

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

**Спеціальність: 121 «Інженерія програмного забезпечення»**

**Освітньо-професійна програма: «Розробка програмного забезпечення»**

**Група: 4РП-08**

# **Дипломний проект**

**здобувачки освіти денної форми навчання**

**РП.08.02.000.ДП**

***БЕЛЯЄВОЇ  
АННИ ВІКТОРІВНИ***

**м. Одеса  
2025 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: **121 «Інженерія програмного забезпечення»**

Освітньо-професійна програма: **«Розробка програмного забезпечення»**

Група: **4РП-08**

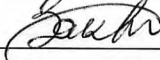
## ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

### Розробка веб-застосунку інтерактивної мапи пошуку безпритульних тварин

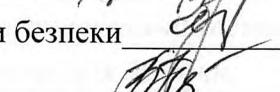
Проектний матеріал складається з пояснювальної записки на 74 сторінках та графічного (презентаційного) матеріалу на 14 аркушах (слайдах)

Дипломник  (Беляєва А.В.)

Керівник  (Закроєв Ю.М.)

#### Консультанти:

з економічного розділу  (Канський М.Ю.)

з розділу охорони праці та техніки безпеки  (Чорновол Н.І.)

з нормоконтролю  (Петрашова В.І.)

старший консультант  (Кривченко Ю.В.)

#### До захисту допущений

Голова циклової комісії  (Кривченко Ю.В.)

Завідувач відділення  (Краснокутська К.Г.)

Захист «26» 06 2025 р. Протокол ЕК № 2

Оцінка ЕК 4/80

Секретар ЕК 

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ  
Спеціальність 121 «Інженерія програмного забезпечення»  
Освітньо-професійна програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:  
Заст. дир. з НВР Беркань І.В.  
« 12 » 08 2025 р.

## ЗАВДАННЯ

### на дипломний проект

Здобувачі освіти Беляєвій Анні Вікторівні  
(прізвище, ім'я, по батькові)

1. Тема проекту Розробка веб-застосунку інтерактивної мапи пошуку безпритульних тварин.

затверджена наказом по коледжу від «14» листопада 2024р. № 246

2. Термін здачі закінченого проекту \_\_\_\_\_

3. Вихідні данні до проекту \_\_\_\_\_

1. Передбачити мінімалістичний UI/ UX;

2. Використовувати технології з сумісними безкоштовними хостинг-провайдерами;

3. Впровадити систему авторизації та реєстрації для безпеки;

4. Застосувати бібліотеки з інтерактивними мапами.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

1. Аналіз предметної області; 2. Технології та засоби розробки;

3. Проектування веб-системи; 4. Розробка монолітного MVC-застосунку;

5. Тестування створеного веб-системи;

6. Економічний розрахунок; 7. Аспекти охорони праці та техніки безпеки.

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Титул; Основні положення; Використані технології; Архітектура системи; Схема бази даних;

Робота з мапою; Робота з API; Робота з механізмами; Реєстрація; Авторизація; Структура

мапи; Додавання маркеру на мапу; Висновки і подальший розвиток.

6. Консультанти по проекту, із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний розділ	Закроєв Ю.М.		
Економічний розділ	Канський М.Ю.		
Розділ охорони праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кривченко Ю.В.		

7. Дата видачі завдання 12.05.2025

Керівник

Закроєв Ю.М.

(підпис)

Завдання прийняв до виконання

Беляєва А.В.

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Формування вступу	15.05.25	виконано
2	Дослідження предметної області	16.05.25	виконано
3	Огляд існуючих рішень	19.05.25	виконано
4	Вибір технічної літератури	20.05.25	виконано
5	Аналіз технологій розробки	22.05.25	виконано
6	Проектування веб-системи	23.05.25	виконано
7	Розробка монолітного MVC веб-застосунку	26.05.25	виконано
8	Тестування створеної веб-системи	24.05.25	виконано
9	Оформлення пояснювальної записки	30.05.25	виконано
10	Підготовка графічних матеріалів	04.06.25	виконано
11	Економічний розрахунок	06.06.25	виконано
12	Опис аспектів охорони праці та техніки безпеки	10.06.25	виконано
13	Підведення висновків	13.06.25	виконано
14	Підготовка доповіді для захисту	16.06.25	виконано

Дипломник

(підпис)

Керівник

(підпис)



# ЗМІСТ

Вступ.....	7
1 Основний розділ.....	9
1.1 Відомості про предметну область .....	9
1.1.1 Проблема безпритульних тварин.....	9
1.1.2 Технології та їх призначення .....	10
1.1.3 Робота з картами за допомогою Leaflet.js.....	13
1.1.4 Огляд сервісів-аналогів .....	15
1.2 Архітектура веб-системи.....	19
1.2.1 Архітектурний шаблон .....	19
1.2.2 Структура бази даних .....	23
1.2.3 Аутентифікація та авторизація .....	26
1.2.4 Робота з картою .....	27
1.2.5 Робота з зображеннями .....	32
1.2.6 Фільтрація даних та обробка даних на мапі.....	34
1.3 Перевірка та мануальне тестування .....	37
1.3.1 Рестрація та сторінка входу .....	37
1.3.2 Тестування роботи карти та відображення інформації на мапі.....	39
1.3.3 Додавання інформації про безпритульну або загублену тварину .....	41
2 Економічний розділ .....	51
2.1 Резюме.....	51
2.2 Визначення трудомісткості розробки ПЗ.....	51
2.3 Розрахунок ціни програмного продукту .....	54
3 Розділ охорони праці та техніки безпеки.....	56
3.1 Основні положення.....	56
3.2 Аналіз умов праці й забезпечення безпеки при виконанні основних видів робіт .....	56
3.3 Гігієнічні вимоги до виробничого середовища .....	56
3.3.1 Вимоги до приміщення експлуатації ПК.....	56
3.3.2 Вимоги до приміщення експлуатації ПК.....	57

					<i>РП 08. 02 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

3.3.3	Виробниче освітлення .....	58
3.3.4	Електробезпека .....	58
3.3.5	Організація робочого місця з ПК .....	58
3.4	Пожежна безпека .....	59
3.5	Висновки.....	60
Висновки.....		61
Перелік використаних інформаційних джерел .....		62
Додаток А. Програмний код MVC-архітектури для пошуку тварин .....		63
Додаток Б. Слайди мультимедійної презентації .....		67

					<i>РП 08. 02 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

## ВСТУП

Проблема безпритульних тварин набуває все більшої актуальності як у країнах Європи, так і в Україні. Зростання кількості безпритульних собак і котів є наслідком низки соціально-економічних та правових чинників: відсутності належного контролю за розмноженням тварин, нерегульованості процесів реєстрації домашніх улюбленців, недостатнього рівня обізнаності населення та обмежених ресурсів для роботи притулків. Безпритульні тварини стають не лише гуманітарною проблемою, а й створюють загрози для громадського здоров'я, безпеки дорожнього руху та екологічного балансу міських екосистем.

У країнах Європейського Союзу питання контролю за чисельністю безпритульних тварин вирішується за допомогою комплексних підходів, що включають обов'язкову стерилізацію, електронну реєстрацію, пропагування відповідального ставлення до тварин через масові інформаційні кампанії, а також активну підтримку програм адопції. В Україні реалізація подібних програм лише набирає обертів, проте для досягнення значних результатів необхідне залучення сучасних цифрових технологій, які можуть оптимізувати облік та моніторинг ситуації на місцях. Одним із перспективних рішень є створення інтерактивних веб-застосунків, які дозволяють оперативно збирати та аналізувати інформацію про безпритульних тварин, підвищуючи ефективність заходів щодо їх захисту та прилаштування.

Інтерактивні мапи на базі сучасних технологій здатні стати важливим інструментом у боротьбі з проблемою безпритульних тварин. Вони забезпечують можливість відображення географічного розташування тварин, фіксацію випадків їх виявлення, стану здоров'я, проведення необхідних ветеринарних заходів та подальшого влаштування у родини. Такі рішення сприяють не тільки кращій координації роботи волонтерів і комунальних служб, а й залученню громадськості до вирішення проблеми. Однією з головних причин зростання популяції безпритульних тварин в містах України є відсутність ефективної системи контролю та обліку домашніх улюбленців» [1].

					<i>РП 08. 02 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

Метою даної роботи є розробка інтерактивної мапи для моніторингу безпритульних тварин, яка надаватиме зручний інструмент для збору, відображення та оновлення даних у режимі реального часу.

Основним завданням є створення веб-застосунку з інтуїтивно зрозумілим інтерфейсом, що дозволить реєструвати нові випадки виявлення тварин, відслідковувати їх стан та аналізувати просторові тенденції. Для реалізації поставленої мети було використано сучасний стек технологій, зокрема мову програмування PHP для обробки серверної логіки, JavaScript для побудови інтерактивної взаємодії на стороні клієнта, а також бібліотеку Leaflet для створення картографічного інтерфейсу, та фреймворк Angular дозволяє створювати динамічні односторінкові додатки з чітким розділенням логіки та представлення [2].

					<i>РП 08. 02 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

# 1 ОСНОВНИЙ РОЗДІЛ

## 1.1 Відомості про предметну область

### 1.1.1 Проблема безпритульних тварин

Проблема безпритульних тварин у сучасному суспільстві є одним із важливих викликів, який охоплює як соціальну, так і екологічну сфери. Відсутність постійного місця проживання у великої кількості собак і котів призводить до низки негативних наслідків: поширення інфекційних захворювань, нападів на людей, загроз для безпеки дорожнього руху, а також до значних гуманітарних страждань самих тварин. За даними різних досліджень, в Україні налічується десятки тисяч безпритульних тварин, і ця цифра постійно зростає, що вказує на необхідність системного підходу до вирішення проблеми.

Причинами загострення ситуації є низка чинників. Перш за все, це низький рівень культури відповідальності власників тварин, відсутність обов'язкової реєстрації та чіпування, обмежені ресурси для роботи притулків, а також недостатня підтримка державних і місцевих програм щодо контролю популяції безпритульних тварин [3]. Значна частина тварин опиняється на вулиці через безвідповідальне ставлення колишніх власників або через відсутність стерилізації. Такі умови сприяють неконтрольованому розмноженню тварин, що призводить до погіршення якості життя у міських та сільських громадах.

Європейські країни демонструють приклади успішної боротьби з проблемою безпритульних тварин через впровадження комплексних програм, які включають обов'язкову стерилізацію, реєстрацію та контроль власників тварин, популяризацію відповідального ставлення до домашніх улюбленців. Такі підходи передбачають не лише гуманні методи контролю чисельності тварин, але й формування позитивної суспільної свідомості щодо їх захисту. В Україні подібні ініціативи наразі розвиваються фрагментарно і часто залежать від активності окремих громадських організацій чи місцевих волонтерських рухів. Відсутність централізованої системи обліку і моніторингу тварин значно ускладнює планування і реалізацію ефективних заходів боротьби з проблемою.

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

Одним із найбільш дієвих рішень у цьому контексті є впровадження цифрових інструментів, які дозволяють збирати, систематизувати та аналізувати дані про безпритульних тварин. Створення інтерактивних мап для моніторингу безпритульних тварин дає можливість у реальному часі отримувати актуальну інформацію про місця знаходження тварин, їх кількість, стан здоров'я, проведені заходи (стерилізація, вакцинація, прилаштування) та інші важливі характеристики. Такі системи сприяють кращій координації роботи волонтерських організацій, притулків, комунальних служб і муніципалітетів, дозволяючи оперативно приймати рішення та ефективно розподіляти ресурси.

Інтерактивні технології також дають змогу залучати ширшу громадськість до вирішення проблеми – кожен охочий може додати дані про знайдену тварину, допомогти з її прилаштуванням або поширити інформацію. Це формує більш активну громадянську позицію та сприяє розвитку суспільної відповідальності. Крім того, наявність структурованої та візуалізованої інформації допомагає проводити аналітичні дослідження, планувати профілактичні заходи та прогнозувати подальший розвиток ситуації.

В загальному принципі, проблема безпритульних тварин потребує не тільки гуманного підходу до її вирішення, але й впровадження інноваційних інформаційних рішень, які б забезпечували ефективний моніторинг, управління даними та залучення широких верств населення до процесу допомоги. Постанова цієї проблеми обумовлює актуальність розробки інтерактивної мапи для моніторингу безпритульних тварин, що й стало предметом дослідження у даній дипломній роботі.

### **1.1.2 Технології та їх призначення**

Для розробки інтерактивної мапи моніторингу безпритульних тварин було обрано набір сучасних технологій, які забезпечують гнучкість, масштабованість, високу продуктивність та зручність у підтримці веб-застосунку. Вибір інструментів обумовлений прагненням створити рішення, що буде працювати ефективно при мінімальних експлуатаційних витратах та залишатиметься доступним для широкого кола користувачів.

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

Основною мовою програмування на стороні сервера є PHP. PHP—це скриптова мова загального призначення, яка широко застосовується у веб-розробці завдяки своїй простоті, гнучкості, високій продуктивності та активній підтримці спільноти розробників.

У даному проєкті було використано версію PHP 8.2, яка відзначається покращенням безпеки, новими можливостями для роботи з типами даних, а також підвищеною швидкістю у порівнянні з попередніми версіями. Завдяки цьому вдалося значно оптимізувати обробку запитів користувачів та знизити навантаження на сервер [4].

Веб-сервером для розгортання застосунку обрано NGINX—потужний, надійний та високопродуктивний сервер, який є одним із найпопулярніших рішень для обробки великої кількості одночасних з'єднань. Завдяки своїй архітектурі з подією на основі обробки запитів, NGINX дозволяє ефективно обслуговувати велику кількість користувачів з мінімальними витратами ресурсів. Це робить його ідеальним вибором для систем, орієнтованих на інтерактивну роботу в реальному часі, як-от мапи моніторингу.

Для зберігання даних використовується реляційна база даних MySQL 8.0, яка є однією з найпоширеніших систем управління базами даних у світі. MySQL забезпечує високу швидкість, підтримку складних транзакцій, потужні засоби оптимізації запитів та широкі можливості масштабування. У нашому проєкті база даних використовується для збереження інформації про безпритульних тварин, їх геолокаційні дані, статус здоров'я, історію переміщень та інші атрибути, необхідні для ефективного моніторингу.

На стороні клієнта для створення інтерактивного інтерфейсу застосовується бібліотека Leaflet. Leaflet є однією з найпопулярніших легковагових бібліотек для побудови мобільних інтерактивних карт. Вона надає прості API для додавання маркерів, створення спеціальних шарів, нанесення полігонів, трасування маршрутів тощо. Завдяки оптимізації під роботу на мобільних пристроях та широкій гнучкості налаштувань, Leaflet ідеально підходить для задачі створення мапи безпритульних тварин з можливістю оперативного оновлення даних.

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

Розробка клієнтської логіки здійснювалася за допомогою нативного JavaScript без використання важких фреймворків, що дозволило забезпечити мінімальний розмір застосунку та високу швидкість завантаження сторінок. JavaScript обробляє динамічні події взаємодії користувача із мапою.

Також надсилає AJAX-запити на сервер для отримання або оновлення інформації про тварин. Завдяки асинхронній роботі з сервером вдалося досягти плавності взаємодії та знизити навантаження на клієнтську частину застосунку.

Стилізація інтерфейсу виконана за допомогою можливостей CSS3, що дозволяє створити сучасний, адаптивний, зручний для сприйняття користувачем вигляд веб-сторінки [5]. Особливу увагу було приділено мобільній адаптивності, щоб користувачі могли працювати із системою як із комп'ютерів, так і зі смартфонів чи планшетів.

Для керування версіями коду та організації командної роботи застосовується система контролю версій Git разом із репозиторієм GitHub. Це дозволяє зберігати історію змін, працювати над різними гілками розробки паралельно та забезпечувати прозорість розробки на всіх етапах.

Вибір саме цих технологій обумовлений їхньою популярністю у сфері веб-розробки, доступністю, активною підтримкою спільноти та гнучкістю в інтеграції між собою. Додатково важливо підкреслити, що застосування відкритих технологій дозволяє мінімізувати витрати на ліцензування та розгортання, що робить проект економічно доцільним навіть за умов обмеженого бюджету.

Однак необхідно враховувати і певні складнощі, притаманні роботі з подібним стеком технологій. Серед основних викликів слід відзначити підтримку актуальності кодової бази на боці клієнта та сервера, своєчасне оновлення версій бібліотек та серверного оточення, а також забезпечення оптимальної роботи системи при збільшенні навантаження. При великих об'ємах даних та збільшенні кількості одночасних користувачів можуть виникати додаткові витрати на масштабування серверної інфраструктури та оптимізацію бази даних, що потребує відповідної технічної підготовки та постійного контролю. Для наочності основні технології, їх призначення, переваги та причини вибору у даному проєкті наведено

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

у таблиці 1.2.

Таблиця 1.2. Вибір технологій для розробки інтерактивної мапи моніторингу

Технологія	Призначення	Переваги	Причина вибору у проєкті
PHP 8.2	Серверна логіка, обробка запитів	Висока продуктивність, безпека, активна підтримка	Розробка надійного і масштабованого бекенду
NGINX	Веб-сервер для обробки запитів і роздачі контенту	Висока швидкість роботи, масштабованість, ефективне кешування	Забезпечення швидкого завантаження сайту при великій кількості користувачів
MySQL 8.0	Система керування реляційною базою даних	Надійність, висока швидкодія, оптимізація складних запитів	Ефективне зберігання та обробка даних про тварин
JavaScript	Розробка клієнтської логіки	Гнучкість, швидкість роботи, широка підтримка браузерів	Динамічна взаємодія з користувачем та інтерактивність карт
Leaflet.js	Побудова інтерактивної карти	Легкість інтеграції, гнучкість налаштувань, оптимізація для мобільних	Створення зручної та швидкої інтерактивної мапи
CSS3	Стилізація інтерфейсу	Адаптивність	Забезпечення зручного інтерфейсу
Git + GitHub	Контроль версій і спільна робота над проєктом	Історія змін, командна робота, зручність у розробці	Управління кодовою базою та спрощення співпраці розробників

Вибраний набір технологій забезпечує всі необхідні умови для побудови ефективного, сучасного та масштабованого рішення для моніторингу безпритульних тварин, а також відкриває можливості для подальшого розширення функціональності та розвитку системи.

### 1.1.3 Робота з картами за допомогою Leaflet.js

Для реалізації інтерактивної мапи у веб-застосунку було обрано бібліотеку Leaflet.js—одну з найпопулярніших JavaScript-бібліотек з відкритим кодом для

створення інтерактивних карт. Leaflet забезпечує легку інтеграцію мап, додавання маркерів, управління шарами та побудову попапів без потреби у складних конфігураціях. Її основною перевагою є компактний розмір, висока продуктивність навіть на мобільних пристроях, а також широка підтримка в екосистемі open-source.

Ініціалізація мапи у проекті відбувається наступним чином:

```
const map = L.map('map').setView([46.4825, 30.7233], 13);  
  
L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {  
  maxZoom: 19  
}).addTo(map);
```

У цьому прикладі карта відкривається у вказаній географічній точці—місто Одеса, Україна—з початковим рівнем масштабування 13. За замовчуванням використовується базовий шар з OpenStreetMap, який надає безкоштовні та актуальні картографічні дані.

Завдяки такому підходу, користувач одразу отримує не лише геолокаційну інформацію, але й візуальну і текстову—що значно підвищує інформативність та ефективність взаємодії. Додатковою перевагою є можливість масштабування карти та фокусування на точках, що дозволяє швидко переміщатися мапою при великій кількості позначок.

У процесі дослідження були протестовані такі функції Leaflet:

- додавання маркерів з кастомним HTML вмістом;
- ручне оновлення шару при нових запитах до бази;
- обробка випадків з порожнім зображенням (вивід заглушки);
- підтримка адаптивності на мобільних пристроях.

