

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко

підпис

«_____» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

«ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ SCREEN CAPTURE»

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 408.21810823

Студент

_____ О. О. Постриган

підпис

«_____» _____ 2022 р.

Керівник ст. викладач

_____ І. О. Кандиба

підпис

«_____» _____ 2022 р.

Консультант канд. техн. наук, доцент

_____ А. О. Алексеева

підпис

«_____» _____ 2022 р.

Миколаїв – 2022

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ	4
ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Дистанційне навчання у освітньому процесі	7
1.2 Постановка задачі.....	8
1.2.1 Опис основних функцій системи	8
1.2.2 Опис ролей користувачів в системі.....	9
1.3 Аналіз аналогічних систем.....	10
1.3.1 Аналіз аналогічних систем: Zoom.....	10
1.3.2 Аналіз аналогічних систем: Google Meet	12
1.3.3 Аналіз аналогічних систем: Moodle ЧНУ Петра могили.....	13
1.4 Специфікації вимог до програмного забезпечення	15
2 МОДЕЛЮВАННЯ ЛОГІЧНОЇ МОДЕЛІ БД ТА ІНТЕРФЕЙСУ.....	22
2.1 Опис сутностей у спроектованій базі даних	22
2.2 Прототипи сторінок	29
Висновок до розділу 2	33
3 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	35
3.1 Опис використаних технологій	35
3.1.1 Node.JS	35
3.1.2 Screen Capture	35
3.1.3 WebRTC	36
3.1.4 MySQL.....	37
3.2 Діаграма класів.....	37
3.2.1 Опис класу Roles	38
3.2.2 Опис класу Users	39
3.2.3 Опис класу Conferences	39
3.2.4 Опис класу Message	40
3.2.5 Опис класу Work	40

3.2.6	Опис класу Favorites	41
3.3	Сценарії використання та діаграма прецедентів.....	41
3.4	Діаграма розгортання.....	46
4.	РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	48
4.1	Розробка бази даних.....	48
4.2	Тестування	51
4.3	Керівництво користувача	56
4.3.1	Авторизація та реєстрація	56
4.3.2	Конференції та чати	58
	Висновки до розділу 4	60
	ВИСНОВКИ.....	61
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ	62
	ДОДАТОК А Код створення бази даних.....	65

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ

IT – інформаційні технології;
API – Application programming interface;
HTML – Hyper Text Markup Language;
CSS – Cascading Style Sheets;
JS – JavaScript;
SQL – Structured Query Language;
JSON – JavaScript Object Notation;
WebRTC – Web Real Time Communication;
SC – Screen Capture;
WAP – Wireless Application Protocol
NAT – Network Address Translation
ER – Entity-relationship

ВСТУП

У зв'язку з останніми подіями з'явилася велика необхідність в системах дистанційного навчання для організації освітнього процесу, вдосконалення та його оптимізації задля підвищення продуктивності процесу здобування освіти. Сутність цих систем полягає у фіксованому контролі освітнього процесу, оцінюванні та проведенні зустрічей для пояснення навчального матеріалу викладачами здобувачам освіти у режимі online за допомогою використання технологій Web Real Time Communication (WebRTC) та Screen Capture.

WebRTC відкриває розширені можливості створювати здвоєний зв'язок між клієнтами та сервером для передачі мультимедійних потоків інформації. WebRTC можна назвати «технологічною переправою», яка надає можливості для справжньої організації та передачі захоплених аудіо та відео потоків. Захопити ці потоки можливо за допомогою технології Screen Capture, без використання додаткових застосунків та розширень, а лише з використанням HTML та JavaScript.

Тема кваліфікаційної роботи є актуальною, оскільки пандемія COVID-19 та наступне вторгнення Російської Федерації на територію України змусили навчальний сектор перевести освітні заклади на дистанційну форму навчання, тому Вебзастосунки, які призначені для організації дистанційного навчання почали набирати велику популярність.

Об'єктом є процес дистанційного навчання закладів вищої освіти.

Предметом є сучасні підходи та технології розробки вебзастосунків на основі інструментарію WebRTC та Screen Capture.

Мета кваліфікаційної роботи – вдосконалення навчального процесу шляхом впровадження технологій WebRTC та Screen Capture у процес дистанційного навчання.

Для досягнення поставленої мети необхідно розв'язати наступні завдання:

– проаналізувати сучасні аналоги та процесів дистанційного навчання;

- розробити вимоги до системи дистанційного навчання;
- проаналізувати ринок наявних технологій і рішень для вирішення завдань;
- спроектувати структуру бази даних та принцип роботи системи дистанційного навчання з використанням технологій WebRTC та Screen Capture;
- реалізувати, протестувати та налагодити вебзастосунок;
- апробувати отримані результати.

Спеціальний розділ про охорону праці є дуже актуальним у наш час тим, що інформаційні технології стали дуже швидко розповсюджуватись та стали сприяти розробці норм та вимог до використання комп'ютерної техніки у навчальному секторі країни. Дані вимоги є регульованими законодавством України.

Апробація результатів. Окремі аспекти дослідження були висвітлені на Всеукраїнській науково-практичній конференції «Могилянські читання – 2021»:

- Постриган О. О., Кандиба І. О. Створення модуля системи дистанційного навчання на основі технології Screen Capture // «Могилянські читання – 2021». Всеукраїнська науково-практична конференція. «Досвід та тенденції розвитку суспільства в Україні: глобальний, національний та регіональний аспекти : тези. Миколаїв : Вид-во ЧНУ імені Петра Могили, 2021. С. 89-90.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дистанційне навчання у освітньому процесі

Підготовка кваліфікованих кадрів з використанням сучасних технологій у галузі освіти зумовила активне впровадження інформаційних технологій у всі сфери діяльності людини.

Згідно із Законом України «Про освіту» [1] та Наказу МОН України «Положення про дистанційне навчання» [2] використання змішаної форми навчання є пріоритетним варіантом здобування освіти. Головною метою «змішаного навчання» є правильне та ефективне розподілення навчального матеріалу та матеріалу, який відводиться на віддалене вивчення студентам та практичний матеріал студентам в аудиторіях.

«Змішане навчання» об'єднує в собі традиційне та дистанційне навчання, яке за допомогою інформаційно-комп'ютерних технологій допомагає організувати освітній процес в такому середовищі, в якому викладачі та учні зможуть навчатися у зручних для себе обставинах [3].

Під дистанційним навчанням мається на увазі організація освітнього процесу із використанням збереженої в базі даних і використаної під час підготовки освітніх програм інформації та спроби забезпечення нею обробки технологій, засобів і навіть інтернет мереж, які забезпечують трансляцію інформації між усіма учасниками процесу освіти [4].

Дистанційними освітніми технологіями виступають технології, які реалізовані із застосуванням інформаційних мереж при взаємодії учнів та системи дистанційного навчання або викладачів [4].

Хоч дистанційне навчання і набуло великої популярності лише останнім часом, але вона не нова. Навпаки, можна сказати, що воно існує вже багато десятиліть у вигляді заочних курсів. Друге життя вона отримала з розвитком інформаційних технологій.

Розвиток інформаційних технологій в останній час був стимульований пандемією COVID-19 та наступним вторгненням Російської Федерації на

територію України, що і зумовило створення систем дистанційного навчання для організації освітнього процесу.

Система дистанційного навчання – це сукупність технологій, які допомагають забезпечити освітній процес між учнями та викладачами для пояснення навчального матеріалу за допомогою інтерактивної взаємодії учнів з викладачами та можливості самостійної роботи з засвоєння матеріалу, який вивчається у процесі навчання.

Після аналізу визначення поняття «дистанційна освіта», можна зробити висновок, що дистанційне навчання – це процес інтерактивної взаємодії здобувача освіти з викладачем під час здобування освіти, який спирається на використання телекомунікаційних та інформаційних технологій, які дозволяють здійснювати навчальний процес на відстані, тобто надає можливість організувати процес самоосвіти [5].

Таким чином, учні та викладачі більше не мають потреби бути присутніми в аудиторії для того щоб вчитися. Це дозволяє керувати інформацією та часом для кращої організації освітнього процесу, роблячи його дуже схожим на особисте навчання, що дозволяє покращити якість засвоєння навчального матеріалу.

1.2 Постановка задачі

1.2.1 Опис основних функцій системи

Система дистанційного навчання з використанням технології Screen Capture, яка розробляється в ході кваліфікаційної роботи бакалавра, має наступний набір функціональних можливостей:

- 1) Реєстрація користувачів в системі та подальше розмежування доступу залежно від ролі в системі.
- 2) Авторизація користувачів в системі та подальше надання доступу до системи залежно від ролі, яку отримав користувач.
- 3) Створення конференції.
- 4) Спілкування за допомогою аудіо та відео.

5) Відправка повідомлень конференції (функція доступна усім учасникам конференції).

6) Видалення повідомлень в конференції (функція доступна лише адміністратору та модератору конференції).

7) Редагування повідомлень в конференції (функція доступна усім учасникам конференції).

8) Використання технології демонстрації екрану (функція доступна усім учасникам конференції).

9) Створення форм для завантаження завдання з його подальшим виконанням учнями (функція доступна лише адміністратору).

10) Завантаження фото конференції (функція доступна лише адміністратору конференції).

11) Завантаження файлів до форми з завданням (функція доступна лише учасникам конференції).

12) Пошук повідомлень в конференції (функція доступна лише учасникам конференції).

13) Додавання користувачів до списку друзів конференції (функція доступна усім користувачам).

14) Пошук користувачів (функція доступна усім користувачам).

15) Журнал та оцінка завдань (функція доступна лише адміністратору конференції).

16) Перегляд оцінок (функція доступна лише учасникам конференції).

1.2.2 Опис ролей користувачів в системі

- Гість (неавторизований користувач).
- Зареєстрований користувач.
- Модератор.
- Адміністратор.

1.3 Аналіз аналогічних систем

На даний момент сфера дистанційного навчання має у наявності велику кількість як комерційних, так і безкоштовних вебзастосунків, які допомагають організувати освітній процес, але кожен з них має як свої недоліки, так і свої переваги. Через це потрібно виділити недоліки, які потрібно буде виправити у системі, яка розробляється.

Під час аналізу аналогічних систем виникли невеликі складнощі, тому що вони не мають ліцензію opensource проектів, через що майже неможливо дізнатися детальнішу інформацію, ніж було добуто під час аналізу аналогічних систем дистанційного навчання.

Головним недоліком усіх систем дистанційного навчання є те, що вони мають порізно функцій, які для кращої роботи потрібно об'єднати в один застосунок. Тобто можна зробити висновок, що на даний момент майже не існує застосунків, які б мали всі функції одночасно.

1.3.1 Аналіз аналогічних систем: Zoom

Першим аналогом системи дистанційного навчання, яка має велику популярність та розповсюдження можна назвати Zoom. Сервіс набув велику базу прихильників завдяки легкості використання відеоконференцій і відсутності зайвих функцій. Zoom позиціонує себе як система дистанційного проведення відеоконференцій у форматі високої чіткості [6]. Даний програмний продукт розроблений 2 квітня 2013 року компанією Zoom Video Communications. Zoom надає користувачу можливість організувати відеоконференції, які дозволяють підключитись до 100 пристроїв одночасно, але з 40 хвилинним обмеженням для безкоштовних акаунтів. Мовами реалізації даного програмного забезпечення є Java та C++ [7].



Рисунок 1.1 – Зовнішній вигляд Zoom [8]

Нижче представлено перелік *основних функцій* даного програмного забезпечення [6]:

- можливість додавати слухачів;
- обмін повідомленнями;
- можливість використовувати демонстрацію екрану;
- можливість спілкуватися за допомогою голосового зв'язку;
- реєстрація та авторизація;
- відправлення приватних повідомлень.

Переваги Zoom [9]:

- стильний та зрозумілий інтерфейс;
- можливість розширення функціоналу;
- можливість обрати бажану мову локалізації інтерфейсу;
- велика спільнота;
- можливість підключитися до конференції без авторизації у системі.

Недоліки Zoom [9]:

- 40 хвилин безкоштовного спілкування за сеанс;

- комерційний продукт;
- відсутня можливість завантаження завдань;
- відсутня система оцінок та журнал користувачів;
- слабка система захисту конференцій;
- відсутність відображення прогресу при виконанні завдань.

Отже, Zoom є найбільш популярною та розвинутою системою дистанційного навчання, але має дуже слабку систему захисту конференцій. Розробники продовжують розвивати свій продукт, але деякі функції, на жаль, працюють некоректно.

1.3.2 Аналіз аналогічних систем: Google Meet

Google Meet – це система дистанційного проведення відеоконференцій розроблена компанією Google у березні 2017 року. Також як і Zoom дозволяє створювати захищену відеоконференцію, але на відміну Zoom він має безкоштовну ліцензію на використання та не має часового обмеження. Мовами реалізації даного програмного забезпечення є Java та C++ [10].



Рисунок 1.2 – Зовнішній вигляд Google Meet [12]

Нижче представлено перелік *основних функцій* даного програмного забезпечення:

- обмін повідомленнями;
- створення конференцій;
- спілкування за допомогою аудіо зв'язку;
- спілкування за допомогою відео зв'язку;
- реєстрація та авторизація користувача.

Переваги Google Meet [11]:

- простий та інтуїтивно зрозумілий інтерфейс;
- шифрування конференцій;
- велика спільнота;
- відсутні часові обмеження.

Недоліки Google Meet [11]:

- відсутня можливість завантаження завдань;
- відсутня система оцінок та журнал користувачів;
- відсутність відображення прогресу при виконанні завдань;
- відсутня можливість розширення функціоналу за допомогою плагінів.

Отже, Google Meet є популярною та розвинутою системою дистанційного навчання, та при цьому має дуже чудову систему захисту конференцій. Розробники провели дуже чудову роботу над тестуванням застосунку, що позитивно відобразилось на його стабільній роботі.

1.3.3 Аналіз аналогічних систем: Moodle ЧНУ Петра могили

Moodle ЧНУ Петра Могили – це система дистанційного навчання. Moodle ЧНУ Петра Могили на відміну від Google Meet і Zoom не дозволяє створювати захищену відеоконференцію та не може демонструвати зображення на екрані, але він має безкоштовну ліцензію на використання та не має часового обмеження. Він позиціонує себе як LMS (Learning Management System), тобто це програмне забезпечення, яке допомагає при адмініструванні, відстежуванні,

складанні звітів та проведенні освітніх курсів та програм навчання. Мовою реалізації даного програмного забезпечення є PHP.

