

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,  
управління та адміністрування  
Кафедра інформаційних технологій

**Кваліфікаційна робота магістра**

на тему: Веб-сервіс готелю для тварин із системою контролю їх здоров'я

Виконала студентка 2 курсу МІС-22  
групи \_\_\_\_\_  
спеціальності 122 Комп'ютерні науки  
Гордієвська Олександра  
Олександрівна

Керівник к.ф.-м.н.,  
Ткач Тетяна Борисівна

Рецензент к.т.н., доцент  
Гнатовська Г.А.

Одеса 2023

## АНОТАЦІЯ

Кваліфікаційна магістерська робота: 92 с., 6 табл., 64 рис., 1 дод., 24 джерел.

ВЕБ-СЕРВІС, PHP, БАЗА ДАНИХ, МОВА ПРОГРАМУВАННЯ, ОБ'ЄКТ, СИСТЕМА МОНІТОРІНГУ, ПРОГРАМНИЙ КОД, СЕРЕДОВИЩЕ РОЗРОБКИ, ІНТЕРФЕЙС.

Автоматизація в сфері догляду за тваринами, є актуальною через зростання кількості домашніх тварин. Веб-сервіси дозволяють спростити процеси та отримати миттєвий доступ до послуг, тому це є актуальним та важливим завданням, яке відповідає сучасним тенденціям розвитку технологій

Мета роботи – розробка та розміщення на власному домені веб-сервісу із подальшим оновленням.

Об'єкт роботи – створення БД за допомогою PHP та SQL, проектування інтерфейсу веб-сервіса, системи реєстрації та авторизації, розробка модулів для створення звітів.

Предмет роботи – функціональність та можливості бази даних, зручність та особливості інтерфейсу веб-сервісу, безпека та надійність механізмів реєстрації та авторизації, адаптивність та гнучкість модулів для створення звітів.

Задачі роботи: розробка веб-сервісу готелю для тварин, проектування інтерфейсу веб-сервісу, створення системи реєстрації та авторизації для забезпечення безпеки даних користувачів, розробка модулів створення звітів для отримання інформації про стан тварин.

В результаті виконання роботи, було створено веб-сервіс готелю для тварин, використовуючи PHP та SQL як основу розробки. Веб-сервіс включає в себе систему реєстрації та авторизації, модулі для створення звітів, інтерфейс користувача. Використання PHP та SQL дозволило створити гнучкий та масштабований веб-сервіс, який легко оновлюється та має можливість доповнюватися новими функціями в майбутньому.

## **ABSTRACT**

Qualification master's work: 92 p., 64 figures, 6 tables, 1 applications, 24 used literature.

WEB-SERVICE, PHP, DATABASE, PROGRAMMING LANGUAGE, OBJECT, MONITORING SYSTEM, PROGRAM CODE, DEVELOPMENT ENVIRONMENT, INTERFACE.

Automation in the field of animal care is relevant due to the increase in the number of pets. Web services simplify processes and provide instant access to services, making it a relevant and important task that meets modern technology development trends.

The purpose of the work – development and placement on its own domain of a web service with further updating.

The object of work – creation of a database using PHP and SQL, design of the web service interface, registration and authorization system, development of modules for creating reports.

The subject of work – functionality and capabilities of the database, convenience and features of the web service interface, security and reliability of registration and authorization mechanisms, adaptability and flexibility of modules for creating reports. Tasks of work – development of a web service for an animal hotel, design of the web service interface, creation of a registration and authorization system to ensure the security of user data, development of report creation modules to obtain information about the condition of animals.

As a result of the work, a web service for an animal hotel was created using PHP and SQL as the basis for development. The web service includes a registration and authorization system, modules for creating reports, a user interface. The use of PHP and SQL allowed to create a flexible and scalable web service, which is easily updated and has the ability to be supplemented with new functions in the future.

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....	7
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ .....	9
1.1 Веб-сервіси: поняття, принципи роботи, типи .....	9
1.2 Особливості розробки готелів для тварин.....	11
1.3 Аналіз системи віддаленого моніторингу здоров'я.....	13
1.4 Аналіз існуючих систем .....	20
1.5 Постановка завдання.....	28
2 ВИЗНАЧЕННЯ ВИМОГ ДО ВЕБ-СЕРВІСУ .....	30
2.1 Технічне завдання .....	30
2.2 Планування робіт .....	33
3 ПРОЕКТУВАННЯ СИСТЕМИ.....	40
3.1 Проектування інформаційної системи веб-сервісу за допомогою діаграм опису бізнес-процесів IDEF0 .....	40
3.2 Розроблення діаграми варіанту використання.....	46
3.3 Проектування бази даних веб-сервісу готелю.....	50
3.4 Проектування інтерфейсу користувача.....	58
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-СЕРВІСУ ГОТЕЛЮ ДЛЯ ТВАРИН.....	64
4.1 Аналіз існуючих засобів розробки .....	64
4.2 Керівництво користувача .....	77
4.3 Керівництво адміністратора.....	84
ВИСНОВКИ.....	89
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	92
ДОДАТОК А ВИХІДНИЙ КОД ПРОГРАМИ .....	93

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних.

СУБД – система управління базами даних.

ТЗ – технічне завдання.

HTML – HyperText Markup Language, Мова розмітки гіпертексту.

CSS – Cascading Style Sheets, Каскадні таблиці стилів.

PHP – Hypertext Preprocessor, Гіпертекстовий препроцесор.

XML – Extensible Markup Language, Розширювана мова розмітки.

SQL – Structured Query Language, Мова структурованих запитів.

SOAP – Simple Object Access Protocol, Протокол доступу до об'єктів.

POST – Power-On Self-Test, Перевірка апаратного забезпечення комп'ютера.

REST – Representational State Transfer, Передача репрезентативного стану.

WCF – Windows Communication Foundation, Платформа для створення сервіс-орієнтованих додатків.

WSDL – Web Services Description Language, Мова опису веб-сервісів та доступу до них.

UDDI – Universal Description, Discovery and Integration, Стандарт для індексації веб-сервісів.

IDEF0 – Function Modeling, Методологія функціонального моделювання.

UX – User Experience, Досвід користувача.

UI – User Interface, Користувацький інтерфейс.

SEO – Search Engine Optimization, Пошукова оптимізація.

GPS – Global Positioning System, Система глобального позиціонування.

## ВСТУП

Вдосконалення процесів догляду за тваринами стає важливим етапом у світі сучасних технологій. Ця сфера постійно розвивається, використовуючи новітні технології для покращення якості турботи про тварин та поєднуючи в собі розробку, впровадження та популяризацію інноваційних підходів до методів догляду про домашніх улюбленців. Одним з таких напрямків є впровадження веб-сервісів, які допомагають автоматизувати та оптимізувати процеси догляду. Веб-сервіс готелю для тварин із системою контролю їх здоров'я - це інноваційний підхід, який не просто забезпечує комфорт для тварин, але й допомагає власникам краще контролювати стан здоров'я своїх тварин.

Галузь домашнього догляду за тваринами стає важливим елементом взаємин між людьми та їхніми тваринами, оскільки визначається не лише прагненням господарів забезпечити комфорт та задоволення своїх улюбленців, а й відповідальністю за їхнє здоров'я та благополуччя. З появою сучасних підходів до догляду, мобільні додатки, онлайн платформи для медичного моніторингу та засоби відстеження стали не лише зручними інструментами для господарів, але й невід'ємною частиною готелей для тварин. Цей напрямок також визначається інноваційними методами надання послуг, такими як онлайн консультації з ветеринарами та індивідуальні рекомендації щодо харчування та активності тварин.

Метою кваліфікаційної роботи є розробка веб-сервісу готелю для тварин, проектування структур даних, створення програмного коду, розроблення інтерфейсу, проведення тестування ключових етапів. Для реалізації мети дипломної роботи визначено наступні завдання:

- провести порівняльний аналіз систем моніторингу здоров'я;
- визначити вимоги до ключових аспектів веб-сервісу;
- спроектувати базу даних з продуманим зв'язком між елементами;
- дослідити основні етапи створення веб-сервісів;
- за допомогою проаналізованих даних створити веб-сервіс.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ

## 1.1 Веб-сервіси: поняття, принципи роботи, типи

Веб-сервіси є невід'ємною частиною сучасного цифрового світу, оскільки вони охоплюють практично усі сфери діяльності нашого життя [1]. Вони розширюють можливості взаємодії з іншими людьми, інформацією та послугами. Ці сервіси не тільки впливають на наше сприйняття світу, а і роблять його більш доступним та зручним.

Веб-сервіс – це програмний компонент, що надає можливість іншим користувачам взаємодіяти один з одним, а також з інформацією та послугами через Всесвітню павутину. Веб-сервіси діють на основі стандартних веб-протоколів, таких як HTTP (Hypertext Transfer Protocol, який передає веб-сторінки) та може включати в себе різні функції. Веб-сервіси є основою багатьох онлайн-платформ, додатків і послуг, які використовуються щодня і надають можливості отримувати та надавати послуги, розширюючи функціональність для користувачів.

Веб-сервіси представляють собою програмні компоненти, розміщені на віддалених серверах, які дозволяють системам і користувачам здійснювати взаємодію через мережу Інтернет. Ця взаємодія відбувається за допомогою стандартних веб-протоколів, які визначають правила комунікації між клієнтом (користувачем або іншою системою) та сервером, на якому розташований веб-сервіс.

Веб-сервіс обробляє отриманий запит і виконує необхідні операції. Операції можуть бути різними – від отримання і відправлення даних до виконання спеціалізованих функцій. Результат таких операцій формується у відповідь, яка містить необхідні дані, а потім надсилається користувачеві або системі, яка ініціювала запит.

Веб-сервіси мають здатність працювати на віддалених серверах, а це означає, що користувачам не потрібно мати локальні копії програм або баз даних, а вони можуть отримувати доступ до функцій та інформації через мережу

Інтернет. А враховуючи те, що веб-сервіси використовують стандартні веб-протоколи, такі як HTTP, що робить їх сумісними з різними платформами та мовами програмування, такими як Java, Python, PHP. Веб-сервіси можуть підтримувати різні методи або операції, які виконуються за допомогою запитів: GET-запит може використовуватися для отримання даних, а POST-запит для відправки даних на сервер.

Веб-сервіси бувають різних типів, кожен із яких використовується для різних завдань та визначається унікальністю функціоналу, а також має свої переваги та недоліки:

- Веб-сервіси на базі SOAP (Simple Object Access Protocol) і WCF (Windows Communication Foundation) протоколів. В даному випадку передача повідомлення, яке кодується на основі обраного протоколу, відбувається у форматі XML (EXtensible Markup Language), а взаємодія – методом POST (Power-On Self-Test). Перевагами даних протоколів є те, що об'єкти кодуються за допомогою WSDL-схеми веб-сервіса. А недоліком є створення конфігурації з об'єктами високого рівня, які виконують усі низко рівневі операції, а також підключення може вимагати багато часу.
- Веб-сервіси на мові WSDL (Web Services Description Language). Веб-сервіс неможливо використовувати, якщо невідомо його місцезнаходження, тому клієнт, який викликає веб-сервіс, повинен знати, де знаходиться веб-сервіс. Також додаток клієнта повинен знати, роботу веб-сервісу для виклику потрібного веб-сервісу. Це досягається за допомогою WSDL, відомого як мова опису веб-сервісів. Файл WSDL, який є XML-документом, повідомляє клієнтському додатку про роботу веб-сервісу, що дозволяє також зрозуміти спосіб використання сервісу.
- Веб-сервіси на базі UDDI (Universal Description Discovery & Integration). Описує, шукає та публікує, а також надає інформацію про веб-сервіси. UDDI надає репозиторій, в якому можуть розміщуватися файли WSDL, а також діяти як база даних, що містить всі файли WSDL. В реєстрі UDDI



знаходиться необхідна інформація для веб-сервісу для знаходження клієнтським додатком.

- Веб-сервіси в архітектурі REST (Representational State Transfer). REST визначає правила взаємодії з серверами та описує вхідні та вихідні об'єкти для веб-сервісу. Вхідними параметрами можуть бути рядки, числа, дати. Вихідні параметри можна передавати через протокол HTTP, такі як XML. В більшості випадків вистачає взаємодії з сервером через HTTP-протокол та мережевих портів за замовчуванням. У архітектурі просте налаштування взаємодії мережі та великі можливості для формування запитів.

## **1.2 Особливості розробки готелів для тварин**

В наш час зростає популярність готелів для тварин, оскільки люди стали більше дбати про своїх домашніх улюбленців та перейматися за місця для їх перебування під час своєї відсутності. Для забезпечення власникам тварин спокою та можливості відстежувати стан своїх улюбленців на відстані, існують веб-сервіси готелів для тварин. Такі веб-сервіси дозволяють власникам реєструвати своїх тварин, бронювати місця в готелі, замовляти послуги для своїх домашніх тварин, відстежувати їх здоров'я та отримувати звіти про їх стан.

Розглядаючи важливі аспекти веб-сервісів готелів для тварин з наданням різних послуг та контролем стану тварин, треба враховувати широкий спектр технічних та функціональних деталей та вимог [2]. Для кращого розуміння та аналізу особливостей розробки таких сервісів, треба провести докладний огляд кожного аспекту:

- Реєстрація та авторизація. Даний аспект передбачає, що користувачі можуть створювати облікові записи на веб-сервісі, вказуючи свої особисті дані та вибираючи логін та пароль для безпечного доступу до даних. Після реєстрації користувачі зможуть авторизуватися на сервісі, вводячи свої дані, після цього отримати доступ до особистого профілю та

особистого кабінету, де вони зможуть переглянути усі подробиці про свою тварину.

- Система бронювання. Користувачам надається можливість вибору дат перебування своїх тварин у готелі, вибору типу номеру та послуг для тварин, вибір виду тварин, а також додаткових побажань щодо догляду за ними. Після вибору параметрів, система перевіряє доступність місць та обраної дати, надсилає підтвердження бронювання, а користувач може оплатити послуги та проживання після огляду ветеринаром.
- Моніторинг стану здоров'я та надання послуг для тварин. Даний аспект включає в себе використані типи датчиків, такі як термометри та датчики руху, технічні деталі їх збору даних та передача до системи моніторингу в реальному часі. Послуги для тварин можуть включати в себе регулярні медичні огляди у ветеринара, годування за розкладом, прогулянки на свіжому повітрі, а система моніторингу може спостерігати за активністю тварини та, в разі зміни її здоров'я, надсилати сповіщення ветеринару, а власник може передивлятися ці дані у себе в кабінеті.
- Оповіщення. В цьому аспекті розглядаються методи і засоби, які веб-сервіс використовує для надання інформації користувачам про стан їх тварин. Це можуть бути електронні листи, повідомлення через месенджери або через дзвінок від адміністратора.

Розробка готелю для тварин включає в себе не лише технічні аспекти, а також і ряд організаційних та функціональних особливостей. Аналіз цих аспектів дає можливість створити повноцінний готель для тварин, що задовольняє всі потреби клієнтів та їх улюбленців:

- Забезпечення комфорту для тварин. Готель повинен бути максимально комфортним для тварин. Це включає в себе відповідне розташування номерів для кожного виду тварин, зручні умови для їх відпочинку, проживання та розваг, а також мати місце для прогулянки під наглядом персоналу готелю.

- Безпека. Важливо мати системи нагляду та безпеки, які дозволяють контролювати тварин. Це включає в себе відео спостереження, а також охорону, яка забезпечить спокій тварин. Власники тварин можуть мати доступ до відео зі своїми улюбленцями. Додатково є можливим використання системи сигналізації для оповіщення про надзвичайні ситуації.
- Годування, вода та дієта. Готель повинен надавати годування для тварин відповідно до дієти та потреб кожної тварини. Це включає в себе регулярне подавання їжі, доступ до свіжої води та врахування спеціальних дієтних обмежень або потреб.
- Гігієна та догляд. Для забезпечення комфорту та здоров'я тварин готель повинен мати спеціальні приміщення для миття та догляду за тваринами. Ці приміщення повинні бути обладнані відповідними засобами для догляду, миття та грумінгу тварин.
- Медичний догляд. Готель повинен надавати ветеринарну допомогу тваринам. Це включає в себе першочерговий огляд, відвідування ветеринара за гарфіком, надання лікування та догляду в разі захворювань або травм.

Всі вищезазначені аспекти є важливими для успішної розробки та створення готелів для тварин, оскільки вони сприяють створенню сприятливого та безпечного середовища для домашніх улюбленців та власників. Детальний аналіз цих аспектів сприяє забезпеченню якісного обслуговування та задоволенню всіх потреб клієнтів та їхніх тварин, а додаткові аспекти допомагають краще задовольнити вимоги пухнастих улюбленців.

### **1.3 Аналіз систем віддаленого моніторингу здоров'я**

У сучасному світі існує велике розмаїття систем моніторингу за станом домашніх тварин, що робить вибір справжнім викликом. Кожна з таких систем

має свої особливості та пропонує свій ряд унікальних функцій, які базуються на індивідуальних потребах користувачів. Серед систем моніторингу зустрічаються прості рішення, вони спрямовані на базовий контроль за місцезнаходженням та станом здоров'я тварин [3]. Деякі з них не мають власних веб-додатків, а інформацію можна переглядати на спеціальних пристроях, які показують мапу з місцем перебування тварин. Важливими параметрами при виборі пристроїв для користувачів є автономність, тривалість функціонування, дальність працездатності та простота користування, а також можливість розширення функціоналу для більш зручного відстеження та догляду за тваринами.

Сучасні технології пропонують широкий вибір рішень для моніторингу та догляду за улюбленими тваринами. Зростання цього ринку призводить до появи інноваційних пристроїв, які забезпечують високий рівень функціональності, включаючи відстеження місцезнаходження, моніторинг здоров'я та багато інших функцій. Різноманітність пропозицій на цьому ринку є вражаючою, що вказує на постійний розвиток технологій для догляду за улюбленими тваринами. Спостерігається зростаючий інтерес до таких технологій і це вказує на те, що вони стають необхідним елементом для сучасних власників домашніх улюбленців, надаючи їм засоби для більш ефективного та зручного догляду за своїми тваринами (рис. 1.1).

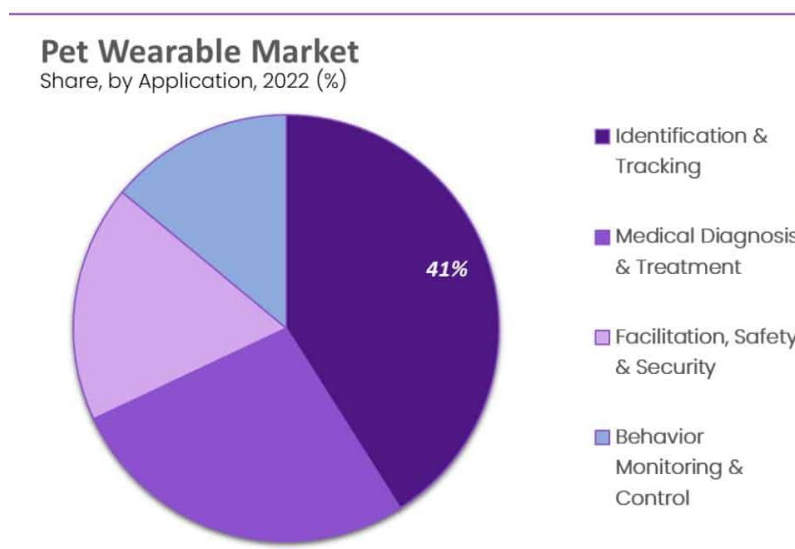


Рисунок 1.1 – Аналіз ринку носимих пристроїв для тварин станом на 2022 рік

Існують різноманітні системи відстеження для тварин на сучасному ринку. Такі системи частіше за все використовують для відстеження своїх домашніх улюбленців пристрої з GPS, а також радіочастотні технології та Bluetooth [4]. Для кожного виду пристроїв відстеження домашніх тварин характерні певні переваги і недоліки, тому вибір оптимальної технології можна здійснити, враховуючи умови і потреби господарів тварин. Серед факторів, які слід враховувати, є якість послуг мобільного зв'язку у регіоні, бажані функції та дальність відстеження тварини.

Радіопристрої – пристрої відстеження з використанням радіочастот (РЧ). Такі пристрої дозволяють господарям визначити місцеположення своїх домашніх тварин за допомогою радіохвиль. Вбудований передавач на пристрої висилає сигнал на приймач та таким чином повідомляють про відстань та напрямок улюбленця. Така технологія незалежна від мобільного зв'язку, що зручно у віддалених місцях, але сигнали можуть блокуватися будівлями чи лісовими ландшафтами, що може ускладнювати пошук тварини через неточність даних місцеположення. Оскільки радіосигнали не передають конкретні координати, доведеться мати при собі приймач, а такий пристрій може бути досить непрактичним чи тяжким, але більшість моделей вже полегшені. Велика кількість радіопристроїв може мати вбудовані датчики, які вимірюють фізичну активність тварини, що може включати кількість кроків, відстань, яку вони подолали, та навіть рівень активності у порівнянні з нормою для їхньої породи та розміру. Також можуть бути враховані такі параметри, як температура тіла та пульс, Усі ці дані власник може переглянути на пристрої чи на телефоні. Частіше за всіх радіопристрої використовують власники великих домашніх тварин, за якими потрібен більш складний догляд (рис. 1.2).

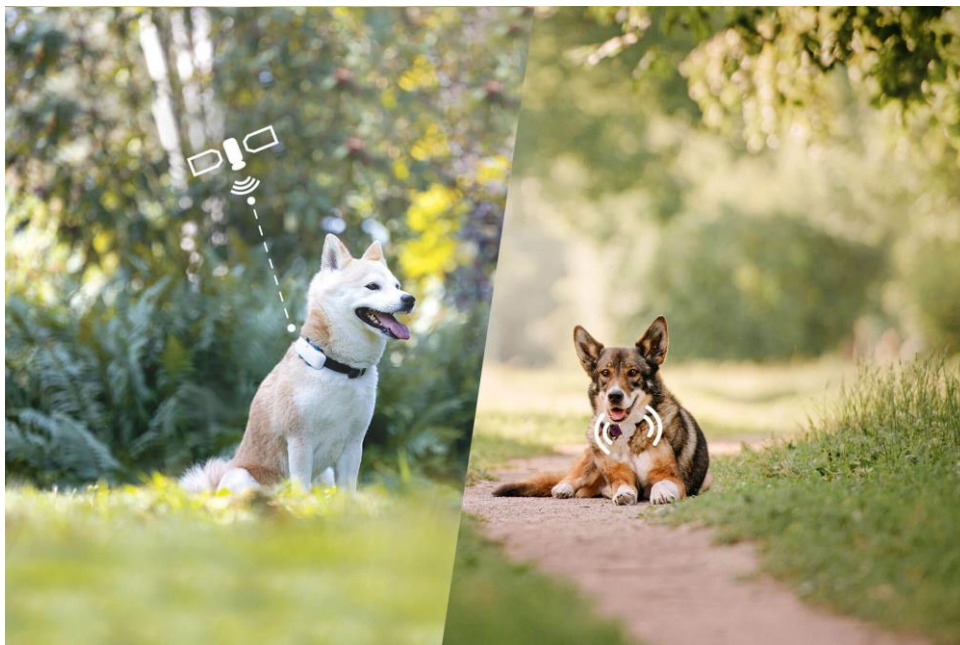


Рисунок 1.2 – Пристрої відстеження з використанням радіочастот

Bluetooth-трекери – вдосконалений інструмент для власників тварин, він дозволяє стежити за своїми улюбленцями в межах дому чи невеликих територій та надає можливість встановлювати зв'язок із смартфоном через технологію Bluetooth (бездротова технологія, яка працює на короткій відстані та забезпечує бездротовий обмін даними між цифровими пристроями). Завдяки компактному дизайну і невеликому акумулятору, трекер є зручним в експлуатації, але не може підходити тваринам, які прогулюються на відкритому просторі, оскільки Bluetooth-трекер (рис. 1.3) може виявитися неефективним на великих відстанях. Такий трекер чудово підходить для домашніх умов для невідкладного виявлення місцезнаходження на невеликій території. Bluetooth-трекери для домашніх тварин є вдосконалим інструментом, що дозволяє власникам не лише стежити за місцеположенням своїх улюбленців в межах дому чи невеликих територій, але і забезпечують ряд корисних функцій для догляду за тваринами. Такі трекери можуть встановлювати зв'язок із смартфоном власника, а невеликий акумулятор роблять їх ідеальними для використання в домашніх умовах. Існують моделі обладнані датчиками температури та вологості для контролю за комфортом

тварини. Трекери можуть відстежувати харчування та надавати власникам інформацію щодо раціону тварини, а деякі також забезпечують нагадування про вакцинації та лікування, а можливість взаємодії з іншими власниками тварин через спеціальні додатки робить ці пристрої більш соціально орієнтованими.