Окрему увагу приділено роботі з маркерами. Усі відмітки, які користувачі додають через форму, динамічно зчитуються з бази даних через API /animal/all і виводяться на мапі у вигляді інтерактивних точок. Для кожного маркера формується спливаюче вікно (роруп), яке містить повну інформацію про тварину та її фотографію [6]:

```
L.marker([s.latitude, s.longitude]).addTo(map)  
  .bindPopup(`  
    <b>Тип тварини:</b> ${s.animal_type}<br>  
    <b>Колір:</b> ${s.color}<br>  
    <b>Розмір:</b> ${s.size}<br>
```

					<b>РП 08. 02 001. 00 ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

```
<b>Стать:</b> ${s.gender}<br>
<b>Вік:</b> ${s.age}<br>
<b>Опис:</b> ${s.description}<br>

`);
```

Модульність Leaflet дозволяє також реалізовувати додаткові функції: фільтрацію маркерів за типом тварини, кластерацію для зручного перегляду великої кількості точок, відображення маршрутів та зони пошуку. Завдяки простому API бібліотека може бути легко розширена плагінами.

Leaflet повністю підтримується сучасними браузерами, не потребує складного налаштування серверної частини й чудово працює в умовах обмежених ресурсів, що робить її ідеальним вибором для веб-застосунків соціального спрямування. Завдяки цьому рішення стало можливим створити зручну та ефективну карту для моніторингу безпритульних тварин, яка працює швидко, стабільно та інформативно.

Результати тестування показали високу стабільність роботи бібліотеки навіть на слабких комп'ютерах та телефонах. Leaflet не потребує встановлення важких залежностей або сторонніх картографічних сервісів, тому є оптимальним рішенням для соціально-орієнтованих платформ.

У перспективі система може бути розширена шляхом інтеграції плагінів:

- leaflet.markercluster—для об'єднання маркерів у групи;
- leaflet-routing-machine—для побудови маршрутів;
- leaflet.draw—для позначення зон або ареалів переміщення тварин;
- geolocation API—для автоматичного визначення позиції користувача при додаванні точки.

Завдяки відкритості екосистеми та великій спільноті Leaflet дозволяє будувати масштабовані картографічні рішення, не втрачаючи простоти у реалізації. У контексті даного проєкту це дозволяє створити не просто карту, а інструмент соціального моніторингу, який поєднує дані, взаємодію та зворотній зв'язок із реальними користувачами.

#### 1.1.4 Огляд сервісів-аналогів

На сьогодні існує кілька рішень, спрямованих на облік та моніторинг

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

безпритульних тварин як в Україні, так і за кордоном. Розглянемо основні приклади таких систем, їхні переваги та недоліки.

Одним із прикладів є мобільний застосунок та веб-платформа Adopt-a-Pet.com (США), яка допомагає безпритульним тваринам знаходити нові родини. Платформа дозволяє шукати тварин за різними параметрами, зручно переглядати фото та інформацію про стан здоров'я тварини [7].

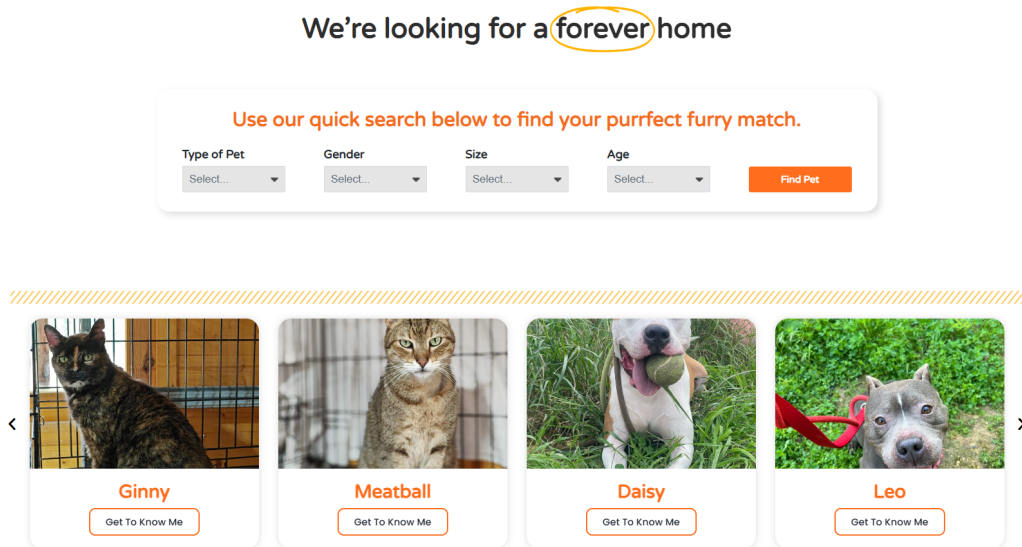


Рисунок 1.1. Інтерфейс застосунку Adopt-a-Pet.com (США)

Переваги:

- Велика база даних тварин з актуальною інформацією;
- Зручний пошук за різними характеристиками;
- Інтеграція з місцевими притулками та волонтерськими організаціями.

Недоліки:

- Орієнтація переважно на регіони США та Канади;
- Відсутність можливості моніторити безпритульних тварин "на вулиці";
- Високі витрати на підтримку серверів і великої кодової бази, що вимагає значних економічних ресурсів.

Іншим рішенням є система PetFinder (США, Канада). Вона надає можливість швидкого пошуку тварин для усиновлення через потужну інтерактивну платформу.

Переваги:

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

- Велика кількість зареєстрованих притулків;
- Гнучка система пошуку та рекомен дацій.

Недоліки:

- Відсутність карти оперативного моніторингу тварин у реальному середовищі;
- Високі операційні витрати на утримання баз даних та інфраструктури, що обмежує масштабування на нові регіони.

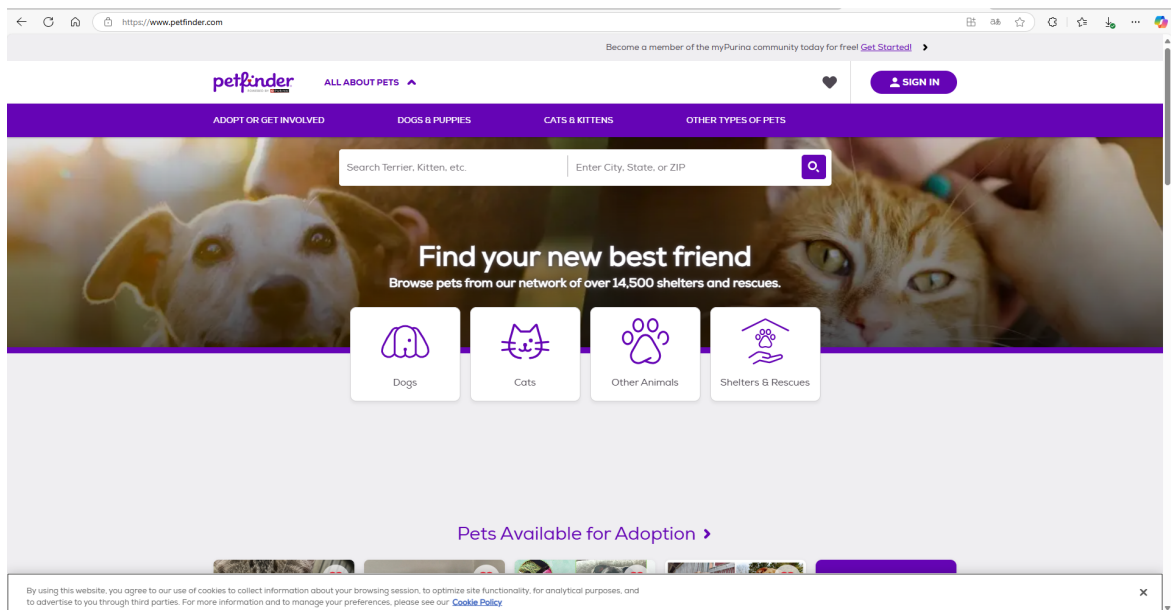


Рисунок 1.2. Інтерфейс застосунку PetFinder (США, Канада)

В Україні прикладом є проект Adopt Don't Stop, який має базу тварин, однак не підтримує інтерактивну карту та не дає можливості відкритого додавання інформації користувачами [8].

Переваги:

- Орієнтація на український ринок;
- Зручна форма заявки на усиновлення.

Недоліки:

- Відсутність автоматичного оновлення інформації;
- Ручне адміністрування бази даних вимагає значних людських ресурсів;
- Відсутність функціоналу оперативного моніторингу вулиць.

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

Додатково слід згадати міжнародні ініціативи на кшталт Stray Animal Database, які інтегровані із державними програмами, проте мають закритий доступ і складну архітектуру.

Переваги:

- Інтеграція з офіційними органами;
- Автоматизація реєстрації стерилізації та вакцинації.

Недоліки:

- Висока вартість розробки та підтримки великих систем;
- Закритість платформи обмежує участь волонтерів;
- Вимагає високого рівня технічної підтримки та постійних витрат на обслуговування.

Таблиця 1.2. Порівняння сервісів-аналогів

Характеристика	Adopt-a-Pet.com	PetFinder	Adopt Don't Stop	Stray Animal Database
Глобальне покриття	США , Канада	США , Канада	Україна	Міжнародне
Інтерактивна карта	Ні	Ні	Ні	Так
Можливість додавання користувачами	Ні	Ні	Частково	Ні
Оновлення даних у реальному часі	Ні	Ні	Ні	Так (частково)
Вартість підтримки	Висока	Висока	Середня	Висока
Відкритість доступу	Так	Так	Так	Ні
Орієнтація на волонтерів	Частково	Частково	Так	Ні

Більшість існуючих рішень мають значні переваги у сфері адопції, однак існують серйозні економічні та організаційні труднощі. Основними проблемами є висока вартість обслуговування серверів і баз даних, складність оновлення великої кодової бази, потреба в кваліфікованому технічному персоналі для підтримки. Також слід зазначити високу кодову базу для підтримки ресурсів, та хмарні сховища.

Це створює передумови для розробки нової інтерактивної системи з оптимізованими витратами, орієнтованої на гнучкість, відкритість для волонтерів та мінімізацію експлуатаційних витрат.

Така система дозволить знизити економічні бар'єри для впровадження та підтримки, забезпечуючи при цьому високу ефективність. Гнучкість і відкритість сприятимуть залученню ширшої спільноти до її розвитку та використання. В результаті це створить стійке рішення, яке мінімізує витрати та максимізує доступність.

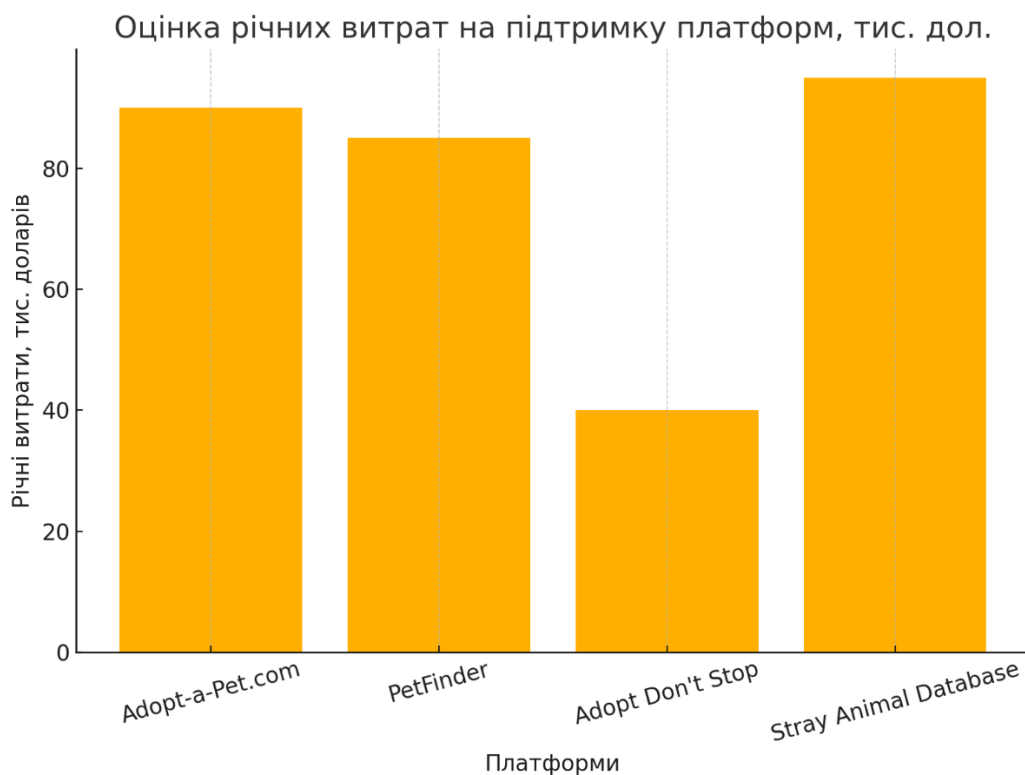


Рисунок 1.3. Орієнтовні річні витрати на підтримку різних платформ

## 1.2 Архітектура веб-системи

### 1.2.1 Архітектурний шаблон

Для забезпечення чіткої структури, підтримуваності та масштабованості системи моніторингу безпритульних тварин було обрано архітектурний шаблон MVC (Model-View-Controller). Використання підходу MVC дозволяє розділити логіку програми на три окремі компоненти: модель (робота з даними), представлення (інтерфейс користувача) та контролер (керування логікою взаємодії між моделями та представленнями).

Структура проекту поділяється на наступні основні директорії:

1. `app/controllers/`—каталог контролерів, що містить класи для обробки запитів користувачів і маршрутизації дій.

2. `app/models/`–каталог моделей, які реалізують взаємодію із базою даних та представляють сутності системи (тварини, користувачі).
3. `app/views/`–каталог представлень, що містить PHP-шаблони для відображення сторінок користувачу (форми авторизації, реєстрації, карта).
4. `app/config/`–налаштування підключення до бази даних та конфігураційні файли системи.
5. `public/`–директорія для відкритого доступу користувачів через браузер (файл `.htaccess` для налаштування роутінгу та індексації, файли стилів і скриптів).
6. Контролери в проєкті відповідають за маршрутизацію запитів та виклик відповідних моделей і представлень:
7. `AnimalController.php`–обробляє логіку, пов'язану з тваринами (додавання нових тварин, виведення списку на карту).
8. `AuthController.php`–забезпечує реєстрацію та авторизацію користувачів.
9. `MapController.php`–відповідає за завантаження інтерактивної карти та даних для її відображення.
10. Моделі є відображенням бізнес-логіки програми:
11. `AnimalModel.php`–взаємодія з базою даних для операцій створення, читання та оновлення інформації про тварин.
12. `UserModel.php`–робота з даними користувачів, авторизація, реєстрація.
13. Представлення розташовані у відповідних підкаталогах:
14. `auth/`–містить файли для сторінок авторизації (`login.php`) та реєстрації (`register.php`).
15. `map/`–містить файл `index.php`, який завантажує та відображає інтерактивну карту із тваринами.

Файл `database.php` в каталозі `config` відповідає за налаштування параметрів підключення до бази даних MySQL, забезпечуючи централізовану точку керування з'єднаннями.

Публічна директорія `public/` містить конфігураційний файл `.htaccess`, який забезпечує коректну маршрутизацію запитів на сервері NGINX/Apache, а також статичні ресурси (стили, скрипти) для завантаження інтерфейсу користувача. Ці

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

статичні ресурси оптимізовані для швидкого завантаження та мінімізації використання трафіку. Вони автоматично кешуються браузером для покращення продуктивності. Конфігурація `.htaccess` також забезпечує безпеку, обмежуючи доступ до службових файлів.

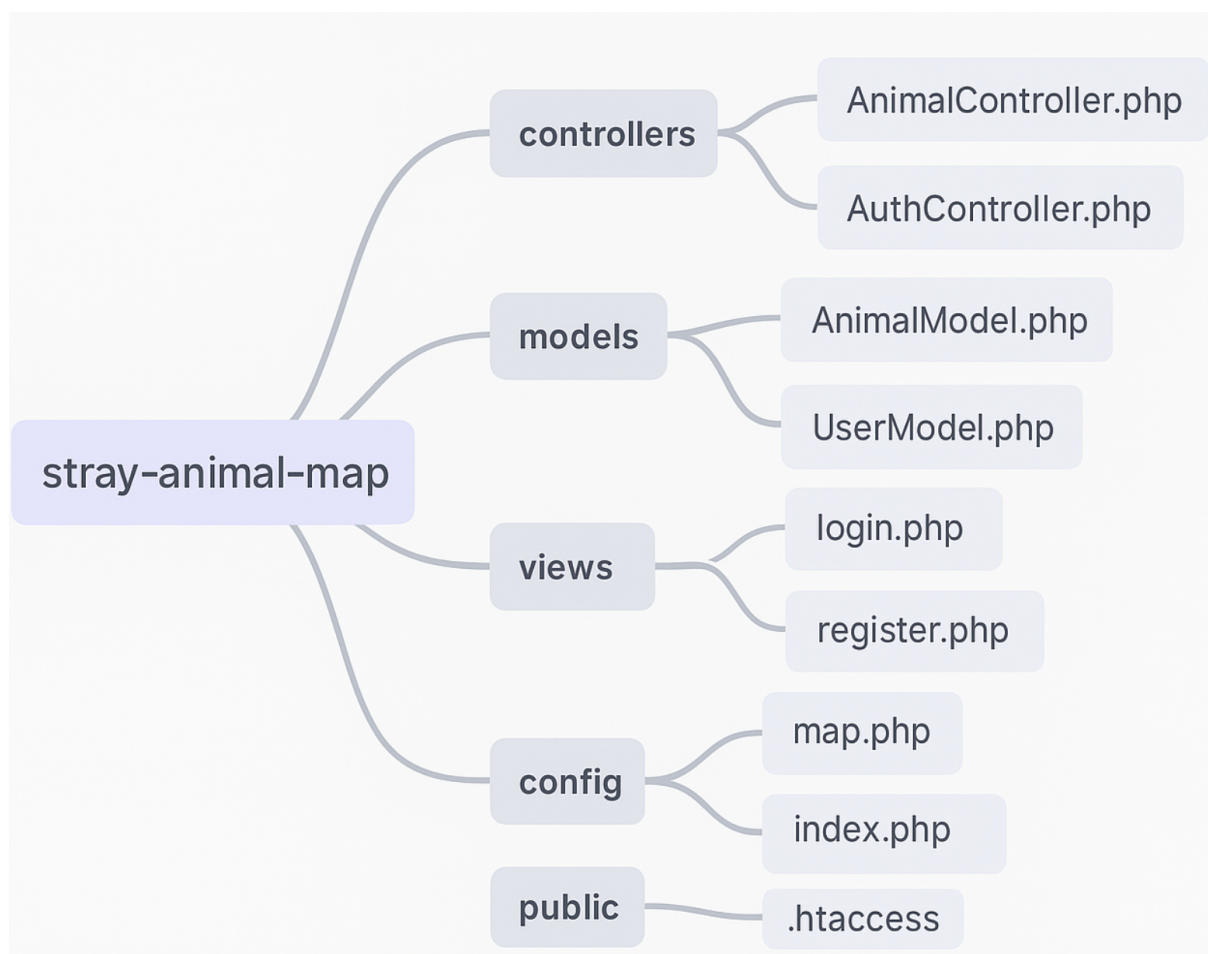


Рисунок 1.4. Взаємозв'язок компонентів системи

Архітектурна схема побудована так, щоб забезпечити:

1. Розділення обов'язків: контролери відповідають за бізнес-логіку, моделі—за дані, а представлення—за відображення.
2. Масштабованість: завдяки MVC легко додавати нові функціональні можливості без переписування існуючого коду.
3. Підтримуваність: модульна структура спрощує тестування, налагодження та подальший розвиток системи.
4. Безпеку: централізоване оброблення запитів дозволяє легко впроваджувати захист від SQL-ін'єкцій, XSS-атак та інших поширених вразливостей.

Взаємодія між компонентами системи відбувається за наступною логікою:

1. Користувач надсилає HTTP-запит через браузер (наприклад, запит на перегляд карти тварин).
2. Веб-сервер обробляє запит і передає його відповідному контролеру (наприклад, MapController).
3. Контролер викликає модель для отримання потрібних даних з бази даних (AnimalModel).
4. Отримані дані передаються до представлення (наприклад, index.php), яке формує HTML-сторінку для користувача.
5. Користувач бачить оновлену карту із даними про безпритульних тварин.

