

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ

ННІ	Економіки та бізнес-освіти
Кафедра	Економіки та цифрового бізнесу
Спеціальність	122 «Комп'ютерні науки»
Форма навчання	Денна

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

	Дорошенко Родіона Олеговича <i>(прізвище, ім'я, по батькові здобувача)</i>		
на тему	Розробка вебзастосунку для вивчення англійської мови <i>(повна назва теми)</i>		
за матеріалами	 <i>(повна назва бази дослідження)</i>		
науковий керівник	К.Т.Н. <i>(наук. ступінь, вчене звання)</i>	 <i>(підпис)</i>	Селезньов М.Є. <i>(прізвище, ініціали)</i>

Робота допущена до захисту в ЕК

Протокол засідання кафедри
від 12 червня 2026 р. № 15

Завідувач кафедри

(підпис)

к.е.н., доцент
наук. ступень, вчене звання

Радько В.М.
прізвище, ініціали

ЗАТВЕРДЖЕНО

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 бізнесуОсвітній ступінь бакалаврСпеціальність 122 Комп'ютерні науки**ЗАТВЕРДЖУЮ**Завідувач кафедри _____ **В.М. Радько**

“30” березня 2026 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ ЗДОБУВАЧУ

Дорошенко Родіону Олеговичу

1. Тема роботи Розробка вебзастосунку для вивчення англійської мовинауковий керівник роботи Селезньов Максим Євгенович, к.т.н.

затверджені наказом вищого навчального закладу від «23» березня 2026 р. № 193-ст

2. Строк подання здобувачем роботи 29.05.2026р.

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

Розділ 1 Теоретичні аспекти розробки вебзастосунку для вивчення англійської мовиРозділ 2 Проектування та розробка вебзастосунку для вивчення англійської мовиРозділ 3 Функціонування та елементи інтерфейсу вебзастосунку для вивчення англійської мовиОб'єкт дослідження – процес вивчення іноземних мов із використанням вебзастосунківПредмет дослідження – методи, технології та програмні засоби розробки вебзастосунків для вивчення англійської мовиМета кваліфікаційної бакалаврської роботи – розробити вебзастосунок для вивчення англійської мови, який забезпечує ефективне засвоєння лексики, розвиток мовних навичок та зручну взаємодію користувача з системою4. Дата видачі завдання 03.04.2026р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної бакалаврської роботи	Строк виконання етапів роботи	Відмітка керівника про виконання етапів (дата, підпис)
1	Підготовка розділу 1	до 17.04.2026р.	16.04.2026р.
2	Підготовка розділу 2	до 08.05.2026р.	12.05.2026р.
3	Підготовка розділу 3	до 25.05.2026р.	24.05.2026р.
4	Реєстрація завершеної кваліфікаційної роботи	до 29.05.2026р.	28.05.2026р.
5	Отримання відгуку від наукового керівника	04.06.2026р.	04.06.2026р.
6	Отримання зовнішньої рецензії	05.06.2026р.	05.06.2026р.
7	Попередній захист кваліфікаційної роботи на кафедрі	08.06.2026р.	08.06.2026р.
8	Перевірка кваліфікаційної роботи на плагіат	12.06.2026р.	12.06.2026р.
9	Допуск кафедрою кваліфікаційної роботи до захисту	12.06.2026р.	12.06.2026р.
10	Підготовка студента до захисту в ЕК	до 19.06.2026р.	18.06.2026р.

Завдання підготував науковий керівник

_____ Селезнєв М. Є.

Завдання одержав здобувач

_____ Дорошенко Р. О.

РЕФЕРАТ

Робота містить 64 сторінки, 14 малюнків, 30 джерел та 1 додаток.

Об'єкт дослідження: процес вивчення іноземних мов із використанням вебзастосунків.

Предмет дослідження: методи, технології та програмні засоби розробки вебзастосунків для вивчення англійської мови

Мета: розробити вебзастосунок для вивчення англійської мови, який забезпечує ефективне засвоєння лексики, розвиток мовних навичок та зручну взаємодію користувача з системою.

У результаті дослідження було проаналізовано сучасні підходи до вивчення іноземних мов із використанням інформаційно-комунікаційних технологій та проведено аналіз існуючих програмних засобів для мовного навчання. Обґрунтовано вибір технологічного стеку та архітектури програмного продукту. Спроектовано інформаційну модель і базу даних вебзастосунку, реалізовано механізми реєстрації та авторизації користувачів, формування навчальних сесій, контролю прогресу та інтервального повторення навчального матеріалу. Розроблено клієнтську та серверну частини вебзастосунку з використанням сучасних вебтехнологій. Проведено аналіз функціональних можливостей системи та оцінено ефективність її використання для організації процесу вивчення англійської мови.

Область застосування: результати роботи можуть бути використані у сфері цифрової освіти, системах дистанційного навчання, онлайн-платформах для вивчення іноземних мов, а також для організації самостійного навчання та контролю прогресу користувачів.

АНГЛІЙСЬКА МОВА, БАЗА ДАНИХ, ВЕБЗАСТОСУНОК, ДИСТАНЦІЙНЕ НАВЧАННЯ, ІНТЕРВАЛЬНЕ ПОВТОРЕННЯ, НАВЧАЛЬНА СИСТЕМА, FASTAPI, POSTGRESQL, REACT, REST API.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ ВЕБЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ	10
1.1 Огляд сучасних способів вивчення іноземних мов з використанням програмних засобів	10
1.2 Аналіз існуючих програмних засобів для вивчення іноземних мов ...	13
1.3 Вибір технологічного стеку та архітектури вебзастосунку	19
Висновки до розділу 1	21
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ	23
2.1 Проєктування інформаційної моделі та бази даних вебзастосунку	23
2.2 Опис основних алгоритмів та методів обробки даних	30
2.3 Реалізація серверної та клієнтської частин вебзастосунку	36
Висновки до розділу 2	45
РОЗДІЛ 3 ФУНКЦІОНУВАННЯ ТА ЕЛЕМЕНТИ ІНТЕРФЕЙСУ ВЕБЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ	47
3.1 Огляд функціоналу розробленого вебзастосунку	47
3.2 Опис взаємодії користувача з розробленим вебзастосунком	53
3.3 Аналіз ефективності розробленого вебзастосунку та можливих обмежень в його використанні	55
Висновки до розділу 3	58
ВИСНОВКИ	60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
ДОДАТКИ	65

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API - Application Programming Interface (інтерфейс програмування застосунків)

БД - база даних

ІКТ - інформаційно-комунікаційні технології

MVC - Model-View-Controller (архітектурний шаблон програмного забезпечення)

REST - Representational State Transfer (архітектурний стиль взаємодії вебсервісів)

JSON - JavaScript Object Notation (формат обміну даними)

JWT - JSON Web Token (токен автентифікації та авторизації)

SQL - Structured Query Language (мова структурованих запитів)

СУБД - система управління базами даних

UI - User Interface (інтерфейс користувача)

UX - User Experience (користувацький досвід)

HTML - HyperText Markup Language (мова розмітки гіпертексту)

CSS - Cascading Style Sheets (каскадні таблиці стилів)

DOM - Document Object Model (об'єктна модель документа)

HTTP - HyperText Transfer Protocol (протокол передачі гіпертексту)

SPA - Single Page Application (односторінковий вебзастосунок)

ORM - Object-Relational Mapping (об'єктно-реляційне відображення)

ER-діаграма - Entity-Relationship Diagram (діаграма «сутність-зв'язок»)

CEFR - Common European Framework of Reference for Languages
(Загальноєвропейські рекомендації з мовної освіти)

ВСТУП

У сучасних умовах розвитку інформаційних технологій все більшого значення набуває цифровізація освітніх процесів. Однією з важливих сфер є вивчення іноземних мов, зокрема англійської, яка є мовою міжнародного спілкування, науки та інформаційних технологій. Незважаючи на значну кількість існуючих освітніх ресурсів, багато з них не забезпечують достатнього рівня інтерактивності, персоналізації та зручності використання. У зв'язку з цим виникає потреба у створенні сучасних вебзастосунків, які дозволяють підвищити ефективність навчання та зробити процес вивчення мови більш доступним і зручним.

Сучасні вебзастосунки відкривають широкі можливості для організації навчального процесу, забезпечуючи швидкий доступ до навчальних матеріалів, інтерактивну взаємодію з контентом та зручний користувацький інтерфейс. Вони дозволяють реалізувати динамічне оновлення даних, оперативну обробку запитів та ефективну взаємодію між користувачем і системою. Використання вебтехнологій у сфері навчання сприяє підвищенню якості освітнього процесу та забезпечує більш гнучкий підхід до засвоєння знань.

Сучасні платформи для вивчення мов поєднують різноманітні підходи до навчання, серед яких гейміфікація, адаптивне навчання, використання мультимедійних матеріалів та інтерактивних вправ. Це дозволяє користувачам ефективно вивчати лексику, граматику, розвивати навички аудіювання, читання та письма. Крім того, такі системи надають можливість відстеження прогресу, що сприяє підвищенню мотивації та самостійності користувачів.

Актуальність теми полягає в необхідності розробки сучасного вебзастосунку для вивчення англійської мови, який забезпечуватиме ефективне засвоєння навчального матеріалу, зручність використання та інтерактивність. Використання сучасних вебтехнологій дозволяє підвищити швидкість роботи системи, покращити взаємодію користувача з інтерфейсом та створити комфортне навчальне середовище.

На сьогоднішній день питання цифрового навчання мов активно досліджується як у практичному, так і в науковому аспектах. Існує велика кількість онлайн-платформ і сервісів, які пропонують інтерактивне навчання, персоналізовані курси та різноманітні інструменти для самостійної роботи. Світова практика демонструє, що розвиток таких систем спрямований на підвищення доступності освіти, оптимізацію навчального процесу та покращення результатів навчання.

Метою дипломної роботи є розробка вебзастосунку для вивчення англійської мови, який забезпечує ефективне засвоєння лексики, розвиток мовних навичок та зручну взаємодію користувача з системою.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- дослідити сучасні підходи до розробки вебзастосунків та їх використання у сфері освіти;
- проаналізувати існуючі програмні засоби для вивчення іноземних мов;
- визначити функціональні вимоги до вебзастосунку;
- спроектувати структуру системи та базу даних;
- розробити інтерфейс користувача;
- реалізувати основний функціонал застосунку;
- провести тестування та оцінити ефективність роботи системи.

Об'єктом дослідження є процес вивчення іноземних мов із використанням вебзастосунків.

Предметом дослідження є методи, технології та програмні засоби розробки вебзастосунків для вивчення англійської мови.

У процесі виконання дипломної роботи були використані такі методи дослідження:

- аналіз наукової та навчальної літератури;
- аналіз технічної документації; системний підхід до проєктування інформаційних систем;
- методи моделювання структури даних і логіки роботи за стосунку;

– практична реалізація програмного продукту з використанням сучасних вебтехнологій.

Інформаційну базу дослідження становлять наукові праці, матеріали з відкритих джерел, технічна документація та приклади сучасних вебзастосунків для вивчення мов.

Практичне значення отриманих результатів полягає у створенні вебзастосунку для вивчення англійської мови, який може бути використаний як інструмент для самостійного навчання, а також як допоміжний засіб у навчальному процесі. Розроблений застосунок дозволяє підвищити ефективність засвоєння матеріалу, покращити організацію навчання та забезпечити зручну взаємодію користувача із системою.

РОЗДІЛ 1

ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ ВЕБЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ

1.1 Огляд сучасних способів вивчення іноземних мов з використанням програмних засобів

У сучасному світі знання іноземних мов, зокрема англійської, є важливою складовою професійного розвитку, міжкультурної комунікації та доступу до глобальних інформаційних ресурсів. У зв'язку зі стрімким розвитком інформаційно-комунікаційних технологій (ІКТ) значно трансформувалися підходи до вивчення мов, що призвело до появи нових ефективних методів навчання з використанням програмних засобів [3]. Використання ІКТ сприяє підвищенню доступності навчальних ресурсів, гнучкості освітнього процесу та його адаптації до індивідуальних потреб користувачів.

Одним із найпоширеніших сучасних підходів є використання мобільних застосунків для вивчення мов. Такі застосунки забезпечують постійний доступ до навчального контенту незалежно від місця перебування користувача. Вони, як правило, включають інтерактивні вправи з граматики, лексики, аудіювання та вимови, що дозволяє комплексно розвивати мовні навички. Особливістю мобільних застосунків є використання гейміфікації - впровадження ігрових елементів, таких як система балів, досягнень та рівнів. Це значно підвищує мотивацію користувачів і стимулює регулярність занять. Інтерактивні цифрові інструменти сприяють більш глибокому залученню користувача до навчального процесу, що позитивно впливає на ефективність засвоєння матеріалу.

Важливе місце у сучасному навчанні займають вебзастосунки, які мають ряд переваг порівняно з традиційними програмними рішеннями. Вони не потребують встановлення, є доступними з будь-якого пристрою з доступом до Інтернету та забезпечують кросплатформеність. Вебресурси виступають ефективним засобом формування мовних компетенцій, оскільки надають

доступ до широкого спектра навчальних матеріалів - текстів, відео, аудіо та інтерактивних вправ.

Крім того, вебзастосунки дозволяють реалізувати персоналізоване навчання, що передбачає адаптацію контенту відповідно до рівня знань користувача, його інтересів та темпу навчання. Завдяки цьому підвищується ефективність освітнього процесу. Дослідження показують, що використання вебресурсів сприяє розвитку всіх ключових мовних навичок - читання, письма, аудіювання та говоріння, що досягається завдяки інтеграції різних типів навчального контенту [18]. Важливою перевагою є також можливість отримання швидкого зворотного зв'язку, що дозволяє оперативно коригувати помилки та вдосконалювати знання.

Значного поширення набули онлайн-платформи та мультимедійні ресурси, які забезпечують більш наочний і інтерактивний формат навчання. До них належать відеоуроки, подкасти, інтерактивні вправи та тести. Використання мультимедійних засобів позитивно впливає на процес засвоєння інформації, оскільки поєднує різні канали сприйняття - зоровий та слуховий. Це сприяє кращому розумінню навчального матеріалу та його довготривалому запам'ятовуванню.

Зокрема, використання відеоконтенту дозволяє користувачам сприймати мову у природному контексті, що сприяє формуванню комунікативної компетенції. Аудіоматеріали, у свою чергу, допомагають розвивати навички сприйняття мови на слух та покращують вимову. Інтеграція мультимедійних елементів створює динамічне освітнє середовище, яке стимулює активну участь користувачів у навчальному процесі.