Рисунок 1.3 – Bluetooth-пристрої для відстеження стану домашніх тварин

Відстеження домашніх тварин за допомогою технології Wi-Fi – Технологія Wi-Fi (Wireless Fidelity – бездротова правдивість відтворення) також використовується у пристроях відстеження домашніх тварин. За допомогою Wi-Fi можна аналізувати потужність сигналу найближчих бездротових мереж, а потім використовувати ці дані для точного визначення місцезнаходження тварини, але цей метод залежить від наявності бездротових мереж у навколишньому середовищі, тому він не є ефективним у випадках слабого сигналу або

відсутності доступу до бездротових мереж. Деякі пристрої відстеження вимірюють кількість кроків, витрачених калорій, а також інші параметри, які вказують на активність чи пасивність тварини протягом дня. Також такі пристрої можуть використовувати сенсори для вимірювання серцевого ритму улюбленця, що допомагає в ранньому виявленні проблем з серцево-судинною системою чи стресом. Аналіз активності та рухи тварини під час сну надають інформацію про якість та тривалість сну улюбленця. Існують також розумні ваги для тварин, які можуть бути інтегровані з системами Wi-Fi для відстеження змін у вазі тварини, що допомагає у визначенні загального стану здоров'я та дієти. За допомогою технології Wi-Fi пристрої можуть взаємодіяти з системами харчування, допомагаючи власникам відстежувати кількість споживаних калорій, контролювати дієту тварини та виявляти алергії чи інші харчові проблеми. В деяких випадках системи можуть надсилати медичні сповіщення власникам, якщо вони виявляють аномалії у фізичному стані тварини або якщо виникають певні проблеми із показниками (рис. 1.3).



Рисунок 1.4 - Wi-Fi-пристрої для моніторингу тварин



GPS (Global Positioning System) відстеження домашніх тварин – ця інноваційна технологія надає надзвичайно ефективний спосіб відстеження домашнього улюбленця, оскільки доставляє точні дані про його місцезнаходження безпосередньо на смартфон власника тварини. Такий пристрій використовує сигнали від супутників та мобільних веж для точного визначення положення тварини, забезпечуючи охоплення значної території, але слід враховувати його обмеженість у випадках низької якості сигналу. Окрім цього, важливо враховувати той факт, що батарея пристрою може швидко розряджатися через регулярні та інтенсивні передачі даних. Відстеження за допомогою GPS-пристрою (рис. 1.5) дозволяє не лише визначати місцезнаходження улюбленця, але й надати ряд додаткових функцій, таких, як моніторинг фізичної активності, історія переміщень, тривалість і якість сну, а деякі передові моделі включають можливості взаємодії з іншими власниками тварин та участь у спільнотах. Цей досвід відкриває нові можливості для активної взаємодії господарів тварин, що надає змогу обміну досвідом між власниками. GPS-відстеження домашніх тварин також може бути використане для конкретного моніторингу їхнього здоров'я. Деякі сучасні GPS-трекери обладнані додатковими сенсорами, які мають можливість вимірювати температуру тіла тварини, що дозволяє власникам вчасно виявляти будь-які ознаки захворювання або стресу, а деякі GPS-трекери мають вбудовані серцеві монітори, які слідкують за пульсом тварини. Відстеження серцевої активності може надати важливу інформацію про загальне здоров'я та фізичну форму улюбленця. Також такі трекери можуть включати функції моніторингу ваги та активності тварини, що може бути корисним для визначення рівня фізичної активності та контролю ваги. Додатково GPS-трекери можуть мати вбудовані датчики для визначення рівня вологості або погодних умов, що допомагає визначити власникам умови, в яких перебуває тварина.

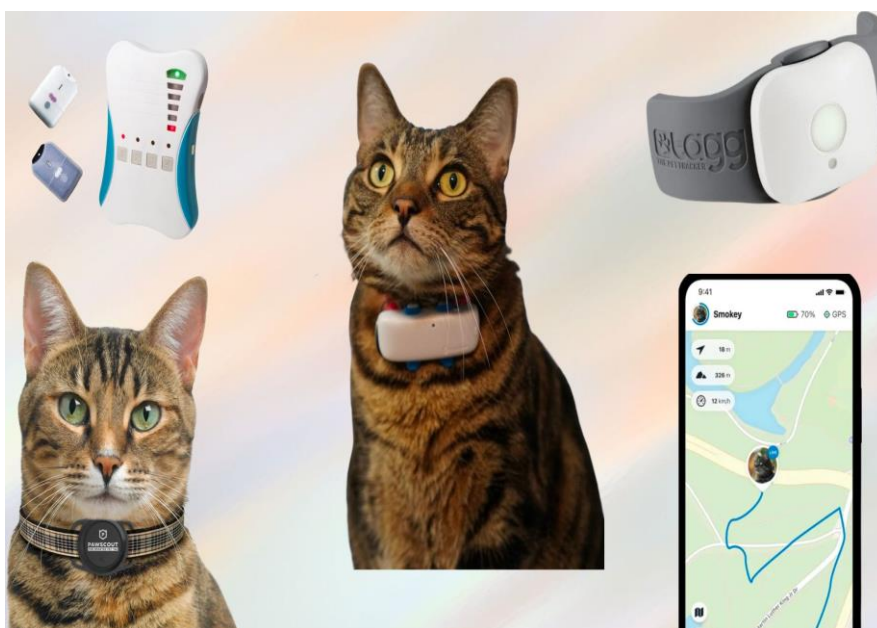


Рисунок 1.5 – GPS-пристрій для відстеження домашніх улюбленців

Трекери відстеження для тварин вирізняються своєю важливістю як засоби контролю та безпеки. Ці пристрої не лише дозволяють власникам швидко знайти своїх улюбленців у випадках втрати, але й забезпечують функції відстежування параметрів здоров'я. Власники можуть відстежувати фізичну активність, якість сну, пульс та інші показники, отримуючи таким чином дані для кращого розуміння потреб та покращення стану своїх тварин. Це сприяє оперативній реакції на можливі проблеми та забезпечує повноцінне та здорове життя домашнім улюбленцям. Такий підхід робить трекери необхідним інструментом для власників, які прагнуть забезпечити найкращий рівень догляду та уваги для своїх тварин.

#### 1.4 Аналіз існуючих систем

Сучасні технології відкривають нові можливості здійснення віддаленого моніторингу здоров'я тварин. Такі інноваційні системи використовують різноманітні технології для вивчення різних аспектів життя тварин, що допомагає власникам тварин у покращенні якості догляду та вирішенні їхніх потреб. Деякі

відомі системи віддаленого моніторингу здоров'я тварин використовують різноманітні підходи та технології для надання детальної інформації про стан та активність тварин [5].

Petpase Smart Collar (рис. 1.6) є передовим відстежувачем здоров'я для домашніх тварин, призначеним для надання власникам докладної інформації про фізичний стан та поведінку пухнастих створінь. Цей носимий пристрій забезпечує високоточний моніторинг ключових параметрів здоров'я.

В цьому пристрої датчик закріплюється на ошийнику та дозволяє власникам отримувати інформацію такі медичні дані тварини, як температура та пульс, частота дихання та серцевих скорочень, рівень активності тварин та інші. Зібрані дані передаються на сервер у мережі Інтернет та заносяться у профіль тварини, а власник може передивитися їх на мобільному пристрої та передати ветеринару, що, у разі перевищення важливих показників, дозволяє вчасно вживати заходів для забезпечення повноцінного та здорового життя тварини. Даний підхід відкриває нові можливості для власників тварин, дозволяючи їм більш ефективно стежити за станом своїх улюбленців, а тваринам не надавати дискомфорту.

Цей передовий ошийник використовує надійний акумулятор, забезпечуючи роботу протягом до 6 тижнів без необхідності в підзарядці у цей період. За допомогою використання новітніх технологій, він оснащений матеріалами, які відтіняють воду та пил, роблячи такий ошийник ідеальним для будь-яких погодних умов, а захист від ударів додає додатковий рівень безпеки. Керування ошийником проходить через кнопки чи індикатор. Трійка різних розмірів, включаючи варіант для тварин вагою понад 3,6 кілограма, гарантує оптимальний вибір та комфорт для різних розмірів чотирилапих. Цей ошийник дозволяє не лише контролювати тварину, але й забезпечувати їй зручність та комфорт.

Переваги PetPase:

- PetPase надає широкий спектр параметрів моніторингу, включаючи температуру, активність, пульс, що дозволяє докладно стежити за станом здоров'я та вчасно реагувати на зміни.

- Висока точність та надійність отриманих даних роблять PetPace відмінним інструментом, а можливість отримувати сповіщення в реальному часі про відхилення від норми дозволяє вчасно реагувати на будь-які проблеми тварини.
- Вбудовані функції для обміну даними та взаємодії з іншими власниками тварин створюють спільноту та дозволяють обмінюватися досвідом та порадами.
- Сучасний дизайн та використання високоякісних матеріалів роблять PetPace не лише функціональним та надійним для тварин, але і привабливим для власників тварин.

#### Недоліки PetPace:

- Одним з основних недоліків є висока вартість, що може викликати сумніви у покупців та зменшити доступність продукту. Також стягується щомісячна комісія за використання інтернет-порталу, що не є зручним для багатьох користувачів.
- PetPace обмежений у функціональності для тренування тварин, що не підходить власникам активних тварин. А також він потребує сумісні технологічні пристрої, що також не є доступним для деяких користувачів.
- Працездатність ошейника повністю залежить від стабільного зв'язку, що може бути проблематичним у деяких місцях з поганим покриттям.
- В разі технічних несправностей, аварій або втрати пристрою існує ризик втрати важливих даних, що може вплинути на повноту та коректність моніторингу здоров'я тварини. Інформація, що зберігається в системі, може стати об'єктом несанкціонованого доступу, який може призвести до розголошення особистих даних та порушення приватності.

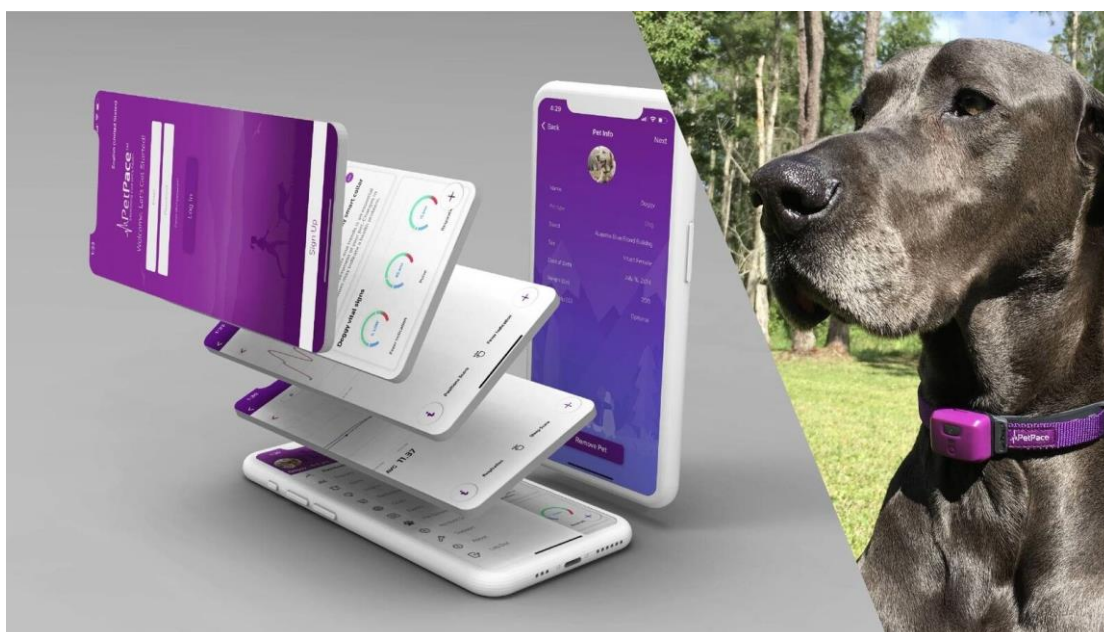


Рисунок 1.6 – Відстежувач PetPace Smart Collar

FitBark – це сучасний трекер для собак, створений для повного відстеження фізичної та психічної активності. Завдяки вбудованим датчикам та інтелектуальній аналітиці, FitBark дозволяє власникам отримати точне розуміння здоров'я та поведінки їх улюбленця.

За допомогою фітнес-трекера FitBark (рис. 1.7) є можливість глибше пізнати стан здоров'я та щастя чотирилапого друга, оскільки господар має повний контроль над станом здоров'я та може відслідковувати настрій та ефективність сну а на основі зібраних даних можна вчасно реагувати на зміни та приймати необхідні міри для покращення здоров'я тварини. Завдяки цьому відстежувачу, з'являється можливість ділитися цінною інформацією зі спеціалістами, щоб забезпечити максимальний рівень комфортного стану улюбленця.

FitBark завдяки потужній батареї забезпечує роботу трекера до 14 днів, що робить можливим не турбуватися про заряджання кожний день. Після вичерпання заряду потрібно підключити пристрій за допомогою кабелю micro-USB та дочекатися повного заряду.

Трекер працює завдяки додатку FitBark, а сам пристрій, який має вигляд кістки розміром 4см, треба прикріпити до нашійника собаки. Після цього вбудовані сенсорні датчики, що знаходяться на звороті трекера, автоматично передаватимуть важливі дані в додаток, де власник може переглянути усі важливі дані свого улюбленця.

FitBark відстежує якість сну, раціон, стан нервової системи та навіть контролює відновлення після хвороби. За допомогою додатка можливо аналізувати результати фізичних вправ, порівнювати їх з результатами інших собак, або встановлювати власні цілі для досягнення потрібного рівня активності.

#### Переваги FitBark:

- FitBark враховує не тільки кількість годин сну, а також і його якість, що дозволяє зрозуміти рівень відновлення та комфорту тварини під час відпочинку. Також трекер дозволяє не лише відслідковувати дані, але й ділитися ними з ветеринаром та іншими власниками тварин, що дозволяє отримувати корисні поради.
- FitBark оснащений міцними та водостійкими матеріалами та здатний витримувати навантаження під час активних ігор та прогулянок чи різних погодних умов. А багата палітра кольорів дозволяє підібрати пасуючий трекер згідно переваг тварини та порадувати око власника.
- FitBark надійно функціонує як у приміщенні, так і на вулиці, не втрачаючи зв'язок навіть під час активних ігор чи поганих погодних умов, що дозволяє не обмежуючи умови для прогулянки улюбленців
- Виробник регулярно випускає оновлення програмного забезпечення та додаткові функції, щоб забезпечити користувачам найсучасніші можливості.

#### Недоліки FitBark:

- FitBark не може замінити повноцінний медичний огляд та діагностику, незважаючи на багатий набір функцій. Також в деяких випадках використання ошейника може бути обмежене для тварин із певними

медичними протипоказаннями або особливою чутливістю до подібних пристроїв.

- Вартість самого трекера FitBark може бути виправданою, але важливо враховувати можливі витрати на додаткові послуги чи підписки, які можуть бути необхідні для повного функціоналу пристрою.
- Для деяких тварин може знадобитися час для пристосування до нового ошийника, особливо якщо вони не звикли до подібних пристроїв, а оскільки FitBark розташований на ошийнику тварини, існує певний ризик його втрати або пошкодження під час дуже активних дій тварини, що може завдати шкоди тварині та принести додаткові витрати.
- У деяких моделях FitBark відсутня можливість вимірювання пульсу та рівня кисню в крові, що може обмежити повноту моніторингу фізіологічного стану тварини.

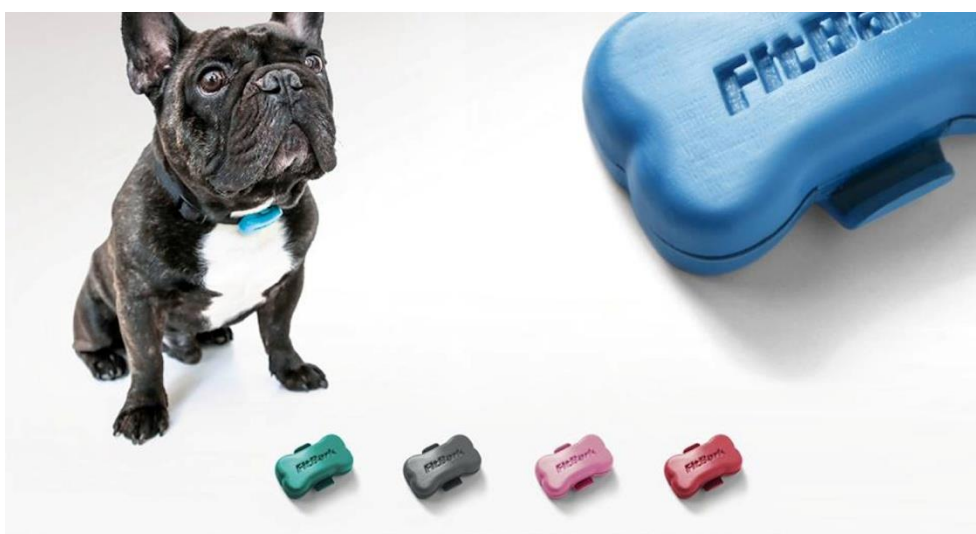


Рисунок 1.7 – Система моніторингу для собак FitBark

Petrics Smart Pet Bed – це розумне ліжко для домашніх улюбленців, яке не лише надає тваринам комфортне місце для відпочинку, але й стежить за їхнім здоров'ям. За допомогою вбудованих сенсорів та інтелектуальних технологій, це розумне ліжко відстежує фізичну активність та поведінку улюбленця.

Petrics Smart Pet Bed (розумна лежанка для домашніх тварин Petrics) відстежує вагу, час відпочинку та активність улюбленця, а також у своїх функціях передбачає контроль клімату, що допомагає тваринам досягнути оптимальних температурних діапазонів. Це розумне ліжко також допомагає власникам переконатися, що їхні улюбленці досягають комфортної ваги в залежності від породи, віку та інших характеристик. Лежанка має вбудовані ваги та є першою розробкою для домашніх тварин, яка пропонує нагрів і охолодження і залежності від оточуючої температури, що дозволяє користувачеві автоматично або вручну регулювати ідеальну температуру для кожної тварини залежно від породи та середовища.

Petrics Smart Pet Bed (рис. 1.8) керується за допомогою додатків Petrics Health або Nutrition App. Розумна лежанка має міцну та зручну конструкцію та доступна в трьох розмірах і має чохол, який можна прати та змінювати. Smart Pet Bed поєднується з трекером відстеження активності Petrics, який відстежує швидкість спалювання калорій домашніх тварин, рівень відпочинку та допомагає з іншими вимірами стану тварини. Водонепроникний трекер, який можна носити, легко синхронізується з будь-яким мобільним пристроєм та має невеликий розмір. Такий трекер має можливість використання без підзарядки протягом 4 місяців термін служби батареї, що спрощує його використання.

#### Переваги Petrics Smart Pet Bed:

- Ліжко оснащено високотехнологічними сенсорами, що вимірюють температуру тіла, серцевий ритм та інші показники, що дозволяє власникам більш детально вивчати стан здоров'я своїх тварин, використовуючи додатки.
- Лежанка має можливість взаємодіяти з ветеринарною системою, спрощуючи обмін інформацією та надаючи зовнішнім спеціалістам доступ до актуальних даних для більш ефективного контролю здоров'я тварини.



- Здатність ліжка взаємодіяти з іншими розумними пристроями в домі, такими як системи освітлення чи безпеки, створює зручне середовище для тварин та їх власників. Також є взаємодія з голосовими асистентами, такими як Amazon Alexa чи Google Assistant, що дозволяє проводити голосовий контроль.
- Лежанка адаптується до конкретних потреб та особливостей кожної тварини, забезпечуючи індивідуалізований підхід до догляду та комфорту, що є одним з важливіших критеріїв під час вибору господарями.

#### Недоліки Petrics Smart Pet Bed:

- З урахуванням фізіологічних особливостей окремих порід тварин, деякі розміри ліжка можуть бути непридатними для використання, а саме для порід з довгим тілом чи нестандартними параметрами необхідні спеціальні розміри для забезпечення комфортного використання ліжка.
- Для дуже активних тварин, які мають звичку стрибати чи активно грати в ліжку, може виникнути проблема утримання ваги на ліжку, що може вимагати додаткових заходів для забезпечення стабільності та безпеки використання.
- При передачі даних для аналізу чи консультації може виникнути питання щодо того, кому саме надається доступ до цієї інформації. Якщо дані про тварину зберігаються в хмарному сховищі або на серверах, існує ризик, що ці дані можуть бути підвергнуті кібератакам або несанкціонованому доступу.
- Деякі розширені функції, такі як індивідуальні поради щодо догляду та здоров'я тварини, можуть вимагати платної підписки або додаткових витрат, що може збільшити загальні витрати на володіння. У випадку пошкодження певних компонентів, вартість їх заміни або ремонту може також становити додаткові витрати для власників тварин.

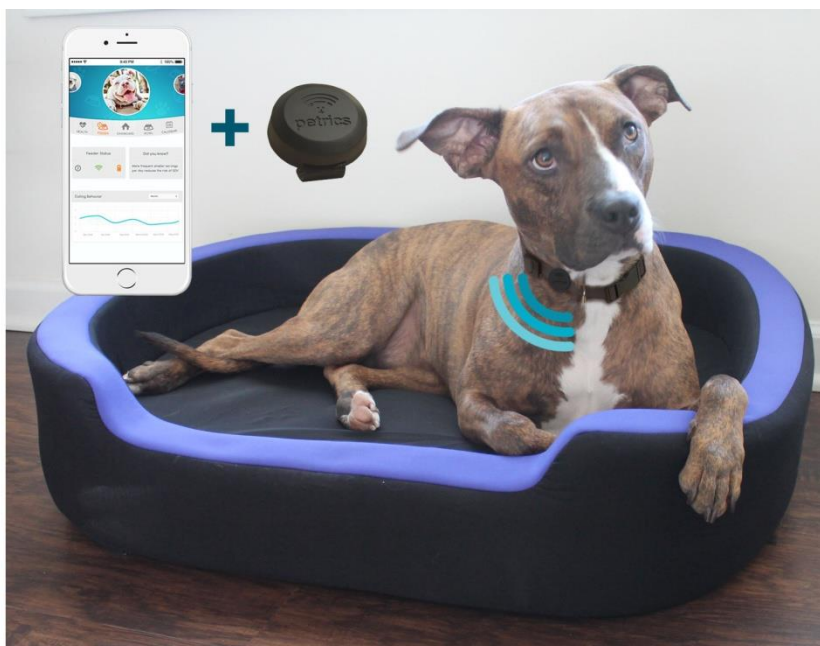


Рисунок 1.8 – Лежанка Petrics Smart Pet Bed з трекером

### 1.5 Постановка задачі

Перед тим, як розпочати розробку веб-сервісу, потрібно чітко визначити основні цілі та завдання, які мають бути розроблені для створення ефективного рішення конкретної задачі. Після уточнення цих питань, можна почати створення короткого опису поставленої задачі для визначення напрямку роботи.

Метою створення веб-сервісу готелю для тварин із системою контролю їх здоров'я є отримання інформації про умови для проживання тварин, надання їм необхідних послуг, контроль стану їх здоров'я за допомогою спеціальних пристроїв та надання відповідних звітів.

Використання веб-сервісу надасть користувачеві зручний доступ до даних про його тварину для комфортного і миттєвого перегляду стану здоров'я та послуг, які надаються [6].

При розробці веб-сервісу було визначено 2 категорії користувачів: користувач та адміністратор, тому система має базу даних, яка керується

адміністратором, вносить і зберігає дані про користувача та має інформацію, яка необхідна для функціонування веб-сервісу.

Веб-сервіс має пропонувати такі функціональні можливості:

- перегляд усієї необхідної інформації про послуги та проживання;
- перегляд контактних даних;
- можливість створення особистого кабінету;
- можливість бронювання послуг та проживання тварини;
- можливість переглядати особисті дані, звіти та бронювання.

На основі аналізу потреб в галузі готелей для тварин та догляду за ними були сформульовані вимоги до веб-сервісу "Fluffy Home". Розроблено спеціальний веб-сервіс для зручного бронювання та отримання послуг для домашніх улюбленців, який додатково забезпечує контроль за їхнім здоров'ям. Цей веб-сервіс спрямований на поліпшення умов для проживання та догляду за тваринами, забезпечуючи доступ до даних про їхнє здоров'я та стан. Розробка "Fluffy Home" має за мету зробити процес обслуговування та догляду за тваринами більш зручним та ефективним для клієнтів.

## 2 ВИЗНАЧЕННЯ ВИМОГ ДО ВЕБ-СЕРВІСУ

### 2.1 Технічне завдання

Технічне завдання (ТЗ) – це документ, який визначає вимоги до продукту, процесу чи системи. Воно включає в себе детальний опис того, що потрібно зробити по поставленим завданням, і які результати очікуються від виконання цих завдань. ТЗ є важливим інструментом у процесі розробки програмного проекту, оскільки воно допомагає всім учасникам мати чітке розуміння того, які перед ними стоять задачі та який результат очікується.

Створюваний проект – це веб-сервіс для готелю для тварин, який включає систему контролю їх здоров'я. Цей веб-сервіс допоможе власникам тварин легко бронювати номери та послуги в готелі, а персоналу готелю - ефективно керувати роботою готелю та контролювати здоров'я тварин за допомогою спеціальних пристроїв та ветеринарів. Технічне завдання для такого проекту допоможе мати чітке розуміння того, які функції повинен мати веб-сервіс, які технології використовувати для його розробки, а також які вимоги висуваються до зберігання даних у БД (база даних) [7]. Завдяки цьому можливо створити веб-сервіс, який буде ефективним, надійним та зручним для користувачів.

