

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Веб сайт лікарні з використанням телемедицини»

Виконав: студентка 2 курсу, групи ІСТ-23м
спеціальності 126 – Інформаційні системи та
технології

(шифр і назва спеціальності)

Олійниченко А.Б.

(ПІБ студента)

Керівник: к.т.н., доцент кафедри АІТ

Кулик Я.А.

(науковий ступінь, вчене звання / посада, ПІБ керівника)

«13» листопада 2024 р.

Опонент: Володимир ОЗЕРАНСЬКИЙ

(науковий ступінь, вчене звання / посада, ПІБ опонента)

: «16» грудня 2024

Допущено до захисту

Завідувач кафедри АІТ

д.т.н., Олег БІСІКАЛО

(науковий ступінь, вчене звання)

«16» 12 2024 року

Вінниця ВНТУ – 2024 року

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій
Рівень вищої освіти _____
Галузь знань – _____
Спеціальність – _____
Освітньо-професійна програма – _____
зображень

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ
д.т.н., проф. Олег БІСІКАЛО
« 19 » вересня 2024р.

ЗАВДАННЯ

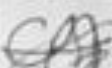
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Олійниченко Анні Богданівні

(ПІБ автора повністю)

- Тема роботи: Веб-сайт лікарні з використанням телемедицини
Керівник роботи: к.т.н., доцент каф. АІТ Кулик Я.А.
Затвердженні наказом ВНТУ від « 17 » вересня 2024 року № 310.
- Строк подання роботи студентом: до « 16 » грудня 2024 року.
- Вихідні дані до роботи: Операційна система Windows 10, JavaScript, фреймворк Vue.JS, Платформа для роботи з базами даних Firebase, обмеження на кількість записів на одного лікаря: до 24 консультацій на день (з урахуванням 20-хвилинного часу на консультацію в 8-годинному робочому дні)
- Зміст текстової частини: Вступ; Дослідження предметної галузі; Вибір технологічних засобів; Розробка програмного забезпечення; Тестування програмного забезпечення; Економічний розділ; Висновки; Список використаних джерел.

5. Перелік ілюстративного (або графічного) матеріалу: Блок-схема веб-сайту;
UML-діаграма класів; UML-діаграма прецедентів; UML-діаграма діяльності;
6. Консультанти розділів роботи

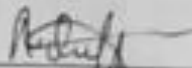
Розділ змістової частини роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1 – 3	доцент кафедри АІТ, к.т.н., Ярослав КУЛИК	18.09.24 	13.12.24 
4	завідувач кафедри ЕПВМ, професор, к.е.н. Олександр ЛЕСЬКО.	 18.09.24	 14.12.24

7. Дата видачі завдання: « 18 » вересня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження предметної галузі	01.09.24 - 10.09.24	виконано
2	Обґрунтування вибору технологічного стеку	12.10.24 - 25.10.24	виконано
3	Розробка програмного забезпечення	05.09.24 - 30.10.24	виконано
4	Тестування розробленого програмного забезпечення	01.10.24 - 12.11.24	виконано
5	Підготовка економічної частини	02.10.24 - 04.11.24	виконано
6	Оформлення пояснювальної записки, графічного матеріалу і презентації	15.10.24 - 15.11.24	виконано
7	Попередній захист роботи	до 05.12.2024	виконано
8	Захист роботи	до 20.12.2024	виконано

Студент


(підпис)

Анна ОЛІЙНИЧЕНКО
(прізвище та ініціали)

Керівник роботи


(підпис)

Ярослав КУЛИК
(прізвище та ініціали)

АНОТАЦІЯ

УДК 004.42

Олійниченко А.Б. Веб-сайт лікарні з використанням телемедицини. Магістерська кваліфікаційна робота зі спеціальності 126 – Інформаційні системи та технології, освітня програма – Інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2024. 127 с.

На укр. мові. Бібліогр.: 31 назв; рис.: 32; табл. 12.

Магістерська кваліфікаційна робота присвячена розробці веб-сайту лікарні з інтегрованими функціями телемедицини, спрямованими на розширення доступу до медичних послуг і підвищення якості взаємодії між лікарями та пацієнтами.

У даній роботі детально досліджується основні компоненти такого сайту, зокрема системи онлайн-консультацій, управління медичними записами та можливості дистанційного моніторингу стану здоров'я.

Ключові слова: телемедицина, веб-сайт лікарні, дистанційна медична допомога, онлайн-консультації, веб-розробка.

ABSTRACT

Oliinychenko A.B. Hospital website with the use of telemedicine. Master's qualification work in the specialty 126 - Information systems and technologies, educational program - Information technologies of data and image analysis. Vinnytsia: VNTU, 2024. 127 c.

In Ukrainian. Bibliography: 31 titles; Figures: 32; Table 12.

The master's thesis is devoted to the development of a hospital website with integrated telemedicine functions aimed at expanding access to medical services and improving the quality of interaction between doctors and patients.

This work examines in detail the main components of such a website, including online consultation systems, medical record management, and remote health monitoring capabilities.

Keywords: telemedicine, hospital website, remote medical care, online consultations, web development.

ЗМІСТ

ВСТУП.....	9
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ГАЛУЗІ.....	12
1.1. Огляд існуючих аналогів телемедичних веб-сайтів	12
1.1.1. Веб-сайт Mayo Clinic	13
1.1.2 Веб-сайт Teladoc Health.....	16
1.1.3 Веб-сайт Amwell	18
1.1.4 Веб-сайт DOXY.me	20
1.1.5 Локальні телемедичні рішення в Україні	23
1.2 Порівняння можливостей запропонованого рішення з існуючими аналогами	26
1.3 Можливості інтеграції телемедичних сервісів у веб-сайти.....	28
1.4 Висновки до розділу	31
2 ВИБІР ТЕХНОЛОГІЧНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ СИСТЕМИ.....	33
2.1 Вибір мов програмування для створення веб-сайту.....	34
2.1.1 HTML, CSS, JavaScript: обґрунтування вибору	35
2.1.2 Vue.js як основний фреймворк.....	39
2.2 Використання Firebase для авторизації та бази даних	41
2.3 Інтеграція відеоконференцій для телемедичних консультацій.....	43
2.4. Висновки до розділу	46
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	49
3.1 Обґрунтування та проектування функціоналу веб-сайту лікарні з використанням телемедицини	49
3.2 Проектування бази даних	51

3.3	Опис структури проекту та основних функцій	65
3.4	Тестування	73
3.5	Висновки до розділу	78
4	ЕКОНОМІЧНИЙ РОЗДІЛ	80
4.1	Проведення комерційного та технологічного аудиту науково-технічної розробки	80
4.2	Розрахунок узагальненого коефіцієнта якості розробки	84
4.3	Розрахунок витрат на проведення науково-дослідної роботи	85
4.3.1	Витрати на оплату праці.....	86
4.3.2	Відрахування на соціальні заходи	89
4.3.3	Сировина та матеріали	89
4.3.4	Розрахунок витрат на комплектуючі	91
4.3.5	Спецустаткування для наукових (експериментальних) робіт.....	91
4.3.6	Програмне забезпечення для наукових (експериментальних) робіт	92
4.3.7	Амортизація обладнання, програмних засобів та приміщень	93
4.3.8	Паливо та енергія для науково-виробничих цілей.....	94
4.3.9	Службові відрядження.....	95
4.3.10	Витрати на роботи, які виконують сторонні підприємства, установи і організації	96
4.3.11	Інші витрати.....	96
4.3.12	Накладні (загальновиробничі) витрати	97
4.4	Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором.....	98
4.5	Висновки до розділу	102
	Висновки	103

Список використаних джерел	105
Додатки.....	108
Додаток А (обов’язковий) Технічне завдання	109
Додаток Б (обов’язковий) Ілюстративна частина.....	112
Додаток В (обов’язковий) Лістинг програмного забезпечення	117
Додаток Г (обов’язковий) Протокол перевірки магістерської кваліфікаційної роботи на наявність текстових запозичень	127

ВСТУП

Швидкий розвиток інформаційних технологій створює нові можливості для впровадження цифрових рішень у різні сфери людської діяльності, включаючи медицину. У сучасному світі доступність медичних послуг є критично важливою для забезпечення належного рівня охорони здоров'я, особливо в умовах пандемій, віддаленості населених пунктів та перевантаженості медичних установ. Телемедицина, як ключова частина цифровізації медицини, сприяє наданню дистанційних медичних послуг та дозволяє значно розширити доступ до якісного медичного обслуговування.

Веб-сайти з телемедичними функціями відіграють важливу роль у цьому процесі. Вони дозволяють пацієнтам записуватися на консультації онлайн, отримувати відео- або текстові консультації від лікарів, зберігати та переглядати медичні записи, а також моніторити стан здоров'я через особистий кабінет. Для лікарів такі веб-рішення забезпечують оптимізацію робочих процесів, зменшення адміністративного навантаження та можливість дистанційної роботи з пацієнтами.

Магістерська кваліфікаційна робота присвячена дослідженню методів розробки веб-сайту з інтеграцією телемедичних функцій та безпосередньо розробці системи для лікарні, яка дозволить автоматизувати процеси взаємодії між лікарями та пацієнтами.

Актуальність. Розвиток телемедицини та її впровадження у медичні установи стало особливо важливим у сучасних умовах, коли медичні заклади потребують рішень для покращення доступності та якості надання послуг. Створення веб-сайту з функціями телемедицини дозволить вирішити такі проблеми, як фізична віддаленість пацієнтів, перевантаження лікарів, зручність доступу до медичної інформації та економія часу на взаємодію між пацієнтом і медичним персоналом.

Метою роботи є розробка веб-сайту лікарні з інтеграцією функцій

телемедицини, який забезпечить пацієнтам дистанційний доступ до медичних послуг, а лікарям — оптимізацію робочого процесу та зменшення витрат часу і адміністративного навантаження.

Об'єктом дослідження є процес розробки веб-сайту для медичних закладів з інтеграцією телемедичних функцій.

Предметом дослідження — методи та технології створення веб-сайтів із підтримкою телемедицини.

Для досягнення мети необхідно розв'язати наступні задачі:

- Дослідити предметну галузь.
- Обрати технологічні засоби для розробки системи.
- Розробити програмне забезпечення та провести тестування розробленого веб-сайту.
- Зробити розрахунок економічної частини.

Методи дослідження. У процесі роботи використано методи аналізу предметної галузі, методи проєктування програмного забезпечення, компонентний підхід до розробки інтерфейсу, а також методи тестування для перевірки працездатності веб-сайту.

Основний науково-технічний результат полягає у розробці комплексного рішення для телемедицини, яке забезпечує не тільки запис на консультації та взаємодію з лікарем, але й функцію моніторингу стану здоров'я пацієнтів через інтерактивний особистий кабінет. Вибір сучасних технологій (Vue.js, Firebase) дозволяє створити систему, яка відзначається високою продуктивністю, масштабованістю та безпекою даних.

Практичною цінністю роботи є створення програмного продукту, який може бути використаний у медичних установах для автоматизації процесів управління взаємодією з пацієнтами. Веб-сайт сприяє покращенню доступності медичних послуг, зниженню навантаження на лікарів та підвищенню якості медичного обслуговування.

Апробація результатів та публікації. За результатами даної роботи

опубліковано доповідь на Міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)» на секцію Інтелектуальних інформаційних технологій та автоматизації [1]

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ГАЛУЗІ

Швидкий розвиток інформаційних технологій суттєво трансформує різні сфери суспільного життя, включаючи медицину. В умовах сучасних викликів, таких як пандемії, віддаленість населених пунктів і перевантаженість медичних установ, виникає потреба у впровадженні інноваційних рішень для покращення доступу до медичних послуг. Одним із таких рішень є телемедицина, яка забезпечує дистанційне обслуговування пацієнтів за допомогою інформаційно-комунікаційних технологій.

Веб-сайти лікарень із інтеграцією телемедичних функцій дозволяють забезпечити якісну взаємодію між пацієнтами та лікарями. Ці платформи надають можливість запису на прийоми, проведення відео- або текстових консультацій, доступу до медичних записів і моніторингу стану здоров'я. У розділі проведено аналіз предметної галузі, включаючи існуючі рішення у сфері телемедицини, їх переваги, недоліки та перспективи розвитку [2].

Основною метою даного розділу є виявлення особливостей існуючих телемедичних платформ, оцінка їх функціональних можливостей і визначення потреб, які повинна задовольнити запропонована система. Це дозволить закласти основу для розробки веб-сайту, що відповідатиме сучасним вимогам якості, доступності та безпеки медичних послуг.

1.1. Огляд існуючих аналогів телемедичних веб-сайтів

Розвиток телемедичних платформ значно покращує доступ до медичних послуг, особливо у віддалених регіонах чи в умовах надзвичайних ситуацій, таких як пандемії. Сучасні телемедичні веб-сайти забезпечують пацієнтам можливість запису на прийом до лікаря, отримання дистанційних консультацій, доступ до електронних медичних записів і навіть функції моніторингу стану

здоров'я. У цьому підрозділі розглянемо найпопулярніші аналоги телемедичних платформ, які стали зразками високоякісного обслуговування у сфері цифрової медицини.

Серед найпопулярніших телемедичних веб-сайтів варто виділити наступні:

- Mayo Clinic. Веб-сайт провідного медичного закладу США, який надає широкий спектр послуг, включаючи відеоконсультації, доступ до результатів аналізів та рекомендацій лікаря.
- Teladoc Health. Глобальна платформа, що забезпечує консультації з лікарями різних спеціалізацій у форматі відеозв'язку або текстових чатів.
- Amwell. Платформа для відеоконсультацій, яка інтегрується з електронними медичними записами (EHR) та підтримує телемедичні послуги для різних типів медичних установ.
- DOXY.me. Просте у використанні рішення для відеоконсультацій, орієнтоване на невеликі медичні заклади та індивідуальних лікарів.

1.1.1. Веб-сайт Mayo Clinic

Одним із найвідоміших прикладів телемедичних платформ є веб-сайт Mayo Clinic, який поєднує у собі сучасний дизайн, широкий функціонал і зручність використання. Mayo Clinic – це один із провідних медичних центрів у світі, який активно впроваджує інноваційні підходи для покращення доступу до медичних послуг. Веб-сайт організації забезпечує дистанційну взаємодію пацієнтів із лікарями, що є важливим елементом у сучасній охороні здоров'я [3].

Основний функціонал веб-сайту:

- Запис на прийом до лікаря. Пацієнти можуть обрати спеціаліста, зручний час прийому та тип консультації (дистанційна або очна). Система автоматично синхронізується з розкладом лікаря.

- Дистанційні консультації. Веб-сайт підтримує функцію проведення відеоконсультацій, що дозволяє лікарям надавати допомогу пацієнтам у режимі реального часу.
- Доступ до медичних записів. Пацієнти мають можливість переглядати свою історію хвороби, результати аналізів, рекомендації лікаря, що зберігаються в електронному форматі.
- Інтеграція з мобільним додатком. Для зручності користувачів веб-сайт синхронізується з мобільним додатком Mayo Clinic, що дозволяє отримувати доступ до медичних даних у будь-який час.
- Персоналізовані рекомендації. Система аналізує дані пацієнтів та надає індивідуальні рекомендації щодо профілактики та лікування захворювань. На рисунку 1.1 представлено головну сторінку проекту Exchanger з функціоналом для проведення обмінної валютної операції.

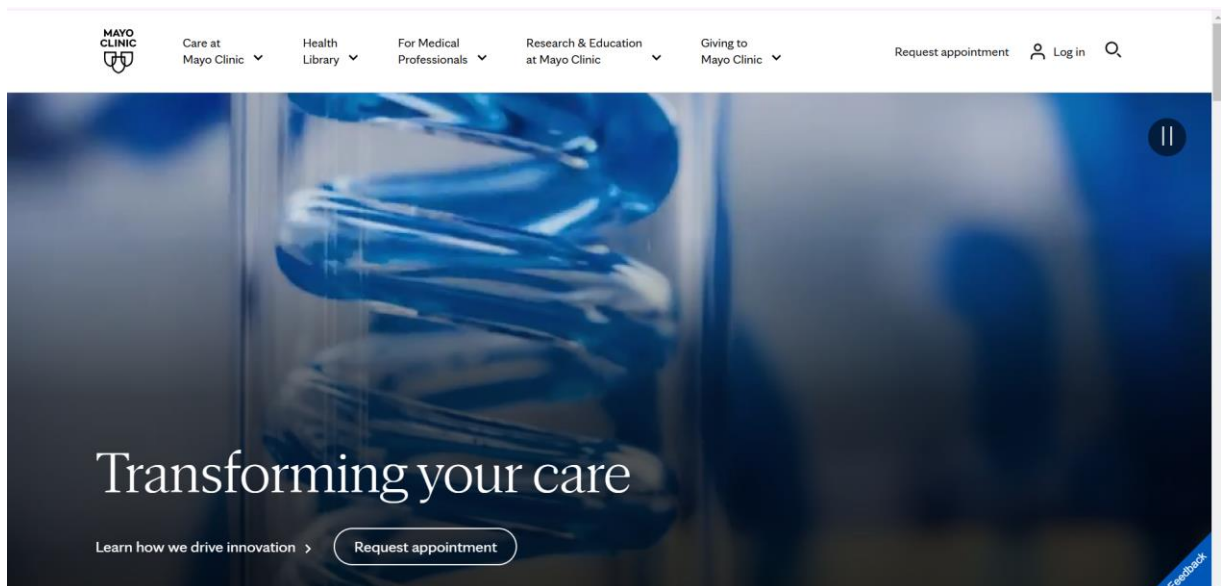


Рисунок 1.1 – Веб-сайт Mayo Clinic

Основний функціонал веб-сайту:

- Висока вартість впровадження та обслуговування. Цей фактор робить платформу недоступною для малих та середніх медичних закладів.

- Складність адаптації до локальних умов. Система розроблена з урахуванням потреб пацієнтів та лікарів США, що створює труднощі при використанні в інших країнах через різницю у нормативно-правовій базі.
- Відсутність доступу до функціоналу без підписки. Платформа вимагає значних фінансових ресурсів від пацієнтів, що обмежує її доступність.
- Складна інтеграція для малих лікарень. Малі заклади не мають змоги інтегрувати свої локальні бази даних з платформою через її закриту структуру.
- Недостатня локалізація інтерфейсу. Веб-сайт не адаптований для багатьох мов і культур, що ускладнює його використання за межами англомовних країн.
- Обмежена підтримка персоналізованих рішень. Хоча платформа надає стандартний функціонал, вона не дозволяє адаптувати модулі під специфічні потреби конкретного закладу.
- Високі вимоги до технічної інфраструктури. Для забезпечення стабільної роботи платформи потрібні високопродуктивні сервери, що збільшує витрати на обслуговування.
- Незручність у впровадженні для закладів із низьким рівнем цифровізації. Платформа потребує наявності попередньо впроваджених цифрових рішень, таких як EHR/EMR-системи.

Веб-сайт Mayo Clinic демонструє високий рівень функціональності та ефективності для великих медичних закладів. Однак його недоліки, зокрема висока вартість, складність інтеграції та обмежена адаптивність, підкреслюють необхідність створення рішень, доступних для малих і середніх лікарень.

Запропонований веб-сайт лікарні з функціями телемедицини враховує ці обмеження та пропонує більш гнучке, економічно вигідне та адаптоване рішення, яке може бути інтегроване у будь-який медичний заклад.

1.1.2 Веб-сайт Teladoc Health

Teladoc Health є однією з провідних платформ телемедицини у світі, яка пропонує широкий спектр медичних послуг у форматі онлайн. Ця платформа орієнтована як на пацієнтів, так і на медичні установи, забезпечуючи зручний доступ до лікарів різних спеціалізацій через відеозв'язок, текстові чати та телефонні консультації [4].

Основний функціонал платформи:

- Онлайн-консультації з лікарями. Пацієнти можуть звертатися до лікарів загальної практики, терапевтів, дерматологів та інших спеціалістів через відеозв'язок або текстовий чат.
- Підтримка психічного здоров'я. Платформа пропонує консультації з психологами, психіатрами та іншими спеціалістами для лікування тривожності, депресії та інших станів.
- Цілодобовий доступ до послуг. Пацієнти можуть отримати допомогу у будь-який час, що особливо важливо в умовах надзвичайних ситуацій.
- Інтеграція з роботодавцями та страховими компаніями. Платформа дозволяє корпоративним клієнтам забезпечувати медичну підтримку своїм співробітникам.

Переваги платформи:

- Широка спеціалізація послуг. Teladoc Health охоплює як фізичне, так і психічне здоров'я, надаючи комплексний підхід до медичних потреб пацієнтів.
- Міжнародна доступність. Платформа працює у багатьох країнах, що робить її універсальним інструментом для пацієнтів у різних регіонах.
- Цілодобова доступність. Пацієнти можуть звертатися до лікаря у будь-який час, що особливо важливо для невідкладних випадків.

Недоліки платформи:

- Висока вартість послуг. Для пацієнтів без медичного страхування використання Teladoc Health може бути дорогим, що обмежує її доступність для широкого загалу.
- Обмежений доступ до локальних лікарів. Платформа здебільшого орієнтована на загальні консультації та не завжди пропонує вузькоспеціалізованих лікарів у локальному регіоні.
- Складність інтеграції для малих лікарень. Малим медичним установам складно інтегрувати платформу через високі вимоги до технічної інфраструктури.
- Відсутність безкоштовних послуг. Платформа не пропонує безкоштовного базового функціоналу, що може знизити її привабливість для користувачів із обмеженими ресурсами.

Teladoc Health є потужною платформою, яка забезпечує високий рівень послуг у сфері телемедицини. Її функціонал охоплює широкий спектр медичних потреб, але висока вартість, обмеження локальної доступності та технічні вимоги роблять її менш привабливою для малих медичних установ та пацієнтів із низьким доходом.

На рисунку 1.2 представлено приклад інтерфейсу Веб-сайту Teladoc Health

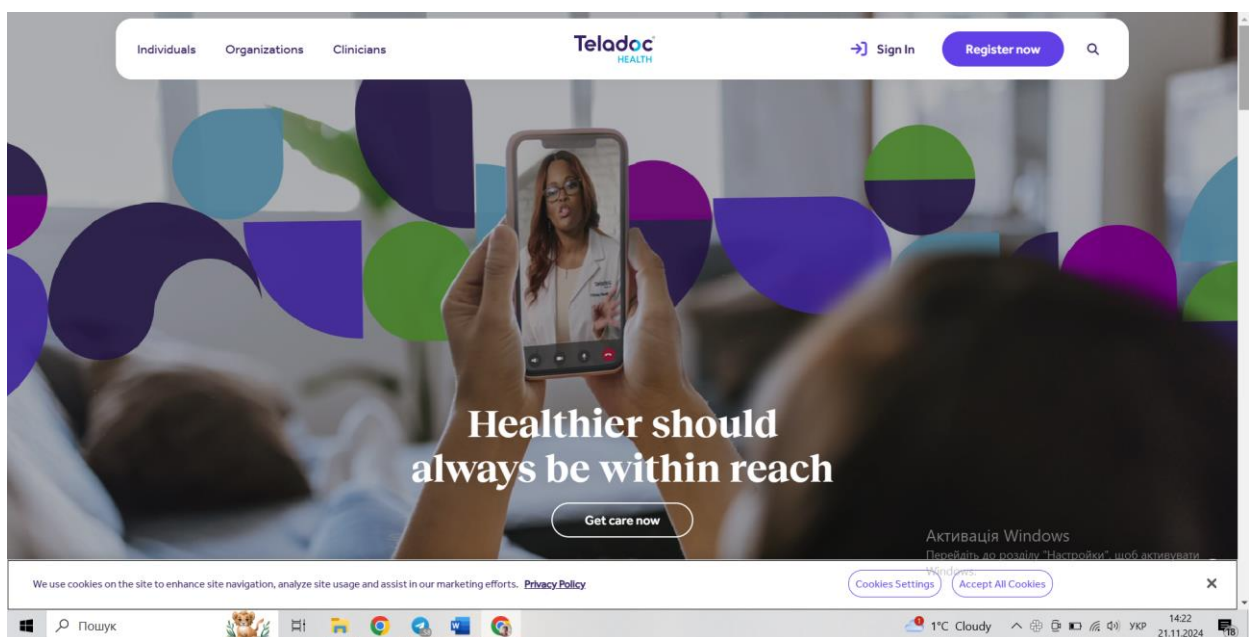


Рисунок 1.2 – Веб-сайт Teladoc Health

Запропонований у цій роботі веб-сайт лікарні з функціями телемедицини орієнтований на вирішення цих проблем, пропонуючи доступне, гнучке та локалізоване рішення, яке враховує потреби як медичних закладів, так і пацієнтів.

1.1.3 Веб-сайт Amwell

Amwell (American Well) є відомою телемедичною платформою, яка забезпечує дистанційний доступ до медичних послуг через веб-інтерфейс та мобільні додатки. Вона орієнтована на великі медичні заклади, страхові компанії та корпоративних клієнтів [5].

