

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем і технологій



Створення вебзастосунку для полегшення комунікації між
викладачами та студентами

Текстова частина до курсової роботи за спеціальністю 121 «Інженерія
програмного забезпечення»

Керівник курсової роботи
асистент Смиш О. Р.

_____ (підпис)

“ ____ ” _____ 2023 р.

Виконала студентка Буденкова Маргарита Іллівна

“ ____ ” _____ 2023 р.

Київ 2023

ЗМІСТ

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ РОБОТИ	5
АНОТАЦІЯ.....	6
ВСТУП	7
РОЗДІЛ 1. ДОСЛІДЖЕННЯ	9
1.1 РЕЗУЛЬТАТИ	9
<i>Рисунок 1. Результати опитування 1.</i>	9
<i>Рисунок 2. Результати опитування 2.</i>	10
<i>Рисунок 3. Результати опитування 3.</i>	10
<i>Рисунок 4. Результати опитування 4.</i>	11
<i>Рисунок 5. Результати опитування 5.</i>	11
<i>Рисунок 6. Результати опитування 6.</i>	12
1.2 ДОСЛІДЖЕННЯ КОРЕЛЯЦІЇ	12
<i>Рисунок 7. Дослідження кореляції 1.</i>	12
<i>Рисунок 8. Дослідження кореляції 2.</i>	13
РОЗДІЛ 2. ОГЛЯД РІШЕНЬ, ЩО ІСНУЮТЬ	14
2.1 GOOGLE FORMS	14
2.2 MICROSOFT TEAMS.....	14
2.3 GOOGLE MEET	15
2.4 REMIND.....	16
2.5 TELEGRAM.....	17
2.6 ДОЦІЛЬНІСТЬ РОЗРОБКИ	17
РОЗДІЛ 3. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ ТА ІНСТРУМЕНТІВ	19
2.1 JAVASCRIPT.....	19
2.1 <i>Node.js</i>	20
2.3 AXIOS	20
2.2 <i>React.js</i>	21
2.3 <i>React Router</i>	21
2.4 <i>React-Bootstrap</i>	22
2.5 <i>MySQL</i>	23
2.6 <i>Sass</i>	23
2.6 MATERIAL UI	24
РОЗДІЛ 4. СТРУКТУРА БАЗИ ДАНИХ	26
3.1. EER-МОДЕЛЬ	26
<i>Рисунок 9. EER-модель застосунку.</i>	26
3.2 РЕЛЯЦІЙНА МОДЕЛЬ.....	26
РОЗДІЛ 5. ТЕХНІЧНІ МОЖЛИВОСТІ ЗАСТОСУНКУ	31

4.1	ЛОГІН І РЕЄСТРАЦІЯ	31
	<i>Рисунок 10. Сторінка реєстрації.....</i>	<i>32</i>
4.2	СТВОРЕННЯ КІМНАТ.....	32
	<i>Рисунок 11. Головна сторінка профілю користувача.....</i>	<i>32</i>
	<i>Додаток 1. Створення коду для кімнати.....</i>	<i>33</i>
4.3	АНОНІМНІ ПОВІДОМЛЕННЯ.....	33
	<i>Рисунок 12. Надсилання анонімних повідомлень</i>	<i>33</i>
	<i>Рисунок 13. Приклад відправки анонімного повідомлення.....</i>	<i>34</i>
4.4	ВИДАЛЕННЯ І РЕДАГУВАННЯ ПОВІДОМЛЕНЬ	34
	<i>Рисунок 14. Структура повідомлення</i>	<i>34</i>
	<i>Рисунок 15. Вікно редагування повідомлення.....</i>	<i>35</i>
4.5	РЕАКЦІЯ НА ПОВІДОМЛЕННЯ	35
	<i>Рисунок 16. Повідомлення з реакцією</i>	<i>36</i>
4.6	СТВОРЕННЯ ЧЕРГИ	36
	<i>Рисунок 17. Черга для запису.....</i>	<i>36</i>
4.7	СТВОРЕННЯ І ПУБЛІКАЦІЯ ОПИТУВАНЬ.....	37
	<i>Рисунок 18. Кнопка для створення опитувань</i>	<i>37</i>
	<i>Рисунок 19. Створення опитувань.....</i>	<i>38</i>
	<i>Рисунок 20. Вигляд опитування.....</i>	<i>38</i>
4.8	ЗВІТНІСТЬ ПРО ВІДВІДУВАННЯ	38
РОЗДІЛ 6. БЕЗПЕКА		40
5.1	ЗБЕРЕЖЕННЯ ДАНИХ	40
5.2	ЗАХИСТ ВІД SQL-ІН'ЄКЦІЙ	40
РОЗДІЛ 7. ПОРІВНЯННЯ СЕРВІСІВ ДЛЯ ГОСТИНГУ		42
6.1	FLY.IO.....	42
6.2	RENDER	43
6.3	RAILWAY.....	43
ВИСНОВКИ.....		45
ДЖЕРЕЛА.....		46

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем і технологій

ЗАТВЕРДЖУЮ

ас. _____ О. Р. Смиш

(підпис)

20 липня 2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту 3 року БП ІІЗ факультету інформатики

Буденковій Маргариті Іллівна

Розробити вебзастосунок для полегшення комунікації між викладачами та студентами

Зміст ТЧ до курсової роботи:

Зміст

Анотація

Вступ

1. Дослідження
2. Огляд рішень, що існують
3. Огляд використаних технологій та інструментів
4. Структура бази даних
5. Технічні можливості застосунку
6. Безпека
7. Порівняння сервісів для гостингу

Висновки

Список літератури

Дата видачі 11 липня 2022 р.

Керівник _____ (підпис)

Завдання отримав _____ (підпис)

Календарний план виконання роботи

№	Етап роботи	Дата
1	Початок роботи, обговорення теми, мети і плану роботи, запуск опитування	08.07.2022–16.07.2022
2	Обговорення перших результатів опитування, визначення актуальності роботи	16.07.2022–30.07.2022
3	Створення «пілотного» дизайну, початок роботи з базою даних для застосунку	17.08.2022–17.09.2022
4	Виступ на конференції у Лейпцизькому університеті	8.09.2022
5	Створення кодової частини	5.02.2023–5.05.2023
6	Створення текстової частини	5.4.2023–5.05.2023

Студент _____

Керівник _____

“ _____ ” _____ 2022 р.

Анотація

Вебзастосунок спрямований на підвищення якості спілкування між викладачами та студентами в онлайн-навчальних умовах. Програму розроблено з використанням React, Node.js та інших додаткових бібліотек. Застосунок має інтерфейс чату, який уможливує анонімне відправлення повідомлень. Він надає кілька функцій для покращення онлайн-взаємодії: створення кімнат для окремих груп людей, створення опитувань, реакції на повідомлення та чергу для запитань чи захисту робіт. Завдяки цим функціям вебзастосунок дає змогу викладачам і студентам ефективніше й результативніше проводити комунікацію.

Вступ

Комунікація між викладачами та студентами – запорука якісно засвоєних знань.

Однак, рідко лекції проходять у форматі питань-відповідей, що може створювати проблеми в розумінні й використанні викладеного матеріалу. Існує багато людей, яким важко вести комунікацію зі старшими та/чи малознайомими людьми. У результаті проведеного у ході роботи дослідження можна стверджувати, що багато людей також бояться озвучувати свої питання так, щоб їх чула вся аудиторія, бо переймаються, що їхнє питання може здатись недоречним.

Таким чином студент втрачає можливість зрозуміти ключові поняття, а викладач так само не розуміє, на яких моментах слід зупинитись і зосередити більше уваги.

Розв'язанням даної проблеми є створення вебзастосунку, який надає можливість людям ставити питання анонімно, а також пропонує інші функції для полегшення навчального процесу: організація робочого простору шляхом створення кімнат, опитувань, запису на захист робіт чи створення черг для того, аби поставити запитання.

Вебзастосунок для полегшення комунікації між викладачами та студентами створено за допомогою відкритої JavaScript бібліотеки React, Node.js та інших допоміжних технологій.

Метою створення сервісу з можливістю анонімної комунікації між викладачами та студентами було полегшення процесу навчання й покращення якості розуміння викладеного матеріалу. Застосунок працюватиме як допоміжний сервіс до звичних навчальних платформ, таких як, Microsoft Teams [1]. Оскільки сервіс пропонує можливість анонімної комунікації, студенти зможуть вільно ставити свої запитання. Це зробить процес навчання більш доступним та простим.

Завданням є розробка застосунку, який пропонує всі потрібні функції в одному місці й слугує єдиною допоміжною платформою для організації навчального онлайн-простору.

Об'єктом дослідження є застосунки, які наразі використовуються під час онлайн-навчання.

Предмет дослідження – визначити й розробити застосунок, який уможливіватиме функціонал, що зазначено в меті розробки.

Наукова новизна цього дослідження полягає у розробці вебзастосунку, який пропонує анонімну комунікацію між викладачами та студентами та інший функціонал для полегшення навчального процесу. Використання сучасних вебтехнологій і анонімності дозволяє створити комфортне та відкрите середовище для ефективної взаємодії, що сприяє кращому засвоєнню знань та покращенню якості навчання.

Розділ 1. Дослідження

Для демонстрації актуальності проблеми проведено дослідження. Залучено понад 300 осіб з українських університетів (Національний університет «Києво-Могилянська академія», Київський національний університет імені Тараса Шевченка, Національний університет біоресурсів і природокористування України, Київський університет імені Бориса Грінченка) усіх років навчання, а також серед викладачів.

