

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ЕЛЕКТРОННОГО МЕДИЧНОГО
ЗАПИСУ
(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-21-2

Малахова А. Р.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Шафроненко А. Ю.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____
(підпис)

Кобилін О. А.
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Малаховій Анастасії Романівні
(прізвище, ім'я, по батькові)1. Тема роботи Розробка застосунку для електронного медичного запису

затверджена наказом університету від 19 травня 2025 року № 381Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 7 червня 2025 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, офіційна документація MongoDB, Express.js, Node.js, бібліотеки Tailwind CSS, Material-UI, JWT, Redis, WebRTC, HL7 FHIR, Docker, Kubernetes, приклади JSON-структур медичних записів, API-документація для систем автентифікації, ресурси з розробки RESTful сервісів, матеріали конференцій з цифрової медицини.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз існуючих платформ електронного медичного запису та їх функціоналу.2. Архітектура та структура застосунку для організації онлайн-запису до лікаря.3. Розробка і тестування вебзастосунку з використанням стеку MERN для медичного запису.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми обробки зображень, постановка задачі, тестові зображення.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	07.04.2025	
2	Аналіз завдання, підбір літератури	08.04.25-10.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-14.04.25	
4	Аналіз технічних засобів	15.04.25-20.04.25	
5	Розробка методу	21.04.25-27.04.25	
6	Програмна реалізація	28.04.25-11.05.25	
7	Оформлення пояснювальної записки	12.05.25-20.05.25	
8	Перевірка на нормоконтроль	21.05.25-01.06.25	
9	Перевірка на плагіат	21.05.25-01.06.25	
10	Рецензування	21.05.25-01.06.25	
11	Підготовка презентації та доповіді	21.05.25-18.06.25	
12	Занесення роботи в електронний архів	02.06.25-18.06.25	
13	Попередній захист кваліфікаційної роботи	02.06.25-18.06.25	

Дата видачі завдання 7 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Шафроненко А. Ю.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 55 с., 16 рис., 26 джерел.

ЕЛЕКТРОННИЙ МЕДИЧНИЙ ЗАПИС, MERN-СТЕК, REACT, NODE.JS, MONGODB, АВТЕНТИФІКАЦІЯ, СИСТЕМА БРОНЮВАННЯ ПРИЙОМІВ, ВЕБРОЗРОБКА.

Об'єктом роботи є система електронного медичного запису з можливістю бронювання прийомів до лікаря.

Метою роботи є розробка вебзастосунку, що дозволяє пацієнтам записуватись на прийом до лікаря, керувати своїми візитами, а адміністраторам – управляти розкладом, переглядати список пацієнтів і редагувати профілі лікарів.

Було створено вебплатформу для електронного медичного запису з двома рівнями автентифікації: для пацієнтів та адміністраторів. Реалізовано систему бронювання, перегляду й керування прийомами. Клієнтська частина розроблена з використанням React.js, серверна частина – на Node.js і Express.js, база даних – MongoDB.

У результаті роботи створено повнофункціональний вебзастосунок, який дає змогу ефективно організувати процес запису до лікаря, автоматизувати адміністративну роботу та забезпечити зручність як для пацієнтів, так і для персоналу медичного закладу.

ELECTRONIC MEDICAL RECORD, MERN STACK, REACT, NODE.JS, MONGODB, AUTHENTICATION, APPOINTMENT BOOKING SYSTEM, WEB DEVELOPMENT.

The object of this work is an electronic medical record system with appointment booking functionality.

The aim of the work is to develop a web application that allows patients to book medical appointments and manage their visits, while administrators can manage schedules, view patient lists, and edit doctor profiles.

A web platform for electronic appointment booking has been developed with two levels of authentication: for patients and administrators. The system includes appointment scheduling, booking management, and user role-based access. The client side is built with React.js, the server side is developed using Node.js and Express.js, and the database is managed with MongoDB.

As a result, a fully functional web application has been created, enabling efficient organization of medical appointments, automation of administrative tasks, and convenience for both patients and healthcare staff.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Аналіз існуючих застосунків для запису на медичні прийоми	9
1.1 Актуальні підходи до організації запису на медичні прийоми.....	9
1.2 Поточний стан систем електронного медичного запису та аналіз існуючих рішень.....	11
1.3 Досвід використання сервісів електронного медичного запису....	14
1.4 Сучасні технології в розвитку цифрових платформ для ведення медичної документації.....	16
1.5 Аналіз функціоналу та ключових характеристик сучасних цифрових платформ для медичного запису	18
1.6 Постановка задачі	21
2 Проектування онлайн-сервісу для запису до лікаря.....	23
2.1 Специфікація вимог до сервісу	23
2.2 Проектування архітектури сервісу для електронного медичного запису	25
2.3 Моделювання структури програми для електронного медичного запису	28
2.4 Клієнтська частина застосунку для електронного медичного запису	32
2.5 Серверна частина застосунку для електронного медичного запису	34
3 Розробка застосунку для електронного медичного запису	37
3.1 Вибір інструментальних засобів для реалізації поставленої задачі.....	37
3.2 Архітектура системи та структура застосунку	39
3.3 Основні можливості застосунку.....	40

3.3.1	Реєстрація та вхід.....	41
3.3.2	Запис до лікаря	41
3.3.3	Панель адміністратора.....	42
3.3.4	Особистий кабінет пацієнта.....	43
3.3.5	Робота з файлами (за потреби)	43
3.4	Тестування роботи застосунку	44
	Висновки	51
	Перелік джерел посилання	53

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

EMR – Electronic Medical Record (електронний медичний запис)

MERN – MongoDB, Express.js, React.js, Node.js (технологічний стек для веброзробки)

JS – JavaScript (мова програмування для веброзробки)

React.js – JavaScript-бібліотека для створення інтерфейсів користувача

Node.js – середовище виконання JavaScript на сервері

Express.js – фреймворк для створення вебсервісів на Node.js

MongoDB – документоорієнтована NoSQL база даних

NoSQL – Not Only SQL (не лише структуровані дані)

JWT – JSON Web Token (токен аутентифікації користувачів)

API – Application Programming Interface (інтерфейс програмного застосунку)

UI – User Interface (користувацький інтерфейс)

UX – User Experience (досвід користувача)

CRUD – Create, Read, Update, Delete (основні операції з базою даних)

SPA – Single Page Application (односторінковий вебзастосунок)

REST – Representational State Transfer (архітектурний стиль вебсервісів)

Auth – Authentication (процес підтвердження особи користувача)

ВСТУП

Інформатизація медицини набирає обертів, і сьогодні важко уявити сучасний медичний заклад без цифрових сервісів. Однією з найбільш затребуваних функцій є можливість онлайн-запису до лікаря та ведення електронної історії хвороб. Це дозволяє не лише полегшити процес взаємодії між пацієнтом і медичним персоналом, а й значно покращити якість обслуговування, підвищити оперативність та знизити навантаження на адміністративний персонал.

Оскільки темп життя стає все швидшим, пацієнти прагнуть зекономити час і мати змогу керувати своїми медичними візитами дистанційно. Це створює потребу у зручних та надійних вебзастосунках, які б дозволяли швидко записатися на прийом, отримати інформацію про лікаря чи переглянути майбутні візити. Адміністратори ж зі свого боку потребують ефективного інструменту для управління розкладом і пацієнтськими запитами.

У цьому контексті використання сучасного MERN-стеку (MongoDB, Express.js, React.js, Node.js) для створення вебзастосунку є цілком виправданим. Завдяки цим технологіям вдається реалізувати як зручний інтерфейс користувача, так і стабільну серверну логіку та надійне зберігання даних. Додаткові механізми автентифікації забезпечують розмежування прав доступу між пацієнтами та адміністраторами, що гарантує безпеку персональної інформації.

Окрім практичного застосування, розроблена система має потенціал для подальшого розширення: від автоматичних нагадувань про прийоми до збору статистики для управління ресурсами клініки. Упровадження таких рішень сприяє не лише підвищенню ефективності лікувального процесу, а й загальному вдосконаленню роботи медичних установ.

1 АНАЛІЗ ІСНУЮЧИХ ЗАСТОСУНКІВ ДЛЯ ЗАПИСУ НА МЕДИЧНІ ПРИЙОМИ

1.1 Актуальні підходи до організації запису на медичні прийоми

Цифровізація сфери охорони здоров'я останніми роками набуває дедалі більшого значення, оскільки ефективна організація роботи медичних закладів напряду впливає на якість обслуговування пацієнтів. Однією з важливих складових цифрової трансформації є впровадження систем електронного запису до лікаря. Традиційні методи – такі як особистий візит до реєстратури або телефонне бронювання – поступово відходять у минуле. Їх замінюють сучасні онлайн-інструменти, які забезпечують швидкий, зручний і доступний для користувачів спосіб керування своїм медичним обслуговуванням.

Сьогодні пацієнти очікують високого рівня сервісу від медичних установ, а тому прагнуть отримати можливість самостійно обирати лікаря, переглядати доступні години прийому та записуватись у зручний час – без зайвого очікування чи необхідності дзвонити на гарячу лінію. Для цього широко використовуються різноманітні платформи та мобільні додатки, серед яких варто виділити такі, як Helsi.me, Doc.ua, Likarni.com, Medics тощо. Ці сервіси дозволяють пацієнтам швидко знайти потрібного спеціаліста, переглянути розклад прийомів, залишити відгук після візиту, а також отримувати автоматичні нагадування про запис.

Крім сторонніх сервісів, чимало приватних клінік і мережевих медичних центрів розробляють власні програмні рішення, які включають не тільки електронний запис, але й інтеграцію з електронною медичною картою (EMR – Electronic Medical Record). Це дає змогу лікарям миттєво отримувати доступ до медичної історії пацієнта, результатів аналізів, попередніх діагнозів і призначень. Завдяки цьому суттєво підвищується ефективність надання медичних послуг і скорочується час на обробку даних [1-4].

Особливу популярність набирають вебзастосунки на основі MERN-стека (MongoDB, Express.js, React.js, Node.js), які дозволяють створити повнофункціональні, масштабовані та гнучкі системи запису. У таких системах реалізуються два рівні доступу: пацієнтський – для користувачів, які можуть реєструватися, переглядати список лікарів, бронювати візити, редагувати або скасовувати записи, та адміністративний – для адміністраторів клініки, які можуть керувати даними лікарів, перевіряти розклад, переглядати історію бронювань і керувати базою пацієнтів. Це не лише зручно, а й дає змогу оптимізувати адміністративні процеси в закладі.

Додатково, у сучасних системах електронного запису з'являється все більше функцій, пов'язаних із штучним інтелектом. Наприклад, алгоритми можуть автоматично аналізувати дані про попит на окремі медичні послуги, визначати найнавантаженіші години та генерувати рекомендації щодо оптимізації графіка лікарів. Штучний інтелект також може допомагати пацієнтам підбирати відповідного лікаря на основі симптомів, попередніх візитів або навіть аналізу зібраної медичної інформації. Такі можливості не лише покращують взаємодію користувачів із системою, а й сприяють більш ефективному розподілу ресурсів клініки.

Ще однією тенденцією є інтеграція сервісів електронного запису з іншими системами – наприклад, календарями, платіжними системами, месенджерами та e-mail-сервісами. Це дозволяє налаштувати повний цикл взаємодії з пацієнтом: від онлайн-бронювання до надсилання повідомлення з результатами обстежень. Також часто впроваджуються можливості офлайн-доступу або PWA-додатки (прогресивні вебзастосунки), які дають змогу використовувати функціонал навіть при поганому інтернет-з'єднанні.

Сучасні користувачі все частіше звертаються до соціальних мереж та онлайн-форумів, щоб знайти відгуки про лікарів чи клініки перед візитом. Це свідчить про зростання довіри до цифрових рецензій та рекомендацій, які стають частиною інформаційного середовища медичного запису. Медичні установи також починають активно вести соціальні сторінки, впроваджувати

онлайн-консультації та цифрові форми зворотного зв'язку для покращення сервісу.

