

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ

ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

ПОЯСНЮВАЛЬНА ЗАПИСКА

до бакалаврської роботи

на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ВИВЧЕННЯ
АНГЛІЙСЬКОЇ МОВИ «ENGLISH CHATTING CLUB» ЗА ДОПОМОГОЮ
МОВИ ПРОГРАМУВАННЯ JAVASCRIPT»**

Виконав: студент 4 курсу, групи ПД– 41

спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Чередніченко М.Д.

(прізвище та ініціали)

Керівник Дібрівний О.А.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормоконтроль _____

(прізвище та ініціали)

Київ – 2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення
Ступінь вищої освіти -«Бакалавр»
Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Інженерії програмного забезпечення
Негоденко О.В.
“ ____ ” _____ 2021 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА

ЧЕРЕДНІЧЕНКО МАКСИМУ ДМИТРОВИЧУ

(прізвище, ім'я, по батькові)

1. Тема роботи: **«Розробка мобільного додатку для вивчення англійської мови «English Chatting Club» за допомогою мови програмування Javascript»**

Керівник роботи: Дібрівний О.А., Старший викладач кафедри ПЗ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «12» березня 2021 року №65.

2. Строк подання студентом роботи «01» червня 2021 року

3. Вхідні дані до роботи

Visual Studio Code;

Android;

Освітні додатки.

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

4.1 Характеристика освітніх додатків.

4.2 Опис технологій.

4.3 Опис проектування системи.

5. Перелік демонстраційного матеріалу (назва основних слайдів)

1. Мета, об'єкт та предмет дослідження
2. Аналоги
3. Порівняння
4. Технічне завдання
5. Програмні засоби реалізації
6. Фреймворк Socket.IO
7. Висновки

6. Дата видачі завдання «19» квітня 2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	30.02-20.03	Виконано
2	Вимоги до системи	22.03-30.03	Виконано
3	Аналіз додатків конкурентів	02.04-05.04	Виконано
4	Концепція та архітектура програмного забезпечення	08.04-30.04	Виконано
5	Вступ, висновки, реферат	02.05-8.05	Виконано
6	Розробка обов'язкових демонстраційних матеріалів	08.05-10.05	
7	Попередній захист роботи	11.05	
8	Здача роботи	01.06	

Студент _____
(підпис)

(прізвище та ініціали)

Керівник роботи _____
(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 34 с., 29 рис., 12 джерел.

REALTIME CHAT, JAVASCRIPT, HTML, CSS, SOCKET IO, NODE JS,
REPL IT, EXPRESS

Об'єкт дослідження – додатки для вивчення англійської мови.

Предмет дослідження – найбільш використовувані функції застосунків для вивчення англійської мови.

Мета роботи – розробити додаток для поліпшення та подальшого розвитку вербальних навичок володіння англійською лексикою.

Науковою новизною даної роботи є створення додатку із раніше не запропонованим функціоналом серед подібних варіантів. Таким функціоналом є система для мовної практики з живими людьми, незалежно від рівня володіння.

У роботі проведено аналіз існуючих додатків, таких як Duolingo, Lingualeo, Memrise, ReWord, Cake. Визначено найбільш важливі та конкуруючі функції таких додатків.

Загальною проблемою цих продуктів є відсутність функціоналу для практики вивченого матеріалу на реалістичних випадках.

На основі результатів проведених досліджень було здійснено розробку додатку для практики англійської мови з використанням мови програмування JavaScript та середовища розробки Repl.it.

ЗМІСТ

РЕФЕРАТ	4
ЗМІСТ.....	5
ВСТУП	8
РОЗДІЛ 1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА ДОДАТКІВ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ	10
1.1. Огляд основних функцій та можливостей.....	10
1.2. Дослідження аналогів	11
1.2.1. Duolingo	12
1.2.2. Lingualeo	14
1.2.3. Memrise	17
1.2.4. ReWord.....	19
1.2.5. Cake	21
1.3. Порівняльна таблиця функцій додатків.....	23
1.4. Висновок по розділу	24
РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ.....	25
2.1. Особливості ОС Android	25
2.2. Етапи розробки.....	26
2.3. Структура програми	26
2.4. Вибір технологій для розробки мобільного додатка	28
2.4.1. JavaScript.....	28
2.4.2. Websocket & Socket io	30
2.4.3. Node JS	32
2.4.4. Express Framework	33
2.4.5. Технологія PWA.....	34

2.5. REPL.IT	35
2.6. Висновок по розділу	37
РОЗДІЛ 3. ДЕМОНСТРАЦІЯ РОБОТИ ЗАСТОСУНКУ	38
3.1. Реалізація додатку	38
ВИСНОВКИ	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	46
ДОДАТОК А	47
ДОДАТОК Б	54
ДОДАТОК В	56
ДОДАТОК Г	59

ВСТУП

В теперішній час люди проводять величезну кількість часу зі своїми пристроями, через що з'явився тренд на використання даних засобів з метою поліпшення свого життя. Існує безліч додатків для навчання або отримання інших навичок, для заняття спортом і дотримання дієт, творчості, створення музики і живопису і т.д. Це дає пересічному користувачеві свободу отримувати нові знання в зручному для нього темпі і в зручний йому час.

Англійська мова є загальновизнаною мовою світових комунікацій. Знання цієї мови відкриває доступ до великого пласту інформації: дозволяє читати книги та дивитися фільми мовою оригіналу, також надає можливість отримувати нові знання на іноземних ресурсах в мережі інтернет. Програмістам і людям, зайнятим в IT-сфері та суміжними з нею, знання цієї мови є необхідним для свого роду діяльності.

В наш час через бурхливий темп життя і роботи, багато людей не можуть дозволити собі індивідуальних репетиторів або курси вивчення мов, так як це займає чимало часу. Люди проводять десятки годин на роботі або в транспорті дорогою на роботу кожен місяць. Наразі, велика кількість людей володіє одним або декількома гаджетами, за допомогою яких вона може поліпшити життя. На допомогу приходять сервіси і додатки, які допомагають вивчати мови, витрачаючи на них всього декілька десятків хвилин кожного дня. Такі застосунки і сервіси можна використовувати як самостійно, так і в парі з вивченням мови за допомогою репетитора або на курсах.

Дані додатки створені для людей з різним рівнем володіння англійською мовою. Зазвичай, застосунки на початку роботи з ними пропонують користувачам пройти невелике тестування на відповідність певному міжнародному рівню і вже після цієї процедури підлаштовуються, адаптуючи навчальну програму під вас. Ви можете вивчати мову в зручному для вас режимі і у вільний час. Застосунки можна використовувати як самостійний інструмент поповнення лексики і повторення граматики, так і в якості додаткового елемента, який допоможе зробити ваше

навчання на курсах англійської або у репетитора ще більш ефективним. Не варто сприймати ці додатки лише як допоміжний інструмент при вивченні іноземної мови, так як вони містять в собі повний набір лінгвістичного функціоналу для повного занурення у предмет вивчення: граматичні вправи, завдання на збільшення лексичного запасу, а також тренування на слухове сприйняття, які допомагають поліпшити ваші усні навички. Додатки такого типу, в більшості випадків, побудовані на основі ігрової форми та містять барвисті зображення, системи мотивації користувача і системи винагороди за виконані завдання, які спонувають до подальшого вивчення. Це дозволяє комфортно використовувати дані додатки людям різного віку для продуктивного і цікавого вивчення іноземних мов.

Актуальність обраної теми полягає в декількох речах, а саме:

- Вирішення проблеми відсутності можливості мовної практики «у смартфоні»
- Вирішення проблеми неможливості збиратися у групи в живу та практикувати мову очно, через пандемію коронавірусу.

Науковою новизною даної роботи є створення додатку із раніше не запропонованим функціоналом серед подібних варіантів. Таким функціоналом є система для мовної практики з живими людьми, незалежно від рівня володіння.

Метою даної роботи є розробка додатку для поліпшення та подальшого розвитку вербальних навичок володіння англійською лексикою.

Завдання бакалаврської роботи:

- проаналізувати існуючі додатки по вивченню іноземних мов;
- виявити недоліки існуючих рішень та запропонувати заходи для їх усунення;
- аналіз можливостей операційної системи Android з точки зору розробки додатків;
- розробити додаток для практики англійської мови у вигляді чату на платформу Android.

РОЗДІЛ 1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА ДОДАТКІВ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ

1.1 Огляд основних функцій та можливостей

З розвитком смартфонів, планшетів і мобільних технологій, ринок додатків для вивчення іноземних мов почав швидко зростати. В даний час існує величезна кількість застосунків, як від великих компаній, так і від незалежних розробників. Вивчення мов стало трендом в часи інтернету, тому сотні тисяч людей щодня користуються такими додатками.

Більшість з існуючих рішень пропонують такий функціонал:

- вивчення нових слів по карткам;
- вивчення граматики та подальше тестування;
- аудіювання;
- переклад певного тексту.

Головним недоліком кожного з них є відсутність практики з носіями мови або людьми які також її вивчають. У людей з хорошим словниковим запасом і відмінними знаннями граматики виникає проблема при безпосередньому спілкуванні з людьми коли потрібно застосувати здобуті знання на практиці.

За статистикою, станом на 2020 рік найбільше зростання користувачів було відзначено саме в категоріях «бізнес» та «освіта». За прогнозами світових аналітичних компаній кількість створюваних додатків буде тільки зростати з кожним роком, а такі категорії як «бізнес» і «освіта» будуть і далі очолювати лідерські позиції.

В більшості випадків, швидке зростання зумовлене також і у зв'язку з поширенням пандемії, через яку освітні курси або школи закриті на невизначений час та збільшеною кількістю вільного часу у людей, через відсутність необхідності їхати в офіс, який вони можуть провести з користю.

1.2 Дослідження аналогів

Магазин додатків Play Market містить дуже багато різних додатків від різних розробників, як великих компаній типу ABBYY lingvo, так і незалежних невеликих студій або програмістів-одинаків. На рисунку 1 можна побачити топ-додатків представлених по пошуковому запиту «Вивчення англійської».

