

Полтавський університет економіки і торгівлі
Навчально-науковий інститут денної освіти
Форма навчання денна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту

Завідувач кафедри

_____ Олена ОЛЬХОВСЬКА

«__» _____ 2026 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«РОЗРОБКА ВЕБ-ДОДАТКУ З НАДАННЯ КОНСУЛЬТАЦІЙ

ПО ДОГЛЯДУ ЗА ТВАРИНАМИ»

зі спеціальності 122 «Комп'ютерні науки»

освітня програма «Комп'ютерні науки»

ступеня бакалавр

Виконавець роботи Дем'яненко Олександр Євгенійович

_____ «__» _____ 202_р.

Науковий керівник Чілікіна Тетяна Володимирівна

_____ «__» _____ 202_р.

Рецензент _____

ПОЛТАВА 2026

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

Олена ОЛЬХОВСЬКА

«__» червень 202__ р.

ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему: «Розробка веб-додатку з надання консультацій по догляду за тваринами»

зі спеціальності 122 «Комп'ютерні науки»

освітня програма «Комп'ютерні науки»

ступеня бакалавр

Прізвище, ім'я, по батькові: Дем'яненко Олександр Євгенійович

Затверджена наказом ректора № ____-Н від «__» _____ 202_ р.

Термін подання студентом роботи «__» _____ 202_ р.

Вихідні дані до кваліфікаційно роботи: публікації з теми, веб-сайти за тематикою роботи

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ РОЗДІЛ

2.1. Огляд робіт з аналогічними завданням та реалізацією

2.2. Переваги та недоліки оглянутих програм

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Основні складові веб-додатків для надання консультаційних послуг

3.2. Особливості інформаційної підтримки користувачів у сфері догляду за тваринами

3.3. Проектування архітектури програмного забезпечення

3.4. Графічне представлення архітектури

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1 Опис розробки веб-додатку з надання консультацій по догляду за тваринами

4.2. Інструкція роботи з веб-додатком

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

Консультанти розділів кваліфікаційної роботи

Розділ	ПІБ, посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Постанова задачі	Чілікіна Т.В.		
Інформаційний огляд	Чілікіна Т.В.		
Теоретична частина	Чілікіна Т.В.		
Практична частина	Чілікіна Т.В.		

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику		
3. Постановка задачі		
4. Інформаційний огляд джерел бібліотек та інтернету		
5. Теоретична частина		
6. Практична частина		
7. Закінчення оформлення		
8. Доповідь студента на кафедрі		
9. Доопрацювання (за необхідністю), рецензування		

Дата видачі завдання «__» _____ 202_ р.

Здобувач вищої освіти Дем'яненко Олександр Євгенійович

Науковий керівник Чілікіна Т.В.

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на _____

(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від « _____ » _____ 202_ р.

Секретар ЕК _____

Затверджую

Зав. кафедрою _____

к.ф.-м.н. Олена Ольховська

« ____ » _____ 202_ р.

Погоджено

Науковий керівник _____

к.ф.-м.н. Чілікіна Т.В.

« ____ » _____ 202_ р.

План

ВСТУП

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ РОЗДІЛ

2.1. Огляд робіт з аналогічними завданням та реалізацією

2.2. Переваги та недоліки оглянутих програм

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Основні складові веб-додатків для надання консультаційних послуг

3.2. Особливості інформаційної підтримки користувачів у сфері догляду за тваринами

3.3. Проектування архітектури програмного забезпечення

3.4. Графічне представлення архітектури

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1 Опис розробки веб-додатку з надання консультацій по догляду за тваринами

4.2. Інструкція роботи з веб-додатком

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

Здобувач вищої освіти _____ Дем'яненко О.Є.

« ____ » _____ 202_ р.

РЕФЕРАТ

Записка: 39 сторінок, рисунків 10, 1 додаток, 27 літературних джерел.

Метою роботи є розробка веб-додатку з надання консультацій по догляду за тваринами, який дозволяє переглядати інформаційні матеріали, створювати консультаційні заявки, зберігати їх у браузері та виконувати попередню оцінку терміновості звернення за описом симптомів.

Об'єктом розробки є процес створення інформаційної підтримки власників домашніх тварин у вигляді веб-додатку

Предметом розробки є веб-додаток з надання консультацій по догляду за тваринами.

Методи дослідження – у процесі виконання роботи були застосовані методи аналізу предметної області, методи проєктування інформаційної архітектури та веб-дизайну, методи створення веб-додатків. Для створення веб-сайту використано HTML5, CSS3, JavaScript, бібліотеки Google Material Icons та IonIcons. Організацію стилів реалізовано на основі CSS Grid та Flexbox. Для динамічної поведінки застосовано чистий JavaScript без використання додаткових фреймворків, а для збереження даних у навчальній версії додатку використано локальне сховище браузера (localStorage).

КЛЮЧОВІ СЛОВА: веб-додаток, HTML5, CSS3, JavaScript, догляд за тваринами.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	11
РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД	14
2.1 Огляд робіт з аналогічними завданнями та реалізацією	14
2.2 Переваги та недоліки оглянутих програм.....	16
РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА.....	19
3.1 Основні складові вебдодатків для надання консультаційних послуг.....	19
3.2 Особливості інформаційної підтримки користувачів у сфері догляду за тваринами.....	20
3.3 Проектування архітектури програмного забезпечення.....	21
3.4 Графічне представлення архітектури	23
РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА.....	25
4.1 Опис розробки веб-додатку з надання консультацій по догляду за тваринами.....	25
4.2 Інструкція роботи з веб-додатком.....	29
ВИСНОВКИ.....	38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	40
ДОДАТКИ.....	39

ВСТУП

Домашні тварини для багатьох людей є частиною повсякденного життя. Власники собак, котів, птахів, кроликів та інших тварин постійно стикаються з питаннями догляду: чим годувати тварину, як підтримувати гігієну, як реагувати на зміни поведінки, коли потрібна профілактика, а коли вже варто звертатися до ветеринарного спеціаліста. На практиці не кожен власник має достатньо знань, щоб швидко оцінити ситуацію та знайти правильне рішення.

Найчастіше люди шукають відповіді в Інтернеті. Це зручно, але не завжди ефективно. Інформація розміщена на різних сайтах, форумах, у блогах або на сторінках ветеринарних клінік. Частина матеріалів може бути занадто загальною, частина - складною для розуміння, а іноді користувачу доводиться переглядати багато сторінок, щоб знайти відповідь на просте питання. Через це виникає потреба у веб-додатку, який би зібрав базові поради в одному місці та дав можливість оформити консультаційне звернення.

Актуальність роботи полягає у створенні простого та зрозумілого сервісу для власників домашніх тварин. Такий веб-додаток може допомогти користувачу швидко знайти інформацію за категорією тварини, переглянути поради з догляду, описати проблему через форму заявки та зберегти звернення для подальшого перегляду. Важливо, що розроблюваний додаток не замінює ветеринарного лікаря. Його основне призначення - інформаційна підтримка користувача та організація консультаційних звернень.

Метою роботи є розробка веб-додатку з надання консультацій по догляду за тваринами, який дозволяє переглядати інформаційні матеріали, створювати консультаційні заявки, зберігати їх у браузері та виконувати попередню оцінку терміновості звернення за описом симптомів.

Об'єктом розробки є процес створення інформаційної підтримки власників домашніх тварин у вигляді веб-додатку

Предметом розробки є веб-додаток з надання консультацій по догляду за тваринами.

Практичне значення роботи полягає у створенні працездатного веб-додатку, який може бути використаний як навчальний програмний продукт або як основа для подальшого розвитку консультаційного сервісу. У майбутньому такий додаток можна доповнити серверною частиною, базою даних, особистими кабінетами користувачів і роллю консультанта.

Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел і додатків. У першому розділі визначено постановку задачі. У другому розділі розглянуто існуючі рішення та їхні особливості. У третьому розділі подано теоретичні основи й графічне представлення архітектури. У четвертому розділі описано практичну реалізацію веб-додатку та інструкцію користувача.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Тема роботи пов'язана з розробкою веб-додатку, який має допомагати власникам домашніх тварин отримувати базові поради з догляду та створювати звернення для консультації. Ідея такого сервісу виникає з практичної потреби: люди часто мають тварин удома, але не завжди знають, як правильно реагувати на зміни в їхній поведінці, харчуванні або самопочутті. У таких випадках користувач зазвичай починає шукати інформацію в Інтернеті, але вона може бути розміщена на різних сайтах, відрізнятися за якістю та не завжди відповідати конкретній ситуації.

Розроблюваний веб-додаток не розглядається як заміна ветеринарної допомоги. Його призначення більш просте: надати користувачу зручний інструмент для перегляду загальних рекомендацій, вибору категорії тварини, заповнення консультаційної заявки та збереження таких звернень. У разі серйозних симптомів система повинна звертати увагу користувача на можливу терміновість ситуації, але остаточне рішення все одно має приймати спеціаліст.

Основна задача, яку потрібно вирішити в межах роботи, полягає у створенні зрозумілого та доступного веб-додатку для інформаційної підтримки власників тварин. Користувач повинен мати можливість швидко знайти потрібний розділ, переглянути поради щодо догляду та за потреби створити заявку з описом проблеми. Важливо, щоб інтерфейс не був перевантаженим, а сама логіка роботи сайту залишалася простою навіть для користувача без технічної підготовки.

