

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

Кваліфікаційна робота бакалавра
з теми: **«Розробка інформаційної системи у вигляді web-ресурсу
та мобільного додатку для надання сервісів догляду за
домашніми тваринами»**

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

Мрачковський Олександр Едуардович

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

Керівник: Федорчук Володимир Анатолійович,

доктор технічних наук, професор

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

Рецензент: Кух Аркадій Миколайович,

професор кафедри фізики, доктор

педагогічних наук, професор

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

м. Кам'янець-Подільський - 2025 р.

АНОТАЦІЯ

Кваліфікаційна робота на тему «Розробка інформаційної системи у вигляді web-ресурсу та мобільного додатку для надання сервісів догляду за домашніми тваринами».

У роботі проаналізовано сучасні цифрові сервіси у сфері ветеринарії, виявлено недоліки наявних рішень в Україні та обґрунтовано потребу в створенні локалізованої цифрової платформи. Розроблено веб-інтерфейс на базі Angular та мобільний додаток із використанням Capacitor, реалізовано бекенд за допомогою Express.js, зберігання даних - у MongoDB. Система містить модулі обліку тварин, медичної історії, клінік, лікарів, алергій та забезпечує гнучке адміністрування. Створено APK-файл для мобільного пристрою.

Отримані результати можуть бути використані ветеринарними клініками, для автоматизації процесів обліку, покращення взаємодії з клієнтами та підвищення якості послуг. Система є придатною для подальшого вдосконалення та впровадження в реальну практику.

Ключові слова: інформаційна система, ветеринарія, мобільний додаток, веб-ресурс, домашні тварини, MongoDB, Angular, waw Framework.

ЗМІСТ

АНОТАЦІЯ.....	2
ВСТУП.....	4
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ... 6	6
1.1 Актуальність теми та огляд існуючих рішень.....	6
1.2 Аналіз потреб користувачів та постановка задачі.....	8
1.3. Технології та засоби реалізації інформаційних систем для ветеринарії	10
Висновки до розділу 1.....	12
РОЗДІЛ 2 ВИБІР ТЕХНОЛОГІЙ ТА ОБҐРУНТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ.....	14
2.1 Аналіз вибору технологій для розробки інформаційної системи.....	14
2.1.1 Фронтенд-технології.....	14
2.1.2 Бекенд-технології.....	15
2.1.3 Системи управління базами даних (СУБД).....	16
2.2 Вибір технологічного стеку для розробки фронтенд- та бекенд-частин	17
2.2.1 Огляд фреймворку waw Framework.....	17
2.2.2 Фронтенд на базі waw Framework.....	17
2.2.3 Бекенд, створений із використанням waw Framework.....	18
2.2.4 Кастомізація та UI-елементи.....	18
2.2.5 Переваги вибраного стеку технологій.....	19
Висновок до розділу 2.....	19
РОЗДІЛ 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	20
3.1. Побудова плану інформаційної системи.....	20
3.2. Створення модулів.....	22
3.3. Формування зв'язків між модулями.....	27
3.4. Розробка користувацьких сторінок.....	29
3.5. Налаштування базового дизайну сторінок.....	36
3.6. Створення АРК для мобільного телефону.....	38
Висновки до розділу 3.....	39
ВИСНОВКИ.....	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	43

ВСТУП

У сучасному суспільстві спостерігається зростання популярності домашніх тварин, що супроводжується розширенням спектру послуг з догляду за ними. Власники тварин дедалі частіше звертаються до ветеринарних клінік, грумінг-салонів, сервісів з вигулу, хостелів та інших спеціалізованих послуг. Разом із цим зростає потреба у цифрових інструментах, які дозволяють зручно та ефективно управляти інформацією про тварин, планувати прийоми до спеціалістів, зберігати історію здоров'я вихованців і взаємодіяти з різними сервісами.

Традиційні способи ведення обліку — паперові носії, нотатки в месенджерах або хаотичні файли є застарілими, незручними і не забезпечують належного рівня організації даних. У багатьох випадках це призводить до дублювання інформації, втрати важливих даних або труднощів при плануванні процедур чи взаємодії з ветеринарами. Відсутність централізованої цифрової системи є суттєвим бар'єром як для власників тварин, так і для фахівців.

У зв'язку з цим розробка інформаційної системи у вигляді web-ресурсу та мобільного додатку для надання сервісів догляду за домашніми тваринами є актуальною задачею, що відповідає викликам часу. Така система покликана забезпечити автоматизацію ключових процесів, підвищити якість сервісу, покращити комунікацію між користувачами й організаціями та оптимізувати управління інформацією про тварин.

Об'єктом дослідження є процеси автоматизації сервісів догляду за домашніми тваринами на основі інформаційних технологій.

Предметом дослідження є архітектура, функціональність та реалізація інформаційної системи у вигляді web-ресурсу та мобільного додатку, яка забезпечує облік домашніх тварин, історію їхнього здоров'я та взаємодію користувачів із сервісами.

Метою роботи є створення прототипу інформаційної системи, що складається з веб-додатку та мобільного клієнта для надання сервісів догляду за домашніми тваринами.

Для досягнення поставленої мети необхідно виконати такі **завдання**:

- Проаналізувати сучасні рішення в галузі цифрових сервісів для домашніх тварин.
- Розробити архітектуру інформаційної системи.
- Реалізувати веб-інтерфейс на основі Angular для керування профілями тварин та історією їхнього здоров'я.
- Створити мобільний додаток для зручного доступу до функціоналу системи з мобільних пристроїв.
- Забезпечити збереження та обробку даних через базу даних MongoDB.
- Провести тестування базового прототипу системи та оцінити її практичну придатність.

У процесі дослідження було використано **методи** аналізу предметної області, моделювання інформаційних систем, проектування клієнт-серверної архітектури, а також практичної реалізації програмних компонентів з використанням фреймворку Angular, бази даних MongoDB та мобільних технологій. Застосовувалися також методи тестування функціональності та взаємодії між модулями системи.

Результати роботи мають практичне значення для власників домашніх тварин, ветеринарних клінік, грумерів і сервісів з вигулу. Розроблений прототип інформаційної системи дозволяє централізовано управляти інформацією про тварин, вести електронну історію хвороб, записуватись до ветеринарів, знаходити клініки та отримувати аналітику. В подальшому система може бути розширена та впроваджена у роботу відповідних підприємств, що дозволить підвищити ефективність обслуговування клієнтів.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Актуальність теми та огляд існуючих рішень

Інформаційні системи у галузі ветеринарної медицини останні роки поступово почали набувати значення в умовах цифрової трансформації медичних послуг. У країнах з розвиненими системами охорони здоров'я тварин впровадження інформаційних технологій допомогло суттєво підвищити якість обслуговування, підвищити рівень комунікації між клініками та власниками тварин та внутрішні процеси в медичних закладах. Одночасно в Україні ця галузь все ще знаходиться на ранній стадії розвитку, що створює передумови для дослідження та впровадження нових рішень у цій галузі.

На міжнародному рівні існує низка систем, орієнтованих на забезпечення взаємодії між власниками тварин і ветеринарними установами. Наприклад, системи PetDesk, Pawprint та VitusVet пропонують мобільні додатки для зберігання медичних даних тварин, планування візитів до клініки, отримання нагадувань про щеплення, доступ до електронної історії хвороби тощо. Такі сервіси дозволяють автоматизувати низку рутинних задач, зменшити ризик втрати медичних даних і покращити досвід користувачів.

Однак більшість зазначених сервісів функціонує на території США або країн Західної Європи. Вони здебільшого не підтримують українську мову, прив'язані до локальних клінік і вимагають підтвердження акаунтів через специфічні форми авторизації. Це суттєво обмежує їх застосування в Україні, де переважають інші технічні та нормативні умови. Додатково, відсутність єдиних стандартів ведення електронної медичної документації тварин ускладнює можливість перенесення закордонного досвіду без істотних адаптацій.

Що до українського ринку, то навіть поверхневий аналіз свідчить про майже повну відсутність сучасних цифрових послуг в області ветеринарії. Приналежна більшість клінік обмежується наявністю сторінки у соціальних мережах або мінімального веб-сайту з контактною інформацією і переліченням послуг. Функціонал для онлайн-запису чи огляду історії лікування тварин в більшості випадків відсутній. Усі медичні дані або залишаються у паперовому вигляді в самій клініці, або передаються власнику тварини у вигляді довідок, фотографій результатів аналізів чи повідомлень у месенджерах. Відповідно це створює додаткові ризики втрату цінної інформації, підвищує рівень бюрократії процесу моніторингу здоров'я тварини і видаляє можливість ефективної взаємодії між різними клініками в разі необхідності повторного звернення.

Ситуація в Україні залишається досить складною. Більшість ветеринарних клінік, особливо в невеликих містах, або взагалі не мають веб-представництв, або використовують стандартні сайти на основі шаблонів, де можна знайти лише базову інформацію: адресу, контактний номер та список послуг. Випадки, коли сайт дозволяє онлайн-запис чи містить інтерактивну карту, - поодинокі. Власники домашніх тварин часто змушені зберігати дані про лікування своїх улюбленців у вигляді паперових довідок, повідомлень у месенджерах або взагалі покладатися на пам'ять.

Для порівняння, на заході вже давно функціонують системи, що покращують взаємодію між клініками та клієнтами. Наприклад, PetDesk дозволяє власникам тварин керувати медичними даними, отримувати нагадування про щеплення, а також комунікувати з лікарем через мобільний додаток. Інша система, Pawprint, спеціалізується на централізованому збереженні медичних записів та обліку кількох тварин у рамках одного профілю. Водночас, більшість таких платформ орієнтовані на ринок США, що ускладнює використання в Україні: мова інтерфейсу, складність реєстрації без прив'язки до конкретної клініки, а також непристосованість до локального законодавства.

У підсумку, ринок України залишається відкритим для впровадження подібних систем. Створення адаптованої платформи може заповнити наявну прогалину в інфраструктурі цифрових ветеринарних сервісів. Враховуючи зростаючу популярність домашніх тварин і підвищену увагу до здоров'я улюбленців, запровадження доступної й інтуїтивно зрозумілої інформаційної системи виглядає доцільним та перспективним кроком. Таким чином, розробка веб-застосунку, що забезпечить зручний облік тварин, доступ до медичних даних, функції запису на прийом і взаємодію з клініками, може стати вагомим внеском у цифровізацію ветеринарної медицини України.

1.2 Аналіз потреб користувачів та постановка задачі

Інформаційні системи для ветеринарних клінік та власників домашніх тварин набувають все більшого поширення у світі, особливо у розвинених країнах. Ці системи забезпечують зручний облік медичних даних тварин, планування візитів, автоматичні нагадування про процедури та вакцини, а також спрощують комунікацію між клініками і власниками. Розглянемо найбільш відомі і широко використовувані приклади таких систем.

