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

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інженерії програмного забезпечення, канд. техн. наук, доцент

_____Є. О. Давиденко

«____» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

ПРОГРАМНИЙ ЗАСТОСУНОК ПЕРЕКЛАДУ СЛІВ

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 409.21810919

Студент: _____ К. В. Недуга
підпис
«__» червня 2022 р.

Керівник: канд. техн. наук, доцент _____ Є. О. Давиденко
підпис
«__» червня 2022 р.

Консультант: канд. техн. наук, доцент _____ А. О. Алексєєва
підпис
«__» червня 2022 р.

Миколаїв – 2022

ЗМІСТ

ВСТУП

Світ дуже швидко розвивається даючи нам багато нових можливостей для подорожей, навчання або роботи за кордоном. Але нажаль без знання мови користуватися цими можливостями важко. Для сучасної людини вивчення іноземної мови є невід'ємною частиною повсякденного життя. Хтось вчиться самостійно, а хтось звертається за допомогою до професіоналів. Але так чи інакше помічником у цій справі стане програмний застосунок для перекладу слів.

Даний застосунок розроблений для мовної онлайн школи Engine. Найважливіша функція цієї системи швидко та точно перекладати слова, допомагаючи цим самим у вивченні іноземної мови.

Існує ще одне важливе застосування для системи. Представлений застосунок виступає також рекламною складовою для онлайн школи. Пройдучи реєстрацію користувач отримує доступ до додаткових функцій.

Основні особливості програмного застосунку, який буде створений в ході виконання бакалаврської кваліфікаційної роботи:

1) Переклад слів на трьох мовах.
2) Реєстрація (за бажанням).
3) Додаткові функції які стануть доступні після реєстрації

Тема роботи є актуальною, оскільки кількість людей які вивчають іноземну мову, зросла та з кожним роком буде зростати. До того ж цим сервісом можна користуватися не тільки для вивчення мови.

Об'єкт кваліфікаційної роботи – процес перекладу іноземних слів на українську мову та навпаки.

Предмет кваліфікаційної роботи – підходи використання фреймворку Laravel у розробці лінгвістичного програмного забезпечення.

Мета кваліфікаційної роботи – організація та полегшення процесу вивчення іноземної мови, за рахунок швидкого перекладу та додаткових матеріалів які знаходяться у системі.

Для досягнення поставленої мети необхідно розв'язати наступні

завдання:

– аналіз предметної області;

– розробка вимог до системи перекладу слів на основі інформації про предметну область;

– налагодження роботи баз даних;

– проєктування системи перекладу;

– реалізація, тестування та відлагодження вебзастосунку для відстеження помилок.

Для розв’язання поставлених завдань буде використана мова програмування php.

Актуальність спеціального розділу про охорону праці зумовлена тим, що розповсюдження та широке використання інформаційних технологій сприяло розробці норм та вимог до використання комп’ютерної техніки на підприємствах. Дані норми та вимоги регулюються законодавством України. Саме тому дуже важливо створювати належні умови праці, виключати або мінімізувати ризики для життя та здоров’я робітників.

Метою спеціального розділу про охорону праці є дослідження питань охорони праці, які безпосередньо пов’язані з діяльністю розробника програмного забезпечення.

1 АНАЛІТИЧНИЙ ОГЛЯД СИСТЕМ ДЛЯ ПЕРЕКЛАДУ СЛІВ

1.1 Існуючі системи для перекладу слів

У наш час існує велика кількість онлайн систем які використовуються для перекладу слів, вивчення іноземної мови тощо. Ще декілька десятків років тому назад, людство користувалось надрукованими словниками. Це приносило деякі незручності у вивченні мови чи в перекладі тексту , адже переклад одного слова займав багато часу. Також книгу потрібно було носити з собою, а деколи вони були великими і важкими. З приходом автоматизованих систем для перекладу вищесказані проблеми зникли. Нам більше не потрібно витрачати багато часу на пошук одного слова, бо переклад об’ємного тексту тепер займає секунди.

Зараз існує безліч мобільних або десктопних застосунків для прекладу слів, текстів, документів різних форматів та онлайн перекладачів. Достатньо ввести у пошуковому рядку слово “перекладач” і в ту ж секунду браузер надасть нам великий список лінгвістичних систем з різноманітними функціями де кожен зможе знайти найбільш підходящий варіант для задоволення своїх потреб. Переклад слів, текстів, документів, голосове відтворення слів, рукописне або голосове введення тексту тощо. Це все те що полегшує та пришвидшує процес вивчення іноземної мови і не тільки.

Вагомий плюс таких систем , це звісно швидкий переклад. Ну і в кожній такій системі є свої додаткові функції через які користувач надає перевагу саме їй. Мінуси є також у абсолютно будь-якому такому застосунку. Кожний користувач сам для себе виділяє плюси та мінуси під час використання даних систем.

Якщо в пошуковому рядку ввести “перекладач”, ми отримаємо результат у вигляді наступного списку онлайн систем:

– Google translate – найбільш відомий та використовуваний онлайн застосунок; він найперший з’являється у результатах пошуку; сервіс надає можливість перекладу на 108 мов світу, перекладу документів, голосове та письмове введення тексту, голосове відтворення слів, розпізнавання тексту на фото, перегляд історії перекладу, переклад сайту по посиланню. Також у цього

сервісу є мобільний застосунок з більш широким функціоналом. Але ця система не підійде для високоточного перекладу слів, (про це користувача попереджають й самі розробники) та не має можливості перекладати текст з великим обсягом (рис.1.1) [1].

– Deepl – високоточний онлайн перекладач від німецької компанії Deepl GmbH. Служба використовує згорткові нейронні мережі, які навчаються на основі бази Linguee. Переклад генерується за допомогою суперкомп'ютера. Перевагами даної системи є переклад тексту у необмеженому об’ємі, переклад файлів різних типів (.pdf, .docx, .pptx), редагування перекладеного тексту одразу на сторінці та захист даних. Мінусів дуже мало, але вони є. Переклад у необмеженому об’ємі можливий тільки якщо користувач оформить платну підписку, без неї можна завантажити текст довжиною у 5000 символів. Нажаль платна підписка доступна у обмеженому списку країн, наша країна до цього списку не входить (рис.1.2) [2].

– Reverso – онлайн перекладач який належить французькій компанії Reverso – Softissimo. Сервіс працює завдяки штучному інтелекту (нейронний машинний переклад). Система включає в себе велику кількість функцій: переклад файлів різних типів на 25 мов, можливість підбору синонімів, вбудована перевірка орфографії для вхідного тексту (що робить його більш точним), відмінювання слів на 10 мовах, голосова вимова перекладеного тексту. Присутня реєстрація, але вона є не обов’язковою. Після оформлення преміум підписки відривається доступ до більшої кількості функцій. Наприклад можна перекладати файли з PDF, Word або Power Point зберігаючи вихідний шаблон (рис.1.3) [3].

Кожен такий онлайн сервіс має свої переваги так недоліки. До основної і найбільшої переваги можна віднести швидкий переклад як одного слова так і великого тексту. Недоліком є те що переклад тексту є не завжди точним.

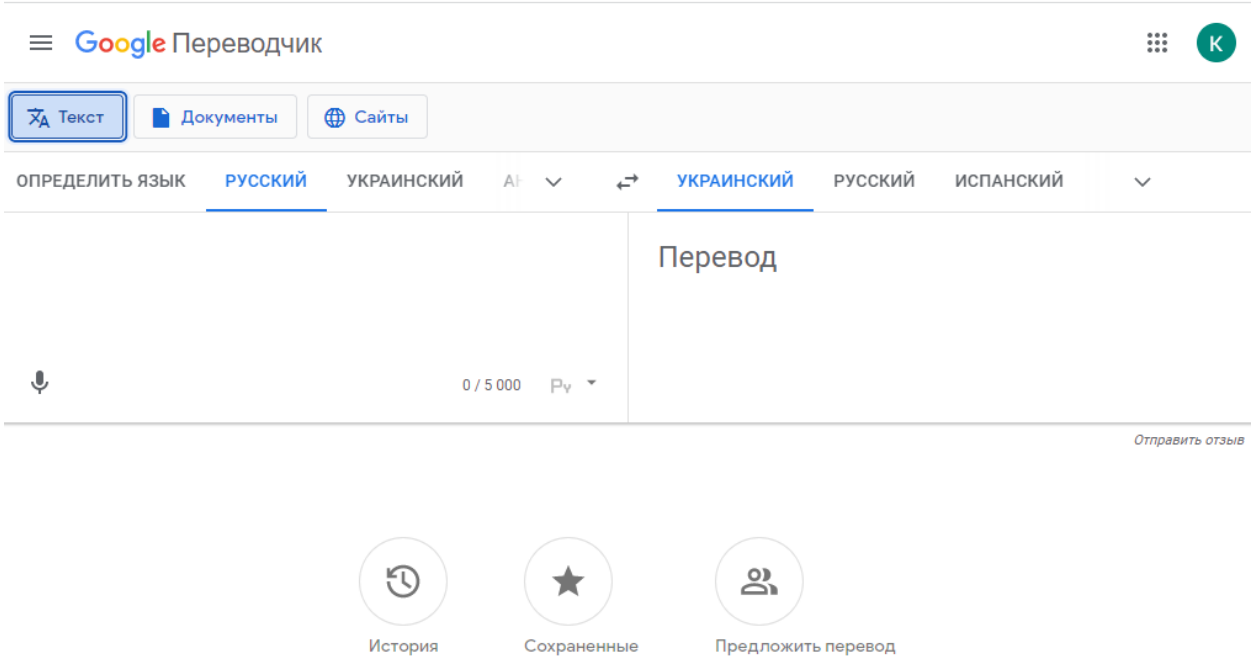

Рисунок 1.1 – Зовнішній вигляд Google Translate

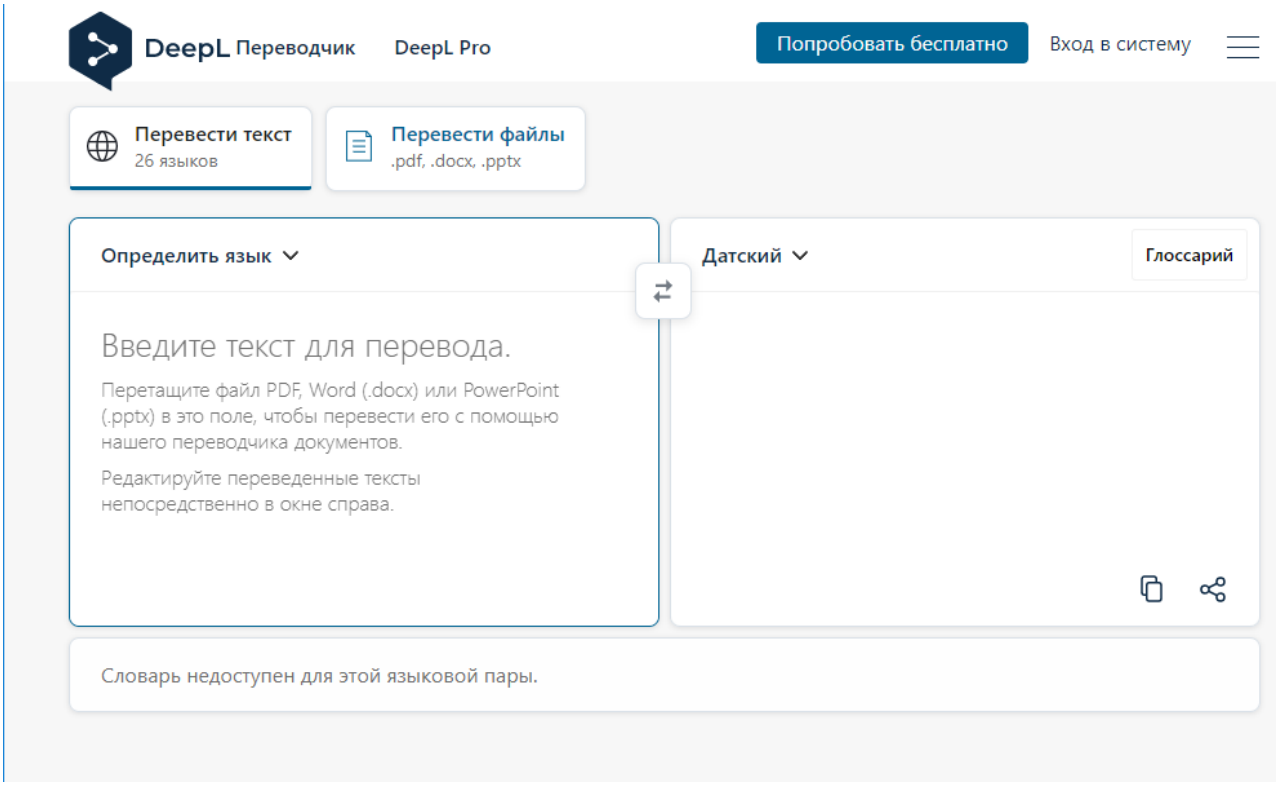

Рисунок 1.2 – Зовнішній вигляд Deepl

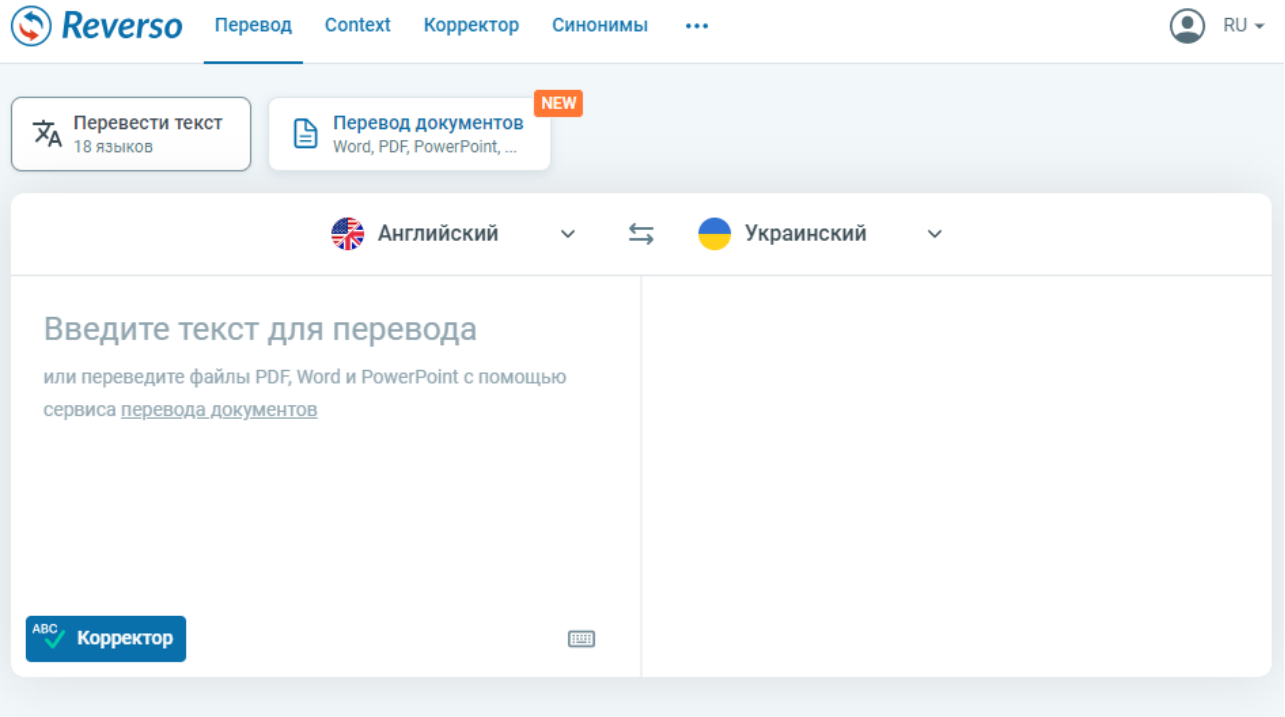

Рисунок 1.3 – Зовнішній вигляд Reverso

Основний список сервісів які представлені вище, першими потрапляють на око користувачу, коли у пошуковий рядок вводиться запит стосовно перекладу. Хоча дані онлайн сервіси представлені першими у результаті пошуку, але у мережі їх набагато більше, і усі вони схожі між собою, тому цих трьох застосунків буде цілком достатньо для загального представлення схожих систем.

Також майже кожний такий сервіс додатково ще має мобільні та десктопні застосунки. Вони містять у собі більш широкий функціонал. Найбільшою популярністю користуються саме мобільні застосунки адже це найбільш зручно мати під рукою доступ до функцій перекладу. Менш затребуваними залишаються десктопні програми.

Для прикладу можна розглянути мобільний застосунок Google Translate (рис. 1.4). Функціоналу який він містить у собі більш ніж достатньо для задоволення термінових потреб користувача. На відміну від онлайн версії тут доступне рукописне та голосове введення тексту. Також є дуже зручна функція через яку користувачі надають перевагу саме цьому застосунку – це розпізнавання та переклад тексту на фото. Також доступна функція “розмова”,

можна обрати дві будь-які мови, за допомогою голосового розпізнавання можна спілкуватися не знаючи мови (особливо корисно та зручно у подорожах). Доступний синхронний переклад. Перекладений текст відображається у історії, також його можна зберігати для зручності, для цього відведений спеціальний розділ. Найбільшим плюсом цього застосунку є офлайн переклад, необхідно лише завантажити потрібний мовний пакет і можливо буде користуватися усіма вищеперерахованими функціями без наявності інтернет з'єднання.

