

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

Кафедра комп'ютеризованих систем управління

**ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри**

_____ Литвиненко О.Є.
“ _____ ” _____ 2022 р.

**ДИПЛОМНИЙ ПРОЕКТ
(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
“БАКАЛАВР”**

Тема: Online-сервіс пошуку домашніх тварин

Виконавець: _____ **Карпенко А.В.**

Керівник: _____ **к.т.н., доцент Халімон Н. Ф.**

Нормоконтролер: _____ **Тупота Є.В.**

Київ 2022

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютеризованих систем управління

Спеціальність 123 «Комп'ютерна інженерія»

(шифр, найменування)

Освітньо-професійна програма «Системне програмування»

Форма навчання денна

ЗАТВЕРЖУЮ

Завідувач кафедри

Литвиненко О. Є.

« ____ » _____ 2022 р.

ЗАВДАННЯ

на виконання дипломної роботи (проекту)

Карпенка Андрія Васильовича

(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема дипломної роботи (проекту): Online-сервіс пошуку домашніх тварин

затверджена наказом ректора від "15" лютого 2022 року № 251/ст.

2. Термін виконання роботи (проекту): з 16.05.2022 до 19.06.2022

3. Вихідні дані до роботи (проекту): мови програмування JavaScript і Java, реляційна система управління базами даних PostgreSQL, мови розмітки HTML і CSS, інтегроване середовище розробки Visual Studio Code

4. Зміст пояснювальної записки:

1) Організація online-сервісів пошуку.

2) Проектування веб-додатку для пошуку домашніх тварин.

3) Розробка веб-додатку для пошуку домашніх тварин.

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:

1) ER-діаграма бази даних веб-додатку для пошуку домашніх тварин

2) Головна сторінка веб-додатку для пошуку домашніх тварин

3) Сторінка реєстрації веб-додатку для пошуку домашніх тварин

4) Сторінка входу веб-додатку для пошуку домашніх тварин

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Ознайомитись з постановкою задачі дипломного проектування	18.05.22	
2	Вивчити спеціальну літературу і технічну документацію	14.05.22	
3	Проаналізувати системи управління базами даних та утиліти для адміністрування	16.05.22	
4	Написати розділ 1.	18.05.22	
5	Дослідити параметри системних баз даних, подання, оператори, функції, змінні мови <i>Transact-SQL</i> .	19.05.22	
6	Написати розділ 1.	21.05.22	
7	Провести проектування програмного засобу моніторингу параметрів відновлення бази даних кадрів підприємства	25.05.22	
8	Написати розділ 2.	27.05.22	
9	Провести розробку програмного засобу моніторингу параметрів відновлення бази даних кадрів підприємства	07.06.22	
10	Написати розділ 3.	12.06.22	
11	Оформити пояснювальну записку	13.06.22	
12	Підготувати графічний демонстраційний матеріал та доповідь	14.06.22	

7. Дата видачі завдання: “22” грудня 2021 р.

Керівник дипломної роботи (проекту) _____ Халімон Н.Ф.
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Карпенко А.В.
(підпис випускника) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту «*Online*-сервіс пошуку домашніх тварин»: *n* с., *n* рис., *n* літературних джерел.

Об'єкт проектування – інтернет-ресурси для пошуку інформації.

Предмет проектування – розробка *online*-сервісу пошуку домашніх тварин.

Метод проектування – застосування крос-платформного програмного засобу для розробки веб-додатків.

Дипломний проект присвячено тематиці створення програмного забезпечення для пошуку домашніх тварин.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1 ОРГАНІЗАЦІЯ ONLINE-СЕРВІСІВ ПОШУКУ	8
1.1. Особливості пошуку даних на сайтах	8
1.2. Існуючі online-сервіси пошуку домашніх тварин.....	12
1.3. Висновки до розділу.....	16
РОЗДІЛ 2 ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ ДЛЯ ПОШУКУ ДОМАШНІХ ТВАРИН.....	17
2.1. Функції веб-додатку для пошуку домашніх тварин	17
2.2. Проектування бази даних веб-додатку для пошуку домашніх тварин..	18
2.3. Проектування інтерфейсу веб-додатку для пошуку домашніх тварин .	20
2.4. Висновки до розділу.....	25
РОЗДІЛ 3 РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ ПОШУКУ ДОМАШНІХ ТВАРИН.....	27
3.1. Налаштування середовища розробки	27
3.2. Створення проекту веб-додатку для пошуку домашніх тварин	28
3.3. Процес розробки веб-додатку для пошуку домашніх тварин.....	31
3.4. Висновки до розділу.....	40
ВИСНОВКИ	41
СПИСОК БІБЛОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	43

ВСТУП

Веб-додатки та онлайн-сервіси, на відміну від звичайних сайтів, забезпечують інтерактивну взаємодію з користувачем. Це програмне забезпечення, яке взаємодіє з потужним веб-ресурсом, наприклад порталом для пошуку. В умовах війни для пошуку зниклих за різними категоріями створюється спеціальне програмне забезпечення, тому тема дипломного проєкту “Online-сервіс пошуку домашніх тварин” є актуальним завданням.

Пошук даних – це процес знаходження та обробки потрібної інформації або встановлення її відсутності у певному наборі даних. Пошук можна застосовувати для структурованих та неструктурованих даних. Структуровані дані – це дані, які відповідають моделі даних, мають чітко визначену структуру, послідовний порядок, є доступними та можуть бути використані людиною або комп'ютерною програмою. Неструктуровані дані – це дані, які не відповідають моделі даних і не мають чітко ідентифікованої структури, тому не можуть використані комп'ютерною програмою без попередньої структуризації.

Пошук даних здійснюється за допомогою спеціальних алгоритмів – алгоритмів пошуку. Існує величезна кількість алгоритмів пошуку. Їхня складність варіюється від простих лінійних алгоритмів до надзвичайно ефективних, але обмежених алгоритмів бінарного пошуку, а також алгоритмів, заснованих на представленні базової множини в іншій, більш підходящій для пошуку формі.

Для пошуку даних на сайтах використовують пошукові системи. Це системи, які дозволяють користувачам швидко й зручно отримувати доступ до потрібної інформації. Пошукові системи є надзвичайно затребуваними, оскільки обсяги інформації збільшуються з кожним днем. Найчастіше їх використовують як веб-сервіси для пошуку текстової або графічної інформації в Інтернеті, але існують також системи, які здатні шукати файли на *FTP*-серверах, товари в інтернет-магазинах, інформацію в групах новин *Usenet* тощо.

Online-сервіси пошуку домашніх тварин – програмне забезпечення, створене з метою допомоги людям у знаходженні або передачі їхніх домашніх улюбленців. Вони покращують ефективність роботи благодійних організацій,

підвищуючи рівень інформованості людей щодо даної проблеми й заохочуючи їх до волонтерської діяльності.

Найбільш поширеними серед *online*-сервісів пошуку домашніх тварин є *Petfinder*, *RSPCA* й *Adopt a Pet*. Їх можна знайти найпершими в *Google*-пошуку за відповідним запитом. Кожен із них представляє унікальне рішення, хоча й нагадує за функціоналом інші сервіси.

В другому розділі описано процес проектування веб-додатку для пошуку домашніх тварин. Він включає в себе визначення функціоналу веб-додатку, проектування бази даних, загального інтерфейсу і сторінок веб-додатку.

В третьому розділі описано процес розробки веб-додатку для пошуку домашніх тварин. До нього входять налаштування середовища розробки, розгортання відповідного проекту, створення бази даних, загального інтерфейсу і сторінок веб-додатку.

РОЗДІЛ 1

ОРГАНІЗАЦІЯ ONLINE-СЕРВІСІВ ПОШУКУ

1.1. Особливості пошуку даних на сайтах

Пошук даних – це процес знаходження та обробки потрібної інформації або встановлення її відсутності у певному наборі даних. Поняття пошуку даних безпосередньо пов'язане з поняттями фільтрації та сортування, що являють собою спосіб пошуку підмножини даних та спосіб їх упорядкування відповідно [1].

Для пошуку даних на сайтах використовують пошукові системи. Пошукова система – алгоритми, реалізовані сукупністю комп'ютерних програм, що надають користувачеві можливість швидкого доступу до необхідної інформації за допомогою пошуку в великій колекції доступних даних. Одне з найвідоміших застосувань пошукових систем – веб-сервіси для пошуку текстової або графічної інформації в Інтернеті. Існують також системи, які здатні шукати файли на *FTP*-серверах, товари в інтернет-магазинах, інформацію в групах новин *Usenet*. Для пошуку інформації за допомогою пошукової системи, користувач формулює пошуковий запит. Робота пошукової системи полягає в тому, щоб на запит користувача знайти документи, що містять або зазначені ключові слова, або слова, пов'язані з ключовими словами. Врешті-решт пошукова система генерує сторінку результатів пошуку. Така сторінка може містити різні типи результатів, наприклад веб-сторінки, зображення, аудіофайли. Деякі пошукові системи також отримують інформацію з відповідних баз даних та каталогів ресурсів в Інтернеті [2].

Кафедра КСУ

НАУ 22 20 28 000 ПЗ

<i>Виконав</i>	<i>Карпенко А.В.</i>			<i>Організація online-сервісів пошуку</i>	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Халімон Н.Ф.</i>				<i>Д</i>	8	9
<i>Консульт.</i>					<i>СП-435</i>		
<i>Норм. контр.</i>	<i>Тютота Є.В.</i>						
<i>Зав. Каф.</i>	<i>Литвиненко О.Є.</i>						

Для пошуку потрібних відомостей найзручніше скористатися сучасними пошуковими системами, які дозволяють швидко виявити необхідні відомості та забезпечують точність та повноту пошуку. При роботі з ними достатньо задати ключові слова, що найбільш точно відображають потрібну інформацію, або скласти складніший запит із ключових слів для уточнення області пошуку. Після введення запиту на пошук користувач отримає список посилань на документи в Інтернеті, які зазвичай називаються веб-сторінками або просто сторінками, в яких містяться вказані ключові слова. Зазвичай посилання доповнюються фрагментами тексту з документа, який часто допомагає визначити тематику знайденої сторінки. Натиснувши на посилання, можна перейти до вибраного документа [3].

Пошук можна застосовувати для структурованих та неструктурованих даних. Структуровані дані – це дані, які відповідають моделі даних, мають чітко визначену структуру, послідовний порядок, є доступними та можуть бути використані людиною або комп'ютерною програмою. Структуровані дані зазвичай зберігаються у таблицях баз даних зі стовпцями та рядками, які чітко визначають їхні атрибути. Для керування структурованими даними, що зберігаються в базах даних часто використовується мова запитів *SQL (Structured Query Language)*. Характеристики структурованих даних:

- відповідають моделі даних і мають чітко ідентифіковану структуру;
- зберігаються у вигляді рядків і стовпців;
- мають конкретне визначення, формат і зміст;
- зберігаються у фіксованих полях всередині запису або файлу;
- подібні сутності групуються разом, щоб утворити відносини або класи;
- сутності в одній групі мають однакові атрибути;
- можуть безперешкодно використовуватися комп'ютерними програмами;
- елементи даних є адресними, тому їх можна ефективно аналізувати та обробляти.