Одним із лідерів є система PetDesk (США), яка поєднує функції мобільного додатку та веб-платформи. Вона дозволяє власникам тварин зберігати історію хвороб, планувати прийоми у ветеринара, отримувати сповіщення про майбутні вакцинації та ліки, а також безпосередньо комунікувати з клінікою. PetDesk інтегрується з інформаційними системами клінік, що дозволяє лікарям ефективно керувати пацієнтами. Популярність цієї системи пов'язана з високою зручністю використання, широким функціоналом та надійністю.

Інша популярна платформа Pawprint спеціалізується на збереженні повної медичної документації тварин у хмарі. Власник може додавати документи, фотографії аналізів, а також відстежувати стан здоров'я своїх

улюбленців. Система підтримує кілька профілів для різних тварин, що особливо зручно для сімей з великою кількістю домашніх улюбленців. Також Pawprint пропонує нагадування і можливість замовлення ліків онлайн.

Варто згадати і VitusVet — систему, яка окрім обліку даних і нагадувань, надає функції телемедицини — відеоконсультації з ветеринаром. Такий підхід особливо актуальний у сучасних умовах, коли віддалений доступ до фахівця стає важливим.

Ще одним цікавим рішенням є Vetstoria — платформа для онлайн-бронювання візитів у клініки, що дозволяє автоматизувати процес запису та розподілу часу лікарів. Цей сервіс також включає аналітичні інструменти для клінік, що допомагають оптимізувати графіки роботи.

Попри широкі можливості цих систем, їх використання в Україні стикається з низкою викликів. По-перше, більшість із них орієнтовані на англomовний ринок, що створює мовний бар'єр для українських користувачів. По-друге, специфіка локального законодавства, а також стандарти ведення медичної документації відрізняються від тих, що застосовуються у США чи Європі. Це ускладнює інтеграцію зарубіжних рішень без значних доопрацювань і адаптації.

Крім того, відсутність локалізованих ветеринарних платформ в Україні свідчить про невирішену проблему цифрової трансформації у цій сфері. Станом на сьогодні, більшість українських ветеринарних клінік використовують застарілі або примітивні засоби для ведення обліку та комунікації, що значно знижує якість сервісу та рівень задоволеності клієнтів.

Варто зазначити, що на пострадянському просторі є кілька спроб створення подібних платформ, але вони не набули широкого визнання через обмежений функціонал, відсутність мобільних додатків або слабку підтримку користувачів.

Отже, аналіз існуючих аналогів у світі свідчить про перспективність розробки адаптованої інформаційної системи для ветеринарних клінік і

власників тварин в Україні. Навіть невеликий локальний проєкт чи макет може стати корисним інструментом для демонстрації можливостей цифровізації у цій сфері та слугувати основою для подальшого розвитку. Така система, створена з урахуванням специфіки ринку і локальних потреб, може допомогти покращити якість обслуговування, оптимізувати облік та спростити комунікацію між власниками тварин і ветеринарними закладами на прикладі окремого регіону чи клініки.

1.3. Технології та засоби реалізації інформаційних систем для ветеринарії

Розробка інформаційних систем для ветеринарії є складним та багатогранним процесом, що потребує застосування різноманітних технологічних підходів і інструментів. Основна мета таких систем полягає у забезпеченні зручного, надійного та ефективного обліку медичних даних тварин, організації комунікації між власниками та ветеринарними клініками, а також оптимізації внутрішніх процесів у ветеринарних установах.

В сучасних умовах цифрової трансформації особливу роль відіграють веб-технології, які дозволяють створювати інтерактивні та багатофункціональні веб-застосунки. Зокрема, популярними фреймворками для розробки фронтенду є Angular, React та Vue.js. Вони дають можливість реалізувати динамічні інтерфейси користувача, які швидко реагують на дії та запити користувачів. Angular, наприклад, часто використовується для побудови складних SPA (Single Page Application), що зручні для роботи в браузері без необхідності постійного перезавантаження сторінок.

Для серверної частини все більше застосовується Node.js — платформа, яка дозволяє запускати JavaScript на сервері. Разом з популярним фреймворком Express.js вона забезпечує гнучкість і продуктивність у обробці запитів від клієнтів. Такий підхід дозволяє створити швидкий та ефективний

бекенд, який обробляє інформацію, взаємодіє з базою даних і керує бізнес-логікою системи.

Що стосується баз даних, для зберігання інформації про тварин, їх історію хвороб, записи на прийом та інші важливі дані зазвичай використовують реляційні (SQL) або документоорієнтовані (NoSQL) бази даних. Реляційні бази, як-от MySQL або PostgreSQL, мають чітку структуру таблиць і відносин між ними, що добре підходить для систем із суворою схемою даних. Водночас документоорієнтовані бази, такі як MongoDB, дозволяють зберігати дані у вигляді гнучких документів JSON-подібної структури, що зручно при роботі з різномірною або часто змінною інформацією. Вибір конкретної бази даних залежить від поставлених завдань, особливостей даних та вимог до продуктивності.

Окрім базових технологій, важливим аспектом є забезпечення безпеки інформації. Медичні дані тварин, як і будь-які персональні дані, потребують надійного захисту. Тому при розробці системи застосовують стандарти шифрування, наприклад, протокол HTTPS для захищеного обміну даними між клієнтом і сервером. Також використовується аутентифікація та авторизація користувачів, часто на основі технології JWT (JSON Web Tokens), що дозволяє ідентифікувати користувачів та контролювати доступ до різних функцій системи.

Враховуючи сучасні тенденції, все більшої популярності набувають мобільні додатки, які дозволяють власникам тварин отримувати доступ до медичної інформації у будь-який час і в будь-якому місці. Вони можуть надсилати нагадування про прийоми до ветеринара, вакцинування чи прийом ліків, а також забезпечують просту комунікацію з клінікою. Для розробки мобільних додатків часто використовують такі інструменти, як Flutter, React Native або NativeScript, які дозволяють створювати кросплатформенні додатки з єдиним кодовим базисом.

Ще одним цікавим напрямом є використання хмарних сервісів, зокрема платформи Firebase від Google. Вона надає розробникам широкий спектр

інструментів: від зберігання даних і аутентифікації користувачів до аналітики і відправки push-повідомлень. Використання таких сервісів особливо вигідне для невеликих команд розробників, оскільки значно спрощує налаштування інфраструктури і дозволяє зосередитися на функціональності системи.

Підсумовуючи, можна сказати, що сучасні технології надають широкі можливості для створення ефективних інформаційних систем у ветеринарній сфері. Вибір правильного інструментарію і архітектури є ключовим для досягнення високої якості, безпеки та зручності користування. Використання сучасних веб-технологій, мобільних платформ та хмарних сервісів сприяє покращенню цифрової інфраструктури ветеринарних послуг і робить їх більш доступними для користувачів.

Висновки до розділу 1

Проведений огляд та аналіз стану розвитку інформаційних систем у ветеринарній медицині підтвердили, що на світовому рівні існує низка ефективних рішень, які суттєво полегшують комунікацію між власниками тварин і медичними установами, забезпечують збереження та швидкий доступ до медичних даних, а також допомагають у плануванні лікування та профілактичних заходів. Однак більшість цих систем орієнтовані на ринки країн із розвиненою інфраструктурою та мають специфічні особливості, що ускладнює їх безпосереднє впровадження в Україні без суттєвих адаптацій.

В Україні ситуація з цифровими ветеринарними сервісами є далеко не оптимальною: відсутність єдиних стандартів ведення електронної медичної документації, недостатній рівень цифрової грамотності окремих клінік та обмежений функціонал наявних веб-ресурсів створюють значні бар'єри для розвитку цього напрямку. Це відкриває перспективи для створення локальних систем, які будуть адаптовані до особливостей українського ринку, законодавства та потреб користувачів.

Окрім цього, аналіз сучасних технологічних засобів показує, що існує широкий спектр інструментів для розробки інформаційних систем із різними рівнями складності та функціоналу. Від використання простих веб-технологій для створення сайтів із базовою інформацією до впровадження мобільних додатків та систем управління базами даних — сучасний технічний арсенал дозволяє реалізувати навіть складні проєкти в умовах обмежених ресурсів.

Таким чином, загальний висновок полягає у тому, що цифровізація ветеринарної сфери в Україні має великий потенціал і є актуальним напрямом для дослідження та розробки. Розробка локальних інформаційних систем, адаптованих до реальних потреб користувачів і особливостей галузі, здатна підвищити якість обслуговування, сприяти більш ефективній роботі клінік і забезпечити кращу комунікацію між усіма учасниками процесу.

РОЗДІЛ 2 ВИБІР ТЕХНОЛОГІЙ ТА ОБҐРУНТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ

2.1 Аналіз вибору технологій для розробки інформаційної системи

В сучасних інформаційних системах, зокрема у сфері ветеринарії та догляду за домашніми тваринами, особливо важливо вибрати правильний технологічний стек, який забезпечить надійність, масштабованість, гнучкість і зручність у подальшій підтримці системи. У цьому підрозділі розглянемо основні напрямки технологій, які використовуються у розробці таких систем: фронтенд, бекенд, бази даних, а також хмарні сервіси та API.

2.1.1 Фронтенд-технології

Фронтенд — це частина системи, з якою взаємодіють кінцеві користувачі. Він відповідає за інтерфейс, візуалізацію даних і взаємодію з користувачем.

Angular — це повноцінний фреймворк, який включає в себе не лише бібліотеки для побудови UI, а й комплексні рішення для роутингу, роботи з формами, валідації, управління станом додатка. Він базується на TypeScript, що підвищує типобезпечність і полегшує масштабування. Angular підходить для великих корпоративних проєктів, де потрібна строгість архітектури. Однак цей фреймворк може бути більш складним для новачків через багат шаровість і значний обсяг концептів, які потрібно освоїти.

React — це бібліотека, яка концентрується на побудові UI-компонентів. Для повноцінної розробки додатка потрібні додаткові бібліотеки (React Router, Redux чи Context API, Axios тощо). React є дуже популярним завдяки простоті базового API, гнучкості та швидкості роботи. Проте, архітектура проєктів на React менш стандартизована, що може створювати плутанину у великих командах без чітких конвенцій.

Vue.js — поєднує простоту освоєння з потужністю фреймворку. Vue дозволяє швидко створювати адаптивні інтерфейси, має гарну документацію та підтримку TypeScript. Через поміркований розмір і легкість освоєння він ідеально підходить для середніх проєктів і швидких прототипів.

Приклади використання:

Система PetDesk, популярна в США, використовує React для фронтенду, що забезпечує їм гнучкість у створенні веб- та мобільних додатків, а також швидкі оновлення UI.

Європейські ветеринарні клініки застосовують Angular для внутрішніх CRM-систем, оскільки цей фреймворк сприяє підтримці складних бізнес-логік та масштабування.

