

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки та програмної інженерії
Кафедра інженерії програмного забезпечення

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
Катерина НЕСТЕРЕНКО
“ ” 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
“БАКАЛАВР”

Тема: Web-застосунок створення комунікаційних майданчиків
з вивчення англійської мови

Виконавець: Волошин Віталій Вадимович

Керівник: к.т.н. Талалаєв Володимир Опанасович

Нормоконтролер: Варнавський В'ячеслав Володимирович

Київ 2024

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки та програмної інженерії
Кафедра інженерії програмного забезпечення
Освітній ступінь бакалавр
Спеціальність 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма «Програмне забезпечення систем»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Катерина НЕСТЕРЕНКО
" ____ " _____ 2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи студента
Волошина Віталія Вадимовича

1. Тема проєкту: «Web-застосунок створення комунікаційних майданчиків з вивчення англійської мови»
затверджена наказом ректора від 8.12.2023 р. № 2483/ст
2. Термін виконання проєкту: з 3.01.2024 р. до 29.02.2024 р.
3. Вихідні дані до проєкту: програмний продукт розробити у вигляді вебзастосунка для вивчення англійської мови.
4. Зміст пояснювальної записки:
 1. Аналіз предметної області.
 2. Обґрунтування вибору технологій для реалізації програмного застосунку.
 3. Програмна реалізація застосунку.
 4. Робота користувача з розробленою системою.
5. Перелік обов'язкових слайдів презентації:
 1. -
 2. -
 3. -
 4. -
 5. -
 6. -

6. Календарний план-графік

№ пор	Завдання	Термін виконання	Відмітка про виконання
1.	Складання та затвердження графіку кваліфікаційної роботи Написання 1 розділу, представлення керівнику	03.01.22 – 09.01.22	
2.	Попередній друк 1 розділу та допоміжних сторінок (чернетка) — титульної, завдання, графіка, реферат, список скорочень, зміст, вступ, список джерел, 1-й нормоконтроль.	10.01.22 – 17.01.22	
3.	Написання 2 розділу, представлення керівнику	18.01.22 – 22.01.22	
4.	Написання 3 розділу, представлення керівнику	23.01.22 – 26.01.22	
5.	Написання 4 розділу, представлення керівнику	27.01.22 – 31.01.22	
6.	Загальне редагування та друк пояснювальної записки, графічного матеріалу	01.02.22 – 09.02.22	
7.	Проходження нормоконтролю, перевірка на антиплагіат, перепліт пояснювальної записки.	01.02.22 – 05.02.22	
8.	Розробка тексту доповіді. Оформлення графічного матеріалу для презентації	05.02.22 – 09.02.22	
9.	Отримання відгуку керівника, рецензії.	10.02.22 – 22.02.22	
10.	Підготовка матеріалів для передачі секретарю ДЕК (ПЗ, CD-R з електронними копіями ПЗ, презентації, відгук керівника, рецензія) в папці	23.02.22 – 28.02.22	

7. Дата видачі завдання 08.12.2024

Керівник:

Завдання прийняв до виконання:

Дата

к.т.н. Володимир ТАЛАЛАЄВ

Віталій ВОЛОШИН

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Web-застосунок створення комунікаційних майданчиків з вивчення англійської мови»: 52 с., 34 рис., 3 табл., 22 інформаційних джерел.

АВТОМАТИЗОВАНИЙ ПОШУК, ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ, ТЕХНОЛОГІЧНИЙ СТЕК MERN, WEB-ЗАСТОСУНОК, КОМУНІКАЦІЙНИЙ МАЙДАНЧИК

Об'єкт розробки — вебзастосунок для пошуку співрозмовників з вивчення англійської мови на базі технологічного стека MERN.

Мета роботи — розробити комунікаційний майданчик у формі вебзастосунка з вивчення англійської мови.

ABSTRACT

Explanatory note to the thesis "English learning web application in the shape of a social network": 52 p., 34 Fig., 3 table., 22 information sources.

AUTOMATED SEARCH, ENGLISH LANGUAGE LEARNING, MERN STACK, WEB APPLICATION, COMMUNICATION PLATFORM

Property development – a web application for connecting with English language learners based on the MERN stack.

Purpose – to develop a communication platform for learning English in form of a web application

ЗМІСТ

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1. Поняття вебзастосунка.....	10
1.2. Задача розробки вебзастосунка для створення комунікаційного майданчика з вивчення англійської мови.....	12
1.2.1. Знаходження користувачів за певними критеріями.....	13
1.2.2. Перегляд профілів користувачів системи та надсилання запитів на спілкування.....	14
1.3. Аналіз наявних систем.....	14
1.3.1. Платформа з вивчення англійської Hellotalk.....	15
1.3.2. Вивчення англійської мови за допомогою застосунку Coffee Strap.	16
1.3.3. Платформа для вивчення мови Mylanguageexchange.....	18
Висновки.....	19
РОЗДІЛ 2. ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАСТОСУНКУ.....	20
2.1. Технологічний стек MERN.....	21
2.1.1. База даних MongoDB.....	22
2.1.2. Фреймворк Express.js.....	24
2.1.3. Бібліотека React.....	25
2.1.4. Платформа Node.js.....	26
2.2. Мова програмування JavaScript.....	27
2.3. Мова програмування TypeScript.....	28
2.4. Колекція готових компонентів Shadcn/ui.....	29
2.5. Середовище розробки Visual Studio Code.....	30
2.6. Система керування версіями Git та вебсервіс Github.....	31

Висновки.....	33
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ	34
3.1. Процес автентифікації.....	34
3.2. Автоматизація пошуку користувачів за подібними інтересами.....	36
3.3. Структура бази даних.....	38
Висновки.....	39
РОЗДІЛ 4. РОБОТА КОРИСТУВАЧА З РОЗРОБЛЕНОЮ СИСТЕМОЮ...	40
4.1. Ініціалізація вебзастосунка.....	40
4.2. Огляд функціональних можливостей системи.....	40
Висновки.....	49
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

Клієнт — це програма або пристрій, який звертається до іншого пристрою (сервера) для отримання певної інформації

Сервер — це комп'ютер або система, яка надає відповіді клієнтам або ініціює ще якісь дії за запитом клієнта.

СУБД — система управління базами даних

MERN — аббревіатура 4 основних технологій (MongoDB, Express.js, React, Node.js), що входять до складу стека

JSON — текстовий формат обміну даними між комп'ютерами

UI — користувацький інтерфейс

ВСТУП

У сучасному глобалізованому світі для більшості людей вивчення англійської мови є першорядним пріоритетом. Це основна мова міжнародного бізнесу, науки та технологій. Отже, володіння англійською забезпечує її носіям сильну конкурентну перевагу та можливості для співпраці та просування.

Ефективне спілкування є основним елементом вивчення мови, а комунікативна практика відіграє ключову роль у досягненні просунутого рівня. Участь у розмовах у реальному житті, у дискусіях і практика з носіями мови є важливими компонентами опанування мови.

Розвиток вебтехнологій дає змогу будь-кому використати їх на свою користь під час покращення розмовних навичок іноземною мовою. Соціальні мережі, комунікаційні майданчики та сервіси з онлайн-навчання створюють простір для спілкування людей з іншими, даючи їм змогу покращити рівень володіння мовою.

Створення вебзастосунка, що допоможе зацікавленим особам покращити свої комунікативні навички шляхом ефективної взаємодії один з одним, є основним завданням. Фокус лежить на спрощенні процесу пошуку та встановленні відповідностей між користувачами шляхом автоматизації та аналізу персональних даних користувача.

Актуальність теми кваліфікаційної роботи “Web-застосунок створення комунікаційних майданчиків з вивчення англійської мови” полягає в тому, що вивчення англійської мови онлайн та пошук спільнот щороку набирає шалених обертів. Розробка зручного та доступного майданчика може допомогти задовольнити популярний запит користувачів.

Мета кваліфікаційної роботи — розробити комунікаційний майданчик у формі вебзастосунка з вивчення англійської мови.

Відповідно до поставленої мети роботи визначено **основні завдання дослідження**:

- визначення основних вимог та аналіз наявних систем;
- аналіз та вибір технологій для реалізації проєкту;
- практичне застосування обраних технологій, проєктування архітектури та бази даних;
- розробка зручного UI та реалізація функціональності вебзастосунка.

Об'єктом дослідження кваліфікаційної роботи є вебзастосунок для пошуку співрозмовників з вивчення англійської мови на базі технологічного стека MERN.

Предмет дослідження — вебзастосунок, що персоналізовано поєднує співрозмовників з вивчення англійської мови із можливістю взаємодії у вигляді повідомлень.

Для досягнення поставленої мети використано наступні засоби та методи: мова програмування TypeScript, бібліотека React та колекція готових компонентів Shadcn/ui, платформа Node.js, фреймворк Express.js та база даних MongoDB.

Наукова новизна полягає у вдосконаленому підході до вивчення англійської мови.

Практичне значення отриманих результатів рекомендується використати для детального ознайомлення із функціональними можливостями системи з метою їх подальшого вдосконалення, що може стати цінною ініціативою, щоб зробити мовну освіту більш доступною та ефективною в нашому взаємопов'язаному світі.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Поняття вебзастосунка

Вебзастосунок, також відомий як вебдодаток, являє собою програмне забезпечення або програму, в якому клієнтом виступає браузер, а сервером — вебсервер [1]. Користувацький інтерфейс вебзастосунка розробляється з використанням бібліотек або фреймворків JavaScript, таких, як React, Vue та Angular. Для написання серверної частини може використовуватися широкий спектр мов програмування, таких, як JavaScript, Python, PHP, Java тощо.

Поняття вебзастосунка та вебсайту часто плутають, проте вони мають істотні відмінності у функціональності та цілях. Варто зазначити, що вебзастосунок і вебсайт виконують різні завдання та володіють різними характеристиками розробки. Вебзастосунок зазвичай спрямований на активну взаємодію з користувачем та надає можливість проведення різноманітних маніпуляцій з даними. Користувачі можуть здійснювати різні дії та взаємодіяти з різними елементами, що робить його більш гнучким та інтерактивним інструментом.

Вебсайт зазвичай фокусується на представленні інформаційного контенту без можливості активної взаємодії з користувачем.

Кафедра ІІЗ				НАУ 24 05 18 000 ІІЗ			
<i>Розроб.</i>	Волошин В.В.			АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Керівник</i>	Талаласв В.О.					10	10
					ПІ-501Бз		
<i>Н.-контр.</i>	Варнавський В.В.						

Відвідувачі сайту можуть переглядати текстову інформацію, але не мають можливості проводити маніпуляції з даними чи змінювати контент. Іншою важливою відмінністю є підхід до автентифікації: вебзастосунки зазвичай потребують автентифікації користувача для забезпечення безпеки та обмеження доступу до функціонала, тоді як для більшості вебсайтів автентифікація не обов'язкова і може використовуватися лише для обмеженого доступу до контенту.

У таблиці 1.1 подано більш детальне порівняння між вебзастосунками та вебсайтами, розкриваючи їхні основні особливості та відмінності.

Таблиця 1.1

Порівняльна характеристика вебзастосунка та вебсайту

Параметр	Вебзастосунок	Вебсайт
Основне призначення	Активна взаємодія з користувачем, проведення різних маніпуляцій з даними	Служить для публікації інформаційного контенту та не передбачає активної взаємодії з користувачем
Автентифікація	Вимагає автентифікації користувача для забезпечення безпеки та обмеження доступу до певних ресурсів	Не завжди обов'язкова, але може використовуватися лише для обмеженого доступу, наприклад, за допомогою паролю, який видає адміністратор ресурсу
Завдання та складність	Зазвичай мають широкий функціонал та розв'язують різні завдання, що може вимагати великих зусиль для розробки та підтримки	Зазвичай простіші у розробці й мають обмежений функціонал, оскільки їх основна мета полягає в представленні інформації користувачам
Внесення змін у проєкт	Зазвичай потрібно ретельно працювати з кодом, проводити ревізію та тестування всіх складових, що може вимагати значних зусиль та часу	Достатньо внести декілька змін у HTML-код сторінки для оновлення чи зміни контенту без значних труднощів

Щодо роботи вебзастосунків, то її можна описати наступним чином: користувач відправляє запит, який подається до вебсервера. Після отримання запиту, вебсервер опрацьовує його, в разі необхідності звертається до бази даних та формує відповідь клієнту. Важливо зазначити, що чим складніші запити, тим більше часу потрібно для їх опрацювання та отримання відповіді від сервера.

