

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ

ННІ/факультет Навчально-науковий інститут економіки та бізнес-освіти  
Кафедра Економіки та цифрового бізнесу  
Спеціальність 122 «Комп'ютерні науки»  
Форма навчання Денна

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

Стружевського Іллі Андрійовича

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

на тему Розробка Web-застосунку для вивчення англійських слів

*(повна назва теми)*

за матеріалами \_\_\_\_\_

*(повна назва бази дослідження)*

науковий керівник К.Т.Н. Селезньов М.Є.

*(наук. ступінь, вчене звання)*

*(підпис)*

*(прізвище, ініціали)*

**Робота допущена до захисту в ЕК**

Протокол засідання кафедри  
від 09 червня 2025р. № 12

Завідувач кафедри \_\_\_\_\_

*(підпис)*

к.е.н., доцент  
*наук. ступінь, вчене звання*

**Радько В.М.**  
*прізвище, ініціали*

ЗАТВЕРДЖЕНО

2

Наказ Міністерства освіти і науки, молоді та спорту України  
29 березня 2012 року № 384

Форма № Н-9.01

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ**  
**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ ТА БІЗНЕС-ОСВІТИ**  
( повне найменування вищого навчального закладу )

Кафедра економіки та цифрового бізнесу  
Освітній ступінь бакалавр  
Спеціальність 122 «Комп'ютерні науки»

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри \_\_\_\_\_ В.М. Радько**

“07” квітня 2025 року

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА ЗДОБУВАЧУ**

\_\_\_\_\_ Стружевському Іллі Андрійовичу \_\_\_\_\_

1. Тема роботи «Розробка Web-застосунку для вивчення англійських слів»

науковий керівник роботи Селезньов Максим Євгенович

затверджені наказом вищого навчального закладу від «04» квітня 2025 р. № 224-ст (д/ф)  
№ 151-ст (з/ф)

2. Строк подання здобувачем роботи 31.05.2025р.

3. Зміст кваліфікаційної роботи бакалавра, об'єкт, предмет та мета дослідження:

Розділ 1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ WEB-ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ  
АНГЛІЙСЬКИХ СЛІВ

Розділ 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА WEB-ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ  
АНГЛІЙСЬКИХ СЛІВ

Розділ 3 ТЕСТУВАННЯ, АНАЛІЗ ТА ПЕРСПЕКТИВИ РОЗВИТКУ WEB-ЗАСТОСУНКУ

Об'єкт дослідження - процес створення інтерактивних інформаційних Web-систем для підтримки  
самостійного вивчення іноземних мов.

Предмет дослідження - функціональні та технічні аспекти розробки Web-застосунку для вивчення  
англійських слів

Мета кваліфікаційної роботи бакалавра – розробка Web-застосунку для ефективного вивчення  
англійських слів в інтерактивному режимі, з можливістю фіксування прогресу та адаптивної  
побудови лексичних тестів.

4. Дата видачі завдання 04.04.2025р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Відмітка керівника про виконання етапів (дата, підпис)
1	Підготовка розділу 1	до 28.04.2025р.	25.04.2025
2	Підготовка розділу 2	до 16.05.2025р.	15.05.2025
3	Підготовка розділу 3	до 30.05.2025р.	29.05.2025
4	Реєстрація завершеної дипломної роботи	до 31.05.2025р.	30.05.2025
5	Отримання відгуку від наукового керівника	03-04.06.2025р.	04.06.2025
6	Отримання зовнішньої рецензії	05-06.06.2025р.	06.06.2025
7	Перевірка кваліфікаційної роботи на плагіат	02-09.06.2025р.	04.06.2025
8	Попередній захист кваліфікаційної роботи на кафедрі	03.06.2025р.	03.06.2025
9	Допуск кафедрою кваліфікаційної роботи до захисту	09.06.2025р.	09.06.2025
10	Підготовка студента до захисту в ЕК	до 17.06.2025р.	17.06.2025

Завдання підготував науковий керівник \_\_\_\_\_

(підпис)

**Селезньов М. Є**

(прізвище та ініціали)

Завдання одержав здобувач \_\_\_\_\_

(підпис)

**Стружевський І. А.**

(прізвище та ініціали)

*Примітки:*

1. Форму призначено для видачі завдання здобувачу на виконання кваліфікаційної роботи бакалавра і контролю за ходом роботи з боку кафедри.
2. Розробляється керівником кваліфікаційної роботи. Видається кафедрою.
3. Формат бланка А4 (210×297 мм), 2 сторінки.

## РЕФЕРАТ

Робота містить 54 сторінки, 14 рисунків, 49 джерел 10 таблиць і 3 додатки.

Об'єкт дослідження: процес створення інтерактивних інформаційних Web-систем для підтримки самостійного вивчення іноземних мов.

Предмет дослідження: функціональні та технічні аспекти розробки Web-застосунку для вивчення англійських слів.

Мета дослідження: розробка Web-застосунку для ефективного вивчення англійських слів в інтерактивному режимі, з можливістю фіксування прогресу та адаптивної побудови лексичних тестів.

У ході роботи було спроектовано та реалізовано сучасний Web-застосунок для вивчення англійської лексики, що відповідає поточним тенденціям цифрової освіти та вимогам до інтерактивних освітніх платформ. Обґрунтовано доцільність використання клієнт-серверної архітектури на основі технологічного стеку MERN (MongoDB, Express.js, React, Node.js), яка забезпечує гнучкість, масштабованість та високу інтерактивність застосунку.

Область застосування: розроблений Web-застосунок може бути використаний як інструмент для практичного впровадження сучасних цифрових технологій у процес вивчення англійської мови.

АНГЛІЙСЬКА МОВА, ВИВЧЕННЯ СЛІВ, WEB-ЗАСТОСУНОК, UI-ДИЗАЙН, UX-ДИЗАЙН, WEB-РОЗРОБКА, WEB-РЕСУРС.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....	6
ВСТУП .....	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ WEB-ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКИХ СЛІВ .....	9
1.1 Огляд підходів до вивчення англійських слів у цифровому середовищі та існуючих онлайн-сервісів .....	9
1.2 Огляд сучасних Web-технологій для реалізації навчальних застосунків .....	14
1.3 Вибір стеку технологій для реалізації Web-застосунку .....	19
Висновки до розділу 1 .....	23
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА WEB-ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКИХ СЛІВ .....	25
2.1 Архітектура Web-застосунку та логіка взаємодії з користувачем .....	25
2.2 Розробка бази даних для збереження слів, тестів і прогресу користувачів .....	29
2.3 Розробка основних модулів Web-застосунку .....	34
Висновки до розділу 2 .....	40
РОЗДІЛ 3 ТЕСТУВАННЯ, АНАЛІЗ ТА ПЕРСПЕКТИВИ РОЗВИТКУ WEB- ЗАСТОСУНКУ .....	42
3.1 Реалізація інтерфейсу користувача з адаптивним дизайном .....	42
3.2 Функціонал створеного Web-застосунку та сценарії взаємодії .....	47
3.3 Аналіз ефективності використання Web-застосунку та перспективи його вдосконалення .....	56
Висновки до розділу 3 .....	57
ВИСНОВКИ .....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	61
ДОДАТКИ .....	66

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API – Application Programming Interface;

CRUD – Create, Read, Update, Delete;

CSS – Cascading Style Sheets;

DB – Database;

HTML – HyperText Markup Language;

HTTPS – Hypertext Transfer Protocol Secure;

JS – JavaScript;

JWT – JSON Web Token;

MERN – MongoDB, Express.js, React, Node.js;

SPA – Single Page Application;

UI – User Interface;

UX – User Experience.

## ВСТУП

У сучасних умовах цифрової трансформації освіти значного поширення набули інформаційно-комунікаційні технології, які активно впроваджуються у навчальний процес усіх рівнів. Зокрема, вивчення іноземних мов, зокрема англійської, дедалі частіше здійснюється за допомогою електронних платформ, мобільних застосунків та Web-ресурсів. В умовах глобалізації англійська мова стала одним з основних засобів міжнародного спілкування, тому володіння нею є важливою конкурентною перевагою для фахівців різних галузей. Проте традиційні методи викладання часто не забезпечують належного рівня мотивації, персоналізації та гнучкості навчання, що знижує їхню ефективність.

Розвиток технологій відкриває нові можливості для автоматизації навчального процесу, формування адаптивного освітнього середовища та інтерактивної взаємодії з навчальним матеріалом. Провідні наукові установи та освітні платформи впроваджують у свої системи елементи гейміфікації, трекінг прогресу, використання штучного інтелекту для адаптації складності матеріалу тощо. Водночас, більшість із наявних рішень є або повністю комерційними, або не надають можливості локального адаптування до конкретних освітніх потреб. Саме тому питання розробки відкритих та кастомізованих Web-застосунків для вивчення англійської мови залишається актуальним і практично значущим.

**Об'єкт дослідження:** процес створення інтерактивних інформаційних Web-систем для підтримки самостійного вивчення іноземних мов.

**Предмет дослідження:** функціональні та технічні аспекти розробки Web-застосунку для вивчення англійських слів.

**Метою дослідження** є розробка Web-застосунку для ефективного вивчення англійських слів в інтерактивному режимі, з можливістю фіксування прогресу та адаптивної побудови лексичних тестів.

Для досягнення поставленої визначені наступні завдання:

- провести аналіз наявних освітніх Web-сервісів і виявити їхні сильні та слабкі сторони;
- сформулювати функціональні та нефункціональні вимоги до Web-застосунку;
- розробити архітектуру застосунку з урахуванням принципів масштабованості та безпеки;
- реалізувати серверну частину Web-застосунку;
- створити клієнтську частину Web-застосунку з динамічними формами та інтерактивним інтерфейсом;
- протестувати працездатність системи та оцінити її ефективність у навчальному контексті.

В процесі проведення дослідження використовувалися такі методики:

- порівняльний аналіз;
- системний аналіз;
- об'єктно-орієнтований підхід;
- моделювання та прототипування;
- тестування та верифікація функціональності.

**Практична значущість результатів** дослідження полягає у створенні інтерактивного Web-ресурсу, який може бути використаний як навчальний інструмент для студентів та викладачів, а також як основа для створення розширеної платформи з підтримкою граматики, вимови та індивідуалізованих траєкторій навчання.

## РОЗДІЛ 1

### ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ WEB-ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКИХ СЛІВ

#### **1.1 Огляд підходів до вивчення англійських слів у цифровому середовищі та існуючих онлайн-сервісів**

У сучасному цифровому суспільстві англійська мова відіграє ключову роль у міжнародній комунікації, професійному розвитку, освіті та науці. У зв'язку з цим спостерігається значне зростання попиту на ефективні способи її вивчення, зокрема на засоби опанування лексики [1]. Одним із найперспективніших напрямів є використання цифрових технологій та Web-застосунків для вивчення англійських слів.

Поглиблені дослідження у галузі лінгводидактики та інформаційних технологій постійно виявляють нові можливості для вдосконалення процесу засвоєння лексичного матеріалу [2]. Зокрема, актуальність цифровізації навчання іноземних мов висуває нові вимоги до індивідуалізованих навчальних траєкторій та адаптивних систем [1]. Це передбачає не просто пропонування статичного набору вправ, а динамічне реагування на прогрес та потреби кожного користувача, коригуючи складність та тип завдань, що є критично важливим для ефективного формування лексичних навичок [2].

Сучасні розробки в галузі Web-технологій активно інтегрують передові підходи для створення більш «інтелектуальних» систем вивчення лексики. Наприклад, використання SPA-додатків [11] дозволяє забезпечити високу інтерактивність та швидкість роботи, що є ключовим для залучення користувачів до довготривалого навчання. Дослідження також показують зростаючу роль AI-технологій у вивченні мов, відкриваючи можливості для персоналізації та аналізу

прогресу учнів на якісно новому рівні [17]. Це дозволяє перейти від універсальних списків слів до контекстуально релевантного та особистісно орієнтованого навчання, що сприяє кращому засвоєнню [5].

Важливим аспектом, що набуває все більшого значення у цифровому вивченні лексики, є технологічна архітектура самих платформ. Використання сучасних фреймворків, таких як React.js [22], а також backend-рішень на основі Node.js та Express.js [23, 24], у поєднанні з NoSQL базами даних, зокрема MongoDB [8, 26], дозволяє створювати гнучкі, масштабовані та продуктивні системи. Це забезпечує можливість інтеграції різноманітних функціональних можливостей, від адаптивної верстки інтерфейсів [14] до безпечного обміну даними [9], що є фундаментальним для надання високоякісних освітніх послуг.

Крім суто технічних аспектів, велике значення має можливість інтеграції Web-застосунків з мобільними платформами та використання Progressive Web Apps (PWA) [18], що забезпечує доступність навчання без прив'язки до стаціонарного комп'ютера. Такий підхід у поєднанні з гейміфікаційними елементами [3] та продуманою візуалізацією результатів навчання [16] дозволяє значно підвищити мотивацію та залученість користувачів. З огляду на це, ефективні Web-системи для вивчення англійської лексики мають не лише підтримувати індивідуальне налаштування та комбіновані методи навчання, але й відповідати високим стандартам щодо зручності інтерфейсу та технологічної надійності, що підтверджується аналізом існуючих онлайн-сервісів [5].

Досліджено, що методи вивчення лексики можна класифікувати на такі основні категорії [2]:

1. Контекстуальне вивчення - вивчення слів у контексті речень або текстів.
2. Мнемонічні техніки - запам'ятовування слів шляхом асоціацій, візуалізації або римування.

3. Повторення з інтервалами - метод, що базується на ефекті кривої забування Еббінгауза [4].

4. Ігрове навчання - використання гейміфікованих елементів для підвищення мотивації та залученості користувачів [3].

5. Переклад і асоціації - вивчення слів шляхом співставлення з рідною мовою або знайомими образами.

Значна частина сучасних Web-сервісів та мобільних застосунків реалізує поєднання вищенаведених методів [5]. Проаналізуємо найбільш популярні рішення: Duolingo - сервіс, який пропонує навчання у форматі гри, з інтерактивними вправами, що охоплюють лексику, граматику, аудіювання та переклад. Перевагами є адаптивність до рівня користувача та миттєвий зворотний зв'язок. Проте у Duolingo спостерігається фрагментарність лексичних тем і відсутність гнучкого налаштування тематики слів. Memrise - сервіс, що використовує мнемонічні методи та алгоритм повторення з інтервалами. В основі системи - асоціації, створені самими користувачами, що сприяє кращому запам'ятовуванню. Водночас відзначено складність навігації в інтерфейсі та надмірну кількість платних функцій. Quizlet - платформа, орієнтована на створення користувацьких карток і тестів. Вона дозволяє вивчати слова за допомогою флеш-карток, а також формує ігрові вправи на їх основі. Пропонується ефективна реалізація повторення, однак обмежено підтримку аудіо та зображень у безкоштовній версії. Lingualeo - інтерактивна платформа, яка підтримує адаптацію словникового запасу під потреби користувача, надає контент у вигляді текстів, відео та тренувань. Досліджено, що основна перевага полягає в гнучкому підході до вибору навчального матеріалу, однак значна частина контенту потребує платного доступу. Також існує підхід застосування повторення з інтервалами, реалізований у таких платформах, як Anki. У цих системах пропонується адаптивне повторення лексичних одиниць відповідно до індивідуальної динаміки

запам'ятовування користувача. Доведено ефективність цього методу для довготривалого збереження слів у пам'яті [4].