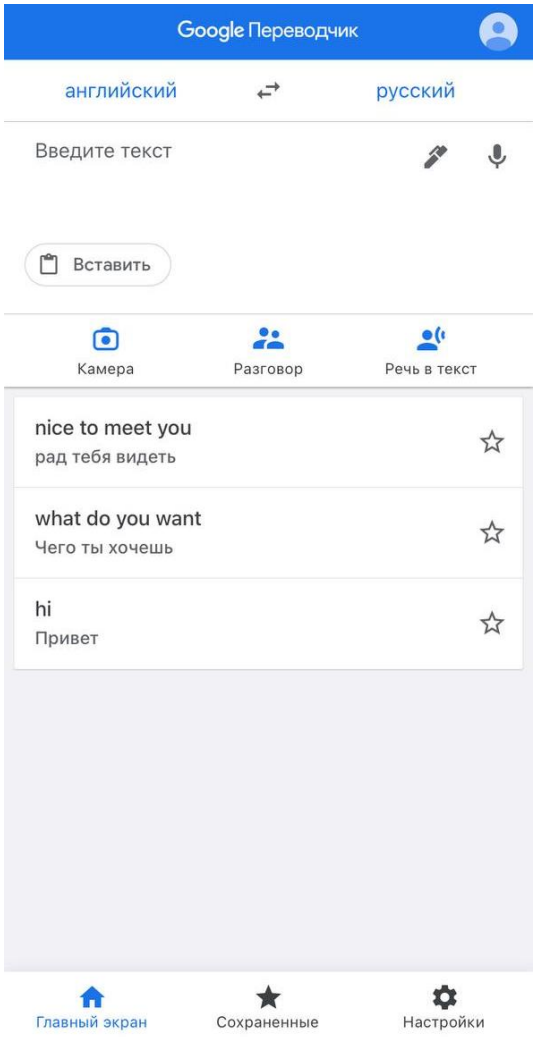

Рисунок 1.4 – Зовнішній вигляд мобільного застосунку Google Translate

Для більш детального аналізу розглянемо мобільний та десктопний застосунок Deepl. Його функціонал майже схожий з функціоналом попереднього застосунку. Той самий голосовий ввід, переклад тексту по фото збереження перекладеного текст, голосове відтворення. Але не зважаючи на широкий функціонал є вагомий мінус. Без інтернет підключення переклад буде не

можливий. Також переклад по фото доступний не на всіх мовах з представленого списку(рис. 1.5).

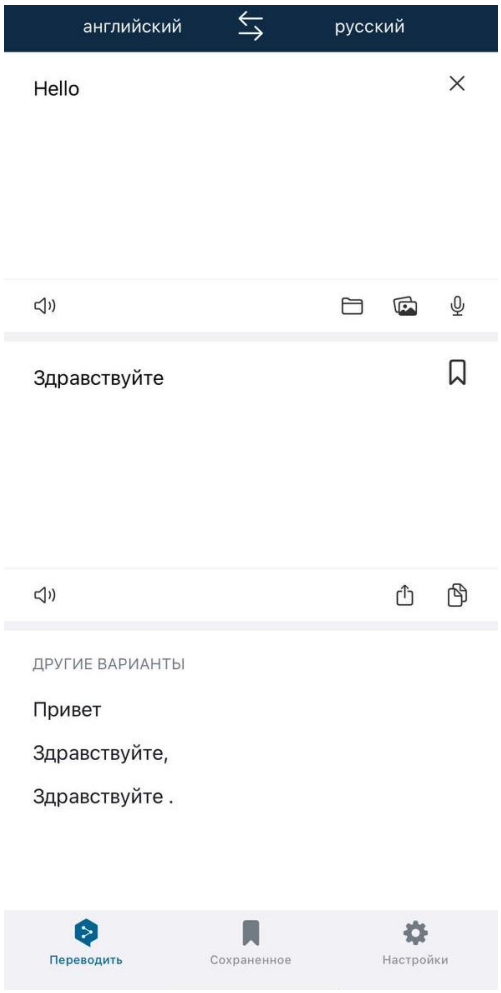

Рисунок 1.5 – Зовнішній вигляд мобільного застосунку Deepl

У десктопній версії програми, окрім функцій які були вказані на абзац вище, доступні ще декілька: переклад документів формату .docx та .pptx, можливість перекладу зі снімку екрану, вибір форми звернення, а також вибір глосарію. Найбільший мінус як і у мобільному застосунку це відсутність офлайн перекладу і порівняно з онлайн сервісом набагато менший функціонал(рис. 1.6).

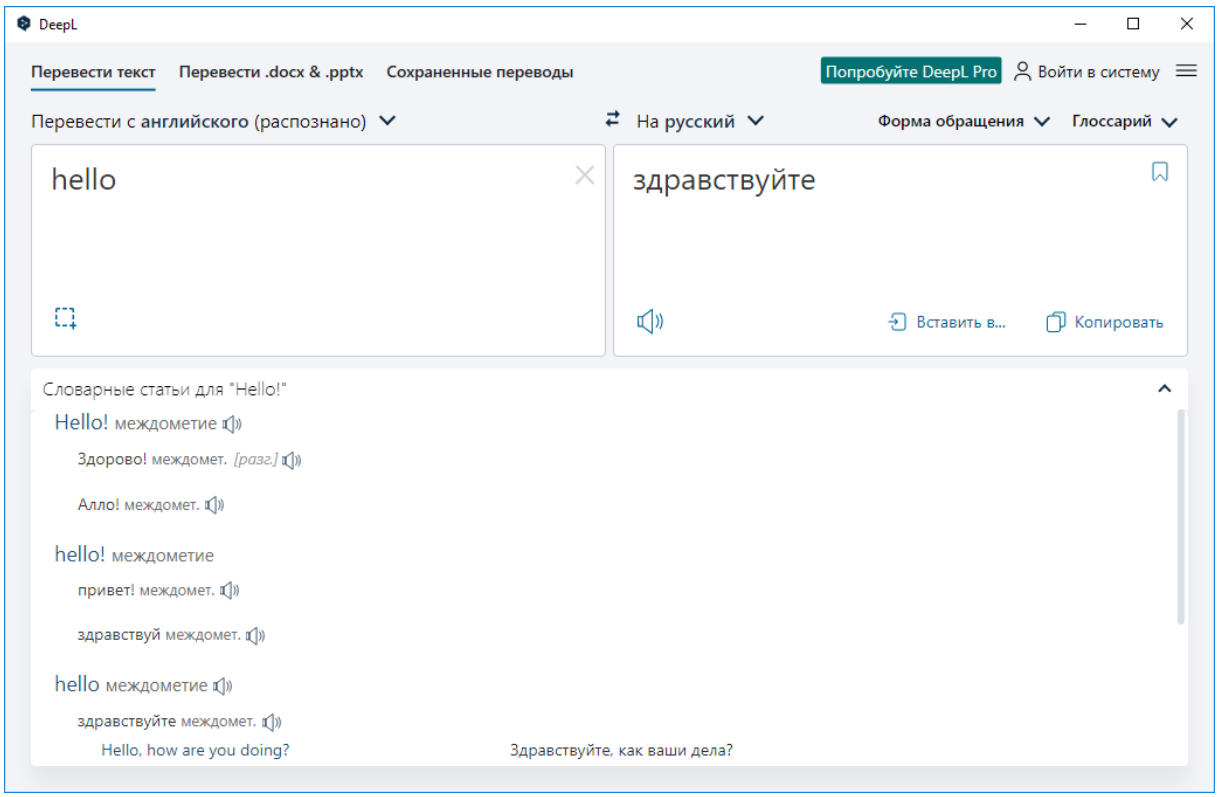

Рисунок 1.6 – Зовнішній вигляд десктопного застосунку Deepl

Розглянемо таблицю 1.1 у якій представлені основні переваги та недоліки розглянутих аналогів

Таблиця 1.1 – Порівняння переваг недоліків аналогів

	Платна підписка	Необмежений об’єм перекладу	Переклад файлів	Голосове відтворення тексту
GoogleTranslate	-	-	+	+
Deepl	+	+	+	+
Reverso	+	+	+	+

Отже, головні переваги: швидкий переклад, голосове введення і переклад документів. Із мінусів це не завжди точний переклад, обмежений об’єм тексту.

1.2 Загальні відомості про створений модуль, висвітлення проблем та їх вирішення

У наш час майже кожна людина, а особливо молоде покоління, стикається з вивченням іноземної мови як невід'ємної частини повсякденного життя: в садочку, в школі, в університеті, самостійно чи з викладачем. Зараз знання іноземної мови необхідна складова життя: для навчання, подорожей, безпроблемної комунікації з людьми інших національностей. При сучасних обставинах у світі, як наприклад Covid-19, коли усі перейшли на дистанційну форму навчання, великої популярності набули різноманітні мовні онлайн школи. І як вище було вказано, незмінним помічником у цій справі стане програмний застосунок для перекладу слів.

Отже, для вирішення поставленої задачі, нам необхідно реалізувати наступні функції:

- реєстрація/авторизація користувача;
- переклад слів і тексту до об'ємом до 255 символів;
- зберігання файлів з граматикою двох мов;
- скачування файлів з граматикою;
- можливість перегляду історії скачаних файлів;
- зміна мови місцями.

Вище вказаний функціонал полегшить користувачеві вивчення іноземної мови, запам'ятовування нових слів, переклад невеликих текстів або фраз. Якщо потрібно вивчити або повторити теми з граматики, спеціальний розділ де їх можна переглянути та скачати стане в нагоді.

1.3 Специфікації вимог до програмного забезпечення

ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

Призначення застосунку, для якого розробляється програмне забезпечення

Дане програмне застосунок створений для перекладу слів або невеликого за об'ємом тексту, зберігання та скачування допоміжних для вивчення мови файлів ,

за рахунок інтеграції зі спеціалізованим API.

Погодження, що ухвалені в програмній документації

Було погоджено, що для створення загального програмного забезпечення та його злагодженої роботи буде використовуватися допоміжний фреймворк – Laravel.

Межі проєкту ПЗ

Крайня дата завершення роботи над програмним забезпеченням – 00.00.2022р.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Дане програмне забезпечення не має обмежень у його використанні.

Характеристики користувачів

Основні характеристики користувачів: наявність підключення до мережі Інтернет та ноутбук або персональний комп’ютер.

Загальна структура і склад системи

Основні частини для створення програмного забезпечення: база даних; вебзастосунок; API (Application programming interface).

Загальні обмеження

Єдине обмеження для роботи з програмним застосунком – це наявність Інтернету.

ФУНКЦІЇ ПРОГРАМНОГО ЗАСТОСУНКУ ПЕРЕКЛАДУ СЛІВ

Функція перекладу слова

Опис функції

Функція перекладу допомагає перекласти слово або текст обсягом 255 символів з трьох мов.

Вхідна і вихідна інформація

Вхідна інформація: введене слово або текст;

вихідна інформація: перекладений текст и слово на вибрану мову.

Функціональні вимоги

База даних із словами та стабільне підключення до мережі інтернет.

Функція реєстрації/авторизації користувача.

Опис функції

Дана функція дозволяє користувачеві зареєструватися та надалі при необхідності авторизовуватися у застосунку.

Вхідна і вихідна інформація

Вхідна інформація: дані про користувача, а саме пошта та пароль.

Вихідна інформація: доступ до застосунку.

Функціональні вимоги

База даних, доступ до мережі інтернет та пошти.

Функції розділу з граматичними файлами

Опис функції

Дана функція дає користувачеві доступ до файлів з граматикою на двох мовах та можливість скачати ці файли на свій комп’ютер чи ноутбук.

Вхідна і вихідна інформація

Вхідна інформація: файли з необхідною інформацією;

Вихідна інформація: перегляд файлів.

Функціональні вимоги

База даних, доступ до мережі інтернет.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

В даному програмному застосунку основним джерелом вхідної інформації являється сам користувач. Він вручну задає потрібні данні, в нашому випадку це обрані користувачем слова або текст.

Нормативно-довідкова інформація (класифікатори, довідники тощо)

Немає вимог до даного пункту.

Вимоги до способів організації, збереження та ведення інформації

Обмін даними повинен відбуватися використовуючи REST API. В якості бази даних для програмного застосунку обрати MySQL.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Серйозних технічних вимог немає. Комп’ютер чи ноутбук з будь-яким процесором, наприклад INTEL Core i7-3740QR, такий процесор підтримує

необхідні для розробки програми.

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмної системи

Архітектура програмного забезпечення проста, вона складається з клієнтської частини та БД.

Системне програмне забезпечення

Застосунок має бути побудований за допомогою фреймворку Laravel. Для написання фронтенду обрати технологію vue.js, для бекенду використовувати – php. Обмін даними повинен відбуватися використовуючи REST API. В якості бази даних для програмного забезпечення обрати MySQL.

Мережне програмне забезпечення

Для створення ПЗ знадобиться будь-яка операційна система, HTML-редактори: JetBrains PhpStorm, Visual Studio Code та Node.js; Google Chrome браузер – для зручного перегляду вебсторінок.

Програмне забезпечення ведення інформаційної бази

Все ведення інформаційної бази має відбуватися через БД MySQL.

Мова і технологія розробки ПЗ

Програмне забезпечення повинно бути розроблене за допомогою фреймворку Laravel. Мови розробки – Java Script та Php.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Вебінтерфейс повинен відповідати усім вимогам UX/UI, це дозволить користувачеві не витрачати багато часу на те щоб зрозуміти як працює система. Шаблон вебзастосунка складається з трьох вкладок, де перша містить у собі вікно для перекладу, друга – розділ у якому розміщені граматичні файли, третя – це вкладка для перегляду історії скачаних файлів.

Апаратний інтерфейс

Апаратний інтерфейс – це девайс користувача, яким він буде користуватися (персональний комп’ютер, ноутбук).

Програмний інтерфейс

Laravel – безкоштовний php-фреймворк з відкритим вихідним кодом, розроблений для виготовлення складний сайтів.

Комунікаційний протокол

Використання мережних протоколів Wireless Application Protocol — протокол безпроводового доступу та TCP/IP.

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

Будь-який користувач який забажає користуватись цим програмним застосунком зможе це зробити.

Супроводжуваність

Супроводження – не передбачене у даному застосунку.

Переносимість

Програмне забезпечення може працювати на будь-якій операційній системі.

Продуктивність

Продуктивність залежить від швидкості мережі Інтернет. Продуктивність застосунка повинна контролюватися часом виконання запитів, що не повинні перевищувати 2.5 секунд.

Надійність

Усі дані, які вказуються користувачем повинні бути приватними і виключити можливості їх отримання будь якими способам для інших користувачів. Кожний користувач може отримати доступ до своїх даних лише авторизуватися у системі.

Безпека

Реєстрація та вхід до кабінету користувача – авторизація повинна відбуватися з токеном.

ІНШІ ВИМОГИ

Усі вимоги описані вище.

Висновки до розділу 1

Отже, в першому розділі проведено огляд та аналіз існуючих систем для перекладу слів. Розглянуті застосунки що містять в собі форми для перекладу слів

тощо. Також розглянуті мобільні та десктопні застосунки де користувач має доступ до більш широкого функціоналу. Визначені основні переваги та недоліки оглянутих застосунків. Визначено, що більшість являються комерційними розробками і недоступні ні у вигляді програмного коду, ні у вигляді формального опису алгоритмів. Сформульовані задачі досліджень дипломної роботи та основні вимоги щодо створення програмного забезпечення.

2 АНАЛІЗ ЗАСОСОБІВ ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАСТОСУНКУ

2.1 Етапи розробки вебзастосунків

Розробка будь-якого програмного застосунку потребує чітких та послідовних дій. Це стосується і розробки вебзастосунків. Існує певна структура якої необхідно дотримуватися під час створення системи.

Етапи розробки вебзастосунку складаються з наступних кроків:

- розробка технічного завдання;
- створення дизайну;
- верстка;
- програмування;
- тестування.

Дотримуючись вищеперерахованих кроків, можна структурувати та полегшити процес розробки свого вебзастосунку. Далі розглянемо кожен етап розробки вебзастосунку окремо.

Розробка технічного завдання – якщо це стосується розробки комерційного проекту, де розробник працює з справжнім замовником, технічне завдання це документ, який містить у собі загальну інформацію про компанію замовника, а також вимоги до структури, зовнішнього вигляду, оформлення та наповнення майбутнього сайту. У нашому випадку це вимоги до майбутнього програмного застосунку, його структури, наповнення та зовнішнього вигляду. Це початковий етап розробки вебзастосунку, під час якого визначаються основні складові майбутньої системи. Цей етап являється не менш важливим, адже від правильно

поставлених завдань, залежить те як буде виглядати та працювати майбутній вебзастосунок.