Джерелами структурованих даних можуть бути бази даних *SQL*, електронні таблиці (наприклад *Excel*), системи *OLTP (Online Transaction Processing)*, онлайн-форми, датчики *GPS* або *RFID*, журнали мережі, веб-сервера,

медичні прилади. Переваги структурованих даних: чітко визначена структура, яка забезпечує зручне зберігання та доступ; можна індексувати на основі текстового рядка, а також атрибутів; підлягають інтелектуальному аналізу, тобто несуть певний інформаційний зміст; зручні операції оновлення та видалення; зручні операції *Business Intelligence*, такі як зберігання даних; масштабованість; безпечність [4].

Неструктуровані дані – це дані, які не відповідають моделі даних і не мають чітко ідентифікованої структури, тому не можуть використані комп'ютерною програмою без попередньої структуризації. Неструктуровані дані не організовані заздалегідь визначеним способом або не мають попередньо визначеної моделі даних, тому вони не підходять для зберігання у базах даних.

Характеристики неструктурованих даних:

- не відповідають моделі даних і не мають чіткої структури;
- не можуть зберігатися у вигляді рядків і стовпців, як у базах даних;
- не відповідають жодним семантикам або правилам;
- не мають певного формату чи послідовності;
- непридатні для використання комп'ютерними програмами.

Джерелами неструктурованих даних можуть веб-сторінки, зображення (*JPEG, GIF, PNG* тощо), відео, пам'ятки, звіти, документи *Word*, презентації *PowerPoint*, опитування тощо. Переваги неструктурованих даних: не мають належного формату або послідовності; гнучкі через відсутність структури; є портативними; легко масштабуються; однорідні; масово використовуються у різноманітних програмах для аналітики та бізнес-аналітики. Недоліки неструктурованих даних: складно зберігати та керувати даними; індексування таких даних є складним і схильним до помилок через нечітку структуру та відсутність попередньо визначених атрибутів, через що результати пошуку не є дуже точними; вразливість. Проблеми, які виникають при зберіганні неструктурованих даних: великі обсяги інформації; збереження медіа-контенту (відео, зображення, аудіо тощо); складність виконання операцій оновлення, видалення та пошуку; вартість зберігання висока в порівнянні зі структурованими

даними; індексація. Можливі рішення проблем зберігання неструктурованих даних: конвертація даних в зручні для обробки формати; використання *CAS*-системи (*Content Addressable Storage*), що зберігає дані на основі їх метаданих; збереження даних у форматі *XML*; збереження даних в СУБД, яка підтримує тип даних *BLOB*. Неструктуровані дані проблематично інтерпретувати звичайними алгоритмами та індексувати. Тому дістати з них інформацію складніше. Для цього можна застосувати одне серед можливих рішень:

- класифікувати дані;
- зберігати дані у віртуальному сховищі та налаштувати їх автоматичне тегування;
- використовувати платформи типу *XOLAP*; *XOLAP* допомагає витягувати інформацію з електронних листів і документів на основі *XML*;
- використовувати штучний інтелект для інтерпретування даних [5].

Пошук даних здійснюється за допомогою спеціальних алгоритмів – алгоритмів пошуку. Існує величезна кількість алгоритмів пошуку. Їхня складність варіюється від простих лінійних алгоритмів до надзвичайно ефективних, але обмежених алгоритмів бінарного пошуку, а також алгоритмів, заснованих на представленні базової множини в іншій, більш підходящій для пошуку формі. Останні з перелічених алгоритмів мають особливе практичне значення, оскільки застосовуються в реальних додатках, які виконують вибір і зберігають масу інформації в величезних базах даних. Для вирішення задач пошуку не існує єдиного алгоритму, який би краще підходив до всіх випадків. Деякі з алгоритмів виконуються швидше інших, але для їх роботи потрібна додаткова оперативна пам'ять. Інші виконуються дуже швидко, але їх можна застосовувати тільки для попередньо відсортованих масивів. На відміну від алгоритмів сортування, в алгоритмах пошуку немає проблем стійкості, але при їх використанні можуть виникнути інші складності. Зокрема, у випадку, коли оброблювані дані можуть змінюватися, причому кількість змін відповідає кількості операцій пошуку, пошук слід розглядати в нерозривному зв'язку з двома іншими операціями - додаванням елементів у набір даних та видаленням з нього. У подібних ситуаціях необхідно

видозмінити структуру даних і алгоритми так, щоб досягти рівноваги між вимогами, передбаченими до кожної операції. Крім того, організація дуже великих наборів даних з метою виконання в них ефективного пошуку (а також додавання та видалення елементів) є надзвичайно складною задачею, практичне рішення якої особливо важливе з точки зору застосування [6].

1.2. Існуючі online-сервіси пошуку домашніх тварин

Online-сервіси пошуку домашніх тварин – програмне забезпечення, створене з метою допомоги людям у знаходженні або передачі їхніх домашніх улюбленців. Вони покращують ефективність роботи благодійних організацій, підвищуючи рівень інформованості людей щодо даної проблеми й заохочуючи їх до волонтерської діяльності. Особливо гостро проблема бездомних тварин постала в Україні від початку російського вторгнення, внаслідок якого мільйони людей кинули власні домівки, залишивши своїх улюбленців наодинці. Тому на сьогодні такі сервіси актуальні як ніколи.

Серед існуючих *online*-сервісів пошуку домашніх тварин слід відмітити *Petfinder*, *RSPCA* й *Adopt a Pet*. Їх можна знайти найпершими в *Google*-пошуку за відповідним запитом, а це значить, що вони є найбільш популярними та успішними. Розглянемо кожний з них.

Petfinder – *online*-сервіс пошуку домашніх тварин, що шукають собі хазяїна (рис 1.1).

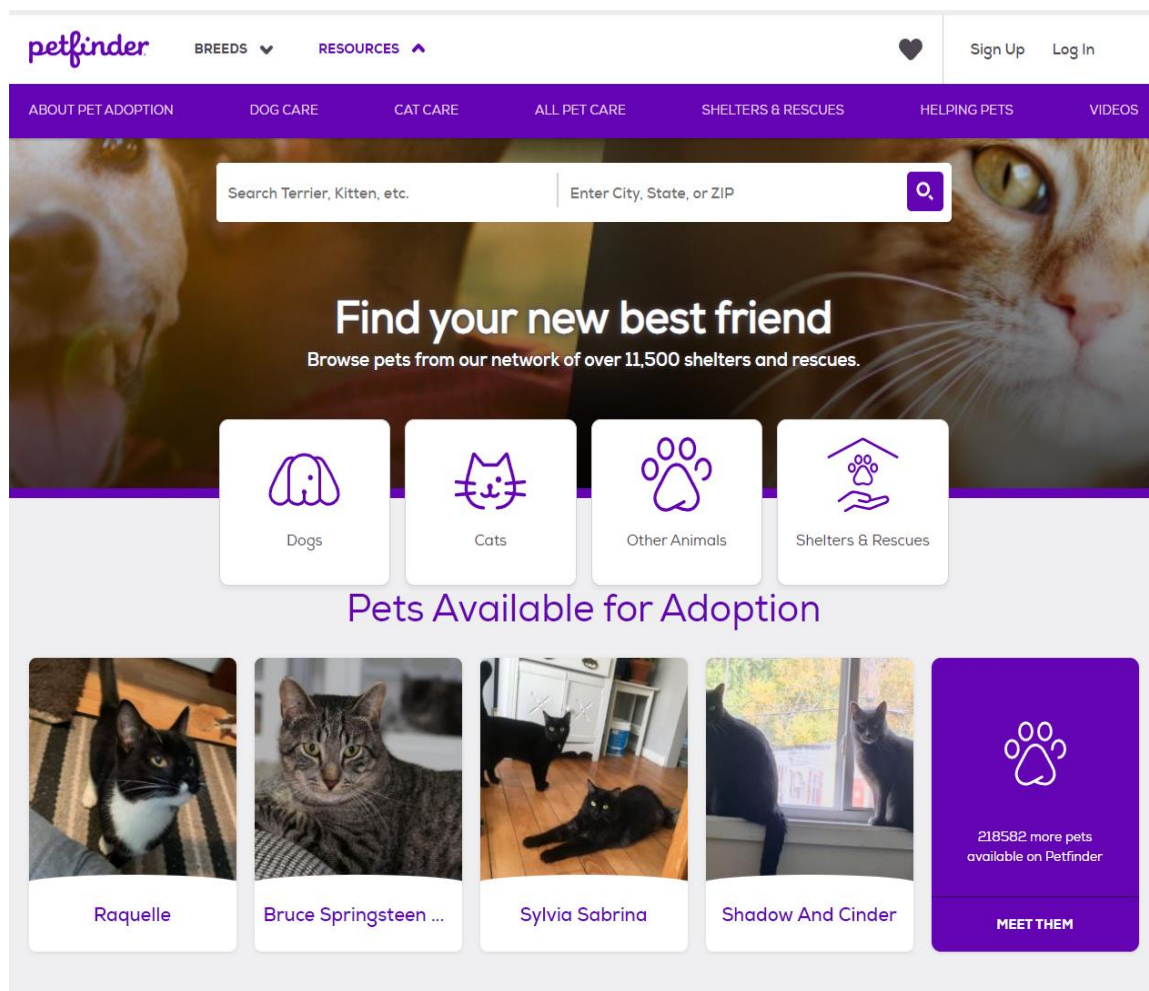


Рис. 1.1. *Online-сервіс пошуку домашніх тварин Petfinder*

Petfinder співпрацює близько з 11000 притулків та благодійних організацій, а тому налічує величезну кількість домашніх тварин, що оновлюється кожного дня. Пошук реалізовано таким чином, щоб можна було врахувати всі бажання користувача: можна вибрати вид тварини, її колір, місцезнаходження, вік, розмір тощо. Також перед пошуком користувачу пропонується пройти опитування для того, щоб в'яснити, кого він має намір приручити. Окрім пошуку сервіс містить свій блог зі статтями та відеоматеріалами, присвяченими, як правило, темам волонтерства, догляду та дресируванню домашніх тварин, а також відповіді на найпоширеніші запитання. Серед недоліків варто відмітити, що сервіс діє лише в обмеженій кількості країн: США, Канаді та Мексиці, й зобов'язує користувача реєструватись для того, щоб створити заявку на приручення [7].

