

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ
АНГЛІЙСЬКОЇ МОВИ

DEVELOPMENT OF A WEB APPLICATION FOR ENGLISH
LANGUAGE LEARNING

Виконав(ла): студент(ка) 4 курсу, групи КНТ-111

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ГРЕБІНЕЦЬ О.В.

(ПРИЗВИЩЕ та ініціали)

Керівник ФЕДОРОНЧАК Т.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент ДЬЯЧУК Т.С.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Факультет КНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2025 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ГРЕБІНЦЯ Олександра Володимировича
 (ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка вебзастосунку для вивчення англійської мови
Development of a Web Application for English Language Learning

керівник проєкту (роботи) к.т.н., доцент, ФЕДОРОНЧАК Тетяна Василівна,
 (науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 02 червня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження.
2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ФЕДОРОНЧАК Т.В., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання «14» квітня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)


(підпис)

Олександр ГРЕБІНЕЦЬ

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис)

Тетяна ФЕДОРОНЧАК

(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 76 с., 4 табл., 59 рис., 3 дод., 21 джерел.

ВЕБЗАСТОСУНОК, REACT, DJANGO, ТЕСТУВАННЯ, АНГЛІЙСЬКА МОВА, КОРИСТУВАЧ, АДМІНІСТРАТОР.

Об'єкт дослідження – процес розробки програмних засобів для вивчення англійської мови.

Предмет дослідження – програмні засоби для вивчення англійської мови.

Мета роботи – розробка ПЗ у вигляді вебзастосунку, що допомагає користувачам самостійно опановувати навички англійської мови шляхом інтерактивного навчання.

Матеріали, методи та технічні засоби: бібліотека React, фреймворк Django, персональний комп'ютер з процесором AMD Ryzen 7 5800x під управлінням операційної системи Microsoft Windows 10.

Результати. Створено програмний вебзастосунок за допомогою React та Django, що допомагає користувачу вивчати англійську мову за допомогою тестувань, а також розділом з інформацією про граматичні часи.

Висновки. Розроблено програмний засіб у вигляді вебзастосунку для вивчення англійської мови, що забезпечує підхід до навчання шляхом складання тестів та розділом з вивченням граматичних часів .

Галузь використання – самостійне опанування англійської мови з будь-якої точки світу.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 76 pages, 4 tables, 59 figures, 3 appendixes, 21 sources.

WEB APPLICATION, REACT, DJANGO, TESTING, ENGLISH LANGUAGE, USER, ADMINISTRATOR.

Object of study is the process of developing software tools for English language learning.

Subject of study are software tools for English language learning.

The purpose of the work is to develop software in the form of a web application that helps users independently master English language skills through interactive learning.

Materials, methods, and tools: React library, Django framework, a personal computer with an AMD Ryzen 7 5800x processor running the Microsoft Windows 10 operating system.

Results. A web application was created using React and Django that helps learning English to user through tests, as well as a section with information about grammatical tenses.

Conclusions. A software tool in the form of a web application for learning English has been developed, which provides an approach to learning through taking tests and a section on studying grammatical tenses.

The field of use is a independent learning of English from anywhere in the world.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок.....	8
Вступ	9
1 Аналіз проблеми та постановка завдань дослідження.....	10
1.1 Актуальність онлайн навчання іноземних мов	10
1.2 Огляд існуючих аналогів.....	11
1.2.1 Learn English Online British Council	11
1.2.2 GrammarWay.....	13
1.2.3 Learning English Cambridge English.....	14
1.3 Постановка завдань дослідження.....	17
1.4 Висновки за розділом 1	18
2 Матеріали і методи	19
2.1 Вибір front-end засобу для реалізації клієнтської частини.....	19
2.2 Вибір back-end засобу для реалізації серверної частини.....	20
2.3 Вибір IDE	22
2.4 Висновки за розділом 2	24
3 Опис програми.....	25
3.1 Структура ПЗ	25
3.2 Розробка БД	25
3.2.1 Таблиця даних про тестування	26
3.2.2 Таблиці даних про питання тестувань.....	27
3.2.3 Таблиці даних про користувачів	27
3.3 Проектування інтерфейсу ПЗ.....	29
3.4 Прецеденти ПЗ.....	32
3.5 Характеристики процесу розробки ПЗ.....	33
3.6 Опис розроблених методів ПЗ	33
3.6.1 Функціонал тестування за навичками	33
3.6.2 Функціонал перегляду інформації про граматичні часи	34
3.7 Висновки за розділом 3	34

4	Експлуатація, тестування та експериментальне дослідження програми.....	35
4.1	Призначення й умови застосування програми.....	35
4.2	Характеристики програми.....	35
4.3	Інструкція по експлуатації програми.....	36
4.3.1	Звернення до програми.....	36
4.3.2	Вхідні та вихідні дані	36
4.4	Виконання та тестування програми.....	37
4.4.1	Виконання та тестування клієнтської частини.....	37
4.4.2	Виконання та тестування серверної частини.....	46
4.5	Висновки за розділом 4.....	50
	Висновки	52
	Перелік джерел посилання	53
	Додаток А Технічне завдання.....	55
	Додаток Б Фрагмент тексту програми.....	59
	Додаток В Слайди презентації	69

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– Application Programming Interface;
CSRF	– Cross-Site Request Forgery;
CSS	– Cascading Style Sheets;
DOM	– Document Object Model;
ER	– Entity-Relationship;
HTML	– HyperText Markup Language;
HTTP	– HyperText Transfer Protocol;
IDE	– Integrated Development Environment,
JS	– JavaScript;
JSON	– JavaScript Object Notation;
JSX	– JavaScript XML;
JWT	– JSON Web Token;
ORM	– Object-relational mapping;
SQL	– Structured Query Language;
UML	– Unified Modeling Language;
URL	– Uniform Resource Locator;
XML	– eXtensible Markup Language;
XSS	– Cross-site scripting;
БД	– база даних;
ПЗ	– програмне забезпечення.

ВСТУП

Англійська мова є універсальним способом спілкування, що об'єднує людей, культури та ідеї у сучасному глобалізованому світі. Вона охоплює різні сфери життя від освіти до дипломатії, є спільною основою для людей з різним мовним походженням для комунікації, сприяючи взаєморозумінню та ефективної співпраці. Значна поширеність у міжнародному дискурсі та академічних дослідженнях підкреслює центральну роль мови у сприянні зв'язкам та обміну знаннями [1].

Згідно з даними Ethnologue за 2023 рік, англійською мовою розмовляють приблизно 1,5 мільярда людей у всьому світі, що робить її другою за поширеністю мовою у світі. Відповідно до звіту від Британської Ради, приблизно 375 мільйонів людей розмовляють англійською як рідною мовою та 1 мільярд розмовляють нею як іноземною [1].

У 2023 році в Україні у межах національної програми Future Perfect було проведено дослідження для визначення мети вивчення громадянами англійської мови, ключовими висновками якого були:

- для більшості англійська мова є потребою сьогодення та інструмент для доступу до нових можливостей, а також підвищення рівня і якості життя;
- потреба у вивченні англійської мови серед громадян є однаковою і в побутовому спілкуванні, і в професійному контексті;
- за твердженнями представників бізнесу, вимоги щодо рівня володіння англійською кандидатами на посади постійно зростають, проте станом на сьогодні знайти співробітника з потрібним рівнем англійської не є важкою задачею, а оплата праці корелює зі знаннями та навичками [2].

Метою даної дипломної кваліфікаційної роботи є розробка вебзастосунку для вивчення англійської мови шляхом інтерактивного навчання, що передбачає систему авторизації, різні види тестування та ознайомлення з граматичними часами англійської мови.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Актуальність онлайн навчання іноземних мов

У наш час Інтернет відіграє важливу роль у створенні інформаційного простору глобального суспільства, слугує фізичною основою доступу до вебсайтів і багатьох систем (протоколів) передавання даних [3].

За останнє десятиліття технологічний та суспільний розвиток сприяв тому, що програми та курси вищої освіти пропонуються частіше у онлайн-форматі. З суспільної точки зору, потреба в розширених та якісніших можливостях дистанційної освіти була зумовлена нестачею кадрів з необхідними навичками у сільській місцевості та географічною нерівністю в доступі до вищої освіти. Це також було закликано через потребу в навчанні протягом усього життя та подальшій освіті, враховуючи зростання рівня безробіття та зміну умов на ринку праці [4].

Пандемія COVID-19 глибоко змінила життя та професійну практику. У той час, коли навчальні заклади по всьому світу були змушені закритися, викладачі мов та ті, хто вивчає мови, були змушені викладати та вивчати мови онлайн у глобальному масштабі. Хоча технології все більше інтегруються в мовну освіту останніми роками та знайшли широке застосування в добре забезпечених ресурсах контекстах, тисячі викладачів мов та тих, хто вивчає мови, раптово були змушені використовувати Інтернет як єдиний засіб для викладання та навчання – вперше та без належної підготовки. Багато хто не тільки був погано підготовлений до виклику викладання та вивчення мов онлайн, але й у багатьох випадках був погано забезпечений місцевою інфраструктурою та ресурсами. Крім того, існує занепокоєння, що це раптове, повсюдне та масштабне збільшення використання онлайн-навчання могло посилити вплив нерівного доступу до інфраструктури та ресурсів [5].

1.2 Огляд існуючих аналогів

1.2.1 Learn English Online | British Council

Learn English Online | British Council – онлайн-платформа для вивчення англійської мови, що призначена для людей які бажають покращити свій рівень знань за допомогою різних тестувань, довідників або записавшись на онлайн курси. Головна сторінка платформи зображена на рис. 1.1.

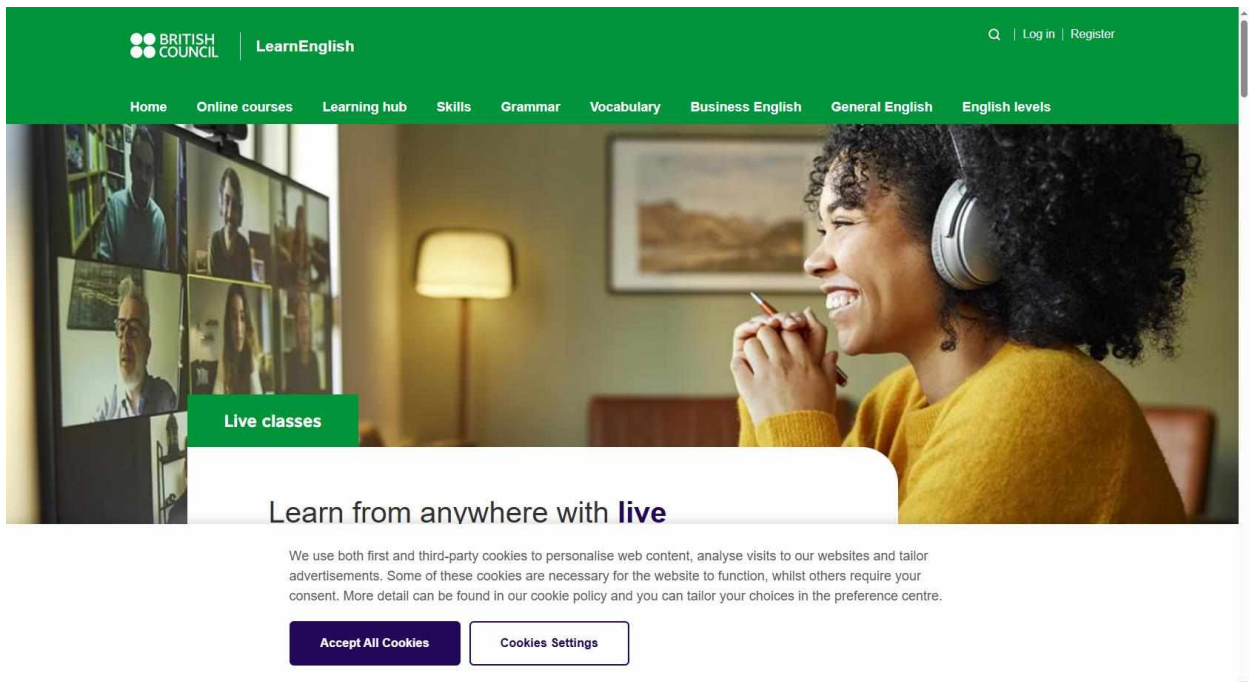


Рисунок 1.1 – Головна сторінка “Learn English Online | British Council” [6]

У рамках самостійного навчання користувач може обрати наступні розділи:

- аудіювання (listening): прослухати аудіозапис та виконати поставлені завдання;
- читання (reading): прочитати текст та виконати поставлені завдання;
- написання (writing): прочитати текст та підказки і виконати поставлені завдання;
- розмова (speaking): прослухати відео і виконати поставлені завдання.

- словник (vocabulary): співставити елемент на фотографії зі значенням слова або обрати одну з двох запронованих ігор.

Приклад тестування зображено на рис. 1.2.

A request from your boss



Listen to requests from a manager to practise and improve your listening skills.
Do the preparation task first. Then listen to the audio and do the exercises.

Preparation

0:00 / 1:39



Online English courses

Our websites

- ✓ LearnEnglish Kids for children aged 5 to 12
- ✓ LearnEnglish Teens for children aged 13 to 17
- ✓ TeachingEnglish for teachers and teacher educators

Help

- ✓ Free online English test
- ✓ Sign up for our newsletter
- ✓ Frequently asked questions
- ✓ Gettinn started

Рисунок 1.2 – Приклад тестування

Після вибору відповідей на питання, користувач бачить результат у кількості балів та правильні і неправильні відповіді. Приклад зображено на рис. 1.3.

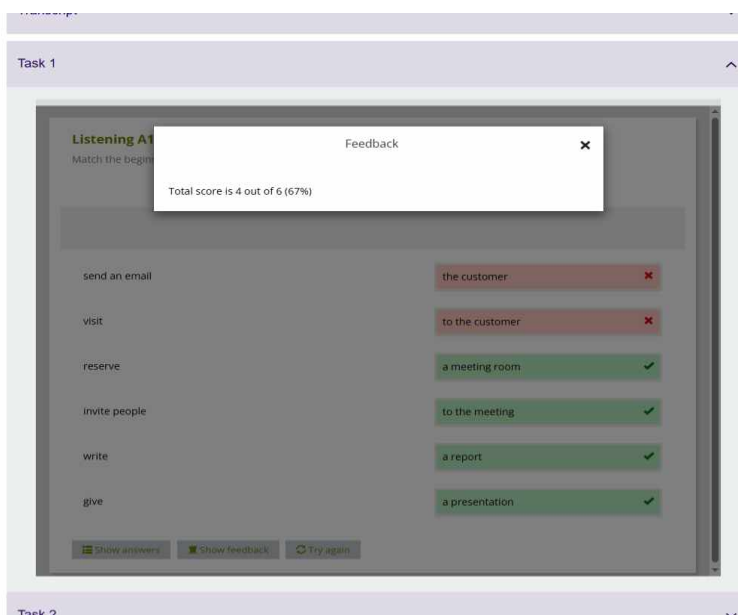


Рисунок 1.3 – Приклад результату тестування

У розділі “Grammar” можна покращити свою граматику, обравши бажаний рівень англійської мови і в ньому одну із запропонованих тем. Приклад зображено на рис. 1.4.