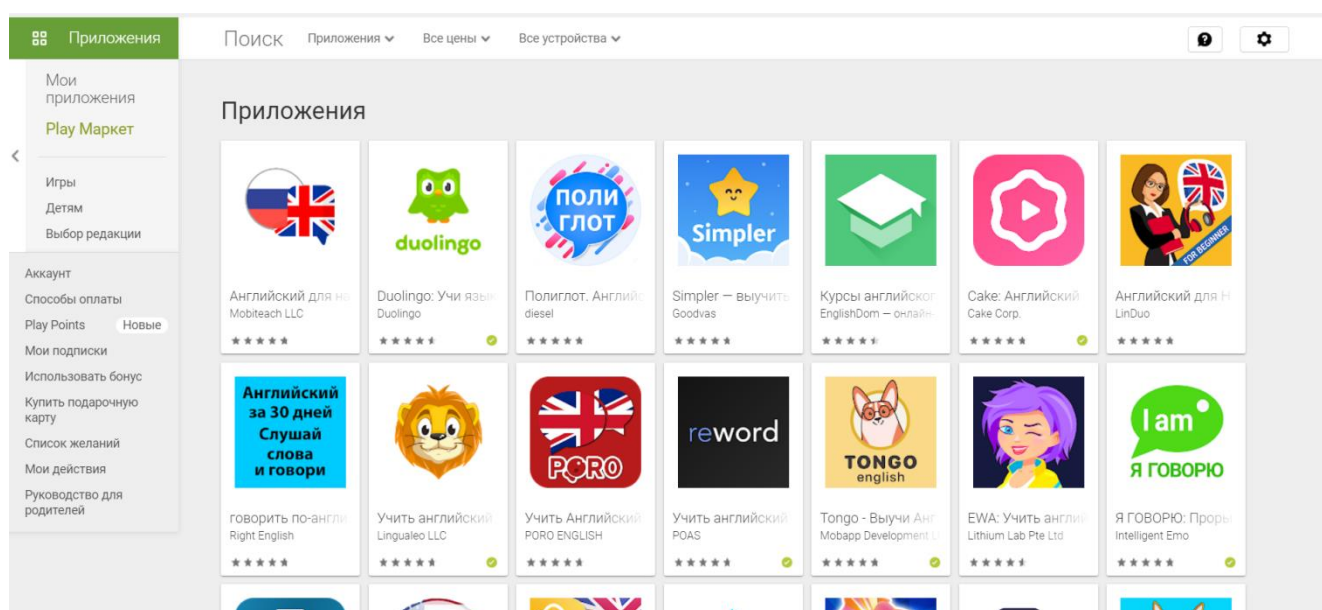


Рисунок 1 - Топ додатків пошукового запиту «Вивчення англійської»

Можна простежити, що здебільшого, лідерами є саме ті програми, що пропонують вивчення нових слів по системі інтервального повторення, такі як:

- Simpler;
- ReWord;
- Memrise;
- Worldwide.

А також програми, що пропонують комплексні рішення для вивчення граматики, лексики, усного сприймання та промови, такі як:

- Duolingo;
- Lingualeo;
- Tongo;

- EWA.

Тож серед усіх існуючих додатків, які займають лідерські позиції, я обрав декілька, які, на мою думку, мають найпривабливіший і затребуваний функціонал.

1.2.1 Duolingo

Duolingo пропонує численні письмові уроки і диктанти, проте розмовним навичкам приділяється менше уваги. У Дуолінго є ігрове дерево навичок(рис 1.1), за яким просуваються користувачі, і словниковий розділ, де можна практикувати вже вивчені слова.



Рисунок 1.1 - Дерево навичок

Користувачі отримують «Досвід» (монети, бали) у міру вивчення мови, наприклад, після проходження уроку. Навички вважаються вивченими, коли

користувачі виконують всі пов'язані з ними уроки. За один урок можна заробити 10 очок. У Дуолінго також є функція тренування на час, коли користувачам дається 30 секунд і двадцять питань. За кожен правильну відповідь дається одне очко досвіду і сім або десять додаткових секунд (час залежить від довжини питання). За один курс користувач може вивчити до 2000 слів.

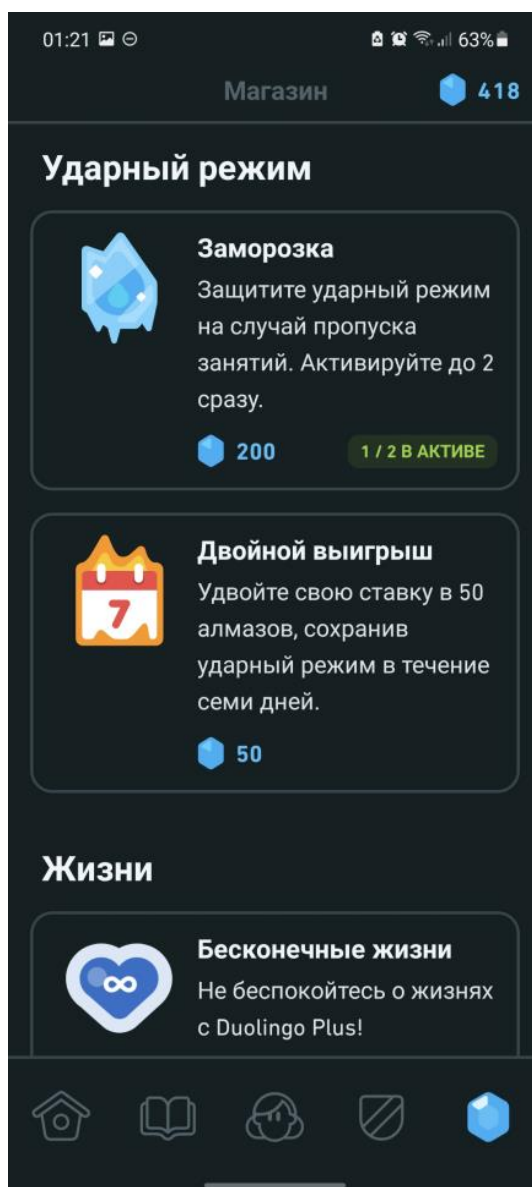


Рис 1.2 - Внутрішній магазин

За проходження всіх уроків в категорії користувачу видаються Алмази, внутрішня ігрова валюта. Алмази можна витратити в ігровому магазині(рис.1.2) або дарувати користувачам, які залишили корисний коментар на форумі.

Дуолінго використовує для навчання підхід, заснований на аналізі великої кількості статистичних даних. На кожному етапі система запам'ятовує, які питання викликали у користувачів труднощі і які помилки були здійснені. Потім вона агрегує ці дані і використовує для машинного навчання. Таким чином формуються індивідуальні уроки.

До переваг можна віднести:

- зручний інтерфейс користувача;
- систему мотивації;
- подачу матеріалу в ігровій формі;
- побудову завдань, згідно з якими новачкам буде не складно почати навчання.

До недоліків можна віднести:

- відсутність вибору рівня за яким можливо навчатись;
- відсутність можливості обрати тип заняття(аудіювання, письмо, граматика).

1.2.2 Lingualeo

Додаток для вивчення англійської мови. У ньому більше різноманітних уроків, ніж в Duolingo: реальні тексти, відео, різні типи завдань(рис.1.3).

Користувач починає з проходження тесту на знання мови і заповнює список своїх інтересів. На їх основі Lingualeo створює персональний план навчання, виконання якого в різних категоріях навичок - від сприйняття мови на слух до зростання словникового запасу і зменшення числа граматичних помилок - користувач бачить весь прогрес в особистому кабінеті.

Lingualeo пропонує вивчати мову на цікавих користувачеві матеріалах: аудиокнигах і піснях, відеозаписах або текстах, розміщених у відкритих джерелах або завантажених іншими учасниками(рис.1.4). Серед джерел оригінального тексту і аудіо - виступи на конференціях TED і курси освітнього сайту Coursera.

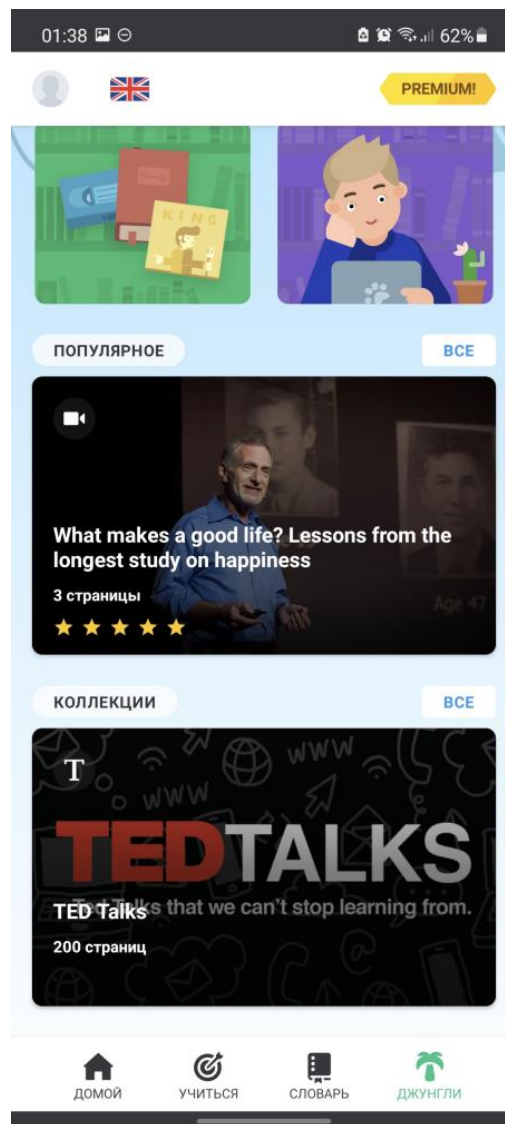


Рисунок 1.3 - Типи завдань у додатку Lingualeo

В процесі користувач може самостійно вибрати незнайомі слова для вправ або використовувати тематичні добірки. Доступні тренування граматики і вимови, ігри, особистий словник з асоціаціями і журнал, в якому зазначається прогрес навчання.

Тож перевагами цього додатку є:

- велика кількість різних завдань для виконання;
- початковий тест, який адаптує навчальну програму під рівень знань користувача;
- створення персонального навчального плану.



- Рисунок 1.4 - Матеріали для вивчення

Недоліки:

- нав'язлива реклама підписки на додаток;
- автоматична вивантаження застосунку з фонових процесів смартфона, що може спричинити незручності при використанні додатка;
- відсутність темної теми.

1.2.3 Memrise

Додаток, спрямований допомогти запам'ятовувати слова і фрази,

використовуючи інтервальні повторення(рис.1.5).

