

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інженерії програмного забезпечення**

ДОПУЩЕНО ДО ЗАХИСТУ  
Завідувач кафедри \_\_\_\_\_ Є. О. Давиденко  
*підпис*  
«\_\_» \_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**  
**ВЕБЗАСТОСУНОК ВИВЧЕННЯ ІНОЗЕМНИХ МОВ НА ОСНОВІ**  
**МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ**

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 409.22010918

***Здобувач освіти***

\_\_\_\_\_ М. О. Очковський  
*підпис*  
«\_\_» \_\_\_\_\_ 2024 р.

***Керівник*** д-р. техн. наук, проф.

\_\_\_\_\_ А. В. Швед  
*підпис*  
«\_\_» \_\_\_\_\_ 2024 р.

**Консультант** канд. техн. наук, доцент

\_\_\_\_\_ А. О. Алексєєва  
*підпис*  
«\_\_» \_\_\_\_\_ 2024 р.

**Миколаїв 2024**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра**  
**Могили****Факультет комп'ютерних наук**  
**Кафедра інженерії програмного забезпечення**

ЗАТВЕРДЖУЮ  
Зав. кафедри  
\_\_\_\_\_ Є. О. Давиденко  
«\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**на виконання кваліфікаційної роботи бакалавра**

**Видано здобувачу освіти групи 409 факультету комп'ютерних наук**  
**Очковському Миколі Олеговичу**

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

1. Тема кваліфікаційної роботи:

Вебзастосунок вивчення іноземних мов на основі методів штучного інтелекту

Затверджена наказом по ЧНУ ім. П.Могили від « 22 » грудня 2023 р. № 269

2. Строк представлення кваліфікаційної роботи: «\_» \_\_ 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом є вебзастосунок вивчення іноземних мов

4. Перелік питань, що підлягають розробці:

- дослідження предметної області;
- аналіз існуючих аналогів та виявлення проблем;
- формування вимог до програмного забезпечення;
- проектування програмного забезпечення;
- розробка програмного забезпечення;
- тестування роботи програмного забезпечення;
- проведення аналізу результатів розробки.

5. Перелік графічних матеріалів:

Презентація.

6. Завдання до спеціальної частини

Аналіз охорони праці на робочих місцях фахівців з інформаційних технологій.

---

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алексєєва А.О.	Кафедра екології	Спеціальна частина зохорони праці

Керівник роботи д-р.техн.наук, проф. Швед А. В.  
(посада, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Завдання прийнято до виконання

Очковський Микола Олегович  
(прізвище, ім'я, по батькові здобувача освіти)

\_\_\_\_\_  
(підпис)

Дата видачі завдання «\_\_» \_\_ 2024 р.

**КАЛЕНДАРНИЙ ПЛАН**  
**виконання кваліфікаційної роботи**

Тема кваліфікаційної роботи:

Вебзастосунок вивчення іноземних мов на основі методів штучного інтелекту

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	22.12.2023 р.	23.12.2023 р.	Виконано
2.	Огляд літератури за темою роботи	23.12.2023 р.	10.01.2024 р.	Виконано
3.	Складання календарного плану КРБ	01.02.2024 р.	10.02.2024 р.	Виконано
4.	Аналіз предметної області	11.02.2024 р.	15.02.2024 р.	Виконано
5.	Розробка проектних рішень	16.02.2024 р.	20.03.2024 р.	Виконано
6.	Моделювання та конструювання ПЗ	20.02.2024 р.	29.02.2024 р.	Виконано
7.	Кодування, тестування розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	01.04.2024 р.	30.05.2024 р.	Виконано
8.	Оформлення КРБ та презентації	01.08.2024 р.	04.06.2024 р.	Виконано
9.	Розробка спеціальної частини з охорони праці	01.05.2024 р.	15.05.2024 р.	Виконано
10.	Попередній захист	03.06.2024 р.	05.06.2024 р.	Виконано
11.	Відгук керівника КРБ	08.06.2024 р.	08.06.2024 р.	Виконано
12.	Завершення оформлення КРБ та презентації	10.06.2024 р.	17.06.2024 р.	Виконано
13.	Рецензування	10.06.2024 р.	17.06.2024 р.	Виконано
14.	Захист кваліфікаційної роботи	27.06.2024 р.	27.06.2024 р.	Виконано

Розробив здобувач освіти Очковський Микола Олегович

(прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

«\_\_» \_\_\_\_\_ 202\_ р.

Керівник роботи д-р.техн.наук, проф. Швед А. В.

(посада, прізвище, ім'я, по батькові)

(підпис)

«\_\_» \_\_\_\_\_ 202\_ р.

## АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Вебзастосунок вивчення іноземних мов на основі методів штучного інтелекту»

Здобувач освіти 409 гр.: Очковський Микола Олегович

Керівник: д-р. техн. наук, проф. Швед Альона Володимирівна

Кваліфікаційна робота бакалавра присвячена розробці веборієнтованого застосунку для вивчення іноземної мови як засобу підвищення ефективності засвоєння навчального матеріалу та мотивації до вивчення іноземної мови шляхом застосування методів штучного інтелекту.

**Об'єкт роботи:** процес формування іншомовної компетентності користувачів на основі сучасних інформаційних технологій.

**Предмет роботи:** технології та інструментальні засоби розробки вебзастосунків.

**Мета:** розробка веборієнтованого застосунку вивчення іноземної мови як засобу підвищення ефективності засвоєння навчального матеріалу та мотивації до вивчення іноземної мови за рахунок застосування методів штучного інтелекту.

Кваліфікаційна робота бакалавра складається з вступу, чотирьох розділів, висновків та переліку джерел посилання.

У вступі визначається актуальність теми, що приймається за мету та проводиться невеликий огляд поставленого завдання, предмету та об'єкту дослідження.

У першому розділі проведено аналіз застосунків-аналогів, представлено їх сильні та слабкі сторони, також їх основні функції. Проводиться аналіз програмного забезпечення, що розробляється та його характеристики. Наступною частиною формується та описується специфікація вимог до програмного забезпечення, що розробляється.

У другому розділі визначаються та модулюються процеси розробки програмного забезпечення, описуються функціональні можливості програмного

забезпечення. В результаті процесу розробки складено детальний алгоритм розв'язування поставленої задачі.

У третьому розділі проводиться огляд стеку технологій для клієнтської, серверної частини програмного забезпечення. Наведено UML-діаграми, які відіграють важливу роль у проєктуванні вебзастосунку.

У четвертому розділі наведено програмну реалізацію функціоналу вебзастосунку для вивчення іноземних мов. Описується проведена робота над базою даних, створення та налаштування серверних процесів та розробка користувацького інтерфейсу.

У висновках проводиться аналіз роботи та отриманих результатів.

КРБ викладена на 56 сторінки, вона містить 4 розділи, 36 ілюстрацій, 5 таблиць, 15 джерел в переліку посилань.

Ключові слова: *програмне забезпечення, машинне навчання, іноземна мова, вебзастосунок, вивчення, обробка природньої мови.*

## **ABSTRACT**

of the Bachelor's Thesis

«Web application for learning foreign languages based on artificial intelligence methods»

Student.: Ochkovskyi Mykola

Supervisor: Dr. Sc., Professor Shved A. V.

The bachelor's thesis is devoted to the development of a web-based application for learning a foreign language as a means of increasing the efficiency of learning and motivation to learn a foreign language by applying artificial intelligence methods.

**Object:** the process of forming users foreign language competence based on modern information technologies.

**Subject:** technologies and tools for developing web applications.

**Objective:** development of a web-based application for learning a foreign language as a means of increasing the efficiency of learning and motivation to learn a foreign language through the use of artificial intelligence methods.

The bachelor's thesis consists of an introduction, four chapters, conclusions and a list of references.

The introduction determines the relevance of the topic taken as the goal and a brief overview of the task, subject and object of research.

The first section analyzes analogous applications, presenting their strengths and weaknesses, as well as their main functions. The software under development and its characteristics are analyzed. The next part of the article describes the specification of requirements for the software to be developed.

The second section defines and modulates the software development processes and describes the software functionality. As a result of the development process, a detailed algorithm for solving the task is compiled. The conclusions analyze the work and the results obtained.

The third section provides an overview of the technology stack for the client and server parts of the software. UML diagrams are presented, which play an important role in the design of a web application.

The fourth section presents the software implementation of the web application functionality for learning foreign languages. It describes the work done on the database, the creation and configuration of server processes, and the development of the user interface.

The qualification work of the bachelor is presented on 56 pages, it contains 4 sections, 36 illustrations, 5 tables, 15 sources in the list of references.

Keywords: *software, machine learning, foreign language, web application, learning, natural language processing.*



## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Використання методів штучного інтелекту у вивченні іноземних мов.....	6
1.2 Огляд існуючих аналогів.....	7
1.3 Аналіз системи, що розробляється.....	14
1.4 Специфікація вимог.....	16
Висновки до розділу 1.....	21
2 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	22
2.1 Діаграма прецедентів.....	22
2.2 Діаграма діяльності.....	26
2.3 Діаграма розгортання.....	29
2.4 Діаграма взаємодії.....	31
Висновки до розділу 2.....	34
3 ОГЛЯД СТЕКУ ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ ВЕБЗАСТОСУНКУ.....	35
3.1 Використані мови програмування.....	35
3.2 Технології розробки клієнтської та серверної частини.....	36
3.3 Огляд словників англійської мови.....	39
3.4 Технології обробки природньої мови.....	40
Висновки до розділу 3.....	42
4 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	43
4.1 Діаграми класів.....	43
4.2 Формування єдиної бази даних слів.....	44
4.3 Реалізація алгоритму обробки природньої мови.....	47
4.4 Створення інтерфейсу для взаємодії з моделю.....	49
Висновки до розділу 4.....	53
ВИСНОВКИ.....	54
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	55

## ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ	–	програмне забезпечення
МП	–	мова програмування
ІМ	–	іноземна мова
БД	–	база даних
ШІ	–	штучний інтелект
NLP	–	natural language processing
UML	–	unified modeling language
API	–	application programming interface
WE	–	word embedding
FE	–	front-end
BE	–	back-end
CRUD	–	create read update delete
JSON	–	javascript object notation
NLTK	–	python and the natural language toolkit

## ВСТУП

**Актуальність теми** кваліфікаційної роботи бакалавра зумовлена тенденцією останніх десятиліть, а саме, швидкою глобалізацією світу та розширенням англійської мови, як домінантною у світі. Тільки за останні 5 років, кількість людей, які мають другу мову, як англійську збільшилося з 750 мільйонів до більше ніж 1.1 мільярда людей. Загалом кількість людей, які володіють англійською становить 1.45 мільярда людей з очікуванням, зростанням до 2 мільярди людей до 2050 року. З людей, які володіють англійською, як іноземною, можна виділити, лише малий відсоток людей, які можуть вільно розмовляти нею, в той час, коли більшість людей мають середній, нижче середнього та вище середнього рівня англійської [1].

В сучасному світі знання декількох мов є важливим фактором для особистого та професійного розвитку. Для сучасного існування в глобальній мережі, знання англійської мови є критично важливим, згідно з статистикою 58 відсотків всього контенту в інтернеті написано, саме англійською мовою. Якщо, проблематично знайти інформацію рідною мовою, то скоріш за все, ця інформація доступна на англійській, також більшість навчального матеріалу, випускається на англійській.

**Об'єкт роботи:** процес формування іншомовної компетентності користувачів на основі сучасних інформаційних технологій.

**Предмет роботи:** технології та інструментальні засоби розробки вебзастосунків.

**Мета:** розробка веборієнтованого застосунку вивчення іноземної мови як засобу підвищення ефективності засвоєння навчального матеріалу та мотивації до вивчення іноземної мови за рахунок застосування методів штучного інтелекту.

Для досягнення поставленої мети необхідно вирішити наступні **завдання:**

- провести аналіз предметної області;
- провести аналіз аналогів розроблюваного вебзастосунку;
- специфікація вимог до ПЗ – вебзастосунку;
- провести аналіз методів штучного інтелекту, що будуть реалізовані у вебзастосунку для вирішення поставлених задач;

- моделювання та проектування ПЗ вебзастосунку;
- розробити front-end частину вебзастосунку на базі технологій React, Tailwind CSS;
- розробити back-end частину вебзастосунку на базі технологій NodeJS, Python, PostgreSQL;
- апробація та тестування ПЗ вебзастосунку.

**Практичне значення:** вебзастосунок вивчення іноземної мови надає можливість ефективно застосовувати, сучасні технології та інструменти для вдосконалення рівня володіння мови. Кінцевий застосунок корисний для більшості людей, які мають середній та нижче середній рівень володіння мови, оскільки може допомогти з покращенням рівня володіння.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Використання методів штучного інтелекту у вивченні іноземних мов

ШІ, або штучний інтелект – це технологія, яка дозволяє комп'ютерам і машинам імітувати людський інтелект і здатність вирішувати проблеми. ШІ пройшов через багато циклів хайпу, але навіть скептикам здається, що випуск ChatGPT знаменує собою поворотний момент. Дивлячись на сучасні тенденції, стрибок вперед відбувся в обробці природної мови [2].

В останні декілька років, ШІ швидко змінює багато аспектів життя людей. Сфера вивчення іноземних, не є винятком, тому розвиток ШІ у цій сфері, має потенціал для підвищення ефективності навчання та використання нових, революційних методів навчання.

Категоризація інструментів ШІ для навчання [3]:

1) інструменти ШІ, орієнтовані на учня, – це програмне забезпечення, яке учні використовують для вивчення предмета;

2) системи, орієнтовані на викладачів, використовуються викладачами з метою зменшення їхнього навантаження та підвищення ефективності виконання конкретних автоматизованих завдань, таких як адміністрування, оцінювання, зворотний зв'язок і виявлення плагіату;

3) інструменти ШІ, орієнтовані на систему, надають інформацію адміністраторам і менеджерам на інституційному рівні, наприклад, допомагають відстежувати тенденції відсіву студентів на факультетах або в коледжах.

Форми застосування ШІ в процесі вивчення іноземної мови [4]:

4) створення персоналізованих навчальних матеріалів. На основі відповідей учня під час навчання адаптивні освітні системи формують його навчальну траєкторію за допомогою призначених навчальних матеріалів. Деякі інструменти на основі штучного інтелекту можуть налаштовувати навчальні матеріали для конкретного учня, курсу, тощо;

5) використання інструментів машинного перекладу. Машинний переклад – це процес, коли комп'ютерне програмне забезпечення використовується для

перекладу тексту з однієї мови на іншу. Результати досліджень показали, що машинний переклад покращує навички у сфері редагування та сприяє зменшенню кількості лексичних і граматичних помилок;

б) використання асистентів для написання текстів зі ШІ. Асистенти для написання текстів зі штучним інтелектом, допомагають користувачам на різних етапах процесу написання текстів. Використовуючи системи штучного інтелекту, вони виправляють граматичні помилки в написаному тексті, надають рекомендації щодо подальшого вдосконалення та додаткові ресурси для подальшого навчання.