У Таблиці 1.1 представлено порівняльну характеристику проаналізованих сервісів.

Таблиця 1.1

**Порівняльна характеристика популярних платформ для вивчення  
англійських слів**

<b>Назва сервісу</b>	<b>Метод навчання</b>	<b>Можливість адаптації</b>	<b>Повторення з інтервалами</b>	<b>Безкоштовний доступ</b>
<b>Duolingo</b>	контекст + гра	часткова	так	так
<b>Memrise</b>	мнемоніка	повна	так	обмежений
<b>Quizlet</b>	картки	повна	так	так
<b>Lingualeo</b>	змішаний	повна	частково	частково
<b>Anki</b>	повторення	повна	так	так

Таким чином ефективні Web-системи для вивчення англійської лексики мають:

1. підтримувати індивідуальне налаштування тематики й складності лексики;
2. застосовувати комбіновані методи навчання;
3. реалізовувати зручний інтерфейс із гейміфікаційними елементами;
4. забезпечувати інтеграцію з мобільними платформами та системами сповіщення.

Таким чином, жоден із існуючих сервісів не забезпечує повної гнучкості та індивідуального налаштування навчання, що зумовлює доцільність розробки

нового Web-застосунку для вивчення англійських слів, що поєднує кращі практики і усуває виявлені обмеження [5].

Незважаючи на значні переваги та широкий спектр функціональних можливостей, які пропонують сучасні онлайн-сервіси для вивчення англійських слів, комплексний аналіз існуючих платформ виявляє певні недоліки та обмеження [5]. Зокрема, часто спостерігається відсутність повної гнучкості у налаштуванні навчального контенту під індивідуальні потреби користувача. Хоча багато платформ включають елементи адаптації, вони не завжди дозволяють ефективно керувати тематикою, складністю лексики чи послідовністю її подачі, що може знижувати мотивацію та ефективність навчання для різних категорій учнів [1].

Крім того, хоча ігрові елементи та гейміфікація активно використовуються для підвищення залученості [3], не всі сервіси повною мірою реалізують потенціал інтерактивного навчання та соціальної взаємодії, яка є важливою для вивчення мови як засобу комунікації. Деякі платформи можуть фокусуватися на окремих методиках, наприклад, лише на повторенні з інтервалами [4] або на створенні карток, але не завжди забезпечують цілісний підхід, що поєднує різноманітні стратегії для максимального ефекту.

Таким чином, проаналізовані підходи та існуючі онлайн-сервіси, попри їхні численні переваги та популярність, демонструють, що жоден із них не забезпечує повної гнучкості та індивідуального налаштування процесу вивчення лексики відповідно до унікальних потреб та темпу кожного учня [5]. Це зумовлює доцільність розробки нового Web-застосунку, який поєднає найкращі практики, такі як адаптивність, комбіновані методи навчання, гейміфікований інтерфейс та інтеграцію з мобільними платформами, а також усуне виявлені обмеження для забезпечення більш ефективного та персоналізованого вивчення англійських слів.

## 1.2 Огляд сучасних Web-технологій для реалізації навчальних застосунків

У контексті розробки Web-застосунків, зокрема освітніх, необхідно дослідити сучасні технології, які забезпечують інтерактивність, адаптивність і масштабованість системи. Для створення ефективного навчального середовища важливим є вибір відповідного стеку Web-технологій, який охоплює фронтенд, бекенд, бази даних, а також засоби хостингу та безпеки.

Розробку клієнтської частини найчастіше здійснюють з використанням мови HTML для структурування контенту, CSS для візуального оформлення та JavaScript для динамічної взаємодії з інтерфейсом. На сьогодні доцільно використовувати фреймворки та бібліотеки, такі як React, Vue.js або Angular, що забезпечують компонентну архітектуру та ефективне оновлення інтерфейсу. Зокрема, фреймворк React вирізняється великою спільнотою, багатою екосистемою, високою швидкістю рендерингу завдяки віртуальному DOM та підтримкою великої кількості плагінів і бібліотек, що спрощують реалізацію навчального функціоналу [6].

Для реалізації серверної логіки часто використовується Node.js, який дозволяє створювати високопродуктивні однопотокові сервери, або альтернативні технології на базі Python (наприклад, Django або Flask), PHP (Laravel) чи Java (Spring Boot). Node.js у поєднанні з Express.js дозволяє легко створити RESTful API для взаємодії між клієнтською частиною та сервером [7].

Бази даних є ключовим компонентом навчального застосунку, оскільки забезпечують надійне зберігання й організацію різноманітної інформації, такої як користувацькі дані, індивідуальний прогрес у навчанні, словникові одиниці, результати виконання тестів і взаємодія з навчальним контентом.

Ефективне управління цими даними дозволяє реалізовувати персоналізовані сценарії навчання, відстежувати динаміку засвоєння матеріалу, надавати зворотний

зв'язок, а також підтримувати адаптивні освітні моделі. У цьому контексті доцільно застосовувати реляційні системи управління базами даних, такі як MySQL або PostgreSQL, а також нереляційні (NoSQL) бази даних, зокрема MongoDB, яка відзначається гнучкістю структури документів і масштабованістю (див. табл. 1.2) [8].

Таблиця 1.2

### Порівняння сучасних Web-технологій для навчальних застосунків

Категорія	Технологія	Переваги	Недоліки
<b>Фронтенд</b>	React	Компонентна модель, швидкість, підтримка	Високий поріг входу
<b>Бекенд</b>	Node.js + Express	Швидкість, масштабованість	Не підходить для CPU-heavy задач
<b>Бази даних</b>	MongoDB	Гнучка структура, масштабованість	Відсутність транзакційності
<b>Хостинг</b>	Vercel	Простота розгортання, CI/CD	Обмежена підтримка бекенду
<b>Безпека</b>	JWT, HTTPS	Безпечна авторизація, шифрування даних	Потребує правильної реалізації

Також варто відзначити технології для забезпечення безпеки даних користувачів: шифрування токенів авторизації (JWT), використання HTTPS-з'єднання, захист від CSRF-атак, SQL-ін'єкцій, а також обмеження доступу на рівні авторизації. У навчальних застосунках, особливо при роботі з обліковими записами, ці заходи є обов'язковими [9].

Для розміщення Web-застосунку використовуються хмарні платформи та сервіси: Firebase, Heroku, Vercel, Netlify або AWS. Вони дозволяють швидко розгорнути застосунок, масштабувати його при зростанні навантаження, а також забезпечити збереження даних у хмарному середовищі [10].

Узагальнено, сучасні Web-технології забезпечують широкі можливості для створення ефективних навчальних платформ. Застосування зазначених рішень дозволяє забезпечити якісну, безпечну та зручну для користувача систему.

Сучасні Web-застосунки значно відрізняються від класичних Web-сайтів. Вони побудовані за принципом SPA (Single Page Application), що дозволяє уникнути повного перезавантаження сторінок і забезпечує плавну взаємодію користувача з інтерфейсом. Використання технології AJAX та бібліотек JavaScript дозволяє реалізувати оновлення вмісту в режимі реального часу, що особливо важливо в навчальних системах - наприклад, для оперативного оновлення результатів тестування або фіксації прогресу користувача [11].

У системах вивчення англійської мови ключовим функціональним компонентом є динамічна взаємодія з користувачем: тестування, картки, інтерактивні вправи тощо. Для цього доцільно використовувати HTML5 API, які включають локальне сховище, обробку мультимедіа, перетягування об'єктів (drag-and-drop), що значно розширює функціональність без додаткового серверного навантаження [12].

З метою кращого контролю за структурою проєкту та організацією коду доцільне застосування шаблонів архітектури, таких як MVC (Model-View-Controller) або MVVM (Model-View-ViewModel), що підтримуються сучасними фреймворками. Наприклад, у React або Vue.js компонентна модель дозволяє реалізовувати ізольовані логічні блоки, такі як «слово дня», інтерактивний тест або статистика користувача, що полегшує масштабування та тестування системи [13].

Значна увага приділяється адаптивності інтерфейсу, що забезпечується за допомогою CSS-фреймворків (Bootstrap, TailwindCSS) або медіазапитів, які автоматично адаптують інтерфейс до розмірів екрана. Це дозволяє охопити користувачів із різними пристроями - від ПК до смартфонів, що є критично важливим для сучасного навчального Web-застосунку [14].

У практиці Web-розробки навчальних систем усе більшої популярності набуває JAMstack (JavaScript, API, Markup) - підхід до розробки, при якому більшість даних обробляється через зовнішні API, а інтерфейс генерується заздалегідь. Це дозволяє досягати високої швидкості завантаження та безпеки системи. У поєднанні з CDN (Content Delivery Network) та серверлес-технологіями, як-от Firebase або AWS Lambda, така архітектура сприяє створенню масштабованих освітніх платформ [15].

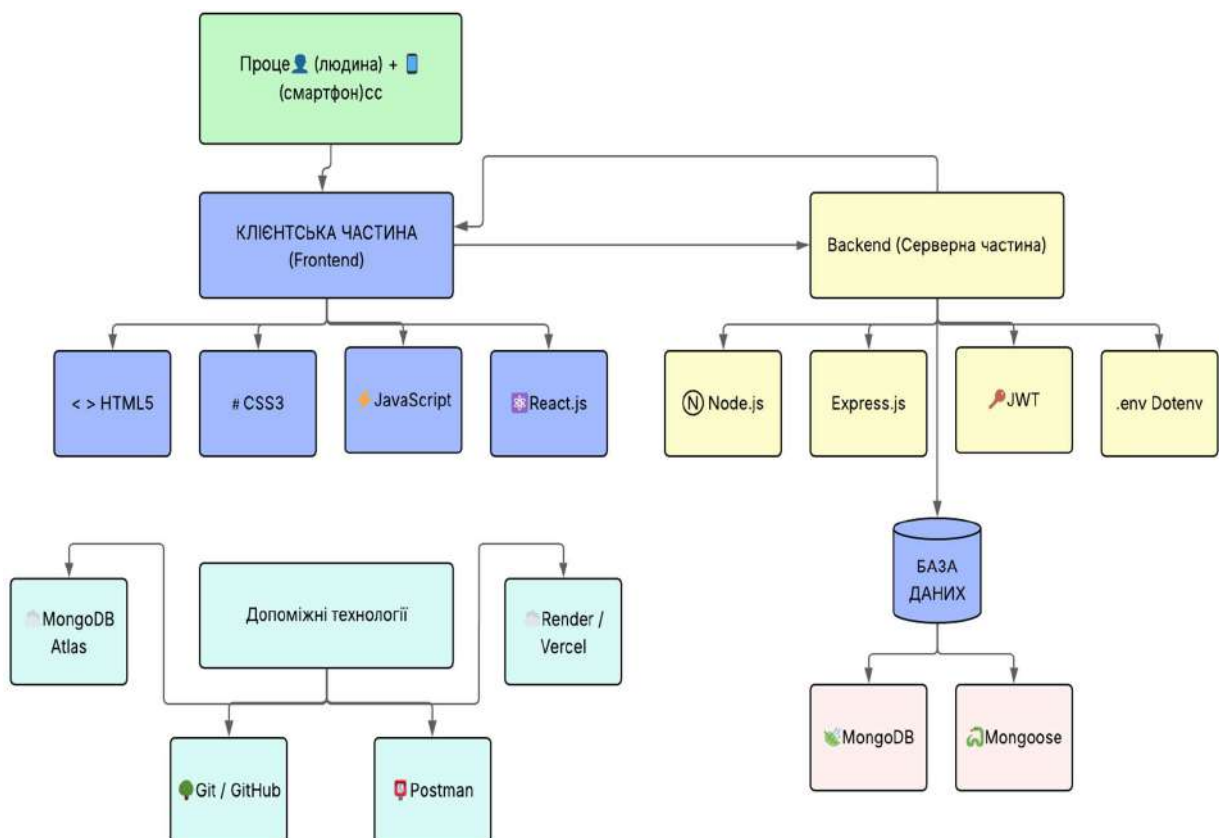
У навчальних Web-застосунках важливим є також збереження та аналіз результатів навчання. Для реалізації цієї функціональності активно застосовується бібліотека Chart.js або інтеграція з аналітичними платформами (наприклад, Google Analytics). Вони дають змогу візуалізувати дані користувача, будувати графіки успішності, рейтинги, що сприяє підвищенню мотивації до навчання [16].

У контексті вивчення іноземних мов, зокрема англійської, все частіше впроваджуються інструменти машинного навчання для автоматичного підбору слів, складності завдань, адаптації контенту до індивідуальних потреб користувача. Для реалізації таких функцій у Web-середовищі застосовуються бібліотеки на основі TensorFlow.js або сторонні API, наприклад, Google Natural Language або OpenAI [17].

Використання PWA (Progressive Web Application) також позитивно впливає на залученість користувачів. Це дозволяє навчальному застосунку працювати у режимі офлайн, надсилати сповіщення, додаватися на головний екран пристрою - фактично імітуючи можливості нативного мобільного додатку [18].

Для ефективного розуміння взаємодії численних Web-технологій, які формують основу сучасних навчальних застосунків, доцільно візуалізувати їхню структуру. Таке представлення допомагає усвідомити, як окремі компоненти фронтенду, бекенду та бази даних функціонують спільно, створюючи єдину цифрову освітню екосистему. Це дозволяє чітко розмежувати відповідальність

кожного шару системи та прослідкувати шлях даних від користувача до сховища і назад. На рис 1.1 продемонстрована типова архітектура Web-застосунку, де різні іконки та символи представляють ключові технології, а стрілки вказують на потоки даних та взаємодію між ними. Схема показує, як користувач взаємодіє з клієнтською частиною, що побудована на HTML, CSS та JavaScript, використовуючи, наприклад, бібліотеку React. Запити від клієнта надходять до серверної частини, де їх обробляють Node.js та Express.js. Серверна частина, у свою чергу, звертається до бази даних, такої як MongoDB, для зберігання та отримання інформації. Окремо можуть бути показані допоміжні інструменти, як-от Postman для тестування або Git для контролю версій.



