

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

_____ Євгенія СУЛЕМА

«___» _____ 2025 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інженерія програмного
забезпечення мультимедійних та інформаційно-пошукових систем»**

спеціальності 121 Інженерія програмного забезпечення

**на тему: «Вебзастосунок для підтримки спільноти власників домашніх
тварин та пошуку і публікації оголошень»**

Виконала:

студентка IV курсу, групи КП-11

Небесна Анна Андріївна _____

Керівник:

доцент кафедри ПЗКС, к.т.н., доцент,

Саяпіна Інна Олександрівна _____

Консультант з нормоконтролю:

доцент кафедри ПЗКС, к.т.н., доцент,

Онай Микола Володимирович _____

Рецензент:

доцент кафедри СПСКС, д-р філ.,

Сулема Ольга Костянтинівна _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студентка _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення
мультимедійних та інформаційно-пошукових систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Євгенія СУЛЕМА

«__» _____ 2024 р.

ЗАВДАННЯ

на дипломний проєкт студентці

Небесній Анні Андріївні

1. Тема проєкту «Вебзастосунок для підтримки спільноти власників домашніх тварин та пошуку і публікації оголошень», керівник проєкту Саяпіна Інна Олександрівна, доцент кафедри ПЗКС, к.т.н., доцент, затверджені наказом по університету №1808-С від «29» травня 2025 р.
2. Термін подання студентом проєкту «13» червня 2025 р.
3. Вихідні дані до проєкту: див. Технічне завдання.
4. Зміст пояснювальної записки:
 - огляд та аналіз існуючих рішень;
 - обґрунтування вибору засобів реалізації;
 - розроблення вебсервісу «PetPlace»;
 - аналіз розробленого вебсервісу.
5. Перелік обов'язкового графічного матеріалу:
 - структура бази даних. ER-діаграма (креслення);
 - варіанти використання. UML-діаграма (креслення);
 - дерево проблем вебзастосунку (плакат);
 - узагальнена архітектура системи (плакат).

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Онай М.В., доцент		

7. Дата видачі завдання «31» жовтня 2024 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	10.11.2024	
2.	Розроблення та узгодження технічного завдання	21.11.2024	
3.	Розроблення структури вебзастосунку	19.12.2024	
4.	Підготовка матеріалів першого розділу дипломного проєкту	31.12.2024	
5.	Розроблення дизайну сторінок та графічних елементів	01.02.2025	
6.	Підготовка матеріалів другого розділу дипломного проєкту	22.02.2025	
7.	Програмна реалізація вебзастосунку	15.03.2025	
8.	Тестування вебзастосунку	18.03.2025	
9.	Підготовка матеріалів третього розділу дипломного проєкту	28.03.2025	
10.	Підготовка матеріалів четвертого розділу дипломного проєкту	13.04.2025	
11.	Підготовка графічної частини дипломного проєкту	22.04.2025	
12.	Оформлення документації дипломного проєкту	25.05.2025	

Студентка

Анна НЕБЕСНА

Керівник проєкту

Інна САЯПІНА

АНОТАЦІЯ

Даний дипломний проєкт присвячений розробці вебзастосунку «PetPlace» – спеціалізованої багатофункціональної платформи для власників домашніх тварин, тих, хто шукає собі улюбленця, притулків, розплідників тощо. Метою роботи є створення єдиного онлайн-простору, що об'єднує функціональність тематичної дошки оголошень, онлайн-магазину, бази знань та спільноти любителів тварин. Застосунок покликаний спростити пошук, продаж або віддачу тварин, забезпечити доступ до товарів та послуг для них, а також надати перевірену інформацію щодо догляду та можливість обміну досвідом.

У рамках проєкту спроектовано та реалізовано багаторівневу клієнт-серверну архітектуру. Клієнтська частина розроблена за допомогою Angular, що забезпечує динамічний односторінковий інтерфейс. Серверна частина реалізована на ASP.NET Core для високої продуктивності та безпеки, з обміном даними через REST API та SignalR для взаємодії в реальному часі. Для роботи з базою даних використано Entity Framework Core та Microsoft SQL Server, розгорнутий у Docker. Хостинг вебзастосунку здійснюється у хмарному середовищі Microsoft Azure.

Результатом роботи є повнофункціональна платформа, що дозволяє користувачам ефективно знаходити тварин завдяки розширеним фільтрам, створювати профілі улюбленців та використовувати їх для автоматизованого пошуку, знаходити супутні товари та послуги, а також отримувати відповіді на питання на форумі чи у вбудованій базі знань. Особливу увагу приділено зручності інтерфейсу та точності пошуку. Практичне значення проєкту полягає у створенні централізованого та зручного рішення для потреб широкої аудиторії власників домашніх тварин, що має потенціал для подальшого розширення функціональності та інтеграції з ветеринарними клініками, притулками і сервісами доставки.

ABSTRACT

This diploma project is dedicated to the development of the PetPlace web application – a specialized multifunctional platform for pet owners, those looking for a pet, shelters, kennels, etc. The goal of the project is to create a single online space that combines the functionality of a thematic bulletin board, online store, knowledge base, and community of animal lovers. The application is designed to simplify the search, sale, or adoption of animals, provide access to goods and services for them, as well as provide verified information on care and the opportunity to exchange experiences.

As part of the project, a multi-level client-server architecture was designed and implemented. The client side was developed using Angular, which provides a dynamic single-page interface. The server part is implemented on ASP.NET Core for high performance and security, with data exchange via REST API and SignalR for real-time interaction. Entity Framework Core and Microsoft SQL Server, deployed in Docker, are used to work with the database. The web application is hosted in the Microsoft Azure cloud environment.

The result is a fully functional platform that allows users to efficiently find animals using advanced filters, create pet profiles and use them for automated searches, find related products and services, and get answers to questions on the forum or in the built-in knowledge base. Particular attention was paid to the convenience of the interface and the accuracy of the search. The practical significance of the project lies in the creation of a centralized and convenient solution for the needs of a wide audience of pet owners, which has the potential for further expansion of functionality and integration with veterinary clinics, shelters, and delivery services.

ДП.045440-01-90 Вебзастосунок для підтримки спільноти власників домашніх тварин та пошуку і публікації оголошень. Відомість проєкту

[illegible]

[illegible]

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Євгенія СУЛЕМА

“ ____ ” _____ 2024 р.

**ВЕБЗАСТОСУНОК ДЛЯ ПІДТРИМКИ СПІЛЬНОТИ ВЛАСНИКІВ
ДОМАШНІХ ТВАРИН ТА ПОШУКУ І ПУБЛІКАЦІЇ ОГОЛОШЕНЬ**

Технічне завдання

ДП.045440-02-91

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Інна САЯПІНА

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Анна НЕБЕСНА

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ	3
2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ.....	3
3. ПРИЗНАЧЕННЯ РОЗРОБКИ	3
4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ	3
5. ВИМОГИ ДО ПРОЄКТНОЇ ДОКУМЕНТАЦІЇ.....	4
6. ЕТАПИ ПРОЄКТУВАННЯ	4
7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ	5

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: вебзастосунок для підтримки спільноти власників домашніх тварин та пошуку і публікації оголошень.

Галузь застосування: інформаційні технології.

2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ

Підставою для розроблення є завдання на дипломне проєктування, затверджене кафедрою програмного забезпечення комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім. Ігоря Сікорського).

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для використання власниками домашніх тварин, продавцями зоотоварів і послуг, притулками, заводчиками, а також фахівцями у сфері догляду за тваринами. Платформа дозволяє публікувати та шукати оголошення, користуватися індивідуальними фільтрами для різних категорій тварин, отримувати персоналізований підбір товарів для свого улюбленця, брати участь у тематичних дискусіях на форумі та отримувати корисну інформацію зі стрічки статей.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Вебзастосунок повинен забезпечувати такі основні функції:

- 1) наявність широкої категоризації за типами тварин;
- 2) можливість розміщувати оголошення з динамічними характеристиками для кожної категорії;
- 3) можливість знайти супутні товари та послуги;

- 4) наявність розширених та унікальних фільтрів окремо для кожної категорії оголошень;
- 5) цифровий профіль тварини в особистому кабінеті користувача;
- 6) наявність стрічки корисних матеріалів;
- 7) наявність форуму;
- 8) можливість обміну повідомленнями.

5. ВИМОГИ ДО ПРОЄКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання проєкту повинна бути розроблена наступна документація:

- 1) пояснювальна записка;
- 2) програма та методика тестування;
- 3) керівництво користувача;
- 4) креслення:
 - «Структура бази даних. ER-діаграма»;
 - «Архітектура застосунку. Структурна схема».

6. ЕТАПИ ПРОЄКТУВАННЯ

Вивчення літератури за тематикою роботи.....	13.11.2024
Розроблення та узгодження технічного завдання	24.11.2024
Розроблення структури вебзастосунку	17.12.2025
Розроблення дизайну сторінок та графічних елементів	01.02.2025
Програмна реалізація вебзастосунку	16.03.2025
Тестування вебзастосунку	20.03.2025
Підготовка матеріалів текстової частини проєкту	20.04.2025
Підготовка матеріалів графічної частини проєкту	10.05.2025
Оформлення технічної документації проєкту.....	24.05.2025

7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ

Тестування розробленого програмного продукту виконується відповідно до «Програми та методики тестування».

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Євгенія СУЛЕМА

“___” _____ 2025 р.

**ВЕБЗАСТОСУНОК ДЛЯ ПІДТРИМКИ СПІЛЬНОТИ ВЛАСНИКІВ
ДОМАШНІХ ТВАРИН ТА ПОШУКУ І ПУБЛІКАЦІЇ ОГОЛОШЕНЬ**

Пояснювальна записка

ДП.045440-03-81

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Інна САЯПІНА

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Анна НЕБЕСНА

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	3
ВСТУП.....	4
1. ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	5
1.1. Загальні положення та аналіз предметної області	5
1.2. Аналіз існуючих рішень	7
1.3. Результати проведеного аналізу	16
1.4. Висновки до розділу 1	18
2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ.....	19
2.1. Вибір технологій для розроблення серверної частини.....	19
2.2. Обґрунтування вибору системи керування базами даних.....	26
2.3. Вибір технологій для розроблення клієнтської частини	30
2.4. Висновки до розділу 2	34
3. РОЗРОБЛЕННЯ ВЕБСЕРВІСУ «PETPLACE»	35
3.1. Формулювання вимог до розроблюваної системи.....	35
3.2. Структура бази даних	42
3.3. Архітектура вебзастосунку	49
3.4. Висновки до розділу 3	50
4. АНАЛІЗ РОЗРОБЛЕНОГО ВЕБСЕРВІСУ	52
4.1. Опис реалізованого користувацького інтерфейсу	52
4.2. Тестування розробленого вебзастосунку	57
4.3. Рекомендації щодо подальшого вдосконалення	61
4.4. Висновки до розділу 4	62
ВИСНОВКИ	64
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ	66
ДОДАТКИ	69

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

ORM (Object-Relational Mapping) – спосіб роботи з базою даних через об'єкти мови програмування без прямого використання SQL.

Стек – набір технологій (мови, фреймворки, бібліотеки), що використовуються у проєкті.

Маркетплейс – платформа для розміщення та пошуку товарів або послуг різними користувачами.

Фреймворк – готовий каркас для розробки застосунків із заздалегідь визначеною структурою.

Бандл – зібраний у єдиний файл набір ресурсів (JavaScript, CSS) для швидшого завантаження сайту.

SPA (Single Page Application) – застосунок, що працює в межах однієї сторінки без повного перезавантаження.

API (Application Programming Interface) – це набір правил і протоколів, за якими окремі частини програм або зовнішні системи можуть взаємодіяти між собою.

JWT (JSON Web Token) – формат токена для безпечної авторизації та передачі даних між клієнтом і сервером.

Бібліотека – набір готових функцій, які спрощують розробку, не нав'язуючи структури.

EAV (Entity-Attribute-Value) – модель зберігання даних, що дозволяє мати різні атрибути для різних об'єктів.

CLI (Command Line Interface) – інтерфейс для роботи з програмами через текстові команди в терміналі.

РСКБД (реляційна система керування базами даних) – це система, яка забезпечує зберігання, обробку та керування даними у вигляді таблиць з підтримкою зв'язків і цілісності.

ВСТУП

Історія одомашнення тварин розпочалася ще задовго до появи розвинених суспільств і сучасних технологій – за різними даними близько 35 000 років [1] до теперішнього часу, що говорить про тісний зв'язок з ними та їх важливу роль у житті людини. Сьогодні за статистикою у більш ніж третини домогосподарств світу проживає хоча б одна собака, а майже у чверті – кіт [2], з чого випливає, що домашні тварини стали не лише вірними супутниками, а й невід'ємною частиною повсякденного життя та справжніми членами сім'ї, піклування про яких перетворилось на цілу індустрію.

Останні десятиліття кількість бажаючих стати власниками хатніх улюбленців постійно зростає, що прямо пропорційно впливає на попит на товари та послуги, пов'язаних з їх утриманням. Значний «бум» у статистиці викликала пандемія COVID-19 [3], коли люди були змушені залишатися вдома, що призводило до бажання урізноманітнити рутину та позбутися самотності. Однак зі збільшенням числа власників не стоїть на місці й кількість покинутих тварин в притулках. Особливо ця проблема загострилась в Україні після повномасштабного вторгнення, коли без домівки опинилося у середньому на 60% більше тварин, ніж це траплялося до війни [4], що значно перевищує можливості притулків.

Існуючі сервіси оголошень чи зоомагазини не в силі комплексно вирішувати питання зручного пошуку тварин, супутніх товарів і послуг, співпраці з притулками та, тим паче, формування активної спільноти любителів та власників домашніх улюбленців. Аналізуючи усі вищезазначені виклики, з якими стикаються потенційні власники чи ті, що вже стали ними, притулки та навіть самі тварини, можна дійти висновку, що існує потреба в спеціалізованій онлайн-платформі, що б об'єднала ці різні групи споживачів та зацікавлених сторін.

1. ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

У першому розділі буде проаналізовано проблеми, з якими стикаються люди під час пошуку домашніх тварин, товарів та послуг для них, а також розглянемо сучасні програмні рішення, що частково покривають ці потреби. Особливу увагу буде приділено аналізу функціональності, що наразі реалізована у популярних онлайн-платформах, їхнім перевагам, недолікам та рівню задоволення запитів користувачів.

В межах даного розділу буде виконано такі завдання:

- 1) проведення аналізу ринку з метою визначення основних проблем та потреб у сфері взаємодії з домашніми улюбленцями;
- 2) здійснення огляду програмного забезпечення, яке використовується для пошуку тварин, зоотоварів та послуг, оцінено їхню функціональність та зручність використання;
- 3) формування порівняльної таблиці результатів аналізу;
- 4) визначення ключових функціональних можливостей, що мають бути реалізовані в процесі розробки спеціалізованої вебплатформи.

1.1. Загальні положення та аналіз предметної області

У наш час ефективний пошук інформації, товарів та послуг відіграє вирішальну роль у щоденному цифровому житті користувачів. Особливо це стосується тематики домашніх тварин – сфери, що має свою унікальну специфіку, динаміку й емоційне значення для мільйонів людей. В Україні наразі відсутня єдина, спеціалізована онлайн-платформа, що дозволяла б швидко й зручно знайти домашнього улюбленця, придбати для нього необхідні товари, замовити послуги або проконсультуватися з іншими власниками.

Натомість користувачі змушені шукати тварин та товари для них на загальних дошках оголошень або в соціальних мережах, де інформація часто розпорошена, неповна або неструктурована. А пошук спеціалістів

(ветеринарів, кінологів, грумерів) або відповідей на важливі запитання взагалі перетворюється на квест, який вимагає багато часу та зусиль.

Для більш конкретного уявлення існуючих викликів було побудовано дерево проблем, що представлено на рис. 1.1. Воно висвітлює основні складнощі, з якими стикаються користувачі, зокрема відсутність гнучких фільтрів, цифрових профілів тварин, перевірених продавців, форуму та централізованої бази знань.



Рис. 1.1. Дерево проблем, що виникають у сфері сервісів для домашніх тварин

Запропонований застосунок покликаний вирішити наведені проблеми шляхом створення єдиної спеціалізованої платформи, орієнтованої виключно на тему домашніх тварин.

Це не просто дошка оголошень, а багатофункціональний сервіс, який дозволяє:

- 1) знайти чи виставити тварину на продаж або в добрі руки;
- 2) за допомогою розширених фільтрів швидко підібрати тварину, навіть якщо користувач не орієнтується в породах;
- 3) придбати товари (корм, аксесуари, одяг тощо) і замовити послуги для улюбленця, не виходячи з застосунку;
- 4) створити профіль тварини для персоналізованих рекомендацій і спрощеного пошуку;
- 5) ознайомитись із перевіреними матеріалами щодо догляду, харчування, здоров'я тощо;
- 6) отримати відповіді на свої питання на форумі або, в перспективі, проконсультуватись із фахівцем.

Таким чином, застосунок поєднує функції тематичної дошки оголошень, онлайн-магазину, бази знань та спільноти любителів тварин. Особливий акцент зроблено на зручність інтерфейсу та точність пошуку за рахунок унікальних фільтрів, адаптованих до кожної категорії тварин (собаки, коти, риби, птахи, рептилії, худоба тощо).

Отже, аналіз предметної області підтверджує актуальність розробки подібного сервісу. Він здатен не лише зекономити час користувача та допомогти ефективніше вирішувати практичні завдання, як-от пошук чи продаж тварин, а й сприяти популяризації відповідального ставлення до їх утримання та створювати середовище для обміну досвідом, поширення корисної інформації, об'єднуючи власників у єдину спільноту.

1.2. Аналіз існуючих рішень

Перед розробкою власного застосунку доцільно проаналізувати наявні на ринку сервіси, що мають схожу функціональність та є досить успішними. Це дозволить визначити їхні сильні сторони, виявити недоліки та певні функціональні й концептуальні прогалини, які може заповнити нова

платформа. У межах цього підпункту буде розглянуто три популярні сервіси – OLX [5], є-Пет [6] та Lapa.shop [7], які так чи інакше пов’язані з темою купівлі, продажу, обміну або догляду за домашніми тваринами. Їх порівняння допоможе краще оцінити поточний стан ринку, потреби користувачів та напрямки, у яких новий застосунок може запропонувати інноваційні рішення.

Основними критеріями для проведення об’єктивного аналізу визначено наступні:

1. Наявність широкої категоризації за типами тварин.
2. Можливість придбати супутні товари та послуги.
3. Наявність розширених та унікальних фільтрів окремо для кожної категорії оголошень.
4. Цифровий профіль тварини.
5. Наявність стрічки корисних матеріалів.
6. Наявність форуму.
7. Система рейтингів і перевірки продавців.
8. Персоналізований пошук та рекомендації.
9. Дизайн та зручність користування.

Здійснимо детальний огляд кожного із зазначених сервісів окремо з урахуванням вищенаведених вимог.

1.2.1. OLX

OLX – один із найбільших онлайн-майданчиків в Україні, що вже багато років утримує стабільні позиції на ринку купівлі-продажу. Його популярність, широка аудиторія та висока впізнаваність забезпечили сервісу репутацію надійного та зручного інструменту для розміщення оголошень найрізноманітніших категорій – від електроніки й одягу до тварин та нерухомості (рис. 1.2). Завдяки цьому OLX став одним із головних конкурентів для нових платформ, що прагнуть зайняти нішу оголошень, зокрема у сфері зоотоварів.

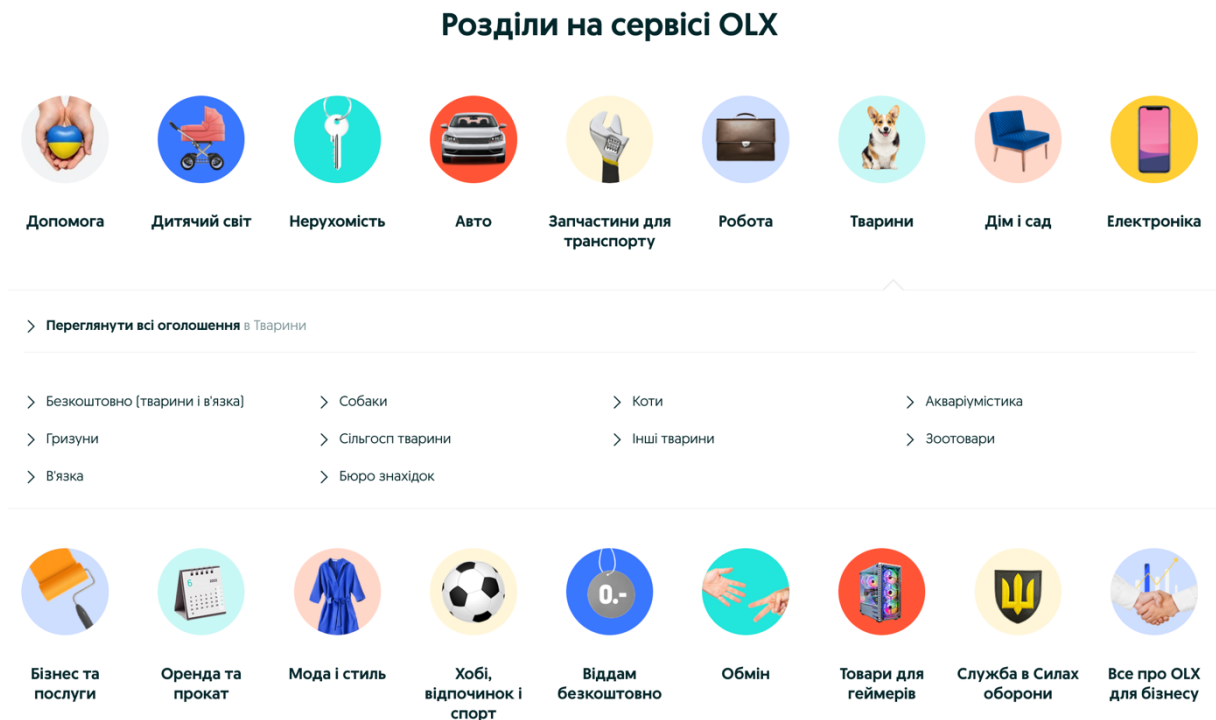


Рис. 1.2. Розділи на сервісі OLX з підрозділами для тварин

У контексті оголошень про домашніх улюбленців OLX виступає сильним конкурентом, однак має і ряд суттєвих обмежень, зумовлених його універсальністю.

Серед сильних сторін OLX можна виділити:

1. Велика база різноманітних оголошень, у тому числі в категорії тварин.
2. Високий рівень довіри серед користувачів завдяки рокам стабільної роботи.
3. Зручний, інтуїтивно зрозумілий інтерфейс, адаптований до мобільних пристроїв.
4. Система рейтингів, яка допомагає визначати надійних продавців.
5. Інтегрована система безпечної оплати та доставки через популярні служби.
6. Можливість знайти супутні товари для тварин в одному застосунку.

Однак у сфері зоотоварів і пошуку тварин були виявлені й наступні слабкі сторони:

1. Відсутність вузької спеціалізації: поруч із тваринами оголошення про одяг, меблі, роботу тощо.
2. Недостатньо гнучкі фільтри (рис. 1.3): обмеження лише базовими параметрами (ціна, порода, регіон).
3. Відсутність можливості створити профіль улюбленця та здійснювати персоналізований пошук.
4. Відсутні форуми, статті, поради або спільноти для власників тварин.
5. Велика кількість нерелевантних оголошень ускладнює пошук і вимагає багато часу.
6. Сервіс стає перенасиченим функціональністю, що розсіює увагу користувача.

Що шукаєте? Уся Україна Пошук

☐ Тільки з фото ☐ Тільки з доставкою

Фільтри

Категорія Собаки

Ціна Від: до:

Порода собаки Коргі

Всі оголошення Бізнес Приватні

Продаж собак та цуценят Коргі

Рис. 1.3. Доступні фільтри OLX для пошуку собак

Таким чином, OLX хоча й є зручною і популярною платформою для загальних потреб, усе ж поступається спеціалізованим сервісам, коли мова йде про відповідальний вибір домашнього улюбленця та подальший пошук товарів і послуг для нього.

1.2.2. є-Пет

є-Пет – досить новий український сервіс, який стрімко розвивається у сфері пошуку та продажу домашніх тварин (рис. 1.4). Хоча платформа ще не досягла масштабів найбільших маркетплейсів, вона вже має вагомі переваги, що пов’язані з її більш вузькою спеціалізацією та орієнтацією саме на власників і майбутніх власників тварин.