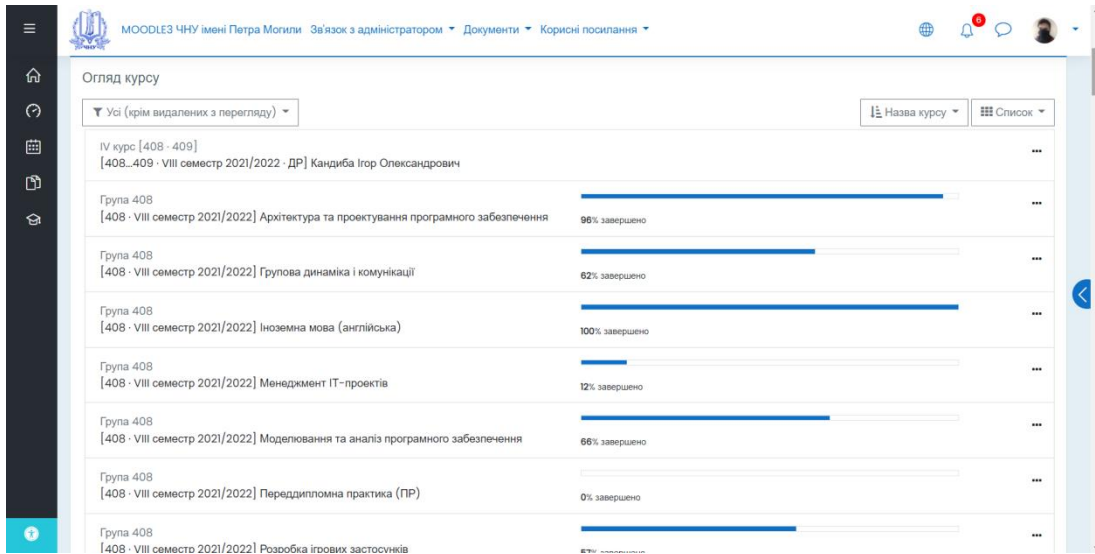


Рисунок 1.3 – Зовнішній вигляд Moodle ЧНУ Петра Могили

Нижче представлено перелік *основних функцій* даного програмного забезпечення:

- обмін повідомленнями;
- створення курсів;
- створення форм для завантаження виконаного ДЗ;
- реєстрація та авторизація користувача.

Переваги Moodle ЧНУ Петра Могили:

- система оцінок;
- велика спільнота;
- відсутні часові обмеження.

Недоліки Moodle ЧНУ Петра Могили:

- незбалансований інтерфейс;
- відсутня система відеоконференцій;
- відсутність демонстрації екрана;
- відсутня можливість розширення функціоналу за допомогою плагінів;
- проблеми з адаптивністю інтерфейсу.

Отже, Moodle ЧНУ ім. Петра Могили є розвинутою системою дистанційного навчання. Розробники продовжують розвивати свій продукт, але деякі функції, на жаль, працюють некоректно.

1.4 Специфікації вимог до програмного забезпечення

ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

Призначення системи (застосунку), для якої розробляється програмне забезпечення

Даний вебзастосунок створено для організації дистанційного навчання за допомогою новітніх технологій з передачі аудіо та відео потоків.

Погодження, що ухвалені в програмній документації

Для створення вебзастосунку та його злагодженої роботи було погоджено рішення використати допоміжні фреймворки – Node.JS та Angular.

Межі проекту ПЗ

Дедлайн роботи над вебзастосунком – 31.05.2022р.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Даний вебзастосунок не має обмежень у сферах його застосування та рекомендований для використання у сфері освіти.

Характеристики користувачів

Основні характеристики користувачів: мати при собі смартфон, планшет або персональний комп'ютер з підключенням до мережі інтернет.

Загальна структура системи

Програмне забезпечення складається з наступних частин: backend; frontend; database.

Загальні обмеження

Єдиним обмеженням роботи вебзастосунку є наявність інтернет зв'язку та його швидкість.

ФУНКЦІЇ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ SCREEN CAPTURE

Функція пошуку повідомлення в конференції

Опис функції

Функція пошуку повідомлення в конференції допомагає знайти повідомлення, які знаходяться в конференції за ключовим словом.

Вхідна та вихідна інформація

Вхідна інформація – заданий текст;

Вихідна інформація – повідомлення, які співпадають з введеним текстом.

Функціональні вимоги

База даних з збереженими повідомленнями та мережа Інтернет.

Функція створення конференції

Опис функції

Функція створення конференції допомагає створити конференцію.

Вхідна та вихідна інформація

Вхідна інформація – назва, опис та фото;

Вихідна інформація – створена конференція.

Функціональні вимоги

З'єднання з базою даних та мережа Інтернет.

Функція спілкування за допомогою аудіо та відео

Опис функції

Функція спілкування за допомогою аудіо та відео допомагає організувати мультимедійні конференції з інтерактивним поясненням матеріалу.

Вхідна та вихідна інформація

Вхідна інформація – обрана конференція;

Вихідна інформація – транслявання аудіо та відео потоків.

Функціональні вимоги

Доступ до бази даних та мережа Інтернет.

Функція відправлення повідомлень

Опис функції

Функція дозволяє відправляти повідомлення в чаті конференції.

Вхідна та вихідна інформація

Вхідна інформація – обрана конференція, повідомлення;

Вихідна інформація – відправлене повідомлення.

Функціональні вимоги

Доступ до бази даних та мережа Інтернет.

Функція редагування повідомлення

Опис функції

Функція допомагає редагувати відправлене користувачем повідомлення.

Вхідна та вихідна інформація

Вхідна інформація – конференція, обране повідомлення;

Вихідна інформація – відредаговане повідомлення.

Функціональні вимоги

Доступ до бази даних та мережа Інтернет.

Функція видалення повідомлення

Опис функції

Функція допомагає видалити відправлене користувачем повідомлення.

Вхідна та вихідна інформація

Вхідна інформація – конференція, обране повідомлення;

Вихідна інформація – видалене повідомлення.

Функціональні вимоги

Доступ до бази даних, роль адміністратора та мережа Інтернет.

Функція демонстрації екрана

Опис функції

Функція дозволяє користувачу транслювати контент з робочого столу свого пристрою.

Вхідна та вихідна інформація

Вхідна інформація – конференція, захоплена область робочого столу пристрою;

Вихідна інформація – транслявання відео та аудіо потоків.

Функціональні вимоги

Доступ до бази даних та мережа Інтернет.

Функція створення форм для завантаження виконаного завдання

Опис функції

Функція дозволяє створити певну форму в яку потім можна завантажити виконане завдання.

Вхідна та вихідна інформація

Вхідна інформація – назва завдання, опис завдання, оцінка;

Вихідна інформація – готова форма завдання.

Функціональні вимоги

Доступ до бази даних, роль адміністратора та мережа Інтернет.

Функція перегляду оцінки за завдання

Опис функції

Функція дозволяє переглянути оцінку за виконане завдання.

Вхідна та вихідна інформація

Вхідна інформація – обране завдання;

Вихідна інформація – оцінка.

Функціональні вимоги

Доступ до бази даних та мережа Інтернет.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Джерела і зміст вхідної інформації (даних)

В даному вебзастосунку джерелом вхідної інформації є користувач.

Нормативно-довідкова інформація (класифікатори, довідники тощо)

Вимоги до даного пункту відсутні.

Вимоги до способів організації, збереження та ведення інформації

Обмін даними відбувається на основі використання Rest API. В якості

бази даних для програмного забезпечення обрано MySQL.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Комп'ютер чи ноутбук може бути з будь-якими характеристиками, але не нижче наступних:

- ОС: Manjaro Linux
- CPU: Intel Core i3-6006u 2.00GHz;
- RAM: 16GB;
- SSD: 1TB;

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмної системи

Архітектура програмної системи складається з наступних складових:

- Backend – NodeJS, Express, WebRTC;
- Frontend – Angular;
- Data Base – MySQL.

Мережне програмне забезпечення

Для створення вебзастосунку потрібно встановити наступні компоненти:

- ОС Manjaro Linux;
- Редактор коду: Visual Studio Code;
- Пакетний менеджер NPM;
- Будь-який браузер для перегляду вебсторінок.

Програмне забезпечення ведення інформаційної бази

Робота з базою даних буде виконуватись за допомогою MariaDB.

Мова і технологія розробки ПЗ

Вебзастосунок повинен бути розроблений за допомогою фреймворків Angular та NodeJS. Мови розробки JavaScript.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Інтерфейс застосунку повинен задовольняти усі вимоги UX/UI, що дуже позитивно вплине на розуміння роботи застосунку.

Адаптивний інтерфейс

Адаптивним інтерфейсом є девайс, яким користувач користується (персональний комп'ютер, ноутбук, планшет, смартфон).

Програмний інтерфейс

Angular

Комунікаційний протокол

Вебзастосунок повинен використовувати протокол бездротового доступу WAP.

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

Кожен користувач, який бажає скористатися послугами застосунку може користуватися ним безкоштовно.

Супроводження

Вебзастосунок повинен постійно вдосконалюватися та супроводжуватися.

Переносимість

Програмне забезпечення повинне працювати на будь-яких девайсах та операційних системах, які мають встановлений браузер останньої версії.

Продуктивність

Швидкість інтернету є найважливішим фактором при використанні даного програмного забезпечення. Продуктивність вебзастосунку вимірюється у кількості запитів в секунду. Відправлення запиту не повинне перевищувати 2-х секунд.

Надійність

Усі дані користувачів повинні залишатися приватними і не повинні потрапити до сторонніх користувачів. Кожен користувач обов'язково повинен отримати доступ до своїх даних лише після авторизації у системі.

Безпека

Дані користувача при реєстрації та авторизації повинні буди зашифровані

та захищені.

Висновки до розділу 1

В першому розділі проведено аналіз аналогічних систем дистанційного навчання. В результаті аналізу було визначено основні переваги та недоліки проаналізованих систем. Визначено, що деякі вебзастосунки є комерційними розробками, тому вони не розповсюджуються під ліцензією opensource, через що майже неможливо здобути більш детальнішу інформацію. Сформовані основні вимоги щодо створення програмного забезпечення. Вебзастосунок буде відповідати усім стандартам дистанційної освіти.

2 МОДЕЛЮВАННЯ ЛОГІЧНОЇ МОДЕЛІ БД ТА ІНТЕРФЕЙСУ

2.1 Опис сутностей у спроектованій базі даних

Моделлю «сутність-зв'язок» називають модель, яка описує концептуальні схеми за допомогою певних узагальнених конструкцій [15].

Предметне моделювання в моделі «сутність-зв'язок» базується на використанні графічних діаграм, які включають невелику кількість різних компонентів. Головними постулатами ER-моделі є сутність, зв'язок і атрибут. ER-моделлю виступає модель даних, або як її ще називають, засіб опису моделей даних [13].

Сутність – це представлений об'єкт, інформація про який повинна зберігатися в базі даних. На діаграмі ER-моделі сутність зображується у вигляді прямокутника з її назвою. Кожна сутність завжди повинна мати унікальне ім'я. Ім'я сутності - це ім'я типу об'єкта, а не конкретний приклад цього типу [13].

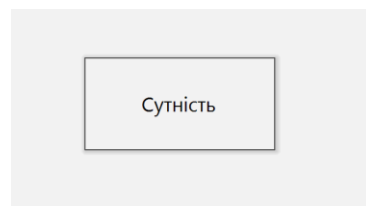


Рисунок 2.1 – Сутність

Атрибут – це характеристика сутності, яка призначена для ідентифікації, кількісної характеристики, класифікації або відношення стану сутності [13]. На діаграмі атрибут зображується у вигляді прямокутника з його назвою [14].

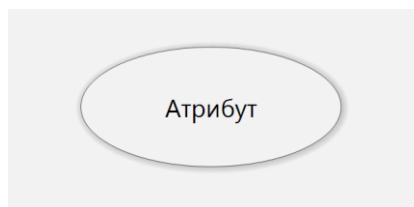


Рисунок 2.2 – Атрибут

Зв'язок – це асоціація між двома або більше сутностями, що є суттєвою для аналізованої предметної області [16].

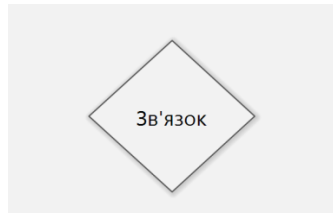


Рисунок 2.3 – Зв'язок

Зв'язок позначають дієсловом: «відноситься», «належить». Він відображає відношення однієї сутності до іншої. Можна виділити такі типи зв'язку:

- один до багатьох, наприклад завод і випущені трактори;
- багато до багатьох, наприклад вчителі і студенти.

Сутність «Користувачі» має в собі інформацію про користувачів, при цьому містить 9 звичайних атрибутів та 2 зовнішніх ключі. Зовнішні ключі мають відношення один до більшості і містять у собі посилання на сутності «Ролі» та «Конференції».



Рисунок 2.4 – Логічна модель сутності «Користувачі»

Звичайні атрибути:

- ідентифікатор користувача;
- ім'я користувача;
- прізвище користувача;
- логін користувача;
- пароль користувача;
- посилання на фото користувача;
- дата створення аккаунту користувача;
- дата редагування аккаунту користувача;
- статус користувача.

Зовнішні ключі:

- ідентифікатор ролі;
- ідентифікатор конференції.

Сутність «Ролі» має в собі інформацію про ролі, і при цьому містить 5 звичайних атрибутів.

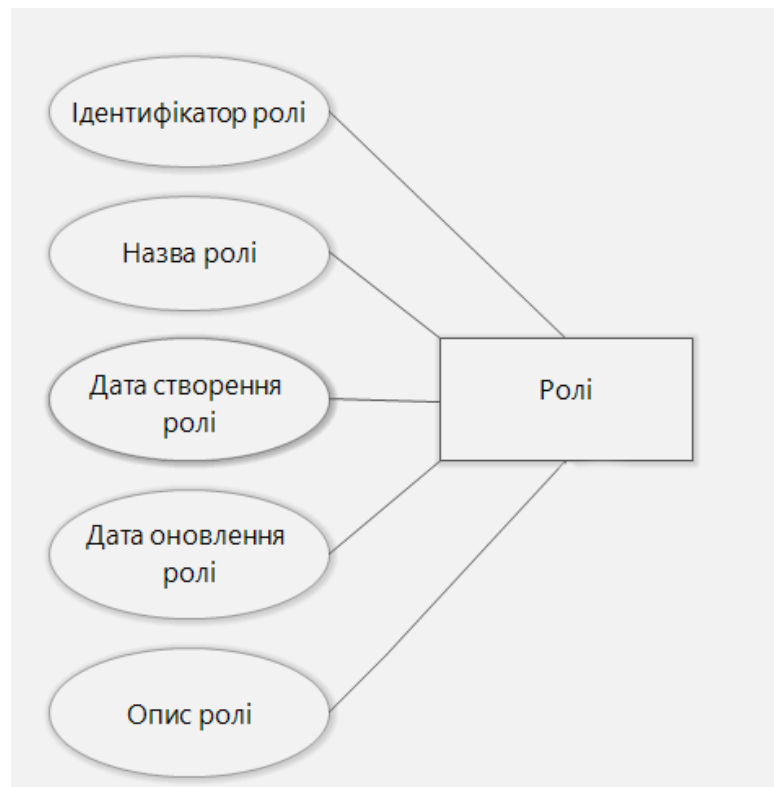


Рисунок 2.5 – Логічна модель сутності «Ролі»

Звичайні атрибути:

- ідентифікатор ролі;
- назва ролі;
- дата створення ролі;
- дата оновлення ролі;
- опис ролі.