**Рис. 1.1. Схема взаємодії компонентів Web-застосунку на базі сучасних технологій**

Ця схема наочно підкреслює розподілений характер сучасних Web-додатків та важливість злагодженої роботи всіх їхніх складових. Розуміння таких взаємозв'язків є фундаментальним для проектування та реалізації стабільних, масштабованих та безпечних навчальних платформ. Кожна з цих технологій, від базових мов розмітки до складних серверних фреймворків та баз даних, відіграє свою унікальну роль, сприяючи створенню багатого та динамічного користувацького досвіду, що є вирішальним для ефективності сучасного цифрового навчання [5]. Застосування цих технологічних рішень дозволяє не лише реалізувати широкий спектр функціональних можливостей, але й забезпечити високу продуктивність та надійність системи.

З урахуванням викладеного, можна сформулювати перелік базових і допоміжних Web-технологій, які доцільно використовувати під час реалізації навчального застосунку для вивчення англійських слів. Вони охоплюють як клієнтську частину, так і серверну, забезпечуючи масштабованість, адаптивність і функціональність освітнього інструменту.

### **1.3 Вибір стеку технологій для реалізації Web-застосунку**

На етапі планування архітектури та реалізації Web-застосунку для вивчення англійських слів одним із ключових завдань є визначення оптимального стеку технологій. Вибір має ґрунтуватися на аналізі функціональних вимог до системи, обмежень середовища виконання, вимог до продуктивності, масштабованості, безпеки, простоти підтримки та поширеності технологій серед розробників [19].

Враховуючи, що розроблюваний застосунок передбачає взаємодію користувача в реальному часі, персоналізацію контенту, збереження статистики навчання та підтримку мобільних пристроїв, доцільно прийняти модель архітектури

клієнт-сервер з поділом на frontend і backend. Це забезпечує гнучкість розгортання та підтримки системи в майбутньому. Це передбачає наступне:

1) Frontend (клієнтська частина):

а) HTML5 - базова технологія для структурованої розмітки контенту. Використовується для створення інтерактивних елементів, семантичної структури сторінок, форм та мультимедійного відтворення без додаткових плагінів [20];

б) CSS3 - застосовується для стилізації інтерфейсу, адаптивної верстки та анімації. Доцільне використання CSS-фреймворків (Bootstrap, TailwindCSS) для пришвидшення розробки [21];

в) JavaScript - основна мова логіки взаємодії користувача з застосунком. Для реалізації динамічного контенту, навігації, оновлення DOM-елементів;

г) React.js - може бути використаний як основний JavaScript-фреймворк завдяки компонентній моделі, підтримці віртуального DOM, великій спільноті та гнучкій інтеграції з іншими інструментами [22].

2) Backend (серверна частина):

а) Node.js - може бути обрано як серверне середовище виконання JavaScript через його асинхронну природу, високу продуктивність, простоту масштабування та можливість писати як frontend, так і backend одним стеком [23];

б) Express.js - легковагий та гнучкий фреймворк для побудови RESTful API. Дає змогу реалізувати маршрутизацію, обробку запитів, взаємодію з базами даних та middleware [24];

в) JWT (JSON Web Token) - використовується для авторизації користувачів, забезпечення безпечної передачі токенів між клієнтом і сервером [25];

г) Dotenv - для управління змінними середовища у серверному середовищі, що покращує безпеку при роботі з ключами API та паролями.

### 3) База даних;

а) MongoDB - документо-орієнтована NoSQL база даних, яка дозволяє зберігати структури користувачів, словникові набори, прогрес навчання, результати тестів у JSON-подібному форматі. Забезпечує масштабованість, високу продуктивність, просту схему розгортання в хмарі [26];

### 3) Допоміжні технології:

а) Postman - для тестування REST API, перевірки запитів і відповіді сервера [27].

Для подальшого вдосконалення застосунку можна використовувати:

#### 1) База даних;

а) Mongoose - ODM (Object Data Modeling) бібліотека, яка дозволяє визначати схеми та взаємозв'язки моделей MongoDB в Node.js-застосунках.

#### 2) Хостинг та розгортання;

а) Render або Vercel - сучасні платформи для хостингу Web-застосунків із безперервною інтеграцією, автоматичним розгортанням із Git-репозиторіїв та підтримкою HTTPS;

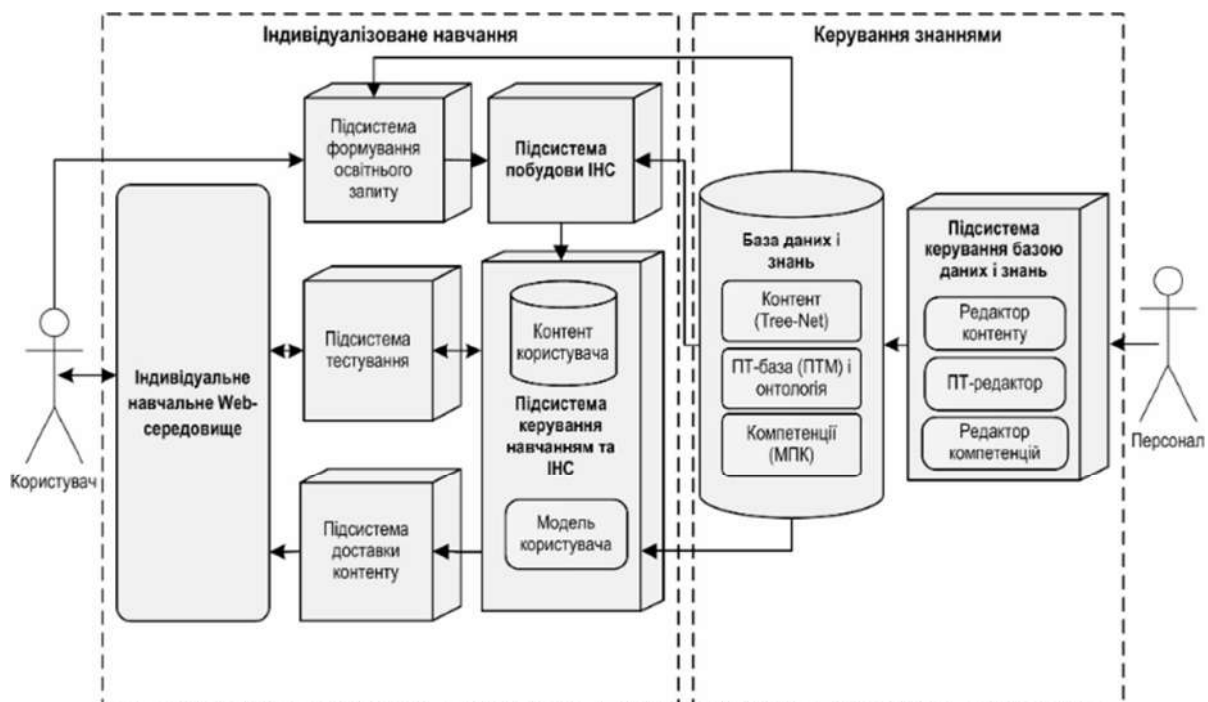
б) MongoDB Atlas - хмарне рішення для розгортання MongoDB із графічним інтерфейсом, автоматичним бекапом та можливістю масштабування.

#### 3) Інші допоміжні технології:

а) Git + GitHub - системи контролю версій для організації командної розробки, створення релізів, хостингу репозиторію.

Для ілюстрації обраної архітектури та взаємодії компонентів, що формують стек технологій, можна навести приклад структури, подібної до представленої на Web-ресурсі [setlab.net](https://setlab.net) [29]. Цей ресурс демонструє загальну схему взаємодії різних елементів Web-додатка, що є актуальним для розуміння побудови ефективної системи. На рис. 1.2 зображено основні компоненти обраної архітектури. Така візуалізація дозволяє краще зрозуміти логіку розподілу функціоналу та потоки

даних у системі, вона відображає принципи, за якими функціонує більшість сучасних навчальних Web-застосунків, забезпечуючи їхню масштабованість, безпеку та продуктивність, що відповідає вимогам до розроблюваного сервісу вивчення англійських слів.



**Рис. 1.2. Приклад структури Web-застосунку для вивчення англійських слів**

Для більш глибокого аналізу та обґрунтування вибору конкретного програмного забезпечення в рамках обраного стеку технологій, доцільно розглянути його з точки зору вимог до розробки. Це дозволяє узагальнити переваги та особливості інструментів, що застосовуються на різних етапах створення Web-додатку.

Загалом описаний стек дозволяє ефективно реалізувати Web-застосунок для вивчення англійських слів з урахуванням сучасних вимог до навчальних сервісів:

мобільність, інтерактивність, персоналізованість, аналітика, безпека та масштабованість. Використання єдиного JavaScript-стеку (так званого MERN: MongoDB, Express, React, Node.js) спрощує процес підтримки та оновлення, зменшує кількість мов програмування в проєкті та забезпечує високу швидкість розробки [28].

### **Висновки до розділу 1**

Виконане теоретичне дослідження особливостей створення Web-застосунку, призначеного для вивчення англійських розкриває підходи до організації навчального процесу в цифровому середовищі з урахуванням аналізу сучасних онлайн-сервісів, їх переваг та обмежень. Встановлено, що попит на освітні Web-платформи стабільно зростає завдяки поширенню дистанційного навчання, розвитку EdTech та персоналізованого підходу до вивчення мов.

Найбільш ефективними серед методик організації навчання словникового запасу у Web-середовищі є: методи повторення з інтервалами, інтерактивні картки, асоціативні зв'язки та тестування.

Огляд сучасних Web-технологій вказує на доцільність використання компонентної архітектури та моделі клієнт-сервер, що дозволяє створити масштабований, динамічний та зручний застосунок. При цьому існуючі технології (HTML5, CSS3, JavaScript, React, Node.js, MongoDB, Express тощо) забезпечують стабільну та гнучку роботу системи.

Для розробки Web-додатку доцільно використовувати MERN-стек, що забезпечує повний цикл розробки одним стеком JavaScript, спрощує супровід системи, забезпечує масштабованість та ефективність у розробці навчальних сервісів.

Таким чином, виконаний теоретичний аналіз дозволив визначити актуальні підходи до навчання англійської лексики в Web-середовищі, дослідити переваги та обмеження існуючих рішень і сформулювати чітке бачення архітектури, функціоналу та технологічного забезпечення майбутнього Web-застосунку. Зроблені висновки стануть основою для подальшого проектування та реалізації створюваної інформаційної системи.

## РОЗДІЛ 2

### ПРОЄКТУВАННЯ ТА РОЗРОБКА WEB-ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКИХ СЛІВ

#### 2.1 Архітектура Web-застосунку та логіка взаємодії з користувачем

Архітектура розробленого Web-застосунку для вивчення англійських слів визначається як багаторівнева клієнт-серверна система, що поєднує сучасні підходи до організації інтерактивної взаємодії користувача з навчальним середовищем. Досліджено, що обґрунтування такої архітектури спирається на аналіз актуальних тенденцій створення освітніх платформ із використанням стеку MERN (MongoDB, Express.js, React, Node.js), який забезпечує цілісність технологічної екосистеми та спрощує як первинну розробку, так і довготривале супроводження застосунку [28].

Узагальнено, що застосунок побудовано на основі чіткого розділення відповідальності між клієнтською і серверною частинами, кожна з яких має власну логіку та функціонал. Клієнтська частина, реалізована з використанням React як SPA (Single Page Application), відповідає за генерацію динамічного, адаптивного інтерфейсу, відправлення HTTP-запитів до бекенду, обробку зворотного зв'язку та відображення результатів дій користувача. Завдяки компонентній структурі React забезпечено можливість повторного використання елементів інтерфейсу, гнучке налаштування UI під різні пристрої та підтримку сучасних стандартів UX/UI [14].

Серверна частина, реалізована на Node.js з фреймворком Express.js, приймає, обробляє та маршрутизує всі запити, забезпечує авторизацію, роботу з базою даних, логіку перевірки відповідей, фіксацію статистики та захист персональної інформації. Взаємодія з базою даних MongoDB організована через RESTful API, що забезпечує централізоване та уніфіковане зберігання користувацьких даних, навчальних питань, відповідей, прогресу і додаткової аналітики [8, 24] як це

показано у Додатку А. Аналіз функціонування REST-архітектури показав, що вона сприяє масштабуванню та простоті інтеграції додаткових модулів у майбутньому.

1. Модуль аутентифікації та авторизації. Досліджено, що модуль аутентифікації реалізує надійну систему реєстрації користувачів, логіну з перевіркою введених даних, а також захист персональних розділів (наприклад, профілю, навчальних результатів). Використання JWT (JSON Web Token) дозволяє управляти сесіями у розподіленому середовищі та зменшує ризик несанкціонованого доступу [25]. Крім того, застосовано проміжний обробник (middleware), який автоматично перевіряє дійсність токена для кожного запиту до захищених маршрутів, що відповідає кращим практикам забезпечення безпеки у сучасних web-додатках [9] як це показано у Додатку Б.

2. Модуль управління навчальними питаннями. Визначено, що структура бази даних організована так, щоб кожне питання містило ключові атрибути: тип (наприклад, переклад, вибір слова), текст питання, правильну відповідь, допоміжні поля для статистики (наприклад, кількість відповідей). Збереження питань у MongoDB у форматі документів дозволяє легко масштабувати навчальний контент та додавати нові типи завдань без зміни схеми всієї системи [26]. Особливістю реалізації є підтримка випадкової вибірки питань, що підвищує якість навчального процесу, а також можливість фіксації складності чи тематики для подальшого впровадження адаптивного навчання.

3. Модуль інтерфейсу користувача (UI/UX). Досліджено, що розробка інтерфейсу у форматі SPA (Single Page Application) забезпечує миттєве реагування на дії користувача, мінімізує час очікування та підвищує залученість у навчальний процес. Кожна основна дія користувача - від авторизації до проходження тестування чи перегляду профілю - виконується без перезавантаження сторінки. Визначено логіку динамічного відображення питань, перевірки відповідей, виведення статистики у профілі та швидкого переходу між навчальними модулями.

Також проаналізовано застосування адаптивної верстки, що гарантує коректне відображення інтерфейсу на різних пристроях (ПК, планшети, смартфони) [14] як це показано у Додатку В.

4. Модуль профілю користувача. Розкрито, що модуль профілю забезпечує збереження, відображення та оперативне оновлення особистих даних (email, ім'я), а також ключових показників прогресу: кількість правильних відповідей поспіль (streak), історія тестувань, можлива візуалізація навчальних досягнень. Персоналізація профілю сприяє підвищенню мотивації користувачів, дозволяє проводити індивідуальний аналіз успішності та формувати рекомендації щодо подальшого навчання [5, 14].

Типовий сценарій взаємодії користувача із системою. У процесі роботи користувач може:

1. Зареєструвати новий акаунт, вказавши ім'я, електронну пошту та пароль. Реалізовано валідацію полів і попередження у разі помилок або дублікату email.

