

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

«Платформа для вивчення англійських слів»

Виконав студент групи ПЗ-20-4

МЕЛЬНИК Павло Іванович

Керівник роботи:

ВЛАСЕНКО Олег Васильович

Рецензент:

САВІЦЬКИЙ Роман Святославович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«26» лютого 2024р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **МЕЛЬНИК Павло Іванович**
Керівник роботи: **ВЛАСЕНКО Олег Васильович**
Тема роботи: **«Платформа для вивчення англійських слів»**,
затверджена Наказом закладу вищої освіти від **«23» лютого 2024р., №74/с**
Вихідні дані для роботи: **процес вивчення англійських слів, заснований на методології Пола Нейшна, з використанням інтерактивних методів та гейміфікацією для залучення користувачів.**
Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Олег ВЛАСЕНКО	02.03.2024	02.03.2024
2	Олег ВЛАСЕНКО	04.04.2024	04.04.2024
3	Олег ВЛАСЕНКО	02.05.2024	02.05.2024

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з веб додатку платформи для вивчення англійських слів та пояснювальної записки. Пояснювальна записка до випускної роботи містить 79 сторінок, 46 ілюстрацій та 15 таблиць.

Метою роботи є створення платформи для вивчення англійських слів з використанням сучасних технологій та інтеграцією методології Пола Нейшна для ефективного навчання.

Архітектура платформи для вивчення англійських слів базується на сучасних веб-технологіях, що забезпечують інтуїтивний інтерфейс та високу продуктивність. Інтерактивні методи, такі як списки флеш-карток, забезпечують ефективне запам'ятовування нових слів. Гейміфікація, включаючи статистику по пройденим словам, яка стимулює користувачів до регулярного навчання та відстеження прогресу.

Додаток має зручний інтерфейс, який дозволяє користувачам легко організовувати свій навчальний процес, встановлювати цілі вивчення слів та відслідковувати свій прогрес.

КЛЮЧОВІ СЛОВА: АНГЛІЙСЬКА МОВА, ВЕБСЕРВІС, ГЕЙМІФІКАЦІЯ, ІНТЕРАКТИВНЕ НАВЧАННЯ, ПЛАТФОРМА.

					ІПЗ.КР.Б-121-24-ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Платформа для вивчення англійських слів	Літ.	Арк.	Акрушіє
Розроб.		Мельник П.І.						
Керівник		Власенко О.В.					3	79
						Житомирська політехніка, група ІПЗ-20-4		
Н. Контр.		Тичина О.П.						
Зав. каф.		Вакалюк Т.А.						

ABSTRACT

The bachelor's thesis consists of a web application platform for learning English words and an explanatory note. The explanatory note for the thesis contains 79 pages, 46 illustrations, and 15 tables.

The goal of the work is to create a platform for learning English words using modern technologies and integrating Paul Nation's methodology for effective learning.

The architecture of the platform for learning English words is based on modern web technologies, ensuring an intuitive interface and high performance. Interactive methods, such as flashcard lists, ensure effective memorization of new words. Gamification, including statistics on words learned, stimulates users to regular learning and progress tracking.

The application has a user-friendly interface that allows users to easily organize their learning process, set word learning goals, and track their progress.

KEYWORDS: ENGLISH LANGUAGE, GAMIFICATION, INTERACTIVE LEARNING, PLATFORM, WEBSERVICE.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ПЛАТФОРМИ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКИХ СЛІВ.....	9
1.1 Постановка задачі	9
1.2 Аналіз аналогів програмного продукту	11
1.3 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення.....	15
Висновки до першого розділу	17
РОЗДІЛ 2 ПРОЄКТУВАННЯ ПЛАТФОРМИ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКИХ СЛІВ	19
2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи	19
2.2 Розробка бази даних системи	29
2.3 Проектування та реалізація алгоритмів роботи системи	32
2.4 Реалізація функціоналу платформи для вивчення англійських слів...	33
Висновки до другого розділу.....	39
РОЗДІЛ 3 ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ПЛАТФОРМОЮ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКИХ СЛІВ.....	40
3.1 Порядок встановлення та налаштування платформи	40
3.2 Структура інтерфейсу платформи для вивчення англійських слів	41
3.3 Тестування платформи.....	59
Висновки до третього розділу	61
ВИСНОВКИ	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	64
ДОДАТКИ.....	67

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – прикладний програмний інтерфейс

CRUD – основні операції, що використовуються для управління даними

DApps – Децентралізований додаток

MVC – схема розділення даних додатків і керуючої логіки на три окремі компоненти Model-View-Controller (Create , Read , Update , Delete)

SDK – набір для розробки програмного забезпечення (software development kit)

БД – база даних

ПЗ – програмне забезпечення

СУБД – система управління базами даних

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми полягає в необхідності використання сучасних технологій для покращення процесу вивчення іноземних мов. Зростаючий попит на вивчення англійської мови вимагає нових підходів та інструментів, що роблять навчання більш ефективним та доступним. Інтеграція методології Пола Нейшна, яка є однією з найефективніших у сфері навчання мовам, з сучасними веб-технологіями дозволяє створити платформу, яка відповідає потребам різних категорій користувачів.

Метою випускної кваліфікаційної роботи бакалавра є розробка платформи для вивчення англійських слів з використанням сучасних технологій та інтеграцією методології Пола Нейшна для ефективного навчання. Для реалізації поставленої мети необхідно провести огляд існуючих навчальних платформ, методів інтерактивного навчання та гейміфікації, а також розробити архітектуру платформи з урахуванням потреб користувачів.

Встановлена мета обумовлює наступні **завдання**:

1. Огляд літератури та аналіз існуючих рішень:
 - Провести аналіз існуючих платформ для вивчення англійських слів.
 - Вивчити методи інтерактивного навчання та гейміфікації, які використовуються у сучасних навчальних платформах.
2. Розробка концепції та структури платформи:
 - Визначити основні функціональні вимоги до платформи.
 - Спроектувати архітектуру платформи, включаючи базу даних, модулі та інтерфейси.
3. Реалізація **основних функцій** платформи:
 - Реалізувати систему реєстрації та авторизації користувачів.
 - Створити модулі для управління словами та списками слів.
 - Інтегрувати методологію Пола Нейшна для ефективного навчання.
4. Розробка інтерфейсу користувача:

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

- Спроектувати та реалізувати зручний та інтуїтивний інтерфейс користувача.
- Забезпечити можливість додавання коментарів та оцінок до слів та списків слів.

5. Впровадження елементів гейміфікації:

- Розробити систему відстеження прогресу користувачів.
- Інтегрувати елементи гейміфікації для стимулювання користувачів до регулярного навчання.

6. Тестування та оцінка ефективності платформи:

- Провести тестування функціональності платформи.
- Оцінити ефективність використання платформи на основі зворотного зв'язку від користувачів та результатів їхнього навчання.

Об'єктом дослідження є процес вивчення англійських слів з використанням інтерактивних методів та гейміфікації.

Предметом дослідження є функціональність платформи для вивчення англійських слів, її можливості та переваги у порівнянні з традиційними методами навчання. Дослідження також охоплює аналіз інтерактивних методів, таких як списки флеш-карток, і їх вплив на ефективність навчання, а також роль гейміфікації в стимулюванні користувачів до регулярного навчання та відстеження прогресу.

Методи аналізу та інформаційна база дослідження включають використання аналітичних методів для оцінки ефективності інтерактивних методів навчання та гейміфікації, а також аналіз існуючих платформ і навчальних підходів. Інформаційною базою дослідження будуть публікації наукових статей, книги та інші джерела, що висвітлюють тему навчання англійської мови.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ПЛАТФОРМИ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКИХ СЛІВ

1.1 Постановка задачі

Метою кваліфікаційної роботи бакалавра є створення платформи для вивчення англійських слів, використовуючи сучасні технології, що дозволить легко користуватись та масштабувати її у майбутньому.

Вивчення англійської мови є актуальним завданням для багатьох людей по всьому світу, адже англійська мова є міжнародною мовою спілкування [1]. Сучасні технології дозволяють створювати інноваційні рішення, які роблять процес навчання більш ефективним та доступним [2]. Одним з таких рішень є платформа для вивчення англійських слів, яка використовує інтерактивні методи та гейміфікацію [3].

Платформа для вивчення англійських слів має представляти собою декілька програмних модулів, що забезпечують різноманітні функції та можливості для користувачів:

- Модуль навчання. Модуль дозволяє користувачам створювати індивідуальні списки слів в вигляді флеш-карток. Він надає можливість в подальшому вчити власні списки та списки інших користувачів [4].
- Модуль гейміфікації. Модуль включає статистику по пройденим словам і спискам, що стимулює користувачів до регулярного навчання та відстеження прогресу [5].
- Модуль пошуку. Модуль надає засоби для розумного пошуку слів за багатьма параметрами та детальну інформацію про слово з його застосуванням, прикладами тощо [6].

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

- Модуль адміністрування. Модуль забезпечує адміністрування додатку, що дає включає в себе новий функціонал для створення списків, які не мають звичайні користувачі платформи [7].

В результаті повинно бути створено веб-сайт, на якому будь-який бажаючий може створити свій навчальний план та почати вивчати англійські слова [8]. Також важливою складовою є зручність використання, оскільки всі модулі мають бути інтегровані в єдину систему, що дозволяє користувачам легко орієнтуватись на платформі [9].

Основними етапами розробки платформи є:

1. Розробка дизайну веб-сайту:

- створення загальної концепції дизайну додатку;
- створення UI компонентів;
- розробка дизайну сторінок.

2. Побудова архітектури клієнтської частини додатку.

3. Побудова архітектури серверної частини додатку.

4. Розробка системи авторизації та ролей користувачів.

5. Розробка основного функціоналу:

- створення логіки авторизації користувачів з відповідними ролями;
- створення бази даних для зберігання інформації [10];
- імплементація серверної частини на клієнтській стороні;
- організація спілкування серверної частини з клієнтською;
- створення та інтеграція інтерактивних навчальних модулів.

6. Тестування додатку:

- функціональне тестування додатку;
- тестування графічного інтерфейсу.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Результатом виконання плану дій є створення платформи, за допомогою якої користувачі можуть створювати індивідуальні навчальні плани, відслідковувати свій прогрес та ефективно вивчати англійські слова за допомогою сучасних інтерактивних методів [11].

1.2 Аналіз аналогів програмного продукту

Розробка платформи для вивчення англійських слів потребує глибокого аналізу конкурентів на ринку. Існує багато популярних додатків для вивчення мов, які використовують сучасні технології та інтерактивні методи навчання [12]. Проведемо аналіз трьох таких додатків.

1. Duolingo [13] (рис. 1.1)

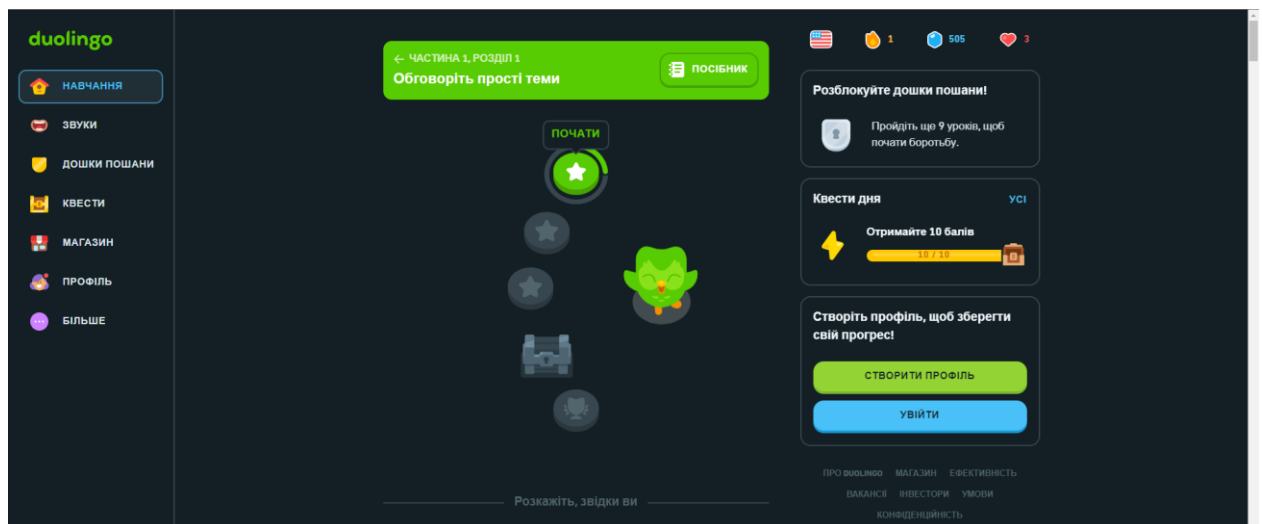


Рис. 1.1 Платформа для вивчення мов Duolingo

Duolingo є одним з найпопулярніших додатків для вивчення мов у світі, який має інтуїтивно зрозумілий інтерфейс та використовується мільйонами людей [14].

Основними перевагами Duolingo є:

- Інтерактивний та гейміфікований процес навчання [15].
- Безкоштовний доступ до основного контенту.
- Наявність мобільних додатків для iOS та Android.
- Підтримка багатьох мов для вивчення.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

- Можливість вивчати декілька мов одночасно.

Недоліками є:

- Обмежений безкоштовний доступ до деяких функцій.
- Реклама в безкоштовній версії.
- Обмеженість у глибині матеріалу для просунутих рівнів.

2. Memrise [16] (рис. 1.2)

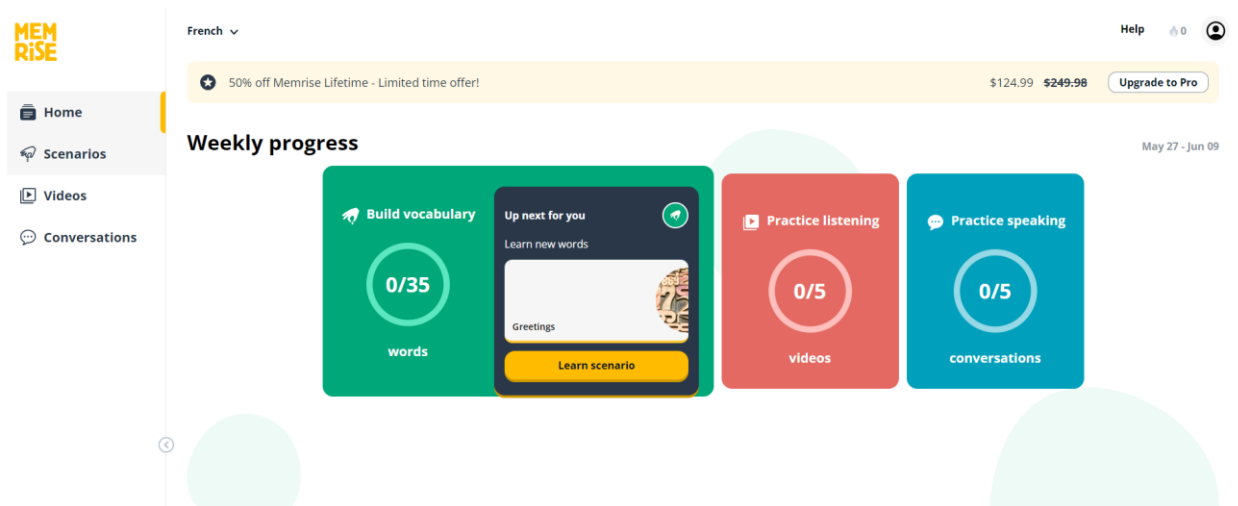


Рис. 1.2 Платформа для вивчення мов Memrise

Memrise є платформою, що пропонує навчання мов за допомогою флеш-карток та відео-уроків, що робить процес навчання більш живим та захоплюючим [17].

Основними перевагами Memrise є:

- Використання реальних відео з носіями мови.
- Гейміфікація та система нагород.
- Підтримка декількох мов для вивчення.
- Наявність мобільних додатків для iOS та Android.
- Можливість створення власних курсів.

Недоліками є:

- Деякі курси доступні лише за підпискою.
- Обмежений контент для деяких мов.
- Не завжди точна локалізація.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Babbel [18] (рис. 1.3)

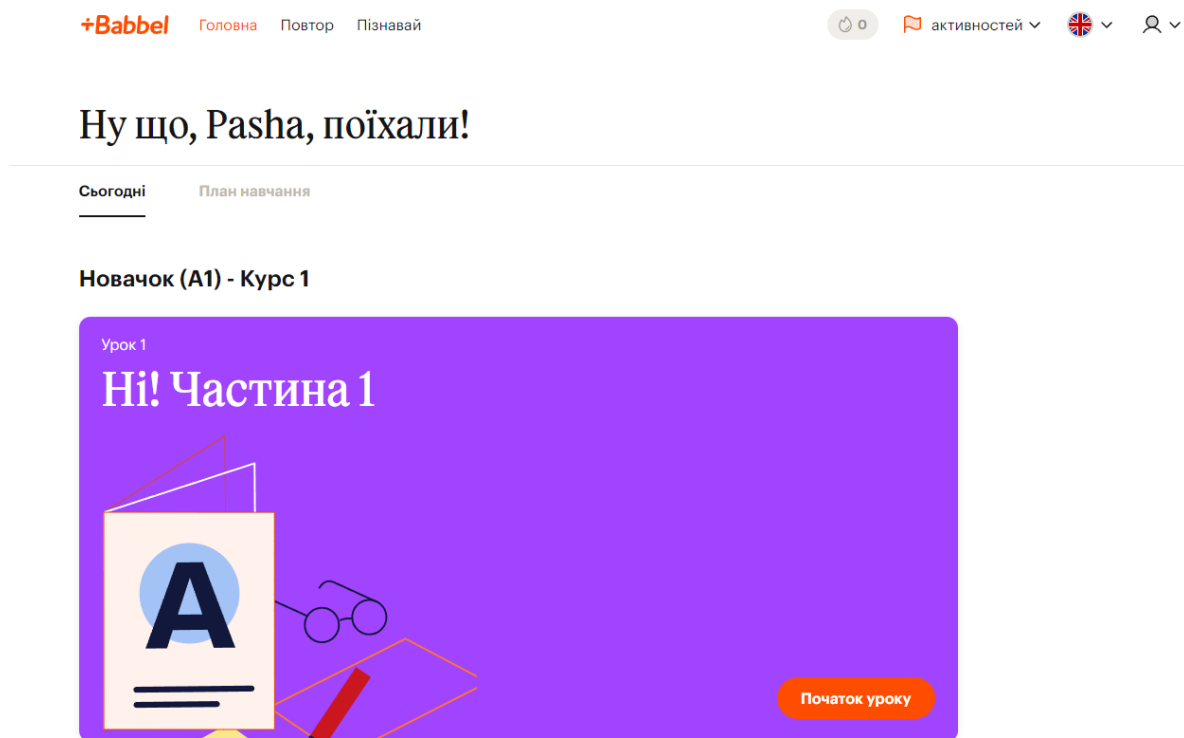


Рис. 1.3 Платформа для вивчення мов Babbel

Babbel є комерційним додатком для вивчення мов, що пропонує структуровані заняття з акцентом на розмовну практику та граматику [19].

Основними перевагами Babbel є:

- Високоякісні курси, створені лінгвістами.
- Інтерактивні заняття з акцентом на практичні навички.
- Підтримка багатьох мов.
- Можливість завантаження уроків для офлайн навчання.
- Інтеграція з реальними життєвими ситуаціями.

Недоліками є:

- Всі курси доступні лише за підпискою.
- Відсутність безкоштовного контенту.

Виходячи з наявних переваг та недоліків, складемо порівняльну таблицю вищезгаданих програмних аналогів (табл. 1.1)

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Порівняння платформ

Характеристика	Duolingo	Memrise	Babbel
Інтерактивний процес навчання	+	+	+
Гейміфікація	+	+	-
Безкоштовний доступ	+	+	-
Наявність мобільних додатків	+	+	+
Підтримка кількох мов	+	+	+
Реальні відео з носіями мови	-	+	-
Створення власних курсів	-	+	-

Переглянувши платформи для вивчення англійських слів, можна виділити основні відмінності: деякі з них мають інтерактивний та гейміфікований процес навчання, інші пропонують високоякісні курси з акцентом на практичні навички, а треті використовують реальні відео з носіями мови [20]. Важливими факторами є наявність мобільних додатків, можливість створення власних курсів та система нагород [21]. Але у всіх є мінус в вигляді платного контенту, без якого важко досягнути продуктивного навчання, в когось це менший вплив на загальний контент, в когось весь контент платний.

