МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук
Кафедра інформатики

ДИПЛОМНА РОБОТА

Рівень вищої освіти бакалавр

на тему: Розробка інформаційної системи «Допомога тваринам»

Виконала студентка 4 курсу групи К-42
Напрям підготовки 6.050101
Комп'ютерні науки
Ротаєнко Юлія Віталіївна

Керівник к.ф.-м.н., ст.викл.
Ткач Тетяна Борисівна
Рецензент к.т.н., доцент
Перелигін Борис Вікторович

Одеса 2018

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....5
ВСТУП.....6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ.....7
1.1 Дослідження предмета, цілей і особливостей інформаційних систем з надання допомоги тваринам.....7
1.2 Огляд і аналіз існуючих систем.....8
1.3 Постановка завдання.....12
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ «ДОПОМОГА ТВАРИНАМ».....13
2.1 Формування вимог до розроблюваної системі і аналіз здійсненності завдання.....13
2.2 Проектування Web-сторінок інформаційної системи.....19
2.3 Проектування БД системи.....21
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....26
3.1 Вибір і обґрунтування архітектури системи та програмних засобів для її реалізації.....26
3.1.1 Вибір архітектури системи.....27
3.1.2 Обґрунтування вибору сервера додатків.....29
3.1.3 Обґрунтування вибору системи управління базами даних.....29
3.1.4 Обґрунтування вибору мови програмування.....31
3.2 Керівництво користувача.....34
3.3 Керівництво адміністратора.....40
ВИСНОВКИ.....47
ПЕРЕЛІК ПОСИЛАНЬ.....49

ПЕРЕЛІК СКОРОЧЕНЬ

БД – База даних
ІС – Інформаційна система
ПЗ – Програмне забезпечення
СКБД – Система керування базами даних
CSS – Cascading StyleSheet (каскадні таблиці стилів)
CMS – Content management system (система керування вмістом)
HTTP – HyperText Transfer Protocol (протокол передачі гіпертексту)
HTML – HyperText Markup Language (мова розмітки гіпертексту)
PHP – PHP: HyperText Preprocessor (PHP: препроцесор гіпертексту)
SQL – Structured Query Language (структурована мова запросів)
WP – система управління WordPress

ВСТУП

На сьогоднішній день на території України перебуває велика кількість безпритульних тварин, що, у свою чергу, призводить до погіршення санітарно-епідеміологічного та екологічного стану країни, якості життя мешканців і гостей, загибелі тварин та жорстокого поводження з ними. Тому виникає нагальна необхідність принципово вирішувати проблему. Спробою врятувати життя та здоров'я тварин, а також допомогти їм знайти нових господарів є створення інформаційної системи з надання допомоги тваринам.

Інформаційна система з надання допомоги безпритульним тваринам реалізована у вигляді мережевої інформаційної системи притулку.

Основною метою створення даної інформаційної системи є можливість впровадження її в мережу Інтернет для просування благодійної діяльності щодо безпритульних тварин. Веб-система створюється для притулку, який надає можливість подбати про долю тварин. Надається інформація про сам притулок, про його послуги та забезпечується зручний спосіб зв`язку.

Пояснювальна записка до дипломної роботи містить: 49 сторінок, 30 рисунків, 14 посилань.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ

1.1 Дослідження предмета, цілей і особливостей інформаційних систем з надання допомоги тваринам

Необхідність розроблення інформаційної системи «Допомога тваринам» викликана перебуванням великої кількості безпритульних тварин на території України, що, у свою чергу, призводить до погіршення його санітарно-епідеміологічного та екологічного стану, якості життя мешканців і гостей країни, загибелі тварин та жорстокого поводження з ними.

В наш час існує низка важливих аргументів на користь розробки цієї системи. Можна виділити такі:

1) організаційно-економічні: відсутні притулки, відсутнє фінансове забезпечення питань відлову та утримання безпритульних тварин у притулку за рахунок бюджету, не передбачені видатки на проведення обліку, ідентифікації і супроводу безпритульних тварин, на проведення тематичних інформаційно-просвітницьких заходів тощо;

2) виробничо-технічні: відсутня комунальна виробничо-технічна база ветеринарного обслуговування і післяопераційного утримання безпритульних тварин, транспортування, введення сучасних гуманних методів їх знерухомлення та відлову.

Таким чином, з врахуванням ситуації, що склалась в Україні у сфері поводження з безпритульними тваринами, виникла нагальна необхідність принципово вирішувати проблему.

Дана інформаційна система розроблена для притулку «Кошкин Дом», який опікується долею безпритульних котів та собак. Притулок знаходиться у двокімнатній квартирі на проспекті Шевченка, в м.Одеса, де волонтери тимчасово розміщують вихованців до тих пір, поки не прибудують в хороші руки. Всі тварини стерилізовані і щеплені від інфекцій, а також привчені до всіх необхідних процедур.

Система створюється з метою впровадження в мережу Інтернет та поширення допомоги даного виду в нашій країні. Система нагадає людям, що робити добрі справи зовсім не складно, а радість вдячних очей – безцінна. Тим більше, що для цього не потрібно нічого особливого – важлива будь-яка допомога.

Для досягнення даної мети необхідно вирішити ряд дослідних задач:

1) аналіз існуючих інформаційних систем для допомоги тваринам;

2) дослідження основних моментів, що спонукають допомагати тваринам;

3) формування вимог на розробку інформаційної системи;

4) проектування та розробка інформаційної системи з надання допомоги тваринам.

1.2 Огляд і аналіз існуючих систем

Для визначення вимог, функцій, логіки розроблюваної системи був проведений аналіз вже створених інформаційних систем, а також інформаційного забезпечення, яке буде використовуватись для подальшого проектування структури роботи.

