МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІМ. Ю.М. ПОТЕБНІ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
АВТОМАТИЗОВАНИХ СИСТЕМ

Кваліфікаційна робота

другий (магістерський)
(рівень вищої освіти)

на тему Дослідження принципів створення і інтеграції відеострімінгових сервісів на Web-сайті

Виконав: студент 2 курсу, групи 8.1210-іпз
спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)
освітньої програми Інженерія програмного забезпечення
(код і назва освітньої програми)
Л. М. Правдивець
(ініціали та прізвище)
Керівник доцент, к.т.н, Л. О. Лимаренко
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)
Рецензент директор ТОВ «Альтер Віжен Груп»
В.С. Тряпічко
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя
2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІМ. Ю.М. ПОТЕБНІ

Кафедра _____ програмного забезпечення автоматизованих систем
Рівень вищої освіти _____другий (магістерський)_____
Спеціальність ___121_Інженерія програмного забезпечення_____
(код та назва)
Освітня програма ___Інженерія програмного забезпечення
(код та назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри_В.Г. Вербицький
"_01_" __вересня_____2021_року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

__Правдивець Лілії Миколаївні__
(прізвище, ім'я, по батькові)
1.Тема роботи __Дослідження принципів створення і інтеграції відеострімінгових сервісів на Web-сайт_____

керівник роботи ___Лимаренко Юлія Олексіївна, доцент_____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом ЗНУ від ___"30" червня 2021 року № 974-с_____

2. Строк подання студентом кваліфікаційної роботи_____30.11.2020_____
3. Вихідні дані магістерської роботи
- комплект нормативних документів ;
- технічне завдання до роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
- огляд та збір літератури стосовно теми кваліфікаційної роботи;
- огляд та аналіз існуючих рішень та аналогів;
- дослідження проблеми потокової передачі відео та розробка методів її вирішення;
- створення програмного продукту та його опис;
- перелік вимог для роботи програми;
- дослідження поставленої проблеми та розробка висновків та пропозицій.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
___слайдів презентації

6. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата Завдання прийняв

7. Дата видачі завдання 01.09.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1	Аналіз предметної області	02.09-10.09.21	виконано
2	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	11.09-12.09.2021	виконано
3	Аналіз існуючих методів рішення	13.09-17.09.20	виконано
4	Дослідження області потокової передачі відео	18.09-24.09.20	виконано
5	Узгодження подальших дій з науковим керівником	25.09-26.09.20	виконано
6	Аналіз теоретичних відомостей	27.09-15.10.20	виконано
7	Проектування інтерфейсу web-сайту для проведення відеоконференцій	15.10-23.10.20	виконано
8	Узгодження інтерфейсу з науковим керівником	23.10-24.10.20	виконано
9	Реалізація функціоналу потокової передачі відео	25.10-14.11.20	виконано
10	Представлення отриманих результатів науковому керівнику і узгодження плану подальшого дослідження	15.11-16.11.20	виконано
11	Реалізація функціоналу проведення відеоконференцій	16.11-23.11.20	виконано
12	Проведення аналізу можливостей розроблених програмних застосунків	24.11-09.12.21	виконано
13	Оформлення звіту	10.12-28.12.21	виконано

Студент _____ Л.М. Правдивець
(підпис) (прізвище та ініціали)

Керівник роботи _____ Ю.О. Лимаренко
(підпис) (прізвище та ініціали)

Нормоконтроль пройдено

Нормоконтролер _____ І.А. Скрипник
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Сторінок: 85

Рисунків: 35

Таблиць: 3

Джерел: 22

Правдивець Лілія Миколаївна. Дослідження принципів створення і інтеграції відеострімінгових сервісів на web-сайт.

Кваліфікаційна робота для здобуття ступеня вищої освіти магістра за спеціальністю 121 — Інженерія програмного забезпечення, науковий керівник Ю. О. Лимаренко. Інженерний навчально-науковий інститут Запорізького національного університету.

Метою роботи є проаналізувати існуючі відеострімінгові платформи, дослідити та порівняти інструменти для створення відеострімінгової платформи та розробити власний веб-сайт с інтеграцією Twilio API.

Результатом роботи є веб-сайт для проведення відеоконференцій за допомогою інтегрованого Twillio API.

Ключові слова: *vue.js, компонент, framework, html, css, javascript, firebase, twilio api, node.js*

SUMMARY

Pages: 85

Figures: 35

Tables: 3

Sources: 22

Pravdyvets Liliia. Research of principles for creation and integration of video streaming services on the Web-site.

Qualification work for higher Master's degree in specialty 121 — Software Engineering, supervisor Yuliya Lyimarenko. Engineering Educational Scientific Institute of Zaporizhia National University.

The aim of the work is to analyze existing video streaming platforms, explore and compare tools for creating a video streaming platform and develop your own website with Twilio API integration.

The result is a website for video conferencing using the integrated Twillio API.

Keywords: vue.js, component, framework, html, css, javascript, firebase, twilio api, node.js

ЗМІСТ

ВСТУП

Актуальність теми

Актуальність відеострімінга обумовленна не тільки інформаційним прогресом, але і людськими потребами. Складно уявити сучасний світ без можливості швидкого зв'язку не тільки по через слухавку телефону, але й за допомогою відеозв'язку. Можливість відеозв'язку говорить не лише про способи комунікацій між людьми, але й про швидку, надійну передачу важкої, складної інформації. Потокова передача мультимедійної інформації вийшла на рівень, коли вона не просто існує як єдиний спосіб передачі, але й поліпшується, через що виникають різні методи відправки даних, що конкурують між собою.

Можливість відправки мультимедійної інформації стимулює розвиток усіх галузей життя. Наука, бізнес, розваги — це все зараз потребує комунікацій, швидкого отримання та передачі інформації і потокова передача надає таку можливість.

Пряма трансляція — це еволюція телевізійних передач. Причини, через які телевізійне телебачення справило такий великий вплив, полягали в тому, що воно надало глядачам сучасний, інформативний та доступний контент. Хоча це все ще популярно, з моменту трансляції в Інтернеті передплата на телебачення зменшується. Його зниження можна пояснити зростанням потокової передачі, зокрема зміною поведінки глядачів. Інтернет-індустрія відео показала глядачам, що вони повинні контролювати, щоб дивитись, що вони хочуть, коли вони цього хочуть.

Однією з причин, чому пряма трансляція може допомогти охопити більше людей, є те, що платформи надають перевагу живому контенту.

Незважаючи на те, що глядачі все ще мають обмеження в часі на пряму трансляцію, це більш гнучко, ніж на телевізорі. Вони можуть дивитись на своїх телефонах, і в більшості випадків вони також можуть переглядати запис відео в прямому ефірі пізніше. Однією з основних причин, чому пряма трансляція

настільки важлива для брендів та приватних осіб, є рівень взаємодії та залучення, який вона пропонує. Жодна інша платформа або маркетингова стратегія не допускає такого рівня взаємодії. Пряма трансляція також має найвищий рівень залучення всіх типів контенту.

Мета і завдання дослідження

Метою роботи є проаналізувати існуючі відеострімінгові платформи, дослідити та порівняти інструменти для створення відеострімінгової платформи та розробити власний веб-сайт с інтеграцією Twillio API.

Об’єкт дослідження

Об’єктом дослідження є системи відеострімінгу.

Предмет дослідження

Предметом дослідження є відеострімінг.

Методи дослідження

Для отримання необхідних результатів дослідження використовувалися певні методи:

- Знаходження та обробка літератури з розробки відеострімінгового сервісу
- Аналіз існуючих рішень.
- Метод порівняння

Наукова новизна одержаних результатів

Порівняльний аналіз інструментів для створення відеострімінгу та існуючих відеострімінгових платформ.

Практичне значення одержаних результатів

Відеострімінговий сервіс з інтегрованим Twillio API.

Апробація одержаних результатів

Підсумки дослідження були представлені на XIV університетській науково-практичній конференції студентів, аспірантів, докторантів і молодих вчених Запорізького національного університету «Молода наука-2021» [1], а також на І Всеукраїнській науково-практичній конференції здобувачів вищої

освіти, аспірантів та молодих вчених Інженерного навчально-наукового інституту Запорізького національного університету [2].

Глосарій

Фреймворк — це платформа для розробки програмних додатків, що визначає їх майбутню структуру за для полегшення та прискорення процесу розробки.

Npm (node package manager) — менеджер пакетів Node.js. Використовується для завантаження пакетів із хмарного сервера npm, на хмарні сервера. Менеджер npm складається з інтерфейсу командного рядка (CLI) та онлайн-репозиторіїв, що містять JavaScript пакети.

Visual Studio Code — редактор вихідного коду, розроблений компанією Microsoft для операційних систем Windows, Linux і macOS. Зручний редактор коду для кросплатформної розробки веб-сторінок і хмарних додатків. Включає в себе відладчик, засоби для рефакторинга, інструменти для роботи з Git, підсвічування синтаксису та IntelliSense. Має можливості для кастомізації: теми оформлення, гарячі клавіши і файли конфігурації.

Vue.js — це прогресивний фреймворк на мові програмування JavaScript з відкритим вихідним кодом для створення користувацьких інтерфейсів.

HTML (HyperText Markup Language — «мова гіпертекстової розмітки») — стандартизована мова розмітки документів для перегляду веб-сторінок в браузері.

CSS (англ. Каскадні таблиці стилів) — формальна мова, що застосовується до елементів документ та використовується для стилізації зовнішнього вигляд сторінок.

JavaScript — це мова програмування, що дозволяє зробити веб-сторінку інтерактивною за допомогою реагування на різні дії користувача.

Firebase — це хмарний сервіс для розробки, тестування, поширення та монетизації додатків під Android, IOS або Web, що значно спрощує розробку.

Vue компоненти — це перевикористовуємі блоки коду, що містять в собі опис зовнішнього виду частин веб-сайту та реалізацію їх функціональної (програмної) частини. Компоненти допомагають в створенні модульної структури сайту, що зручна в підтримці.

MVVM (Model-View-ViewModel) — це шаблон проектування, що застосовується під час проектування архітектури застосунків (додатків). Дозволяє відокремити логіку додатки від візуальної частини. Патерн є архітектурним – задає загальну архітектуру програми. MVVM складається з трьох компонентів: моделі (Model), моделі представлення (ViewModel) і представлення (View).

CDN (Мережа доставки контенту) — це група географічно розподілених серверів для прискорення доставки контенту за рахунок скорочення шляху передачі мережею.

Encoder (Кодувальник) — це апаратний пристрій або програмний додаток, який підготовлює файл для потокової передачі.

Decoder (Декодувальник) — це апаратний пристрій або програмний додаток, що перетворює кодований цифровий потік в аудіо, відео, субтитри для перегляду в веб-браузері.

Кодеки — це стандарти цифрового стиснення відео і аудіо.

Git — розподілена система керування версіями файлів та спільної роботи над проектами. Git є однією з найефективніших, надійних і високопродуктивних систем керування версіями, що надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні і злитті гілок. Для забезпечення цілісності історії та стійкості до змін заднім числом використовуються криптографічні методи, також можлива прив'язка цифрових підписів розробників до тегів і комітів.

GitHub — один з найбільших та найвідоміших веб-сервісів для хостингу та спільної розробки програмного забезпечення. Базується на системі керування версіями Git і розроблений на Ruby on Rails і Erlang компанією GitHub.

Donate («пожертвування») — це популярна форма пожертвування від підписників, є один із способів заробітку в Інтернеті.

Веб-сервер — це сервер, що отримує з браузера користувачів HTTP запити та повертає HTTP відповіді: сторінки в браузері, картинки, тощо.

White label — концепція, що передбачає виробництво продуктів або послуг однією компанією під брендом іншої компанії, так ніби це виготовила компанія-власник бренду

РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ПОТОКОВОЇ ПЕРЕДАЧІ ВІДЕО

1.1 Що таке відеострімінг

Відеострімінг — це потокова передача мультимедійної інформації (відео та аудіо) віддаленим користувачам у режимі реального часу [3].

1.2 Початок роботи з потоковою передачею даних

Для потокової передачі мультимедійної інформації перш за все потрібно:

1. відео та аудіо джерела;
2. кодувальник;
3. CDN (Мережа доставки контенту);
4. стабільне інтернет-з'єднання.

1.3 Як відбувається потокова передача даних

Кодувальник захоплює зображення з камери і аудіо з мікрофона, стискає і перетворює їх в формат, який підходить для потокової передачі або запису.

Стиснення відбувається за допомогою кодеків. Відеокодек і аудіокодек стискають свою доріжку. Після чого вони синхронізуються і зберігаються в одному медіаконтейнера, який називається форматом. Формат також може містити метадані, прикладом яких можуть бути субтитри, картинки попереднього перегляду.

Після стиснення формати відправляються по мережі.

Аудіо- або відеоплеєр в браузері на клієнтській стороні приймає потік пакетів даних(форматів) і інтерпретує їх як відео або аудіо за допомогою декодувальника. Схема зображена на рисунку 1.

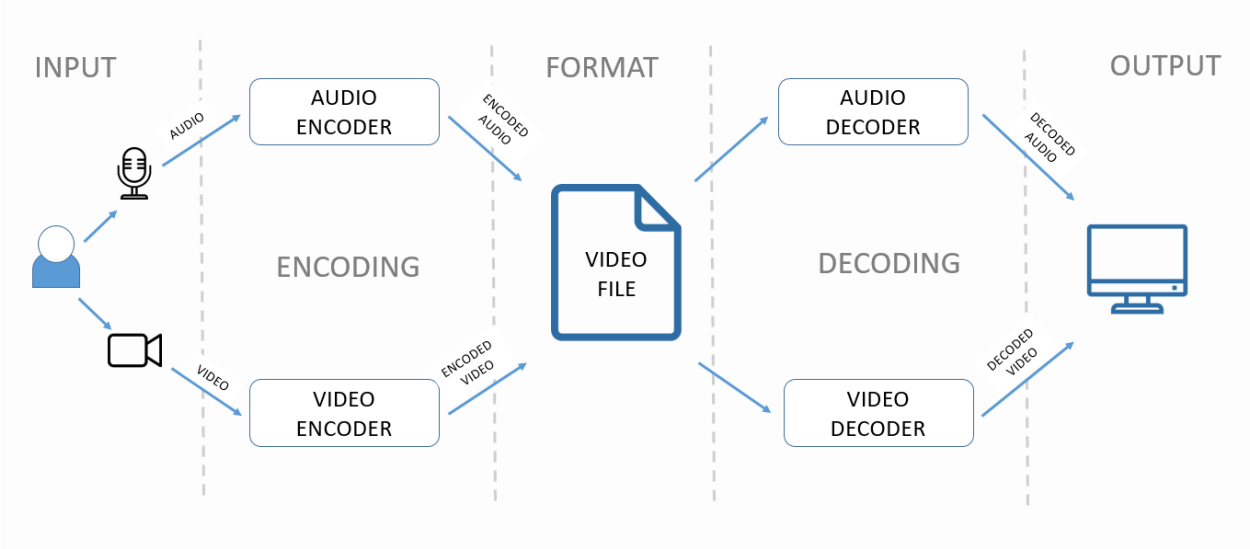

Рис.*1 Потокова передача даних*

1.4 Потокове обслуговування

Після успішного кодування файл завантажується на сервер доставки. Сервер доставки може бути або звичайним веб-сервером, або сервером потокової передачі.

Стандартний веб-сервер використовується для розміщення основної частини сайтів і просто завантажує файл. Такі веб-сервери дешевше, більш прості у використанні та використовують вже існуючу інфраструктуру. Потокова передача веб-сервера використовує дві технології: Hyper Text Transport Protocol (HTTP) та Transmission Control Protocol (TCP). Недоліком звичайних веб-серверів є те, що вони підтримують лише прогресивне завантаження — програвач мультимедіа починає програвання збереженої частини файлу, в той же час збираючи решту файлу з веб-сервера.

Сервер потокової передачі спеціально адаптован для доставки медіафайлів. Програвання файлу починається практично миттєво, доставка виконується у «фоновому режимі». Великою перевагою є інтерактивна навігація, яка дозволяє переміщуватися по файлу за допомогою управління плеєром. Сервер

потрібен знайти та видати користувачу правильні частини файлу, використовуючи індекс.

Досі існує проблема контроля швидкості доставки даних, тому що швидкість доставки залежить від швидкості, з якою цей потік був закодований. Одним зі способом вирішення цієї проблеми є кодування з кількома швидкостями, а потім вибір з оптимальною. За цю зміну файлів з різними швидкостями також відповідає сервер [4].

1.5 Мережі доставки контенту

Мережа доставки контенту (CDN) — це група географічно розподілених серверів для прискорення доставки контенту за рахунок скорочення шляху передачі мережею. Схема роботи зображена на рисунку 2.

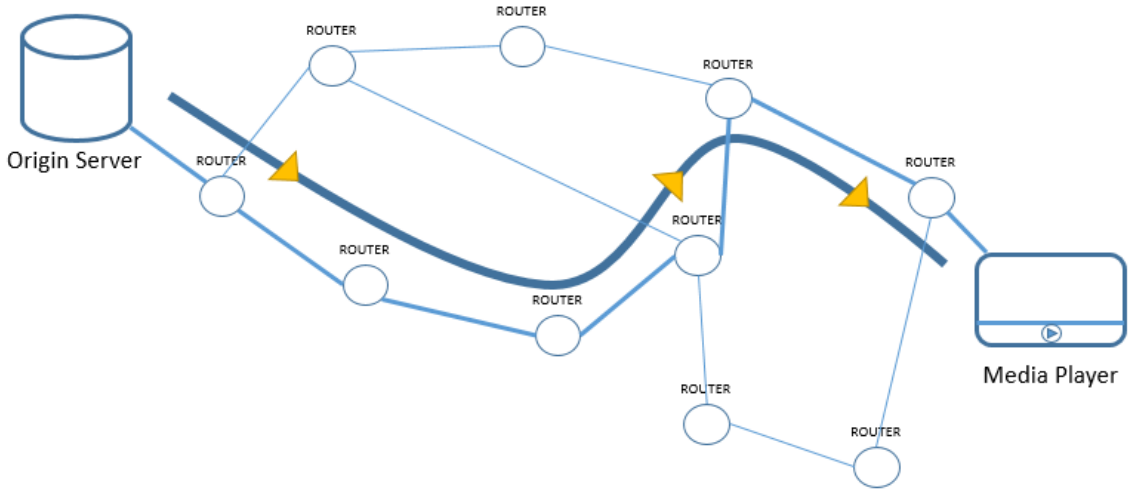

Рис.*2 Мережа доставки контенту*

Основна мета мереж доставки контенту — обслуговувати контент ближче до користувача.

Контент без використання CDN може проходити велику кількість маршрутизаторів. Також на кожному етапі пакети можуть бути втрачені або затримані, що має істотний вплив при потоковій передачі, яка вимагає регулярної швидкості доставки даних, щоб уникнути зависання зображення або втрати аудіо. CDN вирішує ці проблеми, обслуговуючи контент із наближеної точки до користувача (POP).

Мережа доставки контенту виконує чотири функції:

- інтелектуальна маршрутизація;
- прикордонне кешування вмісту;
- проксі-обслуговування;
- розділення прямих веб-трансляцій.

Інтелектуальна маршрутизація — це накладення понад звичайну Інтернет-маршрутизацію. Запити до сервера-джерела для мультимедійних кліпів перехоплюються та перенаправляються на найближчий доступний прикордонний сервер. Інтелектуальна маршрутизація також доставляє контент із сервера-джерела на прикордонний сервер за найкращим маршрутом, минаючи перевантажені посилання [4].

Центри обробки даних використовують кешування для тимчасового зберігання копії файлів, щоб швидше отримати доступ. CDN кешують вміст, такий як веб-сторінки, зображення та відео, на проксі-серверах поруч із вашим фізичним розташуванням.