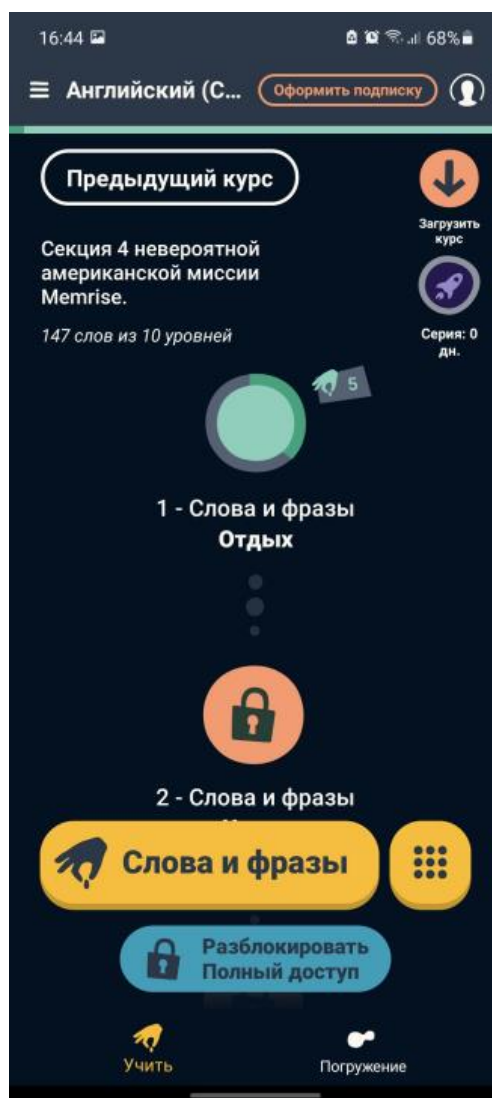


Рисунок 1.5 - Интерфейс додатку Memrise

Додаток містить велику базу озвучених слів та фраз, а також для багатьох записано відео їх вимови різними людьми з різними акцентами(рис.1.6). Існують готові списки слів і фраз для різного рівня володіння мовою і на різні теми. Розробники радять поєднувати використання програми та сайту: вони синхронізують призначений для користувача прогрес. На сайті присутня можливість додавати до складних слів картинки і / або текстові асоціації. У мобільній версії цей функціонал не було реалізовано.

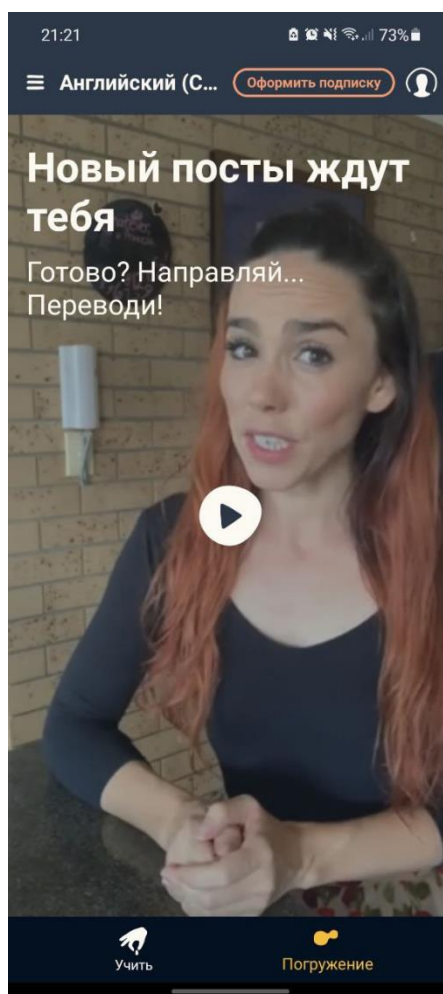


Рисунок 1.6 – Розділ з записаними відео вимови різних слів

Перевагами цього додатку можна вважати:

- майже всі слова озвучені носіями мови;
- зручний інтерфейс;
- велика кількість доступних мов для вивчення;
- адаптивний механізм навчання;
- система мотивації.

Недоліками можна вважати:

- обмежений функціонал в рамках безкоштовної підписки.

1.2.4 ReWord

Додаток-словник(рис.1.7), в якому реалізована система вивчення слів по

карткам та інтервального повторення вивченого. Включає в себе великий список необхідних слів, які можна вибирати в залежності від необхідного рівня або групи слів, наприклад, “Oxford-3000” – найбільш використовувані слова у повсякденному житті. Також користувач має можливість додати свої слова до словника та періодично повторювати їх.

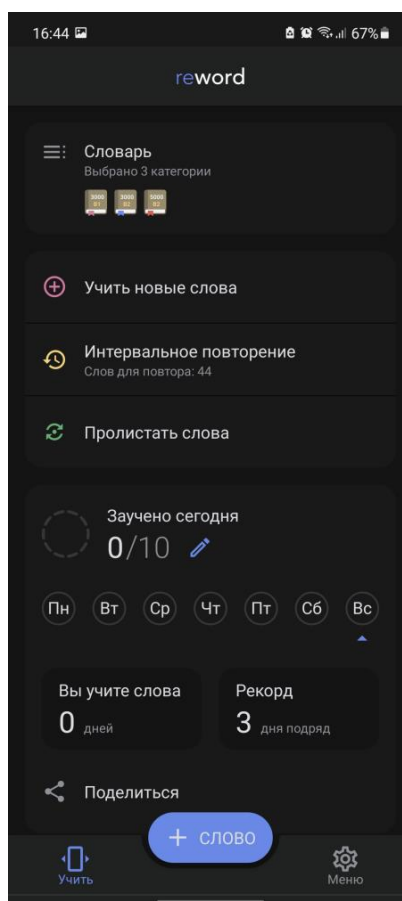


Рисунок 1.7 - Інтерфейс додатку ReWord

Містить в собі систему відстеження відвідувань та рекордів користувача, на основі яких додаток надсилає повідомлення у разі пропуску більш ніж 3 днів підряд. Також, підраховується кількість вивчених слів та слів які ще треба повторити, для того, аби перейти до наступних по порядку.

При повторенні слова(рис.1.8) у користувача є 3 кнопки, які пропонують наступний функціонал:

- ввести з клавіатури;

- подивитись переклад;
- обрати з 4-х варіантів;
- згадав слово;
- не згадав слово.

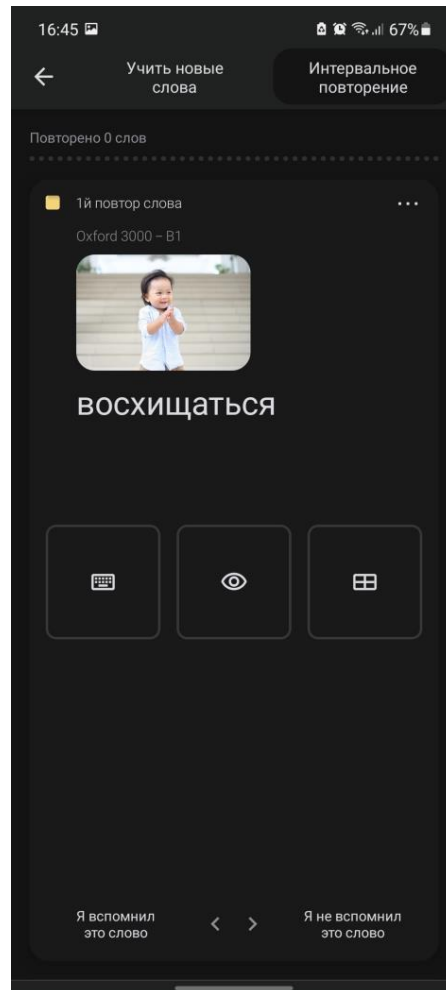


Рисунок 1.8 - Экран повторения слова

Преваги:

- великий словниковий запас;
- можливість додавати свої слова для вивчення;
- інтуїтивний інтерфейс;
- низька ціна за необмежений доступ до програми.

Недоліки:

- не всі слова мають показову картинку та озвучення.

1.2.5 Cake

Додаток, у якому реалізований функціонал аудіювання(рис.1.9). Користувач підписується на певні мовні групи та переглядає відео з YouTube, у яких присутні ключові фрази, які застосунок виділяє їх та користувач впродовж трьох повторів має запам'ятати дані слова або словосполучення. Завдяки вбудованому штучному інтелекту користувачу час від часу пропонується пройти завдання на повторення вивченого.

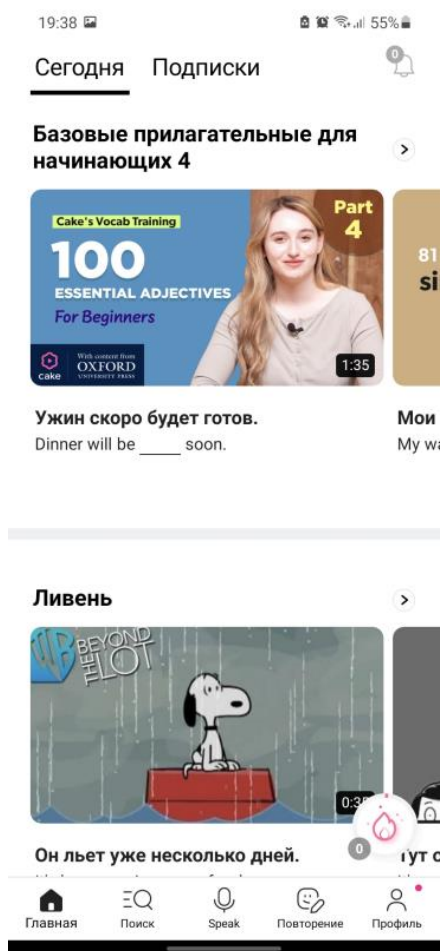


Рисунок 1.9 - Интерфейс додатка Cake

Користувач має змогу вивчати фрази, які є найбільш вживаними у повсякденному житті та комунікації. Вбудований помічник допоможе оцінити правильність промови. Також, у цьому додатку реалізований функціонал

практичних уроків, які імітують діалог з носієм мови. Користувач може спостерігати та слухати як проходить діалог. За бажанням, можливо додати фразу або слово до словника і повернутися до нього ще раз пізніше.



Рисунок 1.10 - Система нагороди за відвідування

За вдало пройдені завдання користувач отримує нагороду у вигляді «зірок»(рис.1.10), які просувають його вище за таблицею лідерів. Деяких людей це може непогано мотивувати відвідувати додаток кожен день та виконувати якусь кількість завдань, щоб заробити якомога більше «зірок» та потрапити на найвище місце у таблиці.

Переваги:

- система мотивації користувача;
- приклади життєвих діалогів, завдяки яким можливо засвоїти самі використовувані фрази або запитання та відповіді на них;
- штучний інтелект, який підбирає контент на основі інтересів користувача;
- повністю безкоштовне використання.

Під час використання додатка я не виявив в ньому недоліків.

1.3 Порівняльна таблиця функцій додатків

Функції	Duolingo	Lingualeo	Memrise	ReWord	Cake
Вправи на запам'ятовування слів	+	+	+	+	+
Озвучення слів	+	+	+	+	+
Візуалізація слів	+	+	-	+	-
Система мотивації користувачів	+	-	-	+	+
Вартість використання	Від 0,99 \$ до 125,99 \$	Від 0,99 \$ до 374,99 \$	Від 0,99 \$ до 139,99 \$	Від 0,99 \$ до 25,99 \$	-
Розмовна практика	-	-	-	-	+
Кількість користувачів	>100 000 000	>5 000 000	>10 000 000	>1 000 000	>50 000 000

Таблиця 1. Порівняння функціоналу додатків

1.4 Висновок по розділу

В ході аналізу сучасних аналогів мобільних додатків для практики англійської мови було виявлено, що користувачі здебільшого звертають увагу саме на аудіо-візуальний метод вивчення англійської при використанні застосунків на смартфонах. Тобто на додатки у яких є присутнім більше картинок або аудіо матеріалів, таких як у додатку Cake або Lingualeo.

