

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____Ю. П. Кондратенко
«__» _____ 2023 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

ВЕБЗАСТОСУНОК ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 402.22130201

Виконав студент 4-го курсу, групи 402
_____ *С. Ю. Данкович*
«__» червня 2023 р.

Керівник: канд. техн. наук, доцент
_____ *Є. В. Сіденко*
«__» червня 2023 р.

Миколаїв – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
« ____ » _____ 20__ р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

Видано студенту групи 402 факультету комп'ютерних наук Данковичу Сергію Юрійовичу.

1. Тема кваліфікаційної роботи «Вебзастосунок для вивчення іноземних мов».

Керівник роботи Сіденко Євген Вікторович, канд. техн. наук, доцент.

Затв. наказом Ректора ЧНУ ім. Петра Могили від « ____ » _____ 20__ р. № _____

2. Строк представлення кваліфікаційної роботи студентом « ____ » _____ 20__ р.

3. Вхідні (початкові) дані до роботи: опис електронного навчання, технологій та методологій.

Очікуваний результат: Вебзастосунок для вивчення іноземних мов з використанням технології комунікації в реальному часі для створення відеоконференцій та допоміжного інструментарію для кооперативного та індивідуального вивчення мов.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

– аналіз сучасного стану задачі, щодо розробки вебзастосунку для вивчення іноземних мов;

– огляд існуючих методів для вирішення задачі створення групових відео дзвінків;

– моделювання та проектування інформаційної системи для вебзастосунку;

– опис результатів реалізації вебзастосунку та тестування.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: «Аналіз освітлення на робочу місці»

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Боженко А. Л., викладач кафедри екології	

Керівник роботи канд. техн. наук, доцент Сіденко Є. В.

(наук. ступінь, вчене звання, прізвище та ініціали)

(підпис)

Завдання прийнято до виконання Данкович С. Ю.

(прізвище та ініціали)

(підпис)

Дата видачі завдання «23» _____ листопада _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: «Вебзастосунок для вивчення іноземних мов»

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників БКР	29.10.2022	30.10.2022	
2	Отримання завдання на виконання БКР	11.11.2022	11.11.2022	
3	Складання календарного плану роботи на весь період виконання БКР	01.12.2022	06.12.2022	
4	Отримання завдання на переддипломну практику	27.04.2023	27.04.2023	
5	Проходження переддипломної практики, збір та аналіз матеріалів до БКР	01.05.2023	14.05.2022	
6	Розробка звіту з переддипломної практики	15.05.2023	17.05.2023	
7	Виконання БКР: аналіз сучасного стану задачі створення вебзастосунку для вивчення іноземних мов, дослідження методів вирішення задачі, моделювання інформаційної системи, програмна розробка та тестування вебзастосунку.	15.05.2023	19.06.2023	
8	Попередній захист БКР на засіданні комісії кафедри	29.05.2023	30.05.2023	
9	Доробка та остаточне оформлення БКР	02.06.2023	19.06.2023	
10	Подання БКР рецензенту	15.06.2023	17.06.2023	
11	Подання БКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	19.06.2023	22.06.2023	
12	Захист БКР перед екзаменаційною комісією (ЕК)	26.06.2023	29.06.2023	

Розробив студент Данкович С. Ю.

(прізвище та ініціали)

(підпис)

Керівник роботи канд. техн. наук, доцент Сіденко Є. В.

(наук. ступінь, вчене звання, прізвище та ініціали)

(підпис)

« 07 » 12 2022 р.

АНОТАЦІЯ

бакалаврської кваліфікаційної роботи студента групи 402 ЧНУ ім. Петра

Могили

Данковича Сергія Юрійовича

Тема: «Вебзастосунок для вивчення іноземних мов»

Актуальність зумовлена зростаючим попитом на застосунки для вивчення іноземних мов, що підкреслюються глобальним зростанням ринку онлайн-вивчення мов.

Об'єктом роботи – процеси комунікації та навчання у реальному часі.

Предметом роботи – програмні засоби та технології для вивчення іноземних мов.

Метою бакалаврської кваліфікаційної роботи є підвищення рівня комунікації та вивчення іноземної мови з використанням технології відеоконференції.

Пояснювальна записка складається зі вступу, чотирьох розділів, висновків та додатків.

У першому розділі розглядається сучасний стан задачі вивчення іноземних мов та ставляться задачі на виконання.

У другому розділі досліджено методи та технології за якими будуть проектуватися інформаційні системи.

У третьому розділі описані головні процеси в вебзастосунку за допомогою інформаційних систем.

У четвертому розділі наведено результати розробки та огляд ефективності вебзастосунку.

В результаті розроблено вебзастосунок з використанням технології відеоконференції, що забезпечує комунікацію у реальному часі та надає ряд інструментів для кооперативного або індивідуального вивчення іноземних мов.

Бакалаврська кваліфікаційна робота містить __ сторінок, __ рисунків, __ таблиць, __ використаних джерел та __ додатків.

Ключові слова: вебзастосунок, WebRTC, електронне навчання.

ABSTRACT

**bachelor's qualification work of a student of 402 group at Black Sea National
University named after Petro Mohyla**

Serhii Dankovych

Topic: «Web application for learning foreign languages»

The relevance is due to the growing demand for applications for learning foreign languages, highlighted by the global growth of the online language learning market.

The object of the work is communication and learning processes in real time.

The subject of the work is software and technologies for learning foreign languages.

The purpose of the bachelor's qualification work is to improve the level of communication and learning a foreign language using video conference technology.

The explanatory note consists of an introduction, four chapters, conclusions and appendices.

The first chapter examines the current state of the task of learning foreign languages and sets the tasks to be performed.

In the second chapter, the methods and technologies by which information systems are built are studied.

The third section describes the main processes in the web application using information systems.

The fourth chapter presents the results of development and an overview of the effectiveness of the web application.

As a result, a web application was developed using video conferencing technology, which provides real-time communication and provides a number of tools for cooperative or individual study of foreign languages.

Bachelor's qualification work contains __ pages, __ figures, __ tables, __ used sources and __ appendices.

Keywords: web application, WebRTC, e-learning.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 АНАЛІЗ СУЧАСНОГО СТАНУ ЗАДАЧІ СТВОРЕННЯ ВЕБЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ.....	6
1.1 Опис предметної області.....	6
1.2 Огляд та аналіз наявних аналогів	8
1.3 Огляд та аналіз наявних публікацій	11
1.4 Огляд існуючих технологій для вирішення поставленої задачі	12
1.5 Постановка задачі.....	14
Висновки до розділу 1	16
2 МЕТОДИ ТА ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ СТВОРЕННЯ ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ.....	17
2.1 Використання Socket.IO з технологією WebRTC.....	17
2.1 Методи для вирішення задачі створення відеоконференції	18
2.2 Вибір топології з'єднання за допомогою методу аналізу ієрархій.....	20
Висновки до розділу 2	27
3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	28
3.1 Проєктування інформаційної системи	28
Висновки до розділу 3	32
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ.....	33
4.1 Опис програмної реалізації.....	33
4.2 Огляд ефективності.....	46
Висновки до розділу 4	50
ВИСНОВКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	52
ДОДАТОК А Лістинг програмного коду.....	55

ПЕРЕЛІК СКОРОЧЕНЬ

API – Application Programming Interface

HTTP – Hypertext Transfer Protocol

JWT – JSON Web Token

TCP – Transmission Control Protocol

WebRTC – Web Real-Time Communication

ВСТУП

За останні декілька років з'явився високий попит на електронне навчання при вивченні іноземних мов. Вивчення іноземної мови дає можливість спілкуватися з людьми з різною культурою та походженням. Має великий вплив на кар'єру і тому роботодавці переважно шукають тих кандидатів, які володіють кількома мовами, особливо якщо вони працюють у сферах, де треба взаємодіяти на міжнародному рівні. Також, було досліджено, що вивчення мов може значно покращити стан пам'яті та головного мозку, що впливає на вміння розв'язувати проблеми та на відтермінування деменції. Знання іноземної мови на достатньому рівні вносить вагомий внесок для покращення досвіду під час подорожі.

Слід зазначити, що розмова є одним з основних способів спілкування та взаємодії. Нині, розвиток розмовних навичок є найважливішим напрямком у вивченні іноземної мови для ефективного вираження ідей, думок та емоцій.

При вивченні іноземних мов за допомогою електронного навчання існує багато переваг, що полягають у зручності, інтерактивності та можливості для спілкування та співпраці [1].

Використання технологій є невіддільним процесом у вивченні іноземних мов у всьому світі. Великий внесок в електронне навчання здійснено за допомогою технології відеоконференції, що дозволяє користувачам здійснювати комунікацію у реальному часі та візуально сприймати інформацію та взаємодіяти з іншими користувачами.

Людина сприймає, приблизно, 80% інформації візуально, тому технологія відеоконференції є найважливішим джерелом набуття знань при вивченні іноземних мов. За допомогою відеоконференцій можна здобути досвід вивчення мови, практикуючи спілкування у віртуальній кімнати. Учасникам відеоконференції додатково надається можливість спостерігати за мовою тіла, виразом обличчя та іншими невербальними сигналами, що позитивно впливає на результати навчання та підкріплює розмовні навички.

Відеоконференція є формою синхронного навчання, що здійснюється за допомогою синхронних інструментів, що більше схоже на особисту взаємодію між людьми. Може використовувати текстові повідомлення, відео та додаткові інструменти для вивчення мов [2].

Завдяки використанню застосунку для відеоконференцій, користувачі можуть відточувати свої розмовні здібності під час вивчення мов в кооперативному або індивідуальному режимі. Тобто, це підтверджує ідею про те, що відеоконференції можуть допомогти користувачам покращити свої розмовні здібності та тому, вкрай важливо, щоб було створено більше застосунків для спілкування іноземними мовами.

Вебзастосунок повинен бути спрямований на розв'язання проблем, з якими стикаються при вивченні іноземних мов, особливо з точки зору розмовних навичок, якими часто нехтують у традиційному навчанні.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ЗАДАЧІ СТВОРЕННЯ ВЕБЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ

1.1 Опис предметної області

Електронне навчання, що також називають онлайн або дистанційним навчанням. Це метод навчання, який використовує комп'ютерну мережу та цифрові технології для надання доступу до навчального контенту і створений для полегшення взаємодії між студентами та викладачами. Електронне навчання стає все більш актуальним в останні роки, особливо зі стрімким прогресом в цифрових технологіях та в необхідності в дистанційному навчанні.

Одна з головних переваг електронного навчання полягає в її гнучкості. Завдяки електронному навчанню, учні можуть взаємодіяти один з одним в будь-який час та з будь-якого місця, за умови якщо вони підключенні до мережі інтернет [3]. Ще однією важливою перевагою електронного навчання є його економічна ефективність. Порівняно з традиційним навчанням, електронне навчання є більш економічно ефективним, оскільки не потребує надлишкових витрат на фізичні аудиторії, підручники чи інші навчальні матеріали. Впровадження електронного навчання особливо корисно для організацій, яким потрібно навчати велику кількість людей, оскільки електронному навчанню властиво масштабування щоб вмістити велику кількість користувачів. Такий вид навчання можна персоналізувати відповідно до потреб окремих учнів які можуть вирішувати, що вони хочуть саме вивчати, визначити темп навчання та визначитися з ресурсами, які вони використовують. Таким чином, це дає можливість учням зосередитися на своїх інтересах або сферах, які потребують розвитку.

Вагомий внесок в електронне навчання зроблено за допомогою технології відеоконференції, головною метою якого створення віртуального простору, використовуючи комунікацію у реальному часі, що дає можливість учасникам

спілкуватися один з одним, ставити запитання, обговорювати навчальні матеріали та швидко вирішувати проблеми, що виникають під час зустрічі [4].

Відеоконференція – це незалежний від місцеперебування сеанс зв'язку між двома користувачами або групою користувачів. Кількість учасників на екрані безпосередньо залежить від певного режиму конференції та ролі користувача в поточній конференції. Існує чотири типи відеоконференцій.

Відеодзвінок – це тип відеоконференції, коли учасники спілкуються у реальному часі віч-на-віч за допомогою відео та аудіо. Цей тип підходить для приватних розмов, інтерв'ю та зустрічей на дистанційній роботі, коли потрібно спілкуватися лише двом людям.

Симетрична відеоконференція – всі учасники відображаються на екрані одночасно, що дозволяє всім бачити та чути один одного. Даний тип використовується для групових зустрічей, онлайн-класів, віртуальних заходів та інших ситуацій, коли кілька учасників повинні взаємодіяти один з одним.

Відеолекція – схожа на режим симетричної конференції, але основна увага приділяється тільки одному користувачу, що виступає з певною темою чи задачею. Доповідач є основним спікером, і всі учасники можуть бачити й чути його та додатково для учасників надається можливість задати питання, що робить процес навчання більш інтерактивним.

Конференція – підходить для одного або декількох доповідачів, які проводять презентацію або організують нараду. У цьому режимі доповідач з'являється на екрані, його можуть бачити та чути всі учасники. Однак доповідачі не бачать і не чують учасників, що робить цей режим більш формальним і структурованим методом спілкування.

Отже, спілкування в реальному часі має значний вплив на навчання, сприяючи персоналізованому досвіду навчання, моделюванню процесу спілкування в віртуальному середовищі, полегшуючи доступ до експертів та ресурсів, заохочуючи на активне залучення та мотивацію, створюючи нові можливості для синхронного навчання.

1.2 Огляд та аналіз наявних аналогів

Zoom – це застосунок для відеоконференцій, що набув популярності під час пандемії, коли людям доводилося працювати або вчитись в віддалених місцях.

Застосунок має зручний інтерфейс, у якому легко орієнтуватися, що дає можливість використати його широким колом користувачів (рис. 1.1). Користувачі можуть ділитися своїм екраном з іншими учасниками відеоконференції, що зручно під час представлення матеріалів доповіді. Наявний функціонал зміни фону, що корисно для того щоб сфокусувати увагу на доповідачі. У застосунку також є можливість записувати зустріч, щоб переглянути та поділитися записом зустрічі. Що стосується безпеки, Zoom використовує додаткові підходи для захисту користувацьких даних від несанкціонованого доступу, як захист паролем або кімнати очікування.

Проаналізувавши даний застосунок можна дійти до висновку, що застосунок Zoom направлений для соціальних дзвінків або для ділових зустрічей та не має функціоналу для вивчення іноземних мов, наприклад, з можливістю включення певних інструментів для кооперативного або індивідуального вивчення мов з оцінюванням результатів, що може допомогти учням відстежувати свій прогрес і визначати сфери, які потрібно вдосконалити.

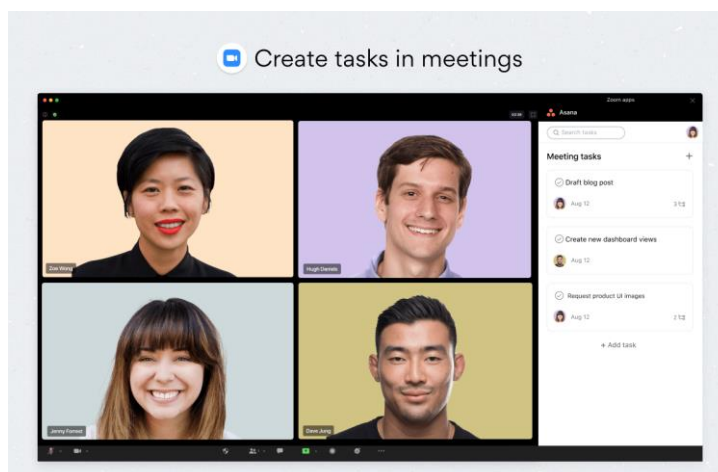


Рисунок 1.1 – Застосунок Zoom

Discord – застосунок для спілкування, створений переважно для геймерів. Спілкування відбувається на створених серверів, на яких може під'єднатися кожен користувач, що має певний доступ до серверу (рис. 1.2). Сервер може мати кілька каналів для різних цілей, наприклад голосові канали для розмов у реальному часі та текстові канали для обміну повідомленнями. Застосунок підтримує високоякісні голосові та відео дзвінки з низькою затримкою. Надає можливість здійснювати приватні дзвінки, групові дзвінки або приєднуватися до голосових каналів. Наявний широкий спектр ботів, яких можна додавати на сервери для виконання різноманітних завдань. Discord дозволяє користувачам створювати та використовувати в повідомленнях власні емодзі в чатах. Застосунок інтегрується з іншими програмами, такими як:

- Spotify;
- Twitch;
- YouTube.

Проаналізувавши даний застосунок можна зробити висновки, що Discord має ряд внутрішніх інструментів які можуть бути корисними для вивчення іноземних мов, але дані застосунки не спрямовані саме на вивчення мов та не мають безкоштовного доступу. В вебзастосунку для вивчення іноземних мов планується інтегрувати більш ширший спектр інтерактивних інструментів, вправ та тестів, що можуть допомогти користувачам відпрацювати свої навички та отримати негайний зворотний зв'язок.