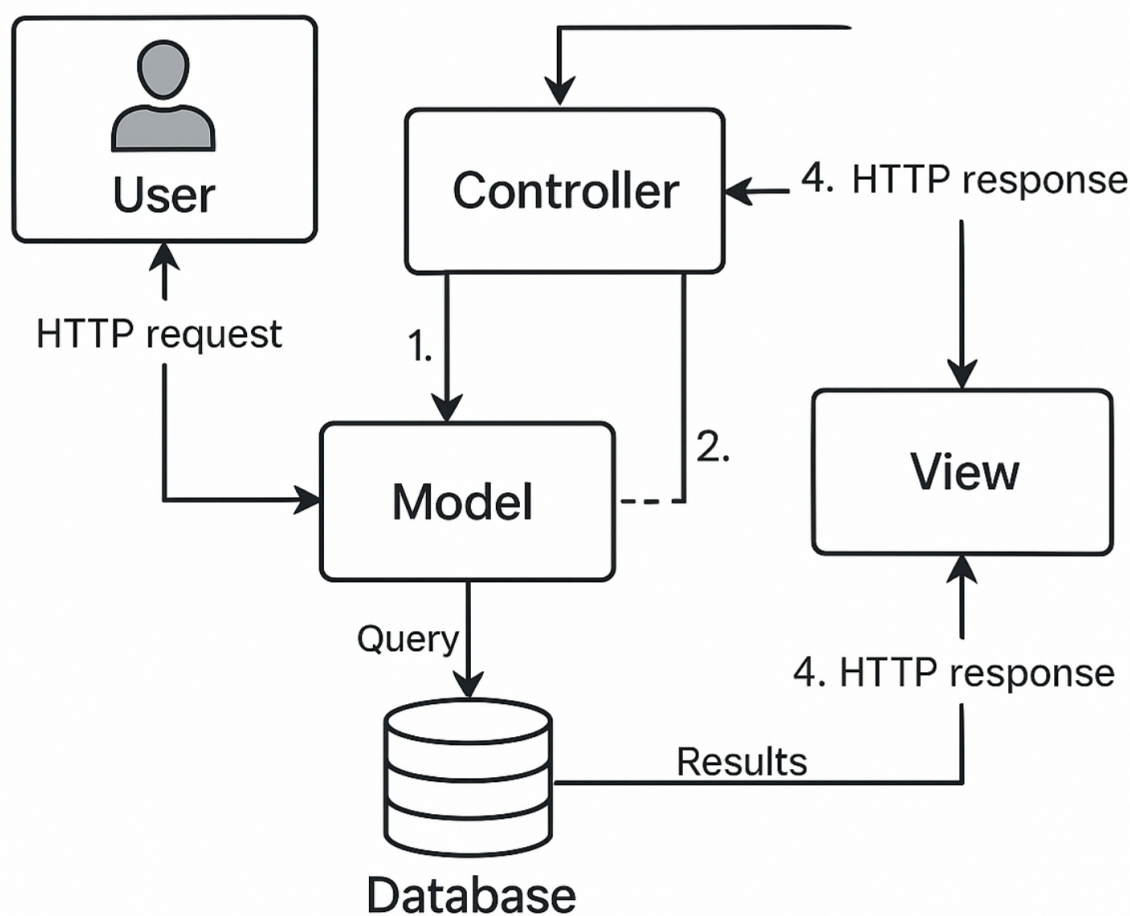


Рисунок 1.5. Архітектура веб-системи

Архітектура проєкту також передбачає можливість легкого масштабування у майбутньому, шляхом додавання нових моделей (наприклад, для ведення історії переміщень тварин), нових контролерів (для роботи з адміністративною панеллю) та розширення функціональності інтерактивної карти.

## 1.2.2 Структура бази даних

Проектування бази даних є одним з ключових етапів створення веб-застосунку, оскільки правильна структура сутностей та зв'язків між ними забезпечує ефективне зберігання, доступ і обробку даних. На цьому етапі здійснюється аналіз функціональних вимог системи, виявляються основні об'єкти, з якими буде працювати застосунок, а також визначаються їхні атрибути, типи, унікальність та взаємозв'язки. Також враховуються вимоги до безпеки, швидкодії та масштабованості системи.

У межах системи моніторингу безпритульних тварин передбачено дві основні сутності: користувачі системи та позначки тварин на мапі. Кожна з цих сутностей реалізується окремою таблицею в реляційній базі даних MySQL.

Першою таблицею є users, що зберігає дані зареєстрованих користувачів.

Таблиця містить поля:

1. id—унікальний ідентифікатор користувача (первинний ключ);
2. email—унікальна електронна адреса користувача;
3. password\_hash—захешований пароль для забезпечення безпеки.

Ця таблиця є базовою для реалізації авторизації користувачів та прив'язки їх дій до конкретних облікових записів. Важливою особливістю є використання хешування паролів, що відповідає сучасним вимогам до захисту персональних даних.

Другою таблицею є animal\_spots, яка зберігає дані про виявлених безпритульних тварин. Структура таблиці включає наступні поля:

1. id—первинний ключ запису;
2. user\_id—зовнішній ключ, що вказує на користувача, який додав позначку;
3. latitude, longitude—координати тварини на мапі;
4. animal\_type, color, size, gender, age—описові атрибути тварини;
5. description—текстовий опис, який дозволяє надати додаткову інформацію (наприклад, стан здоров'я);
6. created\_at—дата і час створення запису.

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

Зв'язок між таблицями реалізований через зовнішній ключ `user_id`, що забезпечує відображення належності кожної позначки до конкретного користувача системи. Такий підхід дозволяє відслідковувати активність, фільтрувати записи по автору, а також забезпечує можливість реалізації функціоналу персональних кабінетів.

Структура обох таблиць відповідає першим трьом нормальним формам:

1. Перша нормальна форма (1NF): усі поля є атомарними, тобто містять лише одне значення;
2. Друга нормальна форма (2NF): всі неключові поля залежать від первинного ключа таблиці;
3. Третя нормальна форма (3NF): усі атрибути не мають транзитивної залежності—тобто, кожне поле залежить тільки від первинного ключа, а не від інших атрибутів.

Така структура дозволяє мінімізувати дублювання даних, забезпечити логічну цілісність і гнучкість у роботі з інформацією.

У таблиці нижче представлена повна структура бази даних:

Таблиця 1.3. Структура таблиці `users`

Назва поля	Тип даних	Унікальне?	Обов'язкове?
<code>id</code>	INT	Так	Так
<code>email</code>	VARCHAR(100)	Так	Так
<code>password_hash</code>	VARCHAR(255)	Ні	Так

Таблиця `'animal_spots'` є основною сутністю у базі даних розробленого веб-застосунку, яка зберігає інформацію про усі відмітки тварин, додані користувачами на мапу. Вона відображає не лише географічне розташування об'єкта, а й містить розширений опис, що дає змогу ідентифікувати тварину та відфільтрувати відмітки за критеріями. Кожен запис у цій таблиці відповідає одному повідомленню користувача про помічену безпритульну або загублену тварину.

Таблиця також дозволяє швидко знаходити потрібні дані за допомогою пошукових фільтрів. Вона є основою для відображення тварин на інтерактивній мапі. Дані з цієї таблиці використовуються для аналізу та покращення роботи

застосунку. Користувачі можуть взаємодіяти з цими даними через зручний інтерфейс.

Поле `id` є первинним ключем і автоматично інкрементується. Воно гарантує унікальність кожного запису. `user\_id` виступає зовнішнім ключем і пов'язує запис із таблицею користувачів.

Поля `latitude` та `longitude` мають формат `DECIMAL(9,6)` і визначають координати точки на мапі з точністю до метра. Ці значення є обов'язковими, оскільки без координат неможливо відобразити маркер на карті.

Атрибути `animal\_type`, `color`, `size`, `gender` та `age` є текстовими полями, які описують зовнішні характеристики тварини. Усі вони є обов'язковими, оскільки без цієї інформації запис не буде корисним для інших користувачів. Поле `description` дає змогу додати додатковий текстовий коментар, який може включати особливості поведінки, обставини виявлення тощо. Це поле не є обов'язковим, але значно підвищує цінність повідомлення.

Поле `created\_at` зберігає дату та час створення запису і задається автоматично на рівні SQL. Це дозволяє фільтрувати маркери за часом та виводити новіші вгорі. У сукупності структура таблиці повністю відповідає потребам застосунку, дозволяючи ефективно зберігати та обробляти інформацію про тварин. Вона є нормально структурованою, відповідає першій, другій і третій нормальній формі, що гарантує відсутність надмірності та логічну цілісність даних.

Таблиця 1.4. Структура таблиці animal\_spots

Назва поля	Тип даних	Унікальне?	Обов'язкове?	Ключ
<b>id</b>	INT	Так	Так	Первинний
<b>user_id</b>	INT	Ні	Ні	Зовнішній
<b>latitude</b>	DECIMAL(9,6)	Ні	Так	—
<b>longitude</b>	DECIMAL(9,6)	Ні	Так	—
<b>animal_type</b>	VARCHAR(50)	Ні	Так	—
<b>color</b>	VARCHAR(50)	Ні	Так	—
<b>size</b>	VARCHAR(50)	Ні	Так	—
<b>gender</b>	VARCHAR(10)	Ні	Так	—
<b>age</b>	VARCHAR(50)	Ні	Так	—
<b>description</b>	TEXT	Ні	Ні	—
<b>created_at</b>	DATETIME	Ні	Ні	—

### 1.2.3 Аутентифікація та авторизація

Для забезпечення контролю доступу та безпечної взаємодії користувачів із системою у веб-застосунку реалізовано повноцінний функціонал аутентифікації та авторизації. Його основна мета—забезпечити, щоб тільки зареєстровані користувачі мали змогу додавати та переглядати відмітки на мапі, завантажувати зображення та отримувати доступ до інших функцій.

Реєстрація нового користувача.

Процес реєстрації реалізовано у методі `register()` контролера `AuthController`. Після надсилання POST-запиту сервер отримує введені дані (`email` та `password`), здійснює хешування паролю за допомогою `password_hash()` і зберігає користувача до бази даних:

```
$password = isset($_POST['password']) ? $_POST['password'] : '';
$hash = password_hash($password, PASSWORD_DEFAULT);
if (UserModel::create($email, $hash)) {
    header('Location: /login');
    exit;
}
```

Хешування пароля за алгоритмом `PASSWORD_DEFAULT` є безпечним та динамічно адаптується до сучасних стандартів (наприклад, `bcrypt`), забезпечуючи захист персональних даних у разі витоку бази.

Вхід до системи (аутентифікація).

Метод `login()` виконує перевірку відповідності `email` і паролю. Спочатку викликається функція `findByEmail()` моделі `UserModel`, що повертає дані користувача з бази, а потім порівнюється хеш паролю за допомогою `password_verify()`:

```
$user = UserModel::findByEmail($email);
if ($user && password_verify($password, $user['password_hash'])) {
    session_start();
    $_SESSION['user_id'] = $user['id'];
    header('Location: /stray-animal-map/public/map');
    exit;
}
```

У разі успішного входу, система ініціалізує сесію та зберігає у неї `user_id`. Завдяки цьому користувач отримує доступ до закритих розділів сайту (наприклад, створення позначок на карті, завантаження фото тощо).

Вихід із системи (`logout`).

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

Вихід реалізовано методом `logout()`, який знищує активну сесію:

```
session_start();
session_destroy();
header('Location: /login');
exit;
```

Це забезпечує повне завершення авторизаційної сесії та захист від несанкціонованого доступу до функціоналу. На всіх захищених маршрутах (наприклад, `/animal/create`, `/animal/uploadPhoto`) реалізовано перевірку на наявність активної сесії користувача:

```
session_start();
if (!isset($_SESSION['user_id'])) {
    echo json_encode(['error' => 'Неавторизований користувач.']);
    exit;
}
```

Зокрема, усі критичні дії доступні лише авторизованим користувачам, що відповідає вимогам захисту персональних даних.

Клієнтські форми.

Сторінки реєстрації та входу реалізовані як окремі представлення (файли `register.php` та `login.php`). Вони містять мінімалістичний адаптивний інтерфейс, поля для введення email та пароля, кнопку входу або реєстрації, а також повідомлення про помилки (наприклад, при неправильному паролі).

```
<form method="post">
  <label>Email:</label>
  <input type="email" name="email" required>
  <label>Пароль:</label>
  <input type="password" name="password" required>
  <input type="submit" value="Увійти">
</form>
```

У разі невдалої спроби вхід/реєстрація відображається повідомлення про помилку, яке передається змінною `$error`. Модель `UserModel.php` містить два основні методи:

- `create()`—створення нового користувача з email та хешем пароля;
- `findByEmail()`—пошук користувача за email (для входу).

```
public static function create($email, $password_hash) {
    global $pdo;
    $stmt = $pdo->prepare("INSERT INTO users (email, password_hash) VALUES (?, ?)");
    return $stmt->execute([$email, $password_hash]);
}
```

#### 1.2.4 Робота з картою

Однією з ключових функціональних складових веб-застосунку є

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

інтерактивна мапа, яка відображає дані про місцезнаходження безпритульних тварин. Для реалізації цієї частини системи було використано JavaScript-бібліотеку Leaflet, яка забезпечує відображення динамічної мапи на основі сервісу OpenStreetMap.

Основна карта ініціалізується за допомогою методу `L.map()` з центруванням на координатах міста Одеса ([46.4825, 30.7233]) та масштабом 13:

```
const map = L.map('map').setView([46.4825, 30.7233], 13);
L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
  maxZoom: 19,
}).addTo(map);
```

Карта автоматично підвантажує дані про тварин, які зберігаються у базі даних. Для цього використовується AJAX-запит до маршруту `/animal/all`, який обробляється методом `all()` контролера `AnimalController.php`. Цей метод повертає JSON-масив об'єктів із бази даних:

```
public function all() {
  ob_clean();
  header('Content-Type: application/json; charset=utf-8');
  echo json_encode(AnimalModel::getAllSpots());
}
```

Усі записи про тварин, отримані з бази даних, додаються на карту у вигляді маркерів. Для кожного маркера створюється підпис (popup) з інформацією про тип тварини, колір, вік, розмір, стать, опис, а також фото:

```
spots.forEach(s => {
  const photoUrl = s.photo
    ? `/${stray-animal-map/public/uploads/${s.photo}`
    : 'https://via.placeholder.com/150?text=Фото+відсутнє';

  L.marker([s.latitude, s.longitude]).addTo(map)
    .bindPopup(
      `Тип тварини: ${s.animal_type}<br>
      <b>Колір: ${s.color}<br>
      <b>Розмір: ${s.size}<br>
      <b>Стать: ${s.gender}<br>
      <b>Вік: ${s.age}<br>
      <b>Опис: ${s.description}<br>
      `
    );
});
```

Користувач може самостійно додавати нові позначки тварин через інтерактивну форму. При натисканні кнопки "+" відкривається модальне вікно, яке містить поля для введення типу тварини, кольору, віку, розміру, статі, опису та завантаження фото.

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

Уся інформація передається у форматі JSON до контролера `AnimalController` за маршрутом `/animal/create`.

Фрагмент коду, що відповідає за відправлення запиту на сервер:

```
fetch('/stray-animal-map/public/animal/create', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify(body)
})
```

На сервері запит обробляється методом `create()`. Контролер перевіряє, чи користувач авторизований, розпаковує JSON, валідує дані та передає їх до моделі `AnimalModel`:

```
public function create() {
    session_start();
    if (!isset($_SESSION['user_id'])) {
        echo json_encode(['error' => 'Неавторизований користувач.']);
        exit;
    }
    $input = json_decode(file_get_contents('php://input'), true);
    $lat = $input['lat'] ?? null;
    ...
    if ($lat && $lng && ...) {
        $animal_id = AnimalModel::createSpot(...);
        echo json_encode(['success' => true, 'animal_id' => $animal_id]);
    } else {
        echo json_encode(['error' => 'Будь ласка, заповніть усі поля.']);
    }
}
```

Метод `createSpot()` у класі `AnimalModel` виконує SQL-запит до бази даних:

```
public static function createSpot($user_id, $lat, $lng, ...) {
    global $pdo;
    $stmt = $pdo->prepare("INSERT INTO animal_spots (...) VALUES (?, ?, ?, ..., NOW())");
    $stmt->execute(...);
    return $pdo->lastInsertId();
}
```

Після успішного додавання інформації про тварину на сервер, система надсилає другий запит, який завантажує фото (через `FormData` та endpoint `/animal/uploadPhoto`). Файл зберігається в директорії `public/uploads/`, а його назва записується у відповідне поле таблиці.

```
if (move_uploaded_file($_FILES['photo']['tmp_name'], $targetFile)) {
    AnimalModel::updatePhoto($animal_id, $photoName);
    echo json_encode(['success' => true]);
}
```

Після цього новий маркер одразу додається на карту у клієнтській частині з уже введеними даними.

Система обробляє всі основні сценарії:

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

- відображення маркерів усіх наявних тварин;
- додавання нових точок через форму;
- валідація введених даних на клієнті та сервері;
- захист дій авторизацією (використання `$_SESSION['user_id']`);
- збереження файлів з перевіркою на помилки.

Тут реалізація інтерактивної мапи дозволяє будь-якому зареєстрованому користувачеві системи оперативно додавати інформацію про безпритульних тварин, що сприяє підвищенню ефективності взаємодії між волонтерами, притулками та звичайними громадянами. Інтерфейс створений максимально інтуїтивно, а обробка даних – безпечною та зручною.

Крім базової функціональності додавання та перегляду маркерів, важливу роль у реалізації карти відіграє забезпечення надійності, безпеки та зручності використання на різних типах пристроїв.

По-перше, кожен запит до серверної частини проходить обов’язкову перевірку авторизації. Наприклад, перед створенням нової позначки виконується перевірка наявності `$_SESSION['user_id']`. Це гарантує, що тільки зареєстровані користувачі мають право взаємодіяти з системою.

Якщо користувач не авторизований, сервер повертає відповідь у форматі JSON з повідомленням про помилку:

```
if (!isset($_SESSION['user_id'])) {
    echo json_encode(['error' => 'Неавторизований користувач.']);
    exit;
}
```

Подібний підхід забезпечує захист від несанкціонованого доступу та підвищує загальний рівень інформаційної безпеки платформи. Крім того, всі запити, які обробляються на стороні сервера, додатково проходять перевірку на наявність усіх обов’язкових полів (`lat`, `lng`, `animal_type`, `color`, тощо). Якщо хоч одного з них бракує – користувач отримує чітке повідомлення про необхідність заповнити всі поля:

```
_encode(['error' => 'Будь ласка, заповніть усі поля.']);
```

У процесі додавання фото реалізована підтримка безпечного збереження зображень. Перед завантаженням перевіряється, чи наявний файл, чи немає

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

помилки завантаження (`$_FILES['photo']['error'] == 0`), а також обробка унікальної назви файлу для уникнення конфліктів:

```
$photoName = uniqid() . '_' . basename($_FILES['photo']['name']);
```

Це дозволяє уникати перезапису файлів, які можуть мати однакову назву, та забезпечує надійне збереження медіа-контенту.

Клієнтський інтерфейс спроектовано з урахуванням адаптивності до мобільних пристроїв. Завдяки використанню Leaflet та стандартних стилів CSS3, карта коректно масштабується та працює на будь-яких пристроях: смартфонах, планшетах та настільних комп'ютерах. Користувач має змогу додати точку незалежно від типу пристрою—це особливо важливо, оскільки виявлення тварини часто відбувається “в полі”.

Весь фронтенд-код побудований за принципом “Single Responsibility”, тобто кожна функція виконує лише одну логічну дію: `addSpot()`—додає нову точку, `openModal()`—відкриває форму, `closeModal()`—закриває її. Це дозволяє легко розширювати функціональність у майбутньому (наприклад, додавання категорій або зміни статусу тварини).

Окремо варто зазначити, що система передбачає асинхронну взаємодію з сервером (AJAX), що дозволяє оновлювати карту в режимі реального часу без перезавантаження сторінки. Це значно покращує UX—користувач одразу бачить свою позначку після додавання:

```
L.marker([center.lat, center.lng]).addTo(map).bindPopup(...).openPopup();
```

Також використання Leaflet дозволяє у перспективі:

- підключати додаткові шари з геоданими (наприклад, дані про зони з підвищеним ризиком);
- реалізувати кластеризацію позначок при великій кількості тварин;
- вбудувати фільтри (тип тварини, район, дата тощо);
- підтримувати оновлення даних через WebSocket у реальному часі.