Сутність «Конференції» має в собі інформацію про конференції користувачів, і при цьому містить 5 звичайних атрибутів.

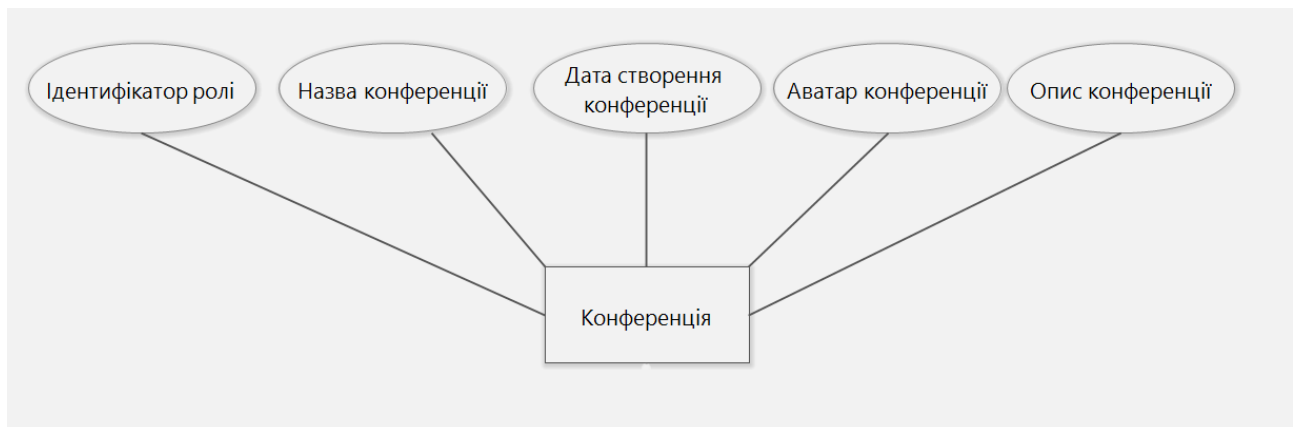


Рисунок 2.6 – Логічна модель сутності «Конференції»

Звичайні атрибути:

- ідентифікатор конференції;
- назва конференції;
- дата створення конференції;
- дата оновлення конференції;
- посилання на фото конференції;
- опис конференції.

Сутність «Повідомлення» має в собі інформацію про повідомлення, користувачів які відправили повідомлення, у якій конференції, і при цьому містить 3 звичайних атрибутів та 2 зовнішніх ключі. Зовнішні ключі мають відношення один до більшості і містять у собі посилання на сутності «Користувачі» та «Конференції».

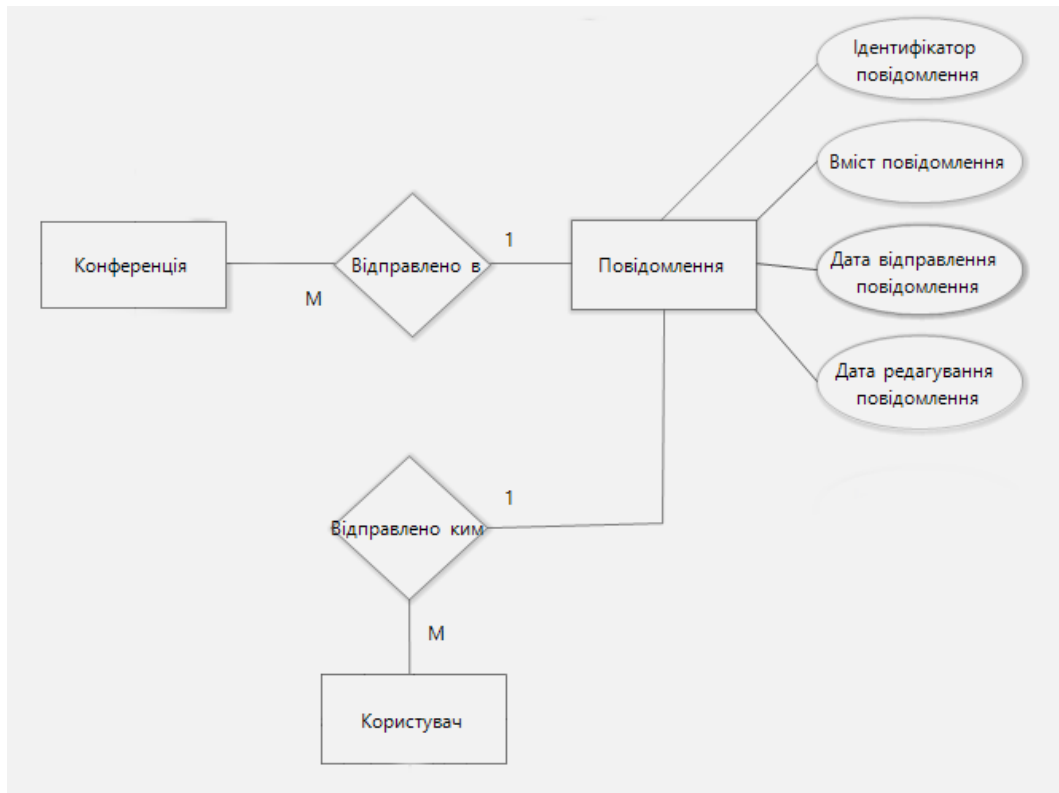


Рисунок 2.7 – Логічна модель сутності «Повідомлення»

Звичайні атрибути:

- ідентифікатор повідомлення;
- вміст повідомлення;
- дата відправлення повідомлення;
- дата редагування повідомлення.

Зовнішні ключі:

- ідентифікатор користувача;
- ідентифікатор конференції.

Сутність «Завдання» має в собі інформацію про завдання користувачів, і при цьому містить 6 звичайних атрибутів та 1 зовнішній ключ. Зовнішні ключі мають відношення один до більшості і містять у собі посилання на сутності «Користувачі».

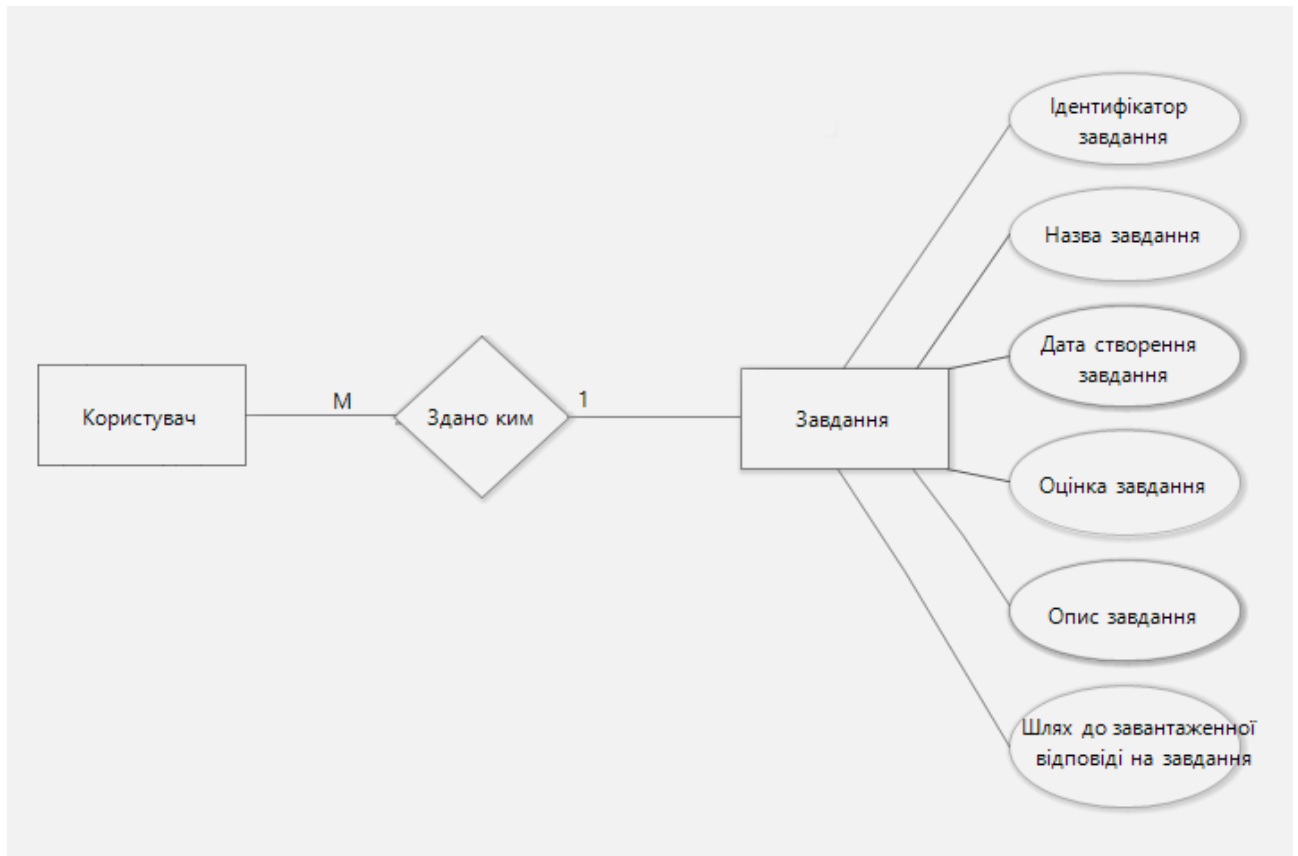


Рисунок 2.8 – Логічна модель сутності «Завдання»

Звичайні атрибути:

- ідентифікатор завдання;
- назва завдання;
- оцінка завдання;
- суть завдання;
- шлях до завантаженої відповіді на завдання;
- дата створення завдання.

Зовнішні ключі:

- ідентифікатор користувача.

Сутність «Друзі» має в собі інформацію про список користувачів, яких користувач додав до списку друзів і при цьому містить 6 звичайних атрибутів та 1 зовнішній ключ. Зовнішні ключі мають відношення один до більшості і містять у собі посилання на сутності «Користувачі».

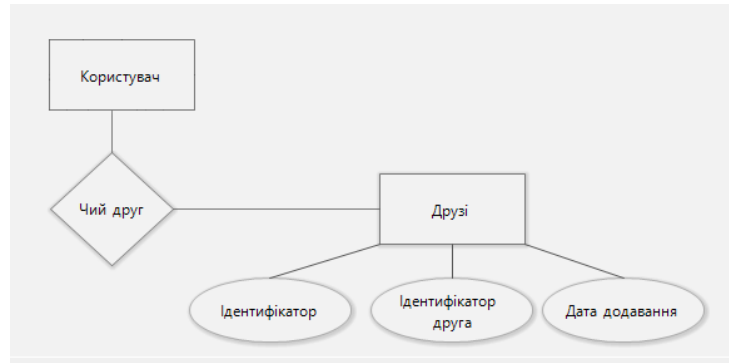


Рисунок 2.9 – Логічна модель сутності «Друзі»

Звичайні атрибути:

- ідентифікатор;
- ідентифікатор доданого користувача;
- дата додавання.

Зовнішні ключі:

- ідентифікатор користувача.

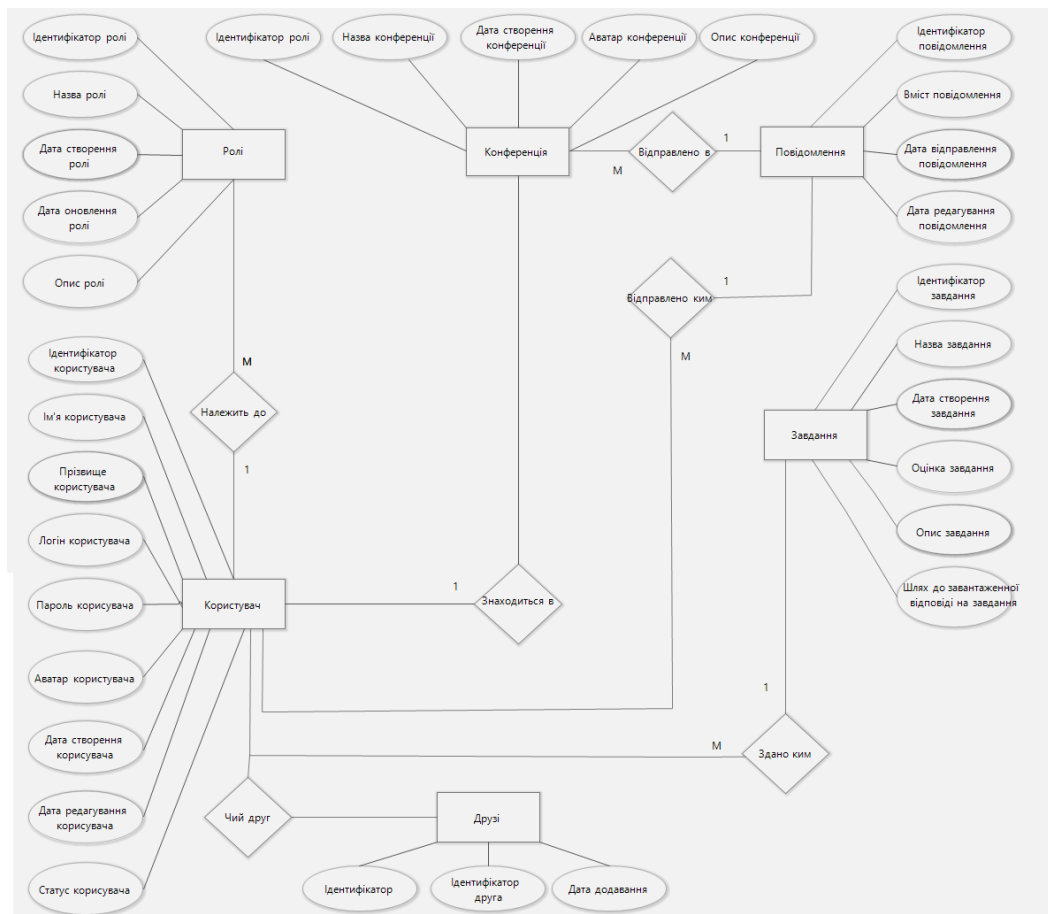


Рисунок 2.10 – Логічна структура всіх сутностей

Таблиця 2.1 – Опис сутностей у спроектованій базі даних

Назва сутності	Опис вмісту
Users	Таблиця містить в собі дані користувачів.
Conferences	Таблиця містить в собі дані конференції та користувачів, які знаходяться у конференції.
Favorites	Таблиця містить в собі список користувачів, яких користувач додав до друзів.
Message	Таблиця містить в собі відправлені користувачем повідомлення та ідентифікатор користувача.
Roles	Таблиця містить в собі список ролей.
Work	Таблиця містить в собі інформацію про завдання користувачів.