7) спілкування з чат-ботами. Чат-боти – це групи комп'ютерних програм, призначених для імітації інтелектуальної мовної взаємодії з людиною. Користувачі можуть навчатися в процесі безпосереднього спілкування з роботом та надавати персоналізовані відповіді на повідомлення учнів, оцінювати їхню успішність і давати поради щодо того, що учням потрібно вдосконалити;

8) застосування програмного забезпечення для вивчення мови зі ШІ. Хмарні онлайн-платформи, що включають NLP, краудсорсинг, елементи гейміфікації, автоматичне розпізнавання мови, автоматичне генерування мовлення та додатки з ШІ-помічником для написання текстів;

9) інтелектуальна віртуальна реальність – це складна система, що об'єднує інструменти розмовного ШІ, технології усвідомлення просторового контексту, а також системи розпізнавання жестів і міміки, NLP, розпізнавання мовлення і технології розуміння природної мови. Надає змогу, практикувати розмовну мову з аватарами на основі ШІ, які імітують реалістичні розмови з носіями мови, що дає змогу набутти вільного володіння мовою та впевненості завдяки високо персоналізованій практиці.

## **1.2 Огляд існуючих аналогів**

Аналіз застосунків-аналогів до ПЗ, що розробляється, проводиться для аналізу переваг та недоліків конкурентів, для виявлення та прогнозування найбільш конкурентоспроможного функціонала, майбутнього застосунку. Аналіз застосунків-аналогів є важливим процесом на основі якого, відбувається складання

специфікації вимог до ПЗ, що розробляється. Для аналізу застосунків-аналогів було обрано: Duolingo (табл. 1.1), Praktika (табл. 1.2), WordUp (табл. 1.3).

Duolingo [5] найбільш популярний та великий за масштабом застосунок для вивчення іноземних, з найбільшою кількістю активних та загальних користувачів. [1] Застосунок використовує гейміфікований підхід для навчання та надає великий спектр вибору мови для навчання, а саме, понад 40 мов.

Таблиця 1.1 – Опис Duolingo

<b>Назва</b>	<b>Duolingo</b>
<b>Розробник</b>	Duolingo, Inc.
<b>Архітектура</b>	Mobile application, web-platform
<b>Мова реалізації</b>	Java, Python, Kotlin, Scala, R, SQL
<b>Функції</b>	<ol style="list-style-type: none"> <li>1) вивчення ІМ;</li> <li>2) вивчення ІМ спираючись на рівень користувача;</li> <li>3) провадження персоналізованого плану навчання та рекомендацій;</li> <li>4) провадження ігрової форми вивчення ІМ;</li> <li>5) вивчення великої кількості мов;</li> <li>6) надання сертифікатів для підтвердження рівня володіння ІМ;</li> </ol>
<b>Переваги</b>	<ol style="list-style-type: none"> <li>1) гейміфікована форма навчання;</li> <li>2) надання можливості вивчення ІМ з повного нуля;</li> <li>3) зосередження навчального процесу на поступовому освоєнні конкретних аспектів ІМ;</li> <li>4) мотиваційна модель для утримання користувачів;</li> <li>5) безкоштовний для використання;</li> <li>6) наявність великого ком'юніті;</li> </ol>

	<p>7) наявність тесту для визначення рівня володіння ІМ;                  8) наявність великих ресурсів у компанії;</p>
<b>Недоліки</b>	<p>1) модульна структура реалізована лише поступовим шляхом;                  2) фокусування лише на ігровій формі навчання;                  3) відсутність можливості вести навчання на мові, яка вивчається;                  4) відсутність поглиблення в теми, які вивчаються;                  5) застосунок не підходить для людей які мають вищий рівень володіння ІМ за середній;                  6) неможливість набути рівня вільного володіння ІМ;                  7) відсутність розмовної практики в навчанні;</p>
<b>Джерело інформації</b>	<a href="https://www.duolingo.com/">https://www.duolingo.com/</a>

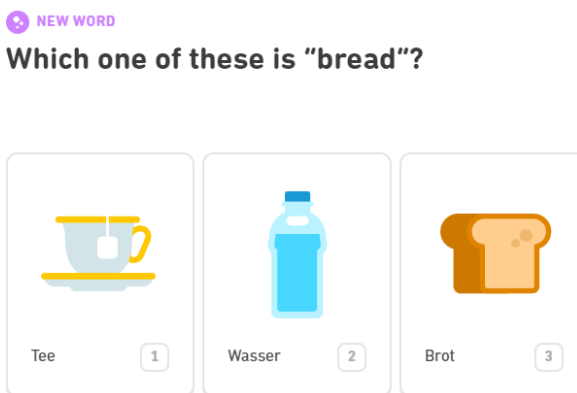


Рисунок 1.1 – Приклад інтерфейсу Duolingo

Дивлячись на інтерфейс цього застосунку, можна зробити висновки, що цей застосунок робить основний фокус на ігровий формат та вивчення основних концепцій ІМ. Загалом Duolingo є найбільшою компанією, яка надає послуги з вивченням ІМ та має найбільшу кількість персоналу, що дає перевагу над іншими.

Застосунок Praktika [6], є одним з найцікавіших на ринку вивчення ІМ, яка фокусується на використанні машинного інтелекту. Основна особливість, цього



застосунку, полягає в застосуванні реалістичних 3D аватарів для реалізації комунікації між застосунком та користувачем.

Таблиця 1.2 – Опис Praktika

<b>Назва</b>	<b>Praktika</b>
<b>Розробник</b>	Praktika.ai
<b>Архітектура</b>	Mobile application
<b>Мова реалізації</b>	Python, Flutter
<b>Функції</b>	<ol style="list-style-type: none"><li>1) вивчення англійської;</li><li>2) вивчення англійської спираючись на рівень користувача;</li><li>3) провадження персоналізованого плану навчання та рекомендацій;</li><li>4) вивчення англійської шляхом комунікації між користувачем та застосунком;</li><li>5) використання аватарів для провадження комунікації;</li><li>6) використання засобів штучного інтелекту для навчання;</li></ol>
<b>Переваги</b>	<ol style="list-style-type: none"><li>1) використання генеративного штучного інтелекту;</li><li>2) надання можливості вивчення ім з повного нуля;</li><li>3) модульна форма навчання;</li><li>4) вивчення шляхом спілкування між користувачем та застосунком;</li><li>5) безкоштовний для використання;</li><li>6) провадження персоналізованого плану навчання;</li><li>7) провадження персоналізованого фідбеку;</li></ol>

	8) використання аватарів для провадження реалістичної комунікації;
<b>Недоліки</b>	1) наявність лише мобільної версії; 2) фокусування лише на спілкуванні шляхом розмови; 3) відсутність поглиблення в теми, які вивчаються;
<b>Джерело інформації</b>	<a href="https://praktika.ai/">https://praktika.ai/</a>

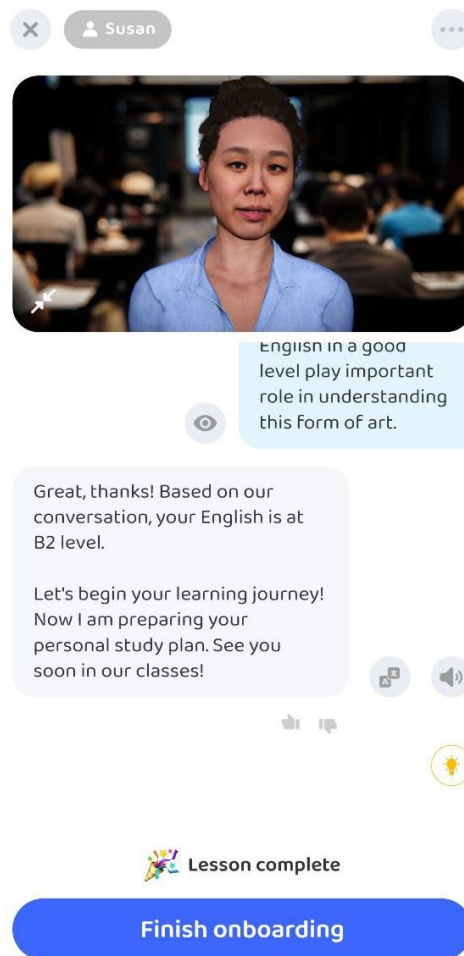


Рисунок 1.2 – Інтерфейс Praktika

Аналізуючи інтерфейс застосунку Praktika, можна зробити висновок, що компанія фокусується на розвитку 3D аватарів з імітацією роботи репетитора. Загалом застосунок, має одну з провідних позицій у своїх жанрів, не дивлячись на лімітовані ресурси.

Застосунок WordUp [7] фокусується на вивченні та підвищенні компетентності в володінні англійської мови, шляхом розширення словникового запасу для загального розуміння та поглибленого розуміння певних тем. Також фокусуванні на застосуванні передових методів штучного інтелекту.

Таблиця 1.3 – Опис WordUp

<b>Назва</b>	WordUp
<b>Розробник</b>	Geeks Ltd
<b>Архітектура</b>	Mobile application
<b>Мова реалізації</b>	Python, C#, .NET, Flutter
<b>Функції</b>	<ol style="list-style-type: none"><li>1) вивчення англійської;</li><li>2) вивчення англійської спираючись на рівень користувача;</li><li>3) провадження персоналізованих налаштувань, щодо рівня володіння ІМ;</li><li>4) вивчення англійської шляхом підвищення рівня словникового запасу;</li><li>5) використання засобів штучного інтелекту для навчання;</li></ol>
<b>Переваги</b>	<ol style="list-style-type: none"><li>1) наявність безкоштовного плану;</li><li>2) надання можливості вивчення ІМ з повного нуля;</li><li>3) розділення модулів вивчення словника на окремі теми;</li><li>4) поглиблення у сфері застосування слів;</li><li>5) провадження текстового, голосового та візуального опису слів;</li><li>6) наявність широкого спектра слів;</li></ol>

	7) можливість використання вивчає мої мови, як основної для роботи застосунку;
<b>Недоліки</b>	1) більшість функцій. доступні лише з платним планом; 2) поглиблене фокусування лише на вивченні словникового запасу; 3) наявність лише застосунку для мобільних пристроїв; 4) використання, лише у зв'язці з іншими інструментами для вдосконалення навичок володіння ім;
<b>Джерело інформації</b>	<a href="https://www.wordupapp.co/">https://www.wordupapp.co/</a>

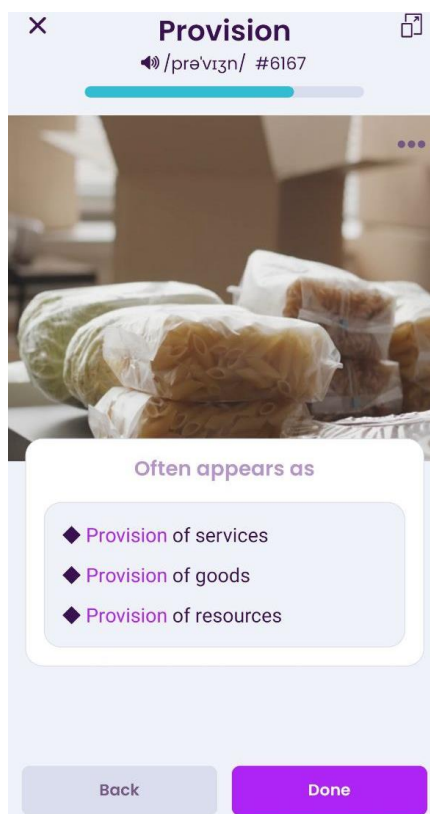


Рисунок 1.3 – Інтерфейс Praktika

Аналізуючи інтерфейс застосунку WordUp, можна зробити висновки, що цей застосунок фокусується на вивченні та вдосконаленні словникового запасу користувача. Інтерфейс є простим та інтуїтивно зрозумілим.

### 1.3 Аналіз системи, що розробляється

З аналізу ринку вивчення іноземних мов, можна зробити висновок, що цей ринок є одним з найбільшим та найперспективнішим на ринці інформаційних технологій, який швидко зростає та очікує подальшого зростання. Ринок характеризується високим попитом на ефективні та доступні методи вивчення мов, а також різноманітністю продуктів та методів навчання. Технології відіграють все більшу роль у вивченні мов, а саме використання штучного інтелекту, що призводить до розробки нових інноваційних продуктів та послуг. Конкуренція на ринку вивчення іноземних мов є також великою та налічує різноманітну направленість ПЗ на вивчення окремих галузей ІМ, також спостерігається різна форма навчання, яка використовується в подібних застосунках [1].

Спираючись на аналіз застосунків-аналогів, було визначено напрям ПЗ, яке розробляється. Вебзастосунок буде фокусуватися на процесі підвищення іншомовної компетентності користувачів, а саме словникового запасу та розумінні матеріалів більш високого рівня володіння ІМ, шляхом використання сучасних інформаційних технологій, а саме методів машинного навчання. Вебзастосунок буде орієнтований на людей, які мають певний рівень володіння ІМ та мають бажання підвищити свій рівень компетенції в ІМ.

Таблиця 1.4 – Опис системи що розробляється

<b>Основні задачі</b>	<ol style="list-style-type: none"><li>1) реєстрація нових користувачів;</li><li>2) авторизація користувачів;</li><li>3) провадження можливостей для вивчення ім;</li><li>4) вивчення англійської шляхом підвищення рівня словникового запасу;</li><li>5) можливість перегляду словника слів іноземної мови;</li><li>6) можливість перегляду словника синонімів слів іноземної мови;</li><li>7) можливість перегляду слів вище або нижче за рівнем;</li></ol>
-----------------------	--

	<p>8) категоризація слів за рівнем;</p> <p>9) аналіз тексту та підбір схожих слів вище за рівнем;</p> <p>10) формування власних словників;</p>
<b>Користувачі системи</b>	<p>1) авторизований користувач;</p> <p>2) неавторизований користувач;</p> <p>3) адміністратор;</p>
<b>Сценарії роботи</b>	<p>1) користувач переглядає визначення певного слова;</p> <p>2) користувач вводить певне слово, вибирає тип підбору слова та отримує підбір слів, згідно з вибраним типом;</p> <p>3) авторизований користувач складає кастомізований словник з вибраними значеннями слів та запускає тест для вивчення вибраних слів;</p> <p>4) користувач вводить текст, вибирає мод для аналізу тексту, після аналізу виділяється слова, згідно з вибраним модом;</p>
<b>Засоби апаратної та програмної реалізації</b>	<p>1) front-end: react, typescript;</p> <p>2) back-end: nodejs;</p> <p>3) machine learning: llama, word2vec;</p> <p>4) databases: postgresql</p>
<b>Вихідні дані</b>	<p>1) список слів ІМ;</p> <p>2) кастомізований підбір слів за певною категорією;</p> <p>3) тест для кастомізованого словника;</p>

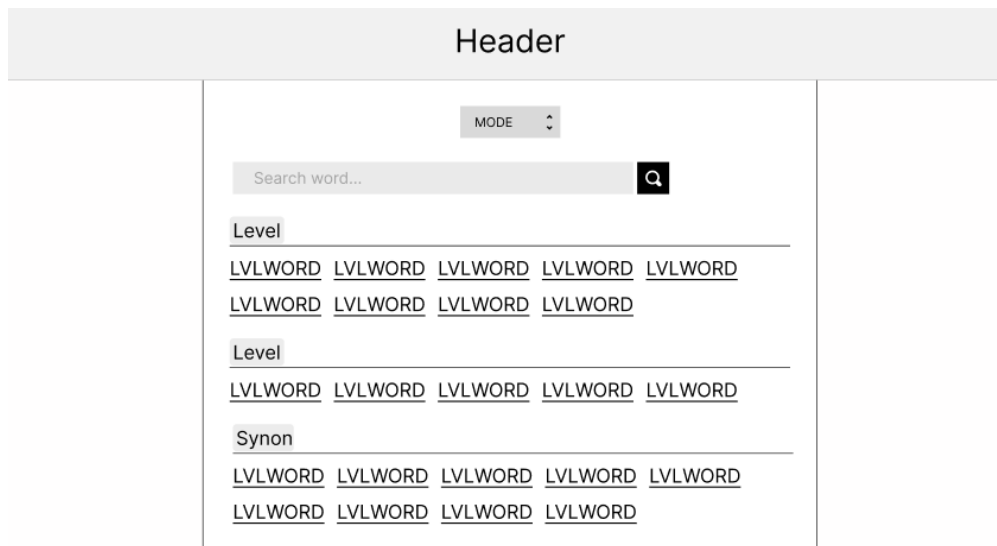


Рисунок 1.4 – Макет розробляемого вебзастосунку

Для створення mock-up, було використано безкоштовний застосунок «Figma». На рис. 1.4 зображено, сторінку, яка містить, поле пошуку для слова, яке користувач вводить. Ця сторінка робить запит до сервера, після обробляє запит та видає схожі слова, але більш високого або меншого рівня. Приклад: big => voluminous. Або підбирає синоніми до цього слова.