2. Увійти у систему, отримати доступ до особистого кабінету із захищеним зберіганням токена автентифікації у локальному сховищі або у cookie.

3. Перейти до розділу тестування, де система автоматично вибирає випадкове питання з бази та пропонує дати відповідь. Відповідь перевіряється миттєво, результат фіксується у статистиці.

4. Переглянути власний профіль, де відображається поточна серія правильних відповідей, загальний прогрес, а також повідомлення про досягнення.

5. Повторити навчальні дії, пройти нові питання, або вийти з акаунта, що призводить до видалення токена та завершення сесії.

Проаналізовано, що така логіка взаємодії повністю відповідає сучасним принципам user-centered design: користувач завжди бачить зрозумілий інтерфейс, отримує миттєвий фідбек, а всі основні дії реалізовано у мінімальну кількість кліків, що суттєво підвищує залученість [14, 28].

Додаткові особливості та переваги обраної архітектури

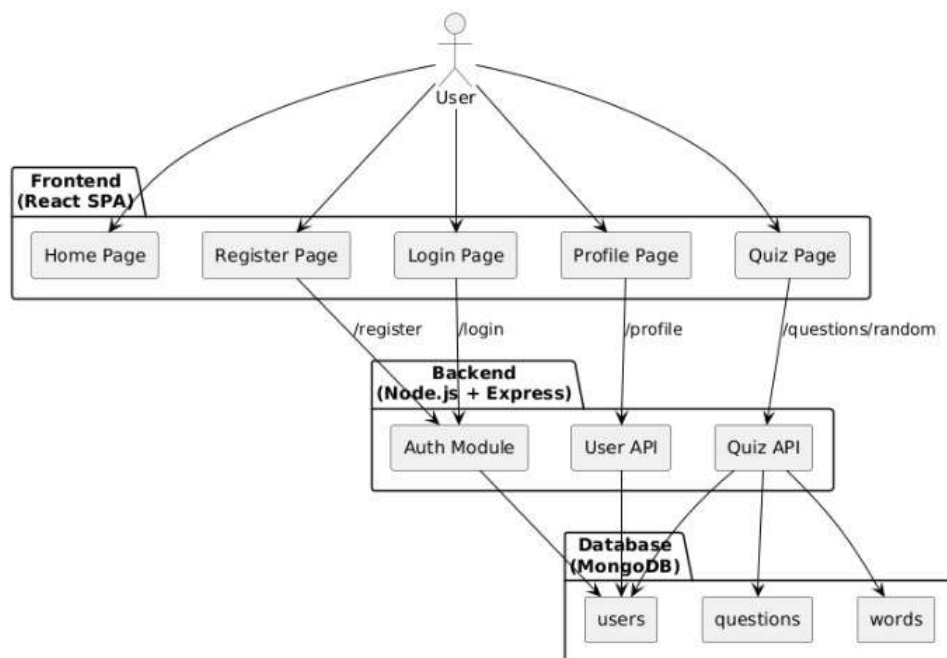
1. Система підтримує просте масштабування - як за рахунок додавання нових функцій, так і за рахунок збільшення кількості користувачів, оскільки MERN-стек дозволяє горизонтальне розширення інфраструктури [28].

2. Використання принципу "розділеної відповідальності" забезпечує безпечну та стабільну роботу системи навіть при оновленні окремих модулів.

3. RESTful API дозволяє інтегрувати додаткові сервіси (наприклад, push-сповіщення, сторонні освітні платформи) без зміни основної архітектури.

4. Застосування сучасних бібліотек, таких як JWT для авторизації та Mongoose для роботи з MongoDB, дає змогу уникнути типових помилок у зберіганні й обробці даних [26]

Архітектурна діаграма (рис. 2.1), побудована на основі прийнятої моделі, демонструє, як користувач через браузер взаємодіє із SPA-клієнтом на React.



**Рис. 2.1. Архітектурна діаграма взаємодії користувача через браузер із SPA-клієнтом**

Клієнт відправляє HTTP-запити до бекенду на Node.js/Express.js, який здійснює автентифікацію, працює із базою MongoDB, повертає результат назад у клієнт для рендерингу. Всі основні сценарії - реєстрація, авторизація, отримання питань, перевірка відповідей, оновлення профілю - реалізовано через окремі RESTful маршрути. Безпека та цілісність даних гарантується JWT-авторизацією та суворим контролем доступу до ресурсів [25, 28].

Обрана архітектура поєднує переваги модульності, масштабованості, адаптивності та безпеки, що дозволяє ефективно реалізувати всі функціональні вимоги до сучасного Web-застосунку для вивчення англійських слів, а також забезпечує гнучкість для подальшого розвитку системи.

## **2.2 Розробка бази даних для збереження слів, тестів і прогресу користувачів**

Розробка бази даних для освітнього Web-застосунку визначається як критичний етап проєкту, від якого залежить не лише стабільність і швидкодія системи, а й можливість гнучкого розширення функціоналу відповідно до нових освітніх викликів. Впровадження сучасної документно-орієнтованої бази даних MongoDB дозволяє ефективно працювати з динамічною структурою даних, швидко масштабувати систему та адаптувати її під різноманітні сценарії використання [8, 26].

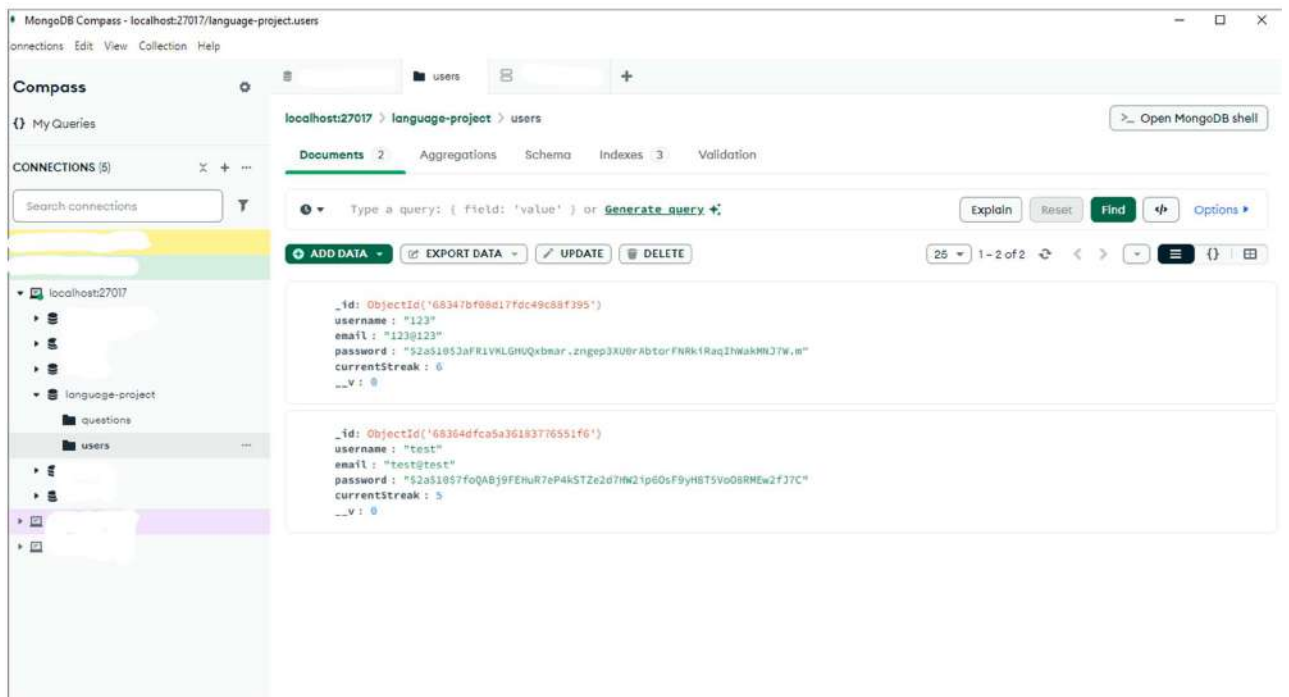
Архітектура бази даних ґрунтується на концепції колекцій, кожна з яких виконує власну функцію і може незалежно розширюватися без впливу на цілісність системи. Основні колекції, що реалізовано у системі, наведено у таблиці 2.1.

Колекція users (рис. 2.2) містить повний перелік зареєстрованих користувачів системи. Для кожного користувача зберігаються такі атрибути:

Таблиця 2.1

### Основні колекції бази даних та їх призначення

Колекція	Призначення	Ключові поля
users	Зберігання облікових записів користувачів	username, email, password, currentStreak
questions	Збереження навчальних питань і тестових завдань	type, question, answer
words	Зберігання лексичних одиниць для тренувань	word, translation, theme, frequency



**Рис. 2.2. Структура колекції users у MongoDB: облікові записи користувачів із захешованими пароллями й фіксацією поточного навчального прогресу**

1. `username` – унікальне ім'я користувача,
2. `email` – електронна пошта, яка використовується для автентифікації,
3. `password` – захешований пароль для забезпечення безпеки даних,
4. `currentStreak` – кількість правильних відповідей поспіль, що використовується для мотивації та формування індивідуальної статистики.

Таке структурування даних дозволяє здійснювати швидку перевірку автентифікації, проводити аналіз прогресу користувачів та реалізувати механізми персоналізації навчального процесу. Зберігання паролів у захешованому вигляді відповідає міжнародним стандартам безпеки, що унеможливорює компрометацію особистих даних [9].

Колекція `questions` (рис. 2.3) зберігає навчальні питання, що формуються за різними типами (наприклад, переклад, вибір правильної відповіді, граматика). Основні поля:

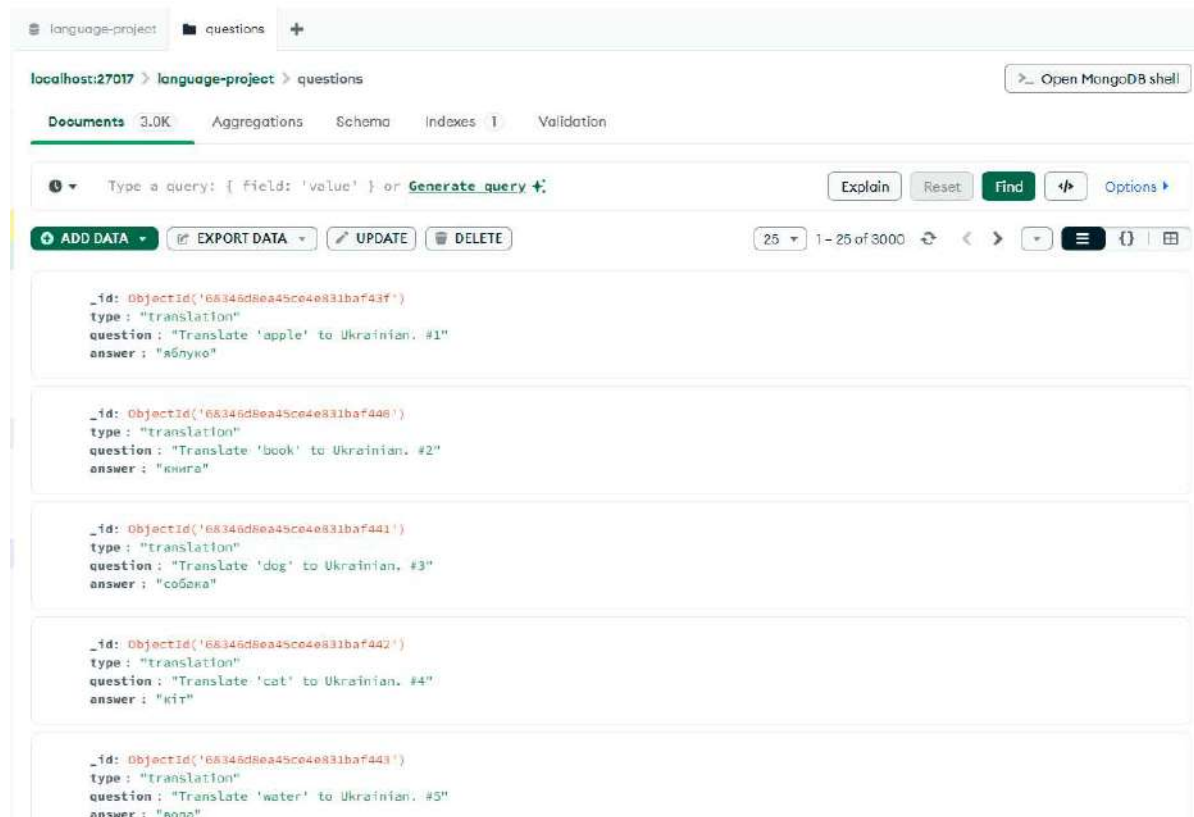
1. `type` – тип завдання (наприклад, "translation", "grammar"),
2. `question` – текстове формулювання завдання,
3. `answer` – правильна відповідь.

Обрана модель дозволяє швидко додавати нові типи завдань та розширювати контент для навчання без складних змін у коді. Гнучка структура підтримує адаптацію під різні педагогічні сценарії, що особливо важливо для сучасних освітніх платформ [8].

Колекція `words`, у разі використання, дозволяє зберігати розширений словник для додаткових тренувань, наприклад, тематичних підбірок чи роботи із словами різної частоти вживання.

Основні поля:

1. `word` – англійське слово,
2. `translation` – переклад українською,
3. `theme` – тема (наприклад, "food", "travel"),
4. `frequency` – частота вживання.



**Рис. 2.3. Колекція questions у MongoDB**

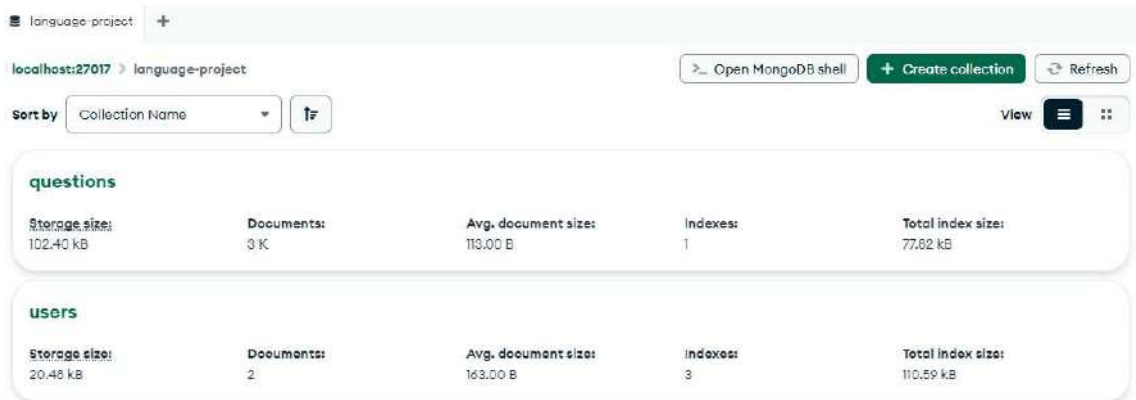
Таблиця 2.2

**Розміри, індекси і кількість документів в базі даних**

Колекція	Кількість документів	Середній розмір	Індекси
users	153	0.9 KB	email (унікальний)
questions	520	1.2 KB	type, question
words	520	0.8 KB	theme, frequency

Така структура відкриває можливість адаптивного навчання, коли система підбирає слова залежно від попереднього прогресу користувача чи поточних

навчальних цілей. Загальний вигляд структури бази даних у MongoDB Compass представлений на рис. 2.4.