Окрім формального навчання, значну роль відіграє неформальне онлайн-середовище. Воно включає використання соціальних мереж, форумів, блогів та відеоплатформ, які дозволяють користувачам взаємодіяти між собою та практикувати мову у реальних комунікативних ситуаціях. Таке середовище сприяє розвитку самостійності користувача та інтегрує навчання у повсякденне життя.

Дослідження підтверджують, що неформальне онлайн-навчання створює умови для постійної мовної практики, оскільки користувачі можуть взаємодіяти з контентом у зручний для них час. Крім того, активна участь у цифрових спільнотах сприяє розвитку навичок письма та комунікації, а також формує здатність до самостійного навчання.

Особливу роль у сучасному навчанні відіграють технології Web 2.0, які забезпечують інтерактивність та співпрацю між користувачами. До таких технологій належать блоги, вікі-системи, соціальні мережі та інші сервіси, що дозволяють створювати та обмінюватися контентом. Використання Web 2.0 сприяє розвитку критичного мислення, комунікативних навичок та підвищує мотивацію до навчання.

Важливою перевагою цих технологій є можливість реалізації спільного навчання (*collaborative learning*), у процесі якого користувачі можуть обмінюватися знаннями, виконувати групові завдання та отримувати зворотний зв'язок. Це робить навчальний процес більш інтерактивним і наближеним до реальних умов використання мови.

Сучасні підходи до вивчення іноземних мов також базуються на принципах природного засвоєння мови. Вони передбачають активне використання автентичних матеріалів, таких як відео, діалоги та тексти, а також регулярну практику мовлення. Такий підхід дозволяє краще розуміти структуру мови та її використання у реальному житті.

Важливим елементом є мікронавчання (*microlearning*), яке передбачає поділ навчального матеріалу на невеликі частини та регулярне його повторення. Це сприяє кращому засвоєнню інформації та підтримці мотивації користувача. Також важливо створювати комфортне середовище навчання, у якому користувач не боїться робити помилки, оскільки вони є природною частиною процесу навчання.

Окрему категорію становлять цифрові платформи, що реалізують сучасні підходи до запам'ятовування інформації. Вони базуються на принципах повторення матеріалу через певні інтервали часу та використанні адаптивних

алгоритмів навчання. Це дозволяє оптимізувати процес засвоєння лексики та підвищити ефективність навчання.

Такі системи враховують індивідуальні особливості користувача, рівень його знань та швидкість засвоєння матеріалу. Крім того, вони забезпечують можливість відстеження прогресу, що дозволяє користувачам контролювати результати навчання та своєчасно коригувати навчальний процес. Використання подібних підходів сприяє довготривалому збереженню знань і формуванню стійких мовних навичок.

Незважаючи на значну кількість існуючих програмних рішень, багато з них мають певні недоліки, такі як недостатня персоналізація, обмежена адаптивність або складність інтерфейсу. Це створює потребу у розробці нових вебзастосунків, які поєднуюватимуть сучасні технології, ефективні методики навчання та зручність використання.

Таким чином, сучасні програмні засоби відкривають широкі можливості для вивчення іноземних мов. Їх використання дозволяє забезпечити доступність навчання, підвищити його ефективність та адаптувати освітній процес до індивідуальних потреб користувача, що є важливим фактором у сучасному цифровому середовищі.

1.2 Аналіз існуючих програмних засобів для вивчення іноземних мов

У сучасному освітньому середовищі існує велика кількість програмних засобів для вивчення іноземних мов. Вони відрізняються за функціональністю, підходами до навчання та рівнем інтерактивності. Розглянемо найбільш популярні рішення.

1.2.1 Аналіз платформи Duolingo

Одним із найпопулярніших програмних засобів для вивчення іноземних мов є Duolingo - безкоштовна онлайн-платформа, яка використовує ігровий

підхід до навчання. Вона була створена з метою зробити процес вивчення мов доступним, цікавим та ефективним для широкого кола користувачів. На сьогоднішній день платформа підтримує десятки мов і має мільйони активних користувачів по всьому світу.

Основою навчального процесу в Duolingo є короткі інтерактивні уроки, які охоплюють різні аспекти мовної підготовки, зокрема лексику, граматику, аудіювання та читання. Кожен урок побудований у формі серії завдань, які включають переклад тексту, вибір правильної відповіді, складання речень із запропонованих слів, а також вправи на розпізнавання мовлення. Такий підхід дозволяє залучати користувача до активної взаємодії з матеріалом і сприяє кращому засвоєнню знань.

Структура навчання організована у вигляді рівнів, які поступово ускладнюються. Користувач проходить так зване «дерево навичок», де кожен блок відповідає певній темі або граматичному аспекту. Це забезпечує послідовність навчання та дозволяє поступово розширювати мовні знання. Крім того, система заохочує регулярність занять шляхом встановлення щоденних цілей.

Однією з ключових особливостей платформи є використання гейміфікації. У процесі навчання користувач отримує бали досвіду, досягнення, серії днів без перерв (streak), а також може змагатися з іншими користувачами у рейтингових таблицях. Такий підхід значно підвищує мотивацію та стимулює користувачів до регулярної практики. Гейміфікація також знижує психологічний бар'єр перед навчанням, роблячи його більш схожим на гру.

Важливим елементом платформи є адаптивність навчального процесу. Duolingo використовує алгоритми машинного навчання для аналізу дій користувача, його помилок та швидкості виконання завдань [5]. На основі цих даних система може пропонувати індивідуалізовані вправи, повторення складних тем і оптимальний темп навчання. Це дозволяє враховувати рівень підготовки користувача та підвищує ефективність засвоєння матеріалу.

Крім того, платформа активно використовує мультимедійні можливості. Вправи на аудіювання включають озвучення носіями мови, що допомагає користувачам звикати до правильної вимови та інтонації. Також доступні вправи на вимову, які дозволяють практикувати усне мовлення за допомогою розпізнавання голосу.

Суттєвою перевагою Duolingo є доступність. Платформа має вебверсію та мобільні застосунки, що дозволяє користувачам навчатися у будь-який час і в будь-якому місці. Більшість функцій доступні безкоштовно, що робить її привабливою для широкої аудиторії.

Дослідження ефективності використання платформи показують, що результати навчання користувачів можуть бути співставними з результатами студентів, які проходять традиційні курси, зокрема у розвитку навичок читання та аудіювання. Це підтверджує доцільність використання подібних цифрових інструментів у процесі вивчення мов.

Разом із тим, платформа має і певні недоліки. Зокрема, вона не завжди забезпечує глибоке розуміння граматичних правил, оскільки навчання здебільшого базується на інтуїтивному підході. Також обмеженою є практика живого мовлення, оскільки взаємодія з іншими користувачами або носіями мови практично відсутня. Крім того, деякі завдання можуть повторюватися, що знижує інтерес до навчання на більш високих рівнях.

Ще одним обмеженням є орієнтація платформи переважно на початковий та середній рівні володіння мовою. Для користувачів із високим рівнем знань функціональність може бути недостатньо складною або різноманітною.

Таким чином, Duolingo є ефективним інструментом для початкового етапу вивчення іноземних мов, який поєднує доступність, інтерактивність та мотиваційні механізми. Водночас для досягнення високого рівня володіння мовою доцільно поєднувати його з іншими методами навчання, що забезпечують розвиток усного мовлення та глибше розуміння граматики.

1.2.2 Аналіз платформи Memrise

Memrise є популярним програмним засобом для вивчення іноземних мов, який орієнтується насамперед на розвиток словникового запасу та навичок сприйняття мови на слух. Платформа поєднує сучасні цифрові технології з когнітивними підходами до навчання, що дозволяє зробити процес засвоєння лексики більш ефективним і наближеним до реального мовного середовища [7].

Однією з ключових особливостей Memrise є використання відеоматеріалів за участю носіїв мови. Користувачам пропонуються короткі відеофрагменти, у яких реальні люди вимовляють слова та фрази у природному контексті. Це дозволяє не лише запам'ятовувати нову лексику, але й формувати правильну вимову, інтонацію та розуміння живої мови. Такий підхід значно підвищує ефективність аудіювання та допомагає краще адаптуватися до реальних умов спілкування.

Навчальний процес у Memrise побудований на основі методів інтервального повторення (spaced repetition), які передбачають повторення навчального матеріалу через певні проміжки часу [8]. Цей підхід базується на особливостях роботи пам'яті людини та сприяє довготривалому збереженню інформації. Система автоматично визначає, які слова або фрази користувач засвоїв недостатньо добре, і пропонує їх для повторення у відповідний момент.

Крім того, платформа використовує різноманітні інтерактивні вправи, серед яких: вибір правильного варіанту, введення перекладу, складання слів із частин, а також вправи на аудіювання. Такі завдання сприяють активній взаємодії користувача з навчальним матеріалом і дозволяють закріплювати знання у різних формах.

Ще однією перевагою Memrise є структура навчання, яка поділяється на короткі модулі або сесії. Це відповідає концепції мікронавчання (microlearning), що передбачає засвоєння інформації невеликими порціями. Такий підхід дозволяє користувачам ефективно використовувати навіть обмежений час для навчання, що особливо актуально в умовах сучасного темпу життя.

Платформа також надає можливість відстеження прогресу користувача, що дозволяє оцінювати результати навчання та підтримувати мотивацію. Наявність статистики, досягнень і візуалізації результатів сприяє більш усвідомленому підходу до навчання та стимулює регулярність занять.

Важливою особливістю є можливість вивчення лексики у контексті. Завдяки використанню прикладів речень і реальних ситуацій, користувач не лише запам'ятовує окремі слова, але й розуміє їхнє застосування у мовленні. Це сприяє формуванню комунікативної компетенції.

Разом із тим, платформа має певні обмеження. Зокрема, основна увага приділяється запам'ятовуванню слів і фраз, тоді як граматичні аспекти мови представлені менш детально. Це може ускладнювати формування навичок побудови складних речень і розуміння граматичних правил. Крім того, можливості розвитку усного мовлення є обмеженими, оскільки взаємодія з іншими користувачами або викладачами відсутня.

Також варто зазначити, що ефективність використання Memrise значною мірою залежить від регулярності занять. Без систематичного повторення матеріалу користувач може швидко втрачати засвоєні знання.

Таким чином, Memrise є ефективним інструментом для вивчення лексики та розвитку навичок аудіювання. Завдяки використанню відеоматеріалів, інтервального повторення та інтерактивних вправ платформа забезпечує високий рівень залученості користувача. Водночас для комплексного вивчення мови доцільно поєднувати її з іншими ресурсами, що забезпечують розвиток граматики та усного мовлення.

1.2.3 Аналіз платформи Busuu

Busuu є сучасною платформою для вивчення іноземних мов, яка поєднує самостійне навчання з елементами соціальної взаємодії. Такий підхід дозволяє не лише засвоювати теоретичний матеріал, а й застосовувати отримані знання у

практичному спілкуванні, що є важливим для формування комунікативної компетенції.

Однією з ключових особливостей платформи є можливість взаємодії з іншими користувачами, зокрема носіями мови. Користувачі можуть виконувати письмові або усні завдання та отримувати зворотний зв'язок від інших учасників спільноти. Це дозволяє виправляти помилки, вдосконалювати мовлення та отримувати практичний досвід використання мови у реальних ситуаціях. Такий підхід значно підвищує ефективність навчання, оскільки забезпечує інтерактивність і залучення до комунікації.

Навчальний контент у Busuu організований у вигляді структурованих курсів, які відповідають міжнародним стандартам CEFR (Common European Framework of Reference for Languages) [10]. Це дозволяє чітко визначити рівень володіння мовою (A1–C1) та відстежувати прогрес користувача. Курси включають різноманітні вправи, спрямовані на розвиток лексики, граматики, читання, аудіювання та письма.

Платформа також використовує адаптивний підхід до навчання. На основі початкового тестування та подальшої активності користувача формується індивідуальний план навчання. Це дозволяє враховувати рівень підготовки, цілі та темп навчання кожного користувача. Крім того, система може нагадувати про необхідність повторення матеріалу, що сприяє кращому закріпленню знань.

Важливою перевагою Busuu є орієнтація на практичне застосування мови. Завдання часто моделюють реальні життєві ситуації, такі як знайомство, подорожі або ділове спілкування. Це дозволяє користувачам не лише запам'ятовувати слова та правила, а й навчатися використовувати їх у відповідному контексті.

Крім того, платформа надає можливість проходження тестів і отримання сертифікатів, які підтверджують рівень володіння мовою. Це може бути корисним для користувачів, які планують використовувати знання мови у професійній діяльності або навчанні.

Разом із тим, платформа має певні недоліки. Основним із них є обмежений функціонал безкоштовної версії. Багато можливостей, зокрема доступ до повного курсу, розширених вправ і сертифікації, доступні лише за умови оформлення платної підписки. Це може обмежувати доступність платформи для деяких користувачів.

Також варто зазначити, що ефективність соціальної взаємодії залежить від активності спільноти. У деяких випадках користувач може не отримати швидкого або достатньо якісного зворотного зв'язку. Крім того, хоча платформа охоплює всі основні мовні навички, вона може не забезпечувати достатньо глибокого теоретичного пояснення граматики для користувачів, які потребують більш детального аналізу мовних структур.

Проведений аналіз показує, що Busuu займає важливе місце серед сучасних програмних засобів для вивчення мов завдяки поєднанню індивідуального навчання та соціальної взаємодії. На відміну від платформ, орієнтованих переважно на гейміфікацію (Duolingo) або запам'ятовування лексики (Memrise), Busuu робить акцент на практичному використанні мови та комунікації.

Таким чином, платформа є ефективним інструментом для комплексного вивчення іноземної мови, особливо для розвитку навичок письма та говоріння. Водночас для досягнення максимального результату доцільно поєднувати її з іншими ресурсами, що доповнюють окремі аспекти навчання.

1.3 Вибір технологічного стеку та архітектури вебзастосунку

Розробка сучасного вебзастосунку потребує обґрунтованого вибору технологічного стеку та архітектурного підходу, що забезпечують ефективність, масштабованість та зручність подальшої підтримки системи. У межах даного проєкту для створення вебзастосунку з вивчення англійської мови було обрано стек технологій, який включає фронтенд на основі React із використанням MobX та Material UI, бекенд на базі FastAPI та систему

керування базами даних PostgreSQL. Архітектурним підходом обрано модель MVC (Model-View-Controller).

