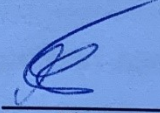


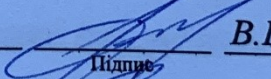
Хмельницький національний університет
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра комп'ютерних наук та інформаційних технологій

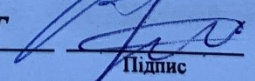
КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Інформаційна система забезпечення життєдіяльності притулку
для тварин

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент 4 курсу, група КН-17-2  Т.Б. Мариніч
Курс, група виконавця Підпис Ініціали, прізвище

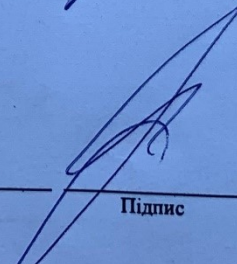
Керівник: к.фіз-мат.н. доцент каф. КНІТ  В.Ц. Михалевський
Науковий ступінь, посада Підпис Ініціали, прізвище

Нормоконтроль: к.т.н., доцент каф. КНІТ  Р.О. Багрій
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КНІТ, д.т.н., професор

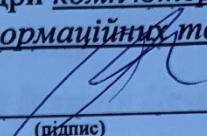
08 сервня 2021 р.

 О.В. Бармак
Підпис Ініціали, прізвище

Хмельницький 2021

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
 Факультет програмування та комп'ютерних і телекомунікаційних систем
 Кафедра комп'ютерних наук та інформаційних технологій
 Освітній ступінь бакалавр
 Галузь знань 12 – Інформаційні технології
 Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ
 Завідувач кафедри комп'ютерних наук
та інформаційних технологій


 (підпис)
 д.т.н., професор О.В. Бармак
 «05» лютого 2021 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Інформаційна система забезпечення життєдіяльності притулку для тварин»

2. Завдання видано студенту Маринічу Тимофію Богдановичу
 (прізвище, ім'я, по батькові)

3. Керівник роботи доцент Міхалевський Віталій Цезарійович
 (посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від «05» лютого 2021 р. № 11

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи кваліфікаційної роботи є - створення інформаційної системи забезпечення життєдіяльності притулку для тварин, що полегшить роботу працівникам притулку та сприятиме збільшенню кількості забраних вихованців з притулку.

Виконавець: студент 4 курсу, група КН-17-2
 Курс, група виконавця


 Підпис

Т.Б. Мариніч
 Ініціали, прізвище

Керівник: к.фіз-мат.н., доцент
 Науковий ступінь, посада


 Підпис

В.Ц. Міхалевський
 Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи: «Інформаційна система забезпечення життєдіяльності притулку для тварин»

Автор кваліфікаційної роботи: Марніч Тимофій Богданович

Керівник кваліфікаційної роботи: доцент Михалевський Віталій Цезарійович

Кваліфікаційна робота містить:

Пояснювальна записка, стор.	Кількість рисунків	Кількість таблиць	Кількість джерел інформації	Кількість додатків
55	34	2	14	2

Метою розробки кваліфікаційної роботи є створення інформаційної системи забезпечення життєдіяльності притулку для тварин, що полегшить роботу працівникам притулку та сприятиме збільшенню кількості забраних вихованців з притулку.

Студент групи КН 17-2


(підпис, дата)

Марніч Т.Б.

Зміст

Перелік скорочень	5
Вступ	6
Розділ 1 Характеристика предметної області і постановка задачі	7
1.1 Аналіз предметної області	7
1.2. Аналіз інформаційного забезпечення предметної області	10
1.3 Аналіз сучасних засобів створення програмного забезпечення	13
1.4 Постановка задачі	21
Розділ 2 Проектування структури інформаційної системи	23
2.1 Функціональна структура та бізнес-процеси системи	23
2.2 Інформаційна структура системи	24
2.3 Вибір засобів розробки інформаційної системи	28
2.3.1 Вибір мови програмування	28
2.3.2 Вибір бібліотек	29
2.3.3 Вибір системи керування базами даних	32
2.3.4 Вибір мови розмітки та стилю	36
Розділ 3 Програмна реалізація	40
3.1 Структура і функціональне призначення модулів системи, їх взаємозв'язок	40
3.2 Розробка програмних модулів	42
3.3 Інструкція користувача	48
3.4 Вимоги до апаратних та програмних засобів	53
Висновки	54
Перелік посилань	55
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
JS	Java Script
КН	Комп'ютерні науки
SSR	Server Site Rendering

Вступ

Метою розробки кваліфікаційної роботи є створення інформаційної системи забезпечення життєдіяльності притулку для тварин, який полегшить роботу працівникам притулків та сприятиме збільшенню кількості забраних вихованців з притулку.

Розробка буде виконуватися за допомогою бібліотеки React, на платформі JavaScript.

Інформаційні технології - це певна система методів, процесів та різних використань обчислювальної техніки і систем зв'язку для конкретних дій, а саме : створення, збір, передача, пошук, оброблення та поширення інформації з метою більш ефективної організації діяльності людей.

Інформаційна система (ІС) - система, призначена для зберігання, пошуку та обробки інформації, і відповідні організаційні ресурси (людські, технічні, фінансові та т. Д.), Які забезпечують і поширюють інформацію

Отже, звідси можна зробити висновок, що використання інформаційних систем покращить роботу притулків.

Розділ 1

Характеристика предметної області і постановка задачі

1.1 Аналіз предметної області

Актуальною є проблема безпритульних тварин, оскільки щороку росте чисельність некерованих або ж, залишених на смітниках та інших безлюдних місцях, тварин. Вони представляють собою небезпеку, так як можуть поводитись агресивно, якщо її спровокувати.

Створення подібної системи допоможе захистити їх від негуманного вбивства, також люди матимуть змогу допомагати тваринам, зокрема: спонсори чи небайдужі зможуть матеріально утримувати тварин, адже тепер вони будуть знати, куди конкретно звертатись, а саме: забезпечувати їжею, надавати можливість медичної допомоги, при потребі та, звичайно, знаходити та забирати додому нових вихованців у притулку.

Притулок для тварин - тимчасовий або постійний захист тварин, які опинилися для непризначеному для них місці. [1]

Відповідно до Закону України «Про захист тварин від жорстокого поводження» від 21 лютого 2006 року №3447-IV:

1. тварини – це істоти, що належать до живих організмів відносяться до фауни : птаці, домашні, зоопаркові, лабораторні та циркові;
2. домашні тварини – собаки, коти та інші тварини, які приручені людьми та довгий час живуть у середовищі та не мають життєздатних диких популяцій.
3. сільськогосподарські тварини - тварини, що тримають люди для отримання сировини та продуктів;
4. безпритульні тварини – тварини, що залишилися самі на вулиці, які вже прожили у людей або призначенні тільки для життя у середовищі з людиною.
5. експериментальна тварина – призначення для проведення різних аналізів та дослідів; [2]

Загальні вимоги та механізм проведення організації робіт з утримання домашніх тварин у притулках встановлено наказом Державного комітету ветеринарної медицини України «Про затвердження Ветеринарно-санітарних вимог до утримання тварин у притулках» від 15 жовтня 2010 року № 439.

Відповідно до даного Положення:

1. Притулок призначений для створення певних умов перебування тварин у притулку. Діяльність притулку повина:

- проводити різні заходи
- просвітницьку діяльність
- забезпечити виконання різних державних програм
- слідкувати за численістю безпритульних тварин

У своїй діяльності притулок повинен дотримуватися встановлених ветеринарно-санітарних, протипожежних вимог;

2. Тварині у притулку надається ветеринарна допомога

3. Притулок повинен обов'язково мати облаштовані приміщення (відділення):

- a. приймальне відділення;
- b. карантинне відділення;
- c. ізолятор;
- d. відділення з утримання тварин.

4. Має вестися журнал реєстрації про кожну тварину яка надійшла до притулку та карта обліку;

5. Тварини після встановлення на облік знаходяться до 14 днів на карантині.

6. Коли тварина знаходиться на карантині вона має бути обов'язково оглянута лікарем ветеринаром [3]

Враховуючи положення Закону України «Про захист тварин від жорстокого поводження» від 21 лютого 2006 року №3447-IV встановлюються загальні правила утримання тварин, що виключають жорстокість. До них відносять:

1. Умови утримання мають бути відповідні з їхнім видом та біологічною необхідністю;
2. Утримання тварин у притулку мають повністю відповідати потребі тварини у їжі, сну тощо.;
3. Наявна кількість тварин у притулку не повинна перевищувати кількості ніж може задати місце притулок;
4. Притулок має бути повністю забезпеченим для природнього перебуванням тварин у ньому;
5. Тварини можуть перебувати у дитячих закладах та інших установах відповідно до вимог Закону.

Також варто наголосити на тому, що при утриманні тварин у притулку повинні бути забезпечені:

1. Притулок повинен дотримуватися належних санітарних умов утримання тварин;
2. Дотримання вимог законодавства з питань поводження з тваринами;
3. Мають бути проведені певні заходи для запобігання розмноження безпритульних тварин;
4. Належні умови щодо своєчасного надання ветеринарної допомоги.

У притулку повинні щоденно має проводитись:

1. огляд тварин;
2. огляд вольєрів;
3. годування тварин;
4. миття посуду для тварин;
5. миття і дезінфекція вольєрів;
6. миття і дезінфекція місць для вигулу і підсобних приміщень;
7. вигул тварин, які потребують соціалізації та психологічної реабілітації.[2]

Притулки для тварин можуть існувати у різноманітних виглядах:

1. Існує багато благодійних організацій який функціонує за рахунок благодійних внесків громадянами та за рахунок коштів спонсорів
 2. Створюються державні установи завдяки адміністрації міста, у яких перебувають безпритульні тварини. Також ці установи займаються виготовленням корма та надають платні консультації та послуги
 3. Також існують домашні притулки на офіційному рівні які знаходяться у квартирах, будинках та приватних секторах. Діють повністю на добровільній основі.
- [4]

Враховуючи вищевказане можна дійти до висновку, що створення інформаційної системи притулку для тварин є нагальною потребою сьогодення, оскільки є потреба у автоматизації та контролю за виконанням усіх умов, регламентованих наказом Державного комітету ветеринарної медицини України «Про затвердження Ветеринарно-санітарних вимог до утримання тварин у притулках» від 15 жовтня 2010 року № 439, Законом України «Про захист тварин від жорстокого поводження» від 21 лютого 2006 року №3447-IV та ін., й можливості допомогти матеріально та безпосередньо подарувавши тваринам дім.»

1.2. Аналіз інформаційного забезпечення предметної області

Головна мета притулку - це врятування тварини з подальшим прилаштуванням її в люблячу сім'ю. Саме тому важливою складовою притулку має бути власний сайт з актуальною інформацією про тварин, які там перебувають.

На жаль, зараз не далеко не всі притулки мають власний інформаційний портал, що дає можливість сформувавши думку про негативну динаміку поширення безпритульних тварин у містах.

Було розглянуто вже існуючі інформаційні портали щодо безпритульних тварин. В основному переваги таких сайтів у тому, що сидячи вдома люди можуть допомогти притулку коштами за допомогою «донатів» та переглядом карток тварин для вибору вихованців онлайн.

Основні недоліки полягають у тому, що існує дуже мало таких сайтів та лише незначний відсоток внесений у карточки, які можливо переглянути, а саме тому із кожним роком постає все більша проблема фіксації нових вихованців, проведення їх стерилізації. У сучасних умовах відвідування безпосередньо притулків викликає певні складнощі, а тому створення таких онлайн порталів допоможе здійснювати допомогу кожному бажаючому лише у пару кліків, обирати нових вихованців онлайн та тільки тоді їх забирати із притулку.

Для аналізу було обрано інформаційний портал «Притулок для тварин Сіріус».

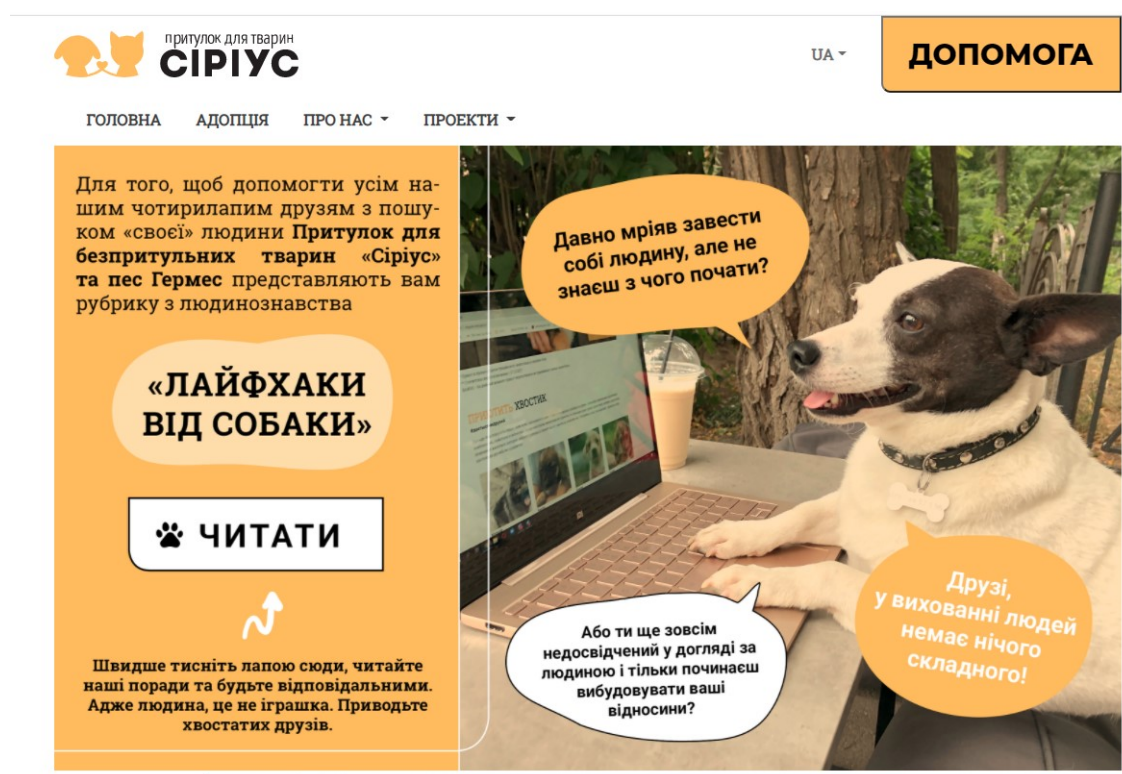
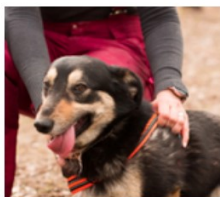


Рисунок 1.1 – Сторінка сайту «Сіріус»

Цей сайт несе в собі вирішення основних проблем притулків, а саме полегшення людям знайти собі вихованця та формування додаткового бюджету притулку.