2.2 Прототипи сторінок

Mock-up – це макет, який використовують дизайнери для візуалізації та демонстрації замовникові програмного продукту. Вони дозволяють продемонструвати свої напрацювання не в «чорній коробці», а в зв'язці з предметами, для яких воно розробляється [17].

Для кращого розуміння та візуалізації інтерфейсу було розроблено mock-up сторінок. Вони допомагають побачити приблизний макет інтерфейсу, який має отримати застосунок на релізі [17].

Неавторизований користувач, в першу чергу, потрапляє на сторінку з формою авторизації. Якщо користувач вже зареєстрований у системі, тоді він вводить електронну пошту та пароль, після чого натискає кнопку «Увійти». Якщо користувач ще незареєстрований у системі, або просто забув пароль, тоді можна натиснути відповідну кнопку «Відновити пароль» чи «Реєстрація», що можна побачити на рисунку 2.11.

The image shows a registration form for ADCU. At the top, it says 'ADCU' and 'Зареєструйтесь на сайті за допомогою номеру телефона або електронної пошти'. Below this are several input fields: 'Логін' (Login) with 'Введіть логін', 'ФІО' (Full Name) with 'Введіть ФІО', 'Електронна пошта' (Email) with 'cuda735@gmail.com', 'Телефон' (Phone) with '+380(XX)XXX-XX-XX', 'Пароль' (Password) with '*****', and 'Підтвердження пароля' (Password Confirmation) with '*****'. There are two radio buttons: 'Викладач' (Lecturer) and 'Студент' (Student). A checkbox is checked, with the text 'Я погоджуюсь із правилами використання сервісу, а також із передачею та обробкою моїх даних.' Below the form are three buttons: 'Зареєструватись', 'Увійти', and 'Відновити пароль'.

Рисунок 2.11 – Маскун реєстрації

На сторінці реєстрації необхідно вигадати логін та пароль, обов'язково ввести ФІО з електронною поштою та обрати тип акаунту користувача, після чого, підтвердити згоду на обробку персональних даних. Якщо користувачем дані були введені правильно, то користувач перейде до головної сторінки. Вибір типу користувача впливає на подальше надання функціоналу, який отримає користувач під час використання програмного забезпечення, що і продемонстровано на рисунку 2.12. Для викладачів він має більшу кількість функціоналу, наприклад:

- створення конференцій;
- створення спеціальних форм з завданнями для подальшого завантаження відповідей користувачів;
- надання ролей доступу іншим користувачам;
- управління конференціями.

ADCU

Увійдіть на сайт за допомогою електронної пошти

Електронна пошта: cuda735@gmail.com

Пароль: *****

Запам'ятати мене

Увійти

Вперше на нашому сайті? Зареєструйтесь

Ви забули свій пароль? Відновити пароль

Рисунок 2.12 – Маск авторизації

У чаті користувач має можливість писати та відправляти, редагувати та видаляти повідомлення. Такий користувач може підключитися або почати відеоконференцію натиснувши кнопку «Відеоконференція».

ADCU

Ім'я конференції: [X] [Відеоконференція] [Інше]

Ім'я користувача: [X]
Повідомлення користувача

Поле для введення [Відправити]

Рисунок 2.13 – Маск чату

У режимі «конференції» користувач має змогу транслювати та вимикати трансляцію відео та аудіо потоків. Також користувач має можливість надсилати повідомлення до чату конференції.

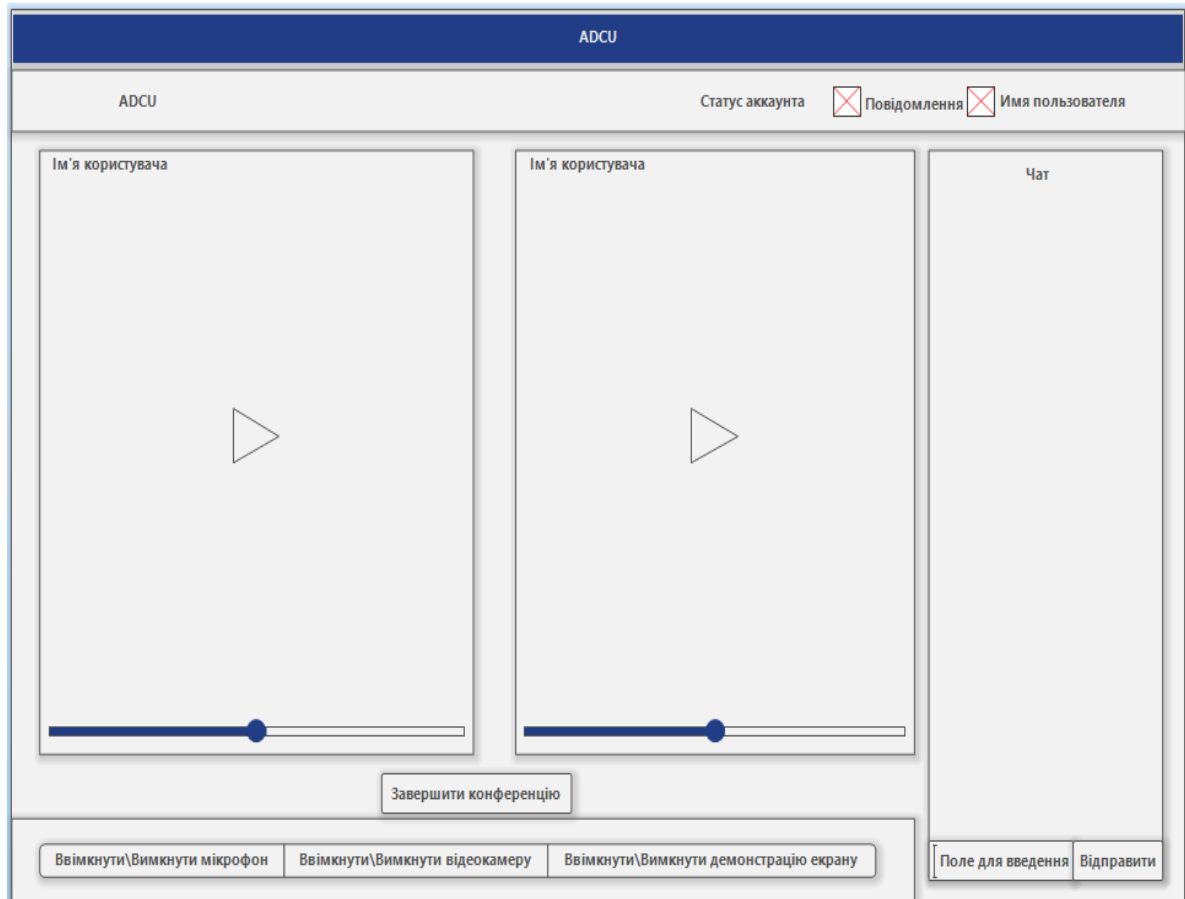


Рисунок 2.14 – Мокир конференції

На сторінці «Завдання», користувач має змогу побачити на верхній панелі статус акаунту користувача, перейти до повідомлень та перейти до перегляду або редагування інформації користувача, яка знаходиться у акаунті. В формі «Завдання» користувач може побачити наступне:

- назву завдання;
- ім'я викладача, який створив форму для завантаження завдання;
- дату виконання;
- назву завдання;
- опис завдання;
- оцінку;

- статус завдання;
- кнопку «Завантажити», яка дозволяє завантажити виконане завдання.

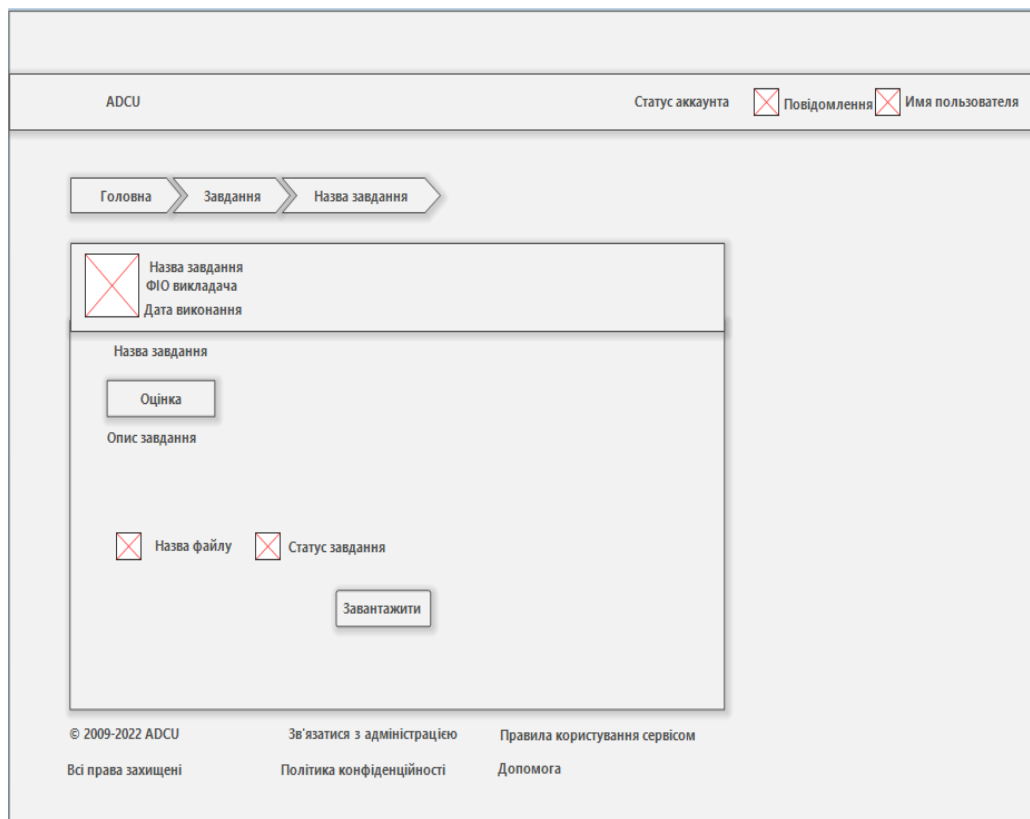


Рисунок 2.15 – Мок-ап завдання

Висновок до розділу 2

В другому розділі описано моделювання програмного забезпечення для дистанційного навчання з використанням системи Screen Capture.

Представлено логічну модель за допомогою діаграми «сутність-зв'язок», яка містить 6 сутностей, які пов'язані між собою зв'язками один до багатьох. Найбільшою сутністю у системі є сутність «Користувачі», яка містить дані користувачів.

У другому пункті було розроблено mock-up інтерфейсу системи, як реєстрація, авторизація, конференція, чат, завдання. Вони дозволяють продемонструвати свої напрацювання не в «чорній коробці», а в зв'язці з предметами, для яких воно розробляється. Для кращого розуміння та візуалізації інтерфейсу було розроблено mock-up сторінок.

3 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Опис використаних технологій

Вебзастосунок працює на основі введення та виведення великої кількості інформації, тому під час написання серверної частини потрібно рухатись у напрямку великої швидкості обробки запитів. До найпоширеніших backend технологій відносяться наступні технології: Python, Java, C# та Node.JS. Node.JS найкраще підходить до поставлених завдань за ряд своїх особливостей та переваг над іншими.

3.1.1 Node.JS

Переваги *Node.JS* [18]:

– **Легкість вивчення.** Node.JS використовує найпопулярнішу мову програмування JavaScript, яка є легкою у вивченні та має велику спільноту розробників.

– **Швидкість.** Node.JS використовує швидкий інтерпретатор JavaScript коду під назвою V8, який розробила компанія Google. Інтерпретатор компілює JavaScript у машинний код, що прискорює і полегшує виконання коду.

Бібліотека *Express* вирішує велику кількість складнощів при розробці вебзастосунку та допомагає легко працювати з обробником HTTP маршрутів.

3.1.2 Screen Capture

Screen Capture – це технологія, що набула широкої популярності у різних галузях людської діяльності: проведення дистанційних конференцій, взаємодія з колегами, технічна підтримка та ін. Основна можливість цієї технології – демонстрація екрану користувача. Актуальність технології Screen Capture зростає через пандемію COVID – 19, коли весь світ відчув потребу у засобах комунікації в режимі реального часу. Особливо це на собі відчула система освіти, адже виникла потреба організовувати дистанційне навчання в режимі реального часу для демонстрації та пояснення навчального матеріалу. З описаного вище, можна зробити висновок, що технологія Screen Capture є дуже

корисною, а створення системи дистанційного навчання на основі цієї технології актуальне.

Існують лише три поверхні захоплення зображення з дисплею [19]:

- весь екран (захоплення всіх вікон додатків які знаходяться на робочому столі вашого комп'ютера);
- вікно (захоплення певного вікна запущеного застосунку);
- вкладка (захоплення вмісту вкладки браузера);

Поверхні можна розділити на два типи: видиму і логічну. Видима поверхня – це поверхня, яка відображається великим планом на екрані, логічна поверхня – це поверхня, яка закрита частково, або перекрита іншим об'єктом [19].

Захоплення контенту, який знаходиться на екрані відбувається у вигляді потоку `MediaStream`, що, в свою чергу, запускається викликом методу `navigator.mediaDevices.getDisplayMedia`. В результаті повертаємо об'єкт `MediaStream`, який містить в собі захоплене зображення з екрана. Це надає можливість використовувати захоплений потік всюди, де йде приймання потоку в якості вступних параметрів. Захоплення звуку є необов'язковим навіть тоді, коли контент запитує потік зі звуком і відео, об'єкт `MediaStream`, який ми повертаємо може як і раніше мати в складі лише відеодоріжку без звуку. Для запиту доступу до дисплея зі звуком, в метод `getDisplayMedia()` передаються параметри `video: true` та `audio: true` [19].

