

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

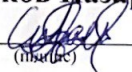
**РОЗРОБКА ТЕРМІНОЛОГІЧНОГО ІНШОМОВНОГО
ОНЛАЙН СЛОВНИКА З ВИКОРИСТАННЯМ PYPDF2**

**THE DEVELOPMENT OF THE TERMINOLOGICAL FOREIGN
LANGUAGE ONLINE DICTIONARY WITH A USAGE OF
PYPDF2**

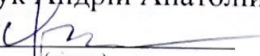
спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ІПЗс-21
Старіков Назарій Валерійович


(підпис)

Керівник:
к.т.н., доцент
Ящук Андрій Анатолійович


(підпис)

Кваліфікаційну роботу
допущено до захисту
« 8 » 06 2024 р.
к.т.н., доцент
Гарант освітньої програми:
Ліщина Наталія Миколаївна

(підпис)

Луцьк – 2024

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 121 Інженерія програмного забезпечення

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

М.А. Микитюк
« 2 » 02 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Старікову Назарію Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: розробка термінологічного іншомовного онлайн словника з використанням PyPDF2

Керівник роботи: к.т.н., доцент, Яцук Андрій Анатолійович

затверджені наказом закладу вищої освіти від «30» грудня 2023 р. № 477/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «5» червня 2024 р.

3. Вихідні дані до роботи: технічне та програмне забезпечення ЕОМ, вимоги до розробки програмного забезпечення, ергономічні вимоги до функціонування програмного засобу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):

Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення, опис функціонального наповнення об'єкта проектування, практична реалізація об'єкта проектування.

5. Перелік графічного матеріалу: 11 рисунків.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Яцук А. А.		
Специфікації вимог до розробленої системи	Яцук А. А.		
Розробка об'єкта проектування	Яцук А. А.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		4,2 %	
Академічна доброчесність	Яцук А. А.		

7. Дата видачі завдання «2» лютого 2024 р.

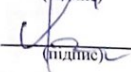
КАЛЕНДАРНИЙ ПЛАН

№ 3/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	07.02.2024 р.	вик.
2	Аналіз проблеми розробки та впровадження об'єкту проектування	27.02.2024 р.	вик.
3	Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання	12.03.2024 р.	вик.
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	17.04.2024 р.	вик.
5	Практична реалізація об'єкта проектування та розробка бази даних	18.05.2024 р.	вик.
6	Тестування та налагодження об'єкта проектування	01.06.2024 р.	вик.
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру	05.06.2024 р.	вик.

Здобувач вищої освіти

Керівник кваліфікаційної роботи


(підпис)


(підпис)

Старіков Н. В.
(прізвище, ініціали)

Яцук А. А.
(прізвище, ініціали)

**Рецензія
на кваліфікаційну роботу**

Здобувач вищої освіти: *Старіков Назарій Валерійович*

Тема: *Розробка термінологічного іншомовного онлайн-словника з використанням PyPDF2.*

Коротка характеристика кваліфікаційної роботи та прийнятих рішень:

Кваліфікаційна робота бакалавра складається з трьох розділів, в яких проаналізовані проблематика та поставлені завдання за темою роботи, здійснено теоретичне дослідження та практичну реалізацію для реалізації процесів електронної комерції, здійснено тестування розробленої системи

Самостійні розробки і пропозиції автора: *Автор самостійно розробив термінологічний іншомовний онлайн-словник з використанням PyPDF2*

Практичне значення роботи *полягає в тому, що використовуючи PyPDF з PDF файлу отримано програмний засіб: термінологічний іншомовний онлайн-словник зі зручною функцією пошуку і перегляду термінів.*

Недоліки: *Істотних недоліків в роботі не виявлено.*

Загальний висновок: *У кваліфікаційній роботі бакалавра здійснено аналіз предметної області, вибір методів і засобів для розробки, розроблено програмний продукт, здійснено його тестування. Усі поставлені завдання виконано в повному обсязі.*

Кваліфікаційна робота здобувача освіти Старікова Назарія Валерійовича рекомендується до захисту експертній комісії та заслуговує позитивної оцінки.

Рецензент кваліфікаційної роботи: Саварин Павло Вікторович, к.п.н., доцент кафедри ЦОТ


(підпис)

«07» червня 2024 р.

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

ВІДГУК
КЕРІВНИКА НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Здобувач вищої освіти Старіков Назарій Валерійович
(прізвище, ім'я, по батькові)

група ІІЗс-21

Тема кваліфікаційної роботи: РОЗРОБКА ТЕРМІНОЛОГІЧНОГО ІНШОМОВНОГО
ОНЛАЙН СЛОВНИКА З ВИКОРИСТАННЯМ PUPDF2.

Керівник к.т.н., доц. Ящук А. А.

Актуальність теми. В умовах сучасного інформаційного суспільства знання іноземних мов стає необхідним інструментом для професійного та особистого розвитку. Одним з найважливіших аспектів вивчення мови є оволодіння спеціалізованою термінологією, яка використовується в різних галузях науки, техніки, бізнесу та інших сферах. В епоху цифрових технологій та Інтернету створення онлайн-ресурсів, які полегшують процес вивчення іноземної мови, набуває особливої актуальності.

Об'єкт дослідження – використання технологій парсингу та розробки веб сайтів.

Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи. Було проведено аналіз предметної області, визначено виоги до програмного продукту, здійснено парсинг PDF-файлу, що містить іншомовні терміни та їх визначення зі створенням бази даних. Розроблено онлайн словник іншомовних термінів. Здійснено тестування та налагодження системи.

Загальний висновок

Роботу виконано у повному обсязі у відповідності до поставлених завдань, а робота здобувача Старікова Н. В. може бути оцінена на позитивну оцінку.

Керівник 
(підпис)

Ящук А. А.
(прізвище, ім'я, по батькові)

«07» 06 2024 р.

АНОТАЦІЯ

Старіков Н. В. Розробка термінологічного іншомовного онлайн словника з використанням PyPDF2. Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2024.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків і пропозицій, списку використаних джерел та додатків.

У першому розділі здійснено аналіз сучасного стану проблеми розробки термінологічних словників і сформульовано завдання на проект. В другому розділі визначено вимоги до розроблюваної системи, а також засоби, методи і алгоритми розробки. У третьому розділі розроблена методологія правильного порядку реалізації продукту від початкової стадії розробки до фінальної. Саме в третьому розділі проаналізовано основні методи та технології, що використовуються для створення електронних словників, зокрема засоби обробки PDF-документів. Особлива увага приділяється бібліотеці PyPDF2, її можливостям та обмеженням. Розглянуто процеси парсингу PDF-файлів, вилучення текстової інформації та її подальша обробки для формування термінологічної бази даних. Результати розробки демонструють можливості розробки відповідних термінологічних онлайн словників. У висновках узагальнено інформацію, відображену у попередніх частинах.

Ключові слова: термінологічний словник, PyPDF2, PDF-документи, блог, база даних.

ABSTRACT

Starikov N. V. The development of the terminological foreign language online dictionary with a usage of PyPDF2. Manuscript.

Bachelor's qualifying thesis of the educational program "Software Engineering". Lutsk National Technical University. Lutsk, 2024.

The bachelor's qualifying thesis consists of an introduction, three sections, conclusions and proposals, a list of used sources and annexes.

In the first chapter, an analysis of the modern state of the problem of terminological dictionary development was carried out and the objective for the project was formulated. In the second chapter, the requirements for the developing system were defined as the means, methods and algorithms. In the third chapter, the methodology of correct product implementation order from the initial stage to the final one was developed. Namely in the third chapter, the basic methods and technologies were analyzed which were used for creating electronic dictionaries, specifically processing means of PDF documents. The special attention was paid for the library PyPDF2, its opportunities and restrictions. The parsing processes of PDF files, extraction of text information and its subsequent processing for forming the terminological database were reviewed. The development outcomes demonstrate the possibilities of the development of the appropriate terminological online dictionaries. In the conclusions the information was summarized and displayed in the previous parts.

Keywords: terminological dictionary, PyPDF2, PDF documents, blog, database.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ РОЗРОБКИ ТЕРМІНОЛОГІЧНОГО ІНШОМОВНОГО ОНЛАЙН СЛОВНИКА З ВИКОРИСТАННЯМ RUPDF2 І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра.....	12
Висновки до розділу 1	13
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ.....	15
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	15
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	19
Висновки до розділу 2	22
РОЗДІЛ 3 РОЗРОБКА ТЕРМІНОЛОГІЧНОГО ІНШОМОВНОГО ОНЛАЙН СЛОВНИКА З ВИКОРИСТАННЯМ RUPDF2	24
3.1 Практична реалізація об'єкта проектування	24
3.2 Тестування та налагодження інформаційно-комп'ютерної системи	29
3.3 Розробка бази даних.....	30
3.4 Захист інформаційно-комп'ютерної системи	32
Висновки до розділу 3	33
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	36

ВСТУП

Актуальність теми. Розробка термінологічних онлайн словників – це давня тема, яка бере своє коріння ще з початку поступового розповсюдження мережі Інтернет, коли якість апаратного та програмного забезпечення була відносно низькою. Необхідність вивчати мови, терміни та визначення – це базова необхідність в освіті.

В умовах сучасного інформаційного суспільства знання іноземних мов стає необхідним інструментом для професійного та особистого розвитку. Одним з найважливіших аспектів вивчення мови є оволодіння спеціалізованою термінологією, яка використовується в різних галузях науки, техніки, бізнесу та інших сферах. В епоху цифрових технологій та Інтернету створення онлайн-ресурсів, які полегшують процес вивчення іноземної мови, набуває особливої актуальності.

Дана кваліфікаційна робота бакалавра присвячена розробці термінологічного іншомовного онлайн словника з використанням бібліотеки PyPDF2. PyPDF2 – це потужний інструмент для роботи з PDF-документами, який дозволяє здійснювати обробку та аналіз текстових даних, що містяться в PDF-файлах.