## 1.4 Специфікація вимог

### ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

**Призначення системи (застосунку), для якої розробляється програмне забезпечення**

Призначенням застосунку є вивчення іноземної мови, для підвищення ефективності засвоєння навчального матеріалу та мотивації до вивчення іноземної мови за рахунок застосування методів штучного інтелекту.

### Погодження, що ухвалені в програмній документації

Було погоджено, що для створення ПЗ та його злагодженої роботи будуть використовуватися допоміжні фреймворки – React та NodeJS.

### Межі проєкту ПЗ

Крайня дата завершення роботи над ПЗ – 04.06.2024р.

### ЗАГАЛЬНИЙ ОПИС

## **Сфера застосування**

Дане ПЗ має обмеження у його застосування, а саме можливість використання цього ПЗ буде лише у веббраузері, тобто треба буде мати веббраузер на мобільному пристрої або на ПК.

## **Характеристики користувачів**

Основні характеристики користувачів: наявність пристрою на якому є можливість використання веббраузера та підключення до мережі Інтернет.

## **Загальна структура і склад системи**

Основні частини для створення програмного забезпечення: вебзастосунок, сервер NodeJS, сервер для використання бібліотек на основі машинного навчання, сервер бази даних.

## **Загальні обмеження**

Обмеження для роботи з ПЗ – наявність веббраузера та підключення до мережі Інтернет.

## **ФУНКЦІЇ СИСТЕМИ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ**

### **Функція перегляду визначення слова**

#### **Опис функції**

Функція перегляду визначення слова дозволяє користувачу ввести слово, яке потребує визначення та подивитися тлумачення слова, його характеристики та його застосування.

### **Вхідна і вихідна інформація**

Вхідна інформація – слово, яке потребує визначення;

Вихідна інформація – інформація про введене слово.

### **Функціональні вимоги**

БД з великою кількістю слів та їхніми характеристиками.

### **Функція підбору синонімів до слів**

#### **Опис функції**

Функція підбору синонімів до слів допомагає користувачу підібрати схожі слова за значенням до введеного слова, також проводить категоризацію цього слова за рівнем володіння.



### **Вхідна і вихідна інформація**

Вхідна інформація – слово;

Вихідна інформація – список слів, що мають подібне значення.

### **Функціональні вимоги**

БД з великою кількістю слів та їхніми характеристиками. ПЗ для підбору синонімів та категоризації слів.

### **Функція аналізу тексту**

#### **Опис функції**

Функція аналізу тексту, аналізує текст на використання слів різних за рівнем володіння.

### **Вхідна і вихідна інформація**

Вхідна інформація – текст;

Вихідна інформація – лист слів, які вищого або нижчого за рівнем.

### **Функціональні вимоги**

БД з великою кількістю слів та їхніми характеристиками. ПЗ для аналізу тексту.

## **ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ**

### **Джерела і зміст вхідної інформації (даних)**

В ПЗ, що розробляється джерелом вхідної інформації є користувач та його локальна інтернет-мережа.

### **Нормативно-довідкова інформація (класифікатори, довідники тощо)**

В якості словників були обрано відкрите ПЗ, а саме WordNet та The Online Text Plain English Dictionary [8, 9].

### **Вимоги до способів організації, збереження та ведення інформації**

В якості БД обрано PostgreSQL.

## **ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ**

Вимоги до технічного забезпечення не мають жорстких обмежень. Користувач повинен мати комп'ютер, ноутбук, смартфон або інший девайс на якому є можливість встановлення веббраузера та стабільне підключення до мережі Інтернет.

## **ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

### **Архітектура програмної системи**

Архітектура програмного забезпечення складається з клієнтського застосунку (React, Typescript), серверних застосунків (NodeJS, Python), баз даних (PostgreSQL).

### **Системне програмне забезпечення**

Для написання front-end частини обрано фреймворк React, для back-end частини – платформу NodeJS, для застосування машинного навчання – Python. В якості БД обрано PostgreSQL.

### **Мережне програмне забезпечення**

Для створення ПЗ використовується ОС Windows 10, у якості редактору коду обрано VS Code, для роботи з базою даних використано pgAdmin або командну строку, для перегляду вебсторінок – Google Chrome..

### **Програмне забезпечення ведення інформаційної бази**

Ведення інформаційної бази відбувається за допомогою CRUD-операцій з PostgreSQL.

### **Мова і технологія розробки ПЗ**

Програмне забезпечення має бути розроблене з використанням фреймворків React та NodeJS. Бібліотек для машинного навчання – word2vec. Мови розробки – Python, TypeScript.

## **ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ**

### **Інтерфейс користувача**

Вебклієнт має бути зручним у використанні, інтуїтивно зрозумілим та відповідати усім вимогам UX та UI, для створення максимально приємного та зрозумілого досвіду для користувачів, щоб вони могли легко знаходити те, що їм потрібно. Шаблон сторінки повинен бути розділений на дві частини, де перша частина – це набір сторінок, який знаходиться у верхній частині та розміщає посилання на всі функції вебзастосунка, інша сторона, що розміщена знизу відображає контент певної сторінки.

### **Апаратний інтерфейс**

Апаратний інтерфейс – пристрій користувача (ПК, ноутбук чи планшет), який він буде використовувати для взаємодії з сторінками вебклієнта.

### **Програмний інтерфейс**

React – фреймворк з відкритим вихідним кодом для розробки вебінтерфейсів від компанії Meta мовою програмування Javascript або Typescript. NodeJS – платформа, що дозволяє створювати серверні застосунки для реалізації REST запитів, мовою програмування Javascript або Typescript.

### **Комунікаційний протокол**

Застосунок базується на використанні мережних протоколів WAP – протокол бездротової передачі даних та TCP/IP.

## **ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

### **Доступність**

Застосунок є доступним для всіх користувачів, що мають комп'ютер, ноутбук, смартфон або інший девайс на якому є можливість встановлення веббраузера та стабільне підключення до мережі Інтернет.

### **Супроводжуваність**

Супроводження не передбачається.

### **Переносимість**

Вебзастосунок має сумісність з усіма сучасними браузерами.

### **Продуктивність**

Продуктивність ПЗ залежить від швидкості мережі Інтернет та завантаженості серверів .

### **Надійність**

Вхідним даним при реєстрації, буде надана конфіденціальність. Застосунок не дозволить несанкціонований доступ до даних. Користувач матиме змогу отримати доступ до своїх даних і переглянути їх у будь-який час, але з міркувань безпеки вам потрібно буде спочатку увійти в систему. ПЗ буде зроблено всіх можливих мір для виключення можливості отримання персональних даних.

### **Безпека**

Функціонал який дозволяє взаємодіяти з даними має перевіряти токен авторизація та права доступу.

### **Висновки до розділу 1**

В першому розділі кваліфікаційної роботи бакалавра розглянуто та проведено аналіз застосунків-аналогів для вивчення ІМ, проведено аналіз сильних та слабких сторін у конкурентів, також спільні риси що мають їхні застосунки. Набуто практичних навичок з формування специфікації вимог до програмного забезпечення. Під час аналізу, було звернено увагу на архітектуру застосунків-аналогів та їх фокусування у сфері вивчення ІМ.

Було, проведено аналіз системи, що розробляється, шляхом опису основних задач та сценаріїв використання основних функцій застосунку. Наведено приклад інтерфейсу до застосунку. Також розроблено специфікацію вимог до ПЗ, що розробляється, також представлено повний опис поведінки програмного забезпечення.

## 2 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Діаграма прецедентів

Сценарії використання – це перелік дій, які можуть бути виконані користувачами у процесі взаємодії користувача та ПЗ. Сценарії являють собою взаємодію між системою та зовнішніми акторами, показуючи, як система, може відреагувати у певному випадку, а саме, вони описують послідовність кроків, які виконуються користувачем, та відповідні дії системи. Це використовується для прогнозування певного сценарію, не враховуючи технічні чинники. Написання та моделювання сценарію, допомагає в аналізі застосування ПЗ, що розробляється та дозволяє визначати кроки, які необхідно виконати користувачу для досягнення поставленої мети.

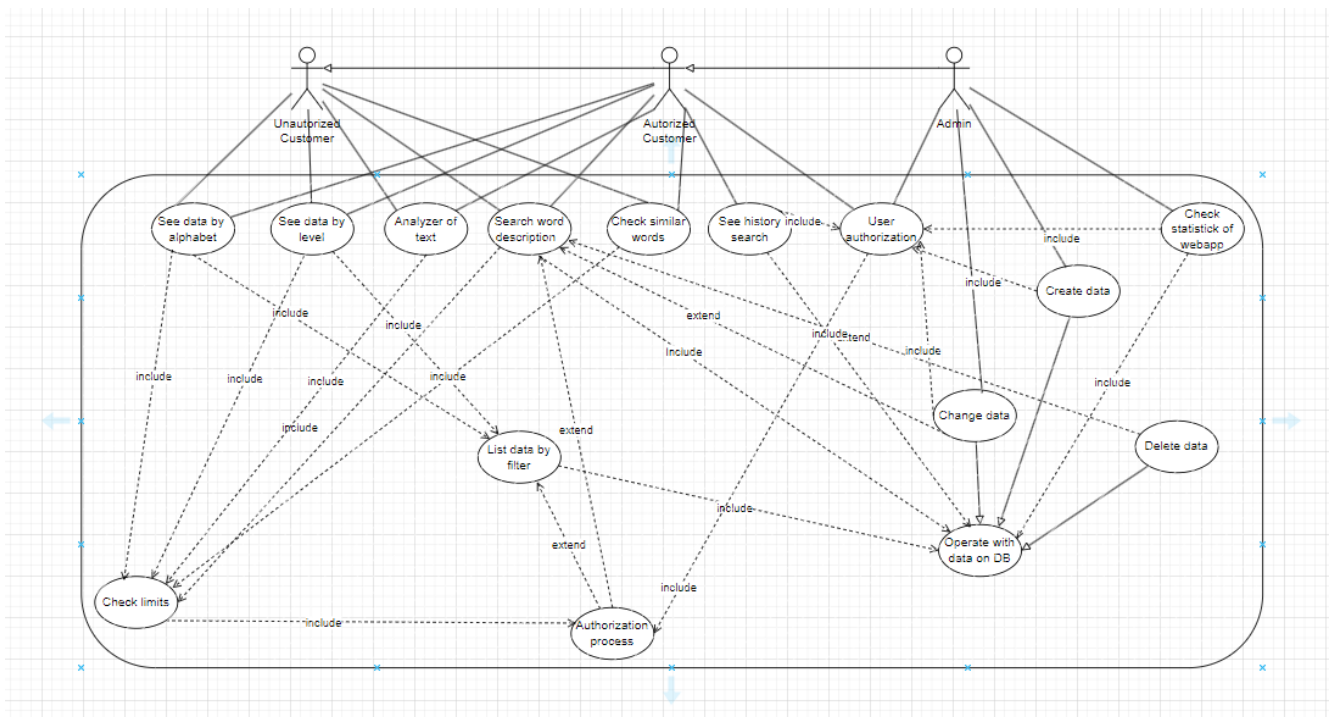


Рисунок 2.1 – Діаграма прецедентів

Існує три форми написання сценаріїв використання:

1) коротка – представляє опис в один абзац одного зі сценаріїв роботи системи; зазвичай вони необхідні під час початкового аналізу вимог до майбутньої системи;

2) поверхнева – поверхневий опис у вільній формі усіх сценаріїв (головного та альтернативних) одного з варіантів використання системи;

3) повна – всі кроки і дії детально описані, включають перед та пост умови виконання. Застосовується для надважливих сценаріїв використання;

**Короткий:** Користувач шукає слова розподілені за рівнем

Користувач відкриває вебзастосунок. Користувач відкриває розділ застосунку в якому розміщені, слова за певним рівнем. Користувач знаходить розділ рівня слів. Обирає певний рівень та переглядає слова цього рівня.

**Поверхневий:** Користувач шукає значення слова

*Головний сценарій (успішний):*

Користувач відкриває вебзастосунок. Користувач відкриває розділ застосунку для пошуку значення слів. Користувач бачить поле для вводу слова та вводить слово, яке його цікавить. Запит відправляється у бізнес частину. Система пошуку слів, знаходить слово із запиту та відправляє його у клієнтську частину. Клієнтський застосунок обробив відповідь від сервера. Користувач отримав на головному екрані значення слова, яке шукав.

*Альтернативні сценарії:*

1) користувач відкриває вебзастосунок, розуміє, що користувацький інтерфейс, незрозумілий та незручний та закриває вебзастосунок;

2) користувач вводить слово, натискає кнопку знайти, збій у клієнтській частині, запит не відбувається, користувач закриває вебзастосунок;

3) користувач вводить слово, відправляється запит, система пошуку слів не знаходить слово, яке шукав користувач, у клієнтському інтерфейсу, відображається повідомлення з результатом пошуку;

4) користувач вводить слово на не англійській розкладці, виконується валідація строчки запиту, валідація провалена, відображається сторінка з помилкою;

5) технічний збій роботи системи. Виводиться повідомлення „немає зв’язку з сервером баз даних”;

**Повний:** Реєстрація в вебзастосунку

**Scope:** система реєстрації в вебзастосунку

**Level:** мета користувача

**Primary Actor:** вебзастосунок

**Stakeholders and interests:**

1) користувач: Зацікавлений в реєстрації у вебзастосунку для отримання певних бенефітів/функціоналу;

2) адмін: Зацікавлений в збиранні певних даних з користувача для подальшого, використання цієї інформації для покращення проєкту;

3) компанія (продавець): зацікавлена в зростанні аудиторії вебзастосунку та покращення метрик зростання компанії;

**Preconditions:** вебзастосунок має працювати без технічних проблем

**Main Success Scenario:**

1) користувач відкриває вебзастосунок;

2) користувач використовує декілька функцій вебзастосунку;

3) користувач вирішує розблокувати фічі та натискає на кнопку зареєструватися;

4) відкривається сторінка з реєстрацією та пропонується ввести пошту, пароль або увійти в акаунт за допомогою oauth;

5) користувач вводить пошту, пароль та натискає на кнопку створити акаунт;

6) система бізнес-логіки обробляє запит на створення акаунта;

7) система бізнес-логіки перевіряє данні із запита;

8) система бізнес-логіки відправляє меседж на пошту для підтвердження створення акаунта;

9) користувач відкриває пошту та підтверджує створення акаунта;

10) користувач відразу переходить на сторінку вебзастосунка, зі своїм акаунтом;

**Extensions:**

а) користувач відкриває вебзастосунок, використовує декілька функцій, бачить основний функціонал лімітований, йому пропонується зареєструватися, він відмовляється реєструватися та залишає вебзастосунок;

b) відкривається сторінка з реєстрацією та користувач вирішує авторизуватися за допомогою OAuth:

- 1) користувач заходить до акаунту за допомогою акаунта гугл;
- 2) користувач прочитавши дані, які будуть надані для OAuth вирішає відмовитися:
  - та продовжити стандартну реєстрацію;
  - зупинити процес реєстрації;

c) система бізнес-логіки обробляє запит на створення акаунта;

- 1) провалідовавши данні користувача, система видає помилку та виводить її користувачу;
- 2) система знаходить такий самий акаунт у одній з баз даних:
  - повертає меседж користувачу, що акаунт з такою поштою вже існує;
  - пропонує залогінитися;

d) Через технічний збій, система бізнес-логіки не може відправити меседж на пошту для підтвердження та життя посилання на активацію прострочується.

e) Система бізнес-логіки відправляє меседж на пошту для підтвердження створення акаунта, але користувач забув, за цей лінк, та підриває процес створення акаунту.

f) Користувач хоче зайти на вебзастосунок, але технічний збій, положив всі сервера та вони не працюють.

g) Користувач вводить пошту, пароль та натискає на кнопку створити акаунт, але механізм валідації спрацьовує та на сторінці відображено цю помилку.