Як результат, організація електронного медичного запису в умовах сучасних технологій – це не просто зручна альтернатива традиційним методам, а стратегічно важливий елемент ефективної та клієнтоорієнтованої медицини. Індивідуальні вебзастосунки, створені з використанням сучасних стеків розробки, відкривають нові можливості як для пацієнтів, так і для адміністрації клінік, дозволяючи мінімізувати навантаження на персонал, підвищити якість сервісу та забезпечити кращу доступність медичних послуг.

1.2 Поточний стан систем електронного медичного запису та аналіз існуючих рішень

У сучасному світі цифрових технологій, коли автоматизація охоплює все більше сфер життя, медична галузь також переживає активну трансформацію. Одним із ключових напрямів цієї трансформації є впровадження електронних медичних записів (ЕМЗ) – систем, які забезпечують зручний, структурований та захищений доступ до інформації про пацієнтів, їхні медичні візити, діагнози, історію лікування, призначення лікарів та інші супровідні дані.

На сьогоднішній день існує велика кількість комерційних та відкритих рішень, які надають функціонал для управління медичними записами. Програмні платформи, як-от ZocDoc, DrChrono, SimplePractice, MediCloud, Healthie та Kareo, пропонують широкий набір можливостей: від запису на прийом до телемедичних консультацій, ведення електронної картки пацієнта, виставлення рахунків, електронних рецептів, інтеграції з лабораторіями та страхуванням. Проте, не дивлячись на багатий функціонал, багато з цих сервісів мають істотні недоліки, які обмежують їхнє ефективне використання в умовах реального медичного закладу.

По-перше, значна частина рішень орієнтована на великі клініки або госпітальні комплекси, що має наслідком високі витрати на ліцензування, обслуговування та налаштування. Це створює бар'єр для впровадження ЕМЗ у приватних медичних кабінетах, невеликих клініках або на індивідуальному рівні практикуючих лікарів. Крім того, деякі сервіси не надають достатньої гнучкості в кастомізації інтерфейсу чи функціоналу під конкретні потреби користувачів.

По-друге, багато рішень не мають достатнього рівня зручності з точки зору користувачького досвіду. Незрозумілий або перевантажений інтерфейс, застарілий дизайн, відсутність мобільної адаптації або відсутність повноцінного SPA-рішення (Single Page Application) створюють труднощі в повсякденному використанні, як для пацієнтів, так і для лікарів. Замість того, щоб спростити процес комунікації, такі платформи можуть ускладнювати його через зайві кроки або недостатньо логічну навігацію.

Ще одним поширеним викликом є відсутність ефективної системи ролей та рівнів доступу. У багатьох рішеннях пацієнт і лікар мають надто схожий функціонал, що створює ризики для безпеки даних. У той час як у повноцінному сучасному застосунку має бути реалізовано як мінімум два рівні доступу: для пацієнта – з можливістю реєстрації, перегляду розкладу лікарів, бронювання візитів, перегляду особистої історії; та для адміністратора – з можливістю керування візитами, додавання нових лікарів, перегляду статистики та редагування профілів.

На тлі вищезгаданих проблем і недоліків, все більше розробників звертаються до створення індивідуальних рішень, побудованих на сучасних технологіях, зокрема стеку MERN (MongoDB, Express.js, React.js, Node.js). Такий підхід дозволяє не лише реалізувати необхідну функціональність, а й забезпечити гнучкість, масштабованість та високий рівень безпеки. MERN-стек забезпечує створення інтерактивних, швидких і надійних вебдодатків, де фронтенд на React.js дозволяє побудувати приємний та адаптивний інтерфейс,

a Node.js та Express.js на бекенді – обробку запитів, авторизацію, захист API та зв'язок із базою даних MongoDB [5-9].

Крім того, при створенні сучасного ЕМЗ-застосунку варто враховувати такі компоненти, як:

- адаптивність інтерфейсу – користувач повинен мати змогу користуватись системою з мобільного, планшета чи комп'ютера;
- інтелектуальні сповіщення та нагадування – для пацієнтів про майбутні візити, для лікарів – про нові бронювання;
- історія змін – усі дії мають бути відслідковані для медичної відповідальності;
- шифрування даних – з метою дотримання стандартів безпеки, наприклад, HIPAA чи GDPR;
- пошукова система – для швидкого знаходження пацієнта, дати, лікаря, тощо;
- аналітика – можливість бачити кількість записів, популярні години візитів, навантаження на лікарів.

Розробка власного вебзастосунку для електронного медичного запису, адаптованого під потреби конкретної медичної установи, дозволяє уникнути обмежень сторонніх рішень і забезпечити комфортну роботу з пацієнтами. Це не лише покращує якість медичного обслуговування, а й оптимізує внутрішні процеси, економить час персоналу та підвищує загальну ефективність медичного сервісу.

Таким чином, потреба у нових, адаптивних та персоналізованих електронних медичних системах зростає. І саме створення таких рішень на основі сучасних технологій, з урахуванням реальних потреб лікарів та пацієнтів, є актуальним напрямом як у розробці, так і в дослідженнях.

1.3 Досвід використання сервісів електронного медичного запису

У сучасних умовах цифровізації практично всіх сфер життя, медична галузь також зазнає значних трансформацій. Особливої актуальності набувають сервіси, що дозволяють пацієнтам взаємодіяти з лікарями, клініками та медичними установами онлайн. Одним із ключових інструментів, що забезпечує такий рівень комунікації та обслуговування, є електронний медичний запис (ЕМЗ).

Платформи для електронного запису до лікаря стали звичними для мільйонів користувачів у всьому світі. Замість тривалого очікування в телефонних чергах або фізичного візиту до реєстратури, люди тепер мають змогу здійснювати запис на прийом до спеціаліста всього за кілька кліків. Такий формат є особливо зручним для зайнятих людей, літніх пацієнтів або мешканців віддалених населених пунктів, яким складно дістатися до лікарні чи поліклініки.

Досвід використання подібних сервісів демонструє, що цифрові рішення суттєво підвищують ефективність організації медичного обслуговування. Наприклад, користувачі можуть не лише бронювати прийом до лікаря, а й отримувати сповіщення про майбутні візити, змінювати час запису, скасовувати зустрічі, переглядати історію попередніх звернень, рецепти, аналізи, або залишати відгуки про надану допомогу. Усе це відбувається через зручний вебінтерфейс або мобільний застосунок.

Крім того, розвиток цифрових медичних рішень створює сприятливі умови для персоналізації обслуговування. Сервіси можуть зберігати вподобання пацієнтів, наприклад, щодо зручного часу відвідування чи вибраного лікаря, на основі чого формуються індивідуальні пропозиції. Використовуючи базові алгоритми машинного навчання, платформи можуть навіть прогнозувати потенційні потреби пацієнта, рекомендуючи, наприклад, пройти певні обстеження згідно з його віком, діагнозами або медичною історією.

З технічної точки зору, більшість сучасних систем ЕМЗ функціонують на основі хмарних технологій, що забезпечує швидкий доступ до даних з будь-якого пристрою – комп'ютера, планшета чи смартфона. Для користувача це означає, що його медична інформація постійно під рукою, а отже, він може оперативно відреагувати на будь-які медичні потреби – записатися на прийом, проконсультуватися з лікарем онлайн, переглянути результати аналізів тощо.

Однак реальний користувацький досвід показує, що не всі сервіси працюють однаково ефективно. Багато пацієнтів зазначають наявність таких проблем, як повільне завантаження сторінок, застарілий дизайн, неінтуїтивний інтерфейс, складна система реєстрації або відсутність можливості швидко отримати підтримку. Такі бар'єри можуть серйозно знижувати рівень задоволеності від користування, особливо серед людей похилого віку або тих, хто не має високої цифрової грамотності.

Водночас, досвід лікарів також вартий уваги. Для них якісний сервіс ЕМЗ – це не просто система запису, а повноцінна цифрова платформа для ведення пацієнтів, яка включає електронні картки, аналітичні інструменти, шаблони для ведення медичних записів, автоматичне формування графіків прийому, а також механізми сповіщень і нагадувань. Особливо важливо, щоб система дозволяла лікарям швидко знаходити потрібну інформацію, зручно вести документацію та взаємодіяти з пацієнтами у реальному часі.

На сьогодні у світі вже є чимало прикладів успішного впровадження систем ЕМЗ. Наприклад, у США діють такі платформи, як MyChart, Epic, Kareo, які інтегрують різні функції – від запису до лікаря до онлайн-консультацій і оплати послуг. В Україні також з'являються вітчизняні рішення, як-от helsi.me, яке дозволяє мешканцям багатьох міст записуватись на прийом до лікаря через інтернет, переглядати електронні рецепти, або отримувати консультації через відеозв'язок.

Зростає також кількість клінік, які переходять на повністю електронний формат роботи, відмовляючись від паперової документації. Це дозволяє уникнути дублювання даних, втрати медичної інформації, помилок у

призначеннях або людського фактора при обробці інформації. Такий підхід позитивно позначається як на пацієнтах, так і на медперсоналі.

В умовах постійного зростання попиту на цифрові медичні послуги, розробка зручного, швидкого та функціонального застосунку для ЕМЗ, побудованого на сучасних технологіях (наприклад, MERN-стеку – MongoDB, Express.js, React.js, Node.js), є надзвичайно актуальним завданням. Подібний проєкт дозволить забезпечити двосторонню взаємодію між лікарем і пацієнтом, ефективно управляти медичними записами та вивести рівень медичного обслуговування на нову якість [10-15].

1.4 Сучасні технології в розвитку цифрових платформ для ведення медичної документації

У наш час цифрові технології активно трансформують медичну сферу, зокрема процес ведення медичної документації. Системи електронного медичного запису (ЕМЗ) стають важливою частиною інфраструктури охорони здоров'я, адже дозволяють автоматизувати більшість рутинних завдань, покращити обслуговування пацієнтів та підвищити ефективність роботи медичного персоналу.

Однією з ключових інновацій у цій галузі є використання штучного інтелекту. Завдяки йому медичні платформи навчаються аналізувати дані пацієнтів і надавати точні рекомендації щодо лікування, профілактики або обстежень. Такі системи здатні не лише зберігати інформацію, а й інтерпретувати її, створюючи індивідуальні плани дій як для лікарів, так і для пацієнтів.

Ще одним важливим технологічним напрямом є аналіз великих даних (Big Data). За допомогою спеціальних алгоритмів можливо виявляти закономірності у стані здоров'я різних груп пацієнтів, прогнозувати ризики захворювань, а також виявляти потенційні епідеміологічні загрози. Це

відкриває широкі можливості для запобігання хворобам та стратегічного планування в межах медичних установ.

Широке впровадження мобільних додатків для пацієнтів дозволяє зробити медичні послуги більш доступними. Пацієнти можуть у будь-який час переглядати свою історію хвороб, результати обстежень, отримані призначення, а також записуватись на прийом або отримувати консультації онлайн. Це особливо актуально в умовах віддаленої медицини та під час пандемій.

Окремої уваги заслуговує хмарне зберігання даних, яке забезпечує доступ до інформації з будь-якого пристрою. Завдяки цьому лікарі можуть оперативно отримувати необхідні дані навіть у разі зміни місця роботи або під час консультації з іншими фахівцями. Крім того, зменшується ризик втрати даних, адже вся інформація дублюється на надійних серверах.

Сьогодні все більше уваги приділяється безпеці та захисту особистої інформації, тому технології на кшталт блокчейну активно інтегруються в медичні сервіси. Вони забезпечують прозорість, надійність та контроль доступу до чутливих даних. Кожна дія в системі фіксується, і пацієнт завжди знає, хто переглядав його медичні записи.

Сучасні сервіси також впроваджують відеозв'язок і віртуальні консультації, які дозволяють зменшити потребу у фізичних візитах до лікаря. Це зручно для пацієнтів із віддалених населених пунктів, осіб із обмеженою мобільністю або в разі легких симптомів, що не потребують фізичного огляду.