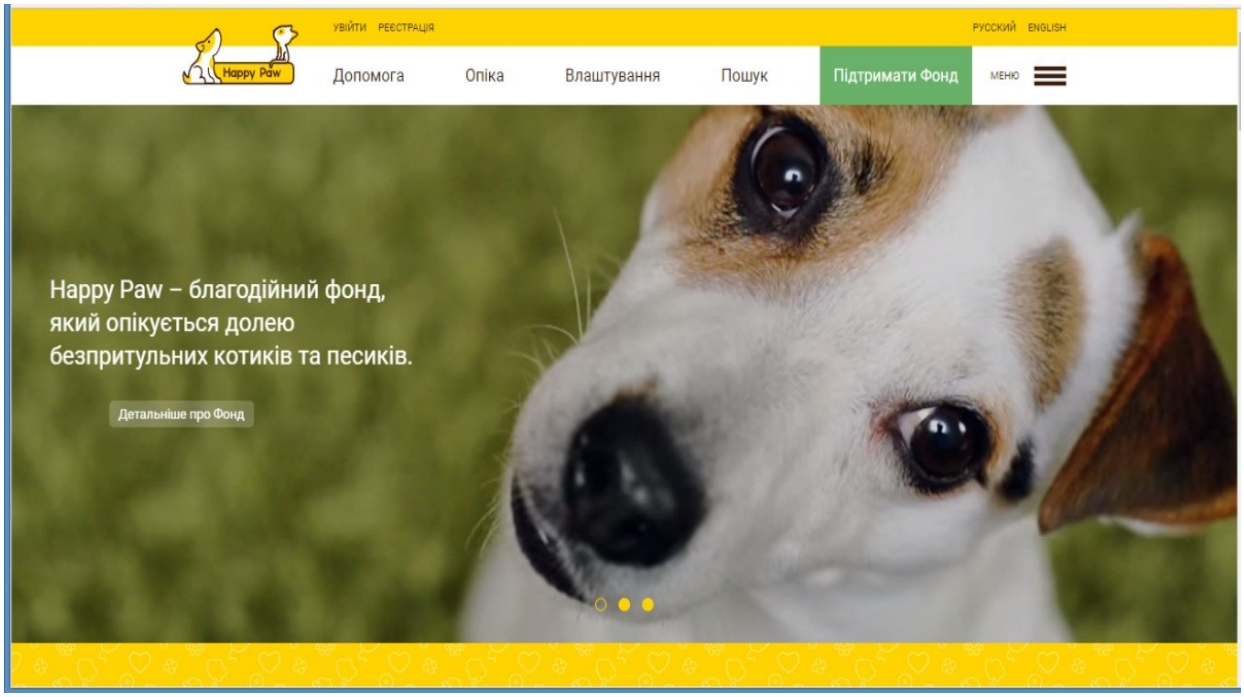
У ході аналізу дається коротка інформація про дану предметну область, опрацьовуються існуючі інформаційні системи, здійснюється аналіз технологій, які можуть бути використані при створенні Web-системи для організації, що спеціалізується на наданні допомоги безпритульним тваринам. Результати отримані шляхом пошуку аналогічних систем в мережі Інтернет та аналізу наданих ними функцій.

Огляд існуючих інформаційних систем показав, що їх не досить багато, але й ті, не всі дійсно дотримуються обіцяної допомоги. Інформаційні системи більшості фондів та притулків просто не оновлюють ніякої інформації, не демонструють свої результати. Дуже важливо мати умови, які не нанесуть шкоду тваринам. Люди не можуть бути впевнені в тому, що це, наприклад, справжній притулок, якому потрібна допомога в різних потребах для тварин, тому необхідно внести до веб-системи інформацію про все (фото,звіти,адресу), які будуть підтверджувати,що притулок функціонує на даний момент. Також, слід організувати інтерфейс, який буде зрозумілий будь-якому користувачеві, та зручну форму перегляду інформації про тварин.

Був проведений короткий аналіз та огляд веб-систем, але більшість існують, як групи у соціальних мережах, тому розглядаються веб-системи не тільки Одеси, а й інших міст України.

Веб-система київського благодійного фонду «Happy Paw» – http://happypaw.ua/ua – є інформаційним ресурсом (рис. 1.1). Інформаційна система даного фонду має такі характеристики: мови програмування JavaScript, html, веб-сервер Nginx, виконана з використанням бібліотеки JQuery.

Надає послуги допомоги, влаштування, опіки та пошуку загублених тварин, містить невелику галерею, надає можливість прийняття інформації трьома мовами. Вчасно оновлюється інформація. Явних недоліків системи не існує, лиш не надається можливості залишати та проглядати відгуки



відвідувачів [1].

Рисунок 1.1 – Головна сторінка «Happy Paw»

Наступна веб-система притулку для тварин «Сіріус». Режим доступу: http://www.dogcat.com.ua/ (рис. 1.2). Дана система створена за допомогою: мови програмування PHP, веб-серверу Nginx та Apache, бібліотек Popper.js, Jquery.

Веб-система надає можливість дізнатись про всіх тварин, які перебувають у притулку та, навіть, вибрати собі друга по певних параметрах (розмір,колір,стать,вік та стерилізація). Має великий асортимент фотографій питомців. Надається можливість стати волонтером, спонсором чи куратором (взяти під опіку). На сайті є також контакти притулку та деяка інформація про них, пошук по сайту та блог новин. У системі відсутня можливість зареєструватись та оформити заявку на опіку.

В цілому система має простий дизайн, лаконічність, доступність в користуванні, тобто зі змістом легко розібратися [2].

Наступною розглянутою системою буде БФ «Благомай». Режим доступу: http://charitymay.com/uk/we-help-uk1/help-animals_uk (рис. 1.3). Представлена система не містить галереї, інформації про тварин, які там

знаходяться, відгуки відвідувачів. Є лиш можливість переглянути, що потребує притулок та реквізити на пожертвування.

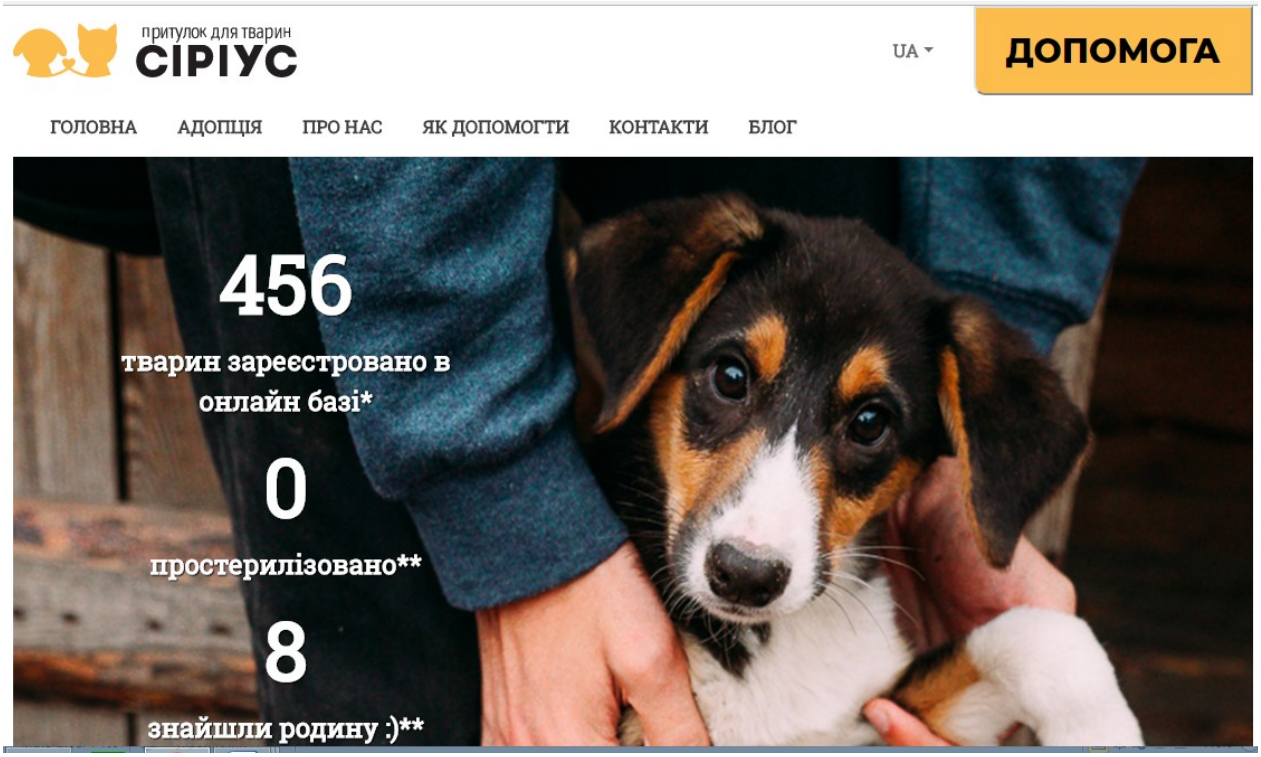


Рисунок 1.2 – Головна сторінка «Сіріус»