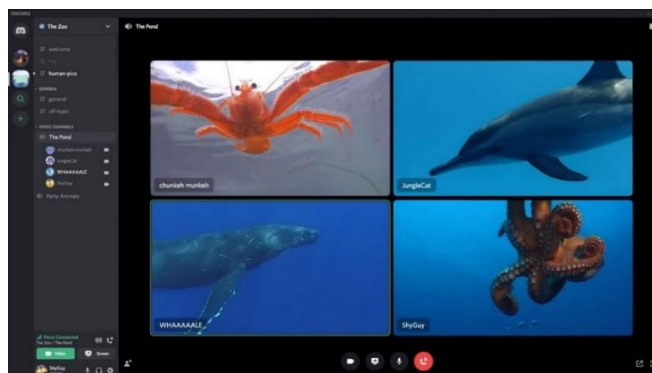


Рисунок 1.2 – Застосунок Discord

Google Meet – це платформа для відеоконференцій. Спочатку він був розроблений для корпоративних користувачів, але пізніше став доступним для всіх власників облікових записів Google, включаючи окремих осіб і навчальні заклади.

Google Meet пропонує високоякісні відеодзвінки та відеоконференції, у яких можуть брати участь до 100 учасників. Даний застосунок підтримує спільний доступ до екрана, що корисно для віддаленої співпраці та презентацій. Google Meet має свою екосистему та інтегрується з Google Calendar, що полегшує планування зустрічей і надсилання запрошень та з іншими програмами Google для досягнення легкої співпраці. Застосунок пропонує живі субтитри для учасників, що мають проблеми з розумінням звуку. Він дозволяє організаторам записувати зустрічі для використання в майбутньому або ділитися з тими, хто не зміг бути присутнім. Google Meet має кілька функцій безпеки, як шифрування та приватні зустрічі, для захисту від небажаних учасників.

Хоча Google Meet має дуже простий та зарозумілий інтерфейс (рис. 1.3), але він не направлений на вивчення іноземних мов, в ньому не вистачає певних особливостей для вивчення мов, які дозволяють користувачам безперешкодно спілкуватися та взаємодіяти, також відсутня можливість користуватися словниками, або перекладачем.



Рисунок 1.3 – Застосунок Google Meet

1.3 Огляд та аналіз наявних публікацій

У публікації [5] автор Li-Tang Yu представляє дослідження, яке вивчає вплив відеоконференцій на вивчення іноземної мови. Результати дослідження свідчать про те, що відеоконференції можуть мати позитивний вплив на вивчення другої мови, зокрема на покращення навичок говоріння та аудіювання.

Публікація [6] – це дослідження автора S. ÖZTÜRK, яке вивчає функціональність, продуктивність і зручність використання продуктів для відеоконференцій на основі технології WebRTC (Web Real-Time Communication). Було проведено серію тестів, щоб оцінити ефективність технології у різних сценаріях.

Автори Dongming Tang і Liqun Zhang пропонують у публікації [7] новий метод змішування аудіо та відео для покращення продуктивності та надійності технології WebRTC. Визначено кілька обмежень існуючих методів мікшування аудіо та відео, які використовуються в WebRTC, включаючи високу затримку, низьку якість і великі обчислювальні витрати.

Розглянуто в публікації [8] авторами якої є Cristian Cola та Honoriu Valean впровадження технології WebRTC для багатокористувацьких конференцій. Пропоноване рішення передбачає використання централізованого сервера для керування зв'язком між кількома клієнтами. Автори також пропонують використовувати медіа сервер для обробки відеоданих, щоб зменшити навантаження на клієнтські пристрої. Пропоноване рішення оцінюється за допомогою серії експериментів, які демонструють ефективність запропонованого підходу.

Публікація [9] авторів Riza Arda Kirmiziloglu та A. Murat Tekalp пропонує новий підхід для надання багатосторонньої послуги в мережах 5G з використанням багатоадресної IP-адреси за допомогою Software Defined Networking (SDN). Автори представляють концепцію SDN і її потенціал для оптимізації мережевого трафіку в мережах 5G та пропонують метод багатоадресної IP-адреси з урахуванням

затримки та пропускну здатності за допомогою SDN, який використовує масштабване кодування відео для зменшення вимог до пропускну здатності.

У публікації [10] авторів George Suciuc, Stefan Stefanescu, Cristian Beceanu, та Marian Ceararu розглянуто комплексний аналіз ролі технології WebRTC у спілкуванні в реальному часі та відеоконференціях. Запропонували рішення для вирішення проблем впровадження систем на основі WebRTC, що забезпечують цінну інформацію для розробників і дослідників, які працюють у цій галузі.

У публікації [11] автори Boni García, Micael Gallego, Francisco Gortázar, та Antonia Bertolino розглядають питання вимірювання Quality of Experience (QoE) у додатках WebRTC. Автори пропонують структуру для оцінки QoE у додатках WebRTC, що включає набір показників для вимірювання продуктивності мережі, якості медіа та сприйняття користувачами.

1.4 Огляд існуючих технологій для вирішення поставленої задачі

MongoDB – це документоорієнтована система керування базами даних NoSQL з відкритим вихідним кодом, яка зберігає дані в гнучкому, схожому на JSON форматі, відомому як BSON (Binary JSON). На відміну від традиційних реляційних баз даних, MongoDB не використовує таблиці з рядками та стовпцями для зберігання даних та використовує колекцію документів, які можуть мати складну ієрархічну структуру та можуть бути індексовані та отримані за різними критеріями.

Node.js – це міжплатформне середовище виконання JavaScript коду, яке дозволяє розробникам створювати швидкі, масштабовані та високопродуктивні мережеві програми. Він використовує керовану подіями модель вводу та виводу без блокування, що робить його ідеальним для створення застосунків у реальному часі, яким необхідно обробляти велику кількість одночасних з'єднань. Середовище побудовано на основі двигуна V8 JavaScript, який також використовується Google Chrome та надає низку вбудованих модулів і бібліотек для роботи з операціями вводу та виводу, мережами, файловими системами тощо.

Express.js – це фреймворк вебзастосунків для Node.js, що спрощує процес створення вебзастосунків і API (Application Programming Interface). Він надає набір надійних функцій та інструментів для створення вебзастосунків, включаючи маршрутизацію, проміжне програмне забезпечення, механізми шаблонів. Express.js також надає потужну систему маршрутизації, яка дозволяє легко зіставляти запити HTTP з певними обробниками коду, а також підтримує різноманітні механізми шаблонів, такі як Pug і Handlebars для створення динамічних вебсторінок. Також, він надає проміжне програмне забезпечення для обробки типових завдань, таких як журналювання, розбір запитів HTTP та обробка помилок.

Socket.IO – це бібліотека з відкритим кодом для створення вебзастосунків у реальному часі, які вимагають двонаправленого зв'язку між клієнтом і сервером. Він створений на основі протоколу WebSocket, який забезпечує зв'язок між клієнтом і сервером у режимі реального часу з малою затримкою. Socket.IO надає простий API для надсилання та отримання повідомлень між клієнтом і сервером, підтримує широкий спектр платформ. Socket.IO використовує керовану подіями архітектуру, що означає, що він може обробляти велику кількість одночасних з'єднань і легко підлягати масштабуванню в міру зростання кількості користувачів та пристроїв. Він також підтримує функціонал кімнат та просторів імен, які спрощують групування клієнтів і керування спілкуванням між ними.

React – це бібліотека JavaScript з відкритим вихідним кодом для створення інтерфейсів користувача для вебзастосунків. React використовує компонентну архітектуру, яка дозволяє розробникам створювати багаторазові компоненти для інтерфейсу користувача, які можна використовувати в багатьох місцях у програмі. Компоненти створено за допомогою комбінації JavaScript, HTML і CSS і можуть бути легко створені для створення складних інтерфейсів користувача.

WebRTC (Web Real-Time Communications) – це технологія, що забезпечує одноранговий зв'язок між браузерами та іншими пристроями, не вимагаючи додаткового програмного забезпечення чи плагінів. WebRTC надає набір API, які дозволяють розробникам створювати програми для спілкування в реальному часі.

Він використовує комбінацію технологій, у тому числі WebSockets, STUN (Session Traversal Utilities for NAT) та TURN (Traversal Using Relay NAT), RTP (Real-time Transport Protocol) та RTCP (Real-Time Transport Control Protocol), ICE (Interactive Connectivity Establishment).

Вебтокени JSON (JWT) – широко використовуваний стандарт для створення безпечних токенів для автентифікації. JWT розроблено, щоб забезпечити простий і надійний спосіб автентифікації та авторизації користувачів без необхідності зберігати інформацію про користувачів на сервері.

Redux – бібліотека керування станом. Він підтримує односпрямований потік даних і пропонує передбачуваний контейнер стану для керування станом програми.

1.5 Постановка задачі

Вивчення іноземних мов завжди було невід’ємною частиною освіти та розвитку особистості. У сучасному світі актуальність вивчення іноземної мови важко не до оцінити. Актуальність підкреслюється зростаючим попитом на застосунки для вивчення мов та онлайн-курси.

Відповідно до звіту [12] проведеного компанією Meticulous Research, очікується, що глобальний ринок онлайн-вивчення мов зросте на 13,1% з 2021 по 2029 рік, досягнувши розміру ринку в 31,81 мільярда доларів США до 2029 року. У звіті зазначено, що зростаюча глобалізація та потреба в ефективній комунікації в реальному часі спонукають до зростання ринку онлайн-вивчення мов. Також, наголошується на зростаючому попиті на персоналізованому вивченню мови, попит на доступні та недорогі застосунки та на застосування технологій електронного навчання як на деяких ключових силах даного ринку.

Тема дослідження полягає в створенні вебзастосунку для покращення процесу вивчення іноземних мов з використанням сучасних технологій. Віртуальні кімнати забезпечують спільне та інтерактивне навчальне середовище. Учасники можуть брати участь у кооперативних заходах для вивчення у групах за допомогою ряду інструментів щоб разом працювати над завданнями та отримувати зворотний

зв'язок або індивідуальних заходах, завдання яких сфокусовані на персональне вивчення мов. Такий підхід може покращити зацікавленість та мотивацію студентів до вивчення іноземної мови.

Центральна проблема полягає у тому, що існуючі методи та інструменти вивчення іноземних мов не забезпечують повноцінного та ефективного навчання з використанням сучасних методів та технологій. Віртуальні кімнати для удосконалення розмовних навичок потребують більш специфічних варіантах відеоконференцій з інтерактивними елементами, варіативністю, завданнями та тестами, що направлені на вивчення мов та подальшим аналізом результатів, ніж застосунки загального використання.

Вивчення іноземної мови з використанням ресурсів електронного навчання та технології відеоконференції забезпечують можливість створення віртуальних занять у реальному часі, спрямованих на імітацію розмовних ситуацій. Це дозволить користувачам поліпшувати свої розмовні навички в безпечному та сприятливому середовищі та аналізувати результати для підвищення мотивації та ефективності вивчення мови.

Отже, метою цієї роботи є підвищення рівня комунікації та вивчення іноземної мови з використанням технології відеоконференції.

Об'єктом бакалаврської кваліфікаційної роботи є процеси комунікації та навчання у реальному часі.

Предметом роботи є програмні засоби та технології для вивчення іноземних мов.

Дослідження передбачає розробку та тестування вебзастосунку для вивчення іноземних мов, розробку системи оцінювання знань користувачів, а також аналіз інтеграції різних інструментів та для досягнення максимальної ефективності вивчення мови.

Завдання для досягнення поставленої мети:

– проаналізувати наявні публікації та аналоги, що пов'язані з вивченням іноземних мов, їхні функціональні можливості та недоліки;

- дослідити технологію WebRTC та інші технології, які можуть бути використані для реалізації вебзастосунку;
- розробити інформаційну систему вебзастосунку, визначити його основні функції та інструменти і відобразити їх за допомогою інформаційних діаграм;
- розробити вебзастосунок, що буде забезпечувати можливість проведення відеоконференцій та використовувати різноманітні інструменти для вивчення іноземних мов.
- протестувати вебзастосунок.

Висновки до розділу 1

Виходячи з викладеного в розділі аналізу, стає зрозуміло, що вивчення іноземної мови має численні переваги, зокрема покращує комунікативні навички, покращує когнітивні здібності та покращує можливості працевлаштування. З розвитком технологій електронне навчання стало ефективним і зручним способом вивчення мови.

В цьому розділі досліджено метод електронного навчання для вивчення іноземних мов за допомогою відеоконференцій, що дозволяє добути досвід вивчення мови, практикуючи спілкування у віртуальних кімнатах. Також було розглянуто переваги застосування відеоконференцій та позитивний вплив на розвиток розмовних навичок.

Проведений аналіз існуючих застосунків за яким можна зробити висновок, що для більш ефективного удосконалення розмовних навичок користувачі потребують більше інтерактивних елементів, сфокусованих на вивчення мов та подальшим аналізом результатів.

Тому було вирішено розробити вебзастосунок для вивчення іноземних мов.

2 МЕТОДИ ТА ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ СТВОРЕННЯ ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ

2.1 Використання Socket.IO з технологією WebRTC

Для того щоб розробити вебзастосунок з технологією WebRTC треба зробити зв'язок між клієнтом та сервером. Для даного завдання використаємо бібліотеку Socket.IO, що спрощує двосторонній зв'язок у реальному часі між клієнтами та серверами, забезпечуючи безперебійну та надійну систему для доставки та отримання даних в різних форматах [13]. Він розроблений на основі протоколу WebSocket і надає додаткові гарантії. WebSocket є протоколом інтернет-зв'язку, що пропонує двосторонній канал через з'єднання TCP (Transmission Control Protocol).

Socket.IO в основному використовується в Node.js, що забезпечує масштабоване середовище виконання, кероване подіями.

Разом з Socket.IO і WebRTC можна створювати надійні програми для спілкування в реальному часі. Socket.IO служить системою сигналізації, що виконує початкове встановлення з'єднання, керування кімнатами та зв'язок подій між клієнтами та сервером, тоді як WebRTC обробляє фактичну однорангову доставку медіа даних [14].

WebRTC потребує каналу сигналізації для створення з'єднання між двома одноранговими клієнтами. Цей сигнальний канал, відомий як Socket.IO, який обмінюється сигнальними повідомленнями між клієнтами та сервером [15]. Сигнальні повідомлення включають інформацію про узгодження, необхідну для встановлення однорангового з'єднання, описи сеансу та кандидатів ICE (Interactive Connectivity Establishment).

Коли багатьом клієнтам потрібно приєднатися в режимі реального часу до чату або відеоконференції то клієнти можуть приєднуватися до певних кімнат за підтримки Socket.IO, що може адмініструвати кімнати та сприяти спілкуванню між

учасниками. Цей функціонал є основним для створення вебзастосунків з можливістю відеоконференцій або групового чату.

Модель зв'язку на основі подій Socket.IO підходить до каналу даних WebRTC. В розробці вебзастосунку можна використовувати Socket.IO для визначення спеціальних подій і ініціювання дій як на стороні клієнта, так і на стороні сервера. Наприклад, коли користувач приєднується до кімнати або надсилає повідомлення в чаті, Socket.IO може обробити подію та оновити інтерфейс користувача або виконати певні дії на стороні сервера.

2.1 Методи для вирішення задачі створення відеоконференції

Одним з підходів для даної задачі є можливість встановлювати однорангові з'єднання між пристроями, що дозволяє здійснювати прямий зв'язок між ними без необхідності у використанні центрального сервера. З'єднання може бути встановлено в сітчастій топології, де кожен пристрій у мережі підключається до кількох інших пристроїв, утворюючи мережеву структуру. У сітчастій топології з використанням WebRTC дані передаються безпосередньо між користувачами, що призводить до більш економічної експлуатації та надає вищий ступінь конфіденційності [16]. Топологія сітки дозволяє використовувати кілька шляхів для передачі даних між пристроями, що робить мережу більш надійною та стійкою до збоїв. Щоб створити таку мережу, кожен пристрій у мережі повинен мати унікальний ідентифікатор користувача. Коли пристрій хоче підключитися до іншого пристрою в мережі, він надсилає цільовому пристрою запит на підключення, який містить його ідентифікатор. Якщо цільовий пристрій доступний та бажає встановити з'єднання, він відповідає власним ідентифікатором і таким чином пристрої обмінюються своїми можливостями та узгоджують параметри з'єднання. Після встановлення з'єднання пристрої можуть обмінюватися даними напряму, тобто без необхідності використання центрального сервера. Якщо третій

пристрій хоче приєднатися до мережі, він може підключитися до одного або кількох існуючих пристроїв, утворюючи нові зв'язки в топології сітки.

Сітчасті мережі мають кілька переваг перед традиційними моделями зв'язку клієнт-сервер. По-перше, вони більш стійкі до збоїв мережі, оскільки кожен пристрій може підключатися до кількох інших пристроїв, а дані можуть передаватися різними шляхами. По-друге, вони більш безпечні, оскільки дані шифруються наскрізно і немає центрального сервера [17].