Зростає популярність інтегрованих платіжних систем, які дозволяють оплачувати медичні послуги напряму через платформу. Це скорочує час на бюрократичні процедури і робить користування медичними сервісами значно зручнішим як для пацієнтів, так і для клінік.

Інноваційним елементом стають голосові інтерфейси та цифрові асистенти, що дозволяють лікарям керувати записами, ставити діагнози чи призначати лікування за допомогою голосових команд. Це економить час і полегшує роботу, особливо у випадках великого потоку пацієнтів.

Деякі платформи пропонують функції спільного доступу до даних, що дозволяє родичам або декільком лікарям спільно керувати здоров'ям пацієнта. Наприклад, батьки можуть мати повний доступ до історії хвороб дитини, або терапевт – бачити діагнози інших фахівців, що покращує координацію лікування.

Не варто забувати і про відгуки та рейтинги, які користувачі залишають у медичних додатках. Це формує прозору оцінку медичних послуг та дозволяє пацієнтам обирати лікарів на основі реального досвіду інших людей.

Отже, впровадження сучасних технологій у сферу електронного медичного запису сприяє формуванню нової моделі надання медичної допомоги – гнучкої, персоналізованої та доступної. Такі системи дозволяють зменшити навантаження на лікарів, покращити якість обслуговування та зробити медицину більш орієнтованою на потреби кожного окремого пацієнта.

1.5 Аналіз функціоналу та ключових характеристик сучасних цифрових платформ для медичного запису

У сфері охорони здоров'я останніми роками активно впроваджуються цифрові технології, які істотно змінюють спосіб взаємодії між пацієнтом і медичними установами. Одним із найяскравіших прикладів такої трансформації стали електронні сервіси для запису до лікаря. Вони вже давно перестали бути просто зручним додатком – сьогодні це необхідний інструмент, що оптимізує роботу медичних установ і покращує доступ пацієнтів до якісної медичної допомоги.

Ці сервіси забезпечують простий і зрозумілий інтерфейс для пошуку лікаря за спеціалізацією, місцем розташування, часом прийому та іншими параметрами. Пацієнт може без зайвих зусиль знайти необхідного фахівця, переглянути доступні години для прийому, записатися онлайн та отримати

підтвердження – все це займає лише кілька хвилин. У той же час, для лікарів і клінік така система дозволяє автоматизувати ведення розкладу, уникнути накладок у графіку та ефективно керувати потоком пацієнтів.

Сучасні сервіси електронного медичного запису також підтримують функцію нагадувань про візит через електронну пошту або SMS, що суттєво знижує кількість випадків, коли пацієнти не з'являються на прийом. Окрім цього, у багатьох платформах реалізовано зберігання історії звернень, що дозволяє лікарю мати доступ до попередніх візитів пацієнта та приймати обґрунтовані рішення під час наступної консультації.

Більшість сучасних додатків мають схожий набір базових функцій, однак вирізняються своїми унікальними можливостями, рівнем зручності, дизайном інтерфейсу та ступенем персоналізації досвіду користувача. Нижче наведено огляд основних функцій, які реалізовані у популярних медичних системах:

- Doctolib – один із найвідоміших сервісів у Європі, що дозволяє пацієнтам швидко знаходити потрібного спеціаліста за спеціалізацією, локацією та доступним графіком. Також доступна синхронізація з календарем користувача та автоматичні нагадування про візити. Лікарі мають змогу вести базу пацієнтів та відстежувати історію візитів;

- Zocdoc – сервіс, популярний у США, з інтеграцією електронної медичної картки та можливістю залишати відгуки про лікарів. Пацієнти можуть переглядати розклад у реальному часі, бронювати візити та отримувати підтвердження в один клік. Крім того, Zocdoc враховує медичне страхування при підборі лікаря;

- Medix – рішення для клінік, орієнтоване не лише на запис, а й на аналітику відвідуваності, ефективності лікарів, середній час очікування тощо. Також підтримує модуль електронних рецептів, ведення амбулаторної карти та інтеграцію з лабораторіями;

- BookDox – застосунок, який об'єднує функції запису на прийом з рекомендаціями щодо здорового способу життя. Присутні елементи

гейміфікації: пацієнти можуть отримувати бали за регулярні візити, виконання рекомендацій лікарів тощо;

- BetterMed – приклад платформи з розширеними функціями для лікарів: чат з пацієнтом, перегляд результатів аналізів, створення електронних нотаток після прийому, інтеграція з CRM системами клініки;

- MyChart – сервіс з доступом до повної історії хвороби, результатів обстежень, назначень і планів лікування. Також підтримується функція «віртуального прийому» через відеозв'язок;

- DocPlanner – сервіс, який дозволяє лікарям створити повноцінний онлайн-профіль, вказати години прийому, спеціалізацію, досвід, і приймати записи напряму. Система також дозволяє вести статистику, збирати відгуки, та надає модуль комунікації з пацієнтом.

Ще однією важливою можливістю є ведення електронної медичної картки, яка містить діагнози, результати обстежень, рекомендації та назначення. Така централізована система дозволяє лікарям різних спеціальностей швидко ознайомитися з повною клінічною картиною без необхідності повторного збору інформації. Пацієнт, своєю чергою, має постійний доступ до власних медичних даних, що підвищує рівень прозорості й довіри до процесу лікування.

Багато платформ також інтегрують можливість проведення онлайн-консультацій через відеозв'язок. Це особливо актуально в умовах пандемій або для пацієнтів, які знаходяться у віддалених населених пунктах і не мають можливості приїхати особисто. Завдяки цьому доступ до медичної допомоги стає не лише зручнішим, а й більш інклюзивним.

Сучасні сервіси часто мають багатомовну підтримку, що робить їх доступними для ширшої аудиторії, а також інтегруються з платіжними системами, що дозволяє пацієнтам сплачувати послуги безпосередньо в додатку. У деяких випадках навіть передбачена синхронізація з календарем користувача, що допомагає краще організувати свій час.

Усі ці функціональні можливості свідчать про те, що цифрові платформи для медичного запису вже стали не просто альтернативою традиційному запису через реєстратуру, а повноцінною екосистемою, яка забезпечує зручність, швидкість, доступність та високу якість сервісу як для пацієнтів, так і для медичних працівників.

1.6 Постановка задачі

Розробка онлайн-сервісу для планування медичних записів є важливим завданням у сучасному світі, де спостерігається зростаюча потреба в зручних і доступних інструментах для організації візитів до лікарів. В умовах швидко змінюваного ринку та високих вимог до цифрових технологій, розробка вебзастосунку, який інтегрує новітні технології, є ключовою для полегшення доступу до медичних послуг. Тому важливо створити інтуїтивно зрозумілий та ефективний сервіс, який задовольнить потреби користувачів.

Об'єктом роботи є система електронного медичного запису з можливістю бронювання прийомів до лікаря.

Метою роботи є розробка вебзастосунку, що дозволяє пацієнтам записуватись на прийом до лікаря, керувати своїми візитами, а адміністраторам – управляти розкладом, переглядати список пацієнтів і редагувати профілі лікарів.

Для досягнення цієї мети необхідно вирішити наступні завдання:

- аналіз існуючих сервісів для медичного запису – це дозволить виявити основні переваги та недоліки сучасних платформ для запису на прийом до лікаря, що допоможе визначити, які функціональні можливості потрібно інтегрувати в розроблювану систему;

- вивчення технологічних тенденцій у галузі онлайн-сервісів для медицини – потрібно вивчити останні досягнення в технологіях, таких як

автоматизоване управління розкладами, інтеграція з електронними медичними картками, штучний інтелект для персоналізації рекомендацій і використання систем для відеоконсультацій;

– проектування архітектури сервісу – необхідно спроектувати функціональну архітектуру системи, що включає модулі для запису до лікаря, створення медичних карток, а також інтеграцію з іншими медичними системами для отримання актуальної інформації;

– розробка механізму запису до лікарів – передбачає реалізацію автоматизованого механізму для запису на прийом до лікаря, враховуючи спеціалізацію лікаря, доступні години прийому та локацію пацієнта;

– розробка алгоритму для персоналізованих рекомендацій – необхідно створити алгоритм, який би враховував попередні візити пацієнта, його захворювання та вподобання, що дозволить надавати рекомендації щодо вибору лікаря, клініки та медичних процедур;

– створення зручного інтерфейсу для користувачів – важливо, щоб інтерфейс був не лише зручним, але й доступним для людей з обмеженими можливостями. Потрібно забезпечити швидкий доступ до всіх функцій сервісу, щоб користувачі могли безперешкодно організовувати свої візити до лікаря;

– розробка адміністративного інтерфейсу для лікарів і медичних установ, необхідно створити функціонал для медичних працівників, що дозволяє їм легко управляти своїм графіком, бачити записані прийоми, а також керувати історією звернень пацієнтів.

2 ПРОЄКТУВАННЯ ОНЛАЙН-СЕРВІСУ ДЛЯ ЗАПИСУ ДО ЛІКАРЯ

2.1 Специфікація вимог до сервісу

Розробка онлайн-сервісу для запису до лікарів – це важливий крок у розвитку медичних послуг, який дозволяє значно спростити та автоматизувати процес планування медичних візитів. Вебзастосунок, який має на меті забезпечити зручний доступ до медичних послуг та оптимізацію процесу запису, може стати потужним інструментом для пацієнтів, лікарів і медичних установ. Сервіс надає можливість користувачам швидко знаходити лікарів потрібної спеціалізації, переглядати їхні графіки роботи, а також записуватись на консультації або лікування через зручний інтерфейс.

Метою розробки цього сервісу є створення платформи, що дозволяє пацієнтам без зайвих труднощів записуватися на прийом до лікаря, враховуючи свої побажання щодо часу, спеціалізації лікаря, місця розташування медичного закладу та інших параметрів. Однією з основних переваг такого підходу є можливість користувачів вибрати лікаря, спираючись на відгуки, рейтинги та доступність в певний час.

Система сервісу має два основних типи користувачів: неавторизованих та авторизованих. Неавторизовані користувачі можуть ознайомитись з основною інформацією про сервіс, переглянути наявні лікарні та лікарів, а також отримати загальні відомості про медичні послуги. Проте вони не можуть записуватися на прийом або зберігати свою медичну історію в системі. Авторизовані користувачі, в свою чергу, отримують доступ до всього функціоналу сервісу, що включає можливість створення записів до лікарів, перегляд та редагування особистих даних, а також доступ до історії попередніх прийомів.

Процес запису до лікаря є інтуїтивно зрозумілим і складається з кількох етапів, що забезпечує користувачеві максимальний комфорт при плануванні

медичних візитів. Першим кроком є введення основних параметрів, таких як спеціалізація лікаря, бажаний час прийому та місце розташування медичного закладу. Після цього система автоматично генерує список доступних варіантів на основі введених даних, включаючи лікарів з відповідною кваліфікацією та вільні часи для прийому.

Користувач має можливість переглядати подробиці про лікарів, включаючи їхні рейтинги, досвід, відгуки пацієнтів та цінову категорію послуг. Якщо один з варіантів задовольняє вимоги користувача, можна безпосередньо забронювати час для візиту. Після підтвердження запису, пацієнт отримує детальну інформацію про медичний заклад, адресу, контактні дані клініки, а також можливість скасувати або перенести прийом, якщо це необхідно.

Один із ключових елементів сервісу – це автоматичне планування запису, що базується на введених користувачем даних. Процес запису до лікаря проходить через кілька етапів:

- введення параметрів запису. Користувач зазначає, до якого лікаря чи клініки він хоче записатися, вказує дату і час прийому, а також додає свої побажання щодо типу консультації або обраної спеціалізації;
- формування пропозицій. Система аналізує введені дані та пропонує оптимальні варіанти запису, враховуючи доступність лікаря, його спеціалізацію та графік прийому;
- деталізація та уточнення. Після отримання початкової пропозиції користувач може переглянути доступні варіанти, вибрати найзручніший час або змінити лікаря, якщо це необхідно;
- підтвердження запису. Після вибору найбільш підходящого варіанту, користувач підтверджує запис до лікаря, отримуючи детальну інформацію про час прийому, адресу клініки та контактні дані.