1.1 Результати

З опитування бачимо такі результати:

1. Чи бувало таке, що Ви НЕ ставили питання на парі, бо воно здавалось вам дурним? (Див. Рис. 1)

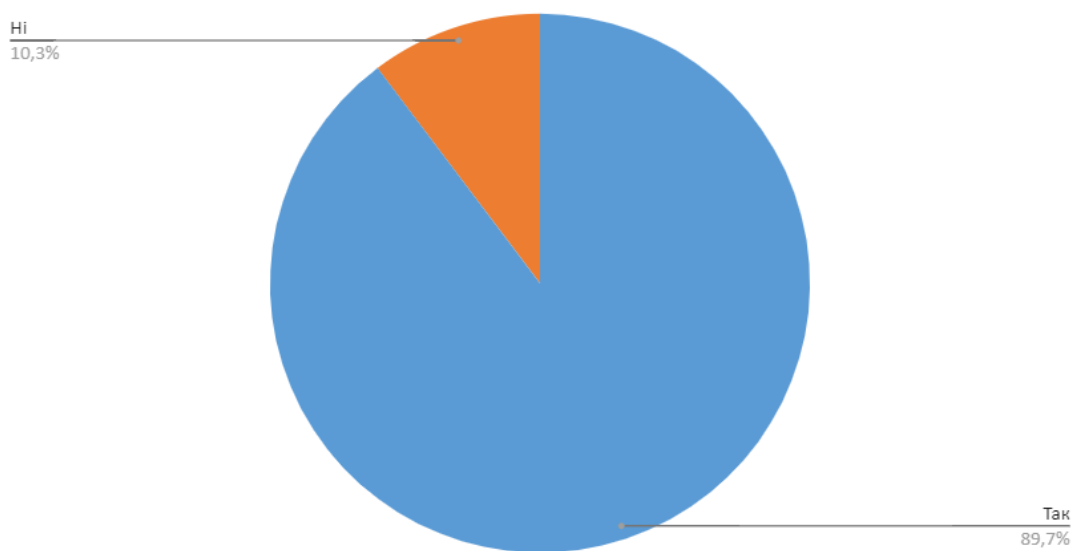


Рисунок 1. Результати опитування 1.

2. Чи звертались Ви потім до викладачів додатково з цим питанням? (Див. Рис. 2)

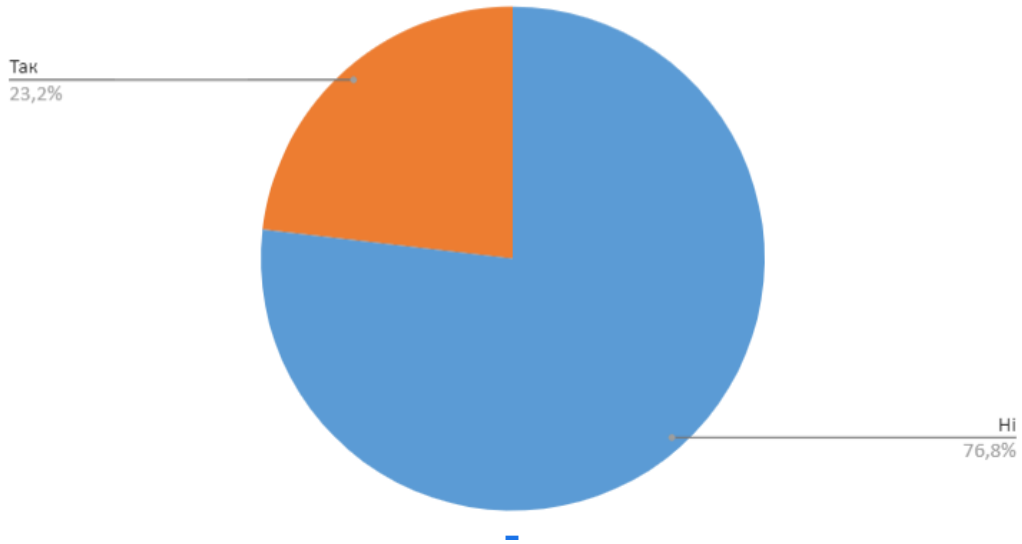


Рисунок 2. Результати опитування 2.

3. Чи бувало таке, що після питання викладача "Чи все було зрозуміло?" Вам було щось незрозуміло, але ви промовчали? (Див. Рис. 3)

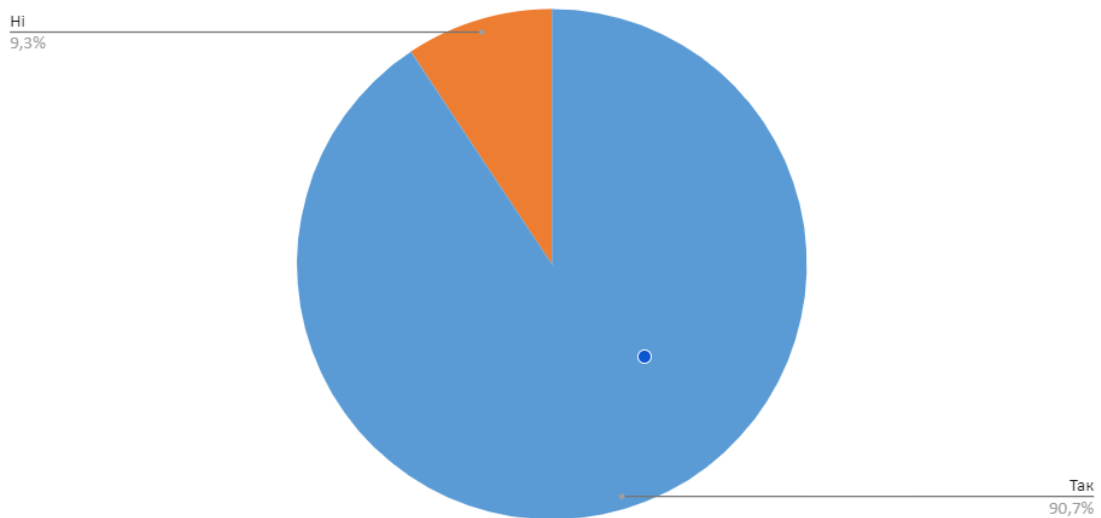


Рисунок 3. Результати опитування 3.

4. Як зазвичай Ви ведете комунікацію з викладачем під час пари?

(Див. Рис. 4)

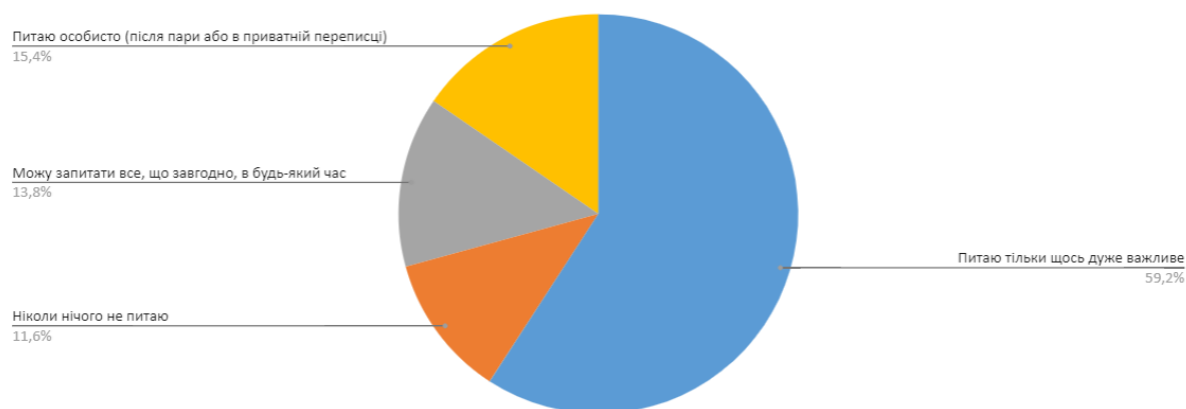


Рисунок 4. Результати опитування 4.

5. Чи бувало таке, що після повторного пояснення матеріалу, Вам все одно було щось незрозуміло, але Ви більше це не перепитували, бо соромилися? (Див. Рис. 5)

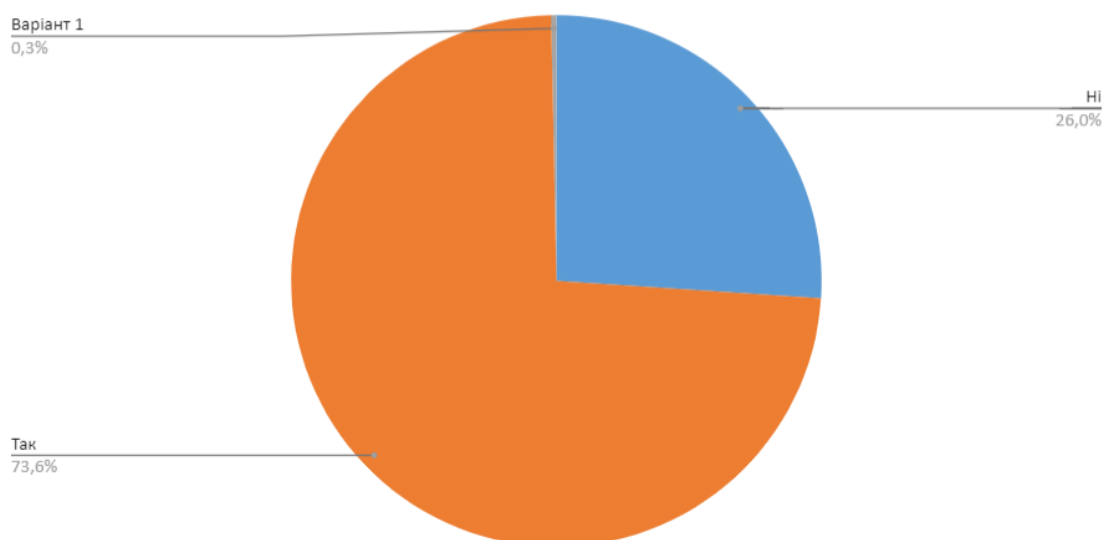


Рисунок 5. Результати опитування 5.