Основний функціонал веб-сайту:

- Дистанційні консультації. Amwell забезпечує проведення відеозустрічей між пацієнтом і лікарем у реальному часі.
- Інтеграція з EHR-системами. Платформа підтримує синхронізацію з електронними медичними записами, що полегшує доступ лікарів до історії хвороби пацієнтів.
- Цілодобовий доступ. Пацієнти можуть отримати консультації у будь-який час, що особливо корисно для невідкладних випадків.;

Переваги веб-сайту:

- Інтеграція з EHR-системами. Це забезпечує лікарям швидкий доступ до медичних записів пацієнтів.
- Цілодобова доступність. Пацієнти мають можливість звертатися до лікаря у будь-який час. На рисунку 1.3 представлено користувацький інтерфейс обмінника валют Moonpay.

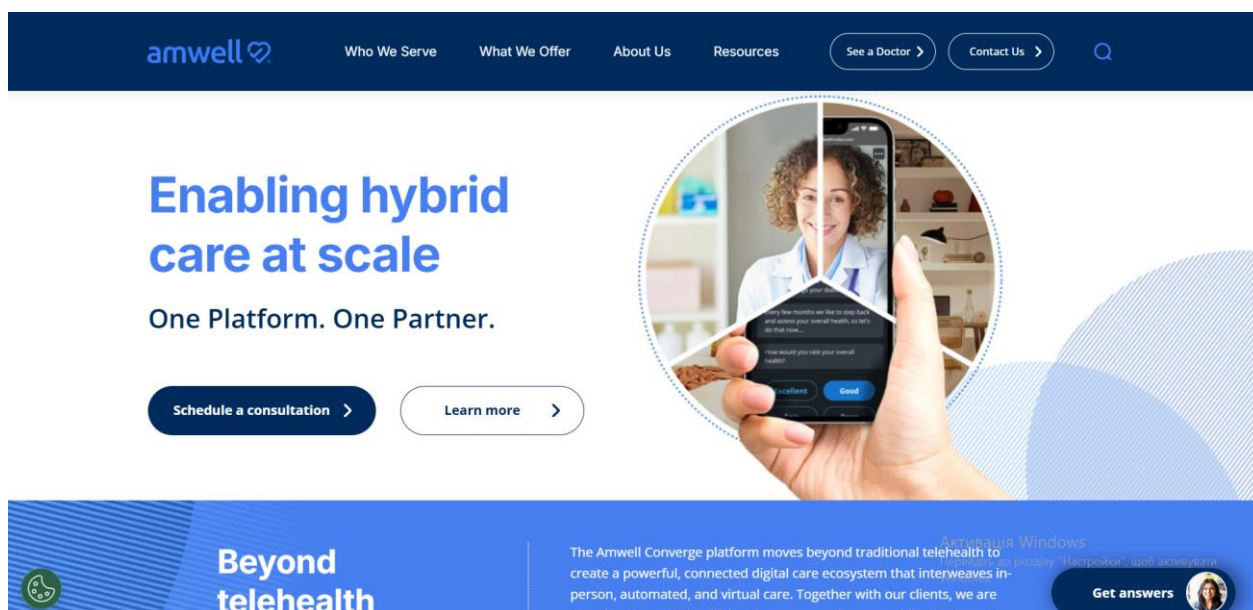


Рисунок 1.3 – Веб-сайт Amwell

Недоліки веб-сайту:

- Висока вартість впровадження та обслуговування. Платформа орієнтована на великі клініки, що робить її недоступною для малих медичних закладів.
- Складність локалізації. Веб-сайт не адаптований до багатьох мов і культур, що обмежує його використання у країнах, де англійська не є основною мовою.
- Обмежена гнучкість. Платформа має фіксовану архітектуру, яка не дозволяє адаптувати функціонал до потреб конкретного закладу.
- Залежність від якісного інтернет-з'єднання. Для проведення відеоконсультацій необхідний стабільний інтернет, що може бути проблемою у віддалених регіонах.
- Потреба в технічній інфраструктурі. Система потребує попередньої інтеграції з EHR-системами та сучасного серверного обладнання.
- Відсутність безкоштовного функціоналу. Базові послуги недоступні без платного доступу, що знижує її привабливість для пацієнтів із низьким рівнем доходу.

Amwell є ефективним рішенням для великих медичних установ, але її висока вартість, складність адаптації до локальних умов та технічні вимоги обмежують можливість використання платформи у малих лікарнях.

Запропонований у цій роботі веб-сайт лікарні з телемедичними функціями спрямований на вирішення цих проблем, пропонуючи доступне та гнучке рішення, яке може легко інтегруватися у будь-який медичний заклад.

1.1.4 Веб-сайт DOXY.me

DOXY.me є телемедичною платформою, яка спеціалізується на проведенні відеоконсультацій між лікарями та пацієнтами. Це просте у використанні рішення, орієнтоване на невеликі медичні установи, приватних лікарів та пацієнтів, які потребують зручного і швидкого доступу до дистанційної медичної допомоги. DOXY.me не вимагає завантаження додаткового програмного забезпечення, що робить її особливо зручною для користувачів [6].

Основний функціонал веб-сайту:

- Відеоконсультації. Платформа забезпечує високоякісне відеоз'єднання для дистанційної взаємодії лікаря з пацієнтом.
- Сумісність із будь-яким браузером. DOXY.me працює через веб-браузери, що дозволяє уникнути необхідності встановлення додаткових програм.
- Простота налаштування. Платформа не потребує складної реєстрації або додаткових інструментів для налаштування, що знижує бар'єр входу.
 - Функція "віртуальної кімнати". Кожен лікар має власну віртуальну кімнату, до якої пацієнти можуть приєднатися за спеціальним посиланням.

Переваги веб-сайту:

- Простота використання. DOXY.me не вимагає складного навчання або технічної підготовки, що робить її ідеальною для невеликих лікарень або індивідуальних лікарів.
- Безкоштовний базовий функціонал. Платформа надає можливість проведення консультацій безкоштовно, що приваблює лікарів із обмеженим бюджетом.
- Відсутність необхідності завантаження програмного забезпечення. Це значно спрощує процес використання для пацієнтів.

- Дотримання стандартів безпеки. Платформа відповідає вимогам HIPAA та GDPR щодо збереження конфіденційності даних.

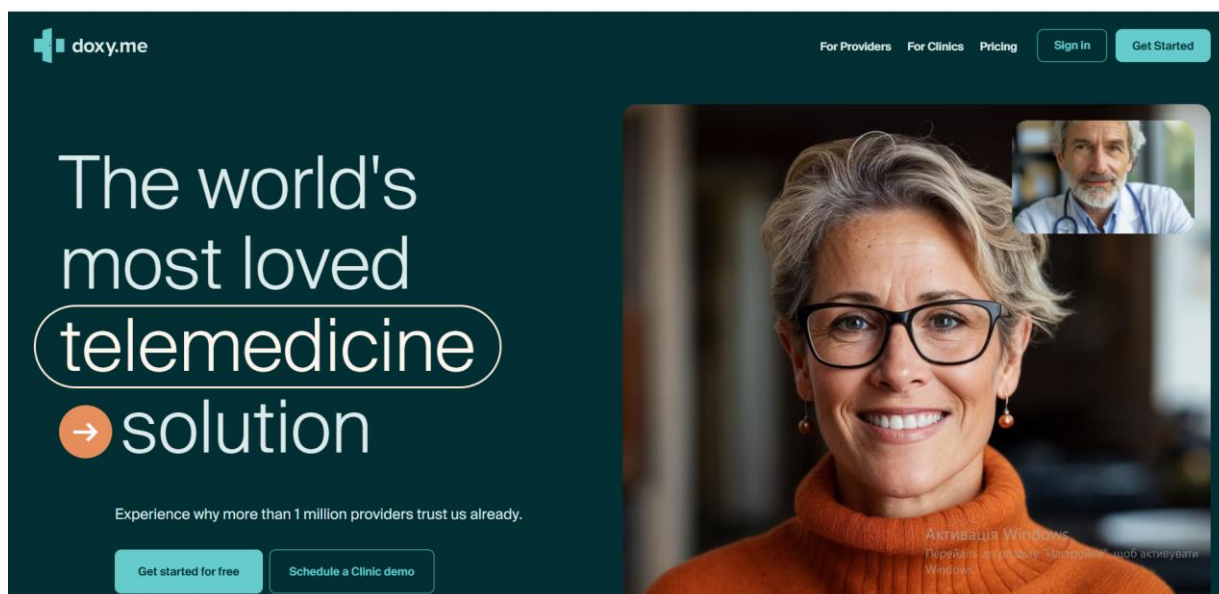


Рисунок 1.3 – Веб-сайт DOXY.me

Недоліки веб-сайту:

- Обмежений функціонал. У порівнянні з великими телемедичними платформами, DOXY.me надає лише базові можливості відеоконсультацій, без інтеграції з EHR-системами або інших розширених функцій.
- Залежність від інтернет-з'єднання. Якість відеозв'язку безпосередньо залежить від швидкості та стабільності інтернету, що може бути проблемою у віддалених регіонах.
- Відсутність інструментів для моніторингу стану здоров'я. Платформа не пропонує можливості введення або аналізу даних пацієнта, таких як артеріальний тиск, рівень цукру тощо.
- Відсутність персоналізації. DOXY.me не дозволяє лікарям або установам змінювати інтерфейс чи функціонал відповідно до своїх потреб.
- Складнощі в масштабуванні. Платформа не оптимізована для великих медичних закладів із великим потоком пацієнтів.

- Обмежена технічна підтримка. Безкоштовна версія платформи не передбачає оперативної технічної підтримки, що може створювати труднощі у разі виникнення проблем.
- Відсутність інтеграції з платіжними системами. Пацієнти не можуть здійснювати оплату послуг безпосередньо через платформу, що додає незручностей для лікарів і пацієнтів.

DOXY.me є популярним вибором для невеликих лікарень і приватних лікарів, оскільки забезпечує швидку та зручну організацію дистанційних консультацій без необхідності складного налаштування. Інтерфейс платформи інтуїтивно зрозумілий, що дозволяє лікарям і пацієнтам легко розпочати користування. Однак обмежений функціонал платформи зменшує її ефективність у більш складних сценаріях. Відсутність підтримки розширених інтеграцій, таких як електронні медичні записи (EHR/EMR), обмежує її використання у великих медичних установах, де потрібно автоматизувати й об'єднувати різні аспекти роботи. Крім того, DOXY.me стикається з труднощами в масштабуванні, що унеможливорює її ефективне застосування у великих мережах клінік або лікарень із великою кількістю пацієнтів.

Запропонований у цій роботі веб-сайт лікарні з інтегрованими телемедичними функціями розроблено з урахуванням недоліків DOXY.me. Наша система пропонує гнучке рішення, яке не тільки спрощує проведення відеоконсультацій, але й надає розширені можливості моніторингу стану здоров'я пацієнтів. Завдяки інтеграції з іншими системами, такими як електронні медичні записи, система забезпечує зручність доступу до даних і їх використання. Крім того, веб-сайт підтримує роботу з великим потоком користувачів, що дозволяє масштабувати його для застосування як у невеликих клініках, так і у великих лікарнях чи мережах медичних закладів. Це робить запропоноване рішення універсальним інструментом для сучасної медицини.

1.1.5 Локальні телемедичні рішення в Україні

В Україні розвиток телемедицини перебуває на етапі активного формування, що відкриває перспективи для впровадження інноваційних рішень у галузі дистанційної медицини. Наразі на ринку вже існують локальні платформи, які дозволяють пацієнтам записуватися на прийом до лікарів, отримувати онлайн-консультації та переглядати свої медичні записи. Такі сервіси є важливим кроком у розвитку телемедичних послуг, забезпечуючи базовий функціонал для зручності пацієнтів і лікарів.

Проте, більшість наявних рішень в Україні все ще стикаються з низкою обмежень. Часто їх функціонал не охоплює повний спектр послуг, необхідних для комплексного обслуговування пацієнтів. Відсутність інтеграції з електронними медичними записами (EHR/EMR), недостатня підтримка відеозв'язку та складнощі в масштабуванні платформ ускладнюють їх використання у великих медичних установах. Крім того, такі рішення часто не відповідають міжнародним стандартам телемедицини, які передбачають високий рівень безпеки даних, інтерактивність та адаптивність до різних сценаріїв використання.

Ця ситуація створює необхідність у розробці більш універсальних і функціональних платформ, які могли б конкурувати на міжнародному ринку та задовольняти потреби як невеликих медичних закладів, так і великих лікарень. Запропоноване в роботі рішення враховує специфіку українського ринку та пропонує комплексний підхід, який поєднує сучасні технології, гнучкість у використанні та відповідність міжнародним стандартам телемедицини.[7].

Основні локальні телемедичні платформи:

- Helsi.me.

Helsi.me є однією з найбільш відомих платформ в Україні. Вона забезпечує запис на прийом до лікаря, перегляд розкладу лікарів, доступ до результатів аналізів та медичних записів. Платформа інтегрована з системою eHealth, що спрощує управління медичними даними для лікарів та пацієнтів. Однак Helsi.me

обмежена у функціях телемедицини, зокрема, не підтримує відеоконсультації або моніторинг здоров'я пацієнтів.

- Medinet.

Платформа Medinet орієнтована на організацію медичних послуг для приватних клінік. Вона пропонує можливість онлайн-запису, створення електронних медичних записів та управління графіком лікарів. Однак, її функціонал зосереджений переважно на адміністративних завданнях і не включає інтеграцію з EHR-системами чи проведення дистанційних консультацій.

- Doc.ua.

Doc.ua — це онлайн-платформа, яка дозволяє пацієнтам знаходити лікарів, записуватися на прийом та залишати відгуки. Незважаючи на зручний інтерфейс, платформа не пропонує функцій для повноцінних телемедичних консультацій або моніторингу здоров'я пацієнтів.

- Askep.net.

Ця платформа розроблена для телемедичних консультацій у форматі текстових і відеозв'язків. Основними користувачами є сімейні лікарі та пацієнти, які мають підписані декларації. Водночас система має обмежену інтеграцію з іншими медичними рішеннями.

Переваги локальних рішень:

- Доступність для пацієнтів. Більшість платформ безкоштовні або доступні за мінімальну плату.
- Інтеграція з системою eHealth. Це забезпечує лікарям швидкий доступ до медичних даних пацієнтів.
- Локалізація. Платформи розроблені з урахуванням специфіки української медицини.

Недоліки локальних рішень:

- Обмежений функціонал. Більшість платформ не підтримують повноцінні відеоконсультації, моніторинг стану здоров'я чи інтеграцію з міжнародними стандартами телемедицини.
- Недостатня безпека даних. Не всі рішення відповідають сучасним стандартам збереження конфіденційної інформації (наприклад, GDPR або HIPAA).
- Залежність від державного регулювання. Інтеграція з eHealth обмежує їхню гнучкість у розширенні функціоналу.
- Відсутність комплексних рішень. Платформи зосереджуються на окремих аспектах медичних послуг і не пропонують інтегрованого підходу для лікарень.
- Технічні обмеження. Багато платформ не оптимізовані для використання у віддалених регіонах із поганим інтернет-з'єднанням.
- Недостатня підтримка телемедицини. Відеоконсультації або можливість введення даних про стан здоров'я пацієнта є рідкісними функціями серед українських платформ. DOXY.me є зручним і доступним рішенням для невеликих лікарень і приватних лікарів, яке дозволяє швидко організовувати дистанційні консультації. Однак її обмежений функціонал, відсутність розширених можливостей інтеграції та персоналізації, а також складнощі в масштабуванні роблять платформу менш придатною для великих медичних установ.

Локальні телемедичні рішення в Україні мають потенціал для розвитку, але значно поступаються міжнародним платформам за функціоналом, безпекою та масштабованістю. Їхній фокус здебільшого зосереджений на адміністративних завданнях, що не дозволяє повноцінно реалізувати потенціал телемедицини.

Запропонований у цій роботі веб-сайт лікарні з телемедичними функціями спрямований на подолання цих обмежень. Він забезпечує розширений функціонал, зокрема відеоконсультації, моніторинг стану здоров'я та інтеграцію з існуючими системами, що дозволяє реалізувати сучасні стандарти телемедицини.

1.2 Порівняння можливостей запропонованого рішення з існуючими аналогами

У ході аналізу телемедицинських платформ було здійснено детальний розгляд як міжнародних, так і локальних рішень, включаючи такі відомі сервіси, як Mayo Clinic, Teladoc Health, Amwell, DOXY.me, а також українські платформи Helsi.me та Doc.ua. Кожна з цих платформ вирізняється своїми унікальними особливостями, функціоналом і підходом до реалізації телемедицинських послуг.

Міжнародні платформи, такі як Mayo Clinic і Teladoc Health, пропонують широкий спектр функцій, які здатні задовольнити потреби великих медичних закладів, забезпечуючи високий рівень інтеграції з медичними записами, можливості відеоконсультацій і додаткові сервіси для пацієнтів. Однак такі рішення здебільшого вимагають значних фінансових вкладень і технічної інфраструктури, що ускладнює їх використання у невеликих або середніх лікарнях.

Українські сервіси, такі як Helsi.me та Doc.ua, зосереджені переважно на базовому функціоналі, зокрема записі на прийом і доступі до інформації про лікарів. Хоча вони є доступними та зручними для пацієнтів, ці рішення обмежені в контексті функціоналу, що стосується комплексного обслуговування пацієнтів. Наприклад, їм бракує інтеграції з сучасними технологіями моніторингу стану здоров'я або підтримки відеоконсультацій на високому рівні.

Аналіз цих платформ продемонстрував, що більшість із них або орієнтовані на великі медичні заклади з високими бюджетами, або мають обмежений функціонал, який не враховує сучасні вимоги телемедицини, такі як інтерактивність, масштабованість і персоналізовані підходи до обслуговування пацієнтів. Це підкреслює потребу в розробці універсального рішення, яке поєднуватиме доступність, функціональність і адаптивність для різних медичних закладів, зокрема й тих, що функціонують у межах України.

Основні відмінності запропонованого рішення:

- Доступність для малих і середніх лікарень.

Більшість міжнародних платформ, таких як Mayo Clinic або Teladoc Health, орієнтовані на великі медичні центри, що робить їх недосяжними для малих лікарень через високу вартість впровадження та обслуговування. Запропоноване рішення забезпечує економічну доступність для невеликих установ, дозволяючи їм інтегрувати сучасні телемедичні функції за помірну ціну [8].

- Локалізація та адаптація до місцевих умов.

Більшість міжнародних платформ не враховують специфіку місцевого законодавства та особливості української медицини. Запропонований веб-сайт підтримує локалізацію інтерфейсу, інтеграцію з системою eHealth та враховує вимоги українського законодавства.

- Розширений функціонал.

На відміну від локальних рішень, таких як Helsi.me або Doc.ua, запропонована платформа надає не лише функції запису на прийом, але й можливість:

- проведення відеоконсультацій;
- інтеграції з платіжними системами для оплати медичних послуг онлайн;
- введення даних пацієнтом (наприклад, рівень цукру в крові, тиск) з можливістю їх аналізу лікарем у реальному часі.

- Гнучкість і масштабованість.

Запропоноване рішення базується на сучасних веб-технологіях (Vue.js, Firebase), що дозволяє легко адаптувати систему до змінних потреб лікарень, забезпечуючи масштабування від локальних клінік до великих медичних центрів.

- Безпека даних.

У той час як деякі локальні платформи мають проблеми із захистом даних, запропоноване рішення використовує Firebase для шифрування та захисту медичної інформації відповідно до стандартів GDPR і HIPAA.

Платформа	Відеоконсультації	Інтеграція з EHR	Моніторинг стану здоров'я	Локалізація	Доступність для малих лікарень	Вартість
Mayo Clinic	Так	Так	Ні	Ні	Ні	Висока
Teladoc Health	Так	Частково	Ні	Ні	Ні	Висока
Amwell	Так	Так	Ні	Ні	Ні	Висока
DOXY.me	Так	Ні	Ні	Частково	Так	Низька
Helsi.me	Ні	Так	Ні	Так	Так	Низька
Запропоноване рішення	Так	Так	Так	Так	Так	Середня

Рисунок 1.4 - Порівняння можливостей

Запропоноване рішення є більш універсальним, ніж існуючі аналоги, оскільки поєднує в собі переваги міжнародних платформ із врахуванням потреб локального ринку. Воно забезпечує доступність, локалізацію, розширений функціонал та безпеку, що робить його ідеальним вибором для українських медичних установ.

1.3 Можливості інтеграції телемедичних сервісів у веб-сайти

Інтеграція телемедичних сервісів у веб-сайти лікарень створює потужний інструмент для модернізації системи охорони здоров'я, надаючи пацієнтам можливість отримувати якісну медичну допомогу без фізичного відвідування закладу. Використання сучасних технологій, таких як відеоконференції, інтерактивні особисті кабінети та онлайн-моніторинг стану здоров'я, забезпечує зручну взаємодію між лікарями та пацієнтами в реальному часі.

Це рішення дозволяє оптимізувати роботу медичного персоналу, зменшуючи адміністративне навантаження та автоматизуючи процеси, такі як запис на прийом, нагадування про консультації та обробка медичних записів. Крім того, такі інструменти є незамінними в умовах віддалених або сільських регіонів, де доступ до медичних закладів може бути ускладнений.

Інтеграція телемедичних функцій також сприяє підвищенню рівня доступності медичної допомоги, дозволяючи пацієнтам отримувати необхідні послуги незалежно від місця їхнього перебування. Це особливо важливо для людей із обмеженою мобільністю, хронічними захворюваннями чи тих, хто проживає у віддалених районах. Таким чином, телемедичні сервіси не тільки підвищують ефективність медичної допомоги, але й роблять її більш персоналізованою та доступною для всіх категорій населення.

Основні функції, які можуть бути інтегровані у веб-сайти лікарень:

- Запис на прийом онлайн.

Пацієнти можуть самостійно обирати зручний час та дату для консультації, переглядати доступні графіки лікарів та отримувати автоматичні підтвердження запису.

- Відеоконсультації.

Інтеграція відеозв'язку дозволяє лікарям проводити дистанційні прийоми, оцінювати стан здоров'я пацієнтів та надавати рекомендації у режимі реального часу. Це особливо актуально для первинних консультацій або моніторингу пацієнтів із хронічними захворюваннями.

- Особистий кабінет пацієнта.

У рамках особистого кабінету пацієнти можуть отримувати доступ до своїх медичних записів, результатів аналізів, рекомендацій лікаря та розкладу прийомів. Це підвищує прозорість і контроль за процесом лікування.

- Моніторинг стану здоров'я.

Веб-сайти можуть бути інтегровані з системами, які дозволяють пацієнтам вводити показники здоров'я, такі як артеріальний тиск, рівень цукру в крові чи температура тіла. Лікарі можуть аналізувати ці дані для прийняття рішень щодо подальшого лікування.

- Онлайн-оплата послуг.

Додатковою зручністю є можливість інтеграції платіжних систем для сплати за консультації чи обстеження безпосередньо через веб-сайт.

- Автоматичні нагадування.

Пацієнти можуть отримувати сповіщення про майбутні прийоми, відеозустрічі чи необхідність пройти обстеження. Це знижує ймовірність пропущених візитів і допомагає підтримувати дисципліну лікування.

- Захист даних та конфіденційність.

Інтеграція функцій шифрування та автентифікації користувачів забезпечує захист персональних і медичних даних пацієнтів відповідно до міжнародних стандартів, таких як GDPR та HIPAA.

Технічні аспекти інтеграції:

- Фронтенд-розробка.

Використання сучасних технологій, таких як Vue.js, дозволяє створювати інтерактивний і зручний інтерфейс для пацієнтів та лікарів.

- Бекенд-інтеграція.

Для зберігання даних та організації авторизації можна використовувати такі платформи, як Firebase, що забезпечують надійність, масштабованість та безпеку.

- Інтеграція з EHR-системами.

Телемедичний функціонал можна синхронізувати з електронними медичними записами для зберігання і доступу до історії хвороб пацієнтів.

- Використання API для відеозв'язку.

Інтеграція сервісів, таких як Twilio або Jitsi Meet, дозволяє організувати стабільний і якісний відеозв'язок.

- Модульна структура.

Використання модульної архітектури дозволяє поступово розширювати функціонал веб-сайту без суттєвих змін у його базовій структурі.

Переваги інтеграції телемедичних сервісів у веб-сайти:

- Зручність для пацієнтів, які отримують доступ до медичних послуг із будь-якого місця та пристрою.
- Оптимізація роботи лікарів завдяки автоматизації рутинних процесів (запис, ведення медичних записів, нагадування).
- Економія часу та ресурсів як для лікарень, так і для пацієнтів.

- Підвищення доступності медичних послуг, особливо для пацієнтів у віддалених регіонах.

Інтеграція телемедичних сервісів у веб-сайти лікарень значно розширює можливості медичних закладів, створюючи новий стандарт у наданні медичних послуг. Завдяки впровадженню сучасних технологій, таких як відеоконсультації, моніторинг здоров'я в реальному часі та інтеграція електронних медичних записів, запропоноване рішення сприяє підвищенню якості медичного обслуговування.

Ключовою перевагою такої інтеграції є забезпечення високого рівня захищеності даних, що є критично важливим у медичній сфері. Використання сучасних протоколів безпеки гарантує конфіденційність інформації пацієнтів та мінімізує ризики витоку даних. Крім того, запропонована система вирізняється гнучкістю у налаштуванні, дозволяючи адаптувати функціонал до потреб конкретного медичного закладу — від невеликих клінік до великих багатопрофільних лікарень.