Також, ми маємо звернути увагу на те, що зараз на ринку мобільних навчальних додатків відсутнє рішення для практики в умовах реальних випадків спілкування. Реалізація саме цього функціоналу, на мою думку, зможе вирішити проблему з відсутністю можливості відточувати граматичні та лексичні здобутки маючи діалог з іншими користувачами на різноманітні теми.

Саме тому було вирішено зробити акцент на цьому і розробити додаток, у якому можливо буде декільком особам спілкуватися між собою, тим самим покращувати свої навички користування лексикою та граматиною.

Відрізнитися від аналогів додаток буде наступними речами:

- простота у використанні;
- соціальна складова;
- великий вибір тем для розмови;
- можливість залишитися анонімом;
- перспективи розвитку;
- кросплатформеність;
- відсутність обмеження по рівням;

Розрахункова аудиторія користувачів цього додатка - особи від 14 до 65 років, з різним рівнем знань.

2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Особливості ОС Android

Android – операційна система для смартфонів, і багатьох інших пристроїв, наприклад телевізорів, годинників, планшетів і так далі. Була розроблена компанією Android Ink, котру пізніше викупив американський гігант Google.

Наразі частка ОС Android на ринку складає приблизно 85%. Вона є популярною для розробників через те, що має ряд функціональних переваг над її прямим конкурентом IOS, наприклад:

- Інструменти розробника реалізовані краще та доступні для усіх охочих, в той час як для розробки під IOS треба мати підходяще середовище та інструментарій

- Доступність вихідного коду. Більша частина коду Android є відкритою, тобто будь-хто може завантажити його на свій комп'ютер та розібратися як все влаштовано та працює.

- Якісна документація. Для Android розробників створено багато посібників, які скорочують час на пошук варіантів вирішення виникаючих проблем або питань в процесі розробки ПЗ.

- Велика кількість ентузіастів. Для новачків існує багато ресурсів, де їм зможуть допомогти більш досвідчені програмісти та де вони зможуть подивитися приклади реалізації того, чи іншого коду.

- Полегшена процедура публікації застосунків у магазині Google play та одноразовий платіж за просування, а не щорічний у App Store за кожний створений додаток.

- Сфери застосування. Для ОС Android програмують застосунки не тільки для смартфонів, а й для розумних годинників, Smart TV, гаджетів віртуальної та доповненої реальності, приладів штучного інтелекту, екосистем «розумний дім» та всіх супутніх приладів до нього.

2.2 Етапи розробки

Порядок розробки застосунку:

- Аналіз предметної області;
- Вибір засобів для реалізації;
- Розробка технічного завдання;
- Опанування програмного забезпечення;
- Розробка клієнт-серверного додатка;
- Проведення тестування.

2.3 Структура програми

На рисунку 2 зображена загальна структура програми.

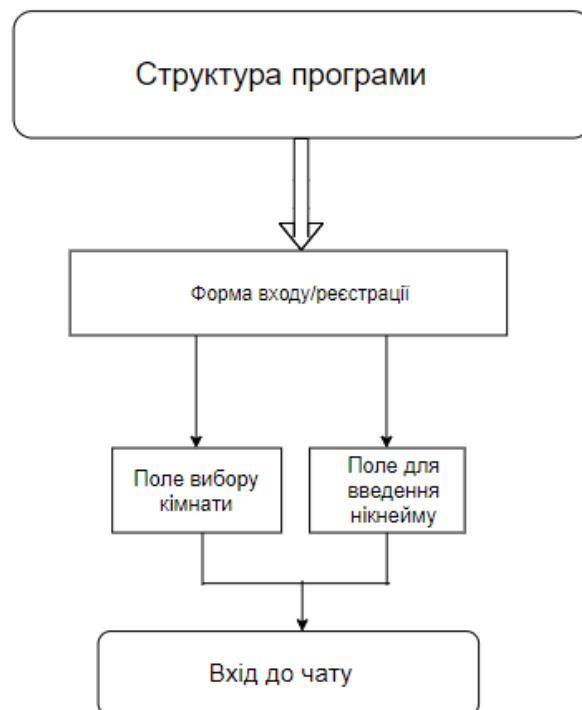


Рис. 2 – загальна структура програми

Програма містить наступні компоненти:

- Форма входу/реєстрації. Для початку роботи з додатком користувачу необхідно увійти в обліковий запис, використовуючи один з наступних способів ідентифікації:
 - Google;
 - GitHub;
 - електронну пошту та пароль.

Це потрібно для надання користувачеві можливості користуватися всім функціоналом додатка, подальшої його ідентифікації у системі, доступу до соціальної складової застосунка та можливості допомогти у разі виникнення проблем під час його використання.

- Поле вибору кімнати. Користувачу буде запропоновано обрати одну з доступних кімнат за інтересом. Всього буде інтегровано 6 тематик у відповідних розділах, серед них:
 - кіно;
 - музика;
 - ігри;
 - загальний;
 - бізнес;
 - подорожі.
- Поле для введення нікнейму. Користувач має придумати та ввести ім'я, яке буде задовольняти користувача та не буде порушувати правила використання додатку та яке буде відображатися в чаті для інших людей.
- Вхід до чату. Кнопка, яка додає користувача до обраної кімнати та передає боту інформацію з нікнеймом, який при вході в чат вітає юзера, використовуючи надану інформацію.

2.4 Вибір технологій для розробки мобільного додатка

На сьогоднішній день лідерами серед мобільних ОС є Android та IOS. В процесі дослідження теми виявилось, що розробка для обох систем нативного додатку зайняла б дуже багато часу, в особливості на вивчення мови програмування Swift або Objective-C, а для Android – Java або Kotlin. Саме тому, аби мати більшу аудиторію користувачів я вирішив використати для розробки мову програмування JavaScript, Node.JS, фреймворк Socket.io та перспективну технологію PWA, яка дасть змогу використовуючи браузер зробити додаток на смартфоні за лічені кроки та можливість користуватися розробкою не лише на смартфоні, а й за необхідністю на комп'ютері. В якості середовища розробки буду використовувати cloud based IDE - REPL.IT.

2.4.1 JavaScript

JavaScript (Часто скорочувано **JS**) це легка, інтерпретована, об'єктно-орієнтована мова з першокласними функціями і найбільш відома як мова сценаріїв веб-сторінок, але він використовується і в багатьох не браузерних середовищах[1].

Це динамічна мова сценаріїв на основі прототипів, яка підтримує об'єктно-орієнтований, імперативний та функціональний стилі програмування.

```
<script>  
  
    // Here is your code  
  
</script>
```

Рисунок 2.2 – Вбудований у HTML скрипт

JavaScript працює на клієнтській частині веб-додатка, що може бути

використано для проектування/програмування поведінки веб-сторінок у випадку виникнення події. JavaScript - це проста у вивченні, а також потужна мова сценаріїв, широко використовувана для контролю поведінки веб-сторінок.

Клієнтський JavaScript - найпоширеніша форма мови. Сценарій повинен бути включений у документ HTML(рис.2.2) або посилатися на нього(рис.2.3), щоб код був інтерпретований браузером.

Це означає, що веб-сторінка не повинна бути статичним HTML, але може включати програми, які взаємодіють з користувачем, керують браузером та динамічно створюють вміст HTML.

```
<script src="script.js" ></script>
```

Рисунок 2.3 – Посилання на скрипт

Клієнтський механізм JavaScript надає багато переваг перед традиційними серверними сценаріями CGI. Наприклад, ви можете використовувати JavaScript, щоб перевірити, чи ввів користувач правильну адресу електронної пошти у поле форми. Код JavaScript виконується, коли користувач подає форму, і лише якщо всі записи є дійсними, вони будуть надіслані на веб-сервер[2].

```
<html>
  <body>
    <script language = "javascript" type = "text/javascript">
      <!--
        document.write("Hello World!")
      //-->
    </script>
  </body>
</html>
```

Рисунок 2.4 – Виведення на екран фрази «Hello World!»

JavaScript можна використовувати для захоплення ініційованих

користувачем подій, таких як клацання кнопок, навігація посиланнями та інші дії, які користувач ініціює явно або неявно.

Переваги JavaScript:

- Менше взаємодії з сервером - Ви можете перевірити введені користувачем дані перед відправкою сторінки на сервер. Це економить серверний трафік, а це означає менше навантаження на ваш сервер.
- Негайний зворотний зв'язок з відвідувачами - їм не потрібно чекати перезавантаження сторінки, щоб побачити, чи не забули вони щось ввести.
- Підвищена інтерактивність - Ви можете створювати інтерфейси, які реагують, коли користувач наводить на них курсор миші або активує їх за допомогою клавіатури.
- Багатші інтерфейси - Ви можете використовувати JavaScript, щоб включити такі елементи, як компоненти перетягування та повзунки, щоб надати розширений інтерфейс відвідувачам вашого сайту[3].

Обмеження JavaScript:

- Клієнтський JavaScript не дозволяє читати або писати файли. Це було збережено з міркувань безпеки.
- JavaScript не можна використовувати для мережеских програм, оскільки така підтримка відсутня.
- JavaScript не має багатопотокових або багатопроцесорних можливостей

2.4.2 WebSocket & Socket.IO

WebSocket - це незалежний протокол, заснований на протоколі TCP. Це дозволяє тісніше взаємодіяти між браузером та веб-сайтом, допомагаючи розповсюджувати інтерактивний вміст та створювати програми в режимі реального часу[4].

Socket.IO - це бібліотека JavaScript для веб-додатків реального часу. Це забезпечує двосторонній зв'язок у реальному часі між веб-клієнтами та серверами. Він складається з двох частин: бібліотеки на стороні клієнта, яка працює в браузері,

та бібліотеки на стороні сервера для Node.js.

Веб-сокети використовуються в наступних випадках:

- Чати
- Багатокористувацькі ігри
- Спільне редагування
- Сторінки з новинами
- Додатки, які працюють на основі місцезнаходження.

