

Міністерство освіти і науки України  
Кам'янець-Подільський національний університет імені Івана Огієнка  
Фізико-математичний факультет  
Кафедра комп'ютерних наук

**Кваліфікаційна робота бакалавра**  
з теми: «Розробка інформаційної системи підтримки web-ресурсу та мобільного додатку для керування процесом лікування»

Виконав: здобувач вищої освіти групи KNms1-B22  
спеціальності 122 Комп'ютерні науки

Сас Максим Олексійович

(прізвище, ім'я та по батькові здобувача вищої освіти)

Керівник: Федорчук Володимир Анатолійович,

доктор технічних наук, професор

(прізвище, ім'я та по батькові, науковий ступінь, вчене звання керівника)

Рецензент: Оптасюк Сергій Васильович,

кандидат фізико-математичних наук, доцент,

завідувач кафедри фізики

(прізвище, ім'я та по батькові, науковий ступінь, вчене звання рецензента)

## АНОТАЦІЯ

У дипломній роботі розглянуто процес розробки інформаційної системи підтримки web-ресурсу та мобільного додатку для керування процесом лікування. Актуальність дослідження зумовлена зростаючою потребою в цифровізації медичної сфери, зокрема, у створенні зручних, функціональних та адаптивних систем зберігання й обробки медичних даних.

Об'єктом дослідження є процес цифрової взаємодії між пацієнтами, лікарями та медичними установами. Предметом – програмна реалізація системи для управління медичною інформацією з урахуванням сучасних вимог до інтерфейсу, безпеки та масштабованості.

Метою дослідження є створення програмного засобу, який забезпечує централізований облік медичних записів, підтримку пацієнтів і лікарів, зручну навігацію, авторизацію користувачів, а також адаптивність до мобільних пристроїв.

У роботі проаналізовано існуючі програмні рішення, визначено їх недоліки та сформульовано вимоги до нової системи. Застосовано методи об'єктно-орієнтованого програмування, моделювання баз даних, архітектурного проектування, UX/UI-дизайну. Використано технології Angular, Node.js, MongoDB, Express, авторський фреймворк WAW.

Результатом роботи є функціональна інформаційна система Health, яка забезпечує зручну взаємодію з медичною інформацією, має сучасний інтерфейс, гнучку структуру та готова до впровадження в практичну діяльність приватних клінік або індивідуальних фахівців.

Робота може бути корисною для здобувачів вищої освіти спеціальності 122 «Комп'ютерні науки», а також для розробників програмного забезпечення у сфері медицини.

## ЗМІСТ

АНОТАЦІЯ.....	2
ВСТУП.....	4
РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ ІНФОРМАЦІЙНИХ СИСТЕМ У СФЕРІ ОХОРОНИ ЗДОРОВ'Я.....	6
1.1 Огляд існуючих інформаційних систем у медицині .....	6
1.2 Проблеми, пов'язані з використанням наявних рішень .....	7
1.3. Вимоги до сучасних медичних інформаційних систем.....	8
1.4. Вибір напрямку дослідження та постановка задачі .....	10
Висновки до розділу 1 .....	12
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ «Health».....	13
2.1. Вибір архітектури системи .....	13
2.2. Обґрунтування вибору технологій і середовищ розробки.....	14
2.3. Проєктування структури бази даних.....	15
2.4. Опис основних модулів системи .....	17
2.5. Інтерфейс користувача: принципи UX/UI-дизайну.....	21
Висновки до розділу 2 .....	25
РОЗДІЛ 3 ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ «Health».....	26
3.1. Методика тестування системи.....	26
3.2. Аналіз результатів тестування.....	27
3.3. Перспективи впровадження та подальшого розвитку.....	28
Висновки до розділу 3 .....	30
ВИСНОВКИ .....	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	33

## ВСТУП

У сучасних умовах інформатизації суспільства стрімко зростає потреба у впровадженні цифрових технологій у сферу охорони здоров'я. Зокрема, все більшої актуальності набувають мобільні додатки та web-системи, що дозволяють пацієнтам і лікарям оперативно взаємодіяти, зберігати, переглядати й оновлювати медичну інформацію. Проте існуючі рішення часто є вузькоспеціалізованими, мають обмежену функціональність або не забезпечують інтеграції з іншими медичними структурами. Це ускладнює процес лікування, координацію між учасниками медичного процесу та загалом знижує ефективність системи охорони здоров'я.

Актуальність теми зумовлена необхідністю створення гнучкої, функціональної та зручної в користуванні платформи, яка б дозволяла організовано зберігати медичні записи, ефективно взаємодіяти з лікарями, знаходити клініки та аптеки, а також адаптувалася до потреб як пацієнтів, так і медичного персоналу. Саме тому було запропоновано створити інформаційну систему підтримки web-ресурсу та мобільного додатку для керування процесом лікування – Health.

**Об'єктом дослідження** є процес цифровізації обміну медичною інформацією між учасниками системи охорони здоров'я.

**Предметом дослідження** виступає інформаційна система для web-ресурсу та мобільного додатку, що забезпечує інтерактивну взаємодію між лікарями, пацієнтами, медичними закладами й аптеками з можливістю ведення історії хвороб, пошуку установ, створення записів та зручного управління медичними даними.

**Метою дослідження** є розробка інтегрованої інформаційної системи - додатку Health, що дозволяє ефективно зберігати, обробляти та візуалізувати медичну інформацію, підтримує взаємодію між медичними закладами та користувачами, забезпечує швидкий доступ до даних, а також адаптивний та інтуїтивно зрозумілий інтерфейс.

Для досягнення цієї мети в роботі вирішуються наступні **завдання**:

1. Провести аналіз існуючих рішень у сфері медичних інформаційних систем.
2. Визначити функціональні й нефункціональні вимоги до розроблюваного додатку.
3. Спроекувати архітектуру системи, модулі й базу даних.
4. Реалізувати front-end і back-end частини додатку «з нуля».
5. Забезпечити зручний доступ до інформації про пацієнтів, лікарів, клініки, аптеки та медичні записи.
6. Реалізувати механізми пошуку, фільтрації, авторизації користувачів та адаптивного інтерфейсу.
7. Провести тестування функціоналу додатку та оцінити його ефективність.

**Методи дослідження:** для реалізації поставлених завдань використовувались методи об'єктно-орієнтованого програмування, проєктування інформаційних систем, моделювання баз даних, UX/UI-дизайну, а також аналітичні методи для вивчення потреб користувачів та порівняння з аналогами. Технічною основою стали: MongoDB, Node.js, Express, Angular [5], а також WAW-фреймворк.

**Практичне значення** одержаних результатів полягає у створенні універсального інструменту, який можна впроваджувати у діяльність медичних закладів для підвищення ефективності ведення пацієнтів, оптимізації доступу до медичних послуг і зменшення навантаження на персонал.

Структура роботи. Дипломна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі описано постановку задачі, аналіз аналогів та обґрунтування вибору технологій. У другому – спроектовано архітектуру системи та реалізовано її основні модулі. У третьому – проведено тестування та оцінено ефективність розробленого продукту.

## РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ ІНФОРМАЦІЙНИХ СИСТЕМ У СФЕРІ ОХОРОНИ ЗДОРОВ'Я

### 1.1 Огляд існуючих інформаційних систем у медицині

У сфері охорони здоров'я інформаційні системи відіграють ключову роль у забезпеченні якості медичного обслуговування, оптимізації процесів лікування та збереженні медичних даних. Протягом останніх десятиліть було створено низку рішень, які автоматизують різні аспекти медичної діяльності – від ведення електронної історії хвороб до аналітики та управління медичними установами.

Серед найвідоміших систем можна виділити:

- Medstar – система для обліку пацієнтів, медичних записів, управління персоналом і адміністративними процесами. Поширена у клініках середньої ланки.
- Mediware – платформа, орієнтована на лабораторні дослідження та фармацевтичне забезпечення. Забезпечує інтеграцію з електронними рецептами.
- DrChrono – хмарна система, що дозволяє створювати медичні записи, виставляти рахунки, вести календар прийомів. Має мобільну версію для лікарів.
- OpenMRS – відкрита медична система з можливістю кастомізації під різні країни та клініки. Широко використовується в країнах, що розвиваються.
- МКМІС (Україна) – державна система медичних записів, яка забезпечує базову електронну взаємодію між лікарями та пацієнтами через портал Helsi.

Попри наявність численних рішень, багато з них мають суттєві обмеження: відсутність зручного мобільного доступу, слабка інтеграція з

аптеками та іншими медичними закладами, застарілий або перевантажений інтерфейс, а також низька гнучкість під час масштабування.

Наведений аналіз свідчить про актуальність розробки нової системи, яка б поєднувала кращі практики попередників із сучасними вимогами до UI/UX, мобільності, безпеки й адаптивності.

## **1.2 Проблеми, пов'язані з використанням наявних рішень**

Попри наявність великої кількості розроблених інформаційних систем у галузі медицини, їхнє впровадження та ефективне використання стикається з низкою серйозних викликів. Нижче наведено основні проблеми, характерні для багатьох таких рішень:

### **1. Фрагментованість медичних даних.**

Часто дані про пацієнтів зберігаються в різних локальних базах або сервісах без можливості централізованого доступу. Це ускладнює комплексний підхід до діагностики та лікування, адже лікар не має повної інформації про історію хвороб пацієнта.

### **2. Відсутність мобільності та адаптивності.**

Багато існуючих систем не мають мобільних версій або адаптивних інтерфейсів, що значно знижує зручність роботи як для лікарів, так і для пацієнтів, особливо в умовах сучасної тенденції до використання смартфонів як основного пристрою доступу.

### **3. Незручність інтерфейсу та складність у користуванні.**

Інтерфейси багатьох систем є застарілими або перевантаженими функціями, що робить їх складними у використанні для користувачів без спеціальної ІТ-підготовки. Це призводить до помилок і додаткового часу на навчання персоналу.

### **4. Обмежені можливості пошуку та фільтрації.**

Часто відсутні або слабо реалізовані функції фільтрації пацієнтів, лікарів або записів за потрібними критеріями. Це ускладнює навігацію в системі при великому обсязі даних.

Враховання зазначених недоліків є надзвичайно важливим при проєктуванні нових медичних інформаційних систем. Власне, саме на подолання вищенаведених проблем і спрямована розробка системи Health, яка повинна забезпечити інтуїтивний інтерфейс, мобільність, зручні інструменти для управління медичною інформацією та гнучку інтеграцію з іншими сервісами.

### **1.3. Вимоги до сучасних медичних інформаційних систем**

З урахуванням виявлених проблем, сучасні медичні інформаційні системи повинні відповідати низці вимог, що забезпечують їх ефективність, безпеку, зручність та масштабованість. Ці вимоги можна умовно поділити на функціональні та нефункціональні.

#### **1. Функціональні вимоги:**

- **Управління даними пацієнтів.**

Система повинна надавати можливість створення, редагування, перегляду та видалення профілів пацієнтів. Інформація має бути структурованою (особисті дані, контактна інформація, історія хвороб, поточне лікування тощо).

- **Ведення медичних записів.**

Повинна підтримуватись реєстрація симптомів, діагнозів, аналізів, призначень і лікування з можливістю сортування та фільтрації за типом запису.

- **Пошук медичних установ та спеціалістів.**

Інформаційна система повинна надавати можливість пошуку клінік, лікарів і аптек за різними параметрами.

- Система авторизації користувачів.

Повинна підтримуватись багаторівнева авторизація (наприклад, лікар, адміністратор), з різними рівнями доступу до функцій та даних.

- Фільтрація та сортування інформації.

Користувачі повинні мати змогу фільтрувати пацієнтів за віком, статтю, категорією тощо.

- Інтеграція з зовнішніми ресурсами.

Система має підтримувати інтеграцію з картографічними сервісами (Google Maps), сторонніми медичними платформами (наприклад, [tabletki.ua](http://tabletki.ua)), державними реєстрами.

- Адаптивний інтерфейс.

Веб-інтерфейс повинен бути адаптований для використання на мобільних пристроях різних розмірів.

## 2. Нефункціональні вимоги:

- Безпека даних.

Система повинна відповідати вимогам конфіденційності та захисту персональних медичних даних. Необхідна реалізація таких механізмів як шифрування даних, захист від несанкціонованого доступу.

- Швидкодія та масштабованість.

Система має забезпечувати стабільну роботу при великій кількості одночасних користувачів і підтримувати масштабування без втрати продуктивності.

- Надійність і відмовостійкість.

Передбачено збереження даних у разі збою.

- Зручність та інтуїтивність інтерфейсу.

Інтерфейс має бути простим у користуванні як для лікарів, так і для пацієнтів, без потреби у спеціальному навчанні.

- Легкість у впровадженні та супроводі.

Важливо, щоб система могла бути розгорнута без значних витрат ресурсів, а її підтримка не вимагала залучення вузькоспеціалізованих ІТ-фахівців.

Задоволення цих вимог є критичним для створення дійсно ефективною та зручною системою, яка зможе покращити взаємодію між усіма учасниками процесу лікування. У розробці додатку Health ці аспекти враховано на етапі постановки задач і реалізації функціоналу.

#### **1.4. Вибір напрямку дослідження та постановка задачі**

На основі проведеного аналізу сучасного стану медичних інформаційних систем, а також виявлених недоліків і актуальних вимог, обґрунтовано доцільність розробки універсального web-ресурсу та мобільного додатку для підтримки процесу лікування. Враховуючи зростаючі вимоги до доступності, зручності й швидкодії систем охорони здоров'я, вибір напрямку дослідження зумовлено прагненням створити платформу, яка б відповідала як функціональним, так і нефункціональним вимогам, розглянутим у попередньому підпункті [3].