Різновиди вебзастосунків:

- SPA (Single Page Application) — односторінковий інтерактивний додаток. Весь контент додатка динамічно змінюється без перезавантаження сторінки [2].
- MPA (Multi Page Application) — це додатки, які працюють за традиційною схемою. При переході на нову сторінку, браузер робить новий запит на отримання всіх ресурсів [3].

1.2. Задача розробки вебзастосунка для створення комунікаційного майданчика з вивчення англійської мови

Розмовна практика є важливим етапом вивчення іноземної мови. До того ж жива комунікація робить процес не тільки корисним та цікавим, а й дозволяє зануритися у культуру країни, дізнатись більше про традиції та побут іноземців, що є додатковою мотивацією, можливістю розширити свій кругозір та завести нові знайомства.

В сучасному світі низький рівень вивчення англійської мови стає проблемою через обмежену можливість практики та відсутність ефективної комунікації. Багато людей, навіть маючи базові знання, натрапляють на труднощі у використанні мови в реальних ситуаціях. Метою є створення нової системи у вигляді комунікаційного майданчика, спрямованого на покращення навичок англійської мови, що дозволить користувачам отримати додаткову

практику та забезпечити ефективне вивчення мови у контексті реального спілкування.

Створення вебзастосунка з автоматизованим пошуком відповідних користувачів може значно полегшити цей процес. Взаємодія з представниками інших культур в онлайн-середовищі дозволить користувачам не лише вдосконалювати свої мовні навички, але й розширювати свої знання про світ та поглиблювати розуміння інших культур. Такий застосунок стане не лише ефективним інструментом для вивчення мови, але й засобом для взаєморозуміння та обміну досвідом, ідеями та поглядами між людьми з різних частин світу.

1.2.1. Знаходження користувачів за певними критеріями

Створення функціонала, що надає користувачеві можливість знаходити інших користувачів за певними критеріями, є ключовою задачею в розробці платформи. Метою є створення зручної та ефективної системи, яка дозволяє користувачам знаходити та встановлювати зв'язок з іншими користувачами відповідно до їх потреб.

Цей функціонал розв'язує проблему соціальної взаємодії та спілкування серед користувачів, дозволяючи їм знаходити осіб із потрібним рівнем англійської задля комунікації та практики. Він базується на концепції пошуку та фільтрації, щоб створити зручний та ефективний інструмент для формування спільнот та взаємодії між користувачами.

Завдання включає розробку алгоритмів пошуку та фільтрації, які враховують різні критерії, такі як стать, рівень англійської, місце проживання співбесідника та рамки віку. Користувач повинен мати можливість вибирати та встановлювати параметри пошуку, а система повинна ефективно аналізувати профілі користувачів та повертати релевантні результати.

1.2.2. Перегляд профілів користувачів системи та надсилання запитів на спілкування

Забезпечення можливості користувачам переглядати профілі інших користувачів системи та надсилати запити на спілкування є важливою функцією для взаємодії та розвитку соціальної сторони платформи. Мета полягає у створенні інструменту, який дає можливість користувачам знайомитися та спілкуватися, розширюючи їхні можливості взаємодії та мережі контактів.

Ця функція передбачає створення інтерфейсу, що дозволяє користувачам переглядати інформацію про інших учасників спільноти, що включає основні дані, такі як ім'я, фотографію, вік, відомості про рівень англійської, місце проживання, рідну мову та опис користувача. Крім того, користувач повинен мати можливість вислати запит на спілкування із конкретним користувачем.

Задача охоплює розробку безпечного та зручного механізму надсилання запитів на спілкування.

1.3. Аналіз наявних систем

Вивчення англійської мови включає не лише освоєння лексичних та граматичних правил, але й активний процес спілкування. З огляду на зростання попиту на інтерактивні та інноваційні методи навчання, виникає необхідність у визначенні ефективних комунікаційних майданчиків для вивчення англійської мови.

Метою даного дослідження є аналіз наявних систем, призначених для вивчення англійської мови через комунікаційні майданчики. Зокрема, об'єктом дослідження є різноманітні вебсервіси, мобільні додатки та інші інтерактивні засоби, які надають можливість спілкування та взаємодії для покращення навичок в англійській мові.

1.3.1. Платформа з вивчення англійської Hellotalk

Hellotalk — це навчальна спільнота, що складається з 40 мільйонів учасників з понад 150 країн світу. По-перше, він зберігає свою актуальність завдяки різноманітній можливості вивчення понад 100 мов, надаючи унікальний досвід для всіх мовознавців [4]. Однією з ключових переваг є можливість спілкування з носіями мови по всьому світу, що робить процес вивчення більш ефективним.

Пошукові функції Hellotalk дозволяють легко та швидко знаходити носіїв англійської чи будь-якої іншої мови за різними критеріями, включаючи країну чи місто. Функціонал Hellotalk допомагає здійснювати практику та вивчення мови вдома чи в дорозі (рис. 1.1). Інтеграція звукових файлів дозволяє перетворювати мовлення на текст, що полегшує процес навчання та взаємодії з користувачами додатка.



Рис. 1.1. Застосунок Hellotalk

Однією з визначальних рис є можливість безплатних дзвінків, забезпечуючи якість звуку, схожу на особисте спілкування. Це не тільки сприяє навчанню, але й дозволяє налагоджувати особисті зв'язки з партнерами з вивчення мови. HelloTalk також створює можливість для корекції граматики та вимови, завдяки ефективним інструментам. Це дозволяє вивчати мову так, як вона використовується в реальному житті, з орієнтацією на сучасний спосіб мовлення.

На основі відгуків користувачів, HelloTalk виявляється ефективним інструментом для навчання мов. Користувачі відзначають швидкий прогрес у вивченні нових мов та збагачення своєї культурної свідомості через взаємодію з іншими користувачами з різних країн.

1.3.2. Вивчення англійської мови за допомогою застосунку Coffee Strap

У сучасному світі, де комунікація стає ключовою навичкою, інноваційні підходи до вивчення мов стають все більш важливими. Один із таких унікальних ресурсів, який виокремлюється серед інших, – це Coffee Strap, що пропонує неординарний метод вивчення англійської мови через випадкове спілкування [5].

Coffee Strap пропонує своїм користувачам унікальний «чат-рулетка» підхід до вивчення мов. Користувачі можуть обрати одну з 12 мов для вивчення, а потім система випадковим чином з'єднає їх з іншими користувачами. Однак недоліком застосунку є те, що під час випадкового з'єднання не враховуються хобі та інтереси, щоб спростити процес знаходження спільної мови між користувачами.

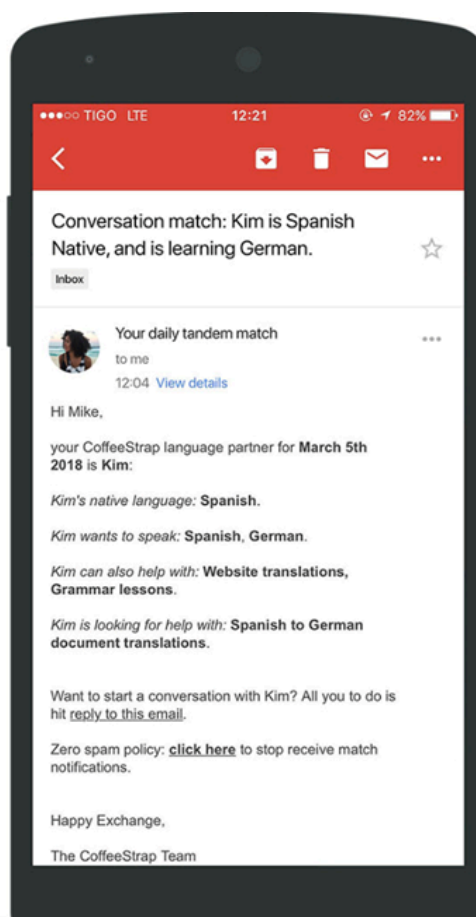


Рис. 1.2. Інтерфейс застосунку Coffee Strap

Один із ключових аспектів Coffee Strap — обмеження часу розмови до 7 хвилин. Це дозволяє користувачам ефективно використовувати час, не відволікаючись на довгі та безплідні розмови. Після закінчення цього часу система автоматично переходить до пошуку нового співрозмовника, що створює динамічне та ефективне середовище для вивчення мови.

Основна перевага такого підходу полягає в тому, що він допомагає подолати страх перед спілкуванням англійською мовою через невпевненість у своїх знаннях. Хоча цей підхід може не підходити всім, його експериментальний характер і можливість випробування нових підходів до навчання англійської мови роблять його важливим гравцем в області вивчення мов.

1.3.3. Платформа для вивчення мов Mylanguageexchange

Mylanguageexchange, з понад 3 мільйонами користувачів зі 175 країн, є глобальним майданчиком для вивчення мов, де кожен може знайти співрозмовників для поліпшення своєї англійської та інших мов світу [6].

Однією з ключових переваг Mylanguageexchange є можливість участі у розмовних онлайн-клубах. Групуючись у команди до 4 осіб, користувачі можуть проводити половину заняття англійською, а іншу половину — мовою, якою вони володіють. Це не тільки забезпечує ефективне вивчення, але й створює дружню атмосферу для комунікації та взаємодопомоги.

Крім того, ресурс пропонує онлайн-бібліотеки, словникові ігри та можливість листування. Отже, користувачі можуть обмінюватися різноманітними мовними ресурсами, щоб розширити свій словниковий запас та покращити навички англійської мови.

Кожен профіль на сайті має ретельний опис із вказівкою інтересів, віку та цілей, що допомагає знаходити ідеальних партнерів для мовної практики.

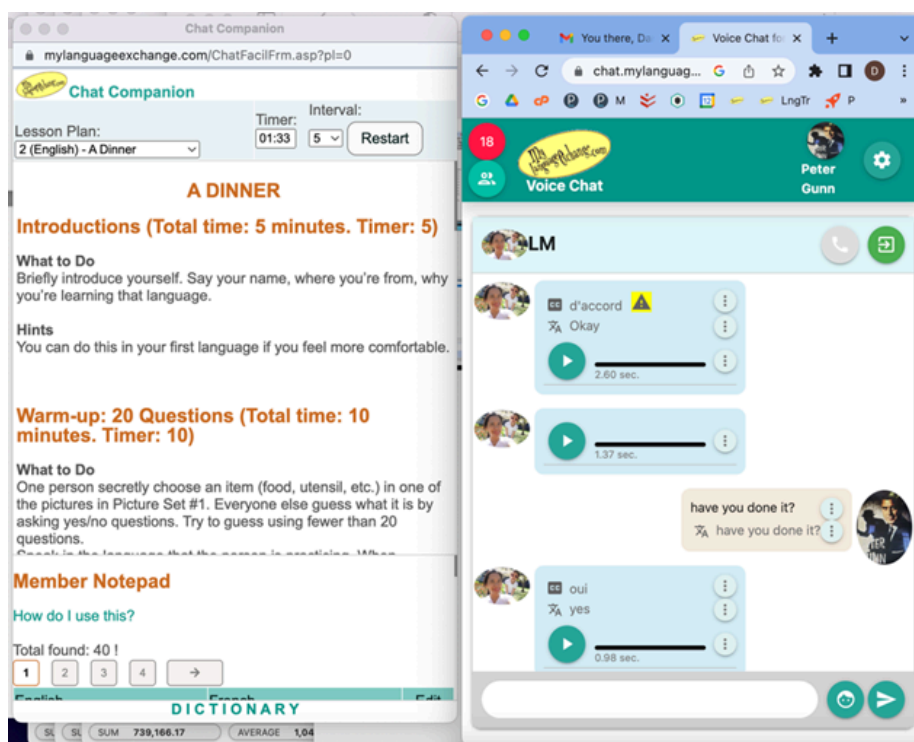


Рис. 1.3. Інтерфейс застосунку Mylanguageexchange

Попри цільовий орієнтований функціонал та глобальне охоплення, Mylanguageexchange також має свої нюанси. Деякі користувачі можуть відзначити застарілість інтерфейсу та складність навігації у платформі.