RSPCA (*Royal Society for the Prevention of Cruelty to Animals*) – альтернативний *online*-сервіс пошуку та захисту домашніх тварин однієї з найбільших благодійних організацій, що функціонує з 1824 року (рис. 1.2).

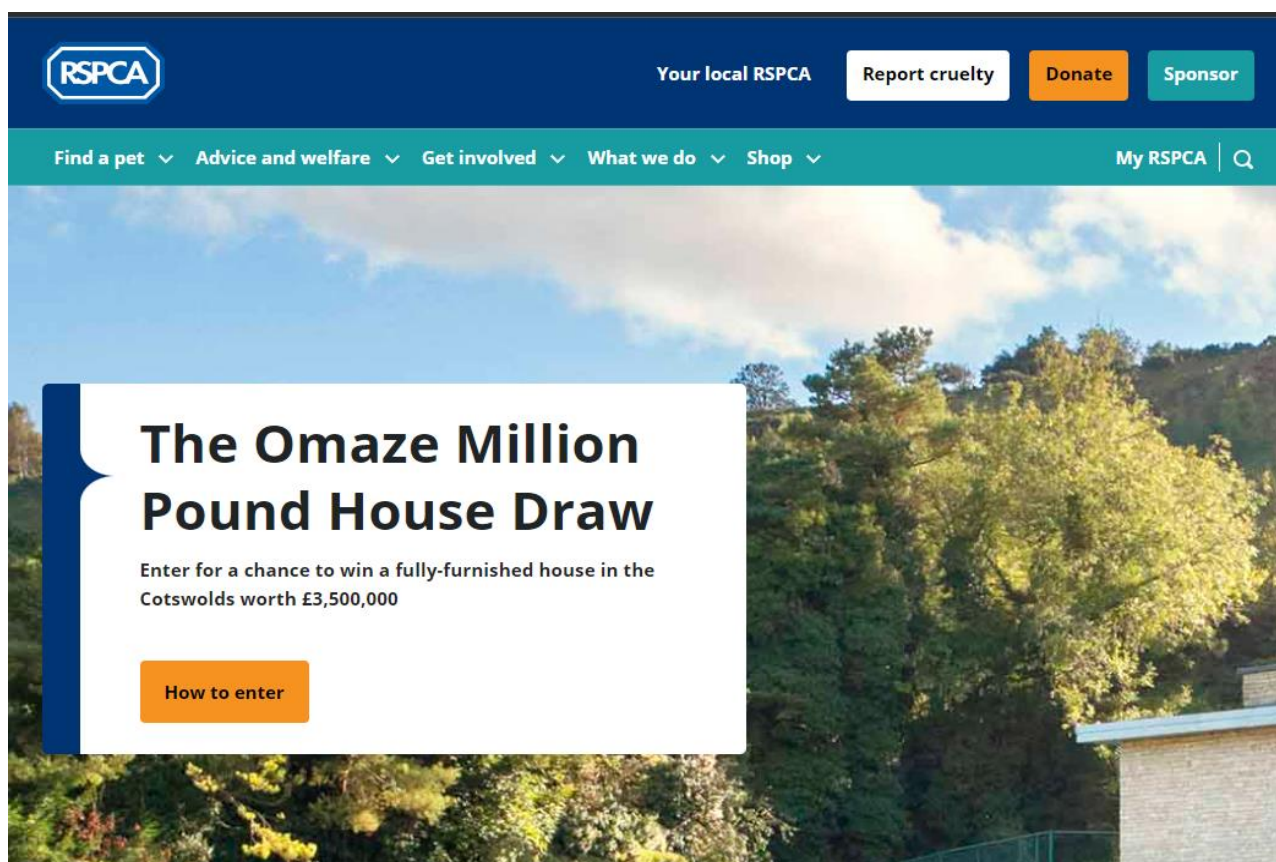


Рис. 1.2. *Online*-сервіс пошуку домашніх тварин *RSPCA*

Сервіс дуже схожий на *Petfinder*: він також складається із блогу та пошуку, щоправда не такому гнучкому, як у *Petfinder*. Блог містить статті, здебільшого, на тему охорони тварин та рекомендаціями щодо обходження з ними. До того ж *RSPCA* дає можливість створювати пожертвування й має власний магазин, в якому можна придбати страхування і товари для домашніх тварин. Окрім цього сервіс дозволяє створювати й відправляти скарги на засвідчені випадки жорстокості по відношенню до тварин. Недоліки: *RSPCA* працює лише в межах Англії та Уельсу, для відповіді на оголошення користувач обов'язково має зареєструватися [8].

Adopt a Pet – ще один *online*-сервіс пошуку домашніх тварин компанії *Kinship*, що займається допомогою для притулків (рис. 1.3).

Adopt the Perfect Pet

from more than 17,000 animal shelters & rescues

Dogs Cats Other Pets Shelters & Rescues


Location Invalid Location

Enter a zip / postal or city, state (2 letter state).

Age Breed

Search

Need to Rehome Your Pet?
Adopt-a-Pet.com is now offering our Rehome program — the safe, reliable, and free way to find a loving new home for your pet.
[Learn More >](#)



Newsletter Signup

Get useful adoption info and latest pet training tips.

Subscribe

By signing up for our newsletter, you consent to us contacting you by email. You can unsubscribe at any time by clicking the "Unsubscribe" link in each newsletter.

Barks and Purrs

The latest news and chatter from social media

 **Twitter**
[@adoptapetcom](#)

[Read More >](#)

Рис. 1.3. Online-сервіс пошуку домашніх тварин *Adopt a pet*

Пошукова система сервісу являє собою дещо середнє між пошуковими системами *Petfinder* та *RSPCA*: вона менш гнучка за пошук *Petfinder*, але складніша за пошук *RSPCA*. Є можливість вказати вид, місцезнаходження, вік і породу. *Adopt a Pet* також включає в себе блог, що містить статті, рекомендації та посібники. Як і *RSPCA*, сервіс має власний магазин з товарами для домашніх тварин. Окрім пошуку, *Adopt a Pet* дає можливість розміщувати власних улюбленців. Інші користувачі побачать таке оголошення й зможуть зв'язатися з його автором. Недоліки: працює лише в США та Канаді, для створення/відповіді на оголошення вимагає від користувача зареєструватися [9].

1.3. Висновки до розділу

Пошук даних – це процес знаходження та обробки потрібної інформації або встановлення її відсутності у певному наборі даних. Для пошуку даних на сайтах використовують пошукові системи. Пошукова система – алгоритми, реалізовані сукупністю комп'ютерних програм, що надають користувачеві можливість швидкого доступу до необхідної інформації за допомогою пошуку в великій колекції доступних даних.

Пошук можна застосовувати для структурованих та неструктурованих даних. Структуровані дані – це дані, які відповідають моделі даних, мають чітко визначену структуру, послідовний порядок, є доступними та можуть бути використані людиною або комп'ютерною програмою. Неструктуровані дані – це дані, які не відповідають моделі даних і не мають чітко ідентифікованої структури, тому не можуть використані комп'ютерною програмою без попередньої структуризації.

Online-сервіси пошуку домашніх тварин – програмне забезпечення, створене з метою допомоги людям у знаходженні або передачі їхніх домашніх улюбленців. Серед існуючих *online*-сервісів пошуку домашніх було розглянуто *Petfinder*, *RSPCA* й *Adopt a Pet*.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ ДЛЯ ПОШУКУ ДОМАШНІХ ТВАРИН

2.1. Функції веб-додатку для пошуку домашніх тварин

При проектуванні веб-додатку для пошуку домашніх тварин було визначено наступні функції:

- аутентифікація (реєстрація та вхід);
- створення оголошень про загублених або зниклих домашніх тварин;
- пошук оголошень про загублених або зниклих домашніх тварин;
- перегляд оголошень про загублених або зниклих домашніх тварин.

За аутентифікацію має відповідати окремий модуль, що реалізує реєстрацію, вхід та вихід користувача, а також налаштовуватиме доступ до захищених даних. Реєстрація передбачає можливість створення нового акаунта після надання користувачем необхідної інформації. Така інформація включає в себе прізвище, ім'я, електронну пошту та пароль. Після відправки її на сервер у базі даних створюється відповідний запис акаунту, а також відбувається автоматичний вхід. У будь-який момент можна вийти з нього й зайти пізніше за потреби. Вхід відбувається за електронною поштою та паролем до конкретного акаунту.

Створити оголошення можна лише авторизованому користувачу. Для цього потрібно вказати тип оголошення, заголовок, опис, локацію, вид тварини, номер телефону й фото. Оголошення можуть бути створені для загублених та знайдених тварин. Локація та вид тварини, вказаної в оголошенні, можуть приймати лише ті значення, які визначені відповідними моделями даних на сервері.

Кафедра КСУ				<i>НАУ 22 20 28 000 ПЗ</i>			
<i>Виконав</i>	<i>Карпенко А.В.</i>			<i>Проектування веб-додатку для пошуку домашніх тварин</i>	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Халімон Н.Ф.</i>				<i>Д</i>	<i>17</i>	<i>10</i>
<i>Консульт.</i>					<i>СП-435</i>		
<i>Норм. контр.</i>	<i>Тупота Є.В.</i>						
<i>Зав. Каф.</i>	<i>Литвиненко О.Є.</i>						

Пошук оголошень відбувається або за змістом (заголовком, описом), або за фото. Для пошуку за змістом можна застосувати фільтри за типом оголошення, локацією та видом тварини. Для пошуку за фото можна застосувати лише фільтр за типом оголошення. У результаті пошуку має бути представлено таку інформацію про оголошення:

- фото;
- заголовок;
- вид тварини;
- локація;
- дата публікації.

При перегляді оголошення відображається повна інформація про нього. Вона включає в себе фото оголошення, заголовок, опис, дату публікації, локацію, а також інформацію про автора: його ім'я, прізвище й номер телефону.

2.2. Проектування бази даних веб-додатку для пошуку домашніх тварин

При проектуванні бази даних веб-додатку для пошуку домашніх тварин було використано реляційну систему управління базами даних *PostgreSQL*. Вона реалізовує більшу частину *SQL*-стандарту й сучасних функцій, завдяки чому є надійним і перевіреним рішенням.

Усього під час проектування бази даних веб-додатку для пошуку домашніх тварин було створено чотири таблиці бази даних. Це таблиці *users*, *announcements*, *cities* і *pets*.

Таблиця *users* призначена для зберігання інформації про користувачів. Вона має такі поля: *id* – *UUID*-ідентифікатор запису, має бути унікальним; *name* – ім'я користувача, символьний тип даних; *surname* – прізвище користувача, символьний тип даних; *email* – електронна пошта користувача, символьний тип даних, має бути унікальною; *password* – пароль користувача, символьний тип даних.