Створення дизайну – це сукупність графічних елементів шрифтів кольорів що будуть розміщенні на сайті. Дизайн сайту це його зовнішнє оформлення, яке має не відлякати користувача, а навпаки змусити його залишитись на сторінці та почати використовувати застосунок. Це не менш важливий крок у створенні вебзастосунку. На даному етапі розробляється зовнішній вигляд сторінки. Обираються кольори, шрифти та загальний стиль усіх елементів які будуть наповнювати вебзастосунок. Як правило на даному етапі також виготовляється мокап вебсторінки. Роль мокапа продемонструвати проект таким як він має виглядати у кінцевому результаті.

Верстка – етап розробки під час якого створений дизайн перетворюється у програмний код. Іншими словами це з'єднання та розміщення на сторінці різних елементів вебзастосунку: текстових блоків, зображень, відео тощо[10]. Від верстки залежить коректність зображень в браузері, зовнішній вигляд сторінки на різних пристроях, або іншими словами адаптивність. Розробники перетворюють макети дизайну у інтерактивні вебсторінки.

Програмування – на даному етапі, після того як розробники інтерфейсу перетворили макети дизайну у інтерактивні вебсторінки, ці сторінки інтегрують з базою даних тобто перетворюють сайт у повноцінний функціонуючий інструмент, а не просто закодоване зображення.

Тестування – один із зарешальних етапів у розробці вебзвастосунку. Готовий застосунок перевіряється на відповідність макету дизайну, продуктивності сумісності з браузером. Це фінальна перевірка якості, оцінка готового результату як з точки зору розробника, так і зі стороні користувача. Перевіряється функціональність застосунку. Оцінюється загальна ефективність проекту.

2.2 Вибір програмних середовищ для розробки застосунків

Розробка застосунку потребує ретельного підходу. Вибір технологій, підбір програмного середовища – це важливі пункти від яких залежить уся подальша робота розроблюваного застосунку.

При виборі середовища для розробки програмного застосунку слід звернути увагу на наступних декілька факторів[6]:

- Складність процесу проєктування програмного забезпечення для конкретного середовища;
- наявність інструментальних засобів розроблення програмного забезпечення;
- можливість внесення поправок у програму;
- наявність засобів проектування інтерфейсу користувача;
- швидкість виконання та надійність роботи програми;

Відштовхуючись від вищеперерахованих факторів розробники повинні обирати найбільш підходяще по усім критеріям середовище для створення власного застосунку.

У якості програмного середовища для розробки застосунку було вирішено обрати VScode (рис. 2.1). Це безкоштовний редактор коду від компанії Microsoft для Windows, MacOS та Linux, який дає можливість розробляти будь-який вид застосунків (зокрема і вебзастосунків) на будь-якій мові програмування.

Рисунок 2.1 – Логотип Visual Studio Code

Дане програмне середовище підходить для розробки нашого програмного застосунку перекладу слів.

2.3 Аналіз та вибір фреймворку

Під час створення вебзастосунку чи сайту перед розробником завжди постає вибір: CMS чи фреймворк. CMS – (Content management system) це система що використовується для забезпечення організації процесу створення, редагування та управління контентом сайту. Фреймворк – це своєрідний каркас або шаблон для побудови сайту який можна доповнювати власним кодом; відповідає за обробку помилок, роботу з базами даних, захист паролем тощо. Фреймворк на відміну від CMS більш гнучкий та продуктивний.

Програмний застосунок для перекладу слів вимагає індивідуального підходу у розробці. Тому для його створення буде використано фреймворк.

Але у фреймворка як і інших способів розробки є свої переваги та недоліки:

До основних переваг фреймворку можна віднести:

– зручність написання коду;

– програмні рішення написані за допомогою фреймворків як правило витримують більше навантаження;

– спрощує супроводжування проєкту;

Недоліки фреймворків:

– фрейморки задають структуру майбутнього проєкту, тому для вирішення не стандартних задач не підійдуть;

– складність у вивченні і кожен фреймворк окрема система. Тому при переході від одного до іншого можуть виникнути складнощі у опануванні.

Для написання застосунку було обрану мову PHP. Далі наведена порівняльна таблиця 2.1 найбільш популярних фреймворків на мові PHP[7]. Таблиця була зроблена з метою вибору найбільш підходящого фреймворку для розробки програмного застосунку перекладу слів.

Таблиця 2.1 – Порівняння фреймворків на мові php

Критерії	**Laravel**	**Yii**	**Zend**
Останній стабільний реліз	Laravel 5.8	Yii 2	Zend 3.0.0
Опис	Безкоштовний php-фреймворк з відкритим вихідним кодом. Має багато різноманітних складових що дозволяють полегшити розробку.	Безкоштовний об’єктно - орієнотований фреймворк. Завдяки своїм складовим компонентам підходить для розробки великомасштабних вебзастосунків.	Вільний об’єктно-орієнтований php-фреймворк який підходить для розробки великих комерційних проєктів
Переваги	Має величезну кількість інструментів.	Гнучкий можна використовувати для побудови різних типів вебзвстосунків Вважається найшвидшим фреймфорком.	Гнучкий механізм кешування та піддтримка великої кількості баз даних.
Шаблонізатори	Blade – дозволяє використовувати код в представленнях PHP.	Не використовує шаблонізатори	Не мітсить

Ккінець таблиці 2.1

Використовувані бази даних	MySQL, Postgres, SQLite, і SQL Server.	MySQL, PostgreSQL, SQLite 2 і 3, Microsoft SQL Server 2008, Oracle, CUBRID, Sphinx.	MariaDB, MySQL, Oracle Database, IBM DB2, Microsoft SQL Server, PostgreSQL.

На основі попереднього аналізу представлених фреймворків, для розробки програмного застосунку перекладу слів було обрано Laravel.

Laravel – php фреймворк з відкритим вихідним кодом, створений спеціально для розробки складних вебсайтів та вебзастосунків[19]. Даний фреймворк зручний у використанні, має гнучку систему маршрутизації. Містить у собі зручний механізм обробки помилок та виключень. Також вагомим плюсом є те що фреймворк має вбудований механізм авторизації та аутентифікації що спрощує роботу з цими елементами. До того ж у Laravel (рис 2.2) є вбудовані бібліотеки і модулі, кожен модуль інтегрований з менеджером Composer що спрощує оновлення.

Рисунок 2.2 – Логотип Laravel

Обраний фреймворк містить як плюси так і мінуси. Розглянемо переваги фреймворку Laravel:

– підвищена безпека;

– вбудована аутентифікація користувачів через форму;

– відритий код. Це дозволяє будь-якому розробнику вносити зміни у свій проект під час його створення;

– містить шаблоназітор blade завдяки якому можна багато разів використовувати один і той же шаблон в різних частинах застосунку;

– міграції баз даних. Розробник може змінювати структуру своєї бази даних та відійти на крок назад у разі помилки або якщо щось необхідно змінити;

– вбудована можливість обробки помилок або виключень. Система знаходить помилку та повідомляє про це користувача.

– швидкість розробки проєкту;

– найбільш підходящий для розробки вебзастосунків;

– різноманіття готових пакетів з можливістю пілаштування під потреби нашого застосунку;

– висока швидкість завантаження сторінок;

– можливість самостійно вибрати спосіб зберігання об’єктів (доступний широкий ряд технологій);

– консоль artisan;

– використання ORM, технології для зв’язку бази даних та мови програмування що дозволяє пришвидшити роботу;

Розглянувши даний фреймворк, проаналізувавши його переваги та недоліки, порівнявши з іншими популярними технологіями можна сказати що даний фреймворк є найбільш підходящим для нашого програмного застосунку перекладу слів та задовольняє усі потреби при розробці даного застосунку.

2.4 Архітектурний шаблон

У основі фреймворку Laravel який ми обрали для розробки нашого програмного застосунку лежить структура MVC яка ізолює код та відділяє компоненти один від одного.

Рисунок 2.3 – Вигляд MVC

Шаблон проєктування у даному фреймворку – MVC (Model View Controller) (рис. 2.3).

Розглянемо більш детально кожну складову цієї архітектури. Модель (Model) – компонент який відповідає за обробку та верифікацію даних, звернення до бази даних. Тобто завдяки їй можна додавати записи до бази даних, видаляти переглядати їх або змінювати . Задача моделі – доставити дані користувачеві. Вона реагує на команду контролера. Не взаємодіє на пряму з користувачем.

Представлення (View) відповідає за візуалізацію інформації яку він отримує від моделі. Відображає дані на рівні користувацького інтерфейсу. Також представлення визначає зовнішній вигляд інтерфейсу та способи взаємодії з ним. Тобто представлення потрібне для того щоб користувачу було зрозуміло та зручно використовувати застосунок.

Контролер (controller) забезпечує взаємодію з системою. Він обробляє дії користувача, перевіряє отриману інформацію та передає її моделі. Контролер являється точкою входу для програми.

2.5 Вибір бази даних

Також невід’ємною частиною вебзастосунку є база даних, яка необхідна нам для зберігання інформації та подальшої роботи застосунку.

База даних- це набір даних що зберігається у впорядкованому вигляді. Так як база даних є невід’ємною частиною застосунку, нам необхідно обрати систему управління базами даних. У таблиці 2.2 представлені деякі системи управління базами даних і їх порівняння.

Таблиця 2.2 – Порівняння систем управління базами даних

Критерії	**MySQL**	**Oracle**	**SQLite**
Опис	Найпопулярніша БД для вебзастосунків. Швидка та надійна.	Використовується для хмарних середовищ. Дозволяє керувати БД з мільйонами записів. [8]	Потужна вбудована система керування.
Переваги	Проста у роботі, широкий функціонал, безпечна. Має багато функцій	Не потребує великих об’ємів пам’яті для кешу.	Повна база даних міститься в одному файлі,
Недоліки	Складна і не стандартна система безпеки. Не підтримує гаряче резервне копіювання.	Дорога в обслуговуванні	Відсутність системи користувачів, відсутність покращення продуктивності

Після проведення аналізу систем управління базами даних було прийнято рішення обрати MySQL.

MySQL одна з найстарших реляційних баз даних які використовується для вебзастосунків. Підтримує мову програмування SQL(structured query language). Модель роботи клієнт-сервер – це коли декілька комп’ютерів клієнтів відправляють запити та отримують відповіді від централізованої службової машини тобто сервера, яку також можуть називати хост-системою або хостом. Система керування базами даних безкоштовна і підійде для створення невеликих за обсягом проектів. Також є платні доповнення для комерційних проєктів. Так як

ця СКБД містить у собі переважну кількість плюсів, аніж мінусів і по усім критеріям підходить для управління базою даних нашого застосунку для перекладу слів.

Детальніше розглянемо переваги даної СКБД[22]:

– висока швидкість роботи;

– призначена для роботи у мережі інтернет але незважаючи на це має добре розвинену систему захисту від несанкціонованого доступу;

– сервер дозволяє одночасно підключатися необмеженій кількості користувачів;

– система являється гнучкою. Легко налаштовується під потреби того чи іншого сайту;

– MySQL працює на операційних системах різноманітних систем;

– дана система являється безкоштовною;

– відкритий код системи завдяки чому розробник може відредагувати власний SQL-сервер;

– система забезпечує достатньо високий рівень безпеки при зберіганні та передачі даних;

– підтримка декількох запитів одночасно;

Представлених переваг більше ніж достатньо для того щоб обрати саме цю систему управління базами даних нашого застосунку.

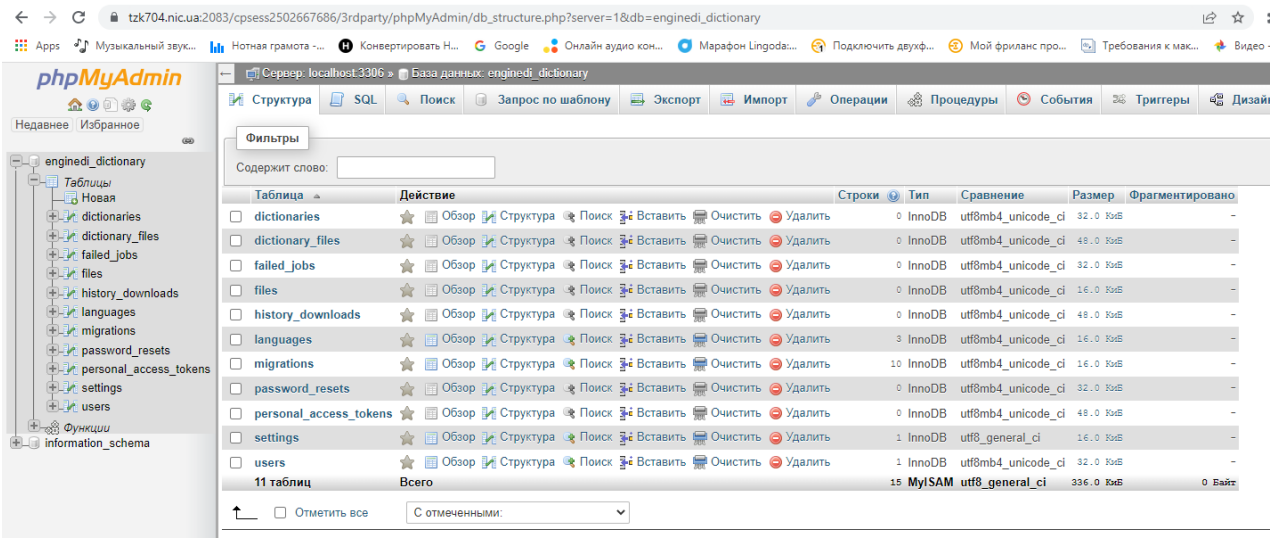

Рисунок 2.4 – PhpMyAdmin

Для зручності у роботі використовується застосунок PhpMyAdmin [21], на основі мови програмування PHP, з відкритим вихідним кодом. Він надає вебінтерфейс для керування базами даних MySQL що робить керування більш зручним.

2.6 Вибір технологій для створення інтерфейсу

Зовнішній вигляд інтерфейсу також являється важливою складовою застосунку через яку користувач буде взаємодіяти з системою. Візуальна частина програмного застосунку відіграє значну роль у знайомстві з системою. Адже важлива не тільки швидка робота застосунку, але й приємний зовнішній вигляд, не нав'язливий дизайн.

У першу чергу користувач оцінює зовнішній вигляд інтерфейсу, а потім робить висновок із побаченого та обирає користуватись даним програмним застосунком чи ні.

У наш час представлений широкий вибір мов програмування та технологій для написання фронтендної частини застосунку.

Фронтенд – це частина у веброзробці яка відповідає за взаємодію користувача з системою, спрощує прийняття та розуміння розміщеної інформації на сайті[18]. На наступній таблиці 2.3 ми проаналізуємо та порівняємо декілька найпоширеніших технологій фронтенду.

Таблиця 2.3 – Порівняльна характеристика технологій фронтенду

Критерії	**Vue**	**Agular**	**JQuery**
Опис	Платформа для створення користувацьких інтерфейсів	Фреймворк з відкритим вихідним кодом використовується для розробки односторінкових вебстайтів[17]	Бібліотека що використовується для взаємодії JavaScript та HTML.

Кінець таблиці 2.3

Переваги	Простий у використанні. Також є можливість інтегрувати його у інші фреймворки.	Легко розробляти компоненти та модулі, спрощена двухстороння прив’язка даних. Підходить для розробки невеликих за обсягом односторінкових сайтів.	Допомагає швидко відлагодити та протестувати застосунок, оптимізує сучасний код під застарілі браузери.
Недоліки	Проблеми із стабільністю, не підходить для великих проектів	Складний у розумінні і використанні, не гнучкий Дуже низька продуктивність. Не має сумісності між версіями.	Застарілий API, не має рівня даних, не підходить для великомасштабних проєктів.

У результаті проведеного аналізу технологій які знаходяться у таблиці 2.3, порівняння їх переваг та недоліків було вирішено обрати Vue js. Саме ця технологія найкраще підходить для розробки інтерфейсу нашого програмного застосунку перекладу слів.

Vue js – це фреймворк створений для розробки користувацьких інтерфейсів на мові JavaScript [15]. Вирішує різні задачі рівня представлення, спрощує роботу з іншими бібліотеками. Дозволяє розділяти код на компоненти та збирати їх у єдиний застосунок. Компоненти можна використовувати у інших застосунках.

Розглянемо переваги даної технології:

– проста у використанні, можна кодувати всього у декілька рядків з мінімальними зусиллями;

– є можливість інтеграції у сторонні фреймворки, це дозволяє налаштовувати у відповідності з потребами;

– дружній інтерфейс;

– легкий у осовоєнні;

– гнучкий завдяки чому можна створювати власні шаблони з віртуальними вузлами;

– дозволяє розробляти великомасштабні шаблони для багатократного використання;

– двухстороння прив’язка даних, описує зв’язок між оновленнями та представленням, тобто будь-які зміни користувацького інтерфейсу внесені користувачем відображаються у даних та навпаки будь-які зміни даних, відображаються у користувацькому інтерфейсі[9];

– велика кількість супутньої документації;

– підходить для розробки маленьких за об’ємом, не комерційних проектів;

Рисунок 2.5 – Логотип Vue js

Розглянута технологія підходить для оформлення візуальної частини нашого програмного застосунку перекладу слів. Дана технологія найчастіше використовується як раз з тим фреймворком який ми обрали для розробки нашого програмного застосунку перекладу слів. Фреймворк Laravel. Окрім переваг які

були описані у списку вище, Vue js також забезпечує однорідність. А однорідність у свою чергу забезпечує розробку сторінок застосунку у одному стилі де усі елементи будуть поєднуватися між собою, а загальна картинка буде виглядати цілісно та приємна для відвідувачів програмного застосунку. Адже як показує практика користувач завжди віддасть перевагу сторінці яка оформлена у одному стилі і буде приємна для його ока. Не буде перевантажена, а дизайн буде у спокійних кольорах.