Метою роботи є розробка веб-додатку з надання консультацій по догляду за тваринами, який дозволяє переглядати інформаційні матеріали, створювати консультаційні заявки, зберігати їх у браузері та визначати попередній рівень терміновості звернення за описом симптомів.

Для досягнення цієї мети потрібно виконати кілька практичних завдань. Насамперед необхідно визначити, для кого створюється веб-додаток і які функції він повинен мати. Далі потрібно продумати структуру сторінок, щоб

користувач міг легко переходити між головною сторінкою, каталогом тварин, порадами, формою консультації та списком заявок. Також потрібно реалізувати перевірку введених даних у формі, збереження заявок у локальному сховищі браузера та просту логіку оцінки терміновості звернення.

У межах роботи передбачається виконати такі завдання:

Для досягнення поставленої мети потрібно виконати такі завдання:

- а) проаналізувати предметну область догляду за домашніми тваринами;
- б) розглянути існуючі веб-ресурси з подібним призначенням;
- в) визначити основні вимоги до веб-додатку;
- г) спроектувати структуру сторінок та інтерфейс програмного продукту;
- г) реалізувати каталог тварин, сторінку порад і форму консультаційної заявки;
- д) забезпечити збереження створених заявок користувача;
- е) реалізувати просту оцінку рівня терміновості звернення;
- є) перевірити роботу основних функцій веб-додатку.

Об'єктом розробки є процес інформаційної підтримки власників домашніх тварин під час пошуку порад і створення консультаційних звернень.

Предметом розробки є веб-додаток з надання консультацій по догляду за тваринами, який містить інформаційні сторінки, каталог тварин, поради з догляду, форму заявки, список звернень користувача та механізм попередньої оцінки терміновості проблеми.

Веб-додаток повинен складатися з кількох основних сторінок. Головна сторінка має коротко пояснювати призначення сервісу. Сторінка «Про сервіс» повинна містити загальний опис можливостей додатку. У каталозі тварин мають бути подані основні категорії домашніх тварин. На сторінці порад користувач повинен мати змогу переглядати рекомендації та фільтрувати їх за категорією. Сторінка консультації повинна містити форму заявки, а сторінка заявок - список уже створених звернень. Додатково доцільно передбачити сторінки FAQ та контактів, щоб сайт виглядав завершеним.

До функціональних вимог веб-додатку належать перегляд інформаційних сторінок, відображення категорій тварин, перегляд і фільтрація порад, створення консультаційної заявки, перевірка обов'язкових полів, збереження заявки, перегляд списку заявок, зміна статусу звернення та його видалення.

Для збереження даних у навчальній версії додатку доцільно використати локальне сховище браузера. Такий варіант дозволяє зберігати заявки користувача без створення серверної частини та бази даних. Після оновлення сторінки користувач зможе знову побачити раніше створені звернення, що робить роботу з додатком більш зручною [10].

До нефункціональних вимог належать простота інтерфейсу, адаптивність для різних пристроїв, зрозуміла структура сторінок, швидке завантаження та можливість подальшого розширення. Оскільки робота виконується як навчальний проект, важливо також, щоб код мав логічну організацію: окремі сторінки, компоненти, файли з даними та допоміжні функції.

Для реалізації користувацького інтерфейсу можна використати React, оскільки ця бібліотека дає змогу будувати додаток із окремих компонентів. Наприклад, окремими компонентами можуть бути меню, картка тварини, картка поради, форма заявки та картка створеного звернення [3].

Окремої уваги потребує логіка оцінки терміновості звернення. Вона не повинна сприйматися як медична діагностика. Її задача - лише попередньо виділити звернення, у яких користувач вказав потенційно небезпечні симптоми. Наприклад, якщо в описі є слова на зразок «кров», «судоми», «отруєння», «травма», «не їсть», «блювання» або «втрата свідомості», система може позначити заявку як більш термінову.

Очікуваним результатом роботи є працездатний веб-додаток, який дозволяє користувачу переглядати поради щодо догляду за тваринами, створювати консультаційні заявки, зберігати їх у браузері та переглядати список звернень. Такий програмний продукт має продемонструвати базові навички аналізу предметної області, проєктування веб-додатку, створення інтерфейсу, роботи з даними та тестування основних функцій.

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Огляд робіт з аналогічними завданнями та реалізацією

Для розробки веб-додатку з надання консультацій по догляду за тваринами доцільно розглянути вже наявні рішення, які частково виконують схожі функції. До таких рішень можна віднести інформаційні сайти про домашніх тварин, веб-ресурси ветеринарних клінік, онлайн-довідники, форуми власників тварин та сервіси для консультацій зі спеціалістами. Кожен із цих типів ресурсів має свої особливості, тому їх аналіз допомагає зрозуміти, які функції варто реалізувати у власному веб-додатку.

Першу групу становлять інформаційні сайти про догляд за тваринами. На таких ресурсах зазвичай розміщуються статті про харчування, гігієну, поведінку, профілактику захворювань та умови утримання домашніх тварин. Користувач може знайти поради для різних категорій тварин: собак, котів, птахів, кроликів або гризунів. Такі сайти корисні тим, що містять багато матеріалів в одному місці. Наприклад, довідкові ресурси PetMD подають інформацію про здоров'я, симптоми, догляд і профілактику для різних видів домашніх тварин [16].

Однак інформаційні сайти здебільшого працюють як електронні довідники. Користувач може прочитати статтю, але не завжди має можливість описати власну ситуацію, створити консультаційне звернення або зберегти історію своїх заявок. Через це такі ресурси не повністю вирішують задачу організації консультаційного процесу.

Другу групу становлять сайти ветеринарних клінік. Їхнє основне призначення - надати користувачу інформацію про послуги клініки, спеціалістів, графік роботи, адресу, телефони та можливість запису на прийом. Часто на таких сайтах є окремі статті або короткі поради для власників тварин. Перевагою таких ресурсів є зв'язок із реальними спеціалістами. Якщо власник помічає у тварини небезпечні симптоми, він може швидко знайти контакти клініки та звернутися по допомогу.

Разом із цим сайти клінік зазвичай орієнтовані на конкретну установу. Їхня головна функція - запис на прийом або реклама послуг. Вони не завжди мають зручний каталог порад, фільтрацію за видами тварин або можливість зберігати власні звернення. Для користувача, який поки що хоче лише отримати базову інформацію або сформулювати проблему, такий формат може бути не зовсім зручним.

Третю групу становлять форуми та онлайн-спільноти власників тварин. На таких платформах користувачі обмінюються особистим досвідом, ставлять питання та отримують відповіді від інших учасників. Перевага форумів полягає у великій кількості реальних ситуацій. Людина може знайти схожий випадок і подивитися, як діяли інші власники тварин.

Недоліком форумів є те, що інформація на них не завжди перевірена. Поради можуть давати люди без ветеринарної освіти, тому такі рекомендації можуть бути неточними або навіть шкідливими. Крім того, обговорення на форумах часто мають хаотичну структуру, через що пошук потрібної відповіді займає багато часу.

Четверту групу становлять онлайн-сервіси консультацій. Вони найближчі до теми цієї роботи, оскільки дозволяють користувачу поставити питання, описати проблему та отримати відповідь від спеціаліста. Такі сервіси можуть мати форму заявки, чат, особистий кабінет або систему повідомлень. Їхньою перевагою є більша персоналізація порівняно зі звичайними статтями.

Проте частина таких сервісів потребує реєстрації, оплати або складного заповнення форми. Для навчального веб-додатку доцільно реалізувати простішу модель: користувач заповнює заявку, система зберігає її в браузері та визначає попередній рівень терміновості за симптомами. Такий підхід не замінює повноцінну консультацію, але показує основну логіку роботи консультаційної системи.

Також важливо врахувати професійні джерела з догляду за тваринами. Наприклад, рекомендації міжнародних ветеринарних організацій можуть використовуватися як орієнтир для підготовки інформаційних матеріалів про

профілактику, харчування та вакцинацію [14], [15]. Це дає змогу уникати випадкових порад і робити зміст веб-додатку більш обґрунтованим.

З технічної сторони аналогічні веб-ресурси зазвичай будуються на стандартних веб-технологіях. HTML відповідає за структуру сторінок, CSS - за зовнішній вигляд, JavaScript - за інтерактивність і обробку дій користувача [7], [8], [9]. У сучасних веб-додатках часто використовуються бібліотеки для побудови інтерфейсу, зокрема React, що дозволяє створювати систему з окремих компонентів [3].

Отже, огляд аналогічних рішень показує, що наявні ресурси частково вирішують задачу інформаційної підтримки власників тварин. Інформаційні сайти надають багато матеріалів, сайти клінік забезпечують зв'язок зі спеціалістами, форуми містять практичний досвід, а консультаційні сервіси дають можливість поставити питання. Проте не всі ці можливості зазвичай поєднані в одному простому веб-додатку. Саме тому у власній розробці доцільно об'єднати каталог порад, форму звернення, збереження заявок і попередню оцінку терміновості.

2.2 Переваги та труднощі у використанні оглянутих програм

Після розгляду аналогічних рішень можна визначити, що більшість із них справді допомагає власникам тварин знаходити потрібну інформацію. Найбільш помітною перевагою є доступність. Користувач може відкрити сайт із комп'ютера або телефона, переглянути поради, знайти контакти ветеринарної клініки чи прочитати досвід інших власників тварин. Для простих питань цього часто достатньо, особливо якщо користувач шукає загальні рекомендації щодо харчування, гігієни або умов утримання.