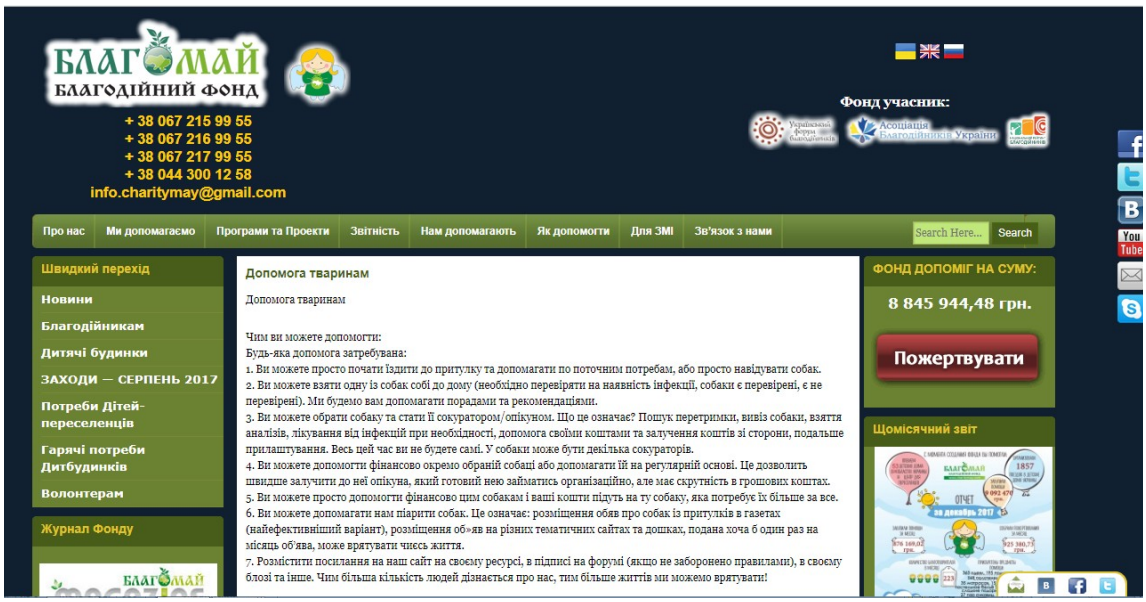


Рисунок 1.3 – Головна сторінка ІС «Благомай»

Інформаційна система «Благомай» має такі характеристики: мови програмування JavaScript, html, веб-сервер Apache, бібліотеки VK, JQuery, Facebook SDK, та збір статистики по відвідувачам за допомогою Google Analytics і Yandex.Metrika. Дана веб-система викликає недовіру. Але є повністю адаптивною і інформаційною системою [3].

У ході аналізу існуючих систем, розглянуто платформу «GladPet», режим доступу до якої: https://gladpet.org/ (рис. 1.4). Це онлайн-сервіс для влаштування і допомоги бездомним тваринам від соціальної платформи "Моє місто". На сайті вказані вихованці з багатьох одеських притулків, які вже щеплені і доглянуті. Проект пропонує швидко і безкоштовно знайти улюбленця, вибравши його з величезної бази даних. Інформаційна система має такі характеристики: мова програмування JavaScript, html та веб-сервер Nginx, бібліотека jQuery, lodash.

Дана система викликає довіру, та своїм зручним та легкодоступним інтерфейсом дозволяє привернути максимум уваги користувачів [4].

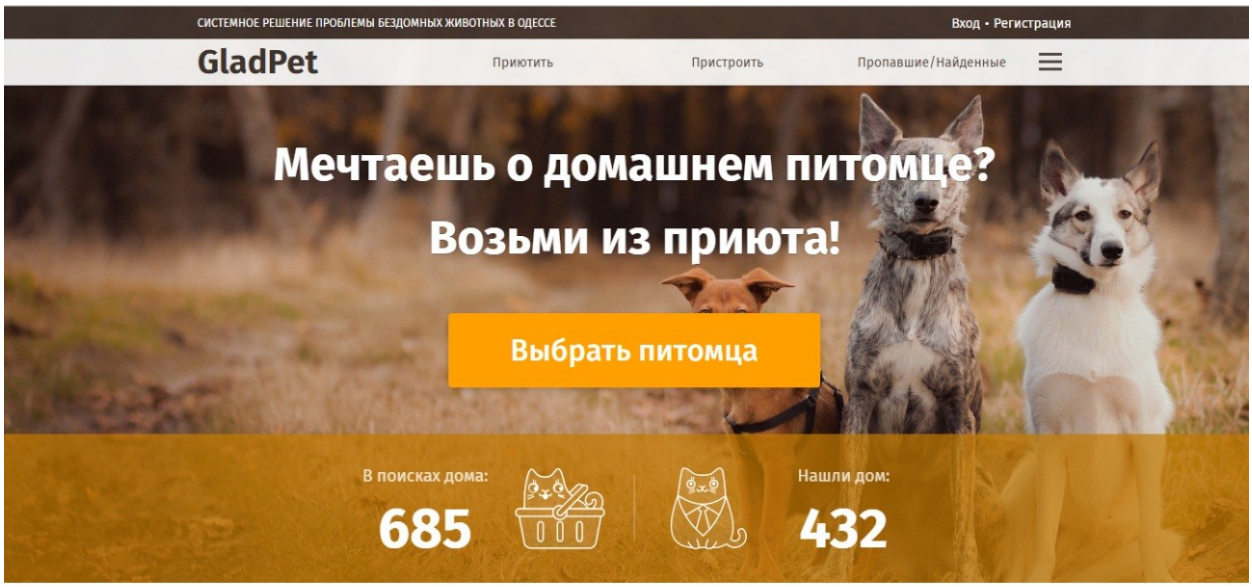


Рисунок 1.4 – Головна сторінка «GladPet»

В результаті проведеного розгляду та аналізу існуючих інформаційних систем допомоги тварин, визначено, що більшість розглянутих веб-систем забезпечують лише інформаційну присутність в мережі Інтернет, не підтримуючи належним чином спілкування з працівниками в мережі Інтернет, деякі не мають галерей та не пропонують ніяких послуг, рідко оновлюється інформація, та дуже мало подібних систем. Даний аналіз існуючих систем з обраної предметної області дозволив сформулювати

функціональні вимоги, що висуваються до розробки інформаційної системи для допомоги безпритульним тваринам.