Висновки

Вебзастосунок — це програмне забезпечення, в якому клієнтом виступає браузер, а сервером — вебсервер. Вебзастосунки ззовні нічим не відрізняються від вебсайтів для більшої кількості користувачів. Однак вебзастосунки здатні виконувати широкий спектр завдань, таких, як активна взаємодія з користувачем, його автентифікація та авторизація, проведення різноманітних маніпуляцій з даними, в той час, як вебсайти носять більш інформаційний характер.

Створення вебзастосунка комунікаційного майданчика для вивчення англійської мови є актуальною задачею, оскільки він дозволяє поліпшити навички мовлення та розуміння через взаємодію з носіями мови. Такі системи надають користувачам можливість знаходити співрозмовників за певними критеріями та встановлювати зв'язок з ними для практики англійської мови у реальних ситуаціях спілкування.

Аналіз наявних систем показує, що існують різні платформи та додатки, які закривають цю можливість, але варто зазначити, що кожне рішення має свої переваги та недоліки, надаючи при цьому свої опції та функціональні можливості.

Таким чином, створення платформи, яка зможе допомогти знайти співрозмовника за вказаними критеріями та можливістю подальшої взаємодії може значно полегшити процес вивчення мови, тим самим задовольнити шалений попит зацікавлених осіб.

РОЗДІЛ 2

ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАСТОСУНКУ

Одне з найважливіших і найвідповідальніших завдань, яке необхідно виконати перед початком будь-якого проекту, — це планування. Незалежно від того, який бізнес-продукт розробляється, до планування потрібно підходити дуже ретельно. Адже надійний та якісний план дає змогу мінімізувати ризики та збільшити шанси на успіх проекту.

В індустрії інформаційних технологій одним з етапів планування є вибір технологічного стека. Технологічний стек — це сукупність інструментів та засобів програмної реалізації, що використовуються для створення програмного застосунку.

Для розробки Web-застосунку комунікаційних майданчиків з вивчення англійської мови було обрано мову програмування TypeScript як для клієнтської, так і серверної частини, технологічний стек MERN, середовище розробки Visual Studio Code та розподілену систему керування версіями файлів Git.

Кафедра ІІЗ				НАУ 24 05 18 000 ІІЗ			
<i>Розроб.</i>	Волошин В.В.			ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАСТОСУНКУ	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Керівник</i>	Талалаєв В.О.					20	14
					ПІ-501Бз		
<i>Н.-контр.</i>	Варнавський В.В.						

2.1. Технологічний стек MERN

MERN — це аббревіатура 4 основних технологій, що входять до складу стека [7]:

- MongoDB — документо-орієнтована система керування базами даних.
- Express.js — швидкий на гнучкий фреймворк для вебзастосунків, побудованих на Node.js.
- React — JavaScript бібліотека для створення інтерфейсів користувача.
- Node.js — платформа для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript.

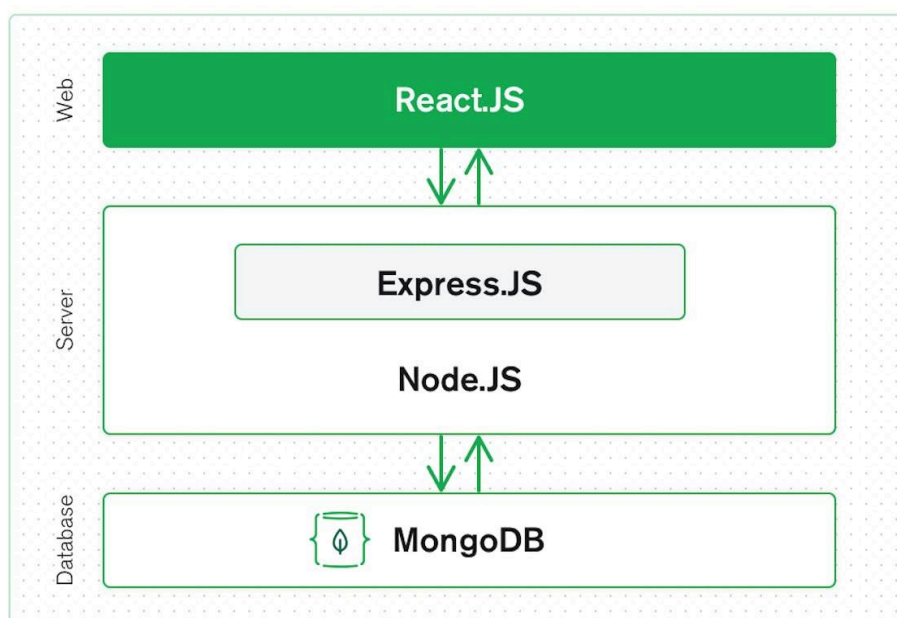


Рис. 2.1. Тривірнева архітектура застосунку

Технологічний стек MERN дозволяє легко будувати тривірневу архітектуру [8], що містить:

- Рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві та введення від нього керуючих команд. За вимогами безпеки та масштабованості не повинен мати прямого зв'язку з базою даних та бути навантаженим основною бізнес-логікою, адже це не його зона відповідальності.

- Прикладний рівень, який реалізує основну бізнес-логіку застосунку і на якому здійснюється необхідна обробка інформації.
- Рівень управління даними, який забезпечує зберігання даних та доступ до них.

Таким чином, MERN стек відомий своєю повнотою та єдністю. Дозволяє швидко та якісно будувати full-stack застосунки, що відповідають традиційній трирівневій архітектурі, гарантуючи чудовий зв'язок між клієнтом, сервером та базою даних, використовуючи React для створення клієнтських інтерфейсів, Node.js та Express.js для написання серверної частини застосунку та MongoDB для гнучкого збереження даних.

2.1.1. База даних MongoDB

MongoDB представляє сучасну, документо-орієнтовану систему керування базами даних, яка надає гнучкі та ефективні можливості для зберігання та обробки даних [9]. На відміну від інших СУБД, MongoDB є нереляційною, тобто зберігає дані без чітких зв'язків між собою. Основною концепцією MongoDB є використання колекцій та документів, де кожна колекція являє собою сукупність документів.

messages				
Storage size: 20.48 kB	Documents: 1	Avg. document size: 139.00 B	Indexes: 1	Total index size: 20.48 kB
tokens				
Storage size: 20.48 kB	Documents: 13	Avg. document size: 242.00 B	Indexes: 1	Total index size: 36.86 kB
users				
Storage size: 20.48 kB	Documents: 13	Avg. document size: 534.00 B	Indexes: 2	Total index size: 73.73 kB

Рис. 2.2. Список колекцій MongoDB

```

_id: ObjectId('65a1be979d3573c768fa9a05')
email: "voloshin.v.02@gmail.com"
password: "$2b$12$yASa4LLuM7Fl2o1PWnqE0.MciwP1a9MsjQ6zERCkoIo8gw9P48Eg2"
isActive: true
username: "Vitalii Voloshyn"
profilePicture: "http://localhost:3000/img/profilePicture-1705704282150-938362197.png"
▼ interests: Array (3)
  0: "Math"
  1: "Programming"
  2: "Science"
createdAt: 2024-01-12T22:35:03.272+00:00
updatedAt: 2024-01-19T22:54:17.310+00:00
__v: 0
birthdate: 2002-04-06T22:00:00.000+00:00
country: "UA"
description: "Hello guys! If you are into football, drop me a line :)"
gender: "Male"
languageLevel: 4
nativeLanguage: "uk"
age: 21

```

Рис. 2.3. Документ колекції User

MongoDB зберігає записи даних у вигляді документів BSON [10]. Цей формат використовує бінарне кодування, що дозволяє ефективно обробляти дані та забезпечує значно швидший доступ до інформації порівняно з JSON.

Таблиця 2.1

Порівняльна характеристика JSON та BSON

	JSON	BSON
Кодування	UTF-8 String	Binary
Типи даних	String, Boolean, Number, Array, Object, null	String, Boolean, Number (Integer, Float, Long, Decimal128...), Array, null, Date, BinData
Можливість зчитування	Люди, машини	Машини

Окрім того, як було зазначено на початку розділу, MongoDB є нереляційною базою даних та має відмінності в термінології, порівнюючи з реляційними базами даних [11].

Порівняння термінів SQL з MongoDB

SQL	MongoDB
База даних	База даних
Таблиця	Колекція
Рядок	Документ (BSON)
Колонка	Поле

2.1.2. Фреймворк Express.js

Express.js — швидкий, гнучкий та мінімалістичний фреймворк для вебзастосунків, побудованих на Node.js, що надає широкий набір функціональності та був названий де-факто стандартом для створення серверної частини вебзастосунка [12].

У традиційних вебзастосунках, вебсервер очікує на HTTP-запити від клієнта [13], а отримавши їх, обере відповідний обробник маршруту та делегує йому подальші дії для цього запиту. Зазвичай, написання такого функціонала з нуля може бути досить часозатратним та непростим завданням. Express.js вирішує це завдання за користувача, маючи у своєму розпорядженні безліч допоміжних HTTP-методів та проміжних обробників, тим самим дає змогу створювати надійні API легко і швидко.

```
const router = express.Router();
router.use(verifyToken);

router.route('/').get(messageController.getMessages);
router.route('/send').post(messageController.sendMessage);
```

Рис. 2.4. Приклад маршрутів

Також, однією з переваг використання Express.js є можливість легкої інтеграції проміжних функцій, так званих middleware. Middleware — це функція, що має доступ до об'єкта запиту, відповіді та має змогу викликати наступну проміжну функцію обробки в циклі “запит-відповідь”. Наприклад, ми можемо використати проміжну функцію для перевірки чи є користувач активованим, якщо так — викликаємо наступну функцію, якщо ні — закінчуємо цикл “запит-відповідь”.

```
export const verifyStatus = (
  req: Request,
  res: Response,
  next: NextFunction
) => {
  if (!req.isActivated) {
    return next(new AppError('Your account is not activated', 403));
  }

  next();
};
```

Рис. 2.5. Використання проміжної функції

2.1.3. Бібліотека React

React є відкритою бібліотекою JavaScript, призначеною для розробки інтерфейсів користувача [14]. Він був створений командою Facebook та швидко завоював популярність серед розробників з усього світу.

Однією з ключових концепцій у React є компоненти — самостійні блоки коду, відповідальні за відтворення певної частини користувацького інтерфейсу. Незалежно від того, чи ви працюєте з сотнями інших розробників, чи наодинці, принцип побудови компонентів залишається незмінним, а його головна мета — дозволити вам легко інтегрувати частини, створені групами людей, командами та організаціями, спрощуючи процес створення складних інтерфейсів.

React компоненти — це звичайні функції JavaScript, які приймають параметри (props) та повертають JSX — синтаксичне розширення для JavaScript, що дозволяє писати HTML-подібну розмітку всередині JavaScript файлу.

```
const UserList = () => {
  const { ref, inView } = useInView();
  const { data, isPending, isError, fetchNextPage, isFetchingNextPage, hasNextPage } = useUsers();

  useEffect(() => {
    if (inView && hasNextPage) {
      fetchNextPage();
    }
  }, [inView, hasNextPage, fetchNextPage]);

  if (isPending) {
    return <Loader />;
  }

  if (isError) {
    return <InfoMessage title="Something went wrong" />;
  }

  if (!data.pages[0].length) {
    return <InfoMessage title="Nobody was found 🙄" description="Try changing the filters" />;
  }

  return (
    <div>
      <ul className="grid gap-5 sm:grid-cols-2 lg:grid-cols-3">
        {data.pages.map((users) => users.map((user) => <User key={user._id} user={user} />))}
      </ul>
      <span className="absolute opacity-0" ref={ref}></span>

      {isFetchingNextPage && (
        <div className="mt-5 flex justify-center">
          <Loader />
        </div>
      )}
    </div>
  );
};
```

Рис. 2.6. React компонент

Підсумовуючи, можна зробити висновок, що компоненти — це будівельні блоки UI, з яких складається наш застосунок.