Таке рішення не тільки підвищує ефективність медичних процесів, але й сприяє доступності послуг для пацієнтів, незалежно від їхнього місця проживання. Інтеграція телемедичних сервісів відкриває нові перспективи для розвитку медицини, дозволяючи лікарням швидко адаптуватися до змін у сфері охорони здоров'я та забезпечувати високоякісне обслуговування, яке відповідає сучасним стандартам.

1.4 Висновки до розділу

У першому розділі було проведено детальне дослідження предметної галузі телемедицини, що дозволило виявити ключові аспекти впровадження телемедичних сервісів у сучасну медичну практику. Проведений аналіз продемонстрував значний потенціал телемедичних платформ для покращення доступності, якості та ефективності надання медичних послуг.

Огляд існуючих телемедичних платформ, таких як Mayo Clinic, Teladoc Health, Amwell, DOXY.me та локальні рішення України, показав, що більшість із них або мають обмежений функціонал, або орієнтовані на великі медичні установи, що ускладнює їх застосування в малих та середніх лікарнях. Усі аналізовані платформи мають свої сильні сторони, такі як інтеграція з електронними медичними записами чи підтримка відеоконсультацій, проте значні недоліки, зокрема висока вартість, обмежена гнучкість і недостатня локалізація, створюють бар'єри для їх масового впровадження.

Запропоноване рішення має на меті усунути ці обмеження. Воно орієнтоване на створення доступної, локалізованої та гнучкої платформи для медичних закладів, що дозволить легко інтегрувати сучасні телемедичні сервіси у роботу лікарень. Аналіз можливостей інтеграції показав, що сучасні веб-технології, такі як Vue.js та Firebase, дозволяють створити масштабоване рішення із зручним інтерфейсом, високим рівнем безпеки даних та функціями, які підвищують ефективність взаємодії між лікарями та пацієнтами.

Таким чином, результати першого розділу слугують міцною основою для подальшого проєктування та реалізації веб-сайту з телемедичними функціями, що відповідає сучасним вимогам і враховує специфіку українського ринку медичних послуг.

2 ВИБІР ТЕХНОЛОГІЧНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ СИСТЕМИ

Розробка веб-сайту лікарні з інтегрованими телемедицинськими функціями є багатогранним і складним процесом, що потребує обґрунтованого вибору технологій для досягнення максимальної ефективності. Від правильного підбору мов програмування, фреймворків і інструментів залежить не лише реалізація ключових функцій, але й продуктивність, стабільність, безпека, зручність у використанні та довготривала підтримка системи.

Сучасні телемедичні платформи висувають високі вимоги до функціоналу, інтерактивності та безпеки даних. Вони мають забезпечувати швидкий і безпечний обмін інформацією між пацієнтами та лікарями, підтримувати відеозв'язок у реальному часі, динамічно оновлювати дані та легко масштабуватися під час збільшення кількості користувачів. Крім того, технології мають бути зручними для підтримки, що особливо важливо для медичних установ із обмеженими ресурсами на технічну підтримку.

Для реалізації цього проєкту було обрано низку технологій, які оптимально відповідають поставленим вимогам. Основу фронтенду складають мови HTML, CSS та JavaScript, які забезпечують створення інтерактивного інтерфейсу з високою продуктивністю. Як основний фреймворк для розробки фронтенду обрано Vue.js, який поєднує простоту у використанні з потужними можливостями для створення динамічних веб-додатків. Firebase було обрано для управління базами даних і авторизацією, оскільки ця технологія забезпечує високу безпеку даних, інтеграцію в реальному часі та простоту у використанні. Для організації відеозв'язку застосовано PeerJS, який дозволяє створювати стабільні однорангові з'єднання, необхідні для проведення телемедичних консультацій.

Вибір цих технологій базувався на їхній відповідності основним вимогам проєкту, таким як інтерактивність, продуктивність, безпека та масштабованість. У цьому розділі буде детально описано обґрунтування кожного вибору,

включаючи порівняння з альтернативними технологіями, їхні переваги та недоліки. Такий підхід дозволить зрозуміти, чому обрані технологічні засоби є найкращим рішенням для створення функціонального, надійного та сучасного веб-сайту лікарні з телемедицинськими функціями.

2.1 Вибір мов програмування для створення веб-сайту

Для створення веб-сайту лікарні з інтегрованими телемедицинськими функціями особливу увагу необхідно приділити вибору мов програмування, які гарантують стабільність роботи системи, високу продуктивність та зручність як для користувачів, так і для розробників. Важливо врахувати специфіку проєкту, яка передбачає розробку інтерактивного інтерфейсу для пацієнтів і лікарів, здатного підтримувати функції відеозв'язку, динамічного оновлення даних та персоналізованого доступу до медичних записів.

На етапі розробки веб-інтерфейсу необхідно обирати мови програмування та бібліотеки, які дозволяють створювати сучасний і адаптивний дизайн, що забезпечує зручність взаємодії користувачів із системою. HTML та CSS виступають основою для структури та візуального оформлення веб-сайту, а JavaScript забезпечує динамічність інтерфейсу, додаючи інтерактивні елементи, такі як форми, повідомлення та інтеграцію із зовнішніми сервісами.

Для серверного середовища важливо використовувати технології, які забезпечують надійність зберігання та обробки даних, особливо у сфері медицини, де питання безпеки конфіденційної інформації є критично важливим. Вибір відповідних інструментів дозволяє оптимізувати роботу системи, скоротити час розробки та забезпечити можливість масштабування проєкту у майбутньому.

У підсумку, обрані технології повинні не лише відповідати вимогам сучасного програмного забезпечення, але й враховувати специфіку медичної

галузі, дозволяючи створити інтуїтивно зрозумілу та функціональну платформу, яка задовольнить потреби як пацієнтів, так і лікарів.

2.1.1 HTML, CSS, JavaScript: обґрунтування вибору

Для реалізації фронтенд-частини веб-сайту лікарні обрано HTML, CSS і JavaScript, які є стандартними технологіями веб-розробки. Їхній вибір обумовлений здатністю забезпечити інтерактивність, адаптивність і привабливий дизайн сайту. Ці технології поєднують високу продуктивність, гнучкість та широку підтримку серед сучасних браузерів і розробницьких інструментів.

HTML (HyperText Markup Language) є фундаментом будь-якого веб-додатку, виступаючи базовим інструментом для створення структури веб-сторінок та організації елементів інтерфейсу. Його універсальність дозволяє створювати як прості, так і складні багатосторінкові додатки, які можуть включати різноманітні інтерактивні компоненти та динамічний контент. Альтернативні інструменти, такі як WYSIWYG-редактори (наприклад, Webflow або Wix), пропонують зручність для швидкого прототипування або створення невеликих проєктів без необхідності знання коду [9]. Проте такі редактори мають значні обмеження, коли мова йде про гнучкість, кастомізацію або реалізацію складних інтеграцій із сучасними технологіями. HTML, завдяки своїй простоті, масштабованості та широкій підтримці браузерами, залишається найкращим вибором для створення веб-додатків будь-якої складності. Його використання відкриває безліч можливостей для налаштування та адаптації структури проєкту відповідно до специфічних потреб, забезпечуючи довговічність і зручність підтримки створених систем.

На рисунку 2.1 наведено порівняльну характеристику HTML із можливими альтернативами.

Критерій	HTML	Markdown	WYSIWYG-редактори
Призначення	Створення структури веб-сторінок.	Легке форматування тексту для документів.	Автоматизоване створення сторінок без коду.
Гнучкість	Максимальна, підтримує інтеграцію зі стилями (CSS) і скриптами (JavaScript).	Обмежена можливостями форматування.	Дуже низька, складно кастомізувати.
Сфера застосування	Усі типи веб-додатків.	Прості статичні сторінки або документація.	Лише прості сайти чи прототипи.
Легкість освоєння	Легкий для початківців.	Дуже простий, не потребує знань програмування.	Потребує роботи з інтерфейсом, але не дає розуміння коду.
Масштабованість	Легко масштабується для складних веб-додатків.	Не підходить для складних систем.	Неможливість масштабування для реальних проєктів.

Рисунок 2.1 – Порівняльна характеристика HTML із можливими альтернативами

CSS (Cascading Style Sheets) є ключовим інструментом для візуального оформлення веб-сторінок, забезпечуючи можливість додати привабливість, адаптивність і унікальність до інтерфейсу [10]. Він використовується для визначення кольорів, шрифтів, розташування елементів, а також адаптації дизайну під різні пристрої, такі як смартфони чи планшети. На відміну від фреймворків, таких як Bootstrap або Tailwind CSS, які надають готові стилі й компоненти, базовий CSS забезпечує розробникам повний контроль над стилізацією [11]. Це дозволяє створювати унікальний дизайн, повністю адаптований до потреб конкретного проєкту, у нашому випадку — веб-сайту лікарні.

Хоча використання фреймворків може суттєво пришвидшити процес розробки, особливо на початкових етапах, вони часто додають зайвий код, який не використовується, але впливає на продуктивність і може ускладнити подальшу підтримку [12]. Базовий CSS, навпаки, дозволяє уникнути подібних проблем, забезпечуючи чистоту коду та точне налаштування стилів. Це особливо важливо для телемедичного веб-сайту, де необхідно забезпечити як зручність для

користувачів, так і відповідність візуального оформлення медичним стандартам. CSS також дозволяє легко інтегрувати адаптивність, що робить веб-сайт зручним для доступу з будь-якого пристрою, включаючи мобільні телефони та планшети, які часто використовуються пацієнтами для запису на консультації.

Критерій	CSS	Bootstrap	Tailwind CSS
Призначення	Надає стилі для елементів веб-сторінки.	Забезпечує набір готових стилів і компонентів.	Надає класи для швидкого стилізування елементів.
Гнучкість	Повний контроль над дизайном.	Обмежена можливість кастомізації готових компонентів.	Висока гнучкість, але класи можуть перевантажувати HTML.
Простота використання	Вимагає знань щодо створення стилів із нуля.	Легкий старт завдяки готовим компонентам.	Легкий старт, але складно підтримувати великий проєкт.
Масштабованість	Легко масштабується, забезпечує унікальність дизайну.	Підходить для середніх і великих проєктів із шаблонним дизайном.	Добре масштабується, але складність підтримки зростає зі збільшенням проєкту.
Продуктивність	Висока, оскільки немає зайвого коду.	Може містити непотрібні стилі, що впливають на продуктивність.	Менш ефективний через надлишок класів у HTML.

Рисунок 2.2 – Порівняльна характеристика CSS із можливими альтернативами

JavaScript є основним інструментом для забезпечення динамічності та інтерактивності веб-сайтів, що особливо актуально для нашого проєкту, який включає функції телемедицини. Він дозволяє реалізовувати складні елементи взаємодії, такі як відеоконсультації між лікарем і пацієнтом, динамічне оновлення вмісту сторінки чи валідація форм у реальному часі, що покращує користувацький досвід і знижує ризик помилок. Завдяки своїй нативній підтримці браузером, JavaScript не потребує додаткових інструментів для роботи, що значно спрощує розробку та скорочує час впровадження [13].

У порівнянні з TypeScript, який є надбудовою над JavaScript і додає сувору типізацію, чистий JavaScript більш підходить для проєктів із невеликою командою або обмеженим часом реалізації. TypeScript, хоча й забезпечує кращу структурованість коду для великих проєктів, потребує додаткової компіляції, що збільшує час налаштування та залежність від інструментів. У випадку нашого веб-сайту, де важливими є швидкість розробки та гнучкість, JavaScript є оптимальним вибором.

Альтернативи, такі як Python через WebAssembly, хоча й дозволяють використовувати Python для створення веб-додатків, мають значні накладні витрати. Використання WebAssembly вимагає додаткової трансляції коду для роботи в браузері, що значно знижує продуктивність і ускладнює інтеграцію. JavaScript, як мова, розроблена спеціально для браузерів, працює швидше та ефективніше, що робить його найкращим рішенням для інтерактивного та динамічного функціоналу, який необхідний для телемедичного веб-сайту.

На рисунку 2.3 наведено порівняльну характеристику JavaScript із можливими альтернативами.

Критерій	JavaScript	TypeScript	Python (WebAssembly)
Призначення	Динаміка та інтерактивність.	Те саме, з додатковою типізацією.	Серверні обчислення у браузері.
Продуктивність	Висока, нативна підтримка браузерів.	Подібна до JavaScript.	Низька, через трансляцію коду.
Масштабованість	Легко масштабується для будь-яких проєктів.	Ідеальна для великих проєктів.	Підходить для специфічних задач.
Залежність від інструментів	Мінімальна, працює без додаткових налаштувань.	Потребує компілятора.	Вимагає складної інтеграції.

Рисунок 2.3 – Порівняльна характеристика JavaScript із можливими альтернативами

Перевагою цього набору технологій є їхня універсальність та легка інтеграція з іншими інструментами, такими як фреймворки для фронтенду чи бібліотеки для роботи з даними. До недоліків можна віднести залежність від додаткових засобів для створення складніших функцій, таких як інтеграція відеозв'язку або оптимізація продуктивності для великих проєктів. Однак ці обмеження можна подолати завдяки використанню сучасних підходів до розробки.

Таким чином, HTML, CSS і JavaScript є найкращим вибором для створення веб-сайту лікарні, оскільки вони забезпечують гнучкість, масштабованість і високу якість кінцевого продукту. Їхній вибір дозволяє реалізувати всі необхідні функції для телемедичних послуг, відповідаючи сучасним вимогам користувачів і лікарів.

2.1.2 Vue.js як основний фреймворк

Для створення динамічного інтерфейсу веб-сайту лікарні з телемедициними функціями було обрано Vue.js як основний фронтенд-фреймворк [14]. Цей інструмент забезпечує продуктивність, гнучкість і зручність у використанні, що дозволяє швидко створювати сучасні веб-додатки з багатим функціоналом. Vue.js використовує компонентну архітектуру, яка дозволяє організувати код у вигляді окремих блоків, забезпечуючи простоту підтримки й можливість масштабування проєкту. Крім того, його реактивність автоматично оновлює інтерфейс у відповідь на зміни даних, що є критично важливим для динамічних систем, таких як телемедицина.

Однією з ключових переваг Vue.js є легкість інтеграції з іншими технологіями. Фреймворк може поступово впроваджуватися у вже існуючий проєкт або використовуватися як основа для створення нового додатка. Vue.js також підтримує екосистему інструментів, таких як Vue Router для

маршрутизації та Vuex для управління станом, що забезпечує повноцінний функціонал для великих і малих проєктів.

Нижче наведено порівняння Vue.js із популярними альтернативами:

- React, як один із найпопулярніших фреймворків, забезпечує розробникам широкі можливості завдяки своїй розвиненій екосистемі та великій спільноті користувачів, яка постійно створює нові бібліотеки та інструменти. Проте для початку роботи з React розробникам часто доводиться витратити більше часу на налаштування середовища та освоєння концепцій, таких як управління станом чи використання додаткових бібліотек. Vue.js, у порівнянні, має більш простий і зрозумілий синтаксис, що дозволяє швидко розпочати розробку, особливо для менш досвідчених фахівців. Цей фреймворк пропонує інтуїтивний підхід, який значно полегшує роботу з малими й середніми проєктами, де важливими є швидкість реалізації та зручність використання базових інструментів [15].
- Angular вирізняється широким спектром вбудованих функцій та інструментів, що дозволяють розробникам створювати складні, багатофункціональні додатки. Проте така багатогранність нерідко супроводжується громіздким синтаксисом і підвищеною складністю використання, що може стати викликом для менших команд або проєктів із обмеженими ресурсами. У порівнянні з цим, Vue.js пропонує більш інтуїтивний і гнучкий підхід до розробки, зосереджуючись на легкості використання і швидкому освоєнні, що особливо важливо для невеликих проєктів або команд, які прагнуть досягти результату за короткий час [16].

Основні переваги фреймворку Vue.js для розробки веб-сайту лікарні з використанням телемедицини:

- Компонентна архітектура: Код організовується у вигляді незалежних блоків, що полегшує підтримку та повторне використання.
- Реактивність: Автоматичне оновлення інтерфейсу при зміні стану даних.
- Гнучкість: Можливість поступового впровадження у вже існуючі проєкти.
- Ефективність: Легка вага фреймворка забезпечує швидкий рендеринг.

- Екосистема: Vue Router, Vuex і інші інструменти спрощують розробку.

На рисунку 2.4 наведено порівняльну характеристику фреймворку Vue.js, React та Angular.

Критерій	Vue.js	React	Angular
Простота навчання	Легкий у вивченні, зрозуміла документація.	Потребує більше часу для освоєння.	Складний через громіздкий синтаксис.
Компонентна архітектура	Підтримує модульну архітектуру.	Потребує додаткових бібліотек.	Чітка, але громізка структура.
Гнучкість	Легко інтегрується у проекти будь-якого типу.	Вимагає більше налаштувань.	Менш гнучкий через складність.
Продуктивність	Легка вага, швидкий рендеринг.	Висока продуктивність після оптимізації.	Може бути надмірним для простих проектів.
Масштабованість	Оптимальний для середніх проектів.	Добре масштабується.	Ідеальний для великих корпоративних систем.

Рисунок 2.4 – Порівняльна характеристика Vue.js, React та Angular

Vue.js є оптимальним вибором для створення веб-сайту лікарні завдяки його простоті, гнучкості й здатності забезпечити ефективну роботу динамічного інтерфейсу. У порівнянні з альтернативами Vue.js пропонує кращий баланс між функціональністю та простотою впровадження, що дозволяє реалізувати всі вимоги до телемедичних функцій.

2.2 Використання Firebase для авторизації та бази даних

Для забезпечення надійного та зручного управління авторизацією користувачів і зберіганням даних було обрано платформу Firebase. Це хмарне рішення від Google, яке надає широкий спектр інструментів і сервісів для реалізації бекенд-функціоналу, що дозволяє спростити розробку та управління базою даних і системою авторизації.

Firebase було обрано з кількох причин. По-перше, він має вбудовану підтримку для аутентифікації користувачів, включаючи електронну пошту, Google, Facebook та інші методи входу [17]. По-друге, Firestore – хмарна база даних Firebase – дозволяє зберігати структуровані дані з можливістю їхньої швидкої обробки, синхронізації та масштабування. Нижче наведено порівняння Firebase з іншими популярними рішеннями для авторизації та баз даних.

Порівняння Firebase із альтернативами:

- Firebase та AWS Cognito: Firebase забезпечує простіший процес інтеграції та використання, тоді як AWS Cognito підходить для складніших корпоративних проєктів, але має вищий поріг входу [18].
- Firebase Firestore та MySQL: Firestore не потребує встановлення серверів і автоматично масштабується, тоді як MySQL вимагає ручного налаштування та підтримки.
- Firebase та MongoDB Atlas: Firebase спрощує синхронізацію даних у реальному часі, тоді як MongoDB більше підходить для великих структурованих даних із гнучкою схемою.

Переваги використання Firebase:

- Простота інтеграції: Firebase має SDK для багатьох мов програмування, що полегшує його впровадження.
- Готова система авторизації: Підтримка кількох методів входу, включаючи багатфакторну аутентифікацію.
- Хмарна база даних (Firestore): Дані синхронізуються в реальному часі та доступні для всіх авторизованих користувачів.
- Безпека: Firebase пропонує вбудовані правила безпеки для захисту даних.

На рисунку 2.5 наведено порівняльну характеристику Firebase із можливими альтернативами.

Критерій	Firebase	AWS Amplify	Supabase
Простота інтеграції	Легка, готові SDK для швидкого старту.	Складна, вимагає налаштування AWS-ресурсів.	Проста, але потребує більше ручної роботи з базою.
Реальний час	Підтримка реального часу в Firestore.	Відсутня підтримка реального часу.	Присутня, але менш оптимізована.
Автоматичне масштабування	Вбудоване, працює "із коробки".	Потребує ручного налаштування ресурсів.	Обмежене можливостями PostgreSQL.
Екосистема	Широка, підтримує базу даних, аутентифікацію, хостинг і багато іншого.	Менш інтегрована екосистема.	Зосереджена переважно на базі даних.

Рисунок 2.5 – Порівняльна характеристика Firebase із можливими альтернативами

Firebase є ідеальним рішенням для створення веб-сайту лікарні з телемедичними функціями завдяки своїй простоті, гнучкості та підтримці сучасних стандартів безпеки. Його інтеграція дозволяє швидко налаштувати авторизацію, зберігати дані пацієнтів і лікарів, а також забезпечити синхронізацію даних у реальному часі. У порівнянні з альтернативами, Firebase забезпечує оптимальний баланс між продуктивністю, зручністю використання та можливостями масштабування.

2.3 Інтеграція відеоконференцій для телемедичних консультацій

Однією з ключових функцій телемедичного веб-сайту є можливість організації відеоконференцій між лікарем і пацієнтом, що забезпечує дистанційне надання медичних послуг і спрощує взаємодію пацієнта з медичним закладом. Ця функція є важливою складовою сучасних телемедичних платформ, оскільки дозволяє лікарям проводити консультації в реальному часі, оцінювати стан пацієнтів, а також обговорювати результати обстежень або лікувальні плани без необхідності фізичного відвідування лікарні.

Для реалізації цієї функціональності в нашому проєкті обрано бібліотеку PeerJS, яка є зручним і сучасним інструментом для створення однорангових з'єднань на основі технології WebRTC. PeerJS значно спрощує впровадження відеозв'язку, оскільки надає доступ до потужностей WebRTC через високорівневий API, знімаючи необхідність написання складного низькорівневого коду [19]. Це дозволяє розробникам зосередитися на функціональності, що прямо впливає на зручність і якість обслуговування користувачів, замість вирішення технічних питань на рівні мережевих протоколів.

PeerJS надає можливість швидко створювати надійні та ефективні з'єднання для обміну відео- та аудіоданими, що є ключовим аспектом роботи телемедицинських систем. Це особливо важливо для забезпечення стабільності зв'язку та мінімізації затримок під час консультацій, адже якість відео- та аудіозв'язку безпосередньо впливає на точність діагностування та комфорт пацієнтів. Такий підхід дозволяє створити надійну, гнучку та просту в інтеграції систему, яка відповідає сучасним потребам галузі охорони здоров'я.

PeerJS було обрано через його переваги:

- Бібліотека забезпечує високорівневий API, який значно спрощує створення WebRTC-з'єднань.
- Легко інтегрується у фронтенд-проєкти, такі як ті, що базуються на Vue.js.
- Забезпечує прямий обмін даними між клієнтами, що знижує затримки та не потребує потужного серверного ресурсу для ретрансляції відео.
- PeerJS може працювати як із хмарними, так і локальними серверами, що робить його адаптивним до потреб проєкту.

PeerJS будує з'єднання на основі технології WebRTC, яка дозволяє клієнтам (пацієнту й лікарю) обмінюватися відео та аудіоданими напряму. Для цього використовується так званий "сигнальний сервер", який відповідає за обмін початковими даними для встановлення з'єднання. У нашому випадку:

1. Лікар і пацієнт отримують унікальні PeerJS-ідентифікатори.
2. Через сигнальний сервер ідентифікатори передаються, і встановлюється однорангове з'єднання.
3. Відео та аудіо передаються безпосередньо між клієнтами, з мінімальними затримками.

Переваги PeerJS у порівнянні з альтернативами:

- PeerJS та OpenTok (TokBox): PeerJS є легшим і простішим у використанні для невеликих проєктів, таких як веб-сайт лікарні. OpenTok надає більше функцій для масштабних корпоративних платформ, але має складніший API.
- PeerJS та Jitsi Meet: Jitsi вимагає більше ресурсів для розгортання власного сервера, тоді як PeerJS дозволяє використовувати вже існуючий сигнальний сервер або легко створити свій.
- PeerJS та Zoom SDK: Zoom SDK є комерційним продуктом із платним доступом до багатьох функцій. PeerJS є безкоштовним і відкритим, що робить його економічно вигідним вибором.

Розглянемо переваги, які роблять PeerJS оптимальним вибором для інтеграції відеоконференцій:

- Мінімальні затримки: P2P-з'єднання дозволяє знизити затримки під час відеозв'язку.
- Простота інтеграції: Зручний API дозволяє швидко впровадити відеоконференції у проєкт.
- Відкритий вихідний код: PeerJS є безкоштовною бібліотекою з активною спільнотою розробників.
- Гнучкість: Легко інтегрується з будь-якою фронтенд-платформою, включаючи Vue.js.
- Ефективність: Переважна більшість операцій виконується на клієнтській стороні, що знижує навантаження на сервер.

На рисунку 2.6 представлено порівняння PeerJS із альтернативами

Критерій	PeerJS	Jitsi Meet	Zoom SDK
Простота інтеграції	Легкий у використанні, простий API.	Потребує налаштування серверів.	Складна інтеграція, специфічні вимоги.
Функціональність	Відео, аудіо, P2P-з'єднання.	Групові конференції, запис, екран.	Розширені функції, аналітика, запис.
Вимоги до серверів	Мінімальні (сигнальний сервер).	Високі (власний сервер або хмара).	Використовує сервери Zoom.
Вартість	Безкоштовний.	Безкоштовний для базового використання.	Платний, залежить від тарифу.
Масштабованість	Підходить для невеликих проєктів.	Добре працює для середніх і великих.	Масштабованість забезпечує Zoom.

