

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА

### Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Віртуальна система управління медичним закладом  
(тема)

Виконав:

Студент II курсу, групи СКСМ-22-2  
Холоша С. В.  
(прізвище, ініціали)

Спеціальність 123 – Комп'ютерна інженерія  
(код і повна назва спеціальності)

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані  
комп'ютерні системи  
(повна назва освітньої програми)

Керівник: доц. Ларченко Л.В.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри АПОТ



(підпис)

Чумаченко С.В.


(прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
Кафедра Автоматизації проектування обчислювальної техніки  
Рівень вищої освіти другий (магістерський)  
Спеціальність 123 – Комп'ютерна інженерія  
(код і повна назва)  
Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)  
Освітня програма Спеціалізовані комп'ютерні системи  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри   
(підпис)

“ ” \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ  
НА АТЕСТАЦІЙНУ РОБОТУ**

студентові Холоші Сергію Вікторовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Віртуальна система управління медичним закладом

затверджена наказом по університету від " 03 " 11 2023 р. № 1282 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 26.01.2024

3. Вихідні дані до роботи

Віртуальна система управління

Мова програмування PHP, MySQL

4. Перелік питань, що потрібно опрацювати в роботі

1 Аналіз предметної області та постановка задачі

2 Проектування архітектури системи

3 Розробка особистого кабінету

4 Тестування та оцінка ефективності

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 19 слайдів


6. Консультанти розділів роботи (заповнюється 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

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	01.09.2023-06.09.2023	
2	Аналіз предметної області	06.09.2023-30.09.2023	
3	Аналіз джерел з проблемної галузі	30.09.2023-08.10.2023	
4	Проектування архітектури системи	08.10.2023-12.10.2023	
5	Вибір технологій для розробки	12.10.2023-20.10.2023	
6	Написання програмної реалізації розроблених функціональних можливостей	20.10.2023-20.11.2023	
7	Проведення тестування та оцінки ефективності	20.11.2023-30.11.2023	
8	Оформлення пояснювальної записки	30.11.2023-18.12.2023	
9	Оформлення графічного матеріалу	19.12.2023-24.12.2023	
10	Перевірка виконаного проекту керівником	25.12.2023-10.01.2024	
11	Захист кваліфікаційної роботи	26.01.2024	

Дата видачі завдання 01.09.2023

Студент   
(підпис)

Керівник роботи   
(підпис)

доц. Ларченко Л.В.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка містить 61 сторінок, 20 рисунків, 11 джерел за переліком посилань.

ВІРТУАЛЬНА СИСТЕМА, КАБІНЕТ ПАЦІЄНТА, ЕЛЕКТРОННИЙ РЕЦЕПТ, ТЕЛЕМЕДИЦИНА, WEB, LARAVEL, HL7, XML, PHP, MYSQL

В кваліфікаційній роботі досліджено та розроблено віртуальну систему управління медичним закладом з використанням сучасних тенденцій та стандартів.

Створена електронна медична система дозволяє зручно та оперативно отримувати доступ до особистої медичної інформації пацієнтів, особливо в ситуаціях, коли потрібне термінове лікування. В рамках системи реалізовано можливість онлайн реєстрації та консультацій, що дозволяє пацієнтам легко реєструватися та отримувати консультації від медичних фахівців через Інтернет. Це робить доступ до медичної допомоги більш гнучким та ефективним у невідкладних випадках.

Розроблена медична інформаційна система стала ефективним інструментом для забезпечення доступу, обміну та управління медичною інформацією, враховуючи сучасні вимоги до збереження та обробки медичних даних.

## ABSTRACT

The explanatory note consists of 61 pages, 20 figures, and 11 references according to the bibliography.

VIRTUAL SYSTEM, PATIENT CABINET, ELECTRONIC PRESCRIPTION, TELEMEDICINE, WEB, LARAVEL, HL7, XML, PHP, MYSQL

In the qualification work, a virtual management system for a medical institution has been researched and developed using modern trends and standards. The created electronic system allows convenient and prompt access to patients' personal medical information, especially in situations requiring urgent treatment. The system incorporates online registration and consultations, enabling patients to easily register and receive consultations from medical professionals over the Internet. This makes access to medical assistance more flexible and effective in emergency situations.

The developed medical information system has become an efficient tool for ensuring access, exchange, and management of medical information, considering contemporary requirements for the storage and processing of medical data.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	8
ВСТУП .....	9
1 АНІЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	11
1.1 Особливості сучасних тенденцій обміну медичною інформацією .....	11
1.2 Вимоги до медичної інформаційної системи .....	13
1.3 Аналіз функціональних вимог і можливостей системи .....	14
1.4 Апаратне програмне забезпечення.....	16
1.5 Мета та постановка завдання .....	19
2 ПРОЕКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ .....	21
2.1 Вибір технологій для розробки медичної інформаційної системи .....	21
2.2 Функціональні блоки та їх взаємодія.....	25
2.3 Опис бази даних та функціональних особливостей .....	27
3 РОЗРОБКА ОСОБИСТОГО КАБІНЕТУ КОРИСТУВАЧА .....	31
3.1 Реєстрація та авторизація користувача .....	31
3.1.1 Реєстрація користувача.....	31
3.1.2 Реєстрація нового користувача.....	32
3.2 Профіль користувача.....	36
3.2.1 Профіль кабінету для лікаря.....	36
3.2.2 Профіль кабінету для працівника лабораторії та діагностики.....	47
3.2.3 Профіль кабінету пацієнта.....	48
4 ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ.....	54
4.1 Розробка тестових кейсів .....	54
ВИСНОВКИ .....	58
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	60

ДОДАТОК А Графічна частина кваліфікаційної роботи.....	62
ДОДАТОК Б Лістинг коду програм.....	72

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ID – ідентифікатор;

ПЗ – програмне забезпечення;

МКХ-10-АМ – міжнародна класифікація хвороб

АКМІ – Австралійський класифікатор медичних інтревенцій

МІС – медична інформаційна система

MySQL - My Structured Query Language

XML - eXtensible Markup Language

JSON - JavaScript Object Notation

CSV - Comma-Separated Values

PHP - Hypertext Preprocessor

API - Application Programming Interface

REST - Representational State Transfer

## ВСТУП

У світі, що стрімко переходить до цифрової ери, роль електронних систем віддаленого доступу набуває парадигмального значення. Нестримний розвиток технологій та зростаюча потреба в комфортному та безпечному отриманні послуг підштовхують до створення ефективних інструментів для доступу до різноманітних сервісів.

В найближчому майбутньому очікується, галузь охорони здоров'я перейде до, так званої, профілактичної медицини. Віртуальні платформи будуть відігравати ключову роль в цій трансформації. Мета сучасної моделі медицини спрямована на створення ціннісного підходу до лікування, що зосереджене головним чином на результат. В даний час в таких секторах, як фінанси, електронна документація та медицина, спостерігається їх перетворення завдяки електронним системам віддаленого доступу. Ці системи не тільки полегшують повсякденне життя, але й стають критичними для забезпечення безпеки та доступності послуг у важливі моменти, особливо в періоди надзвичайних ситуацій, таких як пандемія Covid-19.

Пандемія послужила своєрідним викликом для медичної галузі, підкресливши необхідність негайного розвитку та впровадження медичних інформаційних систем. Обмеження фізичних контактів та необхідність швидкого доступу до медичної допомоги підкреслили важливість електронних платформ для консультацій та доступу до медичних послуг. У цьому контексті створення медичної інформаційної системи з особистим кабінетом користувача є не тільки актуальним, але й стратегічно важливим завданням. Такий особистий кабінет створює можливість надати кожному користувачу зручний, інтуїтивний та безпечний доступ до власної медичної інформації, забезпечуючи необхідну конфіденційність та зручність у взаємодії з медичними послугами в онлайн-режимі.

Метою кваліфікаційної роботи є розробка віртуальної системи

управління медичним закладом з особистим кабінетом користувача. Дана система спрямована на створення інтерфейсу, що поєднує в собі зручність, функціональність та безпеку, щоб забезпечити користувачам максимальний комфорт у керуванні власною медичною інформацією.

## 1 АНЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

В розділі проаналізовано функціональні особливості управління віртуальними медичними системами, розглянуто їх застосування на практиці.

### 1.1 Особливості сучасних тенденцій обміну медичною інформацією

Сучасна тенденція цифрових перетворень системи управління та надання послуг з використанням технологій Інтернет речей [1] спонукає до оцифровізації сфери охорони здоров'я в Україні з метою забезпечення прозорості та ефективності управління нею. Впровадження системи електронного здоров'я має суттєве значення для розвитку електронного урядування в Україні.

Дедалі більше використовуються сучасні технології для поліпшення надання медичних послуг та забезпечення комфорту пацієнтів. З основних тенденцій, що спостерігається, можна відмітити наступні, що наведені нижче.

Віддалений доступ до медичних послуг. За допомогою електронних систем віддаленого доступу, пацієнти можуть отримати консультацію лікаря або взяти участь у віртуальних прийомах без потреби фізичної присутності в медичній установі. Це особливо було актуально в умовах пандемії COVID-19, коли зменшення фізичних контактів є важливим заходом для збереження здоров'я.

Зберігання та обмін медичною інформацією. Електронні системи дозволяють зручно та безпечно зберігати медичну інформацію пацієнтів, таку як медичні записи, результати досліджень, рецепти та інші дані. В результаті дані можливості дозволяють лікарям швидко отримувати доступ до необхідної інформації та забезпечувати її безпеку та цілісність [2].

Мобільні застосунки для медичної допомоги. Зростання використання мобільних технологій призвело до розробки мобільних застосунків, які дозволяють пацієнтам отримувати доступ до медичних послуг та інформації на своїх смартфонах або планшетах. Це дає можливість зручно керувати своїм здоров'ям, записуватися на прийоми, переглядати результати досліджень та отримувати нагадування про ліки.

Інтеграція з іншими системами. Медичні інформаційні системи повинні мати здатність інтегрувати з іншими системами, такими як лабораторні системи, системи управління запасами та системи електронної медичної документації з метою покращення ефективності роботи, уникнення повторної введення даних та забезпечення потрібну інформацію відразу всім зацікавленим сторонам [3].

Забезпечення безпеки та конфіденційності. Медична інформаційна система повинна мати високий рівень безпеки для захисту конфіденційності медичних даних пацієнтів. Це включає застосування шифрування, контролю доступу та забезпечення аудиту доступу до інформації.

Телемедицина. Розвиток телекомунікаційних технологій дозволяє проводити консультації та обстеження пацієнтів дистанційно. Це особливо корисно для людей, які проживають в віддалених районах або мають обмежену мобільність. Телемедицина також забезпечує можливість взаємодії між лікарями для консультацій та обміну досвідом.

Геноміка та персоналізована медицина. Використання генетичних досліджень та аналізу геномів пацієнтів дозволяє розуміти індивідуальні особливості організму та розробляти персоналізовані методи лікування, що відкриває нові можливості для точного діагностування, прогнозування ризиків та вибору оптимального лікування.

Штучний інтелект і машинне навчання. Використання штучного інтелекту та машинного навчання дозволяє аналізувати великі обсяги медичних даних, виявляти патерни та тренди, а також підтримувати прийняття рішень у медицині. Наприклад, системи штучного інтелекту

можуть допомагати у виявленні захворювань на ранній стадії, прогнозуванні ходу лікування та розробці індивідуальних підходів до пацієнтів.

## 1.2 Вимоги до медичної інформаційної системи

Враховуючи сучасні тенденції в медичній сфері, вимоги до медичної системи повинні включати наступні аспекти:

- забезпечення безпеки та конфіденційності;
- інтеграція та обмін даними;
- електронні медичні записи;
- віддалений доступ;
- телемедицина;
- мобільний доступ та мобільні застосунки;
- підтримка досліджень та аналітики.

Медичні показники або дані пацієнта є чутливою інформацією. Тому медична система повинна мати високий рівень захисту даних та забезпечувати конфіденційність медичної інформації пацієнтів. Це включає шифрування даних, контроль доступу, аудит доступу до інформації та відповідність вимогам щодо захисту персональних даних. Для нормальної роботи системи користувачів потрібно розділити за ролями та правами доступу.

Система повинна розрізнявати типи користувачів, таких як:

- адміністратор системи з повними правами доступу до налаштувань роботи системи;
- головний адміністратор, для якого відводиться головна роль в управлінні системою. Головний адміністратор повинен мати змогу додавання або видалення лікарів з системи та налаштовувати функціональні особливості системи;
- лікар. Повинен бачити інформацію про своїх пацієнтів, їх медичні показники, лабораторні дослідження;

- пацієнт. Доступ до особистої інформації, історії відвідувань, лабораторних досліджень, діагнозів тощо;
- медичний працівник (медсестра або адміністратор). Може записувати пацієнта до лікаря, редагувати інформацію про прийом, змінювати част та дату прийому;
- фінансовий фахівець. Може виставляти рахунки на оплату корегувати ціни, управляє ціновою політикою медичного закладу.

Медична інформаційна система повинна бути здатною інтегруватися з іншим програмним забезпеченням, які використовуються в медичній установі, наприклад, лабораторними системами, та системами електронної медичної документації. Це дозволяє забезпечити доступ до необхідної інформації для пацієнта та лікаря. Система повинна підтримувати створення та зберігання інформації, яку вносить сам користувач.

Медична система має підтримувати можливість віддаленого доступу до медичних послуг та консультацій для пацієнтів. Це включає віртуальні прийоми, електронну консультацію, телемедицинські послуги тощо. Важливо, щоб система забезпечувала надійне з'єднання та безпечну передачу медичних даних.

Медична інформаційна система може забезпечувати підтримку досліджень та аналітику, використовуючи дані, зібрані з різних джерел [4]. Це дозволяє використовувати дані для досліджень, забезпечувати поліпшення якості медичної допомоги та оптимізації процесів у медичній установі.

### 1.3 Аналіз функціональних вимог і можливостей системи

Аналіз функціональних вимог і можливостей медичної інформаційної системи допомагає визначити, які конкретні функції повинна виконувати система для задоволення потреб медичного закладу.

Система повинна мати можливість реєструвати нових пацієнтів та зберігати їх основну інформацію, таку як особисті дані, контактна інформація, медична історія тощо.

В кожного пацієнта є свій унікальний ID номер, який призначається програмним забезпеченням медичної установи. По номеру ID має відбуватись завантаження даних до особистого кабінета. В самій системі потрібно зробити верифікацію користувача особистого кабінету з ПЗ медичної установи.

В особистому кабінеті користувач повинен бачити інформацію про розклад графік роботи, та вільний час спеціаліста. Найменування послуг які надає медичний заклад. Пацієнти повинні мати можливість записатися на прийом за допомогою системи, наприклад, через інтернет або мобільний застосунок.

Система має забезпечувати створення та зберігання електронних медичних записів пацієнтів. Це включає інформацію про медичні візити, діагнози, рецепти, результати обстежень та лабораторних досліджень.

Електронні записи про візити мають містити в собі:

- час та дату прийому;
- спеціаліста, до якого звертався пацієнт;
- діагнози, відображені у вигляді МКХ-10-АМ та АКМІ;
- лабораторні дослідження та призначені лікарські засоби.

Деякі лабораторні дослідження мають відображатися у вигляді графіка, час - норма - результат. Таким чином, система буде наглядно показувати зміни в результатах дослідження та стану здоров'я.

Медичні дані пацієнта є чутливою інформацією, тому система буде розділена на декілька частин, які розглянуті далі.

ПЗ медичної установи - в даному місці зберігається вся основна інформація про пацієнта, історія звернень, діагнози, лабораторні дослідження та інше.

Центр обробки даних - серверно-програмне забезпечення, яке буде обробляти запити від користувача, та за потреби надсилати до програмного забезпечення медичної установи [5]. В даному місці буде зберігатися інформація про користувача, яку він бажає зберегти.