Побудований, модуль роботи з картою спроектований з урахуванням:

- інтуїтивності для звичайного користувача;
- гнучкості для волонтерів;

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

- масштабованості для розширення в майбутньому;
- захисту та цілісності даних завдяки авторизації, перевіркам і обмеженню доступу.

### 1.2.5 Робота з зображеннями

Окрім базового збереження текстової інформації про тварин, важливою функціональною складовою веб-застосунку є можливість завантаження фотографій, що значно покращує ефективність ідентифікації безпритульних тварин користувачами та волонтерами. Реалізація підтримки зображень вимагає обробки файлів на стороні як клієнта, так і сервера, дотримання вимог безпеки, а також забезпечення оптимального збереження файлів.

Завантаження зображення користувачем.

У клієнтському інтерфейсі, в рамках форми додавання нової позначки, користувач має змогу вибрати зображення тварини через елемент `<input type="file" id="photo">`, яке обов'язково додається разом з іншими атрибутами (тип, колір, розмір, вік, стать, опис). Після відправки основної інформації про тварину (через `/animal/create`), система виконує другий запит для завантаження зображення за допомогою `FormData`:

```
const photo = document.getElementById('photo').files[0];
if (photo) {
  const photoData = new FormData();
  photoData.append('photo', photo);
  photoData.append('animal_id', result.animal_id);

  return fetch('/stray-animal-map/public/animal/uploadPhoto', {
    method: 'POST',
    body: photoData
  });
}
```

Обробка на стороні сервера.

На бекенді, завантаження обробляється методом `uploadPhoto()` класу `AnimalController`. Перед початком обробки відбувається перевірка авторизації користувача:

```
session_start();
if (!isset($_SESSION['user_id'])) {
  echo json_encode(['error' => 'Неавторизований користувач.']);
  exit;
}
```

Після цього перевіряється, чи надане зображення, та чи не виникла помилка

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

завантаження:

```
if ($animal_id && !empty($_FILES['photo']) && $_FILES['photo']['error'] == 0)
```

У разі успішної перевірки файл зберігається у директорії /public/uploads/.

Якщо ця директорія не існує—вона створюється динамічно:

```
$targetDir = __DIR__ . '/../public/uploads/';  
if (!file_exists($targetDir)) {  
    mkdir($targetDir, 0777, true);  
}
```

Назва зображення генерується автоматично за допомогою функції uniqid()

для уникнення конфліктів:

```
$photoName = uniqid() . '_' . basename($_FILES['photo']['name']);
```

Після успішного збереження файлу, в таблиці animal\_spots оновлюється відповідне поле photo методом AnimalModel::updatePhoto():

```
public static function updatePhoto($animal_id, $photo) {  
    global $pdo;  
    $stmt = $pdo->prepare("UPDATE animal_spots SET photo = ? WHERE id = ?");  
    return $stmt->execute([$photo, $animal_id]);  
}
```

Відображення зображень на мапі.

Після отримання масиву всіх позначок, у кожному елементі перевіряється наявність фото. Якщо зображення відсутнє, використовується заглушка-зображення:

```
const photoUrl = s.photo  
    ? `/stray-animal-map/public/uploads/${s.photo}`  
    : 'https://via.placeholder.com/150?text=Фото+відсутнє';
```

В загальному, маркери на мапі завжди мають візуальну складову, що покращує взаємодію користувача з інтерфейсом та підвищує довіру до даних.

Безпека та обмеження.

Під час завантаження зображень враховано кілька аспектів безпеки:

- унікальні назви файлів для запобігання перезапису;
- фізичне обмеження директорії uploads/ на зберігання тільки потрібних типів файлів (налаштування .htaccess);
- відсутність прямого виконання файлів у директорії;
- відсутність виведення шляху до сервера, лише шлях до public URL.

У разі помилки завантаження, сервер повертає відповідне повідомлення у форматі JSON:

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

```
echo json_encode(['error' => 'Не вдалося завантажити фото.']);
```

Перспективи розширення.

Надалі реалізація роботи з зображеннями може бути доповнена:

- обмеженням формату (JPEG/PNG);
- обмеженням розміру (наприклад, не більше 5MB);
- автоматичним створенням мініатюр;
- збереженням EXIF-метаданих (якщо будуть використовуватись камери);
- створенням галереї фотографій для кожної тварини.

### 1.2.6 Фільтрація даних та обробка даних на мапі

Фільтрація даних у веб-застосунках, що працюють з географічною інформацією, є важливою складовою користувацького досвіду. У контексті розробки системи інтерактивної мапи для моніторингу безпритульних тварин, вона дозволяє не лише ефективно відображати маркери на мапі, а й оптимізувати навантаження на систему, уникнути інформаційного перевантаження користувача та забезпечити актуальність і релевантність виведених даних.

Після початкового завантаження усі записи з бази тварин витягуються за допомогою AJAX-запиту до маршруту `/animal/all`. Дані у форматі JSON зберігаються у клієнтській змінній, наприклад `allSpots`, яка є основним джерелом для фільтрації. Наступна логіка фільтрації виконується повністю на клієнті за допомогою JavaScript.

Реалізація фільтрації на клієнтському боці має низку переваг, серед яких найважливішими є швидкість взаємодії та мінімальне навантаження на сервер. Оскільки усі дані вже доступні у вигляді масиву JavaScript-об'єктів, подальша їх обробка зводиться до застосування методів фільтрації, таких як `.filter()`, що дозволяє миттєво отримати підмножину об'єктів, які відповідають обраним критеріям. Це особливо корисно у випадках, коли користувач хоче швидко змінити умови відображення, наприклад, побачити лише тварин жіночої статі, або лише котів, які були додані протягом останнього тижня.

Окрім основних критеріїв типу тварини, кольору, віку, статі система підтримує також фільтрацію за часом публікації. Для цього використовується

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

зіставлення дати створення запису з поточною датою. Різниця обчислюється в днях, і якщо вона не перевищує певного порогу (наприклад, 3 або 7 днів), запис включається у відфільтрований результат. Це дозволяє волонтерам та громадським організаціям оперативно реагувати лише на актуальні повідомлення, не витрачаючи час на перегляд застарілих точок на мапі.

Ще однією особливістю реалізації є підтримка комбінації фільтрів. Це означає, що користувач може одночасно застосовувати кілька умов — наприклад, шукати лише собак білого кольору, середнього розміру, чоловічої статі, доданих протягом останніх трьох днів. Система гнучко комбінує усі ці обмеження, забезпечуючи точний та швидкий результат. Такий підхід не лише підвищує зручність використання, а й сприяє ефективному пошуку потрібної інформації серед великої кількості даних.

Розглянемо конкретну реалізацію. Припустимо, маємо фільтри у вигляді випадючих списків HTML:

```
<select id="filterType">
  <option value="all">Усі</option>
  <option value="dog">Собака</option>
  <option value="cat">Кіт</option>
</select>

<select id="filterGender">
  <option value="all">Усі</option>
  <option value="male">Самець</option>
  <option value="female">Самка</option>
</select>
```

Після вибору фільтрів викликається функція `filterSpots()`, яка аналізує поточні значення елементів `` та відфільтровує маркери:

```
function filterSpots() {
  const type = document.getElementById('filterType').value;
  const gender = document.getElementById('filterGender').value;

  const filtered = allSpots.filter(s => {
    return (type === 'all' || s.animal_type === type) &&
      (gender === 'all' || s.gender === gender);
  });

  renderSpots(filtered);
}
```

Ця функція виконує чисту обробку масиву `allSpots`, після чого результат передається до функції `renderSpots()`, що відповідає за оновлення мапи.

Функція `renderSpots()` очищує попередні маркери на мапі, використовуючи метод `map.eachLayer()`, і додає нові маркери для фільтрованих даних:

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

```

function renderSpots(spots) {
  map.eachLayer(layer => {
    if (layer instanceof L.Marker) {
      map.removeLayer(layer);
    }
  });

  spots.forEach(s => {
    const marker = L.marker([s.latitude, s.longitude]).addTo(map);
    marker.bindPopup(`Тип: ${s.animal_type}Колір: ${s.color}`);
  });
}

```

Цей підхід забезпечує високу продуктивність навіть за великої кількості маркерів, оскільки усі операції виконуються лише в пам'яті браузера без залучення серверу.

Додатково реалізовано фільтрацію за датою створення відмітки. Це дозволяє виводити лише актуальні записи:

```

const daysLimit = 7;
const now = new Date();

const filtered = allSpots.filter(s => {
  const createdAt = new Date(s.created_at);
  const diffDays = (now - createdAt) / (1000 * 60 * 60 * 24);
  return diffDays <= daysLimit;
});

```

Дана логіка використовується у комбінації з іншими фільтрами, створюючи багаторівневу систему фільтрації.

Важливо зазначити, що для комфортного користування системою фільтри реалізовано адаптивно. На мобільних пристроях вони згруповані в collapsible-панель, що автоматично відкривається при натисканні і ховається після вибору.

Окрім клієнтської реалізації, передбачено можливість серверної фільтрації для масштабування. У майбутньому можна використовувати параметри GET-запитів до API:

```

/animal/all?type=dog&gender=male&days=7

```

та обробляти їх у контролері:

```

$type = $_GET['type'] ?? '';
$gender = $_GET['gender'] ?? '';
$days = $_GET['days'] ?? 0;

```

Це дозволить зменшити навантаження на клієнт при великій кількості даних.

Будемо вважати, що реалізація фільтрації на мапі в межах дипломного проекту є багаторівневою системою обробки даних, яка працює ефективно, гнучко й масштабовано. Завдяки комбінації клієнтських і серверних технологій

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

забезпечено швидку взаємодію, наочне візуальне представлення й високу зручність для кінцевого користувача.

Система дозволяє легко адаптуватися до змін у вхідних даних без втрати продуктивності. Вона забезпечує стабільну роботу навіть за умов високого навантаження, зберігаючи швидкість відгуку. Оптимізовані алгоритми гарантують точність фільтрації та мінімальні витрати ресурсів.

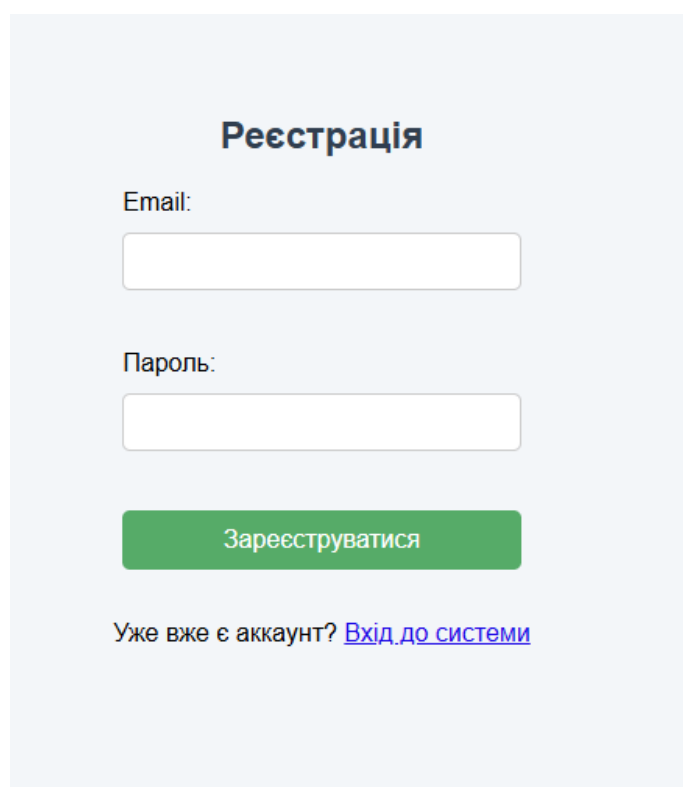
## 1.3 Перевірка та мануальне тестування

### 1.3.1 Реєстрація та сторінка входу

Одним із критичних компонентів веб-застосунку є механізм реєстрації та авторизації користувачів. Для забезпечення цілісності та надійності системи було проведено тестування відповідних елементів інтерфейсу та обробки даних.

Мета тестування—переконатись, що користувачі можуть безпечно створювати акаунти, входити до системи, а також що всі дії коректно відображаються у базі даних.

Форма реєстрації представлена інтуїтивно зрозумілим інтерфейсом, що містить два обов'язкових поля—email та пароль, а також кнопку "Зареєструватися".



The image shows a registration form with the following elements:

- Title: **Реєстрація**
- Label: Email: followed by an empty text input field.
- Label: Пароль: followed by an empty text input field.
- Button: A green button with the text "Зареєструватися".
- Text: "Уже вже є акаунт? [Вхід до системи](#)"

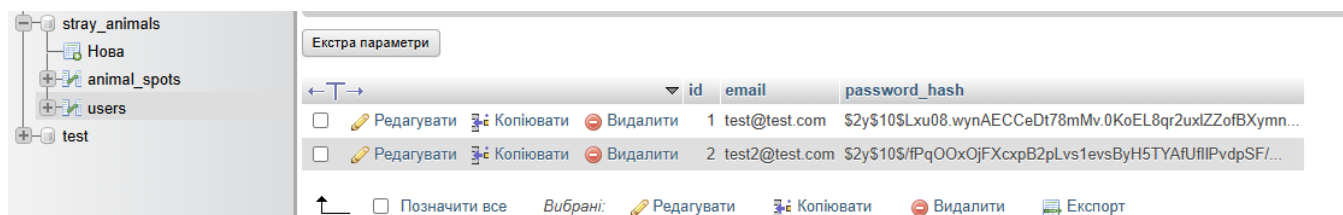
Рисунок 1.6. Форма реєстрації нового користувача

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

Після заповнення форми (наприклад, test2@test.com / 12345678) і натискання кнопки, дані надсилаються методом POST на сервер, де відбувається їхня обробка у методі register() контролера AuthController. Пароль одразу хешується функцією password\_hash() і зберігається у базі.

У процесі тестування підтверджено, що:

- при успішній реєстрації користувача перенаправляє на сторінку входу;
- в базі даних users створюється новий запис з email та хешованим паролем;
- не допускається реєстрація з уже наявною email-адресою.

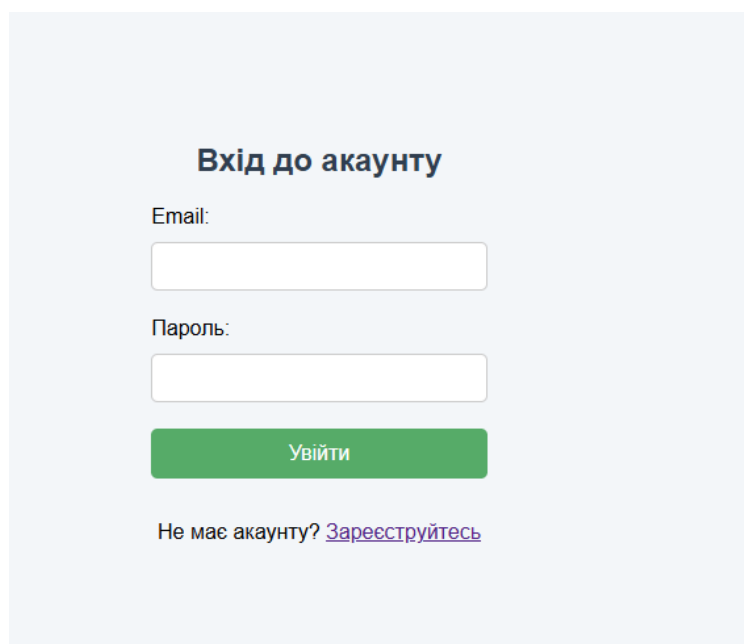


	id	email	password_hash
<input type="checkbox"/>	1	test@test.com	\$2y\$10\$Lxu08.wynAECeDt78mMv.0KoEL8qr2uxlZZofBXymn...
<input type="checkbox"/>	2	test2@test.com	\$2y\$10\$/PqOOxOjFXcxpB2pLvs1evsByH5TYAfUfllPvdpSF/...

Рисунок 1.7. Вміст таблиці users після реєстрації

На скріншоті видно, що поля email зберігаються у відкритому вигляді, тоді як password\_hash—це криптографічний хеш (наприклад, bcrypt), що не дозволяє відновити пароль у зворотному напрямку.

Форма входу також побудована на двох полях—email і пароль, а також містить кнопку "Увійти". У разі помилки (невірний пароль, неіснуючий email) система виводить повідомлення про помилку червоним кольором над формою.



**Вхід до акаунту**

Email:

Пароль:

Не має акаунту? [Зареєструйтесь](#)

Рисунок 1.8. Сторінка входу до системи

При правильному введенні даних, метод login() виконує перевірку email у базі та використання функції password\_verify() для звірки пароля.

Якщо облікові дані правильні—створюється сесія \$\_SESSION['user\_id'], і користувач автоматично перенаправляється на карту (/map).

Під час тестування підтверджено:

- вхід працює лише при наявності точного відповідника email + пароль;
- система успішно ініціалізує сесію;
- у разі виходу з системи (/logout) сесія знищується, і доступ до карти блокується.

Було перевірено, що всі ключові дії (додавання тварини, завантаження фото) заблоковані для неавторизованих користувачів. Це досягається перевірками на рівні контролерів:

```
session_start();
if (!isset($_SESSION['user_id'])) {
    echo json_encode(['error' => 'Неавторизований користувач.']);
    exit;
}
```

### 1.3.2 Тестування роботи карти та відображення інформації на мапі

Одним з найважливіших компонентів у веб-застосунку є інтерактивна мапа, яка виконує роль головного засобу візуалізації інформації про безпритульних тварин. Для перевірки її стабільності, коректного функціонування, зручності використання та точності відображення даних було проведено комплексне тестування.

Мапа реалізована за допомогою бібліотеки Leaflet.js у поєднанні з сервісом OpenStreetMap. При завантаженні сторінки /map, карта ініціалізується в центрі міста Одеса:

```
const map = L.map('map').setView([46.4825, 30.7233], 13);
L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    maxZoom: 19,
}).addTo(map);
```

У ході тестування було підтверджено, що карта коректно підтягує всі наявні відмітки з бази даних. Для цього використовується запит до ендпоінту /animal/all, який повертає JSON-дані всіх збережених тварин у базі. Запит виконується при кожному завантаженні сторінки:

					<b>РП 08. 02 001. 00 ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

```

fetch('/stray-animal-map/public/animal/all')
  .then(res => res.json())
  .then(spots => {
    spots.forEach(s => {
      const photoUrl = s.photo
        ? `/stray-animal-map/public/uploads/${s.photo}`
        : 'https://via.placeholder.com/150?text=Фото+відсутнє';

      L.marker([s.latitude, s.longitude]).addTo(map)
        .bindPopup(`
          <b>Тип тварини:</b> ${s.animal_type}<br>
          <b>Колір:</b> ${s.color}<br>
          <b>Розмір:</b> ${s.size}<br>

          <br>
          <b>Стать:</b> ${s.gender}<br>
          <b>Вік:</b> ${s.age}<br>
          <b>Опис:</b> ${s.description}<br>
          `
        );
    });
  });

```

На практиці це означає, що кожен запис, який користувач додає через форму, після обробки сервером та збереження до таблиці `animal_spots`, автоматично з'являється на мапі при наступному завантаженні сторінки.

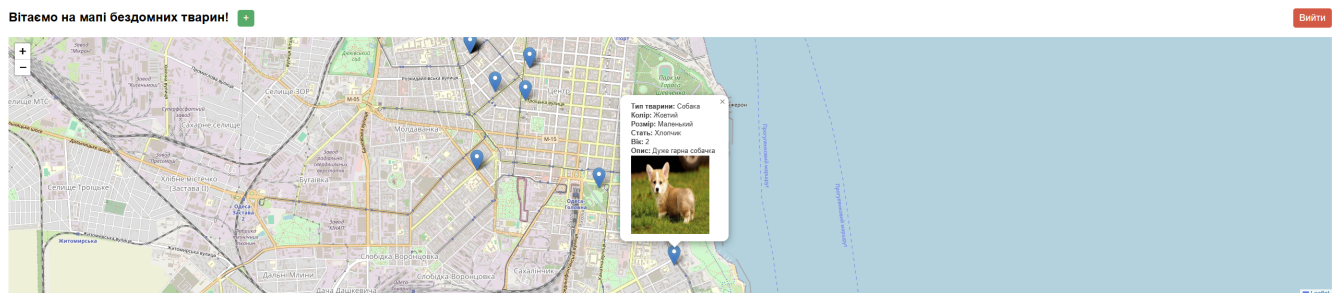


Рисунок 1.9. Відображення тварини на мапі з фотографією

На зображенні видно, як система відображає маркер з інформацією про тварину. При натисканні на нього з'являється рор-уп вікно, де відображається:

- тип тварини (напр. «Собака»);
- колір, розмір, стать;
- приблизний вік;
- короткий опис;
- завантажене фото (або заглушка).