2.1.2 Бекенд-технології

Бекенд відповідає за логіку додатку, обробку запитів, роботу з базою даних та інтеграцію із зовнішніми сервісами.

Node.js — це середовище виконання JavaScript на сервері. Однією з основних переваг є те, що фронтенд і бекенд можуть використовувати одну мову, що спрощує розробку та підтримку. Node.js підходить для створення масштабованих RESTful API, має велику екосистему модулів (npm), а також підтримує асинхронні операції, що дозволяє ефективно працювати з великим потоком запитів.

Django — фреймворк для мови Python, що надає готову адміністративну панель, ORM, систему автентифікації, ідеально підходить для швидкої розробки повноцінних веб-додатків. Django славиться своєю безпекою та стабільністю, має багату екосистему.

Spring Boot — потужний Java-фреймворк, орієнтований на корпоративні рішення. Забезпечує гнучкість, надійність, та широкий спектр інструментів для створення масштабованих веб-сервісів з високими вимогами до безпеки і транзакційності.

Приклади:

Проекти, такі як Rawprint, використовують Node.js через його швидкість і можливість масштабування, а також інтеграцію з різними API, необхідними для збору та обробки медичних даних.

Багато державних систем і медичних платформ у Європі розробляються на Django завдяки швидкості реалізації та надійності.

2.1.3 Системи управління базами даних (СУБД)

Вибір бази даних має критичне значення для зберігання, пошуку та аналізу інформації про тварин, їх історію хвороб, записи прийомів тощо.

MongoDB — документоорієнтована NoSQL база, що зберігає дані у форматі JSON-подібних документів. Це дає гнучкість у зберіганні структурованих і напівструктурованих даних. MongoDB підходить для швидкого прототипування, підтримує масштабування і розподілене зберігання даних.

MySQL — традиційна реляційна база даних, яка добре підходить для систем з чіткою структурою даних і потребою у складних запитах. Має розвинений механізм транзакцій і забезпечує цілісність даних.

PostgreSQL — ще одна популярна реляційна СУБД з розширеними можливостями, такими як підтримка геопросторових даних, складних типів і повнотекстового пошуку. Постгрес використовується для аналітики та систем, де важлива надійність і розширена функціональність.

Приклади:

Багато медичних інформаційних систем застосовують MongoDB для зберігання різноманітних медичних даних, зокрема у стартапах і малих проєктах, де швидкість розробки і гнучкість важливі.

Великі клінічні рішення, що підтримують складні бізнес-процеси, часто використовують PostgreSQL або MySQL для гарантії надійності та точності даних.

2.2 Вибір технологічного стеку для розробки фронтенд- та бекенд-частин

2.2.1 Огляд фреймворку waw Framework

Для розробки фронтенд-частини інформаційної системи було обрано waw Framework, розроблений компанією WebArtWork. Цей фреймворк базується на платформі Node.js і позиціонується як мінімалістичне, надзвичайно легке ядро, яке містить лише базову логіку запуску проєкту. Вся функціональність додається у вигляді окремих модулів, що дає змогу адаптувати систему під будь-які потреби без зайвого програмного навантаження.

Ключові переваги waw Framework:

Мінімалістичне ядро: містить лише базову логіку ініціалізації, що забезпечує високу швидкодію та мінімальні затрати ресурсів.

Модульна архітектура: додає функціонал за допомогою підключення необхідних модулів, що дає максимальну гнучкість.

Підтримка сучасних фронтенд-технологій: сумісний з Angular, React, Vue та іншими.

Повноцінна підтримка бекенду: окремий тип проєкту для серверної частини з власним набором модулів і налаштувань.

Можливість розширення: через розробку власних модулів легко інтегрувати нові технології.

2.2.2 Фронтенд на базі waw Framework

У цьому проєкті фронтенд створений за допомогою waw Framework із застосуванням Angular [6]. Такий вибір обумовлений широкими можливостями Angular для побудови складних, інтерактивних інтерфейсів та підтримкою модульної архітектури, що відповідає концепції waw.

Для стилізації використовується Sass - препроцесор, який розширює CSS новими можливостями:

- Змінні для керування кольорами, відступами та шрифтами.
- Вкладені селектори для покращеної організації коду.
- Міксіни, які дозволяють уникати дублювання стилів.
- Функції і оператори для створення динамічних стилів.

Це сприяє більш структурованому та підтримуваному CSS-коду, що особливо актуально для масштабних проєктів із численними UI-елементами.

2.2.3 Бекенд, створений із використанням waw Framework

Важливо відмітити, що waw Framework також має спеціальний тип проєкту, призначений для розробки серверної частини. Це дозволяє розробникам працювати у єдиному середовищі та уніфікувати стек технологій.

За замовчуванням бекенд використовує перевірені технології:

Express.js — популярний веб-фреймворк для Node.js, що забезпечує швидке і гнучке створення API, маршрутизацію HTTP-запитів та обробку middleware.

Socket.io — бібліотека для реалізації комунікації в режимі реального часу між сервером і клієнтом, що дозволяє організувати, наприклад, чат, живі оновлення або інтерактивні ігри.

MongoDB — документна база даних NoSQL, яка зберігає інформацію у вигляді гнучких JSON-подібних документів, легко масштабується і підходить для зберігання як структурованих, так і напівструктурованих даних.

2.2.4 Кастомізація та UI-елементи

Однією із сильних сторін waw Framework є можливість створення кастомних UI-компонентів. Це дозволяє розробляти унікальні, інтуїтивно зрозумілі елементи інтерфейсу, які повністю відповідають функціональним і естетичним вимогам проєкту.

Використання власних компонентів підвищує привабливість і зручність користування системою, що є особливо важливим для забезпечення позитивного користувацького досвіду.

2.2.5 Переваги вибраного стеку технологій

Використання waw Framework у комбінації з Angular, Sass, Express.js, Socket.io та MongoDB забезпечує:

Уніфікацію розробки: однаковий підхід і технології для фронтенду та бекенду.

Модульність: можливість включати лише необхідні компоненти, що оптимізує розмір і продуктивність програми.

Гнучкість та масштабованість: система легко розширюється та адаптується під нові вимоги.

Зручність підтримки: добре структурований код і сучасні інструменти полегшують супровід проекту.

Висновок до розділу 2

У цьому розділі проведено аналіз сучасних технологій та підходів до розробки веб-додатків, що відповідають поставленим завданням проекту. Розглянуто різні фреймворки, платформи та бази даних, які забезпечують модульність, гнучкість і масштабованість систем. Для реалізації проекту було обрано waw Framework від Web Art Work — мінімалістичний та модульний інструмент, що дозволяє ефективно створювати як фронтенд, так і бекенд рішення. Цей вибір обґрунтований можливістю швидкої розробки, легкого масштабування та підтримкою сучасних технологій, що відповідають вимогам системи. Проведений аналіз підтверджує доцільність застосування саме цих технологій для створення якісного та сучасного веб-додатку.

РОЗДІЛ 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Побудова плану інформаційної системи

Розробка інформаційної системи розпочалась зі створення детального плану, в якому були визначені основні структурні компоненти майбутнього проєкту: модулі, сторінки користувацького інтерфейсу та логічні зв'язки між ними. Такий підхід дозволив ще на початковому етапі сформуванню чіткого бачення архітектури системи та забезпечити її подальшу масштабованість і зручність у використанні.

Модулі

Кожен модуль відповідає за зберігання та обробку окремого типу даних, пов'язаних із домашніми тваринами або послугами догляду за ними. Було передбачено створення таких модулів:

- `pet` — зберігає базову інформацію про тварин, включаючи ім'я, стать, вік, породу, фото.
- `petrecord` — використовується для збереження історії хвороб, діагнозів, процедур, вакцинацій тощо.
- `petdoctor` — містить дані про ветеринарів, їхню спеціалізацію, контакти.
- `petclinic` — включає інформацію про клініки: назву, місцезнаходження, робочі години, методи зв'язку.
- `petallergy` — містить перелік алергій та їх опис, для зручного додавання тваринам.
- `petplace` — модуль для зберігання інформації про клініки.

Сторінки користувача

Для забезпечення зручного доступу до даних і взаємодії з системою було спроектовано наступні сторінки:

- `My Pets` — основна сторінка з переліком усіх тварин, які належать користувачу.

- New Home — сторінка з тваринами, які шукають нову родину.
- Pet Profile — детальна сторінка з повною інформацією про обрану тварину.
- Pet History — розділ з історією здоров'я, що відображає список процедур, що відбувалися з твариною.
- Pet Clinics — сторінка з переліком клінік з їх описом та характеристиками.
- Pet Doctors — сторінка для пошуку ветеринарів за місцезнаходженням.
- Admin pages — окремі сторінки адміністрування для кожного з модулів, що дозволяють додавати, редагувати або видаляти дані.

Сценарії використання

Під час планування також були змодельовані типові сценарії взаємодії користувача із системою. Це дозволило забезпечити логічне поєднання сторінок і модулів, а також сформувану очікувану поведінку системи:

- Створення профілю тварини, заповнення базових даних.
- Додавання подій до історії тварини (візити до лікаря, вакцинація).
- Пошук тварин, які потребують нового дому.
- Перегляд ветеринарних клінік.
- Використання пошуку ветеринарів.
- Управління інформацією з боку адміністратора.

Таким чином, уже на першому етапі розробки було чітко окреслено функціональну структуру майбутньої системи. Це забезпечило ефективне планування подальшої роботи та дозволило уникнути архітектурних помилок у майбутньому.

3.2. Створення модулів

Після затвердження архітектурного плану системи наступним етапом стало безпосереднє створення функціональних модулів. Кожен модуль реалізовує окрему сутність, яка зберігається у базі даних MongoDB[7], та має власну логіку обробки і відображення даних у фронтенді та бекенді. Для реалізації інтерфейсу адміністрування будуть використовуватись шаблонні компоненти, створені з використанням `waw Framework` [8].

Для створення модулів, використовується консольна команда:

```
waw add <назва модуля>
```

Далі потрібно обрати тип модуля і вказати його назву, що буде відображатися на фронтенді. Після чого новий модуль з'являвся за таким шляхом:

```
src/app/modules/<назва модуля>
```

Створення модуля **Pets**

Після створення модуля `Pets`, щоб його можна було зручно відкривати, потрібно додати його на фронтенд. Для цього відкривається файл з таким шляхом:

```
src/app/core/theme/user/user.component.html
```

де в компонент `"nav__burger"` додається наступний блок коду:

```
<a [routerLinkActiveOptions]="{ exact: true }"
routerLinkActive="_activeLink"
routerLink="/pets"
class="nav__burger-link"
(click)="hideSidebar()" >
<span class="material-icons">pets</span>
<span translate>Theme.Pets</span></a>
```

Він являється універсальним посиланням на сторінку модуля, назва якого вказується в рядку:

```
routerLink="/pets"
```

Створене посилання має такий вигляд (див. рис. 3.1).