Великою перевагою мережі доставки контенту є те, що пряма веб-трансляція може бути поділена на точки присутності користувача. При цьому потік для кожного клієнта замінюється одним потоком із сервера-джерела. Кінцева мета полягатиме у можливості багатоадресної розсилки в розрізнених мережах, що забезпечує найкращий потенціал для зменшення навантаження мережі для прямих веб-трансляцій.

1.6 Переваги відеострімінгових сервісів

Буферизація. Буферизація — це попередне завантаженя форматів при потоковій передачі даних. Завантаження форматів (частин файлів) відбувається на кілька секунд раніше їх програвання та не потребує повного завантаження великого файлу як при завантаженні на локальне сховище.

Буферизація передбачає використання буфера для тимчасового зберігання даних (Рис. 3). Це забезпечує безперервне відтворення відео при відсутності затримок у мережі. У разі поганого Інтернет-з'єднання, завантаженні мережі або проблем на сервері буферизація може зайняти більше часу, а медіафайл зупинить програвання [5].

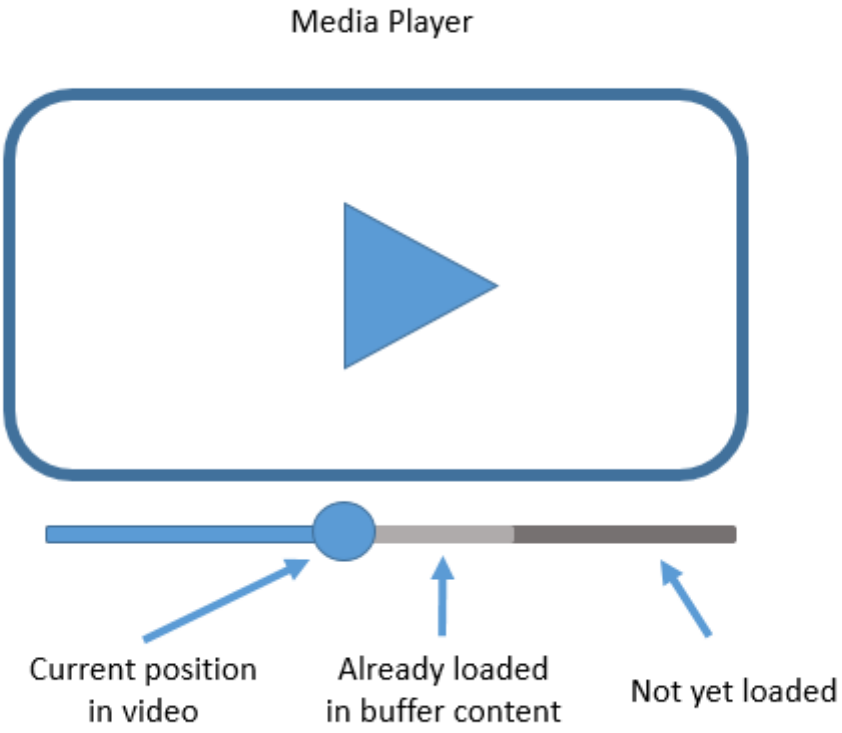

Рис.3 *Буферизація*

З вище зазначеного виходить, що буферизація забезпечує миттєве програвання файлів, уникаючи тривалого часу завантаження та витрати місця на жорсткому диску.

Комунікація. Потокова передача даних у реальному часі — гарний спосіб зв’язку з віддаленими користувачами будь-то друзі або клієнти.

Доступ. Потокові платформи спростили перегляд медіафайлів. Негайний доступ до кінофільмів, відео та аудіо. Все, що потрібно для цього — це лише стабільне Інтернет з’єднання. Потокові платформи надають можливість вибирати з величезного переліку варіанті та без дотримання розкладу як телебачення.

1.7 Опис проблеми

Кожен стикався з проблемами поганої картинки, нерозбірливого звука або зависання. У дзвінку трьох людей з цим ще можна миритися, а ось на значній відстані заході, тим більше платному, таке стане великим мінусом.

Глядачі хочуть спілкуватися один з одним і зі спікерами, глядачі хочуть відволіктися від рутини і отримати нові враження. Глядач сидить перед тим же екраном, що і кожен день — тобто обстановка буденна. І якщо спікери йому видно наочно, то ось інші глядачі куди менш помітні, ніж в офлайні: хоча є чати, спілкування не напрошується само собою. Замість яскравої події, де людина залучена в те, що відбувається, можна отримати ситуацію «він просто сидить і дивиться довгі монологи».

Зараз для будь-яких онлайн-заходів знайдуться відповідні сервіси — від YouTube і Twitch до Spatial Chat і Gather.Town. Є ще одна проблема, менш очевидна, але не менш суттєва — сервісів багато, але вони погано інтегруються один з одним. У них можуть бути абсолютно різні системи авторизації, інтерфейси, підходи.

1.8 Приклади використання відеострімінгу

Пряма трансляція використовується для багатьох цілей у багатьох галузях. Пряма трансляція призначена для того, щоб допомогти людям транслювати події, які вони не можуть відвідати особисто. Компанії та організації використовують пряму трансляцію для взаємодії зі підлеглеми та клієнтами.

Найпопулярніші випадків використання потокової трансляції включають:

- комунікації;
- трансляція подій;
- онлайн-освіта (лекції, тренінги тощо);
- трансляція спортивних подій;
- концертне потокове передавання;
- маркетинг.

1.9 Типи відеострімінгу

Існує дві моделі потокової передачі відео [6]:

1. відео за запитом;
2. пряма трансляція.

Приклади за моделями зображені на рисунку 4.

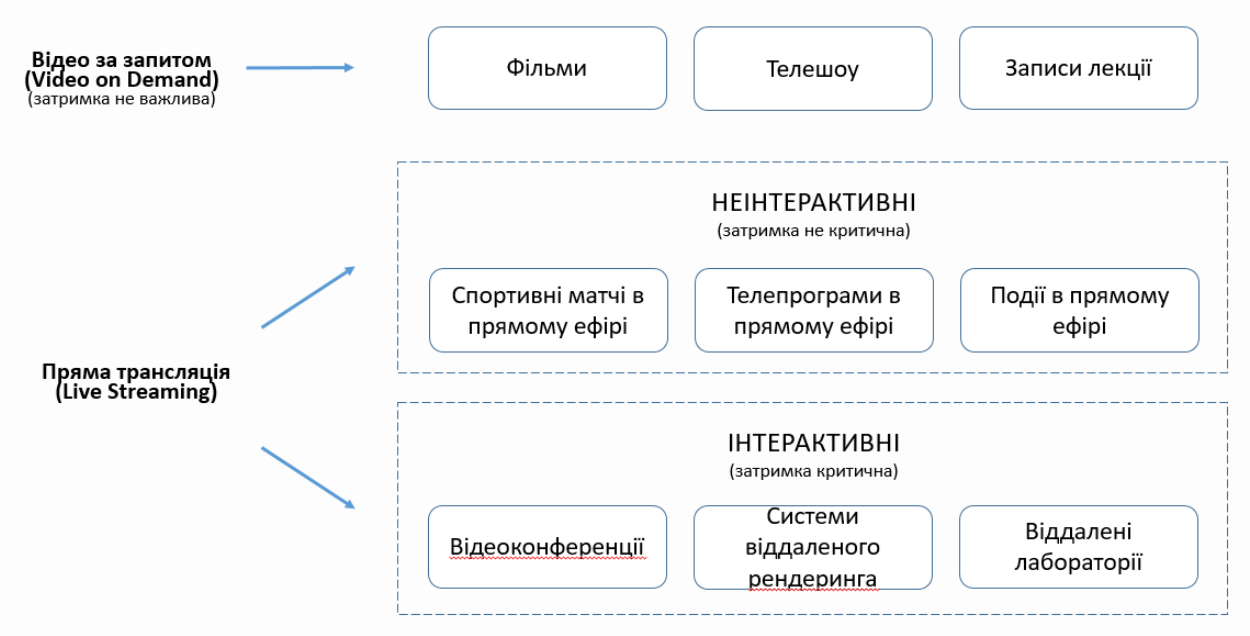

Рис.*4 Типи відеострімінгу*

Відео за запитом зберігається тривалий час на сервері та його перегляд не потребує завантаження повного файлу. Прикладом такої моделі виступають онлайн-кінотеатри, відеохостінги, записані заздалегідь програми по телебаченню.

Оскільки відео існують задовго до того, як їх побачать користувачі, вони можуть бути попередньо оброблені за бажанням. Вони можуть використовувати сильне стиснення і готувати відео з різною якістю і швидкістю передачі. Крім того, їх можна відносно легко передавати через адаптивну потокову передачу. Крім того, вони покладаються на буферизацію, щоб забезпечити кращу якість, незважаючи на проблеми з мережею, і щоб мати можливість використовувати більший кадр стиснення.

Базова архітектура відео за запитом складається з дисків, на яких зберігаються відеофайли, буфера, який виділяється для кожного запиту користувача, та сервера, який витягує відеодані з дисків в буфер. Схема зображена на рисунку 5.

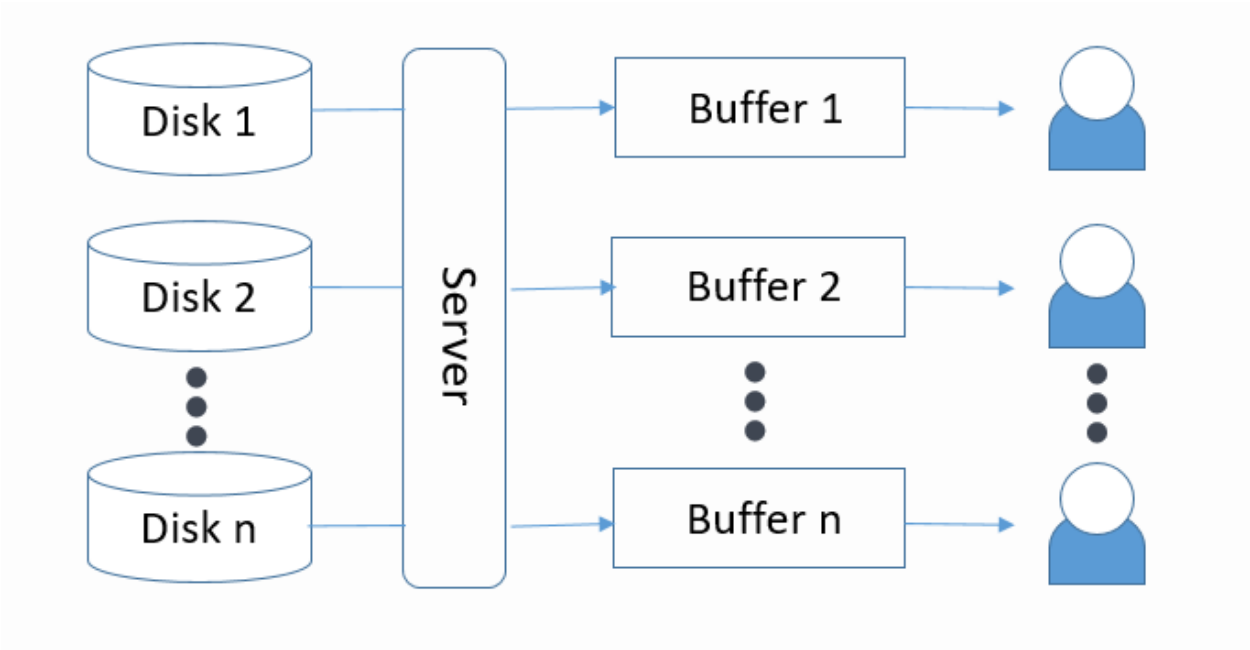

Рис. *5 Відео за запитом*

Пряма трансляція надсилається через Інтернет у режимі реального часу без попереднього запису та збереження.

Прямі трансляції поділяються на інтерактивні та неінтерактивні за своєю реалізацією.

В системі прямої трансляції контент зазвичай створюється під час трансляції. Однак все ще існує дуже значна затримка між моментом захоплення кадру і моментом його відображення на цільовому пристрої. Ця затримка є не тільки результатом затримки мережі або обладнання. Він вбудований в конструкцію для досягнення більш високої масштабованості. У неінтерактивних прямих трансляціях затримка в кілька секунд зазвичай не впливає на QoE (якість сприйняття) і дозволяє використовувати такі методи, як буферизація, сегментація відео і кодеки руху з високим ступенем стиснення.

При інтерактивній прямій трансляції користувачі не просто пасивні глядачі контенту. Натомість вони здатні взаємодіяти з ним або через нього, впливаючи на потік. Це накладає сильне обмеження на максимально прийнятну затримку відображення-зйомки, якої немає в неінтерактивних прямих трансляціях.

Архітектура прямої трансляції виглядає наступник чином (Див. рис. 6):

1. відеокамера або веб-камера захоплює відео;
2. відео надсилається в кодер через карту камери або інше з'єднання;
3. кодер перетворює файли у формати, які можна передавати;
4. кодер передає потокове відео на онлайн-платформу відео (або безпосередньо на CDN) через RTMP;
5. відео доставляється з відео CDN у відеоплеєр HTML5, спрямований на глядача.

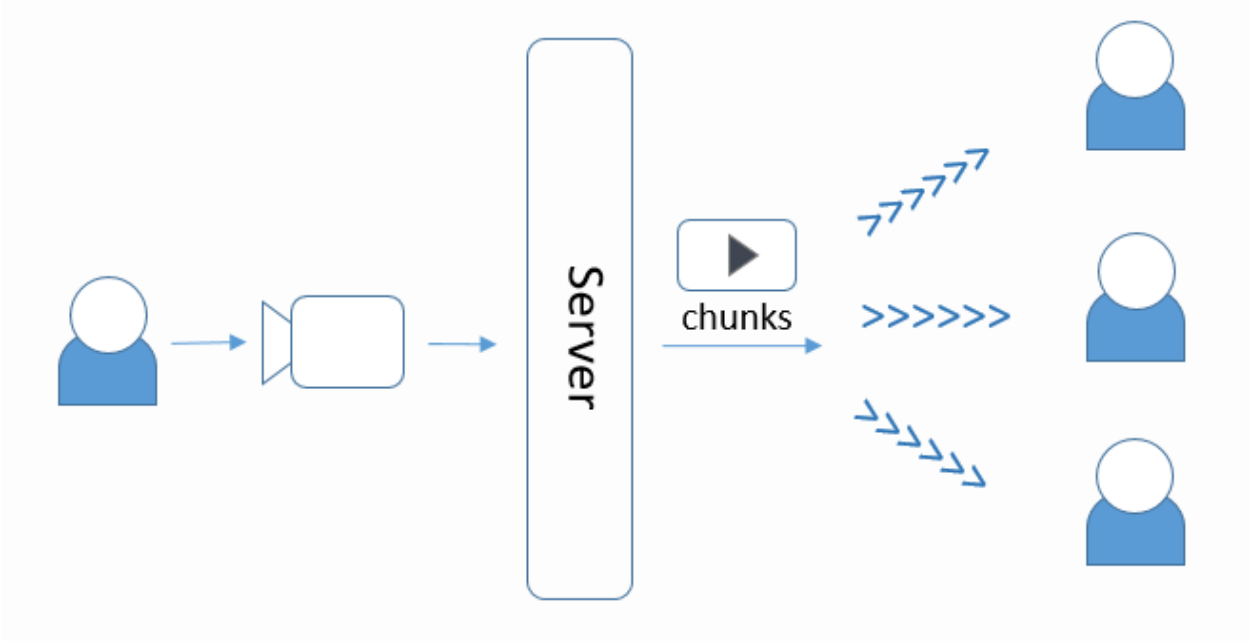

Рис. 6 *Пряма трансляція*

1.10 Постановка задачі

Метою роботи є проаналізувати існуючі відеострімінгові платформи, дослідити та порівняти інструменти для створення відеострімінгової платформи та розробити власний веб-сайт с інтеграцією Twillio API.

РОЗДІЛ 2 ДОСЛІЖДЕННЯ МОЖЛИВОСТЕЙ ІСНУЮЧИХ СЕРВІСІВ ДЛЯ ВІДЕОСТРІМІНГА

2.1 Відмінності неінтерактивних та інтерактивних прямих трансляцій

Порівняння будуть проводитися на прикладі прямого ефіру та відеоконференції.

У прямому ефірі організатор має повний контроль над тим, що бачать глядачі на відміну від відеоконференцій, в яких кожен користувач є учасником конференції. Організатор може заблокувати учасника або виключити звук та відео, але не може контролювати, що саме будуть говорити та показувати користувачі.

Відеоконференції забезпечують двосторонній зв’язок між користувачами та організатором за допомогою аудіо, відео та чату. В прямому ефірі організатор не бачить та не чує глядачів, спілкування зазвичай відбувається через чат.

У прямому ефірі немає обмежень на кількість користувачів, так як один потік організатора транслюється всім учасникам. У відеоконференціях існують обмеження на кількість людей в основному через апаратну частину, тому що необхідно передавати звук і відео учасників та організатора з меншою затримкою, що залежить від апаратного забезпечення кожного з учасників конференції, пропускної спроможності і програмної обробки відео.

Відео у прямих ефірах адаптується до пропускної здатності користувача, тому підтримують більш високу якість відео, але з більшою затримкою. В відеоконференції вона відсутня, тому користувачі з низькою пропускною здатністю можуть мати проблеми зі звуком і пікселізацією. Затримка в хвилину вважається нормою для прямого ефіру, тому що при прямому ефірі важливо передати картинку високої якості. Для відеоконференції затримка потрібна бути мінімальною для збереження імітації спілкування в реальному часі [3].

Результати порівняння представлені у таблиці 1.

Таблиця *1*

Порівняння неінтерактивних та інтерактивних прямих трансляцій

	Прямий ефір	**Відеоконференція**
Контроль власника над трансляцією	є	немає
Спілкування	чат	аудіо, відео, чат
Ліміт на кількість глядачів	немає	є
Якість	більш висока якість відео та аудіо	гірша якість відео та аудіо
Затримка	більша	менша

2.2 Існуючі сервіси для неінтерактивних прямих трансляцій

Twitch. Twitch — це найбільший стрімінговий майданчик за даними Streamlabs для потокової передачі в основному розважального та спортивного контенту, інтерфейс якого зображено на рисунку 7 [7].

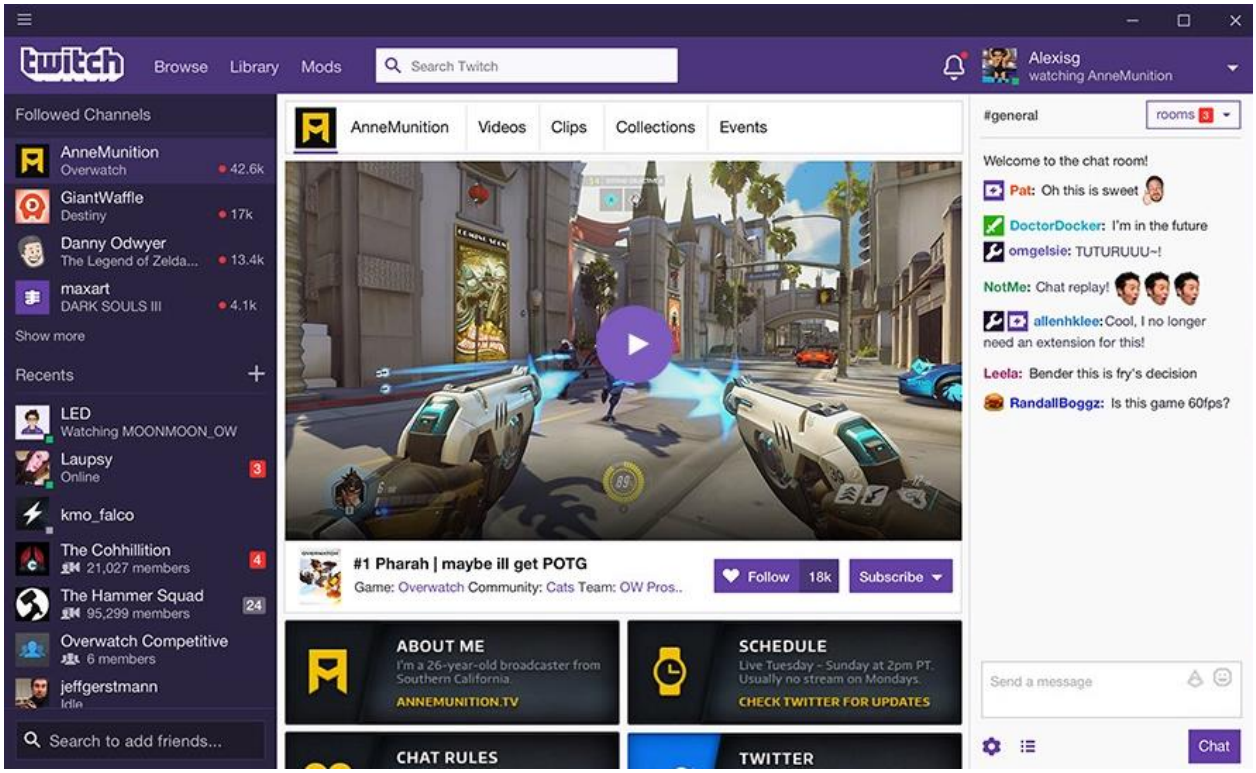

Рис.*7 Інтерфейс Twitch*

Twitch доступний через браузер на будь-якому пристрої, в тому числі для iOS і Android. Платформа проводить трансляції кіберспортівних турнірів, геймплея, а також співпрацює з найбільшими конференціями та світовими чемпіонатами в ігровій індустрії.