Користувач - після автентифікації, користувачу буде надана вся інформація яка зберігається в програмному забезпеченні медичної установи та на серверах медичної інформаційної системи (рис. 1.1).

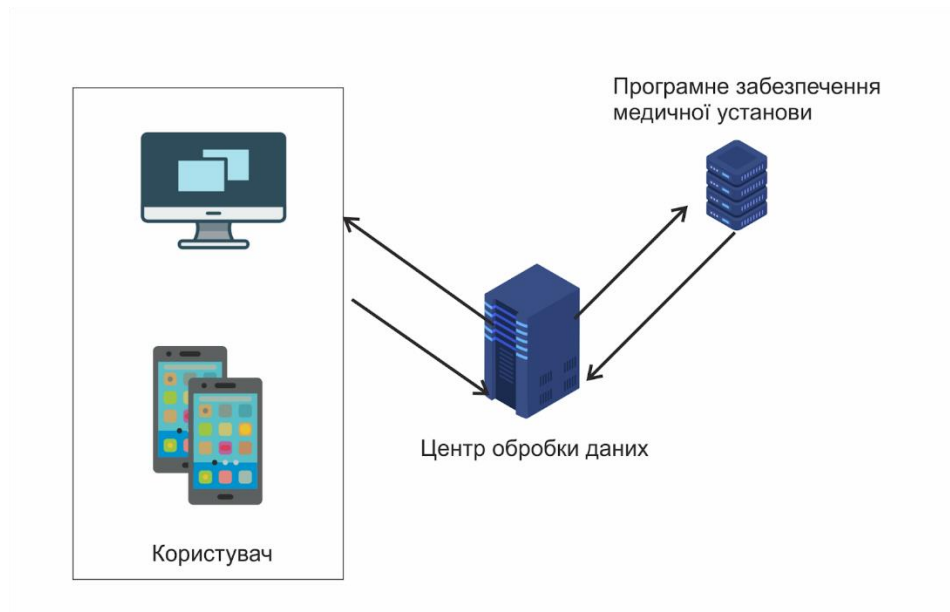


Рисунок 1.1 - Загальна структура системи

#### 1.4 Апаратне програмне забезпечення

Система медичної інформаційної системи (МІС) вимагає надійного та продуктивного сервера для забезпечення ефективної роботи та обробки великого обсягу медичних даних [6]. Необхідно використовувати сервер із високопродуктивним процесором, таким як Intel Core i5 або аналогічний, для забезпечення швидкісної обробки та відповідності вимогам сучасних медичних додатків.

Процесор класу Intel Core i5 може забезпечити достатню потужність для виконання різноманітних операцій, пов'язаних із зберіганням, обробкою

та аналізом медичних даних. Це особливо важливо у сфері охорони здоров'я, де частіше виникають вимоги до великої швидкості обробки інформації для надання миттєвої медичної допомоги.

Додатково, рекомендується наявність щонайменше 8 ГБ оперативної пам'яті для забезпечення плавної роботи системи та обробки багатьох запитів одночасно. Інтеграція SSD-накопичувача обсягом не менше 256 ГБ дозволить швидко завантажувати дані та запускати додатки, забезпечуючи високу швидкість доступу до інформації.

Використання надійної реляційної бази даних (РБД) є критично важливим аспектом для ефективної роботи системи медичної інформаційної системи (МІС). В даному випадку необхідно обрати між двома найбільш визнаними та надійними РБД - MySQL та PostgreSQL.

MySQL відомий своєю простотою в управлінні та встановленні, а також широко використовується в галузі розробки. Ця система баз даних добре підходить для середніх за розміром проєктів та забезпечує добру продуктивність у багатьох випадках. MySQL часто обирається за своєю простотою та швидкодією у виконанні простих операцій [7].

PostgreSQL є більш потужним та гнучким рішенням, що надає розширені можливості для управління даними та підтримує високий рівень стандартів SQL. Він забезпечує надійність та роботу з великими обсягами даних. PostgreSQL також підтримує широкий спектр типів даних та має високу рівень консистентності даних.

Забезпечення сумісності та оптимального користувацького досвіду на клієнтських пристроях є важливим аспектом проєктування медичної інформаційної системи. Нижче розглянуті деякі ключові вимоги до клієнтських пристроїв:

Сучасні веб-браузери. Користувачі системи повинні мати можливість користуватися сучасними веб-браузерами для найкращого функціоналу та безпеки. Рекомендовано використовувати браузери, такі як Google Chrome,

Mozilla Firefox та Safari, які підтримують останні технології веб-розробки та забезпечують безпечне середовище для користувачів.

Мінімальне розширення екрану. З метою забезпечення комфортного відображення інтерфейсу системи важливо, щоб клієнтські пристрої мали мінімальне розширення екрану не менше 1280x800 пікселів. Це забезпечить достатній простір для відображення інформації та зручне використання функцій системи.

Респонсивний дизайн. Розробка інтерфейсу системи повинна враховувати різні розміри екранів та пристроїв. Використання респонсивного дизайну дозволить адаптувати веб-інтерфейс до різних розмірів екранів, забезпечуючи зручне користування на різних пристроях, таких як комп'ютери, планшети та смартфони.

Безпека та приватність. Враховуючи особливості обробки медичної інформації, важливо визначити та дотримуватися вимог щодо безпеки та приватності веб-з'єднання. Використання шифрування та забезпечення конфіденційності даних допоможе запобігти можливим загрозам.

Забезпечення регулярного та надійного створення резервних копій бази даних та файлів є критично важливою частиною стратегії управління даними. Нижче наведено деталізацію цього процесу:

Регулярність резервних копій. Система повинна автоматично створювати регулярні резервні копії бази даних та файлів. Рекомендується встановити графік, що враховує обсяг та частоту змін в базі даних, для мінімізації можливого втрати даних.

Методи зберігання резервних копій. Застосування зовнішніх засобів для зберігання резервних копій є розумним підходом. Це може включати в себе використання хмарних сервісів (наприклад, Amazon S3, Google Cloud Storage) або віддалених серверів. Такий підхід гарантує, що резервні копії залишаються недоступними для можливих проблем, таких як атаки на сервер чи фізичні пошкодження обладнання.

Автоматизація процесу. Процес створення резервних копій повинен бути повністю автоматизованим. Використання спеціальних програм або інструментів для автоматичного виконання цього завдання забезпечить його регулярність та точність.

Перевірка та відновлення. Система повинна регулярно проводити тести відновлення, переконуючись, що резервні копії можуть бути успішно відновлені в разі необхідності. Це забезпечить ефективність процесу та впевненість у відновленні даних.

Захист від несанкціонованого доступу. Резервні копії повинні бути захищені від несанкціонованого доступу. Застосування шифрування для зберігання та транспортування резервних копій забезпечить конфіденційність інформації в них.

### 1.5 Мета та постановка завдання

Метою кваліфікаційної роботи є розробка медичної інформаційної системи, та реалізація особистого кабінету користувача.

Основним завданням є створення проекту є створення зручного та функціонального інтерфейсу, який дозволить користувачам комфортно та безпечно керувати своєю медичною інформацією.

Для досягнення поставленої мети потрібно вирішити наступні задачі:

- аналіз вимог до створюваної системи;
- аналіз функціональних та нефункціональних вимог до особистого кабінету користувача;
- розробка архітектури для управління медичною системою;
- реалізація основних функцій медичної системи;
- зручний інтерфейс для комунікації з медичним персоналом;
- тестування та валідацію віртуальної медичної системи.

Особистий кабінет користувача повинен мати наступні функції:

- аутентифікацію, авторизацію користувача, відображення медичної інформації;
- можливість перегляду медичної історії, включаючи дані про попередні хвороби, проведені процедури, призначення ліків та інші записи;
- зручний перегляд та завантаження медичних звітів, аналізів, рентгенівських знімків та інших досліджень;
- можливість записатися на прийом до лікаря або скасувати існуючий запис;
- отримання сповіщень та нагадувань про заплановані візити, процедури або прийом ліків;
- зручний інтерфейс для комунікації з медичним персоналом, наприклад, чат або система обміну повідомленнями;
- оновлення даних за допомогою підключення файлів типу XML, JSON, CSV.

Після розробки системи необхідно провести тестування та валідацію, щоб переконатися в її коректній роботі та відповідності вимогам. На основі отриманих результатів можна зробити висновки щодо ефективності та якості системи, а також надати рекомендації щодо подальшого розвитку та вдосконалення.

## 2 ПРОЕКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ

В поточному розділі описується критерії, яким повинна відповідати сучасна віртуальна система управління медичним закладом. Розглянуто стандарти обміну медичними даними. Розуміння критерії та стандартів дозволило реалізувати функціональні блоки системи, та розробити базу даних.

### 2.1 Вибір технологій для розробки медичної інформаційної системи

При проектуванні медичної інформаційної системи для медичної установи були обрані технології, які відповідають ефективній реалізації, що приведені нижче:

- доступність системи для всіх користувачів;
- гнучкість системи;
- модульність;
- масштабування.

Найбільш відповідним до цих критеріїв є створення web - версії системи. Яка може бути доступна з будь-якого пристрою, який підключений до мережі Internet та адаптуватися під найбільш використовувані пристрої (Desktop, mobile).

Для розробки WEB-версії медичної інформаційної системи був використаний фреймворк Laravel. Laravel є популярним фреймворком для розробки веб-додатків з високою продуктивністю та зручними інструментами для розробників. Він базується на мові програмування PHP і надає багато функціональних можливостей, які можна використовувати для реалізації функцій медичної інформаційної системи [8].

Основні переваги використання Laravel для проекту приведено нижче.

Екосистема та готові рішення. Laravel має широку екосистему, що включає в себе різні розширення, пакети та готові рішення. Наприклад, пакет Laravel Passport можна використовувати для реалізації системи аутентифікації та авторизації. Також Laravel має вбудовану систему маршрутизації, ORM для роботи з базою даних та інші корисні функції.

Модульність та компонентна архітектура: Laravel дозволяє будувати додатки з використанням модульної архітектури. Функціональність можна розділити на окремі модулі та компоненти, що спрощує розробку, тестування та підтримку системи.

Міграції бази даних: Laravel надає зручні інструменти для управління міграціями бази даних. Можливо створювати, змінювати та відновлювати міграції, що дозволяє контролювати структуру бази даних і забезпечувати легку розгортку системи на різних середовищах.

Шаблонізація та фронтенд: Laravel має вбудовану систему шаблонізації Blade, яка дозволяє створювати зручні та динамічні шаблони для веб-сторінок.

Тестування: Laravel має вбудовану систему для написання автоматизованих тестів, що допомагає забезпечити якість коду та перевірити правильність роботи системи. Створені функціональні та модульні тести для перевірки різних аспектів функціональності системи.

Крім фреймворка Laravel, для розробки WEB-версії будуть використані:

База даних: для зберігання медичної інформації була використана реляційна база даних, MySQL.

Веб-сервер: в системі був використаний веб-сервер Apache для розгортання системи.

Інтерфейси API: якщо потрібна інтеграція з іншими медичними системами, створений функціональний блок API для обміну даними, використовуючи REST або GraphQL протоколи.

Архітектурні принципи: принципи чистої архітектури (Clean Architecture), які розділяють бізнес-логіку, інтерфейси та інфраструктуру. Це сприятиме забезпеченню модульності, розширюваності та тестованості системи.

Інтеграція: для інтеграції з іншими медичними системами використовуються стандартизовані протоколи, такі як HL7 (Health Level Seven) або FHIR (Fast Healthcare Interoperability Resources).

Health Level Seven (HL7) - це стандарт обміну інформацією в галузі охорони здоров'я, який використовується для електронного обміну медичними даними між різними інформаційними системами в охороні здоров'я. HL7 був розроблений некомерційною організацією HL7 International з метою стандартизації та забезпечення взаємодії в галузі охорони здоров'я.

HL7 визначає структуру, формат та протоколи обміну даними між різними системами охорони здоров'я, включаючи системи електронного медичного запису (ЕМЗ), системи управління лікарнями (HIS), лабораторні інформаційні системи (LIS) та інші. Стандарт HL7 дозволяє різним системам взаємодіяти та обмінюватися даними, незважаючи на те, що вони можуть бути розроблені різними виробниками та мати різні технологічні реалізації.

HL7 визначає різні специфікації, відомі як версії стандарту HL7, кожна з яких описує певні типи повідомлень, сегменти даних та синтаксис обміну. Наприклад, HL7 версія 2.x є однією з найбільш поширених версій стандарту і включає специфікації для різних видів повідомлень, таких як повідомлення про прийом пацієнта, результати лабораторних досліджень, замовлення на медичні процедури [9].

HL7 також розробив новішу версію стандарту під назвою HL7 FHIR (Fast Healthcare Interoperability Resources), яка базується на сучасних веб-технологіях, таких як RESTful API, JSON та XML. FHIR надає більш гнучкий та легкий підхід до обміну даними в галузі охорони здоров'я і спрямований на спрощення розробки та впровадження взаємодіючих систем.

В цілому, HL7 відіграє важливу роль в галузі охорони здоров'я, полегшуючи обмін даними між різними системами та сприяючи покращенню якості охорони здоров'я та координації медичної допомоги.

FHIR базується на сучасних веб-стандартах, таких як RESTful API, JSON (JavaScript Object Notation) та XML (eXtensible Markup Language), що робить його більш гнучким і легким у використанні порівняно з попередніми версіями стандарту HL7.

Основні особливості FHIR розглянуті нижче.

Модульність. FHIR використовує поняття ресурсів (Resources), які представляють окремі компоненти медичної інформації, такі як пацієнти, лікарні, лабораторні результати тощо. Це дозволяє використовувати лише необхідні ресурси для конкретних випадків взаємодії, спрощуючи процес обміну даними.

Простота використання. FHIR використовує зрозумілу семантику і просту структуру даних, що полегшує розробку і використання систем, побудованих на його основі. За допомогою RESTful API, розробники можуть звертатися до ресурсів, отримувати та передавати дані, використовуючи звичні HTTP-запити.

Інтероперабельність. FHIR спрямований на створення стандартів та розширень, які дозволяють системам охорони здоров'я взаємодіяти між собою, незалежно від їхнього походження або рівня складності. Це сприяє поліпшенню обміну даними, об'єднанню інформації з різних джерел та підвищенню якості медичного обслуговування.

Розширюваність. FHIR надає можливість розширювати стандарт, додавати власні ресурси та визначати нові елементи даних відповідно до конкретних потреб організацій в галузі охорони здоров'я.

FHIR став все більш поширеним і використовується як великими медичними закладами, так і розробниками медичного програмного забезпечення для покращення взаємодії та обміну медичною інформацією в інформаційних системах охорони здоров'я.

Безпека. Для забезпечення конфіденційності та безпеки медичної інформації використовується шифрування даних, механізми аутентифікації та авторизації, а також заходи для запобігання вторгнень (наприклад, захист від SQL-ін'єкцій та кросс-сайтових скриптів).

## 2.2 Функціональні блоки та їх взаємодія

Віртуальна система управління медичним закладом розроблена з врахуванням різноманітних функціональних блоків для того, щоб найкращим чином відповідати потребам користувачів. Основні з них включають аутентифікацію та авторизацію, реєстрацію користувачів, онлайн оплату послуг та розклад графіку роботи та запис на прийом.

Аутентифікація та авторизація відіграють ключову роль у системі, забезпечуючи ідентифікацію користувачів та контроль їх рівнів доступу. Система підтримує кілька варіантів реєстрації, включаючи введення особистих даних через спеціальне посилання та реєстрацію, проведену головним адміністратором.

Система надає можливість онлайн оплати медичних послуг. Це включає в себе реалізацію безпечних методів електронного платежу, щоб спростити процес оплати для користувачів та полегшити фінансовий облік.

