

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Веб-технології»)

на тему:

«Вебплатформа управління ветеринарною клінікою»

Виконала студентка групи ВТ-22-1
ВОЗНЮК Владислава Юріївна

Керівник роботи:
НЕРОДА Сергій Іванович

Рецензент:
ГРОМСЬКИЙ Олександр Олександрович

Житомир – 2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна Вакалюк
«28» лютого 2026 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ВОЗНЮК Владислава Юріївна**

Керівник роботи: **НЕРОДА Сергій Іванович**

Тема роботи: **«Вебплатформа управління ветеринарною клінікою»**,
затверджена Наказом закладу вищої освіти від **«27» лютого 2026 р., №92/с**
Вихідні дані для роботи: **об'єктом дослідження є інформаційні технології управління процесами діяльності ветеринарної клініки. Предметом дослідження є використання веборієнтованих технологій для розробки системи управління процесами обліку, планування та обслуговування клієнтів у ветеринарній клініці.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Нерода С.І.	01.03.2026р.	16.04.2026р.
2	Нерода С.І.	01.04.2026р.	16.05.2026р.
3	Нерода С.І.	01.05.2026р.	16.05.2026р.

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з вебплатформи для управління ветеринарною клінікою та пояснювальної записки. Пояснювальна записка до випускної кваліфікаційної роботи містить 77 сторінок, 34 ілюстрації, 36 таблиць, 10 сторінок додатків.

Метою роботи є розробка вебплатформи для автоматизації основних процесів діяльності ветеринарної клініки, зокрема управління клієнтами, обліку тварин, ведення медичних карток, планування прийомів та організації роботи персоналу.

Розроблена система реалізована на основі клієнт-серверної архітектури із використанням сучасних вебтехнологій. Користувацький інтерфейс створено за допомогою фреймворку Vue.js, серверну частину реалізовано на платформі Node.js із використанням Express, а для зберігання даних використовується система керування базами даних PostgreSQL. Розмежування прав доступу забезпечується шляхом використання ролей адміністратора, ветеринарного лікаря та клієнта.

Вебплатформа надає можливість вести облік клієнтів і тварин, управляти записами на прийом, формувати та редагувати медичні картки, контролювати розклад лікарів, отримувати системні сповіщення та переглядати історію лікування. Реалізований інтерфейс забезпечує зручну взаємодію користувачів із системою та дозволяє ефективно автоматизувати роботу ветеринарної клініки.

КЛЮЧОВІ СЛОВА: ВЕБПЛАТФОРМА, ВЕТЕРИНАРНА КЛІНІКА, АВТОМАТИЗАЦІЯ, МЕДИЧНА КАРТКА, ЗАПИС НА ПРИЙОМ.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ABSTRACT

The bachelor's qualification thesis consists of a web platform for veterinary clinic management and an explanatory report. The explanatory report comprises 77 pages, 34 figures, 36 tables and 10 pages of appendices.

The purpose of the thesis is to develop a web platform for automating the core processes of a veterinary clinic, including client management, pet record management, maintenance of electronic medical records, appointment scheduling, and staff activity management.

The developed system is based on a client-server architecture and implemented using modern web technologies. The user interface is developed with the Vue.js framework, while the server-side component is implemented on the Node.js platform using the Express framework. PostgreSQL is employed as the database management system for data storage. Access control is provided through a role-based model that distinguishes between administrator, veterinarian, and client roles.

The web platform enables the management of clients and pets, appointment scheduling, creation and modification of medical records, supervision of veterinarians' schedules, receipt of system notifications, and access to treatment histories. The implemented user interface ensures convenient interaction with the system and facilitates efficient automation of veterinary clinic operations.

KEYWORDS: WEB PLATFORM, VETERINARY CLINIC, AUTOMATION, MEDICAL RECORD, APPOINTMENT SCHEDULING.

		Вознюк В.Ю			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Нерода С.І				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ	4
ABSTRACT	5
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	8
ВСТУП	9
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ.....	12
1.1 Постановка задачі	12
1.2 Аналіз аналогів програмного продукту	14
1.3 Архітектура та узагальнена структура програмного проєкту	18
1.4 Інструментальні засоби реалізації програмного продукту та вимоги до апаратного забезпечення	21
1.5 Варіанти використання системи.....	24
Висновки до першого розділу.	28
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБПЛАТФОРМИ УПРАВЛІННЯ ВЕТЕРИНАРНОЮ КЛІНІКОЮ	30
2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи	30
2.2 Розробка бази даних системи.....	42
2.3 Проєктування та реалізація алгоритмів роботи системи	49
2.4 Реалізація функціоналу системи вебплатформи управління ветеринарною клінікою	52
Висновки до другого розділу.....	56
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБПЛАТФОРМОЮ УПРАВЛІННЯ ВЕТЕРИНАРНОЮ КЛІНІКОЮ	60
3.1 Порядок встановлення та налаштування параметрів вебплатформи ..	60
3.2 Структура інтерфейсу вебплатформи управління ветеринарною клінікою.....	63
3.3 Інтерфейс для адміністратора вебплатформи	71
3.4 Тестування вебплатформи	79

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		6

Висновки до третього розділу	84
ВИСНОВКИ.....	86
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	88
ДОДАТКИ.....	91

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		7

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

ПЗ – програмне забезпечення

СУБД – система управління базами даних

API – програмний інтерфейс застосунку

HTTP – протокол передачі гіпертексту

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. Активна цифровізація сфери послуг зумовлює необхідність впровадження інформаційних систем для автоматизації управлінських процесів. У діяльності ветеринарних клінік актуальними проблемами є неефективне планування прийомів, відсутність єдиного інформаційного середовища, складність ведення обліку пацієнтів і клієнтів, а також високий ризик помилок при ручній обробці даних.

На практиці діяльність багатьох ветеринарних клінік супроводжується рядом організаційних труднощів. До них належать складність планування робочого часу лікарів, необхідність постійного контролю записів на прийом, ведення історії лікування тварин, облік клієнтської бази та оперативне отримання необхідної інформації персоналом. Відсутність єдиної централізованої системи управління часто ускладнює взаємодію між працівниками клініки, збільшує час на виконання адміністративних операцій та негативно впливає на якість обслуговування клієнтів.

Використання сучасних вебплатформ дозволяє вирішити зазначені проблеми шляхом автоматизації основних бізнес-процесів ветеринарної клініки. Важливим аспектом автоматизації є підвищення якості обслуговування клієнтів: наявність електронної системи запису, доступ до інформації про власних тварин та перегляд історії лікування створюють додаткові зручності для власників тварин, а лікарі отримують швидкий доступ до медичної інформації, що сприяє ефективнішому прийняттю рішень.

Таким чином, розробка вебплатформи для управління ветеринарною клінікою є актуальним завданням, що відповідає сучасним тенденціям розвитку інформаційних технологій та потребам автоматизації діяльності ветеринарних закладів.

Метою дослідження є розробка вебплатформи, що дозволить користувачам здійснювати оперативне управління всіма ключовими бізнес-процесами в єдиному онлайн-середовищі, підвищуючи якість обслуговування та внутрішню ефективність клініки.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Для реалізації даної мети необхідно виконати ряд завдань. Насамперед потрібно проаналізувати предметну область та виявити основні бізнес-процеси ветеринарної клініки, що потребують автоматизації, а також здійснити огляд і порівняльний аналіз існуючих програмних аналогів з метою визначення їх переваг і недоліків. На основі отриманих результатів необхідно сформулювати функціональні та нефункціональні вимоги до вебплатформи і спроектувати архітектуру системи та структуру бази даних, що забезпечуватиме зберігання і обробку клінічних та адміністративних даних. Наступним кроком є розробка модульної структури вебплатформи з урахуванням рольової моделі доступу для адміністратора, лікаря та власника тварини, після чого слід реалізувати основні функціональні модулі системи: управління записами на прийом, ведення медичних карток пацієнтів, облік клієнтської бази та формування звітності. Завершальним етапом є проведення функціонального та інтеграційного тестування розробленого програмного забезпечення для перевірки коректності роботи реалізованих модулів.

Об'єктом дослідження є інформаційні технології управління процесами діяльності ветеринарної клініки.

Предметом дослідження є використання веборієнтованих технологій для розробки системи управління процесами обліку, планування та обслуговування клієнтів у ветеринарній клініці.

Методи дослідження. У процесі виконання роботи використано такі методи: аналіз та порівняння існуючих програмних аналогів для визначення їх переваг і недоліків; системний аналіз предметної області для виявлення основних бізнес-процесів ветеринарної клініки; метод структурного проєктування для розробки архітектури системи та бази даних; метод об'єктно-орієнтованого проєктування для побудови модульної структури програмного комплексу; методи функціонального та інтеграційного тестування для перевірки коректності роботи реалізованих модулів.

Інформаційна база дослідження включає наукові та технічні джерела з питань проєктування інформаційних систем і веброзробки, офіційну

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		10

документацію використаних технологій і фреймворків, матеріали щодо існуючих програмних рішень для управління ветеринарними клініками, а також стандарти проєктування програмного забезпечення.

Практичне значення одержаних результатів. Розроблена вебплатформа може бути безпосередньо впроваджена у діяльність ветеринарних клінік малого та середнього масштабу. Система автоматизує основні адміністративні та клінічні процеси, зменшує навантаження на персонал і мінімізує ймовірність помилок при обробці інформації. Використання хмарної інфраструктури забезпечує доступність платформи без необхідності придбання власного серверного обладнання, що знижує вартість впровадження. Результати роботи також можуть бути використані як основа для подальшого розширення системи або адаптації під потреби суміжних сфер – наприклад, приватних медичних кабінетів чи сервісних центрів.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		11

РОЗДІЛ 1 АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ

1.1 Постановка задачі

Основним призначенням вебплатформи для управління ветеринарною клінікою є надання комплексу інформаційних, транзакційних та аналітичних послуг з метою забезпечення централізованого, ефективного та безпечного управління всіма клінічними та адміністративними процесами. Використання сучасних вебтехнологій дозволяє автоматизувати значну частину операційної діяльності ветеринарного закладу, зменшити кількість ручних операцій, мінімізувати ризик виникнення помилок під час обробки інформації та підвищити якість обслуговування клієнтів. Перед початком моделювання та аналізу необхідно ознайомитися з основним поняттям платформи та особливостями її функціонування.

Вебплатформа управління ветеринарною клінікою – це спеціалізований багатокористувацький вебзастосунок, у межах якого здійснюється повна автоматизація внутрішньої діяльності клініки. Уся діяльність – від запису на прийом, ведення історії хвороби тварини, фіксації діагнозів та процедур до обліку фінансів і планування графіків – відбувається в межах цього інтегрованого інтернет-середовища. Такий тип платформи значно спрощує взаємодію між персоналом та забезпечує безперервний доступ до медичних даних, відповідно до ролей користувачів, незалежно від часу чи місця перебування.

Використання вебплатформи дозволяє відмовитися від паперової документації або розрізнених програмних засобів, які часто використовуються для ведення обліку в невеликих ветеринарних закладах. Централізація даних забезпечує швидкий доступ до інформації, спрощує процес її пошуку та оновлення, а також сприяє підвищенню ефективності роботи персоналу. Крім того, всі зміни, внесені до системи, автоматично зберігаються в базі даних і стають доступними користувачам відповідно до їхніх ролей та рівнів доступу.

Вебплатформа у цьому контексті виступає як інтерактивна професійна платформа, яка об'єднує різноманітні сервіси, ключові для ветеринарної

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		12

практики. Кожен з таких сервісів виконує визначену функцію та вирішує конкретні потреби як персоналу клініки, так і її клієнтів.

Інформаційна функція платформи забезпечує надання структурованих даних про пацієнтів, медичні протоколи, історію вакцинацій, результати лабораторних досліджень та довідники діагнозів. Транзакційна функція охоплює здійснення запису на прийом, фіксацію виконаних процедур та управління складськими запасами медикаментів. Комунікаційна функція реалізується через інтеграцію внутрішніх повідомлень між лікарями, автоматичне сповіщення клієнтів та облік їхніх зауважень і побажань. Аналітична функція забезпечує формування звітів і графіків, використання фільтрів та пошуку для швидкого доступу до даних і моніторингу ключових показників роботи клініки.

Система має охоплювати три основні ролі користувачів: адміністратор, лікар та власник тварини. Кожна роль передбачає різний рівень доступу до функцій та даних системи. Адміністратор здійснює загальне управління платформою: керує обліковими записами користувачів, налаштовує розклад роботи клініки, формує фінансові звіти та контролює стан складських запасів. Лікар отримує доступ до медичних карток пацієнтів, може фіксувати результати оглядів, призначати лікування та переглядати власний розклад прийомів. Власник тварини має можливість реєструватися в системі, записуватися на прийом, переглядати інформацію про своїх тварин та відстежувати історію їх лікування.

Моделювання та аналіз вебплатформи передбачають кілька послідовних етапів. На першому етапі проводиться аналіз потреб користувачів – детальне вивчення вимог різних ролей (лікаря, адміністратора та власника тварини) для визначення їхніх специфічних очікувань від системи. На основі цього аналізу визначається цільова аудиторія платформи та формуються вимоги до інтерфейсу, безпеки й надійності збереження медичних даних. Завершальним етапом є розробка технічного завдання, яке документує детальні вимоги до функціональності, структури даних, безпеки та швидкодії системи.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		13

1.2 Аналіз аналогів програмного продукту

Розробка вебплатформи управління ветеринарною клінікою потребує аналізу конкурентів на ринку.

Одним із поширених рішень є система Provet Cloud – універсальне cloud-рішення для ветеринарних практик, інтерфейс якої наведено на рисунку 1.1.

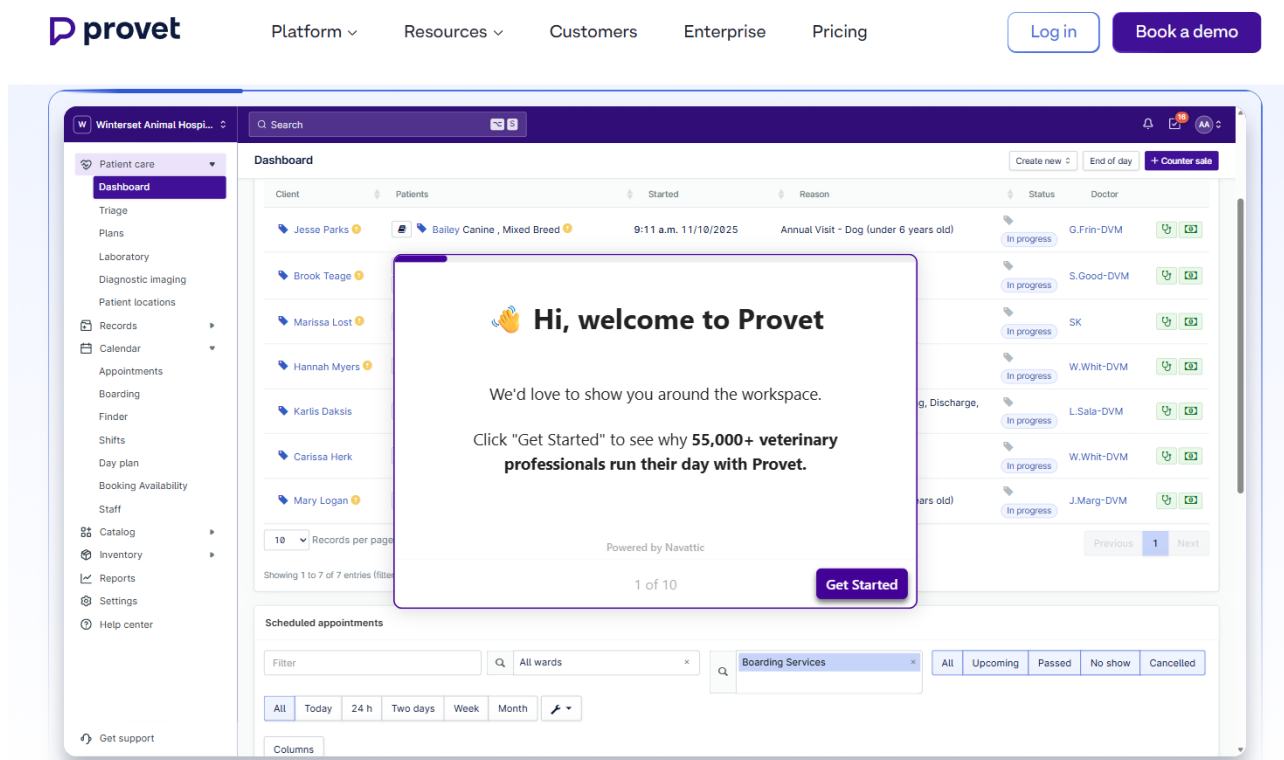


Рис.1.1 – Платформа Provet Cloud [1]

Provet Cloud – це сучасне cloud-програмне забезпечення для управління ветеринарними клініками будь-якого масштабу, що поєднує ведення медичних карток, графіків, комунікацій з клієнтами та звітності.

З плюсів можна виокремити інтуїтивно зрозумілий інтерфейс та зручні інструменти для щоденної роботи, розширений функціонал для комунікацій із клієнтами: автоматичні нагадування, CRM-модулі. Хмарні дані зберігаються централізовано, що підвищує зручність та безпеку, також є підтримка аналітики та звітності для управлінських рішень. Разом з тим система повністю залежить

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		14

від доступу до мережі Інтернет, а її налаштування та освоєння можуть вимагати додаткового часу для персоналу клініки.

Перейдемо до наступного аналога ENOTE – CRM-системи для автоматизації ветеринарної клініки, зображеної на рисунку 1.2.

Тільки в ENOTE є повний цикл обліку руху грошей із подвійним записом!

Як люблять говорити бізнес-консультанти: "Облік – це мова менеджменту". За допомогою ENOTE керівник отримує повну інформацію щодо ефективності роботи клініки як бізнесу.

Кращі бізнес-практики

- Електронна реєстрація
- Підходить для клінік і малих, і великих
- Кваліфікована підтримка

Досвід багатьох клінік

- Інтеграція з торговим обладнанням
- Кол-центр з інтегрованою телефонією
- Збереження в хмарному сервісі

[Читати більше →](#)

Ми на зв'язку та готові допомогти!

Рис.1.2 – CRM-система ENOTE [2]

ENOTE – хмарне рішення, орієнтоване на автоматизацію процесів ветеринарних клінік, аптек і суміжних закладів. Пропонує облік пацієнтів, медичні картки, інтеграцію з CRM-функціями та зберігання даних у хмарі.

Основними перевагами рішення є простота впровадження, можливість масштабування відповідно до потреб закладу та відсутність необхідності у власній серверній інфраструктурі. Водночас функціональні можливості системи є більш обмеженими порівняно з комплексними рішеннями для великих клінік, а окремі спеціалізовані модулі можуть бути відсутні або реалізовані частково.

Наступний аналог – це Clinicia, веборієнтована система управління ветеринарною клінікою, що дозволяє зберігати повні електронні медичні записи тварин, планувати графіки прийомів, вести облік запасів медикаментів та

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		15

формувати аналітичні звіти. Система підтримує автоматичні нагадування для вакцинацій і сповіщення клієнтів через email/WhatsApp.

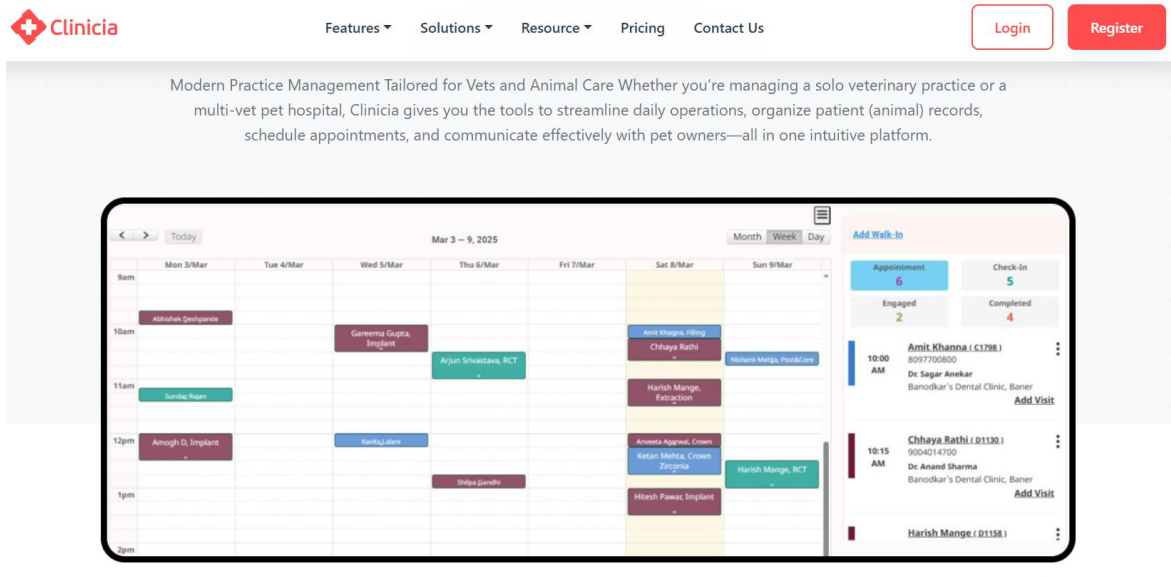


Рис.1.3 – Головна сторінка Clinicia [3]

До плюсів можна віднести наявність повного функціоналу медичних записів та планування, включно з автоматичними нагадуваннями для власників тварин. Можливість управління запасами що дозволяє своєчасно контролювати залишки препаратів та уникати їх нестачі. Підтримка розмежування прав доступу користувачів забезпечує належний рівень безпеки даних та чіткий розподіл обов'язків між персоналом клініки. До недоліків можна віднести комерційне рішення з ймовірною платною підпискою, що може бути затратним для малих клінік. Крім того, не всі функції доступні в базовому тарифі, а відсутність повноцінної демо-версії ускладнює оцінку системи перед придбанням ліцензії.