Мета роботи полягає у створенні зручного та функціонального інструменту, який дозволить користувачам швидко знаходити значення необхідних термінів.

Завдання до кваліфікаційної роботи – створити термінологічний іншомовний онлайн словник.

Об'єктом кваліфікаційної роботи бакалавра є використання технологій парсингу та розробки веб сайтів.

Предметом кваліфікаційної роботи бакалавра є розробка та реалізація онлайн-інструменту для створення словника з іншомовних джерел.

Практична цінність розробки полягає у забезпеченні людей можливістю переглядати визначення до термінів за допомогою онлайн джерел в навчальних цілях.

Перелік цілей, які були намічені на хід виконання даного програмного продукту:

- здійснити аналіз предметної області та методології розробки термінологічного іншомовного онлайн словника;
- ознайомитись з основами роботи з консольними проектами на мові програмування Python з використанням різних бібліотек та отримати досвід пошуку відповідної технології, а також ознайомитись з технологіями вилучення інформації з PDF файлів за допомогою мови програмування Python;
- реалізувати простий блог на сайті програмного продукту, де буде наявний словник;
- здійснити тестування і налагодження даного проекту;
- забезпечити захист інформаційно-комп'ютерної системи.

РОЗДІЛ 1

АНАЛІЗ РОЗРОБКИ ТЕРМІНОЛОГІЧНОГО ІНШОМОВНОГО ОНЛАЙН СЛОВНИКА З ВИКОРИСТАННЯМ RUPDF2 І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

Термінологічний іншомовний онлайн словник – це словник реалізований у якості веб сайту або мобільного додатку, інформація якого доступна за умов підключення пристрою до мережі Інтернет, та який несе в собі дві словникових функції, а саме: перекладну та термінологічну. Відповідно, перекладна функція включає в себе переклад з однієї мови на іншу, а термінологічна включає в себе надання визначень поняттям та термінам.

Як відомо, вивчення іноземних мов є достатньо складним процесом. У певних випадках, люди вивчають мови заради професійної мети. Різні професії вимагають знання великої кількості термінів. З метою поліпшення результатів освіти в певних областях протягом історії розповсюдження Інтернету створювалися різноманітні довідники, посібники, словники тощо. Вони в свою чергу доповнюються іншими сервісами такими як форуми, енциклопедії, тощо.

Наприклад, сайт показаний в джерелі [1] виконує функцію посібника та функцію словника термінів. Надзвичайно мала кількість термінів не може бути перевагою, а скоріше навпаки недоліком. З іншого боку, існує безліч додаткових функцій, сукупність яких створює відповідність назві виду веб сайту його справжньому призначенню. Даний сайт є юридичним порталом, де можна знайти інформацію з різних кодексів та коментарів до них.

Призначення даного веб сайту та його загальний вигляд зображені на рисунку 1.1. З нього виходить, що веб сайт має ряд переваг та недоліків, серед яких перевага інформативності та зручності використання та недолік загального дизайну веб сторінки.

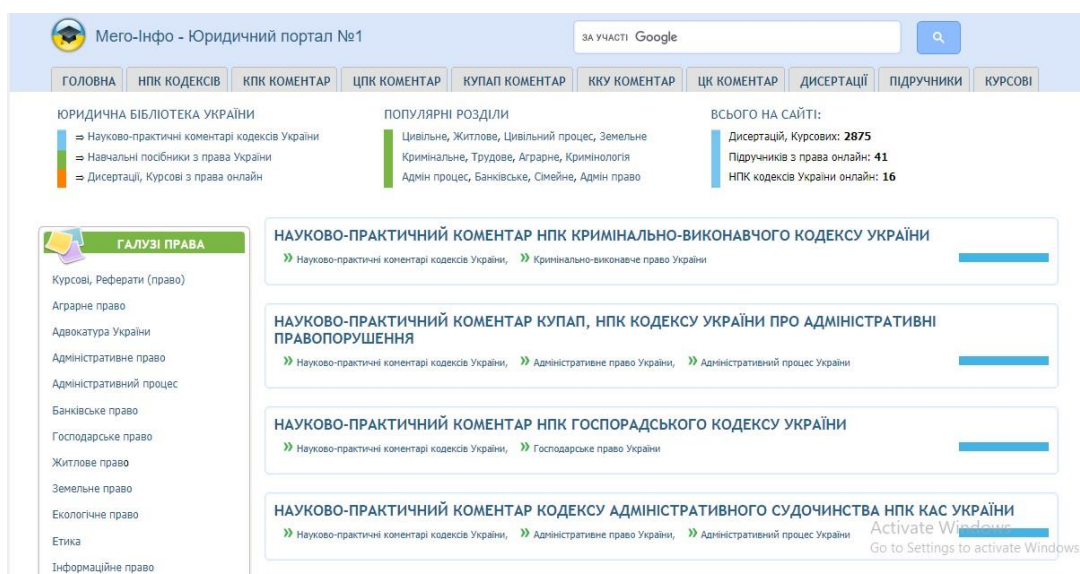


Рисунок 1.1 – Веб сайт юридичного порталу

Веб сайт з джерела [2] є військовим перекладним двомовним словником без ознак термінологічного словника. Тобто терміни, що записані двома мовами, не мають відповідних визначень. Однак, кількість термінів загалом перевищує за 10000. Саме тому, це гідний конкурент, чий контент гідно використовувати для парсингу за явної потреби. Загальний дизайн сайту зображений на рисунку 1.2.



Рисунок 1.2 – Військовий перекладний словник

Ще одним прикладом юридичного порталу є веб сайт джерела [3], що виконує ту ж функцію та має термінологічний словник виключно українською

мовою. Кількість термінів значно більша, що є значно серйознішим конкурентом для програмного продукту даної кваліфікаційної роботи бакалавра.

Відповідно, у ході поліпшення якості ПЗ словники та довідники стають недостатньою мірою для ефективного навчання. Однак, попри наявність нейронних мереж варто визнати, що результати роботи нейронних мереж з метою надання консультації користувачеві у відповідному чат боті на кшталт ChatGPT можливе лише при наявності великої кількості інформації з великої кількості джерел. Найбільш прибутковим періодом розвитку словників та довідників був початок розповсюдження Інтернету на межі XX та XXI сторічч. Однак, у випадку наявності великої кількості альтернативних джерел пізніше з'явилась можливість заміщати менш ефективні джерела отримання інформації на більш зручні та ефективні. Таким чином, станом на 2024 рік ChatGPT є основним конкурентом в освітній галузі та витісняє форуми, посібники, словники, енциклопедії, тощо.

Проблематика розвитку сучасного проекту кваліфікаційної роботи бакалавра переважно стосується необхідності показати свої найліпші уміння в умовах ресурсних обмежень таких як час, фінанси, недоліків, які стосуються обмеженого досвіду у порівнянні з професіоналами з багаторічним стажем, а також переваг таких як поліпшені умови пошуку інформації та її вивчення. Однак, як відомо, найбільш амбітні ідеї проектів переймаються великими компаніями, які володіють великою кількістю трудового ресурсу, що примножує загальну суму використаного на роботу часу, а також загальний досвід використаний працівниками підприємства. Виконання власного проекту наодинці завжди матиме певні обмеження.

Аналізуючи вимоги на ринку варто зрозуміти, що станом на період написання даної кваліфікаційної роботи бакалавра нейронні мережі обмежуються лише тією інформацією, що є в загальному доступі на інших веб сайтах. Аналізуючи веб сайти аналогічного призначення виявляється, що кількість термінологічних словників з перекладацькими можливостями надзвичайно малі, а кількість термінів та визначень наявних на сайті неймовірно низька. Поспілкувавшись з нейронними мережами за декілька місяців до здачі

проекту кваліфікаційної роботи бакалавра було визначено, що нейронна мережа ChatGPT не завжди готова надавати точні визначення та переклади до певних термінів, які мали певне професійне значення. Однак, по мірі набуття певних нових джерел інформації організація виробник ChatGPT та виробники інших нейронних мереж будуть удосконалювати можливості нейронних мереж стосовно надання необхідних термінів та визначень.

Існує велика кількість професій, частина з яких потребує знання англійської мови, а також включає в себе велику кількість термінів та визначення до них. Відповідно, терміни можуть різнитися залежно від роду діяльності. Саме тому, даний програмний продукт має користь, адже відповідно до ідеї зберігає терміни та визначення обома мовами з урахуванням необхідності у правильності перекладу.

Успіх даного проекту може бути реалізованим, у значній мірі, завдяки кількості термінів та їхньої цінності для вивчення певного роду діяльності.

Джерела до пояснювальної записки даної кваліфікаційної роботи бакалавра згадуються відповідно до необхідності при описі процесів розробки, але можуть бути перерахованими зберігаючи відповідність джерел до переліку цілей, які повинні бути досягнуті в ході виконання даної кваліфікаційної роботи бакалавра.

Відповідно до уявлення про проект та технологічні вимоги до його реалізації, реалізація проекту складається зі створення веб сайту та заповнення бази даних відповідними відомостями.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Завдання на кваліфікаційну роботу бакалавра:

- здійснити аналіз предметної області та методології розробки термінологічного іншомовного онлайн словника;
- ознайомитись з основами роботи з консольними проектами на мові програмування Python з використанням різних бібліотек та отримати досвід

пошуку відповідної технології, а також ознайомитись з технологіями вилучення інформації з PDF файлів за допомогою мови програмування Python;

- реалізувати простий блог на сайті програмного продукту, де буде наявний словник;
- здійснити тестування і налагодження даного проекту;
- забезпечити захист інформаційно-комп'ютерної системи.

Висновки до розділу 1

Таким чином, описавши джерела, цілі даної кваліфікаційної роботи бакалавра, аналоги до даного програмного продукту та ідею стосовно розробки даного програмного продукту можна впевнено заявляти, що даний програмний продукт є реалізовуваний та в певній мірі корисний на ринку.