Ще однією перевагою є великий обсяг матеріалів. Інформаційні сайти та довідники містять багато статей про різні види тварин. На таких ресурсах можна знайти поради щодо собак, котів, птахів, гризунів, кроликів та інших домашніх улюбленців. Це зручно, оскільки користувач має можливість самостійно ознайомитися з темою та порівняти кілька рекомендацій.

Сайти ветеринарних клінік мають іншу сильну сторону - вони пов'язані з реальними спеціалістами. Якщо користувач бачить, що стан тварини викликає занепокоєння, він може знайти контакти клініки, дізнатися графік роботи або записатися на прийом. Для серйозних випадків це важливо, оскільки онлайн-інформація не може замінити огляд тварини ветеринаром.

Форуми та спільноти корисні тим, що містять багато практичних прикладів. У таких обговореннях власники тварин часто описують реальні ситуації, з якими вони зіткнулися. Іноді це допомагає краще зрозуміти проблему або побачити, що подібний випадок уже траплявся в інших людей. Проте такі поради потрібно сприймати обережно, бо вони не завжди є професійними.

Окремою перевагою сучасних веб-сервісів є інтерактивність. На відміну від звичайних статей, веб-додаток може обробляти введені користувачем дані, зберігати заявки, фільтрувати інформацію та змінювати вміст сторінки без повного перезавантаження. Такі можливості реалізуються за допомогою JavaScript та інших засобів веб-розробки [9].

Разом із перевагами в оглянутих ресурсах є і певні труднощі. Перша проблема - розрізненість інформації. Користувач часто змушений переходити між різними сайтами: на одному ресурсі є стаття про харчування, на іншому - інформація про симптоми, на третьому - контакти клініки. Це займає час і може заплутати, особливо якщо людина шукає відповідь швидко.

Друга проблема - надто загальний характер рекомендацій. Більшість статей описує типові ситуації, але не враховує конкретну тварину, її вік, поведінку, умови утримання або симптоми. Через це користувач не завжди розуміє, чи підходить знайдена порада саме для його випадку.

Третя складність пов'язана з достовірністю інформації. На форумах і в окремих блогах поради можуть давати люди без спеціальної підготовки. Навіть якщо вони діляться власним досвідом, це не означає, що їхній спосіб вирішення проблеми буде правильним для іншої тварини. Тому в роботі з такими джерелами потрібно бути обережним.

Четверта проблема - відсутність збереження звернень. На багатьох інформаційних сайтах користувач може лише прочитати матеріал. Якщо він хоче описати свою проблему, повернутися до неї пізніше або переглянути попередні звернення, такої можливості часто немає. У розроблюваному веб-додатку цю задачу планується вирішити через збереження заявок у локальному сховищі браузера [10], [11].

Також можна виділити проблему складного або перевантаженого інтерфейсу. Деякі сайти містять багато реклами, зайвих блоків, спливаючих вікон або складне меню. Для користувача, який хоче швидко знайти пораду чи створити звернення, це створює додаткові незручності. Тому у власному веб-додатку важливо зробити структуру простою: головна сторінка, каталог тварин, поради, консультаційна форма та список заявок.

Порівняння оглянутих рішень показує, що найбільш корисним для користувача буде поєднання кількох можливостей в одному сервісі. Інформаційна частина потрібна для перегляду базових порад. Форма заявки потрібна для опису конкретної проблеми. Збереження звернень дозволяє користувачу повернутися до них пізніше. А попередня оцінка терміновості допомагає звернути увагу на потенційно небезпечні симптоми.

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1 Основні поняття веб-додатків для надання консультаційних послуг

Веб-додаток можна розглядати як програму, з якою користувач працює через браузер. На відміну від звичайної інформаційної сторінки, такий додаток не лише показує текст або зображення, а й реагує на дії користувача: відкриває потрібні розділи, обробляє введені дані, перевіряє форми, зберігає інформацію та оновлює окремі елементи сторінки без повного перезавантаження. Для створення таких рішень зазвичай використовують HTML, CSS і JavaScript [7], [8], [9].

HTML відповідає за структуру сторінки. За його допомогою створюються заголовки, блоки тексту, кнопки, списки, форми та інші елементи інтерфейсу. CSS використовується для оформлення зовнішнього вигляду: кольорів, шрифтів, відступів, розташування блоків і адаптації сторінки до різних екранів. JavaScript додає інтерактивність, тобто дозволяє перевірити введені користувачем дані, змінювати вміст сторінки, працювати з подіями та зберігати інформацію у браузері.

У межах цієї роботи консультаційний веб-додаток розглядається як невелика інформаційна система, що поєднує довідкові матеріали та форму звернення. Користувач може переглянути поради щодо догляду за тваринами, обрати потрібну категорію, заповнити заявку та описати проблему. Після цього система зберігає звернення і відображає його у списку заявок.

Для такого додатку важливо одразу визначити його межі. Він не ставить діагноз і не замінює консультацію ветеринарного лікаря. Його задача - допомогти користувачу зорієнтуватися в базових питаннях догляду, зібрати опис проблеми в одному місці та попередньо виділити звернення, які можуть потребувати швидшої реакції. Тому в інтерфейсі потрібно передбачити попередження, що рекомендації мають інформаційний характер [12].

Для реалізації інтерфейсу доцільно використати React. Ця бібліотека дозволяє будувати сайт із невеликих окремих частин - компонентів.

Наприклад, окремими компонентами можуть бути меню сайту, картка тварини, картка поради, форма заявки або елемент списку звернень. Такий підхід зручний для навчального проєкту, оскільки код легше читати, змінювати та доповнювати новими функціями [3], [4].

3.2 Особливості інформаційної підтримки користувачів у сфері догляду за тваринами

Інформаційна підтримка власників тварин має бути простою і практичною. Користувачі можуть мати різний досвід: хтось давно утримує домашніх улюбленців, а хтось тільки вперше взяв тварину додому. Тому матеріали у веб-додатку не повинні бути перевантажені складними термінами. Краще подавати поради коротко, зрозуміло і з поділом за темами.

Важливо враховувати, що догляд залежить від виду тварини. Для собак важливими є прогулянки, фізична активність, соціалізація та профілактичні огляди. Для котів більше значення мають безпечний простір, чистий лоток, стабільний режим і контроль стресу. Для птахів, кроликів і гризунів особливо важливі чистота клітки, правильне харчування, температура та безпечні умови утримання. Через це у веб-додатку доцільно розділити інформацію за категоріями тварин.

Матеріали також потрібно структурувати за змістом. Якщо всі рекомендації розмістити одним великим текстом, користувачу буде складно знайти потрібну відповідь. Зручніше подати окремі блоки: харчування, гігієна, поведінка, профілактика, симптоми та умови утримання. Такий поділ робить сервіс зрозумілішим і скорочує час пошуку інформації.

Окрема частина інформаційної підтримки пов'язана з небезпечними симптомами. Наприклад, кровотеча, судоми, отруєння, втрата свідомості, сильна травма, тривале блювання або утруднене дихання можуть свідчити про ситуацію, коли не варто обмежуватися онлайн-порадою. У таких випадках користувачу потрібно показати зрозуміле попередження про необхідність звернення до ветеринарного спеціаліста.

У розроблюваному веб-додатку попередня оцінка терміновості виконується простим способом - за ключовими словами в описі симптомів. Якщо користувач вводить слова, які можуть вказувати на небезпечний стан, система позначає заявку як середню або високу за рівнем терміновості. Така перевірка не є медичною діагностикою, але вона корисна як додатковий орієнтир для користувача.

Ще одна важлива функція - збереження звернень. Користувач може створити заявку, а потім повернутися до неї пізніше, переглянути опис проблеми або змінити статус. Для навчальної версії достатньо використати localStorage, тобто локальне сховище браузера. Воно дозволяє зберігати дані на стороні користувача і не втрачати їх після оновлення сторінки [10], [11].

3.3 Проєктування архітектури програмного забезпечення

Архітектура програмного забезпечення визначає загальну будову веб-додатку, його основні частини та зв'язки між ними. Для розроблюваного веб-додатку з надання консультацій по догляду за тваринами доцільно використати клієнтську архітектуру, за якої основна логіка роботи виконується у браузері користувача. Такий підхід є достатнім для навчального проєкту, оскільки дозволяє реалізувати перегляд інформації, створення заявок, збереження звернень і базову оцінку терміновості без серверної частини.

Основними елементами архітектури є користувацький інтерфейс, сторінки вебдодатку, повторно використовувані компоненти, файли з даними, допоміжні функції та локальне сховище браузера. Користувач взаємодіє із системою через веб-інтерфейс. Через меню навігації він переходить між сторінками, переглядає поради, заповнює форму консультаційної заявки та працює зі списком створених звернень. Загальна архітектура веб-додатку подана на рисунку 3.1.