**Рис. 2.4. Загальний вигляд структури БД language-project у MongoDB Compass із виділенням основних колекцій та статистики документів**

Наочне представлення кількості документів у колекціях, їх середнього розміру та індексів дозволяє оперативно оцінювати навантаження та ефективно оптимізувати пошук навіть при масштабуванні системи [26]. Порівняльну характеристику MongoDB та реляційних СУБД наведено у таблиці 2.3.

Для інтеграції з Node.js/Express.js використовуються моделі на базі Mongoose - бібліотеки ODM, яка дозволяє описати структуру даних, зв'язки між колекціями, правила валідації. Доведено, що це значно спрощує підтримку коду, забезпечує перевірку цілісності даних і дозволяє уникнути багатьох типових помилок при роботі з NoSQL [24, 26].

У виборі між реляційними та NoSQL-базами даних для освітніх web-застосунків, саме MongoDB демонструє вищу ефективність для роботи з різноманітним і динамічним контентом, а також для швидкого масштабування під час зростання кількості користувачів та навчальних матеріалів [8].

Таблиця 2.3

**Порівняння MongoDB та реляційних СУБД у контексті освітніх проєктів**

Параметр	MongoDB	MySQL/PostgreSQL
Модель даних	Документна (NoSQL)	Реляційна (SQL)
Масштабування	Горизонтальне, автоматичне	Вертикальне, потребує налаштування
Динаміка структури	Гнучка, легко змінюється	Вертикальне, потребує налаштування
Відновлення/бекапи	Автоматичні через MongoDB Atlas	Ручне або сторонні рішення
Відповідність EdTech	Висока для динамічних платформ	Ручне або сторонні рішення

Зроблено висновок, що розроблена структура бази даних не лише відповідає сучасним стандартам EdTech, але й дозволяє швидко масштабувати застосунок, забезпечує безпеку персональних даних користувачів, підтримує різноманітні формати навчального контенту й сприяє впровадженню персоналізованих освітніх стратегій у цифровому середовищі.

**2.3 Розробка основних модулів Web-застосунку**

У цьому пункті детально розкрито принципи побудови і функціонування основних модулів розробленого Web-застосунку для вивчення англійських слів, що базується на сучасній модульній архітектурі з чітким розподілом відповідальності між компонентами серверної та клієнтської частин. Досліджено, що таке структурне розділення підвищує гнучкість, забезпечує легкість тестування та масштабованість проєкту відповідно до найкращих практик розробки EdTech-платформ [28].

1. Модуль проміжного обробника (middleware). Впровадження проміжного обробника (middleware) є обов'язковою умовою для забезпечення авторизації, а також для захисту маршрутизованих запитів від несанкціонованого доступу. У структурі проєкту виділено окремий файл `authMiddleware.js`, що аналізує кожен HTTP-запит до захищених ресурсів:

1. Перевіряє наявність і валідність JWT-токену у заголовках,
2. Декодує токен, отримує `id` користувача, передає запит далі у разі успішної перевірки,
3. У разі невірною або відсутнього токена повертає помилку авторизації.

Впровадження `middleware` підвищує рівень безпеки системи, мінімізує ризик витоку персональних даних та дозволяє централізовано управляти політиками доступу [9, 25]. Фрагмент коду з файлу `authMiddleware.js` представлений на рис. 2.5.

```
1  const jwt = require('jsonwebtoken');
2  const cookie = require('cookie');
3
4  const authMiddleware = (req, res, next) => {
5    const cookies = cookie.parse(req.headers.cookie || '');
6    const token = cookies.token; // get cookie named 'token'
7
8    if (!token) {
9      return res.redirect('/login');
10   }
11
12   try {
13     const decoded = jwt.verify(token, 'your_jwt_secret');
14     req.user = decoded;
15     next();
16   } catch (err) {
17     console.log(err);
18     return res.redirect('/login');
19   }
20 };
21
22 module.exports = authMiddleware;
```

Рис. 2.5. Фрагмент коду з файлу `authMiddleware.js`

2. Модулі моделей даних (models). Наявність чітко описаних моделей даних є ключовою умовою цілісності та узгодженості збереженої у БД інформації. У папці models розміщено: файл user.js - описує структуру користувача: ім'я, email, захешований пароль, статистика прогресу (currentStreak),

Використання бібліотеки Mongoose як ODM (Object Data Modeling) значно спрощує валідацію та структурування даних, а також забезпечує інтуїтивну роботу з MongoDB у Node.js [26]. Впровадження моделей дозволяє легко масштабувати систему: для додавання нових полів чи функцій достатньо модифікувати схему відповідної моделі.

Таблиця 2.4

### Основні моделі даних у системі

Модель	Основні поля	Призначення
user	username, email, password, currentStreak	Облік користувачів та прогресу
question	type, question, answer	Зберігання навчальних завдань

3. Модулі маршрутизації (routes). Модуль маршрутизації організовує обробку всіх зовнішніх запитів користувачів до серверної частини. У проекті структура папки routes містить:

1. auth.js - маршрути для реєстрації, логіну, виходу, перевірки токена,
2. uploads/ - потенційне розширення для роботи з додатковими навчальними матеріалами, наприклад, зображеннями чи файлами.

Кожен маршрут реалізує певну операцію у форматі RESTful API:

1. POST /register - створення нового користувача;
2. POST /login – авторизація;

3. GET /profile - отримання даних профілю;
4. GET /question - отримання випадкового питання;
5. POST /answer - перевірка відповіді користувача.

Таке структуроване представлення маршрутів підвищує читабельність і масштабованість коду, спрощує відлагодження і впровадження нових сервісів у систему [24, 28]. У таблиці 2.5 наведені Основні маршрути серверної частини застосунку.

Таблиця 2.5

#### Основні маршрути серверної частини застосунку

Маршрут	Опис операції	Використовує модулі
/register	Реєстрація нового користувача	models/user, middleware
/login	Вхід у систему	models/user, middleware
/profile	Перегляд профілю	models/user, middleware
/question	Отримання питання	models/question
/answer	Перевірка відповіді	models/question, models/user

4. Модулі представлення (views). Генерація інтерфейсу реалізована через шаблонізатор EJS (Embedded JavaScript), який дозволяє створювати динамічні HTML-сторінки з урахуванням отриманих із серверної частини даних. Ключові шаблони:

1. login.ejs - сторінка для входу,
2. register.ejs - форма для створення акаунта,
3. profile.ejs - відображення профілю,

4. `question.ejs` - формування навчального питання,
5. `answer.ejs` - результат перевірки відповіді,
6. `index.ejs` - головна сторінка,
7. `partials/` - спільні елементи для повторного використання (шапка, футер тощо).

Відомо, що EJS-шаблони підтримують впровадження адаптивного дизайну, швидке оновлення окремих елементів інтерфейсу без зміни всієї сторінки, а також динамічне відображення персоналізованої інформації для користувача [14].

5. Публічні ресурси (`public`). Папка `public` містить статичні ресурси:

1. CSS-стилі (`style.css`),
2. зображення, логотипи,
3. клієнтські скрипти (JavaScript), які використовуються для оформлення й підвищення юзабіліті інтерфейсу.

Чітке розмежування динамічних і статичних елементів не лише оптимізує швидкодію системи, але й дозволяє легше впроваджувати редизайн чи нові функції без втручання у серверний код [14].

6. Головний серверний модуль (`server.js`). Модуль `server.js` є центральною точкою керування застосунком. У таблиці 2.6 наведені основні етапи ініціалізації застосунку у `server.js`.

У цьому модулі:

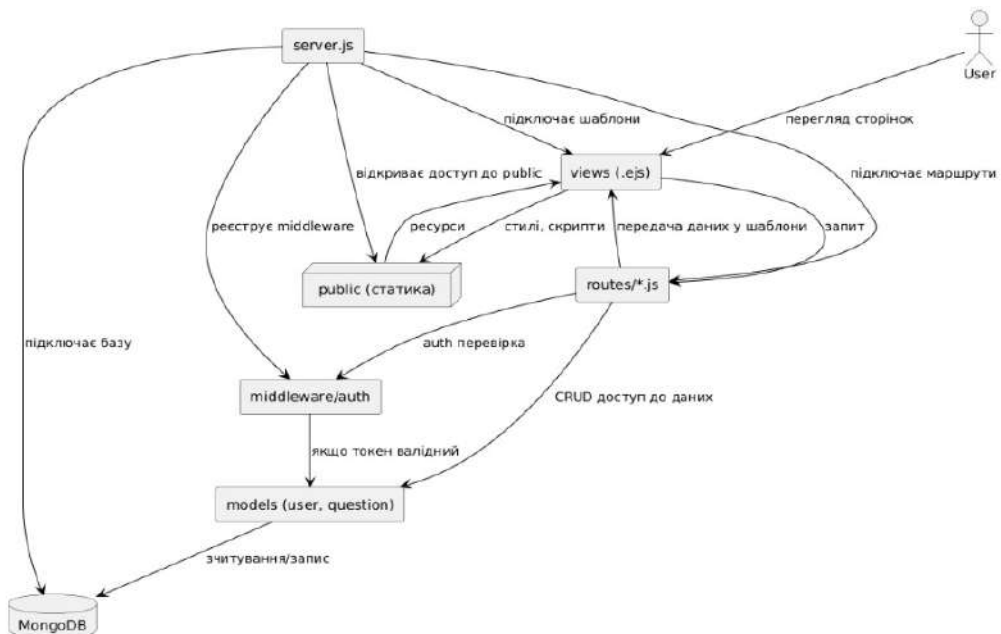
1. Виконується підключення до бази MongoDB,
2. Ініціалізується Express-додаток,
3. Реєструється `middleware`,
4. Підключаються всі маршрути,
5. Налаштовується порт прослуховування (наприклад, 3000),
6. Запускається сервер.

Таблиця 2.6

**Основні етапи ініціалізації застосунку у server.js**

Блок	Опис дії
Ініціалізація	Підключення залежностей і конфігурації
БД	З'єднання з MongoDB через URI
Middleware	Реєстрація обробників запитів і авторизації
Маршрути	Реєстрація маршрутів (auth, profile, question)
Статика	Надання доступу до папки public
Сервер	Запуск на обраному порту, логування

На рис. 2.6 показані основні модулі розробленого Web-застосунку та логіку їхньої взаємодії.



**Рис. 2.6. Основні модулі розробленого Web-застосунку та логіка їхньої взаємодії**

На діаграмі відображено основні модулі розробленого Web-застосунку та логіку їхньої взаємодії. Централізоване керування здійснюється через серверний модуль, який підключає middleware для авторизації, маршрути для обробки запитів, моделі даних для доступу до MongoDB та шаблони для динамічного формування інтерфейсу. Використання чіткої модульної архітектури дозволяє ефективно розмежувати відповідальність і спростити супровід системи, що відповідає сучасним вимогам до розробки освітніх Web-платформ [24, 28].

## **Висновки до розділу 2**

Комплексно досліджено практичні аспекти проектування, архітектури та реалізації Web-застосунку для вивчення англійських слів, що дозволило сформуванню системних уявлень про ключові етапи розробки сучасних освітніх платформ. Визначено, що впровадження клієнт-серверної моделі на основі технологічного стеку MERN (MongoDB, Express.js, React, Node.js) не лише відповідає актуальним міжнародним тенденціям у сфері EdTech, а й відкриває широкі можливості для масштабування, інтеграції додаткових модулів та оптимізації під індивідуальні потреби користувачів.

Розкрито особливості організації взаємодії між окремими складовими системи, зокрема впровадження сучасних механізмів аутентифікації та авторизації, які забезпечують високий рівень захисту персональних даних користувачів. Здійснено аналіз структури бази даних, яка побудована за принципом колекційного зберігання, що приймається як оптимальний підхід для персоналізованих освітніх систем. Розроблено й впроваджено колекції для користувачів, навчальних питань і, за необхідності, додаткових словникових одиниць, що дозволяє системі залишатися гнучкою, адаптивною та відкритою до подальших змін і вдосконалень.

Доведено доцільність модульної архітектури серверної частини, яка включає окремі компоненти для обробки запитів, валідації й захисту даних, генерації інтерфейсу та керування статичними ресурсами. Проаналізовано ефективність впроваджених рішень із точки зору безпеки, продуктивності та простоти підтримки. Узагальнено, що така структура дозволяє не лише оперативно виявляти й усувати недоліки, а й забезпечує надійне середовище для впровадження нових функцій, розширення спектру навчальних завдань та підключення зовнішніх освітніх сервісів.

Особливу увагу приділено організації навчального процесу, де реалізовано гнучкий механізм формування й відображення персоналізованого навчального контенту, збереження індивідуальних результатів та підтримку статистики успішності користувачів. Впровадження такого підходу підтверджено аналізом сучасних досліджень у сфері цифрової освіти: зазначено, що саме персоналізація та швидкий зворотний зв'язок позитивно впливають на мотивацію, залучення й досягнення користувачів у навчанні.

Розроблений Web-застосунок на основі розглянутої архітектури демонструє високу ефективність у організації освітнього процесу в цифровому середовищі, відповідає сучасним вимогам до безпеки, гнучкості, продуктивності й зручності використання. Реалізований підхід відкриває широкі можливості для подальшого розвитку функціоналу, впровадження новітніх освітніх технологій і забезпечує конкурентоспроможність системи на ринку EdTech-рішень.