Функція розкладу графіку роботи лікарів і система запису на прийом є ключовою для зручного взаємодії пацієнтів із медичним закладом. Вона дозволяє користувачам переглядати графік роботи лікарів та обирати зручний для них час для запису на прийом.

Після успішної реєстрації система автоматично відправляє лист на електронну пошту користувача з посиланням для підтвердження реєстрації. Це забезпечує безпеку та конфіденційність особистої інформації користувача.

Верифікованим користувачам надається доступ до повного функціоналу особистого кабінету, де вони можуть переглядати свою медичну

інформацію, отримувати результати досліджень, записуватися на прийом та виконувати інші необхідні дії. Не верифікованим користувачам надається узагальнююча інформація про графік роботи, але без відображення та вибору вільного часу лікаря. Також для таких користувачів не доступна інформація про історію відвідувань, лабораторних досліджень, та будь яких записів, які стосуються цього пацієнта.

Управління медичними записами. Дозволяє створювати, зберігати та оновлювати електронні медичні записи пацієнтів, включаючи діагнози, рецепти, результати обстежень та лабораторних досліджень.

Додавання власних записів. Верифікованим користувачам, надана можливість введення власних описів про стан свого здоров'я, а саме: ритм серця, тиск, вага, рівень цукру, та інші.

Ці дані доступні для лікаря у якого обстежується пацієнт. Також користувач може бачити зміни свої показників здоров'я у вигляді графіків.

Візуалізація результатів досліджень. Забезпечує можливість відобразити результати лабораторних досліджень у вигляді графіків або інших наглядних форматів для аналізу змін у стані здоров'я. Також користувач зможе завантажити результати лабораторних досліджень у форматі PDF.

Інтеграція. Медична система не зберігає в собі медичну історію, результати лабораторних досліджень, рецепти та інші записи які стосуються медичної інформації.

Всі дані користувач отримує з бази даних програмного забезпечення медичного закладу. Для цього розроблені модулі інтеграції та обміну даними між системами. Зокрема, обмін даним можна проводити за допомогою передачі даних в XML, CSV або JSON форматах. При передачі цих даних, в файлах не буде міститись персональної інформації.

Забезпечення доступу до інформації про пацієнтів: Медична інформаційна система здатна обмінюватися даними з модулем лікарні, що дозволяє лікарям отримувати доступ до повної медичної історії пацієнта,

результатів обстежень, діагнозів, призначень лікування та іншої важливої інформації, необхідної для надання належних медичних послуг.

**Інтеграція з лабораторіями:** Система забезпечує можливість передавати запити на проведення лабораторних досліджень до лабораторій та отримувати результати цих досліджень. Це дозволяє лікарям отримувати актуальні дані про стан пацієнта, а також забезпечує автоматизовану обробку результатів досліджень у медичній системі.

**Обмін даними з аптеками:** Система повинна підтримує обмін даними з аптеками для забезпечення замовлення та отримання лікарських препаратів для пацієнтів. Це включає передачу електронних рецептів, замовлення та підтвердження поставок ліків, контроль залишків препаратів тощо.

Інтеграція з модулем лікарні, лабораторіями та аптеками дозволяє забезпечити плавний обмін даними, уникнути помилок та зберегти час медичного персоналу, сприяючи ефективному наданню медичних послуг та оптимальному використанню ресурсів.

### 2.3 Опис бази даних та функціональних особливостей

Для ефективного та безпечного зберігання медичної інформації було вирішено розробити базу даних, що включає в себе різні таблиці для різних типів даних, таких як пацієнтські дані, історії хвороб, результати аналізів тощо.

Структура бази даних побудована з урахуванням потреб лікарів, медсестер, адміністраторів та інших користувачів. Вона має забезпечувати швидкий доступ до актуальної інформації та зручні інструменти для роботи з даними [10].

База даних створена з урахуванням функціональних блоків системи. При розробці бази даних враховувалася модульність системи. База даних складається з наступних основних таблиць, що описані нижче.

Таблиця користувача (Users) містить:

1) унікальний ідентифікатор користувача. Це автоматично збільшуване ціле число;

2) ID відділу, до якого належить користувач. Зовнішній ключ, який посилається на ID відділу;

3) ім'я користувача;

4) прізвище користувача;

5) електронна адреса користувача;

6) хеш пароля користувача.

7) номер телефону;

8) стать;

9) дата народження;

Таблиця лікаря містить:

1) унікальний ідентифікатор спеціаліста;

2) ID користувача, пов'язаного із спеціалістом (зовнішній ключ, посилається на ID з таблиці 'users');

3) ID відділення лікарні, до якого відноситься спеціаліст;

4) назва або опис спеціальності спеціаліста;

5) поля для відстеження часу створення та останнього оновлення запису.

Також база даних має таблиці:

- таблиця послуг медичного закладу;

- таблиця лабораторних досліджень;

- таблиця лікарських рецептів;

- таблиця керування вільними місцями;

- таблиця рахунків, оплати, та інші.

У контексті розробки та управління медичною інформаційною системою (МІС), безпека та конфіденційність даних є критичними аспектами. Одним із ключових елементів забезпечення цієї безпеки є застосування високої якості методів шифрування для особистих та конфіденційних даних.

В реалізації системи використані сучасні методи шифрування, такі як bcrypt для хешування паролів користувачів. Bcrypt, як адаптований алгоритм хешування паролів, використовує хеш-функції та (salt) для ефективного ускладнення атак на витягування даних.

Особисті дані користувачів, такі як номер телефону, ім'я та прізвище, також піддаються захисту через хешування. Використання хеш-функцій, таких як MD5 або SHA-256, дозволяє захистити ці дані від несанкціонованого доступу або зловмисних атак.

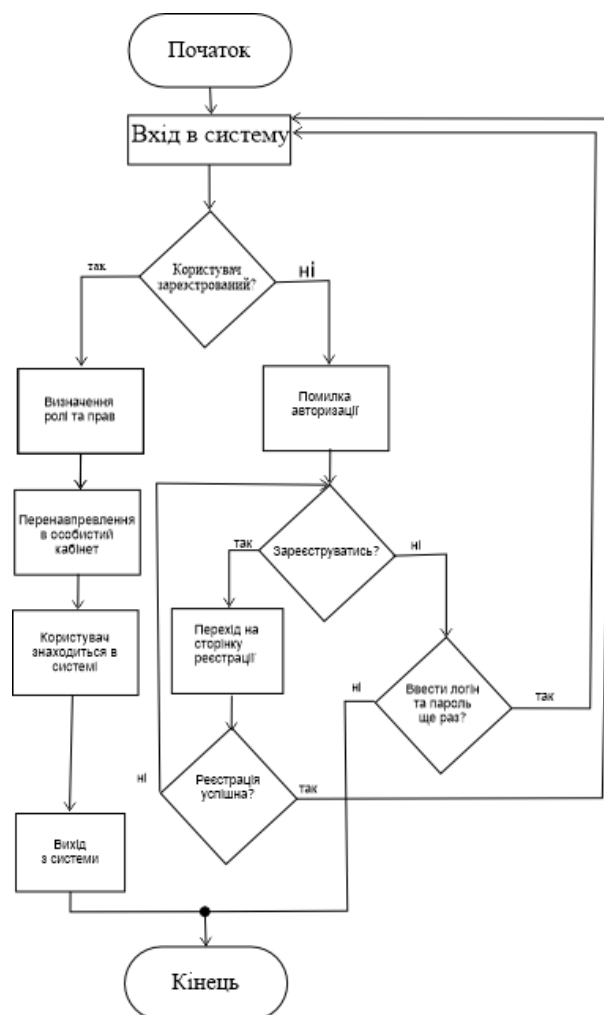


Рисунок 2.1 – Алгоритм входу користувача в систему

Забезпечення безпеки бази даних є критичним завданням. Ретельно підібрані алгоритми шифрування для зберігання конфіденційної інформації

в базі даних, такі щоб навіть у випадку несанкціонованого доступу до даних, вони залишалися недоступними для розшифрування.

Реалізація системи контролю версій для забезпечення цілісності та відстеження змін в базі даних. Забезпечення високої доступності та ефективної роботи бази даних – це невід'ємна частина розробки систем, особливо коли йдеться про додатки з високим рівнем завантаження та критичні для бізнесу сервіси. Використання технік реплікації є ключовим аспектом для досягнення цих цілей.

Однією з найважливіших переваг використання реплікації є можливість створювати копії основної бази даних на віддалених серверах. Це забезпечує високу доступність системи, оскільки у випадку відмови чи виходу з ладу основного сервера, репліки можуть негайно взяти на себе обслуговування, забезпечуючи безперервну роботу сервісу.

Використання реплікації дозволяє розподілити читальне та записувальне навантаження між основною базою та репліками. Це полегшує тиск на основний сервер та дозволяє оптимізувати роботу системи за рахунок паралельності обробки запитів.

Реплікації дозволяють застосовувати шифрування для безпечної передачі даних між репліками та основною базою. Це забезпечує конфіденційність та цілісність інформації.

Не менш важливий етап в проектуванні бази даних - це оптимізація запитів та індексація в системі. Який має на меті підвищення продуктивності та швидкодії витягування інформації.

Перед проектуванням важливо використовувати інструменти профілювання запитів для виявлення обтяжливих та неефективних конструкцій SQL-коду. Це надасть змогу зосередитися на оптимізації саме тих запитів, які суттєво впливають на продуктивність.

## 3 РОЗРОБКА ОСОБИСТОГО КАБІНЕТУ КОРИСТУВАЧА

В даному розділі описані методи реєстрації нових користувачів їх ролі та надання прав групам користувачів. Роз'яснені особливості в роботі кожної групи користувача, дії одних і тих функціональних блоків для різних ролей.

### 3.1 Реєстрація та авторизація користувача

Для створення web версії медичної інформаційної системи був обраний фреймворк Laravel. Він забезпечує зручний та елегантний спосіб розробки веб-додатків, надаючи широкий набір функціональностей та інструментів для покращення продуктивності розробника.

#### 3.1.1 Реєстрація користувача

Для ефективної реалізації функціоналу реєстрації та авторизації користувачів у системі передбачено впровадження системи управління ролями та правами. Ця система надає можливість диференціації доступу до функцій та ресурсів для різних категорій користувачів. В проекті передбачено наступні ролі:

- головний адміністратор. Має повний доступ до всіх функцій та ресурсів системи. Може керувати ролями та правами інших користувачів. Здатен створювати нових користувачів та призначати їм ролі;
- бухгалтер. Має доступ до функцій, пов'язаних з фінансовим управлінням та обліком. Зможе переглядати та редагувати фінансові дані.
- лікар. Має можливість переглядати медичні записи пацієнтів. Здатен створювати та редагувати медичні записи.
- медсестра. Має обмежений доступ до медичних записів пацієнтів та може здійснювати деякі редагування в медичних записах за певними правилами.

- лаборант: Може додавати результати аналізів пацієнта.
- фармацевт: Керує запасами ліків.
- пацієнт. Має обмежений доступ до особистої інформації та медичних записів. Зможе заповнювати анкети та переглядати власні медичні дані.

### 3.1.2 Реєстрація нового користувача

Для реєстрації нового користувача потрібно створити відповідну таблицю з полями в MySQL. (рис 3.1)

Таблиця «Users» містить:

- Id: унікальний ідентифікатор користувача в системі;
- department\_id: id підрозділу до якого належить користувач;
- first\_name: ім'я користувача. Шифроване поле;
- last\_name: прізвище користувача. Шифроване поле;
- email: email користувача, унікальний в системі;
- password: пароль для входу в особистий кабінет;
- designation: службове поле для медичного працівника;
- phone: номер телефону користувача, унікальний в системі;
- gender: стать користувача;
- qualification: посада;
- blood\_group: група крові;
- created\_at: дата реєстрації користувача;
- dob: дата народження

	id	department_id	first_name	last_name	email	password	designation	phone	gender	qualification	blood_group
	1	1	Super	Admin	admin@hms.com	\$2y\$10\$7TN2ftzMTSOcJh3nBvZ0pUbs9MMr7ZT70aRzMvPmr...	NULL	7878454512	1	NULL	B+
	5	3	Sergey	Kholosha	sergey.kholosha@gmail.com	\$2y\$10\$z70IZKlvoutTXHtp83ELeuorq1UXEH6veMdt9ILfaG...	NULL	+380975486799	0	NULL	NULL
	6	3	Сергій	Холоша	serhil.kholosha@gmail.com	\$2y\$10\$PYM1D9SeP8MguxQ0rdLe4RCAQXZQif7Jgm6VX2kk2b...	NULL	500647211	0	NULL	NULL
	7	2	Володимир	Шевченко	admin@skh.com.ua	\$2y\$10\$THe4E1wOga3YsXRfm3.kedcblHe7MlyujM5DqsyXHg...	Кардіолог	+380970001234	0	Kardionoria	NULL
	8	5	reception	User	receptionists@hms.com	\$2y\$10\$JjO9Yz34mzIYr7q8qlaOjQBz6OfirLtojnY3m2WWqy...	NULL	+380950000000	0	NULL	NULL
	9	9	lab	user	lab@skh.com.ua	\$2y\$10\$3q7vVoPMN2sfuy7lPdykutmZufJ78f0f9fEboYw...	LAB	+3800960000000	0	lab	1

Рисунок 3.1 - Приклад таблиці “Users”

Після створення відповідних полів в базі даних, потрібно налаштувати форму реєстрації та визначити поля які будуть обов'язковими. Як було вказано в підрозділі 2.2 Функціональні блоки та їх взаємодія, користувач повинен ввести свій номер телефону, email, та пароль. Для зручності вводу та уніфікації всіх номерів телефонів, в поле “Телефон” додана маска, яка не дозволяє введення номеру з більшою або меншою кількістю цифр. Запит на створення таблиці “users” приведено у лістингу 3.1, валідація приведена у лістингу 3.2.

### Лістинг 3.1- Запит на створення таблиці “users”

```
public function up(): void
{
    Schema::create('users', function (Blueprint $table) {
        $table->bigIncrements('id');
        $table->bigInteger('department_id')->unsigned()-
>nullable();
        $table->string('first_name', 100);
        $table->string('last_name', 100)->nullable();
        $table->string('email', 100)->unique();
        $table->string('password', 100);
        $table->string('designation', 100)->nullable();
        $table->string('phone', 12)->nullable();
        $table->integer('gender');
        $table->string('qualification', 100)->nullable();
        $table->string('blood_group', 100)->nullable();
        $table->date('dob')->nullable();
        $table->timestamp('email_verified_at')->nullable();
        $table->integer('owner_id')->nullable();
        $table->string('owner_type', 100)->nullable();
        $table->boolean('status');
        $table->string('language', 100)->default('ua');
```

```

$table->rememberToken();
$table->timestamps();
});
}

```

### Лістинг 3.2 - Валідація

```

protected function validator($data)
{
    return Validator::make($data, [
        'first_name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email:filter',
'max:255', 'unique:users'],
        'phone' => ['required'],
        'password' => ['required', 'string', 'min:6', 'confirmed'],
    ], [
        'password.min' =>
__('messages.user.validate_min_character_password'),
    ]);
}

```

На рис 3.2 приведено сторінку з формою реєстрації.

Реєстрація

<p>Ім'я*</p> <input type="text" value="Ім'я"/>	<p>Прізвище*</p> <input type="text" value="Прізвище"/>
<p>EMAIL*</p> <input type="text" value="Email"/>	<p>Телефон*</p> <input type="text" value="+380 50 123 4567"/>
<p>Пароль*</p> <input type="password" value="Password"/>	<p>Підтвердити пароль*</p> <input type="password" value="Підтвердіть Пароль"/>
<p>Стать*</p> <p><input checked="" type="radio"/> Чоловіча <input type="radio"/> Жіноча</p>	
<p><input type="button" value="Зареєструватися"/></p>	

Рисунок 3.2 - Сторінка з формою реєстрації

Після підтвердження коректного заповнення полів, сервіс відправляє на вказаний email код для верифікації. Після верифікації користувачеві відкривається можливість користування особистим кабінетом. Вигляд сторінки створення нового користувача приведений на рис. 3.3.