Враховуючи інформацію, отриману при дослідженні, було прийнято рішення створити платформу для вивчення англійських слів, яка поєднає інтерактивні методи навчання та гейміфікацію. Також зробити весь контент абсолютно безкоштовним, щоб повний обсяг додатку був доступним для кожного. Додаток буде мати інтуїтивний дизайн, що дозволить користувачам легко організовувати свій навчальний процес та відслідковувати свій прогрес. Платформа буде підтримувати сучасні технології, забезпечуючи високу продуктивність та доступність на різних пристроях.

1.3 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Веб додаток для вивчення англійських слів буде побудований на клієнт-серверній архітектурі. Серверна частина буде написана з використанням сучасного backend-фреймворку Laravel на PHP. Для клієнтської частини буде використано JavaScript та jQuery. Як сховище буде використовуватись база даних MySQL.

PHP Laravel

Laravel - це фреймворк для розробки серверних застосунків на PHP. Він пропонує багато корисних інструментів та можливостей для створення високоякісних, масштабованих та легко тестованих додатків.

Переваги Laravel:

- Модульність: Laravel підтримує модульну архітектуру, що дозволяє легко організовувати код та зменшує його складність.
- Безпека: Laravel надає вбудовані механізми захисту від поширених веб-вразливостей, таких як SQL-ін'єкції та XSS.
- Розширюваність: Завдяки великій кількості пакетів та розширень, Laravel дозволяє швидко додавати новий функціонал до додатку.
- Спільнота та документація: Laravel має активну спільноту розробників та детальну документацію, що робить його вивчення та використання простішим.

JavaScript та jQuery

JavaScript є основною мовою для розробки інтерактивного frontend. Використання jQuery дозволяє спростити маніпуляції з DOM, обробку подій та AJAX-запити.

Переваги JavaScript та jQuery:

- Інтерактивність: JavaScript та jQuery дозволяють створювати динамічні та інтерактивні інтерфейси користувача.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

- Кроссплатформеність: JavaScript працює на всіх сучасних веб-браузерах, що забезпечує доступність додатку на різних пристроях.
- Простота використання: jQuery спрощує багато завдань, які можуть бути складними при використанні чистого JavaScript, що дозволяє швидше розробляти функціонал.

MySQL

MySQL є однією з найпопулярніших систем управління базами даних. Вибір MySQL обумовлений наступними причинами:

- Продуктивність: MySQL забезпечує високу швидкість роботи та ефективне управління великими обсягами даних.
- Надійність: СУБД MySQL відома своєю надійністю та стабільністю, що важливо для забезпечення безперебійної роботи платформи.
- Спільнота та підтримка: MySQL має велику спільноту користувачів, що дозволяє швидко знаходити рішення на виникаючі питання та проблеми.

Вимоги до апаратного забезпечення

Для забезпечення стабільної та ефективної роботи платформи для вивчення англійських слів, необхідне наступне апаратне забезпечення:

1. Сервер:

- Процесор: багатоядерний процесор з тактовою частотою не менше 2.5 ГГц.
- Оперативна пам'ять: не менше 8 ГБ для забезпечення плавної роботи додатку та обробки запитів.
- Жорсткий диск:
- SSD-диск об'ємом не менше 256 ГБ для швидкого зчитування та запису даних.
- Мережеве підключення: високошвидкісне інтернет-з'єднання для забезпечення швидкої роботи додатку при великій кількості користувачів.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Клієнтські пристрої:

- Браузер: сучасний веб-браузер з підтримкою JavaScript та HTML5 (Google Chrome, Mozilla Firefox, Microsoft Edge тощо).
- Оперативна пам'ять: не менше 2 ГБ для плавної роботи веб-додатку.
- Процесор: багатоядерний процесор з тактовою частотою не менше 1.5 ГГц.
- Роздільна здатність екрану: мінімум 1024x768 пікселів для коректного відображення інтерфейсу додатку.

Використання цих інструментальних засобів та відповідного апаратного забезпечення дозволить створити ефективну, стабільну та зручну платформу для вивчення англійських слів, що задовольнить потреби користувачів та забезпечить високу якість навчання.

Висновки до першого розділу

Проведено аналіз предметної області, за допомогою якого були визначені основні вимоги до платформи для вивчення англійських слів, основною задачею якої є забезпечення користувачів ефективними інструментами для вивчення нових слів та відстеження свого прогресу.

В процесі дослідження аналогічних і схожих за концепцією продуктів були проаналізовані та порівняні різні платформи для вивчення мов, зокрема Duolingo, Memrise та Babbel, які пропонують інтерактивні методи навчання, гейміфікацію та різноманітні підходи до організації навчального процесу. Було визначено їх основні переваги та недоліки, що дозволило сформулювати більш точні вимоги до власної платформи.

На основі відповідних вимог та критеріїв був сформований набір основних технологій і засобів реалізації для розробки веб додатку, включаючи PHP Laravel для backend-розробки, JavaScript та jQuery для front-розробки, а також MySQL для управління базою даних. Цей набір технологій забезпечить

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

високу продуктивність, зручність використання та можливість масштабування платформи у майбутньому.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЄКТУВАННЯ ПЛАТФОРМИ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКИХ СЛІВ

2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи

Платформа для вивчення англійських слів надає веб інтерфейс, через який користувачі можуть здійснювати різні операції для полегшення процесу навчання [22]. Функціональність платформи залежить від активності користувачів, які можуть бути поділені на кілька основних груп: звичайні користувачі, які вивчають слова, авторизовані користувачі, адміністратори, які керують контентом платформи [23]. Також додаток представлений з підтримкою 2 мов: англійська і українська, для користувачів, які мають знання з англійської на вищому рівні представлений інтерфейс і опис слів на англійському, щоб повністю бути в англійському середовищі на платформі, а для користувачів, яким ще важко перейти повністю на англійську версію є можливість переглядати інтерфейс і опис слів українською [24].

Неавторизовані користувачі можуть здійснювати наступні дії:

- Використовувати смарт-пошук слів за різними фільтрами.
- Змінювати мову додатка.
- Переглядати різноманітну інформацію про слово (опис, синоніми, антоніми, співзвучні слова, приклади використання) та додавати слова до своїх списків.
- Переглядати словник за алфавітом.
- Переглядати різноманітні списки слів, створені іншими користувачами або адміністраторами (зі зображеннями, обкладинками та розділами) [25].

Авторизовані користувачі можуть здійснювати наступні дії:

- Створювати прості списки слів у вигляді флеш-карток, які використовуються для навчання.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

- Змінювати мову додатка.
- Переглядати статистику по власно пройденим словам і спискам.
- Додавати списки до улюблених
- Використовувати смарт-пошук слів за різними фільтрами.
- Переглядати різноманітну інформацію про слово (опис, синоніми, антоніми, співзвучні слова, приклади використання) та додавати слова до своїх списків.
- Переглядати словник за алфавітом.
- Переглядати різноманітні списки слів, створені іншими користувачами або адміністраторами (зі зображеннями, обкладинками та розділами).

Адміністратори можуть здійснювати наступні дії:

- Створювати унікальні списки, які можуть мати обкладинку та розділи для групування слів.
- Змінювати мову додатка.
- Створювати прості списки слів у вигляді флеш-карток, які використовуються для навчання.
- Переглядати статистику по власно пройденим словам і спискам.
- Додавати списки до улюблених
- Використовувати смарт-пошук слів за різними фільтрами.
- Переглядати різноманітну інформацію про слово (опис, синоніми, антоніми, співзвучні слова, приклади використання) та додавати слова до своїх списків.
- Переглядати словник за алфавітом.
- Переглядати різноманітні списки слів, створені іншими користувачами або адміністраторами (зі зображеннями, обкладинками та розділами).

Для наглядної взаємодії між елементами додатку було створено Діаграму варіантів використання (рис. 2.1).

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

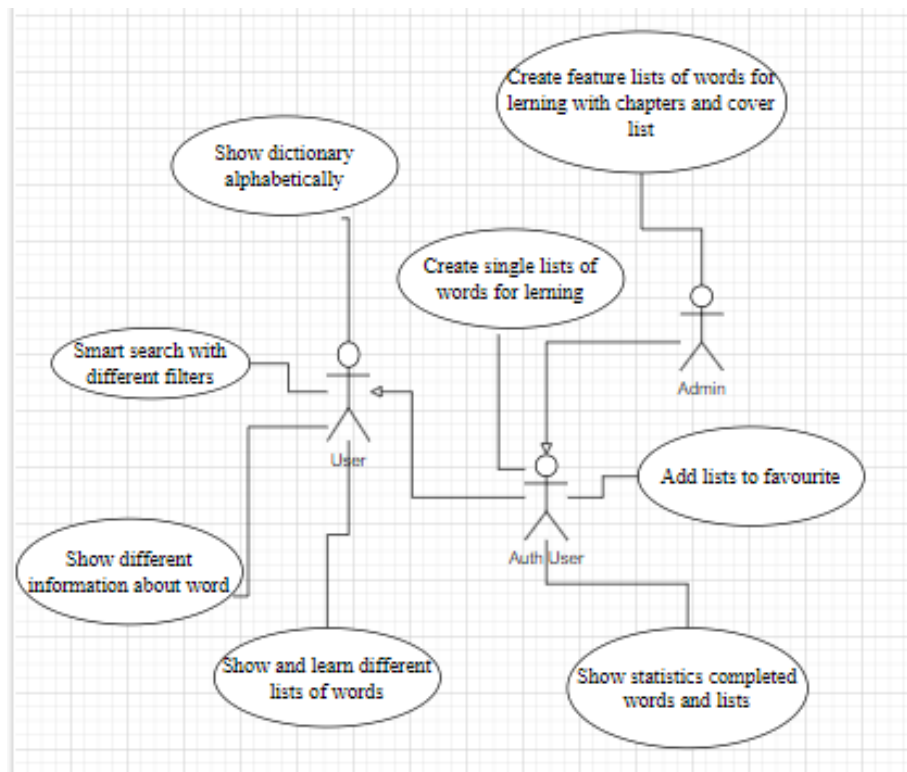


Рис. 2.1 Діаграма варіантів використання

Користувачі можуть взаємодіяти з платформою через різні сценарії використання, забезпечуючи інтерактивність та персоналізацію навчального процесу [26]. Це дозволяє користувачам ефективно організовувати свій процес навчання, встановлювати цілі та відслідковувати свій прогрес, що сприяє більш ефективному та мотивованому вивченню англійських слів [27].

Опис діаграми варіантів використання наведено в (табл. 2.1) – (табл. 2.9)

Таблиця 2.1

Опис прецеденту «Смарт-пошук слів»

Назва варіанту використання	Смарт-пошук слів
Основні актори	Неавторизовані користувачі, авторизовані користувачі, адміністратор
Короткий опис	Користувачі можуть шукати слова за допомогою різних фільтрів, таких як частина мови, діапазон кількості складів, синонім/антонім до заданого слова та інші.

Таблиця 2.2

Опис прецеденту «Інформація про слово»

Назва варіанту використання	Інформація про слово
Основні актори	Неавторизовані користувачі, авторизовані користувачі, адміністратор
Короткий опис	Користувачі можуть переглядати детальну інформацію про слово, включаючи опис, синоніми, антоніми, співзвучні слова, приклади використання та інше.

Таблиця 2.3

Опис прецеденту «Словник за алфавітом»

Назва варіанту використання	Словник за алфавітом
Основні актори	Неавторизовані користувачі, авторизовані користувачі, адміністратор
Короткий опис	Користувачі можуть переглядати слова у словнику в алфавітному порядку.

Таблиця 2.4

Опис прецеденту «Перегляд/Вивчення списків слів»

Назва варіанту використання	Перегляд/Вивчення списків слів
Основні актори	Неавторизовані користувачі, авторизовані користувачі, адміністратор
Короткий опис	Користувачі можуть переглядати різноманітні списки слів у вигляді флеш карток, створені іншими користувачами або адміністраторами.

Таблиця 2.5

Опис прецеденту «Створення простих списків слів»

Назва варіанту використання	Створення простих списків слів
Основні актори	Авторизовані користувачі, адміністратор
Короткий опис	Користувачі можуть створювати власні прості списки слів у вигляді флеш-карток для навчання, що є одним із пунктів методології Поля Нейшна про зручність запам'ятовування нових і вже вивчених раніше слів у вигляді карток, на зворотній стороні яких представлений опис слова.

Таблиця 2.6

Опис прецеденту «Створення розширених списків слів»

Назва варіанту використання	Створення розширених списків слів
Основні актори	Адміністратор
Короткий опис	Адміністратори можуть створювати унікальні списки, які можуть мати обкладинку та розділи для групування слів.

Таблиця 2.7

Опис прецеденту «Перегляд статистики»

Назва варіанту використання	Перегляд статистики
Основні актори	Авторизовані користувачі, адміністратор
Короткий опис	Користувачі можуть переглядати статистику по власно пройденим словам і спискам за вибраний період.

Таблиця 2.8

Опис прецеденту «Додавання/видалення списків до/з улюблених»

Назва варіанту використання	Додавання/видалення списків до/з улюблених
Основні актори	Авторизовані користувачі, адміністратор
Короткий опис	Користувачі можуть додавати/видаляти списки до/з улюблених, з метою вивчення їх в подальшому

Таблиця 2.9

Опис прецеденту «Зміна мови додатка»

Назва варіанту використання	Зміна мови додатка
Основні актори	Неавторизовані користувачі, авторизовані користувачі, адміністратор
Короткий опис	Користувачі можуть змінити інтерфейс і описи до слів на українську або англійські мови. Англійський варіант посиляється на методологію Пола Нейшна, використану для створення додатку, одним із пунктом якої є повне занурення в англійське середовище [28].

Основна ціль розробки платформи для вивчення англійських слів полягає в створенні інтерактивного онлайн-середовища, яке дозволяє користувачам ефективно вивчати нові слова за допомогою сучасних технологій та методів навчання. Платформа спрямована на забезпечення зручного та інтуїтивного інтерфейсу, який сприятиме активному залученню користувачів у процес навчання та надасть їм можливість персоналізувати свої навчальні плани.

Основна мета полягає у створенні платформи для вивчення англійських слів з використанням сучасних технологій та інтеграцією методології Пола

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

Найшна для ефективного навчання. Платформа надає можливість користувачам здійснювати пошук слів за допомогою різних фільтрів, створювати списки слів у вигляді флеш-карток, переглядати детальну інформацію про слова, а також відслідковувати свій прогрес за допомогою статистики, змінювати мову додатку. Завдяки цьому, основною метою є створення умов для успішного і ефективного навчання англійської мови, що допоможе користувачам покращити свої мовні навички, що можуть мати значний вплив на їхню освіту, кар'єру та особистий розвиток.

Вимоги користувачів

Внутрішній користувач – адміністратор

1. Створення унікальних списків слів з розділами та обкладинкою.
2. Авторизація.
3. Створення простих списків слів у вигляді флеш-карток.
4. Перегляд/вивчення списків слів.
5. Можливість редагування/видалення своїх списків
6. Можливість додати/видалити списки до/з улюблених
7. Перегляд детальної інформації про слова.
8. Використання смарт-пошуку слів.
9. Перегляд статистики по пройденим словам і спискам.
10. Перегляд словника за алфавітом
11. Зміна мови додатку

Внутрішній користувач – авторизований користувач

1. Авторизація.
2. Створення простих списків слів у вигляді флеш-карток.
3. Перегляд/вивчення списків слів.
4. Можливість редагування/видалення своїх списків
5. Можливість додати/видалити списки до/з улюблених
6. Перегляд детальної інформації про слова.
7. Використання смарт-пошуку слів.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

8. Перегляд статистики по пройденим словам і спискам.
9. Перегляд словника за алфавітом
10. Зміна мови додатку

Внутрішній користувач – неавторизований користувач

1. Перегляд/вивчення списків слів.
2. Перегляд словника за алфавітом.
3. Перегляд детальної інформації про слова.
4. Використання смарт-пошуку слів.
5. Зміна мови додатку

Функціональні вимоги

1. В системі повинна бути представлена можливість авторизації користувачів двома способами: за допомогою звичайної авторизації та за допомогою Google акаунту.
2. Можливість збереження інформації про слова та списки.
3. Забезпечити користувачам переглядати статистику по пройденим словам і спискам.

Нефункціональні вимоги

1. **Сприйняття**
 - Час, потрібний для освоєння веб-сайту – від 3-х до 5-ти хвилин, для досвідчених користувачів – менше 2-х хвилин.
 - Час відповіді веб-сайту на дії користувача не повинен перевищувати 3 секунд.
 - Інтерфейс веб-сайту повинен бути інтуїтивно зрозумілим та зручним для користувача, та не вимагати від нього додаткової підготовки.
2. **Надійність**
 - Веб-сайт має бути доступний цілодобово.
3. **Продуктивність**

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

- Велика кількість користувачів мають змогу переглядати веб ресурс одночасно.

Системні вимоги

1. Вимоги до середовища виконання

- Платформа повинна задовольняти вимоги на пристроях з наступною мінімальною комплектацією:
 - 2 Гб оперативної пам'яті.
 - Процесор з тактовою частотою не нижче 1.4 GHz.
 - Вихід в мережу Інтернет.

2. Вимоги до серверу інформаційної системи

- У ядрі системи повинна бути документаційно орієнтована база даних у зв'язці з орієнтованим веб додатком для забезпечення збереження великих даних (слова та інформація про них).
- Підтримка сервером технологій Laravel [29].

Аналіз функціональних вимог дозволяє виділити наступні сутності, що забезпечать програмну реалізацію додатку. Наведено Діаграму класів, яка використовується для його роботи (рис. 2.2).

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

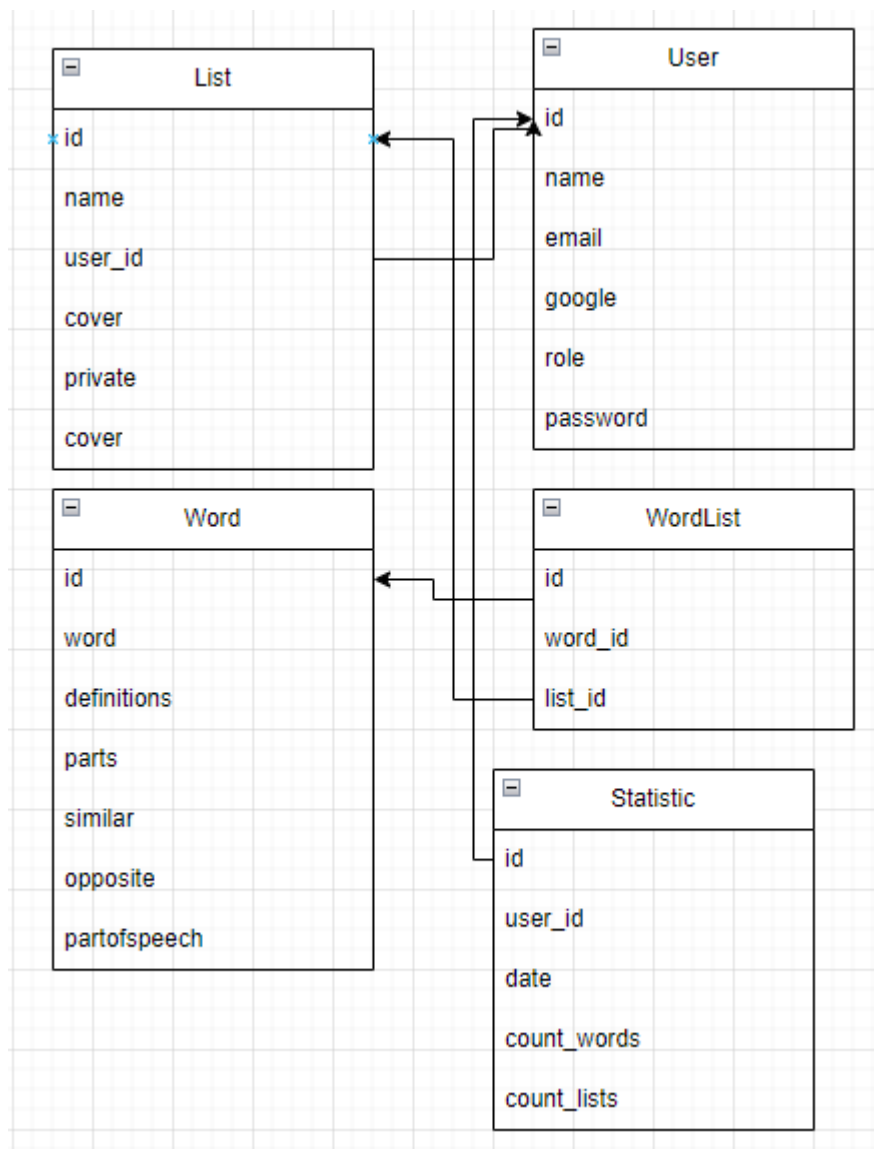


Рис. 2.2 Діаграма класів

User – клас, який відповідає за користувача

- register () – відповідає за реєстрацію.
- login () – відповідає за авторизацію.
- getProfile() – відповідає за отримання інформації профілю користувача.
- updateProfile() – відповідає за оновлення інформації профілю користувача.