2.1.4. Платформа Node.js

На відміну від інших мов програмування, JavaScript міг працювати лише з браузерами, які мали вбудований движок виконання коду. З 2009 року все змінилося. Тоді було створено програмне середовище Node.js, яке дозволило розробникам використовувати мову JavaScript для створення серверної частини

програми. Це фактично уніфікувало використання однієї мови програмування для розробки вебзастосунків.

Підсумовуючи, Node.js — це однопоточне кросплатформне середовище з відкритим вихідним кодом для виконання мережевих застосунків, написаних мовою JavaScript [15].

2.2. Мова програмування JavaScript

До появи мови JavaScript, сторінки браузера були надзвичайно простими та статичними. Після завантаження сторінки все, що можна було зробити — прогортати вгору та вниз, або перейти за посиланням. Хоча спочатку JavaScript був створений для того, щоб додати інтерактивність і «оживити вебсторінку», з часом, через надзвичайно стрімкий розвиток мови, він давно вийшов за рамки того, для чого був створений.

На сьогодні, JavaScript може виконуватися не тільки в браузері, але й на сервері або на будь-якому пристрої, який має спеціальну програму — рушій JavaScript [16]. Тож, важко уявити створення сучасних вебзастосунків без мови програмування JavaScript.

Вбудований в браузер JavaScript має такі можливості, як:

- Додавати, видаляти або модифікувати HTML-код на сторінці, а також працювати зі стилями.
- Реагувати на користувацькі дії, такі як натискання миші, клавіші клавіатури, переміщення вказівника та безліч інших ключових подій.
- Надсилати мережеві запити до віддалених серверів.
- Отримувати та надсилати Cookie — невеликі рядки з даними, які зберігаються безпосередньо в браузері. Вони є частиною HTTP-протоколу.

- Запам'ятовувати дані на стороні клієнта, зберігаючи їх в local або session storage.

Саме цей перелік функціональних можливостей робить мову програмування JavaScript унікальною. Тому не дивно, що JavaScript — найбільш поширена мова програмування у веброзробці.

2.3. Мова програмування TypeScript

Не зважаючи на всі переваги JavaScript, існують певні недоліки, які можуть виникати при розробці великих та складних проєктів. Наприклад, мова програмування JavaScript має динамічну типізацію. Це означає, що змінні в межах програми можуть вільно змінювати свій тип даних. Тобто, числове значення може перетворитись на рядкове, або навпаки, що може призвести до неочікуваних результатів та помилок. Крім того, виявлення помилок відбувається під час виконання коду, що ускладнює процес розробки.

TypeScript була представлена компанією Microsoft восени 2012 року та позиціонується як засіб розробки вебзастосунків, що розширює можливості JavaScript. Це означає, що розробники й далі можуть використовувати весь наявний JavaScript-код без змін. Головною особливістю є те, що TypeScript додає статичну типізацію, можливість описувати типи даних та спрощує процес роботи у команді, що допомагає створювати надійніші додатки та виявляти помилки на етапі компіляції [17].

```
type Gender = 'Male' | 'Female';

You, 4 weeks ago | 1 author (You)
export interface IUser {
  _id: string;
  isActivated: boolean;
  username: string;
  profilePicture: string;
  interests: string[];
  languageLevel?: number;
  nativeLanguage?: TLanguageCode;
  country?: TCountryCode;
  gender?: Gender;
  birthDate?: string;
  description?: string;
  age?: number;
}
```

Рис. 2.7. Інтерфейс об'єкта користувача

2.4. Колекція готових компонентів Shadcn/ui

Однією з областей, на яку розробники витрачають багато часу, є стилізація вебзастосунків. Окрім стилізації, варто приділити значну увагу на такі речі, як користувацький досвід та доступність контенту, щоб розробити інтерфейс максимально простим і зрозумілим.

Необхідно враховувати, що сучасний інтернет став одним з основних ресурсів для пошуку інформації та освіти, а також для розваг. Важливе вплив на сприйняття інформації виникає не лише від її змісту, але й від дизайну. Вебсторінки повинні бути доступними для всіх користувачів. Наприклад, неможливість збільшення тексту ускладнює взаємодію для людей із зоровими проблемами. Зображення без описів ускладнюють доступ до контенту для тих, хто користується Screen reader (комп'ютерна програма, яка розпізнає й

інтерпретує текст та перетворює його на "синтезоване мовлення") для читання контенту.

Для виявлення та подолання цих труднощів необхідно звертати увагу на принципи доступності вебзастосунка. Вебресурс, який дотримується принципів доступності, виправляє бар'єри, що перешкоджають використанню його людьми із різними обмеженнями. Таким чином, забезпечуються рівні можливості для всіх користувачів.

Shadcn/ui — це колекція готових компонентів, які ми можемо скопіювати, модифікувати та додати до нашого проєкту, надаючи нам всі права власності й повний контроль над ними [18]. Тим самим спрощує процес розробки, розв'язує всі вищеперелічені проблеми та надає неймовірну гнучкість під час розробки.

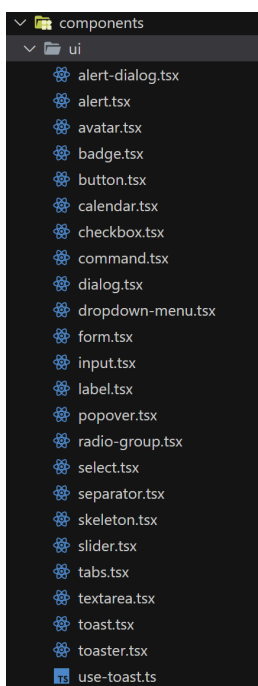


Рис 2.8. Набір компонентів, що використовувався під час розробки

2.5. Середовище розробки Visual Studio Code

Visual Studio Code — це простий, але потужний редактор вихідного коду. Він є мультиплатформеним, тобто доступний на таких операційних системах як

Windows, macOS та Linux. Підтримує JavaScript, TypeScript і Node.js, а також має велику екосистему розширень для інших мов (наприклад, C++, C#, Java, Python, PHP і Go) [19].

Окрім цього Visual Studio Code надає додаткові можливості завдяки розширенням, які можуть вносити додатковий корисний функціонал до наявного набору.

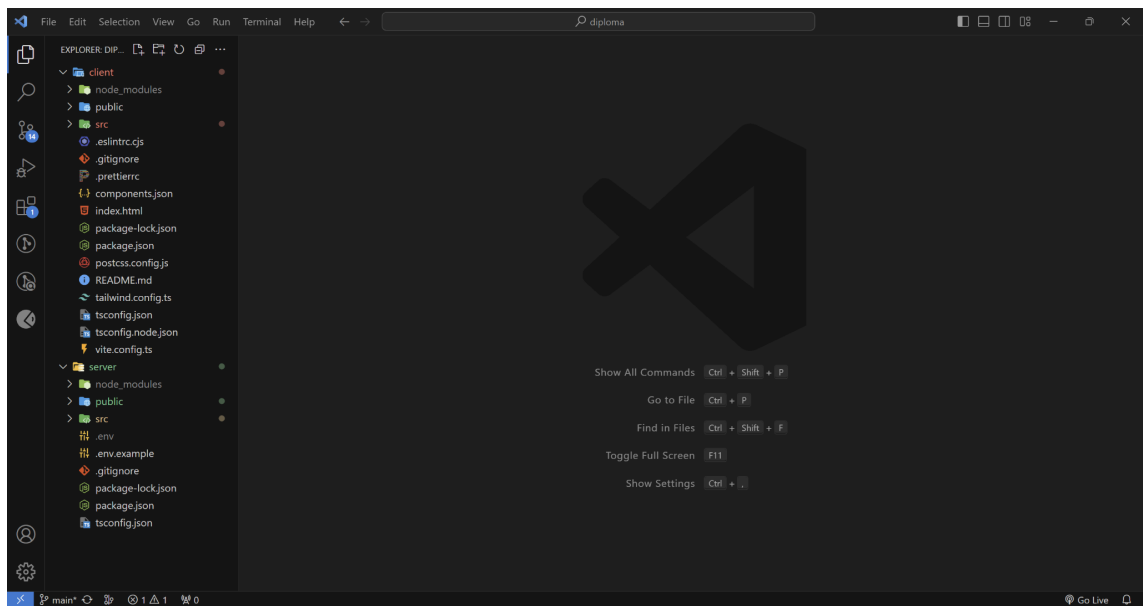


Рис. 2.9. Інтерфейс Visual Studio Code

2.6. Система керування версіями Git та вебсервіс Github

Під час розробки програмного застосунку, керування кодом є важливою складовою. На перший погляд, це може здатися простим завданням, але зі зростанням масштабів проекту, управління змінами може стати значним викликом. На щастя, сьогодні існують технології, які здатні вирішити це завдання. Однією з таких є система керування Git.

Git — це розподілена система керування версіями файлів проекту, що служить для відстеження змін в коді, фіксуючи "знімки" різних станів розробки [20]. Цей інструмент сприяє співпраці розробників, уникненню конфліктів та зберіганню повної історії змін.

```
volos@Vitalii MINGW64 ~/Desktop/pet-projects/diploma (main)
$ git log
commit 412ae6c48322688eebdd77dcf31b0b785f2d66e5 (HEAD -> main, origin/main)
Author: Vitalii <voloshin.v.02@gmail.com>
Date: Sun Jan 14 15:12:56 2024 +0100

    feat: messaging

commit 2ec0bc34c670b5f05df9422d69a78d2c9a2a9dd7
Author: Vitalii <voloshin.v.02@gmail.com>
Date: Sat Jan 13 00:20:07 2024 +0100

    change profile picture

commit 24b61f5453c7bcff2c6944bc760bd2d4d57a14a8
Author: Vitalii <voloshin.v.02@gmail.com>
Date: Fri Jan 12 14:34:32 2024 +0100

    search by keyword
```

Рис. 2.10. Історія комітів

Git став невіднятною складовою сучасної розробки та має безліч переваг, таких як:

- Контроль змін. Ця функція дає змогу ефективно спостерігати за змінами в коді та проводити виправлення без стресу. Це дозволяє розробникам легко відстежувати, як розвивається код.
- Співпраця з іншими розробниками. Це дозволяє ефективно працювати в команді, дозволяючи розробникам одночасно працювати над одним проєктом, не перешкоджаючи прогресу один одного.
- Безпека. Git гарантує безпеку коду, дозволяючи відновлювати видалені файли та повертати зміни за допомогою історії комітів. Це допомагає уникнути втрати важливих даних та легко їх відновити у разі помилки.

GitHub — один з найбільших вебсервісів для спільної розробки програмного забезпечення [21]. На платформі GitHub можна зберігати код, оцінювати роботи інших розробників, створювати приватні та публічні репозиторії, взаємодіяти з репозиторіями, керувати правами доступу та відстежувати помилки.

Name	Last commit message	Last commit date
..		
public/img	feat: messaging	3 days ago
src	feat: messaging	3 days ago
.env.example	init	last month
.gitignore	init	last month
package-lock.json	refactor, user filtering	5 days ago
package.json	refactor, user filtering	5 days ago
tsconfig.json	auth	3 weeks ago

Рис. 2.11. Інтерфейс GitHub репозиторію

Висновки

Підсумовуючи розділ 2, вибір технологій майбутнього проєкту грає ключову роль у його успіху та якості.

Використовуючи технологічний стек MERN можна створювати повноцінні full-stack вебзастосунки з тривірневою архітектурою. Можна зробити висновок, що MongoDB надає гнучкість та масштабованість для зберігання та управління даними, Express.js дозволяє створювати серверну частину застосунку, React забезпечує динамічний та зручний інтерфейс користувача, а Node.js дозволяє об'єднати ці компоненти в єдину функціональну систему.