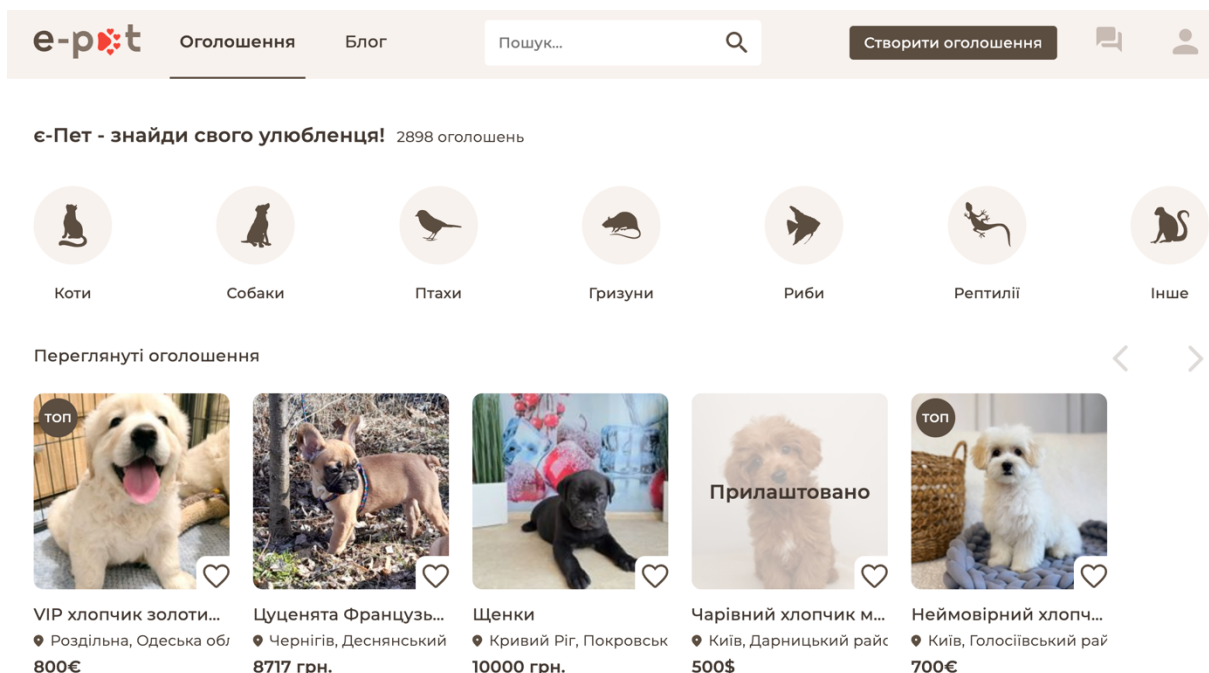


Рис. 1.4. Головна сторінка є-Пет з категоріями та переглянутими оголошеннями

Сильні сторони є-Пет:

1. Інтуїтивний і приємний інтерфейс, що адаптується для різних пристроїв.
2. Відносно розширені можливості фільтрації оголошень: порода, вид, ціна, стать, вік, місто, стан здоров’я, документи, тип продавця тощо (рис. 1.6).
3. Окремий розділ з описами порід і можливість переглянути відповідну корисну інформацію під кожним оголошенням.

4. Наявність блогу з порадами, історіями та фактами про догляд за тваринами (рис. 1.5).
5. Можливість перейти на акаунти продавців у соціальних мережах для перевірки їхньої активності та подальшої комунікації.

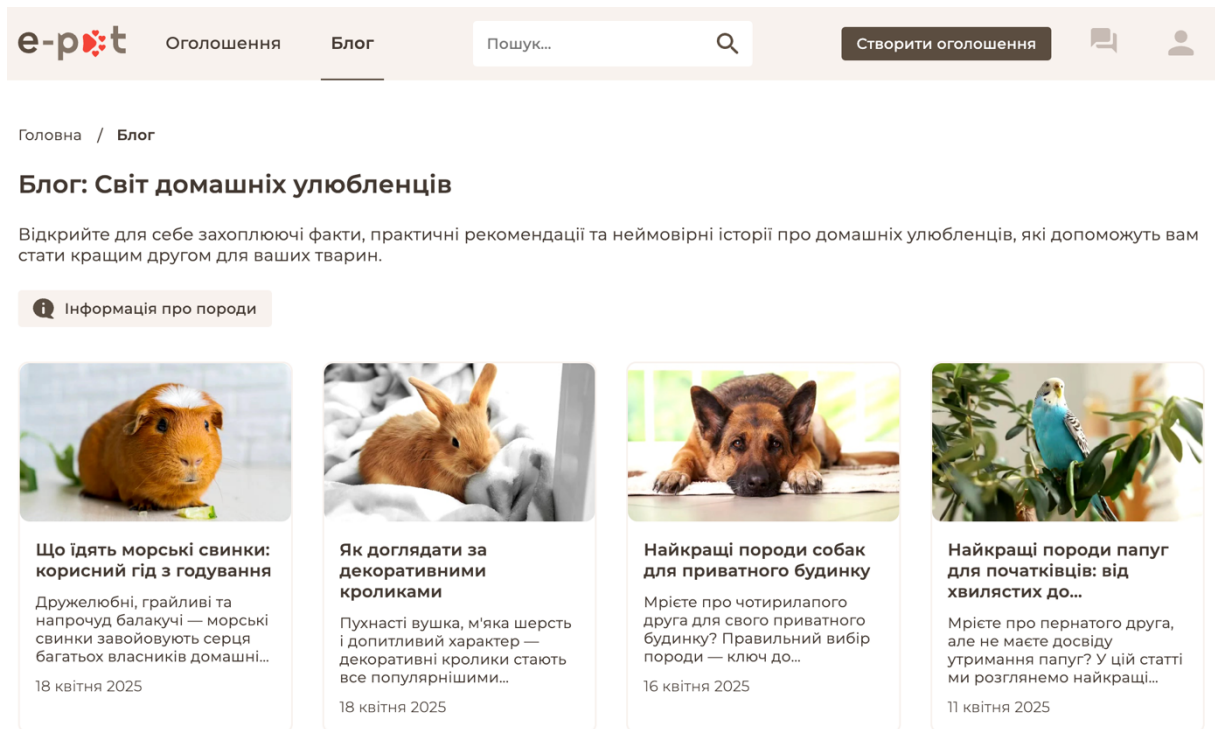


Рис. 1.5. Блог є-Пет з корисними статтями

Слабкі сторони є-Пет:

1. Порівняно невелика кількість оголошень через молодість проєкту.
2. Обмежене представлення категорій тварин і невелика кількість описів порід менш популярних видів.
3. Відсутність можливості здійснювати покупку напряду через платформу, що збільшує ризики для користувачів.
4. Немає окремого розділу для придбання супутніх товарів чи пошуку послуг для тварин.
5. Відсутній форум для обміну досвідом між власниками тварин.
6. Відсутність можливості для створення профілів тварин і відповідно персоналізованого пошуку на основі нього.

Ці обмеження свідчать про те, що проєкт перебуває на ранньому етапі розвитку та має значний потенціал для вдосконалення. Вони також відкривають можливості для створення більш комплексного та функціонального рішення, яке краще відповідатиме потребам цільової аудиторії.

Фільтри

Стать

Самець Самка

Вік (у місяцях)

до 3 3 - 6 6 - 12 від 12

Здоров'я

Чіп Стерилізація

Обробка від паразитів

Вакцинація

Документи

Ветпаспорт Родовід

FCI/КСУ Щеняча метрика

Рис. 1.6. Частина доступних фільтрів є-Пет для пошуку собак

У цілому, є-Пет демонструє перспективний підхід до розвитку спеціалізованої платформи для власників тварин. Однак, незважаючи на загальну зручність та фокусованість на тематиці домашніх улюбленців, певні функціональні обмеження та прогалини залишають простір для вдосконалення та росту.

1.2.3. Lapa.shop

Lapa.shop – нішева українська платформа, орієнтована на продаж виключно породистих котів та собак (рис. 1.7). Сервіс об'єднує перевірених

заводчиків та акцентує увагу на якості, здоров'ї й чистоті походження тварин. Його головною місією є створення безпечного простору для відповідальної купівлі домашніх улюбленців, що має на меті стати альтернативою традиційним дошкам оголошень (рис. 1.8).

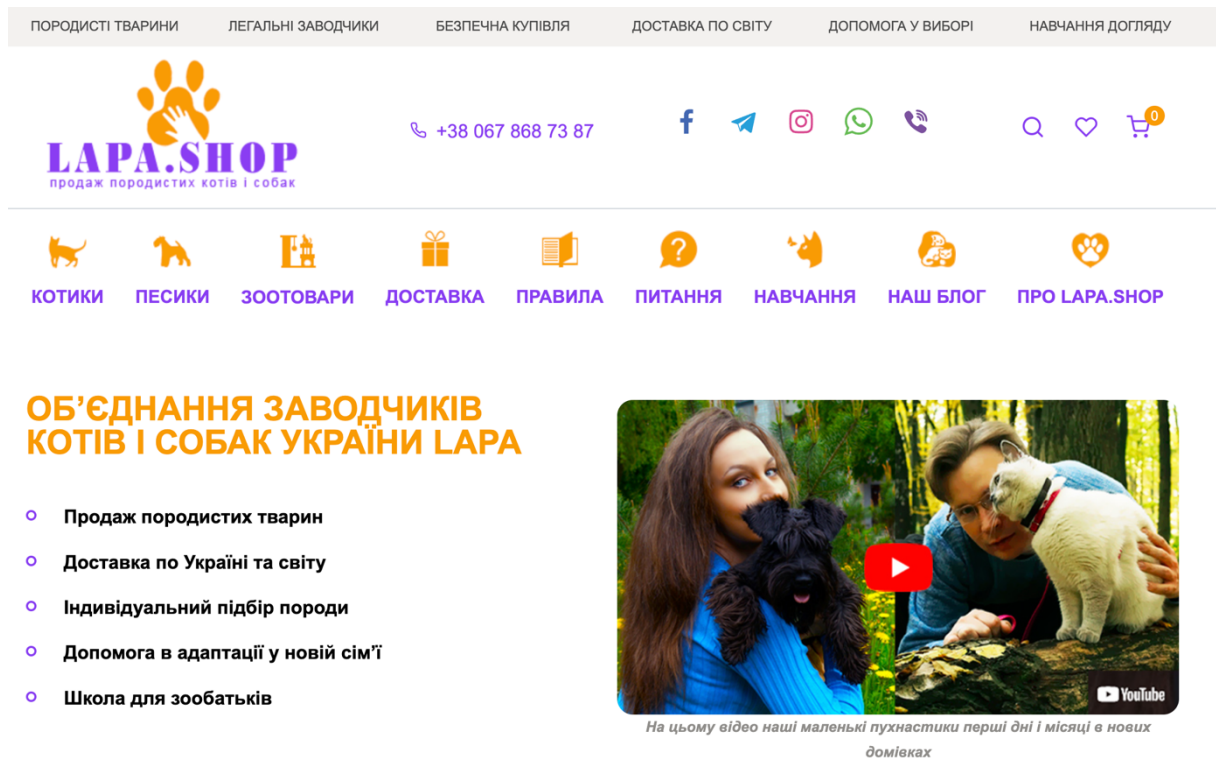


Рис. 1.7. Головна сторінка Lapa.shop з розділами та ключовими послугами

Блог об'єднання заводчиків Lapa: наші уроки, статті та огляди

Корисні статті та огляди від сервісу lapa.shop, навчальні матеріали, а також наші повідомлення.

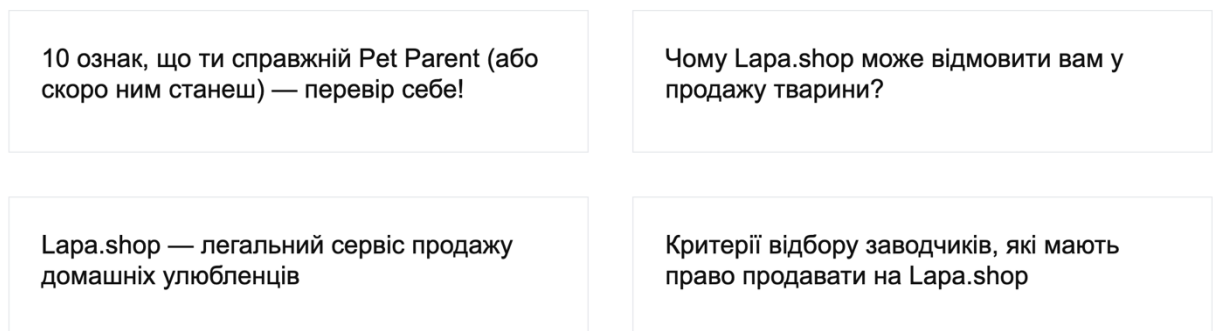


Рис. 1.8. Блог Lapa.shop з корисними матеріалами

Застосунок помітно вирізняється серед конкурентів завдяки наступним особливостям:

1. Робота лише з перевіреними заводчиками, що мінімізує ризики шахрайства.
2. Орієнтація на породистих та здорових тварин із підтвердженим походженням.
3. Наявність додаткових послуг: індивідуальний підбір породи, допомога в адаптації, школа для зообатьків.
4. Зручність вибору породи завдяки наявності фото для кожної.
5. Можливість доставки тварин по Україні та за кордон.
6. Наявність власного блогу із навчальними матеріалами та оглядами.

Знайди найкращого в світі песика!

На сайті об'єднання заводчиків Лара можна купити породисте цуценя в Україні з доставкою. Тут продають лише офіційні заводчики! Породу підтверджено документом про походження. Консультації професіоналів на всіх етапах купівлі.

- **Гарантія породи**
- **Кращі розплідники**
- **Допомога у виборі улюбленця**
- **Навчання догляду**



ТІЛЬКИ ПРИ ПОКУПКІ
НА LARA.SHOP

Офіційне підтвердження породи: родовід/метрика для всіх котів і собак.

Можливість онлайн-оглядин kota чи собаки і особисте знайомство з заводчиком.

Повернення коштів, якщо тварина не може бути доставлена через форс-мажор або ветогляд показав погані результати.

Навчання догляду і 1 рік безкоштовних консультацій.

Рис. 1.9. Сторінка Lara.shop для вибору бажаної породи собаки

Водночас, попри низку переваг, платформа має й певні обмеження:

1. Спеціалізація виключно на котах та собаках.
2. Незручний пошук – відсутні будь-які фільтри.

3. Відсутність можливості самостійно розміщувати оголошення користувачами або співпрацювати з притулками.
4. Обмежений вибір аксесуарів та супутніх товарів.
5. Відсутність форуму або спільноти для взаємодії власників тварин.
6. Доступність різних порід залежить від наявності, що іноді змушує користувачів чекати.
7. Дещо застарілий інтерфейс, який поступається сучасним тенденціям вебдизайну.
8. Відсутність персоналізованих рекомендацій або гнучкої системи підбору товарів і послуг під конкретного улюбленця.

Отже, Lapa.shop є якісною та безпечною платформою для відповідальної купівлі породистих котів і собак, однак її вузька спеціалізація і функціональні обмеження поки що не дозволяють повністю закрити потреби ширшої аудиторії власників тварин та, наприклад, притулків.

1.3. Результати проведеного аналізу

Аналіз існуючих рішень дозволив виявити як сильні сторони аналогічних платформ, так і їхні недоліки, що обмежують зручність та ефективність користування. Розглянуті сервіси – OLX, є-Пет та Lapa.shop – хоча й мають спільну мету сприяти пошуку домашніх улюбленців, однак орієнтуються на різні аудиторії і суттєво відрізняються за функціональністю та рівнем спеціалізації.

OLX є універсальною платформою з великою базою оголошень, співпрацею з сервісами доставки, вбудованою безпечною оплатою та високим рівнем довіри користувачів, але через свою багатопрофільність не може забезпечити всі потреби користувачів у питаннях пошуку зоотоварів і тварин. є-Пет вигідно вирізняється розширеними фільтрами, корисними статтями та зосередженістю на тематиці домашніх улюбленців, проте поступається кількістю оголошень, відсутністю інтегрованої системи

оплати та оголошень з зоотоварами чи послугами. Lapa.shop має дещо інший вектор розвитку і пропонує надійний та «білий» продаж породистих тварин завдяки співпраці виключно з перевіреними заводчиками, однак обмежується лише певними видами тварин, що не завжди в наявності, та не має зручного пошуку.

Щоб узагальнити результати порівняння, було складено табл. 1.1 з метою оцінки відповідності платформ основним критеріям аналізу.

Таблиця 1.1

Порівняльна таблиця аналогів

Вимога \ Застосунок	OLX	є-Пет	Lapa.shop
Широка категоризація за типами тварин	+	+	—
Продаж супутніх товарів та послуг	+	—	+/-
Розширені фільтри пошуку для кожної категорії	—	+	—
Цифровий профіль тварини	—	—	—
Стрічка корисних матеріалів	—	+	+
Форум	—	—	—
Рейтингова система та перевірка продавців	+	+/-	+
Персоналізований пошук та рекомендації	—	—	—
Сучасний дизайн та зручність користування	+	+	—

Як видно з наведеної таблиці, жодна з розглянутих платформ не поєднує у собі всі необхідні можливості для повноцінної підтримки користувачів у сфері пошуку, купівлі та догляду за домашніми

улюбленцями. Кожен із сервісів має свої сильні сторони, проте окремі аспекти, як-от обмежений вибір категорій тварин, відсутність розширених фільтрів, неможливість персоналізації чи взаємодії спільноти, залишаються недоопрацьованими, що відкриває перспективи для створення нового спеціалізованого застосунку.

Таким чином, продукт, що розробляється, має на меті об'єднати переваги наявних рішень і водночас усунути їхні недоліки, створюючи комплексне рішення для спільноти любителів тварин.

1.4. Висновки до розділу 1

У першому розділі було здійснено огляд і аналіз існуючих рішень у сфері онлайн-платформ для розміщення оголошень про домашніх тварин, супутні товари та послуги. Спочатку розглянуто актуальність створення спеціалізованого застосунку, побудовано дерево проблем галузі та окреслено основні особливості майбутнього сервісу.

Далі було проведено детальний аналіз існуючих аналогів – OLX, є-Пет та Lapa.shop – за визначеними критеріями. Виявлено їхні сильні та слабкі сторони у контексті роботи з оголошеннями про тварин та зоотовари. На основі цього аналізу було сформульовано порівняльні висновки та наведено узагальнюючу таблицю, що дозволила наочно оцінити відповідність платформ вимогам користувачів.

Аналіз показав, що жодна з розглянутих платформ не забезпечує повного покриття усіх ключових потреб: зручного персоналізованого пошуку, роботи з профілями тварин, можливості відразу придбати супутні товари і послуги, а також створення простору для комунікації власників разом з наданням доступу до якісної інформації. Це підтверджує доцільність розробки нового спеціалізованого застосунку, який об'єднає ці можливості в єдиній системі.

2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ

У цьому розділі буде виконано технічне обґрунтування засобів розроблення вебзастосунку для підтримки спільноти власників домашніх тварин. Зокрема, розглядатимуться можливі варіанти технологій для створення клієнтської та серверної частин, а також вибір системи керування базами даних.

Для розроблення застосунку обрано клієнт-серверну архітектуру, яка забезпечує розділення логіки між фронтендом та бекендом, сприяє кращій масштабованості, повторному використанню компонентів і спрощенню супроводу системи.

У межах аналізу буде порівняно популярні мови та платформи програмування для серверної частини, клієнтські фреймворки, а також системи управління базами даних. За результатами аналізу буде зроблено обґрунтований вибір оптимальних засобів реалізації для кожного компоненту системи.

2.1. Вибір технологій для розроблення серверної частини

На поточному етапі важливо обрати серверні технології, які забезпечать стабільну та безпечну роботу майбутнього вебзастосунку. Оскільки функціональність охоплюватиме обробку складних запитів, реєстрацію та авторизацію користувачів, пошук, сортування і фільтрацію, листування та сповіщення в реальному часі, рольову модель доступу тощо, вибір повинен бути зваженим.

Визначимо основні критерії порівняння:

1. Продуктивність у роботі з великими обсягами даних та запитів.
2. Підтримка та зручність асинхронної обробки операцій.
3. Наявність вбудованої або легко інтегрованої підтримки реального часу через WebSocket чи відповідні бібліотеки.

4. Можливість побудови безпечної автентифікації та рольової авторизації, включно з підтримкою сучасних стандартів безпеки (OAuth2, JWT).
5. Зручність масштабування та побудови модульної архітектури.
6. Швидкість розробки та наявність зрозумілої і вичерпної документації.
7. Розвиненість екосистеми та наявність перевірених бібліотек.
8. Ступінь типізації та контроль над даними.

Для оцінки будуть розглянуті такі варіанти: C# (ASP.NET Core), PHP (Laravel) та Python (Django). На основі аналізу за цими критеріями буде обґрунтовано вибір оптимальної технології для серверної частини застосунку.

2.1.1. C# та фреймворк ASP.NET Core

C# – найпопулярніша мова програмування платформи .NET від компанії Microsoft [8]. Її можна охарактеризувати як сучасну, об'єктно-орієнтовану, статично типізовану, що активно розвивається та вирізняється високою продуктивністю, підтримкою багатьох парадигм програмування і широкими можливостями для роботи з різними середовищами – від десктопних застосунків до хмарних сервісів.

У зв'язці із кросплатформенним фреймворком ASP.NET Core мова C# забезпечує розробників потужним інструментарієм для створення високопродуктивних, безпечних і масштабованих вебзастосунків та API. Така комбінація дозволяє ефективно реалізувати функціональність, необхідну для складних систем – з гнучкою автентифікацією, обробкою запитів у реальному часі, масштабуванням і зручною інтеграцією з базами даних та сторонніми сервісами.

Оцінимо сильні та слабкі сторони цього стеку у контексті розроблюваного застосунку.

Переваги:

1. Висока продуктивність та асинхронність за замовчуванням. ASP.NET Core є одним із найшвидших фреймворків на ринку [9]. Він побудований із врахуванням асинхронної обробки запитів завдяки широкому використанню `async/await` та має оптимізовану роботу з пам'яттю. Все це особливо важливо для платформи, яка потенційно матиме значну кількість користувачів та оголошень, а також виконуватиме складні операції пошуку та фільтрації.
2. Підтримка реального часу через WebSockets. Завдяки вбудованій підтримці бібліотеки SignalR можна легко реалізовувати двосторонній зв'язок між клієнтом і сервером. Це спрощує додавання функціональності онлайн-статусів користувачів, приватних чатів, сповіщень тощо.
3. Статична типізація. C# забезпечує сувору перевірку типів на етапі компіляції, що мінімізує помилки і полегшує супровід великих проєктів.
4. Готові рішення для безпечної автентифікації та авторизації. Вбудований механізм ASP.NET Identity забезпечує просте впровадження реєстрації, входу, рольової моделі доступу, токен-автентифікації (JWT) та OAuth2 для підвищення безпеки.
5. Гнучка інтеграція із сучасними базами даних. Entity Framework Core забезпечує зручне відображення об'єктів у базі даних, підтримуючи складні запити, зв'язки та міграції через ORM-інструмент.
6. Масштабованість. Фреймворк підтримує поділ на модулі або перехід до мікросервісної архітектури без суттєвої зміни структури проєкту, що важливо для поступового розвитку платформи.
7. Розвинута екосистема. Наявність великої кількості бібліотек (NuGet-пакетів), відкритого коду, навчальних матеріалів, активної спільноти і вбудованих рішень значно полегшує розробку.

Недоліки:

1. Вищий поріг входу для початківців. Може знадобитись глибше розуміння принципів ООП, асинхронності та налаштування середовища, що може ускладнити старт порівняно з простішими платформами.
2. Потенційно вищі витрати на хостинг. Незважаючи на підтримку Linux і кросплатформенність, розгортання .NET-застосунків іноді пов'язане з більшими витратами на інфраструктуру у порівнянні з іншим стеком технологій.
3. Ресурсомісткість. Для обробки великої кількості дрібних запитів може знадобитися додаткова оптимізація пам'яті порівняно з деякими іншими платформами.

2.1.2. PHP та фреймворк Laravel

Популярною альтернативою для розробки серверної частини такого роду застосунків, а саме маркетплейсів, є використання мови програмування PHP у поєднанні з фреймворком Laravel.