6. Чи покращилось би ваше розуміння матеріалу, якби в свій час ви могли поставити всі ті питання, які вас турбували? (Див. Рис. 6)

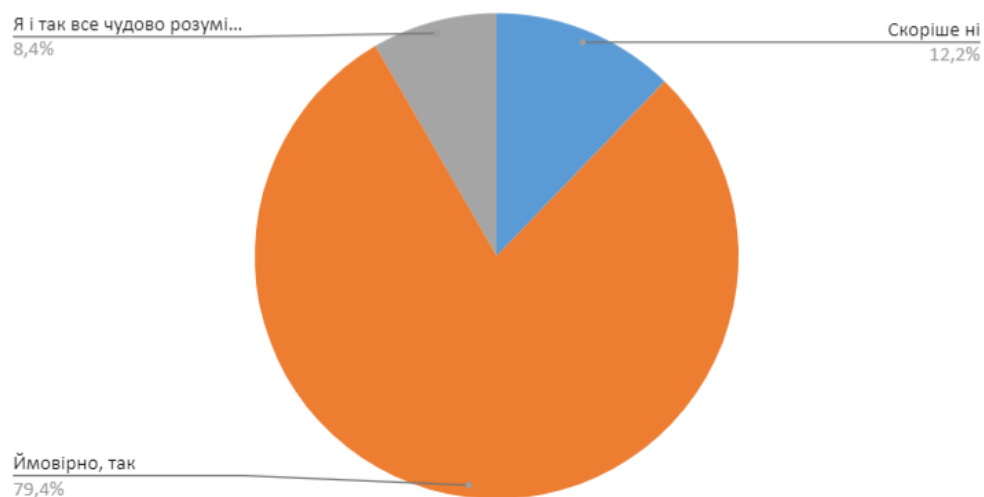


Рисунок 6. Результати опитування 6.

1.2 Дослідження кореляції

Використовуючи результати опитування, видно кореляцію (див. Рис.7):



Рисунок 7. Дослідження кореляції 1.

За результатами виявлено, що $\approx 80\%$ студентів стикались з проблемою страху поставити питання на парі серед інших студентів, через те, що воно здавалось дурним. Натомість лише 23% опитуваних повертались до цього питання після пари, сам на сам з викладачем. Це свідчить про те, що студенти відчують більшу впевненість у приватному спілкуванні з викладачем, ніж на публіці. Однак, варто зауважити, що уточнювальні питання можуть бути корисними для всієї аудиторії.

Лише 20,5% студентів (див. Рис. 8) звертались до викладачів після пари, якщо під час викладення матеріалу зіштовхувались з труднощами. Натомість 69% студентів цього не робили. Багато студентів не мають впевненості сказати про те, що пояснений матеріал досі незрозумілий або, що відповідь на запитання не до кінця прояснила деякі аспекти матеріалу.



Рисунок 8. Дослідження кореляції 2.

Розділ 2. Огляд рішень, що існують

2.1 Google Forms

Google Forms є інструментом для створення опитувань та тестів. Хоча його можна використовувати для комунікації між вчителями та учнями, це не найкращий інструмент для цього з таких причин:

- Google Forms не призначений для комунікації у реальному часі, тобто студенти не отримують негайної відповіді на свої питання. Однак це може стати ключовим моментом для того, аби зрозуміти матеріал;

- Google Forms має обмежену функціональність для комунікації. Наприклад, платформа не пропонує групового розсилання повідомлень, а також не надає можливості створення окремого навчального простору;

- хоча Google Forms уможливорює збирати анонімні відповіді, він не забезпечує шифрування «end-to-end» [2], тобто повідомлення можуть бути перехоплені або прочитані зловмисниками;

- Google Forms в основному призначений для одноразових опитувань або тестів. Видається незручним заповнення нової форми кожного разу, коли учні хочуть зв'язатися зі своїм викладачем.

У цілому, Google Forms є корисним для збору відгуків або для проміжних тестувань, однак він не закриває потребу швидкої та анонімної комунікації між викладачами й студентами.

2.2 Microsoft Teams

Microsoft Teams — інструмент, який використовується як основний для комунікації під час освітнього процесу. Однак ця платформа видається не оптимальним варіантом для анонімної комунікації між вчителями та учнями з таких причин:

- Microsoft Teams перевантажений певними налаштуваннями, що може спричинити дискомфорт у деяких викладачів та студентів. Застосунок пропонує широкий спектр додаткових функцій, які не сприяють до комунікації між викладачами та студентами;

- хоча Microsoft Teams уможливорює створювати анонімні облікові записи, програма не пропонує анонімного розсилання повідомлень у групах або каналах, тобто учні не мають можливості відправляти повідомлення своєму викладачеві анонімно;

- Microsoft Teams є хмарною платформою, тому повідомлення та дані зберігаються на серверах Microsoft [3]. Хоча Microsoft пропонує потужні заходи безпеки для захиту даних користувачів, завжди існує безпековий ризик;

- Microsoft Teams потребує стабільного інтернет-з'єднання та сучасної техніки для якісної роботи [4]. Це може бути перешкодою для деяких студентів, які не мають доступу до надійного Інтернету або сучасних пристроїв.

У цілому, Microsoft Teams є насиченим інструментом для комунікації та співпраці, однак перевантаженість програми створює додаткові проблеми.

2.3 Google Meet

Google Meet – це платформа для відеоконференцій, що дає змогу проводити віртуальні зустрічі та уроки. Хоча вона пропонує опцію анонімних повідомлень, це не оптимальний варіант для постійної комунікації між студентами та викладачами з таких причин:

- Google Meet не формує централізовану платформу для організації онлайн-простору та керування комунікацією. Ця платформа призначена переважно для відеоконференцій, тобто вона не найзручніший інструмент для асинхронної комунікації;

- оскільки Google Meet в основному призначений для відеоконференцій, то не надає додаткових інтерактивних функцій. Це ускладнює взаємодію між користувачами;

- для використання Google Meet потрібно мати або створити акаунт у Google. Даний підхід може створити проблеми для тих, хто не користується цим сервісом на постійній основі.

Загалом, Google Meet надає можливості анонімного спілкування, однак ця платформа є неоптимальним варіантом для постійного зв'язку та співпраці, тому що не забезпечує окремий навчальний онлайн-простір.

2.4 Remind

Remind є застосунком для обміну повідомленнями, призначеним для комунікації між викладачами та студентами. Але він має такі суттєві недоліки:

- Remind призначений для одностороннього спілкування між викладачами та студентами, тобто здебільшого працює як інформаційна платформа;

- застосунок надає небагато інтерактивних функцій. Це ускладнює взаємодію зі студентами та ефективну співпрацю;

У цілому, Remind є корисним інструментом для онлайн-навчання, але не найкращим варіантом для постійної комунікації. Незважаючи на те, що Remind розроблений як застосунок для керування навчальним процесом і його організації, він надає набагато більше можливостей викладачам, аніж студентам. Для останніх застосунок є лише новинним.

2.5 Telegram

Telegram є відомою платформою для обміну повідомленнями, файлами та аудіоматеріалами, що часто використано в навчальному процесі. [5] Він уможлиблює створення опитувань, чатів та реакцій.

Також Telegram відомий своїми потужними заходами безпеки та шифруванням даних. Це дає змогу користувачам безпечно вести комунікацію.

Однак, Telegram не дає можливості надсилати повідомлення анонімно – лише створювати анонімні профілі користувачів. Це ускладнює роботу викладачеві, оскільки той не може відслідкувати присутність на заняттях.

2.6 Доцільність розробки

Створення застосунку, який полегшує комунікацію між викладачами та студентами, є важливим у наш час. Останніми роками освітній процес сильно змінився: став більш технологічним [6]; це змушує викладачів шукати різні способи для залучення студентів і комунікації з ними, для оптимізації навчання.

Ефективна комунікація є надзвичайно важливою для створення продуктивного навчального середовища. Викладачам потрібно мати можливість виходити на зв'язок зі своїми студентам регулярно для надання відгуків на завдання, порад та підтримки або відповідей на запитання. Студенти, з іншого боку, також повинні мати можливість спілкуватися зі своїми викладачами: просити допомоги, дізнаватись оцінки, ділитись відгуками.

Застосунок, який дає змогу надсилати повідомлення анонімно, може допомогти зняти бар'єри у комунікації. Наприклад, стане легше для інтровертованих учнів звертатися по допомогу, якщо вони можуть зробити це анонімно.

Крім того, платформа допоможе оптимізувати наступні процеси: проведення опитувань або швидких тестувань та створення черги для відповідей та/або захисту робіт.

Розділ 3. Огляд використаних технологій та інструментів

Застосунок вирішено розробляти, використовуючи вебтехнології, тому що програми, встановлені на комп'ютер чи іншу техніку мають такі недоліки:

- такі програми потребують встановлення, в той час, як вебзастосунки доступні через мережу Інтернет та можуть бути використані з будь-якого пристрою;
- такі програми зазвичай потребують оновлення на кожному пристрої окремо, коли вебзастосунки можуть бути оновлені централізовано;
- Такі програми менш безпечні, оскільки можуть бути атакованими вірусами та зловмисним програмними засобами, в той час, як вебзастосунки є більш безпечними, оскільки не потребують встановлення на локальний пристрій.

2.1 JavaScript

JavaScript — динамічна, об'єктноорієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Вона найбільш відома як мова для вебсторінок, але багато технологій та платформ також її використовують, наприклад Node.js. [7]

Для розробки цього вебзастосунку JavaScript обрано, тому що:

- її можна виконувати у веббраузері, що дає можливість створювати динамічні інтерактивні вебсторінки, які реагують на дії користувача в режимі реального часу та не вимагають оновлення сторінки;
- доступні численні бібліотеки та фреймворки JavaScript, наприклад, React, Angular і Vue.js, які спрощують процес розробки та дають можливість швидше створювати складні програми;
- JavaScript легко інтегрується з HTML і CSS, в наслідок чого можна створювати інтерактивні інтерфейси користувача.