Проведений аналіз сучасного стану проблеми розробки іншомовних термінологічних онлайн словників свідчить про значний потенціал і необхідність створення подібних програмних продуктів. Вивчення іноземних мов, особливо в контексті професійної діяльності, вимагає не лише базових знань мови, але й глибокого розуміння термінології, специфічної для певної галузі. Враховуючи сучасні технологічні можливості, створення онлайн словників з перекладами та визначеннями термінів може суттєво підвищити ефективність навчання і професійної діяльності.

Значна частина сучасних онлайн словників не надає комплексної інформації, що включає як переклад, так і визначення термінів. Це створює певні труднощі для користувачів, які потребують детального розуміння термінології. Саме тому, проектування і реалізація багатомовного термінологічного онлайн словника, що включає обидві функції – перекладну і термінологічну – є важливим кроком у розвитку освітніх інструментів.

Розробка такого програмного продукту вимагає оволодіння різними технологіями і інструментами, такими як Python для парсингу і роботи з PDF-файлами, використання frontend і backend фреймворків для створення веб-сайту,

а також робота з базами даних за допомогою MySQL. Всі ці технологічні аспекти є ключовими для успішного виконання проекту і забезпечення його функціональності. Саме виконання таких цілей ідуть пліч-о-пліч з навчальними цілями даної кваліфікаційної роботи бакалавра.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Термінологічний іншомовний онлайн словник – це досить простий програмний продукт, а його користувачі характеризуються достатньо простими вимогами та потребами. У загальних рисах, при введенні вхідних даних, що являють собою терміни потрібно вивести визначення на відповідній мові у якості вихідних даних. Це дуже проста задумка, адже в минулому існували веб сайти, які буквально складалися з однієї або двох сторінок та очевидно виконували ті самі функції. Однак, потрібно не забувати, що веб сайт може являти собою довідкову інформаційну систему. Саме тому, є перспектива розвитку програмних продуктів, які суміщатимуть функціонал блогу та словника, що загалом можливо з використанням певних фреймворків.

Забезпечення безпеки системи є ключовим аспектом розробки, тому необхідно впровадити механізми аутентифікації та авторизації користувачів.

Таким чином, цілі користувача залежать від багатьох факторів включаючи: вимоги до програмних продуктів зроблених конкурентами та загальні проблеми людей, які користуватимуться продукцією. На мою суб'єктивну думку, починати треба з проблематики використання онлайн словників.

Успішний термінологічний онлайн словник відповідає таким вимогам як надійність та зручність. Під надійністю потрібно розуміти, що словник надає тільки правильні визначення на існуючі терміни. Під зручністю потрібно розуміти, що онлайн словник повинен швидко відповідати на запити та характеризуватися певними ознаками ергономіки.

Ергономіка – це наука, яка досліджує забезпечення технічних систем комфортними умовами використання. Станом на сьогоднішній день, на мою думку, ніяких проблем з ергономікою при виробництві простих веб сайтів бути не може, адже ергономіка виражається виключно в UX/UI дизайні, який в

більшості веб сайтів більш менш однаковий за якістю та ціною реалізації. Вивчення HTML5 та SCSS на базовому рівні та наявність уявлення про вибір шрифтів та кольорів є достатнім. Дуже рідко компанії наймають спеціалістів з UX/UI дизайну, адже попит в таких спеціалістів вузько обмежений. Веб сайти, які потребують ергономічного підходу є продукцією великих корпорацій, як наприклад веб сайт «YouTube», де станом на 2024 рік є анімація натиснення на вподобайку. Тим самим мається на увазі, що випадки складних ергономічних розрахунків стосуються тільки великих програмних продуктів, які статистично використовуються часто, що є абсолютною протилежністю мого програмного продукту, який використовується відносно рідко, як більшість веб сайтів в Інтернеті. Також причиною нехтуванням UX/UI дизайном є довгий період навчання даному навику, що підтверджує відео з джерела [4].

Швидкість виконання запитів онлайн словника, що є також ергономічним фактором, вираховується тривалістю виконання запитів у БД та тривалістю виконання програмного коду веб сайту. Станом на сьогодні, Інтернет поліпшується як в доступності, так і в швидкості. Фактор швидкості передачі Інтернет сигналу, а також фактор обмеженості даних словника вказують на те, що продукт витримує вимоги до швидкодії виконання пошукового запиту, але це залежить від кількості даних, які можуть бути наявні в словнику.

Виходячи з попереднього твердження про інформаційну довідкову систему, можна затвердити, що до даного проекту мають бути приєднані функції звичайного блогу. Мета блогу – оповіщати про ті чи інші зміни в галузях, по яким ведеться термінологічний іншомовний словник. Тому даний словник дуже необхідний для певних галузей економіки, список яких залежить від якостей іншомовного термінологічного онлайн словника.

При проектуванні іншомовного термінологічного онлайн словника, необхідно врахувати специфічні вимоги і завдання, які будуть вирішуватися даною системою. Згідно статі джерела [5], Вибір моделі для моделювання програмної системи є ключовим етапом, оскільки він визначає структуру, поведінку та взаємодію компонентів системи. Серед основних моделей, які

можуть бути застосовані для моделювання програмного забезпечення, виділяються наступні: водоспадна модель (Waterfall), спіральна модель (Spiral), модель V-подібного процесу (V-Model) та гнучкі методології (Agile).

Водоспадна модель (Waterfall) – передбачає послідовне виконання етапів розробки програмного забезпечення, де кожен наступний етап починається лише після завершення попереднього. Ця модель добре підходить для проектів з чітко визначеними вимогами, які не змінюються протягом розробки. Однак, для нашого проекту, де можливі зміни та уточнення вимог на різних етапах, водоспадна модель не є оптимальною.

Спіральна модель (Spiral) – поєднує в собі елементи як водоспадної моделі, так і гнучких методологій, акцентуючи увагу на ітеративному підході. Ця модель дозволяє проводити аналіз ризиків і постійне вдосконалення проекту через повторювані цикли. Спіральна модель підходить для великих і складних проектів, де важливий детальний аналіз і контроль ризиків. Проте, ця модель може бути занадто складною для менш масштабного проекту, такого як створення онлайн словника.

Модель V-подібного процесу (V-Model) – модель V-подібного процесу є розширенням водоспадної моделі, де кожен етап розробки має відповідний етап тестування. Ця модель забезпечує високий рівень якості програмного забезпечення, проте, як і водоспадна модель, вона менш гнучка до змін вимог.

Гнучкі методології (Agile) – гнучкі методології, такі як Scrum, Kanban, дозволяють швидко адаптуватися до змін вимог та забезпечують регулярну доставку робочого продукту. Agile підходи передбачають ітеративну розробку з частими релізами, що дозволяє збирати зворотний зв'язок від користувачів і вносити необхідні корективи. Для нашого проекту, де важливо швидко реагувати на зміни і забезпечити постійне вдосконалення функціональності, гнучкі методології є найбільш підходящими.

Моїм рішенням як розробника даного проекту стало притримування водоспадної моделі розробки.

Функціональні вимоги до програмного продукту.

Перш за все, існуватимуть такі сторінки:

- сторінка пошуку;
- сторінка блогу;
- сторінка про сервіс;
- сторінка реєстрації;
- сторінка входу;
- сторінка створення посту;
- сторінка створення категорії посту;
- сторінка списку категорій;
- сторінка списку постів.

Серед списку даних сторінок є сторінка пошуку, де користувач вводить предмет пошуку, а веб сайт показує визначення на даний термін. Сторінка блогу призначена для пошуку та подальшого перегляду статей створених зареєстрованими юзерами. Сторінка про сервіс носить скоріше інформативний характер та оповіщає в про сам сайт та його функцію. Для реєстрації та входу є відповідні дві сторінки.

Робота з постами реалізовується опираючись на концепцію класифікації всіх постів за категоріями. Саме тому є певний список категорій, які можна редагувати та видаляти аналогічно до списку статей, який може піддаватися тим же змінам. Однак створення категорії та створення посту – це окремі сторінки.

При створенні нового посту вибираються одне зображення, одна назва, одна категорія, а також вписується один текстовий масив, що може складатися з безлічі абзаців. Відповідно, всі сторінки поділяються на ті, що доступні всім користувачам і ті, що доступні зареєстрованим. Це надає можливість розширити функції даного веб сайту.

У загальних рисах, вимоги до даного програмного продукту дуже умовно відображено на рисунку 2.1, де відображений явний розділ функцій зареєстрованих та неавторизованих користувачів.