Фронтенд застосунку реалізується за допомогою бібліотеки React, яка є однією з найпопулярніших технологій для створення користувацьких інтерфейсів. Основною перевагою React є компонентний підхід до розробки, що дозволяє створювати повторно використовувані елементи інтерфейсу та спрощує підтримку коду [11]. Крім того, використання віртуального DOM забезпечує високу продуктивність та швидке оновлення інтерфейсу без повного перезавантаження сторінки.

Для управління станом застосунку використовується бібліотека MobX, яка забезпечує реактивний підхід до обробки даних. MobX дозволяє автоматично відслідковувати зміни стану та оновлювати інтерфейс без необхідності складної конфігурації [12]. Це спрощує розробку та підвищує читабельність коду, особливо у випадку динамічних застосунків із великою кількістю взаємозалежних компонентів.

Інтерфейс користувача реалізується із застосуванням бібліотеки Material UI, яка надає готові компоненти, що відповідають сучасним принципам дизайну. Використання цієї бібліотеки дозволяє значно прискорити процес розробки та забезпечити єдиний стиль оформлення застосунку. Крім того, Material UI підтримує адаптивний дизайн, що забезпечує коректне відображення інтерфейсу на різних пристроях.

Бекенд частина застосунку реалізується з використанням фреймворку FastAPI, який базується на мові програмування Python. FastAPI відзначається високою продуктивністю, простотою створення REST API та підтримкою асинхронного програмування [14]. Це дозволяє ефективно обробляти велику кількість запитів і забезпечує швидку взаємодію між клієнтською та серверною частинами застосунку. Додатковою перевагою є автоматична генерація документації API, що спрощує тестування та інтеграцію системи.

Для зберігання даних використовується реляційна система керування базами даних PostgreSQL. Вона забезпечує надійність, підтримку складних

запитів і можливість масштабування [15]. PostgreSQL дозволяє ефективно працювати зі структурованими даними, такими як інформація про користувачів, результати навчання, завдання та статистика прогресу.

Архітектура вебзастосунку побудована за моделлю MVC (Model-View-Controller), яка передбачає розділення системи на три основні компоненти. Модель (Model) відповідає за роботу з даними та бізнес-логіку застосунку. Представлення (View) реалізує інтерфейс користувача та відображення даних. Контролер (Controller) виступає посередником між моделлю та представленням, обробляючи запити користувача та керуючи взаємодією між компонентами.

Використання архітектури MVC дозволяє досягти чіткої структуризації коду, спрощує його підтримку та розширення. Розділення відповідальностей між компонентами зменшує складність системи та підвищує її гнучкість. Це особливо важливо для вебзастосунків, які передбачають подальший розвиток і масштабування.

Таким чином, обраний технологічний стек і архітектурний підхід забезпечують створення ефективного, масштабованого та зручного у використанні вебзастосунку для вивчення іноземних мов. Використання сучасних технологій дозволяє реалізувати інтерактивний інтерфейс, швидку обробку даних та адаптивність до потреб користувачів.

Висновки до розділу 1

У першому розділі дипломної роботи було розглянуто теоретичні аспекти розробки вебзастосунку для вивчення англійської мови. Проведений аналіз сучасних підходів до вивчення іноземних мов показав, що використання інформаційно-комунікаційних технологій суттєво підвищує ефективність навчального процесу за рахунок забезпечення доступності, інтерактивності та індивідуалізації навчання.

Було встановлено, що сучасні програмні засоби, зокрема мобільні застосунки, вебплатформи та мультимедійні ресурси, активно використовують інноваційні підходи, такі як гейміфікація, адаптивне навчання, мікронавчання та технології Web 2.0. Це дозволяє підвищити мотивацію користувачів, забезпечити регулярність навчання та створити умови для ефективного засвоєння мовного матеріалу.

У ході аналізу існуючих програмних засобів (Duolingo, Memrise, Busuu, Quizlet, BBC Learning English) було визначено їхні основні переваги та недоліки. Зокрема, встановлено, що більшість із них орієнтовані на окремі аспекти навчання (лексика, граматика або комунікація) і не забезпечують комплексного підходу до вивчення мови. Це підтверджує актуальність розробки власного вебзастосунку, який поєднуватиме переваги існуючих рішень та усуватиме їхні недоліки.

Також у розділі було обґрунтовано вибір технологічного стеку та архітектури вебзастосунку. Використання сучасних технологій, таких як React, MobX, Material UI для фронтенду, FastAPI для бекенду та PostgreSQL для роботи з базою даних, забезпечує створення продуктивного, масштабованого та зручного у використанні програмного продукту. Обрана архітектура MVC дозволяє ефективно організувати структуру застосунку та спростити його подальшу підтримку і розвиток.

Таким чином, результати проведеного аналізу створюють теоретичне підґрунтя для подальшого проєктування та реалізації вебзастосунку для вивчення іноземних мов, що буде розглянуто у наступних розділах дипломної роботи.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ

2.1 Проєктування інформаційної моделі та бази даних вебзастосунок

Проєктування інформаційної моделі є одним із ключових етапів розробки вебзастосунок, оскільки саме вона визначає структуру даних, логіку їх взаємодії та основу для реалізації функціональних можливостей системи. У контексті вебзастосунок для вивчення англійської мови інформаційна модель повинна забезпечувати ефективне зберігання даних користувача, організацію навчального процесу, облік результатів та підтримку механізмів адаптивного навчання.

Інформаційна модель розробленого вебзастосунок побудована навколо користувача як центрального елемента системи та процесу вивчення іноземної мови. Основна увага приділяється збереженню персональних налаштувань, відстеженню навчального прогресу та організації навчальних сесій.

Для визначення основних функціональних можливостей системи було побудовано діаграму прецедентів використання (див. рис. 2.1). Вона відображає взаємодію користувача із вебзастосунком та демонструє основні сценарії роботи системи, зокрема реєстрацію, авторизацію, проходження навчальних сесій, повторення матеріалу, перегляд статистики та керування налаштуваннями.

Модель даних побудована за принципом реляційної структури, що забезпечує цілісність, узгодженість та ефективність обробки інформації. Основними сутностями системи є користувач, навчальна сесія, навчальні елементи (речення) та прогрес їх засвоєння.

Інформаційна модель включає такі ключові сутності:

- користувач (User);
- навчальна сесія (LearningSession);

- елементи сесії (SessionSentence);
- прогрес користувача (UserSentenceProgress);
- стан вивчення слів (WordState).



Рис. 2.1. Діаграма прецедентів

Кожна з цих сутностей виконує окрему роль у системі та забезпечує реалізацію конкретних функціональних можливостей.

Сутність User є базовою у системі, оскільки всі інші об'єкти прямо або опосередковано пов'язані з користувачем. Вона зберігає основні дані облікового запису та налаштування навчання.

До складу цієї сутності входять:

- унікальний ідентифікатор користувача;

- електронна адреса;
- хеш пароля;
- мова інтерфейсу;
- рідна мова користувача;
- мова, що вивчається;
- кількість навчальних речень на день;
- дата створення облікового запису.

Використання хешування пароля забезпечує безпеку даних користувача та захист від несанкціонованого доступу. Електронна адреса має унікальне обмеження, що дозволяє уникнути дублювання облікових записів.

Збереження мовних налаштувань дозволяє адаптувати навчальний процес відповідно до потреб користувача, а параметр кількості речень на день використовується для формування навчального навантаження.

Навчальний процес у системі організовано у вигляді окремих сесій, що представлені сутністю `LearningSession`. Такий підхід дозволяє структурувати навчання, розділяти його на логічні етапи та зберігати інформацію про кожну взаємодію користувача з системою.

Основні атрибути сутності `LearningSession`..:

- ідентифікатор сесії;
- ідентифікатор користувача;
- тип сесії (навчання або повторення);
- дата створення;
- час початку;
- час завершення.

Наявність типу сесії дозволяє розрізняти первинне навчання та повторення матеріалу, що є важливим для реалізації механізму інтервального повторення. Збереження часових параметрів дає змогу аналізувати активність користувача та ефективність навчання.

Кожна навчальна сесія пов'язана з конкретним користувачем, що реалізує зв'язок типу «один до багатьох».

Сутність SessionSentence відповідає за збереження інформації про конкретні навчальні елементи, що використовуються у межах сесії. У даному випадку такими елементами є речення, які отримуються із зовнішнього джерела.

Основні поля SessionSentence:

- ідентифікатор запису;
- ідентифікатор сесії;
- зовнішній ідентифікатор речення (sentence_id);
- відповідь користувача;
- правильність відповіді;
- час відповіді;
- статус виконання.

Використання зовнішнього ідентифікатора дозволяє не дублювати дані речень у базі, а лише посилатися на них. Це зменшує обсяг бази даних і спрощує її підтримку. Ця сутність відіграє важливу роль у зборі статистики, оскільки дозволяє зберігати детальну інформацію про кожну відповідь користувача. Завдяки цьому можна аналізувати швидкість відповіді, точність та динаміку навчання.

Для збереження довгострокового прогресу користувача використовується сутність UserSentenceProgress. Вона містить інформацію про ступінь засвоєння кожного навчального елемента.

Основні атрибути UserSentenceProgress:

- ідентифікатор користувача;
- ідентифікатор речення;
- текст перекладу;
- індекси пропущених слів;
- статус засвоєння;

- кількість правильних відповідей;
- кількість помилок;
- дата додавання;
- дата завершення;
- дата наступного повторення.

Ця сутність є основою для реалізації адаптивного навчання. Вона дозволяє визначати, які речення вже засвоєні, а які потребують повторення. На основі цих даних формується індивідуальна навчальна програма.

Окрему роль у системі відіграє сутність WordState, яка відповідає за збереження інформації про вивчення окремих слів. Це дозволяє деталізувати навчальний процес і відстежувати прогрес не лише на рівні речень, а й на рівні лексичних одиниць.

Основні поля WordState:

- ідентифікатор користувача;
- слово;
- статус вивчення;
- кількість повторень;
- дата останнього оновлення.

Такий підхід дозволяє реалізувати більш точну систему навчання, яка враховує індивідуальні труднощі користувача.

Для візуалізації структури програмних сутностей та зв'язків між ними було побудовано діаграму класів (див. рис. 2.2). Вона відображає основні об'єкти системи, їх атрибути та асоціації, що використовуються під час реалізації бізнес-логіки вебзастосунку.

База даних застосунку реалізована з використанням SQLAlchemy - сучасної бібліотеки, що поєднує можливості SQLAlchemy для роботи з базами даних та Pydantic для валідації даних [21]. Це дозволяє забезпечити як гнучкість у роботі з БД, так і контроль коректності даних.

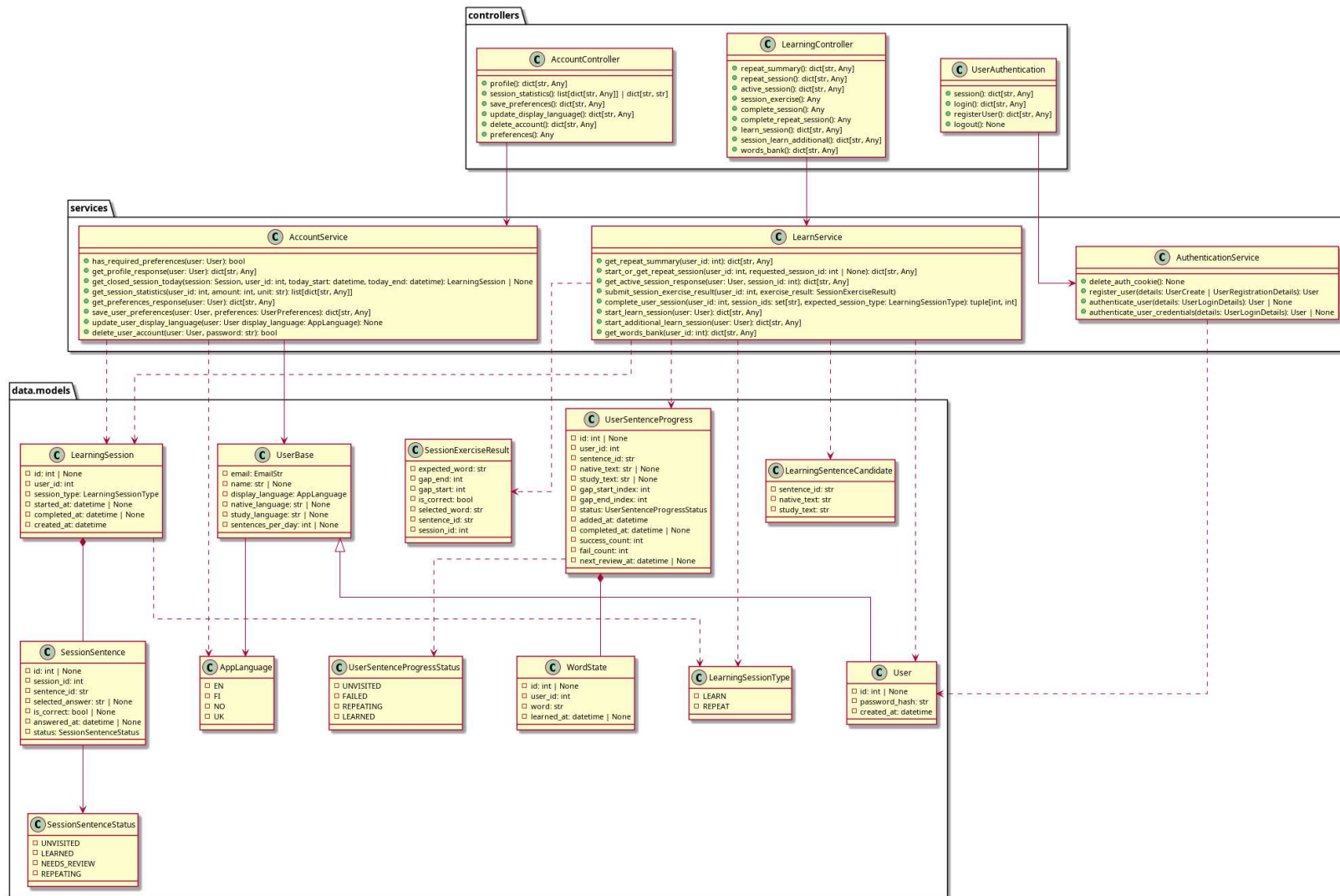


Рис. 2.2. Діаграма класів вебзастосунку

Підключення до бази даних здійснюється через змінну середовища `DATABASE_URL`, що дозволяє легко змінювати конфігурацію системи. Основною СУБД обрано PostgreSQL, яка забезпечує високу надійність, продуктивність та підтримку складних запитів.

