успіхи та рівень засвоєння матеріалу. Для адміністраторів дані оновлюються в режимі реального часу, що дозволяє оперативно відстежувати прогрес користувачів, виявляти закономірності, а також швидко реагувати на навчальні труднощі.

4 ОПИС АРХІТЕКТУРИ СИСТЕМИ

У розділі 4 буде наведено визначення мов програмування з описом використовуваних функцій, а також повний аналіз архітектури системи веб-застосунку. Архітектура системи веб-застосунку з вбудованим цифровим помічником базується на сучасному підході до розробки, тому для забезпечення інтегрованої та ефективної освітньої платформи були обрані передові технології. Застосунок було реалізовано як на фронтенд-, так і на бекенд-рівнях. У фронтенд-частині використано бібліотеку React для створення інтерактивного та зручного інтерфейсу, який забезпечує динамічну адаптацію та оптимізований користувацький досвід. Для бекенду була застосована популярна мова програмування Java разом із відомим Java Web Framework Spring Boot, що забезпечує безпечне з'єднання користувачів завдяки шифруванню, стабільність системи та легкість її масштабування для вирішення майбутніх завдань [3]. Дані зберігаються в базі даних PostgreSQL, що гарантує надійне та ефективне управління інформацією [5].

Загальна архітектура системи побудована відповідно до багаторівневого підходу, що забезпечує легке обслуговування та гнучкість для подальших оновлень системи. У подальшому буде представлено детальний розгляд концепцій повнофункціонального застосунку, API, а також особливостей розробки фронтенд- та бекенд-частин.

Повноцінна веб-розробка передбачає одночасну роботу та взаємодію між двома ключовими складовими веб-застосунку – фронтендом і бекендом. Цей підхід добре знайомий фахівцям у сфері веб-розробки.

Розробники, як правило, володіють базовими знаннями програмування, необхідними для створення інтерфейсу, з яким безпосередньо взаємодіє користувач, а також для відображення та обробки даних, приділяючи особливу увагу зовнішньому вигляду вебсторінки (Front-End). Крім того, вони здатні забезпечити управління даними та їх збереження на сервері, використовуючи

відповідні засоби безпеки для усунення потенційних вразливостей застосунку.

Фронтенд (Front-End) – це та частина системи, де користувачі бачать застосунок і безпосередньо взаємодіють із ним.

З іншого боку, бекенд-частина веб-застосунку є невидимою для кінцевого користувача, оскільки саме в ній реалізована основна логіка програми, взаємодія з базою даних, робота сервера та інші ключові функції, необхідні для безперебійного функціонування системи. Основними складовими повноцінного веб-застосунку є фронтенд, бекенд та база даних.

У фронтенд-частині розробки особлива увага приділяється як користувацькому досвіду (UX), так і інтерфейсу користувача (UI). Попри схожість термінів, це різні концепції: UX охоплює загальне сприйняття роботи з продуктом, тоді як UI відповідає за візуальне оформлення та інтерфейсні елементи. Функції веб-застосунку, що включають візуальні ефекти, анімацію, зображення чи відео на сайті, можуть демонструвати хороший дизайн інтерфейсу, але при цьому забезпечувати незручний користувацький досвід – і навпаки. Інтуїтивно зрозумілий інтерфейс, який не вимагає від користувача спеціальних знань, є ознакою якісно реалізованого UX.

З позиції розробника, фронтенд – це саме та частина, яку бачить користувач під час взаємодії з сайтом. Вона охоплює дизайн, верстку, структуру сторінки та забезпечується за допомогою відповідних технологій програмування.

Бекенд, на відміну від фронтенду, прихований від користувача, але є критично важливим для внутрішнього функціонування системи. Він включає сервер, застосунок та механізми обробки запитів. Сервер обробляє всі запити, що надходять від клієнта (користувача), а логіка застосунку активує необхідні функції відповідно до запитів. Таким чином, бекенд постійно синхронізується з клієнтською частиною через виклики функцій (наприклад, `async-await`, `try-catch`, `sync`), які забезпечують обробку даних.

Серверна архітектура базується на моделі запит-відповідь, що дає змогу ефективно здійснювати комунікацію між клієнтською частиною та сервером і забезпечує надійну роботу веб-застосунку. Сам застосунок, у свою чергу, виступає посередником між сервером і базою даних, передаючи всю необхідну інформацію для отримання чи збереження даних. У бекенд-частині зазвичай зосереджений увесь логічний і функціональний код, який реалізується на сервері відповідно до вимог проєкту.

Сучасні веб-додатки активно використовують інтерфейс прикладного програмування (з англ. «Application Programming Interface», API) – набір протоколів, що дозволяє різним програмам взаємодіяти між собою. API визначає, які методи та типи даних можуть використовуватися для передачі інформації між модулями, що дає розробникам змогу створювати масштабовані системи зі зручною інтеграцією та високою взаємодією між компонентами [8].

Основні концепції API включають:

- кінцеві точки (з англ. «Endpoints») – це конкретні URL-адреси, через які клієнт може звертатися до певних сервісів або ресурсів API;
- запити (з англ. «Requests») – API дозволяє одній програмі надсилати запити до іншої. Найчастіше використовуються такі HTTP-методи: GET – отримання даних із сервера; POST – надсилання нових даних на сервер; PUT – оновлення вже існуючих даних на сервері; DELETE – видалення даних із сервера;
- відповіді (з англ. «Responses»): після обробки запиту, API повертає відповідь, зазвичай у форматі JSON або XML. Response зазвичай містить або запитувані дані, або повідомлення про успішність чи невдачу виконання запиту [4];
- аутентифікація (з англ. «Authentication»): API часто використовують спеціальні токени доступу для перевірки особи користувача або програми, забезпечуючи безпечну взаємодію та захист доступу до ресурсів;

– обмеження швидкості (з англ. «Rate Limiting»): встановлення обмежень на кількість запитів за певний період (наприклад, не більше 100 запитів на хвилину), що запобігає перевантаженню сервісу та підтримує його стабільну роботу.

Існує кілька типів API, найпоширенішим з яких є веб-API, що дозволяють взаємодію між системами через Інтернет – переважно з використанням протоколів передачі гіпертексту (з англ. «Hypertext Transfer Protocol», HTTP або «Hypertext Transfer Protocol Secure», HTTPS). Прикладами є API соціальних мереж (наприклад Twitter, Facebook), які дають змогу стороннім застосункам інтегрувати функціональність цих платформ.

Іншим видом є бібліотечні API або фреймворки, які дозволяють розробникам інтегрувати готові функціональні модулі у свій код, наприклад, бібліотеки JavaScript.

Також виділяють API операційної системи, які забезпечують доступ до системних функцій, зокрема файлових систем, мережевих інтерфейсів або апаратного забезпечення.

API відіграють ключову роль у сучасній розробці програмного забезпечення. Вони дозволяють розробникам легко підключати зовнішні сервіси до власних продуктів без потреби створювати їх із нуля. Це значно зменшує складність, економить час і ресурси, а також забезпечує масштабованість, гнучкість та автоматизацію процесів. Таким чином, API виконують роль каналу зв'язку між системами, відкриваючи широкі можливості для інтеграції, обміну даними та побудови комплексних цифрових рішень [2].

4.1 Технології реалізації серверної частини (Back-End) та бази даних

Для розробки серверної частини навчальної платформи було використано фреймворк Spring Boot, який є одним із найпотужніших та найпопулярніших рішень на основі екосистеми Spring. Його використання дозволяє створювати

вебзастосунки мовою програмування Java з мінімальними налаштуваннями, що робить його ідеальним вибором для швидкої розробки сучасних вебрішень. Однією з переваг Spring Boot є спрощене підключення до баз даних через Java DataBase Connectivity (JDBC) API, що забезпечує швидке та ефективне управління даними. Автоматичне конфігурування, підтримка ORM-інструментів, таких як Hibernate, а також можливість генерації необхідного коду відповідно до специфікацій застосунку дозволяють розробникам зосередитися на реалізації бізнес-логіки, мінімізуючи потребу в ручному налаштуванні. Фреймворк має широку підтримку спільноти, велику кількість вбудованих інструментів, таких як Spring Security, Spring Data тощо. Крім того, Spring Boot чудово інтегрується з сучасною мікросервісною архітектурою, що робить його ідеальним для створення масштабованих вебзастосунків [3].

4.1.1 Аутентифікація за допомогою JWT

Для захисту доступу користувачів у навчальному застосунку було реалізовано механізм аутентифікації на основі JWT (JSON Web Token). Цей підхід дозволяє безпечно здійснювати авторизацію користувача та розмежовувати доступ до функціоналу за ролями: адміністратор (admin) та звичайний користувач [4].

Процес автентифікації за допомогою JWT реалізується в кілька етапів:

- аутентифікація – користувач вводить свою електронну пошту та пароль на сторінці входу. Сервер перевіряє облікові дані в базі даних;
- генерація токена – якщо користувача знайдено, створюється JWT-токен, що містить зашифровану інформацію (claims), наприклад, ID користувача та його роль. Токен підписується секретним ключем;
- надсилання токена – сформований токен передається клієнтській частині (наприклад, вебзастосунку) та зберігається у local storage або у вигляді cookie;

- надсилання запитів – під час виконання захищених дій клієнт передає JWT у запиті, як правило, в заголовку Authorization з префіксом Bearer;

- перевірка токена – сервер перевіряє цифровий підпис токена. Якщо токен є чинним, запит авторизується;

- відповідь сервера – за умови успішної верифікації токена користувач отримує доступ до захищених ресурсів. У разі недійсного токена доступ блокується.

Структура JWT-токену складається з трьох частин:

- Header – містить тип токена та алгоритм підпису;
- Payload – включає основні дані про користувача (claims);
- Signature – створюється за допомогою алгоритму шифрування (у цьому проєкті – HmacSHA256) та гарантує цілісність токена.

Особливістю JWT є те, що серверу не потрібно зберігати сесійну інформацію – усі необхідні дані містяться безпосередньо в токені. Це робить метод ефективним, масштабованим і придатним для роботи з великою кількістю одночасних користувачів. Серед основних переваг JWT-автентифікації – висока безпека, захист персональних даних, зменшення навантаження на сервер і підтримка реального часу.

4.1.2 Система зберігання даних – PostgreSQL

Для зберігання даних у навчальному застосунку було використано систему керування базами даних PostgreSQL. Це одна з найпопулярніших СУБД з відкритим вихідним кодом для реляційного моделювання даних у сучасних вебзастосунках. У контексті даної платформи PostgreSQL використовується для збереження:

- облікових записів користувачів (логін, email, пароль, роль);
- персональних балів за результати вікторин;
- даних про вікторини та питання (таблиці users, quiz_questions тощо).

Серед переваг PostgreSQL:

- висока надійність і стабільність (понад 30 років розвитку);
- розширюваність і підтримка сучасних типів даних, зокрема JSON;
- повна сумісність зі стандартом SQL;
- наявність драйверів для більшості мов програмування, включно з Java;
- підтримка розширень (наприклад, PostGIS для геоданих);
- вбудовані механізми безпеки й резервного копіювання.

Усе це робить PostgreSQL універсальним рішенням для створення надійної та масштабованої системи зберігання даних у межах освітнього застосунку [5].

4.2 Технології реалізації клієнтської частини (Front-End)

Для реалізації клієнтської частини навчального вебзастосунку було використано одну з найпопулярніших JavaScript-бібліотек – React, розроблену компанією Facebook. React широко застосовується для створення інтерактивних, швидких та ефективних користувацьких інтерфейсів, особливо у форматі SPA (Single Page Applications) – односторінкових застосунків.

React базується на компонентному підході, що дозволяє розділити інтерфейс на малі, повторно використовувані та незалежні компоненти. Кожен компонент може керувати власним станом (state) та взаємодіяти з іншими, згідно з логікою, визначеною розробником. Такий підхід значно покращує структурування коду, спрощує його підтримку та розширення.

Однією з ключових технологічних переваг React є використання віртуальної об'єктної моделі документа (з англ. «Virtual Document Object Model», Virtual DOM). Це дозволяє оновлювати лише ті елементи інтерфейсу, які зазнали змін, а не всю сторінку, що забезпечує високу продуктивність і швидкість відображення особливо в динамічних середовищах [6]. React також підтримує односторонній потік даних, де дані передаються від батьківських компонентів до дочірніх. Це забезпечує контрольовану обробку стану та

покращує розуміння змін у інтерфейсі. Ще однією важливою особливістю є використання JSX – синтаксису, який поєднує JavaScript і HTML. Завдяки JSX розробка інтерфейсів стає більш зрозумілою, читабельною та ефективною.

React має широку екосистему готових рішень і бібліотек, серед яких React Router, що відповідає за маршрутизацію сторінок у межах застосунку. Завдяки цьому React є оптимальним вибором для розробки сучасних вебзастосунків із великою кількістю взаємодій та обсягом даних [7].

Взаємодія з API – бібліотека Axios.