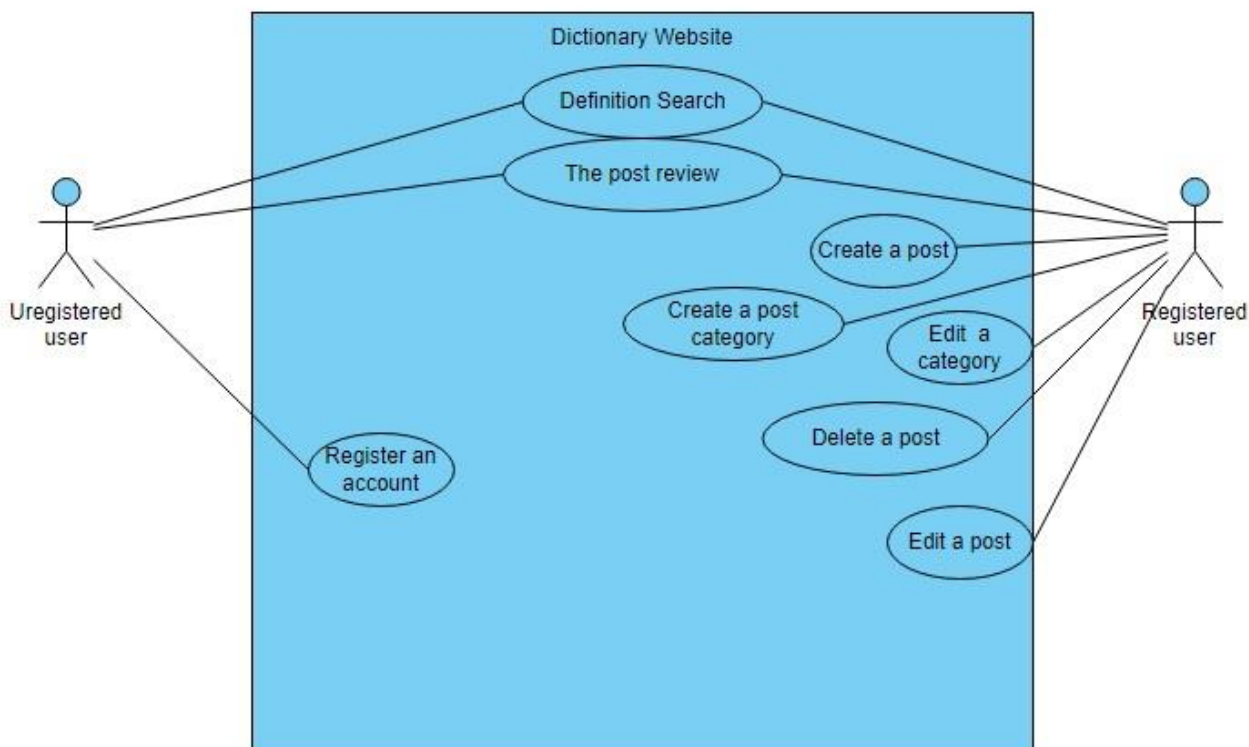


Рисунок 2.1 – Use Case діаграма стосовно даного програмного продукту

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Створення онлайн словника – це перш за все, щонайменше два проекти, куди входять веб сайт програма для заповнення бази даних необхідними даними, які в даному випадку є термінами та визначеннями.

Потрібно врахувати існування наступних компонентів: база даних, програма по заповненню бази даних, веб сайт. Неважко здогадатися, що мова запитів повинна задовольняти вимоги цінової політики, перш за все, також повинна бути достатньо розповсюдженою для широкої доступності застосування даної технології з іншими технологіями. Саме тому, було прийняте рішення створити MySQL базу даних, адже MySQL – це розповсюджена мова запитів для простих веб проектів. При використанні операційної системи Windows в якості локального веб сервера використовувався XAMPP.

При створенні блогів дуже часто використовують Laravel фреймворк. Створення сайту на CMS могло би здешевити виробництво, але не вирішити проблеми з базами даних. Проблема пов'язана з базою даних полягає у об'єднанні двох баз даних, де одна включає в себе дані необхідні для блоггової частини веб сайту, а інша – визначення та терміни. Саме тому, в якості backend фреймворку було прийняте рішення використати Laravel.

Laravel – це backend фреймворк, який базується на мові програмування PHP та являє собою допоміжним інструментом в побудові складних веб сайтів. Такими складними веб сайтами можуть бути блоги, інтернет-магазини, соціальні мережі, форуми, тощо. Базові навички, які можна набути при вивченні Laravel можна звести до CRUD (Create Read Update Delete) застосунку. Згідно зі статтею джерела [6] основою CRUD застосунку є запис нових даних, оновлення даних, зчитування даних, видалення даних. Відповідно, це все виконується при взаємодії з базою даних, адже PHP – це серверна мова.

В нормі, програмні комплекси певної складності складаються з класів, які маючи свої методи виконують ті чи інші функціональні вимоги. Однак, будь-який сайт має властивості CRUD застосунку, де класи в свою чергу виконують дещо подібні функції. Саме тому, згідно з джерелом [7], фреймворком Laravel представлена певна кількість видів класів включаючи моделі, міграції, контролери, тощо. Це спрощує процес роботи над веб сайтом роблячи процес розробки швидшим та простішим.

Однак, існують вимоги стосовно швидкодії роботи веб сайту. Так наприклад, попередній досвід інших професіоналів з галузі створення веб сайтів показує, що використання типу веб сайту відомого під назвою Single Page Application є більш доцільним, адже це пришвидшує виконання користувацьких запитів та операцій.

Згідно зі статтею джерела [8], Single Page Application – це веб сайт, який складається з одного HTML документа та може змінювати його вигляд та URL залежно від дій користувача та завдяки JS коду. В загальних рисах, це особливість frontend фреймворків. Frontend фреймворк завжди характеризується

ускладненими принципами верстки та структури проекту компенсуючи це поліпшенням швидкодії роботи сайту. У якості такого фреймворку у моєму проєкті виступає Vue.js, документація якого розміщена на веб сайті за посиланням джерела [9].

З метою наповнення бази даних необхідними даними потрібно було вибрати джерело отримання термінології та засіб її збереження в БД. Таким джерелом можуть бути PDF файли, інші веб сайти, тощо. У даному випадку, було прийняте рішення скористатися PDF файлом, де збережено чимало термінів англійською мовою та певні програмні засоби для читання PDF файлів з метою збереження тексту в БД.

Так як найбільше різноманітних бібліотек пропонується мовою програмування Python, було прийняте рішення використати мову програмування Python. Серед багаточисельного вибору бібліотек була вибрана PyPDF2, адже вона є в загальному та безплатному доступі.

Згідно з документації наведеної в джерелі [10], суть роботи бібліотеки PyPDF2 дуже проста. Весь текст PDF файлу зберігається в змінну без урахувань переносу на наступний рядок для подальшої обробки. Під поняттям весь текст мається на увазі абсолютно весь текст включаючи номери сторінок та інший текст, який не має відношення до термінів та визначень та який має бути очищений завдяки бібліотеки «Re» мови програмування Python. Основою будь-яких операцій над текстом є синтаксичні ознаки такі як дефіси, розділові знаки, тощо.

Алгоритм пошуку термінів починається з введення користувачем запиту в пошуковий рядок. Система перевіряє введений запит на коректність, після чого пошуковий запит надсилається до бази даних. Виконується пошук термінів у базі даних за введеним запитом, використовуючи індекси для підвищення швидкості пошуку. Результати пошуку сортуються за релевантністю, після чого вони відображаються на веб-сторінці. Алгоритм додавання нового терміну передбачає, що адміністратор вводить новий термін, його переклад та визначення у відповідні поля форми. Система перевіряє введені дані на коректність (наявність

обов'язкових полів, правильність форматів тощо), після чого новий термін додається до бази даних. Виводиться повідомлення про успішне додавання терміну. Алгоритм редагування терміну починається з того, що адміністратор обирає термін для редагування зі списку. Вносяться необхідні зміни до полів терміну, система перевіряє зміни на коректність, після чого змінені дані оновлюються у базі даних. Виводиться повідомлення про успішне редагування терміну.

Висновки до розділу 2

Отже з вище викладеного термінологічний іншомовний онлайн словник досить простий програмний продукт і може використовуватися користувачем на практиці. Існують всі необхідні засоби та технології розробки даного програмного продукту. Даний проект має навчальну та ринкову цінність.

Таким чином, цілі системи полягають у забезпеченні доступу до термінологічного словника, підвищенні ефективності навчання та розширенні бази термінів. Основна мета – надання користувачам можливості здійснювати пошук термінів, переклад та отримання визначень для професійного використання. Це сприяє користувачам у вивченні професійної термінології, надаючи детальні визначення та переклади. Система повинна постійно оновлювати та поповнювати базу даних новими термінами та визначеннями.

У цьому розділі було здійснено комплексний аналіз та визначення вимог до розроблюваного програмного забезпечення, а також проведено детальне проектування інформаційної системи. Вибір певного типу моделі для моделювання програмної системи та її елементів був обґрунтований на основі аналізу специфіки завдань, що стоять перед системою. Застосовано відповідні методи виявлення і аналізу вимог, що дозволило чітко визначити функціональні характеристики системи. Використання формалізованих методів синтезу моделей програмного забезпечення забезпечило структурований підхід до проектування.

Огляд існуючих інструментів, які можуть бути використані для вирішення задачі розробки програмного забезпечення, дозволив обрати найбільш підходящі засоби для створення інформаційної системи. Проведено детальний аналіз можливостей, переваг і недоліків цих інструментів. Було вибрано конкретні засоби для вирішення задач предметної області, наведено їх детальний опис та продемонстровано методику використання для вирішення поставлених задач.

РОЗДІЛ 3

РОЗРОБКА ТЕРМІНОЛОГІЧНОГО ІНШОМОВНОГО ОНЛАЙН СЛОВНИКА З ВИКОРИСТАННЯМ PYPDF2

3.1 Практична реалізація об'єкта проєктування

Процес розробки веб сайту починається з інсталяції Laravel 9 проєкту, Vue.js 3 та «vite». Далі необхідно видаливши весь контент сторінки welcome.blade.php підключити основний компонент відомий під назвою App.vue, де є доступ до всіх інших компонентів, які виникатимуть в ході розробки. Також змінюється вміст app.js файлу, який наявний в Laravel проєкті за замовчуванням. Повний код app.js файлу наведений в лістингу 3.1.

Лістинг 3.1 – Демонстрація файлу app.js

```
import './bootstrap';
import { createApp } from 'vue';
import App from '../js/App.vue'
import './css/app.css'
import router from './router'
import axios from 'axios';

axios.defaults.headers.common['X-CSRF-TOKEN'] =
document.querySelector('meta[name="csrf-token"]').getAttribute('content');

createApp(App).use(router).mount('#app')
```

Кінець лістингу 3.1

Наступним етапом є написання route файлу. За замовчуванням, всі шляхи до сторінок веб сайту прописуються у файлі web.php або у файлі api.php, що знаходяться в директорії .routes. Однак, при розробці застосунку на Vue.js логіка дещо інша, адже single page application – це веб сайт, де відповідно до ідеї є виключно один HTML документ або один blade.php документ. В той ж час, route файли розраховані посилатися на окремі php та html документи, що протирічить

задумці розвитку single page application. Саме, тому прийшлося скористатися vue-router. Код з використанням vue-router відтворений в додатку А.

