

Міністерство освіти і науки України
Університет митної справи та фінансів

Факультет інноваційних технологій
Кафедра комп'ютерних наук та інженерії програмного забезпечення

Кваліфікаційна робота бакалавра

на тему: «Розробка та впровадження вебдодатку для розміщення та пошуку оголошень про прилаштування безпритульних тварин»

Виконала: студентка групи K21-1
спеціальність 122 «Комп'ютерні науки»

Лебідь Яна Юріївна
(прізвище та ініціали)

Керівник: к. ф.-м. н., доц. Лебідь О. Ю.
(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент: ДМСУ, Спеціалізоване
управління розробки та супроводження
програмного забезпечення, Департамент з
питань цифрового розвитку, цифрових
трансформацій та цифровізації
(місце роботи)

головний державний інспектор
відділу розробки
програмного забезпечення
(посада)

Бахтін О. В.
(науковий ступінь, вчене звання, прізвище та ініціали)

АНОТАЦІЯ

Лебідь Я. Ю. Розробка та впровадження вебдодатку для розміщення та пошуку оголошень про прилаштування безпритульних тварин.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за спеціальністю 122 «Комп'ютерні науки». – Університет митної справи та фінансів, Дніпро, 2025.

Кваліфікаційна робота присвячена розробці вебдодатку для розміщення та пошуку оголошень про прилаштування безпритульних тварин. Досліджено актуальність проблеми, проведено аналіз предметної області та існуючих рішень, визначено основні функціональні вимоги до вебдодатку. Для організації процесу розробки було проаналізовано різні методи управління проектами. Для роботи обрано підхід Kanban. На етапі проєктування враховано основні принципи зручного та зрозумілого дизайну (UX/UI). Проаналізовано потреби цільової аудиторії, створено структуру та макет інтерфейсу у Figma. Були визначені основні технології для реалізації проєкту. Клієнтська частина реалізована з використанням HTML, CSS та JavaScript, серверна частина – на базі PHP у поєднанні з системою керування базами даних MySQL. Розробка здійснювалася у середовищі VS Code, керування базою даних – за допомогою phpMyAdmin. Результатом роботи є функціональний вебдодаток, що спрощує пошук господарів для безпритульних тварин. Також запропоновано рекомендації щодо його подальшого вдосконалення.

Список публікацій здобувача.

Лебідь О. Ю., Лебідь Я. Ю. Методологія розробки програмного забезпечення Kanban: принципи та переваги. Інноваційні технології, моделі управління кібербезпекою : зб. тез доп. міжнар. наук.-практ. конф., м. Дніпро, 18-20 квіт. 2023 р. Дніпро, 2023. С.25-28.

Ключові слова: вебдодаток, веброзробка, Kanban, публікація оголошень, пошук тварин, дослідження, проєктування, UX/UI дизайн, база даних, HTML, CSS, JavaScript, Figma, VS Code.

ABSTRACT

Lebid Y. Y. Development and implementation of a web application for posting and searching pet adoption ads.

Qualification work for the bachelor's degree in speciality 122 «Computer Science». – University of Customs and Finance, Dnipro, 2025.

The qualification work is devoted to the development of a web application for posting and searching advertisements related to the adoption of stray animals. The relevance of the problem was investigated, the subject area and existing solutions were analyzed, and the main functional requirements for the web application were defined. Various project management methodologies were reviewed and the Kanban approach was selected for the development process. During the design phase, the fundamental principles of user-friendly and intuitive UX/UI design were taken into account. The needs of the target audience were analyzed, and the interface structure and mockup were created in Figma. The key technologies for project implementation were identified. The client side was developed using HTML, CSS, and JavaScript, while the server side was implemented with PHP and the MySQL database management system. The development was carried out in the VS Code environment, and the database was managed using phpMyAdmin. The result of the work is a functional web application that simplifies the process of finding new owners for stray animals. Recommendations for further improvement of the application are also proposed.

List of Applicant's Publications.

Lebid O., Lebid Y. Software Development Methodology Kanban: Principles and Advantages. In: Innovative Technologies, Cybersecurity Management Models: Proceedings of the International Scientific and Practical Conference, Dnipro, April 18–20, 2023. Dnipro, 2023. pp. 25–28.

Keywords: web application, web development, Kanban, ad publishing, animal search, research, projecting, UX/UI design, database, HTML, CSS, JavaScript, Figma, VS Code.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.	
ПОСТАНОВКА ЗАВДАННЯ	8
1.1 Актуальність проблеми прилаштування безпритульних тварин	8
1.2 Аналіз проблеми та предметної області	10
1.3 Аналіз існуючих рішень	12
1.4 Постановка завдання та формулювання вимог	17
1.5 Висновки до першого розділу	18
РОЗДІЛ 2. ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ	20
2.1 Поняття та класифікація вебдодатків. Вибір технологій веброзробки	20
2.2 Вибір серверної частини та бази даних	22
2.3 Застосування принципів UX/UI дизайну у веброзробці	23
2.4 Дослідження методів та засобів вирішення поставлених завдань (Agile, Scrum, Kanban)	26
2.5 Висновки до другого розділу	29
РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ ВЕБДОДАТКУ ДЛЯ РОЗМІЩЕННЯ ТА ПОШУКУ ОГЛОШЕНЬ ПРО ПРИЛАШТУВАННЯ БЕЗПРИТУЛЬНИХ ТВАРИН	30
3.1 Аналіз потреб користувачів і формування функціональних вимог до вебдодатку	30
3.2 Проектування UX/UI дизайну	35
3.3 Реалізація вебдодатку: Frontend і Backend розробка	38
3.4 Результати реалізації вебдодатку та рекомендації щодо подальшого вдосконалення	46
3.5 Висновки до третього розділу	54
ВИСНОВКИ	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	60
ДОДАТКИ	63

ВСТУП

Актуальність дослідження. Проблема безпритульних тварин завжди була досить актуальною. А з початком повномасштабного вторгнення в Україні ця ситуація особливо загострилася. Зросла кількість випадків, коли тварини опиняються на вулиці – покинуті господарями, залишені у зоні активних бойових дій та на тимчасово окупованих територіях. Щоденно волонтери, притулки та небайдужі громадяни шукають способи допомогти тваринам знайти новий дім, проте часто стикаються з труднощами: інформація розкидана по різних джерелах, її складно знайти, відсутні зручні централізовані сервіси, а багато існуючих платформ мають застарілий або заплутаний інтерфейс.

Особливої уваги заслуговує одна з найбільших екологічних катастроф останніх років – підлив окупантами Каховської ГЕС у 2023 році. Цей теракт спричинив масштабне затоплення частини Херсонської області, тоді під водою опинилися будинки, сільськогосподарські угіддя та десятки населених пунктів. Точна кількість загиблих людей, тварин і обсяг завданої шкоди досі невідома, однак, безперечно, йдеться про тисячі постраждалих. Масово гинули найбільш вразливі та беззахисні – домашні та дикі тварини, які не мали можливості врятуватись. Українські волонтери та небайдужі люди рятували тварин у особливих та надскладних умовах, під російськими обстрілами, ризикуючи життям.

Такі події доводять, наскільки важливо мати ефективний інструмент для підтримки діяльності волонтерів, притулків та всіх небайдужих громадян, які хочуть допомагати тваринам. Саме тому було прийняте рішення створити спеціальну онлайн-платформу, яка об'єднає усіх охочих та де можна буде швидко публікувати оголошення про тварин, що загубилися або потребують допомоги. Такий ресурс прискорить процес прилаштування безпритульних тварин, спростить комунікацію, а також дасть можливість ефективно публікувати та шукати оголошення. Тому розробка спеціального вебдодатку – це не лише актуальний, а й соціально значущий проєкт.

Метою кваліфікаційної роботи є створення сучасного вебдодатку, який дозволяє розміщувати та переглядати оголошення про тварин, що потребують нового дому, тимчасової перетримки або загубилися та розшукують своїх власників.

Для досягнення мети визначено такі завдання:

- визначити актуальність проблеми прилаштування тварин;
- проаналізувати існуючі рішення, що допомагають у пошуку домівок для безпритульних тварин;
- сформулювати вимоги до вебзастосунку;
- обрати відповідні технології для розробки клієнтської та серверної частини вебзастосунку, а також базу даних;
- дослідити принципи UX/UI дизайну у веброзробці;
- дослідити підходи до організації розробки (Agile, Scrum, Kanban), обрати найбільш доцільний метод для роботи;
- провести аналіз потреб користувачів і визначити основні вимоги на основі вивчення цільової аудиторії;
- спроектувати архітектуру системи та розробити UX/UI дизайн інтерфейсу вебдодатку;
- реалізувати функціонал клієнтської та серверної частин;
- протестувати роботу системи та визначити перспективи вдосконалення та подальшого розвитку продукту.

Об'єктом дослідження є сучасні технології розробки вебзастосунків, зокрема методи, засоби та інструменти створення вебінтерфейсів.

Предметом дослідження є розробка та впровадження вебдодатку для розміщення та пошуку оголошень про прилаштування безпритульних тварин.

Методи дослідження. В роботі було використано порівняльний аналіз, опитування цільової аудиторії, UX-дослідження, методи проектування програмного забезпечення.

Методи вирішення завдань. Робота над проектом базувалася на каскадному підході до розробки. Для організації процесу використовувався

Kanban. Дизайн інтерфейсу розроблено з урахуванням принципів UX/UI дизайну. Для реалізації фронтенду обрано HTML, CSS та JavaScript, для серверної частини – мову PHP, а для збереження даних – систему керування базами даних MySQL.

Практичне значення роботи. Створення готового рішення, яке можна впровадити у діяльність притулків чи волонтерських організацій, а також яке зможуть використовувати небайдужі громадяни для пошуку або розміщення тварин. Вебдодаток має спростити процес прилаштування тварин, забезпечити комунікацію між волонтерами, притулками та користувачами і сприяти підвищенню обізнаності суспільства щодо проблеми безпритульних тварин. При цьому платформа має відповідати таким критеріям: простий та інтуїтивно зрозумілий інтерфейс, зручність користування, надійність зберігання та обробки даних, відповідність принципам UX/UI дизайну та адаптивність для мобільних пристроїв. Отримані результати мають важливе соціальне значення, тому що допомагають вирішити актуальну проблему та полегшують процес допомоги тваринам, а також підвищують рівень обізнаності людей та відповідального ставлення до тварин, що потребують допомоги.

Інформаційною базою дослідження є законодавчі акти України, книжкові та наукові публікації, електронні ресурси, а також результати власного опитування користувачів.

Економічний ефект від реалізації. Економічна вигода від використання платформи не планується, оскільки вона створена як волонтерський проєкт. Проте в майбутньому можна заснувати благодійний фонд, який допоможе розвивати платформу та збирати кошти на допомогу тваринам.

Структура та обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел, додатків; подана на 59 сторінках; містить 35 рисунків, 4 додатки. Список використаних джерел складає 22 найменування.

РОЗДІЛ 1. АНАЛІЗ ТА ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність проблеми прилаштування безпритульних тварин

Безпритульні тварини – це досить актуальна соціальна проблема не тільки в Україні, а й в усьому світі. В нашій країні питання захисту тварин регулюється Законом України «Про захист тварин від жорстокого поводження» [15]. Він встановлює правила гуманного ставлення до тварин і визначає покарання за жорстоке поводження з ними. Закон пояснює, яких правил повинні дотримуватись власники тварин, регулює діяльність притулків і зоозахисних організацій, а також передбачає покарання за жорстоке поводження з тваринами – адміністративну та кримінальну відповідальність.

На жаль, через безвідповідальне ставлення господарів, відсутність контролю за розмноженням тварин, нестачу притулків та їхні обмежені ресурси кожного року тисячі тварин опиняються на вулиці. Останні кілька років ця проблема набула ще більшого масштабу та загострилася через повномасштабне російське вторгнення. Адже мільйони людей змушені залишати свої домівки і дуже часто вони не мали можливості забрати з собою домашніх улюбленців. Так, у зруйнованих містах і селах залишилися сотні тисяч покинутих тварин, які стали жертвами бойових дій, нестачі їжі та води, а також жорстокого поводження. Волонтери та притулки роблять усе можливе, але через таке стрімке зростання безпритульних тварин їм бракує ресурсів і підтримки. За інформацією Антикризового медіа-центру «Через повномасштабну війну кількість безпритульних тварин в Україні збільшилася в десятки разів. Щодня тисячі українських собак і котів гинуть від голоду, зневоднення й поранень. Абсолютно всі притулки України переповнені – у кожному десь удвічі-втричі більше тварин, ніж було до повномасштабної війни. Проте найгіршою є ситуація в деокупованих та прифронтових регіонах.» [12]. А за інформацією онлайн-ресурсу «УП. Життя» в Україні станом на листопад 2023 року нараховувалося мінімум 140 тисяч безпритульних тварин [20]. Безперечно,

зважаючи на складну ситуацію, що триває й досі, у 2025 році ця цифра, ймовірно, ще вища, що лише підкреслює нагальну потребу у вирішенні цієї проблеми.

У багатьох містах України є досить багато притулків, як великих так і маленьких, проте їхніх ресурсів часто недостатньо для забезпечення належного догляду за всіма тваринами, особливо у маленьких містах та селах. У вирішенні цієї проблеми велику роль відіграють цифрові технології. Так, волонтери та організації часто шукають для тварин нових господарів через соціальні мережі (такі як Facebook чи Instagram), власні сайти або загальні дошки оголошень, як, наприклад, OLX. Проте ці способи не завжди ефективні, бо інформація розпорошена, немає єдиної бази, а взаємодія між волонтерами, притулками та людьми ускладнена. Важливим кроком у вирішенні цієї проблеми може стати створення вебдодатка, який зробить процес прилаштування тварин простішим і зручнішим.

Розробку сервісу публікації оголошень про безпритульних та загублених тварин доцільно реалізувати у вигляді вебдодатку. Для цього будемо використовувати сучасні вебтехнології. Оскільки вони надають зручний доступ до інформації через Інтернет, дають можливість додавати різноманітні функції для взаємодії з користувачами, а також дозволяють створити інтерфейс, який буде зрозумілий та зручний для різних відвідувачів. У роботі [16] розглядається поняття вебтехнології і описано їх розвиток. Спочатку це були просто засоби для створення вебсторінок з текстом, зображеннями, відео та іншими мультимедійними елементами. Проте з часом їх значення стало ширшим і бачимо, що тепер вебтехнології – це всі технічні й програмні засоби, які допомагають людям спілкуватися, працювати разом, знаходити інформацію та користуватися різними онлайн-сервісами через Інтернет.

Тому можемо стверджувати, що вебдодаток зможе об'єднати волонтерів, притулки, зоозахисні організації та всіх, хто хоче прихистити тварину, у єдину екосистему. Це значно спростить процес пошуку та прилаштування, пришвидшить передачу тварин у добрі руки й забезпечить необхідною

інформацією. Крім того, платформа допоможе координувати зусилля волонтерів, вести облік врятованих тварин і сприятиме пошуку як тимчасового, так і постійного прихистку. А також важливою функцією стане можливість для користувачів самостійно публікувати оголошення, які зможе побачити широка аудиторія. Це безперечно збільшить шанси на швидке прилаштування тварин. Крім того, притулки також зможуть отримати більше підтримки від небайдужих громадян.

Тож можна зробити висновок, що, зважаючи на актуальну соціальну проблему, особливо в умовах воєнного стану, розробка подібних рішень буде дуже актуальною. Отже, майбутній вебдодаток може зробити значний внесок у розв'язання проблеми безпритульних тварин в Україні.

1.2 Аналіз проблеми та предметної області

Зараз у світі існує велика кількість типів платформ для розміщення оголошень про прилаштування тварин, які користуються популярністю. Як вже було зазначено раніше, популярними в Україні платформами для публікації таких оголошень є соціальні мережі (Facebook, Instagram, Telegram) та дошки оголошень (OLX). Вони мають досить широку аудиторію та дозволяють швидко поширювати інформацію, проте їхній основний недолік – це те, що через велику кількість загальних оголошень інформація про тварин може загубитися серед іншого контенту. Крім того, там часто бракує спеціалізованих функцій для фільтрації та управління оголошеннями.

Серед українських спеціалізованих платформ можна виділити власні сайти притулків, такі як «Сіріус» [19], «Нарру Рав» [5], «Домівка врятованих тварин», «В добрі руки». Їхня особливість – вони зосереджені на прилаштуванні саме своїх підопічних. Це є їхньою перевагою, оскільки вся інформація перевірена та актуальна, проте їхня база оголошень обмежується лише тваринами, що перебувають у цих організаціях, а розмістити оголошення про інших тварин немає можливості. Є також і деякі спеціалізовані сайти, що

мають більше можливостей, такі як «Look4Paws» [8], «GladPet» [3], «Pethelp», «Adopt.ua» [1] та інші. Вони збирають оголошення з різних джерел і дозволяють зручно шукати тварин за віком, породою, місцезнаходженням та іншими параметрами.

Доцільно також звернути увагу і на зарубіжні платформи, такі як «AdoptAPet» та «Petfinder». Це одні з найпопулярніших платформ у США та інших країнах. Вони мають ефективні алгоритми підбору, можливість створення профілів для користувачів та волонтерських організацій, а також можливість інтеграції з притулками та ветеринарними клініками. Їхньою сильною стороною є масштабність бази даних та автоматизовані рекомендації тварин, проте вони не надто зручні та малодоступні для користувачів з України, оскільки не мають локалізації, адаптації до місцевих умов та не підтримують українську мову.

У таблиці Б.1 (Додаток Б) порівнюємо чотири основні типи платформ: соціальні мережі, дошки оголошень, сайти притулків та спеціалізовані сайти. Аналіз показав, що кожен тип має свої переваги та недоліки. Бачимо, що соціальні мережі забезпечують велике охоплення, але не мають зручних фільтрів та перевірки оголошень. Дошки оголошень дозволяють знайти тварину швидше, але вони не завжди спеціалізовані на темі прилаштування. Тоді як сайти притулків містять лише перевірену інформацію та мають меншу аудиторію. Спеціалізовані платформи є найбільш зручним варіантом для користувачів, оскільки вони поєднують структурований пошук, перевірку даних і підтримку волонтерів.

Порівнюючи всі ці платформи, можна визначити найефективніші підходи до покращення користувацького досвіду, виявити основні проблеми та знайти способи їх вирішення. Все це допоможе зібрати корисні ідеї для розробки та покращення майбутнього вебдодатку.

Загалом проведений аналіз допоміг визначити ключові аспекти, які варто врахувати при розробці структури власного продукту. Так, врахувавши сильні сторони платформ та усунувши їхні недоліки, можемо створити ефективний,

зручний та корисний сервіс для прилаштування тварин, що відповідатиме потребам українських користувачів.

Тепер можемо визначити основні напрямки для розробки власного веб-додатку: можливість розміщення та пошуку оголошень, деталізовані фільтри, забезпечення широкого охоплення, інтуїтивний та зрозумілий дизайн.

1.3 Аналіз існуючих рішень

Перед початком розробки вебдодатку варто проаналізувати вже існуючі рішення у цій сфері. Основна мета – потрібно виявити сильні та слабкі сторони, а також визначити ключові елементи, які будуть враховані при створенні нашого вебдодатку. Дослідження вже існуючих рішень – це один із найважливіших етапів у створенні продукту. Це допомагає врахувати найкращі практики, виявити їхні сильні сторони, а також знайти недоліки та уникнути поширених помилок. При аналізі українських платформ можна зрозуміти, які функції найбільш затребувані користувачами, а також можна визначити їхні проблеми, наприклад, незручний чи застарілий інтерфейс, складність пошуку та фільтрації, відсутність єдиної бази даних тощо.

Варто зазначити, що навіть маловідомі або недоступні для українських користувачів зарубіжні платформи можуть бути корисними. Вони можуть дати ідеї для нових рішень, наприклад, кращого підбору тварин чи зручнішої роботи з притулками. У майбутніх дослідженнях ці ресурси можуть допомогти зрозуміти, як найкраще подавати інформацію про тварин, що шукають дім. Також це допоможе зрозуміти, які способи сортування та фільтрації оголошень є найзручнішими для користувачів, як автоматизувати оновлення даних, щоб вони постійно були актуальними, а також які ще функції можна впровадити, щоб зробити вебдодаток зручним та приємним у використанні. Також слід взяти до уваги та дослідити успішні UX/UI рішення, адаптувати для українського користувача та впровадити їх у свій вебдодаток.

Для оцінки візьмемо такі ключові аспекти, як: функціональність, UX/UI дизайн, переваги та недоліки. Спершу визначимо, які основні та додаткові функції доступні на сайті. Який саме функціонал робить сайт більш зручним та ефективним. Оцінимо також зручність користування, тобто наскільки сайт інтуїтивно зрозумілий для користувачів. Чи легко користувач знаходить необхідну інформацію за допомогою навігації. Подивимося на загальний дизайн та візуальне сприйняття, чи виглядає сайт сучасно, які кольори використовуються, чи є елементи, що відволікають увагу або ускладнюють взаємодію. Також важливо перевірити адаптивність сайту. Чи зручно користуватися ним на мобільних пристроях і чи має він адаптивну версію. Крім того, треба оцінити, що працює добре на кожному сайті, чому користувачі можуть обирати саме цей ресурс для пошуку тварин чи розміщення оголошень. А також проаналізувати слабкі місця, що потребують вдосконалення, і те, що можна було б покращити, щоб уникнути подібних помилок при розробці вебдодатку.

Отже, для дослідження оберемо п'ять популярних сайтів, що надають можливість розміщення оголошень про тварин, які шукають нових власників, зробимо порівняння обраних сайтів:

1) «Сіріус» [19] – один з найбільших притулків для безпритульних тварин в Україні. Користувачі можуть переглядати оголошення про тварин, що шукають дім, дізнаватися про події та новини притулку, а також робити пожертви.