Таблиця *announcements* призначена для зберігання інформації про створені оголошення. Вона має такі поля: *id* – *UUID*-ідентифікатор запису, має бути унікальним; *title* – заголовок оголошення, символьний тип даних; *description* – опис оголошення, символьний тип даних; *date* – дата публікації оголошення, тип даних «дата/час»; *location* – локація оголошення, внутрішній ключ зв'язку з таблицею *cities*, *OID*-тип даних; *user_id* – зовнішній ключ зв'язку з таблицею *users*, *OID*-тип даних; *phone* – номер телефону автора оголошення, символьний тип даних; *photoUrl* – посилання на фото оголошення, символьний тип даних; *species* – вид домашньої тварини, вказаної в оголошенні, внутрішній ключ зв'язку з таблицею *pets*, *OID*-тип даних.

Таблиця *cities* призначена для зберігання інформації про локації, для яких може бути створене оголошення. Вона має такі поля: *id* – *UUID*-ідентифікатор запису, має бути унікальним; *name* – назва локації, символьний тип даних.

Таблиця *pets* призначена для зберігання інформації про види домашніх тварин, для яких може бути створене оголошення. Вона має такі поля: *id* – *UUID*-ідентифікатор запису, має бути унікальним; *name* – назва виду домашньої тварини, символьний тип даних.

У результаті проектування бази даних веб-додатку для пошуку домашніх тварин було спроектовано *ER*-діаграму [10]. Її зображено на рис. 2.1.

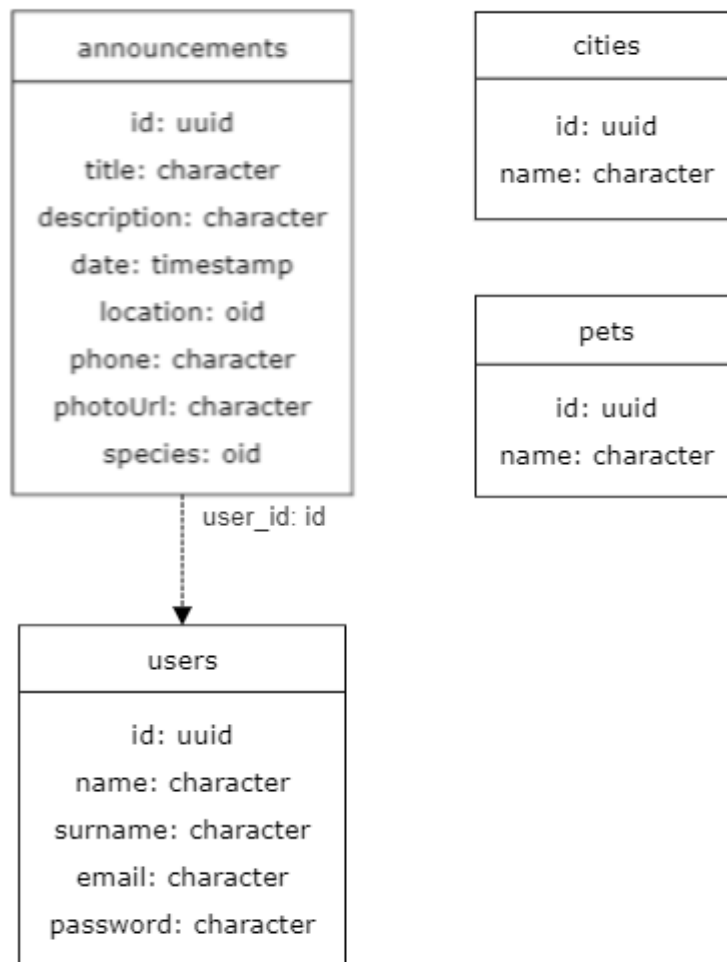


Рис. 2.1. *ER*-діаграма бази даних веб-додатку для пошуку домашніх тварин

2.3. Проектування інтерфейсу веб-додатку для пошуку домашніх тварин

Інтерфейс веб-додатку для пошуку домашніх тварин має бути зручним для використання різними категоріями людей на різних пристроях та у різних середовищах. Для цього при його проектуванні має бути дотримано наступні принципи:

- доступність;
- адаптивність;
- кросбраузерність.

Доступність передбачає створення такого інтерфейсу, яким можуть без перешкод користуватись усі люди, в тому числі й люди з обмеженими можливостями. Вона стосується, в першу чергу, наповнення сторінок та його оформлення. Створення доступного інтерфейсу вимагає дотримання певних вимог. Такі вимоги висвітлено у веб-стандарті *W3C – Accessibility*. До них можна віднести, наприклад, використання голосового вводу, дотримання достатньої контрастності розмітки або дотримання семантики при створенні розмітки для користувачів, що користуються зчитувачами екранів через проблеми з зором, лише клавіатурою або мишкою тощо. Доступність веб-додатку завжди можна легко оцінити за допомогою спеціальних інструментів. До них належить, наприклад, *Lighthouse*, що доступний в будь-який сучасному браузері [11].

Адаптивність інтерфейсу характеризує його властивість підлаштовуватись під пристрій користувача. Це стосується як його вигляду, так і його зручності. Проблема є дуже актуальною, оскільки спектр пристроїв, які використовуються для перегляду веб-сторінок, є досить широким. До нього відносяться мобільні телефони, планшети, ноутбуки, комп'ютери, телевізори тощо. Всі ці пристрої відрізняються розмірами екранів й операційними системами. Адаптивним інтерфейс можна вважати тоді, коли ним буде зручно користуватися на будь-якому з перелічених пристроїв. Тестувати адаптивність можна за допомогою призначених для цього інструментів. Усі сучасні браузери надають такі інструменти прямо із коробки. Вони дозволяють змінювати розмір вікна й версію – настільну або мобільну – браузера. Крім того, вони містять список пресетів (шаблонів) найпопулярніших пристроїв. Якщо вибрати один із таких пресетів, то для вікна браузеру автоматично встановляться розміри екрану вибраного пристрою [12].

Кросбраузерність інтерфейсу характеризує його сталий вигляд і функціонування у різних браузерах. На сьогоднішній день існує величезна кількість браузерів і їх версій. Серед них – *Google Chrome, Firefox, Microsoft Edge, Opera, Safari, Samsung Internet* та інші. Усі вони розробляються, орієнтуючись на сучасні веб-стандарти, проте це відбувається під контролем різних команд. До

того ж в основі перелічених браузерів лежать різні двигуни, що напряду визначає їхню поведінку. Очевидно, що в такому випадку браузери значно можуть відрізнитися один від одного. Тому дуже важливо, щоб інтерфейс виглядав і поведив себе однаково незалежно від браузера. Головною проблемою у досягненні цієї цілі є те, що не всі браузери є кросплатформеними, тобто підтримують не всі операційні системи, а це значить, що протестувати інтерфейс для всіх браузерів в межах однієї системи фактично неможливо. Для цього існують емулятори. Це спеціальні програми, що імітують поведінку браузерів. Вони дозволяють без особливих зусиль переконатися в тому, що інтерфейс є кросбраузерним [13].

Інтерфейс веб-додатку для пошуку домашніх тварин складається із головної сторінки, сторінок реєстрації, входу, створення оголошення, пошуку оголошень зниклих домашніх тварин, пошуку оголошень знайдених домашніх тварин, пошуку оголошень за фото, перегляду конкретного оголошення. Кожна зі сторінок складається з хедеру, бічного меню та її вмісту.

Хедер містить логотип і кнопку. Клік по логотипу викликає перехід на головну сторінку. Кнопка меню призначена для розгортання бічного меню.

Навігаційне меню за замовчуванням приховане. Його можна розгорнути, натиснувши на відповідну кнопку у хедері. Воно з'явиться з лівого боку. Меню містить посилання на сторінки пошуку зниклих домашніх тварин, загублених домашніх тварин та пошуку за фото. Якщо користувача авторизовано, то для нього також відображається посилання на сторінку створення оголошення та кнопка виходу з аканту. В протилежному випадку відображаються посилання на сторінки входу та реєстрації. Меню приховується при переході за обраним посиланням або при натисканні на області поза ним.

Головна сторінка представляє собою банер. Банер містить заголовок, який пояснює, для чого призначений додаток, і допоміжну навігацію для користувача з посиланнями на сторінки пошуку оголошень знайдених домашніх тварин і створення оголошення.

Сторінка реєстрації складається із заголовку «Створити акаунт» та форми. Форма містить текстові поля для вводу електронної пошти, імені, прізвища, паролю і кнопку «Створити», яка за замовчуванням є неактивною. Кнопка стає активною після введення коректних значень для кожного з полів. Поля валідуються кожного разу після вводу всередині них символу. При провалі валідації під відповідним полем з'являється повідомлення про помилку. Так, електронна пошта є обов'язковим полем відповідного формату; прізвище, ім'я й пароль також є обов'язковими. Після натискання кнопки «Створити» введені дані відправляються на сервер. Там валідується на унікальність електронна пошта. У разі помилки надсилання форми скасовується, а під нею відображається повідомлення про помилку. При успіху ж створюється новий акаунт і відбувається вхід у нього. Після входу користувача перенаправляє на головну сторінку.

Сторінка входу складається із заголовку «Війти в акаунт» та форми. Форма містить текстові поля «Емейл», «Пароль» і кнопку «Війти», яка за замовчуванням є неактивною. Кнопка стає активною, коли всі поля заповнені. Після натискання на кнопку дані відправляються на сервер. Якщо дані не валідні, то надсилання форми скасовується і під нею відображається повідомлення про помилку. Інакше відбувається вхід до акаунту і перенаправлення користувача на головну сторінку.

Сторінка створення оголошення складається із заголовку «Створення оголошення» та форми. Форма містить радіо-кнопки вибору типу оголошення, текстові поля для вводу назви оголошення, опису оголошення і номеру телефону автора, кнопку завантаження зображення оголошення, селектив вибору локації оголошення і виду домашньої тварини, кнопку «Розмістити» і кнопку «Очистити», за допомогою якої можна в будь-який момент скинути форму до початкового порожнього стану. Доступні опції вибору локації оголошення та виду домашньої тварини завантажуються з серверу, де вони й зберігаються. За замовчуванням обрано радіо-кнопку, що відповідає типу оголошення для загублених тварин, першу локацію оголошення і перший вид домашньої тварин із відповідних списків, а кнопка «Розмістити» є неактивною. Кнопка стає активною

після введення коректних значень для кожного з текстового полів і прикріплення фото оголошення. Поля валідуються кожного разу після вводу всередині них символу. При провалі валідації під відповідним полем з'являється повідомлення про помилку. Так, назва оголошення є обов'язковим полем довжиною від 10 до 50 символів; опис оголошення є обов'язковим полем довжиною до 200 символів; номер телефону є обов'язковим полем довжиною у 10 символів і відповідного формату; фото оголошення є обов'язковим полем. При натисканні кнопки «Розмістити» введені дані відправляються на сервер. У разі помилки надсилання форми скасовується, а під нею відображається повідомлення про помилку. При успіху ж створюється нове оголошення й відбувається перенаправлення користувача на сторінку оголошень відповідного типу.