### **Special Requirements:**

- 1) сервера, які забезпечують роботу вебзастосунку;
- 2) domain name;
- 3) зручний інтерфейс;
- 4) забезпечення безпеки даних;
- 5) підтримка різних методів аутентифікації;



### **Technology and Data Variations List:**

- 1) на клієнтській та серверній частині є валідатори для даних;
- 2) створення зручного інтерфейсу реєстрації;
- 3) зручна обробка помилок;
- 4) забезпечення безпеки паролів та трансферу даних;
- 5) підпис покупця для підтвердження кредитної операції робиться на товарному чеці;
- 6) реалізація авторизації через сторонні сервіси;
- 7) реалізація механізму відновлення паролю;
- 8) механізми захисту від атак;

### **Frequency of Occurrence:**

Система може працювати майже безперервно, перивається на оновлення.

### **Miscellaneous (Open Issues):**

- 1) GDPR;
- 2) мультиплатформність;
- 3) бізнес-аналітика та звітність;
- 4) інтеграція з іншими системами;
- 5) доступність;
- 6) контроль версій та релізів;

## **2.2 Діаграма діяльності**

Діаграма діяльності, використовується для моделювання та візуалізації послідовності дій, необхідних для виконання певного бізнес-процесу або алгоритму. Діаграма дозволяє моделювати послідовності бізнес-процесів або дій, реалізованих методами класів. Зазначені послідовності можуть являти собою альтернативні галузей процесу обробки даних або галузям, які можуть виконуватися паралельно. Діаграми діяльності є аналогом блок-схеми будь-якого алгоритму. Вони, як і діаграми станів та переходів, відображаються у вигляді орієнтованого графу, вершинами якого є дії, а ребрами – переходи між діями. Кожний стан на діаграмі діяльності відповідає виконанню деякої елементарної

операції, а перехід в наступний стан виконується тільки після завершення цієї операції. Таким чином, діаграму діяльності можна вважати приватним випадком діаграми станів. Основним напрямком використання діаграми діяльності є візуалізація особливостей реалізації операцій класів, коли необхідно надати алгоритми їх виконання [10].

Для ПЗ, що розробляється було створено три діаграми діяльності:

1) реєстрація в застосунку (рис 2.2). Діаграма містить кроки, які необхідно пройти для реєстрації;

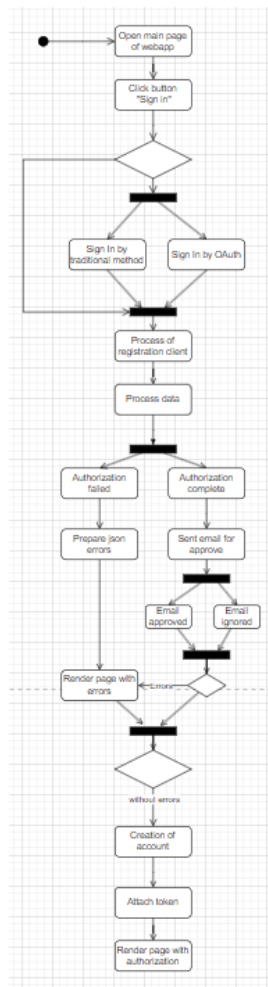


Рисунок 2.2 – Діаграма діяльності реєстрації

2) пошук слова (рис 2.3). Діаграма містить кроки та можливі варіанти, які можливі при виконанні сценарію пошуку слова;

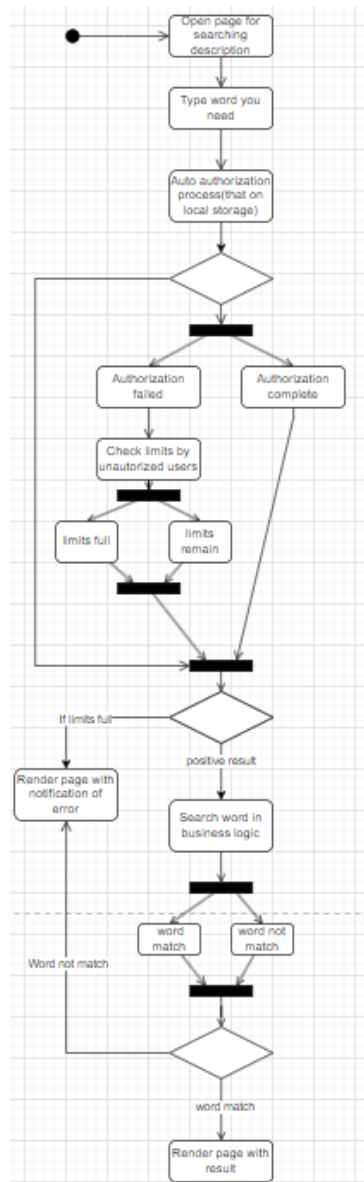


Рисунок 2.3 – Діаграма діяльності пошуку слова

Дана діаграма містить декілька кроків: відкриття сторінки де розміщено ПЗ, введення слова для пошуку його визначення, процес авторизації після якого йду розгалуження варіантів, якщо авторизація успішна застосунок продовжує свою роботу на звіряє чи знайдено слово, та видає результат в залежності від знаходження слова в базі даних, при помилці рендериться сторінка, з назвою помилки; якщо авторизація не була пройдена, спочатку робиться перевірка лімітів для неавторизованого користувача, якщо ліміти не вичерпані, застосунок продовжує функціонувати, якщо ліміти вичерпані рендериться сторінка, в якій сказано, що ліміти використання вичерпані.

3) третя діаграма (рис 2.4) зображує кроки та можливі варіанти, виконання функції аналізу тексту в ПЗ, що розробляється. В діаграмі присутні наступні кроки: відкриття сторінки, процес авторизації та перевірки лімітів використання, вибір типу аналізу тексту, після авторизаційного процесу відбувається процес токенізації тексту, перевірки типу аналізу та бізнес-логіки, що буде задіяна для цієї функції ПЗ, що розробляється;

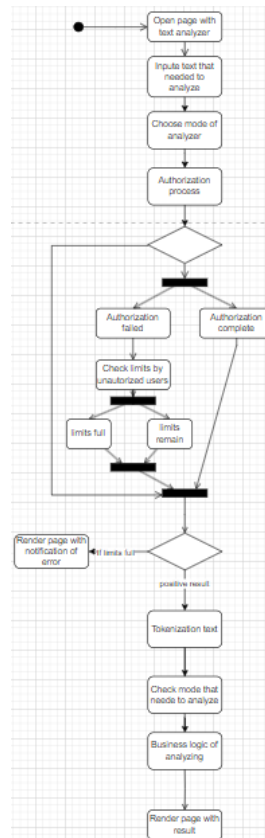


Рисунок 2.4 – Діаграма діяльності аналізу тексту

У другій та третій діаграмі присутні схожі блоки у вигляді авторизаційного процесу, але суттєво відрізняються виконанням процесів ПЗ.

### 2.3 Діаграма розгортання

Діаграма розгортання – це тип діаграми, який визначає фізичне обладнання, на якому працюватиме програмна система. Він також визначає спосіб розгортання програмного забезпечення на базовому обладнанні. Він прив’язує частини програмного забезпечення системи до пристрою, який збирається її виконувати.

Діаграма розгортання відображає програмне забезпечення, створена в процесі розробки фізичної системи, яка його виконує. У розподілених системах він моделює розподіл програмного забезпечення між фізичними вузлами [11].

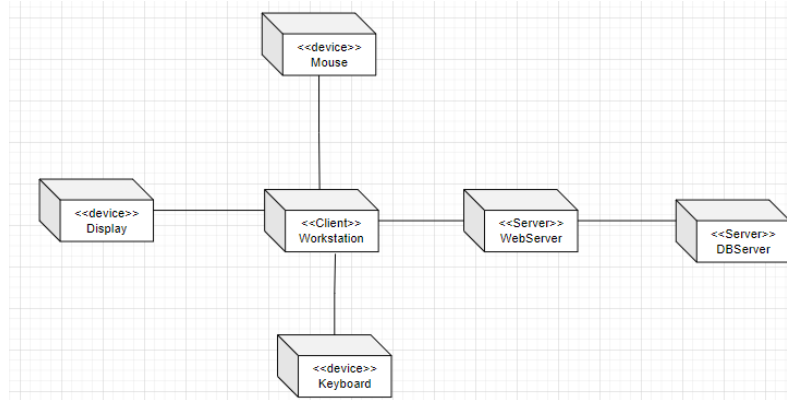


Рисунок 2.5 – Діаграма розгортання

На діаграмі розгортання ПЗ, яке розробляється (рис 2.5), зображено вузли у вигляді прямокутних паралелепіпедів, вони поділені на обчислювані фізичні ресурси та програмні ресурси. В даному випадку вузли пристрою містять, такі об’єкти, як клавіатура, дисплей та загалом ПК. Вузли середовища, являють собою сервіс, який дозволяє виконувати інші програмні елементи, в даному випадку це вебсервер та сервер для бази даних. Ця діаграма містить трьох шарову архітектуру (табл 2.1).

Таблиця 2.1 – Опис системи що розробляється

№ шару	Компоненти	Опис
1	Client – workstation Devices: 1) display; 2) keyboard; 3) mouses;	Робоча станція користувача, яка використовується для роботи з застосунком.
2	Server – Webserver	Вебсер – розміщує, всю бізнес-логіку застосунку, що розробляється
3	Server – DBServer	Розміщує сервер з базою даних.

Було представлено таблицю, яка ілюструє архітектуру ПЗ, що розробляється.

## 2.4 Діаграма взаємодії

Діаграми послідовності є одним із способів формалізації сценаріїв використання. Її перевага заключається в тому, що на ранніх стадіях опису сценаріїв можливо з'ясувати склад взаємодіючих компонентів та описати потік повідомлень від одних компонентів до інших. Ці компоненти та потоки повідомлень в подальшому будуть трансформовані в конкретні класи (об'єкти), методи цих об'єктів. Відповідно, одразу ж з'ясовується і модель системи подій (Actions), які дані класи (об'єкти) будуть підтримувати та обробляти [12].

Перша діаграма послідовності ілюструє (рис 2.6) процес реєстрації у застосунку.

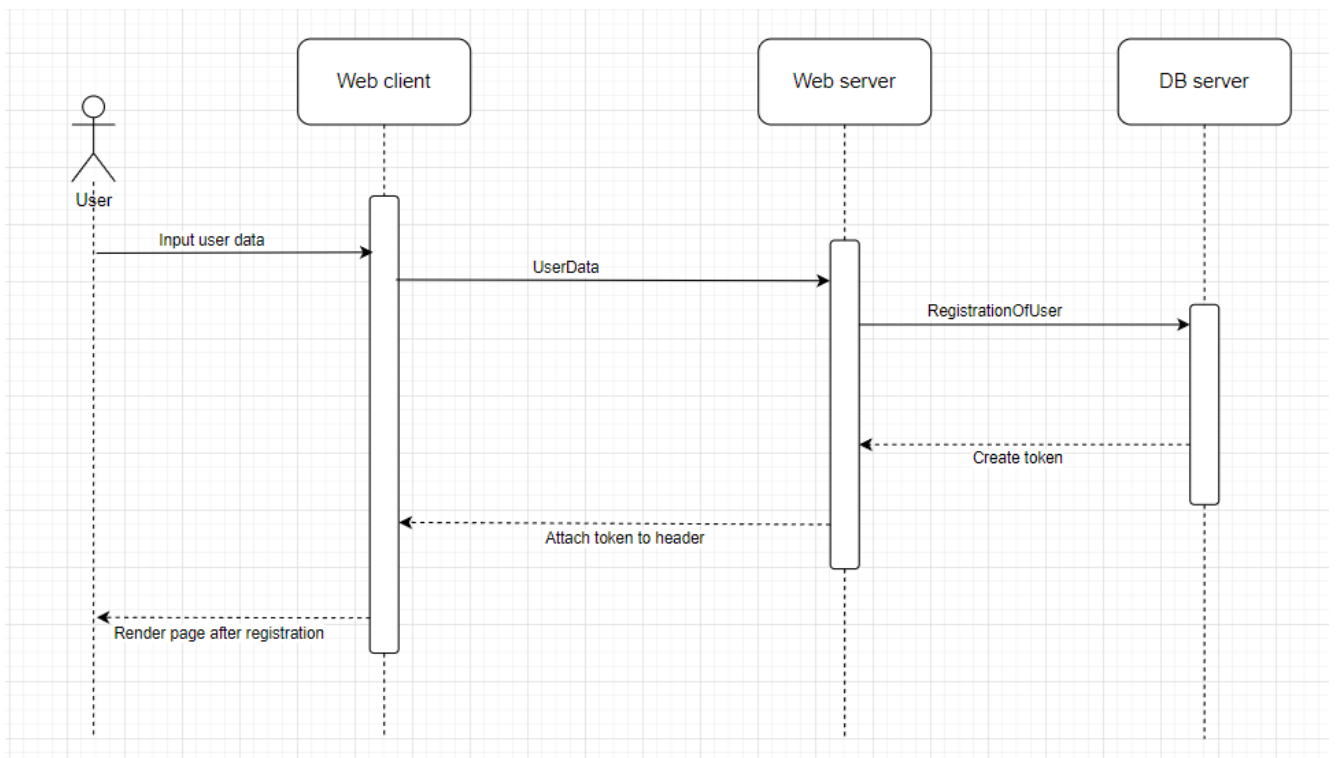


Рисунок 2.6 – Діаграма послідовності реєстрації

Ця діаграма ілюструє поетапну взаємодію користувача для виконання реєстрації у застосунку, а саме користувач вводить дані для реєстрації, потім ці дані передаються з вебклієнта на сервер, на сервері робиться реєстрація, хешуються дані, створюється токен, потім дані заносяться у БД, після прикріплюється токен

до хедеру користувача та рендериться сторінка з реєстрацією, на якій користувач може продовжити роботу.