Після детального опису проекту, наступним кроком є визначення цілей та завдань веб-сервісу. Цілі допоможуть отримати розуміння того, до чого ми прагнемо, а завдання дадуть чітке розуміння того, як ми можемо досягти цих цілей. Надалі розглянемо цілі та завдання створюваного проекту детальніше.

Цілі проекту:

- Створити зручний веб-сервіс для бронювання номерів та послуг в готелі для тварин.
- Впровадити систему контролю здоров'я тварин, яка дозволить персоналу готелю відстежувати стан здоров'я кожної тварини.
- Забезпечити безпечне використання даних користувачів.

Завдання проекту:

- Створити інтуїтивно зрозумілу систему бронювання, яка дозволить власникам тварин без зусиль бронювати номери та послуги.
- Розробити систему управління, яка дозволить персоналу готелю ефективно координувати всі бронювання та відстежувати їх статус.
- Впровадити систему моніторингу здоров'я, яка включатиме в себе використання спеціалізованого обладнання та ветеринарів для контролю стану здоров'я тварин.

Наступним кроком є опис вимог до користувачів, Важливо зазначити, що вони є ключовим елементом при розробці веб-сервісу. Вони визначають, як користувачі будуть взаємодіяти з сервісом, і які функції вони зможуть використовувати.

Основні вимоги, які було встановлено для власників тварин (користувачів веб-сервісу):

- Власники тварин повинні мати можливість реєструватися та мати особистий кабінет.
- Входити в систему та робити бронювання
- Виходити з особистого кабінету та переглядати поточну інформацію про готель.
- Переглядати звіт про стан здоров'я своїх тварин.
- Отримувати необхідну інформацію за контактними даними.

Адміністратори системи також відіграють важливу роль у функціонуванні веб-сервісу, тому що вони відповідають за керування обліковими записами користувачів у базі даних та за технічні аспекти веб-сервісу.

Адміністратори повинні виконувати такі функції:

- Адміністратори повинні мати можливість переглядати, видаляти та керувати бронюваннями.
- Вони повинні мати можливість зв'язуватися з користувачами за контактними даними, наданими користувачами під час реєстрації.

- Вони повинні мати можливість створювати звіти про стан здоров'я тварин, які знаходяться в готелі.
- Повинні мати можливість налаштовувати параметри системи, включаючи ціни та доступні послуги.
- Мати можливість аналізувати використання сервісу для його подальшого покращення.

Для правильного функціонування веб-сервісу, важливо враховувати технічні вимоги [8].

Для веб-сервісу готелю для тварин встановлені такі технічні вимоги:

- База даних: База даних може бути розроблена за допомогою MySQL (Structured Query Language)— потужної системи управління базами даних, що забезпечить ефективне зберігання та обробку інформації.
- Мова програмування: Для розробки логіки веб-сервісу використовується мова програмування PHP, що є популярним інструментом для створення веб-сайтів.
- Сумісність з веб-браузерами: Веб-сервіс повинен бути сумісний з усіма сучасними веб-браузерами, такими як Chrome, Firefox, Safari, Edge, забезпечуючи доступність та комфорт користувачів.
- Підтримка відкритих серверів: Веб-сервіс повинен бути сумісний із відкритими серверами, такими як Open Server, для спрощення розгортання та локального тестування сервісу під час розробки.

Для зручного та зрозумілого використання веб-сервісу, потрібно встановити також вимоги і до дизайну.

Дизайн сайту повинен мати наступні характеристики:

- Бути простим та інтуїтивно зрозумілим, щоб користувачі могли легко навігуватися та використовувати веб-сервіс.
- Інтерфейс користувача повинен бути зручним для використання як на настільних комп'ютерах, так і на мобільних пристроях, щоб користувачі могли використовувати веб-сервіс на різних пристроях.

- Зрозуміла логіка та простий інтерфейс для реєстрації, входу, здійснення бронювання, перегляду стану здоров'я своїх тварин та даних бронювання.

## 2.2 Планування робіт

Планування робіт – це критично важливий етап у процесі розробки будь-якого веб-сервісу [9]. Воно включає в себе визначення основних етапів розробки та встановлення рамок для кожного з них, а також розподіл ресурсів на кожен етап. У контексті створюваного проекту – веб-сервісу готелю для тварин із системою контролю їх здоров'я, планування робіт включатиме розробку прототипу та оптимізацію веб-сервісу.

Важливо пам'ятати, що планування вимагає гнучкості, оскільки можуть виникнути нові вимоги або проблеми, які потребують змін у плані. Крім того, планування також включає в себе та оцінку прогресу, щоб впевнитися в тому, що проект відповідає встановленим цілям.

Останній аспект планування робіт – це розробка детального плану для кожного етапу розробки веб-сервісу готелю для тварин із системою контролю їх здоров'я. Це включає в себе визначення конкретних завдань, їх пріоритетів, а також оцінку необхідних ресурсів для кожного етапу. Завдяки цьому процесу, можливо краще розуміти обсяг роботи, який необхідний для успішного виконання та завершення проекту.

Розробка прототипу – це перший крок у процесі створення веб-сервісу готелю для тварин [10]. Прототип служить ключовим інструментом для визначення основних функцій та інтерфейсу веб-сервісу. Він дозволяє розробнику веб-сервісу чітко представити структуру та функціональність майбутнього проекту, що сприяє ефективному плануванню та розробці завдань.

Основний елемент прототипу – це інтерфейс користувача. Інтерфейс користувача відіграє важливу роль у взаємодії користувача з веб-сервісом та розуміння його основного призначення. Він включає в себе дизайн та розташування елементів на сторінках веб-сервісу, що містить в собі елементи,

такі як меню навігації, форми для введення даних, кнопки та полоси прокрутки. Меню навігації допомагає користувачам легко перемикатися між різними частинами веб-сервісу для ознайомлення із потрібною інформацією. Кнопки та інші елементи керування дозволяють користувачам взаємодіяти з веб-сервісом, виконуючи різні дії, такі як бронювання та перегляд інформації. Розробка ефективного інтерфейсу користувача вимагає розуміння потреб користувачів, а також врахування розробки зручного дизайну інтерфейсу. Це включає в себе створення інтуїтивно зрозумілого дизайну, який допомагає користувачам легко зрозуміти, як використовувати веб-сервіс, а також забезпечити довіру користувача до використовуваного сервісу. Прототипи інтерфейсу користувача наведені на рисунку 2.1.

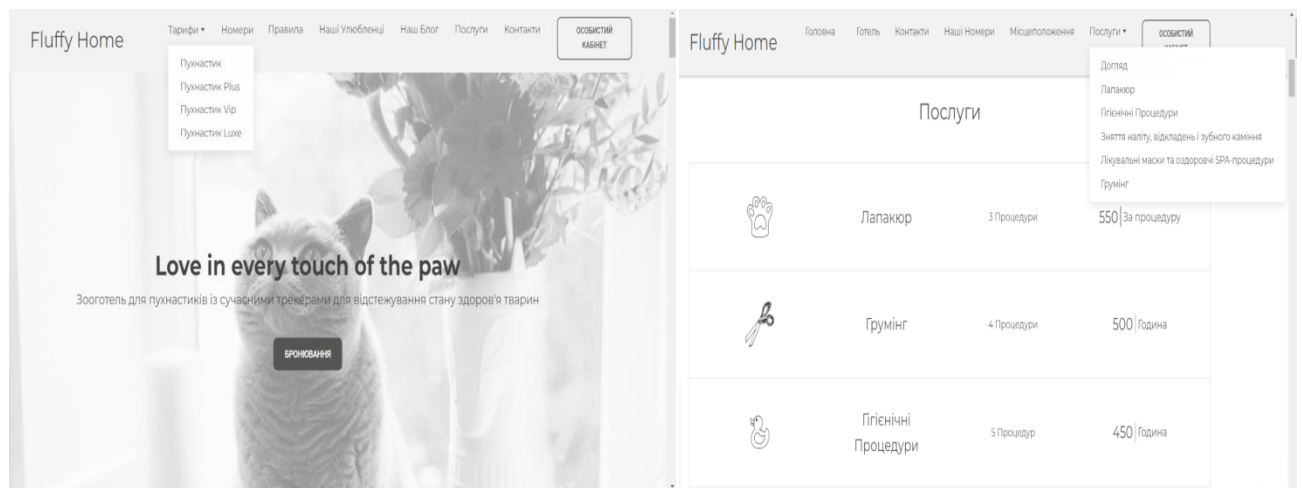


Рис. 2.1 – Прототипи інтерфейсу користувача

Наступними ключовими параметрами є форма реєстрації та авторизації, які є основою взаємодії користувача із веб-сервісом готелю для тварин. В контексті розробки даного проекту, форма авторизації та реєстрації стає невід'ємною частиною користувацького інтерфейсу, забезпечуючи користувачеві подальші можливості бронювання послуг для тварини. Ці дві форми визначають способи ідентифікації користувачів і створення їх облікових записів, створюючи доступ до особистого кабінету. Форма реєстрації – це перший крок взаємодії нового



користувача з сервісом, яка пропонує поля для введення основних особистих даних для їх подальшого використання. Ключовою метою є забезпечення простого створення облікового запису, а також можливості надання додаткової інформації, яка може бути використана у подальшому для ознайомлення користувача. Форма авторизації вже стає точкою доступу для вже зареєстрованих користувачів. Вона має бути також простою та зручною, дозволяючи введення раніше збережених даних для швидкого доступу до особистої інформації. Прототипи форми реєстрації та авторизації користувача наведені на рисунку 2.2.

Рисунок 2.2 показує два прототипи форм: «Реєстрація» та «Авторизація». Форма «Реєстрація» має два текстових поля: «Логін» та «Пароль», та кнопку «Зареєструватися». Форма «Авторизація» має два текстових поля: «Логін» та «Пароль», та кнопку «Увійти» та кнопку «Зареєструватися».

Рисунок 2.2 – Прототипи форми реєстрації та авторизації користувача

Подальшу ключову роль відіграє форма бронювання, яка допомагає оптимізувати процес резервування. Форма бронювання визначає зручний та простий спосіб бронювання місць проживання та послуг для тварин, спрощуючи взаємодію користувача із веб-сервісом. Головною метою є скоротити час користувача для отримання доступних послуг шляхом вибору необхідних послуг через вже готову форму. Після завершення бронювання, користувач отримує підтвердження та інформацію про своє бронювання у особистому кабінеті. Прототип форми бронювання для користувача наведено на рис. 2.3.

ПНБ  
 \_\_\_\_\_  
 Пошта  
 \_\_\_\_\_  
 Телефон  
 \_\_\_\_\_  
 Ім'я Тварини  
 \_\_\_\_\_  
 Вік  
 \_\_\_\_\_  
 Стан Здоров'я  
 \_\_\_\_\_  
 Кіт \_\_\_\_\_ Собака \_\_\_\_\_  
 Дата  
 дд. мм. гггг   
 Оберіть Послугу   
 Оберіть Тариф   
 Оформити Бронювання

Рисунок 2.3 – Прототип форми бронювання для користувача

Останнім важливим пунктом для власника тварини є особистий кабінет користувача, який є головним елементом простору, де клієнти можуть зручно взаємодіяти зі своїм обліковим записом та переглядати особисті дані. Також користувачі можуть переглядати звіти про стан здоров'я тварин та перевіряти історію бронювань. Цей функціонал дозволяє користувачам зменшити час на пошук інформації бронювання та спрощує доступ до необхідних даних. Прототип особистого кабінету користувача наведено на рисунку 2.4.

Особистий Кабінет

Контактні дані Дані про тварину
 Дані про послуги Бронювання
 Дата на бронювання

Рисунок 2.4 – Прототип особистого кабінету користувача

Після створення прототипу веб-сервісу наступним важливим кроком є його оптимізація [11]. Оптимізація – це процес вдосконалення веб-сервісу з метою покращення його продуктивності, доступності, безпеки та використання. Це може включати в себе все, починаючи від оптимізації коду та бази даних до покращення дизайну інтерфейсу та наповнення контентом, який відповідає потребам користувачів. Оптимізація може допомогти збільшити швидкість завантаження сторінок, покращити пошук сервісу та досвід використання користувачем.

Оптимізація веб-сервісу готелю для тварин має наступні аспекти:

- Оптимізація продуктивності: Використання кешування для зберігання часто використовуваної інформації є важливим елементом оптимізації продуктивності, що дозволяє зберігати результати важких операцій для уникнення їх повторного виконання. Такий підхід дозволяє значно зменшити час відгуку сервера. Оптимізація коду також відіграє важливу роль. Це може включати в себе перероблення коду для покращення його ефективності, видалення непотрібного коду, а також використання більш ефективних структур даних.
- Оптимізація для пошукових систем (SEO - Search Engine Optimization): SEO включає в себе ряд стратегій, спрямованих на покращення видимості веб-сайту в результатах пошуку. Використання ключових слів в заголовках та контенті може допомогти покращити більш успішне розміщення сайту. Створення контенту, який відповідає потребам користувачів, може збільшити кількість відвідувачів веб-сервісу.
- Оптимізація використання ресурсів: Ефективне використання ресурсів включає в себе використання ефективних алгоритмів для обробки даних, оптимізацію запитів до бази даних для зменшення навантаження на сервер, а також управління сесіями користувачів для зменшення використання пам'яті.
- Оптимізація користувацького досвіду (UX – User Experience): Спрощення та поліпшення функції навігації по сайту та оптимізація форм бронювання можуть зробити процес бронювання простішим та швидшим для користувачів. Це може включати в себе впровадження інтуїтивно зрозумілої

навігації, поліпшення та прискорення відгуку на дії користувача, а також впровадження зручних для користувача функцій, таких як автозаповнення форм авторизації.

- Оптимізація мобільного досвіду: Важливо забезпечити оптимізацію і можливість опрацювання і на мобільних пристроях. Це може включати в себе використання адаптивного дизайну, оптимізацію швидкості завантаження на мобільних пристроях, а також впровадження функцій, які властиві для мобільних пристроїв.
- Оптимізація локалізації: Веб-сервіс потрібно оптимізувати для різних мов для більш зручного використання. Це може включати в себе не лише переклад контенту на різні мови, а також забезпечення підтримки різних форматів дати та часу.
- Оптимізація аналітики: Аналітика допомагає зрозуміти, як користувачі використовують веб-сервіс, що дозволяє покращувати його у подальшому. Це може включати в себе відстеження взаємодії користувачів та час проведений на веб-сервісі.
- Оптимізація соціальних медіа: Для привернення відвідувачів на веб-сервіс, потрібно оптимізувати його для соціальних медіа, що може включати в себе додавання кнопок з посиланнями для поширення в соціальних медіа користувачів, та використання соціальних медіа для просування контенту веб-сервісу.
- Оптимізація взаємодії з користувачем (UI – User Interface): Це включає в себе створення привабливого та зручного для користувача інтерфейсу, що надає можливість використовувати кольори та шрифти, які відповідають тематиці веб-сервісу зооготелю, а також впровадити зручні для користувача елементи керування, такі як кнопки, форми та меню.
- Оптимізація відгуку на дії користувача: Це включає в себе відповідь на дії користувача в реальному часі. Така оптимізація надає можливість надавати відгук користувачу під час заповнення форми чи надавати відповідь на запити користувача швидше.

- Оптимізація використання мультимедіа: Включає в себе використання зображень та відео для покращення взаємодії користувача з веб-сервісом. Оптимізація розміру та формату мультимедійних файлів дозволяє їм завантажуватися швидше та відображатися правильно на всіх пристроях.
- Оптимізація тестування та відлагодження: Створення ефективних процедур тестування та відлагодження для веб-сервісу допомагає регулярно проводити виявляти та виправляти помилки, а використання інструментів відлагодження є зручним способом для виявлення проблем з продуктивністю.

Планування робіт є критично важливим етапом у створенні веб-сервісу готелю для тварин із системою контролю їх здоров'я. Під час аналізу цього етапу, було розглянуто два ключові аспекти: розробка прототипу та оптимізація.

На етапі розробки прототипу було враховано основні етапи створення інтерфейсу користувача. Прототип дозволяє спроектувати в майбутній веб-сервіс, визначити його функціонал та виявити недоліки на етапі розробки. Цей підхід сприяє вдосконаленню функціональності та покращенню взаємодії користувача із веб-сервісом.

На етапі дослідження оптимізації веб-сервісу були виявлені ключові аспекти, які покращують його ефективність. Використання кешування, оптимізація коду, SEO, ефективне використання ресурсів та оптимізація користувацького досвіду, які сприяють створенню якісного та ефективного веб-сервісу.

Враховуючи ці аспекти, можна зазначити, що планування робіт дозволило створити шлях для розробки веб-сервісу зооготелю для тварин, враховуючи потреби користувачів та забезпечуючи його оптимальну продуктивність та функціональність.

## **3 ПРОЕКТУВАННЯ СИСТЕМИ**

### **3.1 Проектування інформаційної системи веб-сервісу за допомогою діаграм опису бізнес-процесів IDEF0**

Для ефективного проектування інформаційної системи веб-сервісу зооготелю для тварин із системою контролю їх здоров'я, використовується методологія IDEF0 (Function Modeling) [12].

IDEF0 являє собою методологію, яка спеціалізується на функціональному моделюванні та графічному представленні процесів. Така методологія підходить для формалізації та опису бізнес-процесів. Однією з ключових особливостей IDEF0 є її зосередженість на ієрархічному представленні об'єктів. Такий підход спрощує розуміння предметної області та відрізняє його від WorkFlow, адже IDEF0 зосереджується на логічних зв'язках між роботами, а не на послідовності їх виконання в часі.

В рамках проектування веб-сервісу зооготелю було створено таблицю інтерфейсних дуг, яка відіграє важливу роль у визначенні ключових аспектів взаємодії веб-сервісу з його оточенням.

Таблиця включає наступні елементи:

- Входи – різні види даних, які отримує веб-сервіс. Такі види даних важливі виконання функцій веб-сервісу.
- Виходи – це результати, які створює веб-сервіс. Вони показують вплив веб-сервісу на оточення.
- Технічні вимоги та специфікації – вимоги та процедури, які веб-сервіс повинен дотримуватися для забезпечення якості та надійності веб-сервісу.
- Відповідальні учасники процесу – особи чи системи, які відповідають за виконання процесу. Вони відіграють роль у розумінні відповідальності за різні аспекти веб-сервісу.

Елементи допомагають краще зрозуміти, як веб-сервіс взаємодіє з різними аспектами системи. Детальні дані інтерфейсних дуг представлені в таблиці 3.1.

Таблиця 3.1 – Визначення інтерфейсних дуг контекстної діаграми

№	Інтерфейсні дуги	Зміст
1	Входи	Послуги, що надаються веб-сервісом. Оформлення бронювання послуг на веб-сервісі. Інформація про послуги. Дані про бронювання замовлення. Бронювання користувачів.
2	Виходи	Відомості про кількість бронювань користувачів; Прайс-лист з цінами на всі послуги веб-сервісу зооготелю.
3	Технічні вимоги та специфікація	Положення про веб-сервіс зооготелю та взаємодію з клієнтами. Вимоги до збору, обробки та представлення даних. Процедури тестування та валідації.
4	Відповідальні учасники процесу	Користувач. Адміністратор. Веб-сервіс готелю для тварин із системою контролю їх здоров'я.

Після визначення інтерфейсних дуг, створюється контекстна діаграма бізнес-процесу. Контекстна діаграма допомагає краще зрозуміти бізнес-процес, визначити межі та виявити ключові взаємодії. Діаграма також може виявити можливі ризики або проблеми, що можуть виникнути в процесі, а також слугувати основою для більш детального моделювання бізнес-процесу. Контекстну діаграму бізнес-процесу наведено на рисунку 3.1.

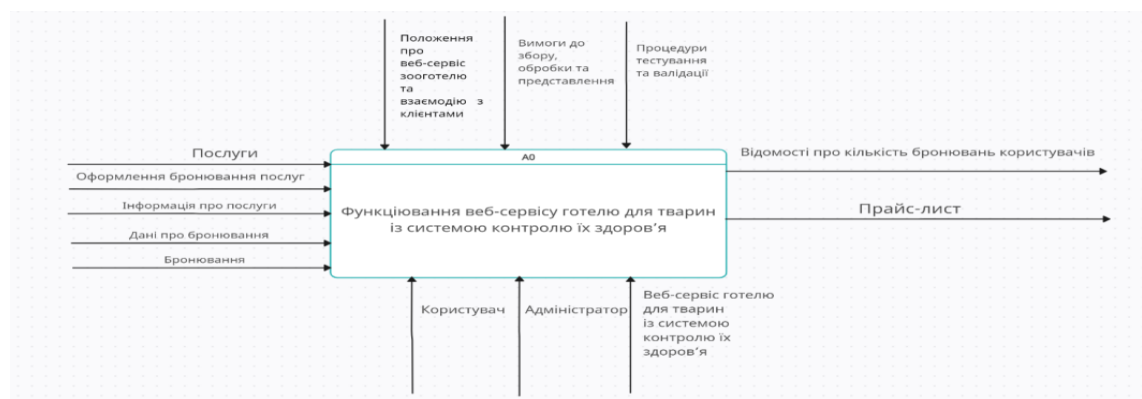


Рисунок 3.1 – Контекстна діаграма бізнес-процесу

Після створення контекстної діаграми, проведено декомпозицію першого рівня. Зміст всіх підпроцесів декомпозиції першого рівня детально описано в таблиці. Декомпозиція першого рівня – процес розбиття вихідної системи на підсистеми, які виконують поставлені задачі. Декомпозиція першого рівня дозволяє спростити аналіз і проектування системи, а також забезпечити її гнучкість. Зміст кожного підпроцесу описує, які дії виконуються в рамках підпроцесу і які дані використовуються чи створюються. Декомпозиція першого рівня представлена у таблиці 3.2.

Таблиця 3.2 – Підпроцеси декомпозиції першого рівня

№	Підпроцес	Зміст
1	Формування прайс-листа послуг	Формування прайс-листа з урахуванням можливостей користувачів. Контроль списку послуг. Формування прайс-листа з урахуванням замовлень.
2	Оформлення заявки на бронювання послуг	Додавання бронювання до особистого кабінету. Оформлення бронювання користувачем. Відправлення заявки на бронювання (підтвердження заявки).
3	Формування звітності з оформлень бронювання	Вибір списку бронювання для формування звітів бронювання. Обробка інформації по заявкам на бронювання. Підрахунок кількості заявок на бронювання. Формування звітності по заявкам.

Наступним кроком буде представлення декомпозицію першого рівня у вигляді діаграми, яка містить наступні підпроцеси:

- Формування прайс-листа послуг.
- Оформлення заявки на бронювання послуг.
- Формування звітності з оформлень бронювання.

Діаграма декомпозиції першого рівня представлена на рисунку 3.2.



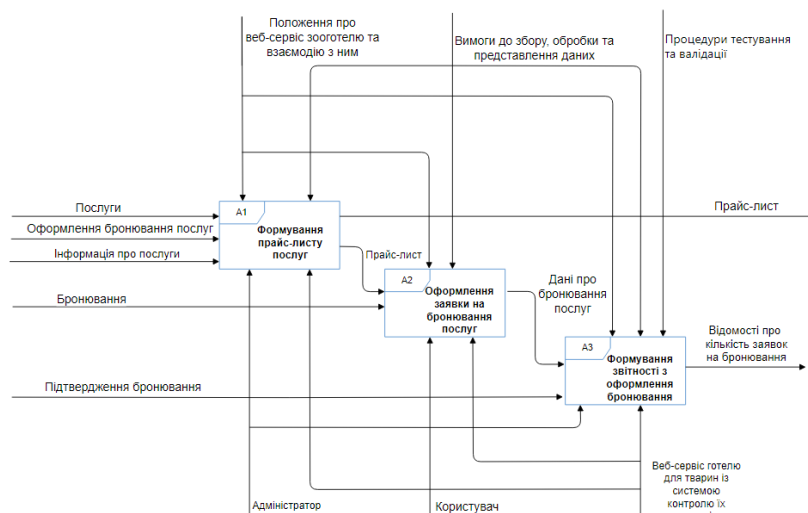


Рисунок 3.2 – Діаграма декомпозиції першого рівня

Формування прайс-листа послуг полягає в тому, що готель визначає, які послуги він надає своїм гостям, скільки вони коштують, які умови діють для користувачів. Після цього зооготель створює документ, в якому перелічує всі свої послуги і ціни, і розміщує його на своєму веб-сайті, регулярно оновлюючи свій прайс-лист, враховуючи зміни в попиті. Декомпозицію підпроцесу «Формування прайс-листа послуг» наведено на рисунку 3.3.