The screenshot shows a web form titled "Новий користувач" (New User) with a "Назад" (Back) button in the top right corner. The form is organized into two columns:

- Left Column:**
  - Ім'я\*** (Name): Input field with placeholder "Ім'я".
  - Прізвище\*** (Surname): Input field with placeholder "Прізвище".
  - Email:\***: Input field with placeholder "Email".
  - Телефон\*** (Phone): Input field with a country code dropdown set to "+380" and a placeholder "Phone".
  - Стать\*** (Gender): Radio buttons for "Чоловіча" (Male) and "Жіноча" (Female).
- Right Column:**
  - Роль\*** (Role): Dropdown menu with "Select Role" and a downward arrow.
  - Дата народження** (Date of Birth): Input field with placeholder "Дата народження".
  - Пароль\*** (Password): Input field with placeholder "Пароль".

At the bottom right of the form, there are two buttons: "Зберегти" (Save) in blue and "Відмінити" (Cancel) in grey.

Рисунок 3.3 - Вигляд сторінки створення нового користувача

Головний адміністратор має право створювати нових користувачів та обирати для них ролі. Для цього йому потрібно зайти в систему під своїм логіном та паролем, перейти на вкладку «Користувачі» і заповнити відповідні поля.

Доступ для персоналу медичної установи надається тільки головним адміністратором. Головний адміністратор може реєструвати, видаляти, редагувати дані працівників медичної установи.

Перед реєстрацією медичного працівника в системі потрібно створити підрозділи до яких відноситься працівник. Система не дозволить реєструвати працівника без його посади.

Після успішної реєстрації, на email відправляється лист верифікації, перейшовши по посиланню користувач відразу авторизується в системі.

## 3.2 Профіль користувача

Після успішної верифікації в системі, користувачеві будуть запропоновані опції кабінету залежно від ролі.

### 3.2.1 Профіль кабінету для лікаря

Для лікаря розроблено ряд спеціальних функціональних можливостей, спрямованих на полегшення та оптимізацію його робочого процесу. Після успішної авторизації у системі, лікар автоматично перенаправляється на інтуїтивно зрозумілу сторінку історії записів на прийом, яка надає зручний доступ до необхідної інформації.

На цій сторінці історії записів на прийом лікар може використовувати різноманітні фільтри для швидкого доступу до необхідних даних. Зокрема, він може сортувати записи за датою, а також визначати їх статус (виконаний, в очікуванні, скасований).

Додатково, для зручності обробки і аналізу інформації, лікар має можливість експорту даних у формат Excel, що дозволяє зберігати та обробляти дані в зручному для нього форматі (рис. 3.4). Код формування виводу записів на прийом наведено у лістингу 3.3.

ПАЦІЄНТ	ЛІКАР	ЛІКАРСЬКЕ ВІДПОВІДАННЯ	ДАТА	ДІЯ
SK Сергій Халаша sergeykhlosha@gmail.com	ВШ Володимир Шевченко admin@skh.com.ua	кардіологія	11:00 ранку 10 січня 2024 р.	Скасовано
SK Сергій Халаша sergeykhlosha@gmail.com	ВШ Володимир Шевченко admin@skh.com.ua	кардіологія	13:00 10 січня 2024 р.	Підтверджено
SK Сергій Халаша sergeykhlosha@gmail.com	ВШ Володимир Шевченко admin@skh.com.ua	кардіологія	10:00 ранку 10 січня 2024 р.	Підтверджено
SK Сергій Халаша sergeykhlosha@gmail.com	ВШ Володимир Шевченко admin@skh.com.ua	кардіологія	10:00 ранку 10 січня 2024 р.	Підтверджено

Показати 10 Показано 4 результати

Рисунок 3.4 - Вигляд сторінки записів на прийом на сторінці лікаря

## Лістинг 3.3 – Код формування виводу записів на прийом

```

public static $rules = [
    'patient_id' => 'required',
    'doctor_id' => 'required',
    'department_id' => 'required',
    'opd_date' => 'required',
    'problem' => 'string|nullable', ];

public function prepareAppointment()
    { return [
        'id' => $this->id ?? __('messages.common.n/a'),
        'doctor_name' => $this->doctor->doctorUser->full_name ??
        __('messages.common.n/a'),
        'appointment_date' => isset($this->opd_date) ?
        Carbon::parse($this->opd_date)->format('d M, Y') :
        __('messages.common.n/a'),
        'appointment_time' => isset($this->opd_date) ?
        \Carbon\Carbon::parse($this->opd_date)->isoFormat('LT') :
        __('messages.common.n/a'),
        'doctor_department' => $this->department->title ??
        __('messages.common.n/a'),
        'doctor_image_url' => $this->doctor->doctorUser-
        >getApiImageUrlAttribute(), ];}

public function prepareAppointmentForDoctor()
    { return [
        'id' => $this->id ?? __('messages.common.n/a'),
        'patient_name' => $this->patient->patientUser->full_name ??
        __('messages.common.n/a'),

```

```

    'appointment_date' => isset($this->opd_date) ?
Carbon::parse($this->opd_date)->format('jS M, y') :
__('messages.common.n/a'),

    'appointment_time' => isset($this->opd_date) ?
\Carbon\Carbon::parse($this->opd_date)->isoFormat('LT') :
__('messages.common.n/a'),

    'patient_image' => $this->patient->patientUser-
>getApiImageUrlAttribute(), ]; }

```

Для лікаря доступна інформація про пацієнтів. На сторінці пацієнти, доступний список всіх пацієнтів які були на прийомі, або зробили запис (рис. 3.5). Список пацієнтів можливо імпортувати в Excel.

Для спрощення роботи з новими пацієнтами лікарю додана можливість реєструвати в системі нових пацієнтів.

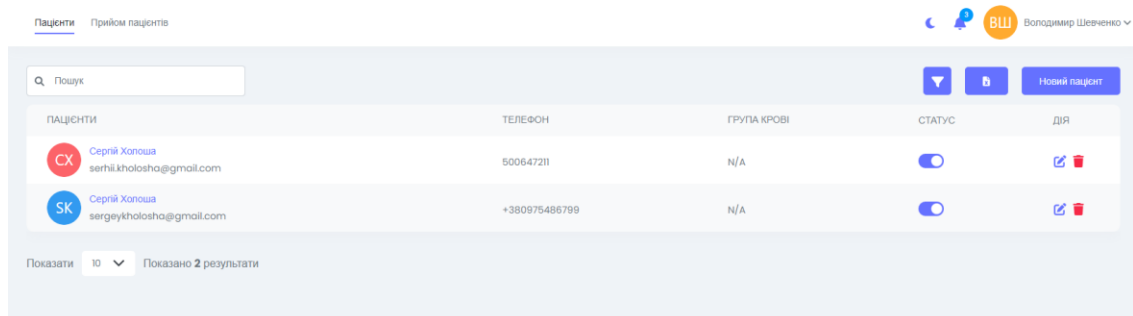


Рисунок 3.5 – Список пацієнтів

При виборі пацієнта для лікаря відкривається наступна інформація:

- загальна інформація;
- інформація про лікування;
- статус рахунків;
- документи які надав пацієнт або завантажили інші працівники медичного закладу;
- інформація про щеплення.

Формування запиту для відображення інформації про пацієнта приведено у лістингу 3.4.

#### Лістинг 3.4 – Формування запиту для відображення інформації про пацієнта

```
public function preparePatientCaseDetailData() {
    return [
        'id' => $this->id,
        'case_id' => $this->case_id,
        'case_date' => isset($this->date) ?
\Carbon\Carbon::parse($this->date)->translatedFormat('jS M,
Y, g:i A') : __('messages.common.n/a'),
        'patient' => $this->patient->patientUser->full_name ??
__('messages.common.n/a'),
        'phone' => $this->phone ?? __('messages.common.n/a'),
        'fee' => $this->fee ?? __('messages.common.n/a'),
        'created_on' => $this->created_at->diffForHumans() ??
__('messages.common.n/a'),
        'description' => $this->description ??
__('messages.common.n/a'),
        'currency' => getCurrencySymbol(), ]; }

```

Крім інформації про пацієнта, для лікаря в системі доступні функції, спрямовані на управління різними аспектами його професійної діяльності.

Керування графіком прийому. Лікар може легко налаштовувати свій графік роботи, визначати доступні для запису години та інші параметри.

Створення запису прийому пацієнта. Функціонал дозволяє лікарю швидко та зручно створювати нові записи на прийом для пацієнтів.

Процес створення рецепту в системі надає лікареві ефективний інструмент для точного та швидкого виписування лікарських призначень пацієнтам. Електронний формат рецептів дозволяє лікареві легко включати в

них різноманітну інформацію, що сприяє якісному наданню медичних послуг та оптимізації процесу лікування.

Під час створення електронного рецепту лікар може включити в нього додаткові дані, такі як номер медичної страховки, номер довідки та соціальний статус пацієнта. Це не лише сприяє збору повної інформації про пацієнта, але й дозволяє формувати пільгові ціни на ліки відповідно до його страхового та соціального статусу.

Система також передбачає можливість завантаження даних про ліки, що дозволяє лікарю швидко знайти потрібний препарат серед доступних в системі. Це забезпечує точність та стандартизацію лікарських призначень, а також виключає можливість помилок через невірний вибір лікувального засобу. Лікар може визначити дозування ліків, розподілити прийоми на різні часи доби та визначити кількість прийомів на день, давши пацієнту детальні та конкретні інструкції. Це дозволяє підтримувати правильний режим лікування та покращує взаєморозуміння між лікарем і пацієнтом.

Для випадків, коли певного препарату немає у списку, передбачена можливість "Додавання нового препарату". Ця функція дозволяє лікареві вручну ввести інформацію про новий препарат, роблячи систему максимально гнучкою та адаптованою до реальних потреб медичної практики (рис 3.6).

Рисунок 3.6 - Вигляд сторінки створення рецепту

## Лістинг 3.5 – Код відтворення форми для створення рецепту

```

<table>
<thead>
<tr>
<th>{{ __('messages.common.no') }}</th>
<th>{{ __('messages.medicine.medicine') }}</th>
<th>{{ __('messages.medicine.brand') }}</th>
<th>{{ __('messages.medicine.category') }}</th>
<th>{{ __('messages.medicine.salt_composition') }}</th>
<th>{{ __('messages.medicine.selling_price') }}</th>
<th>{{ __('messages.medicine.buying_price') }}</th>
<th>{{ __('messages.medicine.side_effects') }}</th>
<th>{{ __('messages.medicine.description') }}</th>
</tr>
</thead>
<tbody>
@foreach($medicines as $medicine)
<tr>
<td>{{ $loop->iteration }}</td>
<td>{{ $medicine->name }}</td>
<td>{{ $medicine->brand->name }}</td>
<td>{{ $medicine->category->name }}</td>
<td>{{ $medicine->salt_composition }}</td>
<td>{{ number_format($medicine->selling_price, 2) }}</td>
<td>{{ number_format($medicine->buying_price, 2) }}</td>
<td>{!!
                !empty($medicine->side_effects)
                ?
nl2br(e($medicine->side_effects))    :    __('messages.common.n/a')
!!}</td>
        <td>{!! !empty($medicine->description) ? nl2br(e($medicine-
>description)) : __('messages.common.n/a') !!}</td>
</tr>
@endforeach
</tbody>

```

</table>

Під час створення нового рецепту лікар повинен заповнити історію хвороби з якою звернувся пацієнт та ввести діагностичні дані (рис. 3.7).

Фізична інформація

Гіпертонія: <input type="text" value="Гіпертонія"/>	Харчова алергія: <input type="text" value="Харчова алергія"/>	Схильність до кровотечі: <input type="text" value="Схильність до кровотечі"/>	Хвороба серця: <input type="text" value="Хвороба серця"/>
діабет: <input type="text" value="Діабетичний"/>	Додано в: <input type="text" value="Додано АТ"/>	Вагітність жінки: <input type="text" value="Вагітність жінки"/>	Грудне вигодовування: <input type="text" value="Грудне вигодовування"/>
Поточні ліки: <input type="text" value="Поточні ліки"/>	Хірургія: <input type="text" value="Хірургія"/>	ДТП: <input type="text" value="ДТП"/>	інші: <input type="text" value="інші"/>
Пульс: <input type="text" value="Пульс"/>	температура: <input type="text" value="температура"/>		
Опис проблеми: <input type="text" value="опис проблеми"/>			

Рисунок 3.7 – Блок для введення діагностичних даних

**Додавання медичних документів:** Можливість завантажувати та прикріплювати медичні документи для збереження та обміну інформацією.

**Запис даних діагностики:** Лікар може вносити результати діагностичних обстежень та ведення медичної історії пацієнтів.

**Створення онлайн консультації:** Можливість проводити консультації з пацієнтами в режимі онлайн, забезпечуючи дистанційний доступ до медичних порад.

Щоб створити онлайн консультацію спочатку потрібно ввести API від програмного забезпечення Zoom (рис. 3.8), обробка запиту приведена у лістингу 3.6.

Рисунок 3.8 - Вікно з полями для введення API Zoom

### Лістинг 3.6 – Обробка запиту зі створення консультації

```

class UserZoomCredential extends Model
{
    public static $rules = [
        'zoom_api_key' => 'required',
        'zoom_api_secret' => 'required',
    ];
    protected $table = 'user_zoom_credential';
    protected $fillable = [
        'user_id',
        'zoom_api_key',
        'zoom_api_secret',
    ];
    protected $casts = [
        'id' => 'integer',
        'user_id' => 'integer',
        'zoom_api_key' => 'string',
        'zoom_api_secret' => 'string',
    ];
    public function user(): BelongsTo
    {

```

```

return $this->belongsTo(User::class, 'user_id');
}
}

```

Після збереження даних API, лікар може створити запрошення на відеоконференцію. Для цього йому потрібно заповнити дані які відображені на формі рисунка 3.9. Після підтвердження форми, пацієнт отримує сповіщення на електронну пошту, та в особистий кабінет.

Нова консультація в прямому ефірі

Назва консультації: \*

Дата консультації: \*

Тривалість консультації Хвилини: \*

Ім'я пацієнта: \*

Ім'я лікаря: \*

тип: \*

Номер типу: \*

Відео ведучого: \*  Увімкнути  Вимкнено

Відео клієнта: \*  Увімкнути

опис:

Рисунок 3.9 - Вікно з полями для запрошення на відео конференцію

Код створення форми для Zoom приведено у лістингу 3.7.

### Лістинг 3.7 – Код створення форми для Zoom

```

$response = $client->request('POST',
'/v2/users/me/meetings', [

```

```

'headers' => [
'Authorization' => 'Bearer '.$zoomOAuth->access_token, ],
'json' => [
'topic' => $data['consultation_title'],
'type' => 2,
'start_time' => $this-
>toZoomTimeFormat($data['consultation_date']),
'duration' => $data['consultation_duration_minutes'],
'agenda' => (! empty($data['description'])) ?
$data['description'] : null,
'password' => '123456',
'settings' => [
'host_video' => ($data['host_video'] ==
LiveConsultation::HOST_ENABLE) ? true : false,
'participant_video' => ($data['participant_video'] ==
LiveConsultation::CLIENT_ENABLE) ? true : false,
'waiting_room' => true,  ],  ],  ]);

```

Створений інтегрований підхід дозволяє лікарю максимально ефективно керувати своєю практикою та забезпечує високий рівень обслуговування для пацієнтів.

Формування звітів: Аналіз та генерація звітів для внутрішнього та зовнішнього використання.

Відправлення SMS-сповіщень пацієнту: Зручна система комунікації для повідомлення пацієнтів про записи, нагадування про прийоми та інше.