Основні характеристики [7]:

- інструменти для донату;
- в основне використання – трансляція відеоігор;
- онлайн-чат;
- інтегрування плеєру на власний сайт;
- наявність партнерської програми;
- присутня реклама;
- чат.

Переваги:

- мобільне мовлення;
- сумісний з пристроями iOS та Android;

- детальна аналітика;
- інтеграція з Amazon;
- висока якість відео.

Недоліки:

- дуже обтяжуючий інтерфейс;
- проблеми верстки через локалізацію зображені на рисунку 8;
- немає вдосконалених інструментів потокового відео;
- тільки партнери можуть транслювати з роздільною здатністю 1080p;
- половина заробітку від підписок відходять платформі;
- контент належить Twitch и перші 24 години його не можна публікувати в інших місцях;
- немає можливості створити приватну конференцію.

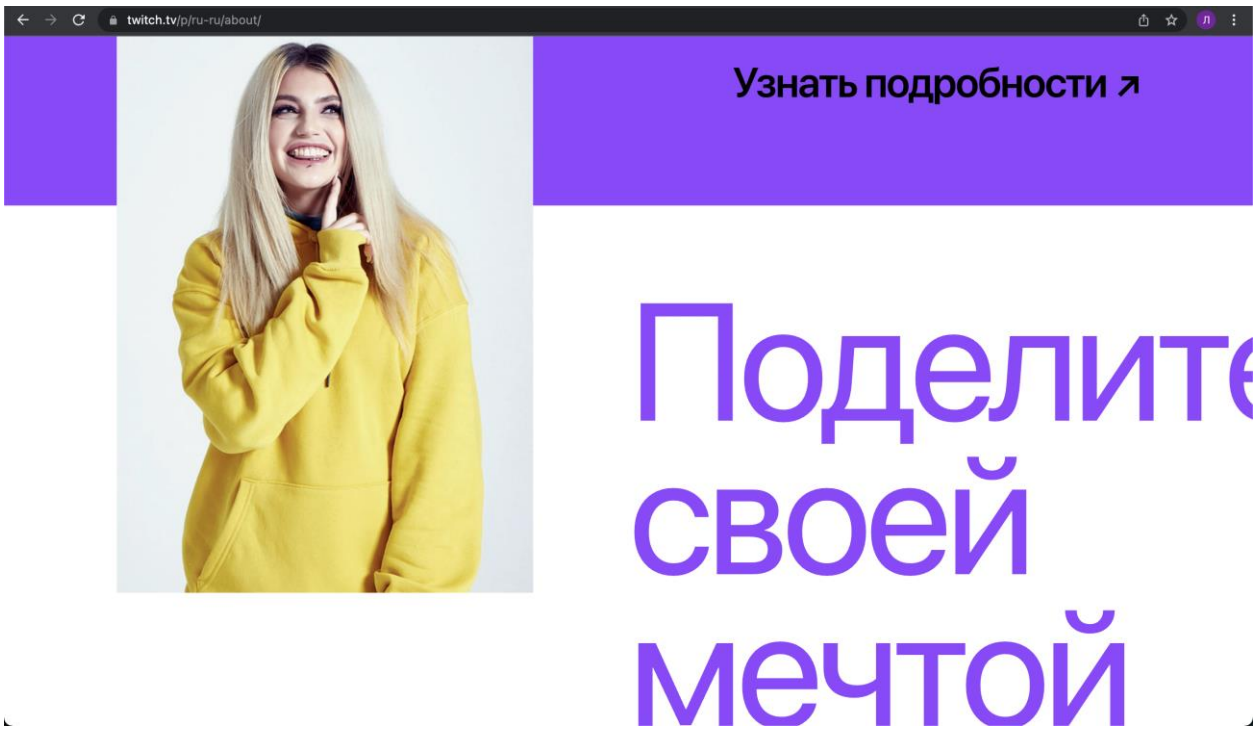

Рис. *8 Проблеми інтерфейсу*

Тарифні плани:

Twitch не вимагає коштів від авторів контенту за використання їх платформи, але він дозволяє їм заробляти гроші на своїх глядачах. Ця програма називається Twitch Subscriptions, і вона включає 3 рівні [7]:

- підписка на 1 рівень — 4,99 доларів/ місяць;
- підписка на 2 рівень — 9,99 доларів/ місяць;
- підписка на 3 рівень — 24,99 доларів/ місяць.

YouTube Live. YouTube була одна з перших платформ, яка зробила онлайн-трансляцію відео популярною [8].

Основні характеристики [8]:

- монетизація на основі реклами обмежена для користувачів з великим підписом та часом перегляду;
- користується попитом у споживачів;
- легко ділитися відео;
- інтеграція відео на інші веб-сайти;
- немає white-labeling;
- можна налаштувати рівні доступу для трансляції;
- чат;
- висока якість відео;
- зрозумілий інтерфейс.

Переваги:

- безкоштовне використання;
- глядачі знайомі з сайтом;
- зручний для користувача;
- легка інтеграція плеєра на веб-сайт;
- хостинг для прямого ефіру та VOD.

Недоліки:

- немає white-labeling;
- лише монетизація на основі реклами (з обмеженнями);

- обмеження права власності на вміст мовниками;
- рекламні матеріали та логотипи сторонніх виробників, що не стосуються бренду;
- повідомлення у чат може залишити лише залогований користувач.

Вартість:

Youtube Live безкоштовний.

Facebook Live. Facebook Live пропонує транслювання відео у прямому ефірі. Інтерфейс зображено на рисунку 9.

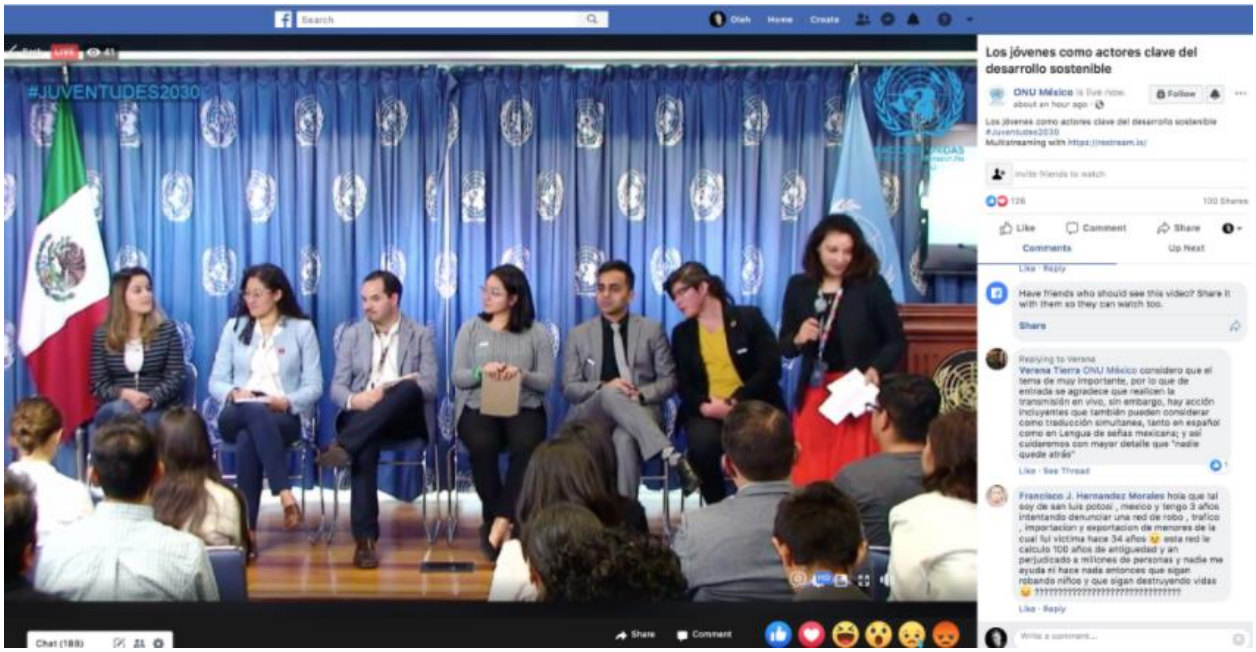

Рис.9 *Інтерфейс Facebook Live*

Прямі трансляції можуть розміщуватись в особистих профілях, на сторінках та в групах.

Facebook зазвичай використовується підприємствами та організаціями. Оскільки це безкоштовна платформа, Facebook ставить багато проблем, пов’язаних із володінням відео, обмеженнями часу на відео, монетизацією тощо. Окрім цих кількох технічних обмежень, пряма трансляція з Facebook Live є цінною для спілкування з існуючою аудиторією.

Основні характеристики [9]:

- простота обміну відео;
- живе коментування та реагування;
- деякі інструменти для ділового використання;
- відсутня аналітика;
- віщання з мобільного потребує додаток Facebook.

Переваги:

- додаток безкоштовний;
- глядачі знайомі з платформою;
- онлайн-чат;

Недоліки:

- немає варіантів монетизації;
- обмеження за часом на прямі відео трансляції;
- немає обмеження щодо доступу;
- відсутня аналітика
- відео контент погано індексується пошуковими системами.

Вартість:

Facebook Live безкоштовний.

Підсумки. Результати огляду неінтерактивних прямих трансляцій наведені у таблиці 2.

Таблиця *2*

Неінтерактивні трансляції

	Twitch	**Youtube Live**	**Facebook Live**
Вартість	безкоштовний	безкоштовний	безкоштовний
Монетизація	є	є	немає
Інтерфейс	складний наявні проблеми з локалізацією	зрозумілий	зрозумілий
Висока якість відео	1080p тільки для партнерів	+	
Мобільний додаток	+	+	+

2.3 Існуючі сервіси для інтерактивних прямих трансляцій

Zoom. Zoom — програма для організації відеоконференцій, розроблена компанією Zoom Video Communications. Програма має зрозумілий інтерфейс та виділяється швидкою роботою. Інтерфейс зображено на рисунку 10

Доступна у вигляді клієнта веб-браузера для комп'ютерів, мобільних додатків під iOS та Android, а також розширення для Chrome та Firefox.

Zoom також пропонує плагіни для комфортної роботи з іншими сервісами:

- Плагін для Microsoft Outlook — запуск та планування конференції через панель інструментів Microsoft Outlook.
- Плагін Zoom для IBM Notes — запуск та планування конференції в IBM Notes.
- Плагін клієнта Zoom для демонстрації екрану iPhone/iPad — автоматично встановлюється при демонстрації екрану iPhone/iPad віддаленим учасникам під час конференції.

- Плагін Zoom для Skype для бізнесу — дозволяє розпочати конференцію Zoom та запрошувати контакти з Skype.
- Контролери для Zoom Rooms — управління конференціями Zoom Room у конференц-залі за допомогою iPad або планшета на базі ОС Android або Windows.

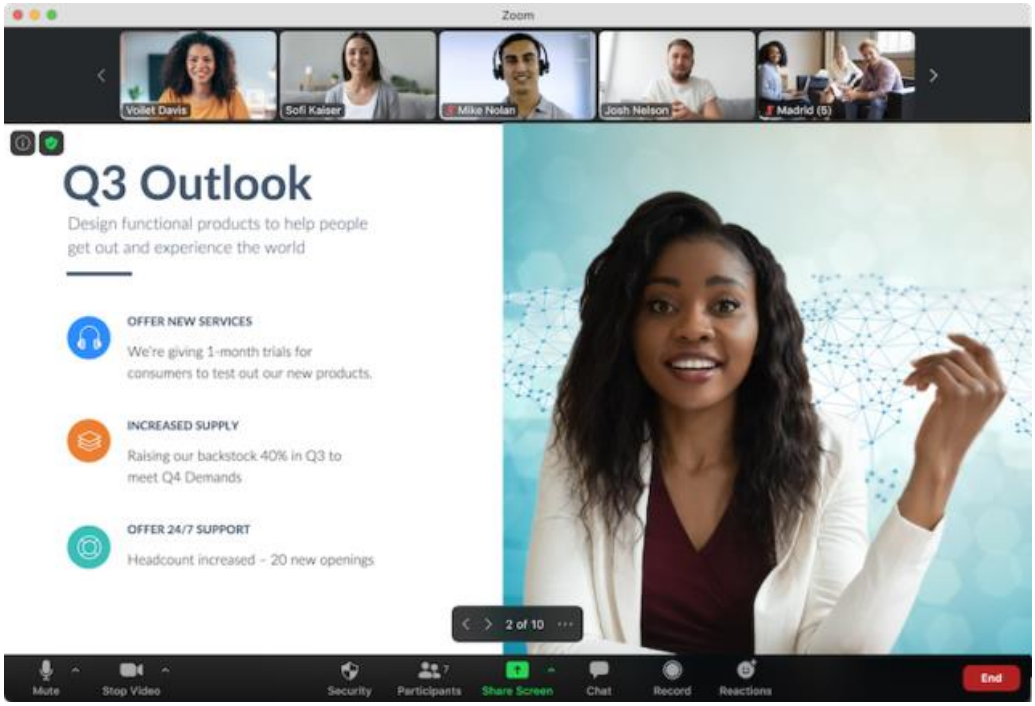

Рис.*10 Інтерфейс Zoom*

Основний функціонал [10]:

- організацію відео- або аудіозв'язку;
- демонстрація екрану;
- спільний доступ до файлів;
- колективний чат;
- запис;
- реакції;
- обмін текстовими та графічними повідомленнями;
- зберігання даних;
- розсилку запрошень на онлайн-конференцію;

- злиття з усіма операційними системами;
- використання віртуальних фонів та ефектів (Див. рис.11);
- відображення з ефектом присутності (Див. рис.12);
- функція «підправити мій зовнішній вигляд»;
- спільне використання екрану;
- безкоштовні функції IP-телефонії та тарифні номери.

Тарифні плани [10]:

- Базовий — безкоштовний. Має обмеження кількість учасників — до 100 учасників та на час групових конференції — 40 хвилин. Безлімітні конференції віч-на-віч.
- Професіональний — 14,99 доларів/місяць. Знімається ліміт у 40 хвилин та надається можливість запису відео в хмару.
- Бізнес — 19,99 доларів/місяць. Підвищується максимальна кількість учасників до 300, відкривається хмарне сховище для загальних матеріалів та можливості кастомізації.
- Підприємство — 19,99 доларів/місяць для великих підприємств. Підвищується максимальна кількість учасників до 500 та хмарне сховище стає необмеженим.

Рис.*11 Віртуальні фони та ефекти*

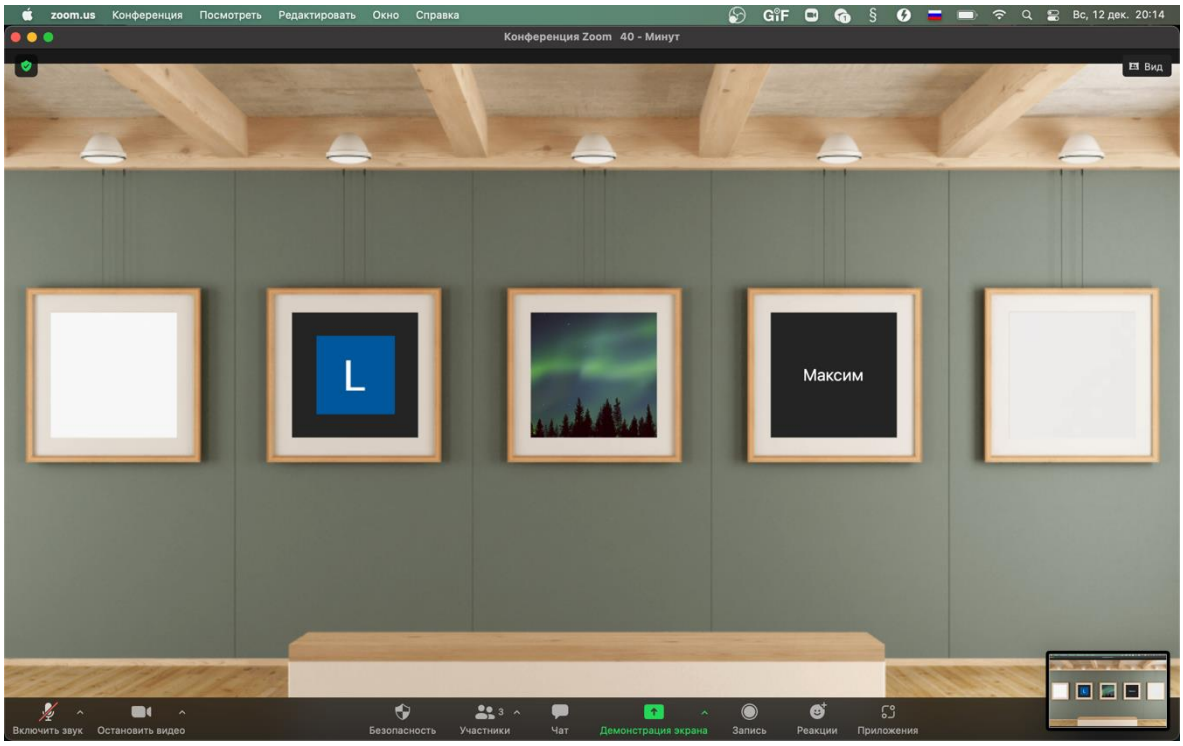

Рис.*12 Відображення з ефектом присутності*

Переваги:

- можливість користування безкоштовною версією;
- подарункові хвилини для Базового тарифного плану;
- зручний інтерфейс;
- підтримка багатьох операційних систем;
- багатий функціонал для мобільного застосунку.

Недоліки:

- незручно користуватися посиланнями;
- перепідключення кожні 40 хвилин у безкоштовній версії;
- необхідність встановлювати додаток.

Google Meet. Google Meet — це сервіс відео-телефонного зв'язку та відеоконференцій, розроблений компанією Google. Інтерфейс зображено на рисунку 13.