перш ніж купувати собаку чи когось з інших тварин, зайдіть до нас. У притулку завжди є інші друзі, буде кому до душі – це компаньйони, лівивці та непосидьки. Коли ми беремо тваринку з притулку, то рятуємо одразу трьох: тваринку, бездомну тварину з вулиці, що займе звільнене місце у притулку, і, звичайно, Людину в собі. Не купуйте тваринку, якщо друга не продається!

ДОПОМОГА



Соня
Дівчинка / 1 рік



Спарк
Хлопчик / 2 роки



Марио
Хлопчик / 3 роки



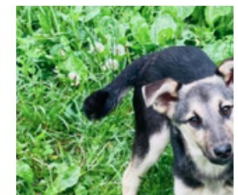
Лили
Дівчинка / 2 роки



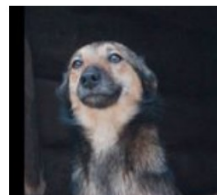
Ариель
Дівчинка / 3 місяці



Джесси
Дівчинка / 1 рік



Габи
Дівчинка / 6 місяців



Ефи
Дівчинка / 1 рік

Рисунок 1.2 - Сторінка з тваринами у притулку «Сіріус»

Основною «фічею» була окрема виведена сторінка з тваринами, саме так користувачі сайту можуть не виходячи з дому вибрати собі нового «члена сім'ї»

Досліджуючи зразки з'ясувалось, що сайтів притулків дуже мало, та не всі вони відповідають актуальності та вирішенню проблем притулків. Важливий напрям є популяризація існуючих сайтів та створення нових.

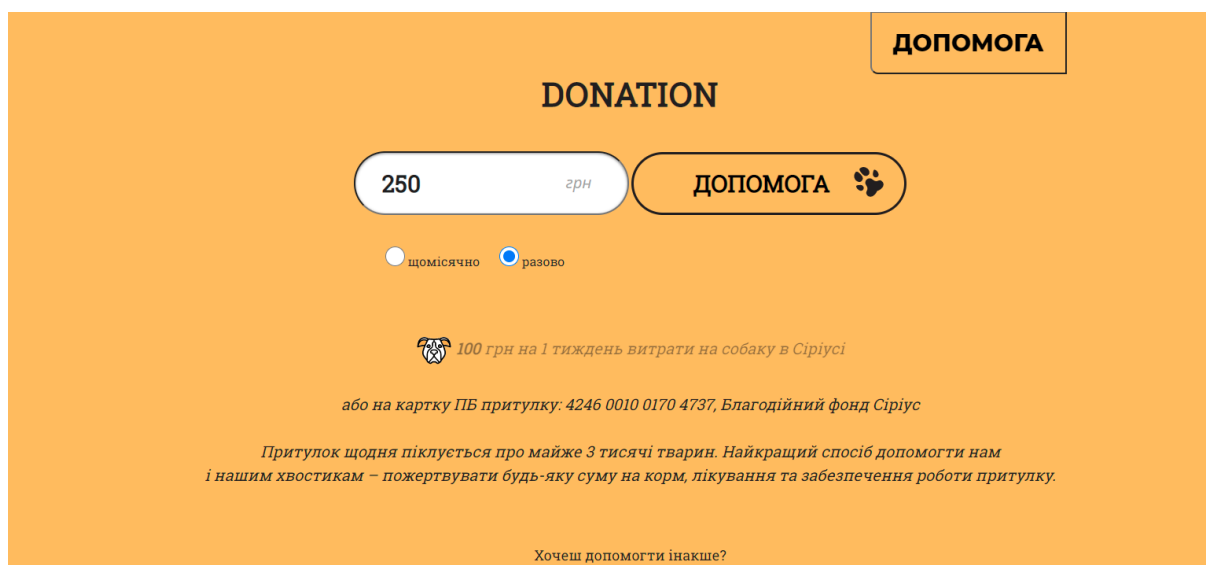


Рисунок 1.3 – Допомога

Однією з не менш важливих складових сайту Сіріуса була сторінка з вкладенням допомога. У сучасних та зайнятих людей немає часу на відвідування та у фізичній допомозі благодійним проектам, саме тому можливо «законатити» сайту певні кошти або підписатися на них, та кожен місяць автоматично будуть вираховуватися кошти на допомогу клубу. Кожен може допомагати по різному, тому саме цей варіант мені подумався найбільше.

1.3 Аналіз сучасних засобів створення програмного забезпечення

Наразі основними видами програмної платформи є: .NET, PHP, Mono, BLINK та Java. Основними типами застосування є:

- вебзастосування,
- віконне десктоп-застосуванням
- застосування для мобільних пристроїв.

.NET Framework (вимовляється як «крапкова мережа») - це програмне забезпечення, розроблене корпорацією Майкрософт, яке працює в основному на Microsoft Windows. Він включає велику бібліотеку класів, що називається Framework Class Library (FCL), та забезпечує взаємодію мови (кожна мова може використовувати

код, написаний іншими мовами) для декількох мов програмування. Програми, написані для .NET Framework, виконуються в програмному середовищі (на відміну від апаратного середовища) з назвою Common Language Runtime (CLR).

CLR - це віртуальна машина програми, яка надає такі послуги, як безпека, управління пам'яттю та обробка винятків. Таким чином, комп'ютерний код, написаний за допомогою .NET Framework, називається «керованим кодом». FCL і CLR разом складають .NET Framework.

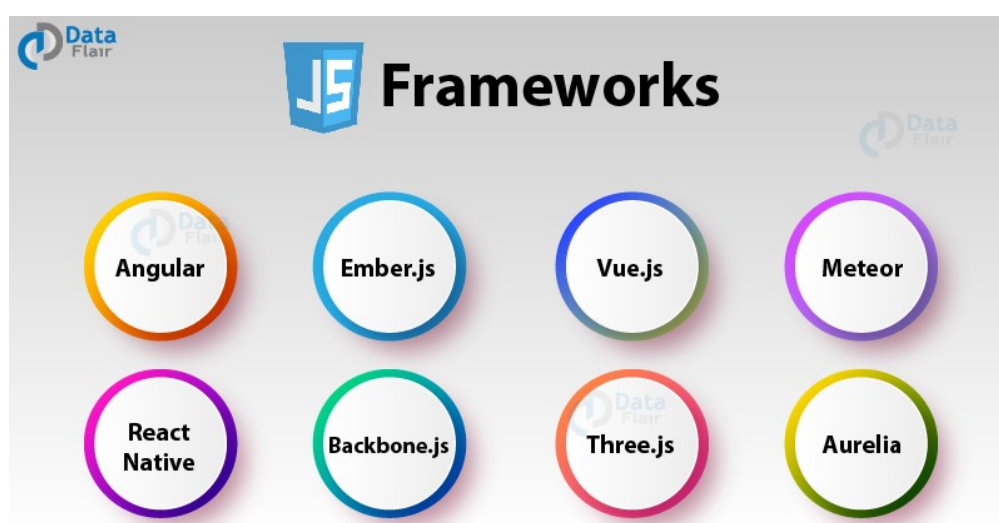


Рисунок 1.4 – різновиди фреймворків

.NET Framework була створена в 1998 році корпорацією Microsoft та Adobe Systems з метою забезпечення загальної інфраструктури операційної системи для платформ розробки Microsoft Windows (1993), SQL Server (1994) та Acrobat (1998). Основними компонентами фреймворку є CLR та FCL. .NET Framework доступний як окремий продукт, так і як набір бібліотек, що дозволяють стороннім розробникам створювати та розповсюджувати програми на багатьох мовах та технологіях програмування, таких як C # (2001), ASP. NET (2000), C ++ [14]

Основними недоліками .NET є :

1. складності в установці додатків і конфлікти між старими і новими версіями компонентів;

2. відсутність контролю коректного звільнення оперативної пам'яті при виконанні програми («втеча пам'яті»);

3. неможливість повною мірою використовувати декілька мов програмування при розробці складних додатків.

Переваги .NET:

1. повні можливості взаємодії з існуючими кодом;

2. повноцінне використання кілька мов програмування при розробці складних додатків;

3. загальна середовище виконання для всіх додатків незалежно від мови програмування, на якій розроблено програму;

4. бібліотека базових класів, яка забезпечує приховування всіх труднощів, пов'язаних з безпосереднім використанням викликів більшості прикладних API;

5. спрощення установки додатків методом копіювання файлів і відсутність конфліктів між старими і новими версіями компонентів.

Web-додатки. є застосунок, в якому клієнтом виступає браузером, а сервером - веб-сервер.



Рисунок 1.5 - різні операційні системи

Браузер може бути реалізацією так званих тонких клієнтів - логіка застосунку зосереджується на сервері, а функція браузера полягає в основному у відображенні інформації, завантаженої мережі з сервера, і передачі назад даних користувача. [5]

Основною перевагою такого підходу є той факт, що клієнти не залежать від конкретної операційної системи користувача, тому веб-застосунки є між платформними сервісами. Унаслідок цієї універсальності і відносної простоти розробки веб-застосунки стали широко популярними в кінці 1990-х - початку 2000-х років. [6]

На даний момент усі операційні системи використовують веб-браузер як клієнт, навіть ОС моєї компанії. У наш час ви можете знайти багато людей, які відходять від веб-браузера та працюють із програмами об'єктно-орієнтованим чином. У розробці ОС ми знаходимо можливість підтримати такий тип моделі розвитку. Об'єктно-орієнтована система реалізована програмним забезпеченням сервера, а клієнт реалізований у веб-браузері.

Основний плюс цього провадження є те, що клієнти не відносяться до однієї операційної системи, саме тому є можливість використання міжплатформними сервісами. [7]

На сьогодні існує величезна кількість систем створення веб-додатків, які підрозділяють на різні категорії:

HTML - при розробці інтерфейсу важливий суцільний текстовий редактор. У часи, коли люди більше сподіваються на послуги веб-розробки, традиційні старі редактори, такі як Notepad, Visual Studio тощо, часто асоціюються з головними болями.

```

1  <!DOCTYPE HTML>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Кваст</title>
6    </head>
7    <body>
8      <div class="page-header">
9        <div class="header-top">
10       <div class="logo">
11         
12       </div>
13       <ul>
14         <li><a href="#">История</a> </li>
15         <li><a href="#">Производство</a> </li>
16         <li><a href="#">Ассортимент</a> </li>
17         <li><a href="#">Где купить</a> </li>
18       </ul>
19     </div>
20     <br>
21     <p>Крафтовый квас<br>
22     всему голова!</p>
23   </div>
24 </body>
25 </html>

```

Рисунок 1.6 - Приклад HTML – коду

Сьогодні у нас є кращий спосіб редагування HTML і CSS, і він називається HTML або Text-Editor.

Засоби створення веб-додатків, які є досить потужними: Adobe Contribute «Dreamweaver Lite», Weaver, Abaddon GoLive, Amaya, Bluevoda, HoPDoG, iWeb, Media Lab, SiteGrinder Sandvox, SeaSanya Composer, Softpress Freeway, RapidWeaver, WorldWideWeb і т.п.) – текстові редактори (Emacs, gedit, Notepad, TextEdit, UltraEdit, vi і т.п.); Для реалізації веб-сайту було обрано середовище розробки PHP, так як воно є найпопулярнішою системою розробки веб-ресурсів. [6]

Personal Home Page Tools - програма сценаріїв, призначена для створення HTML - сторінки на стороні веб-сервера, а потім вилучення з неї вмісту для клієнта. Він повинен бути простим у користуванні та потужним. Значна сила обмежена тим, що багато мов сценаріїв, насправді, не мають.

PHP - одна з найпоширеніших мов, що використовуються в Інтернеті, і вона вам знадобиться для розробки сайту в .NET Core. Раніше PHP була основною мовою, яка використовувалася для веб-розробки, але з ростом важливості ASP.NET, можливо, вам доведеться розглянути інші варіанти. PHP все ще є найбільш широко використовуваною мовою, однак із зростанням популярності інших мов, таких як Node JS та .NET Core, втрачаються деякі переваги цих опцій.

PHP підтримується в основному у хостинг-провайдерів. PHP - проект доступного програмного забезпечення.

```

30 //Преобразование в "сущности" - убирает кавычки html
31
32 $title1 = htmlspecialchars($title1, ENT_QUOTES);
33 $text1 = htmlspecialchars($text1, ENT_QUOTES);
34 $date1 = htmlspecialchars($date1, ENT_QUOTES);
35 $full_text = htmlspecialchars($full_text, ENT_QUOTES);
36 $date_nextd = htmlspecialchars($date_nextd, ENT_QUOTES);
37 $html_in = htmlspecialchars($html_in, ENT_QUOTES);
38
39 /*запись данных в базу*/
40 $link_test=mysqli_query($bd, "SELECT * FROM `php_dns`.`habra_post`");
41 if(!$link_test){
42     $bd-> query("INSERT IGNORE INTO habra_post SET title='$title1', text_body='$text1', reg_date='$date1',
43     full_text='$full_text', next_date='$date_nextd', link='$html_in'");
44     if($bd==false)
45     {
46         // echo "<br>011 ОК";
47     }
48     // else
49     // {
50     //     echo mysqli_error();
51     }
52 }}}
53
54 //Вывод инфы из MySQL
55
56 $sql_out = mysqli_query($bd,
57 "SELECT
58     `id`,
59     `title`,
60     `text_body`,
61     `reg_date`,
62     `full_text`,
63     `next_date`,
64     `link`
65 FROM
66     `php_dns`.`habra_post`");
67

```

Рисунок 1.7 - Приклад PHP – коду

PHP може передаватися веб-сервером у HTML-код, який передає інформацію на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий html-код. Це є перевага з точки зору безпеки,

але погіршує інтерактивність сторінок. Але ніхто не забороняє використовувати PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта. [4]

Одним з основних інструментів, що я використовував у своєму проєкті - це React.

JavaScript є досить компактною та гнучкою мовою. Розробники надали широкий спектр інструментів, що підтримують основу мови JavaScript, що відкриває величезну кількість додаткових функціональних можливостей з мінімальними зусиллями. Одним з таких інструментів є плагін JScript.TextMate.