Основні таблиці бази даних:

- `user` - зберігає дані користувачів;
- `learningsession` - містить інформацію про навчальні сесії;
- `sessionsentence` - зберігає відповіді користувача у сесіях;
- `usersentenceprogress` - містить прогрес користувача;
- `wordstate` - зберігає інформацію про вивчення слів.

Між таблицями реалізовано наступні зв'язки:

- один користувач має багато сесій;
- одна сесія містить багато речень;
- один користувач має багато записів прогресу;
- один користувач має багато записів стану слів.

Такі зв'язки забезпечують логічну узгодженість даних та дозволяють ефективно виконувати запити.

Для підвищення продуктивності бази даних використовуються індекси за основними полями:

- `user_id`;
- `sentence_id`;
- `word`;
- `created_at`.

Також передбачено унікальність email користувача.

Структуру бази даних та зв'язки між таблицями відображено на ER-діаграмі (див. рис. 2.3). Дана діаграма демонструє логічну організацію даних, первинні та зовнішні ключі, а також типи зв'язків між сутностями системи.

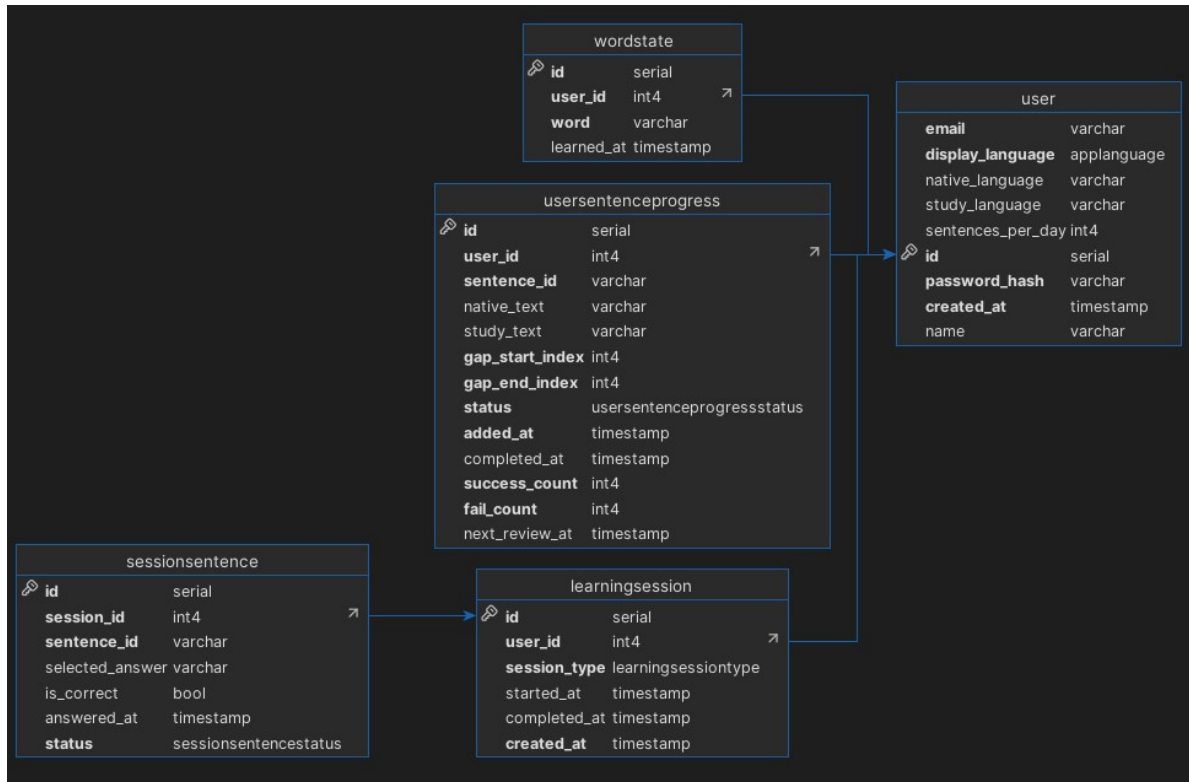


Рис. 2.3. ER діаграма вебзастосунку

Запропонована інформаційна модель має ряд переваг:

- забезпечує гнучкість і масштабованість системи;
- підтримує персоналізоване навчання;
- дозволяє реалізувати інтервальне повторення;
- забезпечує детальний аналіз навчального процесу;
- оптимізує зберігання даних.

2.2 Опис основних алгоритмів та методів обробки даних

Одним із ключових аспектів розробки вебзастосунку для вивчення англійської мови є реалізація алгоритмів обробки даних, які забезпечують коректну роботу системи, персоналізацію навчального процесу, контроль прогресу користувача та формування статистики. У сучасних освітніх вебсистемах алгоритми адаптивного навчання та аналізу прогресу є важливою складовою ефективного освітнього середовища.

У розробленому вебзастосунку реалізовано декілька основних алгоритмів, пов'язаних із автентифікацією користувача, створенням навчальних сесій, перевіркою відповідей, накопиченням прогресу та повторенням матеріалу.

Одним із базових механізмів системи є алгоритм реєстрації та авторизації користувача. Його основною метою є забезпечення безпечного доступу до функціоналу вебзастосунку та захист персональних даних.

Під час реєстрації користувач вводить електронну адресу, пароль та базові налаштування навчання. Перед збереженням дані проходять перевірку коректності (валідацію). Зокрема, система перевіряє правильність формату email, унікальність облікового запису та відповідність пароля встановленим вимогам.

Для забезпечення безпеки пароль не зберігається у відкритому вигляді. Перед записом до бази даних він проходить процедуру хешування [24]. Такий підхід відповідає сучасним рекомендаціям щодо безпечної розробки вебзастосунків та захисту персональних даних користувачів. У результаті в базі зберігається лише хеш пароля, що значно знижує ризик компрометації даних користувача у випадку несанкціонованого доступу до системи.

Після успішної авторизації користувачу видається токен доступу, який містить ідентифікатор користувача та термін дії сесії. Для цього часто використовується технологія JWT (JSON Web Token), що дозволяє реалізувати безпечну автентифікацію без необхідності повторного введення пароля під час кожного запиту до сервера.

JWT являє собою компактний токен у форматі JSON, який може безпечно передаватися між клієнтом і сервером. Основною перевагою даної технології є те, що серверу не потрібно зберігати інформацію про активну сесію користувача, оскільки всі необхідні дані містяться безпосередньо у токені. Це дозволяє зменшити навантаження на сервер та спрощує масштабування вебзастосунку.

Структурно JWT складається з трьох частин:

- заголовка (header);
- корисного навантаження (payload);
- цифрового підпису (signature).

У заголовку міститься інформація про тип токена та алгоритм шифрування. Корисне навантаження містить дані користувача, зокрема його ідентифікатор, роль або інші службові параметри. Цифровий підпис використовується для перевірки цілісності токена та захисту від підробки.

Після входу в систему токен передається клієнту та використовується для подальших запитів до API. При кожному запиті сервер перевіряє дійсність токена, його підпис і термін дії. Якщо токен є коректним, користувач отримує доступ до відповідного функціоналу системи.

Одним із центральних елементів системи є алгоритм формування навчальних сесій. Його основним завданням є створення персоналізованого набору навчального матеріалу відповідно до параметрів користувача.

Під час створення сесії система використовує такі дані:

- рідна мова користувача;
- мова, що вивчається;
- кількість речень на день;
- поточний прогрес навчання.

Для отримання навчального контенту використовується зовнішнє API сервісу Tatoeba Project, яке надає базу речень різними мовами [26]. Використання відкритих мовних корпусів є поширеним підходом у сучасних системах вивчення мов, оскільки це дозволяє працювати з автентичними прикладами мовлення.

Після отримання даних запускається етап фільтрації. На цьому етапі:

- видаляються дублікати;
- виключаються речення, які вже присутні у прогресі користувача;
- перевіряється коректність перекладів;
- відбираються речення відповідного рівня складності.

Після завершення фільтрації система формує навчальну сесію та створює записи у таблиці SessionSentence. Для кожного речення задається початковий статус виконання.

Такий підхід дозволяє уникнути повторення однакових завдань і забезпечує індивідуалізацію навчального процесу. Дослідження показують, що персоналізація навчального контенту позитивно впливає на мотивацію користувача та ефективність засвоєння матеріалу.

Під час проходження навчальних вправ система виконує обробку відповідей користувача. Основною метою цього алгоритму є перевірка правильності виконання завдання та збереження результатів.

Після введення відповіді система:

- Отримує правильний варіант.
- Порівнює його з відповіддю користувача.
- Визначає правильність виконання.
- Фіксує час відповіді.
- Оновлює статус завдання.

Інформація про відповідь записується до таблиці SessionSentence. Це дозволяє зберігати історію виконання вправ і надалі використовувати її для аналітики.

Також система враховує кількість помилок і правильних відповідей, що використовується для оцінки рівня засвоєння матеріалу. Подібні механізми широко застосовуються у сучасних системах електронного навчання для побудови адаптивних моделей навчання.

Для реалізації персоналізованого навчання у системі використовується механізм накопичення прогресу користувача. Він дозволяє зберігати інформацію про ступінь засвоєння кожного речення та окремих слів.

Після завершення вправи система оновлює запис у таблиці UserSentenceProgress:

- збільшує кількість правильних або помилкових відповідей;

- змінює статус вивчення;
- оновлює дату останнього проходження;
- встановлює дату наступного повторення.

Якщо користувач стабільно дає правильні відповіді, речення може бути переведене у статус «вивчено». У разі помилок система продовжує включати матеріал до наступних навчальних сесій.

Окремо ведеться облік стану окремих слів у таблиці WordState. Це дозволяє визначати слова, які викликають труднощі, та приділяти їм додаткову увагу. Аналіз індивідуального прогресу є важливою складовою адаптивних освітніх систем, оскільки дозволяє враховувати особливості кожного користувача.

Однією з важливих функцій вебзастосунку є реалізація механізму інтервального повторення (spaced repetition). Даний підхід базується на принципі повторення інформації через певні проміжки часу для покращення довготривалого запам'ятовування.

Концепція інтервального повторення базується на дослідженнях Hermann Ebbinghaus, який у своїх роботах досліджував процеси запам'ятовування та забування інформації. У результаті експериментів ним була сформульована так звана «крива забування» - модель, що демонструє швидке зниження рівня запам'ятовування інформації з часом за відсутності повторення.

Дослідження показали, що найбільший обсяг інформації людина втрачає протягом перших годин або днів після вивчення матеріалу. Надалі швидкість забування поступово зменшується. Водночас Еббінгауз встановив, що повторення інформації у певні моменти часу дозволяє значно сповільнити процес забування та перевести знання у довготривалу пам'ять.

На основі цих досліджень був сформований метод інтервального повторення, який передбачає повторення навчального матеріалу через поступово збільшувані проміжки часу. Після першого успішного засвоєння матеріал повторюється через короткий інтервал, наприклад через один день.

Якщо користувач знову правильно відтворює інформацію, наступний інтервал збільшується - до декількох днів, тижнів або навіть місяців.

У випадку помилки інтервал повторення скорочується або залишається без змін, що дозволяє приділяти більше уваги складному матеріалу. Такий механізм забезпечує оптимальне навантаження на пам'ять користувача та підвищує ефективність засвоєння інформації.

Сучасні цифрові платформи для вивчення мов активно використовують даний підхід, оскільки він демонструє високу ефективність при засвоєнні нової лексики та мовних конструкцій. Інтервальне повторення дозволяє не лише покращити короткочасне запам'ятовування, а й сприяє формуванню стійких довготривалих знань.

У розробленому вебзастосунку цей принцип використовується для автоматичного визначення часу повторення речень та слів залежно від успішності користувача. Завдяки цьому система адаптується до індивідуального темпу навчання та дозволяє ефективніше закріплювати вивчений матеріал.

Методи формування статистики. У вебзастосунку також реалізовано механізми збору та аналізу статистичних даних. Основною метою є надання користувачу інформації про результати навчання та рівень прогресу.

Система виконує групування даних за різними часовими періодами:

- день;
- тиждень;
- місяць;
- рік.

На основі накопичених даних обчислюються:

- кількість вивчених речень;
- кількість повторених речень;
- кількість вивчених слів;
- відсоток правильних відповідей;

- середній час відповіді;
- активність користувача.

Отримані результати можуть використовуватися для побудови графіків та візуалізації прогресу навчання. Візуалізація статистики позитивно впливає на мотивацію користувачів та допомагає контролювати ефективність навчання.

2.3 Реалізація серверної та клієнтської частин вебзастосунку

Реалізація вебзастосунку для вивчення англійської мови передбачає створення двох основних складових системи - серверної (backend) та клієнтської (frontend) частин. Такий підхід дозволяє розділити логіку обробки даних і користувацький інтерфейс, що спрощує підтримку, масштабування та подальший розвиток програмного продукту.

Серверна частина вебзастосунку реалізована мовою програмування Python із використанням фреймворку FastAPI. Вибір даного фреймворку обумовлений його високою продуктивністю, підтримкою асинхронної обробки запитів та зручними засобами для створення REST API.

FastAPI дозволяє швидко створювати HTTP-маршрути, обробляти запити клієнта та повертати дані у форматі JSON. Крім того, фреймворк автоматично генерує документацію API, що значно спрощує тестування та взаємодію між серверною і клієнтською частинами застосунку.

Архітектура backend-частини поділена на декілька логічних рівнів, що відповідає принципам багатошарової архітектури. Такий підхід забезпечує кращу структурованість коду та спрощує його підтримку.

У папці controllers розміщено контролери, які відповідають за обробку API-маршрутів. Вони приймають HTTP-запити від клієнта, виконують базову перевірку даних та передають їх до рівня бізнес-логіки. Основні маршрути реалізують:

- реєстрацію та авторизацію користувачів;

- роботу з обліковим записом;
- створення навчальних сесій;
- отримання статистики;
- перевірку відповідей;
- оновлення прогресу навчання.