2.1 Node.js

Node.js – це відкрита серверна платформа, яка виконує JavaScript-код на боці сервера. Вона базується на JavaScript-рушії V8 [8], що розроблено компанією Google. Node.js уможливорює створення високопродуктивних серверних додатків, які можуть обробляти великі обсяги даних та запитів [9].

Цю технологію використано з таких причин:

- Node.js забезпечує високу продуктивність завдяки асинхронному програмуванню та викликам I/O, що не блокуються [10]. Це дає змогу обробляти великі обсяги даних та запитів за короткий час;
- Node.js уможливорює легко масштабувати серверні додатки за допомогою кластеризації та розподіленням на мікросервіси. Це дає можливість збільшити обсяги даних, які може обробляти додаток та збільшити швидкість відповіді на запити;
- Node.js має велику кількість модулів, які розширюють функціональність серверного додатка. Ця особливість дає можливість використовувати готові рішення для різних задач, що зменшує час розробки та спрощує процес програмування;
- Node.js запускає серверні додатки без необхідності встановлювати та налаштовувати спеціальні сервіси. Це забезпечує швидкий та зручний процес розгортання застосунку.

2.3 Axios

Axios — це заснований на Promises [11] HTTP-Client для Node.js і браузера. На боці сервера він використовує власний HTTP-модуль Node.js, тоді як на боці клієнта, тобто в браузері, він використовує XMLHttpRequests. [12]

Використання Axios корисне у застосунку, тому що:

- підтримується в усіх браузерах;
- використовує Promises, що є сучасним підходом до обробки асинхронних функцій у JavaScript;
- підтримує функцію скасування незавершених HTTP-запитів, що зменшує навантаження на сервер;

2.2 React.js

React.js – це бібліотека JavaScript, призначена для розробки інтерфейсу користувача (UI) у вебзастосунках. React.js дає змогу створювати компоненти, які можуть бути повторно використані та містити логіку відображення даних на боці клієнта. [13]

React.js базується на концепції компонентів [14]. Кожен компонент містить HTML-код, JavaScript-логіку та CSS-стили. Компоненти можна вкладати один в одного, щоб легше створювати інтерфейси.

Основні переваги використання React.js для вебзастосунків:

- React.js використовує віртуальний DOM (Document Object Model), що дає змогу зменшити кількість звернень до реального DOM, тобто збільшує продуктивність вебзастосунку;
- React.js дає змогу легко розширювати функціональність вебзастосунку та збільшувати обсяги даних, що обробляються та/чи відображаються на екрані;
- використання JSX-синтаксису [15] зробить код більш читабельним та легким для розуміння.

2.3 React Router

React Router – це бібліотека JavaScript для React.js, яка реалізує маршрутизацію вебзастосунку на боці клієнта. Вона дає можливість

користувачам переміщуватися між різними сторінками застосунку без необхідності перезавантаження сторінки. [16]

Основні функції React Router:

- React Router надає можливість вирізняти URL-адреси, що збігаються з різними сторінками вебзастосунку. Кожен шлях URL-адреси пов'язаний з компонентом React.js, який відповідає за відображення вмісту на певній сторінці;

- React Router дає можливість переміщатися між різними сторінками застосунку за допомогою навігаційних покликань, які можуть вбудовуватись в компоненти React.js;

- React Router уможливорює налаштування анімації переходів між сторінками вебзастосунку, що робить платформу більш привабливою для користувача.

- React Router керує історією браузера під час переміщення між сторінками, що дає користувачу можливість повертатися до попередніх розділів сайту.

2.4 React-Bootstrap

React-Bootstrap – це бібліотека компонентів React.js, яка базується на CSS-фреймворку Bootstrap. Вона містить набір готових компонентів, які використані для розробки інтерфейсу вебзастосунку [17].

React Bootstrap використано з таких причин:

- React-Bootstrap містить велику кількість готових компонентів: кнопки, форми, таблиці, меню, модальні вікна. Ці компоненти використовуються в різних частинах вебзастосунку, що економить час та спрощує процес розробки;

- React-Bootstrap встановлює параметри готових компонентів: розмір, колір, шрифт та інші параметри – це дає змогу швидко експериментувати з кольорами й формами, підлаштовувати компоненти під потреби;

- React-Bootstrap сумісний із великою кількістю браузерів і пристроїв, тобто зовнішній вигляд застосунку підтримуватиметься і правильно відображатиметься.

2.5 MySQL

MySQL – це відкрита система управління базами даних (СУБД) на основі мови SQL (Structured Query Language) [18]. MySQL є однією з найпопулярніших СУБД, яка використовується в багатьох застосунках. Вона зберігає та керує великими обсягами даних [19].

Основні функції MySQL:

- MySQL надає можливість зберігати великі обсяги даних у структурованому форматі та забезпечує швидкий доступ до них. У цій системі можна виконувати різні операції з даними: додавання, видалення, оновлення та пошук;

- MySQL легко масштабує базу даних за допомогою розподіленої архітектури та кластеризації серверів, що дає змогу збільшити обсяги даних та кількість запитів, які можна обробляти;

- MySQL має механізми захисту даних: автентифікація користувачів, шифрування даних та контроль доступу, що забезпечує безпеку даних та запобігає несанкціонованому доступу.

2.6 Sass

Sass (Syntactically Awesome Style Sheets) – це препроцесор CSS, що надає розробникам спроможність створювати організований та зрозумілий CSS-код. Sass розширює стандартні CSS можливості за допомогою змінних, вкладених селекторів, міксинів [20]. Sass може бути зкомпільованим до стандартного CSS, що уможлиблює його використання на вебсайтах. [21]

Цю технологію використано з таких причин:

- Sass використовує змінні для зберігання значень кольорів, шрифтів, розмірів, що дає змогу легко змінювати значення на всьому вебсайті в одному місці;
- Sass уможливорює вкладання селекторів, що забезпечує більш чистий та зрозумілий CSS-код, через що легко зрозуміти взаємозв'язок між елементами на сторінці;
- Sass використовує міксини для збереження блоків CSS-коду, що використано на різних частинах вебсайту. Ця функція зменшує дублювання коду та спрощує процес розробки;
- Sass використовує функції для обчислення значень, тобто зменшує кількість коду.

2.6 Material UI

Material UI – це бібліотека компонентів React з відкритим кодом, що реалізує Google Material Design. Вона містить повну колекцію компонентів, які готові до використання при розробці [22].

Переваги використання цієї технології такі:

- Material UI містить набір готових компонентів, які можна використовувати для створення вебсторінок, що дає розробникам можливість прискорити процес розробки та зменшити кількість коду;
- Material UI розроблено спеціально для React, тому він інтегрується до React-застосунків;

- Material UI пропонує готові дизайни, які відповідають концепції Material Design від Google, що дає змогу розробникам створювати вебсторінки з сучасним дизайном без використання додаткових зовнішніх бібліотек;

- Material UI уможлиблює налаштування кожного компоненту відокремлено, тобто досягати відповідності дизайну до вимог застосунку та потреб користувачів.

Розділ 4. Структура бази даних

3.1. EER-модель

Перед розробкою застосунку створено EER-модель. Це зроблено для того, аби краще розуміти, як робити типізацію полів та валідацію, якісніше формувати SQL-запити, усунути надлишковість та неузгодженість даних.

ER-модель програми має такий вигляд (див. Рис. 9):

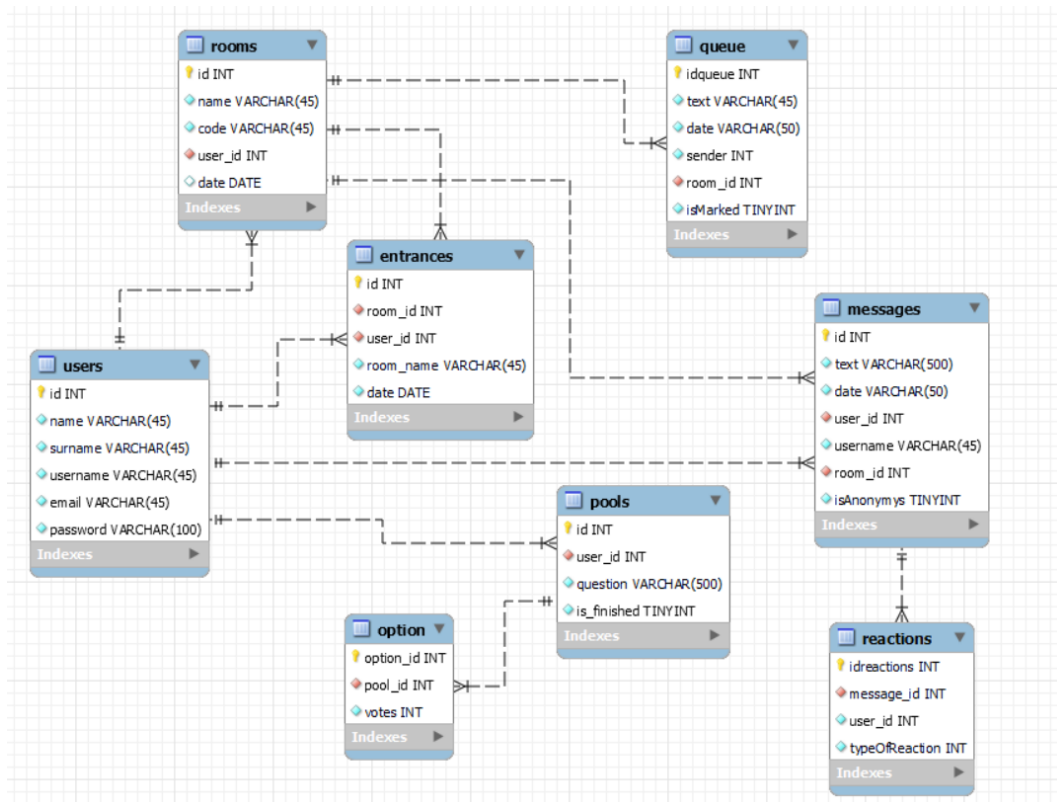


Рисунок 9. EER-модель застосунку.