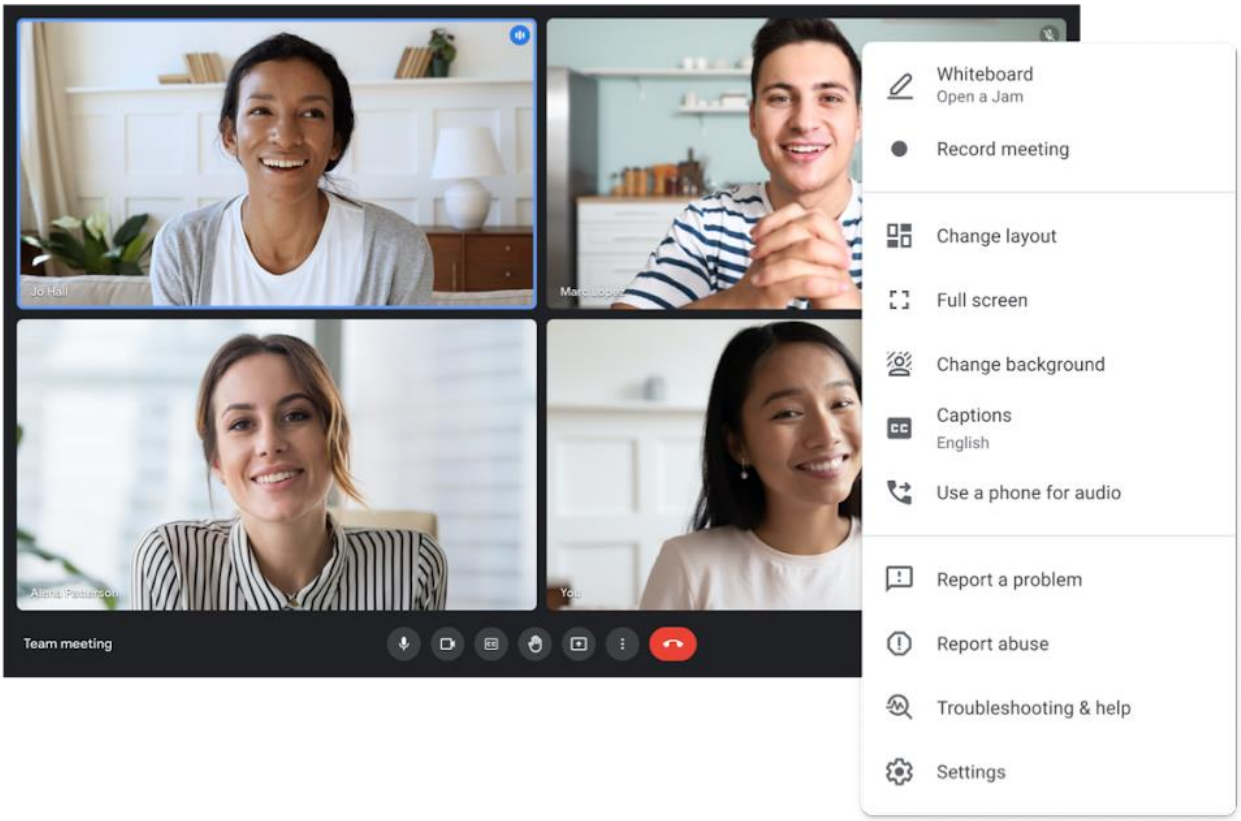

Рис.*13 Інтерфейс Google Meet*

Основний функціонал [11]:

- підтримка відеозустріч до 250 активних учасників;
- трансляція конференцій на 100 тисяч глядачів;
- підключення до конференцій субтитрів для іноземних учасників;
- планування зустрічей у календарях Google та Microsoft Outlook;
- показ презентацій;
- надсилання файлів учасникам;
- демонстрація робочого столу;
- біла дошка (Див. рис.14);
- опитування та голосування;
- текстовий чат;
- спільний доступ до екрану;
- запис зустрічей.

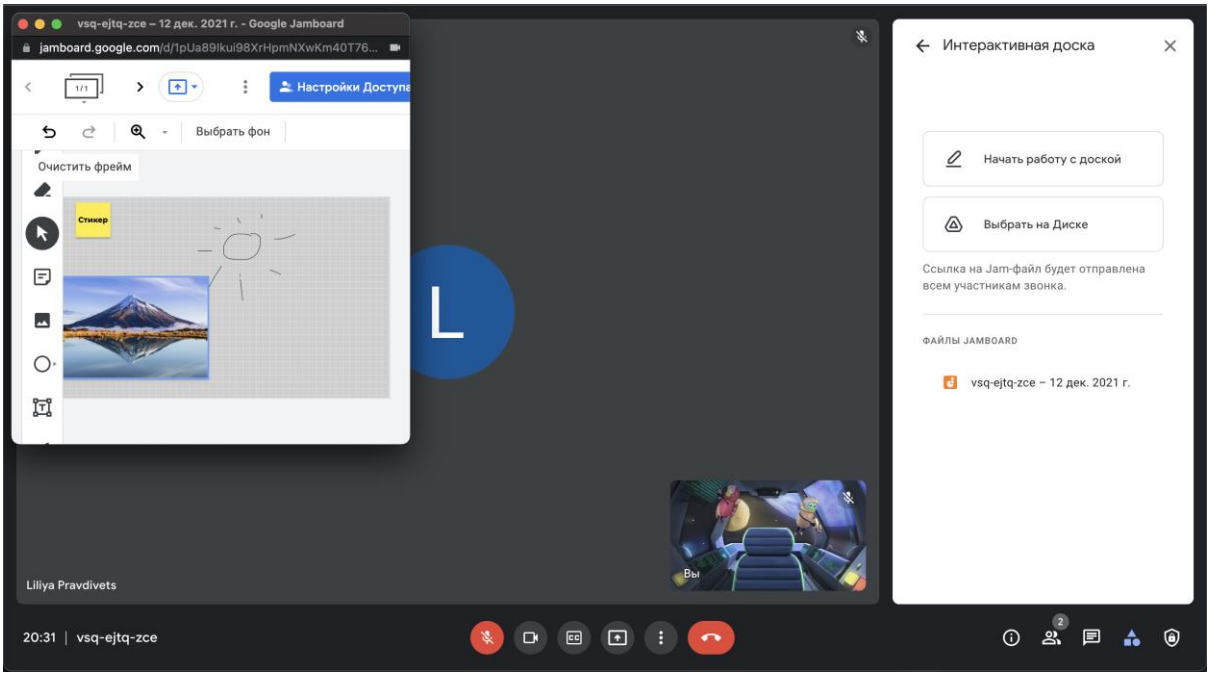

Рис.*14 Біла дошка*

Тарифні плани [11]:

- Google Meet — безкоштовний. Має обмеження кількість учасників — до 100 учасників та на час групових конференції — 1 година.
- Google Workspace Essentials — 8 доларів/місяць за кожного активного користувача. Підвищується кількість учасників до 150 та 24 години групових конференцій.
- Google Workspace Enterprise — за ціною потрібно звернутися до відділу продаж. Підвищується кількість учасників до 250. З'являється можливість трансляцій у домені на 100000 глядачів. Надається можливість дзвонити на конференції за допомогою телефонного номера.

Переваги:

- не потрібно завантажувати та встановлювати додаток;
- синхронізація контактів з Google акаунта;
- прив'язка Google Calendar;
- можливість дзвонити на конференції за допомогою телефонного номера у платному тарифі;

- інтеграція сервісів.

Недоліки:

- учасники конференції повинні мати обліковий запис Google;
- запис конференції можливий лише в платному тарифі;
- не виводиться витрачений час конференції.

Cisco Webex Meetings. Cisco Webex Meetings — це сервіс для проведення онлайн-нарад. Інтерфейс зображено на рисунку 15.

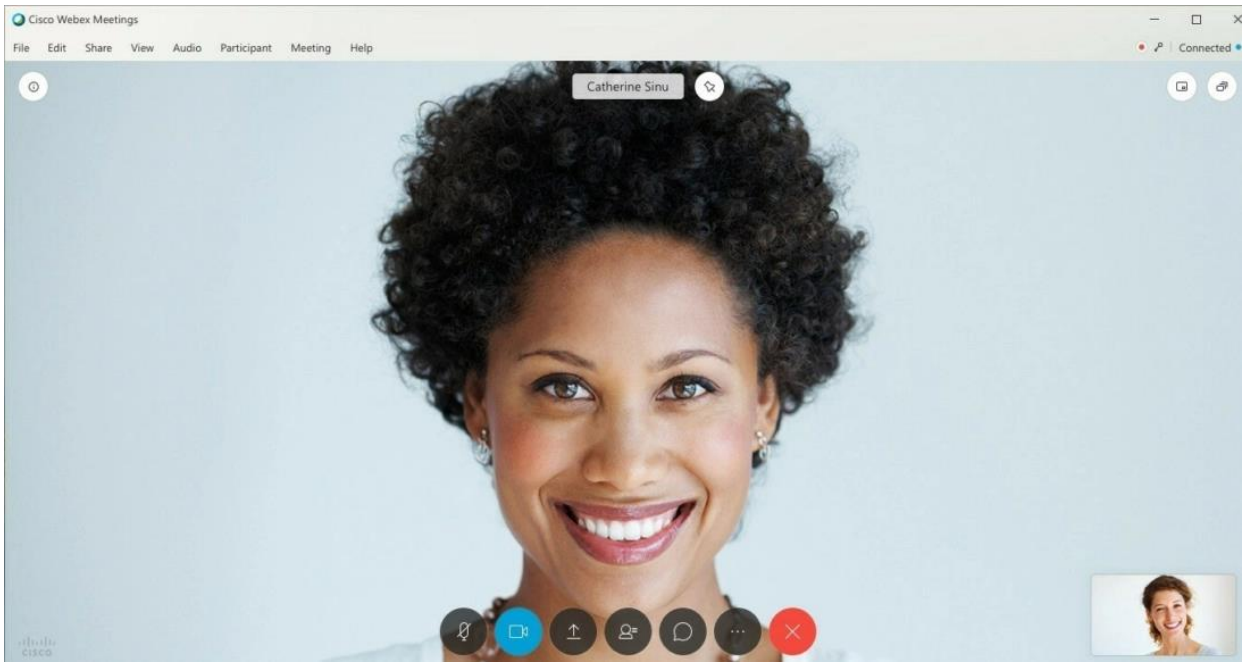

Рис.*15 Інтерфейс Cisco Webex Meetings*

Основний функціонал:

- приєднання за допомогою комп'ютера або мобільного пристрою;
- тестування аудіо та відео перед нарадою;
- вимкнення відео перед входом на нараду;
- перемикання між переглядом активного виступаючого та поданням у вигляді сітки (Див. рис.16);
- запис наради;
- збереження запису на комп'ютері;

- спільний доступ до робочого столу;
- спільний доступ до файлу;
- створення віртуальної дошки та надання до неї спільного доступу;
- початок нарад у Microsoft Teams, Slack або Workplace від Facebook.

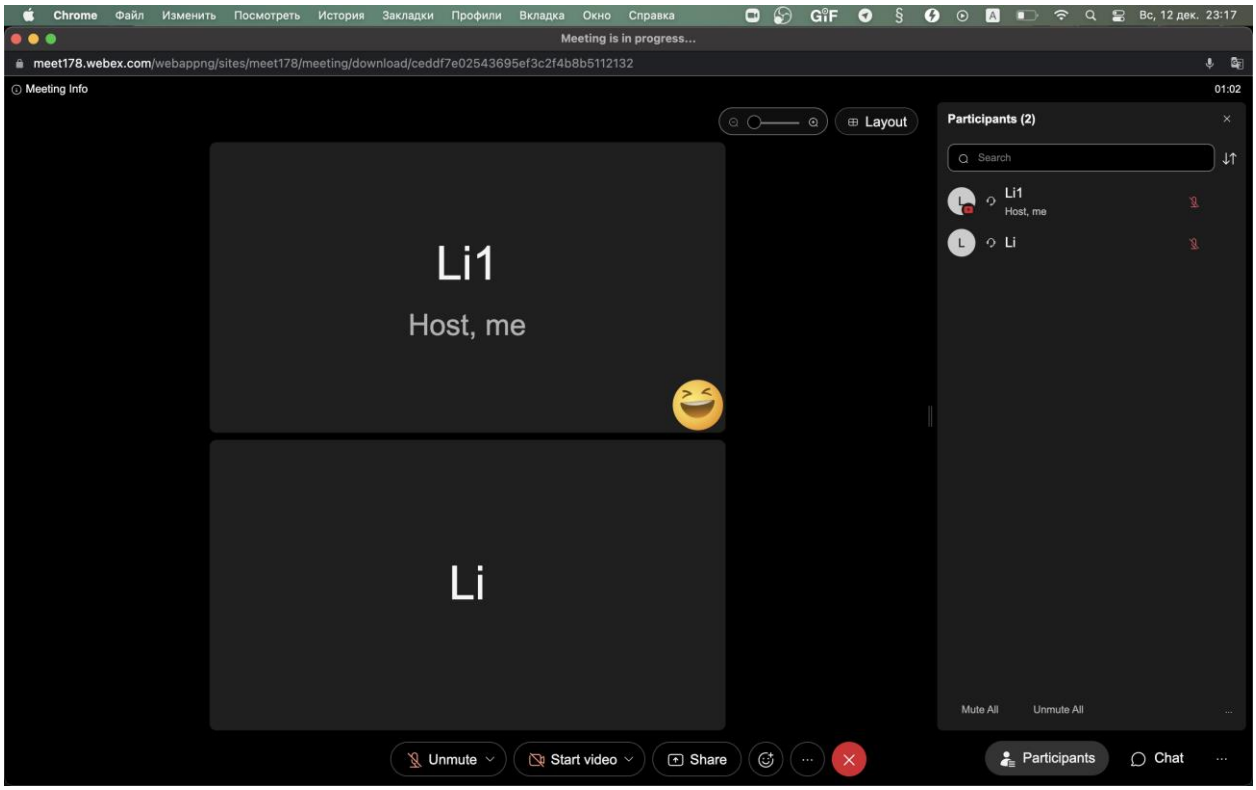

Рис.*16 Макет сітки*

Переваги:

- можна увімкнути наскрізне шифрування;
- наявна кімната очікування для персональних кімнат.

Недоліки:

- Безкоштовний запис конференції можливо зберегти лише локально на комп’ютері Webex.
- Відтворення запису на мобільних пристроях недоступне.
- Новий інтерфейс недоступний для користувачів Linux.
- Користувачі пристроїв Mac можуть одночасно приєднуватися лише до однієї event-наради.

- Проблеми верстки через локалізацію (Див. рис.17, 18).

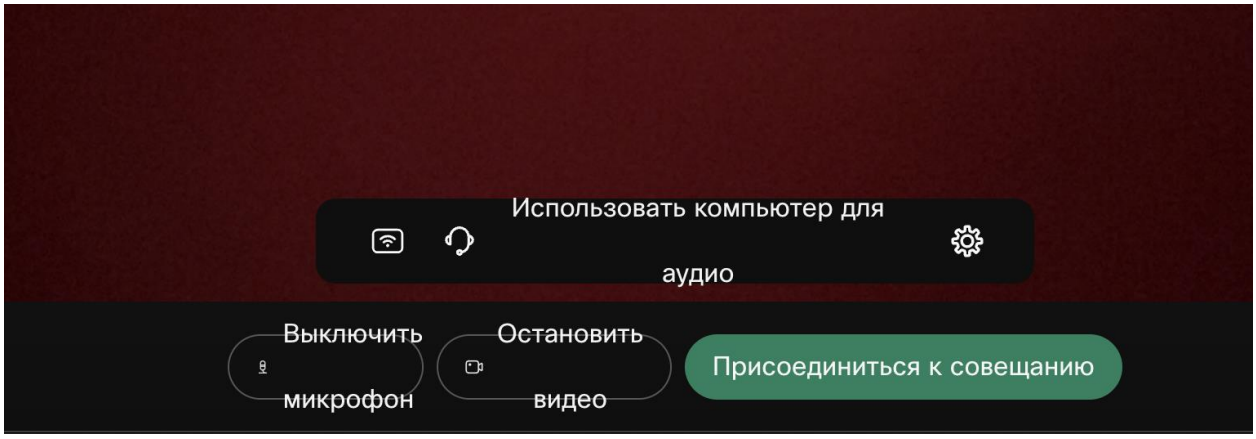

Рис.*17 Проблеми інтерфейсу*

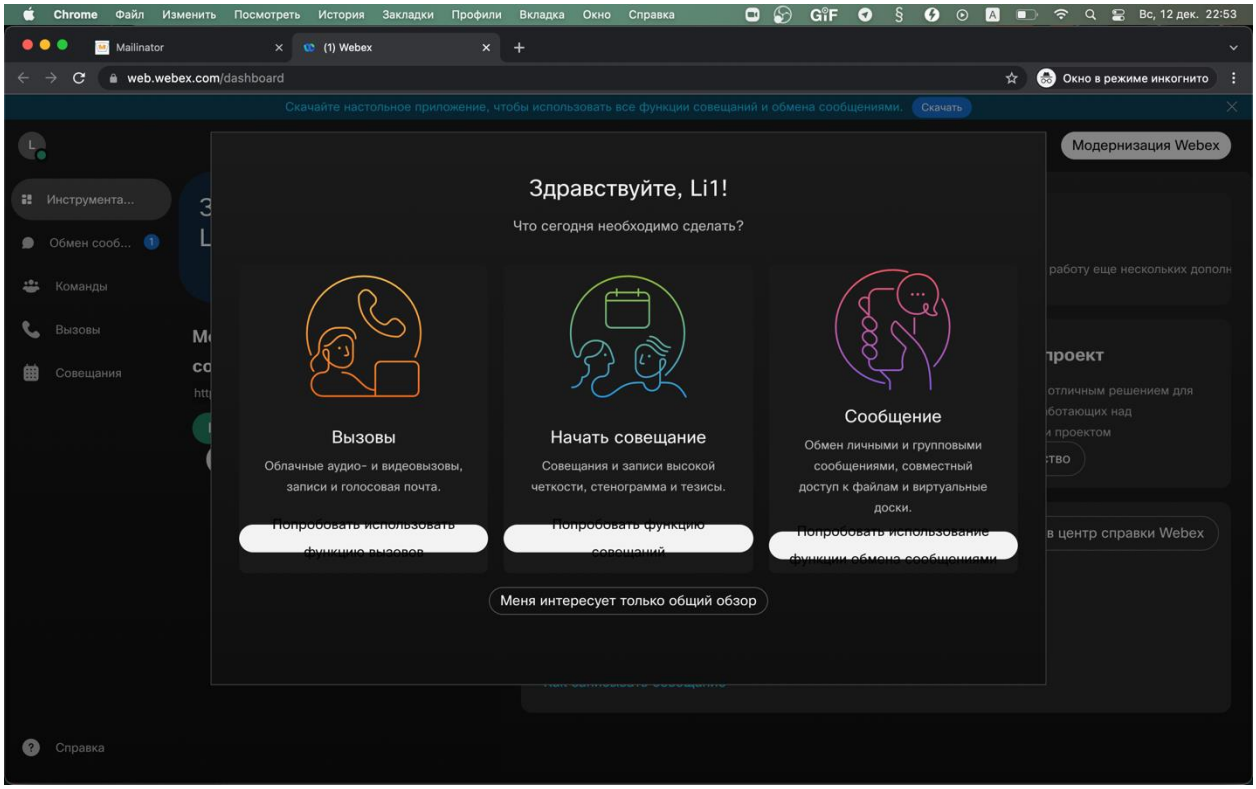

Рис.*18 Проблеми інтерфейсу*

Підсумки. Результати огляду інтерактивних прямих трансляцій наведені у таблиці 3.

- Таблиця *3*

Інтерактивні трансляції

	Zoom	**Google Meet**	**Cisco Webex Meetings**
Безкоштовне користування	+	+	+
Запис	локальне збереження	лише в платному тарифі	в безкоштовній версії лише локальне збереження на комп’ютер
Інтерфейс	зрозумілий	зрозумілий	менш зрозумілий, недоліки інтерфейсу псують враження
Висока якість відео	+	+	+
Користування	додаток (потребує встановлення)	наявна можливість проведення конференцій в браузері за наявністю облікового запису	браузер, додаток
Мобільний додаток	+	+	+

РОЗДІЛ 3 АНАЛІЗ ТЕХНОЛОГІЙ ПОТОКОВОЇ ПЕРЕДАЧІ ВІДЕО

3.1 Основні поняття

Кодувальник. Кодувальник стискає і перетворює вхідний аудіо-відео сигнал в цифровий, відповідний для передачі по мережі формат. Кодувальник необхідний тому, що більшість відеоджерел призначені для запису великих і громіздких відеофайлів не призначених для потокової передачі в реальному часі.