1.3 Постановка завдання

При здійсненні розробки інформаційної системи з надання допомоги безпритульним тваринам повинно забезпечуватися:

– зрозумілий інтерфейс та зручна форма перегляду представленої інформації;
– можливість легкого доступу до потрібної інформації;
– інформація щодо розміщення та контактів притулку;
– фотогалерея тварин та інформація про притулок;
– перелік послуг, що надаються(стати волонтером, стати опікуном);
– звіт про роботу притулку у вигляді новин дня чи тижня;
– відгуки відвідувачів.

Ці всі розділи є важливими для забезпечення зручного користування веб-системою. Адже притулок для бездомних тварин повинен бути відкритими для волонтерів та всіх людей, що допомагають рятувати їхні життя.

В web-системі повинні бути визначені наступні категорії користувачів:

1) користувач-відвідувач;
2) користувач-гість;
3) користувач-адміністратор.

Необхідно створити веб-сервер, на якому буде зберігатись та вноситись інформація про надходження та перебування тварин в притулку та відгуки відвідувачів.

Функціями для користувача-відвідувача є можливість перегляду каталогу тварин, стати опікуном (відправити заявку на її опіку), додати відгук, ознайомлення з наданою інформацією, стати волонтером.

Функціями для користувача-гостя є можливість перегляду каталогу тварин та відгуків відвідувачів, стати волонтером та ознайомлення з наданою інформацією.

Функціями для користувача-адміністратора є можливість управління авторизацією в системі, додавання, видалення, редагування будь-якої інформації, наприклад, можливість додавати нові фотографії та нову інформацію або видаляти вже непотрібне, керування правами користувачів, повний контроль над працездатністю системи.

Таким чином, розроблена інформаційна система з надання допомоги безпритульним тваринам є закінченою web-системою з достатньо широким функціоналом для комфортної роботи з системою.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ «ДОПОМОГА ТВАРИНАМ»

2.1 Формування вимог до розроблюваної системі і аналіз здійсненності завдання

Для того, щоб розпочати проектування інформаційної системи потрібно визначити ряд вимог до неї. Аналіз вимог до системи розглядає функції, потреби користувача, надійність та безпеку, вимоги до зовнішніх інтерфейсів та ін. Вони оцінюються за допомогою критеріїв реалізації і можливості перевірки при тестуванні.

Перш ніж перейти до проектування системи, важливо встановити завдання досліджуваної системи і способи їх взаємодії. Необхідно визначити основні функціональні вимоги, які пропоновані до системи, тобто визначити діапазон завдань системи та програми бази даних, складу її користувачів і областей застосування.

Багато людей, дізнавшись про притулок, готові підтримати його, але їх підтримка закінчується там, де потрібно дії, а не слова. З досвіду роботи багатьох притулків можна сказати, що певна частина не коштів, а кормів, ліків і деяких інших речей буде надходити від населення. Є якісь рухи, які організовують постійно збори коштів, і серед них знаходяться ті, які не байдужі до тварин, вони будуть надавати якусь допомогу, але про нормальну величину грошових коштів або навіть матеріалів годі й казати. Тому притулку варто задуматися про просування своєї діяльності, про донесення інформації про себе, викликати резонанс в суспільстві, змусити людей бути не байдужими до такої проблеми. Потрібно звертатися в пресу, вести свої сторінки в соціальних мережах, розробити інформаційну систему в мережі Інтернет, залучати добровольців. Якщо про притулок ніхто не знає, то він так і залишиться місцем, куди його власниця приносить підібраних бездомних тварин, які і допомоги особливої отримати не можуть.

Популярність притулку може виявитися і не дуже корисною, тому що збільшиться потік людей, що несуть всіх тварин без розбору, а от бажаючих забрати тварин не надто збільшиться. Однак, рано чи пізно, ставши

популярним, притулок, про який говорять і який приваблює чимало співчуваючих, зможе приймати досить велику кількість вихованців, виходжувати їх і тут же передавати в добрі руки.

Проектування даної системи спричинено тим, що люди не беруть тварин з притулків щороку, а значить, у них немає притулку, в який вони заходять регулярно. Це не улюблений магазин продуктів. Будь-який притулок буде для них новим і невідомим місцем. У такій ситуації людині буде зручніше подивитися всіх тварин в одному місці, а не збирати інформацію по крихтах з груп «ВКонтакте» та «Facebook». У цьому полягає цінність сайту і причина ним користуватися.

Отже, існує чотири базові принципи, які необхідно враховувати при створенні сайту:

1) Мета. Хороший дизайн завжди починається з формулювання завдання. Перш ніж робити хоч щось, потрібно запитати себе про те, для чого взагалі потрібна кожна веб-сторінка. Якщо мети немає, то створювати її не варто.

2) Естетика зовнішнього вигляду. Сайт повинен мати гарний та сучасний вигляд.

3) Релевантний і оригінальний контент. Вміст сайту має відповідати певним завданням, контент повинен бути унікальним. Плагіат – це погано і карається пошуковими системами.

4) Зрозуміла навігація. Те, наскільки просто і зрозуміло можна пересуватися по сайту, дуже важливо. Будь-яка його сторінка повинна бути доступна за три кліка з будь-якої іншої сторінки. Навігація повинна бути простою і зрозумілою – це позитивно позначається на конверсії і навіть на ранжируванні в пошукових системах.

Сайт притулку – це його візитна картка. Це перша і часто єдина річ, яку бачать гості та відвідувачі. Якщо сайт притулку виглядає добре і сучасно, то його імідж в очах всіх людей буде хорошим, про вас будуть думати, як про професіоналів у своїй галузі.

Крім того, пророблений і якісний сайт вище ранжирується пошуковими системами, посиланнями на нього користувачі мережі діляться частіше.

Для того, щоб збільшити кількість відвідувачів та ймовірності прилаштувати більшість тваринок, потрібно застосовувати основні прийоми лендінгу.

Метою лендінгу в даній інформаційній системі буде: взяти під опіку тварину або (та) стати волонтером.

Зовнішній вигляд сторінки акуратний та чистий, тому не виникає ніяких труднощів при її перегляді. Оформлення тексту класичне: друк темними буквами на світлому фоні. Використовується розбивка сторінки на секції за
допомогою елементів дизайну, щоб кожну секцію можна було відрізнити від попередньої.

Зміст розташованих елементів, наприклад «Чому варто брати тварину саме з притулку?», переконує користувача зробити це, так як причини дійсно схиляють задуматись над цим вибором.

Блок з «Останніми новинами тижня» підтверджує існування даного притулку, і тим самим, веде звіт перед своїми відвідувачами. Й одночасно спонукає до того, щоб хоч якимось чином допомогти тваринам, так як новини несуть не тільки позитивний характер, а й навпаки, реальні проблеми (хвороби, недостатня кількість їжі, борги у ветеринарних клініках).