Word – клас, який відповідає за слово

- getWordDetails() – відповідає за отримання детальної інформації про слово.
- getRandomWord() – відповідає за отримання випадкового слова.

- `getSynonyms()` – відповідає за отримання синонімів слова.
- `getAntonyms()` – відповідає за отримання антонімів слова.
- `getExamples()` – відповідає за отримання прикладів використання слова.
- `getDictionary()` – відповідає за перегляд словника.

WordList – клас, який відповідає за списки слів

- `createWordList()` – відповідає за створення списку слів користувачем.
- `addWordToList()` – відповідає за додавання слова до списку.
- `addListToFavourite()` – відповідає за додавання списку до улюблених.
- `removeListFromFavourite ()` – відповідає за видалення списку з улюблених.
- `removeWordFromList()` – відповідає за видалення слова зі списку.
- `getWordLists()` – відповідає за отримання всіх списків слів користувача.
- `getPublicWordLists()` – відповідає за отримання всіх публічних списків слів.
- `addChapter()` – відповідає за додавання розділу до списку

Statistics – клас, який відповідає за статистику

- `getWordStatistics()` – відповідає за отримання статистики по пройденим словам.
- `getListStatistics()` – відповідає за отримання статистики по пройденим спискам слів.
- `getUserStatistics()` – відповідає за отримання загальної статистики користувача.

2.2 Розробка бази даних системи

Для збереження файлів та інформації у додатку використовуються MySQL, це система управління базами даних, яка базується на реляційній моделі [30].

База даних системи складається з чотирьох таблиць:

- Users – таблиця, яка зберігає дані користувачів.
- Words – таблиця, яка зберігає інформацію про слова.
- Lists – таблиця, яка зберігає списки.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

- WordLists – таблиця, яка зберігає слова в списках.
- Statistics – таблиця, яка зберігає статистичні дані про користувачів.

Опис кожної з таблиць бази даних наведений у (табл. 2.10) – (табл. 2.14)

Таблиця 2.10

Опис таблиці Users

Поле	Тип	Призначення
id	int	Ідентифікатор
name	varchar	Ім'я користувача
email	varchar	Email користувача
password	varchar	Пароль користувача

Таблиця 2.11

Опис таблиці Words

Поле	Тип	Призначення
id	int	Ідентифікатор
word	varchar	Слово
definition	text	Значення слова
similar	text	Синоніми
opposite	text	Антоніми
examples	text	Приклади використання

Таблиця 2.12

Опис таблиці Lists

Поле	Тип	Призначення
id	int	Ідентифікатор
name	varchar	Назва слова
user_id	int	Ідентифікатор автора
cover	varchar	Обкладинка

private	bool	Приватність
---------	------	-------------

Таблиця 2.13

Опис таблиці WordLists

Поле	Тип	Призначення
id	int	Ідентифікатор
word_id	int	Ідентифікатор Слова
list_id	int	Ідентифікатор Списку

Таблиця 2.14

Опис таблиці Statistics

Поле	Тип	Призначення
id	int	Ідентифікатор
date	date	Дата
user_id	int	Ідентифікатор юзера
count_words	int	Кількість слів
count_lists	int	Кількість списків

Використання реляційної бази даних MySQL для збереження інформації на платформі вивчення англійських слів забезпечує надійне та ефективне управління даними. Структура бази даних (рис. 2.3) дозволяє зберігати дані про користувачів, слова, списки слів та статистику, що забезпечує зручність та функціональність платформи. Інтеграція різних таблиць дає можливість реалізувати всі необхідні функції для ефективного навчання та відстеження прогресу користувачів.



Рис. 2.3 Діаграма бази даних

2.3 Проектування та реалізація алгоритмів роботи системи

Додаток є простим у використанні, має всі основні функції платформи, що дозволяє користувачам легко взаємодіяти з платформою. Зображено алгоритм роботи з платформою (рис. 2.4), також описано детально його пункти.

Коли ми переходимо на головну сторінку ми бачимо сторінку з пошуком, який вбудований в хедер і інформацію випадкового слова. Випадкове слово вибирається з слів в яких частота використання більша за 5.

Далі в нас є вибір перейти до словника зі всіма словами, перейти до списку всіх списків з словами або авторизуватися. Якщо ми не авторизовані ми можемо тільки вчити вибраний список слів. Якщо авторизовані у нас з'являється такі вкладки: Мої листи, Улюблені листи, Статистика.

Якщо користувач адміністратор, тоді він може створити як унікальний список з обкладинкою та розділами, так і простий список; простий користувач може створити тільки простий список.

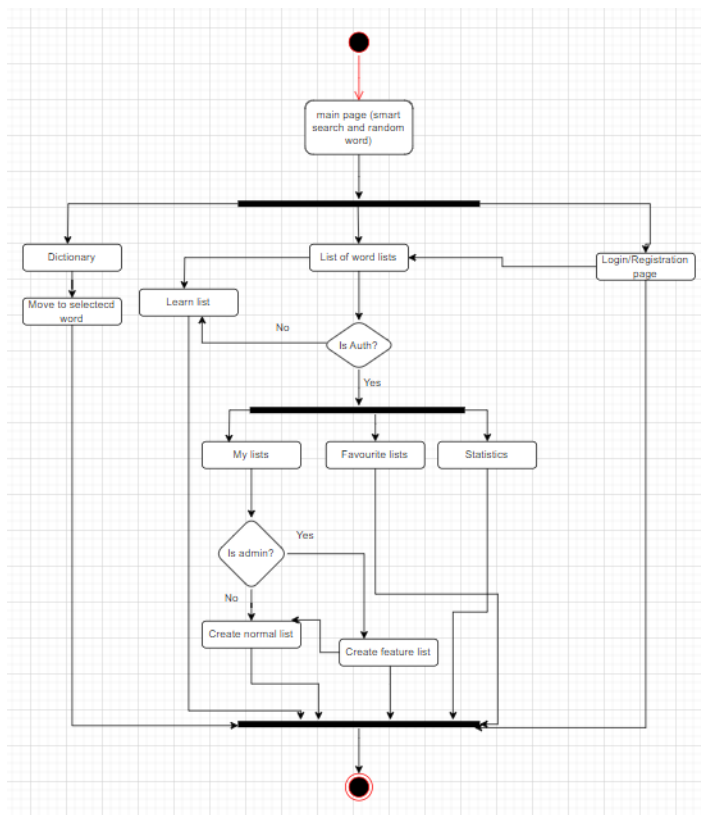


Рис. 2.4 Діаграма активності

2.4 Реалізація функціоналу платформи для вивчення англійських слів

Реалізація функціоналу платформи для вивчення англійських слів включає кілька основних етапів та компонентів, що забезпечують її роботу та інтерактивність для користувачів.

Реєстрація та авторизація користувачів

Платформа забезпечує два способи авторизації користувачів: за допомогою звичайної авторизації (логін та пароль) та через інтеграцію з Google акаунтом.

- Реєстрація: Користувачі можуть зареєструватися, надавши необхідні дані, такі як ім'я користувача, пароль та електронна пошта та інші. (приклад. 2.1)

- Авторизація: Після реєстрації користувачі можуть увійти в систему, використовуючи свої облікові дані.

Приклад 2.1 Код для функції Реєстрація користувача

```
public function register(Request $request)
{
    $validatedData = $request->validate([
        'username' => 'required|string|max:255',
        'email' => 'required|string|email|max:255|unique:users',
        'password' => 'required|string|min:8|confirmed',
    ]);

    $user = User::create([
        'username' => $validatedData['username'],
        'email' => $validatedData['email'],
        'password' => Hash::make($validatedData['password']),
    ]);

    return response()->json(['message' => 'User registered successfully', 'user' =>
    $user], 201);
}
```

Словник та управління словами

- Перегляд детальної інформації про слово: Користувачі можуть переглядати значення слова, його синоніми, антоніми, приклади використання та інші мовні аспекти. (приклад. 2.2)
- Смарт-пошук слів: Платформа забезпечує можливість пошуку слів за різними фільтрами, такими як частина мови, складність, категорія тощо. (приклад. 2.3)
- Словник: Платформа забезпечує можливість перегляду словника за алфавітом. (приклад. 2.4)

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

Приклад 2.2 Код для функції Отримання інформації про слово

```
public function getWordDetails($wordId)
{
    $word = Word::findOrFail($wordId);
    return response()->json($word);
}
```

Приклад 2.3 Код для функції Отримання масиву слів підходящих для
фільтрів

```
public function searchWords(Request $request)
{
    $words = Word::getWordByFilter($request);
    return response()->json($words);
}
```

Приклад 2.4 Код для функції Отримання словника для конкретної літери

```
public function dictionary(Request $request)
{
    $words = Word::getWordsStartWith($request);
    return response()->json($words);
}
```

Управління списками слів

- Створення списків слів: Користувачі можуть створювати власні списки слів для навчання. (приклад. 2.5)
- Додавання слів до списків: Користувачі можуть додавати слова до своїх списків. (приклад. 2.6)
- Перегляд списків: Користувачі можуть переглядати свої списки слів та публічні списки, створені іншими користувачами або адміністраторами. (приклад. 2.7)

- Додавання розділів до списків: Адміністратор може організовувати слова в списках, додаючи розділи для кращого структурування матеріалу. (приклад. 2.8)

Приклад 2.5 Код для функції Створення списків слів

```
public function createList(Request $request)
{
    $list = List::create(["name" => $request-> name, 'user_id' => auth()->id()]);
    return response()->json($list);
}
```

Приклад 2.6 Код для функції Додавання слів до списків

```
public function addWordToList(Request $request)
{
    $wordList = WordList::create([
        'list_id' => $request->list_id,
        'word_id' => $request->list_id,
    ]);
    return $wordList;
}
```

Приклад 2.7 Код для функції Перегляд списків

```
public function getLists(Request $request)
{
    $lists = List::all();
    return response()->json($lists);
}
```

Приклад 2.8 Код для функції Додавання розділів до списків

```
public function addChapter(Request $request)
{
    $chapter = Chapter::create([
```

```

        "list_id" => $request->list_id,
        "words" => $request->words
    ]);
    return response()->json($chapter);
}

```

Перегляд статистики

Користувачі можуть переглядати свою статистику по пройденим словам і спискам, що допомагає відстежувати прогрес у навчанні. (приклад. 2.9)

Приклад 2.9 Код для функції Вивід статистики по юзеру

```

public function getStatisticByUser(Request $request)
{
    $statistic = Statistic::where('user_id', auth()->id())->get();
    return response()->json($statistic);
}

```

Зміна мови додатку

Користувачі можуть переглядати свою статистику по пройденим словам і спискам, що допомагає відстежувати прогрес у навчанні. (приклад. 2.10)

Приклад 2.10 Код для функції Вивід статистики по юзеру

```

public function changeLocale(Request $request)
{
    $lang = $request->input('language', 'en')
    session(['applocale' => $lang])
    return response()->json($lang);
}

```

Підсумовуючи, було створено діаграму компонентів платформи для вивчення англійських слів (рис. 2.5) і визначено основні компоненти збору.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

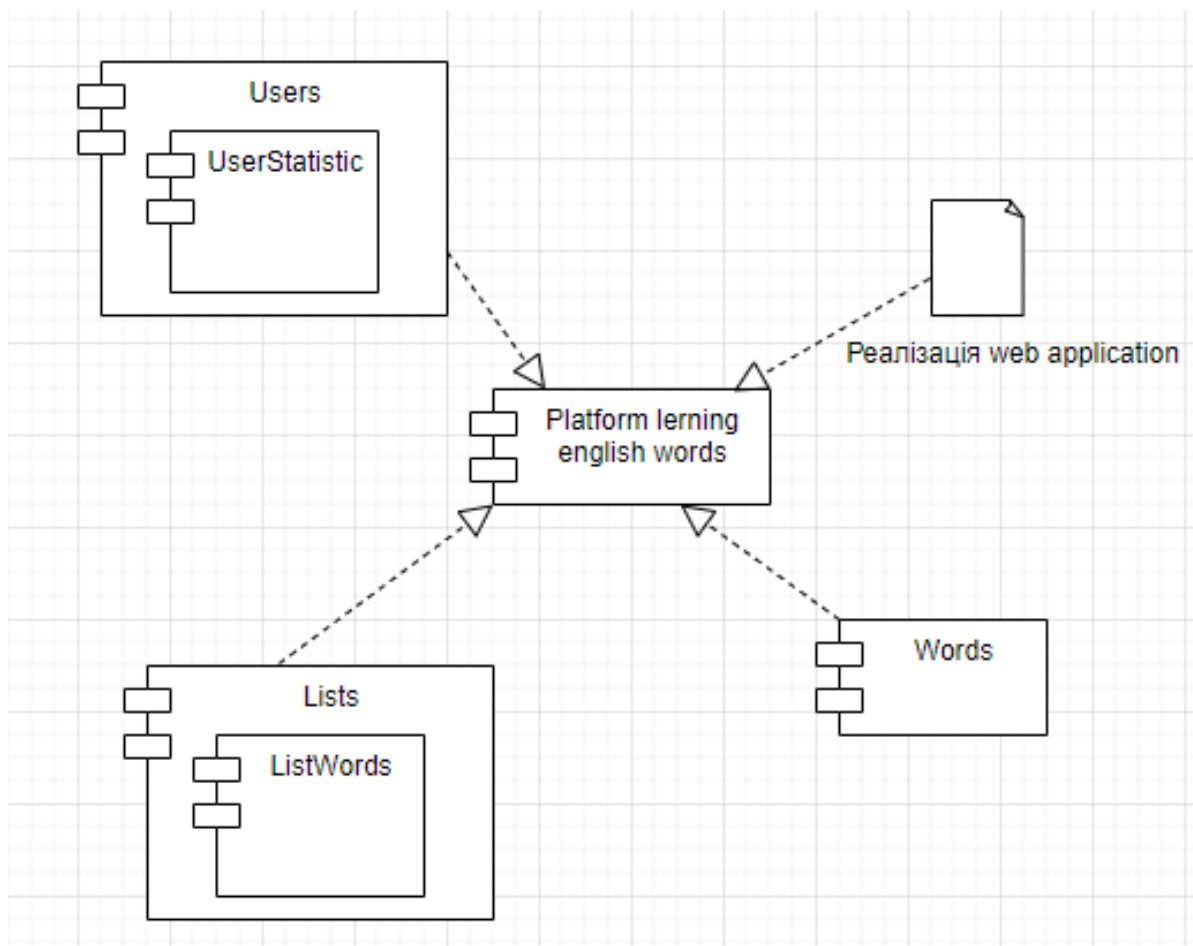


Рис. 2.5 Діаграма компонентів

На діаграмі визначені такі елементи:

- Platform learning english words– компонент платформи;
- Words – компонент слів, в ньому зберігається основна база слів і інформація про них;
- Lists – компонент списку, в ньому також зберігається компонент ListWords
- це слова які відносяться до конкретного списку;
- Users – компонент який зберігає інформацію про користувача, також в ньому зберігається компонент UserStatistic, який відповідає за статистику для конкретного юзера;

Висновки до другого розділу

В процесі проектування платформи для вивчення англійських слів були проаналізовані всі варіанти використання програмного продукту відповідно до вимог програмного забезпечення. На основі отриманого результату була спроектована структура платформи. Проектування включало в себе моделювання структури проекту та взаємодії між їх складовими частинами.

Також було спроектовано базу даних для зберігання інформації про користувачів, слова, списки слів та статистику. Ця база даних забезпечує надійне та ефективне управління даними, що дозволяє користувачам взаємодіяти з платформою та використовувати її функціонал.

Було описано алгоритм роботи з додатком та описані послідовні пункти варіантів його використання. Це включало в себе реєстрацію та авторизацію користувачів, перегляд та управління словами і списками слів, а також відстеження статистики.

Також був наведений опис основного функціоналу додатку – це створення та управління списками слів, додавання нових слів, перегляд детальної інформації про слова, а також відстеження прогресу користувачів через статистику. Були розглянуті основні функції, які забезпечують зручність та ефективність використання платформи для навчання англійської мови.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ПЛАТФОРМОЮ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКИХ СЛІВ

3.1 Порядок встановлення та налаштування платформи

Встановлення та налаштування платформи вимагає розгортання всіх його компонентів та налаштування як кожного елементу окремо, так і загальної системи в цілому. Для опису архітектури платформи було створено діаграму розгортання (рис. 3.1).

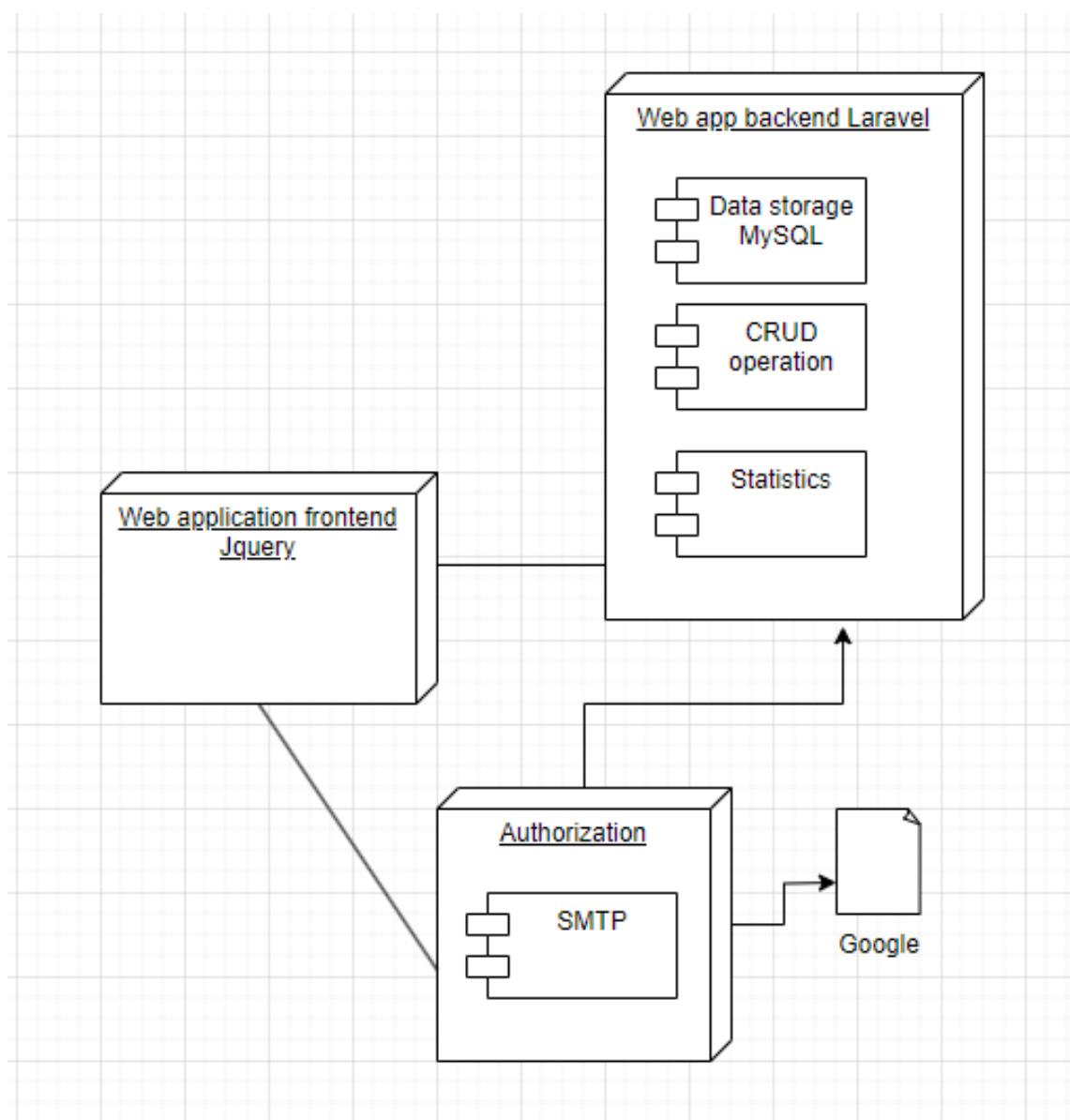


Рис. 3.1 Діаграма розгортання платформи для вивчення англійських слів

Опис процес розгортання.