Сторінка пошуку оголошень зниклих домашніх тварин складається з форми пошуку оголошень і списку оголошень, знайдених за введеним текстом. Форма пошуку містить текстове поле для вводу пошукового запиту, кнопку пошуку, фільтри за типом оголошення та видом домашньої тварини. Фільтри представлені у вигляді мультиселектів, тобто селектів з вибором одного чи більше варіантів. Доступні опції вибору локації оголошення та виду домашньої тварини для цих селектів завантажуються з серверу, де вони й зберігаються. Пошук відбувається при натисканні на кнопку пошуку або при зміні значення одного із фільтрів. Результат пошуку відображається у вигляді списку оголошень зниклих домашніх тварин або повідомлення про його відсутність. Кожен елемент списку є карточкою з інформацією про оголошення: його фото, заголовок, вид домашньої тварини, локація і дата публікації. Якщо клікнути по заголовку, то можна перейти на сторінку повного перегляду цього оголошення.

Сторінка пошуку оголошень знайдених домашніх тварин складається з форми пошуку оголошень і списку оголошень, знайдених за введеним текстом. Форма пошуку містить текстове поле для вводу пошукового запиту, кнопку пошуку, фільтри за типом оголошення та видом домашньої тварини. Фільтри представлені у вигляді мультиселектів, тобто селектів з вибором одного чи більше варіантів. Доступні опції вибору локації оголошення та виду домашньої тварини

для цих селективів завантажуються з серверу, де вони й зберігаються. Пошук відбувається при натисканні на кнопку пошуку або при зміні значення одного із фільтрів. Результат пошуку відображається у вигляді списку оголошень знайдених домашніх тварин або повідомлення про його відсутність. Кожен елемент списку є карточкою з інформацією про оголошення: його фото, заголовок, вид домашньої тварини, локація і дата публікації. Якщо натиснути лівою кнопкою миші по заголовку, то можна перейти на сторінку повного перегляду цього оголошення.

Сторінка пошуку оголошень за фото складається із форми пошуку оголошень і списку оголошень, знайдених за прикріпленим фото. Форма пошуку містить кнопку завантаження зображення шуканої домашньої тварини, кнопку пошуку, фільтри типу оголошення, представлені у вигляді радіо-кнопок. За замовчуванням обрано радіо-кнопку, що відповідає типу оголошення для загублених тварин, а кнопка пошуку є неактивною. Вона стає активною після прикріплення будь-якого зображення. Пошук відбувається тільки при натисканні на неї. Результат пошуку відображається у вигляді списку оголошень обраного типу або повідомлення про його відсутність. Кожен елемент списку є карточкою з інформацією про оголошення: його фото, заголовок, вид домашньої тварини, локація і дата публікації. Якщо натиснути лівою кнопкою миші по заголовку, то можна перейти на сторінку повного перегляду цього оголошення.

Сторінка перегляду оголошення містить повну інформацію про оголошення. Вона складається з фото оголошення, заголовку оголошення, дати публікації оголошення, локації оголошення та контактних даних автора оголошення, а саме його прізвища, імені й номеру телефону.

2.4. Висновки до розділу

При проектуванні веб-додатку для пошуку домашніх тварин було визначено наступні функції:

- аутентифікація (реєстрація та вхід);
- створення оголошень про загублених або зниклих домашніх тварин;

- пошук оголошень про загублених або зниклих домашніх тварин;
- перегляд оголошень про загублених або зниклих домашніх тварин.

При проектуванні бази даних веб-додатку для пошуку домашніх тварин було використано реляційну систему управління базами даних *PostgreSQL*. Усього під час проектування бази даних веб-додатку для пошуку домашніх тварин було створено чотири таблиці бази даних. Це таблиці *users*, *announcements*, *cities* і *pets*.

Інтерфейс веб-додатку для пошуку домашніх тварин має бути зручним для використання різними категоріями людей на різних пристроях та у різних середовищах. Для цього при його проектуванні має бути дотримано наступні принципи:

- доступність;
- адаптивність;
- кросбраузерність.

Інтерфейс веб-додатку для пошуку домашніх тварин складається із головної сторінки, сторінок реєстрації, входу, створення оголошення, пошуку оголошень зниклих домашніх тварин, пошуку оголошень знайдених домашніх тварин, пошуку оголошень за фото, перегляду конкретного оголошення.

РОЗДІЛ 3

РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ ПОШУКУ ДОМАШНІХ ТВАРИН

3.1. Налаштування середовища розробки

Початковим етапом розробки веб-додатку для пошуку домашніх тварин було налаштування середовища розробки. В якості середовища розробки обрано *Visual Studio Code*. Сам процес розробки відбувався в операційній системі *Windows* на платформі *Node.js*. Також в ході розробки було застосовано пакетний менеджер *npm*, який є складовою частиною платформи *Node.js*, а також інтерфейс командного рядку *Vue CLI*.

Visual Studio Code – безкоштовне кросплатформене інтегроване середовище розробки, створене компанією *Microsoft*. Використовується, насамперед, для розробки веб-додатків і хмарних додатків. Завантажити *Visual Studio Code* можна з офіційного сайту продукту. Після встановлення і запуску майже все готово для написання коду: *Visual Studio Code* із коробки надає інструменти для роботи з *HTML*, *CSS* і *JavaScript*, проте для роботи з фреймворком *Vue.js* виконано додаткове налаштування. Для забезпечення найкращих умов розробки також встановлено плагін *Vetur* [14].

Node.js – платформа для виконання *JavaScript*-коду поза межами браузера. Головною складовою частиною платформи є пакетний менеджер *npm*. Він дозволяє розгортати *Node.js*-проекти, управляти їх залежностями або глобальними (системними) залежностями, а також виконувати скрипти та сценарії. Встановити *Node.js* можна з офіційного сайту продукту. На вибір надаються дві її версії: *LTS* (*Long-Term Support*) і поточна експериментальна. Після встановлення платформи у командному рядку будуть доступні імена командлетів *node* та *npm*. Серед найчастіше використовуваних команд слід відмітити *node*, що запускає *JavaScript*-файл, *npm* *init*, що

Кафедра КСУ				НАУ 22 20 28 000 ПЗ			
Виконав	Карпенко А.В.			Розробка веб-додатку для пошуку домашніх тварин	Літера	Аркуш	Аркушів
Керівник	Халімон Н.Ф.				Д	27	14
Консульт.					СП-435		
Норм. контр.	Тупота С.В.						
Зав. Каф.	Литвиненко О.Є.						

створює *Node.js*-проект, *npm install*, що встановлює залежності, і *npm run*, що запускає скрипт [15].

При розробці було використано програмне забезпечення *Vue CLI*. *Vue CLI* – інструмент для створення *Vue.js*-проектів, що передбачає гнучку їхню конфігурацію, оптимізує їх, задає шаблон архітектури, автоматично налаштовує сервер розробки, надає скрипти для запуску цього серверу, збірки тощо. По своїй суті є абстракцією над *webpack* – системою збору *Node.js*-додатків. Також має плагіни – *npm*-пакети, що надають додаткових можливостей проектам [16]. Архітектура, яку пропонує *Vue CLI*, передбачає застосування *SFC*-підходу (*Single File Components*). За такого підходу весь вихідний код додатку представлений у вигляді деревовидної структури *Vue*-компонентів, що поєднують у собі одночасно модифікований для зв'язування даних *HTML*-шаблон, *JavaScript*-код і *CSS*-стилі, чим реалізують *MVVM*-підхід (*Model-View-ViewModel*) [17]. Щоб встановити *Vue CLI*, потрібна *Node.js* версії 8.9 або вище. Для цього треба скористатися пакетним менеджером *npm*, а саме застосувати команду *npm install -g @vue/cli*. У командному рядку з'явиться командлет *vue* з основними командами *vue create* і *vue ui*. Перша запускає майстер створення проекту у консолі, а друга викликає графічний інтерфейс, за допомогою якого створити проект можна інтерактивно.

3.2. Створення проекту веб-додатку для пошуку домашніх тварин

Для створення проекту веб-додатку для пошуку домашніх тварин було використано: пакетний менеджер *npm* платформи *Node.js*, термінал інтегрованого середовища розробки *Visual Studio Code*, інтерфейс командного рядку *Vue CLI*, мови програмування *JavaScript* і *Java*. Першим кроком було введення команди *vue create pet-savior* у терміналі. В результаті її виконання відкрився майстер налаштування проекту, і було обрано налаштування за замовчуванням. Після завершення налаштування було створено папку проекту *pet-savior* з усіма необхідними файлами. Їх структуру зображено на рис. 3.1.

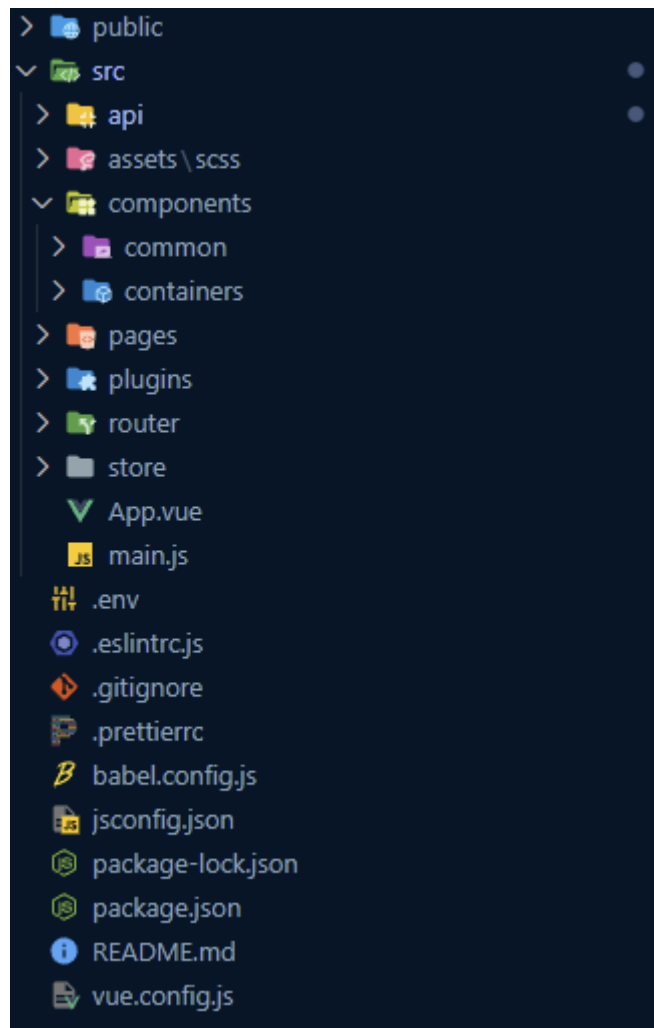


Рис. 3.1. Структура проекту веб-додатку для пошуку домашніх тварин

Директорія *public* містить скомпільований код додатку для запуску його на сервері розробки. Це файл розмітки *index.html*, файли *CSS*-стилів та *JavaScript*-файли.

Директорія *src* містить увесь вихідний код проекту. До неї належать:

- модуль *api*, призначений для формування *HTTP*-запитів клієнтської частини до серверної частини;
- модуль *assets* з медіа-файлами: зовнішніми таблицями стилів, фото, шрифтами, іконками тощо;
- модуль *components* з усіма *Vue*-компонентами: *common*-компонентами, що призначені для перевикористання або розширення, та компонентами-контейнерами, що є розширеннями *common*-компонентів або окремими компонентами, які обов'язково реалізують конкретну бізнес-логіку;

- модуль *pages* з усіма сторінками додатку;
- модуль *plugins* з плагінами – окремими модулями, які потрібно запустити при ініціалізації додатку;
- модуль *router* з конфігурацією маршрутизатора та шляхами до сторінок додатку;
- модуль *store*, що представляє собою глобальне сховище даних додатку;
- файл *App.vue* – кореневий *Vue*-компонент додатку;
- файл *main.js* – вхідна точка додатку.

Файл *.env* – конфігурація середовища виконання коду. Він містить змінні, які можуть бути використанні під час компіляції проекту.

Файли *.eslintrc* і *.prettierrc* містять конфігурації інструментів для форматування коду. Ці інструменти забезпечують сталий вигляд коду по всьому проекту.

Файл *.gitignore* містить інформацію, потрібну для коректної роботи системи контролю версій *Git*. Він повідомляє *Git* про файли, які не потрібно відстежувати.

Файл *jsconfig.json* містить інформацію для середовища розробки. Покращує умови розробки.

Файл *babel.config.js* містить конфігурацію транспайлеру. Транспайлер – інструмент, який забезпечує підтримку сучасної специфікації *JavaScript* у старих браузерах.

Файл *package.json* містить усю інформацію про проект. Це його назва, версія, доступні для виконання скрипти, встановлені залежності додатку й залежності, необхідні для розробки, а також конфігурації деяких інструментів та інша корисна інформація.

Файл *package-lock.json* представляє собою граф залежностей проекту. Потрібен *npm* для їх коректного встановлення.

Файл *README.md* містить опис проекту. Це може бути інформація про призначення проекту, технологічний стек, команду розробників тощо.

Файл *vue.config.js* містить конфігурацію самого додатку. Це може бути, наприклад, хост і порт серверу, на якому запускається додаток.

Перед початком роботи з проектом було встановлено всі його залежності за допомогою команди *npm install*. Справа в тому, що файли залежностей можуть досягати великих розмірів. Через це вони не постачаються разом з іншими файлами проекту, а встановлюються вручну розробником. Після виконання команди в директорії проекту з'явилася директорія *node_modules* з усіма залежностями.

Проект має два скрипти: *serve* і *build*. Скрипт *serve* запускає локальний сервер для розробки. Скрипт *build* будує оптимізовану версію додатку, готову для публікації. Визвати обидва скрипти можна ввівши у консолі команди *npm run serve* і *npm run build* відповідно.

3.3. Процес розробки веб-додатку для пошуку домашніх тварин

Розробка веб-додатку для пошуку домашніх тварин включала в себе наступні етапи: створення базової розмітки; створення модуля аутентифікації разом із сторінками реєстрації *PageRegistration* і входу *PageAuthorization*; реалізації функції створення оголошення і відповідної сторінки *PageCreateAd*; реалізації функції пошуку оголошень разом зі створенням сторінок пошуку зниклих домашніх тварин *PageLostAds*, пошуку знайдених домашніх тварин *PageFoundAds*, пошуку за фото *PagePhotoMatching*; створення сторінки перегляду оголошення *PageAd*. Для створення усієї розмітки було використано UI-фреймворк *Vuetify*, для виконання *HTTP*-запитів до *API*-серверу – *JavaScript*-бібліотеку *axios*.

Розробку веб-додатку для пошуку домашніх тварин було розпочато зі створення базової розмітки. До неї входять компоненти хедера *Header* і бічного меню *NavigationBar*.

Хедер розміщується у верхній частині екрану. У середину хедеру додано логотип-гіперпосилання та кнопку у вигляді бургеру. Компонент логотипу *HeaderLogo* має вхідний параметр *to* зі значенням */*. Він визначає, за яким

маршрутом потрібно направити користувача при взаємодії з посиланням: або при натисканні лівою кнопкою миші по ньому, або при фокусі на ньому за допомогою клавіші *Tab* і натисканні клавіші *Enter*. В даному випадку користувача направляє на головну сторінку. Для кнопки же додано обробник події *click*, який управляє переключенням стану бічного меню з прихованого до видимого і навпаки.

Компонент бічного меню *NavigationBar* за замовчанням приховано. За його видимість відповідає вхідний параметр *isVisible*, що приймає значення булевого типу і початково дорівнює *false*. Переключити значення цього параметру можна за допомогою кнопки меню у хедері. При переході зі стану *false* до *true* відбувається розгортання меню, в іншому ж випадку – згортання. У середині меню розміщено компоненти *RouterLink*, що реалізують гіперпосилання на сторінки пошуку зниклих тварин, пошуку знайдених тварин та пошуку за фото. Для авторизованого користувача також відображаються посилання на сторінку створення оголошення і кнопка виходу з акаунту, а для неавторизованого користувача – посилання на сторінки реєстрації та входу.

Наступною було створено головну сторінку *PageHome* (рис. 3.2).

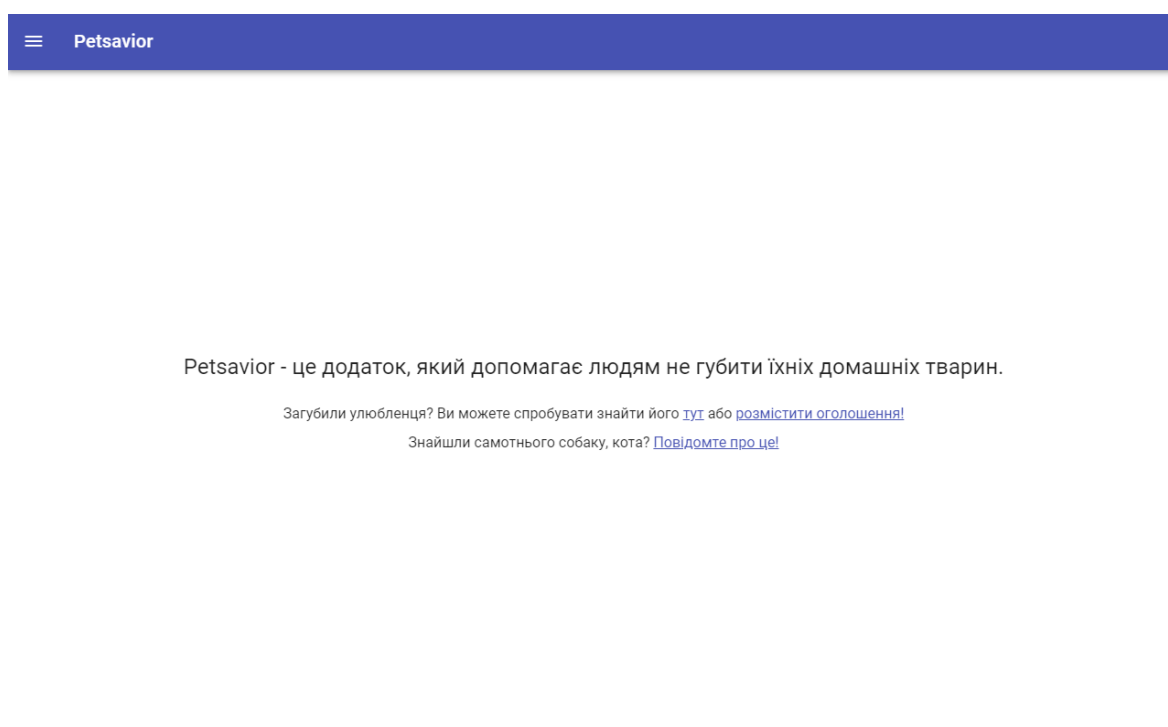
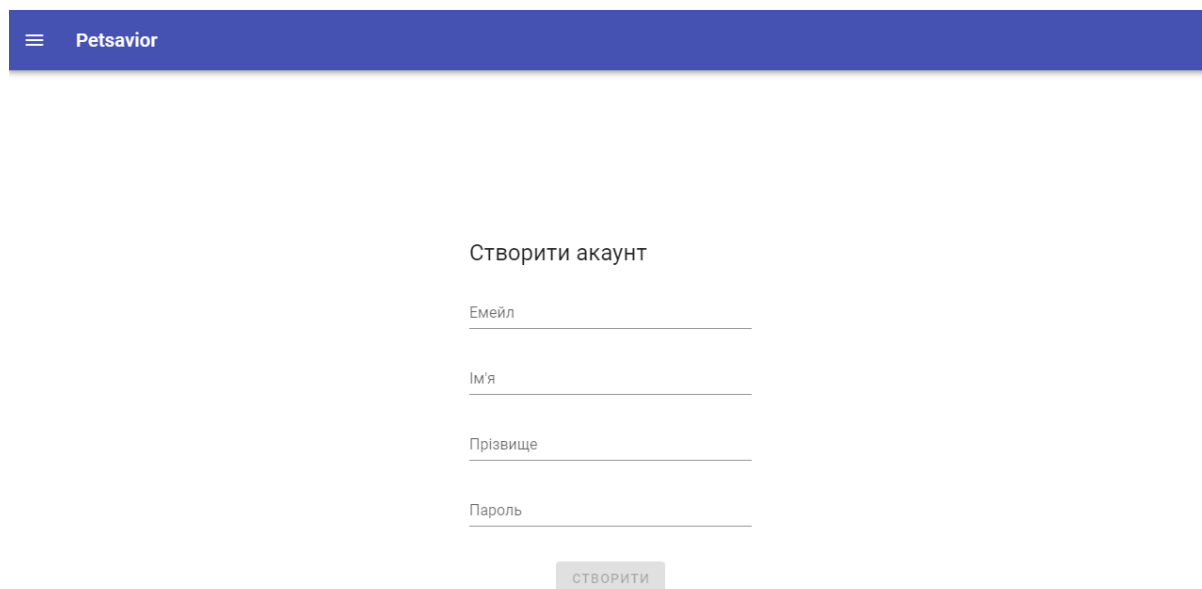


Рис. 3.2. Головна сторінка веб-додатку для пошуку домашніх тварин

В якості вмісту було додано елемент заголовку *h5* та два елементи параграфу *p*, що містить гіперпосилання на сторінки пошуку знайдених домашніх тварин і створення оголошення.

Слідом було реалізовано сторінку реєстрації користувача *PageRegistration* (рис. 3.3).



Створити акаунт

Емейл

Ім'я

Прізвище

Пароль

СТВОРИТИ

Рис. 3.3. Сторінка реєстрації у веб-додатку для пошуку домашніх тварин

Вона складається із заголовку і компоненту форми. Всередині компоненту форми розміщено текстові поля для вводу електронної пошти, імені, прізвища та паролю, а також неактивну кнопку відправки форми. Кожен компонент текстового поля має вхідний параметр *label*, що приймає значення строкового типу, яке буде відображатися в якості назви поля, а також обробники на подію *input*, що відповідають за валідацію вмісту та збереження введеного значення. Усі поля форми валідуються на заповнення, а в полі електронної пошти має також бути введена валідна електронна пошта (рис. 3.4).

Емейл
ф
Невалідний email

Ім'я
Ім'я є обов'язковим

Прізвище
Прізвище є обов'язковим

Пароль
Пароль є обов'язковим

Рис. 3.4. Валідації текстових полів у формі реєстрації

Якщо всі валідації пройдено, то значення вхідного параметру *disabled* компоненту кнопки перемикається з *true* на *false*, що робить кнопку активною. При натисканні лівою кнопкою миші по кнопці на компоненті формі виникає подія *submit*. Обробник цієї події відправляє вміст форму у *POST*-запиті на ендпоінт */registration API*-серверу. Там введена електронна пошта зіставляється з електронними поштами акаунтів, збережених в таблиці *users* бази даних, і, якщо акаунту з такою поштою не існує, у цій таблиці створюється запис нового акаунту. Сервер у відповіді передає дані цього акаунту і токен, за допомогою якого відбувається вхід. Після цього користувача перенаправляє на головну сторінку. У протилежному ж випадку сервер відповідає помилкою, яку оброблює клієнтська частина, показуючи відповідне повідомлення (рис. 3.5).

Емейл
andrewkarp00@gmail.com

Ім'я
Andrew

Прізвище
Karpenko

Пароль
.....

СТВОРИТИ

Щось пішло не так :(

Рис. 3.5. Повідомлення про серверну помилку при спробі зареєструватися

Після реалізації реєстрації розроблено функцію виходу. Для цього до відповідної кнопки усередині бічного меню було додано обробник *logout* на подію *click*, який видаляє сесію і, як наслідок, спричинює вихід з акаунту.

Згодом було створено сторінку входу *PageAuthorization* (рис. 3.6).

☰ Petsavior

Війти в акаунт

Емейл

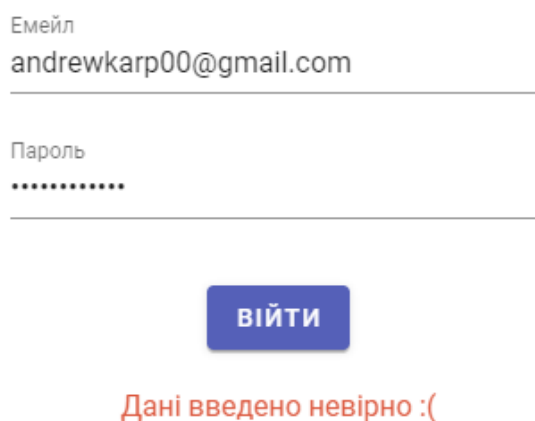
Пароль

ВІЙТИ

Рис. 3.6. Сторінка входу у веб-додатку для пошуку домашніх тварин

Вона, як і сторінка реєстрації, складається з заголовку і компоненту форми. Компонент форми містить текстові поля для вводу електронної пошти й паролю, а

також неактивну кнопку відправки. Кожен компонент текстового поля має вхідний параметр *label*, що приймає значення строкового типу, яке буде відображатися в якості назви поля, і обробники на подію *input*, що відповідають за збереження введеного значення. Стан кнопки змінюється з неактивного на активний після заповнення усіх полів. Натискання лівою кнопкою миші по кнопці викликає подію *submit* на компоненті форми, при обробці якої вміст форми відправляється у *POST*-запиті на ендпоінт */authentication* API-серверу. На сервері відбувається валідація даних для входу. Якщо вона проходить успішно, то сервер створює сесію і передає у відповіді дані акаунту й токен для входу, після чого на клієнтській частині відбувається перенаправлення користувача на головну сторінку. Інакше ж виводиться повідомлення про помилку (рис. 3.7).



The image shows a login form with two input fields. The first field is labeled 'Емейл' (Email) and contains the text 'andrewkarp00@gmail.com'. The second field is labeled 'Пароль' (Password) and contains a series of dots. Below the fields is a blue button with the text 'ВІЙТИ' (Log In). Below the button is a red error message: 'Дані введено невірно :((Data entered is incorrect :(

Рис. 3.7. Повідомлення про серверну помилку при спробі входу до акаунту

Далі було реалізовано сторінку створення оголошень *PageCreateAd*. Вона містить елемент заголовку *h5* з текстом «Створення оголошення» і елемент форми *form*. Форма містить компоненти радіо-кнопок *VRadio*, що відповідають за вибір типу оголошення, компоненти текстових полів для вводу заголовку, опису оголошення і номеру телефону автора, компоненти селектів, що відповідають за вибір локації оголошення і виду домашньої тварини, компонент кнопки прикріплення фото, що буде використано в якості фото оголошення. Доступні локації та види тварин отримуються у відповіді API-сервера на *GET*-запити до ендпоінтів */cities* і */pets* відповідно. За замовчуванням обрано тип оголошень для

загублених домашніх тварин, першу локацію і перший вид домашньої тварини зі списку. Кожен компонент текстового поля має вхідний параметр *label*, що приймає значення строкового типу, яке буде відображатися в якості назви поля, а також обробники на подію *input*, що відповідають за валідацію вмісту та збереження введеного значення. Усі текстові поля форми є обов'язковими до заповнення, також обов'язково має бути прикріплене фото оголошення. До того ж заголовок оголошення має складатися мінімум із десяти й максимум із п'ятдесяти символів, опис оголошення повинен мати до двохсот символів, номер телефону повинен мати десять символів і бути валідним. Якщо всі валідації пройдено, то стан кнопки відправки форми змінюється з неактивного на активний. При натисканні лівою кнопкою миші по ній на формі виникає подія *submit*. На цю подію встановлено обробник *create*, що надсилає вміст форми у *POST*-запиті на ендпоінт */createAdvertisementLost* або */createAdvertisementFound* API-серверу в залежності від обраного типу оголошення. В результаті створюється нове оголошення, яке зберігається в таблицю *announcements* бази даних, а користувача перенаправляє або на сторінку пошуку загублених тварин, або на сторінку пошуку знайдених тварин в залежності від обраного типу оголошення.

На сторінці пошуку оголошень зниклих домашніх тварин *PageLostAds* розміщено компоненти пошукової форми *SearchForm* і блоку карточок оголошень *AdList*. Компонент пошукової форми містить компоненти текстового поля *VTextField* для вводу пошукового запити, кнопки відправки *VBtn*, селектив-фільтрів *Filters* за видом домашньої тварини й локацією оголошення. При завантаженні сторінки відбуваються *GET*-запити до API-серверу на ендпоінти */pets*, */cities* і */getAllAdvertisementLost*. Запит до */pets* повертає масив опцій для селекта-фільтру за видом домашньої тварини, запит до */cities* – масив опцій для селекта-фільтру за локацією, запит до */getAllAdvertisementLost* – список усіх існуючих в базі даних оголошень зниклих домашніх тварин. Оголошення зі списку відображаються у вигляді компонентів карточок *AdCard* усередині компоненту блоку карточок оголошень *AdList*. Кожен компонент карточки *AdCard* приймає в якості вхідного параметру об'єкт оголошення і відображає його фото

або картинку-плейсхолдер в разі відсутності фото, заголовок, підзаголовок, в якому вказується вид зниклої домашньої тварини, і футер, усередині якого вказано локацію і дату публікації оголошення. Заголовок містить гіперпосилання на сторінку повного перегляду оголошення. Компонент пошукової форми *SearchForm* відповідає за уточнення результатів пошуку. При натисканні лівою кнопкою миші по кнопці відправки або при зміні активної опції будь-якого із селектив-фільтрів на формі виникає подія *submit* і спрацьовує її обробник *search*, що повторно відправляє *GET*-запит до *API*-серверу на ендпоінт */getAllAdvertisementLost* із застосуванням фільтрів, які передаються до нього у вигляді *query*-параметрів. На сервері формується список оголошень зниклих домашніх тварин, що задовольняє умовам пошуку. Після отримання цього списку компонент блоку карточок оголошень *AdList* перерисовується, заповнюючись карточками актуальних оголошень.

На сторінці пошуку оголошень знайдених домашніх тварин *PageFoundAds* розміщено компоненти пошукової форми *SearchForm* і блок карточок оголошень *AdList*. Компонент пошукової форми *SearchForm* містить компоненти текстового поля *VTextField* для вводу пошукового запиту, кнопки відправки *VBtn*, селектив-фільтрів *Filters* за видом домашньої тварини й локацією оголошення. При завантаженні сторінки відбуваються *GET*-запити до *API*-серверу на ендпоінти */pets*, */cities* і */getAllAdvertisementFound*. Запит до */pets* повертає масив опцій для селекта-фільтру за видом домашньої тварини, запит до */cities* – масив опцій для селекта-фільтру за локацією, запит до */getAllAdvertisementFound* – список усіх існуючих в базі даних оголошень знайдених домашніх тварин. Оголошення зі списку відображаються у вигляді компонентів карточок *AdCard* усередині компоненту блоку карточок оголошень *AdList*. Кожен компонент карточки *AdCard* приймає в якості вхідного параметру об'єкт оголошення і відображає його фото або картинку-плейсхолдер в разі відсутності фото, заголовок, підзаголовок, в якому вказується вид знайденої домашньої тварини, і футер, усередині якого вказано локацію і дату публікації оголошення. Заголовок містить гіперпосилання на сторінку повного перегляду оголошення. Компонент пошукової форми

SearchForm відповідає за уточнення результатів пошуку. При натисканні лівою кнопкою миші по кнопці відправки або при зміні активної опції будь-якого з селектив-фільтрів на формі виникає подія *submit* і спрацьовує її обробник *search*, що повторно відправляє *GET*-запит до *API*-серверу на ендпоінт */getAllAdvertisementFound* із застосуванням фільтрів, які передаються до нього у вигляді *query*-параметрів. На сервері формується список оголошень знайдених домашніх тварин, що задовольняє умовам пошуку. Після отримання цього списку компонент блоку карточок оголошень *AdList* перерисовується, заповнюючись карточками актуальних оголошень.