PHP вже багато років активно використовується для створення вебзастосунків різного рівня складності, а Laravel надає зручну структуру та набір інструментів, які значно прискорюють типові завдання.

Даний стек орієнтований на простоту, гнучкість і високу швидкість розробки, що робить його популярним вибором для невеликих команд та проєктів із широкою, але стандартною функціональністю. Розглянемо його крізь призму вимог до розроблюваної платформи.

Переваги:

1. Швидкий старт розробки. Зручний і зрозумілий синтаксис та добре структурована документація прискорюють розробку типових компонентів.

2. Доступність екосистеми та спільноти. PHP/Laravel має активну спільноту, розвинений репозиторій пакетів (Composer/Packagist) та надає багато готових інструментів «із коробки».
3. Зручна реалізація авторизації та автентифікації. Laravel Breeze/Jetstream/Passport надають готові рішення для безпечної автентифікації, авторизації та управління ролями/дозволами.

Недоліки:

1. Обмежена продуктивність складних запитів. При обробці великих обсягів даних та складних, багатопараметричних запитів продуктивність може бути нижчою порівняно зі скомпільованими мовами.
2. Відсутня зручна вбудована підтримка асинхронності. За своєю природою PHP є синхронним [10], а асинхронність реалізується через зовнішні бібліотеки (ReactPHP, Swoole) або специфічні підходи, що може ускладнити архітектуру.
3. Динамічна типізація. Динамічна природа може ускладнити контроль над дуже різноманітними та складними структурами даних, збільшуючи ризик помилок.
4. Складність реалізації реального часу. Виникає потреба в інтеграції окремих серверів та бібліотек для WebSocket.

2.1.3. Python та фреймворк Django

Іншим потужним та популярним стеком для веброзробки є мова програмування Python у поєднанні з фреймворком високого рівня Django. Django відомий своєю філософією «батареї в комплекті» [11], прагнучи забезпечити розробників усіма необхідними вбудованими компонентами для типових задач. Python, завдяки своїй читабельності та широкому спектру застосувань, є привабливим вибором для багатьох проєктів. Проаналізуємо його переваги та недоліки в контексті реалізації платформи для домашніх тварин.

Переваги:

1. Швидкість розробки. Django надає готові рішення для багатьох стандартних можливостей (ORM, система автентифікації, маршрутизація, форми, шаблони, адміністративна панель тощо), що значно прискорює початкову розробку основних модулів.
2. Широка екосистема та спільнота. Python має великий вибір бібліотек, а Django користується значною популярністю, що забезпечує доступ до великої кількості готових рішень та активної спільноти розробників для вирішення поширених проблем.
3. Інтегровані засоби безпеки. Django має вбудовані механізми для захисту від поширених вебзагроз та зручну систему управління користувачами та їхніми дозволами.
4. Зручна модульність. Структура проєктів у Django розділена на незалежні «додатки» (apps), що сприяє побудові модульної та легко підтримуваної архітектури.

Недоліки:

1. Потенційні обмеження продуктивності. Global Interpreter Lock (GIL) у стандартному CPython може обмежувати ефективність використання можливостей багатоядерних процесорів для чисто обчислювальних завдань.
2. Слабка типізація. Як і PHP, Python є динамічно типізованою мовою (хоча підтримує type hints).
3. Особливості асинхронної логіки. Django був спочатку синхронним, і хоча з версії 3.x з'явилась підтримка async, ця модель може відрізнятися від підходів, реалізованих в інших технологіях, та вимагати специфічного розуміння та інтеграції для ефективної роботи з I/O-залежними операціями.
4. Менша ефективність при високому навантаженні. Python-процеси споживають більше ресурсів у порівнянні з мовами типу C# або Java, особливо при великій кількості одночасних запитів.

5. Неоптимальна продуктивність ORM при складних запитах. Django ORM зручний, але не завжди ефективний при складних фільтрах і об'єднаннях та часто доводиться писати «сирі» SQL-запити.

2.1.4. Результати аналізу

У ході вибору серверної частини для реалізації дипломного вебзастосунку було проаналізовано три сучасні технології: PHP у зв'язці з фреймворком Laravel, Python із Django та C# з ASP.NET Core. Порівняння відбувалося за критеріями, що безпосередньо впливають на надійність, продуктивність і масштабованість майбутньої системи: підтримка асинхронної обробки, якість реалізації автентифікації, можливість роботи в реальному часі, розширюваність, типізація, документація, екосистема тощо.

Laravel показав зручність у створенні базової бізнес-логіки та швидкий вхід для розробника, проте його динамічна природа, обмеженість у масштабуванні та менш гнучкі можливості асинхронної обробки створюють потенційні труднощі при реалізації навантаженого фільтрами та взаємодією маркетплейсу. Django продемонстрував швидкість побудови стандартних модулів і зручну інтеграцію базових сервісів, однак у складних сценаріях фільтрації та в умовах високого навантаження виявився менш гнучким, а динамічна типізація Python не забезпечує належного контролю над даними в умовах швидко зростаючого коду.

Натомість C# у зв'язці з ASP.NET Core виявився найкраще пристосованим до специфіки проєкту. Його строгий контроль типів дозволяє ефективно працювати зі складними структурами даних, що застосовуватиметься в проєкті. Платформа забезпечує надійний фундамент для побудови масштабованого, продуктивного та захищеного застосунку із підтримкою сучасних підходів до автентифікації та авторизації, реалізації асинхронних запитів і зручною архітектурною роздільністю. Саме тому серед усіх розглянутих варіантів було обрано C# і ASP.NET Core як

найдоцільніший технологічний стек для серверної частини дипломної роботи.

2.2. Обґрунтування вибору системи керування базами даних

У рамках розробки вебзастосунку важливим етапом стало визначення оптимальної системи керування базами даних, яка б відповідала вимогам до зберігання, обробки та масштабування структурованої та частково динамічної інформації. Особливості предметної області, зокрема наявність сутностей із гнучкими атрибутами (наприклад, тварини з різними характеристиками), вимагали врахування як реляційних, так і документно-орієнтованих підходів до моделювання даних. Для обґрунтованого вибору було проведено порівняння двох сучасних СКБД – Microsoft SQL Server як представника класичних реляційних систем та MongoDB як приклад гнучкої NoSQL-архітектури.

2.2.1. Microsoft SQL Server

Microsoft SQL Server є однією з провідних реляційних систем керування базами даних (РСКБД), що характеризується широким набором функціональних можливостей та розвиненою екосистемою. Розглянемо її потенційні переваги та недоліки для використання у проєкті багатофункціональної платформи для домашніх тварин.

Переваги:

1. Надійна підтримка реляційної моделі, що забезпечує чітку структурування даних, визначення зв'язків та гарантування цілісності даних за допомогою стандартних механізмів (зовнішні ключі, обмеження, ACID-транзакції).
2. Потужні можливості для складних запитів. SQL дозволяє ефективно здійснювати складні операції з даними, включаючи розширену фільтрацію та агрегацію, що є важливим для реалізації функціональності пошуку у застосунку.

3. Висока надійність та масштабованість. MS SQL Server відомий своєю стабільністю та різноманітними опціями масштабування для обробки зростаючих обсягів даних та навантаження [12].
4. Зручна інтеграція з технологіями Microsoft. Для проєктів, що використовують стек Microsoft (наприклад, ASP.NET Core), інтеграція з MS SQL Server може спростити процес розробки та розгортання завдяки наявності нативних інструментів та бібліотек.
5. Наявність різноманітних механізмів безпеки, що можуть забезпечити надійний захист даних користувачів та платформи.

Недоліки:

1. Комерційне ліцензування. MS SQL Server є комерційним продуктом, що може призвести до значних витрат на ліцензії, особливо при великих обсягах даних та високих вимогах до продуктивності.
2. Менша гнучкість схеми порівняно з NoSQL. Реляційна модель вимагає чіткого визначення схеми даних, що може бути менш гнучким при часто змінюваних вимогах до структури даних або великій кількості необов'язкових атрибутів.
3. Залежність від платформи. Хоча MS SQL Server доступний на Linux, його екосистема та найкраща підтримка традиційно пов'язані з операційною системою Windows.

2.2.2. MongoDB

MongoDB є документо-орієнтованою NoSQL системою керування базами даних з відкритим вихідним кодом. Вона зберігає дані у гнучких JSON-подібних документах, що дозволяє представляти складні структури даних без необхідності жорсткого визначення схеми [13]. Наведемо її сильні та слабкі сторони для реалізації платформи.

Переваги:

1. Гнучкість схеми. MongoDB дозволяє документам мати різну структуру, що може спростити управління великою кількістю унікальних атрибутів, характерних для різних видів тварин та категорій товарів.
2. Горизонтальна масштабованість. MongoDB розроблена з урахуванням горизонтального масштабування, що може бути важливим для обробки великих обсягів даних та високого навантаження у майбутньому.
3. Простота розробки. Робота з JSON-подібними документами може бути інтуїтивно зрозумілою для розробників вебзастосунків, оскільки цей формат часто використовується для обміну даними між клієнтом та сервером.
4. Відкритий вихідний код та безкоштовне використання (Community Edition). Основна версія MongoDB є безкоштовною для використання, що може знизити початкові витрати на ліцензування.

Недоліки:

1. Складність забезпечення ACID-транзакцій. У порівнянні з реляційними СУБД, гарантії консистентності даних можуть бути нижчими, що ризиковано для важливої інформації.
2. Потенційні складності при реалізації складних зв'язків. Хоча MongoDB підтримує вкладені документи, реалізація складних, нормалізованих зв'язків «багато-до-багатьох», типових для реляційних баз даних, може бути менш ефективною та вимагати додаткових зусиль на рівні застосунку.
3. Продуктивність складних запитів. Пошук з багатьма критеріями, особливо по різних атрибутах, може бути повільнішим та складнішим в оптимізації, ніж SQL-запити.
4. Зрілість екосистеми інструментів. Хоча екосистема MongoDB постійно розвивається, набір зрілих інструментів для

адміністрування, моніторингу та аналітики може бути меншим порівняно з такими усталеними РСКБД, як Microsoft SQL Server.

5. Ризики для цілісності. Відсутність жорсткої схеми може ускладнити забезпечення консистентності даних на рівні бази даних, перекладаючи більшу відповідальність за валідацію та контроль на рівень застосунку.

2.2.3. Результати аналізу

Проведений порівняльний аналіз двох ключових підходів до організації зберігання даних – реляційної СУБД Microsoft SQL Server та документо-орієнтованої NoSQL СУБД MongoDB – дозволив визначити оптимальний вектор руху для забезпечення ефективної роботи застосунку.

MongoDB, завдяки своїй гнучкій схемі, демонструє переваги в управлінні слабоструктурованими або часто змінюваними даними, а також забезпечує ефективне горизонтальне масштабування. Ці характеристики могли б здаватися привабливими з огляду на необхідність управління різноманітними атрибутами тварин та товарів, однак, враховуючи критичну важливість забезпечення високої консистентності даних, особливо при складних бізнес-процесах, а також необхідність ефективної обробки складних запитів з численними критеріями фільтрації, що є ключовим для функціональності пошуку на платформі, Microsoft SQL Server виявився більш обґрунтованим вибором.

MS SQL Server, як надійна реляційна СУБД, забезпечує строгу підтримку ACID-транзакцій, що гарантує цілісність даних на рівні бази даних. Потужний SQL та оптимізатор запитів MS SQL Server надають значні переваги при виконанні складних операцій з даними, що є важливим для ефективної реалізації комплексного пошуку. Крім того, зріла екосистема інструментів, надійність та можливості масштабування MS SQL Server роблять його кращим вибором для довгострокової підтримки та розвитку платформи. Враховуючи також вибір ASP.NET Core як основної технології

серверної частини, інтеграція з MS SQL Server є більш природною та забезпечує зручність розробки та розгортання.

Таким чином, незважаючи на гнучкість MongoDB, для забезпечення надійності, консистентності даних та високої продуктивності при складних запитах, особливо в контексті вимог до управління різнорідними даними, Microsoft SQL Server є найбільш оптимальним вибором серед систем керування базами даних для розроблюваної платформи.

2.3. Вибір технологій для розроблення клієнтської частини

Для створення інтерактивного та зручного інтерфейсу вебзастосунку важливим є вибір технологій клієнтської розробки. Серед популярних JavaScript-інструментів виділяються Angular, React та Vue.js. Кожен з них має свої особливості, переваги та недоліки. У цьому розділі буде проведено порівняльний аналіз цих технологій для визначення оптимального рішення, що відповідатиме потребам розроблюваної платформи для домашніх тварин.

2.3.1. Angular

Angular – це комплексний TypeScript-фреймворк з відкритим вихідним кодом, розроблений Google для побудови масштабних та складних односторінкових вебзастосунків (SPA). Він надає розробникам структуровану архітектуру, багатий набір вбудованих інструментів та чіткі правила розробки [14].

Переваги:

1. Структурована архітектура (MVC/MVVM), що сприяє кращій організації кодової бази, її підтримці та масштабованості, особливо у великих проєктах.
2. Багатий набір вбудованих можливостей (маршрутизація, форми, модуль HTTP тощо).

3. Використання TypeScript, що забезпечує статичну типізацію, яка полегшує виявлення помилок на етапі розробки, покращує читабельність коду та спрощує його підтримку.
4. Потужний CLI (Command Line Interface). Angular CLI надає зручні інструменти для автоматизації рутинних завдань, таких як створення компонентів, сервісів, модулів, збірка проєкту та запуск тестів, що підвищує продуктивність розробників.
5. Велика активна спільнота та підтримка Google.

Недоліки:

1. Вища крива навчання через комплексність та велику кількість концепцій.
2. Більший розмір бандла, що може впливати на завантаження.
3. Складніші оновлення між версіями.
4. Може бути надмірним для простих проєктів.

2.3.2. React

React – це JavaScript-бібліотека з відкритим вихідним кодом, розроблена Meta для побудови інтерактивних та динамічних користувацьких інтерфейсів. Вона фокусується на компонентному підході та ефективному оновленні DOM за допомогою віртуального DOM.

Переваги:

1. Простота вивчення основ. Базові концепції React відносно прості для розуміння, особливо для розробників, знайомих з JavaScript.
2. Компонентний підхід. React заохочує розбиття інтерфейсу на незалежні багаторазові компоненти, що полегшує розробку, тестування та підтримку коду, а також сприяє його повторному використанню [15].
3. Віртуальний DOM, що використовується для оптимізації оновлень реального DOM і, як наслідок, призводить до підвищення продуктивності застосунків, особливо при частих змінах стану.

Недоліки:

1. Відсутність повної структури «з коробки» (потрібно самостійно обирати та інтегрувати рішення для маршрутизації, стану тощо).
2. Вища залежність від знань сторонніх бібліотек.
3. Менша передбачуваність кодової бази через гнучкість та відсутність жорстких архітектурних патернів на рівні бібліотеки.
4. Складніше масштабування без чітких архітектурних рішень.

2.3.3. *Vue.js*

Vue.js – це прогресивний JavaScript-фреймворк з відкритим вихідним кодом, призначений для побудови користувацьких інтерфейсів та односторінкових застосунків [16]. Він розроблений з метою бути поступово інтегрованим, що дозволяє використовувати його як невелику бібліотеку для покращення існуючого HTML, так і як повноцінний фреймворк для складних SPA.

Переваги:

1. Легкий поріг входження. Vue.js часто вважається одним з найлегших для вивчення фреймворків завдяки своєму простому синтаксису та зрозумілій документації.
2. Хороша продуктивність «з коробки». Vue.js забезпечує гарну продуктивність рендерингу та оновлення DOM завдяки своїй реактивній системі.
3. Гнучкість та можливість поступової інтеграції в існуючі проєкти.

Недоліки:

1. Недостатні зрілість екосистеми та активність спільноти.
2. Менш чітка архітектура для великих проєктів. Без суворих стандартів структура проєкту може стати нечіткою при масштабуванні.

3. Обмежена офіційна підтримка великих інструментів. Частина рішень підтримується спільнотою, що може впливати на стабільність у довгостроковій перспективі.
4. Менша кількість enterprise-орієнтованих рішень «з коробки».

2.3.4. Результати аналізу

Проведений порівняльний аналіз Angular, React та Vue.js був ключовим для визначення оптимальної технології для розробки клієнтської частини платформи для домашніх тварин. Кожна з цих технологій має свої особливості, які були оцінені з точки зору потреб проєкту.

React, хоча й пропонує гнучкість та велику екосистему, вимагає значних зусиль для організації структури великих застосунків та вибору необхідних інструментів. Vue.js, легкий у вивченні, може мати обмеження у масштабованості та наявності готових enterprise-рішень порівняно з більш зрілими фреймворками.

Angular, як комплексний фреймворк, надає чітку архітектуру, що є значною перевагою для великих проєктів, полегшуючи розробку, підтримку та масштабування. Використання TypeScript підвищує надійність коду. Angular також включає багато необхідних інструментів «з коробки», таких як маршрутизація та управління станом, що зменшує необхідність у виборі та інтеграції сторонніх бібліотек. Потужний CLI спрощує розробку, а велика спільнота та підтримка Google забезпечують стабільність та розвиток фреймворку.

Враховуючи складність та масштаб запланованої платформи, а також необхідність забезпечення стабільності кодової бази та ефективної роботи, Angular представляється найбільш збалансованим та оптимальним вибором для розробки клієнтської частини, поєднуючи в собі потужність, структуру та розвинену екосистему.

2.4. Висновки до розділу 2

У другому розділі було здійснено порівняльний аналіз ключових технологій, необхідних для розробки багатофункціональної платформи для домашніх тварин, охоплюючи серверний рівень, систему керування базами даних та клієнтську частину.

Аналіз варіантів для серверної розробки (C#/.NET Core, PHP/Laravel, Python/Django) обґрунтував вибір C# та ASP.NET Core як найбільш продуктивного, надійного та придатного для управління складними даними рішення, що є важливим для реалізації розширеної функціональності платформи.

Порівняння Microsoft SQL Server та MongoDB як систем керування базами даних показало, що для забезпечення цілісності даних, ефективної обробки складних запитів та надійної роботи з реляційними даними оптимальним вибором є Microsoft SQL Server.

Щодо технологій клієнтської розробки (Angular, React, Vue.js), перевагу було надано Angular. Його структурована архітектура, використання TypeScript та багатий набір вбудованих інструментів забезпечують кращу організацію коду, полегшують розробку великих та складних інтерфейсів, а також сприяють ефективній роботі команди розробників (за наявності в майбутньому).

Таким чином, на основі проведеного аналізу, для реалізації проекту було обрано наступний стек технологій: серверна частина – C# та ASP.NET Core, база даних – Microsoft SQL Server, клієнтська частина – Angular. Цей вибір зумовлений прагненням забезпечити високу продуктивність, надійність, масштабованість та зручність розробки, що є ключовими факторами успішної реалізації комплексного уніфікованого сервісу.

3. РОЗРОБЛЕННЯ ВЕБСЕРВІСУ «PETPLACE»

Цей розділ присвячений детальному опису процесу проєктування та розробки вебзастосунку «PetPlace» – спеціалізованої платформи для розміщення оголошень, пов'язаних із домашніми тваринами, їх утриманням, продажем, обміном, пошуком товарів і послуг, а також комунікацією спільноти власників тварин. У межах цього розділу описуються основні етапи побудови програмної системи, починаючи з формулювання функціональних вимог, моделювання поведінки користувачів і закінчуючи технічними аспектами розробки.

Особливу увагу буде приділено структурі бази даних, яка є основою для зберігання та управління всією інформацією платформи. Вона має враховувати складність предметної області – необхідність роботи з численними категоріями тварин, динамічними фільтрами, профілями улюбленців і великою кількістю супутньої інформації.

Насамкінець, буде розглянуто архітектуру вебзастосунку, яка має забезпечувати розподіл обов'язків між клієнтською та серверною частинами, сприяючи гнучкості, масштабованості та підтримці модульного підходу.

Описані в цьому розділі етапи та рішення закладатимуть основу для створення повноцінного та функціонального вебсервісу «PetPlace».

3.1. Формулювання вимог до розроблюваної системи

Ефективна розробка програмного забезпечення починається з чіткого розуміння потреб користувачів та функціональних можливостей, які повинна підтримувати майбутня система. Формулювання функціональних вимог дозволить окреслити основні сценарії використання платформи, її ключову функціональність та очікувану поведінку. Окрім того, критично важливим для забезпечення позитивного користувацького досвіду та надійності платформи є визначення нефункціональних вимог, що

стосуються якості роботи системи, її безпеки та зручності використання. Окрему увагу варто приділити діаграмі прецедентів, яка наочно демонструє основні сценарії взаємодії користувачів із системою та дозволяє краще зрозуміти її загальну структуру й логіку роботи.

3.1.1. Опис функціональних вимог

Функціональні вимоги – це формалізований опис дій, які має виконувати програмне забезпечення для забезпечення очікуваного результату [17]. Вони визначають, що саме система повинна робити з погляду користувача, і є основою для розробки, тестування та подальшої оцінки ефективності продукту і відповідності очікуванням.

У контексті вебзастосунку «PetPlace» функціональні вимоги сформовано з урахуванням специфіки платформи, що поєднує в собі функціональність онлайн-маркетплейсу для купівлі/продажу тварин і пов'язаних з ними товарів або послуг, а також інформаційного ресурсу з корисними статтями, форумом і можливістю створення профілів улюбленців. Наведені в табл. 3.1 вимоги охоплюють ключові сценарії взаємодії користувача із системою, як-от авторизацією, публікацією оголошень, фільтрацією результатів, спілкуванням між користувачами, переглядом контенту тощо.

Таблиця 3.1

Функціональні вимоги до вебзастосунку

Код вимоги	Опис вимоги
FR1	Користувач повинен мати можливість створити оголошення про тварину, товар або послугу. Водночас система повинна динамічно виводити поля та доступні значення, залежно від вибраної категорії

FR2	Користувач повинен мати змогу додавати до свого профілю одного або кількох улюбленців із зазначенням індивідуальних характеристик відповідно до типу тварини
FR3	Вебінтерфейс має забезпечувати фільтрацію оголошень за ключовими параметрами, що відрізняються в залежності від категорії (наприклад, порода, вік, матеріал, тип послуги тощо)
FR4	Для товарів та послуг має бути реалізовано вибір цільового типу тварини, на основі підкатегорій, що мають батьківську категорію «Тварини»
FR5	Користувачі повинні мати змогу прикріплювати зображення до своїх оголошень із подальшим їх збереженням у хмарному сховищі
FR6	Кожен автор оголошення може редагувати або видаляти його, якщо воно ще не втратило актуальності або не було заблоковане адміністрацією
FR7	Система повинна надавати можливість переглядати список доступних оголошень в обраній категорії
FR8	Сервіс повинен надавати можливість зберігати обрані оголошення в «улюблене» для швидкого доступу
FR9	Для зареєстрованих користувачів реалізується особистий кабінет з переглядом створених оголошень, інформації про улюбленців та активність на платформі
FR10	Користувач повинен мати можливість пошуку товарів та послуг на основі обраного профілю улюбленця із автоматичним застосуванням усіх відповідних фільтрів
FR11	Для кожної послуги автор може вказати доступні дні та часові проміжки, в які ця послуга надається
FR12	Користувач повинен мати змогу писати повідомлення власникам оголошень через вбудовану систему повідомлень, яка зберігає історію діалогу
FR13	Адміністратор повинен мати змогу переглядати, блокувати або видаляти будь-яке оголошення, яке порушує правила користування сервісом

FR14	Зареєстрований користувач повинен мати можливість створювати нові теми на форумі, обираючи відповідну категорію (догляд, здоров'я, поведінка тощо), додаючи заголовок та опис
FR15	Система повинна дозволяти відповідати на теми інших користувачів
FR16	Система повинна містити розділ з корисними статтями та порадами для власників домашніх тварин
FR17	Система повинна забезпечувати можливість пошуку статей за категоріями та ключовими словами
FR18	Система повинна забезпечувати можливість пошуку тем на форумі за категоріями та ключовими словами
FR19	Кожна публікація у стрічці повинна містити заголовок, короткий опис, повний текст, зображення, дату публікації та категорію
FR20	Користувач повинен мати змогу додавати до оголошення кілька фото. Система має забезпечувати перегляд доданих зображень у вигляді галереї з можливістю перемикання, відкриття в повному розмірі та вибору головного фото
FR21	Система обміну повідомленнями повинна відображати час надходження кожного повідомлення, а також час його перегляду адресатом
FR22	Інтерфейс платформи повинен відображати онлайн-статус автора оголошення біля його аватару в картці оголошення або на сторінці профілю
FR23	Система повинна забезпечувати авторизацію користувачів на основі JWT-токенів та ASP.NET Core Identity. Після успішного входу користувач отримує токен, який додається до заголовків запитів для доступу до захищених ресурсів

3.1.2. Опис нефункціональних вимог

Окрім функціональних можливостей, успішність будь-якого програмного продукту суттєво залежить від його нефункціональних

характеристик (табл. 3.2). Нефункціональні вимоги визначають, наскільки добре система повинна виконувати свої функції, охоплюючи такі аспекти, як продуктивність, надійність, безпека, зручність використання тощо [18].