WEB APP BACKEND – розгортання серверної частини додатку, для зберігання та проведення CRUD операцій з різними сутностями додатку, також для зберігання статистики по користувачам, використовується фреймворк php Laravel.

WEB APP FRONTEND – розгортання клієнтської частини додатку яка відповідає за візуалізацію даних та обміном даних між серверною частиною та клієнтською, використовується JavaScript та його бібліотека JQuery.

Authorization – розгортання авторизації, використовується 2 шляхи, або звичайний через email, password або через Google account. Також потрібно задати свої параметри SMTP сервера для відправок електронних листів користувачам, в випадках відновленню паролю та інших.

3.2 Структура інтерфейсу платформи для вивчення англійських слів

При переході на нашу платформу користувача зустрічає головна сторінка, яка реалізована як випалкове популярне слово з інформацією про нього, також присутній хедер в якому розташовується логотип, сторінка з списками слів, словник слів, вибір мови та можливість авторизуватися на сайті. Також в хедер вбудований розумний пошук слів для швидкого знаходження потрібного слова. (рис. 3.2)

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

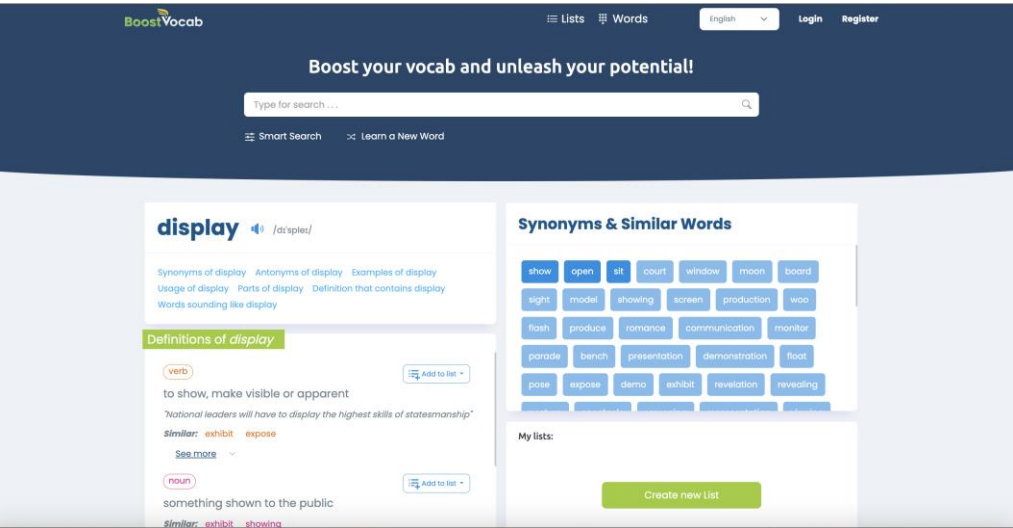


Рис. 3.2 Початкова сторінка

Далі розглянемо сторінку реєстрації. Нас зустрічає форма заповнення з полями Ім'я, Прізвище, Дату народження, Електронна пошта, Пароль і Підтвердження паролю. Також якщо в нас вже є акаунт одразу під формою можна швидко перейти на сторінку входу. Ми можемо зареєструватися за допомогою вище перерахованих полів, а також за допомогою Google акаунта, що спрощує процес реєстрації для користувачів. Звичайно присутня валідація на розмір паролю, раніше створений акаунт за введеною поштою, перевірка підтвердження паролю і перевірка на виповнення 13-річчя користувача з повідомленням про введення електронної пошти одного з батьків. Присутня захист від ботів з допомогою Google reCAPTCHA v2. (рис. 3.3)

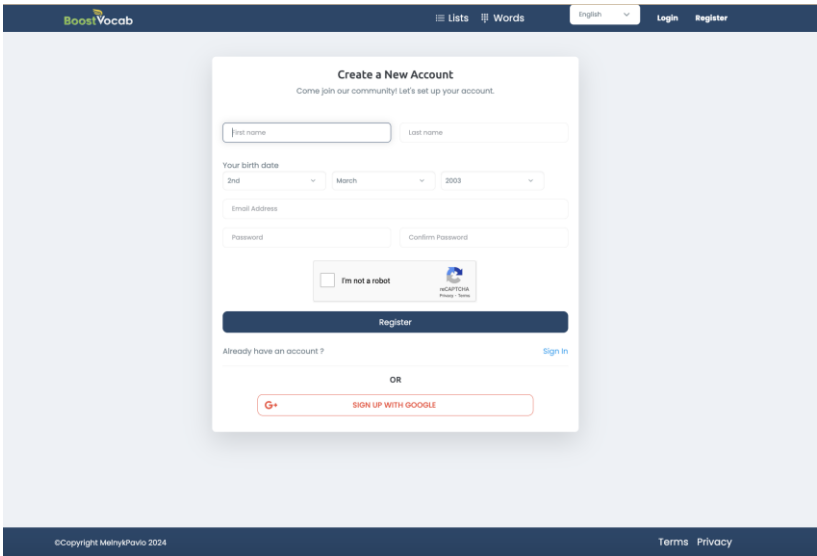


Рис. 3.3 Реєстрація

Сторінка Входу реалізовано простою формою з полями вводу електронної пошти і паролем, також з можливістю входу через Google акаунт (рис. 3.4). Також на формі є швидкий перехід на сторінку реєстрації і перехід на сторінку відновлення паролю.

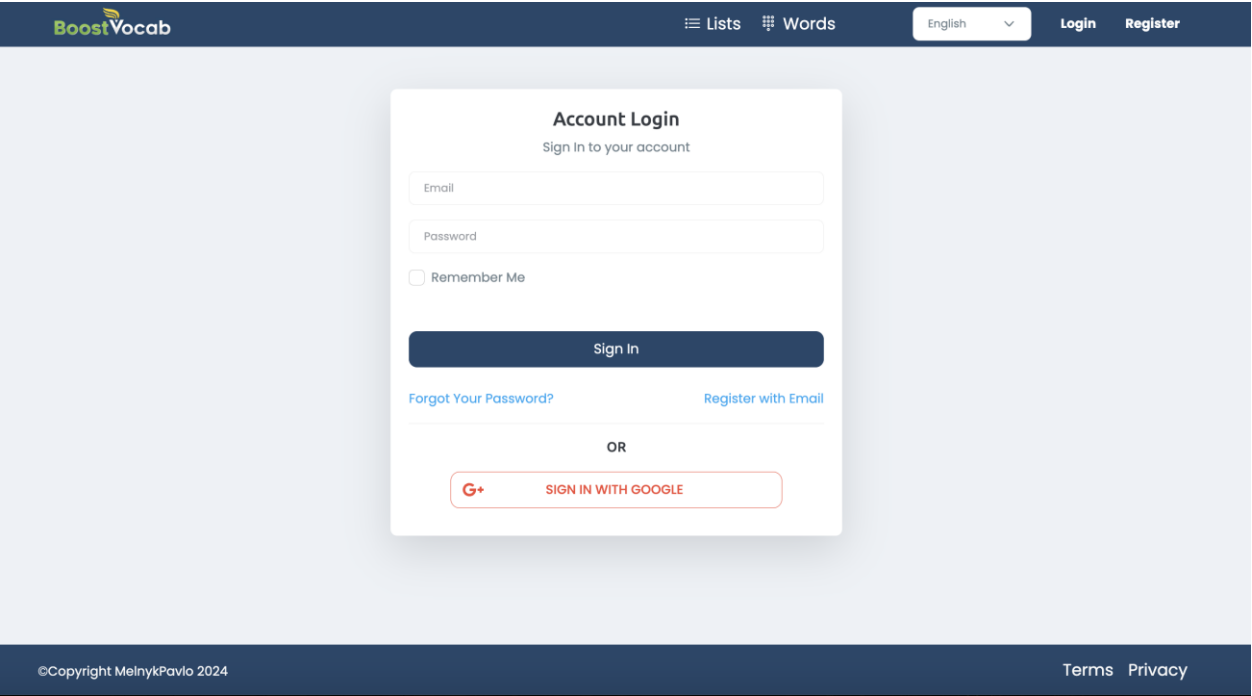


Рис. 3.4 Реєстрація

Процес відновлення паролю полягає з декількох етапів:

- Введення на сторінці відновлення паролю пошти за якою зареєстрований акаунт (рис. 3.5)
- Отримання листа в поштову скриньку від нашої платформи з посиланням на встановлення нового паролю (рис. 3.6)
- Перехід за посиланням і встановленням нового паролю від платформи (рис. 3.7)

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

BoostVocab

Lists Words English Login Register

Boost your vocab and unleash your potential!

Type for search...

Smart Search Learn a New Word

Reset Password

Email Address

Send Password Reset Link

Рис. 3.5 Сторінка введення пошти для відновлення паролю

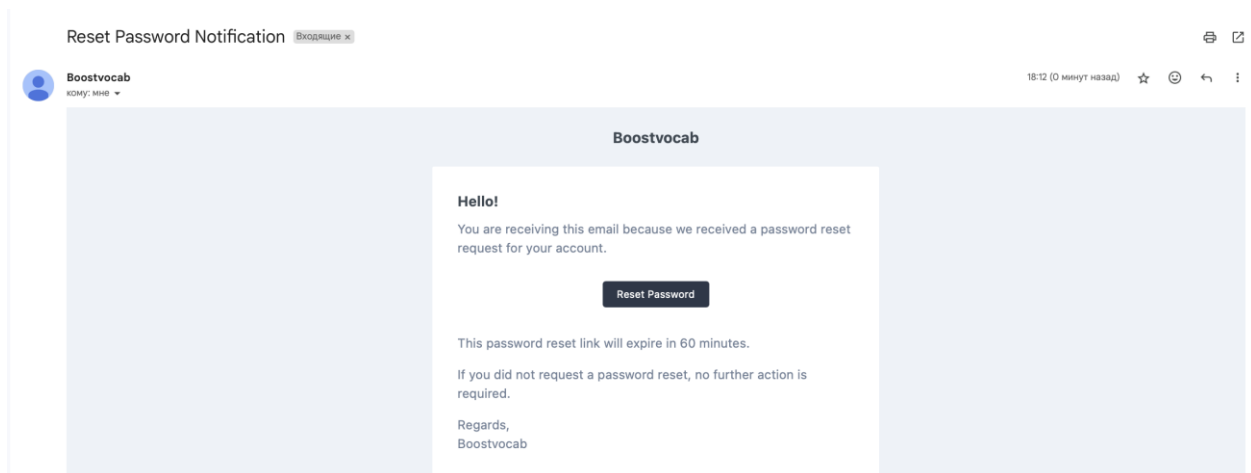


Рис. 3.6 Приклад електронного листа від додатку

BoostVocab

Lists Words English Login Register

Boost your vocab and unleash your potential!

Type for search...

Smart Search Learn a New Word

Reset Password

Email Address [Resend Email](#)

Password

Confirm Password

Reset Password

Рис. 3.7 Встановлення нового паролю

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

Розглянемо смарт пошук слів (рис. 3.8). Він представляє собою поле для введення тексту і форму для введення фільтрів, а саме вибір частини мови, діапазон кількості складів, синоніми або антоніми до введеного слова, закінчується на певний склад, введення типу слова, введення слова, яке має міститися в описі, приклад до введеного слова, частина введеного слова. При виборі деяких фільтрів (рис. 3.9), показується вірний результат (рис. 3.10)

Рис. 3.8 Смарт пошук слів

Рис. 3.9 Введення деяких фільтрів

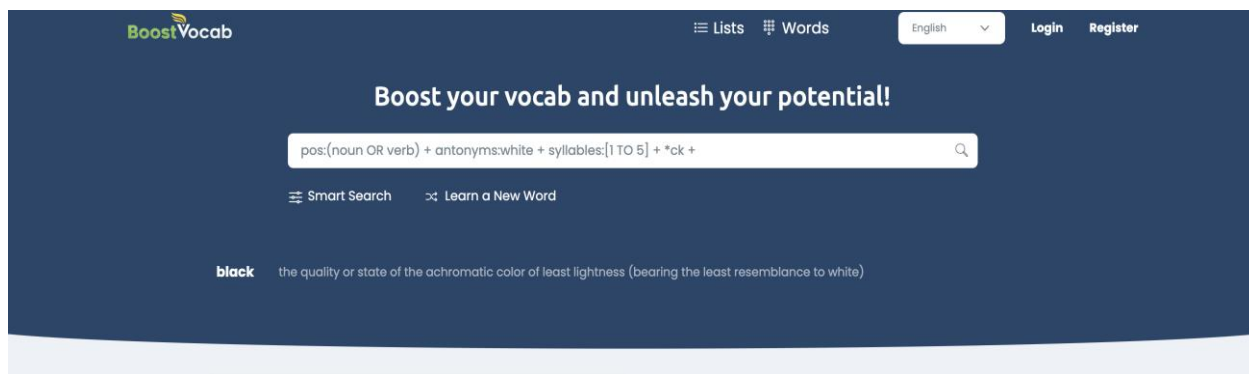


Рис. 3.10 Виведення результату пошуку

Розглянемо сторінку з інформацією про слово (рис. 3.11), вона містить такі блоки:

- Блок з самим словом з транскрипцією, кнопкою аудіо відтворення та перехід на інші корисні сторінки з словом, а саме синоніми слова, антоніми, приклади використання, частини якоїсь групи, описів інших слів, в якому міститься вибране слово та слова, які співзвучні по вимові з вибраним словом (рис. 3.12)
- Блок з вибраною корисною інформацією (рис. 3.13).
- Блок з описами слова, частинами мови, прикладами використання, синонімами, типи до якого відноситься слово та кнопкою додавання слова в свій список (рис. 3.14).
- Блок з моїми списками (рис. 3.15)

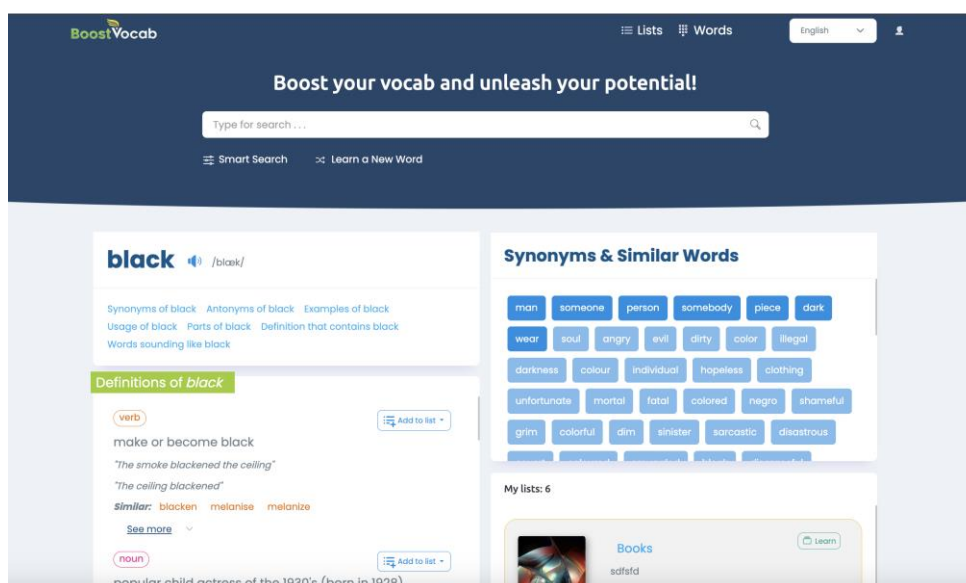


Рис. 3.11 Інформація по слову

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

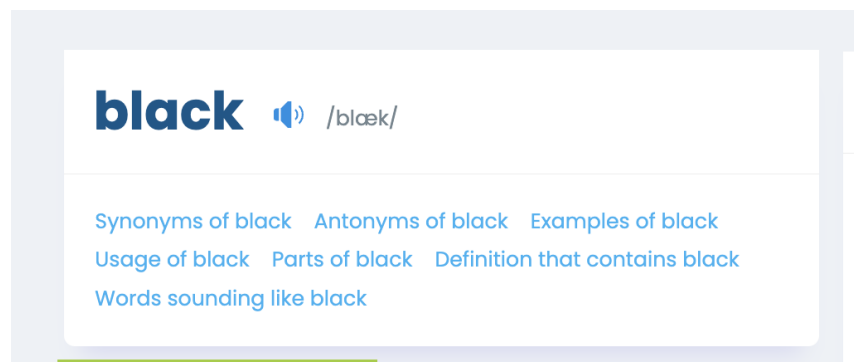


Рис. 3.12 Блок з самим словом



Рис. 3.13 Блок з вибраною інформацією

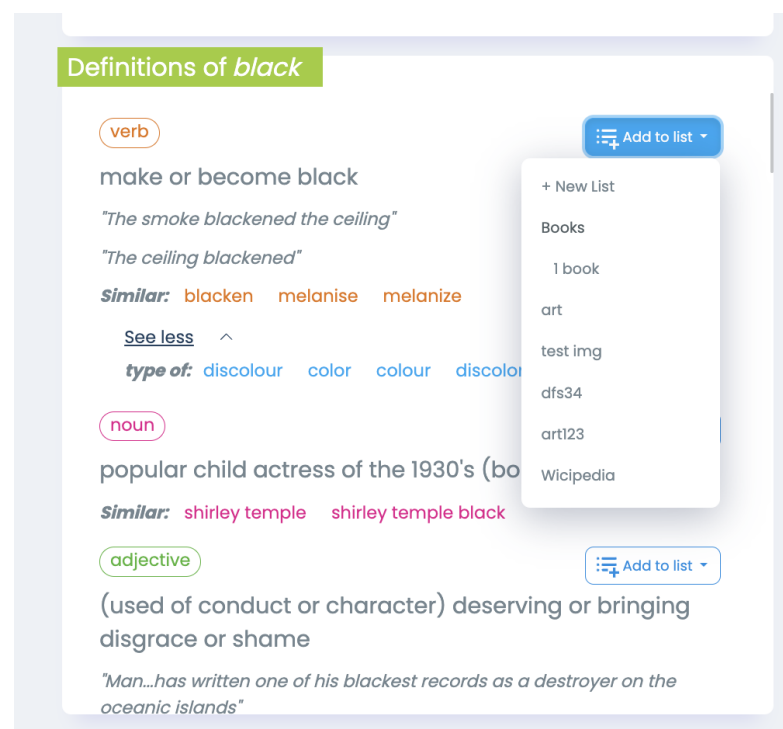


Рис. 3.14 Блок з можливими описами слова

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

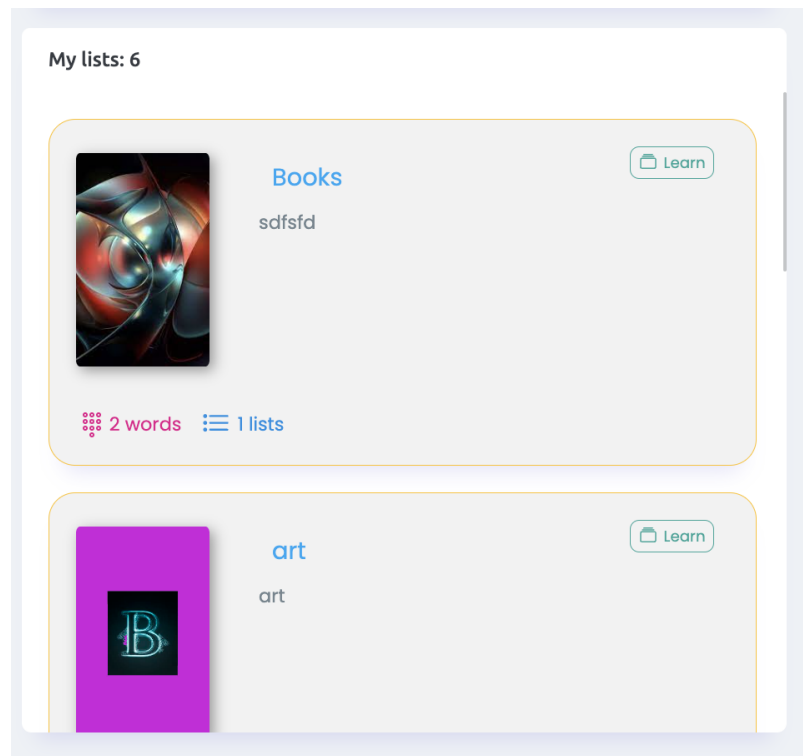


Рис. 3.15 Блок з моїми списками

В області смарт пошуку присутня кнопка Learn a new word, яка веде на сторінку популярного випадкового слова (рис. 3.16)