На сайті пропонується 6 варіантів допомоги: фінансова підтримка, усиновлення тварини, волонтерство, допомога речами, опіка тварини та організація благодійних кампаній. Користувачі можуть переглядати профілі тварин з фото, описами та детальною інформацією. Однією з цікавих особливостей є «колесо фортуни», яке випадковим чином пропонує тварину для усиновлення. Є додаткові функції для тих, хто поки не готовий взяти тварину – «Погуляти з твариною», «Взяти під опіку». Також є інформація про події та новини притулку, можливість фінансової підтримки онлайн, контактна

інформація для зв'язку та можливість перегляду звітів про надану допомогу. Користувач може додавати тварин до обраного, а також ознайомитися з правилами усиновлення. Сайт має сучасний, яскравий та привабливий вигляд з приємними кольорами та типографікою. Є велика кількість графічного контенту та інтерактивних елементів, тому сайт привабливий і зручний у користуванні. Навігація інтуїтивно зрозуміла. Завдяки адаптивному дизайну сайт комфортно використовувати на різних пристроях. Є якісні та яскраві фотографії тварин, що додають ще більше привабливості сайту. Серед переваг можна виділити відсутність зайвої інформації, що дозволяє зосередитися лише на важливих даних про тварин, а також добре продумана навігація, яка забезпечує легкий доступ до потрібної інформації. Є велика база оголошень, ефективний пошук і фільтри. Говорячи про недоліки, то на сайті мало інформації про тварин на картках.

2) «GladPet» [3] – онлайн-ресурс для прилаштування та допомоги безхатнім тваринам. Користувачі можуть обирати котів чи собак для адопції, є можливість опублікувати оголошення, стати волонтером або переглянути список притулків.

Цей ресурс має велику базу тварин, що шукають дім, а також розділ з тваринами, які вже знайшли своїх власників. Користувачі можуть переглядати профілі тварин з фото та описами, а також є блог з корисними статтями. Ще можна здійснити фінансову допомогу онлайн. Сайт також надає контактну інформацію для зв'язку, можливість створити акаунт та публікувати тварин для знаходження власників, додавати організації та залишати заявки для волонтерства. Є опції додавати тварин до обраного, переглядати звіти про надану допомогу, а також є можливість заповнити форму та задати питання. Сайт має приємні кольори та типографіку, що робить його зручним для користування. Навігація інтуїтивно зрозуміла, є дуже детальні фільтри для пошуку. Також є адаптивний дизайн, завдяки якому сайт комфортно використовувати на різних пристроях. Говорячи про переваги, то тут добре продумана навігація, велика база оголошень, ефективний пошук. Картки

містять достатньо інформації, тому користувачі одразу отримують основні дані про тварину, не відкриваючи при цьому нову сторінку. Проте варто зазначити, що дизайн сайту виглядає дещо застарілим.

3) «Happy Paw» [5] – благодійний фонд, що займається допомогою безпритульним тваринам, їх прилаштуванням та підтримкою притулків.

На сайті є можливість переглядати профілі тварин із фото та детальним описом. Можна розміщувати оголошення про пошук господарів, а також публікувати інформацію про знайдену або загублену тварину. Також доступна опція взяття тварини під опіку та здійснення фінансової підтримки онлайн. Є розділ з переліком притулків та волонтерів. Є можливість заповнити форму та надіслати повідомлення організації. Загалом сайт зручний у користуванні, проте виглядає дещо застарілим, має недостатньо інформації про тварин у деяких оголошеннях і має ускладнену навігацію, тому важко одразу знайти необхідну інформацію про притулки. Сайт коректно відображається на мобільних пристроях, завдяки адаптивному дизайну, також має ефективний пошук та досить велику базу з оголошеннями.

4) «Adopt.ua» [1] – база безпритульних тварин з усієї України, що пропонує зручний та ефективний сервіс для адопції.

Сайт містить окремий розділ із інформацією про тварин, які постраждали від військових дій і потребують евакуації та тварин, що знаходяться в безпеці. Також є можливість здійснювати фінансову підтримку онлайн, переглядати анкети тварин із детальним описом і фото. Дизайн сайту зручний у користуванні, але виглядає застарілим. Навігація добре продумана, а сайт коректно адаптується під мобільні пристрої. Варто відмітити логічну структуру розділів, велику базу оголошень, ефективний пошук із дуже детальними фільтрами, а також розгорнуту інформацію про кожну тварину. Є і деякі недоліки, наприклад, застарілий візуальний стиль, відсутність головної сторінки для ознайомлення з сервісом (користувач одразу потрапляє на сторінку з оголошеннями), а також обмежений функціонал (можливість лише переглядати та розміщувати оголошення).

5) «Look4Paws» [8] – платформа для пошуку та прилаштування безпритульних тварин, що об'єднує волонтерів, притулки та потенційних власників.

На цій платформі можна переглядати профілі тварин із фото, описами та детальною інформацією, а також здійснювати фінансову підтримку онлайн. Платформа пропонує зручний пошук за різними критеріями, можливість розміщення оголошень навіть без реєстрації, а також тут є блог із корисними статтями. Для зв'язку з організацією передбачена спеціальна форма. Сайт має зручну навігацію та адаптивний дизайн, тому ним зручно користуватися на різних пристроях. Варто відмітити деякі сильні сторони цього сайту. Тут є велика база оголошень, детальна інформація про кожну тварину, зручне та помітне поле пошуку на головному екрані. На сторінці анкети тварини також пропонуються інші тварини з цього самого міста, а в оголошеннях вказана дата публікації – це важлива деталь, оскільки можна одразу оцінити їхню актуальність. Крім того, є і слабкі місця, що можна вдосконалити, наприклад, недостатньо інформації про тварин на картках, а також відсутність контактних даних. Тому для зв'язку щодо адопції необхідно заповнювати спеціальну форму.

Отже, у ході дослідження було проаналізовано п'ять популярних онлайн-платформ, які допомагають безпритульним тваринам знайти нових власників. Було оцінено їхню функціональність, зручність використання та дизайн, а також виявили основні переваги та недоліки.

Загалом всі розглянуті сайти пропонують широкий вибір тварин для адопції, можливість перегляду профілів із фото та детальними описами, а також різні варіанти допомоги – від фінансової підтримки до волонтерства. Деякі платформи мають розширений функціонал, як-от розділи з тваринами, які постраждали від війни, або можливість взяти тварину під опіку, а ще розміщення оголошення з пошуком загубленої чи знайденої тварини.

Говорячи про дизайн, то варто зазначити, що деякі сайти мають сучасний, привабливий вигляд із якісною графікою, інтерактивними елементами та

зручною навігацією, тоді як інші виглядають дещо застарілими та потребують оновлення. Важливим також є наявність ефективного пошуку та детальних фільтрів, що значно спрощує процес підбору тварини.

Серед основних проблем, з якими стикаються користувачі, можна виділити нестачу інформації на картках тварин, складність у доступі до контактних даних та інколи складну і неочевидну навігацію. Також деяким платформам варто оновити дизайн і зробити його зручнішим та привабливішим, щоб відповідати сучасним вимогам.

Загалом дослідження показало, що всі платформи виконують свою головну місію – допомагають безпритульним тваринам знайти дім. Однак для підвищення ефективності варто вдосконалити інтерфейси, зробити оголошення більш інформативними та забезпечити простішу комунікацію між користувачами та волонтерами.

1.4 Постановка завдання та формулювання вимог

Було виявлено, що у сучасному суспільстві та особливо в умовах війни проблема безпритульних тварин залишається актуальною. Багато волонтерів, організацій та звичайних людей займаються їх порятунком та пошуком нових домівок. Проте, провівши дослідження, бачимо, що існуючі платформи для розміщення оголошень мають деякі недоліки та не завжди адаптовані під потреби пошуку родини для тварин. Це створює додаткові труднощі для людей, які прагнуть допомогти. Метою даної роботи є розробка та впровадження вебдодатку, який дозволить користувачам ефективно розміщувати та знаходити оголошення про прилаштування безпритульних тварин. Основна функціональність додатку буде спрямована на спрощення процесу взаємодії між людьми, що бажають взяти тварину, та тими, хто шукає для неї новий дім.

Для забезпечення повноцінного функціонування вебдодатку необхідно реалізувати кілька ключових можливостей. Головна сторінка міститиме короткий опис платформи, її переваги та можливості. Каталог оголошень

відображатиме всі актуальні оголошення (список доступних тварин для адопції) у вигляді карток із фото, описом та основною інформацією. За допомогою фільтрації можна буде обирати тварин за видом, віком, регіоном та іншими критеріями. Сторінка оголошення міститиме детальну інформацію про тварину та фото. Користувачі зможуть зв'язатися з власником оголошення за допомогою контактів. На сторінці додавання оголошення буде розміщена форма для створення нового запису, що міститиме завантаження фото, опис тварини та контактні дані. В особистому кабінеті люди зможуть переглядати та редагувати власні оголошення. Для взаємодії з платформою користувачі проходилимуть реєстрацію через Email.

Майбутній вебдодаток має також відповідати і нефункціональним вимогам. Зокрема, він має бути зручним у використанні, з простим інтерфейсом, який буде адаптований для різних пристроїв, зрозумілою навігацією та швидким доступом до основних функцій. Він має працювати швидко – сторінки повинні завантажуватися без затримок, а пошук і фільтрація оголошень виконуватися миттєво. Редагувати або видаляти оголошення зможуть лише їхні власники. Важливо, щоб додаток стабільно працював та був сумісним із сучасними браузерами, такими як Google Chrome, Mozilla Firefox, Safari та Edge.

Розроблений вебдодаток допоможе людям, які хочуть знайти дім для безпритульних тварин або шукають улюбленця. Він спростить розміщення та пошук оголошень, забезпечить швидкий зв'язок між користувачами. У результаті цей проєкт сприятиме вирішенню актуальної проблеми – допоможе швидко прилаштовувати тварин і полегшить цей процес для всіх учасників.

1.5 Висновки до першого розділу

У цьому розділі було розглянуто проблему безпритульних тварин в Україні, зокрема, як війна погіршила ситуацію. Було доведено, наскільки

важливо мати ефективний інструмент для підтримки діяльності волонтерів, притулків та всіх небайдужих громадян, які хочуть допомагати тваринам.

Прийнято рішення створити спеціальну онлайн-платформу, яка об'єднає усіх охочих та де можна буде швидко публікувати оголошення про тварин, що загубилися або потребують допомоги. Такий ресурс прискорить процес прилаштування безпритульних тварин, спростить комунікацію, а також дасть можливість ефективно публікувати та шукати оголошення.

Було проведено порівняльний аналіз соціальних мереж, дошок оголошень, сайтів притулків і спеціалізованих сервісів, виявлено їхні плюси й мінуси. Також проаналізовано, як працюють притулки, волонтери та різні платформи для пошуку нових домівок для тварин.

З'ясувалося, що багато платформ мають недоліки та деякі проблеми, з якими можуть стикатися користувачі, такі як, наприклад, складний пошук, відсутність перевірки інформації, обмежений вибір тварин, застарілий дизайн тощо.

Було сформульовано основні вимоги до вебдодатку, серед яких: можливість швидкого розміщення та пошуку оголошень за фільтрами; перегляд оголошень без обов'язкової реєстрації; особистий кабінет для зареєстрованих користувачів з можливістю керування оголошеннями; зручний та адаптивний інтерфейс; швидке завантаження даних та сумісність із сучасними браузерами.

Отже, отримані результати аналізу, проведеного у цьому розділі, стануть основою для подальшої розробки вебдодатку, який сприятиме швидшому та ефективнішому пошуку домівок для безпритульних тварин.

РОЗДІЛ 2. ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ

2.1 Поняття та класифікація вебдодатків. Вибір технологій веброзробки

Вебтехнології вже стали невід'ємною частиною нашого життя, вони впливають на різні його сфери і допомагають нам у роботі, спілкуванні, розвагах та пошуку інформації. Завдяки їм можна навчатися онлайн, робити покупки, працювати з дому, обмінюватися даними та користуватися безліччю онлайн-сервісів тощо.

Одним із ключових елементів вебтехнологій є вебдодатки – це програмне забезпечення, яке працює у веб-браузері та дає можливість користувачу взаємодіяти з сервером через мережу Інтернет, не потребуючи встановлення на пристрій. Вебдодатки широко використовуються у різноманітних сферах нашого життя – від бізнесу й фінансів до освіти, медицини, розваг і державних послуг. Вони полегшують життя та спрощують повсякденні завдання, так, можна швидко отримати потрібну інформацію, навчатися чи працювати онлайн, спілкуватися з людьми з усього світу, здійснювати покупки та багато іншого. При цьому достатньо для цього мати будь-який пристрій з доступом до Інтернету.

Вебзастосунок – це розподілений застосунок, який працює через Інтернет у браузері. Він взаємодіє із сервером для обробки даних та збереження даних. Від звичайних сайтів вебзастосунки відрізняються тим, що мають більше можливостей, наприклад, додавання оголошень, пошук інформації та комунікація між користувачами. Крім того, вони забезпечують швидкий доступ до бази даних у реальному часі, саме тому вони досить зручні та ефективні для користувача.

За даними джерела [22] вебдодатки можна класифікувати залежно від того, як користувач взаємодіє з ними:

- статичні вебдодатки відображають статичну інформацію – незмінний текст, зображення, відео. Вони не змінюються в залежності від дій користувача;

– динамічні вебдодатки реагують на запити та змінюються в залежності від дій користувача. При цьому вони містять динамічну інформацію, наприклад, результати пошуку, новини або прогноз погоди;

– інтерактивні вебдодатки дозволяють користувачам взаємодіяти з інтерфейсом, наприклад, відповідати на запитання, виконувати завдання, обмінюватися повідомленнями тощо. Такі застосунки використовуються у різних сферах, зокрема для навчання, розваг або роботи. Як приклад таких застосунків – освітні платформи, месенджери, ігри, різні інструменти для роботи та сервіси.

Для створення вебдодатку, де можна розміщувати та шукати оголошення про прилаштування безпритульних тварин, спочатку реалізуємо фронтенд-частину. Використаємо HTML і CSS для створення структури сторінок та оформлення зовнішнього вигляду.

HTML (англ. HyperText Markup Language – мова розмітки гіпертексту) – це стандартизована мова, яка використовується для створення структури вебсторінок, що відображаються у браузері. Браузер отримує HTML документ від сервера через протоколи HTTP або HTTPS або завантажує його з локального диска, після чого інтерпретує код в інтерфейс, який відображатиметься на екрані. CSS – каскадні таблиці стилів (англ. Cascading Style Sheets) – спеціальна мова, що використовується для запису оформлення сторінок, написаних мовами розмітки даних [14].

Інтерактивність платформи забезпечить JavaScript, з його допомогою сайт буде більш живим, а користувач зможе взаємодіяти із сайтом. JavaScript – це багатопарадигмальна, динамічна мова зі своїми типами та операторами, стандартними вбудованими об'єктами та методами. Її синтаксис базується на мовах Java та C – багато структур з цих мов застосовуються і в JavaScript. Дана мова підтримує об'єктно-орієнтоване програмування з прототипами об'єктів і класами [6]. Її використання дозволить реалізувати динамічні елементи, такі як інтерактивні кнопки, фільтрацію та валідацію форм.

Бекенд-частина буде реалізована за допомогою мови програмування PHP, що буде обробляти запити від користувачів та керуватиме усією логікою застосунку, а також буде взаємодіяти з базою даних. Для зберігання та оновлення оголошень використаємо реляційну базу даних MySQL. У ній буде зберігатися уся необхідна інформація.

Обравши ці технології, зможемо забезпечити швидку та продуктивну роботу додатку, зручність для користувачів, а в майбутньому розширювати функціонал та покращувати користувацький досвід.

2.2 Вибір серверної частини та бази даних

Для того аби застосунок повноцінно працював, обробляв запити користувачів, взаємодіяв з базою даних та мав певну логіку маємо реалізувати не тільки клієнтську частину, а і серверну. Як вже було зазначено раніше, у межах проєкту реалізація бекенд-частини буде за допомогою мови програмування PHP, а для зберігання інформації використовуватиметься система управління базами даних MySQL.

PHP – це популярна мова програмування для розробки вебсайтів. Вона нескладна у використанні, має простий синтаксис та багато готових рішень, підтримується більшістю хостингів, а також добре працює разом з HTML та JavaScript. Мова PHP використовується на етапі оброблення запиту сервером.

Суть принципу клієнт-серверної моделі полягає у тому, що перегляд вебсайту здійснюється за участю двох комп'ютерів: клієнта і сервера. Клієнт – це комп'ютер користувача, на якому відкривається вебсторінка, а сервер – це комп'ютер, де ця сторінка зберігається. Вони з'єднані через глобальну мережу та таким чином обмінюються даними. Після запуску браузера і введення адреси сайту користувачем, браузер надсилає запит на сервер. Після чого цей запит обробляється та надсилається відповідь – код сторінки, який браузер відображає на екрані. Цей документ представляється у форматі HTML, а також може містити каскадні таблиці стилів (CSS) та сценарії, написані на мові

JavaScript. Так, сервер надсилає набір інструкцій, а браузер їх виконує. Оскільки PHP працює на сервері, інколи частина обробки може виконатися до того, як відповідь буде надіслана клієнту. Так, сервер, отримавши запит, може виконати PHP-сценарій, згенерувати необхідний HTML-код і лише після цього надіслати браузеру. Таким чином, PHP працює під час обробки запиту сервером, а результати цієї обробки вже потрапляють до клієнта [13]. За допомогою цієї технології буде реалізовано такі функції, як реєстрація та авторизація користувачів, додавання та редагування оголошень, відображення каталогу, а також зв'язок сайту з базою даних.

Для надійної роботи з даними обрано систему MySQL. Вона дозволить зручно зберігати інформацію у вигляді таблиць, легко шукати потрібні дані та підтримувати одночасно багато користувачів. У базі даних будуть зберігатися облікові записи користувачів, оголошення та параметри тварин, які потрібні для фільтрації в каталозі.

Отже, завдяки використанню цих технологій, серверна частина сайту зможе ефективно обробляти запити, працювати з базою даних і забезпечуватиме правильну роботу основних функцій сайту.

2.3 Застосування принципів UX/UI дизайну у веброзробці

Ключову роль у створенні ефективних, зручних та водночас естетично привабливих цифрових продуктів відіграє UX/UI дизайн. UX (User Experience) визначає логіку взаємодії користувачів із додатком, тоді як UI (User Interface) – його візуальне оформлення.

UX відповідає за створення максимально комфортної взаємодії, відповідає за зручність користування продуктом. UX-проекування – це процес визначення та формування досвіду, який користувач отримує, взаємодіючи з продуктом. Кінцевою метою проектування UX є не тільки задоволення запитів користувача. Процес проектування спрямований на те, щоб допомогти

продукту досягти поставлених бізнес-цілей і зробити так, щоб ці цілі збіглися з цілями користувача [21].

UI-дизайн включає розробку інтуїтивно зрозумілих, привабливих та зручних інтерфейсів. Тобто, його завдання – забезпечити візуальну частину сайту чи додатку, яка буде спрощувати взаємодію користувача із системою, покращуватиме досвід та загалом буде приємною для юзера.

Як зазначається у статті NNGroup [10], створення ефективного UI передбачає дотримання основних принципів візуального дизайну, серед яких:

- масштаб і візуальна ієрархія – розташування елементів відповідно до їхньої важливості;
- баланс і контраст – гармонійне поєднання кольорів, форм та текстур;
- принципи гештальту – впорядкування елементів для зручного сприйняття.

Крім того, важливо забезпечувати просту навігацію, консистентність і єдиний стиль – узгодженість шрифтів, кольорової палітри та елементів інтерфейсу. А також адаптивність – коректне відображення інтерфейсу на різних пристроях.

Також якісний UX/UI дизайн передбачає глибоке розуміння потреб користувачів, тому на етапі розробки інтерфейсу важливо проводити UX-дослідження (вони допомагають аналізувати та виявляти очікування користувачів) і тестування прототипів, що дозволяє виявляти та виправляти потенційні проблеми ще на етапі проєктування.

З появою адаптивного дизайну сайти почали автоматично підлаштовуватися під різні екрани телефонів та комп'ютерів, тому на етапі проєктування важливо наперед продумувати, як розміщувати важливі елементи інтерфейсу і перевіряти, як вони виглядатимуть на різних пристроях. Також дуже важливо враховувати особливості найпопулярніших мобільних операційних систем – iOS та Android, кожна з яких має свої правила оформлення інтерфейсу. Сучасні дизайн-системи допомагають створювати інтуїтивні та візуально привабливі інтерфейси. Наприклад, Google розробила

власний стиль дизайну для Android – Material UI [4]. Apple створила Human Interface Guidelines [2], там детально описано, як має виглядати інтерфейс на iPhone чи iPad.

Так, говорячи про Material UI, то він заснований на принципах реального світу, імітує світло, тіні, глибину, що створює відчуття реалістичності та глибини інтерфейсу. Ця дизайн-система використовується в багатьох Android-додатках і вебпроектах. Натомість, Human Interface Guidelines (HIG) – це набір правил дизайну Apple, що визначає стиль iOS-додатків. Вони акцентують увагу на простоті, мінімалізмі, інтуїтивності та плавних анімаціях. Який стиль та підхід обрати, залежить від платформи, типу продукту та потреб користувачів [2, 4, 18].

Важливим аспектом сучасного UX/UI дизайну є забезпечення доступності (accessibility) – створення інтерфейсів, зручних для людей з різними фізичними або когнітивними особливостями. Для того, аби інтерфейс був доступним потрібно, наприклад, робити хороший контраст між текстом і фоном, дозволяти збільшувати шрифт, додавати підписи до зображень (alt-тексти), забезпечити можливість користуватися сайтом з клавіатури. Також варто враховувати потреби людей із вадами зору, слуху або з порушеннями руху. Такий підхід не лише зробить продукт зручнішим для всіх, а й відповідатиме міжнародним вимогам, як-от стандарт WCAG.

Зазвичай для розробки дизайну макетів і прототипів використовують такі популярні інструменти, як Figma та Adobe XD. Також для спрощення роботи на ранніх стадіях проєктування використовується інструмент FigJam. Вони дають можливість швидко робити створювати, редагувати та тестувати інтерфейси.