Кодувальники бувають програмні та апаратні.

Приклади програмних кодувальників:

- Telestream WireCast.
- Haivision KulaByte Encoders.
- Microsoft Expression Encoder Pro.

Декодувальник. Декодувальник перетворює формат в кадри, які доступні глядачу [13].

Частота кадрів. Частота кадрів – показує скільки кадрів захоплює камера за одну секунду [14]. Чим більше кадрів — тим плавнішим буде відео. Частота кадрів позначається як fps (frame per second) та не впливає на якість відео.

Основні стандарти [14]:

- 24 fps — стандарт для кіно
- 25 fps і 30 fps — стандарти для цифрового відео
- 60 fps — використовується для запису динамічного відео для програвання у сповільненому варіанті, наприклад, для трансляції спорту або комп'ютерних ігор

Вихідна роздільна здатність. Вихідна роздільна здатність екрану — це загальна кількість пікселів або точок, які можуть бути упаковані в фізичний розмір екрану.

Вихідна роздільна здатність відео — це загальна кількість пікселів в одному кадрі. [15]

Вихідна роздільна здатність екрану та відео можуть відрізнятися. Відео часто масштабується в залежності від фізичного розміру екрану. Тому, якщо різниця між роздільною здатністю екрана набагато більша за роздільну здатність відео, якість відео може погіршитися.

Стандарти роздільної здатності відео:

- 2160p: 3840 x 2160;
- 1440p: 2560 x 1440;
- 1080p: 1920 x 1080;
- 720p: 1280 x 720;
- 480p: 854 x 480;
- 360p: 640 x 360;
- 240p: 426 x 240.

Бітрейт. Бітрейт — скільки відео даних завантажується в секунду. Зазвичай зазначається в кілобітах в секунду (Кбіт/с), хоча тажок використовуються і в мегабіта в секунду (Мбіт/с).

Загальний діапазон значень: від 1000 до 8000 Кбіт/с. Найбільш поширені 1000 Кбіт/с (абсолютний мінімум для прямої трансляції) 2500 Кбіт/с, 3000 Кбіт/с, 5000 Кбіт/с. Це число залежить від частоти кадрів та роздільної здатності відео : чим вище частота кадрів і роздільна, тим вище повинна бути швидкість передачі для плавного і якісного стріму.

Розмір файлу дорівнює битрейту (килобит в секунду) помноженому на тривалість. Високий бітрейт відео сприяє високій якості відео та більшому розміру, низький – низькій якості з меншим розміром файлу, що актуально

тільки при порівнянні одного й того самого відео з однаковою роздільною здатнітю.

Кодек. Кодек — це спосіб стиснення (кодування) аудіо- і відеофайлів для більш швидкої передачі. Найбільш поширеним зараз є кодек H.264.

Кодеки поділяються на стиснення з втратами та без втрат.

Перекодування (Transcoding). Транскодування еквівалентно потокової передачі відео, за винятком того, що вихідні дані відправляються у файл. [] Перекодування робить можливим багатобітове та адаптивне потокове передавання. Бітрейт відноситься до якості відео. Перекодування відео створює кілька передач або версій одного відеофайлу з різними якостями.

Адаптивні бітрейтні відеоплеєри автоматично вибирають відповідну передачу на основі швидкості Інтернету глядачів. Це допомагає уникнути будь-якого відставання та буферизації у випадку, якщо у глядача погане з’єднання з Інтернетом.

3.2 Існуючі рішення для потокової передачі відео

WebRTC. WebRTC (Web Real Time Communications) — це стандарт, що описує потокову передачу аудіо, відео даних та іншого контенту між браузерами або іншими додатками в режимі реального часу без потреби встановлення плагінів чи інших розширень [16]. Технологія перетворює браузер в термінал відеоконференцзв'язку та для початку спілкування, досить лише відкрити веб-сторінку конференції в браузері.

Як працює WebRTC:

1. Користувач відкриває сторінку, що містить вміст WebRTC.
2. Браузер запитує доступ до веб-камери та мікрофона за необхідністю, наприклад, при участі у відеоконференції. У випадках перегляду прямих трансляцій вони не потрібні. Пристрої без доступу не використовуються.

3. У браузері, ініціюючим з'єднання, формується SDP-пакет, який містить всю необхідну інформацію про параметри з'єднань: які дані будуть передаватися, за допомогою яких кодеків та інше.
4. У залежності від реалізації технологій ініціатор з'єднань передає цей пакет іншим учасникам. Найчастіше для цього використовується сигнальний сервер і протокол WebSocket [16].
5. На приймаючій стороні браузер отримує пакет SDP, а потім генерує подібний, але з отриманою інформацією з першого. Другий пакет відправляється назад, до ініціюючої сторони. Тепер обидва клієнти вже мають мінімальне представлення одного про друзів.
6. У залежності від реалізації, паралельно з попередніми шагами відбувається аналіз стану підключень до мереж. Клієнтам передається адреса STUN-сервера, який використовується, щоб дізнатися зовнішній пристрій IP-адреси. Він порівнюється з внутрішнім IP-адресом для того, щоб визначити, що використовується NAT у даному підключенні, і, якщо так, для маршрутизації використовуються UDP-пакети. У більш сложних випадках (наприклад, при використанні подвійного NAT) застосовується TURN-сервер. Вони, по суті, є ретрансляторами, переважною сполукою клієнт-клієнт (P2P) у клієнт-сервер-клієнт [16].
7. Якщо всі шаги пройдени успішно, то з'єднання встановлюється. Першочергово визивається подія onicecandidate, яка передає інформацію про IP-адреси, налаштування NAT, спроби підключення між клієнтами.

Переваги стандарту:

- Не потрібне встановлення додаткового програмного забезпечення.
- Висока якість зв'язку завдяки використанню сучасних відео- та аудіокодеків; автоматичного налаштевання якості потоку за умов з'єднання; вбудованої системи ехо- і шумоподавань; автоматичне регулювання рівня чутливості мікрофонів учасників (АРУ).

- Високий рівень безпеки: усі з'єднання захищені та зашифровані відповідно до протоколу DTLS та SRTP. При цьому WebRTC працює лише за протоколом HTTPS, використовуючи технологічний сайт, повинен бути підписаний сертифікатом [10].
- Підтримка технологій SVC додана як частина реалізації кодеків VP9 та AV1. Крім того, що на поточний момент все ще не реалізовано в самих браузерах, програмні рішення TrueConf дозволяють використовувати SVC у браузерних клієнтах.
- Є вбудований механізм охоплення контенту, наприклад, робочого стола.
- Можливість реалізації любого інтерфейсу управління на основі HTML5 та JavaScript.
- Проект з відкритим вихідним кодом - можна вступити у свій продукт або послугу.
- Постійна крос-платформенність: одночасно і те, що додаток WebRTC буде однорідно добре працювати на будь-якій операційній системі, десктопній або мобільній, за умови, що браузер підтримує WebRTC. Це значно економлять ресурси на розробці ПО.

Недоліки стандарту:

- Всі WebRTC рішення несумісні між собою, тому що стандарт описує лише способи передачі відео і звуку, залишаючи реалізацію способів адресації абонентів, відстеження їх доступності, обміну повідомленнями та файлами, планування та іншого за розробником. Іншими словами, ви не зможете зателефонувати з одного WebRTC програми до іншої.
- Для користувачів, які турбуються про свою приватності, неприємним відкриттям стане те, що WebRTC визначає їх реальні IP-адреси. При цьому зберегти анонімність не допоможе ні проксі, ні

використання мережі Tor. Приховати IP-адресу можна за допомогою різних VPN сервісів, а також при використанні TURN-сервера. При необхідності використання WebRTC можна відключити.

- WebRTC не підтримує віддалене управління робочим столом. Так, транслювати те, що відбувається на екрані пристрою можна, але це буде такий же односторонній відеопотік, як і зображення, що передається з камери і способу взаємодії з джерелом потоку немає. Зроблено це з міркувань безпеки: код Javascript не може керувати будь-чим за межами поточного вікна браузера. Більше можливостей, включаючи віддалене управління робочим столом, можна отримати при використанні спеціально розроблених клієнтських додатків вендорів ВКС.

Створення RTCPeerConnection наведено у лістингу 1.

Лістинг *1 Створення RTCPeerConnection*

```
socket.on("watcher", id => {
  const peerConnection = new RTCPeerConnection(config);
  peerConnections[id] = peerConnection;
  let stream = video.srcObject;
  stream.getTracks().forEach(track =>
peerConnection.addTrack(track, stream));
  peerConnection.onicecandidate = event => {
    if (event.candidate) {
      socket.emit("candidate", id, event.candidate);
    }
  };
  peerConnection
    .createOffer()
    .then(sdp => peerConnection.setLocalDescription(sdp))
    .then(() => {
```

```
      socket.emit("offer", id,
peerConnection.localDescription);
    });
});
socket.on("answer", (id, description) => {
  peerConnections[id].setRemoteDescription(description);
});
socket.on("candidate", (id, candidate) => {
  peerConnections[id].addIceCandidate(new
RTCIceCandidate(candidate));
});
```

Twilio. Twilio — це хмарне програмне забезпечення з моделлю PaaS, яке забезпечує платформу як структуру послуг для своїх голосових та відеозв’язків [17].

Платформа дозволяє легко інтегрувати різні методи комунікації та використовувати існуючі навички веб-розробки, код, сервери, бази даних і карму для швидкого і надійного рішення проблем зі зв'язком.

Система створена для здійснення дешевих телефонних дзвінків, покупки номерів для різних країн, створення викликів між кількома телефонами з одного облікового запису і запуску конференцій. Звідси можна навіть створювати свої скрипти для кінцевої настройки. В цілому це безкоштовний сервіс, платити потрібно лише за дзвінки, конференції та повідомлення.

Основні характеристики сервісу Twilio [17]:

- API номерів телефонів.
- Перетворення тексту в мову і відтворення аудіо.
- Запис телефонної розмови і збереження.
- Створення конференцій.
- API для масового обслуговування.
- Черги викликів.

- Перетворення мови в текст.
- Статуси зворотних викликів і журнал.
- Глобальне охоплення.
- Легке підключення до акаунту Twilio.
- Вибір вхідного номера.
- Функціональність офісної АТС.
- Один вхідний номер для дзвінків на всі телефони.
- Пошук контактів в мережі для здійснення дзвінків.
- Користувацька кнопка для створення власних скриптів.
- Інтеграція з контакт-системою.
- Купівля тимчасових номерів, використовуючи обліковий запис Twilio.
- Повний журнал дзвінків.

Основні особливості Twilio API [17]:

- Twilio API (Application Programming Interfaces) — це комунікаційна платформа, яка може бути інтегрована з різними додатками.
- API Twilio з'єднують та оптимізують комунікаційні мережі, що дозволяє користувачам здійснювати дзвінки та обмінюватися по всьому світу.
- Проста інтеграція: підтримка різноманітних мов програмування.
- Регіональна сумісність: Розробники можуть керувати та налаштовувати систему залежно від потреб конкретного регіону, пропонуючи регіональну сумісність.
- Партнерське рішення Twilio: Twilio пропонує афілійовані рішення, що пропонують маркетингову допомогу, різноманітні програми, сертифікації, навчання тощо.
- Програмовані SMS: користувачі можуть надсилати та отримувати текстові повідомлення з будь-якої точки світу.

- Програмований голос: здійснювання або приймання дзвінків, а контроль їх якості та масштабованість.
- API електронної пошти Twilio SendGrid: створення власних електронних листів, які також можна автоматизувати.
- Програмований чат: інтеграція чату за допомогою Twilio API на мобільний або веб-додаток.
- Програмоване відео: Twilio API дозволяє розробникам створювати відеозв'язок у реальному часі або аудіо-додатки HD.
- Програмований факс: Twilio API також може поєднуватися з деякими системами для надсилання та отримання факсових повідомлень.
- API Twilio для WhatsApp: WhatsApp — це найбільш часто використовувана платформа; Twilio API також пропонує інтеграцію з WhatsApp.

Варіанти цін на API Twilio [17]:

- «Початковий» — безплатно;
- «PRO» — 99,00 доларів на місяць.
- «Бизнес» — 999,00 доларів на місяць.

Підтримка браузерів зображена на рисунку 19.

	Chrome	Firefox	Safari	Edge (Chromium)
Android	✓	✓	-	-
iOS	*	*	✓	-
Linux	✓	✓	-	-
macOS	✓	✓	✓	✓
Windows	✓	✓	-	✓

Рис. *19 Підтримувані браузери*

* Chrome і Firefox для iOS не мають доступу до API WebRTC, на відміну від Safari для iOS.

Підключення до кімнати:

Використання `connect`, для підключення до кімнати веб-програми. Після підключення можна надсилати та отримувати аудіо- та відеопотоки з іншими учасниками, які підключені до кімнати. Приклад підключення наведений у лістингу 2.

Лістинг *2 Підключення до кімнати*

```
const { connect } = require('twilio-video');
connect('$TOKEN', { name:'my-new-room' }).then(room => {
  console.log(`Successfully joined a Room: ${room}`);
  room.on('participantConnected', participant => {
```

```
    console.log(`A remote Participant connected:
${participant}`);
  });
}, error => {
  console.error(`Unable to connect to Room:
${error.message}`);
});
```

Налаштування медіа:

Захоплення місцевих медіафайлів з мікрофона, камери або спільного використання екрана на різних платформах наведено у лістингу 3.

Лістинг 3 Налаштування медіа

```
const { connect, createLocalTracks } = require('twilio-
video');

// Option 1
createLocalTracks({
  audio: true,
  video: { width: 640 }
}).then(localTracks => {
  return connect('$TOKEN', {
    name: 'my-room-name',
    tracks: localTracks
  });
}).then(room => {
  console.log(`Connected to Room: ${room.name}`);
});

// Option 2
connect('$TOKEN', {
  audio: true,
```

```
  name: 'my-room-name',
  video: { width: 640 }
}).then(room => {
  console.log(`Connected to Room: ${room.name}`);
});
```

Переваги рішення:

- великий стек рішень;
- можливість інтегрування кількох в одному додатку;
- невелика ціна;
- перевага якості між іншими готовими веб рішеннями;
- простий в користуванні;
- багато прикладів під різні мови та фреймворки.

Недоліки:

- документація з помилками;
- не підтримується для Chrome і Firefox для iOS.

CONTUS MirrorFly. CONTUS MirrorFly — провідне комунікаційне рішення для організацій та підприємств для створення або інтеграції функцій обміну повідомленнями, голосовими та відеодзвінками у свої веб-програми та мобільні додатки. Рішення пропонує настроювані API та SDK з оновленими функціями та параметрами хостингу, такими як локальний або хмарний [18].

Можливості CONTUS MirrorFly [18]:

- Розмови в режимі реального часу з необмеженою кількістю користувачів, створивши публічні або приватні групи під централізованим каналом.
- CONTUS MirrorFly можна інтегрувати в будь-які існуючі програми, такі як iOS, Android та веб-програми, без необхідності знання кодування.

- Проактивний чат — функція оснащена індикаторами друку, її присутність у мережі та додавання тегів робиться для підвищення її чуйності під час спілкування в чаті
- CONTUS MirroFly надає багаторазову автентифікацію, яка допомагає отримати доступ до всього SDK для відеодзвінків з обліковими даними для використання всієї послуги. Все це з оперативним сповіщенням користувачів до кінця.
- Інтерактивна голосова відповідь (IVR) дозволяє користувачеві взаємодіяти з системою, де інформація вже буде попередньо записана людським голосом. І коли клієнт розміщує свої запитання, відповідь буде дана відповідно. Таким чином, це може зменшити час очікування клієнта на підключення до агента.
- Ви можете здійснювати необмежені дзвінки різним користувачам по всьому світу.
- Голосовий моніторинг та звітування дозволяє управляти, аналізувати, відстежувати всі деталі дзвінків, що включає коефіцієнт зниження, час обробки дзвінків і коефіцієнт конверсії дзвінків для кращого прийняття рішень.
- Прямий ефір дозволяє мати можливість прямої трансляції з вмістом на вимогу, таким як вебінари, зустрічі проектів для користувачів по всьому каналу. Ця програма доступна для груп.

Вартість [18]:

- Essentials — 999 доларів на місяць;
- Growth —1,999 доларів на місяць;
- Premium —2,999 доларів на місяць.

Переваги:

- наявність звітності та аналітики;
- вбудовані інструменти монетизації;
- комплексні варіанти зв’язку;

- сумісна праця на дошці;
- обмін файлами;
- крос-платформна підтримка;
- підтримка низької пропускної здатності;
- проста інтеграція;
- використовування наскрізного шифрування.

Недоліки:

- висока ціна

РОЗДІЛ 4 РОЗРОБКА ВІДЕОСТРІМІНГОВОГО СЕРВІСУ

4.1 Засоби реалізації

Фреймворк Vue.js. Vue.js — це фреймворк мови JavaScript з відкритим вихідним кодом для створення користувацьких інтерфейсів в парадигмі реактивного програмування. Він легко інтегрується в проекти з використанням інших JavaScript-бібліотек [19].

Веб-сторінка організовується у вигляді дерева вкладених компонентів як зображено на рисунку 20. Компоненти дозволяють створити елементи користувача, які можна повторно перевикористовувати. Це допомагає оптимізувати код за допомогою уникнення дублювання.

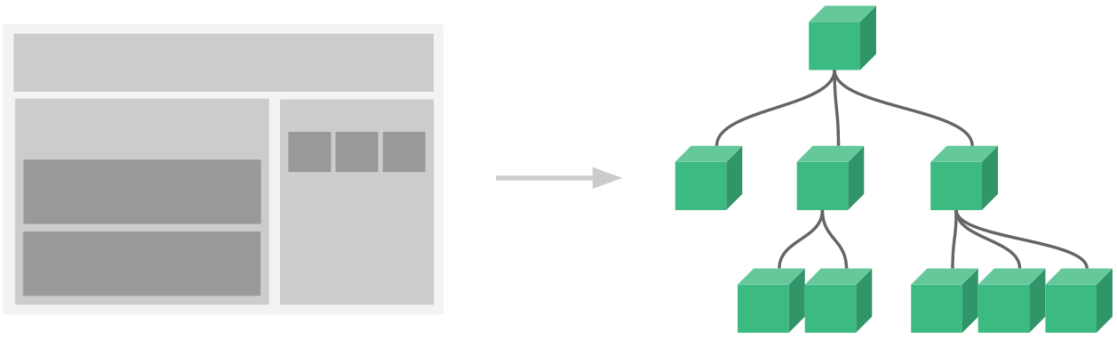

Рис. *20 Дерево компонентів веб-сторінки*

Існує два види реєстрації компонентів: локальна та глобальна. Компоненти, зареєстровані локально, можуть визиватися лише з компонентів в який вони зареєстровані, глобальні ж — з будь-якого компонента у дереві.

Vue.js використовує Virtual DOM. Зміни не вносяться одразу у DOM , спочатку створюється копія DOM дерева у формі JavaScript структури даних та якщо потрібно внести зміни у дерево, вони спочатку вносяться до JavaScript структури даних, порівнюються з початковим DOM, а потом змінюються лише

остаточні зміни у реальному DOM. Це дозволяє переписувати лише ті компоненти, які були змінені, а не завантажувати знову цілу сторінку, що також відображаться на швидкості роботи зі сторінкою.

Firebase. Firebase надає зрозумілі API для роботи з додатками, що значно спрощує та пришвидшує процеси розробки. Платформа надає аналітичні можливості для моніторингу статистики дій користувачів, роботи програми і відстеження критичних ситуації. Firebase кросплатформений завдяки пакетам розробника для Android, iOS, JavaScript, C++ та серверним бібліотекам чи REST API [20].