Важливим елементом цього сервісу є функція збереження медичної історії користувача. Після кожного запису пацієнт може переглядати всі свої попередні візити, дізнатися, який лікар був обраний, та які медичні послуги

були надані. Це дозволяє не тільки зберігати важливу інформацію про здоров'я, але й забезпечує можливість для лікаря швидко ознайомитися з історією хвороби пацієнта, якщо це необхідно для подальшого лікування.

Іншою важливою особливістю сервісу є можливість оцінки вартості медичних послуг. Залежно від вибраної клініки та лікаря, система прогнозує приблизну вартість візиту, що дозволяє користувачеві заздалегідь планувати бюджет. Крім того, платформа надає можливість вибору різних варіантів медичних послуг, щоб забезпечити доступ до найкращих пропозицій для різних категорій пацієнтів.

Ще одним важливим аспектом є наявність функції нагадувань про прийом. Система надсилає пацієнтові автоматичні сповіщення за декілька годин або день до запланованої консультації, що допомагає уникнути забутих візитів або недотримання графіку.

Платформа забезпечує високий рівень безпеки даних, що є критичним для медичних сервісів. Всі особисті дані користувачів, включаючи медичні відомості та історію записів, зберігаються з дотриманням найвищих стандартів конфіденційності та захисту. Це дає можливість користувачам безпечно використовувати платформу для ведення медичних записів і взаємодії з лікарями без побоювань за витік інформації.

2.2 Проектування архітектури сервісу для електронного медичного запису

Проект для створення системи електронного медичного запису передбачає розробку вебзастосунку, що дозволить пацієнтам записуватися на прийом до лікаря через інтерфейс, а адміністраторам забезпечить можливість керувати записами та профілями лікарів. Архітектура системи базуватиметься на сучасному стосунку MERN, що дозволяє створювати масштабовані і високопродуктивні вебзастосунки.

Для реалізації клієнтської частини буде використано React.js, що забезпечить інтерактивність та динамічність інтерфейсу. Реалізація цієї частини передбачає використання таких інструментів, як React Router для навігації по сайту, а також Redux або Context API для керування станом додатку. Це дозволить користувачам мати швидкий доступ до потрібних функцій без постійних перезавантажень сторінок, створюючи комфортну взаємодію з сервісом. Для стилізації інтерфейсу обрано Tailwind CSS, що дозволить створювати адаптивний дизайн, який однаково добре виглядатиме як на мобільних пристроях, так і на десктопах. Такий підхід забезпечить зручність при розробці, дозволяючи швидко та безпечно створювати сторінки з гармонійним дизайном без необхідності писати велику кількість кастомного CSS.

Серверна частина вебзастосунку буде побудована за допомогою Node.js та Express.js, що дозволить створювати легкі та масштабовані сервери з високою продуктивністю. Це дозволить забезпечити ефективне оброблення запитів від користувачів та забезпечити безпечну взаємодію з базою даних. В якості бази даних обрана MongoDB, що є документно-орієнтованою і дозволяє швидко зберігати і обробляти великі об'єми даних. MongoDB ідеально підходить для зберігання таких даних, як профілі пацієнтів, лікарів, записи на прийом та інша необхідна інформація. Для управління авторизацією користувачів застосовуватиметься JWT (JSON Web Token), що дозволить забезпечити безпечний доступ до особистих даних.

Система передбачає два основних типи користувачів: пацієнтів і адміністраторів. Пацієнти зможуть створювати акаунти, обирати лікарів, записуватись на прийом, а також переглядати історію своїх записів. Пацієнти зможуть змінювати дані свого профілю, а також скасувати або змінити існуючі записи на прийом. Адміністратори ж матимуть можливість переглядати загальну статистику записів і лікарів через спеціальний дашборд. Адміністратор зможе додавати нових лікарів до системи, редагувати їх профілі, а також видаляти лікарів та записувати нові дані за необхідністю.

Адміністратор також матиме можливість переглядати всю історію записів і скасовувати або змінювати їх за потреби.

Забезпечення безпеки даних є одним з основних завдань проєкту. Для цього буде використано механізм аутентифікації за допомогою JWT, що дозволить гарантувати захист особистих даних користувачів і лікарів. Також буде застосовуватись шифрування паролів перед їх зберіганням у базі даних, що забезпечить високий рівень конфіденційності. Всі медичні дані, записи та інші важливі файли будуть зберігатись у хмарних сховищах, що дозволить швидко завантажувати та отримувати необхідні файли без навантаження на основний сервер.

Крім того, система повинна мати функцію управління медичними записами, що дозволяє лікарям і пацієнтам легко керувати своїми прийомами через особисті кабінети. Пацієнти, увійшовши в систему, зможуть обирати лікарів з різних спеціалізацій, бачити їхні доступні часи для прийому і записуватись на них. Лікарі ж зможуть оновлювати свої профілі, включаючи інформацію про доступні часи прийому, а також керувати записами на прийом. Адміністратори системи, в свою чергу, матимуть доступ до повного списку лікарів і пацієнтів, а також до статистики за кількістю записів і прийомів.

Також передбачається інтеграція із зовнішніми системами для отримання додаткової інформації та забезпечення повної функціональності. Наприклад, система зможе підключатись до різних медичних баз даних для перевірки актуальності лікарських ліцензій, а також забезпечувати зручність отримання звітів про виконану роботу лікарями та статистику по прийомах. Всі ці функції та можливості разом створюють потужний інструмент для медичних установ, що забезпечує високий рівень сервісу для пацієнтів і лікарів.

2.3 Моделювання структури програми для електронного медичного запису

Моделювання є критично важливим етапом у процесі розробки вебзастосунку, оскільки воно дозволяє чітко визначити архітектуру системи, взаємодію між компонентами та поведінку користувачів. Для застосунку електронного медичного запису було виділено три основні категорії користувачів, кожна з яких має унікальний набір функціональних можливостей:

- неавторизований користувач – має обмежений доступ до системи;
- авторизований користувач (пацієнт) – отримує доступ до основних функцій запису;
- адміністратор – має розширені права для керування системою.

Для наочної візуалізації взаємодії користувачів із системою було розроблено діаграму прецедентів (рис. 2.1), яка відображає всі можливі сценарії використання застосунку.

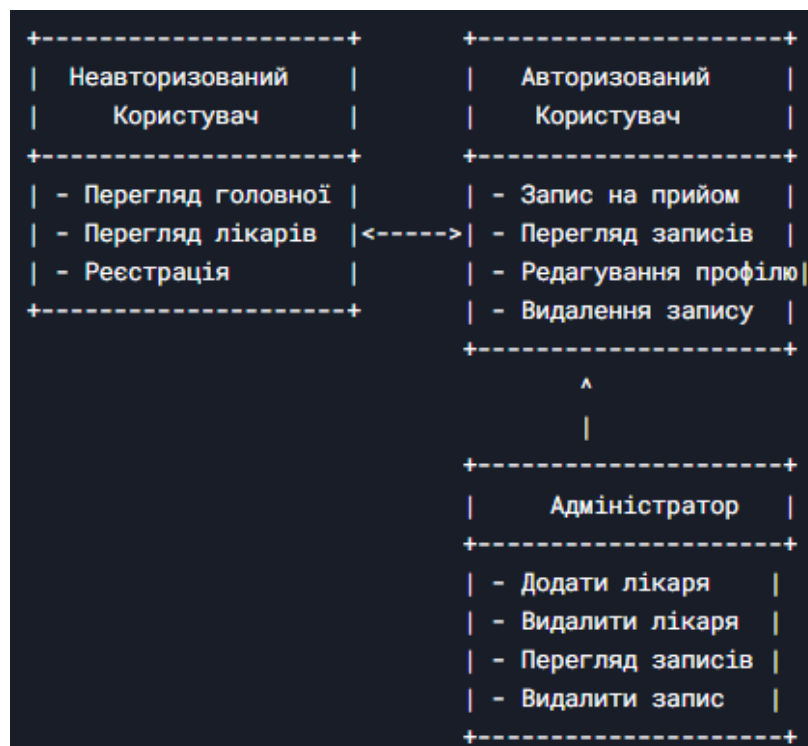


Рисунок 2.1 – Діаграма прецедентів

Діаграма прецедентів демонструє наступні функціональні можливості застосунку.

Для неавторизованих користувачів:

- перегляд головної сторінки з інформацією про сервіс;
- огляд списку лікарів, відсортованих за спеціальностями;
- можливість реєстрації в системі через електронну пошту або соціальні мережі.

Для авторизованих користувачів (пацієнтів):

- запис на прийом до лікаря з вибором дати та часу;
- перегляд активних та минулих записів у особистому кабінеті;
- редагування персональних даних профілю;
- скасування запланованих записів.

Для адміністраторів:

- перегляд аналітичної панелі з статистикою (кількість пацієнтів, записів, лікарів);
- додавання нових лікарів до системи з указанням спеціальності та графіку роботи;
- видалення або редагування профілів лікарів;
- керування списком записів (перегляд, фільтрація, видалення).

Діаграма послідовності авторизації (рис. 2.2) деталізує процес входу користувача в систему:

- користувач вводить облікові дані (електронна пошта/пароль) або обирає авторизацію через Google;
- Frontend-частина застосунку відправляє запит на сервер для перевірки даних;
- сервер перевіряє надані дані, звертаючись до бази даних MongoDB;
- після успішної валідації сервер створює сесію та генерує JWT-токен;
- користувач отримує доступ до особистого кабінету, де може керувати записами;

– цей процес забезпечує безпеку даних та захищає систему від несанкціонованого доступу.

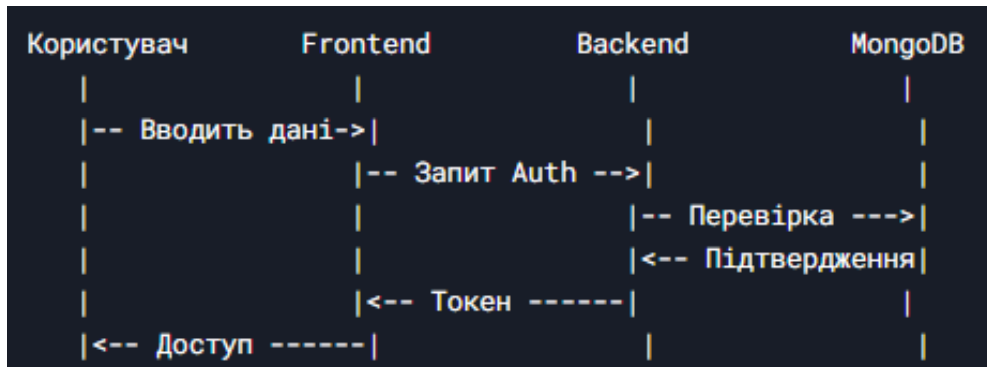


Рисунок 2.2 – Діаграма послідовності для авторизації користувача

Діаграма послідовності запису на прийом (рис. 2.3) ілюструє механізм взаємодії між користувачем та системою:

- користувач обирає лікаря зі списку доступних спеціалістів;
- система відображає вільні слоти для запису на основі графіку лікаря;
- користувач обирає зручну дату та час, після чого підтверджує запис;
- Frontend відправляє запит на сервер, який перевіряє доступність обраного слота;
- якщо слот вільний, сервер зберігає запис у базі даних та надсилає підтвердження;
- користувач отримує сповіщення про успішний запис, а інтерфейс оновлюється.