Отже, UX/UI дизайн відіграє дуже важливу роль у створенні сучасних цифрових продуктів. Саме від нього залежить, наскільки легко користувачі розумітимуть, як користуватися сайтом або додатком, чи буде зручно та приємно це робити. Саме поєднання продуманої логіки (UX) та привабливого вигляду (UI) допомагає зробити зручний та корисний продукт. Крім того, варто пам'ятати про адаптивність для різних пристроїв, особливості мобільних

платформ, використання дизайн-систем та принципи доступності для всіх людей. Важливо добре вивчати потреби користувачів, проводити попередні дослідження цільової аудиторії аби майбутні додатки чи сайти відповідали сучасним вимогам та очікуванням людей.

2.4 Дослідження методів та засобів вирішення поставлених завдань (Agile, Scrum, Kanban)

Розробка вебдодатку складатиметься з кількох пов'язаних етапів – проектування, дизайн та програмування. Щоб краще організувати роботу над усім процесом, варто використовувати сучасні методи розробки програмного забезпечення, зокрема гнучкі методології (Agile) та такі популярні фреймворки, як Scrum і Kanban. Розглянемо кожну з них детальніше.

Agile – це підхід до розробки програмного забезпечення, коли робота відбувається поступово, з постійною перевіркою та вдосконаленням. Команда постійно спілкується з користувачами, отримує зворотний зв'язок і може швидко вносити зміни до завершення проєкту. Основна ідея Agile полягає у тому, що не треба чекати кінця роботи, а перевіряти та вдосконалювати продукт на кожному етапі. Ключову роль тут відіграє зворотний зв'язок – як від учасників команди, так і від потенційних користувачів. Методологія Agile офіційно з'явилася у 2001 році після публікації Agile Manifesto. З того часу з'явилося багато гнучких фреймворків, таких як Scrum, Kanban, Lean і Extreme Programming (XP) [11].

Наступний фреймворк – Scrum. Це командний підхід, який дозволяє ефективно застосовувати принципи Agile завдяки чіткій організації процесу. Уся робота поділяється на короткі цикли – спринти, тривалість кожного зазвичай 1-4 тижні. Команда обирає задачі, працює над їх реалізацією впродовж спринту, далі демонструє проміжний результат та обговорює подальші кроки. Scrum допомагає зробити роботу прозорою, регулярно оцінювати, як усе

просувається, і швидко змінювати план, наприклад, додавати нові функції або враховувати побажання майбутніх користувачів.

Kanban – ще один популярний метод організації гнучкої розробки, що допомагає команді бачити весь процес виконання завдання. На відміну від Scrum, тут немає чітко визначених спринтів, але є можливість контролювати навантаження та порядок виконання завдань. Усі задачі розміщуються на спеціальній дошці, де учасники команди можуть легко їх відслідковувати у будь-який час.

Така канбан-дошка допомагає візуально подати завдання, визначити обсяг незавершеної роботи, підвищити її ефективність та швидкість. Цей метод базується на використанні карток, які позначають початок і завершення певних етапів. Така система дозволяє легко розподіляти завдання, керувати проєктами будь-якої складності, швидко виявляти проблемні місця. Також дошка не тільки допомагає виявити недоліки, така візуалізація процесу показує внесок та цінність кожного члена команди, що мотивує і підсилює командну роботу. Почати побудову системи Канбан можна з простої дошки з трьома основними колонками: «Зробити» (To do), «У роботі» (Doing), «Готово» (Done) [17] (Додаток А). Такий підхід зручний для невеликих команд або особистої роботи, коли важливо чітко бачити, на якому етапі знаходиться кожне завдання. Kanban допомагає тримати рівномірний ритм роботи та уникати перевантаження.

Kanban і Scrum мають деякі спільні ідеї, але підходять до роботи по-різному. Їх не слід плутати між собою. Деякі команди поєднують принципи Kanban і Scrum у підхід під назвою «scrumban». Вони беруть із Scrum фіксовану тривалість спринтів і ролі, а з Kanban – обмеження на кількість завдань у роботі та контроль часу виконання [7].

Під час реалізації вебдодатку для розміщення та пошуку оголошень про прилаштування безпритульних тварин важливо приділити увагу вибору відповідного підходу до організації процесу розробки, щоденного планування та відстеження прогресу. Попередньо було проаналізовано основні гнучкі методології розробки програмного забезпечення: Agile, Scrum та Kanban.

Найзручнішим методом для роботи над даним проєктом є Kanban. Завдяки йому можна планувати й організовувати завдання без чітких часових обмежень та легко змінювати процес за потреби у будь-який момент.

У рамках Kanban створимо візуальну дошку у Figma, яка буде зручним інструментом для керування завданнями (рис. 2.1). Поділимо дошку на кілька колонок:

- To Do (До виконання) – задачі, які ще потрібно виконати;
- In Progress (У процесі) – завдання, над якими зараз працюють;
- Done (Виконано) – завершені задачі.

Кожне завдання оформлено у вигляді картки з коротким описом, що допоможе легко бачити, на якому етапі воно знаходиться, визначати пріоритети та за потреби змінювати план дій.

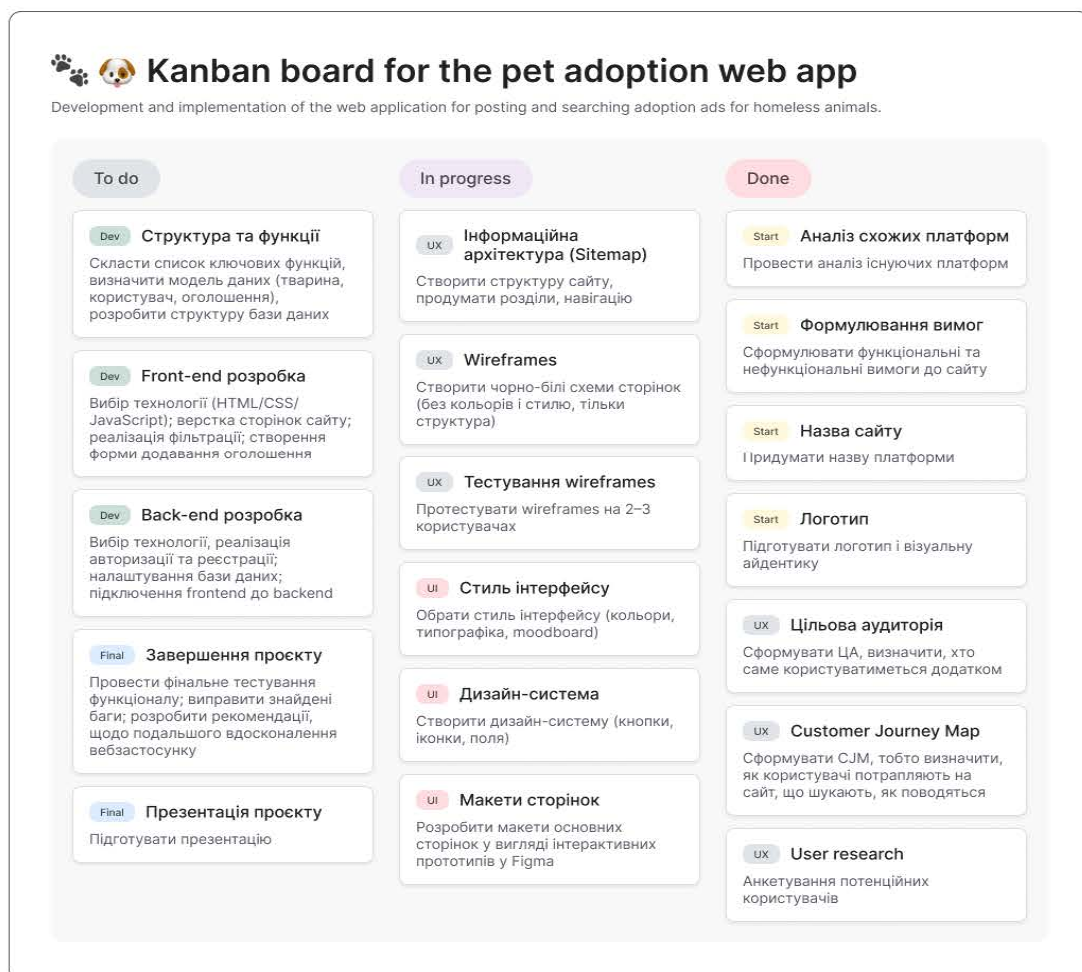


Рисунок 2.1 – Kanban дошка для візуалізації та керування процесами під час роботи над проєктом

2.5 Висновки до другого розділу

У даному розділі було проведено детальний огляд і аналіз основних технологій, методів та підходів, що використовуються у сучасній веброзробці. Зокрема, було розглянуто поняття вебдодатків, їх класифікацію та ключові технології для створення клієнтської та серверної частини вебзастосунку.

Для фронтенду обрано базові технології – HTML, CSS та JavaScript, які забезпечують структуру, стильове оформлення та інтерактивність, що дозволить створити зрозумілий, привабливий і зручний вебдодаток. Серверна частина буде реалізована на PHP, а для зберігання даних буде використовуватися база даних MySQL. Це дозволить ефективно обробляти запити користувачів та зберігати усі необхідні дані.

Також було приділено окрему увагу важливості UX/UI дизайну. Розглянуто основні принципи, що допомагають зробити інтерфейс зручним, зрозумілим та адаптивним для різних пристроїв. Було визначено ключові принципи проектування інтерфейсів, а також особливості адаптивного дизайну й актуальні дизайн-системи, такі як Material UI та Human Interface Guidelines. Крім того, було відзначено важливість доступності для всіх користувачів.

Окремо було розглянуто сучасні підходи до розробки – Agile, Scrum та Kanban. Найкращим варіантом для цього проєкту виявився метод Kanban, який дозволяє зручно планувати і контролювати виконання завдань.

Отже, у результаті проведеного аналізу, було обрано набір технологій та інструментів, які дозволять розробити повноцінний та функціональний вебдодаток.

РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ ВЕБДОДАТКУ ДЛЯ РОЗМІЩЕННЯ ТА ПОШУКУ ОГЛОШЕНЬ ПРО ПРИЛАШТУВАННЯ БЕЗПРИТУЛЬНИХ ТВАРИН

3.1 Аналіз потреб користувачів і формування функціональних вимог

На початковому етапі розробки було проведено дослідження цільової аудиторії для того аби визначити основні потреби та очікування потенційних користувачів, а також сформувані чіткі функціональні вимоги для майбутнього вебдодатку. Цільову аудиторію умовно поділено на три основні категорії:

- люди, які хочуть знайти для себе домашню тварину;
- люди, які шукають дім для безпритульної тварини;
- люди, які знайшли загублену тварину і шукають її власника.

Це можуть бути волонтери, працівники притулків, власники тварин або просто небайдужі користувачі. Для кожної з цих груп було визначено їх ключові потреби, цілі та проблеми. Структуру цільової аудиторії представлено графічно у Figma (рис. 3.1).

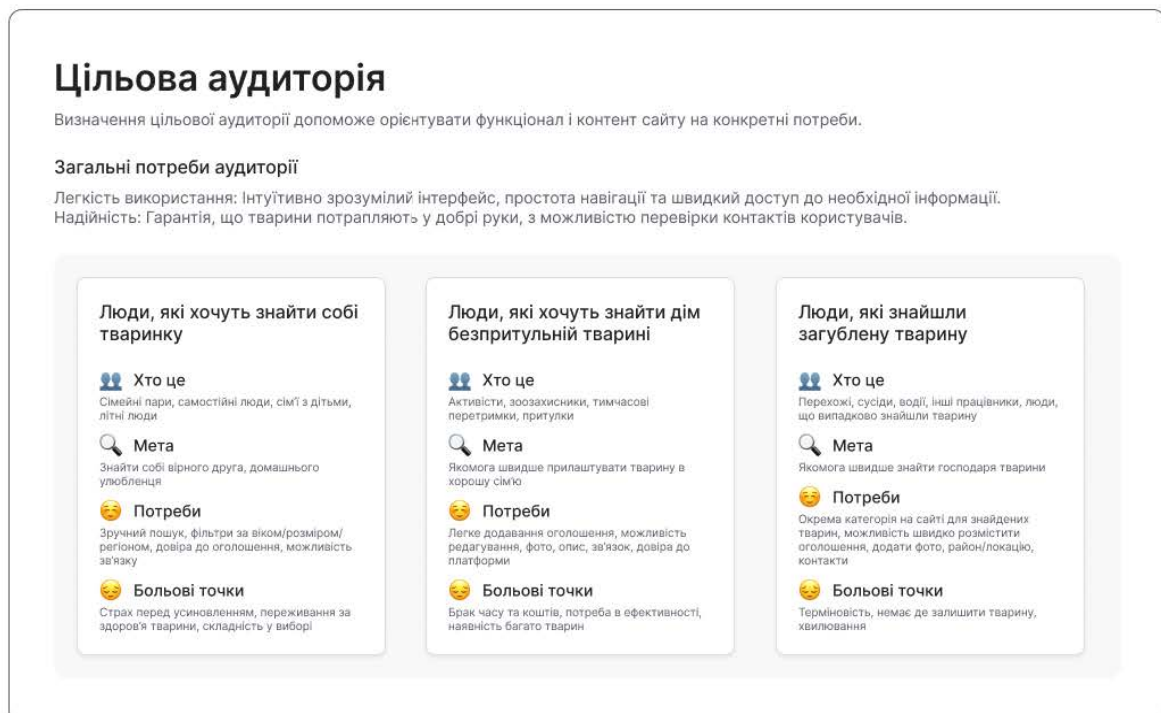


Рисунок 3.1 – Візуалізація результатів дослідження цільової аудиторії

Для збору якісної інформації було створено анонімне опитування у Google Forms (рис. 3.2), у якому взяли участь 20 респондентів (рис. 3.3). Отримавши статистику відповідей на питання, можна краще зрозуміти цільову аудиторію та мати уявлення про портрет користувача, можна зрозуміти, у яких ситуаціях виникає потреба у використанні платформи та обрати той функціонал для сайту, який точно буде затребуваним.

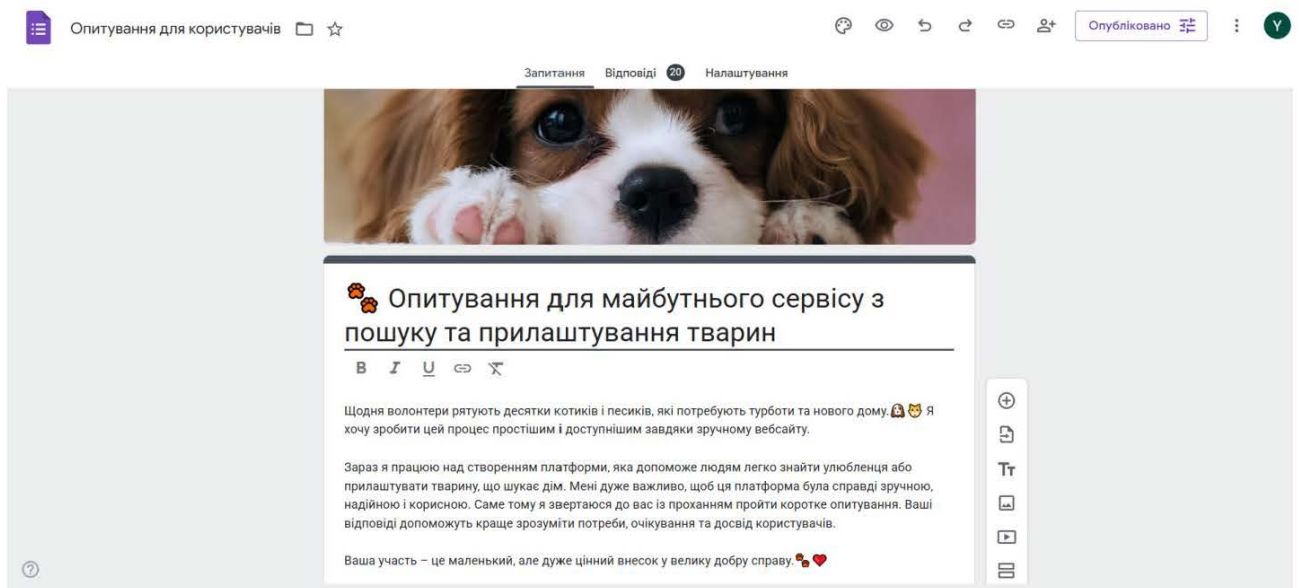


Рисунок 3.2 – Опитування у Google Forms для майбутніх користувачів

Загалом отримано такі результати дослідження та можна зробити певні висновки:

- 60% опитаних – особи віком 18-25 років, тобто основна аудиторія – молоді люди. Саме тому необхідно створити зручний та сучасний дизайн, забезпечити швидкий доступ та адаптивний інтерфейс;
- половина респондентів з Дніпропетровської області, а решта з інших регіонів України (зокрема Донецької, Київської, Львівської, Миколаївської, Харківської, Хмельницької, Вінницької областей). Тому важливо, щоб вебдодаток був універсальним, працював по всій країні з можливістю фільтрації оголошень за областями та містами;

– 40% проживає у великих містах, 25% – у невеликих містах або селищах, 15% – у селах, 20% – за межами України. Враховуючи це, можна передбачити можливість доставки тварини або в майбутньому додати англійську версію сайту;

– 90% респондентів мають домашніх тварин, решта мали досвід у минулому. Це свідчить про високий рівень залученості користувачів у цю тему;

– 20% брали тварину з притулку або від волонтерів, 45% мали досвід прилаштування, 10% шукали власника загубленої тварини, 25% допомагали тваринам у різний спосіб. Враховуючи це, є потреба зробити декілька типів оголошень: «У пошуках дому», «Загублений», «Знайдений»;

– найпопулярніші сервіси, які вже використовувалися для пошуку/прилаштування тварин: соціальні мережі (50%), чати у Telegram або Viber (45%), особисті знайомства (40%). Менш популярні – сайти притулків, OLX, ветеринарні клініки. В майбутньому можна додати можливість ділитися оголошеннями в соцмережах або ж розробити окремий бот;

– 65% опитаних готові шукати або прилаштовувати тварин, 55% – зацікавлені у пошуку власника загубленої тварини. 45% просто хотіли б підтримати ініціативу. Тому платформа має підтримувати різні сценарії взаємодії та можливість розміщувати різні типи оголошень;

– 55% зацікавлені в собаках, 35% – у котах, ще 35% – у будь-яких тваринах. Тому доцільно буде зробити фільтрацію за типом тварини;

– для 60% респондентів важливо, аби тварина знаходилася у їх місті чи області, 20% готові їхати чи організовувати доставку. Це знову підтверджує важливість фільтрації за регіоном;

– основні критерії при виборі тварини: фото, вік, стать, характер, порода, стан здоров'я, вакцинація, поведінка (важливі для 65–85% респондентів). Менш важливі: розмір тварини, місце перебування, контакти. Виходячи з цього, оголошення мають бути структурованими, з усією необхідною інформацією;

– 95% опитаних користувачів надають перевагу месенджерам (Telegram, Viber, WhatsApp тощо) як засобу зв'язку. 40% надали перевагу телефону, 25% –

формі на сайті. Електронна пошта не викликала зацікавленості. Саме тому доцільно буде при створенні оголошення запропонувати залишити свої номер телефону та месенджер для зв'язку;

— на пропозицію обрати, які функції на сайті були б корисними, обирали такі варіанти: фільтри для пошуку, швидкий зв'язок із волонтером, створення оголошення про зникнення тварини, окремий розділ для загублених/знайдених тварин, особистий кабінет користувача. Також людей цікавили такі варіанти: блок із порадами, щодо догляду, мапа з розташуванням тварин/притулків, авторизація через Google/Facebook, збереження оголошень. Ці побажання варто врахувати в майбутніх оновленнях.

	A	B	C	D	E	F	G
1	Позначка часу	Скільки вам років?	Звідики ви?	Де ви проживаєте?	Чи є у вас зараз домашні тварини?	Чи мали ви досвід пошуку або прилаштув.	Якими сервісами ви користувалися під час пошуку?
2	28.04.2025 10:21:48	36-45	Дніпропетровська обл	У селі	Так, маю одну або кілька тварин	Прилаштовував тварину, яка шукала дім	Взагалі не користувався
3	28.04.2025 10:55:48	26-35	Дніпропетровська обл	У селі	Так, маю одну або кілька тварин	Прилаштовував тварину, яка шукала дім	Чати/групи у Telegram або Viber
4	28.04.2025 11:21:18	18-25	Дніпропетровська обл	У невеликому місті або в селі	Так, маю одну або кілька тварин	Прилаштовував тварину, яка шукала дім	Особисті знайомства
5	28.04.2025 16:25:02	26-35	Миколаївська область	У великому місті (понад 100 тис. мешк.)	Ні, але мав раніше	Прилаштовував тварину, яка шукала дім	Соцмережі (Facebook, Instagram), Чати/групи
6	28.04.2025 17:22:51	18-25	Київська область	У невеликому місті або в селі	Так, маю одну або кілька тварин	Брав тварину з притулку або від волонтерів, П	Соцмережі (Facebook, Instagram), Чати/групи
7	28.04.2025 18:06:01	26-35	Дніпропетровська обл	У великому місті (понад 100 тис. мешк.)	Так, маю одну або кілька тварин	Брав тварину з притулку або від волонтерів, П	Соцмережі (Facebook, Instagram), Чати/групи
8	28.04.2025 20:19:09	26-35	Київська область	У невеликому місті або в селі	Так, маю одну або кілька тварин	Брав тварину з притулку або від волонтерів, П	Соцмережі (Facebook, Instagram), Чати/групи
9	30.04.2025 14:10:19	18-25	Дніпропетровська обл	У невеликому місті або в селі	Так, маю одну або кілька тварин	Не мав досвіду, але планую	Соцмережі (Facebook, Instagram), Чати/групи
10	30.04.2025 14:12:08	18-25	Дніпропетровська обл	У великому місті (понад 100 тис. мешк.)	Так, маю одну або кілька тварин	Прилаштовував тварину, яка шукала дім, Допс	Соцмережі (Facebook, Instagram), Сайти притулків
11	30.04.2025 14:21:33	18-25	Дніпропетровська обл	У великому місті (понад 100 тис. мешк.)	Так, маю одну або кілька тварин	Допомагав тваринам, але не забирав додому	Чати/групи у Telegram або Viber, Особисті знайомства
12	30.04.2025 17:17:19	18-25	Дніпропетровська обл	У невеликому місті або в селі	Так, маю одну або кілька тварин	Брав тварину з притулку або від волонтерів, П	Особисті знайомства
13	23.05.2025 16:48:36	26-35	м. Київ	У великому місті (понад 100 тис. мешк.)	Так, маю одну або кілька тварин	Прилаштовував тварину, яка шукала дім	Соцмережі (Facebook, Instagram), OLX
14	24.05.2025 16:01:24	18-25	Дніпропетровська обл	За межами України	Так, маю одну або кілька тварин	Не мав досвіду, але планую	Соцмережі (Facebook, Instagram), Чати/групи
15	24.05.2025 17:19:29	18-25	Вінницька область	За межами України	Так, маю одну або кілька тварин	Не мав досвіду, але планую	Взагалі не користувався

Рисунок 3.3 – Результати опитування у вигляді таблиці

Тепер на основі отриманих результатів сформуємо основні вимоги до функціональності майбутнього вебдодатку:

— головна сторінка повинна містити коротку інформацією про платформу та мати інтуїтивну навігацію до основних розділів;

— каталог оголошень має відображати усі опубліковані оголошення з можливістю фільтрації за такими параметрами: стать, область, місто, тип оголошення, розмір, вік, здоров'я, особливі обставини;

- форма створення оголошення повинна дати можливість створити оголошення, завантаживши фото та додавши опис тварини, контакти автора, місто та тип оголошення («У пошуках дому», «Загублений», «Знайдений»);

- сторінка оголошення має містити повну інформацію про тварину: ім'я, місто, фотографії, опис, дата публікації, контактні дані для зв'язку з автором оголошення. Додатково повинні відображатися ще 4 випадкових оголошення, що можуть зацікавити користувача;

- особистий кабінет повинен надавати можливість керування власними оголошеннями (редагування, видалення, зміна статусу на «Дім знайдено»).