Рис. 3.1 – Посилання на модуль

Сторінка модуля має свій базовий дизайн і виглядає так (див. рис. 3.2).

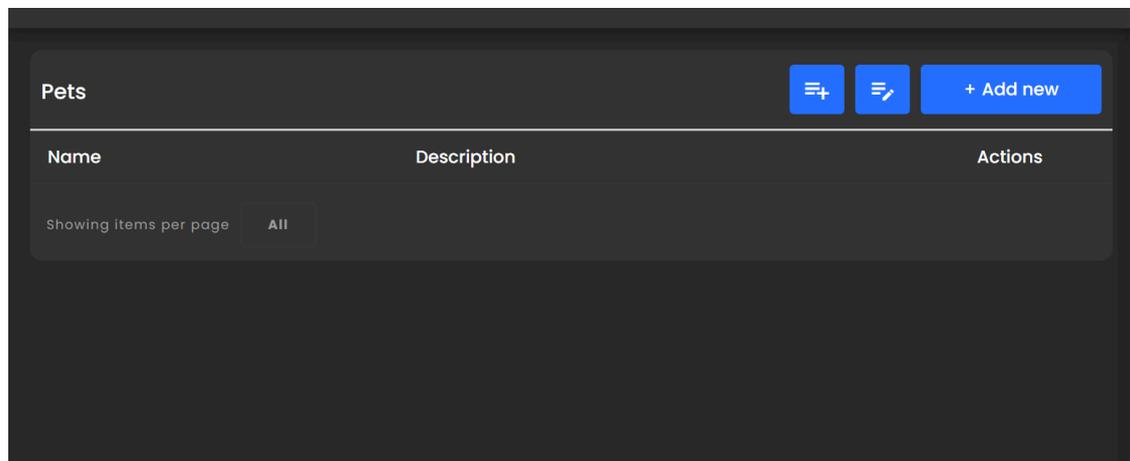


Рис. 3.2 – Сторінка модуля Pets

Зараз в документів є поля лише для вказування імені та опису, тому потрібно додати всі поля, що нам потрібно. Для цього в папці модуля, відкриваємо файл за таким шляхом:

```
interfaces/pet.interface.ts
```

Це файл інтерфейсу модуля, де й вказуються поля, з якими працює фронтенд. Додаємо наступні потрібні нам поля, щоб інтерфейс мав такий вигляд:

```
export interface Pet extends CrudDocument {
  name: string;
  thumb: string;
  species: string;
```

```

breed: string;
age: string;
gender: string;
description: string;
allergies: any;
adoptable: boolean;
author: string; }

```

Далі у папці модуля знаходимо файл за шляхом:

```
pages/pets.component.ts
```

Знаходимо змінну масиву `columns`, та прописуємо туди поля, які хочемо відображати на сторінці модуля.

```
columns = ['name', 'species', 'breed', 'age', 'gender', 'adoptable', 'description' ];
```

Тепер сторінка модуля Pets має такий вигляд (див. рис. 3.3).

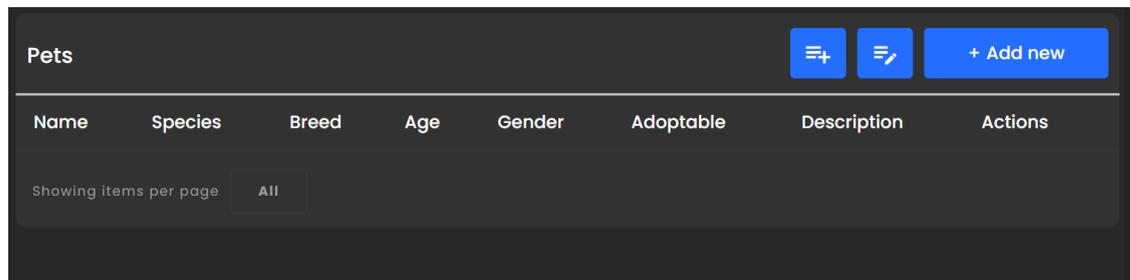


Рис. 3.3 – Сторінка модуля Pets

Тепер потрібно налаштувати спосіб додавання тварин. Для цього вже існує кнопка "Add new", але вона не містить потрібних полів (див. рис. 3.4).

The screenshot shows a modal form for adding a new pet. The form is titled 'Pet' and has a close button (X) in the top right corner. It contains two input fields: 'Name' with the placeholder text 'fill pet name' and 'Description' with the placeholder text 'fill pet description'. At the bottom of the form is a prominent blue button labeled 'Create'.

Рис. 3.4 – Форма модуля Pets

Щоб їх додати, потрібно в папці модуля, відкрити файл за таким шляхом:

```
formcomponents/pet.formcomponents.ts
```

Тут описаний об'єкт форми, який зберігає в собі інформацію про поля їх назву, підписи та тип. Такий вигляд має поле імені:

```
{
  name: 'Text',
  key: 'name',
  focused: true,
  fields: [
    {
      name: 'Placeholder',
      value: 'fill pet name'
    },
    {
      name: 'Label',
      value: 'Name'
    }
  ]
}
```

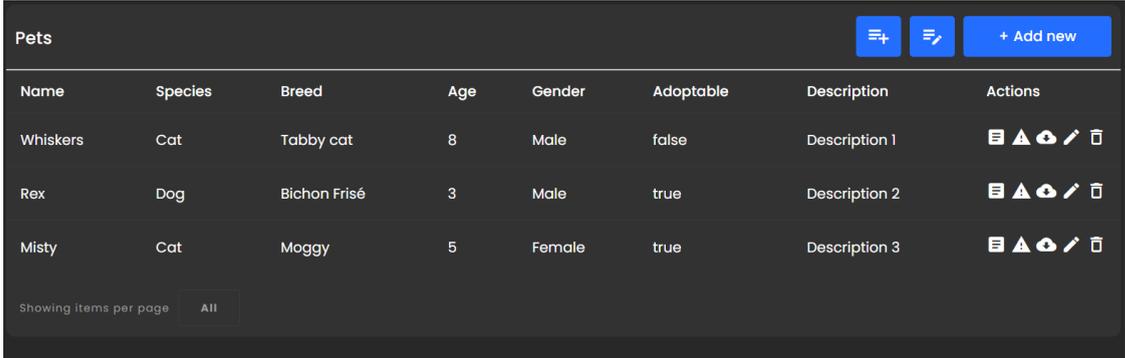
З подібних блоків формується форма з потрібними полями і в результаті має такий вигляд (див. рис. 3.5).

The screenshot shows a dark-themed form titled "Pet" with a close button (X) in the top right corner. The form is organized into sections:

- Name:** A text input field with the placeholder "fill pet name" and a close button (X).
- Species:** A dropdown menu with a plus icon and the text "choose pet species".
- Breed:** A dropdown menu with a plus icon and the text "choose pet breed".
- Age:** A text input field with the placeholder "fill pet age" and a close button (X).
- Gender:** A dropdown menu with a plus icon and the text "choose pet gender".
- Adoptable:** A checkbox that is currently checked, with the label "Adoptable".
- Description:** A text input field with the placeholder "fill pet description" and a close button (X).

Рис. 3.5 – Форма модуля Pets

Тепер є можливість створити перший запис тварини. Після створення він з'являється на сторінці модуля (див. рис. 3.6).



Name	Species	Breed	Age	Gender	Adoptable	Description	Actions
Whiskers	Cat	Tabby cat	8	Male	false	Description 1	   
Rex	Dog	Bichon Frisé	3	Male	true	Description 2	   
Misty	Cat	Moggy	5	Female	true	Description 3	   

Showing items per page: All

Рис. 3.6 – Сторінка модуля Pets

За таким принципом створюються всі модулі та додаються в меню (див. рис. 3.7).

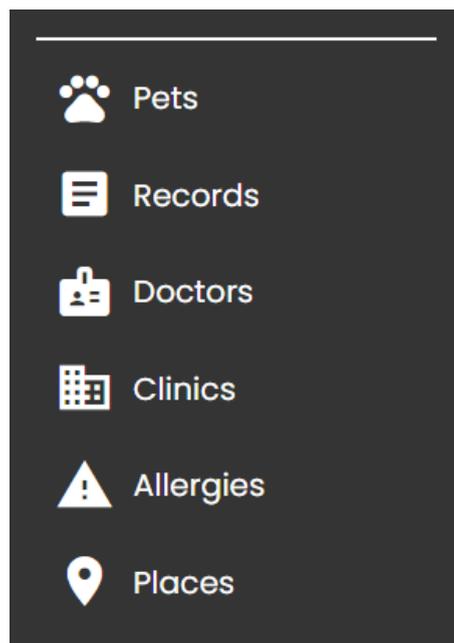


Рис. 3.7 – Меню модулів

Ці сторінки призначені лише для адміністрування і не відображаються в звичайних користувачів. Для них будуть створені окремі сторінки.

3.3. Формування зв'язків між модулями

Щоб створена тварина належала конкретному користувачу, і відображалася лише в нього в профілі, потрібно налаштувати зв'язки. В `Waw Framework`, вже реалізовано додавання ID користувача в документ під час створення. Для отримання списку тварин користувача, в файлі `"pets.component.ts"` будується запит до бекенду, що отримує лише записи, що належать цьому конкретному користувачу. Код запиту:

```
get rows(): Pet[] {
    return this._petService.petsByAuthor[this._userService.user._id];
}
```

Тепер при завантаженні сторінки, користувач отримуватиме лише записи про доданих ним тварин.

Схожий функціонал потрібно створити і для модуля `"petrecord"`, щоб створені записи були пов'язані з конкретною твариною. Для цього в інтерфейсі модуля `"petrecord"`, створено поле `"pet"` в яке зберігається ID тварини, щоб потім використовуючи `query` запит, отримувати записи конкретної тварини за її ID.

Щоб відкрити сторінку модуля з записами конкретної тварини, в стовпчику `"Actions"`, використовується кнопка у вигляді списку (див. рис. 3.8).

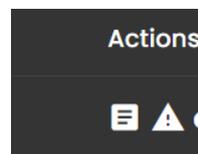


Рис. 3.8 – Кнопка сторінки з записами

Натискаючи на неї, відкривається сторінка модуля `"petrecord"`, яка має одну відмінність, в її посиланні вказується ID тварини, якій належать ці записи (див. рис. 3.9).

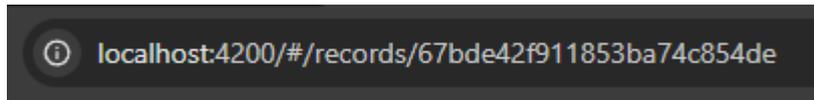


Рис. 3.9 – Посилання з ID

На самій сторінці видно записи цієї тварини з можливістю додавання нових та редагування наявних (див. рис. 3.10).