Сторінка пошуку оголошень за фото *PagePhotoMatching* складається з компонентів пошукової форми *SearchForm* і блоку карточок оголошень *AdList*. Початково блок карточок оголошень пустий. Усередині пошукової форми розташовано компоненти радіо-кнопок *VRadio* для вибору типу оголошень, кнопки прикріплення фото оголошення *VFileInput* і кнопки відправлення форми *VBtn*. За замовчуванням обрано тип оголошень зниклих домашніх тварин. Кнопка відправлення є неактивною до моменту прикріплення фото. При натисканні лівою кнопкою миші по ній на формі виникає подія *submit*. Спрацьовує обробник *onSubmit*, який відправляє *POST*-запит з прикріпленим фото до *API*-серверу на ендпоінт */similarityLost*, якщо обрано тип оголошення зниклих домашніх тварин, або на ендпоінт */similarityFound*, якщо обрано тип оголошення знайдений домашніх тварин. На сервері відбувається порівняння прикріпленого фото з фото усіх оголошень і створюється список оголошень, фото яких співпадає з прикріпленим більш, ніж на п'ятдесят відсотків. Після отримання цього списку компонент блоку карточок оголошень *AdList* формує компоненти карточок актуальних оголошень *AdCard*. Кожен компонент карточки *AdCard* приймає в якості вхідного параметру об'єкт оголошення і відображає його фото або картинку-плейсхолдер в разі відсутності фото, заголовок, підзаголовок, в якому вказується вид домашньої тварини, і футер, усередині якого вказано локацію і дату публікації оголошення. Заголовок містить гіперпосилання на сторінку повного перегляду оголошення.

Сторінка повного перегляду оголошення *PageAd* у своїй *URL*-адресі містить *id* оголошення, про яке потрібно показати повну інформацію. За цим *id* при завантаженні сторінки до *API*-серверу надсилається *GET*-запит на ендпоінт */getAdvertisementById*. Сервер знаходить відповідне оголошення у базі даних і повертає його у своїй відповіді. Після отримання відповіді на сторінці відображаються фото оголошення або картинка-плейсхолдер у разі його відсутності, заголовок оголошення, дата публікації оголошення, опис оголошення, повне ім'я і номер телефону автора оголошення.

3.4. Висновки до розділу

Початковим етапом розробки веб-додатку для пошуку домашніх тварин було налаштування середовища розробки. В якості середовища розробки обрано *Visual Studio Code*. Сам процес розробки відбувався в операційній системі *Windows* на платформі *Node.js*. Також в ході розробки було застосовано пакетний менеджер *npm*, який є складовою частиною платформи *Node.js*, а також інтерфейс командного рядку *Vue CLI*.

Для створення проекту веб-додатку для пошуку домашніх тварин було використано: пакетний менеджер *npm* платформи *Node.js*, термінал інтегрованого середовища розробки *Visual Studio Code*, інтерфейс командного рядку *Vue CLI*, мови програмування *JavaScript* і *Java*.

Розробка веб-додатку для пошуку домашніх тварин включала в себе наступні етапи: створення базової розмітки; створення модуля аутентифікації разом із сторінками реєстрації *PageRegistration* і входу *PageAuthorization*; реалізації функції створення оголошення і відповідної сторінки *PageCreateAd*; реалізації функції пошуку оголошень разом зі створенням сторінок пошуку зниклих домашніх тварин *PageLostAds*, пошуку знайдених домашніх тварин *PageFoundAds*, пошуку за фото *PagePhotoMatching*; створення сторінки перегляду оголошення *PageAd*.

Проект було виконано з дотриманням діючих стандартів та положень [18], [19].

ВИСНОВКИ

Пошук даних – це процес знаходження та обробки потрібної інформації або встановлення її відсутності у певному наборі даних. Для пошуку даних на сайтах використовують пошукові системи. Пошукова система – алгоритми, реалізовані сукупністю комп'ютерних програм, що надають користувачеві можливість швидкого доступу до необхідної інформації за допомогою пошуку в великій колекції доступних даних.

Пошук можна застосовувати для структурованих та неструктурованих даних. Структуровані дані – це дані, які відповідають моделі даних, мають чітко визначену структуру, послідовний порядок, є доступними та можуть бути використані людиною або комп'ютерною програмою. Неструктуровані дані – це дані, які не відповідають моделі даних і не мають чітко ідентифікованої структури, тому не можуть використані комп'ютерною програмою без попередньої структуризації.

Online-сервіси пошуку домашніх тварин – програмне забезпечення, створене з метою допомоги людям у знаходженні або передачі їхніх домашніх улюбленців. Серед існуючих *online*-сервісів пошуку домашніх було розглянуто *Petfinder*, *RSPCA* й *Adopt a Pet*.

При проектуванні веб-додатку для пошуку домашніх тварин було визначено наступні функції:

- аутентифікація (реєстрація та вхід);
- створення оголошень про загублених або зниклих домашніх тварин;
- пошук оголошень про загублених або зниклих домашніх тварин;
- перегляд оголошень про загублених або зниклих домашніх тварин.

При проектуванні бази даних веб-додатку для пошуку домашніх тварин було використано реляційну систему управління базами даних *PostgreSQL*. Усього під час проектування бази даних веб-додатку для пошуку домашніх тварин було створено чотири таблиці бази даних. Це таблиці *users*, *announcements*, *cities* і *pets*.

Інтерфейс веб-додатку для пошуку домашніх тварин має бути зручним для використання різними категоріями людей на різних пристроях та у різних середовищах. Для цього при його проектуванні має бути дотримано наступні принципи:

- доступність;
- адаптивність;
- кросбраузерність.

Інтерфейс веб-додатку для пошуку домашніх тварин складається із головної сторінки, сторінок реєстрації, входу, створення оголошення, пошуку оголошень зниклих домашніх тварин, пошуку оголошень знайдених домашніх тварин, пошуку оголошень за фото, перегляду конкретного оголошення.

Початковим етапом розробки веб-додатку для пошуку домашніх тварин було налаштування середовища розробки. В якості середовища розробки обрано *Visual Studio Code*. Сам процес розробки відбувався в операційній системі *Windows* на платформі *Node.js*. Також в ході розробки було застосовано пакетний менеджер *npm*, який є складовою частиною платформи *Node.js*, а також інтерфейс командного рядку *Vue CLI*.

Для створення проекту веб-додатку для пошуку домашніх тварин було використано: пакетний менеджер *npm* платформи *Node.js*, термінал інтегрованого середовища розробки *Visual Studio Code*, інтерфейс командного рядку *Vue CLI*, мови програмування *JavaScript* і *Java*.

Розробка веб-додатку для пошуку домашніх тварин включала в себе наступні етапи: створення базової розмітки; створення модуля аутентифікації разом із сторінками реєстрації *PageRegistration* і входу *PageAuthorization*; реалізації функції створення оголошення і відповідної сторінки *PageCreateAd*; реалізації функції пошуку оголошень разом зі створенням сторінок пошуку зниклих домашніх тварин *PageLostAds*, пошуку знайдених домашніх тварин *PageFoundAds*, пошуку за фото *PagePhotoMatching*; створення сторінки перегляду оголошення *PageAd*.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дональд Кнут . Мистецтво програмування, том 3. Сортування і пошук = The Art of Computer Programming, vol.3. Sorting and Searching. — 2-е вид. — М.: «Уільямс», 2007. — С. 824.
2. Ланде Д. В. Пошук знань в Internet. — М.: Діалектика, 2005. — 272 с.
3. Chu H., Rosenthal M. Search engines for the World Wide Web: A comparative study and evaluation methodology // Proceedings of the Annual Meeting of the American Society for Information Science: journal. — 1996. — том 33. — с. 127—135.
4. What is Structured Data? [Електронний ресурс]. — Режим доступу: <https://www.geeksforgeeks.org/what-is-structured-data/> (дата звернення 19.05.2022).
5. What is Unstructured Data? [Електронний ресурс]. — Режим доступу: <https://www.geeksforgeeks.org/what-is-unstructured-data/> (дата звернення 19.05.2022).
6. Роберт Седжвік. Фундаментальні алгоритми на С. Аналіз/Структури даних/Сортування/Пошук = Algorithms in C. Fundamentals/Data Structures/Sorting/Searching. — СПб.: ДіаСофтЮП, 2003. — С. 672.
7. Petfinder [Електронний ресурс]. — Режим доступу: <https://www.petfinder.com/> (дата звернення 20.05.2022).
8. RSPCA [Електронний ресурс]. — Режим доступу: <https://www.rspca.org.uk/> (дата звернення 20.05.2022).
9. Adopt a Pet [Електронний ресурс]. — Режим доступу: <https://www.adoptapet.com/> (дата звернення 20.05.2022).
10. Конноллі Т., Бегг К. Бази даних. Проектування, реалізація і супроводження. Теорія і практика = Database Systems: A Practical Approach to Design, Implementation, and Management. — 3-е вид. — М.: Уільямс, 2003. — 1436 с.
11. What is accessibility? [Електронний ресурс]. — Режим доступу: https://developer.mozilla.org/en-US/docs/Learn/Accessibility/What_is_accessibility (дата звернення 23.05.2022).

12. HTML5 – Responsive Web Design [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-us/archive/msdn-magazine/2011/november/html5-responsive-web-design> (дата звернення 23.05.2022).
13. Facts About W3C [Електронний ресурс]. – Режим доступу: <https://www.w3.org/> (дата звернення 23.05.2022).
14. Visual Studio Code [Електронний ресурс]. – Режим доступу: <https://code.visualstudio.com/> (дата звернення 26.05.2022).
15. Node.js [Електронний ресурс]. – Режим доступу: <https://nodejs.org/> (дата звернення 26.05.2022).
16. Vue CLI [Електронний ресурс]. – Режим доступу: <https://cli.vuejs.org/> (дата звернення 26.05.2022).
17. Vue.js [Електронний ресурс]. – Режим доступу: <https://v2.vuejs.org/> (дата звернення 26.05.2022).
18. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення / Держстандарт України. – Вид. офіц. – [Чинний від 1995-02-23]. – Київ, 2007. – 86с.
19. Слободян О. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: Видавництво НАУ, 2017. – 63с.