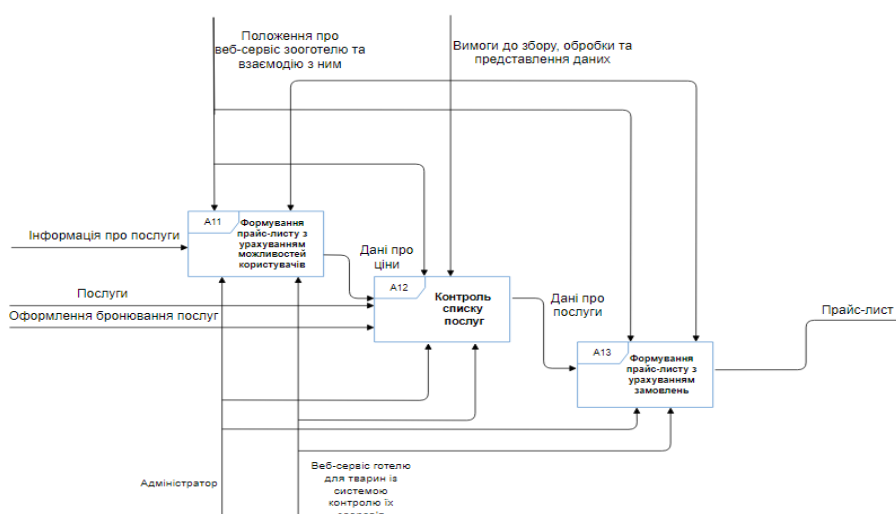


Рисунок 3.3 – Декомпозиція підпроцесу «Формування прайс-листа послуг»

Оформлення заявки на бронювання послуг полягає в тому, що користувач вибирає ті послуги, які йому потрібні, і заповнює спеціальну форму на веб-сервісі готелю або зв'язується з готелем для тварин за контактними даними. Користувач вказує свої дані та дані для бронювання, а готель і підтверджує заявку клієнта. Декомпозицію підпроцесу «Оформлення заявки на бронювання послуг» наведено на рисунку 3.4.

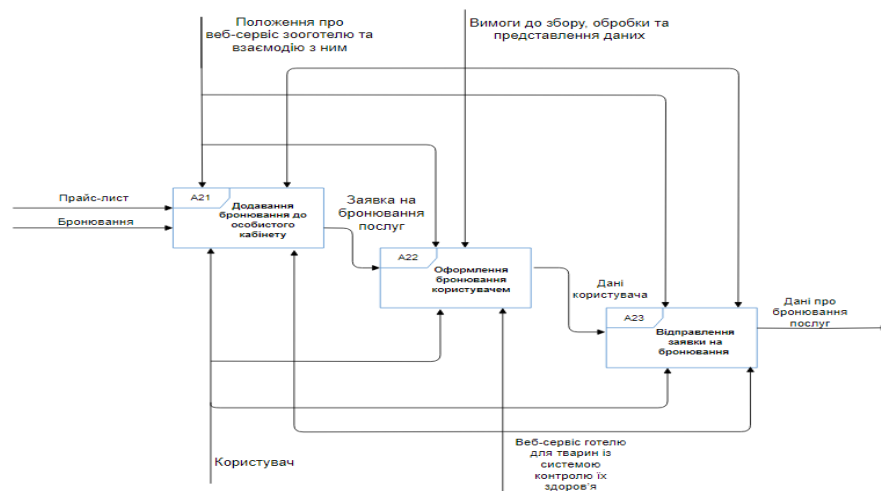


Рисунок 3.4 – Декомпозиція підпроцесу «Оформлення заявки на бронювання послуг»

Формування звітності з оформлень бронювання полягає в тому, що веб-сервіс готелю для тварин збирає і обробляє всю інформацію про заявки на бронювання послуг, які надійшли від користувачів та використовує цю інформацію для аналізу своєї діяльності, планування ресурсів та контролю якості, Веб-сервіс створює різні види звітів про кількість і структуру заявок, про доходи і витрати, про рівень задоволеності і лояльності клієнтів та про ефективність. Декомпозицію підпроцесу «Формування звітності з оформлень бронювання» наведено на рисунку 3.5.

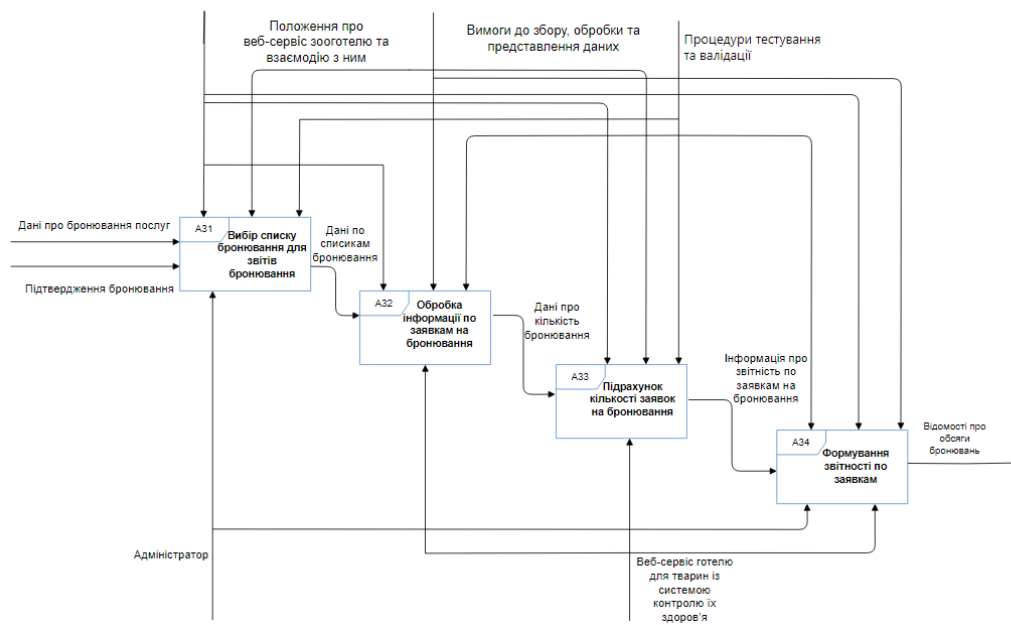


Рисунок 3.5 – Декомпозиція підпроцесу «Формування звітності з оформлень бронювання»

Таким чином було проведено функціональний аналіз і моделювання бізнес-процесу для веб-сервісу готелю для тварин із контролем стану їх здоров'я за допомогою використання стандарту IDEF0, який дозволив описати функції, їх взаємозв'язки і потоки даних в системі. Було побудовано трирівневу модель бізнес-процесу, яка демонструє деталізацію процесів веб-сервісу.

### 3.2 Розроблення діаграми варіанту використання

Діаграми – це спосіб графічного представлення інформації про системи, процеси, дані чи відносини. Вони допомагають спрощувати, структурувати та аналізувати складну інформацію в зрозумілій формі. Вони також можуть служити основою для розробки, реалізації, тестування та управління системами.

Існує багато видів діаграм, які відрізняються за своїм призначенням, змістом, формою і стилем. Одним з найпоширеніших стандартів для моделювання систем є UML (Unified Modeling Language), який пропонує різні типи діаграм для

опису різних аспектів системи [13]. Для створюваного веб-сервісу буде розглянуто основні типи діаграм UML, які використовуються для моделювання функціональності, поведінки і взаємодії системи: діаграми варіанту використання, діаграми станів, діаграми діяльності і діаграми послідовності.

Діаграма варіантів використання – це тип діаграми, який демонструє, як різні користувачі або сутності (актори) можуть використовувати систему для досягнення поставлених задач [14]. Вона не вказує, як система реалізує ці функції, а тільки що вона надає та відображає функціональні вимоги до системи з погляду акторів. На графічній діаграмі варіанти використання позначаються еліпсами, а актори - фігурами людей. Між акторами та варіантами використання можуть бути показані різні види відношень, такі як асоціація, включення, розширення та узагальнення.

На веб-сервісі готелю для тварин взаємодіють такі актори, як новий користувач, авторизований користувач та адміністратор, а також присутня база даних. Новий користувач є актором, який ще не пройшов процес реєстрації та має можливість лише переглядати інформацію про веб-сервіс. Авторизований користувач після процесу авторизації має змогу оформлювати бронювання та переглядати деталі бронювання у особистому кабінеті. Адміністратор має можливість взаємодіяти з усіма об'єктами у базі даних та безпосередньо керувати бронюваннями користувача.

Варіанти використання включають зв'язки “include”, які вказують на взаємозв'язок між різними варіантами використання. Наприклад, варіант використання “Авторизація” може включати в себе варіант використання “Перегляд та керування особистими даними”, оскільки користувач повинен бути авторизований, перш ніж він зможе переглядати або керувати своїми особистими даними. Також варіант використання “Формування бронювання” може включати в себе варіант використання “Реєстрація”, оскільки новий користувач повинен спочатку зареєструватися, перш ніж він зможе зробити бронювання.

На основі визначених варіантів використання та акторів системи була створена діаграма варіантів використання, яка представлена на рисунку 3.6.

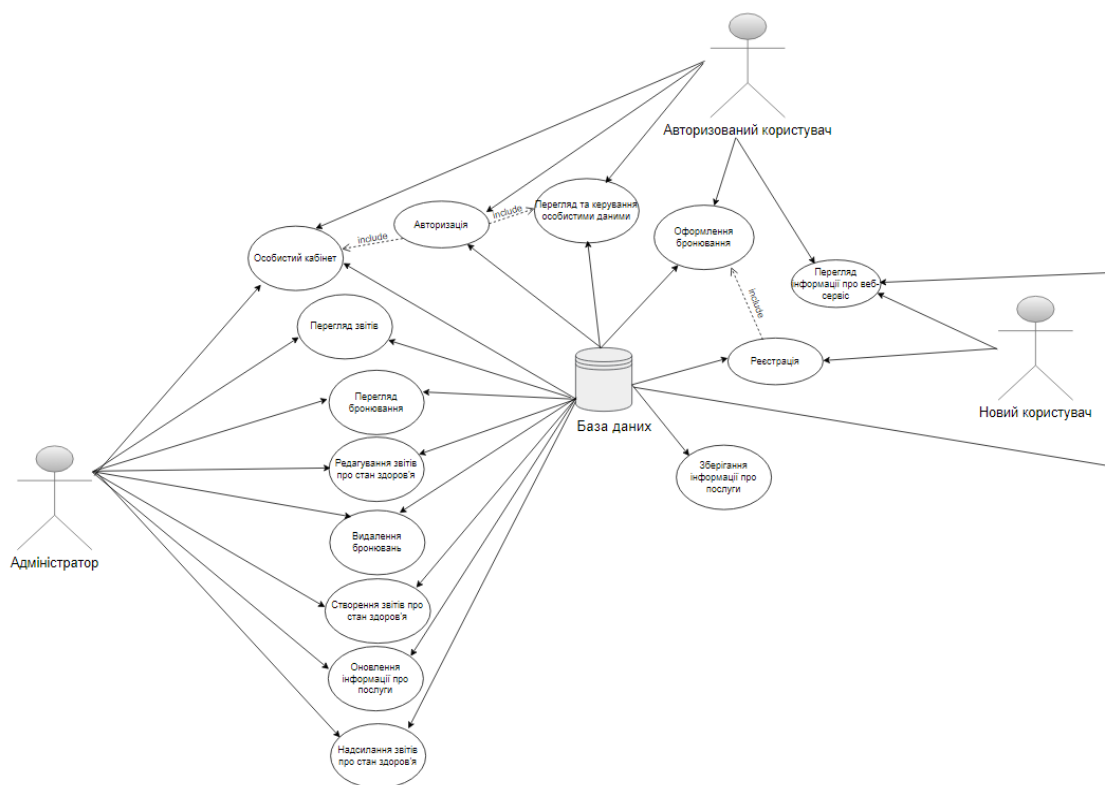


Рисунок 3.6 – Діаграма варіантів використання веб-сервісу

Для відображення поведінки окремих об'єктів використовуються діаграми станів, також можуть бути використані для опису функціональності різних компонентів моделей, таких як сценарії використання, актори, підсистеми, операції та методи.

Діаграма станів (statechart diagram) представляє собою спеціальний тип графа, який відображає певний автомат. Вузли графа представляють можливі стани автомата, які зображуються відповідними графічними символами, а ребра відображають переходи автомата з одного стану в інший [15]. Такі діаграми допомагають краще розуміти, як система відповідає на різні події та зміни стану. На рисунку 3.7 представлена діаграма переходів станів, яка показує стани, у яких знаходиться додаток у ході роботи.

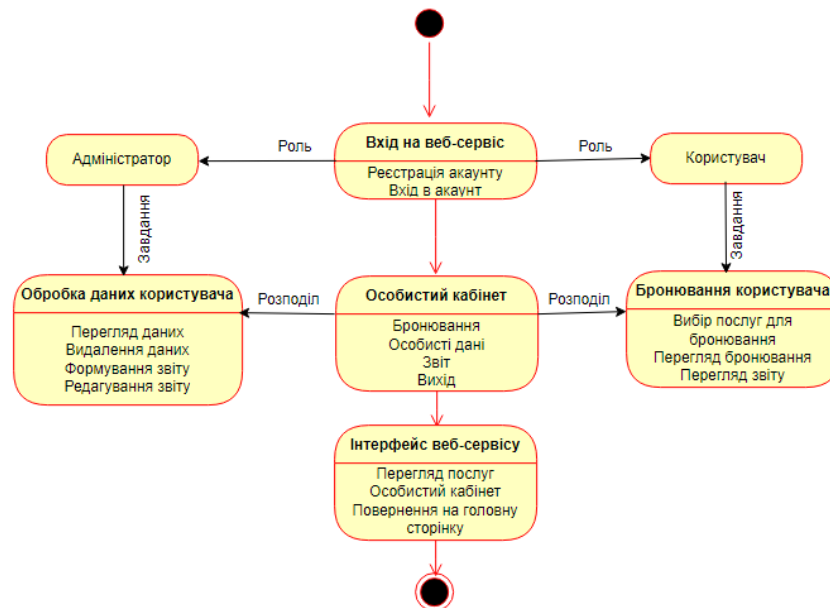


Рисунок 3.7 - Діаграма переходів стану веб-сервісу

Для того, щоби проілюструвати процеси, які відбуваються в системі, використовується діаграма діяльності (Activity Diagram). Вона відображає послідовність дій та переходи між ними, демонструючи, як система реагує на різні події та зміни стану. Цей тип діаграми включає можливість відображення паралельного виконання дій та синхронізації [16].

Діаграма діяльності відображає дії, які можуть виконуватися різними учасниками, і показує, як потік керування переходить від однієї операції до іншої. Використання діаграм діяльності допомагає краще розуміти, як система відповідає на різні події та зміни стану, що є важливим для проектування та розробки систем. Для системи була створена діаграма діяльності, яка відображає послідовність дій користувача при оформленні бронювання (рис. 3.8).

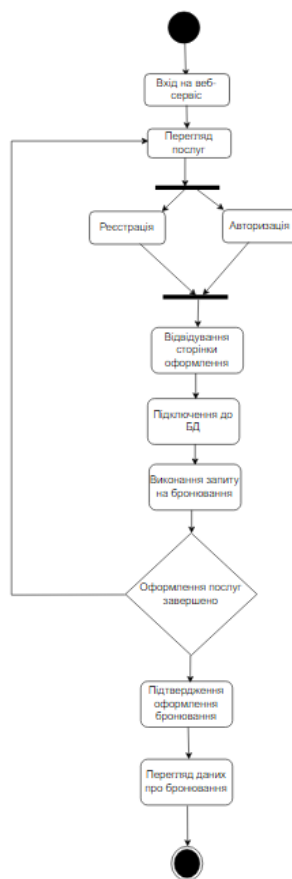


Рисунок 3.8 – Діаграма діяльності оформлення бронювання

Для відображення взаємодії між об'єктами в системі протягом певного періоду часу була створена діаграма послідовності, яка є важливим інструментом візуалізації [17]. Ця діаграма допомагає краще розуміти, як об'єкти в системі взаємодіють один з одним, і як повідомлення передаються між ними, а також відслідковувати життєвий цикл об'єкта від початку до кінця. На діаграмі послідовності об'єкти представлені у вигляді вертикальних ліній, а повідомлення між ними - у вигляді горизонтальних стрілок. Порядок повідомлень визначається зліва направо, а час - зверху вниз.

Для веб-сервісу готелю для тварин, діаграма послідовності відображає взаємодію між користувачем, системою бронювання та базою даних готелю. Користувач відправляє запит на бронювання, система бронювання потім обробляє цей запит, а база даних готелю зберігає інформацію про бронювання (рис. 3.9).

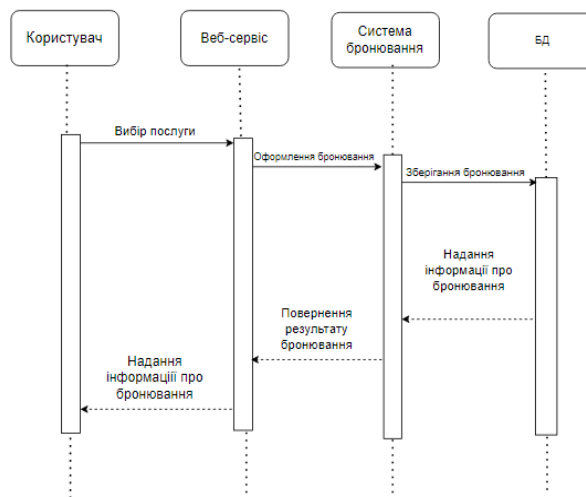


Рисунок 3.9 – Діаграма послідовності

### 3.3 Проектування бази даних веб-сервісу готелю

Основою веб-сервісу готелю для тварин із контролем їх здоров'я є база даних (БД), яка зберігає всю необхідну інформацію. База даних розроблена з використанням SQL та MySQL, які забезпечують надійне та ефективне зберігання і обробку даних. SQL (Structured Query Language) – це стандартизована мова, яка дозволяє визначати структуру, курувати та запитувати дані в БД. MySQL – це відкрита система управління реляційними базами даних (СУБД), яка використовує SQL як основний спосіб зв'язку з базою даних. СУБД включає в себе різні таблиці та зв'язки між ними, що дозволяють веб-сервісу надавати користувачам всю необхідну інформацію та послуги [18].

Для керування базою даних SQL та СУБД MySQL, використовується phpMyAdmin - вільний інструмент, написаний на PHP, який дозволяє адмініструвати MySQL через веб-інтерфейс. За допомогою phpMyAdmin можна легко створювати, модифікувати та видаляти бази даних, таблиці, поля або рядки. Також можна виконувати SQL-запити, управляти користувачами та правами доступу, експортувати та імпортувати дані в різні формати.



Використання цих інструментів дозволяє створити надійну та ефективну базу даних для веб-сервісу готелю для тварин. Ця база даних відповідає всім поставленим вимогам, забезпечуючи правильне функціонування веб-сервісу.

Концептуальне проектування бази даних є наступним кроком у процесі створення бази даних. Воно полягає у визначенні основних понять та відношень, які характеризують аналізовану область. Концептуальне проектування має на меті створити абстрактне уявлення про базу даних, що включає визначення типів сутностей, зв'язків між ними та атрибутів. Для розробки цієї моделі даних, потрібно виконати чотири кроки: визначити сутності, які представляють об'єкти або поняття; визначити взаємозв'язки, які показують, як сутності пов'язані між собою; визначити атрибути, які характеризують властивості сутностей; призначити первинні та альтернативні ключі, які ідентифікують сутності та дозволяють здійснювати пошук та зв'язування даних.

Для побудови моделі бази даних для веб-сервісу готелю для тварин із системою контролю їх здоров'я, треба визначити основні сутності бази даних, а саме: Користувач (users), Адміністратор (worker), Бронювання (animal), Звіт (reports). Після цього визначаються взаємозв'язки між сутностями: Користувач може отримати один або більше Звітів, Звіт належить до одного Користувача, Користувач може зробити одне або більше Бронювань, Бронювання включає одну або більше Послуг, Бронювання стосується одного Звіту, Звіт має один запис про Здоров'я. Далі визначаються атрибути сутностей: Користувач має атрибути номер Користувача, Логін, Пароль; Бронювання має атрибути Ім'я тварини, Вид тварини, Вік, Стан здоров'я, Назва Послуги, Дата Бронювання; Звіт має атрибути Номер Тварини, Температура, Якість Сну, Додаткові Зауваження, Фото, Відео. Та призначимо декілька первинних та альтернативних ключів для сутностей: Користувач має первинний ключ Телефон та альтернативний ключ Пошта; Бронювання має первинний ключ Ім'я Тварини та альтернативний ключ Вид+Тип; Адміністратор має первинний ключ Ім'я.

Тепер розглянемо окремо та детально кожен сутність. Сутність Користувач містить атрибути (табл. 3.3): id користувача, логін користувача, пароль

користувача, хеш користувача, айпі користувача. Нижче наведено опис кожного атрибуту:

- user\_id: це первинний ключ таблиці, який є цілим числом без знаку. Він автоматично збільшується при додаванні нового користувача та не може бути порожнім.
- user\_login: це ім'я користувача, яке є рядком з 30 символів, використовує кодування cp1251\_general\_ci, яке підтримує кирилицю. Також не може бути порожнім.
- user\_password: це пароль користувача, який є рядком з 32 символів. Він також використовує кодування cp1251\_general\_ci та не може бути порожнім.
- user\_hash: це хеш-код користувача, який є рядком з 32 символів. Він використовує кодування cp1251\_general\_ci та може бути порожнім, якщо користувач не авторизований.
- user\_ip: це IP-адреса користувача, яка є цілим числом без знаку, не може бути порожньою і має значення 0 за замовчуванням.

Таблиця 3.3 – Сутність «Користувач»

№	Атрибут	Тип
1	user_id	int
2	user_login	varchar (3)
3	user_password	varchar (3)
4	user_hash	varchar (3)
5	user_ip	int

Наступна сутність – сутність Адміністратор містить атрибути (табл. 3.4): id адміністратора, ім'я, пароль адміністратора. Нижче наведено опис кожного атрибуту:

- id: це первинний ключ таблиці, який є цілим числом. Він автоматично збільшується при додаванні нового користувача та не може бути порожнім.

- name: це ім'я адміністратора, яке є рядком з 255 символів. Воно використовує кодування utf8mb4\_unicode\_ci, яке підтримує унікод, не може бути порожнім.
- password: це пароль адміністратора, який є рядком з 255 символів. Він також використовує кодування utf8mb4\_unicode\_ci та не може бути порожнім.

Таблиця 3.4 – Сутність «Адміністратор»

№	Атрибут	Тип
1	id	int
2	name	varchar (255)
3	password	varchar (255)

Далі розглянемо сутність Звіти, яка містить такі атрибути (табл. 3.5): id, id тварини, температура, якість сну, додаткові записи, шлях до фото, шлях до відео, дата звіту. Нижче наведено опис кожного атрибуту:

- id: це первинний ключ таблиці, який є цілим числом. Він автоматично збільшується при додаванні нового запису. Він не може бути порожнім.
- animal\_id: це зовнішній ключ таблиці, який є цілим числом. Він посилається на id тварини з іншої таблиці. Він може бути порожнім або мати значення NULL, якщо тварина не визначена.
- temperature: це температура тіла тварини, яка є десятковим числом з п'ятьма цифрами перед комою та однією після. Вона вимірюється у градусах Цельсія. Вона може бути порожньою або мати значення NULL, якщо температура не виміряна.
- sleep\_quality: це якість сну тварини, яка є цілим числом. Вона вимірюється у балах від 1 до 10, де 1 означає поганий сон, а 10 означає хороший сон. Вона може бути порожньою або мати значення NULL, якщо сон не оцінений.
- additional\_notes: це додаткові зауваження про тварину, які є текстом. Вони використовують кодування utf8mb4\_0900\_ai\_ci, яке підтримує унікод.

Вони можуть бути порожніми або мати значення NULL, якщо зауважень немає.

- photo\_path: це шлях до фотографії тварини, який є рядком з 255 символів. Він використовує кодування utf8mb4\_0900\_ai\_ci. Він може бути порожнім або мати значення NULL, якщо фотографії немає.
- video\_path: це шлях до відео тварини, який є рядком з 255 символів. Він використовує кодування utf8mb4\_0900\_ai\_ci. Він може бути порожнім або мати значення NULL, якщо відеошлях не вказано.
- report\_date: це дата створення запису про тварину, яка використовує значення date. Вона використовує формат рік-місяць-день при формуванні звіту та також може бути порожньою або мати значення NULL, якщо дата створення звіту не вказана.

Таблиця 3.5 – Сутність «Звіт»

№	Атрибут	Тип	№	Атрибут	Тип
1	id	int	4	additional_notes	text
2	animal_id	int	5	photo_path	varchar (255)
2	temperature	demical (3,5)	6	video_path	varchar (255)
3	sleep_quality	int	7	report_date	date

Остання сутність – сутність Бронювання, яка містить такі атрибути (табл. 3.6): id, ім'я, телефон, пошта, тварина, порода тварини, дата, кімната, процедури манікюру, гігієнічні процедури, розмір тварини для гігієнічних процедур, розмір тварини для лікувальних процедур, процедура гурмінгу, ім'я тварини, вік тварини, здоров'я тварини. Нижче наведено опис кожного атрибуту:

- id: це первинний ключ таблиці, який є цілим числом. Він автоматично збільшується при додаванні нового замовлення та не може бути порожнім.
- name: це ПІБ користувача, який оформив бронювання. Цей атрибут є рядком з 255 символів. Він використовує кодування utf8mb4\_unicode\_ci,

яке підтримує унікод, може бути порожнім або мати значення NULL, якщо ім'я не вказане.