Запропоноване рішення – інформаційна система Health, яка має на меті поєднати зручний інтерфейс для пацієнтів і лікарів з гнучкою модульною архітектурою, яка дозволяє ефективно управляти медичною інформацією, фільтрувати, сортувати й оновлювати записи в режимі реального часу, шукати лікарів, клініки, аптеки, а також інтегрувати зовнішні сервіси (наприклад, картографічні або аптечні системи).

Основною проблемою, яку вирішує дана система, є відсутність вітчизняних інструментів, що одночасно забезпечують:

- мобільність доступу до медичних даних;
- гнучку систему керування користувачами;
- простоту візуального представлення великої кількості даних;

- адаптацію під потреби різних типів користувачів (лікарів, пацієнтів, адміністраторів);
- підтримку інтеграції з іншими сервісами без необхідності ручного дублювання даних.

Таким чином, у межах дослідження було поставлено наступні задачі:

1. Проаналізувати існуючі інформаційні системи у сфері охорони здоров'я та виявити їх переваги й недоліки.
2. Визначити функціональні та нефункціональні вимоги до нової системи.
3. Обґрунтувати вибір інструментів і технологій для реалізації програмного забезпечення.
4. Розробити архітектуру системи, базу даних та програмні модулі.
5. Реалізувати інтерфейс користувача з урахуванням сучасних принципів UX/UI-дизайну.
6. Реалізувати функціонал для управління пацієнтами, лікарями, клініками, аптеками та місцями.
7. Забезпечити мобільну адаптацію додатку та підтримку світлої/темної тем.
8. Провести тестування системи та оцінити її відповідність поставленим вимогам.

Таким чином, дослідження спрямоване не лише на створення нової інформаційної системи, але й на обґрунтування ефективності її використання у практиці медичних закладів.

## Висновки до розділу 1

У першому розділі було проведено аналіз сучасного стану інформаційних систем у сфері охорони здоров'я, розглянуто відомі приклади таких систем та виявлено їх основні недоліки. Показано, що більшість існуючих рішень є недостатньо гнучкими, не забезпечують інтеграції з зовнішніми сервісами, мають застарілі інтерфейси або не підтримують мобільний доступ. Водночас сучасні умови вимагають створення медичних інформаційних систем, що відповідають принципам мобільності, зручності, безпеки й масштабованості.

На основі виявлених проблем сформульовано перелік функціональних та нефункціональних вимог до сучасної медичної системи. Було обґрунтовано доцільність створення інтегрованого додатку Health як комплексного інструменту для зберігання, обробки та доступу до медичних даних, з можливістю взаємодії між лікарями, пацієнтами та медичними закладами.

Поставлені задачі дослідження охоплюють як проєктування архітектури, так і реалізацію та тестування функціональних модулів системи. Таким чином, вибір напрямку дослідження є обґрунтованим і актуальним для розвитку цифрових рішень у сфері охорони здоров'я.

## РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ «HEALTH»

### 2.1. Вибір архітектури системи

У процесі розробки інформаційної системи Health було прийнято рішення використовувати трирівневу архітектуру, яка передбачає розділення на такі рівні:

1. Клієнтський рівень (Front-end) – відповідає за взаємодію з користувачем через браузер або мобільний пристрій. Реалізований з використанням фреймворку Angular, що дозволяє створювати адаптивний, динамічний і зручний інтерфейс користувача.

2. Серверний рівень (Back-end) – обробляє логіку додатку, взаємодіє з базою даних, обробляє запити від клієнта та відправляє відповіді. Для реалізації використано Node.js у поєднанні з фреймворком Express.js [7].

3. Рівень зберігання даних (Database) – відповідає за зберігання структурованої інформації про пацієнтів, медичні записи, клініки, лікарів, аптеки тощо. У якості СУБД обрано MongoDB, що добре підходить для роботи з JSON-подібними документами та забезпечує високу продуктивність при масштабуванні.

Для внутрішнього управління компонентами системи використовується фреймворк WAW, який спрощує структуру проєкту, забезпечує централізовану маршрутизацію, підтримку модулів і розширень, а також зменшує залежність від зовнішніх рішень.

Переваги обраної архітектури:

- Висока масштабованість;
- Розділення відповідальностей між компонентами;
- Гнучкість при розширенні функціональності;
- Зручність у супроводі й тестуванні;
- Адаптивність до мобільних платформ.

Таким чином, трирівнева архітектура у поєднанні з сучасними інструментами дозволяє створити гнучку, розширювану та надійну систему для автоматизації медичних процесів.

## 2.2. Обґрунтування вибору технологій і середовищ розробки

Для реалізації інформаційної системи Health було обрано сучасні інструменти та середовища, що забезпечують високу продуктивність, масштабованість, зручність у розробці й підтримці.

### 1. Середовище розробки.

- Visual Studio Code – легке, швидке та функціональне середовище, що підтримує велику кількість розширень, автодоповнення, підсвічування синтаксису та інтеграцію з системами контролю версій.

### 2. Технології клієнтської частини.

- Angular – фронтенд-фреймворк, що дозволяє створювати масштабовані односторінкові додатки з чіткою структурою. Підтримує компонентний підхід, двосторонню прив'язку даних, ін'єкцію залежностей та адаптивний дизайн.

- SCSS – препроцесор для стилів, що дозволяє організувати CSS-код у більш зручному та підтримуваному вигляді.

### 3. Технології серверної частини.

- Node.js – середовище виконання JavaScript, яке дозволяє будувати асинхронні, масштабовані сервери. Забезпечує високу продуктивність за рахунок неблокуючої архітектури.

- Express.js – легкий фреймворк для Node.js, який спрощує побудову REST API, маршрутизацію, обробку запитів і підключення до бази даних.

### 4. База даних.

- MongoDB – документоорієнтована база даних, яка дозволяє зберігати дані у форматі JSON. Забезпечує гнучкість структури, масштабованість та зручну роботу з вкладеними даними.

#### 5. Авторський фреймворк.

- WAW framework – фреймворк, який був спеціально створений для прискорення розробки додатків. Забезпечує стандартизацію структури проєкту, підтримку тем, адаптивного дизайну, навігації та управління станом.

#### 6. Зовнішні сервіси та інтеграції.

- Google Maps – для візуалізації місць на мапі.
- Tabletki.ua – для створення посилань на препарати, доступні в аптеках.
- Шрифти та іконки – використано Google Fonts, Material Icons, Lucide.

### 2.3. Проєктування структури бази даних

Однією з ключових частин будь-якої інформаційної системи є її база даних. Для реалізації гнучкої, масштабованої та швидкої взаємодії з медичною інформацією у проєкті Health було обрано документоорієнтовану базу даних MongoDB [9].

Використання MongoDB дає змогу:

- зберігати складні, вкладені структури даних (наприклад, історії записів пацієнта);
- гнучко змінювати схему без необхідності міграцій;
- ефективно працювати з великими обсягами записів.

Основні колекції бази даних:

#### 1. Users.

Зберігає інформацію про користувачів системи: логін, пароль (у зашифрованому вигляді), та інші поля.