Додавання TypeScript до цього стеку відіграло також важливу роль. Розширюючи функціональні можливості JavaScript, TypeScript надає такі функції, як статичний тип, інтерфейси, класи та модулі, що призводить до покращення якості коду та зменшенню помилок у процесі розробки.

Оскільки велика увага приділяється можливості активної взаємодії користувача з вебзастосунком, сформований стек відповідає сучасним стандартам розробки та надає змогу розв'язувати усі поставлені задачі та забезпечити коректність роботи вебзастосунка.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ

3.1. Процес автентифікації

Автентифікація є важливою складовою сучасного вебзастосунка. Захист конфіденційної інформації та запобігання несанкціонованому доступу є важливим завданням під час процесу розробки, тому основний фокус у розробці серверної частини проєкту був саме на автентифікацію та авторизацію користувачів до системи.

Одним зі стандартів автентифікації та авторизації користувачів в клієнт-серверних застосунках є JSON Web Token (JWT) [22]. Токен JWT складається з трьох частин: заголовка, корисного навантаження (наприклад, роль користувача, чи є він активованим та будь-яка інша інформація для перевірки прав доступу) та підпису (або даних шифрування). Сервер генерує токен, підписує його секретним ключем та передає клієнту, який надалі використовує цей токен для підтвердження своєї особи. Таким чином, як тільки користувач успішно увійшов у систему, використовуючи свої облікові дані, йому буде надано JSON Web Token, що дозволяє отримати доступ до певних ресурсів, які не обмежені цим токеном. Токен, який надається сервером, називається access token (токен доступу). Кожного разу, коли користувач хоче отримати доступ до певного ресурсу, він повинен

Кафедра ІІЗ				НАУ 24 05 18 000 ІІЗ			
<i>Розроб.</i>	Волошин В.В.			ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Керівник</i>	Талалаєв В.О.					34	6
					ПІ-501Бз		
<i>Н.-контр.</i>	Варнавський В.В.						

прикріпити цей токен до заголовків запиту, і сервер, своєю чергою, вирішує, чи має користувач право отримати відповідні дані. Варто пам'ятати, що підписаний токен захищений від підробки, але може бути прочитаним будь-ким, тому не варто додавати до корисного навантаження чутливу інформацію, таку як пароль користувача.

Будь-який користувач з токеном доступу автоматично проходить процес автентифікації, незалежно від того чи є він справжнім або зловмисним, тому за мірками безпеки, токен доступу роблять короткотривалим, як правило від 15 хвилин до 1 години, адже ймовірність того, що токен буде скомпрометований або викрадений, зростає, чим довше час його життєдіяльності.

```
export const verifyToken = async (
  req: Request,
  res: Response,
  next: NextFunction
) => {
  try {
    const { accessToken } = req.cookies;

    if (!accessToken) {
      return next(new AppError('You are not logged in. No token', 401));
    }

    const userData = jwt.verify(
      accessToken,
      process.env.ACCESS_TOKEN_SECRET!
    ) as AccessToken;

    const currentUser = await User.findById(userData.userId);

    if (!currentUser) {
      return next(
        new AppError(
          'The user belonging to this token does no longer exist',
          401
        )
      );
    }

    req.userId = currentUser.id;
    req.isActivated = currentUser.isActivated;
    req.user = currentUser;

    next();
  } catch (err) {
    next(err);
  }
};
```

Рис. 3.1. Перевірка access токену на валідність

Примушення користувача проходити процес автентифікації кожні 15 хвилин призведе до несприятливого користувацького досвіду. Тому на розв'язання цієї проблеми, окрім access token, варто також генерувати refresh token, який дозволяє отримувати нову пару токенів, не вимагаючи від користувача повторного проходження процесу автентифікації. Оскільки задачею проєкту є як створення безпечної системи, так і зручного інтерфейсу, access token і refresh token у парі є поширеним підходом та чудово виконує свою функцію, задовольняючи одночасно обидві задачі (див. рис. 3.2).

```
export const refresh = async (req: Request, res: Response, next: NextFunction) => {
  try {
    const { refreshToken } = req.cookies;

    if (!refreshToken) {
      return next(new AppError('No token. Access denied', 401));
    }

    const userData = jwt.verify(refreshToken, process.env.REFRESH_TOKEN_SECRET!) as RefreshToken;

    const tokenFromDb = await tokenService.findToken(refreshToken);

    if (!tokenFromDb) {
      return next(new AppError('This token does not belong to you. Access denied', 401));
    }

    const user = await User.findById(userData.userId);
    if (!user) {
      return next(new AppError('User not found. Access denied', 401));
    }

    createSendTokens(user, 200, res, next);
  } catch (err) {
    next(err);
  }
};
```

Рис. 3.2. Оновлення access та refresh токенів

3.2. Автоматизація пошуку користувачів за подібними інтересами

Розроблена система не лише відкриває можливості взаємодії шляхом обміну повідомлень зі співрозмовниками для покращення комунікативних навичок англійською мовою, але також автоматизує процес пошуку осіб на основі особистих даних поточного користувача, що сприяє не лише зручності, але й надає персоналізований підхід до вивчення нової мови.

Під час отримання нових навичок, людина зазвичай виходить за межу особистого комфорту, що заважає їй більш ефективно навчатися. У вебзастосунку використано персоналізований підхід для підбору подібних користувачів, аби створити сприятливі умови та допомогти користувачам подолати дискомфорт під час нав'язування контакту з іншими користувачами системи.

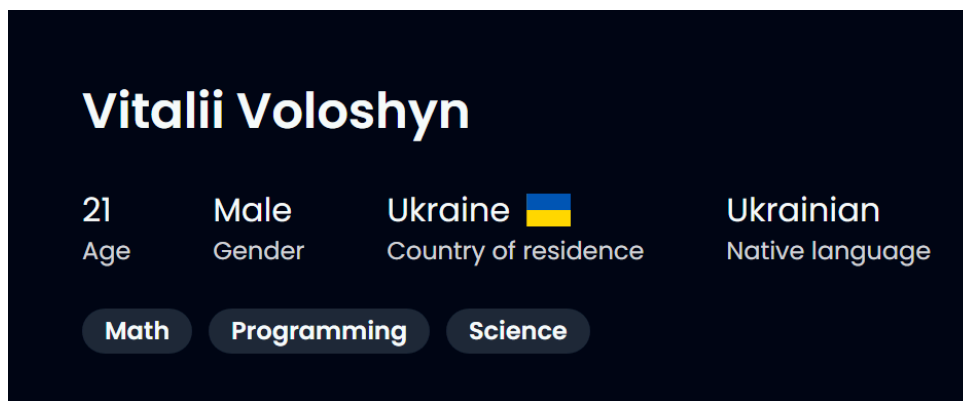


Рис. 3.3. Особисті дані поточного користувача

На рис. 3.3 можна побачити, що поточний користувач має такий спектр інтересів, як математика, програмування та наука. Проаналізувавши ці дані, система автоматично пропонує впорядкований список користувачів із найбільшою кількістю збігів за спільними інтересами. Першими у списку відібраних користувачів йдуть ті, у кого найбільше збігаються інтереси із поточним користувачем. Що нижче знаходиться користувач у списку, то менша ймовірність, що користувачі легко знайдуть спільну мову, виходячи з того, які інтереси вони самі вказали у своїх профілях. Таким чином, цей підхід максимізує ймовірність ефективної взаємодії, адже не завжди легко знайти спільну мову з незнайомою людиною.

Explore our community 🗣️

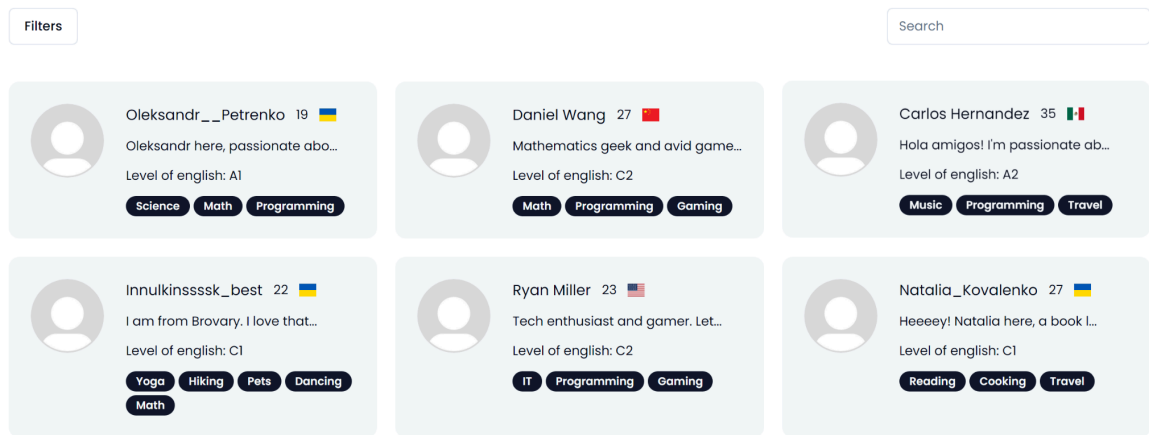


Рис. 3.4. Впорядкований список користувачів

3.3. Структура бази даних

Бази даних є важливим елементом у вебзастосунках для зберігання та організації великої кількості даних. Забезпечуючи швидкий доступ і оптимізовані запити, вони полегшують взаємодію користувачів з додатком, що дозволяє ефективно керувати даними.

Оскільки було обрано базу даних MongoDB, варто зазначити, що MongoDB є нереляційною базою даних. Унікальною характеристикою цієї бази даних є те, що схема даних є динамічною та може бути змінена в будь-який час. Доступ до даних ускладнюється, але такі СУБД вирізняються швидкістю і продуктивністю.

Для визначення структури бази даних, в якій зберігається інформація, розглянемо три основні моделі: User, Token та Message.

- User — описує користувача. Важливо зазначити, що пароль, за мірками безпеки, заборонено зберігати у вигляді простого тексту, адже це є прямою загрозою конфіденційності. Замість цього, було використано бібліотеку bcrypt з алгоритмом одностороннього хешування.

- Token — описує refresh token користувача. Описуючи refresh token в окремій колекції, ми гарантуємо, що тільки власник конкретного refresh token може оновити свої пари токенів, а також цей підхід дозволяє оптимізувати доступ до даних користувача.
- Message — описує повідомлення між користувачами, має такі поля, як sender — посилання на користувача, що відправив повідомлення, recipient — посилання на користувача, що отримав повідомлення та message — зміст повідомлення.

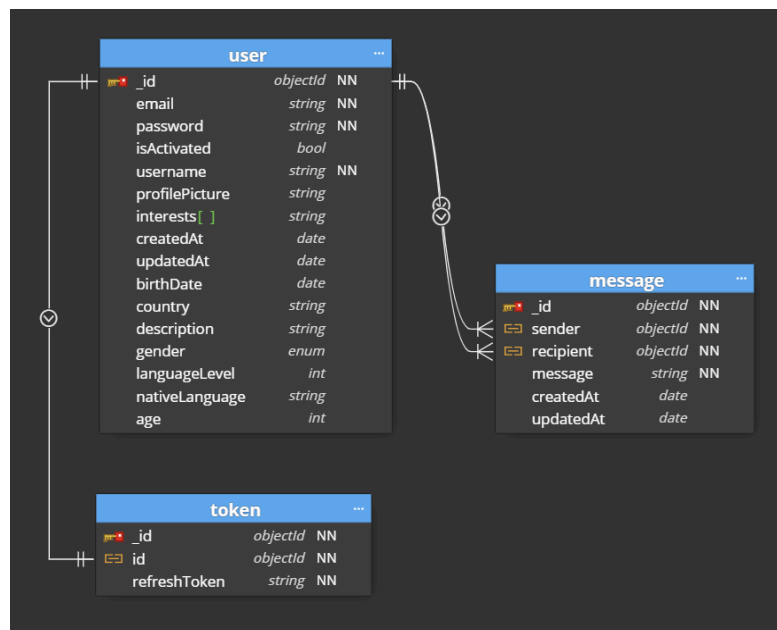


Рис. 3.5. Структура бази даних

Висновки

Підсумовуючи, програмна реалізація вебзастосунка охоплює: ефективний процес автентифікації за допомогою технології JWT, що забезпечує безпеку та конфіденційність користувачів. Автоматизація пошуку користувачів за подібними інтересами вдосконалює процес навчання та взаємодію між ними. Структура бази даних відображає високий рівень організації та ефективного управління інформацією, що сприяє стабільності та швидкодії системи.