Рисунок 2.6 – Порівняння бібліотек

PeerJS є ідеальним рішенням для впровадження відеоконференцій у телемедичному веб-сайті лікарні. Його використання дозволяє реалізувати стабільний і зручний відеозв'язок між лікарем і пацієнтом із мінімальними затратами часу й ресурсів. У порівнянні з іншими інструментами, PeerJS пропонує простоту інтеграції, ефективність і економічну вигідність, що робить його оптимальним вибором для нашого проєкту.

2.4. Висновки до розділу

У межах цього розділу було проведено детальний аналіз і обґрунтування вибору технологій, необхідних для створення веб-сайту лікарні з функціями телемедицини. Особлива увага приділялася вибору мов програмування, фреймворків, баз даних та інструментів для реалізації ключових функціональних можливостей. Вибрані технології спрямовані на забезпечення сучасного, інтерактивного та зручного для користувачів інтерфейсу, а також надійності та масштабованості бекенд-системи.

Для створення структури веб-сторінок було обрано HTML як базову технологію, що є стандартом у веб-розробці. Разом із CSS, яке відповідає за

стилізацію та адаптивний дизайн, і JavaScript, що додає динамічності, ці технології забезпечують повноцінну основу для реалізації функціоналу сайту. У процесі порівняння з альтернативами, такими як Bootstrap, Tailwind CSS, Markdown та іншими, обґрунтовано, що обраний набір технологій найбільш ефективно відповідає потребам проєкту.

Для розробки фронтенд-частини було обрано фреймворк Vue.js, який виділяється своєю простотою, гнучкістю та підтримкою компонентної архітектури. У ході аналізу Vue.js було порівняно з React і Angular, що підтвердило його переваги для реалізації проєктів малого та середнього масштабу, до яких належить і наш веб-сайт. Легкість навчання, простота інтеграції та можливість поступового впровадження у вже існуючі системи роблять Vue.js оптимальним вибором.

Для організації бекенд-частини і зберігання даних було обрано Firebase, який є хмарним сервісом із широкою екосистемою інструментів. Firebase забезпечує готову систему авторизації користувачів, підтримку роботи в реальному часі та автоматичне масштабування. У порівнянні з AWS Amplify і Supabase, Firebase показав себе як найбільш оптимальне рішення, особливо для проєктів, які потребують швидкого розгортання та мінімальних витрат на підтримку інфраструктури.

Для реалізації функції відеоконференцій між лікарем і пацієнтом обрано бібліотеку PeerJS, яка працює на основі технології WebRTC. PeerJS надає зручний інструментарій для організації однорангових з'єднань, що дозволяє забезпечити відео- та аудіозв'язок із мінімальними затримками. У ході порівняння з такими рішеннями, як Jitsi Meet і Zoom SDK, PeerJS було обрано через його простоту інтеграції та відсутність високих вимог до серверних ресурсів.

Таким чином, вибрані технології забезпечують баланс між простотою впровадження, продуктивністю та гнучкістю. Завдяки їхній інтеграції проєкт отримає сучасний функціонал, необхідний для організації телемедицини консультацій, а також надійність і зручність використання як для лікарів, так і

для пацієнтів. Запропоновані рішення сприяють створенню ефективного веб-сайту, який відповідає сучасним вимогам медичної галузі.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розділ "Розробка програмного забезпечення" є ключовим етапом у створенні веб-сайту лікарні з використанням телемедицини, адже саме на цьому етапі реалізуються всі технічні рішення та ідеї, закладені під час попередніх етапів аналізу та планування. Від якості розробки програмного забезпечення залежить не лише функціональність, але й стабільність, продуктивність та безпека системи, що є критично важливим у медичній галузі.

У цьому розділі увагу зосереджено на детальному обґрунтуванні обраних рішень для розробки функціоналу веб-сайту, який включає інтеграцію телемедичних сервісів, підтримку відеоконсультацій, зберігання медичних записів та забезпечення зручності використання для лікарів і пацієнтів. Основним завданням розробки є створення системи, яка відповідає сучасним стандартам якості, безпеки та масштабованості, при цьому залишаючись інтуїтивно зрозумілою та легкою у використанні.

Далі у цьому розділі буде детально розглянуто, як відбувається обґрунтування функціоналу, його проектування та технічна реалізація. Особлива увага приділяється забезпеченню інтерактивності системи, гнучкості в налаштуванні та адаптивності до потреб користувачів.

3.1 Обґрунтування та проектування функціоналу веб-сайту лікарні з використанням телемедицини

Проект веб-сайту лікарні з інтеграцією телемедичних технологій спрямований на забезпечення доступу до медичних послуг в онлайн-режимі. Основна мета — створити платформу, яка об'єднує пацієнтів та лікарів, дозволяючи їм взаємодіяти з використанням сучасних цифрових інструментів. Даний проект покликаний вирішити проблеми обмеженого доступу до якісної

медичної допомоги, зокрема в умовах віддалених регіонів або за наявності інших бар'єрів, які ускладнюють фізичний візит до лікарні.

Функціонал платформи охоплює можливість реєстрації та авторизації користувачів, запису на прийом до лікаря, зберігання медичних даних, організації відеоконсультацій тощо. Реалізація даних функцій дозволить значно підвищити ефективність роботи медичного закладу, полегшити доступ пацієнтів до консультацій та забезпечити якісний обмін інформацією між лікарем і пацієнтом.

Застосування телемедицини в рамках веб-сайту дозволяє лікарні не тільки підвищити рівень сервісу, але й значно розширити спектр медичних послуг, які можуть бути надані в режимі онлайн. Інтеграція бази даних Firebase забезпечує надійне управління даними користувачів, їх безпеку та доступність у реальному часі [20].

Для забезпечення ефективності роботи платформи було розроблено функціональні вимоги [21], наведені в таблиці 3.1.

Таблиця 3.1 – Функціональні вимоги до веб-сайту лікарні з телемедичними функціями

№	Вимоги
1	Користувачі (пацієнти та лікарі) повинні мати можливість реєстрації та авторизації через веб-сайт.
2	Після авторизації користувач отримує доступ до персонального профілю.
3	Пацієнти можуть переглядати список лікарів, доступних для запису на консультацію.
4	Надається функція запису на прийом до лікаря з можливістю вибору дати та часу.
5	Лікарі можуть переглядати записи пацієнтів через свій профіль.
6	Платформа дозволяє зберігати медичні картки пацієнтів у форматі електронних файлів.

7	Інтеграція системи відеозв'язку для організації онлайн-консультацій між лікарем та пацієнтом.
8	Забезпечення безпеки даних через використання Firebase для управління базою даних та аутентифікації.

3.2 Проектування бази даних

Проектування структури бази даних є одним із ключових етапів розробки веб-сайту лікарні з інтегрованими функціями телемедицини, оскільки саме від цього залежить ефективність зберігання, обробки та доступу до інформації. Добре спроектована база даних забезпечує надійну основу для функціонування всіх сервісів системи, включаючи управління записами пацієнтів, збереження медичних даних, підтримку відеоконсультацій і обробку запитів від лікарів і пацієнтів.

Процес проектування бази даних включає кілька важливих етапів. Спершу визначаються основні сутності, такі як пацієнти, лікарі, медичні записи, консультації та платіжні операції. Для кожної сутності визначаються атрибути, які описують її характеристики. Наприклад, для сутності "пацієнт" це можуть бути ім'я, вік, контактна інформація, історія захворювань, а для "консультації" — дата, час, лікар, тип консультації та статус.

Наступним кроком є визначення зв'язків між сутностями. У системі телемедицини сутності часто взаємопов'язані: наприклад, один пацієнт може мати кілька медичних записів або консультацій, а один лікар може обслуговувати кількох пацієнтів. Ці зв'язки моделюються за допомогою відповідних типів відносин (один-до-багатьох, багато-до-багатьох), які дозволяють ефективно організувати структуру даних.

Фінальним етапом є створення логічної та фізичної моделей даних, які відповідають вимогам системи. Це включає вибір типів даних для кожного атрибута, нормалізацію для усунення дублювання даних і оптимізацію для

підвищення швидкості виконання запитів. У нашому випадку база даних розробляється з урахуванням високих вимог до конфіденційності, тому особлива увага приділяється шифруванню даних та управлінню доступом.

Таким чином, правильно спроектована структура бази даних є основою для забезпечення стабільної, безпечної та ефективної роботи веб-сайту з телемедичними функціями, дозволяючи реалізувати всі передбачені функції та забезпечити високий рівень зручності для користувачів.

Для початку визначимо основні сутності бази даних, необхідні для реалізації функціоналу веб-сайту, і їхні атрибути.

Основні сутності та їх атрибути

1. Користувачі (users) – для зберігання інформації про пацієнтів і лікарів:
Унікальний ідентифікатор користувача (userID)
 - Ім'я (firstName)
 - Прізвище (lastName)
 - Роль (role) – лікар або пацієнт
 - Email
 - Пароль
 - Дата реєстрації (registrationDate)
2. Профіль лікаря (doctorProfile) – додаткова інформація про лікаря:
 - Унікальний ідентифікатор профілю (profileID)
 - Спеціалізація (specialization)
 - Досвід роботи (experienceYears)
 - Рейтинг (rating)
 - userID (зовнішній ключ)
3. Записи на прийом (appointments) – для збереження інформації про запис до лікаря:
 - Унікальний ідентифікатор запису (appointmentID)
 - userID пацієнта
 - userID лікаря
 - Дата та час прийому (appointmentDateTime)

- Статус запису (status)
4. Медичні картки (medicalRecords) – для зберігання електронних медичних документів:
- Унікальний ідентифікатор картки (recordID)
 - userID пацієнта
 - Назва документа (documentName)
 - Посилання на файл (fileURL)
 - Дата завантаження (uploadDate)
5. Чат (chat) – для організації онлайн-спілкування між лікарем і пацієнтом:
- Унікальний ідентифікатор чату (chatID)
 - userID лікаря
 - userID пацієнта
6. Повідомлення (messages) – для збереження історії повідомлень у чаті:
- Унікальний ідентифікатор повідомлення (messageID)
 - chatID
 - Відправник (senderID)
 - Текст повідомлення (text)
 - Мітка часу (timestamp)

Наступним етапом проектування бази даних є визначення типів зв'язків між сутностями. У базах даних зв'язки є ключовим елементом, який визначає взаємодію між таблицями та забезпечує ефективну організацію і доступ до даних [22].

У реляційних базах даних розрізняють три основні типи зв'язків між сутностями:

- Один до одного: кожен запис у першій таблиці пов'язаний з одним і лише одним записом у другій таблиці, і навпаки. Це взаємозв'язок, де кожен елемент одного набору даних може мати тільки одного відповідного елемента в іншому наборі.

- Один до багатьох: один запис у першій таблиці може бути пов'язаний з декількома записами у другій таблиці, тоді як кожен запис другої таблиці пов'язаний лише з одним записом у першій таблиці.
- Багато до багатьох: кожен запис у першій таблиці може бути пов'язаний з багатьма записами у другій таблиці, і навпаки. Для реалізації такого зв'язку використовується проміжна таблиця, яка фіксує всі можливі зв'язки між двома сутностями.

Для визначення зв'язків у проєктованій базі даних розглянемо приклад взаємодії між сутностями «Користувач» та «Записи на прийом». Зрозуміло, що тип зв'язку тут – «один до багатьох», оскільки один користувач може мати багато записів на прийом, але кожен запис на прийом пов'язаний лише з одним користувачем.

На основі цього аналізу можна створити ER-діаграму, яка ілюструє зв'язок між сутностями «Користувач» та «Записи на прийом». Вона демонструє, як дані організовані та взаємодіють між собою у базі даних, що сприяє ефективному зберігання і обробці інформації.

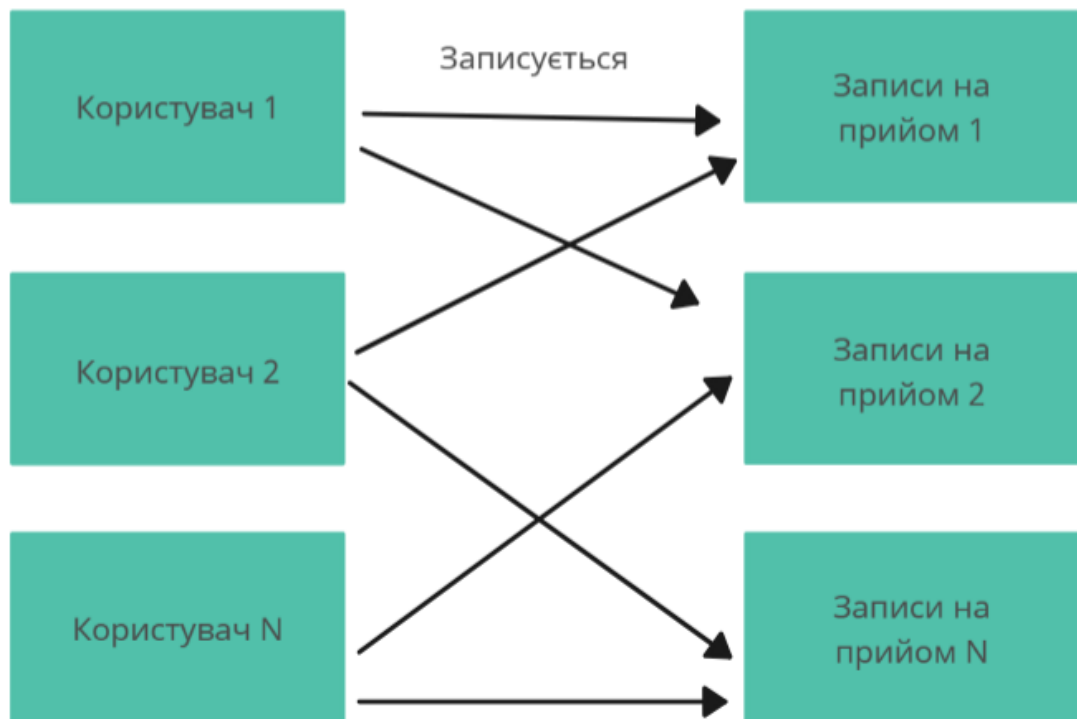


Рисунок 3.1 – ER-діаграма сутностей «Користувач – Записи на прийом»

Далі розглянемо взаємозв'язки між іншими сутностями проєктованої бази даних, які є ключовими для забезпечення логіки роботи системи.

Зв'язок між сутностями «Запис на прийом» та «Профіль лікаря». Кожен запис на прийом пов'язаний з одним лікарем через його профіль, а один профіль лікаря може мати багато записів. Таким чином, зв'язок між цими сутностями є «один до багатьох». На рисунку 3.2 представлено ER-діаграму сутностей «Запис на прийом – Профіль лікаря».

Зв'язок між сутностями «Користувач» та «Медична картка». Один користувач може зберігати багато медичних карток, але кожна медична картка належить лише одному користувачеві. Тому зв'язок між сутностями «Користувач» та «Медична картка» також є «один до багатьох». На рисунку 3.3 представлено ER-діаграму сутностей «Користувач – Медична картка».

Зв'язок між сутностями «Чат» та «Повідомлення». Один чат може містити багато повідомлень, але кожне повідомлення належить лише одному чату. Зв'язок між цими сутностями є «один до багатьох». На рисунку 3.4 представлено ER-діаграму сутностей «Чат – Повідомлення».

Зв'язок між сутностями «Користувач» та «Чат». Один користувач може спілкуватися в багатьох чатах, але в чаті може приймати участь один користувач. Тому зв'язок між сутностями «Користувач» та «Чат» є «один до багатьох». На рисунку 3.5 представлено ER-діаграму сутностей «Користувач – Чат».

Зв'язок між сутностями «Користувач» та «Профіль лікаря». Кожен запис у таблиці DoctorProfile відповідає лише одному запису в таблиці Users, оскільки профіль лікаря створюється тільки для користувачів із роллю "лікар". Тому зв'язок між сутностями «Користувач» та «Профіль лікаря» є «один до одного». На рисунку 3.6 представлено ER-діаграму сутностей «Користувач – Профіль лікаря».

Зв'язок між сутностями «Профіль лікаря» та «Чат». Один профіль лікаря може спілкуватися в багатьох чатах, але в чаті може приймати участь один профіль лікаря. Тому зв'язок між сутностями «Профіль лікаря» та «Чат» є «один

до багатьох». На рисунку 3.7 представлено ER-діаграму сутностей «Профіль лікаря – Чат».

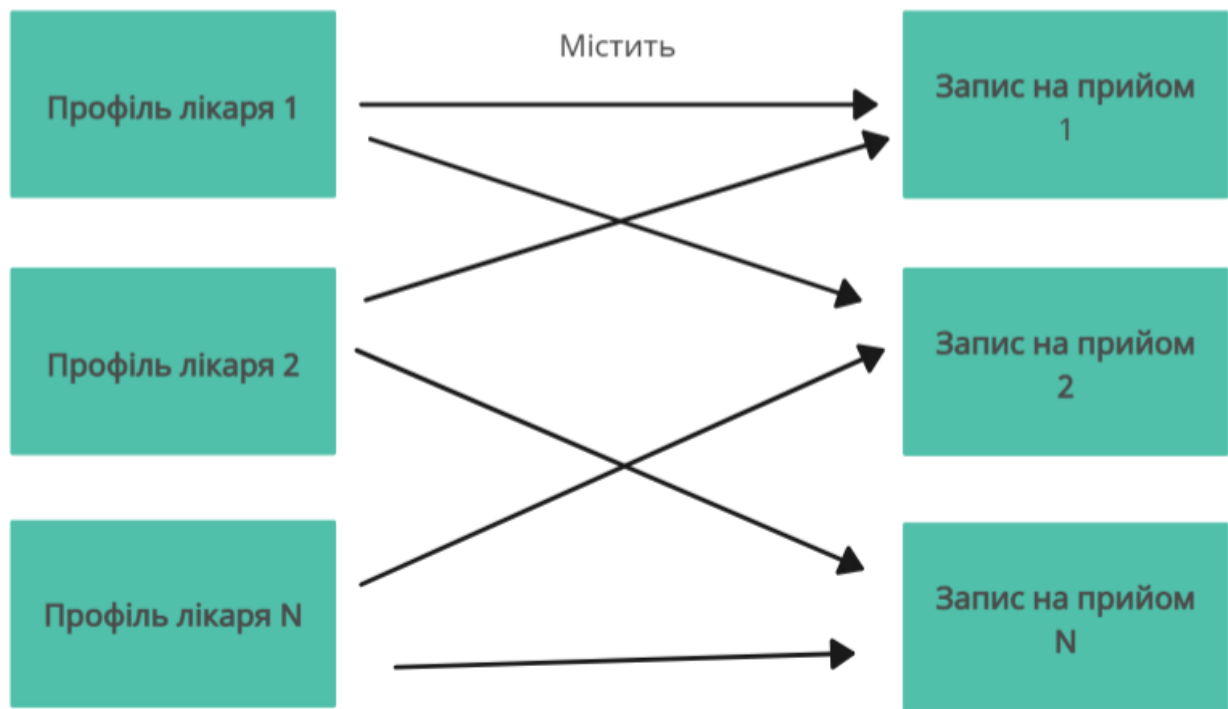


Рисунок 3.2 – ER-діаграма сутностей «Запис на прийом – Профіль лікаря»

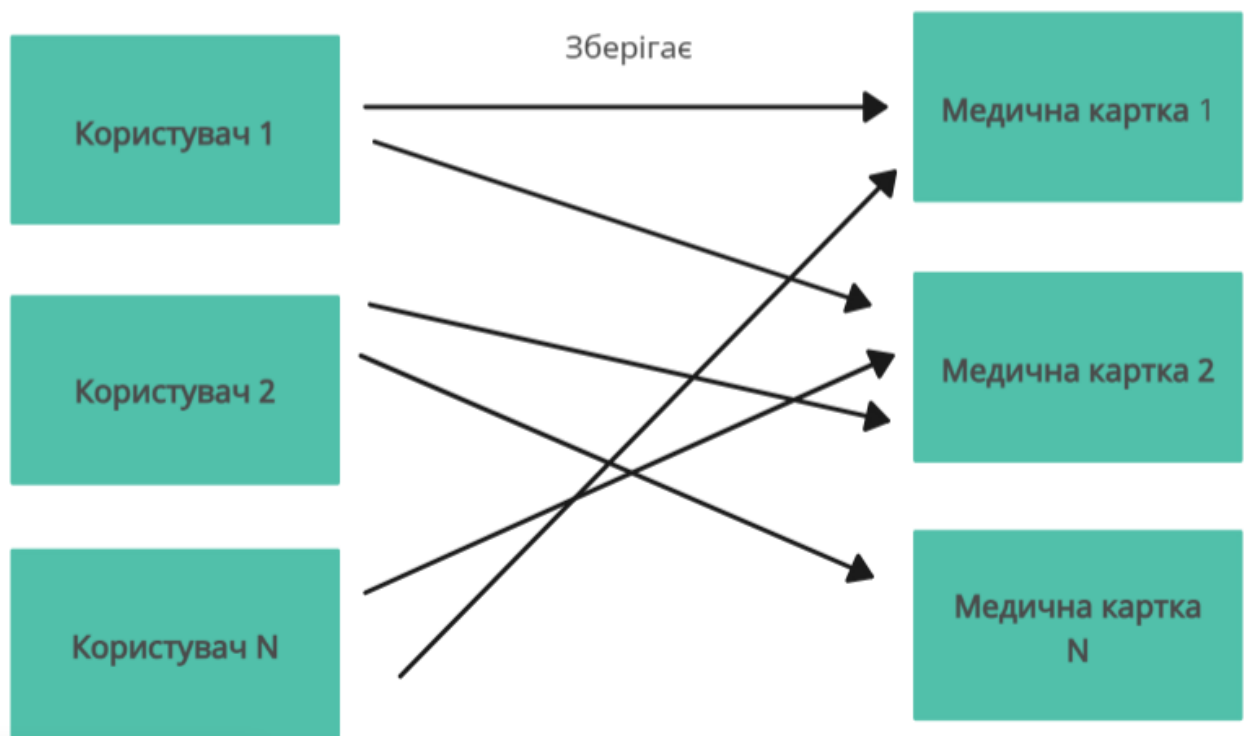


Рисунок 3.3 – ER-діаграма сутностей «Користувач – Медична картка»

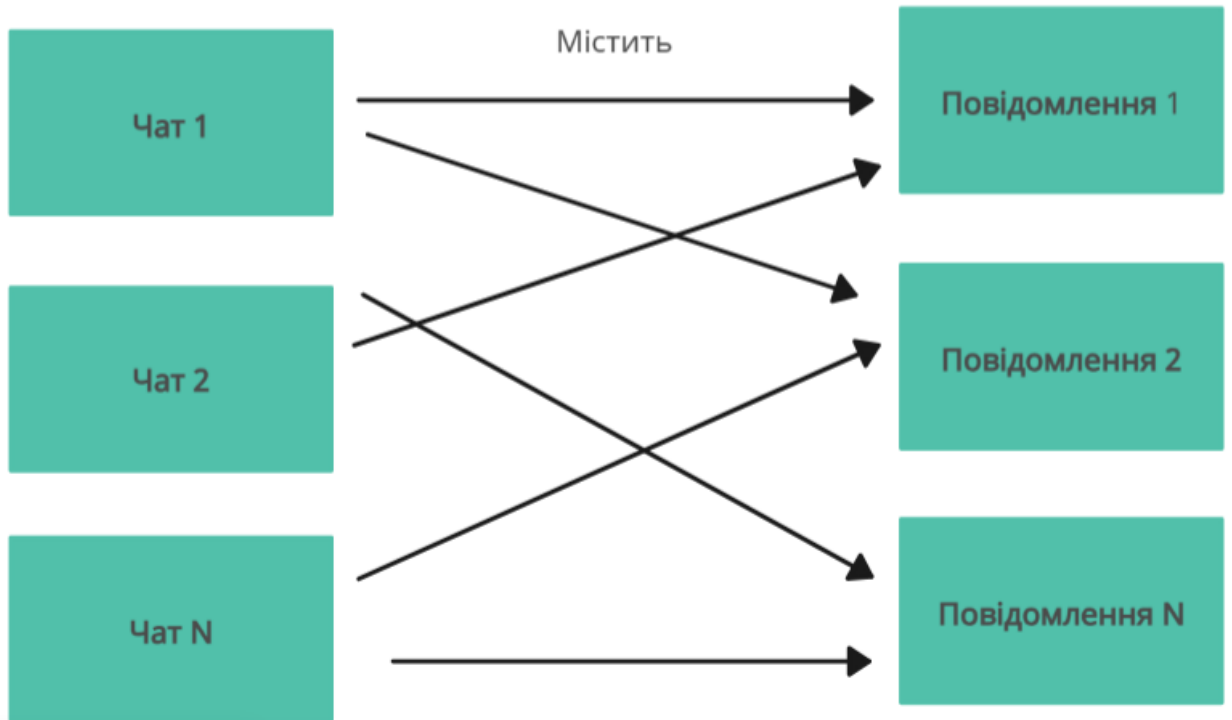


Рисунок 3.4 – ER-діаграма сутностей «Чат – Повідомлення»