Для вебсервісу «PetPlace» забезпечення якості, швидкодії та надійного захисту даних є критично важливим для довіри користувачів та стабільної роботи платформи.

Таблиця 3.2

Нефункціональні вимоги до вебзастосунку

Код вимоги	Опис вимоги
NFR1	Клієнтська частина повинна бути повністю сумісною з основними сучасними браузерами (Chrome, Opera, Safari, Edge)
NFR2	Повинні бути дотримані базові принципи доступності вебконтенту (WCAG) для користувачів з обмеженими можливостями
NFR3	Час завантаження основних сторінок вебсервісу (головна, оголошення, форум, стрічка статей) не повинен перевищувати 3 секунди для 95% користувачів при середньому інтернет-з'єднанні
NFR4	Час виконання пошукових запитів не повинен перевищувати 2 секунди для 90% запитів
NFR5	Повинні бути передбачені механізми резервного копіювання даних для запобігання їх втраті у випадку збоїв
NFR6	Доступ до захищених ресурсів повинен контролюватися через JWT токени, а дані користувачів передаватися лише через HTTPS
NFR7	Медіафайли мають зберігатися у хмарному сервісі Cloudinary, доступ до якого здійснюється за допомогою API-ключа та секрету, що зберігаються у змінних середовища і не доступні з клієнтського коду

NFR8	Повинні бути вжиті базові заходи для запобігання поширеним вебатакам (наприклад, валідація вхідних даних)
NFR9	Інтерфейс має бути адаптивним та коректно відображатися на пристроях з різними розмірами екранів

3.1.3. Діаграма прецедентів

Для наочного представлення функціональних вимог та візуалізації взаємодії користувачів з вебсервісом «PetPlace» було розроблено діаграму прецедентів (Use Case Diagram), що зображена на рис. 3.1. Цей інструмент мови UML дозволяє абстрагуватися від деталей реалізації та зосередитися на тому, що система надає своїм користувачам, через набір сценаріїв використання, що відображають очікувану поведінку програми [19].

У випадку «PetPlace» основний фокус діаграми зосереджений на ролі зареєстрованого користувача як основного актора, що взаємодіє з системою. Він може виконувати широкий спектр дій, що охоплюють перегляд стрічки корисних матеріалів, активну участь у форумах (включно зі створенням нових тем), налаштування власного профілю та вподобань, а також ключові операції маркетплейсу, такі як створення та перегляд оголошень чи можливість зв'язку з іншими користувачами для здійснення покупок та угод.

Для відображення складності та взаємозалежності на діаграмі застосовуються зв'язки *extend* та *include*. Зв'язок *extend* показує додаткові, опціональні можливості, які можуть розширювати базовий прецедент (наприклад, створення тем на форумі або застосування фільтрів під час перегляду оголошень). Натомість зв'язок *include* вказує на обов'язкові дії, які є невід'ємною частиною виконання основного прецеденту (наприклад, використання шаблону категорії при створенні оголошення). Такий підхід дозволяє чітко візуалізувати логіку системи та взаємозв'язки між її функціональними компонентами.

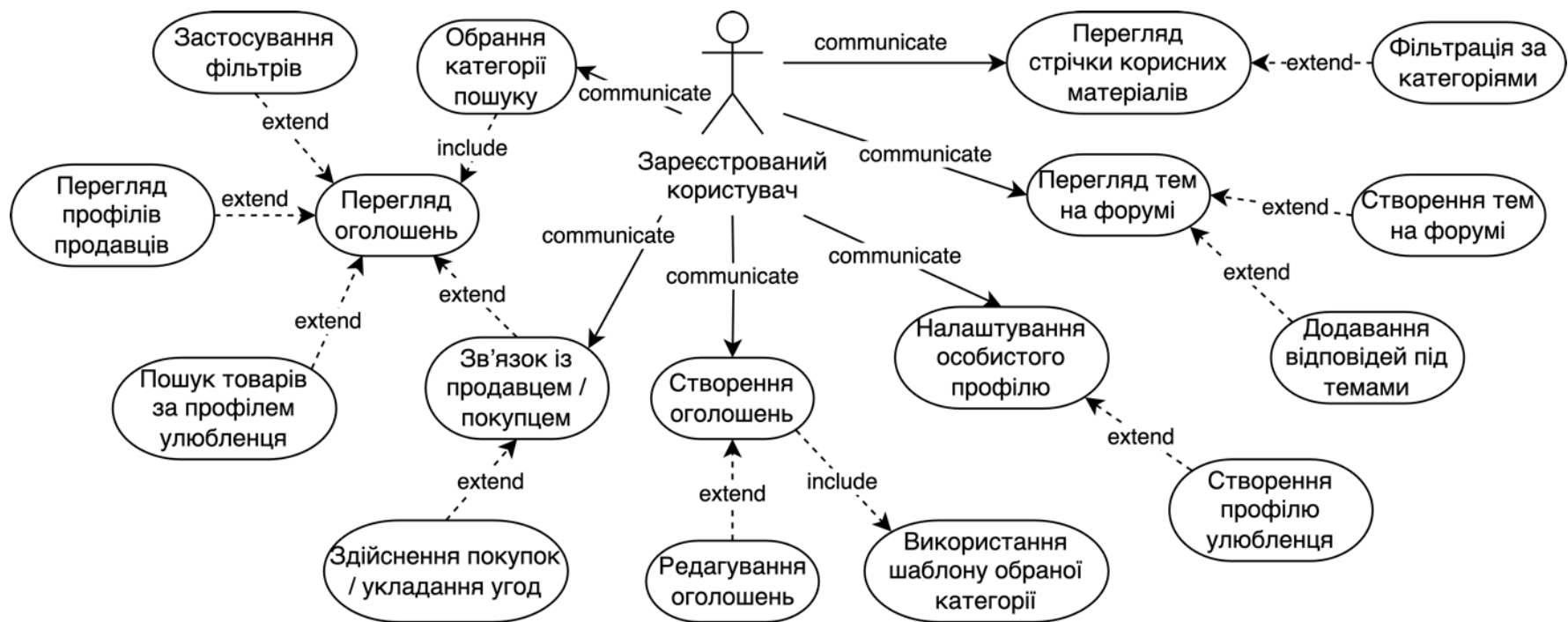


Рис. 3.1. UML-діаграма прецедентів

3.2. Структура бази даних

Щоб вебсервіс «PetPlace» працював злагоджено та міг ефективно керувати всією інформацією – потрібна чітка організація даних. Структура бази даних визначає, яким чином інформація про користувачів, їхніх тварин, оголошення, форумні обговорення та інші сутності буде організована та взаємопов'язана.

Однією з ключових особливостей розробленої моделі є застосування підходу Entity-Attribute-Value (EAV) для гнучкого управління атрибутами оголошень та профілів тварин. Такий підхід дозволяє зберігати динамічні характеристики об'єктів без необхідності створення окремих таблиць кожному типу, що особливо актуально у випадку великої кількості категорій із різними властивостями.

Нижче представлено опис логічної моделі бази даних, включаючи опис основних таблиць, їхніх атрибутів та зв'язків між ними.

Користувач (User) – зберігає основні облікові та контактні дані зареєстрованих користувачів платформи:

- 1) id: унікальний ідентифікатор користувача;
- 2) first_name: ім'я користувача;
- 3) last_name: прізвище користувача;
- 4) phone: номер телефону;
- 5) email: електронна адреса;
- 6) birth_date: дата народження;
- 7) gender: стать;
- 8) city: місто проживання;
- 9) country: країна проживання;
- 10) created_at: дата створення облікового запису;
- 11) last_active: дата останньої активності.

Оголошення (Ad) – базова сутність, що містить загальну інформацію про будь-яке оголошення незалежно від типу:

- 1) id: унікальний ідентифікатор оголошення;

- 2) user_id: дані користувача (User), який створив оголошення;
- 3) category_id: зовнішній ключ категорії оголошення (AdCategory);
- 4) title: заголовок оголошення;
- 5) description: опис;
- 6) price: ціна;
- 7) city: місто розміщення;
- 8) country: країна;
- 9) created_at: дата створення;
- 10) is_active: чи активне оголошення.

Тварина (Animal) – сутність-наслідник таблиці Ad, використовується для представлення оголошень про тварин із деталізацією за віком, породою, статтю тощо:

- 1) age_months: вік у місяцях;
- 2) sex: стать тварини;
- 3) weight_kg: вага у кг;
- 4) owner_type: тип власника (приватна особа, розплідник, притулок);
- 5) ad_type: тип оголошення (продаж, знайдено, в добрі руки тощо).

Товар (Product) – сутність-наслідник таблиці Ad, відповідає за збереження оголошень щодо товарів для тварин, включаючи корми, аксесуари, засоби догляду тощо:

- 1) brand: виробник товару;
- 2) condition: стан товару (новий, хороший, б/у);
- 3) animal_type: для якого типу тварини призначений товар.

Послуга (Service) – сутність-наслідник таблиці Ad, містить дані про оголошення з надання послуг для тварин: грумінг, вигул, дресирування тощо:

- 1) animal_type: тип тварини, для якої надається послуга;
- 2) duration_min: тривалість послуги у хвиликах;
- 3) availability: часова доступність (наприклад, «12:00–15:00»).

Категорія оголошень (AdCategory) – використовується для структуризації оголошень за категоріями й підкатегоріями:

- 1) id: унікальний ідентифікатор категорії;
- 2) parent_id: батьківська категорія (null, якщо верхній рівень);
- 3) name: назва категорії.

Фото оголошення (AdPhoto) – зберігає інформацію про прикріплені зображення до оголошень, включаючи посилання та статус модерації:

- 1) id: унікальний ідентифікатор фото;
- 2) ad_id: ідентифікатор оголошення (Ad);
- 3) url: посилання на зображення в Cloudinary;
- 4) is_main: позначення головного фото;
- 5) is_approved: статус модерації;
- 6) public_id: ідентифікатор у Cloudinary.

Обране (Favorite) – дозволяє користувачам додавати оголошення до списку обраного для швидкого доступу:

- 1) ad_id: ідентифікатор обраного оголошення (Ad);
- 2) user_id: дані користувача (User), який додав в обране.

Улюбленець (Pet) – зберігає дані з профілів домашніх тварин користувача:

- 1) id: унікальний ідентифікатор;
- 2) user_id: ідентифікатор власника (User);
- 3) animal_category_id: категорія тварини (AdCategory);
- 4) name: ім'я улюбленця;
- 5) birth_date: дата народження;
- 6) created_at: дата створення профілю;
- 7) photo_url: посилання на фото у Cloudinary;
- 8) photo_public_id: ідентифікатор фото у Cloudinary.

Атрибут (Attribute) – описує параметри, які можуть бути прив'язані до певної категорії оголошення або улюбленця (наприклад, розмір, колір):

- 1) id: унікальний ідентифікатор;

- 2) category_id: ідентифікатор категорії (AdCategory), до якої належить атрибут;
- 3) name: назва атрибута;
- 4) input_type: тип поля (наприклад, список, чекбокс).

Варіанти атрибутів (AttributeOption) – зберігає набір можливих значень для певного атрибута, що обираються під час створення оголошень або профілю тварини:

- 1) id: унікальний ідентифікатор;
- 2) attribute_id: ідентифікатор відповідного атрибута (Attribute);
- 3) value: текст значення.

Значення атрибутів оголошення (AdAttributeValue) – дозволяє закріплювати конкретні значення атрибутів за оголошенням:

- 1) id: унікальний ідентифікатор;
- 2) ad_id: ідентифікатор оголошення (Ad);
- 3) attribute_id: ідентифікатор атрибута (Attribute);
- 4) option_id: обране значення (AttributeOption).

Значення атрибутів улюбленця (PetAttributeValue) – зберігає обрані параметри тварини з переліку доступних атрибутів та їх значень:

- 1) id: унікальний ідентифікатор;
- 2) pet_id: дані улюбленця (Pet);
- 3) attribute_id: ідентифікатор атрибута (Attribute);
- 4) option_id: значення, вибране для атрибута (AttributeOption).

Категорія публікацій (PostCategory) – групує публікації за тематикою (догляд, поради, здоров'я тощо):

- 1) id: унікальний ідентифікатор;
- 2) name: назва категорії.

Публікація (Post) – містить корисні матеріали та статті, пов'язані з темою домашніх тварин:

- 1) id: унікальний ідентифікатор;
- 2) category_id: ідентифікатор категорії (PostCategory);

- 3) image_url: посилання на зображення;
- 4) image_public_id: ідентифікатор фото у Cloudinary;
- 5) title: заголовок;
- 6) description: короткий опис;
- 7) content: повний текст;
- 8) is_featured: чи показується у топі.

Категорія тем (ThreadCategory) – класифікує теми на форумі для зручної навігації:

- 1) id: унікальний ідентифікатор;
- 2) name: назва категорії тем.

Тема форуму (Thread) – відповідає за створення нових тем на форумі з описом проблеми або запитанням:

- 1) id: унікальний ідентифікатор;
- 2) user_id: ідентифікатор автора (User);
- 3) category_id: ідентифікатор категорії (ThreadCategory);
- 4) time_created: час створення;
- 5) subject: тема;
- 6) content: зміст питання;
- 7) replies_count: кількість відповідей.

Відповідь на тему (ThreadReply) – зберігає коментарі та відповіді користувачів на обрані теми форуму:

- 1) id: унікальний ідентифікатор;
- 2) thread_id: зовнішній ключ теми (Thread);
- 3) user_id: дані користувача, що відповідає (User);
- 4) time_created: час відповіді;
- 5) content: текст відповіді.

Повідомлення (Message) – зберігає дані про особисті повідомлення між користувачами:

- 1) id: унікальний ідентифікатор;
- 2) sender_id: дані відправника (User);

- 3) recipient_id: дані отримувача (User);
- 4) content: текст повідомлення;
- 5) time_sent: час відправлення;
- 6) time_read: час прочитання;
- 7) sender_deleted: чи видалив відправник;
- 8) recipient_deleted: чи видалив отримувач.

Після аналізу таблиць бази даних можна зробити висновок, що структура зберігає чітку логіку організації інформації відповідно до потреб платформи.

Усі зв'язки між сутностями мають тип «один-до-багатьох». Наприклад, один користувач може створити багато оголошень, але кожне оголошення належить лише одному користувачу; кожне оголошення може мати декілька фото, проте кожне фото прив'язане до одного оголошення.

Зв'язки в моделі EAV також підтримують цю логіку: один атрибут може мати багато варіантів значень, а кожне оголошення або улюбленець може містити багато прив'язаних атрибутів. Цей підхід забезпечує гнучкість у зберіганні розширюваних даних без потреби створення окремих таблиць під кожну категорію чи вид тварин. Така модель особливо корисна в системах, де атрибути можуть суттєво відрізнятися залежно від типу об'єкта.

Крім того, у структурі реалізовано спадкування сутностей оголошень: Animal, Product та Service успадковують поля від базової сутності Ad, використовуючи підхід complete and disjoint [20], що гарантує, що кожне оголошення обов'язково належить до одного і тільки одного з визначених підкласів і при цьому не може існувати як самостійний екземпляр базового класу.

Для кращої візуалізації на рис. 3.2 представлено ER-діаграму, яка ілюструє основні сутності бази даних, їх атрибути та взаємозв'язки.

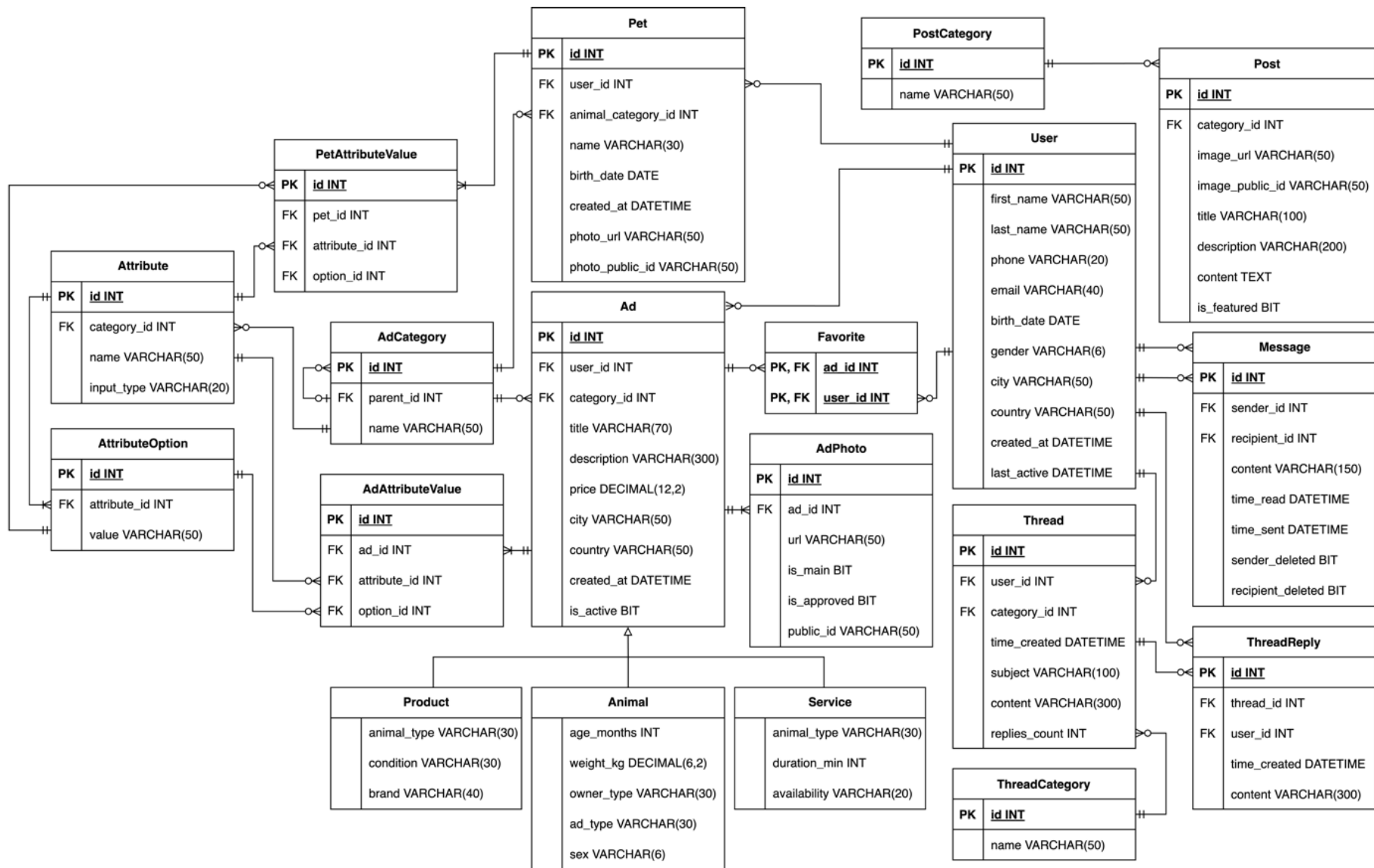


Рис. 3.2. ER-діаграма бази даних

3.3. Архітектура вебзастосунку

Архітектура вебзастосунку «PetPlace» побудована за принципами багаторівневої клієнт-серверної моделі з чітким розділенням відповідальності між клієнтською частиною, серверною логікою та базою даних. Такий підхід забезпечує масштабованість, зручність розроблення, обслуговування та подальший розвиток проєкту.

Клієнтська частина реалізована за допомогою фреймворку Angular, що дозволяє створювати динамічний односторінковий інтерфейс користувача та швидко реагувати на дії користувача без потреби повного перезавантаження сторінки. Angular забезпечує зручну реалізацію маршрутизації, форм, а також інтеграцію з RESTful API та SignalR.

Серверна частина створена на базі ASP.NET Core, який забезпечує високу продуктивність, безпеку та підтримку широкого спектру функціональних можливостей. Обмін даними між фронтендом і сервером здійснюється через стандартні HTTP-запити, а також у режимі реального часу – через бібліотеку SignalR. Вона встановлює постійне двостороннє з'єднання, що дозволяє серверу миттєво передавати оновлення клієнтам без необхідності постійних запитів.

Робота з базою даних реалізована за допомогою Entity Framework Core – ORM-інструменту, який дає змогу працювати з базою даних через об'єктно-орієнтовану модель. Це значно спрощує взаємодію з базою та дозволяє виконувати типові CRUD-операції без написання SQL-запитів вручну. Як СКБД обрано Microsoft SQL Server, що розгортається у контейнеризованому середовищі Docker для забезпечення портативності та ізольованості.

Зберігання медіаконтенту, такого як фотографії тварин та оголошень, здійснюється у хмарному сховищі Cloudinary. Взаємодія з Cloudinary відбувається з серверної частини через захищений API, використовуючи API-key та API-secret. Ці облікові дані зберігаються в безпечному місці на сервері, що запобігає їхньому потраплянню на клієнтську сторону та

несанкціонованому використанню. Серверна частина відповідає за завантаження медіафайлів до Cloudinary та отримання URL-адрес для їх відображення клієнту, забезпечуючи таким чином безпечне керування медіаконтентом.

Таким чином, архітектура системи забезпечує ефективний розподіл навантаження, надійну комунікацію між компонентами, а також підтримку функціональності, необхідної для реалізації ключових можливостей платформи – роботи з оголошеннями, користувачами, чатами, профілями тварин, форумом, стрічкою матеріалів тощо.

Нижче наведено узагальнену схему архітектури вебзастосунку (рис. 3.3), яка демонструє взаємодію між клієнтом, сервером, базою даних та хмарною інфраструктурою.

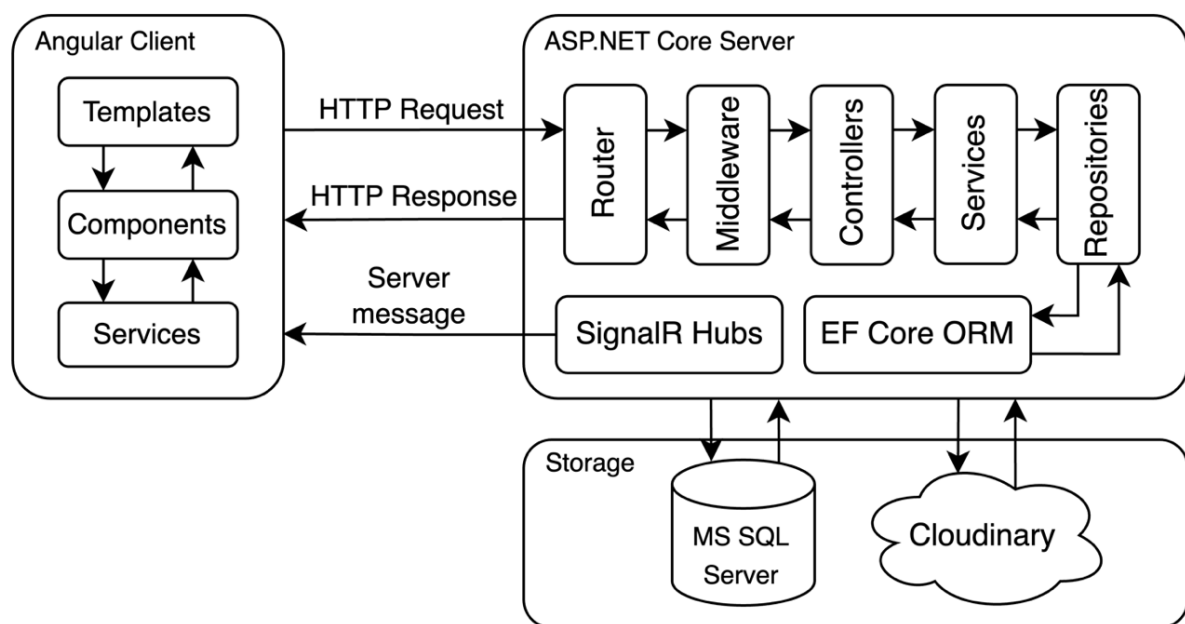


Рис. 3.3. Узагальнена архітектура системи

3.4. Висновки до розділу 3

У цьому розділі було детально розглянуто основні етапи розроблення вебзастосунку «PetPlace», зосереджуючи увагу на вимогах до системи, структурі бази даних та архітектурних рішеннях. На основі аналізу предметної області були сформульовані функціональні та нефункціональні

вимоги, які відображають ключові сценарії використання платформи та очікувану якість її роботи.