Виходячи з наявних переваг та недоліків, складемо порівняльну таблицю вищезгаданих програмних аналогів (табл. 1.1). Вона дозволить наочно оцінити відмінності між системами за ключовими критеріями та обґрунтувати вибір підходу до розробки власного програмного забезпечення.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		16

Таблиця порівняння вебплатформ управління ветеринарною клінікою

Характеристика	Provet Cloud	ENOTE	Clinicia
Ведення медичних карток	+	+	+
Онлайн-запис на прийом	+	+	+
Управління розкладом лікарів	+	+	+
Автоматичні сповіщення	+	-	+
Управління інвентарем	+	+	+-
Аналітика та звітність	+	+-	+
Розмежування ролей користувачів	+	+	+
Хмарне зберігання даних	+	+	+
Простота впровадження	+-	+-	-
Онлайн-консультації	-	-	-
Система відгуків та оцінювання лікарів	-	-	-
Вартість використання	Висока	Середня	Висока

Проведений аналіз існуючих аналогів показує, що сучасні системи управління ветеринарними клініками пропонують різні рівні автоматизації: від базових функцій обліку та планування, до більш комплексних рішень із CRM-інструментами та розширеною аналітикою.

З огляду на недоліки існуючих програмних продуктів, зокрема високу вартість рішень для малих ветеринарних клінік, обмежений функціонал доступних безкоштовних систем та недостатню адаптацію до локальних потреб користувачів, було визначено основні вимоги до розроблюваної вебплатформи. Насамперед система повинна бути доступною для малих і середніх ветеринарних практик, що дозволить впроваджувати її без значних фінансових витрат.

Важливою вимогою є також простота та зручність використання, що досягається завдяки створенню адаптованого інтерфейсу для різних категорій користувачів, зокрема адміністраторів, лікарів та клієнтів.

Крім того, платформа повинна мати веборієнтовану архітектуру та підтримувати можливість масштабування, що забезпечить її ефективне використання як у невеликих ветеринарних кабінетах, так і в більших клініках. Для повноцінної автоматизації діяльності закладу система має охоплювати основні бізнес-процеси, включаючи ведення обліку клієнтів і тварин, планування прийомів, зберігання медичних записів, формування нагадувань та сповіщень, а також забезпечувати базові можливості управління взаємовідносинами з клієнтами. Додатково важливою характеристикою є гнучкість програмного продукту, яка дозволить у майбутньому розширювати його функціональні можливості шляхом розробки мобільних застосунків, інтеграції з лабораторними інформаційними системами або впровадження сервісів дистанційних консультацій та телемедицини.

Такий підхід поєднає переваги існуючих аналогів, усуваючи їхні недоліки, і задовольнить реальні потреби ветеринарних клінік.

1.3 Архітектура та узагальнена структура програмного проєкту

Розроблюваний програмний продукт являє собою вебплатформу для управління ветеринарною клінікою, побудовану на основі сучасних принципів клієнт-серверної архітектури. Такий підхід забезпечує гнучкість і масштабованість системи, централізоване зберігання даних та можливість доступу до функціоналу з будь-якого пристрою, що має веббраузер. Вибір клієнт-серверної моделі обумовлено також вимогами до безпеки: вся критична бізнес-логіка та дані зосереджені на сервері, а клієнтська частина виконує лише функції відображення та введення інформації.

Архітектура системи реалізується за багаторівневим принципом та складається з трьох основних рівнів: рівня представлення, рівня прикладної логіки та рівня даних [7]. Розділення системи на окремі логічні рівні дозволяє ізолювати користувацький інтерфейс від бізнес-логіки та механізмів збереження

		Вознюк В.Ю			ШЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		18

інформації, що підвищує надійність, масштабованість і зручність супроводження програмного продукту. Такий підхід відповідає загальноприйнятим стандартам проєктування програмних систем [22] і дозволяє незалежно модифікувати кожен із рівнів без необхідності внесення змін до інших компонентів, що є особливо важливим у контексті подальшого розвитку системи.

Взаємодія між рівнями здійснюється за чітко визначеними інтерфейсами: клієнтська частина звертається до серверної через REST API за протоколами HTTP та HTTPS [30], а серверна частина взаємодіє з базою даних через відповідний рівень доступу до даних. Така організація забезпечує слабку зв'язність компонентів системи, що є ключовою умовою для її подальшого масштабування та розширення.

Рівень представлення відповідає за взаємодію користувачів із системою та забезпечує зручний і зрозумілий інтерфейс. Даний рівень реалізується у вигляді вебінтерфейсу, доступного через стандартний браузер без необхідності встановлення додаткового програмного забезпечення. Він забезпечує відображення інформації про клієнтів, тварин, медичні записи та заплановані прийоми, а також введення й редагування даних з їх первинною перевіркою на стороні клієнта.

Клієнтську частину реалізовано з використанням фреймворку Vue.js[4], який забезпечує реактивне оновлення інтерфейсу без повного перезавантаження сторінки. Це суттєво підвищує швидкість роботи застосунку та покращує користувацький досвід. Маршрутизацію між розділами платформи реалізовано засобами Vue Router [14], що дозволяє організувати навігацію у вигляді односторінкового застосунку. Управління глобальним станом застосунку здійснюється через бібліотеку Pinia [15], яка забезпечує централізоване зберігання та синхронізацію даних між компонентами інтерфейсу.

Інтерфейс системи адаптовано до трьох категорій користувачів – адміністратора, ветеринарного лікаря та клієнта. Кожна роль отримує персоналізований набір елементів навігації та функціональних модулів, що унеможливорює несанкціонований доступ до даних інших користувачів.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		19

Перевірку прав доступу реалізовано як на рівні клієнтського маршрутизатора, так і на рівні серверного API, що формує подвійний захист від несанкціонованих дій. Такий підхід є стандартною практикою безпеки у вебплатформах і унеможливорює обхід обмежень через пряме звернення до API.

Рівень прикладної логіки є ключовим компонентом системи та відповідає за реалізацію основного функціоналу програмного комплексу. На цьому рівні виконуються операції автентифікації та авторизації користувачів, обробка запитів клієнтської частини, керування записами на прийом, ведення електронних медичних карток тварин, контроль прав доступу до інформації, а також формування статистичних даних. Саме цей рівень забезпечує взаємодію між користувацьким інтерфейсом та базою даних, контролюючи коректність виконання всіх операцій.

Серверну частину реалізовано на платформі Node.js з використанням фреймворку Express.js, що забезпечує високу продуктивність обробки запитів та зручну організацію маршрутів API. Для взаємодії клієнтської та серверної частин використовується бібліотека Axios [17], яка забезпечує асинхронну передачу даних у форматі JSON. Автентифікацію користувачів реалізовано за допомогою бібліотеки Passport.js із підтримкою OAuth 2.0 [27], що дозволяє користувачам входити до системи через обліковий запис Google. Для автоматичного виконання планових задач – зокрема надсилання нагадувань про прийоми та сповіщень про прийом – застосовано бібліотеку Node-Cron [26], яка забезпечує регулярний запуск відповідних процесів за заданим розкладом.

Бізнес-логіку системи організовано за модульним принципом: кожен функціональний модуль (управління користувачами, записи на прийом, медичні картки, розклад, статистика, сповіщення) є самостійною одиницею з чітко визначеними інтерфейсами взаємодії. Це забезпечує незалежність модулів один від одного та спрощує їх тестування і подальше вдосконалення.

Рівень даних призначений для довготривалого збереження інформації, необхідної для роботи програмного комплексу. Для реалізації цього рівня використовується система керування базами даних PostgreSQL[16], у якій

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		20

зберігаються відомості про користувачів, клієнтів, тварин, записи на прийом, медичні картки, розклад лікарів та інші службові дані. Використання PostgreSQL забезпечує підтримку реляційної моделі даних, цілісність інформації завдяки механізмам зовнішніх ключів і транзакцій, надійність збереження та можливість резервного копіювання і відновлення даних.

Базу даних розгорнуто на хмарній платформі Neon[18], яка надає керований PostgreSQL-сервіс[21] із підтримкою автоматичного масштабування та резервного копіювання. Такий підхід усуває необхідність адміністрування власної серверної інфраструктури та забезпечує високу доступність бази даних. Клієнтську частину розгорнуто на платформі Vercel[19], що забезпечує автоматичне розгортання оновлень та стабільну доступність застосунку.

Узагальнена структура системи передбачає такий порядок взаємодії: користувач через браузер звертається до клієнтської частини, реалізованої на Vue.js; клієнтська частина формує HTTP-запити до серверного API на основі Node.js; сервер обробляє запити, виконує необхідну бізнес-логіку та звертається до бази даних PostgreSQL; результат повертається клієнту у форматі JSON і відображається в інтерфейсі. Усі запити між клієнтом і сервером передаються захищеним каналом із використанням протоколу HTTPS[29], що унеможливорює перехоплення даних.

Запропонована архітектура забезпечує модульність, масштабованість та безпечність програмного комплексу, що дозволяє ефективно автоматизувати основні процеси діяльності ветеринарної клініки та створює передумови для подальшого розширення функціональних можливостей системи.

1.4 Інструментальні засоби реалізації програмного продукту та вимоги до апаратного забезпечення

Під час розробки вебплатформи для управління ветеринарною клінікою було прийнято рішення використовувати сучасний стек технологій, орієнтований на швидку розробку, масштабованість, зручність у підтримці та високу продуктивність. Вибір інструментальних засобів базувався на вимогах до функціоналу системи, вимогах до безпеки, інтеграційних можливостей та

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		21

поширеності технологій у сучасній веброзробці [10]. Нижче розглянуто кожен із обраних інструментів, обґрунтовано доцільність його використання та описано роль у загальній структурі проєкту.

Основною мовою програмування обрано JavaScript [5], який є стандартом для розробки клієнтської частини вебзастосунків та широко використовується у сучасних інформаційних системах [13]. JavaScript є динамічною, об'єктно-орієнтованою мовою програмування, що забезпечує інтерактивну взаємодію з користувачем, асинхронний обмін даними з сервером, керування DOM-структурою та динамічну зміну вмісту вебсторінок. Важливою перевагою JavaScript[12] у контексті цього проєкту є його універсальність: завдяки платформі Node.js ту саму мову використано як для клієнтської, так і для серверної частини, що уніфікує кодову базу, спрощує розробку та зменшує поріг входу для нових розробників.

Для реалізації клієнтської частини вебплатформи обрано фреймворк Vue.js [4], який є прогресивним JavaScript-фреймворком для створення інтерфейсів користувача та односторінкових застосунків. Vue.js забезпечує компонентний підхід до розробки – інтерфейс розбивається на логічні, повторно використовувані компоненти, кожен із яких містить власну розмітку, стилі та логіку. Це суттєво спрощує підтримку та масштабування системи, оскільки зміни в одному компоненті не впливають на решту. Реактивна система Vue.js автоматично відстежує залежності між даними та елементами інтерфейсу, що забезпечує миттєве оновлення відображення при зміні стану застосунку без необхідності ручного маніпулювання DOM. Порівняно з альтернативними фреймворками – React [11] та Angular – Vue.js має нижчий поріг входу та зрозумілий синтаксис шаблонів, що прискорює розробку. Стиль написання компонентів регламентовано офіційним керівництвом Vue.js Style Guide [9], дотримання якого забезпечує єдність та читабельність коду в усьому проєкті.

Для організації навігації між сторінками застосунку використано Vue Router [14] – офіційну бібліотеку маршрутизації для Vue.js. Vue Router дозволяє декларативно описувати маршрути застосунку та зіставляти їх із відповідними

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		22

компонентами інтерфейсу. Захист маршрутів реалізовано через механізм навігаційних охоронців (navigation guards), які перевіряють права доступу користувача перед переходом на захищену сторінку. Управління глобальним станом застосунку здійснюється через бібліотеку Pinia [15] – сучасне сховище стану для Vue.js, яке прийшло на зміну Vuex. Pinia надає зрозумілий API для визначення сховищ, підтримує TypeScript та забезпечує зручну синхронізацію даних між компонентами.

Для побудови структури вебсторінок використано мову розмітки HTML[6], а для опису їх зовнішнього вигляду – каскадні таблиці стилів CSS. Дотримання стандартів W3C [23] при написанні розмітки гарантує сумісність інтерфейсу з усіма сучасними браузерами. Адаптивний дизайн інтерфейсу реалізовано засобами CSS[6] із використанням медіазапитів, що забезпечує коректне відображення платформи на пристроях із різними розмірами екранів – від настільних комп'ютерів до планшетів.

Серверну частину реалізовано на платформі Node.js із використанням фреймворку Express.js. Node.js виконує JavaScript-код на стороні сервера та орієнтований на асинхронну, подієво-керовану обробку запитів, що робить його ефективним рішенням для побудови API-серверів. Express.js надає мінімалістичний інструментарій для організації маршрутів та обробки проміжного програмного забезпечення. Для передачі даних між клієнтом і сервером використовується бібліотека Axios [17], яка надає зручний інтерфейс для виконання асинхронних HTTP-запитів та автоматичне перетворення відповідей у формат JSON.

Автентифікацію та авторизацію користувачів реалізовано за допомогою бібліотеки Passport.js [20] із підтримкою протоколу OAuth 2.0 через сервіс Google [28]. Такий підхід дозволяє користувачам входити до системи за допомогою наявного облікового запису Google без необхідності створення окремого пароля. Це підвищує зручність використання та безпеку, оскільки відповідальність за зберігання та захист облікових даних перекладається на надійного

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		23

постачальника ідентифікації. Після успішної автентифікації сервер формує сесію користувача, яка використовується для подальшої авторизації запитів.

Для автоматичного виконання планових задач застосовано бібліотеку Node-Cron [27], яка реалізує планувальник задач на основі синтаксису cron-виразів. За допомогою цього інструменту реалізовано автоматичне надсилання нагадувань клієнтам про заплановані прийоми та сповіщень про необхідність вакцинації тварин. Це дозволяє автоматизувати рутинні комунікаційні задачі клініки без участі адміністратора.

Для контролю версій та організації командної розробки використано систему GitLab [25]. GitLab забезпечує зберігання репозиторію з вихідним кодом, відстеження змін, керування гілками розробки та проведення код-рев'ю. Інструменти GitLab CI/CD [28] дозволяють автоматизувати процеси перевірки коду та розгортання оновлень, що підвищує якість і стабільність програмного продукту.

Щодо вимог до апаратного та програмного забезпечення, розроблена платформа є хмарним вебзастосунком і не висуває специфічних вимог до обладнання кінцевих користувачів. Для роботи із системою достатньо будь-якого пристрою – персонального комп'ютера, ноутбука або планшета – з доступом до мережі Інтернет та встановленим сучасним веббраузером, зокрема Google Chrome, Mozilla Firefox, Microsoft Edge або Safari актуальних версій. Мінімальна рекомендована швидкість інтернет-з'єднання становить 1 Мбіт/с, що є достатнім для коректної роботи інтерфейсу та передачі даних. Серверна інфраструктура повністю забезпечується хмарними платформами Neon та Vercel, що звільняє адміністрацію клініки від необхідності придбання та обслуговування власного серверного обладнання і суттєво знижує загальну вартість впровадження системи.

1.5 Варіанти використання системи

На основі проведеного аналізу вимог було визначено ключових акторів та основні сценарії взаємодії із вебплатформою управління ветеринарною клінікою. Система передбачає три категорії користувачів, кожна з яких має

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		24

власний набір прав та функціональних можливостей, що визначаються її роллю в процесах клініки (таблиця 1.2).

Таблиця 1.2

Основні актори системи

Актор	Опис
Клієнт (власник тварини)	Користувач системи, що записує тварин на прийом, переглядає медичні дані та здійснює оплату
Ветеринар	Медичний працівник, що веде прийоми, медичні записи та призначення
Адміністратор	Працівник клініки, відповідальний за розклад, фінанси, інвентар та управління системою

Кожен актор взаємодіє із системою відповідно до своїх функціональних обов'язків. Нижче розглянуто основні сценарії взаємодії з вебплатформою управління ветеринарною клінікою.

Адміністратор є центральною фігурою в операційній діяльності клініки, він відповідає за реєстрацію клієнтів, управління розкладом лікарів та контроль за загальним станом системи. Реєстрація клієнта відбувається після авторизації адміністратора в системі: він переходить до розділу «Клієнти», обирає функцію додавання нового клієнта та вносить контактні дані власника (ПІБ, телефон, email) і відомості про тварину (вид, порода, вік, особливості здоров'я). Після збереження даних клієнт може увійти в систему та самостійно створювати записи.

Запис на прийом здійснюється через модуль розкладу: адміністратор обирає тип послуги, лікаря, переглядає вільні часові слоти та вносить клієнта у відповідний час. Система автоматично створює запис і надсилає клієнту та лікарю сповіщення. Повний перелік сценаріїв для адміністратора з описом відповідних дій та реакції системи наведено в таблиці 1.3.

Дії адміністратора

Сценарій	Дії адміністратора	Реакція системи
Реєстрація клієнта	Перехід до розділу «Клієнти», внесення контактних даних власника (ПІБ, телефон, email) та відомостей про тварину (вид, порода, вік, особливості здоров'я)	Збереження даних
Перегляд історії лікувань	Пошук картки пацієнта, перегляд діагнозів, призначень та проведених процедур	Відображення повної медичної історії тварини
Управління записами	Перегляд розкладу ветеринарів, додавання змін, перенесення або скасування прийомів	Автоматичне оновлення інформації
Запис клієнта на прийом	Вибір лікаря у модулі розкладу, перегляд вільних часових слотів, внесення клієнта у відповідний час	Створення запису прийому, надсилання сповіщення клієнту, лікарю
Управління графіком лікарів	Перегляд графіка ветеринарів, додавання або коригування вихідних та перерв	Автоматичне оновлення розкладу
Контроль статистики	Перегляд автоматично сформованих звітів і показників роботи клініки	Система самостійно генерує статистику

Окрему роль у системі відіграє клієнт, який взаємодіє з платформою переважно самостійно та має доступ до особистого кабінету, де зосереджено всі необхідні функції. Після самостійної реєстрації клієнт може додавати своїх тварин до профілю, обирати ветеринара і зручний час для прийому, а також отримувати автоматичні нагадування. Окремо важливими функціями є доступ до

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		26

медичної історії тварини та можливість самостійно перенести або скасувати запис, що автоматично відображається в розкладі лікаря. Детальний опис усіх сценаріїв взаємодії клієнта із системою наведено в таблиці 1.4.

Таблиця 1.4

Дії клієнта

Сценарій	Дії клієнта	Реакція системи
Самостійна реєстрація	Заповнення реєстраційної форми (ПІБ, телефон, email), створення облікового запису	Надання доступу до особистого кабінету
Додавання тварини	Перехід до особистого кабінету, внесення відомостей про тварину (вид, порода, вік, особливості здоров'я)	Збереження даних, відображення тварини у профілі клієнта
Онлайн-запис на прийом	Вибір тварини, послуги, ветеринара, дати й часу, підтвердження запису	Створення запису, надсилання сповіщення
Перегляд медичної історії	Відкриття картки тварини, перегляд діагнозів, призначень, щеплень	Відображення повної медичної історії, можливість завантаження результатів аналізів
Перенесення або скасування прийому	Вибір запису в особистому кабінеті, зміна дати й часу або скасування	Оновлення розкладу, надсилання підтвердження клієнту та ветеринару
Отримання сповіщень	-	Автоматичне надсилання нагадувань про прийом

Ветеринар працює з системою безпосередньо під час прийомів та між ними: він переглядає розклад запланованих прийомів, відкриває медичні картки тварин, вносить результати огляду, формує призначення та завантажує файли аналізів. Усі зміни автоматично зберігаються в системі та стають доступними адміністратору й клієнту відповідно до їхніх прав доступу. Повний перелік сценаріїв для лікаря наведено в таблиці 1.5.

Таблиця 1.5

Дії лікаря

Сценарій	Дії лікаря	Реакція системи
Проведення прийому	Відкриття запланованого запису: лікар переглядає історію тварини, вносить результати огляду та формує призначення.	Автоматичне оновлення статусу прийому
Робота з медичною картою	Додавання діагнозу, внесення призначень, завантаження файлів аналізів	Оновлена картка пацієнта
Онлайн-консультація	Проведення відеосеансу або чату з клієнтом	Автоматичне збереження результатів у картці пацієнта
Перенесення або скасування прийому	Додавання діагнозу, внесення призначень, завантаження файлів аналізів	Оновлення запису

Висновки до першого розділу.

Проведено аналіз предметної області, у результаті якого були визначені основні вимоги до вебплатформи управління ветеринарною клінікою, основним призначенням якої є автоматизація процесів запису клієнтів на прийом, ведення медичних карт тварин та управління діяльністю персоналу.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		28

У процесі дослідження аналогічних і схожих за функціональним призначенням програмних продуктів було проаналізовано та порівняно різні системи для управління ветеринарними клініками. Основний функціонал таких систем включає організацію прийомів, облік пацієнтів, збереження медичних даних та управління внутрішніми ресурсами. На основі проведеного аналізу було сформовано основні вимоги до власного програмного продукту.

На основі визначених вимог і критеріїв було обрано набір інструментальних засобів та технологій для реалізації вебплатформи, що забезпечують ефективну, надійну та зручну роботу системи.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				29
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБПЛАТФОРМИ УПРАВЛІННЯ ВЕТЕРИНАРНОЮ КЛІНІКОЮ

2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи

Вебплатформа для управління ветеринарною клінікою надає вебінтерфейс, через який користувачі можуть записуватись на прийом, переглядати медичні картки своїх тварин та взаємодіяти з персоналом клініки. Функціональність системи залежить від активності всіх учасників: адміністраторів, лікарів та власників тварин. Система охоплює повний цикл надання ветеринарних послуг – від реєстрації пацієнта до збереження результатів лікування та формування звітності. Для наочної взаємодії між елементами додатку було створено діаграму варіантів використання (рис. 2.1).