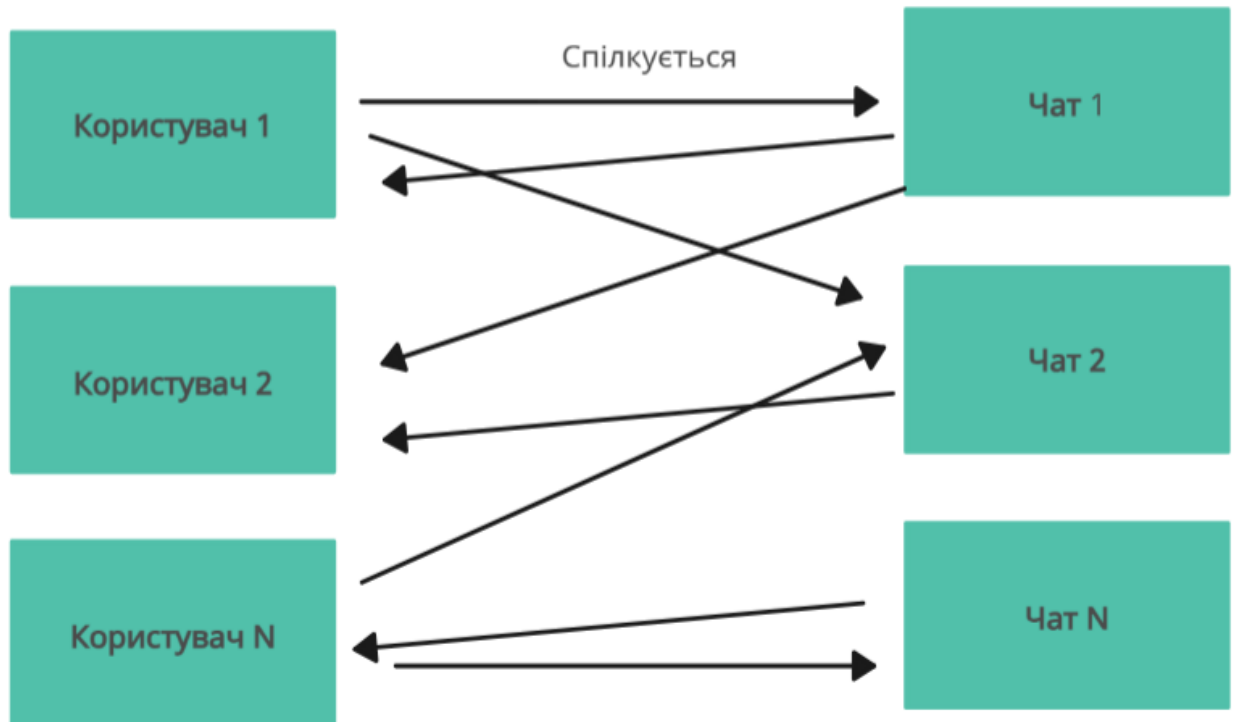


Рисунок 3.5 – ER-діаграма сутностей «Користувач – Чат»

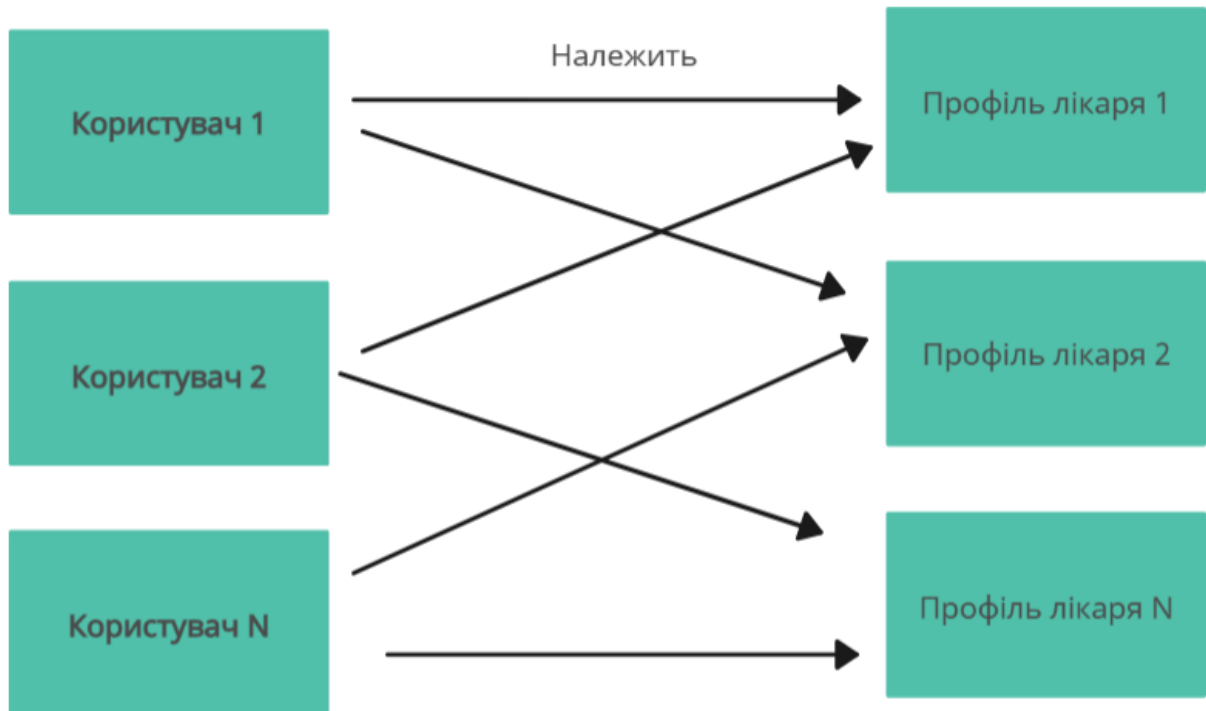


Рисунок 3.6 – ER-діаграма сутностей «Користувач – Профіль лікаря»

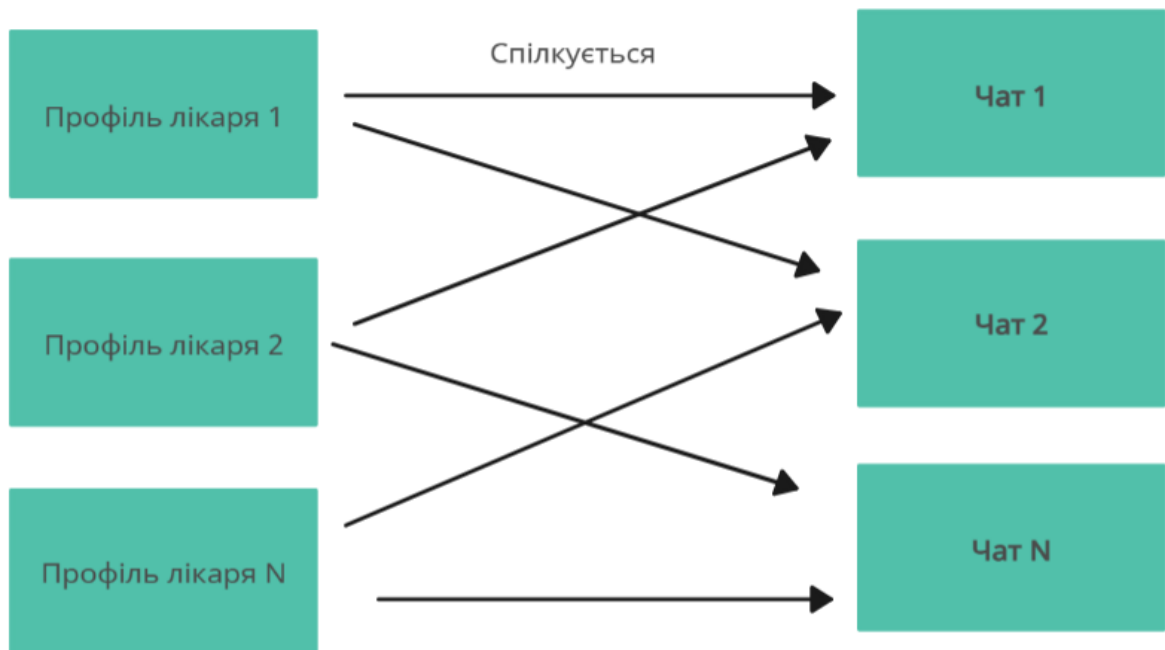


Рисунок 3.7 – ER-діаграма сутностей «Профіль лікаря – Чат»

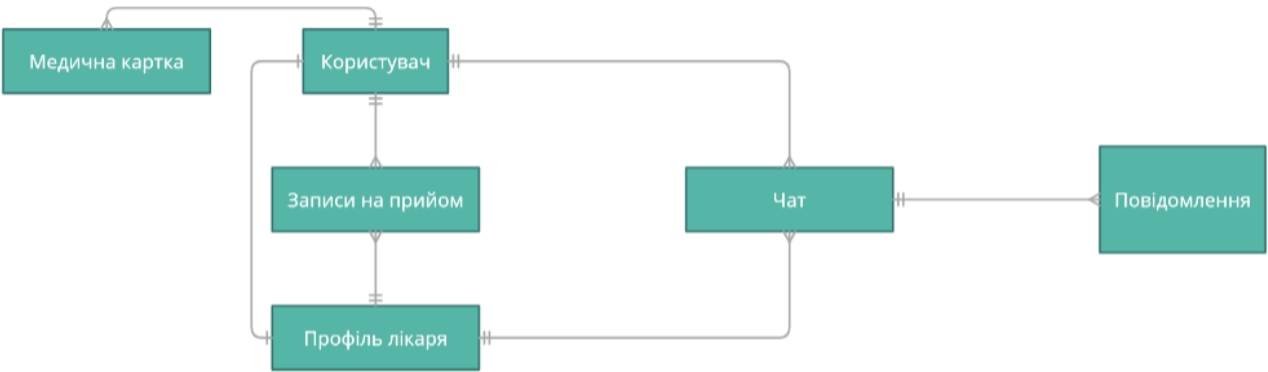


Рисунок 3.8 – Результуюча ER-діаграма сутностей веб-сайту

На основі результуючої ER-діаграми та визначених атрибутів було спроектовано модель бази даних. На рисунку 3.9 представлено спроектовану базу даних для веб-сайту лікарні з використанням телемедицини.

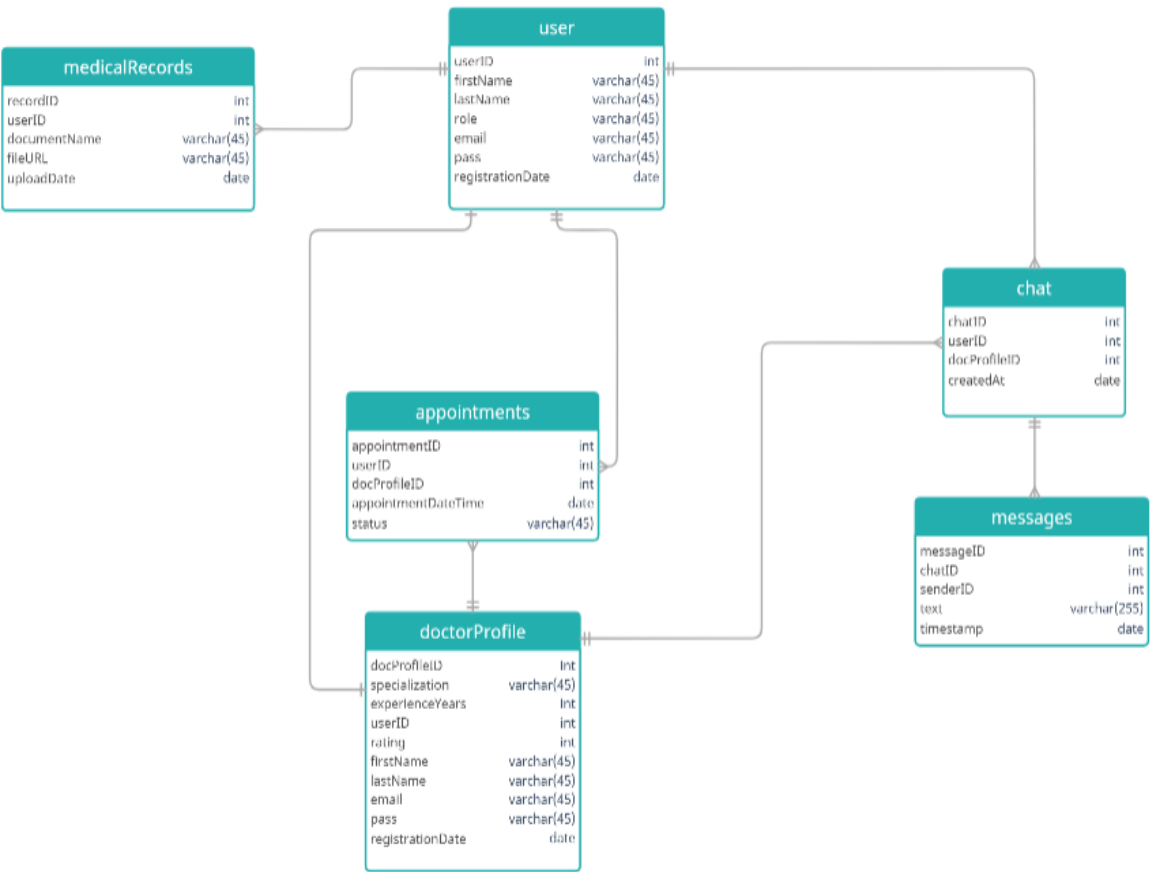


Рисунок 3.9 – Модель бази даних веб сайту

У таблиці `medicalRecords`, поле `userID` є зовнішнім ключем, який створює зв'язок із таблицею `user`, забезпечуючи збереження інформації про те, кому належить конкретна медична картка.

У таблиці `appointments`, поля `userID` та `docProfileID` є зовнішніми ключами. Поле `userID` зв'язує записи на прийом із пацієнтом у таблиці `user`, а поле `docProfileID` зв'язує запис із відповідним профілем лікаря у таблиці `doctorProfile`.

У таблиці `UserChat`, поля `userID` та `chatID` є зовнішніми ключами. Поле `userID` зв'язує учасника чату з таблицею `user`, а поле `chatID` зв'язує його з конкретним чатом у таблиці `chat`.

У таблиці `messages`, поле `chatID` є зовнішнім ключем, який створює зв'язок із таблицею `chat`, забезпечуючи збереження повідомлень, які належать до певного чату.

Розробка телемедичного веб-сайту є багатогранним процесом, що охоплює не лише реалізацію функціонального інтерфейсу, але й створення надійної системи зберігання та обробки даних. Особливу увагу під час розробки приділено архітектурі бази даних, яка виступає фундаментом для функціонування всіх ключових сервісів платформи. Завдяки продуманій структурі забезпечується швидкий доступ до інформації, інтеграція з різними модулями системи та безпечне зберігання даних пацієнтів.

Процес створення бази даних передбачає визначення всіх сутностей, їх атрибутів і зв'язків, що є необхідними для реалізації таких функцій, як управління записами на прийом, доступ до медичних карток, збереження історії консультацій та інтерактивна комунікація між лікарями й пацієнтами. Для цього базу даних було спроектовано з урахуванням вимог гнучкості, масштабованості та надійності, що дозволяє адаптувати систему до потреб конкретного медичного закладу.

Далі наведена діаграма, яка демонструє взаємодію основних компонентів системи: пацієнтів, лікарів, веб-сайту та бази даних. Вона ілюструє, як різні сутності об'єднані між собою для забезпечення безперебійної роботи всіх

функцій телемедичної платформи, створюючи зручну та ефективну екосистему для користувачів.

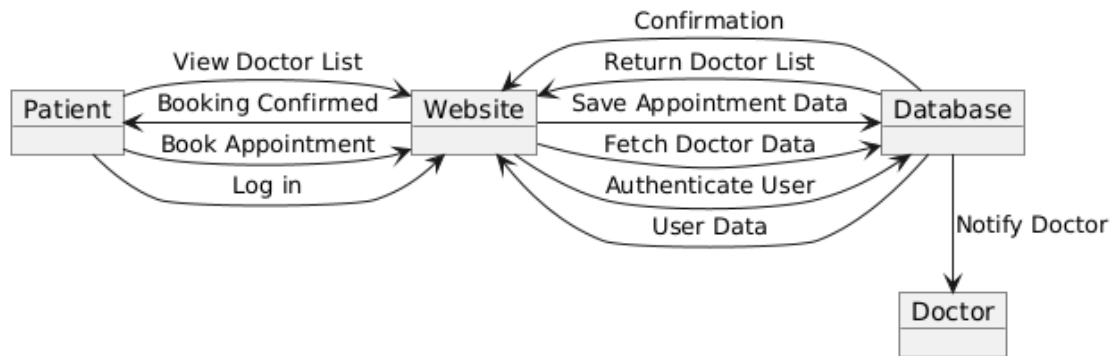


Рисунок 3.10 - Діаграма взаємодії

Ця діаграма ілюструє функціональну архітектуру взаємодії пацієнтів, лікарів, веб-сайту лікарні та бази даних у системі телемедицини. Основні етапи процесу, представлені на діаграмі, забезпечують повний цикл запису на консультацію, підтвердження та інформування відповідних сторін. Основні переваги такої архітектури:

- Централізація даних: База даних слугує надійним сховищем, забезпечуючи безперебійну роботу сервісу.
- Автоматизація процесів: Пацієнт і лікар отримують інформацію в реальному часі, що знижує адміністративне навантаження.
- Зручність для користувачів: Пацієнт може швидко переглядати інформацію про лікарів і записуватися на прийом у кілька кліків.

Ця архітектура ідеально підходить для сучасних телемедичних систем, забезпечуючи високу ефективність і інтерактивність.

Телемедичні платформи надають широкий спектр функціональних можливостей для різних категорій користувачів, таких як пацієнти, лікарі та адміністратори. Кожна з цих груп має власний набір завдань і доступ до специфічних функцій системи. Для досягнення високої ефективності роботи системи та зручності користування необхідно чітко розмежувати доступ і повноваження кожної групи, а також забезпечити їхню інтеграцію в єдиній системі.

На наступній діаграмі представлено основні функціональні можливості телемедичної платформи для кожної категорії користувачів. Ця структура демонструє, як реалізується взаємодія між пацієнтами, лікарями та адміністраторами в системі, забезпечуючи виконання основних задач і доступ до ключових сервісів платформи.

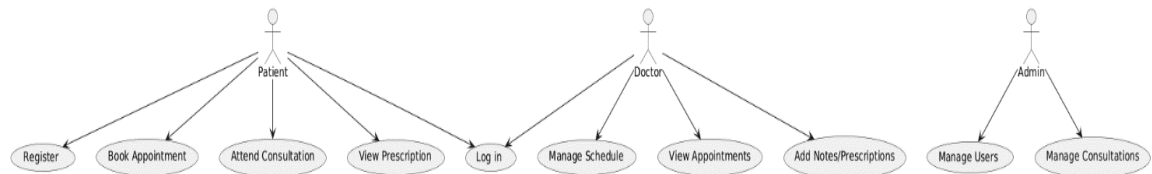


Рисунок 3.11 - Діаграма прецедентів

Діаграма демонструє функціональні можливості системи для кожної категорії користувачів:

- Пацієнти мають доступ до функцій реєстрації, запису на прийом, участі у консультаціях, перегляду рецептів і входу до системи. Такий набір функцій дозволяє забезпечити зручність користування системою і доступність медичних послуг у будь-який час.
- Лікарі отримують можливість керувати своїм розкладом, переглядати записи пацієнтів, а також додавати нотатки або виписувати рецепти. Це сприяє оптимізації їхньої роботи та мінімізації паперової документації.
- Адміністратори виконують управлінські функції, включаючи управління користувачами і консультаціями, що дозволяє підтримувати порядок і ефективність роботи системи.

Представлена структура забезпечує чіткий розподіл завдань і функцій між користувачами, що сприяє безперебійній роботі системи, знижує ризик помилок і підвищує загальну ефективність телемедичної платформи.

У процесі розробки телемедичного веб-сайту важливо деталізувати послідовність дій, які виконуються користувачами та системою для досягнення конкретних цілей. UML-діаграми діяльності є ефективним інструментом для

моделювання таких процесів, оскільки вони дозволяють візуалізувати динамічний аспект системи, показуючи, як користувачі взаємодіють з функціоналом веб-сайту.

Нижче наведена UML-діаграма діяльності, яка описує ключові етапи користування системою — від реєстрації чи авторизації до завершення консультації та надання відгуку. Вона демонструє не лише послідовність дій, але й можливості вибору користувачами певного шляху, що забезпечує гнучкість і багатофункціональність системи.

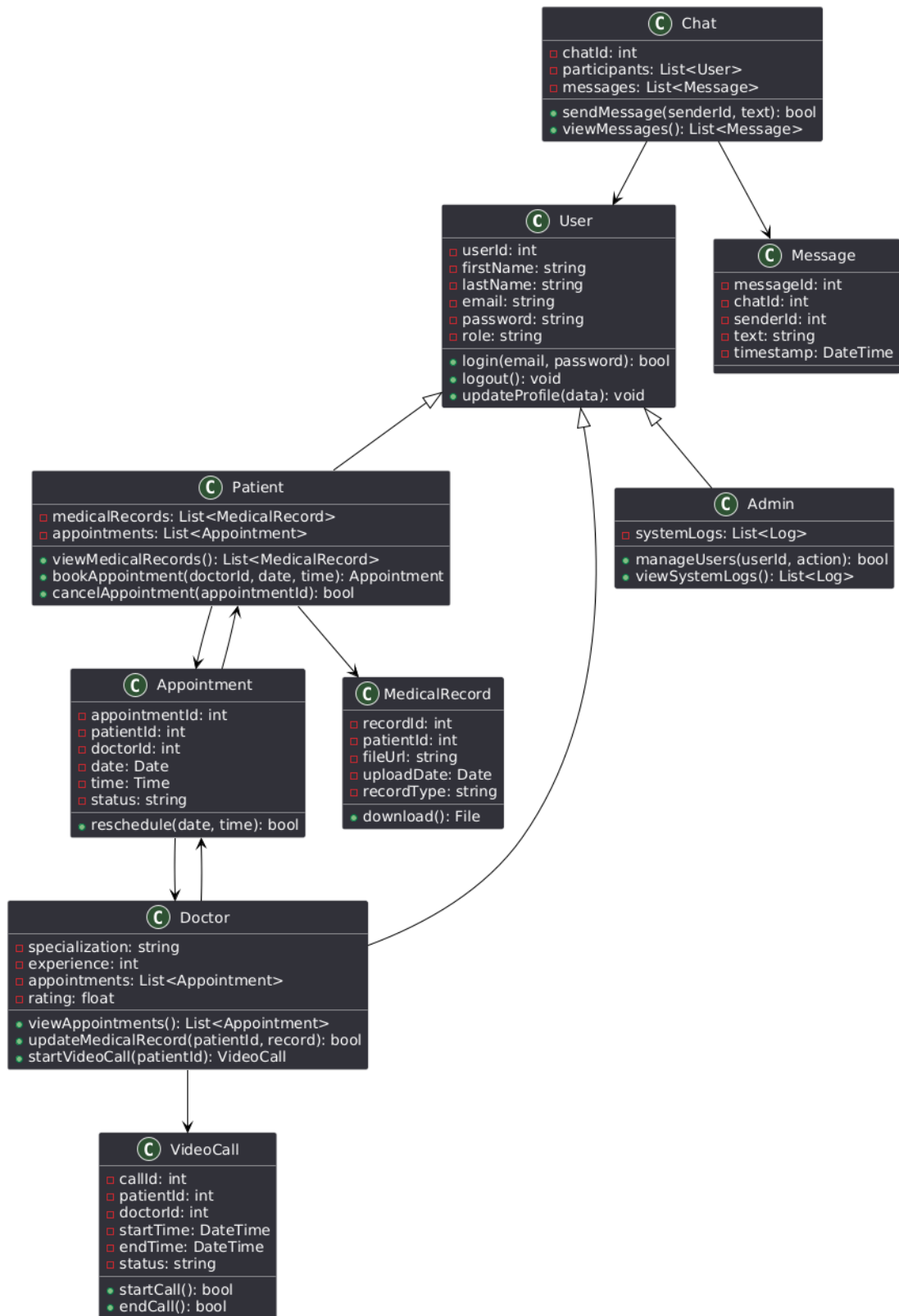


Рисунок 3.12 - Діаграма діяльності

На представленій UML-діаграмі діяльності зображено основні сценарії взаємодії користувача з телемедичною системою:

- Початкові дії: Користувач заходить на веб-сайт і може обрати реєстрацію нового облікового запису або авторизацію. Верифікація контактної інформації забезпечує безпеку та достовірність даних.
- Огляд послуг: Після авторизації користувач має доступ до доступних послуг, таких як запис на консультацію, тестування або відеозв'язок із лікарем.
- Вибір формату консультації: Користувач може обрати між телемедичною консультацією та особистим візитом до лікарні. Для обох випадків реалізовано процес оплати послуг і надання доступу до консультації.
- Консультація з лікарем: Для телемедичного варіанту користувач долучається до відеоконференції, де лікар проводить огляд медичної історії та надає рекомендації чи рецепт. У випадку особистого візиту консультація проводиться безпосередньо в лікарні.
- Післяконсультаційні дії: Користувач отримує доступ до рецептів, результатів тестів або подальших рекомендацій. Важливим елементом є можливість залишити відгук про якість наданих послуг, що сприяє постійному вдосконаленню системи.