Попри те, варто не забувати, що там зберігаються виключно ті route, які належать до групи get запитів. Тобто при будь-яких post, head, put або будь-яких інших запитах прийдеться користуватися route файлами запропонованими Laravel проектом. У даному випадку, таким файлом є api.php.

Розробивши route систему потрібно приступати до розробки самих Vue.js компонентів. Кожен Vue.js компонент представляє собою один файл формату .vue, який включає в себе верстку, стилі, та JavaScript код. В той час, як HTML та CSS – це дуже примітивні та всім відомі технології, я сконцентрував свою увагу на взаємодії HTML елементів з JavaScript скриптом.

Незалежно від кількості даних в БД або від кількості сторінок, всі CRUD застосунки працюють за рахунок збору вхідних даних та відправлення їх на серверну частину проекту в якості запиту та отримання відповіді з метою подальшого виведення на екран та можливої завчасної обробки вихідних даних. Найбільш очевидним та базовим рішенням є бібліотека мови програмування JavaScript під назвою «axios». Як видно з Vue.js компонентів проекту всі поля текстового вводу характеризуються схожими принципами взаємодії, де спочатку створюється набір змінних або масивів, яким присвоєно нульове значення та які змінюють його залежно від вмісту поля введення. Потім існують наступні шанси відбувається запит до відповідного PHP класу та функції керуючись даними з route-файлів, а потім реалізовується вивід вихідних даних в якості вмісту відповідних HTML елементів.

Використовуючи «axios» варто не забувати про серверну частину проекту. У даному випадку, це контролери. Контролер – це клас, методи та функції якого отримують дані від HTTP запитів з метою реалізації коректної HTTP відповіді. Історично існували мікро фреймворки, де контролер – це єдиний відокремлений вид класів для PHP розробки, а класи з іншими функціональними особливостями розроблялися окремо. Однак, сучасний backend фреймворк Laravel пропонує ширший спектр можливих класів написаних мовою PHP, а також можливість

створити свій вид класу зі своїми унікальними властивостями. Так наприклад, клас відомий в англійській мові під назвою «model» відповідає за підключення до бази даних, де один клас зчитує, редагує та видаляє вміст однієї таблиці залежно від нагальної потреби в ході розробки.

Таким чином, з метою знаходження даних у базі даних був написаний код контролера, який включає в себе посилання на три класи моделей.

Модель – це клас запропонований менеджером artisan у Laravel проекті, мета якого обробка даних певної таблиці БД.

Незважаючи на те, що верстка – це питання невибагливе, було прийняте рішення показати результати роботи над дизайном сайту. На рисунку 3.1 зображено сторінку пошуку визначень за термінами.

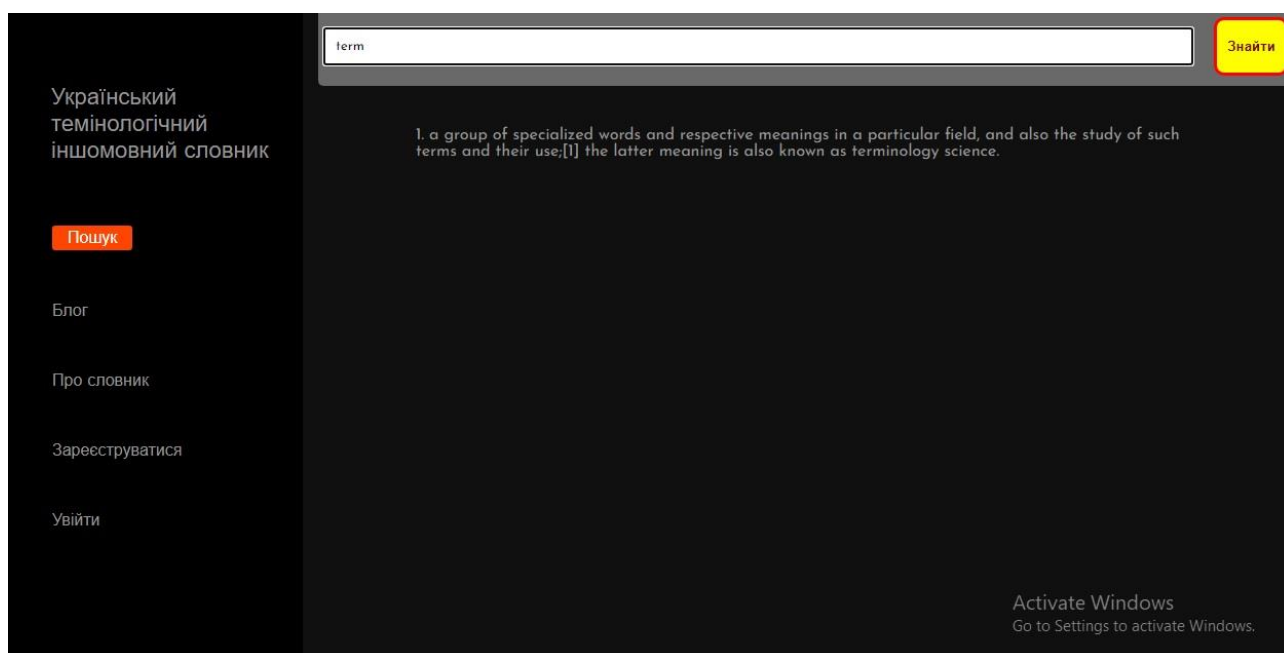


Рисунок 3.1 – Дизайн сторінки пошуку термінів та визначень

Як видно з даних зображень була вибрана темна тема, при якій більшість фону представлена чорним або темно сірими кольорами, а текстова інформація представлена білим або світлим шрифтом. Безліч різноманітних веб сайтів включаючи «Telegram», «YouTube», «Character.ai» та інші мають світлу та темну тему. І темна з певних причин визнається більшістю користувачів як асоціація з

комфортом та зручністю. Це зумовлює їхній вибір в умовах наявності можливості вибору при даних двох опціях.

На рисунку 3.2 зображено сторінку блогу, де представлена можливість шукати блог за назвою та категорією.

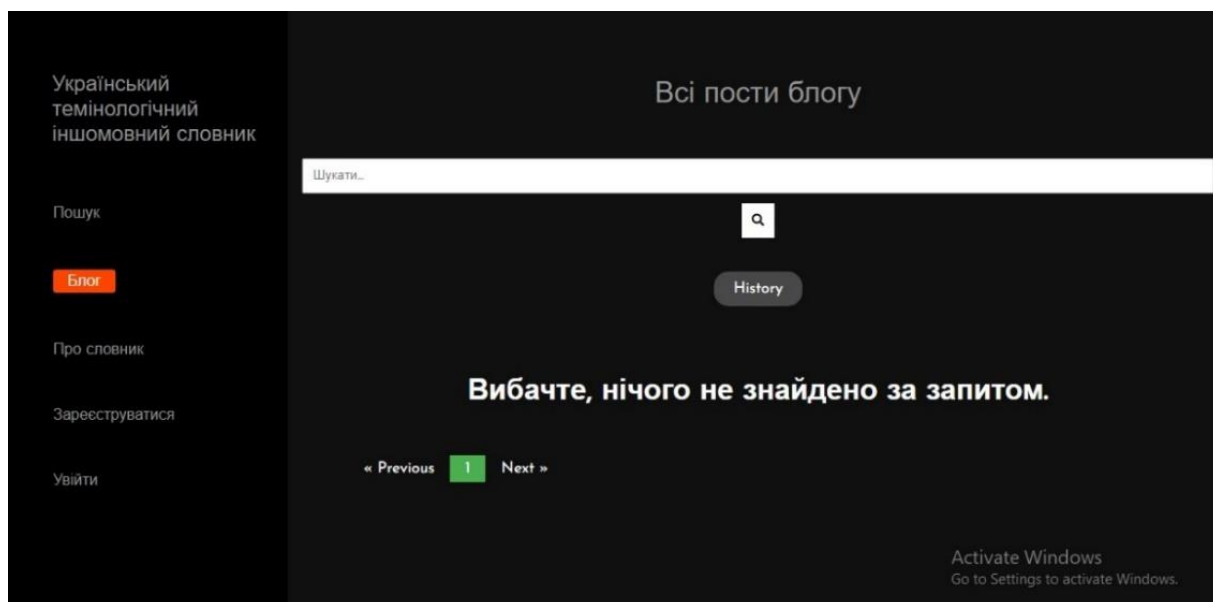


Рисунок 3.2 – Вигляд сторінки блогу

На рисунку 3.3 зображена сторінка реєстрації акаунту.

Рисунок 3.3 – Сторінка реєстрації користувача

На рисунку 3.4 зображена сторінка панелі управління акаунтом.