3.2 Реляційна модель

Для спрощення роботи з базою даних проведено нормалізацію.

USER, відповідає типу сутності «users».

Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK	id	INT	NOT NULL	
	name	Varchar (45)	NOT NULL	
	surname	Varchar (45)	NOT NULL	
	username	Varchar (45)	NOT NULL	
	email	Varchar (45)	NOT NULL	
	password	Varchar (100)	NOT NULL	

Таблиця 1. Сутність «USER»

ROOM, відповідає типу сутності «rooms».

Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK	id	INT	NOT NULL	
	name	Varchar (45)	NOT NULL	
	code	Varchar (45)	NOT NULL	
FK	user_id	INT	NOT NULL	On delete: no action; on update: cascade
	date	Date	NOT NULL	

Таблиця 2. Сутність «ROOM»

ENTRANCE, відповідає типу сутності «entrances».

Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK	id	INT	NOT NULL	
FK	room_id	INT	NOT NULL	On delete: no action; on update: cascade
FK	user_id	INT	NOT NULL	On delete: no action; on update: cascade
	date	Date	NOT NULL	

Таблиця 3. Сутність «ENTRANCE»

QUEUE, відповідає типу сутності «queue».

Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK	idqueue	INT	NOT NULL	
	text	Varchar (50)	NOT NULL	
	date	Date	NOT NULL	
FK	sender	INT	NOT NULL	On delete: no action; on update: cascade
FK	room_id	INT	NOT NULL	On delete: no action; on update: cascade
	isMarked	Boolean	NOT NULL	

Таблиця 4. Сутність «QUEUE»

MESSAGE, відповідає типу сутності «messages».

Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK	id	INT	NOT NULL	
	text	Varchar (500)	NOT NULL	
	date	Date	NOT NULL	
FK	user_id	INT	NOT NULL	On delete: no action; on update: cascade
FK	room_id	INT	NOT NULL	On delete: no action; on update: cascade
	isAnonymous	Boolean	NOT NULL	

Таблиця 5. Сутність «MESSAGE»

REACTION, відповідає типу сутності «reactions».

Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK	idreaction	INT	NOT NULL	
FK	message_id	INT	NOT NULL	On delete: no action; on update: cascade
FK	user_id	INT	NOT NULL	On delete: no action; on update: cascade
	typeOfReaction	INT	NOT NULL	

Таблиця 6. Сутність «REACTION»

POOL, відповідає типу сутності «pools».

Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK	id	INT	NOT NULL	
FK	user_id	INT	NOT NULL	On delete: no action; on update: cascade
	question	Varchar (500)	NOT NULL	
	is_finished	Boolean	NOT NULL	

Таблиця 7. Сутність «POOL»

OPTION, відповідає типу сутності «option».

Ключ	Ім'я атрибуту	Тип атрибуту	NULL/NOT NULL	FK: ON DELETE, ON UPDATE
PK	option_id	INT	NOT NULL	
FK	pool_id	INT	NOT NULL	On delete: no action; on update: cascade
	votes	INT	NOT NULL	

Таблиця 8. Сутність «OPTION»

Розділ 5. Технічні можливості застосунку

4.1 Логін і реєстрація

Застосунок має сторінки логіну (див. Рис. 9) та реєстрації (див. Рис. 10), де користувач має підтвердити свою особистість для входу у систему або створити профіль.

Це необхідно за таких причин:

- кожен користувач має свій власний простір, де зберігаються й обробляються всі його дані;
- лише зареєстровані користувачі мають доступ до системи, що покращує безпеку застосунку та запобігає несанкціонованому доступу до конфіденційних даних та захисту приватності користувачів;
- шляхом перевірки ідентифікації кожного користувача система може відстежувати та контролювати дії користувачів, що допоможе розв'язувати проблеми з порушенням безпеки, якщо такі виникатимуть.

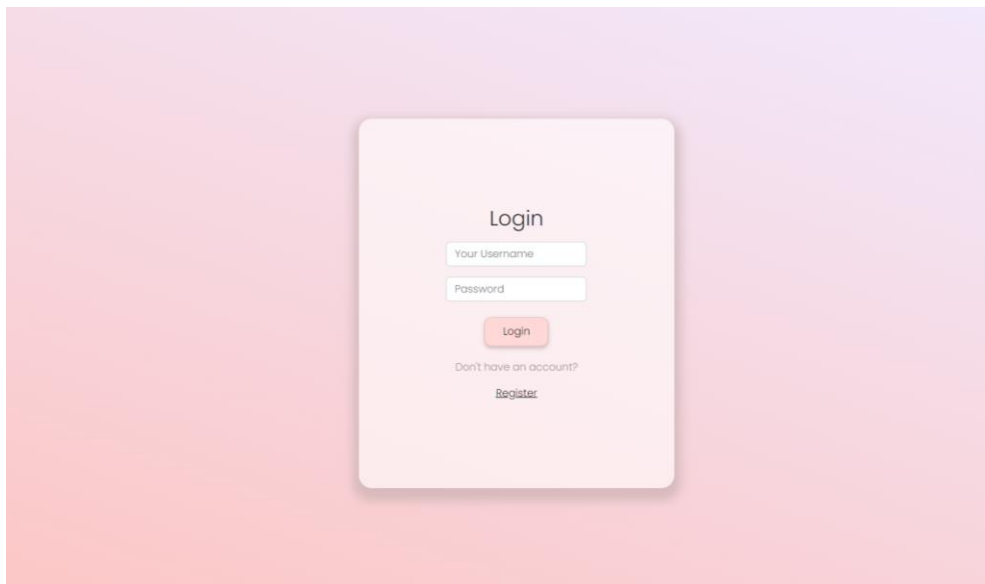


Рисунок 9. Сторінка логіну.

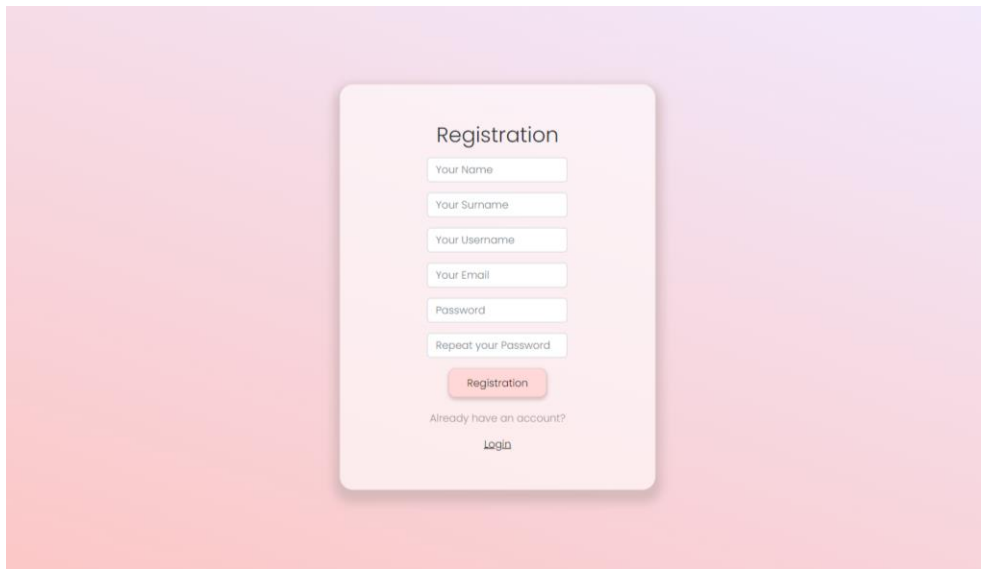


Рисунок 10. Сторінка реєстрації.

4.2 Створення кімнат

Подібно до Microsoft Teams, однією з технічних завдань програми є створення окремих кімнат для кожного предмету або окремих груп викладачів чи/та студентів (див. Рис. 11). Це уможлиблює розділяти простір та проводити різні обговорення.

Користувачі мають змогу легко перемикатися між різними кімнатами, щоб отримати доступ до обговорень, ресурсів та іншої відповідної інформації.

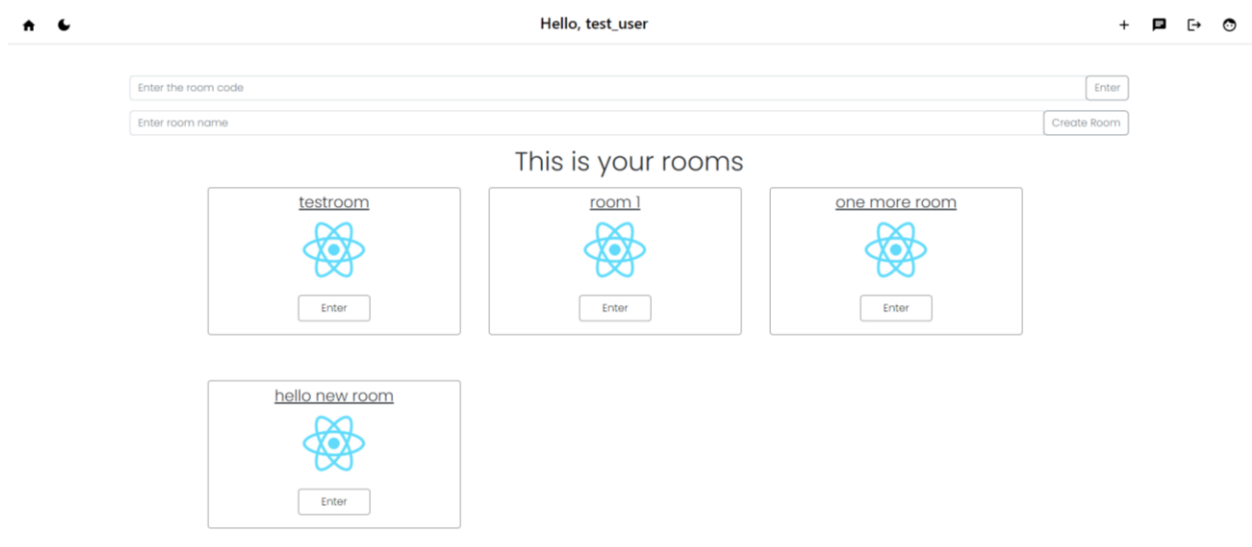


Рисунок 11. Головна сторінка профілю користувача

Перший елемент вводу вможливилює вхід у кімнату, яка вже існує. Зробити це можна за допомогою коду, доступ до якого може отримати людина, що створила дану кімнату.