WebSockets

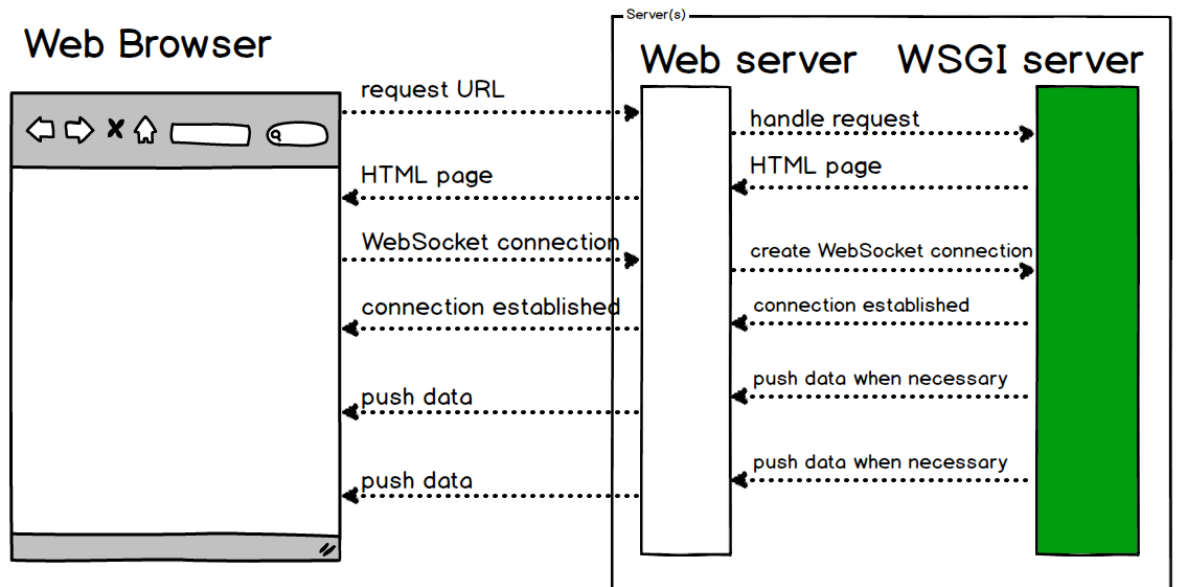


Рисунок 2.5 – Принцип роботи WebSockets

Різниця між веб-сокетами та Socket.IO:

- На відміну від веб-сокетів, Socket.IO дозволяє відправляти повідомлення всім підключеним клієнтам. Наприклад, ви розробляєте чат і хочете мати можливість повідомляти всіх користувачів про підключення нового користувача. Ви легко можете це реалізувати за допомогою однієї операції. При використанні веб-сокетів, для реалізації такого завдання вам буде потрібно список підключених клієнтів і відправлення повідомлень по

одному[5].

- У веб-сокетах складно використовувати проксінг і балансувальник навантаження. Socket.IO підтримує ці технології з коробки.
- Socket.IO підтримує автоматичне перепідключення при розриві з'єднання.
- З Socket.IO легше працювати. Може здатися, що це - кращий засіб для живого спілкування. Однак існує кілька ситуацій, коли краще використовувати веб-сокети. По-перше, вони підтримуються всіма сучасними браузерами, тому ви рідко будете у ситуації коли необхідна підтримка інших технологій, які надаються фреймворком Socket.IO.

2.4.3 Node.JS

Node.JS - це платформа, побудована на виконанні JavaScript скриптів на Chrome для легкої побудови швидких та масштабованих мережевих програм. Node.js використовує керовану подіями неблокувальну модель вводу-виводу, що робить її легкою та ефективною, вона ідеально підходить для додатків у режимі реального часу, що працюють на розподілених пристроях.

Додатки Node.JS написані на JavaScript і можуть запускатися в середовищі виконання Node.JS в OS X, Microsoft Windows та Linux[6].

Node.JS також надає багату бібліотеку різних модулів JavaScript, що значною мірою спрощує розробку веб-додатків за його допомогою.

```
Node.js = Runtime Environment + JavaScript Library
```

Рисунок 2.6 – Принцип роботи Node.js

Особливості Node.js:

- Асинхронні та керовані подіями - Усі API бібліотеки Node.JS є асинхронними, тобто не блокуючими інші дії. По суті, це означає, що сервер на основі Node.JS ніколи не чекає, поки API поверне дані. Сервер переходить до наступного API після його виклику, а механізм сповіщення

про події Node.JS допомагає серверу отримати відповідь від попереднього виклику API.

- Швидке опрацювання - Бібліотека Node.JS, побудована на механізмі JavaScript V8 від Google Chrome, який дозволяє дуже швидко виконувати код.
- Однопоточна, але дуже масштабована - Node.JS використовує однопоточкову модель із циклічною подією. Механізм подій допомагає серверу реагувати неблокуючим способом і робить сервер масштабованим на відміну від традиційних серверів, які створюють обмежені потоки для обробки запитів. Node.JS використовує однопоточкову програму, і ця сама програма може надавати послуги набагато більшій кількості запитів, ніж традиційні сервери, такі як Apache HTTP Server.
- Без буферизації - програми Node.JS ніколи не буферизують будь-які дані. Ці програми просто виводять дані по частинам.
- Ліцензія - Node.JS випускається під ліцензією MIT(ліцензія для розповсюдження вільного програмного забезпечення)

2.4.4 Express Framework

Express - це мінімальна та гнучка структура веб-додатків Node.js, яка забезпечує надійний набір функцій для розробки веб- і мобільних додатків[7]. Це сприяє швидкому розвитку веб-додатків на основі Node. Нижче наведено деякі основні особливості фреймворку Express:

- Дозволяє налаштовувати проміжні програми для відповіді на запити HTTP.
- Визначає таблицю маршрутизації, яка використовується для виконання різних дій на основі методу HTTP та URL-адреси.
- Дозволяє динамічно відображати HTML-сторінки на основі передачі аргументів у шаблони.

2.4.5 Технологія PWA

PWA або Progressive Web Application - технологія, яка дозволяє клієнтам встановити ваш сайт на смартфон як додаток. На українську мову перекладається як «прогресивний веб-додаток». Тепер не потрібно розробляти окремо сайт, додаток під iOS і додаток під Android. Досить мати і підтримувати тільки сайт. Додатки на базі PWA ми використовуємо частіше, ніж здається на перший погляд. Бренди Twitter, Tinder, Uber, Telegram, Starbucks, Forbes, AliExpress, використовують додатки на базі PWA як основні або як допоміжне до мобільного додатку[8].

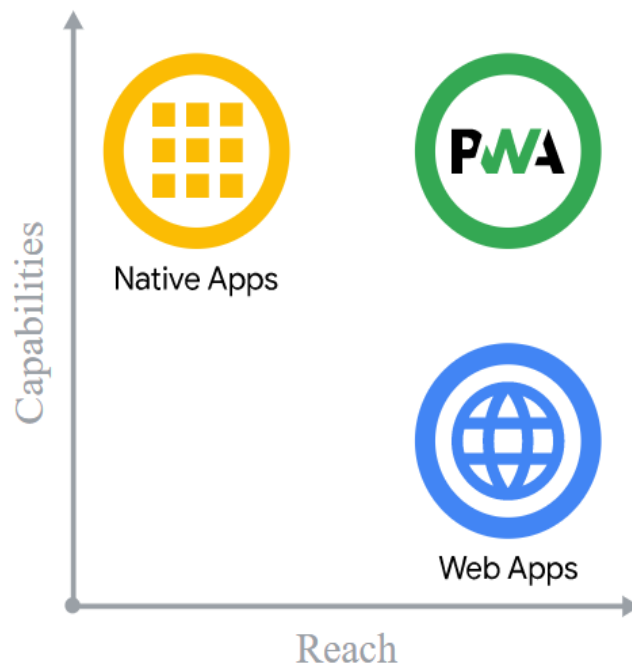


Рисунок 2.7 –PWA technology

Переваги програми на базі PWA:

- Дозволяє відправляти Push-повідомлення.
- Розробка і підтримка на 70% дешевше, ніж у мобільних додатків.
- Економія на повторному притягненні клієнта, так як немає витрат на інструменти повернення клієнта (ретаргетінг, ремаркетинг).
- Не потрібно розміщувати додаток в AppStore і Google Play. Його можна

завантажити безпосередньо з вашого сайту.

- Працює швидше ніж сайт.
- Працює без інтернету.
- Розмір PWA додатки, як правило, не перевищує 1 мб. Це менше, ніж мобільний додаток.
- PWA додаток працює з функціями: геолокація, камера, мікрофон.

Недоліки PWA програми:

- У PWA додатків немає можливості відправляти повідомлення на iPhone, так як в iOS браузер за замовчуванням Safari. Однак в iOS 14 Apple додала функцію вибору браузера за замовчуванням. Це означає, що якщо користувач вибере стандартним браузер не Safari, то повідомлення можна буде відправляти і на iPhone.
- Також PWA програми не підтримують функцію Touch ID і обмежено працюють з функцією Bluetooth[9].

2.5 REPLIT

Replit(рис.2.8) - це стартап із Сан-Франциско та онлайн-середовище IDE (інтегроване середовище розробки). Його назва походить від аббревіатури REPL, що розшифровується як "читання – оцінка – друк циклу".

Replit дозволяє користувачам писати код та створювати програми та веб-сайти за допомогою браузера. Сайт також має різні функції спільної роботи, включаючи можливість багатокористувацького редагування в режимі реального часу за допомогою стрічки чату. Він підтримує понад 50 мов програмування та розмітки, включаючи Java, Python та HTML, що дозволяє користувачам створювати додатки та веб-сайти. Сайт інтегрований з GitHub, платформою хостингу коду, що забезпечує спосіб імпорту та запуску проектів з GitHub. Користувачі також можуть імпортувати проекти від Glitch, який надає подібну послугу до Replit.

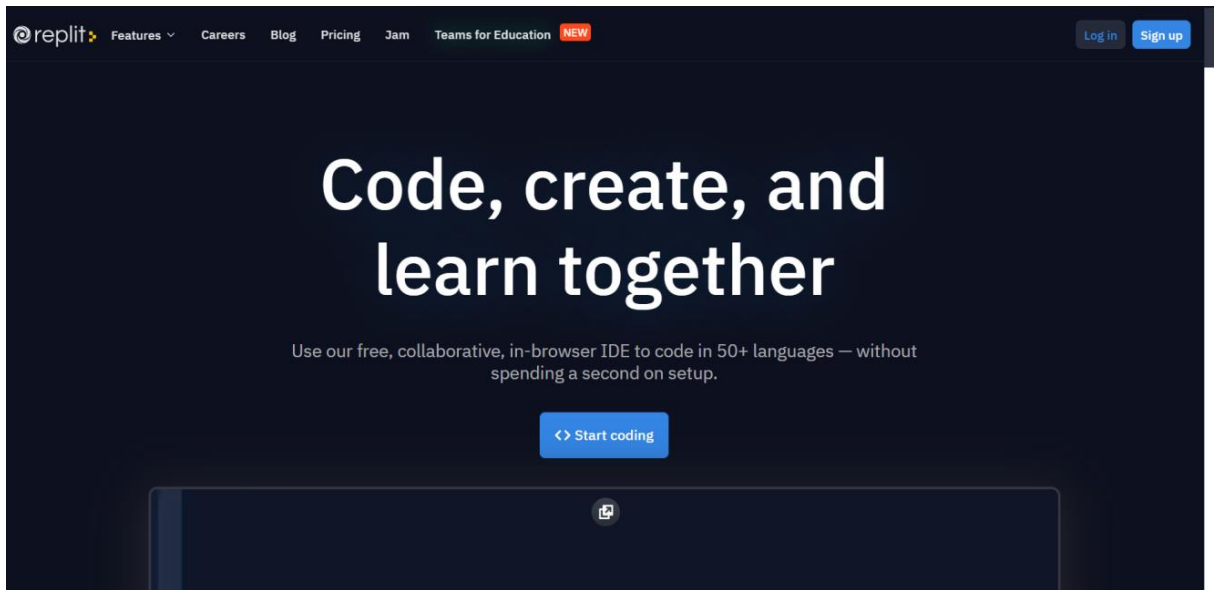


Рисунок 2.8 – Головна сторінка

Платформа надає можливість почати програмувати безпосередньо у вашому браузері(рис.2.9) за лічені секунди, відсікаючи потребу завантажувати архіви та використовувати їх у звичайних середовищах розробки. Користувач може працювати з будь-якого пристрою на якому є браузер та будь-якої ОС – Windows, macOS, Linux. Також, відпадає потреба у налаштуванні свого середовища, тому що у Replit все налаштовується при початку роботи з ним та при створенні проекту[10].