Для обміну даними між клієнтською і серверною частинами застосунку використовувалась популярна бібліотека Axios, яка спрощує створення HTTP-запитів. Її ключові функції включають:

- GET/POST-запити – дозволяють завантажувати або надсилати дані до API [2];
- автоматичне перетворення JSON – дані, що передаються та отримуються від сервера, автоматично конвертуються у формат JSON;
- підтримка перехоплювачів (interceptors) – можливість обробки запитів/відповідей до їхнього відображення у UI;
- обробка помилок – зручні функції для виявлення та керування помилками при виконанні запитів.

Завдяки простоті налаштування та адаптивності, Axios є ефективним інструментом для комунікації між Front-End і Back-End частинами проєкту.

4.3 Використані інструменти розробки та середовище реалізації

Для реалізації вебзастосунку з інтегрованим цифровим помічником було використано сучасне програмне забезпечення та платформи, які забезпечили повноцінну підтримку розробки Front-End, Back-End, управління базою даних та тестування API, що у комплексі гарантує цілісність та стабільність обміну даними між клієнтською та серверною частинами [8].

1. Інструменти та середовище реалізації клієнтської частини.

Клієнтську частину було реалізовано в середовищі Microsoft Visual Studio Code (рис. 4.1) – одному з найпопулярніших редакторів коду для веброзробки. На етапі налаштування:

- було встановлено Node.js;
- за допомогою терміналу VS Code створено проєкт Vite командою `npm create vite@latest`;
- проєкту присвоєно назву: `eLearningWebApp-Frontend`;
- обрано фреймворк: React;
- варіант: JavaScript.

Оскільки, Visual Studio Code підтримує основні технології фронтенду (HTML, CSS, JavaScript), має велику кількість розширень для React, та містить засоби для відлагодження, автозаповнення коду й перевірки синтаксису – це сприяло організованому й швидкому створенню користувацького інтерфейсу [6].

```

1 import React from 'react';
2 import { BrowserRouter, Routes, Route, Navigate } from 'react-router-dom';
3 import NavBar from './src/component/common/NavBar';
4 import FooterComponent from './src/component/common/footer';
5 import QuizHomePage from './src/component/home/QuizHomePage';
6 import LoginPage from './src/component/auth/LoginPage';
7 import RegisterPage from './src/component/auth/RegisterPage';
8 import UserProfilePage from './src/component/profile/UserProfilePage';
9 import EditUserProfilePage from './src/component/profile/EditUserProfilePage';
10 import AdminPage from './component/admin/AdminPage';
11 import { ProtectedRoute, AdminRoute } from './service/AuthRoute';
12 import Questions from './src/component/quiz/Questions';
13 import ManageUsersPage from './src/component/admin/ManageUsersPage';
14 import ManageQuiz from './src/component/admin/ManageQuiz';
15 import Scoreboard from './src/component/quiz/Scoreboard';
16
17 function App() {
18   return (
19     <BrowserRouter>
20       <div className="App">
21         <NavBar />
22         <div className="content">
23           <Routes>
24             <Route exact path="/home" element={<QuizHomePage />} />
25             <Route exact path="/login" element={<LoginPage />} />
26             <Route path="/register" element={<RegisterPage />} />
27
28             <!-- Questions page is protected and has its own layout -->
29             <Route path="/questions" element={<ProtectedRoute element={<Questions />} />} />
30             <Route path="/scoreboard" element={<ProtectedRoute element={<Scoreboard />} />} />
31
32             <Route path="/profile" element={<ProtectedRoute element={<UserProfilePage />} />} />
33             <Route path="/edit-profile" element={<ProtectedRoute element={<EditUserProfilePage />} />} />
34             <Route path="/admin" element={<AdminRoute element={<AdminPage />} />} />
35             <Route path="/manage-users" element={<AdminRoute element={<ManageUsersPage />} />} />
36             <Route path="/manage-quiz" element={<AdminRoute element={<ManageQuiz />} />} />
37
38             <Route path="*" element={<Navigate to="/home" />} />
39           </Routes>
40         </div>
41       </div>
42     </BrowserRouter>
43   );
44 }

```

Рисунок 4.1 – Середовище розробки клієнтської частини системи –
Microsoft Visual Studio Code

2. Інструменти та середовище реалізації серверної частини.

Серверна частина застосунку розроблялася в середовищі IntelliJ IDEA Ultimate, яке забезпечує комплексну підтримку мови Java. Завдяки цьому IDE реалізовано: зручне автозаповнення коду, гнучке налагодження на будь-якому етапі виконання та інтеграцію з фреймворками, зокрема Spring Boot, для створення масштабованих і стабільних вебсервісів. Середовище також дозволяє ефективно організувати структуру проєкту, реалізовувати обмін даними між користувачем та сервером, і керувати потоками даних через API (рис. 4.2).

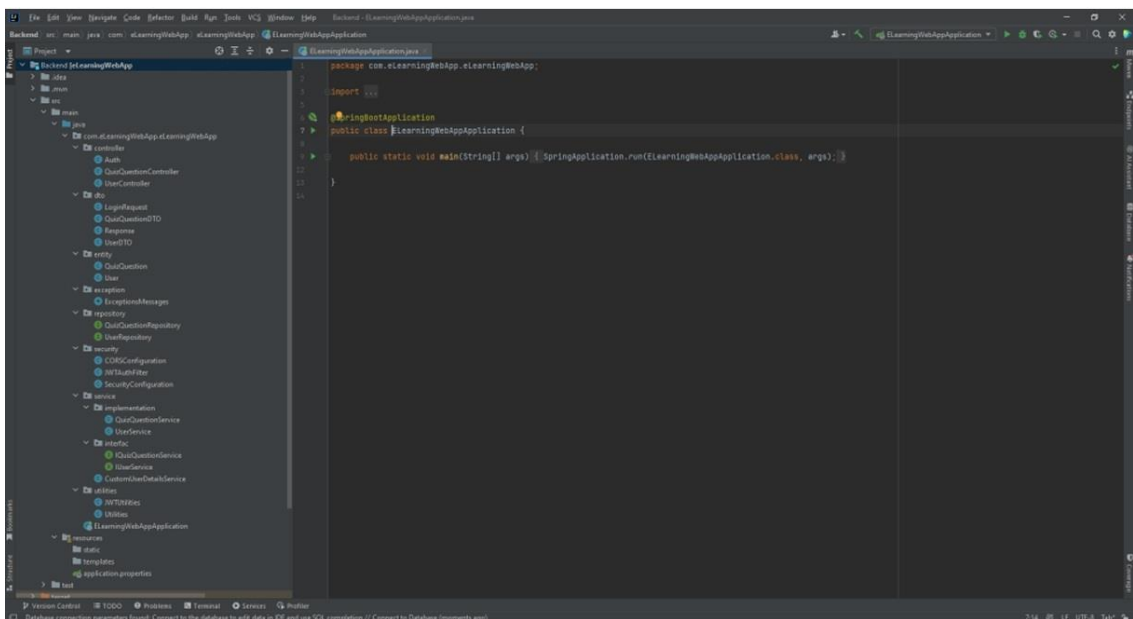


Рисунок 4.2 – Середовище розробки серверної частини системи – IntelliJ IDEA Ultimate

3. Управління та інтерфейс базою даних PostgreSQL.

Управління реляційною базою даних PostgreSQL здійснювалося через phpPgAdmin 4 – графічний інтерфейс, що дозволяє:

- запускати SQL-запити;
- керувати таблицями та структурами даних;
- перевіряти вміст і цілісність інформації у зручному режимі.

Завдяки простому та зрозумілому інтерфейсу, інструмент значно полегшує процес супроводу та адміністрування бази даних (рис. 4.3).

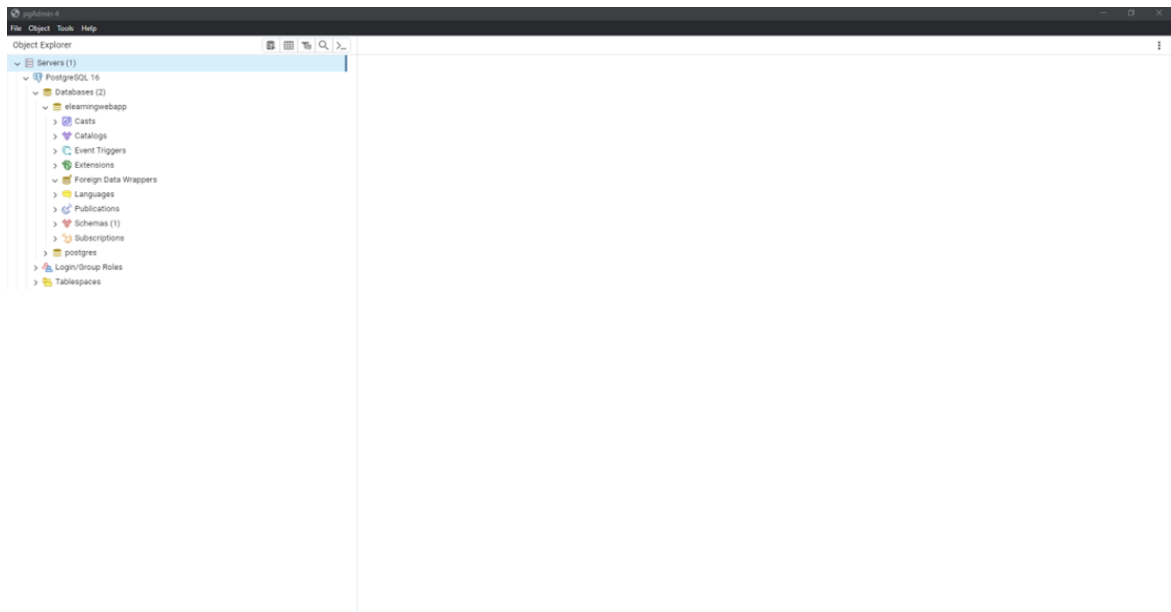


Рисунок 4.3 – Середовище управління базою даних PostgreSQL – phpPgAdmin4

4. Тестування API: інструмент Postman.

Для перевірки коректності роботи API та серверних відповідей використовувався Postman – сучасний інструмент для створення та надсилання HTTP-запитів. Postman дає змогу у реальному часі перевірити правильність роботи запитів, достовірність відповідей API, а також виявити потенційні проблеми у логіці обробки даних на бекенді [8]. Основні функції, що були реалізовані за допомогою Postman:

- тестування REST-ендпойнтів на правильність відповідей;
- перевірка статусів запитів і повідомлень про помилки;
- візуалізація даних, що повертаються з сервера;
- швидке проведення тестів без запуску повного інтерфейсу.

5 ОГЛЯД РЕАЛІЗАЦІЇ E-LERNING ПЛАТФОРМИ

У цьому розділі детально описано функціонування розробленого вебзастосунку, з акцентом як на реалізацію коду, так і на користувацький інтерфейс. Буде розглянуто інтеграцію цифрового помічника (чат-бота) в архітектуру застосунку, роль елементів штучного інтелекту в його роботі. А також буде розглянута побудована ER-діаграма бази даних, що відображає структуру таблиць, створених для потреб навчального середовища.

5.1 Функціонування системи з погляду користувача та адміністратора

Розроблений вебзастосунок для електронного навчання забезпечує інноваційний підхід до вивчення інформаційних технологій за допомогою інтерактивної вікторини, персоналізованої статистики та інтегрованого цифрового помічника на базі штучного інтелекту. Такий формат сприяє інтерактивному навчанню, підвищує мотивацію користувачів та дозволяє відстежувати індивідуальний прогрес.

Користувач має змогу:

- видання не має додаткових методів привернення та утримання уваги;
- проходити тематичні вікторини;
- переглядати власний результат у вигляді кількості набраних балів;
- бачити свою позицію в загальному рейтингу серед інших учасників.

У процесі взаємодії з вікториною користувач може скористатися цифровим помічником, що працює на основі штучного інтелекту. Помічник надає персоналізовані підказки та відповіді на запитання, спрямовані на глибше розуміння теми.

5.1.1 Опис інтерфейсу користувача: робота з вікториною

Головною складовою платформи є вікторина, що складається з запитань з обраної теми. Для наочності роботи системи та її тестування, були створені питання з теми інформатики. Після запуску проекту (спочатку з бекенду, потім з фронтенду за допомогою команди `npm run dev`) користувач потрапляє на головну сторінку застосунку (рис. 5.1).