- phone: це телефонний номер користувача, який оформив бронювання. Він є рядком з 255 символів, також використовує кодування utf8mb4\_unicode\_ci. Він може бути порожнім або мати значення NULL, якщо номер не вказаний.
- mail: це електронна пошта користувача, який оформив бронювання, яка є рядком з 255 символів. Вона також використовує кодування utf8mb4\_unicode\_ci. Вона може бути порожньою або мати значення NULL, якщо пошта не вказана.
- pet: це тип тварини, для якої оформлено бронювання, також є рядком з 255 символів та використовує кодування utf8mb4\_unicode\_ci. Атрибут може бути порожнім або мати значення NULL, якщо ім'я не вказане.
- animal\_type: це порода тварини, для якої оформлено бронювання, атрибут є рядком з 255 символів, також використовує кодування utf8mb4\_unicode\_ci та не може бути порожнім.
- date: це дата замовлення, яка також використовує кодування utf8mb4\_unicode\_ci. Вона може бути порожньою або мати значення NULL, якщо дата не вказана.
- room: це номер кімнати, в якій розміщена тварина, який є рядком з 255 символів. Використовує кодування utf8mb4\_unicode\_ci, може бути порожнім або мати значення NULL, якщо кімната не вказана.
- manicure\_procedure: це назва процедури манікюру, яку замовлено для тварини, є рядком з 255 символів, використовує кодування utf8mb4\_unicode\_ci та може бути порожньою або мати значення NULL, якщо процедура не вказана.
- hygienic\_procedure: це назва гігієнічної процедури, яку замовлено для тварини, яка є рядком з 255 символів, використовує кодування utf8mb4\_unicode\_ci, може бути порожньою або мати значення NULL, якщо процедура не вказана.

- `hygienic_procedure_breed`: це назва породи тварини, для якої замовлено гігієнічну процедуру. Вона використовує кодування `utf8mb4_unicode_ci`, представлена рядком з 255 символів, може бути порожньою або мати значення `NULL`, якщо порода не вказана.
- `medicinal_procedure_breed`: це тип породи тварини, для якої замовлено лікувальну процедуру, також використовує кодування `utf8mb4_unicode_ci` та є рядком з 255 символів. Атрибут може бути порожньою або мати значення `NULL`, якщо порода не вказана.
- `grooming_procedure`: це назва процедури стрижки, яку замовлено для тварини, є рядком з 255 символів, використовує кодування `utf8mb4_unicode_ci`, також може бути порожньою або мати значення `NULL`, якщо процедура не вказана.
- `pet_name`: це ім'я тварини, для якої замовлено процедуру, є рядком з 255 символів та також використовує кодування `utf8mb4_unicode_ci`. Атрибут може бути порожнім чи мати значення `NULL`, якщо ім'я не вказане.
- `pet_age`: це вік тварини, для якої замовлено процедуру, є рядком з 255 символів. Цей атрибут також використовує кодування `utf8mb4_unicode_ci`. Він може бути порожнім або мати значення `NULL`, якщо вік не вказаний.
- `pet_health`: це стан здоров'я тварини, для якої замовлено процедуру, є рядком з 255 символів та також використовує кодування `utf8mb4_unicode_ci`. Такий атрибут може бути порожнім чи мати значення `NULL`, якщо стан не вказаний.

Під час створення бази даних для веб-сервісу готелю для тварин із системою контролю їх здоров'я, ми починаємо з визначення основних понять та відношень, що характеризують область. Показати базу даних буде ліпше у вигляді моделі “сутність-зв'язок”, яка відображає, як поняття пов'язані між собою. В цій моделі використовується тип зв'язку “один-до-багатьох”, який означає, що одне поняття може мати багато понять, пов'язаних з ним.

Таблиця 3.6 – Сутність «Бронювання»

№	Атрибут	Тип	№	Атрибут	Тип
1	id	int	10	manicure_procedure	varchar (255)
2	name	varchar (255)	11	hygienic_procedure	varchar (255)
3	phone	varchar (255)	12	hygienic_procedure_breed	varchar (255)
4	mail	varchar (255)	13	medicinal_procedure_breed	varchar (255)
5	pet	varchar (255)	14	grooming_procedure	varchar (255)
6	animal_type	varchar (255)	15	pet_name	varchar (255)
7	date	varchar (255)	15	pet_age	varchar (255)
8	room	varchar (255)	16	pet_health	varchar (255)

Діаграма “сутність-зв’язок” для бази даних наведена на рисунку 3.10.

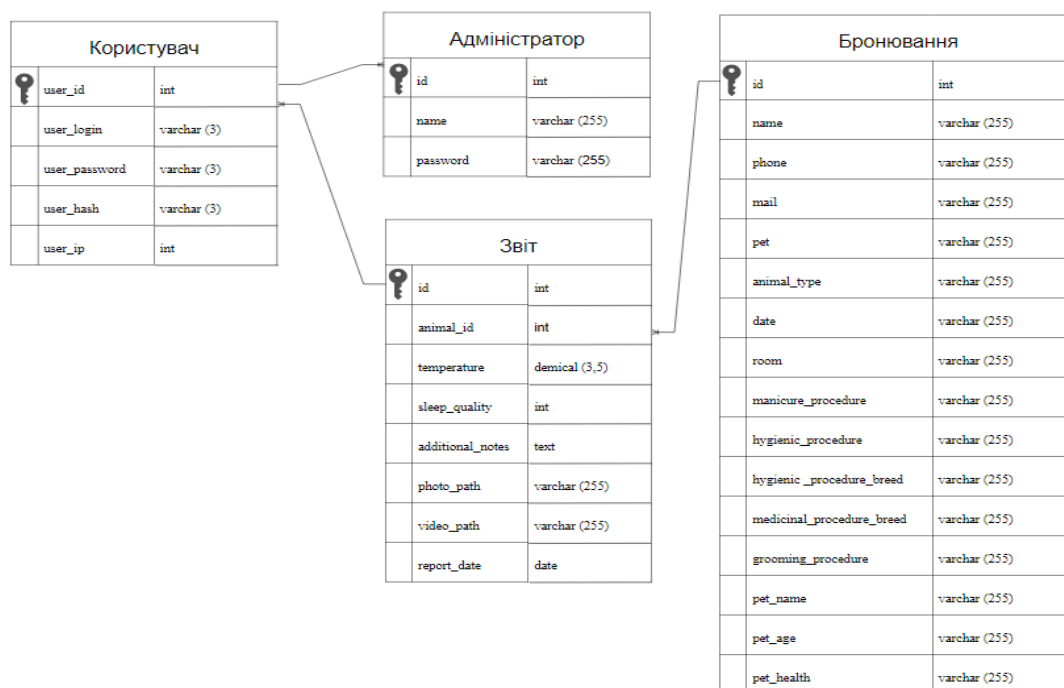


Рисунок 3.10 – Діаграма бази даних «сутність-зв'язок»

Таким чином між сутностями “Користувач” та “Звіт” існує зв’язок один-до-багатьох: багато звітів можуть належати одному користувачу. Між сутностями “Адміністратор” та “Користувач” існує зв’язок один-до-багатьох: багато користувачів можуть бути оформлені адміністраторами. Та між сутностями “Звіт” та “Бронювання” існує зв’язок один-до-багатьох: багато звітів можуть стосуватися

одного бронювання. Ці зв'язки створюють структуровану базу даних, де інформація з однієї таблиці може бути використана для збагачення даних в іншій таблиці, що дозволяє ефективно управляти і аналізувати дані.

### **3.4 Проектування інтерфейсу користувача**

Веб-сервіс готелю для тварин із системою контролю їх здоров'я - це зручний ресурс, який був розроблений з метою надання користувачам простого та ефективного інструменту для управління бронюваннями, контролю стану здоров'я їх улюбленців та взаємодії з оформленням бронювання.

Інтерфейс користувача (UI) розроблений таким чином, щоб забезпечити простоту використання, зручність та ефективність [19]. Він включає в себе ряд ключових елементів, таких як головні елементи сторінок, особистий кабінет користувача, систему бронювання та перегляд звіту про стан здоров'я тварин. Ці елементи розроблені з урахуванням потреб користувачів та спрямовані на забезпечення простоти користування сервісом. Він адаптований для різних пристроїв та розмірів екрану, що забезпечує зручне використання веб-сервісу на будь-якому пристрої. Інтерфейс розроблений таким чином, щоб користувачам було зрозуміло, як інтуїтивно використовувати різні функції без необхідності довгого вивчення процесу.

Адаптація веб-сервісу. Веб-сервіс готелю для тварин із системою контролю їх здоров'я адаптований до різних сценаріїв використання та потреб користувачів. Він враховує різні типи тварин, їх розміри, породи, характеристики та особливості, що допомагає користувачам знайти найкращий варіант розміщення та догляду для своїх улюбленців. Він також надає користувачам можливість вибирати різні типи послуг, такі як догляд, грумінг, гігієнічні та лікувальні процедури тощо, що забезпечує індивідуальний підхід до кожної тварини.

Адекватність інтерфейсу. Інтерфейс веб-сервісу готелю для тварин із системою контролю їх здоров'я адекватний до цілей та завдань, які ставляться перед користувачами. Він надає користувачам всю необхідну інформацію про



готель, умови проживання, ціни, відгуки та контактні дані, що дозволяє користувачам приймати рішення про бронювання. Він також надає користувачам зручний та швидкий спосіб бронювання, а також отримання даних про бронювання, надає користувачам звіти про стан контролю здоров'я тварин, яка дозволяє користувачам переглядати та відстежувати стан здоров'я своїх улюбленців, отримувати повідомлення про будь-які зміни або проблеми, а також зв'язуватися з адміністрацією готелю для отримання більш детальної інформації.

Інші аспекти. Інтерфейс веб-сервісу готелю для тварин також враховує інші аспекти, які впливають на задоволення користувачів та якість сервісу. Він має привабливий та простий дизайн, який використовує зручні для сприйняття кольори, шрифти, іконки та зображення. Він має чітку та логічну структуру, яка дозволяє користувачам легко знаходити та переходити до різних розділів та функцій, має зрозумілу та необхідну навігацію, яка дозволяє користувачам легко повертатися до попередніх сторінок та входити до особистого кабінету. Веб-сервіс має інтерактивний інтерфейс, який відповідає на запити користувачів та надає їм поточну інформацію, має високу продуктивність та надійність, які забезпечують швидке та безперебійне функціонування та має надійне зберігання даних, яке гарантує конфіденційність та цілісність інформації користувачів та тварин.

Детальніше розглянемо ключові елементи користувача та почнемо з головних елементів сторінки. При вході на веб-сервіс, користувач потрапляє на головну сторінку, де одразу бачить інтерактивне меню із доступними посиланнями на елементи сторінки: номери, тарифи, контакти, блог, улюбленці та послуги (рис. 3.11).

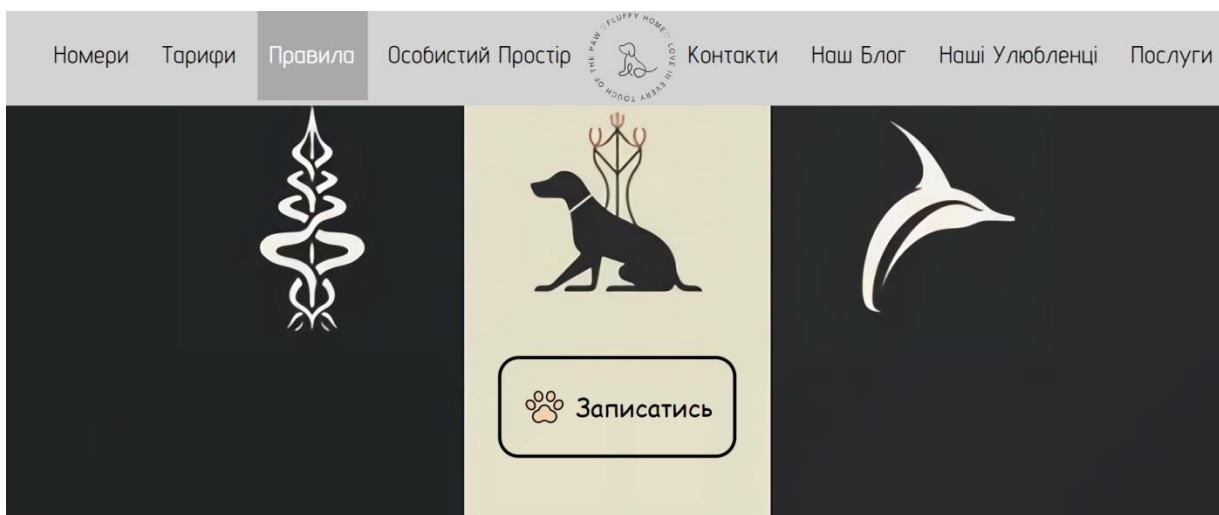


Рисунок 3.11 – Головна сторінка веб-сервісу із інтерактивним меню

Для ознайомлення з більш детальною інформацією, користувач може перейти до потрібного йому розділу, натиснувши на елемент у інтерактивному меню. Кожен розділ надає користувачам можливість отримати більш детальну інформацію про різні аспекти веб-сервісу. Він може містити пояснення ключових функцій, особливостей та переваг сервісу. Також, тут можуть бути розміщені додаткові фотографії, відеоматеріали та відгуки клієнтів. Приклад одного із розділів представлено на рисунку 3.12.



### Правила Прийому та Проживання

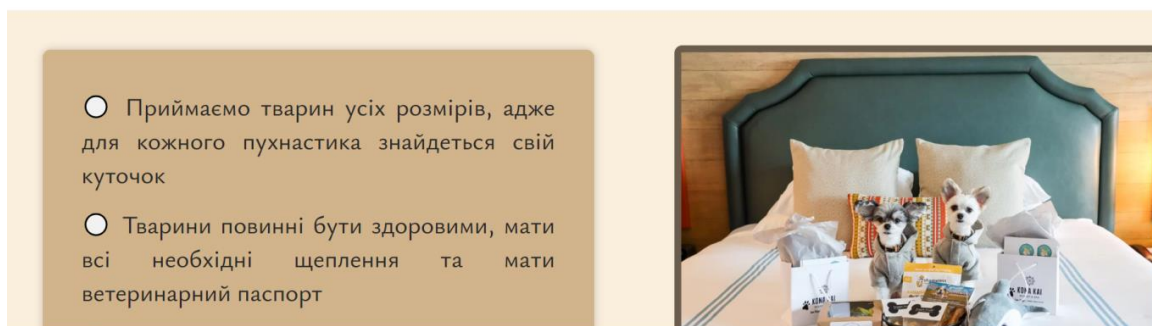
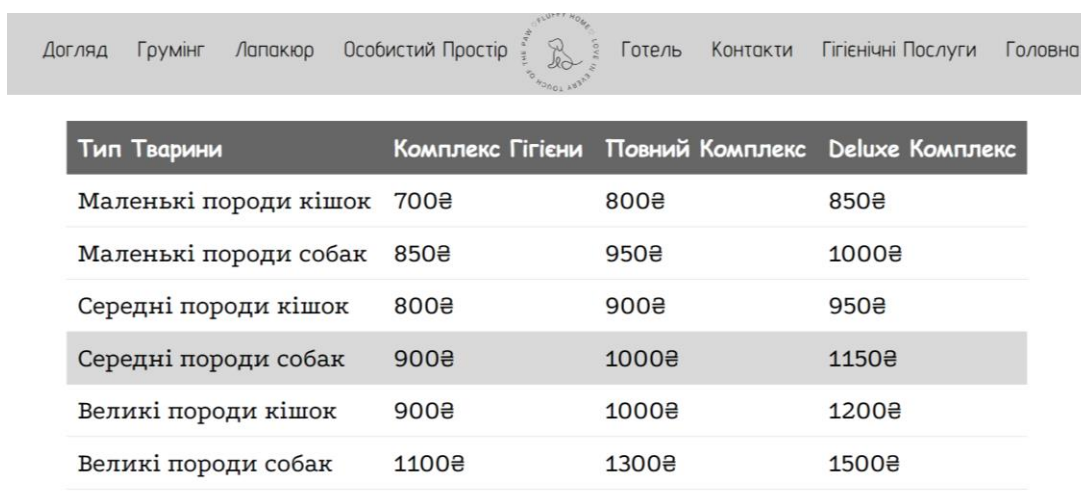


Рисунок 3.12 – Детальна інформація про один із розділів веб-сервісу

Коли користувач ознайомився із вмістом головної сторінки, він може перейти на другу сторінку та переглянути більш детальну інформацію про послуги (рис. 3.13).



The screenshot shows a navigation menu with the following items: [Догляд](#), [Грумінг](#), [Лапакюр](#), [Особистий Простір](#), [Готель](#), [Контакти](#), [Гігієнічні Послуги](#), and [Головна](#). Below the menu is a table with the following data:

Тип Тварини	Комплекс Гігієни	Повний Комплекс	Deluxe Комплекс
Маленькі породи кішок	700€	800€	850€
Маленькі породи собак	850€	950€	1000€
Середні породи кішок	800€	900€	950€
Середні породи собак	900€	1000€	1150€
Великі породи кішок	900€	1000€	1200€
Великі породи собак	1100€	1300€	1500€

Рисунок 3.13 – Друга сторінка веб-сервісу із детальною інформацією про послуги

Також веб-сервіс має адаптацію блоків, що є зручним для використання на девайсах із різними розмірами, таким веб-сервіс завжди готовий забезпечити користувачам зручний та ефективний досвід взаємодії. Приклад адаптації блоків представлено на рисунку 3.14.

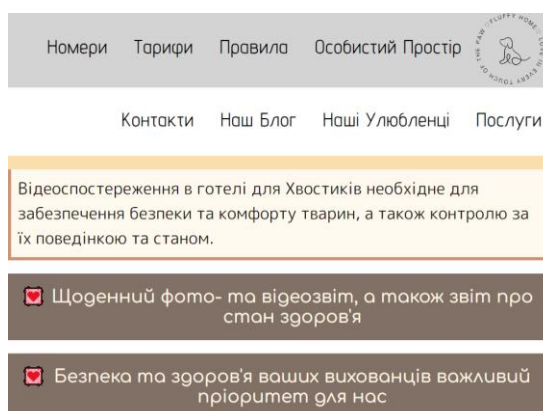


Рисунок 3.14 – Представлення адаптивності блоків веб-сервісу


Для зручності оформлення послуг, користувач має змогу заповнити форму бронювання (рис. 3.15) для спрощення отримання необхідних послуг. Форма бронювання має інформацію про контактні дані користувача та дані про його тварину, включаючи стан здоров'я, а також вибір послуг для тварини та дату бронювання.

The image shows a web form for booking services for a pet. The form is organized into several columns and sections. At the top left, there is a button labeled 'ПОВЕРНУТИСЯ НАЗАД'. The main form is divided into three primary columns: 'Забронювати номер' (Book a room), 'Послуги' (Services), and 'Дані Улюбленця' (Pet Information). The 'Забронювати номер' section includes fields for 'Ваше ПІБ' (Your name), 'Городська Олександра С', 'Ваш номер телефону' (+380660000000), and 'Ваша пошта' (saahnaakaprin@gmail.com). The 'Послуги' section includes 'Достав' (Delivery), 'Вибір виду тварини' (Select animal type) with a dropdown for 'Середні породи собак', 'Лижкояр', 'Вибір виду послуги' (Select service type) with a dropdown for 'Папакор+Пак', 'Гістичні Процедури', and 'Вибір процедури' (Select procedure) with a dropdown for 'Вичищення, видачення ковтунів'. The 'Дані Улюбленця' section includes 'Ваш улюбленець - кіт чи собака?' (Your pet - cat or dog?) with a dropdown for 'Кішка', 'Введіть ім'я улюбленця' (Enter pet name) with a text field 'Джулія', 'Вік улюбленця' (Pet age) with a text field '2 роки', and 'Стан здоров'я улюбленця' (Pet health status) with a dropdown for 'Здоровий'. To the right of these sections, there are additional dropdowns for 'Вибір породи' (Select breed) and 'Вибір послуги' (Select service) with a dropdown for 'Супка та укладання шерсті'. Below these are 'Тарифи' (Rates) with a dropdown for 'Повна ціна' and 'Обрати дату бронювання' (Select booking date) with a date field '14.12.2023'. At the bottom right, there is a button labeled 'СІВОРИТИ ЗАПИТ НА БРОНЮВАННЯ'.

Рисунок 3.15 – Форма бронювання послуг для користувача

Останнім ключовим елементом для користувача є особистий кабінет. Особистий кабінет – це простір користувача, де він має змогу передивлятися список своїх бронювань та отримувати звіт про стан здоров'я своєї тварини, щоб завжди бути в курсі їхнього благополуччя. Все це робить особистий кабінет не просто зручним інструментом, але й особистим помічником у справах нагляду за улюбленцями. Кабінет користувача представлено на рисунку 3.16

Головна Вийти



### Список Бронювань

ID	ПІБ клієнта	Номер телефону клієнта	Пошта	Догляд	Гігієнічні Процедури	Лалажор	Зняття каліту, видалення зубного каменя	Тварина	Лікувальні маски та оздоровчі SPA-процедури	Грумінг	Тариф	Ім'я тварини	Вік тварини	Стан здоров'я тварини	Дата на бронь
72	Гордієвська Олександра Олександрівна	+38066000000	sashaexample@gmail.com	Середні породи собак	Вичісування, видалення ковриків	Лапажор+Лак	Середні	Собака	Середні	Сушка та укладання шерсті	Полнастік Vip	Джуля	2 роки	Здоровий	16.12.2023 <a href="#">Переглянути звіг</a>

### Список Звігів

ID	ID тварини	Температура	Якість сну	Додаткові зауваження	Фото	Відео	Дата звігу
2	72	37.8	96	Ми оцінили загальний стан вашої собаки, і все виглядає чудово! Вона в хорошій фізичній формі і має блискучу шерсть. Проджекуємо дбати про неї, надаючи відповідну доглядову продукцію. Була доглянута спеціалістом, стан здоров'я прекрасний.	<a href="https://lilyhome/photo-sobaki-2">https://lilyhome/photo-sobaki-2</a>	<a href="https://lilyhome/video-sobaki-2">https://lilyhome/video-sobaki-2</a>	2023-11-15

Рисунок 3.16 – Особистий кабінет користувача

## **4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-СЕРВІСУ ГОТЕЛЮ ДЛЯ ТВАРИН**

### **4.1 Аналіз існуючих засобів розробки**

Розробка веб-сервісу для готелю тварин вимагає уважного аналізу та обрання оптимальних інструментів для забезпечення успішного функціонування та задоволення потреб користувачів. Перед початком аналізу інструментів розробки важливо визначити ключові аспекти, які визначатимуть ефективність, простоту та функціональність веб-сервісу [20].

Аналіз інструментів розробки для веб-сервісу готелю для тварин починається з визначення ключових аспектів, на які треба звернути особливу увагу. Серед них можливо виділити важливість забезпечення відповідності вимог користувачів, забезпечуючи зручний інтерфейс, легку навігацію та функціонал, спрямований на потреби власників тварин.

Надійність та забезпечення конфіденційності та цілісності інформації про тварин та їх власників також стають критичними факторами. Забезпечення безпеки даних вимагає відповідних заходів, що дозволяють побудувати довірчі відносини з користувачами [21].

Продуктивність та легка масштабованість також є не менш важливими критеріями. Швидкість роботи системи та її здатність адаптуватися до зростання обсягів даних чи користувачів гарантують плавну та ефективну роботу веб-сервісу в умовах розвитку та розширення його функціоналу.

На етапі аналізу технологій важливо розглядати різні мови програмування, середовища розробки та бази даних. Їх вибір повинен відповідати конкретним потребам проекту, а також сприяти простому та ефективному способу отримання даних про стан тварин.

Загальна мета аналізу полягає в створенні ефективної інфраструктури, яка відповідає сучасним стандартам розробки веб-сервісів. Сучасний підхід гарантує створення високоефективного сервісу для забезпечення потреб користувачів і сприяє успішному функціонуванню готелю для тварин із системою контролю їх здоров'я.

Створення веб-сервісу для готелю тварин визначає необхідність обрати найбільш підходяще середовище розробки для забезпечення успішної інтеракції з ідеєю та задоволення високих вимог користувачів. Справжній вибір середовища розробки визначає якість коду, продуктивність команди, а також майбутню легкість управління та масштабованість системи [22].

У контексті ключового етапа в пошуку оптимального рішення для розробленого проекту, вибір пав на Sublime Text (рис.4.1), який є текстовим редактором для розробки програмного забезпечення та веб-розробки. Він відзначається своєю простотою використання та легкістю інтерфейсу, що робить його популярним серед розробників. Sublime Text пропонує широкий набір функцій, які полегшують написання коду, такі як підсвічування синтаксису, автоматичне завершення коду, робота з великими файлами, підтримка плагінів, аналіз середовища розробки. Детальне вивчення цього інструменту допоможе зробити обґрунтований вибір, спрямований на досягнення високих стандартів ефективності та якості веб-сервісу. Надалі розглянемо переваги, недоліки та особливості Sublime Text у контексті створення готелю для тварин із системою контролю їх здоров'я.