Було спроектовано структуру бази даних із використанням механізмів спадкування, що дозволяє ефективно зберігати оголошення різних типів (тварини, товари, послуги) з урахуванням їхніх специфічних атрибутів. Для динамічного керування фільтрами та властивостями об'єктів застосовано EAV-модель, яка забезпечує гнучкість і масштабованість. Додатково представлено ER-діаграму, яка візуалізує взаємозв'язки між сутностями бази.

Опис архітектури системи засвідчив доцільність застосування багаторівневого підходу із розподілом на клієнтську, серверну частини та СКБД. Використання Angular, ASP.NET Core, EF Core, SignalR, Microsoft SQL Server та хмарного сховища Cloudinary дозволяє забезпечити масштабованість, інтерактивність, стабільну роботу в режимі реального часу та зручність у подальшому обслуговуванні платформи.

Таким чином, результати, отримані в цьому розділі, закладають основу для практичної реалізації сучасного, функціонального та зручного вебзастосунку.

4. АНАЛІЗ РОЗРОБЛЕНОГО ВЕБСЕРВІСУ

У цьому розділі розглядатимуться результати перевірки роботи розробленого вебзастосунку «PetPlace», що охоплює оцінку його функціональності, логіки взаємодії з користувачем, технічної коректності та надійності. Особлива увага приділяється дослідженню користувацького інтерфейсу, а також проведенню серії тестів для підтвердження відповідності вимогам та стабільності роботи всіх реалізованих можливостей.

Здійснення такого аналізу є необхідним етапом життєвого циклу програмного забезпечення, оскільки дозволяє виявити слабкі місця в реалізації, оцінити загальну якість розробки та сформулювати пропозиції та бачення щодо майбутнього вдосконалення. У межах розділу проводиться огляд логіки побудови інтерфейсу, перевірка функціональних сценаріїв та димове тестування системи.

Таким чином, основними завданнями розділу є аналіз реалізованого інтерфейсу, тестування окремих модулів вебзастосунку, а також визначення можливих шляхів покращення системи на основі отриманих результатів.

4.1. Опис реалізованого користувацького інтерфейсу

Однією з ключових складових сучасного вебзастосунку є зручний, зрозумілий та інтуїтивний інтерфейс, який забезпечує ефективну взаємодію користувача з системою. У межах реалізації платформи «PetPlace» було приділено значну увагу саме користувацькому інтерфейсу, адже основна аудиторія сервісу – звичайні користувачі без спеціальної технічної підготовки, які мають потребу швидко знаходити необхідну інформацію або розміщувати власні оголошення.

Інтерфейс розроблено з урахуванням принципів UX/UI-дизайну: збереження логічної структури, візуальна ієрархія, а також зручна навігація між основними розділами сайту. Окрему увагу приділено створенню

зрозумілої системи фільтрації оголошень, що адаптується до кожної категорії (тварини, товари, послуги), а також впровадженню функціональності персоналізованих профілів тварин.

У цьому пункті наведено опис деяких ключових інтерфейсних сторінок вебсервісу, зокрема сторінки оголошень для тварин, сторінки реєстрації та авторизації, розділу форуму, а також сторінки перегляду оголошення з вкладкою листування. Для кожної з них детально описано основні компоненти, логіку розміщення елементів та їх цільове призначення.

Сторінка оголошень «Animal» (Тварини) – ключовий елемент застосунку, що дозволяє користувачам знаходити улюбленців за різноманітними критеріями. Її загальний дизайн інтуїтивно зрозумілий, з чітким розподілом функціональних блоків. У верхній частині сторінки розташована навігаційна панель, що забезпечує швидкий доступ до відповідних розділів сайту, справа від посилань, відображається іконка «вибране» що дозволяє перейти до особистого списку уподобань. Наявна кнопка «Post Ads» (Опублікувати оголошення), яка дозволяє користувачам додавати власні оголошення. Поруч розташована іконка профілю користувача з привітанням, що вказує на авторизацію користувача та доступ до особистого кабінету.

На сторінці оголошень (рис. 4.1), під навігаційною панеллю, розташована горизонтальна панель основних фільтрів. За її допомогою користувач може встановити діапазон цін, обрати стать, вік, вагу та тип власника. Доступні кнопки для застосування фільтрів, скасування всіх налаштувань фільтрації та кнопка сортування результатів. Ліворуч від результатів пошуку знаходиться розширена панель фільтрів. Вона пропонує унікальні фільтри для кожної категорії тварин. У верхній частині цієї панелі розміщені іконки для вибору типу тварини (собака, кіт, пташка, риба, рептилія), що відповідає вимозі щодо широкої категоризації. Нижче розташовані самі фільтри, адаптовані до обраного типу тварини.

Ця розширена система фільтрів реалізує можливість застосування унікальних критеріїв пошуку для кожної категорії оголошень. Основну частину сторінки, праворуч від детальних фільтрів, займає область відображення результатів пошуку. Оголошення представлені у вигляді карток, розташованих сіткою. Кожна картка містить фотографію тварини, її кличку або опис породи, місцезнаходження та ціну. Є можливість додати оголошення до списку обраних. Під списком оголошень знаходиться панель посторінкового виведення, що дозволяє користувачеві легко переміщатися між сторінками результатів пошуку.

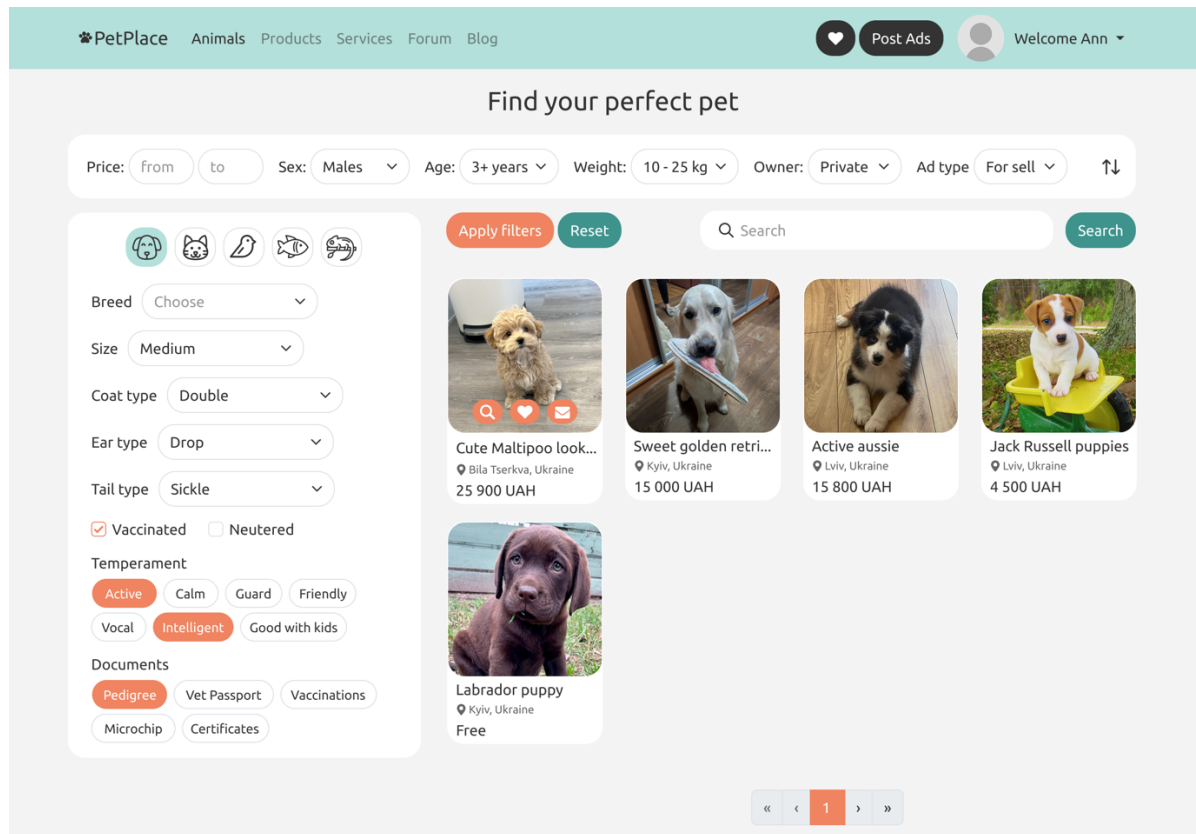


Рис. 4.1. Сторінка оголошень для тварин

Сторінка «Sign up» (Реєстрація) на рис. 4.2 є центральною точкою входу для нових користувачів, які бажають отримати доступ до всіх можливостей платформи. У верхній частині екрана розташовані поля «E-mail» та «Password», а також кнопки «Login» (Увійти) та «Register» (Зареєструватися). Таке розміщення забезпечує швидкий доступ для вже

zareestrovanih korystuvachiv, dozvoliaючи їм avtorizuvatisia, i vodnochас підкреслює можливість реєстрації для нових відвідувачів.

Центральну частину екрану займає форма реєстрації, призначена для введення особистих даних, необхідних для створення облікового запису. Ці поля дозволяють користувачеві ввести таку інформацію, як ім'я, прізвище, електронна пошта, номер телефону, дата народження, місто та країна проживання. У нижній частині форми розташовані дві основні кнопки. Кнопка «Register» (Зареєструватися) слугує для підтвердження введених даних та створення нового облікового запису, надаючи користувачеві доступ до всіх функцій платформи. Кнопка «Cancel» (Скасувати) дозволяє відмінити процес реєстрації та повернутися до попередньої сторінки або основного інтерфейсу застосунку.

The image shows a web application interface for 'PetPlace'. The top navigation bar is teal and contains links to 'Animals', 'Products', 'Services', 'Forum', and 'Blog'. On the right side of this bar are input fields for 'E-mail' and 'Password', along with 'Login' and 'Register' buttons. The central part of the page is a white 'Sign up' form. It starts with a title 'Sign up' in orange. Below it is a gender selection section: 'I am a: ☒ Male ☐ Female'. This is followed by seven input fields: 'First Name', 'Last Name', 'Email', 'Phone', 'Date of birth', 'City', and 'Country'. At the bottom of the form are two buttons: a teal 'Register' button and a grey 'Cancel' button.

Рис. 4.2. Сторінка реєстрації

Сторінка «Forum» (Форум) є ключовим розділом застосунку «PetPlace», що слугує центральним майданчиком для обговорення та обміну інформацією між користувачами щодо догляду за домашніми тваринами. У

верхній частині екрана розміщена кнопка «Ask a Question» (Задати питання), що дозволяє користувачам розпочати нове обговорення на форумі. Ліворуч на сторінці розміщена панель «Categories» (Категорії), що дозволяє користувачам фільтрувати обговорення за тематичними розділами (наприклад, усі обговорення, тренування, годування, породи та здоров'я).

Основну частину сторінки (рис. 4.3) займає область відображення різноманітних обговорень, представлених у вигляді окремих карток; кожна з яких містить аватар користувача, який створив тему, заголовок обговорення, перші рядки повідомлення, що надають контекст обговорення, інформацію про кількість відповідей та час публікації, а також тематичні теги.

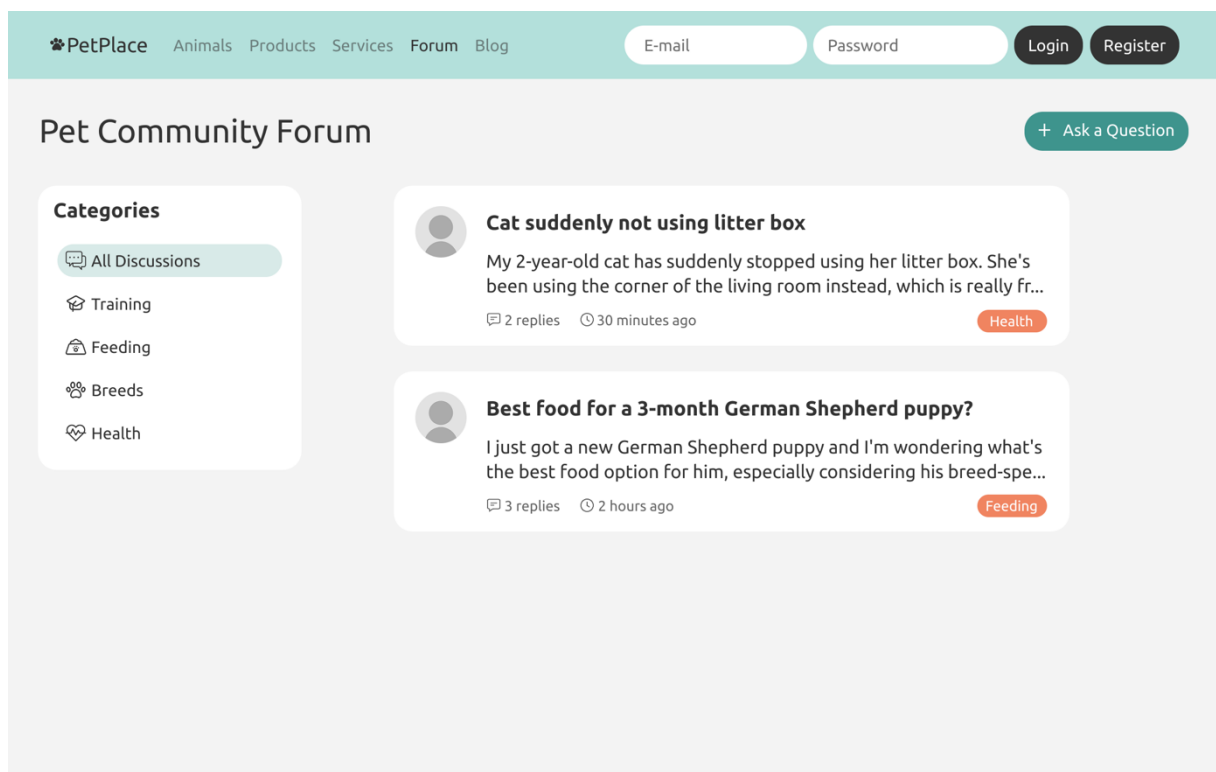


Рис. 4.3. Сторінка тематичного форуму спільноти

Сторінка оголошення з вкладкою «Messages» (Повідомлення) на рис. 4.4 дозволяє користувачам взаємодіяти з автором оголошення та переглядати ключову інформацію про саме оголошення. Ліворуч розташований основний блок з інформацією про тварину, що містить велике

фото та базові відомості про тварину. Нижче можна знайти аватар й інформацію про автора оголошення та час його останньої активності. Поряд розміщені дві кнопки – іконка серця (для додавання до обраних, якщо це ще не було зроблено) та кнопка «Message» (Повідомлення), що дозволяє перейти на потрібну вкладку та зв'язатися з автором. Праворуч від цього блоку розташований основний блок, розділений на три вкладки: «Ad details» (Деталі оголошення) з детальними характеристиками, «Photos» (Фотографії) з галереєю зображень та «Messages» (Повідомлення).

Під вкладками продемонстровано діалог листування, який містить вихідне повідомлення від поточного користувача із зазначенням часу відправлення та його статусу, а також поле для введення повідомлення та кнопка «Send» (Надіслати) для відправлення відповіді.

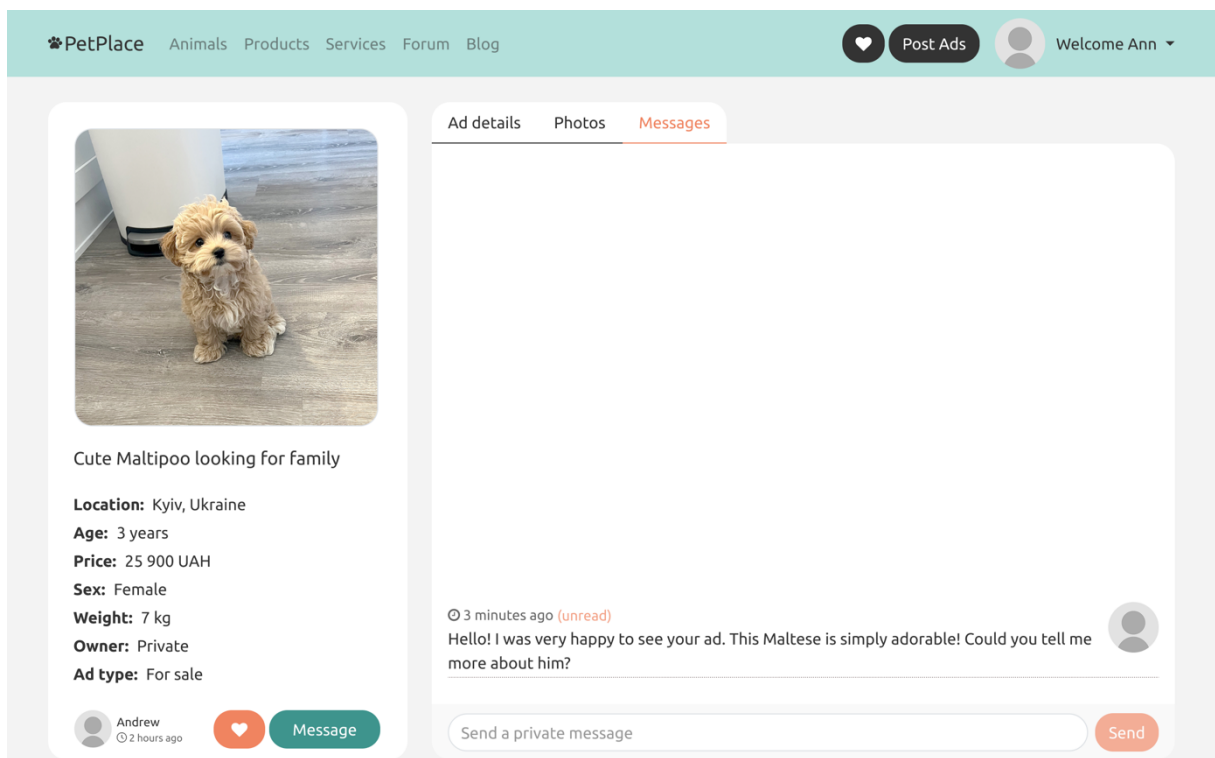


Рис. 4.4. Сторінка листування та картки оголошення

4.2. Тестування розробленого вебзастосунку

З метою забезпечення базової працездатності реалізованого вебсервісу було проведено димне тестування (Smoke Testing). Цей тип

тестування полягає у швидкій перевірці найважливіших можливостей системи з метою виявлення критичних помилок, які можуть завадити подальшому тестуванню або використанню продукту [21].

Димне тестування зазвичай виконується після завершення розробки або основних етапів інтеграції, щоб переконатися, що застосунок запускається, основні модулі взаємодіють належним чином, а ключові сценарії роботи (авторизація, створення та пошук оголошень, взаємодія з користувачами тощо) функціонують без збоїв. Під час перевірки особливу увагу було приділено коректності навігації, завантаженню сторінок та відсутності критичних помилок при базовій взаємодії з інтерфейсом.

У межах цього етапу було складено набір тестових випадків (табл. 4.1), що охоплюють базову функціональність вебзастосунку «PetPlace».

Таблиця 4.1

Тестові випадки димового тестування

№	Код вимоги	Сценарій дій	Очікуваний результат
1	FR1	1. Користувач натискає кнопку на навігаційній панелі «Створити оголошення». 2. Обирає категорію оголошень: «Тварина», «Товар» або «Послуга». 3. Заповнює загальні поля для обраної категорії (заголовки, опис, ціна, тощо). 4. Система динамічно підвантажує додаткові поля відповідно до підкатегорії. 5. Користувач заповнює динамічні поля. 6. Натискає «Опублікувати».	Оголошення успішно створене та відображається в загальній сітці

Продовження табл. 4.1

2	FR2	1. Користувач переходить у «Профіль». 2. Обирає розділ "Мої улюбленці". 3. Натискає «Додати улюбленця». 4. Вказує тип тварини, ім'я, дату народження тощо. 5. Система динамічно підтягує доступні атрибути (порода, розмір, темперамент тощо) відповідно до типу тварини. 6. Користувач заповнює додаткові нові поля. 7. Натискає кнопку «Зберегти».	Улюбленець збережений у профілі, відображається на сторінці користувача
3	FR3	1. Користувач відкриває сторінку з оголошеннями. 2. Обирає категорію «Тварини». 3. Вибирає параметри фільтрації (вік, стать, темперамент тощо).	Список оголошень оновлюється відповідно до обраних фільтрів
4	FR8	1. Користувач переглядає оголошення. 2. Натискає іконку «Додати до улюблених». 3. Переходить у розділ «Улюблені».	Обране оголошення зберігається, відображається в розділі «Улюблені»
5	FR5	1. Користувач при створенні оголошення натискає «Додати фото». 2. Завантажує одне або кілька зображень із пристрою. 3. Обирає головне фото. 4. Натискає «Опублікувати».	Зображення збережені у хмарному сховищі (Cloudinary), відображаються в оголошенні
6	FR6	1. Користувач переходить до списку своїх оголошень у «Профілі». 2. Натискає «Редагувати» на активному оголошенні. 3. Змінює опис або ціну. 4. Натискає «Зберегти зміни».	Оголошення успішно оновлено

7	FR12	1. Користувач відкриває картку оголошення. 2. Натискає «Написати повідомлення». 3. Автоматично переходить на профіль автора оголошення в розділ «Повідомлення». 4. Вводить текст. 5. Натискає «Надіслати».	Повідомлення доставлено. У чаті зберігається історія переписки
8	FR23	1. Користувач відкриває сторінку входу. 2. Вводить електронну пошту та пароль. 3. Натискає «Увійти». 4. Система перевіряє облікові дані, створює JWT-токен.	Користувач авторизований, запити до захищених ресурсів містять токен
9	FR16	1. Користувач переходить у розділ «Корисні матеріали». 2. Вибирає категорію «Здоров'я». 3. Обирає статтю для перегляду.	Відкривається повна версія публікації з усіма її елементами
10	FR15	1. Користувач відкриває тему на форумі. 2. Вводить текст відповіді. 3. Натискає «Надіслати».	Відповідь з'являється в темі нижче основного посту або попередньої відповіді
11	FR14	1. Користувач переходить у розділ «Форум». 2. Обирає категорію «Догляд». 3. Вказує заголовок, опис. 4. Натискає «Створити тему».	Нова тема створена та доступна для перегляду іншим користувачам

Після проведення димного тестування можна переконатися, що основна функціональність вебзастосунку «PetPlace» працює коректно. Усі ключові сценарії, що охоплюють дії з оголошеннями, управління профілем, взаємодію з улюбленцями, систему повідомлень, форум та публікації, були успішно перевірені. Це свідчить про загальну стабільність розгорнутої

версії застосунку та дає змогу перейти до наступних етапів тестування, спрямованих на більш детальну перевірку та виявлення можливих помилок.

4.3. Рекомендації щодо подальшого вдосконалення

Розроблений вебсервіс «PetPlace» на поточному етапі реалізує основну функціональність, необхідну для публікації оголошень, створення профілів тварин, взаємодії користувачів через форум, зручного пошуку товарів і послуг та перегляду стрічки корисних матеріалів. Водночас, враховуючи результати тестування, сучасні вимоги до масштабованих систем та відгуки користувачів схожих платформ, можна сформулювати ряд рекомендацій щодо подальшого вдосконалення системи:

1. Розширення функціональності профілів тварин: додати можливість ведення історії вакцинацій, медичних процедур, рекомендацій по догляду та автоматичної генерації нагадувань про вакцинацію або прийом ліків.
2. Дошка оголошень про зниклих/знайдених тварин: реалізація окремого розділу для розміщення та перегляду оголошень про зниклих або знайдених тварин, що створить централізоване місце для пошуку, на відміну від окремих сторінок у соцмережах для кожного міста, що значно полегшить процес повернення тварин до власників.
3. Персоналізовані рекомендації: впровадження системи рекомендацій на основі машинного навчання для підбору товарів, послуг та статей згідно з профілем тварини, попередніми вподобаннями користувача або історією активності.
4. Система бронювання послуг: впровадження функціональності онлайн-бронювання для ветеринарів, грумерів, кінологів та інших постачальників послуг, що спростить процес запису на прийом та керування розкладом.