Виставлення додаткових рахунків: Можливість формувати та виставляти рахунки за додаткові медичні послуги.

Перегляд лабораторних досліджень: Доступ до результатів лабораторних аналізів та досліджень для забезпечення комплексного медичного обслуговування.

Код запиту для формування даних лабораторних досліджень приведено у лістингу 3.8. Формування PDF файлу з результатами лабораторних досліджень наведено у лістингу 3.9.

### Лістинг 3.8 – Код запиту для формування даних лабораторних досліджень

```
public function create()
{
    $patients = $this->patientRepository->getPatients();
    $doctors = $this->doctorRepository->getDoctors();
    $reportNumber = $this->patientDiagnosisTestRepository-
>getUniqueReportNumber();
    $diagnosisCategory = $this->patientDiagnosisTestRepository-
>getDiagnosisCategory();

    return view('patient_diagnosis_test.create',
compact('patients', 'doctors', 'reportNumber',
'diagnosisCategory')); }

```

### Лістинг 3.9 – Формування PDF файлу з результатами лабораторних досліджень

```
public function convertToPdf(PatientDiagnosisTest
$patientDiagnosisTest) {
    $data = $this->patientDiagnosisTestRepository->getSettingList();
    $data['patientDiagnosisTest'] = $patientDiagnosisTest;
    $data['patientDiagnosisTests'] = $this-
>patientDiagnosisTestRepository-
>getPatientDiagnosisTestProperty($patientDiagnosisTest->id);
    $pdf =
PDF::loadView('patient_diagnosis_test.diagnosis_test_pdf',
$data);
    return $pdf->stream($patientDiagnosisTest->patient->patientUser-
>full_name.'-'. $patientDiagnosisTest->report_number); }
public function patientDiagnosisTestExport() {
    return Excel::download(new PatientDiagnosisTestExport, 'patient-
diagnosis-tests-' . time() . '.xlsx'); }

```

### 3.2.2 Профіль кабінету для працівника лабораторії та діагностики

Профіль кабінету для працівника лабораторії та діагностики визначається рядом важливих функціональних особливостей, спрямованих на ефективне виконання завдань, пов'язаних із медичними дослідженнями та діагностикою. Кожна з цих особливостей сприяє оптимізації робочого процесу та забезпеченню якісного обслуговування пацієнтів.

Функція "Керування банком крові" дозволяє працівнику лабораторії ефективно вести облік та керувати запасами крові. Завдяки цьому інструменту легко визначається наявність різних груп крові, їх кількість та термін придатності, що робить процес планування та організації лабораторних досліджень більш ефективним та безперебійним.

Лаборант може вносити в систему пацієнтів які потребують переливання крові, а також донорів крові. Також, система дозволяє здійснювати моніторинг стану з запасами крові.

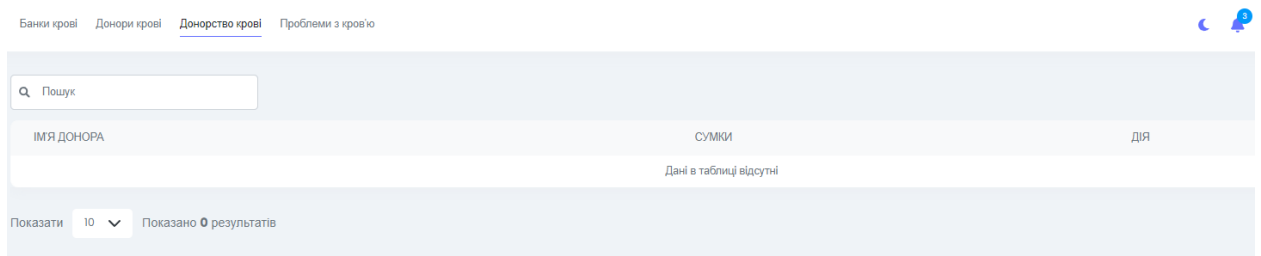


Рисунок 3.10 - Сторінка управління «Банком крові»

Перегляд списку та інформацію про лікаря, надає працівнику доступ до інформації про лікарів, що замовляють лабораторні дослідження. Це включає дані про їхню спеціалізацію, графік роботи, та інші важливі аспекти, сприяючи зручній комунікації між лабораторією та медичним персоналом.

Створення нових лабораторних діагностик, дає можливість працівнику додавати нові методи досліджень або вдосконалювати існуючі. Це дозволяє

лабораторії відповідати на сучасні виклики в галузі медицини та надавати більш точні та надійні результати діагностики.

Створення типів рентгенівських досліджень визначається можливістю працівника лабораторії формувати рентгенівські процедури відповідно до конкретних вимог та потреб пацієнтів. Це важливий інструмент для забезпечення безпеки та ефективності рентгенівських досліджень.

### 3.2.3 Профіль кабінету пацієнта

Профіль кабінету пацієнта в системі надає широкі можливості для зручного та ефективного взаємодії з медичним обслуговуванням. Після успішної авторизації пацієнту стає доступним ряд корисних функцій, спрямованих на покращення процесу отримання медичної допомоги та зручності у веденні особистого медичного обліку.

Однією з ключових можливостей є перегляд інформації про прийоми у лікаря, що надає пацієнту детальну інформацію про назначені та минулі прийоми у лікаря, їхні графіки та розташування.

Для зручності пацієнта реалізована можливість "Створення нового запису до лікаря". Це дозволяє користувачеві самостійно обирати лікаря, відділення, а також вказувати дату та час прийому, надаючи коротку інформацію про причину звернення (рис. 3.11). Пацієнт може також в будь-який момент відмінити запис на прийом, якщо це стане необхідним.

Рисунок 3.11 - Запис на прийом до лікаря через особистий кабінет

Код запиту на формування часу прийому лікаря приведений у лістингу 3.10.

### Лістинг 3.10 – Код запиту на формування часу прийому лікаря

```

public function doctorScheduleList(): JsonResponse {
    $id = getLoggedInUser()->owner_id;
    $schedules = Schedule::with('scheduleDays')-
>where('doctor_id', $id)->first();
    if (empty($schedules)) {
        return $this-
>sendError(__('messages.common.doctor_scheduled').'
'.__('messages.common.not_found'));    }
    $scheduleDays = ScheduleDay::with('schedule')-
>where('doctor_id', $id)->where('schedule_id', $schedules->id)-
>get();
    $per_patient_time =
\Carbon\Carbon::createFromFormat('H:i:s', $schedules-
>per_patient_time)->format('H:i:s');
    $data = [];
    foreach ($scheduleDays as $key => $schedule) {
        $data[] = $schedule->prepareScheduleDay(); }
    $response = [
        'id' => $schedules->id,
        'per_patient_time' => $per_patient_time,
        'schedule' => $data,];
    return $this->sendResponse($response, 'Doctor Scheduled
Retrieved Successfully');}

```

Додатково, у розділі "Перегляд рецептів ліків" пацієнт може знайти інформацію про назначені медикаменти та їхні рецепти, що сприяє правильному та своєчасному вживанню ліків.

Функція "Додавання особистої інформації" дозволяє пацієнту вносити та оновлювати свої особисті дані, щоб лікарі завжди мали доступ до актуальної інформації про пацієнта.

"Перегляд та друк рахунків" визначається можливістю пацієнта переглядати та надрукувати рахунки за надані медичні послуги, що спрощує процес оплати та ведення фінансового обліку.

Окрім того, пацієнт може скористатися "Можливістю проведення відеоконференції" для зручного зв'язку з лікарем без потреби особистого візиту, що особливо актуально в умовах віддаленої медичної консультації.

Інформація про лабораторні дослідження дозволяє пацієнту ознайомитися з результатами та іншою важливою інформацією, отриманою в результаті лабораторних досліджень.

Узагальнюючи, профіль кабінету пацієнта надає повний інструментарій для контролю над власним здоров'ям та взаємодії з медичним закладом у зручний та ефективний спосіб.

#### 3.2.4 Профіль кабінету медсестри

В системі передбачена можливість реєстрації користувача типу медсестра. Для цього типу користувача відведені наступні можливості:

- запис пацієнта на прийом до лікаря;
- перегляд медичної інформації пацієнта (Діагноз, лабораторні дослідження, рецепти ліків).
- редагування графіку роботи лікаря;
- виставлення рахунків на оплату.

#### 3.2.5 Профіль кабінету головного адміністратора

Головний адміністратор наділений найбільшими правами в системі. Він може бачити всю інформацію про стан системи. Панель управління головного адміністратора приведено на рис. 3.12.

Для головного адміністратора додані наступні функціональні можливості:

- реєстрація нових користувачів (пацієнт, лікар, медсестра, бухгалтер, та інші.);
- видалення або призупинення доступу для користувача;
- додавання мовної версії для системи;
- налаштування платіжних даних;
- включення або відключення функціональних модулів;
- додавання/видалення послуг;
- налаштування sms-шлюзів, підключення месенджерів для сповіщення користувачів, або груп користувачів.
- управління сайтом медичного закладу;
- налаштування типів звітів;
- управління медичними запасами.

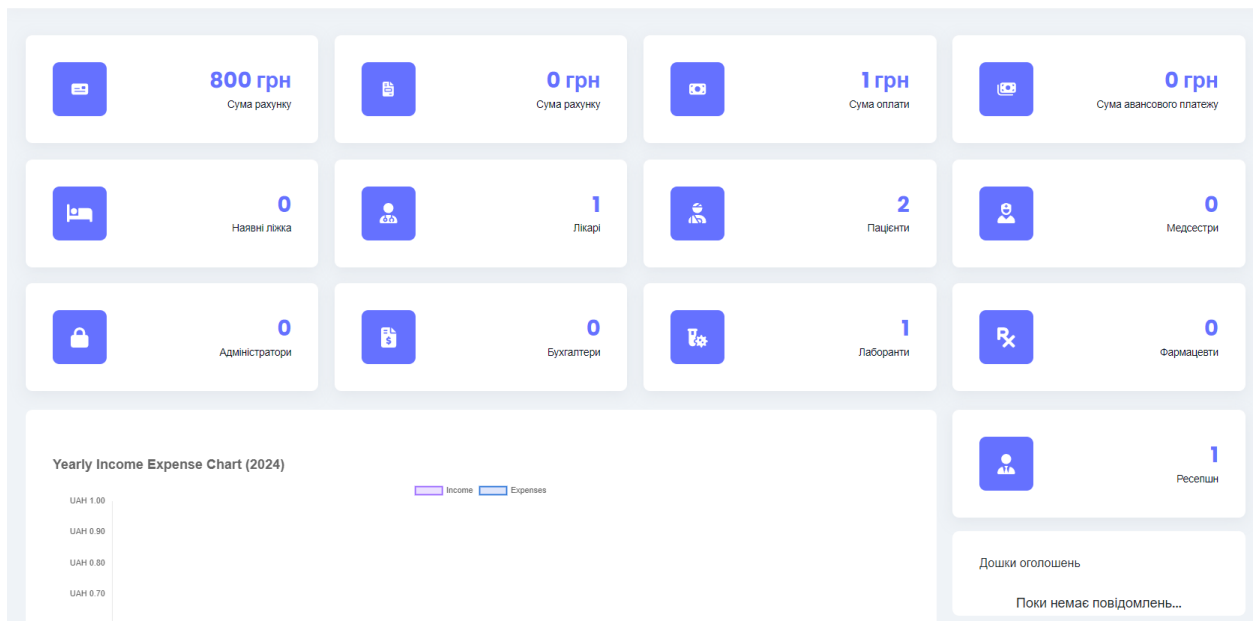


Рисунок 3.12 – Панель управління головного адміністратора

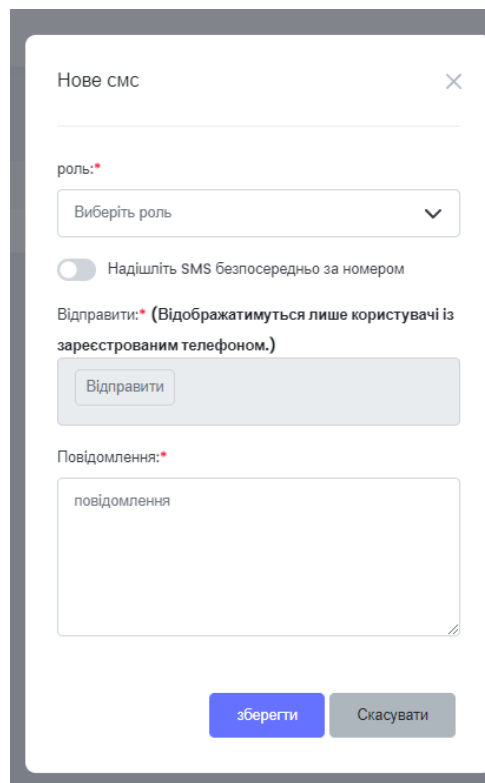
Перед першим запуском системи, в базі даних є запис про головного адміністратора. Це зроблене для того, щоб в нових власників системи

невиникла проблема з доступом на першим налаштуванням. Після проведення налаштування і створення нового адміністратора, іншого можна видалити.

Для адміністратора системи надана функціональна можливість, видаляти користувачів, або призупиняти дію їх особистого кабінету.

Враховуючи те, що не всі медичні заклади мають однакові види послуг, в системі передбачена можливість відключати не потрібний вид функціоналу. Для цього достатньо перейти на сторінку «Налаштування», та вибрати категорію «Управління модулів» і відключити не потрібний.

Для сповіщення користувачів через sms повідомлення в системі потрібно вказати API для sms-шлюзу. Після цього можна відправити повідомлення групі користувачів або конкретному користувачу (рис. 3.13). Код відправки sms повідомлень приведено у лістингу 3.11.



Нове смс

роль:\*

Виберіть роль

Надішліть SMS безпосередньо за номером

Відправити: \* (Відобразитимуться лише користувачі із зареєстрованим телефоном.)

Відправити

Повідомлення:\*

повідомлення

зберегти Скасувати

Рисунок 3.13 – Вікно для відправлення sms повідомлень з системи

## Лістинг 3.11 – Код відправки sms повідомлень

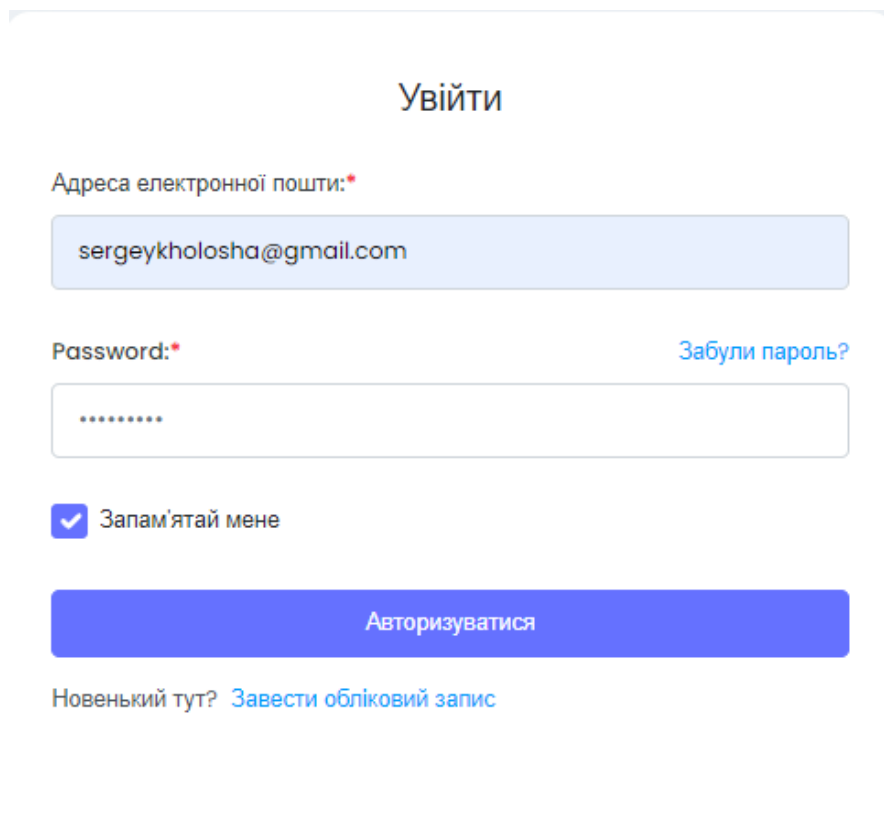
```
public function sendSMS($sendTo, $regionCode, $phone,
$message) {
    $sid = config('twilio.sid');
    $token = config('twilio.token');
    $client = new Client($sid, $token);
    try {
        $sms = Sms::create([
            'send_to' => $sendTo,
            'region_code' => $regionCode,
            'phone_number' => $phone,
            'message' => $message,
            'send_by' => Auth::id(),]);
        $client->messages->create(
            substr($phone, 0, 1) == '+' ? $phone : '+'.$sms->
            >region_code.$sms->phone_number,
            ['from' => config('twilio.from_number'),
            'body' => $message,] );
    } catch (Exception $e) {
        throw new UnprocessableEntityHttpException($e->
            >getMessage());} }}}
```

## 4 ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ

У даному розділі проводиться детальна розробка тестових кейсів для оцінки ефективності та надійності медичної інформаційної системи.