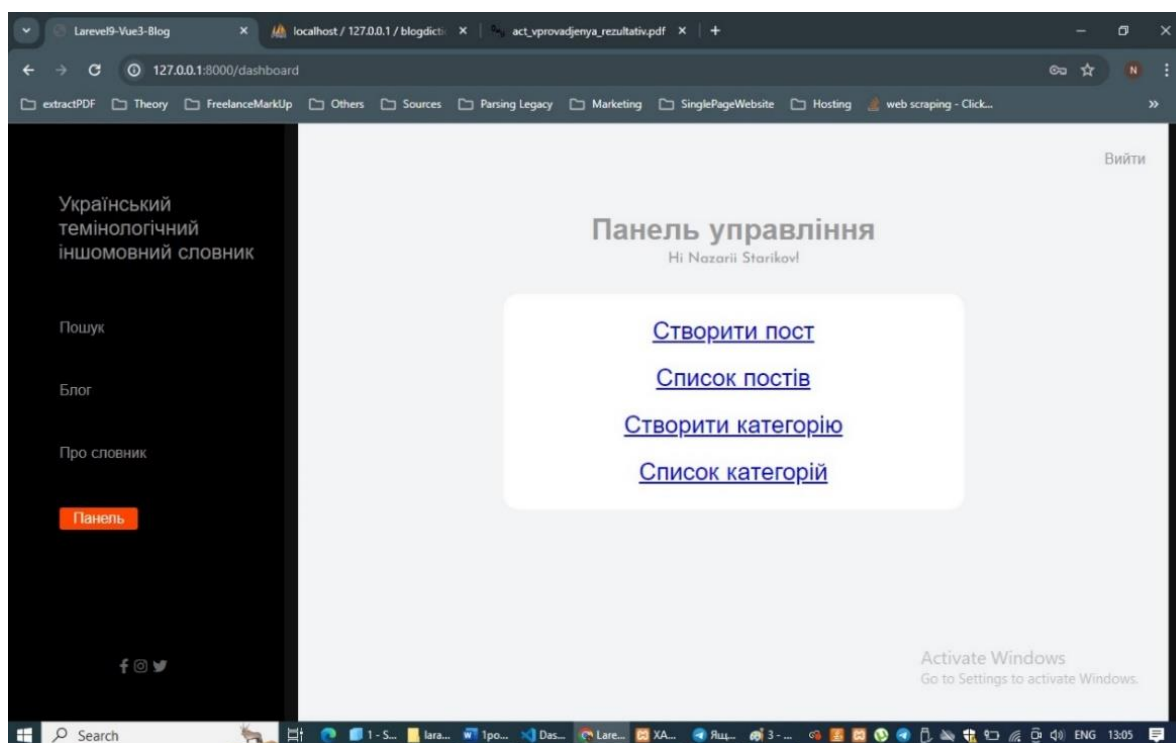


Рисунок 3.4 – Панель управління веб сайту

На рисунку 3.5 наявний дизайн сторінки створення посту.

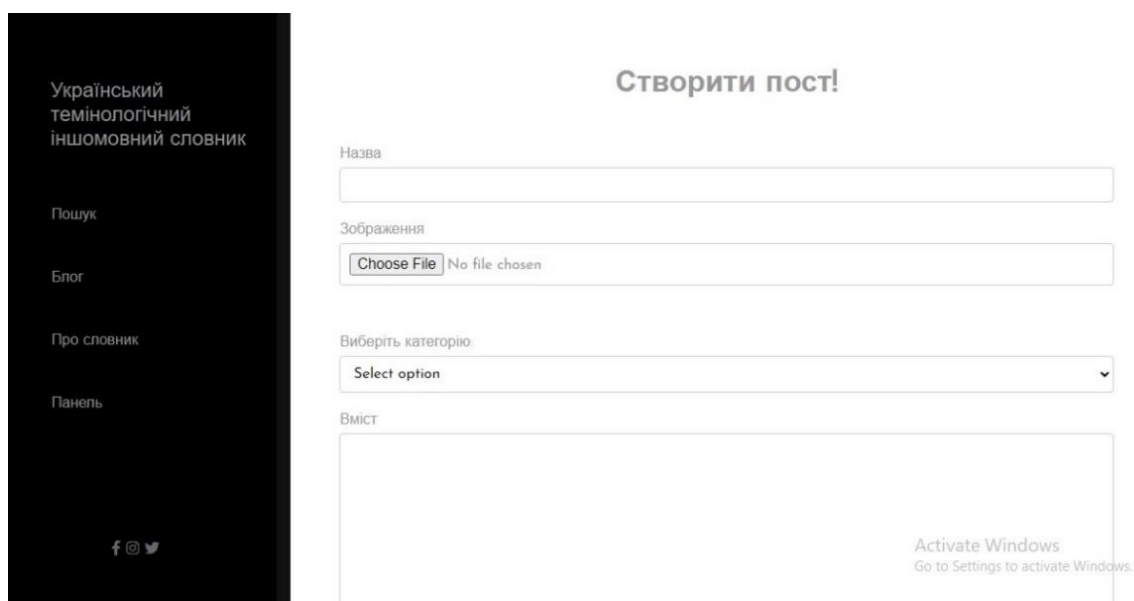


Рисунок 3.5 – Сторінка створення посту

Крім побудови веб сайту варто звернути увагу на процес заповнення бази даних термінами та визначеннями. Джерелом таких термінів та визначень є PDF файл, який являє собою словник під назвою «Department of Defense Dictionary of Military and Associated Terms». Даний словник вміщує терміни та визначення виключно англійською мовою. Визначення та терміни українською мовою отримані завдяки сервісу Google Translate.

Приклад коду програми написаної з використанням бібліотеки PyPDF2 наявний в додатку Б.

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Перш за все, розробка веб сайтів та консольних програм не потребує великих системних вимог попри те, що це залежить від складності та величини проекту. Однак, будь-яка розробка має певні вимоги, які виражаються у вимогах до програмного і апаратного забезпечення. У будь-якому випадку, процес розробки потребує наявності підключення до Інтернету, текстовий редактор, локальний сервер, браузер. Всі загальні вимоги до апаратного забезпечення будуються на базі системних вимог необхідних для вдалого функціонування заданих програм.

В якості текстового редактора був використаний Visual Studio Code, який характеризувався наявністю терміналу, файлового менеджера та можливості інсталяції певних розширень необхідних для зручного користування програмною в цілому. В якості локального веб серверу, який надавав можливість запускати веб сайт та оглядати бази даних за допомогою phpMyAdmin. Такий вибір був зумовлений використанням операційної системи Windows. В якості браузера можна використовувати фактично будь-який, а використовувався Google Chrome.

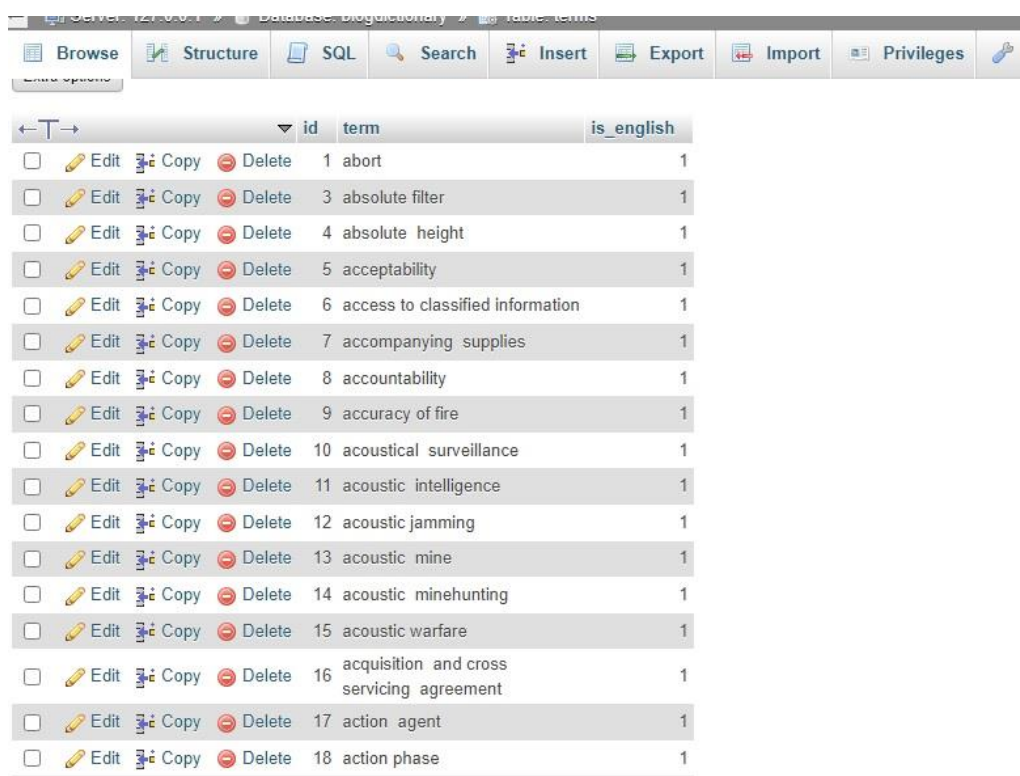
Розвиваючи проект з використанням технологій представлених фреймворками Laravel та Vue.js за допомогою XAMPP вмикаються такі сервіси як: MySQL та Apache. Використовуючи інтерфейс командного рядка artisan можна запустити проект, побудувати проект та виконати інші можливі функції

такі як: почистити кеш, тощо. В процесі розробки бували випадки, коли проект не оновлювався в браузері, що змушувало знаходити інші команди artisan такі як: `php artisan optimize`, тощо.

У випадку певних недоліків JavaScript коду використовуються `alert()` або `console.log()`. У випадку недоліків PHP коду використовуються `log` файл, де можна вивести всі необхідні дані у текстовому форматі.

При розробці програми з використанням мови програмування Python запуск програми реалізовується так само через командний рядок, але результати роботи можна оглянути в командному рядку або в `phpMyAdmin`, адже саме там оглядають бази даних. У випадку наявності помилок, які потребують детального розгляду їх можна оглядати в терміналі використовуючи звичні способи виведення помилок та іншої інформації запропоновані мовою програмування Python.

На рисунку 3.6 наявні результати успішного вставлення текстових даних з PDF файлу в БД.



	id	term	is_english
☐ Edit Copy Delete	1	abort	1
☐ Edit Copy Delete	3	absolute filter	1
☐ Edit Copy Delete	4	absolute height	1
☐ Edit Copy Delete	5	acceptability	1
☐ Edit Copy Delete	6	access to classified information	1
☐ Edit Copy Delete	7	accompanying supplies	1
☐ Edit Copy Delete	8	accountability	1
☐ Edit Copy Delete	9	accuracy of fire	1
☐ Edit Copy Delete	10	acoustical surveillance	1
☐ Edit Copy Delete	11	acoustic intelligence	1
☐ Edit Copy Delete	12	acoustic jamming	1
☐ Edit Copy Delete	13	acoustic mine	1
☐ Edit Copy Delete	14	acoustic minehunting	1
☐ Edit Copy Delete	15	acoustic warfare	1
☐ Edit Copy Delete	16	acquisition and cross servicing agreement	1
☐ Edit Copy Delete	17	action agent	1
☐ Edit Copy Delete	18	action phase	1

Рисунок 3.6 – Вигляд списку термінів у СУБД

3.3 Розробка бази даних

Як видно з принципів розробки даного проекту, по факту, існує дві бази даних, одна з яких зберігає дані стосовно блогу, а інша – стосовно словника. Обидві бази даних об'єднуються в єдину завдяки імпорту однієї бази даних в іншу. Таблиці з однієї БД можуть бути отримані завдяки міграціям, а таблиці іншої БД можуть бути створені штучно в СУБД phpMyAdmin та наповнені даними завдяки програми з використанням бібліотеки PyPDF2. При викладенні веб сайту на хостинг існує можливість загрузити лише одну базу даних.