Ця діаграма діяльності допомагає зрозуміти логіку роботи системи, а також виявити потенційні точки для оптимізації процесів і покращення користувацького досвіду. Вона є основою для розробки інтуїтивно зрозумілого і ефективного веб-сайту з телемедичними функціями.

3.3 Опис структури проекту та основних функцій

У даному підрозділі описано структуру проекту веб-сайту лікарні, який реалізує функціонал телемедицини. На рисунку 3.10 представлена структура проекту, яка складається з ключових файлів і директорій, необхідних для забезпечення роботи системи.

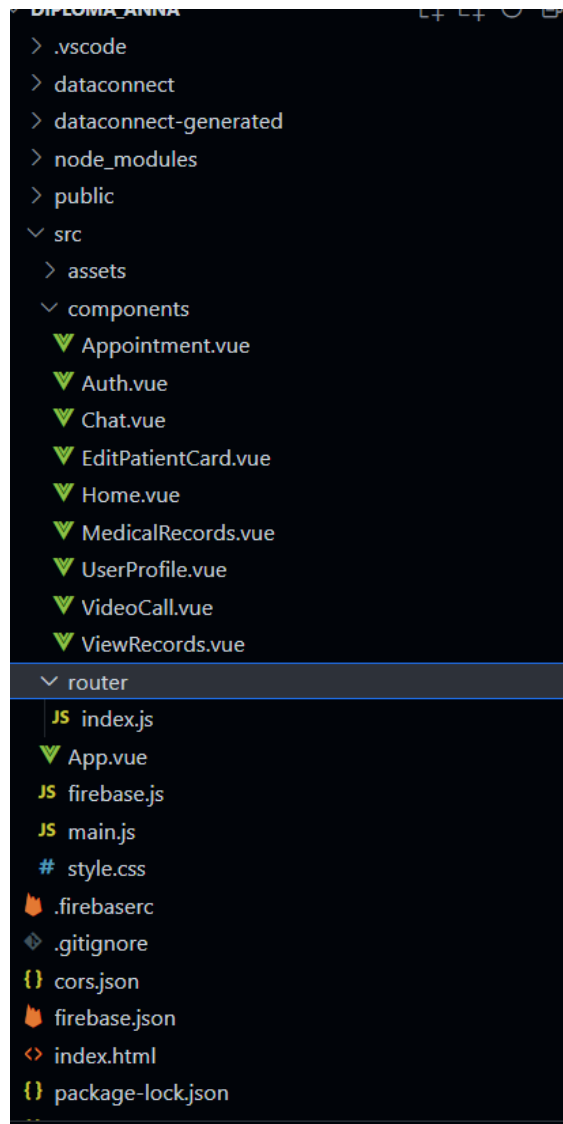


Рисунок 3.13 – Структура проекту

У кореневій папці проекту знаходяться основні конфігураційні файли, такі як `index.html`, `package-lock.json`, `firebase.json` і `cors.json`. Ці файли забезпечують налаштування проекту, зокрема підключення до Firebase, конфігурацію CORS і залежності проекту. Файл `.firebaseconfig` містить параметри для управління середовищами розгортання Firebase, наприклад, для розділення між тестовою та продуктивною версіями бази даних.

Директорія `src` є основною папкою для зберігання вихідного коду проекту. У ній містяться піддиректорії `assets` і `components`, а також файли для роботи з маршрутизацією, стилями та інтеграцією Firebase. На рисунку 3.11 представлена

реалізація інтеграції Firebase. У директорії `components` розміщено Vue-компоненти, кожен з яких відповідає за окремий функціонал системи. Наприклад, компонент `Appointment.vue` відповідає за роботу з записами на прийом, `Auth.vue` – за авторизацію. На рисунку 3.12 представлена реалізація компоненту `Auth.vue`. `VideoCall.vue` реалізує функціонал відеодзвінків між пацієнтами та лікарями [23], а `EditPatientCard.vue` дозволяє лікарям редагувати медичні картки пацієнтів. На рисунку 3.13 та 3.14 представлена реалізація компоненту `VideoCall.vue`. Компонент `Chat.vue` надає можливість обміну повідомленнями між лікарями та пацієнтами, використовуючи вбудовані механізми Firebase для реального часу [24].

```
src > JS firebase.js > ...
1  import { initializeApp } from "firebase/app";
2  import { getAuth } from "firebase/auth";
3  import { getFirestore } from "firebase/firestore";
4  import { getStorage } from "firebase/storage";
5
6  const firebaseConfig = {
7    apiKey: "AIzaSyA_UPTL5hMOYCGGNKiL0urjue82xlBA218",
8    authDomain: "anna-diploma.firebaseio.com",
9    projectId: "anna-diploma",
10   storageBucket: "anna-diploma.appspot.com",
11   messagingSenderId: "2236305352",
12   appId: "1:2236305352:web:373e0db4a106fea5803afd",
13 };
14
15 const app = initializeApp(firebaseConfig);
16
17 const auth = getAuth(app);
18 const db = getFirestore(app);
19 const storage = getStorage(app);
20
21 export { auth, db, storage };|
```

Рисунок 3.14 – підключення firebase

```

async handleSubmit() {
  try {
    if (this.isRegister) {
      await createUserWithEmailAndPassword(auth, this.email, this.password);
    } else {
      await signInWithEmailAndPassword(auth, this.email, this.password);
    }
    router.push("/");
  } catch (error) {
    this.toast.error(error.message);
    console.error("Помилка:", error.message);
  }
},
toggleMode() {
  this.isRegister = !this.isRegister;
},
},
},

```

Рисунок 3.15 – реалізація авторизації

```

async createPeer() {
  this.peer = new Peer();

  this.peer.on("open", (id) => {
    this.peerId = id;
    console.log("Ваш Peer ID:", id);
  });

  this.peer.on("call", async (incomingCall) => {
    console.log("Вхідний дзвінок");

    this.localStream = await navigator.mediaDevices.getUserMedia({
      video: true,
      audio: true,
    });

    this.$refs.localVideo.srcObject = this.localStream;

    incomingCall.answer(this.localStream);

    incomingCall.on("stream", (remoteStream) => {
      this.remoteStream = remoteStream;
      this.$refs.remoteVideo.srcObject = remoteStream;
    });

    this.call = incomingCall;
    this.isCalling = true;
  });
},

```

Рисунок 3.16 – реалізація відеодзвінків. Частина 1

```

    async connectToPeer() {
      if (!this.peerIdToConnect) {
        alert("Будь ласка, введіть Peer ID для підключення");
        return;
      }

      this.localStream = await navigator.mediaDevices.getUserMedia({
        video: true,
        audio: true,
      });

      this.$refs.localVideo.srcObject = this.localStream;

      const outgoingCall = this.peer.call(this.peerIdToConnect, this.localStream);

      outgoingCall.on("stream", (remoteStream) => {
        this.remoteStream = remoteStream;
        this.$refs.remoteVideo.srcObject = remoteStream;
      });

      outgoingCall.on("close", () => {
        console.log("Дзвінок завершено");
        this.isCalling = false;
      });

      this.call = outgoingCall;
      this.isCalling = true;
    },
  },

```

Рисунок 3.17 – реалізація відеодзвінків. Частина 2

Для маршрутизації в проєкті використовується директорія `router`, де файл `index.js` визначає маршрути для навігації між компонентами, такими як `Home.vue`, `MedicalRecords.vue` та `Chat.vue`. Ця структура дозволяє динамічно керувати інтерфейсом користувача залежно від ролі (пацієнт чи лікар) і надає доступ до відповідних функцій. На рисунку 3.15 представлена реалізація медичних записів.

Файл `firebase.js` відповідає за підключення до бази даних Firebase, яка використовується для зберігання даних користувачів, записів на прийом, медичних карток і повідомлень у чаті. Firebase забезпечує інтеграцію з системою аутентифікації, дозволяючи захистити доступ до приватних даних. У файлі реалізовано логіку ініціалізації Firebase, зокрема Firestore, Authentication та Storage [25].

Файл `main.js` є точкою входу для ініціалізації програми, де відбувається підключення до Vue Router, глобальних компонентів та Firebase. Він також відповідає за підключення до системи обробки стану.

```
handleFileUpload(event) {
  this.selectedFile = event.target.files[0];
},
uploadFile() {
  if (!this.selectedFile) return;

  const reader = new FileReader();
  reader.readAsDataURL(this.selectedFile);

  reader.onload = () => {
    const fileData = {
      name: this.selectedFile.name,
      content: reader.result,
    };

    const transaction = this.db.transaction(["files"], "readwrite");
    const store = transaction.objectStore("files");
    store.add(fileData);

    transaction.oncomplete = () => {
      alert("Файл успішно збережено!");
      this.loadFiles();
    };

    transaction.onerror = (event) => {
      console.error("Помилка збереження файлу:", event.target.error);
    };
  };
},
```

Рисунок 3.18 – реалізація медичних записів

Для стилізації проекту використовується файл `style.css`, у якому містяться основні правила оформлення інтерфейсу. Цей файл забезпечує єдиний стиль для всіх компонентів, створюючи приємний користувацький досвід і адаптивність для різних пристроїв.

Особливу увагу приділено структурі директорії `assets`, яка може містити статичні файли, такі як іконки, зображення та додаткові стилі. Це дозволяє чітко відокремлювати статичні ресурси від компонентів і зберігати їх в одному місці.

Для забезпечення безпеки та стабільності роботи проекту використовуються файли `.gitignore` для виключення конфіденційних або тимчасових файлів із системи контролю версій. Наприклад, у `.gitignore` виключаються ключі доступу до Firebase або файли налаштувань, які не повинні потрапляти до репозиторію.

На рисунку 3.16 представлено інтерфейс компонента авторизації, що відповідає за доступ користувачів до системи. Він виконаний у простому та зрозумілому стилі для забезпечення зручності використання.

Авторизація / Реєстрація

Зареєструватися Увійти

test@gmail.com

.....

Зареєструватися

Вже є акаунт? Увійти

Рисунок 3.19 – інтерфейс компонента авторизації

Рисунок 3.17 демонструє компонент для відеодзвінків, який забезпечує онлайн-комунікацію між користувачами. Дизайн компонента передбачає інтуїтивний набір елементів для керування дзвінком.

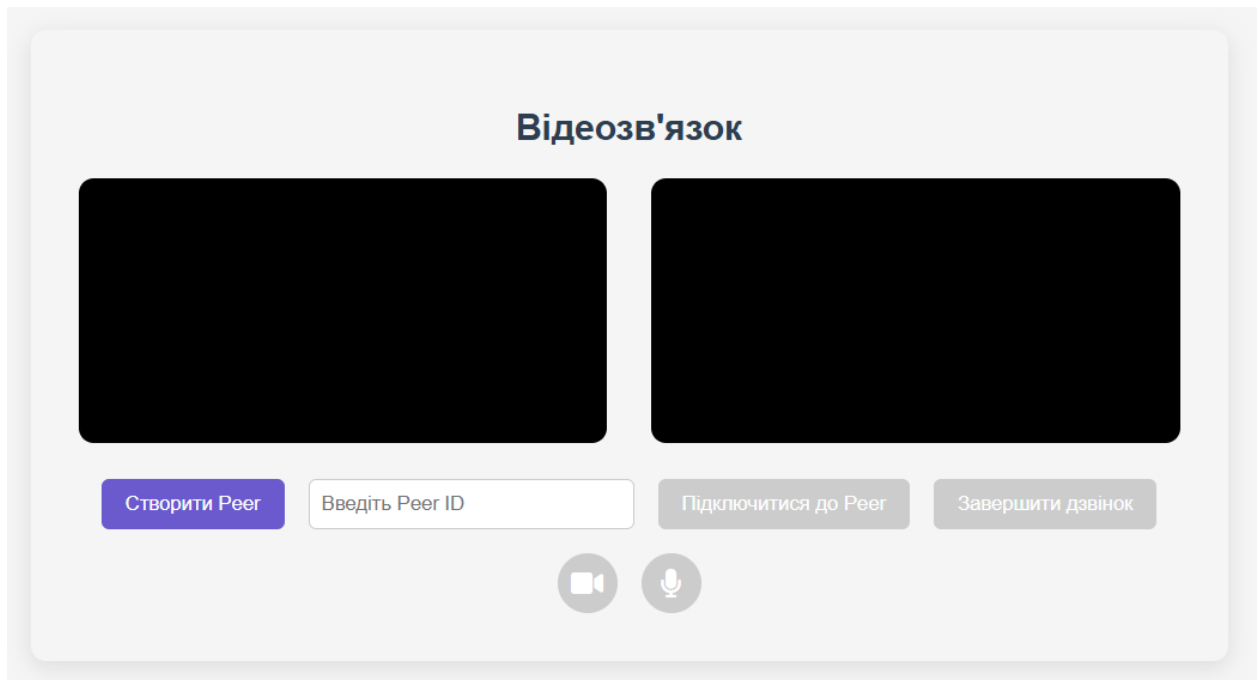


Рисунок 3.20 – інтерфейс компонента відеодзвінків

На рисунку 3.18 зображено вигляд компонента медичних записів. Інтерфейс дозволяє користувачам зручно працювати з необхідною інформацією, зберігаючи її структурованість та доступність.

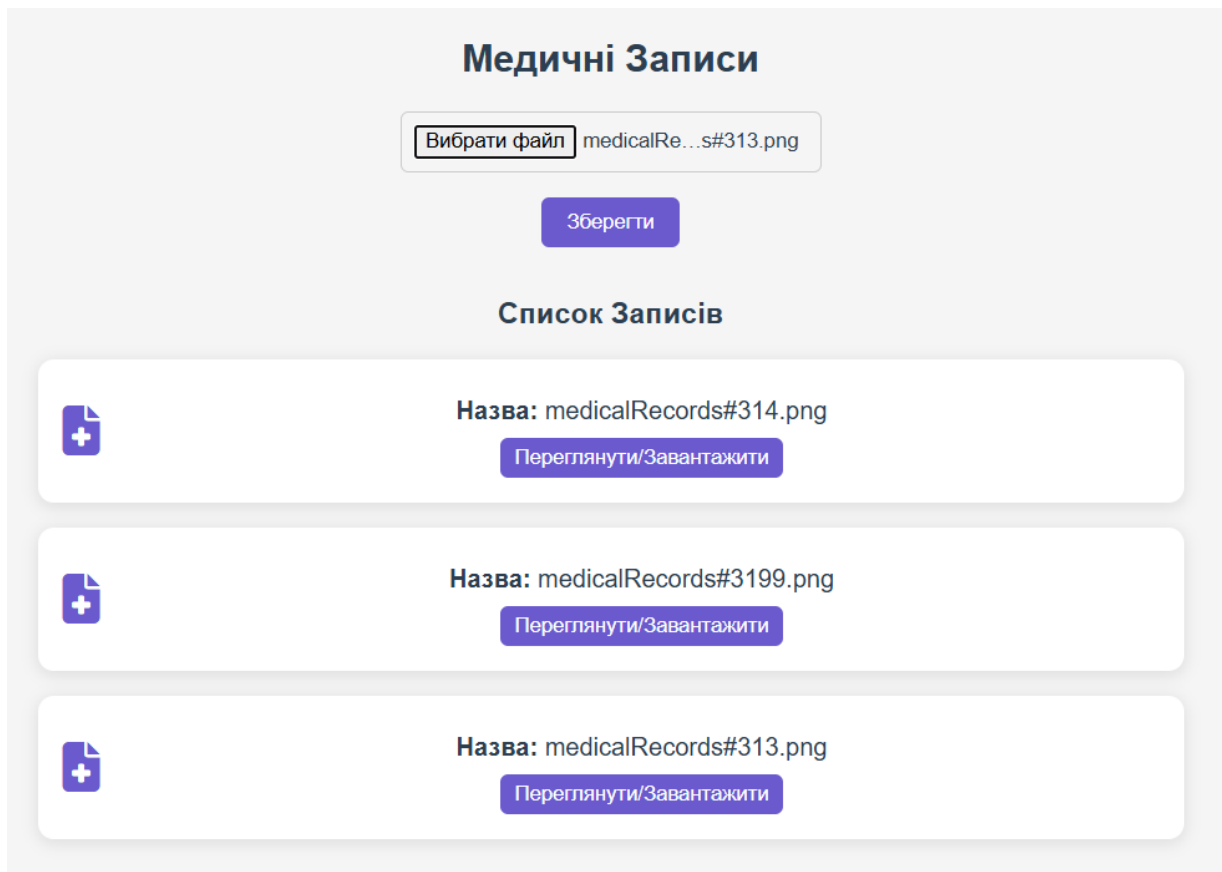


Рисунок 3.21 – Інтерфейс компонента медичних записів

Загалом, структура проекту є чітко організованою, забезпечуючи ефективну інтеграцію між компонентами, базою даних і маршрутизацією. Такий підхід дозволяє легко розширювати функціонал і підтримувати проект у довгостроковій перспективі.

3.4 Тестування

Тестування програмного забезпечення є ключовим етапом розробки системи, що дозволяє оцінити її відповідність функціональним вимогам, стабільність роботи та загальну якість. Основна мета тестування полягає у виявленні потенційних помилок, перевірці коректності виконання функцій та забезпеченні відповідності очікуванням користувачів.

У процесі тестування веб-сайту лікарні з використанням телемедицини було застосовано кілька підходів, кожен із яких забезпечує перевірку певних аспектів системи [26].

Функціональне тестування орієнтоване на перевірку ключових функцій системи, таких як авторизація користувачів, запис на прийом до лікаря, завантаження та збереження медичних карток, а також функціонал відеодзвінків. Основною метою цього типу тестування є переконатися, що система виконує свої функції відповідно до визначених вимог.

Нефункціональне тестування зосереджене на таких аспектах, як продуктивність, масштабованість і стабільність системи. Наприклад, під час тестування було оцінено швидкість завантаження сторінок і обробки запитів до бази даних Firebase, а також перевірено, як система реагує на збільшення кількості одночасних користувачів.

Тестування безпеки включало перевірку правильності обробки конфіденційної інформації, такої як медичні записи пацієнтів та облікові дані. Особливу увагу приділено валідації даних користувачів під час авторизації та захисту від потенційних атак, таких як SQL-ін'єкції чи міжсайтовий скриптинг (XSS).

Позитивне тестування забезпечило перевірку функціоналу системи за умови очікуваних та коректних вхідних даних, таких як правильний формат електронної пошти чи валідні дати запису на прийом. Негативне тестування, у свою чергу, дало змогу перевірити реакцію системи на некоректні дані, наприклад, невірний формат файлів, введення порожніх полів чи неправильні облікові дані під час авторизації.

Для кожного функціонального модуля системи було розроблено низку тест-кейсів. Компонент "Відеодзвінки" було протестовано на якість з'єднання, можливість завершення дзвінка та обробку ситуацій втрати з'єднання.

Усі тест-кейси були спрямовані на забезпечення стабільної та безпечної роботи веб-сайту, що є критично важливим для системи, яка оперує чутливою

медичною інформацією. Таблиця 3.2 містить перелік тест-кейсів, які використовувалися для перевірки функціоналу системи.

Таблиця 3.2 – Тест-кейси функціонального тестування веб-сайту

№	Тест-кейс	Результат
1	Перевірка авторизації користувача	+
	<i>Передумови</i> Користувач повинен мати зареєстрований обліковий запис	
	<i>Кроки</i> 1. Відкрити сторінку авторизації 2. Ввести правильні email і пароль 3. Натиснути кнопку "Увійти"	
	<i>Очікуваний результат</i> Користувач успішно авторизується та перенаправляється на головну сторінку	
2	Перевірка авторизації з некоректними даними	+
	<i>Передумови</i> Користувач не повинен мати зареєстрованого облікового запису	
	<i>Кроки</i> 1. Відкрити сторінку авторизації 2. Ввести некоректний email і пароль 3. Натиснути кнопку "Увійти"	
	<i>Очікуваний результат</i> Відображається повідомлення про помилку: "Некоректні дані для входу"	
3	Створення запису на прийом до лікаря	+
	<i>Передумови</i> Користувач успішно авторизований	
	<i>Кроки</i> 1. Перейти до сторінки запису на прийом 2. Вибрати лікаря зі списку 3. Вибрати дату та час 4. Натиснути кнопку "Підтвердити"	
	<i>Очікуваний результат</i> Запис на прийом успішно створено, з'являється підтвердження: "Ваш запис створено"	
4	Завантаження медичної картки	+
	<i>Передумови</i> Користувач успішно авторизований	
	<i>Кроки</i> 1. Перейти до розділу "Медичні картки" 2. Натиснути кнопку "Завантажити файл" 3. Вибрати файл для завантаження 4. Натиснути "Зберегти"	

	<i>Очікуваний результат</i> Медична картка успішно збережена, з'являється повідомлення: "Документ завантажено"	
5	Перевірка відеодзвінка <i>Передумови</i> Користувач і лікар обидва авторизовані та мають доступ до чату <i>Кроки</i> 1. Перейти до чату 2. Натиснути кнопку "Почати дзвінок" 3. Перевірити якість зв'язку та можливість завершення дзвінка <i>Очікуваний результат</i> Дзвінок успішно ініційовано, з'являється повідомлення про підключення. Можливість завершити дзвінок без помилок	+
6	Перевірка роботи чату <i>Передумови</i> Користувач і лікар обидва авторизовані та мають доступ до чату <i>Кроки</i> 1. Відкрити чат 2. Ввести повідомлення 3. Натиснути кнопку "Відправити" <i>Очікуваний результат</i> Повідомлення успішно доставлено, відображається в чаті обох сторін	+
7	Видалення запису на прийом <i>Передумови</i> Користувач успішно авторизований і має хоча б один запис на прийом <i>Кроки</i> 1. Перейти до розділу "Мої записи" 2. Вибрати запис на прийом зі списку 3. Натиснути кнопку "Скасувати запис" 4. Підтвердити дію у спливаючому вікні <i>Очікуваний результат</i> Запис успішно видалено, відображається повідомлення: "Ваш запис скасовано"	+
8	Завантаження файлу неправильного формату <i>Передумови</i> Користувач успішно авторизований <i>Кроки</i> 1. Перейти до розділу "Медичні картки" 2. Натиснути кнопку "Завантажити файл" 3. Вибрати файл із недопустимим форматом (наприклад, .exe) 4. Натиснути "Зберегти" <i>Очікуваний результат</i> Відображається повідомлення про помилку: "Непідтримуваний формат файлу. Завантажуйте файли у форматах .pdf, .jpg, .png"	+

Отримані результати тестування підтверджують, що веб-сайт лікарні з функціоналом телемедицини повністю відповідає визначеним вимогам. Протягом тестування було перевірено різноманітні сценарії використання, включаючи роботу компонентів авторизації, запису на прийом, завантаження медичних карток, чату та відеодзвінків [27]. Система продемонструвала високу стабільність, коректність і відповідність очікуванням користувачів, що підтверджує її готовність до впровадження у практичне використання.

Розглянемо приклад виконання тест-кейсу №2 – авторизація з неправильними даними. Для цього користувач переходить на сторінку авторизації та вводить невірні дані, наприклад, некоректну комбінацію email та пароля. Після натискання на кнопку "Увійти" система успішно обробляє запит і виводить повідомлення про помилку: "Некоректні дані для входу". Цей результат відображений на рисунку 3.19.

Такий тест дозволяє оцінити надійність механізмів валідації вхідних даних та перевірку правильності облікових записів. Усі помилкові запити були коректно оброблені, а повідомлення про помилку успішно відображалися користувачеві. Таким чином, система показала високу ефективність у запобіганні несанкціонованому доступу, що є критично важливим для забезпечення безпеки даних.

Висновки щодо виконання цього та інших тест-кейсів підтверджують, що система відповідає заявленим вимогам і готова до використання у реальних умовах.

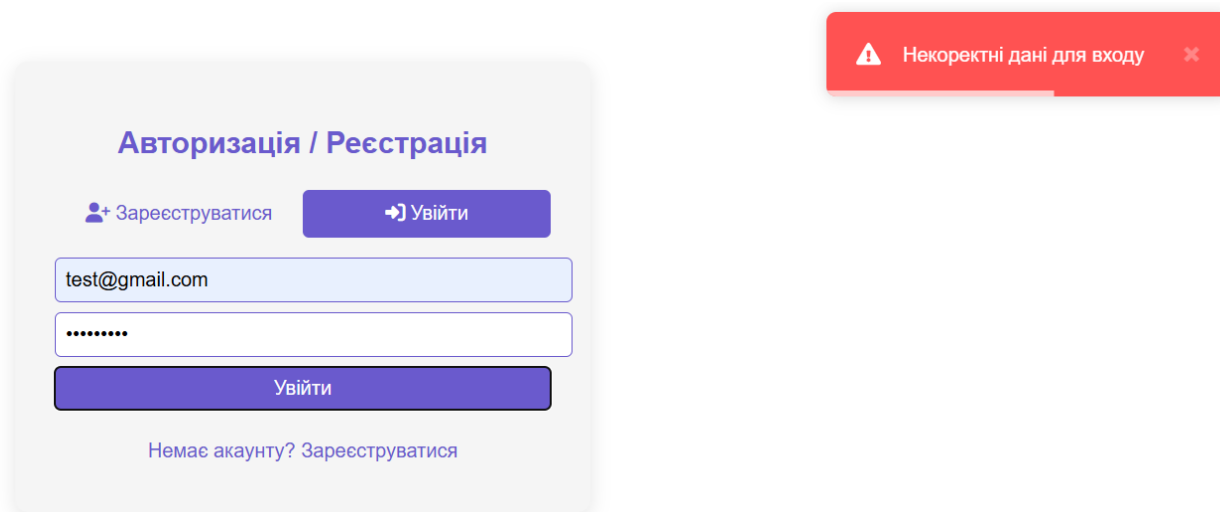


Рисунок 3.22 – Сповіщення про отриману помилку

3.5 Висновки до розділу

У третьому розділі дипломної роботи було детально висвітлено процес проектування та реалізації веб-сайту лікарні з використанням телемедицини. Основну увагу зосереджено на функціональних вимогах, проектуванні структури бази даних, архітектурі проекту та описі ключових компонентів системи, які забезпечують виконання основних функцій.