Загалом, проведене дослідження та зібрана аналітика підтвердила актуальність та затребуваність платформи. Це дасть можливість створити корисний, інтуїтивний і персоналізований сервіс, який відповідатиме реальним потребам користувачів.

Тепер можна використати результати дослідження для створення карти сайту (site map) – схематичного плану структури вебдодатку, який відображає логіку розміщення сторінок, їхню ієрархію та взаємозв'язки. Карта сайту допоможе визначити, які сторінки потрібно розробити та допоможе спланувати логічну структуру проєкту (рис. 3.4).



Рисунок 3.4 – Карта сайту

3.2 Проектування UX/UI дизайну

Розробка дизайну інтерфейсу розпочалася з того, що було створено сценарії поведінки користувачів та визначено логіку взаємодії з платформою. Проаналізовано основні задачі, які користувач має виконати на сайті. На основі цього складено Customer Journey Map (рис. 3.5). Це карта користувача, який шукає тварину, що візуалізує його шлях під час взаємодії з продуктом. Вона допомагає зрозуміти, які дії, емоції та труднощі виникають у користувача на кожному етапі. Так, можна виявити слабкі місця у структурі та функціональності, покращити навігацію сайту та створити більш зручний і ефективний користувацький досвід.

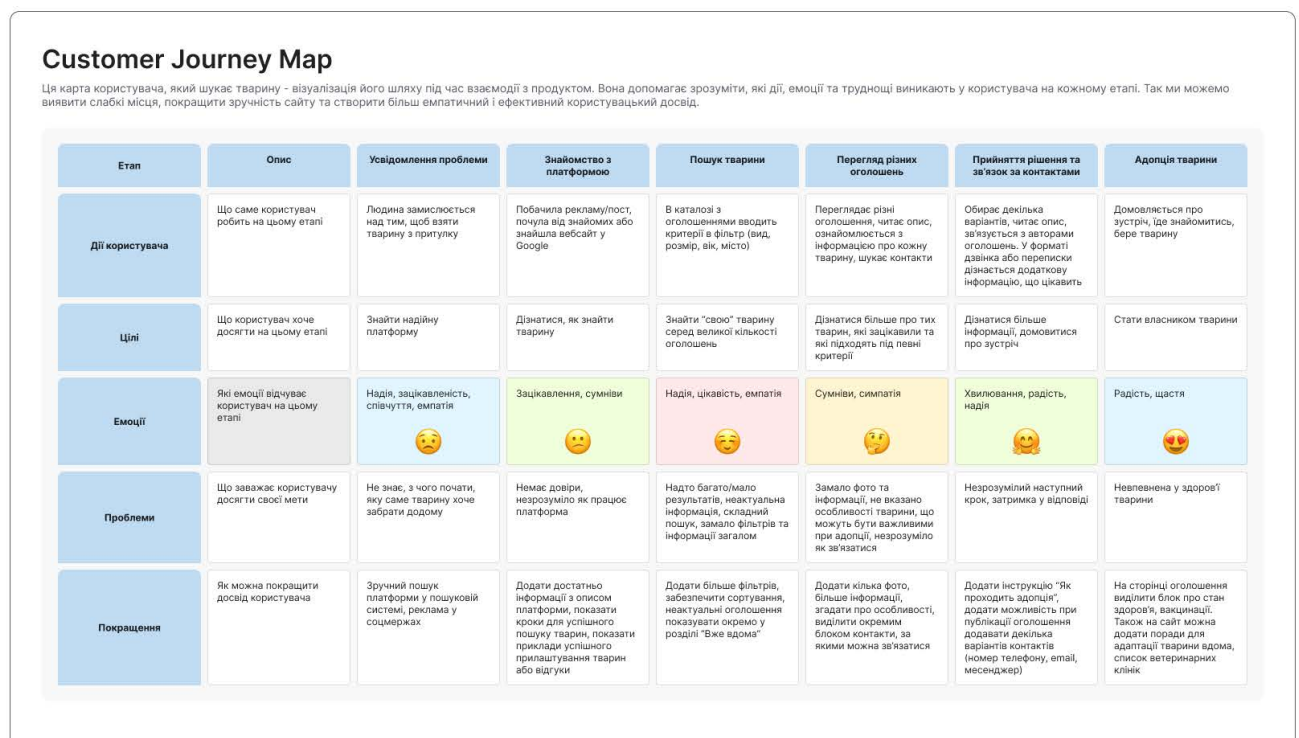


Рисунок 3.5 – Customer Journey Map

Наступним кроком було створення у графічному редакторі Figma wireframes – чорно-білих схематичних макетів сторінок, які демонструють базову структуру інтерфейсу та логіку взаємозв'язків між його елементами (рис. 3.6).

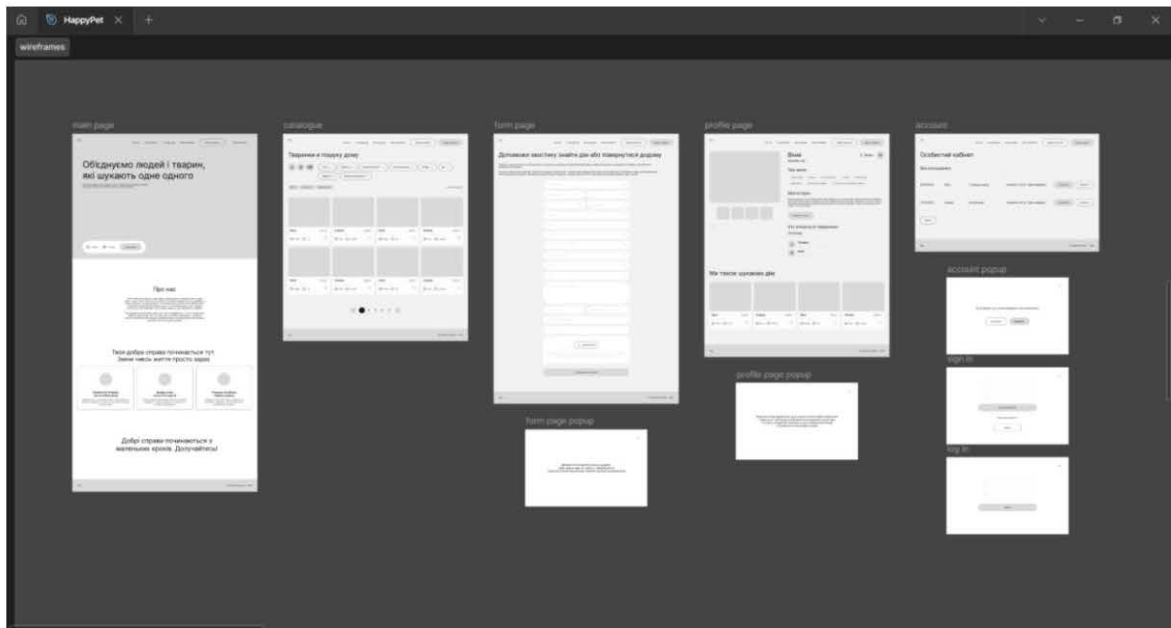


Рисунок 3.6 – Розроблені wireframes сторінок вебдодатку у Figma

На основі wireframes було розроблено повноцінний UI-дизайн у мінімалістичному стилі, який дозволить користувачеві зосередитися на головному – тваринах. Основний акцент був зроблений на великих зображеннях, легкій для читання типографіці, чіткій структурі сторінок та збалансованому поєднанню кольорів. Інтерфейс був спроектований відповідно до принципів доступності та адаптивності (рис. 3.7).

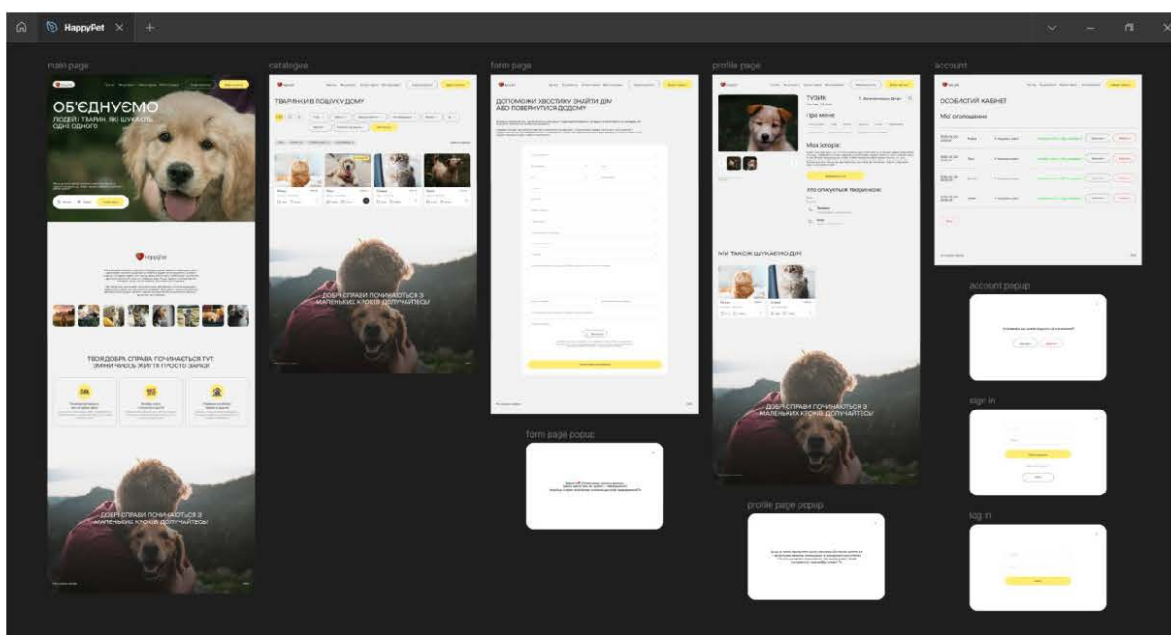


Рисунок 3.7 – Розроблений UI-дизайн сторінок вебдодатку у Figma

Для тестування логіки взаємодії з інтерфейсом та демонстрації роботи окремих сценаріїв у Figma варто зробити прототипи інтерактивними. Це дозволить перевірити зручність навігації, послідовність переходів між екранами та швидко внести необхідні корективи до дизайну ще до початку реалізації (рис. 3.8).

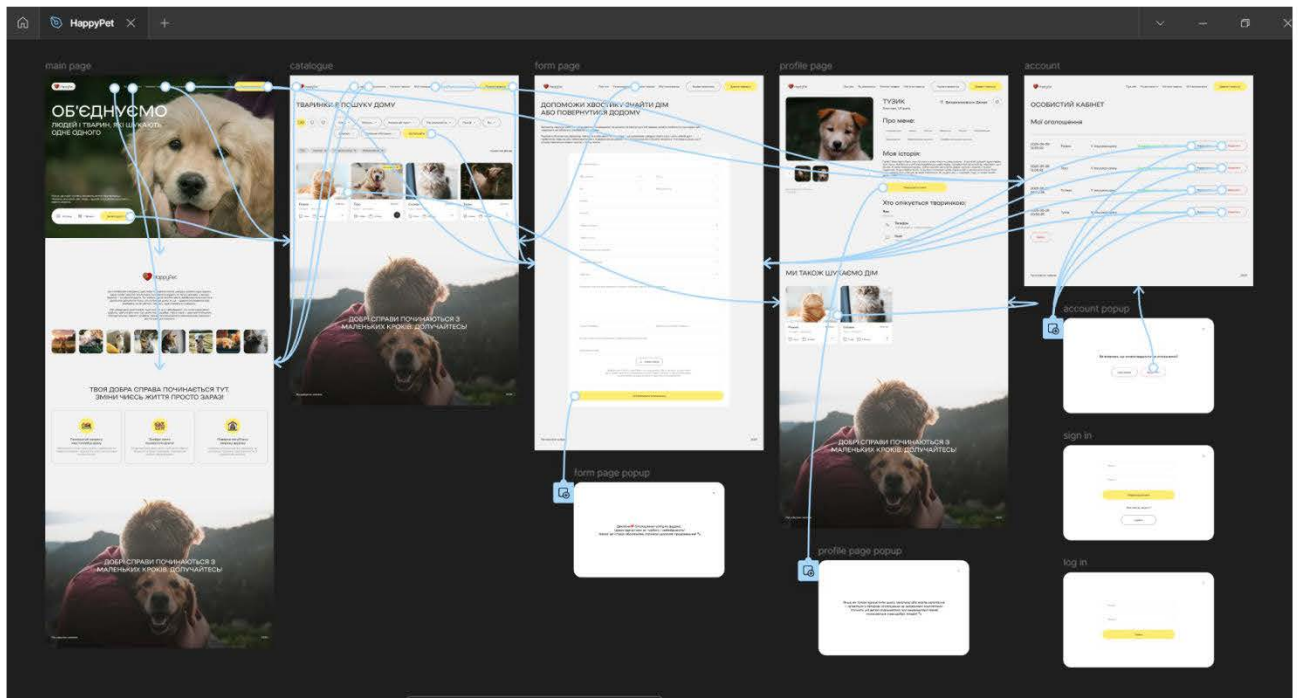


Рисунок 3.8 – Прототипування взаємодії користувача з інтерфейсом у Figma

Під час створення інтерфейсу враховувалися основні принципи UX-дизайну. Вони допомагають зробити вебдодаток зручним, зрозумілим і приємним у використанні. У роботі було враховано деякі відомі правила, які запропонував фахівець Якоб Нільсен [9]. Ці правила часто використовують для оцінки зручності сайтів та додатків. Зокрема, в дизайні було реалізовано наступні принципи:

- видимість статусу системи: інтерфейс повідомляє користувача про результати його дій (повідомлення про успішне додавання оголошення);
- відповідність між системою та реальним світом: термінологія, що використовується є зрозумілою для цільової аудиторії, інформація подається у природному та логічному порядку;

- контроль і свобода користувача: у користувача є можливість легко скасувати дію або повернутися на попередній етап, зокрема – кнопки «назад», «скасувати» та чітка навігація;
- консистентність і стандарти: усі елементи інтерфейсу – кнопки, шрифти, іконки є візуально узгодженими;
- естетичний і мінімалістичний дизайн: інтерфейс виглядає чисто та охайно, без зайвих елементів. Залишилася найважливіша інформація, щоб не перевантажувати користувача зайвим. Такий підхід допомагає краще зосередитися на основних діях – на перегляді тварин і виборі улюбленця.

3.3 Реалізація вебдодатку: Frontend та Backend розробка

Після завершення етапу проєктування розпочалася реалізація клієнтської (Frontend) та серверної (Backend) частин вебдодатку.

Frontend відповідає за візуальне відображення і взаємодію користувача з інтерфейсом. Тож, на початковому етапі було створено структуру HTML-документів для всіх основних сторінок вебдодатку: головної сторінки (рис. 3.9), каталогу оголошень, окремої сторінки оголошення, сторінки додавання оголошення, а також особистого кабінету.

Завдяки семантичній верстці HTML можна логічно розмістити контент на сторінці, використовуючи при цьому такі елементи, як `<header>`, `<main>`, `<section>`, `<footer>`. Семантичні теги допомагають пошуковим роботам краще розуміти структуру й зміст сторінки, роблять HTML-код більш читабельним і структурованим, полегшують застосування стилів і скриптів.

Отже, у результаті вдалося отримати зрозумілу структуру для користувачів та для пошукових систем.

Для додавання інтерактивності використовувалась мова JavaScript. Скрипти обробляють різні дії: відкриття та закриття мобільного меню, адаптивне позиціонування елементів при зміні розміру вікна, роботу слайдерів (головного і мініатюр), натискання кнопок, валідацію форм, адаптивну поведінку хедеру, показ модальних вікон. Реалізовано вибір категорії на головній сторінці, роботу чекбоксів, завантаження зображень, валідацію та відправку форми, а також логіку фільтрації оголошень у каталозі за типом тварини (кішка, собака або обидва варіанти) та інших критеріїв. Код, що відповідає за реалізацію основного слайдера з фотографіями тварини на сторінці оголошення зображень на рисунку 3.11.

```

1  const thumbnailSlider = new Swiper('.profile_slider-thumbnail', {
2    slidesPerView: 2,
3    spaceBetween: 12,
4    watchSlidesProgress: true,
5    slideToClickedSlide: true,
6    breakpoints: {
7      480: {
8        slidesPerView: 3,
9      },
10     768: {
11       slidesPerView: 4,
12     },
13     1000: {
14       slidesPerView: 3,
15     },
16     1280: {
17       slidesPerView: 4,
18     },
19   },
20 });
21
22 const mainSlider = new Swiper('.profile_slider', {
23   slidesPerView: 1,
24   spaceBetween: 20,
25   navigation: {
26     nextEl: '.profile_slider-next',
27     prevEl: '.profile_slider-prev',
28   },
29   thumbs: {
30     swiper: thumbnailSlider,
31   },
32 });
33

```

Рисунок 3.11 – Частина реалізації основного слайдеру (mainSlider) та слайдеру мініатюр (thumbnailSlider) на JavaScript

Паралельно з клієнтською частиною розроблялася і серверна логіка на мові програмування PHP, що забезпечує взаємодію між клієнтом та сервером, а також відповідає за обробку запитів, роботу з базою даних та виконання логіки. Наприклад, скрипт auth.php (рис. 3.12) реалізує автентифікацію користувача – отримує Email і пароль, перевіряє їхню коректність, шукає відповідного

користувача в базі даних, звіряє хеш пароля, і якщо все правильно – зберігає ідентифікатор користувача в сесію та повертає підтвердження успішної авторизації. У разі помилок, наприклад, якщо неправильні дані, відсутній користувач або є проблеми з базою, повертається відповідне повідомлення про помилку.

```

1  <?php
2
3  header('Content-Type: application/json');
4  session_start();
5
6  require '../includes/Base.php';
7
8  $data = json_decode(file_get_contents('php://input'), true);
9  $email = $data['email'] ?? '';
10 $password = $data['password'] ?? '';
11
12 if (!filter_var($email, FILTER_VALIDATE_EMAIL) || $password === '') {
13     echo json_encode(['success' => false, 'error' => 'Введіть email та пароль.']);
14     exit;
15 }
16
17 try {
18     $user = Base::get('users', ['email' => $email]);
19
20     if (!$user || !password_verify($password, $user['password_hash'])) {
21         echo json_encode(['success' => false, 'error' => 'Невірний email або пароль.']);
22         exit;
23     }
24
25     $_SESSION['user_id'] = $user['id'];
26
27     echo json_encode(['success' => true, 'redirect' => 'index.php']);
28 } catch (Exception $e) {
29     echo json_encode(['success' => false, 'error' => 'Помилка сервера.']);
30 }
31

```

Рисунок 3.12 – PHP-скрипт – реалізація автентифікації користувача

Сторінка `profile.php` відповідає за відображення інформації про оголошення тварини, яка шукає дім (рис. 3.13). Цей файл встановлює заголовок сторінки, отримує ідентифікатор оголошення з параметрів URL (`$_GET['id']`), завантажує відповідні дані з бази (вся інформація про тварину, фотографії) перевіряє існування оголошення та виводить його деталі. Якщо оголошення з таким ID не знайдено, користувач перенаправляється на сторінку каталогу. На сторінці відображається слайдер із фотографіями, кличка тварини, місце розташування, вік, інформація про здоров'я (стерилізований, вакцинований, оброблений від паразитів) та особливі обставини (евакуйований з прифронтових територій, потребує тимчасового прихистку, з особливостями,

потребує особливого догляду, маленький, літнього віку), а також історія тваринки та інформація про людину, яка опікується нею.

```

1  <?php
2
3  $pageTitle = 'Тваринка в пошуку дому';
4
5  require __DIR__ . '/includes/head.php';
6  require __DIR__ . '/includes/functions.php';
7  require __DIR__ . '/includes/header.php';
8
9  $id = $_GET['id'];
10
11  $advertisement = Base::get('advertisement', $id);
12
13  if (!$advertisement) {
14      header('Location: catalog.php');
15      exit();
16  }
17
18  $photos = Base::query('select * from advertisement_photo where advertisement_id = ' . $id, [], 'arr');
19  $animal = $advertisement['animal_type'];
20  $dataAnimal = $animal === 'Собака' ? 'dog' : ($animal === 'Котик' ? 'cat' : '');
21  $region = Base::get('regions', $advertisement['region_id']);
22  $city = Base::get('cities', $advertisement['city_id']);
23

```

Рисунок 3.13 – Завантаження даних оголошення про тварину за ID

Також на сторінці оголошення перед футером показуються чотири випадкові інші активні оголошення (крім поточного) (рис. 3.14).

```

25
26  $randomAds = Base::query("SELECT a.*, p.path AS photo, r.name_uk AS region_name, c.name_uk AS city_name
27  FROM advertisement a LEFT JOIN (SELECT advertisement_id, path FROM advertisement_photo GROUP BY advertisement_id ) p
28  ON p.advertisement_id = a.id LEFT JOIN regions r ON r.id = a.region_id LEFT JOIN cities c ON c.id = a.city_id WHERE a.status = 1
29  and a.id != " . $id . " ORDER BY RAND() LIMIT 4", [], 'arr');
30

```

Рисунок 3.14 – Реалізація вибірки випадкових оголошень з бази даних, виключаючи поточне

Для роботи з базою даних MySQL створено окремий клас. Він забезпечує підключення до бази, виконання SQL-запитів та базові CRUD-операції (створення, читання, оновлення, видалення). Це дозволяє ефективно зберігати інформацію про користувачів, оголошення, фотографії та інші дані.