3.1.3 WebRTC

WebRTC – це стек технологій, який містить у собі набір аудіо та відео кодеків для організації транспортних протоколів підключень між клієнтськими пристроями за технологією «точка-точка» [20].

API має два основні інтерфейси: `Stream API` та `PeerConnection`.

Stream API виконує роль отримувача медіапотоку від апаратного забезпечення клієнта та подальшого перетворення його на аудіо та відео потік даних [21].

PeerConnection API виконує роль створення та узгодження з'єднання, обходу NAT, проксі або фаєрволу, та передачі потоку даних через мережу [21].

Переваги WebRTC [22]:

- **Якість.** WebRTC використовує сучасні аудіо та відео кодеки, які допомагають досягти високої якості без додаткових маніпуляцій. WebRTC також має вбудовану функцію усунення звукових перешкод.
- **Адаптивність.** WebRTC має можливість регулювати якість аудіо та відео потоків залежно від якості інтернет з'єднання.

3.1.4 MySQL

MySQL – це система керування базами даних, яка використовується для підтримки реляційних баз даних [23].

Переваги **MySQL** [24]:

- **Масштабованість.** MySQL допомагає забезпечити чудову масштабованість для керування роботи з застосунками навіть у великих сховищах даних.
- **Безпека.** MySQL є захищеною системою управління базами даних. Остання версія MySQL має можливість забезпечити безпеку транзакційної обробки, що у свою чергу забезпечує велику користь у випадку будь-якої електронної комерції, яка вимагає великої кількості грошових переказів.
- **Продуктивність.** MySQL позиціонує себе як рішення для найвибагливіших застосунків та пропонує унікальну систему збереження даних дозволяючи розробникам налаштувати сервер без будь-якого недоліку, що надає можливість забезпечити високу продуктивність виконуючи велику кількість запитів в секунду.

3.2 Діаграма класів

Діаграма класів – це тип UML діаграми, яка допомагає візуалізувати структуру системи з точки зору класів, атрибутів, методів та зав'язків між різними класами [25].

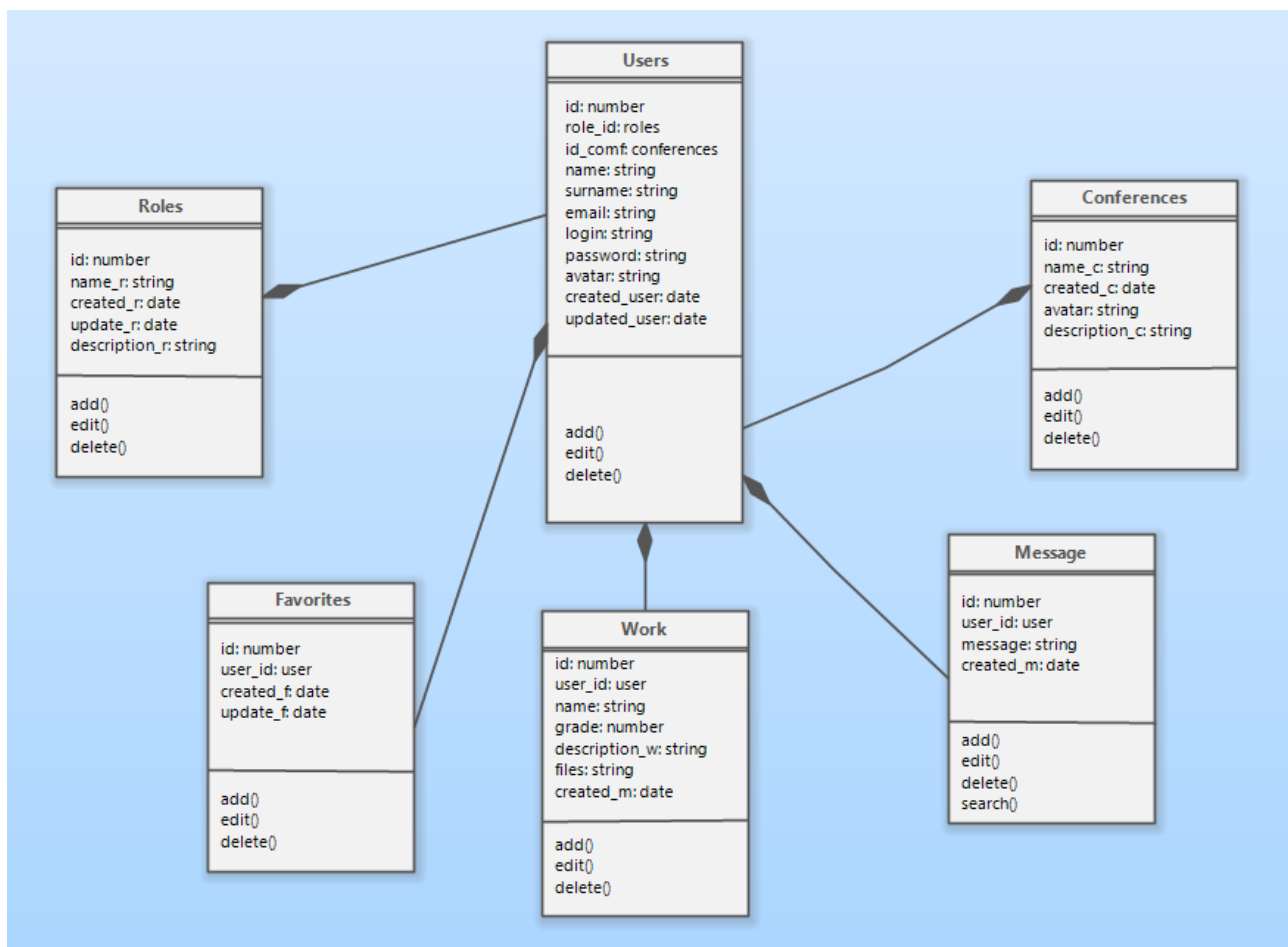


Рисунок 3.1 – Діаграма класів

3.2.1 Опис класу Roles

Roles – забезпечує зберігання основної інформації про ролі доступу користувачів.

Атрибути:

- id – унікальний ідентифікатор ролі;
- name_r – назва ролі;
- created_r – дата створення ролі;
- update_r – дата останнього редагування ролі;
- description_r – опис ролі.

Операції:

- add() – додавання нової ролі;
- edit() – редагування ролі;
- delete() – видалення ролі;
- list_roles() – виведення списку ролей

3.2.2 Опис класу Users

Users – забезпечує зберігання основної інформації про користувачів.

Атрибути:

- id – унікальний ідентифікатор користувача;
- role_id – ідентифікатор ролі;
- id_comf – ідентифікатор конференції;
- name – ім'я користувача;
- surname – прізвище користувача;
- login – логін користувача;
- password – пароль користувача;
- avatar – шлях до фото профіля користувача;
- created_user – дата створення користувача;
- update_user – дата останнього редагування користувача;
- state – статус.

Операції:

- add() – додавання нового користувача;
- edit() – редагування інформації користувача;
- delete() – видалення користувача;
- get_users() – виведення списку користувачів;
- get_info() – виведення інформації про користувача.

3.2.3 Опис класу Conferences

Conferences – забезпечує зберігання основної інформації про конференції користувачів.

Атрибути:

- id – унікальний ідентифікатор конференції;
- name_c – назва конференції;
- created_c – дата створення конференції;
- avatar – картинка конференції;
- description_c – опис конференції.

Операції:

- add() – додавання нової конференції;
- edit() – редагування конференції;
- delete() – видалення конференції;
- list_conf() – виведення списку конференцій.

3.2.4 Опис класу Message

Message – забезпечує зберігання повідомлення користувачів.

Атрибути:

- id – унікальний ідентифікатор конференції;
- user_id – ідентифікатор користувача;
- message – вміст повідомлення;
- created_m – дата відправлення повідомлення.

Операції:

- add() – додавання нового повідомлення;
- edit() – редагування повідомлення;
- delete() – видалення повідомлення;
- display_list_mes() – виведення повідомлень;
- search() – пошук повідомлення.

3.2.5 Опис класу Work

Work – забезпечує зберігання завдань користувачів.

Атрибути:

- id – унікальний ідентифікатор завдання;

- `user_id` – ідентифікатор користувача;
- `name` – назва завдання;
- `grade` – оцінка за завдання;
- `description_w` – суть завдання;
- `files: string` – шлях до завантаженої відповіді на завдання;
- `created_w` – дата створення завдання.

Операції:

- `add()` – додавання нового завдання;
- `edit()` – редагування завдання;
- `delete()` – видалення завдання;
- `display_list_mes()` – виведення завдань.

3.2.6 Опис класу Favorites

`Favorites` – забезпечує зберігання основної інформації про друзів користувача.

Атрибути:

- `id` – унікальний ідентифікатор друга;
- `user_id` – ідентифікатор користувача;
- `created_f` – дата додання друга до списку.

Операції:

- `add()` – додавання нового друга до списку;
- `delete()` – видалення друга зі списку;
- `list_fav()` – виведення списку друзів.

3.3 Сценарії використання та діаграма прецедентів

Діаграма прецедентів – це діаграма, яка візуально зображує різноманітні сценарії взаємодії між користувачами і випадками використання. Діаграма прецедентів також описує функціональні аспекти системи [26].



Рисунок 3.2 – Діаграма прецедентів

Для полегшення розробки застосунку використовують сценарії використання, які дозволяють сформулювати та задокументувати вимоги. Існує 3 форми сценаріїв:

- Коротка форма;
- Поверхнева форма;
- Повна форма.

Сценарій №1 (коротка форма): Пошук повідомлень в чаті

Користувач переходить в чат. Натискає на кнопку «пошук». Система виводить поле для введення пошукового запиту. Користувач написав в пошукове поле запит та натиснув «Enter». Система здійснює пошук в бд за даним запитом (за ключовим словом). Користувач отримує список з результатом виконаного запиту.

Сценарій №2 (поверхнева форма): Демонстрація екрана

Головний сценарій:

- Користувач авторизується в системі.
- Система переходить на сторінку зі списком доступних чатів користувача.
- Користувач переходить в чат.
- Система виводить список повідомлень.
- Користувач натискає на кнопку «демонстрація екрана».
- Система пропонує обрати потрібну поверхню захоплення робочого столу (Весь екран, вікно застосунку, вкладка браузера).
- Користувач обирає потрібну поверхню та натискає на кнопку «захоплення екрану».
- Система стартує стрім контент екрану з поверхні захоплення, яку обрав користувач.

Альтернативні сценарії:

- Розрив зв'язку з інтернетом, виводиться відповідне повідомлення.
- Користувач вирішив завершити демонстрацію екрану.

Сценарій №3: Авторизація

Таблиця 3.1 – Повний сценарій авторизації

Primary Actor	Користувач
Scope	Система дистанційного навчання
Level	Sub-function

Кінець таблиці 3.1

Preconditions	Користувач не авторизований
Stakeholders and interest	Користувач – авторизація надає: <ul style="list-style-type: none"> - Спілкування за допомогою чату. - Демонстрація екрану. - Тд. Вчитель – полегшення викладання навчальних матеріалів.
Main Success Scenario:	<ul style="list-style-type: none"> - Користувач вводить email. - Користувач вводить пароль. - Система перевіряє дані. - Користувач авторизується в системі.
Result	Користувач авторизувався в системі.
1a	Система не може підключитися до бд. Система виводить відповідне повідомлення.
Result	Користувач не авторизований.
2a	Користувач зробив помилку при введенні email або паролю.
Result	Користувач не авторизований.
Special Requirements	Гнучкість.
Technology and Data Variations List	Збереження email та паролю для того, щоб кожного разу не вводити їх знову.
Frequency of Occurrence	Система працюватиме майже безперервно.

Сценарій 4 – Реєстрація.

Таблиця 3.2 – Сценарій реєстрації

Діючі особи	Користувач, система.
Мета	Створення облікового запису.

Кінець таблиці 3.2

Передумова	Користувач не авторизований в системі.
Успішний сценарій: <ol style="list-style-type: none"> 1) Користувач переходить до форми реєстрації. 2) Проводить заповнення контактних даних. 3) Система проводить перевірку введених даних. 4) Система зберігає обліковий запис в базі даних. 5) Користувач отримує повідомлення про успішну реєстрацію та перенаправляється на головну сторінку. 	
Результат	Новий обліковий запис збережено в базі даних.

Сценарій 5 – Пошук повідомлення в чаті.

Таблиця 3.3 – Сценарій пошуку повідомлення в чаті

Діючі особи	Зареєстрований користувач, система
Мета	Пошук повідомлення в чаті.
Успішний сценарій: <ol style="list-style-type: none"> 1. Користувач натиснув на кнопку «пошук». 2. Система вивела поле для введення пошукового запиту. 3. Користувач написав в пошукове поле запит та натиснув «Enter». 4. Система виводить результати запиту. 	
Результат	Користувач отримав повідомлення, які відповідають запиту.

Сценарій №6: Додавання користувача до списку друзів

Таблиця 3.4 – Сценарій додавання користувача до списку друзів

Діючі особи	Авторизований користувач, система.
Мета	Додати користувача до списку друзів.
Передумова	Користувач авторизований в системі.

Кінець таблиці 3.4

Успішний сценарій:	
1) Користувач обирає профіль друга. 2) Користувач додає профіль в список. 3) Система зберігає дані про список друзів користувача.	
Результат	Користувач додав користувача до списку.

Сценарій №7: Написання повідомлення

Таблиця 3.5 – Сценарій написання повідомлення

Діючі особи	Авторизований користувач, система.
Мета	Написати повідомлення.
Передумова	Користувач авторизований в системі.
Успішний сценарій:	
1) Користувач обирає конференцію. 2) Користувач пише текст повідомлення. 3) Натискає кнопку «Відправити». 4) Система записує повідомлення до бд.	
Результат	Користувач написав повідомлення у чаті.

3.4 Діаграма розгортання

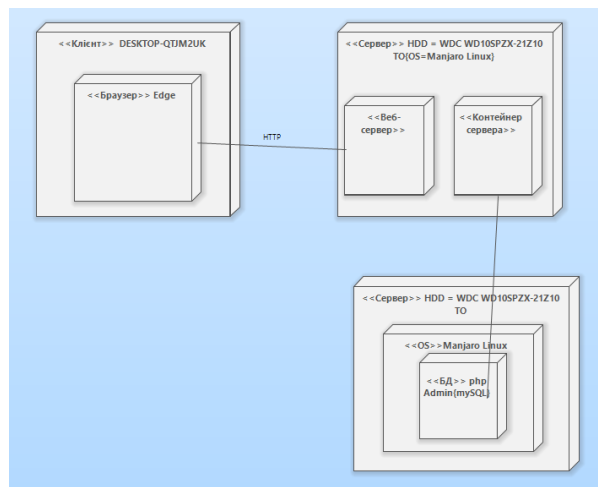


Рисунок 3.3 – Діаграма розгортання

За допомогою діаграми розгортання зазвичай візуалізують взаємодію апаратної частини та програмної частини системи.