Друга діаграма послідовності (рис. 2.7) зображує пошук слова у застосунку та послідовність дій для виконання цієї функції.

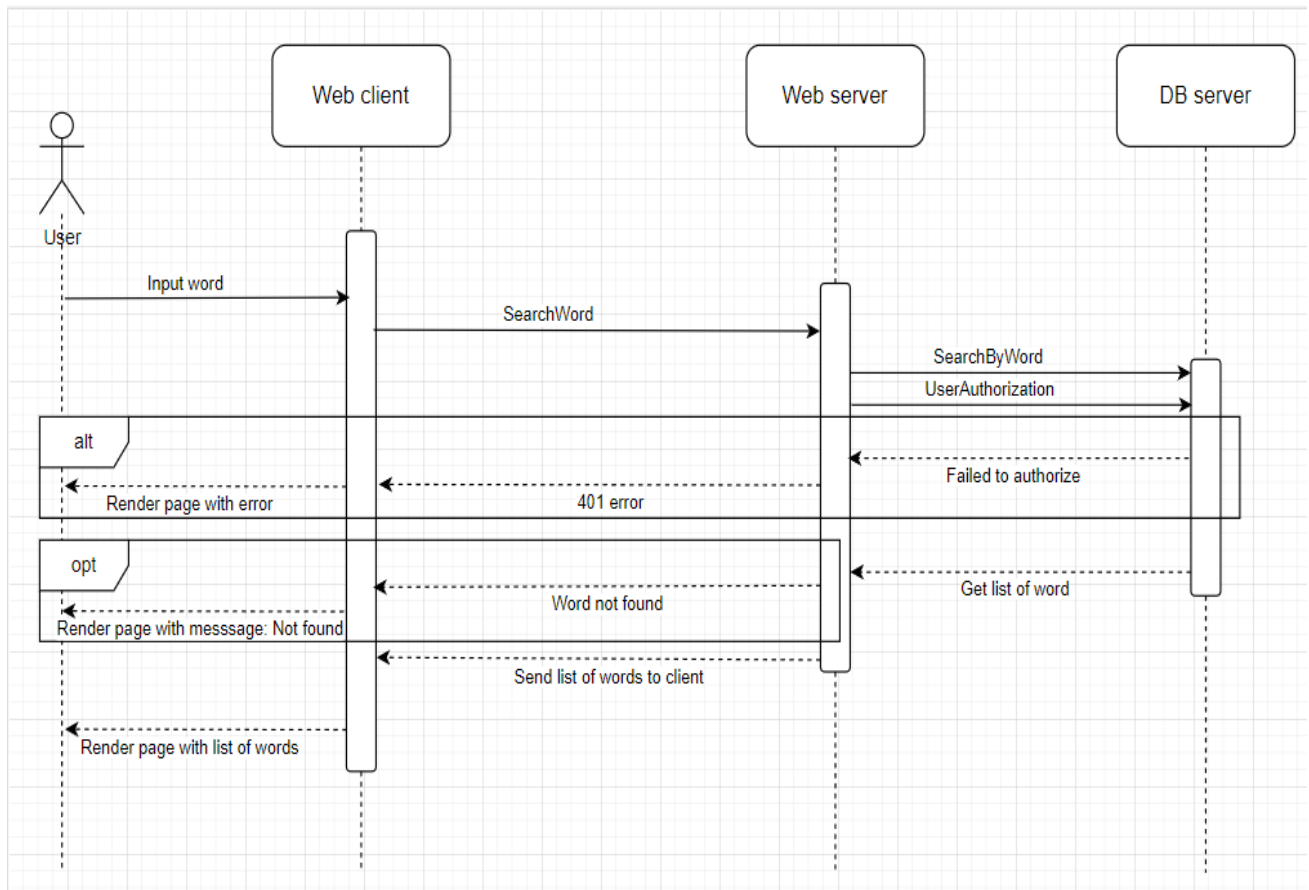


Рисунок 2.7 – Діаграма послідовності пошуку слова

На цій діаграмі, зображено функцію пошуку слова, а саме введення слова для пошуку, потім альтернативний сценарій: якщо авторизація провалена, рендериться сторінка з помилкою 401; присутній опційний сценарій з не знаходженням слова у БД та виведення пустого списку; у фінальному результаті виводиться список слів які схожі на введене слово.

На третій діаграмі послідовності (рис 2.8), зображено функцію аналізу тексту, а саме введення тексту для аналізу у вебклієнті, з вебклієнта текст відсилається на серверну частину, на серверній частині йде перевірка на авторизацію, якщо авторизація провалена рендериться сторінка з помилкою. У

успішному варіанті сервер віддає групу слів згідно з типом аналізу, який рендериться у вебклієнті користувача.

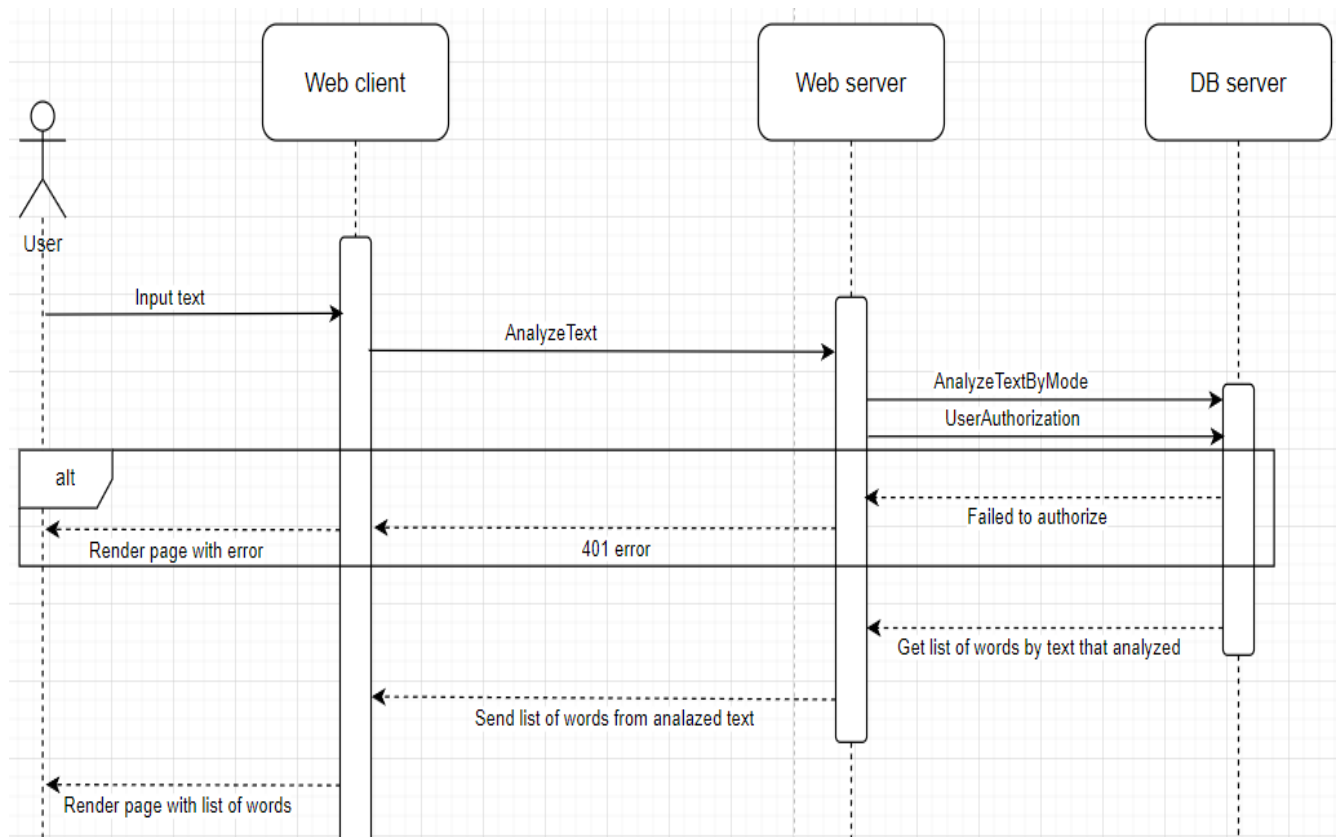


Рисунок 2.8 – Діаграма послідовності аналізу тексту

Хоч друга і третя діаграми схожі, вони містять різні функції застосунку.

У процесі проектування ПЗ, є важливим структура проекту. Структура ПЗ, що розробляється, представлена на рис. 2.9.

На рис 2.9 зображено, клієнта який робить запит на серверну частину, серверну частину у вигляді монолітної структури з API для обробки запитів з клієнтської сторони, у API шляху викликається спеціалізований контролер для обробки цього шляху. У контролері відбувається обробка даних, які були отримані з API, та взаємодія з спеціалізованим сервісом для цього контролера, після виклику сервісу, та проведення операцій контролер віддає отримані назад до API. Сервіс відповідає за взаємодію з БД, а саме для виконання базових CRUD операцій з однією з сутностей.



База даних – розміщена на окремому сервері на виконує операції, які задані у сервісі.

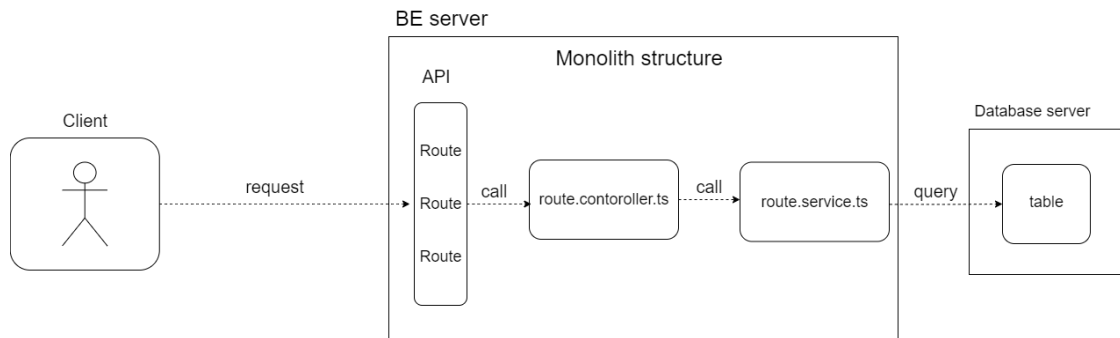


Рисунок 2.9 – Діаграма взаємодії FE, BE та БД

Діаграма, яка представлена на рис. 2.9, відіграє важливу роль у розумінні роботи застосунку та проектуванні ПЗ.

## Висновки до розділу 2

В другому розділі кваліфікаційної роботи бакалавра описано моделювання вебзастосунка вивчення ІМ. Детально описано етапи реалізації вебзастосунка, алгоритми роботи ПЗ. В розділі було наведено діаграму використання з розбором всіх типів сценаріїв використання. Розроблено 3 діаграми діяльності для візуалізації послідовності дій, необхідних для виконання певного сценарію використання вебзастосунку. Продемонстровано архітектуру ПЗ, розробивши діаграму розгортання та таблицю опису всіх шарів архітектури. Для формалізації сценаріїв використання було розроблено 3 діаграми послідовності для різних сценаріїв, щоб на ранніх стадіях опису сценаріїв можливо з'ясувати склад компонентів, що взаємодіють.

## 3 ОГЛЯД СТЕКУ ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ ВЕБЗАСТОСУНКУ

### 3.1 Використані мови програмування

Стек технологій є важливим аспектом у проєктуванні ПЗ. Від вибору мов програмування, фреймворків та бібліотек впливає на те, які можливості може мати ПЗ, також, треба брати на увагу різні технології мають різні характеристики, такі як швидкість, масштабованість та надійність. Правильний вибір технологій може допомогти створити ПЗ, яке буде працювати швидко, без збоїв та ефективно використовувати ресурси. Деякі технології є простішими у вивченні та використанні, цей фактор може вплинути на те, скільки часу та ресурсів буде потрібно для розробки та підтримки ПЗ.

У процесі розробки ПЗ, було використано декілька мов програмування для різних цілей. Основними мовами програмування у ПЗ є Python та Typescript.

Python – одна з найпопулярніших мов програмування. Високорівнева мова загального призначення, універсальність Python дозволяє використовувати її для веброзробки, науки про дані, машинного навчання, написання скриптів та багато іншого. Основним концептом цієї мови програмування є зручність читання та використання нових рядків для завершення команди. Для визначення області видимості, наприклад, області видимості циклів, функцій і класів, Python покладається на відступи з використанням пробілів. Однією з особливостей цієї МП є багато парадигмальність та вибір використання об'єктноорієнтованого, процедурного або функціонального підходу. Python – це мова з відкритим кодом та дуже великою спільнотою, що дає велику кількість бібліотек та відкритих форумів для вирішення проблем.

У ПЗ Python використовується для роботи з NLP та ML, також частково, як допоміжна МП для форматування JSON файлів.

TypeScript – це безкоштовна відкрита мова програмування, що позиціонується, як засіб розробки вебзастосунків, що розширює можливості Javascript, а саме додає до мови необов'язкову статичну типізацію, що дає можливість визначати типи змінних і функцій у своєму кодї, що може допомогти

виявити помилки на ранній стадії розробки та покращити читабельність коду. В останній час мова програмування TypeScript набирає великої популярності, серед спільноти розробників. На відміну від JavaScript, який все ще не підтримує об'єктноорієнтоване програмування, TypeScript підтримує концепції об'єктноорієнтованого програмування у вигляді класів, інтерфейсів та успадкування. TypeScript допомагає підвищити продуктивність розробників, роблячи код простішим для розуміння та підтримки. Найпоширенішим способом використання TypeScript є використання офіційного компілятора TypeScript, який транслює код TypeScript у JavaScript.

У ПЗ TypeScript використовується у BE та FE, та заміняє собою Javascript.

### 3.2 Технології розробки клієнтської та серверної частини

Основними технологіями для розробки FE частини ПЗ є React, TailwindCSS, DaisyUI.

React є популярною бібліотекою з відкритим кодом, для створення користувацьких інтерфейсів, на основі JavaScript та великою спільнотою розробників. React додатки будуються з використанням багаторазових компонентів для інтерфейсу користувача.

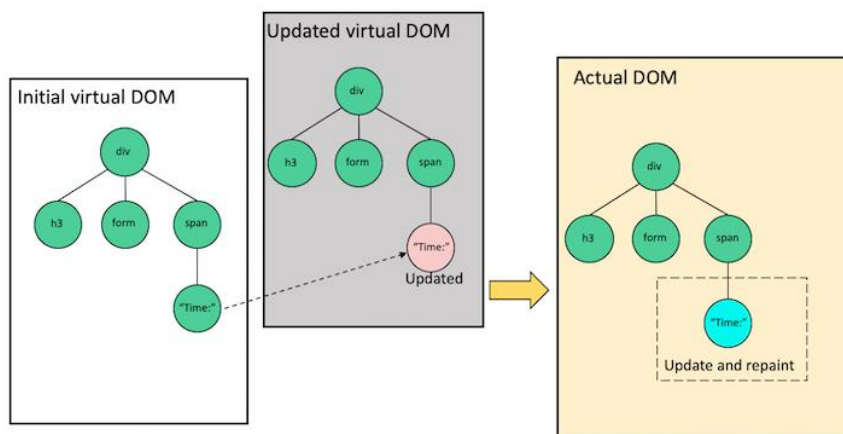


Рисунок 3.1 – Схема роботи React DOM

React використовує віртуальний DOM для ефективного оновлення реального DOM у браузері, зображено на рис. 3.1.