Окрім цього, реалізовані інші PHP-скрипти для обробки запитів із фронтенду, зокрема: скрипт авторизації, що приймає Email та пароль, перевіряє їх на правильність, знаходить користувача в базі, звіряє пароль і, якщо все вірно, авторизує його, зберігаючи ID в сесії; скрипт фільтрації оголошень, що приймає запит з параметрами фільтрації, формує SQL-запит з відповідними

умовами, отримує з бази даних оголошення, які підходять під ці умови, і потім повертає їх разом із загальною кількістю результатів; скрипт створення нового оголошення, що обробляє запит на створення оголошення про тварину від авторизованого користувача, зберігає дані в базу, завантажує фото та повертає відповідь з результатом; скрипт реєстрації нового користувача, що приймає Email і пароль, перевіряє їх коректність та унікальність, створює нового користувача з паролем у базі даних та зберігає його ID в сесію; скрипт оновлення оголошення, що перевіряє, чи користувач авторизований і чи належить саме йому оголошення з таким ID. Якщо так, оновлює запис в базі новими даними з форми.

В адмін-панелі хостингу на adm.tools було знайдено розділ, що відповідає за керування базами даних. Створено нову базу даних. Для їх керування було обрано phpMyAdmin – інструмент, що дозволяє працювати з базою даних через вебінтерфейс. У списку ліворуч була обрана раніше створена база, після чого відкрився інтерфейс, у якому є можливість створювати нові таблиці вручну або через SQL-запити. У вкладці «SQL» за допомогою запиту CREATE TABLE було створено таблицю users (рис. 3.15) з такими полями:

- id – унікальний ідентифікатор користувача (тип INT);
- email – електронна адреса користувача (тип VARCHAR, обмеження NOT NULL, та UNIQUE, щоб не дозволити дублікати);
- password_hash – збережений захешований пароль (тип VARCHAR);
- created_at – дата та час реєстрації (тип TIMESTAMP).

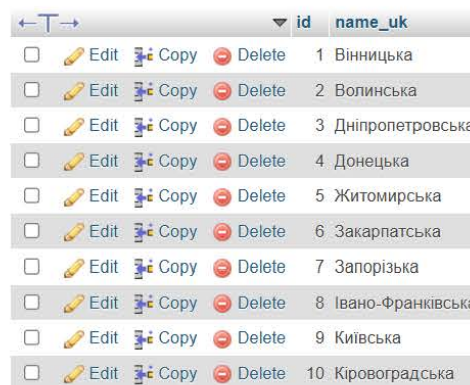
У результаті таблицю users успішно створено. Вона стала доступною для подальшої роботи з реєстрацією, авторизацією та управлінням користувачами через PHP-скрипти.



	id	email	password_hash	created_at
☐	1	leb.ya	\$2y\$10\$VKSid/C4oLyUaMcZUhl.OdplJavy	2025-05-21 23:33:24

Рисунок 3.15 – Таблиця users, де зберігаються дані зареєстрованих користувачів

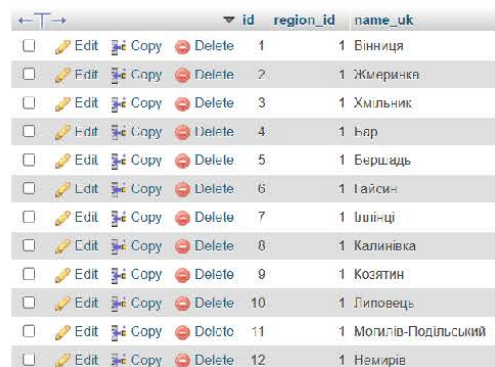
За таким самим принципом було створено таблиці: advertisement – для зберігання всіх оголошень з інформацією, яку користувач вказує у формі створення, а також advertisement_photo для зберігання усіх зображень, що прикріплюються до оголошень. Окремо були створені таблиці regions (рис. 3.16) та cities (рис. 3.17), у яких мають зберігатися області та міста України. Щоб не додавати вручну ці дані, було підготовлено окремий PHP-скрипт, де на основі відкритих джерел (geonames.org) були сформовані масиви назв областей та міст.



	id	name_uk
☐ Edit Copy Delete	1	Вінницька
☐ Edit Copy Delete	2	Волинська
☐ Edit Copy Delete	3	Дніпропетровська
☐ Edit Copy Delete	4	Донецька
☐ Edit Copy Delete	5	Житомирська
☐ Edit Copy Delete	6	Закарпатська
☐ Edit Copy Delete	7	Запорізька
☐ Edit Copy Delete	8	Івано-Франківська
☐ Edit Copy Delete	9	Київська
☐ Edit Copy Delete	10	Кіровоградська

Рисунок 3.16 – Фрагмент таблиці regions, де зберігаються області України

Скрипт зчитував ці масиви та через SQL-запити та додав інформацію до таблиць regions та cities, після чого його було завантажено на сервер. Таблиця cities також містить зовнішній ключ до таблиці regions. Це забезпечує зв'язок між містом та його областю.



	id	region_id	name_uk
☐ Edit Copy Delete	1	1	Вінниця
☐ Edit Copy Delete	2	1	Жмеринка
☐ Edit Copy Delete	3	1	Хмельник
☐ Edit Copy Delete	4	1	Іар
☐ Edit Copy Delete	5	1	Бердичів
☐ Edit Copy Delete	6	1	Тайсін
☐ Edit Copy Delete	7	1	Ішніці
☐ Edit Copy Delete	8	1	Калинівка
☐ Edit Copy Delete	9	1	Козятин
☐ Edit Copy Delete	10	1	Липовець
☐ Edit Copy Delete	11	1	Молішів-Подільський
☐ Edit Copy Delete	12	1	Немирів

Рисунок 3.17 – Фрагмент таблиці cities, де зберігаються населені пункти України

Після створення всіх необхідних таблиць, було здійснено перевірку кожної з них. У вкладці «Структура» перевірено назви та типи полів, а у вкладці «Огляд» – наявність доданих даних. Зокрема, перевірено структуру таблиці advertisement (рис. 3.18), у якій зберігаються всі дані кожного оголошення.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2 user_id	int(11)			No	None			Change Drop More
☐	3 type_post	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	4 animal_type	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	5 gender	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	6 age	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	7 age_unit	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	8 size	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	9 alias	varchar(500)	utf8_general_ci		Yes	NULL			Change Drop More
☐	10 region_id	int(255)			Yes	NULL			Change Drop More
☐	11 city_id	int(11)			Yes	NULL			Change Drop More
☐	12 caretaker	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	13 Sterilized	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	14 Vaccinated	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	15 Treated_for_parasites	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	16 health	varchar(1000)	utf8_general_ci		Yes	NULL			Change Drop More
☐	17 Evacuated_from_frontline_areas	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	18 Needs_temporary_shelter	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	19 With_features	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	20 Elderly	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	21 Requires_special_care	varchar(500)	utf8_general_ci		Yes	NULL			Change Drop More
☐	22 Small_up6m	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	23 special_circumstances	varchar(1000)	utf8_general_ci		Yes	NULL			Change Drop More
☐	24 message	varchar(5000)	utf8_general_ci		Yes	NULL			Change Drop More
☐	25 tel	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	26 tel_second	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	27 socials	varchar(500)	utf8_general_ci		Yes	NULL			Change Drop More
☐	28 user_name	varchar(255)	utf8_general_ci		Yes	NULL			Change Drop More
☐	29 status	int(11)			No	None			Change Drop More
☐	30 created_at	timestamp			No	CURRENT_TIMESTAMP			Change Drop More

Рисунок 3.18 – Структура таблиці cities, де зберігаються населені пункти України

Отже, у результаті перевірки встановлено, що всі оголошення, користувачі, області, міста та зображення будуть коректно зберігатись у відповідних таблицях бази даних. Наприкінці у файлі Base.php було реалізовано підключення до бази даних через раніше збережені дані (назва БД, логін, пароль, хост). Тож, структура бази виявилася повністю готовою для подальшої

взаємодії з фронтендом та бекендом сайту, зокрема – для реалізації функціоналу реєстрації, авторизації, створення та перегляду оголошень. На рисунку 3.19 представлено загальну структуру бази даних із усіма таблицями.



Рисунок 3.19 – Загальна структура бази даних вебдодатку

3.4 Результат реалізації вебдодатку та рекомендації, щодо подальшого вдосконалення

У результаті розробки було створено вебдодаток Happy Pet, який дозволяє розміщувати та шукати оголошення про прилаштування безпритульних тварин (Додаток В, Додаток Д). Усі заплановані функціональні можливості були успішно реалізовані. Вебдодаток відповідає всім основним вимогам, що були визначені на етапі проєктування. Користувачі можуть переглядати каталог тварин, використовувати фільтри для пошуку, створювати облікові записи, додавати оголошення та керувати ними – редагувати, видаляти або змінювати статус у власному кабінеті. Інтерфейс вебдодатку сучасний, лаконічний, зручний у використанні та відповідає принципам UX/UI дизайну.

Для перевірки системи проведемо тестування реалізованого функціоналу:

1) Реєстрація та авторизація.

На головній сторінці (рис. 3.20) натискаємо кнопку «Зареєструватися».

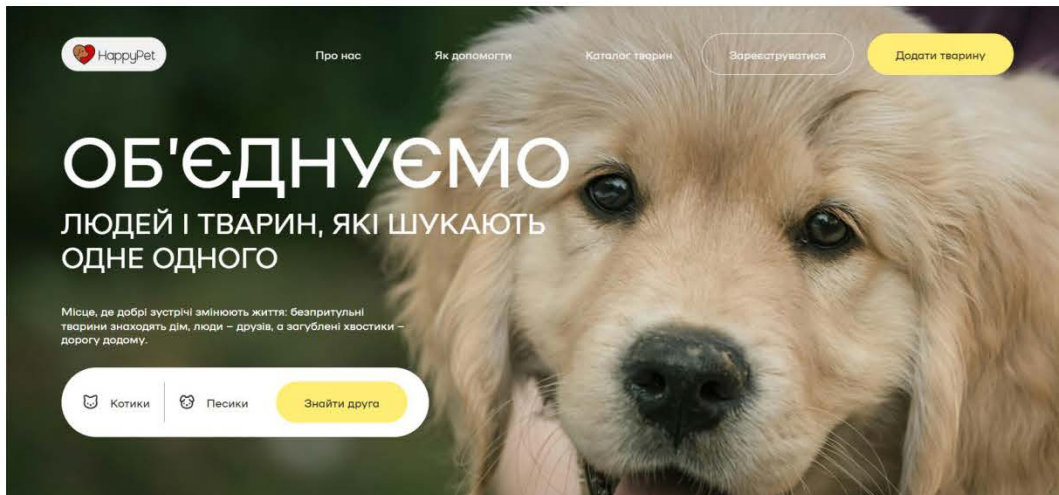


Рисунок 3.20 – Головна сторінка

Далі вводимо електронну пошту та пароль (рис. 3.21). Якщо обліковий запис уже існує – входимо в систему.

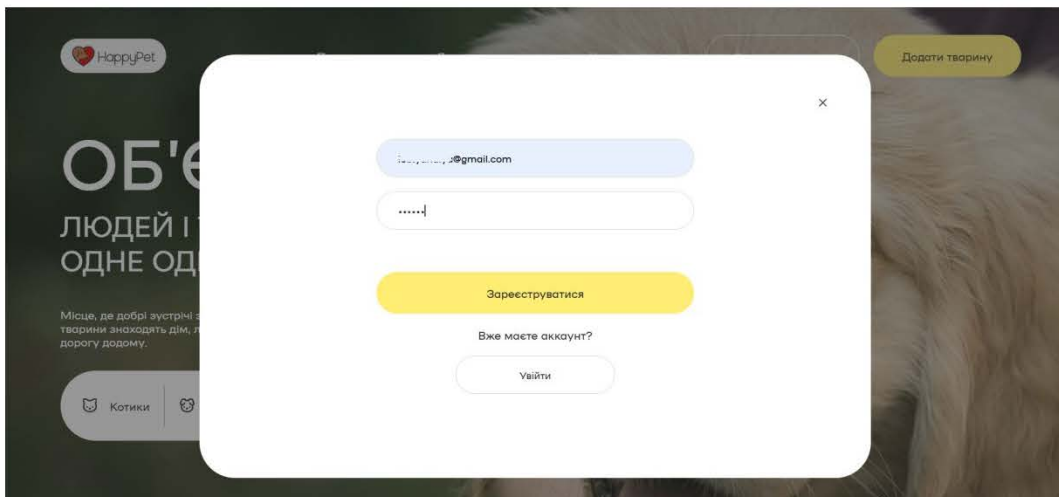


Рисунок 3.21 – Вікно реєстрації/входу

У таблиці users з'являється новий запис з даними нового користувача (рис. 3.22).

				id	email	password_hash	created_at
☐	Edit	Copy	Delete	1	leb.ye	\$2y\$10\$/VKSid/C4oLyUaMcZUhl.OdplJavy	2025-05-21 23:33:24
☐	Edit	Copy	Delete	3	leb.ye	\$2y\$10\$KupUvKJW/YDqWD1fO7x.4u7JZY	2025-05-28 23:35:03

☐ Check all With selected: Edit Copy Delete Export

Рисунок 3.22 – Таблиця users з новим доданим користувачем

2) Створення оголошення.

Після авторизації натискаємо кнопку «Додати оголошення» та заповнюємо поля форми вказуючи детальну інформацію про тварину: тип оголошення, вид тварини, стать, вік, розмір, кличку, область, місто, хто піклується про тварину, здоров'я, особливі обставини (рис. 3.23).

The screenshot shows the 'Додати тварину' (Add Pet) form on the HappyPet website. The form includes the following fields:

- У пошуках дому (In search of home) - dropdown menu
- Собака (Dog) - dropdown menu
- Хлопчик (Boy) - dropdown menu
- 1.5 - text input field
- Роки (Years) - dropdown menu
- Маленький (Small) - dropdown menu
- Тузик (Tuzik) - text input field
- Дніпропетровська (Dnipropetrovsk) - dropdown menu
- Дніпро (Dnipro) - dropdown menu
- Волонтер (Volunteer) - dropdown menu

Рисунок 3.23 – Форма створення оголошення

Додамо також короткий опис – розповідь про тварину, контактну особу, вкажемо номери телефонів та інші засоби для зв'язку. Додатково можна завантажити до 10 фотографій. Після заповнення всіх полів, натискаємо кнопку «Опублікувати оголошення» (рис. 3.24).

The screenshot shows the photo upload section of the form. It includes a 'Завантажити' (Upload) button with an upward arrow icon. Below it, the following text is displayed:

Додайте до 10 фото, зроблених на природному світлі, бажано на простому фоні, щоб увага була зосереджена саме тварині. Зніміть у кількох ракурсах – це допоможе краще показати її характер та особливості.

Below the text are two small square thumbnails showing photos of dogs. At the bottom of the section is a large yellow button labeled 'Опублікувати оголошення' (Publish advertisement).

Рисунок 3.24 –Завантаження зображень при створенні оголошення

Отримуємо повідомлення про успішну публікацію (рис. 3.25).

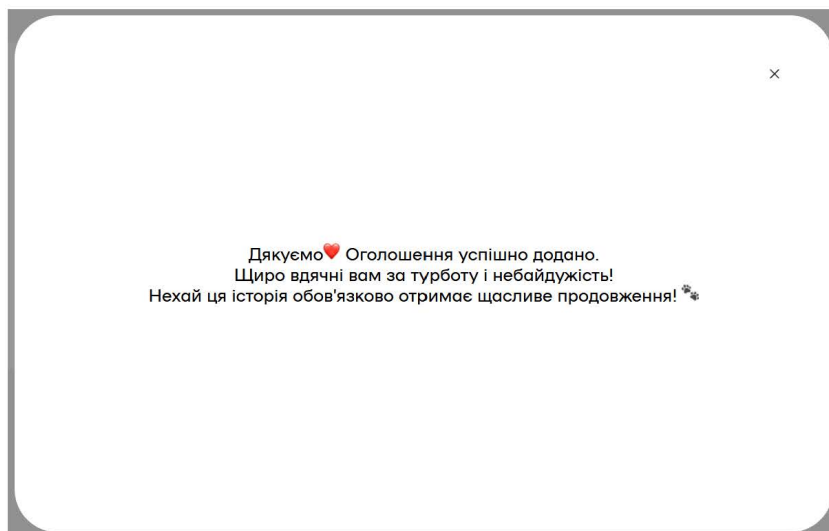


Рисунок 3.25 – Повідомлення про успішну публікацію

3) Відображення оголошення.

Опубліковане оголошення відображається в каталозі у вигляді картки (рис. 3.26). Відкривши його, опиняємось на сторінці цього оголошення (рис. 3.27).

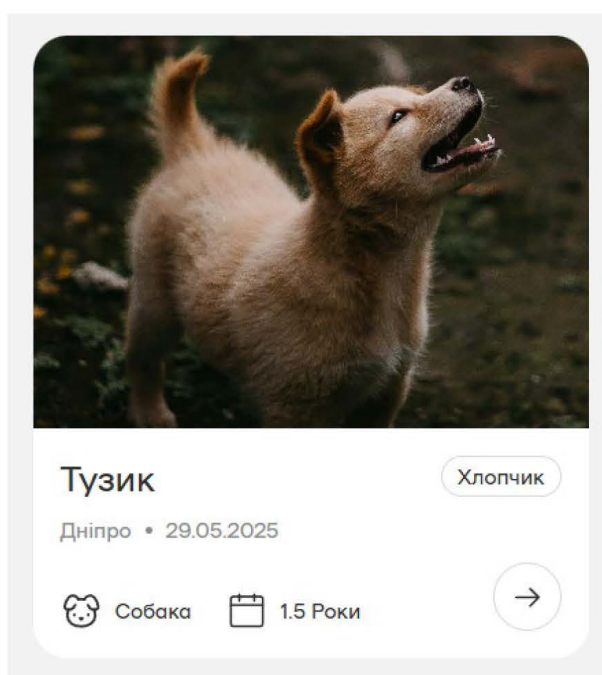


Рисунок 3.26 – Опубліковане оголошення

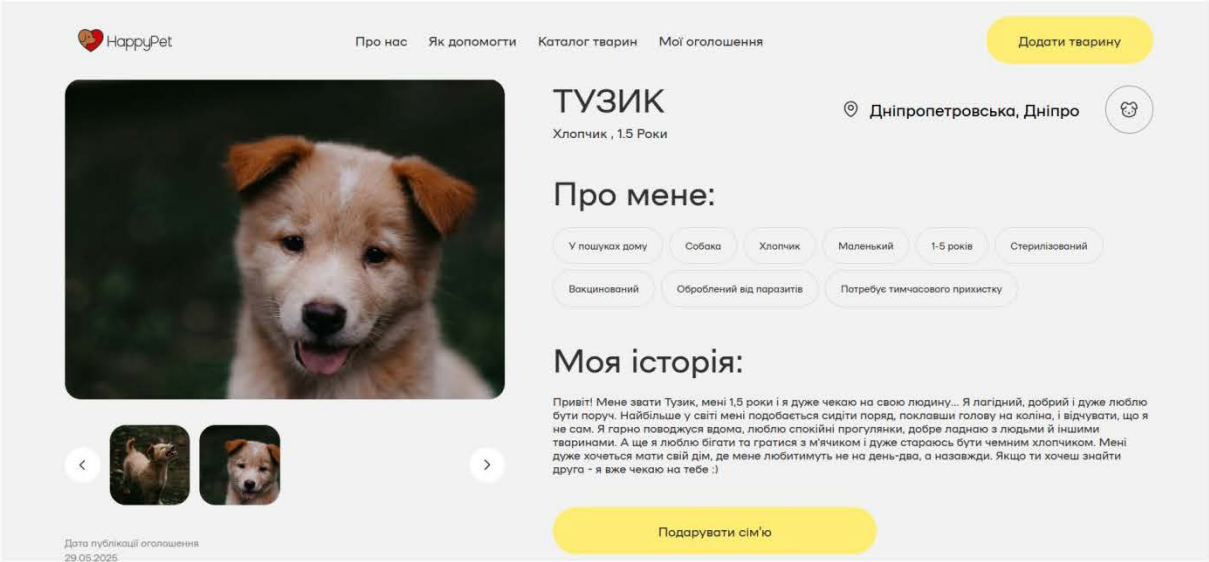


Рисунок 3.27 – Сторінка оголошення

Також його можна побачити в особистому кабінеті (рис. 3.28).