Сервіси Firebase [20]:

- Realtime Database, Firestore, Storage — це серверні хмарні бази даних. Працювати з ними базами можна в оффлайн режимы: всі зміни зберігаються, а потім вносяться до бази даних при вході в режим онлайн.
- Firebase Authentication — дозволяє налаштувати регістрацію та авторизацію користувачів, багатьма способами: за допомогою пошти та паролю або за допомогою акаунтів Facebook, Google, Twitter та інших.
- Firebase Cloud Messaging — дозволяє надсилати сповіщення (пушповідомлення) користувачам додатку .
- Test Lab, Crash Reporting — надає можливість тестування додатку на реальних та емульованих пристроях Google, виконувати моніторинг стану додатку на пристроях та виконувати перевірки з повним звітом про помилки.
- Firebase Analytics — надає повну структуровану інформацію про додаток та користувачів (регіони поширення, час роботи, витрати, кількість поломок та інше).
- Firebase App Indexing — це додавання створеного продукту в пошукові системи Google.

- Firebase Invites — надає можливість створення запрошень на встановлення створеного додатку, які можуть поширювати і вже існуючі користувачі.

Git. Контроль версій — це система, яка записує кожні зміни до файлу або кількох файлів, щоб згодом використати певні версії.

Git — це безкоштовна розподілена система контролю версій, що доступна на всіх основних платформах розробки та має відкритий вихідний код. Відмінною особливість Git є те, що вона розподілена. Це означає, що він швидкий, масштабований та має велику кількість команд для роботи з високо- та низькорівневими операціями [21].

Створення гілок допомагає задати новий напрямок в розробці проекту. Це дозволяє розробляти проект одразу у кількох напрямах, кількома людьми або одною людиною.

Репозиторії — це база даних типу ключ-значення, що містить всю інформацію про історію проекту та зберігає файли на всіх етапах його життєвого циклу.

Реалізація Git заснована на сховищі об’єктів, яке зберігає вихідні файли даних та всю інформацію журналу, необхідну для відновлення файлів.

Наявність локального репозиторія робить git простим у налаштуванні та надає можливість працювати з проектом без підключення до Інтернету.

GitHub. GitHub — це відомий постачальник послуг хостингу репозиторію git, який спрощує роботу репозиторіями завдяки веб-інтерфейсу користувача, а також допомагає в співпраці у проектах із відкритим кодом.

Twilio. Платформа, побудована на основі WebRTC, дозволяє створювати унікальні відеододаткі. Створення індивідуального відео за допомогою спеціальних макетів і віртуального фону за допомогою API та SDK, які працюють у всіх основних браузерах і пристроях.

SaSS. HTML і CSS формують вигляд веб-сторінок. CSS3 надав багато можливостей для стилізації сторінок, але це також ускладнило працю з ним та

його обслуговування. Крім складності, використання лише CSS, може призвести до дублювання коду та зменшення гнучкості додатку, що значно дається взнаки на його ефективність. Препроцесори CSS розширюють CSS код за допомогою сучасних концепцій мови програмування. Препроцесори CSS дозволяють використовувати змінні, функції, вкладенні правила або селектори. Препроцесори CSS надають можливість застосувати принцип «Don’t Repeat Yourself» (DRY) до коду CSS. Його дотримання допомагає уникнути повторення коду [22].

Sass (Syntacically Awesome Stylesheets) — це найвідоміший препроцесор.

При компілюванні SassScript генерує правила CSS для різних селекторів, як зазначено у файлі Sass. Sass може контролювати файли .sass, .scss і генерувати вихідний файл .css щоразу після їх збереження.

Sass з відкритим вихідним кодом, написаним на Ruby.

Axios. Axios — це HTTP клієнт, що використовує Promise за замовчуванням і працює на стороні клієнта і на стороні сервера. Це дозволяє завантаження даних під час рендерингу на сервері).

VueX. Vuex — це патерн керування станом, а також бібліотека для програм на Vue.js. Використовується как централізоване сховище даних для всіх компонентів з правилами, які гарантують, що стан може бути змінено лише передбачуваним чином. Vuex інтегрується з офіційним розширенням vue-devtools та надає можливості використання "машини часу" для налагодження та експорту або імпорт зліпків стану даних.

4.2 Вимоги до апаратного та програмного забезпечення

Мінімальні вимоги до апаратного та програмного забезпечення:

- Операційна система: Windows 7;
- Оперативна пам’ять: 6 Гб;
- Процесор: Core Intel або AMD.

- Доступ до Інтернету
- Наявність мікрофону
- Встановлений браузер

Рекомендовані вимоги до апаратного та програмного забезпечення:

- Операційна система: Windows 10, MacOS Big Sur
- Оперативна пам’ять: 16 Гб;
- Процесор: Intel Core i5 та вище;
- Наявність камери;
- Наявність мікрофону;
- Доступ до Інтернету
- Встановлений браузер Chrome 96, Safari 14

4.3 Опис функціональних можливостей

Рисунок 21 відображає UML діаграму мінімальних потреб до реалізації веб сайту.

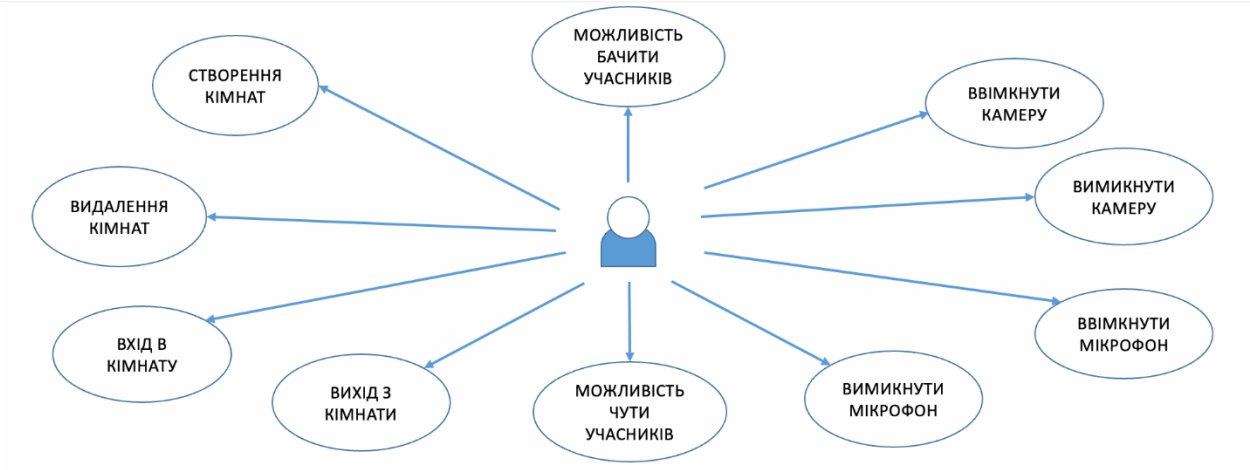

Рис.*21 Мінімальні потреби для реалізації*

4.4 Vue.js компоненти

App.vue. Головний компонент проекту наведено у лістингу 4. Він визиває компоненти верхнього та нижнього барів, а також тіло сторінки в залежності від посилання. Також компонент відповідає за виведення нотифікацій та зміну тем оформлення сайту.

Лістинг *4 Компонент App.vue*

```
<template>
  <div id="app2">
    <Header/>
  <div class="n">
    <notifications/>
  </div>
    <div id="router">
      <router-view></router-view>
    </div>
    <button id="themeBtn" @click="changeTheme">
      <svg v-if="theme == 'light'" id="themeIcon"></svg>
<svg v-if="theme == 'dark'" id="themeIcon"></svg>
</button>
    <Footer />
  </div>
</template>
```

Header.vue. У лістингу 5 наведено компонент, що відповідає за відображення верхньої частини сайту в залежності від сторінки та стану авторизації користувача.

Лістинг *5 Компонент Header.vue*

```
<template>
<div id="header">
  <div class="nameSection" v-if="current_user">
    <span class="h">{{ current_user.displayName }}</span>
```

```
  </div>
  <div class="btnSections" >
    <button v-on:click="showSignUp" class="btnHead" v-
if="!current_user">Sign Up</button>
    <sign-up-form v-if="visibleSignUpForm" v-
on:close="showSignUp" :users="users"/>
    <button v-on:click="showLogIn" class="btnHead" v-
if="!current_user">Log In</button>
    <log-in-form v-if="visibleLogInForm" v-
on:close="showLogIn"/>
    <logout v-if="current_user"/>
  </div>
</div>
</template>
```

Landing сторінка має Log In та Sign Up кнопки. Main Page містить ім’я користувача та кнопку Log Out для виходу з системи. Room Page містить назву кімнати, ім’я користувача та власника кімнати, а також кнопку Log Out для виходу з системи.

Modal.vue. Компонент модального вікна наведено у лістингу 5. Модальне вікно зроблено так, щоб воно могло перевикористовуватися для інших компонентів системи.

Лістинг *5 Компонент Modal.vue*

```
<template>
  <transition name="modal-fade">
    <div class="modal-backdrop">
      <div class="modal"
        role="dialog"
        aria-labelledby="modalTitle"
        aria-describedby="modalDescription"
```

```
      >
        <header
          class="modal-header"
          id="modalTitle"
        >
          <slot name="header"></slot>

        </header>
        <section
          class="modal-body"
          id="modalDescription"
        >
          <slot name="body"></slot>
        </section>
        <footer class="modal-footer"><slot name="footer">

          </slot>
        </footer>
      </div>
    </div>
  </transition>
</template>
```

Signup.vue. Форм реєстрації користувача, наведена у лістингу 6, вбудовується у компонент модального вікна. Містить валідацію полів та виконує запис нових користувачів до бази даних Firebase за допомогою функції реєстрації у лістингу 7.

Лістинг 6 *Компонент Signup.vue*

```
<template>
<modal @close="showSignUp()">
```

```
  <template v-slot:header>
    <button class="btn-close" @click="close</button>
  </template>
  <template v-slot:body>
    <p class="h">Sign Up</p>
<form>
    <label><span for="email">Email</span>
    <input type="text" v-model.trim="email"
@input="clearErrorEmail()" name="email"
class="inputFields"/></label>
    <label><span for="username">Username</span>
    <input type="text" v-model.trim="username"
@input="clearErrorUsername()" name="username"
class="inputFields"/>/label>
  <label><span for="password">Password</span>
    <input type="password" v-model.trim="password"
@input="clearErrorPassword()"
name="password"/></label></form></template>
  <template v-slot:footer>
    <button @click="checkSignUp()" type="submit"
class="btnFill">Sign Up</button>
  </template></modal></template>
```

Лістинг *7 Функція реєстрації користувача*

```
register() {
firebase.auth().createUserWithEmailAndPassword(this.email,
this.password)
      .then(() => {
        firebase.auth().currentUser.updateProfile({
          displayName: this.username
        }).then(() => {
```

```
          this.$notify({ group: 'success', text: "Your
account has been successfully created"});
          this.close();
        }).catch((error) => {
          this.$notify({ group: 'error', text: error});
})
      })
      .catch((error) => {
        this.$notify({ group: 'error', text:
error.message});
        });
     },
```

Login.vue. Компонент входу до системи. Містить поля Email та Password. На front-end виконується перевірка на пусті значення полів. Front-end обробляє наявність користувача за даними параметрами. У лістингу 8 наведено функцію перевірки наявності користувача в системі.

Лістинг *8 Функція перевірки наявності користувача в системі*

```
submit() {
firebase.auth().signInWithEmailAndPassword(this.email,
this.password)
          .then(() => {
            const user = firebase.auth().currentUser;
            this.setCurrentUser(user)
            this.close();
            console.log("Login " + user.refreshToken);
            this.$router.push({path: '/rooms'});
          })
          .catch((error) => {
            this.$notify({type:'error', text:
error.message});
```

```
        });
```

Logout.vue. Компонент відповідає за вихід з системи, функція наведена у лістингу 9. Після натискання кнопки Log out Firebase разлоговує користувача на сервері, VueX Store обнуляє значення користувача на front-end, а сам користувач повинен потрапити на сторінку Landing з кнопками Log In та Sign Up у верхній частині сайту.

Лістинг *9 Функція виходу з системи*

```
logout(){
 firebase.auth().signOut().then(() => {
 this.setCurrentUser(null)
 this.$router.push({path: '/'});
 }).catch((error) => {
    this.$notify({ group: 'error', text: error})});
```

Footer.vue. Компонент, що відповідає за відображення нижньої частини сайту (підвалу). Містить інформацію про сайт, а також посилання у соцмережі.

Landing.vue. Єдина сторінка, що доступна незалогованному користувачу. Містить назву сервісу та кнопки для реєстрації та входу до системи.

CreateRoom.vue. Компонент для створення кімнати. Поле вводу містить валідацію. Неможливо додати пусту кімнату або створення кімнати незалогованним користувачем. Функція створення кімнати наведена на лістингу 10.

Лістинг *10 Функція створення кімнати*

```
createRoom() {
        const user = firebase.auth().currentUser;
        if(user){
            if(!this.newRoomName){
            this.createRoomError = "Room name should not be
empty";
```

```
            this.$notify({ type:'error', text: "Room name
should not be empty"});
        } else{
            this.getTime();
            var room = {
                roomName: this.newRoomName,
                creator: user.displayName,
                time: this.timestamp,
                status: "active" }

            db.collection("rooms").add(room)
                .then((docRef) => {
                    this.$emit('addRoom', {id: docRef.id,
...room});
                })
                .catch((error) => {
                    this.$notify({type:'error', text:
"Error adding room"});
                });
                this.newRoomName = null
        }}
        else{
            this.$notify({type:'error', text: "User is not
logged in to add room"});
        }
    },
```

RoomsList.vue. Компонент виводить список усіх існуючих кімнат. Власнику кімнати надається можливість видалити тільки свою кімнату, функція наведена у лістингу 11.

Лістинг *11 Функція видалення кімнати користувачем*

```
deleteRoom(roomId){
    db.collection("rooms").doc(roomId).delete().then(() =>
    {
      this.rooms.forEach((room, index) => {
        if(room.id == roomId) this.rooms.splice(index, 1);
        });
        this.$notify({ type: "success", text: "Room has
been deleted" }, 2000);
      }).catch((error) => {
        this.$notify({ type:'error', text: "error"});
      });
  },
```

Функція отримання усіх активних кімнат з бази даних наведена у лістингу 12.

Лістинг *12 Функція отримання кімнат з бази даних*

```
getRooms() {
db.collection("rooms").where("status","==","active").orderB
y("time").get()
    .then((snapshot) => {
      snapshot.docs.forEach(doc => {
      const docMap = {  ['id']: doc.id , ...doc.data()}
      this.rooms.push(docMap);
      })});
  },
```

RoomsPage.vue. Поєднує компоненти CreateRoom та RoomsPage для цілостного вигляду сторінки та містить функцію додавання кімнати, що викликається у дочірньому компоненті CreateRoom та наведена у лістингу 13.

Лістинг *13. Функція додавання кімнаті до бази даних.*

```
addRoom(docRef){
this.rooms.push(docRef);
this.$notify({ type: 'success', text: "Room was added"});}
```

Room.vue. Компонент, що відповідає за створення токену, за підключення завдяки ньому до Twilio для подальшої роботи. При першому переході на сторінку генерується токен та створюється кімната на Twilio. При кожному підключені учасника виконується захоплення його локальних пристроїв, функція наведена у лістингу 14, додається плитка з аудіо та відео.

Лістинг *14 Функції захоплення локальних аудіо та відео*

```
createTracks(){
      createLocalVideoTrack().then((track) => {
      this.videoTrack = track;
      setTimeout(() => {
        this.getDevices()
      }, 1000)
      setTimeout(() => {
        this.videoAttach(track)
      }, 1500)
    });
    createLocalAudioTrack().then((track) => {
      this.audioTrack = track;
      setTimeout(() => {
        this.audioAttach(track)
      }, 1000)

    });
  },
```

DeviceOptions.vue. Компонент складається з випадаючого списку, що містить в собі доступні пристрої для зміни в залежності від типу: мікрофон або камера. Функція зміни пристрою наведена у лістингу 15.

Лістинг *15 Функція зміни пристрою*

```
changeDevice(device, kind){
    const localParticipant = this.room.localParticipant;
    console.log(device);
    if(kind === 'videoinput' ){
    createLocalVideoTrack({ deviceId: { exact:
device.deviceId }})
      .then((localVideoTrack) => {
          this.videoTrack = device
          localParticipant.videoTracks.forEach(e => {
            e.track.stop();
            e.unpublish();
            e.track.detach()
          })
 this.room.localParticipant.publishTrack(localVideoTrack)
        setTimeout(() => {
          var el = document.querySelector('#localTrack')
            if(el) {
              el.innerHTML = ""
              el.appendChild(localVideoTrack.attach())
            } else{
              setTimeout(() => {
                el = document.querySelector('#localTrack')
                el.innerHTML = ""
                el.appendChild(localVideoTrack.attach())
              }, 2000)
            }
              }, 1000)
```

```
        })
        }else if(kind === 'audioinput'){
          createLocalAudioTrack({ deviceId: { exact:
device.deviceId }})
      .then((localTrack) => {
          this.audioTrack = device
          localParticipant.audioTracks.forEach(e => {
            e.track.stop();
            e.unpublish();
            e.track.detach()
          })
        this.room.localParticipant.publishTrack(localTrack)
        setTimeout(() => {
          var el =
document.querySelector('#localAudioTrack')
            if(el) {
              el.innerHTML = ""
              el.appendChild(localTrack.attach())
            } else{
              setTimeout(() => {
                el =
document.querySelector('#localAudioTrack')
                el.innerHTML = ""
                el.appendChild(localTrack.attach())
              }, 2000)
            }
              }, 1000)
        })
        }},
```

Participants.vue. Компонент відповідає за приєднання и від'єднання учасників та створення їх плиток. Функції відстеження під'єднання та від'єднання наведені у лістингу 16.

Лістинг *16 Функції відстеження під'єднання та від'єднання учасників*

```
subscribeParticipants() {
this.room.on("participantConnected", (participant) => {
this.checkParticipantsList()});
this.room.on("participantDisconnected", (participant)=> {
this.detachParticipantTracks(participant);
this.enabledParticipants =
this.enabledParticipants.filter(f => f.sid !==
participant.sid);
            this.checkParticipantsList()
```

ParticipantTile.vue. Компонент містить налаштування для окремих плиток. В компоненті реалізовані функції поновлення з'єднання, перепідключення аудіо та відео доріжок за для уникнення появи чорних екранів та непрацюючих плиток. У лістингу 17 наведена одна з таких функцій — функція підписання на доріжки користувача.

Лістинг *17 Функція підписання на доріжки учасника*

```
subscribeToParticipant() {
      let participant = this.findParticipant()
      participant.on('trackPublished', track => {
        setTimeout(() => {
          this.initTrack(track, false)
        }, 1000)
      })
      participant.on('trackUnpublished', track => {
        track.removeAllListeners()
        track?.track?.detach().forEach(function (element) {
```

```
            element.srcObject = null;
            element.remove();
          });
```

Main.js. Головний js файл системи, що містить створення та підключення компонентів наведений у лістингу 18.

Лістинг *18 Main.js*

```
Vue.use(Notifications)
Vue.use(VueRouter)
Vue.use(firestorePlugin)
Vue.config.productionTip = false
Vue.component('sign-up-form', Signup)
Vue.component('log-in-form', Login)
Vue.component('modal', Modal)
Vue.component('Header', Header)
Vue.component('Footer', Footer)
Vue.component('RoomsPage', RoomsPage)
Vue.component('Notification', Notification)

firebase.auth().onAuthStateChanged(user => {
if (!app) {
  app = new Vue({
    render: h => h(App),
    router,
    store
  }).$mount('#app');
  }
});
```

router.js. Файл скриптів, що відповідає за маршрутизацію сторінок. Для налагодження маршрутів використовувалася офіційна бібліотека VueRouter. Маршрути проекту наведені у лістингу 19.