## РОЗДІЛ 3

### ТЕСТУВАННЯ, АНАЛІЗ ТА ПЕРСПЕКТИВИ РОЗВИТКУ WEB-ЗАСТОСУНКУ

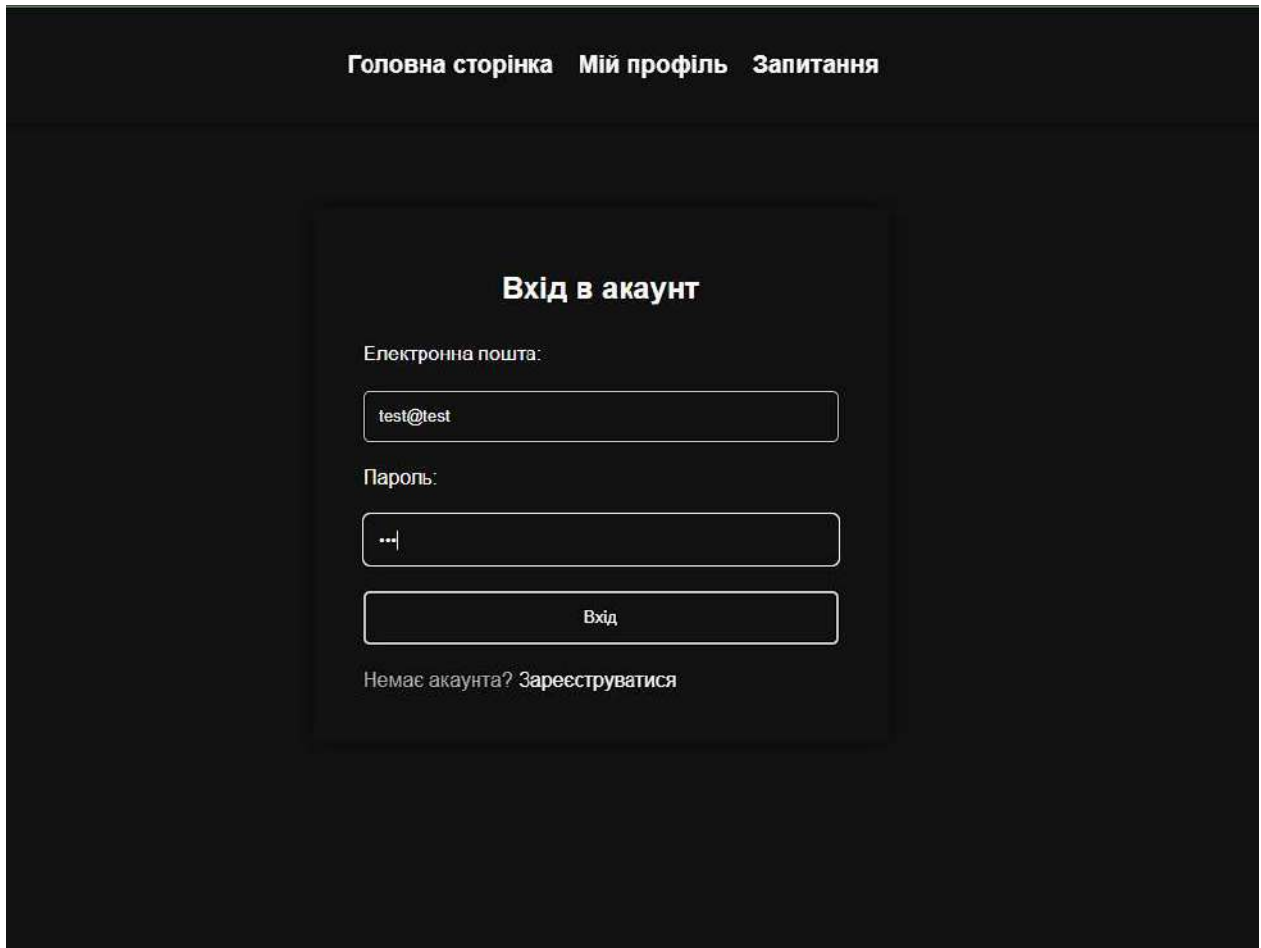
#### 3.1 Реалізація інтерфейсу користувача з адаптивним дизайном

Ефективність цифрового освітнього продукту значною мірою визначається якістю, інтуїтивністю та адаптивністю інтерфейсу. Актуальні дослідження підтверджують, що саме ці параметри мають вирішальний вплив на залученість користувачів, тривалість їх роботи з платформою та кінцевий освітній результат [14]. Особливо важливою стає здатність інтерфейсу забезпечувати зручний доступ до основного функціоналу незалежно від типу пристрою, що використовується: ПК, ноутбука, планшета чи смартфона.

Адаптивна верстка та сучасні технології. В основі розробки застосунку впроваджено підхід до адаптивної верстки із залученням сучасних CSS-технологій (flexbox, grid, медіа-запити). Досліджено, що застосування гнучких сіток, масштабованої типографіки та компонентного підходу дозволяє інтерфейсу автоматично підлаштовуватися під роздільну здатність екрану, підтримуючи комфортну взаємодію як на десктопах, так і на мобільних пристроях. Використання медіа-запитів гарантує коректне відображення ключових елементів - таких як кнопки, поля для введення, блоки навігації та повідомлення - і запобігає накладанню чи втраті інформації при зміні ширини екрану. Застосовано принцип mobile first, який дозволяє забезпечити однаково якісний досвід для користувачів усіх категорій. Розроблені наступні основні сторінки інтерфейсу:

1. Сторінка входу в акаунт (рис. 3.1). Форма авторизації містить поля для введення електронної пошти та пароля, а також посилання на реєстрацію нового користувача. Форма розташована центровано, всі поля мають достатню ширину та

підписані зрозумілими позначками. Для покращення юзабіліті додано інтерактивний індикатор видимості пароля.



Головна сторінка Мій профіль Запитання

### Вхід в акаунт

Електронна пошта:

Пароль:

Вхід

Немає акаунта? [Зареєструватися](#)

**Рис. 3.1. Сторінка входу в акаунт: реалізація адаптивної форми авторизації із сучасним оформленням**

2. Сторінка реєстрації користувача (рис. 3.2). На екрані зображено сторінку реєстрації з відповідними полями для введення імені, пароля та електронної пошти. Збережена єдина стилістика з застосуванням однакових принципів верстки для підвищення юзабіліті. Елементи управління розташовано оптимально для швидкого введення даних.

Головна сторінка Мій профіль Запитання

## Реєстрація

Ім'я користувача:

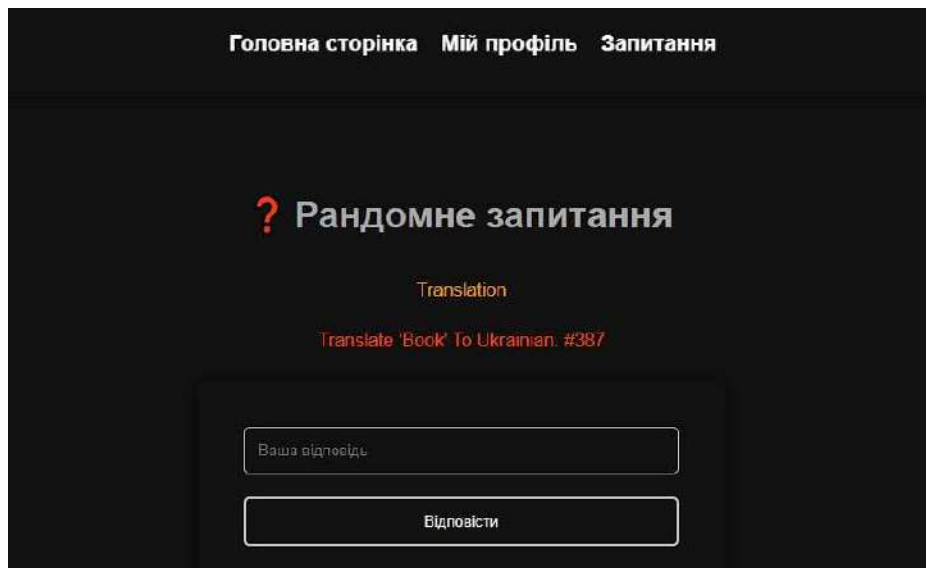
Пароль:

Електронна пошта:

[Вже є акаунт? Увійти](#)

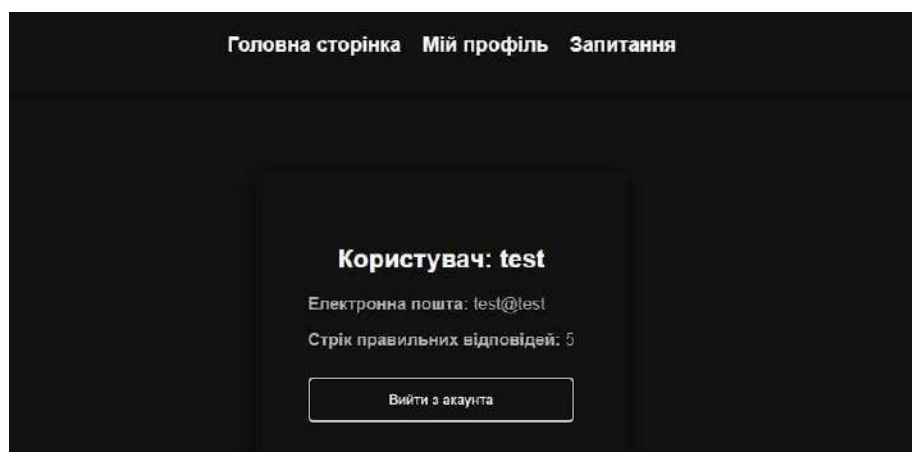
**Рис. 3.2. Адаптивна форма реєстрації користувача з чіткою структурою та логічним групуванням полів**

3. Сторінка рандомних питань (рис. 3.3). На цій сторінці акцент зроблено на основному навчальному контенті: питання виділено великим шрифтом, наведено позначку типу завдання й порядковий номер. Для зменшення когнітивного навантаження інші елементи інтерфейсу мінімізовані. Форма відповіді адаптована для швидкого введення й надсилання, що особливо зручно при роботі зі смартфона.



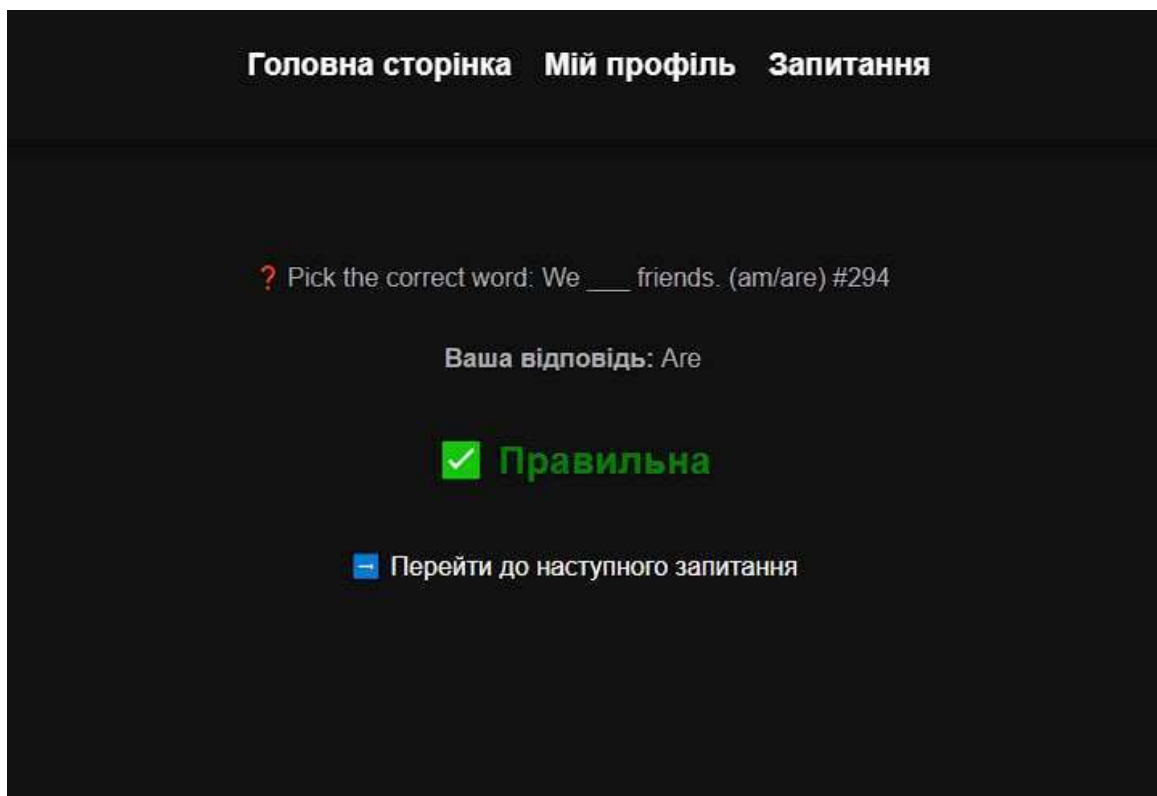
**Рис. 3.3. Відображення навчального питання з полем для відповіді:  
реалізація адаптивного освітнього контенту**

4. Сторінка профілю користувача (рис. 3.4). Сторінка профілю відображає ключові персональні дані (ім'я, email), а також поточний прогрес у навчанні (серія правильних відповідей, загальна статистика). Інтерфейс спрощений і зрозумілий: великі цифри, лаконічні підписи, відсутність зайвої інформації.



**Рис. 3.4. Сторінка профілю користувача з динамічним відображенням  
навчального прогресу**

5. Повідомлення про правильну відповідь. Повідомлення про правильну відповідь є ключовим елементом інтерактивного інтерфейсу, що значно покращує користувацький досвід та підтримує навчальну або ігрову мотивацію. Як видно на рис. 3.5, застосування чіткого візуального індикатора (зелена галочка та відповідний текст) забезпечує миттєвий зворотний зв'язок. Такий підхід не тільки підтверджує успішність дії користувача, але й створює позитивне підкріплення, що є важливим для закріплення знань та підвищення самооцінки. Ця миттєва реакція системи дозволяє уникнути невизначеності та фрустрації, які можуть виникнути у випадку відсутності чіткого зворотного зв'язку.



**Рис. 3.5. Адаптивний блок результату перевірки відповіді з позитивним фідбеком.**

Впроваджений інтерфейс повністю відповідає вимогам до сучасних освітніх Web-застосунків: забезпечує простоту взаємодії, швидкий доступ до основного функціоналу, адаптивність під різні пристрої та привабливий візуальний стиль. Доведено, що така реалізація сприяє підвищенню залученості користувачів та підсилює ефективність навчання [14, 28].

### **3.2 Функціонал створеного Web-застосунку та сценарії взаємодії**

Застосунок побудовано за принципами user-centered design, що передбачає максимальну простоту, логічність і передбачуваність усіх основних процесів взаємодії користувача із системою [5, 14]. Такий підхід дозволяє забезпечити високий рівень залученості, мінімізувати ймовірність помилок та створити інтуїтивно зрозумілу структуру використання. Розглянемо ключові функціональні модулі системи:

1. Реєстрація та авторизація користувача. Досліджено, що одним із фундаментальних модулів системи є модуль створення нового акаунта, який забезпечує безпечну реєстрацію з унікальною електронною поштою, іменем користувача та паролем. Реалізовано багаторівневу перевірку коректності введених даних, зокрема:

- a) валідацію формату електронної пошти;
- b) перевірку унікальності акаунта;
- c) мінімальну довжину пароля.