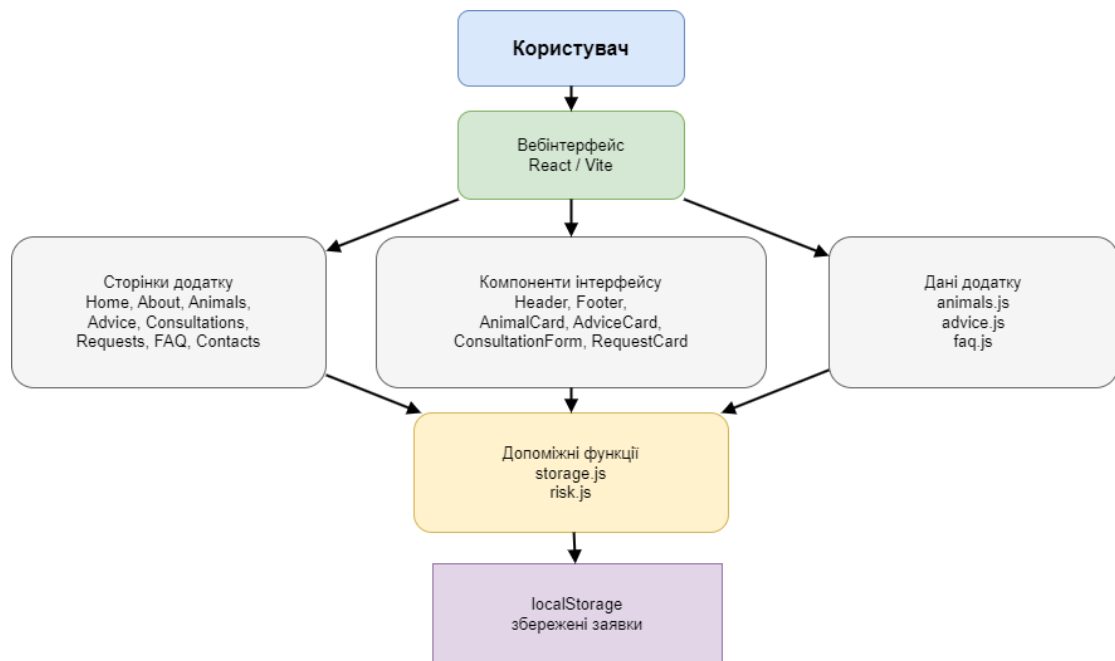


Рисунок 3.1 - Загальна архітектура веб-додатку

Сторінки веб-додатку формують його основну логічну структуру. До них належать головна сторінка, сторінка опису сервісу, каталог тварин, сторінка порад, сторінка консультаційної заявки, сторінка перегляду заявок, сторінка частих питань і сторінка контактів. Кожна сторінка має окреме призначення, що дозволяє зробити інтерфейс більш зрозумілим для користувача. Структура сторінок вебдодатку подана на рисунку 3.2.

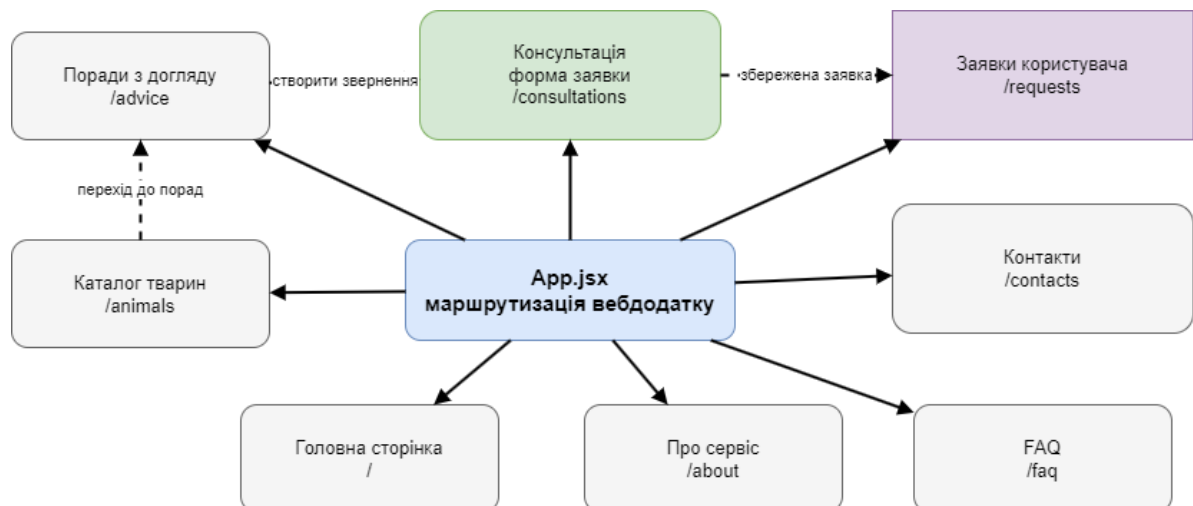


Рисунок 3.2 - Структура сторінок веб-додатку

Компоненти використовуються для побудови повторюваних частин інтерфейсу. Наприклад, компонент заголовка відповідає за навігацію, компонент картки тварини - за виведення інформації про категорію тварини, компонент картки поради - за відображення рекомендації, компонент форми - за створення консультаційної заявки, а компонент картки заявки - за показ одного звернення у списку. Такий підхід робить структуру коду зрозумілішою та спрощує подальше розширення веб-додатку.

Файли з даними використовуються для зберігання статичної інформації: списку тварин, порад з догляду та частих питань. Допоміжні функції відповідають за роботу із заявками, збереженням даних і визначенням рівня терміновості звернення. Заявки користувача зберігаються у локальному сховищі браузера, тому вони залишаються доступними після оновлення сторінки.

Таким чином, запропонована архітектура дозволяє розділити веб-додаток на окремі логічні частини: сторінки, компоненти, дані та службові функції. Це спрощує розробку, тестування й подальше вдосконалення програмного продукту.

3.4 Графічне представлення архітектури

Графічне представлення архітектури використовується для наочного пояснення структури веб-додатку та логіки роботи його основних функцій. У межах даної роботи доцільно подати кілька графічних матеріалів: схему загальної архітектури, структуру сторінок, структуру файлів проєкту, блок-схему створення консультаційної заявки, схему визначення рівня терміновості та діаграму варіантів використання.

Одним із головних процесів у веб-додатку є створення консультаційної заявки. Користувач відкриває сторінку консультації, заповнює форму, після чого система перевіряє введені дані. Якщо форма заповнена неправильно, користувач отримує повідомлення про помилку. Якщо дані коректні, система аналізує симптоми, визначає рівень терміновості, створює об'єкт заявки та

зберігає його у локальному сховищі браузера. Алгоритм поданий в Додатку А.

Окремо розглядається процес попереднього визначення рівня терміновості звернення. Для цього система аналізує текст симптомів, введений користувачем. Якщо у тексті не знайдено небезпечних ключових слів, заявка отримує низький рівень терміновості. Якщо знайдено один збіг, встановлюється середній рівень. Якщо знайдено два або більше збігів, система визначає високий рівень терміновості та показує користувачу попередження про необхідність звернення до ветеринарного спеціаліста.



Рисунок 3.3 - Схема визначення рівня терміновості звернення

Для опису взаємодії користувача із системою використовується діаграма варіантів використання. Основним актором є користувач веб-додатку. Він може переглядати інформаційні сторінки, відкривати каталог тварин, читати поради з догляду, фільтрувати матеріали, створювати консультаційні заявки, переглядати список звернень, змінювати статус заявки та видаляти непотрібні записи.

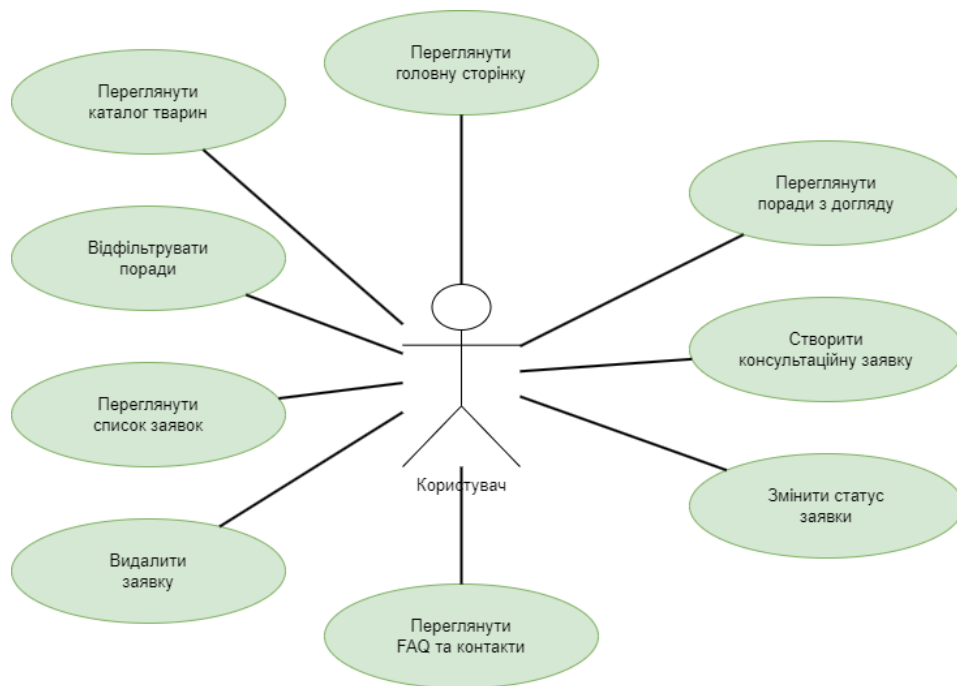


Рисунок 3.4 - Діаграма варіантів використання вебдодатку

Графічне представлення архітектури дозволяє показати не лише загальну структуру програмного продукту, а й основні процеси його роботи. Завдяки схемам легше простежити, як користувач взаємодіє з веб-додатком, як формується консультаційна заявка, як визначається рівень терміновості та як окремі частини системи пов'язані між собою.