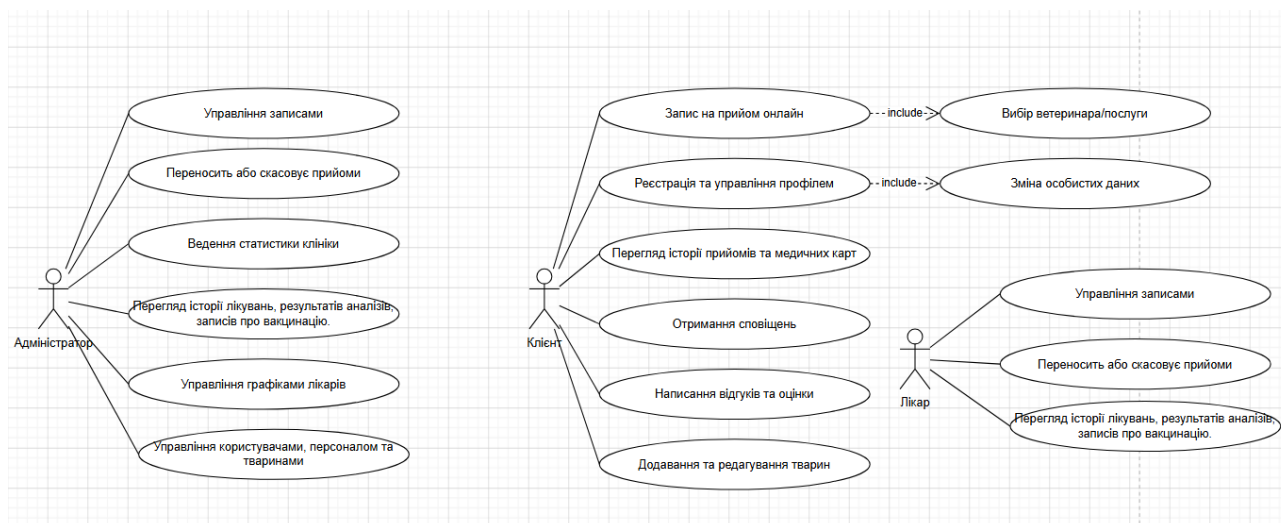


Рис.2.1 – Діаграма варіантів використання

Опис діаграми варіантів використання наведено в таблицях 2.1 – 2.12.

Перший прецедент стосується реєстрації та управління профілем користувача. Як показано в таблиці 2.1, основним актором цього варіанту використання є власник тварини, який може створити обліковий запис, авторизуватись у системі та керувати особистими даними.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		30

Таблиця 2.1

Опис прецеденту «Реєстрація та управління профілем»

Назва варіанту використання	Реєстрація та управління профілем
Основні актори	Власник тварини
Короткий опис	Клієнт може зареєструватись у системі, створивши обліковий запис із особистими даними. Після реєстрації користувач має можливість авторизуватись, редагувати особисті дані, змінювати пароль та керувати інформацією про своїх тварин.

Наступний прецедент описує процес онлайн-запису клієнта на прийом. Таблиця 2.2 відображає основні характеристики цього варіанту використання, де авторизований клієнт самостійно обирає лікаря, послугу та зручний час.

Таблиця 2.2

Опис прецеденту «Запис на прийом онлайн»

Назва варіанту використання	Запис на прийом онлайн
Основні актори	Клієнт (власник тварини)
Короткий опис	Авторизований клієнт може записати свою тварину на прийом до ветеринара, обравши лікаря, послугу, зручну дату та час. Прецедент включає вибір ветеринара або послуги як обов'язковий підпрецедент.

Таблиця 2.3 описує прецедент перегляду медичної інформації. Цей варіант використання є спільним для двох акторів – клієнта та лікаря, кожен із яких має доступ до медичних карток відповідно до своєї ролі.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		31

Таблиця 2.3

Опис прецеденту «Перегляд історії прийомів та медичних карток»

Назва варіанту використання	Перегляд історії прийомів та медичних карток
Основні актори	Клієнт (власник тварини), лікар
Короткий опис	Клієнт може переглядати повну медичну картку своїх тварин, включаючи всі попередні прийоми, встановлені діагнози, призначені лікування та результати аналізів. Лікар має доступ до медичних карток своїх пацієнтів.

Важливим елементом взаємодії системи з клієнтом є автоматичне сповіщення. Таблиця 2.4 відображає прецедент отримання сповіщень, який реалізується без активних дій користувача – система надсилає повідомлення автоматично.

Таблиця 2.4

Опис прецеденту «Отримання сповіщень»

Назва варіанту використання	Отримання сповіщень
Основні актори	Клієнт (власник тварини)
Короткий опис	Система автоматично надсилає клієнту сповіщення про підтвердження або скасування запису на прийом, нагадування перед прийомом.

Для підвищення якості сервісу та забезпечення зворотного зв'язку між клієнтами і персоналом клініки передбачено можливість залишати відгуки. Відповідний прецедент описано в таблиці 2.5.

Таблиця 2.5

Опис прецеденту «Написання відгуків та оцінки»

Назва варіанту використання	Написання відгуків та оцінки
Основні актори	Клієнт (власник тварини)
Короткий опис	Після завершення прийому клієнт може залишити відгук та виставити оцінку лікарю або наданій послугі. Відгуки допомагають іншим клієнтам обирати фахівця та мотивують персонал до якісної роботи.

Окрім управління записами та перегляду медичної інформації, клієнт має можливість самостійно вести реєстр своїх тварин у системі. Таблиця 2.6 описує відповідний прецедент.

Таблиця 2.6

Опис прецеденту «Додавання та редагування тварин»

Назва варіанту використання	Додавання та редагування тварин
Основні актори	Клієнт (власник тварини)
Короткий опис	Клієнт може додати нову тварину в систему та редагувати інформацію про вже наявну.

Наступні прецеденти стосуються функцій адміністратора. Таблиця 2.7 описує управління записами на прийом – одну з ключових адміністративних функцій, що забезпечує повний контроль над розкладом клініки.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		33

Опис прецеденту «Управління записами»

Назва варіанту використання	Управління записами
Основні актори	Адміністратор
Короткий опис	Адміністратор має повний доступ до всіх записів на прийом у системі. Він може переглядати, підтверджувати, редагувати, переносити або скасовувати записи клієнтів.

Невід'ємною складовою адміністративного функціоналу є управління графіками лікарів. Як показано в таблиці 2.8, адміністратор формує робочий розклад персоналу та може вносити до нього зміни у разі потреби.

Опис прецеденту «Управління графіками лікарів»

Назва варіанту використання	Управління графіками лікарів
Основні актори	Адміністратор
Короткий опис	Адміністратор формує графік лікаря, робочі години, також може додавати блокування для перерви або відпустки

Для забезпечення актуальності бази користувачів системи адміністратору надано відповідні інструменти управління. Таблиця 2.9 описує прецедент управління користувачами, що охоплює додавання, редагування та видалення облікових записів. Завдяки цьому функціоналу адміністратор може оперативно реагувати на зміни у складі персоналу клініки: призначати ролі новим співробітникам, коригувати права доступу наявних користувачів або деактивувати облікові записи звільнених працівників. Це забезпечує надійний контроль над тим, хто має доступ до системи та які операції може виконувати.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		34

Таблиця 2.9

Опис прецеденту «Управління користувачами»

Назва варіанту використання	Управління користувачами
Основні актори	Адміністратор
Короткий опис	Адміністратор може додавати нових клієнтів та лікарів, а також редагувати та видаляти наявну інформацію.

Центральним елементом клінічної роботи є ведення медичних карток тварин. Таблиця 2.10 описує прецедент, що є спільним для лікаря та адміністратора і забезпечує повний цикл роботи з медичною документацією пацієнта.

Таблиця 2.10

Опис прецеденту «Перегляд історії лікувань, результатів аналізів та записів про вакцинацію»

Назва варіанту використання	Перегляд та ведення медичних карток
Основні актори	Лікар, адміністратор
Короткий опис	Лікар має доступ до повної медичної картки тварини перед та під час прийому. Після завершення прийому лікар вносить результати огляду, встановлені діагнози, призначене лікування, результати аналізів та інформацію про вакцинацію.

Для контролю ефективності роботи клініки адміністратору доступний модуль статистики. Як описано в таблиці 2.11, цей прецедент дозволяє відстежувати ключові показники діяльності закладу в режимі реального часу. Отримані дані можуть слугувати основою для прийняття управлінських рішень

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		35

щодо оптимізації розкладу роботи персоналу та підвищення якості обслуговування пацієнтів.

Таблиця 2.11

Опис прецеденту «Ведення статистики клініки»

Назва варіанту використання	Ведення статистики клініки
Основні актори	Адміністратор
Короткий опис	Адміністратор має доступ до загальної статистики клініки. Може відстежувати популярні послуги, кількість прийомів, статуси прийомів, завантаженість та рейтинг лікарів.

Останній прецедент, описаний у таблиці 2.12, стосується управління записами з боку ветеринарного лікаря. На відміну від адміністратора, лікар має доступ лише до власного розкладу та може здійснювати обмежені дії щодо записів до нього.

Таблиця 2.12

Опис прецеденту «Управління записами»

Назва варіанту використання	Управління записами
Основні актори	Лікар
Короткий опис	Лікар має доступ записів до записів на прийом до нього, може переносити або скасовувати записи.

Основна мета розробки вебплатформи для ветеринарної клініки полягає у створенні централізованого цифрового середовища, що дозволяє ефективно організувати роботу клініки, забезпечити зручний запис на прийом та прозорий доступ до медичної інформації. Платформа спрямована на автоматизацію ключових бізнес-процесів: управління розкладом лікарів, ведення електронних медичних карток тварин, обробку звернень клієнтів та формування звітів для

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		36

адміністрації. Завдяки розподілу функцій між трьома ролями – адміністратором, лікарем та власником тварини – система забезпечує чіткий контроль над доступом до даних та гарантує персоналізований досвід для кожної категорії користувачів.

Вимоги користувачів визначаються відповідно до ролей, передбачених у системі. Основними категоріями користувачів є адміністратор, лікар та власник тварини, кожна з яких має власний набір функціональних можливостей.

Адміністратор є внутрішнім користувачем системи та відповідає за організацію роботи вебплатформи. Після проходження авторизації він отримує доступ до функцій управління обліковими записами користувачів, що включає створення, редагування та деактивацію облікових записів. Крім того, адміністратор здійснює управління персоналом клініки, зокрема додавання та редагування інформації про лікарів, контролює записи на прийом, керує графіками роботи спеціалістів та переглядає статистичну інформацію щодо діяльності клініки. До його обов'язків також належить підтримка актуальності довідкової інформації та організація взаємодії між персоналом і клієнтами.

Лікар є внутрішнім користувачем системи, який використовує вебплатформу для організації лікувального процесу. Після авторизації він отримує доступ до власного розкладу прийомів, інформації про пацієнтів та їх медичних карток. Лікар має можливість переглядати історію попередніх звернень тварини, фіксувати результати огляду, встановлені діагнози, призначене лікування та інші медичні відомості. Крім того, система надає лікарю інструменти для роботи із записами на прийом.

Власник тварини є зовнішнім користувачем вебплатформи та використовує систему для взаємодії з ветеринарною клінікою. Після реєстрації та авторизації користувач отримує доступ до власного профілю, де може керувати інформацією про своїх тварин. Система надає можливість здійснювати онлайн-запис на прийом до обраного лікаря, переглядати статус створених записів, ознайомлюватися з історією лікування та медичними записами своїх тварин. Крім того, власник тварини може переглядати інформацію про доступні

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		37

послуги клініки, отримувати сповіщення про майбутні прийоми та залишати оцінки щодо якості наданих послуг.

Функціональні вимоги визначають перелік можливостей, які повинна забезпечувати система для задоволення потреб користувачів. Вебплатформа повинна підтримувати механізми реєстрації та авторизації користувачів із розподілом прав доступу відповідно до ролей. Необхідно забезпечити захищений доступ до персональних даних користувачів та медичної інформації про пацієнтів. Система повинна надавати можливість онлайн-запису на прийом із вибором лікаря, дати та часу візиту, а також відображати актуальний графік роботи лікарів і доступні часові слоти. Важливою функцією є ведення електронних медичних карток тварин із повною історією звернень, діагнозів та призначень. Також система повинна забезпечувати можливість внесення результатів прийому лікарем, формування статистичних даних та звітів для адміністрації клініки, а також надсилання користувачам сповіщень про зміни статусу записів та інші важливі події.

Нефункціональні вимоги визначають якісні характеристики програмного продукту. Вебплатформа повинна бути простою та зручною у використанні, забезпечуючи можливість швидкого освоєння інтерфейсу новими користувачами без необхідності проходження спеціального навчання. Час відгуку системи на дії користувача не повинен перевищувати декількох секунд, а інтерфейс має коректно відображатися на персональних комп'ютерах, планшетах та мобільних пристроях. Важливою характеристикою системи є її надійність, що передбачає цілодобову доступність, коректну обробку помилок та перевірку введених користувачами даних перед їх збереженням. Крім того, вебплатформа повинна забезпечувати достатній рівень продуктивності для одночасної роботи значної кількості користувачів без помітного погіршення швидкодії навіть за умов підвищеного навантаження.

Системні вимоги визначають технічні умови функціонування програмного продукту. Робота вебплатформи повинна забезпечуватися на пристроях із мінімальною конфігурацією, що включає процесор із тактовою частотою не

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		38

менше 1,4 ГГц, оперативну пам'ять обсягом від 2 ГБ та наявність стабільного підключення до мережі Інтернет. Серверна частина системи повинна підтримувати роботу з реляційною базою даних для зберігання інформації про користувачів, тварин, прийоми та медичні записи, а також забезпечувати взаємодію між клієнтською та серверною частинами через програмний інтерфейс прикладного програмування. Особлива увага повинна приділятися безпеці даних, що передбачає використання механізмів хешування паролів, захищеного протоколу HTTPS для передачі інформації та сучасних засобів автентифікації та авторизації користувачів.

Аналіз функціональних вимог дозволив виділити основні сутності, що забезпечать програмну реалізацію додатку. На рисунку 2.2 наведено діаграму класів, що використовується для роботи системи.

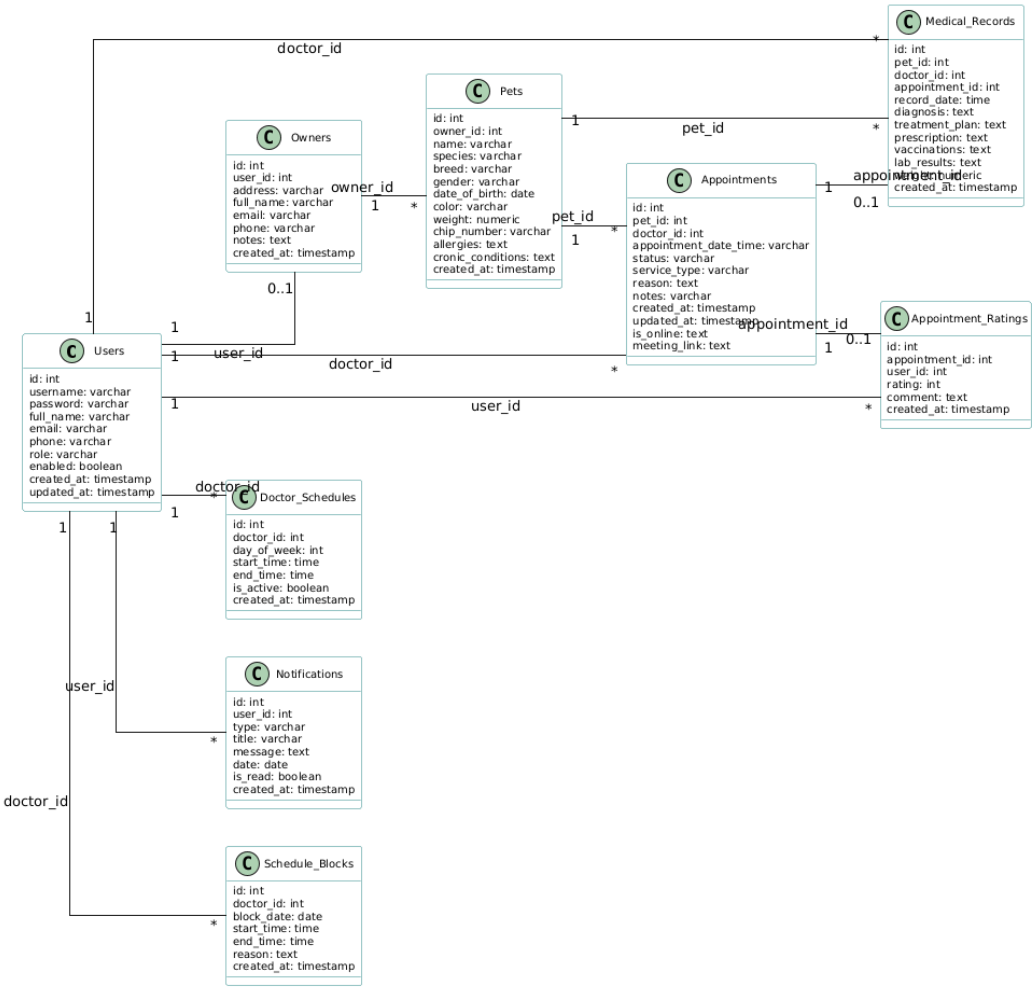


Рис.2.2 – Діаграма класів

Опишемо діаграму класів, наведену на рис. 2.2. Вона відображає основні сутності системи та взаємозв'язки між ними, а також визначає функціональні можливості кожного класу, необхідні для реалізації бізнес-процесів ветеринарної клініки.

Клас User призначений для управління користувачами системи та охоплює всі категорії користувачів, зокрема адміністраторів, лікарів і власників тварин. Даний клас забезпечує виконання операцій реєстрації та авторизації користувачів, оновлення персональної інформації, отримання списку користувачів та видалення облікових записів. Саме цей клас відповідає за ідентифікацію користувачів та розмежування прав доступу відповідно до ролей, визначених у системі.

Клас Owner використовується для зберігання та управління інформацією про власників тварин. Його функціональність включає створення картки власника, отримання переліку зареєстрованих клієнтів, редагування контактної інформації та видалення записів за необхідності. Використання окремого класу дозволяє централізовано зберігати дані про клієнтів клініки та встановлювати зв'язки між власниками та їхніми тваринами.

Клас Pet відповідає за ведення обліку тварин, які є пацієнтами ветеринарної клініки. Він забезпечує створення нових записів про тварин, перегляд їх переліку, отримання детальної інформації про конкретного пацієнта, редагування наявних даних та видалення записів. У межах даного класу зберігаються основні характеристики тварини, включаючи вид, породу, дату народження, номер чіпа та інші відомості, необхідні для ведення медичної документації.

Клас Appointment реалізує функціонал запису на прийом та є одним із центральних елементів системи. Він дозволяє створювати нові записи на прийом, переглядати їх перелік, змінювати статус прийому та скасовувати заплановані візити. Крім того, клас забезпечує можливість отримання інформації про прийоми конкретного лікаря, що використовується під час формування розкладу роботи та планування навантаження персоналу.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		40

Клас MedicalRecord призначений для ведення електронних медичних карток тварин. Він забезпечує створення нових медичних записів, перегляд історії лікування пацієнтів, редагування наявних записів та їх видалення. У медичних картках зберігаються результати оглядів, діагнози, призначення лікарів, відомості про вакцинації та інша інформація, необхідна для якісного ведення історії хвороби тварини.

Клас DoctorSchedule відповідає за управління графіками роботи лікарів. Основними функціями класу є створення та редагування розкладів, отримання інформації про робочі години спеціаліста та видалення графіків. Наявність цього класу дозволяє автоматизувати процес планування прийомів і забезпечити коректне відображення доступного часу для запису клієнтів.

Клас Notification реалізує механізм системних сповіщень. Його функціональність охоплює створення нових повідомлень, отримання списку сповіщень користувача, позначення повідомлень як прочитаних та їх видалення. Завдяки цьому забезпечується своєчасне інформування користувачів про зміни статусу прийомів, створення нових записів та інші важливі події в системі.

Клас AppointmentRating використовується для реалізації механізму оцінювання якості наданих послуг. Він забезпечує створення оцінок та відгуків після завершення прийому, перегляд загального списку оцінок і отримання інформації про оцінки конкретного лікаря. Зібрані відгуки можуть використовуватися адміністрацією клініки для аналізу рівня задоволеності клієнтів та підвищення якості обслуговування.

Між класами встановлені зв'язки, що відображають логіку роботи системи. Один власник може мати декілька тварин, кожна тварина може мати декілька медичних записів та записів на прийом. Лікар може обслуговувати багато прийомів та мати власний графік роботи. Користувачі системи отримують сповіщення щодо змін статусів записів, а після завершення прийому можуть залишати оцінки та відгуки.

Розроблена діаграма класів стала основою для реалізації програмної логіки вебплатформи та проєктування структури бази даних PostgreSQL. Вона

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		41

забезпечує чітке розмежування відповідальностей між компонентами системи та спрощує подальшу підтримку й розширення функціональних можливостей програмного продукту.

2.2 Розробка бази даних системи

Для зберігання інформації у вебплатформі управління ветеринарною клінікою використовується система керування базами даних PostgreSQL. Вона належить до реляційних систем управління базами даних та забезпечує надійне зберігання, обробку й захист інформації. PostgreSQL підтримує механізми транзакцій, зовнішніх ключів, індексування та контролю цілісності даних, що робить її ефективним рішенням для реалізації інформаційних систем медичного призначення.

Структура бази даних забезпечує цілісність даних, мінімізує їх дублювання та дозволяє ефективно організувати взаємозв'язки між основними сутностями системи. Для реалізації функціоналу вебплатформи було створено декілька взаємопов'язаних таблиць, кожна з яких відповідає за зберігання певної категорії інформації.

Опис кожної з таблиць бази даних наведений у таблицях 2.13 – 2.21.

Таблиця Users призначена для зберігання даних користувачів системи. У ній міститься інформація про адміністраторів, лікарів та власників тварин, включаючи облікові дані, контактну інформацію, роль користувача та службові параметри, необхідні для авторизації та розмежування прав доступу.

Таблиця 2.13

Опис таблиці Users

Поле	Тип	Призначення
id	int	Ідентифікатор користувача
username	varchar	Логін користувача
password	varchar	Хешований пароль
full_name	varchar	ПІБ користувача

продовж. табл. 2.13

email	varchar	Електронна пошта
phone	varchar	Номер телефону
role	varchar	Роль користувача
enabled	boolean	Статус активності облікового запису
created_at	timestamp	Дата та час створення запису
updated_at	timestamp	Дата та час останнього оновлення запису

Таблиця Owners використовується для зберігання інформації про власників тварин. У ній зберігаються персональні та контактні дані клієнтів ветеринарної клініки. Наявність окремої таблиці дозволяє встановлювати зв'язок між власником та всіма зареєстрованими на нього тваринами.