Діаграмами розгортання називаються діаграмами на яких відображаються обчислювальні вузли, компоненти та об'єкти, що використовуються на цих вузлах. Діаграми допомагають відобразити основні складові системи, такі як апаратне забезпечення, вузли учасників, їх розосередження і те, як вони взаємопов'язані. Діаграми розгортання можна вважати апаратними компонентами, де присутні складові забезпечення.

На відміну від інших UML-діаграм, які більше описують логічні компоненти системи діаграми розгортання допомагають моделювати апаратну топологію системи.

На рисунку 3.3 продемонстрована діаграма розгортання проекту кваліфікаційної роботи.

Висновки до розділу 3

В третьому розділі описано використані технології при розробці програмного забезпечення. Для кращого зрозуміння роботи системи було розроблено UML діаграму прецедентів у якій зображено взаємодію вебзастосунок з користувачем та сценарії його використання. Також було представлено діаграму розгортання, що дозволяє з розумом організувати компоненти, від яких залежить швидкість та безпека системи. Вебзастосунок буде відповідати описаному у третьому розділі.

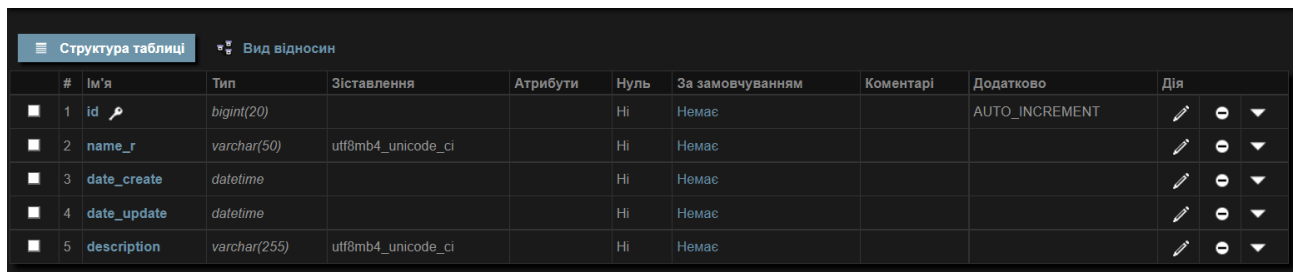
4. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розробка бази даних

Повний лістинг коду створення бази даних винесено до Додатку А. Кожна сутність представлена таблицею.

Таблиця `roles` містить у собі наступні поля:

- `id` – тип даних: `bigint(20)`, автоінкрементоване поле;
- `name_r` – тип даних: `varchar(50)`;
- `date_create` – тип даних: `timestamp`;
- `date_update` – тип даних: `timestamp`.

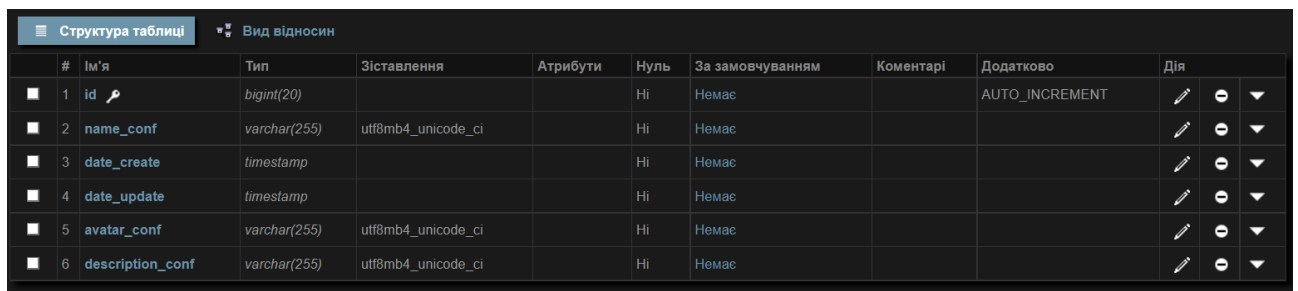


#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	<code>id</code>	<code>bigint(20)</code>			Ні	Немає		AUTO_INCREMENT	✎ ⌵
2	<code>name_r</code>	<code>varchar(50)</code>	<code>utf8mb4_unicode_ci</code>		Ні	Немає			✎ ⌵
3	<code>date_create</code>	<code>datetime</code>			Ні	Немає			✎ ⌵
4	<code>date_update</code>	<code>datetime</code>			Ні	Немає			✎ ⌵
5	<code>description</code>	<code>varchar(255)</code>	<code>utf8mb4_unicode_ci</code>		Ні	Немає			✎ ⌵

Рисунок 4.1 – Структура таблиці `roles`

Таблиця `confs` містить у собі наступні поля:

- `id` – тип даних: `bigint(20)`, автоінкрементоване поле;
- `name_conf` – тип даних: `varchar(50)`;
- `date_create` – тип даних: `timestamp`;
- `date_update` – тип даних: `timestamp`;
- `avatar_conf` – тип даних: `varchar(255)`;
- `description_conf` – тип даних: `varchar(255)`.

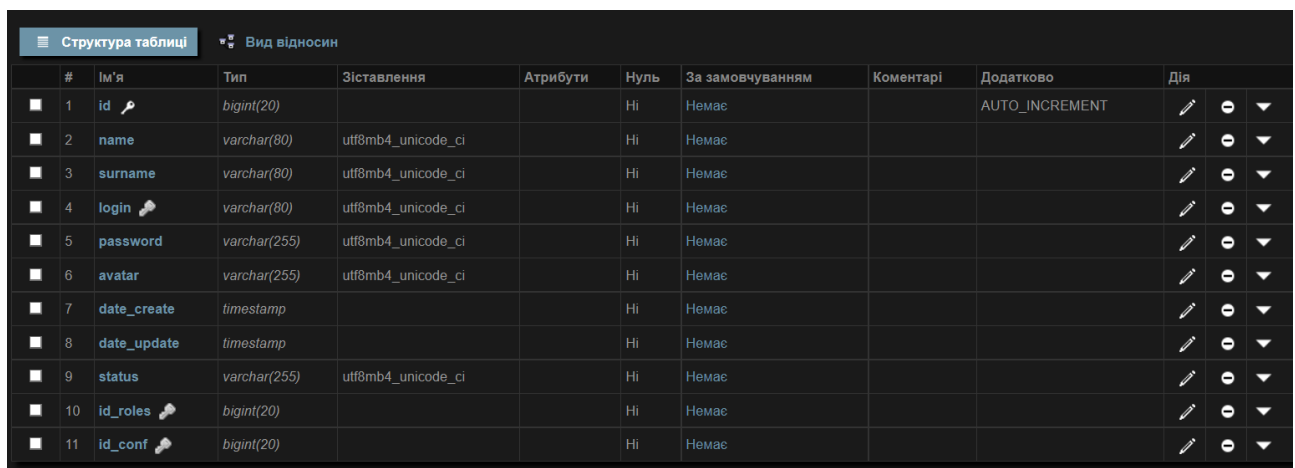


#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	<code>id</code>	<code>bigint(20)</code>			Ні	Немає		AUTO_INCREMENT	✎ ⌵
2	<code>name_conf</code>	<code>varchar(255)</code>	<code>utf8mb4_unicode_ci</code>		Ні	Немає			✎ ⌵
3	<code>date_create</code>	<code>timestamp</code>			Ні	Немає			✎ ⌵
4	<code>date_update</code>	<code>timestamp</code>			Ні	Немає			✎ ⌵
5	<code>avatar_conf</code>	<code>varchar(255)</code>	<code>utf8mb4_unicode_ci</code>		Ні	Немає			✎ ⌵
6	<code>description_conf</code>	<code>varchar(255)</code>	<code>utf8mb4_unicode_ci</code>		Ні	Немає			✎ ⌵

Рисунок 4.2 – Структура таблиці `confs`

Таблиця users містить у собі наступні поля:

- id – тип даних: bigint(20), автоінкрементоване поле;
- name – тип даних: varchar(80);
- surname – тип даних: varchar(80);
- login – тип даних: varchar(80);
- password – тип даних: varchar(255);
- avatar_conf – тип даних: varchar(255);
- date_create – тип даних: timestamp;
- date_update – тип даних: timestamp;
- status – тип даних: varchar(255);
- id_roles – тип даних: bigint(20);
- id_conf – тип даних: bigint(20).



#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint(20)			Ні	Немає		AUTO_INCREMENT	✎ ⌵
2	name	varchar(80)	utf8mb4_unicode_ci		Ні	Немає			✎ ⌵
3	surname	varchar(80)	utf8mb4_unicode_ci		Ні	Немає			✎ ⌵
4	login	varchar(80)	utf8mb4_unicode_ci		Ні	Немає			✎ ⌵
5	password	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			✎ ⌵
6	avatar	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			✎ ⌵
7	date_create	timestamp			Ні	Немає			✎ ⌵
8	date_update	timestamp			Ні	Немає			✎ ⌵
9	status	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			✎ ⌵
10	id_roles	bigint(20)			Ні	Немає			✎ ⌵
11	id_conf	bigint(20)			Ні	Немає			✎ ⌵

Рисунок 4.3 – Структура таблиці users

Таблиця messages містить у собі наступні поля:

- id – тип даних: bigint(20), автоінкрементоване поле;
- message_text – тип даних: text(5000);
- date_create – тип даних: timestamp;
- date_update – тип даних: timestamp;
- id_user – тип даних: bigint(20);
- id_conf – тип даних: bigint(20).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint(20)			Ні	Немає		AUTO_INCREMENT	✎ ⌵
2	message_text	text	utf8mb4_unicode_ci		Ні	Немає			✎ ⌵
3	date_create	timestamp			Ні	Немає			✎ ⌵
4	date_update	timestamp			Ні	Немає			✎ ⌵
5	id_user	bigint(20)			Ні	Немає			✎ ⌵
6	id_conf	bigint(20)			Ні	Немає			✎ ⌵

Рисунок 4.4 – Структура таблиці messages

Таблиця friends містить у собі наступні поля:

- id – тип даних: bigint(20), автоінкрементоване поле;
- message_text – тип даних: text(5000);
- date_invite – тип даних: timestamp;
- id_user – тип даних: bigint(20);
- id_friend – тип даних: bigint(20).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint(20)			Ні	Немає		AUTO_INCREMENT	✎ ⌵
2	id_user	bigint(20)			Ні	Немає			✎ ⌵
3	id_friend	bigint(20)			Ні	Немає			✎ ⌵
4	date_invite	timestamp			Ні	Немає			✎ ⌵

Рисунок 4.5 – Структура таблиці friends

Таблиця task містить у собі наступні поля:

- id – тип даних: bigint(20), автоінкрементоване поле;
- name_task – тип даних: varchar(255);
- description_task – тип даних: text(5000);
- date_create – тип даних: timestamp;
- path_document – тип даних: varchar(255);
- grade – тип даних: int(11);
- id_user – тип даних: bigint(20).

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint(20)			Ні	Немає		AUTO_INCREMENT	✎ ⌵
2	name_task	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			✎ ⌵
3	description_task	text	utf8mb4_unicode_ci		Ні	Немає			✎ ⌵
4	date_creation	timestamp			Ні	Немає			✎ ⌵
5	path_document	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			✎ ⌵
6	grade	int(11)			Ні	Немає			✎ ⌵
7	id_user	bigint(20)			Ні	Немає			✎ ⌵

Рисунок 4.6 – Структура таблиці friends

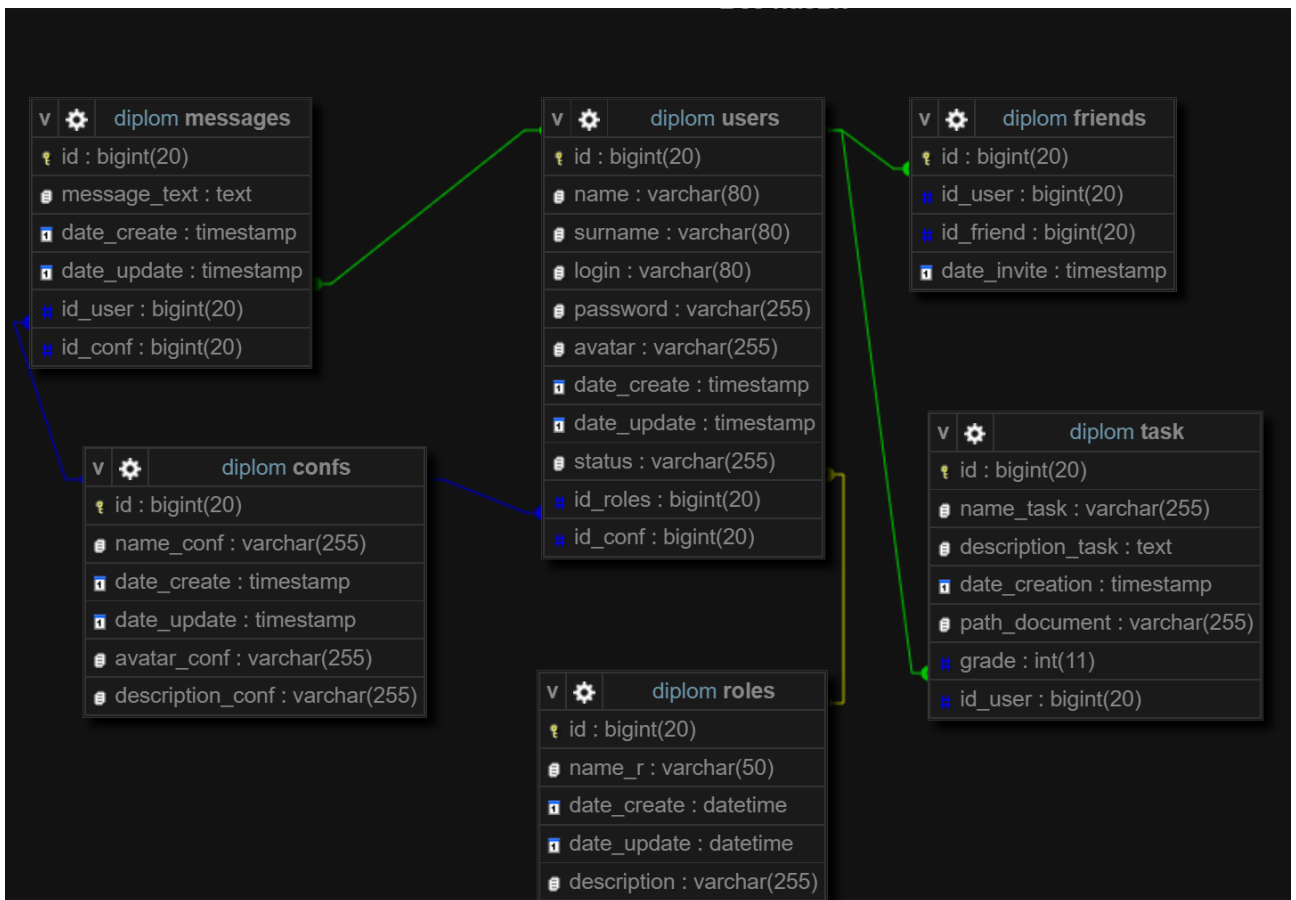


Рисунок 4.6 – Структура бази даних

4.2 Тестування

Таблиця 4.1– Тестування авторизації

Діючі особи	Користувач, система.
Мета	Авторизація у системі.
Передумова	Користувач не авторизований в системі.