Name	Disease	Treatment	Procedure	Notes	Description	Actions
Pet Record 1.1	Pet Disease 1.1	Pet Treatment 1.1	Pet Procedure 1.1	Pet Notes 1.1		
Pet Record 1.2	Pet Disease 1.2	Pet Treatment 1.2	Pet Procedure 1.2	Pet Notes 1.2		

Showing items per page: 20 Page 1

Рис. 3.10 – Модуль з записами

Для реалізації цього функціоналу, було написано функцію, що отримує ID тварини з посилання при вході на сторінку:

```
this._route.paramMap.subscribe((params) => {
  this.pet_id = params.get('pet_id') || "";
});
```

Функція що створює запит до бекенду, використовуючи отриманий ID:

```
private _query(): string {
  let query = "";
  if (this.pet_id) {
    query += (query ? '&' : '') + 'pet=' + this.pet_id;
  }
  return query;
}
```

Після цього бекенд повертає всі записи, що містять в собі даний ID (див. рис. 3.11).

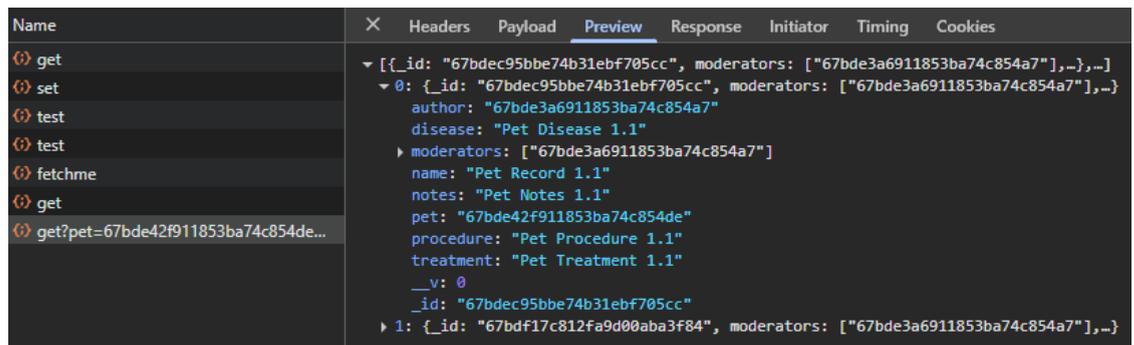


Рис. 3.11 – Відповідь бекенду

В такий самий спосіб створюються зв'язки для модуля "petdoctor" з модулем "petclinic". Де запис про лікаря, містить в собі поле "clinic" з ID лікарні, до якою він відноситься. Цей зв'язок буде використовуватись для фільтрації лікарів по лікарнях.

3.4. Розробка користувацьких сторінок

Користувацькі сторінки — це наступний етап роботи, де будуть розроблятися сторінки з якими будуть взаємодіяти користувачі. Для основних з них, будуть створені посилання в окремому меню вгорі сторінки чи внизу для мобільної версії.

Для створення таких сторінок, використовується консольна команда:

```
waw page user <назва сторінки>
```

Після створення, папка сторінки з'являється за таким шляхом:

```
src/app/pages/user/<назва сторінки>
```

Першою буде сторінка, що відображає список тварин користувача. На ній є плитки з іменем, фото тварини та деякими елементами керування (див. рис. 3.12).

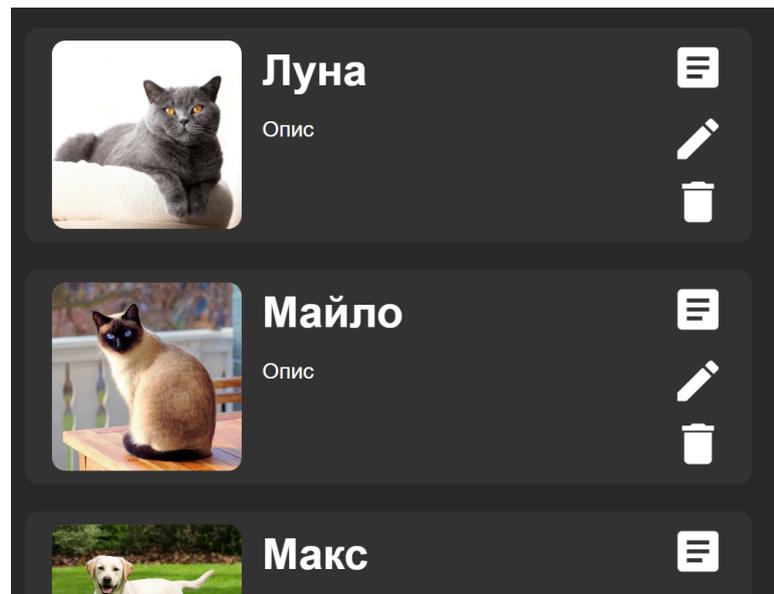


Рис. 3.12 – Сторінка тварин користувача

Список тварин користувача отримується в такий же спосіб, як це було описано в розділі про зв'язки модулів.

Далі створюється сторінка "retprofile". Перехід на сторінку відбувається по кліку на ім'я, або зображення тварини. На цій сторінці відображається докладна інформація по конкретній тварині (див. рис. 3.13).

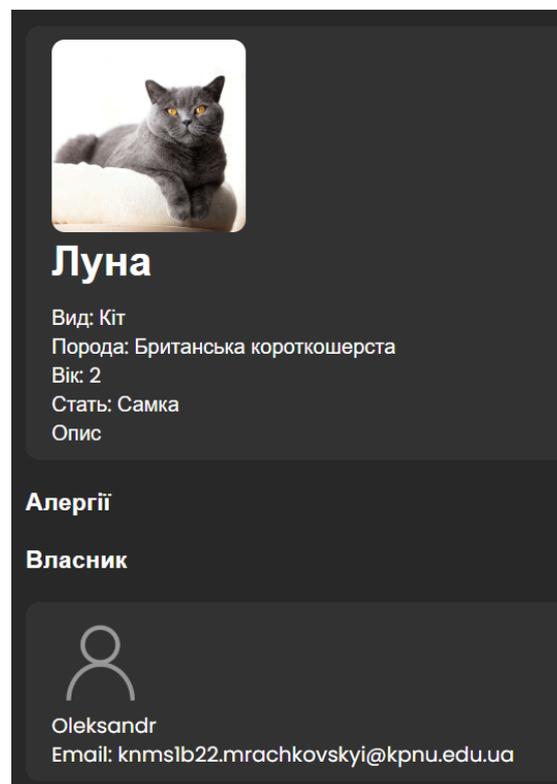


Рис. 3.13 – Сторінка з інформацією про тварину

Далі сторінка з записами тварин "pethistory". Для переходу на сторінку потрібно натиснути на іконку списку, навпроти тварин (див. рис. 3.14).

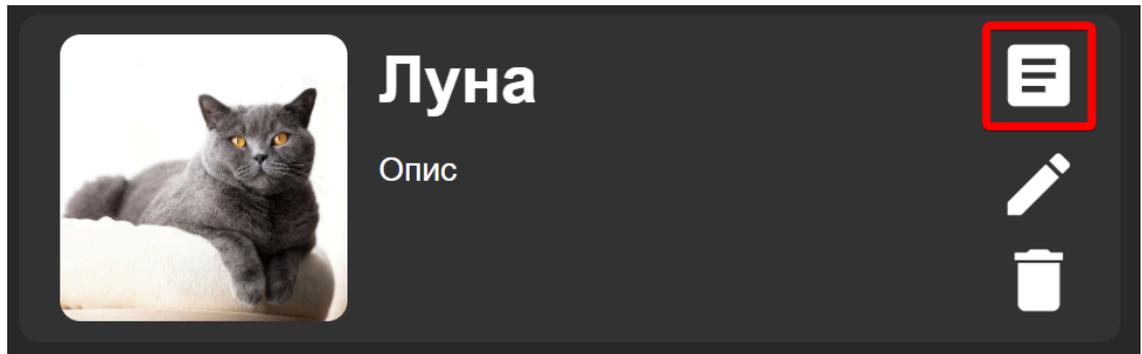


Рис. 3.14 – Кнопка для переходу на сторінку з записами

Записи відображаються у вигляді плиток з назвою, та датою додавання (див. рис. 3.15).

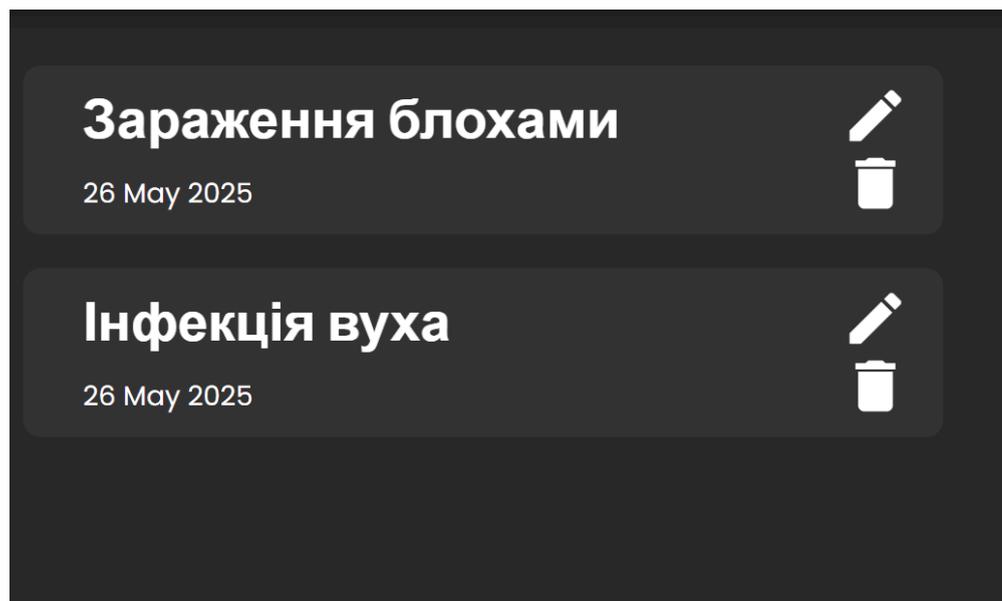


Рис. 3.15 – Список з записами

При натисканні, відкривається сторінка з докладною інформацією про конкретний запис (див. рис. 3.16).