#### Переваги використання Sublime Text:

- Простота та легкість використання: Sublime Text славиться не лише своєю простотою в освоєнні, а також інтуїтивністю інтерфейсу, що сприяє зручному введенню у проект і забезпечує комфортну роботу з кодом. Мінімалістичний дизайн та проста навігація сприяють швидкому освоєнню інструменту.
- Швидкість та ефективність: Швидкість завантаження та реакції на дії під час розробки роблять Sublime Text ефективним інструментом для роботи над різноманітними завданнями. Висока продуктивність дозволяє швидко переходити між файлами та фрагментами коду, що прискорює процес розробки.

- Плагіни: Однією з важливих переваг є можливість використання різноманітних плагінів, які дозволяють розробникам налаштовувати та розширювати функціонал інструменту. Це забезпечує гнучкість у роботі та адаптацію під конкретні вимоги проекту, що є важливим для індивідуалізації розробки.
- Мультиплатформенність: Сумісність із різними операційними системами (Windows, macOS, Linux) дозволяє працювати в середовищі, незалежно від операційної системи.

#### Недоліки використання Sublime Text:

- Відсутність розширених засобів дебагу: Порівняно з іншими інтегрованими середовищами розробки, Sublime Text може виявитися менш зручним для детального дебагу коду. Обмежена функціональність для відлагодження може стати проблемою при вирішенні складних або неочевидних помилок.
- Обмежені можливості для великих та складних проектів: У великих та складних проектах Sublime Text може виявитися менш потужним, оскільки в нього можуть бракувати деякі інструменти для ефективного керування великими обсягами коду та ресурсів. При великій кількості файлів та складних залежностей відсутність певних інструментів може ускладнити розробку та підтримку проекту.

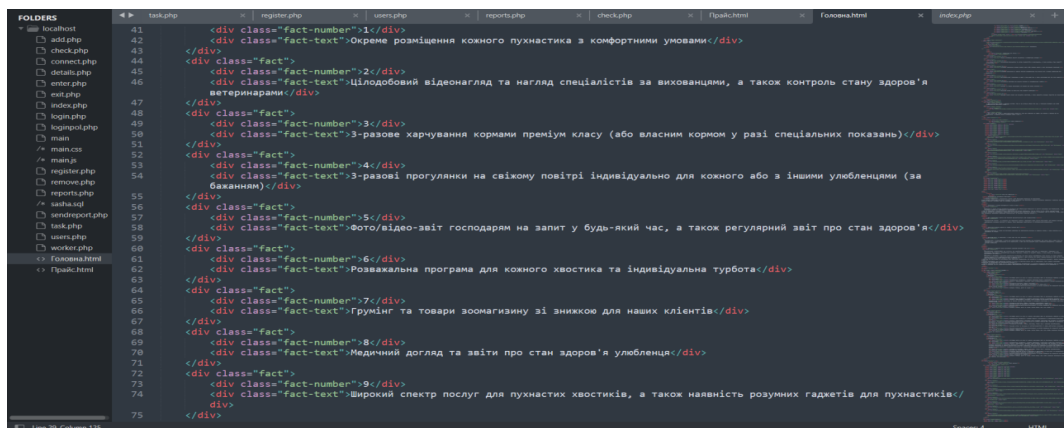


Рисунок 4.1 – Зовнішній вигляд Sublime Text



Основний каркас для подальшого розвитку надає HTML (HyperText Markup Language), а також відіграє ключову роль у визначенні структури та вмісту веб-сервісу готелю для тварин (рис. 4.2). HTML є мовою розмітки, яка визначає структуру та вміст веб-сторінок є фундаментом для будь-якого веб-сервісу. Вона складається з набору тегів, які оточують різні елементи на сторінці та вказують браузеру, як правильно їх відобразити.

Розглянемо основні особливості HTML, які стають дуже важливими в контексті створення ефективного веб-сервісу:

- Теги та структура: HTML використовує різні теги для визначення різних елементів на сторінці. Наприклад, тег `<head>` визначає заголовок сторінки, `<body>` – тіло сторінки, а `<p>` – абзац тексту. Ці теги утворюють структуру, яка полегшує читання та розуміння коду.
- Атрибути: Теги можуть також містити атрибути, які надають додаткову інформацію про елемент. Наприклад, тег `<a>` (гіперпосилання) може містити атрибут `href`, який вказує URL посилання.
- Форми та взаємодія: HTML надає спеціальні теги для створення форм, які дозволяють користувачам взаємодіяти з веб-сайтом. Теги, такі як `<input>`, `<textarea>`, `<select>`, дозволяють збирати від користувачів різні дані, такі як ім'я, телефон чи вибір елементів із списку.
- Списки: HTML дозволяє створювати різні типи списків, такі як нумеровані (`<ol>`), марковані (`<ul>`), та списки з вкладеннями. Це корисно для представлення інформації в порядку чи групування елементів.
- Гіперпосилання та навігація: Використання тега `<a>` для створення гіперпосилань дозволяє користувачам легко переходити між різними сторінками веб-сайту, що особливо важливо для навігації по різних розділах веб-сервісу.
- Вставка мультимедіа: За допомогою HTML можливо легко вставляти мультимедійні елементи, такі як зображення чи відео, щоб візуально представити особливості веб-сервісу.

```

24     </nav>
25 </div>
26 </header>
27 <div class="image-container">
28 <a class="new">
29 
30 <p class="image-caption">Прайс-Лист На Послуги</p>
31 </a>
32 </div>
33
34 <main>
35 <section class="colore">
36 <h2 id="care" class="centered">Догляд</h2>
37 <br><p>Ці послуги Ви можете замовити окремо від проживання чи у додаток до нього (у ціну включено купання та розчісування)</p></br>
38 <table>
39 <tr>
40 <th>Тип Тваринки</th>
41 <th>Комплекс Гігієни</th>
42 <th>Повний Комплекс</th>
43 <th>Deluxe Комплекс</th>
44 </tr>
45 <tr>
46 <td>Маленькі породи кішок</td>
47 <td>700€</td>
48 <td>800€</td>
49 <td>850€</td>
50 </tr>
51 <tr>
52 <td>Маленькі породи собак</td>
53 <td>850€</td>
54 <td>950€</td>
55 <td>1000€</td>
56 </tr>
57 </table>

```

Рисунок 4.2 – Представлення розмітки сторінки послуг

Проаналізуємо далі ще один ключовий аспект, який визначатиме користувацький досвід та функціональність – стилізацію та дизайн за допомогою мови CSS (рис. 4.3). CSS (Cascading Style Sheets) – це мова стилізації, яка визначає, як HTML елементи будуть відображатися на екрані користувача. CSS дозволяє контролювати вигляд та форматування веб-сервісу, забезпечуючи його зовнішній вигляд та структуру. У контексті веб-сервісу для готелю тварин, CSS є ключовим інструментом для створення естетичного та користувацьки-орієнтованого інтерфейсу.

Основні концепції CSS включають в себе використання правил стилю, які визначають, як кожен елемент повинен виглядати. Селектори вказують, які елементи на сторінці підлягають змінам стилю, а властивості задають конкретні аспекти вигляду, такі як кольори, розміри шрифту, відступи.

Каскадність у CSS дозволяє встановлювати пріоритети стилів, а спадкування дозволяє елементам успадковувати стилі від їхніх батьківських елементів. Це робить створення і управління стилями більш систематизованим та легким процесом.

```

1 @import url('https://fonts.googleapis.com/css2?family=Caveat:wght@500&display=swap');
2 @import url('https://fonts.googleapis.com/css2?family=Advent+Pro:wght@200&display=swap');
3 @import url('https://fonts.googleapis.com/css2?family=Podkova&display=swap');
4 @import url('https://fonts.googleapis.com/css2?family=Klee+One:wght@600&display=swap');
5 @import url('https://fonts.googleapis.com/css2?family=Comfartaa:wght@500&display=swap');
6 @import url('https://fonts.googleapis.com/css2?family=M+PLUS+Rounded+1c&display=swap');
7 @import url('https://fonts.googleapis.com/css2?family=Roboto+Flex:opsz@8..144&display=swap');
8 @import url('https://fonts.googleapis.com/css2?family=Ysabeau:wght@500&display=swap');
9 @import url('https://fonts.googleapis.com/css2?family=Kurale&display=swap');
10 @import url('https://fonts.googleapis.com/css2?family=Playfair:opsz@5..1200&display=swap');
11 body {
12   background-color: white;
13   font-size: 36px;
14   font-family: 'Podkova', serif;
15   margin: 0;
16 }
17 .image-caption {
18   text-align: center;
19   font-size: 90px;
20   color: white;
21   background: rgba(0, 0, 0, 0.7);
22   padding: 10px;
23   position: absolute;
24   top: 40%;
25   left: 50%;
26   transform: translateX(-50%);
27   white-space: nowrap;
28   font-family: 'M PLUS Rounded 1c', sans-serif;
29 }
30 .container {
31   display: flex;
32   justify-content: space-between;
33 }
34
35 .menu-container {
36   background: white;
37   position: fixed;
38   width: 100%;

```

Рисунок 4.3 – Структура коду стилізації

Наступним пунктом є використання JavaScript у створенні веб-сервісу для забезпечення інтерактивності, ефективності та зручності для використання користувачами (рис. 4.4). JavaScript - це високорівнева мова програмування, яка використовується для створення динамічного контенту на веб-сторінках та забезпечення їхньої інтерактивності. JavaScript дозволяє реалізувати адаптивний та інтуїтивний інтерфейс при взаємодії з різними функціональними елементами веб-сервісу. Вона підтримується всіма сучасними браузерами і є важливим інструментом для створення сучасних та інтерактивних веб-серверів. Розглянемо переваги та обмеження цього інструменту, визначаючи його ключовий вплив на розробку проекту.

Переваги використання JavaScript:

- Широке охоплення: JavaScript є стандартом веб-розробки і підтримується всіма сучасними браузерами, що робить його універсальним інструментом для створення інтерактивних веб-сервісів.

- **Інтерактивність:** Мова дозволяє створювати динамічний вміст, який може реагувати на дії користувача без перезавантаження сторінки, що поліпшує зв'язок із користувачем та сприяє більш ефективній взаємодії з веб-сервісом.
- **Зручність для розробників:** JavaScript має простий синтаксис, що полегшує вивчення та розробку. Наявність різноманітних бібліотек і фреймворків сприяє швидкій інтеграції нових функцій.
- **Асинхронність:** Підтримка асинхронного програмування дозволяє виконувати операції паралельно та не блокує інші частини коду, що сприяє підвищенню продуктивності веб-сервісу.
- **Легке впровадження:** JavaScript можна легко вбудовувати безпосередньо в HTML-код сторінки, що спрощує процес впровадження та розгортання веб-серверів.

#### Недоліки використання JavaScript:

- **Безпека:** Одним із найбільших недоліків JavaScript є можливість створення вразливих місць, які можуть бути використані для атак на безпеку веб-сервісу. Неналежно забезпечений код може призвести до витоку конфіденційної інформації або інших серйозних проблем.
- **Різні інтерпретації:** JavaScript може вести себе по-різному в різних браузерах, що може призвести до проблем з сумісністю та вимагати додаткових зусиль для вирішення цих питань.
- **Залежність від клієнта:** Оскільки JavaScript виконується на браузері клієнта, користувач може вимкнути виконання скриптів, що призведе до втрати функціоналу та відмови від деяких можливостей веб-сервісу.
- **Перезавантаження сторінки:** Інтенсивне використання JavaScript може призвести до повільного завантаження сторінки, особливо при повільних інтернет-з'єднаннях або застарілих пристроях.

- Необхідність тестування: Враховуючи різні версії браузерів та платформ, тестування JavaScript стає важливим етапом, що може вимагати значних зусиль та ресурсів для забезпечення коректної роботи на всіх пристроях.
- Вплив на продуктивність: Використання великої кількості скриптів та бібліотек може призвести до високого використання системних ресурсів, що може вплинути на продуктивність та витрати енергії пристроїв.

Незважаючи на кількість недоліків, грамотне використання та адаптація JavaScript до конкретних вимог проекту дозволяє досягти високої функціональності та забезпечення задоволення потреб користувачів під час використання веб-сервісу готелю для тварин.

```

mainjs
1 // Визначення функції для адаптації сторінки
2 function adaptPage() {
3     // Отримання ширини вікна
4     var windowHeight = window.innerWidth;
5     // Виклик функції для адаптації шапки
6     adaptHeader(windowWidth);
7     // Виклик функції для адаптації блоків
8     adaptBlocks(windowWidth);
9     // Додаткова можливість адаптації: зміна розміру тексту в блоках
10    blockElements[i].style.fontSize = windowHeight < 768 ? '14px' : '18px';
11 }
12 // Функція для адаптації розміру шапки
13 function adaptHeader(windowWidth) {
14     // Отримання елемента шапки за його ідентифікатором
15     var headerElement = document.getElementById('MenuHeaderId');
16     // Адаптація розміру шрифту в залежності від ширини вікна
17     headerElement.style.fontSize = windowHeight < 768 ? '16px' : '24px';
18 }
19 // Функція для адаптації розміру блоків
20 function adaptBlocks(windowWidth) {
21     // Отримання колекції елементів з класом 'BlockClass'
22     var blockElements = document.getElementsByClassName('BlockClass');
23     // Перебір всіх елементів і адаптація ширини
24     for (var i = 0; i < blockElements.length; i++) {
25         // Адаптація ширини блоків в залежності від ширини вікна
26         blockElements[i].style.width = windowHeight < 768 ? '100%' : '50%';
27     }
28 }
29 }
30 // Виклик функції при завантаженні та зміні розміру вікна
31 window.addEventListener('load', adaptPage);
32 window.addEventListener('resize', adaptPage);
33 function displayImage() {
34     var currentImage = images[currentIndex];
35     // Отримати елемент для відображення зображення
36     var displayElement = document.getElementById('imageDisplay');
37 }
38

```

Рисунок 4.4 – Представлення коду JavaScript

Ще однією важливою мовою при створенні проекту є мова програмування РНР (Nurertext Preprocessor), яка призначена для створення динамічних веб-сервісів та стає невід'ємною складовою створення інтерактивного та ефективного веб-сервісу готелю для тварин (рис. 4.5). РНР відзначається здатністю легко взаємодіяти з базами даних, що робить її ідеальним вибором для обробки

бронювань, зберігання даних про клієнтів та їх тварин, а також надання персоналізованої інформації. Його гнучкість та простота використання в поєднанні з широким спектром можливостей дозволяють PHP ефективно інтегруватися у проект, надаючи йому потужний інструментарій для втілення всіх необхідних функцій веб-сервісу [23].

У контексті веб-сервісу готелю для тварин, PHP може забезпечити ряд особливостей, які сприяють як функціональності, так і ефективності:

- Взаємодія з базою даних: PHP ідеально підходить для взаємодії з базами даних, що дозволяє зберігати та керувати інформацією про клієнтів, бронювання та іншими важливими даними та може легко використовуватися для взаємодії з системами управління базами даних, такими як MySQL або PostgreSQL.
- Обробка форм та введення даних: PHP дозволяє ефективно обробляти дані, які користувачі вводять через веб-форми. Це важливо для реалізації функціоналу бронювання та взаємодії з користувачами, яка знаходиться в пошуках сервісу для своїх тварин.
- Активне управління даними: PHP надає ефективні засоби роботи з базами даних, що робить можливим легке зберігання та оновлення інформації про клієнтів, їхні бронювання та інші ключові дані. Це робить процес управління даними прозорим та ефективним.
- Реалізація системи реєстрації та авторизації: PHP дозволяє створити безпечні системи реєстрації та авторизації користувачів, що забезпечує конфіденційність та безпеку даних.
- Інтеграція з іншими технологіями: PHP може бути ефективно інтегрований з іншими технологіями, такими як HTML, CSS, JavaScript, що дозволяє створювати повноцінний та сучасний функціонал для готелю для тварин із системою контролю їх здоров'я.

```

1 <?php
2 require_once 'connect.php';
3
4 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
5     $name = $_POST['name'];
6     $phone = $_POST['phone'];
7     $mail = $_POST['mail'];
8     $animalType = $_POST['pet'];
9     $manicureProcedure = $_POST['manicure_procedure'];
10    $hygienicProcedure = $_POST['hygienic_procedure'];
11    $shygenicProcedureBreed = $_POST['hygienic_procedure_breed'];
12    $medicinalProcedureBreed = $_POST['medicinal_procedure_breed'];
13    $groomingProcedure = $_POST['grooming_procedure'];
14    $room = $_POST['room'];
15    $petName = $_POST['pet_name'];
16    $petAge = $_POST['pet_age'];
17    $petHealth = $_POST['pet_health'];
18    $date = $_POST['date'];
19    // Додаємо дані в базу даних
20    mysqli_query($connect, "INSERT INTO `animal` (`name`, `phone`, `mail`, `pet`, `date`, `room`, `animal_type`, `manicure_procedure`,
    `hygienic_procedure`, `hygienic_procedure_breed`, `medicinal_procedure_breed`, `grooming_procedure`, `pet_name`, `pet_age`,
    `pet_health`) VALUES ('$name', '$phone', '$mail', '$animalType', '$date', '$room', '$animalType', '$manicureProcedure', '$
    hygienicProcedure', '$shygenicProcedureBreed', '$medicinalProcedureBreed', '$groomingProcedure', '$petName', '$petAge', '$
    petHealth')");
21    header("Location: ".$_SERVER['HTTP_REFERER']);
22 }
23 ?>
24
25 <!DOCTYPE html >
26 <html>
27 <head>
28 <meta charset="UTF-8">
29 <link rel="stylesheet" href="..List/ListCSS/ListStyle.css">
30 </head>
31 <body>
32 <style type="text/css">
33 @import url("https://fonts.googleapis.com/css?family=Kaushan+Script|Libre+Baskerville");
34 body {

```

Рисунок 4.5 – Робота з кодом add.php для додавання бронювання у БД

Використання локального сервера є ключовим аспектом веб-розробки, і Open Server Panel виступає в ролі ефективного інструменту для цього завдання (рис. 4.6). Він надає можливість створення локального серверного середовища безпосередньо на комп'ютері, що дозволяє легко тестувати та вдосконалювати веб-проекти перед їх розгортанням в Інтернеті [24] .

Open Server Panel пропонує зручний інтерфейс для налаштування та управління серверними компонентами, такими як Apache, MySQL, PHP. Це спрощує робочий процес, дозволяючи швидко перевіряти та вносити зміни в код без необхідності завантаження на зовнішній сервер.

Завдяки Open Server Panel можна легко емулювати роботу реального сервера без необхідності публікації змін у відкритий доступ, а також забезпечити високу швидкість та зручність розробки, що є важливими перевагами, особливо на етапі тестування веб-сервісів.

У контексті створення готелю для тварин Open Server Panel виявляється надзвичайно корисним інструментом. Ось кілька аспектів, які особливо важливі для створення веб-сервісу:

- Локальне розгортання та тестування: Open Server Panel дозволяє легко створювати локальний сервер, де можливо розгортати та тестувати веб-сервіс перед його публікацією в Інтернеті, що дозволяє уникнути проблем, пов'язаних з віддаленими серверами та забезпечити стабільну роботу веб-сервісу готелю, перш ніж він стане доступним для широкого загалу.
- Швидка зміна та тестування коду: Open Server Panel забезпечує зручний інтерфейс для управління серверними компонентами та конфігурацією, що дозволяє легко вносити зміни у свій код і тестувати його без зайвих труднощів, прискорюючи процес розробки.
- Ізоляція проекту: Open Server Panel дозволяє створювати окремі веб-сервери для кожного проекту, що забезпечує ізольоване середовище для кожного веб-сервісу, допомагаючи уникнути конфліктів та зберігаючи чистоту коду та баз даних.
- Доступність на локальному рівні: Локальний сервер дозволяє працювати без доступу до Інтернету, що може бути важливим при відсутності стабільного з'єднання або на етапі розробки віддалених місць.

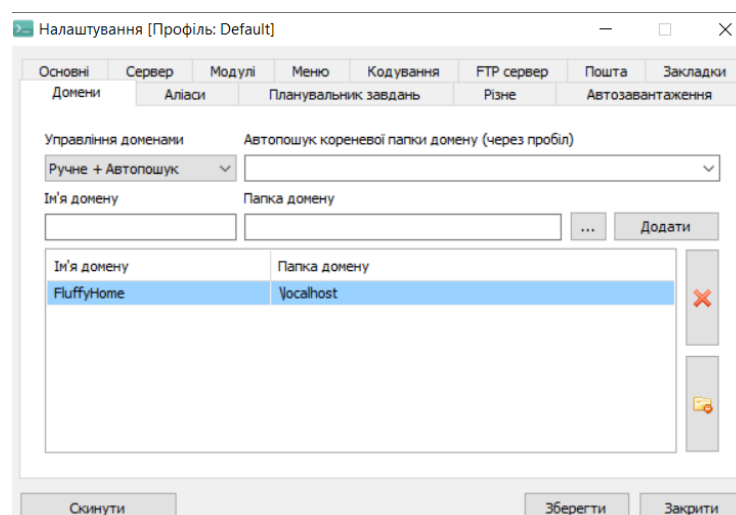


Рисунок 4.6 – Вікно налаштування домену локального серверного середовища  
Open Server Panel



Останнім головним пунктом аналізу розробки веб-сервісу готелю для тварин із контролем стану їх здоров'я є phpMyAdmin, що є веб-інтерфейсом для керування базами даних MySQL за допомогою веб-браузера (рис. 4.7). У контексті веб-сервісу, де обробка та управління даними важливі, phpMyAdmin забезпечує зручний інтерфейс для виконання завдань, таких як додавання нових записів, оновлення інформації та створення звітів. Він надає зручний спосіб взаємодії з базою даних, і ось декілька ключових аспектів, які важливі для створення проекту:

- Графічний Інтерфейс: phpMyAdmin забезпечує графічний інтерфейс, що робить роботу з базою даних більш доступною, особливо для тих, хто не має глибоких знань SQL. Ви можете легко виконувати запити, здійснювати зміни та адмініструвати базу даних безпосередньо через веб-браузер.
- Можливість створення та редагування БД: За допомогою phpMyAdmin можливо легко створювати нові бази даних та редагувати їхню структуру. Це важливо для веб-сервісу, оскільки можливо знадобиться вносити зміни в базу даних під час розширення функціональних можливостей веб-сервісу.
- Імпорт та експорт даних: phpMyAdmin дозволяє легко імпортувати та експортувати дані, що стає зручним для резервного копіювання бази даних, обміну інформацією або переміщення даних між різними середовищами.
- Управління таблицями та запити: Є можливість легко створювати та редагувати таблиці, виконувати SQL-запити, вставляти, оновлювати та видаляти дані, що надає повний контроль над вмістом бази даних.
- Безпека та автентифікація: phpMyAdmin дозволяє конфігурувати права доступу користувачів, забезпечуючи безпеку створеної бази даних, адже є можливість керувати тим, хто має доступ до різних частин бази даних, таким чином надаючи права перевіреною користувачам.

- Резервне Копіювання та Відновлення: Враховуючи важливість надійного збереження даних, phpMyAdmin дозволяє легко створювати резервні копії бази даних та відновлювати дані в разі потреби, що є важливим для запобігання втраті важливої інформації.
- Освідчений Стандарт Веб-Розробки: phpMyAdmin є широко використовуваним та визнаним стандартом для взаємодії з MySQL-базами даних. Він підтримується та оновлюється, що гарантує сумісність та актуальність для розробки веб-сервісів.

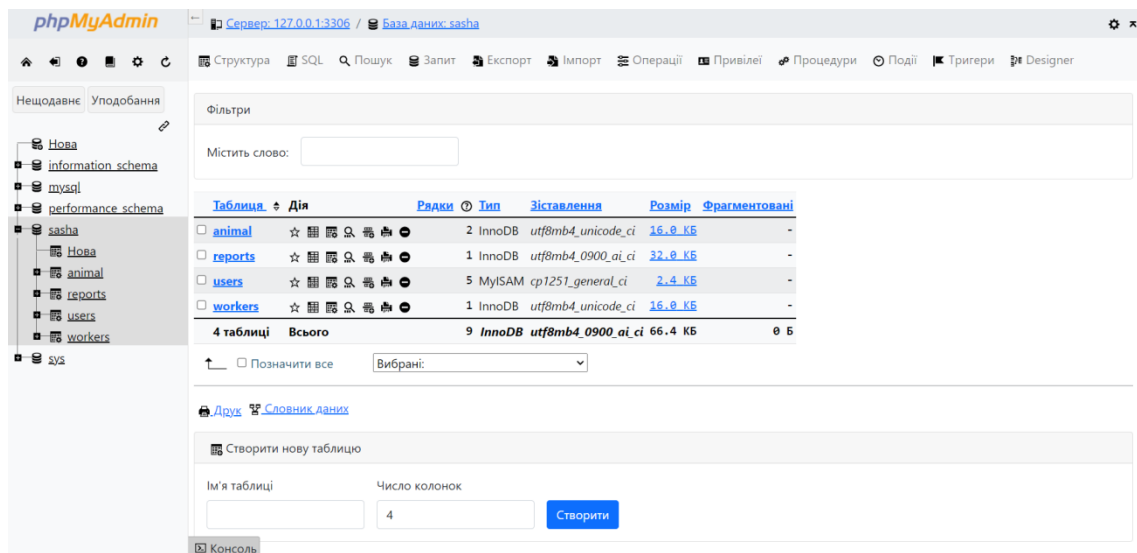


Рисунок 4.7 – Вікно перегляду БД у PhpMyAdmin

За всією цією інфраструктурою можна зробити висновок, що існує ефективна та комплексна система інструментів для розробки веб-сервісу готелю для тварин, яка охоплює всі аспекти від фронтенду до бекенду та бази даних, створюючи оптимальне середовище для розвитку та функціонування проекту [24]. Застосування таких мов програмування, як HTML, CSS, JavaScript та PHP, у поєднанні з базами даних MySQL та зручним інтерфейсом управління phpMyAdmin, надає необхідні ресурси для розробки і підтримки веб-сервісу.