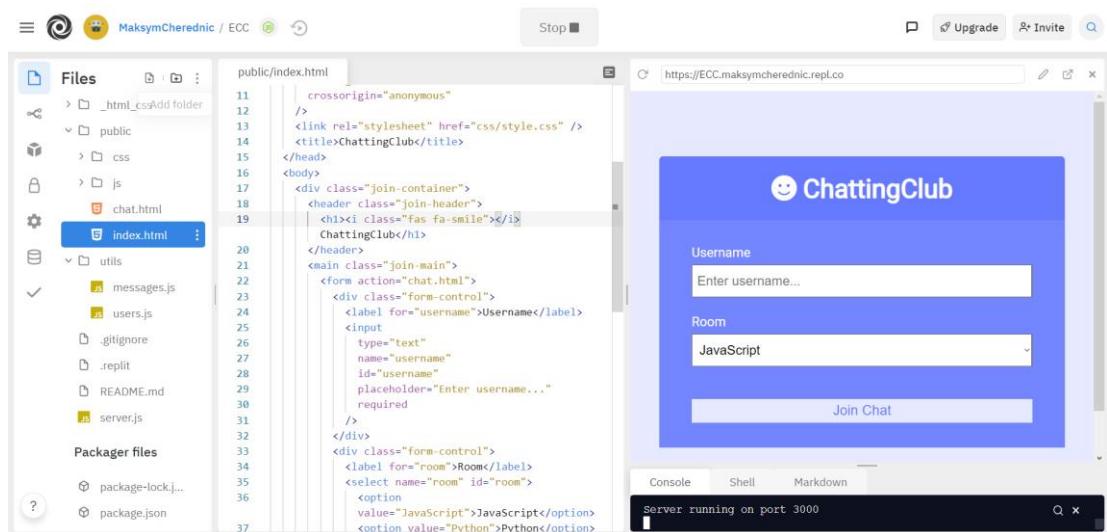


Рисунок 2.9 – Робочі вікна

2.6 Висновок по розділу

Дослідивши існуючі операційні системи та мови програмування, було вирішено зупинитися на мові програмування JS, Node.js та фреймворку Socket.IO.

Також, було вирішено використати знайдене рішення у вигляді онлайн середовища для розробки Repl.it, яке є зручним для роботи з будь-якого девайсу та ОС і має вбудовану інтеграцію з системою контролю версій GitHub, що є дуже зручним при розробці будь-якого додатка, так як після написання коду його автоматично можна відправити у репозиторій.

Дуже важливим є також технологія PWA, яка надає змогу розробити лише 1 додаток, а використовувати його буде можливо як на десктопі, так і на мобільних пристроях. Користувачу буде необхідно мати лише встановлений браузер та зробити кілька кліків, щоб додаток з'явився у нього на пристрої.

Оскільки додаток передбачає певну завантаженість через кількість користувачів, було вирішено скористатися можливостями сервісу Repl.it, а саме віртуальною машиною, на якій додаток зможе працювати без проблем весь час.

Також, можна констатувати той факт, що сучасні додатки краще розробляти кросплатформними, тобто такими, які можливо буде запустити і на смартфоні, і на десктопному пристрої. В цьому лідером є мова програмування JS, яка по своїй суті базується на створенні веб-контенту. Або, якщо йдеться мова про, наприклад, комп'ютерні ігри, то краще і швидше буде використати один з багатьох двигунів, на яких можливо зібрати додаток на будь-яку ОС та будь-який тип пристрою.

3 ДЕМОНСТРАЦІЯ РОБОТИ ЗАСТОСУНКУ

3.1 Реалізація додатка

Після вибору необхідних технологій за допомогою яких буде реалізований додаток, наступним кроком є безпосередньо розробка застосунку.

Як вже було сказано, використовувати будемо мову програмування JavaScript, фреймворк Socket.IO та Node.js. а також мови розмітки HTML та стилів CSS.

Реалізований додаток на даний час має мінімальний обсяг функціоналу, який є основним і має бути присутнім на цьому етапі.

Перша форма, яку бачить користувач при запуску додатка – це головний екран(рис.3)

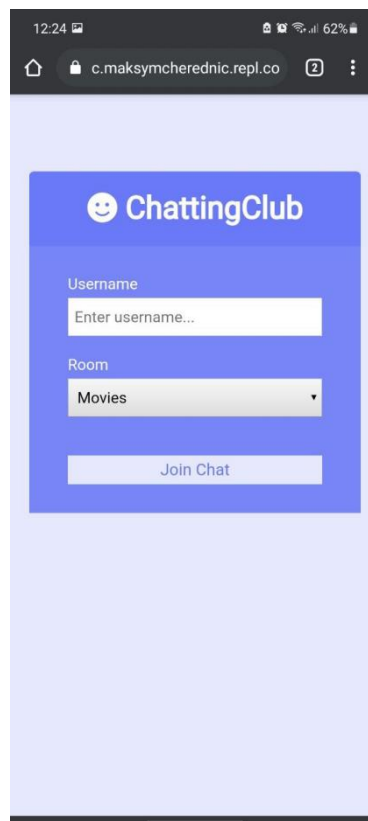


Рисунок 3.1 – Головний екран

На цьому екрані користувач може ввести бажане ім'я для відображення у чаті(рис.3.1) та обрати кімнату за інтересом серед 6 запропонованих варіантів(рис.3.2):

- Movies – Фільми;
- General – Загальний;
- Music – Музика;
- Business – Бізнес;
- Games – Ігри;
- Travel – Подорожі.

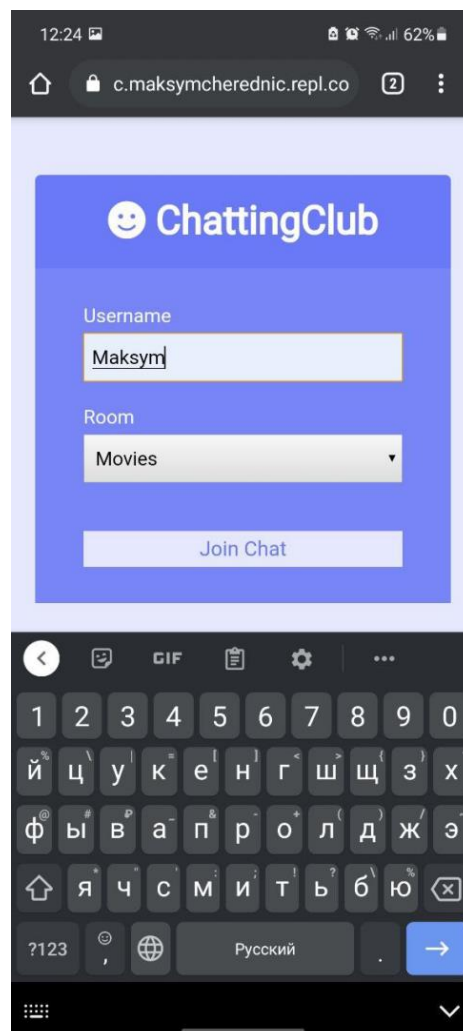


Рисунок 3.3 – Доступні кімнати

Поле введення бажаного імені не має бути пустим, якщо користувач

залишить його пустим то отримає відповідне повідомлення(рис.3.4).

Після того, як користувач вибере собі підходяще ім'я та кімнату до якої він бажає приєднатися відбувається перехід до самого чату, де бот надсилає привітальне повідомлення про приєднання(рис.3.5).

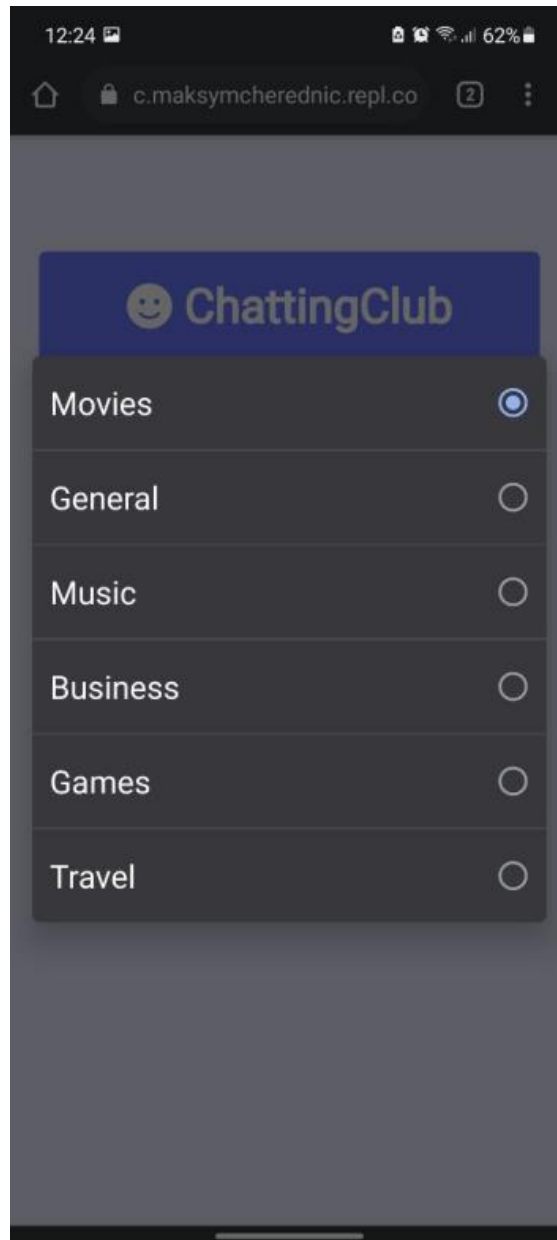


Рисунок 3.2 – Доступні кімнати

Також, бот надасть інформацію користувачу коли до чату приєднається інша людина відповідним повідомленням.