Суть даного проекту полягає у створенні словника, де будуть наявні терміни та визначення на двох мовах, а саме українською та англійською. На один термін може приходитися більше одного визначення, а це означає, що терміни та визначення не можуть знаходитися в єдиній таблиці, адже це не піддається можливості з причини різної кількості термінів та визначень.

Загальний вигляд структури такої бази даних наявний на рисунку 3.7.

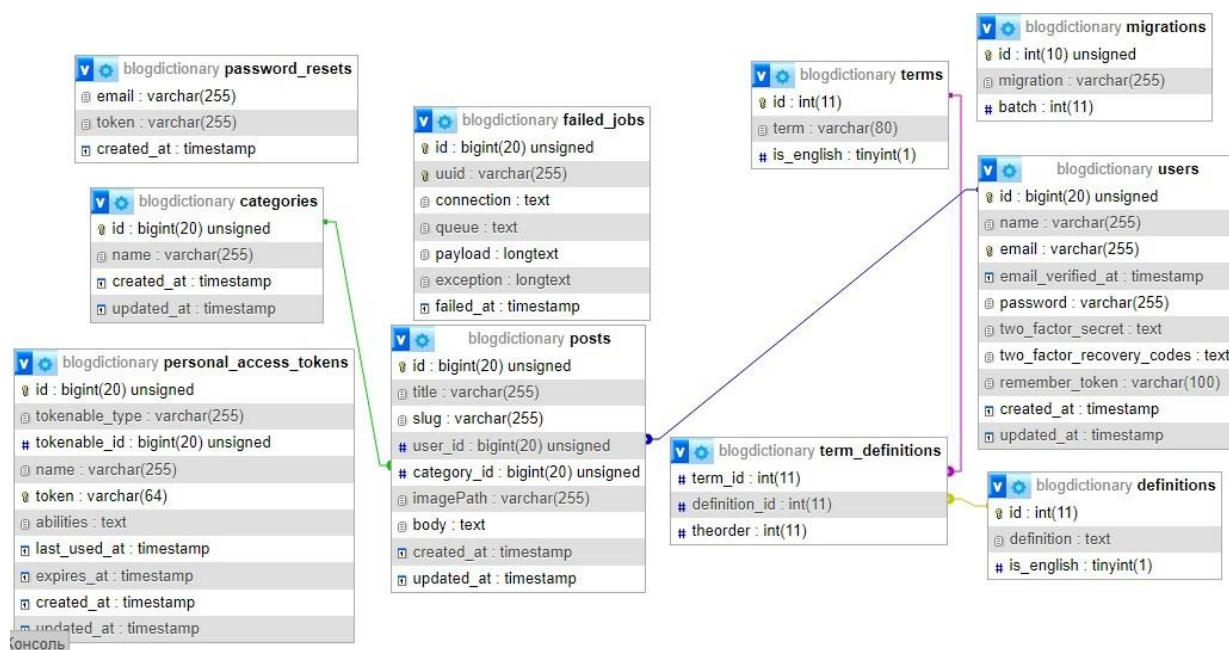


Рисунок 3.7 – Структура бази даних програмного продукту

Теоретично, можливо створити таблицю, де існують поля визначення та термінів, а якщо на один термін декілька визначень, то значення терміну дублюються. Такий підхід, однак, ненадійний. База даних повинна включати щонайменше даних аби виконувати запити за щонайменший період часу.

Виходячи зі структури PDF файлу та його вмісту було прийняте рішення розділити всі терміни та всі визначення на окремі сутності, що виразиться у створенні окремих таблиці та створити ще одну таблицю, де посилаючись на id з двох попередніх таблиць можна визначити відповідність терміну та визначення, а також присвоїти кожному визначенню свій номер, який буде корисний для подальшої нумерації у випадку, якщо термін має два або більше визначень.

Таблицями, які виражають такі сутності є `terms` та `definitions`, а таблиця, яка об'єднує id обох таблиць має назву `term_definitions`. Реляційний зв'язок даних трьох таблиць помітно в дизайнері `phpMyAdmin`.

Також варто взяти до уваги, що існують таблиці, які не грають функціональної ролі у функціонуванні програмного продукту, але грають такі роль при міграції та інших процесах розробки. Так наприклад, для міграції існує спеціальна таблиця, назва якої `migrations`.

Таблиця `users` відіграє ключову роль при збереженні даних юзера включаючи пароль, ім'я, електронна адреса, тощо. Так як пост належить конкретному користувачеві, а також пост належить до певної категорії, то існують реляційні зв'язки між таблицями `post`, `categories` та `users`.

3.4 Захист інформаційно-комп'ютерної системи

Захист інформації – це сукупність методів і засобів, що забезпечують цілісність, конфіденційність і доступність інформації за умов впливу на неї загроз природного або штучного характеру, реалізація яких може призвести до завдання шкоди власникам і користувачам інформації.

Захист інформації ведеться для підтримки таких властивостей інформації як: цілісність, конфіденційність та доступність.

Основою забезпечення конфіденційності є реалізація реєстрації та авторизації зареєстрованих користувачів, кожен з яких має власний пароль. Відповідно, з метою забезпечення їхньої інформаційної безпеки, всі паролі піддаються процесу хешування перед процесом змінення у БД або вставлення нових паролів у БД, про свідчить запис відповідної таблиці користувачів у БД переглянутий за допомогою phpMyAdmin. Це можливо завдяки спеціальному класу, наявність якого передбачена структурою проекту Laravel. Табличний запис такого роду видно на рисунку 3.8.

Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

SELECT * FROM `users`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	id	name	email	email_verified_at	password	two_factor
☐ Edit Copy Delete	2	Nazarii Starikov	stankevican726@gmail.com	NULL	\$2y\$10\$0P5xPS9YIL5yVw6v9QPLoe4V5ocPazixp5qKrlpUGym...	NULL

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

Рисунок 3.8 – Наявність результатів хешування паролів у БД

Висновки до розділу 3

У третьому розділі роботи було розглянуто процес розробки програмного продукту, який поєднує в собі використання сучасних веб-технологій, таких як Laravel, Vue.js та Python. Було детально описано структуру проекту, архітектурні рішення, а також принципи роботи окремих компонентів системи.

Вибір засобів, методів і алгоритмів для вирішення завдань розробки програмного забезпечення був обґрунтований та здійснений на основі докладного аналізу існуючих інструментів. Використання формалізованих методів синтезу моделей програмного забезпечення та складення специфікацій вимог забезпечило чіткість і структурованість розробленої системи.

Розробка серверної частини за допомогою Laravel дозволила створити надійну та масштабовану основу для веб-застосунку, забезпечивши високу продуктивність та безпеку даних.

Клієнтська частина, розроблена з використанням Vue.js, дозволила створити інтерактивний та динамічний інтерфейс користувача. Завдяки реактивності Vue.js, вдалося досягти високої швидкості реагування інтерфейсу на дії користувача та підвищити зручність використання системи.

Таким чином, у ході роботи було досягнуто основної мети - створення ефективного та зручного програмного продукту, який поєднує у собі переваги різних технологій. Запропонований підхід до розробки дозволяє забезпечити високу якість програмного забезпечення, його масштабованість та адаптивність до майбутніх змін та вимог користувачів.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи бакалавра були досягнуті важливі цілі та завдання, поставлені перед дослідженням та розробкою інформаційної системи.

В ході виконання даної кваліфікаційної роботи бакалавра був здійснений аналіз предметної області та методології розробки термінологічного іншомовного онлайн словника та був розроблений іншомовний термінологічний онлайн словник. У результаті даного аналізу було прийняте рішення, що даний проект має шанси користуватися попитом на ринку, адже даний продукт може використовуватися в освітніх цілях. Методологія розробки полягає у збиранні інформації з певних PDF джерел та інтеграції БД з базою даних веб сайту, який також матиме функції блогу.

Однією з цілей даної кваліфікаційної роботи бакалавра було ознайомлення з основами роботи з консольними проектами Python з метою збору текстових даних з PDF файлів. Дана ціль було досягнута успішно враховуючи наявність певних термінів та визначень у базі даних. У якості використовуваної технології було прийнято рішення використати «PyPDF2».

Був реалізований простий блог, де була наявна сторінка, яка виконує функцію словника. Розробка була реалізована завдяки Laravel та Vue.js, а також завдяки мові програмування Python. У даному блозі наявні такі функції як: авторизація, реєстрація, створення та редагування постів за певними категоріями, створення та редагування категорій, пошук постів, пошук словникових даних виражених у термінах та визначеннях.

Здійснено тестування і налагодження даного проекту. В результаті тестування всі недоліки і помилки були виправлені.

Забезпечено захист та інформаційно-комп'ютерної системи, зокрема забезпечено захист цілісності, конфіденційності та доступності інформації.