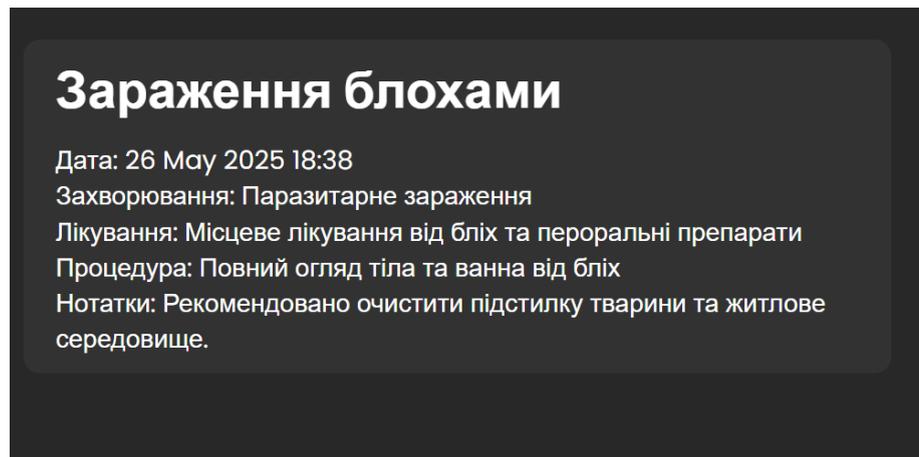


Рис. 3.16 – Список з записами

Таким чином реалізовано сторінки: "petclinic", "petdoctor", "petallergy", "newhome" та допоміжні для них сторінки. Після чого основні сторінки було додано в швидке меню (див. рис. 3.17).

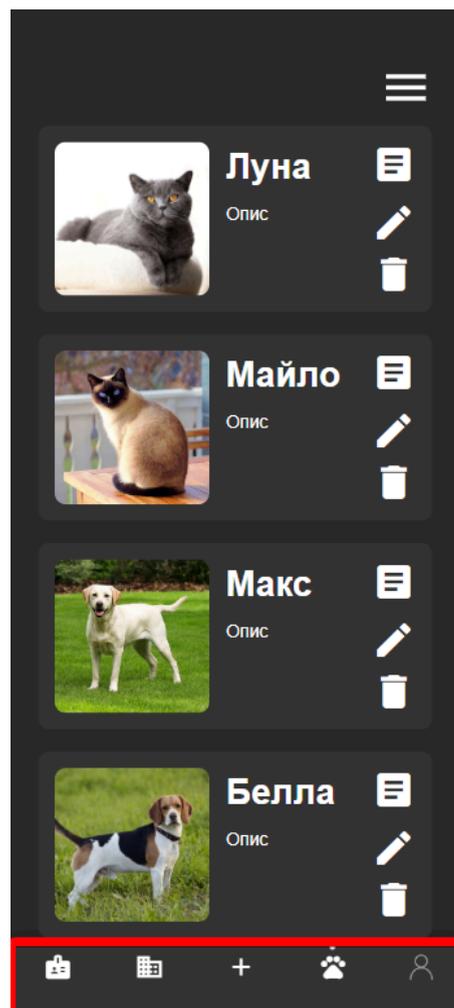


Рис. 3.17 – Швидке меню в мобільній орієнтації

Наступний крок, це додавання елементів на бічну панель сторінок, таких як: елементи створення та редагування, фільтри та пошук.

Для елементів створення та редагування, можна використовувати вбудовані кнопки "wbutton". Вони вже несуть в собі достатню кількість налаштувань, для зручної інтеграції на сторінку (див. рис. 3.18 - 3.19).

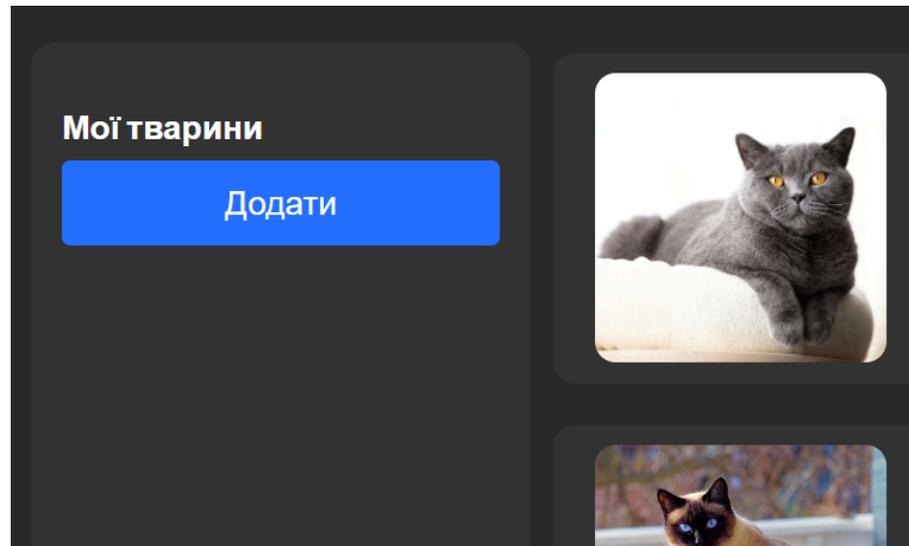


Рис. 3.18 – Кнопка створення профілю тварини

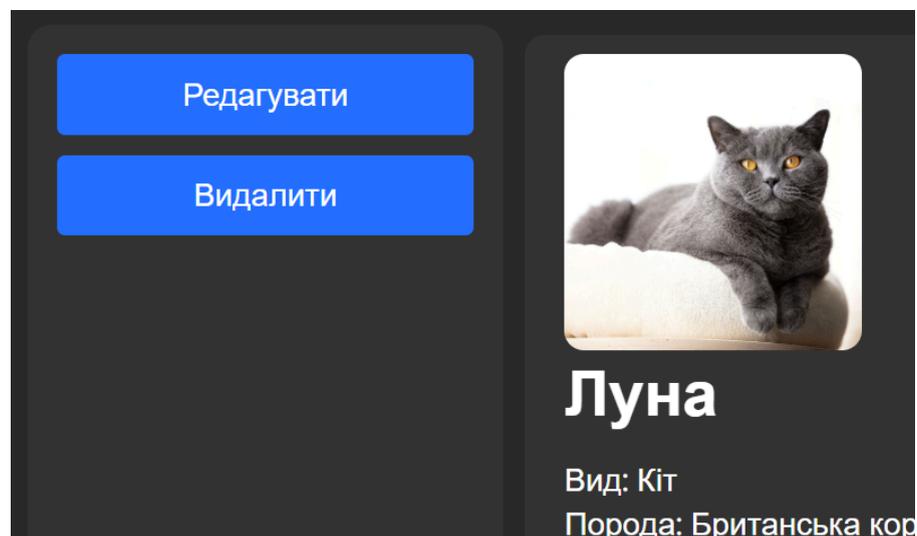


Рис. 3.19 – Кнопки редагування профілю тварини

Щоб додати фільтри використовується вбудований в waw Framework елемент, що називається "wselect". Він створює кастомний випадаючий список, в який можна вручну, або автоматично додавати елементи (див. рис. 3.20 - 3.21).

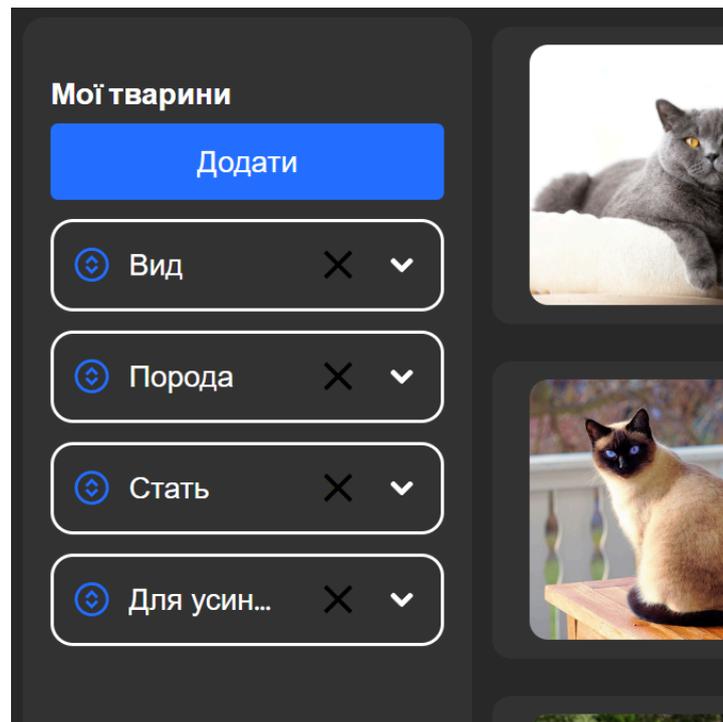


Рис. 3.20 – Елементи фільтрації

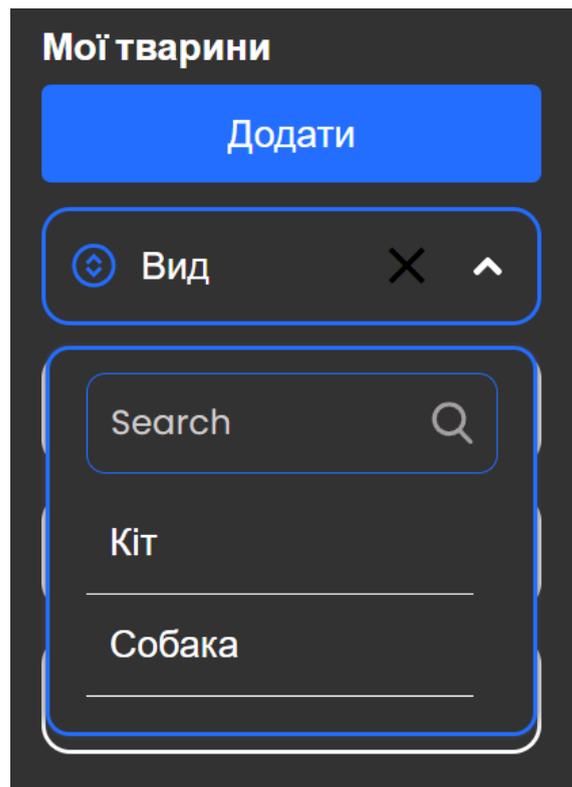


Рис. 3.21 – Елементи фільтрації

Для поля пошуку, використовується елемент "winput". Для самого пошуку, на бекенд відправляється query запит, де засобами mongodb, повертається результат пошуку (див. рис. 3.22).