Наступний підхід полягає у використанні топології Multipoint Conferencing Unit (MCU), що діє як міст між кількома учасниками, дозволяючи їм спілкуватися один з одним одночасно [18]. У топології MCU сервер отримує аудіопотоки та відеопотоки від кожного учасника та змішує їх у єдиний потік, який надсилається назад усім учасникам.

Топологія MCU може обслуговувати більшу кількість учасників, оскільки центральний сервер діє як центр для всіх комунікацій та може надавати більш розширені функції, такі як запис і трансляція конференції. Даний підхід може зменшити вимоги до пропускної здатності для кожного учасника, оскільки сервер може оптимізувати потоки даних на основі умов мережі.

Щоб створити топологію MCU, кожен учасник конференції повинен підключитися до центрального сервера за допомогою з'єднання WebRTC. Сервер повинен підтримувати протокол WebRTC і мати необхідні можливості для змішування потоків даних.

Після встановлення з'єднання кожен учасник може надсилати свої аудіопотоки та відеопотоки на сервер, який змішує їх разом і надсилає об'єднаний потік назад усім учасникам.

Останій підхід, що розглядається для даної задачі – це Selective Forwarding Unit (SFU). Сервер діє як маршрутизатор між декількома учасниками, дозволяючи їм спілкуватися один з одним вибірково [19]. У топології SFU сервер отримує потоки даних від кожного учасника та пересилає їх іншим учасникам.

Топологія SFU може працювати з великою кількістю учасників, оскільки центральний сервер діє як маршрутизатор для всіх комунікацій. Також, вона може зменшити вимоги до пропускну здатності для кожного учасника, оскільки сервер пересилає лише необхідні потоки і за допомогою даної топології можна забезпечити більший контроль над спілкуванням, оскільки кожен учасник може вибрати, які потоки отримувати, а які ігнорувати.

Для того щоб створити топологію SFU, кожен учасник конференції повинен підключитися до центрального сервера. Кожен учасник може надсилати свої потоки даних після встановлення з'єднання з сервером, який пересилає їх іншим учасникам відповідно до їхніх уподобань. Сервер також може надавати додаткові функції, такі як запис і потокове передавання конференції, які можуть покращити якість конференції.

На рисунку 2.1 представлено види топологій для багатокористувацького з'єднання.

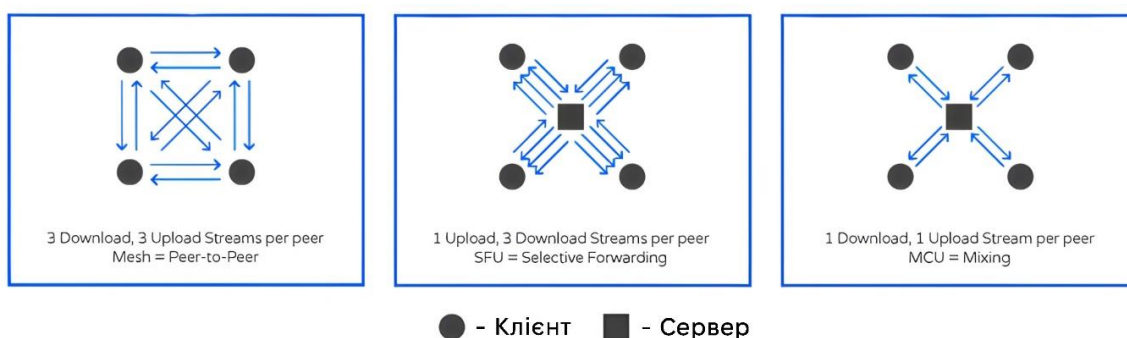


Рисунок 2.1 – Види топологій

2.2 Вибір топології з'єднання за допомогою методу аналізу ієрархій

Метод аналізу ієрархій передбачає використання попарних порівнянь для визначення відносної важливості набору критеріїв або альтернатив [20]. Це дозволяє особам, які приймають рішення, призначати ваги кожному критерію на основі їх передбачуваної важливості та оцінювати альтернативи на основі того,

наскільки вони задовольняють кожному критерію. Для проведення обґрунтованих чисельних порівнянь не слід порівнювати більш ніж дев'ять елементів.

Для нашої задачі треба обрати топологію, що можна реалізувати за допомогою WebRTC. Першочергово треба визначити критерії та альтернативні варіанти для даної задачі.

Критерії:

- вартість сервера (Q1);
- якість відеопотоку (Q2);
- масштабованість (Q3);
- безпека та конфіденційність (Q4);
- простота розгортання та обслуговування (Q5).

Альтернативні варіанти:

- Mesh;
- SFU (Selective Forwarding Unit);
- MCU (Multipoint Control Unit).

Далі треба представити задачу у вигляді ієрархії (рис. 2.2), де на першому рівні знаходиться головна ціль, на другому рівні знаходяться фактори або критерії, які уточнюють головну мету та на третьому знаходяться альтернативні варіанти, які повинні бути оцінені по відношенню до критеріїв другого рівня [21].

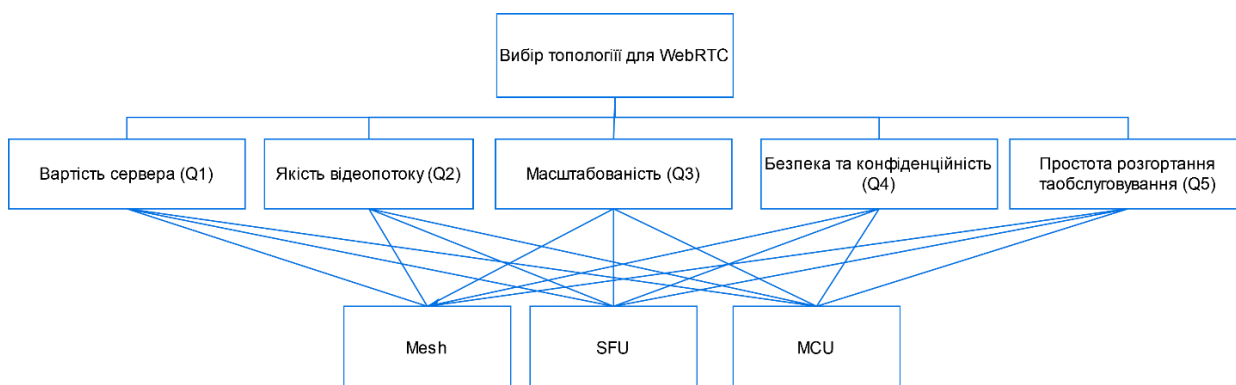


Рисунок 2.2 – Ієрархічне представлення задачі

Щоб побудувати матрицю попарного порівняння (МПП), треба порівняти кожен альтернативу з усіма іншими за шкалою відносної важливості.

Наступним кроком складається матриця попарних порівнянь для порівняння відносної важливості критеріїв на другому рівні по відношенню до загальної мети на першому рівні.

Для кожної матриці попарних порівнянь розраховується вектор локальних пріоритетів, який визначає відносний вплив множини елементів матриці попарних порівнянь на елемент вищого рівня ієрархії.

Для цього необхідно визначити множину власних векторів матриці попарних порівнянь, а потім нормалізувати результат до одиниці, отримавши при цьому вектор локальних пріоритетів. Розрахунок власних векторів здійснюється за наступною формулою:

$$B_i = \sqrt[n]{\prod_{j=1}^n a_{ij}}, \quad (2.1)$$

де a_{ij} – елемент матриці попарних порівнянь;

n – кількість елементів матриці попарних порівнянь.

Розрахунок вектору локальних пріоритетів відбувається за формулою:

$$P_i = \frac{B_i}{\sum_{k=1}^n B_k}, \quad (2.2)$$

де B_i, B_k – власні вектори;

n – кількість елементів матриці попарних порівнянь.

Індекс узгодженості (ІУ) надає інформацію про ступінь порушення чисельної і транзитивної узгодженості. Для розрахунку індексу узгодженості спочатку

необхідно визначити максимальне власне значення. Розрахунок максимального власного значення здійснюється за формулою:

$$\lambda_{max} = \sum_{j=1}^n (\sum_{i=1}^n a_{i,j} \cdot P_j), \quad (2.3)$$

де P_j – вектор пріоритетів;

$a_{i,j}$ – елемент матриці попарних порівнянь;

n – кількість елементів матриці попарних порівнянь.

Індекс узгодженості визначається за наступною формулою:

$$C_I = \frac{\lambda_{max} - n}{n - 1}, \quad (2.4)$$

де λ_{max} – максимальне власне значення;

n – кількість елементів матриці попарних порівнянь.

Відношення узгодженості (ВУ) дозволяє оцінити ступінь відхилення суджень від випадкових суджень. Розрахунок відношення узгодженості здійснюється за формулою:

$$C_R = \frac{C_I}{R_I} \cdot 100\%, \quad (2.5)$$

де C_I – індекс узгодженості суджень матриці попарних порівнянь;

R_I – індекс випадкової узгодженості.

Вектор глобальних пріоритетів синтезуються, починаючи з другого рівня ієрархії і до низу за наступною формулою:

$$PG_i = \sum_{j=1}^n P_i^j \cdot P^j, \quad (2.6)$$

де P_i^j – локальний пріоритет і-го альтернативного варіанту по відношенню до j-го критерію;

P^j – локальний пріоритет j-го критерію по відношенню до головної мети.

Таблиця 2.1 – МПП

	Q1	Q2	Q3	Q4	Q5	B_i	P_i	λ
Q1	1	3	4	5	2	2,60	0,43	0,97
Q2	0,3	1	2	2	0,5	0,92	0,15	1,06
Q3	0,3	0,5	1	1	0,33	0,53	0,09	0,95
Q4	0,2	0,5	1	1	0,33	0,50	0,08	0,99
Q5	0,5	2	3	3	1	1,55	0,25	1,06
$\sum_{j=1}^n$	2,3	7	11	12	4,17	6,11	1	5,03

Таблиця 2.2 – ІУ та ВУ для МПП

C_I	0,01
C_R	0,72%

Таблиця 2.3 – МПП альтернатив по відношенню до критерію «Вартість сервера»

	Mesh	SFU	MCU	B_i	P_i	λ
Mesh	1	3	5	2,46621	0,63699	0,97671
SFU	0,3333	1	3	1	0,25828	1,11923
MCU	0,2	0,333	1	0,40548	0,10473	0,94256
$\sum_{j=1}^n$	1,5333	4,333	9	3,87169	1	3,03851

Таблиця 2.4 – ІУ та ВУ по відношенню до критерію «Вартість сервера»

C_I	0,019
C_R	3,32%

Таблиця 2.5 – МПП альтернатив по відношенню до критерію «Якість відеопотоку»

	Mesh	SFU	MCU	B_i	P_i	λ
Mesh	1	0,25	0,3333	0,43679	0,12196	0,97566
SFU	4	1	2	2	0,55842	0,97724
MCU	3	0,5	1	1,14471	0,31962	1,06539
$\sum_{j=1}^n$	8	1,75	3,3333	3,5815	1	3,01829

Таблиця 2.6 – ІУ та ВУ по відношенню до критерію «Якість відеопотоку»

C_I	0,009
C_R	1,577%

Таблиця 2.7 – МПП альтернатив по відношенню до критерію «Масштабованість»

	Mesh	SFU	MCU	B_i	P_i	λ
Mesh	1	0,25	0,3333	0,43679	0,11722	0,93777
SFU	4	1	3	2,28943	0,61441	0,97282
MCU	3	0,3333	1	1	0,26837	1,16293
$\sum_{j=1}^n$	8	1,583	4,3333	3,72622	1	3,07351

Таблиця 2.8 – ІУ та ВУ по відношенню до критерію «Масштабованість»

C_I	0,037
C_R	6,337%

Таблиця 2.9 – МПП альтернатив по відношенню до критерію «Безпека та конфіденційність»

	Mesh	SFU	MCU	B_i	P_i	λ
Mesh	1	3	4	2,28943	0,62501	0,9896
SFU	0,3333	1	2	0,87358	0,23849	1,07319
MCU	0,25	0,5	1	0,5	0,1365	0,9555
$\sum_{j=1}^n$	1,5833	4,5	7	3,66301	1	3,01829

Таблиця 2.10 – ІУ та ВУ по відношенню до критерію «Безпека та конфіденційність»

C_I	0,009
C_R	1,577%

Таблиця 2.11 – МПП альтернатив по відношенню до критерію «Простота розгортання та обслуговування»

	Mesh	SFU	MCU	B_i	P_i	λ
Mesh	1	3	2	1,81712	0,53961	0,98929
SFU	0,3333	1	0,5	0,55032	0,16342	0,98054
MCU	0,5	2	1	1	0,29696	1,03936
$\sum_{j=1}^n$	1,8333	6	3,5	3,36744	1	3,0092

Таблиця 2.12 – ІУ та ВУ по відношенню до критерію «Простота розгортання та обслуговування»

C_I	0,005
C_R	0,793%

Таблиця 2.13 – Синтез глобальних пріоритетів

	Q1	Q2	Q3	Q4	Q5	PG_i
Mesh	0.637	0.122	0.117	0.625	0.539	0.489
SFU	0.258	0.559	0.614	0.238	0.164	0.309
MCU	0.105	0.320	0.268	0.137	0.297	0.203

З отриманих результатів видно, що альтернативний варіант «Mesh» є найкращим серед інших за визначеними критеріями для даної задачі із загальною оцінкою 0.489 (рис. 2.3). Найгірший альтернативний варіант – «MCU» з оцінкою 0.203.

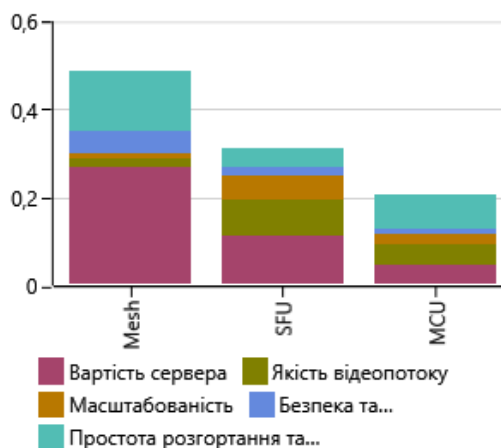


Рисунок 2.3 – Зважені атрибути

Висновки до розділу 2

В даному розділі було прийнято рішення при використанні технологій комунікації у реальному часі та використано метод аналізу ієрархій для забезпечення структурованого підходу до прийняття рішень щодо вибору топології, в якому було враховувано різні критерії, які мають відношення до рішення даної задачі, призначено ваги кожному з цих елементів на основі їх відносної важливості і розраховано остаточний бал для кожної альтернативи, що дозволило прийняти рішення про застосування сітчастої топології, що найкраще відповідає цілям даної розробки.

Судження не порушують чисельної і транзитивної узгодженості та показники відношення узгодженості не більше за 10%.

Проаналізувавши результати вибору топології, можна підкреслити фактори, що вплинули на вибір:

- одноранговий зв'язок, що відбувається безпосередньо між пристроями без центральної точки керування і робить зв'язок дуже стійким до збоїв і зменшує затримку в спілкуванні;
- економічна ефективність, оскільки зв'язок відбувається безпосередньо між пристроями, це зменшує вимоги до пропускну здатності;
- приватність, що забезпечує наскрізне шифрування та є менш вразливим до кібератак.

Отже, приймаємо рішення про використання сітчастої топології для подальшого моделювання та проектування інформаційної системи та програмної реалізації вебзастосунку для вивчення іноземних мов.

3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Проєктування інформаційної системи

Першочергово, розглянемо як працюють процеси входу та реєстрації в вебзастосунку, що підключений до внутрішнього сервера (рис. 3.1).

Користувач після реєстрації може отримати доступ до свого акаунту. Під час входу, сервер отримує запит на вхід від клієнта і перед наданням доступу звіряє надані облікові дані з даними користувача в базі даних. На сервері створюється токен JWT і потім надсилається клієнту у відповідь, якщо облікові дані є валідними.

Клієнт додає токен JWT до заголовка HTTP (HyperText Transfer Protocol) для всіх подальших запитів до ресурсів, що захищені сервером [22]. Отже, токен JWT служить методом автентифікації та авторизації.

Розмістивши JWT у заголовку кожного запиту, клієнт може надсилати запити до захищених ресурсів внутрішнього сервера. Роль серверу полягає в перевірці на дійсність і цілісність токена коли отримує запит з токеном JWT у заголовку і повертає відповідь клієнту, якщо токен JWT дійсний.