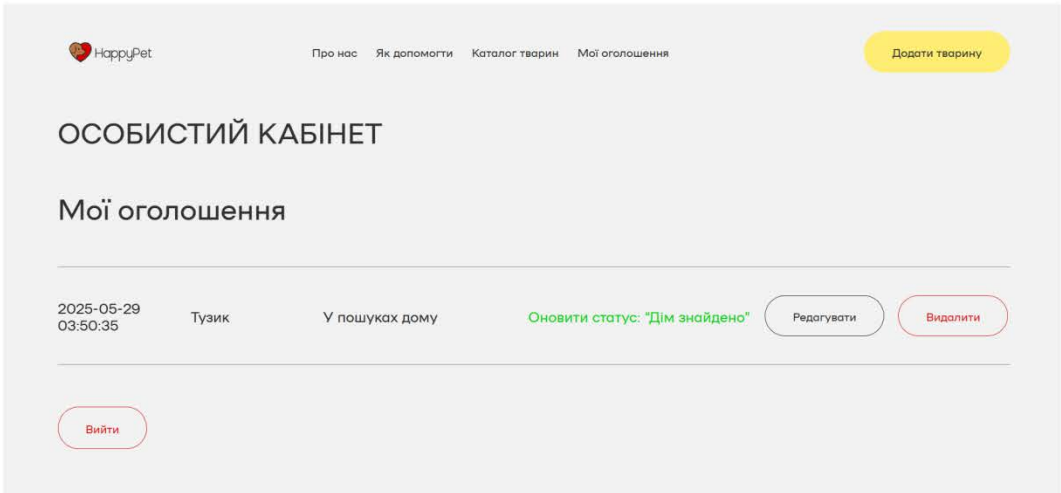


Рисунок 3.28 – Особистий кабінет з можливістю керувати оголошеннями

Крім того, повідомлення з'являється у таблиці advertisement бази даних (рис. 3.29).

	Id	user_id	type_post	animal_type	gender	age	age_unit	size	alias	region_id	city_id	caretaker	Sterilized	Vaccinated	Treated_for_pa
☐	9	1	У пошуках дому	Собака	Хлопчик	1.5	Роки	Маленький	Тузик	3	38	Волонтер	on	on	on
↑	With selected: ☐ Check all ☐ Edit ☐ Copy ☐ Delete ☐ Export														

Рисунок 3.29 – Таблиця advertisement з опублікованим оголошенням

При натисканні кнопки «Подарувати сім'ю» з'являється повідомлення із закликом зв'язатися з автором оголошення за вказаними контактами (рис. 3.30).

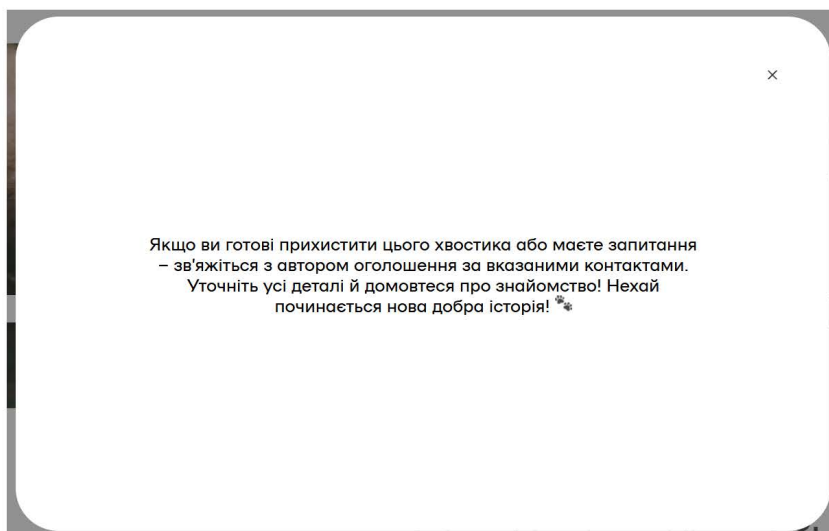


Рисунок 3.30 – Повідомлення про заклик до дії

4) Фільтрація оголошень.

Додамо ще декілька оголошень та протестуємо роботу фільтрів (рис. 3.31).

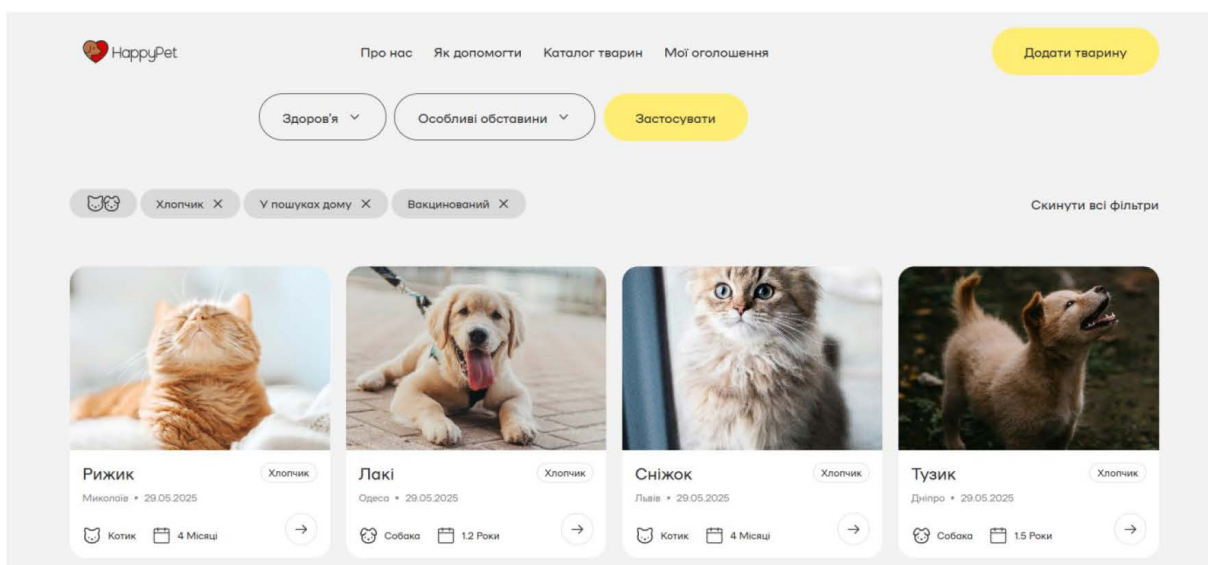
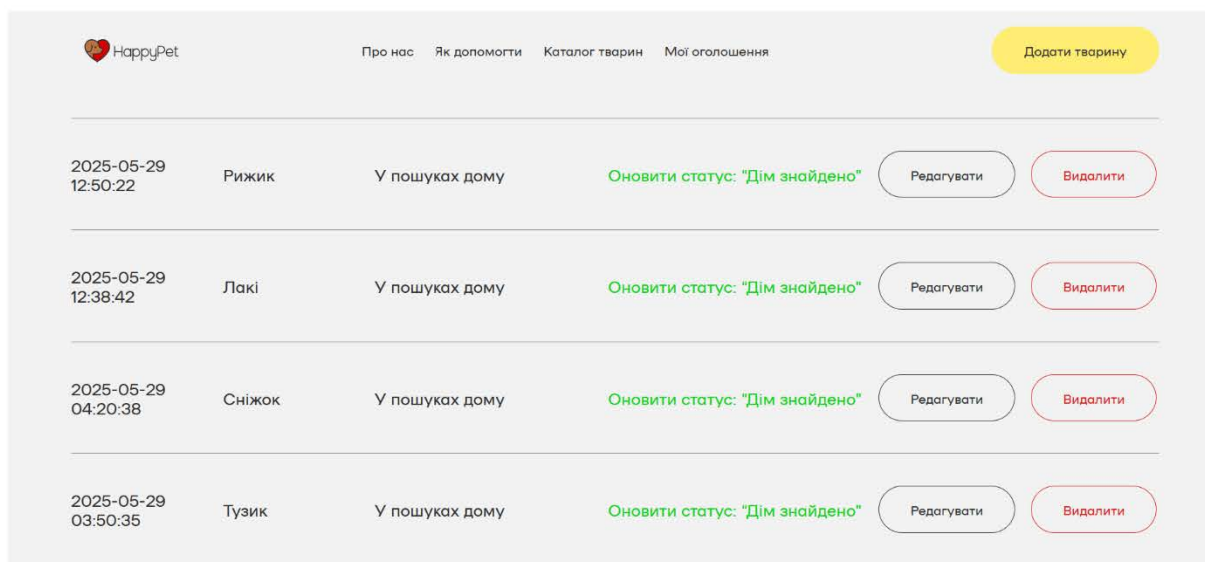


Рисунок 3.31 – Відфільтровані оголошення в каталозі

5) Редагування та зміна статусу оголошення.

У кабінеті користувача доступне редагування та видалення оголошень, а також можливість змінити статус на «Дім знайдено» (рис. 3.32), якщо тварина вже знайшла дім і оголошення неактуальне.



2025-05-29 12:50:22	Рижик	У пошуках дому	Оновити статус: "Дім знайдено"	Редагувати	Видалити
2025-05-29 12:38:42	Лакі	У пошуках дому	Оновити статус: "Дім знайдено"	Редагувати	Видалити
2025-05-29 04:20:38	Сніжок	У пошуках дому	Оновити статус: "Дім знайдено"	Редагувати	Видалити
2025-05-29 03:50:35	Тузик	У пошуках дому	Оновити статус: "Дім знайдено"	Редагувати	Видалити

Рисунок 3.32 – Перелік оголошень у кабінеті користувача

На відповідному оголошенні з'явиться значок «Дім знайдено» (рис. 3.33).

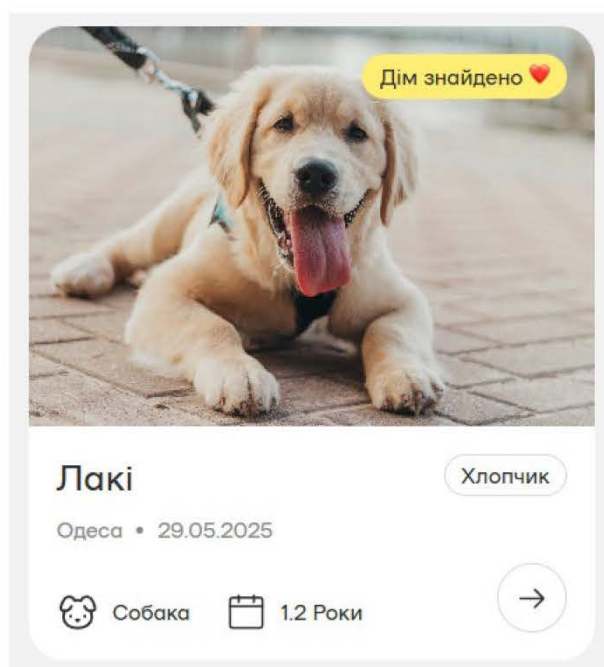


Рисунок 3.33 – Оголошення зі статусом «Дім знайдено»

6) Демонстрація адаптивної мобільної версії сайту.

Інтерфейс коректно адаптується під різні розміри екранів, зберігаючи при цьому зручність навігації та доступ до основних функцій. Користувачі можуть легко здійснювати реєстрацію, авторизацію, перегляд каталогу тварин, застосовувати фільтри, додавати та редагувати оголошення без втрати функціональності (рис. 3.34). Тому можна зробити висновок, що вебдодаток є повністю адаптивним і комфортним для користування як з десктопних, так і з мобільних пристроїв.

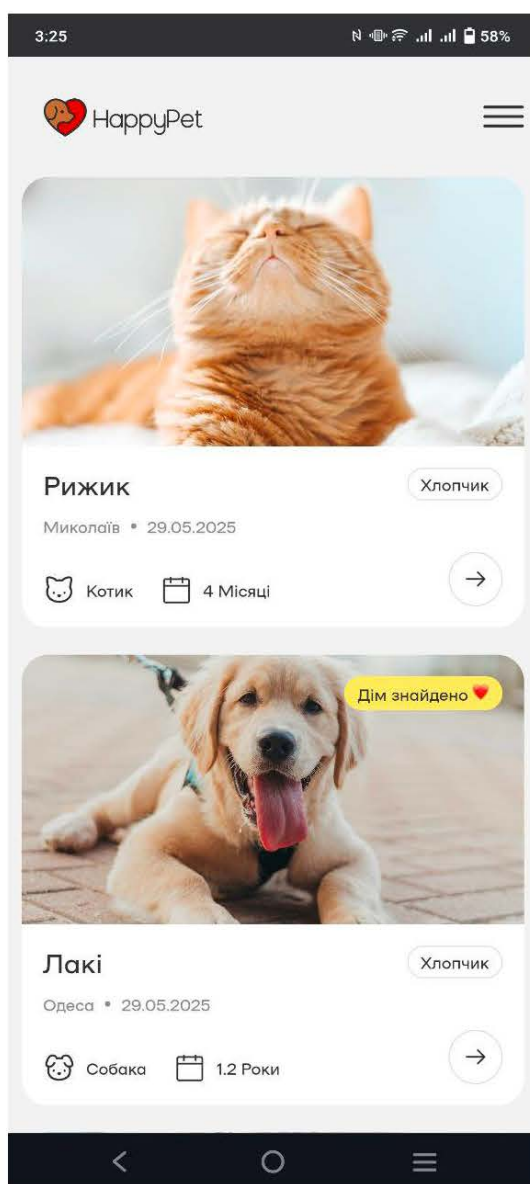


Рисунок 3.34 – Адаптивне відображення інтерфейсу на екрані смартфона

Отже, проведене тестування підтвердило, що всі основні функції працюють стабільно. У результаті перевірки основних сценаріїв встановлено, що вебдодаток виконує всі основні операції: реєстрацію, авторизацію, створення, редагування та фільтрацію оголошень.

У ході тестування було визначено кілька напрямів, які можуть покращити вебдодаток у майбутньому та підвищити його зручність та функціональність, а також залучити більшу кількість користувачів. Зокрема, пропонуються такі вдосконалення:

- додати можливість сортування оголошень за датою публікації;
- реалізувати функцію «Обране» для збереження оголошень;
- впровадити систему повідомлень або чат між користувачами;
- додати можливість авторизації через соціальні мережі;
- створити сторінку з порадами щодо догляду за тваринами;
- додати інтерактивну мапу з локаціями тварин або притулків;
- запровадити модерацію оголошень перед публікацією;
- реалізувати функцію пожертвувань для підтримки тварин і притулків.

Такі вдосконалення дозволять зробити платформу Happy Pet ще більш соціально значущою, зручною та функціональною для користувачів, а також будуть сприяти її розвитку.

3.5 Висновки до третього розділу

У третьому розділі кваліфікаційної роботи було детально розглянуто та описано всі основні етапи створення вебдодатку. Було визначено основні категорії цільової аудиторії. Завдяки проведеному опитуванню вдалося сформулювати обґрунтовані функціональні вимоги, що відповідають реальним очікуванням майбутніх користувачів. Збір даних здійснювався за допомогою Google Forms. Створена логічна структура, карта сайту та Customer Journey Map, що допомогла краще зрозуміти досвід користувача, які дії, емоції та труднощі виникають на кожному етапі та завдяки цьому вдалося зрозуміти, як

саме проєктувати вебдодаток. На основі усіх зібраних даних було спроектовано адаптивний та доступний інтерфейс, який поєднує зручну навігацію, сучасний дизайн та адаптивність до різних типів пристроїв. При цьому було враховано принципи UX/UI дизайну. Інтерактивні прототипи розроблялися у Figma.

Реалізація вебдодатку охоплювала як клієнтську, так і серверну частини. Frontend реалізовано за допомогою HTML (структура сторінки), CSS (стилізація сторінок), JavaScript (інтерактивність). Це забезпечило комфортну та інтуїтивно зрозумілу взаємодію з користувачем. Backend створено за допомогою PHP. Він забезпечує взаємодію між клієнтом та сервером, відповідає за обробку запитів, роботу з базою даних MySQL та виконання логіки. Це забезпечуватиме надійне збереження та обробку даних. Розробка здійснювалася у редакторі коду VS Code.

В результаті була створена повноцінна онлайн-платформа для розміщення та пошуку оголошень про прилаштування безпритульних тварин. Було також приділено увагу адаптивності. Інтерфейс коректно масштабується на різних розмірах екранів, зокрема на мобільних пристроях. Під час тестування було підтверджено стабільну та хорошу роботу всіх основних функцій.

Разом із цим було виявлено декілька напрямів для покращення в майбутньому – розширення функціоналу, інтеграція нових сервісів і покращення користувацького досвіду. Це все сприятиме залученню ширшої аудиторії та підвищенню ефективності платформи у досягненні її соціально важливої мети.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було успішно реалізовано вебдодаток, спрямований на вирішення актуальної проблеми – прилаштування безпритульних тварин. Метою дослідження було створення зручної, функціональної та сучасної онлайн-платформи, яка допомагатиме у пошуку нових домівок для тварин, що залишилися без нагляду, особливо в умовах погіршення гуманітарної ситуації через війну в Україні.

У ході виконання роботи було досліджено проблему прилаштування безпритульних тварин. Визначено, що вона є надзвичайно актуальною, особливо в умовах війни, оскільки у цей період зросла потреба у швидкому та зручному способі взаємодії між тими, хто шукає дім для тварини, і тими, хто готовий її прихистити. Було підтверджено доцільність створення спеціального вебзастосунку, щоб бажаючі могли розміщувати та шукати оголошення і допомагати тваринам.

У межах дослідження було проведено порівняльний аналіз наявних платформ: соціальних мереж, дошок оголошень, сайтів притулків, спеціалізованих сервісів. А також проаналізовано популярні існуючі рішення у цій сфері (сайти «Cіріус», «GladPet», «Happy Paw», «Adopt.ua» та «Look4Paws»). Визначено як позитивні рішення, які можна запозичити, так і поширені недоліки, яких варто уникати при створенні власної платформи. На основі цього було прийнято рішення розробити та впровадити нову платформу, яка зможе спростити процес взаємодії між людьми. Отримані результати аналізу стали основою для формулювання основних вимог до вебдодатку, серед яких: можливість швидкого розміщення та пошуку оголошень за фільтрами; перегляд оголошень без обов'язкової реєстрації; особистий кабінет для зареєстрованих користувачів з можливістю керування оголошеннями; зручний та адаптивний інтерфейс; швидке завантаження даних та сумісність із сучасними браузерами.

Обрано технології для розробки: HTML, CSS, JavaScript – для клієнтської частини, PHP – для серверної логіки, MySQL – для зберігання даних.

Особлива увага приділялася UX/UI дизайну, адаптивності, доступності та сучасним підходам до реалізації інтерфейсу. Було опрацьовано також Material UI та Human Interface Guidelines як приклади сучасних систем дизайну.

Досліджено сучасні підходи до організації розробки, зокрема Agile, Scrum і Kanban, і було обрано найбільш доцільний метод для роботи – Kanban, тому що він найзручніший для управління завданнями та часом у межах цього проекту.

У межах UX-дослідження було визначено цільову аудиторію, проведено опитування за допомогою Google Forms, проаналізовано потреби майбутніх користувачів, створено логічну структуру, карту сайту та Customer Journey Map. На основі усіх зібраних даних було визначено головні функціональні вимоги до платформи:

- головна сторінка повинна містити стислу інформацією про призначення сервісу, а також мати зручну та інтуїтивну навігацію;
- каталог оголошень має відображати усі опубліковані оголошення з можливістю фільтрації за різними параметрами;
- форма створення оголошення повинна дати можливість швидко створити оголошення, завантаживши фото та додавши опис тварини, контакти автора, місто та тип оголошення;
- кожне оголошення повинно мати окрему сторінку, що містить повну інформацію про тварину, дату публікації, фотографії та контактні дані для зв'язку з автором. Додатково на цій сторінці мають відображатися ще 4 випадкових оголошення, що можуть зацікавити користувача;
- особистий кабінет, аби користувачі мали змогу керувати власними оголошеннями.

Загалом, розроблено адаптивний та доступний інтерфейс, що враховує основні принципи UX/UI дизайну – зручність, зрозумілу навігацію, візуальну ієрархію та відповідність очікуванням користувачів. Було вирішено, що

інтерфейс має коректно відображатися на різних типах пристроїв, включно зі смартфонами, планшетами та десктопами. Особливу увагу приділено доступності для користувачів із різними потребами. Інтерактивні прототипи всіх основних сторінок були створені у Figma, де також було протестовано зручність користування до початку розробки.

У середовищі Visual Studio Code було реалізовано основний функціонал вебдодатку. Клієнтська частина, тобто інтерфейс, який бачить користувач, була створена за допомогою мови розмітки HTML, що відповідає за побудову структури сторінок. Для стилізації використовувалася мова CSS. Мова JavaScript застосовувалася для додавання інтерактивності. Серверна частина вебдодатку, яка забезпечує обробку даних та взаємодію з базою даних MySQL, була розроблена за допомогою мови програмування PHP.

Після завершення програмування було проведено тестування вебдодатку, включаючи функціональне тестування, перевірку адаптивності та сумісності з різними браузерами. Зокрема, перевірено коректність його відображення на екранах мобільних пристроїв. Виявлено, що платформа працює стабільно, добре відображається на екранах різних розмірів та не має критичних помилок.

Внаслідок практичної реалізації отримано такі результати:

- створено функціональний вебдодаток, де можна публікувати та шукати оголошення, застосовуючи фільтри;
- реалізовано зручний та адаптивний інтерфейс з інтуїтивною навігацією та фільтрацією оголошень;
- протестовано стабільність роботи та коректність роботи усіх функцій.

Серед основних переваг розробленої платформи:

- простота у використанні, інтуїтивно зрозумілий інтерфейс;
- швидкий пошук оголошення завдяки системі фільтрів;
- адаптивний дизайн, завдяки якому вебдодаток коректно відображається на різних пристроях;
- можливість масштабування та впровадження нового функціоналу;
- соціальна значущість проєкту.

Крім того, було запропоновані можливості, які можна покращити у майбутньому:

- додати можливість сортування оголошень за датою публікації;
- реалізувати функцію «Обране» для збереження оголошень;
- впровадити систему повідомлень або чат між користувачами;
- додати можливість авторизації через соціальні мережі;
- створити сторінку з порадами щодо догляду за тваринами;
- додати інтерактивну мапу з локаціями тварин або притулків;
- запровадити модерацію оголошень перед публікацією;
- додати функцію пожертвувань для підтримки тварин і притулків.