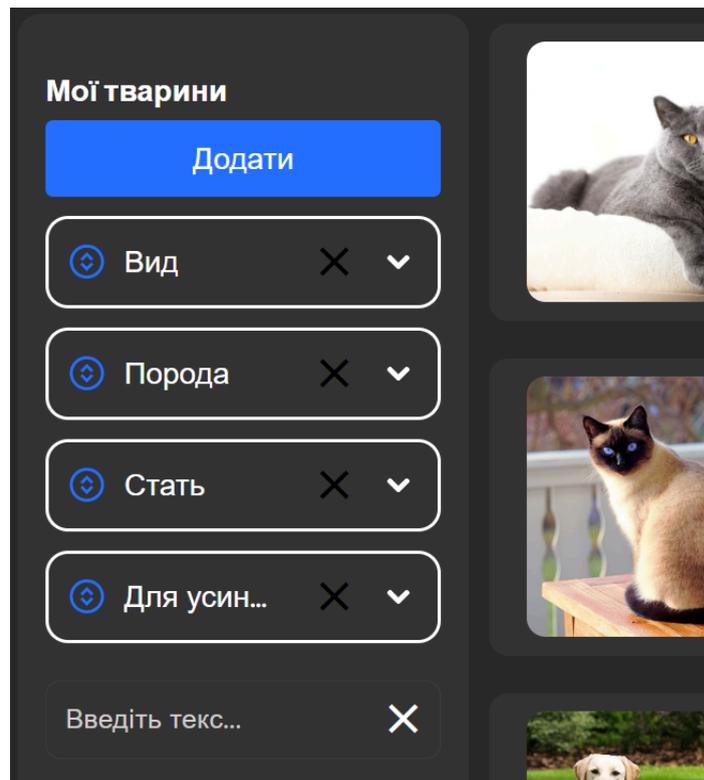


Рис. 3.22 – Поле пошуку

Додано функціонал додавання алергій на сторінці з детальною інформацією про тварину. Після вибору в випадаючому списку, вони з'являються у відповідному розділі профілю (див. рис. 3.23).

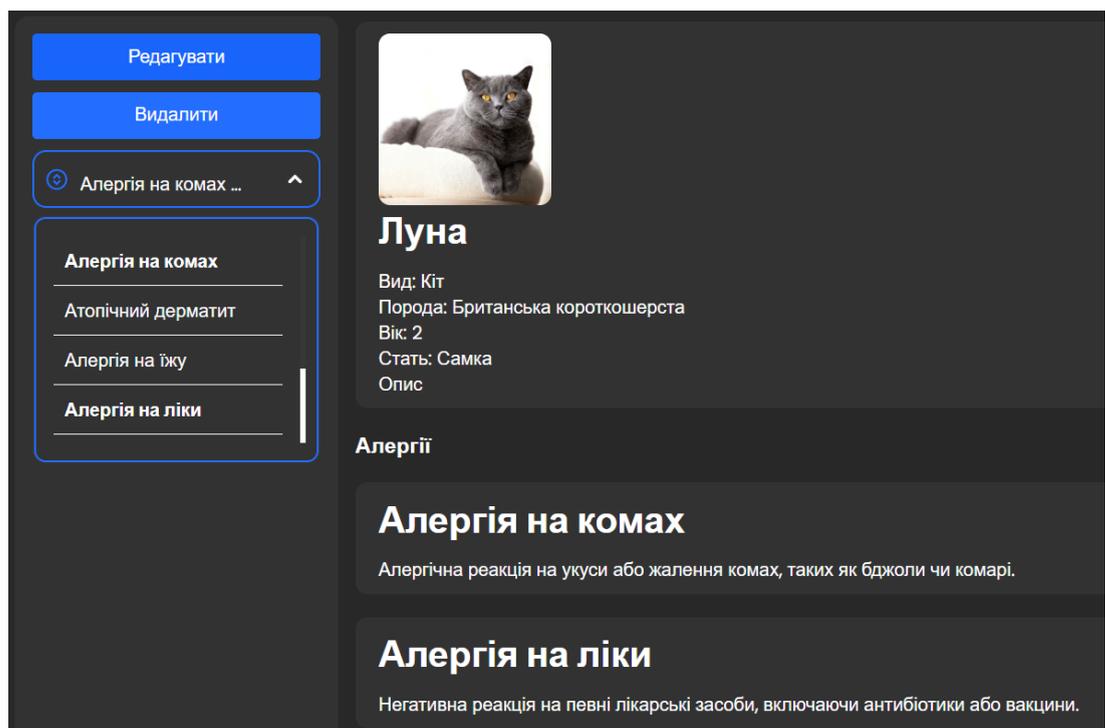


Рис. 3.23 – Список алергій

На сторінці тварин, що шукають новий дім, відображаються лише ті тварини, в яких стоїть статус "adoptable = true" (див. рис. 3.24).

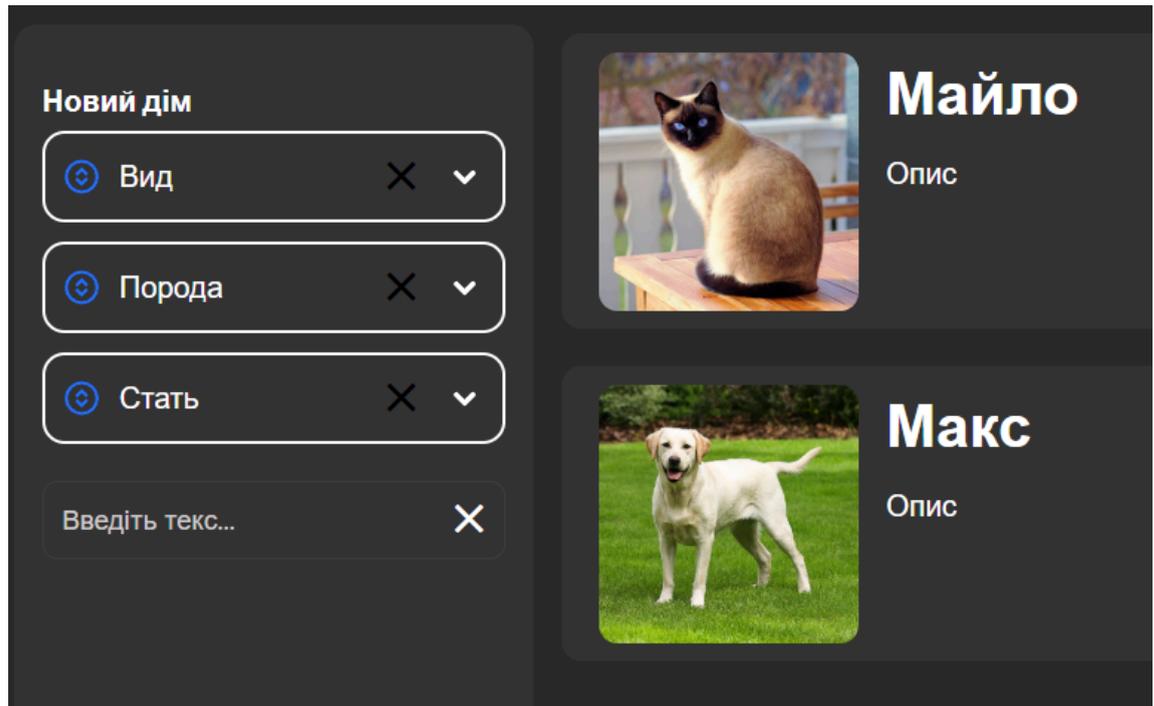


Рис. 3.24 – Список алергій

3.5. Налаштування базового дизайну сторінок

Для внесення редагувань у вигляд фронтенду, потрібно редагувати файли за такими шляхами:

src/scss/layout/_base.scss

src/app/core/theme/user/user.component.scss

В цих файлах знаходяться всі базові стилі сторінок[10]. Редагуючи їх, було створено зручний макет сторінок та налаштовано коректне відображення на пристроях різної форми екрану (див. рис. 3.25 - 3.26).

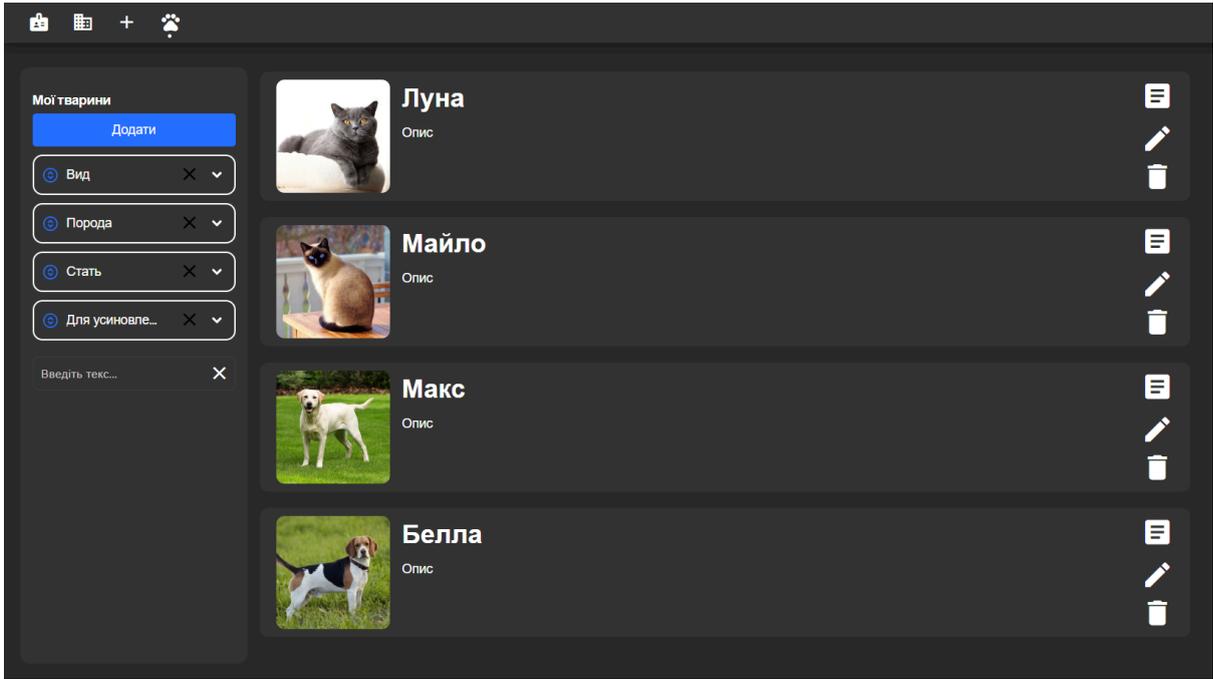


Рис. 3.25 – Сторінка з тваринами в браузері

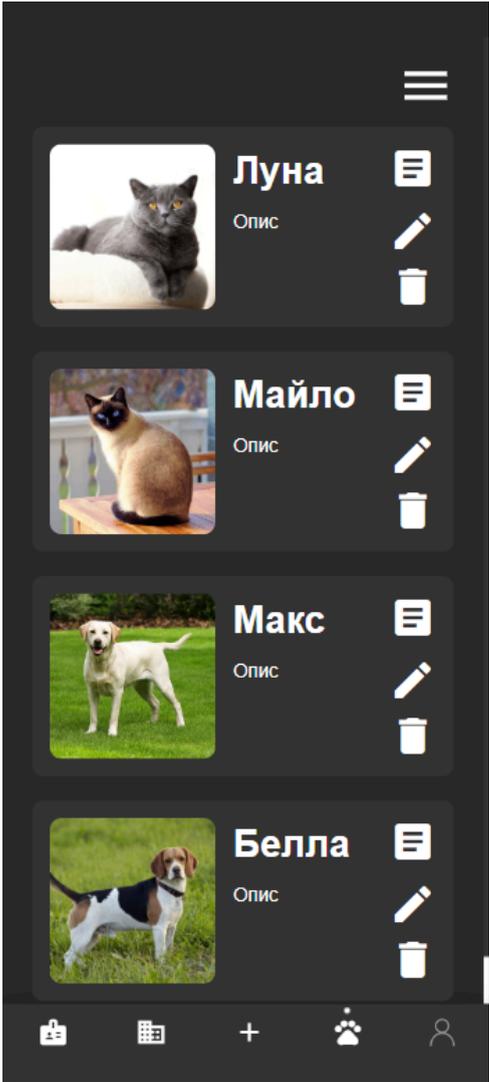


Рис. 3.26 – Сторінка з тваринами в мобільній орієнтації

3.6. Створення APK для мобільного телефону

Для створення APK, спочатку потрібно додати пакет Capacitor[8].