Лістинг *19 Маршрути веб-сайту*

```
export default new VueRouter({
    routes: [
    {
        path: '',
        component: Landing
    },
    {
        path: '/rooms',
        component: Rooms
    },
    {
        path: '/room/:id',
        component: Room
    }
    ],
    mode: 'history'

})
```

firebase.js. Файл скриптів, що відповідає за підключення сервісів Firebase до проекту.

store/index.js. Файл скриптів, що відповідає за збереження поточного користувача для подальшої автентифікації.

main.scss. Головний файл стилів підключений до HTML файлу проекту. Підключення інших файлів стилів до проекту виконується через цей файл.

colors.scss. Файл стилів, що відповідає за значення змінних кольорів в залежності від сторінок та обраних тем. Приклад налаштувань змінних для світлої теми наведено у лістингу 20, для темної теми у лістингу 21.

Лістинг *20 Світла тема*

```
body[theme="light"]{
    --fontColor: white;
    --border: 1px solid white;
    --fillBtn: white;
    --sectionOpacity:rgba(16 18 27 / 40%);
    --box-shadow: -1px 3px 8px -1px rgba(0, 0, 0, 0.2);
}
body[theme="light"][page="main"]{
    --theme: linear-gradient(-45deg, #ee7752, #e73c7e,
#23a6d5, #23d5ab);
}
body[theme="light"][page="room"]{
    --theme: #353046;
}
body[theme="light"][page="landing"]{
    --theme: #887594;
}
```

Лістинг *21 Темна тема*

```
body[theme="dark"]{
    --fontColor: black;
    --border: 1px solid black;
    --fillBtn: white;
    --sectionOpacity:rgba(255 255 255 / 40%);
    --box-shadow: -1px 3px 8px -1px rgba(0, 0, 0, 0.2);
```

```
}
body[theme="dark"][page="main"]{
    --theme: linear-gradient(to bottom right,#688393,
#EA8E7D);
}
body[theme="dark"][page="room"]{
    --theme: #707C89;}
```

notification.scss. Файл стилів для відображення повідомлень на сайті. На сайті існує 4 типа повідомлень: без типу, успіх, попередження, помилка. Кожен тип відрізняється за кольором. Стилі нотифікацій наведені у лістингу 22.

Лістинг *22 Стилі нотифікацій*

```
.vue-notification {
    background-color: rgba(16, 18, 27, 0.7);
    box-shadow: -1px 3px 8px -1px rgba(0, 0, 0, 0.2);
     border-radius: 20px;
    padding: 20px;
    margin: 5px 5px 5px;
    font-size: 15px;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-
serif;
    color: #ffffff;
    border-left: 5px solid #187FE7;
    &.warn {
      border-left-color: #f48a06;
      background-color: rgba(16, 18, 27, 0.7);
      box-shadow: -1px 3px 8px -1px rgba(0, 0, 0, 0.2);
      border-radius: 10px;
    }

    &.error {
      background-color: rgba(16, 18, 27, 0.7);
```

```
      box-shadow: -1px 3px 8px -1px rgba(0, 0, 0, 0.2);
      border-radius: 10px;
      border-left-color: #B82E24;
    }
    &.success {
      border-left-color: #42A85F;
      background-color: rgba(16, 18, 27, 0.7);
      box-shadow: -1px 3px 8px -1px rgba(0, 0, 0, 0.2);
      border-radius: 10px;
    }
  }
```

Header.scss. Файл стилів містить стилі для верхньої частини сайту, включаючи реєстрацію та вхід до системи.

Footer.scss. Файл стилів містить стилі для нижньої частини сайту.

CreateRoom.scss. Файл стилів для секції створення нової кімнати.

RoomsList.scss. Файл стилів для відображення списку існуючих кімнат на сайті.

4.5 Інтерфейс додатку

4.5.1 Прототип

На рисунку 22 зображено прототип лендінгу. На ній користувач зможе зареєструватися або ввійти у систему.

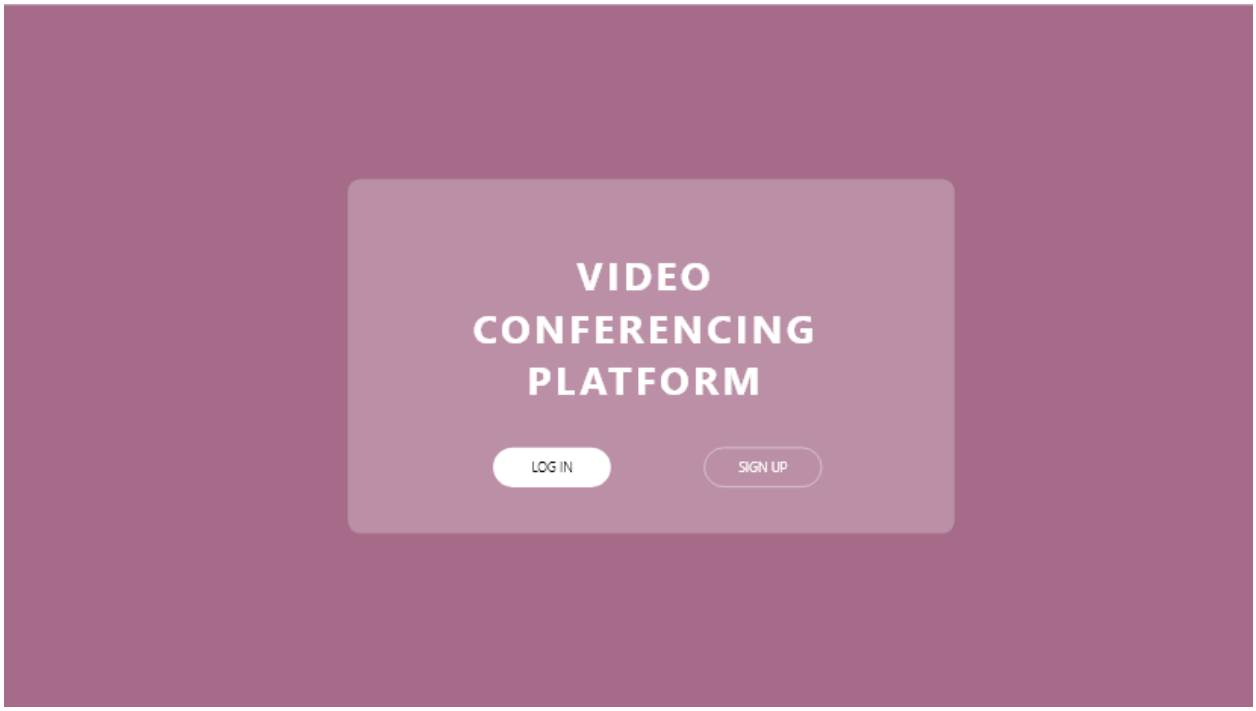

Рис.*22 Прототип лендінгу*

На рисунках 23 та 24 зображено прототип головної сторінки в різних темах. Ця сторінка доступна тільки авторизованому користувачу. На ній користувач може вийти з системи, створити свою кімнату або приєднатися до існуючої.

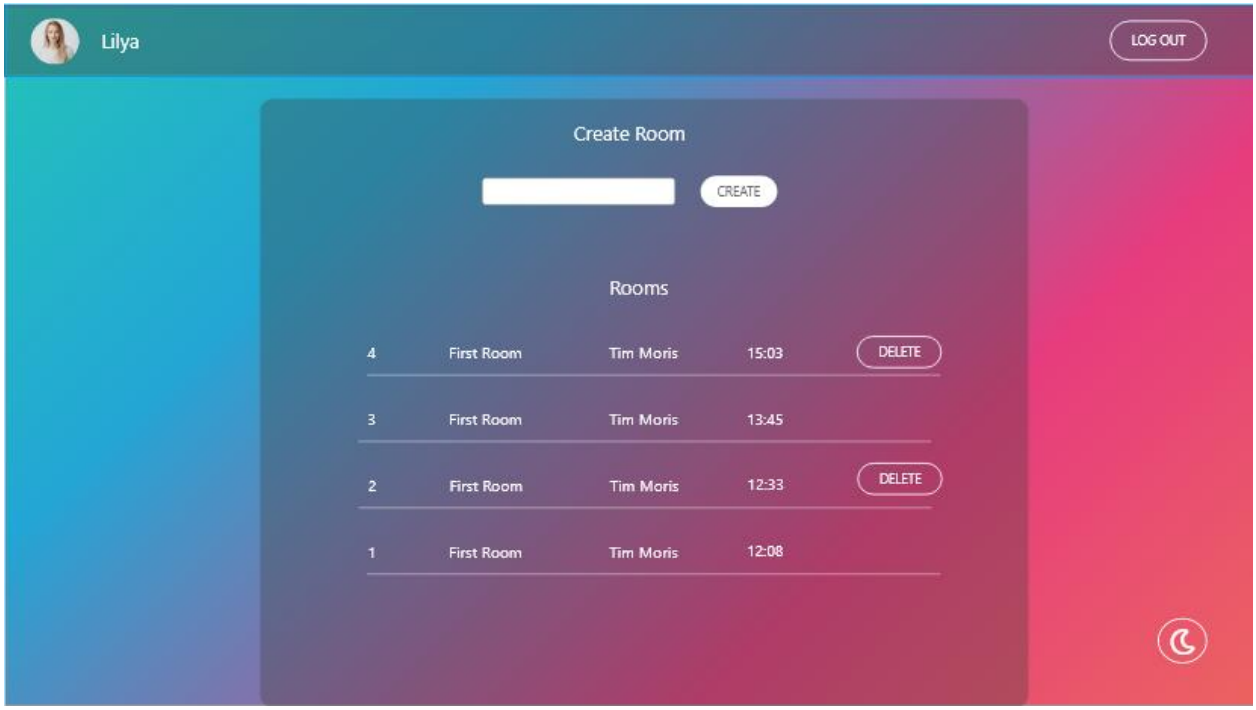

Рис. *23 Прототип головної сторінки у світлій темі*

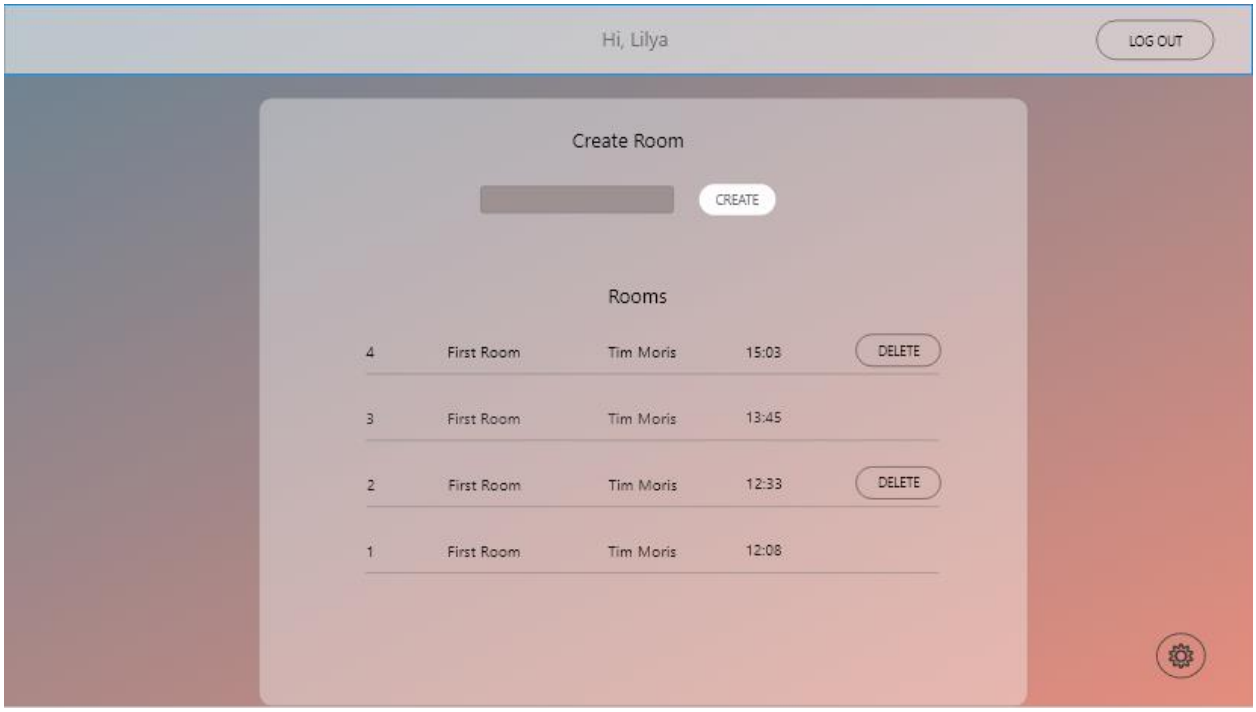

Рис. *24 Прототип головної сторінки у темній темі*

На рисунках 25 та 26 зображені прототипи кімнати в різних темах.

Рис. *25 Прототип кімнати у світлій темі*

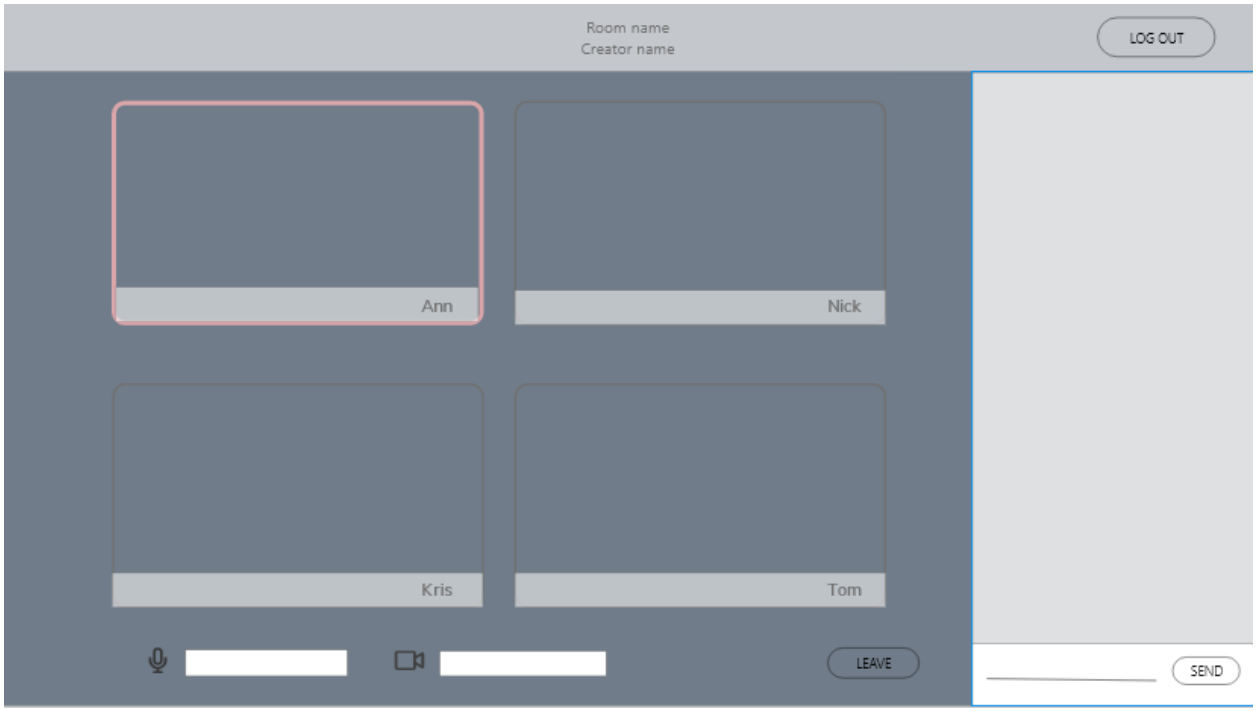

Рис. *26 Прототип кімнати у темній темі*

4.5.2 Реалізація

Сторінка Landing зображена на рисунку 27.

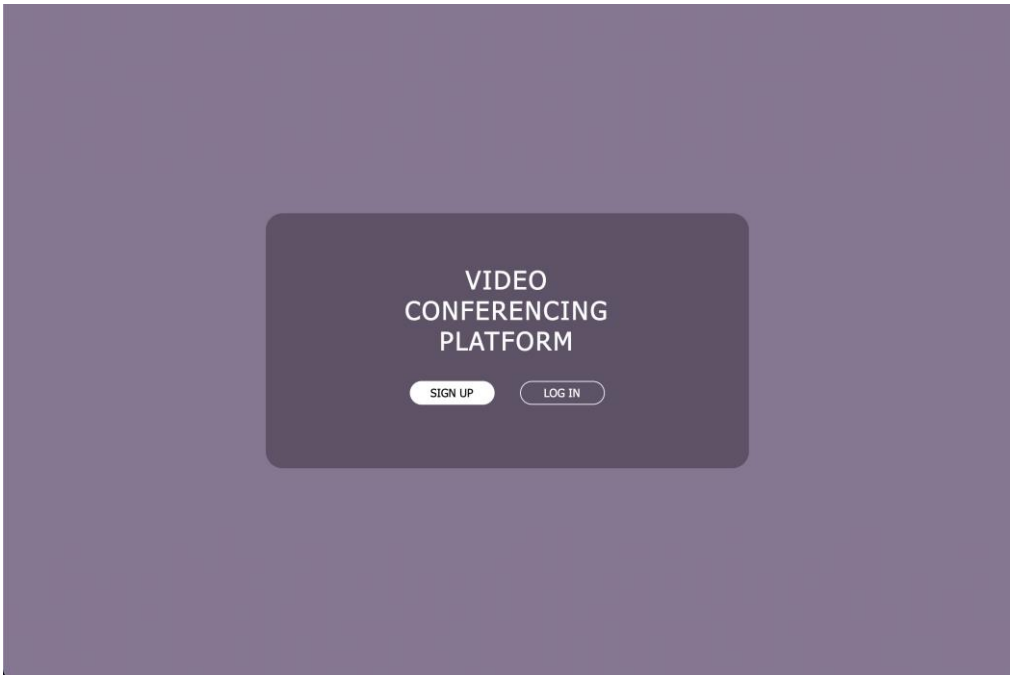

Рис.*27 Landing*

Модальне вікна для реєстрації в систему зображено на рисунку 28. Воно містить поля електронної пошти, назви користувача та паролю.

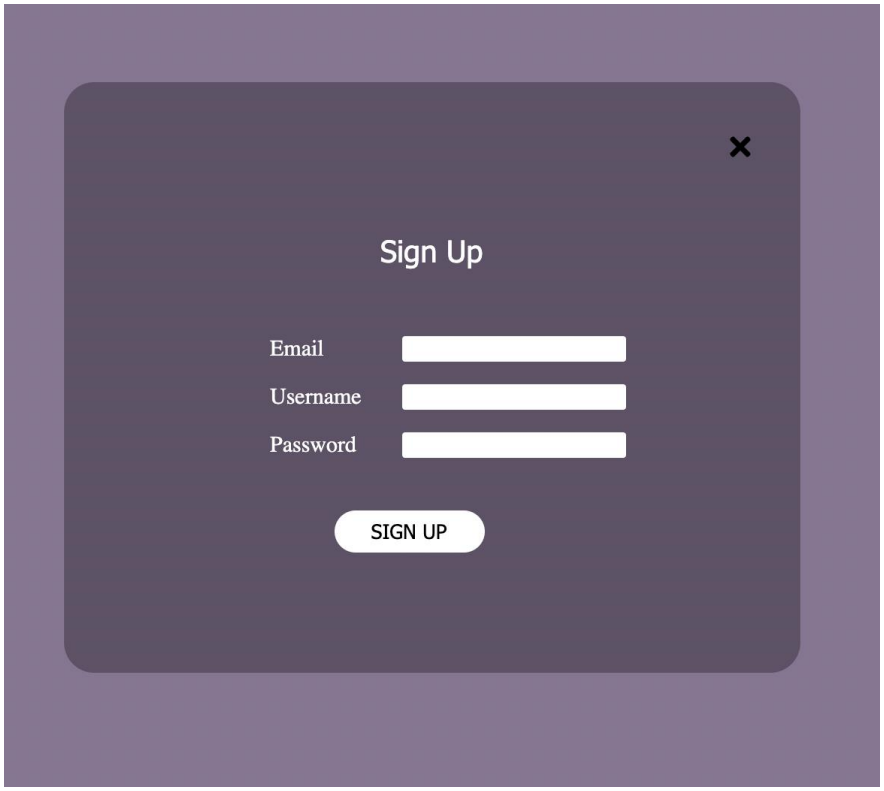

Рис.*28 Модальне вікно реєстрації*

Вигляд бази даних зберігаємих кімнат зображено на рисунку 29.