Отже, у даній кваліфікаційній роботі було проаналізовано проблему, визначено вимоги, знайдено оптимальні технології та створено корисний інструмент, який може підвищити ефективність комунікації між користувачами, а також спростити процес прилаштування тварин, що потребують нового дому, тимчасової перетримки або загубилися та розшуковують своїх власників.

У результаті було створено повністю функціональний вебдодаток з можливістю додавання та пошуку оголошень, адаптований до різних розмірів екранів і повністю готовий до використання. Так, результати роботи повністю відповідають поставленій меті та завданням. Проєкт має практичну та соціальну цінність, а в майбутньому його можна масштабувати, впроваджувати нові функції та активно використовувати для захисту і допомоги тваринам.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Adopt.ua. Онлайн-база безпритульних тварин з усієї України. URL: <https://www.adopt.ua/> (дата звернення: 14.04.2025)
2. Apple Human Interface Guidelines. URL: <https://developer.apple.com/design/human-interface-guidelines/> (дата звернення: 12.04.2025)
3. GladPet. Онлайн-ресурс з прилаштування (адопції) та допомоги безхатнім тваринам. URL: <https://gladpet.org/> (дата звернення: 14.04.2025)
4. Google Material Design Guidelines. URL: <https://m3.material.io/> (дата звернення: 12.04.2025)
5. Харпурав. Український благодійний фонд допомоги безпритульним тваринам. URL: <https://harpurow.ua> (дата звернення: 14.04.2025)
6. JavaScript language overview. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language_overview (дата звернення: 09.04.2025)
7. Kanban. How the kanban methodology applies to software development. URL: <https://www.atlassian.com/agile/kanban> (дата звернення: 13.04.2025)
8. Look4Paws. Онлайн-платформа для адопції безпритульних тварин. URL: <https://look4paws.club4paws.com/> (дата звернення: 14.04.2025)
9. Nielsen Norman Group. 10 Usability Heuristics for User Interface Design. URL: <https://www.nngroup.com/articles/ten-usability-heuristics/> (дата звернення: 20.05.2025)
10. Nielsen Norman Group. 5 Principles of Visual Design in UX. URL: <https://www.nngroup.com/articles/principles-visual-design/> (дата звернення: 11.04.2025)
11. The Agile Coach. Atlassian's no-nonsense guide to Agile. URL: <https://www.atlassian.com/agile> (дата звернення: 13.04.2025)
12. Безпритульні тварини у прифронтових регіонах: які проблеми та як держава може допомогти. URL: <https://acmc.ua/bezprytulni-tvaryny-u->

pryfrontovyyh-regionah-yaki-problemy-ta-yak-derzhava-mozhe-dopomogty/ (дата звернення: 05.04.2025)

13. Васильєв О. Програмування мовою PHP: навч. посіб. Київ : Видавництво Ліра-К, 2022. 368 с.

14. Довідник по CSS властивостям. URL: <https://css.in.ua/css/properties> (дата звернення: 09.04.2025)

15. Закон України від 15.11.2024 №3447-IV. «Про захист тварин від жорстокого поводження». URL: <https://zakon.rada.gov.ua/laws/show/3447-15#top> (дата звернення: 05.04.2025)

16. Кільченко А. В., Поповський О. І., Тебенко О-р. В., Тебенко О-й. В., Матросова Н. М.; Упорядник: Кільченко А. В. Базові поняття і терміни веб-технологій – К. : ІТЗН НАПН України, 2014. – 49 с.

17. Лебідь О. Ю., Лебідь Я. Ю. Методологія розробки програмного забезпечення Kanban: принципи та переваги. Інноваційні технології, моделі управління кібербезпекою : зб. тез доп. міжнар. наук.-практ. конф., м. Дніпро, 18-20 квіт. 2023 р. Дніпро, 2023. С.25-28.

18. Мосіюк О. О. UX проєктування сайтів як основа підготовки спеціалістів WEB-дизайну. Наукові записки Бердянського державного педагогічного університету. Серія: Педагогічні науки: зб. наук. пр. – Бердянськ : БДПУ, 2017. – 311 с.

19. Сіріус. Найбільший притулок для бездомних тварин в Україні. URL: <https://dogcat.com.ua/> (дата звернення: 14.04.2025)

20. УП. Життя: В Україні нараховується мінімум 140 тисяч безпритульних тварин. URL: <https://life.pravda.com.ua/society/2023/11/1/257378/> (дата звернення: 05.04.2025)

21. Чемерис Г. Ю. UX/UI дизайн : навчальний посібник для здобувачів ступеня вищої освіти бакалавра спеціальності «Дизайн» освітньо професійної програми «Графічний дизайн». Запоріжжя : ЗНУ, 2021. 290 с.

22. Що таке веб-додаток: чим сайт відрізняється від вебдодатку? URL:
<https://outsourcing.team/ua/blog/development/shho-take-veb-dodatok-chim-sajt-vidriznyayetsya-vid-veb-dodatku/> (дата звернення: 07.04.2025)

ДОДАТКИ

ДОДАТОК А

Копія наукової публікації за темою кваліфікаційної роботи

**МЕТОДОЛОГІЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ KANBAN:
ПРИНЦИПИ ТА ПЕРЕВАГИ**

О. Лебідь, Я. Лебідь

Університет митної справи та фінансів

У будь-якого програмного забезпечення є свій життєвий цикл – етапи, через які воно проходить з початку створення до кінця розробки. Найчастіше це підготовка, проєктування, створення та підтримка. Етапи можуть називатися по-різному і поділятися на більш дрібні стадії. Необхідність вибирати методології розробки програмного забезпечення обґрунтована якісним підходом. Оскільки без чіткого плану дій можна почати створення продукту, але так і не закінчити його. Існують різноманітні методології, що покращують роботу. Так, наприклад, каскадна модель, ітеративна та інкрементна моделі, V-модель, RAD та Agile моделі.

Одним із підходів Agile є так звана система постановки завдань та організації робочих процесів Канбан, вона направлена на ефективне досягнення поставленої мети. Головний принцип полягає в обговоренні продуктивності у режимі реального часу, наочності просування роботи. Учасники команди мають змогу бачити стан завдань у будь-який час, оскільки вони візуально представлені на спеціальній дошці.

Інструмент управління Agile-проектами, Канбан-дошка, допомагає візуально подати завдання, визначити обсяг незавершеної роботи, підвищити її ефективність та швидкість. Метод заснований на управлінні виконанням завдань за допомогою карток, що сигналізують про настання та завершення певних етапів. Така система дозволяє делегувати завдання, керувати реалізацією проєкту будь-якої складності, швидко виявляти слабкі місця. Учасникам команди простіше виконувати завдання, оформлені візуально, зростає ефективність роботи при одночасному зниженні навантаження, до того ж вирішується проблема зірваних дедлайнів та завдань, що забули виконати. Також дошка не тільки допомагає виявити недоліки, така візуалізація процесу показує внесок та цінність кожного співробітника, що є джерелом мотивації для людей. Почати побудову своєї системи Канбан можна з простої дошки з трьома основними колонками: Зробити (To do), У роботі (Doing), Готово (Done). Але у випадку з розробкою програмного забезпечення можуть містити такі розділи, як: обговорювані завдання; узгоджені завдання; написання коду; тестування; підтвердження; виконані завдання тощо.

Вперше концепцію Канбан-дошки розробила та впровадила на своїх заводах компанія Toyota. Візуальна система управління завданнями допомогла менеджерам швидко підвищити ефективність організації виробничого процесу та постачання. Його основною метою спочатку була мінімізація відходів без шкоди продуктивності. Головне завдання – створення більшої цінності для клієнта без збільшення витрат. Завдяки своїй ефективності Канбан залишив сферу автомобільної промисловості та був застосований в інших секторах: фінансах, маркетингу, IT-індустрії (розробка програмного забезпечення) та ін [3].

Важливо знати основні принципи, щоб користуватися даною методикою, оскільки однієї дошки недостатньо. Це скоріше побажання для успішної роботи, аніж жорсткі правила. У цій методиці змішалися принципи agile-методологій і lean-мислення. Отож, перший принцип – це візуалізація процесів. Варто забезпечити, щоб усі поставлені завдання було додано у план. Їхній статус варто оновлювати в міру завершення етапів, що вже пройшли. Таким чином можна стежити за прогресом та крок за кроком просуватися вперед і бачити завдання, для вирішення яких необхідно більше часу та допомоги. Друге – згрупування завдань. Найпростіше – розділити завдання на три колонки: «Треба виконати», «Виконується», «Виконано». Такий підхід передбачає візуалізацію робочого процесу та переміщення отриманого завдання з одного етапу на інший. По-третє, потрібно звертати особливу увагу до незавершених завдань. Якщо завдання затримуються на якомусь з етапів, варто розібратися в причинах, розподілити ресурси у разі потреби або надати потрібну підтримку для завершення роботи. Ще один важливий принцип – постійне вдосконалення та покращення [2]. Оскільки контроль за дотриманням термінів виконання завдань та їх переміщенням з одного рівня на інший у цій системі допомагає бачити слабкі місця в роботі. Таким чином, можна чітко визначити, де слід приділити більше часу роботі, а в яких ситуаціях потрібно змінити навантаження. Також дуже важливо підтримувати лідерство на всіх рівнях, так як лідерські дії на всіх рівнях – від окремих працівників і аж до старшого менеджменту – дуже хороша ознака [1].

Порівняння Kanban і Scrum. Існує ще така система управління проєктами, яка на перший погляд схожа з Канбаном, що носить назву Скрам (англ. scrum), вона теж відображає процес вирішення завдань і допомагає досягати поставлених цілей. Однак між цими двома методами є суттєві відмінності. Варто виділити головні з них:

Продовження додатка А

1. У Скрамі робота ділиться на фази, які називають спринтами. Кожен із них вирішує певне завдання — частину проекту. Після закінчення спринту є якийсь конкретний результат, який можна оцінити чи презентувати замовнику. У Канбані робота над завданням розділена на кілька етапів і вона вважається вирішеною тільки після того, як пройшла їх усі.

2. У Канбані можна розділити процес на будь-які відповідні етапи. У Скрамі є конкретна структура, що дозволяє сфокусуватись на результатах.

3. У Скрамі не можна вносити зміни в процесі роботи, щоб не переривати спринт. У Канбані навпаки – можна змінювати перебіг подій, якщо це знадобиться.

4. У Канбані немає вимог щодо проведення щоденних зустрічей для оцінки результатів виконаної роботи. У Скрамі це основа виконання проекту.

5. У системі Скрам є чітко визначені ролі: власник продукту або менеджер продукту, скрам-майстер та команда. У Канбані такого поділу немає, роботою керують усі учасники процесу.

Переваги та недоліки. У методології Канбан немає жорстких правил чи обмежень. Тому її використовують у найрізноманітніших сферах. Тож можна виділити основні переваги, якими вона наділена:

1. Наочність просування роботи та прозорість робочого процесу. Одна з головних переваг даної методології управління процесами. Легше виявляти та усувати проблеми, коли є доступ до поставлених завдань та результативності просування у всіх членів команди. Так кожен може швидко отримати інформацію про стан проекту.

2. Гнучкість планування. Команда зосереджується лише на певному завданні, незважаючи на те, що їх може бути декілька. Також керівник проєкту може змінювати пріоритети виконання роботи, не торкаючись безпосередньо робочого процесу. Після завершення одного завдання команда одразу розпочинає наступне.

3. Швидкість виконання. З'являється безліч способів для менеджерів проєктів уважно стежити за розподілом роботи та робити обґрунтований аналіз.

4. Контролює терміни виконання. Така методологія забезпечує відстежування робочого процесу, дозволяє оптимізувати його тривалість і прогнозувати час, який буде потрібен для вирішення майбутніх задач.

5. Задоволеність клієнтів. Можливість скоротити втрати, концентруючи роботу лише над тими завданнями, які потрібні зараз.

Серед недоліків можна виділити те, що цей метод не підходить для довгострокового планування. Тобто він розрахований для досягнення швидких цілей, де робота вибудовується на вирішенні актуальних завдань, при цьому їх пріоритетність може змінюватися залежно від обставин. Також він не підходить для команд з великою кількістю учасників. Оскільки чим більше людей задіяно у робочому процесі, тим складніше контролювати виконання завдань. Тож краще було б, щоб в одній команді було не більш як десять осіб, а якщо виконання завдання вимагає багато людей, слід розбити їх на невеликі групи і створити для кожної окрему систему Канбан.

Зважаючи на вищезазначені переваги та недоліки, все частіше віддають перевагу саме цій методології, оскільки вона може полегшити роботу в багатьох інших сферах, відмінних від IT-індустрії. Візьмемо, наприклад, промислове виробництво – різні ділянки виробництва мають власні плани, які враховують реальну обстановку. У результаті одні деталі виробляються надміру, інші опиняються в дефіциті – звідси перебої у роботі та невиконання плану. Методологія Канбан пропонує такий тип виробництва, при якому робота у такому разі ведеться точно і у поставлений термін. Замовлення виконуються після надходження, а замість плану – попит споживачів. Кожна наступна ділянка виробництва замовляє попереднім ті деталі, які потрібні у необхідній кількості. Деталі подаються одночасно у потрібні точки, готова продукція відправляється споживачеві. Як наслідок, зменшення ризиків, грамотне регулювання складських запасів.

Продовження додатка А

Дана система допоможе також при збільшенні продажів – оскільки одним із найважливіших напрямків у B2B-бізнесі є робота з клієнтами, вона ідеально підходить для прискорення воронки продажів. Так менеджер вибирає свої угоди та визначає статус клієнта. Щоб пересунути картку на дошці вперед, потрібно підштовхнути клієнта до покупки. Далі треба продати товар клієнту, з яким уже ведеться робота і тільки тоді можна братися до наступної справи. На кожному етапі в свою чергу має бути список умов подальшого просування угоди. Таким чином аналіз роботи етапів даної угоди сприяє оптимізації процесу. А щоб створити досконалу систему роботи з клієнтами, слід відстежувати статуси, які накопичуються на етапах воронки продажів, проводити аналіз ситуації, що складається, вживати відповідних заходів, тим самим змінюючи систему в потрібну сторону.

На думку авторів така система може бути дуже корисною і в особистому плануванні. Вона, безумовно, ефективніша за щоденники та списки, а дошка, завдяки своїй наочності, допомагає упорядковувати домашні та особисті справи. Таким чином, можна спланувати свій місяць чи тиждень, визначивши для себе першочергові задачі та ставити цілі, які крок за кроком будуть досягнуті.

Можна підвести підсумок, що Канбан – досить ефективна методологія, яку можна використовувати у різних сферах. Проте залежно від завдань, які вирішує команда, керівник проєкту може вибрати будь-яку іншу систему організації спільної роботи та управління. Кожна з яких має свої сильні та слабкі сторони. Не можна однозначно сказати, який із методів кращий, тому потрібно вибирати той підхід, який найбільш зручний для команди і найбільше підходить для вирішення певних завдань.

1. What is Kanban? – Basics. Get to know the advantages, disadvantages, and principles of Kanban. – URL: <https://sendpulse.com/support/glossary/kanban>
2. Методологія Kanban: можливості та принципи роботи. – URL: <https://para.school/blog/management/metodologija-kanban-vozmozhnosti-i-printsipu-rabotu>
3. Kanban (development). – URL: [https://en.wikipedia.org/wiki/Kanban_\(development\)](https://en.wikipedia.org/wiki/Kanban_(development))

Джерело: складено автором на основі [17].

Порівняльний аналіз платформ для прилаштування безпритульних тварин

Тип платформи	Соціальні мережі	Дошки оголошень	Сайти притулків	Спеціалізовані сайти
1	2	3	4	5
Приклади платформ	Facebook, Instagram, Telegram	OLX	«Сіріус», «Happy Paw», «Домівка врятованих тварин»	«Look4Paws», «Happy Paw», «GladPet»
Опис	Групи, сторінки та канали, де користувачі розміщують оголошення.	Платформи для розміщення безкоштовних та платних оголошень.	Офіційні сайти організацій, волонтерів, що займаються порядком тварин.	Платформи, які збирають оголошення з різних джерел та забезпечують зручний пошук.
Охоплення аудиторії	Високе охоплення завдяки широкій аудиторії, що активно та швидко поширює інформацію.	Високе охоплення, проте аудиторія здебільшого складається з людей, які активно шукають конкретні пропозиції.	Низьке охоплення, оскільки аудиторія – це користувачі, що цілеспрямовано шукають тварину або хочуть підтримати притулок.	Середнє охоплення, так як аудиторія складається з людей, які активно шукають тварин для адопції або підтримують притулки.
Зручність пошуку	Функціонал пошуку обмежений, оскільки немає спеціалізованих фільтрів для сортування оголошень про тварин, пошук відбувається через хештеги або ключові слова.	Пошук зручний, наявні базові фільтри, це дозволяє здійснювати пошук за категоріями, проте немає розширених можливостей для вибору тварин.	Пошук зазвичай зручний, є структурований каталог та пошук за параметрами, хоча функціонал може варіюватися залежно від сайту.	Розширений пошук, є фільтри, категорії, геолокація, сортування за параметрами.

Продовження додатка Б

1	2	3	4	5
Інтерактивність	Високий рівень інтерактивності, користувачі можуть залишати коментарі, ділитися записами та взаємодіяти з авторами оголошень.	Середній рівень інтерактивності, оскільки взаємодія обмежується поданням оголошень та відповідями на них.	Низька інтерактивність, оскільки взаємодія обмежена переглядом інформації та можливістю зв'язатися з притулком.	Середній рівень інтерактивності включає можливість створювати облікові записи, залишати заявки на адопцію та отримувати рекомендації
Перевірка оголошень	Будь-хто розміщує оголошення.	Оголошення частково перевіряються, є можливість поскаржитись на шахрайство.	Достовірна інформація, оскільки публікуються усі тварини під опікою притулку.	Оголошення перевіряються адміністрацією.
Додаткові можливості	Коментарі, чати, можливість швидкого зв'язку з автором.	Є платні функції просування, рейтинг оголошень.	Детальна інформація про стан тварини, контакти притулку, історії порятунку.	Співпраця з волонтерами, система відгуків, інтеграція з притулками.
Можливість зворотного зв'язку	Коментарі, особисті повідомлення.	Відгуки, чат на платформі.	Контактні дані, заявки через сайт.	Контактна форма, заявки на прилаштування.
Формат подачі інформації	Текст, фото, відео, історії.	Текст, фото, відео.	Текст, фото, інколи відео.	Текст, фото, відео, відгуки, рейтинги.
Основні недоліки	Відсутність зручного пошуку, багато спаму, складність перевірки інформації, публікації про тварин губляться серед великої кількості іншого контенту.	Немає спеціалізації на тваринах, можливе шахрайство, обмежена комунікація між користувачами.	Обмежена кількість тварин (лише ті, що є в конкретному притулку), слабе охоплення аудиторії, оновлення інформації залежить від адміністрації, що може затримувати актуальність даних.	Обмежене охоплення в порівнянні з соцмережами та дошками оголошень, залежність від волонтерської підтримки та фінансування, необхідність реєстрації для розміщення оголошень.