Цей функціонал є візуально зручним і дає змогу швидко оцінити ситуацію на місцевості. Для тестування було виконано кілька ітерацій:

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

- додавання нової позначки—інформація з’являється без помилок;
- маркери розташовуються на коректних координатах (лат/лон);
- зображення завантажуються та прив’язуються до конкретного запису;
- у випадку відсутності зображення—виводиться заглушка.

Також було протестовано масштабування карти, переміщення, відкриття pop-up вікон на різних пристроях. Мапа повністю адаптивна, коректно масштабується під мобільні екрани.

Крім цього, важливо відзначити:

- інтерфейс не потребує перезавантаження сторінки для оновлення маркерів—нова точка додається одразу після успішного запиту;
- збереження маркерів виконується на основі координат центру мапи на момент натискання кнопки.

З технічної точки зору, реалізація Leaflet дає можливість у майбутньому розширити карту фільтрами, категоріями, кластерами або навіть heatmap-візуалізацією.

### 1.3.3 Додавання інформації про безпритульну або загублену тварину

Однією з ключових функцій веб-застосунку є можливість користувача внести на мапу інформацію про помічену безпритульну або загублену тварину. Цей процес є інтерактивним, інтуїтивним і реалізується через модальну форму, яка відкривається натисканням кнопки «+» на головному екрані мапи.

Форма містить набір обов’язкових полів, що дозволяє користувачеві ввести максимально детальну інформацію, включаючи зовнішній вигляд тварини, її поведінку, вік, стать і фото. Такий підхід дозволяє створювати повноцінні позначки, що не лише фіксують місцезнаходження тварини, а й надають волонтерам та іншим користувачам необхідні дані для подальших дій.

Завдяки простій структурі та логічному розташуванню елементів інтерфейсу, користувач може швидко заповнити форму навіть із мобільного пристрою. Усі введені дані обробляються на сервері та миттєво відображаються на мапі у вигляді маркера з описом. Це створює відчуття живої

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

взаємодії із системою та сприяє оперативному реагуванню у випадках, що потребують швидкої допомоги тварині.

Користувач може легко відредагувати або видалити свою позначку, якщо інформація змінилася. Система також дозволяє додавати коментарі до існуючих маркерів, що спрощує комунікацію між користувачами. Це робить процес допомоги тваринам ще більш прозорим і ефективним.

## Додати інформацію про тварину

Тип тварини:

Колір:

Розмір:

Стать:

Вік:

Опис:

Фото:

 No file chosen

**Поставити відмітку на мапі**

Рисунок 1.10. Форма додавання інформації про тварину

У формі користувач повинен вказати:

- Тип тварини (кіт, собака, інше);

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

- Колір;
- Розмір;
- Стать;
- Вік;
- Короткий опис ситуації;
- Фото тварини (необов'язково, але бажано).

Уся логіка обробки форми реалізована у функції JavaScript `addSpot()`, яка:

- отримує координати центру мапи;
- формує JSON-запит із введеними користувачем даними;
- надсилає запит методом POST на ендпоінт `/animal/create`;
- після підтвердження успіху—надсилає зображення через `FormData` на `/animal/uploadPhoto`.

```
fetch('/stray-animal-map/public/animal/create', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    lat: center.lat,
    lng: center.lng,
    animal_type: ...,
    ...
  })
}).then(response => response.json())
  .then(result => {
    if (result.success) {
      // завантаження зображення
    }
  });
```

Сторона сервера відповідає за:

перевірку авторизації користувача (через `$_SESSION['user_id']`);

перевірку всіх обов'язкових полів;

створення нового запису в таблиці `animal_spots`;

повернення `animal_id` у відповіді.

```
if ($lat && $lng && $animal_type && $color && $size && $gender && $age) {
  $animal_id = AnimalModel::createSpot(...);
  echo json_encode(['success' => true, 'animal_id' => $animal_id]);
}
```

Запис включає також дату (`created_at`) і NULL-поле `photo`, яке оновлюється пізніше, якщо додається зображення.

Фото передається окремим запитом через `multipart/form-data`. Перед збереженням файл перевіряється, отримує унікальне ім'я та розміщується в

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

директорії /public/uploads/.

```
$photoName = uniqid() . '_' . basename($_FILES['photo']['name']);  
move_uploaded_file($_FILES['photo']['tmp_name'], $targetFile);  
AnimalModel::updatePhoto($animal_id, $photoName);
```

У кодї, фото зв'язується з відповідною точкою на мапі.

Тестування API додавання тварин проводилося як вручну, так і через інструменти Postman та cURL. Перевірялися такі сценарії:

Таблиця 1.5. Тестування роботи API за допомогою Postman та cURL

Тестовий сценарій	Вхідні дані	Очікуваний результат	Статус
Успішне додавання повної інформації	усі поля заповнені + фото	JSON {success: true}	Пройшло тестування
Відсутнє поле color	відсутнє color	JSON {error: "заповніть усі поля"}	Пройшло тестування
Користувач не авторизований	без сесії	{error: "Неавторизований користувач"}	Пройшло тестування
Невірний тип файлу	фото у форматі .exe	помилка завантаження	Пройшло тестування
Ручне завантаження через cURL	JSON + файл	збереження + запис у БД	Пройшло тестування

Використовувалися типові запити:

```
curl -X POST http://localhost/stray-animal-map/public/animal/create \  
-H "Content-Type: application/json" \  
-d '{"lat":46.48, "lng":30.73, ...}'
```

та

```
curl -F "photo=@dog.jpg" -F "animal_id=15" \  
http://localhost/stray-animal-map/public/animal/uploadPhoto
```

Однією з ключових функцій розробленого веб-застосунку є можливість користувачів самостійно додавати інформацію про виявлених безпритульних або загублених тварин безпосередньо на інтерактивну мапу. Цей механізм реалізовано з урахуванням простоти, швидкості та зручності. В основі лежить форма, яка відкривається у модальному вікні після натискання кнопки «+» на головній сторінці карти. Користувач заповнює поля про тип тварини, її вік, колір, розмір, стать та короткий опис обставин. Обов'язковим є також визначення геолокації, яка автоматично береться з центру мапи або може бути уточнена вручну. Для більшої інформативності дозволяється додати фотографію тварини.

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

Вся логіка взаємодії з сервером реалізована на JavaScript за допомогою Fetch API. Спочатку відбувається запит типу POST до маршруту /animal/create, який обробляє контролер AnimalController на бекенді. Після успішного збереження даних у базі, у відповідь повертається ідентифікатор створеної тварини. Цей ID потім використовується у другому запиті – для завантаження зображення. Фото передається окремим запитом як FormData, де вказується файл та ідентифікатор тварини. На сервері зображення зберігається на файлову систему і до відповідного запису в БД додається назва файлу.

Серверна частина перевіряє авторизацію користувача (через PHP-сесію), коректність даних, обробляє помилки та повертає результат у форматі JSON. Це дозволяє клієнту зрозуміти, чи успішно відбулася операція, та у разі потреби–вивести повідомлення про помилку (наприклад, 'заповніть усі поля', або 'неавторизований користувач'). Таке розмежування логіки підвищує безпеку системи і дозволяє гнучко масштабувати функціональність у майбутньому.

У рамках тестування функціоналу було перевірено як введення валідних даних, так і типові помилки. Наприклад, система коректно обробляє випадки відсутності обов'язкового поля, невірному типу файлу або спроби доступу без авторизації. Також перевірено ручне завантаження даних через Postman або cURL–це підтвердило відповідність API REST-принципам. Усі відповіді сервера були у форматі JSON із чітким вмістом ключів (success/error), що дозволяє легко інтегрувати API в мобільні або сторонні додатки у майбутньому.

Окремо варто підкреслити інтуїтивність інтерфейсу. Модальне вікно з формою має адаптивну верстку, що дозволяє заповнити її навіть на мобільному пристрої. Кнопка виклику форми знаходиться поруч із заголовком, тому користувач не витрачає час на її пошук. Усі поля логічно згруповані, використано селектори замість введення вільного тексту там, де це можливо. Це знижує ймовірність помилок під час заповнення. Фотографії не є обов'язковими, однак при їх наявності–суттєво покращують цінність точки на мапі.

Після завершення процесу додавання тварини–маркер з'являється на мапі миттєво, без перезавантаження сторінки. Для цього використовується внутрішній

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

метод Leaflet, який додає новий елемент у DOM і відкриває pop-up із описом. У карті, користувач не лише бачить результат своїх дій, а й відчуває зворотний зв'язок, що формує позитивний досвід використання системи. Успішність реалізації цієї функції визначається не лише технічною стабільністю, а й емоційною зручністю—що особливо важливо для соціальних сервісів.

Ще одним елементом, що покращує досвід взаємодії, є система сповіщень. Вона реалізована у вигляді тимчасових повідомлень (toast), які з'являються у верхній частині екрана після кожної ключової дії користувача – додавання, редагування або помилки. Повідомлення мають кольорову індикацію (зелений – успіх, червоний – помилка, жовтий – попередження), що забезпечує швидке візуальне сприйняття результату дії.

З метою підвищення ефективності обробки даних на сервері реалізована система кешування останніх запитів користувача. Це дозволяє уникнути повторної обробки і зменшує навантаження на базу даних при частому оновленні інформації, наприклад, на сторінці перегляду останніх показників споживання. Дані оновлюються лише тоді, коли виникає зміна, а не кожного разу при відкритті сторінки, що оптимізує швидкодію.

Інтерфейс мапи також підтримує кластеризацію точок, що особливо актуально у випадках, коли в системі зареєстровано багато об'єктів. Кластери автоматично групуються при зменшенні масштабу і розгортаються при наближенні, дозволяючи легко орієнтуватися у великій кількості даних. Це реалізовано за допомогою плагіна Leaflet.markercluster, що забезпечує високу продуктивність навіть на слабких пристроях.

Для адміністраторів передбачено окремий розділ панелі керування з розширеними можливостями: перегляд усіх доданих записів, фільтрація за статусом, експорт у CSV та ручне редагування. Доступ до цього розділу обмежено перевіркою ролі користувача, яка задається при створенні облікового запису. Завдяки цьому досягається розмежування прав доступу та підвищується безпека системи.

У процесі розробки особливу увагу приділено адаптивності всіх елементів.

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

Кожна форма, таблиця чи графік автоматично підлаштовується під розміри екрана, включаючи вертикальну прокрутку, зміну розташування елементів та масштабування шрифтів. Це дозволяє використовувати додаток з однаковою зручністю як на смартфонах, так і на настільних комп'ютерах, не втрачаючи функціональності або читабельності.

У майбутньому функціонал можна розширити, наприклад, можливістю редагування чи видалення своїх записів, запровадженням категорій (помічено, знайдено, небезпека), або ж підключенням мапових фільтрів для перегляду лише актуальних чи нещодавніх записів. Водночас, збереження простої логіки взаємодії користувача з мапою має залишатися ключовим принципом розвитку цього інструменту.

Усі відповіді були у форматі JSON з чіткою вказівкою на помилки або успіх.

Форма реалізована як модальне вікно, що не перенавантажує сторінку, не потребує оновлення і дозволяє користувачеві одразу бачити результат додавання. Це дозволяє оперативно реагувати на ситуації «в полі».

Після додавання точки система автоматично створює маркер на мапі:

```
L.marker([lat, lng]).addTo(map).bindPopup(...).openPopup();
```

На зображенні користувач бачить результат одразу після натискання кнопки.

Функція додавання тварини реалізована згідно з принципами REST, враховує безпеку, валідацію, обробку помилок і зворотний зв'язок. Реалізація не потребує додаткових залежностей, окрім Leaflet та нативного PHP, що робить її легкою та ефективною.

У межах удосконалення функціональності веб-застосунку було реалізовано можливість видалення раніше доданих маркерів з мапи. Реалізація функції є важливою для забезпечення актуальності даних та зручності користувачів. У випадках, коли тварина вже знайдена, прилаштована або інформація була внесена помилково, користувач може швидко та безперешкодно прибрати відповідну позначку з карти.

З технічної точки зору реалізація цієї можливості передбачала створення нового маршруту на сервері, який приймає запит на видалення конкретного запису з бази даних. Маршрут було реалізовано у контролері MapController.php, а для

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

виконання операції безпосередньо в базі використано метод `deleteById()` у моделі `AnimalModel`. Запит на видалення здійснюється за допомогою HTTP-методу `POST` і передає ідентифікатор позначки у форматі `JSON`.

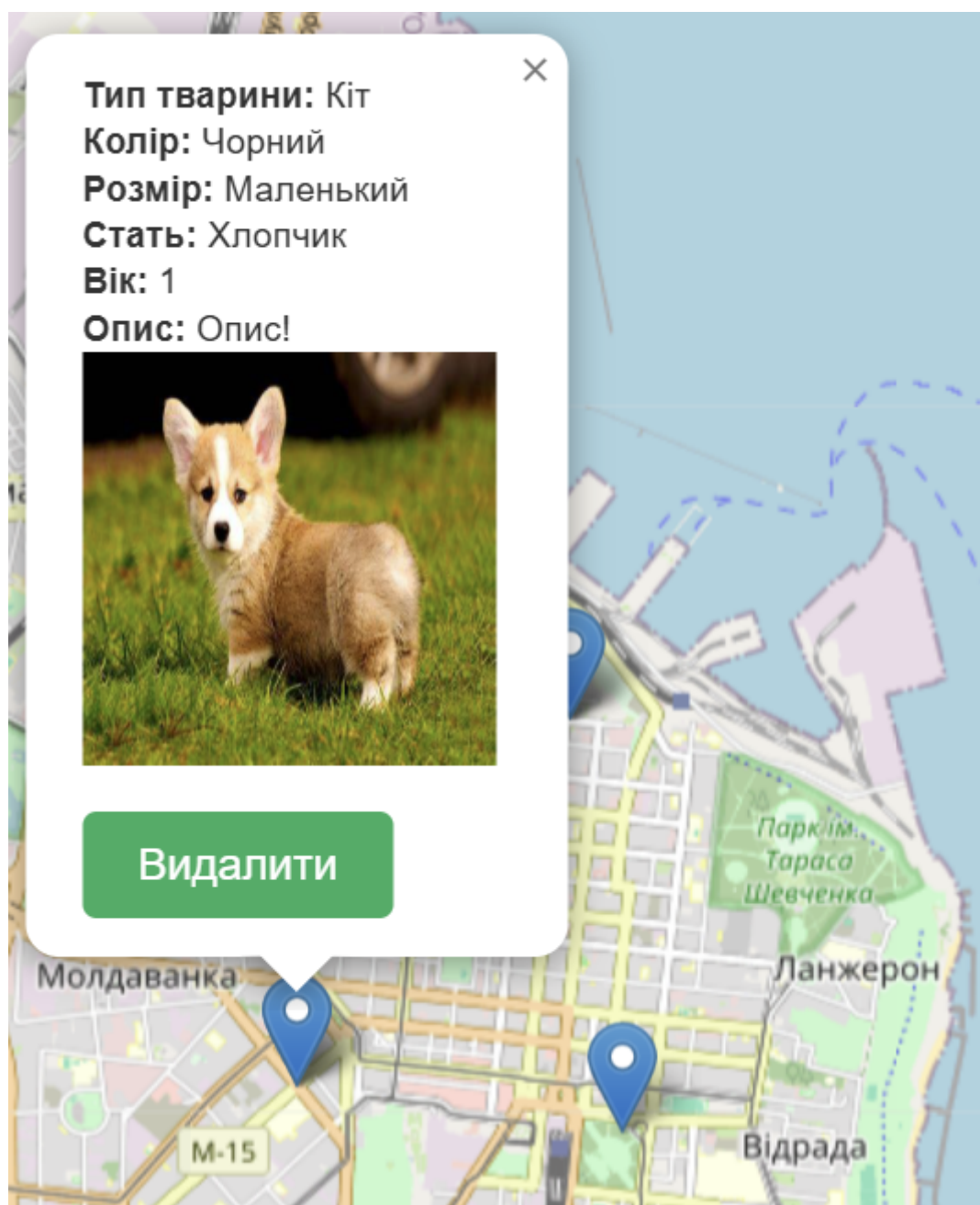


Рисунок 1.11. Приклад розміщення кнопки видалення на сторінки карти

На стороні клієнта було внесено зміни до JavaScript-коду, який відповідає за рендеринг мапи. У кожен маркер, що відображається на мапі, додано кнопку «Видалити». Вона викликає функцію `deleteMarker`, яка відправляє асинхронний `fetch`-запит на сервер, передаючи ID запису, що підлягає видаленню. Після отримання позитивної відповіді з сервера маркер автоматично видаляється з карти

без необхідності перезавантаження сторінки.

Особливу увагу було приділено коректному визначенню саме того маркера, що потрібно видалити. Оскільки кнопка знаходиться всередині роруп-вікна маркера, у JavaScript було реалізовано механізм визначення об'єкта маркера через DOM-навігацію: з кнопки виділяється відповідний роруп, який асоційовано з конкретним marker. Видалення забезпечує точність операції видалення і виключило можливість помилкового видалення інших точок.

На серверному боці маршрут /map/delete попередньо додається до системи маршрутизації у файлі public/index.php. Це дозволяє правильно спрямовувати запити та обробляти їх у відповідному контролері. Усі запити перевіряються на валідність: сервер відкидає запити без ID або з некоректним форматом. У відповідь повертається об'єкт JSON зі статусом операції.

У базі даних видалення здійснюється з таблиці animal\_spots, де зберігаються всі позначки. SQL-запит має вигляд DELETE FROM animal\_spots WHERE id = ?, що забезпечує одночасно простоту та ефективність. Для безпечного виконання запиту використано підготовлені інструкції PDO з прив'язкою параметрів, що дозволяє уникнути SQL-ін'єкцій.

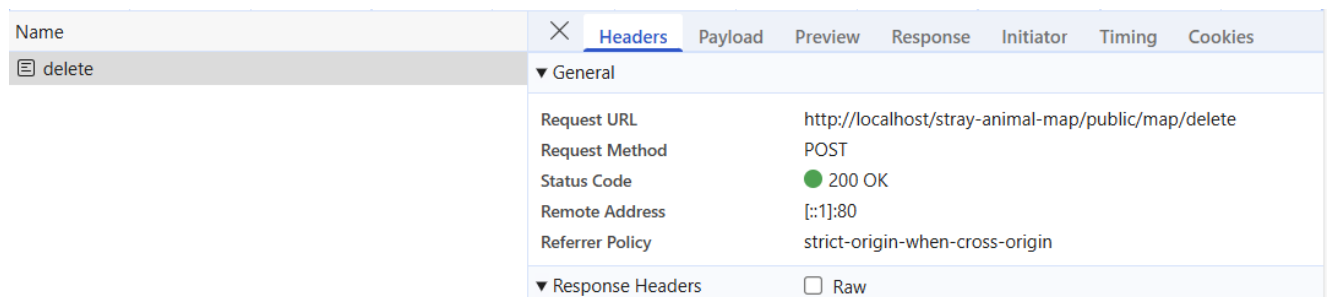


Рисунок 1.12. Результат роботи функції видалення

З боку користувача взаємодія є максимально інтуїтивною: після натискання кнопки підтвердження користувач бачить повідомлення про успішне видалення, а відповідний маркер зникає з карти. Якщо ж операція завершується помилкою (наприклад, через відсутній запис у базі), користувач отримує відповідне попередження.