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1 Опис розробки веб-додатку з надання консультацій по догляду за тваринами

Для створення веб-сайту використано HTML5, CSS3, JavaScript, бібліотеки Google Material Icons та IonIcons. Організацію стилів реалізовано на основі CSS Grid та Flexbox. Для динамічної поведінки застосовано чистий JavaScript без використання додаткових фреймворків, а для збереження даних у навчальній версії додатку використано локальне сховище браузера (localStorage).

Розглянемо більш детально створення веб-застосунку.

На першому етапі практичної реалізації було визначено загальну структуру веб-додатку. Для зручності користувача сайт поділено на кілька сторінок, кожна з яких виконує окрему функцію. Такий підхід дозволяє уникнути перевантаження інтерфейсу та забезпечує логічну навігацію між розділами. До складу веб-додатку входять такі основні сторінки:

```
function App() {
  return (
    <div className="app">
      <Header />

      <main className="main-content">
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/about" element={<About />} />
          <Route path="/animals" element={<Animals />} />
          <Route path="/advice" element={<Advice />} />
          <Route path="/consultations" element={<Consultations />} />
          <Route path="/requests" element={<Requests />} />
          <Route path="/faq" element={<FAQ />} />
          <Route path="/contacts" element={<Contacts />} />
        </Routes>
      </main>

      <Footer />
    </div>
  );
}
```

Рисунок 4.1 – Основні сторінки програми

Головна сторінка є початковою точкою взаємодії користувача із системою. На ній розміщено назву сервісу, короткий опис його призначення та кнопки швидкого переходу до основних функцій. Основний зміст цієї сторінки полягає у тому, щоб одразу пояснити користувачу, для чого призначений веб-додаток і які можливості він надає.

Сторінка «Про сервіс» містить опис ідеї веб-додатку. На ній зазначено, що сервіс призначений для інформаційної підтримки власників домашніх тварин. Також на цій сторінці доцільно розмістити попередження про те, що веб-додаток не замінює консультацію ветеринарного лікаря, а лише допомагає користувачу отримати базову інформацію та оформити звернення.

Каталог тварин містить перелік основних категорій домашніх тварин. Для кожної категорії подано короткий опис, базові потреби та можливі питання догляду. Наприклад, для собак зазначено необхідність прогулянок, фізичної активності, правильного харчування та профілактики. Для котів акцент зроблено на безпечному просторі, чистоті лотка, режимі годування та контролі стресу. Для птахів, кроликів і гризунів описано умови утримання, чистоту клітки та особливості раціону.

Реалізація системи роутингу та навігації .Логіка переміщення користувача по сторінках додатка реалізована через декоратори @app.get у модулі main.py. Система підтримує багаторівневу структуру URL-адрес.

Сторінка порад з догляду призначена для перегляду інформаційних матеріалів. Кожна порада має назву, категорію тварини, короткий опис і текст рекомендації. Для підвищення зручності користувача реалізовано можливість фільтрації порад за категорією. Це дозволяє не переглядати всі матеріали одразу, а швидко знайти інформацію, яка стосується конкретного виду тварини.

Сторінка консультаційної заявки містить форму, за допомогою якої користувач може описати свою проблему. У формі передбачено поля для введення імені власника, контактних даних, виду тварини, імені тварини, віку,

опису проблеми та симптомів. Також користувач може вказати бажаний спосіб зв'язку.

Перед збереженням заявки система перевіряє правильність заповнення форми. обов'язковими є поля з іменем користувача, контактними даними, видом тварини, описом проблеми та симптомами. Якщо користувач не заповнив необхідні поля або ввів занадто короткий опис, система повинна повідомити про помилку та не створювати заявку до виправлення даних.

Статус нової заявки встановлюється як «Нова». Надалі користувач може змінити статус на «В обробці» або «Завершена». Це дозволяє імітувати просту систему керування консультаційними зверненнями.

Окремим елементом практичної реалізації є алгоритм попередньої оцінки терміновості звернення. Він аналізує текст симптомів, який вводить користувач. Якщо в описі є слова, що можуть свідчити про небезпечний стан тварини, заявка отримує підвищений рівень терміновості. До таких слів можуть належати: «кров», «судоми», «отруєння», «травма», «блювання», «не їсть», «втрата свідомості», «не дихає», «не встає». У спрощеному вигляді логіка роботи алгоритму така:

```

const dangerousWords = [
  'кров',
  'судоми',
  'отруєння',
  'травма',
  'блювання',
  'не їсть',
  'втрата свідомості',
  'не дихає',
  'не встає',
];

export function detectRisk(symptoms) {
  const normalizedText = symptoms.toLowerCase();

  const matchedWords = dangerousWords.filter((word) =>
    normalizedText.includes(word),
  );

  if (matchedWords.length >= 2) {
    return {
      level: 'Високий',
      matchedWords,
      warning:
        'Виявлено небезпечні симптоми. Рекомендується терміново звернутися до ветеринарного спеціаліста.',
    };
  }

  if (matchedWords.length === 1) {
    return {
      level: 'Середній',
      matchedWords,
      warning: '',
    };
  }

  return {
    level: 'Низький',
    matchedWords: [],
    warning: '',
  };
}

```

Рисунок 4.2 – Алгоритм вищначення стану тварини

Цей алгоритм не виконує медичної діагностики. Він лише допомагає користувачу звернути увагу на потенційно небезпечну ситуацію. У разі високого рівня терміновості система повинна показувати попередження про необхідність звернення до ветеринарного спеціаліста.

Для збереження заявок використовується локальне сховище браузера. Після створення заявки вона записується у localStorage. Під час відкриття сторінки заявок система зчитує збережені дані та відображає їх у вигляді списку. Завдяки цьому заявки не зникають після оновлення сторінки.

Сторінка перегляду заявок відображає всі створені користувачем звернення. Для кожної заявки показується дата, ім'я власника, вид тварини, опис проблеми, рівень терміновості та статус. Також передбачено кнопки для

зміни статусу та видалення заявки. Видалення заявки оновлює у локальному сховищі.

```
// src/utils/storage.js
const REQUESTS_KEY = 'petcare_consult_requests';

export function loadRequests() {
  try {
    return JSON.parse(localStorage.getItem(REQUESTS_KEY)) || [];
  } catch {
    return [];
  }
}

export function saveRequests(requests) {
  localStorage.setItem(REQUESTS_KEY, JSON.stringify(requests));
}

export function addRequest(request) {
  const requests = loadRequests();
  const updatedRequests = [request, ...requests];

  saveRequests(updatedRequests);
  return updatedRequests;
}

export function updateRequestStatus(id, status) {
  const updatedRequests = loadRequests().map((request) =>
    request.id === id ? { ...request, status } : request,
  );

  saveRequests(updatedRequests);
  return updatedRequests;
}

export function deleteRequest(id) {
  const updatedRequests = loadRequests().filter(
    (request) => request.id !== id,
  );

  saveRequests(updatedRequests);
  return updatedRequests;
}
```

Рисунок 4.3– Сбереження заявок у localStorage

Для реалізації інтерфейсу використовується компонентний підхід.

Відображення HTML-шаблонів (Jinja2) Для формування інтерфейсу використано шаблонізатор Jinja2. Дані, отримані з бази даних (заголовки, текст поради), динамічно вставляються у відповідні HTML-теги, що дозволяє створювати єдиний макет сторінки для різних категорій консультацій.

Основні елементи веб-додатку винесено в окремі компоненти. Наприклад, компонент заголовка відповідає за навігаційне меню, компонент картки тварини - за відображення категорій, компонент картки поради - за виведення рекомендацій, компонент форми - за створення заявки, а компонент картки заявки - за відображення одного звернення.

4.2 Інструкція роботи з веб-додатком

Для початку роботи користувач відкриває веб-додаток у браузері. Після завантаження головної сторінки він бачить короткий опис сервісу та навігаційне меню. За допомогою меню можна перейти до будь-якого розділу веб-додатку.