Рисунок 3.4 – Повідомлення про пусте поле

Після відправки користувачем будь-якого повідомлення у чат, воно відобразиться у стрічці повідомлень, підписане ім'ям яке обрав користувач при вході та поруч буде відображений час відправки повідомлення(рис.3.6)

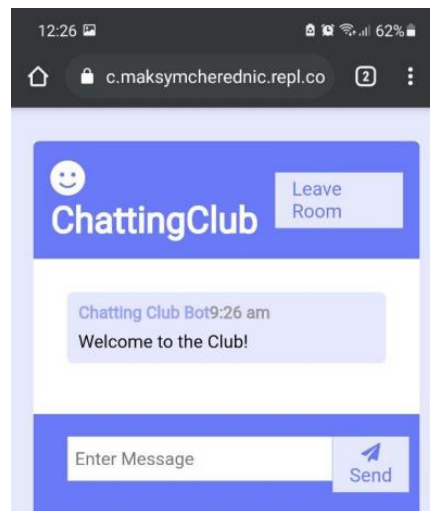


Рисунок 3.5 – Привітальне повідомлення

Через особливості верстки в десктопному додатку користувач додатково може бачити користувачів які зараз приєднані та назву кімнати(рис.3.7), чого немає в мобільному варіанті. Але планується, що такий функціонал з'явиться в ближчих версіях мобільного додатка.

Коли користувач залишає чат, бот також відображає повідомлення про це у чат, вказуючи ім'я користувача та час коли він залишив кімнату(рис.3.8).

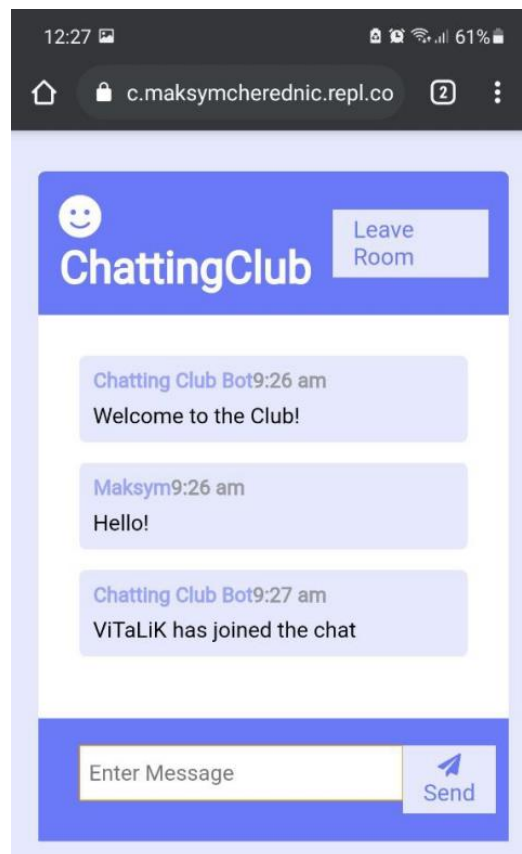


Рисунок 3.6 – Повідомлення про приєднання іншої людини



Рисунок 3.7 – Десктопна версія додатка



Рисунок 3.8 – Повідомлення про вихід людини з кімнати

При виході з кімнати користувача також система питає підтвердження дії(рис.3.9)

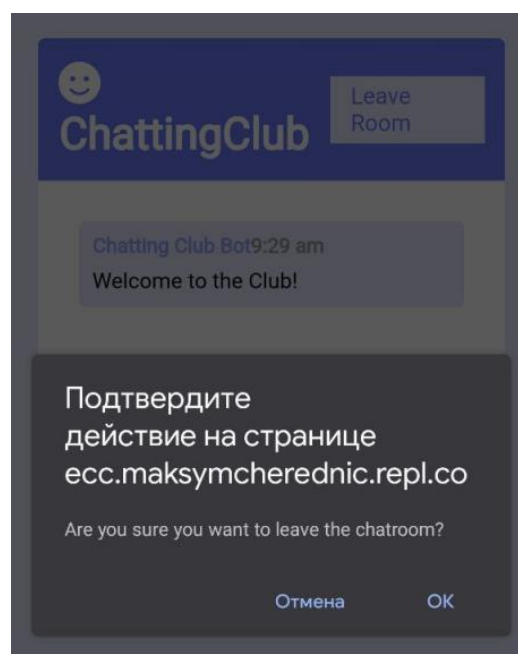


Рисунок 3.9 – Підтвердження дії

Після цього, користувач буде повернений у головне меню додатка, де зможе або вийти цілком, або обрати іншу кімнату серед доступних.

ВИСНОВКИ

Сьогодні на ринку додатків для вивчення англійської мови існує досить багато пропозицій. Але всі вони відрізняються своїм функціоналом, існують додатки з комплексним підходом до вивчення мови, існують додатки з вузьким функціоналом, направленим лише на відточення конкретної навички(аудіювання, граматика або лексика). Деякі з них пропонують складні рішення, інші – навпаки, занадто прості.

У ході виконання роботи було проаналізовано декілька додатків лідерів ринку мовних застосунків для операційної системи Android та виявлено найзатребуваніші для користувача функціональні рішення, а саме:

- функція словника;
- вступне тестування для визначення рівня знань;
- озвучення слів;
- граматичні вправи;
- вивчення нових слів по карткам;
- практика вивченого у діалозі з іншими.

Остання функція була реалізована у додатку “English Chatting Club”.

Дослідивши технології які необхідні для реалізації системи було зроблено висновок про доцільність використання перспективної технології PWA, яка дає можливість використовувати додаток не лише на операційній системі Android, а й на IOS, Windows, та інших, де є встановлений браузер та мови програмування JavaScript у сукупності з фреймворком Socket.IO, адже саме вони дають змогу реалізувати додаток такого типу.

В результаті було розроблено додаток для практики англійської мови «English Chatting Club». Додаток на даний час має лише одну, основну реалізовану функцію – чат для користувачів. Застосунок має простий та зрозумілий інтерфейс.

В майбутньому планується покращити додаток, додавши ще декілька функціональних рішень, а саме:

- вбудований перекладач, завдяки якому користувачі зможуть зробити переклад слова або фрази безпосередньо у чаті;
- темна тема для смартфонів;
- додати до можливостей бота функцію відстеження мови користувача та заборону на використання інших мов, окрім англійської;
- створення особистого кабінету користувача;
- можливість додавати друзів та у подальшому спілкуватися з ними.

Отже, після дослідження теми було виявлено, що ринок освітніх додатків пропонує величезну кількість рішень для вивчення мов або покращення мовних навичок своїх користувачів. Залишається лише знайти підходяще та використовувати за побажаннями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Marijn Haverbeke. Eloquent JavaScript: A Modern Introduction to Programming /- 2017 – с. 194-223.
2. Metanit.com сайт про програмування [Електронний ресурс] – Режим доступу: <https://metanit.com/web/>.
3. Introduction to Node.js [Електронний ресурс] – Режим доступу: <https://nodejs.dev/learn/>.
4. Codecademy Learn Express [Електронний ресурс] – Режим доступу: <https://www.codecademy.com/learn/learn-express/>.
5. Eric Elliott. Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Moderns JS Libraries /- 2019 – с.136-145.
6. Progressive Web App Training [Електронний ресурс] – Режим доступу: <https://developers.google.com/web/ilt/pwa/>.
7. Freecodecamp How to build a progressive web app [Електронний ресурс] – Режим доступу: <https://www.freecodecamp.org/news/build-a-pwa-from-scratch-with-html-css-and-javascript/>.
8. Mike Cantelon, Marc Harter, TJ Holowaychuk, Nathan Rajlich. Node.js in Action /- 2018 – с.121-134
9. CSS tutorial [Електронний ресурс] – Режим доступу: <https://www.w3schools.com/css/default.asp/>.
10. The best cloud IDE [Електронний ресурс] – Режим доступу: <https://www.codeinwp.com/blog/best-cloud-ide/>.
11. Ilya Grigorik. High Performance Browser Networking /- 2015 – с.90-130.
12. Socket.IO tutorial [Електронний ресурс] – Режим доступу: <https://www.tutorialspoint.com/socket.io/index.htm/>.

ДОДАТОК А

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.12.1/css/all.min.css"
      integrity="sha256-mmglkCYLUQbXn0B1SRqzHar6dCnv9oZFPEC1g1cwlkk="
      crossorigin="anonymous"
    />
    <link rel="stylesheet" href="css/style.css" />
    <title>ChattingClub</title>
  </head>
  <body>
    <div class="join-container">
      <header class="join-header">
        <h1><i class="fas fa-smile"></i> ChattingClub</h1>
      </header>
      <main class="join-main">
        <form action="chat.html">
          <div class="form-control">
            <label for="username">Username</label>
            <input
              type="text"
              name="username"
              id="username"
              placeholder="Enter username..."
              required
            />
          </div>
          <div class="form-control">
            <label for="room">Room</label>
            <select name="room" id="room">
              <option value="Movies">Movies</option>
              <option value="General">General</option>
              <option value="Music">Music</option>
              <option value="Business">Business</option>
              <option value="Games">Games</option>
              <option value="Travel">Travel</option>
            </select>
          </div>
        </form>
      </main>
    </div>
  </body>
</html>
```



```
        </select>
    </div>
    <button type="submit" class="btn">Join Chat</button>
</form>
</main>
</div>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.12.1/css/all.min.css"
      integrity="sha256-mmglkCYLUQbXn0B1SRqzHar6dCnv9oZFPEC1g1cwlkk="
      crossorigin="anonymous"
    />
    <link rel="stylesheet" href="css/style.css" />
    <title>ECC</title>
  </head>
  <body>
    <div class="chat-container">
      <header class="chat-header">
        <h1><i class="fas fa-smile"></i> ChattingClub</h1>
        <a id="leave-btn" class="btn">Leave Room</a>
      </header>
      <main class="chat-main">
        <div class="chat-sidebar">
          <h3><i class="fas fa-comments"></i> Room Name:</h3>
          <h2 id="room-name"></h2>
          <h3><i class="fas fa-users"></i> Users</h3>
          <ul id="users"></ul>
        </div>
        <div class="chat-messages"></div>
      </main>
      <div class="chat-form-container">
        <form id="chat-form">
          <input
            id="msg"
            type="text"
            placeholder="Enter Message"
          />
        </form>
      </div>
    </div>
```

```
        required
        autocomplete="off"
    />
    <button class="btn"><i class="fas fa-paper-plane"></i> Send</button>
</form>
</div>
</div>