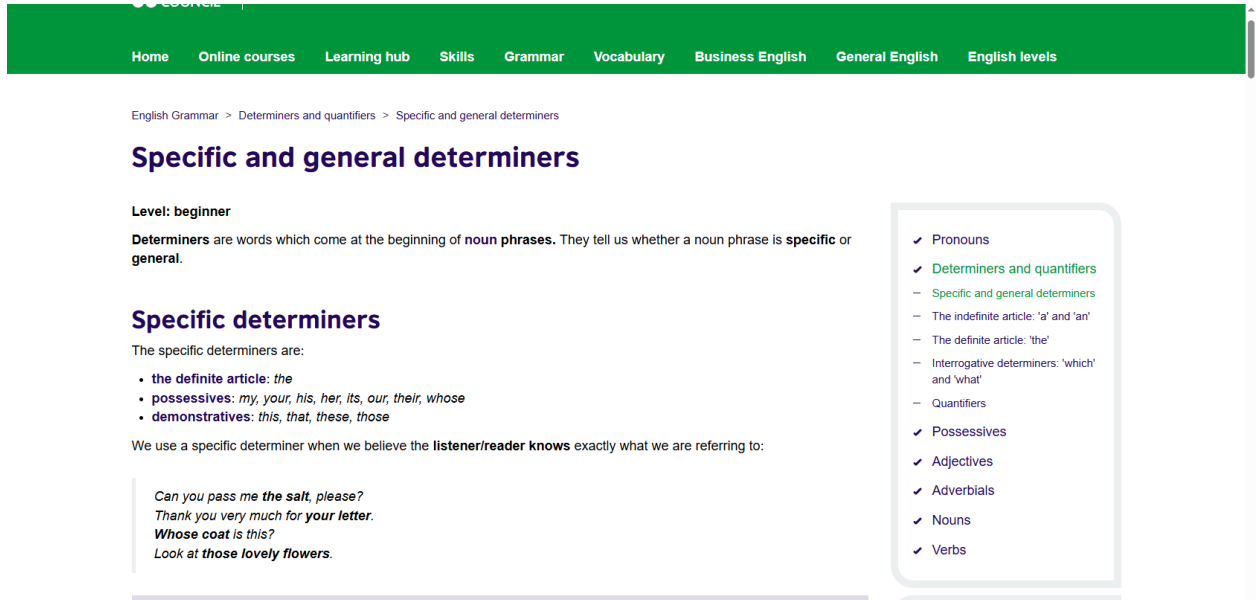


Рисунок 1.4 – Розділ “Specific and general determiners” розділу граматики

1.2.2 GrammarWay

GrammarWay – онлайн-платформа, що містить інформацію про граматику англійської мови. Головна сторінка зображена на рис. 1.5.

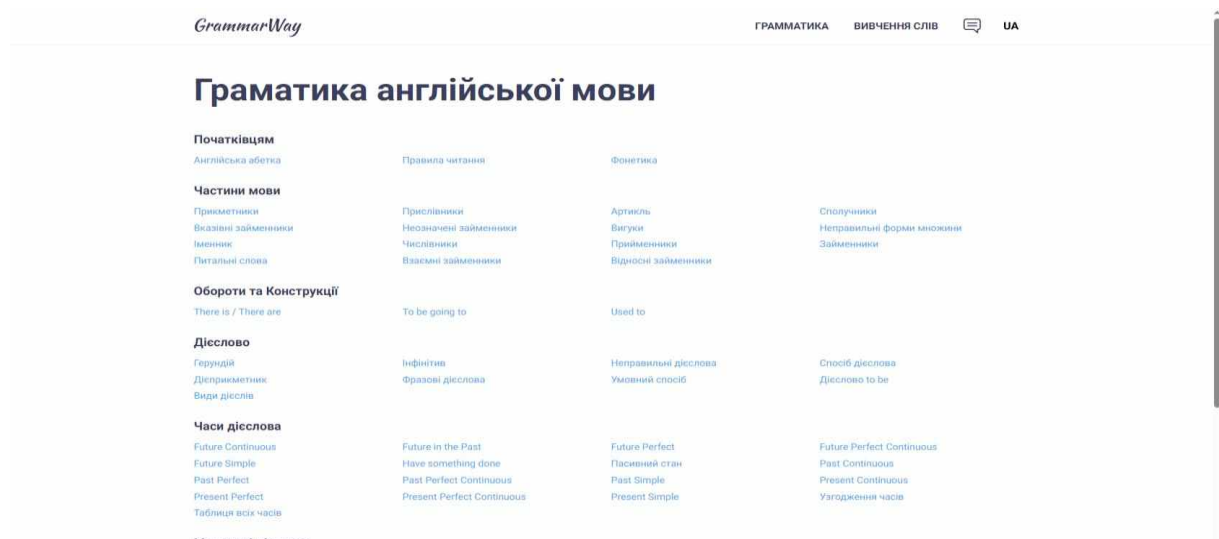


Рисунок 1.5 – Головна сторінка “GrammarWay” [7]

Користувач може обрати розділ, що його цікавить. Після цього буде виведено інформацію про розділ з прикладами, що зображено на рис. 1.6.

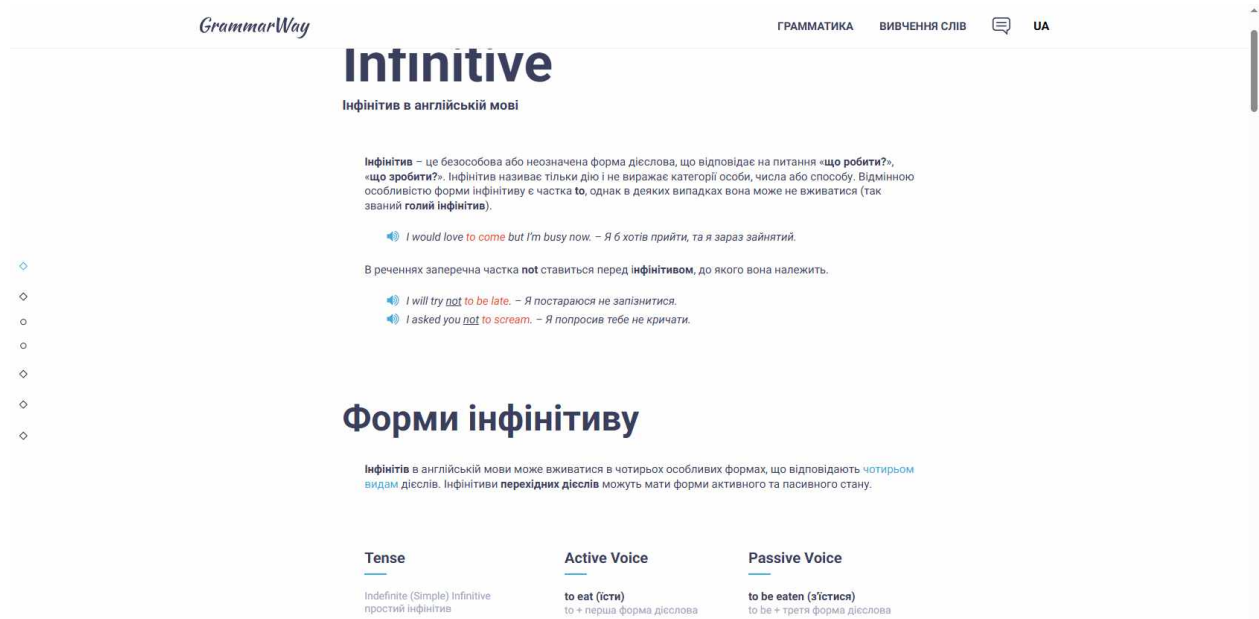


Рисунок 1.6 – Приклад сторінки розділу

1.2.3 Learning English | Cambridge English

Learning English | Cambridge English – онлайн-платформа для вивчення англійської мови. Головна сторінка зображена на рис. 1.7.

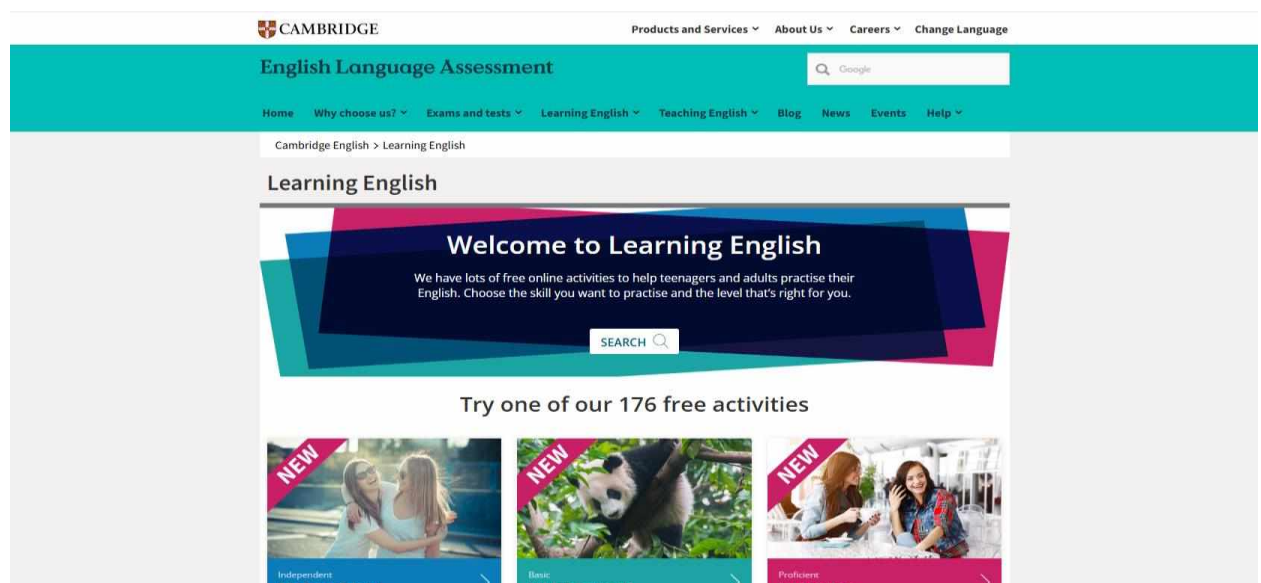


Рисунок 1.7 – Головна сторінка “Learning English | Cambridge English” [8]

Навчання передбачається у вигляді активностей, де необхідно відповісти на поставлені питання. Приклад правильної відповіді на питання зображено на рис. 1.8.

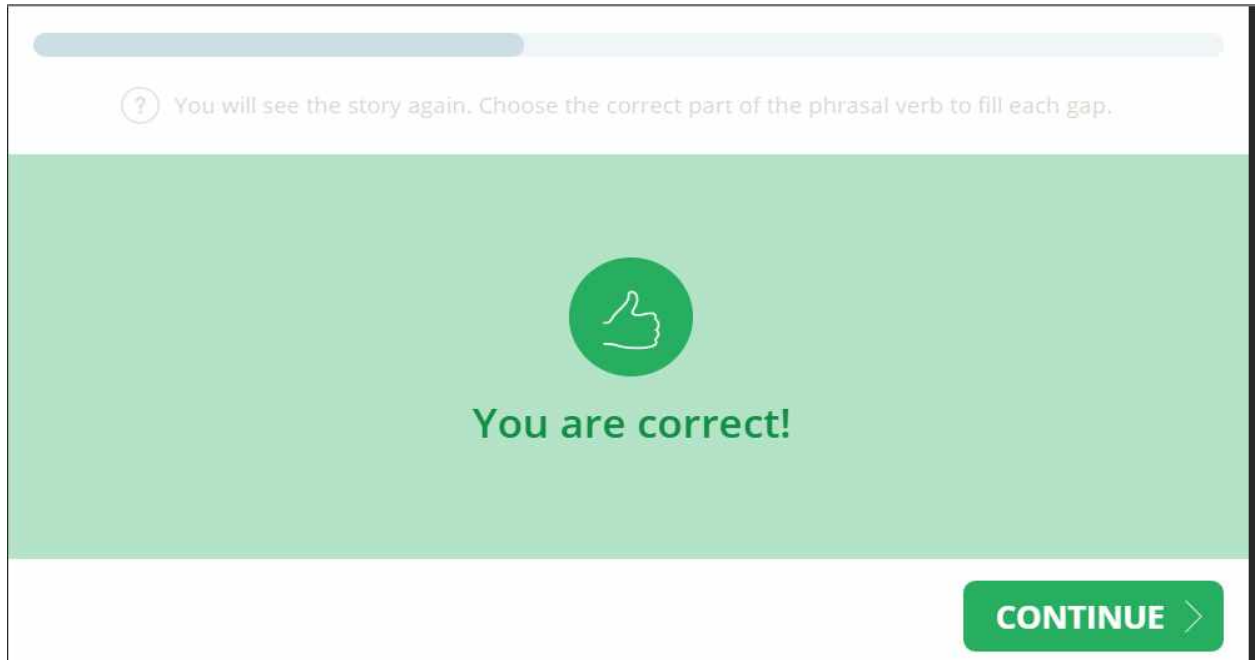


Рисунок 1.8 – Приклад правильної відповіді на питання

Приклад неправильної відповіді на питання зображено на рис. 1.9.

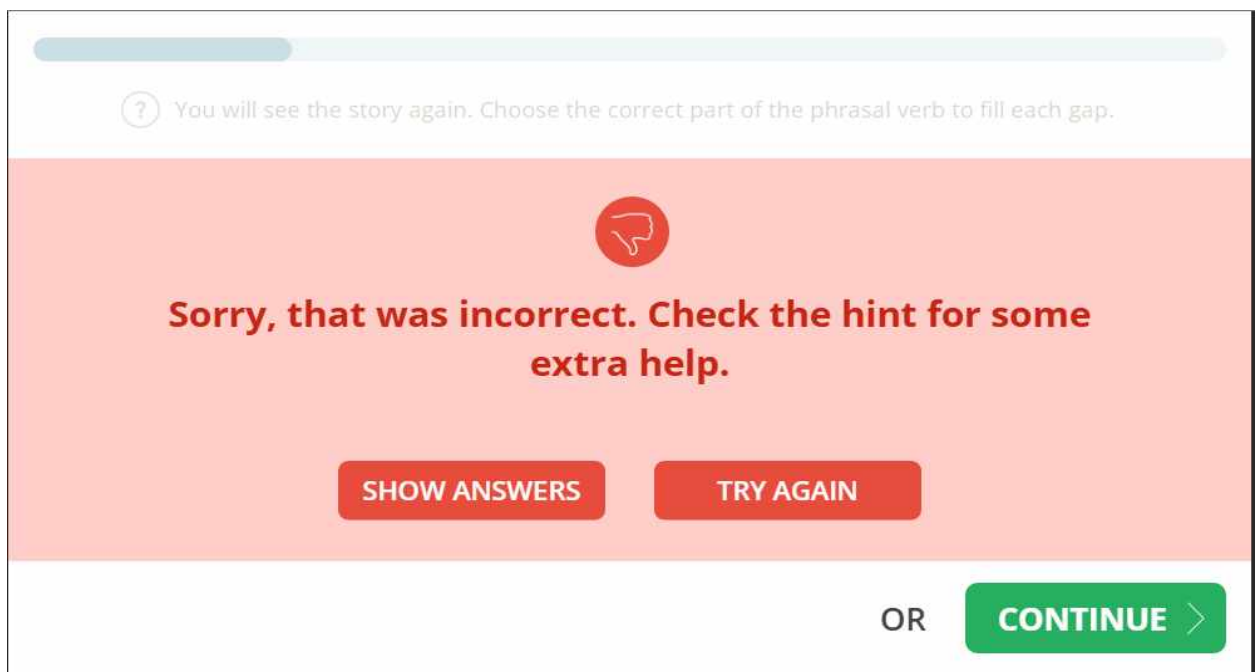


Рисунок 1.9 – Приклад неправильної відповіді на питання

Після відповідей користувач бачить результат складання тестування у відсотках. Приклад зображено на рис. 1.10.