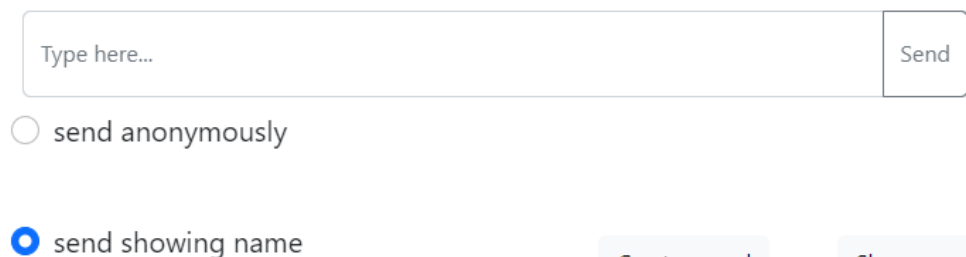
Другий елемент вводу дає можливість створити кімнату самостійно, вказавши лише назву. Код генерується автоматично (див. Додаток 1).

```
function generateCode(length) {  
  let result = '';  
  const characters =  
  'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';  
  const charactersLength = characters.length;  
  let counter = 0;  
  while (counter < length) {  
    result += characters.charAt(Math.floor(Math.random() * charactersLength));  
    counter += 1;  
  }  
  return result;  
}
```

Додаток 1. Створення коду для кімнати

4.3 Анонімні повідомлення

У застосунку функціональність анонімного чату може застосовуватись до окремих повідомлень шляхом вказання бажаного надсилання: з іменем користувача чи без (див. Рис. 12, рис. 13).



The image shows a user interface for sending a message. At the top, there is a text input field with the placeholder text "Type here..." and a "Send" button to its right. Below the input field, there are two radio buttons. The first radio button is unselected and is labeled "send anonymously". The second radio button is selected (indicated by a blue dot) and is labeled "send showing name". To the right of the second radio button, there are two faint, partially visible labels: "show name" and "show name".

Рисунок 12. Надсилання анонімних повідомлень

За замовчуванням повідомлення надсилається із зазначенням імені відправника.

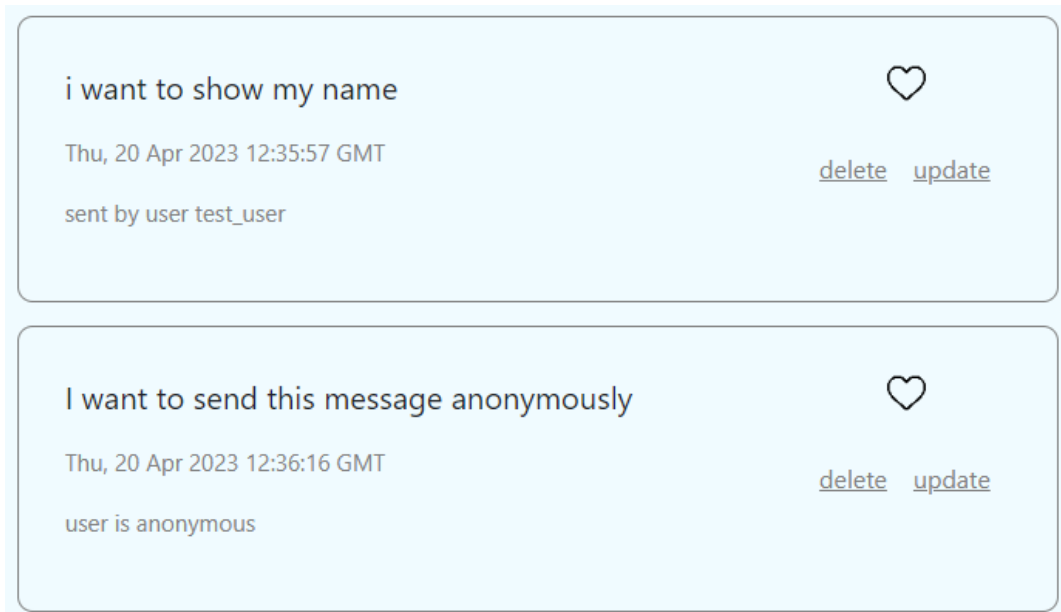


Рисунок 13. Приклад відправки анонімного повідомлення

4.4 Видалення і редагування повідомлень

Структура повідомлення виглядає наступним чином (див. Рис. 14):

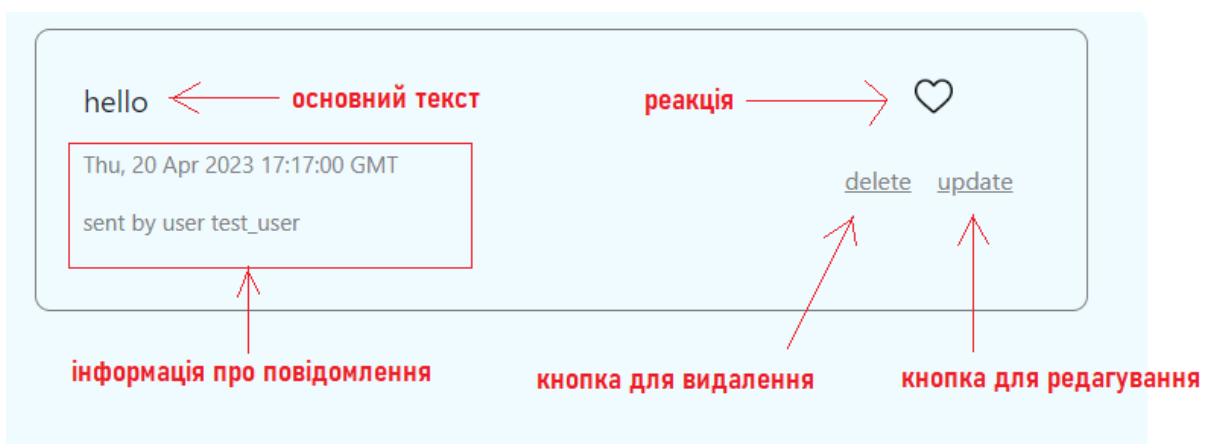


Рисунок 14. Структура повідомлення

Кнопка «Delete» надає можливість видалити повідомлення, якщо воно було надіслане користувачем, що наразі авторизований у системі.

Кнопка «Update» уможлиблює редагування повідомлення, якщо воно було надіслане користувачем, що наразі авторизований у системі. (див. Рис. 15).

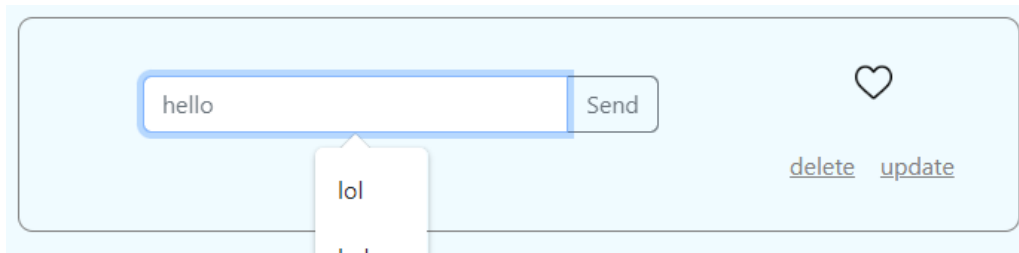


Рисунок 15. Вікно редагування повідомлення

4.5 Реакція на повідомлення

Реакція на повідомлення – це функція, що дає користувачам змогу виражати свої емоції або думки про повідомлення без прямої відповіді (див. Рис. 16).

Переваги такого підходу:

- спосіб показати свою реакцію на повідомлення без прямої відповіді, що є корисним в групових чатах, де користувачі не мають бажання відповідати на кожне повідомлення, але все ж хочуть показати свою підтримку або згоду;
- користувачі можуть швидко й легко реагувати на повідомлення одним клацом, не повинні набирати відповідь – це економить час;
- реакція на повідомлення може збільшити залученість у застосунку для комунікації, заохочуючи користувачів активніше брати участь у розмовах. Користувачі більш схильні взаємодіяти з повідомленнями, коли у них є можливість реагувати на них.

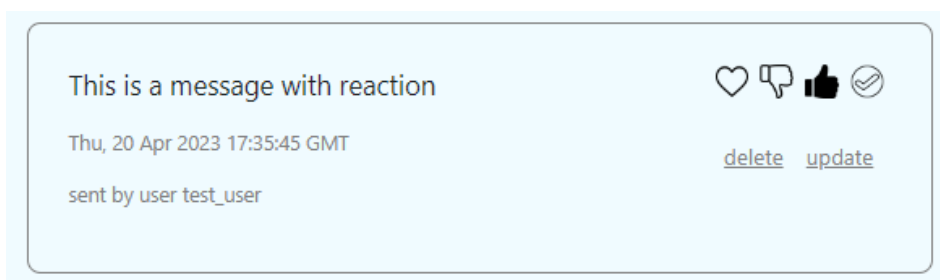


Рисунок 16. Повідомлення з реакцією

4.6 Створення черги

Черга – це функція, яка дає користувачам можливість створювати чергу для питань, відповідей та/або захисту робіт (див. Рис. 17). Наразі для закриття такої потреби часто використовується Excel [23]. Це також створює перезавантаженість застосунками, які завжди мають бути під рукою.

Переваги такої функції:

- кожен користувач отримує свій час, щоб поставити запитання або захистити роботу. Це допомагає уникнути плутанини та конфліктів;
- модератор має можливість пріоритезувати та адресувати запитання.