5. Модерація та верифікація користувачів: реалізувати механізми підтвердження особи продавця або надавача послуг для підвищення довіри між учасниками платформи, а також систему скарг і рейтингів.
6. Адмін-панель для управління контентом: створення окремого зручного адміністративного інтерфейсу для керування користувачами, оголошеннями, категоріями, статтями, форумом та фільтрами.
7. Монетизація та підтримка ініціатив: впровадження можливості прийому оплат за преміум-оголошення, платні послуги або донати на користь притулків.
8. Інтеграція доставки та безпечних платежів (за прикладом OLX): впровадження співпраці зі службами доставки та системи безпечної онлайн-оплати, аналогічно функціональності "OLX Доставка", для зручних і захищених транзакцій та мінімізації шахрайства.
9. Розширення на інші ринки: після успішного запуску в межах України – адаптація платформи для використання в інших країнах, із локалізацією інтерфейсу, валют і категорій.

Впровадження зазначених поліпшень дозволить значно підвищити якість користувацького досвіду, розширити цільову аудиторію, підвищити конкурентоспроможність проєкту та сформувати повноцінну екосистему для власників і любителів тварин.

4.4. Висновки до розділу 4

У даному розділі було проведено всебічний аналіз розробленого вебзастосунку «PetPlace». Зокрема, розглянуто структуру реалізованого інтерфейсу, що орієнтований на зручність користувачів, зокрема швидкий пошук оголошень за допомогою динамічних фільтрів, реєстрацію та авторизацію, доступ до форуму спільноти та зв'язок з авторами оголошень.

Детально описано ключові елементи кожної сторінки, що дозволяють ефективно взаємодіяти з функціональністю платформи.

Проведене тестування показало, що основні модулі працюють коректно, застосунок стабільний у роботі, а навігація є інтуїтивно зрозумілою. При цьому було виявлено низку незначних недоліків і потенційних точок росту, які можуть бути враховані при подальшому розвитку системи.

У результаті аналізу сформульовано перелік практичних рекомендацій щодо вдосконалення платформи, зокрема: персоналізовані рекомендації, покращення пошукових можливостей, впровадження системи рейтингів, верифікації користувачів та платіжної інфраструктури.

Отже, вебзастосунок «PetPlace» заклав надійне підґрунтя для подальшого розвитку повноцінного сервісу, що здатен об'єднати власників тварин, продавців, притулків та інших учасників зооспільноти на єдиній спеціалізованій платформі.

ВИСНОВКИ

Метою даного дипломного проєкту було розроблення багатофункціонального вебсервісу «PetPlace», покликаного створити централізовану онлайн-платформу для власників домашніх тварин. Цей сервіс має надати користувачам зручні інструменти для пошуку та розміщення оголошень про тварин, товари та послуги, а також забезпечити активну взаємодію в рамках спільноти через форум та корисні матеріали. Проєкт був ініційований після аналізу існуючих рішень на ринку, який виявив потребу в більш комплексному та інтегрованому сервісі, що поєднує функціональність маркетплейсу та соціальної взаємодії.

На етапі проєктування було ретельно обґрунтовано вибір технологічного стеку. Для серверної частини, що відповідає за бізнес-логіку та взаємодію з базою даних, було обрано C# та фреймворк ASP.NET Core. Цей вибір зумовлений високою продуктивністю, надійністю, можливостями масштабування та зручною інтеграцією з корпоративними рішеннями. Як основна система керування базами даних, що забезпечує зберігання та управління всією інформацією платформи, було визначено Microsoft SQL Server. Вирішальними стали його переваги у забезпеченні цілісності даних, потужність для виконання складних реляційних запитів та зріла екосистема. Для розробки клієнтської частини, що відповідає за користувацький інтерфейс та досвід, було обрано фреймворк Angular. Його структурована архітектура, використання TypeScript та багатий набір вбудованих інструментів забезпечують ефективну розробку складних та масштабованих інтерфейсів.

Під час реалізації вебсервісу «PetPlace» було впроваджено ключові функціональні можливості, що охоплюють реєстрацію та автентифікацію користувачів, розширену систему оголошень з гнучкими фільтрами (реалізованими за моделлю Entity–Attribute–Value), інтерактивний форум для обговорень, можливість створення детальних профілів домашніх

тварин, а також доступ до корисних матеріалів. Архітектура системи була побудована за принципами багаторівневого підходу, що забезпечило чіткий поділ відповідальності між компонентами та сприятиме гнучкості у подальшому розвитку. Логічна структура бази даних, представлена у вигляді ER-діаграми, відображає всі ключові сутності та їхні взаємозв'язки, оптимізовані для ефективного зберігання та отримання даних.

Розроблений вебсервіс відповідає висунутим функціональним та нефункціональним вимогам. Проведене тестування, що включало функціональну перевірку, оцінку користувацького інтерфейсу та базову перевірку продуктивності, підтвердило коректність роботи всіх реалізованих можливостей та загальну стабільність застосунку.

Проект «PetPlace» має значне практичне значення, надаючи власникам домашніх тварин зручний та інтегрований інструмент для взаємодії, пошуку та обміну інформацією. Гнучка архітектура та обраний технологічний стек створюють міцну основу для подальшого розвитку та розширення функціональності, що може включати інтеграцію платіжних систем, поглиблену оптимізацію продуктивності, персоналізований пошук та адміністративної панелі.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. What are the global pet ownership trends? [Електронний ресурс]. / Francis Galibert – Режим доступу: <https://www.sciencedirect.com/science/article/pii/S1631069110003008> – Дата доступу: 01.02.2025 р.
2. Mans Best Friend: Global Pet Ownership and Feeding Trends [Електронний ресурс]. – Режим доступу: <https://nielseniq.com/global/en/insights/report/2016/mans-best-friend-global-pet-ownership-and-feeding-trends/> – Дата доступу: 03.02.2025 р.
3. Global Trends in the Pet Population [Електронний ресурс]. – Режим доступу: <https://healthforanimals.org/reports/pet-care-report/global-trends-in-the-pet-population/#ownership> – Дата доступу: 06.02.2025 р.
4. Дослідження: Як війна вплинула на притулки для тварин [Електронний ресурс]. – Режим доступу: <https://www.savepetsofukraine.kormotech.com/post/дослідження-як-війна-вплинула-на-притулки-для-тварин> – Дата доступу: 07.02.2025 р.
5. OLX.ua [Електронний ресурс]. – Режим доступу: <https://www.olx.ua/uk/> – Дата доступу: 09.02.2025 р.
6. Є-Пет [Електронний ресурс]. – Режим доступу: <https://e-pet.com.ua> – Дата доступу: 14.02.2025 р.
7. Лара.shop [Електронний ресурс]. – Режим доступу: <https://lara.shop> – Дата доступу: 17.02.2025 р.
8. A tour of C# [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/overview> – Дата доступу: 25.02.2025 р.
9. ASP.NET [Електронний ресурс]. – Режим доступу: <https://dotnet.microsoft.com/en-us/apps/aspnet> – Дата доступу: 26.02.2025 р.
10. Asynchronous Programming in PHP [Електронний ресурс]. – Режим доступу: <https://phpconference.com/blog/asynchronous-programming-in->

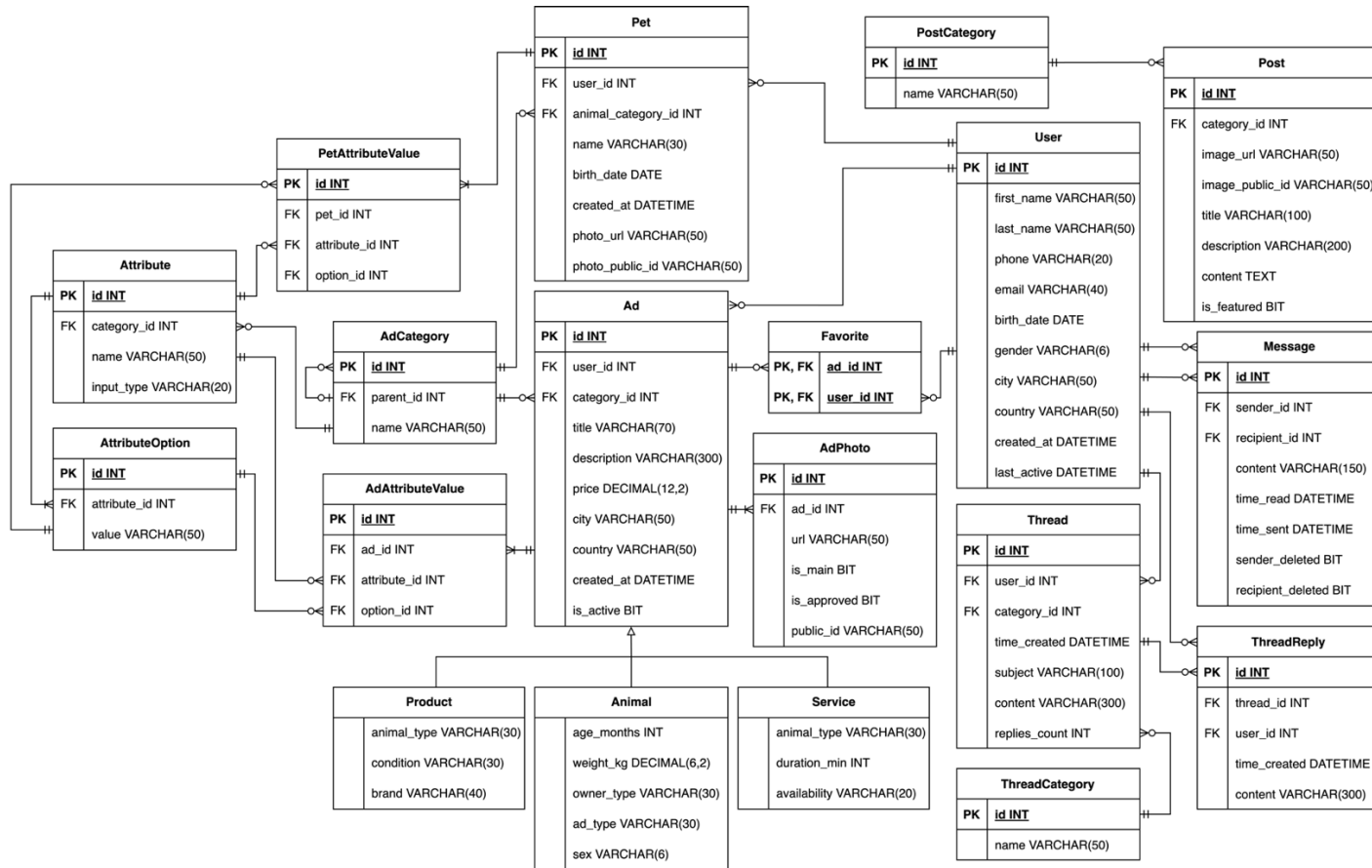
php/#:~:text=The%20Synchronous%20nature%20of%20PHP&text=This%20is%20because%20the%20built,before%20the%20next%20one%20starts – Дата доступа: 28.02.2025 г.

11. Introduction to Django [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/enUS/docs/Learn_web_development/Extensions/Server-side/Django/Introduction – Дата доступа: 02.03.2025 г.
12. MS SQL Server: History and Advantages [Электронный ресурс]. – Режим доступа: <https://bytescout.com/blog/2014/09/ms-sql-server-history-and-advantages.html> – Дата доступа: 13.03.2025 г.
13. Why use MongoDB? [Электронный ресурс]. – Режим доступа: <https://www.mongodb.com/resources/products/fundamentals/why-use-mongodb> – Дата доступа: 14.03.2025 г.
14. Exploring the Advantages of Angular [Электронный ресурс]. – Режим доступа: <https://medium.com/@rictorres.uyu/exploring-the-advantages-of-angular-1695d35885eb> – Дата доступа: 19.03.2025 г.
15. Why we use React.js in the development [Электронный ресурс]. – Режим доступа: <https://www.techmagic.co/blog/why-we-use-react-js-in-the-development> – Дата доступа: 20.03.2025 г.
16. Vue.js Guide: Introduction [Электронный ресурс]. – Режим доступа: <https://vuejs.org/guide/introduction> – Дата доступа: 21.03.2025 г.
17. Functional Requirements [Электронный ресурс]. – Режим доступа: <https://www.nuclino.com/articles/functional-requirements> – Дата доступа: 25.03.2025 г.
18. Non-functional Requirements: Examples, Types, Approaches [Электронный ресурс]. – Режим доступа: <https://training.qatestlab.com/blog/technical-articles/non-functional-requirements-examples-types-approaches/> – Дата доступа: 26.03.2025 г.
19. Use Case Diagrams [Электронный ресурс]. – Режим доступа: <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case> – Дата доступа: 27.03.2025 г.

20. Don't get wrong: Explained Guide to Choosing a Database Design Notation for ERD in a while [Электронный ресурс]. – Режим доступа: <https://medium.com/@ericgcc/dont-get-wrong-explained-guide-to-choosing-a-database-design-notation-for-erd-in-a-while-7747925a7531> – Дата доступа: 09.04.2025 г.
21. Smoke testing (software) [Электронный ресурс]. – Режим доступа: [https://en.wikipedia.org/wiki/Smoke_testing_\(software\)](https://en.wikipedia.org/wiki/Smoke_testing_(software)) – Дата доступа: 14.04.2025 г.

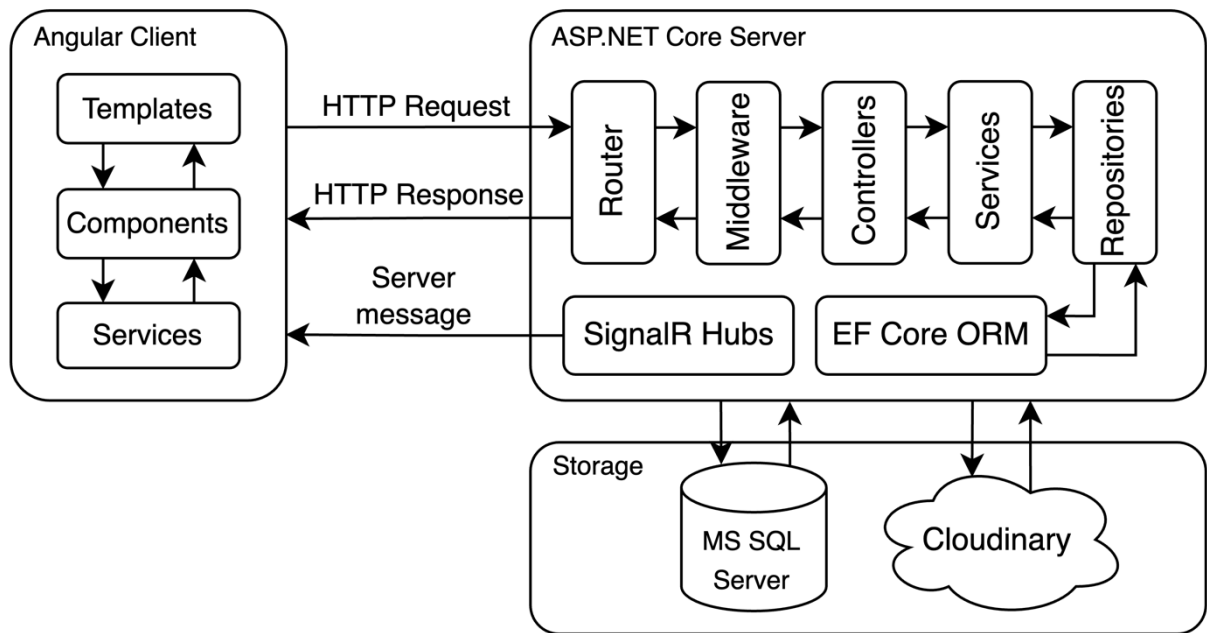
ДОДАТКИ

Додаток 1
Копії графічних матеріалів



ДП.045440-06-99

Вебзастосунок для підтримки спільноти власників домашніх тварин та пошуку і публікації оголошень. Структура бази даних. ER-діаграма

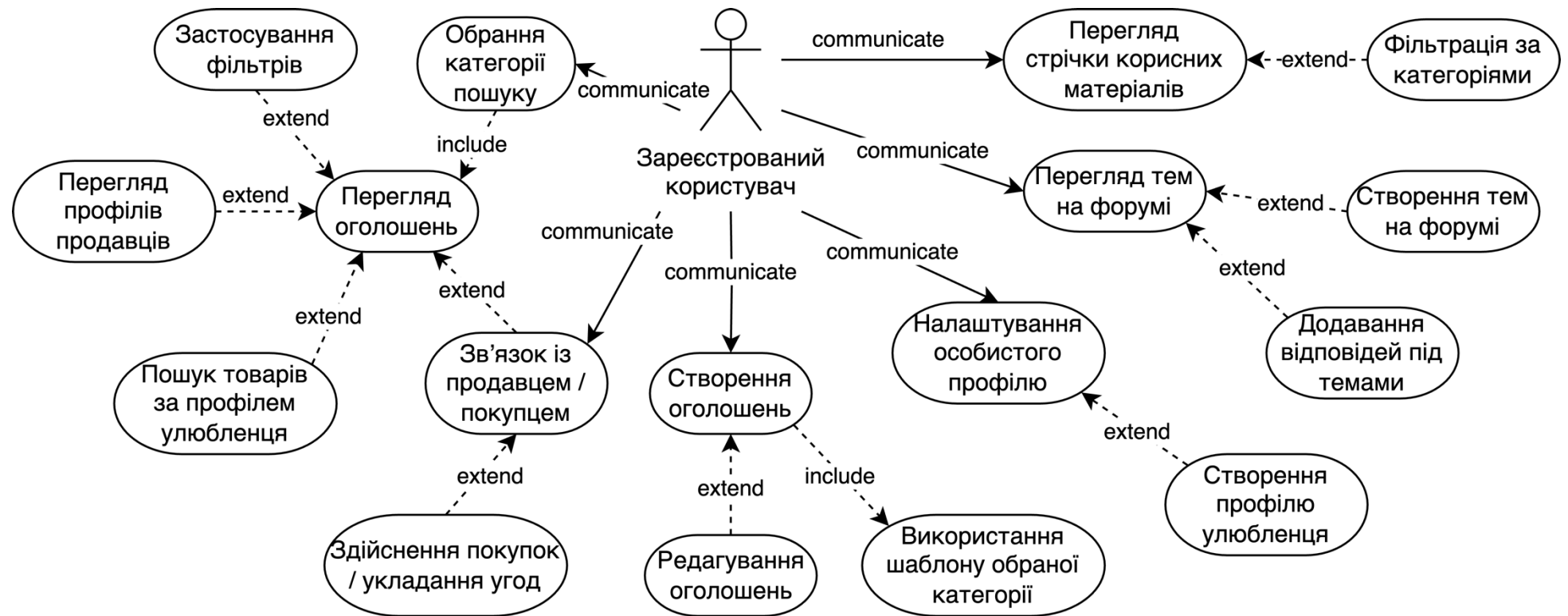


ДП.045440-07-99

Вебзастосунок для підтримки спільноти власників домашніх тварин та пошуку і публікації оголошень. Архітектура застосунку. Структурна схема



Дерево проблем вебзастосунку
Небесна А.А., група КП-11



Варіанти використання. UML-діаграма
Небесна А.А., група КП-11

Додаток 2
Копія презентації

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

**ВЕБЗАСТОСУНОК ДЛЯ ПІДТРИМКИ СПІЛЬНОТИ
ВЛАСНИКІВ ДОМАШНІХ ТВАРИН ТА ПОШУКУ І
ПУБЛІКАЦІЇ ОГОЛОШЕНЬ**

Виконала: Небесна А.А.

Керівник: доцент кафедри ПЗКС, к.т.н., доцент Саяпіна І.О.

Київ – 2025

1/18



АКТУАЛЬНІСТЬ

- Зростання кількості власників домашніх тварин і попиту на товари, послуги та інформацію про догляд.
- Відсутність єдиного спеціалізованого сервісу та можливості взаємодії спільноти власників домашніх улюбленців.
- Переповненість притулків і потреба в ефективному інструменті для прилаштування покинутих тварин.
- Незручність пошуку тварин, товарів і послуг через відсутність гнучких фільтрів у наявних платформах.



ПОСТАНОВКА ЗАДАЧІ

Мета проєкту: створити єдиний онлайн-простір для любителів тварин, що поєднує дошку оголошень, базу знань та спільноту, спрощуючи доступ до інформації, товарів та послуг.

Завдання:

1. Проаналізувати аналогічні програмні рішення для власників домашніх тварин, що є на ринку.
2. Визначити вимоги до вебзастосунку.
3. Обрати та обґрунтувати засоби реалізації.
4. Розробити вебзастосунок для підтримки спільноти власників домашніх тварин та пошуку і публікації оголошень, що враховує недоліки наявних рішень.
5. Провести тестування розробленого програмного забезпечення.
6. Визначити напрямки подальшого розвитку проєкту.

НАЯВНІ АНАЛОГИ



е-Пет



Lapa.shop



OLX

ФУНКЦІОНАЛЬНІ ВИМОГИ ДО РОЗРОБЛЮВАНОВОГО ПЗ



Вебзастосунок повинен забезпечувати такі основні вимоги:

- Наявність широкої категоризації за типами тварин (собаки, коти, птахи, риби, рептилії тощо).
- Можливість розміщувати оголошення з динамічними характеристиками для кожної категорії.
- Можливість знайти супутні товари та послуги.
- Наявність розширених та унікальних фільтрів окремо для кожної категорії оголошень.
- Цифровий профіль тварини в особистому кабінеті користувача.
- Наявність стрічки корисних матеріалів.
- Наявність форуму.
- Можливість обміну повідомленнями.