Таблиця 2.14

Опис таблиці Owners

Поле	Тип	Призначення
id	int	Ідентифікатор власника
user_id	int	Посилання на користувача
address	varchar	Адреса проживання
full_name	varchar	ПІБ користувача
email	varchar	Електронна пошта
phone	varchar	Номер телефону
notes	text	Додаткова інформація
created_at	timestamp	Дата та час створення запису

Таблиця Pets містить інформацію про тварин, які є пацієнтами клініки. Для кожної тварини зберігаються такі дані, як кличка, вид, порода, дата народження,

стать, вага, номер електронного чіпа, інформація про алергії та хронічні захворювання. Кожен запис пов'язаний із конкретним власником, що дозволяє формувати повну картку пацієнта.

Таблиця 2.15

Опис таблиці Pets

Поле	Тип	Призначення
id	int	Ідентифікатор тварини
owner_id	int	Власник тварини
name	varchar	Кличка
species	varchar	Вид тварини
breed	varchar	Порода
gender	varchar	Стать
date_of_birth	date	Дата народження
color	varchar	Колір
weight	num	Вага
chip_number	varchar	Номер чіпу
allergies	text	Алергії
cronic_conditions	text	Хронічні захворювання
created_at	timestamp	Додано в систему

Таблиця Appointments призначена для зберігання інформації про записи на прийом. У ній містяться відомості про дату та час прийому, обраного лікаря, тварину, статус запису та інші параметри, необхідні для організації процесу обслуговування клієнтів. Дана таблиця є одним із центральних елементів системи та забезпечує взаємодію між лікарями, клієнтами й пацієнтами.

Опис таблиці Appointments

Поле	Тип	Призначення
id	int	Ідентифікатор запису
pet_id	int	Ідентифікатор тварини
doctor_id	int	Лікар
appointment_date_time	varchar	Дата та час прийому
status	varchar	Статус прийому
service_type	varchar	Вид послуги
reason	text	Причина звернення
notes	varchar	Додаткова інформація
created_at	timestamp	Дата та час створення запису
updated_at	timestamp	Дата та час останнього оновлення запису
is_online	text	Статус в мережі
meeting_link	text	Посилання на онлайн-консультацію

Таблиця `Medical_Records` використовується для ведення електронних медичних карток тварин. У ній зберігаються результати оглядів, діагнози, призначене лікування, рекомендації лікарів, відомості про вакцинацію та інша медична інформація. Зв'язок медичних записів із конкретними тваринами та прийомами дозволяє формувати повну історію лікування кожного пацієнта. Це забезпечує лікарям швидкий доступ до попередніх даних під час повторних візитів, що сприяє точнішій діагностиці та більш ефективному підбору лікування. Завдяки структурованому зберіганню медичної інформації система виключає ризик втрати важливих відомостей, характерний для паперового документообігу.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		45

Опис таблиці Medical_Records

Поле	Тип	Призначення
id	int	Ідентифікатор запису
pet_id	int	Ідентифікатор тварини
doctor_id	int	Лікар
appointment_id	int	Номер запису
record_date	time	Дата запису
diagnosis	text	Діагноз
treatment_plan	text	План лікування
prescription	text	Призначення
vaccinations	text	Вакцинації
lab_results	text	Результати аналізів
weight	num	Вага
created_at	timestamp	Дата та час створення запису

Таблиця Doctor_Schedules забезпечує зберігання графіків роботи лікарів. У ній фіксуються робочі дні, години початку та завершення зміни, а також інші параметри, необхідні для формування доступних часових слотів під час запису на прийом. На основі цих даних система автоматично розраховує вільні проміжки часу для кожного лікаря та відображає їх клієнту у зручному форматі при виборі зручного часу візиту. Зв'язок таблиці з даними про лікарів дозволяє коректно відображати індивідуальний розклад кожного спеціаліста, враховуючи можливі відмінності у тривалості робочого дня чи вихідних днів. Це забезпечує точність і актуальність інформації про доступність лікарів, мінімізуючи ймовірність виникнення конфліктів при записі на прийом.

Таблиця 2.18

Опис таблиці Doctor_Schedules

Поле	Тип	Призначення
id	int	Ідентифікатор запису
doctor_id	int	Лікар
day_of_week	int	Дата та час прийому
start_time	time	Початок зміни
end_time	time	Кінець зміни
is_active	boolean	Статус прийому
created_at	timestamp	Дата та час створення запису

Таблиця Notifications призначена для реалізації системи сповіщень. У ній зберігаються повідомлення, що надсилаються користувачам під час виконання різних операцій у системі. Для кожного сповіщення фіксуються його тип, зміст, дата створення та статус прочитання.

Таблиця 2.19

Опис таблиці Notifications

Поле	Тип	Призначення
id	int	Ідентифікатор запису
user_id	int	Одержувач
type	varchar	Тип сповіщення
title	varchar	Заголовок
message	text	Текст повідомлення
date	date	Дата
Is_read	boolean	Статус прочитання
created_at	timestamp	Дата створення запису

Таблиця Appointment_Ratings використовується для зберігання оцінок та відгуків клієнтів після завершення прийому. Інформація з цієї таблиці може

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				47
Змн.	Арк.	№ докум.	Підпис	Дата		

використовуватися для аналізу якості надання послуг, оцінювання роботи лікарів та отримання зворотного зв'язку від клієнтів.

Таблиця 2.20

Опис таблиці Appointment_Ratings

Поле	Тип	Призначення
id	int	Ідентифікатор запису
appointment_id	int	Номер запису
user_id	int	Автор відгуку
rating	int	Оцінка
comment	text	Коментар
created_at	timestamp	Дата створення запису

Таблиця Schedule_Blocks призначена для зберігання інформації про блокування окремих часових проміжків у графіку роботи лікарів. Такі блокування можуть використовуватися для позначення відпусток, перерв, лікарняних або інших причин тимчасової недоступності спеціаліста. Наявність даної таблиці дозволяє системі автоматично виключати заблоковані часові інтервали з переліку доступних для запису на прийом.

Таблиця 2.21

Опис таблиці Schedule_blocks

Поле	Тип	Призначення
id	int	Ідентифікатор запису
doctor_id	int	Лікар
block_date	date	Дата блокування
start_time	time	Початок блокування
end_time	time	Кінець блокування
reason	text	Причина блокування
created_at	timestamp	Дата створення запису

Для забезпечення цілісності даних між таблицями використовуються зовнішні ключі. Наприклад, кожна тварина пов'язана з конкретним власником, кожний запис на прийом містить посилання на лікаря та тварину, а медичний запис створюється для певної тварини та прийому. Використання реляційної моделі даних дозволяє уникнути дублювання інформації та забезпечує ефективну взаємодію між модулями системи.

Розроблена структура бази даних забезпечує зберігання всієї необхідної інформації для функціонування вебплатформи, включаючи управління користувачами, облік тварин, ведення медичних карток, формування графіків роботи лікарів, запис на прийом та систему сповіщень.

2.3 Проєктування та реалізація алгоритмів роботи системи

Розроблена вебплатформа управління ветеринарною клінікою забезпечує автоматизацію основних процесів діяльності закладу та об'єднує всі необхідні функції в єдиному інформаційному середовищі. Система дозволяє здійснювати управління користувачами, власниками тварин, пацієнтами, записами на прийом, медичними картками, графіками роботи лікарів, сповіщеннями та оцінками клієнтів. Використання централізованої бази даних забезпечує цілісність інформації та швидкий доступ до неї для всіх авторизованих користувачів відповідно до наданих прав доступу. Алгоритм роботи системи наведено на рисунку 2.3, а також детально описано його основні етапи.

Після відкриття вебплатформи користувач потрапляє на головну сторінку, де може ознайомитися з інформацією про ветеринарну клініку, переліком доступних послуг та перейти до сторінки авторизації. Для роботи з персональними даними та функціональними можливостями системи необхідно пройти процедуру входу. Під час авторизації користувач вводить свої облікові дані, які передаються на сервер для перевірки. У разі успішної перевірки система створює користувацьку сесію та визначає роль користувача. Залежно від отриманої ролі відкривається відповідний інтерфейс із доступом лише до дозволених функцій. Адміністратор має найширші права доступу та виконує функції управління роботою всієї системи.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		49

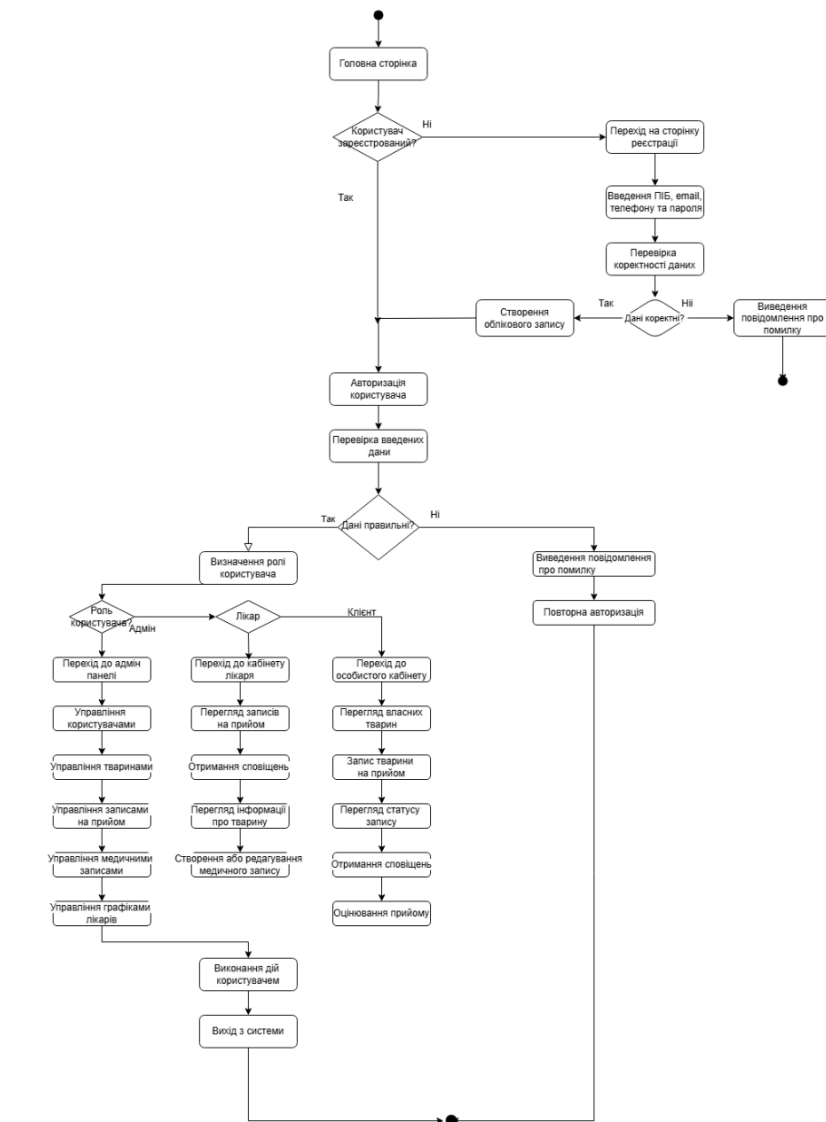


Рис.2.3 Діаграма активності.

Після входу він отримує доступ до адміністративної панелі, де може переглядати статистичну інформацію, керувати користувачами, графіками роботи лікарів, записами на прийом та медичними картками пацієнтів. Також адміністратор здійснює контроль коректності даних, підтримує актуальність інформації та забезпечує належне функціонування всіх модулів системи. Важливою складовою алгоритму роботи системи є механізм управління графіками роботи лікарів. Під час створення або редагування графіка адміністратор задає робочі дні та години прийому. На основі цих даних система формує доступні часові інтервали для запису клієнтів. Якщо певний лікар

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		50

перебуває у відпустці, на лікарняному або тимчасово відсутній, відповідні часові проміжки можуть бути заблоковані, що унеможлиблює створення нових записів на цей період.

Після створення або зміни запису на прийом система автоматично формує відповідні сповіщення. Вони можуть інформувати користувачів про успішне бронювання часу, зміну статусу прийому, необхідність оцінити якість наданих послуг або інші події. Наявність механізму сповіщень забезпечує оперативне інформування користувачів та покращує взаємодію між клієнтами і персоналом клініки.

Лікар після авторизації переходить до власного робочого кабінету, де отримує доступ до інформації про заплановані прийоми та закріплених за ним пацієнтів. Перед початком прийому лікар може переглянути дані про тварину, ознайомитися з історією попередніх звернень, результатами лікування та медичними записами. Після завершення огляду лікар формує новий медичний запис, у якому фіксує результати обстеження, встановлений діагноз, рекомендації та призначене лікування. Усі внесені зміни автоматично зберігаються в базі даних та стають доступними для подальшого використання.

Клієнт після входу до системи отримує доступ до власного кабінету, де може переглядати інформацію про зареєстрованих тварин, створювати записи на прийом та переглядати історію відвідувань клініки. Під час запису на прийом користувач обирає необхідну тварину, лікаря, дату та доступний часовий слот. Система автоматично перевіряє графік роботи лікаря та доступність обраного часу. Після успішного створення запису інформація зберігається в базі даних та відображається всім учасникам процесу. Після завершення прийому клієнт отримує можливість залишити оцінку та коментар щодо якості обслуговування. Відгук зберігається в системі та використовується для аналізу рівня задоволеності клієнтів. Отримані оцінки можуть бути використані адміністрацією клініки для контролю якості надання послуг та вдосконалення організації роботи закладу.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		51

Таким чином реалізований алгоритм забезпечує взаємодію між усіма учасниками процесу лікування та обслуговування тварин. Використання рольової моделі доступу дозволяє розмежувати функціональні можливості користувачів та забезпечити безпечне використання вебплатформи.

2.4 Реалізація функціоналу системи вебплатформи управління ветеринарною клінікою

Розглянемо основні етапи реалізації функціоналу вебплатформи управління ветеринарною клінікою. Система складається з клієнтської частини, серверної частини та бази даних PostgreSQL. Клієнтська частина забезпечує взаємодію користувача з інтерфейсом. Серверна частина відповідає за обробку запитів, реалізацію бізнес-логіки, перевірку прав доступу та взаємодію з базою даних. База даних використовується для централізованого зберігання інформації про користувачів, власників тварин, пацієнтів, прийоми, медичні записи та інші об'єкти системи.

Першим етапом роботи із системою є авторизація користувача. Для цього користувач вводить логін і пароль, після чого дані передаються на сервер для перевірки. Сервер виконує автентифікацію облікового запису, перевіряє коректність введених даних та визначає роль користувача. Залежно від ролі система відкриває відповідний інтерфейс: адміністративну панель, кабінет лікаря або особистий кабінет клієнта. Реалізований механізм розмежування прав доступу дозволяє обмежити виконання окремих операцій лише для певних категорій користувачів, що підвищує рівень безпеки та забезпечує захист службової інформації. Приклад реалізації авторизації наведено в додатку А [1].

Після авторизації адміністратор отримує доступ до функціоналу управління користувачами. Даний модуль забезпечує централізоване адміністрування облікових записів системи. Адміністратор може переглядати список користувачів, створювати нові облікові записи, редагувати контактні дані, змінювати ролі та керувати статусом активності користувачів. Це дозволяє підтримувати актуальність інформації про персонал клініки та клієнтів, а також

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		52

контролювати доступ до функціональних можливостей вебплатформи. Приклад реалізації модуля управління користувачами наведено в додатку А[2].

Одним із важливих функціональних модулів системи є модуль обліку тварин. Він дає змогу зберігати інформацію про тварин, їх власників, породу, вид, дату народження, вагу, номер чіпу, алергії та хронічні захворювання. Завдяки цьому лікар має доступ до основної інформації про пацієнта перед проведенням прийому. Приклад реалізації функції додавання тварини наведено в додатку А[3].

Наступним етапом є реалізація запису тварини на прийом. Даний модуль забезпечує організацію взаємодії між клієнтами та лікарями ветеринарної клініки. Під час створення запису користувач обирає тварину, лікаря, дату та час прийому, а також визначає тип послуги. Після підтвердження дані зберігаються у базі даних та стають доступними для перегляду відповідним користувачам. Система враховує доступність лікаря згідно з його графіком роботи, що дозволяє уникати конфліктів під час бронювання часу. Додатково підтримується можливість організації онлайн-консультацій шляхом збереження посилання на зустріч. Приклад реалізації створення запису на прийом наведено в додатку А[4].

Після проведення прийому лікар має можливість створити медичний запис. У ньому зазначаються діагноз, план лікування, призначення, вакцинації, результати аналізів та інші дані, необхідні для подальшого лікування тварини. Усі записи пов'язані з конкретною твариною, лікарем та прийомом, що дозволяє формувати повну історію звернень пацієнта та забезпечує швидкий доступ до попередньої інформації під час наступних візитів. Приклад реалізації створення медичного запису наведено в додатку А[5].

Для організації робочого процесу клініки реалізовано модуль графіків лікарів. За його допомогою адміністратор може формувати розклад роботи персоналу, визначати робочі дні, години прийому та контролювати завантаженість спеціалістів. Додатково передбачено можливість блокування окремих часових проміжків у разі відпустки, тимчасової відсутності або інших обставин, що унеможливають проведення прийомів. Використання даного

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		53

модуля дозволяє оптимізувати процес планування та забезпечити рівномірний розподіл навантаження між лікарями. Приклад реалізації роботи з графіком лікаря наведено в додатку А[6].

Окремо реалізовано систему сповіщень. Сповіщення можуть надсилатися при створенні нового запису на прийом, зміні його статусу, появі необхідності залишити оцінку після завершення прийому або виконанні інших операцій. Кожне повідомлення містить тип сповіщення, заголовок, текст повідомлення, дату створення та ознаку прочитання. Наявність даного механізму сприяє покращенню взаємодії між користувачами та підвищує зручність роботи із системою. Приклад реалізації отримання та позначення сповіщень як прочитаних наведено в додатку А[7].

Після завершення прийому клієнт може залишити оцінку та коментар щодо якості наданих послуг. Для цього реалізовано модуль оцінювання прийомів, який дозволяє залишати числову оцінку та текстовий коментар. Відгук прив'язується до конкретного запису на прийом та зберігається в системі для подальшого аналізу. Отримані оцінки можуть використовуватися адміністрацією клініки для контролю рівня задоволеності клієнтів, виявлення проблемних аспектів обслуговування та підвищення якості надання ветеринарних послуг. Приклад реалізації додавання відгуку наведено в додатку А[8].

Підсумовуючи, у межах реалізації вебплатформи було створено діаграму компонентів (рис. 2.4), яка відображає структуру програмного забезпечення та взаємодію між його основними функціональними модулями. Архітектура системи побудована таким чином, щоб забезпечити розподіл функціональності між окремими компонентами, кожен з яких відповідає за виконання визначеного набору завдань.

До складу системи входить модуль авторизації та перевірки ролей користувачів, який забезпечує автентифікацію користувачів та розмежування прав доступу відповідно до їхніх ролей. Управління обліковими записами реалізується за допомогою модуля управління користувачами, який відповідає за створення, редагування та адміністрування користувачів системи. Для роботи з

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		54

клієнтською базою використовується окремий модуль управління власниками тварин, що забезпечує зберігання та обробку інформації про клієнтів ветеринарної клініки.

Функції ведення обліку пацієнтів реалізовані в модулі управління тваринами, який дозволяє зберігати та оновлювати інформацію про тварин. Організація прийомів здійснюється за допомогою модуля запису на прийом, що забезпечує створення, редагування та контроль записів. Для ведення історії лікування та зберігання медичної інформації використовується модуль медичних записів, який дозволяє формувати електронні картки пацієнтів та накопичувати інформацію про результати оглядів, діагнози й призначене лікування.

Планування роботи персоналу реалізується за допомогою модуля графіків роботи лікарів та модуля блокування часових проміжків. Їх спільне використання забезпечує формування актуального розкладу прийомів, врахування робочого часу лікарів та обмеження доступності окремих часових інтервалів у разі відпустки, лікарняного чи інших обставин. Для інформування користувачів про події в системі використовується модуль сповіщень, який забезпечує формування та доставку повідомлень про створення записів, зміну їх статусу та інші важливі події. Оцінювання якості наданих послуг реалізується за допомогою модуля оцінювання прийомів, який дозволяє збирати відгуки клієнтів та аналізувати рівень їх задоволеності.

Таким чином, реалізований функціонал забезпечує повний цикл роботи ветеринарної клініки, починаючи від авторизації користувача та управління даними про клієнтів і тварин, продовжуючи плануванням та проведенням прийомів, веденням електронної медичної документації й завершуючи надсиланням сповіщень та отриманням зворотного зв'язку від клієнтів. Взаємодія між усіма компонентами системи дозволяє автоматизувати основні бізнес-процеси ветеринарної клініки, підвищити ефективність роботи персоналу та покращити якість обслуговування власників тварин.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		55



Рис.2.4 Діаграма компонентів.

На діаграмі компонентів визначено основні складові вебплатформи та взаємозв'язки між ними. Центральним елементом архітектури є вебплатформа управління ветеринарною клінікою, яка об'єднує всі функціональні модулі системи та забезпечує взаємодію користувачів із програмним продуктом через єдиний інтерфейс доступу. Саме цей компонент координує обмін даними між окремими підсистемами та забезпечує виконання бізнес-процесів ветеринарної клініки.

Клієнтська частина системи реалізована за допомогою фреймворку Vue.js та призначена для взаємодії користувача із системою. Вона відповідає за відображення сторінок вебзастосунку, форм введення даних, особистих кабінетів користувачів, таблиць, календарів та інших елементів інтерфейсу. Крім того, клієнтська частина забезпечує надсилання запитів до серверної частини та відображення результатів їх виконання у зручному для користувача вигляді.