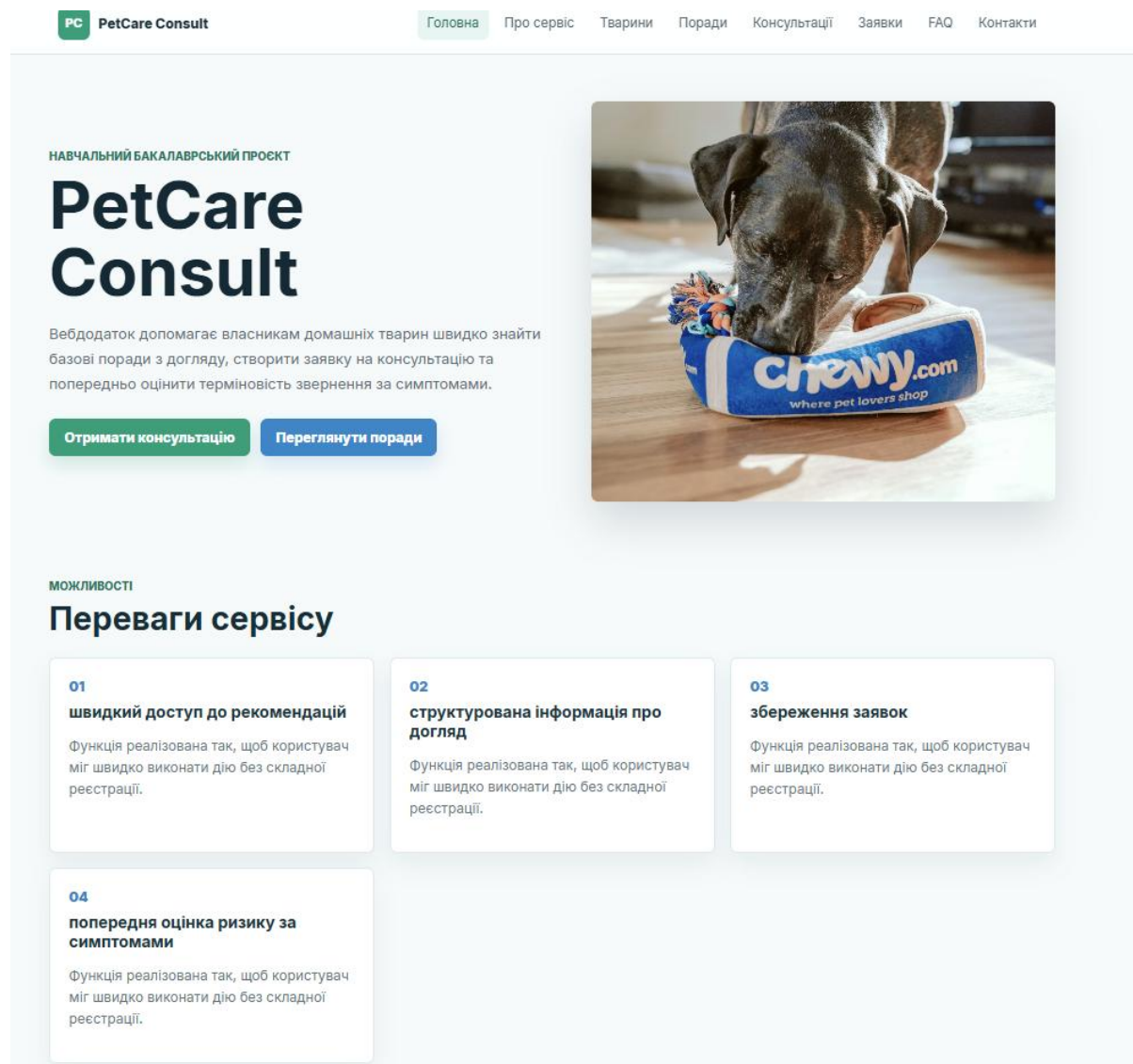


Рисунок 4.4 – Головна сторінка

Щоб дізнатися для кого сервіс, користувач переходить на інформаційне вікно про сервіс.

ПРО СЕРВІС

Систематизована допомога для власників тварин

Багато власників домашніх тварин шукають поради у випадкових джерелах і не завжди розуміють, які симптоми є справді небезпечними. PetCare Consult впорядковує базову інформацію та дає простий механізм створення консультаційної заявки.

Призначення вебдодатку

Додаток створений для перегляду матеріалів про догляд, вибору категорії тварини, подання заявки на консультацію та збереження історії звернень у браузері користувача.

Цільова аудиторія

- власники домашніх тварин;
- люди, які планують завести тварину;
- волонтери або працівники притулків.

Важливе попередження


PetCare Consult не замінює ветеринарного лікаря. Якщо стан тварини різко погіршується або є небезпечні симптоми, потрібно звернутися до ветеринарної клініки.

Рисунок 4.5 – Сторінка про сервіс

Щоб переглянути інформацію про тварин, користувач переходить до сторінки «Каталог тварин».

Категорії домашніх тварин

Оберіть категорію, щоб ознайомитися з базовими потребами, типовими проблемами та перейти до відповідних порад з догляду.




Собаки

Активні соціальні тварини, яким потрібні прогулянки, навчання, стабільний режим і регулярна профілактика.

Базові потреби:
збалансоване харчування, щоденні прогулянки, вакцинація, гігієна шерсті

Типові проблеми:
стрес без активності, паразити, алергії, травми під час прогулянок

[Переглянути поради](#)




Коти

Самостійні домашні улюбленці, для яких важливі безпечний простір, чистий лоток і контроль харчування.

Базові потреби:
чиста вода, лоток, дряпка, місце для відпочинку

Типові проблеми:
стрес від змін, проблеми з сечовиділенням, зайва вага, ковтуни шерсті

[Переглянути поради](#)



Птахи

Чутливі до умов утримання тварини, яким потрібні простора клітка, правильне світло та спокій.

Базові потреби:
простора клітка, чисті поїлки, безпечні жердинки, різноманітний корм

Типові проблеми:
протяги, стрес від шуму, нестача активності, травми крил

[Переглянути поради](#)

Рисунок 4.6 – Головна сторінка

Для перегляду рекомендацій потрібно відкрити сторінку «Поради». На цій сторінці розміщено список інформаційних матеріалів.

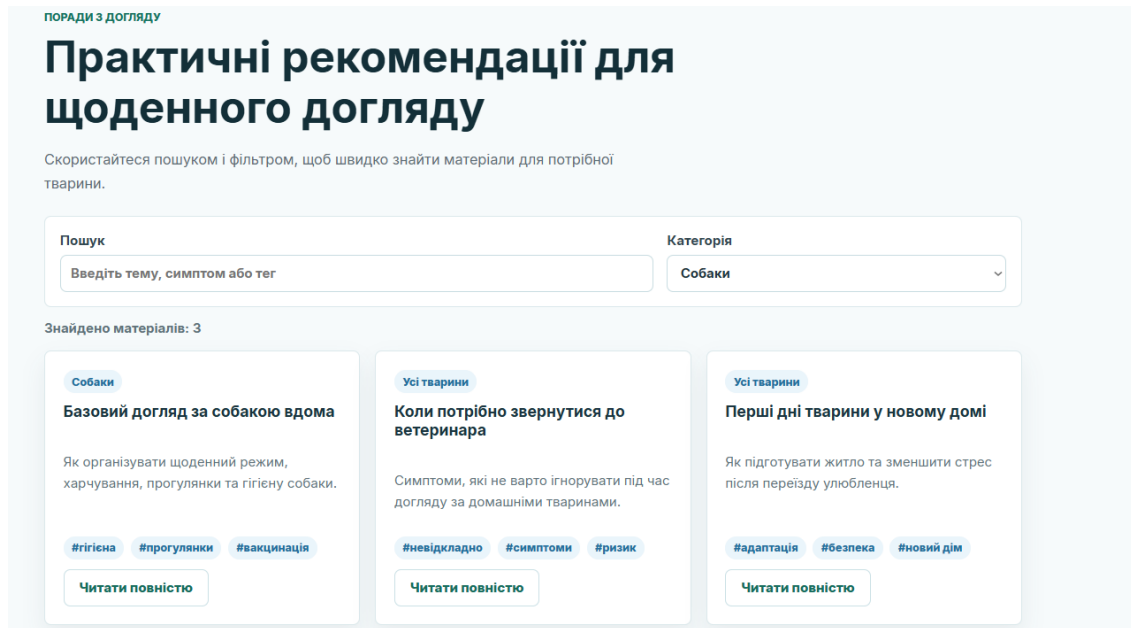


Рисунок 4.7 – Каталог порад

Для створення консультаційної заявки користувач переходить на сторінку «Консультація». На сторінці відображається форма, яку необхідно заповнити.

Рисунок 4.8 – Створення консультації

Якщо обов'язкові поля не заповнені або опис проблеми занадто короткий, система повідомляє користувача про помилку. Після створення

заявки користувач може перейти на сторінку «Заявки». На цій сторінці відображається список усіх створених звернень.

Список створених звернень

Тут відображаються всі заявки, збережені у LocalStorage. Можна змінити статус звернення або видалити заявку.

#REQ-590720 Низький

Бодя - Собаки

Дата 20.05.2026, 11:26:30	Власник Олена
Контакт teodaniil2@gmail. com	Статус Нова

Дуже щаслива собка, я їй вчора купила нову іграшку

Змінити статус

Нова

Рисунок 4.9 – Список створених звернень

Рівень терміновості може бути низьким, середнім або високим. Низький рівень означає, що у введеному описі не виявлено небезпечних ключових слів. Середній рівень означає, що є один потенційно небезпечний симптом.

#REQ-186727 Високий

Тварина без імені - Собаки

Дата 20.05.2026, 11:36:26	Власник Олена
Контакт teodaniil2@gmail. com	Статус Нова

Не їсть не спить, все дуже погано

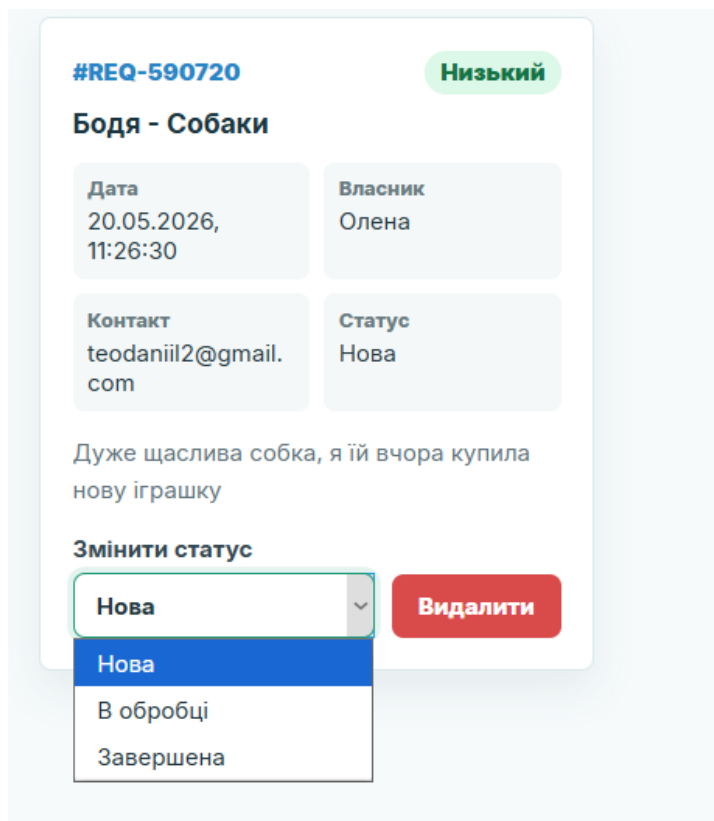
Опис містить небезпечні симптоми.
Рекомендується терміново
звернутися до ветеринарного
лікаря. Онлайн-консультація не
замінює невідкладну допомогу.