Функція видалення маркерів також має важливе значення у контексті підтримки інформаційної достовірності мапи. Завдяки їй користувачі можуть не

лише додавати нові точки, а й очищувати мапу від застарілої або некоректної інформації. Швидкість запису сприяє підтримці актуальності системи та підвищує довіру до неї з боку громади.

З технічної точки зору дана функціональність демонструє гнучкість MVC-архітектури проекту, яка дозволяє легко розширювати застосунок без суттєвих змін у структурі. Уся логіка була реалізована за допомогою вже наявних моделей та контролерів, що засвідчує хорошу модульність системи.

Додавання функції видалення стало черговим кроком до підвищення ефективності взаємодії користувача з веб-застосунком. Реалізація запити на видалення розширило можливості системи у напрямі забезпечення точного, актуального та контрольованого моніторингу безпритульних тварин на території міста.

					<i>РП 08. 02 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

## 2 ЕКОНОМІЧНИЙ РОЗДІЛ

### 2.1 Резюме

У дипломному проєкті розроблено веб-систему для допомоги у виявленні безпритульних тварин. Він забезпечує фіксацію місць їхнього перебування, візуалізацію даних на мапі та обробку звернень. Оцінювання якості відбувається за критеріями зручності, результативності та актуальності. Якість системи оцінюється за критеріями зручності використання, продуктивності та практичної цінності.

### 2.2 Визначення трудомісткості розробки ПЗ

Тривалість розробки програмного продукту залежить від його обсягу, складності, кваліфікації розробників і встановлених ринком термінів. Метод структурної аналогії дозволяє оцінити обсяг у тисячах умовних машинних команд на основі подібного програмного забезпечення.

Табл. 2.1 містить аналоги ПЗ з подібними функціями; обраний варіант виділено сірим.

Таблиця 2.1. Каталог аналогів

Найменування ПП	Обсяг функції ПП – $V_0$ , умовних. машинних командах
1. ПП автоматизації засобів по каталогу	680 – 7000
2. ПП автоматизованих розрахунків	1300 – 8600
3. ПП введення інформації	1060 – 5750

Після вибору аналога з обсягом  $V_0$  (умовні машинні команди), трудомісткість визначається за табл. 2.2.

Таблиця 2.2. Обсяг ПП

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262
4.00	283
5.00	306
6.00	330
7.00	357

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
8.00	385
9.00	414
10.00	445

На основі отриманого значення за довідником визначають укрупнену норму часу, скориговану коефіцієнтом  $K_k = 0,7-0,8$  для умов розробки на комп'ютері:

$$T_{ap} = 306 \times 0,8 = 244,8 \text{ (люд/годин)} \quad (2.1)$$

Трудомісткість визначається для кожного етапу окремо, з урахуванням складності, новизни та використання стандартних модулів, за відповідними формулами:

$$T_{T3} = T^a \rho \times L_1 \times K_H \quad (2.2)$$

$$T_{TII} = T^a \rho \times L_2 \times K_H \quad (2.3)$$

$$T_{PPI} = T^a \rho \times L_3 \times K_H \times K_T \quad (2.4)$$

Для розрахунку використовуються такі коефіцієнти:

- $L_i$  – частка  $i$ -го етапу (табл. 2.3);
- $K_H$  – коефіцієнт новизни (табл. 2.4);
- $K_T$  – коефіцієнт використання типових програм (табл. 2.5).

Наш варіант виділено сірим.

Таблиця 2.3. Питомі коефіцієнти трудомісткості стадії у загальній трудомісткості розробки ПП

Код стадії	Ступінь новизни		
	А	Б	В
T3 ( $L_1$ )	0,15	0,12	0,12
TII ( $L_2$ )	0,16	0,15	0,11
PPI ( $L_3$ )	0,55	0,58	0,61

Таблиця 2.4. Значення коефіцієнта новизни

Код ступеня новизни	Ступінь новизни	Значення $K_H$
А	Принципово новий ПП	1,75 – 1,2
Б	ПП – розвиток визначеного параметричного ряду	1,0 – 0,8

Код ступеня новизни	Ступінь новизни	Значення $K_n$
В	ПП, що має аналог	0,7

Таблиця 2.5. Значення коефіцієнта використання типових програм

Ступінь охоплення реалізованих функцій розробленого ПП типовими програмами, %	Значення $K_T$
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Тепер розраховуємо трудомісткість для всіх етапів і зводимо у табл. 2.6:

Трудомісткість технічного завдання:

$$T_{tz} = Ta * L_1 * K_n = 244,8 * 0,12 * 0,8 = 23,50 \text{ (люд/годин)} \quad (2.2)$$

Трудомісткість розробки технічного проєкту:

$$T_{tp} = Ta * L_2 * K_n = 244,8 * 0,15 * 0,8 = 29,37 \text{ (люд/годин)} \quad (2.3)$$

Трудомісткість розробки робочого проєкту:

$$T_{rp} = Ta * L_3 * K_n * K_m = 244,8 * 0,58 * 0,8 * 0,7 = 79,51 \text{ (люд/годин)} \quad (2.4)$$

Для розрахунків визначили обсяг документації по етапах:

- технічне завдання  $N_{tz}=2$  (стор);
- розробка ТП  $N_{tp}=45$  (стор);
- розробка робочого проєкту  $N_{rp}=4$  (стор);
- пояснювальна записка відповідно  $N_{pz}=9$  (стор).

Таблиця 2.6. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин		
	1.ТЗ	$T_{РТЗ}=23,50$	$T_{кк}=0,7*N_{ТЗ}=0,7*2=1,4$
2.Розробка ТП	$T_{РТП}=29,37$	$T_{кк}=0,7*N_{ТП}=0,7*45=31,5$	$T_{нк}=0,15*N_{ТП}=0,15*45=6,75$
3.Розробка РП	$T_{Ррп}=79,51$	$T_{кк}=0,7*N_{рп}=0,7*9=6,3$	$T_{нк}=0,15*N_{рп}=0,15*9=1,35$
4.Розробка ПЗ	$T_{Пз}=1,5*N_{Пз}=1,5*9=13,5$	$T_{кк}=0,7*N_{ТЗ}=0,7*9=6,3$	$T_{нк}=0,15*N_{Пз}=0,15*9=1,35$

Усього, в т.ч.:	201,07		
- на розробку	Тр=145,82		
- контроль керівника		Ткк=45,5	
- нормоконтроль			Тнк=9,75

### 2.3 Розрахунок ціни програмного продукту

Розраховуємо основну зарплату виконавців, матеріальні та загальні витрати на розробку ПП. Зарплата наведена в табл. 2.7. З 1 січня 2025 року мінімальна місячна зарплата – 8000 грн, погодинна ставка – 48 грн (згідно зі ст. 8 Закону про Держбюджет України).

Таблиця 2.7. Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	145,82	60,00	8749,20
2.Контроль керівника	45,5	120,00	5460,00
3.Нормоконтроль	9,75	120,00	1170,00
Усього	-	-	Зо= 15379,20

Розрахуємо матеріальні витрати на розробку ПП та наведемо їх у табл. 2.8.

Таблиця 2.8. Розрахунок матеріальних витрат на розробку

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	70	5.0	350,0
Разом	-	-	-	$V_{Mi}=350,0$
Транспортно – заготівельні Витрати (10%)				40,00
Усього				$V_M=V_{Mi}+V_{тр\_з}= 390,00$

За отриманими даними складена калькуляція планової собівартості ПП, наведена в табл. 2.9.

					<b>РП 08. 02 002. 00 ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

Таблиця 2.9. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	390,00	$V_M$ (див. табл. 2.8)
2. Основна заробітна плата	15379,20	$Z_o$ (див. табл. 2.7)
3. Додаткова заробітна плата	1537,92	$Z_d = 0.1 * Z_o = 15379,20 * 0,1$
4. Відрахування до єдиного фонду соціального внеску	3721,77	$V_{с.с.в.} = 0.22 * (Z_o + Z_d) = 0,22 * (15379,20 + 1537,92)$
5. Накладні витрати	6251,68	$V_{нак.} = 0.4 * Z_o = 0.4 * 15379,20$
6. Повна собівартість	27280,57	$C_{пов} = V_M + Z_o + Z_d + V_{с.с.в.} + V_{нак.} = 390,00 + 15379,20 + 1537,92 + 3721,77 + 6251,68$

Розмір прибутку розраховується за формулою:

$$П = (C_n * P) / 100 = (27280,57 * 10) / 100 = 2728,06 \text{ грн.} \quad (2.5)$$

Де  $p$  – плановий рівень рентабельності (10-15%).

Оптова ціна розраховується за формулою:

$$C_o = C_n + П = 27280,57 + 2728,06 = 30008,63 \text{ грн.} \quad (2.6)$$

За отриманими даними, ціна реалізації ПП за формулою становить:

$$C_p = C_o + ПДВ = 30008,63 + 30008,63 * 0.2 = 36010,35 \text{ грн.} \quad (2.7)$$

## **3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ**

### **3.1 Основні положення**

У сучасному світі розвиток технологій та виробництва невід’ємно пов’язаний з автоматизацією та оптимізацією процесів. Сьогодні важко уявити підприємство, де господарська діяльність ведеться без використання комп’ютерної техніки. Через масовий характер робіт, що виконуються за допомогою комп’ютерів, законодавство України встановлює чіткі норми та вимоги до їх використання, зокрема щодо охорони праці при роботі з комп’ютером.

У розділі охорони праці дипломного проекту розглядається розробка веб-системи для пошуку безпритульних тварин.

### **3.2 Аналіз умов праці й забезпечення безпеки при виконанні основних видів робіт**

Аналіз умов праці показує, що на працівників можуть негативно впливати такі фізичні та психофізіологічні фактори:

- підвищена або знижена температура, вологість повітря робочої зони;
- недостатнє освітлення робочого місця;
- підвищений рівень шуму;
- підвищена іонізація повітря та рівень електромагнітних випромінювань;
- нервово-психічні та фізичні перевантаження.

### **3.3 Гігієнічні вимоги до виробничого середовища**

На робочому місці програміста мають бути створені умови для безпечної та високопродуктивної праці.

#### **3.3.1 Вимоги до приміщення експлуатації ПК**

Приміщення, в яких планується установка та подальша робота з комп’ютером, повинні відповідати проектній документації будинку. Роботодавець повинен враховувати санітарні нормативи освітлення, вимоги до параметрів мікроклімату (температура, відносна вологість), ступеня і сили вібрації, звукового

					<i>РП 08. 02 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

шуму і вогнестійкості приміщення, а також характеристики електромагнітного, ультрафіолетового та інфрачервоного полів.

Заборонено встановлювати комп'ютери в приміщеннях, розташованих у підвалах будинків. Для уникнення можливих аварій та замикань, поряд з приміщеннями, де вестиметься робота з комп'ютером (над чи під ними), також не дозволяється проведення робіт, що потребують здійснення надмірно вологих технологічних процесів.

### 3.3.2 Вимоги до приміщення експлуатації ПК

Робота за енерговитратами відноситься до категорії легких робіт 1а, 1б, тому повинні дотримуватися наступні вимоги згідно ДСанПіН 3.3.2.-007-98.

У таблиці 3.1 наведено норми мікроклімату для приміщень з ВДТ ЕОМ та ПЕМ.

Таблиця 3.1. Норми мікроклімату для приміщень з ВДТ ЕОМ та ПЕМ

Пора року	Категорія робіт	Температура повітря, °С, не більше	Відносна вологість повітря %	Швидкість руху повітря, м/с
Холодна	Легка-1а	22-24	40-60	0,1
	Легка-1б	21-23	40-60	0,1
Тепла	Легка-1а	23-25	40-60	0,1
	Легка-1б	22-24	40-60	0,1

Рівні позитивних і негативних іонів у повітрі приміщень з ВДТ мають відповідати санітарно-гігієнічним нормам № 2152-80.

У таблиці 3.2 наведено санітарно-гігієнічні нормам № 2152-80.

Таблиця 3.2. Санітарно-гігієнічні нормам № 2152-80

Рівні	Число іонів в 1 см <sup>3</sup> повітря	Число іонів в 1 см <sup>3</sup> повітря
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально допустимі	50000	50000

Для підтримки в приміщенні нормального складу повітря, що відповідає гігієнічним вимогам, та видалення з нього шкідливих речовин використовують

вентиляцію. При природній вентиляції (за допомогою вікон) повітря надходить у приміщення і видаляється внаслідок різниці температур. Але вона має низку недоліків. Тому дипломним проектом передбачено встановлення припливно-витяжної системи вентиляції, а також можливе застосування кондиціонерів.

### **3.3.3 Виробниче освітлення**

Освітлення приміщення має природне та штучне походження. Природне освітлення подається через віконні прорізи, бокове. Для штучного освітлення у приміщенні використовуються люмінесцентні лампи, які в порівнянні з лампами розжарювання мають ряд істотних переваг. Так, за спектральним складом світла вони близькі до природного світла, мають підвищену світлову віддачу, триваліший термін служби. Норма освітленості на робочих місцях складає 300-500 лк.

### **3.3.4 Електробезпека**

Значення сили струму, що проходить через організм людини, залежить від напруги, під якою перебуває людина, і від опору ділянки тіла, до якого прикладена ця напруга. Джерелом живлячої напруги є мережа змінного струму з напругою 220 В, на яку поширюється ГОСТ 25861-83.

Для попередження уражень електричним струмом необхідно чітко й у повному обсязі виконувати правила проведення робіт і правила технічної експлуатації. Необхідно виключити можливість доступу оператора до частин устаткування, що працює під небезпечною напругою, до неізольованих частин, призначених для роботи при малій напрузі й не підключених до захисного заземлення, а також підводити електроживлення до ПЕОМ від розетки за допомогою спеціальної вилки із заземлюючим контактом.

### **3.3.5 Організація робочого місця з ПК**

Робоче місце повинно відповідати ергономічним та санітарним вимогам (ДСанПіН 3.3.2.-007-98):

- індивідуальне регулювання меблів (висота столу, крісла);
- дисплей розташовується так, щоб верхній край був на рівні очей;
- відстань до монітора: 60–90 см, рекомендовано ~70 см;

					<i>РП 08. 02 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

- частота мерехтіння екрана: не менше 70 Гц, рекомендовано 100 Гц;
- поверхня столу – матова;
- забезпечення відсутності відблисків на екрані.

На рисунку 3.1 зображено конструкцію робочого місця користувача ПК.

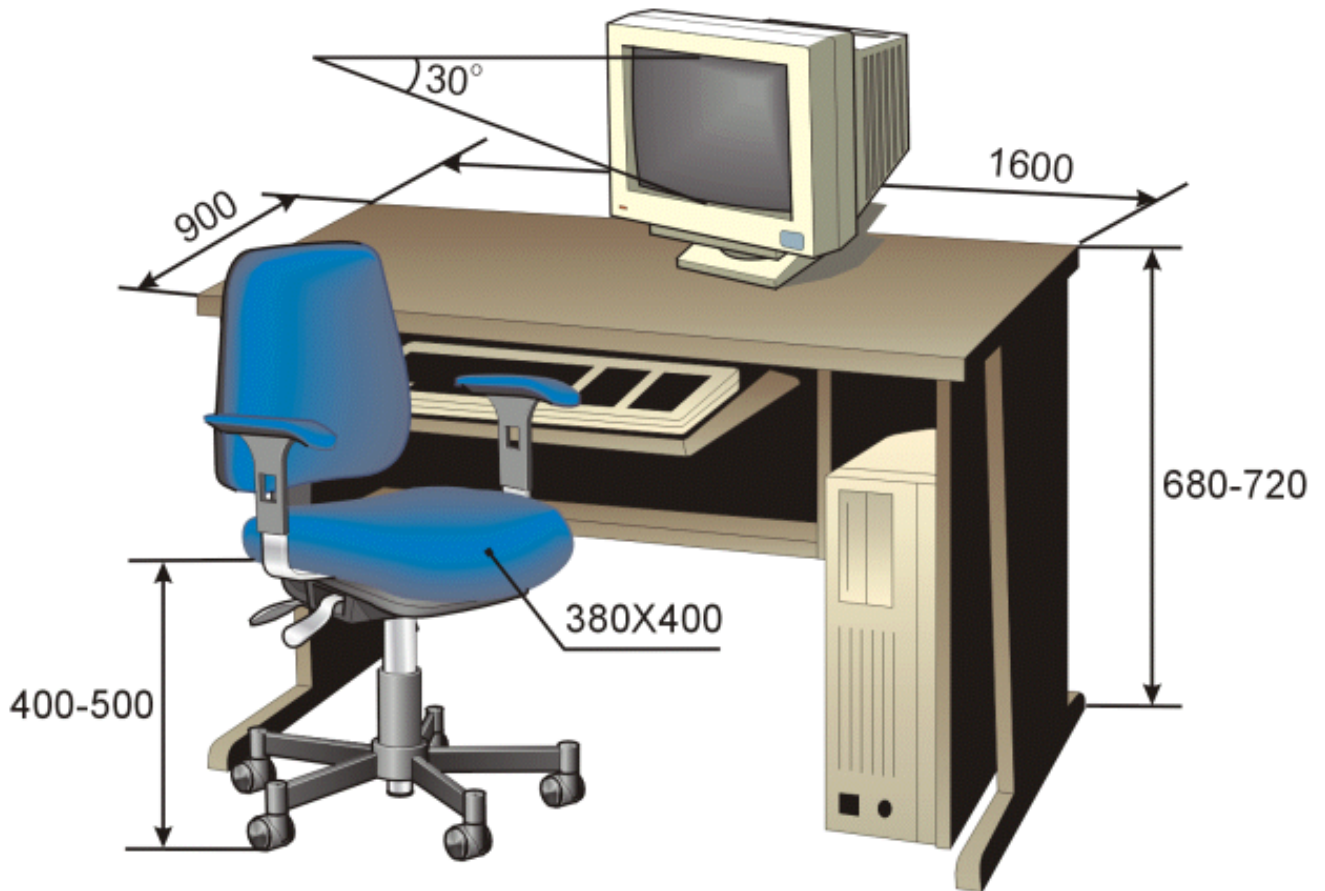


Рисунок 3.1. Конструкція робочого місця користувача ПК

Для зниження втоми та гіподинамії рекомендовано виконання вправ відповідно до ДСанПіН 3.3.2.-007-98.

### 3.4 Пожежна безпека

Приміщення, у якому виконується робота з персональними комп'ютерами, повинне відповідати вимогам пожежної безпеки, визначеним у нормативних документах, таких як НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні».

Основними джерелами виникнення пожеж у приміщеннях з ПК є коротке замикання електромережі, перевантаження мереж, несправність електрообладнання або необережне поводження з електроприладами.

					<i>РП 08. 02 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

З метою забезпечення пожежної безпеки необхідно дотримуватись наступних вимог:

- використовувати тільки справні електроприлади з сертифікацією;
- не допускати перенавантаження електромережі;
- розетки, вимикачі та проводка повинні бути в належному технічному стані;
- після завершення роботи слід вимикати обладнання з мережі;
- на робочому місці повинні бути вогнегасники відповідного типу (наприклад, вуглекислотний або порошковий), у кількості та обсязі згідно з нормами;
- співробітники повинні бути ознайомлені з інструкціями з пожежної безпеки та проходити щорічний інструктаж.

У разі виникнення пожежі необхідно негайно повідомити службу порятунку за номером 101, залишити приміщення, дотримуючись плану евакуації, та по можливості скористатися первинними засобами пожежогасіння.

### **3.5 Висновки**

Організація безпечного робочого місця програміста є важливою складовою ефективного та продуктивного процесу створення програмного забезпечення. Забезпечення належного мікроклімату, ергономічного облаштування, електробезпеки та дотримання режиму праці дозволяє знизити ризики захворювань, зберегти працездатність персоналу та підвищити якість роботи.

					<i>РП 08. 02 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

## ВИСНОВКИ

У результаті виконання роботи створено веб-застосунок для моніторингу безпритульних тварин. Система дозволяє фіксувати місцеперебування тварин, додавати фото, описувати стан, переглядати інформацію на інтерактивній мапі, а також редагувати та фільтрувати дані. Серверна частина реалізована мовою PHP, клієнтська – з використанням JavaScript і бібліотеки Leaflet для картографії.