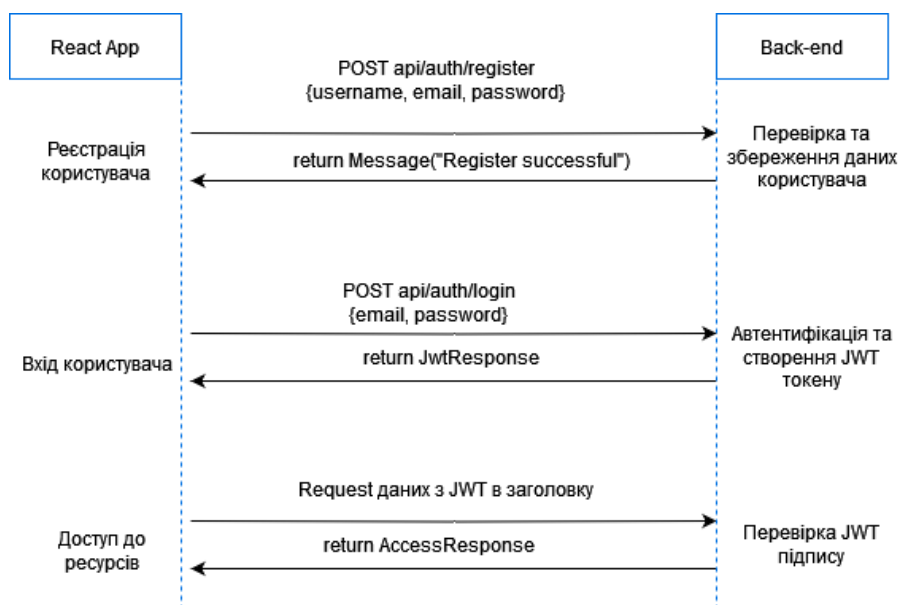


Рисунок 3.1 – Діаграма послідовності автентифікації

Потік даних і архітектуру вебзастосунку, що обробляє завдання пов'язані з автентифікацією за допомогою Redux, React Router і Redux Thunk Middleware зображено на рисунку 3.2.

Redux – надійний контейнер стану, використовується для керування станом програми [23]. Щоб отримати доступ до стану програми, компонент встановлює з'єднання з Redux Store. Будь-який компонент усередині програми може отримати доступ до Redux Store, який керує загальним станом програми.

Компоненти входу та реєстрації служать формами входу та реєстрації вебзастосунку React. Вони відповідають за збір інформації, введеної користувачем, і керування процедурою подання. Коли компоненти входу та реєстрації отримують надсилання форми, вони надсилають дії автентифікації до проміжного програмного забезпечення Redux Thunk. Проміжне програмне забезпечення Redux Thunk виконує виклики API, пов'язані з автентифікацією, за допомогою сервісу автентифікації. Методи сервісу автентифікації використовують axios для виконання HTTP-запитів. Він також зберігає або отримує токени JWT з локального сховища веббраузера.

Кінцеві точки API на стороні сервера, відповідальні за автентифікацію та реєстрацію користувачів, спілкуються з сервісом автентифікації. Ці кінцеві точки API взаємодіють із базою даних, перевіряють дані користувача та створюють необхідні події.

Далі сервіс автентифікації надсилає дані до Redux Thunk Middleware після отримання відповідей API. Потім проміжне програмне забезпечення надсилає відповідні команди для оновлення Redux Store відповідними даними, такими як успішна чи невдала автентифікація користувача.

Компонент «Dashboard» відображає головну панель та певні дані користувача після успішного входу.

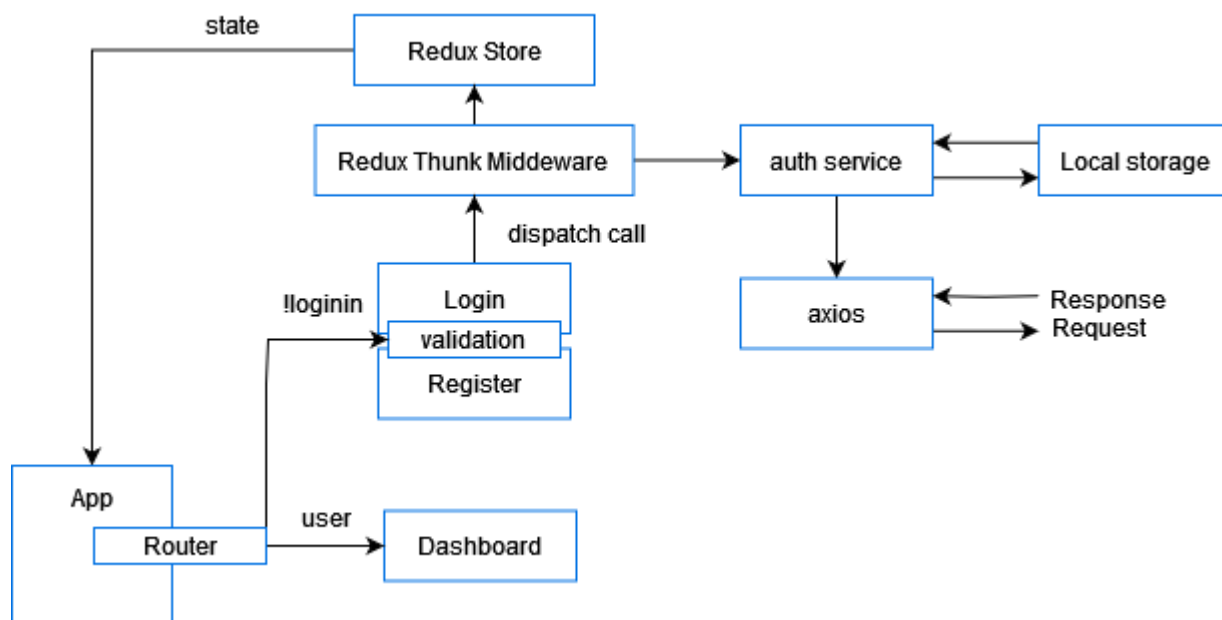


Рисунок 3.2 – Діаграма потоку даних і архітектури вебзастосунку

Модель «сутність-зв'язок» (рис. 3.3) потрібна щоб візуально визначити основні сутності для створення схеми Mongoose, що визначає структуру та властивості документа в колекції MongoDB та для позначення зв'язків між сутностями [24]. За допомогою визначених Schema можливо вказати обмеження, індекси, очікувані атрибути та значення.

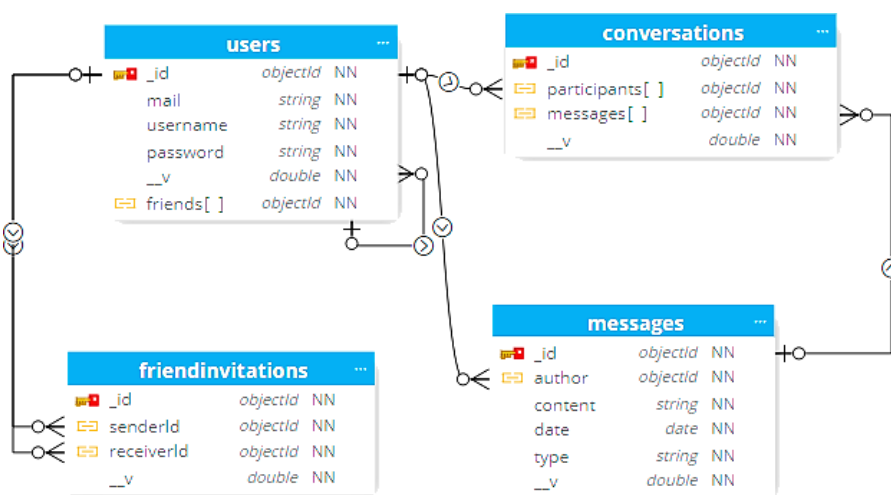


Рисунок 3.3 – Модель «сутність-зв'язок»

Для того, щоб користувачі мали можливість безпосередньо спілкуватися один з одним, не використовуючи при цьому сервер-посередник, треба застосувати певну послідовність дій [25].

Для авторизованих користувачів надається можливість використовувати кімнати до яких клієнт подає запит на приєднання за певним ідентифікатором або створює її. У результаті користувач підключається до певної кімнати, що дозволяє йому отримувати події, призначені для цього простору.

Отже, за допомогою Socket.IO відправник і одержувач потрапляють у спільний простір де вони мають певний функціонал за яким вони можуть побудувати канал зв'язку.

Запит на з'єднання надсилає відправник одержувачу. Для надсилання цього запиту використовується підключення Socket.IO. Запит на підключення приймає одержувач, який потім підключається до відправника. В результаті вони запускають свої потоки даних після успішного з'єднання (рис. 3.4).

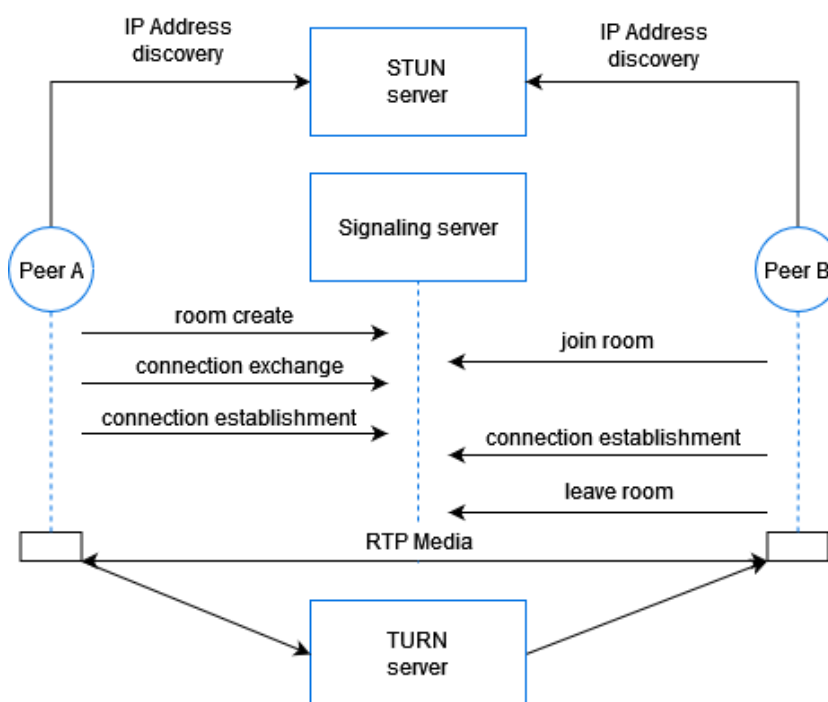


Рисунок 3.4 – Діаграма послідовності з'єднання

Висновки до розділу 3

Проектування системи пропонує дисциплінований і методичний підхід до організації, інтеграції та планування різних компонентів. Проектування системи полегшує створення ефективних, надійних, орієнтованих на користувача рішень, враховуючи цілі, ресурси, здатність до масштабування, досвід користувача та оптимізацію продуктивності та закладає основу для досягнення бажаних цілей, оптимізації загального виконання проєкту.

В даному розділі було розглянуто проектування інформаційної системи вебзастосунку для вивчення іноземних мов.

Розглянуто процес автентифікації за допомогою діаграми послідовності на якій зображено основні події під час входу та реєстрації.

Побудовано архітектуру вебзастосунку та відображено потік даних, де показано як встановлюється з'єднання з Redux Store та надання доступу до стану програми.

Візуально визначено основні сутності, використовуючи модель «сутність-зв'язок», що визначає структуру та властивості документа в колекції та для позначення зв'язків між сутностями.

Сформовано послідовність з'єднання користувачів, що потрапляють у спільний простір, де вони мають певний функціонал для створення відеоконференції.

Отже, було побудовано інформаційну систему для основних етапів в ході розробки вебзастосунку.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ

4.1 Опис програмної реалізації

Щоб отримати доступу до вебзастосунку треба зареєструвати обліковий запис (рис. 4.1). Для цього треба заповнити поля вводу форми. Поле електронної скриньки обов'язково повинно бути заповненим унікальним та валідним значенням, що включає в себе назву поштової скриньки та доменне ім'я, яке представляє адміністративну область поштової скриньки. Поле вводу, що пов'язане з ім'ям користувача повинно мати більше ніж три символи. Поле вводу пароля повинно бути в діапазоні від 6 до 20 символів.

Поля для формування облікового запису проходять валідацію як на стороні сервера так і на стороні клієнта.

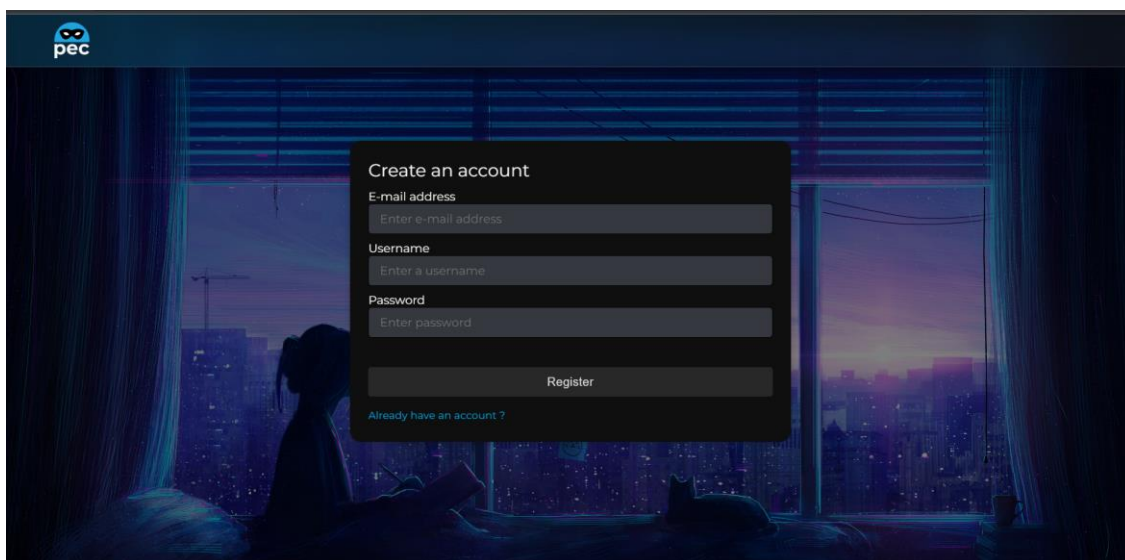


Рисунок 4.1 – Форма реєстрації користувача

Якщо вже є наявний обліковий запис користувача то для доступу до головної сторінки вебзастосунку треба перейти до форми входу (рис. 4.2) та ввести дійсні дані користувача. На клієнтській частині відбувається перевірка на даних валідність та на стороні сервера перевіряється валідність та достовірність введених даних, перевіряючи вхідні дані з даними в базі даних. У разі успішного входу користувачу

певний тип інструментарію для вивчення іноземної мови або під'єднатися до кімнати чи запросити нового учасника.

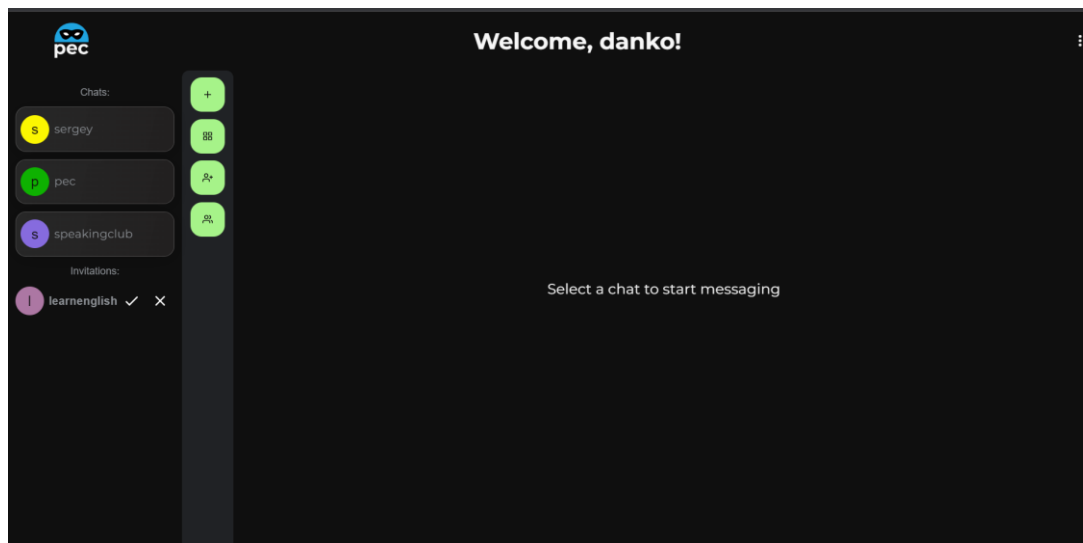


Рисунок 4.4 – Головна сторінка вебзастосунку

Можна додати користувачів до закріпленого списку щоб мати можливість кооперативного вивчення іноземних мов. В формі запрошення (рис. 4.5) треба вказати ідентифікатор користувача, що існує в системі. Після того як було відправлено запрошення, користувач якому адресувалося дане запрошення, отримає його. Користувач може підтвердити або скасувати запрошення. У разі заповнення форми запрошення невірними даними, користувач буде повідомлений про це.