РОЗДІЛ 4

РОБОТА КОРИСТУВАЧА З РОЗРОБЛЕНОЮ СИСТЕМОЮ

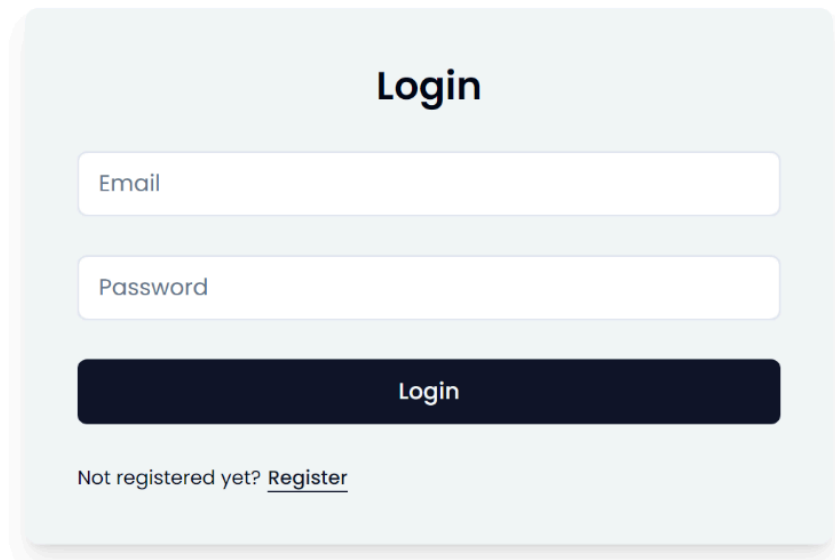
4.1. Ініціалізація вебзастосунка

Для ініціалізації застосунку, необхідно переконатись, що на поточній системі встановлено середовище Node.js та Git. Після цього, шляхом клонування віддаленого репозиторію з GitHub, можна отримати доступ до проєкту. Наступний крок — встановлення всіх необхідних залежностей, використовуючи npm (менеджер пакунків для мови програмування JavaScript). Як результат вищеперелічених дій, ми отримуємо можливість працювати із застосунком на локальній машині.

4.2. Огляд функціональних можливостей системи

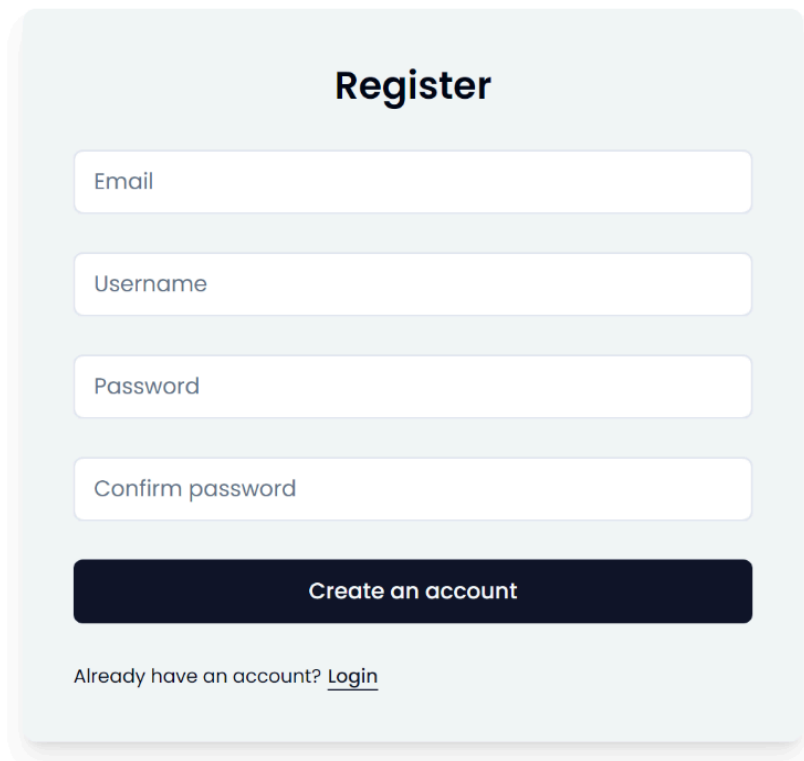
Як згадувалося у розділі 3 та підрозділі 3.1, першим та одним з найголовніших процесів застосунку є автентифікація користувача. Процес виглядає наступним чином: при запуску вебзастосунка користувачеві пропонується виконати першочергову дію — зареєструватися, у випадку відсутності облікового запису, або увійти в систему, якщо він вже здійснив реєстрацію раніше.

Кафедра ІПЗ				НАУ 24 05 18 000 ПЗ			
<i>Розроб.</i>	Волошин В.В.			РОБОТА КОРИСТУВАЧА З РОЗРОБЛЕНОЮ СИСТЕМОЮ	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Керівник</i>	Талалаєв В.О.					40	10
					ПІ-501Бз		
<i>Н.-контр.</i>	Варнавський В.В.						



The image shows a login form titled "Login". It features two input fields: "Email" and "Password". Below these fields is a dark blue button labeled "Login". At the bottom of the form, there is a link that says "Not registered yet? [Register](#)".

Рис. 4.1. Вхід у систему

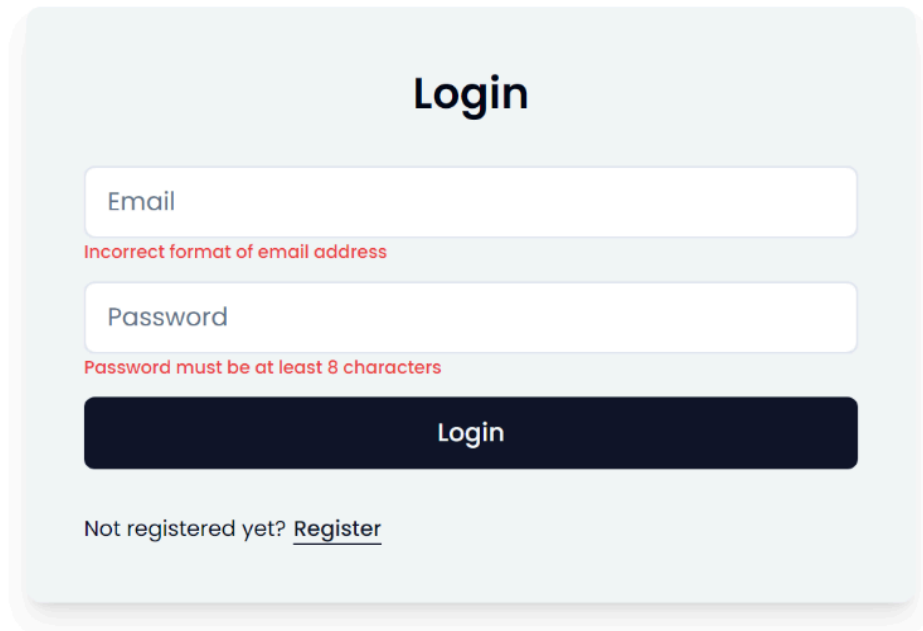


The image shows a registration form titled "Register". It features four input fields: "Email", "Username", "Password", and "Confirm password". Below these fields is a dark blue button labeled "Create an account". At the bottom of the form, there is a link that says "Already have an account? [Login](#)".

Рис. 4.2. Реєстрація користувача

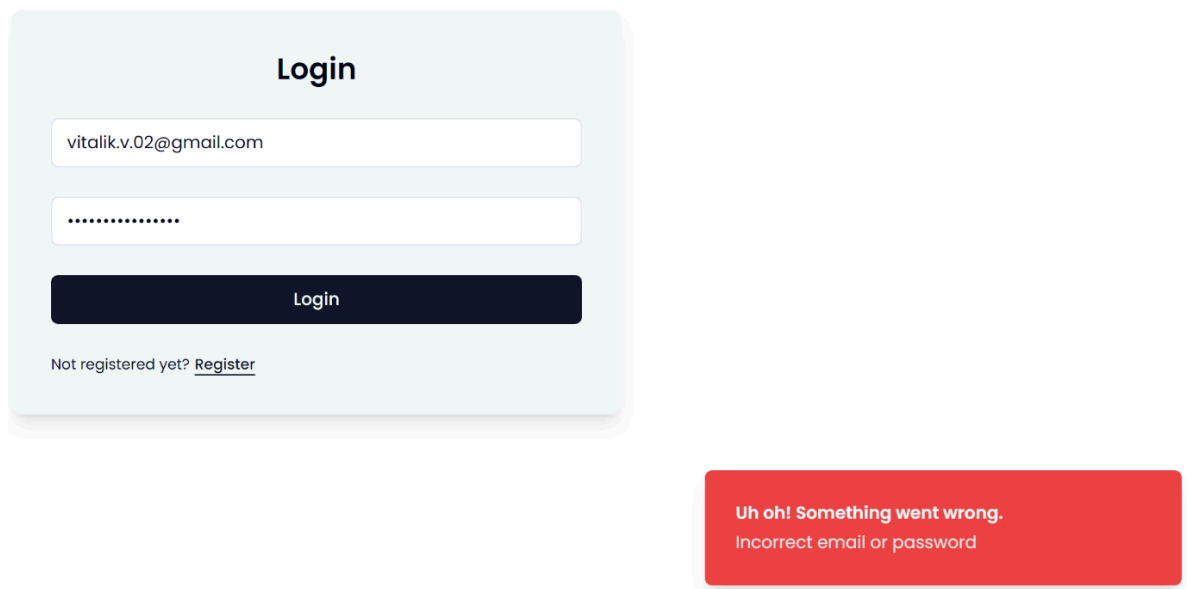
Валідація на стороні сервера та на стороні клієнта повинна проводитися у двох етапах, щоб забезпечити ефективний контроль за вхідними даними. Попри те, що валідація на стороні сервера гарантує безпеку та вірність даних, валідація на стороні клієнта також важлива. Цей метод дозволяє миттєво виявляти та відображати помилки чи неправильні дані, що покращує швидкість

та ефективність взаємодії з вебзастосунками. Цей метод також знижує навантаження на сервер, оскільки частина валідації може бути виконана в браузері користувача, що робить процес більш оптимізованим.



The image shows a login form titled "Login". It has two input fields: "Email" and "Password". Below the "Email" field, there is a red error message: "Incorrect format of email address". Below the "Password" field, there is a red error message: "Password must be at least 8 characters". At the bottom of the form, there is a dark blue "Login" button and a link that says "Not registered yet? [Register](#)".

Рис. 4.3. Валідація вхідних даних при вході у систему



The image shows a login form titled "Login" with the email field filled with "vitalik.v.02@gmail.com" and the password field filled with ".....". Below the form is a dark blue "Login" button and a link that says "Not registered yet? [Register](#)". To the right of the form, there is a red error message box that says "Uh oh! Something went wrong. Incorrect email or password".

Рис. 4.4. Помилка входу у систему

Після завершення реєстрації користувач отримує доступ до головної сторінки, яка містить список активованих користувачів системи.