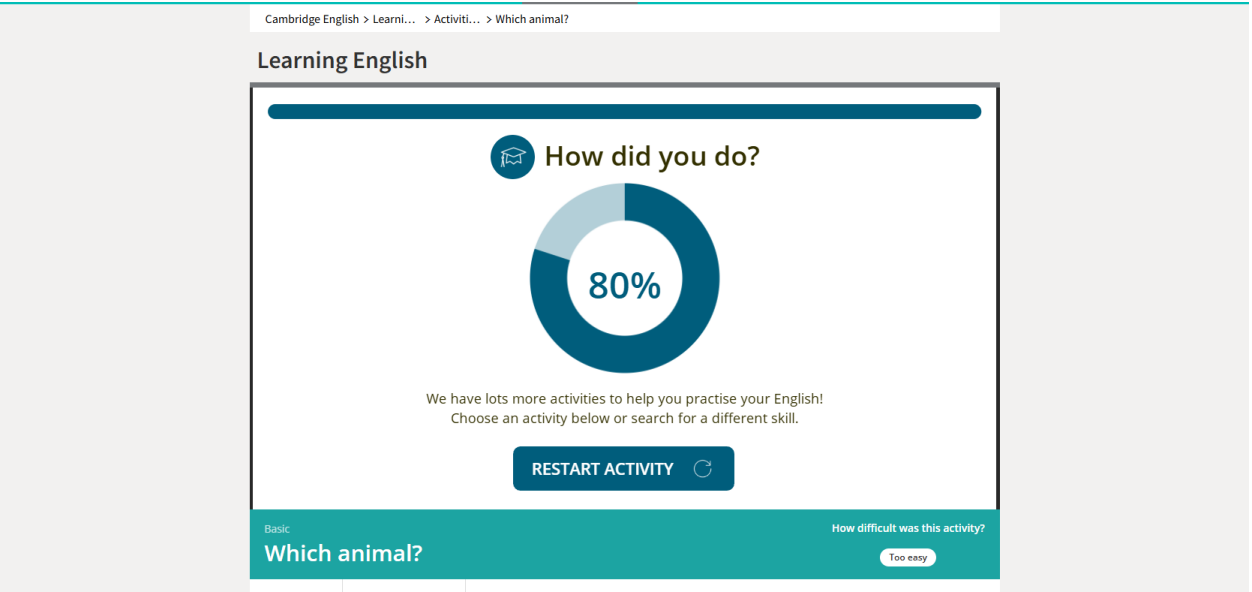


Рисунок 1.10 – Приклад виконання активності

Порівняння існуючих аналогів подано в табл. 1.1.

Таблица 1.1 – Порівняння існуючих аналогів

Критерій	Learn English Online British Council	GrammarWay	Learning English Cambridge English
1	2	3	4
Наявність активностей	+	-	+
Наявність інформації про граматичні часи	+	+	-
Наявність безкоштовного контенту	+	+	+

Продовження таблиці 1.1

1	2	3	4
Наявність української локалізації	-	+	-
Відсутність реклами	+	+	+

Порівнявши існуючі аналоги, можна зробити висновок про те, що платформи в основному орієнтовані або на активність користувача, або на граматику. Під час виконання активностей, не маючи інформації про граматику важко відповідати на запитання і бажано її знаходити саме на даній платформі, а не шукати в іншій, що зекономить час користувачу. У той же час, платформи, орієнтовані на граматику, мають відносно невелику можливість закріпити знання лише у рамках заданих прикладів використання. Також контент даних аналогів в основному безкоштовний, окрім додаткових курсів з викладачами. В аналогах, що не мають української локалізації, користувач може не розібратися в тому що йому необхідно робити, не використовуючи перекладачі. Зважаючи на це, у рамках розробки варто поєднати і активності, і граматику.

1.3 Постановка завдань дослідження

Метою даної роботи є розробка вебзастосунку для вивчення англійської мови, який включає в себе як клієнтську частину, за допомогою якої користувач взаємодіє із системою та покращує свої знання англійської так і серверну частину, яка дозволяє адміністраторам керувати вмістом застосунку.

Вебзастосунок має відповідати наступним функціональним вимогам:

- можливість реєстрація та авторизація користувачів;

- можливість редагування даних про користувачів та тестування;
- адміністративна панель для внесення даних;
- тестування для перевірки знань користувача (читання, слухання, граматика, словник);
- розділ з граматичними часами англійської мови.

1.4 Висновки за розділом 1

У першому розділі було описано актуальність проблеми онлайн навчання іноземних мов, розглянути існуючі аналоги, що мають на меті вирішення даної проблеми. Було проведено аналіз переваг та недолік аналогів, на основі якого сформульовано вимоги до майбутнього вебзастосунку, враховуючи потреби аудиторії.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір front-end засобу для реалізації клієнтської частини

Для реалізації клієнтської частини застосунку було використано React.

React – бібліотека JS, розроблена компанією Meta Platforms, що дозволяє створювати інтерфейси користувача. Для написання коду використовується синтаксис розмітки JSX [9]. Він є більш суворішим за HTML та для повернення тегів JSX їх необхідно обгорнути у спільний батьківський об'єкт [10].

React-додатки складаються з компонентів. Компонент — це частина інтерфейсу користувача, яка має власну логіку і зовнішній вигляд та може бути розміром з кнопку або на цілу сторінку. Він може бути вкладений в інший компонент та обов'язково починається з великої літери. Головний компонент у файлі визначається ключовими словами “export default”. Для визначення стилів необхідно вказати CSS-клас за допомогою атрибута “className” та описати правила у окремому CSS-файлі. Для роботи зі змінними використовуються фігурні дужки. Для компактності коду можна використовувати наступний вираз: { умова ? (true) : (false) } або { умова && (true) } (якщо не потрібна else частина). Для відображення списку компонентів можна використовувати функцію map() [10].

Для зв'язку з серверною частиною використовується HTTP-клієнт axios що дозволяє обробляти JSON дані, підтримує запити на перехоплення та відповіді і багато іншого [11].

Серед основних особливостей React можна виділити наступні:

- придатність коду до повторного використання;
- можливість вставляти JS об'єкти всередині HTML завдяки JSX (для браузерів необхідна компіляція у код JS з використанням Babel);
- наявність багатьох розширень для створення повноцінних інтерфейсних програм;

- наявність віртуального DOM, що пришвидшує роботу вебзастосунку [12].

Порівняння front-end засобів для розробки наведено у табл. 2.1.

Таблиця 2.1 – Порівняння front-end засобів

Критерій	React	Vue.js	Angular
Документація	++	+	+
Продуктивність	++	++	+
Гнучкість	++	++	+
Спільнота	++	+	++
Масштабованість	++	+	++

На основі цього порівняння можна зробити висновок, що React є найбільш оптимальним варіантом для розробки front-end частини вебзастосунку. За даними критеріями взагалом він демонструє найкращі результати, що забезпечує високу ефективність розробки.

2.2 Вибір back-end засобу для реалізації серверної частини

Django — це високорівневий безкоштовний веб-фреймворк на Python з відкритим кодом, який сприяє швидкій розробці та чистому, прагматичному дизайну. Він позбавляє розробника значної частини клопоту веб-розробки щоб можна було зосередитися на написанні свого додатку [13].

Серед основних особливостей Django можна виділити наступні:

- вбудований захист Django від більшості XSS-атак, підробки міжсайтових (CSRF) запитів, SQL-ін'єкцій та багато іншого [14];
- автоматичний інтерфейс адміністрування для керування контентом [15];
- налаштування бази даних за допомогою моделей та міграцій [16];

- можливість створення веб-API за допомогою пакету Django Rest Framework [17];
- можливість створювати URL-адреси як завгодно, без обмежень фреймворку [18];
- підтримка аутентифікації на основі JWT з використанням пакету Simple JWT [19].

Порівняння back-end засобів для розробки наведено у табл. 2.2.

Таблиця 2.2 – Порівняння back-end засобів

Критерій	Django	Node.js	Laravel
Наявність влаштованої БД	+	-	-
Безпека	+	-	+
Наявність ORM	+ (вбудований)	+ (окрема бібліотека)	+ (вбудований)
Наявність вбудованої адмін-панелі	+	-	-
Масштабованість	+	+	+

На основі цього порівняння можна зробити висновок, що Django є найбільш оптимальним варіантом для розробки back-end частини вебзастосунку. Наявність вбудованої безпеки від різних найпоширеніших атак, ORM технології, бази даних та адмін-панелі одночасно дозволяє розробнику зосередитися на реалізації логіки програми і не витратити час на реалізацію базових важливих елементів застосунку.

2.3 Вибір IDE

Visual Studio Code – безкоштовна IDE від компанії Microsoft, що спрощує програмування для розробників ПЗ і, як наслідок, підвищує продуктивність роботи.

Серед основних особливостей Visual Studio Code можна виділити наступні:

- можливість використовувати для роботи з окремими файлами для швидкого редагування або відкрити папку як робочу область;
- вбудована підтримка багатьох мов програмування, яку можна розширити встановивши необхідне розширення мови в однойменному розділі (англ. “Extensions”);
- вбудований термінал, за допомогою якого можливо вводити консольні команди з середовища;
- технологія IntelliSense, що пропонує підказки під час написання коду;
- інтеграція з Git [20].

Інтерфейс Visual Studio Code зображено на рис. 2.1.

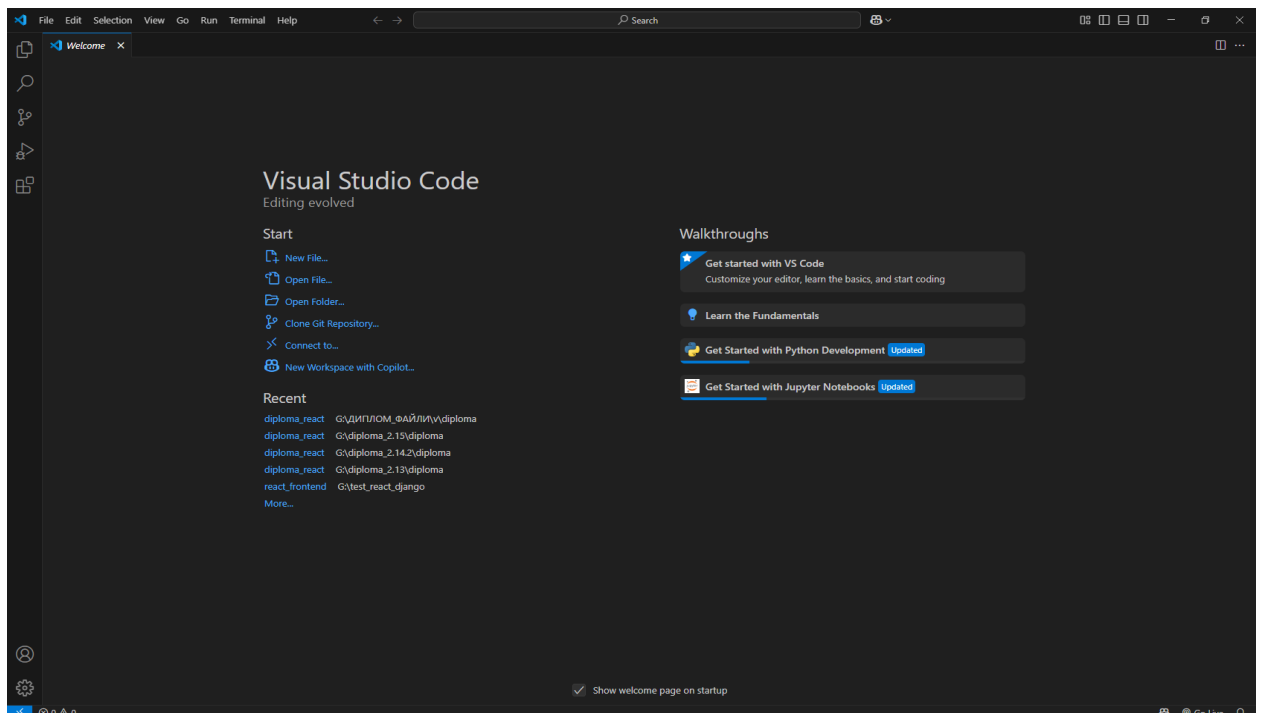


Рисунок 2.1 – Інтерфейс Visual Studio Code

PyCharm – спеціалізована IDE, що надає широкий спектр інструментів для розробників Python з метою продуктивної розробки. Серед основних особливостей PyCharm можна виділити наступні:

- кросплатформленість (Windows, macOS, Linux);
- інтеграція з системами контролю версій;
- допомога під час написання коду (автодоповнення, генерація тощо);
- навігація та пошук, що допомагає знайти необхідний текст у коді [21].

Порівняння IDE наведено у табл. 2.3.

Таблиця 2.3 – Порівняння IDE

Критерій	Visual Studio Code	PyCharm	Eclipse
Кросплатформленість	+	+	+
Мови програмування	C#, JavaScript, Python, C++ тощо	Python	Java, C/C++
Зручність використання	++	++	+-
Інтеграція з Git	+	+	+ (через плагін EGit)
Продуктивність	++	++	+-

На основі цього порівняння можна зробити висновок, що для розробки на Django краще використовувати PyCharm, який є спеціалізованим під Python. Для розробки на React краще використовувати Visual Studio Code, оскільки дана IDE підтримує JavaScript. Обидва середовища розробки демонструють високу продуктивність, зручність використання та можливість інтеграції з системою керування версіями Git.

2.4 Висновки за розділом 2

У даному розділі було розглянуто передові технології front-end та back-end розробки, сучасні IDE. Проаналізовано переваги та недоліки, враховуючи які, для розробки було обрано front-end бібліотеку React з використанням IDE Visual Studio Code та back-end фреймворк Django з PyCharm.

3 ОПИС ПРОГРАМИ

3.1 Структура ПЗ

Загальна структура ПЗ на основі обраної архітектури відображена на рис. 3.1.

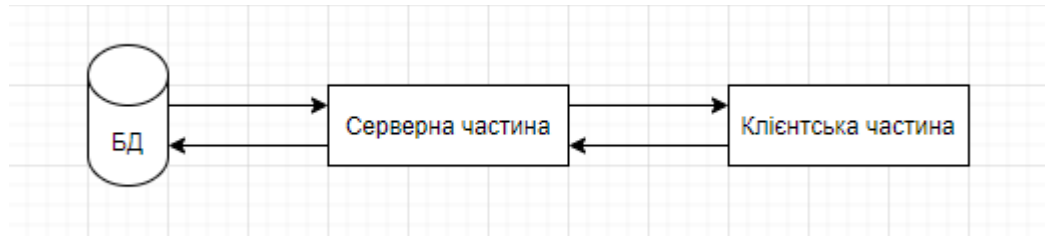


Рисунок 3.1 – Структура ПЗ на основі клієнт-серверної архітектури

Завдяки даній архітектурі між клієнтом та сервером є ефективна взаємодія, що дозволяє швидко оброблювати запити користувача.

3.2 Розробка БД

В даній роботі використовується БД SQLite, що вбудована за замовчуванням в Django. В контексті розробки це є доцільним вибором завдяки простоті у використанні, відсутності потреби в окремому сервері та ПЗ для його роботи.

Розроблена БД включає в себе як таблиці, згенеровані фреймворком за замовчуванням під час міграцій, так і таблиці, сформовано на основі користувацьких моделей, про які буде описано нижче.