Серверна частина системи реалізована на основі платформи Node.js із використанням фреймворку Express. Вона виконує обробку запитів від клієнтської частини, реалізує механізми авторизації та автентифікації користувачів, здійснює перевірку прав доступу та забезпечує виконання бізнес-логіки системи. Сервер також відповідає за взаємодію з базою даних та формування відповідей на запити користувачів.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		56

Для зберігання інформації використовується база даних PostgreSQL, яка є окремим компонентом архітектури. У базі даних зберігаються відомості про користувачів, власників тварин, пацієнтів, записи на прийом, медичні картки, графіки роботи лікарів, системні сповіщення та оцінки прийомів. Використання централізованого сховища даних дозволяє забезпечити цілісність інформації та підтримувати актуальність даних у всіх функціональних модулях системи.

Одним із функціональних компонентів є модуль управління користувачами. Він забезпечує створення нових облікових записів, редагування персональної інформації, зміну ролей та статусів користувачів, а також контроль доступу до функціональних можливостей системи. Даний модуль використовується адміністраторами для управління користувачами вебплатформи.

Компонент обліку тварин та власників призначений для зберігання і обробки інформації про клієнтів клініки та їхніх тварин. Він забезпечує створення, редагування та перегляд відповідних записів, а також підтримує взаємозв'язок між власниками та зареєстрованими пацієнтами. Наявність такого модуля дозволяє швидко отримувати необхідну інформацію під час запису на прийом або ведення медичної документації.

Компонент запису на прийом відповідає за організацію процесу планування візитів до ветеринарної клініки. За його допомогою користувачі можуть створювати нові записи, переглядати інформацію про заплановані прийоми, змінювати параметри запису або скасовувати його. Робота цього компонента безпосередньо пов'язана з модулем графіків роботи лікарів, що дозволяє враховувати доступність спеціалістів під час вибору часу прийому.

Модуль медичних записів забезпечує ведення електронних карток пацієнтів. У ньому зберігається інформація про проведені огляди, встановлені діагнози, результати аналізів, призначене лікування, вакцинації та інші медичні відомості. Даний компонент відіграє важливу роль у формуванні повної історії лікування тварини та забезпечує швидкий доступ до необхідної інформації для лікаря.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		57

Компонент графіка роботи лікарів використовується для формування та управління робочим розкладом персоналу. За його допомогою визначаються робочі дні, години прийому та доступні часові інтервали для запису клієнтів. Додатково передбачено можливість блокування окремих часових проміжків у разі відпустки, лікарняного або інших причин тимчасової відсутності лікаря.

Компонент сповіщень та оцінювання забезпечує підтримку зворотного зв'язку між користувачами та системою. Він відповідає за надсилання повідомлень про створення або зміну записів на прийом, нагадування про майбутні візити та інші важливі події. Крім того, даний модуль дозволяє клієнтам залишати оцінки та відгуки після завершення прийому, що сприяє контролю якості надання ветеринарних послуг та підвищенню рівня задоволеності клієнтів.

Таким чином, діаграма компонентів демонструє структуру вебплатформи та відображає взаємодію між її основними складовими. Розподіл функціональності між окремими компонентами забезпечує модульність системи, спрощує її супровід та створює можливість подальшого розширення функціональних можливостей без суттєвих змін існуючої архітектури.

Висновки до другого розділу

У процесі проєктування вебплатформи управління ветеринарною клінікою були проаналізовані функціональні вимоги до програмного продукту та визначені основні варіанти його використання. На основі проведеного аналізу було розроблено архітектуру системи, що забезпечує взаємодію між клієнтською частиною, серверною частиною та базою даних.

У межах проєктування було побудовано діаграму варіантів використання, діаграму класів, діаграму активності та діаграму компонентів, що дозволило детально описати структуру системи, взаємозв'язки між її складовими та логіку роботи основних функціональних модулів.

Також було спроектовано реляційну базу даних PostgreSQL, яка забезпечує зберігання інформації про користувачів, власників тварин, тварин, записи на прийом, медичні картки, графіки роботи лікарів, сповіщення та оцінки прийомів.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		58

Розроблена структура бази даних дозволяє забезпечити цілісність даних, швидкий доступ до інформації та ефективну взаємодію між окремими підсистемами програмного продукту.

Було описано алгоритм роботи системи та визначено послідовність дій користувачів залежно від їх ролей. У системі передбачено три основні ролі: адміністратор, лікар та клієнт. Для кожної ролі визначено набір функціональних можливостей та прав доступу, що забезпечує безпечне використання вебплатформи та розмежування відповідальності між користувачами.

Крім того, було розглянуто реалізацію основного функціоналу системи, зокрема авторизацію користувачів, управління обліковими записами, облік тварин та їх власників, створення записів на прийом, ведення медичних карток, управління графіками роботи лікарів, надсилення сповіщень та оцінювання якості наданих послуг. Реалізовані функціональні модулі забезпечують повний цикл роботи ветеринарної клініки та створюють основу для подальшої реалізації програмного продукту.

Отримані результати проектування стали основою для практичної реалізації вебплатформи та дозволили сформуванню чіткої структури системи, яка відповідає поставленим вимогам і забезпечує ефективну автоматизацію діяльності ветеринарної клініки.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		59

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБПЛАТФОРМОЮ УПРАВЛІННЯ ВЕТЕРИНАРНОЮ КЛІНІКОЮ

3.1 Порядок встановлення та налаштування параметрів вебплатформи

Встановлення та налаштування вебплатформи управління ветеринарною клінікою передбачає розгортання клієнтської частини, серверної частини та бази даних, а також налаштування взаємодії між усіма компонентами системи.

Для опису архітектури програмного продукту було створено діаграму розгортання (рис. 3.1).

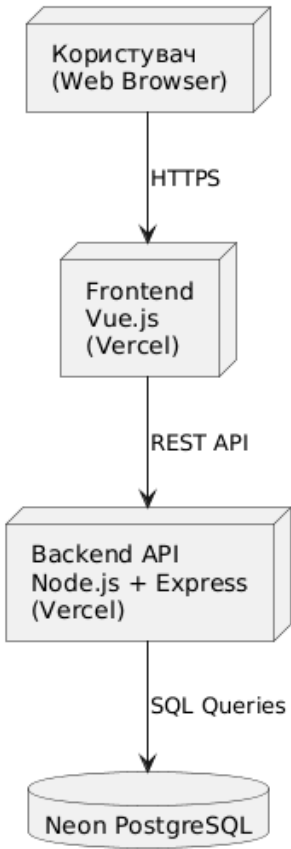


Рис.3.1 Діаграма розгортання

Frontend Application – клієнтська частина програмного продукту, реалізована за допомогою фреймворку Vue.js. Вона відповідає за взаємодію користувачів із системою, відображення даних, формування інтерфейсу та

надсилання запитів до серверної частини. Клієнтська частина розгорнута на хмарній платформі Vercel.

Backend API – серверна частина системи, реалізована з використанням Node.js та Express. Вона забезпечує обробку HTTP-запитів, авторизацію користувачів, перевірку прав доступу, реалізацію бізнес-логіки та взаємодію з базою даних.

Серверна частина побудована за модульним принципом та включає компоненти, що забезпечують автентифікацію користувачів, управління обліковими записами, роботу з даними про власників і тварин, обробку записів на прийом, ведення медичної документації, формування графіків роботи лікарів, надсилання сповіщень та зберігання оцінок клієнтів. Модулі взаємодіють між собою через внутрішні програмні інтерфейси та використовують спільну базу даних PostgreSQL для зберігання інформації. Така структура спрощує супровід програмного забезпечення та дозволяє розширювати функціональні можливості системи без зміни її загальної архітектури. Серверна частина також розгорнута на платформі Vercel та взаємодіє з клієнтською частиною за допомогою REST API.

Neon PostgreSQL Database – хмарна база даних PostgreSQL, що розгорнута на платформі Neon. Вона забезпечує зберігання та обробку інформації про користувачів, власників тварин, тварин, записи на прийом, медичні картки, графіки роботи лікарів, сповіщення та оцінки прийомів. Використання Neon дозволяє забезпечити надійне зберігання даних, автоматичне резервування та доступ до бази даних через мережу Інтернет.

Під час розгортання вебплатформи особлива увага приділяється налаштуванню взаємодії між усіма компонентами системи. Після запуску клієнтської та серверної частин виконується перевірка коректності мережевої взаємодії між ними.

Після налаштування взаємодії між клієнтською та серверною частинами виконується підключення серверного застосунку до бази даних. На цьому етапі перевіряється коректність параметрів з'єднання, доступність бази даних та

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		61

можливість виконання операцій читання і запису даних. Успішне встановлення з'єднання забезпечує можливість централізованого зберігання інформації про користувачів, тварин, прийоми та медичні записи.

Після завершення налаштування системи проводиться перевірка працездатності основних функціональних модулів. Крім того, перевіряється коректність збереження та отримання інформації з бази даних, а також правильність відображення даних у користувацькому інтерфейсі.

Окремо перевіряється механізм обробки помилок та виняткових ситуацій. У випадку введення некоректних даних або спроби виконання забороненої операції система формує відповідні повідомлення та не допускає порушення цілісності даних. Це дозволяє підвищити надійність роботи вебплатформи та забезпечити стабільне функціонування всіх її компонентів.

Завершальним етапом розгортання є введення системи в експлуатацію. Після успішного проходження всіх перевірок вебплатформа стає доступною користувачам через мережу Інтернет. Подальша робота системи не потребує встановлення додаткового програмного забезпечення на робочих місцях користувачів, оскільки доступ до всіх функцій здійснюється за допомогою веббраузера.

Таким чином, архітектура вебплатформи побудована за трирівневою моделлю «клієнт – сервер – база даних», що забезпечує масштабованість, зручність супроводу та безпечне зберігання інформації. Розділення системи на окремі логічні рівні дозволяє незалежно модернізувати клієнтську та серверну частини, спрощує підтримку програмного забезпечення та забезпечує можливість подальшого розширення функціоналу без суттєвих змін існуючої архітектури. Використання хмарної інфраструктури також сприяє підвищенню доступності системи, спрощує процес оновлення програмного забезпечення та забезпечує централізоване керування всіма компонентами вебплатформи.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		62

3.2 Структура інтерфейсу вебплатформи управління ветеринарною клінікою

Інтерфейс користувача є важливою складовою вебплатформи управління ветеринарною клінікою, оскільки саме через нього здійснюється взаємодія користувачів із функціональними можливостями системи. Під час проєктування інтерфейсу основна увага приділялася забезпеченню зручності використання, зрозумілості навігації, швидкому доступу до необхідної інформації та адаптивності для різних пристроїв.

Після завантаження користувач опиняється на сторінці головного меню зображеного на рисунку 3.2.

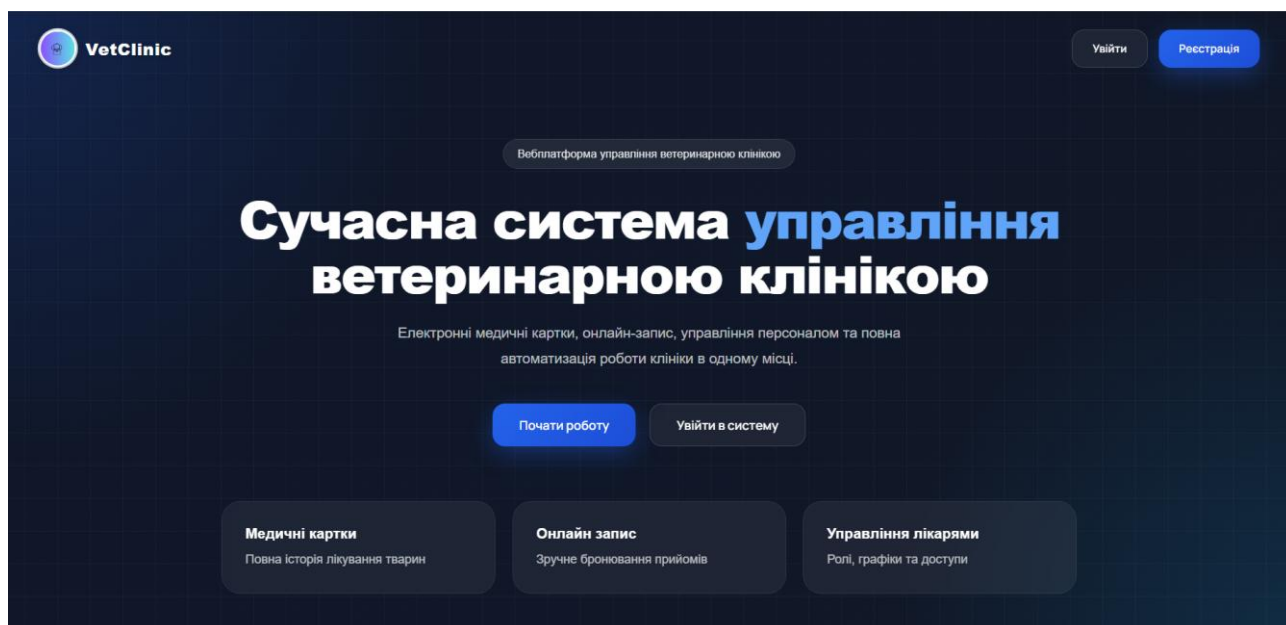


Рис.3.2 – Головна сторінка

Далі користувач натискає на кнопку “Увійти в систему”, якщо користувач уже зареєстрований в системі, то він може авторизуватися (рис.3.3), якщо ні – то під формою для входу є посилання на сторінку реєстрації (рис 3.4), де користувач може створити свій профіль.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		63

VetClinic
Система управління клінікою

Вхід до системи

Увійдіть у свій акаунт для продовження роботи

Логін

Введіть логін

Пароль

Введіть пароль

Увійти

або

Увійти через Google

Демо-акаунти (пароль: password123)

admin doctor1 owner1

Немає акаунту? [Зареєструватися](#)

Рис.3.3 – Сторінка авторизації

VetClinic
Створення нового акаунта

Реєстрація

Створіть обліковий запис власника тварини

ПІБ *

Іванов Іван Іванович

Логін *

ivanov_ivan

Email Телефон

email@example.com +380671234567

Пароль *

Мінімум 6 символів

Підтвердження пароля *

Повторіть пароль

Зареєструватися

Вже є акаунт? [Увійти](#)

Рис.3.4 – Сторінка реєстрації

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		64

Після авторизації користувач потрапляє до особистого кабінету, він містить основну інформацію про користувача та його активність у системі. Тут відображаються дані профілю, заплановані прийоми та актуальні повідомлення.(рис.3.5)

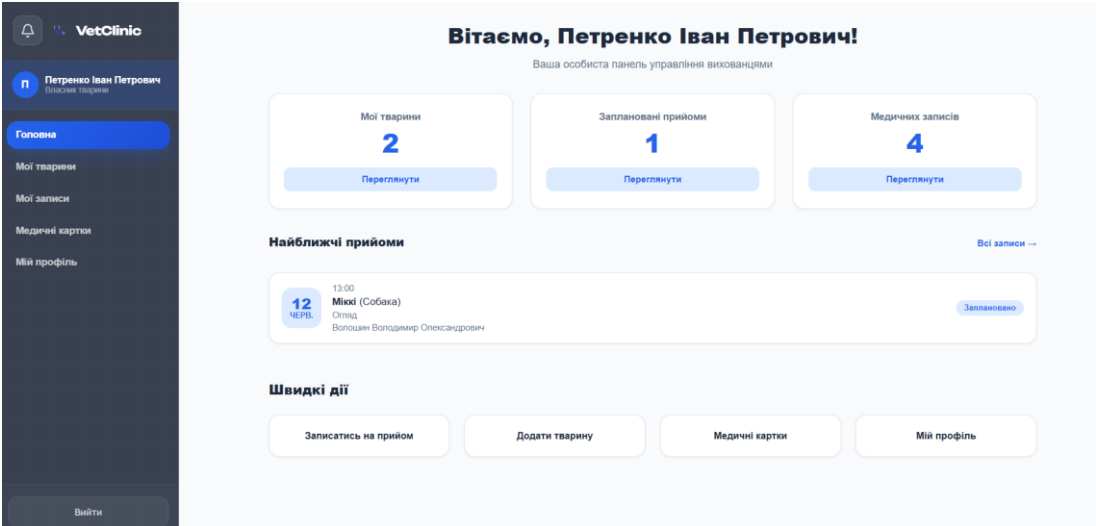


Рис.3.5 – Особистий кабінет користувача

Далі користувач може перейти до сторінки Мої тварини, даний розділ призначений для перегляду інформації про зареєстрованих тварин. Користувач може переглядати характеристики своїх улюбленців та їхню медичну історію (рис.3.6).

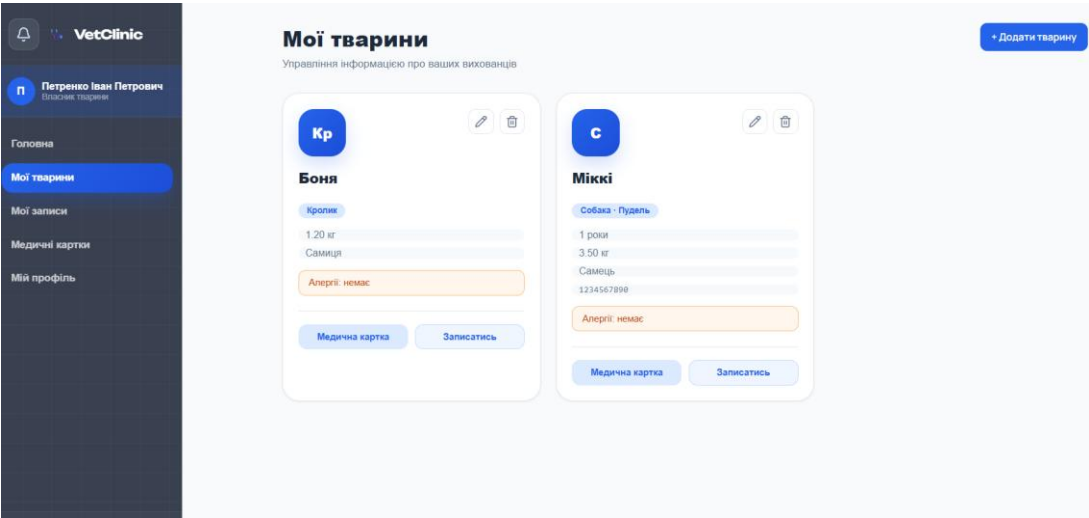


Рис.3.6 – Сторінка мої тварини

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		65

Наступна функція – це запис на прийом, вона дозволяє обрати ветеринарного лікаря, дату та час візиту. Після підтвердження дані автоматично зберігаються в системі та відображаються в розкладі клініки (рис.3.7). Також користувач може переглядати усі минулі та майбутні записи.(рис.3.8)

Рис.3.7 – Запис на прийом

Рис.3.8 – Записи користувача

Наступна сторінка забезпечує доступ до електронної медичної картки тварини. Користувач може переглядати інформацію про проведені огляди, встановлені діагнози та призначене лікування. (рис.3.9). Є можливість фільтрації за датою запису та кличкою тварини.

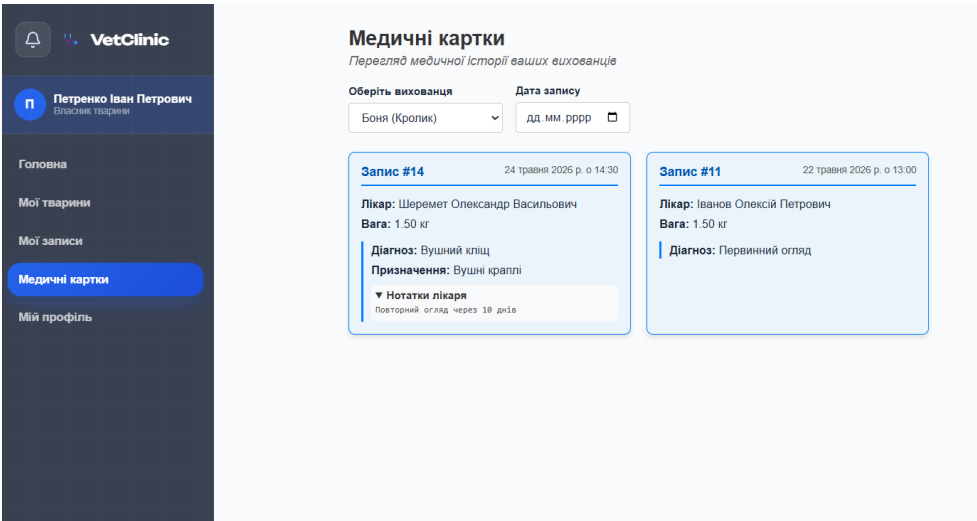


Рис.3.9 – Медичні картки тварини

Система сповіщень інформує клієнта про майбутні прийоми, зміни у розкладі або оновлення медичних записів. Це дозволяє своєчасно отримувати важливу інформацію та підтримувати зв’язок із клінікою.(рис.3.10)

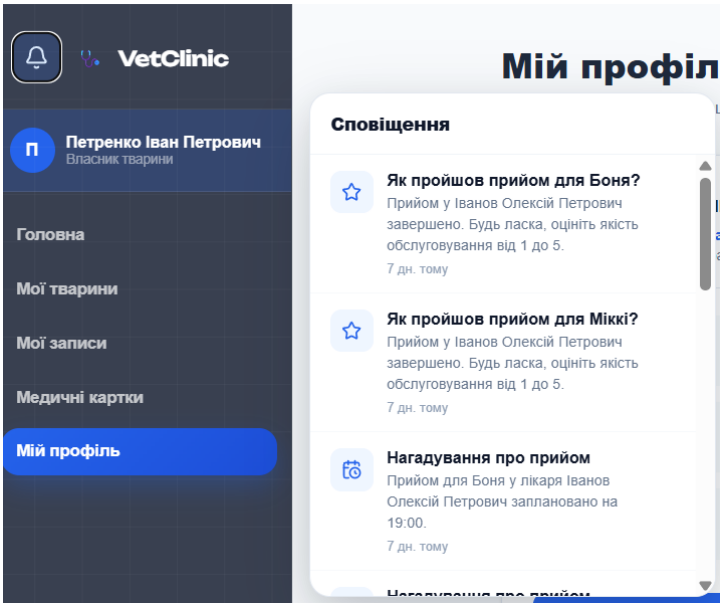


Рис.3.10 – Сповіщення

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		67

Також у користувача є можливість редагувати профіль та переглядати статистику (рис.3.11).