Hey, Vitalii Voloshyn 🌟
To unlock all features, please activate your account. As soon as you fill in all data, your account will be activated 🌟
[Activate](#)

Explore our community 🙌

Filters












 <p>Oleksandr__Petrenko 19 🇺🇦 Oleksandr here, passionate abo... Level of english: A1 Science Math Programming</p>	 <p>Natalia_Kovalenko 27 🇺🇦 Heeeeey! Natalia here, a book L... Level of english: C1 Reading Cooking Travel</p>	 <p>Carlos Hernandez 35 🇮🇹 Hola amigos! I'm passionate ab... Level of english: A2 Music Programming Travel</p>
 <p>Olivia Brown 38 🇺🇸 Music lover and art enthusiast... Level of english: B1 Music Art Reading</p>	 <p>Innulkinsssk_best 22 🇺🇦 I am from Brovary. I love that... Level of english: C1 Yoga Hiking Pets Dancing Math</p>	 <p>Daniel Wang 27 🇨🇳 Mathematics geek and avid game... Level of english: C2 Math Programming Gaming</p>
 <p>Maria Garcia 32 🇪🇸 Culinary adventurer with a pas... Level of english: C1 Cooking Travel Photography</p>	 <p>Sam Taylor 34 🇬🇧 Bookworm and movie buff. Let's... Level of english: B1 Movies Reading Writing</p>	 <p>Ryan Miller 23 🇺🇸 Tech enthusiast and gamer. Let... Level of english: C2 IT Programming Gaming</p>
 <p>Alice Smith 25 🇫🇷 Hey there! Passionate about tr... Level of english: C1 Travel Photography Reading</p>	 <p>Ruslan Shvets 21 🇺🇦 Hey everyone! I'm passionate a... Level of english: B2 Sport Music Cooking</p>	

Рис. 4.5. Головна сторінка

Доки профіль користувача не активований, автоматизований пошук не застосовується, Також сам користувач не з'являється у списку усіх користувачів, а можливість взаємодіяти з іншими користувачами обмежена. У цей момент користувач має можливість ознайомитися з застосунком та його функціоналом. Як тільки користувач вирішить отримати доступ до усіх можливостей, які надає вебзастосунок, йому необхідно ввести свої особисті дані, щоб активувати профіль. Пізніше він зможе змінити свої дані у разі потреби.

Hey, Vitalii Voloshyn 🙌
To unlock all features, please activate your account. As soon as you fill in all data, your account will be activated ☀️

[Activate](#)

Edit Profile

Username
Vitalii Voloshyn

About me
Hello everyone! Let's talk :)

Date of birth
April 7th, 2002 📅

Gender
 Male
 Female

Country of residence
🇺🇦 Ukraine

Interests

<input checked="" type="checkbox"/> Sport	<input type="checkbox"/> Science	<input checked="" type="checkbox"/> IT	<input type="checkbox"/> Math
<input checked="" type="checkbox"/> Music	<input type="checkbox"/> Art	<input type="checkbox"/> Cooking	<input type="checkbox"/> Travel
<input type="checkbox"/> Photography	<input type="checkbox"/> Fashion	<input type="checkbox"/> Reading	<input type="checkbox"/> Gaming
<input checked="" type="checkbox"/> Movies	<input type="checkbox"/> Gardening	<input type="checkbox"/> Programming	<input type="checkbox"/> Dancing
<input type="checkbox"/> Writing	<input type="checkbox"/> Yoga	<input type="checkbox"/> Hiking	<input type="checkbox"/> Pets

Native language
Ukrainian

Select your level of English
B2

Submit

Рис. 4.6. Редагування профілю

Кожен користувач може у будь-який момент переглянути свій профіль та змінити фотографію за бажанням. Допустимі формати фотографії наступні: jpeg, jpg, png.

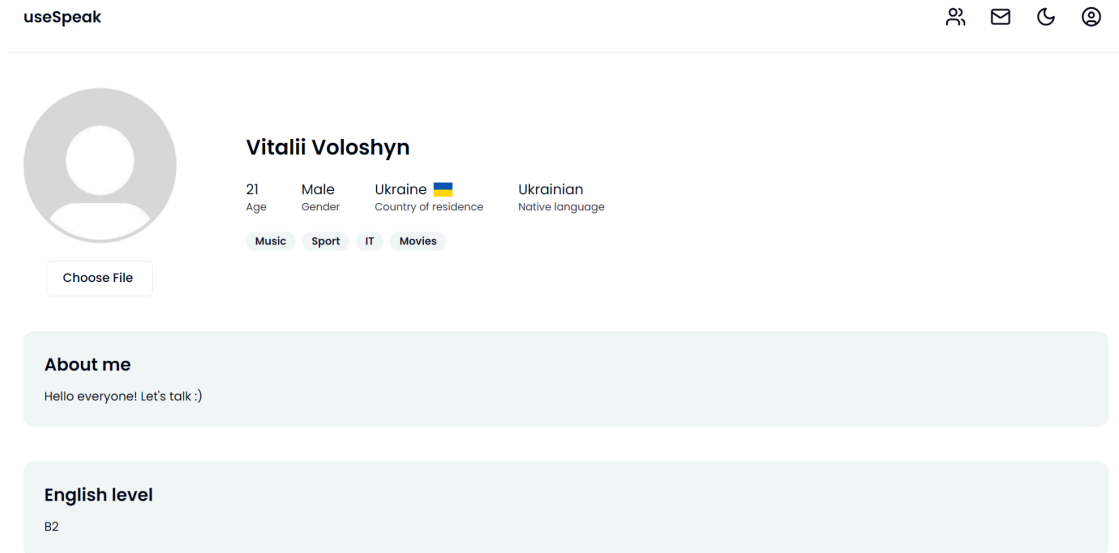


Рис. 4.7. Профіль користувача

Після успішного додавання фотографії, профіль виглядатиме подібним чином, а у правому нижньому кутку з'явиться комунікат, що підтверджує успішність виконаної дії (див. рис. 4.8).

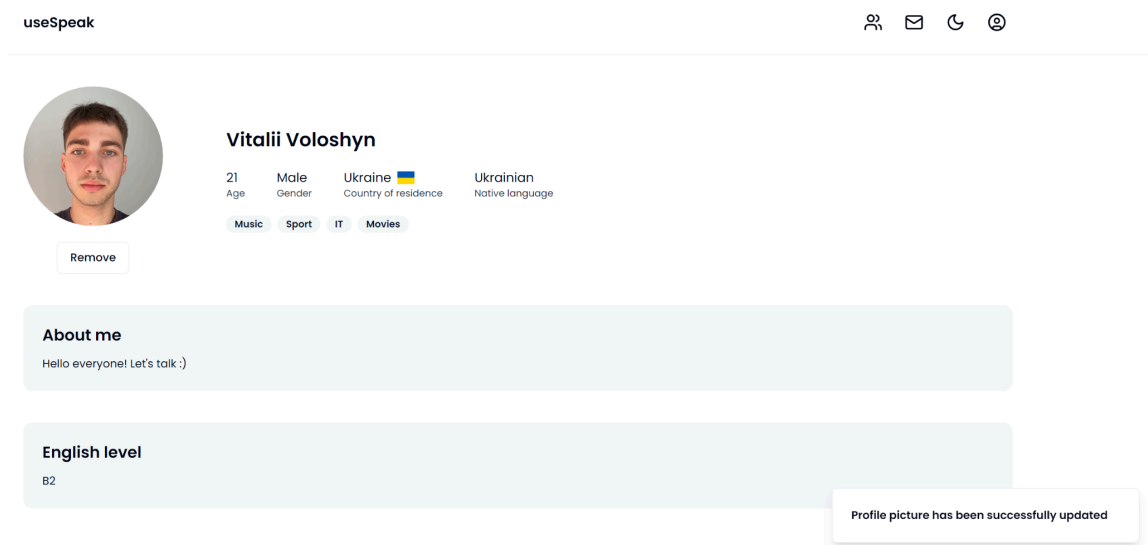




Рис. 4.8. Профіль користувача після додавання фотографії


Після успішної активації профілю, користувач отримує можливість взаємодії з іншими учасниками системи. У цей момент пошук інших користувачів персоналізується під конкретного користувача за найбільшою кількістю спільних інтересів.


Explore our community 🙌


Filters


 Ruslan Shvets 21 🇺🇦
Hey everyone! I'm passionate a...
Level of english: B2
Sport **Music** **Cooking**


 Carlos Hernandez 35 🇮🇹
Hola amigos! I'm passionate ab...
Level of english: A2
Music **Programming** **Travel**


 Olivia Brown 38 🇺🇸
Music lover and art enthusiast...
Level of english: B1
Music **Art** **Reading**


 Sam Taylor 34 🇬🇧
Bookworm and movie buff. Let's...
Level of english: B1
Movies **Reading** **Writing**


 Ryan Miller 23 🇺🇸
Tech enthusiast and gamer. Let...
Level of english: C2
IT **Programming** **Gaming**

 Oleksandr__Petrenko 19 🇺🇦
Oleksandr here, passionate abo...
Level of english: A1
Science **Math** **Programming**

 Natalia_Kovalenko 27 🇺🇦
Heeeeey! Natalia here, a book l...
Level of english: C1
Reading **Cooking** **Travel**

 Innulkinssssk_best 22 🇺🇦
I am from Brovary. I love that...
Level of english: C1
Yoga **Hiking** **Pets** **Dancing**
Math

 Daniel Wang 27 🇨🇳
Mathematics geek and avid game...
Level of english: C2
Math **Programming** **Gaming**

 Maria Garcia 32 🇪🇸
Culinary adventurer with a pas...
Level of english: C1
Cooking **Travel** **Photography**


 Alice Smith 25 🇫🇷
Hey there! Passionate about tr...
Level of english: C1
Travel **Photography** **Reading**

Рис. 4.9. Головна сторінка після активації профілю

Користувач має можливість покращити виведені результати автоматизованого пошуку інших користувачів за допомогою певних критеріїв-фільтрів. Вони охоплюють пошук за гендером, мінімальним вказаним рівнем англійської мови, країни та діапазоном віку користувачів.

Find like-minded people ×

Select gender

Male
 Female

Select minimum level of English

C1 ▼

Select country

United Kingdom ▼

Select age

Min: 20


 Max: 26

Reset
Apply

Рис. 4.10. Фільтрація користувачів

Користувач має змогу переглянути профіль іншого користувача та надіслати запит на спілкування.

useSpeak 👤 ✉️ 🔄 🌐



Send message

Sam Taylor

34 Male United Kingdom English
Age Gender Country of residence Native language

Movies
Reading
Writing

About me

Bookworm and movie buff. Let's discuss stories!

English level

B1

Рис. 4.11. Профіль користувача

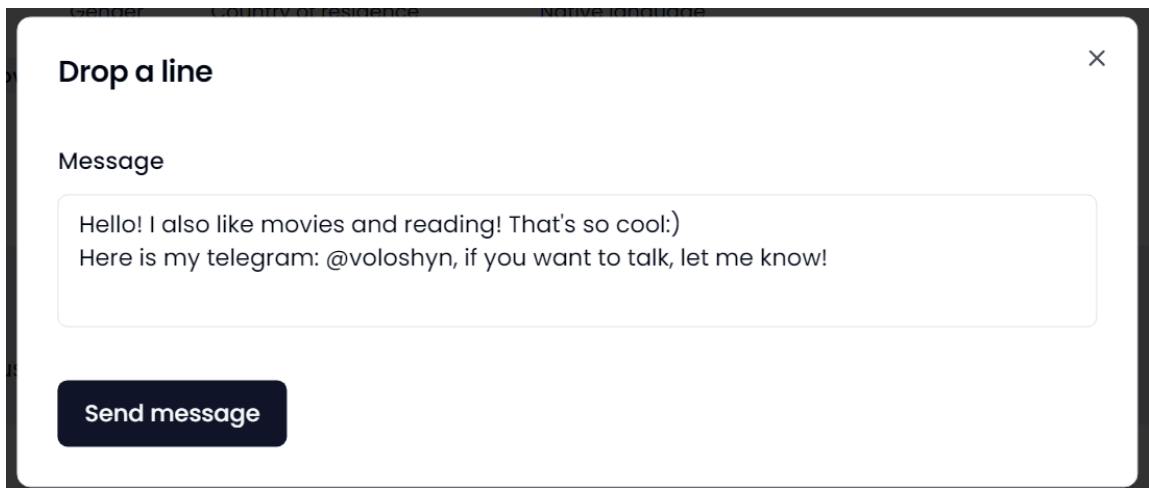


Рис. 4.12. Запит на спілкування

Список відправлених та отриманих повідомлень можна переглянути у відведених для цього вкладці Вох. По лівій стороні будуть знаходитись короткі версії повідомлень, у той час, як по правій їх повний зміст.