Кінець таблиці 4.1

Успішний сценарій:	
<ul style="list-style-type: none"> – Користувач переходить до форми авторизації. – Проводить заповнення контактних даних. – Система проводить перевірку введених даних. – Система знаходить обліковий запис в базі даних. – Користувач отримує повідомлення про успішну авторизацію та перенаправляється на головну сторінку. 	
Сценарій успішний. Користувач авторизувався.	
Розширення	
1a	Відсутній зв'язок з базою даних. Система демонструє відповідне повідомлення. Результат: користувач не може авторизуватися.
2a	Користувач не заповнив одне з полів. Система демонструє відповідне повідомлення. Результат: користувач не може авторизуватися.
3a	Відсутній зв'язок з інтернетом. Система демонструє відповідне повідомлення. Результат: користувач не може авторизуватися.
4a	Користувача не існує. Система демонструє відповідне повідомлення. Результат: користувач не може авторизуватися.
Усі розширення успішно виконані.	

Таблиця 4.2– Тестування реєстрації

Діючі особи	Користувач, система.
Мета	Створення облікового запису.
Передумова	Користувач не авторизований в системі.

Кінець таблиці 4.2

Успішний сценарій:	
<ul style="list-style-type: none"> – Користувач переходить до форми реєстрації. – Проводить заповнення контактних даних. – Система проводить перевірку введених даних. – Система зберігає обліковий запис в базі даних. – Користувач отримує повідомлення про успішну реєстрацію та перенаправляється на головну сторінку. 	
Сценарій успішний. Збережено новий обліковий запис	
Розширення	
1a	Відсутній зв'язок з базою даних. Система демонструє відповідне повідомлення. Результат: обліковий запис користувача не зареєстровано.
2a	Користувач не заповнив одне з полів. Система демонструє відповідне повідомлення. Результат: обліковий запис користувача не зареєстровано.
3a	Відсутній зв'язок з інтернетом. Система демонструє відповідне повідомлення. Результат: обліковий запис користувача не зареєстровано
Усі розширення успішно виконані.	

Таблиця 4.3 – Тестування пошуку повідомлення

Діючі особи	Зареєстрований користувач, система
Мета	Пошук повідомлення в чаті.
Успішний сценарій:	
<ul style="list-style-type: none"> – Користувач натиснув на кнопку «пошук». 	

- Система вивела поле для введення пошукового запиту.

Кінець таблиці 4.3

- Користувач написав в пошукове поле запит та натиснув «Enter».
- Система виводить результати запиту.

Сценарій успішний. Користувач отримав повідомлення, які відповідають запиту.

Розширення

1a	Відсутній зв'язок з базою даних. Система демонструє відповідне повідомлення. Результат: користувач не може знайти повідомлення.
2a	Такого повідомлення немає. Система демонструє відповідне повідомлення. Результат: користувач не може знайти повідомлення.
3a	Відсутній зв'язок з інтернетом. Система демонструє відповідне повідомлення. Результат: користувач не може знайти повідомлення.

Усі розширення успішно виконані.

Таблиця 4.4 – Тестування демонстрації екрану

Діючі особи	Зареєстрований користувач, система
Мета	Демонстрація екрану.
Успішний сценарій:	
<ul style="list-style-type: none"> – Користувач авторизується в системі. – Система переходить на сторінку зі списком доступних чатів користувача. – Користувач переходить в чат. – Система виводить список повідомлень. – Користувач натискає на кнопку «демонстрація екрана». 	

- Система пропонує обрати потрібну поверхню захоплення робочого столу (Весь екран, вікно застосунку, вкладка браузера).

Кінець таблиці 4.4

<ul style="list-style-type: none"> – Користувач обирає потрібну поверхню та натискає на кнопку «захоплення екрану». – Система стартує стрім контент екрану з поверхні захоплення, яку обрав користувач. 	
Сценарій успішний. Користувач демонструє контент екрану з поверхні захоплення, яку обрав.	
Розширення	
1a	Відсутній зв'язок з інтернетом. Система демонструє відповідне повідомлення. Результат: користувач не може скористуватися цією функцією.
Усі розширення успішно виконані.	

Таблиця 4.5 – Тестування відправлення повідомлення

Діючі особи	Авторизований користувач, система
Передумова	Користувач авторизований у системі
Мета	Відправити повідомлення.
Успішний сценарій: <ul style="list-style-type: none"> – Користувач обирає конференцію. – Користувач пише текст повідомлення. – Натискає кнопку «Відправити». – Система записує повідомлення до бд. 	
Сценарій успішний. Користувач відправив повідомлення.	
Розширення	
1a	Відсутній зв'язок з інтернетом. Система демонструє

	відповідне повідомлення. Результат: користувач не може відправити повідомлення.
--	---

Кінець таблиці 4.5

2а	Відсутній зв'язок з базою даних. Система демонструє відповідне повідомлення. Результат: користувач не може відправити повідомлення.
Усі розширення успішно виконані.	

4.3 Керівництво користувача

Під час розробки застосунку було розроблено функції, які допомагають вирішити поставлені задачі. Знизу описані найголовніші функції, які реалізовані в застосунку.

4.3.1 Авторизація та реєстрація

Неавторизований користувач, в першу чергу, потрапляє на сторінку з формою авторизації. Якщо користувач вже зареєстрований у системі, тоді він вводить електронну пошту та пароль, після чого натискає кнопку «Увійти». Якщо користувач ще незареєстрований у системі, або просто забув пароль, тоді можна натиснути відповідну кнопку «Відновити пароль» чи «Реєстрація».

Авторизація має найголовнішу роль в захисті даних від третіх осіб та головним запобіжником для збереження своєї конфіденційної інформації. Також треба робити акцент на зручність і захищеність управління сесіями та авторизацією з чим і допоможе JSON Web Token.

JSON Web Token – це стандарт передачі пакетів даних між користувачами в інтернеті. Він слугує мостом, який шифрує пакети даних між користувачами та сервером [28].

Під час виконання авторизації створюється токен, який містить у своєму тілі збережену інформацію про аккаунт користувача, а саме:

- ідентифікатор;

- адресу електронної пошти;
- пароль;

Токен під час цього зберігається у local storage і додає в заголовку кожного запиту по ключу «authorization». Сервер також виконує перевірку валідності токена для захищених маршрутів від сторонніх користувачів.

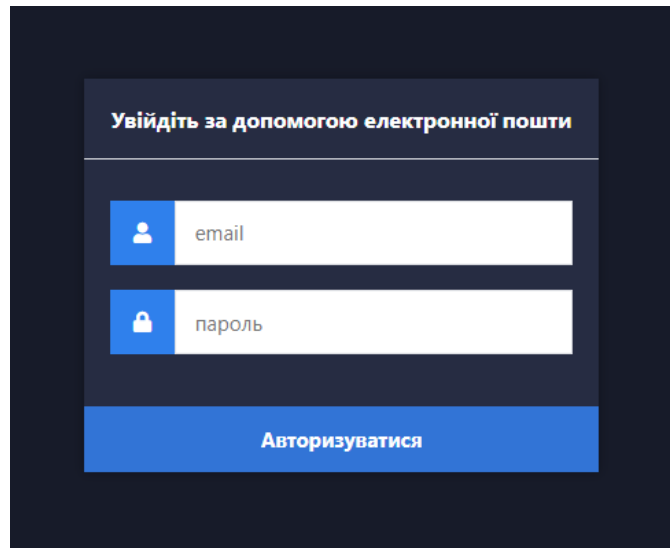


Рисунок 4.7 – Демонстрація роботи форми логін

Токен складається з трьох наступних складових [28]:

- заголовок. Заголовок являє собою код, який містить інформацію про тип і про алгоритм шифрування. Представлений в форматі base64-коду;
- тіло токена. Тіло токена містить ідентифікатор і корисну інформацію про користувача. Він також представлений в форматі base64-коду;
- цифровий підпис. Ця частина містить ключ для шифрування всього токена.

На сторінці реєстрації надається можливість вигадати логін та пароль, обов'язково ввести ФІО з електронною поштою та обрати тип аккаунту користувача, після чого, підтвердити згоду на обробку персональних даних. Якщо користувачем дані були введені правильно, то користувач перейде до головної сторінки. Вибір типу користувача впливає на подальше надання

функціоналу, який отримує користувач під час використання програмного забезпечення.

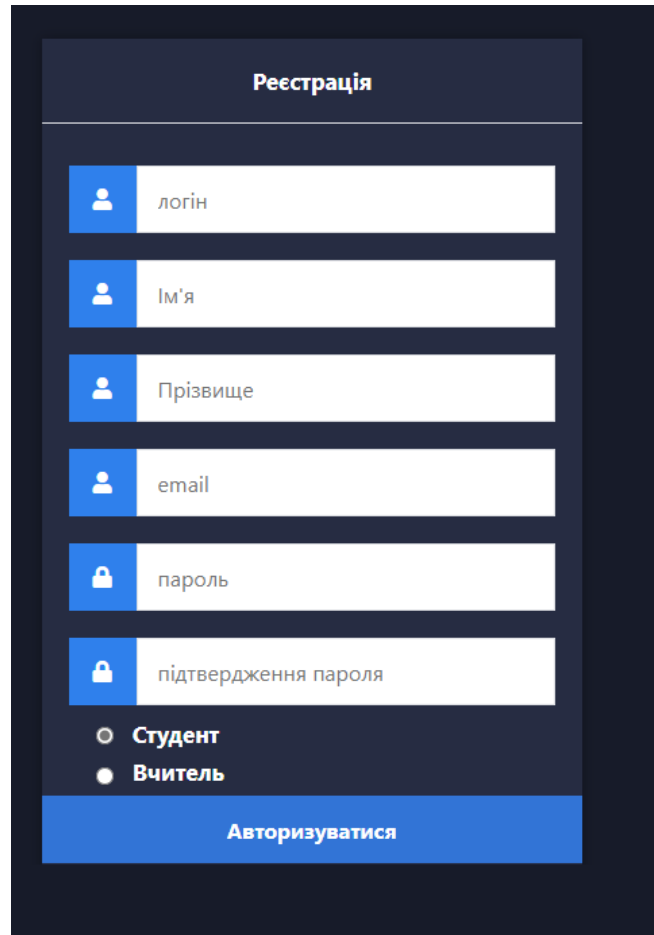


Рисунок 4.8 – Демонстрація роботи форми реєстрація

Для викладачів він має більшу кількість функціоналу, наприклад:

- створення конференцій;
- створення спеціальних форм з завданнями для подальшого завантаження відповідей користувачів;
- надання ролей доступу іншим користувачам;
- управління конференціями.

4.3.2 Конференції та чати

В конференції користувач має змогу транслювати та вимикати трансляцію відео та аудіо потоків. Також користувач має можливість надсилати повідомлення до чату конференції та додавати людей до конференції.

Аудіо та відео потоки захоплюються за допомогою

navigator.mediaDevices. getUserMedia(MediaOptions) для захоплення контенту з камери та navigator.mediaDevices.getDisplayMedia(MediaOptions) для захоплення контенту з екрану. Для онлайн конференції та чату головним фактором є якість та швидкість, тому для задоволення потреб підходять WebSockets та Peer Connection. AJAX запити не підходять для реалізації мультимедійної потокової передачі даних, тому що вони повільні, (вони навіть не є подіями) тому ніяк неможливо дізнатися чи прийшло нове повідомлення, якщо клієнт не відправить на сервер запит для оновлення даних. WebSockets вирішують цю проблему та дозволяють реалізувати потокову передачу даних та повідомлень з мінімальною затримкою за допомогою зведеного з'єднання між клієнтською частиною та сервером. Зведеное з'єднання працює без затримок, тому клієнт може отримувати мультимедійний потік та повідомлення коли завгодно, що допомагає уникнути зайвих HTTP запитів.

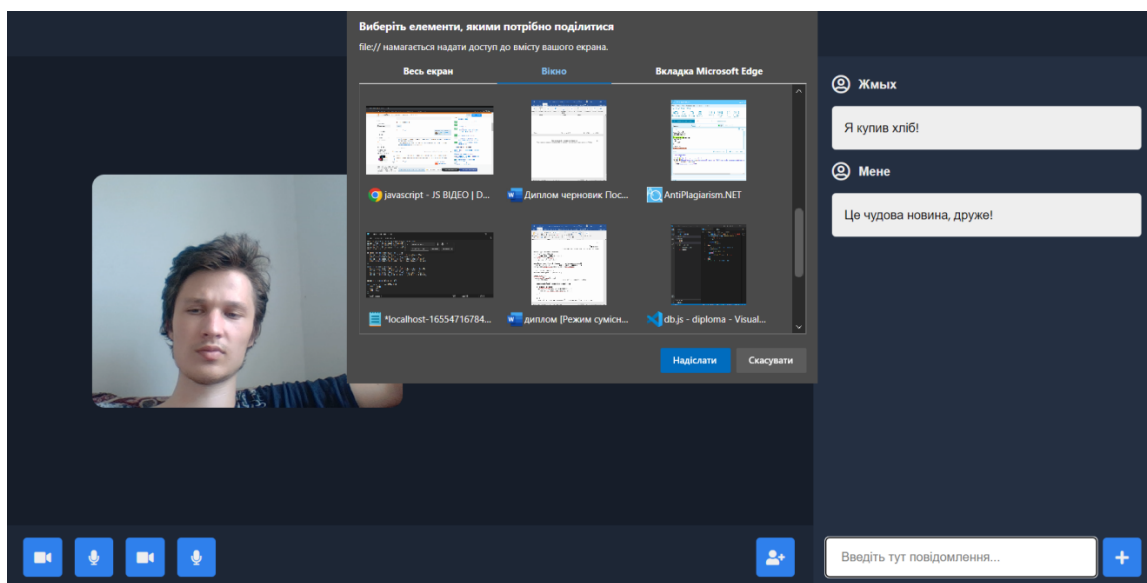


Рисунок 4.9 – Демонстрація роботи Screen Capture та чату

На боці сервера використана бібліотека socket.io. Бібліотека слугує для відправки/отримання оновлень у реальному часі та має менший функціонал ніж AJAX, тому при використанні WebSockets вважається, що користувач вже був авторизований у системі.