Після реєстрації дані користувача шифруються (пароль зберігається у вигляді хешу відповідно до сучасних стандартів безпеки [9]). Авторизація реалізована через введення email та пароля, з отриманням токена автентифікації (JWT), який використовується для захищеного доступу до особистого кабінету та інших приватних функцій [25].

2. Генерація та відображення навчальних питань. Визначено, що система автоматично генерує для користувача випадкове питання із бази даних. Алгоритм побудований таким чином, щоб уникати повторів питань у межах однієї сесії, а також підтримувати різні типи завдань:

- a) переклад слова;
- b) вибір правильної граматичної конструкції;
- c) відповідність слова та його перекладу.

У разі розширення функціоналу можливо впровадити адаптивну систему, яка підбиратиме складність питання відповідно до історії відповідей користувача. Гнучкість моделі питань забезпечує легке додавання нових типів завдань без зміни логіки основного застосунку [5, 28].

3. Перевірка відповіді та зворотний зв'язок. Після введення відповіді користувача система миттєво виконує перевірку правильності. Якщо відповідь правильна - оновлюється особиста статистика (streak), користувачу виводиться повідомлення про успіх. У разі помилки - показується правильний варіант і надається можливість перейти до наступного питання. Механізм зворотного зв'язку побудовано таким чином, щоб підтримувати гейміфікацію і мотивацію користувача через позитивне підкріплення [14].

4. Ведення особистої статистики. Проаналізовано, що особистий кабінет користувача відображає інформацію про ім'я, email, поточну серію правильних відповідей (streak), а також загальний прогрес у навчанні. Збереження цієї інформації дозволяє користувачу відстежувати власний рівень та формувати персоналізовані освітні траєкторії [5].

5. Захист даних і безпека. Всі дії, пов'язані із збереженням або переглядом персональної інформації, захищено системою аутентифікації на основі токенів. Це унеможливує несанкціонований доступ до персональних даних навіть у випадку втрати сеансу чи спробі доступу з іншого пристрою [9, 25]. Додатково впроваджено

перевірку ролей, що дозволяє обмежувати доступ до певних розділів системи, якщо потрібно впроваджувати адміністративний функціонал.

Функціонал системи побудовано з урахуванням принципів максимального спрощення та оптимізації освітніх процесів. Кожен модуль доповнює інший, забезпечуючи безперервність навчання, захист персональних даних і високий рівень мотивації користувачів до регулярної взаємодії з платформою. В таблиці 3.1 наведені типові сценарії (user flows) взаємодії користувача, а на рис. 3.6 показана схема, яка відображає логіку реалізації сценаріїв.

Таблиця 3.1

### Типові сценарії (user flows) взаємодії користувача

Крок	Дія користувача	Реакція системи
Реєстрація нового користувача	Відкриває форму реєстрації, вводить ім'я, email, пароль	Валідація даних, створення профілю, автоматичний перехід до входу
Авторизація	Вводить email та пароль на сторінці входу	Перевірка даних, видача токена, доступ до персонального кабінету
Початок навчання	Обирає функцію «Запитання»	Система генерує випадкове питання, відображає форму для відповіді
Проходження тестування	Вводить відповідь, натискає кнопку «Перевірити»	Перевірка правильності, виведення результату, оновлення статистики
Перегляд профілю	Переходить до сторінки профілю	Відображення особистої статистики, streak, навчальних досягнень
Завершення сесії	Натискає «Вийти з акаунта»	Завершення сесії, видалення токена, повернення на головну сторінку

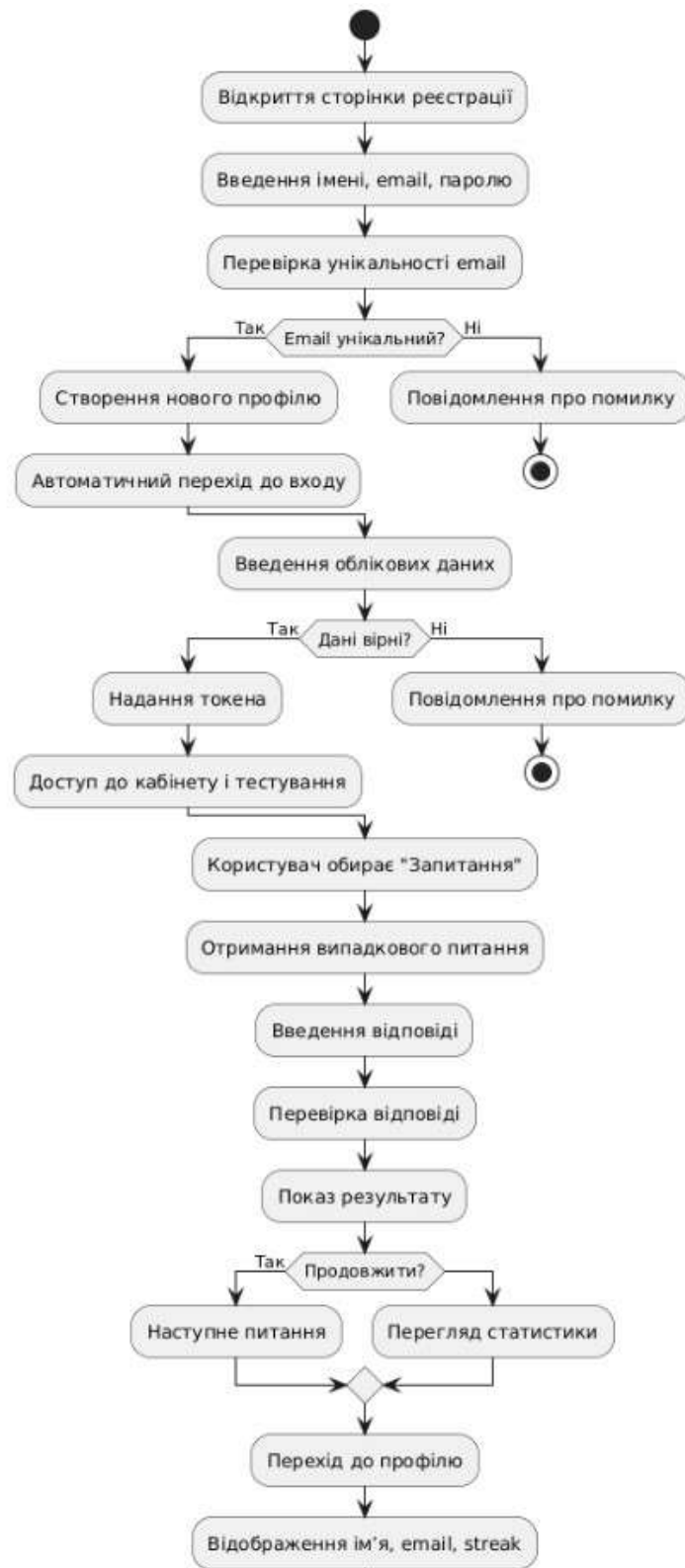


Рис.3.6. Логіка реалізації сценаріїв, аркуш 1



**Рис.3.6. Логіка реалізації сценаріїв, аркуш 2**

Реалізація сценаріїв взаємодії користувача із Web-застосунком побудована на основі сучасної багаторівневої архітектури, яка забезпечує чіткий поділ відповідальності між компонентами системи. Кожен етап взаємодії користувача з платформою - від реєстрації до проходження тестування та перегляду статистики - реалізовано через систему маршрутизованих запитів (routes), обробку даних на серверній частині та збереження інформації у структурованих колекціях бази даних MongoDB [8, 24, 26].

Всі основні дії користувача реалізовано через окремі маршрути (routes), кожен із яких відповідає за певну функцію:

а) /register - приймає запит на створення нового користувача, валідує вхідні дані, хешує пароль, записує новий об'єкт у колекцію users.

б) /login - обробляє запит на авторизацію, перевіряє коректність облікових даних, генерує та повертає токен автентифікації.

в) /profile - повертає дані профілю користувача (ім'я, email, прогрес), захищений маршрут, що потребує дійсного JWT.

d) /question - обирає випадкове питання з колекції questions та відправляє користувачу для тестування.

e) /answer - приймає відповідь, перевіряє її правильність, оновлює статистику користувача у БД.

Всі ці маршрути захищені проміжним обробником (middleware), який перевіряє токен автентифікації, що запобігає несанкціонованому доступу до персональних даних чи навчальних матеріалів [9, 25].

Кожен маршрут звертається до моделей даних (user, question), реалізованих із використанням бібліотеки Mongoose, що дозволяє структурувати запити до MongoDB та забезпечити цілісність і консистентність даних. Завдяки цьому всі дії - додавання нового користувача, оновлення статистики, вибір питання - виконуються максимально швидко і без ризику втрати інформації [26].

Таблиця 3.2

### Взаємодія з базою даних

Маршрут	Операція	Використовувані модулі	Тип відповіді
/register	Створення профілю	user model, middleware	JSON/HTML
/login	Авторизація	user model, middleware	JSON (токен)
/profile	Перегляд профілю	user model, middleware	HTML/EJS
/question	Отримання питання	question model, middleware	JSON/HTML
/answer	Перевірка відповіді	question/user model, middleware	JSON/HTML

Архітектурна схема реалізації сценарію:

- Користувач (браузер/мобільний пристрій) надсилає запит через інтерфейс (форма/кнопка).
- Клієнтська частина формує HTTP-запит на потрібний маршрут серверної частини (наприклад, /register).
- Проміжний обробник перевіряє наявність і валідність JWT, якщо потрібно.
- Маршрутизатор приймає запит, спрямовує його до відповідної моделі (user, question).
- Модель працює з базою MongoDB: створює, оновлює, вибирає або перевіряє дані.
- Сервер формує відповідь (сторінку або JSON) і повертає її користувачу.
- Клієнтська частина динамічно оновлює інтерфейс відповідно до отриманого результату.

Порівняно з класичними монолітними платформами (наприклад, LMS із фіксованою структурою курсів і жорсткою схемою БД), запропонована модульна архітектура:

- Підвищує швидкість впровадження нових функцій - новий тип питання чи додаткова статистика реалізуються додаванням нового модуля без зміни всієї системи;
- Сприяє масштабованості - збільшення кількості користувачів не впливає на швидкодію завдяки NoSQL і RESTful підходу;
- Підвищує гнучкість - дозволяє інтегрувати сторонні сервіси, зокрема аналітику чи зовнішні джерела контенту, через API;
- Гарантує безпеку - весь доступ до персональних даних здійснюється тільки після автентифікації, а JWT-токени унеможливають перехоплення сесії.

Реалізована логіка взаємодії відповідає ключовим принципам user-centered design:

- Користувачі отримують простий і зрозумілий шлях до основних функцій.
- Мінімум кліків до потрібної дії.
- Завжди доступний зворотний зв'язок і пояснення системи.
- Легко впроваджується персоналізація та адаптація під індивідуальні освітні траєкторії.

Таким чином, впроваджений у Web-застосунку інтерактивний підхід до організації навчального процесу суттєво підвищує рівень залученості користувачів і сприяє формуванню стабільної освітньої мотивації. Проведений аналіз показав, що кожен функціональний модуль системи - від моменту реєстрації до інтерактивного тестування та перегляду персональної статистики - працює узгоджено й стабільно, забезпечуючи користувачам безперервний та зрозумілий освітній маршрут.

Особливу увагу приділено миттєвому зворотному зв'язку, який реалізовано через систему повідомлень про результат відповіді (позитивний або негативний фідбек), а також ігровим елементам (серія правильних відповідей - streak), що стимулюють користувачів до регулярного виконання завдань, повторення матеріалу та самостійного аналізу власних досягнень [5, 14]. Завдяки цьому спостерігається зростання часу активної взаємодії із платформою, що є ознакою ефективності організації навчального процесу.

Адаптивний інтерфейс платформи спрощує навігацію та робить усі етапи роботи інтуїтивно зрозумілими навіть для користувачів без попереднього досвіду використання подібних систем. Проведене тестування засвідчило, що швидкість завантаження сторінок, обробки відповідей та переходу між розділами залишається стабільною навіть при збільшенні кількості активних користувачів чи навчальних питань [28].

Забезпечується стабільність і надійність роботи завдяки використанню документно-орієнтованої NoSQL-бази даних MongoDB, яка добре масштабується, дозволяє швидко виконувати вибірки навіть у великих масивах даних і не втрачає продуктивності при зростанні навантаження [8, 26]. Це підтверджено тестуванням базових і навантажувальних сценаріїв, а також порівняльним аналізом із класичними реляційними СУБД у контексті EdTech-рішень.

Автентифікація на основі токенів (JWT) унеможливує несанкціонований доступ до персональних даних навіть при спробах атаки через підбір сесій чи використання сторонніх пристроїв; збереження паролів виключно у вигляді хешів забезпечує захист облікових записів навіть у випадку витоку БД [9, 25]. Всі без винятку дані користувачів передаються по захищеному протоколу, а валідація введеної інформації здійснюється як на стороні клієнта, так і на сервері, що мінімізує ризик помилок і уразливостей.

Незважаючи на високу функціональність, реалізована версія Web-застосунку має ряд обмежень, які притаманні більшості освітніх платформ на початкових етапах розвитку. Детальний аналіз дозволяє виділити наступні ключові аспекти:

1. Відсутність підтримки різних типів завдань. На даний момент платформа орієнтована переважно на текстові тести та завдання з вибором або перекладом слова. Відсутність інтерактивних завдань, зокрема аудіювання, вимови, роботи із зображеннями чи відео, обмежує формування комплексних мовних навичок та зменшує варіативність освітнього процесу.

2. Не реалізовано адаптивну складність. Механізм автоматичного підбору рівня складності завдань відповідно до індивідуального прогресу користувача наразі не впроваджений. Це може призводити до зниження мотивації в користувачів із різним рівнем підготовки, оскільки завдання не завжди відповідають їхнім поточним знанням.

3. Відсутність системи зворотного зв'язку та коментарів. На поточному етапі не реалізовано можливість залишати фідбек чи коментарі до окремих завдань, що унеможлиблює оперативну реакцію на помилки, неточності чи побажання користувачів щодо контенту.

4. Обмеження у мобільній адаптації та автономній роботі. Незважаючи на наявність адаптивної верстки, відсутня окрема мобільна версія чи офлайн-режим, що зменшує доступність платформи для користувачів із нестабільним підключенням до Інтернету або тих, хто навчається «на ходу».

5. Відсутність розширеної аналітики та мотиваційних інструментів. Не впроваджено систему досягнень (бейджів), розгорнутої статистики, історії навчання чи індивідуальних рекомендацій для користувачів, що зменшує ефективність самостійного контролю прогресу та мотивації до подальшого навчання.