Під час розробки проаналізовано ситуацію з безпритульними тваринами в Україні та Європі, досліджено інструменти для онлайн-моніторингу соціально важливих явищ, що обґрунтувало вибір архітектури рішення. Зосереджено увагу на зручності інтерфейсу та доступності функціоналу для волонтерів, притулків і громадян.

Система забезпечує швидке внесення даних, зручний пошук та адаптивний інтерфейс для мобільних і десктопних пристроїв. Тестування підтвердило стабільність, відповідність вимогам і зручність використання. Рішення має перспективу розвитку – зокрема інтеграції з іншими сервісами, додавання аналітики та розширення функцій.

					<i>РП 08. 02 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

## ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Білоконська, О. В. Проблеми безпритульних тварин в Україні та шляхи їх вирішення // Екологічна безпека та збалансоване ресурсокористування. – 2020. – № 2. – С. 112–118.
2. Мороз, О. В. Основи веб-програмування: HTML, CSS, JavaScript, Angular. – К. : Академвидав, 2021. – 312 с.
3. Adopt Don't Stop. Український проект допомоги безпритульним тваринам. [Веб-сайт]. URL: <https://adoptdontstop.org/> (дата звернення: 13.06.2025).
4. World Health Organization (WHO). Guidelines for Dog Population Management. – Geneva: WHO, 2015. – 87 p. [Режим доступу: <https://www.who.int/publications-detail/dog-population-management>] (дата звернення: 13.06.2025).
5. Welling, Luke; Thomson, Laura. PHP and MySQL Web Development. 5th Edition. – Addison-Wesley Professional, 2016. – 1008 p.
6. Flanagan, David. JavaScript: The Definitive Guide. 7th Edition. – O'Reilly Media, 2020. – 706 p.
7. Duckett, Jon. JavaScript and JQuery: Interactive Front-End Web Development. – Wiley, 2014. – 640 p.
8. Adopt-a-Pet.com. Find a Pet to Adopt. [Веб-сайт]. URL: <https://www.adoptapet.com/> (дата звернення: 13.06.2025).

					<i>РП 08. 02 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

# ДОДАТОК А. Програмний код MVC-архітектури для пошуку тварин

```
// AnimalModel.php

<?php

require_once __DIR__ . '/../../config/database.php';

class AnimalModel {
    public static function createSpot($user_id, $lat, $lng, $animal_type, $color, $size,
    $gender, $age, $description, $photo) {
        global $pdo;
        $stmt = $pdo->prepare("
        INSERT INTO animal_spots
        (user_id, latitude, longitude, animal_type, color, size, gender, age,
description, photo, created_at)
        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, NOW())
        ");
        $stmt->execute([$user_id, $lat, $lng, $animal_type, $color, $size, $gender,
    $age, $description, $photo]);

        return $pdo->lastInsertId();
    }

    public static function updatePhoto($animal_id, $photo) {
        global $pdo;
        $stmt = $pdo->prepare("UPDATE animal_spots SET photo = ? WHERE id = ?");
        return $stmt->execute([$photo, $animal_id]);
    }

    public static function getAllSpots() {
        global $pdo;
        $stmt = $pdo->query("SELECT * FROM animal_spots ORDER BY created_at DESC");
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
}

// index.php

<script>
    const map = L.map('map').setView([46.4825, 30.7233], 13);

    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
        maxZoom: 19,
    }).addTo(map);

    fetch('/stray-animal-map/public/animal/all')
        .then(res => res.json())
        .then(spots => {
            spots.forEach(s => {
                const photoUrl = s.photo
                ? `/stray-animal-map/public/uploads/${s.photo}`
                : 'https://via.placeholder.com/150?text=Фото+відсутнє';

                L.marker([s.latitude, s.longitude]).addTo(map)
                    .bindPopup(
                        `Тип тварини: ${s.animal_type}<br>
                        <b>Колір:
```

```

        <b>Опис:</b> ${s.description}<br>
        `
    );
    });
});

function addSpot() {
    const center = map.getCenter();

    const body = {
        lat: center.lat,
        lng: center.lng,
        animal_type: document.getElementById('animal_type').value,
        color: document.getElementById('color').value,
        size: document.getElementById('size').value,
        gender: document.getElementById('gender').value,
        age: document.getElementById('age').value,
        description: document.getElementById('description').value
    };

    fetch('/stray-animal-map/public/animal/create', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(body)
    }).then(response => response.json())
        .then(result => {
            if (result.success) {
                const photo = document.getElementById('photo').files[0];
                if (photo) {
                    const photoData = new FormData();
                    photoData.append('photo', photo);
                    photoData.append('animal_id', result.animal_id);

                    return fetch('/stray-animal-map/public/animal/uploadPhoto', {
                        method: 'POST',
                        body: photoData
                    });
                }
            } else {
                alert('Помилка: ' + result.error);
                throw new Error('Зберігання даних не вдалося');
            }
        })
        .then(() => {
            L.marker([center.lat, center.lng]).addTo(map)
                .bindPopup(
                    `

```

```

    }

    function closeModal() {
        document.getElementById('animalModal').style.display = 'none';
    }

    window.onclick = function(event) {
        const modal = document.getElementById('animalModal');
        if (event.target == modal) {
            modal.style.display = 'none';
        }
    }

    document.getElementById('openModalBtn').onclick = function() {
        openModal();
    };
</script>

// AnimalController.php

<?php
require_once __DIR__ . '/../models/AnimalModel.php';

class AnimalController {

    public function create() {
        session_start();
        if (!isset($_SESSION['user_id'])) {
            ob_clean();
            header('Content-Type: application/json; charset=utf-8');
            echo json_encode(['error' => 'Неавторизований користувач.']);
            exit;
        }

        $input = json_decode(file_get_contents('php://input'), true);

        $lat = $input['lat'] ?? null;
        $lng = $input['lng'] ?? null;
        $animal_type = trim($input['animal_type'] ?? '');
        $color = trim($input['color'] ?? '');
        $size = trim($input['size'] ?? '');
        $gender = trim($input['gender'] ?? '');
        $age = trim($input['age'] ?? '');
        $description = trim($input['description'] ?? '');

        if ($lat && $lng && $animal_type && $color && $size && $gender && $age) {
            $animal_id = AnimalModel::createSpot(
                $_SESSION['user_id'],
                $lat,
                $lng,
                $animal_type,
                $color,
                $size,
                $gender,
                $age,
                $description,
                null
            );

            ob_clean();
            header('Content-Type: application/json; charset=utf-8');
            echo json_encode(['success' => true, 'animal_id' => $animal_id]);
        } else {

```

```

        ob_clean();
        header('Content-Type: application/json; charset=utf-8');
        echo json_encode(['error' => 'Будь ласка, заповніть усі поля.']);
    }
}

public function uploadPhoto() {
    session_start();
    if (!isset($_SESSION['user_id'])) {
        ob_clean();
        header('Content-Type: application/json; charset=utf-8');
        echo json_encode(['error' => 'Неавторизований користувач.']);
        exit;
    }

    $animal_id = $_POST['animal_id'] ?? null;

    if ($animal_id && !empty($_FILES['photo']) && $_FILES['photo']['error'] == 0) {
        $targetDir = __DIR__ . '/../..public/uploads/';
        if (!file_exists($targetDir)) {
            mkdir($targetDir, 0777, true);
        }

        $photoName = uniqid() . '_' . basename($_FILES['photo']['name']);
        $targetFile = $targetDir . $photoName;

        if (move_uploaded_file($_FILES['photo']['tmp_name'], $targetFile)) {
            AnimalModel::updatePhoto($animal_id, $photoName);
            ob_clean();
            header('Content-Type: application/json; charset=utf-8');
            echo json_encode(['success' => true]);
        } else {
            ob_clean();
            header('Content-Type: application/json; charset=utf-8');
            echo json_encode(['error' => 'Не вдалося завантажити фото.']);
        }
    } else {
        ob_clean();
        header('Content-Type: application/json; charset=utf-8');
        echo json_encode(['error' => 'Невірні дані для фото.']);
    }
}

public function all() {
    ob_clean();
    header('Content-Type: application/json; charset=utf-8');
    echo json_encode(AnimalModel::getAllSpots());
}
}

```

# ДОДАТОК Б. Слайди мультимедійної презентації

## Розробка інтерактивної мапи для моніторингу безпритульних тварин

---

БЕЛЯЄВА АННА ВІКТОРІВНА

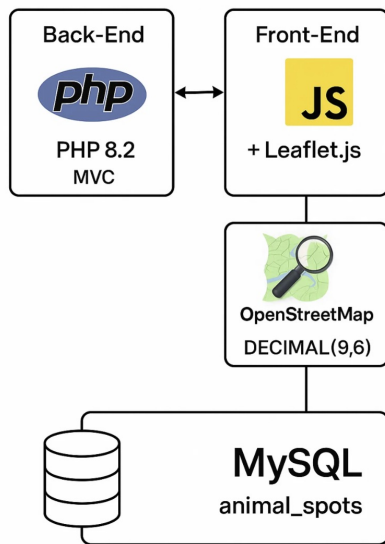
(ГР. 4РП-08)

## Основні положення

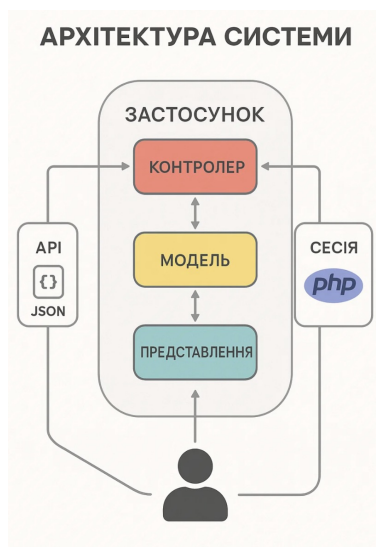
---

Актуальність обраної теми зумовлена соціальною проблемою, яка існує як в Україні, так і в багатьох європейських країнах — значна кількість безпритульних тварин, які не лише страждають, а й можуть нести ризики для населення. Питання захисту тварин, їх обліку та подальшої взаємодії з волонтерами, притулками або небайдужими громадянами потребує сучасних цифрових рішень.

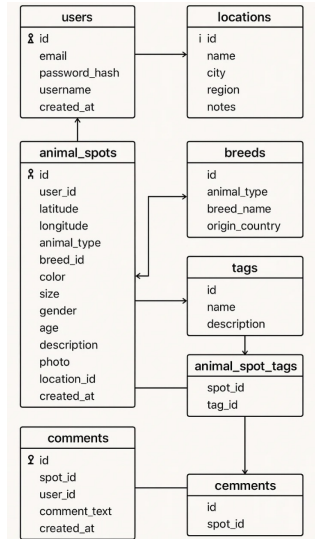
Мета проєкту — створити веб-застосунок з інтерактивною мапою, де користувачі зможуть додавати та переглядати інформацію про знайдених або втрачених тварин у своєму місті. Таким чином, реалізується ідея «цифрової платформи допомоги», що об'єднує мешканців, волонтерів і притулки на базі відкритої інформації.



Використані  
технології



Архітектура  
системи



# Схема бази даних

## Робота з мапою (Leaflet.js)

Мапа створена з використанням Leaflet.js.

Кожна позначка містить координати, опис тварини, тип, вік, стать, фото.

Дані завантажуються динамічно з бази. При кліку на карту відкривається форма додавання тварини.

Користувач сам обирає місце, де її помічено, та додає опис.

```

<script>
const map = L.map('map').setView([46.4825, 30.7233], 13);

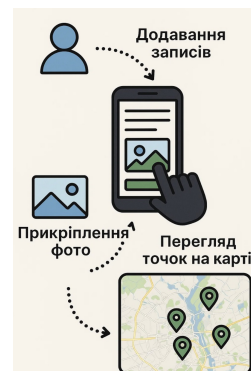
L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
  maxZoom: 19,
}).addTo(map);

fetch('/stray-animal-map/public/animal/all')
  .then(res => res.json())
  .then(spots => {
    spots.forEach(s => {
      const photoUrl = s.photo
        ? '/stray-animal-map/public/uploads/${s.photo}'
        : 'https://via.placeholder.com/150?text=0000+incynne';

      L.marker([s.latitude, s.longitude]).addTo(map)
        .bindPopup(
          `<b>Тип тварини:</b> ${s.animal_type}<br>
          <b>Колір:</b> ${s.color}<br>
          <b>Розмір:</b> ${s.size}<br>
          <b>Стать:</b> ${s.gender}<br>
          <b>Вік:</b> ${s.age}<br>
          <b>Опис:</b> ${s.description}<br>
          <img src=${photoUrl} alt="0000" width="150" height="150">`
        );
    });
  });
</script>
  
```

# Робота з API

Користувачі можуть авторизуватись, додавати записи, прикріплювати фото, а також переглядати всі наявні точки на карті.



# Робота з API

Особливу увагу було приділено роботі з інтерфейсом. Реалізовано модальну форму для додавання тварин, яка відкривається на головній сторінці. Користувач вводить тип тварини, колір, вік, розмір, стать, короткий опис та додає фото. Після натискання кнопки — координати точки додаються до бази даних і маркер автоматично з'являється на мапі. Весь процес реалізовано через API-запити у форматі JSON.

```
AnimalController.php X
app > controllers > AnimalController.php
5 class AnimalController {
7     public function create() {
8         session_start();
9         if (!isset($_SESSION['user_id'])) {
10             ob_clean();
11             header('Content-Type: application/json; charset=utf-8');
12             echo json_encode(['error' => 'Неавторизований користувач.']);
13             exit;
14         }
15         $input = json_decode(file_get_contents('php://input'), true);
16
17         $lat = $input['lat'] ?? null;
18         $lng = $input['lng'] ?? null;
19         $animal_type = trim($input['animal_type'] ?? '');
20         $color = trim($input['color'] ?? '');
21         $size = trim($input['size'] ?? '');
22         $gender = trim($input['gender'] ?? '');
23         $age = trim($input['age'] ?? '');
24         $description = trim($input['description'] ?? '');
25
26
27         if ($lat && $lng && $animal_type && $color && $size && $gender && $age) {
28             $animal_id = AnimalModel::createSpot(
29                 $_SESSION['user_id'],
30                 $lat,
31                 $lng,
32                 $animal_type,
33                 $color,
34                 $size,
35                 $gender,
36                 $age,
37                 $description,
38                 null
39             );
40         }
41     }
42 }
```

# Робота з механізмами

---

Окремо протестовано механізми:

1. авторизації та реєстрації користувачів;
2. завантаження та збереження зображень;
3. додавання та відображення інформації на мапі в режимі реального часу;
4. видалення точки на мапі;
5. перевірку прав доступу через PHP-сесії.

## Реєстрація

---

Після заповнення форми (наприклад, test2@test.com / 12345678) і натискання кнопки, дані надсилаються методом POST на сервер, де відбувається їхня обробка у методі register() контролера AuthController. Пароль одразу хешується функцією password\_hash() і зберігається у базі.

### Реєстрація

Email:

Пароль:

Уже вже є акаунт? [Вхід до системи](#)

# Авторизація

При правильному введенні даних, метод `login()` виконує перевірку email у базі та використання функції `password_verify()` для звірки пароля.

Якщо облікові дані правильні — створюється сесія `$_SESSION['user_id']`, і користувач автоматично перенаправляється на карту (`/map`).

Вхід до акаунту

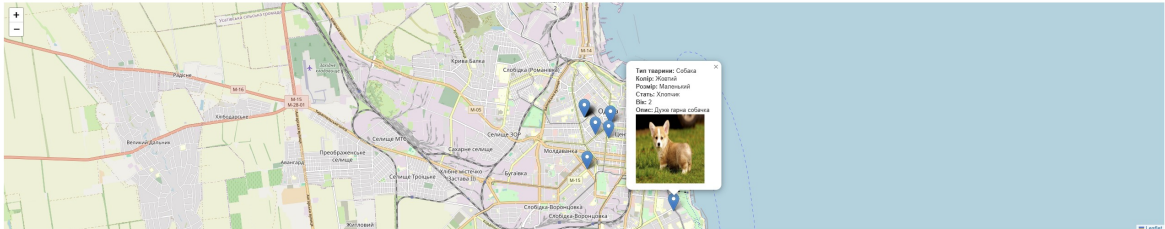
Email:

Пароль:

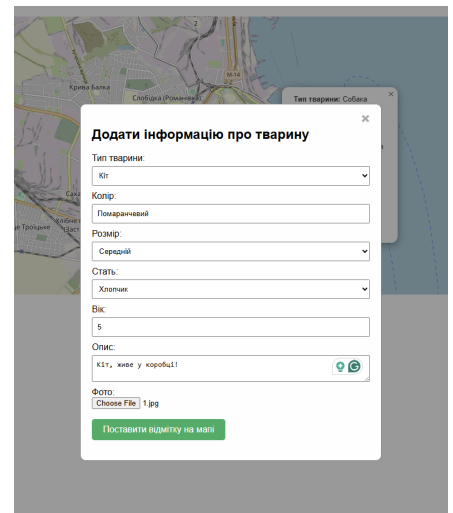
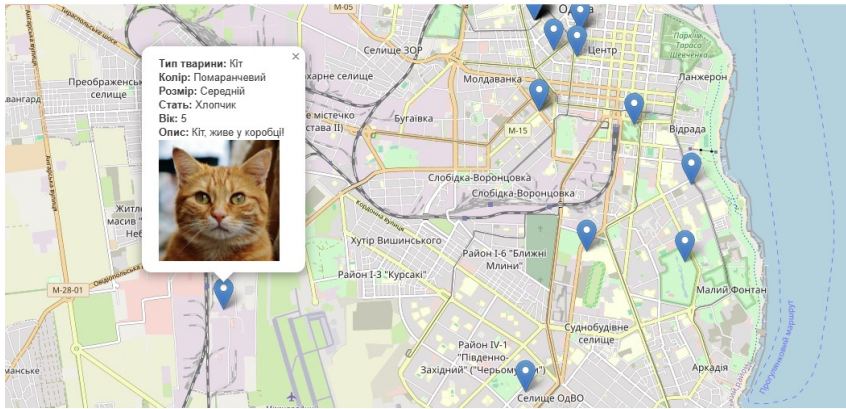
Не має акаунту? [Зареєструйтесь](#)

```
4
5 class AuthController {
6     public function login() {
7         if ($SERVER['REQUEST_METHOD'] === 'POST') {
8             $email = isset($_POST['email']) ? $_POST['email'] : '';
9             $password = isset($_POST['password']) ? $_POST['password'] : '';
10
11             $user = UserModel::findByEmail($email);
12
13             if ($user && password_verify($password, $user['password_hash'])) {
14                 session_start();
15                 $_SESSION['user_id'] = $user['id'];
16                 header('Location: /stray-animal-map/public/map');
17                 exit();
18             } else {
19                 $error = "Помилка в email чи паролю";
20             }
21         }
22
23         require __DIR__ . '/../views/auth/login.php';
24     }
25 }
```

Вітаємо на мапі бездомних тварин!



# Структура мапи



## Додавання маркера на мапу

## Висновки і подальший розвиток

В результаті, я створила повністю функціональний прототип, який може бути адаптований для використання волонтерськими організаціями, притулками або муніципальними службами. Важливо зазначити, що завдяки вибору відкритих технологій, таких як Leaflet.js, OpenStreetMap, Firebase (для тестів), система не потребує ліцензійних платежів і може вільно розвиватись. Таким чином, реалізований веб-застосунок вирішує не лише технічну, а й соціальну проблему, сприяє зменшенню кількості безпритульних тварин та підвищенню обізнаності громадськості.