Для того, щоб остаточно впевнитися в необхідності існування даного закладу, на сторінці відображено блок з результатами його діяльності: скільки тварин уже знайшли свій дім, та скільки ще знаходяться в пошуках родини.

Після основної інформації розміщено посилання на трендові соціальні мережі: Facebook та ВКонтакте. Відвідувачі будуть ділитись даною сторінкою. Крім того, рекомендації і поради користувачів соціальних мереж збільшать довіру до даного притулку. Створено помітні посилання для того щоб, люди змогли легко їх побачити і поділитися контентом. Як результат, можна очікувати, що кількість відвідувачів збільшиться.

Отже, загальна мета системи – перетворення гостів сайту в потенційних відвідувачів та заклик до дій, які необхідні притулку «Кошкин Дом».

В результаті розробки інформаційної системи «Допомога тваринам», що надає допомогу безпритульним тваринам, створена система з розподіленим доступом і інтерфейсом для різних категорій користувачів: користувач-адміністратор, користувач-гість, користувач-відвідувач.

Набір функцій для цих категорій може бути повністю різний, а може і перетинатися.

Сформулюємо функціональні вимоги до інформаційної системи.

Розмежування прав доступу в межах інформаційної системи зазвичай здійснюється шляхом дозволу або заборони на перегляд певної інформації для певної групи користувачів. В інформаційній системі з надання допомоги

тваринам існують такі групи користувачів як: системний адміністратор, відвідувач та гість (рис. 2.1).

Рисунок 2.1 – Користувачі системи з надання допомоги безпритульним тваринам

У кожного з користувачів системи є свої права, які розподіляє системний адміністратор. Залежно від прав, користувачі мають або не мають можливості перегляду тієї чи іншої інформації в системі, або можливість додавання, видалення та редагування інформації. Системний адміністратор розподіляє права логінами і паролями, які потім заносяться в базу даних системи (рис. 2.2).

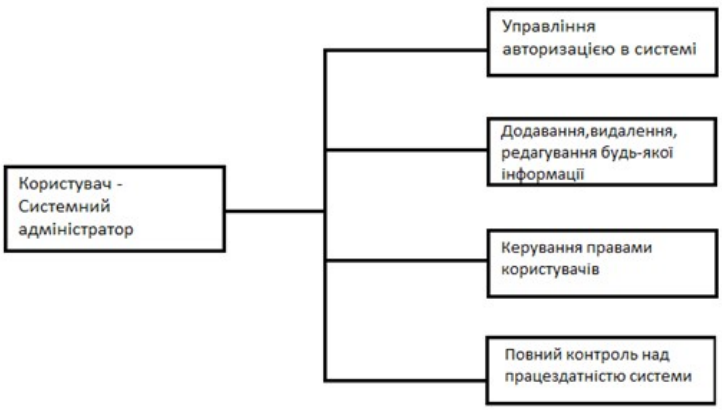


Рисунок 2.2 – Функціональні можливості системного адміністратора

Користувач-гість має можливість стати волонтарем, має змогу перегляду каталогу тварин та відгуків відвідувачів, ознайомлення з наданою інформацією (рис. 2.3).

Рисунок 2.3 – Функціональні можливості користувача-гостя

Користувач Відвідувач має можливість додавання відгуку, перегляду каталогу тварин, ознайомлення з наданою інформацією, можливість стати опікуном та (або) волонтером (рис. 2.4).

Рисунок 2.4 – Функціональні можливості користувача-відвідувача

Веб-система потребує реєстрації користувачів, так як не кожний бажаючий має право залишити відгук та відправити заявку на опіку, яку системний адміністратор обробляє та передає інформацію власниці притулку.

На комп'ютері користувача для коректної роботи веб-системи повинні бути присутні такі програмні продукти:

– веб-оглядач (Mozilla Firefox, Google Chrome, Opera тощо);

– будь-який з браузерів повинен бути оновлений до останніх версій, так як у веб-системі з надання допомоги безпритульним тваринам використовуються сучасні технології створення сторінок.

Коли користувач-гість заходить на головну сторінку інформаційної системи з надання допомоги безпритульним тваринам, то отримує доступ до інформації: « Головна», «Наш притулок», «Тваринки притулку», «Як допомогти», «Контакти», «Відгуки», «Взяти під опіку» та «Стати волонтером».

Головним входом до системи є головна сторінка. Саме від неї залежатиме перше враження, що отримується відвідувачами. Текст на головній сторінці повинен знайомити відвідувача з даним притулком. Сторінки з тваринами повинні відкриватися трохи згодом.

На «Головній» сторінці описуються відомості про тематику веб-сайту та матеріали, які можна побачити на подальших сторінках (дозволяє переглянути зміст веб-сайту).

«Наш притулок» містить в собі інформацію про заснування притулку, про його власницю та певну інформацію про перебування тваринок у притулку.

«Тваринки притулку» складається з двох підрозділів: «Коти» та «Собаки», які знаходяться на даний момент в притулку, і готові до взяття під опіку. Також міститься інформація про кожного з них (приблизний вік, порода, стать, наявність щеплень).

«Як допомогти» відображає інформацію про волонтерський рух, список речей, які потрібні тваринам у побуті (корм, засоби гігієни,ліки та ін.), номер карточки банку власниці, на випадок фінансової допомоги, а також для зручності надаються посилання на веб-банкінг найбільших банків України.

«Контакти» публікує інформацію для зв`язку: адресу та телефони притулку «Кошкин Дом», адреси в соціальних мережах та інформацію про ветеринарну клініку, в якій перебувають тварини з притулку.

«Відгуки» містить відгуки відвідувачів та форму для їх написання.

«Взяти під опіку» – це кнопка, розміщена на частині для завантаження контект-сторінки, яка дозволяє вибрати собі тварину із каталогу та залишити заявку або відразу зв`язатись з власницею притулку по телефоні, щоб забрати тварину до себе.

«Стати волонтером» дозволяє перейти в розділ «Як допомогти».

2.2 Проектування Web-сторінок інформаційної системи

За допомогою проектування системи повинен бути розроблений остаточний ескіз Web-сторінок системи.

Структура шаблону системи складається з елементів, які повинні бути присутніми на всіх сторінках Інтернет-ресурсу.

Навігаційна структура сайту виконує важливі завдання: логічно об'єднує різні інформаційні блоки, відображає поточне місце розташування відвідувача на сайті, надає елементи управління для переходів по сторінках сайту тощо. Як правило, виділяють основну і додаткову навігацію в структурі сайту.

Основна навігація на сайті представлена у вигляді горизонтальних чи вертикальних меню, що включають посилання на розділи сайту, а також окремі сторінки. Основна навігація доступна з будь-якої сторінки сайту, її вигляд не змінюється при переходах між сторінками.

Додаткова навігація забезпечує на сайті додаткову зручність користування. Така навігація може включати так звані «хлібні крихти», блоки тематично «прив'язаних» до поточної сторінки матеріалів тощо. До додаткової навігації також можна віднести внутрішні контекстні посилання, що проставляються на сторінках сайту.