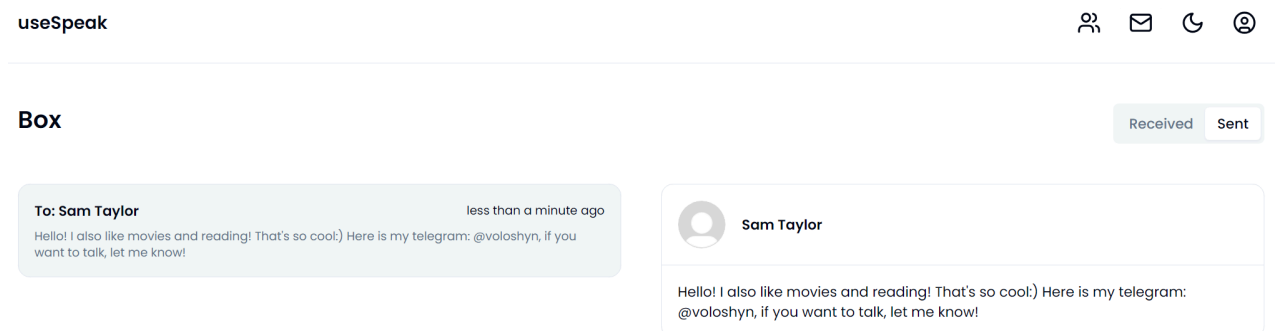


Рис. 4.13. Відправлені повідомлення

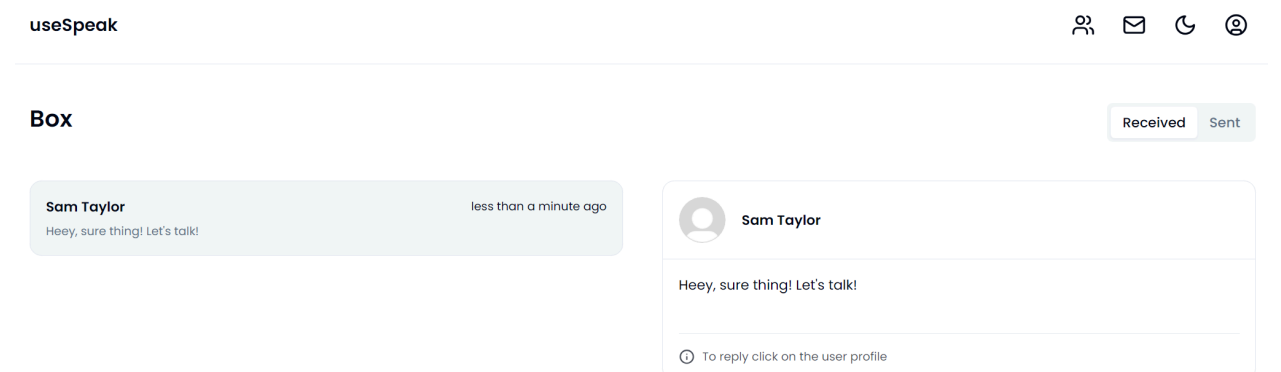


Рис. 4.14. Отримані повідомлення

Користувач має змогу змінити тему застосунку.

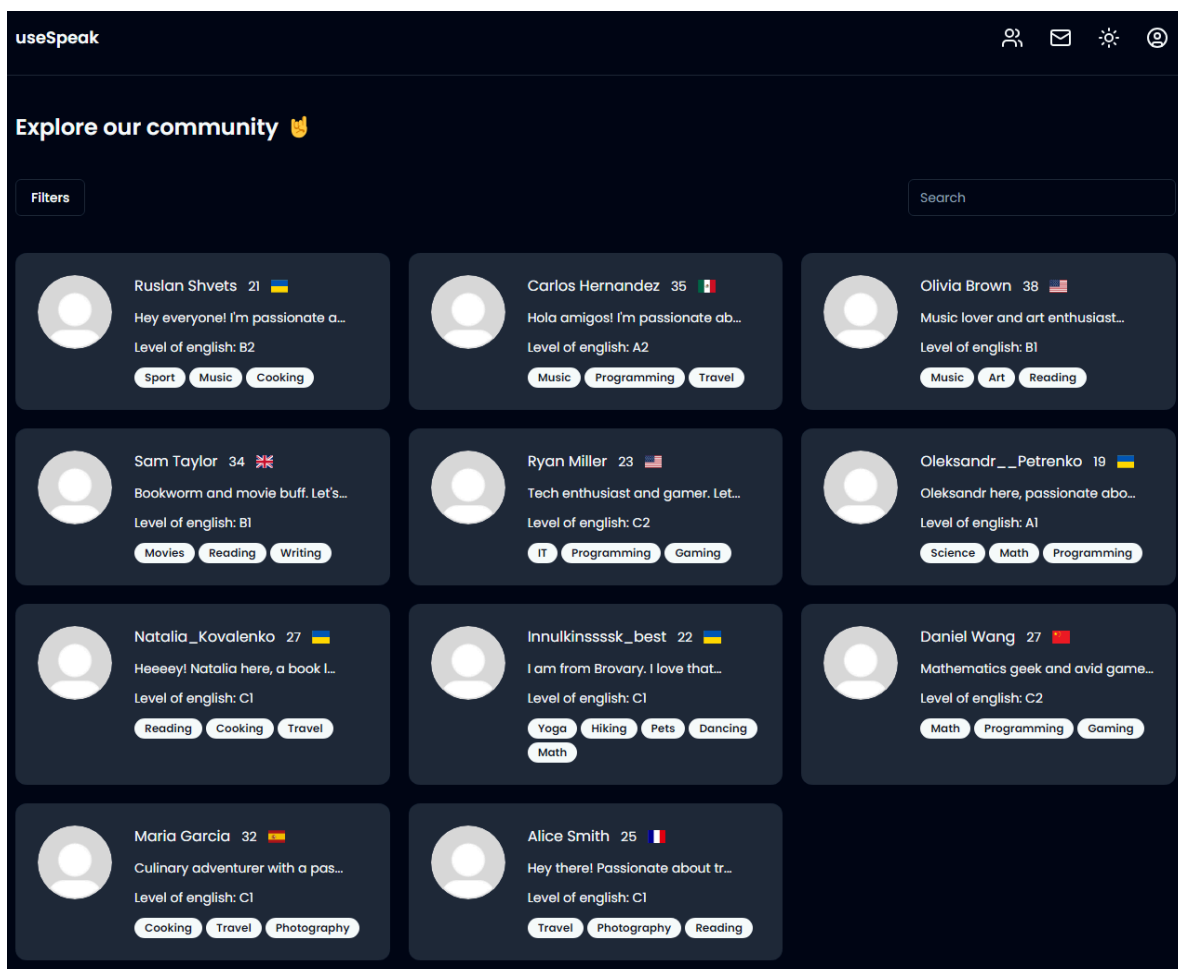


Рис. 4.15. Темна тема застосунку

Висновки

Вебзастосунок вирізняється простим, інтуїтивним та логічним інтерфейсом, що значно полегшує взаємодію та навігацію користувачів у системі. Взаємодія з системою відбувається без зайвих труднощів, а чітка структура інтерфейсу сприяє швидкому досягненню поставлених завдань.

Необхідно відзначити, що простота інтерфейсу не впливає на функціональність вебзастосунка. Система залишається функціонально насиченою, забезпечуючи користувачам широкий спектр можливостей, таких, як реєстрація, вхід у систему, активація та редагування профілю, фільтрація, пошук та взаємодія з іншими користувачами системи.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було детально розглянуто поняття вебзастосунка та проведено порівняльний аналіз між вебзастосунком та вебсайтом. У процесі аналізу предметної області, вивчено та проаналізовано наявні системи із висвітленням їх основних недоліків та переваг, що дало змогу сформулювати чіткі задачі та сформулювати технологічний стек задля перевтілення ідеї в готовий продукт під назвою “Web-застосунок створення комунікаційних майданчиків з вивчення англійської мови”.

Розроблена система не лише відкриває можливості взаємодії шляхом обміну повідомлень між користувачами для покращення комунікативних навичок англійською мовою, але також автоматизує процес пошуку осіб на основі аналізу особистих даних користувача, що сприяє не лише зручності, але й надає персоналізований підхід до вивчення нової мови.

У процесі виконання кваліфікаційної роботи було отримано фундаментальні знання у сфері розробки full-stack додатків, використовуючи технологічний стек MERN та опановано принцип комунікації клієнтської та серверної частини. Завдяки виконанню кваліфікаційної роботи вдалося не лише збагатити теоретичний багаж знань, але й отримати практичні навички, які стануть невіддільною частиною професійного розвитку у цій галузі.

Підсумовуючи, було розроблено онлайн-платформу, націлену на покращення мовної освіти, що не тільки дозволить вдосконалювати свої мовні навички, але й розширювати свої знання про світ та поглиблювати розуміння інших культур. Такий застосунок стане не лише ефективним інструментом для вивчення мови, але й засобом для взаєморозуміння та обміну досвідом, ідеями та поглядами між людьми з різних частин світу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Semmy Purewal. Learning Web App Development : 1st Edition. Sebastopol: O'Reilly & Associates Inc, 2014. 97 с.
2. Emmit Scott. SPA Design and Architecture: Understanding Single Page Web Applications : 1st Edition. New York City : Manning, 2015, 46 с.
3. What is a multi page application. *OhMyCrawl* : вебсайт. URL: <https://www.ohmycrawl.com/faq/what-is-a-multi-page-application/>
4. About Hellotalk. *Hellotalk* : вебсайт. URL: <https://www.hellotalk.com/about>
5. Coffeestrapp. *Coffeestrapp* : вебсайт. URL: <https://www.coffeestrapp.com>
6. About Mylanguageexchange. *Mylanguageexchange* : вебсайт. URL: <https://www.mylanguageexchange.com>
7. Shama Hoque. Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js : 2nd Edition. Birmingham : Packt Publishing, 2020. 484 с.
8. Three-tier architecture overview. *Amazon* : вебсайт. URL: <https://docs.aws.amazon.com/whitepapers/latest/serverless-multi-tier-architectures-api-gateway-lambda/three-tier-architecture-overview.html>
9. Kristina Chodorow. MongoDB: The Definitive Guide: Powerful and Scalable Data Storage : 2nd Edition. California : O'Reilly Media, 2013. 126 с.
10. Explaining BSON with Examples. *MongoDB* : вебсайт. URL: <https://www.mongodb.com/basics/bson>
11. Michael Kaufmann / Andreas Meier. SQL and NoSQL Databases: Modeling, Languages, Security and Architectures for Big Data Management : 2nd Edition. Berlin : Springer, 2023. 347 с.
12. Azat Mardan. Pro Express.js: Master Express.js: The Node.js Framework For Your Web Development : 1st Edition. New York City : Apress, 2014, 144 с.
13. Stephen Ludin / Javier Garza. Learning HTTP/2: A Practical Guide for Beginners : 1st Edition. Newton : O'Reilly Media, 2017. 114 с.
14. React Overview. *React* : вебсайт. URL: <https://react.dev>

15. Mario Casciaro / Luciano Mammino. Node.js Design Patterns: Design and implement production-grade Node.js applications using proven patterns and techniques : 3rd Edition. Birmingham : Packt Publishing, 2020. 520 с.
16. An Introduction to JavaScript. *Javascript.info* : вебсайт. URL: <https://javascript.info/intro>
17. Josh Goldberg. Learning TypeScript : 1st Edition. Sebastopol : O'Reilly Media, 98 с.
18. Introduction Shadcn. *Shadcn* : вебсайт. URL: <https://ui.shadcn.com/docs>
19. Documentation for Visual Studio Code. *Visual Studio* : вебсайт. URL: <https://code.visualstudio.com/docs>
20. Getting Started - What is Git? *Git* : вебсайт. URL: <https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F>
21. What Is GitHub? A Beginner's Introduction to GitHub. *Kinsta* : вебсайт. URL: <https://kinsta.com/knowledgebase/what-is-github>
22. Introduction to JSON Web Tokens. *Jwt.io* : вебсайт. URL: <https://jwt.io/introduction>