Висновки до розділу 2

Отже, у другому розділі було визначено етапи розробки вебзастосунків. Розглянуто кожний етап окремо. Усього існує п’ять етапів розробки вебзастосунків: розробка технічного завдання, створення дизайну, верстка, програмування, тестування. Дані етапи були описані більш детально кожен.

Також був проведений аналіз та вибір технологій які будуть використовуватися для розробки програмного застосунку перекладу слів.

У результаті проведеного аналізу трьох різних фреймворків мови PHP порівняння усіх плюсів і мінусів представлених фреймворків було вирішено обрати фреймворк Laravel.

Для управління базою даних, також після детального аналізу усіх систем контролю даних, порівняння переваг і недоліків кожної окремо, було обрано СКБД MySQL.

Після аналізу найпопулярніших існуючих технологій для розробки інтерфейсу застосунку, порівняння їх позитивних сторін та негативних було вирішено що для оформлення користувацького інтерфейсу нашого програмного застосунку, найбільше підійде – технологія Vue js. У якості програмного середовища для розробки програмного застосунку ми обрали редактор коду VScode.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ ПЕРЕКЛАДУ СЛІВ

3.1 Алгоритм роботи модулю

Модуль прекладу слів максимально простий та зручний у використанні. Має вигляд вебзатосунку. Даний застосунок доступний для будь-якого бажаючого користувача. Користуватись застосунком можна через персональний комп’ютер або ноутбук.

Модуль призначений для перекладу слів та текстів. Основні функції модулю:

– введення тексту і слів;
– переклад введеної інформації;
– виведення перекладеної інформації;

Блок-схема – часто використовуваний тип графічних моделей, який описує алгоритми, де окремий крок представлений у вигляді блоків різної форми які з’єднані лініями між собою, що вказують напрямок послідовності. Зручність цього типу діаграм полягає у тому що, кожна фігура має своє значення дії. Ромб – означає питання, прямокутник – відображення певної дії, паралелепіпед – введення або виведення тощо.

Блок-схема допоможе схематично представити роботу модуля. Також слід зауважити що програмний застосунок є безкоштовним і ним без проблем зможе користуватись абсолютно кожен бажаючий користувач.

Перший крок – обираємо мову тексту з якого потрібно потрібно перекласти та мову для перекладу.

Другий крок – вводимо слово або текст у спеціальне віконечко які ми хочемо перекласти.

Третій крок – для того щоб текст переклався нам не потрібно натискати ніякі кнопки. Прибираємо курсор мишки з вікна де знаходиться текст, що перекладається, і у протилежному вікні з’являється результат з перекладеним текстом.

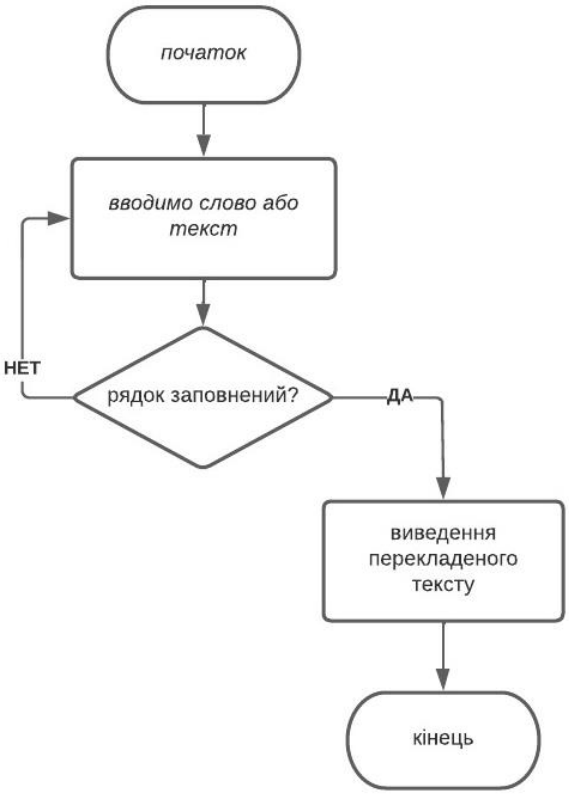

Рисунок 3.1 – Блок-схема алгоритму роботи модулю

Найчастіше блок-схема будується під час етапу розробки програмного забезпечення. Бувають і випадки коли програмування не можливе без попереднього зображення блок-схеми. Маючи перед собою схематичне зображення функціонування програмного забезпечення, буде легше у процесі його розробки.

3.2 Сценарії використання програмного застосунку перекладу слів

Сценарій використання потрібен для того щоб показати як користувач буде взаємодіяти з програмним забезпеченням у реальному світі. Тобто іншими словами – це опис поведінки системи. Методика сценаріїв використання потрібна для того щоб виявити так звані користувацькі або функціональні вимоги.

У програмному застосунку що проєктується існує дві ролі: гість та зареєстрований користувач. На наступних таблицях можна побачити приклади використання програмного застосунку перекладу слів.

Таблиця 3.1 – Сценарій №1: Реєстрація

Діючі особи	Гість, застосунок
Мета	Створення власного облікового запису
Передумова	Користувач не авторизований.
Успішний сценарій: 1. Гість переходить до форми реєстрації. 2. Заповнює необхідні рядки зі своїми даними. 3. Система здійснює валідацію даних. 4. Система зберігає обліковий запис. 5. Система переводить користувача на головну сторінку автоматично.	
Результат	Створено новий обліковий запис.
Розширення	
***а**	Не всі поля при реєстрації заповнені. Рамка вікна стає червоного кольору. **Результат:** користувач не може зареєструватись.
1а	Гість ввів вже існуючі дані. Система виводить повідомлення. **Результат:** Гість не може зареєструватись.

Таблиця 3.2 – Сценарій №2: Авторизація

Діючі особи	Користувач , застосунок
Мета	Вхід до облікового запису
Передумова	Користувач не авторизований.
Успішний сценарій: 1. Користувач переходить до форми авторизації. 2. Користувач вводить необхідні дані для авторизації(пошта, пароль) 3. Система здійснює валідацію даних. 4. Користувач увійшов до сиситеми.	

Кінець таблиці 3.2

Результат	Користувач авторизований.
Розширення	
***а**	Не знайдено користувача з введеною поштою. Вікно стає червоним. **Результат:** користувач не авторизований.
1а	Користувач ввів не правильний пароль. Система виводить відповідне повідомлення. **Результат:** користувач не авторизований.
2а	Користувач залишив поле для даних пустим. Система виведе відповідне повідомлення. **Результат:** Користувач не авторизований.

Таблиця 3.3 – Сценарій №3: Перегляд та скачування додаткових файлів

Діючі особи	Гість, застосунок
Мета	Скачування файлу
Передумова	Користувач авторизований.
Успішний сценарій: 1. Користувач переходить у вкладку Dictionaries. 2. Обирає необхідний розділ. 3. Система виводить на екран текст. 4. Користувач натискає кнопку скачати.	
Результат	Користувач скачав файл
Розширення	
***а**	Не має доступу до бази даних. **Результат:** користувач не може скачати файл.
1а	Не має підключення до інтернету. **Результат:** користувач не може скачати файл.

Для розуміння можливостей кожного з учасників сценарію, необхідно створити діаграму використання. Діаграма використання або діагарма прецедентів – це діаграма на якій зображується відношення між акторами та прецедентами і системі. Зображується як графічне зображення взаємодій між різними елементами в системі.

Діаграма описує зазвичай події та те як вони протікають. На рисунку 3.2 зображена діаграма прецедентів програмного застосунку перекладу слів який розробляється.

Як вище було вказано основними дійовими особами являються є гість(неавторизований користувач) та користувач.

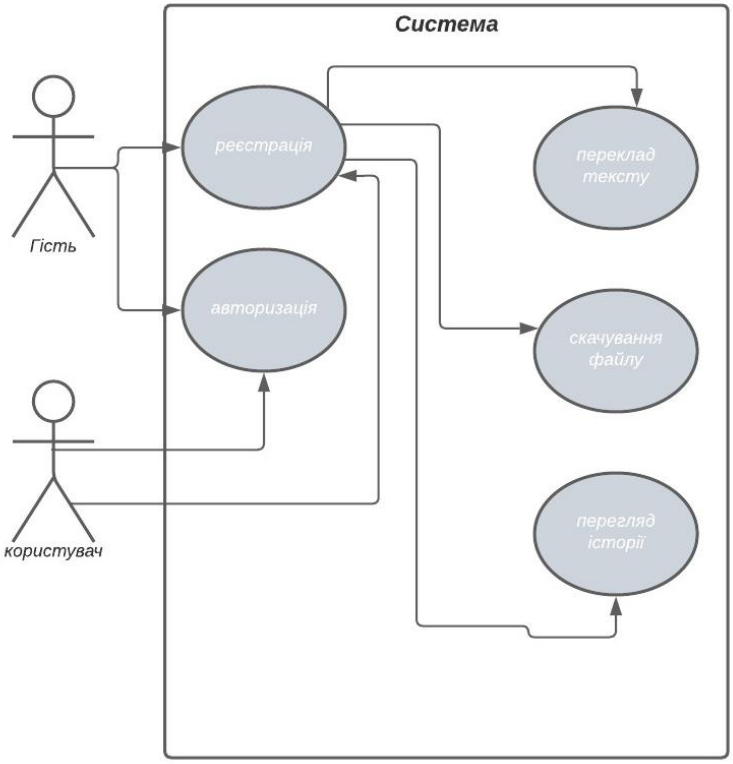

Рисунок 3.2 – Діаграма прецедентів

3.3 Проєктування програмного застосунку перекладу слів

Для візуального представлення користувацького інтерфейсу було створено mock-up. Це один з видів представлення інтерфейсу(детальніше у розділі 2).

На рисунку 3.3 зображений інтерфейс форми реєстрації. Відкриваючи застосунок перед нами з’являється логотип застосунку та у правому верхньому

кутку дві кнопки реєстрації та авторизації. Коли гість натискає кнопку log in перед ним з'являється наступне вікно. Доступ до цієї форми має як гість так і зареєстрований користувач. Для реєстрації користувач має вказати своє ім'я, електронну пошту, придумати пароль та повторного його ввести для підтвердження.

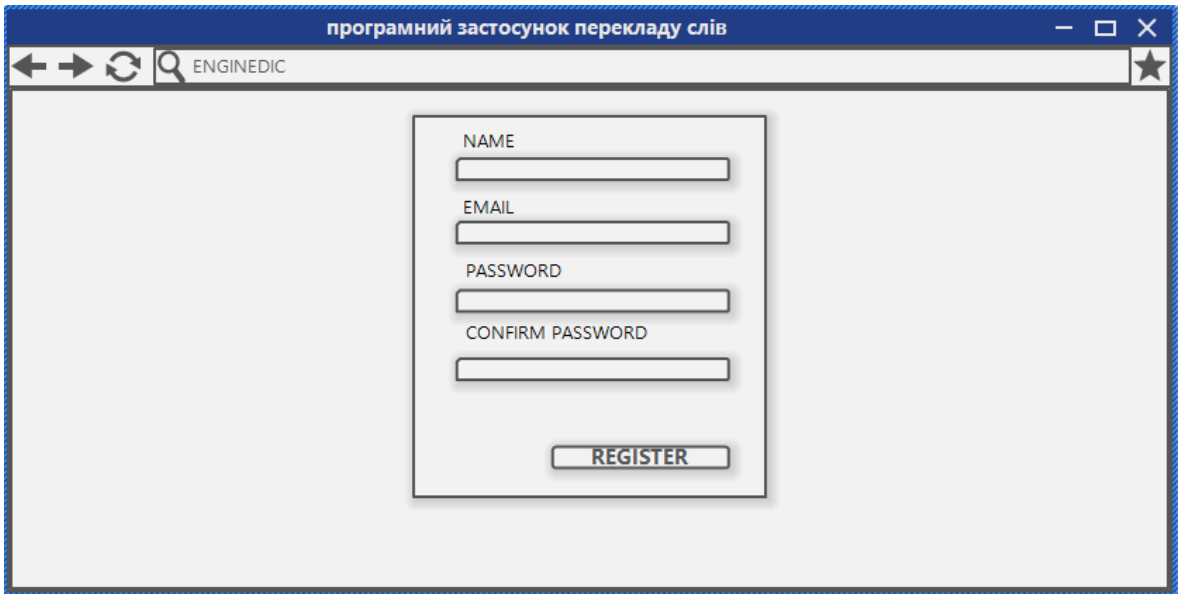

Рисунок 3.3 – Форма реєстрації

На рисунку 3.4 зображена форма для авторизації. Дана форма доступна для зареєстрованого користувача. Форма авторизації максимально проста. Для входу у застосунок, користувач має заповнити всього два поля: електронна пошта, пароль та натиснути кнопку log in. Так користувач потрапляє на головну сторінку нашого програмного застосунку перекладу слів.

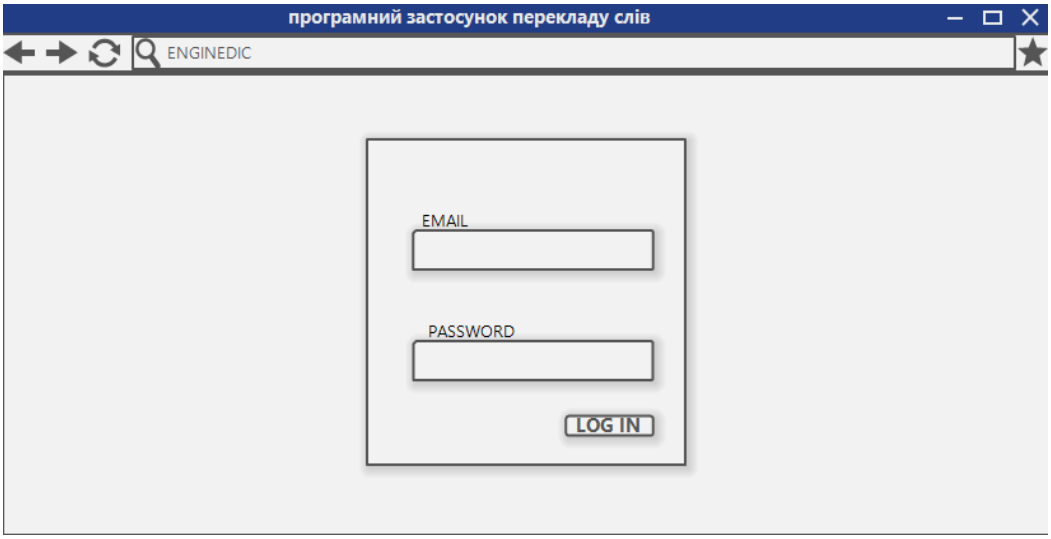

Рисунок 3.4 – Форма авторизації

Після того як користувач авторизувався він потрапляє на сторінку або розділ нашого застосунку під назвою Dashboard. У даній частині знаходяться два вікна для перекладу слів або тексту. Кнопка для зміни мови місцями та кнопки для вибору мови над кожним вікном. Також на цій сторінці ми можемо бачити ще дві вкладки під назвами Dictionaries та History.

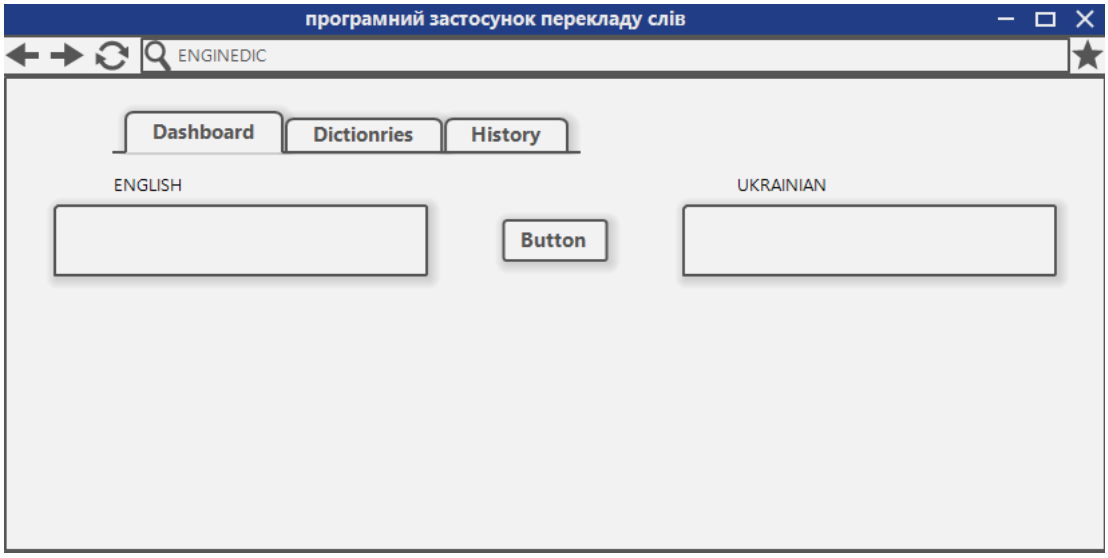

Рисунок 3.5 – Dashboard

Також у даному застосунку є частина або вкладка Dictionaries. Даний розділ мітить у собі матеріали з граматики трьох мов: українська, іспанська та англійська. Потрапляючи на цю сторінку ми бачимо перед собою три розділи які називаються відповідно до мови якій вони належать. Натискаючи кожен розділ перед нами постають підрозділи. Їх кількість у кожному розділі своя. Коли ми переходимо до будь-якого підрозділу, висвічується текст, тобто інформація по граматиці тієї мови яку ми обрали, одна з граматичних тем. Також є можливість скачати файл з вмістом даного підрозділу, натиснувши на посилання для скачування яке знаходиться нижче.