Розроблена структура бази даних включає взаємопов'язані сутності, які підтримують зберігання медичних записів, управління записами на прийом, інтеграцію відеодзвінків та обмін повідомленнями між пацієнтами і лікарями. Інтеграція з Firebase забезпечує безпечне та ефективне зберігання даних, а використання Vue.js дозволяє реалізувати модульність системи, що спрощує її розвиток і підтримку.

У рамках роботи було виконано тестування розробленої системи для перевірки її функціональності та відповідності поставленим вимогам. Зокрема, проведено тестування компонентів авторизації, роботи з медичними записами, запису на прийом, чату та відеодзвінків. Наприклад, тест-кейс авторизації з

неправильними даними продемонстрував, що система коректно обробляє помилки, відображаючи відповідне повідомлення користувачу, забезпечуючи тим самим надійність і безпеку доступу до даних. Результати тестування підтвердили, що всі основні функції працюють стабільно та відповідають вимогам до системи.

Представлені діаграми ілюструють взаємодію між компонентами системи, що забезпечує прозорість логіки її роботи. Гнучка архітектура проекту дозволяє легко розширювати функціональність, зокрема додавати нові модулі, такі як інструменти аналітики стану здоров'я пацієнтів або вдосконалені системи аутентифікації для посилення захисту даних. Діаграми та блоксхеми розробленого веб-сайту наведені у додатку Б – Графічна частина.

Розроблений веб-сайт лікарні є сучасним і зручним інструментом, який сприяє підвищенню доступності медичних послуг, покращенню взаємодії між пацієнтами та лікарями, а також ефективності надання медичної допомоги. Система відповідає вимогам сучасної телемедицини та має потенціал для подальшого розвитку.

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота «Розробка веб-сайту лікарні з використанням телемедицини» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Розробка веб-сайту лікарні з використанням телемедицини» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [28].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі,
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3

	Бали:		
1. Технічна здійсненність концепції	5	5	5
2. Ринкові переваги (наявність аналогів)	2	2	3
3. Ринкові переваги (ціна продукту)	3	3	3
4. Ринкові переваги (технічні властивості)	3	3	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	3
6. Ринкові перспективи (розмір ринку)	3	3	3
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	3	4	4
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	3
Сума балів	36	39	39
Середньоарифметична сума балів CB_c	38,0		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [28].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів CB розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка веб-сайту лікарні з використанням телемедицини» становить 38,0 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення розрахуємо за формулою [29]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (4.1)$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу i -го технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом

і при цьому має виконуватись умова $\sum_{i=1}^k \alpha_i = 1$;

β_i – відносне значення i -го технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ni}}{I_{ai}}, \quad (4.2)$$

де I_{ni} та I_{ai} – чисельні значення конкретного i -го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ni}}; \quad (4.3)$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірювання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Інтерфейс користувача	бал	8	9	1,13	0,25
Безпека передачі даних	%	90	95	1,06	0,15
Час завантаження сторінки	с	4	3	1,33	0,25
Доступність функціоналу для пацієнтів	%	85	95	1,12	0,25
Рівень інтеграції з системами	бал	8	9,5	1,19	0,1

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 1,13 \cdot 0,25 + 1,06 \cdot 0,15 + 1,33 \cdot 0,25 + 1,12 \cdot 0,25 + 1,19 \cdot 0,1 = 1,17.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,17 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Розробка веб-сайту лікарні з використанням телемедицини», під час

планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_o$) розраховуємо у відповідності до посадових окладів працівників, за формулою [29]:

$$З_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.4)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=22$ дні.

$$З_o = 18200,00 \cdot 12 / 22 = 9927,27 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник-координатор проекту	18200,00	827,27	12	9927,27

Інженер-розробник програмного забезпечення формування веб-сайту	17800,00	809,09	32	25890,91
Консультант (лікар-терапевт вищої категорії)	15000,00	681,82	5	3409,09
Лаборант	8050,00	365,91	21	7684,09
Всього				46911,36

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_p$) за відповідними найменуваннями робіт НДР на тему «Розробка веб-сайту лікарні з використанням телемедицини» розраховуємо за формулою:

$$З_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.5)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (4.6)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo $M_M=8000,00$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [30];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 22$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_1 = 8000,00 \cdot 1,10 \cdot 1,15 / (22 \cdot 8) = 57,50 \text{ грн.}$$

$$З_{pl} = 57,50 \cdot 8,00 = 460,00 \text{ грн.}$$

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Підготовка робочого місця інженера-розробника програмного забезпечення	8,00	2	1,10	57,50	460,00
Інсталяція програмного забезпечення середовища розробки	5,00	3	1,35	70,57	352,84
Компіляція блоків веб-сайту	7,00	4	1,50	78,41	548,86
Підготовка локального серверного обладнання для проведення тестування працездатності веб-сайту	6,00	5	1,70	88,86	533,18
Налагодження програмних блоків збору інформації та координації діяльності	4,50	5	1,70	88,86	399,89
Тестування працездатності програмного забезпечення	10,00	2	1,10	57,50	575,00
Всього					2869,77

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{доп}} = (Z_o + Z_p) \cdot \frac{H_{\text{доп}}}{100\%}, \quad (4.7)$$

де $H_{\text{доп}}$ – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$Z_{\text{доп}} = (46911,36 + 2869,77) \cdot 11 / 100\% = 5475,93 \text{ грн.}$$

4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{доп}}) \cdot \frac{H_{\text{зн}}}{100\%} \quad (4.8)$$

де $H_{\text{зн}}$ – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (46911,36 + 2869,77 + 5475,93) \cdot 22 / 100\% = 12156,55 \text{ грн.}$$

4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Розробка веб-сайту лікарні з використанням телемедицини».

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{ej}, \quad (4.9)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

C_{ej} – вартість відходів j -го найменування, грн/кг.

$$M_1 = 4,0 \cdot 195,00 \cdot 1,1 - 0 \cdot 0 = 858,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір Ultra Plus	195,00	4,0	0	0	858,00
Папір для записів Parers Light A5	114,00	2,0	0	0	250,80
Органайзер офісний OFFICE	194,00	1,0	0	0	213,40
Канцелярське приладдя (набір офісного працівника)	172,00	4,0	0	0	756,80
Картридж для принтера Erixon EZ2500	920,00	2,0	0	0	2024,00
Диск оптичний New CD-RW	28,00	3,0	0	0	92,40
Flesh-пам'ять King 64 GB	144,00	2,0	0	0	316,80

Тека для паперів ВОХ	105,00	5,0	0	0	577,50
Всього					5089,70

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_e), які використовують при проведенні НДР на тему «Розробка веб-сайту лікарні з використанням телемедицини» відсутні.

4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.і}} \cdot K_i, \quad (4.10)$$

де C_i – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{\text{пр.і}}$ – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань устаткування.

$$B_{\text{спец}} = 2599,00 \cdot 1 \cdot 1,1 = 2858,90 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.8 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Маршрутизатор інтернет WiFi6 TP-Link Archer AX23	1	2599,00	2858,90
Всього			2858,90

4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{\text{прог}} = \sum_{i=1}^k C_{\text{прог}} \cdot C_{\text{прог.і}} \cdot K_i, \quad (4.11)$$

де $C_{\text{прог}}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прог.і}}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань програмних засобів.

$$B_{\text{прог}} = 40,38 \cdot 1 \cdot 1,1 = 44,42 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 4.9 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Середовище розробки Visual Studio	1	40,38	44,42

Хмарний хостинг Cloud Professional для отримання даних (доступ, місяць)	2	560,00	1232,00
Абонентна плата доступу до мережі Internet (за місяць)	2	400,00	880,00
Collaboratory	1	40,38	44,42
Підтримка домену (місяць)	2	40,38	88,84
VDS (Virtual Dedic Server) Сервер для середовища розробки та тестування	2	3450,00	7590,00
Всього			9879,67

4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.12)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (22599,00 \cdot 2) / (2 \cdot 12) = 1883,25 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.10 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
-------------------------	-------------------------	-------------------------------------	---	---------------------------------

Персональний комп'ютер розробника програмного забезпечення (HP Laptop 15seq2053ua)	22599,00	2	2	1883,25
Робоче місце розробника ПЗ	8415,00	5	2	280,50
Пристрої виводу інформації	7430,00	4	2	309,58
Оргтехніка	7050,00	5	2	235,00
Приміщення лабораторії розробки	415000,00	30	2	2305,56
Операційна система Microsoft Windows 10 Home	4200,00	3	2	233,33
Microsoft Office 2016	4800,00	3	2	266,67
Електронно-обчислювальний комплекс обробки даних на базі QUBE QB i9 10900KF RTX 3060 TI 8GB 1642	59699,00	4	2	2487,46
Всього				8001,35

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{gni}}{\eta_i}, \quad (4.13)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

Π_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $\Pi_e = 10,98$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,42 \cdot 240,0 \cdot 10,98 \cdot 0,95 / 0,97 = 1106,78 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.11 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер розробника програмного забезпечення (HP Laptop 15seq2053ua)	0,42	240,0	1106,78
Робоче місце розробника ПЗ	0,36	240,0	948,67
Пристрої виводу інформації	0,12	20,0	26,35
Електронно-обчислювальний комплекс обробки даних на базі QUBE QB i9 10900KF RTX 3060 TI 8GB 1642	0,10	160,0	175,68
Оргтехніка	0,56	2,5	15,37
Маршрутизатор інтернет WiFi6 TP-Link Archer AX23	0,03	240,0	79,06
Всього			2351,92

4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Розробка веб-сайту лікарні з використанням телемедицини» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням

досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (4.14)$$

де H_{cv} – норма нарахування за статтею «Службові відрядження», приймемо $H_{cv} = 20\%$.

$$B_{cv} = (46911,36 + 2869,77) \cdot 20 / 100\% = 9956,23 \text{ грн.}$$

4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» відсутні.

4.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ie}}{100\%}, \quad (4.15)$$

де H_{ie} – норма нарахування за статтею «Інші витрати», приймемо $H_{ie} = 50\%$.

$$I_e = (46911,36 + 2869,77) \cdot 50 / 100\% = 24890,57 \text{ грн.}$$

4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (З_o + З_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.16)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{нзв} = 100\%$.

$$B_{нзв} = (46911,36 + 2869,77) \cdot 100 / 100\% = 49781,14 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Розробка веб-сайту лікарні з використанням телемедицини» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_o + З_p + З_{доо} + З_n + M + K_v + B_{спец} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сп} + I_v + B_{нзв}. \quad (4.17)$$

$$B_{заг} = 46911,36 + 2869,77 + 5475,93 + 12156,55 + 5089,70 + 0,00 + 2858,90 + 9879,67 + 8001,35 + 2351,92 + 9956,23 + 0,00 + 24890,57 + 49781,14 = 180223,08 \text{ грн.}$$

Загальні витрати $ЗВ$ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.18)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta = 0,9$.

$$ЗВ = 180223,08 / 0,9 = 200247,87 \text{ грн.}$$

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Розробка веб-сайту лікарні з використанням телемедицини» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	200	400	500	300

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, прийmemo 20000 осіб;

C_0 – вартість послуги у році до впровадження інформаційної системи, прийmemo 3000,00 грн;

$\pm \Delta C_0$ – зміна вартості послуги від впровадження результатів, прийmemo 419,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [28]:

$$\Delta\Pi_i = (\pm\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.19)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).
Прийmemo $\rho = 40\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (419,00 \cdot 20000,00 + 3419,00 \cdot 200) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 2467528,91 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (419,00 \cdot 20000,00 + 3419,00 \cdot 600) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 2839844,34 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (419,00 \cdot 20000,00 + 3419,00 \cdot 1100) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 3305238,62 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (419,00 \cdot 20000,00 + 3419,00 \cdot 1400) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 3584475,18 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків $\Pi\Pi$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^i}, \quad (4.20)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau=0,15$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} PPP &= 2467528,91/(1+0,15)^1 + 2839844,34/(1+0,15)^2 + 3305238,62/(1+0,15)^3 + \\ &+ 3584475,18/(1+0,15)^4 = 2145677,31 + 2147330,31 + 2173248,04 + 2049435,32 = \\ &= 8515690,99 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, \quad (4.21)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв}=2$;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 200247,87 грн.

$$PV = k_{инв} \cdot 3B = 2 \cdot 200247,87 = 400495,74 \text{ грн.}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = PPP - PV \quad (4.22)$$

де PPP – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 8515690,99 грн;

PV – теперішня вартість початкових інвестицій, 400495,74 грн.

$$E_{abc} = PPP - PV = 8515690,99 - 400495,74 = 8115195,25 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_g = \sqrt[T_{жс}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.23)$$

де E_{abc} – абсолютний економічний ефект вкладених інвестицій, 8115195,25 грн;

PV – теперішня вартість початкових інвестицій, 400495,74 грн;

$T_{жс}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$$E_g = \sqrt[T_{жс}]{1 + \frac{E_{abc}}{PV}} - 1 = (1 + 8115195,25/400495,74)^{1/4} = 1,15.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій τ_{min} :

$$\tau_{min} = d + f, \quad (4.24)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = 0,09$;

f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,15.

$\tau_{min} = 0,09 + 0,15 = 0,24 < 1,15$ свідчить про те, що внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Розробка веб-сайту лікарні з використанням телемедицини» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_{\epsilon}}, \quad (4.25)$$

де E_{ϵ} – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 1,15 = 0,87 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.5 Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка веб-сайту лікарні з використанням телемедицини» становить 38,0 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,17 рази.

Також термін окупності становить 0,87 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Розробка веб-сайту лікарні з використанням телемедицини».

ВИСНОВКИ

Магістерська кваліфікаційна робота присвячена розробці веб-сайту лікарні з інтеграцією телемедичних функцій. У процесі виконання роботи були проведені дослідження предметної галузі, обґрунтований вибір технологій для реалізації системи, здійснена розробка веб-сайту та його тестування, а також оцінена економічна ефективність впровадження розробки.

У першому розділі здійснено аналіз предметної галузі та розглянуто існуючі аналоги телемедичних платформ, такі як Mayo Clinic, Teladoc Health, Amwell, DOXY.me та локальні українські рішення. Було виявлено переваги та недоліки цих платформ, що дозволило визначити унікальні особливості запропонованого веб-сайту. Основна увага зосереджена на адаптації системи до потреб локального ринку, зокрема в частині доступності, інтеграції та функціональності.

Другий розділ присвячено вибору технологічних засобів, які забезпечують ефективну реалізацію проєкту. Для фронтенду було обрано HTML, CSS, JavaScript та Vue.js, які забезпечують гнучкість, масштабованість і зручність у розробці. Firebase було інтегровано для управління базою даних і аутентифікацією, що гарантує безпеку даних. PeerJS обрано для організації відеоконференцій, що дозволяє реалізувати сучасні телемедичні функції, зокрема відеоконсультації.

Третій розділ охоплює реалізацію функціоналу веб-сайту та його тестування. Реалізовано основні модулі, включаючи запис на прийом до лікаря, відеозв'язок, управління записами пацієнтів і моніторинг стану здоров'я. Тестування підтвердило стабільність роботи системи та її відповідність поставленим вимогам, а також зручність для пацієнтів і лікарів.

Обмеження на кількість записів на одного лікаря становить до 24 консультацій на день, що обумовлено 8-годинним робочим днем із розрахунком на 20 хвилин на одну онлайн-консультацію. Важливо зазначити, що онлайн-

консультація займає менше часу, ніж офлайн-прийом, який зазвичай триває щонайменше 30 хвилин. Це дозволяє підвищити ефективність роботи лікаря та приймати більше пацієнтів у порівнянні з традиційними методами.

Четвертий розділ спрямований на економічну оцінку розробки. Було виконано розрахунок витрат на реалізацію проєкту, проаналізовано термін окупності інвестицій, а також визначено комерційний потенціал системи. Результати свідчать про доцільність впровадження розробленого веб-сайту, який демонструє високу економічну ефективність завдяки мінімальним витратам на розробку та швидкому поверненню інвестицій.

Згідно з проведеними дослідженнями, рівень комерційного потенціалу розробки становить 38,0 бала, що вказує на її комерційну важливість (рівень потенціалу вище середнього). При оцінюванні за технічними параметрами науково-технічна розробка перевищує якість існуючих аналогів приблизно в 1,17 рази. Крім того, термін окупності проєкту становить 0,87 року, що значно менше від загальноприйнятого показника в 3 роки. Це свідчить про високу економічну доцільність розробки та її привабливість для потенційних інвесторів, які можуть профінансувати впровадження системи та її виведення на ринок.

Отже, проведена робота повністю відповідає поставленій меті та задачам. Розроблений веб-сайт лікарні з телемедичними функціями дозволяє забезпечити зручний і сучасний доступ до медичних послуг, підвищити ефективність надання медичної допомоги та сприяти розвитку телемедицини в Україні. Отримані результати можуть бути основою для подальших досліджень і вдосконалення телемедичних платформ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Олійниченко А. Б., Кулик Я. А. Веб-сайт лікарні з використанням телемедицини. *Молодь в науці: дослідження, проблеми, перспективи* : Молодь в науці: дослідж., проблеми, перспективи (МН-2025), м. Вінниця, 9 груд. 2024 р. 2024. URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/view/22808> (дата звернення: 09.12.2024).
2. What to Expect at This Year's WHO Executive Board Meeting. *unfoundation.org*. URL: https://unfoundation.org/blog/post/what-to-expect-at-this-years-who-executive-board-meeting/?gad_source=1&gclid=CjwKCAiAjeW6BhBAEiwAdKltMtpJ4lxV3nI1-ab0kZI2wjNhGiPr01wmTv_y9UfYjydHX8IJxI-3tRoCbSoQAvD_BwE (дата звернення: 06.11.2024).
3. Top-ranked Hospital in the Nation. *Mayo Clinic*. URL: <https://www.mayoclinic.org> (дата звернення: 16.10.2024).
4. Telehealth & Telemedicine Provider | Teladoc Health. *Telehealth & Telemedicine Provider* | Teladoc Health. URL: <https://www.teladochealth.com> (дата звернення: 15.10.2024).
5. Hybrid Care at Scale | Amwell. Hybrid Care at Scale | *Amwell*. URL: <https://www.amwell.com> (дата звернення: 12.10.2024).
6. The Simple, Free, and Secure Telemedicine Platform | Doxy.me. *Doxy.me*. URL: <https://www.doxy.me> (дата звернення: 09.10.2024).
7. Міністерство цифрової трансформації України. *Міністерство цифрової трансформації України*. URL: <https://thedigital.gov.ua> (дата звернення: 08.11.2024).
8. Market research reports, consulting: Global Market Insights Inc. *Global Market Insights Inc.* URL: <https://www.gminsights.com> (дата звернення: 12.10.2024).
9. Mozilla Developer Network. HTML Documentation. *MDN Web Docs*. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата звернення: 11.12.2024).

10. CSS Flexbox Layout Guide. *CSS-TRICKS*. URL: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/> (дата звернення: 17.09.2024).
11. *Bootstrap Official Website*. URL: <https://getbootstrap.com> (дата звернення: 14.10.2024).
12. Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. *Tailwind CSS - Rapidly build modern websites without ever leaving your HTML*. URL: <https://tailwindcss.com> (дата звернення: 14.11.2024).
13. ECMAScript® 2025 Language Specification. TC39 - *Specifying JavaScript*. URL: <https://tc39.es/ecma262/> (дата звернення: 06.11.2024).
14. Vue.js. *Vue.js - The Progressive JavaScript Framework* | Vue.js. URL: <https://vuejs.org> (дата звернення: 25.09.2024).
15. React. *React*. URL: <https://react.dev> (дата звернення: 30.10.2024).
16. Angular. Home • *Angular*. URL: <https://angular.io> (дата звернення: 11.12.2024).
17. Firebase Documentation. *Firebase*. URL: <https://firebase.google.com/docs> (дата звернення: 10.10.2024).
18. Full Stack Development - Web and Mobile Apps - AWS Amplify. *Amazon Web Services, Inc.* URL: <https://aws.amazon.com/amplify/> (дата звернення: 31.10.2024).
19. PeerJS - Simple peer-to-peer with WebRTC. *PeerJS - Simple peer-to-peer with WebRTC*. URL: <https://peerjs.com> (дата звернення: 05.12.2024).
20. Advanced Authentication Strategies in Web Applications. URL: <https://www.authstrategies.com/articles> (дата звернення: 11.12.2024).
21. *Telemedicine Innovations. Enhancing Communication with WebRTC*. URL: <https://telemedtech.org/webrtc-guide> (дата звернення: 11.12.2024).
22. *Advanced Data Security for Healthcare Systems*. URL: <https://securehealthdata.com/articles/security-practices> (дата звернення: 16.10.2024).
23. *Video Streaming in Real-Time Applications. WebRTC Insights*. URL: <https://videotech.io/webrtc> (дата звернення: 13.11.2024).

24. *Performance Optimization for Web Systems*. URL: <https://webperformanceguide.com/articles> (дата звернення: 11.12.2024).
25. *Healthcare Data Management Best Practices*. URL: <https://healthcaredata.com/resources> (дата звернення: 12.09.2024).
26. *Open Source Contributions to WebRTC Development*. URL: <https://opensourcewebrtc.org/resources> (дата звернення: 11.12.2024).
27. *Testing Strategies for Telemedicine Platforms*. URL: <https://telemedicinetesting.com/strategies> (дата звернення: 12.09.2024).
28. *Scalable Frontend Architectures for Modern Web Applications*. URL: <https://scalabilityfrontend.com/articles> (дата звернення: 25.09.2024).
29. Козловський В. О., Лесько О. Й., Кавецький В. В. 29. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. Вінниця : ВНТУ, 2021. – 42 с.
30. Козловський В. О., Причепі І. В. 30. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум. Вінниця : ВНТУ, 2015. – 113 с.

Додатки

Додаток А (обов'язковий)**ТЕХНІЧНЕ ЗАВДАННЯ**

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ

_____ д.т.н., проф. Олег

БІСІКАЛО

« 14 » жовтня 2024 року

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

ВЕБ-САЙТ ЛІКАРНІ З ВИКОРИСТАННЯМ ТЕЛЕМЕДИЦИНИ

08-31. МКР.005.02.000 ТЗ

Керівник: к.т.н., доц. каф. АІТ

_____ Ярослав КУЛИК

« 11 » жовтня 2024 р.

Розробив студент гр. ІСТ-22м

_____ Анна ОЛІЙНИЧЕНКО

« 11 » жовтня 2024 р.

1. Назва та галузь застосування.

Веб-сайт лікарні з використанням телемедицини.

2. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____2024р., та індивідуальне завдання на МКР, затверджене протоколом №__ засідання кафедри АІТ від «__» _____ 2024р.

3. Мета та призначення роботи.

Метою дослідження є розробка веб-сайту лікарні з інтеграцією функцій телемедицини, який забезпечить пацієнтам дистанційний доступ до медичних послуг, а лікарям — оптимізацію робочого процесу та зменшення адміністративного навантаження.

4. Джерела розробки:

1) World Health Organization. Telemedicine Opportunities and Developments in Member States [Електронний ресурс]. – Режим доступу: <https://www.who.int/telemedicines> (дата звернення: 09.10.2024).

2) Мінцифри України. Офіційний сайт про розвиток цифрових послуг в Україні [Електронний ресурс]. – Режим доступу: <https://thedigital.gov.ua> (дата звернення: 09.10.2024).

3) Global Market Insights. Telemedicine Market Size 2023 [Електронний ресурс]. – Режим доступу: <https://www.gminsights.com> (дата звернення: 09.10.2024).

4) Vue.js Official Guide [Електронний ресурс]. – Режим доступу: <https://vuejs.org> (дата звернення: 15.10.2024).

5) PeerJS Documentation [Електронний ресурс]. – Режим доступу: <https://peerjs.com> (дата звернення: 20.11.2024).

5. Показники призначення

Основні технічні вимоги та мінімальні системні вимоги до програми:

- Стабільний доступ до мережі Інтернет.
- Сучасний браузер із підтримкою HTML5 та WebRTC.
- ОС: Android 8+, iOS 12+, Windows 10+.
- Камера та мікрофон для відеоконсультацій.