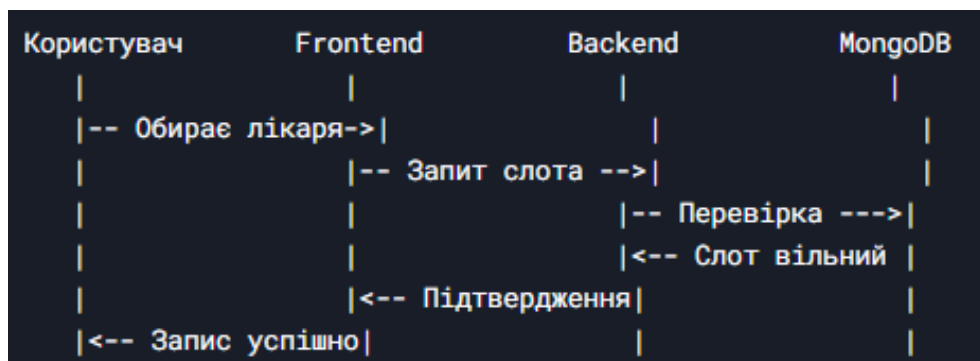


Рисунок 2.3 – Діаграма послідовності для запису на прийом

У разі зайнятості обраного часу система пропонує альтернативні варіанти.

Діаграма класів є ключовим інструментом для візуалізації структури даних системи та взаємозв'язків між основними сутностями. У контексті застосунку для електронного медичного запису було виділено три центральні класи: User (Користувач), Doctor (Лікар) та Appointment (Запис на прийом). Кожен з цих класів містить атрибути, які описують їхні властивості, а також зв'язки між ними (рис. 2.4).

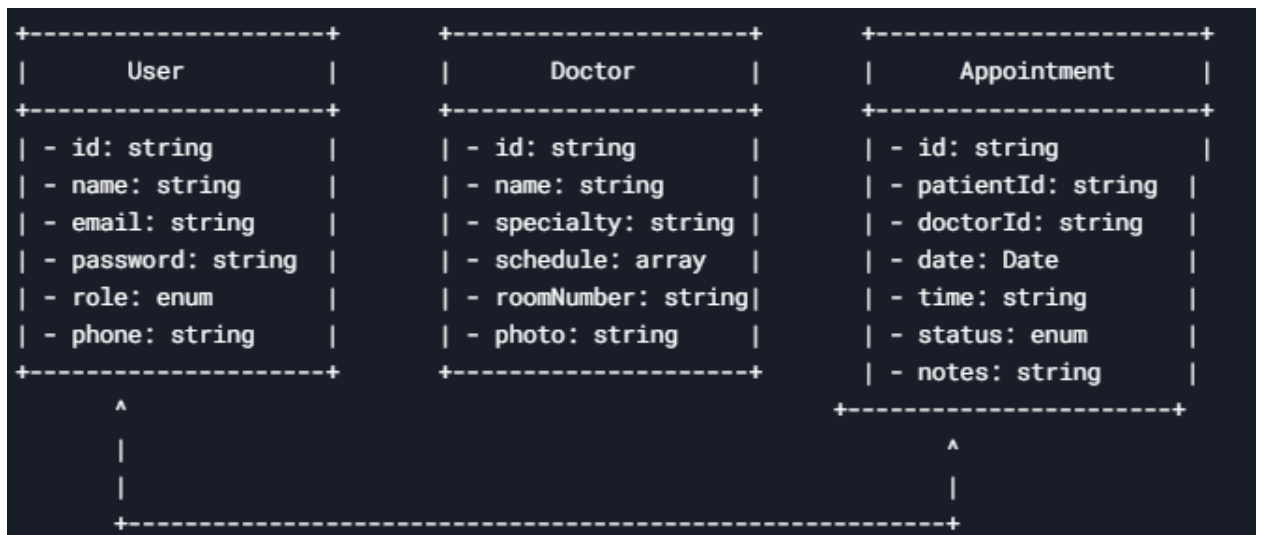


Рисунок 2.4 – Діаграма класів

Клас User (Користувач) відповідає за облікові дані всіх користувачів системи (пацієнтів та адміністраторів):

- id – унікальний ідентифікатор;
- name – повне ім'я користувача;
- email – електронна пошта для авторизації;
- password – захешований пароль;
- role – роль (пацієнт, адміністратор);
- phone – контактний телефон (опціонально).

Клас Doctor (Лікар) містить інформацію про лікарів, які працюють у системі:

- id – унікальний ідентифікатор;
- name – ім'я та прізвище лікаря;
- specialty – спеціальність (терапевт, хірург тощо);
- schedule – масив з графіком роботи (наприклад, {day: "Monday", start: "09:00", end: "17:00"});
- roomNumber – номер кабінету;
- photo – посилання на фото лікаря.

Клас Appointment (Запис на прийом) відображає записи пацієнтів до лікарів:

- id – унікальний ідентифікатор запису;
- patientId – зв'язок з класом User (хто записався);
- doctorId – зв'язок з класом Doctor (до кого запис);
- date – дата прийому;
- time – час прийому;
- status – статус (активний, скасований, завершений);
- notes – додаткові нотатки (наприклад, симптоми).

2.4 Клієнтська частина застосунку для електронного медичного запису

Frontend-частина медичного застосунку розроблена на сучасному стеку технологій MERN (MongoDB, Express.js, React, Node.js), що забезпечує високу продуктивність, масштабованість та зручний користувацький досвід. Основу архітектури складає React JS, який дозволив створити динамічний та інтерактивний інтерфейс з можливістю швидкого оновлення даних без перезавантаження сторінки. Для побудови користувацького інтерфейсу використано сучасні підходи компонентної розробки, що дозволило створити модульну та легко підтримувану структуру коду [16-19].

Стилізація реалізована за допомогою комбінації CSS Modules та бібліотеки Material-UI, що забезпечило як гнучкість у дизайні, так і готові

компоненти для прискорення розробки. Для стану додатка використано Redux Toolkit, який ефективно керує глобальним станом, особливо при роботі з даними про лікарів, записи та користувацькі налаштування. Маршрутизація реалізована через React Router, що забезпечує плавну навігацію між різними розділами застосунку без перезавантаження сторінки.

Особливу увагу приділено інтерфейсу запису на прийом, де реалізовано інтуїтивно зрозумілий календар з відображенням доступних слотів, що інтегрується з бекендом через REST API. Для відображення списку лікарів створено компонент з фільтрацією за спеціальністю, рейтингом та доступністю, що дозволяє користувачам швидко знаходити потрібного фахівця. Форма запису включає вибір дати, часу, а також можливість додати коментар або симптоми, що передається лікарю.

Для адміністративної панелі розроблено окремий інтерфейс з розширеними функціями управління. Він включає dashboard з аналітикою (кількість пацієнтів, записів, лікарів), інструменти для додавання/редагування профілів лікарів з можливістю завантаження фото та детальної інформації про спеціалізацію, а також комплексну систему управління записами з фільтрацією та пошуком. Всі зміни в адмінці відображаються в реальному часі завдяки оптимізованій архітектурі взаємодії з бекендом.

Система автентифікації реалізована з розділенням на три рівні доступу: гості (можуть переглядати головну сторінку та список лікарів), авторизовані користувачі (мають доступ до особистого кабінету та записів) та адміністратори (повний доступ до управління системою). Для забезпечення безпеки використано JWT-токени, що зберігаються в httpOnly cookies, а також механізми перевірки прав доступу на рівні як фронтенду, так і бекенду.

Інтерфейс повністю адаптований для різних пристроїв – від десктопів до мобільних телефонів, що досягнуто за допомогою responsive дизайну з використанням медіа-запитів та гнучких компонентів. Для покращення продуктивності реалізовано lazy loading компонентів, оптимізовано

зображення та використано мемоізацію для часто оновлюваних елементів інтерфейсу.

Особливістю рішення стала інтеграція системи сповіщень, яка інформує користувачів про підтвердження запису, нагадування про майбутні візити та зміни в розкладі лікарів. Для адміністраторів реалізовано систему оповіщень про нові записи та критичні події в системі. Всі інтерфейсні компоненти пройшли тестування на юзабіліті, що дозволило створити інтуїтивно зрозумілий та зручний для всіх категорій користувачів застосунок.

2.5 Серверна частина застосунку для електронного медичного запису

Backend-архітектура нашого медичного застосунку являє собою високотехнологічну багаторівневу систему, розроблену на основі Node.js та Express.js, яка забезпечує надійну та безпечну роботу всіх функціональних модулів. Основним компонентом бекенду є RESTful API, що побудоване з урахуванням сучасних стандартів веброзробки та медичної інформатики. Система використовує MongoDB як основну базу даних, що дозволяє ефективно зберігати та обробляти структуровані та напівструктуровані дані, такі як профілі користувачів, медичні записи, розклади лікарів та історію прийомів. Для забезпечення високої продуктивності та масштабованості реалізовано кешування даних за допомогою Redis, що особливо важливо для операцій, які вимагають швидкого доступу до часто використовуваної інформації, таких як перегляд доступних слотів для запису або пошук лікарів за спеціальністю.

Автентифікація та авторизація користувачів реалізована через JWT (JSON Web Tokens) з використанням httpOnly cookies для підвищення безпеки. Система розділяє користувачів на три основні ролі: гості, авторизовані пацієнти та адміністратори, кожна з яких має чітко визначений набір прав доступу. Для обробки медичних даних реалізовано спеціальний модуль, який

відповідає за валідацію, нормалізацію та зберігання інформації про стан здоров'я пацієнтів, діагнози лікарів та призначені курси лікування. Цей модуль інтегрується з системою електронних медичних записів (EMR) та підтримує експорт даних у стандартизованих форматах, таких як HL7 FHIR для забезпечення сумісності з іншими медичними системами.

Для обробки запитів, пов'язаних із записом на прийом, створено окремий сервіс, який включає складну логіку перевірки доступності лікарів, управління розкладами та обробки повторюваних записів. Цей сервіс інтегрується з системою календарного планування та відправляє автоматичні сповіщення пацієнтам та лікарям про майбутні візити, зміни в розкладі або скасування прийомів. Для забезпечення високої доступності сервісу реалізовано механізми балансування навантаження та відмовостійкості, що дозволяє системі коректно працювати навіть у разі часткових збоїв окремих компонентів.

Система звітності та аналітики дозволяє адміністраторам отримувати детальну статистику про роботу медичного закладу, включаючи дані про завантаженість лікарів, популярність спеціальностей, середній час очікування прийому та інші ключові показники ефективності. Для обробки великих обсягів даних використовується агрегаційний фреймворк MongoDB, що дозволяє генерувати складні звіти в реальному часі без значного навантаження на систему.

Інтеграційний шар бекенду забезпечує зв'язок із зовнішніми системами, такими як платіжні шлюзи, лабораторні інформаційні системи (LIS) та телемедичні платформи. Для кожного типу інтеграції реалізовано окремі адаптери, які абстрагують специфіку роботи з зовнішніми API та забезпечують стабільну роботу навіть при тимчасовій недоступності зовнішніх сервісів. Система черг повідомлень на основі RabbitMQ використовується для асинхронної обробки запитів, що не вимагають миттєвої реакції, таких як відправка електронних листів або SMS-сповіщень.

Для забезпечення безпеки медичних даних реалізовано комплексний підхід, який включає шифрування даних як під час передачі (TLS), так і під час зберігання, деталізоване логування всіх операцій з конфіденційною інформацією, регулярні аудити безпеки та механізми автоматичного блокування підозрілих дій. Система відповідає вимогам GDPR та інших регуляторних стандартів у сфері захисту медичних даних.

Моніторинг продуктивності системи здійснюється за допомогою спеціалізованих інструментів, таких як Prometheus для збору метрик та Grafana для їх візуалізації. Це дозволяє оперативно виявляти потенційні проблеми з продуктивністю та масштабувати систему відповідно до зростаючого навантаження. Для автоматизації процесів розгортання та управління інфраструктурою використовується Docker у поєднанні з Kubernetes, що забезпечує високу доступність сервісу та можливість швидкого масштабування окремих компонентів системи.

Тестування бекенд-системи включає кілька рівнів: модульні тести для окремих функцій, інтеграційні тести для перевірки взаємодії компонентів, навантажувальні тести для оцінки продуктивності під високим навантаженням та тести безпеки для виявлення потенційних вразливостей. Вся інфраструктура тестування інтегрована в CI/CD пайплайн, що дозволяє автоматично перевіряти якість коду при кожному оновленні системи.