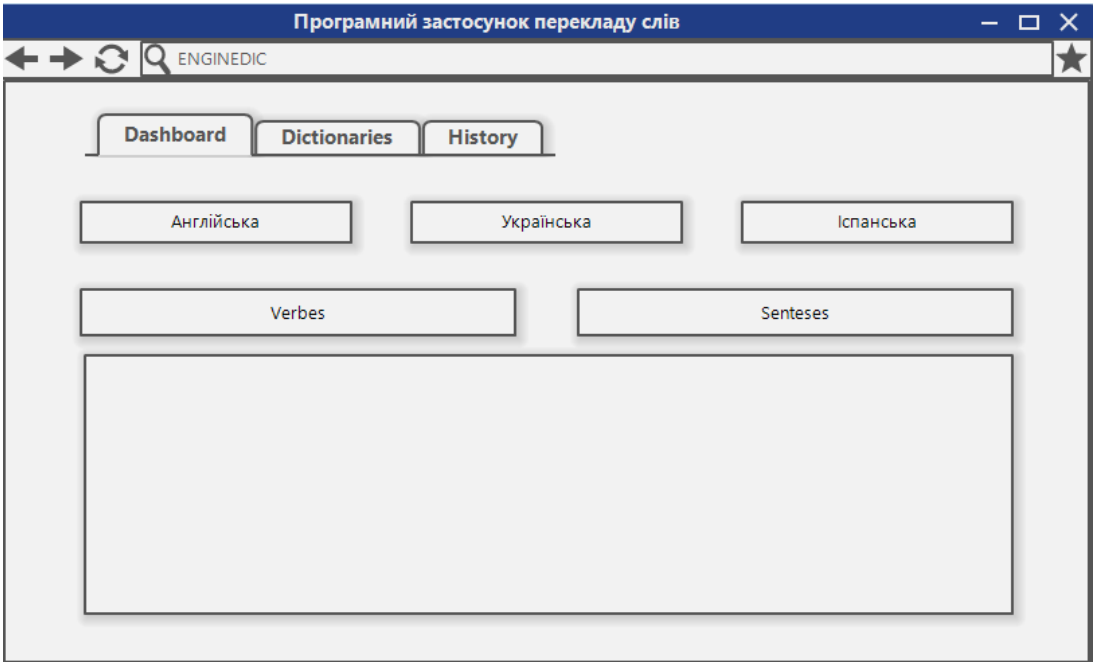

Рисунок 3.6 – Dictionaries

Третя вкладка нашого програмного застосунку називається History. Дана вкладка зроблена для зручності користувача, бо завжди можна передивитись які файли були скачані та у який час. Як уже можна було зрозуміти у даному розділі відображається інформація про дату, час та скачані користувачем файли.

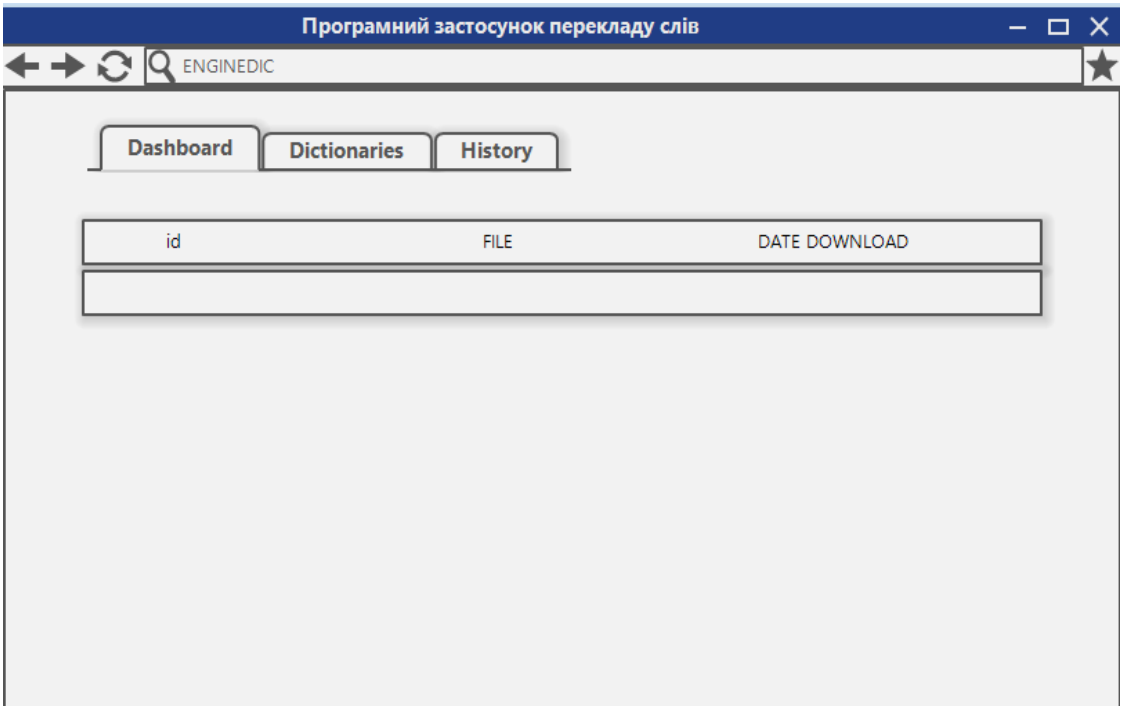

Рисунок 3.7 – History

3.4 Діаграма класів

Діаграма класів – це UML діаграма яка описує структуру застосунку з точки зору класів, характеристик, методів та взаємодію між ними. Дана діаграма є частиною мови UML.

Таке моделювання необхідно для розуміння системи.

UML (Unified Modeling Language) – це уніфікована мова моделювання. Використовується для графічного представлення опису та моделювання об’єктів у розробці програмного забезпечення, вебзастосунків, інформаційних систем, бізнес-процесів тощо[11]. У собі ця мова містить:

- класи;
- їх атрибути;
- методи;
- взаємозв'язки між об'єктами;

На рисунку 3.8 зображена діаграма класів програмного застосунку перекладу слів.

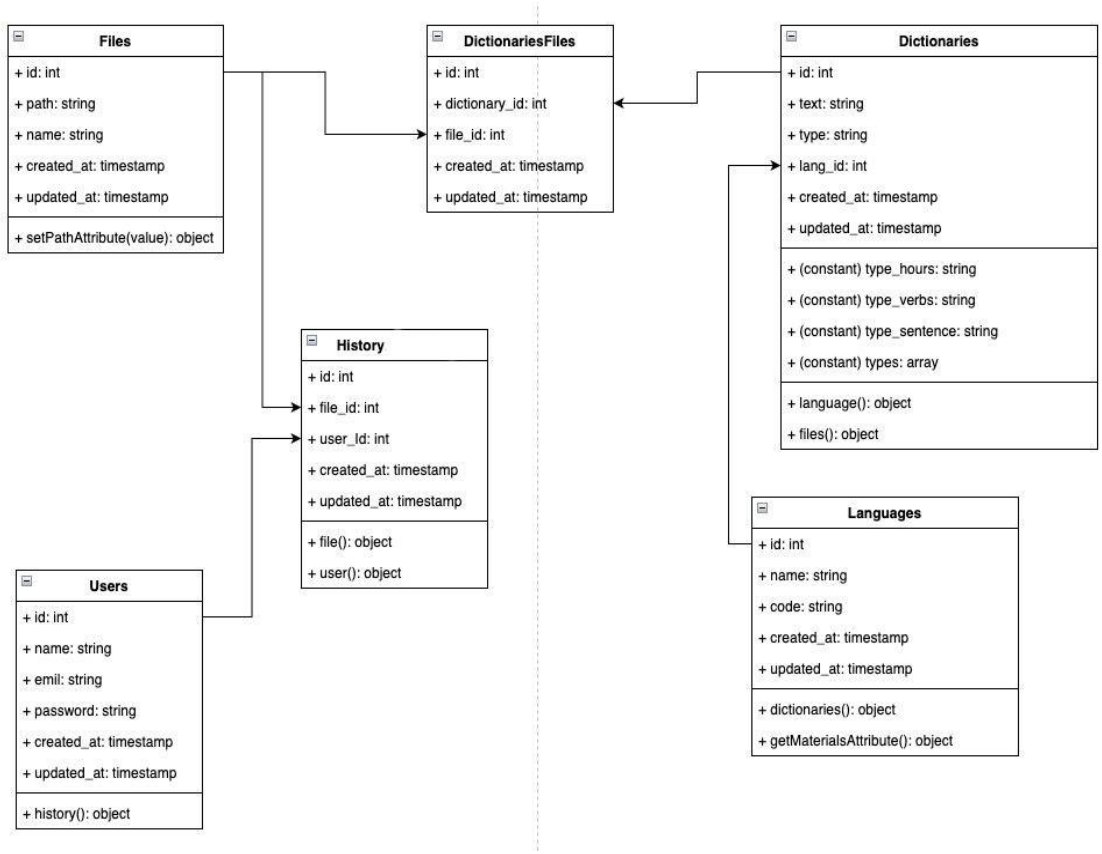

Рисунок 3.8 – Діаграма класів

До класів застосунку належать:

– Files;
– DictionariesFiles;
– Dictionaries;
– History;
– Users;
– Languages.

Таблиця 3.4 – Класи програмного застосунку перекладу слів

Клас	Атрибути	Методи
Files	id	setPathAttribute() виповнює роль завантаження и зберігання файлів
	path	
	name	
	created_at	
	updated_at	
DictionariesFiles	Id	–
	dictionary_id	
	file_id	
	created_at	
	updated_at	
Dictionaries	id	laungauge() повертає дані із сутності language()
	text	
	type	
	lang_id	files() повертає список файлів
	created_at	
	updated_at	
History	id	file() повертає дані із сутності files
	file_id	
	updated_id	

Кінець таблиці 3.4

	created_id	user() повертає дані з сутності users
	user_id	
Users	id	history() повертає залежність із таблиці HistoryDownload
	name	
	email	
	password	
	created_id	
	updated_id	
Languages	id	dictionaries() повертає словники по конкретній мові
	name	
	code	
	created_id	getMaterialsAttribute() виповнює роль сортування для словника по типу конкретної мови
	updated_id	

3.5 Проєктування бази даних

Для того щоб спроєктувати базу даних програмного застосунку нам необхідно створити ERD-діаграму. ERD-діаграма (Entity Relational Diagram) – це діаграма сутність-зв'язок, використовують для створення моделей даних. Вона надає послідовні засоби визначення даних і зв’язків[20]. Іншими словами це візуальне представлення бази даних. На рисунку 3.9 представлена ERD застосунку який проєктується.

Сутність Dictionaries містить у собі змінні: id – номер таблиці, text – містить дані по перекладачу, type – означає тип словника, lang_id – повертає мову із зв’язку з таблиці language, created_at – повертає дату створення запису, update_at – повертає дату оновлення запису.

Сутність DictionariesFiles (проміжна таблиця між сутностями Dictionaries та Files) містить змінні: id – номер таблиці, dictionary_id – має зв’язок з Dictionaries, повертає дані по словнику, file_id зв’язок з сутністю Files, повертає дані по файлу

словника, created_at – повертає дату створення запису, update_at – повертає дату оновлення запису.

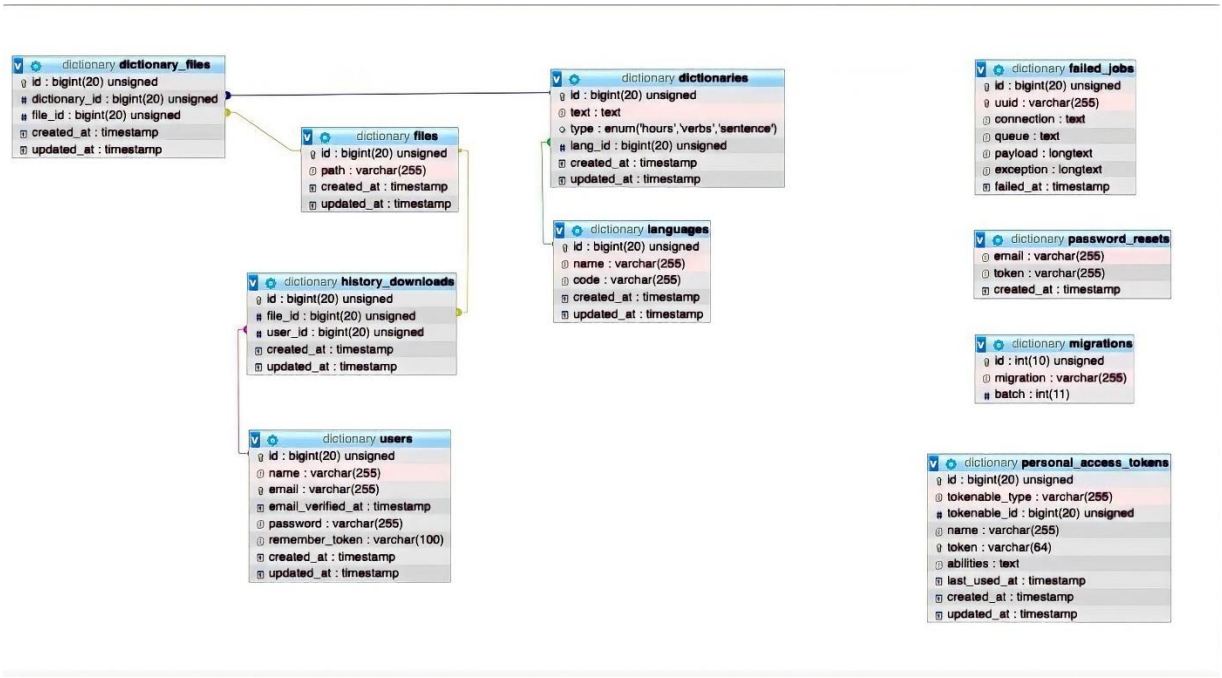

Рисунок 3.9 – ERD

Сутність Files являється сховищем файлів для словників, містить у собі змінні: id – номер таблиці, path – повертає шлях до файлу, name – повертає назву файла, created_at – повертає дату створення запису, update_at – повертає дату оновлення запису.

Сутність HistoryDownload зберігає у собі історію скачувань файлів користувача, містить у собі такі змінні: id – номер таблиці, file_id – зв’язок з сутністю Files, повертає дані по файлу словника, user_id – має зв'язок з сутністю Users, повертає дані користувача, created_at – повертає дату створення запису, update_at – повертає дату оновлення запису.

Сутність Languages зберігає у собі усі доступні мови для словників, має змінні: id – номер таблиці, name – повертає явну назву мови, code – повертає адміністративний код мови, він необхідний для форми перекладу, created_at – повертає дату створення запису, update_at – повертає дату оновлення запису.

Сутність Users зберігає дані про користувача які є у нас в системі. Змінні: id – номер таблиці, name – повертає ім’я користувача, email – повертає ім’я

користувача, password – зберігає пароль користувача для входу у систему у шифрованому вигляді, created_at – повертає дату створення запису, update_at – повертає дату оновлення запису.

3.6 Діаграма станів

Діаграма станів – це одна видів діаграм UML, що визначає зміну станів об’єкту у часі. Така діаграма потрібна для подання тих станів, у яких програмний засіб може знаходитися у різні моменти часу.

На рисунку 3.10 представлена діаграма класів для перегляду та скачування інформації з вкладки dictionaries.

Користувач переходить на сторінку dictionaries. На сторінці міститься декілька кнопок, вони проіменовані відповідно до мови. Для прикладу ми взяли одну кнопку. Коли користувач натискає на кнопку “англійська”, на сторінці з’являється необхідна інформація. Після перегляду інформації, нижче знаходиться посилання на скачування файлу. Користувач натискає на нього і файл скачується. В результаті необхідна дія може вважатись виконаною.