Навігаційна структура сайту вимагає ретельного обмірковування з найперших стадій розробки, адже при спробах внести зміни у вже готову структуру сайту практично неминуча плутанина, поява «битих» посилань.

Ось кілька основних правил, що стосуються розробки навігаційної структури сайту:

1) з будь-якої сторінки сайту повинне бути доступним посилання на головну (домашню) сторінку. Найчастіше подібне посилання називається «Головна» або «Home», а ще практично стандартом у веб-дизайні стало створення активного посилання на головну сторінку на логотипі сайту (посилання-картинка);

2) всі посилання на сайті повинні мати інтуїтивно зрозумілий відвідувачам вид. Так, посилання в контенті сторінок сайту традиційно

виділяються підкресленням і відмінним від решти тексту кольором, посилання меню мають аналогічний вигляд або ж являють собою кнопки. Якщо посилання візуально не можна виділити на тлі інших елементів сторінок сайту або ж якщо посилання мають незвичний для більшості відвідувачів вигляд, то це робить навігацію по сайту вкрай незручною;

3) структура навігації по сайту повинна бути такою, щоб відвідувачам у будь-який момент часу було ясно, на якій сторінці вони знаходяться, як з цієї сторінки можна потрапити на інші сторінки, як повернутися до попередньої сторінки. При цьому якщо для виконання переходу від поточної сторінки сайту до якої-небудь ще потрібно 4, а то і 5-6 кліків мишею, то це говорить про низький рівень юзабіліті розробленої навігаційної структури.

Розроблено шаблон сторінок веб-системи з надання допомоги безпритульним тваринам. Всі сторінки проектованої системи виконані в єдиному стилі (рис. 2.5).

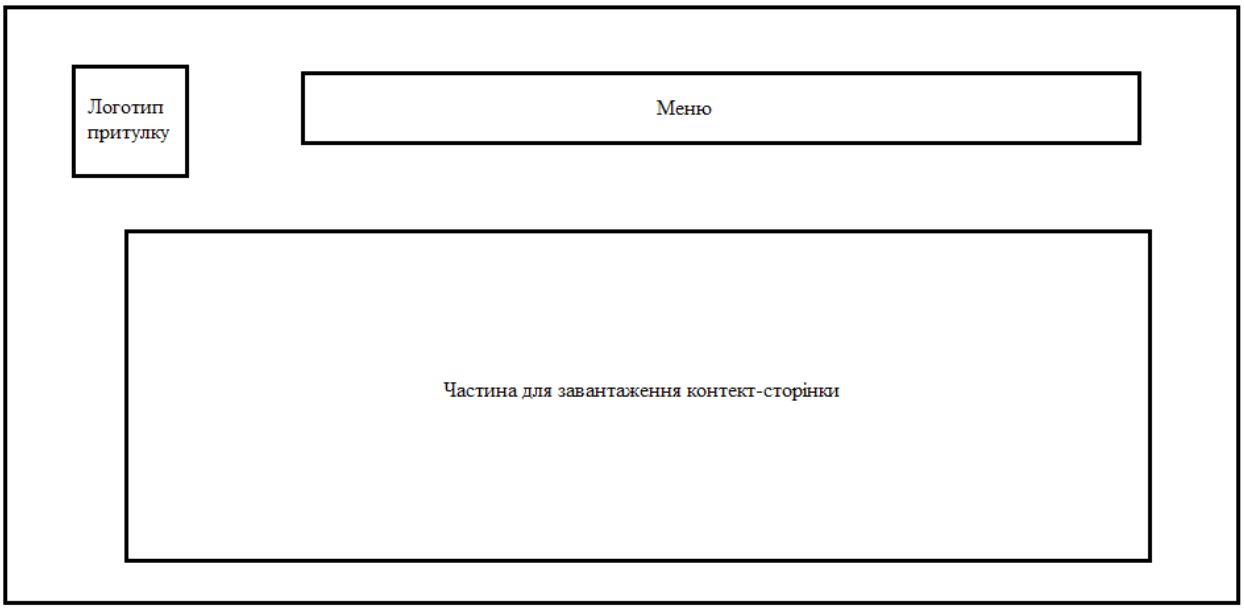


Рисунок 2.5 – Шаблон сторінки інформаційної системи з надання допомоги безпритульним тваринам

Шаблони зручні тим, що більшість сторінок верстають за подобою однієї сторінки майже автоматично. Структура шаблону складається з елементів, які повинні бути присутніми на всіх сторінках системи.

Тема реєстрації та авторизації є дуже популярною та практично необхідною для кожного сайту. Приблизно так виглядає цей процес (рис. 2.6).

Форма авторизації на сайті створюється для того, щоб сайт, або окрема сторінка сайту була доступна, і передивлялась тільки окремим колом відвідувачів, а не кожним, будь-яким відвідувачем.

В даній інформаційній системі вона створюється для того, щоб лиш відвідувачі, а не просто гості, могли залишати відгуки та відправити заявку на опіку тварини.

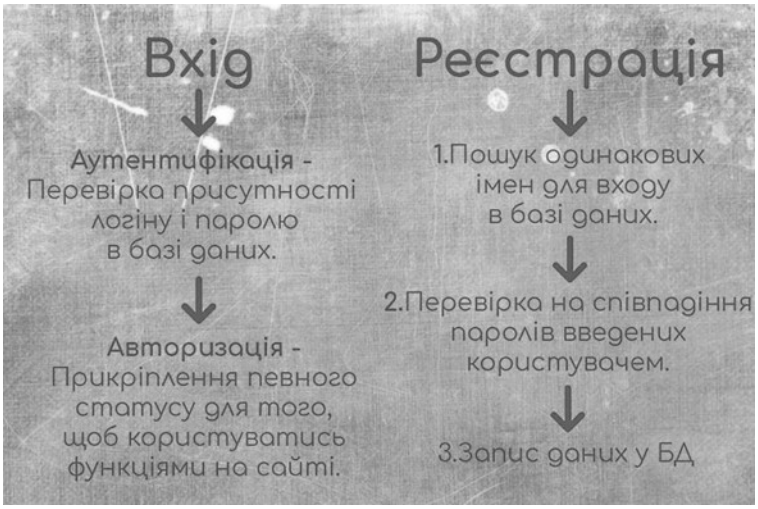


Рисунок 2.6 – Процес реєстрації та авторизації інформаційної системи з надання допомоги безпритульним тваринам

А щоб допомогти притулку матеріально, тобто стати волонтером, використовуються переходи на платіжні системи банків.

Сьогодні для грошових переказів в Інтернеті найзручніший, найдоступніший і найперспективніший засіб – це платіжні системи. Розібратися, що таке платіжні системи та як підключити систему оплати на сайт, не дуже важко. У використанні платіжних систем немає нічого складного. Реєстрація у електронних платіжних системах є абсолютно безкоштовною, а служба технічної підтримки допоможе отримати відповіді на будь-які питання щодо використання.