The image shows a vertical queue interface. At the top, there are three input fields, each containing a message and a checkbox to its right. The messages are "hello world", "i dnt know anything", and "margo". Below these is a large, empty light blue rectangular area. At the bottom, there is a text input field with the placeholder "Your name here" and a "Send" button to its right.

Рисунок 17. Черга для запису

4.7 Створення і публікація опитувань

Опитування – ще одна технічна можливість застосунку, що допомагає проводити швидкі тестування чи асинхронно вирішувати організаційні питання (див. Рис. 18).

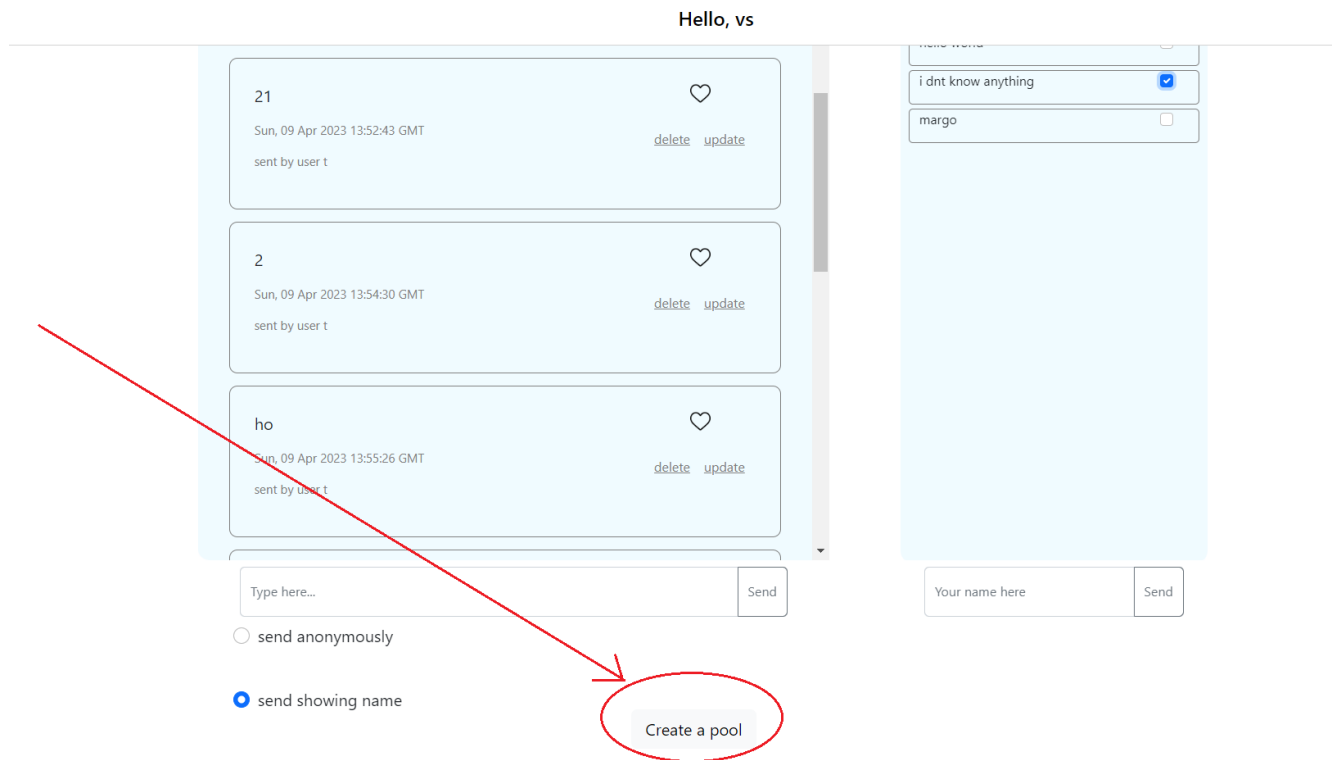


Рисунок 18. Кнопка для створення опитувань

Для створення опитувань потрібно натиснути на кнопку «Create a pool». Заповнені поля можуть мати такий вигляд (див. Рис. 19):

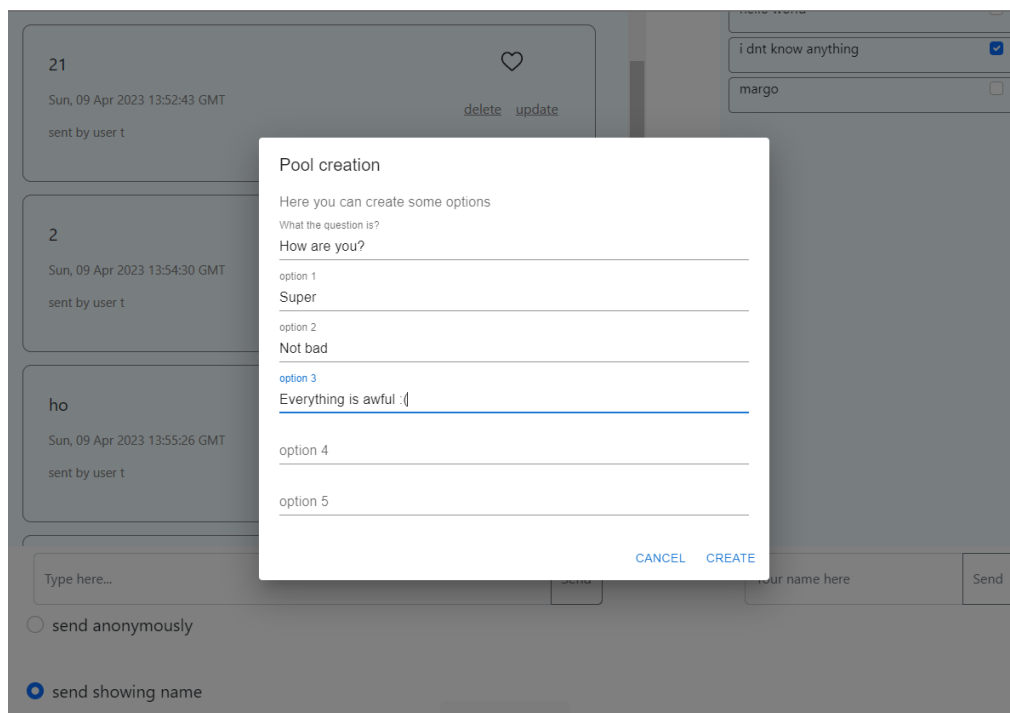


Рисунок 19. Створення опитувань

Після натискання на клавішу «Create» опитування з'являється внизу сторінки для всіх користувачів кімнатою (див. Рис. 20).

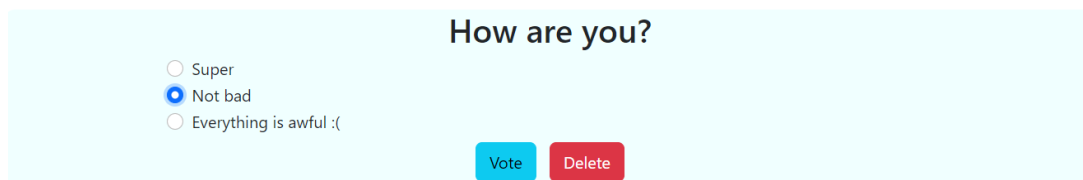


Рисунок 20. Вигляд опитування

Кнопка «Vote» дає можливість проголосувати за вибраний варіант, в той час як кнопка «Delete» з'являється тільки для того, хто створив опитування, і уможлиблює його видалення.

4.8 Звітність про відвідування

Оскільки деяким викладачам треба відстежувати відвідуваність лекційних чи семінарських занять студентами, запроваджено функцію, яка дає змогу

отримати звітність про відвідування кімнати за день. При натисканні на кнопку «Get Students» (див. Рис. 21), викладач може вивантажити PDF-файл (див. Табл. 9) з прізвищами студентів, які відвідували кімнату в цей день.

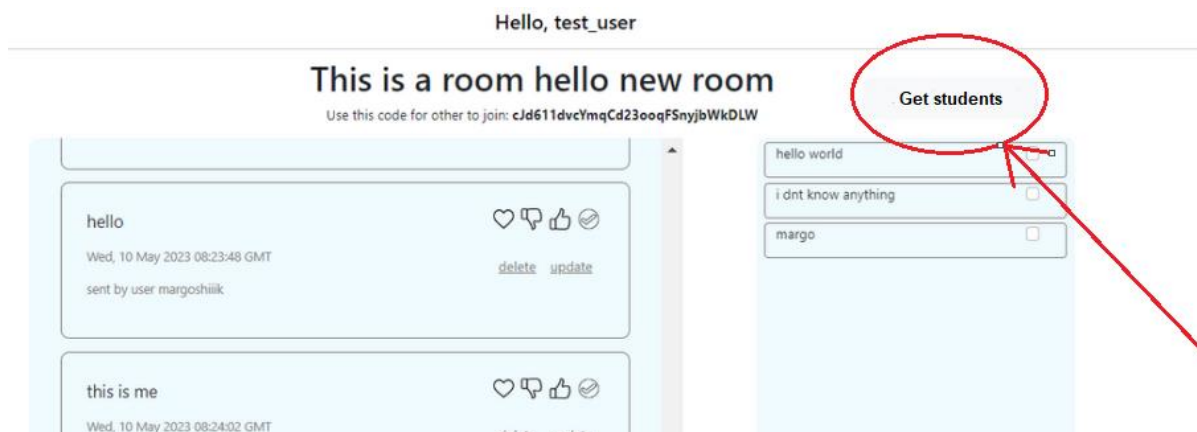


Рисунок 21. Розміщення кнопки «Get students»

Room	Name	Surname	Date
Test_room	Marharyta	Budenkova	5.05.2023
Test_room	Tanya	Zhovtukha	5.05.2023
Test_room	Sveta	Lukichova	5.05.2023

Таблиця 9. Вигляд таблиці, який зберігається у PDF-форматі.