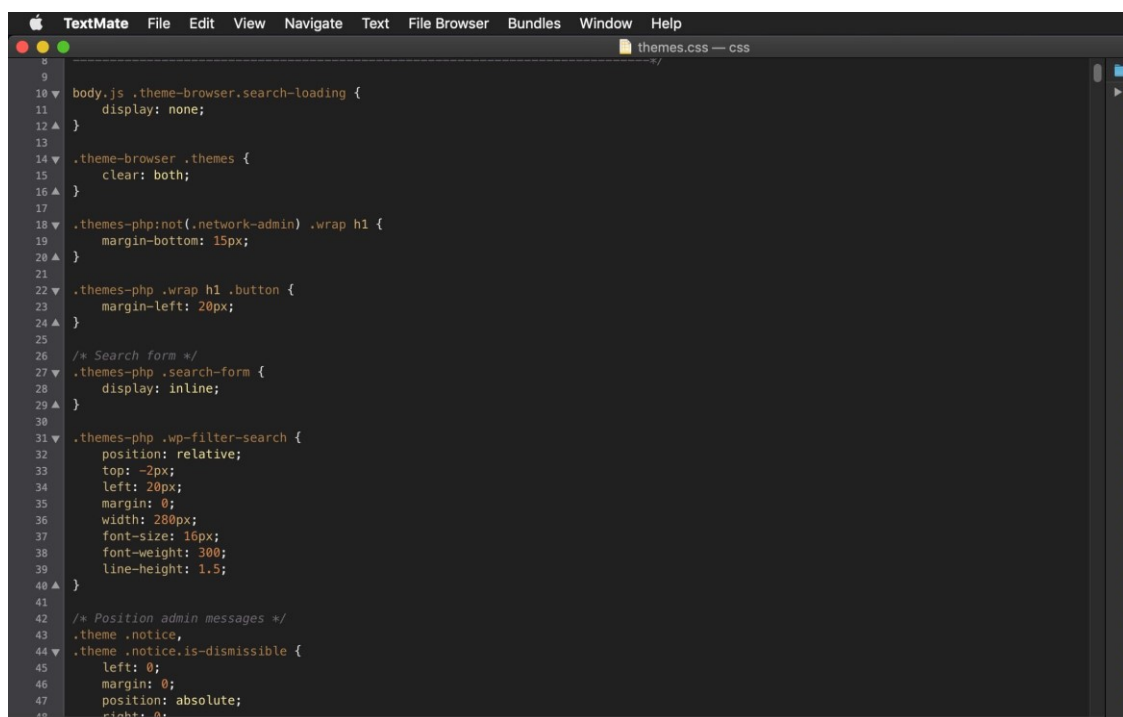
The image shows a screenshot of the TextMate text editor. The window title is 'TextMate' and the file being edited is 'themes.css'. The code is CSS, with syntax highlighting in shades of blue and green. Line numbers are visible on the left side of the editor. The code includes various CSS selectors and properties, such as 'display: none;', 'clear: both;', 'margin-bottom: 15px;', 'margin-left: 20px;', 'display: inline;', 'position: relative;', 'font-weight: 300;', and 'position: absolute;'. The editor interface includes a menu bar with options like File, Edit, View, Navigate, Text, File Browser, Bundles, Window, and Help.

Рисунок 1.8 – Невдалий код TextMate

Оскільки я особливо люблю HTML та CSS, перше, що я спробував із TextMate, це перетворити стандартні таблиці стилів у JavaScript, щоб їх можна було використовувати в додатку. Я не був дуже задоволений результатом, оскільки, наприклад, однією з найбільших таблиць стилів CSS, що використовується в Інтернеті - принаймні, згідно з Google Analytics - був файл .css Вікіпедії.

Програмні інтерфейси (APIs) для браузерів - які вже є у самих браузерах що функціонують завдякую динаміці HTML та вони використовують CSS – стилі.

Фреймворки та бібліотеки третіх осіб, які ви можете застосувати до вашого HTML, щоб прискорити створення сайтів та застосунків. [8]

Переваги JavaScript :

1. Швидкість - JavaScript :

- Є досить швидким, завдяки тому,що запускається відразу у браузері користувача.
- Не вимагає зовнішніх ресурсів
- Не затримується через виклики серверного кода
- Підтримує компіляцію ЛТ
- Не потребує компіляції перед запуском
- Простота у використанні.

2. Популярність

- JS використовується повсюду
- Використовується в основному на сервері
- Велика кількість ресурсів таких,як : StackOverFlow,GitHub

3. Сумісність

- Є можливість вставити у будь яку сторінку
- Підтримка інших мов

4. Навантаження сервера

- Зменшує попит на сервери
- Прості програми можуть бути навіть без сервера

5. Розширені інтерфейси

- Можливість додавати такі функції як : перетягування, додавати повзунки, а це значно покращу взаємодію користувача з сайтом

6. Розширена функціональність
 - Можливість написання фрагментів коду у сторонніх доповненнях
7. Універсальність
 - Існує дуже багато способів користуватися JS через сервери Node.js за допомогою Express
 - Використовувати бази даних різних документів
 - Використовувати JS на інтерфесі для клієнтів

Недоліки JavaScript:

1. Сторона клієнта забезпечує захист – іноді помилки та недоліки можуть використовуватися для зловмисних цілей
2. Підтримка браузера – Браузер Google може один і той самий код відображати по іншому на відмінну з другими браузерами. У нас час є дійсно мінімальна різниця між різними браузерами. [1]

Отже, зважаючи на наведені переваги та недоліки розгул платформ, було обрано основну платформу JavaScript.

1.4 Постановка задачі

Метою розробки дипломної роботи є створення інформаційної системи забезпечення життєдіяльності притулку для тварин на платформі JavaScript, що виконує основну функцію: надання актуальної інформації користувачам.

До групи функцій роботи з викликами та подіями належать наступні функції:

1. інформація про діяльності притулку для тварин;
2. детальний опис кожної тварини (картки);
3. робота з новинами;
4. розподілення по категоріям;
5. пошук та вибірка;

Для досягнення мети є доцільно використовувати: бібліотеку React на базі JavaScript.

Розділ 2

Проектування структури інформаційної системи

2.1 Функціональна структура та бізнес-процеси системи

Інформаційна система забезпечення життєдіяльності притулку для тварин призначена для освідомлення людей про соціальні проблеми у суспільстві та можливості допомоги притулку.

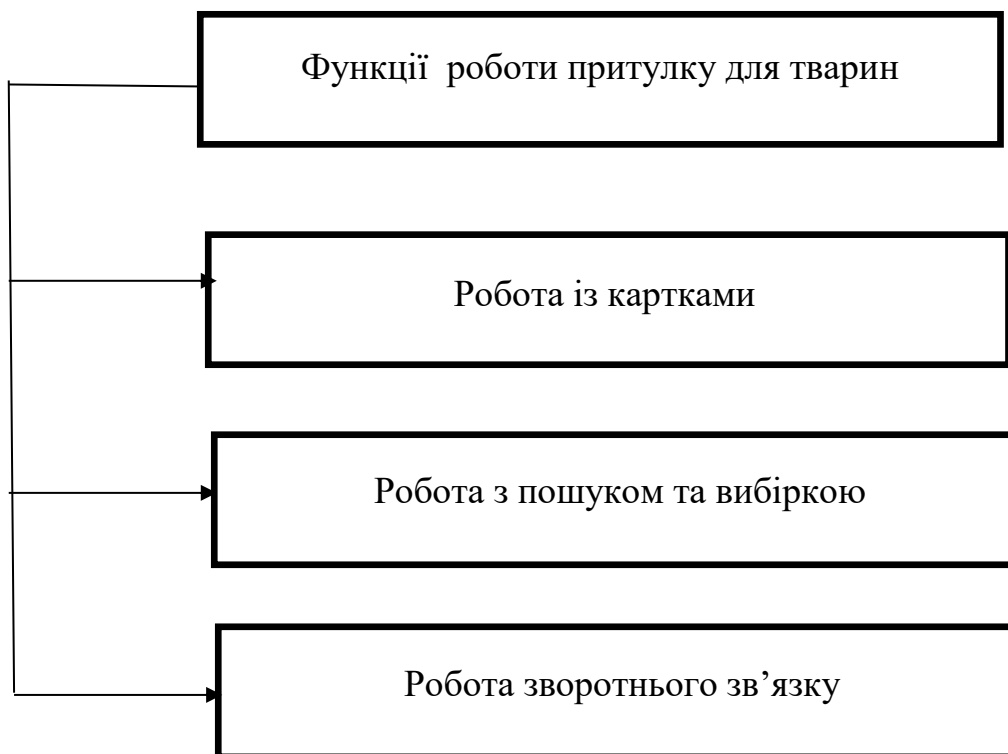


Рисунок 2.1. – Схема бізнес-процесів

Для інформаційної системи роботи з наданням інформації, пошуком, вибіркою, допомозі притулку потрібні наступні бізнес-процеси:

1. бізнес-процес роботи з картками;
2. бізнес-процес роботи з пошуком та вибіркою;
3. бізнес процес роботи зворотнього зв'язку.

Бізнес-процес «Бізнес-процес роботи з картками». Даний бізнес-процес надає можливість краще побачити та отримати більше інформації про кожну тварину та її особливості.

Зокрема, автоматизації підлягають наступні функції:

1. реєстрація нової тварини;
2. заповнення інформації стосовно неї;
3. перегляд переліку особливостей: кличка, вік, стать, розмір тварини, вид тварини.

Бізнес-процес «Бізнес-процес роботи з пошуком та вибіркою». Даний бізнес-процес надає можливість допомогти у виборі вихованця за певними критеріями. Зокрема, пошук тварини несе у собі наступні функції:

1. пошук по імені тварини;
2. вибір тварин за атрибутами.

Бізнес-процес «Бізнес-процес роботи із зворотнім зв'язком». Даний бізнес-процес надає можливість залишити про себе інформацію щодо зацікавленістю певного вихованця або благодійного внеску.

Зокрема, автоматизації підлягають наступні функції:

1. реєстрація нової особи;
2. збереження інформації про зареєстровану заявку.

2.2 Інформаційна структура системи

Головна сторінка призначена для надання актуальної інформації про притулок та новини у ньому.

Вибір резидента - сторінка призначена для отримання інформації про вихованців у клубі та пошуку і фільтрації тварин.

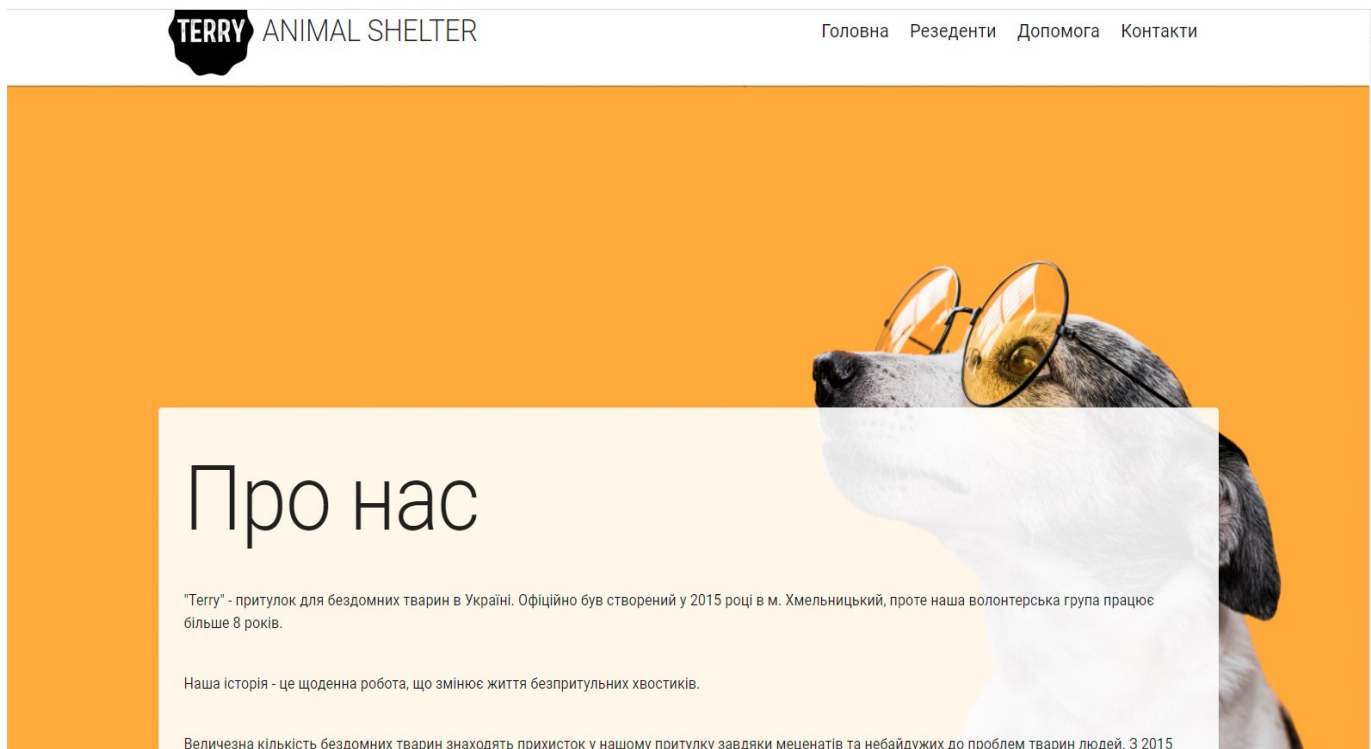


Рисунок 2.2 – Головна сторінка сайту

Пошук та фільтрація має у собі критерії : вид, розмір, стать.

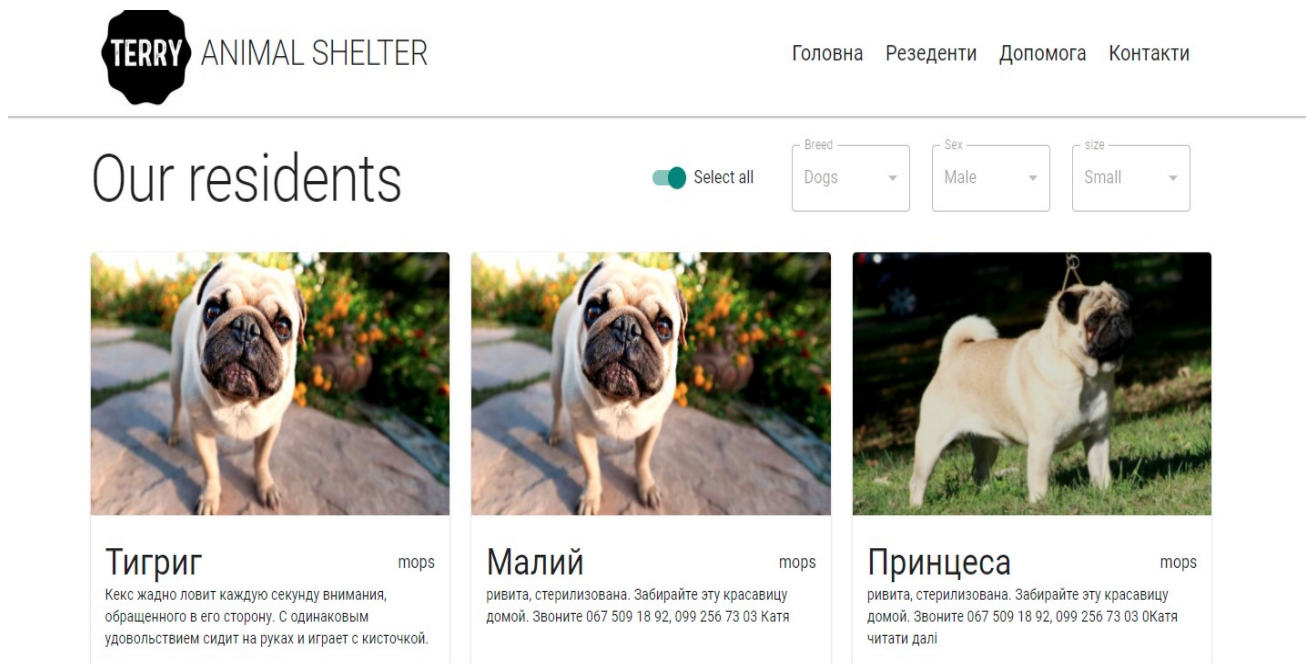
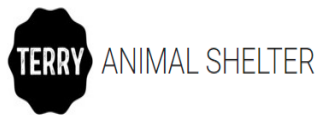


Рисунок 2.3 – Сторінка сайту з резидентами

Допомога клубу – сторінка яка містить у собі інформацію про можливість зробити внесок у забезпеченні необхідним у притулку.



Головна Резеденти Допомога Контакти



Залишилися запитання?

ПІБ *

Електронна пошта *

Номер телефону

Ваше повідомлення

ВІДПРАВИТИ

Рисунок 2.4 – Сторінка сайту «Домомога»

Контакти – сторінка у якій можливо продивитися усі соціальні мережі роботи притулку, місце розташування притулку та зворотній зв'язок з притулком.

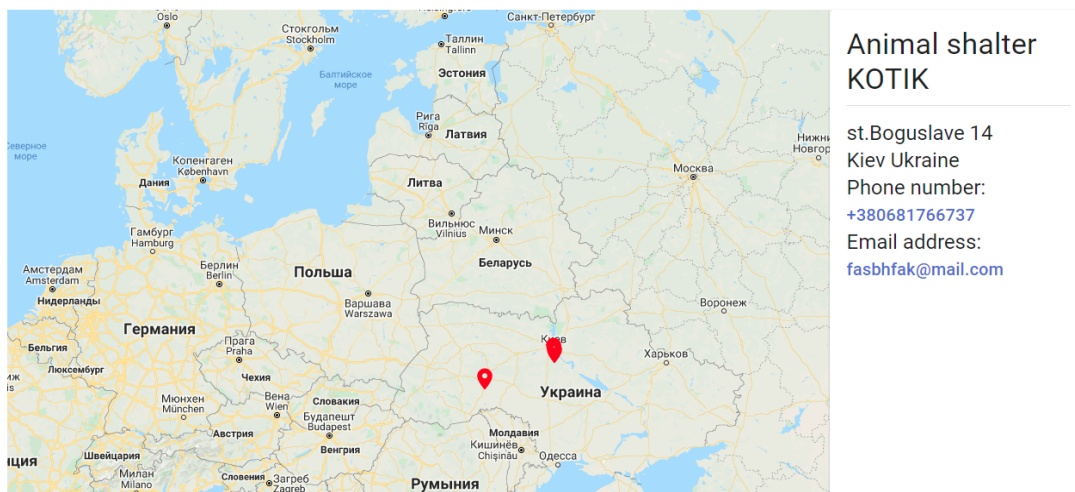


Рисунок 2.5 – Сторінка сайту «Контакти»

Програмний продукт повинен мати наступну структуру:

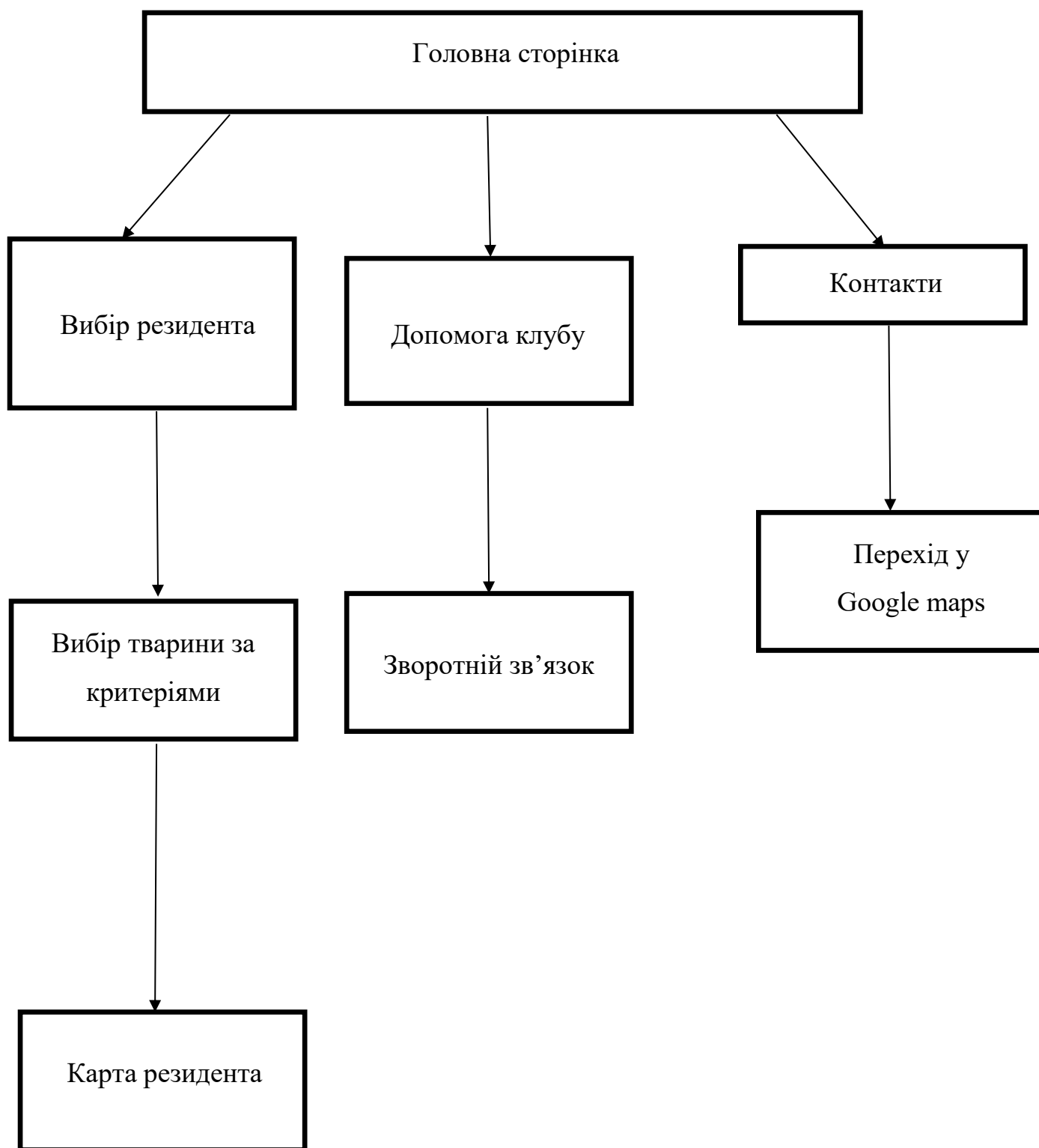


Рисунок 2.6 – Схема сторінок сайту

2.3 Вибір засобів розробки інформаційної системи

2.3.1 Вибір мови програмування

JavaScript - це динамічна, однорежимна, багатопарадигмна мова програмування.

На синтаксис мови програмування JavaScript впливали C / C ++ та Java / Java EE, і інтерпретується інтерпретатором JavaScript, написаним в даний час на JavaScript.

Призначення JavaScript полягає у створенні користувальницьких програм (таких як веб-сторінки, рідні настільні програми та динамічні сценарії).

Це дозволяє писати код динамічно, але все одно вимагає його інтерпретації інтерпретатором.

При програмуванні на JavaScript слід звернути увагу на два аспекти:

Код, що використовує ключове слово `function`, є лише синтаксичним цукром для виразу, тоді як код, що використовує ключове слово `function`, є функцією, яка оголошена в коді.

TypeScript - дозволяє відображати статично код Java. Звідси Java з класами, декораторами, типізацією все більше подібний на Java у гарному сенсі цього слова.

Особливості JavaScript.

На JavaScript створюються додатки зі сторони клієнта.

Завдяки цьому можна вивчити лише одну мову програмування та використовувати частину вже отриманих знань у `frond-end`, `Back-end` та реалізовувати свої знання у роботобудуванні, у створенні мобільних додатків, у `Desktop` – додатках `reado`

Ось, JavaScript є здебільшого усюди якою програмою ми б не користувались - забезпечує нам інтерактивність сайтів.

Так, взагалі JavaScript використовується у фронтенд-розробці. У найближчому часі JS залишиться основною на цій позиції на мою думку. Все, що створюється на щось подібне JavaScript, в кінці конвертується в JavaScript.

JavaScript разом з HTML і CSS це те, що знає кожен програміст, у якого є вже певні знання у програмуванні.

У JavaScript розвинена інфраструктура.

Наприклад:

1. бібліотеки та фреймворки для створення додатків (React, Vue);
2. збірщики (Webpack, Gulp);
3. допоміжні бібліотеки;
4. генератори статичних сайтів (Gatsby.js, Next.js).

Автоматизація Тестування на JavaScript відзначають наступні переваги:

1. Швидкість написання автотестів
2. Великий вибір фреймворків;
3. Більша взаємодія у середині команди [9]

Отже, у кожній мові програмування є свої переваги і недоліки. Для даного проєкту найкраще підходить саме ця мова програмування, так як головною ціллю є створення сайту у веб-браузері.

2.3.2 Вибір бібліотек

React (раніше React.js, ReactJS) - це відкрита бібліотека JavaScript для користувальницьких інтерфейсів, що стосується часткового оновлення вмісту веб-сторінок, що відбувається при розробці односторінкових додатків. Він натхненний фреймворком Flux для мобільних веб-додатків.

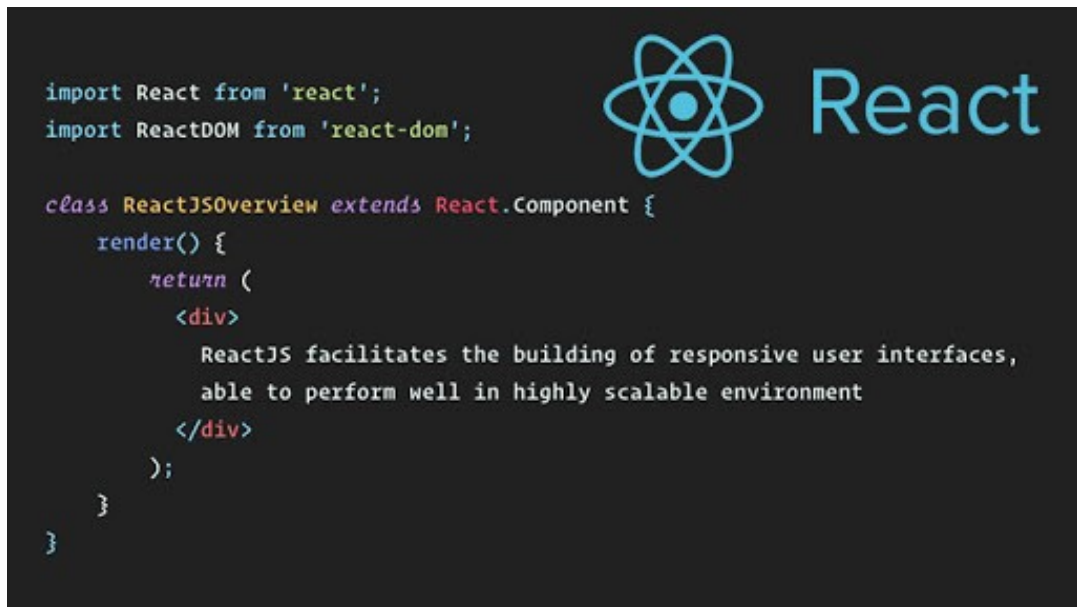


Рисунок 2.7 – Приклад використання React

React працює як бібліотека інтерфейсу користувача з частковим оновленням, де розробники маніпулюють компонентами (наприклад, піктограмою чи кнопкою) на основі взаємодії з користувачем. На відміну від Flux, який трактує компоненти як модель стану, React трактує компоненти як прості «види» фактичного стану програми. Кожен компонент «подання» може зберігати певну кількість пов'язаних змінних та вводити їх у відповідну частину свого «користувальницького інтерфейсу» або, простіше кажучи, у «область перегляду». Вигляд «записаний» у JSX, XML-подібній мові розмітки.

React дає змогу розробникам будувати великі WEB-застосунки, які обробляють дані та мають змогу змінюватися з часом . [10]

Доступний широкий набір компонентів:

1. автодоповнення;
2. кнопки;
3. картки;
4. діалогові вікна;
5. іконки;
6. меню;

7. слайдери та ін.

Material UI також пропонує React-теми та шаблони, тому ви легко можете додати користувацьку тему до вашого застосунку.

SSR - (анг. Аббревіатура від Server Side Rendering) це технологія, яка дозволяє, за допомогою Node.js, запускати JavaScript код на сервері (а не в браузері як зазвичай) і готовий результат відправляти користувачеві, уникаючи зайвого навантаження на його браузер і комп'ютер.

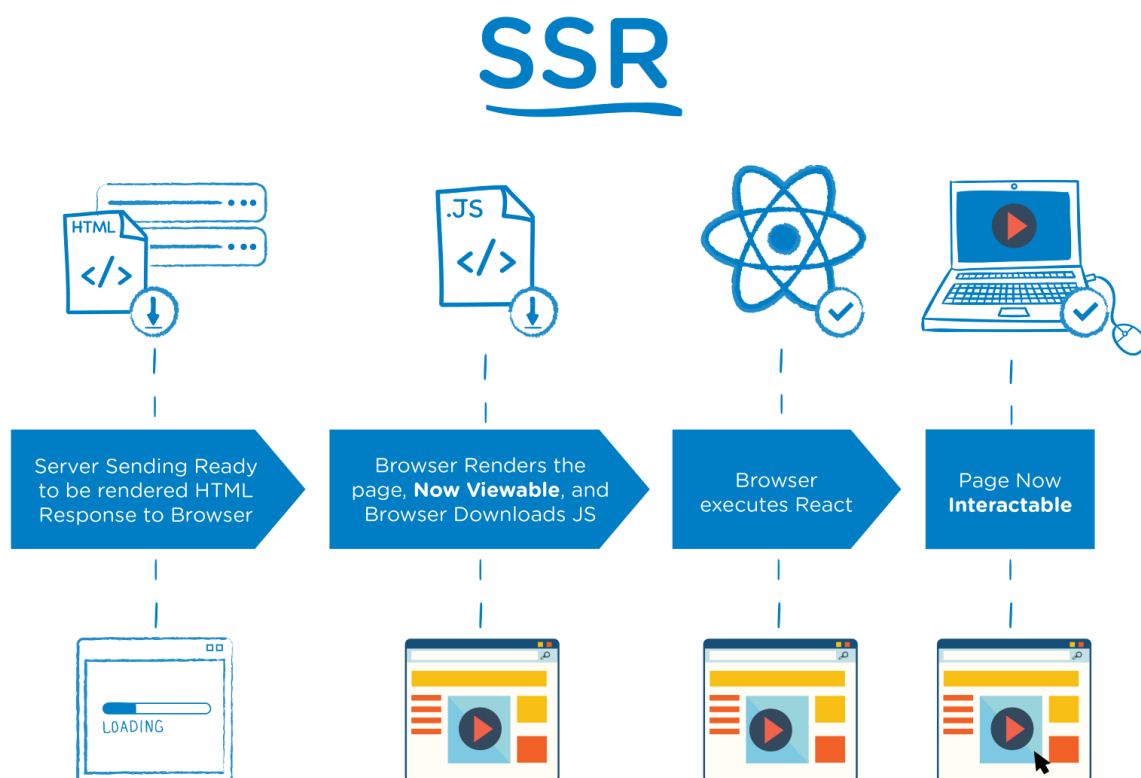


Рисунок 2.8 – Послідовність SSR

Коли користувач заходить на сайт, зроблений за допомогою JavaScript, то всі розрахунки і отрисовка графічних елементів покладається на його браузер.

З одного боку, це допомагає знижувати навантаження на сервер, тим самим дозволяючи власнику сайту трохи заощадити.

З іншого боку, якщо елементів, що створюються за допомогою JavaScript, багато, то комп'ютер користувача буде сильно навантажений.

І це логічно, адже браузер їсть і так дуже багато оперативної пам'яті. А якщо ми перекладаємо все серверні обчислення на браузер користувача, то навантаження на його систему зростає.

SSR допомагає знизити навантаження на браузер користувача, це особливо важливо, якщо користувач використовує телефон або слабкий комп'ютер. [11]

Next.js - безкоштовний і відкритий JavaScript фреймворк, створений поверх React.js для створення SSR-додатків. [12]



Рисунок 2.9 – Формат Next.JS

2.3.3 Вибір системи керування базами даних

Я використовував у своєму проєкті Firebase. Сам Firebase-це хмарна технологія, у якій використовується технологія RealtimeDataBase, яка є СКБД.

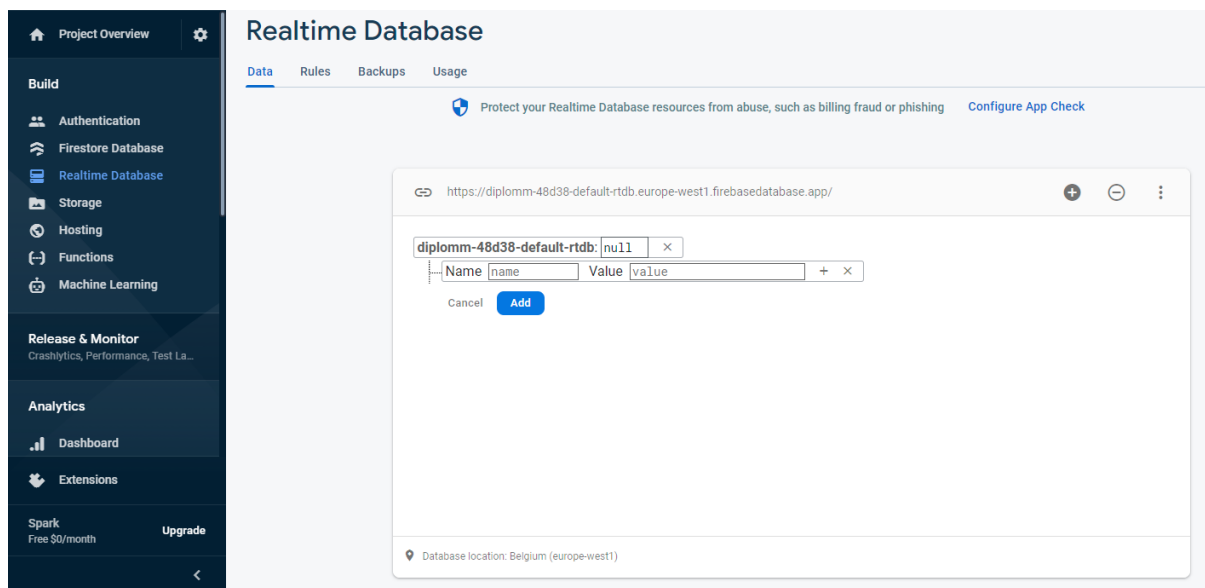


Рисунок 2.10 – Firebase DataBase

За допомоги JsonFile усі картинки та тексти зберігаються у RealTimeDatabase та відображаються на сайті, та автоматично добавляють інформацію з сторінки «Допомога» про залишену заявку.

Firebase спочатку підтримується усіма основними сучасними браузерами. Однак Json не настільки широко підтримується у всіх браузерах. Я використовував Json для створення прив'язок між браузерами на клієнті, таких як Angular 2, React та Sails.js, наприклад. Прив'язав Json до React. За допомогою Json клієнт може проаналізувати вихідні дані, щоб переконатися, що вони відповідають веб-API та специфікації JSDoc.

Останнім часом для розробки веб-додатків стало популярним таке рішення, яке називається Firebase. Давайте розглянемо, що це таке.

А конкретно, в Firebase є таке поняття як Realtime Database давайте розбиратися, що це таке.

Firebase - це комплексна платформа для розробки веб-додатків від компанії Google.

Ця платформа - набір певних сервісів, які включені в продукт Firebase. Там ви можете побачити безліч сервісів: Cloud Firestore, Cloud Functions і те, про що ми

будемо говорити в цьому відео - це Realtime Database (база даних в режимі реального часу).

Коли говорять про Firebase, розуміють цей сервіс.

Використовуючи її в своїй роботі ми отримуємо швидкість і зручність роботи з даними в реальному часі. Fire (вогонь), щось таке динамічне, постійно мінливий.

Що значить режим реального часу?

Завдяки Firebase ми можемо створювати такі системи, які будуть оновлювати свої дані в режимі реального часу на всіх пристроях, які будуть цією системою користуватися.

Firebase робили для мобільних телефонів, а саме додатків, але можливо його використовувати, як Веб-додаток

Користувачі можуть користуватися одним і тим самим додатком на різних пристроях

Такий плюс ось цієї системи.

У традиційному додатку рішення такого завдання проблемно.

Realtime database - це зберігання і синхронізація даних в реальному часі.

Це документо-орієнтована база даних.

База даних працює з Json файлами

Це універсальність роботи з різними пристроями і нам не потрібно готувати з таблиць дані в цей формат.

```

{
  "orders":
  [
    {
      "orderno": "748745375",
      "date": "Jun 30, 2008 1:54:23 AM",
      "trackingno": "TN0039291",
      "customer" :
      {
        "custid": "11045",
        "fname" : "Sue",
        "lname" : "Hatfield",
        "address" : "1409 Silver Street",
        "city" : "Ashland",
        "state" : "NE",
        "zip" : "68003"
      }
    }
  ]
}

```

Рисунок 2.11 – Приклад Json формату

Саме завдяки Json файлам у сайті вибудовані картки резидентів та інформація про найближчі притулки на гугл карті.

Набагато опрацьовує інформацію швидше, ніж реляційні СКБД.

Уже всередині бази даних дані у нас зберігаються в цьому форматі. Дані зберігаються в хмарі. Ми можемо на будь-якому пристрої перейти в нашу базу даних і щось там поміняти.

Ідея в тому, що в цьому сервісі ми платимо за те, чим користуємося. Але, головне, що там є безкоштовний тарифний план і для простих, невеликих проектів, ви можете користуватися Firebase безкоштовно.

Звичайно, у цього тарифного плану є обмеження. База даних обмежена в зберіганні 1GB даних. Але, для того, щоб розмістити 1GB даних, потрібно постаратися. За обсягом завантаження ліміт 10Gb в місяць.

І ще одне обмеження безкоштовного тарифного плану, що на 1 проект може бути тільки одна база даних.

Але, для розробки якихось простих додатків це цілком може підійти. Якщо вам буде потрібно щось більш просунуте, ви завжди можете перейти на платну версію.

Така ідея. Рішення, яке останнім часом набуває все більшої популярності.

Ну, і в наступних відео я хотів би зупинитися докладніше якомога користуватися цією системою. Як взаємодіяти з нею за допомогою мови програмування PHP.

Конструктор відповідає за фактичну реалізацію веб-API та бере дані з уже створеного облікового запису сховища.

2.3.4 Вибір мови розмітки та стилю

Мовою розмітки я обрав HTML.

HTML (Hypertext Markup Language) - це мова гіпертекстової розмітки. Ця розмітка створюється за допомогою тегів (тобто за допомогою «міток») - наборів символів, що входять в незграбні дужки. [13]

Наприклад, основний тег сторінки html пишеться наступним чином - `<html>`. Будь-яка сторінка в інтернеті складається з безлічі тегів. Звичайно, це не те, що ми звикли бачити, коли заходимо в інтернет. Кожен з цих тегів грає певну важливу роль.

Таблиці стилів CSS або каскадні стилі - це те, що використовується в сучасному дизайні веб-сайтів для застосування візуального вигляду до сторінки. Хоча HTML створює структуру сторінки і Javascript може обробляти поведінку, зовнішній вигляд веб-сайту - це домен CSS.

Що стосується цих стилів, то вони найчастіше застосовуються за допомогою зовнішніх таблиць стилів, але ви також можете застосувати стилі CSS до одного конкретного елемента, використовуючи те, що відомі як «стилі вбудованих».

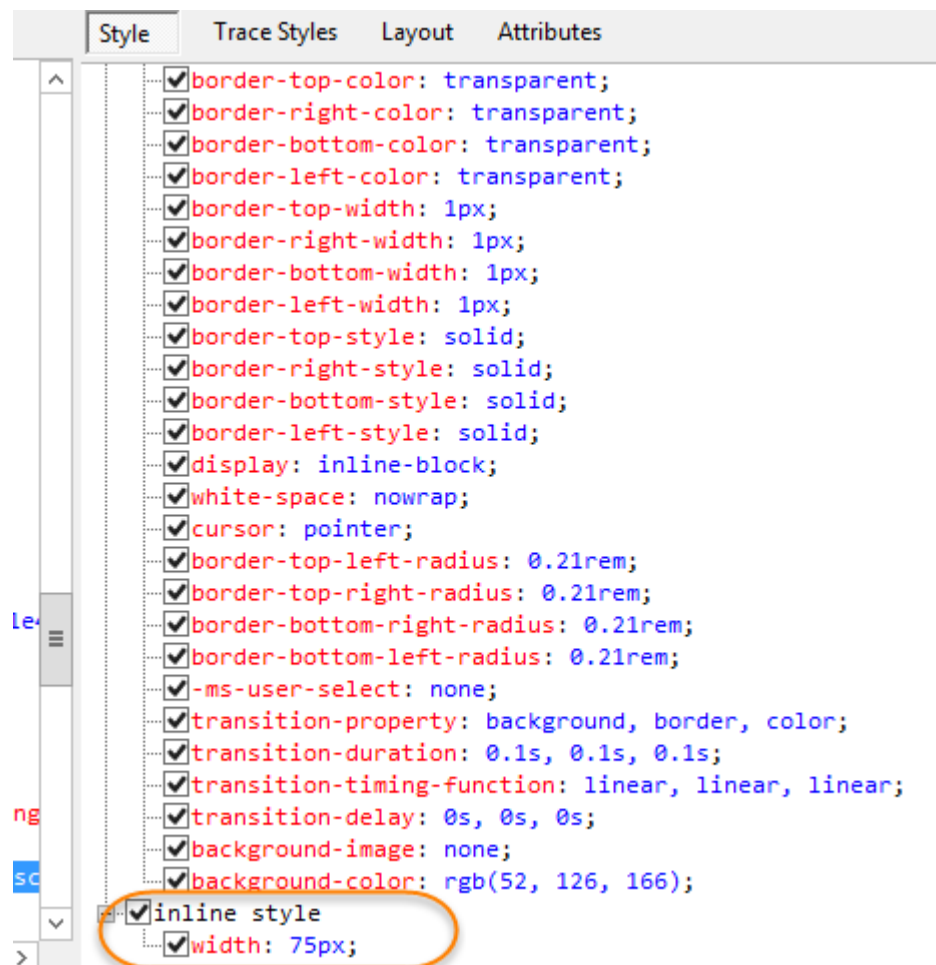


Рисунок 2.12 – Різноманітність CSS-стилів

Вбудовані стилі - це стилі CSS, які застосовуються безпосередньо в HTML-сторінці сторінки. У цього підходу є і переваги, і недоліки. Спочатку давайте розглянемо, як саме написані ці стилі.

2.3.5 Вибір редактору програмного коду

Visual Studio Code - це посібник зі створення, редагування та введення сучасних веб-програм та програми для хмарних систем. Visual Studio Code розповсюджується безкоштовно та доступний у версіях для Windows, Linux та OS X.

Редактор включає вбудований налагоджувач, інструменти для роботи з Git та інструменти рефакторингу, навігацію кодом, автозаповнення стандартних конструкцій та контекстні підказки.

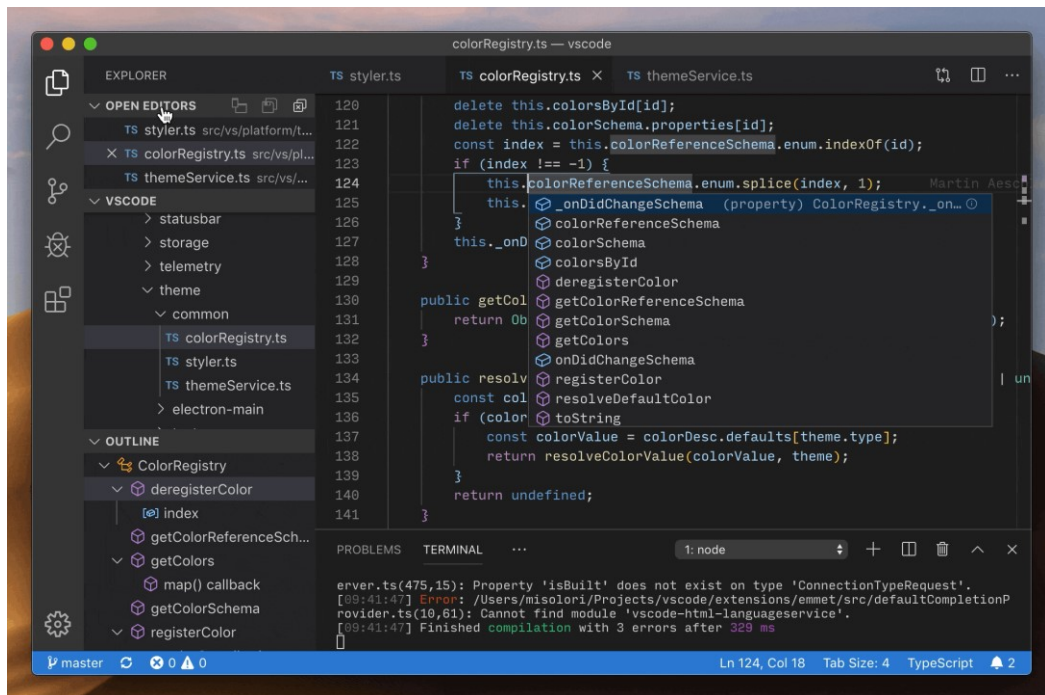


Рисунок 2.13 – Інтерфейс Visual Studio

Visual Studio Code побудований на платформі з відкритим кодом Electron, яка дозволяє створювати міжплатформові настільні програми з використанням HTML, CSS та JavaScript.

Структура Electron дозволяє користувачам створювати сучасні програми, що використовують HTML, CSS, JavaScript та SVG.

Використовуючи компоненти з відкритим кодом, користувачі можуть розширити платформу, включивши веб-технології, необхідні для чудового досвіду: включаючи підтримку локалізації, програми UWP, багату додаткову екосистему, прив'язки Electron API і навіть ігрові движки на основі браузера.

Однією з найпопулярніших функцій Visual Studio Code є CodeLens, яка надає детальну інформацію. Продукт підтримує розробку для платформ ASP.NET і Node.js, і заявляє себе, як певне рішення яке може оминати інтеграцію у середовище проекту. Мови яку можуть пітримувати ці технології: JavaScript, C++, C#, TypeScript, jade . [8]

Отже, за основу було взято JavaScript та головною бібліотекою до нього було обрано саме React. Для більш комфортного користування сайту було обрано Server

Site Rendering. Для відображення інформації на сайті, найпростішою та найпопулярнішою зв'язкою було використано JavaScript+HTML+CSS.

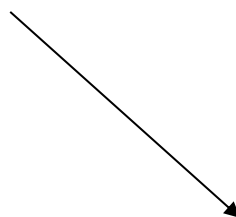
Розділ 3

Програмна реалізація

3.1 Структура і функціональне призначення модулів системи, їх взаємозв'язок

За своєю структурою програмне забезпечення для надання інформації про роботу притулку тварин складається з чотирьох блоків, кожен із яких чітко відповідає своєму призначенню, яке полягає в реалізації певної функції системи.

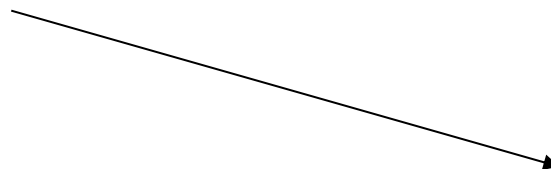
1. home-модуль, який відповідає за роботу форми відображення основної інформації про роботу притулку;



Головна Резеденти Допомога Контакти

Рисунок 3.1 – Вкладка Головна

2. contacts-модуль, який відповідає за роботу форми відображення інформації про відділення притулку;



Головна Резеденти Допомога Контакти

Рисунок 3.2 – Вкладка Контакти

3. residents-модуль, який відповідає за роботу форми відображення резидентів притулку;

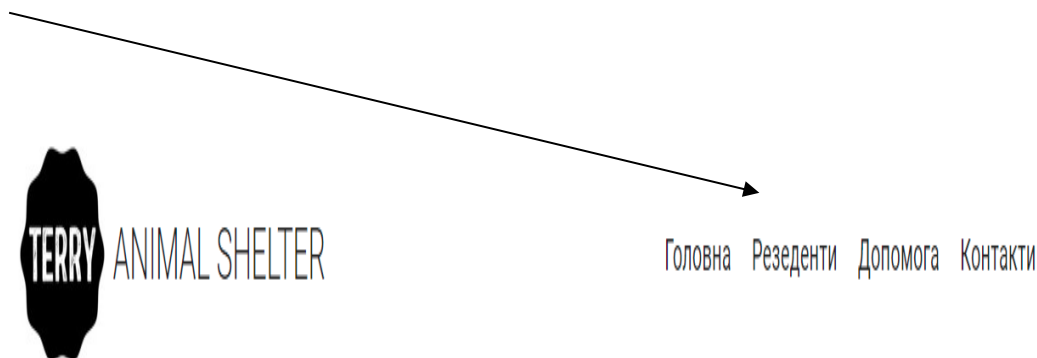


Рисунок 3.3 – Вкладка Резиденти

4. help-модуль, який відповідає за роботу форми зворотнього зв'язку.

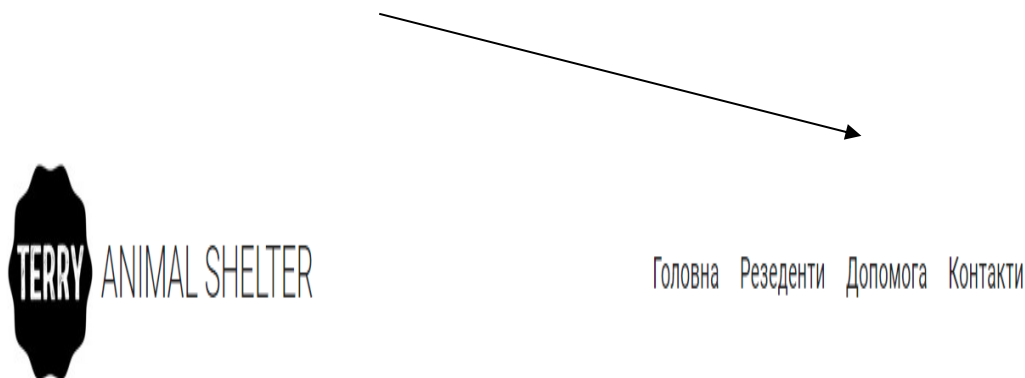


Рисунок 3.4 –Вкладка Допомога

Модуль help являє собою основний функціональний модуль, який включає процедури та функції, які відповідають за:

- відображення інформації про роботу притулку;

Модуль contacts створений для відображення місця розташування притулку у вигляді міток на карті та детальнішої інформації про обраний притулок. Цей модуль являє собою основний функціональний модуль, який включає процедури та функції, які відповідають за:

- відображення інформації про місце розташування притулку;
- відображення інформації про засоби зв'язку з притулками.

Модуль `residents` являє собою основний функціональний модуль, який включає процедури та функції, які відповідають за:

- відображення інформації про резидентів притулку;
- вибір за заданими параметрами.

Модуль `help` необхідний для відображення форми зворотнього зв'язку, валідації введених даних щоб запобігти засмічення бази даних недостовірною інформацією, запис отриманих даних за допомогою систем керування базами даних та являє собою основний функціональний модуль, який включає процедури та функції, які відповідають за:

- заповнення форми контакту з притулком;
- відправку даних.

3.2 Розробка програмних модулів

Найважчий модуль у роботі був зворотній зв'язок з користувачем, тому саме ця частина коду демонструє про створення полів для заповнення даних та передачі до `Firebase`.

```

56
57 const writeData = ({ name, email, desc, phone }) => {
58   try {
59     database
60       .ref("massages/" + database.ref().child("massages").push().key)
61       .set({
62         username: name,
63         phone: phone,
64         email: email,
65         description: desc,
66       });
67   } catch (e) {
68     console.log(e);
69   }
70 };
71
72 const validationSchema = yup.object().shape({
73   email: yup.string().trim().required(),
74   desc: yup.string().trim().required(),
75   phone: yup.string().trim().required(),
76   name: yup.string().trim().required(),
77 });
78
79 export const SectionForm = () => {

```

Рисунок 3.5 – Код який збирає інформацію

Також не менш важливим модулем було підключення гугл карти до проекту. Використовуючи карти стала можливість встановлювати точки з прописаною адресою та інформацію по ній.

```

51
52 const maps = {
53   a: {
54     address: "st.Boguslave 14",
55     city: "Kiev",
56     country: "Ukraine",
57     phone: "+380681766737",
58     email: "fasbhak@mail.com",
59     title: "Animal shalter KOTIK",
60   },
61   b: { address: "Podilska 8", city: "Khmelnitsky", country: "Ukraine", phone: "+3787219432", email: "dhwj@gmail.com", title: "Shelter2.0" },
62   c: { address: "", city: "", country: "", phone: "", email: "", title: "" },
63 };
64
65 export const SectionContacts = () => {
66   const classes = useStyles();
67   const [selected, setSelected] = React.useState("a");
68
69   return (
70     <Container style={{ position: "relative" }}>
71       <Card
72         style={{
73           position: "absolute",
74           width: 300,
75           height: "100%",
76           zIndex: 20,
77           right: 0,
78           padding: 20,
79         }}
80       >
81         <Typography variant="h4">{maps[selected].title}</Typography>
82         <Divider
83           style={{ marginTop: 10, marginBottom: 16 }}
84           orientation="horizontal"
85         />
86         <Typography variant="h5">{maps[selected].address}</Typography>
87         <Typography variant="h5">
88           {maps[selected].city} {maps[selected].country}
89         </Typography>
90         <Typography variant="h5">Phone number:</Typography>
91         <Link href={`tel: ${maps[selected].phone}`}>
92           <Typography variant="h6">{maps[selected].phone}</Typography>
93         </Link>
94         <Typography variant="h5">Email address:</Typography>

```

Рисунок 3.6 – код з google maps

Модуль пошуку тварини за критеріями був реалізований таким чином: або є вибрані усі позиції або позиція підбирається за критерієм.

```

37   cardDesc: {
38     paddingTop: "102px",
39   },
40 });
41
42 export const SectionResidents = () => {
43   const classes = useStyles();
44
45   const [breed, setBreed] = useState("dog");
46   const [size, setSize] = useState("xs");
47   const [sex, setSex] = useState("male");
48   const [isAllSelected, setIsAllSelected] = useState(true);
49
50   const filteredData = data.data.filter((item) =>
51     isAllSelected
52     ? true
53     : item.species === breed && item.sex === sex && item.size === siz
54   );
55
56   return (
57     <Container className={classes.root}>
58       <Grid container justify="between">
59         <Grid item xs={6} alignItems="center" container>
60           <Typography variant="h2">Our residents</Typography>
61         </Grid>
62         <Grid
63           item
64           xs={6}
65           alignItems="center"
66           justify="flex-end"
67           container
68           spacing={3}
69         >
70           <Grid item xs={3}>
71             <FormControlLabel
72               color="secondary"
73               control={
74                 <Switch
75                   checked={isAllSelected}
76                   onChange={() => setIsAllSelected((prev) => !prev)}
77                 />
78               }
79               label="Select all"
80             />
81           </Grid>
82           <Grid item xs={3}>

```

Рисунок 3.7 – Код вибірки сайту

Крім того є реалізований освітній матеріал, за допомогою циклу та певних елементів html і css.

Після процесу побудови модулів потрібно здійснити їх тестування задля впевненості працездатності програмного забезпечення.

```

59 export const Header = (props) => {
60   const classes = useStyles();
61   const { sections, title } = props;
62
63   return (
64     <Card className={classes.root}>
65       <Container>
66         <Grid container alignItems="center">
67           <Grid
68             item
69             container
70             alignItems="center"
71             display="flex"
72             direction="row"
73             xs={6}>
74             >
75               
76               <Typography
77                 component="h2"
78                 variant="h2"
79                 noWrap
80                 className={classes.toolbarTitle}>
81                 >
82                   Animal shelter
83               </Typography>
84             </Grid>
85             <Grid
86               item
87               xs={6}>
88               container
89               justify="flex-end"
90               display="flex"
91               spacing={3}>
92               >
93               {sections.map((section) => (
94                 <Grid item className={classes.link_container}>
95                   <Link
96                     color="inherit"
97                     noWrap
98                     key={section.title}
99                     variant="body2"
100                    href={section.url}
101                    className={classes.toolbarLink}>
102                   >
103                   <Typography

```

Рисунок 3.8 – код з логотипом сайту та його позицією

3.3 Тестування інформаційної системи

При тестуванні сайту було використано спосіб номінального користування.

Головна Резеденти Допомога Контакти

Рисунок 3.9 – Можливі сторінки сайту

3

При наведенні курсором на кнопку вона вірно виділяється для користувача та тим самим спрощує переміщення користувача по сайту.

Our residents

 Select all

 Breed
 Dogs ▾

 Sex
 Female ▾

 size
 Small ▾

Малий

mops

 ривита, стерилізована. Забирайте эту красавицу
 домой. Звоните 067 509 18 92, 099 256 73 03 Катя

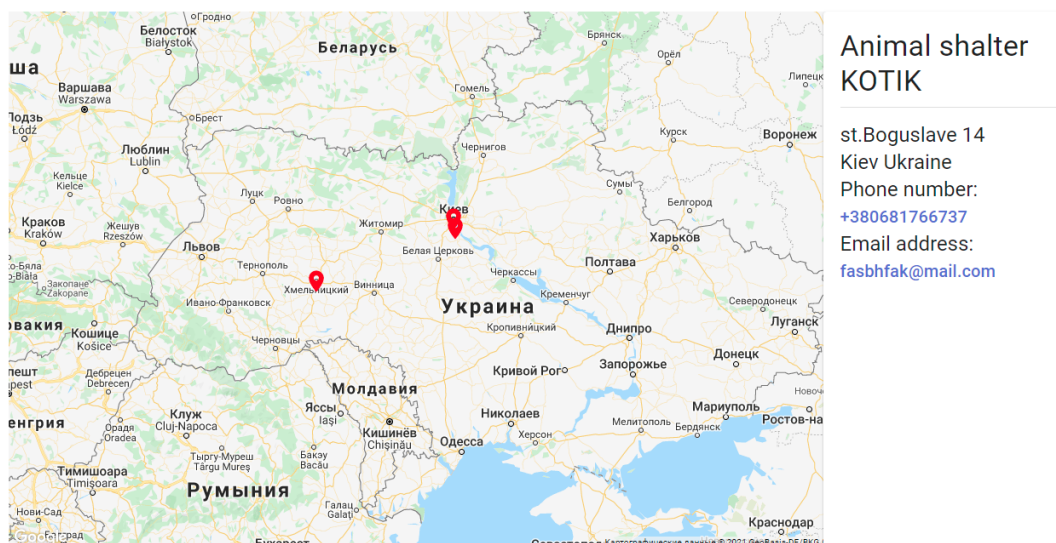
Рисунок 3.10 – Сторінка вибірки

При налаштуванні індивідуальної вибірки за критеріями сайт вірно показує тварин які відповідають характеристикам.

Далі сайт тестувався за певними критеріями :

- Сайт однаково виглядає і працює у всіх потрібних браузерях.
- Кнопки натискаються і після цього щось відбувається, слайдери крутяться
- Усі JavaScript-скрипти працюють коректно.
- Відображається правильний контент.
- Правильно написані заголовки.
- Завантажуються правильні шрифти.
- Фав іконка встановлена.
- Текст нормально відображається
- Курсор інтерактивний на інтерактивних елементах і звичайний на звичайних.
- Сторінка не роз'їжджається, якщо включити блокувальник реклами.

При виборі точки на карті відображається місцезнаходження клубу, його адреса, номер телефону та емаїл адрес



Animal shalter KOTIK

st.Boguslave 14

Kiev Ukraine

Phone number:

+380681766737

Email address:

fasbhak@mail.com

Рисунок 3.11 – Карта з відміткою геопозиції клубів

3.4 Інструкція користувача



Головна Резеденти Допомога Контакти

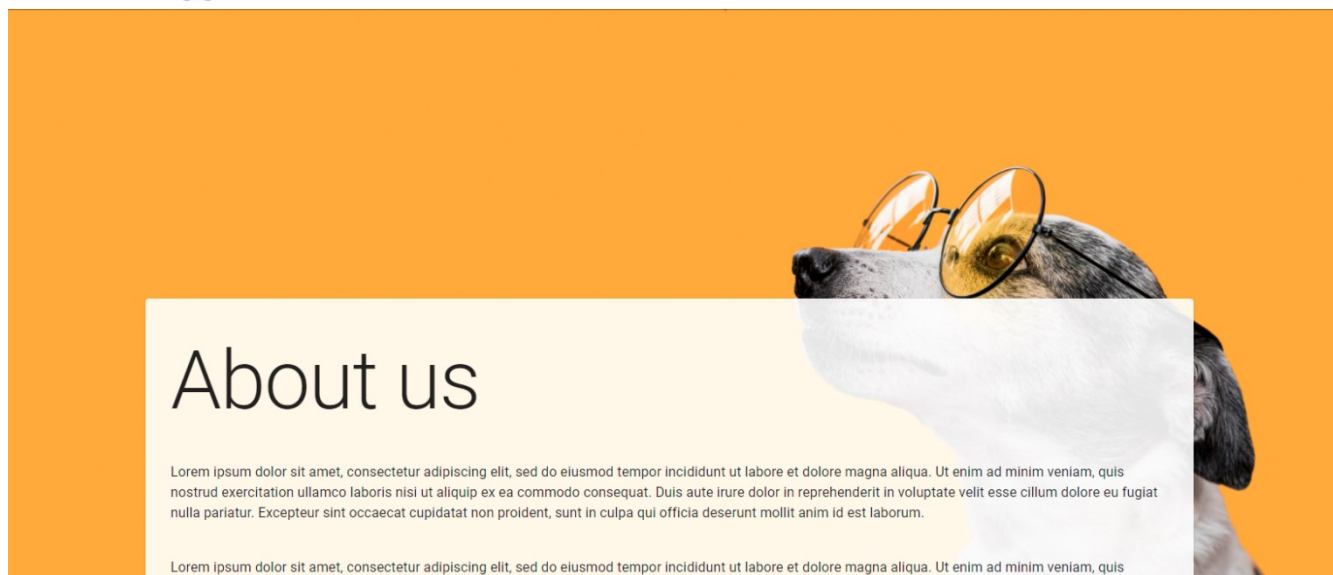


Рисунок 3.12 – Головна сторінка сайту

Першою справою відвідувач сайту бачить головну сторінку сайту з основною інформацією про клуб та головними кнопками які відповідно ведуть на інші сторінки сайту

Our residents

 Select all

 Breed
Dogs

 Sex
Male

 size
Small

Тигриг

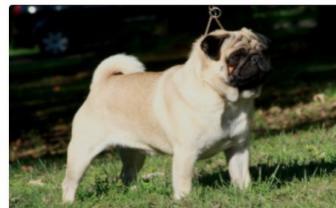
mops

Кекс жадно ловит каждую секунду внимания, обращенного в его сторону. С одинаковым удовольствием сидит на руках и играет с кисточкой.


Малий

mops

ривита, стерилизована. Забирайте эту красавицу домой. Звоните 067 509 18 92, 099 256 73 03 Катя


Принцеса

mops

ривита, стерилизована. Забирайте эту красавицу домой. Звоните 067 509 18 92, 099 256 73 03 Катя
читати далі

Рис 3.13 – Включена вибірка усі тварини

При виборі повзунка – Select all – користувач бачить повністю усіх тварин, які є у запропонованому списку.

Our residents

 Select all

 Breed
Cat

 Sex
Female

 size
Large

[ГОЛОВНА](#)
[РЕЗЕДЕНТИ](#)
[ДОПОМОГА](#)
[КОНТАКТИ](#)

Рисунок 3.14 – Вибірка статі тварини

Our residents

 Select all

 Breed
Dogs

 Sex
Male

 size
Medium


Тигриг

mops

Кекс жадно ловит каждую секунду внимания, обращенного в его сторону. С одинаковым удовольствием сидит на руках и играет с кисточкой.

Рисунок 3.15 – Вибірка розміру тварини

Отже користувач при виборі кращих йому критеріїв зможе бистріше знайти тварину яка йому підходить, тим самим набагато спростить вибір тваринки.



Залишилися запитання?

 ПІБ *

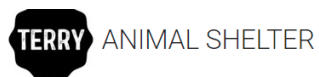
 Електронна пошта *

 Номер телефону

 Ваше повідомлення

Рисунок 3.16 – Форми де можливо залишити свої данні

При переході на сторінку Допомога користувач може задати питання або написати як він може допомогти клубу. Усі строки підписані та заповнювати потрібно відповідно.



Головна Резеденти Допомога Контакти

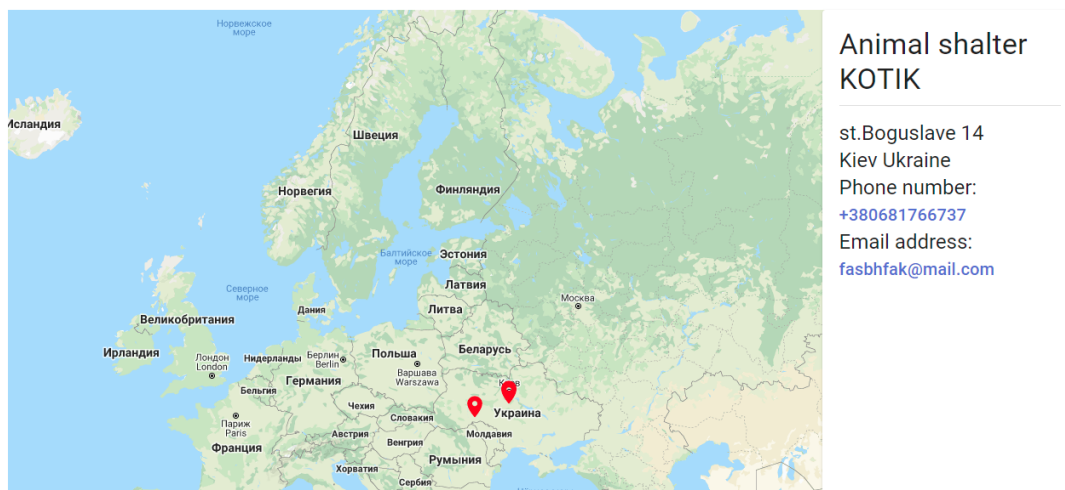


Рисунок 3.17 – Карта з позначками притулків

Нажимаючи на вже заготовлені відмітки на карті користувач може відразу дізнатися де знаходиться найблищий притулок до нього та продивитися контакти відповідного притулку.

Також, підводив сайт до домену та тестував його на різних google-інструментами.

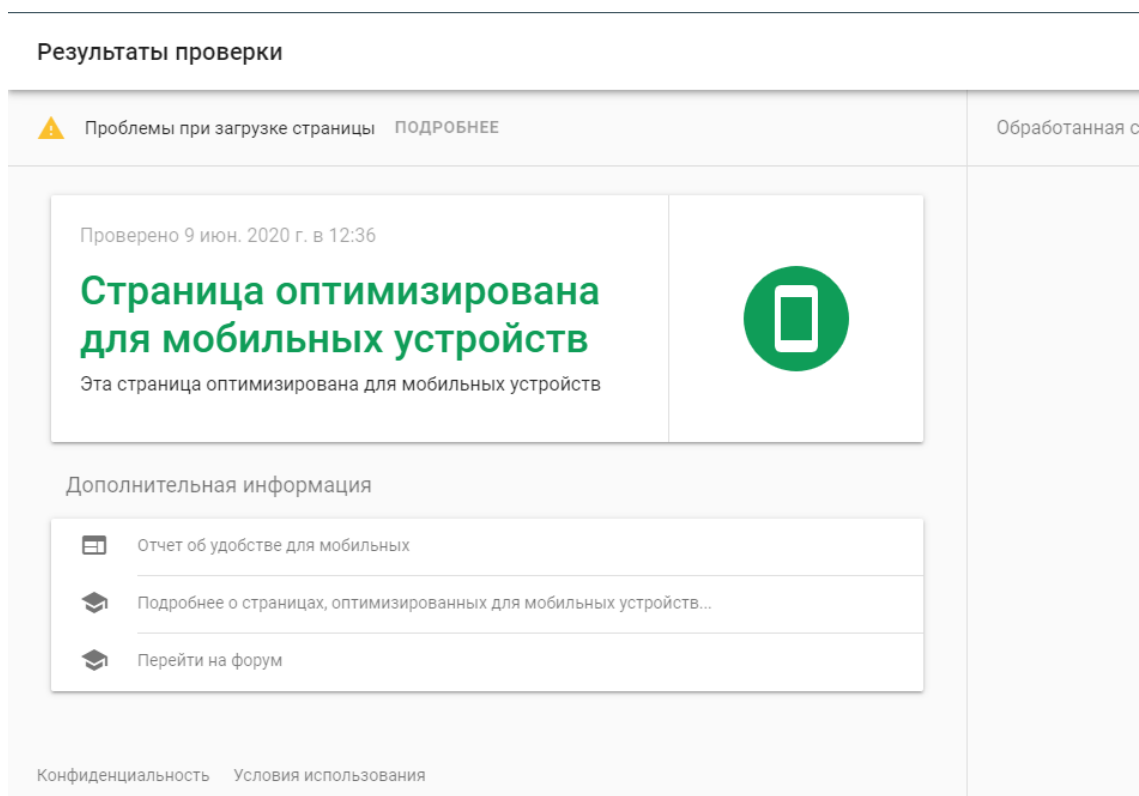


Рисунок 3.18 – перевірка на оптимізацію під телефони

Також була здійснена перевірка у сервісі Google PageSpeed для мобільних пристроїв та комп'ютерів.

Сайт є доволі непогано оптимізований для комп'ютерів. Отримав оцінку 75 з 100 можливих.

Сайт отримав певні зауваження від Google PageSpeed

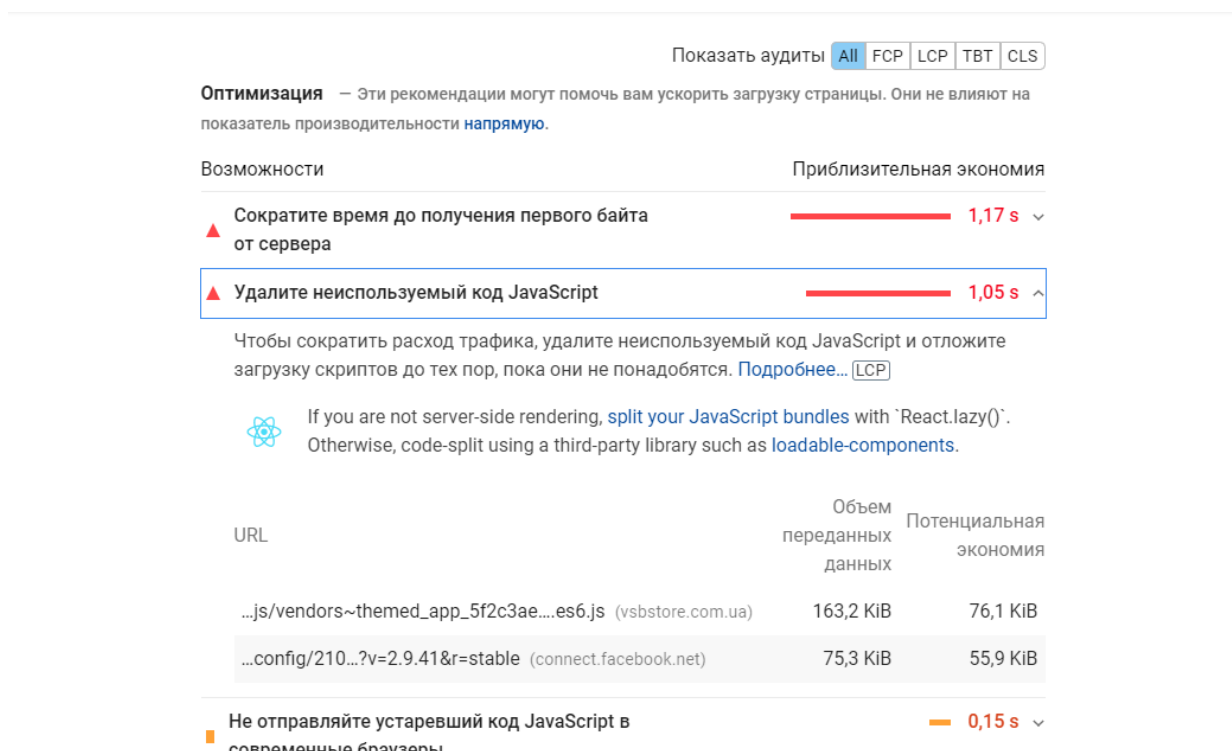


Рисунок 3.19 – Зауваження Google PageSpeed

3.5 Вимоги до апаратних та програмних засобів

1. Процесор: з тактовою частотою 1 ГГц (або швидший) і підтримкою технологій PAE, NX і SSE2.
2. Оперативна пам'ять: 1 ГБ (для 32-розрядної версії) або 2 ГБ (для 64-розрядної версії).
3. Вільний дисковий простір: 16 ГБ (для 32-розрядної версії) або 20 ГБ (для 64-розрядної версії).
4. Графічна плата: графічний пристрій Microsoft DirectX 9 із драйвером WDDM
5. Доступ до Інтернету (може стягуватися плата за послуги провайдера)

Висновки

В результаті виконання кваліфікаційної роботи бакалавра було розроблено сайт для притулку тварин на платформі JavaScript. У результаті роботи було отримано інформаційний сайт з можливістю вибірки та зворотною формою.

Так як сайт планувався інформаційний вдосконалювати його немає сенсу, але для вдосконалення сайту можливо зробити кабінет для контент – менеджера, який би зміг за допомогою особистого кабінету динамічно змінювати інформацію на сайті.

До групи функцій роботи з викликами та подіями належать наступні функції:

- Функції роботи притулку для тварин;
- Робота із картками;
- Робота з участю підрозділів у події;
- Робота з пошуком та вибіркою;
- Робота зворотнього зв'язку;
- Робота з входженням працівників у підрозділ.

Інформаційна система для безпритульних тварин виконує усі функції, які були поставлені у постановці задачі. Даний проект можуть використовувати притулки.

13. HTML — мова розмітки гіпертексту. URL:
<http://www.znannya.org/?view=html>

14. NET Framework. URL: https://uk.wikipedia.org/wiki/.NET_Framework

ДОДАТКИ

Додаток А

Програмні коди основних модулів інформаційна система забезпечення життєдіяльності притулку для тварин

Лістинг app.js

```
import React from "react";
import PropTypes from "prop-types";
import Head from "next/head";
import { ThemeProvider } from "@material-ui/core/styles";
import CssBaseline from "@material-ui/core/CssBaseline";
import theme from "../src/styles/theme";

export default function MyApp(props) {
  const { Component, pageProps } = props;

  React.useEffect(() => {
    // Remove the server-side injected CSS.
    const jssStyles = document.querySelector("#jss-server-side");
    if (jssStyles) {
      jssStyles.parentElement.removeChild(jssStyles);
    }
  }, []);

  return (
    <React.Fragment>
      <Head>
        <title>My page</title>
        <meta
          name="viewport"
          content="minimum-scale=1, initial-scale=1, width=device-width"
        />
      </Head>
      <ThemeProvider theme={theme}>
        {/* CssBaseline kickstart an elegant, consistent, and simple baseline to build
upon. */}
        <CssBaseline />
        <Component {...pageProps} />
      </ThemeProvider>
    </React.Fragment>
  );
}
```

```

        </ThemeProvider>
      </React.Fragment>
    );
  }

```

```

MyApp.propTypes = {
  Component: PropTypes.elementType.isRequired,
  pageProps: PropTypes.object.isRequired,
};

```

Лістинг document.js

```

import React from "react";
import Document, { Html, Head, Main, NextScript } from "next/document";
import { ServerStyleSheets } from "@material-ui/core/styles";
import theme from "../src/styles/theme";

export default class MyDocument extends Document {
  render() {
    return (
      <Html lang="en">
        <Head>
          {/* PWA primary color */}
          <meta name="theme-color" content={theme.palette.primary.main} />
          <link
            rel="stylesheet"
            href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"
            />
        </Head>
        <body>
          <Main />
          <NextScript />
        </body>
      </Html>
    );
  }
}

// `getInitialProps` belongs to `_document` (instead of `_app`),

```

```

// it's compatible with server-side generation (SSG).
MyDocument.getInitialProps = async (ctx) => {
  // Resolution order
  //
  // On the server:
  // 1. app.getInitialProps
  // 2. page.getInitialProps
  // 3. document.getInitialProps
  // 4. app.render
  // 5. page.render
  // 6. document.render
  //
  // On the server with error:
  // 1. document.getInitialProps
  // 2. app.render
  // 3. page.render
  // 4. document.render
  //
  // On the client
  // 1. app.getInitialProps
  // 2. page.getInitialProps
  // 3. app.render
  // 4. page.render

  // Render app and page and get the context of the page with collected side effects.
  const sheets = new ServerStyleSheets();
  const originalRenderPage = ctx.renderPage;

  ctx.renderPage = () =>
    originalRenderPage({
      enhanceApp: (App) => (props) => sheets.collect(<App {...props} />),
    });

  const initialProps = await Document.getInitialProps(ctx);

  return {
    ...initialProps,
    // Styles fragment is rendered after the app and page rendering finish.

```

```

    styles: [
      ...React.Children.toArray(initialProps.styles),
      sheets.getStyleElement(),
    ],
  };
};

```

ЛІСТИНГ index.js

```

import { SectionMain, SectionSecond } from "../src/components";
import { makeStyles } from "@material-ui/core";
import { MainLayout } from "../src/layouts";

```

```

const useStyles = makeStyles((theme) => ({
  blur: {
    filter: "blur(10px)",
  },
  root: {
    position: "relative",
    padding: "340px",
  },
  img_bckg: {
    zIndex: "-1",
    position: "absolute",
    width: "100%",
    top: 0,
  },
}));

```

```

export default function Home() {
  const classes = useStyles();

  return (
    <MainLayout>
      <div className={classes.root}>
        <SectionMain />
        
        <SectionSecond />
      </div>
    </MainLayout>
  );
}

```

);
}

Презентація

Головна мета - створення сайту для інформаційної частини просвітлення людей про притулки

Результатів: промірени 1350 000 (0,74 сек.)

Реклама - <https://www.pawz.ua/> • 067 353 0600

П'ять міфів про притулки для бездомних тварин, які пора...
Корми для собак і котів найкращі бренди: Optima, Royal Canin, Klub-4 Paws, Miu, Jovena

Корми для котів
Корми преміум і супер-преміум класу
Для усіх порід і будь-якого віку

Розпродаж місяця
Купуй корми для своїх чотириногих з величезною знижкою

Даруємо 4 кг Optima!
Замовляй великий пакет корму для котів і отримай в подарунок такий же пакет

Даруємо корм Club 4 Paws
Купуй блок паучів (24 шт.) для котів і отримай дружок в подарунок

Опція: Часи роботи

Тільки Поводження Z Тваринами
3.0 ★★★★★ (2) - [показати](#)
10 км

Притулок для бездомних тварин (Притулок для тварин)
Телефон організації: [номер]
Притулок для бездомних тварин — місце содержания бездомных, потерявших или брошенных животных, преимущественно собак и кошек. [Викликати](#)

Похожие запросы: [пошук]

Вибірка сайту - основний модуль у якому можливо підібрати тварину за певними критеріями

TERRY ANIMAL SHELTER

Головна Резиденти Допомога Контакти

Our residents Select all

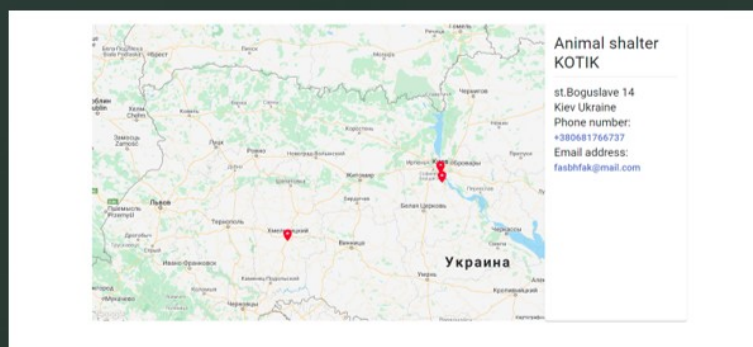
Вік: Dogo, Male, Small

Тигриг тварина
Величезний пуг, який кожен секунду викидає, облизнувши в його сторону. С надзвичайною здатністю сидіти на уздовж і зупинити і частіше.

Малий тварина
Рівель, сестричківська. Забирайте цю красуню додому. Звоніть 067 500 18 92, 099 256 73 03 Катя

Принцеса тварина
Рівель, сестричківська. Забирайте цю красуню додому. Звоніть 067 500 18 92, 099 256 73 03 Катя

Найблищий притулок. Взаємодія з google maps



Зворотній зв'язок

TERRY ANIMAL SHELTER Головна Резеднти Допомога Контакти

🐾
Залишилися запитання?

ГБС*

Ваше повідомлення

ВІДПРАВИТИ

Висновки

В результаті виконання професійної практики було розроблено сайт для притулку тварин на платформі JavaScript. У результаті роботи було отримано інформаційний сайт з можливістю вибірки та зворотною формою. Так як сайт планувався інформаційний вдосконалювати його немає сенсу, але для вдосконалення сайту можливо зробити кабінет для контент – менеджера, який би зміг за допомогою особистого кабінету динамічно змінювати інформацію на сайті.

До групи функцій роботи з викликами та подіями належать наступні функції:

- Функції роботи притулку для тварин
- інформація про діяльності притулку для тварин;
- детальний опис кожної тварини (картки);
- робота з новинами;
- розподілення по категоріям;
- пошук та вибірка;

Інформаційна система для безпритульних тварин виконує усі функції, які були поставленні у постановці задачі. Даний проект можуть використовувати притулки.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

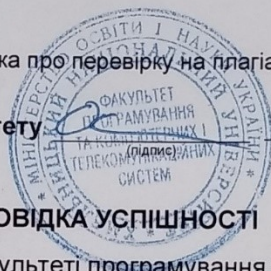
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ

Направляється студент Мариніч Т. Б. на захист дипломного проекту (роботи)
(прізвище, ініціали)
за спеціальністю 122 - Комп'ютерні науки

На тему: Інформаційна система забезпечення життєдіяльності притулку для тварин

Дипломний проект (робота), рецензія і довідка про перевірку на плагіат додаються.

Декан факультету



САВЕНКО О. С.
(прізвище та ініціали)

ДОВІДКА УСПІШНОСТІ

Мариніч Т. Б. за період навчання на факультеті програмування та комп'ютерних і телекомунікаційних систем з 2017 по 2021 роки. повністю виконав навчальний план спеціальності з такими розподілом оцінок за:

національною шкалою: відмінно 0,00 %, добре 28,12 %, задовільно 71,88 %.
шкалою ЄКТС: А 0,00 %, В 3,64 %, С 20,00 %, D 20,00 %, E 56,36 %.

Методист факультету

[Signature]
(підпис)

(прізвище та ініціали)

ВИСНОВОК КЕРІВНИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ) ТА ОБГРУНТУВАННЯ ОЦІНКИ

Студент Мариніч Т. Б. *отримав роботу по виконанню абстрактного робота гачка та виконав першого. Тема робота актуальна, розроблена модель системи та алгоритм роботи це притулок для тварин. Ідентифікація системи виконується за допомогою вимогам, КРБ виконана згідно методичних вимог. Забрали, що впливають на роботу програмною забезпечення, не введено.*

Оцінка дипломного проекту (роботи) добре

Керівник дипломного прокту (роботи)

[Signature]
(підпис)

Михалевський В. С.
(прізвище та ініціали)

" 8 " 06 2021 р.

ВИСНОВОК КАФЕДРИ ПРО ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Дипломний проект (роботу) розглянуто. Студент Мариніч Т. Б. допускається до захисту цього

Завідувач кафедри

ВНІТ
(назва)

[Signature]
(підпис, прізвище, ініціали)

Барман О. В.
(підпис, прізвище, ініціали)

" 9 " 06 2021 р.

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 23.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 13%**

ID: 93459 Название: Інформаційна система забезпечення життєдіяльності притулку для тварин Добавлено в БД: 2021-06-12 Авторы: Т.Б. Мариніч Руководители: В.Ц Міхалевський Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	35606	351	10241 (29%)	116 (33%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы
90927	Название: ЗВІТ з професійної практики Добавлено в БД: 2021-05-19 Авторы: Мариніч Т.Б. Руководители: Скрипник Т.К. Консультанты: Опоненты:	8055 (23.0%)	85 (24.0%)



Ім'я користувача:
Кафедра КН

Дата перевірки:
12.06.2021 10:05:41 EEST

Дата звіту:
12.06.2021 11:18:16 EEST

ID перевірки:
1008278057

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005671

Назва документа: **Мариніч Тимофій антиплагіат 55 Lite**

Кількість сторінок: 55 Кількість слів: 5906 Кількість символів: 46256 Розмір файлу: 11.25 MB ID файлу: 1008347789

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

12.6%
Схожість

Найбільша схожість: 3.03% з джерелом з Бібліотеки (ID файлу: 1008317721)

10.4% Джерела з Інтернету

312

Сторінка 57

4.67% Джерела з Бібліотеки

64

Сторінка 59

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

4

Підозріле форматування

15
сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна система забезпечення життєдіяльності притулку для тварин

Автор: Мариніч Т.Б., студент групи КН-17-2

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.фіз.-мат.н., доцент кафедри КНІТ Міхалевський В.Ц.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	-
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	-
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	-

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні;
- 3) до запозичень входять фрагменти програмного коду, що на мають авторства і містять поширені конструкції;
- 4) серед запозичень знаходяться загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 12,6% і адресується до першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КНІТ

В. Ц. Міхалевський

О. В. Мазурець

О. В. Бармак

Хмельницький національний університет
Кафедра комп'ютерних наук та інформаційних технологій

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента групи КН-17-2 Мариніча Тимофій Богдановича

за темою Інформаційна система забезпечення життєдіяльності притулку для тварин.

Актуальність і значення теми: сайт для перегляду тварин у притулку з можливістю обрати тварину за критеріями та з можливістю допомоги притулку.

1. Оцінка запропонованих моделей, підходів, алгоритмів, інформаційної складової та засобів розробки: сайт несе у собі актуальність перегляду тварин у притулку переглядом новин у ньому та можливістю написати про свою проблему або допомогти притулку

2. Оцінка розробленої інформаційної системи, її практична цінність та економічна доцільність: інформаційний сайт несе у собі більше благодійну та інформаційну місію для пізнання та розголошенню проблеми з безпритульними тваринами

3. Загальний висновок та оцінка: вимоги поставленої задачі виконані в повному обсязі та несе у собі, в основному, інформаційну складову, працює вірно, рекомендовано для провадження.

Робота заслуговує на оцінку « Добре »

Рецензент к.т.н., доц. Медведчук Ж.К.