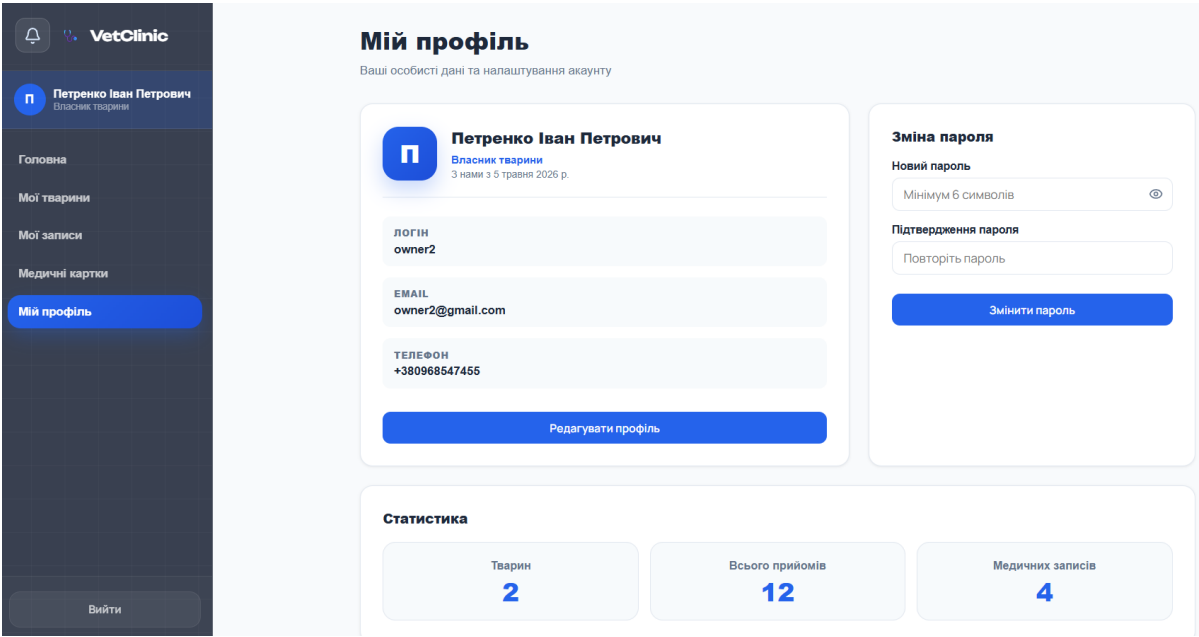


Рис.3.11 – Особистий профіль користувача

Далі розглянемо роль лікаря, інформаційна панель містить коротку інформацію про заплановані прийоми, кількість пацієнтів та актуальні повідомлення. Це дозволяє швидко отримати доступ до необхідних даних для роботи (рис.3.12).

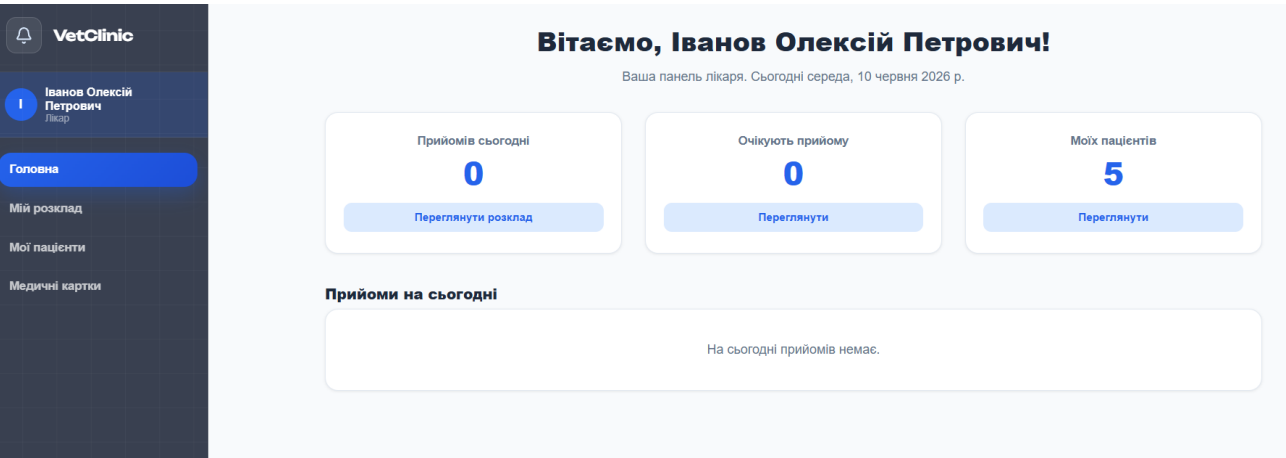


Рис.3.12 – Інформаційна панель лікаря

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		68

Наступним є модуль розкладу, який забезпечує перегляд запланованих прийомів ветеринарного лікаря. На сторінці відображаються дата, час, інформація про пацієнта та причина звернення (рис.3.13). Це спрощує планування робочого процесу.

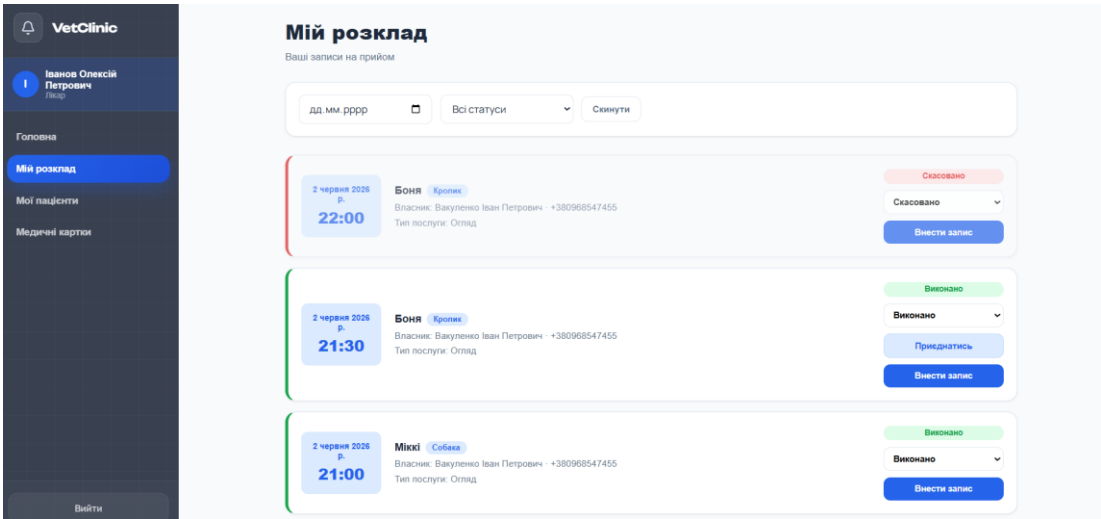


Рис.3.13 – Розклад лікаря

Сторінка пацієнтів містить перелік тварин, які закріплені за лікарем або мають заплановані прийоми. Лікар може швидко знайти необхідного пацієнта та перейти до його медичної картки (рис.3.14).

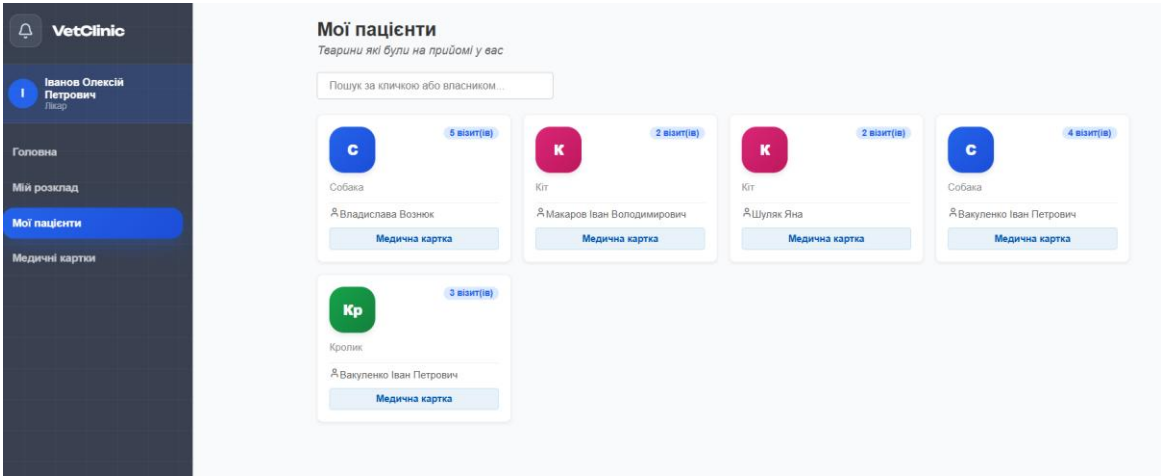


Рис.3.14 – Пацієнти лікаря

Модуль медичних карток забезпечує доступ до історії лікування тварин. Лікар може переглядати попередні діагнози, призначення, результати оглядів та іншу медичну інформацію (рис.3.15).

Медичні картки
Перегляд та ведення медичних записів пацієнтів

Оберіть пацієнта
Рудий (Кіт) — Макаров Іван Володимирович

Дата запису
дд. мм. рррр

+ Новий запис

Запис #122 березня 2026 р. о 23:58

Вага при прийомі: 4.10 кг

Діагноз: Вушний кліщ

План лікування: У вуха вносять гігієнічний лосьйон обсягом 0,5-1 мл. Вушні раковини акуратно складаються і м'яко масажуються біля основи близько 1 хвилини. Тварину відпускають і дозволяють їй потрясти вухами 10-15 хвилин.

Призначення: Ауроکلін

Вакцинації: Від сказу

► Результати аналізів

► Нотатки

Редагувати

Рис.3.15 – Медичні картки

Функціонал дозволяє створювати нові записи про огляди пацієнтів. Під час прийому лікар вносить результати обстеження, встановлений діагноз та призначене лікування (рис.3.16).

Новий медичний запис

Дата прийому
22 березня 2026 р. о 20:13 — Діт

Вага (кг)
4

Діагноз
Введіть діагноз...

План лікування
Опишіть план лікування...

Призначення / Ліки
Амоксицилін 250мг 2р/день...

Вакцинації
Rabies, DHPP...

Результати аналізів
Загальний аналіз крові: в нормі...

Нотатки
Додаткові нотатки...

Рис.3.16 – Створення нового запису

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				70
Змн.	Арк.	№ докум.	Підпис	Дата		

Система сповіщень інформує лікаря про нові записи на прийом, зміни в розкладі та інші події, що потребують уваги. Це забезпечує оперативність роботи та своєчасне реагування на зміни (рис.3.17).

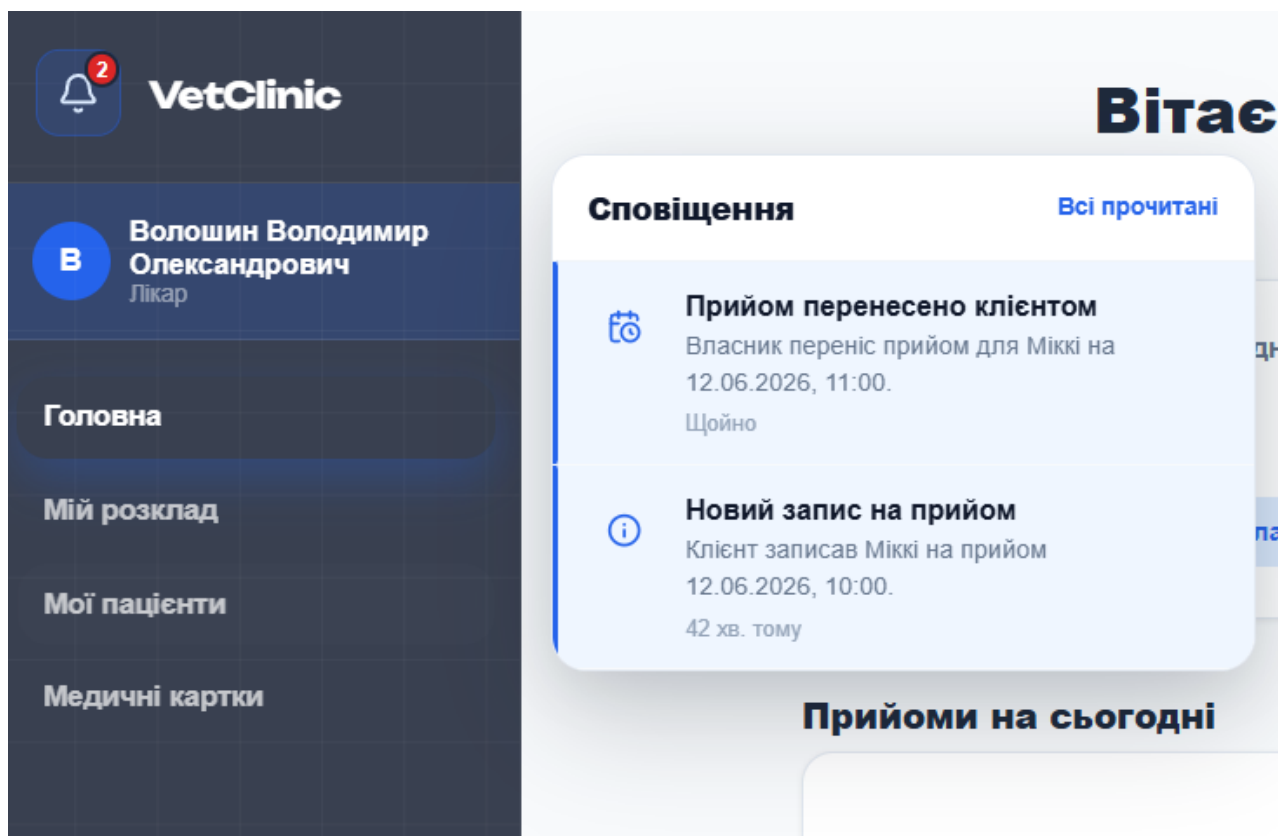


Рис.3.17 – Сповіщення

3.3 Інтерфейс для адміністратора вебплатформи

Розглянемо роль адміністратора, для нього передбачено інтерфейс із розширеними можливостями управління системою. Після успішної перевірки даних система виконує авторизацію та перенаправляє користувача до адміністративної панелі (рис.3.18 – 3.19). Інформаційна панель адміністратора відображає узагальнену інформацію про роботу ветеринарної клініки. На сторінці представлено основні статистичні показники, зокрема кількість користувачів, тварин, записів на прийом та медичних карток. Також є діаграми популярних послуг, статистика за останні 6 місяців, статуси прийомів та рейтинг

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		71

лікарів. Даний модуль дозволяє швидко оцінити поточний стан системи.

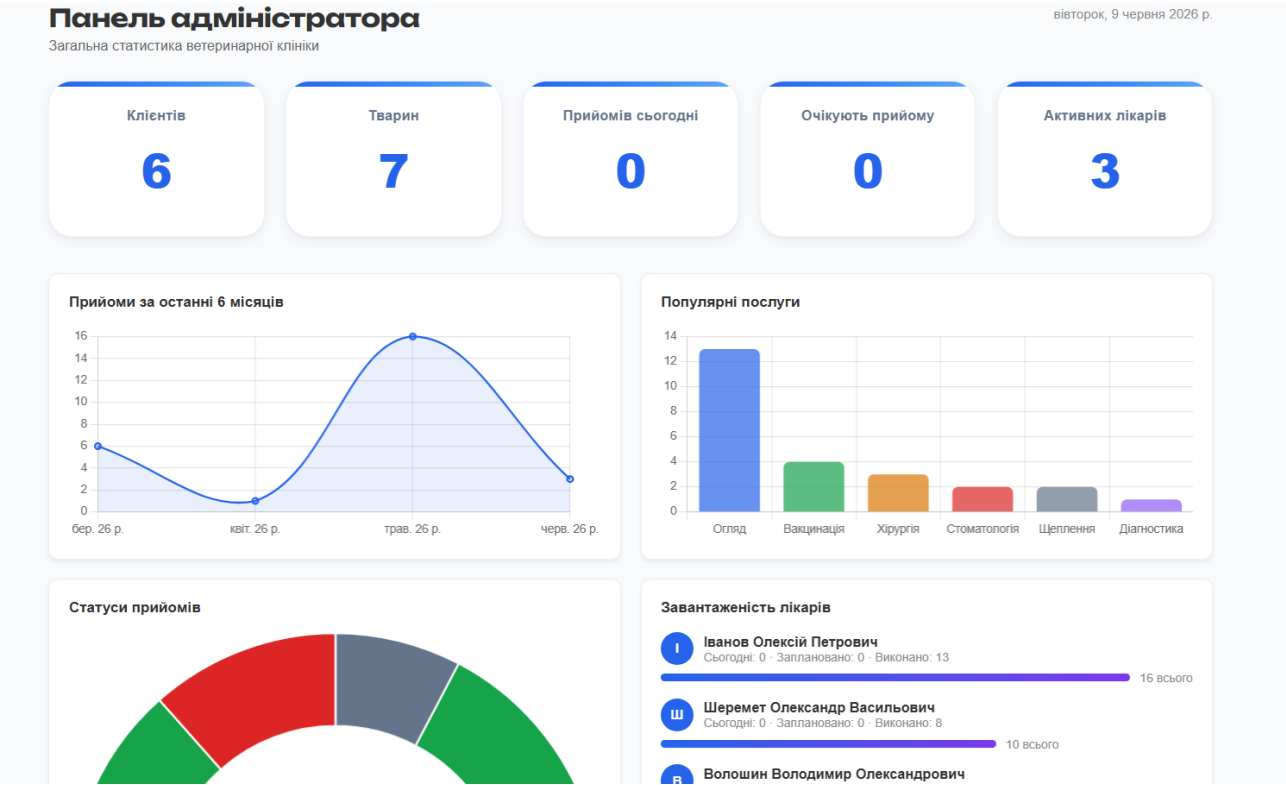


Рис.3.18 – Статистика ветеринарної клініки

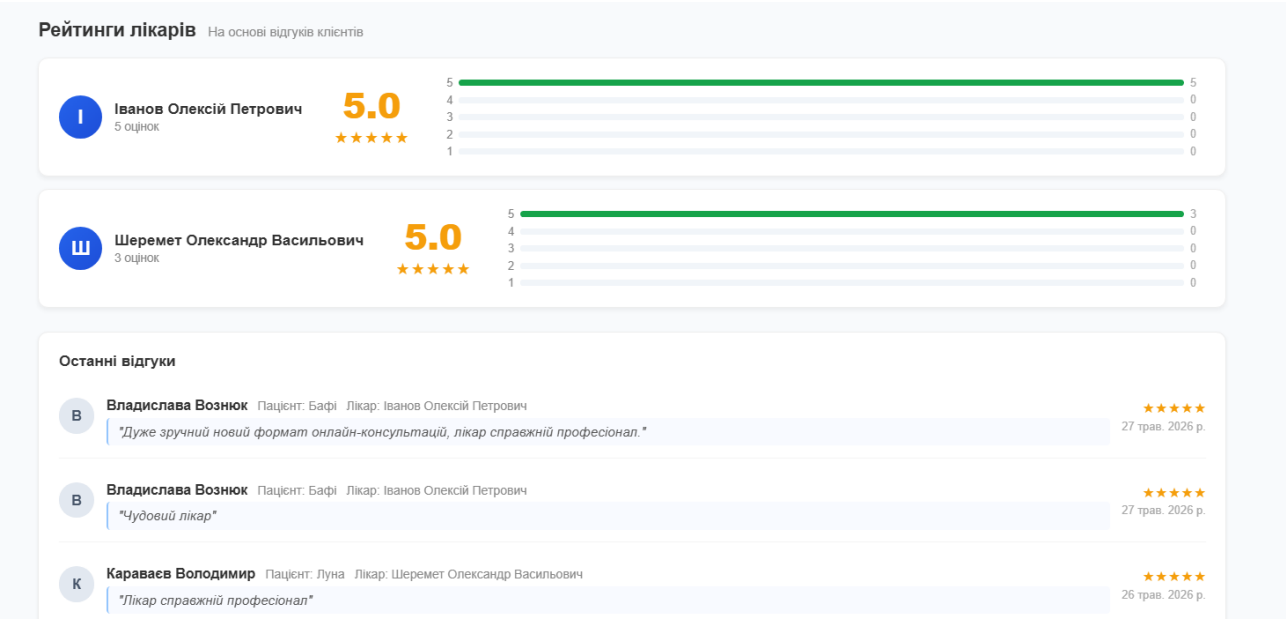


Рис.3.19 – Статистика ветеринарної клініки

Далі переходимо до управління користувачами, адміністратор може переглядати список користувачів, створювати нові облікові записи, редагувати персональні дані(ПІБ, логін, електронну пошту, номер телефону, роль, пароль) та змінювати ролі користувачів. Також передбачено можливість пошуку та сортування за алфавітом, ролями користувачів (рис.3.20).

Управління користувачами						
Створення, редагування та видалення акаунтів						
Пошук за іменем або логіном... А → Я Всі Адміни Лікарі Власники						
КОРИСТУВАЧ	ЛОГІН	РОЛЬ	КОНТАКТИ	СТАТУС	ДАТА РЕЄСТРАЦІЇ	ДІЇ
A	admin	Адмін	admin@vetclinic.ua +380000000000	Активний	16.03.2026	  
B	google_112323894278610619410	Власник	vt221_vvyu@student.ztu.edu.ua	Активний	22.03.2026	  
B	volosh_doc	Лікар	volosh@gmail.com +380501236985	Активний	07.06.2026	  
I	doctor1	Лікар	doctor1@vetclinic.ua +380671234567	Активний	16.03.2026	  
K	karav@gmail.com	Власник	karav@gmail.com	Активний	26.05.2026	  
M	mak_ivan	Власник	ivan@gmail.com +380985236589	Активний	16.03.2026	  
M	mark	Власник	mark@gmail.com +380967458213	Активний	16.03.2026	  
P	owner2	Власник	owner2@gmail.com +380968547458	Активний	05.05.2026	  

Рис.3.20 – Управління користувачами

Наступна функція – це управління персоналом, адміністратор може додати працівника, редагувати інформацію про вже наявного (ПІБ, логін, роль, номер телефону, пароль та тип послуги, які надає лікар), блокувати та видаляти з системи. Реалізована можливість пошуку за іменем або електронною поштою, також доступна коротка статистика про кожного лікаря, а саме кількість прийомів на сьогодні, запланованих та загальна кількість проведених (рис.3.21).