Основна бізнес-логіка системи винесена у папку `services`. Тут реалізовано алгоритми реєстрації, авторизації, формування навчальних сесій, обробки відповідей, інтервального повторення та збору статистики. Відокремлення бізнес-логіки від контролерів дозволяє зробити систему більш модульною та зручною для тестування.

Для роботи з базою даних використовується бібліотека `SQLModel`, яка поєднує можливості `SQLAlchemy` та `Pydantic` [22]. Використання ORM (Object Relational Mapping) дозволяє працювати з таблицями бази даних через об'єкти Python, що спрощує взаємодію з даними та зменшує кількість SQL-запитів у коді.

Моделі даних описують основні сутності системи:

- користувачів;
- навчальні сесії;
- речення у сесіях;
- прогрес користувача;
- вивчені слова.

Підключення до бази даних налаштовується через змінну середовища `DATABASE_URL`, що забезпечує гнучкість конфігурації застосунку. Основною системою керування базами даних обрано PostgreSQL, оскільки вона забезпечує високу продуктивність, підтримку складних запитів і надійне зберігання інформації.

Однією з важливих складових серверної частини є механізм автентифікації користувача. Паролі не зберігаються у відкритому вигляді - перед записом до бази даних вони проходять процедуру хешування. Це

забезпечує захист персональних даних користувача та відповідає сучасним вимогам безпеки вебзастосунків.

Після успішного входу в систему користувач отримує токен доступу, який передається через cookie. Для цього використовується технологія JWT (JSON Web Token), яка дозволяє реалізувати безпечну авторизацію без необхідності зберігання серверної сесії. Захищені API-маршрути перед виконанням запиту перевіряють наявність і валідність токена (див. рис. 2.4).

```
def create_auth_token(user: User, expires_at: datetime) -> str:
    header = _json_base64url_encode({"alg": "HS256", "typ": "JWT"})
    payload = _json_base64url_encode(
        {
            "sub": str(user.id),
            "email": user.email,
            "exp": int(expires_at.timestamp()),
        }
    )
    unsigned_token = f"{header}.{payload}"
    signature = hmac.new(
        AUTH_TOKEN_SECRET.encode("utf-8"),
        unsigned_token.encode("ascii"),
        hashlib.sha256,
    ).digest()
    return f"{unsigned_token}.{_base64url_encode(signature)}"
```

Рис. 2.4. Фрагмент програмного коду, що відповідає за реалізацію JWT-автентифікації

Також серверна частина взаємодіє із зовнішнім API сервісу Tatoes Project для отримання речень різними мовами. Отримані дані проходять фільтрацію, після чого використовуються для формування персоналізованих навчальних сесій (див. рис. 2.5). Повний код файлу authentication.py, в якості прикладу, наведений в додатку А.

Серверна частина також реалізує механізми:

- обробки статистики;
- інтервального повторення;

- накопичення прогресу;
- збереження історії навчання;
- формування рекомендацій для повторення матеріалу.

```
def _collect_learning_sentences(
    user: User,
    target_count: int | None = None,
    excluded_sentence_ids: set[str] | None = None,
) -> list[LearningSentenceCandidate] | None:
    assert user.id is not None
    assert user.native_language is not None
    assert user.study_language is not None
    assert user.sentences_per_day is not None

    target_count = user.sentences_per_day if target_count is None else target_count
    if target_count <= 0:
        return []

    sentences: list[LearningSentenceCandidate] = []
    sentence_ids: set[str] = set(excluded_sentence_ids or set())
    failed_request_count = 0
    next_url: str | None = _build_tatoeba_url(
        native_language=user.native_language,
        study_language=user.study_language,
        limit=target_count,
    )
    request_count = 0

    logger.debug(...)

    while len(sentences) < target_count and failed_request_count < MAX_FAILED_SENTENCE_REQUESTS:
        if next_url is None:
            failed_request_count += 1
            logger.debug(...)
            next_url = _build_tatoeba_url(
                native_language=user.native_language,
                study_language=user.study_language,
                limit=target_count - len(sentences),
            )
            logger.debug("Restarting Tatoeba search from first page: url=%s", next_url)
            continue

        request_count += 1
        logger.debug(...)
        payload = _request_tatoeba_page(next_url)
        if payload is None:
            failed_request_count += 1
            logger.debug(...)
            next_url = _build_tatoeba_url(
                native_language=user.native_language,
                study_language=user.study_language,
                limit=target_count - len(sentences),
            )
            logger.debug("Retrying Tatoeba search from first page after failed payload: url=%s", next_url)
            continue
```

Рис. 2.5. Фрагмент коду, що відповідає за реалізацію алгоритму формування навчальної сесії

Таким чином, backend забезпечує повний цикл обробки даних: від отримання запиту користувача до збереження результатів навчання та формування персоналізованого контенту.

Клієнтська частина вебзастосунку реалізована за допомогою бібліотеки React та мови програмування TypeScript. Використання TypeScript дозволяє додатково типізувати дані та компоненти, що підвищує надійність коду, спрощує пошук помилок і покращує підтримку проєкту під час подальшої розробки.

Для створення користувацького інтерфейсу використовується бібліотека Material UI, яка надає готові компоненти відповідно до концепції Material Design. Завдяки цьому реалізовано сучасний, адаптивний та зручний інтерфейс користувача. Material UI дозволяє швидко створювати форми, кнопки, діалогові вікна, повідомлення, навігаційні панелі та інші елементи інтерфейсу без необхідності їх реалізації з нуля.

Вебзастосунок побудований за принципом SPA (Single Page Application), тобто односторінкового застосунку. У такій архітектурі перехід між сторінками виконується без повного перезавантаження сайту, що значно покращує швидкість роботи та забезпечує більш комфортну взаємодію користувача із системою.

Для реалізації маршрутизації використовується React Router. Це дозволяє організувати навігацію між різними сторінками застосунку та підтримує динамічну зміну контенту без оновлення сторінки у браузері.

У клієнтській частині реалізовано такі основні сторінки:

- сторінка реєстрації;
- сторінка входу;
- головна сторінка;
- профіль користувача;
- сторінка налаштувань;
- сторінка навчальної сесії;

- сторінка повторення матеріалу.

Кожна сторінка представлена окремим React-компонентом, що забезпечує модульність структури застосунку та повторне використання елементів інтерфейсу (див. рис. 2.6).

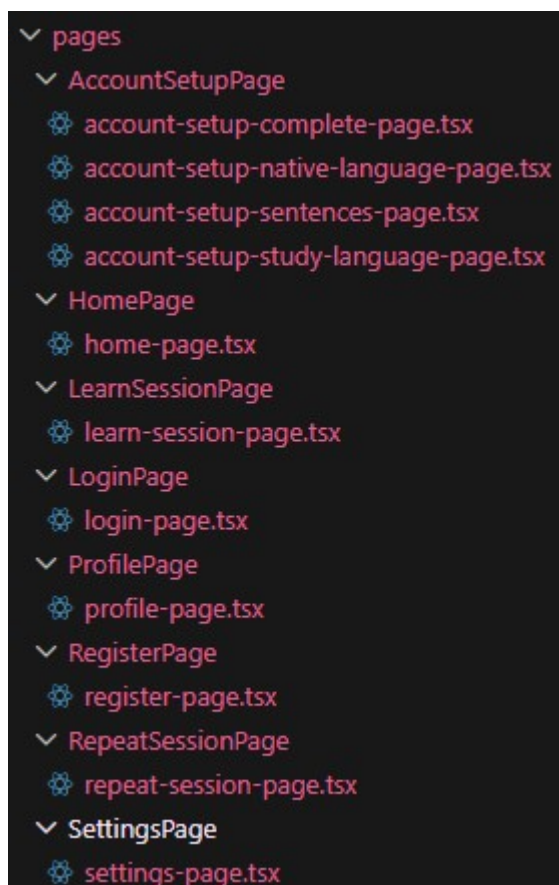


Рис. 2.6. Структура React-компонентів сторінок вебзастосунку

Для керування станом застосунку використовується бібліотека MobX. Вона забезпечує централізоване зберігання даних і автоматичне оновлення інтерфейсу при зміні стану. Завдяки реактивному підходу MobX значно спрощується синхронізація даних між різними компонентами застосунку.

У системі реалізовано декілька окремих сховищ, кожне з яких відповідає за певну частину функціональності:

- авторизацію користувача;
- налаштування облікового запису;

- навчальні сесії;
- повторення матеріалу;
- список вивчених слів;
- систему повідомлень;
- блокування інтерфейсу під час завантаження даних.

Такий підхід дозволяє розділити логіку роботи застосунку на окремі незалежні модулі та зробити код більш структурованим і підтримуваним.

Взаємодія клієнтської частини з сервером здійснюється через HTTP-запити до REST API. Для цього реалізовано окремі асинхронні процедури, які відповідають за:

- надсилання даних реєстрації;
- авторизацію користувача;
- оновлення налаштувань;
- відправлення відповідей у вправах;
- отримання профілю користувача;
- завантаження статистики;
- отримання навчальних сесій.

Обмін даними між frontend та backend здійснюється у форматі JSON. Авторизаційні cookie автоматично передаються разом із запитом, що дозволяє підтримувати сесію користувача без необхідності повторної авторизації.

У клієнтській частині також реалізовано механізми валідації форм. Перед відправленням даних система перевіряє правильність введених значень, наприклад:

- коректність email;
- довжину пароля;
- заповнення обов'язкових полів;
- допустимі значення налаштувань.

Це дозволяє зменшити кількість помилкових запитів до сервера та покращує користувацький досвід.

Окрему увагу необхідно приділити обробці помилок. У разі виникнення помилки під час виконання запиту система відображає користувачу відповідне повідомлення. Для цього реалізовано окремий механізм повідомлень та сповіщень, який використовується для:

- повідомлень про успішні дії;
- попереджень;
- помилок авторизації;
- проблем із мережею;
- інформаційних повідомлень (див. рис. 2.7).

```
class NotificationService implements INotificationService {
  @observable private queue: INotificationMessage[] = [];

  private nextId = 1;

  constructor() {
    makeObservable(this);
  }

  @computed
  get currentNotification(): INotificationMessage | null {
    return this.queue[0] ?? null;
  }

  @action.bound
  push(message: string, status: NotificationStatus) {
    this.queue.push({
      id: this.nextId,
      message,
      status,
    });
    this.nextId += 1;
  }

  @action.bound
  dismissCurrent() {
    if (this.queue.length === 0) {
      return;
    }

    this.queue.shift();
  }
}

export default NotificationService;
```

Рис. 2.7. Фрагмент коду, що відповідає за реалізацію механізму повідомлень користувача

Також у застосунку реалізована підтримка декількох мов інтерфейсу (див. рис. 2.8). Це дозволяє користувачу обирати зручну мову відображення елементів системи та покращує доступність вебзастосунку для ширшої аудиторії [30].

```
class I18nService {
  @observable language: ApLanguage = resolveInitialLanguage();

  private catalogs: Partial<Record<ApLanguage, TranslationCatalog>> = {};

  constructor() {
    makeObservable(this);
  }

  initialize = async () => {
    const loadedCatalogs = await Promise.all(
      apLanguages.map(async (language) => [language, await loadTranslationCatalog(language)] as const),
    );

    loadedCatalogs.forEach(([language, catalog]) => {
      this.catalogs[language] = catalog;
    });
  };

  @action.bound
  setLanguage(nextLanguage: ApLanguage) {
    this.language = nextLanguage;

    if (typeof window !== 'undefined') {
      window.localStorage.setItem(LANGUAGE_STORAGE_KEY, nextLanguage);
    }
  }

  t = (key: TranslationKey, params?: TranslationParams): string => {
    const template =
      this.catalogs[this.language]?.[key] ?? this.catalogs[defaultLanguage]?.[key] ?? key;

    return applyTranslationParams(template, params);
  };
}

export const i18nService = new I18nService();
```

Рис. 2.8. Фрагмент коду, що відповідає за реалізацію підтримки багатомовного інтерфейсу вебзастосунку

Для тестування frontend-частини без необхідності запуску реального сервера реалізовано mock-режим роботи API.

У цьому режимі застосунок використовує тестові дані та імітує відповіді сервера. Такий підхід дозволяє:

- тестувати інтерфейс незалежно від backend;
- спрощувати розробку окремих компонентів;
- перевіряти поведінку застосунку при різних сценаріях роботи.

Крім того, у клієнтській частині реалізовано адаптивний дизайн, що забезпечує коректне відображення інтерфейсу на різних пристроях - персональних комп'ютерах, планшетах і смартфонах.

Таким чином, клієнтська частина вебзастосунку забезпечує повноцінну взаємодію користувача з системою, відображення навчального процесу, введення та обробку даних, отримання інформації з сервера та формування зручного інтерактивного інтерфейсу для вивчення англійської мови.

Висновки до розділу 2

У другому розділі було розглянуто процес проєктування та розробки вебзастосунку для вивчення англійської мови. У ході роботи спроектовано інформаційну модель системи та структуру бази даних, які забезпечують збереження даних користувачів, навчальних сесій, прогресу навчання та результатів повторення матеріалу. Розроблена структура бази даних дозволяє реалізувати персоналізований підхід до навчання, накопичення статистики та механізми адаптивного повторення.

Проаналізовано та описано основні алгоритми й методи обробки даних, що використовуються у вебзастосунку. Зокрема, реалізовано алгоритми реєстрації та автентифікації користувачів, формування навчальних сесій, перевірки відповідей, накопичення прогресу та інтервального повторення матеріалу. Використання механізму spaced repetition дозволяє підвищити ефективність запам'ятовування інформації та забезпечити більш результативне вивчення лексики й мовних конструкцій.

У процесі розробки серверної частини застосунку було використано мову програмування Python та фреймворк FastAPI. Серверна частина забезпечує обробку HTTP-запитів, взаємодію з базою даних, авторизацію користувачів,

роботу з навчальними сесіями та формування статистики навчання. Для збереження даних використовується PostgreSQL, а доступ до даних реалізовано за допомогою SQLModel.

Клієнтська частина вебзастосунку реалізована з використанням React та TypeScript. Для створення інтерфейсу використано бібліотеку Material UI, а для керування станом застосунку - MobX. Реалізовано адаптивний інтерфейс, систему маршрутизації, підтримку багатомовності, валідацію форм та взаємодію з REST API.