В цілому було виконано всі поставлені завдання, а розроблений онлайн словник іншомовних термінів має практичну цінність.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мего-Інфо Юридичний портал №1. URL: <http://mego.info/> (дата звернення: 31.01.2024).
2. English-Ukrainian military dictionary Англо-український військовий словник. URL: <https://english-military-dictionary.org.ua/> (дата звернення 31.01.2024).
3. Одна потужна ІТ-платформа для управління інформацією та мінімізації ризиків усієї компанії URL: <https://liga360.ligazakon.net/> (дата звернення: 01.02.2024).
4. UX Design Tutorial For Beginners | (Course 1/7) Foundations User Experience Design. Відео соціальної мережі YouTube каналу MyLesson. URL: https://www.youtube.com/watch?v=6qLq7xkodA8&list=PLtS8Ubq2bIIUWixdH_1Q2yzZh8QJSBVmT&index=1 (дата звернення: 12.02.2024).
5. Ключові методології розробки програмного забезпечення: робота команди зсередини. Wezom. URL: <https://wezom.com.ua/ua/blog/metodologija-razrabotki-programmnogo-obespechenija> (дата звернення: 16.02.2024).
6. What is CRUD? CodeAcademy. URL: <https://www.codecademy.com/article/what-is-crud> (дата звернення: 01.03.2024).
7. Документація до фреймворку Laravel. URL: <https://laravel.com/docs/9.x> (дата звернення: 04.03.2024).
8. SPA (Single-page application). Стаття веб сайту MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Glossary/SPA> (дата звернення 05.03.2024).
9. Документація до фреймворку Vue.js URL: <https://vuejs.org/guide/introduction.html> (дата звернення: 07.03.2024).
10. Документація до бібліотеки PyPDF2. URL: <https://pypdf2.readthedocs.io/en/3.x/> (дата звернення: 11.03.2024).

ДОДАТКИ

Додаток А

Лістинг файлу index.js

```

import { createRouter, createWebHistory } from "vue-router";

import About from '../pages/About.vue'
import Blog from "../pages/Blog.vue";
import SingleBlog from "../pages/SingleBlog.vue";
import Login from "../pages/Login.vue";
import Register from "../pages/Register.vue";
import Dashboard from "../pages/Dashboard.vue";
import                                CreateCategories                                from
"../pages/categories/CreateCategories.vue";
import CategoriesList from "../pages/categories/CategoriesList.vue";
import EditCategories from "../pages/categories/EditCategories.vue";
import CreatePosts from "../pages/posts/CreatePosts.vue";
import                                DashboardPostsList                                from
"../pages/posts/DashboardPostsList.vue";
import EditPosts from "../pages/posts/EditPosts.vue";
import Search from "../pages/Search.vue";

const routes = [
  {
    path: "/",
    name: "Search",
    component: Search,
  },
  {
    path: "/about",
    name: "About",
    component: About,
  },
  {
    path: "/blog",
    name: "Blog",
    component: Blog,
  },
  {
    path: "/blog/:slug",
    name: "SingleBlog",
    component: SingleBlog,
    props: true,
  },
  {
    path: "/login",
    name: "Login",
    component: Login,
    meta: { requiresGuest: true }
  },
  {

```

```

path: "/register",
name: "Register",
component: Register,
meta: { requiresGuest: true }
},
{
path: "/dashboard",
name: "Dashboard",
component: Dashboard,
meta: { requiresAuth: true }
},
{
path: "/categories/create",
name: "CreateCategories",
component: CreateCategories,
meta: { requiresAuth: true },
},

{
path: "/categories",
name: "CategoriesList",
component: CategoriesList,
meta: { requiresAuth: true },
},

{
path: "/categories/:id/edit",
name: "EditCategories",
component: EditCategories,
meta: { requiresAuth: true },
props: true
},

{
path: "/posts/create",
name: "CreatePosts",
component: CreatePosts,
meta: { requiresAuth: true },
},

{
path: "/dashboard-posts",
name: "DashboardPostsList",
component: DashboardPostsList,
meta: { requiresAuth: true },
},

{
path: "/posts/:slug/edit",
name: "EditPosts",
component: EditPosts,

```

```
meta: { requiresAuth: true },
props: true
},
];

const router = createRouter({
  history: createWebHistory(),
  routes,
});

router.beforeEach((to, from) => {
  const authenticated = localStorage.getItem("authenticated");

  if (to.meta.requiresGuest && authenticated) {
    return {
      name: "Dashboard",
    };
  } else if (to.meta.requiresAuth && !authenticated) {
    return {
      name: "Login",
    };
  }
});

export default router;
```

Кінець лістингу

Додаток Б

Лістинг файлу start.py

```

import PyPDF2
import string
import re
import mysql.connector
from mysql.connector import errorcode
import os
# <= 522
def extract_text_from_pdf(pdf_path, page_number):
    text = ""
    with open(pdf_path, "rb") as file:
        reader = PyPDF2.PdfReader(file)
        if page_number < 0 or page_number >= len(reader.pages):
            print("Invalid page number")
            return None
        page = reader.pages[page_number]
        text = page.extract_text()
    return text

'''def DefinitionNumber(text):
    i = 0
    while i < len(text):
        if text[i - 1].isdigit() == True and i > 0 and text[i] ==
'.':
            return True
        i += 1
    return False'''

'''def filter_printable_characters(text):
    printable_characters = set(string.printable)
    dash_character = '-'
    printable_characters.update(dash_character)
    return ''.join(filter(lambda x: x in printable_characters,
text))'''

cnx      =      mysql.connector.connect(user='root',      password='',
host='localhost', database='dictionary')
cursor = cnx.cursor()

pdf_path  =  "C:\\Users\\user\\Desktop\\MD\\Doubted\\Department of
Defense3.pdf"
page = 8
TheWholeText = ""
'''EnglishAlphabet
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
punctuation_symbols = string.punctuation'''
while page <= 12:

```

```

        text = extract_text_from_pdf(pdf_path, page)
        cleaned_text = re.sub(r'As Amended Through April 2010', '', text)
# Remove "As Amended Through April 2010"
        cleaned_text = re.sub(r'– See evaluation', '', text)
        cleaned_text = re.sub(r'(?<!-)(?<!\d)\n(?!\d)', ' ',
cleaned_text) # Replace newline not preceded by "-" or digit with space
        cleaned_text = re.sub(r'\s*-\s*', ' - ', cleaned_text) # Add
spaces around hyphens
        cleaned_text = re.sub(r' - (?!.*? - )', '\n', cleaned_text) #
Add newline after definitions
        cleaned_text = re.sub(r'\([^)]*\)', '', cleaned_text) # Remove
anything within parentheses
        cleaned_text = re.sub(r'See also.*?[\.\n]', '', cleaned_text) #
Remove "See also" and text after it until a period or newline
        cleaned_text = re.sub(r'\(.*\)', '', cleaned_text) # Remove
"(*)"
        cleaned_text = re.sub(r'(\d+)\.\s+', r'\1. ', cleaned_text) #
Add space after numbered definitions
        cleaned_text = re.sub(r'\b(\d+)\s+', r'\1 ', cleaned_text) #
Remove space after numbers at the beginning of definitions
        cleaned_text = re.sub(r'\b[A-Za-z]\b', '', cleaned_text) #
Remove single alphabet letters
        cleaned_text = re.sub(r'\bJP\s\d+\s*-\s*\d+\b', '',
cleaned_text)
        page_number_to_remove = str(page - 7)
        cleaned_text = re.sub(rf'\b{page_number_to_remove}\b(?:!\. )', '',
cleaned_text)
        TheWholeText += cleaned_text.strip()
        cleaned_text = ""
        page += 1

#print(TheWholeText)

dollar = False
OtherCharacter = False
TheTextPart = ""
TermId = 1
DefinitionId = 1
OrderDefinition = 1

i = 0
while i < len(TheWholeText):
    if TheWholeText[i] == '$':
        if dollar == False:
            dollar = True
        else:
            term_sql = "INSERT INTO terms (term, is_english) VALUES
(%s, %s)"

            cursor.execute(term_sql, (TheTextPart.strip(), True))
            cnx.commit()
            TheTextPart = ""

```

```

        dollar = False
        OrderDefinition = 1
    if TheWholeText[i] == '&':
        if OtherCharacter == False:
            OtherCharacter = True
        else:
            #definitionnumber = DefinitionNumber(TheTextPart)
            #if definitionnumber == False:
            sql = "INSERT INTO definitions (definition, is_english)
VALUES (%s, %s)"
            cursor.execute(sql, (TheTextPart.strip(), True))
            cnx.commit()
            sql = "INSERT INTO term_definitions (term_id,
definition_id, theorder) VALUES(%s, %s, %s)"
            cursor.execute(sql, (TermId, DefinitionId,
OrderDefinition))
            cnx.commit()
            DefinitionId += 1
            OrderDefinition += 1
            '''else:
                theorder = 1
                thenumber = ""
                Definition = False
                onedefinitiontext = ""
                j = 0
                while j < len(TheTextPart):
                    if TheTextPart[j].isdigit() == True:
                        thenumber += TheTextPart[j]
                        Definition = False
                        """sql = "INSERT INTO definitions
(definition, is_english) VALUES (%s, %s)"
                        cursor.execute(sql,
(onedefinitiontext.strip(), True))
                        cnx.commit()
                        sql = "INSERT INTO term_definitions
(term_id, definition_id, theorder) VALUES(%s, %s, %s)"
                        cursor.execute(sql, (TermId,
DefinitionId, theorder))
                        cnx.commit()
                        onedefinitiontext = ""
                        DefinitionId += 1"""
                    if j > 0 and TheTextPart[j] == '.' and
TheTextPart[j - 1].isdigit() == True and thenumber != "" and int(thenumber)
== theorder:
                        thenumber = ""
                        theorder += 1
                        Definition == True
                    if Definition == True:
                        onedefinitiontext += TheTextPart[j]
                    j += 1
                pass'''

```

```
        TheTextPart = ""
        TermId += 1
        OtherCharacter = False
        if TheWholeText[i] != '$' and TheWholeText[i] != '&'amp;' and (dollar
== True or OtherCharacter == True):
            TheTextPart += TheWholeText[i]

        i += 1

    cursor.close()
    cnx.close()
```

Кінець лістингу