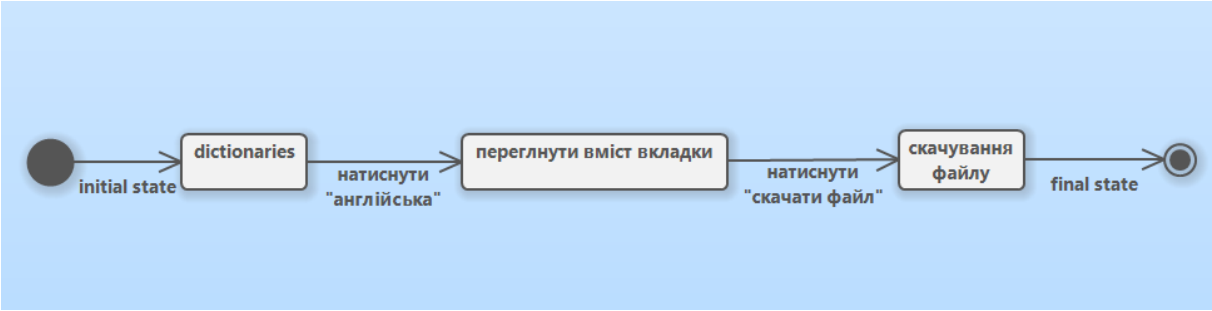

Рисунок 3.10 – Діаграма станів

3.6 Діаграма діяльностей

Діаграма діяльностей – це UMl діаграма яка представляє послідовність дій від одного до іншого[16]. Використовується для моделювання поведінкових об’єктів, систем та аспектів. Їх використовують для дослідження бізнес-процесів, щоб з’ясувати, як вони працюють і що їм потрібно. На рисунку 3.11 зображена діаграма діяльностей нашого програмного застосунку перекладу слів.

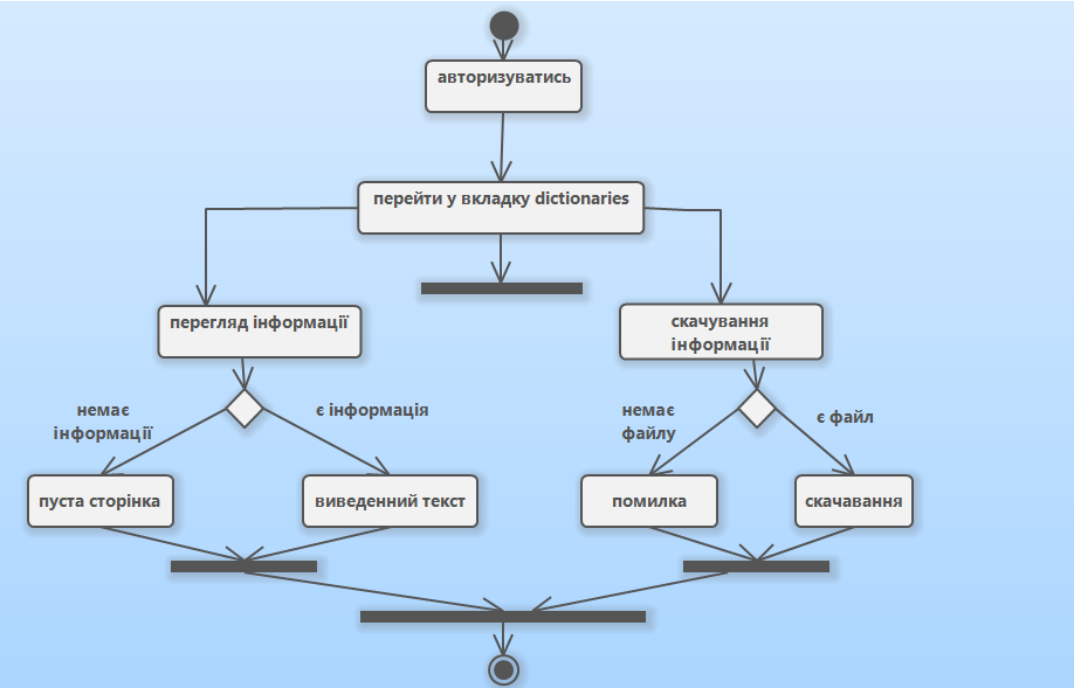

Рисунок 3.11 – Діаграма активностей

На рисунку 3.11 зображена діаграма активностей нашого програмного застосунку.

Є декілька можливостей використання вкладки dictionaries. Перегляд інформації та скачування. Коли користувач хоче переглянути вміст вкладки є два варіанти або на сторінці не має інформації і натискаючи на вкладку ми отримуємо пусту сторінку, або вона присутня і користувач може її переглянути.

Є можливість скачати інформацію. Не має файлу, виводиться помилка при спробі скачати, чи файл міститься і при спробі скачати його користувачу вдається це зробити.

Висновки до розділу 3

Отже третій розділ кваліфікаційної роботи був присвячений проєктуванню нашого програмного застосунку перекладу слів. Під час написання даного розділу були написані сценарії використання які необхідні для подальшої розробки застосунку. Створені діаграми класів, а також таблиця яка містить у собі назву класів, атрибути, методи та їх призначення. Діаграма бази даних. Також створені дві UML-діаграми: діагарама активностей та станів.

4 РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ

4.1 Створення бази даних

Невід'ємною частиною робочого застосунку є база даних. Розглянемо як була побудована та підключена база даних у нашому програмному застосунку.

Для початку, щоб не створювати кожне поле вручну, нам потрібно сформувати міграції. Міграції – простими словами, це код який вносить потрібні зміни у базу даних та дозволяє їх відмінити [12]. У результаті формування міграцій у нас вийшло 6 таблиць.

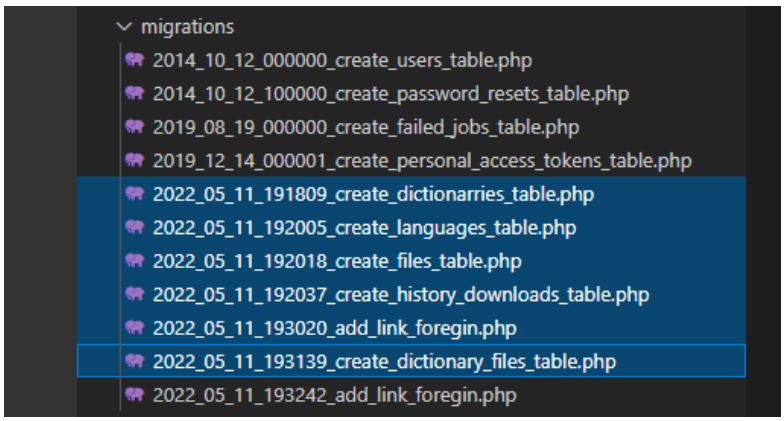

Рисунок 4.1 – Таблиці бази даних

Для створення міграції у консолі нам необхідно використовувати Artisan-команду make:migration. Artisan – це інтерфейс командного рядка, який входить у склад Laravel [13]. Також можна використовувати параметри –table та –create. Параметр –table відповідає за створення таблиці з відповідним ім'ям. А параметр –create,безпосередньо вказує на те що таблиця створюється. Для запуску усіх міграцій які містяться у нашому проєкті, у консолі, ми використовуємо команду php artisan migrate.

У класі міграцій є два методи:

1. up – використовується для додавання нових таблиць, стовбців чи індексів у базу даних;
2. down – відміняє операції виповнені попереднім методом.

На рисунку 4.2 представлений фрагмент коду, класу CreateLanguagesTable. У цьому класі містяться два методи up та down. У public function up() ми можемо побачити як створюється таблиця, за допомогою методу create().

```php
class CreateLanguagesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('languages', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('code');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('languages');
    }
}
```

Рисунок 4.2 – Міграційний клас CreateLanguagesTable

Як ми можемо побачити у цьому файлі створюється таблиця languages яка у собі містить два рядки: name та code. Створюється ця таблиця за допомогою фасаду Schema та, як вище було сказано, завдяки елементу create. Фасад Laravel - це так званий клас який забезпечує доступ до об’єкта з контейнера [14]. Інші 5 таблиць які зображені на рисунку 4.1, створені таким самим чином.

Далі ми створюємо окрему міграцію, для побудови зв’язку між попередніми шістьма таблицями. Вона знаходиться у файлі, який представлений на рисунку 4.3.

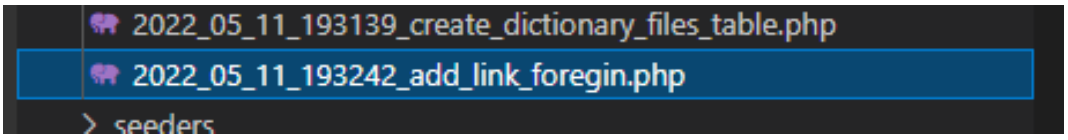

Рисунок 4.3 – Файл міграції

Наступним кроком є створення моделей даних та діаграми класів. На наступному рисунку 4.4 зображені моделі нашого програмного застосунку перекладу слів.

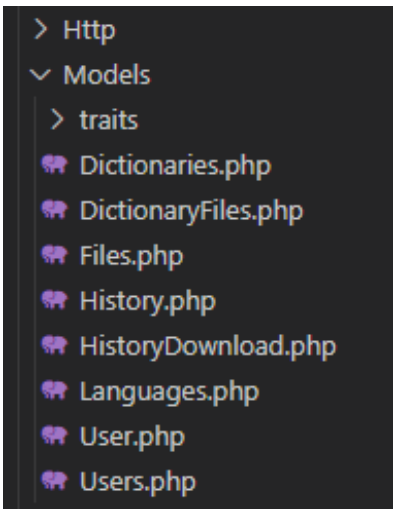

Рисунок 4.4 – Моделі застосунку

Ці дії більш детально вже описані у третьому розділі.

Фінальним кроком у роботі з базами даних, є їх підключення до проєкту. Підключення бази даних відбувається у файлі .env. Ця дія необхідна нам для коректної роботи застосунку. На рисунку 4.5 зображена частина коду що відповідає за підключення бази даних до нашого програмного застосунку перекладу слів.

```
10
11    DB_CONNECTION=mysql
12    DB_HOST=127.0.0.1
13    DB_PORT=3306
14    DB_DATABASE=dictionary
15    DB_USERNAME=root
16    DB_PASSWORD=root
17
```

Рисунок 4.5 – Підключення проєкту до бази даних

Розглянемо більш детально за що відповідає кожен елемент:

- DB_CONNECTION, драйвер для підключення до бази даних;
- DB_HOST, адреса (хост) для підключення;
- DB_PORT, порт бази даних (у нашому випадку стандартний);
- DB_DATBASE, ім’я бази даних;
- DB_USERNAME, DB_PASSWORD, ім’я та пароль.

Після підключення бази даних до проекту, наш програмний застосунок перекладу слів почне коректно працювати та виконувати свої безпосередні функції.

Також для зручності роботи з базою даних ми використовуємо застосунок phpMyAdmin. Даний застосунок нам необхідний для того щоб візуально переглянути структуру таблиць та зв'язки між цими таблицями, тобто він надає нам графічне представлення бази даних. Перевірити результати виповнених міграцій і у разі необхідності відмінити виповнені операції та виконати все знову. Також є можливість додати поле до бази даних одразу у цьому застосунку.

Завдяки графічному представленню бази даних ми маємо вже готову таблицю баз даних, що значно полегшує роботу розробнику. Таблиці бази даних можна передивитись у третьому розділі.

4.2 Створення сторінок застосунку.

Як було вказано раніше фреймворк Laravel має архітектуру MVC (Model View Controller) тобто модель, представлення і котролер.

Притримуючись архітектури MVC, для правильної роботи застосунку нам потрібні контролери. Саме через котролери відбувається взаємодія з представленням та базою даних. На рисунку 4.6 зображені контролери які задіяні у нашому програмному застосунку.

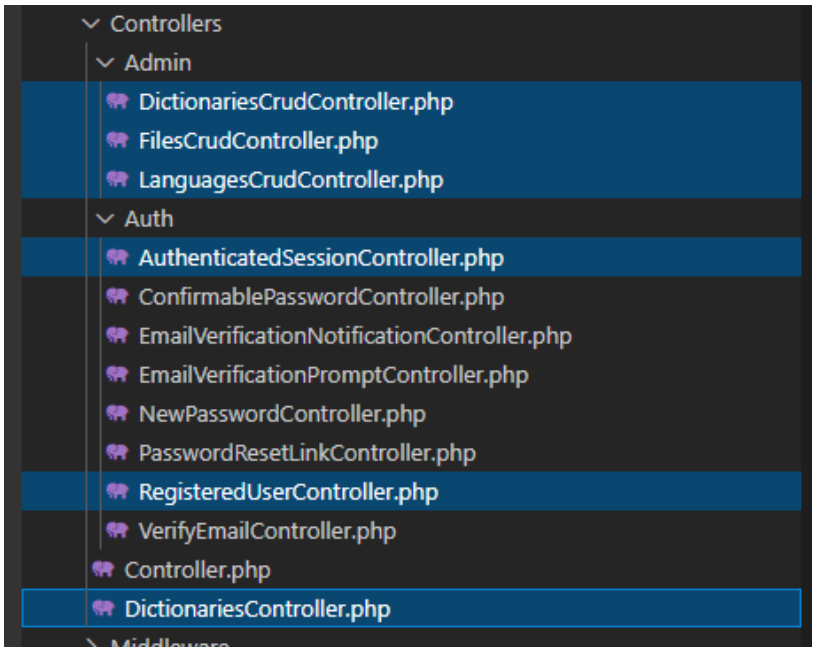

Рисунок 4.6 – Контролери застосунку

Розберемо деякі котролери більш детально.

DictionariesController – це контролер який відповідає за вимальовування облікового запису користувача після авторизації.

Методи конролеру:

1. create() – відкриває сторінку з словниками. З цього місця передаються у представлення мови з моделі Languages, які далі використовуються для вимальовування усіх доступних мов та словників по мовам. Представлення знаходиться у файлі Dictionaries.vue.

2. index() – відкриває форму для перекладів текстів. З цього місця передаються у представлення мови з моделі Languages. В цьому випадку мови використовуються в формі для перекладу текстів. Представлення знаходиться у файлі Dashboard.vue.

3. history() – відкриває сторінку з історією файлів які скачав користувач. З цього місця передаються дані з моделі History. Дані з цієї моделі беруться з деякими умовами. Береться id користувача який авторизований та вивантажуються усі файли та дані які відносяться до цього користувача. Представлення знаходиться у файлі History.vue.

4. saveVisit() – метод використовується для запису історії про те коли користувач скачав файл. Повертає void. Приймає post запити. Представлення не має.

5. store() – метод використовується для форми перекладу та приймає post запити. З цього місця передаються у представлення мови з моделі Languages. В цьому випадку мови використовуються в формі для перекладу текстів. Повертає json об’єкт – результат перекладу.

RegisteredUserController – відповідає за реєстрацію користувача у системі. Під час реєстрації ми маємо дві змінні: пошта та пароль.

При заповнені форми йде звертання на бекенд сторінки, там стоїть перевірка чи є у нас змінні, якщо вони не пусті то програма піде працювати далі за алгоритмом, а якщо пусті то буде помилка у вигляді червоної рамки.

Далі після пройденої перевірки, відбувається перевірка даних. Після обробки даних відбувається запис до бази даних. Якщо запис пройшов успішно то відбудеться вхід до системи, якщо станеться помилка, робота застосунку призупиниться, дані не запишуться, вхід до системи не відбудеться.

Методи контролера:

1. create() – представляє форму реєстрації. Представлення знаходиться у файлі Register.vue.

2. store() – метод відповідає за обробку даних з форми реєстрації. Якщо усі дані підходять і без помилок, тобто оброблюються, реєструє, авторизує, пропускає у систему з перенаправленням на головну сторінку.

AuthenticatedSessionController – виповнює вхід у систему.

Авторизація схожа за принципом до реєстрації. Користувач вводить свої дані, відбувається перевірка чи пуста форма для входу чи містить у собі змінні, також відбувається перевірка паролю, якщо ніде не має ніяких помилок то відбувається вхід у систему і можна починати працювати з інструментами у кабінеті.

Методи даного контролера:

1. create() – представляє форму для входу у систему. Передає у систему два параметри. Представлення знаходиться у файлі Login.vue.

2. store() – виконує вхід у систему, але на цей раз перевіряє користувача який був зареєстрований раніше.

3. destroy() – метод що виповнює роль виходу користувача із системи. Тобто відбувається розлогінування користувача з системи. Система знищує його сесію та відправляє на першу сторінку.

Також система має файл web.php. У цьому файлі формуються усі маршрути по яким користувач переміщується по сторінці.

Для розробки візуальної частини сайту, як сказано було у третьому розділі, ми використали Vue.js та InertiaJs.

InertiaJs – це аддон, або додатковий компонент для фреймворку Laravel. Він дозволяє замінити шаблонізатор Laravel blade та налаштувати передачу даних у Vue компонент без зайвих витрат ресурсів. Тобто це дозволяє створювати SPA(Single

Page Application) іншими словами односторінкові сайти.

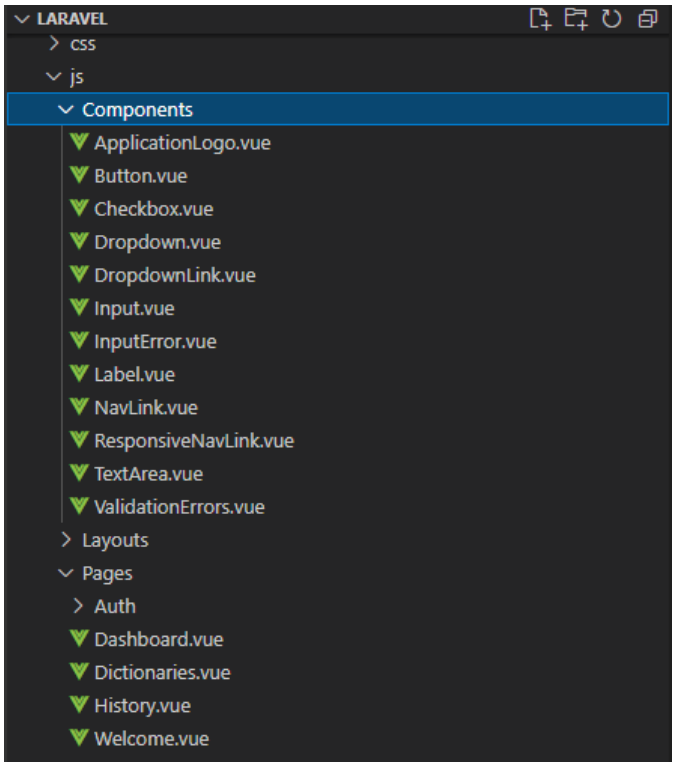

Рисунок 4.6 – Представлення та елементи для роботи з даними

На даному рисунку забражені файли що відповідають за представлення, тобто за візуальну частину застосунку та елементи роботи за даними.

4.3 Інтерфейс та функціонал сторінок

Розглянемо інтерфейс нашого програмного застосунку перекладу слів.

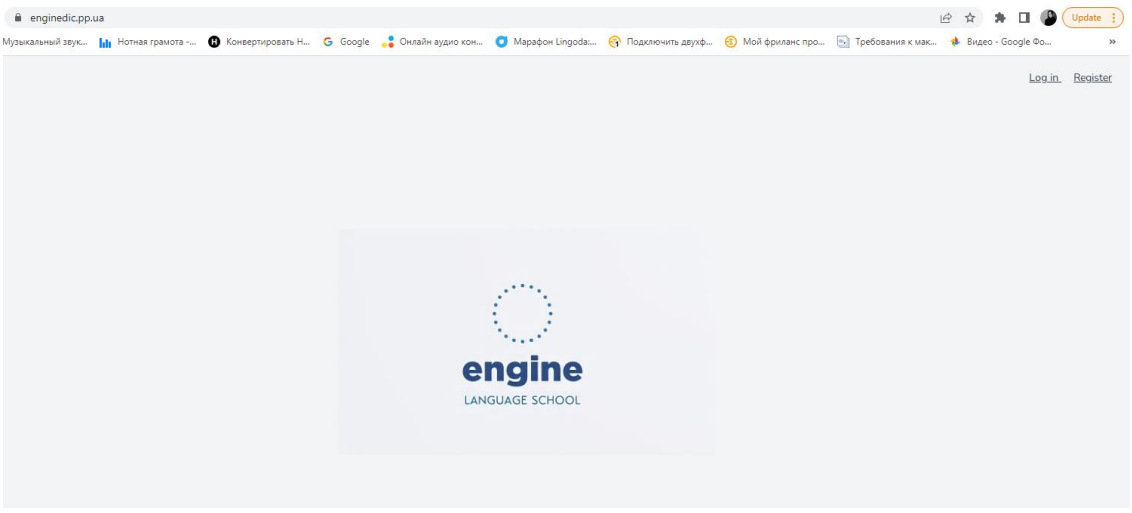

Рисунок 4.7 – Вигляд сторінки програмного застосунку

Коли користувач відкриває затосунок, перше що зустрічає його це логотип мовної школи для якої розроблявся даний застосунок. У лівому верхньому кутку ми можемо побачити дві так звані кнопки. Log in та Register. Вони відповідають за реєстрацію гостя та авторизацію користувача.

Натиснувши кнопку Register перед гостем з’являється вікно з формою для реєстрації.

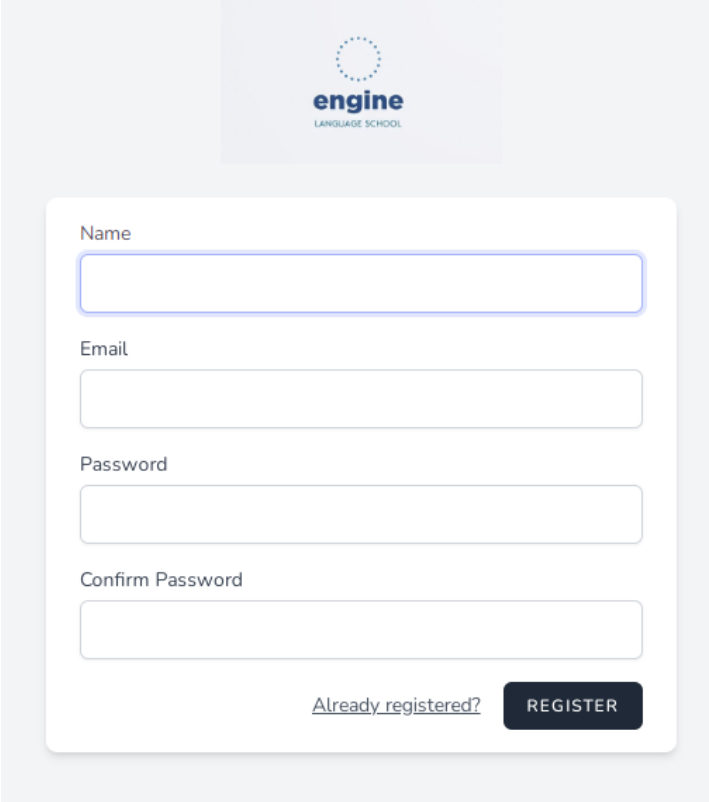

Рисунок 4.8 – Форма реєстрації

Гість заповню чотири поля: ім’я, власна електронна пошта, пароль та поле для підтвердження паролю, натискає кнопку Register. Як було вже сказано раніше після виконаних дій, система перекидує користувача на головну сторінку застосунку.

Також на рисунку 4.7 була представлена ще одна кнопка, Log in. Дана кнопка використовується для входу вже зареєстрованого користувача. Форма для входу зображена на рисунку 4.9. Користувач заповнює два поля: власна електронна пошта та пароль, натискає кнопку Log in. Також є кнопка remember me яка дозволяє

запам’ятати дані щоб у подальшому, для повторного входу у застсунок не потрібно було знову повторно вводити ці дані.

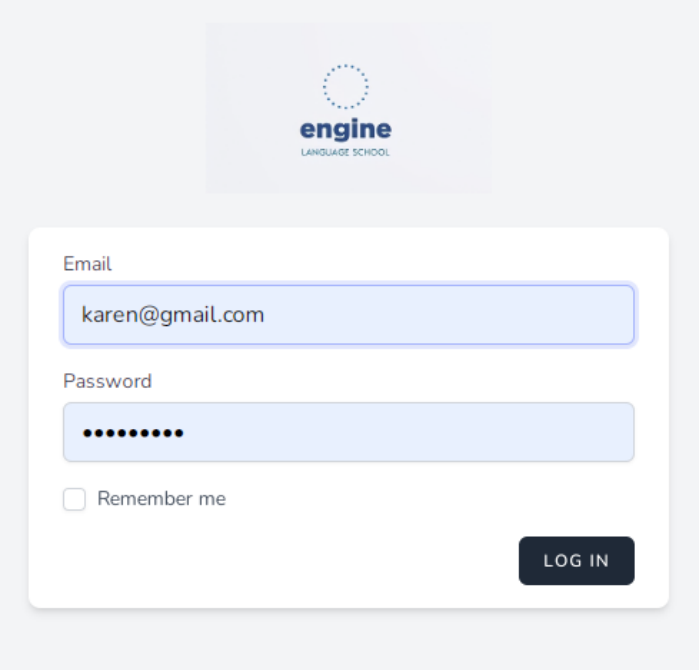

Рисунок 4.9 – Форма авторизації

Після успішної авторизації, користувач потрапляє на головну сторінку програмного застосунку перекладу слів.

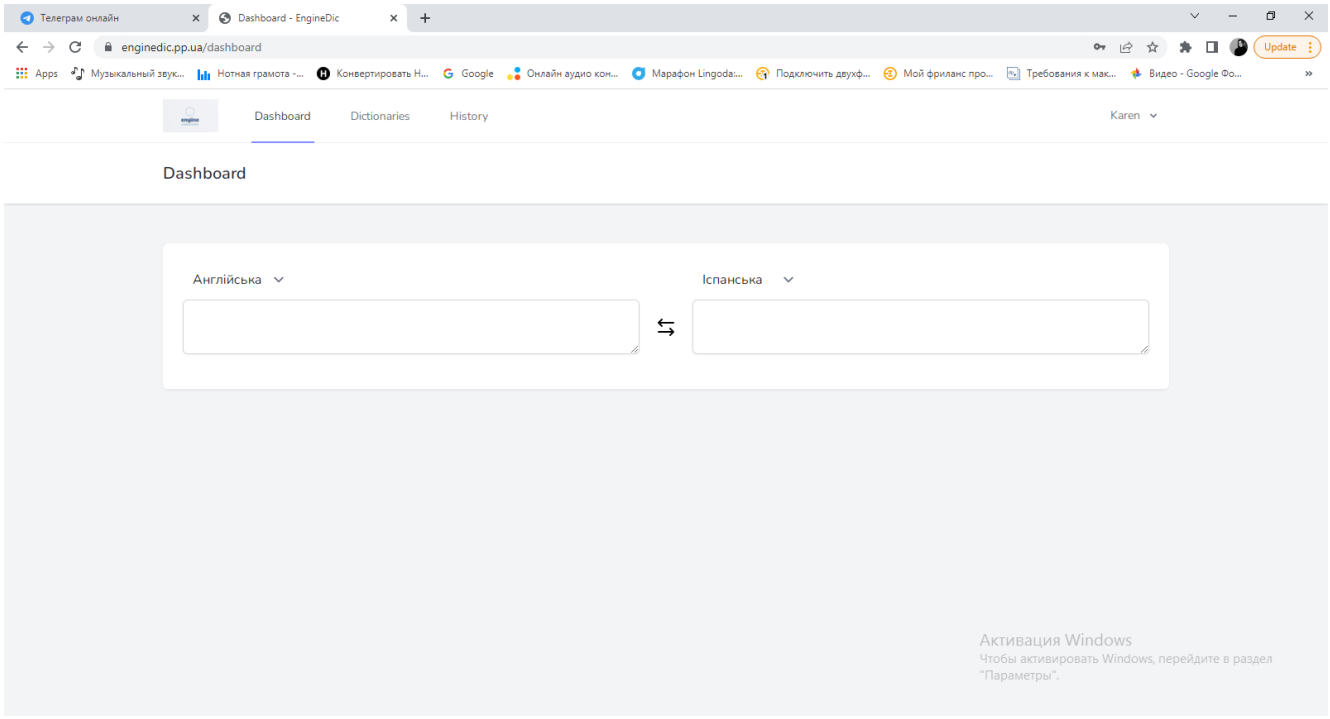

Рисунок 4.10 – Головна сторінка програмного застосунку

Головна сторінка програмного застосунку перекладу слів також являється і сторінкою для перекладу слів, на якій користувач безпосередньо вводить слово або текст для перекладу та отримує бажаний результат.

На даній сторінці ми маємо два вікна. Вікно для вводу слова або тексту які необхідно перекласти, та вікно для виводу результату перекладу. Над кожним вікном є випадаючий список з мовами з яких та на які потрібно перекласти. Також між вікнами є стрілки для зміни мови місцями.

У верхній частині сторінки, окрім Dashboard є ще дві вкладки, Dictionaries та History. Розглянемо детальніше кожну з них.

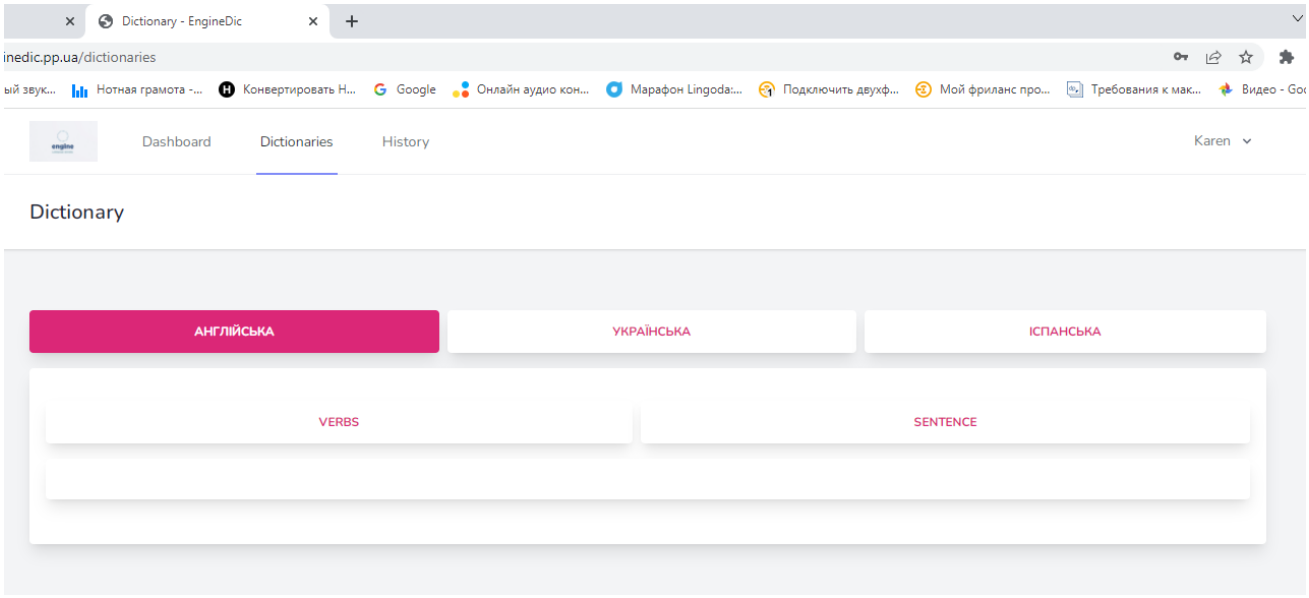

Рисунок 4.11 – Сторінка з додатковими матеріалами

Даний розділ потрібний нам для перегляду та скачування додаткових матеріалів з граматикою.

На сторінці є три розділи які розбиті по мовах. Обираючи необхідну нам мову ми бачимо нижче іще дві вкладки, це безпосередньо граматичні теми. Обираючи одну необхідну нам тему, ми можемо побачити матеріал представлений у вигляді тексту, а нижче посилання на скачування файлу. Користувач переглядає вміст розділу. Текст з граматичними темами відокремлений один від одного сірою лінією. У кінці тексту знаходиться посилання на файл для скачування, з цим самим текстом. Детально все зображено на рисунку 4.12.

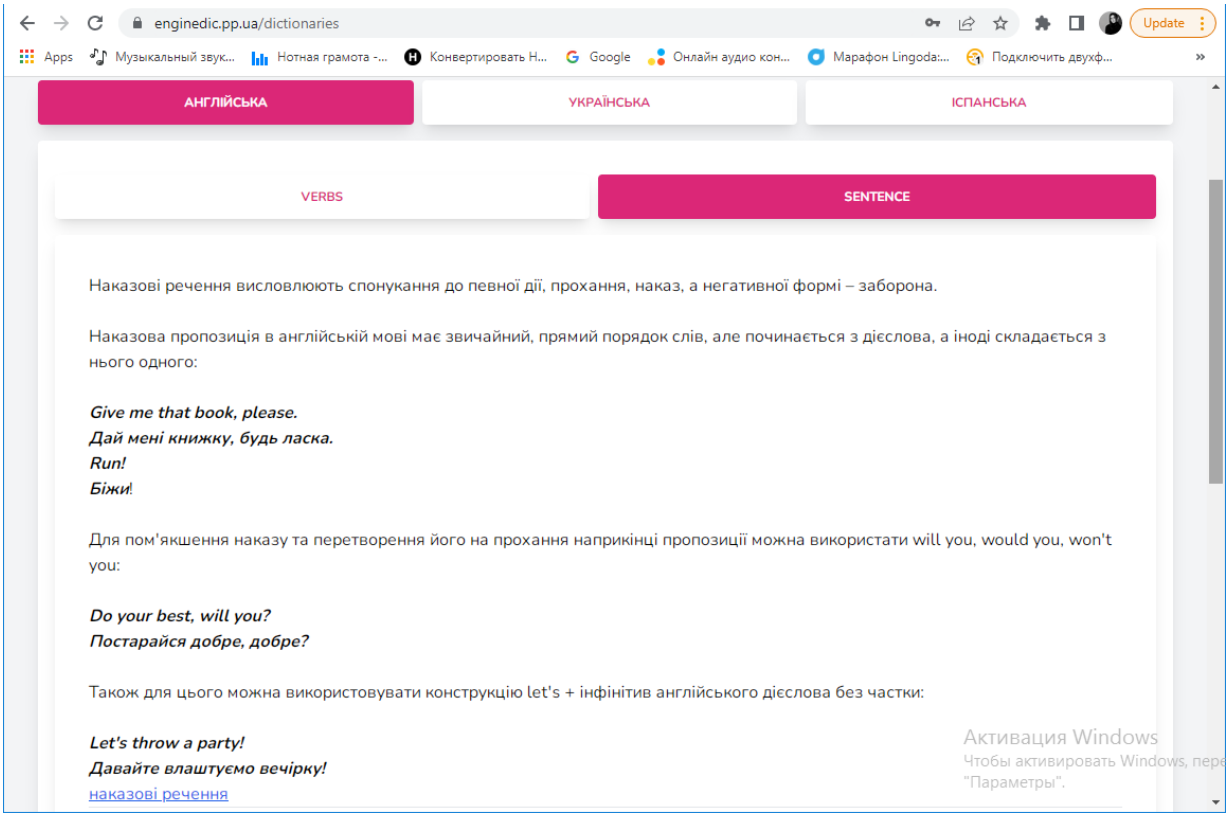

Рисунок 4.12 – Розділ з додатковими матеріалами

Третя вкладка у нашому програмному застосунку перекладу слів має назву History. Її основне призначення це перегляд історії скачаних файлів з розділу Dictionaries.

Користувач заходить у даний розділ. Перед ним з’являється умовна табличка яка складається з трьох стовпців:

– id, порядковий номер скачування;
– file, назва скачаного файлу;
– data, дата та час скачування.

Ця функція розроблена для зручного пошуку раніше скачаних файлів. І як бонус, якщо кількість даних дійсно велика, і користувач забув у якому розділі знаходиться той чи інший файл з необхідною інформацією, можна знайти його у списку вже раніше скачаних файлів, за датою наприклад, та скачати його знову саме у цій вкладці History.

На рисунку 4.13 можна переглянути як виглядає дана сторінка та усі елементи на ній.

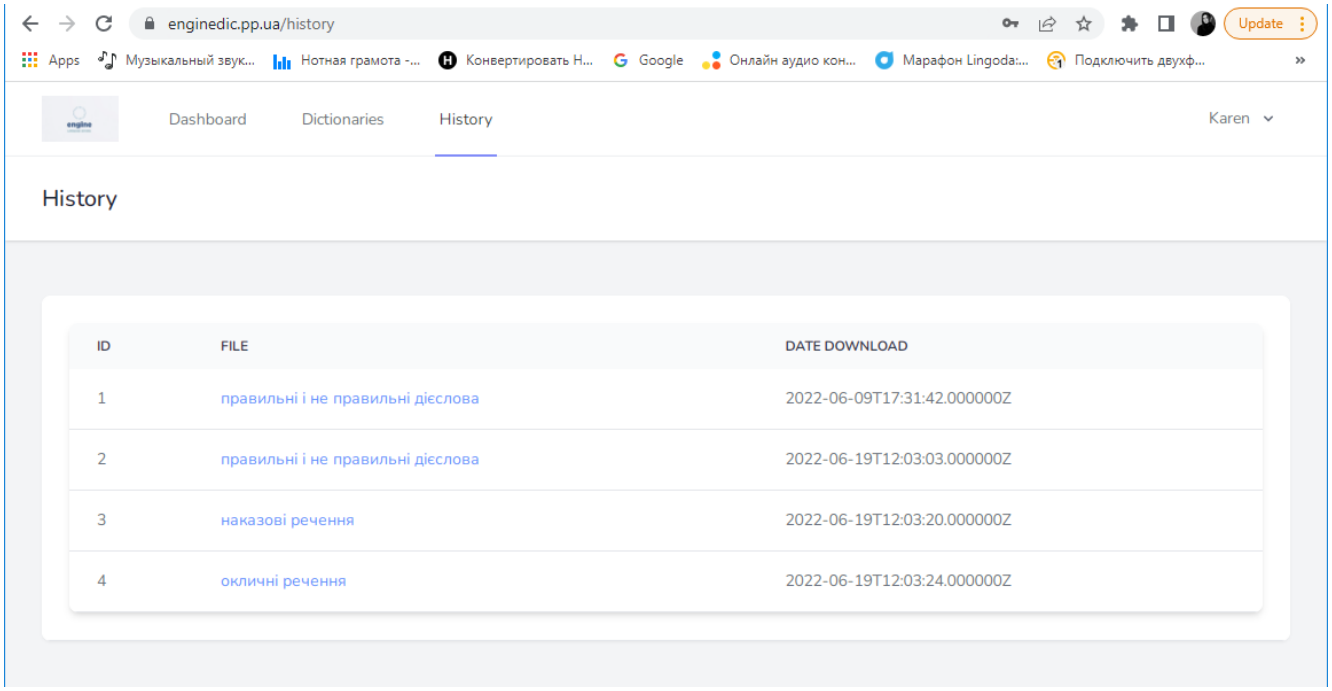

Рисунок 4.13 – Вигляд вкладки History

Також на сторінці ми маємо елемент який відповідає за вихід користувача із системи. Він знаходиться у правому верхньому кутку, та має назву ідентичну назві облікового запису користувача який зараз авторизований. Даний елемент зображений на рисунку 4.14.

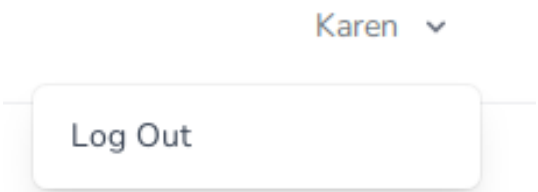

Рисунок 4.14 – Кнопка виходу із облікового запису

Натискаючи цю кнопку користувач виходить із системи і потрапляє на першу сторінку з якої ми почали. І потім для того щоб можна було користуватись застосунком необхідно знову авторизуватися.

4.4 Тестування застосунку

Для перевірки справної роботи застосунку, проводиться його тестування. Тестування – це процес дослідження та випробування готового програмного

застосунку. У наведених нижче таблицях представлені результати тестування нашого готового програмного застосунку перекладу слів.

Таблиця 4.1 – Реєстрація

Діючі особи	Гість, застосунок
Мета	Створення власного облікового запису
Передумова	Користувач не авторизований.
Успішний сценарій: 1. Гість переходить до форми реєстрації. 2. Заповнює необхідні рядки зі своїми даними. 3. Система здійснює валідацію даних. 4. Система зберігає обліковий запис. 5. Система переводить користувача на головну сторінку автоматично.	
Результат	Створено новий обліковий запис.
Розширення	
***а**	Не всі поля при реєстрації заповнені. Рамка вікна стає червоного кольору. **Результат:** користувач не може зареєструватись.
1а	Гість ввів вже існуючі дані. Система виводить повідомлення. **Результат:** Гість не може зареєструватись.
Усі сценарії розширення успішно виконані.	

Таблиця 4.2 – Авторизація

Діючі особи	Користувач , застосунок
Мета	Вхід до облікового запису
Передумова	Користувач не авторизований.
Успішний сценарій: 1. Користувач переходить до форми авторизації. 2. Користувач вводить необхідні дані для авторизації(пошта, пароль)	

Кінець таблиці 4.2

3. Система здійснює валідацію даних. 4. Користувач увійшов до сиситеми.	
Результат	Користувач авторизований.
Розширення	
***а**	Не знайдено користувача з введеною поштою. Вікно стає червоним. **Результат:** користувач не авторизований.
1а	Користувач ввів не правильний пароль. Система виводить відповідне повідомлення. **Результат:** користувач не авторизований.
2а	Користувач залишив поле для даних пустим. Система виведе відповідне повідомлення. **Результат:** Користувач не авторизований.
Усі сценарії розширення успішно виконані.	

Таблиця 4.3 – Перегляд та скачування додаткових файлів

Діючі особи	користувач, застосунок
Мета	Скачування файлу
Передумова	Користувач авторизований.
Успішний сценарій: 5. Користувач переходить у вкладку Dictionaries. 6. Обирає необхідний розділ. 7. Система виводить на екран текст. 8. Користувач натискає кнопку скачати.	
Результат	Користувач скачав файл
Розширення	
***а**	Не має доступу до бази даних. **Результат:** користувач не може скачати файл.

Кінець таблиці 4.3

1а	Не має підключення до інтернету. **Результат:** користувач не може скачати файл.
Усі сценарії розширення успішно виконані.	

Таблиця 4.4 – Переклад слова

Діючі особи	Гість, застосунок
Мета	Переклад слова чи тексту
Передумова	Користувач авторизований.
Успішний сценарій: 1. Користувач переходить до сторінки перекладача. 2. Вводить слово або текст. 3. Система відправляє інформацію на переклад, звертається на API Google. 4. API Google повертає перекладений текст.	
Результат	Створено новий обліковий запис.
Розширення	
***а**	Текст більше ніж 250 символів. Попередження, рамка вікна стає червоного кольору. **Результат:** не відбудеться переклад слова
1а	Не має з’єднання з інтернетом. **Результат:** перекладу слова або тексту не відбудеться.
Усі сценарії розширення успішно виконані.	

Результати проведених тестів показали що програмний застосунок перекладу слів та усі його функції працюють справно.

Висновки до розділу 4

Отже, у останньому, четвертому розділі кваліфікаційної роботи була описана програмна реалізація програмного застосунку перекладу слів. А саме створення та підключення бази даних. Для створення бази даних ми використовували міграції.

Такий метод створення баз даних є більш зручним та сучасним. У результаті було створено шість таблиць. На основі цих таблиць побудовані діаграми класів та діаграми баз даних. Для зручності роботи з базою даних використовували застосунок PhpMyAdmin. Також ми розглянули підключення бази даних до програмного застосунку, елементи які відповідають за це.

Було розглянуто створення сторінок представлення, візуальної частини застосунку яка полегшує роботу користувача з застосунком. Для розробки сторінок було використано InertiaJs додатковий інструмент для js та Vue.js. Розглянули архітектуру яка використовується у фреймворку Laravel. Відповідно до цієї архітектури були створені контролери. Також ми описали деякі з цих контролерів та їх методи.

Також був представлений інтерфейс застосунку та описані його елементи. Представлені скрини готового робочого програмного застосунку перекладу слів.

Проведене тестування програмного застосунку для того щоб перевірити чи коректно працюють уся функції та елементи застосунку.

У результаті проведених тестів було визначено що програмний застосунок перекладу слів, уся його існуючі функції та елементи працюють справно. Застосунок придатний для використання.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було орагнізовано та полегшено процес вивчення іноземної мови, завдяки швидкому перекладу слів та додатковим матеріалам.

Досягнення поставленої мети було реалізовано за рахунок виконання наступних завдань:

– аналіз предметної області;

– розробка вимог до системи перекладу слів на основі інформації про предметну область;

– налагодження роботи баз даних;

– проєктування системи перекладу;

– реалізація, тестування та відлагодження вебзастосунку для відстеження помилок.

При проведені аналізу предметної області було визначено актуальність та необхідність програмних застосунків перекладу слів, зумовлена великими ростом відстоку людей які вивчають іноземну мову самостійно.

Для формування вимог програмного застосунку перекладу слів були проаналізовані аналоги, які користуються найбільшою популярністю на даний момент. Визначено їх основні переваги та недоліки. У результаті було створено технічне завдання та специфікація вимог для програмного застосунку що розроблявся.

Далі були розглянуті етапи створення вебзастосунків та їх послідовність. Проаналізовані фреймворки на мові програмування php, визначені їх переваги та недоліки. У результаті проведеного аналізу був обраний фреймворк Laravel для розробки застосунку.

Також були проаналізовані декілька СКБД, визначені їх плюси і мінуси. У результаті була обрана СКБД MySQL.

Такі самі операції були проведені і з технологіями які використовуються у розробці інтерфейсу. У результаті були обрані InertiaJS та Vue.Js.

У третьому розділі були створені сценарії використання програмного

застосунку. Також були створені мокапи застосунку, так зване попереднє представлення інтерфейсу, які ми використовували при подальшій розробці візуальної частини програмного застосунку.

Була створена діаграма класів застосунку. Також ці класи були описані, визначені основні атрибути та методи кожного класу. Описано за що відповідає кожен метод. Створена ERD, діаграма діяльностей та станів.

У останньому розділі описувалась програмна реалізація нашого застосунку.А саме створення та підключення бази даних. Для створення бази даних ми використовували міграції. Такий метод створення баз даних є більш зручним та сучасним. У результаті було створено шість таблиць. На основі цих таблиць побудовані діаграми класів та діаграми баз даних. Також ми розглянули підключення бази даних до програмного застосунку, елементи які відповідають за це.

Було розглянуто створення сторінок представлення, візуальної частини застосунку яка полегшує роботу користувача з застосунком. Розглянули архітектуру яка використовується у фреймворку Laravel. Відповідно до цієї архітектури були створені контролери. Також ми описали деякі з цих контролерів та їх методи.

Також був представлений інтерфейс застосунку та описані його елементи. Представлені скрини готового робочого програмного застосунку перекладу слів.

Проведене тестування програмного застосунку для того щоб перевірити чи коректно працюють уся функції та елементи застосунку.

Результати тестування показали що застосунок працює коректно та готовий для використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Огляд Google Translate. URL: https://startpack.ru/application/google-translate. (дата звернення: 13.05.2022)

2. Огляд Deepl Translator. URL: https://startpack.ru/application/deepl. (дата звернення: 13.05.2022)

3. Огляд Reverso. URL: https://langformula.ru/reverso-context/. (дата звернення: 14.05.2022)

4. Коваленко О.В., Порівняльний аналіз найпоширеніших фреймворків для PHP у сфері веб-розробок. Наукові записки. 2020. №10. URL: https://core.ac.uk/download/pdf/84825979.pdf

5. Москаль В. Р. Метод проектування веб-застосунків. 2021. С. 14-15

6. Обоснование выбора программной среды разработки приложения. URL:https://studwood.net/1852079/informatika/obosnovanie_vybora_programmnoy_sredy_razrabotki_prilozheniya (дата звернення: 03.06.2022)

7. Обзор php фреймворков. URL: https://unetway.com/blog/php-framework-review(дата звернення: 03.06.2022)

8. Сравнение современных СУБД. URL: https://drach.pro/blog/hi-tech/item/145-db-comparison (дата звернення: 05.06.2022)

9. Десять лучших языков программирования для фронтенда. URL: https://blog.back4app.com/ru/ (дата звернення: 03.06.2022)

10. Верстка. URL: https://blog.skillfactory.ru/glossary/verstka/ (дата звернення: 07.06.2022)

11. Унифицированный язык моделирования UML. URL: https://samara.mgpu.ru/~dzhadzha/dis/15/200.html (дата звернення: 10.06.2022)

12. Для чего нужны миграции базы даных на проэктах. URL: https://maximaster.ru/blog/migrations/ (дата звернення: 13.06.2022)

13. Консоль Artisan. URL: https://laravel.su/docs/5.4/artisan (дата звернення: 17.06.2022)

14. Фасади. URL: https://laravel.su/docs/8.x/facades#:~:text=%D0%92%20%D0%BF%D1%80%D0%B8

%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B8%20Laravel%20%D1%84%D0%B0%D1%81%D0%B0%D0%B4%20%E2%80%93%20%D1%8D%D1%82%D0%BE,%5CSupport%5CFacades%5CFacade%20. (дата звернення: 17.06.2022)

15. What is the Vue? URL: https://vuejs.org/guide/introduction.html (дата звернення: 19.06.2022)

16. Буч Г., Якобсон А., Рамбо Д. UML. Керівництво користувача. 2006. 496с.

17. Angular. Преимущества Angular. URL: https://blog.skillfactory.ru/glossary/angular/(дата звернення: 10.06.2022)

18. Что такое фронтенд? URL: https://dan-it.com.ua/blog/razrabotka-so-storony-front-end-chto-jeto-takoe-i-chem-otlichaetsja-ot-back-end/ (дата звернення: 12.06.2022)

19. Larvel Owerview. URL: https://www.tutorialspoint.com/laravel/laravel_overview.htm#:~:text=Laravel%20is%20an%20open%2Dsource,is%20more%20structured%20and%20pragmatic. (дата звернення: 12.06.2022)

20. What is Entity Relationship Diagram (ERD). URL: https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/(дата звернення: 12.06.2022)

21. PhpMyAdmin introduction. URL: https://docs.phpmyadmin.net/en/latest/intro.html#:~:text=phpMyAdmin%20is%20a%20free%20software,queries%2C%20and%20adding%20user%20accounts. (дата звернення: 12.06.2022)

22. Система управления базами даных MySQL. URL: https://depix.ru/articles/sistema_upravleniya_bazami_dannyh_mysql (дата звернення: 12.06.2022)

23. Этапы разработки и создания сайта. URL: https://webtune.com.ua/ru/statti/web-razrabotka-ru/etapy-sozdaniya-sajta/(дата звернення: 12.06.2022)