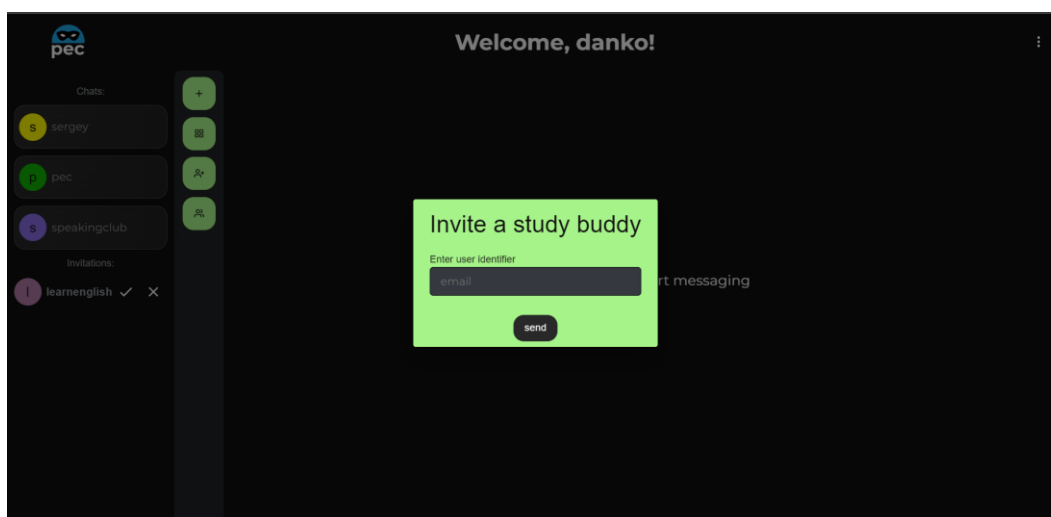


Рисунок 4.5 – Форма запрошення

На панелі навігації можна створити кімнату для відеоконференції (рис. 4.6). У кожній кімнаті наявний ідентифікатор кімнати за яким відбувається з'єднання з іншими користувачами.

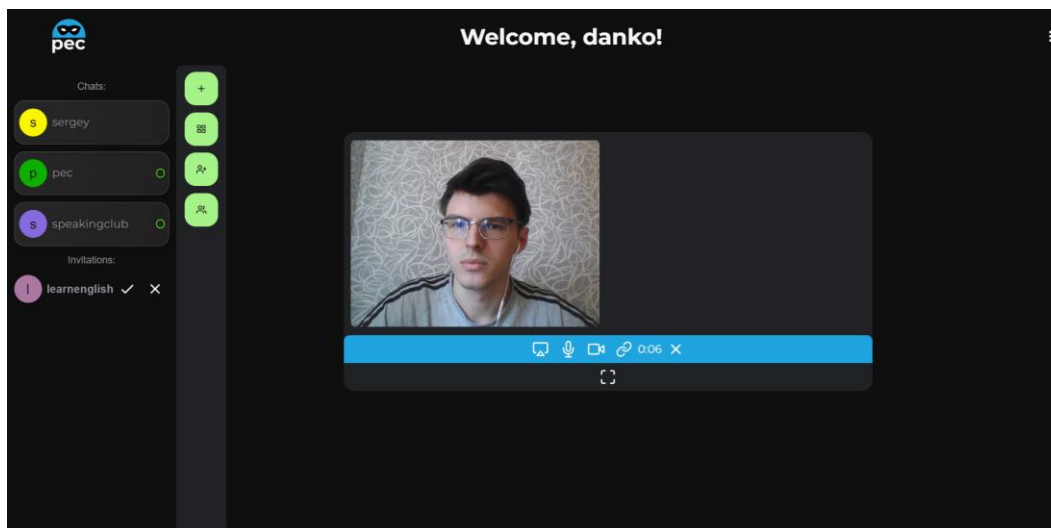


Рисунок 4.6 – Створення кімнати для відеоконференції

На рисунку 4.7 зображено приклад даних, що зберігаються у Redux Store при створенні кімнати.

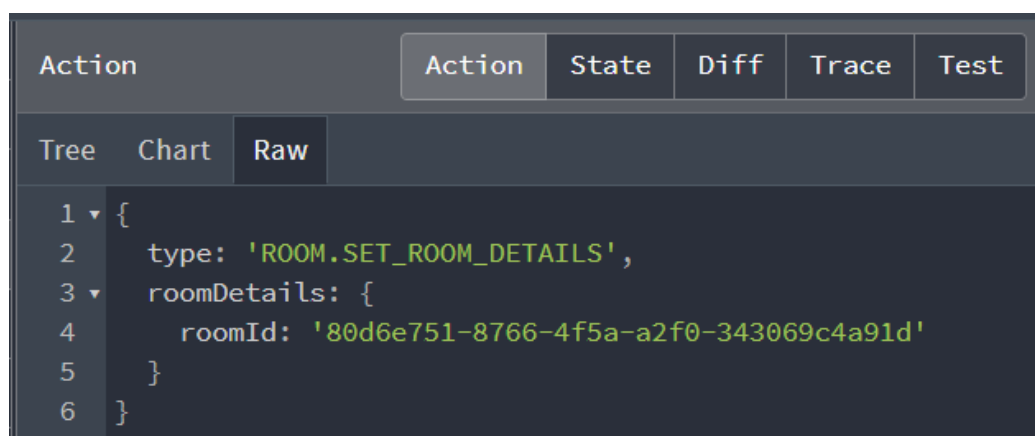


Рисунок 4.7 – Дані кімнати

На рисунку 4.8 зображено панель для взаємодії та управління аудіо та відео даними в кімнаті.



Рисунок 4.8 – Панель відеоконференції

Можна скопіювати ідентифікатор кімнати та переслати обраному користувачу (рис. 4.9).

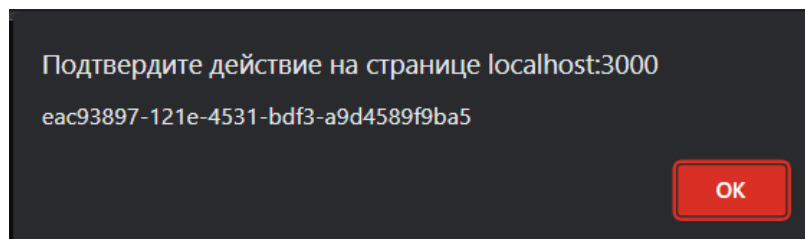


Рисунок 4.9 – Копіювання ідентифікатора кімнати

За допомогою панелі відеоконференції можна обрати функціонал для поширення контенту екрана (рис. 4.10).

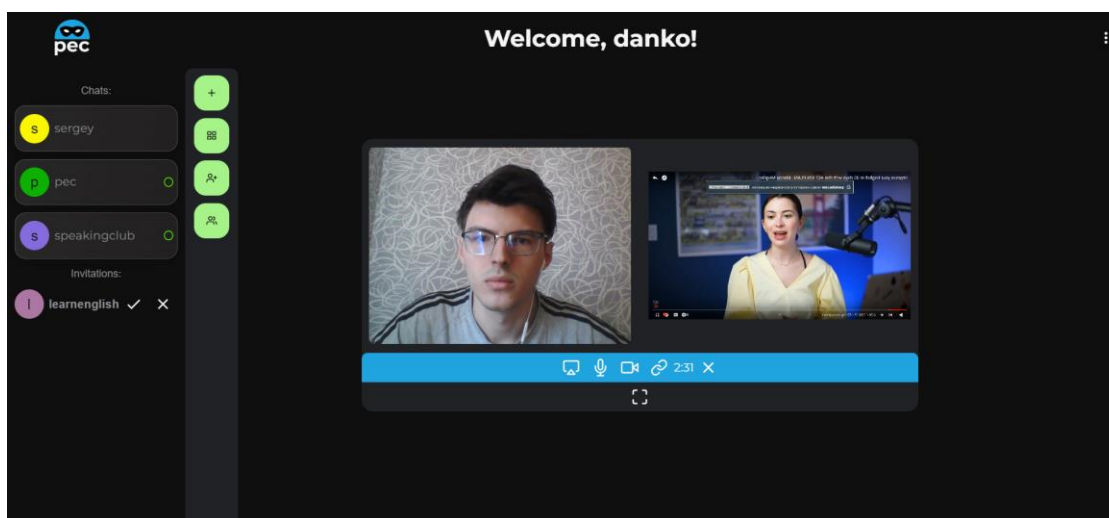


Рисунок 4.10 – Поширення контенту екрана

Інший користувач, що має ідентифікатор створеної кімнати може під'єднатися до неї за допомогою функціоналу панелі навігації (рис. 4.11).

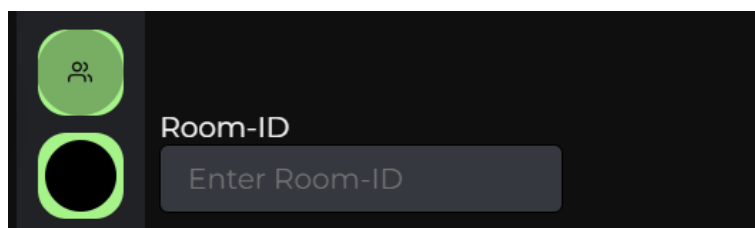


Рисунок 4.11 – Вхід до кімнати

Після введення ідентифікатору створеної кімнати, користувач додається до кімнати та створюється з'єднання для відеоконференції (рис. 4.12).

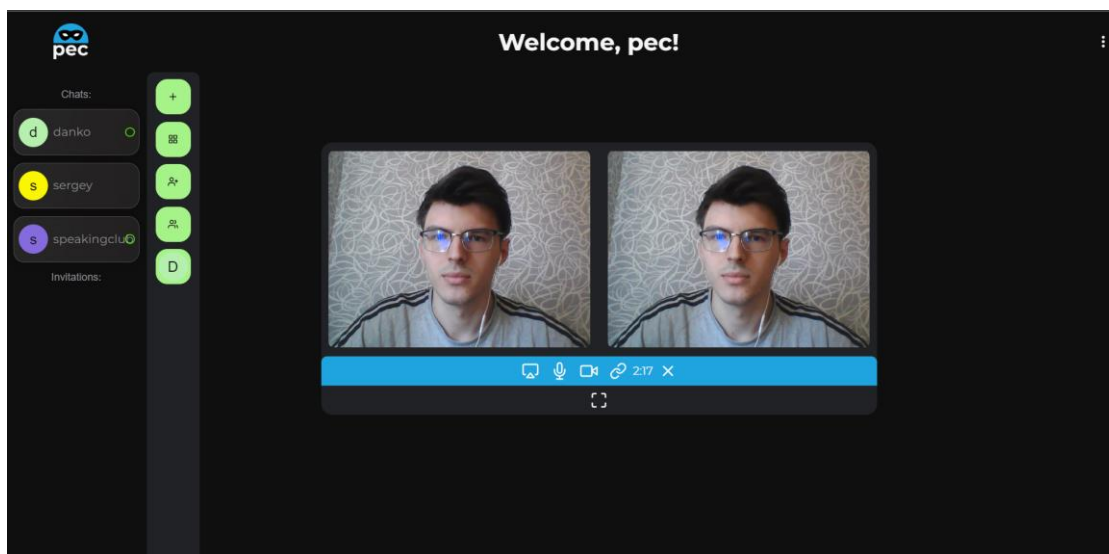


Рисунок 4.12 – З'єднання між двома користувачами

За допомогою використання сітчастої топології надається можливість під'єднатися до спільної кімнати більшої кількості користувачів. Можливість в багатокористувацькому з'єднанні зображено на рисунку 4.13.

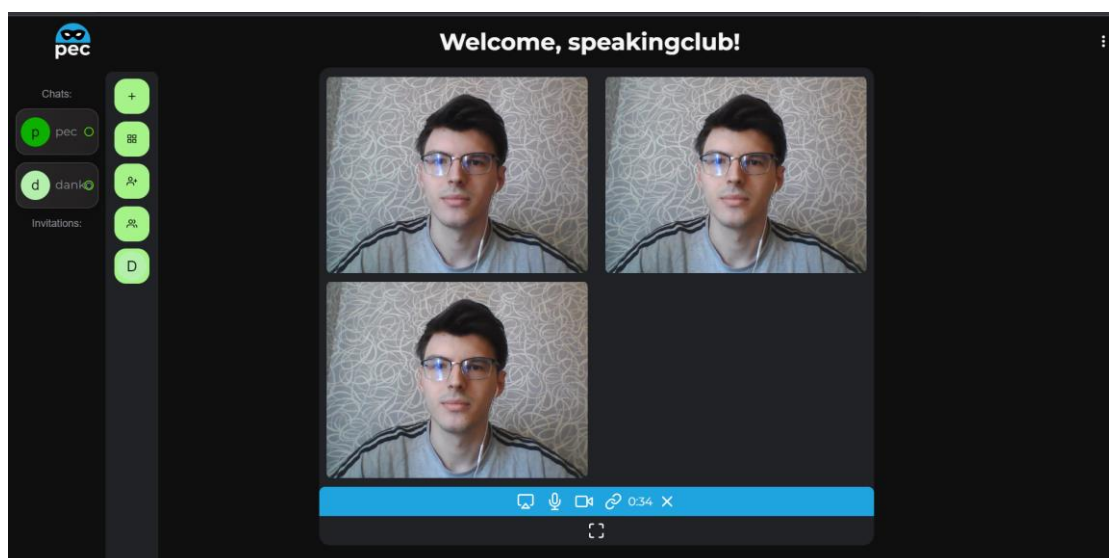


Рисунок 4.13 – Багатокористувацьке з'єднання

На панелі навігації також доступне меню для вибору інструментів для вивчення іноземних мов (рис 4.14).

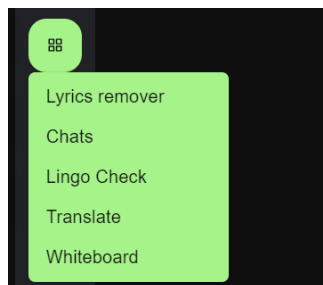


Рисунок 4.14 – Меню вибору

В меню можна обрати чат для обміну повідомленнями (рис. 4.15).

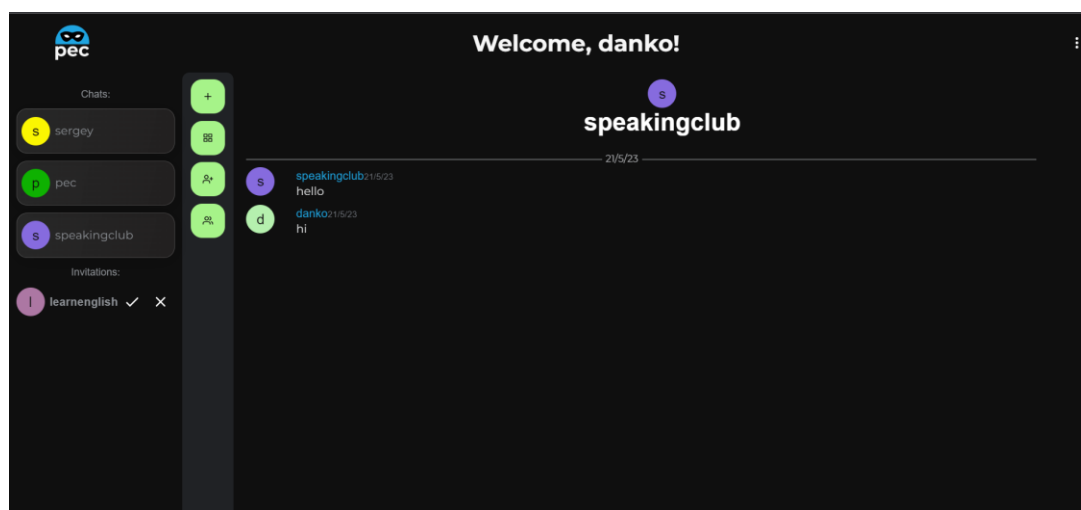


Рисунок 4.15 – Чат для обміну повідомлень

Крім того, в меню доступна вправа для побудови речень методом перетаскування слів відповідно до перекладу (рис. 4.16).

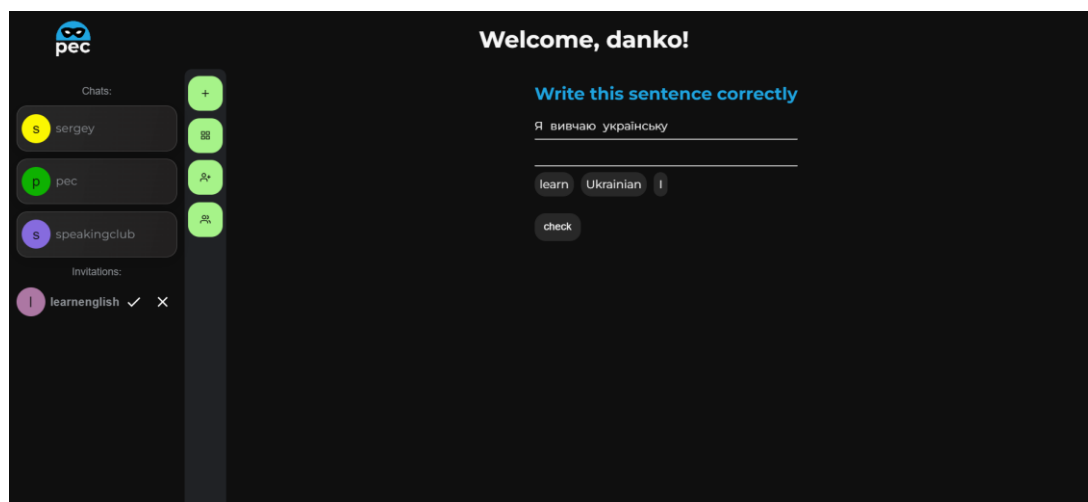


Рисунок 4.16 – Вправа для побудови речень

Доступно в меню перекладання слів (рис. 4.17). На вхід подається текст та мова на яку треба перекласти, за вхідним текстом автоматично визначається вхідна мова.