Архітектура бекенду розроблена з урахуванням майбутнього розширення функціоналу, включаючи можливість інтеграції зі штучним інтелектом для аналізу медичних даних, підтримку блокчейн-технологій для забезпечення незмінності медичних записів та розвиток телемедичних можливостей для віддалених консультацій. Система має чітку модульну структуру, що дозволяє легко додавати нові функції без необхідності значних змін у існуючому коді, забезпечуючи таким чином довгострокову підтримку та розвиток застосунку.

3 РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ЕЛЕКТРОННОГО МЕДИЧНОГО ЗАПИСУ

3.1 Вибір інструментальних засобів для реалізації поставленої задачі

Як основу клієнтської частини було обрано React у поєднанні з TypeScript, що дозволило поєднати переваги компонентного підходу зі строгою типізацією, критично важливою для медичних додатків, де точність даних є пріоритетом. Використання Next.js як фреймворку додало можливості серверного рендерингу та статичної генерації сторінок, що значно покращило SEO-оптимізацію та швидкість завантаження інтерфейсу, особливо важливу для користувачів з обмеженим інтернет-з'єднанням.

Для побудови інтерфейсу адміністративної панелі та пацієнтського кабінету було інтегровано Material-UI як основну бібліотеку компонентів, що забезпечило готові рішення для доступності (a11y) та адаптивності, а також Tailwind CSS для кастомізації дизайн-системи під специфічні вимоги медичного інтерфейсу. Стилізація форм введення медичних даних була особливо ретельно продумана - використано комбінацію CSS-in-JS (Emotion) для динамічних стилів та SCSS для глобальних стилів, що дозволило досягти балансу між продуктивністю та гнучкістю.

Серверна частина реалізована на Node.js з використанням Express.js, що забезпечило високу продуктивність при обробці медичних запитів та масштабованість системи. Як основну базу даних обрано MongoDB через її гнучкість у роботі з напівструктурованими медичними даними, зокрема для зберігання історії хвороб, результатів аналізів та складних медичних протоколів. Для підвищення швидкодії реалізовано кешування через Redis, особливо ефективно для частих запитів до розкладів лікарів та доступних слотів запису.

Система автентифікації побудована на JWT з додатковими механізмами безпеки, включаючи багатофакторну автентифікацію для лікарів та адміністраторів. Для керування доступом реалізовано рольову модель (RBAC) з п'ятьма рівнями прав: гость, пацієнт, лікар, адміністратор клініки, суперадміністратор. Кожен рівень має чітко визначені permissions, що контролюються на рівні middleware Express.js.

Для обробки медичних зображень та документів інтегровано AWS S3 з власними сервісами препроцесингу, що дозволяє автоматично анонімізувати чутливі дані на зображеннях (наприклад, прикривати особливі ознаки на рентгенографіях) перед зберіганням. Система документообігу побудована на PDFKit для генерації медичних висновків та направлень у стандартизованих форматах, сумісних з вимогами МОЗ.

Інтеграція з зовнішніми медичними системами реалізована через REST API та WebSockets для реального часу, з підтримкою стандартів HL7 FHIR для обміну медичними даними. Для телемедицини використано WebRTC з власними серверами TURN/STUN, що забезпечують конфіденційність відеоконсультацій. Платформа включає модуль машинного навчання на Python (інтеграція через gRPC) для аналізу медичних даних та підказок діагнозів.

Розробницький процес включав сучасні інструменти CI/CD – GitHub Actions для автоматизованого тестування, Docker та Kubernetes для контейнеризації та оркестрації, Prometheus+Grafana для моніторингу продуктивності в реальному часі. Тестування охоплювало модульні тести (Jest), інтеграційні тести (Cypress), навантажувальні тести (k6) та аудит безпеки (OWASP ZAP).

Вибір технологій для мобільної версії базувався на React Native з використанням 85% спільного коду з вебверсією, що значно прискорило розробку. Для офлайн-роботи реалізовано синхронізацію даних через PouchDB/CouchDB, критично важливу для роботи в умовах нестабільного інтернет-з'єднання в віддалених регіонах [20-25].

Спеціалізовані медичні модулі включають: систему електронного рецептування з інтеграцією до державного реєстру, інструменти для дистанційного моніторингу стану пацієнтів через IoT-пристрої, модуль штучного інтелекту для аналізу медичних зображень (інтеграція з TensorFlow.js), а також систему керування медичними протоколами з автоматичним оновленням згідно з міжнародними рекомендаціями.

3.2 Архітектура системи та структура застосунку

Система електронного запису до лікаря створена за допомогою MERN-стека, який об'єднує чотири технології: MongoDB, Express.js, React і Node.js. Такий підхід дозволяє розробити швидкий, надійний та зручний у користуванні вебзастосунок.

Інтерфейс користувача побудований за допомогою React, що дозволяє миттєво змінювати сторінки без їх повного оновлення. Усі частини інтерфейсу організовані в окремі компоненти відповідно до ролі користувача:

- домашня сторінка – загальний огляд, інформація про лікарів та спеціальності;
- сторінка лікаря – профіль спеціаліста з можливістю запису;
- кабінет користувача – управління особистими записами;
- адмін-панель – сторінка для адміністратора з розширеними функціями керування.

Ключові модулі:

- аутентифікація (Auth) – вхід, реєстрація, перевірка прав доступу через токени;
- записи (Appointments) – запис на прийом, редагування або скасування;
- лікарі (Doctors) – перегляд лікарів та їх спеціалізацій;

- адмінка (Admin Dashboard) – адміністрування користувачів, лікарів, записів;
- профіль (User Profile) – зміна особистої інформації та перегляд історії записів;
- серверна логіка (API Server) – обробка запитів до бази даних.

Back-end частина працює на Node.js, а Express.js забезпечує маршрутизацію та логіку обробки запитів. Всі дані зберігаються в MongoDB, де основні колекції:

- users – дані пацієнтів, логіни, ролі;
- doctors – інформація про медиків і графіки прийому;
- appointments – бронювання прийомів, дата, час, статус;
- admin – службові записи для адміністративної ролі.

Навігація в React-застосунку побудована таким чином:

- / – домашня сторінка;
- /login, /register – вхід та реєстрація;
- /doctors – список медиків;
- /doctor/:id – перегляд окремого лікаря;
- /appointments – сторінка записів користувача;
- /profile – кабінет пацієнта;
- /admin/dashboard – доступний лише адміністраторам.

3.3 Основні можливості застосунку

Основною метою створення цієї системи є забезпечення зручного та швидкого способу запису до лікаря в онлайн-режимі. Завдяки цьому сервісу пацієнти можуть уникнути черг, легко знаходити потрібного фахівця та керувати своїми візитами, а адміністрація – контролювати всі процеси,

пов'язані з медичним обслуговуванням. Усі функції були розроблені з урахуванням реальних потреб користувачів і медичних закладів.

3.3.1 Реєстрація та вхід

Система підтримує авторизацію двох основних типів користувачів: пацієнтів (звичайних користувачів) та адміністраторів (персонал, який має доступ до керування даними). Реєстрація проходить через просту форму, де потрібно ввести особисту інформацію: ім'я, електронну пошту, пароль та інші дані. Після успішної реєстрації користувач автоматично входить у систему.

При вході система перевіряє правильність введених даних і, якщо вони відповідають існуючим у базі, генерує унікальний токен авторизації (JWT), який зберігається в браузері. Це дозволяє зберігати сеанс користувача навіть після перезавантаження сторінки.

З міркувань безпеки, усі паролі користувачів шифруються за допомогою алгоритму bcrypt і не зберігаються у відкритому вигляді. У разі неправильного вводу даних користувач отримує сповіщення з поясненням помилки (наприклад, «невірний пароль» або «такий користувач не існує»).

3.3.2 Запис до лікаря

Одна з основних функцій застосунку – можливість онлайн-запису на прийом до обраного лікаря. Після входу в систему користувач переходить до розділу «Лікарі», де доступний список спеціалістів з короткою інформацією (ім'я, спеціальність, досвід, фото). Клікнувши на профіль лікаря, користувач бачить детальну інформацію про нього, включно з вільними датами та часом для запису.

Інтерфейс вибору зручного часу інтуїтивно зрозумілий: користувач бачить календар, де вже заброньовані години неактивні. Після підтвердження запису дані зберігаються у базі, і система надсилає підтвердження (наприклад, у вигляді сповіщення на екрані).

Також користувач у будь-який момент може змінити час прийому або повністю скасувати запис. Усі ці дії синхронізуються із системою та автоматично оновлюють графік доступності лікаря.

3.3.3 Панель адміністратора

Для адміністраторів передбачений окремий розділ системи – адмін-панель, доступ до якого відкривається лише за наявності відповідних прав (роль admin). Цей інтерфейс містить функції для повного контролю над усією системою:

- статистика: в адмінці відображається загальна інформація по системі – кількість зареєстрованих пацієнтів, активних записів, кількість лікарів. це дозволяє швидко оцінювати завантаженість закладу;
- керування записами: адміністратор може переглядати, змінювати або видаляти записи пацієнтів. це корисно в разі технічних помилок або за потреби ручного втручання;
- додавання/видалення лікарів: через панель можна створити новий профіль лікаря, вказати його спеціальність, досвід, розклад, завантажити фото тощо. за потреби профіль лікаря можна деактивувати або видалити;
- журнал активностей (опційно): у систему можна додати логування дій користувачів – реєстрація, вхід, зміна даних, створення записів. це дозволяє аналізувати поведінку користувачів і виявляти підозрілу активність.

3.3.4 Особистий кабінет пацієнта

Після входу пацієнт отримує доступ до персонального кабінету, де зібрані всі важливі функції:

- редагування профілю: користувач може змінити свої особисті дані – ім'я, електронну пошту, номер телефону тощо. усі зміни зберігаються одразу в базі;

- історія записів: у цьому розділі відображаються всі попередні та майбутні записи до лікаря. для зручності записи розділені за статусами: «майбутній», «пройдено», «скасовано»;

- можливість скасування або зміни запису: якщо користувач змінює плани, він може скасувати запис або вибрати іншу дату і час.

Цей функціонал дозволяє пацієнтам керувати своїм медичним обслуговуванням повністю самостійно, без необхідності дзвонити в клініку.

3.3.5 Робота з файлами (за потреби)

У перспективі система може бути розширена можливістю завантаження файлів. Наприклад:

- фото лікарів – для покращення зручності перегляду профілів медиків;
- документи пацієнтів – направлення, медичні довідки, результати аналізів.

Для зберігання таких файлів використовується зовнішнє хмарне сховище – наприклад, Cloudinary або Firebase Storage. Після завантаження система зберігає лише посилання на файл у базі даних, а сам файл зберігається у хмарі, що дозволяє зменшити навантаження на сервер.

Користувач або адміністратор зможе переглядати ці документи, завантажувати їх або видаляти за потреби.

3.4 Тестування роботи застосунку

Для перевірки коректності реалізації електронної системи медичних записів було проведено всебічне тестування функціональності як з боку звичайного користувача (пацієнта), так і з боку адміністратора системи. Ретельно опрацьовано всі основні користувацькі сценарії: від завантаження головної сторінки та авторизації до запису на прийом, редагування профілю, адміністрування лікарів і контролю за всіма записами. Метою тестування є перевірка правильності роботи користувацького інтерфейсу, надійності обробки запитів, взаємодії з базою даних, а також логіки захисту та контролю доступу.

Для кожного етапу були зроблені скріншоти, які ілюструють як виглядає інтерфейс на тому чи іншому кроці, а також підтверджують правильну роботу застосунку у відповідності до очікуваних сценаріїв використання.