2.3 Проектування БД системи

Створення бази даних слід розпочинати з проектування. В результаті має бути визначена структура бази, а саме склад таблиць, їхня структура та логічні зв'язки. Структура реляційної таблиці відображає склад стовпців, їхню послідовність, тип даних кожного стовпця та їх розмір, а також ключ таблиці. Процес проектування здійснюється на вибір двома підходами.

Перший підход починають з визначення основних задач. Для того, щоб знайти їх розв`язок потрібно створити базу, та визначити потреби цих задач у даних [5-6].

Під час другого підходу потрібно визначити предметну область (сферу), здійснити аналіз її даних і встановити типові об'єкти даної сфери. Найбільш раціональним підходом проектування бази даних є поєднання обох підходів.

Розпочинаємо розробку структури бази, тобто структури її основних таблиць:

1) Роботу починаємо з визначення генерального переліку полів.

Для бази даних системи виділимо наступні сутності: Роль відвідувача, Відвідувач, Відгук, Тварина притулку та Опіка.

Сутність Роль відвідувача містить наступні атрибути: номер ролі (первинний ключ), назва ролі.

Сутність Відвідувач: номер відвідувача (первинний ключ), номер ролі, ім’я, прізвище, номер телефону, електронну адресу.

Сутність Відгук: номер відгуку (первинний ключ), номер відвідувача, номер ролі, ім’я відвідувача та відгук відвідувача.

Сутність Тварина: унікальний ідентифікатор тварини, ім’я тварини, стать тварини, порода, вік, медичні процедури (наявність щеплень).

Сутність Опіка: номер опіки,номер відвідувача, номер ролі та номер тварини.

2) Відповідно до типу даних, що розміщуються в кожному полі, визначаємо тип кожного поля.

3) Розподіляємо поля генерального списку по базових таблицях.

4) Для кожної таблиці визначаємо ключове поле. Ключовим вибираємо поле, дані в якому повторюватись не можуть. Якщо ж у таблиці взагалі немає полів, які можна було б використовувати як ключові, завжди можна ввести додаткове поле типу лічильник , воно за визначенням не може містити дані, що повторюються.

Результат проектування таблиць бази даних наведено нижче у вигляді таблиць (табл. 2.1 – 2.6).

Таблиця 2.1 – Список всіх сутностей бази даних інформаційної системи

rol_Visitor	Роль відвідувача
visitor	Відвідувач
response	Відгук

animal	Тварина притулку
caring	Опіка

Таблиця 2.2 – Роль відвідувача (rol_Visitor)

№	Поле	Атрибут	Тип
1	id_rol	Ідентифікатор ролі	int(11)
2	nameOf_rol	Назва ролі	varchar(25)

Таблиця 2.3 – Відвідувач (visitor)

№	Поле	Атрибут	Тип
1	id_visitor	Унікальний ідентифікатор відвідувача	int(11)
2	id_rol	Ідентифікатор ролі	int(11)
3	name_visitor	Ім’я відвідувача	varchar(25)
4	surname_visitor	Прізвище відвідувача	varchar(25)
5	phone	Номер телефону відвідувача	varchar(13)
6	e-mail	Електронна адреса відвідувача	varchar(25)

Таблиця 2.4 – Відгук (review)

№	Поле	Атрибут	Тип
1	id_review	Унікальний ідентифікатор відгуку	int(11)
2	id_ visitor	Ідентифікатор відвідувача	int(11)
3	id_rol	Ідентифікатор ролі	int(11)
4	name_visitor	Ім’я відвідувача	varchar(25)
5	review	Відгук відвідувача	varchar(250)

Таблиця 2.5 – Тварина (animal)

№	Поле	Атрибут	Тип
1	id_anim	Унікальний ідентифікатор тварини	int(11)
2	name_anim	Ім’я тварини	varchar(25)
3	sex_anim	Стать тварини	varchar(45)
4	breed	Порода	varchar(50)
5	age	Вік	double(11)
6	med_Proc	Медичні процедури (наявність	varchar(250

		щеплень)	)

Таблиця 2.6 – Опіка (caring)

№	Поле	Атрибут	Тип
1	id_ caring	Унікальний ідентифікатор опіки	int(11)
2	id_ visitor	Ідентифікатор відвідувача	int(11)
3	id_rol	Ідентифікатор ролі	int(11)
4	id_anim	Унікальний ідентифікатор тварини	int(11)

5) На наступному етапі визначаємо зв'язки між таблицями (схему даних). Зв'язки між таблицями організовуються на основі спільного поля, причому в одній із таблиць воно обов'язково має бути ключовим. Тобто на стороні «один» має бути ключове поле, яке не повторюється, значення на стороні «багато» можуть повторюватися.

Уявімо базу даних веб-системи з надання допомоги тваринам у вигляді моделі «Сутність–Зв'язок» (рис. 2.7).

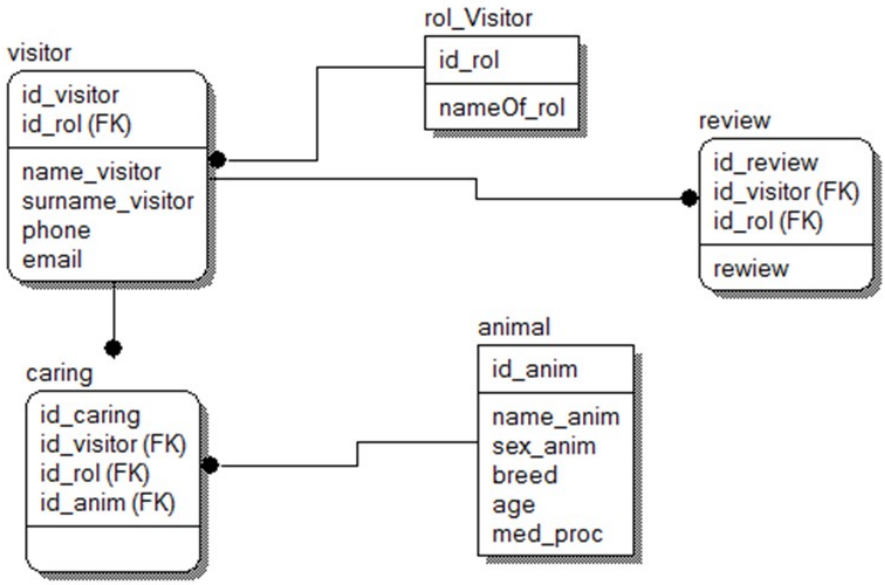


Рисунок 2.7 – Діаграма «Сутність–Зв'язок» БД web-системи з надання допомоги безпритульним тваринам

Між таблицями Роль відвідувача та Відвідувач існує зв’язок типу «один до багатьох»: на одну роль багато претендентів.

Між таблицями Відвідувач і Відгук – зв’язок типу «один до багатьох»: у одного відвідувача може бути декілька відгуків.

Між таблицями Тварина і Опіка – зв’язок типу «один до багатьох».