Таким чином створено архітектурну основу сучасного вебзастосунку для вивчення англійської мови, який забезпечує персоналізоване навчання, контроль прогресу користувача та інтерактивну взаємодію з системою. Отримані результати підтверджують доцільність використання сучасних вебтехнологій для створення ефективних освітніх платформ.

РОЗДІЛ 3

ФУНКЦІОНУВАННЯ ТА ЕЛЕМЕНТИ ІНТЕРФЕЙСУ ВЕБЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ

3.1 Огляд функціоналу розробленого вебзастосунок

Розроблений вебзастосунок призначений для вивчення англійської мови з використанням персоналізованих навчальних сесій, механізмів повторення матеріалу та системи відстеження прогресу користувача. Основною метою застосунок є забезпечення зручного та інтерактивного середовища для регулярного вивчення іноземної мови.

Функціональні можливості системи охоплюють процес реєстрації та авторизації користувачів, налаштування параметрів навчання, проходження навчальних сесій, повторення матеріалу, перегляд статистики та керування профілем користувача.

Однією з базових функцій вебзастосунку є створення облікового запису користувача та подальша авторизація у системі. Під час реєстрації користувач вводить електронну адресу, пароль, рідну мову, мову вивчення та кількість речень для щоденного навчання.

Після успішної реєстрації користувач може увійти до системи за допомогою email та пароля. Для забезпечення безпеки даних використовується механізм автентифікації з JWT-токенами та захищеними cookie.

Після входу в систему користувач отримує доступ до всіх функцій вебзастосунку та персональних навчальних даних.

Головна сторінка є центральним елементом взаємодії користувача із системою (див. рис. 3.1). На ній можна швидко оцінити стан навчання, почати новий урок або перевірити готовність повторення.

Інтерфейс головної сторінки побудований таким чином, щоб користувач міг швидко отримати доступ до основних функцій навчального процесу. На сторінці відображається загальна інформація про поточні налаштування

навчання, зокрема рідна мова, мова, що вивчається, та встановлена денна ціль. Також користувач може переглянути кількість речень, що очікують повторення, наявність активної навчальної сесії та поточний стан навчального процесу.

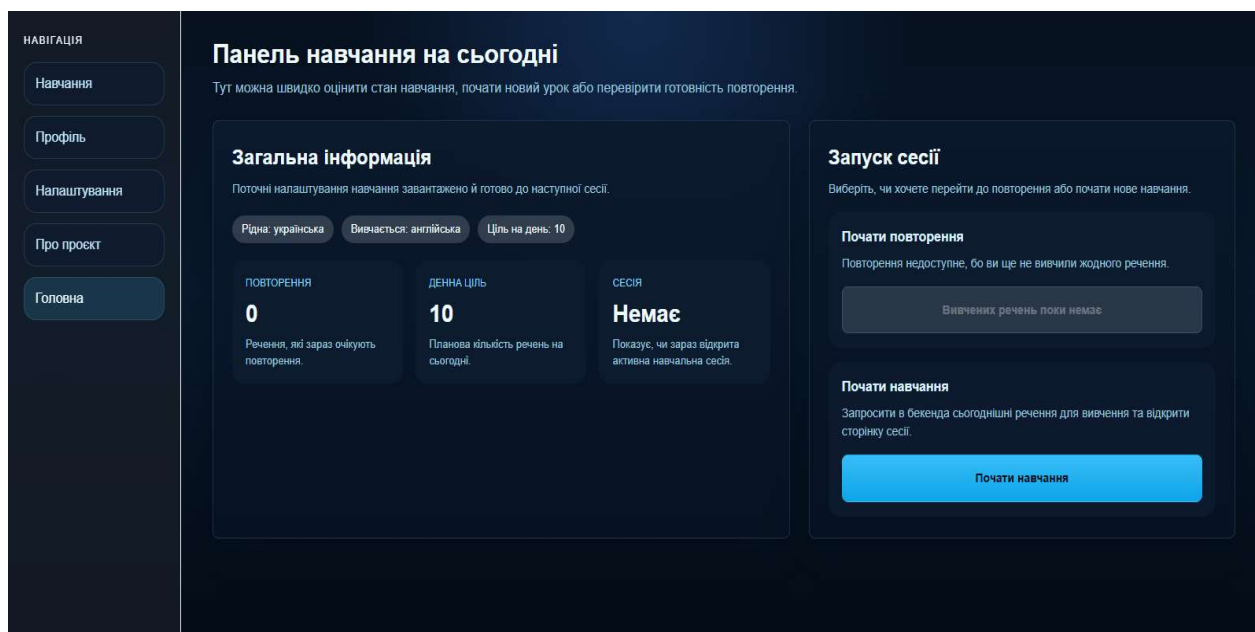


Рис. 3.1. Головна сторінка вебзастосунку

Окремий блок призначений для запуску навчання та повторення матеріалу. З його допомогою користувач може:

- розпочати нову навчальну сесію;
- перейти до сесії повторення вивченого матеріалу (за наявності доступних речень для повторення).

Для навігації між іншими розділами вебзастосунку використовується бічне меню, яке забезпечує швидкий перехід до сторінок навчання, профілю користувача, налаштувань, інформації про проєкт та головної сторінки.

Формування навчальних сесій. Однією з основних функцій системи є автоматичне формування навчальних сесій. Під час створення сесії застосунок враховує:

- рідну мову користувача;
- мову, що вивчається;

- кількість речень на день;
- речення, які вже були вивчені раніше.

Для отримання навчального матеріалу система взаємодіє з сервісом Tatoeba, який надає приклади речень різними мовами.

Після отримання даних система виконує фільтрацію речень, видаляє дублікати та формує персоналізований набір вправ для користувача.

Основною частиною навчального процесу є проходження вправ. Користувачу демонструється речення рідною мовою та речення мовою яку користувач обрав для вивчення при створенні облікового запису із пропуском (див. рис. 3.2).

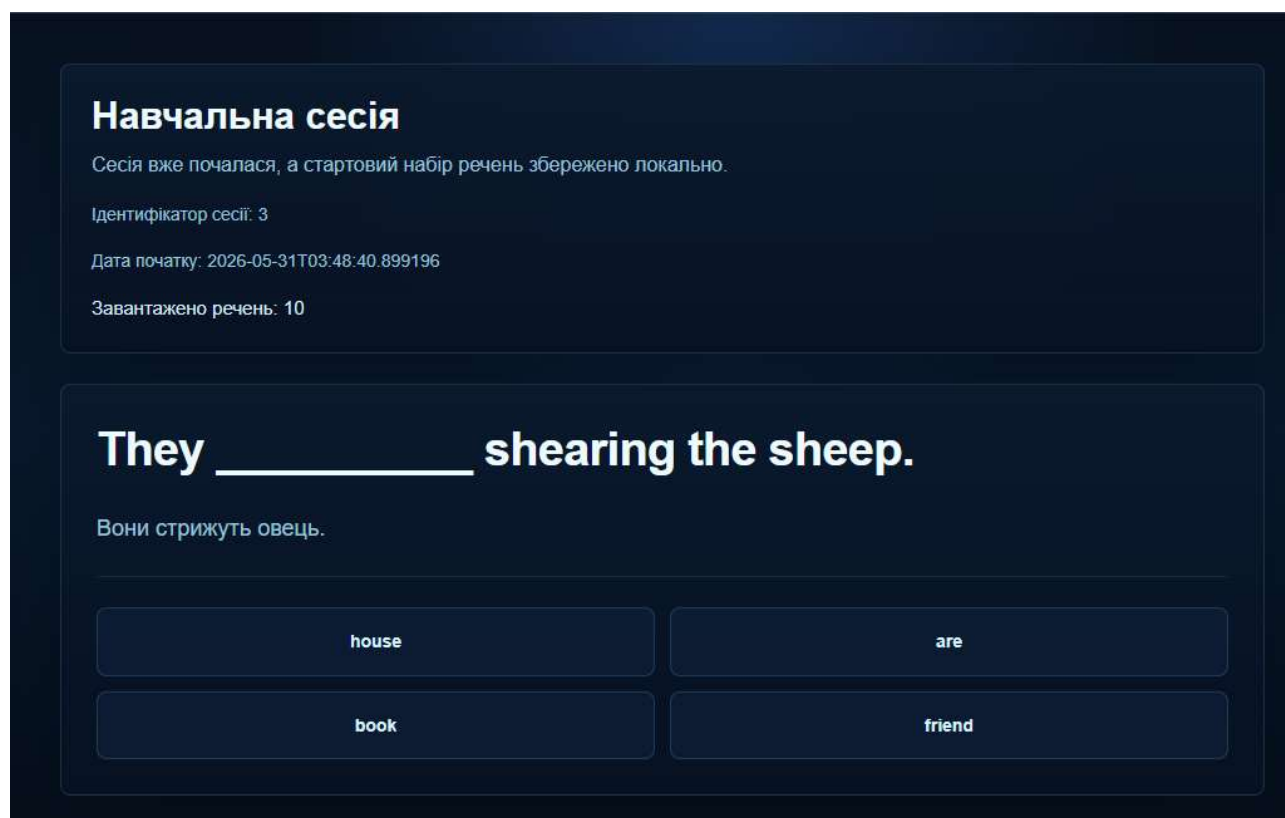


Рис. 3.2. Вигляд вправи навчальної сесії

Пропонується обрати слово, яке повинно бути на місці пропуску.

Після введення відповіді система:

- перевіряє правильність виконання;
- оновлює статус речення;

– зберігає результат у базі даних.

У разі правильної відповіді вона виділяється зеленим кольором, а речення відправляється до списку речень на повторення (див. рис 3.3).

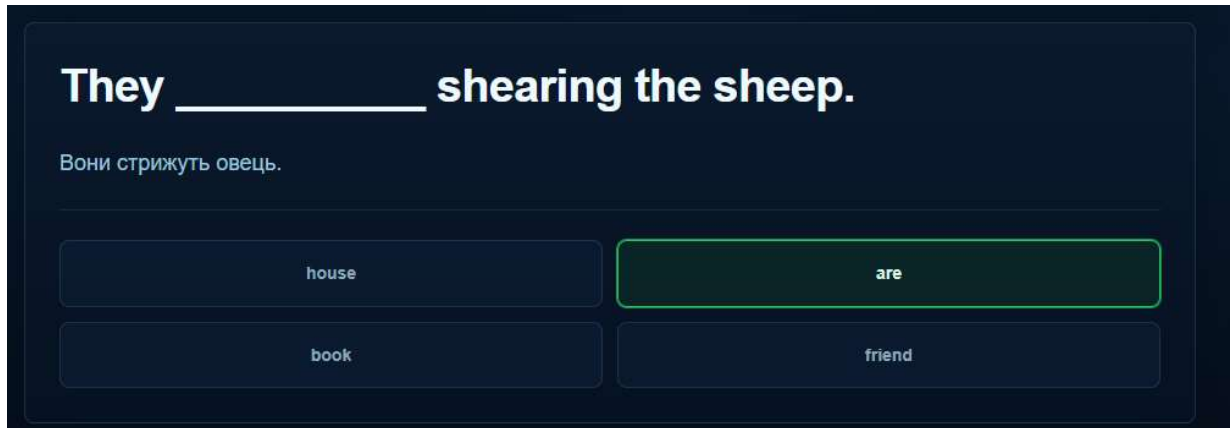


Рис. 3.3. Вигляд вебзастосунку у разі правильної відповіді користувача

Якщо користувач допускає помилку, варіант відповіді виділяється червоним, а речення буде показано повторно в кінці сесії та додано у список речень на повторення (див. рис 3.4).



Рис. 3.4. Вигляд вебзастосунку у разі неправильної відповіді користувача

Такий підхід дозволяє адаптувати навчальний процес до рівня знань користувача та забезпечує поступове закріплення матеріалу.

Також, у вебзастосунку реалізовано механізм повторення вивченого матеріалу на основі принципу інтервального повторення. Система автоматично визначає, які речення необхідно повторити, та формує окремі сесії повторення.

Частота повторення залежить від:

- кількості правильних відповідей;
- кількості помилок;
- часу останнього проходження.

Якщо користувач успішно відповідає на питання, інтервал повторення збільшується. У випадку помилки система частіше повертає матеріал до повторення. Реалізація такого механізму дозволяє ефективніше закріплювати нову лексику та мовні конструкції.

Крім того, у вебзастосунку реалізовано сторінку налаштувань, яка дозволяє користувачу керувати основними параметрами роботи системи та персоналізувати інтерфейс відповідно до власних уподобань (див. рис 3.5).

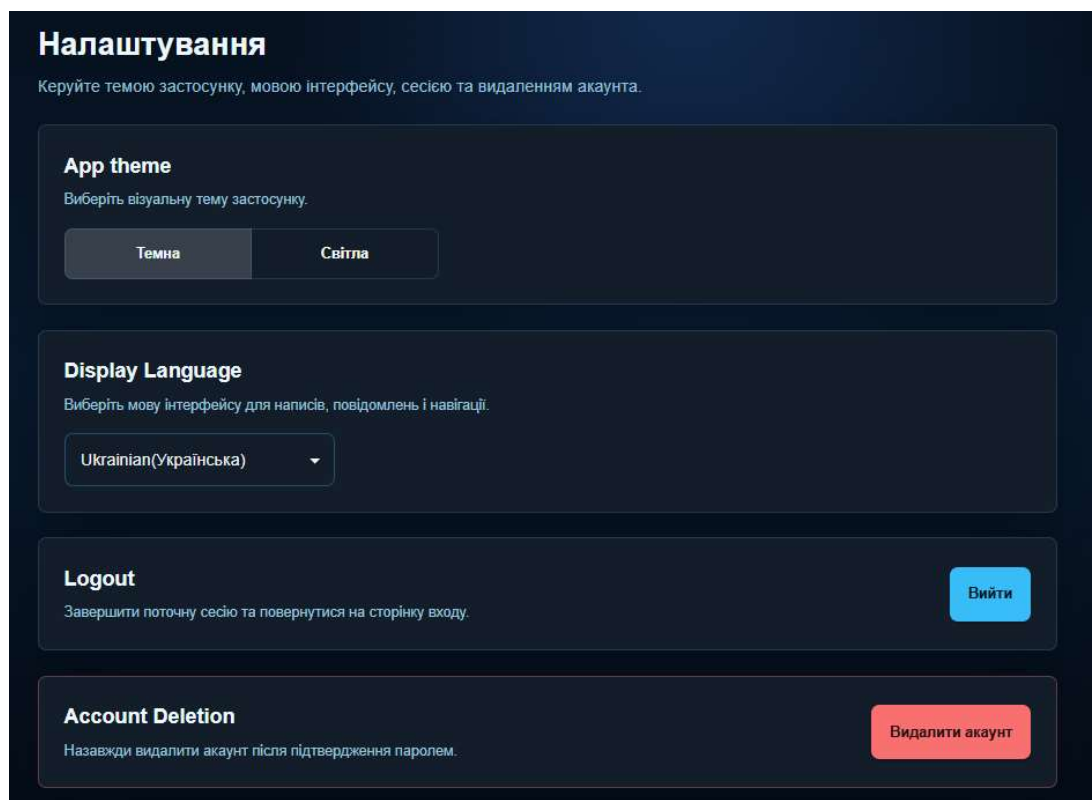


Рис. 3.5 Сторінка налаштувань

На сторінці налаштувань користувач може:

- обирати тему оформлення застосунку (світлу або темну);
- змінювати мову інтерфейсу;
- завершувати поточну сесію та виконувати вихід із системи;
- видаляти власний обліковий запис після підтвердження відповідної дії.