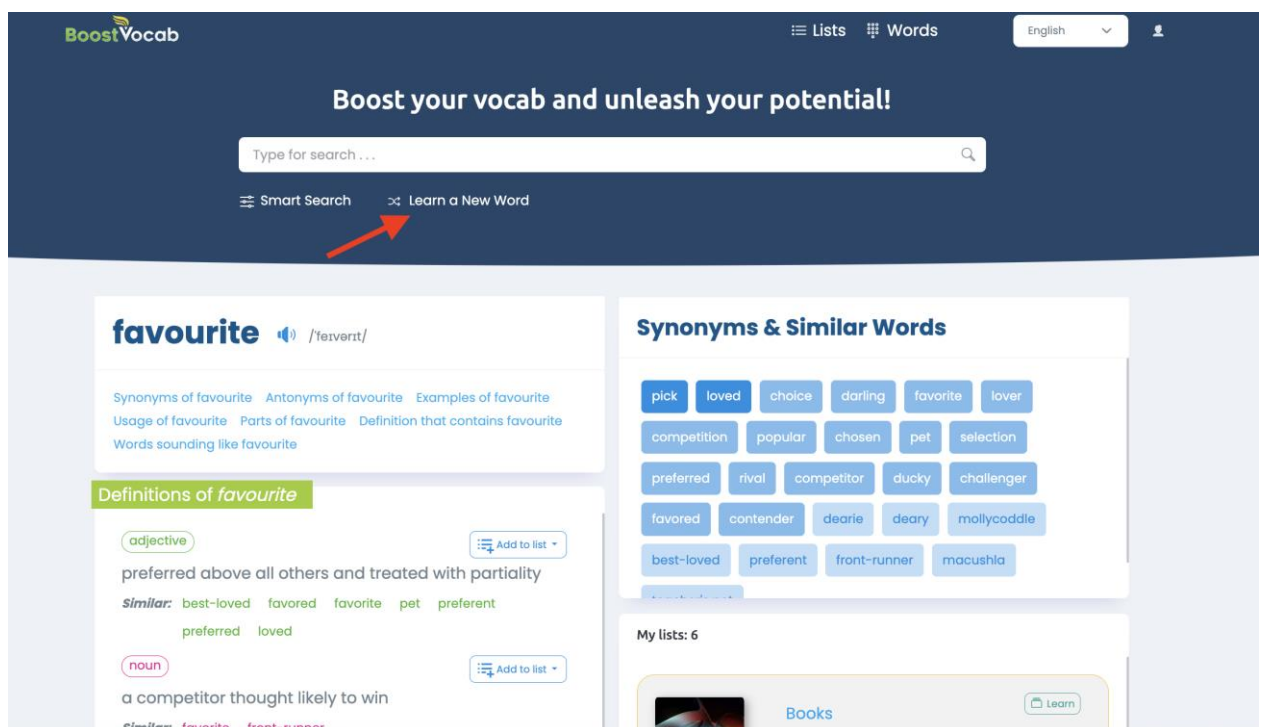


Рис. 3.16 Сторінка вивчення випадкового слова

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

Розглянемо сторінку словника (рис. 3.17). В ньому в верхній частині можна обрати букву, слова, які починаються на котру нас цікавлять. Далі в вигляді блоків йдуть слова за алфавітом з корисними посиланнями, які можуть нас зацікавити. Після основного блоку йде пагінація цих слів та блок з недавно переглянутими словами та моїми списками.

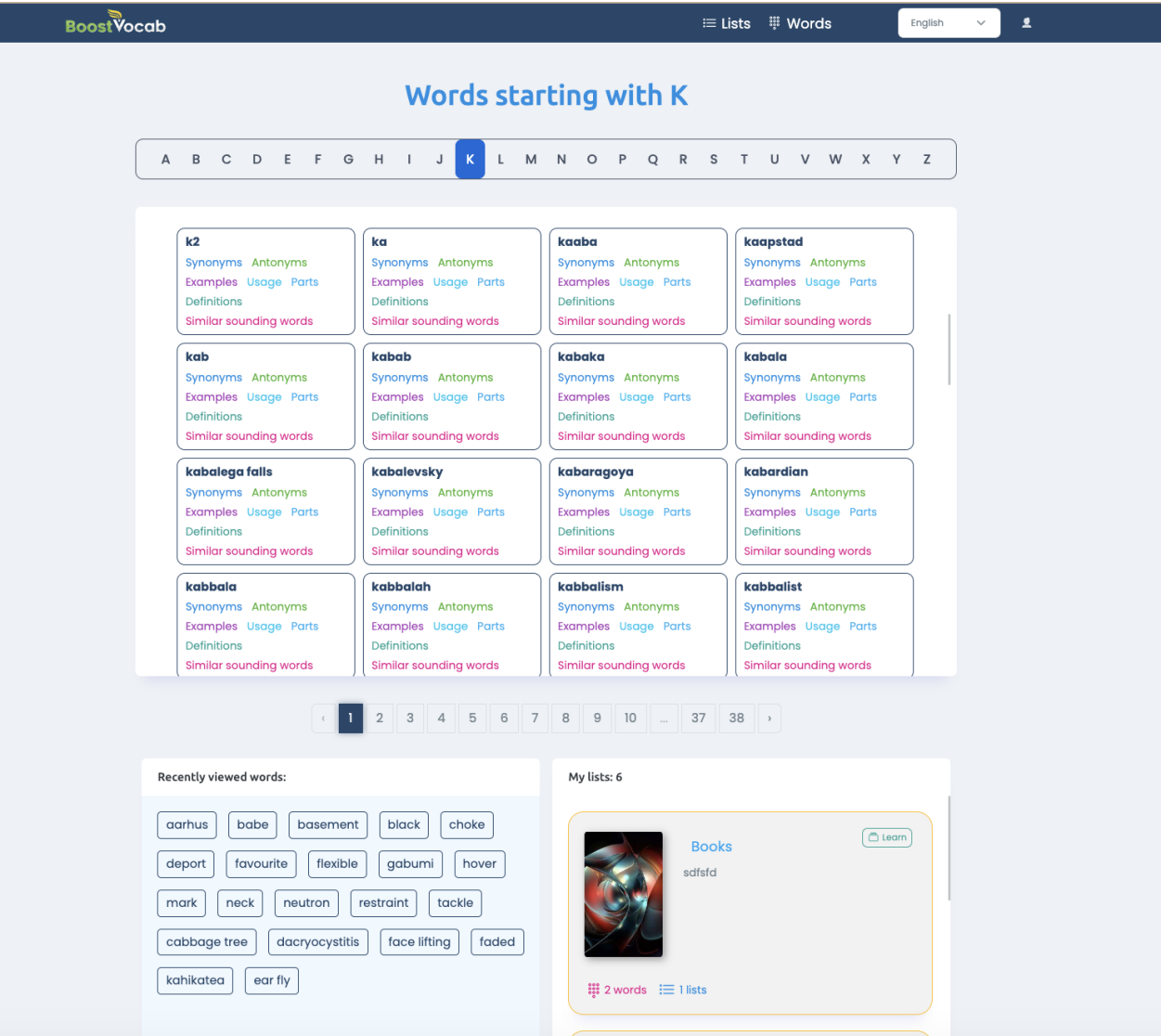


Рис. 3.17 Сторінка словника

Далі перейдемо до основної сторінки платформи - перегляд всіх доступних списків (рис. 3.18). В хедері присутній фільтр за категоріями списку та пошуку по назві. Є можливість Додати до улюблених або перейти до Вивчення списку. Ця сторінка також поділена на блоки, розглянемо кожен:

- Розширені списки (списки, які мають розділи) (рис. 3.19)

- Рекомендовані списки (звичайні списки, які створив адміністратор) з окремою пагінацією для них(рис. 3.20)
- Публічні списки користувачів платформи в сортуванні по новизні) з окремою пагінацією для них (рис. 3.21)

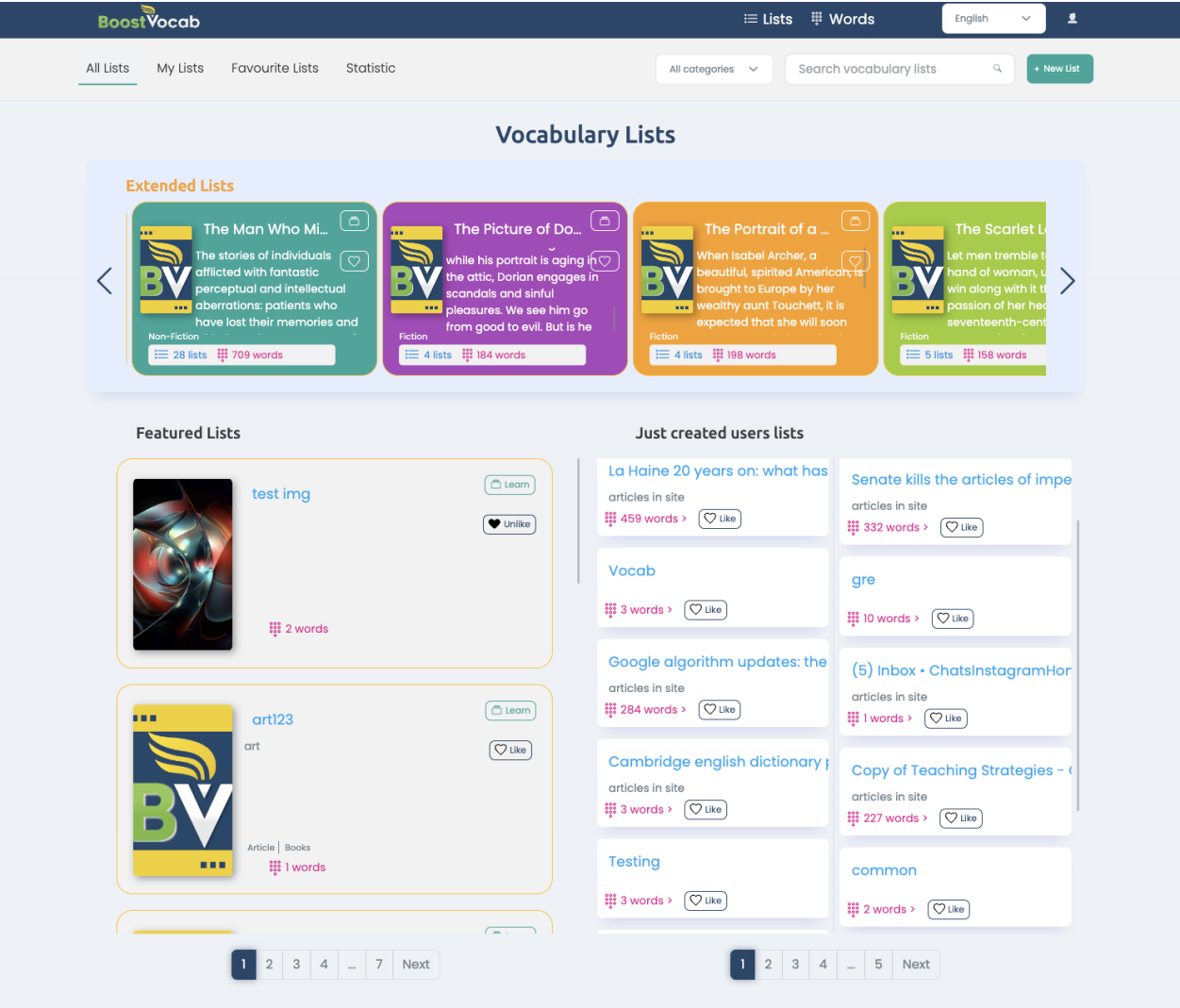


Рис. 3.18 Сторінка всіх списків

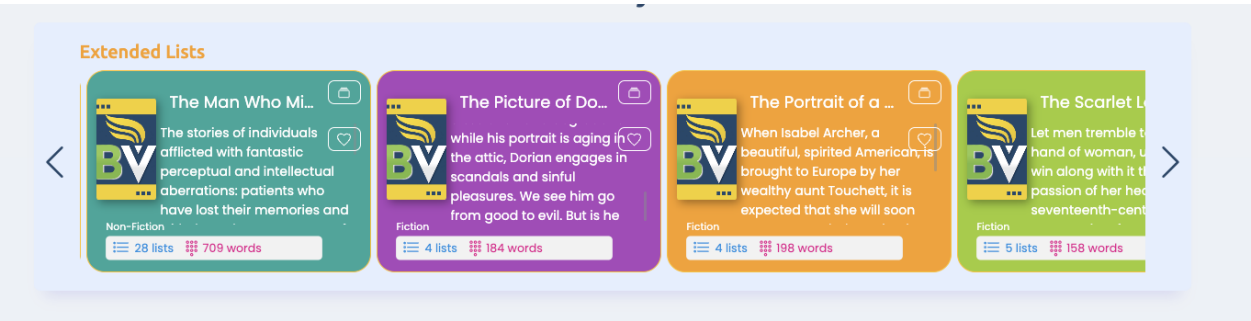


Рис. 3.19 Блок розширених списків

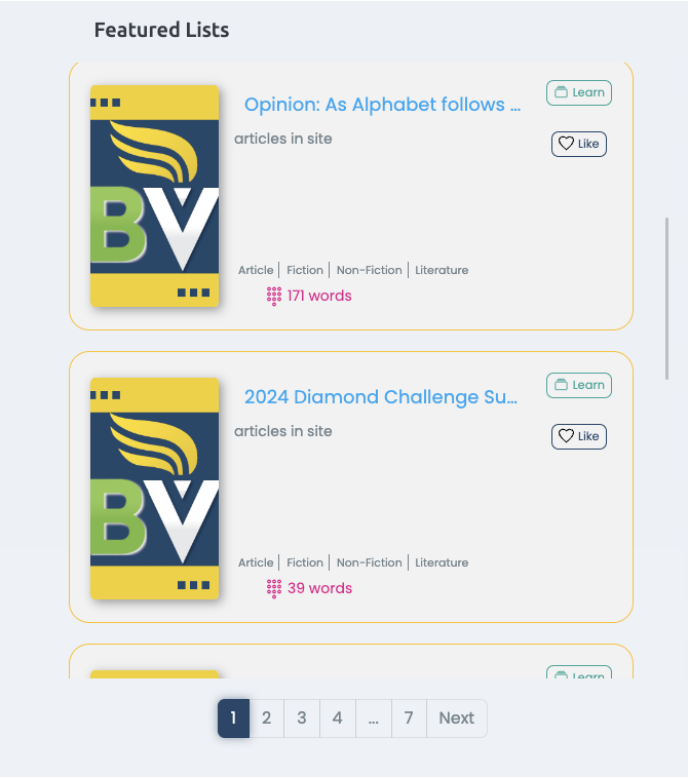


Рис. 3.20 Блок рекомендованих списків

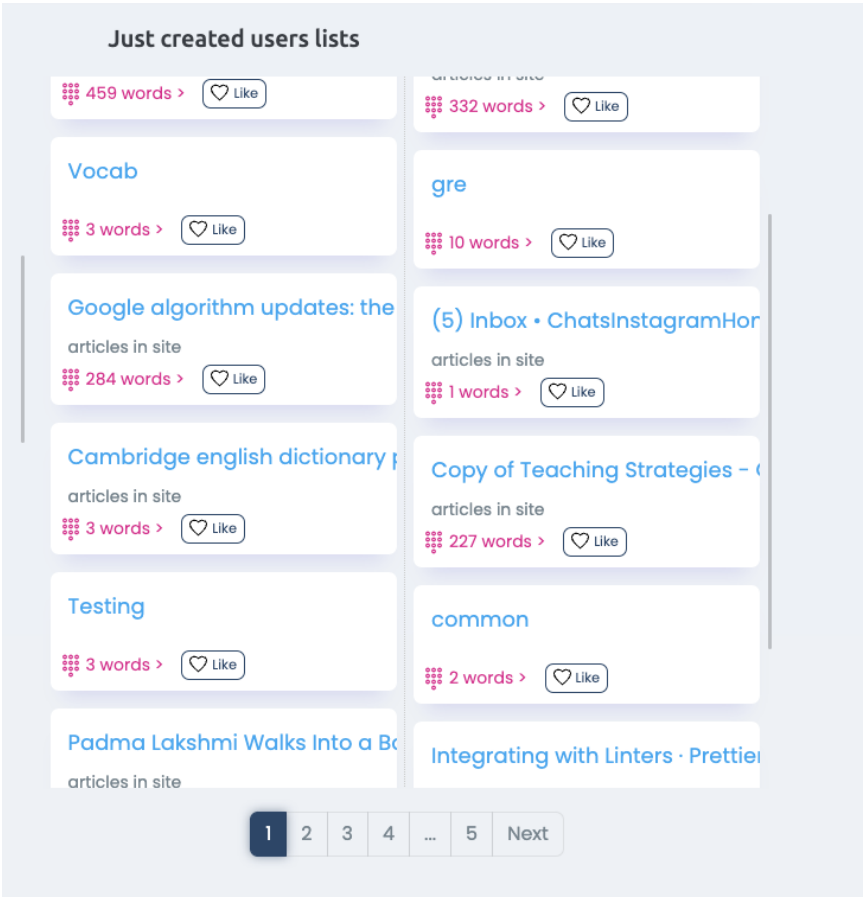


Рис. 3.21 Блок публічних користувацьких списків

Далі по вкладкам в хедері можемо перейти до моїх списків (рис. 3.22). Там з'являється можливість редагувати та видаляти списки свої списки.

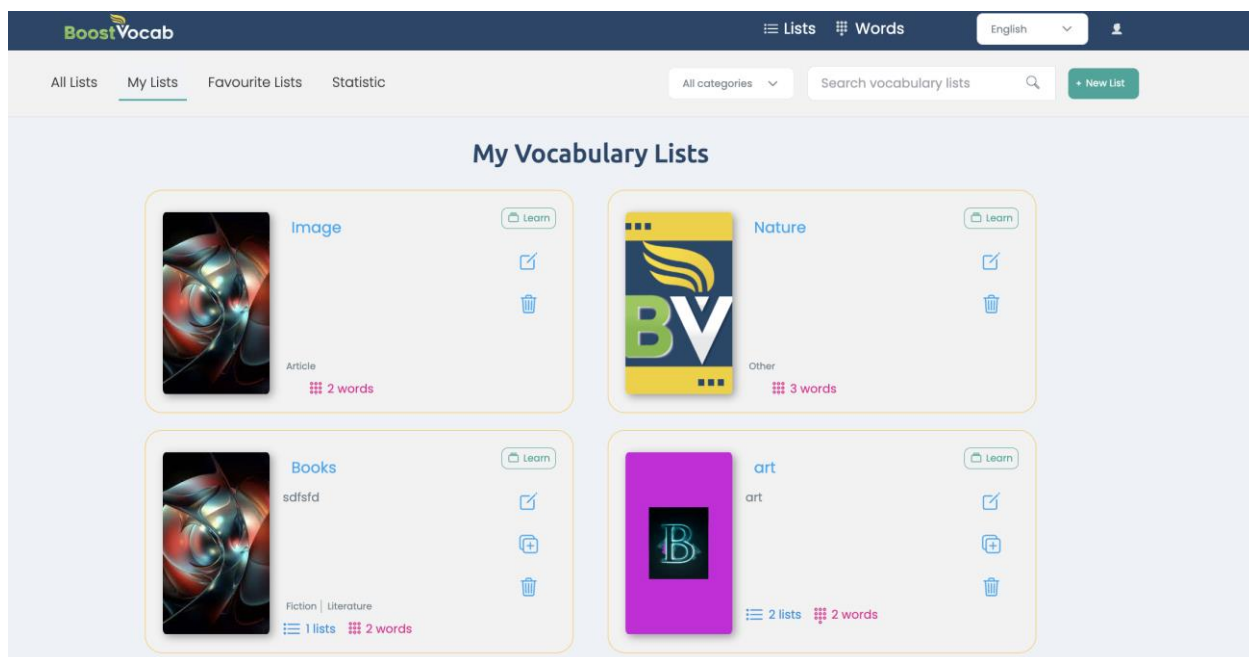


Рис. 3.22 Сторінка моїх списків

Розглянемо процес додавання нового списку, він доступний при кліку в хедері на кнопку + New list. В адміністратора вона відкриває меню вибору списку (рис. 3.23), який він хоче створити (Parent list - це список, який може містити розділи, і Normal list - це звичайний список який доступний звичайному користувачу), в звичайного користувача переходить одразу на форму додавання звичайного списку.

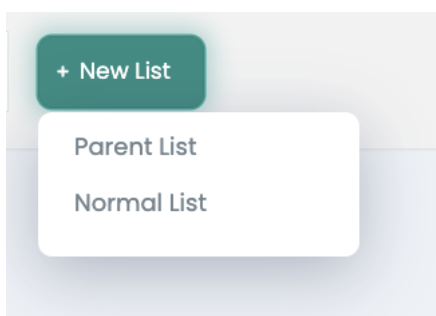


Рис. 3.23 Вибір типу списку для адміністратора

Розглянемо процес додавання Розширеного списку з розділами. Спочатку ми переходимо на створення спільної обкладинки для наших розділів (рис. 3.24),

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

де вводиться назва, опис, категорія (рис. 3.25), картинка, та вибір приватним чи публічним буде список.

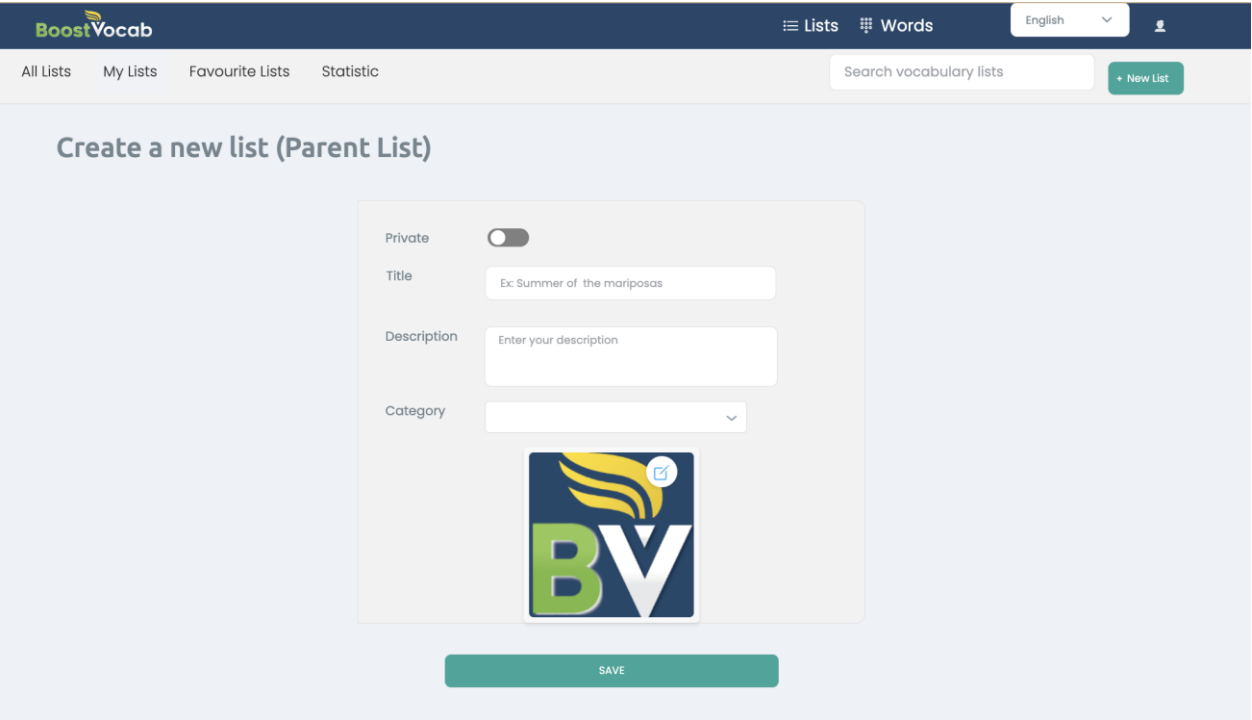


Рис. 3.24 Сторінка створення обкладинки до розширеного списку

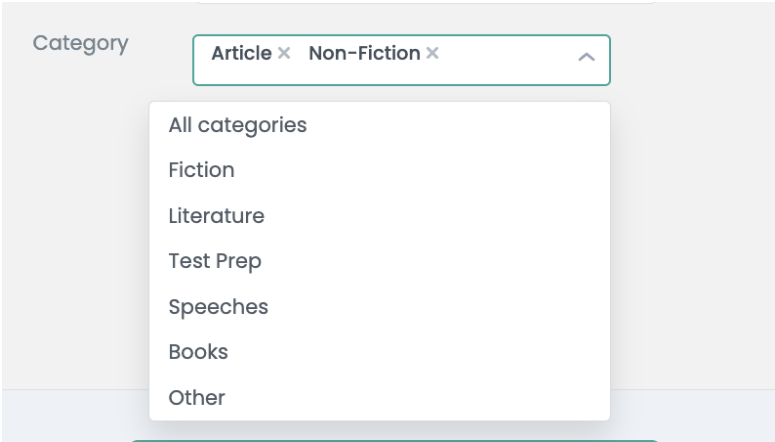


Рис. 3.25 Мультиселект з вибором категорії для списку

Далі нас одразу кидає на сторінку створення першого розділу для цього списку. На цій сторінці в правому верхньому куті в нас знаходиться інформація про спільну обкладинку, в верхньому лівому Назва і Опис даного розділу і нижче пошук і вибір слів для додавання в розділ (рис. 3.26). Також є можливість писати до кожного слова свої кастомні нотатки, які потім будуть

відображатися при вивченні слова (рис. 3.27). Є можливість одразу керувати розділами (додавати ще нові або видаляти)

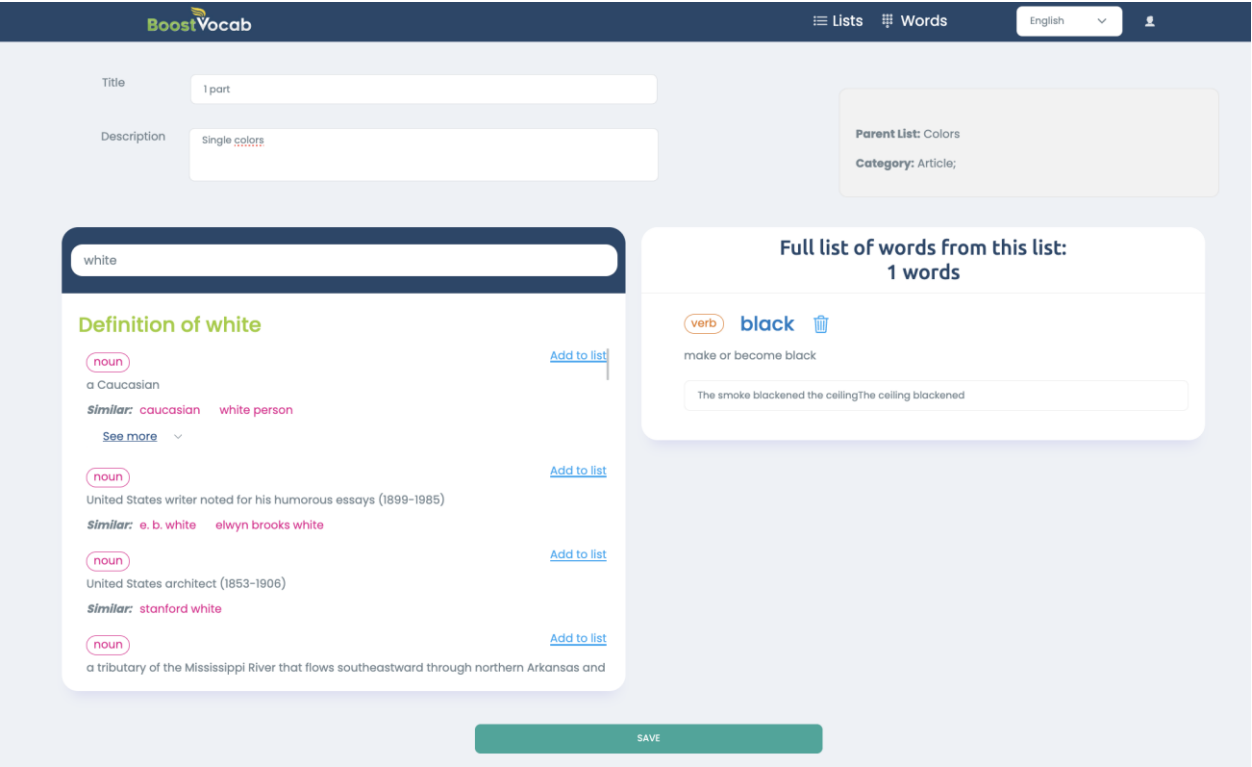


Рис. 3.26 Сторінка створення розділу

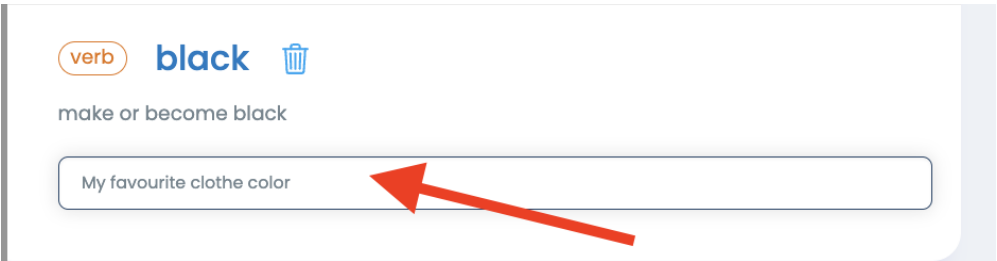


Рис. 3.27 Нотаток до слова

Процес створення звичайного списку досить схожий, там ми вибираємо приватність, назву, опис, категорію, картинку (тільки для адміністратора) і переходимо на сторінку додавання слів (рис. 3.28), (рис. 3.29).

BoostVocab Lists Words English

All Lists My Lists Favourite Lists Statistic Search vocabulary lists + New List

Create a new list

Private ☐

Title: Nature

Description: All natures words in the world

Category: Other x

BV

Next

Рис. 3.28 Сторінка створення звичайного списку

BoostVocab Lists Words English

All Lists My Lists Favourite Lists Statistic Search vocabulary lists + New List

Nature

All natures words in the world

Parent List: Category: Other

sun

Definition of sun

noun

first day of the week; observed as a day of rest and worship by most Christians

Similar: dominicus lord's day sunday

[See more](#)

Full list of words from this list: 1 words

tree

stretch (a shoe) on a shoetree

Рис. 3.29 Додавання слів до звичайного списку

Перейдемо до сторінки Улюблених списків, сторінка являю собою список, де ми можемо видалити з улюблених конкретний список або перейти до його вивчення (рис. 3.30)

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				55
Змн.	Арк.	№ докум.	Підпис	Дата		

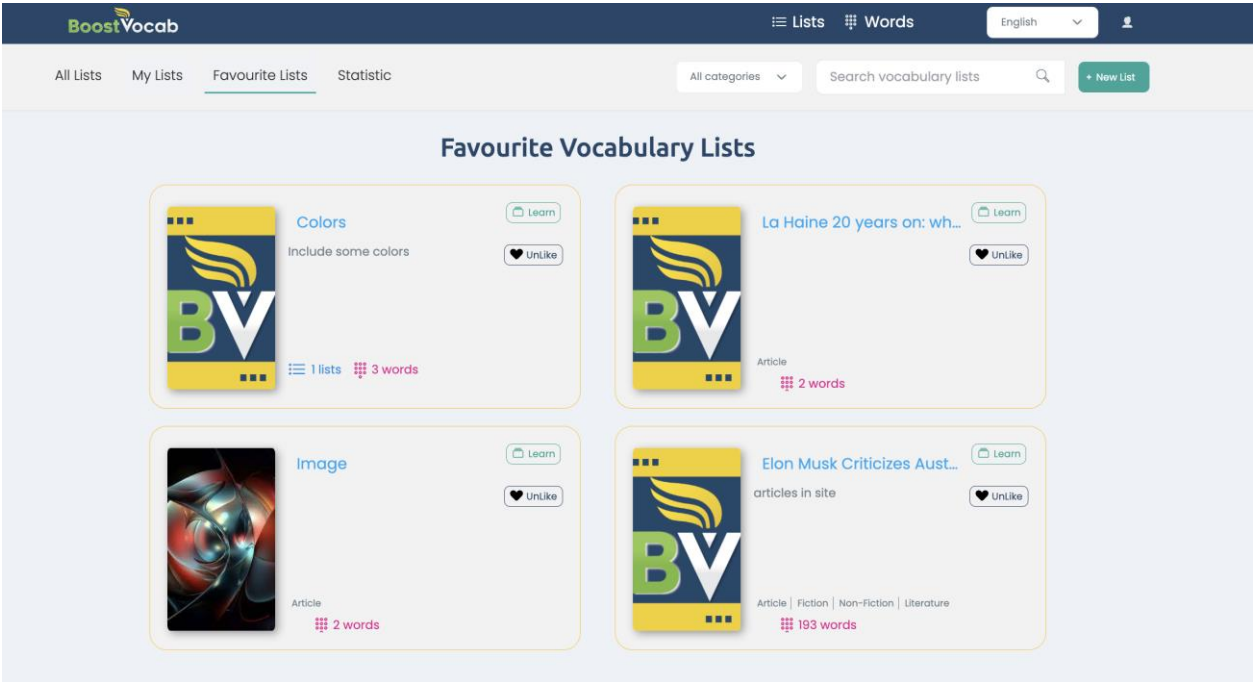


Рис. 3.30 Сторінка улюблених списків

Розглянемо сторінку статистики, вона представлена у вигляді фільтру по періоду (поточний тиждень, поточний місяць, поточний рік), суму кількості переглянутих списків і листів за вибраний період і 2 графіки з наглядною статистикою по даті. (рис. 3.31)



Рис. 3.31 Сторінка статистики

Сторінка Вивчення списку являє собою описом списку в правому верхньому куті (назва, картинка, опис, категорія і інформація про кількість розділів, слів списку), в центральній частині знаходиться карусель з флеш-картками розділу/списку, з вибором розділу над каруселлю (якщо є), і в нижній частині загальну структуру списку по розділам, які можуть бути представлені в 2 виглядах (тільки слова (рис. 3.33). або слова разом з описами (рис. 3.34)) (рис. 3.32).

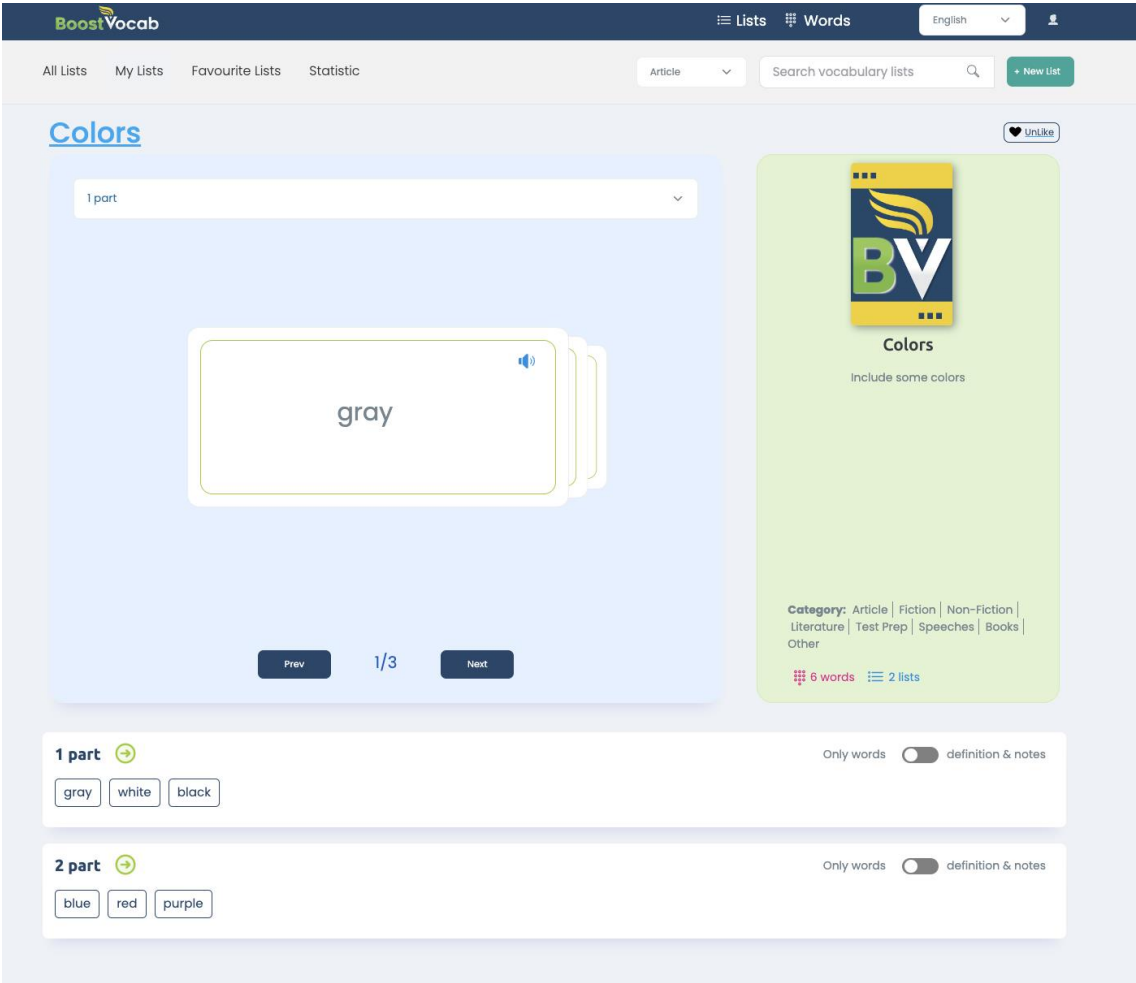


Рис. 3.32 Сторінка Вивчення списку

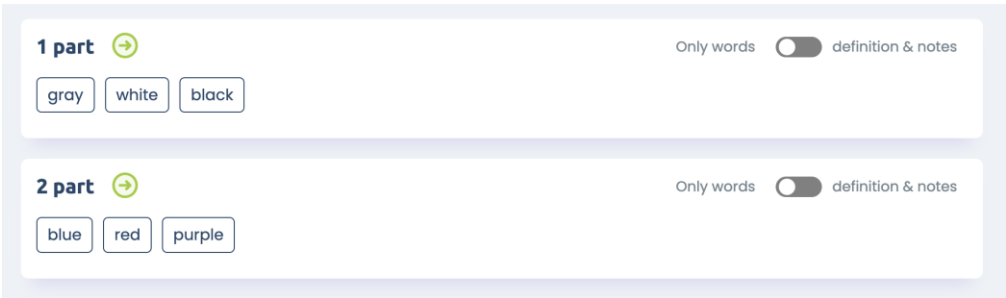


Рис. 3.33 Вигляд структури в режимі тільки слова

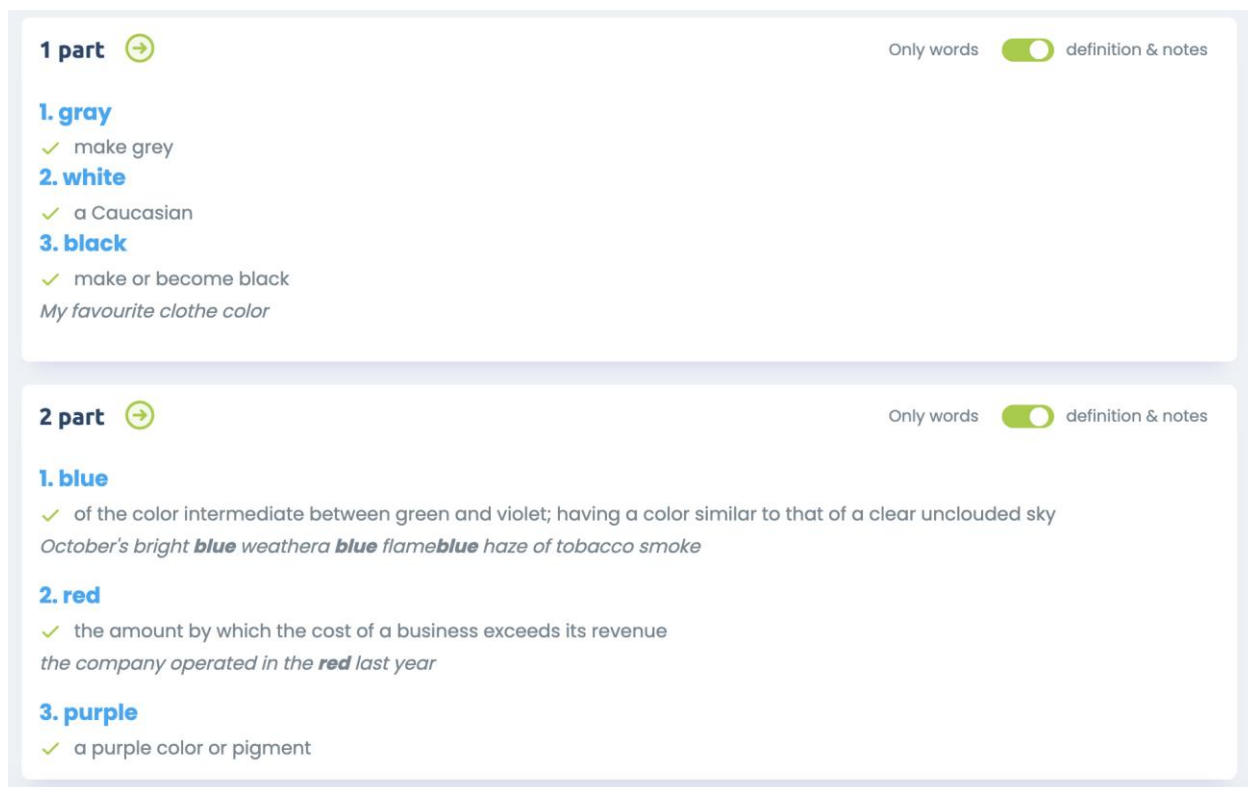


Рис. 3.34 Вигляд структури в режимі слова з описами

Розглянемо саму флеш-картку слова (рис. 3.35). В верхній стороні може бути написана нотатка, яку ми залишали при створенні списку до слова, і сама картка в якій є можливість натиснути на кнопку гучномовця і слово буде озвучене в аудіо. Коли ми натискаємо на картку, то вона перевертається і там є інформація про частину мови і опис слова (рис. 3.36).

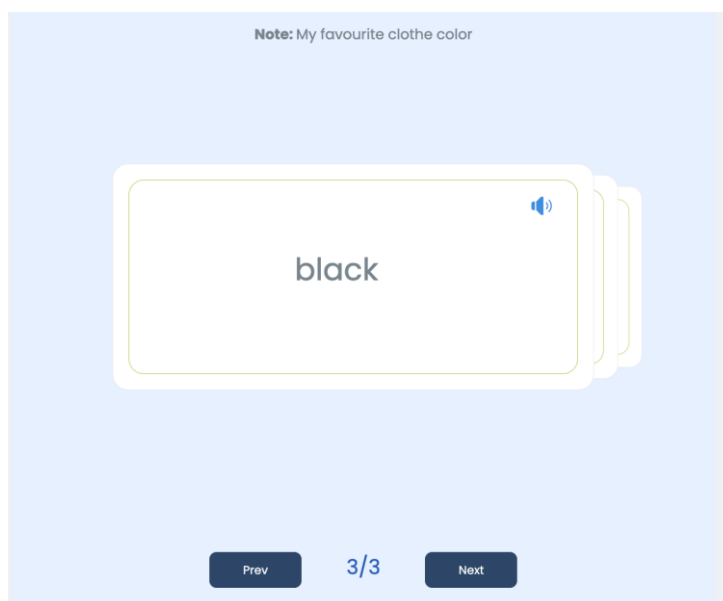


Рис. 3.35 Флеш картка

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				58
Змн.	Арк.	№ докум.	Підпис	Дата		

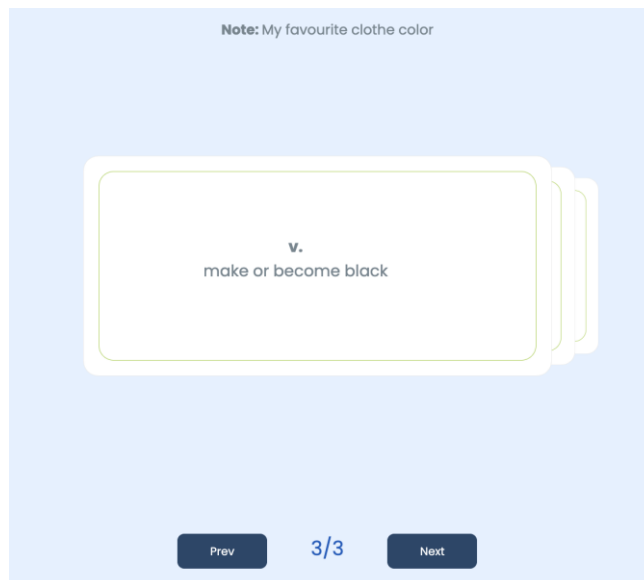


Рис. 3.36 Флеш картка в перевернутому вигляді

3.3 Тестування платформи

За допомогою онлайн сервісів було проведено тестування веб-додатку. Для цього використовувалися інструменти GTMetrix та Pagespeed. GTMetrix є інструментом для оцінки продуктивності та швидкості завантаження веб сторінок. Він надає звіти та аналіз продуктивності вашого веб-сайту, враховуючи різні метрики, такі як час завантаження сторінки, розмір сторінки, кількість запитів до сервера та багато іншого. Після завершення тестування був отриманий звіт, що містить різноманітну інформацію про продуктивність сайту (рис. 3.37).

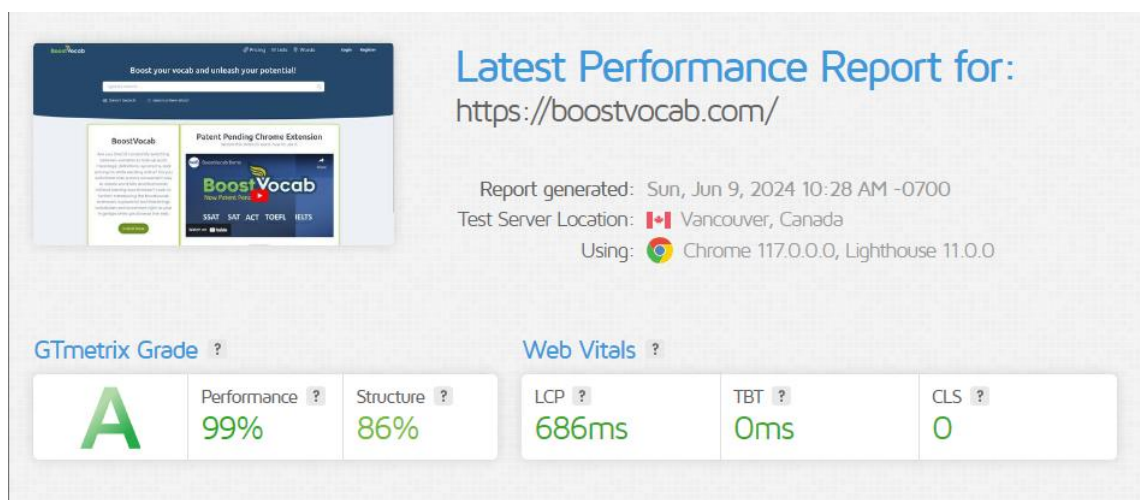


Рис. 3.37 Флеш картка в перевернутому вигляді

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				59
Змн.	Арк.	№ докум.	Підпис	Дата		

Звіти GTMetrix включають такі дані, як загальний час завантаження сторінки, час завантаження контенту, кількість запитів до сервера, розмір сторінки, швидкість завантаження на різних типах з'єднань та інші метрики. Ця інформація допомагає вам оцінити продуктивність та швидкість вашого веб-сайту, що дозволяє вам вжити заходів для покращення цих показників (рис. 3.38).

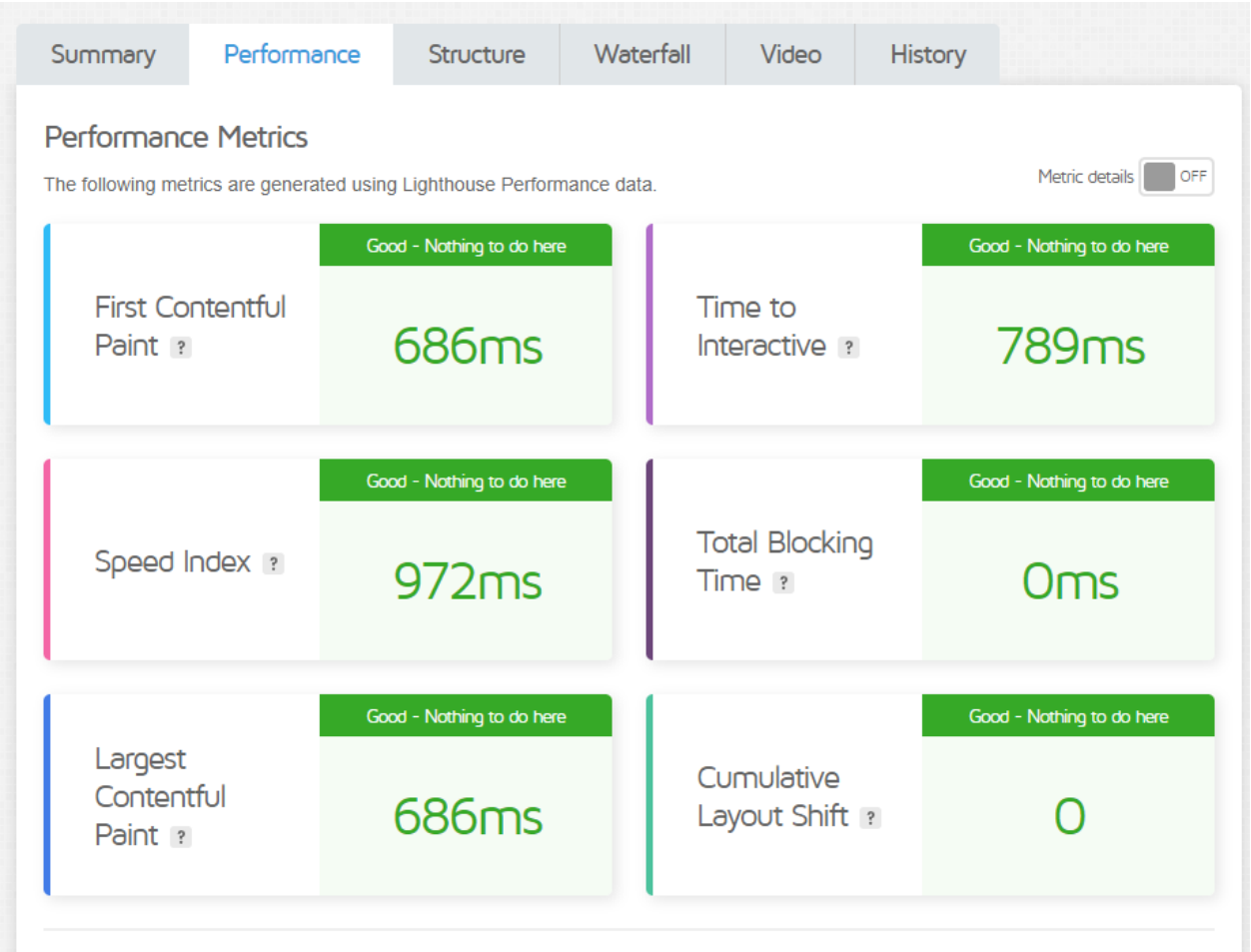


Рис. 3.38 Тестування швидкості веб додатку за допомогою GTMetrix

GTMetrix також пропонує ряд інструментів та функцій для оптимізації продуктивності веб-сайту, включаючи можливість зберігати та порівнювати звіти, відслідковувати продуктивність, отримувати сповіщення про проблеми з продуктивністю та багато іншого.

Висновки до третього розділу

У цьому розділі було представлено детальну схему розгортання платформи для вивчення англійських слів. Описано кожен структурний елемент системи, а також продемонстровано його взаємодію з іншими частинами платформи. Це дозволило зрозуміти архітектуру системи та взаємозв'язки між її компонентами.

Далі було надано опис інтерфейсу розробленого програмного продукту. Розглянуто всі функціональні можливості додатку залежно від ролі користувача.

Окрім того, було продемонстровано, як працює система статистики. Показано різницю між ними та продемонстровано, які дані відображають різні діаграми та статистичні показники.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було спроектовано та реалізовано платформу для вивчення англійських слів, проведено аналіз предметної області та порівняно різні платформи, що використовуються для вивчення мов. На основі цього аналізу були визначені основні вимоги до платформи для вивчення англійських слів. В процесі дослідження була розроблена архітектура, підходи та основні технології реалізації веб додатку.

Метою роботи було відтворення методології Пола Нейшна, яка є однією з найефективніших у сфері навчання мовам, з використанням сучасних веб-технологій. Проектування платформи включало аналіз варіантів використання програмного продукту та проектування структури, включаючи моделювання взаємодії між складовими частинами. Також було спроектовано базу даних для зберігання інформації про користувачів, слова, списки слів та статистику.

Описано алгоритм роботи з додатком та послідовні пункти варіантів його використання. Розглянуто основний функціонал, включаючи реєстрацію та авторизацію користувачів, перегляд та управління словами і списками слів, а також відстеження прогресу користувачів через статистику.

Також була надана детальна схема розгортання платформи для вивчення англійських слів, описано взаємодію структурних елементів системи та наведено опис інтерфейсу розробленого програмного продукту для різних типів користувачів. Показано роботу системи статистики для користувачів, що надає інформацію про активність користувачів.

Архітектура платформи побудована на базі сучасних веб-технологій, що забезпечує інтуїтивний інтерфейс та високу продуктивність. Інтерактивні методи, такі як списки флеш-карток, забезпечують ефективне запам'ятовування нових слів. Гейміфікація, включаючи статистику по пройденим словам, стимулює користувачів до регулярного навчання та відстеження прогресу.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				62
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток має зручний інтерфейс, який дозволяє користувачам легко організовувати свій навчальний процес, встановлювати цілі вивчення слів та відслідковувати свій прогрес. Користувачі можуть взаємодіяти зі списками слів, додавати нові слова, переглядати детальну інформацію про слова та отримувати зворотний зв'язок через коментарі та оцінки.

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				63
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. English as a global language [Електронний ресурс] – Режим доступу до ресурсу: <https://www.britannica.com/topic/English-language/Global-importance>
2. Innovations in language learning technologies [Електронний ресурс] – Режим доступу до ресурсу: <https://arbook.info/tehnologiyi-yaki-pokrashhuyut-osvitnij-dosvid>
3. Interactive learning methods and gamification [Електронний ресурс] – Режим доступу до ресурсу: <http://blog.ed-era.com/gamification-languages>
4. Creating custom flashcard lists [Електронний ресурс] – Режим доступу до ресурсу: <https://www.quizlet.com>
5. Gamification in education [Електронний ресурс] – Режим доступу до ресурсу: <https://www.elucidat.com/blog/gamification-in-elearning-examples/>
6. Smart search capabilities [Електронний ресурс] – Режим доступу до ресурсу: <https://searchengineland.com/what-is-intelligent-search-and-how-does-it-benefit-knowledge-management-345781>
7. Administration module functionalities [Електронний ресурс] – Режим доступу до ресурсу: <https://www.drupal.org/docs/7/administering-drupal-7-site/>
8. Effective web design principles [Електронний ресурс] – Режим доступу до ресурсу: <https://cxl.com/blog/universal-web-design-principles/>
9. System integration [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/cloud/learn/system-integration>
10. Database management with MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mysql.com/why-mysql/>
11. Benefits of interactive learning [Електронний ресурс] – Режим доступу до ресурсу: <https://mussila.com/the-top-6-benefits-of-interactive-learning/>

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				64
Змн.	Арк.	№ докум.	Підпис	Дата		

12. Competitive analysis of language learning apps [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@kellensho/a-competitive-analysis-of-language-learning-apps-23a69234fd15>
13. Duolingo [Електронний ресурс] – Режим доступу до ресурсу: <https://www.duolingo.com/>
14. Popularity of Duolingo [Електронний ресурс] – Режим доступу до ресурсу: <https://www.statista.com/statistics/1268052/duolingo-users-worldwide/>
15. Interactive and gamified learning with Duolingo [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.duolingo.com/gamification-design/>
16. Memrise [Електронний ресурс] – Режим доступу до ресурсу: <https://www.memrise.com/>
17. Memrise's learning approach [Електронний ресурс] – Режим доступу до ресурсу: <https://www.educationalappstore.com/app/memrise>
18. Babbel [Електронний ресурс] – Режим доступу до ресурсу: <https://www.babbel.com/>
19. Structured lessons in Babbel [Електронний ресурс] – Режим доступу до ресурсу: <https://preply.com/en/blog/duolingo-vs-babbel/>
20. Comparing language learning platforms [Електронний ресурс] – Режим доступу до ресурсу: <https://www.pcmag.com/picks/the-best-language-learning-software>
21. Key factors in language learning apps [Електронний ресурс] – Режим доступу до ресурсу: <https://triare.net/insights/language-learning-apps/>
22. Williams, R. Modern Web Development with Laravel. Journal of Web Engineering, 2020. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.journalofwebengineering.com/modern-laravel>
23. Thompson, G. MySQL for Educational Data Management. Journal of Database Systems, 2021. [Електронний ресурс] – Режим доступу до ресурсу:

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				65
Змн.	Арк.	№ докум.	Підпис	Дата		

https://www.researchgate.net/publication/338653026_Modeling_and_Management_Big_Data_in_Databases-A_Systematic_Literature_Review

24. Lee, K. Multilingual Support in Learning Applications. Language Learning Technology, 2021. [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/378285627_Technology-Assisted_Language_Learning_Systems_A_Systematic_Literature_Review
25. Brown, L. Interactive Learning Platforms. Language Technology Review, 2019. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.lltjournal.org/>
26. Green, L. Activity Diagrams in Software Engineering. Journal of Software Design, 2021. [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/351041535_A_UML_20_Activity_DiagramsCSP_Integrated_Approach_for_Modeling_and_Verification_of_Software_Systems
27. Campbell, P. User Experience and Response Time in Learning Applications. Journal of Digital Education, 2020. [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/353271483_Exploring_user_experience_of_learning_management_system
28. Nelson, R., & Baker, F. Paul Nation's Methodology for Vocabulary Learning. International Journal of Language Education, 2021. [Електронний ресурс] – Режим доступу до ресурсу: https://www.researchgate.net/publication/304042102_Language_Education_-_Vocabulary
29. Black, J. Laravel for Building Interactive Web Applications. Journal of Web Development, 2021. [Електронний ресурс] – Режим доступу до ресурсу: <https://admiral-studios.com/reasons-to-use-laravel-for-web-development>
30. MySQL Documentation. Oracle Corporation, 2023. [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc>