Команди для цього:

```
npm i @capacitor/core
```

```
npm i -D @capacitor/cli
```

Далі потрібно створити файл конфігурації, де потрібно вказати назву для APK та шлях до папки:

```
dist/app/browser
```

куди й створюється білд проекту після виконання команди:

```
npm run build
```

Далі потрібно встановити пакет Capacitor для Android та створити відповідний проєкт

```
npm i @capacitor/android
```

```
npm run capacitor add android
```

Щоб синхронізувати проєкти та всі залежності, виконується команда:

```
npm run sync
```

За умови, що Android Studio встановлено, виконується наступна команда:

```
npm run capacitor open android
```

Що запускає Android Studio[11] з проєктом. Далі в випадіючому меню потрібно обрати Build > Build App Bundle(s)/APK(s) > BuildAPK(s). Після очікування, з'являється повідомлення про завершення створення з посиланням на файл. На телефоні застосунок має наступний вигляд (див. рис. 3.27).

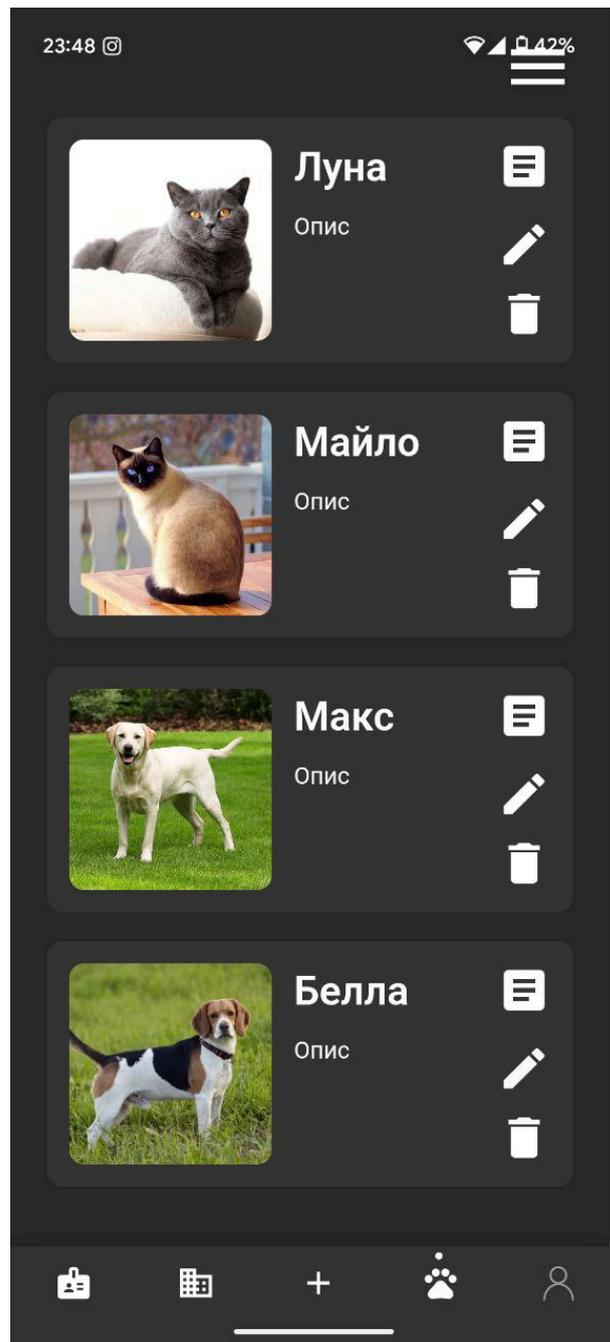


Рис. 3.27 – Сторінка з тваринами на телефоні

Висновки до розділу 3

Третій розділ, представляє детальний опис практичної реалізації інформаційної системи для надання сервісів догляду за домашніми тваринами. Робота над створенням системи відбувалася поетапно, починаючи з формування чіткого плану, що включав перелік необхідних модулів,

користувацьких сторінок та логічних зв'язків між ними. Такий підхід дозволив раціонально організувати процес розробки та забезпечити цілісність архітектури.

У межах реалізації було створено набір модулів (наприклад, pet, petrecord, petdoctor тощо), кожен з яких відповідає за зберігання та обробку специфічних даних, пов'язаних із тваринами або ветеринарними закладами. Використання інструментів waw Framework забезпечило швидке створення структур модуля та їхнє підключення до адміністративного інтерфейсу. Окрему увагу приділено формуванню логічних зв'язків між модулями - зокрема, між тваринами та їх історіями хвороб, лікарями та клініками, що дозволило реалізувати персоналізоване відображення даних.

Особливу роль відіграли користувацькі сторінки, які забезпечують інтуїтивну взаємодію кінцевого користувача із системою. Була реалізована навігація, перегляд профілів тварин, історії лікувань, пошук клінік та лікарів, а також окремий розділ для тварин, що потребують нового дому. Інтерфейс доповнено можливостями фільтрації, пошуку, додавання та редагування даних, що підвищує зручність користування.

Фінальним етапом стало створення APK-файлу мобільного застосунку за допомогою Capacitor, що дозволило забезпечити мультиплатформність і доступність розробленої системи для мобільних пристроїв.

Розділ демонструє основні етапи створення системи, які охоплюють ключові аспекти її функціонування - від структурування даних до реалізації візуального інтерфейсу та підготовки мобільної версії застосунку у форматі APK.

ВИСНОВКИ

У ході виконання дипломної роботи було розроблено інформаційну систему у вигляді веб-ресурсу та мобільного застосунку, що орієнтована на надання сервісів догляду за домашніми тваринами. Актуальність теми зумовлена потребою в цифрових інструментах для власників тварин і ветеринарних фахівців, оскільки більшість клінік в Україні не використовують спеціалізоване програмне забезпечення для ведення обліку та взаємодії з клієнтами.

У першому розділі було проаналізовано сучасний стан галузі, вивчено вітчизняний та зарубіжний досвід впровадження цифрових рішень у сфері ветеринарної медицини. Також розглянуто приклади наявних систем та засоби, що використовуються для їх реалізації, що дало змогу сформулювати ключові вимоги до майбутньої системи.

Другий розділ був присвячений вибору технологій, що стали основою реалізації системи. Після розгляду декількох варіантів було обґрунтовано доцільність використання `vue` Framework, розробленого компанією `Vue.js`, у поєднанні з `Angular` на фронтенді та `Express.js` і `MongoDB` на бекенді. Обраний стек технологій дозволяє забезпечити модульність, гнучкість, швидку розробку та можливість створення мобільної версії.

У третьому розділі було поетапно описано процес реалізації системи. Початково був побудований план із переліком модулів, сторінок та зв'язків між ними. Надалі реалізовано адміністрування даних через генерацію модулів, налаштування їх схеми, додавання до адміністративного інтерфейсу. Сформовано зв'язки між сутностями, створено користувацький інтерфейс із можливістю перегляду, фільтрації та редагування даних. Також підготовлено APK-файл для мобільної версії системи, що забезпечує її доступність з мобільних пристроїв.

Результатом роботи стала інформаційна система, що відповідає поставленим вимогам, має логічно спроектувати архітектуру та зручний інтерфейс. Отриманий досвід і реалізоване рішення створюють основу для подальшого вдосконалення системи та потенційного впровадження її у ветеринарну практику.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008:2015 «Звіти у сфері науки і техніки. Структура та правила оформлення». URL: https://science.kname.edu.ua/images/dok/derzhstandart_3008_2015.pdf
2. Л. М. Воєвідко, В. В. Кобильник ; [наук. ред. С. А. Копилов]. Методичні рекомендації з написання та оформлення дипломних робіт. (проектів) студентами Кам'янець-Подільського національного університету імені Івана Огієнка. Кам'янець-Подільський. 2018. URL: <http://elar.kpnu.edu.ua/xmlui/bitstream/handle/123456789/1641/Metodychni-rekomendatsii-z-napysannia-ta-oformlennia-dyplomnykh-robit%28proektiv%29-studentamy-K-PNU-im-I.Ohiiienka.pdf?sequence=1&isAllowed=y>
3. Методичні рекомендації з написання та оформлення дипломних робіт (проектів) студентами Кам'янець-Подільського національного університету імені Івана Огієнка. Кам'янець-Подільський. 2017. URL: <https://drive.google.com/file/d/1B9cMdn0Uz4ASvTmsj5YW4yFY-z85tJOO/view>
4. Методичні рекомендації до написання курсових робіт (проектів) у Кам'янець-Подільському національному університеті імені Івана Огієнка. URL: https://drive.google.com/file/d/1GiIJozc_1itWK0sZIXO29LFbccxJz-fz/view
5. Мрачковський О.Е. Розробка інформаційної системи у вигляді web-ресурсу та мобільного додатку для надання сервісів догляду за домашніми тваринами. *Збірник матеріалів наукової конференції за підсумками науково-дослідної роботи здобувачів вищої освіти фізико-математичного факультету у 2024-2025 н. р.* 2025. С. 54-56. URL: <https://fizmat.kpnu.edu.ua/studentski-konferentsii/>
6. Офіційна документація Angular. URL: <https://angular.dev/overview> (Дата звернення: 25.04.2025).

7. Офіційний сайт MongoDB. URL: <https://www.mongodb.com/> (Дата звернення: 26.04.2025).
8. Документація waw Framework. URL: <https://wawjs.wiki/docs> (Дата звернення: 28.05.2025).
9. Офіційна документація CapacitorJS. URL: <https://capacitorjs.com/docs> (Дата звернення: 16.05.2025).
10. W3Schools Online Web Tutorials. URL: <https://www.w3schools.com/> (Дата звернення: 18.05.2025)
11. Android Studio. URL: <https://developer.android.com/develop> (Дата звернення: 20.05.2025)