Функція вибору теми оформлення дозволяє адаптувати зовнішній вигляд вебзастосунку до індивідуальних потреб користувача та забезпечити комфортну роботу в різних умовах освітлення. Зміна мови інтерфейсу забезпечує підтримку локалізації системи та підвищує зручність використання для користувачів з різних мовних середовищ.

Крім того, сторінка налаштувань містить засоби керування обліковим записом. Користувач може завершити поточну авторизовану сесію за допомогою функції виходу із системи або повністю видалити свій обліковий запис разом із пов'язаними даними. Такий підхід забезпечує додатковий контроль над персональними даними та відповідає сучасним вимогам щодо конфіденційності інформації.

Сторінка профілю користувача. Для перегляду персональної інформації та результатів навчання у вебзастосунку реалізовано сторінку профілю користувача (див. рис. 3.6). Вона об'єднує дані облікового запису, мовні налаштування та статистичну інформацію щодо навчального процесу.

У лівій частині сторінки відображається основна інформація про користувача, зокрема його ім'я, електронна адреса, рідна мова та мова навчання. Також користувач може переглядати загальну кількість вивчених слів та кількість речень, які наразі очікують повторення. Це дозволяє швидко оцінити власний прогрес без необхідності переходу до інших розділів системи.

У правій частині сторінки розміщено блок статистики навчальних сесій. Тут відображаються узагальнені показники навчальної активності, серед яких кількість вивчених речень, повторених речень, запам'ятованих слів та

завершених навчальних сесій. Для більш детального аналізу користувач може обирати різні часові проміжки перегляду статистики: за останній тиждень, місяць, три місяці, шість місяців або рік.

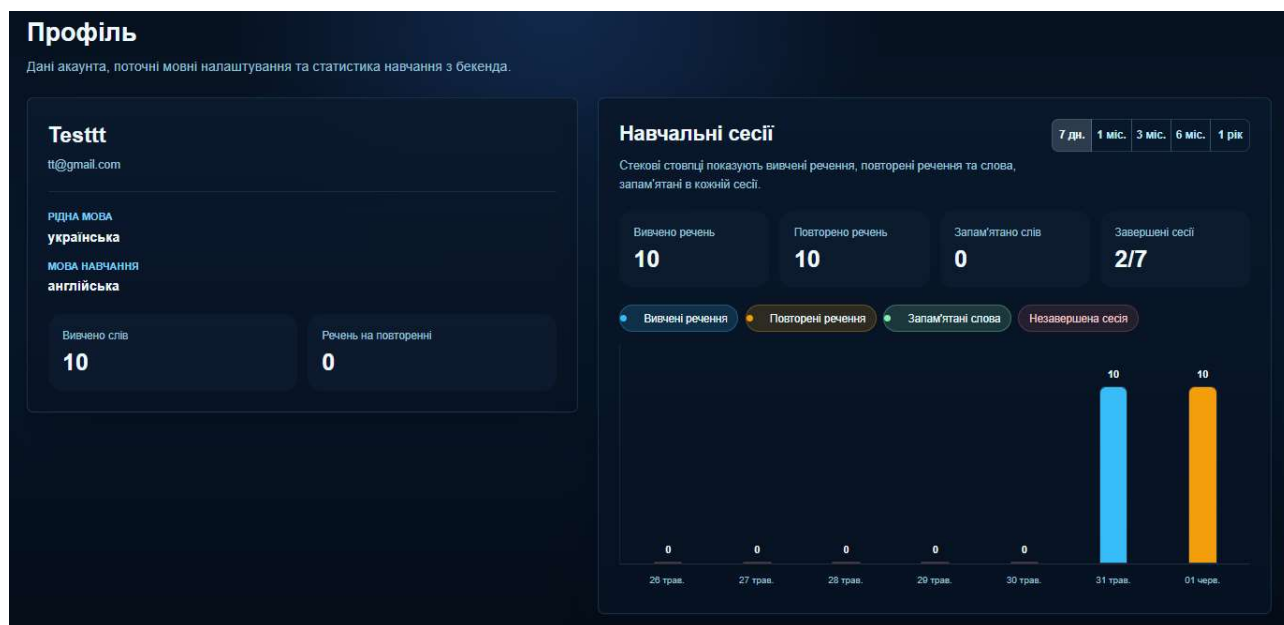


Рис. 3.6. Сторінка профілю користувача

Інтерфейс вебзастосунку розроблений з врахуванням принципів адаптивного дизайну. Завдяки використанню Material UI система коректно відображається на різних типах пристроїв:

- персональних комп'ютерах;
- планшетах;
- смартфонах.

Крім того, інтерфейс побудований таким чином, щоб мінімізувати кількість зайвих дій користувача та забезпечити швидкий доступ до основних функцій системи.

3.2 Опис взаємодії користувача з розробленим вебзастосунком

Взаємодія користувача з вебзастосунком для вивчення англійської мови побудована таким чином, щоб забезпечити простий, зрозумілий та зручний

процес навчання. Інтерфейс системи орієнтований на мінімізацію складних дій та швидкий доступ до основного функціоналу застосунку. Усі основні можливості системи доступні через графічний інтерфейс, реалізований за допомогою React та Material UI.

Першим етапом взаємодії із системою є створення облікового запису. Для цього користувач переходить на сторінку реєстрації та вводить необхідні дані:

- ім'я
- електронну адресу;
- пароль;
- рідну мову;
- мову, що вивчається;
- бажану кількість речень для щоденного навчання.

Після введення даних система виконує перевірку правильності заповнення форми. У випадку помилок користувач отримує відповідні повідомлення з поясненням проблеми. Якщо дані введені коректно, створюється новий обліковий запис і користувач отримує доступ до функціоналу вебзастосунку.

Після реєстрації користувач може виконати вхід до системи за допомогою електронної адреси та пароля. Після успішної авторизації сервер генерує JWT-токен, який використовується для підтвердження особи користувача під час подальшої роботи з застосунком.

Після входу користувач потрапляє на головну сторінку застосунку. Вона є центральним елементом взаємодії з системою та містить основну інформацію про навчальний процес.

Однією з основних форм взаємодії користувача із системою є проходження навчальних вправ. Після запуску сесії застосунком автоматично формує набір речень для навчання.

Користувачу відображається речення, після чого він обирає один із доступних варіантів.

Після кожної відповіді система миттєво перевіряє результат та повідомляє користувача про правильність виконання. У разі помилки відображається правильний варіант відповіді.

Такий формат взаємодії забезпечує інтерактивність навчального процесу та дозволяє користувачу швидко отримувати зворотний зв'язок.

Окремим режимом роботи застосунку є повторення матеріалу. Система автоматично формує список речень, які необхідно повторити відповідно до алгоритму інтервального повторення.

Користувач проходить вправи аналогічно до звичайної навчальної сесії, однак акцент робиться на закріпленні вже вивченого матеріалу.

Після успішного проходження повторення система:

- оновлює прогрес користувача;
- змінює дату наступного повторення;
- збільшує інтервал повторення.

Якщо користувач допускає помилки, система частіше повертає матеріал до повторного вивчення.

Для контролю результатів навчання у вебзастосунку реалізовано сторінку статистики. Вона дозволяє користувачу аналізувати власний прогрес та оцінювати ефективність навчання. Інформація представлена у вигляді графіку, що спрощує сприйняття статистичних даних.

3.3 Аналіз ефективності розробленого вебзастосунку та можливих обмежень в його використанні

Оцінка ефективності вебзастосунку є важливим етапом аналізу результатів розробки програмного продукту. Вона дозволяє визначити, наскільки створена система відповідає поставленим цілям, забезпечує зручність використання та сприяє ефективному вивченню англійської мови.

Розроблений вебзастосунок орієнтований на створення персоналізованого середовища навчання, яке поєднує сучасні вебтехнології, адаптивний інтерфейс

та механізми повторення матеріалу. Аналіз функціоналу системи показує, що застосунок дозволяє автоматизувати основні процеси навчання та забезпечує користувачу постійний доступ до освітнього контенту.

Однією з головних переваг розробленого вебзастосунку є персоналізація навчального процесу. Система враховує:

- рідну мову користувача;
- мову, що вивчається;
- кількість навчального матеріалу;
- поточний прогрес;
- результати попередніх сесій.

Завдяки цьому користувач отримує індивідуально сформовані навчальні сесії, що дозволяє адаптувати процес навчання до власного рівня підготовки та темпу засвоєння матеріалу.

Реалізація механізму інтервального повторення позитивно впливає на довготривале запам'ятовування лексики та мовних конструкцій. Система автоматично визначає оптимальний час повторення матеріалу, що сприяє більш ефективному закріпленню знань.

Ще одним важливим фактором ефективності є зручність взаємодії користувача з системою. Інтерфейс вебзастосунку реалізовано за допомогою Material UI, що дозволило створити сучасний та зрозумілий дизайн.

До переваг інтерфейсу можна віднести:

- просту навігацію;
- логічне розташування елементів;
- адаптивний дизайн;
- швидке оновлення сторінок;
- підтримку декількох мов інтерфейсу.

Використання React забезпечує динамічне оновлення даних без повного перезавантаження сторінки, що підвищує швидкість роботи застосунку та покращує користувацький досвід.

Адаптивність інтерфейсу дозволяє комфортно використовувати систему як на персональних комп'ютерах, так і на мобільних пристроях. Це підвищує доступність навчального процесу та дає можливість користувачам працювати із застосунком у будь-який час.

Серверна частина застосунку реалізована за допомогою FastAPI, який забезпечує високу швидкість обробки HTTP-запитів та підтримує асинхронну роботу.

Використання REST API дозволяє ефективно організувати взаємодію між клієнтською та серверною частинами системи. Сервер забезпечує:

- обробку запитів користувачів;
- формування навчальних сесій;
- збереження прогресу;
- перевірку відповідей;
- генерацію статистики;
- роботу з механізмом авторизації.

Для зберігання інформації використовується PostgreSQL, що забезпечує надійність зберігання даних та підтримку складних запитів.

Крім того, використання ORM-бібліотеки SQLAlchemy спрощує роботу з базою даних та покращує структуру програмного коду.

Застосунок орієнтований на розвиток декількох мовних навичок одночасно:

- читання;
- розуміння тексту;
- запам'ятовування лексики;
- формування мовних асоціацій.

Використання реальних речень із сервісу Tatoeba Project дозволяє працювати з автентичними прикладами мовлення, що позитивно впливає на розуміння структури мови та використання слів у контексті.

Система повторення матеріалу сприяє формуванню стабільних мовних навичок та забезпечує регулярну взаємодію користувача з навчальним контентом. Крім того, статистика навчання дозволяє користувачам контролювати власний прогрес, що позитивно впливає на мотивацію та дисципліну під час навчання.

Незважаючи на наявність значної кількості функцій, розроблений вебзастосунок має певні обмеження.

Одним із таких обмежень є залежність від зовнішнього API сервісу Tatoeba Project. У випадку недоступності сервісу або зміни його API можуть виникати проблеми з отриманням нових речень для навчання.

Також застосунок орієнтований переважно на роботу з лексикою та реченнями, тому можливості поглибленого вивчення граматики залишаються обмеженими. У поточній реалізації система не містить окремого модуля теоретичних граматичних пояснень або складних граматичних вправ.

Ще одним обмеженням є відсутність повноцінної підтримки аудіо та розпізнавання мовлення. Через це застосунок менш ефективний для розвитку навичок вимови та усного мовлення порівняно з деякими сучасними платформами.

Крім того, ефективність навчання значною мірою залежить від регулярності використання системи самим користувачем. Нерегулярне проходження сесій може знижувати результативність механізму інтервального повторення.

Висновки до розділу 3

У третьому розділі було розглянуто особливості функціонування та елементи інтерфейсу розробленого вебзастосунку для вивчення англійської мови. Проведений аналіз показав, що система забезпечує комплексний підхід до організації навчального процесу та надає користувачеві зручний інструмент для вивчення іноземної мови.

Виконано огляд основного функціоналу вебзастосунку, зокрема механізмів реєстрації та авторизації користувачів, формування навчальних сесій, проходження вправ, повторення матеріалу, перегляду статистики та налаштування параметрів навчання. Реалізовані функції дозволяють організувати персоналізований процес навчання та забезпечують адаптацію системи до індивідуальних потреб користувача.

Також описано взаємодію користувача з застосунком. Проаналізовано основні етапи роботи із системою - від створення облікового запису до проходження навчальних сесій та аналізу результатів навчання. Встановлено, що інтерфейс застосунку побудований за принципами простоти, інтерактивності та адаптивності, що позитивно впливає на зручність використання системи на різних типах пристроїв.

У ході аналізу ефективності вебзастосунку визначено, що реалізовані механізми інтервального повторення, персоналізації навчання та автоматичного формування вправ сприяють підвищенню ефективності засвоєння матеріалу. Використання сучасних вебтехнологій забезпечує швидку роботу системи, а інтеграція з зовнішніми джерелами даних дозволяє використовувати автентичний навчальний контент.

Крім переваг системи, було розглянуто й можливі обмеження вебзастосунку, серед яких залежність від зовнішнього API, обмежені можливості роботи з аудіо та недостатнє опрацювання граматичних аспектів мови. Водночас визначено перспективні напрями подальшого розвитку системи, пов'язані з розширенням функціоналу, додаванням мультимедійних можливостей та впровадженням більш складних адаптивних алгоритмів навчання.