ОБРАНІ ЗАСОБИ РЕАЛІЗАЦІЇ



C#



ASP.NET Core



SignalR



TypeScript



Entity Framework
Core



Microsoft SQL
Server



Bootstrap



Angular

АРХІТЕКТУРА ВЕБЗАСТОСУНКУ

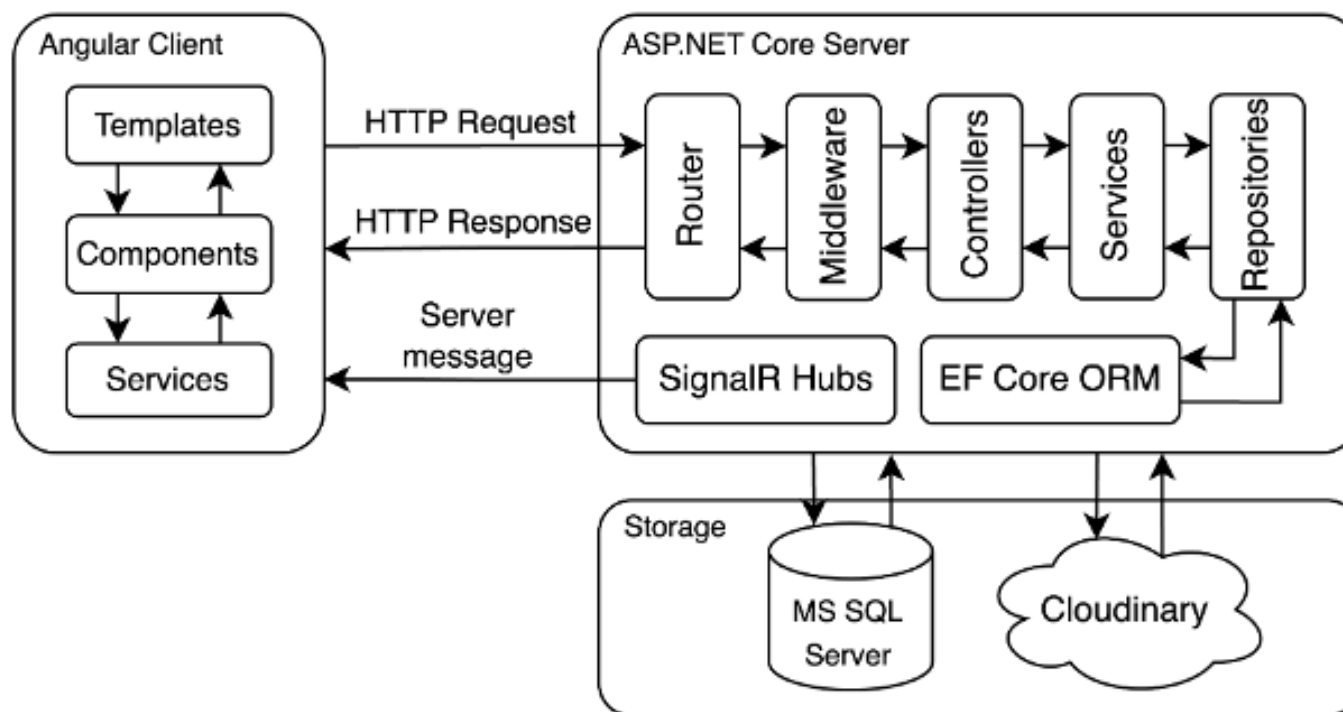
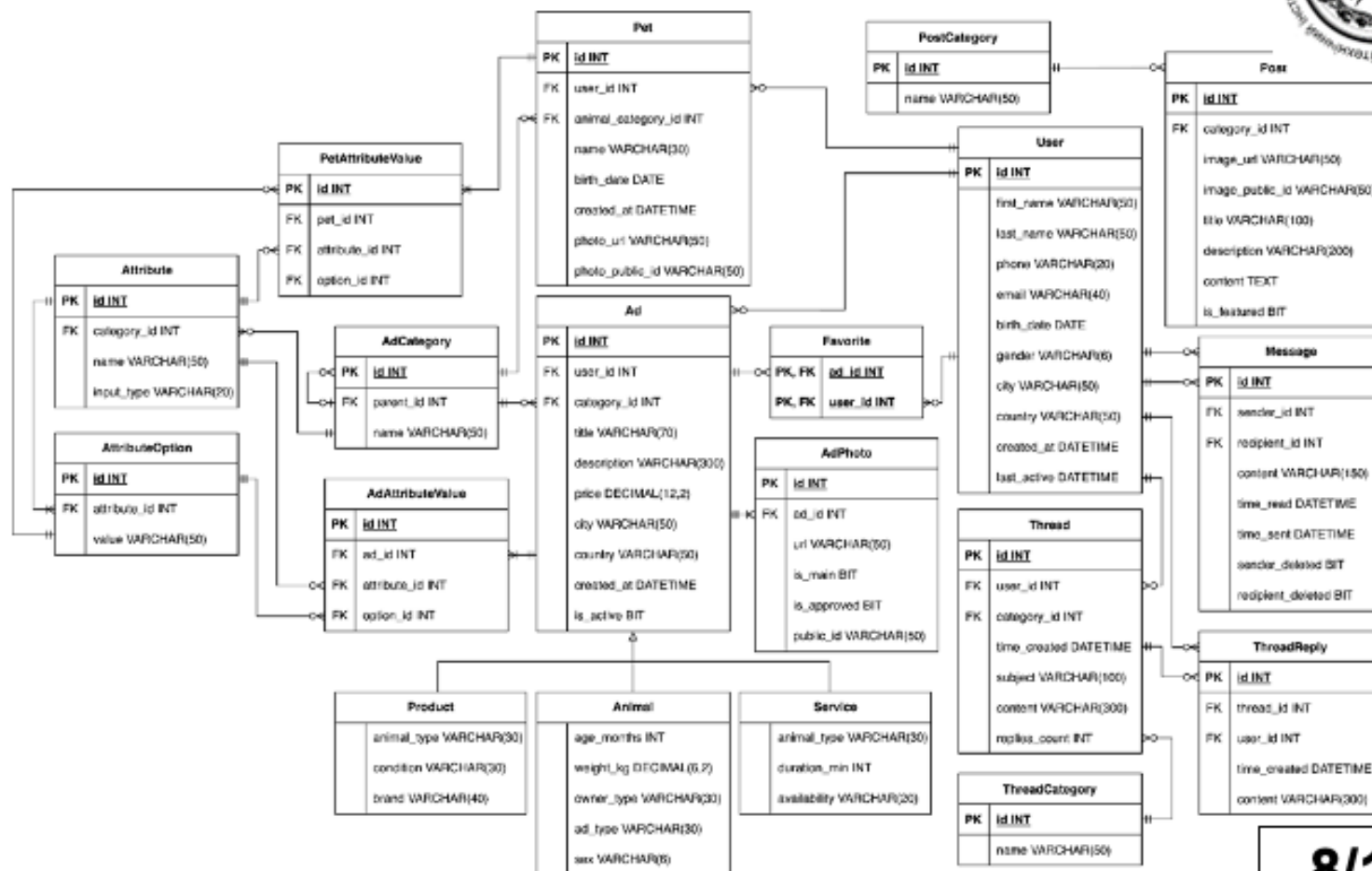


СХЕМА БАЗИ ДАНИХ ЗАСТОСУНКУ



ДІАГРАМА ПРЕЦЕДЕНТІВ





СТОРІНКА ЛИСТУВАННЯ



Animals Products Services Forum Blog

Post Ads

Welcome Ann ▾

Cute Maltipoo looking for family

Location: Kyiv, Ukraine

Age: 3 years

Price: 25 900 UAH

Sex: Female

Weight: 7 kg

Owner: Private

Ad type: For sale

Andrew
2 hours ago

Message

Ad details Photos Messages

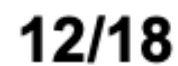
3 minutes ago (unread)

Hello! I was very happy to see your ad. This Maltese is simply adorable! Could you tell me more about him?

Send a private message

Send

11/18



СТОРІНКА РЕЄСТРАЦІЇ



★PetPlace Animals Products Services Forum Blog

E-mail Password Login Register

Sign up

I am a: ☒ Male ☐ Female

First Name

Last Name

Email

Phone

Date of birth

City

Country

Register Cancel

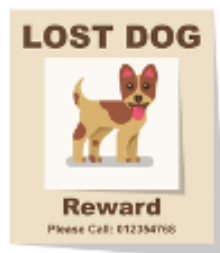
13/18

ПОРІВНЯННЯ З АНАЛОГАМИ



Критерій	OLX	є-Пет	Lapa.shop	PetPlace
Широка категоризація за типами тварин	+	+	-	+
Продаж супутніх товарів та послуг	+	-	+/-	+
Розширені фільтри пошуку для кожної категорії	-	+	-	+
Цифровий профіль тварини	-	-	-	+
Стрічка корисних матеріалів	-	+	+	+
Форум	-	-	-	+

НАПРЯМКИ ПОДАЛЬШОГО РОЗВИТКУ ПРОДУКТУ



Дошка зниклих і
знайдених тварин



Система онлайн-бронювання
послуг через платформу



Інтеграція доставки та
безпечних платежів



Рекомендації на основі
машинного навчання



ВИСНОВКИ

- Проаналізовано наявні рішення на ринку.
- Визначено вимоги до розроблюваного програмного забезпечення.
- Обрано програмні засоби для розробки і обгрунтовано їх вибір.
- Розроблено архітектуру вебзастосунку.
- Реалізовано поставлені до ПЗ вимоги.
- Протестовано розроблене ПЗ, виправлено виявлені помилки.
- Запропоновано напрямки подальшого розвитку продукту.

ПЕРЕВІРКА НА ПЛАГІАТ



Звіт подібності

метадані

Назва організації
National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute

Заголовок
КП-11-Небесна

Автор
Науковий керівник / Експерт
Небесна Анна Андріївна Саяпіна Інна Олександрівна

підрозділ
ФПМ, К-ра програмного забезпечення комп'ютерних систем

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



10

Довжина фрази для коефіцієнта подібності 2

10287

Кількість слів

82879

Кількість символів

17/18



ДЯКУЮ ЗА УВАГУ

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Євгенія СУЛЕМА

“ ____ ” _____ 2024 р.

**ВЕБЗАСТОСУНОК ДЛЯ ПІДТРИМКИ СПІЛЬНОТИ ВЛАСНИКІВ
ДОМАШНІХ ТВАРИН ТА ПОШУКУ І ПУБЛІКАЦІЇ ОГОЛОШЕНЬ**

Програма та методика тестування

ДП.045440-04-51

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Інна САЯПІНА

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Анна НЕБЕСНА

ЗМІСТ

1. ОБ'ЄКТ ВИПРОБУВАНЬ.....	3
2. МЕТА ТЕСТУВАННЯ	3
3. МЕТОДИ ТЕСТУВАННЯ.....	3
4. ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ.....	4

1. ОБ'ЄКТ ВИПРОБУВАНЬ

Вебзастосунок для підтримки спільноти власників домашніх тварин та пошуку і публікації оголошень, забезпечує пошук і розміщення оголошень про тварин, доступ до товарів і послуг, базу знань і спільноту. Розроблений з використанням Angular для клієнтської частини, ASP.NET Core та REST API зі SignalR для серверної частини, а також Entity Framework Core та Microsoft SQL Server для роботи з базою даних.

2. МЕТА ТЕСТУВАННЯ

У процесі тестування повинна бути перевірена коректна робота вебінтерфейсу та взаємодія його елементів у межах усіх основних функцій системи, зокрема динамічного формування оголошень відповідно до обраної категорії тварин, застосування унікальних фільтрів для кожного типу оголошення, пошуку супутніх товарів і послуг, а також функціонування цифрового профілю тварини в особистому кабінеті користувача. Тестування також охоплює перевірку стабільності роботи стрічки корисних матеріалів, функціональності форуму та системи обміну повідомленнями, правильності обробки запитів користувачів з різними правами доступу, а також безперебійної взаємодії клієнтської частини з сервером і базою даних. Особливу увагу приділено відповідності реалізованої логіки вимогам, визначеним у технічному завданні, та зручності користування для різних типів користувачів.

3. МЕТОДИ ТЕСТУВАННЯ

Упродовж усього циклу розробки застосунку проводилось тестування нових функціональних можливостей із використанням методу димного тестування та підходу Grey Box, що дозволяло на ранніх етапах виявляти та

усувати помилки, а також забезпечити стабільність роботи системи при подальших змінах. Для перевірки реалізованих сценаріїв імітувалася робота кінцевого користувача, формувався перелік функціональних можливостей і відповідних тестових випадків, які вручну відтворювались, після чого результати порівнювались з очікуваними. У разі виявлення відхилень проводилось виправлення коду з подальшим повторним тестуванням, що дозволяло ефективно контролювати якість продукту на всіх етапах розроблення.

4. ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Працездатність вебзастосунку перевірялась за допомогою ручного динамічного тестування. Воно включало введення граничних та некоректних значень у вхідні поля, тестування на відповідність функціональним вимогам технічного завдання, а також перевірку зручності навігації та доступності функцій для кожної ролі. Додатково проводився статичний аналіз коду серверної частини. Окремо тестувалась коректність відображення інтерфейсу у різних браузерах (Chrome, Firefox та Edge). Було виконано тестування стійкості системи до помилкових дій з боку користувачів і стабільності роботи при завантаженні великої кількості даних. Для відлагодження та тестування функціональності використовувалися Angular DevTools і Postman.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Євгенія СУЛЕМА

“___” _____ 2025 р.

ВЕБЗАСТОСУНОК ДЛЯ ПІДТРИМКИ СПІЛЬНОТИ ВЛАСНИКІВ
ДОМАШНІХ ТВАРИН ТА ПОШУКУ І ПУБЛІКАЦІЇ ОГОЛОШЕНЬ

Керівництво користувача

ДП.045440-05-34

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Інна САЯПІНА

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Анна НЕБЕСНА

ЗМІСТ

1. ОПИС СТРУКТУРИ ВЕБЗАСТОСУНКУ	3
2. ПРОЦЕДУРА РЕЄСТРАЦІЇ ТА АВТОРИЗАЦІЇ	4
3. СТОРІНКА ОГОЛОШЕНЬ ПРО ТВАРИН	6
4. МОДУЛЬ ДОДАВАННЯ ОГОЛОШЕННЯ.....	8
4.1. Заповнення базових даних	8
4.2. Заповнення загальних даних про тварин.....	8
4.3. Заповнення детальних характеристик	9
4.4. Заповнення контактних даних	10
5. СТОРІНКА ЛИСТУВАННЯ ТА КАРТКИ ОГОЛОШЕННЯ	11
6. СТОРІНКА БЛОГУ	13
7. СТОРІНКА ФОРУМУ	14
8. СТОРІНКА ПРОФІЛЮ КОРИСТУВАЧА	15
9. ПРОЦЕДУРА ОБРОБЛЕННЯ ПОМИЛОК.....	16

1. ОПИС СТРУКТУРИ ВЕБЗАСТОСУНКУ

Вебзастосунок «PetPlace» призначений для розміщення, пошуку та перегляду оголошень, що стосуються домашніх тварин, а також для підтримки спільноти користувачів за допомогою форуму та блогу. Платформа орієнтована на широку аудиторію – як звичайних користувачів, що шукають улюбленця або товари для нього, так і продавців, спеціалістів чи власників бізнесу в цій сфері.

Інтерфейс вебзастосунку динамічно змінюється залежно від ролі та авторизованості користувача. Більшість сторінок (оголошення всіх категорій, форум, блог, перегляд картки оголошення) доступна для перегляду навіть без входу до системи, проте виконання будь-яких інших дій вимагатиме авторизації.

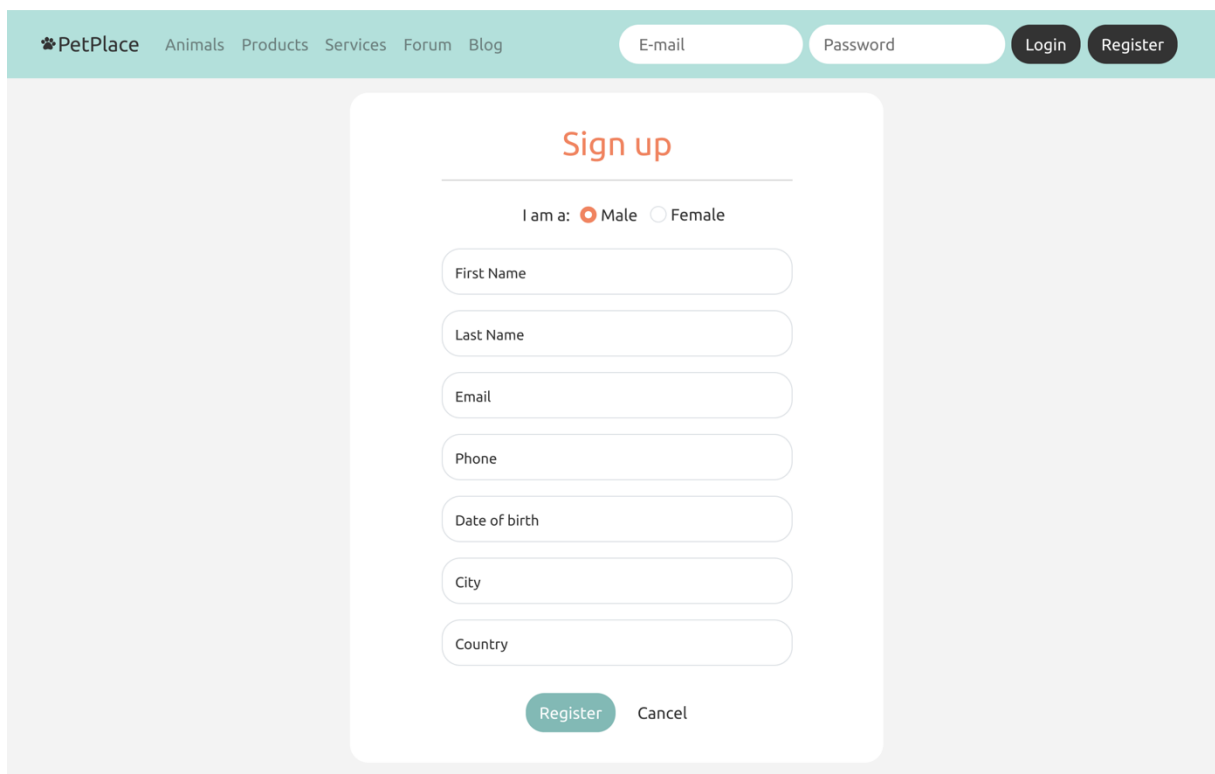
Вебзастосунок включає такі основні сторінки:

- 1) сторінка реєстрації та авторизації;
- 2) сторінка оголошень про тварин;
- 3) сторінка додавання оголошення;
- 4) сторінка картки оголошення та листування з продавцем;
- 5) сторінка блогу;
- 6) сторінка форуму;
- 7) сторінка помилки.

Після входу в систему користувач отримує доступ до додаткових можливостей: створення оголошень, персонального профілю, листування, створення тем на форумі, залишення відповідей тощо. Таким чином, структура вебзастосунку відповідає потребам як звичайних користувачів, так і продавців, пропонуючи інтуїтивно зрозумілий інтерфейс для взаємодії в межах єдиної зооспільноти.

2. ПРОЦЕДУРА РЕЄСТРАЦІЇ ТА АВТОРИЗАЦІЇ

Користувач, який потрапляє на сторінку «Sign up», бачить форму реєстрації, що розташована по центру екрана (рис. 1). У ній новий користувач може ввести особисту інформацію: ім'я, прізвище, електронну пошту, номер телефону, дату народження, місто та країну проживання. Після заповнення всіх обов'язкових полів користувач натискає кнопку «Register». Це створює новий обліковий запис, надає доступ до повної функціональності платформи, і користувач автоматично переходить на головну сторінку, вже будучи авторизованим. Користувач також має можливість натиснути кнопку «Cancel», щоб скасувати процес реєстрації та повернутися на попередню сторінку або до головного інтерфейсу застосунку.



The screenshot shows the 'Sign up' form on the PetPlace website. The header includes the PetPlace logo and navigation links: Animals, Products, Services, Forum, and Blog. On the right, there are input fields for 'E-mail' and 'Password', and buttons for 'Login' and 'Register'. The main form is titled 'Sign up' and includes a gender selection option ('I am a: ☒ Male ☐ Female'). Below this are input fields for 'First Name', 'Last Name', 'Email', 'Phone', 'Date of birth', 'City', and 'Country'. At the bottom of the form are buttons for 'Register' and 'Cancel'.

Рис. 1. Сторінка реєстрації та авторизації

У навігаційній частині сторінки розміщено поля «E-mail» та «Password», а також кнопки «Login» і «Register». Зареєстровані користувачі

можуть одразу авторизуватися, ввівши свої дані та натиснувши кнопку «Login», після чого вони будуть перенаправлені на головну сторінку вебсайту.

3. СТОРІНКА ОГОЛОШЕНЬ ПРО ТВАРИН

Після авторизації користувач може перейти до сторінки оголошень про тварин (рис. 2) за допомогою навігаційної панелі. Дана сторінка є одним із ключових елементів застосунку та забезпечує зручний пошук тварин за численними критеріями.

У верхній частині сторінки розміщено навігаційне меню, яке дозволяє швидко перемикатися між основними розділами: «Оголошення», «Форум», «Блог» тощо. Праворуч від навігаційних посилань відображається іконка «вибране» для доступу до списку збережених оголошень, кнопка «Опублікувати оголошення» та іконка профілю користувача з привітанням.

Основна структура сторінки поділена на такі функціональні блоки:

1. Горизонтальна панель основних фільтрів. Безпосередньо під навігацією розміщена панель швидкого фільтрування, де користувач може задати діапазон цін, стать, вік, вагу тварини або тип продавця. Тут же доступні кнопки для застосування фільтрів, скидання вибраних параметрів та сортування результатів.
2. Бокова (розширена) панель фільтрів. Ліворуч від списку оголошень розташовані іконки категорій тварин (собаки, коти, птахи, риби, рептилії), що дозволяють користувачу швидко перейти до фільтрів, специфічних для обраного типу тварини. Наприклад, для собак можуть бути доступні параметри розміру, типу шерсті, темпераменту тощо. Ця панель адаптивна й автоматично змінює набір доступних фільтрів відповідно до вибраної категорії.
3. Область результатів пошуку. Праву частину сторінки займають картки оголошень, які відображаються у вигляді сітки. Кожна картка містить фотографію тварини, заголовок, ціну та місцезнаходження. При наведенні на оголошення стають доступними кнопки додавання оголошення до «вибраного», перегляду оголошення та переходу до чату з продавцем.

4. Посторінковий навігатор. Унизу сторінки реалізовано елемент пагінації, який дозволяє зручно перемикатися між сторінками з результатами.

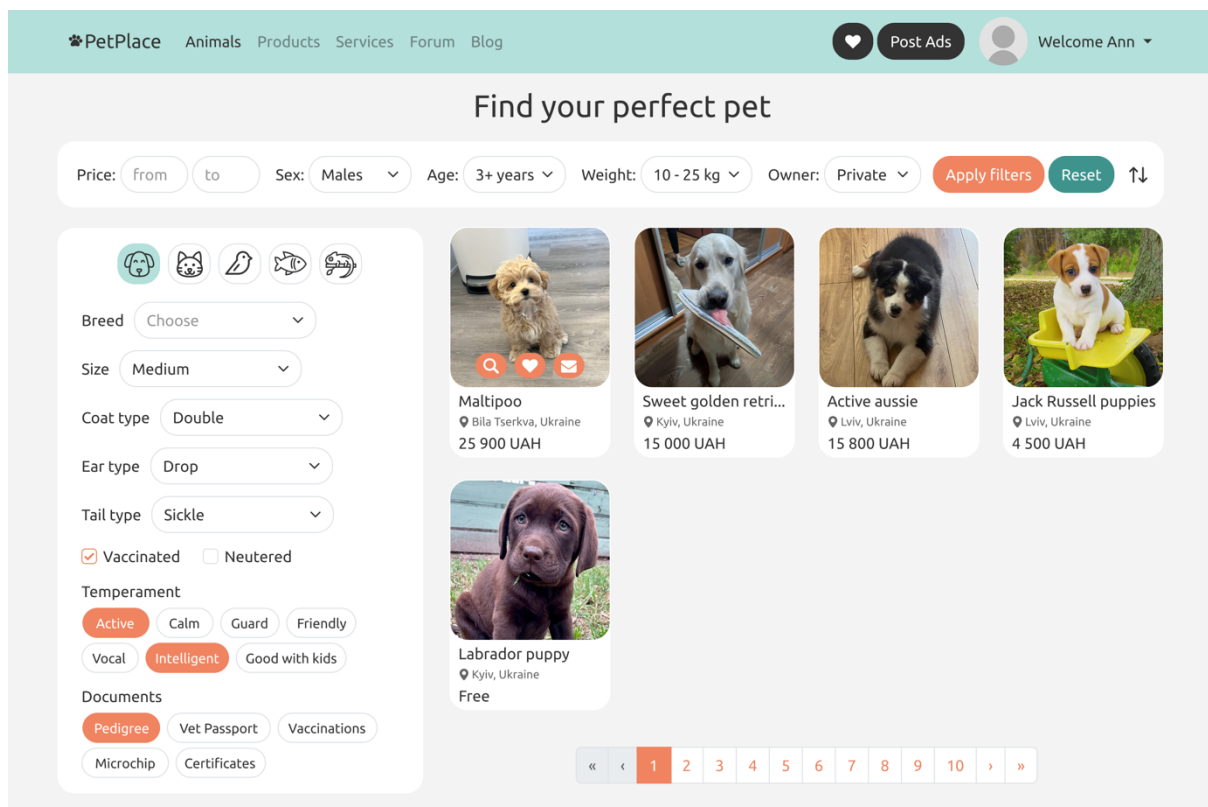


Рис. 2. Сторінка оголошень про тварин

4. МОДУЛЬ ДОДАВАННЯ ОГОЛОШЕННЯ

4.1. Заповнення базових даних

Після переходу на сторінку створення нового оголошення користувачу пропонується обрати тип оголошення: тварина, товар або послуга. Після цього активується форма для заповнення базових даних, які є універсальними для всіх типів оголошень (рис. 3).

Серед обов'язкових полів у цьому розділі можна виділити заголовок, опис, ціну та вибір валюти, а також можливість додати до 7 зображень, які допомагають візуалізувати об'єкт оголошення та зробити його більш привабливим для інших користувачів.