Ця функція дозволяє перемальовуватися лише ті частини інтерфейсу, які дійсно змінюються, що покращує ефективність.

Замість стандартної версії React з JavaScript, у ПЗ використовується React з TypeScript. Також разом з React, використовується декілька бібліотек, які облегшують роботи ПЗ, а саме  `Yup` ,  `formik` .

`Yup`  – це також популярна бібліотека JavaScript для перевірки даних. Вона використовується для визначення правил для того, які дані очікуються, а потім перевіряє, чи відповідають надані дані цим правилам. Це корисно при створенні вебдодатків, щоб переконатися, що користувачі вводять дані в правильному форматі.

`Formik`  – це популярна бібліотека з відкритим вихідним кодом, розроблена спеціально для спрощення створення та обробки форм у React. Вона вирішує деякі з повторюваних аспектів розробки форм.  `Formik`  надає вбудовані функції для визначення правил валідації та відображення чітких повідомлень про помилки для користувача.

Для стилізації React компонентів, використовується бібліотеки  `Tailwind CSS`  та  `daisyUI` .

`Tailwind CSS`  – це популярний фреймворк CSS з відкритим вихідним кодом, який використовує інший підхід до стилізації ніж стандартні (рис. 3.2). Замість готових компонентів (кнопок, меню тощо),  `Tailwind`  надає набір низькорівневих класів утиліт, які націлені на конкретні аспекти дизайну елемента, такі як поля, відступи, розмір шрифту або колір фону.  `Tailwind`  легко інтегрується з адаптивним дизайном. Це забезпечує велику гнучкість і прискорює розробку.

```
<header className="border-b-2 border-solid border-gray-400">  
  <div className="max-w-[1280px] w-full flex justify-around center-h">  
    <Link to="/" className="text-xl uppercase font-bold py-2 px-5">
```

Рисунок 3.2 – Приклад стилізації за допомогою  `TailwindCSS`

`DaisyUI`  – це додаткова бібліотека для  `Tailwind CSS` . Вона будується на основі основної функціональності  `Tailwind` , щоб надати колекцію готових компонентів інтерфейсу користувача.  `DaisyUI`  пропонує багатий набір попередньо розроблених

компонентів, таких як кнопки, картки, модальні елементи, навігаційні меню та багато іншого. Ці компоненти використовують утилітарні класи Tailwind, що дозволяє легко інтегрувати їх у проект.

Для розробки BE частини ПЗ, використовується NodeJS, ExpressJS, TypeORM, та TypeScript у якості основної мови програмування.

NodeJS – це кросплатформенне середовище виконання JavaScript, яке дозволяє розробникам використовувати JavaScript для створення серверних додатків і скриптів. Традиційно JavaScript використовувався для розробки інтерфейсів веббраузерів. Node.js дозволяє розробникам використовувати ту ж мову на стороні сервера, що може спростити процес розробки. NodeJS використовує движок JavaScript V8, той самий, що забезпечує роботу Google Chrome, для виконання коду JavaScript поза веббраузером (рис. 3.3). NodeJS призначений для створення масштабованих мережевих додатків. Він використовує архітектуру, керовану подіями, та асинхронне програмування для ефективної обробки декількох запитів без блокування.

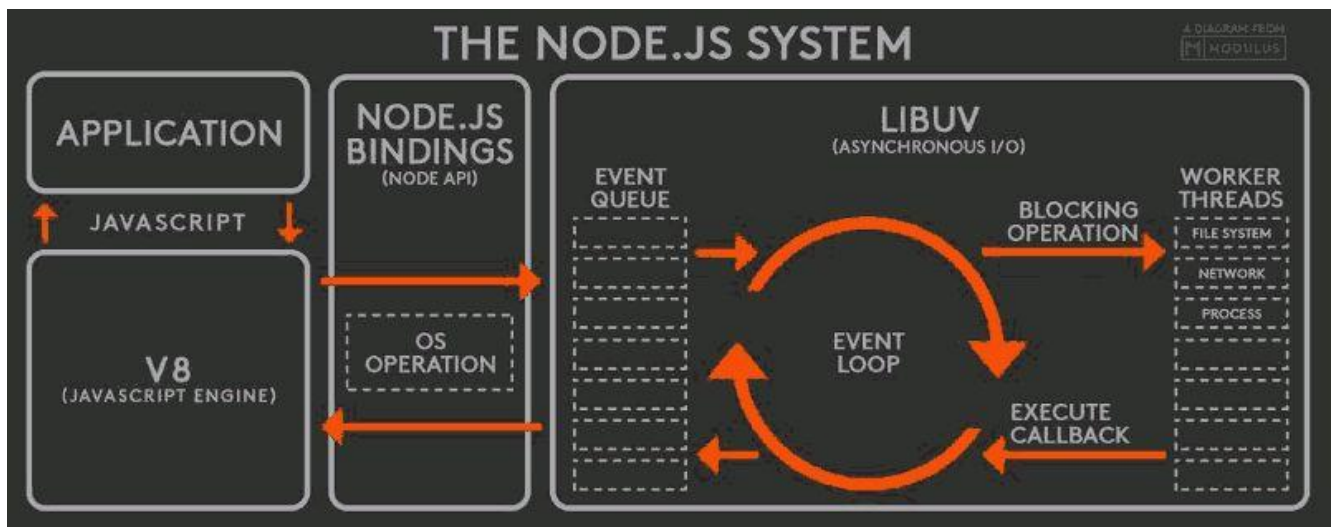


Рисунок 3.3 – Принцип роботи NodeJS

Переваги використання NodeJS:

- повно стекова розробка за допомогою JavaScript;
- масштабованість і продуктивність;
- велика спільнота та екосистема.

ExpressJS – фреймворк з для Node.js, який спрощує створення вебдодатків та API. Він надає надійний набір функцій для розробників, включаючи: маршрутизація, проміжне програмне забезпечення, шаблонування.

TypeORM – фреймворк, написаний на TypeScript, який спеціально розроблений для використання з JavaScript. Він дозволяє взаємодіяти з реляційними базами даних у більш об'єктноорієнтованого спосіб за допомогою коду JavaScript. Саме завдяки цьому фреймворку реалізована взаємодія між FE частиною застосунку та базою даних, також надає можливість реалізувати сутності на рис. 3.1, 3.2 у середовищі розробки на працювати напряму з ними.

### **3.3 Огляд словників англійської мови**

Для вебзастосунка вивчення іноземних мов, інформаційна база є однією з важливих компонентів ПЗ. В ПЗ, що розробляється, використовуються словники з відкритим кодом, які дозволяють без обмежень використовувати їх для розробки будь-яких видів застосунків. БД зі словами та значенням до цих слів, є важливим функціоналом для ПЗ, що розробляється.

WordNet – це велика лексична база даних англійської мови. Іменники, дієслова, прикметники та прислівники згруповані в набори когнітивних синонімів (synsets), кожен з яких виражає окреме поняття. WordNet також є вільно і публічно доступним для завантаження. Структура WordNet робить його корисним інструментом для комп'ютерної лінгвістики та обробки природної мови [9]. У застосунку використовується модифікована версія, цього словника, а саме Wordset by Wordset, Inc..

Wordset – це структурований словник. Головною метою, цього проекту було, зручність використання та створення спільноти для колективної роботи над створенням словника. Цей словник є проектом компанії Wordset, Inc., який ставив ціль у створенні словника з англійської мови у відкритому доступі на основі словника WordNet, активна фаза відбувалася з 2015 до 2017 року, але проєкт був закритий. Словник поширює більше ніж 100 тисяч унікальних слів, з визначеннями до них, кожне слово може мати декілька значень, загальна кількість значень сягає

більше ніж 177 тисяч, також визначається частина мови до кожного значення слова. На відмінну від WordNet, який є набором інструментів природної мови, цей словник становить собою базу даних у вигляді файлів формату JSON, у алфавітному порядку. Всі значення та визначення слів, дотримуються певних правил, які визначенні у спільноті [13].

OPTED від EDMTDev – це словник, який є суспільним надбанням з відкритим доступом, вільний для використання для будь-яких проєктів, як комерційних, так і некомерційних. Словник був розроблений на основі US Webster's Unabridged Dictionary 1913 року, тому деякі слова та визначення можуть бути застарілими у порівнянні з сьогоденням. Не дивлячись на крихітні недоліки, цей словник є гарним для використання у розробці ПЗ. Компанія EDMTDev оцифрувала цей словник у формат JSON та SQLite БД [8].

### **3.4 Технології обробки природної мови**

Застосунок, що розробляється використовує деякі методи ШІ, найбільше використовується методи однієї з галузі ШІ – NLP.

NLP [14] – поєднує обчислювальну лінгвістику - моделювання людської мови на основі правил – зі статистичними моделями та моделями машинного навчання, що дозволяє комп'ютерам і цифровим пристроям розпізнавати, розуміти і генерувати текст і мовлення.

NLP, лежить в основі додатків і пристроїв, які можуть:

- перекладати текст з однієї мови на іншу;
- реагувати на введені або вимовлені команди;
- розпізнавати або аутентифікувати користувачів за голосом;
- узагальнювати великі обсяги тексту;
- оцінювати наміри або смисл тексту чи мовлення.

Однією з функцій застосунку, будуть операції з синонімами, тому використання синсетів є одним з важливих компонентів застосунку, для цього використовується WordNet [9], який вбудований у NLTK.

Основним зв'язком між словами у WordNet є синонімія. Синоніми – слова, що позначають одне й те саме поняття і є взаємозамінними в багатьох контекстах. Кожна з 117 000 синонімічних груп WordNet пов'язана з іншими групами за допомогою невеликої кількості «концептуальних зв'язків» (рис. 3.4).

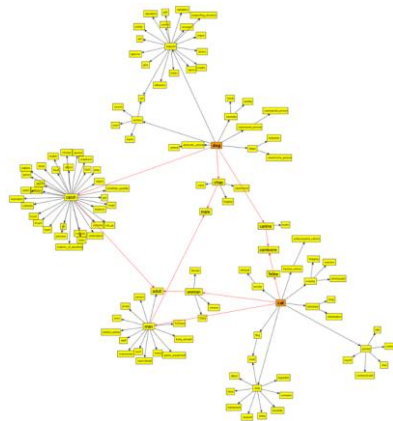


Рисунок 3.4 – Ілюстрація концептуальних зв'язків WordNet

Python надає широкий спектр інструментів і бібліотек для вирішення конкретних завдань NLP. Багато з них можна знайти в Natural Language Toolkit, або NLTK, колекції бібліотек, програм та освітніх ресурсів з відкритим вихідним кодом для створення програм NLP.

NLTK включає бібліотеки для багатьох завдань NLP, перерахованих вище, а також бібліотеки для підзадач, таких як синтаксичний аналіз речень, сегментація слів, стемінг і лематизація і токенізація. Він також включає бібліотеки для реалізації таких можливостей, як семантичне міркування, здатність робити логічні висновки на основі фактів, витягнутих з тексту [14].

Word2Vec – це метод, який використовується в NLP для перетворення слів у числові представлення. Ці представлення, які називаються вбудовуваннями слів, фіксують значення і зв'язки між словами на основі оточуючого тексту. Word2Vec навчається на великому масиві даних тексту. Це дозволяє йому вивчати контекст, в якому використовуються слова. У цьому процесі кожному слову присвоюється унікальний вектор чисел. Слова зі схожими значеннями матимуть вектори, розташовані близько один до одного в цьому числовому просторі.



У застосунку використовується готова модель WE, від компанії Google. Модель навчається на великому наборі текстових даних, зокрема, на частині Google News, що містить близько 100 мільярдів слів. Модель містить векторні представлення для 3 мільйонів слів і словосполучень. Кожне слово або словосполучення представлено 300-вимірним вектором. Ця характеристика робить модель Google корисною для різних завдань NLP, в тому числі для текстової схожості.

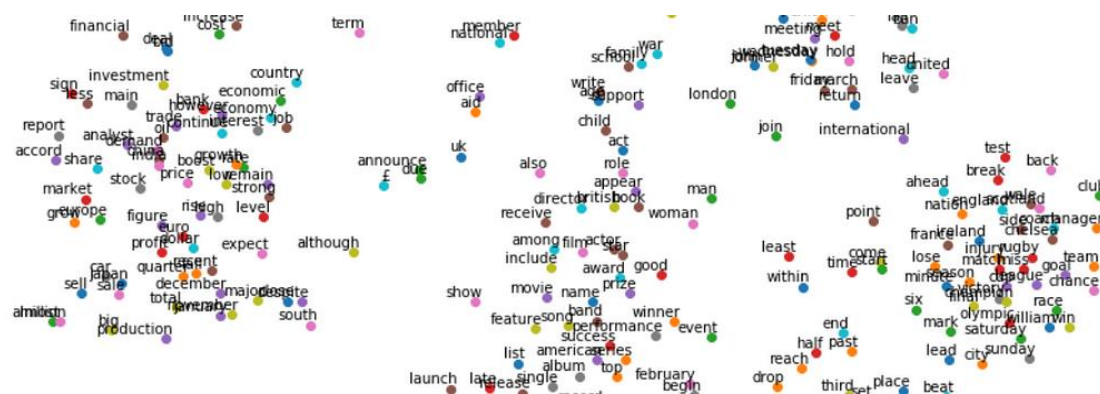


Рисунок 3.5 – Репрезинтація векторів моделі

На рис. 3.5 можна побачити ілюстрацію векторів та їх розташування між собою.

### Висновки до розділу 3

В третьому розділі кваліфікаційної роботи бакалавра описано проектування вебзастосунку вивчення іноземних мов на основі методів штучного інтелекту. Наведено UML-діаграми, які відіграють важливу роль у проектуванні застосунку. Було проведено огляд стеку технологій для клієнтської, серверної частини програмного забезпечення.

При огляді стеку технологій було надано характеристику, певним елементам система, розглянуто їх місце у ПЗ та основні функції. Розглянуто актуальність мов програмування вебзастосунка, фреймворків, а також БД.

## 4 ПРОГРАМНА РЕАЛІЗАЦІЯ

### 4.1 Діаграми класів

UML-діаграми є одним з найзручнішим способом описання бізнес-процесів та відображення структури проєкту для ПЗ. Діаграма класів є однією з основних для опису об'єктноорієнтованої структури застосунку. Діаграма класів визначає сутності, демонструє властивості системи, операції та зв'язки між об'єктами для опису структури системи. Діаграма класів, відіграє важливу роль у проєктуванні структури ПЗ, бо застосовується для переведення сутностей у програмний код.

В даному розділі представлено 2 діаграми класів для користувача та для словника.

Діаграма класів словника (рис. 4.1) ілюструє взаємодію між 4 сутностями, а саме слово, визначення, приклад та першу літеру слова.