## 2. Patients.

Містить дані про пацієнтів: ім'я, стать, дата народження, категорія (дитина, дорослий, літній), лікар, медичні записи, контактна інформація тощо.

## 3. Records.

Медичні записи, пов'язані з пацієнтами. Запис містить тип (симптом, діагноз, аналіз, лікування), дату створення, опис.

## 4. Clinics.

Інформація про клініки: назва, адреса, контактні дані, спеціалізація, опис послуг, години роботи.

## 5. Doctors.

Профілі лікарів: ім'я, спеціалізація, досвід, вартість прийому, контактна інформація.

## 6. Pharmacies.

Відомості про аптеки: назва, адреса, робочі години, контактні дані.

## 7. Places.

Загальні місця, прив'язані до геолокації: назва, тип (клініка, аптека), координати для Google Maps, короткий опис.

## 8. Drugs.

Ліки: назва, опис, список аптек, де вони доступні, посилання на [tabletki.ua](http://tabletki.ua) (див. рис. 2.1).

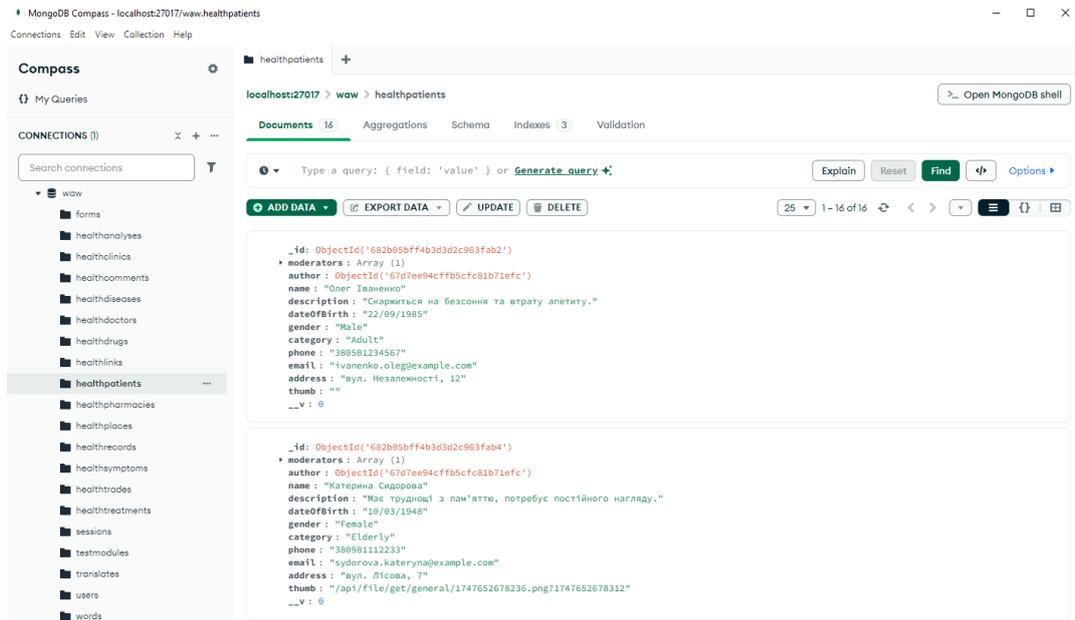


Рисунок 2.1 – База пацієнтів в MongoDB

Усі зв'язки між документами реалізуються за допомогою збереження ідентифікаторів (ObjectId), що дозволяє зберігати відносну нормалізацію при високій гнучкості.

Такий підхід дозволяє системі працювати швидко, а також ефективно обробляти запити навіть при великій кількості користувачів.

## 2.4. Опис основних модулів системи

Інформаційна система Health реалізована як набір модулів, кожен з яких відповідає за окремий аспект функціональності системи. Такий підхід забезпечує логічну організацію коду, спрощує підтримку і розширення додатку.

### 1. Модуль healthpatient.

Забезпечує управління інформацією про пацієнтів:

- створення та редагування профілю;
- фільтрація за статтю та категорією (дитина, дорослий, літній);
- пошук за іменем;

- перегляд детальної інформації;
- зв'язок із медичними записами (див. рис. 2.2).

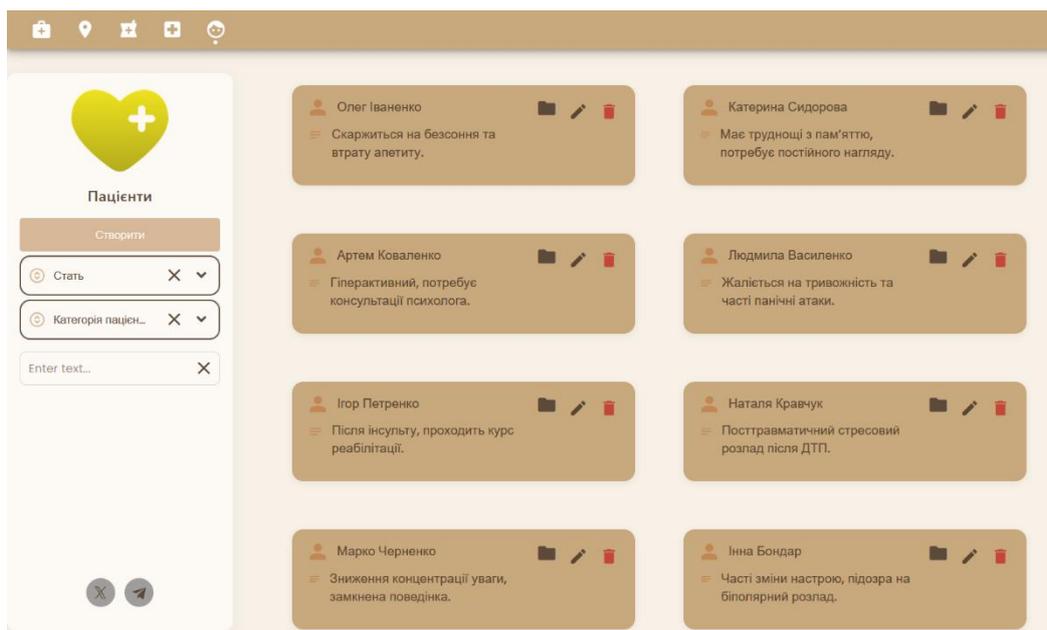


Рисунок 2.2 – Інтерфейс модуля 'Пацієнти' з фільтрами та пошуком

## 2. Модуль healthrecord.

Служить для створення, перегляду, редагування та видалення медичних записів:

- типи записів: симптом, діагноз, аналіз, лікування;
- сортування за типом і датою;
- зберігання хронології історії хвороби;
- можливість перегляду на сторінці пацієнта (див. рис. 2.3).