Після переходу на головну сторінку застосунку користувач, незалежно від наявності облікового запису, отримує доступ до базової інформації про сервіс (рис. 3.1). Тут міститься короткий опис платформи, її переваг, можливостей, а також представлено перелік лікарів, які вже додані до системи. Всі лікарі розділені за категоріями відповідно до своєї спеціалізації, що значно спрощує пошук потрібного фахівця. Відображення відбувається у вигляді зручних карток із фото, ім'ям, спеціальністю та кнопкою перегляду детальнішої інформації.

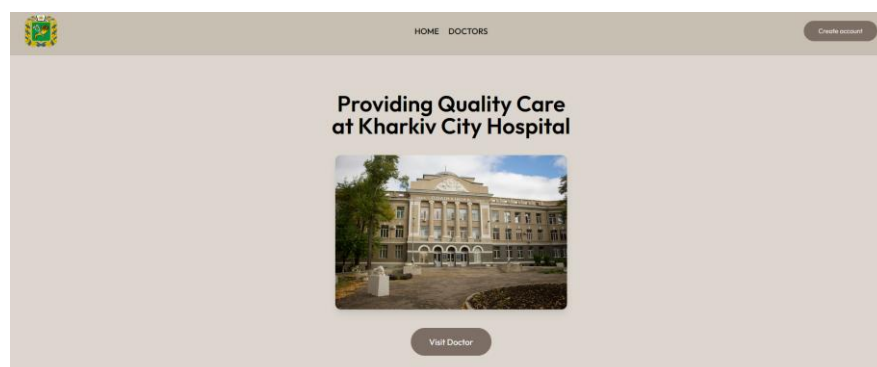


Рисунок 3.1 – Завантаження головної сторінки

У верхній частині сторінки також присутня кнопка «Створити аккаунт», яка дозволяє користувачу отримати повний доступ до функціоналу системи. Незареєстрований користувач може ознайомитися з інтерфейсом, але не має змоги записатися на прийом або редагувати дані.

Процес реєстрації нового користувача побудований інтуїтивно. Для створення облікового запису необхідно ввести базові персональні дані: ім'я, електронну пошту та пароль. Після успішної реєстрації користувач автоматично входить до системи та перенаправляється на головну сторінку вже в авторизованому стані (рис. 3.2).

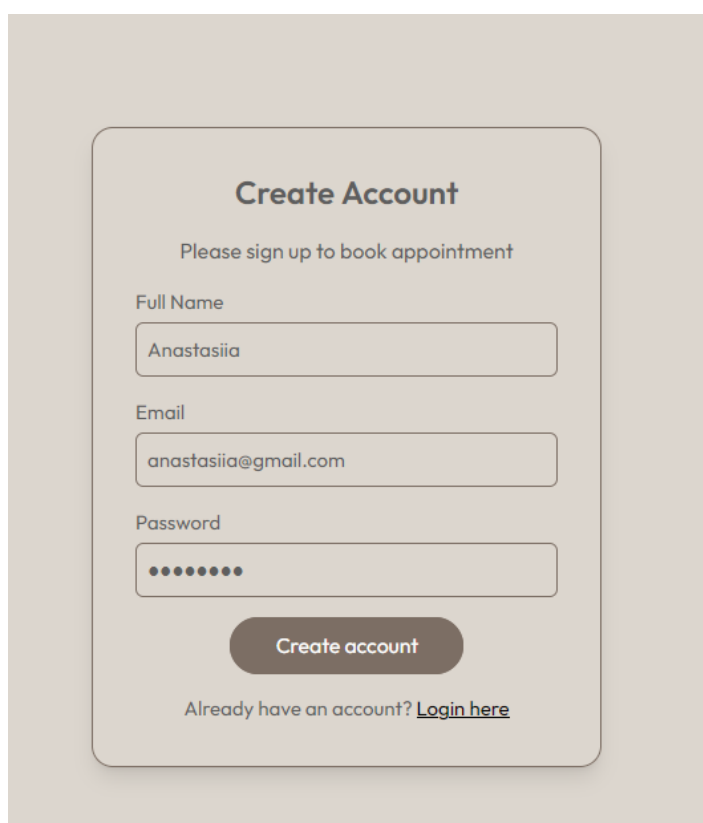
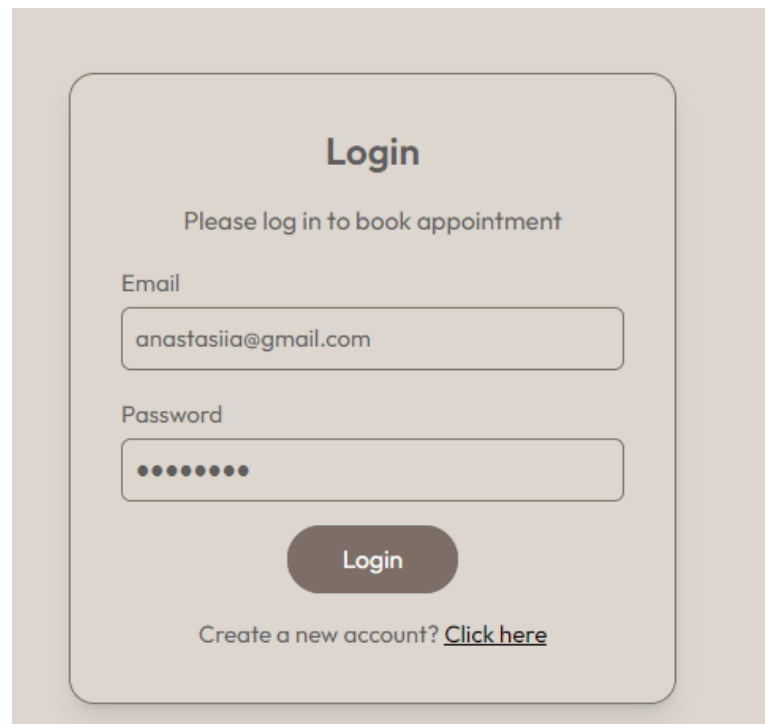
The image shows a registration form titled "Create Account" on a light gray background. The form is enclosed in a rounded rectangle with a thin border. At the top, the title "Create Account" is centered in a bold, dark font. Below the title is a subtitle: "Please sign up to book appointment". The form contains three input fields: "Full Name" with the text "Anastasiia", "Email" with the text "anastasiia@gmail.com", and "Password" with a masked password represented by seven dots. Below the input fields is a dark, rounded button with the text "Create account". At the bottom of the form, there is a link: "Already have an account? [Login here](#)".

Рисунок 3.2 – Реєстрація користувача

Авторизація виконується за допомогою звичайного логіна та пароля. Система перевіряє коректність введених даних, і у випадку помилки надає відповідне повідомлення (рис. 3.3). Важливо, що дані користувача передаються безпечно через захищене з'єднання, а паролі зберігаються у базі у хешованому вигляді.



The image shows a login form titled "Login" with the instruction "Please log in to book appointment". It contains two input fields: "Email" with the value "anastasiia@gmail.com" and "Password" with masked characters. A "Login" button is positioned below the fields. At the bottom, there is a link: "Create a new account? [Click here](#)".

Рисунок 3.3 – Авторизація користувача

Після входу до системи користувач має доступ до повного списку лікарів. Усі лікарі згруповані за спеціальностями – це дозволяє швидко знаходити потрібного фахівця. Картка лікаря містить таку інформацію, як ім'я, спеціалізація, досвід роботи, короткий опис і рейтинг (рис. 3.4).



The image displays a list of medical specialties on the left and a doctor's profile card on the right. The specialties are: General physician, Gynecologist (highlighted), Dermatologist, Pediatricians, Neurologist, and Gastroenterologist. The profile card for "Oksana" (Gynecologist) shows a "No Photo" placeholder, a green dot indicating "Available", and the name "Oksana" with the specialty "Gynecologist".

Рисунок 3.4 – Перегляд лікарів і вибір спеціальності

Користувач може натиснути на обраного врача, що перенаправляє до форми вибору дати та часу візиту. Дана кнопка активна лише для авторизованих користувачів.

Після вибору лікаря система відкриває форму, де користувач може обрати доступні дні та години для запису. Вільні слоти відображаються у вигляді інтерактивного календаря. Користувач обирає зручну дату, вводить причину звернення або інші коментарі, після чого натискає кнопку «Записатися на прийом» (рис. 3.5).

Система миттєво записує цю інформацію до бази даних, і у профілі користувача з'являється новий запис. При цьому користувач бачить повідомлення про успішне бронювання. Запис одразу стає видимим і для адміністратора системи (рис. 3.6).

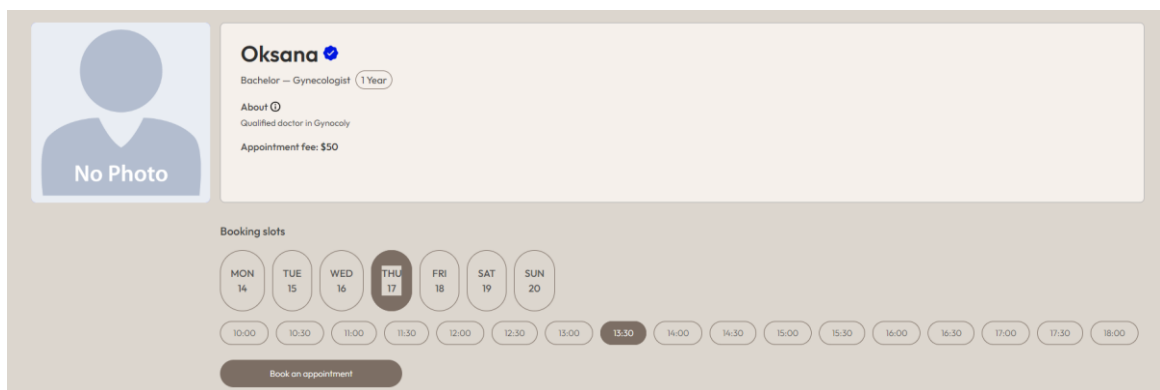


Рисунок 3.5 – Сторінка вибору дати та часу візиту до лікаря

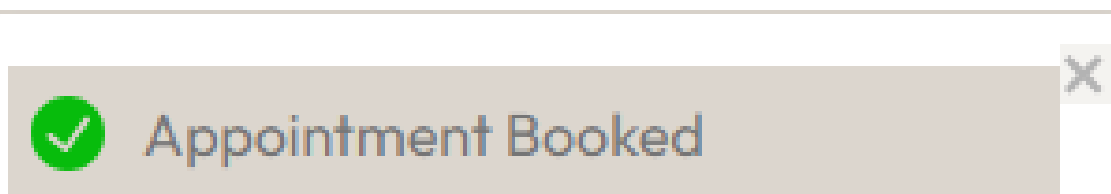


Рисунок 3.6 – Повідомлення про успішне бронювання

У розділі «Мій профіль» користувач може змінювати персональну інформацію про себе, таку як, адресу, телефон, ім'я, обновлювати фото профілю тощо (рис. 3.7).

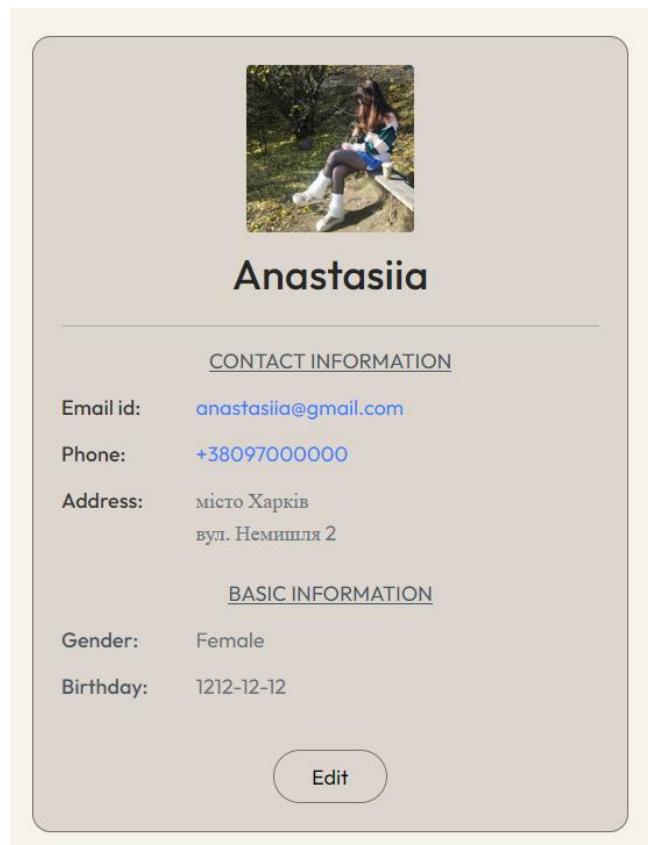


Рисунок 3.7 – Розділ «Мій профіль»

У розділі «Мої записи» користувач бачить список усіх своїх записів до лікарів (рис. 3.8). Тут також реалізовано можливість скасування запису або перегляду детальної інформації про візит. Якщо користувач вирішив, що візит неактуальний – він може натиснути кнопку «Видалити», після чого запис буде вилучений із бази та активна кнопка деактивується (рис. 3.9).