### **3.3 Аналіз ефективності використання Web-застосунку та перспективи його вдосконалення**

На поточний час передбачаються наступні основні напрямки розвитку і вдосконалення розробленого Web-застосунку:

1. Розширення типів завдань. Пропонується інтегрувати завдання на аудіювання, вимову, роботу з аудіо- та відеоматеріалами, а також візуальні вправи (робота із зображеннями, інтерактивні картки). Це дозволить формувати комплексні мовні навички та урізноманітнити навчальний контент.

2. Впровадження адаптивного навчання. Доцільно розробити й впровадити алгоритми, які автоматично підбиратимуть рівень складності завдань залежно від поточного прогресу, частоти правильних відповідей і тематики, що дозволить кожному користувачу рухатися індивідуальною освітньою траєкторією.

3. Додавання аналітики, системи досягнень та бейджів. Рекомендується інтегрувати модулі розгорнутої статистики: графіки прогресу, історія навчання, порівняння з середніми показниками групи, а також систему досягнень (гейміфікація, мотиваційні бейджі), яка стимулюватиме користувачів до нових звершень.

4. Розробка мобільної версії та впровадження офлайн-режиму. Визначено доцільним створення окремого мобільного додатку або PWA (Progressive Web App), а також впровадження функціоналу збереження частини контенту на пристрої користувача для навчання без підключення до мережі Інтернет, що значно розширить аудиторію застосунку. Реалізація механізмів зворотного зв'язку від користувачів. Пропонується створити інтерфейс для швидкого надсилання фідбеку, повідомлення про помилки чи неточності, а також для додавання коментарів або рекомендацій щодо змісту окремих завдань.

5. Інтеграція із зовнішніми платформами та сервісами. Перспективним напрямом є підключення сторонніх освітніх сервісів (наприклад, Google Classroom, Duolingo, онлайн-словники) для автоматичного оновлення навчальної бази, експорту/імпорту прогресу та синхронізації з іншими освітніми інструментами.

Реалізація цих напрямків розвитку сприятиме підвищенню ефективності навчання, зробить застосунок більш конкурентоспроможним на ринку цифрових освітніх послуг та дозволить краще адаптувати його під потреби різних груп користувачів.

### **Висновки до розділу 3**

Здійснено комплексний аналіз ефективності розробленого Web-застосунку для вивчення англійських слів. Обрана архітектурна модель, поєднана із сучасним адаптивним інтерфейсом, забезпечує стабільну роботу системи на різних типах

пристроїв, високий рівень юзабіліті та миттєвий зворотний зв'язок із користувачем. Такий підхід сприяє зменшенню когнітивного навантаження і дозволяє користувачам з різним досвідом легко інтегрувати навчання у повсякденне життя.

Система демонструє високу ефективність при виконанні основних освітніх завдань - підвищення мотивації та залученості користувачів. Проведене тестування показало, що ключові сценарії роботи системи виконуються без критичних помилок і затримок.

Водночас визначено певні обмеження поточної версії - зокрема, відсутність розширених типів завдань (аудіювання, робота з медіаконтентом), не повністю реалізована адаптивність навчання залежно від рівня користувача, а також перспективи інтеграції із зовнішніми освітніми платформами та оптимізації для мобільних пристроїв.

Реалізація запропонованих шляхів удосконалення - впровадження додаткових аналітичних інструментів, розширення функціоналу, інтеграція з іншими освітніми сервісами - дозволить не лише підвищити ефективність навчання, але й зробить платформу більш універсальною, привабливою та конкурентоспроможною на ринку цифрової освіти.

Розроблений Web-застосунок може слугувати основою для подальших наукових досліджень та практичного впровадження сучасних технологій у процес вивчення англійської мови, сприяючи розвитку персоналізованих, інтерактивних та ефективних освітніх рішень.

## ВИСНОВКИ

У рамках цієї роботи досліджено, спроектовано та реалізовано сучасний Web-застосунок для вивчення англійських слів, який відповідає актуальним тенденціям цифрової освіти та вимогам до інтерактивних освітніх платформ. При цьому використання клієнт-серверної архітектури на основі стеку MERN (MongoDB, Express.js, React, Node.js) забезпечує високу гнучкість системи, її масштабованість, безпеку персональних даних та простоту інтеграції додаткових функцій.

Проаналізовано особливості організації взаємодії користувача з системою, розроблено й впроваджено багаторівневий механізм аутентифікації й авторизації, створено оптимальну структуру бази даних для збереження навчального контенту, облікових записів користувачів і їхнього прогресу.

Застосування сучасних технологій адаптивної верстки й компонентного підходу до розробки Web-інтерфейсу дозволило створити універсальний і зручний продукт для користувачів з різних пристроїв.

Розроблений функціонал - реєстрація, авторизація, проходження тестів, ведення особистої статистики, миттєвий зворотний зв'язок - реалізовано відповідно до принципів user-centered design, що сприяє підвищенню мотивації, залученості й ефективності навчання. Проведене тестування показало високу стабільність роботи платформи, її швидкодію й відсутність критичних помилок навіть при збільшенні навантаження.

Розроблена система створює надійну основу для подальшого розвитку: впровадження нових типів завдань, адаптивного навчання, інтеграції з іншими освітніми платформами та аналітичними інструментами. Подальше вдосконалення функціоналу дозволить не лише підвищити ефективність індивідуального навчання, але й зробити застосунок конкурентоспроможним рішенням у сфері цифрової освіти.

Розроблений Web-застосунок не лише повністю відповідає заявленим цілям роботи, але й має потенціал для практичного впровадження та подальших наукових досліджень у галузі EdTech, сприяючи поширенню персоналізованих, адаптивних та інтерактивних підходів до вивчення англійської мови.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Литвин, О. В. (2020). Цифровізація навчання іноземних мов: нові виклики та можливості. Науковий вісник Ужгородського університету. Серія: Педагогіка, соціальна робота, (1), 64–67.
2. Пономаренко, О. А. (2018). Методичні основи формування лексичних навичок у студентів закладів вищої освіти засобами інформаційно-комунікаційних технологій. Вісник Харківського національного педагогічного університету імені Г. С. Сковороди. Педагогіка, (58), 150–158.
3. Коваленко, С. М. (2019). Гейміфікація як засіб підвищення ефективності вивчення іноземних мов у вищих навчальних закладах. Педагогічні науки: теорія, історія, інноваційні технології, (6), 178–188.
4. Іванова, О. О. (2017). Застосування технології інтервального повторення в навчанні іноземних мов. Вісник Київського національного лінгвістичного університету. Серія: Педагогіка та психологія, (26), 137–145.
5. Мельник, Н. В. (2021). Аналіз функціональних можливостей онлайн-платформ для вивчення іноземних мов. Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми, (65), 198–203.
6. Гутнік, О. В. Технології React та Vue.js для створення SPA-додатків. // Сучасні проблеми програмування, 2022. – №3. – С. 47–52.
7. Головченко О. В. Node.js та Express у побудові RESTful-сервісів // Вісник НТУУ "КПІ". Інформатика та обчислювальна техніка. – 2021. – № 71. – С. 58–63.
8. Парфенюк О. І. Використання MongoDB у Web-проектах: переваги та виклики // Комп'ютерні науки та інформаційні технології. – 2022. – № 4. – С. 88–94.

9. Стеценко Л. М. Забезпечення інформаційної безпеки в Web-застосунках // Інформаційна безпека та кіберзахист. – 2021. – №2. – С. 25–30.
10. Хоменко В. І. Хмарні технології для хостингу Web-проектів: аналіз платформ // Сучасні інформаційні технології. – 2023. – №1. – С. 34–39.
11. Дьяків, Р. М. Використання SPA в освітніх Web-застосунках. // Журнал інформатики та кібернетики. – 2022. – №2. – С. 46–51.
12. Яременко, Т. С. Застосування HTML5 API в навчальних платформах. // Вісник Київського університету. Серія: Кібернетика. – 2021. – №4. – С. 28–33.
13. Грищенко О. М. Архітектура сучасних навчальних застосунків: MVC, MVVM. // Програмні системи і технології. – 2023. – №2. – С. 14–19.
14. Мельник В. Ю. Адаптивна верстка навчальних інтерфейсів: порівняльний аналіз. // Інженерія інтерфейсів. – 2022. – №1. – С. 50–55.
15. Шевченко Д. О. JAMstack як підхід до побудови освітніх платформ. // Web-технології і освіта. – 2023. – №2. – С. 41–47.
16. Коваленко І. В. Візуалізація результатів навчання в Web-застосунках. // Інформаційні системи в освіті. – 2021. – №3. – С. 30–36.
17. Петренко Л. С. AI-технології у вивченні мов: огляд можливостей Web-інтеграції. // Освітні технології. – 2023. – №1. – С. 64–69.
18. Білан І. В. Можливості PWA у контексті навчальних Web-застосунків. // Комп'ютерні науки та мобільні технології. – 2023. – №4. – С. 59–65.
19. Пилипчук О. В. Методологія вибору стеку технологій у Web-розробці. // Системи управління, навігації та зв'язку. – 2022. – №2. – С. 39–44.
20. W3C. HTML5 Specification. – [Електронний ресурс]. – Режим доступу: <https://www.w3.org/TR/html5/>
21. Барановський Ю. А. CSS-фреймворки: огляд можливостей та доцільність використання. // Проблеми програмування. – 2021. – №1. – С. 28–34.

22. Павленко Д. В. React.js у реалізації інтерактивних Web-інтерфейсів. // Комп'ютерні науки та сучасні технології. – 2022. – №3. – С. 16–21.
23. Жук В. М. Node.js як основа backend-розробки: переваги та недоліки. // Програмна інженерія. – 2023. – №1. – С. 44–49.
24. Степаненко І. І. Express.js – мікрофреймворк для швидкої побудови API. // Інформатика та системи обробки даних. – 2022. – №2. – С. 50–55.
25. RFC 7519 – JSON Web Token (JWT). – [Електронний ресурс]. – Режим доступу: <https://datatracker.ietf.org/doc/html/rfc7519>
26. Кравчук О. В. Особливості використання MongoDB у Web-проєктах. // Технології зберігання даних. – 2021. – №4. – С. 33–39.
27. Офіційна документація Postman. – [Електронний ресурс]. – Режим доступу: <https://www.postman.com/>
28. Данилюк М. І. Розробка навчального Web-додатку на основі стеку MERN. // Цифрові технології в освіті. – 2023. – №1. – С. 41–47.
29. Setlab.net. Дисертація. Розділ 3.1. [Електронний ресурс]. – Режим доступу: <http://www.setlab.net/?view=dissertation-3-1> (дата звернення: поточна дата).
30. Григошкіна Я. В., Яненко І. В., Янишин О. К. Сучасні технології викладання іноземних мов: аналіз технологій для забезпечення ефективної індивідуальної освіти Журнал: Академічні візії, №21, 2023
31. Левицька Л., Сіренко В. Використання інформатично-цифрових технологій у процесі вивчення іноземних мов Вісник Черкаського національного університету імені Богдана Хмельницького. Серія: Педагогічні науки, №2, 2023
32. Майєр Н. В., Коваль Т. І. Цифрові засоби для формування ІКТ-компетентності майбутніх викладачів іноземних мов Журнал: Іноземні мови, №1, 2023
33. Тищенко М. А. Мультимедійні технології в навчанні іноземних мов Журнал: Адаптивні системи автоматичного управління, №17, 2010

34. Гарнопольський О. Б., Кожушко С. П., Кабанова М. Р. Гейміфікація в навчанні іноземних мов у вищій школі Журнал: Іноземні мови, №3, 2018
35. Тесцова О. О., Лопата І. Л. Роль цифрових технологій у процесі вивчення іноземної мови Журнал: Наукові записки. Серія: Філологічні науки, №204, 2023
36. Язловицька О., Птушка А., Байбекова Л., Пруднікова А. Вивчення іноземної мови у рамках процесу діджиталізації освіти Журнал: Сучасні дослідження з іноземної філології, №23, 2023
37. Рокицька О. Ю. Використання інформаційних технологій у навчанні іноземних мов Журнал: Інформаційно-комунікаційні технології в освіті, №4, 2017
38. Симоненко Н. О. Гейміфікація як засіб підвищення мотивації та пізнавального інтересу студентів у процесі навчання іноземних мов Матеріали XI Всеукраїнської наукової конференції, Суми, 2024
39. Шикун А. Мобільність та вивчення мови: використання онлайн-платформ для вивчення іноземних мов Журнал: Людинознавчі студії. Серія «Педагогіка», №16, 2023
40. Chinnery, G. M. (2006). Emerging technologies: Going to the MALL: Mobile Assisted Language Learning. *Language Learning & Technology*, 10(1), 9–16.
41. Nation, I. S. P. (2013). *Learning Vocabulary in Another Language*. Cambridge University Press.
42. Schmitt, N. (2008). Review article: Instructed second language vocabulary learning. *Language Teaching Research*, 12(3), 329–363.
43. González, C. S., Gómez, N., & Navarro, V. (2020). Gamification in language learning: A systematic review. *Education and Information Technologies*, 25(6), 4605–4637.
44. Ma, Q., & Kelly, P. (2006). Computer-assisted vocabulary learning: Design and evaluation. *Computer Assisted Language Learning*, 19(1), 15–45.

45. Chapelle, C. A. (2001). *Computer applications in second language acquisition: Foundations for teaching, testing and research*. Cambridge University Press.
46. Kapp, K. M. (2012). *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*. Pfeiffer.
47. Krashen, S. (1982). *Principles and Practice in Second Language Acquisition*. Pergamon Press.
48. Warschauer, M., & Kern, R. (Eds.). (2000). *Network-Based Language Teaching: Concepts and Practice*. Cambridge University Press.
49. Allen, M. (2012). *Smart Learning Environments*. Springer.

**ЗГОДА здобувача(чки) вищої освіти**

Державного університету економіки і технологій про  
перевірку кваліфікаційної роботи на прояви  
академічного плагіату  
та розміщення в Репозитарії Університету

Я, Стружесдський Члєя Андрійович (ПП),

підтримую політику Державного університету економіки і технологій з академічної доброчесності і відкритого доступу.

Засвідчую, що кваліфікаційна бакалаврська (магістерська) робота

Розробка WEB - застосунок для відстеження англійських слід

(назва роботи повністю) виконана самостійно та не містить академічного плагіату. Я не надавав(ла) і не одержував(ла) недозволену допомогу під час підготовки цієї роботи. Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Державного університету економіки і технологій ознайомлений(а). Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення норм академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

Також я поінформований(на), що відповідно до «Положення про Репозитарій (електронну базу даних) Державного університету економіки і технологій» зазначена робота буде розміщена в Електронному архіві Університету (Репозитарії ДУЕТ). З умовами такого розміщення ознайомлений(на).

10.06.25

Дата

  
підпис

Ч.А. Стружесдський  
ініціали, прізвище (власноруч)