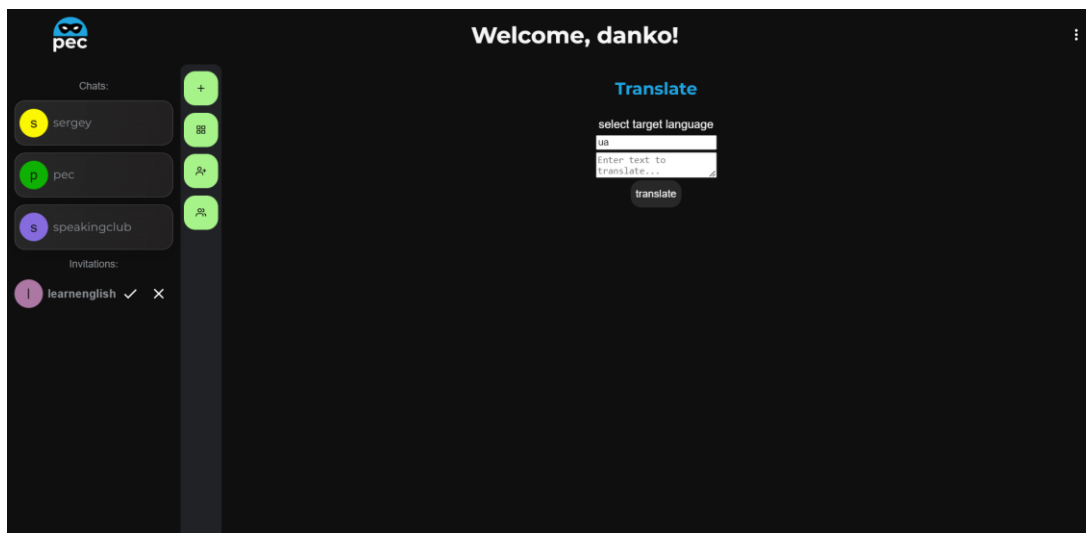


Рисунок 4.17 – Перекладач

В меню для кооперативного режиму доступна дошка для малювання. Можна обрати колір та за використовуючи курсор написати чи зобразити щось (рис. 4.18).

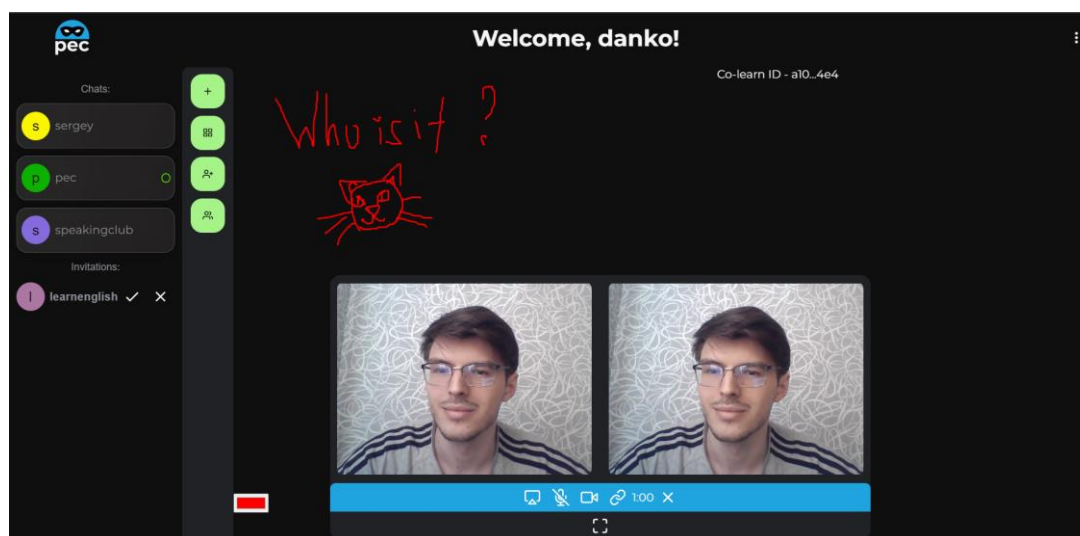


Рисунок 4.18 – Дошка для малювання

Дошка знаходиться в спільному просторі з учасниками які знаходяться в певній кімнаті. Учасники можуть додавати, видаляти свої написи та малюнки, бачити написи інших учасників (рис. 4.19).

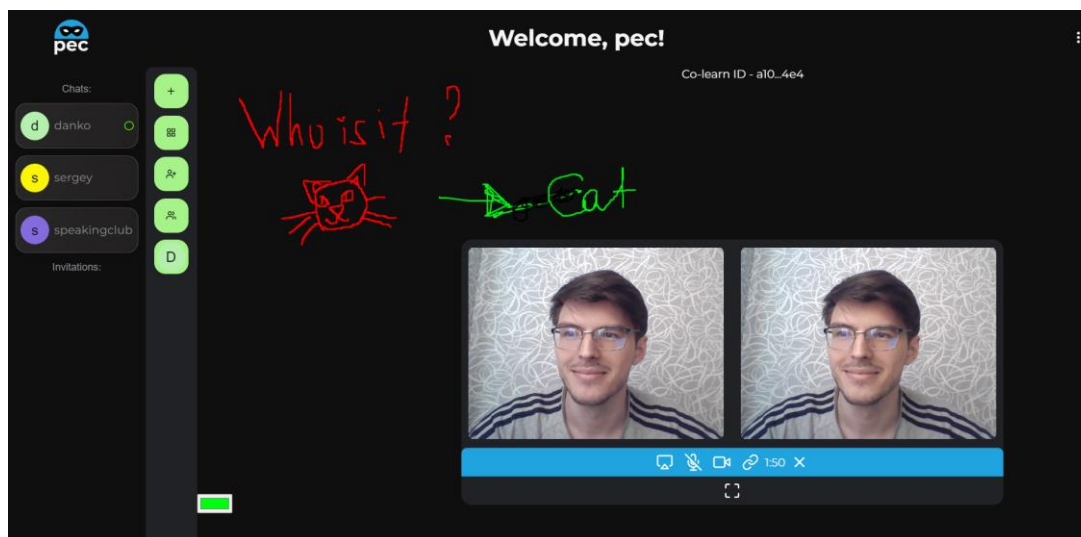


Рисунок 4.19 – Взаємодія з іншими користувачами

Також в меню доступний інструментарій для в кооперативного або індивідуального вивчення за допомогою пісень (рис. 4.20). Додаток полягає в тому, що він автоматизує процес перетворення тексту на вправи за яким можна слухати пісню та вставляти пропущенні слова, що розвиває навички аудіювання. Взаємодія відбувається за допомогою перетаскування слів.

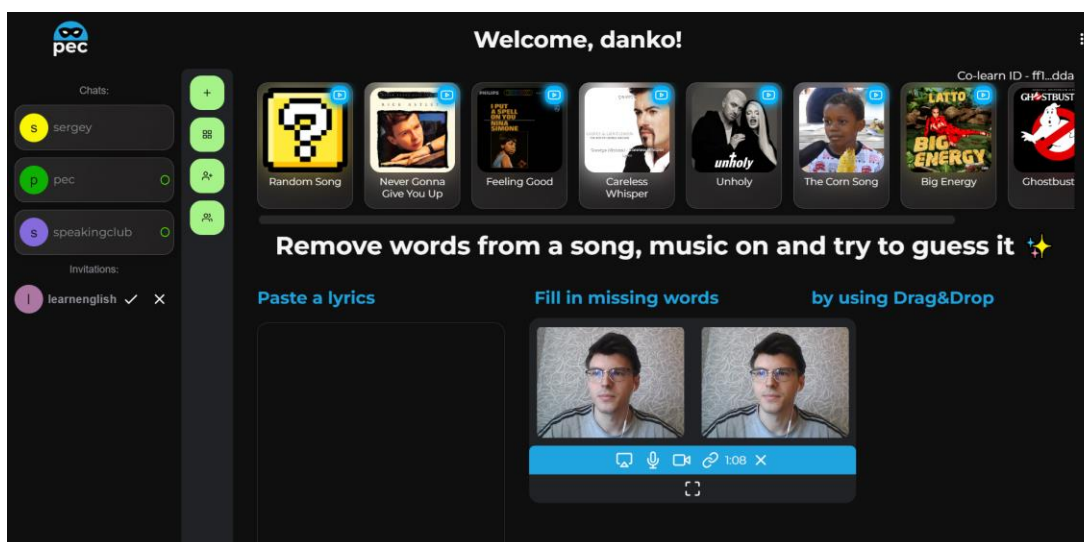


Рисунок 4.20 – Додаток для розвитку аудіювання

На рисунку 4.21 зображено процес кооперативного навчання. Учасники знаходять в одній кімнаті. Події, що відбуваються на одному з клієнтів автоматично відбуваються на іншому (рис. 4.22).

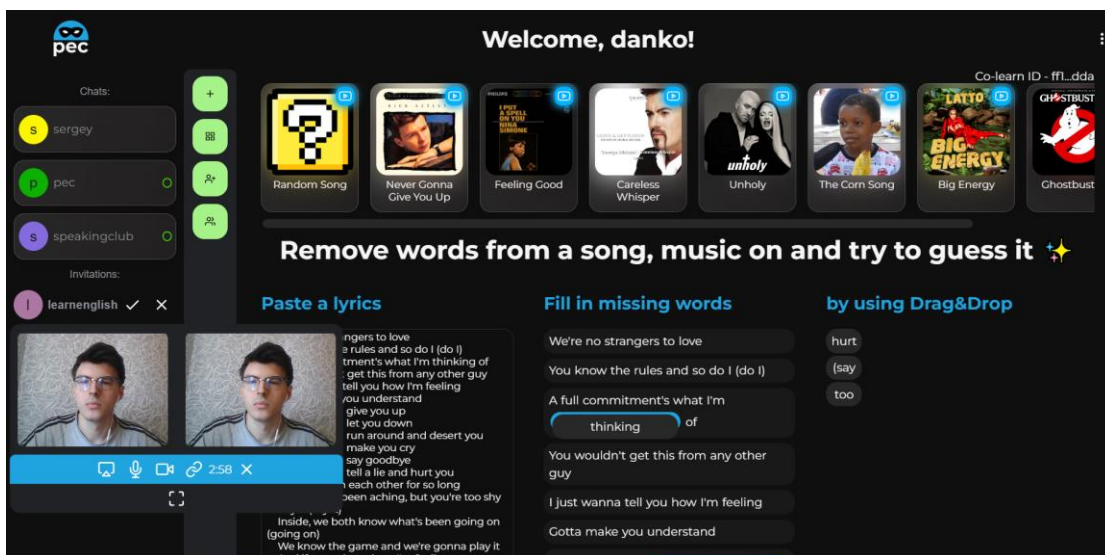


Рисунок 4.21 – Процес перетаскування слів

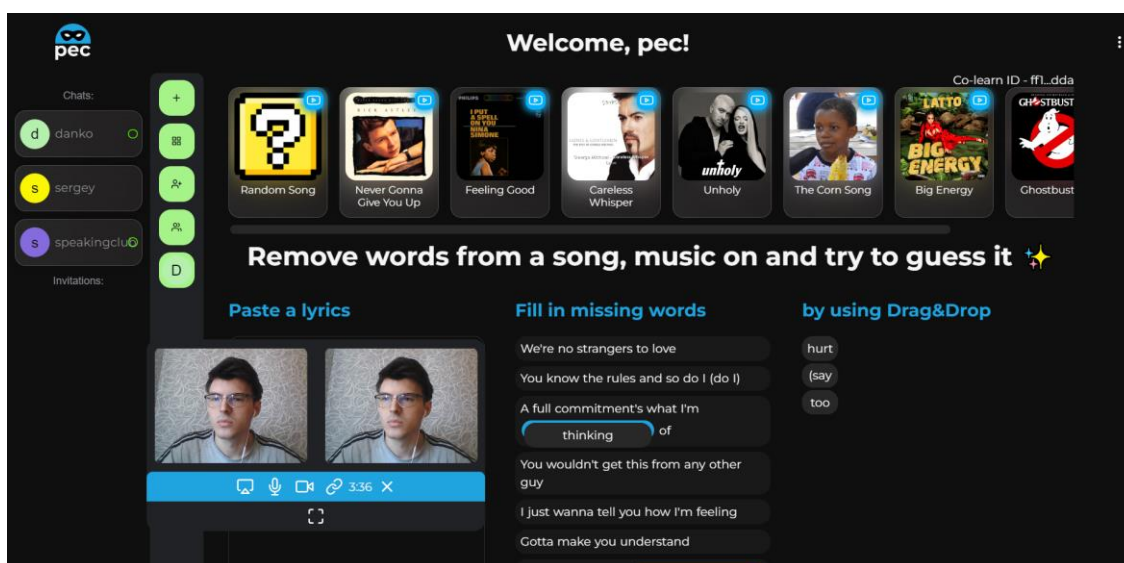


Рисунок 4.22 – Зміни в усіх учасників кімнати

Передбачена оцінка результатів, результати зберігаються для подальшої обробки (рис. 4.23).

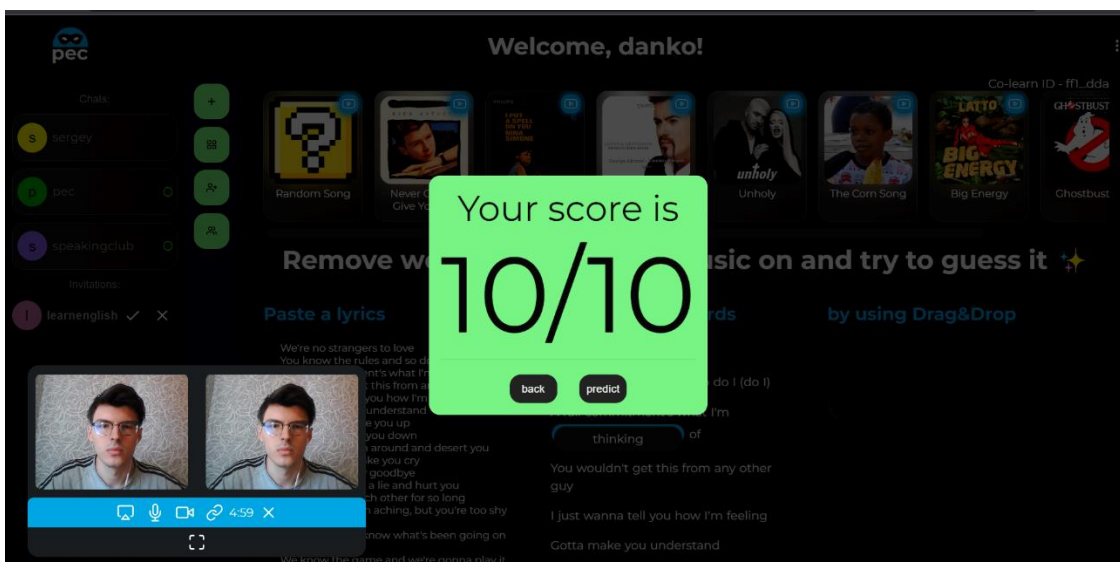


Рисунок 4.23 – Оцінка результатів

Якщо користувач помилився то даний інструментарій покаже правильну відповідь та надасть посилання для ознайомлення з помилковим або пропущеним словом (рис. 4.24).