ER-діаграма БД зображена на рис. 3.2.

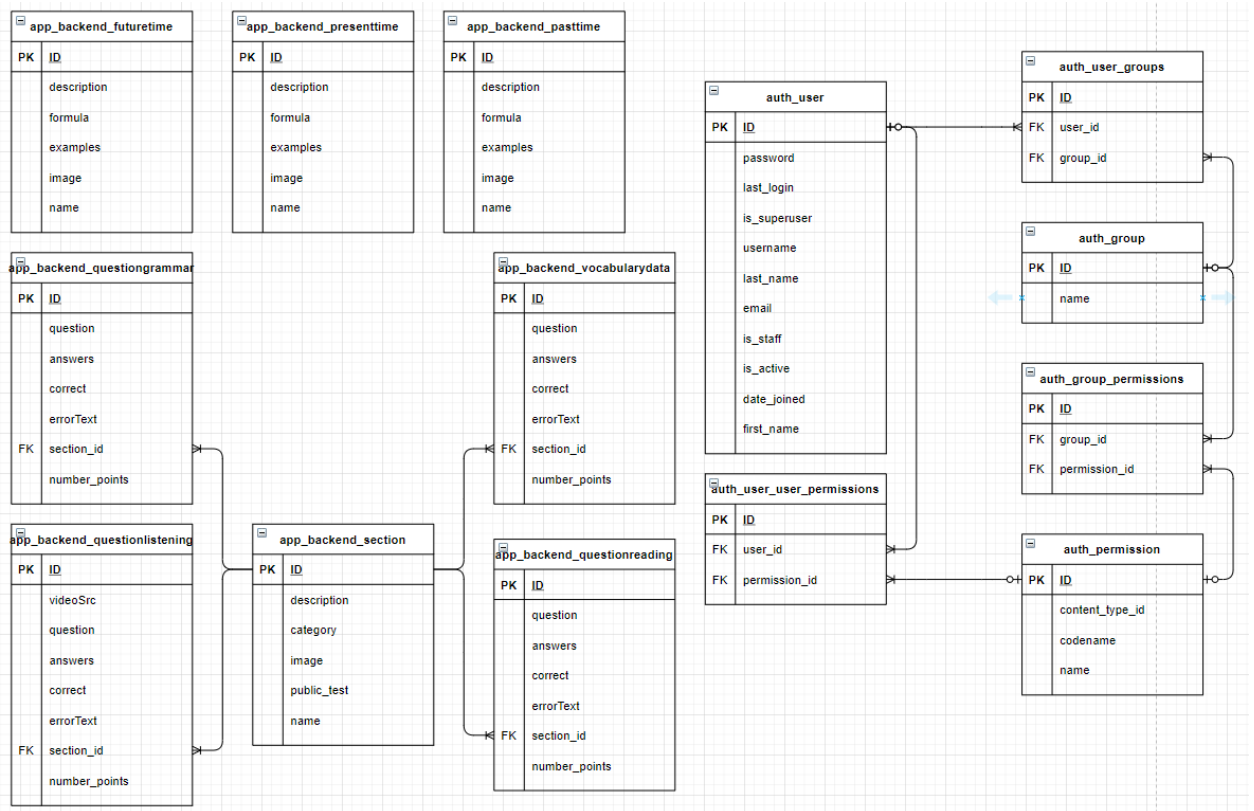


Рисунок 3.2 – ER-діаграма БД

3.2.1 Таблиця даних про тестування

Таблиця `app_backend_section` зберігає дані про тестування та містить наступні стовпці:

- ID – унікальний ідентифікатор тестування;
- `description` – опис завдання тестування;
- `category` – категорія тестування (читання, аудіювання, граматика, словник);
- `image` – картинка тестування, що відображається в клієнтській частині під час вибору;
- `public_test` – визначення доступності для неавторизованих користувачів;
- `name` – назва тестування.

3.2.2 Таблиці даних про питання тестувань

Таблиці `app_backend_questiongrammar`, `app_backend_questionreading`, `app_backend_vocabularydata`, `app_backend_questionlistening` зберігають дані про питання тестувань. Перші три таблиці мають ідентичну структуру, у той час як остання – `app_backendquestionlistening` – має додатковий стовпець для зберігання посилання на відео у YouTube, яке переглядає користувач.

Структура перших трьох таблиць виглядає наступним чином:

- ID – унікальний номер питання;
- `section_id` – тестування, до якого відноситься питання;
- `question` – питання, на яке треба відповісти користувачу;
- `answers` – відповіді до тестування у JSON форматі;
- `correct` – правильна відповідь на питання;
- `errorText` – текст, який виводиться користувачу у разі неправильної відповіді на питання;
- `number_points` – кількість балів за правильну відповідь на питання.

Структура таблиці `app_backend_questionlistening`, окрім вказаних вище стовпців, має додаткове поле:

- `videoSrc` – посилання на відео у YouTube.

3.2.3 Таблиці даних про користувачів

БД користує набір таблиць, що створюються автоматично фреймворком Django для зберігання інформації про користувача за замовчуванням. У вебзастосунку використовується вбудована у фреймворк модель користувача без змін, що містить основні дані про нього.

Таблиця `auth_user` містить дані про зареєстрованих користувачів та складається з наступних стовпців:

- ID: унікальний ідентифікатор користувача в БД;
- `password`: пароль користувача;

- last_login: дата останньої авторизації користувача;
- is_superuser: чи є користувач адміністратором;
- username: ім'я користувача;
- last_name: прізвище користувача;
- email: електронна пошта користувача;
- is_staff: доступ до адмін-панелі;
- is_active: чи активований обліковий запис користувача;
- is_superuser: чи має повний доступ;
- last_login: останній логін користувача;
- date_joined: дата створення користувача в БД.

Таблиця auth_group містить дані про групи користувачів та складається з наступних стовпців:

- ID: унікальний ідентифікатор групи в БД;
- name: назва групи.

Таблиця auth_user_groups пов'язує користувачів та групи і містить наступні стовпці:

- ID: унікальний ідентифікатор у таблиці;
- user_id: ідентифікатор користувача;
- group_id: ідентифікатор групи користувача.

Таблиця auth_group_permissions пов'язує групи та дозволи користувачів і містить наступні стовпці:

- ID: унікальний ідентифікатор у таблиці;
- group_id: ідентифікатор групи користувача;
- permission_id: ідентифікатор дозволу.

Таблиця auth_user_user_permissions пов'язує групи та дозволи користувачів і містить наступні стовпці:

- ID: унікальний ідентифікатор у таблиці;
- user_id: ідентифікатор користувача;
- permission_id: ідентифікатор дозволу.

Прототип сторінки вибору тестування зображено на рис. 3.5.

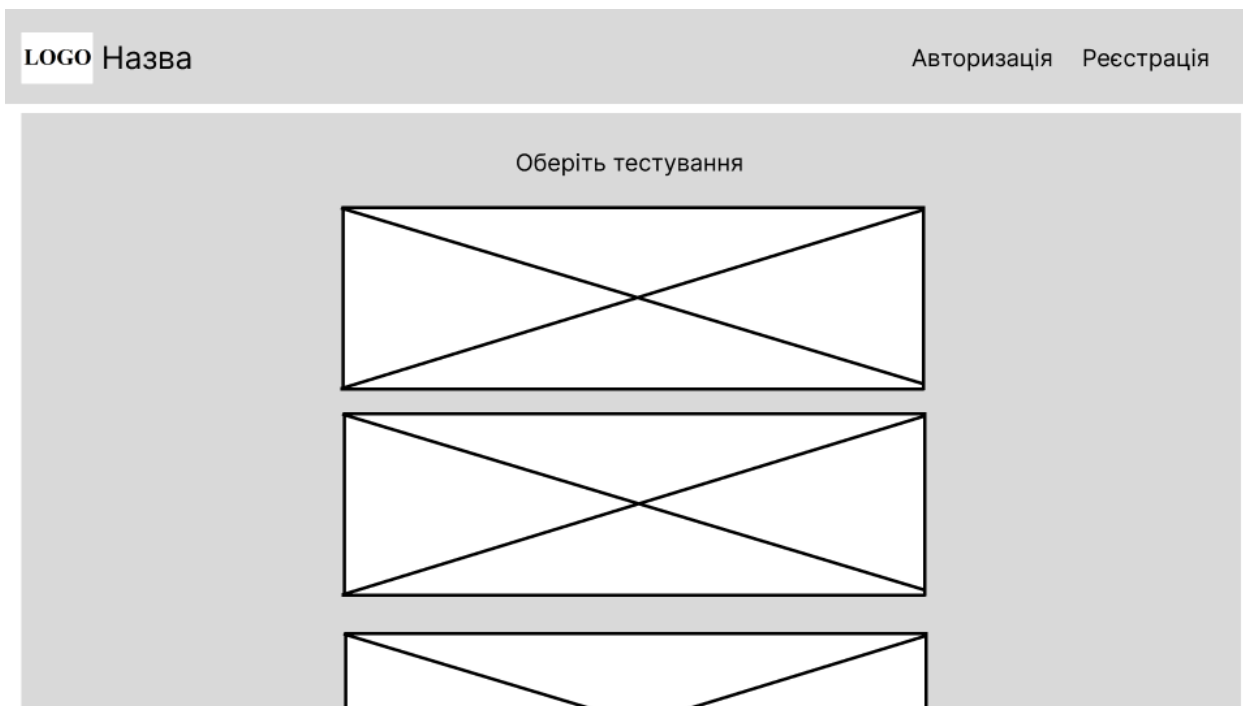


Рисунок 3.5 – Прототип сторінки вибору тестування

Прототип сторінки авторизації зображено на рис. 3.6.

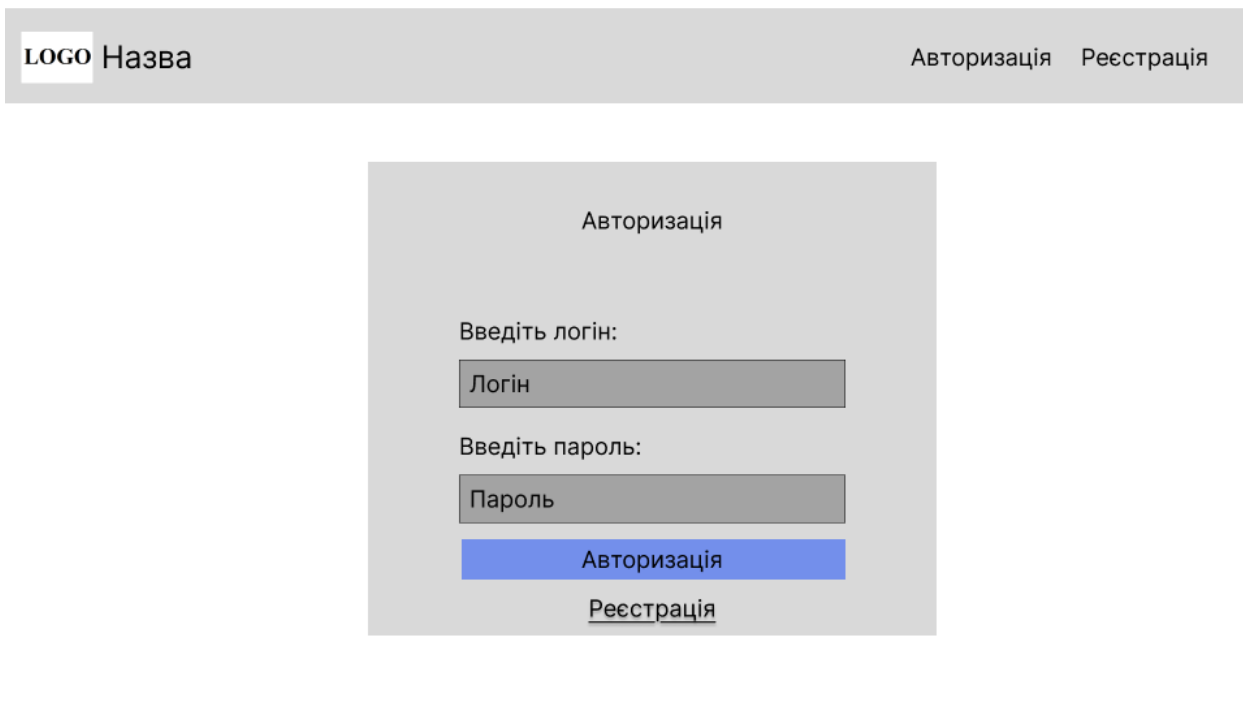


Рисунок 3.6 – Прототип сторінки авторизації

Прототип тестування типу “Читання” зображено на рис. 3.7.

LOGO

Назва

АвторизаціяРеєстрація

Тест читання

Прочитайте текст та дайте відповідь на питання

Test reading text test reading text. Test reading text test reading text. Test reading text test reading text. Test reading text test reading text. Test reading text test reading text. Test reading text test reading text.

Reading question, what is your answer?

Answer 1

Answer 2

Answer 3

Підтвердити відповідь

Рисунок 3.7 – Прототип тестування типу “Читання”

Прототип адмін-панелі зображено на рис. 3.8.

Адмін-панель	
Список таблиць	
Тестування	+
Питання тестування типу "Читання"	+
Питання тестування типу "Аудіювання"	+
Питання тестування типу "Граматика"	+
Питання тестування типу "Словник"	+
Дані про минулі часи	+
Дані про теперішні	

Рисунок 3.8 – Прототип адмін-панелі

3.4 Прецеденти ПЗ

Діаграма прецедентів, що описує взаємодію з системою як користувача, так і адміністратора, зображена на рис. 3.9.

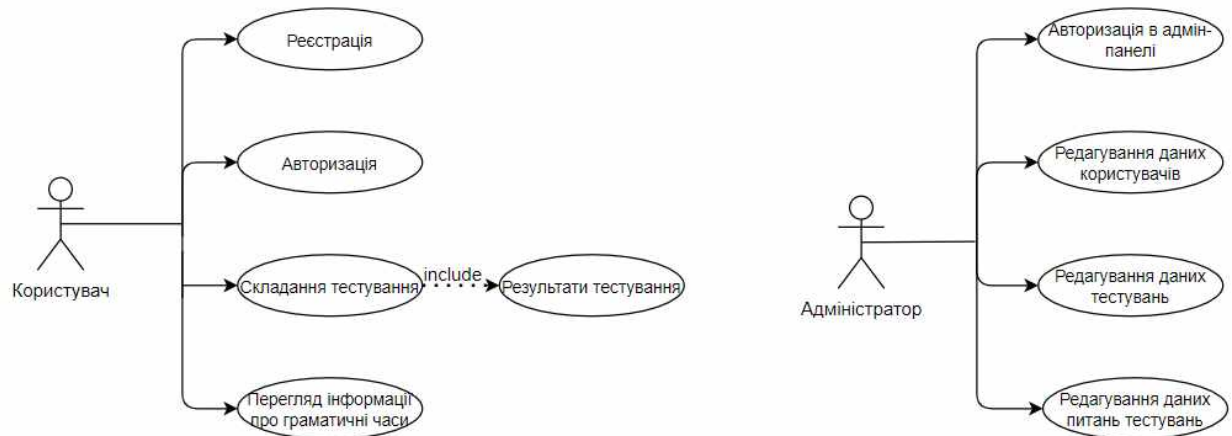


Рисунок 3.9 – Діаграма прецедентів

З розробленої діаграми прецедентів можна зробити висновок, що у вебзастосунку передбачено два типи користувачів:

- звичайний користувач, який має можливість виконувати наступні дії:
 - реєстрація у вебзастосунку;
 - авторизація у вебзастосунку;
 - складання тестування та отримання результатів після того, як він відповість на усі питання;
 - вивчення граматичних часів;
- адміністратор, який має можливість виконувати наступні дії:
 - авторизація в адмін-панелі вебзастосунку;
 - редагування даних користувачів в БД;
 - редагування даних тестувань в БД;
 - редагування даних питань тестувань в БД.