Розділ 6. Безпека

5.1 Збереження даних

Однією з важливих компонентів безпеки є правильне збереження даних. Найбільш вразливими елементами є поля паролів користувачів, тому для них приділено особливу увагу.

Паролі користувачів зберігаються в базі даних зашифрованими за допомогою хеш-функції `bcrypt` завдяки допоміжній бібліотеці `node.bcrypt.js` [24].

`bcrypt` використовує пароль, що надано користувачем, і перетворює його на хеш. Хеш – це те, що зберігається в базі даних. Це запобігає доступу до паролів користувачів у разі порушення даних. На противагу деяким іншим алгоритмам хешування паролів, які просто хешують простий пароль, `bcrypt` використовує сіль.

Перед хешуванням простого пароля генерується сіль – унікальний згенерований рядок, який забезпечує додатковий рівень безпеки для згенерованого хешу. Потім він додається до пароля, і все хешується (простий пароль і сіль). Це допомагає під час атаки зломисників, бо, якщо пароль вгадати можливо, то сіль – ні. [25]

5.2 Захист від SQL-ін'єкцій

Наступну небезпеку можуть складати SQL-ін'єкції.

SQL-ін'єкція - це вид атак на вебзастосунки, що використовується для отримання незаконного доступу до бази даних. Уведені у вебформу дані можуть бути використані для зміни, видалення або отримання конфіденційної інформації.

Оскільки спеціальних бібліотек для захисту від SQL-ін'єкцій не існує, використано такі методи для захисту даних:

1. параметризовані запити: дані передаються як параметри запиту, що унеможлиблює введення некоректних даних, які можуть бути використані як SQL-ін'єкції;
2. валідація та фільтрація введених даних: регулярні вирази для введення некоректних даних, типізація полів, а також кількість символів на введення.

Розділ 7. Порівняння сервісів для гостингу

Гостинг – важлива складова життєвого процесу застосування. Він допомагає тестувати застосунок в умовах реального використання і знаходити помилки, які не були помічені у середовищі розробки. Це, своєю чергою, допомагає підвищити якість програми.

Більш того, гостинг дає користувачам доступ до застосування, тобто робить його відкритим для більшої аудиторії, дає можливість зібрати відгуки, що також сприяють майбутньому розвитку застосування.

Fly.io, Render та Railway – найвідоміші сервіси для гостингу.

6.1 Fly.io

Fly.io — це хмарна платформа, яка спеціалізується на запуску контейнерних програм [26]. Вона пропонує велику мережу серверів і автоматичний розподіл навантаження для забезпечення швидкої роботи. Fly.io підтримує низку варіантів розгортання, включаючи контейнери Docker і статичні сайти. Ціна залежить від масштабів використання. [27]

Fly.io підтримує багато варіантів гостингу, таких як статичні сайти або сайти з використанням Docker, тобто можуть розміщуватись сайти будь-якого навантаження і розміру. Також, в залежності від трафіку, Fly.io пропонує різні тарифи. До того ж тут є вбудовані SSL-сертифікати й запроваджена проста інтеграція з Git та GitHub.

З недоліків можна виділити лише труднощі, які можуть з'являтися під час запровадження даної технології у застосунок. Однак документація допомагає з ними розібратись.

6.2 Render

Render — це хмарна платформа, яка дає можливість розміщувати вебзастосунки, бази даних і статичні сайти. Платформа має інтуїтивно зрозумілий інтерфейс користувача. Render також пропонує автоматичні SSL-сертифікати та просту інтеграцію з GitHub і GitLab. Однак ціна є вищою порівняно з іншими платформами, особливо для великих програм. [28]

Render легший у використанні, має простий інтерфейс користувача, в наслідок чого платформа пропонує менше опцій для налаштувань. Попри це, ціна за надані послуги є вищою, ніж пропонує Fly.io.

Однак платформа підтримує гостинг більшої кількості видів застосунків.

6.3 Railway

Railway — це платформа для гостингу, на якій можна розгорнути застосунок в хмарному середовищі. Вона розроблена спеціально для розміщення застосунків Rails і Node.js і може не підходити для програм, створених за допомогою інших технологій. Має прості параметри розгортання через інтерфейс командного рядка [29].

Платформа розроблена для розміщення Rails- і Node.js-програм, тобто вона не універсальна.

Railway також не пропонує великого переліку налаштувань, що, звичайно, робить його простішим у використанні, але більш обмеженим.

З іншого боку, платформа пропонує безплатну підписку, що підійде проектам з невеликим бюджетом або його відсутністю.

Отже, після порівняння платформ, що були зазначені вище, прийнято рішення для гостингу застосунку використовувати Render, оскільки він пропонує більший різновид застосунків, які можуть розміщуватись на сервері,

простіший інтерфейс користувача в порівнянні з іншими хмарними платформами, а також Render пропонує зрозумілу і детальну документацію з користування.

Висновки

В наслідок проведеного дослідження виявлено, що розробка застосунку для підтримання і покращення комунікації між викладачами та студентами є доцільною та важливою для сучасного освітнього процесу.

Після огляду рішень, що існують, доведено, що застосунки не надають всього потрібного функціоналу, тобто доводиться використовувати декілька застосунків одночасно для того, аби закрити всі потреби.

Створення вебзастосунку є більш доцільним, ніж створення комп'ютерної програми, оскільки спрощується доступ до неї, а також полегшується процес запровадження оновлень. Крім того, такі застосунки є безпечнішими.

Застосунок повинен бути простим та інтуїтивно зрозумілим у використанні, щоб не потребував багато часу на вивчення.

Технічні можливості застосунку надають можливість використовувати його як єдиний допоміжний інструмент до платформ, які запроваджують відеоконференції.

Найкращим варіантом для гостингу застосунку є Render, оскільки підтримує більшу кількість різновидних застосунків, зокрема вебзастосунок, має зручний інтерфейс користувача й зрозумілу документацію.

Застосунок є безпечним, оскільки паролі зберігаються у зашифрованому вигляді, а також запроваджено запобігання SQL-атакам.

Джерела

1. Офіційна сторінка платформи «Microsoft Teams». URL:
<https://www.microsoft.com/fi-fi/microsoft-teams/group-chat-software?rtc=1>
2. Lutkevich B., Bacon M. What is End-to-End Encryption (E2EE) and How Does it Work? Security. URL:
<https://www.techtarget.com/searchsecurity/definition/end-to-end-encryption-E2EE>
3. Microsoft Teams admin documentation - Microsoft Teams. Microsoft Learn: Build skills that open doors in your career. URL:
<https://learn.microsoft.com/en-us/MicrosoftTeams/>
4. Hardware requirements for Microsoft Teams - Microsoft Teams. Microsoft Learn: Build skills that open doors in your career. URL:
<https://learn.microsoft.com/en-us/microsoftteams/hardware-requirements-for-the-teams-app>
5. Telegram – a new era of messaging. Telegram. URL:
<https://telegram.org/>
6. Marr B. The 2 Biggest Future Trends In Education. Forbes. URL:
<https://www.forbes.com/sites/bernardmarr/2022/01/21/the-2-biggest-future-trends-in-education/?sh=a16027f2d6fe>
7. JavaScript | MDN. MDN Web Docs. URL:
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
8. V8 JavaScript engine. V8 JavaScript engine. URL: <https://v8.dev/>
9. Index | Node.js v17.9.1 Documentation. Node.js. URL:
<https://nodejs.org/docs/latest-v17.x/api/>
10. Why Nodejs is so fast? - Nile Bits. Nile Bits. URL:
<https://www.nilebits.com/blog/2022/06/why-nodejs-is-so-fast/>
10. Promise. The Modern JavaScript Tutorial. URL:
<https://javascript.info/promise-basics>
11. Getting Started | Axios Docs. Axios. URL: <https://axios-http.com/docs/intro>

12. React – A JavaScript library for building user interfaces. React – A JavaScript library for building user interfaces. URL: <https://legacy.reactjs.org/>
13. Components and Props – React. React – A JavaScript library for building user interfaces. URL: <https://legacy.reactjs.org/docs/components-and-props.html>
14. Introducing JSX – React. React – A JavaScript library for building user interfaces. URL: <https://legacy.reactjs.org/docs/introducing-jsx.html>
15. Feature Overview v6.11.1. Home v6.11.1 | React Router. URL: <https://reactrouter.com/en/main/start/overview>
16. Get started with Bootstrap. Bootstrap · The most popular HTML, CSS, and JS library in the world. URL: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
17. Loshin P., Sirkin J. What is Structured Query Language (SQL)? Data Management. URL: <https://www.techtarget.com/searchdatamanagement/definition/SQL>
18. Офіційна документація MySQL. URL: <https://dev.mysql.com/doc/>
19. Основи препроцесора Sass: змінні, міксини, оператори, імпорт, вкладеність, спадкування - Linkomania. Linkomania. URL: <http://linkomania.com.ua/osnovi-preprotsesora-sass-zminni-miksini-operatori-import-vkladenist-spadkuvannya/>
20. Sass: Documentation. Sass: Syntactically Awesome Style Sheets. URL: <https://sass-lang.com/documentation/>
21. Material UI: Documentation. URL: <https://mui.com/material-ui/getting-started/overview/>
22. Excel: Documentation. URL: <https://support.microsoft.com/en-us/excel>
23. node.bcrypt.js: Documentation. URL: <https://www.npmjs.com/package/bcrypt>
24. Dan Arias. Hashing in Action: Understanding bcrypt, 2021. URL: <https://auth0.com/blog/hashing-in-action-understanding-bcrypt/>

25. Exploring Containerized Applications: How They Work and Differ from Virtual Machines. URL: <https://traefik.io/glossary/containerized-applications/>
26. Fly.io Docs. URL: <https://fly.io/docs/>
27. Render Docs. URL: <https://render.com/docs>
28. Railway Docs. Railway Docs. URL: <https://docs.railway.app/>