## РЕЦЕНЗІЯ

на дипломний проект здобувача (здобувачки) освіти  
відділення комп'ютерних систем

*Беляєвої Анни Вікторівни*

(прізвище, ім'я та по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Розробка програмного забезпечення»

Керівник дипломного проекту (роботи) Закроєв Юрій Михайлович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка веб-застосунку інтерактивної мапи пошуку безпритульних тварин

Обсяг розрахунково-пояснювальної записки 74 сторінок

Обсяг графічної (презентаційної) частини 14 аркушів (слайдів)

### ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту завданню

*Представлений на рецензію дипломний проект відповідає затвердженій темі та виконаний відповідно технічному завданню. Дипломний проект присвячений проблемі пошуку тварин та складається з пояснювальної записки, додатку з програмним кодом та мультимедійної презентації, що містить приклади роботи програми.*

б) характеристика виконання кожного розділу дипломного проекту

*Пояснювальна записка складається з основного розділу (аналізу предметної області, проектування застосунку, реалізації застосунку, тестування застосунку), економічного розділу, розділу охорони праці та додатків. Перелічені розділи поетапно охоплюють розробку, виконані докладно та обґрунтовано. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки оператора КТ. Економічний розділ проекту містить розрахунок витрат на НДР та реалізацію проекту.*

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

*Графічна частина складається з 14 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять ілюстративні схеми, скріншоти роботи програмного застосунку, передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання графічної частини проекту та пояснювальної записки добра, розробку виконано у повному обсязі.*

г) перелік позитивних якостей дипломного проекту Чітка структура проекту з розділенням логіки та належною організацією файлової системи. Реалізація функціоналу реєстрації, авторизації та керування сесіями користувача з дотриманням вимог безпеки. Інтерактивна робота з картою, включаючи додавання маркерів, обробку даних, завантаження фото та реалізацію фільтрації.

д) основні недоліки дипломного проекту Мінімальне використання сучасних фреймворків, що могло б надати більші можливості для масштабування та безпеки. Бажано було б розглянути можливість додавання додаткових модулів: кластеризація маркерів, інтеграція з геолокаційними API або підтримка WebSocket для оновлень у реальному часі.  
Деякі недоліки оформлення пояснювальної записки

Оцінка розрахункової частини	<u>Відмінно</u>
Оцінка графічної частини	<u>Добре</u>
Загальна оцінка	<u>Відмінно</u>

Прізвище, ім'я, по батькові рецензента к.т.н. Рудніченко Микола Дмитрович

Місце роботи і посада рецензента Національний університет «Одеська політехніка», доцент кафедри інформаційних технологій



2025 р.

## ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти  
відділення комп'ютерних систем

*Беляєвої Анни Вікторівни*

(прізвище, ім'я та по батькові)

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Тема дипломного проекту: Розробка веб-застосунку інтерактивної мапи пошуку безпритульних тварин.

### ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка до дипломного проекту містить 74 сторінки. У пояснювальній записці описано етапи розробки веб-застосунку інтерактивної мапи пошуку безпритульних тварин засобами PHP, MVC, MySQL, HTML, CSS та JavaScript. Графічна частина складається з окремих 14 слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної записки та слайдів добра.

б) самостійність роботи над проектом: Протягом виконання дипломного проекту здобувачка освіти поступово та послідовно виконувала всі етапи, проявляла ініціативу в створенні загальної концепції та реалізації роботи. Всі роботи здобувачка освіти виконувала самостійно, з оглядом на рекомендації керівника.

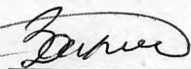
в) теоретична підготовка випускника (випускниці): Здобувачка освіти під час роботи над дипломним проектом вивчила достатньо багато літературних та інтернет-джерел за даною тематикою. Вважаю, що теоретична підготовка дипломника достатня і вона готова до захисту проекту.

г) вміння розв'язувати виробничі та конструкторські питання Під час виконання дипломного проекту здобувачка освіти показала вміння організовано працювати над поставленим завданням, застосовувати знання у галузі програмування та математики, розробляти, встановлювати та налаштовувати спеціалізоване програмне забезпечення, оформлювати слайди та складати презентації, користуючись сучасними комп'ютерними програмними засобами, такими як MS VS Code, PHP, MVC, HTML, CSS, JavaScript.

Оцінка розрахункової частини Добре  
Оцінка графічної частини Відмінно  
Загальна оцінка Відмінно

Прізвище, ім'я, по батькові керівника дипломного проекту Закроєв Юрій Михайлович

Місце роботи і посада керівника дипломного проекту Директор ТОВ «Біг ВОШ»

Підпис 

«20» 06 2025 р.

**ДОЗВІЛ  
НА РОЗМІЩЕННЯ  
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ  
(ДИПЛОМНОГО ПРОЕКТУ)  
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

**Беляєва Анна Вікторівна,**  
здобувачка освіти гр. 4РП-08, та

**Закроєв Юрій Михайлович,**  
керівник дипломного проекту,

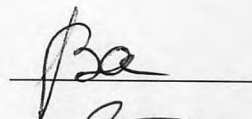
не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

**«Розробка веб-застосунку інтерактивної мапи пошуку безпритульних тварин» (авторка роботи – Беляєва А.В., керівник роботи – Закроєв Ю.М.)**

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2025 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

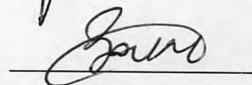
Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Беляєва А.В. /

Керівник



/ Закроєв Ю.М. /

«16» червня 2025 р.

# Д О В І Д К А

циклової комісії КТ та ПІ  
про допуск до захисту дипломного проєкту  
здобувача (здобувачки) освіти ІV курсу  
відділення комп'ютерних систем групи 4РП-08

*Беляєвої Анни Вікторівни*

на тему Розробка веб-застосунку інтерактивної мапи  
пошуку безпритульних тварин

Висновок відповідальної особи за проведення нормоконтролю:  
пояснювальна записка до дипломного проєкту виконана з некритичними  
порушеннями ДСТУ та оформлена відповідно до вимог Положення про  
дипломне проєктування

  
(підпис)

20.06.2025  
(дата)

Петрашова В.І.  
(П.І.Б.)

Висновок відповідальної особи за перевірку роботи на наявність академічного  
плагиату згідно звіту про перевірку від 13.06.2025 р. значення коефіцієнту  
подібності в роботі становить 16,14%, коефіцієнт цитування – 1,15%.

  
(підпис)

20.06.2025  
(дата)

Краснокутська К.Г.  
(П.І.Б.)

**Попередня експертиза (малий захист) дипломного проєкту**

здобувача (здобувачки) освіти

Беляєвої А.В.  
(П.І.Б.)

проведена « 20 » червня 2025 р.

Висновки Пояснювальна записка до дипломного проєкту виконана у повному  
обсязі. Випускна кваліфікаційна робота (дипломний проєкт) відповідає  
вимогам Положення про дипломне проєктування та рекомендована до  
захисту.

Голова ЦК КТ та ПІ

  
(підпис)

Кривченко Ю.В.  
(П.І.Б.)

## Звіт подібності

## метадані

Назва організації

Odesa Technical Professional College of Odesa National University of Technology

Заголовок

Розробка веб-застосунку інтерактивної мапи пошуку безпритульних тварин умовах

Автор

Науковий керівник / Експерт

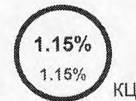
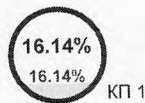
Беляєва Анна Вікторівна Закровс Юрій Михайлович

підрозділ

Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету"

## Обсяг знайдених подібностей

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



25

Довжина фрази для коефіцієнта подібності 2

13733

Кількість слів

115228

Кількість символів

## Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		17
Інтервали		0
Мікропробіли		2
Білі знаки		44
Парафрази (SmartMarks)		94

## Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

## 10 найдовших фраз

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	Копію тексту
1	<a href="https://card-file.ontu.edu.ua/bitstreams/53ed22ad-8700-4162-b97a-082a1ad472d6/download">https://card-file.ontu.edu.ua/bitstreams/53ed22ad-8700-4162-b97a-082a1ad472d6/download</a>	80 0.58 %
2	<a href="https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download">https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download</a>	62 0.45 %
3	<a href="https://card-file.ontu.edu.ua/bitstreams/2b75599e-e1ac-412d-bf09-10d2eb49022f/download">https://card-file.ontu.edu.ua/bitstreams/2b75599e-e1ac-412d-bf09-10d2eb49022f/download</a>	58 0.42 %

4	Розробка 3D-гри у жанрі survival-horror з налаштуваннями рівнів складності 6/12/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	50 0.36 %
5	<a href="https://card-file.ontu.edu.ua/bitstreams/bbed74c8-2ea7-44c5-8d00-0fe3fd9790ee/download">https://card-file.ontu.edu.ua/bitstreams/bbed74c8-2ea7-44c5-8d00-0fe3fd9790ee/download</a>	41 0.30 %
6	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/4bb7255e-46d4-4349-9726-9698476da02d/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/4bb7255e-46d4-4349-9726-9698476da02d/content</a>	39 0.28 %
7	<a href="https://card-file.ontu.edu.ua/bitstreams/29489599-0581-4ce6-8890-c3b13d9f2e0e/download">https://card-file.ontu.edu.ua/bitstreams/29489599-0581-4ce6-8890-c3b13d9f2e0e/download</a>	37 0.27 %
8	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/995bdcec-4e4d-4321-8070-4d6badcb8e49/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/995bdcec-4e4d-4321-8070-4d6badcb8e49/content</a>	36 0.26 %
9	<a href="https://card-file.ontu.edu.ua/bitstreams/53ed22ad-8700-4162-b97a-082a1ad472d6/download">https://card-file.ontu.edu.ua/bitstreams/53ed22ad-8700-4162-b97a-082a1ad472d6/download</a>	30 0.22 %
10	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content</a>	30 0.22 %

### з домашньої бази даних (0.84 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Розробка 3D-гри у жанрі survival-horror з налаштуваннями рівнів складності 6/12/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	102 (4) 0.74 %
2	Розробка web-застосунку для генерації повідомлень із використанням технологій штучного інтелекту 6/11/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	8 (1) 0.06 %
3	Створення web-застосунку цифрового помічника з використанням Open AI 5/28/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	6 (1) 0.04 %

### з програми обміну базами даних (0.61 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Крута_409_inz_2025 5/28/2025 Ukrainian national aviation university (Фаховий коледж інженерії, управління та землевпорядкування Національного авіаційного університету)	47 (9) 0.34 %
2	Розробка веб-застосунку «Фото банк» , захистом скачування фото 6/16/2023 State University of Telecommunications (HHIT)	12 (1) 0.09 %
3	Lb_1_Nasheba_A.docx 4/22/2023 Yuriy Fedkovych Chernivtsi National University(CNU) course papers (Deanery)	11 (1) 0.08 %
4	Курсова (1).pdf 6/11/2025 National University of Water and Environmental Engineering (National University of Water and Environmental Engineering)	8 (1) 0.06 %

5 ВЕБСИСТЕМА ФІКСАЦІЇ АРХЕОЛОГІЧНИХ МАТЕРІАЛІВ В ПОЛЬОВИХ УМОВАХ 6 (1) 0.04 %  
 6/25/2024  
 Rivne Professional college of National University of Life and Environmental sciences of Ukraine (ВСП  
 „Рівненський фаховий коледж НУБіП України“)

з Інтернету (14.69 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/44c16132-5f53-48e2-b6c0-61e9a2f0fd75/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/44c16132-5f53-48e2-b6c0-61e9a2f0fd75/content</a>	544 (45) 3.96 %
2	<a href="https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download">https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download</a>	204 (13) 1.49 %
3	<a href="https://card-file.ontu.edu.ua/bitstreams/53ed22ad-8700-4162-b97a-082a1ad472d6/download">https://card-file.ontu.edu.ua/bitstreams/53ed22ad-8700-4162-b97a-082a1ad472d6/download</a>	187 (9) 1.36 %
4	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content</a>	115 (9) 0.84 %
5	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/4bb7255e-46d4-4349-9726-9698476da02d/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/4bb7255e-46d4-4349-9726-9698476da02d/content</a>	104 (6) 0.76 %
6	<a href="https://card-file.ontu.edu.ua/bitstreams/bbed74c8-2ea7-44c5-8d00-0fe3fd9790ee/download">https://card-file.ontu.edu.ua/bitstreams/bbed74c8-2ea7-44c5-8d00-0fe3fd9790ee/download</a>	95 (6) 0.69 %
7	<a href="https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download">https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download</a>	88 (2) 0.64 %
8	<a href="https://card-file.ontu.edu.ua/bitstreams/29489599-0581-4ce6-8890-c3b13d9f2e0e/download">https://card-file.ontu.edu.ua/bitstreams/29489599-0581-4ce6-8890-c3b13d9f2e0e/download</a>	82 (5) 0.60 %
9	<a href="https://card-file.ontu.edu.ua/bitstreams/11562741-24e6-4201-bc41-a00c8013fca1/download">https://card-file.ontu.edu.ua/bitstreams/11562741-24e6-4201-bc41-a00c8013fca1/download</a>	73 (4) 0.53 %
10	<a href="https://card-file.ontu.edu.ua/bitstreams/2b75599e-e1ac-412d-bf09-10d2eb49022f/download">https://card-file.ontu.edu.ua/bitstreams/2b75599e-e1ac-412d-bf09-10d2eb49022f/download</a>	58 (1) 0.42 %
11	<a href="https://card-file.ontu.edu.ua/bitstreams/035f6436-20b4-4ee6-8e99-bede670e308b/download">https://card-file.ontu.edu.ua/bitstreams/035f6436-20b4-4ee6-8e99-bede670e308b/download</a>	56 (6) 0.41 %
12	<a href="https://card-file.ontu.edu.ua/bitstreams/549ee9fe-7574-4ae5-b500-9fe2711f33e6/download">https://card-file.ontu.edu.ua/bitstreams/549ee9fe-7574-4ae5-b500-9fe2711f33e6/download</a>	55 (5) 0.40 %
13	<a href="https://card-file.ontu.edu.ua/bitstreams/62baa43e-b968-4993-bb54-8cf8761a89b2/download">https://card-file.ontu.edu.ua/bitstreams/62baa43e-b968-4993-bb54-8cf8761a89b2/download</a>	55 (5) 0.40 %
14	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/995bdcec-4e4d-4321-8070-4d6badcb8e49/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/995bdcec-4e4d-4321-8070-4d6badcb8e49/content</a>	36 (1) 0.26 %
15	<a href="https://card-file.ontu.edu.ua/bitstreams/dfa57ac3-98fa-4c22-86e7-0549d1254d89/download">https://card-file.ontu.edu.ua/bitstreams/dfa57ac3-98fa-4c22-86e7-0549d1254d89/download</a>	35 (3) 0.25 %
16	<a href="https://card-file.ontu.edu.ua/bitstreams/0e72a3b9-bdd7-4711-a3c6-dedc1d4287cc/download">https://card-file.ontu.edu.ua/bitstreams/0e72a3b9-bdd7-4711-a3c6-dedc1d4287cc/download</a>	34 (4) 0.25 %
17	<a href="https://card-file.ontu.edu.ua/bitstreams/aed610a6-43ef-47e0-9066-e85c89456f3e/download">https://card-file.ontu.edu.ua/bitstreams/aed610a6-43ef-47e0-9066-e85c89456f3e/download</a>	30 (3) 0.22 %
18	<a href="https://codemax.app/snippet/how-to-dynamically-add-markers-using-folium-and-javascript-based-on-user-input-coordinates/">https://codemax.app/snippet/how-to-dynamically-add-markers-using-folium-and-javascript-based-on-user-input-coordinates/</a>	30 (2) 0.22 %
19	<a href="https://leksii.org/4-601.html">https://leksii.org/4-601.html</a>	23 (3) 0.17 %
20	<a href="https://card-file.ontu.edu.ua/bitstreams/55e2b8f2-7d3c-4235-99fc-2be51199b96d/download">https://card-file.ontu.edu.ua/bitstreams/55e2b8f2-7d3c-4235-99fc-2be51199b96d/download</a>	21 (1) 0.15 %
21	<a href="https://yak.koshachek.com/articles/normalni-formi-shem-vidnosin.html">https://yak.koshachek.com/articles/normalni-formi-shem-vidnosin.html</a>	18 (3) 0.13 %
22	<a href="https://essuir.sumdu.edu.ua/bitstream-download/123456789/92975/1/Logvin_bac_rob.pdf">https://essuir.sumdu.edu.ua/bitstream-download/123456789/92975/1/Logvin_bac_rob.pdf</a>	16 (2) 0.12 %
23	<a href="https://www.cs.ubbcluj.ro/files/curricula/2020/syllabus/IR_sem4_MLR5015_ro_bufny_2020_5156.pdf">https://www.cs.ubbcluj.ro/files/curricula/2020/syllabus/IR_sem4_MLR5015_ro_bufny_2020_5156.pdf</a>	13 (1) 0.09 %
24	<a href="https://card-file.ontu.edu.ua/bitstreams/6cf43324-8f08-4031-ba42-f80b18efbcb8/download">https://card-file.ontu.edu.ua/bitstreams/6cf43324-8f08-4031-ba42-f80b18efbcb8/download</a>	12 (2) 0.09 %
25	<a href="https://card-file.ontu.edu.ua/bitstreams/82a6d375-2b69-4233-b80f-fbfd149b7747/download">https://card-file.ontu.edu.ua/bitstreams/82a6d375-2b69-4233-b80f-fbfd149b7747/download</a>	11 (1) 0.08 %
26	<a href="https://dspace.znu.edu.ua/jspui/bitstream/12345/20137/1/%D0%A1%D1%82%D0%B0%D0%BD%D0%B5%D0%BD%D0%BA%D0%BE_%D0%B4%D0%B8%D0%BF%D0%BB%D0%BE%D0%BC.pdf">https://dspace.znu.edu.ua/jspui/bitstream/12345/20137/1/%D0%A1%D1%82%D0%B0%D0%BD%D0%B5%D0%BD%D0%BA%D0%BE_%D0%B4%D0%B8%D0%BF%D0%BB%D0%BE%D0%BC.pdf</a>	6 (1) 0.04 %

27	<a href="https://otipb.at.ua/load/okhorona_praci_pri_roboti_z_komp_39_luterom/24-1-0-5262">https://otipb.at.ua/load/okhorona_praci_pri_roboti_z_komp_39_luterom/24-1-0-5262</a>	6 (1) 0.04 %
28	<a href="https://card-file.ontu.edu.ua/bitstreams/7b1e10b9-0ac2-4b07-afc4-8cdf7db780/download">https://card-file.ontu.edu.ua/bitstreams/7b1e10b9-0ac2-4b07-afc4-8cdf7db780/download</a>	5 (1) 0.04 %
29	<a href="https://card-file.ontu.edu.ua/bitstreams/5badd81-7907-40c5-8572-81a1bd781b2a/download">https://card-file.ontu.edu.ua/bitstreams/5badd81-7907-40c5-8572-81a1bd781b2a/download</a>	5 (1) 0.04 %

### Список прийнятих фрагментів (немає прийнятих фрагментів)

ПОРЯДКОВИЙ НОМЕР	ЗМІСТ	КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-------	---------------------------------------

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»  
Освітньо-професійна програма: «Розробка програмного забезпечення» Група: 4РП-08

Дипломний проект  
здобувачки освіти денної форми навчання РП. 08.02.000.ДП

БЕЛЯЄВОЇ  
АННИ ВІКТОРІВНИ

м. Одеса  
2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»  
Освітньо-професійна програма: «Розробка програмного забезпечення»  
Група: 4РП-08

ПОЯСНЮВАЛЬНА ЗАПИСКА  
до дипломного проекту на тему:

Проектний матеріал складається з пояснювальної записки на \_\_\_\_\_ сторінках та графічного (презентаційного) матеріалу на \_\_\_\_\_ аркушах (слайдах)

Дипломник \_\_\_\_\_ (Беляєва А.В.)  
Керівник \_\_\_\_\_ (Закроєв Ю.М.)

Консультанти: з економічного розділу \_\_\_\_\_ (Канський М.Ю.)  
з розділу охорони праці та техніки безпеки \_\_\_\_\_ (Чорновол Н.І.) з нормоконтролю \_\_\_\_\_ (Петрашова В.І.) старший консультант \_\_\_\_\_ (Кривченко Ю.В.) До захисту допущений Голова циклової комісії \_\_\_\_\_ (Кривченко Ю.В.)  
Завідувач відділення \_\_\_\_\_ (Краснокутська К.Г.)

Захист « \_\_\_\_\_ » 2025 р. Протокол ЕК No \_\_\_\_\_ Оцінка ЕК \_\_\_\_\_ Секретар ЕК \_\_\_\_\_

Розробка веб-застосунку інтерактивної мапи пошуку

безпритульних тварин  
умовах

14  
74

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення \_\_\_\_\_ Комісія \_\_\_\_\_  
Спеціальність \_\_\_\_\_  
Освітньо-професійна програма \_\_\_\_\_

ЗАТВЕРДЖУЮ: