МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
Кафедра Ком'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
Аліна САВЧЕНКО
«_____»_____2023р.

КВАЛІФІКАЦІЙНА РОБОТА
(ДИПЛОМНИЙ ПРОЄКТ, ПОЯСНЮВАЛЬНА ЗАПИСКА)
ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ **"БАКАЛАВР"**
ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ
"ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ"

Тема: «WEB-додаток "Допомога тваринам" із застосуванням CMS WordPress»

Виконавиця: студентка групи УС-412Б Коваленко Оксана Василівна
Керівник: к.н.т., доцент Моденов Юрій Борисович
Нормоконтролер: Олександр ШЕВЧЕНКО

Київ 2023

НАЦІЩНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних наук і технологій*

Кафедра *Комп'ютерних інформаційних технологій*

Галузь знань, спеціальність, освітньо-професійна програма: 12 "Інформаційні технології", 122 "Комп'ютерні науки", "Інформаційні управляючі системи та технології"

ЗАТВЕРДЖУЮ

Завідувач випускової кафедри

_____Аліна САВЧЕНКО

«___»_____2023р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Коваленко Оксани Василівни

1. **Тема роботи:** «WEB-додаток "Допомога тваринам" із застосуванням CMS WordPress» затверджена наказом ректора від «01» травня 2023 р. №623/ст.
2. **Термін роботи:** з 15.05.2023р. по 16.06.2023р.
3. **Вихідні дані роботи:** демонстративна версія веб-додатку "Допомога тваринам".
4. **Зміст пояснювальної записки:** огляд та аналіз предметної області, аналіз та порівняння обраних тезнологій для WEB-додатку, опис прототипу WEB-додатку "Допомога тваринам" із застосуванням CMS WordPress.
5. **Перелік обов'язкового ілюстративного матеріалу:** слайди презентації MS PowerPoint.

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1	Аналіз та дослідження предметної області.	15.05.2023р.-17.05.2023р.	
2	Огляд та аналіз технологій та фреймворків.	18.05.2023р.-24.05.2023р.	
3	Проектування рівнів web-сервісу. Написання першого розділу кваліфікаційної роботи.	25.05.2023р.-01.06.2023р.	
4	Розробка web-сервісу з використанням оглянутих технологій. Написання другого розділу кваліфікаційної роботи.	01.06.2023р.-05.06.2023р.	
5	Написання третього розділу кваліфікаційної роботи. Завершення створення та оформлення пояснювальної записки.	06.06.2023р.-12.06.2023р.	
6	Підготовка презентації та доповіді.	13.06.2023р.-16.06.2023р.	

7. Дата видачі завдання: «15» травня 2023р.

Керівник дипломної роботи Юрій МОДЕНОВ

Завдання прийняла до виконання Оксана КОВАЛЕНКО

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи на тему: **«Волонтерський Веб-додаток «Допомога тваринам» з застосуванням CMS WordPress»** містить: 48 сторінок, 14 рисунків, 18 інформаційних джерел.

Об’єкт дослідження – веб-додатки та їх можливості, мова програмування JavaScript та її бібліотека React.

Предмет дослідження – веб-додаток для допомоги тваринам.

Мета дипломної роботи – розглянути актуальність розробки веб-додатків, отримати прототип-додатку для допомоги тваринам.

Методи дослідження – мова програмування JavaScript, CSS, CMS WordPress

Результати кваліфікаційної роботи рекомендується використовувати для створення власного проекту для для допомоги тваринам.

Для розробки веб-додатку використано такі мови як програмування JavaScript, HTML, CSS а також CMS WordPress.

WEB-ДОДАТОК, ВЕБ-ДОДАТОК, ТВАРИНИ, ДОПОМОГА ТВАРИНАМ, РОЗРОБКА, CSS, HTML, JAVASCRIPT, ПРОЕКТУВАННЯ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ 6
ВСТУП 7
РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 9
1.1. Аналіз предметної області застосування 9
1.2. Переваги та недоліки веб-додатків 10
1.2.1. Переваги створення веб-додатків 10
1.2.2. Недоліки створення веб-додатків 11
1.3. Класифікація веб-додатків 12
1.4. Архітектура та принципи роботи веб-додатку 16
1.5. Порівняння сайтів аналогів 18
1.6. Висновок до розділу 1 22
РОЗДІЛ 2. АНАЛІЗ ТА ПОРІВНЯННЯ ОБРАНИХ ТЕХНОЛОГІЙ ДЛЯ WEB-ДОДАТКУ 25
2.1. Розробка макету 25
2.2. HTML 34
2.3. CSS 36
2.4. JavaScript 37
2.5. Висновок до розділу 2 40
РОЗДІЛ 3. ОПИС ПРОТОТИПУ WEB-ДОДАТКУ «ДОПОМОГА ТВАРИНАМ» З ЗАСТОСУВАННЯМ CMS WORDPRESS 42
3.1. Глобальний дизайн інтерфейсу веб-додатку та навігація 42
3.2. Розробка інтерфейсу сторінок 44
3.3. Створення макету 46
3.4. Верстка макету 51
3.5. Висновок до розділу 3 55
ВИСНОВКИ 57
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ 58
ДОДАТКИ 61

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

HTML – HyperText Markup Language

CSS – Cascade Style Sheets

HTTP – HyperText Transfer Protocol

JS – JavaScript

URL – Uniform Resource Locator

W3C – World Wide Web Consortium

CDN – Content Delivery System

SPDY – Speedy

PDF – Portable Document Format

DNS – Domain Name System

TCP – Transmission Control Protocol

UI – User Interface

UX – User Experience

SQL – Structured Query Language

IP – Internet Protocol

ВСТУП

Є багато веб-додатків, які допомагають тваринам. Ось декілька прикладів:

1. Petfinder - це додаток для пошуку та прийняття домашніх тварин. Він має базу даних з тисячами тварин, які шукають нового власника.
2. ASPCA - це додаток від Американського товариства захисту тварин, який надає інформацію про догляд за тваринами, поради щодо їх виховання та здоров’я.
3. iNaturalist - це додаток для збору інформації про дикі тварини. Він дозволяє користувачам знімати фотографії тварин, визначати їх вид та ділитися своїми спостереженнями з іншими користувачами.
4. HappyCow - це додаток для вегетаріанців та веганів, який допомагає знайти ресторани та кафе з вегетаріанською та веганською кухнею. Цей додаток допомагає зменшити попит на м’ясні продукти та сприяє зменшенню страждання тварин.
5. BarkHappy - це додаток для власників собак, який дозволяє знаходити друзів для своєї собаки та зустрічатися з іншими власниками собак для прогулянок та інших активностей.

Ці додатки допомагають людям доглядати за тваринами, знаходити нових власників для домашніх тварин та підтримувати інтерес до дикої природи та її збереження.

Зараз все більше уваги приділяється безпеці веб-додатків. Безпека web-додатків відрізняється від захисту традиційного ПЗ.

З усього вище сказаного однозначно зрозуміла актуальність питання дослідження Веб-додатків та використання новітніх технологій.

В рамках кваліфікаційної роботи перед нами поставлено завдання розробити Web-додаток для допомоги тваринам.

Мета роботи. Проаналізувати специфіку діяльності Веб-додатків та взагалі, їх відмінності між собою, розглянути новітні засоби в розробці Веб-додатків, дізнатись більше про мови програмування та новітні фреймворки, розробити макет додатку та розробити Веб-додаток для допомоги тваринам.

Завдання на кваліфікаційну роботу:

1. провести аналіз предметної області використання веб-додатків;
2. провести порівняння веб-додатків аналогів, виявити їх недоліки та переваги;
3. розглянути особливості сучасних програмних засобів по розробці Web-додатків використовуючи мови: HTML, CSS, JavaScript;
4. розробити дизайн веб-додатку;
5. розробити Веб-додаток для допомоги тваринам.

Об'єктом дослідження є проектування та розробка веб-додатку для допомоги тваринам.

Предметом дослідження є процес розробки веб-додатків

Для досягнення поставленої мети й виконання завдань застосовано сучасні технології та засоби розробки веб-застосунків, зокрема стек HTML/CSS/JS.

Веб-додаток складатиметься з головної сторінки, на якій зображатиметься коротко інформація стосовно додатку та чим він займається, логотип додатку та кнопка для відкривання меню даного веб-додатку. В меню можна обрати сторінку на яку саме ви хочете перейти, а саме : «Про нас» - сторінка з інформацією про додаток, перелік розділів, «Контакти» - інформація з номерами телефонів, а також посилання на сторінки в соціальних мережах.

РОЗДІЛ 1
ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз предметної області застосування

Веб-додаток - це комплекс веб-сторінок, які надають користувачеві необхідні функції та можливості, що виконуються через веб-браузер як клієнта. Веб-додатки можуть працювати на будь-яких платформах або пристроях, що підтримують веб-браузер.

Веб-додатки можуть бути розроблені з використанням різноманітних технологій, таких як HTML, CSS, JavaScript, PHP, Python тощо. Вони можуть мати різну функціональність, від простих статичних сторінок до складних додатків, що включають в себе бази даних, аналітичні інструменти, можливості для взаємодії з користувачем тощо.

Переваги веб-додатків полягають у їх доступності та універсальності, оскільки користувачі можуть використовувати їх на будь-якому пристрої з веб-браузером та підключенням до Інтернету. Крім того, веб-додатки можуть бути легко оновлені та розширені без необхідності встановлювати додаткові програми на пристрій користувача.

1.2. Переваги та недоліки веб-додатків

1.2.1. Переваги створення веб-додатків

Існує багато переваг в створенні веб-додатків, ось деякі з них:

1) доступність: веб-додатки можуть бути доступні з будь-якого пристрою з підключенням до Інтернету та веб-браузером, що дозволяє користувачам легко отримувати доступ до додатків з будь-якого місця та на будь-якому пристрої;
2) універсальність: веб-додатки не залежать від операційної системи або конкретного пристрою, що дозволяє користувачам використовувати їх на будь-якій платформі або пристрої;
3) легка оновлення: оновлення веб-додатків можуть бути виконані централізовано на сервері, що забезпечує легкість і швидкість оновлення;
4) масштабованість: веб-додатки можуть бути легко масштабовані залежно від потреб користувачів та збільшення кількості користувачів;
5) низькі вимоги до системних ресурсів: веб-додатки не вимагають великої кількості системних ресурсів на пристрої користувача, оскільки більшість ресурсів обробки відбувається на сервері;
6) низькі витрати на розробку та обслуговування: порівняно зі створенням настільних додатків, розробка та обслуговування веб-додатків може бути менш коштовною;
7) доступ до багатьох користувачів: веб-додатки можуть бути доступні для великої кількості користувачів з різних частин світу, що дозволяє досягати більшої аудиторії та ринків.

Веб-додатки зазвичай доступні через веб-браузер на будь-якому пристрої з підключенням до Інтернету, включаючи комп'ютери, смартфони, планшети та інші пристрої. Це дозволяє користувачам отримувати доступ до додатків на будь-якому пристрої та з будь-якого місця, де є доступ до Інтернету. Крім того, веб-додатки не

потребують встановлення на пристрій користувача, що робить їх ще більш доступними та зручними для використання.

Отже, створення веб-додатків може бути вигідним рішенням для компаній та організацій.

HTML та CSS дійсно є основними мовами розмітки та стилізації веб-сторінок, що використовуються для розробки веб-додатків. Однак, веб-додатки зазвичай включають у себе інші технології, такі як JavaScript, для створення взаємодії з користувачем та динамічної поведінки веб-сторінки. Крім того, веб-додатки можуть використовувати різноманітні серверні технології, такі як PHP, Ruby, Python тощо, для обробки запитів користувачів та доступу до баз даних.

Отже, для розробки веб-додатків необхідно мати знання не тільки HTML та CSS, але й інших технологій та мов програмування, залежно від вимог та функціональності додатка. Успішні веб-додатки також вимагають досвіду у проектуванні, розробці та тестуванні, а також знань з питань безпеки та оптимізації продуктивності веб-додатку.

1.2.2. Недоліки створення веб-додатків

Більшість сучасних інформаційних систем створюються як Web-сайтів, тому безпеки необхідно приділити особливу увагу. Як тільки з'являється пристрій з операційною системою та програмним додатком, він одразу ж приваблює інтерес зловмисників. Вони починають його вивчати та зламувати. В останній рік цій тенденції присвячено багато доповідей на великих конференціях хакерів. Розробники не завжди приділяють особливу увагу повному захисту та безпеці створених сайтів, хоча вона і відіграє дуже важливу роль, тому що кожен сайт так чи інакше може бути підданий злому хакерів, якщо враховувати те, що сайти можуть бути, щонайменше, різноманітні : електронні бібліотеки, соціальні мережі, Web-портали різних навчальних закладів, онлайн-магазини, офіційні сайти будь-яких організацій, банків та інше. Насамперед, необхідно звернути увагу на безпеку веб-сервера, оскільки саме він відповідає за прийом та обробку HTTP-запитів від клієнтів до веб-сайту.

Досить важливим моментом є те, що географічне розташування сервера аж ніяк не впливає на його захист. Здійснити атаку на нього можна з будь-якої точки доступу. Це пов'язано з тим, що Web-сервери в силу своєї відкритості розраховані на передачу інформації між користувачами і тому мають безліч уразів мостів. Наприклад, зловмисник може внести якісь зміни (модифікації) код HTTP сервера чи сервера бази даних, або самих сторінок сайту, змінивши його початкову функціональність.

Захист веб-сайту допомагає захистити ваш веб-сайт від: DDoS-атаки: це шкідлива атака, яка порушує нормальну роботу веб-сайту. Він робить це, перевантажуючи навколишню інфраструктуру веб-сайту непотрібним інтернет-трафіком.

Шкідливі програми: використовується для розповсюдження спаму, крадіжки конфіденційної інформації клієнтів та отримання несанкціонованого доступу до сайту.

У випадку з браузерними додатками також можуть виникнути проблеми з сумісністю, особливо якщо браузер чи його версія не підтримує деякі функції, що використовуються у додатку. Крім того, деякі браузери можуть підтримувати різні версії JavaScript, що може призвести до проблем зі сумісністю, якщо додаток написаний на старій версії цієї мови програмування. Окрім цього, розробники браузерних додатків також повинні турбуватися про підтримку різних платформ та оперативних систем, що може вимагати додаткової роботи з оптимізації та адаптації додатку для різних пристроїв та платформ. Тому, хоча браузерні додатки мають свої переваги, вони також можуть потребувати певних зусиль з боку розробників для забезпечення їх сумісності та доступності на різних пристроях і платформах.

1.3. Класифікація веб-додатків

Існує кілька способів класифікації веб-додатків. Основні з них наступні:

1) а залежності від місця виконання:
    - клієнт-серверні веб-додатки - виконуються на сервері та передають результати до клієнтського браузера;

- клієнтські веб-додатки - виконуються на клієнтському пристрої, зазвичай, за допомогою JavaScript, а сервер використовується для отримання та збереження даних.

Клієнт-серверні веб-додатки є одним з видів веб-додатків, вони виконуються на сервері і взаємодіють з клієнтським браузером через мережу Інтернет. Клієнт-серверна архітектура передбачає, що весь код додатку розташований на сервері, а браузер використовується як інтерфейс для взаємодії з користувачем. При цьому клієнтський браузер відправляє запити до сервера, який оброблює їх і повертає відповіді в браузер. Наприклад, при запиті на відкриття певної веб-сторінки, сервер відправляє браузеру HTML-код цієї сторінки, який потім рендериться в браузері. Додатково, клієнт-серверні додатки можуть використовувати різноманітні протоколи для обміну даними між браузером та сервером, наприклад, HTTP, WebSocket, WebRTC тощо.

Клієнтські веб-додатки виконуються на клієнтському пристрої, зазвичай, за допомогою JavaScript. Цей тип веб-додатків є популярним завдяки тому, що вони працюють безпосередньо в браузері користувача і не потребують постійного звернення до сервера для отримання даних. Клієнтські веб-додатки зазвичай складаються з HTML, CSS та JavaScript-коду, що виконується безпосередньо в браузері користувача. Додатки цього типу зазвичай працюють швидко та ефективно, оскільки не потребують звернення до сервера для отримання даних та виконання операцій. Однак, вони можуть бути обмежені доступом до системних ресурсів та можуть працювати тільки в певних браузерах або на певних платформах.

Клієнтські веб-додатки можуть виконуватися як в автономному режимі (наприклад, додатки для обробки фото або відео, які працюють в браузері без підключення до Інтернету), так і в режимі онлайн (наприклад, онлайн-ігри, соціальні мережі тощо).

2) В залежності від рівня складності:

- статичні веб-додатки - створюються за допомогою HTML та CSS та мають обмежені можливості;

- динамічні веб-додатки - використовуються скрипти на стороні сервера та клієнта, що дозволяє створювати більш складні додатки зі змінним контентом;
- інтерактивні веб-додатки - мають додаткові функції для взаємодії з користувачем, такі як анімація, графічний інтерфейс та інші.

Так, статичні веб-додатки створюються за допомогою HTML та CSS і мають обмежені можливості. Це означає, що вони не можуть взаємодіяти з сервером та змінювати вміст в залежності від дій користувача. Статичні веб-додатки зазвичай використовуються для створення простих веб-сайтів зі статичним вмістом, таких як візитівки, каталоги товарів, розклади та інші сторінки, які не потребують динамічного оновлення. Однак, з появою нових технологій, таких як JavaScript, AJAX та HTML5, можливості статичних веб-додатків значно розширилися. Наприклад, за допомогою JavaScript можна створити веб-сторінки з динамічним вмістом та інтерактивними елементами, такими як віджети, анімація та відео. Крім того, статичні веб-додатки можуть бути корисними для створення простих прототипів та тестування дизайну веб-сторінок до розробки повноцінних динамічних веб-додатків.

Так, динамічні веб-додатки використовують скрипти на стороні сервера та клієнта, щоб забезпечити динамічне оновлення вмісту на веб-сторінках та взаємодію з користувачем. Скрипти на стороні сервера, такі як PHP, Python або Ruby, використовуються для обробки запитів від користувачів та генерації динамічного вмісту для відображення на веб-сторінках. Наприклад, скрипти можуть взаємодіяти з базою даних для отримання даних та генерації вмісту для відображення на веб-сторінках. Скрипти на стороні клієнта, такі як JavaScript, використовуються для взаємодії з користувачем та зміни вмісту на веб-сторінках без повного оновлення сторінки. Наприклад, скрипти можуть створювати інтерактивні елементи, такі як форми, віджети та анімація, які можуть реагувати на дії користувача без необхідності оновлення всієї сторінки. Динамічні веб-додатки використовуються для створення різноманітних веб-сайтів, від простих блогів та інтернет-магазинів до складних соціальних мереж та веб-додатків для корпоративного використання.

Так, інтерактивні веб-додатки мають додаткові функції, що дозволяють користувачеві взаємодіяти з додатком та забезпечують більш зручний та ефективний досвід користування. Наприклад, це можуть бути функції перетягування та перенесення об'єктів, можливість вибору зі списку або календаря, відображення спливаючих вікон та повідомлень, а також багато іншого. Інтерактивні веб-додатки можуть бути створені з використанням різних технологій, таких як HTML, CSS, JavaScript, AJAX, та інших.

3) В залежності від способу використання:
   - публічні веб-додатки - доступні для використання всіма користувачами в Інтернеті;
   - приватні веб-додатки - доступні тільки для певних користувачів, які мають право доступу до конкретної частини додатку.

Так, публічні веб-додатки доступні для використання всіма користувачами в Інтернеті без обмежень або з обмеженим доступом. Наприклад, публічний веб-додаток може бути доступний для користування всім, хто відвідує веб-сайт, на якому він розміщений. Це можуть бути соціальні мережі, пошукові системи, онлайн-магазини та інші веб-додатки, які надають свої послуги широкому колу користувачів.

При створенні публічного веб-додатку необхідно враховувати питання безпеки, доступності та продуктивності, оскільки користувачі з різних місць світу будуть звертатися до додатку з різних пристроїв та мереж, що може вплинути на швидкість та роботу додатку. Також важливо забезпечити адекватний захист від зловмисних атак та забезпечити конфіденційність даних користувачів, що використовують додаток.

Приватні веб-додатки призначені для обмеженого кола користувачів і не доступні загальному публічному доступу. Вони зазвичай мають механізми автентифікації та авторизації, які дозволяють ідентифікувати користувача і контролювати його права доступу до різних ресурсів додатку.

Приватні веб-додатки можуть бути внутрішніми для певної організації, компанії або групи користувачів, які мають спільні потреби і завдання. Наприклад, це можуть бути внутрішні системи управління документами, системи обліку робочого часу або системи управління проектами. Їх особливість полягає в тому, що вони можуть бути

налаштовані під конкретні потреби та вимоги організації і містити конфіденційну інформацію, яку не потрібно розголошувати загальному публіці.

Крім цього, веб-додатки також можна класифікувати за типом діяльності, для якої вони призначені, такі як банківські додатки, соціальні мережі, інтернет-магазини та інші.

1.4. Архітектура та принципи роботи веб-додатку

Архітектура веб-додатку визначається залежно від його функціональності та завдань, які він вирішує. Зазвичай, веб-додатки складаються з трьох компонентів: клієнтської сторони, серверної сторони та бази даних.

Клієнтська сторона включає в себе веб-браузер, який відображає інтерфейс користувача та взаємодіє з сервером. На цій стороні можуть використовуватись такі технології як HTML, CSS та JavaScript для відображення контенту та взаємодії з користувачем.

Клієнтська сторона веб-додатку є тією частиною, яка виконується на боці клієнта - веб-браузера. Вона складається з HTML-сторінок, CSS-стилів та JavaScript-коду. Головною функцією клієнтської сторони є забезпечення інтерфейсу для користувача, взаємодії з ним та передачі даних на сервер. HTML-сторінки відповідають за структуру веб-додатку та відображення контенту. CSS-стилі використовуються для стилізації та оформлення веб-сторінок. JavaScript-код забезпечує інтерактивність та динаміку веб-додатку, яка дає можливість користувачам взаємодіяти з додатком. Клієнтська сторона також відповідає за валідацію введених даних, перевірку правильності заповнення форм, відображення повідомлень про помилки та інші функції, що покращують користувацький досвід. Клієнтська сторона веб-додатку може бути розроблена за допомогою різноманітних фреймворків та бібліотек, таких як Angular, React, Vue.js та інших. Ці інструменти дають можливість розробникам швидко створювати складні інтерфейси та динамічні взаємодії без необхідності писати весь код з нуля. Загалом, клієнтська сторона веб-додатку відіграє

важливу роль у забезпеченні функціональності та користувацького досвіду, і її розробка потребує вміння використовувати різні інструменти та технології.

Серверна сторона забезпечує обробку запитів та передачу даних між клієнтом та базою даних. На цій стороні можуть використовуватись такі технології як PHP, Python, Ruby, Java та інші для розробки логіки додатку та забезпечення зв'язку між клієнтом та базою даних. Серверна сторона веб-додатка відповідає за обробку запитів, збереження та оновлення даних, взаємодію з базою даних, аутентифікацію та авторизацію користувачів та інші бізнес-логічні операції. Основні компоненти серверної сторони веб-додатка включають в себе веб-сервер, базу даних та програмне забезпечення для обробки запитів. Веб-сервер забезпечує обробку запитів користувачів та відправку відповідей у вигляді HTML-сторінок або інших форматів даних. Це може бути програмне забезпечення, таке як Apache або Nginx, або спеціальна платформа, така як Node.js. Програмне забезпечення для обробки запитів виконує бізнес-логіку веб-додатка та взаємодіє з базою даних. Це може бути серверний код, написаний на мовах програмування, таких як PHP, Python, Ruby, Java або C #, або спеціальні фреймворки, такі як Laravel для PHP, Flask для Python або Spring для Java. Окрім цього, серверна сторона може мати компоненти для роботи зі сторонніми сервісами, такими як платіжні системи, сервіси електронної пошти, соціальні мережі тощо. Принципи роботи серверної сторони полягають у обробці запитів від клієнтів та відправці їм відповідей. Клієнтські запити надходять до сервера за допомогою протоколу HTTP, який визначає тип запиту та параметри, необхідні для його виконання.

База даних забезпечує зберігання та організацію даних, необхідних для роботи веб-додатка. Вона може бути реляційною, наприклад MySQL або PostgreSQL, або нереляційною, такою як MongoDB або Cassandra та інші для зберігання даних та взаємодії з серверною стороною.

Принципи роботи веб-додатку полягають у взаємодії між клієнтом та сервером за допомогою протоколу HTTP. Клієнт ініціює запит до сервера, який оброблює цей запит та повертає результат у вигляді веб-сторінки. Клієнт може взаємодіяти з додатком, відправляючи запити до сервера за допомогою кнопок, посилань або форм

на веб-сторінках. Сервер оброблює ці запити та повертає відповіді у вигляді HTML-сторінок

1.5. Порівняння сайтів аналогів

Багато людей хвилюються про бездомних тварин, які живуть на вулицях. Є декілька можливих рішень для допомоги цим тваринам.

1. Зверніться до місцевої організації захисту тварин. Багато міст мають такі організації, які працюють з безпритульними тваринами, забезпечують їм медичну допомогу, харчування та надають тимчасове притулок. Зверніться до цих організацій для отримання поради та допомоги.
2. Допоможіть безпосередньо тваринам. Якщо ви бачите безпритульну тварину, яка потребує допомоги, ви можете надати їй їжу та воду. Також можна купити спеціальну їжу для безпритульних тварин та розмістити її на вулиці, щоб допомогти їм вижити.
3. Розгляньте можливість усиновлення тварини. Багато безпритульних тварин можуть бути дуже добре підготовані для того, щоб стати домашнім улюбленцем. Зв’яжіться з місцевими тваринними притулками, щоб дізнатися, як можна усиновити тварину.
4. Розповідайте про проблему. Розповідайте про безпритульних тварин та проблему, яка з ними пов’язана, своїм друзям, родичам та співробітникам. Це може допомогти збільшити увагу до проблеми та залучити більше ресурсів для її вирішення.
5. Будьте уважними. Намагайтеся не заважати безпритульним тваринам та уникайте небезпечних ситуацій.

Є багато веб-додатків, які допомагають тваринам. Ось декілька прикладів:

1. Petfinder - це додаток для пошуку та прийняття домашніх тварин. Він має базу даних з тисячами тварин, які шукають нового власника.

2. ASPCA - це додаток від Американського товариства захисту тварин, який надає інформацію про догляд за тваринами, поради щодо їх виховання та здоров’я.
3. iNaturalist - це додаток для збору інформації про дикі тварини. Він дозволяє користувачам знімати фотографії тварин, визначати їх вид та ділитися своїми спостереженнями з іншими користувачами.
4. HappyCow - це додаток для вегетаріанців та веганів, який допомагає знайти ресторани та кафе з вегетаріанською та веганською кухнею. Цей додаток допомагає зменшити попит на м’ясні продукти та сприяє зменшенню страждання тварин.
5. BarkHappy - це додаток для власників собак, який дозволяє знаходити друзів для своєї собаки та зустрічатися з іншими власниками собак для прогулянок та інших активностей.

Ці додатки допомагають людям доглядати за тваринами, знаходити нових власників для домашніх тварин та підтримувати інтерес до дикої природи та її збереження.

Точне число безпритульних тварин на території всієї України важко назвати, оскільки воно постійно змінюється. Залежно від сезону та регіону країни, кількість безпритульних тварин може змінюватися значно. Також важливо враховувати, що не всі безпритульні тварини можуть бути зареєстровані в базах даних, тому фактична кількість безпритульних тварин може бути значно вище, ніж офіційна статистика. Однак, важливо звернути увагу на проблему безпритульних тварин та робити все можливе для того, щоб забезпечити їм належний догляд та захист.

На сьогоднішній день платформа Animal-id збирає та поширює актуальні дані, щодо тих тварин, які опинилися на вулицях. Зокрема, станом на 2023 рік максимальна кількість тварин спостерігається в містах: Миколаїв (4503), Кривий Ріг (3645), Київ (3034), Запоріжжя (2918), Дніпро (2358), Одеса (1882), Ізмаїл (1800). Кількість цих тварин також підраховується й волонтерами в рамках проектів.

Даний застосунок має на меті полегшити процес пошуку домівки для безпритульних тварин та надати можливість швидко передавати інформацію про

знайдених тварин до відповідних служб. Додаток повинен бути простим та інтуїтивно зрозумілим для користувачів, щоб забезпечити його широке використання.

Для досягнення цієї мети додаток може мати наступні функції:

- можливість передачі інформації про знайдених тварин: користувачі можуть зробити фотографію знайденої тварини та ввести додаткову інформацію про неї (наприклад, порода, стать, наявність ошийника або інших ідентифікаторів);
- геолокація: додаток може автоматично визначати геолокацію знайденої тварини та передавати цю інформацію разом з іншою інформацією;
- повідомлення до відповідних служб: додаток може надсилати повідомлення до відповідних служб (наприклад, до місцевої служби збереження тварин або до ветеринарної клініки), щоб повідомити про знайдену тварину та запропонувати допомогу;
- система коментарів: користувачі можуть залишати коментарі під постами про знайдених тварин, щоб ділитися інформацією або запитувати додаткову інформацію;
- можливість збору коштів на лікування: користувачі можуть додавати інформацію про тварин, які потребують лікування, та створювати фонд для збору коштів на їх лікування.

Для того, щоб додаток був доступний для всіх, його можна розробити як веб-додаток.

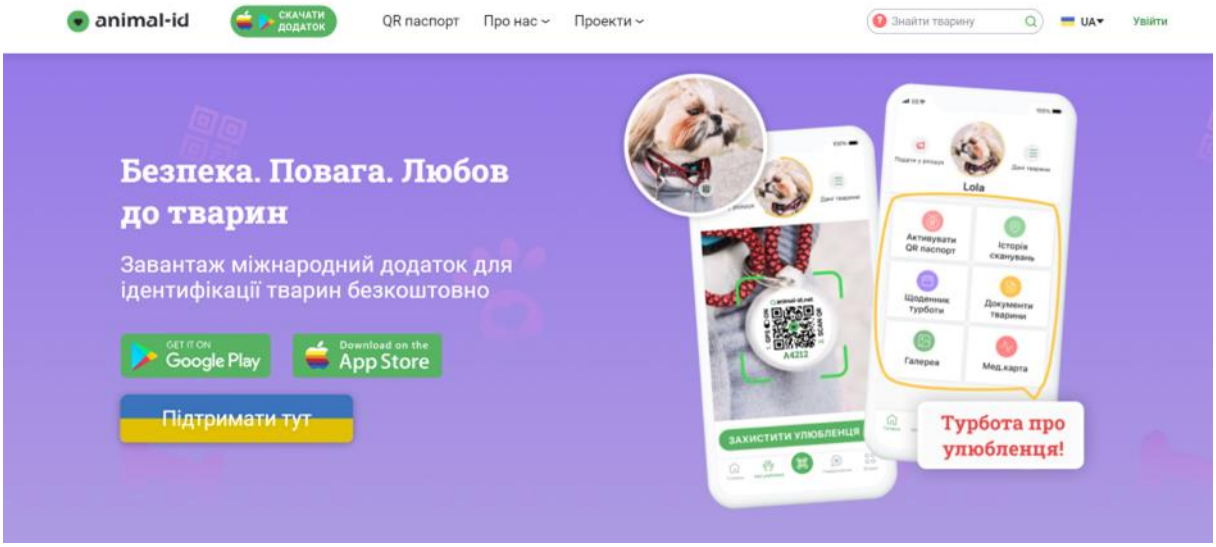

Рис. 1.1. Головна сторінка додатку Animal-id

Такий додаток може стати корисним для багатьох власників тварин, які хочуть забезпечити безпеку та захист своїх улюбленців. Додаток може бути особливо корисним для тих, у кого є тварини, які мають тенденцію втікати, таких як собаки.

Завдяки QR-коду та профілю в додатку, можна швидко та ефективно ідентифікувати тварину, якщо вона загубилась або була викрадена. Це дозволить швидко повернути тварину до її господаря та зменшити травмування та стрес для тварини.

Також, додаток може бути корисним для тих, хто шукає тварину, яка загубилась. Завдяки доступності інформації в додатку, можна швидко знайти власника тварини та повернути його додому.

Однак, для того, щоб такий додаток став ефективним, необхідна велика кількість користувачів та співпраця зі службами, які займаються знаходженням та поверненням загублених тварин. Також, потрібно забезпечити захист конфіденційної інформації власників тварин та їхніх профілів в додатку.

Цей підхід з використанням QR-коду на спеціальному жетоні є дуже ефективним способом ідентифікації тварини та надання необхідної інформації для знаходження власника. Крім того, це також допомагає унеможливити плагіат та фальсифікацію інформації, оскільки інформація є збереженою в світовій базі даних. Крім того, PUSH-повідомлення та email повідомлення можуть допомогти власнику

отримати швидкий та надійний доступ до інформації про загублену тварину. Крім того, доступ до профілю через пошук по мікрочіпу дозволяє швидко знайти тварину та надати допомогу, якщо потрібно. Загалом, такий додаток може стати дуже корисним інструментом для власників тварин та тих, хто допомагає в пошуках загублених тварин.

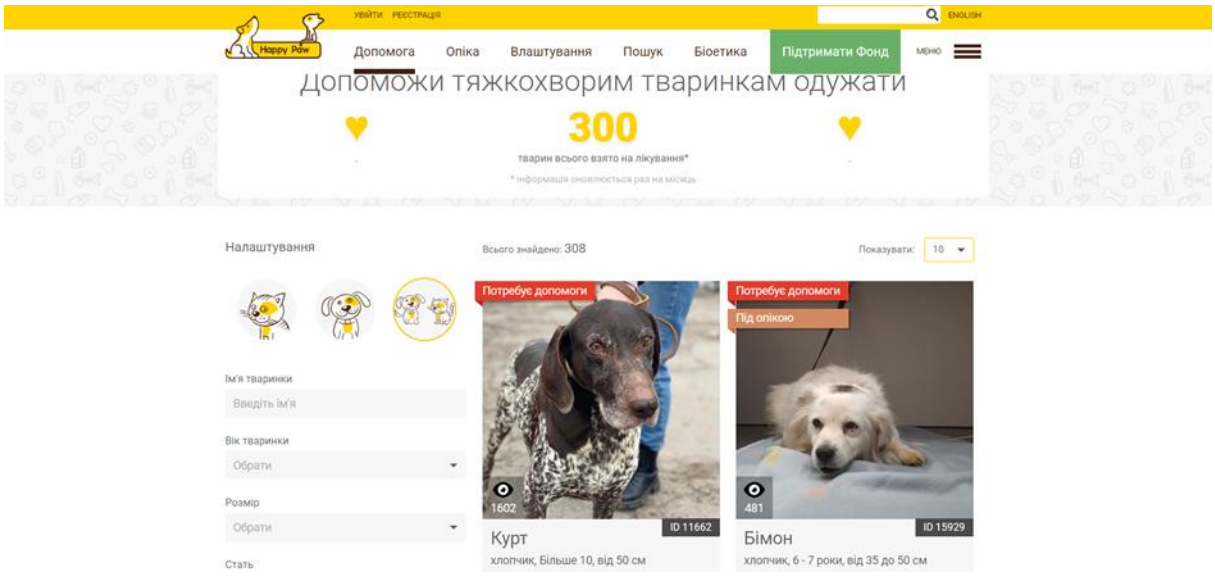

Рис. 1.2. Функції додатку Happy Paw

Додаток Happy Paw – це веб-додаток на підгрунті благодійного фонду з більш ніж десятирічною історією. Він має фугкції донатів, опіки на тваринами, пошуку, надання допомоги, лікування.

1.6. Висновок до розділу 1

1. Веб-додаток - це комплекс веб-сторінок, які надають користувачеві необхідні функції та можливості, що виконуються через веб-браузер як клієнта. Веб-додатки можуть працювати на будь-яких платформах або пристроях, що підтримують веб-браузер. Веб-додатки можуть бути розроблені з використанням різноманітних технологій, таких як HTML, CSS, JavaScript, PHP, Python тощо. Вони можуть мати різну функціональність, від простих статичних сторінок до складних додатків,

що включають в себе бази даних, аналітичні інструменти, можливості для взаємодії з користувачем тощо.

2. Існує кілька способів класифікації веб-додатків.
    - в залежності від місця виконання:
        - клієнт-серверні веб-додатки - виконуються на сервері та передають результати до клієнтського браузера;
        - клієнтські веб-додатки - виконуються на клієнтському пристрої, зазвичай, за допомогою JavaScript, а сервер використовується для отримання та збереження даних;
    - в залежності від рівня складності:
        - статичні веб-додатки - створюються за допомогою HTML та CSS та мають обмежені можливості;
        - динамічні веб-додатки - використовуються скрипти на стороні сервера та клієнта, що дозволяє створювати більш складні додатки зі змінним контентом;
        - інтерактивні веб-додатки - мають додаткові функції для взаємодії з користувачем, такі як анімація, графічний інтерфейс та інші.
3. Архітектура веб-додатку визначається залежно від його функціональності та завдань, які він вирішує. Зазвичай, веб-додатки складаються з трьох компонентів: клієнтської сторони, серверної сторони та бази даних. Клієнтська сторона включає в себе веб-браузер, який відображає інтерфейс користувача та взаємодіє з сервером. На цій стороні можуть використовуватись такі технології як HTML, CSS та JavaScript для відображення контенту та взаємодії з користувачем. Серверна сторона забезпечує обробку запитів та передачу даних між клієнтом та базою даних. На цій стороні можуть використовуватись такі технології як PHP, Python, Ruby, Java та інші для розробки логіки додатку та забезпечення зв'язку між клієнтом та базою даних. Серверна сторона веб-додатка відповідає за обробку запитів, збереження та оновлення даних, взаємодію з базою даних, аутентифікацію та авторизацію користувачів та інші бізнес-логічні операції.

4. Є багато веб-додатків, які допомагають тваринам. Ось декілька прикладів:
    - Petfinder - це додаток для пошуку та прийняття домашніх тварин. Він має базу даних з тисячами тварин, які шукають нового власника.
    - ASPCA - це додаток від Американського товариства захисту тварин, який надає інформацію про догляд за тваринами, поради щодо їх виховання та здоров’я.
    - iNaturalist - це додаток для збору інформації про дикі тварини. Він дозволяє користувачам знімати фотографії тварин, визначати їх вид та ділитися своїми спостереженнями з іншими користувачами.
    - HappyCow - це додаток для вегетаріанців та веганів, який допомагає знайти ресторани та кафе з вегетаріанською та веганською кухнею. Цей додаток допомагає зменшити попит на м’ясні продукти та сприяє зменшенню страждання тварин.
    - BarkHappy - це додаток для власників собак, який дозволяє знаходити друзів для своєї собаки та зустрічатися з іншими власниками собак для прогулянок та інших активностей.

РОЗДІЛ 2
АНАЛІЗ ТА ПОРІВНЯННЯ ОБРАНИХ ТЕХНОЛОГІЙ ДЛЯ WEB-ДОДАТКУ

2.1. Розробка макету

CMS (система управління контентом) грає важливу роль у створенні та управлінні веб-сайтами. CMS допомагають організовувати, зберігати і керувати контентом, таким як тексти, зображення, відео та інші матеріали на сайті. Можна перерахували декілька популярних безкоштовних CMS, таких як Joomla, Drupal, WordPress і Webnode.

Кожна з цих CMS має свої особливості, функціональні можливості та спрямована на різні потреби користувачів. WordPress, зокрема, є однією з найпопулярніших інструментів для створення та управління веб-сайтами, включаючи блоги, корпоративні сайти, електронні магазини тощо. Joomla та Drupal також є потужними і гнучкими CMS, які використовуються для створення складніших і більш високорівневих веб-проектів. Webnode, з іншого боку, володіє простим інтерфейсом та зручними інструментами для швидкого створення сайтів.

Вибір певної CMS залежить від потреб і вимог вашого проекту. Нам потрібно зрозуміти, які функції та можливості необхідні для нашого клубу і яка з CMS відповідає вашим потребам найкраще.

Обрання платформи WordPress для створення веб-додатку «Допомога тваринам» є розумним вибором, з урахуванням його цілей та обмежень. WordPress є відомою платформою для створення веб-сайтів, яка пропонує простий інтерфейс та інтуїтивно зрозумілі інструменти для створення і обслуговування сайту.

Оскільки веб-додаток «Допомога тваринам» працює на волонтерських засадах та обмежений у фінансуванні, важливо обрати платформу, яка не вимагатиме значних витрат або високих кваліфікаційних здібностей для її використання. WordPress відповідає цим вимогам, оскільки дозволяє створювати і керувати сайтом без необхідності програмування або великих знань у галузі веб-розробки.

Крім того, враховуючи, що веб-додаток має на меті надання інформаційної підтримки та привертання нових волонтерів, важливо, щоб платформа була зрозумілою та доступною для різних користувачів. WordPress надає широкий вибір шаблонів та інструментів для розміщення контенту, що полегшує створення інформаційних розділів та приваблює аудиторію.

Усі ці фактори роблять WordPress підходящим варіантом для створення веб-додатку «Допомога тваринам», який допоможе популяризувати ідею допомоги тваринам, залучати нових волонтерів та знайти нові домівки для тварин, які потребують уваги та догляду.

WordPress є однією з найпопулярніших веб-платформ для керування вмістом (Content Management System - CMS). Вона надає зручні інструменти для створення, редагування та керування веб-сайтами. Ось кілька ключових особливостей CMS WordPress:

1) простота використання: WordPress має інтуїтивний і легкий у використанні інтерфейс, що дозволяє навіть не-технічним користувачам створювати і керувати своїми веб-сайтами;
2) гнучкість: WordPress пропонує широкий вибір тем і плагінів, що дозволяють розширити функціональність вашого сайту. Ви можете налаштовувати вигляд та функції сайту за допомогою цих розширень;
3) SEO-придатність: WordPress має вбудовані функції для оптимізації пошукової системи (SEO), що дозволяють покращити видимість вашого сайту у пошукових системах;
4) багатомовність: WordPress підтримує створення багатомовних сайтів, що дозволяє вам легко створювати і управляти сайтами з різною мовною версією;

5) активна спільнота: WordPress має велику та активну спільноту користувачів, яка надає безліч документації, підручників, форумів та підтримки.

WordPress також має багато інших переваг, таких як безпека, масштабованість, підтримка мультимедіа контенту і багато іншого. Вона є популярним вибором для веб-розробників і веб-майстрів, оскільки надає потужні можливості для створення різноманітних веб-сайтів, включаючи особисті блоги, корпоративні сайти, електронні магазини та інші типи сайтів.

Розробка веб-додатку «Допомога тваринам» з функціями, які описані, допоможе створити центральну інформаційну платформу для волонтерів та сприятиме спілкуванню між волонтерами та відвідувачами сайту. Основні функції сайту можуть бути реалізовані за допомогою різних модулів і розділів, таких як:

1) головна сторінка: можна створити привітальну сторінку, з коротким описом організації та її місії, а також відображення останніх новин та подій:
2) розділ "Про нас": тут можна представити докладну інформацію про волонтерський клуб, його цілі, історію, команду волонтерів та їхні досягнення;
3) розділ "Життя тварин": в цьому розділі можна розмістити різноманітну інформацію про життя тварин, їх потреби, догляд, поради щодо утримання та виховання. Можна також додати фото- і відеоматеріали, аудіофайли з звуками тварин тощо;
4) розділ "Новини та заходи": цей розділ міститиме оновлення про актуальні події, заходи, збори, акції, які проводить клуб. Тут можна розмістити анонси, дати і часи заходів, фото- і відеозвіти;
5) контактна інформація: важливо надати контактні дані клубу, такі як адреса, телефон, електронна пошта, а також форму зворотного зв'язку.

Волонтерський веб-додаток "Допомога тваринам" з використанням CMS WordPress є чудовим варіантом для створення інформаційної платформи та забезпечення ефективної співпраці волонтерів та користувачів.

CMS WordPress надає широкі можливості для розробки та налаштування веб-додатків без значних технічних знань. Основні переваги використання WordPress для вашого волонтерського веб-додатку включають:

1) простота використання: WordPress має інтуїтивний і дружній інтерфейс, що дозволяє вам легко створювати, редагувати та оновлювати контент на вашому веб-сайті;
2) гнучкість та розширюваність: WordPress пропонує безліч тем, плагінів та розширень, які дозволяють налаштовувати функціональність вашого веб-додатку під ваші потреби. Ви зможете додати форми зворотного зв'язку, календарі подій, систему коментарів, соціальні медіа-поділки та багато іншого;
3) мобільна готовність: WordPress надає можливість створювати мобільно-адаптивні веб-сайти, що забезпечує оптимальний перегляд і взаємодію з вашим веб-додатком на різних пристроях;
4) SEO-придатність: WordPress має вбудовані функції, які сприяють оптимізації пошукових систем (SEO). Ви зможете легко налаштувати метатеги, URL-адреси та інші елементи для поліпшення видимості вашого веб-додатку у пошукових системах;
5) спільнота та підтримка: WordPress має велику та активну спільноту користувачів, де ви зможете знайти відповіді на свої питання

Для створення структури роботи web-додатку з використанням наведених даних можна розглянути наступну схему:

1. Вхідні дані: запит користувача:
    - дані, що надходять від користувача через інтерфейс web-додатку. Це може бути форма заповнення, кнопки, посилання або інші елементи інтерфейсу, які дозволяють користувачеві взаємодіяти з додатком.
2. Вихідні дані: робочий web-додаток, відображення необхідної інформації:
    - результати обробки вхідних даних, які відображаються у вигляді веб-сторінок, даних або інших елементів, доступних користувачеві. Це

може включати відображення тексту, зображень, таблиць, графіків, форм або інших компонентів.

3. Контроль: права та інструкція адміністратора, права доступу користування:
    - механізми, які забезпечують контроль доступу до робочого web-додатку. Це можуть бути системи аутентифікації та авторизації, ролі та дозволи користувачів, обмеження доступу до певних функцій або даних. Адміністратор може мати особливі права та може налаштовувати систему.
4. Механізми реалізації: програмне забезпечення та технічне забезпечення:
    - компоненти, що використовуються для реалізації web-додатку. Це можуть бути програмні бібліотеки, фреймворки, мови програмування, бази даних, сервери, хостинг-провайдери та інші технології, необхідні для створення та функціонування додатку.

За допомогою програмного забезпечення та технічного забезпечення реалізується обробка вхідних даних.

Детальніша функціональна декомпозиція системи може включати наступні рівні:

1. Рівень авторизації:
    - користувач вводить свої облікові дані (логін та пароль);
    - система перевіряє введені дані на відповідність збереженим обліковим записам;
    - якщо дані вірні, система надає користувачу доступ до функціональності додатку залежно від його ролі (адміністратор, звичайний користувач тощо);
    - якщо дані невірні, користувачу відображається повідомлення про невдалий вхід.
2. Рівень пошуку інформації:
    - користувач вводить критерії пошуку (наприклад, ключові слова, дати, параметри тощо);

- система обробляє введені дані і здійснює пошук інформації в базі даних або зовнішніх джерелах;
- знайдена інформація відображається користувачу у вигляді результатів пошуку;
- користувач може застосувати додаткові фільтри або сортування для точнішого визначення потрібної інформації.

3. Рівень перегляду контенту:
    - користувач вибирає певний елемент або категорію контенту для перегляду;
    - система відображає відповідний контент користувачеві (текст, зображення, відео, тощо) згідно з вибраними налаштуваннями та доступом користувача;
    - користувач може навігувати, скролити, редагувати або видаляти контент залежно від його прав доступу та ролі;
    - користувач також може застосувати фільтри або сортування для зручного перегляду контенту.

На кожному рівні можуть бути додаткові функції та опції, що залежать від специфіки конкретного web-додатку та вимог користувачів. Крім того, ці рівні можуть взаємодіяти між собою для забезпечення повноцінної роботи додатку (наприклад, після авторизації користувач може здійснювати пошук і перегляд контенту).

У процесі "авторизації" включаються такі дані та елементи:

1. Вхідні дані:
    - логін: це ідентифікатор, який введений користувачем для ідентифікації себе в системі;
    - пароль: це конфіденційна інформація, введена користувачем для перевірки його автентичності.
2. Вихідні дані:
    - права доступу: після успішної авторизації система надає користувачу певні права доступу залежно від його ролі або привілеїв. Ці права

визначають, які дії та функціональні можливості доступні користувачеві в межах web-додатку.

3. Контроль:
    - права та інструкція адміністратора: адміністратор системи встановлює та керує правами доступу користувачів. Він визначає, які ресурси, функції та інформація доступні для конкретних ролей або користувачів. Крім того, адміністратор може встановлювати політики безпеки, парольні вимоги та інші правила, пов'язані з авторизацією.
4. Інструменти:
    - технічне та програмне забезпечення: це компоненти та ресурси, необхідні для здійснення процесу авторизації. Вони можуть включати системи аутентифікації та авторизації, бази даних, шифрування, логування, механізми сесій та інші інструменти для забезпечення безпеки та контролю доступу.

У процесі "авторизації" користувачі надають свої облікові дані (логін і пароль), які порівнюються зі збереженими в системі. Якщо дані вірні, користувачу присвоюються певні права доступу, які він може використовувати для здійснення пошуку та перегляду інформації у web-додатку. Адміністратор системи контролює ці права та може встановлювати ролі та політики, що впливають на доступ користувачів.

При декомпозиції процесу "пошук інформації" враховуються наступні дані:

1. Вхідні дані:
    - права доступу: це інформація про рівень доступу користувача до функціональності пошуку інформації. В залежності від прав доступу користувача можуть бути обмеження щодо видимості або доступу до певних даних або функцій пошуку.
2. Вихідні дані:
    - запит: це параметри, введені користувачем для пошуку інформації. Запит може включати ключові слова, фільтри, критерії, сортування та інші параметри, які визначають потрібну інформацію.
3. Контроль:

- права доступу користування: це правила та обмеження, які визначають, яку інформацію користувач може шукати. На основі рівня доступу користувача, система може обмежувати видимість та доступність певних категорій або типів інформації під час пошуку.

4. Інструменти:
    - технічне та програмне забезпечення: це компоненти та ресурси, які використовуються для здійснення процесу пошуку інформації. Це можуть бути пошукові двіжки, бази даних, алгоритми пошуку, індексація, фільтрація та інші інструменти, які допомагають виконувати ефективний пошук інформації в системі.

У процесі "пошук інформації" користувачі з відповідними правами доступу вводять запит для пошуку інформації. Система використовує інструменти та технічне забезпечення для обробки запиту та пошуку в базі даних або інших джерелах інформації. Контроль прав доступу користування забезпечує, що користувачі отримують лише доступну для них інформацію.

У процесі "перегляд контенту" враховуються наступні дані:

1. Вхідні дані:
    - запит: це запит, який користувач вводить для отримання певної інформації або контенту для перегляду. Запит може містити параметри, фільтри, ідентифікатори або інші деталі, що визначають необхідну інформацію.
2. Вихідні дані:
    - інформація: це результат, який користувач отримує для перегляду. Інформація може бути текстовим, візуальним, мультимедійним або будь-яким іншим типом контенту, що задовольняє запит користувача.
3. Контроль:
    - права доступу користування: це правила та обмеження, які визначають, яку інформацію користувач може переглядати. За допомогою прав доступу користувача обмежується доступ до конкретного контенту, категорій або функцій, залежно від його ролі або привілеїв.

4. Інструменти:
    - технічне та програмне забезпечення: це компоненти та ресурси, які використовуються для відображення та доставки контенту користувачеві. Це можуть бути шаблони веб-сторінок, віджети, мультимедійні плеєри, браузери, мережеві протоколи та інші інструменти, які забезпечують відображення контенту.

У процесі "перегляд контенту" користувачі з відповідними правами доступу отримують результати запиту від системи. Система використовує технічне та програмне забезпечення для відображення інформації або контенту користувачам з урахуванням їхніх прав доступу користування.

На основі визначених акторів і операцій використання можна побудувати діаграму варіантів використання (Use Case Diagram). Дана діаграма (рис. 2.1.) візуалізує взаємодію між акторами та системою (в даному випадку, web-додатком) та показує, які операції можуть бути виконані кожним актором. Нижче наведена пропозиція діаграми варіантів використання для вказаних операцій:

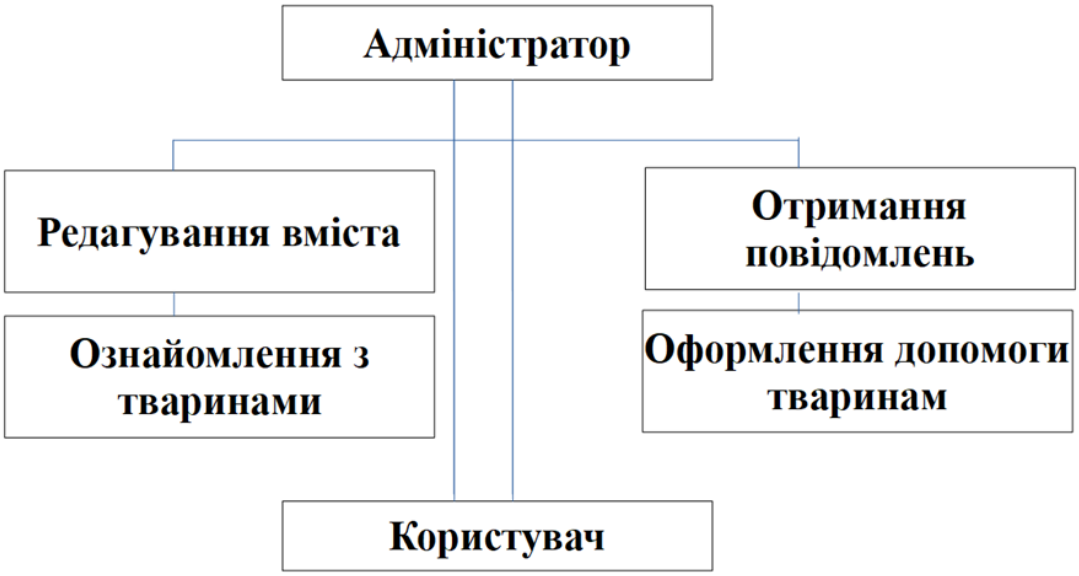

Рис. 2.1. Діаграма варіантів використання (Use Case Diagram)

2.2. HTML

HTML, CSS та JavaScript є основними технологіями для розробки фронтенду веб-додатків. Кожна з цих технологій має свою власну зону відповідальності.

HTML (HyperText Markup Language) використовується для створення структури веб-сторінок та визначення контенту, який відображається на сторінці. HTML визначає різні елементи, такі як заголовки, абзаци, списки, таблиці, зображення та посилання, які використовуються для створення вмісту веб-додатку.

CSS (Cascading Style Sheets) відповідає за зовнішній вигляд веб-додатку. Використовуючи CSS, можна задавати стилі, такі як кольори, шрифти, розміри, положення елементів і т.д. CSS дозволяє створювати привабливі та естетичні веб-сторінки.

JavaScript є мовою програмування, яка використовується для створення динамічної логіки веб-додатків. JavaScript дозволяє взаємодіяти з користувачем, обробляти події, відправляти запити на сервер, змінювати вміст сторінки динамічно, виконувати обчислення та багато іншого. JavaScript є основним інструментом для клієнтського скриптування у браузерах.

Щодо підходів до побудови клієнтської частини веб-додатків, ось кілька основних підходів:

1. Server-side rendering (SSR): у цьому підході веб-сервер генерує HTML-код для кожного запиту та надсилає його до браузера. Це означає, що веб-сторінка повністю рендериться на сервері перед відправкою до клієнта. SSR корисний для покращення швидкодії, SEO-оптимізації та першого відображення сторінки. Основні технології для SSR включають Node.js, Next.js, Nuxt.js та Angular Universal.
2. Client-side rendering (CSR): у цьому підході веб-сторінка завантажується у браузер у вигляді пустого шаблону HTML, а потім клієнтський JavaScript бере на себе відповідальність за рендеринг та динамічне заповнення вмісту сторінки. CSR дозволяє створювати багатофункціональні та інтерактивні

додатки, але може мати певні проблеми з швидкодією та SEO. Основні фреймворки та бібліотеки для CSR включають React, Angular та Vue.js.

3. Static Site Generators (SSG): у цьому підході веб-сайт передбачувано генерується на стадії розробки та зберігається у вигляді статичних файлів. Коли користувач запитує сторінку, веб-сервер просто повертає заздалегідь згенерований файл. SSG використовується для створення швидких та безпечних веб-сайтів з простими потребами динаміки. Деякі популярні SSG включають Gatsby, Next.js (з режимом генерації на стороні сервера) та Jekyll.

Кожен з цих підходів має свої переваги та використовується залежно від конкретних потреб проекту.

HTML є основною мовою розмітки для створення структури веб-сторінок. Використовуючи різні HTML-теги, можна визначати різні елементи на сторінці, такі як заголовки (наприклад, **<h1>**, **<h2>**, тощо), абзаци (** **), списки (наприклад, **<ul>**, **<ol>**, **<li>**), таблиці (**<table>**, **<tr>**, **<td>**), зображення (**<img>**), посилання (**<a>**), форми (**<form>**, **<input>**, <button>) та багато іншого.

HTML визначає структуру веб-сторінки, встановлює відношення між різними елементами та визначає їхнє місце на сторінці. Завдяки HTML, веб-браузер може правильно інтерпретувати та відображати вміст сторінки для користувачів.

Наприклад, такий HTML-код створює заголовок першого рівня і абзац:

```
<h1>Це заголовок першого рівня</h1>
<p>Це абзац тексту.</p>
```

Рис. 2.2. Приклад коду HTML

<h1>Це заголовок першого рівня</h1>

Це абзац тексту.

Після обробки веб-браузером, цей код відображатимеся як:

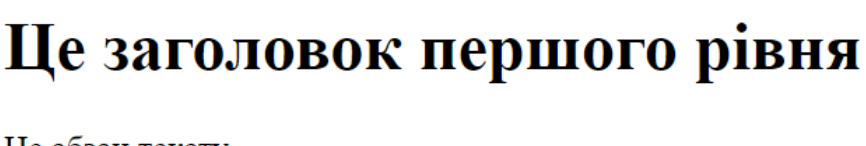

Рис. 2.3. Приклад відображення коду HTML

HTML є основою веб-розробки і використовується для визначення структури та контенту веб-сторінок.

HTML (HyperText Markup Language) визначає структуру та семантику вмісту веб-сторінки, включаючи текст, зображення, посилання, таблиці, форми та інші елементи.

За допомогою HTML-тегів і їх атрибутів, ви можете створювати різні елементи на сторінці та встановлювати їх відношення один до одного. Наприклад, <**h1**> використовується для створення заголовка першого рівня, <**p**> - для абзацу, <**img**> - для вставки зображення, <**a**> - для створення посилання і так далі.

HTML також дозволяє використовувати структурні елементи, такі як <**header**>, <**nav**>, <**main**>, <**footer**>, для логічного групування різних частин веб-сторінки. Це полегшує розуміння структури сторінки як розробниками, так і пошуковими системами.

Крім того, HTML використовується для впровадження різноманітного контенту, такого як звуки, відео, вкладки, форми вводу даних та багато іншого.

Загалом, HTML є основним інструментом для розробки веб-сторінок і визначення їхньої структури та контенту.

2.3. CSS

CSS (Cascading Style Sheets) відповідає за зовнішній вигляд веб-додатку. Використовуючи CSS, можна задавати стилізацію елементів веб-сторінки, таку як кольори, шрифти, розміри, положення елементів, фонові зображення та багато іншого.

CSS дозволяє розділити структуру веб-сторінки, визначену за допомогою HTML, від її зовнішнього вигляду. Це дозволяє змінювати стиль і вигляд веб-сторінки без необхідності змінювати її структуру.

CSS використовується для визначення різних стилів за допомогою правил, що складаються з селекторів та оголошень властивостей. Селектори вказують, які елементи на веб-сторінці будуть застосовуватися до стилів, а оголошення властивостей визначають, які конкретні стилі будуть застосовуватися до вибраних елементів.

Наприклад, такий CSS-код визначає стиль заголовка першого рівня та абзацу:

```css
h1 {
  color: blue;
  font-size: 24px;
}

p {
  color: black;
  font-size: 16px;
}
```

Рис. 2.4. Приклад використання CSS-коду

Це означає, що всі заголовки першого рівня будуть мати синій колір тексту та розмір шрифту 24 пікселі, а всі абзаци матимуть чорний колір тексту та розмір шрифту 16 пікселів.

CSS дозволяє створювати привабливі та естетичні веб-сторінки, надаючи можливість контролювати вигляд та стиль елементів. Використання CSS разом з HTML дозволяє створювати професійний та зручний інтерфейс для користувачів.

2.4. JavaScript

JavaScript є мовою програмування, яка використовується для створення динамічної логіки веб-додатків. Він надає засоби для взаємодії з користувачем,

обробки подій, відправки запитів на сервер, зміни вмісту сторінки динамічно, виконання обчислень та багато іншого.

JavaScript використовується для реалізації клієнтської частини веб-додатків, що виконується безпосередньо в браузері. Він дозволяє додавати інтерактивність та динаміку до веб-сторінок, реагувати на події користувача, валідувати дані, виконувати асинхронні запити до сервера та оновлювати вміст сторінки без перезавантаження.

JavaScript також надає доступ до Document Object Model (DOM) - представлення веб-сторінки в браузері у вигляді дерева об'єктів. За допомогою JavaScript можна змінювати структуру, стиль і вміст сторінки, додавати та видаляти елементи, змінювати атрибути та стилі елементів, маніпулювати подіями, які виникають на сторінці та багато іншого.

Наприклад, за допомогою JavaScript можна створити функцію, яка реагує на клік по кнопці та змінює текстовий вміст елемента:

```
// Отримуємо посилання на елемент кнопки
var button = document.getElementById('myButton');

// Додаємо обробник події для кліку на кнопку
button.addEventListener('click', function() {
  // Змінюємо текстовий вміст елемента
  button.textContent = 'Натиснуто!';
});
```

Рис. 2.5. Приклад JavaScript

JavaScript підтримує роботу з масивами, об'єктами, умовними операторами, циклами та іншими конструкціями програмування, що дозволяє реалізувати складні логічні операції та обробку даних.

Масиви в JavaScript дозволяють зберігати та маніпулювати колекцією значень. Ви можете додавати, видаляти, змінювати елементи масиву та виконувати різні операції, такі як сортування, фільтрація, знаходження максимального або мінімального значення та інші.

Об'єкти в JavaScript представляють собою структури даних, що містять набір ключ-значення. Вони дозволяють організувати дані в більш складних структурах, здатних відображати реальні об'єкти, явища або концепції. Ви можете створювати, змінювати та отримувати значення за ключем у об'єктах.

Умовні оператори, такі як **if**/**else**, дозволяють виконувати певні дії на основі заданих умов. Так, умовні оператори в JavaScript, такі як **if**, **else if** та **else**, дозволяють виконувати певні дії на основі заданих умов. Вони дозволяють програмі приймати рішення на основі значень або стану змінних.

Оператор **if** виконує блок коду, якщо вказана умова є істинною. Наприклад:

```
var number = 10;

if (number > 0) {
  console.log("Число додатне");
}
```

Рис. 2.6. Приклад значення змінної number більше нуля JavaScript

У цьому прикладі, якщо значення змінної number більше нуля, то виводиться повідомлення "Число додатне".

Оператор else if дозволяє встановити додаткові умови для перевірки, якщо попередня умова не виконується. Наприклад:

```
var number = 0;

if (number > 0) {
  console.log("Число додатне");
} else if (number < 0) {
  console.log("Число від'ємне");
} else {
  console.log("Число нуль");
}
```

Рис. 2.7. Приклад значення змінної **number** менше нуля JavaScript

У цьому прикладі, якщо значення змінної **number** менше нуля, виводиться повідомлення "Число від'ємне". Якщо ж жодна з умов не виконується, виводиться повідомлення "Число нуль".

Оператор **else** виконується, якщо жодна з попередніх умов не виконується. Він може бути використаний безпосередньо після оператору **if** або **else if**. В останньому прикладі, блок коду в операторі **else** виконується, якщо значення змінної number не є ні додатним, ні від'ємним.

Ці умовні оператори дозволяють структурувати логіку програми та виконувати різні дії в залежності від умов, що забезпечує більш гнучке управління програмою.

2.5. Висновок до розділу 2

1. Волонтерський веб-додаток "Допомога тваринам" з використанням CMS WordPress є чудовим варіантом для створення інформаційної платформи та забезпечення ефективної співпраці волонтерів та користувачів. CMS WordPress надає широкі можливості для розробки та налаштування веб-додатків.
2. HTML є основою веб-розробки і використовується для визначення структури та контенту веб-сторінок. HTML (HyperText Markup Language) визначає структуру та семантику вмісту веб-сторінки, включаючи текст, зображення, посилання, таблиці, форми та інші елементи.
3. CSS (Cascading Style Sheets) відповідає за зовнішній вигляд веб-додатку. Використовуючи CSS, можна задавати стилізацію елементів веб-сторінки, таку як кольори, шрифти, розміри, положення елементів, фонові зображення та багато іншого. CSS дозволяє розділити структуру веб-сторінки, визначену за допомогою HTML, від її зовнішнього вигляду. Це дозволяє змінювати стиль і вигляд веб-сторінки без необхідності змінювати її структуру. CSS використовується для визначення різних стилів за допомогою правил, що складаються з селекторів та оголошень властивостей. Селектори вказують, які елементи на веб-сторінці буде

застосовано до стилів, а оголошення властивостей визначають, які конкретні стилі будуть застосовувані до вибраних елементів.

4. JavaScript підтримує роботу з масивами, об'єктами, умовними операторами, циклами та іншими конструкціями програмування, що дозволяє реалізувати складні логічні операції та обробку даних. Масиви в JavaScript дозволяють зберігати та маніпулювати колекцією значень та виконувати різні операції, такі як сортування, фільтрація, знаходження максимального або мінімального значення та інші. Об'єкти в JavaScript представляють собою структури даних, що містять набір ключ-значення. Вони дозволяють організувати дані в більш складних структурах, здатних відображати реальні об'єкти, явища або концепції. Можна створювати, змінювати та отримувати значення за ключем у об'єктах.

РОЗДІЛ 3
ОПИС ПРОТОТИПУ WEB-ДОДАТКУ «ДОПОМОГА ТВАРИНАМ» З ЗАСТОСУВАННЯМ CMS WORDPRESS

3.1. Глобальний дизайн інтерфейсу веб-додатку та навігація

Для розробки веб-додатку використано інформаційний матеріал, фотографії та відео, які відповідають тематиці волонтерського проекту допомоги тваринам. З дизайну головної сторінки вирішено зберігати спільну стильову концепцію для всіх сторінок, зокрема:

- розмір та колір шрифту: враховуючи дизайн головної сторінки, встановлено певні розміри шрифту для текстових елементів на всіх сторінках додатку. Колір шрифту також міг бути визначений, щоб відповідати загальній кольоровій схемі та забезпечувати достатню читабельність;
- гарнітура шрифту: для створення єдиної стильової концепції обрано певну гарнітуру шрифту, яка використовувалась на всіх сторінках додатку. Вибір гарнітури шрифту може бути здійснений враховуючи естетичні та читабельні вимоги проекту;
- взаємне розташування елементів: збережено послідовність та логіку розташування елементів на всіх сторінках додатку. Це може включати розташування заголовків, текстових блоків, зображень та кнопок згідно з вимогами дизайну;

- кольорова гама: вибір кольорової гами також заснований на дизайні головної сторінки. Визначити певні кольори або палітру кольорів, які використовуються на всіх сторінках додатку для забезпечення єдиності та гармонії.На першому етапі розробки web-додатку розроблено головну сторінку, в якій дизайн оперувався від обраної тематики волонтерського проекту допомоги тваринам.

Цей етап включав наступні етапи:

1. Визначення тематики: я вибрала тему волонтерського проекту допомоги тваринам. Це слугувало основою для подальшого дизайну та контенту сторінки.
2. Розробка концепції дизайну: На основі обраної тематики, розробила концепцію дизайну головної сторінки. Це включало в себе вибір кольорової палітри, шрифтів, розміщення елементів та загальний стиль, які відповідали тематиці волонтерського проекту.
3. Створення макету: За допомогою дизайнерських інструментів, таких як Adobe Photoshop, створено макет головної сторінки. Макет включав в себе розміщення елементів, кольорову схему, шрифти та інші дизайнерські елементи.
4. Реалізація дизайну з використанням HTML та CSS: На основі створеного макету, розробники перенесли дизайн головної сторінки в реальний код за допомогою HTML та CSS. Вони використовували CSS для задання стилів, таких як кольори, шрифти, розміри, положення елементів і т.д.
5. Додавання контенту: Після реалізації дизайну я додала вміст на головну сторінку. Це включає інформацію про діяльність організації для допомоги тваринам, відгуки клієнтів, корисні факти про волонтерський проект допомоги тваринам.

Цей перший етап спрямований на створення привабливої та інформативної головної сторінки для web-додатку, яка відповідає тематиці волонтерського проекту допомоги тваринам. Це включає :

- Використання відповідної кольорової палітри: вибір кольорів, які відповідають тематиці волонтерського проекту. Наприклад, використання відтінків зеленого або синього, які асоціюються з природою та тваринами.
- Привернення уваги до ключової інформації: Головна сторінка має містити інформацію про діяльність замовника, відгуки клієнтів та корисні факти про проект. Розміщувала цю інформацію в привабливому та видатному способі, щоб вона була легко помітна та привертала увагу користувачів.
- Використання зрозумілого та зручного розташування елементів: Елементи на головній сторінці розташовані з урахуванням логічного та зручного порядку. Наприклад, основна інформація може бути розміщена в центральній частині сторінки, відгуки клієнтів - з одного боку, а корисні факти - з іншого. Це допоможе користувачам швидко зорієнтуватися та знайти потрібну інформацію.
- Використання зображень та відео: додала фотографії та відео, пов'язані з тематикою проекту. Наприклад, зображення котів, волонтерів або сцени допомоги тваринам.

3.2. Розробка інтерфейсу сторінок

Прийняли рішення використовувати Adobe Photoshop, оскільки це програмне забезпечення має широкий функціонал для редагування растрових зображень.

Adobe Photoshop є одним з провідних графічних редакторів, який дозволяє розробникам та дизайнерам створювати детальні макети web-сторінок. Він надає багатий набір інструментів для маніпулювання та редагування растрових зображень, що може бути корисним при створенні дизайну веб-додатку. За допомогою Photoshop можна створювати інтерфейсні елементи, налаштовувати кольори, розміщення елементів та інші дизайнерські аспекти, необхідні для розробки макету web-додатка.

Однак, варто зауважити, що інші редактори, такі як Adobe XD, Sketch та Figma, також є популярними серед розробників та дизайнерів web-додатків. Вони надають інтуїтивний інтерфейс, спеціалізовані функції для дизайну інтерфейсу користувача та

спільну роботу в команді. Вибір редактора залежить від вподобань та потреб розробників проекту.

Зручність використання (Usability) є важливим аспектом розробки web-додатків. Usability ставить у центр уваги зручність та задоволення користувачів при використанні додатку. Основні принципи зручності використання включають:

- простота та зрозумілість: Додаток має бути легким у використанні та зрозумілим для користувачів. Інтерфейс повинен мати чітку та логічну структуру, а навігація - інтуїтивну.
- Ефективність: Додаток повинен дозволяти користувачам виконувати свої завдання швидко та ефективно. Інформація та функціонал мають бути легкодоступними та швидкодіючими.
- Консистентність: Інтерфейс та взаємодія повинні бути консистентними на всіх сторінках та елементах, що взаємодіють один з одним. Це допомагає користувачам легко орієнтуватися та передбачувати реакцію додатку на їхні дії.
- Запобігання помилкам: Додаток повинен надавати захист від помилок та допускати мінімальну кількість помилкових дій. Валідація даних та чітке повідомлення про помилки можуть допомогти уникнути неприємних ситуацій для користувачів.
- Доступність: Додаток має бути доступним для різних користувачів, включаючи людей з обмеженими можливостями. Розміщення контенту, чітка читабельність шрифтів та можливість використання клавіатури для навігації - це лише кілька аспектів доступності, які варто враховувати.

Аналіз кольорів для вибору кольорової гами є важливою частиною процесу дизайну web-додатків. Колір має великий вплив на сприйняття та настрій користувачів. Основні кольори, які я врахувала, включають:

- червоний: червоний колір викликає емоції, такі як енергія, сила, він може використовуватися для акцентування уваги, виділення важливої інформації або створення враження динамізму;

- блакитний: блакитний колір асоціюється зі спокоєм, стабільністю та довірою, він може використовуватися для створення відчуття професійності та надійності;
- зелений: зелений колір пов'язується з природою, свіжістю та рістом, він може використовуватися для позитивного враження, а також для тем, пов'язаних з екологією та сталою розвитком;
- фіолетовий: фіолетовий колір асоціюється з розкішшю, таємничістю та творчістю він може використовуватися для створення враження елегантності та привабливості;
- помаранчевий: помаранчевий колір викликає енергію, радість та оптимізм, він може використовуватися для привертання уваги та створення враження веселощів.

Під час вибору кольорової гами важливо враховувати контекст і цільову аудиторію додатку, а також забезпечити гармонійне поєднання кольорів та читабельність вмісту.

3.3. Створення макету

До основних етапів розробки додатку можна віднести:

1) розробка інтерфейсу користувача;
2) розробка макету;
3) розробка допоміжного функціоналу.

Розробка макету веб-сайту включає кілька етапів, які можуть варіюватися залежно від процесу роботи команди розробників і їхніх вимог. Основні етапи розробки макету веб-сайту включають:

1. Збір вимог: Цей етап включає вихідних даних, встановлення функціональних та нефункціональних вимог до веб-сайту. Важливо визначити тип контенту, структуру, функціональність, дизайн та інші особливості, які будуть враховані при розробці макету.

2. Створення концепту: На цьому етапі розробляється загальна концепція веб-сайту, яка включає його структуру, навігацію, основні функціональні блоки та макет головних сторінок. Важливо зосередитись на зручності використання, добре візуальне представлення та відповідність бренду клієнта.
3. Розробка дизайну: На цьому етапі створюється візуальний дизайн веб-сайту. Він включає в себе вибір кольорової палітри, шрифтів, стилів та компонування елементів на сторінцях. Важливо забезпечити відповідність бренду клієнта, привабливий вигляд та зручність використання.
4. Реалізація макету: На цьому етапі використовували HTML, CSS та інші технології для створення розмітки, стилізації та інтеграції макету зі звичайними елементами веб-сайту, такими як заголовки, навігаційні панелі, форми, зображення.
5. Тестування та затвердження: Після реалізації макету провели тестування для перевірки функціональності, сумісності з різними пристроями та браузерами, а також відповідності вимогам.

Розробка web-додатку з використанням системи управління вмістом (CMS) "WordPress" може бути ефективним вибором для проекту допомоги тваринам. WordPress є популярною та потужною платформою, яка надає зручний інтерфейс для керування контентом та функціоналом веб-додатку.

Переваги використання WordPress включають:

- Простота використання: WordPress має інтуїтивний інтерфейс, що дозволяє користувачам з різним рівнем володіння технологіями легко додавати, редагувати та видаляти вміст на своєму веб-сайті.
- Розширюваність: WordPress має велику кількість безкоштовних та платних розширень (плагінів), які дозволяють розширити функціонал додатку за потребами проекту. Наприклад, для додавання функціоналу, пов'язаного зі збором пожертвувань, адоптацією тварин або створенням блогу про допомогу тваринам.

- Гнучкість дизайну: WordPress надає можливості для кастомізації зовнішнього вигляду додатку за допомогою тем. Можна вибрати або розробити тему, яка відповідає дизайну та потребам проекту допомоги тваринам.
- SEO-придатність: WordPress має вбудовану підтримку для пошукової оптимізації (SEO), що допомагає покращити видимість додатку в пошукових системах. Це може бути важливим аспектом для залучення більшої аудиторії та поширення інформації про допомогу тваринам.

Проте, важливо враховувати, що розробка додатку на базі WordPress вимагає розуміння платформи, а також налагодження безпеки. Оскільки WordPress є популярною платформою, він може бути піддатливим до потенційних кібератак. Потрібно забезпечити належні заходи безпеки, такі як регулярні оновлення, використання надійних розширень та захист від несанкціонованого доступу.

Крім того, розробка додатку на базі WordPress може мати обмеження в зв'язку зі структурою та функціоналом платформи. Якщо є намір реалізувати складний та унікальний функціонал, може знадобитися налаштування та розробка додаткових компонентів, що може бути часо- та ресурсозатратним.

Також важливо зазначити, що вибір системи управління вмістом (CMS) залежить від конкретних потреб та вимог проекту. Існують інші платформи CMS, які також можуть бути варіантом для розробки додатку допомоги тваринам. Перед прийняттям рішення слід ретельно оцінити потреби та можливості проекту.

Для того, що виконувати коректну розробку, процес тестування проекту, подальші налагодження, перш за все необхідно виконати реєстрацію на сайті https://www.wordpress.com.ua/ (рис. 3.1).

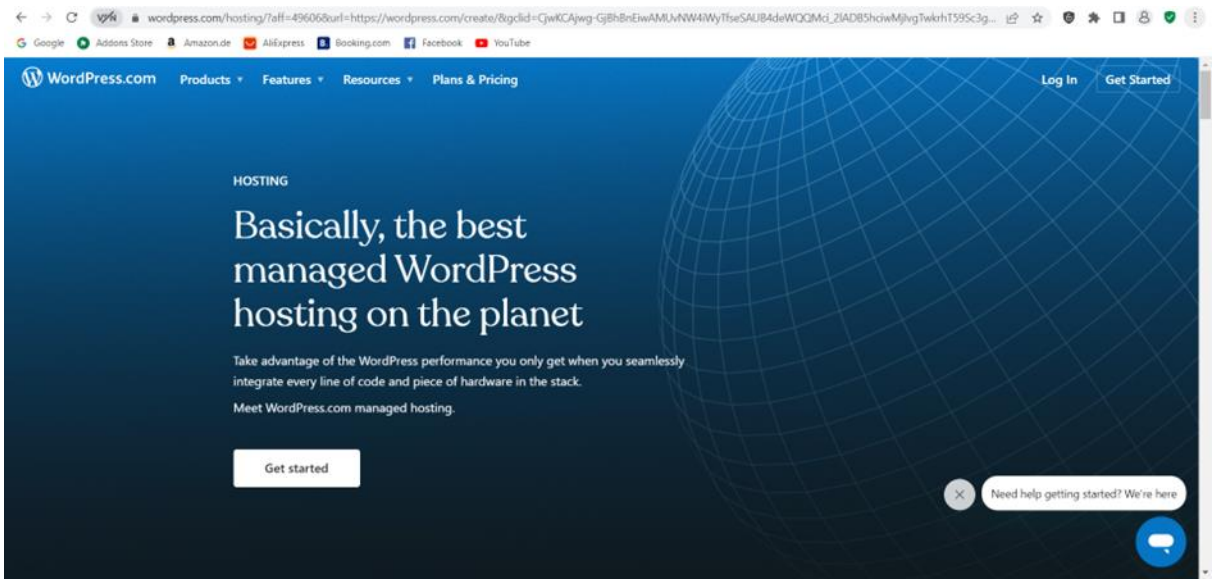

Рис. 3.1. Головна сторінка сайту WordPress

Після того, як реєстрація була виконана успішно, перейшла до процесу розробки сайту. На даному етапі необхідно провести аналіз існуючих шаблонів та обрати раціональний. На рис. 3.2 представлене стартове вікно системи, для проходження авторизації.

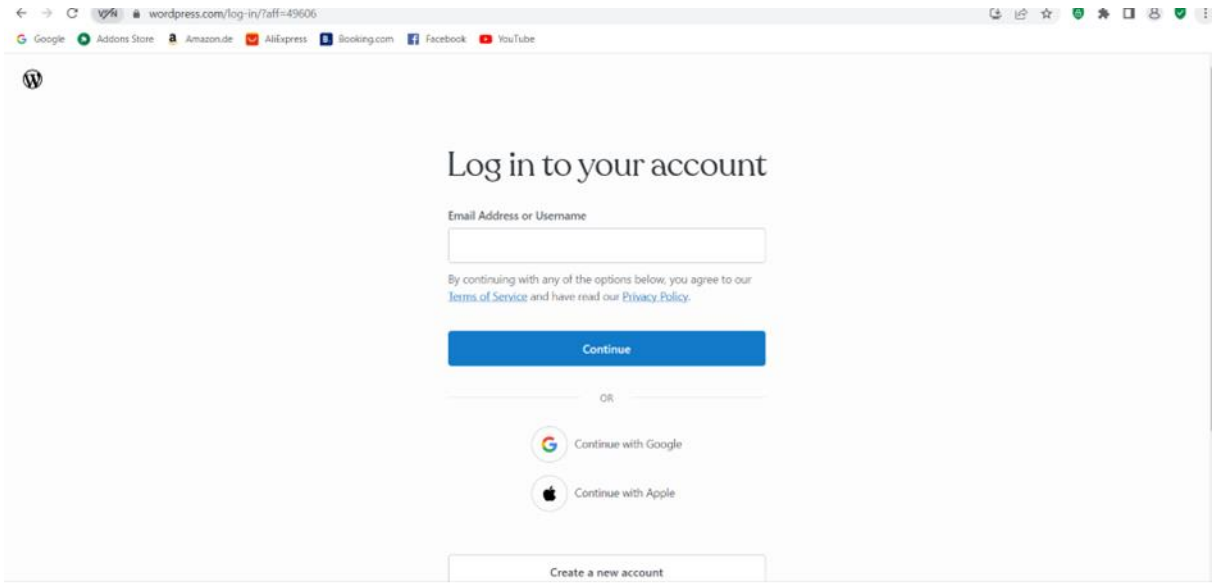

Рис. 3.2. Реєстрація на сайті WordPress

При виборі шаблону для проекту на базі WordPress, потрібно звернути увагу на такі фактори, щоб він відповідав вимогам проекту:

- Легкість редагування шаблону: Обраному шаблону слід мати зрозумілий та простий для редагування код або інтерфейс управління, щоб була можливість вносити зміни до вигляду та функціоналу шаблону зручним способом.
- Наявність необхідних віджетів: Шаблон повинен містити віджети або елементи, які дозволяють реалізувати потрібний функціонал проекту. Наприклад, якщо проект потребує блоку з останніми новинами або слайдера зображень, відповідні віджети повинні бути доступні в шаблоні.
- Редагування віджетів згідно макетів: Шаблон повинен надавати можливість редагування віджетів, щоб я могла відповідно до розроблених макетів налаштовувати зовнішній вигляд різних елементів, таких як заголовки, текстові блоки, кнопки тощо.
- Зміна кольорової гами шаблону: Шаблон повинен надавати можливість зміни кольорової гами, щоб відповідно до бренду або дизайну вибрати потрібні кольори для елементів шаблону.
- Зміна гарнітури шрифту: Шаблон повинен дозволяти змінювати гарнітуру шрифту, яка за замовченням встановлена у шаблоні. Це дозволить команді проекту використовувати власні шрифти або відповідати дизайну проекту.

Вибір конкретного шаблону дійсно залежить від унікальних потреб і вимог проекту. Рекомендується ознайомитися з різними шаблонами, провести дослідження та порівняння їх функціоналу, можливостей налаштування, дизайну, а також переглянути відгуки та рейтинги користувачів.

Під час вибору шаблону, треба враховувати також такі фактори:

- Сумісність з версією WordPress: Треба переконатися, що шаблон сумісний з поточною версією WordPress і продовжуватиме отримувати оновлення та підтримку в майбутньому.
- Підтримка та документація: Треба перевірити, чи є надійна підтримка та наявна документація для шаблону. Це допоможе вирішувати можливі проблеми та швидко знайти відповіді на запитання.

- Відгуки користувачів: Треба прочитатати відгуки від інших користувачів, які вже використовували цей шаблон. Це дозволить отримати додаткові відомості про його переваги та недоліки.
- Мобільна сумісність: Треба переконатися, що шаблон є адаптивним і добре виглядає на різних пристроях, включаючи мобільні телефони та планшети.
- Розширення та плагіни: Треба перевірити, які розширення та плагіни сумісні з обраним шаблоном. Це може бути важливо, якщо планується розширювати функціонал свого додатку.

Загалом, ретельне дослідження та збір відповідної інформації допоможуть знайти шаблон, який найкращим чином відповідає потребам та вимогам проекту.

3.4. Верстка макету

Розробка web-додатку з такою структурою є добрим підходом для забезпечення зручності та логічності навігації користувачів. Давайте розглянемо кожну сторінку окремо і їхні основні елементи:

1. Сторінка "Головна":
    - Логотип клубу: Інтеграція логотипу клубу у верхній частині сторінки.
    - Основна інформація: Відображення ключової інформації про волонтерський проект захисту тварин або іншу відповідну інформацію.
2. Сторінка "Про нас":
    - Логотип клубу: Продовження використання логотипу клубу у верхній частині сторінки.
    - Інформація про клуб: Опис діяльності, цілей та мети клубу, а також інформація про команду, яка його очолює або волонтерів, які в ньому беруть участь.
3. Сторінка "Фотогалерея":
    - Логотип клубу: Продовження використання логотипу клубу у верхній частині сторінки.

- Фотографії: Відображення фотографій, пов'язаних з діяльністю клубу або захистом тварин. Фотографії можуть бути розбиті на категорії або альбоми для зручності навігації.

4. Сторінка "Цікаве":
    - Логотип клубу: Продовження використання логотипу клубу у верхній частині сторінки.
    - Цікаві факти: Відображення цікавих фактів, пов'язаних з тваринами, захистом тварин або іншими відповідними темами. Можливо, використання ілюстративного матеріалу.

Меню навігації (зазвичай розташоване в верхній частині додатку і містить основні розділи)

Отже, основними елементами додатку є:

1. Форма для створення профілю (текстові поля для вводу даних та кнопка збереження)

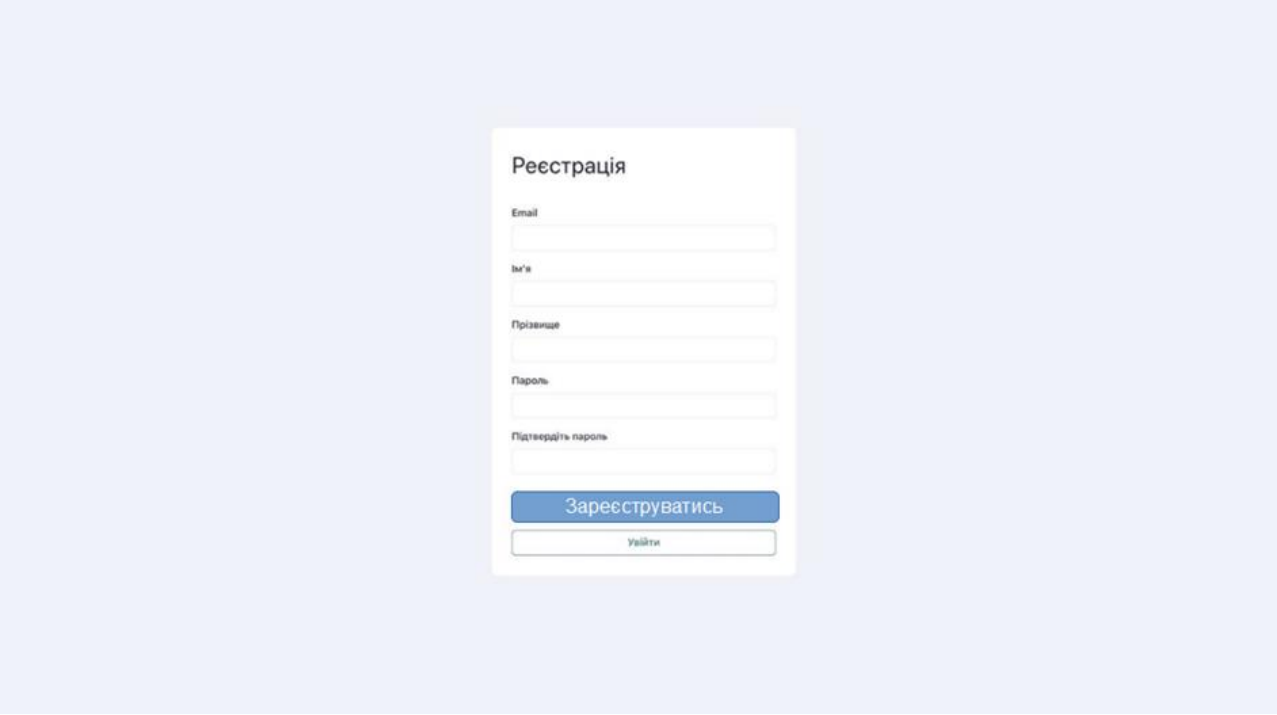

Рис. 3.3. Сторінка реєстрації веб-додатку допомоги тваринам

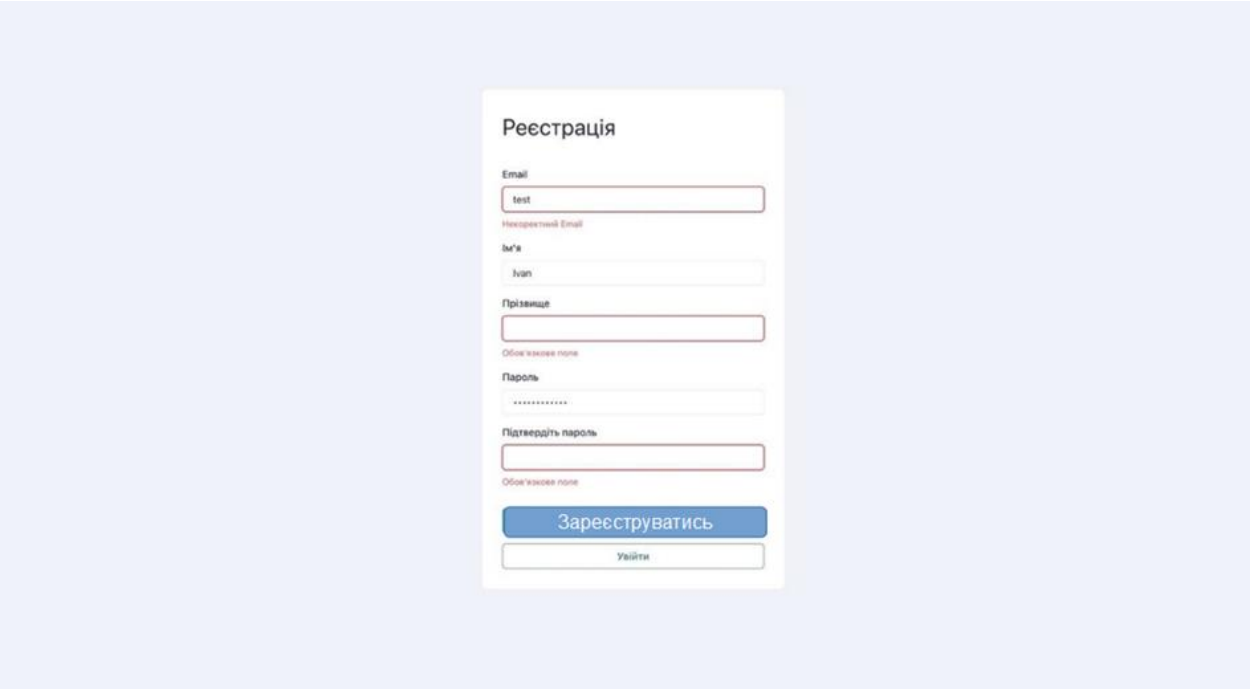

Рис. 3.4. Заповнення реєстраційної форми веб-додатку допомоги тваринам

2. Логотип
3. Головний екран (домашня сторінка з основною інформацією або списком функцій)

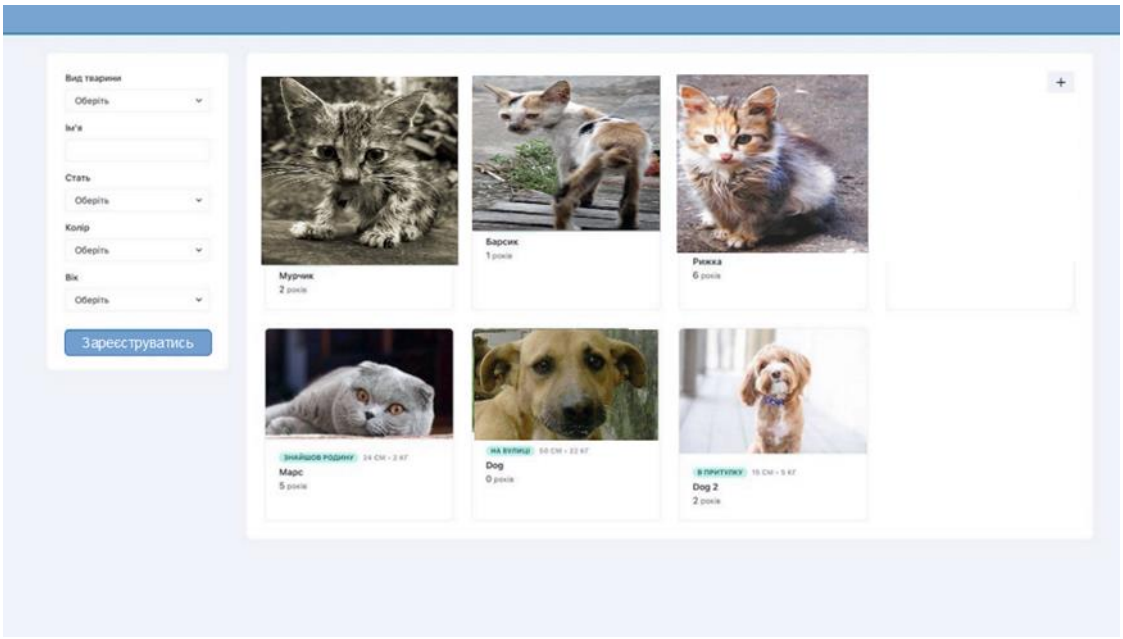

Рис. 3.5. Сторінка з основною інформацією веб-додатку про допомогу тваринам

4. Взаємодія з користувачем (можливість введення тексту, вибору опцій, завантаження файлів тощо)

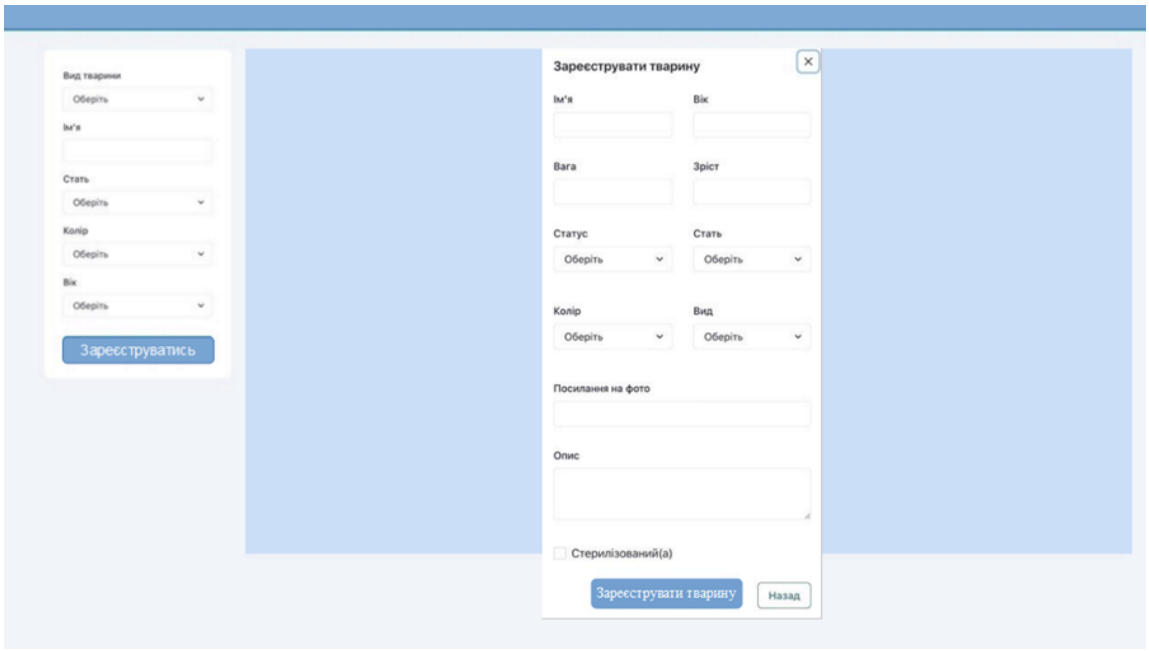

Рис. 3.5. Сторінка взаємодії з користувачем (можливість реєстрації тварини в системі з основною інформацією веб-додатку про допомогу тваринам)

5. Система управління даними (збереження, оновлення, вилучення даних)
6. Налаштування (можливість зміни параметрів додатку, налаштування вигляду, поведінки, аккаунту тощо)
7. Повідомлення і сповіщення (можливість надсилання та отримання повідомлень або сповіщень)
8. Автентифікація та безпека (система аутентифікації користувачів, захист даних)
9. Інтеграція з іншими сервісами або API (можливість підключення до зовнішніх сервісів для отримання додаткової функціональності)

Звичайно, конкретні елементи та їх вигляд можуть варіюватися залежно від типу додатку, його цільової аудиторії та функціональності. Це лише загальні елементи, які часто зустрічаються у багатьох додатках.

Тестування web-додатку проводилось з урахуванням параметрів: − коректна робота всіх обов’язкових функцій web-додатку; − працездатність форми зворотного

зв’язку в різних браузерах і при різних параметрах дозволу екрану; − коректна робота пошуку по додатку; − коректна робота гіперпосилань; − коректна робота форми розсилки; − коректна робота адаптивності, тобто відповідності контенту сторінок web-додатку на різних моніторах.

3.5. Висновок до розділу 3

1. Важливо враховувати, що розробка додатку на базі WordPress вимагає розуміння платформи, а також налагодження безпеки. Оскільки WordPress є популярною платформою, він може бути вразливим перед можливими кібератаками.
2. Розробка web-додатку з такою структурою є чудовим підходом для забезпечення зручності та логічності навігації користувачів. Основні елементи web-додатку:
    - Сторінка "Головна":
        - Логотип клубу: Інтеграція логотипу клубу у верхній частині сторінки.
        - Основна інформація: Відображення ключової інформації про волонтерський проект захисту тварин або іншу відповідну інформацію.
    - Сторінка "Про нас":
        - Логотип клубу: Продовження використання логотипу клубу у верхній частині сторінки.
        - Інформація про клуб: Опис діяльності, цілей та мети клубу, а також інформація про команду, яка його очолює або волонтерів, які в ньому беруть участь.
    - Сторінка "Фотогалерея":
        - Логотип клубу: Продовження використання логотипу клубу у верхній частині сторінки.

- Фотографії: Відображення фотографій, пов'язаних з діяльністю клубу або захистом тварин. Фотографії можуть бути розбиті на категорії або альбоми для зручності навігації.

- Сторінка "Цікаве":
  - Логотип клубу: Продовження використання логотипу клубу у верхній частині сторінки.
  - Цікаві факти: Відображення цікавих фактів, пов'язаних з тваринами, захистом тварин або іншими відповідними темами. Можливо, використання ілюстративного матеріалу.

ВИСНОВКИ

У кваліфікаційній роботі виконано розробку web-додатку допомоги тваринам. Під час роботи досліджено та проаналізовано головні аспекти інформаційних джерел, а саме тематичні сайти, що є діючими в мережі Інтернет та досліджено предметну область.

В результаті проведеного аналізу предметної області та обговорення актуальності проблеми, сформовано мету та необхідні задачі роботи, обрано засоби реалізації проекту та створено технічне завдання до розробки web-додатку.

Спираючись на проведений аналіз, метою кваліфікаційної роботи стала розробка web-додатку допомоги тваринам , а засобом реалізації обрано платформу-конструктор Wordpress.

В результаті планування та розробки був створений web-додаток «Допомога тваринам», що включає в себе такі сторінки, як «Головна», «Про нас», «Фотогалерея», «Тварини», «Контакти», та надає можливість користувачу отримувати корисну та цікаву інформацію про тварин, що потребують допомоги.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Використання уніфікованої мови моделювання. [Електронний ресурс]. – режим доступу: http://iwanoff.inf.ua/oop_ua/LabTraining05.html (дата звернення: 23.05.2023). – Назва з екрана.

2. Robert Martin Clean Code: A Handbook of Agile Software Craftsmanship [Electronic resource]. – Access mode: https://www.amazon.com/Clean-Code-Handbook-SoftwareCraftsmanship/dp/0132350882 (lastaccess: 23.05.2023). – Title from the screen.

3. React Native Official Documentation [Electronic resource]. –Access mode: https://reactnative.dev/docs/environmentsetup (lastaccess: 26.05.2023). – Title from the screen.

4. Bonnie Eisenman Learning React Native: Building Native Mobile Apps with JavaScript – Second edition [Electronic resource]. –Access mode: https://www.amazon.com/Learning-React-NativeBuilding-JavaScript/dp/1491989149 (lastaccess: 26.05.2023). – Title from the screen.

5. Ethan Brown Web Development with Node and Express: Leveraging the JavaScript Stack – Second edition [Electronic resource]. –Access mode: https://www.amazon.com/Web-DevelopmentNode-Express-Leveraging/dp/1491949309 (lastaccess: 27.05.2023) . – Title from the screen.

6. Eric Masiello Mastering React Native [Electronic resource]. –Access mode: https://www.amazon.com/MasteringReact-Native-Eric-Masiello/dp/1785885782 (lastaccess: 28.05.2023). – Title from the screen.

7. Emilio Rodriguez Martinez React Native Blueprints: Create eight exciting native cross-platform mobile applications with JavaScript [Electronic resource]. –Access mode: https://www.amazon.com/React-Native-Blueprints-cross-platformapplications/dp/1787288099 (lastaccess: 28.05.2023). – Title from the screen.

8. Mateusz Grzesiukiewicz Hands-On Design Patterns with React Native: Proven techniques and patterns for efficient native mobile development with JavaScript – First

edition [Electronic resource]. –Access mode: https://www.amazon.com/Hands-Design-Patterns-React-Nativeebook/dp/B07FSRYMH8 (lastaccess: 28.05.2023). – Title from the screen.

9. Dan Ward React Native Cookbook: Recipes for solving common React Native development problems – Second edition [Electronic resource]. –Access mode: https://www.amazon.com/React-NativeCookbook-Industry-Development-ebook/dp/B079TVH9C3 (lastaccess: 02.06.2023). – Title from the screen.

10. Nader Dabit React Native in Action – First edition [Electronic resource]. –Access mode: https://www.amazon.com/React-Native-Action-Nader-Dabit/dp/161729405552 (lastaccess: 02.06.2023). – Title from the screen.

11. Frank Zammetti Practical React Native: Build Two Full Projects and One Full Game using React Native – First edition [Electronic resource]. –Access mode: https://www.amazon.com/PracticalReact-Native-Build-Projects/dp/148423938 (lastaccess: 02.06.2023). – Title from the screen.

12. Eric Bush JavaScript Applications with Node.js, React, React Native and MongoDB: Design, code, test, deploy and manage in Amazon AWS [Electronic resource]. –Access mode: https://www.amazon.com/dp/0997196661?tag=uuid10-20 (lastaccess: 02.06.2023). – Title from the screen.

13. Eric Bush Node.js, MongoDB, React, React Native Full-Stack Fundamentals and Beyond [Electronic resource]. –Access mode: https://www.amazon.com/dp/0997196688?tag=uuid10-20 (lastaccess: 03.06.2023). – Title from the screen.

14. Ethan Holmes, Tom Bray Getting Started with React Native [Electronic resource]. –Access mode: https://www.amazon.com/dp/1785885189?tag=uuid10-20 (lastaccess: 03.06.2023). – Title from the screen.

15. Redux Toolkit Official Documentation [Electronic resource]. –Access mode: https://reduxtoolkit.js.org/introduction/getting-started (lastaccess: 03.06.2023). – Title from the screen.

16. React Native Overview [Electronic resource]. – Access mode: https://geekflare.com/react-native-for-mobile-app/ (lastaccess: 03.06.2023). – Title from the screen.

17. What is HTML and How Does Hypertext Markup Language Work? [Electronic resource]. –Access mode: https://www.theserverside.com/definition/HTML-Hypertext-Markup-Language (lastaccess: 03.06.2023). – Title from the screen.

18. What is CSS? – Tutorialspoint [Electronic resource]. –Access mode: https://www.tutorialspoint.com/css/what_is_css.htm (lastaccess: 03.06.2023). – Title from the screen.

ДОДАТКИ

Додаток А

ТЕХНІЧНЕ ЗАВДАННЯ на створення Web-додатку «Допомога тваринам»

1. Призначення й мета створення web-додатку

1.1. Призначення web-додатку

Веб-додаток призначений для того, щоб волонтери та мешканці могли поділитися цікавою та корисною інформацією з потенційно зацікавленими користувачами, домогти тваринам, представити їх у вигідному ракурсі для адоптації.

1.2. Мета створення веб-додатку

Метою проекту є створення інформаційного простору для використання його користувачами для обміну досвідом та розповсюдження інформації, що може бути корисною та допомоги тваринам.

2. Вимоги до веб-додатку в цілому

2.1. Вимоги до структури й функціонування веб-додатку

Інформаційна система повинна бути реалізована у вигляді сайту, доступного в мережі Інтернет. Сайт повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

2.2. Вимоги до персоналу

Для підтримки сайту й експлуатації веб-інтерфейсу системи керування сайтом від персоналу не повинно вимагатися спеціальних технічних навичок, знання технологій або програмних продуктів, за винятком загальних навичок роботи з персональним комп’ютером і стандартним веб-браузером (наприклад, Google Chrome, тощо).

2.2.1. Вимоги до стилістичного оформлення сайту

Дизайн повинен бути привабливим та цікавий, в тонах, що привертають увагу.

3. Основні вимоги

3.1. Структура веб-додатку

Веб-додаток повинен складатися з наступних розділів:

1) сторінка «Головна» – містить оголошення, рекламу, посилання на магазини для тварин та ветеринарні клініки;
2) сторінка «Про нас» – містить інформацію про головні цілі організації, про мету створення саме цього WEB-додатку та іншу офіційну інформацію;
3) сторінка «Фотогалерея» – тут розміщуються фотоматеріали, що надходять на електронну адресу від членів клубу, волонтерів та відвідувачів;
4) сторінка «Тварини» – містить інформацію про тварин, які потребують допомоги;
5) сторінка «Контакти» – містить інформацію про адресу з визначенням місця розташування офісу клубу на гугл-карті, телефон, мейл організації і посилання в соціальних мережах.

3.2. Навігація

Інтерфейс сайту повинен забезпечувати користувачів інтуїтивно зрозуміле та наочне представлення структури розміщеної на ньому інформації, швидкий і логічний перехід до сторінок і розділів. Для цього, забезпечена наявність навігаційних кнопок на сторінках сайту. Раціонально представлена структура сайту дозволяє користувачам безперешкодно мандрувати сайтом, залишаючись впевненими в тому, що вони завжди зможуть без зусиль повернутися до раніше переглянутих сторінок.

3.3. Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам: Веб-браузер: Chrome, Firefox, Internet Explorer, Oper, тощо; Включена підтримка Flash.

3.4. Функціональні вимоги

1) перегляд статей;
2) перегляд фото та відео контенту;
3) перегляд соціальних мереж клубу;
4) можливість розповсюджувати та ділитися інформацією, яка є на сайті;
5) зворотній зв’язок.

3.5. Наповнення сайту (контент)

Первинна розробка та верстка контенту (інформаційного вмісту) сайту повинна проводитися силами ініціативної волонтерської групи. Розробники обирають текстові та графічні матеріали, а також формують змістове наповнення сайту, його обсяг, оформлення і супровід. Сайт має бути україномовним. Має мати зручну навігацію.