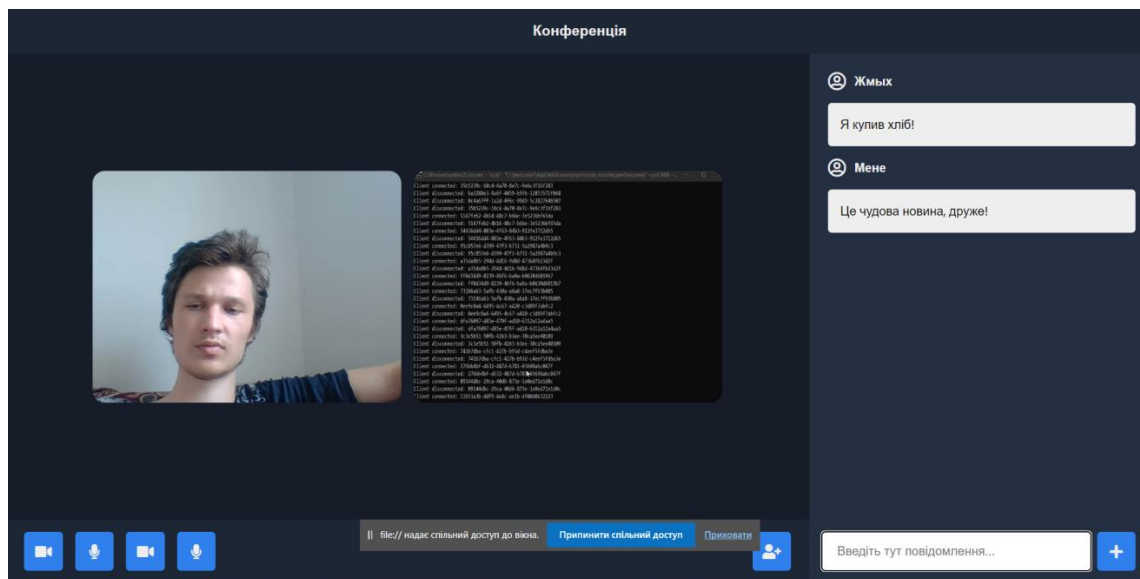


Рисунок 4.10 – Демонстрація роботи Screen Capture та чату

Для реалізації чату у конференції існують кімнати. Відправка даних відбувається за допомогою функції `io.in('id_room').emit('event', 'data')`.

Висновки до розділу 4

В четвертому розділі описано створення бази даних за допомогою СКБД MySQL, яка містить у собі 6 таблиць (users, messages, roles, tasks, confs, friends). Розглянуто та продемонстровано роботу найважливіших функцій вебзастосунку. Виконано тестування сайту.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра було виконано поставлені завдання та на їх основі сформовано звіт, який має у собі систематизацію отриманих результатів.

Проведено аналіз процесу дистанційного навчання, розглянуто існуючі аналоги та представлено їх основні недоліки. Розглянуто закони України, які описують організацію освітнього процесу за допомогою дистанційного навчання. В результаті аналізу було визначено основні переваги та недоліки проаналізованих систем. Визначено, що деякі вебзастосунки є комерційними розробками, тому вони не розповсюджуються під ліцензією opensource, через що майже неможливо здобути більш детальнішу інформацію. Сформовані основні вимоги щодо створення програмного забезпечення.

Описано моделювання програмного забезпечення для дистанційного навчання з використанням системи Screen Capture.

Представлено логічну модель за допомогою діаграми «сутність-зв'язок», яка містить 6 сутностей, які пов'язані між собою зв'язками один до багатьох. Також розроблено mock-up інтерфейсу системи, як реєстрація, авторизація, конференція, чат, завдання. Вони дозволяють продемонструвати свої напрацювання не в «чорній коробці», а в зв'язці з предметами, для яких воно розробляється. Для кращого розуміння та візуалізації інтерфейсу було розроблено mock-up сторінок.

Виконано роботу з опису використаних технологій при розробці програмного забезпечення. Для кращого зрозуміння роботи системи було розроблено UML діаграму прецедентів у якій зображено взаємодію вебзастосунку з користувачем та сценарії його використання. Також було представлено діаграму розгортання, що дозволяє з розумом організувати компоненти, від яких залежить швидкість та безпека системи.

Описано створення бази даних за допомогою СКБД MySQL, яка містить у собі 6 таблиць (users, messages, roles, tasks, confs, friends). Розглянуто та продемонстровано роботу найважливіших функцій вебзастосунку. Виконана реалізація та тестування вебзастосунку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Наказ МОНУ від 25 квітня 2013 року (№ 466) «Положення про дистанційне навчання». URL: <https://zakon.rada.gov.ua/laws/show/z0703-13#Text> (дата звернення: 01.02.2022).
2. Закон України «Про освіту». URL: <http://vnz.org.ua/zakonodavstvo/110-%20zakon-ukrayiny?pro-osvitu>. (дата звернення: 01.02.2022).
3. Що таке змішане навчання? URL: <https://teach-hub.com/shcho-take-zmishane-navchannia/> (дата звернення: 02.02.2022).
4. Ткаченко Л. В., Хмельницька О. С. Особливості впровадження дистанційного навчання в освітній процес закладу вищої освіти // Педагогіка формування творчої особистості у вищій і загальноосвітній школах. 2021. Вип. 75. С.91-93.
5. Жевакіна Н. В. Технологія дистанційного навчання: сутність та особливості // Педагогіка формування творчої особистості у вищій і загальноосвітній школах. 2003. Вип. 4. С. 70.
6. Використання сервісу Zoom в умовах дистанційного навчання. URL: <https://content.hneu.edu.ua/s/Sx1eP17G-> (дата звернення: 05.03.2022).
7. Як Zoom стала найважливішою компанією в епоху коронавірусу. URL: <https://habr.com/ru/post/495258/> (дата звернення: 05.03.2022).
8. Конференции для всех - Zoom Blog. URL: <https://blog.zoom.us/ru/happy-meetings-for-everyone/> (дата звернення: 05.03.2022).
9. Zoom – зручний додаток для відеоконференцій: як працює і де завантажити програму. URL: <https://senior.ua/news/zoom--zruchniy-dodatok-dlya-videokonferency-yak-prasyu--de-zavantazhiti-programu> (дата звернення: 05.03.2022).
10. Google Meet – відеоконференції для бізнеса. URL: <https://workspace.google.com/intl/ru/products/meet/> (дата звернення: 06.03.2022).
11. Переваги та недоліки Google Meet. URL: <https://paradacreativa.es/uk/ventajas-y-desventajas-de-google-meet/> (дата звернення: 06.03.2022).

12. Google Meet (раніше Hangouts Meet) – безкоштовні відеозустрічі
<https://apps.google.com/intl/ru/meet/> (дата звернення: 06.03.2022).

13. Модель «Сутність-зв'язок». URL: https://stud.com.ua/154733/informatika/model_sutnist_zvyazok.com.ua/154733/informatika/model_sutnist_zvyazok (дата звернення: 08.03.2022).

14. Концепція ER-моделі. Поняття сутності (entity). Атрибути. Види атрибутів. URL: <https://www.bestprog.net/ru/2019/01/24/the-concept-of-er-model-the-concept-of-essence-and-communication-attributes-attribute-types-ru/#q05> (дата звернення: 07.03.2022).

15 Step by step: Модель сутність-зв'язок. URL: <https://step.org.ua/konspekt/base/tema2> (дата звернення: 08.03.2022).

16. Основні поняття моделі entity-relationship (сутність зв'язку).
https://stud.com.ua/97322/informatika/osnovni_ponyattya_modeli_entity_relationship_sutnist_zvyazku (дата звернення: 08.03.2022).

17. Що таке мокап і навіщо вони дизайнеру. URL: <https://wezom.academy.ua/chto-takoe-mokapy-i-zachem-oni-dizajneru/> (дата звернення: 08.03.2022).

18. У чому переваги Node.js? URL: <https://artjoker.ua/ru/blog/v-chem-preimushchestva-nodejs/> (дата звернення: 10.03.2022).

19. Використання інтерфейсу Screen Capture API. URL: https://developer.mozilla.org/ru/docs/Web/API/Screen_Capture_API/Using_Screen_Capture (дата звернення: 10.03.2022).

20. Що таке WebRTC і як це відключити. Як організувати WebRTC онлайн трансляцію за допомогою веб камери та VPS сервера Web додатки з webrtc. URL: <https://crashbox.ru/installing-multiple-os/chto-takoe-webrtc-i-kak-eto-otklyuchit-kak-organizovat-webrtc-onlain-translyaciyu-s/> (дата звернення: 10.03.2022).

21. WebRTC native API. URL: <http://www.webrtc.org/reference/native-apis> (дата звернення: 07.03.2022).

22. Основи: що потрібно знати про WebRTC. URL: <https://www.freeconference.com/uk/blog/the-basics-what-you-need-to-know-about-webrtc/> (дата звернення: 10.03.2022).
23. Що таке MySQL? URL: <https://freehost.com.ua/faq/wiki/что-такое-mysql/> (дата звернення: 06.03.2022).
24. Що таке база даних MySQL? URL: <https://uk.education-wiki.com/2686691-what-is-mysql-database> (дата звернення: 06.03.2022).
25. Що таке діаграма класів? URL: <https://uk.theastrologypage.com/class-diagram> (дата звернення: 08.03.2022).
26. Use case diagrams. URL: <https://www.quality-assurance-group.com/use-case-diagrams/> (дата звернення: 07.03.2022).
27. Діаграма розгортання і правила її побудови. URL: <http://um.co.ua/9/9-2/9-29953.html> (дата звернення: 07.03.2022).
28. Загальні відомості про вебтокени JSON (JWT). URL: <https://stackabuse.com/understanding-json-web-tokens-jwt/> (дата звернення: 10.03.2022).

ДОДАТОК А

Код створення бази даних

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
```

```
START TRANSACTION;
```

```
SET time_zone = "+00:00";
```

```
CREATE TABLE `confs` (  
  `id` bigint(20) NOT NULL,  
  `name_conf` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `date_create` timestamp NOT NULL,  
  `date_update` timestamp NOT NULL,  
  `avatar_conf` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `description_conf` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE `friends` (  
  `id` bigint(20) NOT NULL,  
  `id_user` bigint(20) NOT NULL,  
  `id_friend` bigint(20) NOT NULL,  
  `date_invite` timestamp NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE `messages` (  
  `id` bigint(20) NOT NULL,  
  `message_text` text COLLATE utf8mb4_unicode_ci NOT NULL,  
  `date_create` timestamp NOT NULL,  
  `date_update` timestamp NOT NULL,  
  `id_user` bigint(20) NOT NULL,
```

```
`id_conf` bigint(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE `roles` (  
  `id` bigint(20) NOT NULL,  
  `name_r` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `date_create` datetime NOT NULL,  
  `date_update` datetime NOT NULL,  
  `description` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE `task` (  
  `id` bigint(20) NOT NULL,  
  `name_task` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `description_task` text COLLATE utf8mb4_unicode_ci NOT NULL,  
  `date_creation` timestamp NOT NULL,  
  `path_document` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `grade` int(11) NOT NULL,  
  `id_user` bigint(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE `users` (  
  `id` bigint(20) NOT NULL,  
  `name` varchar(80) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `surname` varchar(80) COLLATE utf8mb4_unicode_ci NOT NULL,  
  `login` varchar(80) COLLATE utf8mb4_unicode_ci NOT NULL,
```

```
`password` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,  
`avatar` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,  
`date_create` timestamp NOT NULL,  
`date_update` timestamp NOT NULL,  
`status` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,  
`id_roles` bigint(20) NOT NULL,  
`id_conf` bigint(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_unicode_ci;
```

```
ALTER TABLE `confs`  
ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `friends`  
ADD PRIMARY KEY (`id`),  
ADD KEY `id_user` (`id_user`);
```

```
ALTER TABLE `messages`  
ADD PRIMARY KEY (`id`),  
ADD KEY `id_conf` (`id_conf`),  
ADD KEY `id_user` (`id_user`);
```

```
ALTER TABLE `roles`  
ADD PRIMARY KEY (`id`);
```

```
ALTER TABLE `task`  
ADD PRIMARY KEY (`id`),  
ADD KEY `id_user` (`id_user`);
```

```
ALTER TABLE `users`  
  ADD PRIMARY KEY (`id`),  
  ADD UNIQUE KEY `login` (`login`),  
  ADD KEY `id_conf` (`id_conf`),  
  ADD KEY `id_roles` (`id_roles`);
```

```
ALTER TABLE `confs`  
  MODIFY `id` bigint(20) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `friends`  
  MODIFY `id` bigint(20) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `messages`  
  MODIFY `id` bigint(20) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `roles`  
  MODIFY `id` bigint(20) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `task`  
  MODIFY `id` bigint(20) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `users`  
  MODIFY `id` bigint(20) NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE `friends`  
  ADD CONSTRAINT `friends_ibfk_1` FOREIGN KEY (`id_user`) REFERENCES  
`users` (`id`);
```

```
ALTER TABLE `messages`
```

```
ADD CONSTRAINT `messages_ibfk_1` FOREIGN KEY (`id_conf`)
REFERENCES `confs` (`id`),
ADD CONSTRAINT `messages_ibfk_2` FOREIGN KEY (`id_user`)
REFERENCES `users` (`id`);
```

```
ALTER TABLE `task`
```

```
ADD CONSTRAINT `task_ibfk_1` FOREIGN KEY (`id_user`) REFERENCES
`users` (`id`);
```

```
ALTER TABLE `users`
```

```
ADD CONSTRAINT `users_ibfk_1` FOREIGN KEY (`id_conf`) REFERENCES
`confs` (`id`),
```

```
ADD CONSTRAINT `users_ibfk_2` FOREIGN KEY (`id_roles`) REFERENCES
`roles` (`id`);
```

```
COMMIT;
```