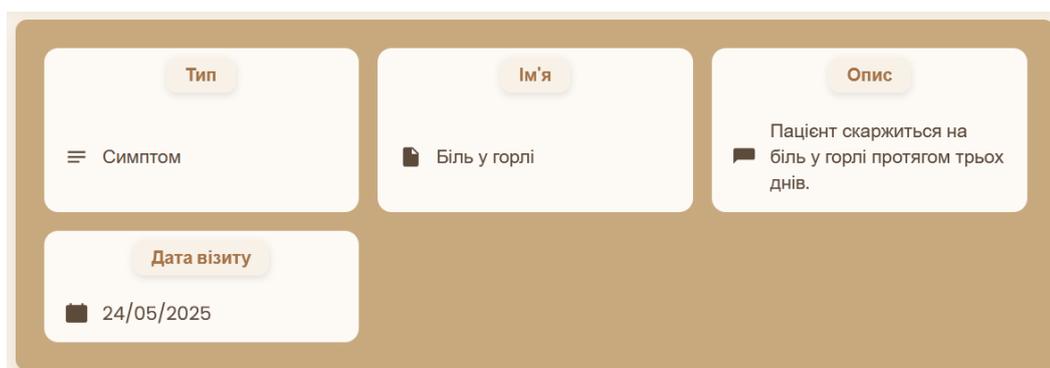


Рисунок 2.3 – Медичний запис пацієнта у вигляді картки

### 3. Модуль healthclinic.

Містить інформацію про клініки:

- створення та редагування профілю клініки;
- опис, контакти, розклад роботи, спеціалізації;
- пошук і фільтрація;
- візуалізація у вигляді банера (див. рис. 2.4).

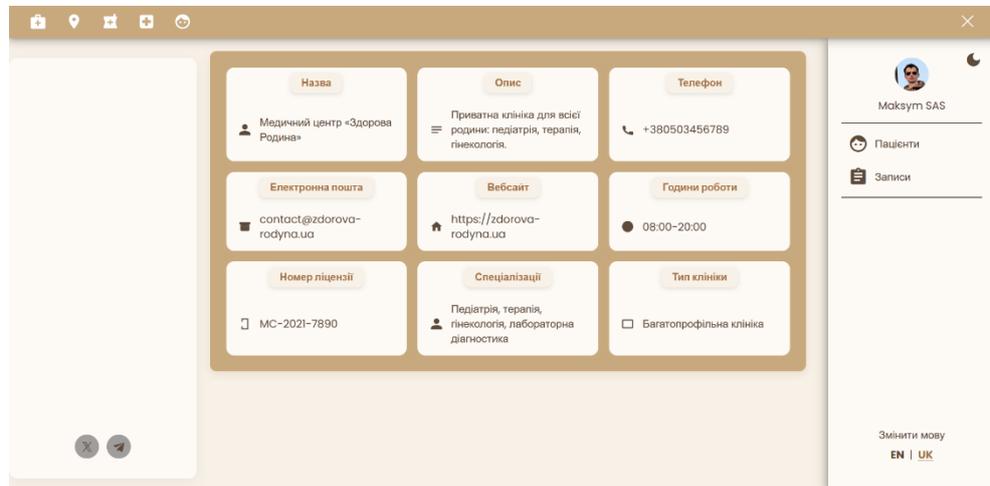


Рисунок 2.4 – Сторінка клініки з інформаційним банером

### 4. Модуль healthdoctor.

Відповідає за дані про лікарів:

- створення профілю лікаря;
- пошук і фільтрація за клінікою;
- опис спеціалізації, вартість прийому, досвід;
- редагування та видалення (див. рис. 2.5).

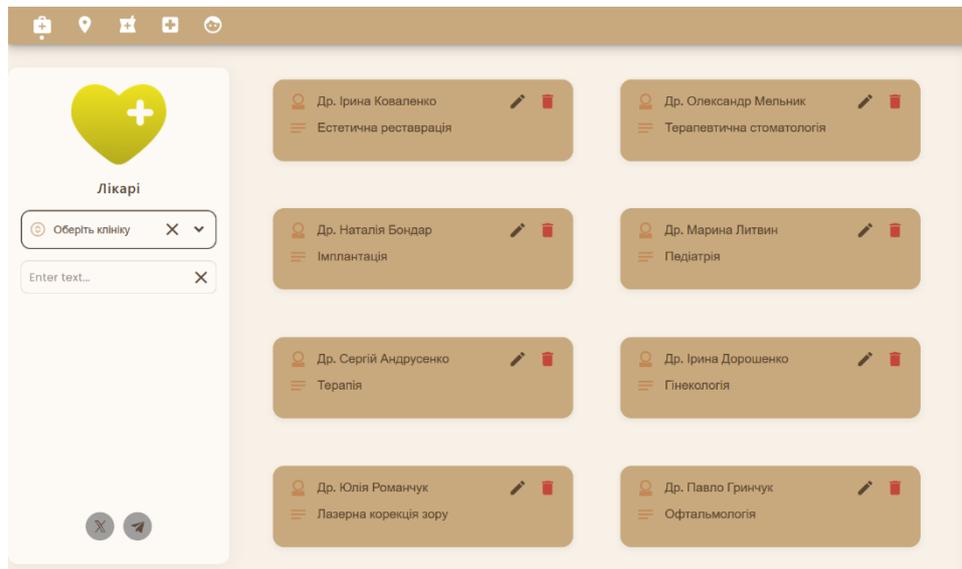


Рисунок 2.5 – Список лікарів з фільтром за клініками

## 5. Модуль healthpharmacy.

Призначений для управління аптеками:

- зберігання інформації про робочий час, контакти, вебсайт;
- інтеграція з [tabletki.ua](http://tabletki.ua) для перевірки наявності ліків [11];
- фільтрація за препаратами (див. рис. 2.6).

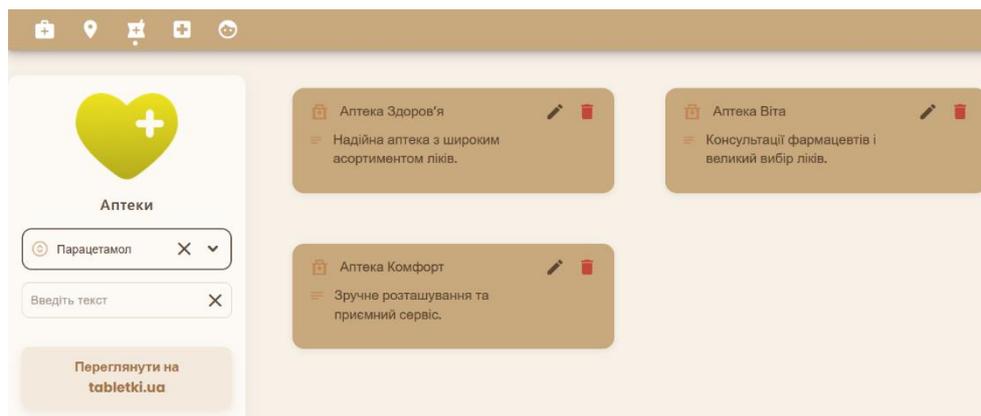


Рисунок 2.6 – Список аптек з можливістю перевірки ліків

## 6. Модуль healthplace.

Модуль для загального управління медичними локаціями:

- зберігання місць (клініки, аптеки);
- фільтрація за типом;
- посилання на Google Maps [8];

- переходи до сторінки локації (див. рис. 2.7).