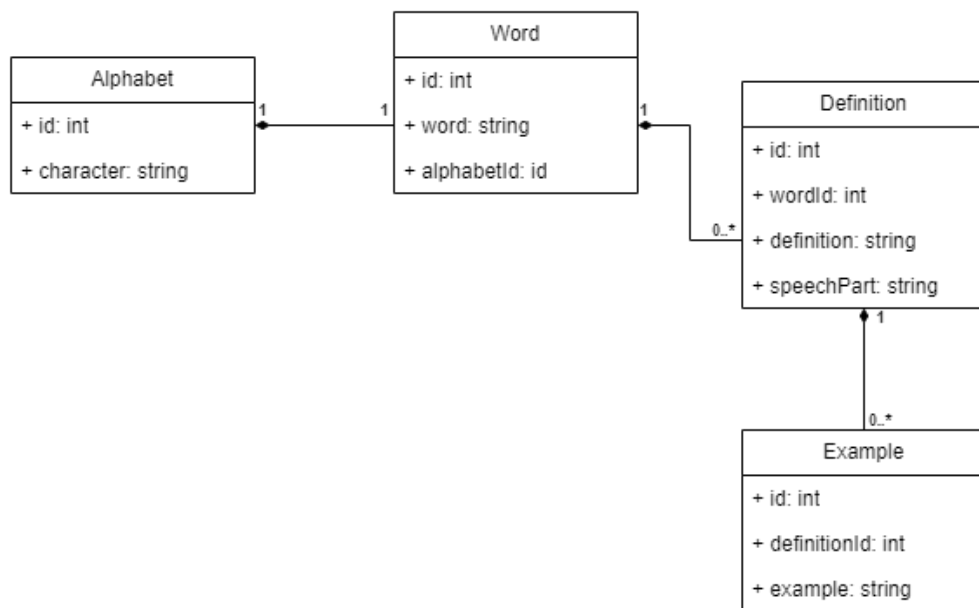


Рисунок 4.1 – UML-діаграма класів словника

Сутності з цієї діаграми класів використовуються для проєктування схеми БД, та провадження моделей у VE частині застосунку для взаємодії з БД. Методи у діаграмі відсутні.

Діаграма класів користувача (рис. 4.2) ілюструє сутності які залученні в процесі авторизації, користувача та зберігання інформації про користувача БД. Методи у діаграмі відсутні, бо ілюструються зв'язки між моделями таблиць БД.

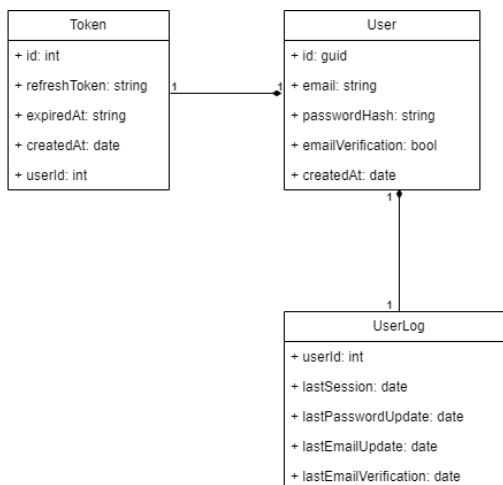


Рисунок 4.2 – UML-діаграма класів користувача

Загалом UML-діаграми, а саме, діаграма класів відіграють важливу роль у проектуванні ПЗ, та зокрема зв'язків між сутностями ПЗ.

#### 4.2 Формування єдиної бази даних слів

В якості інформаційного ресурсу для бази даних було використано словники Wordset та цифрову версію OPTED. В початковому вигляді база даних Wordset, знаходилась у форматі JSON, БД OPTED у форматі SQLite.

```

OPTED
Number of words from OPTED dictionary: 176064
Number of unique words from OPTED dictionary: 110473
Unique words comparing with Wordset: 70073
Number of words that shared with Wordset: 40400

Wordset
Number of words from Wordset dictionary: 108124
Number of unique words from Wordset dictionary: 107943
Unique words comparing with OPTED: 67543
Number of words that shared with OPTED: 40400
    
```

Рисунок 4.3 – Порівняння словників до злиття

В оригінальному вигляді, БД Wordset містила багато зайвих полів, тому було вирішено змінити структуру JSON. Було прибрано більшу частину полів.

Для того, щоб перетворити SQLite формат в Postgres, треба було застосовувати іншостороні бібліотеки. Головною проблемою переформатованої бази даних, стала відсутність структури БД, а саме злиття всіх стовпців одну таблицю. Для більшої зрозумілості системи та зручнішого використання, було прийнято рішення по злиттю двох словників в один, з Postgres, як основною БД. Базу даних було розподілено на таблиці: alphabet, words, definitions, examples.

```

50 with open(f"merge/merged_dictionaries.json", 'r', encoding='utf-8') as file:
51     merged_dictionaries = json.load(file)
52
53     words = 0
54     meanings = 0
55
56     for el in merged_dictionaries:
57         meaning = el['meanings']
58         words += 1
59         meanings += len(meaning)
60

```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

Number of unique words from two dictionaries: 178011  
 Number of meanings to words: 339327

Рисунок 4.4 – Характеристика спільного словника

Однією з ключових проблем, з якою зіткнулися під час розробки системи, стала повторюваність слів. Ця проблема викликала ускладнення в пошуку та обробці інформації, а також могла призвести до неточностей та розбіжностей у даних. Для вирішення цієї проблеми було прийнято рішення звести всі слова до єдиного формату, а саме lowercase. Це означає, що всі допоміжні характеристики слова, такі як його регістр, були перетворені на нижній регістр.

Цей підхід дозволив усунути дублювання записів та спростити порівняння слів. Важливо зазначити, що в базі даних зберігається лише один варіант слова. Це гарантує, що всі посилання на слово, незалежно від їхнього формату, ведуть до одного й того ж запису. Окрім цього, для полегшення сортування слів було створено окрему таблицю, що містить лише першу літеру слова. Ця таблиця дозволяє швидко знаходити групи слів, що починаються з певної літери, що значно економить час та ресурси при пошуку й обробці інформації. Варто підкреслити, що

всі операції з об'єднання та форматування слів були виконані за допомогою мови програмування Python.

Query Query History

```

1 SELECT w.id, alp.character, w.word, def.speech_part, def.definition, ex.example
2 FROM words w
3 INNER JOIN alphabet alp ON w.id_alphabet = alp.id
4 INNER JOIN definitions def ON def.id_word = w.id
5 LEFT JOIN examples ex ON ex.id_definition = def.id
6 WHERE w.word = 'computer'
7 ORDER BY alp.character;
    
```

Data Output Messages Notifications

	id bigint	character text	word text	speech_part text	definition text	example text
1	19275	c	computer	noun	a machine for performing calculations automatically	[null]
2	19275	c	computer	noun	an expert at calculation (or at operating calculating machine...	[null]
3	19275	c	computer	(n.)	One who computes.	[null]

Рисунок 4.5 – Порівняння словників

На рис. 4.5 зображено взаємодію між таблицями.

Після ретельного виконання всіх операцій з об'єднання словників, була розроблена ERM-діаграма, що значно спрощує розуміння структури бази даних та взаємозв'язків між таблицями (рис. 4.6).

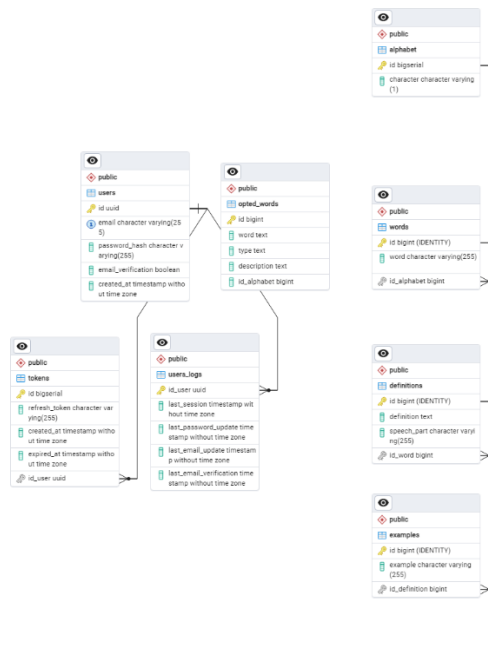


Рисунок 4.6 – ERM діаграма БД

Ця діаграма слугує цінним інструментом розуміння системи, надаючи чітке та лаконічне представлення даних.

### 4.3 Реалізація алгоритму обробки природньої мови

Робота з методами штучного інтелекту займає одну з ключових ролей у цьому застосунку. Особлива увага приділяється роботі з моделлю word2vec, яка використовується для перетворення слів у вектори чисел, що дозволяє комп'ютеру "розуміти" текстові дані. Це перетворення є критичним для подальшого аналізу та обробки текстової інформації. Модель word2vec працює на основі нейронних мереж і дозволяє знаходити семантичні зв'язки між словами. Здатність до семантичного аналізу є надзвичайно корисною для виділення семантичних зв'язків між словами, та знаходження схожих слів. Саме така модель буде використовуватися для знаходження схожих слів.

```
model = KeyedVectors.load_word2vec_format('./model/GoogleNews-vectors-negative300.bin', binary=True)
model.init_sims(replace=True)
model.save('GoogleNews-vectors-norm.bin')
```

Рисунок 4.7 – Код для нормалізації векторів

Оптимізація моделі є невід'ємною частиною процесу її впровадження. Ефективна оптимізація дозволяє значно підвищити точність моделі та зменшити час її тренування, що є важливим для забезпечення швидкої та надійної роботи застосунку.

У вебзастосунку використовується модель від Google на майже 100 млрд. слів, тому ця модель у стандартному вигляді, має дуже низьку ефективність. Для оптимізації було виконано нормалізацію векторів. Нормалізація векторів – це процес перетворення векторів на такі, що мають одиничну довжину.

```
similar_words = model.most_similar(word, topn=30)
words_for_validation = []
for word, score in similar_words:
    words_for_validation.append(word.lower())

validated_words = dict_validation(words_for_validation)
res.extend(validated_word for validated_word in validated_words)
```

Рисунок 4.8 – Код для валідації з БД

На рис. 4.8 зображено, приклад API для взаємодії з іншими сервісами.

Крім того, важливим аспектом є інтеграція `word2vec` з іншими компонентами системи. Це включає розробку інтерфейсів для взаємодії з іншими модулями, забезпечення узгодженості даних та їхньої обробки в реальному часі. Весь цей процес вимагає глибоких знань у галузі машинного навчання, програмування та обробки природної мови. Ця модель налаштована на виявлення семантичних зв'язків, тому деякі слова можуть не вписуватися у контекст синонімів, для вирішення цієї проблеми було вирішено зробити валідацію слів з БД. Для цього робиться запит до сервісу на FE частину систему з словами, там проводиться валідація та повертаються слова, які співпадають з інформаційною базою. Потім слова додаються у об'єднану відповідь, на рис. 4. зображено використання моделі `word2vec` та валідації з БД.

Для інтеграції моделі у застосунок, було прийнято рішення у створенні нескладного API з використанням `FastAPI`. API для використання моделі є необхідною структурою для вебзастосунка.

```
app = FastAPI()

origins = ["http://localhost:5173"]

app.add_middleware(
    CORSMiddleware,
    allow_origins=origins,
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

@app.get("/")
async def root():
    return {"message": "Hello from FastAPI!"}

class Word(BaseModel):
    word: str

class Text(BaseModel):
    text: str

@app.post("/synonyms")
async def synonyms(word: Word):
    received_word = word.word
    synonyms = find_synonyms.find_synonyms(received_word)
    return synonyms

if __name__ == "__main__":
    import uvicorn
    uvicorn.run("main:app", host="localhost", port=8000)
```

Рисунок 4.9 – API для моделі

Також для встановлення семантичних зв'язків, використовуються синсети.

Для цього застосовується wordnet з nltk.corpus, вони надають готові синоніми для певного слова.

```
from nltk.corpus import wordnet
synonyms = wordnet.synsets(word)
first_synset = synonyms[0]
synonym_words = [lemma.name() for lemma in first_synset.lemmas()]
```

Рисунок 4.10 – Використання семантичних зв'язків wordnet

Отже основними засобами завдяки, яким створюються семантичні зв'язки є wordnet та word2vec.

#### 4.4 Створення інтерфейсу для взаємодії з моделю

Розробка клієнтського застосунку є необхідним компонентом багатьох успішних проєктів. Це інтерфейс, через який користувачі взаємодіють з продуктом або сервісом, і він відіграє вирішальну роль у загальному досвіді користування. Клієнтський застосунок має бути простим у використанні та зрозумілим для користувачів, щоб вони могли легко знаходити те, що їм потрібно, і виконувати необхідні дії. Застосунок має бути доступним на різних платформах, таких як мобільні пристрої, настільні комп'ютери та веббраузери, щоб охопити ширшу аудиторію.

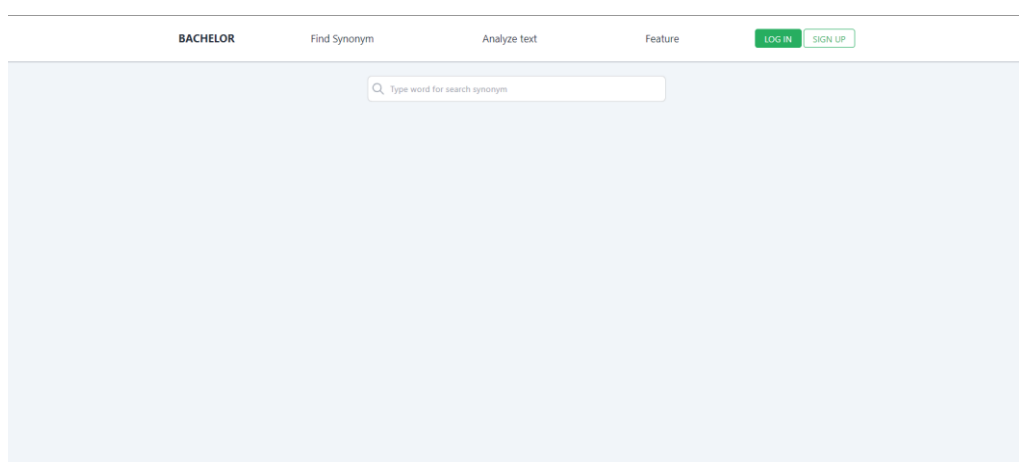


Рисунок 4.11 – Загальна сторінка клієнтського застосунку

В клієнтській архітектурі присутні декілька маршрутів, один з них – це пошук синонімів.



Ця сторінка містить фору, а саме кастомне поле для вводу, та реалізацію відправки форми після натискання клавіші Enter. Після відправки форми робиться запит до серверу з моделлю WE, на сервері з моделлю проводиться вибірка слів з семантичних зв'язків: wordnet та word2vec; після вибірки слів слова приводяться у стандартизований вигляд, а саме lowercase, та звіряється зі спільною базою даних.

Після з серверу на клієнт відсилається список слів синонімів. Після того, як клієнт обробив дані з серверу формується список зі слів синонімів, завдяки useState. Кожне слово в списку представлено у вигляді посилання на окремий шлях. Шлях з окремим словом представляє, сторінку для всіма характеристика цього слова: слово, значення, вид значення та приклади значення.

На рис. 4.12 представлено поле вводу для пошуку синонімів.

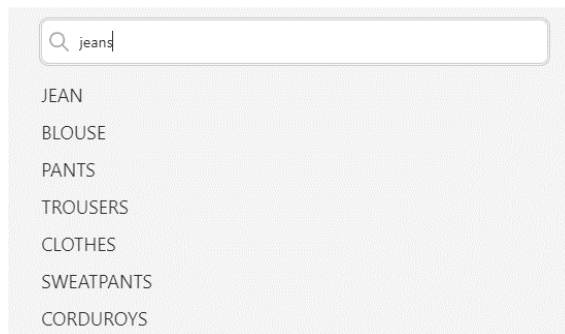


Рисунок 4.12 – Поле вводу та список синонімів

Окрема сторінка розташовується на маршруті, після рендеру сторінку зчитується URL з нього вирізається саме слово та робиться запит на сервер, сервіс на серверній частині оперує з БД.