### 4.1 Розробка тестових кейсів

Першим і важливим етапом процесу перевірки є відпрацювання правильної процедури авторизації користувача, яка полягає у введенні вірних даних. Для успішної авторизації на сайті, користувачеві достатньо ввести той самий email та пароль, які були вказані при реєстрації (рис.4.1).



Увійти

Адреса електронної пошти:\*

sergeykhlosha@gmail.com

Password:\*

[Забули пароль?](#)

Запам'ятай мене

Авторизуватися

Новенький тут? [Завести обліковий запис](#)

Рисунок 4.1 – Введення правильних даних для авторизації

Після успішної авторизації, користувач переходить у свій особистий кабінет за адресою `site.com/patient/my-cases/`. Цей етап також успішно виконаний, оскільки користувач знаходиться на необхідній сторінці (рис. 4.2). На рисунку видно, що користувач має повний доступ до свого особистого кабінету, де він може переглядати свою історію та робити записи на прийом. Це відкриває перед ним широкі можливості в управлінні своїм медичним профілем.

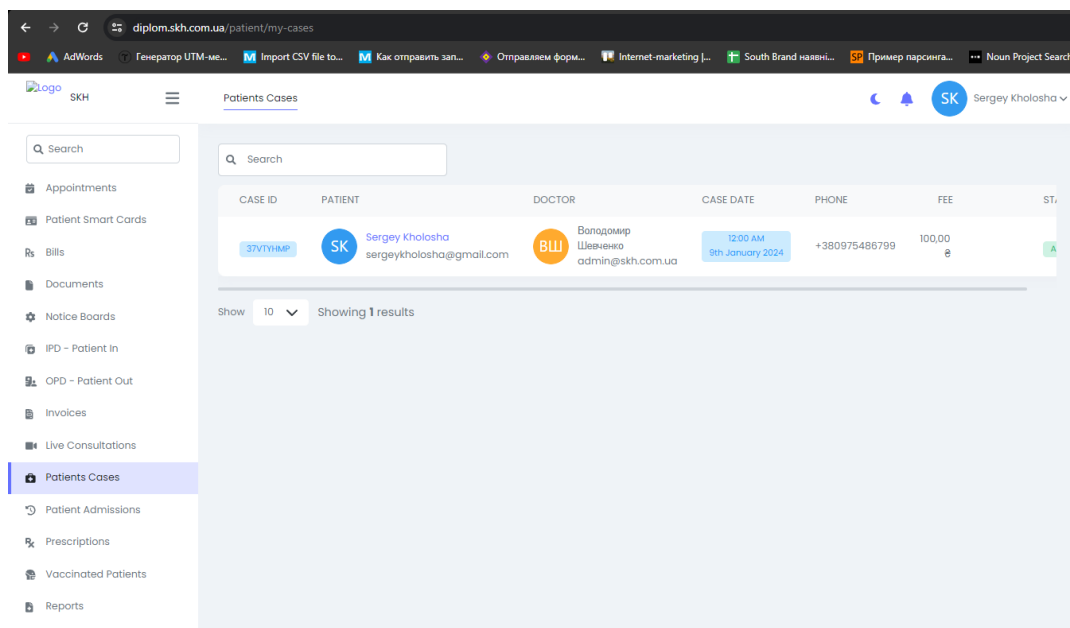


Рисунок 4.2 – Успішна авторизація

Для подальшої перевірки можливості запису на прийом, ми розглянемо процес створення нового запису. Після введення необхідної інформації, новий запис повинен автоматично відобразитись у розкладі лікаря, якого обрав користувач. Це підтверджує ефективність системи та гарантує користувачеві зручний і швидкий доступ до медичних послуг. Заповнена але не збережена форма запису приведена на рис. 4.3, а збережена форма запису – на рис. 4.4.

Рисунок 4.3 - Заповнена але не збережена форма запису

ПАЦІЄНТ	ЛІКАР	ЛІКАРСЬКЕ ВІДДІЛЕННЯ	ДАТА	ДІЯ
Сергій Халаша sergeykhloshah@gmail.com	Володимир Шевченко admin@skh.com.ua	кардіологія	17:00 24 січня 2024 р.	

Рисунок 4.4 – Збережена форма запису

Наступним кроком тестування є введення неправильних даних для авторизації. В правильному режимі форма повинна видати помилку та не пустити користувача до особистого кабінету (рис.4.5.).

Рисунок 4.5 – Помилка при введенні неправильних даних для авторизації

Як видно з рис. 4.5 система показала помилку при спробі авторизації і не пропустила користувача до особистого кабінету.

При спробі введення SQL – ін'єкції типу OR '1'=1'; -- форма також повертає помилку вводу. Це говорить про те, що система виявляє та блокує потенційно небезпечні конструкції SQL.

## ВИСНОВКИ

Кваліфікаційна робота присвячена розробці віртуальної системи управління медичним закладом.

Медична інформаційна система відповідає певним стандартам та вимогам, визначеним при проектуванні. Головні критерії, такі як доступність, гнучкість, безпека, модульність, масштабованість відповідність міжнародним стандартам, були враховані та ефективно реалізовані. Реєстрація та авторизація користувачів на web-сервісі забезпечують легкий та безпечний доступ до особистої медичної інформації. Безпечний обмін даними між пристроєм користувача та програмним забезпеченням медичної установи гарантує конфіденційність та цілісність медичних даних. Наявність REST API дозволяє легко інтегрувати систему з мобільними застосунками, роблячи її дуже зручною та доступною для користувачів з різних пристроїв, що розширює функціональні можливості системи та забезпечує її високу адаптивність до різних потреб користувачів.

У сучасних умовах, електронна система управління та реєстрації медичної інформації виявляється надзвичайно важливою та переважною. Зокрема, система має кілька важливих переваг, які забезпечують її ефективність та актуальність в умовах воєнного конфлікту та глобальної пандемії.

Електронна система дозволяє легко та швидко звертатися до особистої медичної інформації пацієнтів. Це важливо у випадках невідкладної потреби в лікуванні. В системі створена онлайн реєстрація та консультації, при цьому пацієнти можуть здійснювати реєстрацію та отримувати консультації від медичних фахівців онлайн, що робить доступ до медичної допомоги більш гнучким та ефективним.

Розроблена система гарантує захист конфіденційності медичних даних. Електронна система легко масштабується та адаптується до зростаючого обсягу користувачів та змін у потребах системи охорони здоров'я.

Розроблена медична інформаційна система є ефективним інструментом для забезпечення доступу, обміну та управління медичною інформацією, враховуючи вимоги до систем збереження та обробки медичних даних.

Отже, розроблена система не лише відповідає вимогам сучасних стандартів у сфері інформаційних технологій у сфері охорони здоров'я, але і ефективно відповідає викликам, що виникають в умовах війни та пандемій, забезпечуючи надійний та зручний доступ до медичної допомоги та інформації.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Metz C. A Breakthrough for A.I. Technology: Passing an 8th-Grade Science Test. / C. Metz // The New York Times. ISSN 0362-4331. [Електронний ресурс]. Режим доступу: <https://www.nytimes.com/2019/09/04/technology/artificial-intelligence-aristo-passed-test.html>.
2. Nance Jr., John W., Meenan, Christopher; Nagy, Paul G. Future of the Radiology Information System./Jr. Nance, W. John and others // American Journal of Roentgenology. – 2013. – 200 (5). pp. 1064-1070. doi:10.2214/AJR.12.10326. ISSN 0361-803X.
3. Joel Rodrigues. Health Information Systems: Concepts, Methodologies, Tools, and Applications, / Joel Rodrigues // Volume 1. IGI Global. – 2010.– pp. 1011-1021. ISBN 978-1-60566-988-5.
4. Типи інформаційних систем в галузі охорони здоров'я. Госпітальні інформаційні системи та їх розвиток. Індивідуальні медичні картки. Структуризація змісту електронних медичних карток (ЕМК) [Архівовано 11 вересня 2021 у Wayback Machine.]
5. ISO/HL7 27932:2008 HL7. Clinical Document Architecture, Release 2.– 2005.
6. Girosi F., Meili R., Scoville R. Extrapolating evidence of health information technology savings and costs. / F. Girosi, R. Meili, R. Scoville // Santa Monica, Calif.: RAND Corporation, – 2005. – 110 p.
7. Wong D., et al. Changes in intensive care unit nurse task activity after installation of a third generation intensive care unit information system / D. Wong, et al // Critical Care Medicine. – 2003. – Vol. 31. – 10. – pp. 2488–2494.
8. Ellingsen G., Monteiro E. Big is beautiful. Electronic patient records in Norway 1980–2000 / G. Ellingsen, E. Monteiro // Methods of Information in Medicine. – 2003. – Vol. 42.– pp. 1488–1494.

9. Fickel K. Hot-Wiring Hospitals / K. Fickel // Profit Magazine. – 2001. pp.483–489.
10. Холоша С. В. Медична інформаційна система. / С. В. Холоша // Матеріали тез 27-го міжнародного молодіжного форуму «Радіоелектроніка та молодь у ХХІ столітті» - м. Харків, 2023. - С. 61-62.
11. Obinna C., Madubuike, S.M., Chimay J. Anumba, F., Digital Twin-Based Health Care Facilities Management / C. Obinna, S.M., Madubuike, J. Chimay, F. Anumba, // Journal of Computing in Civil Engineering, 2022.– 37(2)/ - pp. 1-11, DOI:10.1061/JCCEE5.CPENG-4842.

## ДОДАТОК А Графічна частина кваліфікаційної роботи

Харківський національний університет радіоелектроніки


Кафедра АПОТ

# ВІРТУАЛЬНА СИСТЕМА УПРАВЛІННЯ МЕДИЧНИМ ЗАКЛАДОМ

Магістр групи СКСМ-22-2  
Холоша Сергій Вікторович

Керівник:  
доц. кафедри АПОТ  
Ларченко Л. В.

Харків 2024

 Харківський національний університет радіоелектроніки,  
кафедра АПОТ, тел. 7021 326, e-mail: ri@kture.kharkov.ua

## Актуальність

У світі, що стрімко переходить до цифрової ери, роль електронних систем віддаленого доступу набуває парадигмального значення. Нестримний розвиток технологій та зростаюча потреба в комфортному та безпечному отриманні послуг підштовхують до створення ефективних інструментів для доступу до різноманітних сервісів.

Пандемія послужила своєрідним викликом для медичної галузі, підкресливши необхідність негайного розвитку та впровадження медичних інформаційних систем. Обмеження фізичних контактів та необхідність швидкого доступу до медичної допомоги підкреслили важливість електронних платформ для консультацій та доступу до медичних послуг. У цьому контексті створення медичної інформаційної системи з особистим кабінетом користувача є не тільки актуальним, але й стратегічно важливим завданням. Такий особистий кабінет створює можливість надати кожному користувачу зручний, інтуїтивний та безпечний доступ до власної медичної інформації, забезпечуючи необхідну конфіденційність та зручність у взаємодії з медичними послугами в онлайн-режимі.



 Харківський національний університет радіоелектроніки,  
кафедра АПОТ, тел. 7021 326, e-mail: ri@kture.kharkov.ua

## Мета та постановка задачі

**Метою** кваліфікаційної роботи є розробка віртуальної системи управління медичним закладом з особистим кабінетом користувача.

Сформульовано наступні задачі кваліфікаційної роботи:

- аналіз вимог до створюваної системи;
- аналіз функціональних та нефункціональних вимог до особистого кабінету користувача;
- розробка архітектури для управління медичною системою;
- реалізація основних функцій медичної системи;
- зручний інтерфейс для комунікації з медичним персоналом;
- розробка особистого кабінету користувача;
- тестування та валідацію віртуальної медичної системи.

## Аналіз функціональних можливостей системи

- Система повинна мати можливість реєструвати нових пацієнтів та зберігати їх основну інформацію, таку як особисті дані, контактна інформація, медична історія тощо.
- Система має забезпечувати створення та зберігання електронних медичних записів пацієнтів. Це включає інформацію про медичні візити, діагнози, рецепти, результати обстежень та лабораторних досліджень.
- В особистому кабінеті користувач повинен бачити інформацію про розклад графік роботи, та вільний час спеціаліста. Найменування послуг які надає медичний заклад. Пацієнти повинні мати можливість записатися на прийом за допомогою системи, наприклад, через інтернет або мобільний застосунок.



## Вибір технологій для розробки

При проектуванні медичної інформаційної системи для медичної установи були обрані технології, які відповідають ефективній реалізації, що приведені нижче:

- доступність системи для всіх користувачів;
- гнучкість системи;
- модульність;
- масштабування.

Стандарти обміну інформацією

Health Level Seven (HL7) - це стандарт обміну інформацією в галузі охорони здоров'я, який використовується для електронного обміну медичними даними між різними інформаційними системами в охороні здоров'я.

HL7 також розробив новішу версію стандарту під назвою HL7 FHIR (Fast Healthcare Interoperability Resources).



Основні особливості FHIR

- Модульність
- Простота використання
- Розширюваність
- Безпека

## Алгоритм входу користувача в систему

**Крок 1:** Користувач вводить email та пароль;

система перевіряє, чи користувач вже зареєстрований в системі.

**Крок 2:** Перевірка Прав Доступу

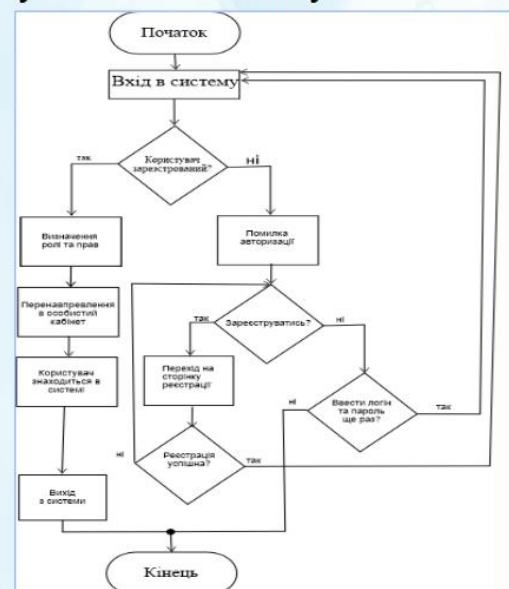
Якщо користувач вже є у нашій базі даних, система проводить аналіз його прав доступу для визначення, які опції та функції доступні.

**Крок 3:** Надання Доступу Згідно Прав

На основі визначених прав доступу, система автоматично надає користувачеві доступ до відповідних функцій та можливостей.

### Реєстрація або Корекція Пароля

Якщо введений email не зареєстрований, наша система активує процедуру реєстрації. У випадку помилкового введення пароля, користувачеві пропонується ввести його ще раз або скористатися опцією реєстрації.



## Функціональні блоки системи та їх взаємодія

Віртуальна система управління медичним закладом розроблена з врахуванням різноманітних функціональних блоків для того, щоб найкращим чином відповідати потребам користувачів.

- Авторизація та Аутентифікація
- Онлайн оплати
- Функція графіку роботи лікарів (відображення вільного часу)
- Управління медичними записами
- візуалізація медичних досліджень
- Інтеграція

## Опис баз даних

- Таблиця користувача (Users)
- Таблиця лікаря
- таблиця послуг медичного закладу;
- таблиця лабораторних досліджень;
- таблиця лікарських рецептів;
- таблиця керування вільними місцями;
- таблиця рахунків, оплати, та інші.

```

show_fields.blade.php | Zoom.php | zoom.php | ZoomRepository.php | Doctor.php | Appoint
97
98
99 CREATE TABLE `advanced_payments` (
100   `id` int UNSIGNED NOT NULL,
101   `patient_id` bigint UNSIGNED NOT NULL,
102   `receipt_no` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
103   `amount` double NOT NULL,
104   `date` date NOT NULL,
105   `created_at` timestamp NULL DEFAULT NULL,
106   `updated_at` timestamp NULL DEFAULT NULL,
107   `currency_symbol` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT NULL
108   ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
109
110
111
112
113 -- Table structure for table `ambulances`
114
115
116 CREATE TABLE `ambulances` (
117   `id` int UNSIGNED NOT NULL,
118   `vehicle_number` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL,
119   `vehicle_model` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
120   `year_made` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
121   `driver_name` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
122   `driver_license` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
123   `driver_contact` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
124   `notes` text COLLATE utf8mb4_unicode_ci,
125   `is_available` tinyint(1) NOT NULL DEFAULT '1',
126   `vehicle_type` int NOT NULL DEFAULT '1',
127   `created_at` timestamp NULL DEFAULT NULL,
128   `updated_at` timestamp NULL DEFAULT NULL
129   ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
130
131
132
133
134 -- Table structure for table `ambulance_calls`
135
136
137 CREATE TABLE `ambulance_calls` (
138   `id` int UNSIGNED NOT NULL,
139   `patient_id` int UNSIGNED NOT NULL,
140   `ambulance_id` int UNSIGNED NOT NULL,
141   `driver_name` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
142   `date` date NOT NULL,
143   `amount` double(8,2) NOT NULL,
144   `created_at` timestamp NULL DEFAULT NULL,
145   `updated_at` timestamp NULL DEFAULT NULL,
146   `currency_symbol` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT NULL
147   ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
148
149
150
151

```

## Безпека баз даних

### Реалізація системи контролю версій:

- Налаштований контроль версій для відстеження змін в базі даних.
- Аудит дій користувачів і реєстрацію змін в структурі бази даних.

### Реалізація системи контролю версій:

- Забезпечує високу доступність та відновлення після збоїв.
- Захист з'єднання між репліками для уникнення несанкціонованого доступу.

### Захист від SQL ін'єкцій:

- Використані параметризовані запити та підготовлені заяви для уникнення SQL ін'єкцій.
- Перевірку та валідацію введених даних перед їх використанням у запитах.

### Шифрування:

Використані методи шифрування для захисту конфіденційних даних в базі. Захист даних під час передачі через мережу (SSL/TLS).

## Реєстрація користувача. Ролі та права

- Головний адміністратор: має повний доступ до всіх функцій та ресурсів системи. Може керувати ролями та правами інших користувачів;
- Бухгалтер: має доступ до функцій, пов'язаних з фінансовим управлінням та обліком. Зможе переглядати та редагувати фінансові дані.
- Лікар: має можливість переглядати медичні записи пацієнтів. Здатен створювати та редагувати медичні записи.
- Медсестра: має обмежений доступ до медичних записів пацієнтів та може здійснювати деякі редагування в медичних записах за певними правилами.
- Лаборант: може додавати результати аналізів пацієнта.
- Фармацевт: керує запасами ліків.
- Пацієнт: має обмежений доступ до особистої інформації та медичних записів. Зможе заповнювати анкети та переглядати власні медичні дані.

Реєстрація

Ім'я*	Прізвище*
<input type="text"/>	<input type="text"/>
EMAIL*	Телефон*
<input type="text"/>	<input type="text" value="+380 50 123 4567"/>
Пароль*	Підтвердити пароль*
<input type="text"/>	<input type="text"/>
Стать*	
<input checked="" type="radio"/> Чоловіча <input type="radio"/> Жіноча	
<input type="button" value="Зареєструватися"/>	



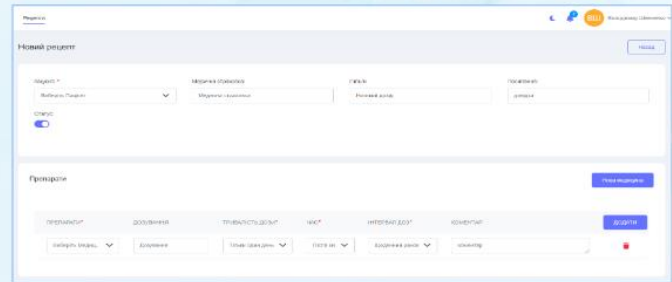
## Кабінет лікаря.2

На сторінці пацієнти, доступний список всіх пацієнтів які були на прийомі, або зробили запис.

При виборі пацієнта для лікаря відкривається наступна інформація:

- загальна інформація;
- інформація про лікування;
- статус рахунків;
- документи які надав пацієнт або завантажили інші працівники медичного закладу;
- інформація про щеплення.

Вигляд сторінки створення рецепту



## Кабінет медсестри

Кабінет медичної сестри був створений для допомоги пацієнтам та лікарям ведення медичної документації в системі. Медсестра має наступні права в системі:

- Запис пацієнта на прийом до лікаря;
- Перегляд медичної інформації пацієнта (діагноз, лабораторні дослідження, рецепти);
- Редагування графіку роботи лікаря;
- Формування рахунків на оплату;
- Реєстрація нових пацієнтів;
- Управління запасами медичних матеріалів;
- Відправка повідомлень пацієнту при зміні дати прийому.



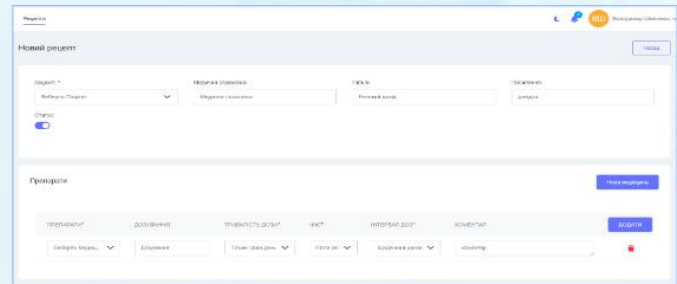
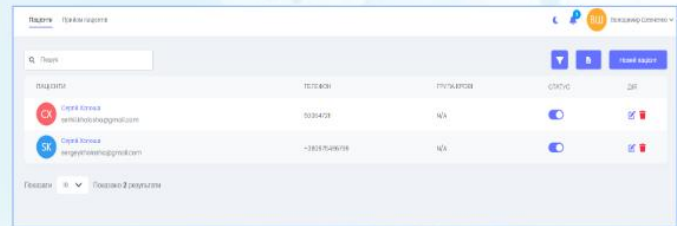
## Кабінет працівника лабораторії та діагностики

На сторінці пацієнти, доступний список всіх пацієнтів які були на прийомі, або зробили запис.

При виборі пацієнта для лікаря відкривається наступна інформація:

- загальна інформація;
- інформація про лікування;
- статус рахунків;
- документи які надав пацієнт або завантажили інші працівники медичного закладу;
- інформація про щеплення.

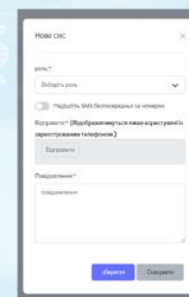
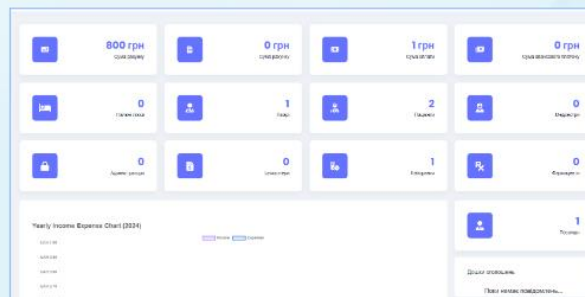
Вигляд сторінки створення рецепту



## Кабінет головного адміністратора

- реєстрація нових користувачів (пацієнт, лікар, медсестра, бухгалтер, та інші.);
- видалення або призупинення доступу для користувача;
- додавання мовної версії для системи;
- налаштування платіжних даних;
- включення або відключення функціональних модулів;

Панель управління головного адміністратора



## Кабінет пацієнта

- Створення нового запису до лікаря
- Перегляд рецептів ліків
- Перегляд та друк рахунків
- Проведення відеоконференції
- Інформація про лабораторні дослідження

Для зручності пацієнта реалізована можливість "Створення нового запису до лікаря". Це дозволяє користувачеві самостійно обирати лікаря, відділення, а також вказувати дату та час прийому, надаючи коротку інформацію про причину звернення. Пацієнт може також в будь-який момент відмінити запис на прийом, якщо це стане необхідним

## Тестування медичної системи

Важливим етапом процесу перевірки є відпрацювання правильної процедури авторизації користувача, яка полягає у введенні вірних даних.

Для успішної авторизації на сайті, користувачеві достатньо ввести той самий email та пароль, які були вказані при реєстрації.

Успішна авторизація



## Висновки

- Кваліфікаційна робота присвячена розробці віртуальної системи управління медичним закладом.
- Медична інформаційна система відповідає певним стандартам та вимогам, визначеним при проектуванні. Головні критерії, такі як доступність, гнучкість, безпека, модульність, масштабованість відповідність міжнародним стандартам, були враховані та ефективно реалізовані.
- Визначені ролі та права користувачів (пацієнт, лікар, лаборант та інші);
- Для кожної ролі користувача в системі створені кабінети відповідно до їх функціональних задач.
- Розроблена медична інформаційна система є ефективним інструментом для забезпечення доступу, обміну та управління медичною інформацією, враховуючи вимоги до систем збереження та обробки медичних даних.
- Отже, розроблена система не лише відповідає вимогам сучасних стандартів у сфері інформаційних технологій у сфері охорони здоров'я, але і ефективно відповідає викликам, що виникають в умовах війни та пандемій, забезпечуючи надійний та зручний доступ до медичної допомоги та інформації.



## ДОДАТОК Б Лістинг коду програм

## Лістинг Б.1 - код головної сторінки панелі управління

```

@extends('layouts.app')
@section('title')
    {{ __('messages.dashboard.dashboard') }}
@endsection
@section('page_css')
    {{-- <link rel="stylesheet" href="{{
asset('css/bootstrap-datetimepicker.css') }}"> --}}
    {{-- <link rel="stylesheet" href="{{
asset('assets/css/daterangepicker.css') }}"> --}}
@endsection
@section('css')
    {{-- <link rel="stylesheet" href="{{
asset('assets/css/detail-header.css') }}"> --}}
@endsection
@section('content')
    <div class="container-fluid">
        <div class="d-flex flex-column">
            <div class="row">
                <div class="col-12 mb-4">
                    <div class="row">
                        @if ($modules['Invoices'] == true)
                            {{-- Invoices Widget --}}
                            <div class="col-xxl-3 col-xl-4 col-
sm-6 widget">
                                <a class="text-decoration-none"
href="{{ route('invoices.index') }}">
                                    <div
                                        class="bg-white shadow-
md rounded-10 p-xxl-10 px-7 py-10 d-flex align-items-center
justify-content-between my-3">
                                            <div
                                                class="bg-primary
widget-icon rounded-10 d-flex align-items-center justify-
content-center">
                                                    <i class="fa-solid
fa-money-check fs-1 text-white"></i>
                                                </div>
                                            <div class="text-end">
                                                <h2 class="fs-1-xxl
fw-bolder text-primary">{{ getCurrencySymbol() }}
                                                    {{
formatCurrency($data['invoiceAmount']) }}</h2>
                                                <h3 class="mb-0 fs-5
fw-bold text-dark">

```

```

__ ('messages.dashboard.total_invoices') }}
    </h3>
  </div>
</div>
</a>
</div>
@endif
@if ($modules['Bills'])
  {{-- Bills Widget --}}
  <div class="col-xxl-3 col-xl-4 col-
sm-6 widget">
    <a href="{{ route('bills.index')
}}" class="text-decoration-none">
      <div
        class="bg-white shadow-
md rounded-10 p-xxl-10 px-7 py-10 d-flex align-items-center
justify-content-between my-3">
        <div
          class="bg-primary
widget-icon rounded-10 d-flex align-items-center justify-
content-center">
          <i class="fa-solid
fa-file-invoice fs-1 text-white"></i>
        </div>
        <div class="text-end">
          <h2 class="fs-1-xxl
fw-bolder text-primary">{{ getCurrencySymbol() }}
          {{
formatCurrency($data['billAmount']) }}</h2>
          <h3 class="mb-0 fs-5
fw-bold text-dark">
            {{
__ ('messages.dashboard.total_bills') }}</h3>
          </div>
        </div>
      </a>
    </div>
  @endif
  @if ($modules['Payments'] == true)
    {{-- Payments Widget --}}
    <div class="col-xxl-3 col-xl-4 col-
sm-6 widget">
      <a href="{{
route('payments.index') }}" class="text-decoration-none">
        <div
          class="bg-white shadow-
md rounded-10 p-xxl-10 px-7 py-10 d-flex align-items-center
justify-content-between my-3">
          <div
            class="bg-primary
widget-icon rounded-10 d-flex align-items-center justify-
content-center">

```

```

fa-money-bill fs-1 text-white"></i>
fa-money-bill fs-1 text-white"></i>
</div>
<div class="text-end">
  <h2 class="fs-1-xxl
fw-bolder text-primary">{{ getCurrencySymbol() }}
  {{
formatCurrency($data['paymentAmount']) }}</h2>
  <h3 class="mb-0 fs-5
fw-bold text-dark">
  {{
__('messages.dashboard.total_payments') }}</h3>
  </div>
</div>
</a>
</div>
@endif
@if ($modules['Advance Payments'] ==
true)
  {{-- Advance Payments Widget --}}
  <div class="col-xxl-3 col-xl-4 col-
sm-6 widget">
    <a href="{{ route('advanced-
payments.index') }}" class="text-decoration-none">
      <div
        class="bg-white shadow-
md rounded-10 p-xxl-10 px-7 py-10 d-flex align-items-center
justify-content-between my-3">
          <div
            class="bg-primary
widget-icon rounded-10 d-flex align-items-center justify-
content-center">
              <i class="fa-solid
fa-money-bills fs-1 text-white"></i>
            </div>
          <div class="text-end">
            <h2 class="fs-1-xxl
fw-bolder text-primary">{{ getCurrencySymbol() }}
            {{
formatCurrency($data['advancePaymentAmount']) }}</h2>
            <h3 class="mb-0 fs-5
fw-bold text-dark">
              {{
__('messages.dashboard.total_advance_payments') }}</h3>
            </div>
          </div>
        </a>
      </div>
    @endif
    @if ($modules['Beds'] == true)
      {{-- Available Beds Widget --}}
      <div class="col-xxl-3 col-xl-4 col-
sm-6 widget">