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				66
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				67
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Contracts\Support\Renderable
     */
    public function index()
    {
        return view('home');
    }

    public function terms()
    {
        return view('terms');
    }

    public function privacy()
    {
        return view('privacy');
    }

    public function changeLang(Request $request)
    {
        $language = $request->input('language', 'en');
        if (in_array($language, ['en', 'uk'])) {
            session(['applocale' => $language]);
        }
        return back();
    }
}

```

```

<?php

namespace App\Http\Controllers;

use Carbon\Carbon;
use Illuminate\Http\Request;

```

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				68
Змн.	Арк.	№ докум.	Підпис	Дата		

```

use App\Models\Lists;
use App\Models\Word;
use App\Models\FavouriteList;
use App\Models\StatisticUserWord;
use App\Models\Category;
use App\Models\UserListsWords;
use App\Models\Definition;

class ListController extends Controller
{
    private $word;
    private $favouriteList;
    private $statistic;
    private $definition;
    private $list;
    private $userList;
    private $category;

    private $responseArr = [];

    public function __construct()
    {
        $this->responseArr = ['error' => true, 'message' => 'Error', 'data' => []];
        $this->word = new Word();
        $this->favouriteList = new FavouriteList();
        $this->statistic = new StatisticUserWord();
        $this->definition = new Definition();
        $this->list = new Lists();
        $this->userList = new UserListsWords();
        $this->category = new Category();
    }

    public function addList()
    {
        return view("addList", ["categories" => $this->category->getAllCategory()]);
    }

    public function addNormalList()
    {
        return view("addNormalList", ["categories" => $this->category->getAllCategory()]);
    }

    public function addChapter($id)
    {
        return view("create_chapter", ["list" => $this->list->getList($id)]);
    }

    public function validationList(Request $request)
    {
        $request->listField = json_decode($request->listField);
        $isbn = isset($request->isbn) ? $request->isbn : null;
    }

```

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				69
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$request->words = $request->words != "undefined" ? json_decode($request->words) : null;
$path = $request->image != "undefined" && isset($request->image) ? $request->image-
>store('imageLists', 'public') : null;
$parentlistid = $request->parentlistid ?? null;
if ($parentlistid) {
    $tmp = $this->list->getChildLists($parentlistid);
    if (count($tmp))
        $order = $tmp[count($tmp) - 1]->order + 1;
    else
        $order = 1;
} else
    $order = null;
$list = $this->list->createNormalList($request->listField, $path, $request->selectedCategory,
$order, $parentlistid, $isbn);
if ($request->words)
    foreach ($request->words as $word) {
        $this->userList->createWord(auth()->user()->id, $list->id, $this->word-
>getWord($word->word)->wordid, $word->word, $word->definition, $word->partofspeech,
$path, $word->usernotes == "" ? null : $word->usernotes);
    }
    return $list;
}

public function validationNormalList(Request $request)
{
    $isbn = isset($request->isbn) ? $request->isbn : null;
    $request->listField = json_decode($request->listField);
    $request->words = $request->words != "undefined" ? json_decode($request->words) : null;
    $path = $request->image != "undefined" && isset($request->image) ? $request->image-
>store('imageLists', 'public') : null;
    $list = $this->list->createNormalList($request->listField, $path, $request->selectedCategory,
null, null, $isbn);
    if ($request->words)
        foreach ($request->words as $word) {
            $this->userList->createWord(auth()->user()->id, $list->id, $this->word-
>getWord($word->word)->wordid, $word->word, $word->definition, $word->partofspeech,
$path, $word->usernotes == "" ? null : $word->usernotes);
        }
    return $list;
}

public function myLists()
{
    if (isset($_GET["category"])) {
        $id = $this->category->getCategoryIdByName($_GET["category"]);
        $lists = $this->list->getAllMyListsByCategory(intval($id), $_GET["query"] ?? null);
    } else
        $lists = $this->list->getAllMyLists($_GET["query"] ?? null);
    $counts = [];
    $countsLists = [];
    foreach ($lists as $list) {

```

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$tmp = 0;
$childs = $this->list->getChildLists($list->id);
if (count($childs)) {
    array_push($countsLists, count($childs));
    foreach ($childs as $child) {
        $tmp += count($this->userList->getListWords($child->id));
    }
} else {
    array_push($countsLists, 0);
    if ($this->userList->getListWords($list->id)) {
        $tmp = count($this->userList->getListWords($list->id));
    }
}
array_push($counts, $tmp);
}
return view('myLists', ["lists" => $lists, "counts" => $counts, "countsLists" => $countsLists,
"categories" => $this->category->getAllCategory()]);
}

```

```

public function allLists()
{
    if (isset($_GET["query"]) && !isset($_GET["category"])) {
        $lists = $this->list->getAllListsByQuery($_GET["query"]);
        $featuredLists = $this->list->getFeaturedLists(null, $_GET["query"]);
        $premiumLists = $this->list->getPremiumLists(null, $_GET["query"]);
    } else if (!isset($_GET["query"]) && isset($_GET["category"])) {
        $id = $this->category->getCategoryIdByName($_GET["category"]);
        $lists = $this->list->getAllListsByCategory(intval($id));
        $featuredLists = $this->list->getFeaturedLists(intval($id), null);
        $premiumLists = $this->list->getPremiumLists(intval($id), null);
    } else if (isset($_GET["query"]) && isset($_GET["category"])) {
        $id = $this->category->getCategoryIdByName($_GET["category"]);
        $lists = $this->list->getAllListsByCategoryQuery($_GET["query"], intval($id));
        $featuredLists = $this->list->getFeaturedLists(intval($id), $_GET["query"]);
        $premiumLists = $this->list->getPremiumLists(intval($id), $_GET["query"]);
    } else if (!isset($_GET["query"]) && !isset($_GET["category"])) {
        $lists = $this->list->getAllLists();
        $featuredLists = $this->list->getFeaturedLists(null, null);
        $premiumLists = $this->list->getPremiumLists(null, null);
    }
    $paginateFeatured = $featuredLists["paginate"];
    $featuredLists = $featuredLists["arr"];
    $paginateNormal = $lists["paginate"];
    $lists = $lists["arr"];

    $counts = [];
    $countsLists = [];
    $countsPremiumLists = [];
    $countsPremium = [];
    $countsFeaturedLists = [];
    $countsFeatured = [];
}

```

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				71
Змн.	Арк.	№ докум.	Підпис	Дата		

```

foreach ($featuredLists as $featuredListList) {
    $tmp = 0;
    $childs = $this->list->getChildLists($featuredListList->id);
    if (count($childs)) {
        array_push($countsFeaturedLists, count($childs));
        foreach ($childs as $child) {
            $tmp += count($this->userList->getListWords($child->id));
        }
    } else {
        array_push($countsFeaturedLists, 0);
        if ($this->userList->getListWords($featuredListList->id)) {
            $tmp = count($this->userList->getListWords($featuredListList->id));
        }
    }
    array_push($countsFeatured, $tmp);
}
foreach ($premiumLists as $premiumList) {
    $tmp = 0;
    $childs = $this->list->getChildLists($premiumList->id);
    if (count($childs)) {
        array_push($countsPremiumLists, count($childs));
        foreach ($childs as $child) {
            $tmp += count($this->userList->getListWords($child->id));
        }
    } else {
        array_push($countsPremiumLists, 0);
        if ($this->userList->getListWords($premiumList->id)) {
            $tmp = count($this->userList->getListWords($premiumList->id));
        }
    }
    array_push($countsPremium, $tmp);
}
foreach ($lists as $list) {
    $tmp = 0;
    $childs = $this->list->getChildLists($list->id);
    if (count($childs)) {
        array_push($countsLists, count($childs));
        foreach ($childs as $child) {
            $tmp += count($this->userList->getListWords($child->id));
        }
    } else {
        array_push($countsLists, 0);
        if ($this->userList->getListWords($list->id)) {
            $tmp = count($this->userList->getListWords($list->id));
        }
    }
    array_push($counts, $tmp);
}
return view('allLists', ["lists" => $lists, "counts" => $counts, "countsLists" => $countsLists,
"premiumLists"      => $premiumLists, "countsPremium"      => $countsPremium,
"countsPremiumLists" => $countsPremiumLists, "featuredLists" => $featuredLists,

```

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				72
Змн.	Арк.	№ докум.	Підпис	Дата		

```

"countsFeatured" => $countsFeatured, "countsFeaturedLists" => $countsFeaturedLists,
"categories" => $this->category->getAllCategory(), "paginateFeatured" => $paginateFeatured,
"paginateNormal" => $paginateNormal, 'favouriteLists' => FavouriteList::where('user_id', auth()-
>id())->get()->pluck('list_id')->toArray());
}

public function editListForm($id)
{
    if (auth()->user()->id != $this->list->getList($id)->listownerid)
        return abort(403);
    return view("editList", ["list" => $this->list->getList($id), "childs" => $this->list-
>getChildLists($id), "categories" => $this->category->getAllCategory()]);
}

public function deleteNormalList($id)
{
    $this->list->deleteChapterList($id);
    $this->userList->deleteWordsByList($id);
}

public function editNormalListForm($id)
{
    if (auth()->user()->id != $this->list->getList($id)->listownerid)
        return abort(403);
    return view("editNormalList", ["list" => $this->list->getList($id), "words" => $this-
>userList->getListWords($id), "categories" => $this->category->getAllCategory()]);
}

public function editList(Request $request, $id)
{
    $request->listField = json_decode($request->listField);
    $isbn = isset($request->isbn) ? $request->isbn : null;
    $request->orderList = explode(",", $request->orderList);
    $list = $this->list->getList($id);
    $path = $request->image != "undefined" && isset($request->image) && $request->image ?
$request->image->store('imageLists', 'public') : $list->image;
    $this->list->editNormalList($id, $request->listField, $list, $path, $request-
>selectedCategory, $isbn);
    if ($request->orderList) {
        $tmp = 0;
        foreach ($request->orderList as $order) {
            $tmp++;
            $this->list->updateOrder($order, $tmp);
        }
    }
}

public function editTitleChapter(Request $request, $id)
{
    $this->list->updateListByColumn($id, "title", $request->title);
}

```

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				73
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public function editDescChapter(Request $request, $id)
{
    $this->list->updateListByColumn($id, "description", $request->description);
}

public function deleteWord(Request $request, $id)
{
    $this->userList->deleteWord($id, $request->word, $request->partofspeech, $request->definition);
}

public function addWord(Request $request, $id)
{
    $this->userList->createWord(auth()->user()->id, $id, $this->word->getWord($request->word)->wordid, $request->word, $request->definition, $request->partofspeech);
}

public function editUserNotes(Request $request, $id)
{
    $this->userList->changeUserNotes($id, $request->word, $request->partofspeech, $request->definition, $request->usernotes);
}

public function editNormalList(Request $request, $id)
{
    $isbn = isset($request->isbn) ? $request->isbn : null;
    $request->listField = json_decode($request->listField);
    $request->words = json_decode($request->words);
    $list = $this->list->getList($id);
    $path = $request->image != "undefined" && isset($request->image) && $request->image ? $request->image->store('imageLists', 'public') : $list->image;
    $this->list->editNormalList($id, $request->listField, $list, $path, $request->selectedCategory, $isbn);
    if ($request->words)
        $this->userList->deleteWordsByList($id);
    foreach ($request->words as $word) {
        $this->userList->createWord(auth()->user()->id, $id, $this->word->getWord($word->word)->wordid, $word->word, $word->definition, $word->partofspeech, $path, $word->usernotes == "" ? null : $word->usernotes);
    }
}

public function addWordToList(Request $request): string
{
    $list = $this->list->getList($request->listid);
    $word = $this->definition->getWordByDefinition($request->word, $request->definition);
    if ($this->userList->checkDuplicateWordByList($word->word, $request->definition, $list->id)) {
        return 'Error';
    }
}

```

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				74
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return $this->userList->createWord($list->listownerid, $list->id, $word->wordid, $word->word, $request->definition, $request->partofspeech, $list->image);
    }

    public function addWordsToList(Request $request)
    {
        if (isset($request->listid)) {
            if (isset($request->words)) {
                $words = $request->words;
                $list = $this->list->getList($request->listid);
                if (count($words) > 0) {
                    for ($i = 0; $i < count($words); $i++) {
                        $word = $this->definition->getWordByDefinition($words[$i]['word'], $words[$i]['definition']);
                        if (!$this->userList->checkDuplicateWordByList($word->word, $words[$i]['definition'], $list->id)) {
                            $this->userList->createWord($list->listownerid, $list->id, $word->wordid, $word->word, $words[$i]['definition'], $words[$i]['partOfSpeech'], $list->image);
                        }
                    }
                    $this->responseArr = ['error' => false, 'message' => 'Done'];
                }
            }
        }
        return response()->json($this->responseArr);
    }

    public function addNewListAndWordsToList(Request $request)
    {
        if (isset($request->description) && isset($request->title) && isset($request->words)) {
            $categories = $this->category->getAllCategory();
            $type = isset($request->type) ? $request->type : false;
            $domain = isset($request->domain) ? $request->domain : "";
            $href = isset($request->href) ? $request->href : "";
            if (count($categories) > 0) {
                $insertCategories = '[';
                $flag = false;
                foreach ($categories as $category) {
                    if (!$flag) {
                        $flag = true;
                    } else {
                        $insertCategories .= ',';
                    }
                    $insertCategories .= $category->id;
                }
                $insertCategories .= "];";
                $list = $this->list->createList($request->title, $request->description, $insertCategories, $type, $domain, $href);
                $words = $request->words;
                if (count($words) > 0) {
                    for ($i = 0; $i < count($words); $i++) {

```

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				75
Змн.	Арк.	№ докум.	Підпис	Дата		

```

                $word = $this->definition->getWordByDefinition($words[$i]['word'],
$words[$i]['definition']);
                if (!($this->userList->checkDuplicateWordByList($word->word,
$words[$i]['definition'], $list->id)) {
                    $this->userList->createWord($list->listownerid, $list->id, $word->wordid,
$word->word, $words[$i]['definition'], $words[$i]['partOfSpeech'], $list->image);
                }
            }
            $this->responseArr = ['error' => false, 'message' => 'Done'];
        }
    }
    return response()->json($this->responseArr);
}

```

```

public function deleteList($id):
\Illuminate\Routing\Redirector|\Illuminate\Contracts\Foundation\Application|\Illuminate\Http\RedirectResponse
{
    foreach ($this->list->getChildLists($id) as $chapter) {
        $this->userList->deleteWordsByList($chapter->id);
    }
    $this->list->deleteList($id);
    return redirect(route("myLists"));
}

```

```

public function deleteChapterList($id)
{
    $this->userList->deleteWordsByList($id);
    $this->list->deleteChapterList($id);
}

```

```

public function learn($name, $id):
\Illuminate\Contracts\View\Factory|\Illuminate\Contracts\View\View|\Illuminate\Contracts\Foundation\Application
{
    $childs = $this->list->getChildLists($id);
    $this->statistic->addList($id);
    $this->userList->getWordsByChapters($childs, true);
    return view("learn", ["list" => $this->list->getList($id), "childs" => $childs, "words" =>
$this->userList->getWordsByChapters($childs, true), "categories" => $this->category->
getAllCategory(), 'favouriteLists' => FavouriteList::where('user_id', auth()->id()->get()->
pluck('list_id')->toArray());
}

```

```

public function learnNormal($id):
\Illuminate\Contracts\View\Factory|\Illuminate\Contracts\View\View|\Illuminate\Contracts\Foundation\Application
{
    $this->statistic->addList($id);
}

```

```

        return view("learn", ["list" => $this->list->getList($id), "words" => $this->userList->getListWords($id, true), "categories" => $this->category->getAllCategory(), 'favouriteLists' => FavouriteList::where('user_id', auth()->id()->get()->pluck('list_id')->toArray());
    }

    public function getAuth(): ?\Illuminate\Contracts\Auth\Authenticatable
    {
        return auth()->user();
    }

    public function addFavouriteList(Request $request)
    {
        $listId = $request->input('list_id');
        $this->favouriteList->addList($listId);
        return response()->json(['message' => 'ok']);
    }

    public function deleteFavouriteList(Request $request)
    {
        $listId = $request->input('list_id');
        $this->favouriteList->deleteList($listId);
        return response()->json(['message' => 'ok']);
    }

    public function favouriteLists(Request $request)
    {
        $lists = $this->favouriteList->getFavouriteLists($_GET["category"] ?? null, $_GET["query"] ?? null, $request->input('page'));
        return view("favouriteLists", ["lists" => $lists, 'query' => $_GET["query"] ?? null, 'category' => $_GET["category"] ?? null, "categories" => $this->category->getAllCategory());
    }

    public function statistic(Request $request)
    {
        $interval = $request->input('interval', "week");
        $data = $this->statistic->getData($interval);
        $datasetsWords = [];
        $datasetsLists = [];
        $labels = [];
        switch ($interval) {
            case 'week':
                $dates = [];
                for ($i = 0; $i < 7; $i++) {
                    $day = Carbon::now()->startOfWeek()->addDays($i);
                    $labels[] = $day->isoFormat('dddd');
                    $dates[] = $day->toDateString();
                }
                foreach ($dates as $date) {
                    $countList = 0;
                    $countWord = 0;
                    foreach ($data->where('list_id', "!=", null) as $list) {

```

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				77
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if ($list->date == $date) {
            $countList++;
            if ($list->subLists && count($list->subLists)) {
                $countWord += $list->list->subLists->sum(fn($sublist) => $sublist->words-
>count());
            } else {
                $countWord += count($list->list->words);
            }
        }
    }
    foreach ($data->where('word_id', "!=" , null) as $word) {
        if ($word->date == $date) {
            $countWord++;
        }
    }
    $datasetsLists[] = $countList;
    $datasetsWords[] = $countWord;
}
break;
case 'month':
    $now = Carbon::now();
    $startOfMonth = $now->copy()->startOfMonth();
    $endOfMonth = $now->copy()->endOfMonth();
    $currentDate = $startOfMonth->copy();
    while ($currentDate->lte($endOfMonth)) {
        $labels[] = $currentDate->toDateString();
        $currentDate = $currentDate->addDay();
    }
    foreach ($labels as $label) {
        $countList = 0;
        $countWord = 0;
        foreach ($data->where('list_id', "!=" , null) as $list) {
            if ($list->date == $label) {
                $countList++;
                if ($list->list->subLists && count($list->list->subLists)) {
                    $countWord += $list->list->subLists->sum(fn($sublist) => $sublist->words-
>count());
                } else {
                    $countWord += count($list->list->words);
                }
            }
        }
        foreach ($data->where('word_id', "!=" , null) as $word) {
            if ($word->date == $label) {
                $countWord++;
            }
        }
        $datasetsLists[] = $countList;
        $datasetsWords[] = $countWord;
    }
}
break;

```

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				78
Змн.	Арк.	№ докум.	Підпис	Дата		

```

case 'year':
  for ($month = 1; $month <= 12; $month++) {
    $date = Carbon::create(null, $month);
    $labels[] = $date->isoFormat('MMMM');
  }
  for ($month = 1; $month <= 12; $month++) {
    $countList = 0;
    $countWord = 0;
    $startOfMonth = Carbon::createFromDate(null, $month, 1)->startOfMonth();
    $endOfMonth = Carbon::createFromDate(null, $month, 1)->endOfMonth();
    foreach ($data->where('list_id', "!=", null) as $list) {
      $recordDate = Carbon::parse($list->date);
      if ($recordDate->between($startOfMonth, $endOfMonth)) {
        $countList++;
        if ($list->list->subLists && count($list->list->subLists)) {
          $countWord += $list->list->subLists->sum(fn($sublist) => $sublist->words-
>count());
        } else {
          $countWord += count($list->list->words);
        }
      }
    }
    foreach ($data->where('word_id', "!=", null) as $word) {
      $recordDate = Carbon::parse($word->date);
      if ($recordDate->between($startOfMonth, $endOfMonth)) {
        $countWord++;
      }
    }
    $datasetsLists[] = $countList;
    $datasetsWords[] = $countWord;
  }
  break;
}
return view("statistic", ['interval' => $interval, 'labels' => $labels, 'datasetsLists' =>
$datasetsLists, 'datasetsWords' => $datasetsWords]);
}

```

		Мельник П.І.			ІПЗ.КР.Б-121-24-ПЗ	Арк.
		Власенко О.В.				79
Змн.	Арк.	№ докум.	Підпис	Дата		