Рисунок В.1 – Головна сторінка «Happy Pet»

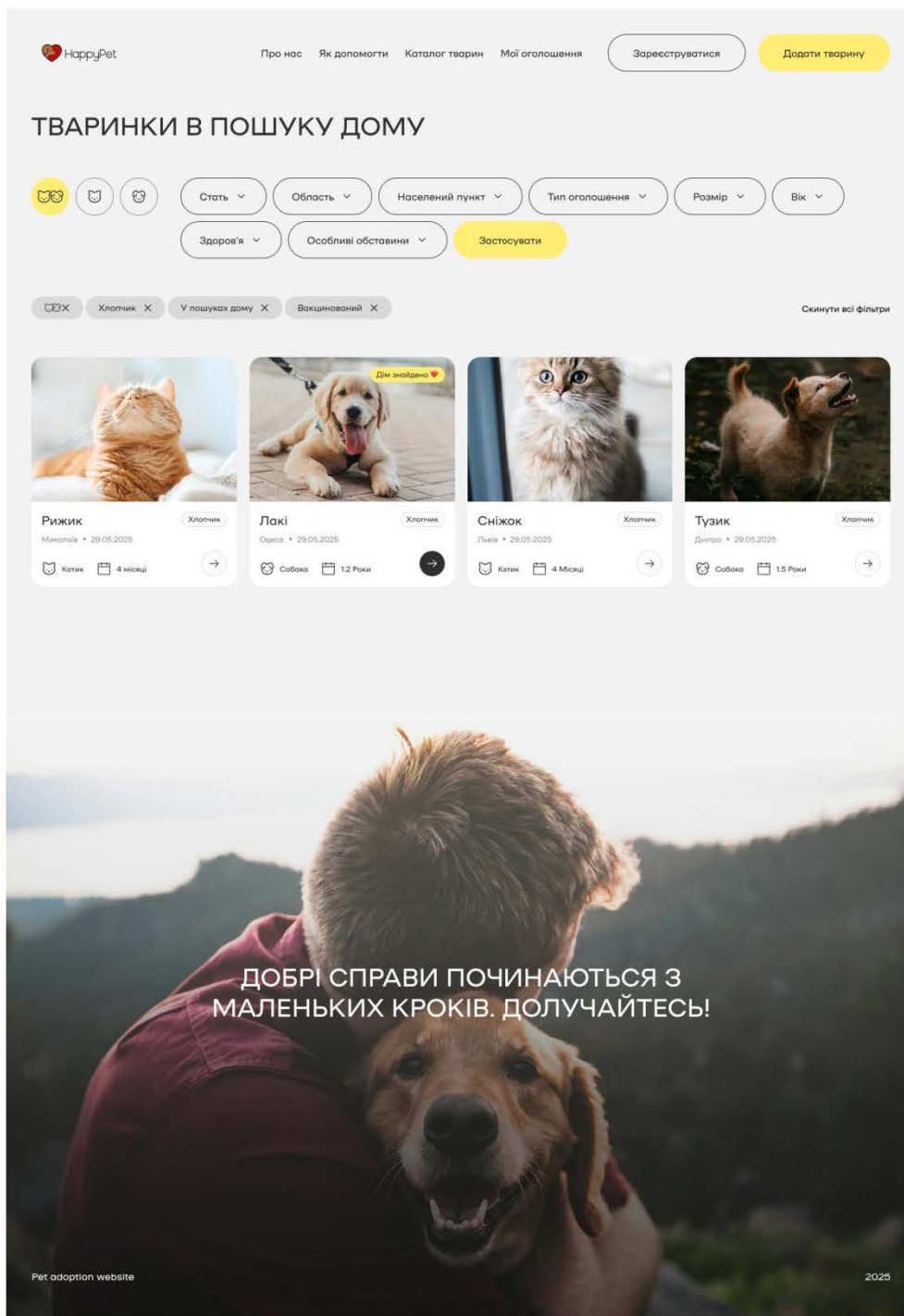


Рисунок В.2 – Сторінка «Каталог»

Продовження додатка В

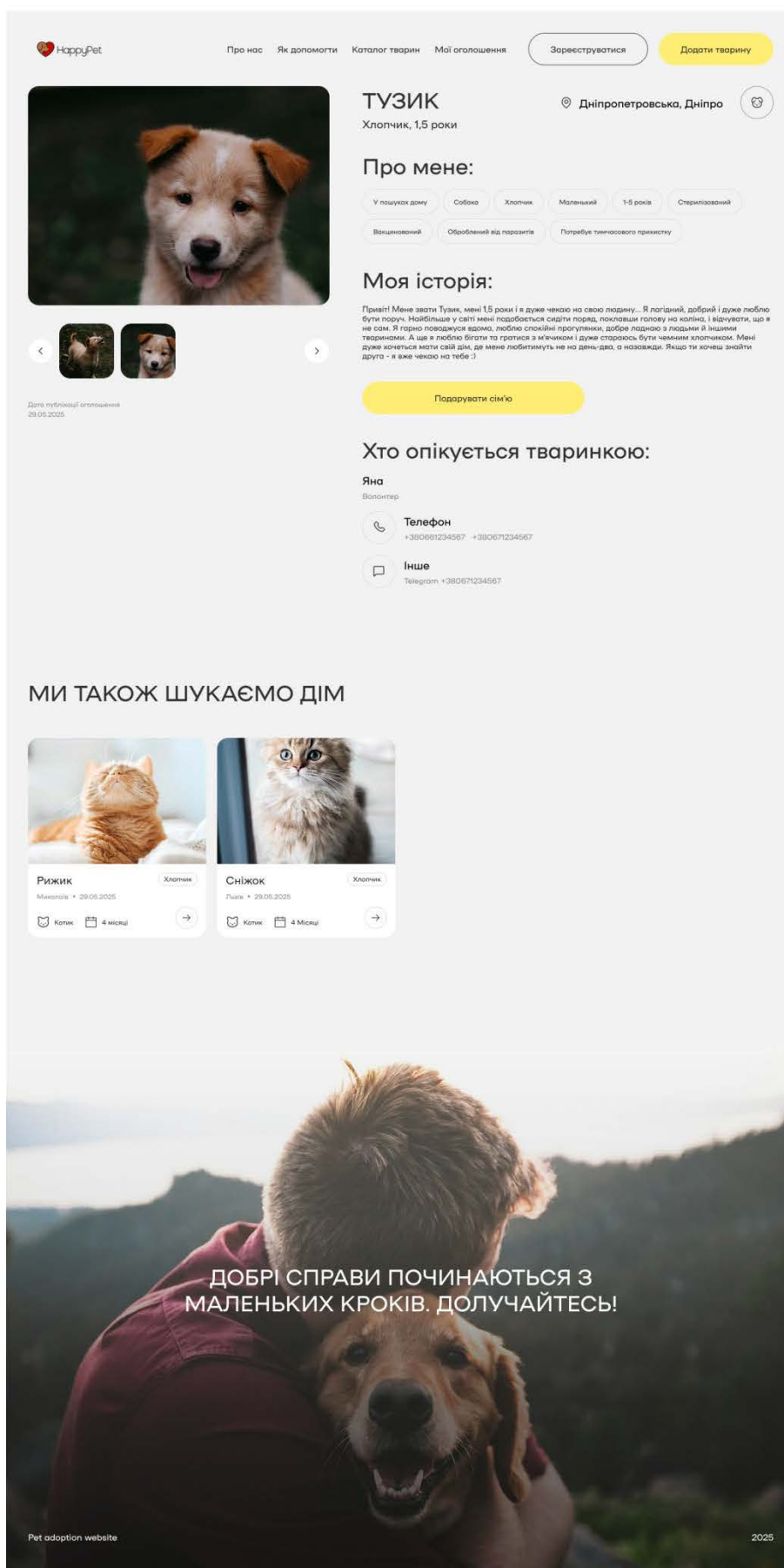


Рисунок В.3 – Сторінка «Оголошення»


Про нас Як допомогти Каталог тварин Мої оголошення Зареєструватися Додати тварину

ДОПОМОЖИ ХВОСТИКУ ЗНАЙТИ ДІМ АБО ПОВЕРНУТИСЯ ДОДОМУ

Заповніть коротку анкету, щоб розмістити оголошення та допомогти безпритульній тварині знайти люблячого господаря або повернути загубленого улюбленця до родини.

Розкажіть більше про характер, звички й особливості пухнастика – це допоможе швидше знайти для нього новий дім і турботливу людину або забезпечити його повернення до рідних. Чим детальніше ви опишете тваринку, тим вищі шанси, що її історія торкнеться серця і змінить чийсь життя.

Тип оголошення

Вид тварини

Стать

Вік

Місця/років

Розмір

Кличка

Оберіть область

Оберіть місто

Хто піклується про тварину

Особливі обставини

Здоров'я

Розкажіть коротко про тваринку: її історію, характер, звички, де ви її знайшли

Номер телефону

Додатковий номер телефону

Як ще з вами можна зв'язатися (Telegram/Viber/WhatsApp)

Контактна особа

Завантажити

Додайте до 10 фото, зроблених на природному світлі, бажано на простому фоні, щоб увага була зосереджена саме на тварині. Знімайте у кількох ракурсах – це допоможе краще показати її характер та особливості.

Опублікувати оголошення

Pet adoption website
2025

Рисунок В.4 – Сторінка «Форма для публікації оголошення»

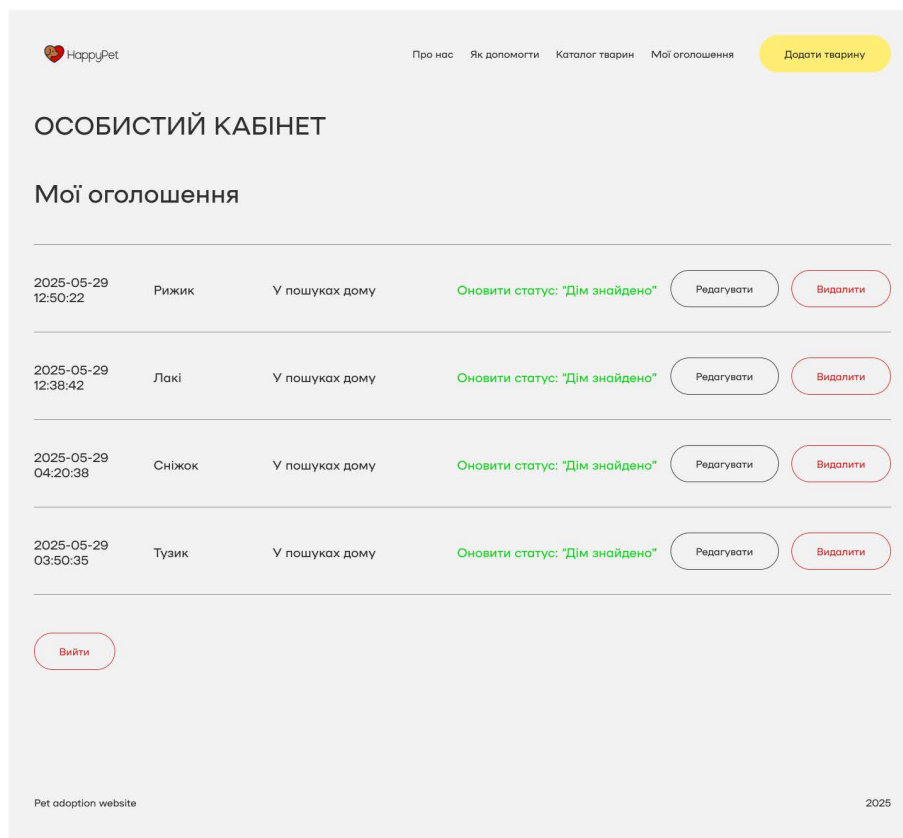


Рисунок В.5 – Сторінка «Особистий кабінет»

Продовження додатка В

The image displays three stacked modal windows, each with a close button (X) in the top right corner.

The first modal window contains the following elements:

- A text input field labeled "Пошта" (Email).
- A text input field labeled "Пароль" (Password).
- A yellow button labeled "Зареєструватися" (Register).
- A text label "Вже маєте акаунт?" (Already have an account?).
- A white button labeled "Увійти" (Login).

The second modal window contains the following elements:

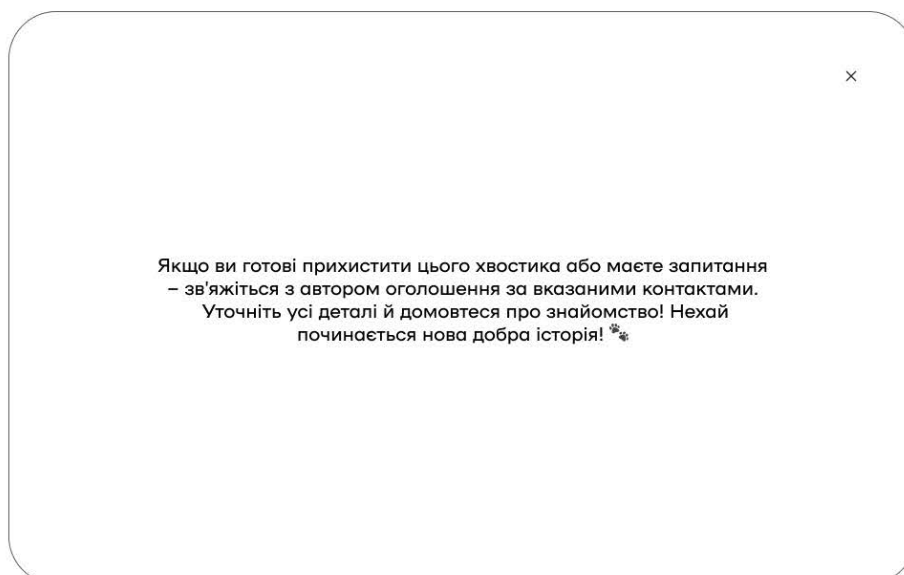
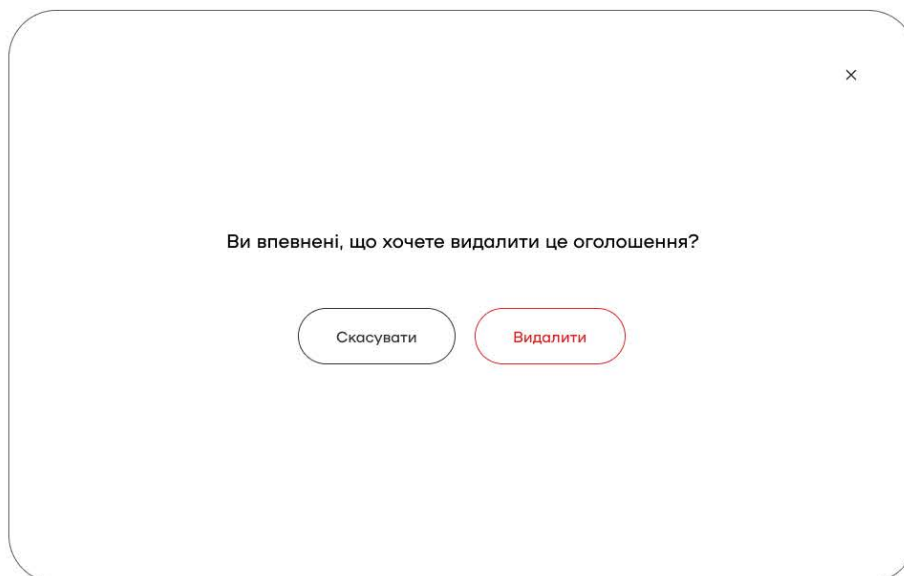
- A text input field labeled "Пошта" (Email).
- A text input field labeled "Пароль" (Password).
- A yellow button labeled "Увійти" (Login).

The third modal window contains the following text:

Дякуємо❤️ Оголошення успішно додано.
Щиро вдячні вам за турботу і небайдужість!
Нехай ця історія обов'язково отримає щасливе продовження! 🍀

Рисунок В.6 – Модальні вікна

Продовження додатка В



Код розробленого вебдодатку

catalog.php

<? php

header('Content-Type: application/json');

require '../includes/Base.php';

```
$payload = json_decode(file_get_contents('php://input'), true);
$offset = isset($payload['offset']) ? (int)$payload['offset'] : 0;
$limit = isset($payload['limit']) ? (int)$payload['limit'] : 20;
$filters = $payload['filters'] ?? [];
```

```
$whereClauses = [
    //'a.status = :status'
];
$params = [
    //'status' => 1
];
```

```
if (!empty($filters['animal_type']))
{
    if ($filters['animal_type'] === 'cat-and-dog') {
        $whereClauses['animal_type'] = 'a.animal_type IN (:animal_dog, :animal_cat)';
        $params['animal_dog'] = 'Собака';
        $params['animal_cat'] = 'Котик';
    } else
    {
        $animalType = '';

        if ($filters['animal_type'] === 'cat') {
            $animalType = 'Котик';
        }
        elseif($filters['animal_type'] === 'dog') {
            $animalType = 'Собака';
        }

        $whereClauses['animal_type'] = 'a.animal_type = :animal_type';
        $params['animal_type'] = $animalType;
    }
}
```

```
if (!empty($filters['ad_type']))
{
    $placeholders = [];
    foreach ($filters['ad_type'] as $i => $val) {
        $key = "type_post_{$i}";
        $placeholders[] = " :$key";
        $params[$key] = $val;
    }
    $whereClauses['ad_type'] = 'a.type_post IN ('.implode(', ', $placeholders). ')';
}
```

```
if (!empty($filters['gender']))
{
    $placeholders = [];
    foreach ($filters['gender'] as $i => $val) {
        $key = "gender_{$i}";
        $placeholders[] = " :$key";
        $params[$key] = $val;
    }
}
```

```

    $whereClauses['gender'] = 'a.gender IN ('.implode(' ', $placeholders). ')';
}

if (!empty($filters['region_id']))
{
    $whereClauses['region'] = 'a.region_id = :region_id';
    $params['region_id'] = (int)$filters['region_id'];
}

if (!empty($filters['city_id']))
{
    $whereClauses['city'] = 'a.city_id = :city_id';
    $params['city_id'] = (int)$filters['city_id'];
}

if (!empty($filters['size']))
{
    $placeholders = [];
    foreach ($filters['size'] as $i => $val) {
        $key = "size_{$i}";
        $placeholders[] = ":{key}";
        $params[$key] = $val;
    }
    $whereClauses['size'] = 'a.size IN ('.implode(' ', $placeholders). ')';
}

if (!empty($filters['age']))
{
    $ageConds = [];
    foreach ($filters['age'] as $range) {
        switch ($range) {
            case 'До 1 року':
                $ageConds[] = "(
                    (a.age_unit = 'Місяці' AND(a.age + 0) <= 12)
                    OR
                    (a.age_unit = 'Роки' AND(a.age + 0) < 1)
                )";
                break;

            case '1-5 років':
                $ageConds[] = "(
                    (a.age_unit = 'Місяці' AND(a.age + 0) BETWEEN 12 AND 60)
                    OR
                    (a.age_unit = 'Роки' AND(a.age + 0) BETWEEN 1 AND 5)
                )";
                break;

            case '5 і більше років':
                $ageConds[] = "(
                    (a.age_unit = 'Місяці' AND(a.age + 0) >= 60)
                    OR
                    (a.age_unit = 'Роки' AND(a.age + 0) >= 5)
                )";
                break;
        }
    }
    if ($ageConds) {
        $whereClauses['age'] = '(' . implode(' OR ', $ageConds) . ')';
    }
}

$healthMap = [
    'Стерилізований' => 'Sterilized',
    'Вакцинований' => 'Vaccinated',
    'Оброблений від паразитів' => 'Treated_for_parasites',

```

```

];
if (!empty($filters['health']))
{
    $healthConds = [];
    foreach ($filters['health'] as $label) {
        if (isset($healthMap[$label]))
        {
            $col = $healthMap[$label];
            $healthConds[] = "a.$col = 'on'";
        }
    }
    if ($healthConds) {
        $whereClauses['health'] = '(' . implode(' OR ', $healthConds) . ')';
    }
}

$specMap = [
    'Евакуйований з прифронтових територій' => 'Evacuated_from_frontline_areas',
    'Потребує тимчасового прихистку' => 'Needs_temporary_shelter',
    'З особливостями (інвалідність, травми тощо)' => 'With_features',
    'Потребує особливого догляду' => 'Requires_special_care',
    'Маленький (до 6 міс.)' => 'Small_up6m',
    'Літнього віку' => 'Elderly',
];
if (!empty($filters['special_conditions']))
{
    $specConds = [];
    foreach ($filters['special_conditions'] as $label) {
        if (isset($specMap[$label]))
        {
            $col = $specMap[$label];
            $specConds[] = "a.$col = 'on'";
        }
    }
    if ($specConds) {
        $whereClauses['special'] = '(' . implode(' OR ', $specConds) . ')';
    }
}

$whereSql = implode(' AND ', $whereClauses);
$sql = "SELECT a.*, r.name_uk AS region_name, c.name_uk AS city_name, (SELECT p.path
FROM advertisement_photo p WHERE p.advertisement_id = a.id LIMIT 1) AS photo FROM
advertisement a LEFT JOIN regions r ON r.id = a.region_id LEFT JOIN cities c ON c.id =
a.city_id WHERE $whereSql ORDER BY a.created_at DESC LIMIT $limit OFFSET $offset";

$ads = Base::query($sql, $params, 'arr');

echo json_encode([
    'ads' => $ads,
    'count' => count($ads),
]);

```

register.php

```
<? php
```

```
header('Content-Type: application/json');
session_start();
```

```
require '../includes/Base.php';
```

```
$data = json_decode(file_get_contents('php://input'), true);
$email = $data['email'] ?? '';
$password = $data['password'] ?? '';
```

```

if (!filter_var($email, FILTER_VALIDATE_EMAIL))
{
    echo json_encode(['success' => false, 'error' => 'Невірний формат email.']);
    exit;
}
if (strlen($password) < 6)
{
    echo json_encode(['success' => false, 'error' => 'Пароль має складатися не менше з 6 символів.']);
    exit;
}

try
{
    $checkExist = Base::get('users', ['email' => $email]);

    if ($checkExist) {
        echo json_encode(['success' => false, 'error' => 'Користувач з таким email вже існує.']);
        exit;
    }

    $hash = password_hash($password, PASSWORD_DEFAULT);

    $userId = Base::add('users', [
        'email' => $email,
        'password_hash' => $hash
    ]);

    $_SESSION['user_id'] = $userId;

    echo json_encode(['success' => true, 'redirect' => 'index.php']);
}
catch (Exception $e) {
    echo json_encode(['success' => false, 'error' => 'Помилка сервера.']);
}

```

deleteAdvertisement.php

```

<? php

session_start();

require 'includes/Base.php';
require 'includes/functions.php';

requireLogin();

if (isset($_GET['id']))
{
    $id = $_GET['id'];

    $checkExistAdvertisement = Base::get('advertisement', [
        'id' => $id,
        'user_id' => $_SESSION['user_id']
    ]);

    if ($checkExistAdvertisement) {
        Base::remove('advertisement', $checkExistAdvertisement['id']);
    }
}

```

```
header('Location: cabinet.php');
exit();
```

updateAdvertisementStatus.php

```
<? php

session_start();

require 'includes/Base.php';
require 'includes/functions.php';

requireLogin();

if (isset($_GET['id']))
{
    $id = $_GET['id'];

    $checkExistAdvertisement = Base::get('advertisement', [
        'id' => $id,
        'user_id' => $_SESSION['user_id']
    ]);

    if ($checkExistAdvertisement) {
        Base::update('advertisement', $checkExistAdvertisement['id'], [
            'status' => 2
        ]);
    }
}

header('Location: cabinet.php');
exit();
```

profile.js

```
const thumbnailSlider = new Swiper('.profile__slider-thumbnail', {
    slidesPerView: 2,
    spaceBetween: 12,
    watchSlidesProgress: true,
    slideToClickedSlide: true,
    breakpoints: {
        480: {
            slidesPerView: 3,
        },
        768: {
            slidesPerView: 4,
        },
        1000: {
            slidesPerView: 3,
        },
        1280: {
            slidesPerView: 4,
        },
    },
});

const mainSlider = new Swiper('.profile__slider', {
    slidesPerView: 1,
    spaceBetween: 20,
```



```

    navigation: {
      nextEl: '.profile__slider-next',
      prevEl: '.profile__slider-prev',
    },
    thumbs:
  {
    swiper: thumbnailSlider,
  },
});

const profileButton = document.querySelector('.profile__button');
const accepted = document.querySelector('.accepted');
const acceptedInner = document.querySelector('.accepted__inner');
const closeButtons = document.querySelectorAll('.accepted__close, .accepted__btn-close');
if (profileButton)
{
  profileButton.addEventListener('click', () => {
    accepted.classList.add('active');
  });
}
closeButtons.forEach(btn => {
  btn.addEventListener('click', () => {
    accepted.classList.remove('active');
  });
});
document.addEventListener('click', (e) => {
  if (
    accepted.classList.contains('active') &&
    !acceptedInner.contains(e.target) &&
    !profileButton.contains(e.target)
  )
  {
    accepted.classList.remove('active');
  }
});

```