Змінити статус

Нова

Рисунок 4.10 – приклад високого пріоритету

На сторінці заявок користувач може змінити статус звернення. Для цього потрібно вибрати відповідний статус: «Нова», «В обробці» або «Завершена». Також користувач може видалити заявку, якщо вона більше не потрібна. Після видалення звернення воно зникає зі списку та видаляється з локального сховища.



The screenshot shows a request card with the following details:

- #REQ-590720** (ID)
- Низький** (Priority)
- Бодя - Собаки** (Subject)
- Дата:** 20.05.2026, 11:26:30
- Власник:** Олена
- Контакт:** teodanil2@gmail.com
- Статус:** Нова

The card also contains a text description: "Дуже щаслива собка, я їй вчора купила нову іграшку".

Below the description is a section titled "Змінити статус" (Change status) with a dropdown menu currently set to "Нова" (New). The dropdown menu is open, showing the following options: "Нова", "В обробці" (In progress), and "Завершена" (Completed). To the right of the dropdown is a red button labeled "Видалити" (Delete).

Рисунок 4.11 – Змінення статусу звернення

Сторінка «FAQ» містить відповіді на поширені питання. Тут користувач може дізнатися, чи замінює веб-додаток ветеринарного лікаря, які тварини підтримуються, як створити заявку та що робити у разі небезпечних симптомів.

Часті питання

Короткі відповіді на питання щодо можливостей, обмежень і використання сервісу.

Чи замінює сайт ветеринара?	-
Ні. Сайт надає базові інформаційні рекомендації та допомагає оцінити терміновість звернення, але не замінює огляд ветеринарного лікаря.	
Які тварини підтримуються?	+
Як швидко можна отримати консультацію?	+
Чи зберігаються заявки?	+
Що робити при небезпечних симптомах?	+
Чи можна використовувати сервіс для профілактики?	+
Які дані потрібно вказати у заявці?	+
Чи можна видалити заявку?	+
Чи можна змінити статус заявки?	+

Рисунок 4.12 – Сторінка FAQ

Сторінка «Контакти» містить умовну контактну інформацію сервісу та форму зворотного зв'язку. Вона призначена для завершення структури вебдодатку та імітації повноцінного інформаційного сервісу.

КОНТАКТИ

Зв'язок із PetCare Consult

Умовні контактні дані навчального сервісу та проста форма зворотного зв'язку.

PetCare Consult Email: support@petcare-consult.local Телефон: +380 44 000 00 00 Адреса: м. Київ, вул. Турботи про тварин, 12 Дані наведені як приклад для бакалаврського навчального проекту.	Ім'я <input type="text" value="Ваше ім'я"/> Email <input type="text" value="email@example.com"/> Повідомлення <input type="text" value="Ваше повідомлення"/> <input type="submit" value="Надіслати"/>
--	--

Рисунок 4.13 – Сторінка контакти

Під час роботи з веб-додатком користувачу слід враховувати, що сервіс має інформаційний характер. Він може допомогти зорієнтуватися у базових питаннях догляду, однак не встановлює діагноз і не замінює ветеринарну допомогу. У випадках гострих симптомів, травм, отруєння, кровотечі, судом або різкого погіршення стану тварини необхідно звернутися до ветеринарного лікаря.

Практична реалізація веб-додатку забезпечує виконання поставлених у роботі задач. Система має зрозумілу структуру, містить інформаційні сторінки, дозволяє створювати та зберігати консультаційні заявки.

ВИСНОВОК

У ході виконання кваліфікаційної роботи бакалавра було спроектовано та практично реалізовано веб-додаток з надання консультацій по догляду за домашніми тваринами, який забезпечує ефективну інформаційну підтримку власників тварин та автоматизацію процесу створення й менеджменту первинних консультаційних звернень.

У першому розділі було проведено детальний аналіз предметної області інформаційної підтримки власників тварин. Визначено основні проблеми, з якими стикаються користувачі під час пошуку порад в Інтернеті, зокрема хаотичність, недостовірність та розрізненість даних. На основі аналізу було сформовано функціональні та нефункціональні вимоги до веб-додатку, обґрунтовано вибір компонентного підходу і визначено структуру майбутнього сервісу.

У другому розділі було виконано розширений інформаційний огляд існуючих аналогів, включаючи довідкові ресурси (PetMD), офіційні платформи ветеринарних клінік, тематичні форуми та міжнародні стандарти ветеринарних асоціацій (AVMA, WSAVA). Виявлено їхні переваги та системні недоліки, такі як перевантаженість інтерфейсів та відсутність можливості локального збереження звернень. Також було обґрунтовано вибір технологічного стеку розробки, що базується на мові JavaScript, бібліотеці React та інструментарії Vite.

У третьому розділі описано теоретичні основи проєктування клієнтської архітектури програмного забезпечення. Розроблено та графічно представлено загальну архітектуру системи, структуру сторінок, діаграму варіантів використання, а також алгоритми взаємодії компонентів із локальним сховищем браузера. Особливу увагу приділено теоретичному обґрунтуванню логіки аналізу симптомів за ключовими словами для попереднього виявлення критичних станів тварин.

У четвертому розділі наведено детальний опис практичної реалізації веб-додатку та розроблено інструкцію користувача. Для створення веб-сайту

використано HTML5, CSS3, JavaScript, бібліотеки Google Material Icons та IonIcons. Організацію стилів реалізовано на основі CSS Grid та Flexbox. Для динамічної поведінки застосовано чистий JavaScript без використання додаткових фреймворків, а для збереження даних у навчальній версії додатку використано локальне сховище браузера (localStorage).

У четвертому розділі наведено детальний опис практичної реалізації веб-додатку та розроблено інструкцію користувача. Створено інтерактивні модулі каталогу тварин, системи фільтрації порад, форми відправки заявок із валідацією полів та сторінки керування створеними зверненнями. Реалізовано стійке збереження даних на стороні клієнта за допомогою технології localStorage та впроваджено працездатний алгоритм градації терміновості заявок (низький, середній, високий пріоритети) з функцією динамічного виведення попереджень про необхідність невідкладної ветеринарної допомоги.

Таким чином, поставлену мету кваліфікаційної роботи повністю досягнуто. Розроблений веб-додаток продемонстрував повну відповідність вимогам щодо адаптивності, простоти інтерфейсу та швидкодії, підтвердив свою практичну цінність і може слугувати надійною базою для подальшого розгортання повноцінного серверного сервісу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні рекомендації до виконання кваліфікаційної роботи для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» ступеня бакалавра / О. В. Ольховська, О. О. Черненко. Полтава : ПУЕТ, 2025. 58 с.
2. React Documentation. React Reference Overview. URL: <https://react.dev/reference/react>
3. React Documentation. Built-in React Hooks. URL: <https://react.dev/reference/react/hooks>
4. Vite Documentation. Getting Started. URL: <https://vite.dev/guide/>
5. Vite Documentation. Why Vite. URL: <https://vite.dev/guide/why>
6. MDN Web Docs. HTML: HyperText Markup Language. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML>
7. MDN Web Docs. CSS: Cascading Style Sheets. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS>
8. MDN Web Docs. JavaScript. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
9. MDN Web Docs. Window: localStorage property. URL: <https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>
10. MDN Web Docs. Web Storage API. URL: https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API
11. American Veterinary Medical Association. Responsible Pet Ownership. URL: <https://www.avma.org/resources-tools/pet-owners/responsible-pet-ownership>
12. American Veterinary Medical Association. Pet Care. URL: <https://www.avma.org/resources-tools/pet-owners/petcare>
13. World Small Animal Veterinary Association. Global Nutrition Guidelines. URL: <https://wsava.org/global-guidelines/global-nutrition-guidelines/>
14. World Small Animal Veterinary Association. Vaccination Guidelines. URL: <https://wsava.org/global-guidelines/vaccination-guidelines/>
15. PetMD. Pet Health Center. URL: <https://www.petmd.com/>