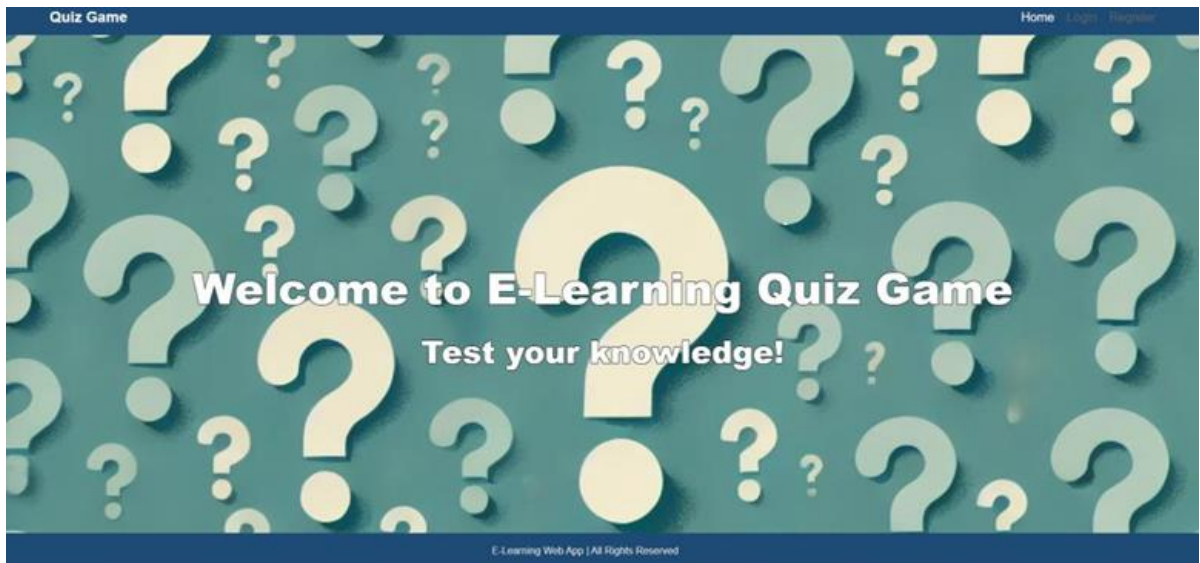


Рисунок 5.1 – Головна сторінка веб-додатку для електронного навчання

З головної сторінки доступні основні дії через навігаційну панель (з англ. «Navigation Bar», `navbar`):

- «Home» – повернення на головну сторінку;
- «Login» – авторизація користувача – для ідентифікації користувача та автоматичного входу в застосунок;
- «Register» – реєстрація нового користувача – для введення даних нового користувача в застосунок.

Якщо користувач не має зареєстрованих даних у додатку та бажає пройти реєстрацію, натискання кнопки «Register» перенаправляє його на відповідну сторінку для заповнення персональних даних (рис. 5.2).

The screenshot shows a web application interface for a 'Quiz Game'. At the top, there are navigation links for 'Home', 'Login', and 'Register'. The main content area features a 'Sign Up' form with the following fields: 'Name' (containing 'akimov'), 'Email' (containing 'b.akimov@gmail.com'), and 'Password' (masked with dots). Below the form is a blue 'Register' button and a link that says 'Already have an account? Login'. The footer of the page reads 'E-Learning Web App | All Rights Reserved'.

Рисунок 5.2 – Сторінка реєстрації – Введення даних нового користувача

При переході на сторінку реєстрації користувач заповнює поля Name, Email, Password, після чого дані зберігаються до бази даних elearningwebapp, де створюється новий запис у таблиці «users» із правами стандартного доступу (рис. 5.3-5.4).

id	email	name	password	role	user_score
[PK] bigint	character varying (255)	character varying (255)	character varying (255)	character varying (255)	integer

Рисунок 5.3 – Таблиця «users» до реєстрації користувача в системі

id	email	name	password	role	user_score
1	b.akimov@gmail.com	akimov	\$2s\$10\$Ag6UF3emosOmC448OK/rJO2xp25FoJk7KLMH8WqDNAVb2TRY13P...	USER	0

Рисунок 5.4 – Таблиця «users» після реєстрації користувача в системі

Як видно з таблиці «users» (рис. 5.4), новий користувач успішно доданий до системи: усі його дані збережено, а пароль – зашифровано з використанням алгоритму хешування. Водночас, користувачу автоматично призначається роль звичайного користувача (user), а його початковий рейтинг (score) дорівнює 0, оскільки він ще не розпочинав проходження вікторини.

Після успішної реєстрації, щоб увійти до системи, користувач натискає кнопку «Login» і переходить на сторінку авторизації, де вводить свої Email та Password для ідентифікації (рис. 5.5).

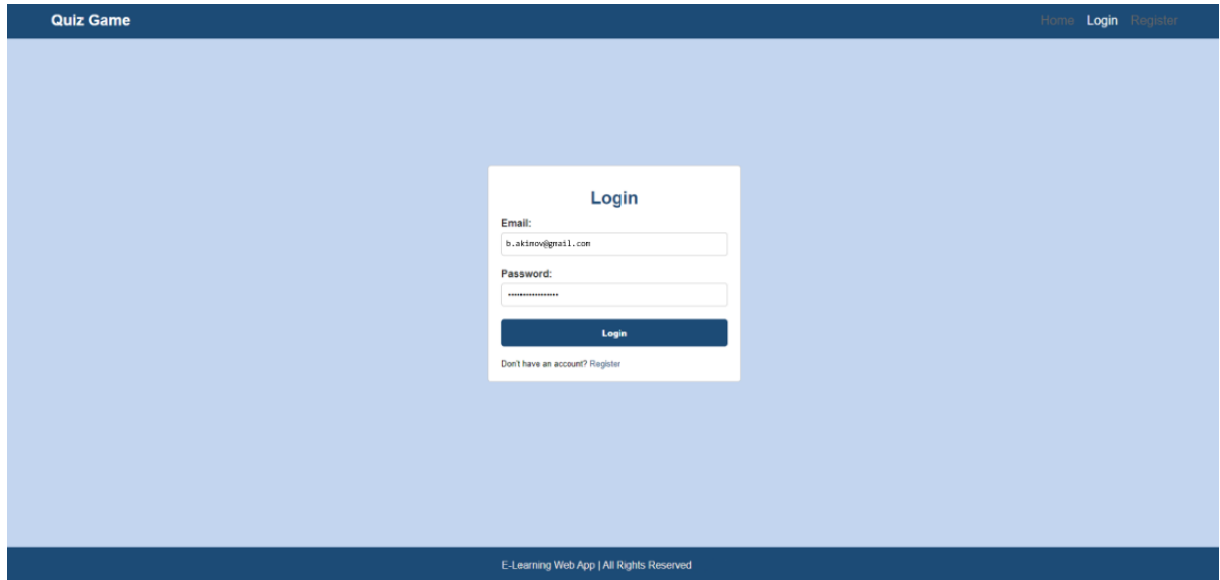


Рисунок 5.5 – Сторінка «Login» – Ідентифікація користувача

У разі успішної ідентифікації система виконує вхід користувача до застосунку та динамічно оновлює навігаційну панель (navbar), додаючи нові функціональні елементи (рис. 5.6). Серед них:

- кнопка «Profile» для перегляду особистих даних користувача (рис. 5.7) з можливістю їх негайного редагування та збереження змін за допомогою кнопок «Edit Profile» та «Update Profile» (рис. 5.8);
- кнопка «Game», яка веде до початку проходження вікторини;
- кнопка «Scoreboard» для перегляду підсумкових балів усіх користувачів;
- кнопка «Logout», що дозволяє завершити сесію та вийти з облікового запису.

Питання, що входять до складу вікторини, зберігаються в таблиці «quiz_questions» бази даних elearningwebapp. Кожне питання має відповідний текст, варіанти відповідей та правильну відповідь, яка використовується для автоматичної перевірки результатів.



Рисунок 5.6 – Оновлена сторінка «Home» (успішний вхід користувача)

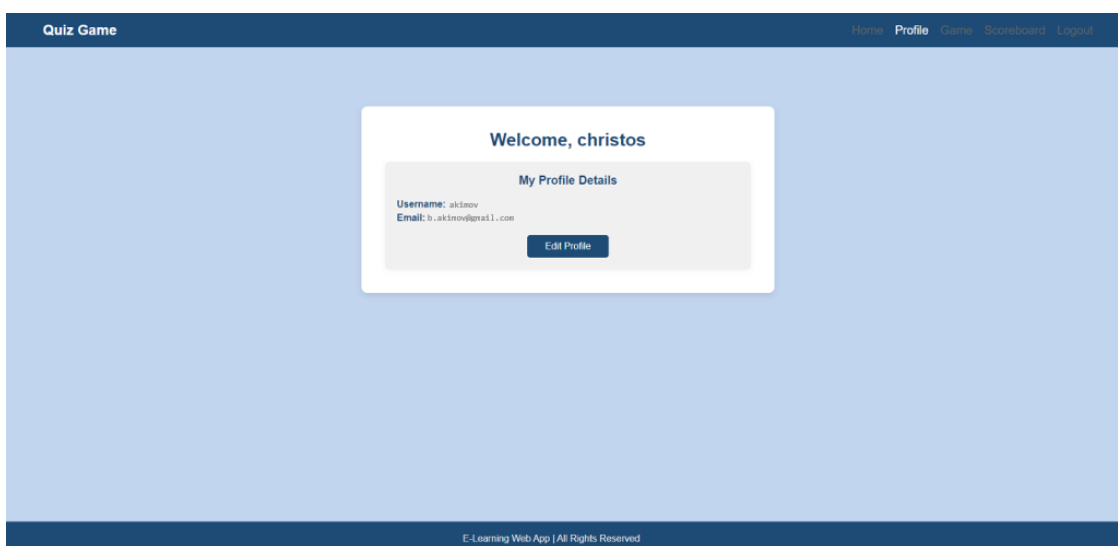


Рисунок 5.7 – Сторінка «Profile» – Перегляд профілю користувача при вході

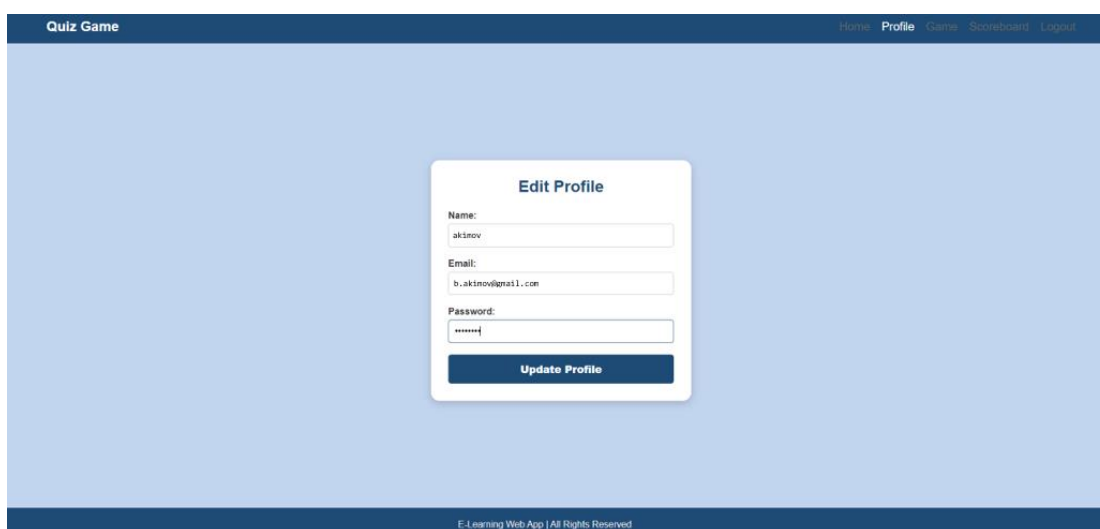


Рисунок 5.8 – Сторінка «Profile» – Оновлення особистої інформації користувача

Вікторина включає наступні запитання та відповіді.

What does CPU stand for? Answer: Central Processing Unit.

Which language is primarily used for Android app development? Answer: Kotlin.

What does RAM stand for? Answer: Random Access Memory.

Which of the following is a programming language? Answer: JavaScript.

What is the time complexity of binary search? Answer: $O(\log n)$.

Which of the following is an operating system? Answer: Windows.

What does HTTP stand for? Answer: HyperText Transfer Protocol.

Which company developed the C programming language? Answer: Bell Labs.

What is the primary purpose of DNS? Answer: Translate domain names to IP addresses.

Which of the following is a NoSQL database? Answer: MongoDB.

На рисунку 5.9 представлено структуру таблиці «quiz_questions» з усіма запитаннями та відповідями, що використовуються у вікторині.

id [PK] bigint	answer_option_1 character varying (255)	answer_option_2 character varying (255)	answer_option_3 character varying (255)	answer_option_4 character varying (255)	correct_answer integer	question_text character varying (255)
1	Central Processing Unit	Computer Personal Unit	Central Process Unit	Computer Power Unit	1	What does CPU stand for?
2	Python	Kotlin	Swift	Java	2	Which language is primarily used for Android app developmen...
3	Read Access Memory	Random Access Memory	Random Allocated Memory	Ready Access Memory	2	What does RAM stand for?
4	HTML	CSS	JavaScript	SQL	3	Which of the following is a programming language?
5	$O(n)$	$O(\log n)$	$O(n^2)$	$O(1)$	2	What is the time complexity of binary search?
6	Windows	Google Chrome	Oracle	Intel	1	Which of the following is an operating system?
7	HyperText Transfer Protocol	HyperText Transfer Procedure	HyperTransfer Text Protocol	HyperText Temporary Protocol	1	What does HTTP stand for?
8	Apple	Microsoft	Bell Labs	IBM	3	Which company developed the C programming language?
9	Resolve IP addresses	Store passwords	Translate domain names to IP addresses	Encrypt data	3	What is the primary purpose of DNS?
10	MySQL	MongoDB	PostgreSQL	SQLite	2	Which of the following is a NoSQL database?

Рисунок 5.9 – Зміст таблиці «quiz_questions»

Після натискання користувачем кнопки «Game» розпочинається вікторина з таймером, що складається з низки запитань. Одночасно активується додаткова функція – вбудований цифровий помічник, який надає персоналізовану підтримку користувачеві. Помічник здатен відповідати на запити та давати підказки у режимі реального часу (рис. 5.10).

Після завершення проходження вікторини система автоматично розраховує підсумковий бал на основі кількості правильних відповідей. Користувачу надається вибір подальших дій (рис. 5.11):

– «Restart Quiz» – повторне проходження тесту;

- «Home» – повернення на головну сторінку;
- «Scoreboard» – перегляд рейтингу всіх користувачів.

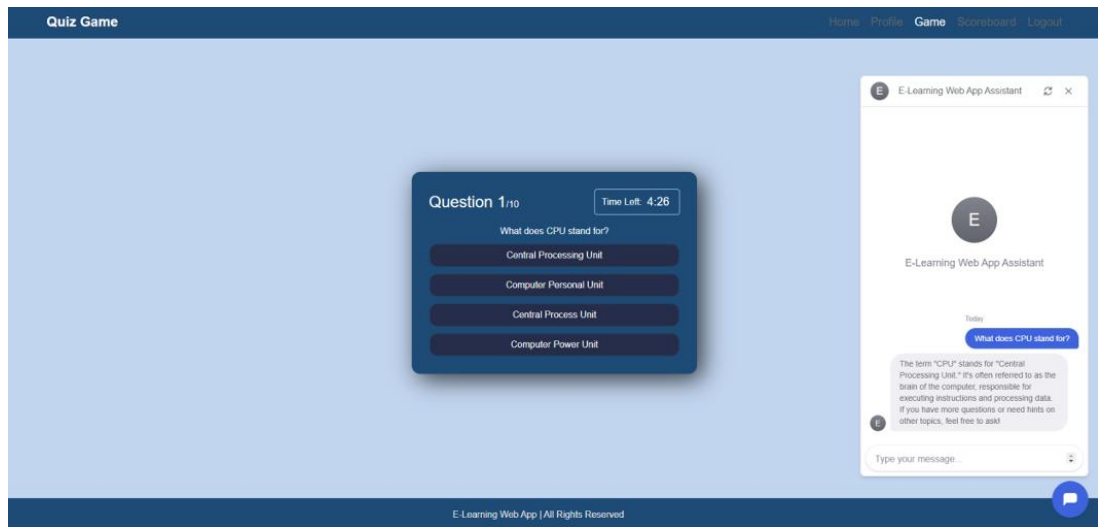


Рисунок 5.10 – Сторінка «Game» – Запитання вікторини з вбудованим цифровим помічником

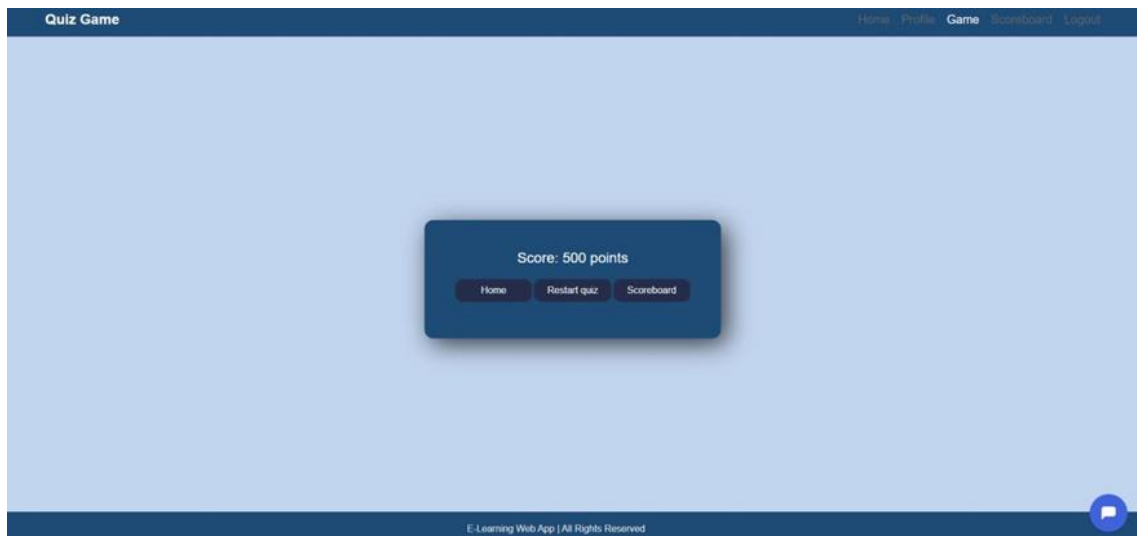


Рисунок 5.11 – Сторінка «Game» – Завершення вікторини та вибір подальших дій

У таблиці рейтингу «Scoreboard» відображаються результати всіх учасників, окрім адміністратора. Дані сортуються за зменшенням бала – від найвищого до найнижчого. Для кожного користувача виводиться його ім'я та відповідний результат (рис. 5.12).

Name	Score
aktinov	500
User Eight	95
User Fourteen	93
User Eighteen	92
User Four	90
User Seven	88
User Three	85
User Thirteen	85

Рисунок 5.12 – Сторінка «Scoreboard» – Відображення результатів для всіх учасників вікторини

У разі, якщо під час проходження вікторини користувач випадково натискає іншу кнопку в навігаційній панелі (navbar), а тест ще не завершено, система попереджає про незбережені відповіді та запитує, як користувач бажає продовжити (рис. 5.13). Це дозволяє уникнути втрати прогресу та забезпечує зручний досвід взаємодії.

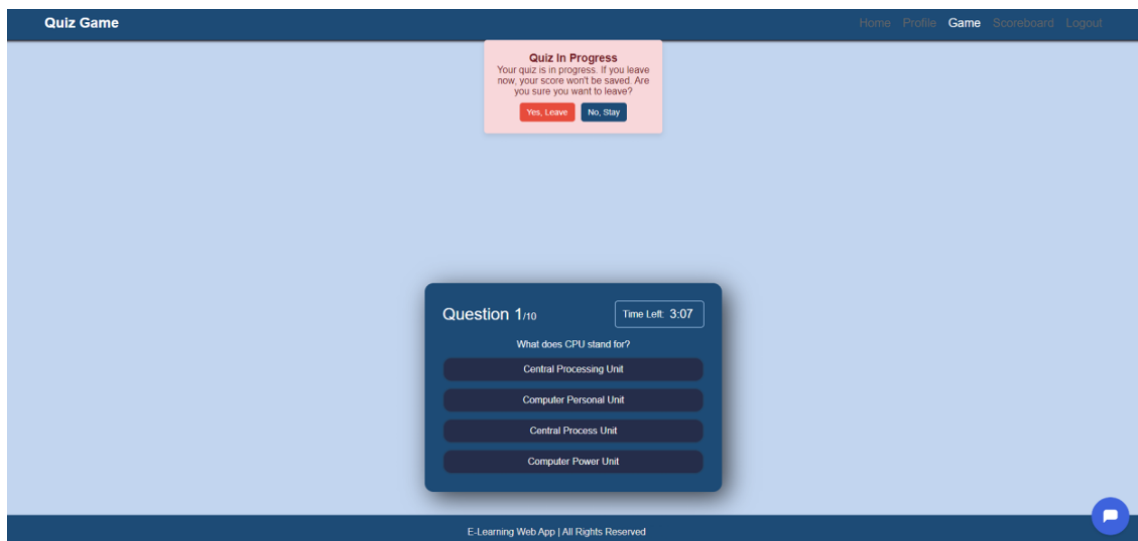


Рисунок 5.13 – Сторінка «Game» – Сповіщення користувача про невиконання тесту

5.1.2 Опис інтерфейсу адміністратора (admin)

У попередньому розділі було продемонстровано роботу освітнього вебзастосунку з точки зору звичайного користувача. Далі розглядається функціональність системи для адміністратора (admin).

Додавання адміністратора до таблиці users бази даних elearningwebapp здійснювалося вручну за допомогою інструменту Postman (рис. 5.14).

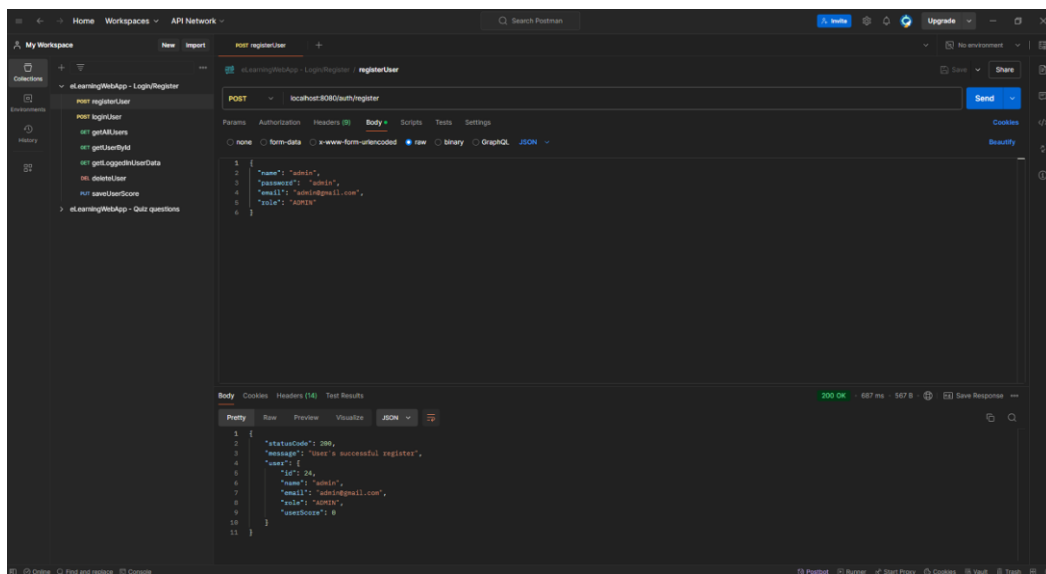


Рисунок 5.14 – Вставка даних адміністратора до бази даних – Postman

Після входу до системи адміністратор перенаправляється на головну сторінку вебзастосунку з доступом до розширених адміністративних функцій (рис. 5.15-5.16).

Адміністратор має доступ до двох основних блоків керування:

- «Manage Users» – перегляд усіх зареєстрованих облікових записів, можливість видалення користувачів як з інтерфейсу застосунку, так і з бази даних (рис. 5.17);

- «Manage Quiz» – повне керування змістом вікторини (рис. 5.18).

У разі спроби випадкового видалення користувача, система виводить повідомлення з підтвердженням дії, запитуючи, чи дійсно адміністратор хоче завершити видалення (рис. 5.19).

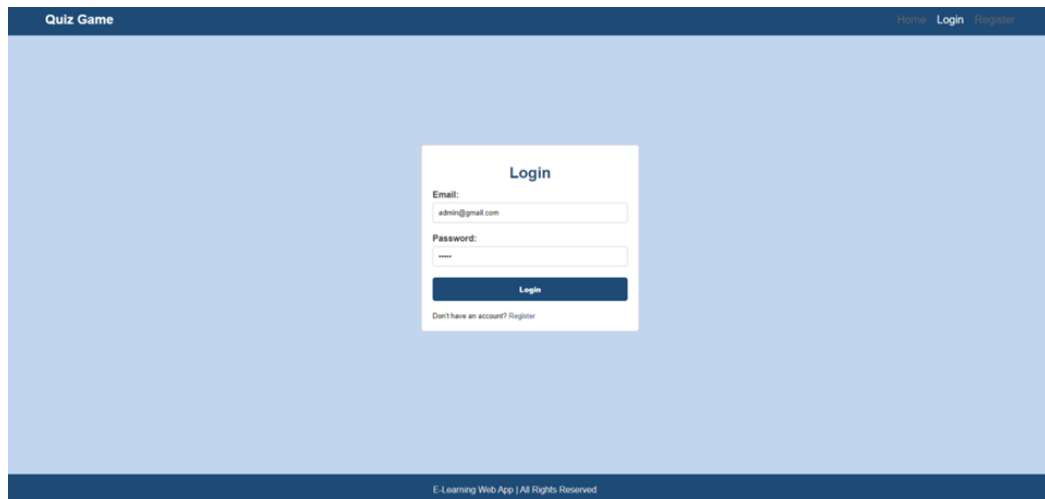


Рисунок 5.15 – Сторінка «Login» – Ідентифікація адміністратора та вхід до програми

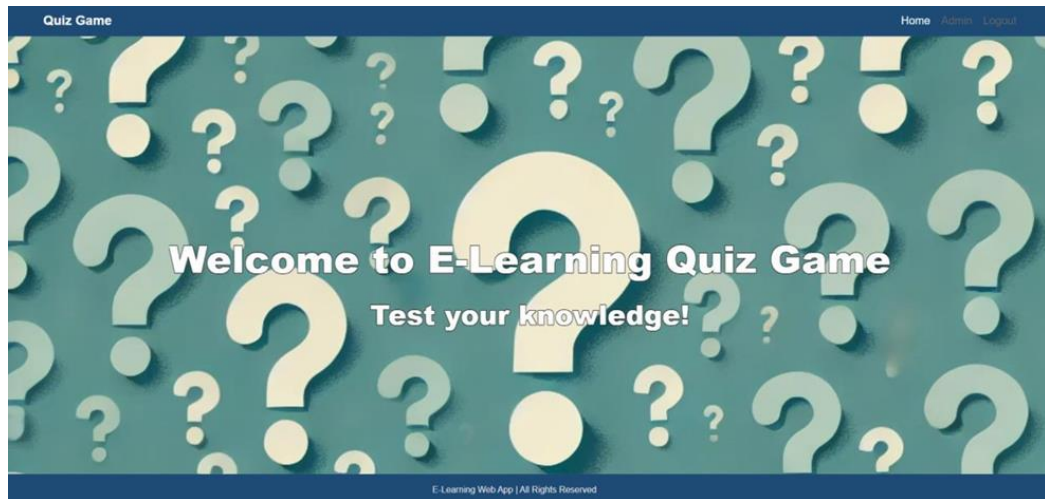


Рисунок 5.16 – Сторінка «Home» – Вигляд адміністратора

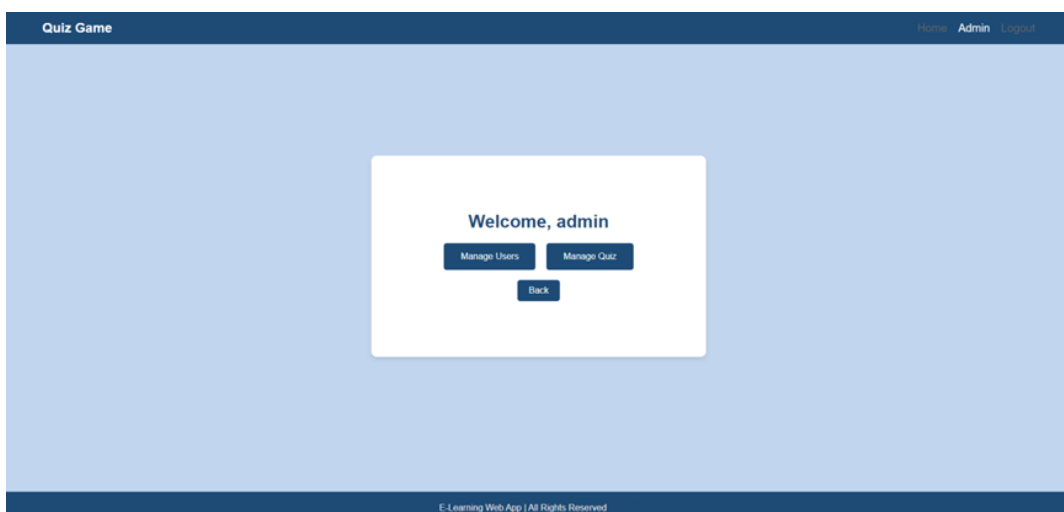


Рисунок 5.17 – Сторінка «Admin» – Функції адміністратора

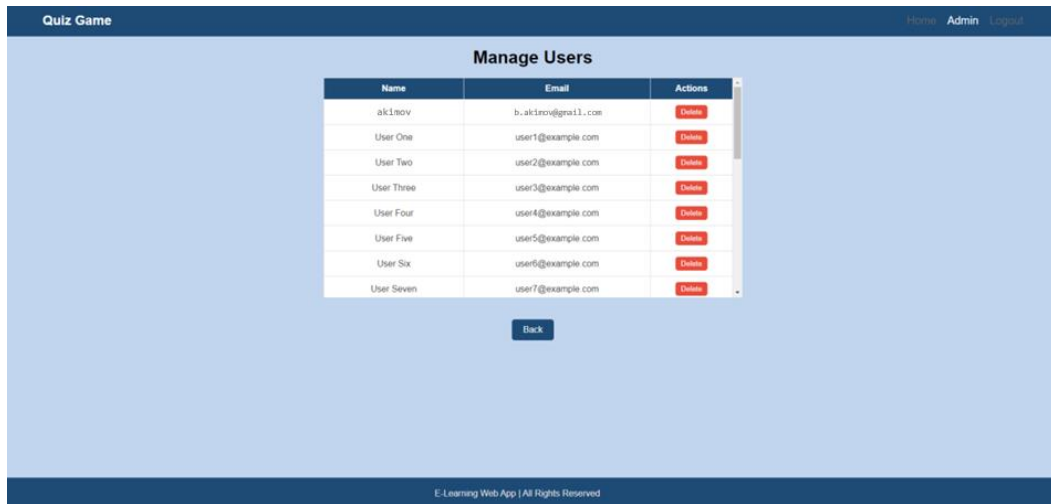


Рисунок 5.18 – Сторінка «Admin» – Відображення всіх користувачів системи та можливість видалення

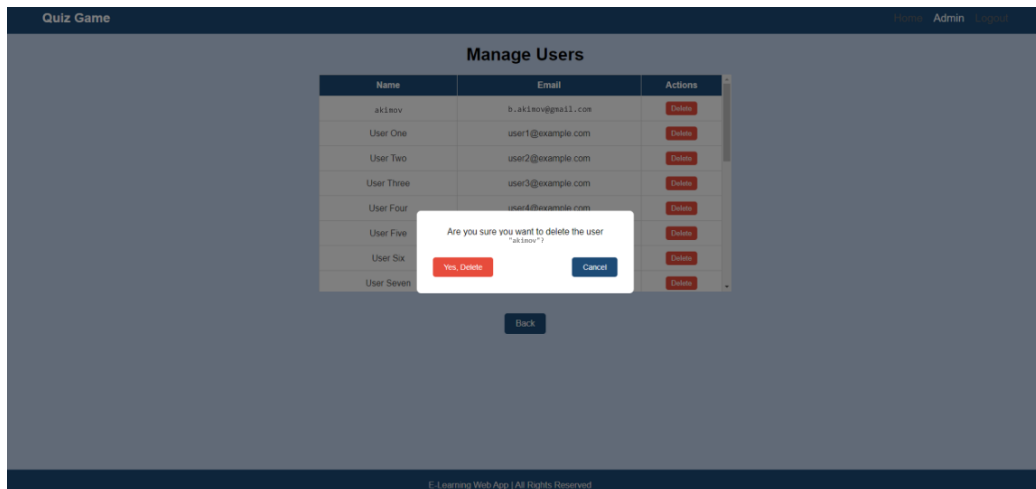


Рисунок 5.19 – Сторінка «Admin» – Попередження про видалення користувача

Якщо адміністратор бажає внести зміни до вікторин, система надає йому відповідний функціонал (рис. 5.20).

Зокрема, він може додати нову вікторину («Add Quiz»), ввівши текст запитання, варіанти відповідей та визначивши одну правильну відповідь (рис. 5.21-5.23).

Крім того, адміністратор має можливість:

- редагувати наявні запитання або відповіді;
- видаляти запитання, варіанти відповідей чи правильні відповіді, які вже збережено в базі даних elearningwebapp, у разі потреби.

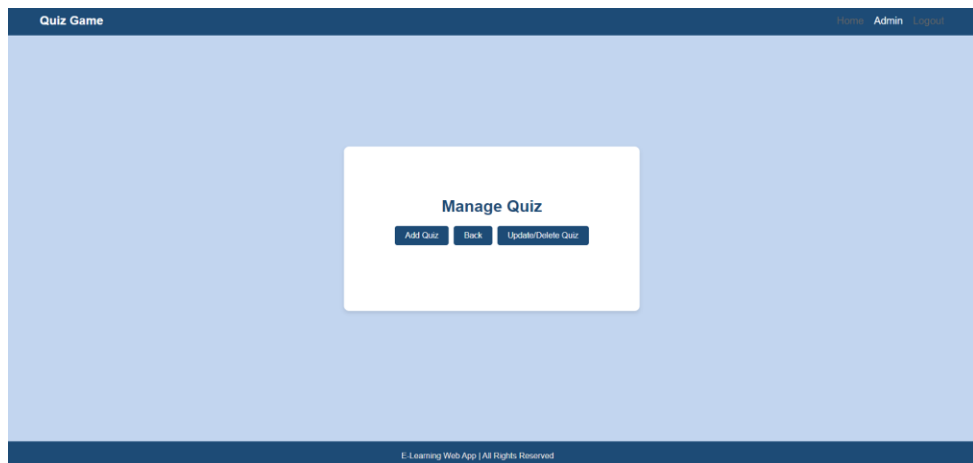


Рисунок 5.20 – Сторінка «Admin/Manage Quiz» – Можливості керування вікторинами

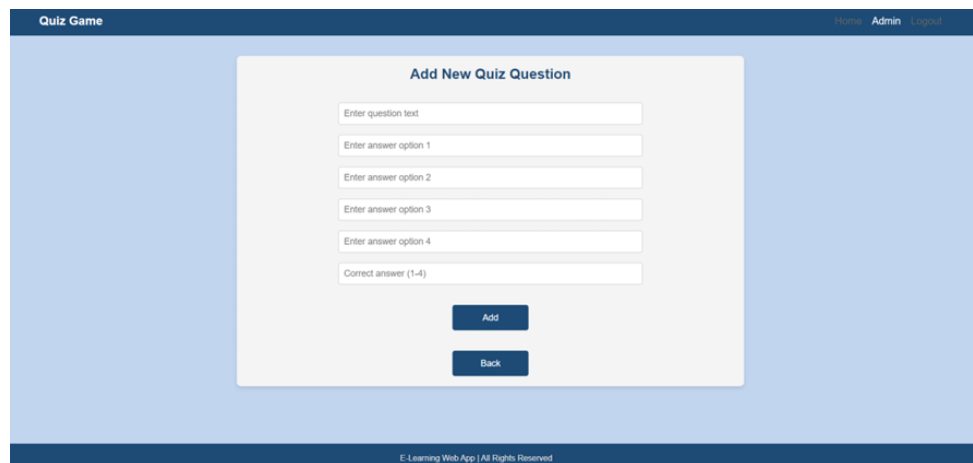


Рисунок 5.21 – Сторінка «Admin/Add New Quiz Question» – Створення вікторини

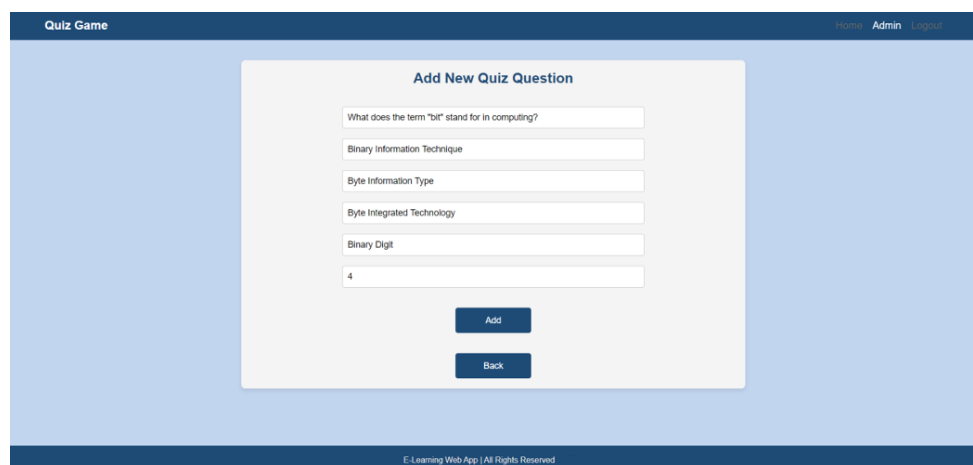


Рисунок 5.22 – Сторінка «Admin/Add New Quiz Question» – Вставка тексту питань та варіантів відповідей

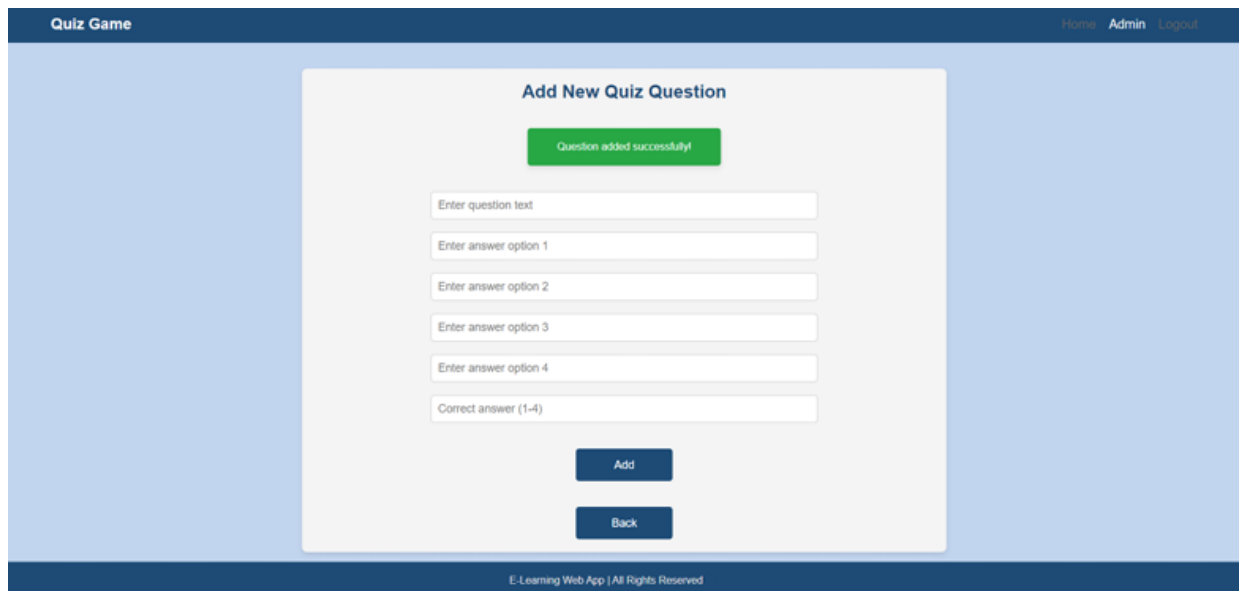


Рисунок 5.23 – Сторінка «Admin/Add New Quiz Question» – Відображення повідомлення про успішне додавання питання вікторини

Як показано на рисунку 5.24-5.25, система дозволяє адміністратору в реальному часі переглядати усі наявні запитання та вносити зміни безпосередньо у веб-інтерфейсі. Ці дії дозволяють гнучко керувати навчальним контентом і своєчасно оновлювати матеріал у системі.

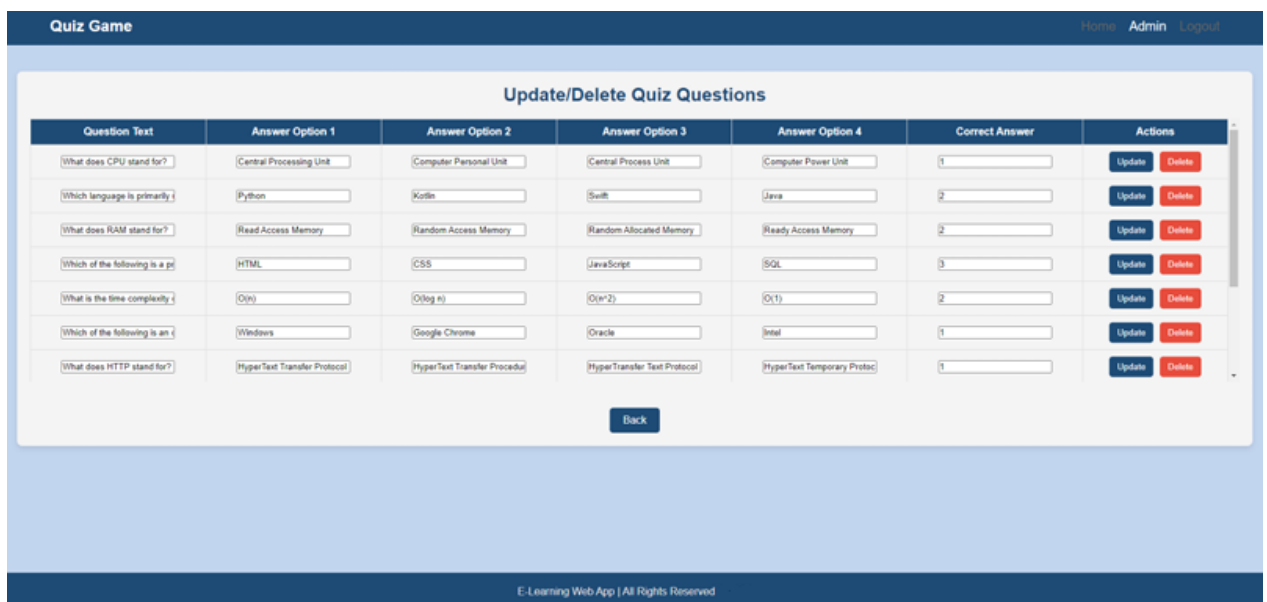


Рисунок 5.24 – Сторінка «Admin/Update/Delete Quiz Questions» – Можливість редагування контенту вікторини

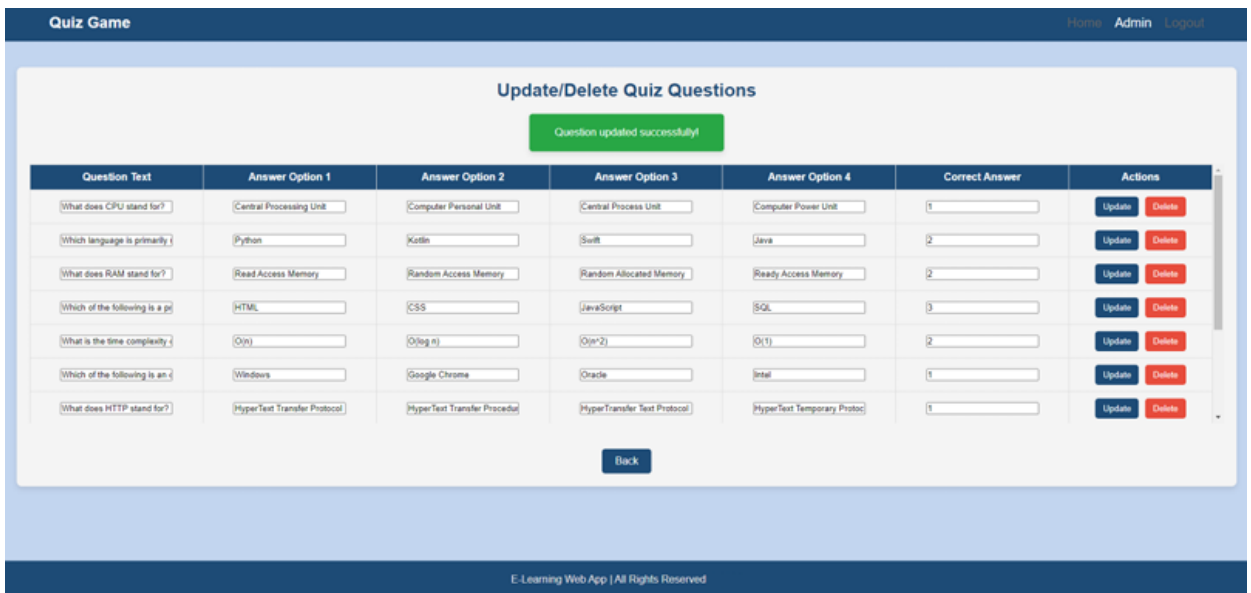


Рисунок 5.25 – Сторінка «Admin/Update/Delete Quiz Questions» – Відображення повідомлення про успішне оновлення контенту вікторини

У випадку спроби видалити запитання з відповідями, система також запитує підтвердження дії та, у разі згоди, повідомляє про успішне виконання операції (рис. 5.26-5.27).

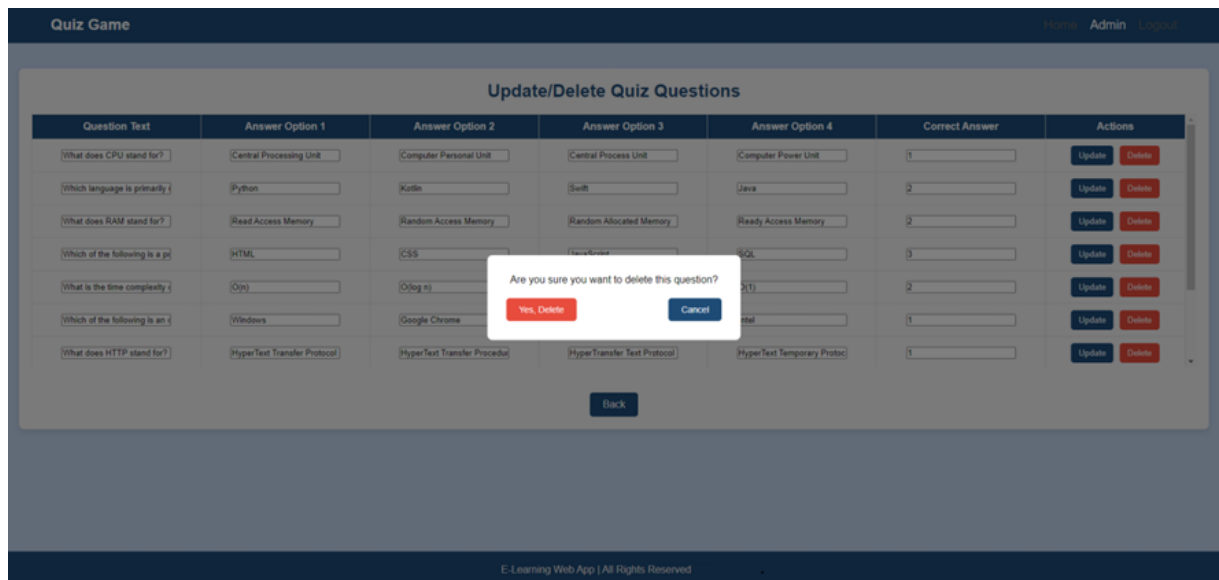


Рисунок 5.26 – Сторінка «Admin/Update/Delete Quiz Questions» – Відображення повідомлення з запитом підтвердження видалення питання вікторини

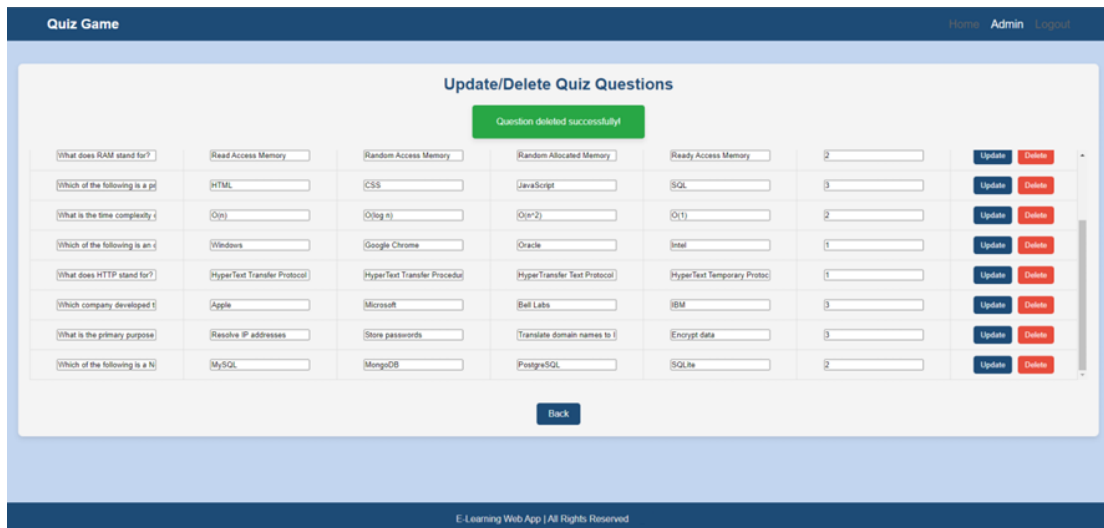


Рисунок 5.27 – Сторінка «Admin/Update/Delete Quiz Questions» – Відображення повідомлення про успішне видалення питання вікторини

При натисканні кнопки Logout, система запитує підтвердження виходу для уникнення випадкового завершення сесії. Це стосується як звичайного користувача, так і адміністратора (рис. 5.28).

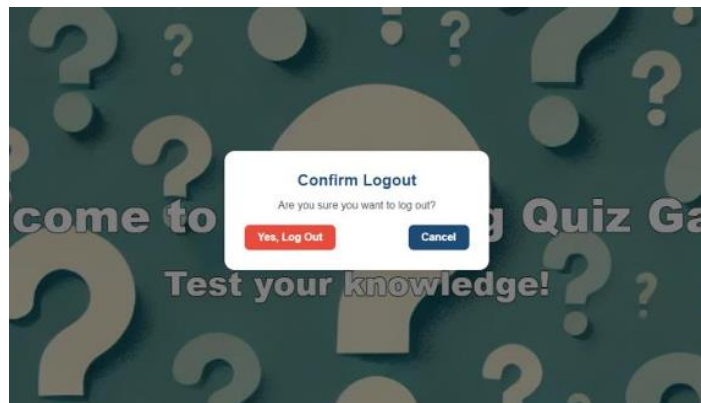


Рисунок 5.28 – Підтвердження виходу з облікового запису – Logout

5.2 Інтеграція цифрового помічника (чат-бота)

Для реалізації цифрового помічника в межах навчального вебзастосунку було використано популярну платформу Botpress. Це відкрите програмне забезпечення з відкритим вихідним кодом, призначене для створення інтелектуальних чат-ботів, які можуть вести інтерактивні розмови в реальному

часі, забезпечуючи персоналізовану підтримку користувачам. Платформа поєднує в собі технології штучного інтелекту (з англ. «Artificial Intelligence», AI) та обробки природної мови (з англ. «Natural Language Processing», NLP), що дозволяє чат-боту розуміти зміст звернень користувача незалежно від складності формулювання. Це забезпечує високу гнучкість при побудові діалогів, а також здатність створювати динамічні й адаптивні відповіді, з урахуванням контексту розмови.

Серед ключових переваг Botpress:

- вбудоване середовище для розробки, тестування та управління діалогами;
- інтеграція з різними платформами (вебзастосунки, месенджери тощо);
- модулі аналітики, які дозволяють оцінити ефективність взаємодії з користувачами;

- гнучкість налаштування і можливість легкої інтеграції до існуючих систем;
- підтримка змін у майбутньому без істотних змін в архітектурі додатку.

На початковому етапі було встановлено платформу Botpress локально у середовищі розробника. Після запуску було проведено первинну конфігурацію, що включала створення нового бота, вибір мови та активацію модулів NLU (Natural Language Understanding).

Як показано на рис. 5.29, платформа Botpress надає візуальний конструктор діалогів Flow Builder, а також емулятор для тестування чат-бота в режимі реального часу. Це дозволяє розробнику будувати логіку розмов без потреби у глибокому програмуванні, використовуючи простий графічний інтерфейс.

Для створення складніших сценаріїв у Botpress передбачено:

- написання дій за допомогою вбудованого JavaScript;
- налаштування інтентів (намірів користувача), сутностей (entities) і слотів (slots) через модуль NLU (Natural Language Understanding);
- можливість зберігання змінних і повторного використання їх у різних частинах розмови для забезпечення персоналізації відповідей;
- підтримка підключення до зовнішніх API або джерел даних, що робить платформу масштабованою та придатною до розширення.

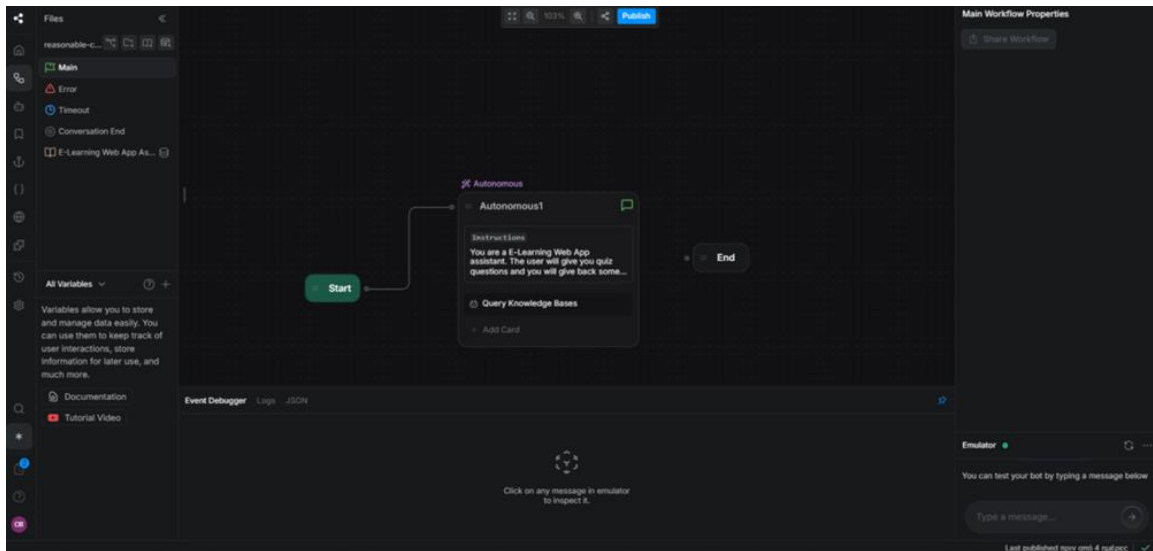


Рисунок 5.29 – Реалізація цифрового помічника у середовищі Botpress – Конструктор діалогів Flow Builder

Таким чином, інтеграція чат-бота Botpress забезпечила платформу інтерактивною підтримкою у режимі реального часу, покращила взаємодію користувача з системою та сприяла індивідуалізації навчального процесу. Реалізоване рішення є масштабованим, гнучким і може бути адаптоване до майбутніх функціональних потреб системи.

5.3 Візуалізація структури бази даних та заповнення тестовими даними

Для перегляду структури бази даних elearningwebapp вебзастосунку, використовувався інструмент pgAdmin 4, за допомогою якого було візуалізовано основні таблиці системи. Як зображено на рисунку 5.30, база даних містить дві ключові таблиці:

- «users», зберігає інформацію про користувачів (ім'я, електронна пошта, пароль, роль, індивідуальний бал);
- «quiz_questions», що містить необхідні поля для збереження запитань вікторини з відповідями.

Обидві таблиці мають поле id, яке виступає первинним ключем, що забезпечує унікальну ідентифікацію записів.

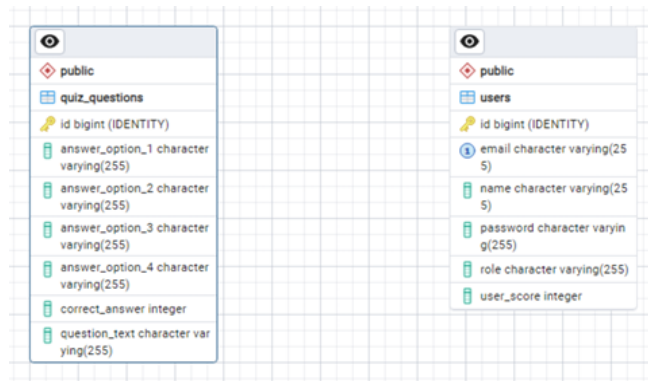


Рисунок 5.30 – Зв'язок між таблицями «users» та «quiz_questions» бази даних elearningwebapp – Інтерфейс інструменту pgAdmin 4

У рамках тестування та демонстрації роботи застосунку було здійснено ручне заповнення таблиці «users» за допомогою SQL-команд. Варто зазначити, що облікові записи адміністратора та користувача з ім'ям akimov мають зашифровані паролі, оскільки їх було зареєстровано через інтерфейс вебзастосунку, який виконує хешування паролів. Для решти користувачів, з метою зручності та створення великої вибірки результатів у рейтингу, дані були додані безпосередньо до бази без проходження бекенд-логіки, що реалізує шифрування. Це дозволило продемонструвати розподіл балів серед користувачів. Решту користувачів було імпортовано за допомогою SQL-команди описаної у Додатку А.

Після виконання запиту `SELECT * FROM users;` в pgAdmin 4, було підтверджено успішне збереження всіх даних у базі elearningwebapp – кожен користувач мав відповідний бал, а адміністратор та користувач «akimov» – хешовані паролі (рис. 5.31).

Як було розглянуто в розділі 5.1.2, адміністратор (admin) має можливість додавати, редагувати та видаляти запитання вікторини безпосередньо через вебінтерфейс. Усі зміни синхронізуються з базою даних у режимі реального часу, що забезпечує зручність керування навчальним контентом без необхідності використання SQL-команд. З метою тестування, для наповнення таблиці «quiz_questions» запитаннями з відповідями було також використано SQL-запит, що наведено в Додатку Б.

	id [PK] bigint	email character varying (255)	name character varying (255)	password character varying (255)	role character varying (255)	user_score integer
1	2	admin@gmail.com	admin	\$2a\$10\$/4MuzJ7QpemOjY1FyciZeYNj1B6z171ohk0fJTNgsN1gH3Wms...	ADMIN	0
2	3	user1@example.com	User One	password1	USER	50
3	4	user2@example.com	User Two	password2	USER	75
4	5	user3@example.com	User Three	password3	USER	85
5	6	user4@example.com	User Four	password4	USER	90
6	7	user5@example.com	User Five	password5	USER	65
7	8	user6@example.com	User Six	password6	USER	78
8	9	user7@example.com	User Seven	password7	USER	88
9	10	user8@example.com	User Eight	password8	USER	95
10	11	user9@example.com	User Nine	password9	USER	50
11	12	user10@example.com	User Ten	password10	USER	80
12	13	user11@example.com	User Eleven	password11	USER	60
13	14	user12@example.com	User Twelve	password12	USER	72
14	15	user13@example.com	User Thirteen	password13	USER	85
15	16	user14@example.com	User Fourteen	password14	USER	93
16	17	user15@example.com	User Fifteen	password15	USER	70
17	18	user16@example.com	User Sixteen	password16	USER	80
18	19	user17@example.com	User Seventeen	password17	USER	68
19	20	user18@example.com	User Eighteen	password18	USER	92
20	21	user19@example.com	User Nineteen	password19	USER	75
21	22	user20@example.com	User Twenty	password20	USER	85
22	1	b.akimov@gmail.com	akimov	\$2a\$10\$5qzkoQZvyglu7Y7bh9pw0R1iqn8ExOK7fIW0wnRSBICM0AENSc...	USER	500

Рисунок 5.31 – Таблиця «users» бази даних elearningwebapp

Після виконання команди `SELECT * FROM quiz_questions;` у середовищі pgAdmin4, було відображено таблицю, що містила всі запитання з відповідними варіантами відповідей та вказаним правильним варіантом за номером (рис. 5.32).

	id [PK] bigint	answer_option_1 character varying (255)	answer_option_2 character varying (255)	answer_option_3 character varying (255)	answer_option_4 character varying (255)	correct_answer integer	question_text character varying (255)
1	1	Central Processing Unit	Computer Personal Unit	Central Process Unit	Computer Power Unit	1	What does CPU stand for?
2	2	Python	Kotlin	Swift	Java	2	Which language is primarily used for Android app developmen...
3	3	Read Access Memory	Random Access Memory	Random Allocated Memory	Ready Access Memory	2	What does RAM stand for?
4	4	HTML	CSS	JavaScript	SQL	3	Which of the following is a programming language?
5	5	O(n)	O(log n)	O(n ²)	O(1)	2	What is the time complexity of binary search?
6	6	Windows	Google Chrome	Oracle	Intel	1	Which of the following is an operating system?
7	7	HyperText Transfer Protocol	HyperText Transfer Procedure	HyperTransfer Text Protocol	HyperText Temporary Protocol	1	What does HTTP stand for?
8	8	Apple	Microsoft	Bell Labs	IBM	3	Which company developed the C programming language?
9	9	Resolve IP addresses	Store passwords	Translate domain names to IP addresses	Encrypt data	3	What is the primary purpose of DNS?
10	10	MySQL	MongoDB	PostgreSQL	SQLite	2	Which of the following is a NoSQL database?

Рисунок 5.32 – Зміст таблиці «quiz_questions» бази даних elearningwebapp

Таким чином, за допомогою інструмента pgAdmin 4 було успішно реалізовано як візуальне представлення структури бази даних, так і процес додавання тестових записів для демонстрації роботи застосунку. Це забезпечило повноцінне тестування функціональності рейтингу, вікторини та авторизації користувачів.

6 ВИКЛИКИ ТА ПОДАЛЬШИЙ РОЗВИТОК СИСТЕМИ

У процесі розробки вебзастосунку для електронного навчання виникла низка технічних та організаційних викликів, подолання яких було необхідним для створення сучасної, функціональної та зручною для користувача системи з інтерактивним контентом. Нижче наведено основні труднощі та описано шляхи, якими вони були вирішені в межах цього проєкту.

6.1 Масштабування даних та продуктивність PostgreSQL

Однією з початкових проблем було зниження продуктивності при обробці великих обсягів даних у базі даних PostgreSQL. Зокрема, під час виконання запитів спостерігалася тимчасова затримка в роботі інтерфейсу. Для вирішення проблеми, було проведено оптимізацію SQL-запитів, реалізовано індексацію ключових полів (наприклад, email, quiz_code), а також застосовано нормалізацію структури бази даних для зменшення надмірності та підвищення ефективності доступу до даних [5].

6.2 Складність у синхронізації Front-End та Back-End

На початкових етапах інтеграції між React (Front-End) та Spring Boot (Back-End) виникли труднощі з обміном даними. Особливо це стосувалося асинхронної обробки запитів та передачі користувацької інформації. Без попереднього досвіду створення подібних систем налагодити ефективну взаємодію було складно. Але поглиблене вивчення офіційної документації обох фреймворків, аналіз прикладів та керівництв, а також тестування REST-запитів через Postman дозволили правильно налаштувати обмін даними через API [8]. Це суттєво пришвидшило та стабілізувало розробку.

6.3 Проблеми з авторизацією через JWT

Під час впровадження механізму JWT-автентифікації спочатку система не змогла успішно верифікувати користувача. Дані зберігалися в базі, проте не відбувалася передача токена між клієнтом і сервером. Як результат – користувач не міг увійти в систему. Подалати проблему допомогло проведення налагодження обміну токенами в заголовках HTTP-запитів, внесено корективи до логіки перевірки підпису токена, а також реалізовано обробку помилок авторизації на фронтенді.

6.4 Захист персональних даних користувачів

З міркувань безпеки постала необхідність захищеного зберігання паролів та впровадження механізмів протидії несанкціонованому доступу до даних. Це ускладнило реалізацію логіки реєстрації та входу. Як результат, було інтегровано бібліотеку BCrypt для хешування паролів, впроваджено RBAC (контроль доступу на основі ролей), що забезпечило розмежування прав між звичайними користувачами та адміністраторами.

6.5 Напрями подальшого розвитку системи

У межах даної кваліфікаційної роботи було розроблено повнофункціональний навчальний вебзастосунок з інтегрованим цифровим помічником. Система побудована із застосуванням сучасного стеку технологій: Java, JavaScript, Spring Boot, React, PostgreSQL, а також платформи Wordpress з відкритим кодом для реалізації інтелектуального чат-бота.

Окрім вже реалізованих функцій, у майбутньому можливе впровадження низки розширень, які дозволять покращити функціонал, підвищити зручність використання та зробити платформу більш масштабованою. Потенційні функціональні вдосконалення:

- авторизація через Google API (OAuth 2.0) для безпечного входу до системи;
- альтернативні способи входу – з використанням облікових записів Facebook або Apple;
- формування вікторини за інтересами користувача (персоналізований контент);
- динамічне оновлення запитань вікторини за допомогою готових API (наприклад, з популярних ігор або платформ);
- підтримка багатомовності вікторин, зокрема англійської, української, німецької тощо;
- створення окремої аналітичної панелі для адміністратора – статистика участі користувачів у тестах, успішність, активність;
- push-сповіщення про оновлення застосунку або зміни в контенті;
- можливість роботи у режимі офлайн, із синхронізацією даних при відновленні підключення;
- покращена адаптивність інтерфейсу для мобільних пристроїв і планшетів;
- відкрите API для сторонніх розробників – можливість додавати власні запитання до системи з зовнішніх платформ;
- застосування методів машинного навчання (ML) для аналізу поведінки користувачів, виявлення прогалин у знаннях тощо;
- розширення функціоналу вікторин, наприклад: демонстрація відео, зображень, інтерактивних підказок;
- форма зворотного зв'язку для повідомлення про помилки, технічні проблеми чи пропозиції користувачів;

Реалізація вказаних розширень дозволить суттєво підвищити якість користувацького досвіду, зробити застосунок гнучким, персоналізованим та масштабованим, а також відкриє нові можливості для інтеграції з іншими платформами. Це перетворить освітню систему на конкурентоспроможний інструмент у сфері онлайн-навчання.

7 ЕКОНОМІЧНА ЧАСТИНА

У результаті виконання кваліфікаційної роботи розроблено навчальний вебзастосунок з інтегрованим цифровим помічником. У сучасних умовах розвитку цифрових технологій значна увага приділяється впровадженню інноваційних ІТ-рішень в освітню сферу. Створення програмного забезпечення для підтримки навчального процесу має бути не лише технічно обґрунтованим, а й економічно доцільним. Враховуючи обмежені ресурси у багатьох освітніх закладах, будь-яке технічне рішення має відповідати критеріям ефективності, рентабельності та масштабованості.

Метою даного розділу є визначення вартості розробки вебзастосунку навчального призначення з вбудованим цифровим помічником, обчислення витрат на його реалізацію, визначення рівня прибутковості проєкту, а також формування остаточної ціни розробки з урахуванням податків.

7.1 Характеристика об'єкта економічного аналізу

Розроблений вебзастосунок являє собою електронну навчальну платформу, що включає:

- систему тестування користувачів за допомогою вікторин;
- можливість реєстрації та авторизації користувачів із збереженням їх особистих результатів;
- механізм гейміфікації з персональним профілем та накопиченням балів;
- інтегрований цифровий помічник (чат-бот), реалізований на базі Botpress;
- базу даних результатів та аналітичний модуль (таблиця лідерів, статистика тощо);
- рольову модель доступу: адміністратор/користувач.

Програмний продукт реалізований за допомогою сучасного стеку технологій: React, Spring Boot, PostgreSQL, Botpress, що забезпечує високу швидкість роботи, розширюваність і можливість подальшої підтримки та масштабування.

7.2 Визначення витрат на розробку

Процес створення програмного забезпечення включав декілька етапів: постановка задачі, розробка технічного завдання, проектування архітектури системи, реалізація коду (бекенд та фронтенд), тестування, налагодження та оформлення супровідної документації.

Для розрахунку загальної вартості було взято середню заробітну плату відповідних фахівців, а також враховано витрати на електроенергію, амортизацію обладнання та соціальні відрахування.

У розробці умовно брали участь три фахівці: Фронтенд-розробник, Бекенд-розробник та UI/UX дизайнер. Розрахунок витрат на заробітну плату наведено у таблиці 7.1.

Таблиця 7.1 – Розрахунок витрат на заробітну плату

№	Посада	Годинна ставка, грн	Обсяг роботи, год	Вартість, грн
1	Фронтенд-розробник	70,00	64	4 480,00
2	Бекенд-розробник	70,00	64	4 480,00
3	UI/UX дизайнер	60,00	64	3 840,00
Разом:				12 800,00

Додаткові виплати (відпустки, лікарняні тощо) умовно становлять 20% від основної:

$$12\,800 * 0,2 = 2\,560,00 \text{ грн.}$$

Єдиний соціальний внесок (ЄСВ) (22%) нараховується на загальну суму заробітної плати:

$$(12\,800 + 2\,560) * 0,22 = 3\,379,60 \text{ грн.}$$

До інших витрат, що формують собівартість розробки, належать витрати на електроенергію та амортизація техніки.

Витрати на електроенергію для роботи 3 комп'ютерів протягом 64 годин:

$$3 * 0,7 * 64 * 4,32 \text{ грн} = 580,61 \text{ грн.}$$

Вартість одного комп'ютера – 24 000,00 грн. Термін експлуатації – 3 роки по 254 робочі дні. Таким чином, амортизація за 1 рік та за час роботи над проєктом становитиме:

$$(24\,000,00 * 3) / (3 * 254 * 8) = 11,81 \text{ грн,}$$

$$11,81 * 64 = 755,91 \text{ грн.}$$

Загальна собівартість розробки наведена у таблиці 7.2.

Таблиця 7.2 – Загальна собівартість розробки

Стаття витрат	Сума, грн
Основна заробітна плата	12 800,00
Додаткова заробітна плата (20 %)	2 560,00
ЄСВ (22 %)	3 379,00
Електроенергія	580,61
Амортизація обладнання	755,91
Разом собівартість	20 075,71

Для оцінки економічної ефективності визначається прибуток у розмірі 30 % від собівартості:

$$20\,075,71 * 0,3 = 5\,765,43 \text{ грн.}$$

Формування ціни з урахуванням податку на додану вартість (ПДВ) наведено у таблиці 7.3.

Таблиця 7.3 – Розрахунок ціни розробки

Показник	Значення, грн
Собівартість	20 075,71
Прибуток (30 %)	6 022,71
Ціна без ПДВ	26 098,43
ПДВ (20 %)	5 219,69
Ціна з ПДВ	31 318,11

За результатами розрахунків встановлено, що загальні витрати на розробку освітнього вебзастосунку з інтегрованим цифровим помічником становлять 20 075,71 грн. За умови реалізації проєкту з 30 % прибутком кінцева ціна з урахуванням ПДВ – 31 318,11 грн.

Проєкт має високий рівень ефективності завдяки використанню безкоштовних технологій з відкритим кодом, можливості масштабування, низьким витратам на підтримку та адаптивності до різних навчальних середовищ. Оскільки система є універсальною, вона може бути використана в навчальних закладах різного рівня – від шкіл до університетів, а також у рамках онлайн-курсів та програм підвищення кваліфікації.

Запропоноване рішення є перспективним як з технологічного, так і з економічного боку, та має потенціал до подальшої комерціалізації або розширення функціоналу з урахуванням потреб конкретних замовників або освітніх платформ.

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було здійснено повний цикл розробки навчального веб-застосунку з інтегрованим цифровим асистентом. У результаті:

- проведено аналіз предметної галузі та виявлено недоліки існуючих e-learning рішень;
- сформульовано вимоги до функціональності та архітектури системи;
- реалізовано повнофункціональний веб-додаток на основі сучасного стеку технологій: React, Spring Boot, PostgreSQL та Botpress.

Особливу увагу було приділено архітектурному проектуванню, забезпеченню безпеки (включно з автентифікацією за допомогою JWT), розмежуванню ролей користувачів, а також реалізації персоналізованої взаємодії завдяки вбудованому чат-боту.

Розроблена система охоплює такі ключові функціональні можливості:

- створення та проходження вікторин з IT-тематики;
- відстеження індивідуального прогресу та результатів користувача;
- отримання рекомендацій і підтримки в режимі реального часу через цифрового асистента;
- адміністрування контенту та керування користувачами через окремий інтерфейс для адміністратора.

Інтерфейс застосунку поєднує інтерактивність, адаптивність до пристроїв та елементи персоналізації, що сприяє підвищенню зацікавленості користувача та ефективності освітнього процесу. Інтеграція Botpress забезпечила інтелектуальний супровід, що відповідає актуальним трендам у цифровій педагогіці.

У рамках проєкту також було проведено економічне обґрунтування, яке дозволило визначити повну собівартість розробки, розрахувати очікуваний прибуток та сформулювати фінальну ціну продукту. Встановлено,

що система є економічно доцільною та фінансово обґрунтованою: повна вартість реалізації MVP-версії становить менше 30 тис. грн, а за умови масштабування та подальшого розвитку продукт має високий потенціал комерціалізації.

Отримані результати засвідчують практичну цінність створеного рішення для застосування як у формальній, так і в неформальній освіті. Запропонована платформа має потенціал до масштабування. У подальшому її функціонал може бути доповнений гейміфікацією, багатомовністю, інтеграцією з LMS-платформами (наприклад, Moodle, Google Classroom), аналітикою поведінки користувачів тощо.

Це відкриває перспективи впровадження платформи в масштабні освітні ініціативи та цифрові трансформаційні проекти у сфері навчання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Burinskiene A. E-Learning: Theory and Practice, 2nd International Scientific Conference «Teaching Methods for Economics and Business Sciences», 2019. URL: https://www.researchgate.net/publication/339451234_E-Learning_Theory_and_Practice (дата звернення: 01.05.2025).
2. Ofoeda R., Boateng J., Effah J. Application Programming Interface (API) Research: A Review of the Past to Inform the Future», International Journal of Enterprise Information Systems, 2019. URL: https://www.researchgate.net/publication/334145268_Application_Programming_Interface_API_Research (дата звернення: 03.05.2025).
3. Luan X. Implementation and Analysis of Software Development in Spring Boot, M.S. thesis, Dept. of Computer Science, California State Polytechnic University, Pomona, 2021.
4. Mahindrakar P., Pujer U., Insights of JSON Web Token, International Journal of Recent Technology and Engineering (IJRTE), 2020.
5. Lockhart T. PostgreSQL User's Guide, The PostgreSQL Development Team, 1996-2000.
6. Chen S., Thaduri U. R., Ballamudi V. K. R., Front-End Development in React: An Overview, Engineering International, 2019.
7. Deineko Zh., Sotnik S., Vovk O., Lyashenko V. Features of Database Types // International Journal of Engineering and Information Systems (IJEAIS) ISSN: 2643-640X. Vol. 5, Issue 10, 2021, P. 73-80.
8. Kore P.P., Lohar M.J., Surve M.T., Jadhav S., API Testing Using Postman Tool, MCA Dept., Shivaji University, Kolhapur, and Yashoda Technical Campus, Satara.
9. Бізюк А.В., Пономарьова О.В. Аналіз функціональних можливостей елементів дизайну для мобільного застосунку // Поліграфічні, мультимедійні та web-технології. 2024. Т. 1. С. 207-208.

10. Haixia He, E-Learning Theory. URL: <https://opentext.wsu.edu/theoreticalmodelsforteachingandresearch/chapter/e-learning-theory/> (дата звернення: 15.05.2025).

11. Anthony G. Picciano, Theories and Frameworks for Online Education: Seeking an Integrated Model, City University of New York Graduate Center and Hunter College. URL: <https://files.eric.ed.gov/fulltext/EJ1154117.pdf> (дата звернення: 14.05.2025).

12. Дейнеко Ж., Леня В. Перспективи та ризики використання штучного інтелекту у самоосвіті // Поліграфічні, мультимедійні та web-технології. 2025. Т. 1. С. 289-290.

13. Влащенко Л.Г., Дейнеко Ж.В., Нікітенко О.М. Створення інтерактивних тестів у видавничій системі LaTeX // Поліграфічні, мультимедійні та web-технології. 2023. Т. 1. С. 261-263.

14. Вовк О.В., Задорожна В.К. Дослідження етапів планування UI та UX сайту // Поліграфічні, мультимедійні та web-технології. 2025. Т. 2. С. 28-30.

15. Глюза М., Вовк О. Основи UX-research // Інформаційні технології в сучасному світі: дослідження молодих вчених. 2024. С. 112.

16. Бізюк А.В., Каряка Ю.М. Дослідження UX/UI прототипу: перевірка вдосконаленої методології на підвищення ефективності взаємодії користувачів з продуктом // Поліграфічні, мультимедійні та web-технології. 2023. Т. 1. С. 122-123.