3.5 Характеристики процесу розробки ПЗ

Клієнтська частина з використанням React реалізована за допомогою компонентів, кожен з яких відповідає за власну логіку, зовнішній вигляд та обробку подій, що дозволяє підтримувати зрозумілу структуру. Користувач після обрання тестування або авторизації чи реєстрації надсилає дані на сервер. Взаємодія з сервером реалізована за допомогою HTTP-запитів через бібліотеку axios та функції fetch.

Для реалізації систем користувача і БД використовуються вбудовані можливості Django, що містять готові системи користувача та адміністратора, і ORM технологія завдяки якій можливо ефективно управляти контентом БД без написання складних SQL-запитів.

3.6 Опис розроблених методів ПЗ

3.6.1 Функціонал тестування за навичками

Тестування за навичками реалізованого за допомогою окремих компонентів, що відповідають кожній навичці. Після вибору користувача на головній сторінці типу тестування його перенаправляє на відповідну сторінку в залежності від категорії, де йому необхідно обрати один з тестів, внесених адміністратором застосунку.

Після вибору тестування користувач бачить перед собою наступні дані:

- назва тесту;
- інструкція до завдання;
- текст завдання (у випадку аудіювання – відео);
- питання, на яке потрібно відповісти;
- варіанти відповідей, серед яких лише один правильний;
- після вибору відповіді – кнопка підтвердження.

Якщо користувач правильно відповів на питання, він отримує кількість балів, визначену адміністратором застосунку. У такому випадку задана

відповідь підсвічується зеленим. У випадку неправильної відповіді вона підсвічується червоним та правильна відповідь – зеленим. Додатково, користувачу виводиться пояснення щодо питання. У такому випадку він отримує 0 балів за питання.

Після того, як користувач відповів на усі питання тестування, він бачить результати у вигляді загальної кількості балів. Також йому демонструється повний перелік усіх питань та кількість балів, набраних за кожне питання.

3.6.2 Функціонал перегляду інформації про граматичні часи

Перегляд інформації про граматичних часів було реалізовано в окремому компоненті.

Коли користувач переходить до сторінки перегляду, йому пропонується обрати один з розділів граматичних часів (Present, Past, Future). Після вибору виду додатково необхідно обрати один з підрозділів граматичних часів, що містять дані, внесені адміністратором застосунку.

Після вибору підвиду користувач бачить перед собою навчальний матеріал, що складається з наступних пунктів:

- опис (де використовується, для чого він);
- формула для позитивного, негативного та питального речень;
- приклади для кожного з вищезазначених видів речень.

3.7 Висновки за розділом 3

У даному розділі було розглянуто структуру ПЗ та розроблено БД, описано проєктування інтерфейсу та прецеденти під час використання ПЗ вебзастосунку. Було визначено характеристики процесу розробки та описано розроблені методи.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Вебзастосунок призначений для доступного та простого вивчення користувачами англійської мови за допомогою складань різних видів тестування та вивченню часів (яким чином вони будуються та де використовуються).

Вебзастосунок складається з клієнтської та серверної частини.

Умовою для функціонування програми є стабільна робота обох частин програми одночасно.

4.2 Характеристики програми

ПЗ вебзастосунку для вивчення англійської мови містить клієнтську та серверну частину. Оскільки користувач використовує браузер для роботи, йому не потрібно завантажувати додаткові програми на свій комп'ютер.

Клієнтська частина відповідає за інтерфейсу користувача, що містить в собі компоненти авторизації, реєстрації, тестувань та розділу граматичних часів. Для реалізації було використано бібліотеку React, що дозволяє створювати інтерактивні та динамічні інтерфейси.

Серверна частина відповідає за збереження даних про тестування, користувачів, граматичні часи і оброблює запити від клієнтської частини. Для реалізації було використано Python-фреймворк Django, для API – пакет Django Rest Framework, що дозволяє обмінюватися даними у форматі JSON із клієнтською частиною.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для роботи з клієнтською частиною необхідно запустити додаток та відкрити її у браузері. При цьому також повинна бути запущена серверна частина застосунку. Взаємодіяти з нею можна також через адміністративну панель за допомогою браузера.

4.3.2 Вхідні та вихідні дані

Клієнтська частина:

- реєстрація: нікнейм, пошта, пароль, ім'я, прізвище – вхідні дані, результат реєстрації (успіх чи помилка) – вихідні дані;
- авторизація: нікнейм, пароль – вхідні дані, результат авторизації (успіх чи помилка) – вихідні дані;
- для тестування: вибір правильної відповіді – вхідні дані, результат відповіді (правильна чи неправильна відповідь) – вихідні дані (якщо питання було останнім – вивід результату тестування).

Серверна частина:

Вхідні дані:

- від клієнтської частини: дані про користувача, інформація про тестування або часи англійської мови;
- через адмін-панель: дані про розділи тестування, питання, про часи англійської мови (під час створення).

Вихідні дані:

- від клієнтської частини: успіх під час створення користувача або додавання необхідного питання, тестування або розділу часів.

4.4 Виконання та тестування програми

Тестування даної програми відбувається шляхом відкриття сторінок та взаємодії з ними шляхом внесення як валідних, так і невалідних даних з метою перевірки обробки виключних ситуацій. Передумови тестування – запуснені за допомогою терміналу клієнтська та серверна частини.

4.4.1 Виконання та тестування клієнтської частини

Головна сторінка клієнтської частини зображена на рис. 4.1.

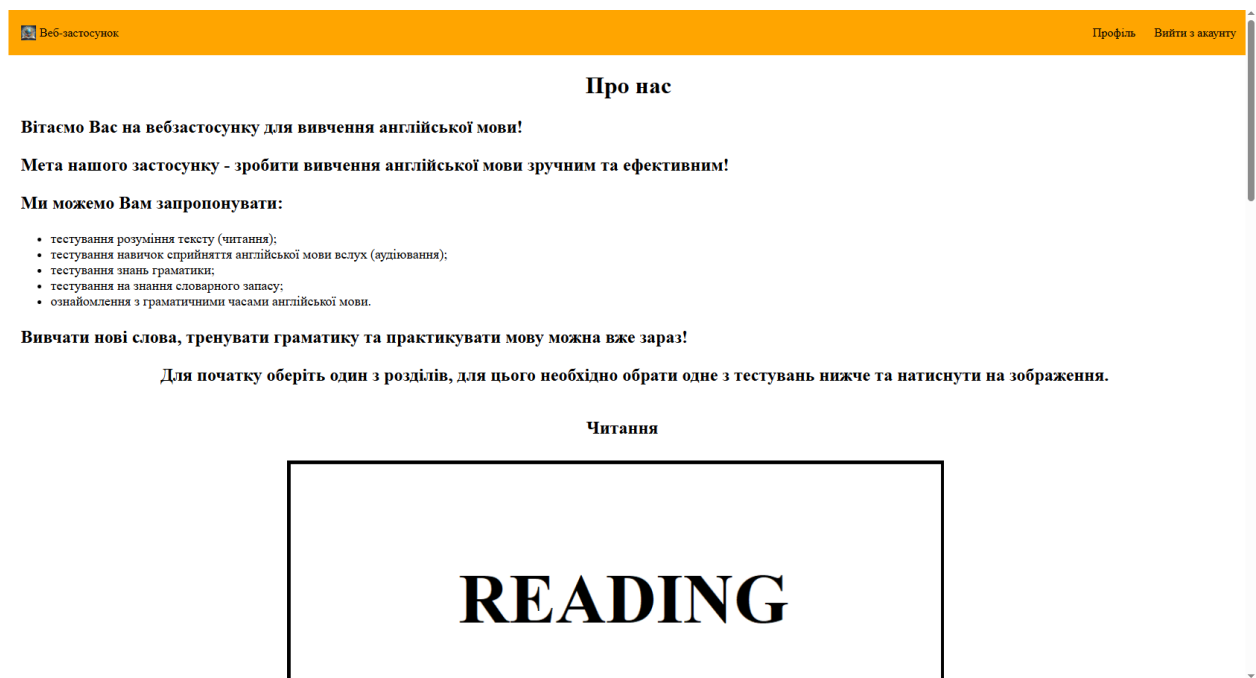


Рисунок 4.1 – Головна сторінка клієнтської частини

Після того, як користувач опинився на головній сторінці, він може зареєструватися, натиснувши на відповідне посилання в шапці застосунку. Форма реєстрації зображена на рис. 4.2.

Веб-застосунок

Авторизація Регістрація

Регістрація

Введіть нікнейм:

Введіть електронну пошту:

Введіть пароль:

Продублюйте пароль:

Введіть ім'я:

Введіть прізвище:

Зареєструватись

[Вже є акаунт? Авторизуйтесь!](#)

Веб-застосунок Розділ тестування Ми у соц. мережах

Рисунок 4.2 – Форма реєстрації користувача вебзастосунку

У випадку, якщо у формі було залишено порожнє поле даних, буде виведена відповідна помилка. Приклад зображено на рис. 4.3.

Веб-застосунок

Авторизація Регістрація

Регістрація

Введіть нікнейм:

test_dpi

Введіть електронну пошту:

Введіть пароль:

Заполните это поле.

Продублюйте пароль:

Введіть ім'я:

test

Введіть прізвище:

Diploma

Зареєструватись

[Вже є акаунт? Авторизуйтесь!](#)

Рисунок 4.3 – Введення невалідних даних під час реєстрації

У випадку, якщо користувач намагається ввести e-mail або нікнейм, який вже зайнято іншим користувачем, буде виведена відповідна помилка. Приклад введення зображено на рис. 4.4.

The screenshot shows a web application interface with an orange header bar. On the left, it says 'Веб-застосунок' and on the right, 'Авторизація' and 'Реєстрація'. The main content area is titled 'Реєстрація'. It contains a form with the following fields and labels: 'Введіть нікнейм:' (with value '12345'), 'Введіть електронну пошту:' (with value 'grebines.oleksandr@gmail.com'), 'Введіть пароль:' (with value '...'), 'Продублюйте пароль:' (with value '...'), 'Введіть ім'я:' (with value 'root'), and 'Введіть прізвище:' (with value 'root'). Below the form, a red error message states 'Цей email вже зареєстрований.' (This email is already registered.). At the bottom of the form, there is a blue button labeled 'Зареєструватись' and a link 'Вже є акаунт? Авторизуйтесь!'.

Рисунок 4.4 – Спроба внести існуючий e-mail в БД під час реєстрації

У випадку введення валідних даних, користувача буде повідомлено про успішність реєстрації та автоматично відкриється форма авторизація, де він зможе внести попередні дані. Форма авторизації зображена на рис. 4.5.

The screenshot shows the same web application interface as Figure 4.4, but the main content area is titled 'Авторизація'. The form contains two fields: 'Введіть нікнейм:' and 'Введіть пароль:'. Below the form, there is a blue button labeled 'Авторизуватись' and a link 'Немає акаунту? Зареєструйтесь!'.

Рисунок 4.5 – Форма авторизації користувача

Випадок введення невалідних даних оброблюється ідентично, як і в формі реєстрації. Якщо ввести некоректні дані користувача (нікнейм або пароль), буде виведено помилку про неправильний логін або пароль. Приклад зображено на рис. 4.6.

The screenshot shows a web application interface with an orange header bar. On the left, it says 'Веб-застосунок' with a small icon. On the right, there are links for 'Авторизація' and 'Реєстрація'. The main content area is titled 'Авторизація'. It contains two input fields: 'Введіть нікнейм:' with the value 'test_dipl' and 'Введіть пароль:' with masked characters '*****'. Below the password field, a red error message reads 'Неправильний нікнейм користувача або пароль'. At the bottom of the form is a blue button labeled 'Авторизуватись' and a link that says 'Немає акаунту? Зареєструйтесь!'.

Рисунок 4.6 – Введення некоректних даних користувача

У випадку введення коректних даних перед користувачем відкривається сторінка профілю, де буде відображено внесені дані під час реєстрації (окрім паролю) та посилання, перейшовши за яким можна обрати розділи тестування. Сторінка профілю зображена на рис. 4.7.

The screenshot shows the user profile page. The orange header bar contains 'Веб-застосунок' on the left and 'Профіль' and 'Вийти з акаунту' on the right. The main content area has a greeting 'Вітаємо, Test Diploma!'. Below it, the following information is displayed: 'Електронна адреса: test_dipl@gmail.com', 'Ім'я: Test', and 'Прізвище: Diploma'. A purple link 'Перейти до тестування' is shown. The footer bar contains 'Веб-застосунок' on the left, 'Розділи тестування' in the center, and 'Ми у соц. мережах' on the right, followed by social media icons.

Рисунок 4.7 – Сторінка профілю

Коли користувач обирає розділ, за яким бажає скласти тестування, йому пропонується обрати один із запронованих тестів. Приклад на основі розділу читання (аналогічно працює з іншими розділами тестування) зображено на рис. 4.8.

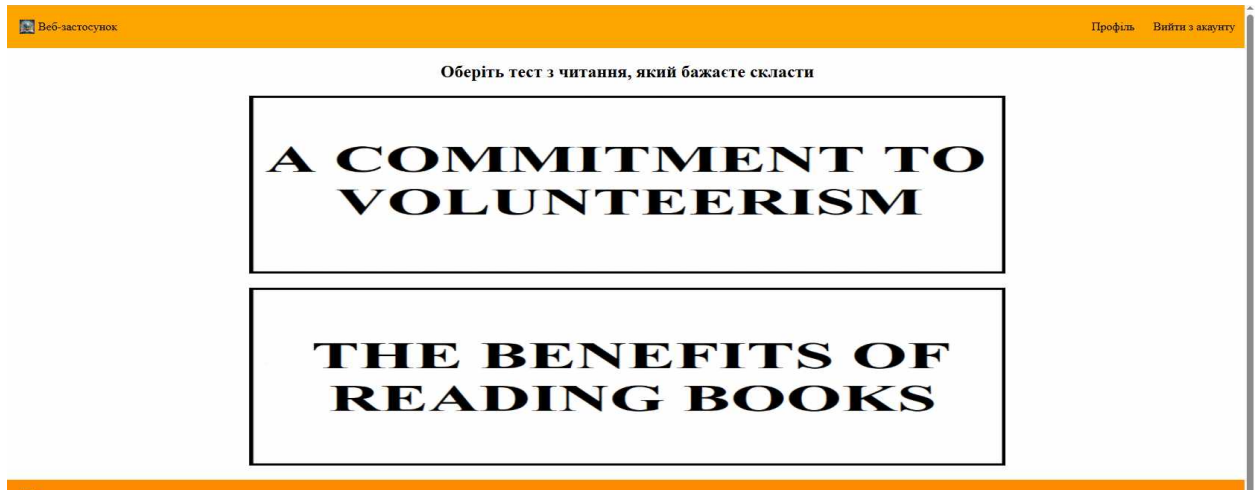


Рисунок 4.8 – Вибір одного із запронованих тестів обраного розділу

Доступ до окремих тестів мають лише авторизовані користувачі. Якщо неавторизований користувач спробує відкрити даний тест, який недоступний для нього, буде виведено повідомлення про помилку. Приклад зображено на рис. 4.9.

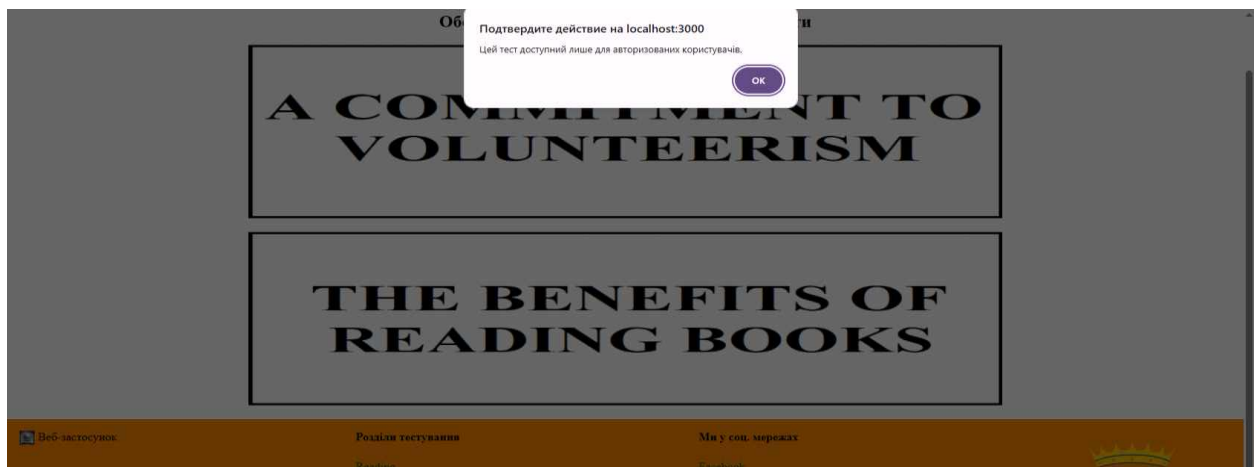


Рисунок 4.9 – Спроба відкриття тесту з обмеженим доступом неавторизованим користувачем

Після вибору тесту користувачу відображається його назва, умова завдання, його текст та питання, на яке потрібно обрати одну із запропонованих відповідей. Приклад зображено на рис. 4.10.



Рисунок 4.10 – Приклад завдання тестування

Після вибору одної із запропонованих відповідей з'являється кнопка підтвердження відповіді по натисканню якої виводиться результат відповіді на питання. Якщо відповідь правильна – відповідь зафарбовується зеленим. Приклад зображено на рис. 4.11.



Рисунок 4.11 – Приклад правильної відповіді на питання

У випадку неправильної відповіді на питання – відповідь зафарбовується зеленим та користувачу буде виведено пояснювальний текст щодо питання. Приклад зображено на рис. 4.12.



Рисунок 4.12 – Приклад неправильної відповіді на питання

Після завершення тестування, користувачу буде виведено результат у балах та написано скільки балів було набрано за кожне питання. Користувач може натиснути кнопку «До вибору розділу» після чого його повертає до моменту вибору тестування. Усі дані про тестування, що надходили від сервера були надіслані до клієнтської частини та оброблені коректно. Приклад сторінки результату тестування зображено на рис. 4.13.

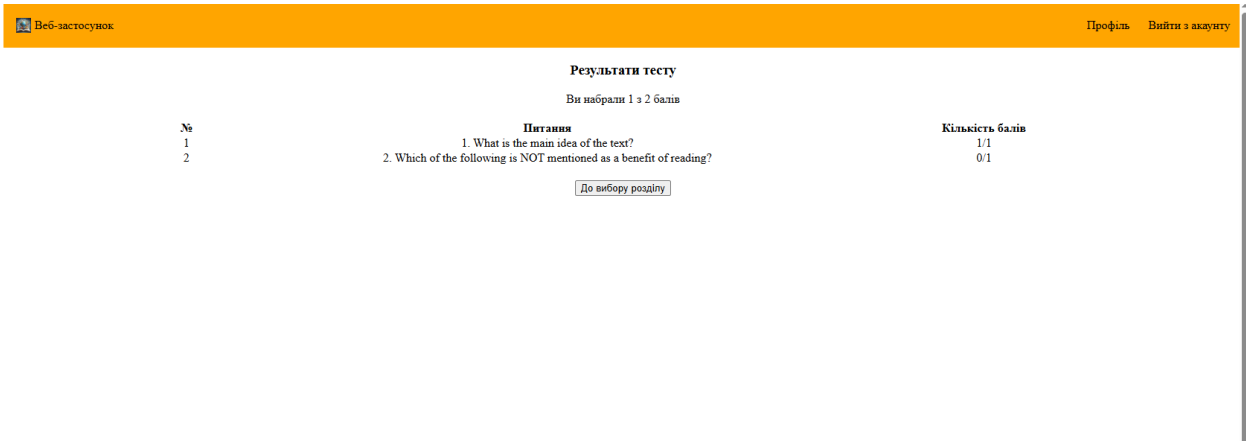


Рисунок 4.13 – Результат тестування користувача

Для інших видів тестування результат є ідентичним, однак у аудіюванні замість тексту питання користувач має прослухати відео на YouTube. Приклад сторінки тестування типу аудіювання зображено на рис. 4.14.

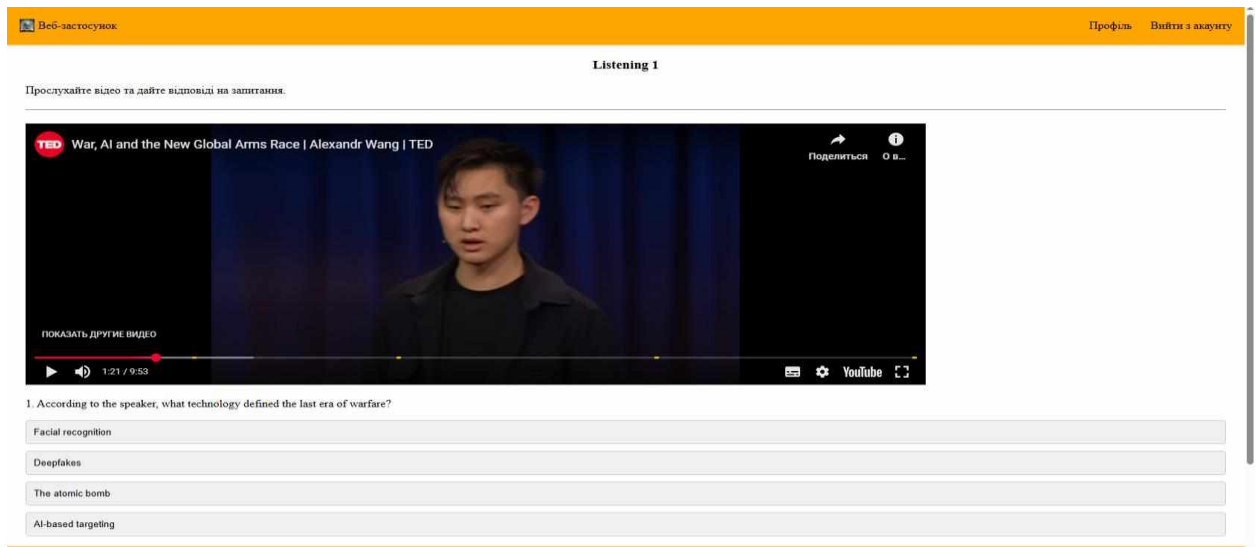


Рисунок 4.14 – Тестування типу аудіювання

Якщо користувач вирішив переглянути розділ про граматичні часи тоді після натискання зображення «Learning Times» йому запропонує обрати один із типів часів (теперішній, минулий, майбутній). Приклад зображено на рис. 4.15.

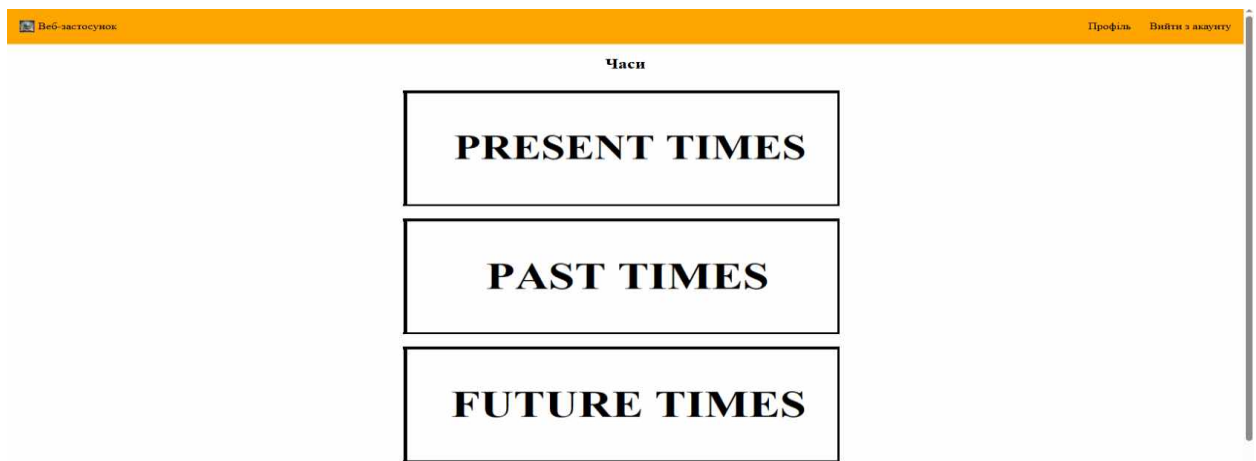


Рисунок 4.15 – Розділ вибору типу часу

Після вибору типу часу користувачу запропонує обрати один із підтипів часу або повернутися назад. Приклад зображено на рис. 4.16.



Рисунок 4.16 – Вибір підтипу часу

Після того, як було обрано підтип, користувач отримає інформацію про час, а саме: де він використовується, формули та приклади для позитивного, негативного та питального речень. Приклад зображено на рис. 4.17.

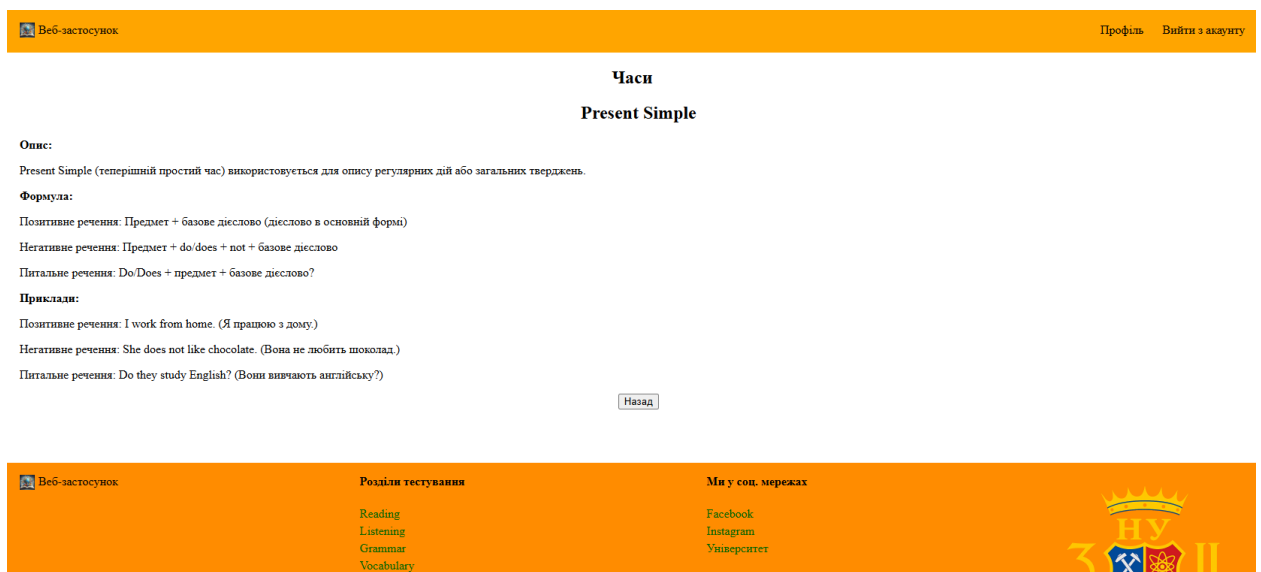


Рисунок 4.17 – Опис підтипу часу

Також було проведено тестування навігації шапки та підвалу застосунку. Переходи на інші сторінки вебзастосунку працюють коректно в обох частинах. Якщо користувач авторизований і бажає вийти з акаунту, за умови натиснення відповідного посилання з'являється сповіщення для підтвердження дії, що зображено на рис. 4.18.

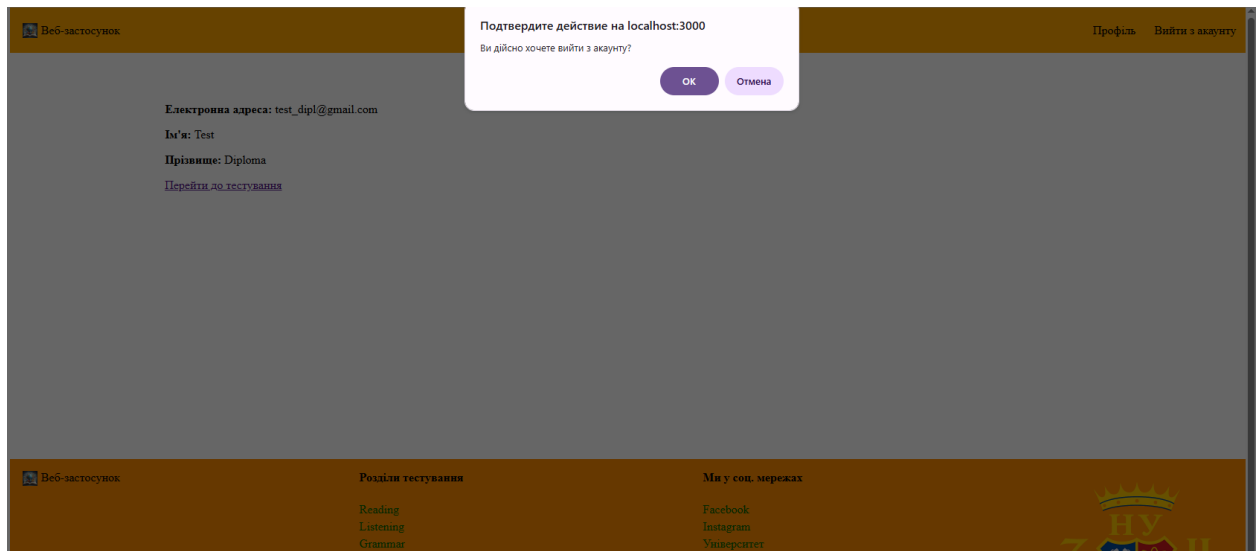


Рисунок 4.18 – Сповіщення підтвердження виходу з акаунту

У результаті тестування клієнтської частини можна зробити висновок, про стабільність роботи інтерфейсу. Усі переходи відбуваються без затримок та на сторінки, що відповідають логіці назви посилань. Це свідчить про правильну реалізацію маршрутизації. Компоненти відображаються коректно та забезпечують належну взаємодію з користувачем застосунку.

4.4.2 Виконання та тестування серверної частини

Після запуску сервера спочатку адміністратору необхідно авторизуватися в адмін-панелі за наданим йому логіном та паролем. Сторінка авторизації зображена на рис. 4.19.

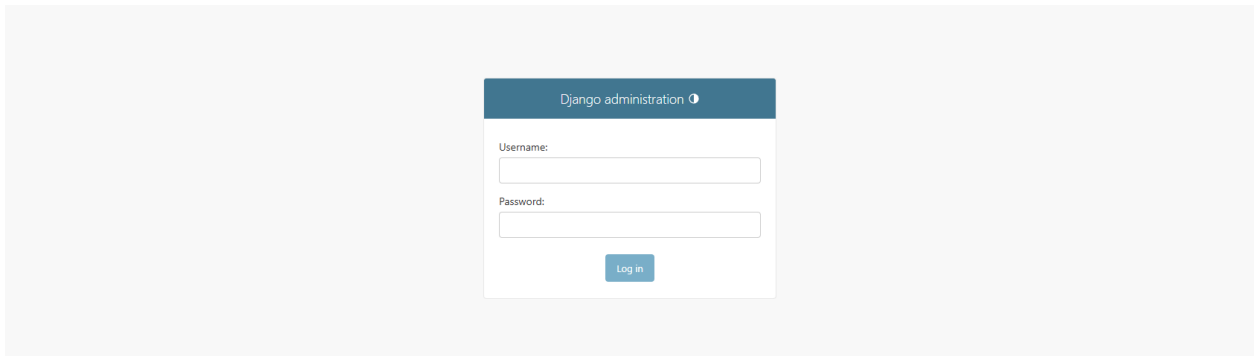


Рисунок 4.19 – Авторизація в адмін-панелі

Після авторизації в адмін-панелі адміністратор бачить перед собою головну сторінку адмін-панелі, що містить таблиці БД про користувачів та тестування, що зображено на рис. 4.20.

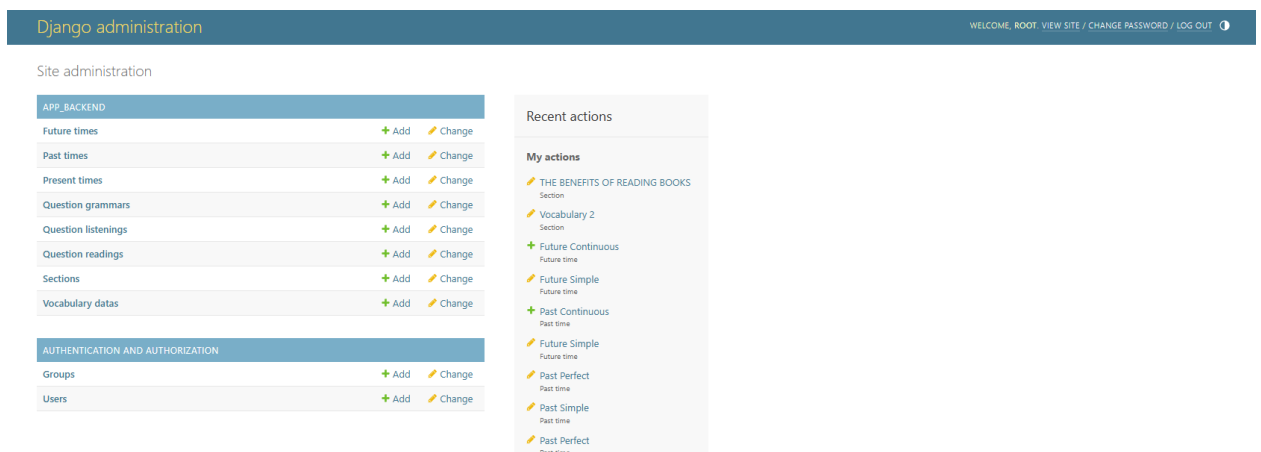


Рисунок 4.20 – Головна сторінка адмін-панелі

На головній сторінці можна обрати модель, що містить інформацію про користувачів або тестування та граматичні часи, де треба змінити дані, або додати нові. У випадку обрання інформації про користувачів, буде відображено повний список усіх користувачів на сервері. У ході тестування клієнтської частини було додано нового користувача через форму, після відправки якої він з'явився у списку, що каже про коректну взаємодію між обома частинами. Приклад сторінки моделі користувача зображено на рис. 4.21.

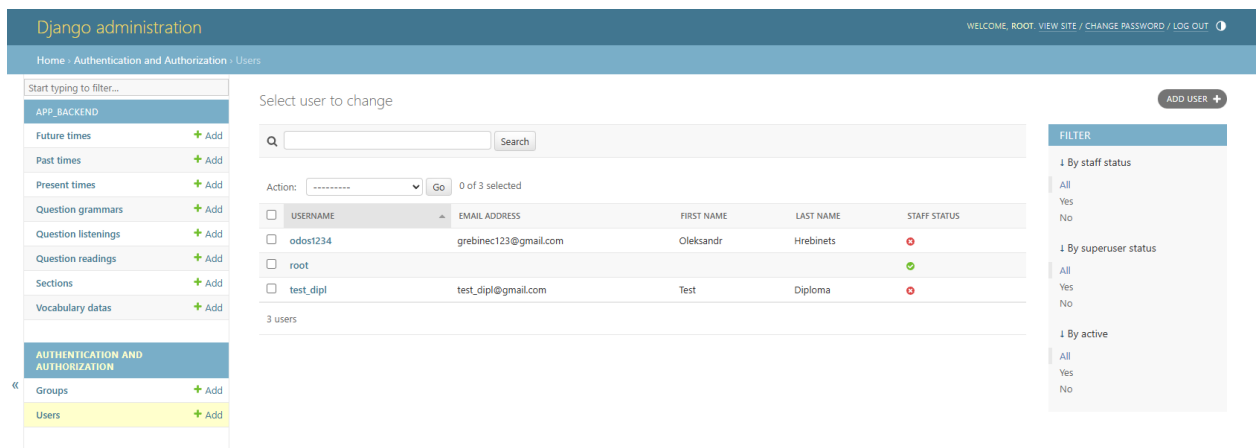


Рисунок 4.21 – Сторінка моделі користувача

За допомогою моделі «Sections» можна визначити нове тестування та визначити наступні параметри:

- назву тестування;
- опис тестування;
- картинку для відображення в клієнтській частині;
- категорію тестування;
- доступність для неавторизованих користувачів (так чи ні).

Інтерфейс внесення даних зображено на рис. 4.22.

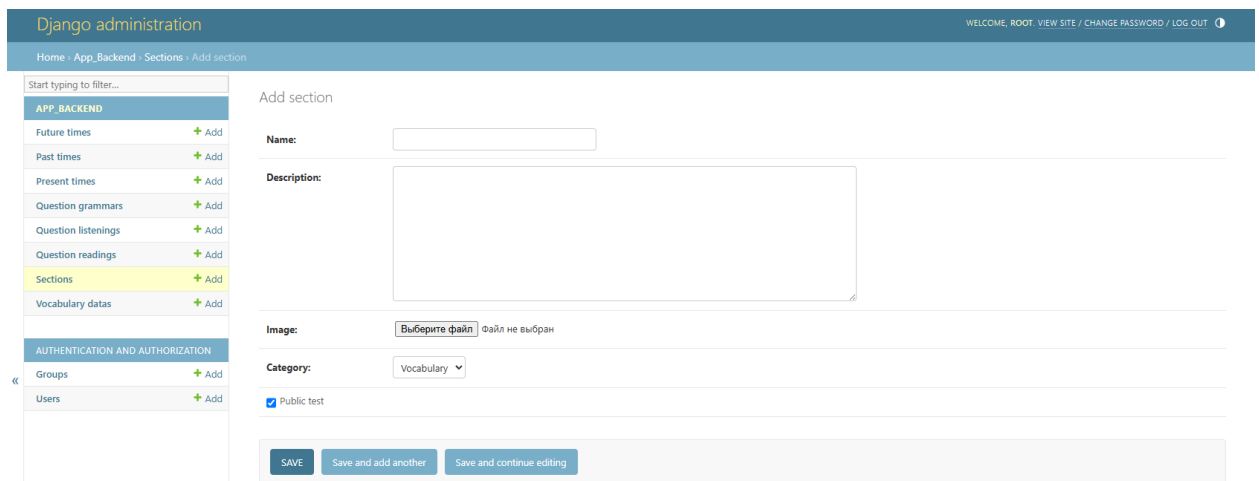


Рисунок 4.22 – Сторінка моделі «Sections»

Моделі “Question ..” та “Vocabulary datas» визначають питання для тестувань. Для внесення даних необхідно:

- визначити тест відповідного розділу, до якого воно буде відноситись;
- відео, що необхідно переглянути для відповіді на питання (для аудіювання);
- написати текст питання;
- відповіді у JSON форматі;
- правильну відповідь серед запропонованих;
- текст, який буде виводитись за умови обрання неправильної відповіді;
- кількість балів за питання.

Інтерфейс внесення даних на прикладі “Question readings” зображено на рис. 4.23.

Рисунок 4.23 – Сторінка моделі “Question readings”

Для моделі “Question listenings” додається ще одне поле з посиланням на відео для відповіді на питання, що зображено на рис. 4.24.

Рисунок 4.24 – Сторінка моделі “Question listenings”

Оскільки усі поля є обов’язковими для заповнення даними відповідного формату, то наявність хоча б одного порожнього виведе повідомлення про помилку. Для поля “Answers” обов’язково треба внести дані JSON формату. Приклад виведення помилки зображено на рис. 4.25.

The screenshot shows a web application interface for adding a question listening. On the left is a sidebar menu with categories like APP_BACKEND, AUTHENTICATION AND AUTHORIZATION, and others. The main area is titled 'Add question listening'. At the top, there's a red box saying 'Please correct the errors below.' Below this, several fields are listed with their respective error messages:

- Section:** This field is required.
- VideoSrc:** This field is required.
- Question:** This field is required.
- Answers:** Enter a valid JSON. not json format
- Correct:** This field is required.
- ErrorText:** This field is required.

Рисунок 4.25 – Приклад виведення помилки

У результаті тестування серверної частини можна зробити висновок, про стабільність роботи сервера. Запити користувача ефективно обробляються без затримок. Дані на клієнтську частину надходять без жодних втрат та помилок. Сервер демонструє стабільну та надійну роботу та забезпечує безперервну обробку даних.

4.5 Висновки за розділом 4

У даному розділі було розглянуто призначення та умови застосування програми. Визначено характеристики, спосіб звернення до програми та вхідні й вихідні дані.

Було проведено тестування клієнтської і серверної частини, з якого можна зробити наступні висновки:

- інтерфейс клієнтської частини правильно реагує на усі дії користувача;
- обидві частини системи стабільно функціонують та взаємодіють між собою під час роботи зі застосунком;
- під час тестування не було виявлено несподіваних помилок, що могли б зупинити роботу системи;
- дані між обома частинами передаються без втрат;
- інтерфейс клієнтської частини правильно реагує на усі дії користувача.

Загалом, це свідчить про надійність та коректність роботи розробленого програмного засобу.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи було проаналізовано предметну область та процес розробки програмного забезпечення вебзастосунку для вивчення англійської мови. Досліджено існуючі аналоги та на основі дослідження було визначено функціональні вимоги.

Для розробки програмного забезпечення було використано бібліотеку React для клієнтської частини та фреймворк Django для серверної.

Розроблено програмне забезпечення вебзастосунку для вивчення англійської мови. Описано структуру, функціонування та прецеденти програмного забезпечення, визначено характеристики процесу розробки.

Виконано тестування програмного забезпечення, у ході якого було зроблено висновок, що програма виконує задані функції та вимоги.

Завдання дипломної кваліфікаційної роботи бакалавра були повністю виконані.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. The Role of the English Language in the World and its Importance Today [Electronic resource]. – Access mode: <https://scientificbulletin.com/index.php/AJOUP/article/view/85/68>.
2. Як українці вивчають англійську мову: результати дослідження [Електронний ресурс]. – Режим доступу: <https://www.kmu.gov.ua/news/yak-ukraintsi-vyvchaiut-anhliisku-movu-rezultaty-doslidzhennia>.
3. Інтернет – Вікіпедія [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82>.
4. Challenges and opportunities related to online studies in higher education [Electronic resource]. – Access mode: <https://link.springer.com/article/10.1007/s10639-025-13406-x>.
5. Teaching and learning languages online: Challenges and responses [Electronic resource]. – Access mode: <https://doi.org/10.1016/j.system.2022.1028195>
6. Learn English Online | British Council [Electronic resource]: – Access mode: <https://learnenglish.britishcouncil.org/>
7. GrammarWay – Граматика англійської мови [Електронний ресурс]: – Access mode: <https://grammarway.com/>.
8. Learning English | Cambridge English [Electronic resource]: – Access mode: <https://www.cambridgeenglish.org/learning-english/>
9. React [Electronic resource]: – Access mode: <https://react.dev/>.
10. Quick Start – React [Electronic resource]: – Access mode: <https://react.dev/learn>.
11. Getting Started | Axios Docs [Electronic resource]: – Access mode: <https://axios-http.com/docs/intro>.
12. What are the features of ReactJS? [Electronic resource]: – Access mode: <https://www.geeksforgeeks.org/what-are-the-features-of-reactjs/>

13. The web framework for perfectionists with deadlines | Django [Electronic resource]: – Access mode: <https://www.djangoproject.com/>
14. Security in Django | Django Documentation [Electronic resource]: – Access mode: <https://docs.djangoproject.com/en/5.2/topics/security/>.
15. The Django admin site | Django Documentation [Electronic resource]: – Access mode: <https://docs.djangoproject.com/en/5.2/ref/contrib/admin/>.
16. Models | Django Documentation [Electronic resource]: – Access mode: <https://docs.djangoproject.com/en/5.2/topics/db/models>.
17. Home – Django Rest Framework [Electronic resource]: – Access mode: <https://www.django-rest-framework.org/>.
18. URL Dispatcher | Django Documentation [Electronic resource]: – Access mode: <https://docs.djangoproject.com/en/5.2/topics/http/urls/>.
19. Simple JWT [Electronic resource]: – Access mode: <https://django-rest-framework-simplejwt.readthedocs.io/en/latest/>
20. Tutorial: Get started with Visual Studio Code [Electronic resource]: – Access mode: <https://code.visualstudio.com/docs/getstarted/getting-started>.
21. Quick Start Guide | PyCharm Documentation [Electronic resource]: – Access mode: <https://www.jetbrains.com/help/pycharm/quick-start-guide.html>.

ДОДАТОК А
Технічне завдання

Вступ

Назва програми – «Вебзастосунок для вивчення англійської мови».

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Розробка вебзастосунку для вивчення англійської мови», затверджене наказом Національного університету «Запорізька політехніка» №209 від 28 квітня 2025 р.

A.2 Призначення розробки

Вебзастосунок призначений для доступного та простого вивчення користувачами англійської мови за допомогою складань різних видів тестування та вивченню часів (яким чином вони будуються та де використовуються).

A.3 Вимоги до програми

A.3.1 Функціональні вимоги

Вебзастосунок має відповідати наступним функціональним вимогам:

- можливість реєстрація та авторизація користувачів;
- можливість редагування даних про користувачів та тестування;
- адміністративна панель для внесення даних;
- тестування для перевірки знань користувача (читання, слухання, граматики, словник);
- розділ з граматичними часами англійської мови.

A.3.2 Вимоги до інтерфейсу програми

Інтерфейс повинен бути зручним для користувачів, мати просту навігацію між розділами. Елементи застосунку мають бути логічно розташовані для швидкого освоєння.

A.3.3 Вимоги до надійності програмного забезпечення

Програмне забезпечення вебзастосунку для вивчення англійської мови повинно стабільно функціонувати та містити обробку збійних ситуацій та різноманітних помилок.

A.3.4 Вимоги до технічних та програмних засобів

Для якісної роботи програми необхідні наступні компоненти зі сторони розробника:

- від 4 ГБ оперативної пам'яті;
- стабільне з'єднання з мережею Інтернет;
- версія Python 3.10 або вище;
- версія React 18.0 (підтримка хуків);
- версія Django 4.0 або вище.

Зі сторони користувача необхідні наступні компоненти:

- браузер, в якому є увімкнений JavaScript;
- стабільне з'єднання з мережею Інтернет.

A.4 Вхідні та вихідні дані

Клієнтська частини:

- реєстрація: нікнейм, пошта, пароль, ім'я, прізвище – вхідні дані, результат реєстрації (успіх чи помилка) – вихідні дані;

- авторизація: нікнейм, пароль – вхідні дані, результат авторизації (успіх чи помилка) – вихідні дані;
- для тестування: вибір правильної відповіді – вхідні дані, результат відповіді (правильна чи неправильна відповідь) – вихідні дані (якщо питання було останнім – вивід результату тестування).

Серверна частина:

Вхідні дані:

- від клієнтської частини: дані про користувача, інформація про тестування або часи англійської мови;
- через адмін-панель: дані про розділи тестування, питання, про часи англійської мови (під час створення).

Вихідні дані:

- від клієнтської частини: успіх під час створення користувача або додавання необхідного питання, тестування або розділу часів.

ДОДАТОК Б
Фрагмент тексту програми

Models.py

```
class Section(models.Model):
    name = models.CharField(max_length=100, unique=True)
    description = models.TextField()
    image = models.ImageField()
    choices = [
        ('vocabulary', 'Vocabulary'),
        ('reading', 'Reading'),
        ('listening', 'Listening'),
        ('grammar', 'Grammar'),
    ]
    category = models.CharField(max_length=20, choices=choices,
default='vocabulary')

    public_test = models.BooleanField(default=True)

    def __str__(self):
        return self.name

class QuestionReading(models.Model):
    section = models.ForeignKey(Section, related_name="questions_reading",
on_delete=models.CASCADE)
    question = models.TextField()
    answers = models.JSONField()
    correct = models.CharField(max_length=255)
    errorText = models.TextField()
    number_points = models.IntegerField(default=1)

    def __str__(self):
        return self.question
```

```

class QuestionGrammar(models.Model):
    section = models.ForeignKey(Section, related_name="questions_grammar",
on_delete=models.CASCADE)
    question = models.TextField()
    answers = models.JSONField()
    correct = models.CharField(max_length=255)
    errorText = models.TextField()
    number_points = models.IntegerField(default=1)

    def __str__(self):
        return self.question

```

```

class QuestionListening(models.Model):
    section = models.ForeignKey(Section, related_name="questions_listening",
on_delete=models.CASCADE)
    videoSrc = models.URLField()
    question = models.TextField()
    answers = models.JSONField()
    correct = models.CharField(max_length=255)
    errorText = models.TextField()
    number_points = models.IntegerField(default=1)

    def __str__(self):
        return self.question

```

```

class VocabularyData(models.Model):
    section = models.ForeignKey(Section, related_name="vocabulary_data",
on_delete=models.CASCADE)

```

```
question = models.TextField()
answers = models.JSONField()
correct = models.CharField(max_length=255)
errorText = models.TextField()
number_points = models.IntegerField(default=1)
```

```
def __str__(self):
    return self.question
```

```
class PresentTime(models.Model):
    name = models.CharField(max_length=100, unique=True)
    image = models.ImageField()
    description = models.TextField()
    formula = models.TextField()
    examples = models.TextField()
```

```
def __str__(self):
    return self.name
```

```
class PastTime(models.Model):
    name = models.CharField(max_length=100, unique=True)
    image = models.ImageField()
    description = models.TextField()
    formula = models.TextField()
    examples = models.TextField()
```

```
def __str__(self):
    return self.name
```

```
class FutureTime(models.Model):
```

```

name = models.CharField(max_length=100, unique=True)
image = models.ImageField()
description = models.TextField()
formula = models.TextField()
examples = models.TextField()

```

```

def __str__(self):
    return self.name

```

Header.js

```

const Header = () => {
    const { user, logout } = useContext(AuthContext);
    const navigate = useNavigate();

    const Logout = () => {
        if(window.confirm('Ви дійсно хочете вийти з акаунту?')) {
            logout();
            navigate('/');
        }
    };

    return (
        <header className="header">
            <div className="header-logo">
                <img className="img" src={logo} alt="Лого" />
                <Link to="/" className="header-title">Веб-застосунок</Link>
            </div>

            <nav className="header-nav">
                {!user ? (

```

```

    <>
      <Link to='/login'>Авторизація</Link>
      <Link to='/register'>Реєстрація</Link>
    </>
  ):(
    <>
      <Link to='/profile'>Профіль</Link>
      <span onClick={Logout} style={{cursor: "pointer"}}>Вийти з
акаунту</span>
    </>
  )}
</nav>
</header>
);
};

```

```
export default Header;
```

Footer.js

```

const Footer = () => {

  return (
    <footer className="footer">
      <div className='footer-column'>
        <div className="footer-logo-web">
          <img className="img" src={logo} alt="Лого" style={{ height: '20px'
}} />
          <Link to="/" style={{ color: 'black'}}>Веб-застосунок</Link>
        </div>
      </div>
    </div>
  )
}

```

```

<div className='footer-column'>
  <h4>Skills</h4>
  <Link to='/reading'>Reading</Link>
  <Link to='/listening'>Listening</Link>
  <Link to='/grammar'>Grammar</Link>
  <Link to='/vocabulary'>Vocabulary</Link>
  <Link to='/times'>Times</Link>
</div>

```

```

<div className='footer-column'>
  <h4>Ми у соц. мережах</h4>
  <a href='https://facebook.com' target="_blank"
rel="nofollow">Facebook</a>
  <a href='https://instagram.com' target="_blank"
rel="nofollow">Instagram</a>
  <a href="https://zp.edu.ua/" target="_blank"
rel="nofollow">Університет</a>
</div>

```

```

<div className='footer-column'>
  <img className="img-nuzp" src={nuzp} alt="НУЗП"/>
</div>

```

```

<div className="footer-copyright">
  © 2025 Веб-застосунок для дипломної роботи.
</div>
</footer>

```

```
);
```

```
}
```

```
export default Footer;
```

```
MainPage.js
```

```
const MainPage = () => {
  useEffect(() => {
    document.title = "Головна сторінка";
  }, [] );

  return (
    <div>
      <div className='main-text'>
        <p><strong>Вітаємо Вас на вебзастосунку для вивчення англійської мови!</strong></p>
        <p><strong>Мета нашого застосунку - зробити вивчення англійської мови зручним та ефективним!</strong></p>
        <p><strong>Ми можемо Вам запропонувати:</strong></p>
        <ul>
          <li>тестування розуміння тексту перед очима;</li>
          <li>тестування сприйняття англійської мови вслух;</li>
          <li>тестування знань граматики;</li>
          <li>тестування на знання словарного запасу;</li>
          <li>розділ, де Ви можете ознайомитись з граматичними часами англійської мови.</li>
        </ul>
        <p><strong>Вивчати нові слова, тренувати граматику та практикувати мову Ви можете використовуючи наш застосунок просто зараз!</strong></p>
      </div>
    </div>
  )
}
```

<p style={{textAlign: "center"}}>Для початку оберіть один з розділів, для цього необхідно обрати одне з тестувань нижче та натиснути на зображення.</p>

</div>

<div className='main-text-img'>

<h2>Читання</h2>

<Link to='/reading'></Link>

<h2>Аудіювання</h2>

<Link to='/listening'></Link>

<h2>Граматика</h2>

<Link to='/grammar'></Link>

<h2>Словник</h2>

<Link to='/vocabulary'></Link>

<h2>Граматичні часи</h2>

<Link to='/times'></Link>

</div>

</div>

);

}

export default MainPage

admin.py

admin.site.register(Section)

admin.site.register(QuestionGrammar)

admin.site.register(QuestionListening)

admin.site.register(QuestionReading)

admin.site.register(VocabularyData)

admin.site.register(PresentTime)

admin.site.register(FutureTime)

admin.site.register(PastTime)

ДОДАТОК В
Слайди презентації



Рисунок В.1 – Слайд 1

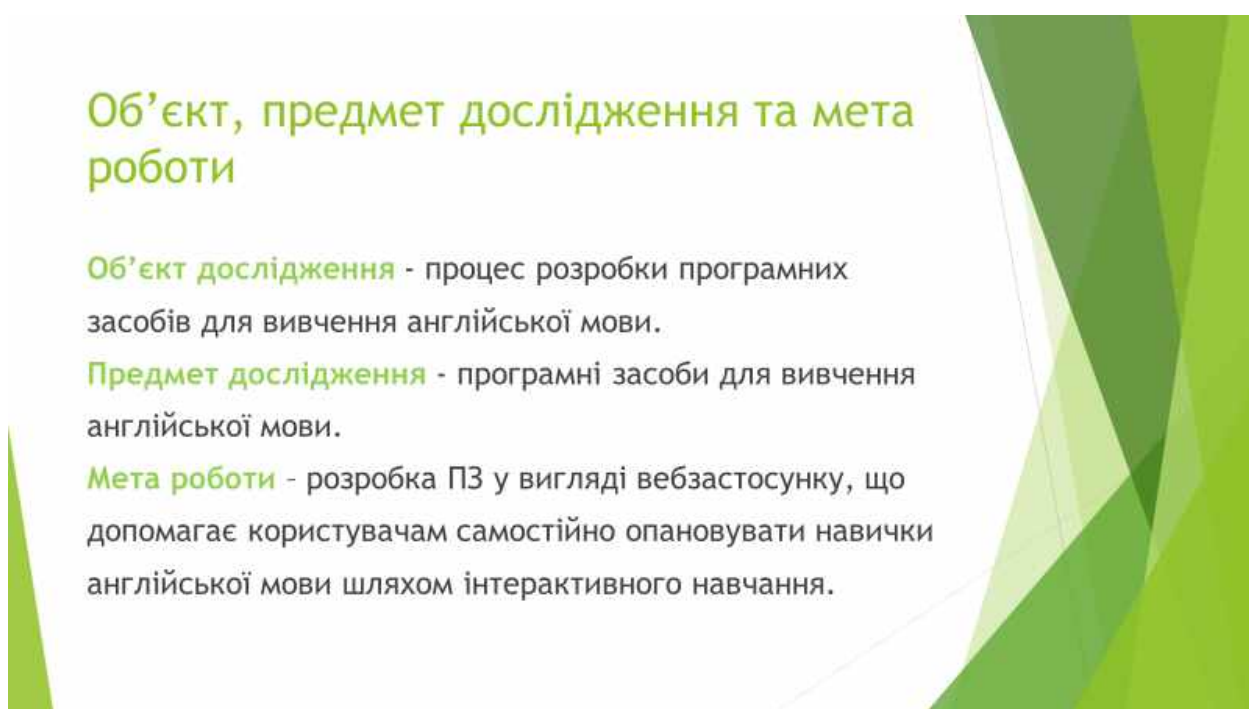


Рисунок В.2 – Слайд 2

Завдання до роботи

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

- виконати аналіз предметної області та існуючих програмних засобів для вивчення англійської мови;
- визначитися з засобами розробки, структурою та функціонуванням програмного забезпечення;
- розробити програмне забезпечення у вигляді вебзастосунку для вивчення англійської мови;
- виконати тестування розробленого програмного забезпечення.

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій	Learn English Online British Council	GrammarWay	Learning English Cambridge English
Наявність активностей	+	-	+
Наявність інформації про граматичні часи	+	+	-
Наявність безкоштовного контенту	+	+	+
Наявність української локалізації	-	+	-
Відсутність реклами	+	+	+

Рисунок В.4 – Слайд 4

Постановка завдання

Метою даної роботи є розробка вебзастосунку для вивчення англійської мови, який включає в себе як клієнтську частину, за допомогою якої користувач взаємодіє із системою та покращує свої знання англійської так і серверну частину, яка дозволяє адміністраторам керувати вмістом застосунку.

Вебзастосунок має відповідати наступним функціональним вимогам:

- можливість реєстрація та авторизація користувачів;
- можливість редагування даних про користувачів та тестування;
- адміністративна панель для внесення даних;
- тестування для перевірки знань користувача (читання, слухання, граматика, словник);
- розділ з граматичними часами англійської мови.

Рисунок В.5 – Слайд 5

Порівняння front-end засобів

Критерій	React	Vue.js	Angular
Документація	++	+	+
Продуктивність	++	++	+
Гнучкість	++	++	+
Спільнота	++	+	++
Масштабованість	++	+	++

Рисунок В.6 – Слайд 6

Порівняння back-end засобів

Критерій	Django	Node.js	Laravel
Наявність влаштованої бази даних	+	-	-
Безпека	+	+	+
Наявність ORM	+ (вбудований)	+ (окрема бібліотека)	+ (вбудований)
Наявність вбудованої адмін-панелі	+	-	-
Масштабованість	+	+	+

Рисунок В.7 – Слайд 7

Порівняння інтегрованих середовищ розробки

Критерій	Visual Studio Code	PyCharm	Eclipse
Кросплатформ-леність	+	+	+
Мови програмування	C#, JavaScript, Python, C++ тощо	Python	Java, C/C++
Зручність використання	++	++	+-
Інтеграція з Git	+	+	+ (через плагін EGit)
Продуктивність	++	++	+-

Рисунок В.8 – Слайд 8

Структура програмного забезпечення

```

graph LR
    БД[(БД)] <--> Серверна_частина[Серверна частина]
    Серверна_частина <--> Клієнтська_частина[Клієнтська частина]
  
```

Рисунок В.9 – Слайд 9

[illegible]

Рисунок В.10 – Слайд 10

Діаграма прецедентів

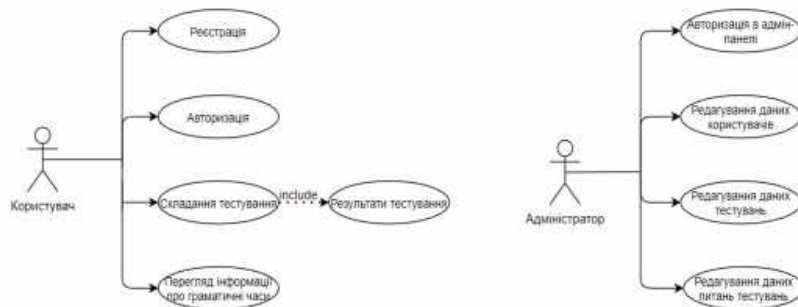


Рисунок В.11 – Слайд 11

Інтерфейс клієнтської частини вебзастосунку



Рисунок В.12 – Слайд 12

Інтерфейс серверної частини вебзастосунку



Рисунок В.13 – Слайд 13

Висновки

В ході виконання дипломної кваліфікаційної роботи було проаналізовано предметну область та процес розробки програмного забезпечення вебзастосунку для вивчення англійської мови. Досліджено існуючі аналоги та на основі дослідження було визначено функціональні вимоги.

Для розробки програмного забезпечення було використано бібліотеку React для клієнтської частини та фреймворк Django для серверної.

Розроблено програмне забезпечення вебзастосунку для вивчення англійської мови. Описано структуру, схему бази даних та прецеденти програмного забезпечення, визначено характеристики процесу розробки.

Виконано тестування програмного забезпечення, у ході якого було зроблено висновок, що програма виконує задані функції та вимоги.

Завдання дипломної кваліфікаційної роботи бакалавра були повністю виконані.

Рисунок В.14 – Слайд 14