16. The Spruce Pets. Pet Care. URL: <https://www.thesprucepets.com/pet-care-4162114>
17. Nielsen J. Usability Engineering. San Francisco: Morgan Kaufmann, 1994. 362 p.
18. Krug S. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. 3rd ed. Berkeley : New Riders, 2014. 216 p.
19. Duckett J. HTML and CSS: Design and Build Websites. Indianapolis : Wiley, 2011. 490 p.
20. Лутц М. Вивчення Python / М. Лутц. — 5-те вид. — К. : Діалектика, 2020. — 1168 с.
21. Мартін Р. Чистий код / Р. Мартін. — Х. : Фабула, 2021. — 448 с.
22. Фаулер М. Архітектура корпоративних програмних додатків / М. Фаулер. — К. : Вільямс, 2018. — 544 с.
23. Python 3.12 documentation [Electronic resource]. — URL: <https://docs.python.org/3/>
24. PyCharm Documentation. Professional developer tools from JetBrains [Electronic resource]. — URL: <https://www.jetbrains.com/pycharm/documentation/>
25. Flask Documentation [Electronic resource]. — URL: <https://flask.palletsprojects.com/>
26. PostgreSQL Documentation [Electronic resource]. — URL: <https://www.postgresql.org/docs/>
27. MDN Web Docs (HTML, CSS, JavaScript) [Electronic resource]. — URL: <https://developer.mozilla.org/>

ДОДАТКИ

Додаток А

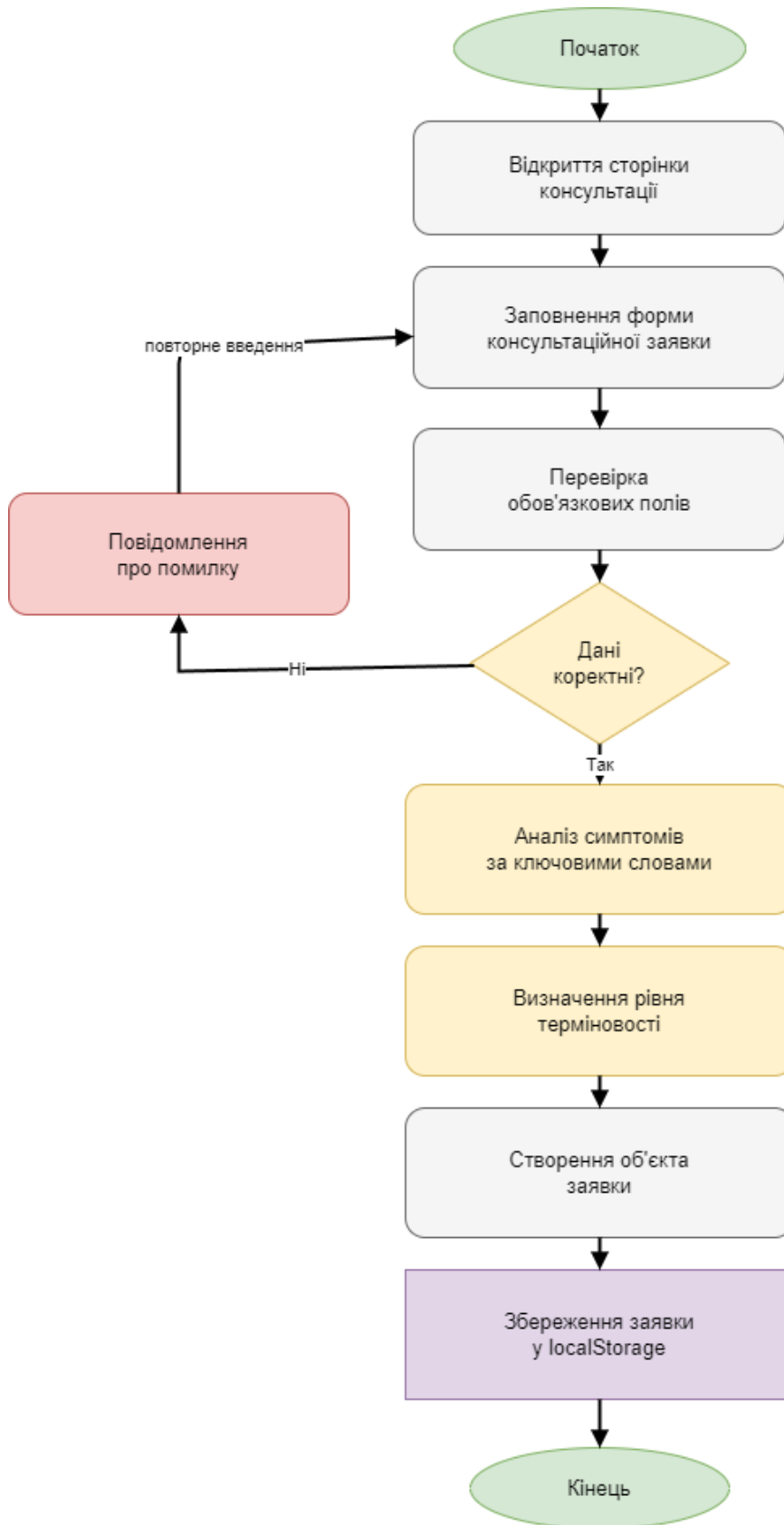


Рисунок А.1 – Блок схема створення консультаційної заявки

Додаток Б

PC PetCare Consult

Головна

Про сервіс

Тварини

Поради

Консультації

Заявки

FAQ

Контакти

НАВЧАЛЬНИЙ БАКАЛАВРСЬКИЙ ПРОЄКТ

PetCare Consult

Вебдодаток допомагає власникам домашніх тварин швидко знайти базові поради з догляду, створити заявку на консультацію та попередньо оцінити терміновість звернення за симптомами.

Отримати консультацію

Переглянути поради



МОЖЛИВОСТІ

Переваги сервісу

01

швидкий доступ до рекомендацій

Функція реалізована так, щоб користувач міг швидко виконати дію без складної реєстрації.

02

структурована інформація про догляд

Функція реалізована так, щоб користувач міг швидко виконати дію без складної реєстрації.

03

збереження заявок

Функція реалізована так, щоб користувач міг швидко виконати дію без складної реєстрації.

04

попередня оцінка ризику за симптомами

Функція реалізована так, щоб користувач міг швидко виконати дію без складної реєстрації.

Рисунок Б.1 – Головна сторінка

Створення заявки на консультацію

Заповніть форму, опишіть проблему та симптоми. Заявка буде збережена у LocalStorage, після чого її можна переглянути, змінити статус або видалити на сторінці заявок.

1

Заповнення форми

Користувач вводить контактні дані, інформацію про тварину та опис проблеми.

2

Оцінка симптомів

Система шукає небезпечні слова та визначає низький, середній або високий ризик.

3

Збереження заявки

Дані залишаються у браузері й доступні після оновлення сторінки.

Ім'я власника *

Наприклад, Олена

Email або телефон *

email@example.com

Вид тварини *

Оберіть вид

Ім'я тварини

Наприклад, Бім

Вік тварини

2 роки

Бажаний спосіб зв'язку

Email

Опис проблеми *

Опишіть ситуацію, що вас турбує

Симптоми *

Наприклад: млявість, блювання, не їсть

Попередня оцінка ризику: **Низький**

Створити заявку

Рисунок Б.2 – Створення заявки

Введіть тему, симптом або тег Усі

Знайдено матеріалів: 8

Собаки

Базовий догляд за собакою вдома

Як організувати щоденний режим, харчування, прогулянки та гігієну собаки.

#гігієна #прогулянки #вакцинація

[Читати повністю](#)

Коти

Як зменшити стрес у кота

Поради для адаптації кота після переїзду, появи нових людей або інших тварин.

#стрес #адаптація #поведінка

[Читати повністю](#)

Птахи

Безпечне утримання декоративних птахів

Основні правила розміщення клітки, очищення та щоденного спостереження за птахом.

#клітка #безпека #профілактика

[Читати повністю](#)

Кролики

Харчування кроликів

Що має бути в основі раціону кролика і яких продуктів краще уникати.

#харчування #сіно #травлення

[Читати повністю](#)

Гризуні

Гігієна клітки гризунів

Як підтримувати чистоту житла хом'яків, морських свинок та інших дрібних тварин.

#гігієна #клітка #здоров'я

[Читати повністю](#)

Усі тварини

Коли потрібно звернутися до ветеринара

Симптоми, які не варто ігнорувати під час догляду за домашніми тваринами.

#невідкладно #симптоми #ризик

[Читати повністю](#)

Усі тварини

Перші дні тварини у новому домі

Як підготувати житло та зменшити стрес після переїзду улюбленця.

#адаптація #безпека #новий дім

[Читати повністю](#)

Рептилії

Базові умови для рептилій

Температура, вологість і освітлення як основа здорового утримання рептилій.

#тераріум #температура #освітлення

[Читати повністю](#)

Рисунок Б.3 – Сторінка порад

FAQ

Часті питання

Короткі відповіді на питання щодо можливостей, обмежень і використання сервісу.

Чи замінює сайт ветеринара? +

Які тварини підтримуються? +

Як швидко можна отримати консультацію? -

У межах навчального проєкту заявка зберігається локально одразу після відправлення. Реальна швидкість відповіді залежала б від роботи консультанта.

Чи зберігаються заявки? +

Що робити при небезпечних симптомах? +

Чи можна використовувати сервіс для профілактики? +

Які дані потрібно вказати у заявці? +

Чи можна видалити заявку? +

Чи можна змінити статус заявки? +

Рисунок Б.4 – Сторінка питань

КОНТАКТИ

Зв'язок із PetCare Consult

Умовні контактні дані навчального сервісу та проста форма зворотного зв'язку.

PetCare Consult

Email: support@petcare-consult.local

Телефон: +380 44 000 00 00

Адреса: м. Київ, вул. Турботи про тварин, 12

Дані наведені як приклад для бакалаврського навчального проєкту.

Ім'я**Email****Повідомлення****Надіслати**

Рисунок Б.5 – Сторінка контакти