```

```

                                <a href="{{ route('beds.index')
}}}" class="text-decoration-none">
                                <div
                                    class="bg-white shadow-
md rounded-10 p-xxl-10 px-7 py-10 d-flex align-items-center
justify-content-between my-3">
                                    <div
                                        class="bg-primary
widget-icon rounded-10 d-flex align-items-center justify-
content-center">
                                        <i class="fa-solid
fa-bed fs-1-xl text-white"></i>
                                    </div>
                                    <div class="text-end">
                                        <h2 class="fs-1-xxl
fw-bolder text-primary">{{ $data['availableBeds'] }}</h2>
                                        <h3 class="mb-0 fs-5
fw-bold text-dark">
                                            {{
__ ('messages.dashboard.available_beds') }}</h3>
                                    </div>
                                </div>
                                </a>
                            </div>
                        @endif
                        @if ($modules['Doctors'] == true)
                            {{-- Doctors Widget --}}
                            <div class="col-xxl-3 col-xl-4 col-
sm-6 widget">
                                <a href="{{
route('doctors.index') }}" class="text-decoration-none">
                                    <div
                                        class="bg-white shadow-
md rounded-10 p-xxl-10 px-7 py-10 d-flex align-items-center
justify-content-between my-3">
                                        <div
                                            class="bg-primary
widget-icon rounded-10 d-flex align-items-center justify-
content-center">
                                                <i class="fa-solid
fa-user-doctor fs-1-xl text-white"></i>
                                            </div>
                                        <div class="text-end">
                                            <h2 class="fs-1-xxl
fw-bolder text-primary">{{ $data['doctors'] }}</h2>
                                            <h3 class="mb-0 fs-5
fw-bold text-dark">{{ __ ('messages.dashboard.doctors') }}
                                                </h3>
                                        </div>
                                    </div>
                                </div>
                            </a>
                        </div>
                    @endif

```

```

@if ($modules['Patients'] == true)
    {{-- Patients Widget --}}
    <div class="col-xxl-3 col-xl-4 col-
sm-6 widget">
        <a href="{{
route('patients.index') }}" class="text-decoration-none">
            <div
                class="bg-white shadow-
md rounded-10 p-xxl-10 px-7 py-10 d-flex align-items-center
justify-content-between my-3">
                <div
                    class="bg-primary
widget-icon rounded-10 d-flex align-items-center justify-
content-center">
                    <i class="fa-solid
fa-user-injured fs-1-xl text-white"></i>
                </div>
                <div class="text-end">
                    <h2 class="fs-1-xxl
fw-bolder text-primary">{{ $data['patients'] }}</h2>
                    <h3 class="mb-0 fs-5
fw-bold text-dark">{{ __('messages.dashboard.patients') }}
                    </h3>
                </div>
            </div>
        </a>
    </div>
@endif
@if ($modules['Nurses'] == true)
    {{-- Nurses Widget --}}
    <div class="col-xxl-3 col-xl-4 col-
sm-6 widget">
        <a href="{{
route('nurses.index') }}" class="text-decoration-none">
            <div
                class="bg-white shadow-
md rounded-10 p-xxl-10 px-7 py-10 d-flex align-items-center
justify-content-between my-3">
                <div
                    class="bg-primary
widget-icon rounded-10 d-flex align-items-center justify-
content-center">
                    <i class="fa-solid
fa-user-nurse fs-1-xl text-white"></i>
                </div>
                <div class="text-end">
                    <h2 class="fs-1-xxl
fw-bolder text-primary">{{ $data['nurses'] }}</h2>
                    <h3 class="mb-0 fs-5
fw-bold text-dark">{{ __('messages.nurses') }}</h3>
                </div>
            </div>
        </a>
    </div>

```

```

        </div>
    @endif
    @if ($modules['Admin'] == true)
        {{-- Admins Widget --}}
        <div class="col-xxl-3 col-xl-4 col-
sm-6 widget">
            <a href="{{
route('admins.index') }}" class="text-decoration-none">
                <div
                    class="bg-white shadow-
md rounded-10 p-xxl-10 px-7 py-10 d-flex align-items-center
justify-content-between my-3">
                    <div
                        class="bg-primary
widget-icon rounded-10 d-flex align-items-center justify-
content-center">
                            <i class="fa-solid
fa-lock fs-1-xl text-white"></i>
                        </div>
                    <div class="text-end">
                        <h2 class="fs-1-xxl
fw-bolder text-primary">{{ $data['admins'] }}</h2>
                        <h3 class="mb-0 fs-5
fw-bold text-dark">{{ __('messages.admin') }}</h3>
                    </div>
                </div>
            </a>
        </div>
    @endif
    @if ($modules['Accountants'] == true)
        {{-- Accountant Widget --}}
        <div class="col-xxl-3 col-xl-4 col-
sm-6 widget">
            <a href="{{
route('accountants.index') }}" class="text-decoration-none">
                <div
                    class="bg-white shadow-
md rounded-10 p-xxl-10 px-7 py-10 d-flex align-items-center
justify-content-between my-3">
                    <div
                        class="bg-primary
widget-icon rounded-10 d-flex align-items-center justify-
content-center">
                            <i class="fa-solid
fa-file-invoice-dollar fs-1-xl text-white"></i>
                        </div>
                    <div class="text-end">
                        <h2 class="fs-1-xxl
fw-bolder text-primary">{{ $data['accountants'] }}</h2>
                        <h3 class="mb-0 fs-5
fw-bold text-dark">{{ __('messages.accountants') }}</h3>
                    </div>
                </div>
            </a>
        </div>
    @endif

```

```

        </a>
      </div>
    @endif
    @if ($modules['Lab Technicians'] ==
true)
        {{-- Lab Technician Widget --}}
        <div class="col-xxl-3 col-xl-4 col-
sm-6 widget">
            <a href="{{ route('lab-
technicians.index') }}" class="text-decoration-none">
                <div
                    class="bg-white shadow-
md rounded-10 p-xxl-10 px-7 py-10 d-flex align-items-center
justify-content-between my-3">
                        <div
                            class="bg-primary
widget-icon rounded-10 d-flex align-items-center justify-
content-center">
                                <i class="fa-solid
fa-vial-virus fs-1-xl text-white"></i>
                            </div>
                            <div class="text-end">
                                <h2 class="fs-1-xxl
fw-bolder text-primary">{{ $data['labTechnicians'] }}</h2>
                                <h3 class="mb-0 fs-5
fw-bold text-dark">{{ __('messages.lab_technicians') }}
                                </h3>
                            </div>
                        </div>
                    </a>
                </div>
            @endif
            @if ($modules['Pharmacists'] == true)
                {{-- Pharmacists Widget --}}
                <div class="col-xxl-3 col-xl-4 col-
sm-6 widget">
                    <a href="{{
route('pharmacists.index') }}" class="text-decoration-none">
                        <div
                            class="bg-white shadow-
md rounded-10 p-xxl-10 px-7 py-10 d-flex align-items-center
justify-content-between my-3">
                                <div
                                    class="bg-primary
widget-icon rounded-10 d-flex align-items-center justify-
content-center">
                                        <i class="fa-solid
fa-prescription fs-1-xl text-white"></i>
                                    </div>
                                    <div class="text-end">
                                        <h2 class="fs-1-xxl
fw-bolder text-primary">{{ $data['pharmacists'] }}</h2>

```



```

        {{
__ ('messages.dashboard.notice_boards') }}
        </h3>
    </div>
    <div class="card-body pt-7 pb-2">
        @if (count($data['noticeBoards']) >
0)
            <table class="table table-
striped">
                <thead>
                    <tr>
                        <th scope="col">{{
__ ('messages.dashboard.title') }}</th>
                        {{-- <th scope="col"
class="text-center">
                            {{
__ ('messages.common.created_on') }}
                            </th> --}}
                    </tr>
                </thead>
                <tbody class="text-gray-600
fw-bold">
                    @foreach
($data['noticeBoards'] as $noticeBoard)
                        <tr>
                            <td>
                                <a
href="javascript:void(0)" data-id="{{ $noticeBoard->id }}"
class="text-decoration-none notice-board-view-btn">{{
Str::limit($noticeBoard->title, 24, '...') }}</a>
                            </td>
                            {{-- <td
class="text-center">
                                <span
class="badge bg-light-info">
                                    {{
\Carbon\Carbon::parse($noticeBoard->created_at)-
>translatedFormat('jS M, Y') }}
                                </span>
                            </td> --}}
                        </tr>
                    @endforeach
                </tbody>
            </table>
        @else
            <h2 class="mb-0 text-center fs-
2">{{ __ ('messages.dashboard.no_notice_yet') }}... </h2>
        @endif
    </div>
</div>
</div>
</div>

```

```

<div class="row my-2">
  <div class="col-xxl-5 col-12 mb-7 mb-xxl-0">
    <div class="card overflow-auto">
      <div class="card-header pb-0 px-10">
        <h3 class="mb-0">
          {{ __('messages.enquiries') }}
        </h3>
      </div>
      <div class="card-body pt-7">
        @if (count($data['enquiries']) > 0)
          <table class="table table-
striped">
                                <thead>
                                  <tr>
                                    <th scope="col">{{
__('messages.enquiry.name') }}</th>
                                    <th scope="col">{{
__('messages.enquiry.email') }}</th>
                                    <th scope="col"
class="text-center text-muted">
                                        {{
__('messages.common.created_on') }}</th>
                                  </tr>
                                </thead>
                                <tbody class="text-gray-600
fw-bold">
                                  @foreach
($data['enquiries'] as $enquiry)
                                    <tr>
                                      <td>
                                        <a href="{{
route('enquiry.show', $enquiry->id) }}"
class="text-primary-800 text-decoration-none mb-1 fs-6">{{
Str::limit($enquiry->full_name, 10, '...') }}</a>
                                      </td>
                                      <td class="text-
start">
                                        <span
class="text-muted fw-bold d-block">{{ $enquiry->email }}</span>
                                      </td>
                                      <td class="text-
center text-muted fw-bold">
                                        <span
class="badge bg-light-info">
                                          {{
\Carbon\Carbon::parse($enquiry->created_at)-
>translatedFormat('jS M, Y') }}
                                        </span>
                                      </td>
                                    </tr>
                                  @endforeach

```

```

                </tbody>
            </table>
            @else
                <h4 class="mb-0 text-center fs-
2">{{ __('messages.dashboard.no_enquiries_yet') }}</h4>
            @endif
        </div>
    </div>
</div>

    {{-- Appointment table --}}
    <div class="col-xxl-7 col-12 mb-7 mb-xxl-0">
        <div class="card overflow-auto">
            <div class="card-header pb-0 px-10">
                <h3 class="mb-0">
                    {{ __('messages.appointments')
}}
                </h3>
            </div>
            <div class="card-body pt-7">
                <livewire:dashboard-appointment-
table>
            </div>
        </div>
    </div>
    {{-- end Appointment table --}}
</div>
    {{--
        Income & Expense Chart --}}
    {{--
        <div class="row"> --}}
    {{--
        <div class="col-lg-
12"> --}}
        {{--
        <div
class="card"> --}}
        {{--
        <div
class="card-body"> --}}
        {{--
        <div
class="row justify-content-between"> --}}
        {{--
        <div
class="col-sm-6 col-md-6 col-lg-6 pt-2"> --}}
        {{--
        <h5>{{ __('messages.dashboard.income_and_expense_report')
}}</h5> --}}
        {{--
        </div> --}}
        {{--
        <div
class="col-md-3"> --}}
        {{--
        <div id="time_range" class="time_range d-flex"> --}}
        {{--
        <i class="far fa-calendar-alt" --}}
        {{--
        aria-hidden="true"></i>&nbsp;&nbsp; <span></span> --}}

```

```

        {{--
<b class="caret"></b> --}}
        {{--
</div> --}}
        {{--
</div> --}}
        {{--
-}}
        {{--
class="table-responsive-sm"> --}}
        {{--
class="pt-2"> --}}
        {{--
<canvas id="daily-work-report" class="mh-400px"></canvas> --}}
        {{--
</div> --}}
        {{--
-}}
        {{--
        {{--
        {{--
        {{--
        {{ Form::hidden('incomeExpenseReportUrl',
route('income-expense-report'), ['id' =>
'dashboardIncomeExpenseReportUrl', 'class' =>
'incomeExpenseReportUrl']) }}
        {{ Form::hidden('currentCurrencyName',
getCurrencySymbol(), ['id' => 'dashboardCurrentCurrencyName',
'class' => 'currentCurrencyName']) }}
        {{--
{{Form::hidden('currencies',json_encode($data['currency']),['id'
=>'createBillDate','class'=>'currencies'])}} --}}
        {{ Form::hidden('income_and_expense_reports',
__('messages.dashboard.income_and_expense_reports'), ['id' =>
'dashboardIncome_and_expense_reports', 'class' =>
'income_and_expense_reports']) }}
        {{ Form::hidden('defaultAvatarImageUrl',
asset('assets/img/avatar.png'), ['id' =>
'dashboardDefaultAvatarImageUrl', 'class' =>
'defaultAvatarImageUrl']) }}
        {{ Form::hidden('noticeBoardUrl', url('notice-
boards'), ['id' => 'dashboardNoticeBoardUrl', 'class' =>
'noticeBoardUrl']) }}
        {{ Form::hidden('dashboardChart',
route('dashboard.chart'), ['id' => 'dashboardChart', 'class' =>
'dashboardChart']) }}

</div>
@include('employees.notice_boards.show_modal')
</div>
@endsection

```

```

{{--    <script
src="{{mix('assets/js/dashboard/dashboard.js')}}"></script> --}}
{{--    <script
src="{{mix('assets/js/custom/input_price_format.js')}}"></script
> --}}

```

## Лістинг Б.2 - код сторінки реєстрації

```

<?php
namespace App\Http\Controllers\Auth;
use App\Http\Controllers\Controller;
use App\Models\Department;
use App\Models\Patient;
use App\Models\User;
use App\Providers\RouteServiceProvider;
use Illuminate\Foundation\Auth\RegistersUsers;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use Illuminate\Support\Facades\App;
class RegisterController extends Controller
{
    use RegistersUsers;
    protected $redirectTo = RouteServiceProvider::HOME;
    public function __construct()
    {
        $this->middleware('guest');
    }

    protected function validator($data)
    {
        return Validator::make($data, [
            'first_name' => ['required', 'string',
'max:255'],
            'email' => ['required', 'string',
'email:filter', 'max:255', 'unique:users'],
            'phone' => ['required'],
            'password' => ['required', 'string', 'min:6',
'confirmed'],
        ], [
            'password.min' =>
__('messages.user.validate_min_character_password'),
        ]);
    }

    protected function create($data)
    {
        $data['password'] = Hash::make($data['password']);
        $data['department_id'] =
Department::whereName('Patient')->first()->id;
        $data['status'] = 1;
        $data['language'] = checkLanguageSession();
    }
}

```





```
App::setLocale(checkLanguageSession());

$user = User::create($data);
$patient = Patient::create(['user_id' => $user-
>id]);

    $user->update(['owner_id' => $patient->id,
'owner_type' => Patient::class]);
    $user->assignRole($data['department_id']);
    return $user;
    }
}
```

## Відомість кваліфікаційної роботи

«Віртуальна система управління медичним закладом»

	Прізвище та ініціали відповідальної особи	Підпис	Дата
Роботу виконав студент групи СКСм-22-2 Структура кваліфікаційної роботи: – пояснювальна записка <u>61</u> с.; – графічний матеріал <u>19</u> арк..	Холоша С. В.		10.01.24
Керівник роботи	Ларченко Л.В.		10.01.24
Перевірка на плагіат здійснена. Оригінальність авторського тексту складає <u>99</u> %	Литвинова Є.І.		18.01.24
Нормоконтроль проведено :	Ларченко Л.В.		22.01.24