Таким чином, результати виконаної роботи підтверджують, що розроблений вебзастосунок є ефективним інструментом для організації процесу вивчення англійської мови та може бути основою для створення повноцінної сучасної освітньої платформи.

ВИСНОВКИ

У результаті виконання дипломної роботи було розроблено вебзастосунок для вивчення англійської мови, призначений для організації навчального процесу, автоматизації контролю прогресу користувачів та підвищення ефективності засвоєння навчального матеріалу.

У ході виконання роботи було проведено аналіз сучасних підходів до створення вебзастосунків для вивчення іноземних мов, а також досліджено функціональні можливості популярних платформ Duolingo, Memrise та Busuu. Проведений аналіз дозволив визначити основні переваги та недоліки існуючих рішень і сформулювати перелік вимог до розроблюваної системи.

Обґрунтовано вибір технологічного стеку та архітектури програмного продукту. Для реалізації клієнтської частини використано React, TypeScript, Material UI та MobX, що забезпечило створення сучасного адаптивного інтерфейсу та ефективне керування станом застосунку. Серверна частина реалізована за допомогою FastAPI та PostgreSQL, що дозволило забезпечити високу продуктивність, масштабованість та надійне зберігання даних. Архітектура застосунку побудована відповідно до моделі MVC, що сприяє зручності супроводу та подальшого розвитку системи.

У процесі проектування було створено інформаційну модель вебзастосунку та розроблено структуру бази даних, яка забезпечує зберігання облікових записів користувачів, навчальних сесій, результатів виконання завдань, статистичних показників та інформації про вивчені слова. Побудована модель дозволяє реалізувати персоналізований підхід до навчання та забезпечує цілісність даних.

У вебзастосунку реалізовано основні алгоритми обробки даних, зокрема алгоритми реєстрації та автентифікації користувачів, формування навчальних сесій, перевірки відповідей, накопичення прогресу та інтервального повторення матеріалу. Застосування механізму інтервального повторення дозволяє підвищити ефективність запам'ятовування нових слів і мовних конструкцій та

забезпечує адаптацію процесу навчання до індивідуальних особливостей користувача.

Розроблений вебзастосунок забезпечує можливість проходження навчальних та повторювальних сесій, перегляду статистики навчання, налаштування параметрів роботи системи та керування обліковим записом. Для отримання навчального матеріалу використовується зовнішній мовний ресурс Tatoeba, що дозволяє працювати з великою кількістю автентичних прикладів речень різними мовами.

Проведений аналіз ефективності розробленого програмного продукту показав, що створена система забезпечує персоналізоване навчання, автоматичний контроль прогресу та зручну взаємодію користувача із навчальним матеріалом. Водночас подальший розвиток застосунку може передбачати впровадження більш складних алгоритмів адаптивного навчання, інтеграцію технологій штучного інтелекту, розширення мовної підтримки та реалізацію мобільної версії системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. European Commission. Digital Education Action Plan [Електронний ресурс]. - Режим доступу: <https://education.ec.europa.eu/focus-topics/digital-education/actions/plan>
2. E-learning 2.0 Research [Електронний ресурс]. - Режим доступу: <https://arxiv.org/abs/1001.4738>
3. Duolingo Efficacy Studies [Електронний ресурс]. - Режим доступу: <https://en.duolingo.com/efficacy/studies>
4. Memrise About (Science & Learning) [Електронний ресурс]. - Режим доступу: <https://www.memrise.com/about>
5. Memrise Spaced Repetition System [Електронний ресурс]. - Режим доступу: <https://memrisebeta.zendesk.com/hc/en-us/articles/24998764126097-How-does-the-spaced-repetition-system-work>
6. Busuu Courses and CEFR [Електронний ресурс]. - Режим доступу: <https://www.busuu.com/en/it-works/courses>
7. React Documentation [Електронний ресурс]. - Режим доступу: <https://react.dev/>
8. MobX Documentation [Електронний ресурс]. - Режим доступу: <https://mobx.js.org/README.html>
9. FastAPI Documentation [Електронний ресурс]. - Режим доступу: <https://fastapi.tiangolo.com/>
10. PostgreSQL Documentation [Електронний ресурс]. - Режим доступу: <https://www.postgresql.org/docs/>
11. SQLAlchemy Documentation [Електронний ресурс]. - Режим доступу: <https://docs.sqlalchemy.org/en/20/>
12. OWASP Password Storage Cheat Sheet [Електронний ресурс]. - Режим доступу: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

13. Tatoeba Project Documentation [Электронный ресурс]. - Режим доступа: <https://tatoeba.org/uk/about>
14. Pydantic Documentation [Электронный ресурс]. - Режим доступа: <https://pydantic.dev/docs/>
15. Nielsen Norman Group. User Interface Design Guidelines [Электронный ресурс]. - Режим доступа: <https://www.nngroup.com/articles/>
16. SuperMemo. History of Spaced Repetition [Электронный ресурс]. – Режим доступа: <https://www.supermemo.com/en/blog/history-of-spaced-repetition>
17. Anki Manual [Электронный ресурс]. – Режим доступа: <https://docs.ankiweb.net/>
18. TypeScript Documentation [Электронный ресурс]. – Режим доступа: <https://www.typescriptlang.org/docs/>
19. Material UI Documentation [Электронный ресурс]. – Режим доступа: <https://mui.com/material-ui/>
20. SQLAlchemy Documentation [Электронный ресурс]. – Режим доступа: <https://sqlmodel.tiangolo.com/>
21. W3C Web Accessibility Initiative (WAI) [Электронный ресурс]. – Режим доступа: <https://www.w3.org/WAI/>
22. MDN Web Docs [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/>
23. MDN Web Docs. HTTP Overview [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>
24. OWASP Top 10 [Электронный ресурс]. – Режим доступа: <https://owasp.org/www-project-top-ten/>
25. JWT.io. Introduction to JSON Web Tokens [Электронный ресурс]. – Режим доступа: <https://jwt.io/introduction>
26. RFC 7519. JSON Web Token (JWT) [Электронный ресурс]. – Режим доступа: <https://datatracker.ietf.org/doc/html/rfc7519>
27. PostgreSQL Tutorial [Электронный ресурс]. – Режим доступа: <https://www.postgresqltutorial.com/>

28. Python Documentation [Электронный ресурс]. – Режим доступа:
<https://docs.python.org/3/>

29. Starlette Documentation [Электронный ресурс]. – Режим доступа:
<https://www.starlette.io/>

30. Neon Documentation [Электронный ресурс]. – Режим доступа:
<https://neon.tech/docs>

Повний код файлу authentication.py

```

import base64
import binascii
import hashlib
import hmac
import json
import os
import secrets
from datetime import datetime, timedelta, timezone
from typing import Any, cast

from fastapi import Response
from sqlalchemy.exc import IntegrityError
from sqlmodel import Session, col, select

from core.database import get_session
from data.models.User import User, UserCreate
from data.payloads.user_login_details import UserLoginDetails
from data.payloads.user_registration_details import UserRegistrationDetails

AUTH_TOKEN_COOKIE_NAME = os.getenv("AUTH_TOKEN_COOKIE_NAME",
"auth_token")
AUTH_TOKEN_EXPIRES_SECONDS = int(
    os.getenv("AUTH_TOKEN_EXPIRES_SECONDS", str(60 * 60 * 24 * 7))
)
AUTH_TOKEN_SECRET = os.getenv("AUTH_TOKEN_SECRET") or
secrets.token_urlsafe(32)

def _base64url_encode(payload: bytes) -> str:
    return base64.urlsafe_b64encode(payload).decode("ascii").rstrip("=")

def _base64url_decode(payload: str) -> bytes:
    padding = "=" * (-len(payload) % 4)
    return base64.urlsafe_b64decode(f"{payload}{padding}".encode("ascii"))

def _json_base64url_encode(payload: dict[str, Any]) -> str:
    json_payload = json.dumps(payload, separators=(",", ":"), sort_keys=True)
    return _base64url_encode(json_payload.encode("utf-8"))

def _json_base64url_decode(payload: str) -> dict[str, Any] | None:
    try:
        decoded_payload = _base64url_decode(payload)
        parsed_payload = json.loads(decoded_payload.decode("utf-8"))
    except (
        binascii.Error,
        json.JSONDecodeError,
        UnicodeDecodeError,
    ):
```

```

        UnicodeEncodeError,
        ValueError,
    ):
        return None

    if not isinstance(parsed_payload, dict):
        return None

    return cast(dict[str, Any], parsed_payload)

def get_auth_token_expires_at() -> datetime:
    return datetime.now(timezone.utc)
timedelta(seconds=AUTH_TOKEN_EXPIRES_SECONDS) +

def create_auth_token(user: User, expires_at: datetime) -> str:
    header = _json_base64url_encode({"alg": "HS256", "typ": "JWT"})
    payload = _json_base64url_encode(
        {
            "sub": str(user.id),
            "email": user.email,
            "exp": int(expires_at.timestamp()),
        }
    )
    unsigned_token = f"{header}.{payload}"
    signature = hmac.new(
        AUTH_TOKEN_SECRET.encode("utf-8"),
        unsigned_token.encode("ascii"),
        hashlib.sha256,
    ).digest()
    return f"{unsigned_token}.{_base64url_encode(signature)}"

def get_valid_auth_token_payload(token: str) -> dict[str, Any] | None:
    token_parts = token.split(".")
    if len(token_parts) != 3:
        return None

    encoded_header, encoded_payload, encoded_signature = token_parts
    header = _json_base64url_decode(encoded_header)
    payload = _json_base64url_decode(encoded_payload)
    if header is None or payload is None:
        return None

    if header.get("alg") != "HS256" or header.get("typ") != "JWT":
        return None

    unsigned_token = f"{encoded_header}.{encoded_payload}"
    expected_signature = hmac.new(
        AUTH_TOKEN_SECRET.encode("utf-8"),

```

```

        unsigned_token.encode("ascii"),
        hashlib.sha256,
    ).digest()

    try:
        signature = _base64url_decode(encoded_signature)
    except (binascii.Error, UnicodeEncodeError, ValueError):
        return None

    if not hmac.compare_digest(signature, expected_signature):
        return None

    expires_timestamp = payload.get("exp")
    if not isinstance(expires_timestamp, int | float) or isinstance(expires_timestamp, bool):
        return None

    try:
        expires_at = datetime.fromtimestamp(expires_timestamp, timezone.utc)
    except (OSError, OverflowError, ValueError):
        return None
    if expires_at <= datetime.now(timezone.utc):
        return None

    if not isinstance(payload.get("sub"), str):
        return None

    return payload

def get_valid_auth_token_expires_at(token: str) -> datetime | None:
    payload = get_valid_auth_token_payload(token)
    if payload is None:
        return None

    expires_timestamp = payload.get("exp")
    if not isinstance(expires_timestamp, int | float) or isinstance(expires_timestamp, bool):
        return None

    try:
        expires_at = datetime.fromtimestamp(expires_timestamp, timezone.utc)
    except (OSError, OverflowError, ValueError):
        return None

    return expires_at

def get_valid_auth_token_user_id(token: str) -> int | None:
    payload = get_valid_auth_token_payload(token)
    if payload is None:
        return None

```

```

try:
    return int(payload["sub"])
except (TypeError, ValueError):
    return None

def _hash_password(password: str) -> str:
    salt = os.urandom(16)
    derived_key = hashlib.scrypt(
        password.encode("utf-8"),
        salt=salt,
        n=2**14,
        r=8,
        p=1,
    )
    encoded_salt = base64.urlsafe_b64encode(salt).decode("ascii")
    encoded_key = base64.urlsafe_b64encode(derived_key).decode("ascii")
    return f"scrypt$16384$8$1${encoded_salt}${encoded_key}"

def verify_password(password: str, password_hash: str) -> bool:
    try:
        algorithm, n_value, r_value, p_value, encoded_salt, encoded_key =
password_hash.split("$")
    except ValueError:
        return False

    if algorithm != "scrypt":
        return False

    try:
        salt = base64.urlsafe_b64decode(encoded_salt.encode("ascii"))
        expected_key = base64.urlsafe_b64decode(encoded_key.encode("ascii"))
        derived_key = hashlib.scrypt(
            password.encode("utf-8"),
            salt=salt,
            n=int(n_value),
            r=int(r_value),
            p=int(p_value),
        )
    except (ValueError, TypeError):
        return False

    return hmac.compare_digest(derived_key, expected_key)

def _normalize_email(email: str) -> str:
    return email.strip().lower()

def delete_auth_cookie(response: Response) -> None:
    response.delete_cookie(

```

```

        key=AUTH_TOKEN_COOKIE_NAME,
        path="/",
        secure=True,
        httponly=True,
        samesite="lax",
    )
def create_user(session: Session, details: UserCreate | UserRegistrationDetails) -> User:
    normalized_email = _normalize_email(str(details.email))
    existing_user = session.exec(
        select(User).where(col(User.email) == normalized_email)
    ).first()
    if existing_user is not None:
        raise ValueError("User with this email already exists")
    name = details.name.strip() if details.name is not None else None
    user = User(
        email=normalized_email,
        name=name or None,
        password_hash=_hash_password(details.password),
    )
    try:
        session.add(user)
        session.commit()
        session.refresh(user)
    except IntegrityError as error:
        session.rollback()
        raise ValueError("User with this email already exists") from error
    return user
def register_user(details: UserCreate | UserRegistrationDetails) -> User:
    with get_session() as session:
        return create_user(session, details)

def authenticate_user(session: Session, details: UserLoginDetails) -> User | None:
    normalized_email = _normalize_email(str(details.email))
    user = session.exec(
        select(User).where(col(User.email) == normalized_email)
    ).first()
    if user is None:
        return None

    if not verify_password(details.password, user.password_hash):
        return None

    return user

def authenticate_user_credentials(details: UserLoginDetails) -> User | None:
    with get_session() as session:
        return authenticate_user(session, details)

```

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

Я, Дорошенко Родіон Олегович,

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

Засвідчую, що кваліфікаційна бакалаврська робота

«Розробка вебзастосунку для вивчення англійської мови»

виконана самостійно та не містить академічного плагіату. Я не надавав і не одержував 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. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення норм академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.

Також я поінформований, 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.

15.06.26 р



Дорошенко Р. О.