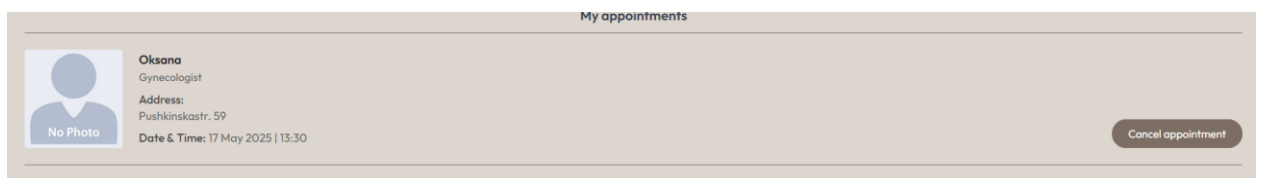


Рисунок 3.8 – Розділ «Мої записи»

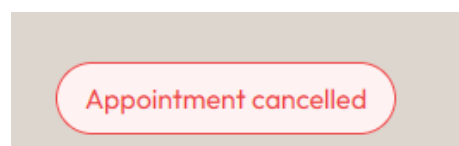


Рисунок 3.9 – Запис відмінений

Після авторизації під обліковим записом адміністратора відкривається адмін-панель, яка містить зведену статистику про роботу платформи. Тут відображаються ключові метрики: кількість зареєстрованих пацієнтів, кількість створених записів, загальна кількість лікарів у базі.

Візуально інформація представлена у вигляді блоків із цифрами, що дозволяє адміністратору оперативно аналізувати стан системи (рис. 3.10).

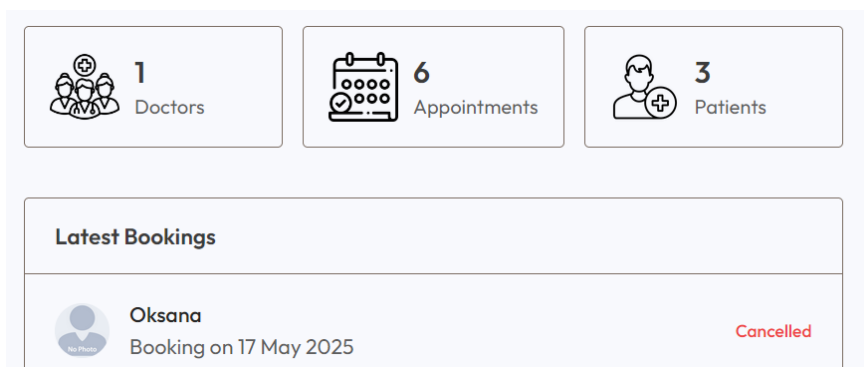


Рисунок 3.10 – Інформація у адмін-панелі

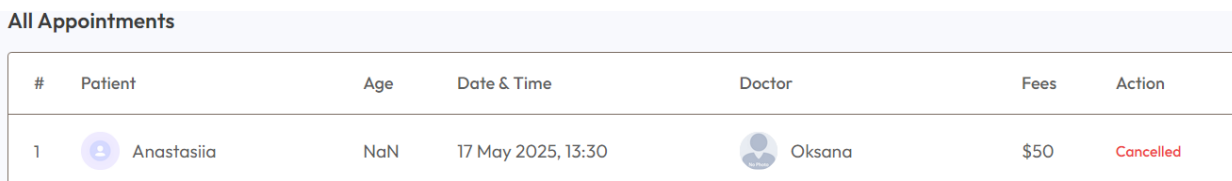
Адміністратор має доступ до розділу «Лікарі», де може переглядати повний список фахівців, редагувати їх дані або видаляти зі списку. Крім того, присутня форма для додавання нового лікаря, яка дозволяє ввести всі необхідні дані: ім'я, спеціалізацію, досвід, контактну інформацію, а також завантажити фотографію (рис. 3.11).

The form for adding a new doctor includes the following fields: 'Your name' (text input with 'Nikita'), 'Speciality' (dropdown menu with 'Neurologist'), 'Experience' (dropdown menu with '10 Years'), 'Degree' (text input with 'Master'), 'Fees' (text input with '100'), and 'Address' (text input with 'Kharkiv'). There is also an 'Upload doctor picture' section with a 'No Photo' icon. A large text area labeled 'About Doctor' contains the text 'Good doctor'. An 'Add doctor' button is located at the bottom.

Рисунок 3.11 – Форма для додавання нового лікаря

Після натискання кнопки «Зберегти», інформація одразу з'являється на сайті й стає доступною для перегляду користувачами.

У відповідному розділі адміністративної панелі адміністратор може переглядати всі записи, створені пацієнтами. Якщо якийсь запис потрібно видалити – наприклад, через помилку або за зверненням користувача – це можна зробити в один клік. Система одразу оновлює інформацію у базі, і запис зникає з профілю пацієнта (рис. 3.12).





#	Patient	Age	Date & Time	Doctor	Fees	Action
1	 Anastasiia	NaN	17 May 2025, 13:30	 Oksana	\$50	Cancelled

Рисунок 3.12 – Перегляд записів

ВИСНОВКИ

У межах кваліфікаційної роботи було розроблено вебзастосунок для електронного управління медичними записами, який забезпечує зручний онлайн-запис на прийом до лікаря, а також ефективне адміністрування пацієнтських візитів. Метою системи є цифровізація процесу медичного обслуговування, підвищення доступності для користувачів та оптимізація щоденної роботи медичних закладів.

У ході виконання роботи було досягнуто всіх поставлених цілей:

- здійснено аналіз актуальних проблем у сфері запису до лікаря, що дозволило виділити ключові вимоги до системи: простота взаємодії, безпечність зберігання даних, підтримка розмежування ролей та можливість швидкого управління записами;

- спроектовано архітектуру вебзастосунку на основі клієнт-серверної моделі з використанням стеку MERN (MongoDB, Express.js, React, Node.js), що дозволило забезпечити масштабованість, швидкодію та зручну взаємодію між клієнтом і сервером;

- реалізовано систему реєстрації та входу з безпечним зберіганням паролів (через bcrypt) та авторизацією на основі JWT-токенів, що гарантує захист персональних даних користувачів;

- створено функціонал онлайн-запису до лікаря з перевіркою доступності часу, можливістю перегляду, редагування та скасування записів, що значно покращує користувацький досвід і зменшує навантаження на персонал;

- розроблено окрему адміністративну панель, яка дозволяє керувати всіма користувачами, лікарями та записами, а також (опційно) переглядати статистику та журнал дій;

- реалізовано особистий кабінет пацієнта з функцією перегляду історії візитів, керування профілем і записами;

– передбачено можливість подальшого розширення системи: зокрема, додавання функцій завантаження медичних документів або інтеграції з зовнішніми медичними базами.

Результатом роботи є функціональний, безпечний і адаптивний вебзастосунок, який може бути використаний як у невеликих приватних клініках, так і в більших медичних установах. Система зручна як для пацієнтів, так і для адміністративного персоналу, забезпечуючи прозорий та ефективний процес планування медичних візитів.

Результати роботи апробовано у вигляді тез доповіді під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ У XXI СТОЛІТТІ» [26].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Hossain, M. S., & Muhammad, G. (2020). Healthcare Big Data Voice Pathology Monitoring System. *IEEE Access*, 8, 12488–12496.
2. Concurrence of technology and healthcare: Electronic health records. *Journal of Family Medicine and Primary Care*, 10(1), 15–19.
3. Dey, N., Ashour, A. S., & Balas, V. E. (2017). *Smart Medical Data Sensing and IoT Systems Design in Healthcare*. Springer.
4. React – A JavaScript library for building user interfaces. URL: <https://reactjs.org/> (дата звернення 10.04.2025).
5. Node.js – JavaScript runtime built on Chrome’s V8 engine. URL: <https://nodejs.org/> (дата звернення 10.04.2025).
6. Express.js – Fast, unopinionated, minimalist web framework for Node.js. URL: <https://expressjs.com/> (дата звернення 10.04.2025).
7. MongoDB – The developer data platform. URL: <https://www.mongodb.com/> (дата звернення 10.04.2025).
8. JWT – JSON Web Tokens for secure authentication. URL: <https://jwt.io/> (дата звернення 10.04.2025).
9. Bcrypt.js – Password hashing for Node.js. URL: <https://www.npmjs.com/package/bcrypt> (дата звернення 10.04.2025).
10. Cloudinary – Image and video upload, storage and transformation. URL: <https://cloudinary.com/> (дата звернення 10.04.2025).
11. Firebase Authentication. URL: <https://firebase.google.com/products/auth> (дата звернення 10.04.2025).
12. OWASP Top Ten Security Risks. URL: <https://owasp.org/www-project-top-ten/> (дата звернення 10.04.2025).
13. WHO guideline: digital health systems. URL: <https://www.who.int/publications> (дата звернення 10.04.2025).

14. Information security management systems. URL: <https://icr-cert.com.ua/iso-27001-2022-information-security-management-systems-isms/> (дата звернення 10.04.2025).
15. Standards for electronic health information. URL: <https://www.hl7.org/> (дата звернення 10.04.2025).
16. Vardhan, K., & Ghosh, S. (2021). Secure architecture for cloud-based healthcare systems. *Journal of Cloud Computing*, 10(1), 1–15.
17. Tripathi, A., & Mishra, A. (2020). Role of Full-Stack Development in Healthcare Web Applications. *International Journal of Computer Applications*, 176(39), 25–30.
18. HIPAA Compliance Guidelines for Developers. URL: <https://www.hipaajournal.com> (дата звернення 10.04.2025).
19. Ullah, A., & Kwak, D. (2022). Smart Healthcare Monitoring Framework using MERN Stack. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 13(7), 112–119.
20. Kaur, R., & Singh, M. (2021). E-Health Record Management System using MERN Stack. *Journal of Emerging Technologies and Innovative Research*, 8(12), 566–570.
21. Naderi, R., & Dehghan, A. (2020). Medical appointment scheduling using cloud computing. *Health Informatics Journal*, 26(1), 276–287.
22. Tailwind CSS – Utility-first CSS framework. URL: <https://tailwindcss.com/> (дата звернення 10.04.2025).
23. React Toastify – Notifications in React apps. URL: <https://fkhadra.github.io/react-toastify/> (дата звернення 10.04.2025).
24. Socket.IO – Real-time, bidirectional and event-based communication. URL: <https://socket.io/> (дата звернення 10.04.2025).
25. GitHub – Best practices for full stack health apps. URL: <https://github.com> (дата звернення 10.04.2025).

26. Малахова А.Р. (2025) Розробка вебзастосунку для електронного медичного запису. *Радіоелектроніка та молодь у XXI столітті: тези доповідей 28-го Міжнародного молодіжного форуму (Харків, 16–19 квітня 2025 р.)*. Харків: ХНУРЕ, 2025. Т. 7. С. 95–96.