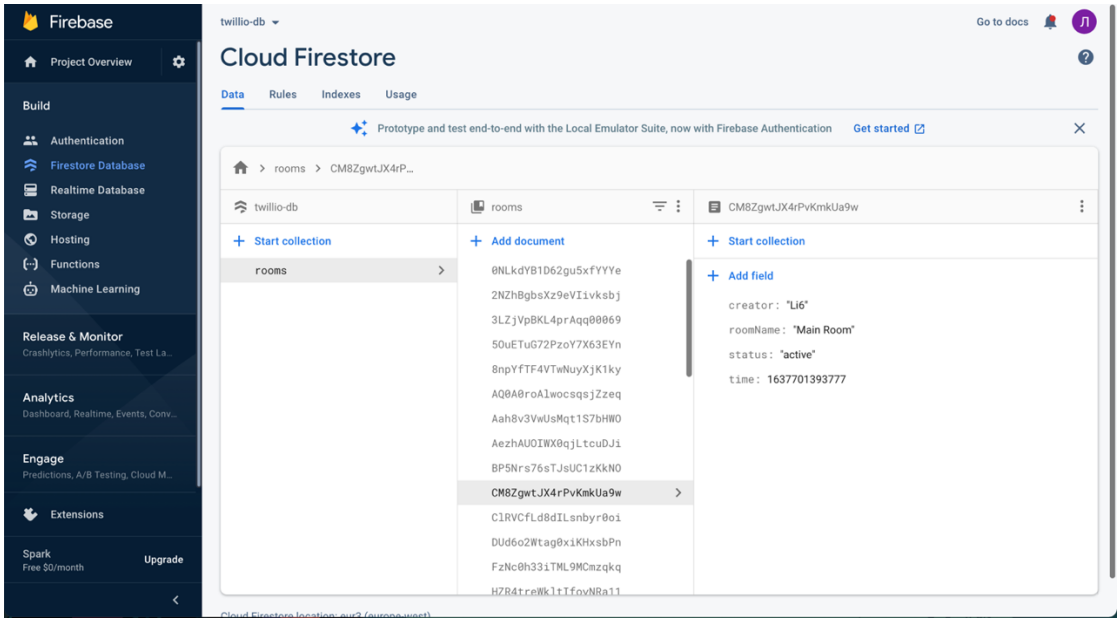

Рис.*29 База даних Firebase*

Форма входу до системи зображена на рисунку 30.

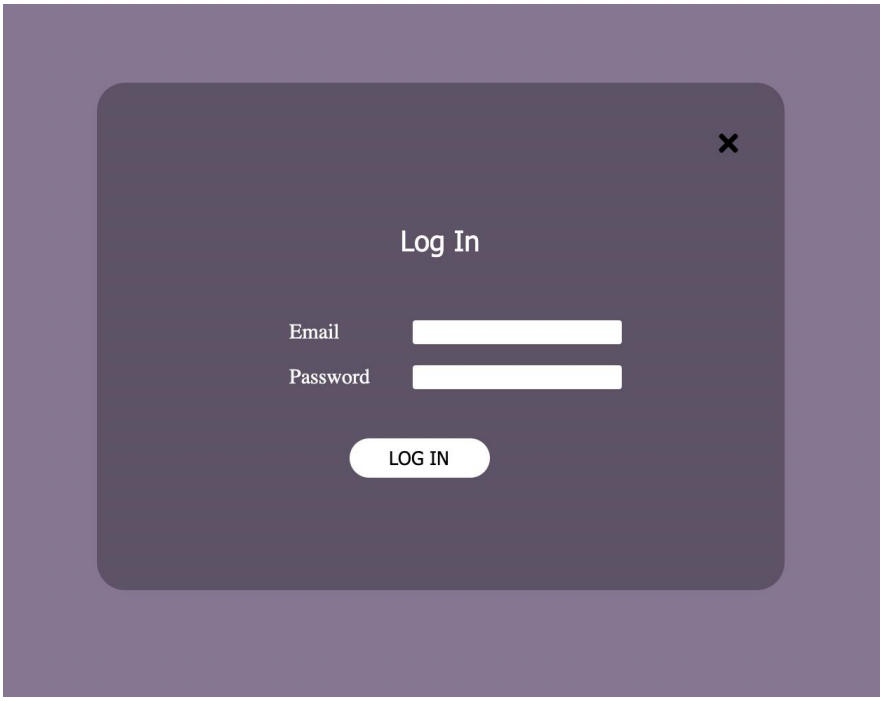

Рис.*30 Модальне вікно входу до системи*

На рисунку 31 зображена головна сторінка, що включає компоненти створення та виводу кімнат у світлій темі. На рисунку 32 та ж сторінка, але у темній темі.

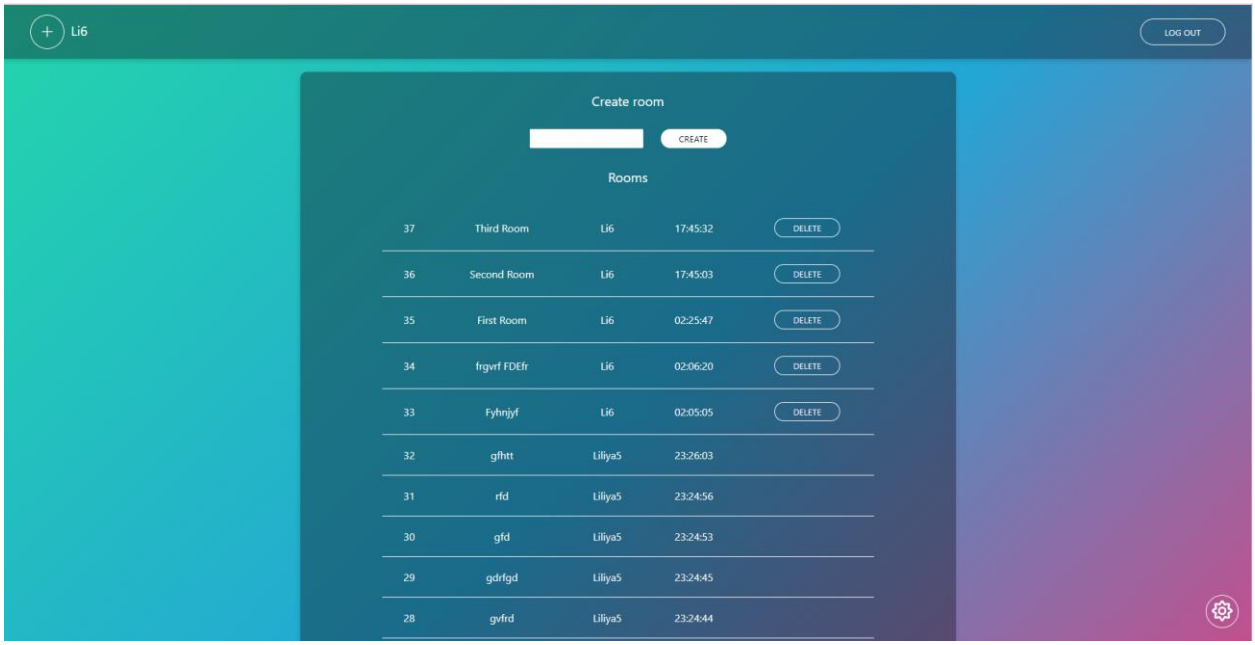

Рис.*31 Список кімнат тема 1*

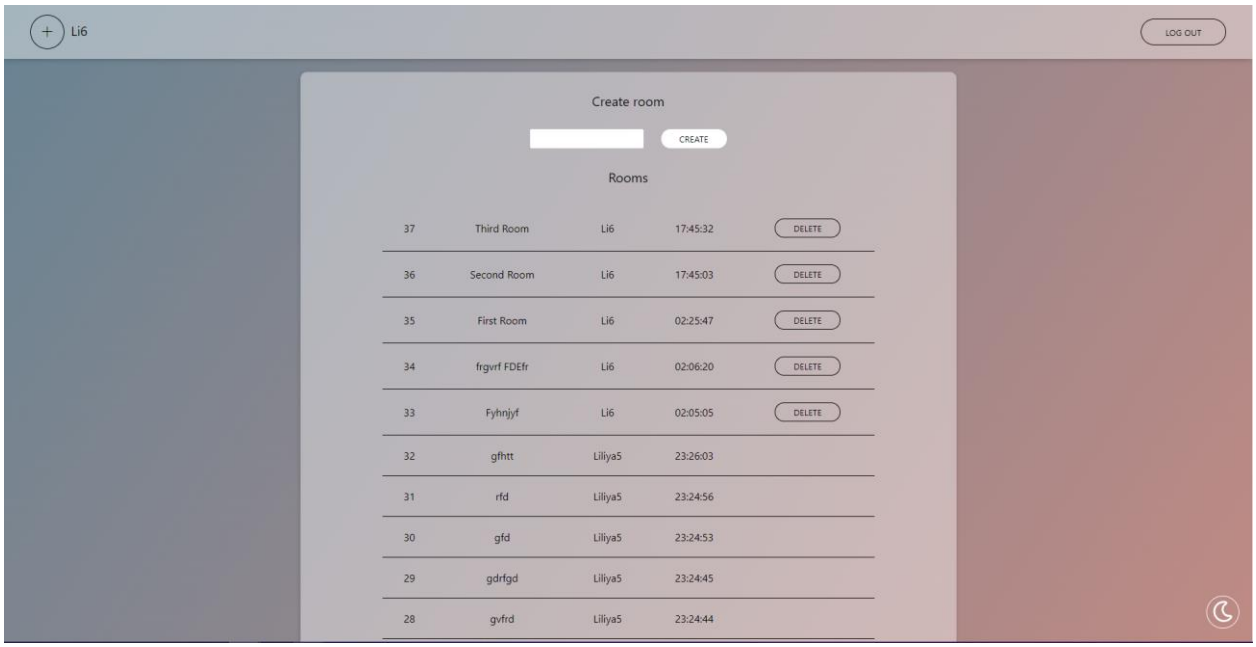

Рис.*32 Список кімнат тема 2*

Сайт має систему нотифікацій, вигляд якої зображено на рисунках 33 та 34.

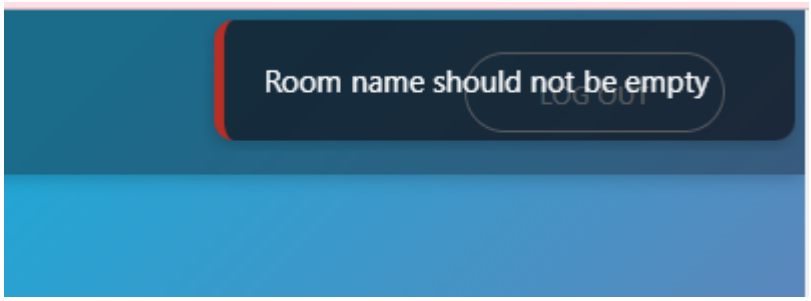

Рис.*33 Нотифікація помилки*

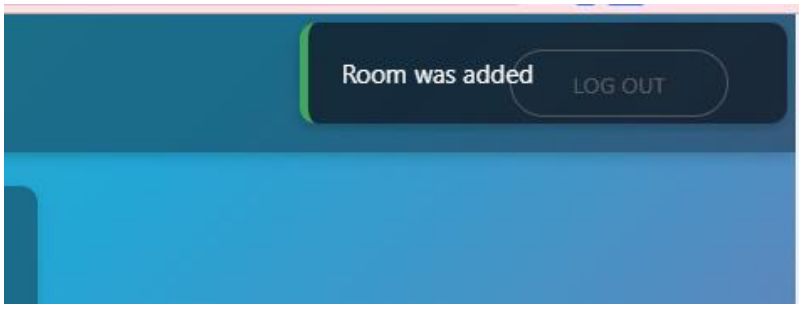

Рис.*34 Нотифікація успіху*

На рисунку 35 зображена кімната з під’єднаним користувачем.

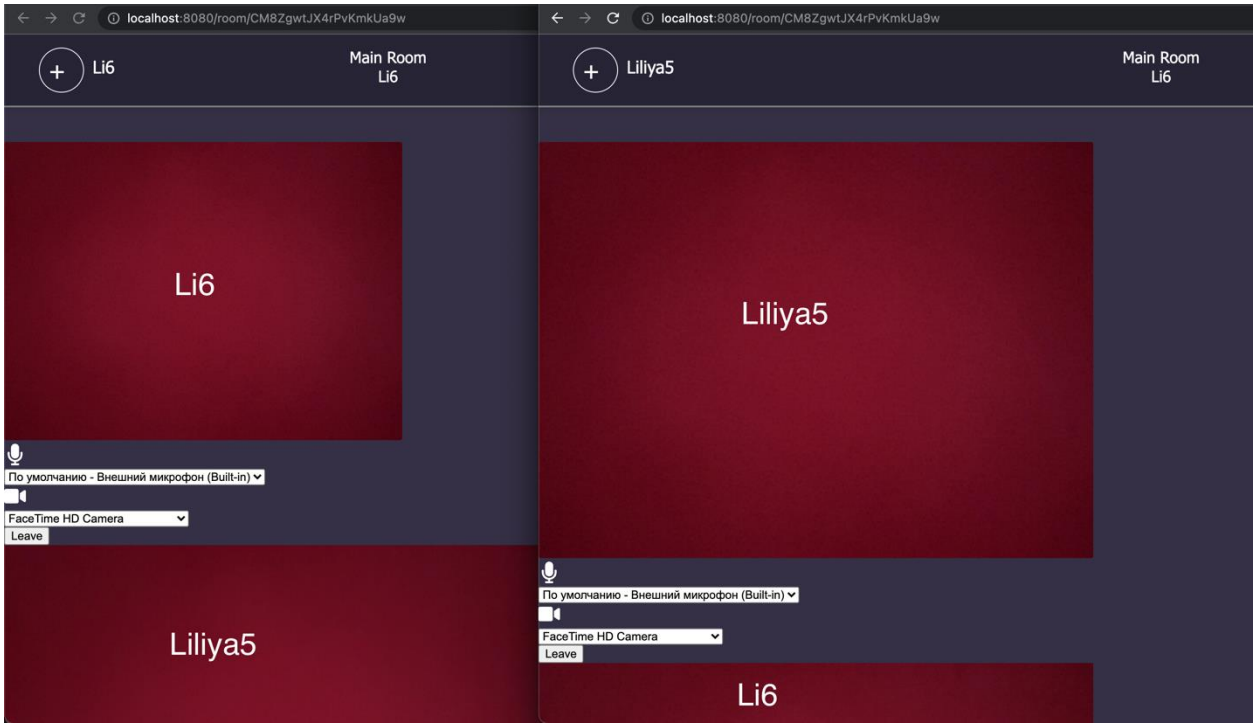

Рис.*35 Сторінка кімнати*

ВИСНОВКИ

В рамках магістерської роботи отримані наступні результати:

1. Проведено дослідження існуючих інструментів для створення відеострімінгового сервісу.
2. Проаналізовано сучасні відеострімінгові сервіси.
3. Розглянуто та проаналізувала переваги та недоліки розглянутих інструментів.
4. Визначено вимоги до розроблюваного сервісу.
5. Ознайомлено з принципами роботи фреймворку Vue.js.
6. Отримано досвід роботи з хмарним сховищем Firebase.
7. Покращено навички роботи з системою контролю версій Git.
8. Розроблено веб-сайт з інтегрованим Twilio API для проведення відеоконференцій за допомогою потокової передачі даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Правдивець Л.М., магістрант, Лимаренко Ю. О., доцент, к.ф.-м.н — науковий керівник. Дослідження інструментів для створення та інтегрування відеострімінгових сервісів на веб-сайт. Збірник наукових праць студентів, аспірантів, докторантів і молодих вчених. *Молода наука-2021*. Запоріжжя: ЗНУ, 2021. Т.5. С. 92-94.
2. Правдивець Л.М., магістрант, Лимаренко Ю. О., доцент, к.ф.-м.н — науковий керівник. Дослідження принципів створення і інтеграції відеострімінгових сервісів на Web-сайт. Матеріали І Всеукраїнської науково-практичної конференції здобувачів вищої освіти, аспірантів та молодих вчених. Актуальні питання сталого наукового-технічного та соціально-економічного розвитку регіонів України. Запоріжжя : ЗНУ, 2021. С. 338-339.
3. What is live streaming. URL: https://www.cloudflare.com/learning/video/what-is-live-streaming/ (дата звернення: 28.06.2021).
4. Zhu Z., Yang Z., Dai Y. Understanding the gift-sending interaction on live-streaming video websites //International Conference on social computing and social media. – Springer, Cham, 2017. – С. 274-285.
5. What is buffering. URL: https://www.cloudflare.com/learning/video/what-is-buffering/ (дата звернення: 28.06.2021).
6. Rodriguez-Gil, L., Orduña, P., García-Zubia, J. et al. Interactive live-streaming technologies and approaches for web-based applications. Multimed Tools Appl 77, 6471–6502 (2018).
7. Twitch About. URL: https://www.twitch.tv/p/en/about/ (дата звернення: 28.06.2021).
8. YouTube Live Streaming API Overview. URL: https://developers.google.com/youtube/v3/live/getting-started (дата звернення: 28.06.2021).

9. Live Video API. URL: https://developers.facebook.com/docs/live-video-api/ (дата звернення: 28.06.2021).
10. Zoom. URL: https://explore.zoom.us/ru/products/meetings/ (дата звернення: 08.11.2021).
11. Google Meet. URL: https://apps.google.com/intl/ru/intl/ru_ALL/meet/how-it-works/ (дата звернення: 08.11.2021).
12. Cisco Webex Meetings. URL: https://www.webex.com/video-conferencing (дата звернення: 08.11.2021).
13. Bing B. Next-generation video coding and streaming. – John Wiley & Sons. – 2015. – С. 10-13.
14. FPS for Live Streaming: The Advanced Guide to Video Frame Rates. URL: https://www.dacast.com/blog/frame-rate-fps/ (дата звернення: 28.06.2021).
15. Fouladi S. et al. Encoding, fast and slow: Low-latency video processing using thousands of tiny threads //14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17). – 2017. – С. 367.
16. Real-time communication for the web. URL: https://webrtc.org (дата звернення: 08.11.2021).
17. JavaScript Platform Overview. URL: https://www.twilio.com/docs/video/javascript-getting-started (дата звернення: 08.11.2021).
18. Build HQ Live Video Calling Experiences With MirrorFly Video Call API & SDK for Web & Mobile Apps. URL: https://www.mirrorfly.com/video-call-solution.php (дата звернення: 08.11.2021).
19. What is Vue.js. URL: https://vuejs.org/v2/guide/ (дата звернення: 08.11.2021).
20. Learn Firebase fundamentals. URL: https://firebase.google.com/docs/guides (дата звернення: 08.11.2021).
21. Chacon S., Straub B. Pro git. – Springer Nature, 2014. – С. 456.
22. Prabhu A. Beginning CSS Preprocessors: With SASS, Compass. js and Less. js. – Apress, 2015.

Декларація

академічної доброчесності

здобувача ступеня вищої освіти ЗНУ

Я, Правдивець Лілія Миколаївна, студент_2_курсу, форми навчання денної, Інженерного навчально-наукового інституту, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти lilu12421@gmail.com, — підтверджую, що написана мною кваліфікаційна робота на тему «**Дослідження принципів створення і інтеграції відеострімінгових сервісів на web-сайт**» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст.42 Закону України «Про освіту», зі змістом яких ознайомлений.

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-систем, а також на архівування моєї роботи в базі даних цієї системи.

Дата 30.11.2021 Підпис_____ Правдивець Лілія Миколаївна

(студент)

Дата 30.11.2021 Підпис_____ Лимаренко Юлія Олексіївна

(науковий керівник)