## 4.2 Керівництво користувача

Перед початком робіт із веб-сервісом готелю для тварин, користувачу необхідно зайти у будь-який зручний для нього браузер та у пошуковому рядку ввести адресу веб-сервісу (рис. 4.8).



Рисунок 4.8 – Введення адреси веб-сервісу

Після успішного підключення до веб-сервісу, користувача перекидає на головну сторінку, де користувач може ознайомитися з усією необхідною інформацією веб-сервісу про готель для тварин (рис. 4.9).

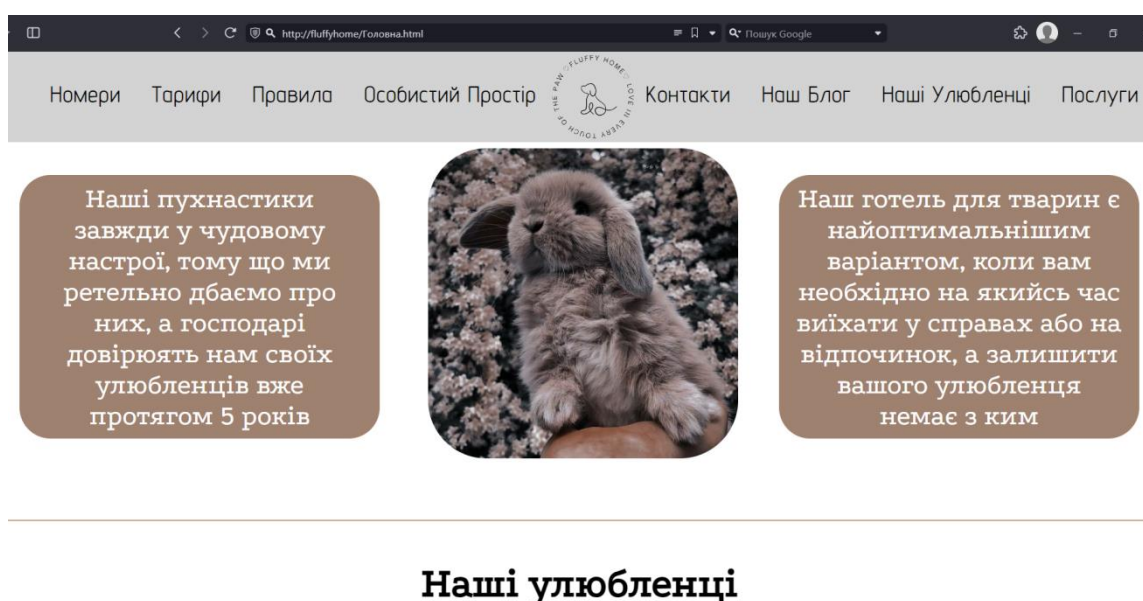


Рисунок 4.9 – Головна сторінка веб-сервісу для тварин

Користувач також має змогу переходити на блоки веб-сервісу використовуючи інтерактивне меню з посиланнями на елементи (рис. 4.10).

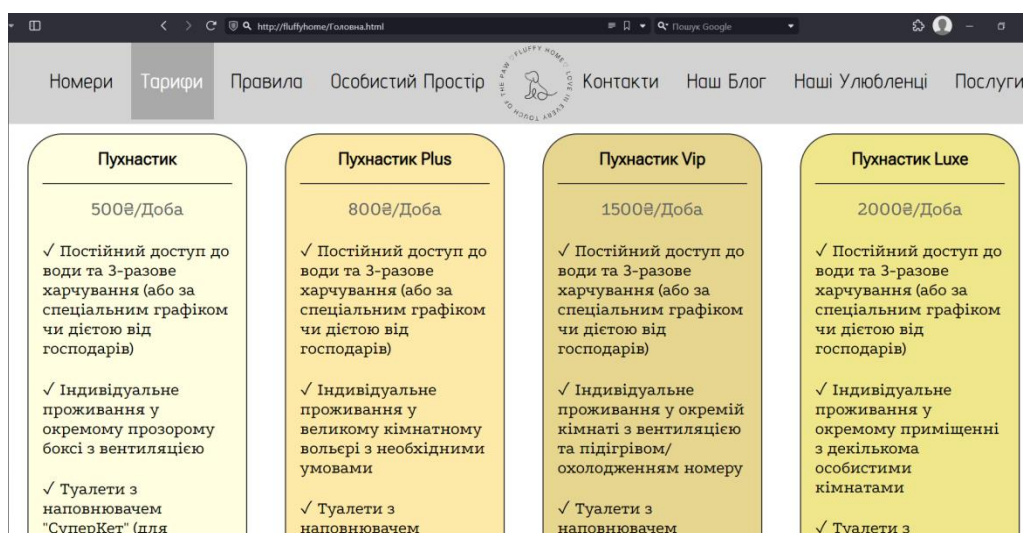


Рисунок 4.10 – Приклад використання інтерактивного меню головної сторінки

Після ознайомлення з інформацією на головній сторінці, користувач може перейти за посиланням на сторінку з послугами (рис. 4.11). На другій сторінці користувач може ознайомитися з необхідними для нього послугами та цінами на них (рис. 4.12).

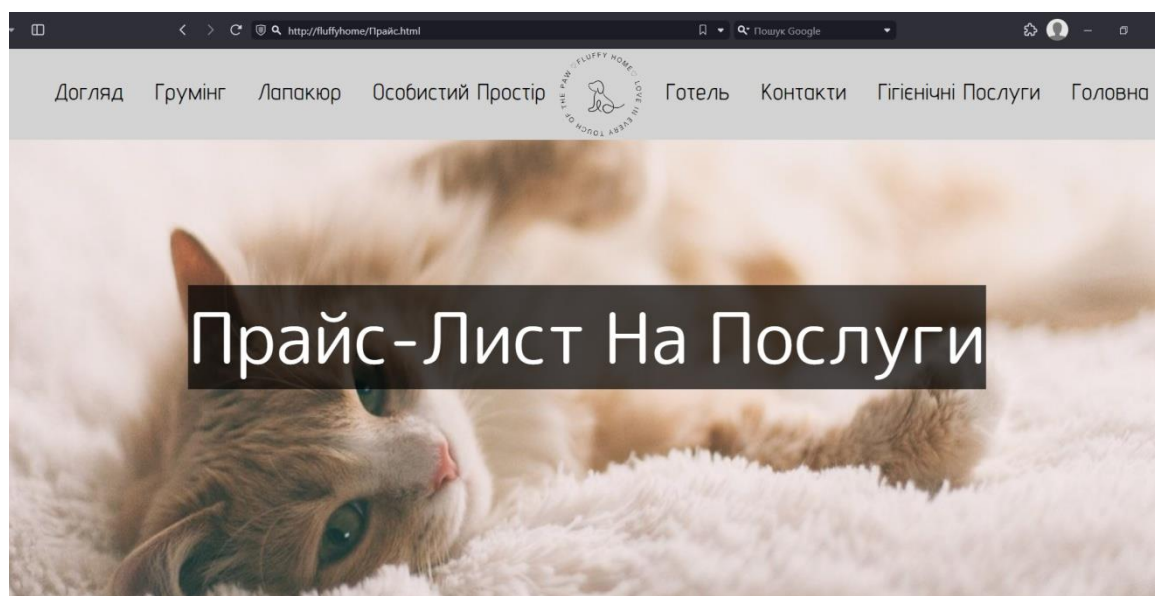
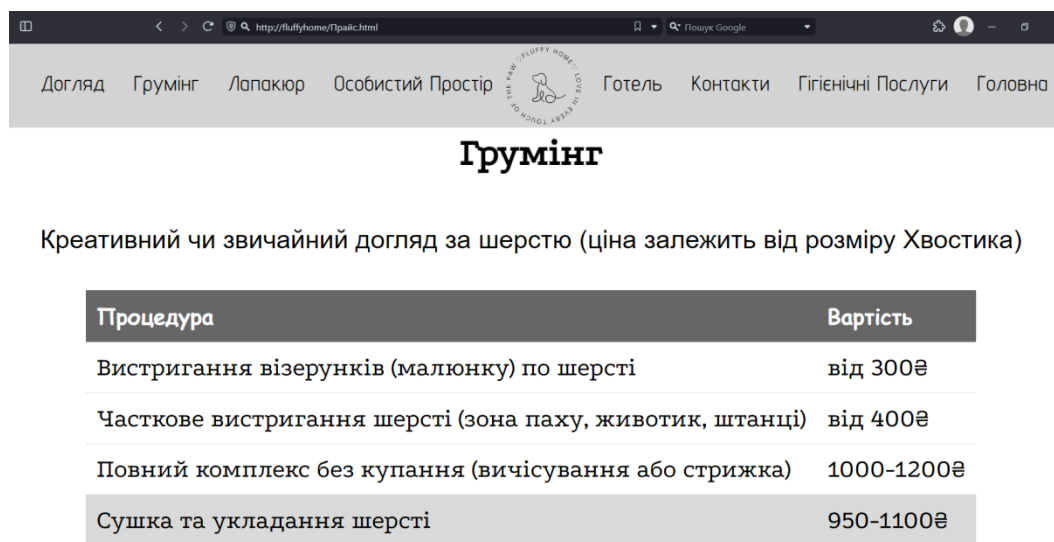


Рисунок 4.12 – Сторінка з послугами веб-сервісу для тварин



Процедура	Вартість
Вистригання візерунків (малюнку) по шерсті	від 300€
Часткове вистригання шерсті (зона паху, животик, штанці)	від 400€
Повний комплекс без купання (вичісування або стрижка)	1000-1200€
Сушка та укладання шерсті	950-1100€

Рисунок 4.13 – Приклад блоку з послугами

Користувач, ознайомившись з усією інформацією про веб-сервіс готелю для тварин, має змогу створити чи увійти у обліковий запис. Для цього він має перейти у меню за посиланням «Особистий Простір» та пройти процедуру реєстрації або авторизації, якщо вже створив акаунт раніше, на веб-сервісі (рис. 4.14).

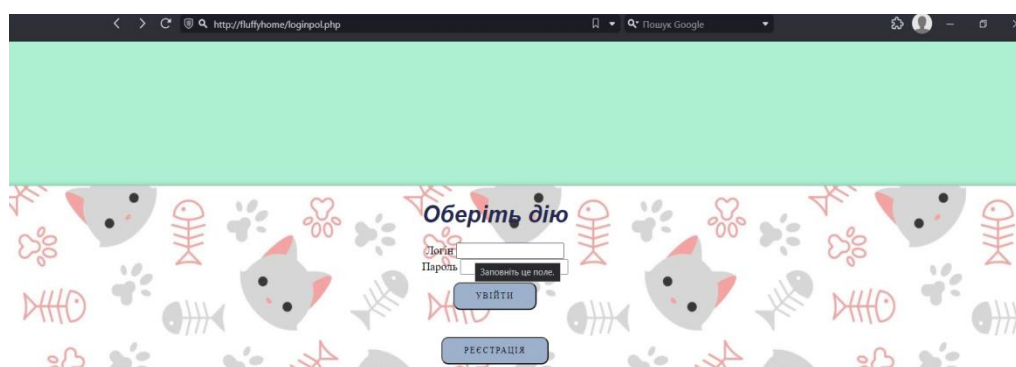


Рисунок 4.14 – Вікно реєстрації та авторизації на веб-сервісі

Якщо користувач обирає реєстрацію, то він має декілька сценаріїв під час цього процесу. Перший сценарій – користувач вводить логін, який вже було

використано іншим користувачем (рис. 4.15). А другий сценарій – успішне проходження процесу та перенаправлення до особистого кабінету (рис. 4.16).

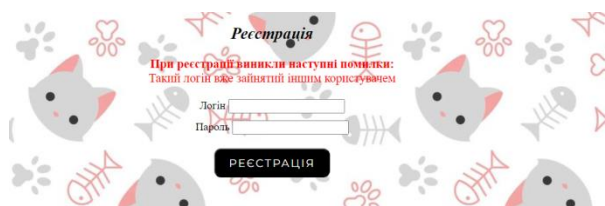


Рисунок 4.15 – Помилка при реєстрації акаунту

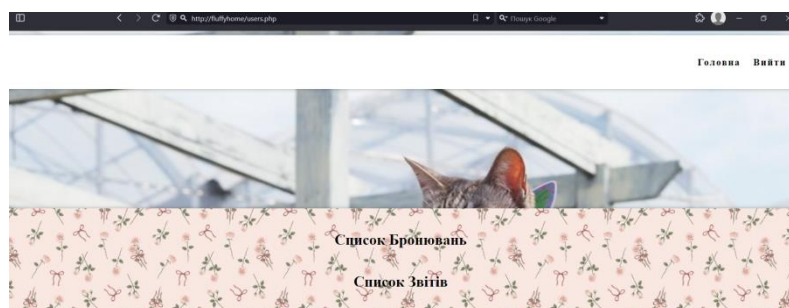


Рисунок 4.16 – Перенаправлення до особистого кабінету користувача

Для оформлення послуг, користувач може повернутися на головну сторінку та перейти на форму бронювання послуг за посиланням «Записатись» (рис. 4.17). У формі бронювання послуг користувач може вказати свої особисті дані, дані тварини та умови бронювання (рис. 4.18).

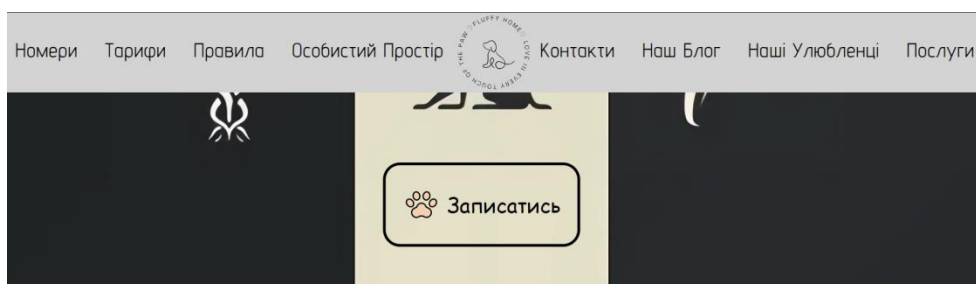


Рисунок 4.17 – Посилання на бронювання послуг

Рисунок 4.18 – Форма бронювання послуг веб-сервісу готелю для тварин

Створивши бронювання, користувач має підтвердити його у спливаючому вікні (рис. 4.19). У разі відмови, користувач може перезаповнити форму, у разі згоди на підтвердження, користувача перекине на сторінку із оповіщенням про успішне бронювання (рис. 4.20).

Рисунок 4.19 – Форма підтвердження бронювання

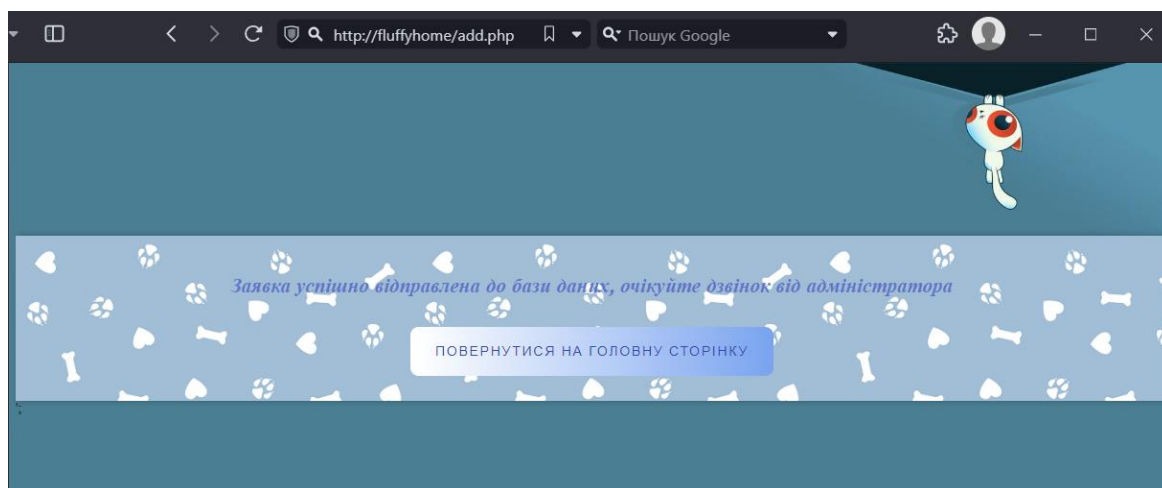


Рисунок 4.20 – Сторінка із оповіщенням про успішне бронювання

Користувач може вийти із особистого кабінету і його перекине назад на форму входу та авторизації. Для авторизації користувач має ввести свій пароль та логін. Для користувача передбачено два сценарії авторизації. В першому сценарії користувач вводить невірні дані і отримує про це повідомлення (рис. 4.21). В другому сценарії користувач успішно авторизується та його перенаправляє до особистого кабінету, де він може переглянути дані бронювання, звіти про стан здоров'я тварини і фото- та відеозвіт (рис. 4.22).

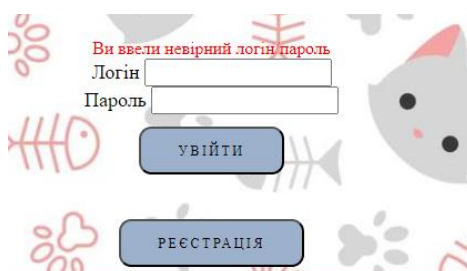


Рисунок 4.21 – Повідомлення про помилку авторизації



Головна Вийти

ID	ПІВ клієнта	Номер телефону клієнта	Пошта	Догляд	Підлячні Процедури	Лапаєкор	Зінтета капту, відвідує і зубного каміня	Тварина	Лікувальні масаж та одорожні SPA-процедури	Грумінг	Тариф	Ідеї тварини	Вік тварини	Стан здоров'я тварини	Дата на бронь
75	Гордасюк Олександр Олександрівна	+380660000000	sashaexample@gmail.com	Середні	Лікувальний-одорожний комплекс для зубів та ясен	Лапаєкор	Середні	Собака	Середні	Сушка та укладання шерсті	Пронастик Ур	Джулія	2 роки	Здоровий	24-12-2023

ID	ID тварини	Температура	Якість сну	Додаткові зауваження	Фото	Відео	Дата звіту
1	75	39.0	97	Джулія була оглянута ветеринаром, стан здоров'я відмінний. Харчується добре та за прописаною дієтою. Активність хороша, сон відмінний, веде себе чинно.	https://t.me/FluffyHome/Julia/photo.jpg	https://t.me/FluffyHome/Julia/video.mp4	2023-12-01

Рисунок 4.22 – Особистий кабінет користувача з особистими даними та даними бронювання

Після перегляду усіх даних, користувач натискає на кнопку «Вийти» та його перекидує на головну сторінку, або за пошуком «Головна» користувач може одразу повернутися на головну сторінку, не виходячи з акаунту (рис. 4.23). Останнім кроком буде закриття сторінки браузера (рис. 4.24).

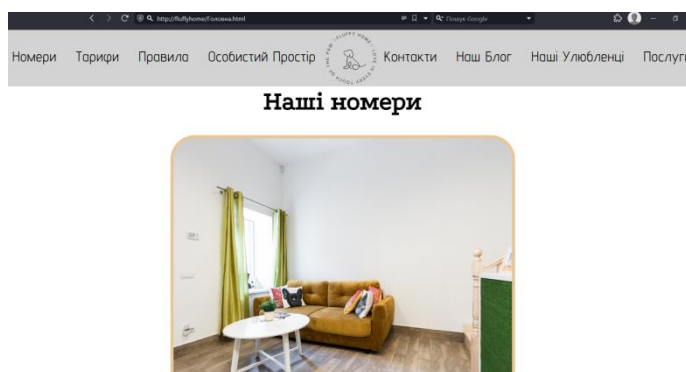


Рисунок 4.23 – Повернення на головну сторінку

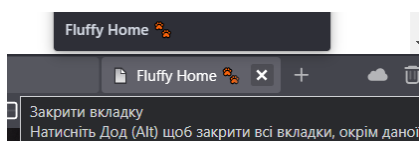


Рисунок 4.24 – Закриття вкладки веб-сервісу

### 4.3 Керівництво адміністратора

Перед початком роботи із веб-сервісом, адміністратор запускає локальний веб-сервер, обрав на панелі задач Open Server Panel та натискає «Запустити» (рис. 4.25).

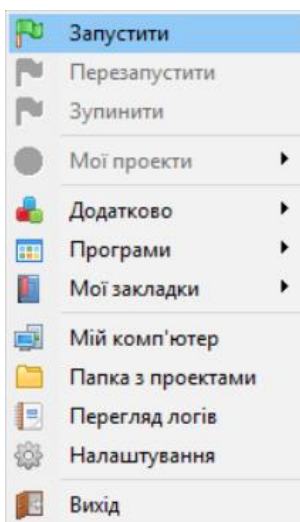


Рисунок 4.25 – Запуск веб-сервера через Open Server Panel

Після успішного запуску веб-сервера, адміністратор обирає проект із веб-серверів у теці «Мої проекти» та запускає веб-сервіс готелю для тварин. (рис. 4.26).

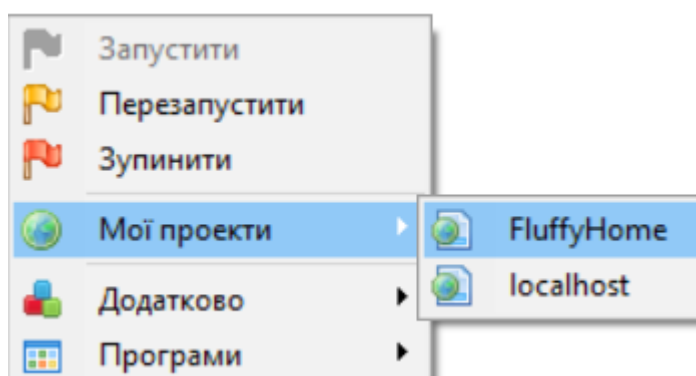


Рисунок 4.26 – Вибір і запуск проекту веб-сервіса готелю для тварин

Після успішного запуску, веб-сервіс відкривається у вікні веб-браузера з повідомленням про успішне підключення до БД. Адміністратор може обрати перехід до головної сторінки чи сторінки авторизації. (рис. 4.27)

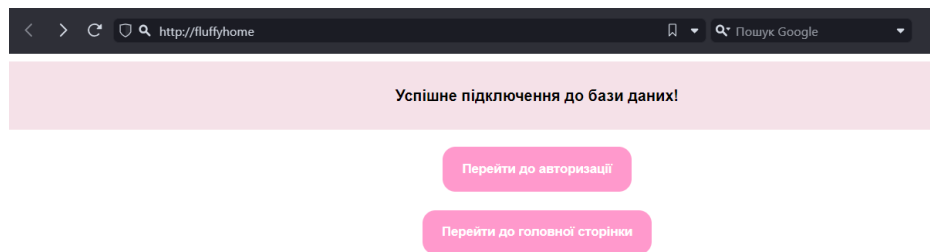


Рисунок 4.27 – Вікно сторінки адміністратора веб-сервіса

Після цього адміністратор переходить на сторінку авторизації його особистих даних (рис. 4.28).

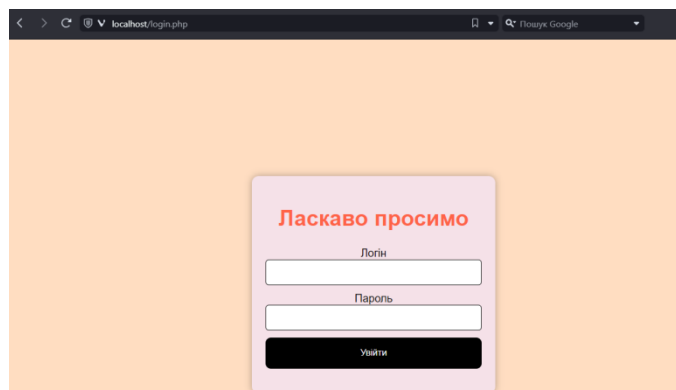


Рисунок 4.28 – Сторінка авторизації адміністратора веб-сервісу

Для введення своїх даних, адміністратор використовує поля для введення логіна та пароля. У разі введення невірних даних, адміністратор отримує від системи помилку входу (рис. 4.29). Якщо адміністратор ввірно вводить дані, то його перекидує на сторінку особистого кабінету (рис. 4.30).