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		73

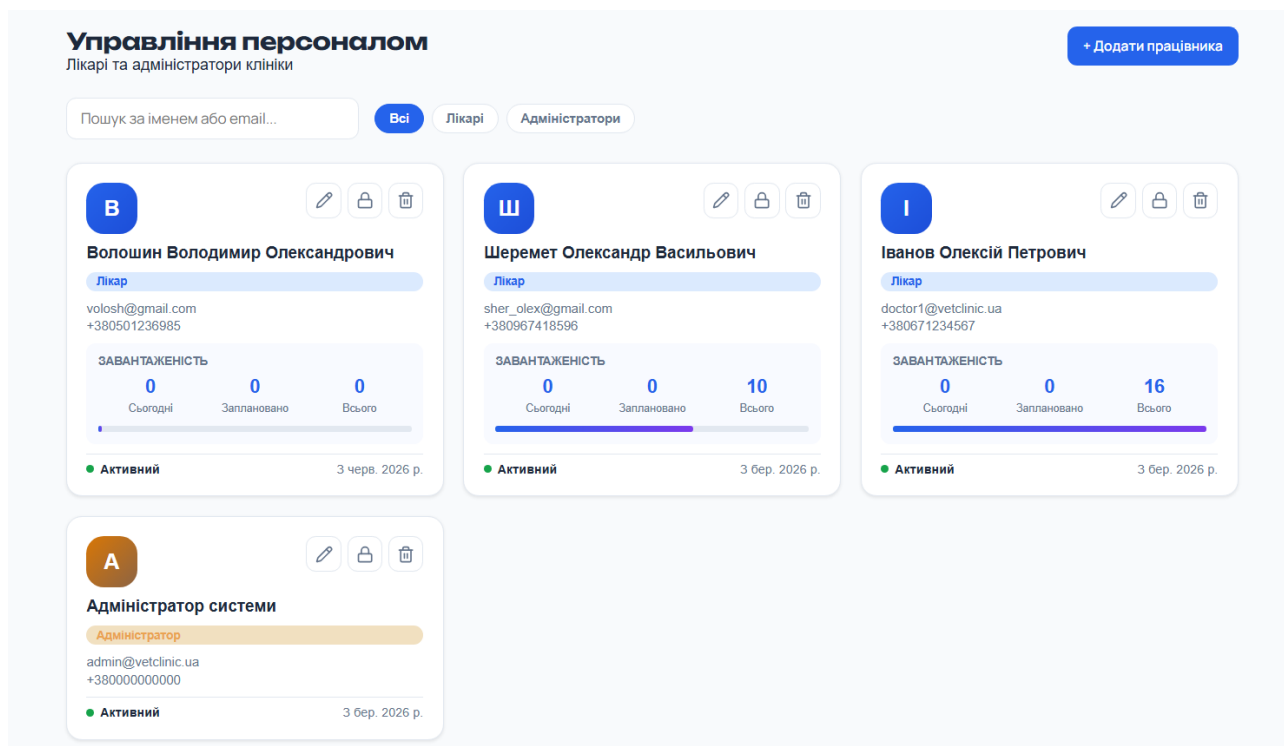


Рис.3.21 – Управління персоналом

Далі розглянемо сторінку розкладу, яка призначена для формування та управління графіком роботи ветеринарних лікарів. Даний модуль дозволяє адміністратору створювати та коригувати розклад роботи персоналу, визначати робочі дні та години прийому, а також вносити зміни у разі зміни графіка. Крім того, передбачена можливість блокування окремих часових проміжків для обідніх перерв, відпусток, службових заходів або інших причин відсутності працівника. Система автоматично враховує встановлені обмеження під час запису клієнтів на прийом, що дозволяє уникнути накладання записів та перевантаження лікарів. Адміністратор може переглядати кількість доступних слотів для запису, оцінювати поточну завантаженість персоналу та планувати робочий час відповідно до потреб ветеринарної клініки, забезпечуючи рівномірний розподіл навантаження між працівниками (рис. 3.22).

Керування розкладом лікарів

Графік роботи, доступні години та блокування

Графік роботи

Доступні слоти

Блокування

Навантаження

Волошин Володимир Олександрович

Зберегти графік

Понеділок

Початок 09:00

Кінець 18:00

18 слотів по 30 хв

Вівторок

Вихідний

Середа

Вихідний

Четвер

Вихідний

П'ятниця

Початок 09:00

Кінець 18:00

18 слотів по 30 хв

Субота

Вихідний

Неділя

Вихідний

Рис.3.22 – Керування розкладом лікаря

Наступний модуль забезпечує управління розкладом прийомів ветеринарної клініки. Адміністратор має доступ до інтерактивного календаря, у якому відображаються всі заплановані візити. Для зручності роботи передбачено декілька режимів перегляду розкладу: за місяць, тиждень або день. Це дозволяє оперативно оцінювати завантаженість лікарів, контролювати кількість записів та планувати робочий процес клініки. У разі необхідності адміністратор може змінювати дату та час прийому, переносити запис на інший часовий інтервал або скасовувати його. Всі внесені зміни автоматично відображаються в системі та стають доступними іншим користувачам відповідно до їхніх прав доступу (рис. 3.23).

Розклад прийомів

Календар записів на прийом

+ Новий запис

Заплановано

В процесі

Виконано

Скасовано

Не з'явився

<

>

Сьогодні

25 – 31 трав. 2026 р.

Місяць

Тиждень

День

	ПН 25.05	ВТ 26.05	СР 27.05	ЧТ 28.05	ПТ 29.05	СБ 30.05	НД 31.05
8							
9							
10			10:00 - 10:30 - Бадя — Увага				
11							
12							
13							
14				14:00 - 14:30 - Бадя — Скасовано			
15							
16			16:00 - 16:30 - Бадя — Огляд				
17			17:30 - 18:00 - Бадя — Огляд				
18							

Рис.3.23 – Розклад прийомів

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		75

Окремий модуль призначений для роботи з переліком записів на прийом. Він дозволяє адміністратору переглядати як майбутні, так і завершені прийоми, контролювати їхній статус та створювати нові записи для клієнтів клініки. Для швидкого пошуку необхідної інформації реалізовано систему фільтрації, яка дає змогу відбирати записи за датою проведення, статусом прийому (запланований, завершений, скасований) та іншими параметрами. Крім того, передбачено пошук за кличкою тварини або даними її власника, що значно спрощує роботу з великим обсягом інформації та підвищує ефективність адміністрування записів (рис. 3.24)

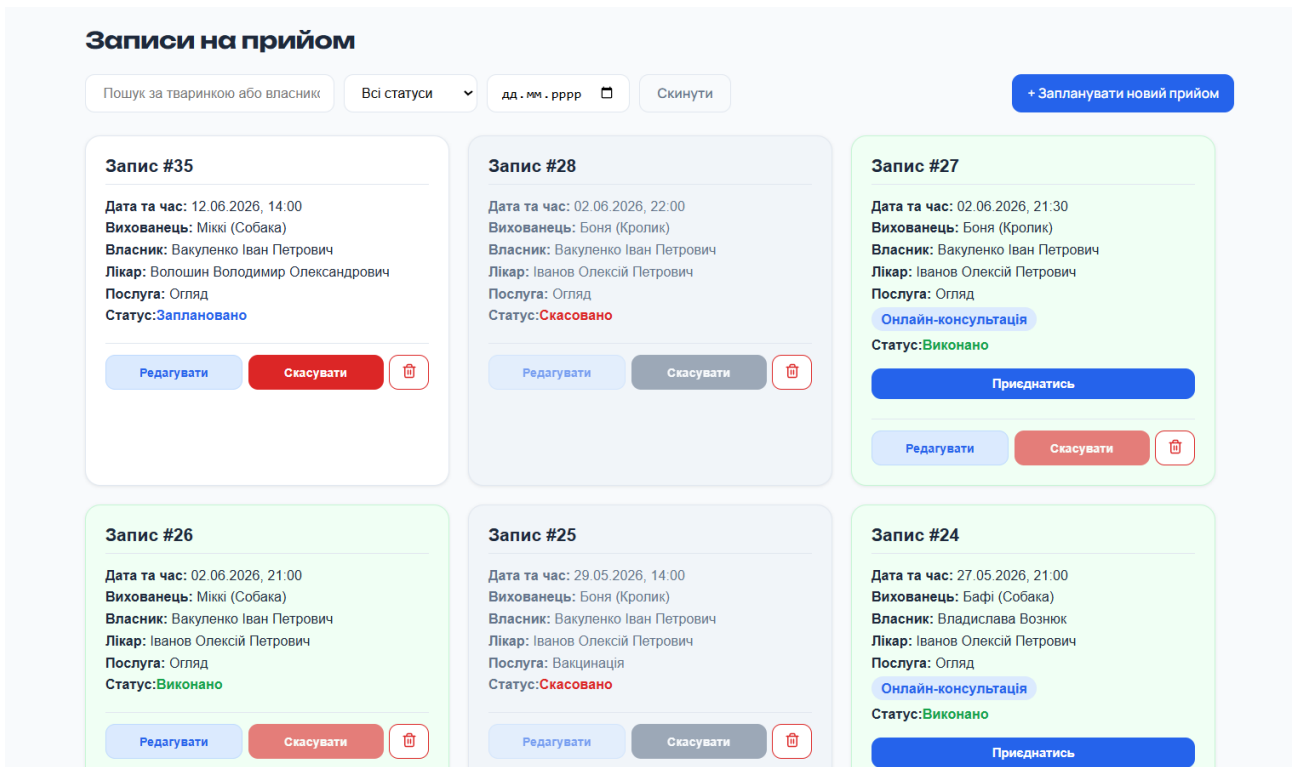


Рис.3.24 – Записи на прийом

Модуль управління тваринами призначений для ведення єдиної бази пацієнтів клініки та забезпечує зручний доступ до інформації про всіх зареєстрованих тварин. Для кожного пацієнта в системі зберігаються основні відомості, зокрема кличка, вид, порода, дата народження, стать та дані власника. Адміністратор має можливість додавати нових тварин до системи, редагувати існуючі записи та контролювати актуальність інформації. Для спрощення роботи

з великою кількістю записів реалізовано механізми пошуку та фільтрації. Зокрема, передбачено фільтрацію за видом тварини, а також пошук за кличкою пацієнта або даними його власника. Такий підхід дозволяє швидко знаходити необхідну інформацію та підтримувати цілісність бази даних ветеринарної клініки (рис. 3.25).

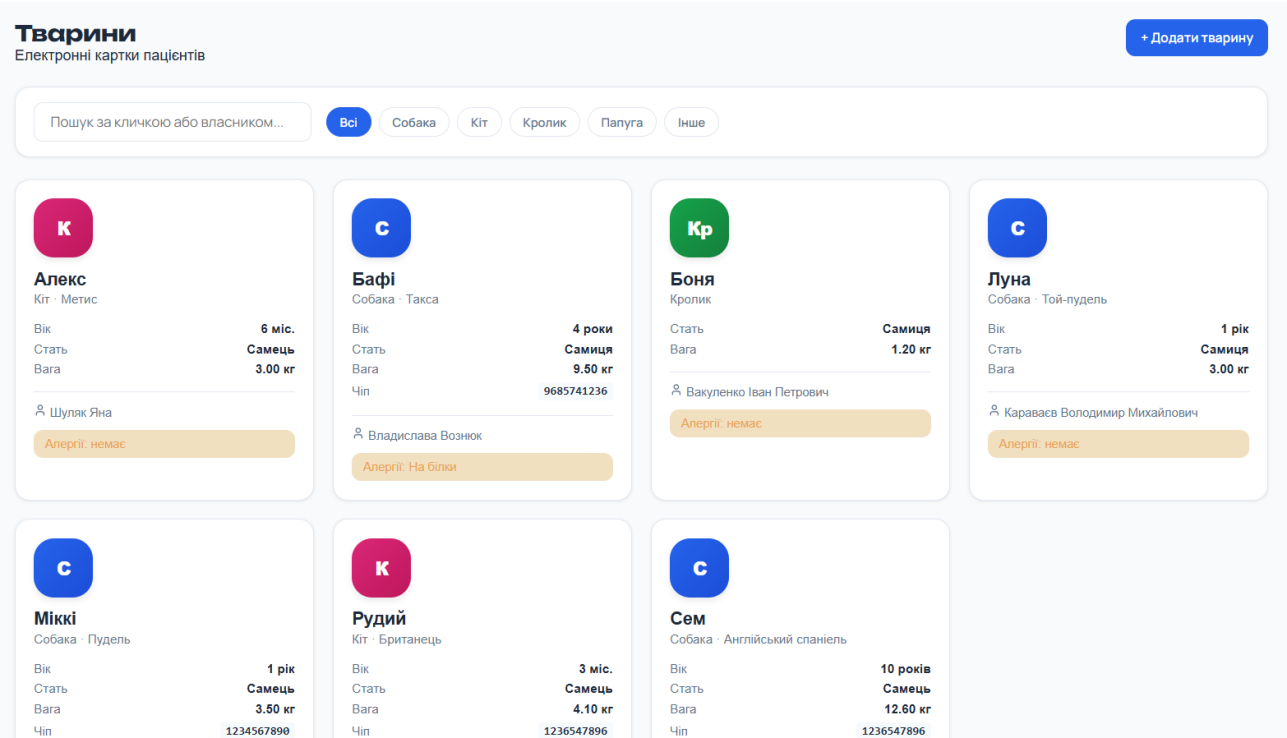


Рис.3.25 – Управління тваринами

Сторінка управління медичними записами забезпечує доступ до електронних медичних карток пацієнтів та є одним із ключових елементів інформаційної системи. У даному модулі зберігається історія звернень тварин до клініки, результати проведених оглядів, встановлені діагнози, призначене лікування, рекомендації лікарів та інша медична інформація. Адміністратор має можливість переглядати наявні записи, вносити зміни до існуючих карток або створювати нові записи у разі необхідності. Для швидкого доступу до потрібної інформації реалізовано механізми пошуку та фільтрації за датою прийому, конкретною твариною або її власником. Використання електронних медичних карток дозволяє забезпечити централізоване зберігання інформації про

пацієнтів, підвищити швидкість обробки даних та спростити доступ до історії лікування тварин для працівників клініки (рис. 3.26)

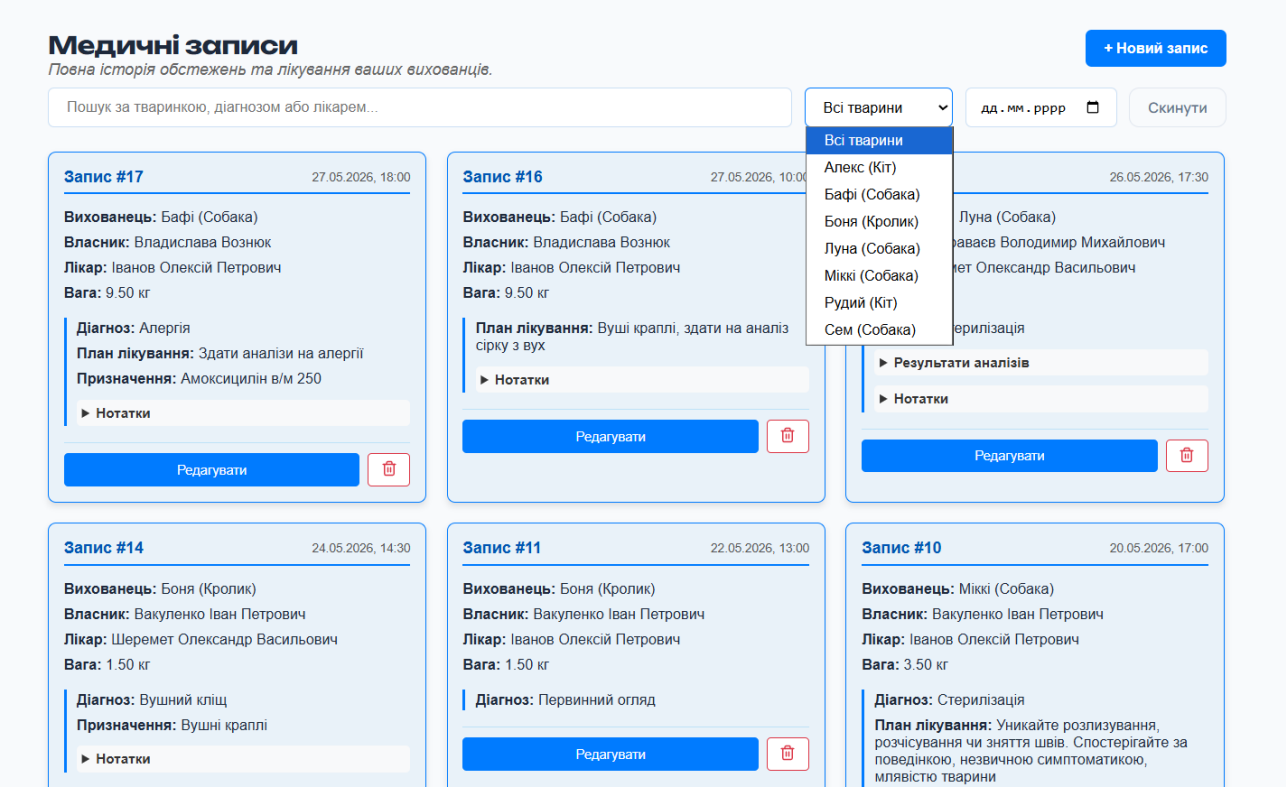


Рис.3.26 – Медичні записи

Сторінка управління клієнтами призначена для централізованого ведення бази власників тварин. На даній сторінці адміністратор може переглядати контактні дані клієнтів, редагувати інформацію про них та контролювати актуальність відомостей, що зберігаються в системі. Для зручності роботи реалізовано механізм пошуку та фільтрації, який дозволяє швидко знаходити потрібного клієнта за прізвищем, ім'ям, адресою електронної пошти або номером телефону. Наявність єдиної клієнтської бази забезпечує впорядковане зберігання інформації, спрощує взаємодію персоналу з відвідувачами клініки та сприяє підвищенню якості обслуговування клієнтів (рис. 3.27).

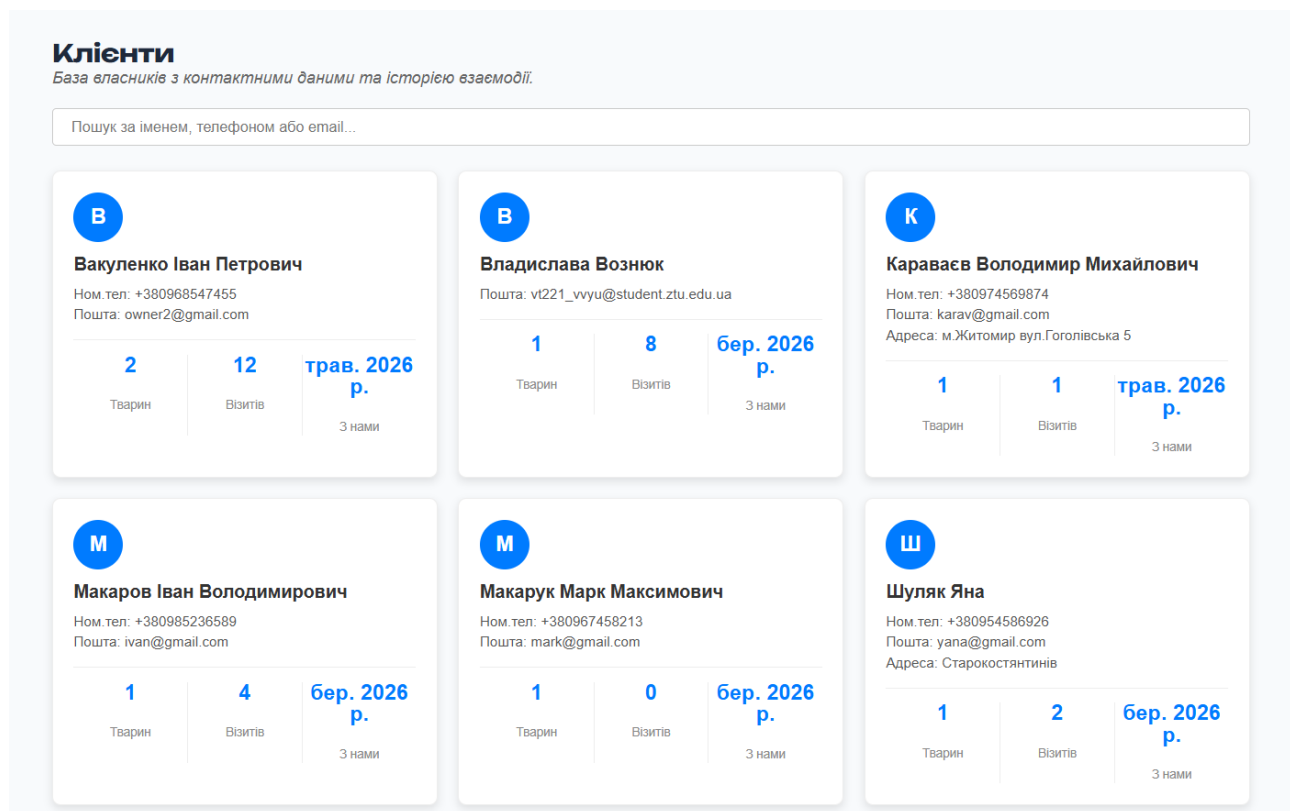


Рис.3.27 – База клієнтів

3.4 Тестування вебплатформи

Тестування вебплатформи управління ветеринарною клінікою є важливим етапом розробки програмного продукту, оскільки дозволяє перевірити коректність реалізації функціональних можливостей системи, виявити можливі помилки та оцінити стабільність роботи програмного забезпечення. Основною метою тестування є перевірка відповідності фактичних результатів роботи системи очікуваним результатам, а також підтвердження правильності реалізації основних бізнес-процесів ветеринарної клініки.