Вихідні дані для проведення робіт:

API Firebase, документація PeerJS, стек технологій для розробки (Vue.js, JavaScript, CSS), обмеження на кількість записів на одного лікаря: до 24 консультацій на день (з урахуванням 20-хвилинного часу на консультацію в 8-годинному робочому дні).

Методи дослідження:

В даній роботі використовуються аналітичні методи, функціональне тестування та методи веб-розробки.

Результати роботи програми:

Система забезпечує запис пацієнтів, проведення відеоконсультацій, управління медичними записами та забезпечує стабільну роботу в реальному часі.

6. Економічні показники

До економічних показників входять:

- витрати на розробку – до 50 тис. грн.

- мінімальна дохідність – не менше 60 тис. грн.

- термін окупності – не більше 1 року

7. Стадії розробки:

а) Дослідження предметної галузі	<u>14.09.24 – 02.10.24</u>
б) Обґрунтування вибору технологічного стеку	<u>03.10.24 – 09.10.24</u>
в) Розробка програмного забезпечення	<u>10.10.24 – 10.11.24</u>
г) Тестування розробленого програмного забезпечення	<u>11.11.24 – 17.11.24</u>
д) Підготовка економічної частини	<u>18.11.24 – 24.11.24</u>
е) Оформлення пояснювальної записки	<u>25.11.24 – 04.12.24</u>

8. Порядок контролю та приймання

Рубіжний контроль провести до «15» листопада 2024 р.

Попередній захист МКР провести до «05» грудня 2024 р.

Захист МКР провести до «20» грудня 2024 р.

Розробив студент групи ІСТ-23м _____ Анна ОЛІЙНИЧЕНКО

Додаток Б (обов'язковий)**ІЛЮСТРАТИВНА ЧАСТИНА****ВЕБ-САЙТ ЛІКАРНІ З ВИКОРИСТАННЯМ ТЕЛЕМЕДИЦИНИ**

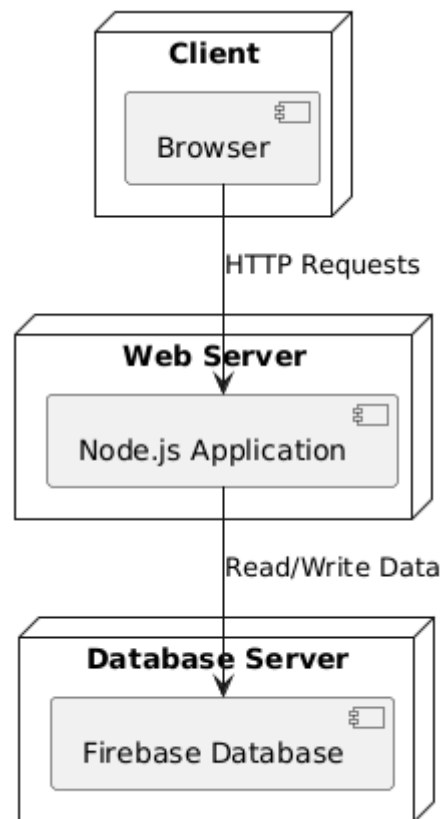


Рисунок Б.1 – Діаграма розгортання



Рисунок Б.2 – UML-діаграма прецедентів

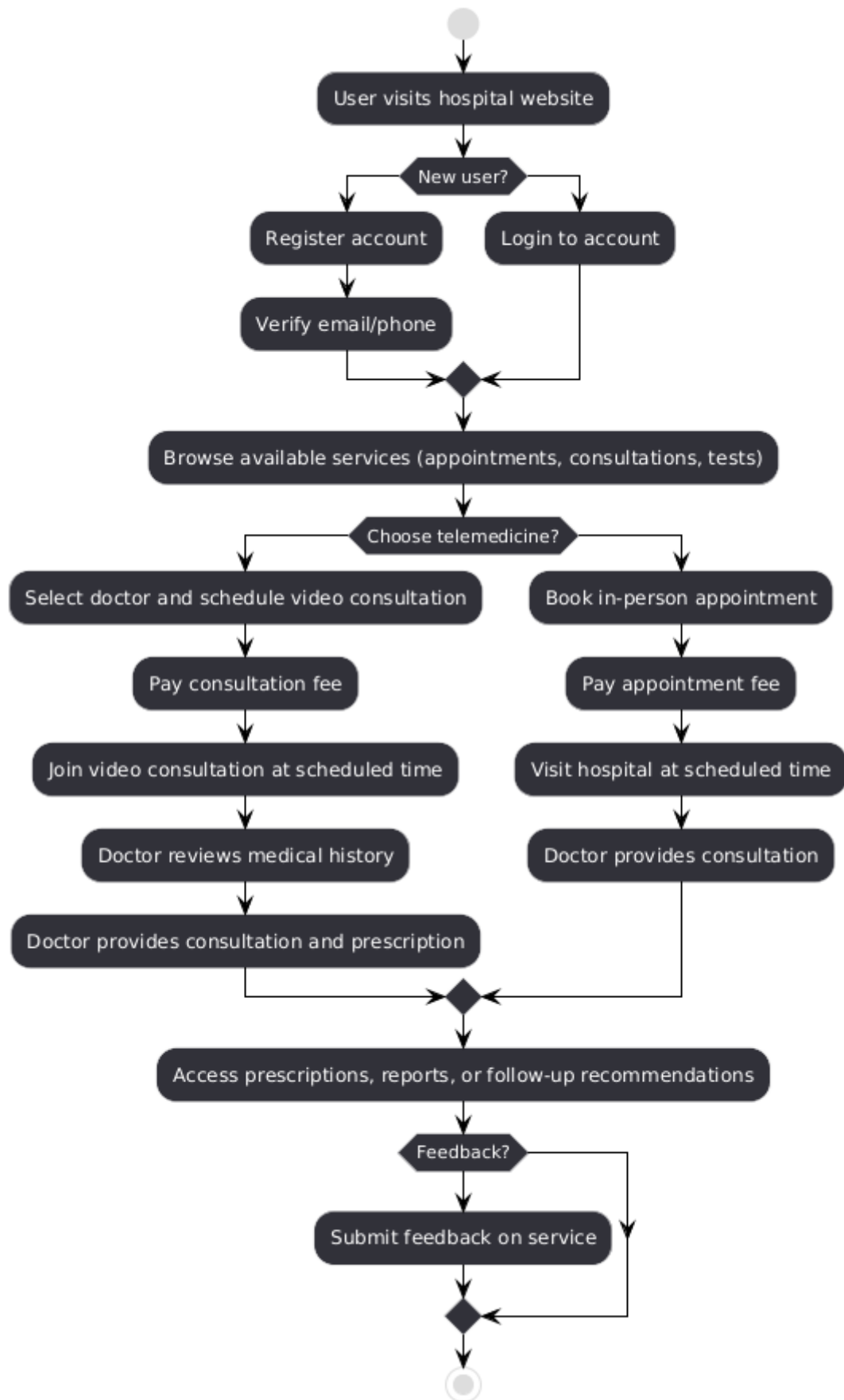


Рисунок Б.3 – UML-діаграма діяльності

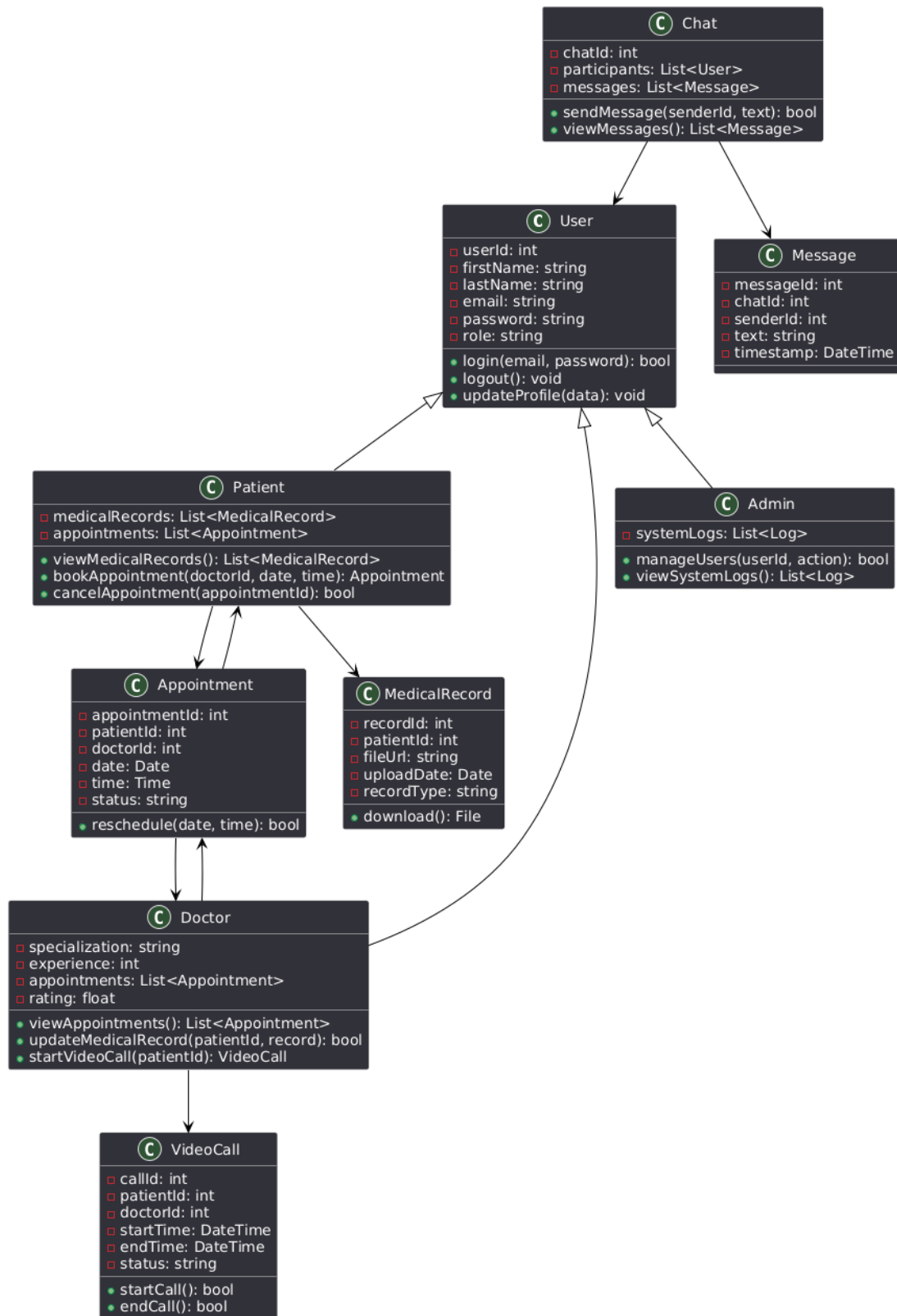


Рисунок Б.4 – UML-діаграма класів

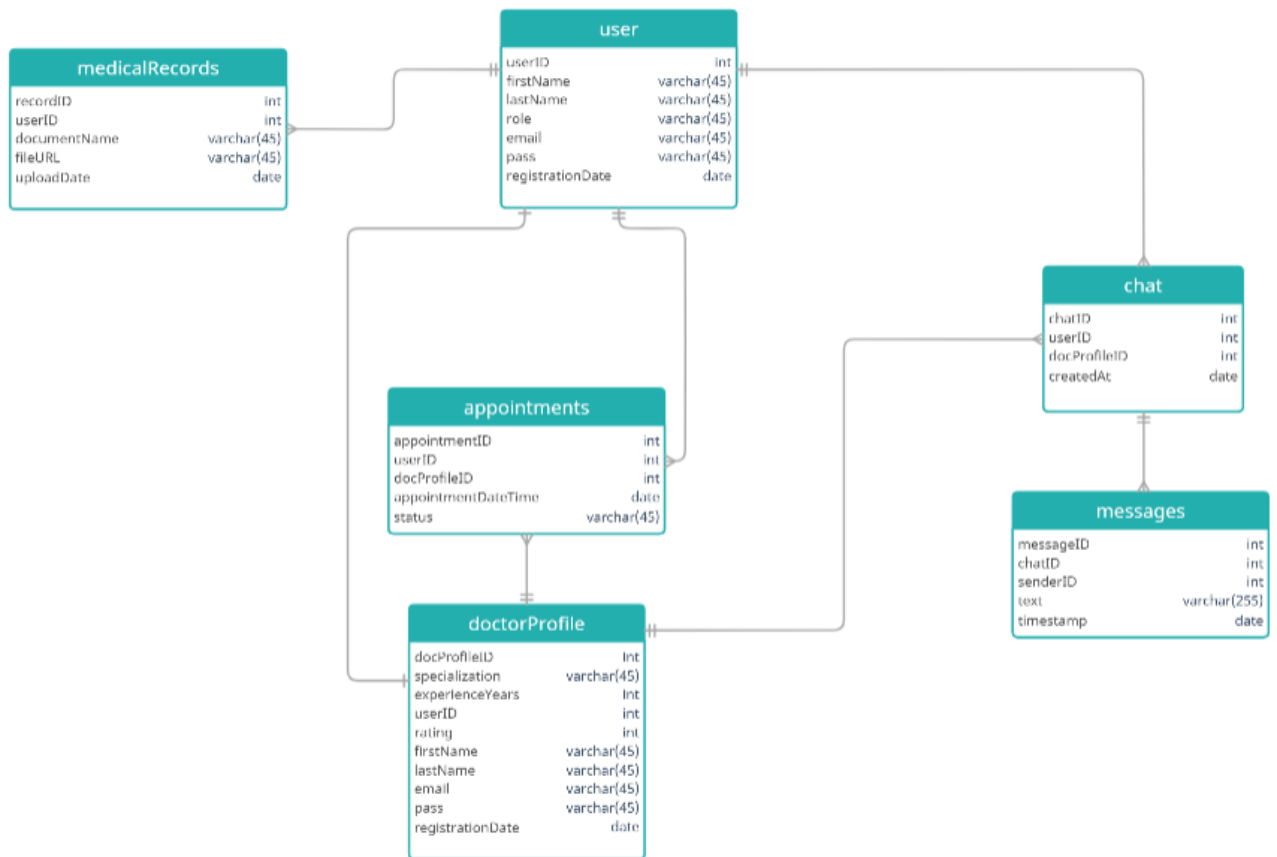


Рисунок Б.5 – Схема бази даних

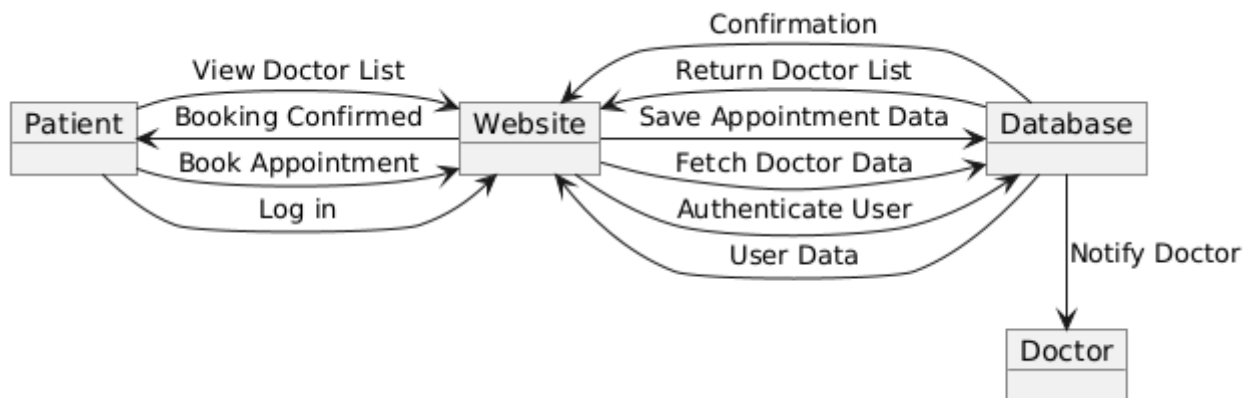


Рисунок Б.6 – Діаграма взаємодії

Додаток В (обов'язковий)

ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

MedicalRecords.vue

```

<template>
  <div class="medical-records-container">
    <h2>Медичні Записи</h2>
    <div class="upload-section" style="display: flex; flex-direction: column; align-items: center;">
      <input type="file" @change="handleFileUpload" />
      <button @click="uploadFile" :disabled="!selectedFile">Зберегти</button>
    </div>

    <div class="records-list">
      <h3>Список Записів</h3>
      <div v-if="files.length === 0" class="no-records">
        Немає збережених файлів
      </div>

      <div v-for="file in files" :key="file.id" class="file-card">
        <i class="fas fa-file-medical"></i>
        <div class="file-details">
          <p><strong>Назва:</strong> {{ file.name }}</p>
          <button @click="downloadFile(file)">Переглянути/Завантажити</button>
        </div>
      </div>
    </div>
  </div>
</template>

<script>
export default {
  data() {
    return {
      selectedFile: null,
      files: [],
      db: null,
    };
  }
};

```

```

},
created() {
  this.initIndexedDB();
},
methods: {
  initIndexedDB() {
    const request = indexedDB.open("MedicalRecordsDB", 1);

    request.onupgradeneeded = (event) => {
      this.db = event.target.result;
      if (!this.db.objectStoreNames.contains("files")) {
        this.db.createObjectStore("files", { keyPath: "id", autoIncrement: true });
      }
    };

    request.onsuccess = (event) => {
      this.db = event.target.result;
      this.loadFiles();
    };

    request.onerror = (event) => {
      console.error("Помилка відкриття бази даних:", event.target.errorCode);
    };
  },
  handleFileUpload(event) {
    this.selectedFile = event.target.files[0];
  },
  uploadFile() {
    if (!this.selectedFile) return;

    const reader = new FileReader();
    reader.readAsDataURL(this.selectedFile);

    reader.onload = () => {
      const fileData = {
        name: this.selectedFile.name,
        content: reader.result,
      };
    };
  }
}

```

```

const transaction = this.db.transaction(["files"], "readwrite");
const store = transaction.objectStore("files");
store.add(fileData);

transaction.oncomplete = () => {
  alert("Файл успішно збережено!");
  this.loadFiles();
};

transaction.onerror = (event) => {
  console.error("Помилка збереження файлу:", event.target.error);
};
},
loadFiles() {
  const transaction = this.db.transaction(["files"], "readonly");
  const store = transaction.objectStore("files");
  const request = store.getAll();

  request.onsuccess = (event) => {
    this.files = event.target.result;
  };

  request.onerror = (event) => {
    console.error("Помилка завантаження файлів:", event.target.error);
  };
},
downloadFile(file) {
  const link = document.createElement("a");
  link.href = file.content;
  link.download = file.name;
  link.click();
},
};
</script>

```

```
<style scoped>
.medical-records-container {
  max-width: 800px;
  margin: auto;
  padding: 2rem;
  background-color: #f5f5f5;
  border-radius: 10px;
  box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1);
}
```

```
.upload-section {
  display: flex;
  align-items: center;
  gap: 1rem;
  margin-bottom: 2rem;
}
```

```
.upload-section input[type="file"] {
  padding: 0.5rem;
  border: 1px solid #ccc;
  border-radius: 5px;
}
```

```
.upload-section button {
  padding: 0.5rem 1rem;
  border: none;
  border-radius: 5px;
  background-color: #6a5acd;
  color: white;
  cursor: pointer;
  transition: background 0.3s;
}
```

```
.upload-section button:hover {
  background-color: #7b68ee;
}
```

```
.upload-section button:disabled {
```

```
background-color: #ccc;
cursor: not-allowed;
}
```

```
.records-list {
  margin-top: 2rem;
}
```

```
.no-records {
text-align: center;
  color: #888;
}
```

```
.file-card {
  display: flex;
  align-items: center;
  background-color: #fff;
  padding: 1rem;
  margin-bottom: 1rem;
  border-radius: 10px;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}
```

```
.file-card i {
  font-size: 2rem;
  color: #6a5acd;
  margin-right: 1rem;
}
```

```
.file-details {
  flex: 1;
}
```

```
.file-details p {
  margin: 0.5rem 0;
}
```

```
.file-details button {
```

```
padding: 0.3rem 0.6rem;
border: none;
border-radius: 5px;
background-color: #6a5acd;
color: white;
cursor: pointer;
transition: background 0.3s;
}
```

```
.file-details button:hover {
  background-color: #7b68ee;
}
</style>
```

VideoCall.vue

```
<template>
  <div class="video-call-container">
    <h2>Відеозв'язок</h2>
    <div class="video-wrapper">
      <video ref="remoteVideo" autoplay playsinline></video>
      <video ref="localVideo" autoplay playsinline muted></video>
    </div>
    <div class="controls">
      <button @click="createPeer">Створити Peer</button>
      <input v-model="peerIdToConnect" placeholder="Введіть Peer ID" />
      <button @click="connectToPeer" :disabled="!peerIdToConnect">Підключитися до Peer</button>
      <button @click="endCall" :disabled="!isCalling">Завершити дзвінок</button>
    </div>
    <div class="control-icons">
      <button @click="toggleCamera" :disabled="!isCalling">
        <i :class="[cameraEnabled ? 'fas fa-video' : 'fas fa-video-slash']"></i>
      </button>
      <button @click="toggleMicrophone" :disabled="!isCalling">
        <i :class="[microphoneEnabled ? 'fas fa-microphone' : 'fas fa-microphone-slash']"></i>
      </button>
    </div>
  </div>
</template>
```

```

<script>
import Peer from "peerjs";

export default {
  data() {
    return {
      peer: null,
      peerId: "",
      peerIdToConnect: "",
      localStream: null,
      remoteStream: null,
      connection: null,
      call: null,
      isCalling: false,
      cameraEnabled: true,
      microphoneEnabled: true,
    };
  },
  methods: {
    async createPeer() {
      this.peer = new Peer();

      this.peer.on("open", (id) => {
        this.peerId = id;
        console.log("Ваш Peer ID:", id);
      });

      this.peer.on("call", async (incomingCall) => {
        console.log("Вхідний дзвінок");

        this.localStream = await navigator.mediaDevices.getUserMedia({
          video: true,
          audio: true,
        });

        this.$refs.localVideo.srcObject = this.localStream;
      });
    }
  }
};

```

```

incomingCall.answer(this.localStream);

incomingCall.on("stream", (remoteStream) => {
  this.remoteStream = remoteStream;
  this.$refs.remoteVideo.srcObject = remoteStream;
});

this.call = incomingCall;
this.isCalling = true;
});
},
async connectToPeer() {
  if (!this.peerIdToConnect) {
    alert("Будь ласка, введіть Peer ID для підключення");
    return;
  }

  this.localStream = await navigator.mediaDevices.getUserMedia({
    video: true,
    audio: true,
  });

  this.$refs.localVideo.srcObject = this.localStream;

  const outgoingCall = this.peer.call(this.peerIdToConnect, this.localStream);

  outgoingCall.on("stream", (remoteStream) => {
    this.remoteStream = remoteStream;
    this.$refs.remoteVideo.srcObject = remoteStream;
  });

  outgoingCall.on("close", () => {
    console.log("Дзвінок завершено");
    this.isCalling = false;
  });

  this.call = outgoingCall;
  this.isCalling = true;

```

```

    },
    endCall() {
      if (this.call) {
        this.call.close();
      }
      if (this.localStream) {
        this.localStream.getTracks().forEach((track) => track.stop());
      }
      this.remoteStream = null;
      this.isCalling = false;
    },
    toggleCamera() {
      this.cameraEnabled = !this.cameraEnabled;
      this.localStream.getVideoTracks().forEach((track) => (track.enabled = this.cameraEnabled));
    },
    toggleMicrophone() {
      this.microphoneEnabled = !this.microphoneEnabled;
      this.localStream.getAudioTracks().forEach((track) => (track.enabled = this.microphoneEnabled));
    },
  },
};
</script>

```

```

<style scoped>
.video-call-container {
  max-width: 800px;
  margin: auto;
  padding: 2rem;
  background-color: #f5f5f5;
  border-radius: 10px;
  box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1);
}

.video-wrapper {
  display: flex;
  justify-content: space-between;
  margin-bottom: 1.5rem;
}

```

```
video {  
  width: 48%; height: auto;  
  border-radius: 10px;  
  background-color: black;  
}
```

```
.controls {  
  display: flex;  
  justify-content: center;  
  flex-wrap: wrap;  
  gap: 1rem;  
  margin-bottom: 1rem;  
}
```

```
.control-icons {  
  display: flex;  
  justify-content: center;  
  gap: 1rem;  
}
```

```
button {  
  padding: 0.5rem 1rem;  
  border: none;  
  border-radius: 5px;  
  background-color: #6a5acd;  
  color: white;  
  cursor: pointer;  
  transition: background 0.3s;  
}
```

```
button:hover {  
  background-color: #7b68ee;  
}
```

```
button:disabled {  
  background-color: #ccc;
```

Додаток Г (обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: «Веб-сайт лікарні з використанням телемедицини»

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра АІТ

Показники звіту подібності Turnitin

Оригінальність 98,0 %

Схожість 2,0 %

Аналіз звіту подібності (відмітити потрібне)

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
 - Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
 - Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Роман МАСЛІЙ
(підпис)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Анна ОЛІЙНИЧЕНКО
(підпис)

Керівник роботи _____ Ярослав КУЛИК
(підпис)