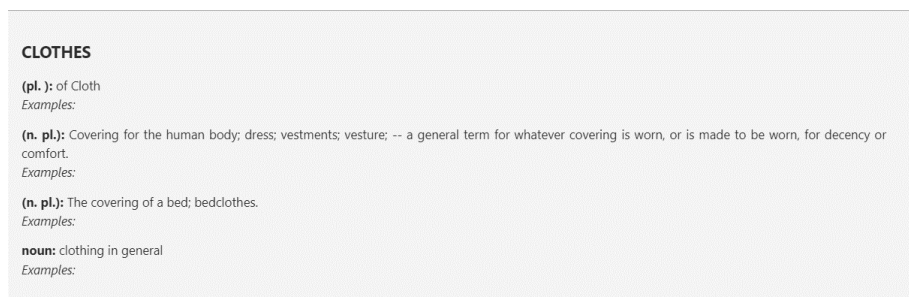


Рисунок 4.13 – Характеристика слова

На рис. 4.13 представлено вигляд сторінки з характеристиками слова.

У вебзастосунку, також наявна функція аналізу тексту на синоніми. Відкривається сторінка з формою, з полем для введення `textarea` та кнопкою, в контрольовані та валідації цієї форми допомагає бібліотека `formik`. В цю форму користувач, може вписати довільне речення, яке відповідає валідації для аналізу слів цього тексту потрібно натиснути кнопку `analyze text`.

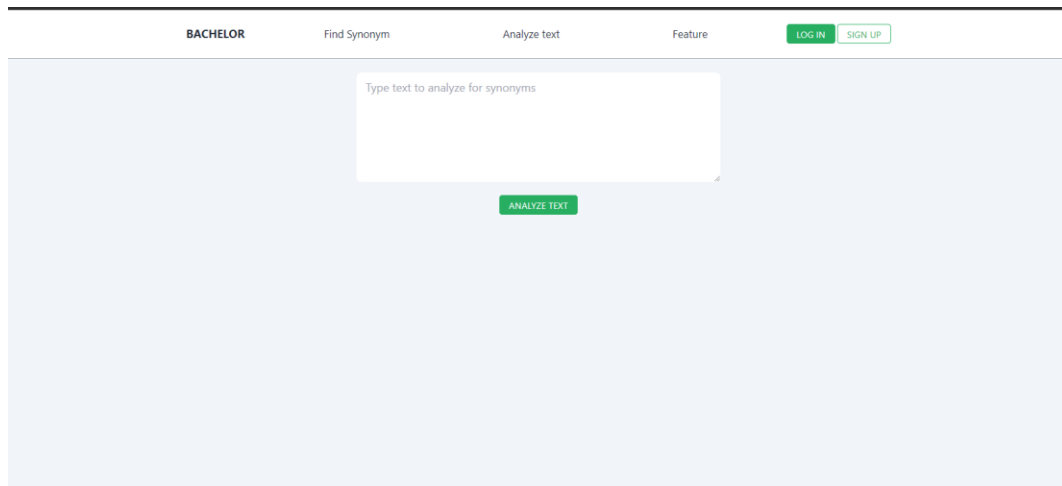


Рисунок 4.14 – Сторінка аналізу тексту

Після того, як користувач ввів речення та відправив форму, форма блокується.

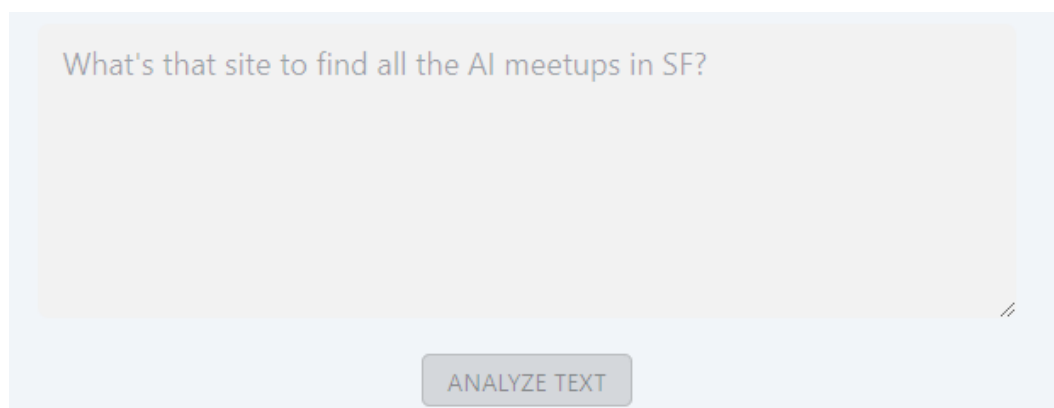


Рисунок 4.15 – Вигляд заблокованої сторінки

Після виправки форми, робиться запит на сервер з моделлю, там тексту оброблюється завдяки `data processing`. Для обробки та чистки тексту використовуються декілька методів, а саме:

- видалення спеціальних символів;

- видалення імен та назв;
- видалення stopwords слів;
- видалення емодзі;
- токенізація речення у набір слів;
- у деяких випадках лематизація слів.

Текст після обробки, відправляється на процес знаходження синонімів. Після знаходження синонімів, всі підібрані синоніми відправляються на клієнт. Клієнт оброблює цей запит, отримує дані та рендерить текст, який був виконаний

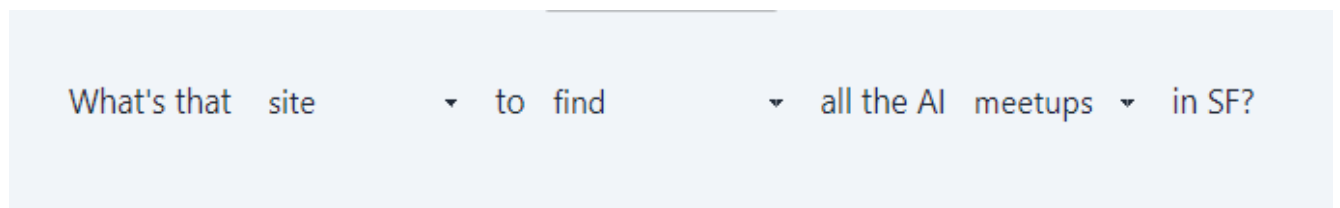


Рисунок 4.16 – Вигляд тексту з синонімами

На рис. 4.16 зображено вигляд тексту з знаходженням синонімів. Для рендеру синонімів, використано select, як перший атрибут, використано оригінальне слово, всі інші в атрибуті option. Використання цього атрибуту є дуже вдалим рішенням: простим та ефективним.

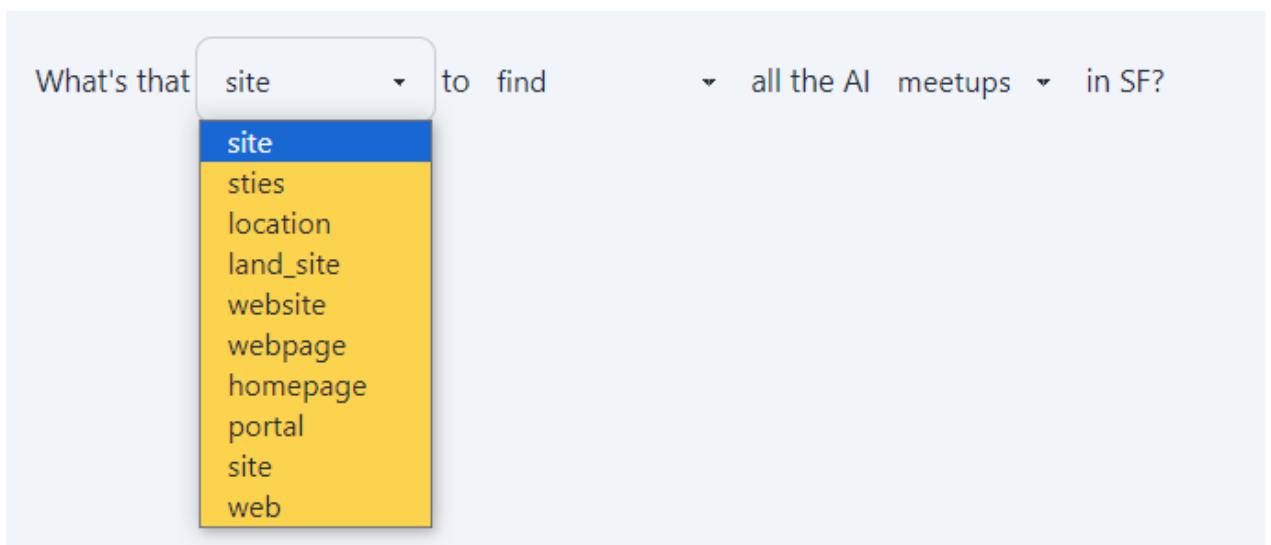
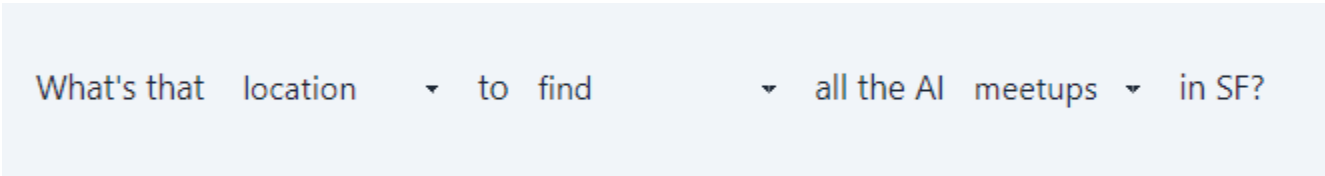


Рисунок 4.17 – Вигляд атрибуту option

На рис 4.17, зображено атрибути option.

Атрибути `option`, дозволяють вибрати синонім до оригінального слова. Після вибору слова можна буде побачити, як слово виглядає з синонімом.



What's that location ▾ to find ▾ all the AI meetups ▾ in SF?

Рисунок 4.16 – Приклад вибору іншого слова

На рис 4.17 зображено, вигляд тексту з вибором іншого слова, в цьому випадку `location`.

#### **Висновки до розділу 4**

В четвертому розділі кваліфікаційної роботи наведено програмну реалізацію функціоналу вебзастосунка для вивчення іноземних мов.

Описано процеси роботи з WE моделлю та методи оптимізації моделі для збільшення швидкості роботи вебзастосунку. Проведена робота над об'єднанням 2 баз даних в єдину базу даних, проведено форматування всіх значень бази даних та створення сутностей для подальшої роботи над вебзастосунком.

Продемонстровано створення користувацьких інтерфейсів для функцій вебзастосунка. Описано взаємодію користувацьких інтерфейсів з серверною частиною, та висвітлено процес роботи вебзастосунка.

## ВИСНОВКИ

Для досягнення поставленої мети було вирішено наступні завдання::

- на основі проведеного аналізу аналогів програмних засобів вивчення іноземних мов, визначені їх переваги та недоліки, сформульовано специфікацію вимог до вебзастосунку вивчення іноземних мов;
- на основі проведеного аналізу підходів та інструментів штучного інтелекту, що активно використовуються під час вивчення іноземних мов, визначено перелік методів, які були реалізовані у вебзастосунку;
- описано процес моделювання та проектування ПЗ вебзастосунку;
- розроблено front-end частину вебзастосунку на базі технологій React, Tailwind CSS;
- розроблено back-end частину вебзастосунку на базі технологій NodeJS, Python, PostgreSQL;
- проведено апробацію та тестування вебзастосунку.

Проведено аналіз застосунків-аналогів для вивчення ІМ, проведено аналіз сильних та слабких сторін у конкурентів, також спільні риси що мають їхні застосунки. Розроблено специфікацію вимог до ПЗ.

Проведено аналіз методів штучного інтелекту, які використовуються у подібних застосунках та зроблено докладний аналіз методів, які були використані у процесі розробки цього програмного забезпечення.

Описано процес моделювання та проектування вебзастосунку та наведено програмну реалізацію функціоналу вебзастосунку для вивчення іноземних мов.

Практичне значення отриманих результатів полягає у тому що застосування методів штучного інтелекту збільшує ефективність вивчення, іноземних мов, шляхом автоматизації, деяких процесів. Застосування штучного інтелекту для знаходження схожих слів, допомагає у вивченні словникового запасу іноземної мови. Розроблений застосунок може бути вдосконалений за допомогою розпізнавання контексту використання слів.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Статистика ринку вивчення іноземних мов. URL: <https://techreport.com/statistics/language-learning-market-statistics/> (Last accessed: 29.04.2024).
2. What is AI? URL: <https://www.ibm.com/topics/artificial-intelligence> (Last accessed: 29.04.2024).
3. Baker, T., & Smith, L. (2019). Educ-AI-tion rebooted? Exploring the future of artificial intelligence in schools and colleges. Retrieved from Nesta Foundation URL: [https://media.nesta.org.uk/documents/Future\\_of\\_AI\\_and\\_education\\_v5\\_WEB.pdf](https://media.nesta.org.uk/documents/Future_of_AI_and_education_v5_WEB.pdf)
4. Pokrivcakova, Silvia. (2019). Preparing teachers for the application of AI-powered technologies in foreign language education. Journal of Language and Cultural Education. 7. 135-153. 10.2478/jolace-2019-0025.
5. Застосунок Duolingo. URL: <https://www.duolingo.com/> (Last accessed: 01.05.2024).
6. Застосунок Praktika. URL: <https://praktika.ai/> (Last accessed: 01.05.2024).
7. Застосунок WordUp. URL: <https://www.wordupapp.co/> (Last accessed: 01.05.2024).
8. Open-source дата база англійських слів EDMTDev. URL: <https://github.com/eddydn/DictionaryDatabase> (Last accessed: 01.05.2024).
9. Open-source дата база англійських слів WordNet. URL: <https://wordnet.princeton.edu/> (Last accessed: 01.05.2024).
10. Опис діаграми діяльності. URL: <http://www.tsatu.edu.ua/kn/wp-content/uploads/sites/16/laboratorna-robota-78-diahrama-dijalnosti.pdf> (Last accessed: 03.05.2024).
11. Опис діаграми розгортання. URL: <https://www.guru99.com/uk/deployment-diagram-uml-example.html> (Last accessed: 04.05.2024).
12. Опис діаграми послідовності. [https://duikt.edu.ua/ua/news-1-626-7897-zastosuvannya-uml-chastina-2-diagrama-poslidovnosti---sequence-diagram\\_kafedra-kompyuternih-nauk-ta-informaciynih-tehnologiy](https://duikt.edu.ua/ua/news-1-626-7897-zastosuvannya-uml-chastina-2-diagrama-poslidovnosti---sequence-diagram_kafedra-kompyuternih-nauk-ta-informaciynih-tehnologiy) (Last accessed: 04.05.2024).

13. Модифікована версія WordNet – Wordset by Wordset, Inc.. URL: <https://github.com/wordset/wordset-dictionary?tab=License-1-ov-file#readme> (Last accessed: 10.05.2024).
14. What is NLP? URL: <https://www.ibm.com/topics/natural-language-processing> (Last accessed: 25.05.2024).
15. Martin Kleppmann. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media, 2017.