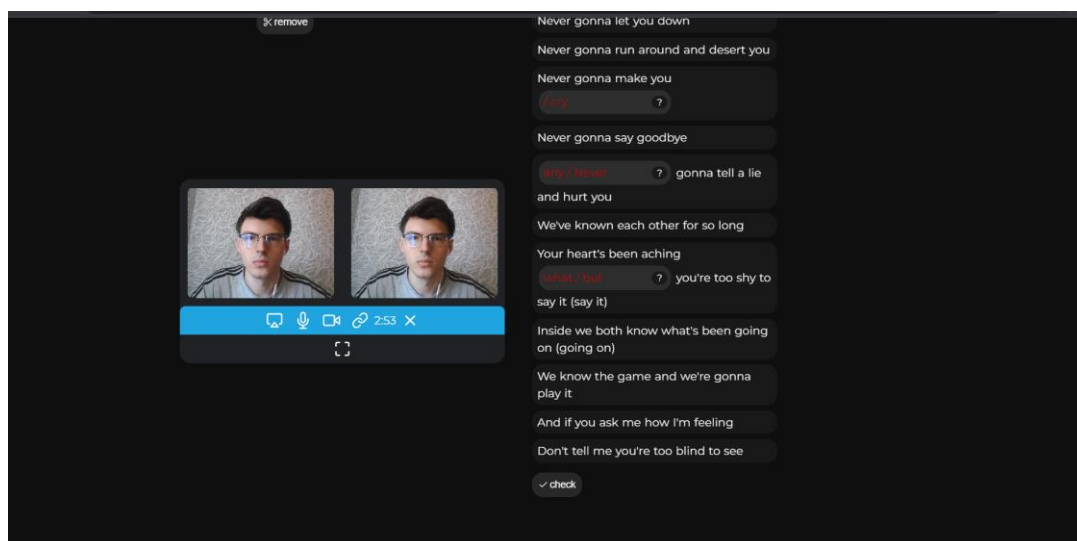


Рисунок 4.24 – Показ помилок

На основі даних, що зберігаються, розроблено прогнозування успішності навчання користувача за допомогою моделі ARIMA (рис. 4.25). Для прогнозування результатів за певний період використовується понад ста записів користувача. Пропущенні записи змінюються на медіанне значення, вони можуть виникнути у разі пропуску занять за певний день. На вхід потрапляє інформація про дату та

кількість відповідей, що формується підбиваючи середнє значення правильних відповідей за день.

Слід зазначити, що даний функціонал не ставить задачі точно показати оцінку користувача. Прогнозування призначено для того щоб показати користувачу можливі його результати та досягнення якщо він буде займатися в такому ж темпі.

Дані користувача обробляються автоматично та додаються до попередніх даних.

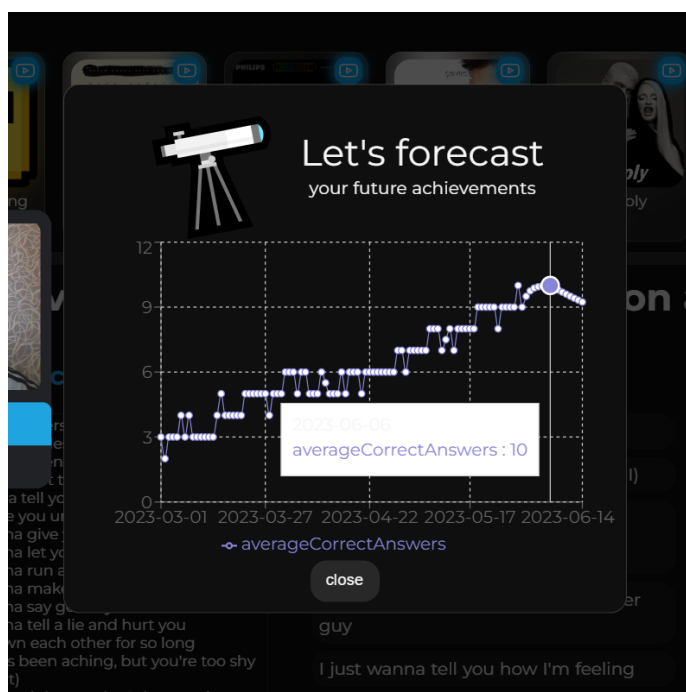


Рисунок 4.25 – Прогнозування результатів користувача

На рисунку 4.26 зображено орієнтований набір даних для прогнозування за допомогою моделі ARIMA.

```

date: "2023-03-01", averageCorrectAnswers: 3,
date: "2023-03-02", averageCorrectAnswers: 2,
date: "2023-03-03", averageCorrectAnswers: 3,
date: "2023-03-04", averageCorrectAnswers: 3,
date: "2023-03-05", averageCorrectAnswers: 3,
date: "2023-03-06", averageCorrectAnswers: 4,
date: "2023-03-07", averageCorrectAnswers: 3,
date: "2023-03-08", averageCorrectAnswers: 4,
date: "2023-03-09", averageCorrectAnswers: 3,
date: "2023-03-10", averageCorrectAnswers: 3,
date: "2023-03-12", averageCorrectAnswers: 3,
date: "2023-03-13", averageCorrectAnswers: 3,
date: "2023-03-14", averageCorrectAnswers: 3,
date: "2023-03-15", averageCorrectAnswers: 4,
date: "2023-03-16", averageCorrectAnswers: 5,
date: "2023-03-17", averageCorrectAnswers: 4,
date: "2023-03-18", averageCorrectAnswers: 4,
date: "2023-03-19", averageCorrectAnswers: 4,
date: "2023-03-20", averageCorrectAnswers: 4,
    
```

Рисунок 4.26 – Набір даних

Представлено основні колекції бази даних, що використовуються в вебзастосунку для збереження даних (рис. 4.27).

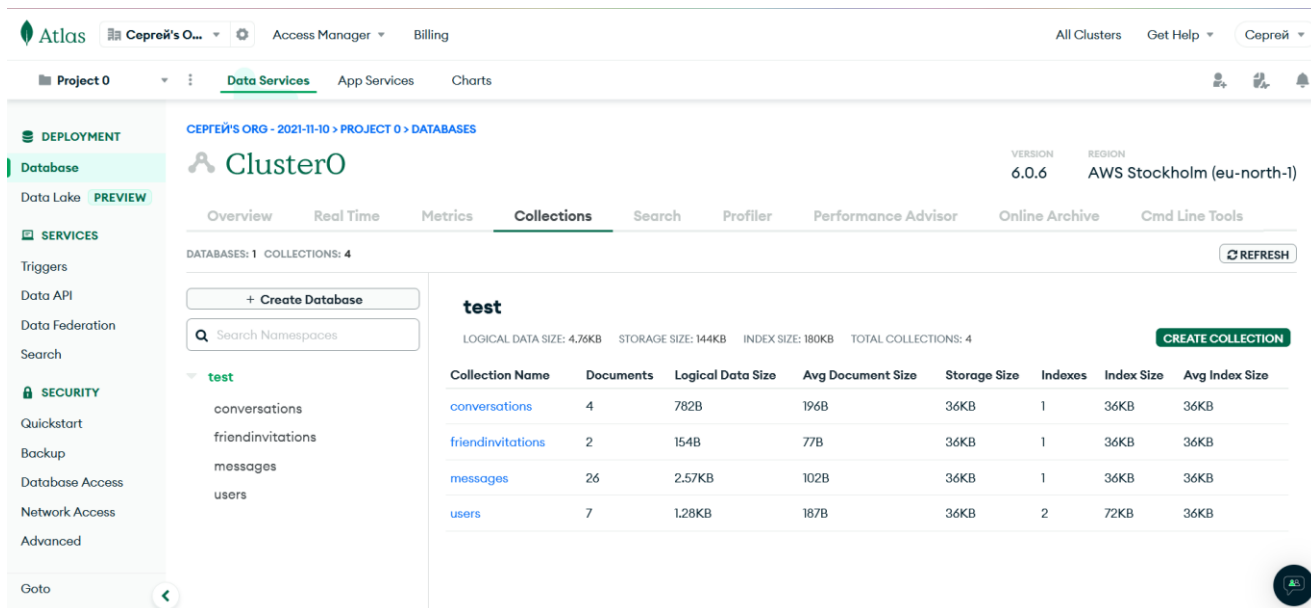


Рисунок 4.27 – База даних на основі MongoDB

Передбачено адміністративну панель для серверу сигналізації (рис. 4.28). Адміністративна панель дозволяє розглянути інформацію про активних клієнтів та інформацію про з'єднання.

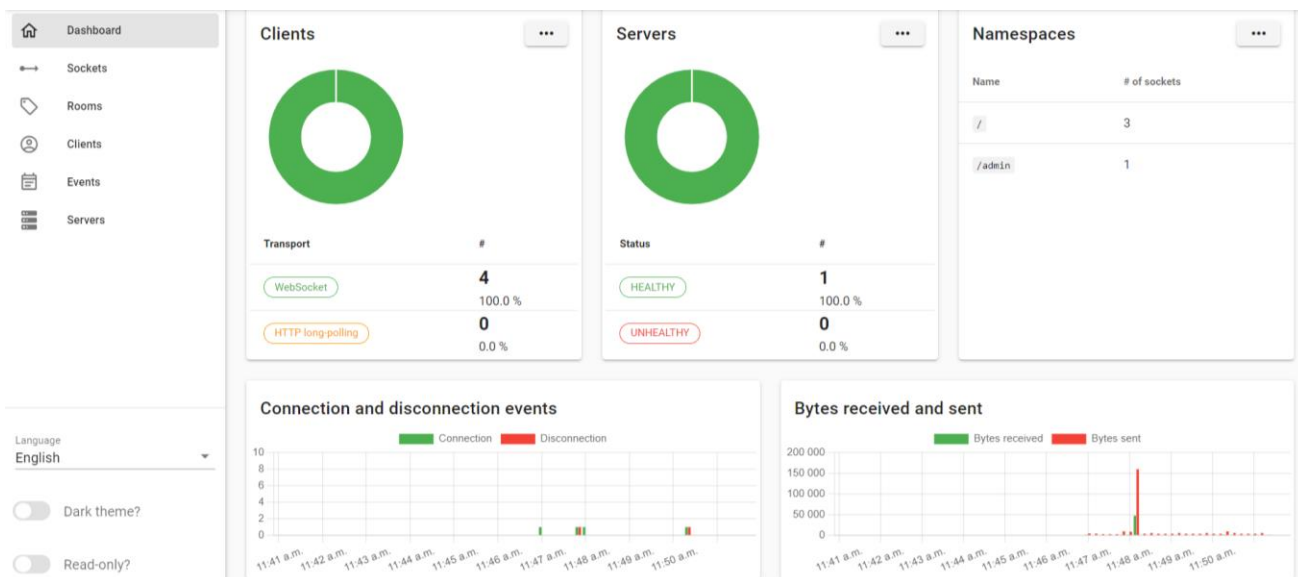


Рисунок 4.28 – Адміністративна панель

4.2 Огляд ефективності

Для оцінки ефективності та отримання статистичних даних використовується метод `getStats`, що надається в `RTCPeerConnection`. За допомогою цього методу доступні різноманітні дані про сеанс WebRTC (рис. 4.29). Кожен звіт проходить циклічно, а його ідентифікатор, тип і конкретна статистика реєструються кожну секунду. Для даного дослідження використовувався процесор Intel(R) Core(TM) i5-10300H 2.50 GHz та RAM – 16 ГБ.

```
Raw report:
{id: 'IT01V770831662', timestamp: 1686304359046, type: 'inbound-rtp', codecId: 'CIT01_96', kind: 'video', ...}
  bytesReceived: 274920
  codecId: "CIT01_96"
  decoderImplementation: "libvpx"
  firCount: 0
  frameHeight: 240
  frameWidth: 320
  framesAssembledFromMultiplePackets: 130
  framesDecoded: 145
  framesDropped: 0
  framesPerSecond: 30
  framesReceived: 145
  freezeCount: 0
  googTimingFrameInfo: "3940358451,12227578,12227578,12227579,12227580,12227580,12227578,12227581,12227581"
  headerBytesReceived: 8168
  id: "IT01V770831662"
  jitter: 0.003
  jitterBufferDelay: 3.173
  jitterBufferEmittedCount: 144
  keyFramesDecoded: 1
  kind: "video"
  lastPacketReceivedTimestamp: 1686304359023
  mediaType: "video"
  mid: "1"
  nackCount: 0
  packetsLost: 0
  packetsReceived: 322
  pauseCount: 0
  pliCount: 0
  powerEfficientDecoder: false
  qpSum: 3872
  ssrc: 770831662
  timestamp: 1686304359046
  totalAssemblyTime: 0.050553
  totalDecodeTime: 0.06562899999999999
  totalFreezesDuration: 0
  totalInterFrameDelay: 4.757
  totalPausesDuration: 0
  totalProcessingDelay: 1.070598
  totalSquaredInterFrameDelay: 0.16270499999999998
  trackIdentifier: "99bae2e0-4ebc-4eb0-a167-8dac103efe3c"
  transportId: "T01"
  type: "inbound-rtp"
  [[Prototype]]: Object
```

Рисунок 4.29 – Звіт про сеанс WebRTC

Дослідимо пропускну здатність в залежності від кількості учасників. Цей показник потокового передавання описує кількість даних, які можна передати за один раз. Також, розглянемо показники, що впливають на пропускну здатність.

Одним з таких показників є бітрейт, що показує кількість даних, що передаються у момент часу. Тобто, вищі налаштування бітрейту отримують потоки з вищою роздільною здатністю.

Для визначення швидкості з якою відображається кожен кадр використовуємо показник частоти кадрів. Цей показник використовує одиниці кадрів за секунду.

Показник роздільної здатності відео – це вимірювання ширини та висоти відео в пікселях. Відео з вищою роздільною здатністю вимагає більшої пропускної здатності для потокового передавання.

За допомогою звіту сформуємо потрібні дані для дослідження (рис. 4.30).

Bandwidth: 3.356571 Mbps	webRTCHandler.js:133
15 fps	webRTCHandler.js:108
Resolution: 480X640	webRTCHandler.js:111
Packet Loss: 0%	webRTCHandler.js:117
Bitrate: 1.771 Mbps	webRTCHandler.js:120

Рисунок 4.30 – Формування даних

Дані збираються на початку сеансу відеоконференції та при завершенні з'єднання, формуються в Excel файл для подальшої обробки (рис. 4.31).

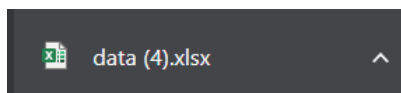


Рисунок 4.31 – Збереження даних

Сітчаста топологія передбачає багатокористувацьке з'єднання. Для передачі даних між учасниками можна створити багато зв'язків. Всі однорангові пристрої з'єднуються між собою для одночасної передачі даних з різних пристроїв. На рисунку 4.32 показано залежність кількості зв'язків від кількості учасників.

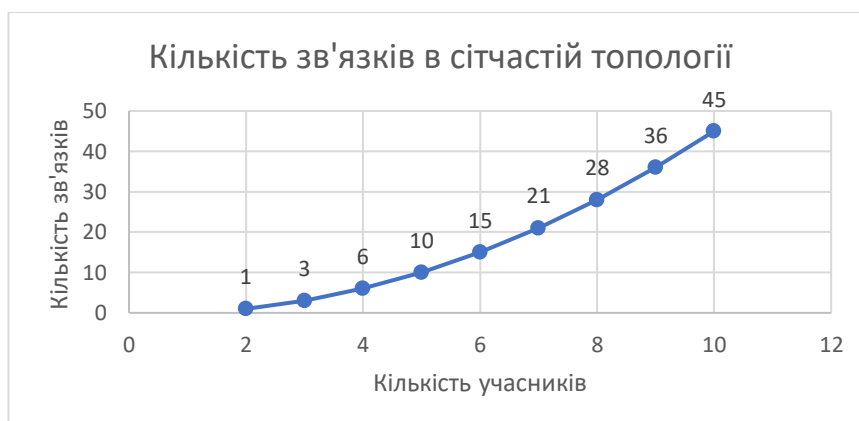


Рисунок 4.32 – Залежність кількості зв'язків від учасників

Сітчаста топологія використовує багато з'єднань між користувачами для передачі даних. Топологія справляється з високим навантаженням за допомогою одночасного отримання даних. Високе навантаження виникає у процесі кодування та декодування кожного з'єднання, що впливає на продуктивність центрального процесору. Отже, з збільшенням кількості користувачів збільшується навантаження на центральний процесор (рис. 4.33). Дане обмеження впливає лише на учасників з мінімальними характеристиками центрального процесору.