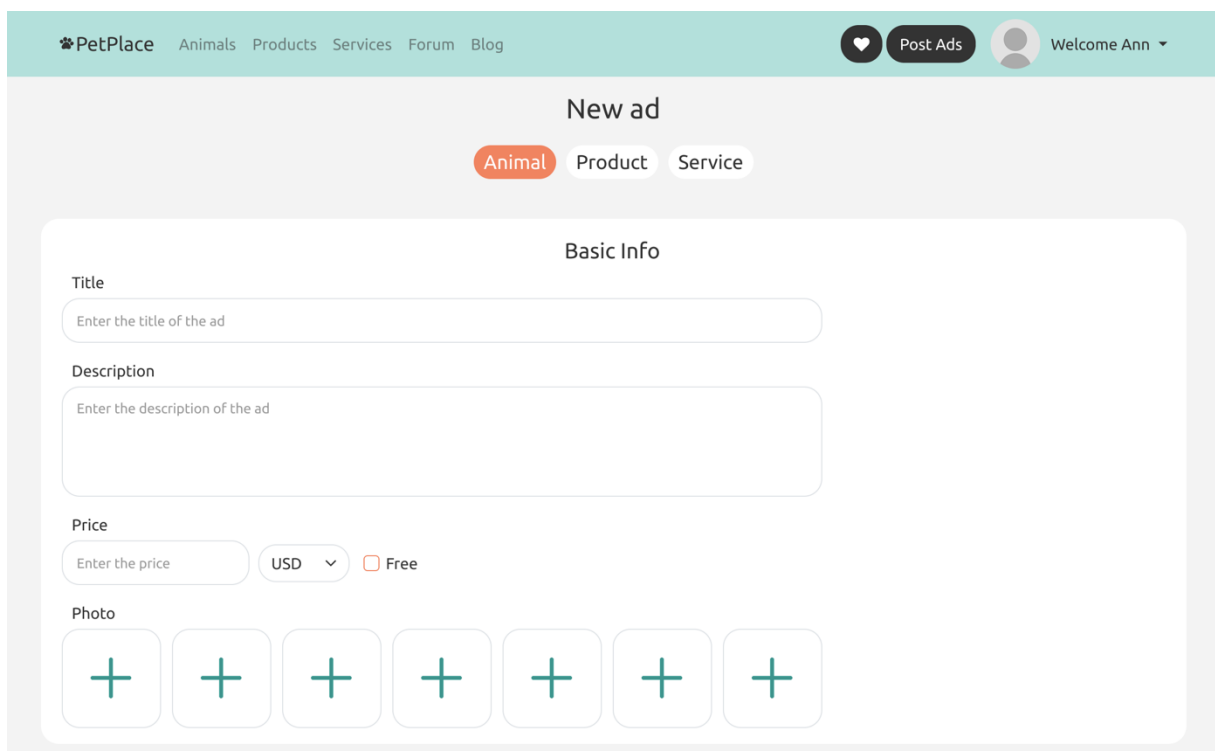


Рис. 3. Розділ базових даних

4.2. Заповнення загальних даних про тварин

У разі вибору типу оголошення «тварина» стає доступним розділ із загальними характеристиками, що застосовуються до будь-якої категорії тварин (рис. 4). На цьому етапі користувач зазначає, до якого типу належить

тварина, наприклад, собака, кіт, птах, риба або рептилія. Далі вказуються стать тварини, її вік (у роках або місяцях), вага (у грамах або кілограмах), тип оголошення (наприклад, продаж, знайдено або в добрі руки) та тип власника (приватна особа, притулок, розплідник тощо). Ці поля створюють узагальнений профіль тварини незалежно від її категорії та слугують основою для подальшого деталізованого опису.

Animal Info

Sex: ☒ Male ☐ Female

Age: months ▼

Weight: kg ▼

Ad type: For sell ▼ Owner: Private ▼

Рис. 4. Розділ загальних даних про тварин

4.3. Заповнення детальних характеристик

Наступний розділ містить деталізовані характеристики, які залежать від обраного типу тварини (рис. 5). У представленому прикладі було вибрано тип «собака». Для цієї категорії користувач може зазначити породу, розмір дорослої особини, тип шерсті, форму вух і хвоста, а також вказати інформацію про стан здоров'я та темперамент. Додатково доступна інформація про те, з ким або з чим тварина ладнає чи, навпаки, не ладнає. У цьому ж розділі є можливість додати дані про наявні документи, наприклад, родовід або ветеринарний паспорт. Такий рівень деталізації допомагає створити повну картину тварини для потенційних покупців або нових власників.

Details

Breed

Size

Coat type

Tail type

Ear type

Health

☒ Vaccinated ☐ Neutered ☐ Dewormed

Temperament

☒ Active ☐ Calm ☐ Guard ☐ Friendly

☐ Vocal ☐ Intelligent ☒ Aggressive

Good with

☒ Children ☐ Other dogs ☐ Farm animals

☐ Cats ☐ Elderly ☐ Car rides ☐ Being alone

☐ Urban lifestyle ☐ Rural lifestyle ☐ Travel

Bad with

☐ Children ☐ Other dogs ☐ Farm animals

☐ Cats ☐ Elderly ☐ Car rides ☒ Being alone

☐ Urban lifestyle ☐ Rural lifestyle ☐ Travel

Documents

☒ Pedigree ☐ Vet Passport ☐ Vaccinations

☐ Microchip ☐ Certificates

Рис. 5. Розділ детальних характеристик відповідного типу тварини

4.4. Заповнення контактних даних

Останнім кроком є заповнення контактної інформації (рис. 6). Користувач вказує електронну адресу, номер телефону, місто та країну проживання. Ці дані дозволяють забезпечити зворотній зв'язок між автором оголошення та зацікавленими особами. Після заповнення всіх необхідних полів користувач натискає кнопку «Опублікувати», після чого оголошення з'являється на платформі та стає доступним для інших користувачів.

Contacts

Email

Phone

Location

Рис. 6. Розділ контактних даних

5. СТОРІНКА ЛИСТУВАННЯ ТА КАРТКИ ОГолоШЕННЯ

Після натискання на кнопку «Повідомлення» у картці оголошення з лістингу оголошень користувач потрапляє на сторінку повного перегляду оголошення, де активується вкладка «Messages» (рис. 7). Дана сторінка об'єднує детальний перегляд інформації про тварину та можливість безпосереднього спілкування з автором оголошення.

Інтерфейс сторінки структурований наступним чином:

1. Блок інформації про оголошення (ліворуч). У верхній частині цього блоку відображається велике зображення тварини, нижче – ключова інформація: назва оголошення, місце розташування, вік тварини, її ціна, стать, вага, тип власника (приватна особа або організація) та тип самого оголошення (продаж, віддам у добрі руки тощо). Під описом тварини розміщена секція з інформацією про автора: його аватар, ім'я або псевдонім, а також дані про останню активність користувача. Поруч знаходяться дві інтерактивні кнопки – іконка «серце» для додавання оголошення до списку обраного (якщо воно ще не додано) та кнопка «Message» (Повідомлення), яка активує вкладку для надсилання повідомлень.
2. Блок із вкладками (праворуч). Основна інформаційна панель праворуч розділена на три вкладки:
 - 1) Ad details (Деталі оголошення) – містить розширену інформацію та характеристики тварини або об'єкта оголошення, зокрема специфічні атрибути, які залежать від категорії (наприклад, для собак – тип шерсті, здоров'я, темперамент тощо);
 - 2) Photos (Фотографії) – демонструє додаткові зображення тварини, що дає змогу краще ознайомитися з нею;
 - 3) Messages (Повідомлення) – відкриває інтерфейс обміну повідомленнями між користувачем та автором оголошення.

3. Діалог листування. У вкладці «Messages» відображається чат, у якому показано історію переписки між поточним користувачем та продавцем. Повідомлення містять текст, позначку часу та статус відправлення. Внизу вкладки розташоване текстове поле для введення нового повідомлення та кнопка «Send» (Надіслати).

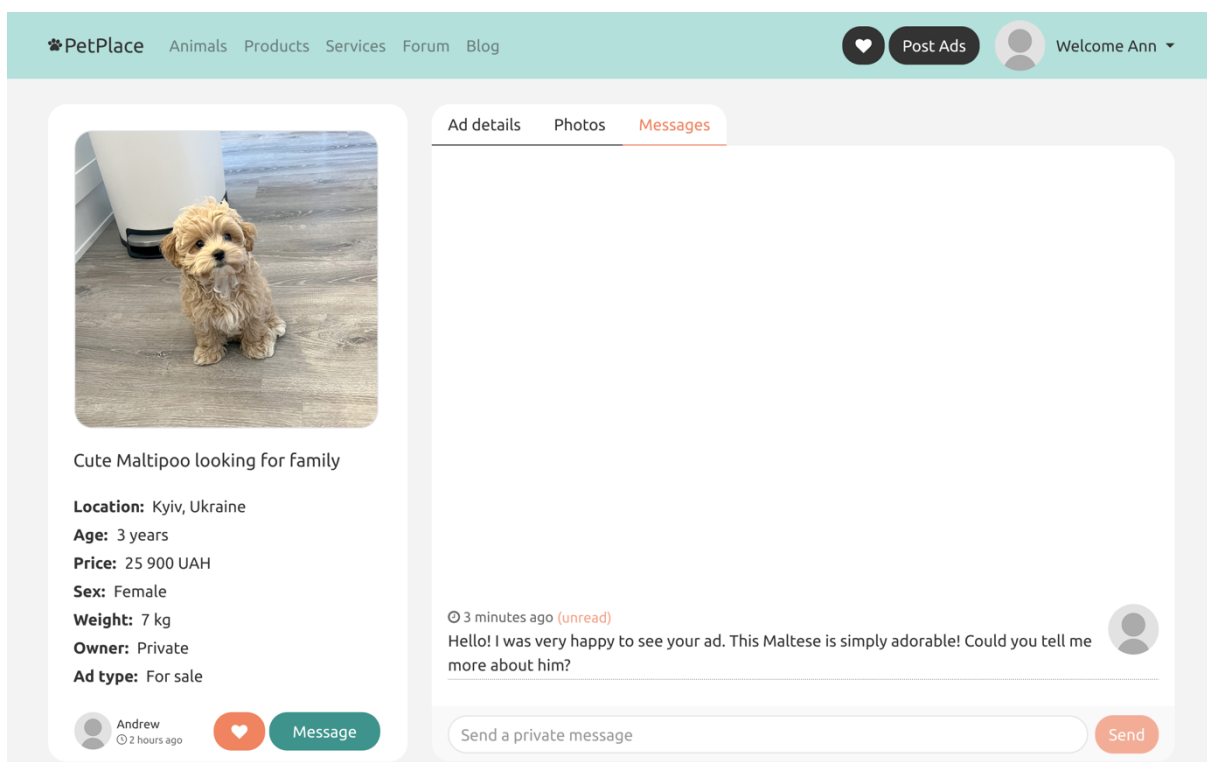


Рис. 7. Сторінка перегляду картки оголошення та листування з продавцем

6. СТОРІНКА БЛОГУ

При переході на сторінку «Blog» користувачу відображаються всі останні пости у вигляді карток, що складаються з заголовка, зображення, частини тексту статті та категорії (рис. 8). Користувач може заповнити поле пошуку та натиснути кнопку «Search», після чого будуть показані всі статті, що відповідають критеріям пошуку. Під полем пошуку також розміщено фільтр категорій, за допомогою якого користувач може обрати теми, що його цікавлять, наприклад: «Training» або «Feeding». Після натискання на категорію відображаються всі пости, що належать до неї. Якщо обрати категорію «All», буде показано всі пости без фільтрації. Натиснувши кнопку «Read more» на обраному пості, користувач переходить на сторінку перегляду статті.

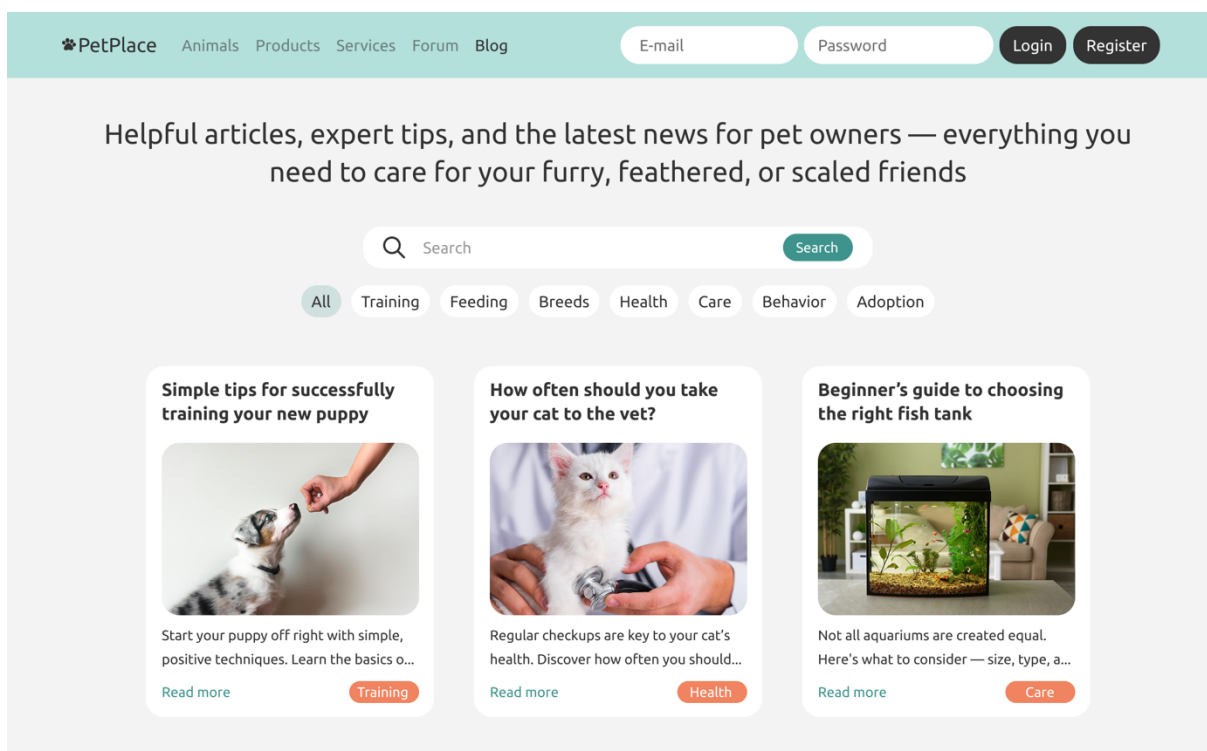


Рис. 8. Вікно перевірки помилок

7. СТОРІНКА ФОРУМУ

Після переходу на сторінку «Forum» користувачу відображаються останні теми обговорень, представлені у вигляді окремих карток (рис. 9). Кожна картка містить аватар користувача, який створив тему, заголовок обговорення, перші рядки повідомлення, кількість відповідей, час створення та категорію.

Користувач має можливість скористатися полем пошуку, розташованим у верхній частині сторінки. Після введення запиту та натискання кнопки «Search» відображаються лише ті обговорення, що відповідають пошуковим критеріям.

Зліва на сторінці розміщена панель «Categories», за допомогою якої користувач може фільтрувати обговорення за тематикою — наприклад, «All discusion» або «Trainning». Після вибору категорії відображаються лише ті теми, що належать до обраного розділу. Користувач має можливість створити нову тему за допомогою кнопки «Ask a Question».

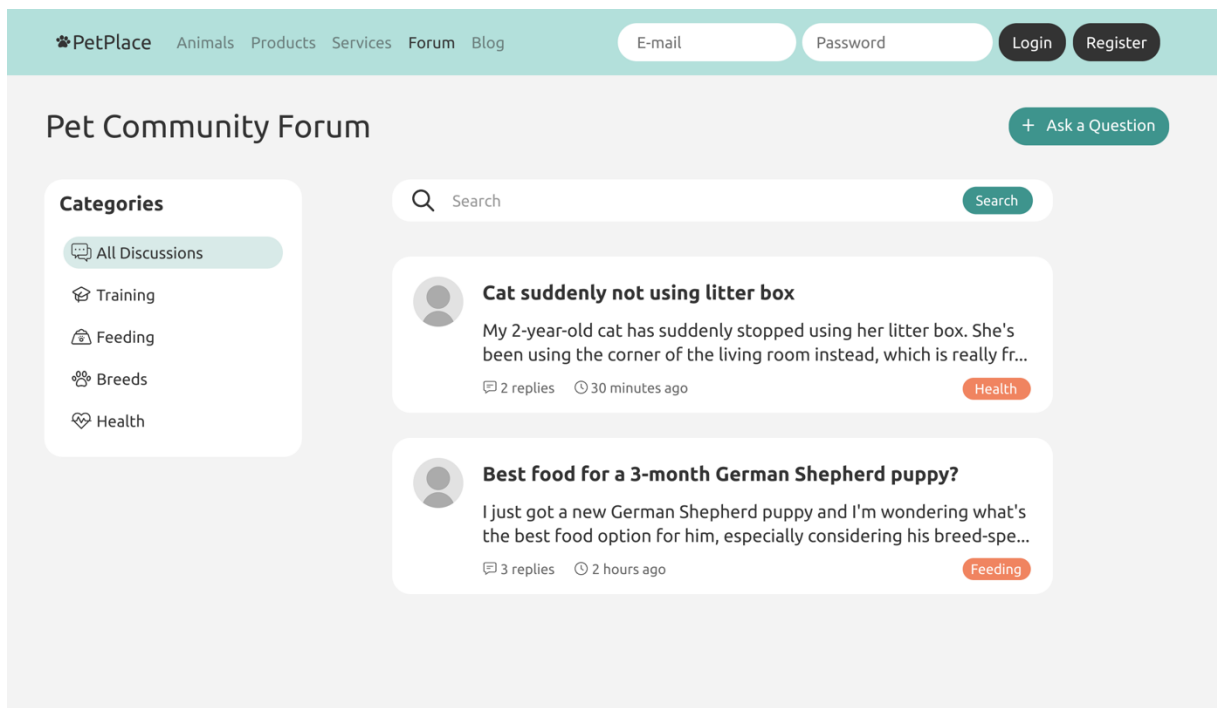


Рис. 9. Сторінка форуму

8. СТОРІНКА ПРОФІЛЮ КОРИСТУВАЧА

Після переходу на сторінку "Profile" користувачу відображається його особиста інформація, пов'язані з ним оголошення, профілі тварин та обрані оголошення (рис. 10).

Зліва на сторінці розташована панель "Personal Info", яка містить основні дані користувача: ім'я, прізвище, електронну пошту, номер телефону, місце проживання та стать. Під цією інформацією розміщена кнопка "Edit", що дозволяє користувачу редагувати свої особисті дані.

Центральну частину екрана займає секція з вкладками: "My ads", "My pets", "Favorites". За замовчуванням активною є вкладка "My ads", де представлені оголошення, створені користувачем. Кожне оголошення відображається у вигляді окремої картки, яка містить фотографію тварини, заголовок, місце розташування та ціну. Праворуч від списку оголошень розміщена картка з іконкою "+", яка дозволяє користувачу створити нове оголошення.

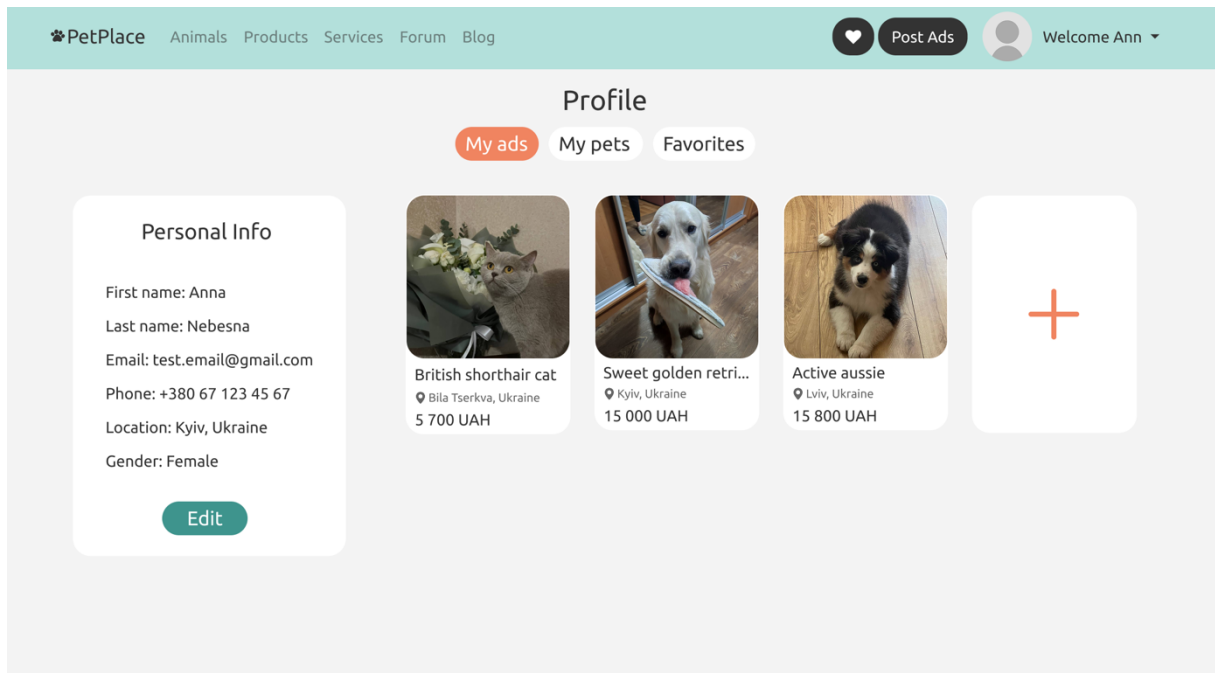


Рис. 10. Сторінка профілю користувача

9. ПРОЦЕДУРА ОБРОБЛЕННЯ ПОМИЛОК

У разі спроби відкрити ресурс, який було видалено або який не існує, користувач потрапляє на сторінку «Page Not Found» («Сторінку не знайдено»). Сторінка містить повідомлення з поясненням ситуації та проханням перевірити правильність введеної адреси або перейти за посиланням на головну сторінку вебзастосунку (рис. 11).

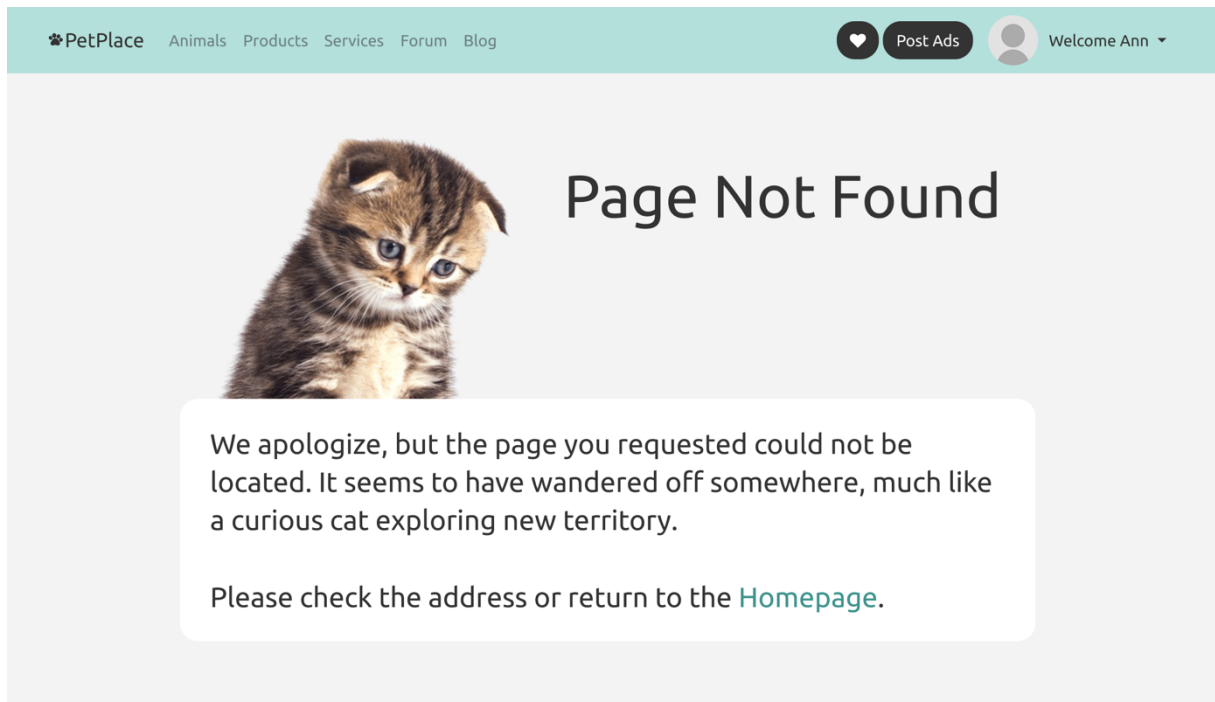


Рис. 11. Сторінка помилки