**Ласкаво просимо**

Ви ввели неправильний логін/пароль

Логін

Пароль

Увійти

Рисунок 4.29 – Помилка входу до особистого кабінету адміністратора

ID клієнта	Номер телефона клієнта	Пошта	Догляд	Гігієнічні Процедури	Лазерор	Зняття м'ялту, видалення зубного каменю	Тварина	Лікувальні масаж та оздоровчі SPA-процедури	Грумінг	Тариф	Ім'я тварини	Вік тварини	Стан здоров'я тварини	Дата на бронь	
75	+380560000000	sashaexample@gmail.com	Середні породи собак	Лікувальний-оздоровчий комплекс для зубів та ясен	Лазерор	Середні	Собака	Середні	Сушка та укладання шерсті	Пуніастик Vip	Джулія	2 роки	Здорова	24-12-2023	Переглянути звіт
76	+380509876543	trem@gmail.com	Малі породи котів	Вісчування, видалення когунів	Лазерор/Лаз	Дрібні	Кішка	Дрібні	Чистіщення паронігальних залоз	Пуніастик	Мурка	3 роки	Здорова	13-01-2024	Переглянути звіт
77	+380501234567	ivalo@gmail.com	Малі породи собак	Чистіщення зубів (дента, зубна паста)	Лазерор	Дрібні	Собака	Дрібні	Лікування та підлікування нігтів	Пуніастик Рід	Барон	2 роки	Хворіє	25-12-2023	Переглянути звіт

ID тварини	Температура	Якість сну	Додаткові зауваження	Фото	Відео	Дата звіту

Рисунок 4.30 – Особистий кабінет адміністратора із даними бронювання та звітами користувачів

У адміністратора є можливість переглядати звіти, для цього він має натиснути кнопку «Переглянути звіт», після цього в нього з'явиться список усіх створених звітів (рис. 4.31). Також адміністратор може створювати звіти за допомогою форми після натискання на кнопку «Створити звіт» (рис. 4.32). У разі помилки при оформленні звіту, адміністратор має можливість редагування звіту (рис. 4.33). Після створення та редагування, адміністратор надсилає звіти до користувача за допомогою кнопки «Надіслати звіт» (рис. 4.34).

Список звітів

Створити звіт

ID	ID тварини	Температура	Якість сну	Додаткові зауваження	Фото	Відео	Дата звіту
1	75	39.0	97	Джуля була оглянута ветеринаром, стан здоров'я відмінний. Харчується добре та за прописаною дією. Активність хороша, сон відмінний, веде себе чужою.	<a href="https://t.me/FluffyHome/JuliaPhoto.jpg">https://t.me/FluffyHome/JuliaPhoto.jpg</a>	<a href="https://t.me/FluffyHome/JuliaVideo.mp4">https://t.me/FluffyHome/JuliaVideo.mp4</a>	2023-12-01
2	76	37.1	95	Активна кішка, оглянув ветеринар та змінив дієту. Сон відмінний, апетит середній.	<a href="https://t.me/FluffyHome/MurkaPhoto.jpg">https://t.me/FluffyHome/MurkaPhoto.jpg</a>	<a href="https://t.me/FluffyHome/MurkaVideo.mp4">https://t.me/FluffyHome/MurkaVideo.mp4</a>	2023-12-02
3	77	39.8	78	Оглянув ветеринар, назначив ліки. Тварина почувається вже краще. Сон ще не дуже хороший, але апетит гарний. Поведінка стримана.	<a href="https://t.me/FluffyHome/BaronPhoto.jpg">https://t.me/FluffyHome/BaronPhoto.jpg</a>	<a href="https://t.me/FluffyHome/BaronVideo.mp4">https://t.me/FluffyHome/BaronVideo.mp4</a>	2023-12-03

Рисунок 4.31 – Перегляд звітів

ID тварини:

Температура:

Якість сну (1-100):

Додаткові зауваження:

Фото (посилання):

Відео (посилання):

Дата звіту:

Закрити

Рисунок 4.32 – Вікно створення звіту

ID тварини:

Температура:

Якість сну (1-100):

Додаткові зауваження:

Фото (посилання):

Відео (посилання):

Дата звіту:

Закрити

Рисунок 4.33 – Вікно редагування звіту

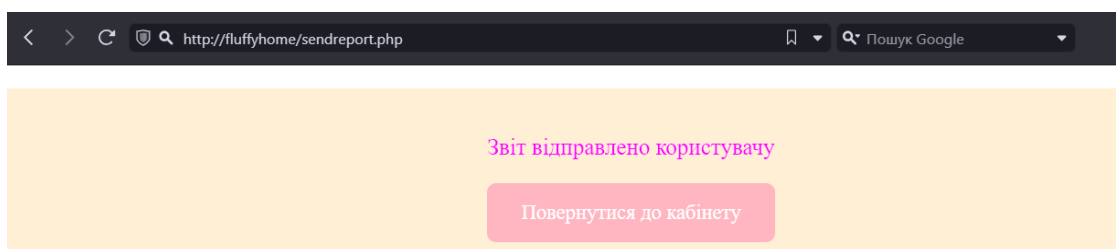


Рисунок 4.34 – Вікно сповіщення про надсилання звіту

Після перегляду усіх даних та звітів, адміністратор може видалити недійсні бронювання (рис. 4.35).

ID	ПІБ клієнта	Номер телефона клієнта	Пошта	Доглад	Післяопер. Процедури	Лапакор	Зняття налітків, відкладення зубного каменю	Тварина	Лікувальні мазки та оздоровчі SPA-процедури	Грумінг	Тариф	Вік тварини	Вік тварини	Стан здоров'я тварини	Дата на бронь		
75	Подвійська Спекія Спекія Спекія Спекія	+380665000000	sashaekample@gmail.com	Середні породи собак	Лікувальний оздоровчий комплекс для зубів та ясен	Лапакор	Середні	Собака	Середні	Стрижка та укладання шерсті	Пухнастик Vip	Дула	2 роки	Здоровий	24-12-2023	Переглянути звіт	Видалити
76	Спекій Спекія Спекія Спекія	+380509876543	brem@gmail.com	Малі породи собак	Вігнуте видалення паразитів	Лапакор+Лак	Дрібні	Кішка	Дрібні	Масаж параназальних залоз	Пухнастик	Мура	3 роки	Здорова	13-01-2024	Переглянути звіт	Видалити

**Список Звітів**

ID	ID тварини	Температура	Якість сну	Додаткові зауваження	Фото	Відео	Дата звіту
1	75	39.0	97	Дула була оглянута ветеринаром, стан здоров'я відмінний. Харчується добре та за прописаною дієтою. Активність хороша, сон відмінний, веде себе чужо.	<a href="https://me.FluffyHome/Julia/photo.jpg">https://me.FluffyHome/Julia/photo.jpg</a>	<a href="https://me.FluffyHome/Julia/video.mp4">https://me.FluffyHome/Julia/video.mp4</a>	2023-12-01
2	76	37.1	95	Активна кішка, оглянула ветеринар та змінив дієту. Сон відмінний, апетит середній.	<a href="https://me.FluffyHome/Maria/photo.jpg">https://me.FluffyHome/Maria/photo.jpg</a>	<a href="https://me.FluffyHome/Maria/video.mp4">https://me.FluffyHome/Maria/video.mp4</a>	2023-12-02

Рисунок 4.35 – Приклад видалення даних

Коли адміністратор завершить усі процедури, він може вийти із кабінету та закрити вікно у браузері (рис. 4.36).

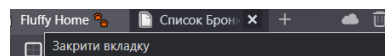


Рисунок 4.36 – Закриття вікна веб-сервіса у браузері

Останнім кроком адміністратора є зупинка сервера. Для цього він повертається до панелі задач, обирає Open Server Panel та натискає «Зупинити» (рис. 4.37).

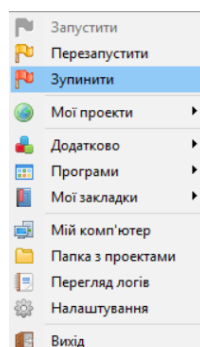


Рисунок 4.37 – Завершення роботи веб-сервера через Open Server Panel

## ВИСНОВКИ

Вдосконалення процесів догляду за тваринами набуває важливості у контексті сучасних технологій, відкриваючи нові можливості для підвищення якості та ефективності турботи. За результатами дослідження та аналізу тенденцій у галузі догляду за тваринами, стало очевидним, що веб-сервіс готелю для тварин із системою контролю їх здоров'я виступає як перспективний інструмент для забезпечення повноцінного догляду за улюбленцями. Сучасні технології моніторингу та системи контролю дозволяють забезпечити точність та надійність відстеження фізіологічних параметрів тварин. Важливість правильного вибору датчиків та обладнання для моніторингу температури, сну, рухової активності визначається їхньою точністю та спроможністю працювати в будь-яких умовах готелю, що свідчить про надійність, точність отриманих даних та перспективне використання на постійній основі.

У рамках цієї дипломної роботи виконано глибокий аналіз галузі сфери веб-сервісу готелю для тварин та виявлено вплив сучасних технологій на цей сектор. Дослідження вказує на те, що відзначається наявність новаторських технологічних рішень, спрямованих на поліпшення та вдосконалення послуг з догляду за домашніми улюбленцями та контролем стану їх здоров'я.

У ході виконання даної роботи був проведений детальний аналіз процесу створення веб-сервісу для готелю тварин із системою контролю їх здоров'я, з особливим акцентом на розробку функціональності та взаємодії з користувачами. Особлива увага була приділена реалізації отримання даних від системи контролю здоров'я тварин, що є ключовою функцією сервісу. У фінальному продукті було реалізовано велику кількість функціональних можливостей, що дозволило створити веб-сервіс, який надає користувачам унікальний та зручний інструмент для догляду за їхніми тваринами. Взаємодія з користувачами та можливість отримувати звіти про здоров'я тварин в реальному часі стали важливими компонентами цього проекту, підкреслюючи його

значущість у контексті покращення процесів догляду за домашніми улюбленцями.

Таким чином розроблений веб-сервіс готелю для тварин із системою контролю їх здоров'я став важливим кроком у поліпшенні процесів догляду за домашніми улюбленцями. Використання мови програмування PHP для розробки веб-сервісу виявилось оптимальним вибором, забезпечуючи ефективну роботу та швидкість розробки. Однак розроблений веб-сервіс не лише відповідає поточним вимогам, а й має потенціал для майбутнього розширення функціоналу. Забезпечена легкість масштабування та гнучкість у впровадженні нових функцій робить цей сервіс перспективним інструментом для подальшого вдосконалення та адаптації до зростаючих потреб користувачів та ветеринарних спеціалістів.

Результати роботи доповідалися на Науковій конференції молодих вчених XXII ОДЕКУ 30 травня 2023р.



## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Проектування веб-сервісів. URL: <https://uk.wikipedia.org/wiki/Веб-служба> (дата звернення 02.11.2023)
2. Збільшення популярності зооготелей. URL: <https://hospitalityinsights.ehl.edu/pet-hotels> (дата звернення 02.11.2023)
3. Сучасні технології відстеження стану тварин URL: <https://www.cogniteq.com/blog/advanced-technologies-pets-health-monitoring-system> (дата звернення 02.11.2023)
4. Цифрові інновації у сфері тваринного добробуту URL: <https://www.geoforce.com/types-of-tracking-technology/> (дата звернення: 03.11.2023)
5. Розвиток охорони здоров'я тварин URL: [https://www.cell.com/trends/ecology-evolution/fulltext/S0169-5347\(23\)00087-3](https://www.cell.com/trends/ecology-evolution/fulltext/S0169-5347(23)00087-3) (дата звернення: 03.11.2023)
6. Забезпечення програмними компонентами готелей для тварин URL: <https://easyweek.io/best-pet-hotel-software.html> (дата звернення: 05.11.2023)
7. Плавнування структури бази даних URL: [https://uk.wikipedia.org/wiki/База\\_даних](https://uk.wikipedia.org/wiki/База_даних) (дата звернення: 05.11.2023)
8. Визначення технічних специфікацій: URL: [https://uk.wikipedia.org/wiki/Технічні\\_умови](https://uk.wikipedia.org/wiki/Технічні_умови) (дата звернення: 08.11.2023)
9. Складові системи планування проекту URL: <https://buklib.net/books/23851/> (дата звернення: 09.11.2023)
10. Розроблення прототипу URL: [https://uk.wikipedia.org/wiki/Прототип\\_\(шаблон\\_проєктування\)](https://uk.wikipedia.org/wiki/Прототип_(шаблон_проєктування)) (дата звернення: 09.11.2023)
11. Пошукова оптимізація URL: [https://uk.wikipedia.org/wiki/Оптимізація\\_для\\_пошукових\\_систем](https://uk.wikipedia.org/wiki/Оптимізація_для_пошукових_систем) (дата звернення: 09.11.2023)
12. Методологія IDEF0 URL: <https://uk.wikipedia.org/wiki/IDEF0> (дата звернення: 10.11.2023)
13. Діаграми UML для моделювання процесів URL: [https://uk.wikipedia.org/wiki/Unified\\_Modeling\\_Language](https://uk.wikipedia.org/wiki/Unified_Modeling_Language) (дата звернення: 10.11.2023)

14. Основи побудови діаграми прецедентів URL: [https://uk.wikipedia.org/wiki/Діаграма\\_прецедентів](https://uk.wikipedia.org/wiki/Діаграма_прецедентів) (дата звернення: 11.11.2023)
15. Діаграма станів UML URL: [https://uk.wikipedia.org/wiki/Діаграма\\_станів\\_\(UML\)](https://uk.wikipedia.org/wiki/Діаграма_станів_(UML)) (дата звернення: 11.11.2023)
16. Принципи побудови діаграми діяльності URL: [https://uk.wikipedia.org/wiki/Діаграма\\_діяльності](https://uk.wikipedia.org/wiki/Діаграма_діяльності) (дата звернення: 12.11.2023)
17. Діаграма послідовності: [https://uk.wikipedia.org/wiki/Діаграма\\_послідовності](https://uk.wikipedia.org/wiki/Діаграма_послідовності) (дата звернення: 13.11.2023)
18. Особливості СУБД URL: <https://highload.today/uk/subd-yaki-buvayut-yak-vibrati/> (дата звернення: 14.11.2023)
19. Створення інтерфейсу користувача URL: [https://uk.wikipedia.org/wiki/Дизайн\\_інтерфейсу\\_користувача](https://uk.wikipedia.org/wiki/Дизайн_інтерфейсу_користувача) (дата звернення: 16.11.2023)
20. Моделі і методи проектування URL: [https://elearning.sumdu.edu.ua/free\\_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/200847/index.html](https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/200847/index.html) (дата звернення: 17.11.2023)
21. Забезпечення інформаційної безпеки URL: [http://www.rusnauka.com/35\\_OINBG\\_2010/Informatica/76346.doc.htm](http://www.rusnauka.com/35_OINBG_2010/Informatica/76346.doc.htm) (дата звернення: 18.11.2023)
22. Вибір середовища розробки URL: <https://loft.sh/blog/the-definitive-guide-to-development-environments/> (дата звернення: 19.11.2023)
23. Офіційна документація PHP URL: <https://www.php.net/> (дата звернення: 21.11.2023)
24. Офіційна документація Open Server Panel URL: <https://ospanel.io/> (дата звернення: 22.11.2023)

## ДОДАТОК А Вихідний код програми

```

<?php
// Підключення до бази даних
require_once 'connect.php';
?>
<!DOCTYPE html >
<html>
<head>
  <meta charset="UTF-8">
  <title>Список Бронювань</title>
</head>
<body>
  <style type="text/css">
/* Імпорт шрифтів */
  @import
url('https://fonts.googleapis.com/css?family=Kaushan+Script|Libre+Baskerville');
/* Стилізація для тіла сторінки */
body {
  background:
url(https://static.apostrophe.ua/uploads/image/19d901addd70a7e1fd0b812916dd7e87.jpg);
  background-size: 100%;
}
/* Стилізація тексту чорного кольору */
.blacktext {
  color: black;
  font-family: 'Libre Baskerville';
}
/* Стилізація тексту білого кольором */
.whitetext {
  color: white;
  font-family: 'Libre Baskerville';
}
/* Верхнє меню навігації */
.top-menu {
  background: white;
  box-shadow: 3px 0 7px rgba(0,0,0,.3);
  padding: 20px;
}
/* Очистка вирівнювання елементів після верхнього меню */
.top-menu:after {
  content: "";
  display: table;
  clear: both;
}
/* Стилізація логотипу на панелі керування*/
.navbar-logo {
display: inline-block;
}
/* Основне меню навігації */
.menu-main {
  list-style: none;
  margin: 0;
  padding: 0;
  float: right;
}
/* Стилізація кожного пункту меню в головному меню */
.menu-main li {
display: inline-block;
}
/* Стилізація посилань у меню */
.menu-main a {

```

```

text-decoration: none;
display: block;
position: relative;
line-height: 61px;
padding-left: 20px;
font-size: 18px;
letter-spacing: 2px;
font-family: 'Libre Baskerville';
font-weight: bold;
color: black;
transition: .3s linear;
}
/* Стилiзацiя псевдо-елемента (стрiлки) при наведеннi */
.menu-main a:before {
content: "";
width: 9px;
height: 9px;
background: black;
position: absolute;
left: 50%;
transform: rotate(45deg) translateX(6.5px);
opacity: 0;
transition: .3s linear;
}
/* Ефект при наведеннi для стрiлки *
.menu-main a:hover:before {
opacity: 1;
}
/* Адаптацiя для респонсивного дизайну головного меню */
@media (max-width: 100%) {
.menu-main {
float: none;
padding-top: 0px;
}
.top-menu {
text-align: center;
padding: 20px 0 0 0;
}
.menu-main a {
padding: 0 10px;}
.menu-main a:before {
transform: rotate(45deg) translateX(-6px);
}
}
@media (max-width: 100%) {
.menu-main li {display: block;}
}
/* Стилiзацiя головного блоку контенту */
.block {
margin-top: 15%;
background: white;
box-shadow: 3px 0 7px rgba(0,0,0,.3);
padding: 20px;
background:
url(https://i.pinimg.com/236x/10/77/aa/1077aa3afd1bcb52244973c6c617ec43.jpg);
}
/* Стилiзацiя головного логотипу */
.logo {
font-family: 'Libre Baskerville';
font-style: italic;
font-weight: bold;
text-align: center;
font-size: 150%;
}
/* Контейнер для розмiщення контенту по центру */

```

```

.flex {
  display: flex;
  justify-content: center;
}
/* Стилiзацiя першого додаткового блоку */
.ad1 {
  height: 550px;
  width: 50%;
  overflow: auto;
  margin-right: 2%;
}
/* Стилiзацiя градiєнтної кнопки */
.gradient-button {
  text-decoration: none;
  display: inline-block;
  color: white;
  padding: 20px 30px;
  margin: 10px 20px;
  border-radius: 10px;
  font-family: 'Montserrat', sans-serif;
  text-transform: uppercase;
  letter-spacing: 2px;
  background-image: linear-gradient(to right, black 0%, black 51%, #9EEFE1 100%);
  background-size: 200% auto;
  transition: .5s;
}
/* Стилiзацiя заголовкiв i комiрок таблицi */
th, td {
  padding: 10px;
}
th {
  background: black;
  color: white;
  font-family: 'Arimo', sans-serif;
}
td {
  background: #C08081;
  font-family: 'Arimo', sans-serif;
}
/* Стилiзацiя модального (виринаючого) вiкна */
.modal {
  display: none;
  position: fixed;
  z-index: 1;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(0, 0, 0, 0.7);
}
/* Стилiзацiя контейнера модального вiкна */
.modal-content {
  background-color: #fff;
  margin: 15% auto;
  padding: 20px;
  width: 70%;
}
</style>
/* Навiгацiйне меню */
<nav class="top-menu">
  <ul class="menu-main">
    <li><a href="Головна.html" >Головна</a>
    </li>
    <li><a href="#" onclick="history.back();" >Вийти</a>
    </li>
  </ul>
</nav>

```

```

    </ul>
</nav>
/* Головний блок контенту */
<nav class="block">
  <body>
/* Таблиця для відображення інформації про бронювання */
  <table>
    /* Заголовки таблиці */
    <tr>
      <th>ID</th>
      <th>ПІБ клієнта</th>
      <th>Номер телефона клієнта</th>
      <th>Пошта</th>
      <th>Догляд</th>
      <th>Гігієнічні Процедури</th>
      <th>Лапакюр</th>
      <th>Зняття наліту, відкладень і зубного каміння</th>
      <th>Тварина</th>
      <th>Лікувальні маски та оздоровчі SPA-процедури</th>
      <th>Грумінг</th>
      <th>Тариф</th>
      <th>Ім'я тварини</th>
      <th>Вік тварини</th>
      <th>Стан здоров'я тварини</th>
      <th>Дата на бронь</th>
    </tr>
/* Отримання даних з таблиці */
    <?php
      $animals = mysqli_query($connect, "SELECT * FROM `animal` ");
      $animals = mysqli_fetch_all($animals);
      foreach ($animals as $animal) {
        ?>
          <tr>
/* Відображення даних для кожного стовпчика в рядку */
            <td><?= $animal[0] ?></td>
            <td><?= $animal[1] ?></td>
            <td><?= $animal[2] ?></td>
            <td><?= $animal[3] ?></td>
            <td><?= $animal[5] ?></td>
            <td><?= $animal[9] ?></td>
            <td><?= $animal[8] ?></td>
            <td><?= $animal[11] ?></td>
            <td><?= $animal[4] ?></td>
            <td><?= $animal[10] ?></td>
            <td><?= $animal[12] ?></td>
            <td><?= $animal[7] ?></td>
            <td><?= $animal[13] ?></td>
            <td><?= $animal[14] ?></td>
            <td><?= $animal[15] ?></td>
            <td><?= $animal[6] ?></td>
          <td>
/* Форма для перегляду деталей звіту */
            <form method="post" action="details.php">
              <input type="hidden" name="report_id" value="<?= $animal[0] ?>">
              <input type="submit" value="Переглянути звіт">
            </form>
          </td>
          <td>
/* Форма для видалення звіту */
            <form method="post" action="remove.php">
              <input type="hidden" name="report_id" value="<?= $animal[0] ?>">
              <input type="submit" value="Видалити">
            </form>
          </td>
        </tr>

```

```

        <?php
        }
        ?>
/* Форма для відправлення звіту */
        <form method="post" action="sendreport.php">
            <input type="hidden" name="report_id" value="<?= $animal[0] ?>">
            <input type="submit" style="width: 200px;
            height: 40px;
            " value="Надіслати звіт">
            <input type="submit" style="width: 200px;
            height: 40px;
            " value="Редагування">
                </form>
            </td>
        </tr>
    </table>
/* Заголовок для списку звітів */
<h2>Список Звітів</h2>
    <table>
        <tr>
            <th>ID</th>
            <th>ID тварини</th>
            <th>Температура</th>
            <th>Якість сну</th>
            <th>Додаткові зауваження</th>
            <th>Фото</th>
            <th>Відео</th>
            <th>Дата звіту</th>
        </tr>
    <?php
/* Отримання даних з таблиці 'reports' */
$reports = mysqli_query($connect, "SELECT * FROM `reports`;");
$reports = mysqli_fetch_all($reports);
    foreach ($reports as $report) {
        ?>
        <tr>
            <td><?= $report[0] ?></td>
            <td><?= $report[1] ?></td>
            <td><?= $report[2] ?></td>
            <td><?= $report[3] ?></td>
            <td><?= $report[4] ?></td>
            <td style="color: darkmagenta;"><?= $report[5] ?></td>
            <td style="color: darkmagenta;"><?= $report[6] ?></td>
            <td><?= $report[7] ?></td>
        </tr>
    <?php
    }
    ?>
    <button id="openModalBtn" style="width: 200px;height: 40px;">Створити
звіт</button>
</table>
/* Кнопка для відкриття модального вікна для створення звіту */
<div id="myModal" class="modal">
    <div class="modal-content">
        <span id="closeModalBtn" style="float: right; cursor:
pointer;">Закрити</span>
        <form action="report.php" method="post" enctype="multipart/form-data">
            <label for="animal_id">ID тварини:</label>
            <input type="text" name="animal_id" required><br><br>

            <label for="temperature">Температура:</label>
            <input type="text" name="temperature" pattern="\d+(\.\d{1})?"
placeholder="xx.x" required><br><br>
            <label for="sleep_quality">Якість сну (1-100):</label>

```

```

        <input type="number" name="sleep_quality" min="1" max="100"
required><br><br>
        <label for="additional_notes">Додаткові зауваження:</label><br>
        <textarea name="additional_notes" rows="4" cols="50"></textarea><br><br>
        <label for="photo">Фото (посилання):</label>
        <input type="text" name="photo"><br><br>
        <label for="video">Відео (посилання):</label>
        <input type="text" name="video"><br><br>
        <label for="report_date">Дата звіту:</label>
        <input type="date" name="report_date" required><br><br>
        <input type="submit" value="Зберегти">
    </form>
</div>
</div>
<script>
/* Відкриття модального вікна при натисканні на кнопку */
document.getElementById("openModalBtn").onclick = function() {
    document.getElementById("myModal").style.display = "block";
}
/* Закриття модального вікна по кнопці чи по кліку поза вікном */
document.getElementById("closeModalBtn").onclick = function() {
    document.getElementById("myModal").style.display = "none";
}

window.onclick = function(event) {
    var modal = document.getElementById("myModal");
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
</script>
</body>
</html>

```