Рисунок 2.7 – Інформація про місце з інтеграцією Google Maps

## 7. Адміністративний модуль.

- надання адміністративних прав;
- підтримка інтерфейсу для адміністраторів.

## 8. Система авторизації.

- окремий інтерфейс логіну;
- захищена перевірка користувача;
- поділ доступу за ролями.

Ця модульна структура забезпечує не лише простоту в навігації для користувача, але й зручність у розробці: кожен модуль має власний маршрут, шаблони, стилі та логіку, що дозволяє масштабувати систему в майбутньому.

## 2.5. Інтерфейс користувача: принципи UX/UI-дизайну

Інтерфейс користувача є одним з ключових чинників ефективності інформаційної системи. У проєкті Health особлива увага приділялась саме дизайну взаємодії (UX) та візуальному оформленню (UI), щоб забезпечити зручність, інтуїтивність і привабливість платформи для різних категорій користувачів: лікарів, адміністраторів та пацієнтів.

Основні принципи UX/UI, реалізовані у системі:

1. Мінімалізм і чистота інтерфейсу.

Інтерфейс розроблено таким чином, щоб користувач одразу бачив лише найнеобхідніші елементи. Додаткові функції доступні через випадаючі меню, вкладки або контекстні кнопки.

## 2. Адаптивний дизайн.

Вся система адаптована до роботи на різних пристроях: десктопах, планшетах та смартфонах. Адаптація реалізована з використанням Angular Flex Layout та SCSS медіа-запитів.

## 3. Підтримка темної та світлої тем.

Для зменшення навантаження на зір користувача додаток підтримує перемикання між світлою та темною темами (див. рис. 2.8).

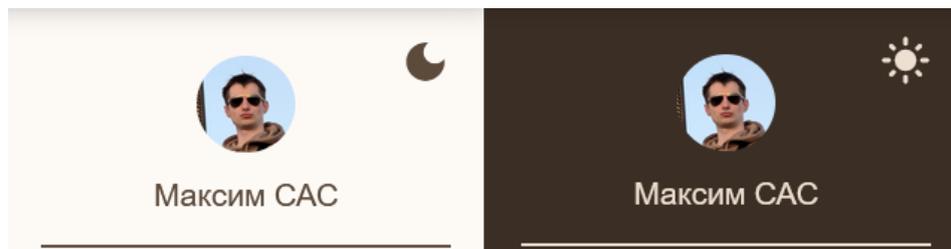


Рисунок 2.8 – Перемикання між світлою та темною темою інтерфейсу

## 4. Логічна ієрархія сторінок.

Навігація організована за допомогою бічного меню (sidebar), у якому згруповано всі основні розділи: пацієнти, записи, клініки, лікарі, аптеки, місця, адміністрування.

## 5. Фільтри і пошук.

У модулях пацієнтів, лікарів, клінік, аптек реалізовані інструменти для швидкого пошуку та фільтрації даних за ключовими параметрами (наприклад, стать, категорія, спеціалізація, наявність ліків тощо) (див. рис. 2.9).

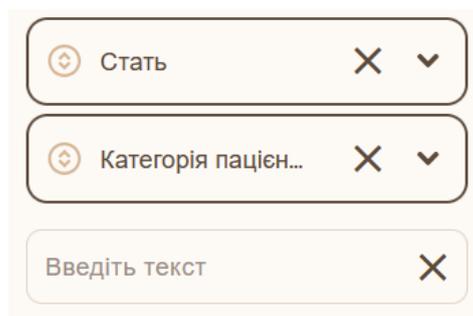


Рисунок 2.9 – Інтерфейс пошуку та фільтрації у списку пацієнтів

## 6. Систематизовані картки.

Усі об'єкти відображаються у вигляді уніфікованих карток: пацієнт, запис, лікар, клініка, аптека. Це дозволяє швидко зорієнтуватися в даних та полегшує візуальне сприйняття інформації (див. рис. 2.10).

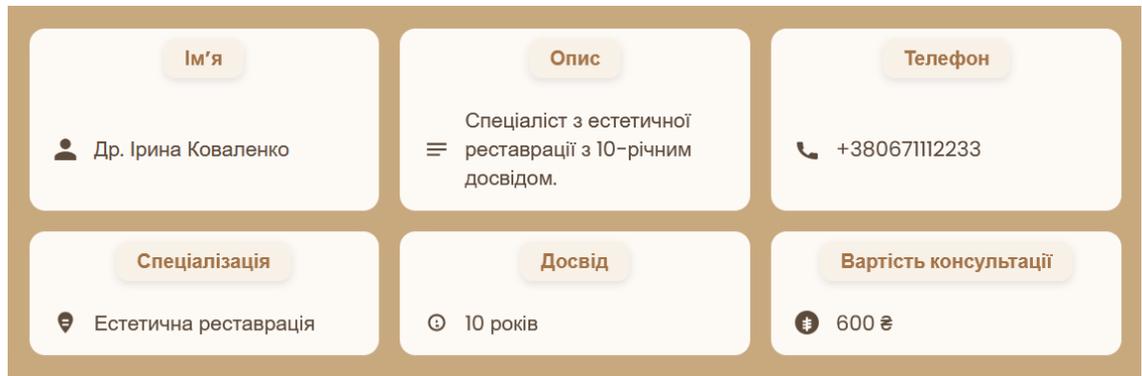


Рисунок 2.10 – Приклад картки лікаря з короткою інформацією

## 7. Анімації та візуальні підказки.

Застосовано плавні анімації при наведенні, переходах між сторінками, відкритті форм. Це покращує загальний користувацький досвід.

## 8. Повідомлення та підтвердження дій.

Система містить спливаючі повідомлення про успішні дії або помилки. Для критичних дій, як-от видалення – використовуються діалоги підтвердження.

Інструменти UX/UI:

- Angular Material – набір компонентів, що забезпечує зручні шаблони та дотримання гайдлайнів Material Design;
- Tailwind (частково) – для кастомізації вигляду [12];
- Іконки SVG та Material Icons – для покращення навігації.

## 9. Мобільна версія.

Створено повноцінний мобільний, адаптивний додаток (див. рис. 2.11).

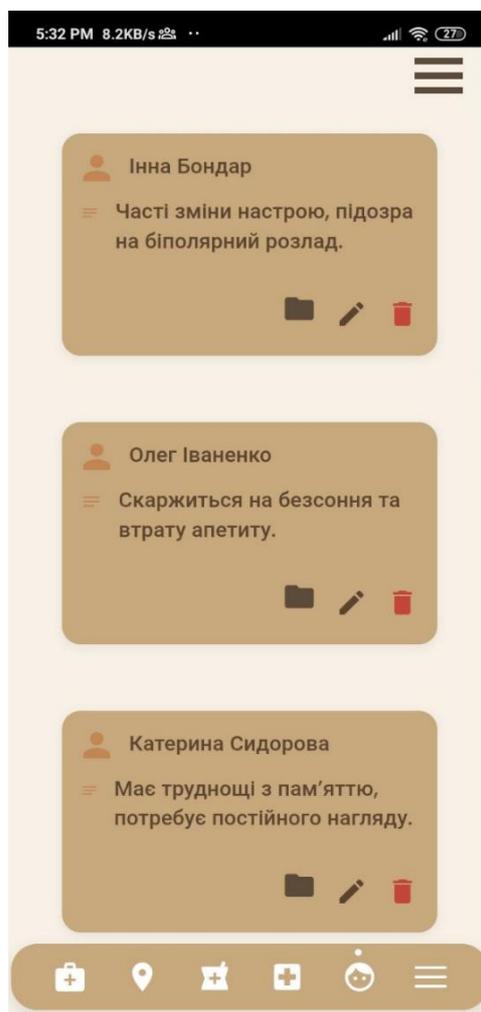


Рисунок 2.11 – Приклад вікна мобільного додатку

Таким чином, інтерфейс системи Health створений з урахуванням сучасних вимог до медичних платформ: він простий, адаптивний, зручний для користувача та легко масштабується з додаванням нових функцій.

## Висновки до розділу 2

У другому розділі було детально розглянуто проєктування та реалізацію інформаційної системи Health. Обґрунтовано вибір трирівневої архітектури, що забезпечує розділення логіки, інтерфейсу користувача та зберігання даних, дозволяючи системі бути масштабованою, гнучкою та зручною для супроводу.

Для реалізації обрано сучасні технології: Angular, Node.js, Express.js, MongoDB, а також авторський фреймворк WAW, який забезпечив уніфіковану структуру проєкту. Було створено ефективну, документоорієнтовану модель бази даних, яка охоплює всі необхідні сутності: користувачів, пацієнтів, записи, клініки, лікарів, аптеки, місця та ліки.

У розділі докладно описано функціональність основних модулів системи, їхній взаємозв'язок і призначення. Окрема увага приділена розробці інтерфейсу користувача з урахуванням сучасних принципів UX/UI: адаптивність, підтримка тем, зручна навігація, швидкий пошук і фільтрація.

Таким чином, реалізована система задовольняє визначені вимоги до сучасних медичних інформаційних платформ і створює надійний фундамент для подальшого розвитку та впровадження в медичну практику.

## РОЗДІЛ 3 ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ «HEALTH»

### 3.1. Методика тестування системи

Для перевірки працездатності та відповідності розробленої системи Health поставленим вимогам було проведено комплексне тестування, яке охоплювало:

#### 1. Функціональне тестування.

Метою було переконатися, що кожна функція додатку працює відповідно до специфікації. Тестування охоплювало:

- створення, редагування, видалення та перегляд пацієнтів;
- роботу з медичними записами різних типів;
- пошук та фільтрацію в модулях пацієнтів, лікарів, клінік, аптек;
- створення та редагування об'єктів: клінік, лікарів, аптек, місць;
- авторизацію користувачів з різними рівнями доступу.

#### 2. UX/UI тестування.

Було перевірено зручність взаємодії з інтерфейсом:

- адаптивність верстки (десктоп, планшет, мобільні екрани);
- коректне відображення елементів у світлій і темній темах;
- зручність пошуку та навігації.

#### 3. Тестування безпеки.

Перевірялися:

- надійність авторизації (використано шифрування паролів);
- захист від некоректних запитів на стороні сервера.

### 3.2. Аналіз результатів тестування

У результаті проведеного комплексного тестування системи Health було підтверджено, що реалізовані функціональні можливості відповідають поставленим вимогам, а інтерфейс забезпечує зручну та ефективну взаємодію користувача з системою.

#### 1. Функціональні результати.

Під час тестування ключових модулів системи встановлено, що:

- Додавання, редагування, перегляд та видалення пацієнтів і записів здійснюється без збоїв;
- Пошук і фільтрація працюють коректно як у браузері, так і в мобільній версії;
- Зв'язки між сутностями працюють відповідно до проєктної логіки;
- Додавання записів різних типів працює стабільно.

#### 2. UI/UX результат.

- Інтерфейс коректно масштабується на різних розширеннях екрана;
- Темна та світла теми перемикаються миттєво без перезавантаження сторінки;
- Усі інтерактивні елементи мають відповідні стилі наведення, фокусу та активності;
- Повідомлення про помилки та успішні дії з'являються вчасно та не заважають основному контенту.

#### 3. Продуктивність і стабільність.

- Час відповіді API при роботі з великими наборами даних становив у середньому менше 200 мс;
- Після 1 години активної роботи не було виявлено витоків пам'яті;
- Система стабільно працює при 10+ одночасних користувачах.

#### 4. Безпека.

- Паролі зберігаються у хешованому вигляді (bcrypt);
- АРІ надійно обмежує доступ до даних неавторизованим користувачам (див. рис. 3.1) [6].

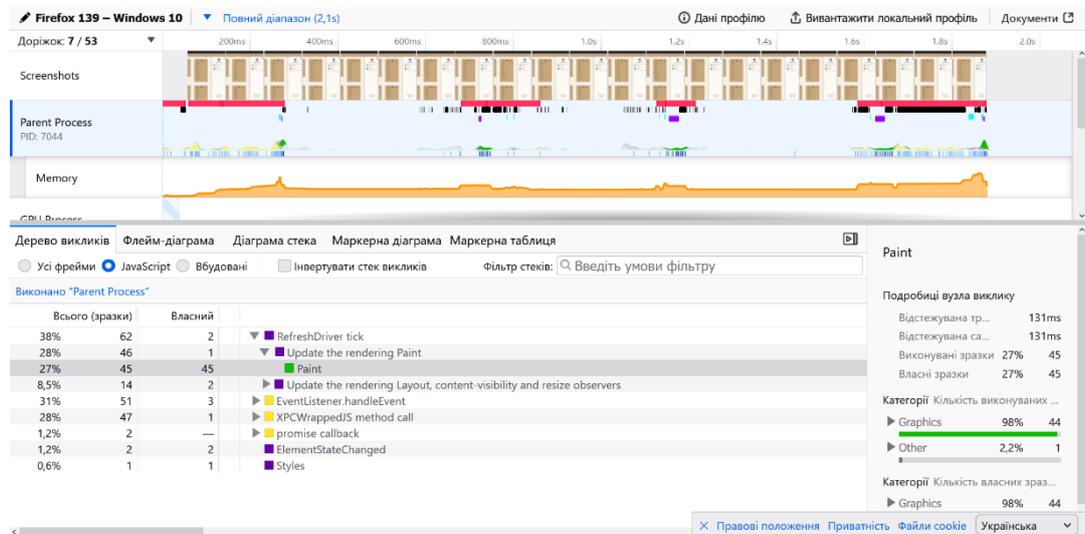


Рисунок 3.1 – Звіт про продуктивність системи при створенні даних

Система Health продемонструвала високу стабільність, відповідність вимогам і продуктивність у рамках усіх перевірених сценаріїв. Інтерфейс не потребує спеціального навчання для користувачів, що підтверджує інтуїтивність дизайну. Отримані результати свідчать про готовність додатку до практичного впровадження у реальне середовище.

### 3.3. Перспективи впровадження та подальшого розвитку

Результати тестування інформаційної системи Health свідчать про її практичну придатність до впровадження у сферу охорони здоров'я. Наявна реалізація покриває ключові процеси взаємодії між лікарем і пацієнтом, а також забезпечує ефективне управління медичними записами, установами та персоналом.

Перспективи впровадження:

1. Використання в приватних клініках.

Завдяки модульній структурі, Health може бути впроваджена у малих та середніх приватних клініках для цифрового обліку пацієнтів, автоматизації записів і внутрішньої координації персоналу.

2. Використання в лікарів-фахівців або сімейних лікарів.

Інтерфейс системи адаптований під індивідуальну практику - лікар може вести власну базу пацієнтів та їх медичних історій без зайвого функціонального навантаження.

Напрями подальшого розвитку:

1. Реєстрація пацієнтів з особистим кабінетом.

Додавання ролі "пацієнт" з можливістю самостійного перегляду історії лікування, онлайн-запису на прийом та доступу до аналізів.

2. Додавання аналітики та звітності.

Побудова дашбордів для аналізу кількості записів, типів звернень, найпопулярніших спеціалізацій тощо.

3. Локалізація та мультимовна підтримка.

Забезпечення підтримки англійської, української та інших мов для поширення додатку серед іноземних користувачів.

4. Хмарне розгортання.

Перетворення додатку на хмарний сервіс з можливістю підключення лікарів і клінік без необхідності індивідуального хостингу.

Таким чином, платформа Health має значний потенціал для масштабування, персоналізації та інтеграції з сучасною інфраструктурою охорони здоров'я.

### Висновки до розділу 3

У третьому розділі було розглянуто процес тестування, аналіз результатів та перспективи впровадження розробленої інформаційної системи Health. Проведене функціональне, інтерфейсне, продуктивне та безпекове тестування підтвердило відповідність системи вимогам, які були поставлені на етапі проєктування. Усі ключові модулі функціонують коректно, інтерфейс зручний та адаптований під різні пристрої, а час відповіді при роботі з великими обсягами даних залишається на стабільно високому рівні.

Система демонструє повну готовність до впровадження у практичну діяльність приватних клінік, лікарських кабінетів або як навчального прикладу для вищих навчальних закладів.

Окреслено основні напрями подальшого розвитку платформи - зокрема, розширення ролей користувачів, інтеграцію з державними реєстрами, додавання аналітики та впровадження хмарної моделі використання.

Таким чином, Health є перспективним рішенням для автоматизації процесів у сфері охорони здоров'я та має потенціал для масштабування і комерційного розповсюдження.

## ВИСНОВКИ

У результаті виконання дипломної роботи на тему «Розробка інформаційної системи підтримки web-ресурсу та мобільного додатку для керування процесом лікування» було досягнуто основної мети - створено повнофункціональну інформаційну систему, яка дозволяє ефективно керувати медичною інформацією та забезпечує зручну взаємодію між лікарями, пацієнтами, клініками та іншими медичними установами.

У ході дослідження було проведено огляд існуючих інформаційних систем у сфері охорони здоров'я, на основі чого виявлено їх основні недоліки: низьку гнучкість, складність у використанні, відсутність мобільних версій та обмежену інтеграцію з іншими сервісами. Це дало змогу сформулювати актуальні функціональні та нефункціональні вимоги до нового програмного продукту. Було обґрунтовано вибір трирівневої архітектури, а також інструментів розробки, серед яких – Angular, Node.js, MongoDB та фреймворк WAW.

Розроблена система реалізує низку ключових функцій: управління пацієнтами та їх записами, пошук лікарів, клінік та аптек, зберігання структурованої медичної інформації, інтеграцію з картографічними сервісами, адаптивний інтерфейс з підтримкою темної та світлої тем, систему авторизації та адміністрування. Проведене функціональне та продуктивне тестування підтвердило стабільність роботи системи, її відповідність поставленим вимогам, зручність інтерфейсу та надійність при взаємодії з великими обсягами даних.

Разом з тим, у ході роботи було виявлено і певні обмеження реалізованої версії. Зокрема, система не включає можливості створення особистих кабінетів пацієнтів, не має аналітичного модуля для побудови статистики, поки що не передбачає прямої інтеграції з державними реєстрами, а

функціональність керування правами користувачів потребує подальшого вдосконалення.

З урахуванням цього, у перспективі доцільним є розширення функціоналу системи, включення ролі пацієнта, впровадження аналітики, додавання мультимовної підтримки, а також адаптація продукту під модель SaaS для комерційного використання.

У підсумку, всі поставлені завдання в межах дипломної роботи виконано, а сформульована мета – досягнута. Результати дослідження мають не лише теоретичну, але й прикладну цінність і можуть бути використані в реальних умовах медичної практики або слугувати основою для подальших досліджень та розробок у цій галузі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008:2015 «Звіти у сфері науки і техніки. Структура та правила оформлення».
2. ДСТУ 8302:2015 Інформація та документація. Бібліографічне посилання.
3. Сас М. О. Розробка інформаційної системи підтримки web-ресурсу та мобільного додатку для керування процесом лікування. *Збірник матеріалів наукової конференції за підсумками науково-дослідної роботи здобувачів вищої освіти фізико-математичного факультету у 2024-2025 н. р.* 2025. С. 69-71.
4. Angular Material Design Components. URL: <https://material.angular.io/> (Дата звернення: 01.03.2025).
5. Angular Official Documentation. URL: <https://angular.io/docs> (Дата звернення: 01.03.2025).
6. Firefox DevTools Overview. URL: <https://firefox-source-docs.mozilla.org/devtools-user/> (Дата звернення: 01.03.2025).
7. Express.js Guide - Fast, unopinionated, minimalist web framework for Node.js. URL: <https://expressjs.com/>(<https://expressjs.com/>) (Дата звернення: 06.03.2025).
8. Google Maps. URL: <https://www.google.com/maps> (Дата звернення: 16.03.2025).
9. MongoDB Documentation - The Complete Guide to NoSQL Document Databases. URL: <https://www.mongodb.com/docs> (Дата звернення: 25.04.2025).
10. Node.js Documentation. URL: <https://nodejs.org/en/docs/> (Дата звернення: 25.04.2025).
11. Tabletki.ua - сервіс пошуку лікарських засобів. URL: <https://tabletki.ua/> (Дата звернення: 01.05.2025).

12. Tailwind CSS Documentation. URL: <https://tailwindcss.com/docs> (Дата звернення: 10.05.2025).
13. WebArtWork. URL: <https://github.com/WebArtWork> (Дата звернення: 16.05.2025).