У процесі тестування було перевірено роботу механізмів авторизації користувачів, управління тваринами, створення записів на прийом, ведення медичних карток, а також функціонування адміністративної панелі. Для кожного сценарію було визначено вхідні дані, очікуваний результат та проведено порівняння з фактичним результатом роботи системи.

Опис тестів наведено в таблицях 3.1 – 3.10.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		79

Таблиця 3.1

Опис тесту «Авторизація користувача»

Кроки для перевірки	Ввести електронну пошту та пароль і натиснути кнопку входу
Вхідні дані	Зареєстрований користувач та коректний пароль
Очікуваний результат	Користувача авторизовано та перенаправлено до особистого кабінету
Фактичний результат	Відповідає

Для перевірки надійності механізму захисту також протестовано сценарій введення некоректних облікових даних. Таблиця 3.2 відображає очікувану реакцію системи на помилковий логін або пароль.

Таблиця 3.2

Опис тесту «Авторизація з некоректним паролем або логіном»

Кроки для перевірки	Ввести неправильну електронну пошту та пароль і натиснути кнопку входу
Вхідні дані	Зареєстрований користувач та некоректний пароль
Очікуваний результат	Відображається повідомлення про помилку авторизації
Фактичний результат	Відповідає

Після перевірки авторизації протестовано функціонал управління тваринами. Таблиця 3.3 описує тест додавання нової тварини до профілю клієнта через відповідну форму. У ході тестування перевірялась коректність збереження введених даних, зокрема імені тварини, виду, породи, віку та інших характеристик. Також було перевірено роботу валідації полів форми – система належним чином реагувала на спроби збереження запису з незаповненими обов'язковими полями, відображаючи відповідні повідомлення про помилки.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		80

Після успішного додавання тварина коректно відображалась у профілі клієнта та була доступна для подальшого запису на прийом.

Таблиця 3.3

Опис тесту «Додавання нової тварини»

Кроки для перевірки	Заповнити форму додавання тварини та натиснути кнопку збереження
Вхідні дані	Коректно заповнені поля форми
Очікуваний результат	Тварину успішно додано до системи
Фактичний результат	Відповідає

Одним із ключових функціональних модулів платформи є запис на прийом. У таблиці 3.4 наведено опис тесту створення запису клієнтом із вибором лікаря, дати та типу послуги.

Таблиця 3.4

Опис тесту «Створення запису на прийом»

Кроки для перевірки	Обрати лікаря, дату, час прийому, тип послуги
Вхідні дані	Коректно заповнена форма запису
Очікуваний результат	Запис успішно створено та відображено у записах
Фактичний результат	Відповідає

Окрім клієнтського функціоналу, перевірено роботу лікарської частини системи. Таблиця 3.5 описує тест створення медичного запису лікарем після проведення огляду пацієнта. У межах тестування перевірялась можливість внесення діагнозу, призначеного лікування, результатів обстеження та додаткових рекомендацій. Система коректно зберігала введені дані та прив'язувала медичний запис до відповідної тварини і конкретного прийому. Також було перевірено, що створений запис стає доступним в історії хвороби пацієнта одразу після збереження, що забезпечує актуальність медичної інформації для всіх авторизованих користувачів.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		81

Таблиця 3.5

Опис тесту «Створення медичного запису лікарем»

Кроки для перевірки	Заповнити форму медичного огляду та зберегти запис
Вхідні дані	Коректні медичні дані
Очікуваний результат	Новий медичний запис збережено в картці пацієнта
Фактичний результат	Відповідає

Перевірено також адміністративні функції системи. Таблиця 3.6 описує тест редагування даних користувача адміністратором через панель управління.

Таблиця 3.6

Опис тесту «Редагування користувача адміністратором»

Кроки для перевірки	Відкрити профіль користувача та змінити його дані
Вхідні дані	Існуючий користувач
Очікуваний результат	Зміни успішно збережені
Фактичний результат	Відповідає

Для перевірки коректності управління записами протестовано сценарії їх видалення та перенесення. Таблиця 3.7 описує тест видалення існуючого запису на прийом із системи.

Таблиця 3.7

Опис тесту «Видалення запису на прийом»

Кроки для перевірки	Обрати запис та натиснути кнопку видалення
Вхідні дані	Існуючий запис
Очікуваний результат	Запис видалено із системи
Фактичний результат	Відповідає

Таблиця 3.8 описує суміжний сценарій – перенесення запису на інший час, що є поширеною операцією в реальній практиці клініки. Тестування

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		82

підтвердило, що система дозволяє змінювати дату та час прийому без втрати пов'язаних даних, зокрема інформації про пацієнта та лікаря. Перевірялась також коректність оновлення розкладу після перенесення: попередній часовий слот звільнявся та ставав доступним для нових записів, тоді як новий час відображався у календарі без конфліктів із вже існуючими прийомами.

Таблиця 3.8

Опис тесту «Перенесення запису на прийом»

Кроки для перевірки	Обрати запис та натиснути кнопку перенести, обрати новий час та дату
Вхідні дані	Існуючий запис
Очікуваний результат	Запис перенесено
Фактичний результат	Відповідає

Окремо протестовано функціонал онлайн-консультацій. Таблиця 3.9 описує тест приєднання авторизованого користувача до активного сеансу онлайн-консультації.

Таблиця 3.9

Опис тесту «Приєднання до онлайн-консультації»

Кроки для перевірки	Перейти до списку записів на прийом, обрати онлайн-консультацію та натиснути кнопку приєднання
Вхідні дані	Авторизований користувач, активний запис на онлайн-консультацію
Очікуваний результат	Відкривається сторінка онлайн-консультації та користувач успішно приєднується до сеансу
Фактичний результат	Відповідає

Завершальним тестом перевірено систему зворотного зв'язку. Таблиця 3.10 описує сценарій оцінювання прийому клієнтом після його завершення.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		83

Опис тесту «Оцінювання прийому»

Кроки для перевірки	Відкрити сповіщення про оцінювання лікаря, обрати оцінку, за потреби залишити коментар та натиснути кнопку підтвердження
Вхідні дані	Завершений прийом, авторизований користувач
Очікуваний результат	Оцінка та коментар успішно збережені в системі, користувач отримує повідомлення про успішне надсилання відгуку
Фактичний результат	Відповідає

За результатами проведеного тестування встановлено, що всі основні функціональні можливості вебплатформи працюють відповідно до поставлених вимог. Отримані фактичні результати збігаються з очікуваними, що підтверджує коректність реалізації механізмів авторизації, управління тваринами, записами на прийом та медичними картками. Проведене тестування підтвердило працездатність і готовність програмного продукту до подальшого використання.

Висновки до третього розділу

В даному розділі було описано структуру інтерфейсу вебплатформи для управління ветеринарною клінікою та розглянуто функціональні можливості, реалізовані для різних категорій користувачів системи. Детально представлено інтерфейси адміністратора, ветеринарного лікаря та клієнта, а також наведено приклади роботи основних модулів вебплатформи.

Система реалізована як веборієнтований клієнт-серверний застосунок, що відповідає сучасним вимогам до розробки інформаційних систем. Під час реалізації було враховано механізми авторизації та автентифікації користувачів, розмежування прав доступу відповідно до ролей, захист персональних даних та забезпечення цілісності інформації, що зберігається у базі даних PostgreSQL.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		84

Опис роботи з вебплатформою включає текстову та ілюстративну частини, які демонструють реалізований функціонал системи та особливості взаємодії користувачів з її компонентами. Розглянуто основні сценарії роботи адміністратора, лікаря та клієнта, що дозволяє оцінити повноту реалізованих можливостей програмного продукту.

У межах розділу також було проведено тестування основних функціональних модулів системи, зокрема авторизації користувачів, управління тваринами, записів на прийом, медичних карток, системи сповіщень, онлайн-консультацій та механізму оцінювання прийомів. Тестування проводилося в умовах, максимально наближених до реальної експлуатації системи. Отримані результати підтвердили коректність роботи розробленої вебплатформи та відповідність фактичних результатів очікуваням.

Таким чином, реалізована вебплатформа забезпечує автоматизацію основних процесів діяльності ветеринарної клініки, спрощує взаємодію між персоналом і клієнтами та може бути використана як основа для подальшого розвитку і розширення функціональних можливостей системи.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				85
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Метою роботи була розробка вебплатформи для управління ветеринарною клінікою, що автоматизує основні клінічні та адміністративні процеси закладу. Поставлену мету досягнуто в повному обсязі: створено повнофункціональну вебплатформу із трирівневою архітектурою «клієнт – сервер – база даних», яка охоплює весь операційний цикл ветеринарної клініки.

З технічного боку платформу реалізовано з використанням сучасного стеку технологій: клієнтську частину побудовано на основі Vue.js із застосуванням Vue Router для навігації та Pinia для управління станом застосунку, а для зберігання даних використано реляційну базу даних PostgreSQL, розгорнуту на хмарній платформі Neon. Розгортання клієнтської частини здійснено через Vercel, що забезпечує стабільну доступність платформи. Архітектурні рішення ґрунтуються на принципах чистої архітектури та патернах проектування, що забезпечує масштабованість системи та зручність її подальшого супроводу.

У межах розробки реалізовано повний комплекс функціональних модулів. Систему управління користувачами та персоналом побудовано з розмежуванням прав доступу для трьох ролей: адміністратора, ветеринарного лікаря та клієнта. Модуль запису на прийом забезпечує онлайн-запис, перенесення та скасування візитів із автоматичним надсиланням сповіщень усім учасникам. Медичні картки тварин дозволяють фіксувати історію хвороби, діагнози, призначення та результати аналізів. Модуль управління розкладом надає можливість переглядати й коригувати графік лікарів у режимі реального часу. Реалізовано також модуль онлайн-консультацій із збереженням їх результатів у картці пацієнта та блок загальної статистики, що автоматично формує показники роботи клініки для контролю адміністратором. Усі реалізовані модулі перевірено тестуванням в умовах, наближених до реальної експлуатації, та відповідають вимогам, сформульованим на етапі аналізу предметної області.

Практична цінність розробленої платформи полягає у можливості її безпосереднього впровадження у діяльність ветеринарних клінік малого та середнього масштабу. Використання хмарної інфраструктури усуває потребу у

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				86
Змн.	Арк.	№ докум.	Підпис	Дата		

власних серверних потужностях, що суттєво знижує вартість впровадження. Порівняно з дослідженими аналогами – Provet Cloud, ENOTE та Clinicia – платформа є україномовною, не потребує складного налаштування та може адаптуватися під специфіку конкретного закладу. Завдяки модульній архітектурі систему можна поетапно розширювати без необхідності переробки вже реалізованого функціоналу.

Перспективами подальшого вдосконалення є інтеграція з лабораторними інформаційними системами для автоматичного імпорту результатів аналізів, додавання модуля фінансового обліку з формуванням рахунків та звітності, розширення аналітичного блоку з побудовою прогнозних показників.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				87
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Provet Cloud. URL: <https://www.provet.cloud/> (дата звернення: 12.02.2026).
2. ENOTE. URL: <https://enote.vet/> (дата звернення: 12.02.2026).
3. Clinicia Veterinary Software. URL: <https://clinicia.com/> (дата звернення: 12.02.2026).
4. Vue.js Documentation. URL: <https://vuejs.org/> (дата звернення: 01.03.2026).
5. JavaScript Documentation. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення: 01.03.2026).
6. MDN Web Docs: JavaScript, HTML, CSS. URL: <https://developer.mozilla.org> (дата звернення: 01.03.2026).
7. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. New Jersey : Prentice Hall, 2017. 432 p.
8. Web.dev: Performance. URL: <https://web.dev/performance/> (дата звернення: 12.03.2026).
9. Vue.js Style Guide. URL: <https://vuejs.org/style-guide> (дата звернення: 25.03.2026).
10. Google Developers. Web Fundamentals. URL: <https://developers.google.com/web> (дата звернення: 25.03.2026).
11. React Documentation. URL: <https://react.dev/> (дата звернення: 25.03.2026).
12. Flanagan D. JavaScript: The Definitive Guide. URL: <https://www.oreilly.com/library/view/javascript-the-definitive/9781491952016/> (дата звернення: 02.04.2026).
13. Web.dev. URL: <https://web.dev/> (дата звернення: 02.04.2026).
14. Vue Router Documentation. URL: <https://router.vuejs.org/> (дата звернення: 25.04.2026).

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				
Змн.	Арк.	№ докум.	Підпис	Дата		88

15. Pinia Documentation. URL: <https://pinia.vuejs.org/> (дата звернення: 12.04.2026).
16. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/> (дата звернення: 25.04.2026).
17. Axios Documentation. URL: <https://axios-http.com/> (дата звернення: 04.05.2026).
18. Neon Documentation. URL: <https://neon.tech/docs> (дата звернення: 12.05.2026).
19. Vercel Documentation. URL: <https://vercel.com/docs> (дата звернення: 12.05.2026).
20. Passport.js Documentation. URL: <https://www.passportjs.org/docs/> (дата звернення: 25.03.2026).
21. PostgreSQL Tutorial. URL: <https://www.postgresqltutorial.com/> (дата звернення: 25.04.2026).
22. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. New York : McGraw-Hill Education, 2019. 880 p.
23. W3C Web Standards. URL: <https://www.w3.org/standards/> (дата звернення: 18.05.2026).
24. UML Resource Page. URL: <https://www.uml.org/> (дата звернення: 18.05.2026).
25. GitLab Documentation. URL: <https://docs.gitlab.com/> (дата звернення: 12.03.2026).
26. Node-Cron Documentation. URL: <https://nodecron.com/> (дата звернення: 20.05.2026).
27. Google OAuth 2.0 Documentation. URL: <https://developers.google.com/identity/protocols/oauth2> (дата звернення: 12.03.2026).
28. GitLab CI/CD Documentation. URL: <https://docs.gitlab.com/ee/ci/> (дата звернення: 20.05.2026).

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Нерода С.І				89
Змн.	Арк.	№ докум.	Підпис	Дата		

29. MDN Web Docs: HTTP Overview. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP> (дата звернення: 02.06.2026).

30. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston : Addison-Wesley, 1994. 416 p.

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				90
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				91
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаток А

Лістинг А.1 – Реалізація авторизації користувача

```
router.post('/login', async (req, res) => {  
  const { username, password } = req.body;  
  
  const { rows } = await pool.query(  
    'SELECT * FROM users WHERE username = $1 AND enabled = true',  
    [username]  
  );  
  
  const user = rows[0];  
  if (!user) {  
    return res.status(401).json({ message: 'Невірний логін або пароль' });  
  }  
  
  const isMatch = await bcrypt.compare(password, user.password);  
  if (!isMatch) {  
    return res.status(401).json({ message: 'Невірний логін або пароль' });  
  }  
  
  const token = jwt.sign(  
    {  
      id: user.id,  
      username: user.username,  
      role: user.role,  
      fullName: user.full_name  
    },  
    process.env.JWT_SECRET,  
    { expiresIn: process.env.JWT_EXPIRES_IN }  
  );  
}
```

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				92
Змн.	Арк.	№ докум.	Підпис	Дата		

```
);
```

```
res.json({  
  token,  
  userId: user.id,  
  username: user.username,  
  fullName: user.full_name,  
  role: user.role,  
  mustChangePassword: user.must_change_password  
});  
});
```

Лістинг А.2 – Реалізація модуля правління користувачами

backend/routes/users.js

```
router.post('/', requireRole('ADMIN'), async (req, res) => {  
  const { username, password, fullName, email, phone, role, services } =  
req.body;  
  
  const hashed = await bcrypt.hash(password, 10);  
  
  const { rows } = await pool.query(  
    `INSERT INTO users  
      (username, password, full_name, email, phone, role, enabled,  
must_change_password)  
      VALUES ($1, $2, $3, $4, $5, $6, true, true)  
      RETURNING id, username, full_name, email, phone, role, enabled`,  
    [username, hashed, fullName, email || null, phone || null, role]  
  );  
  
  const createdUser = rows[0];
```

		Вознюк В.Ю			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Герода С.І				93
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if (role === 'DOCTOR' && Array.isArray(services)) {
  for (const service of services) {
    await pool.query(
      `INSERT INTO doctor_services (doctor_id, service_type)
      VALUES ($1, $2)
      ON CONFLICT (doctor_id, service_type) DO NOTHING`,
      [createdUser.id, service]
    );
  }
}

res.status(201).json(createdUser);
});

```

Лістинг А.3 – Реалізація додавання тварини

backend/routes/pets.js

```

router.post('/', async (req, res) => {
  const {
    ownerId,
    name,
    species,
    breed,
    birthDate,
    weight,
    chipNumber,
    allergies,
    chronicConditions
  } = req.body;

```

		Вознюк В.Ю			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				94
Змн.	Арк.	№ докум.	Підпис	Дата		

```

const { rows } = await pool.query(
  `INSERT INTO pets
    (owner_id, name, species, breed, birth_date, weight,
    chip_number, allergies, chronic_conditions)
  VALUES ($1,$2,$3,$4,$5,$6,$7,$8,$9)
  RETURNING *`,
  [
    ownerId,
    name,
    species,
    breed || null,
    birthDate || null,
    weight || null,
    chipNumber || null,
    allergies || null,
    chronicConditions || null
  ]
);

res.status(201).json(rows[0]);
});

```

ЛІСТИНГ А.4 – Реалізація створення запису на прийом

backend/routes/appointments.js

```

router.post('/', async (req, res) => {
  const {
    petId,
    doctorId,
    appointmentDateTime,
    serviceType,

```

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				95
Змн.	Арк.	№ докум.	Підпис	Дата		

```

reason,
isOnline
} = req.body;

```

```

const { rows } = await pool.query(
  `INSERT INTO appointments
  (pet_id, doctor_id, appointment_date_time, status,
  service_type, reason, is_online)
  VALUES ($1, $2, $3, 'SCHEDULED', $4, $5, $6)
  RETURNING *`,
  [
    petId,
    doctorId,
    appointmentDateTime,
    serviceType || null,
    reason || null,
    Boolean(isOnline)
  ]
);

```

```

let appointment = rows[0];

```

```

if (appointment.is_online) {
  const meetingLink = `https://meet.jit.si/vetclinic-appointment-
  ${appointment.id}`;

```

```

const updated = await pool.query(
  `UPDATE appointments
  SET meeting_link = $1

```

		Вознюк В.Ю			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Герода С.І				96
Змн.	Арк.	№ докум.	Підпис	Дата		

```

WHERE id = $2
RETURNING *,
[meetingLink, appointment.id]
);

appointment = updated.rows[0];
}

res.status(201).json(appointment);
});

```

Лістинг А.5 – Реалізація створення медичного запису

backend/routes/medicalRecords.js

```

router.post('/', async (req, res) => {
  const {
    petId,
    doctorId,
    appointmentId,
    recordDate,
    diagnosis,
    treatmentPlan,
    prescriptions,
    vaccinations,
    labResults,
    notes,
    weight
  } = req.body;

```

```

const { rows } = await pool.query(
  `INSERT INTO medical_records

```

		Вознюк В.Ю			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Нерода С.І				97
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    (pet_id, doctor_id, appointment_id, record_date,
    diagnosis, treatment_plan, prescriptions,
    vaccinations, lab_results, notes, weight)
VALUES ($1,$2,$3,$4,$5,$6,$7,$8,$9,$10,$11)
RETURNING *`,
[
    petId,
    doctorId || null,
    appointmentId || null,
    recordDate || new Date(),
    diagnosis || null,
    treatmentPlan || null,
    prescriptions || null,
    vaccinations || null,
    labResults || null,
    notes || null,
    weight || null
]
);

```

```

    res.status(201).json(rows[0]);
});

```

ЛІСТИНГ А.6 – Реалізація графіка лікаря

backend/routes/shedules.js

```

router.post('/', requireRole('ADMIN'), async (req, res) => {
    const {
        doctorId,
        dayOfWeek,
        startTime,

```

		Вознюк В.Ю			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				98
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    endTime,
    isActive
  } = req.body;

  const { rows } = await pool.query(
    `INSERT INTO doctor_schedules
      (doctor_id, day_of_week, start_time, end_time, is_active)
      VALUES ($1, $2, $3, $4, $5)
      RETURNING *`,
    [
      doctorId,
      dayOfWeek,
      startTime,
      endTime,
      isActive ?? true
    ]
  );

  res.status(201).json(rows[0]);
});

```

ЛІСТИНГ А.7 – Реалізація сповіщень

backend/routes/notifications.js

```

router.get('/', async (req, res) => {
  const { rows } = await pool.query(
    `SELECT *
      FROM notifications
      WHERE user_id = $1
      ORDER BY created_at DESC
      LIMIT 50`,

```

		Вознюк В.Ю			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				99
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    [req.user.id]
  );

  res.json(rows);
});

router.patch('/:id/read', async (req, res) => {
  await pool.query(
    `UPDATE notifications
     SET is_read = true
     WHERE id = $1 AND user_id = $2`,
    [req.params.id, req.user.id]
  );

  res.json({ success: true });
});

```

Лістинг А.8 – Оцінка прийому

backend/routes/rating.js

```

router.post('/', async (req, res) => {
  const { appointmentId, rating, comment } = req.body;

  if (!appointmentId || !rating) {
    return res.status(400).json({
      message: 'Прийом та оцінка обов'язкові'
    });
  }

  const { rows } = await pool.query(
    `INSERT INTO appointment_ratings

```

		Вознюк В.Ю			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				100
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        (appointment_id, user_id, rating, comment)
VALUES ($1, $2, $3, $4)
RETURNING *,
[
    appointmentId,
    req.user.id,
    rating,
    comment || null
]
);

res.status(201).json(rows[0]);
});

```

		Вознюк В.Ю			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Нерода С.І				101
Змн.	Арк.	№ докум.	Підпис	Дата		