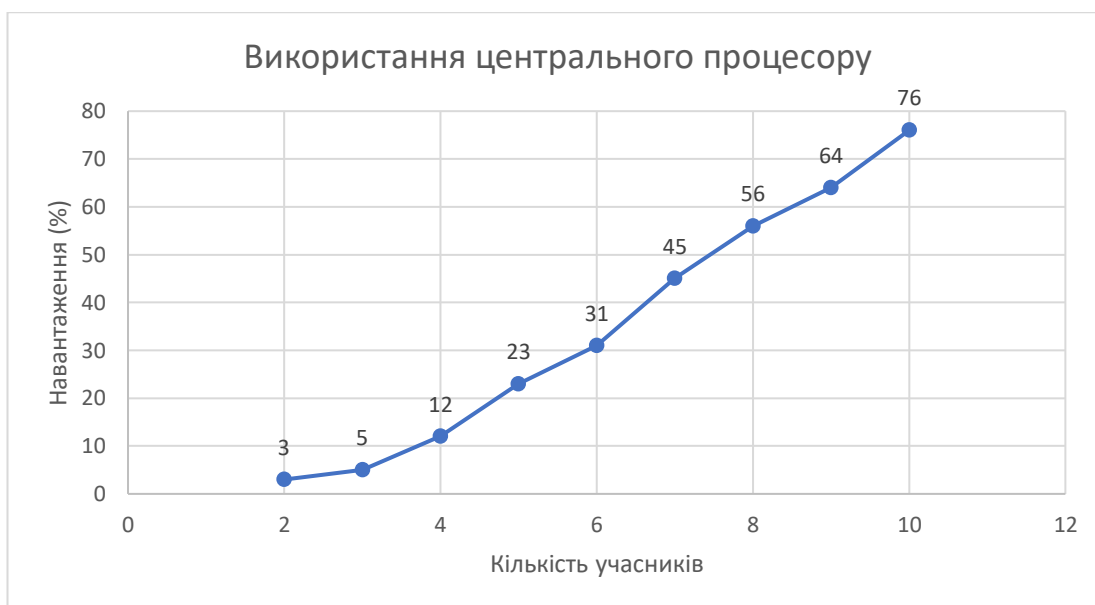


Рисунок 4.33 – Використання центрального процесору

Для досягнення оптимальної передачі даних важливо враховувати доступну пропускну здатність і відповідно регулювати бітрейт, дотримуючись балансу між якістю та ефективним використанням середовища передачі. Бітрейт і пропускну здатність залежать один від одного. Бітрейт обмежений пропускну здатністю, яка може бути використана для конкретного каналу зв'язку. Випадання кадрів, спотворення аудіо або нижча роздільна здатність можуть виникати через втрату або погіршення якості даних, якщо бітрейт перевищує пропускну здатність. Дослідимо дані показники для двох учасників з'єднання (рис. 4.34).

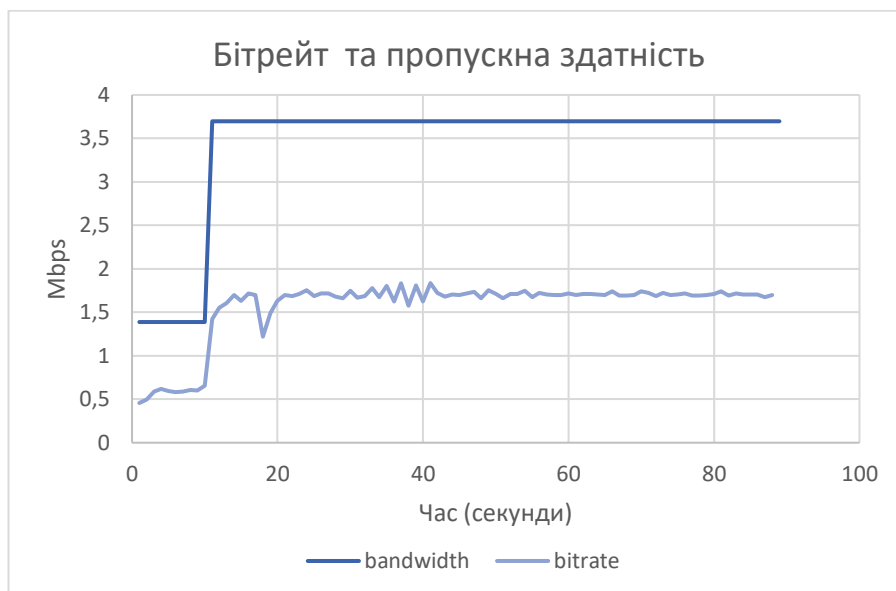


Рисунок 4.34 – Огляд бітрейту та пропускної здатності

На якість комунікації у реальному часі можуть вплинути обмеження пропускної здатності при збільшенні кількості учасників. Учасники можуть зіткнутися із затримками, втратою пакетів або погіршенням якості аудіо та відео, при обмеженій пропускній здатності. Розглянемо залежність пропускної здатності від кількості учасників (рис. 4.35).



Рисунок 4.35 – Показник пропускної здатності

На основі результатів дослідження було сформовано якість досвіду під час взаємодії з вебзастосунком (табл. 4.1).

Таблиця 4.1 – Оцінка якості відео та аудіо

Кількість клієнтів	Тривалість дослідження (хв.)	Якість звуку	Якість відео
2-5	2-5	Відмінно	Відмінно
6-10	2-5	Допустимо	Допустимо
11-15	2-5	Не допустимо	Не допустимо

Висновки до розділу 4

В результаті було розроблено вебзастосунок для вивчення іноземних мов з використання сучасних технологій. Вебзастосунок спрямований на створення віртуальних кімнат, де можна створити зв'язок для відеоконференції та ознайомитись з додатковим інструментарієм для поліпшення навичок говоріння, аудіювання, побудові речень та перекладу. Наявний функціонал для ефективного навчання в кооперативному або індивідуальному режимі. Реалізовано оцінку результатів по додатковим інструментам та прогнозування результатів навчання.

Розглянуто та протестовано ефективність вебзастосунку на основі статистичних даних про сеанс де було досліджено показники пропускну здатності, бітрейту, частоти кадрів та роздільної здатності відео. Визначено їх оптимальні показники та залежність навантаження від кількості учасників.

Дані було проаналізовано та структурно сформовано, що дозволило визначити якість досвіду під час взаємодії з вебзастосунком.

ВИСНОВКИ

В ході виконання бакалаврської кваліфікаційної роботи було розроблено вебзастосунок для вивчення іноземних мов.

Враховуючи всю необхідну інформацію даної предметної області вирішено центральну проблему, яка полягала в тому, що чинні методи та інструменти вивчення іноземних мов не забезпечували повноцінного та ефективного навчання.

Вдалося досягти поставленої мети – підвищення рівня комунікації та вивчення іноземної мови з використанням сучасних технологій, що забезпечують можливість створення віртуальних кімнат спрямованих на імітацію розмовних ситуацій. Учасники можуть брати участь у кооперативних заходах для вивчення у групах за допомогою ряду інструментів щоб разом працювати над вправами та отримувати зворотний зв'язок або в індивідуальних заходах, задання яких сфокусовані на персональне вивчення мов. Такий підхід покращує зацікавленість та мотивацію користувачів до вивчення іноземної мови та поліпшує розмовні навички в безпечному та сприятливому середовищі з подальшим аналізом результатів.

В повному обсязі реалізовано завдання для досягнення поставленої мети, що полягали в розробленні та тестуванні вебзастосунку для вивчення іноземних мов. Розроблено віртуальні кімнати, що дозволяють створювати відеоконференції та використовувати інструменти для вивчення іноземних мов у реальному часі, передбачено систему оцінювання знань користувачів з можливістю прогнозування результатів навчання, виконано огляд ефективності та сформовано, якість досвіду вебзастосунку.

Надалі вебзастосунок можна поліпшити додавши більший спектр інструментів для кооперативного або індивідуального вивчення іноземних мов та підвищити якість сеансу зв'язку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ткач А. В. Електронне навчання: сучасні тренди. 2017. URL: <http://repository.sspu.sumy.ua/handle/123456789/2621> (дата звернення: 15.05.2023).
2. Сотер М. В. Відеоконференція як засіб іншомовної підготовки студентів. Mountain School of Ukrainian Carpaty. 2020. № 22. С. 135–138. URL: <https://doi.org/10.15330/msuc.2020.22.135-138> (дата звернення: 08.06.2023).
3. M. E. Saputri, F. N. Utami and D. Sari, "The Effectiveness of E-Learning Service Quality in Influencing E-Learning Student Satisfaction and Loyalty at Telkom University," 2022 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS), Bandung, Indonesia, 2022, pp. 01-05, doi: 10.1109/ICADEIS56544.2022.10037454.
4. Борисенко К. Засоби навчання студентів іноземної мови у режимі дистанційного навчання. Сучасний стан та перспективи розвитку науки. URL: <https://doi.org/10.36074/18.12.2020.v5.05> (дата звернення: 08.06.2023).
5. Yu, L.-T. The Effect of Videoconferencing on Second-Language Learning: A Meta-Analysis. Behav. Sci. 2022, 12, 169. <https://doi.org/10.3390/bs12060169>
6. S. ÖZTÜRK et al., "Functionality, Performance and Usability Tests of WebRTC Based Video Conferencing Products," 2021 15th Turkish National Software Engineering Symposium (UYMS), Izmir, Turkey, 2021, pp. 1-6, doi: 10.1109/UYMS54260.2021.9659594.
7. D. Tang and L. Zhang, "Audio and Video Mixing Method to Enhance WebRTC," in IEEE Access, vol. 8, pp. 67228-67241, 2020, doi: 10.1109/ACCESS.2020.2985412.
8. C. Cola and H. Valean, "On multi-user web conference using WebRTC," 2014 18th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 2014, pp. 430-433, doi: 10.1109/ICSTCC.2014.6982454.

9. G. Suciu, S. Stefanescu, C. Beceanu and M. Ceaparu, "WebRTC role in real-time communication and video conferencing," 2020 Global Internet of Things Summit (GIoTS), Dublin, Ireland, 2020, pp. 1-6, doi: 10.1109/GIOTS49054.2020.9119656.

10. García, B., Gallego, M., Gortázar, F. et al. Understanding and estimating quality of experience in WebRTC applications. *Computing* 101, 1585–1607 (2019). <https://doi.org/10.1007/s00607-018-0669-7>

11. R. A. Kirmiziloglu and A. M. Tekalp, "Multi-Party WebRTC Services Using Delay and Bandwidth Aware SDN-Assisted IP Multicasting of Scalable Video Over 5G Networks," in *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 1005-1015, April 2020, doi: 10.1109/TMM.2019.2937170.

12. Online Language Learning Market Worth. Meticulous Research. URL: <https://www.meticulousresearch.com/pressrelease/286/online-language-learning-market-2029> (дата звернення: 08.05.2023).

13. Karam, S., & Abdulrahman, B. (2022). Using Socket.io Approach for Many-to-Many Bi-Directional Video Conferencing. *AL-Rafidain Journal of Computer Sciences and Mathematics*, 16(1), 81-86. doi: 10.33899/csmj.2022.174411

14. N. M. Edan, A. Al-Sherbaz and S. Turner, "WebNSM: A Novel Scalable WebRTC Signalling Mechanism for Many-to-Many Video Conferencing," 2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC), San Jose, CA, USA, 2017, pp. 27-33, doi: 10.1109/CIC.2017.00015.

15. VIDEO CONFERENCING USING WEBRTC / J. Kaushal et al. *International Journal of Engineering Applied Sciences and Technology*. 2020. Vol. 5, no. 4. P. 539–545, doi: 10.33564/ijeast.2020.v05i04.084

16. Mesh. URL: <https://bloggeek.me/webrtcglossary/mesh/> (дата звернення: 18.05.2023).

17. Mesh Usecases. URL: <https://mobidev.biz/blog/webrtc-app-development-challenges-use-cases-future> (дата звернення: 15.05.2023).

18. MCU. URL: <https://bloggeek.me/webrtc-multipoint-large-groups/> (дата звернення: 20.05.2023).

19. SFU. URL: <https://bloggeek.me/webrtc-multiparty-video-alternatives/> (дата звернення: 27.05.2023).
20. Мічківський С. М., Прігунов О. В., Римар П. В. М. Системи та методи прийняття рішень: методичні вказівки. Вінниця, ДонНУ імені Василя Стуса, 2019, 76 с.
21. Rosaria de F.S.M. Russo, Roberto Camanho, Criteria in AHP: A Systematic Review of Literature, *Procedia Computer Science*, Volume 55, 2015, Pages 1123-1132, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2015.07.081>.
22. S. Ahmed and Q. Mahmood, "An authentication based scheme for applications using JSON web token," 2019 22nd International Multitopic Conference (INMIC), Islamabad, Pakistan, 2019, pp. 1-6, doi: 10.1109/INMIC48123.2019.9022766.
23. Freeman, A. (2019). Using a Redux Data Store. In: *Pro React 16*. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-4451-7_19
24. Bork, Dominik, Antonio Garmendia, and Manuel Wimmer. "Towards a Multi-Objective Modularization Approach for Entity-Relationship Models." *ER Forum/Posters/Demos*. 2020.
25. Haga, Steve, Wei-Ming Ma, and William Shanchung Chao. "Structure-Behavior Coalescence Method for Formal Specification of UML 2.0 Sequence Diagrams." *J. Comput. Sci. Eng.* 15.4 (2021): 148-159.

ДОДАТОК А

Лістинг програмного коду

```
const express = require("express");
const http = require("http");
const cors = require("cors");
const mongoose = require("mongoose");
require("dotenv").config();

const socketServer = require("./socketServer");
const authRoutes = require("./routes/authRoutes");
const friendInvitationRoutes = require("./routes/friendInvitationRoutes");

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

const app = express();
app.use(express.json());
app.use(cors());

app.use("/api/auth", authRoutes);
app.use("/api/friend-invitation", friendInvitationRoutes);

const server = http.createServer(app);
socketServer.registerSocketServer(server);

mongoose
  .connect(process.env.MONGODB_URI)
  .then(() => {
    server.listen(PORT, () => {
      console.log(`Server: ${PORT}`);
    });
  })
  .catch((err) => {
    console.log("
Error Establishing a Database Connection
");
    console.error(err);
  });

const authSocket = require("./middleware/authSocket");
const newConnectionHandler = require("./socketHandlers/newConnectionHandler");
const disconnectHandler = require("./socketHandlers/disconnectHandler");
const directMessageHandler = require("./socketHandlers/directMessageHandler");
const directChatHistoryHandler = require("./socketHandlers/directChatHistoryHandler");
const roomCreateHandler = require("./socketHandlers/roomCreateHandler");
const roomJoinHandler = require("./socketHandlers/roomJoinHandler");
const roomLeaveHandler = require("./socketHandlers/roomLeaveHandler");
```

```
const roomInitializeConnectionHandler =
require("./socketHandlers/roomInitializeConnectionHandler");
const roomSignalingDataHandler = require("./socketHandlers/roomSignalingDataHandler");

const serverStore = require("./serverStore");
const { instrument } = require("@socket.io/admin-ui");

const registerSocketServer = (server) => {
  const io = require("socket.io")(server, {
    cors: {
      origin: ["http://localhost:3000", "https://admin.socket.io"],
      credentials: true,
    },
  });
  instrument(io, {
    auth: false,
    mode: "development",
  });

  serverStore.setSocketServerInstance(io);

  io.use((socket, next) => {
    authSocket(socket, next);
  });

  const emitOnlineUsers = () => {
    const onlineUsers = serverStore.getOnlineUsers();
    io.emit("online-users", { onlineUsers });
  };

  io.on("connection", (socket) => {
    console.log("user connected");
    console.log(socket.id);

    newConnectionHandler(socket, io);
    emitOnlineUsers();

    socket.on("direct-message", (data) => {
      directMessageHandler(socket, data);
    });

    socket.on("direct-chat-history", (data) => {
      directChatHistoryHandler(socket, data);
    });

    socket.on("room-create", () => {
      roomCreateHandler(socket);
    });
  });
};
```

```
socket.on("room-join", (data) => {
  roomJoinHandler(socket, data);
});

socket.on("room-leave", (data) => {
  roomLeaveHandler(socket, data);
});

socket.on("conn-init", (data) => {
  roomInitializeConnectionHandler(socket, data);
});

socket.on("conn-signal", (data) => {
  roomSignalingDataHandler(socket, data);
});

socket.on("disconnect", () => {
  disconnectHandler(socket);
});

setInterval(() => {
  emitOnlineUsers();
}, [1000 * 8]);

module.exports = {
  registerSocketServer,
};

const express = require("express");
const router = express.Router();
const authControllers = require("../controllers/auth/authControllers");
const Joi = require("joi");
const validator = require("express-joi-validation").createValidator({});
const auth = require("../middleware/auth");
const registerSchema = Joi.object({
  username: Joi.string().min(3).max(12).required(),
  password: Joi.string().min(6).max(12).required(),
  mail: Joi.string().email().required(),
});

const loginSchema = Joi.object({
  password: Joi.string().min(6).max(12).required(),
  mail: Joi.string().email().required(),
});

router.post(
```

```
"/register",
  validator.body(registerSchema),
  authControllers.controllers.postRegister
);
router.post(
  "/login",
  validator.body(loginSchema),
  authControllers.controllers.postLogin
);

module.exports = router;

const express = require("express");
const router = express.Router();
const Joi = require("joi");
const validator = require("express-joi-validation").createValidator({});
const auth = require("../middleware/auth");
const friendInvitationControllers =
  require("../controllers/friendInvitation/friendInvitationControllers");

const postFriendInvitationSchema = Joi.object({
  targetMailAddress: Joi.string().email(),
});

const inviteDecisionSchema = Joi.object({
  id: Joi.string().required(),
});

router.post(
  "/invite",
  auth,
  validator.body(postFriendInvitationSchema),
  friendInvitationControllers.controllers.postInvite
);

router.post(
  "/accept",
  auth,
  validator.body(inviteDecisionSchema),
  friendInvitationControllers.controllers.postAccept
);

router.post(
  "/reject",
  auth,
  validator.body(inviteDecisionSchema),
  friendInvitationControllers.controllers.postReject
);
module.exports = router;
```