<script src="https://cdnjs.cloudflare.com/ajax/libs/qs/6.9.2/qs.min.js"
integrity="sha256-TDxXjkAUay70ae/QJBEPGKkpVslXaHHayklIVglFRT4="
crossorigin="anonymous"
></script>
<script src="/socket.io/socket.io.js"></script>
<script src="js/main.js"></script>
</body>
</html>
```

```
@import url('https://fonts.googleapis.com/css?family=Roboto&display=swap');
```

```
:root {
  --dark-color-a: #667aff;
  --dark-color-b: #7386ff;
  --light-color: #e6e9ff;
  --success-color: #5cb85c;
  --error-color: #d9534f;
}

* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

body {
  font-family: 'Roboto', sans-serif;
  font-size: 16px;
  background: var(--light-color);
  margin: 20px;
}

ul {
  list-style: none;
}
```

```
a {
  text-decoration: none;
}

.btn {
  cursor: pointer;
  padding: 5px 15px;
  background: var(--light-color);
  color: var(--dark-color-a);
  border: 0;
  font-size: 17px;
}

/* Chat Page */

.chat-container {
  max-width: 1100px;
  background: #fff;
  margin: 30px auto;
  overflow: hidden;
}

.chat-header {
  background: var(--dark-color-a);
  color: #fff;
  border-top-left-radius: 5px;
  border-top-right-radius: 5px;
  padding: 15px;
  display: flex;
  align-items: center;
  justify-content: space-between;
}

.chat-main {
  display: grid;
  grid-template-columns: 1fr 3fr;
}

.chat-sidebar {
  background: var(--dark-color-b);
  color: #fff;
  padding: 20px 20px 60px;
  overflow-y: scroll;
}
```

```
.chat-sidebar h2 {
  font-size: 20px;
  background: rgba(0, 0, 0, 0.1);
  padding: 10px;
  margin-bottom: 20px;
}

.chat-sidebar h3 {
  margin-bottom: 15px;
}

.chat-sidebar ul li {
  padding: 10px 0;
}

.chat-messages {
  padding: 30px;
  max-height: 500px;
  overflow-y: scroll;
}

.chat-messages .message {
  padding: 10px;
  margin-bottom: 15px;
  background-color: var(--light-color);
  border-radius: 5px;
  overflow-wrap: break-word;
}

.chat-messages .message .meta {
  font-size: 15px;
  font-weight: bold;
  color: var(--dark-color-b);
  opacity: 0.7;
  margin-bottom: 7px;
}

.chat-messages .message .meta span {
  color: #777;
}

.chat-form-container {
  padding: 20px 30px;
```

```
background-color: var(--dark-color-a);
}
```

```
.chat-form-container form {
  display: flex;
}
```

```
.chat-form-container input[type='text'] {
  font-size: 16px;
  padding: 5px;
  height: 40px;
  flex: 1;
}
```

```
/* Join Page */
```

```
.join-container {
  max-width: 500px;
  margin: 80px auto;
  color: #fff;
}
```

```
.join-header {
  text-align: center;
  padding: 20px;
  background: var(--dark-color-a);
  border-top-left-radius: 5px;
  border-top-right-radius: 5px;
}
```

```
.join-main {
  padding: 30px 40px;
  background: var(--dark-color-b);
}
```

```
.join-main p {
  margin-bottom: 20px;
}
```

```
.join-main .form-control {
  margin-bottom: 20px;
}
```

```
.join-main label {
  display: block;
```

```
margin-bottom: 5px;
}
```

```
.join-main input[type='text'] {
font-size: 16px;
padding: 5px;
height: 40px;
width: 100%;
}
```

```
.join-main select {
font-size: 16px;
padding: 5px;
height: 40px;
width: 100%;
}
```

```
.join-main .btn {
margin-top: 20px;
width: 100%;
}
```

```
@media (max-width: 700px) {
.chat-main {
display: block;
}
}
```

```
.chat-sidebar {
display: none;
}
}
```

ДОДАТОК Б

```
const chatForm = document.getElementById('chat-form');
const chatMessages = document.querySelector('.chat-messages');
const roomName = document.getElementById('room-name');
const userList = document.getElementById('users');
```

```
const { username, room } = Qs.parse(location.search, {
  ignoreQueryPrefix: true,
});
```

```
const socket = io();
```

```
socket.emit('joinRoom', { username, room });
```

```
socket.on('roomUsers', ({ room, users }) => {
  outputRoomName(room);
  outputUsers(users);
});
```

```
socket.on('message', (message) => {
  console.log(message);
  outputMessage(message);
```

```
  chatMessages.scrollTop = chatMessages.scrollHeight;
});
```

```
chatForm.addEventListener('submit', (e) => {
  e.preventDefault();
```

```
  let msg = e.target.elements.msg.value;
```

```
  msg = msg.trim();
```

```
  if (!msg) {
    return false;
  }
```

```
  socket.emit('chatMessage', msg);
```

```

    e.target.elements.msg.value = "";
    e.target.elements.msg.focus();
  });
function outputMessage(message) {
  const div = document.createElement('div');
  div.classList.add('message');
  const p = document.createElement('p');
  p.classList.add('meta');
  p.innerText = message.username;
  p.innerHTML += `<span>${message.time}</span>`;
  div.appendChild(p);
  const para = document.createElement('p');
  para.classList.add('text');
  para.innerText = message.text;
  div.appendChild(para);
  document.querySelector('.chat-messages').appendChild(div);
}

function outputRoomName(room) {
  roomName.innerText = room;
}

function outputUsers(users) {
  userList.innerHTML = "";
  users.forEach((user) => {
    const li = document.createElement('li');
    li.innerText = user.username;
    userList.appendChild(li);
  });
}

document.getElementById('leave-btn').addEventListener('click', () => {
  const leaveRoom = confirm('Are you sure you want to leave the chatroom?');
  if (leaveRoom) {
    window.location = '../index.html';
  } else {
  }
});

```


ДОДАТОК В

```
const path = require('path');
const http = require('http');
const express = require('express');
const socketio = require('socket.io');
const formatMessage = require('./utils/messages');
const {
  userJoin,
  getCurrentUser,
  userLeave,
  getRoomUsers
} = require('./utils/users');

const app = express();
const server = http.createServer(app);
const io = socketio(server);

app.use(express.static(path.join(__dirname, 'public')));

const botName = 'Chatting Club Bot';

io.on('connection', socket => {
  socket.on('joinRoom', ({ username, room }) => {
    const user = userJoin(socket.id, username, room);

    socket.join(user.room);

    socket.emit('message', formatMessage(botName, 'Welcome to the Club!'));

    socket.broadcast
      .to(user.room)
      .emit(
        'message',
        formatMessage(botName, `${user.username} has joined the chat`)
      );

    io.to(user.room).emit('roomUsers', {
      room: user.room,
      users: getRoomUsers(user.room)
    });
  });
});
```

```

    });
  });

  socket.on('chatMessage', msg => {
    const user = getCurrentUser(socket.id);

    io.to(user.room).emit('message', formatMessage(user.username, msg));
  });

  socket.on('disconnect', () => {
    const user = userLeave(socket.id);

    if (user) {
      io.to(user.room).emit(
        'message',
        formatMessage(botName, `${user.username} has left the chat`)
      );

      io.to(user.room).emit('roomUsers', {
        room: user.room,
        users: getRoomUsers(user.room)
      });
    }
  });
});

const PORT = process.env.PORT || 3000;

server.listen(PORT, () => console.log(`Server running on port ${PORT}`));

const moment = require('moment');

function formatMessage(username, text) {
  return {
    username,
    text,
    time: moment().format('h:mm a')
  };
}

module.exports = formatMessage;

```

```
const users = [];  
  
function userJoin(id, username, room) {  
  const user = { id, username, room };  
  
  users.push(user);  
  
  return user;  
}  
  
function getCurrentUser(id) {  
  return users.find(user => user.id === id);  
}  
  
function userLeave(id) {  
  const index = users.findIndex(user => user.id === id);  
  
  if (index !== -1) {  
    return users.splice(index, 1)[0];  
  }  
}  
  
function getRoomUsers(room) {  
  return users.filter(user => user.room === room);  
}  
  
module.exports = {  
  userJoin,  
  getCurrentUser,  
  userLeave,  
  getRoomUsers  
};
```

ДОДАТОК Г



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ВИВЧЕННЯ
АНГЛІЙСЬКОЇ МОВИ «ENGLISH CHATTING CLUB» ЗА
ДОПОМОГОЮ МОВИ ПРОГРАМУВАННЯ JAVASCRIPT

Виконав студент 4 курсу
групи ПД-41
Чередніченко М.Д.
Керівник роботи
Дібрівний О.А.

Київ – 2021

Мета, об'єкт та предмет дослідження

- Мета роботи - розробити додаток для практики англійської мови
- Об'єкт дослідження - додатки для вивчення або практики англійської мови
- Предмет дослідження - функціонал додатків для вивчення або практики англійської мови

АНАЛОГИ



Порівняння

	Lingualeo	Duolingo	Memrise	ReWord	Cake
Лексика	Green	Green	Green	Green	Green
Граматика	Green	Red	Red	Red	Red
Аудіювання	Green	Green	Green	Red	Green
Практика вивченого	Red	Red	Red	Red	Red
Озвучення	Green	Green	Green	Green	Green
Наявність словника	Green	Red	Red	Green	Green
Наявність початкового тестування	Green	Red	Red	Red	Red

Технічне завдання

1. Розробити концепт додатка для практики англійської мови;
2. Використати актуальний стек технологій
3. Забезпечити додаток наступним функціоналом:
 - поле вибору кімнати серед доступних;
 - поле введення бажаного імені для відображення у чаті;
 - відображення імені інших користувачів;
 - кросплатформеність.

Програмні засоби реалізації



Replit - хмарне середовище розробки. Replit дозволяє користувачам писати код та створювати програми та веб-сайти за допомогою браузера. Сайт також має різні функції спільної роботи, включаючи можливість багатокористувацького редагування в режимі реального часу. Він підтримує понад 50 мов програмування та розмітки, включаючи Java, Python, JavaScript та HTML.

Фреймворк Socket.IO



Socket.IO - це бібліотека JavaScript для веб-додатків реального часу. Це забезпечує двосторонній зв'язок у реальному часі між веб-клієнтами та серверами. Він складається з двох частин: бібліотеки на стороні клієнта, яка працює в браузері, та бібліотеки на стороні сервера для Node.js. Обидва компоненти мають майже однаковий API. Як і Node.js, він керується подіями.

Висновки

Найвикористовуванішими функціями додатків для вивчення або практики англійської мови є:

- тестування рівня знань;
- озвучення слів;
- перегляд відео записаними носіями мови;
- спілкування з іншими людьми.

Наступними етапами роботи над додатком є:

- додати темну тему, адже це є дуже затребуваним серед сучасних користувачів;
- додати перекладач, аби новачки могли спілкуватися з більш досвідченими користувачами на одному рівні;
- адаптувати десктоп версію під мобільні девайси;