Між таблицями Відвідувач і Опіка – зв’язок типу «один до багатьох»: один відвідувач може взяти під опіку багато тварин.

Отримані в результаті концептуального проектування об’єкти були приведені до третьої нормальної форми. В більшості випадків третьої нормальної форми достатньо, щоб гарантувати правильність проекту бази даних. Сутність знаходиться в третій нормальній формі, якщо вона вже знаходиться в другій нормальній формі і жоден неідентифікуючий атрибут не залежить від яких-небудь інших неідентифікуючих атрибутів. Атрибути, залежні від інших неідентифікуючих атрибутів, нормалізуються шляхом переміщення залежного атрибута і атрибута, від якого він залежить, в нову сутність. Сутність знаходиться в другій нормальній формі, якщо вона вже знаходиться в першій нормальній формі, і кожний неідентифікуючий атрибут залежить від всього унікального ідентифікатора сутності. Якщо якийсь атрибут не залежить повністю від унікального ідентифікатора сутності, значить, він внесений помилково і повинен бути видалений. Потрібно нормалізувати такий атрибут або знайшовши сутність, до якого він відноситься, або створивши нову сутність, в яку він повинен бути розміщений. Перша нормальна форма забороняє повторювання стовпців, забороняє множинні стовпців і вимагає визначити первинний ключ для таблиці.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір і обґрунтування архітектури системи та програмних засобів для її реалізації

В наш час збільшуються вимоги до проектування програмного забезпечення інформаційних систем, головною складовою якого є створення архітектури програмного забезпечення. Правильно підібрана архітектура збільшує гнучкість системи, полегшує її модифікацію та супровід при неминучій зміні вимог в процесі її функціонування [7-8].

Функції архітектури ПЗ:

- визначає структуру системи;
- визначає поведінку системи;
- концентрується на значущих елементах;
- врівноважує потреби зацікавлених осіб;
- втілює рішення на основі логічного обґрунтування;
- може відповідати деякому архітектурному стилю;
- на архітектуру впливає її оточення;
- впливає на структуру колективу;
- представлена в кожній системі;
- має особливу область застосування.

Інформаційна система з надання допомоги безпритульним тваринам більш схожа на цільову сторінку, метою якої є посилення ефективності реклами щодо благодійної діяльності, збільшення аудиторії (волонтерів та людей, які готові взяти під опіку тварин). Але в даній інформаційній системі використовується комбінація зі звичайних та цільових сторінок.

Головна сторінка основного сайту – найменш використовувана форма цільової сторінки. При переході по посиланню відкривається стартова сторінка сайту, є можливість переходити в підрозділи. Такий тип зручний для seo-оптимізації, його легко підлаштувати під основні запити пошукових систем. Основною метою є спонукання користувача до дії саме зараз: вибрати собі тваринку для взяття під опіку та (або) стати волонтером.

Для посилення ефективності використовуються елементи, що спонукають і закликають до дії, яскраві і великі кнопки, дизайн мінімалізовано, акценти на основних перевагах.

Для успішного результату сторінки необхідно звертати увагу відвідувачів на будь-які засоби, лиш би змусити відвідувача зацікавитись нашими послугами. Для цього необхідно дотримуватись деяких порад:

– оскільки перше, що побачить відвідувач, буде заголовок (і підзаголовок) на сторінці, потрібно постаратись відразу зацікавити його;

– має бути дотримана правильність усіх текстів на сторінці – як синтаксично, так і граматично;

– потрібно завоювати довіру користувача;

– кнопки, які відображають мету нашої системи, повинні виділятися і бути в області видимості на екрані;

– скоротіть кількість посилань до мінімуму, щоб була присутня лиш цікава інформація;

– використовувати будь-які доступні засоби, щоб змусити відвідувача ще більше зацікавитися даними послугами.

3.1.1 Вибір архітектури системи

Архітектура – це базова організація системи, втілена в її компонентах, їх відношеннях між собою та оточенням, а також принципи, що визначають проектування та розвиток системи [9].

Для того, щоб побудувати правильну і надійну архітектуру і грамотно спроектувати інтеграцію програмних систем необхідно чітко слідувати сучасним стандартам в цих областях.

В дипломній роботі інформаційна система «Допомога тваринам» має трирівневу архітектуру (three-tier).

В такій трирівневій архітектурі:

1) програма-клієнт реалізує інтерфейс користувача, передає запити серверу застосувань і приймає від нього відповідь;

2) сервер застосувань реалізує бізнес-логіку і звертається із запитами до сервера «третього рівня» (наприклад, сервера бази даних за даними);

3) сервер третього рівня обслуговує запити сервера застосувань.

Великі обчислювальні навантаження можуть виявитися непосильними для архітектури клієнт-сервер. Коли безліч клієнтів одночасно роблять запити, сервер виявляється в положенні офіціанта з ресторану, який отримав

відразу занадто багато замовлень. Одне з можливих рішень полягає у веденні додаткової ланки між переднім (клієнт) і заднім (сервер) кінцями лінії обробки даних. Тоді на сервер покладається функція зберігання даних, на проміжну ланку – основну обробки, а на клієнтське – представлення результатів.

Клієнт – це інтерфейсний (зазвичай графічний) компонент, який складає перший рівень, власне додаток для кінцевого користувача. Перший рівень не повинен мати прямих зв'язків із БД, має бути навантаженим основною бізнес-логікою і зберігати стан додатка. На перший рівень може бути винесена і зазвичай виноситься найпростіша бізнес-логіка: інтерфейс авторизації, алгоритми шифрування, перевірка внесених значень на допустимість і відповідність формату, нескладні операції (сортування, групування, підрахунок значень) з даними, вже завантаженими на термінал.

Сервер додатків розташовується на другому рівні. На другому рівні зосереджена велика частина бізнес-логіки. Поза ним залишаються фрагменти, що експортуються на термінали, а також вміщені в третій рівень збережені процедури і тригери.

Сервер БД забезпечує зберігання даних і виноситься на третій рівень. Зазвичай це стандартна реляційна або об'єктно-орієнтована СКБД. Якщо третій рівень є БД разом зі збереженими процедурами, тригерами і схемою, яка описує додаток в термінах реляційної моделі, то другий рівень будується як програмний інтерфейс, що пов'язує клієнтські компоненти з прикладною логікою бази даних.

Основними перевагами виділення логіки додатка в окрему складову є можливість повторного використання логіки додатка, підвищення продуктивності використовуваного сервера бази даних, можливість масштабування системи в цілому і відносна незалежність системи від конкретного виробника СКБД.

Переваги такої архітектури очевидні:

– зміни на кожній з ланок можна здійснювати незалежно;

– знижуються навантаження на мережу, оскільки ланки не обмінюються між собою великими об'ємами інформації;

– забезпечується масштабування і проста модернізація устаткування і програмного забезпечення, що підтримує кожна з ланок, у тому числі оновлення серверного парку і термінального устаткування, СКБД і т.д.;

– застосування можуть створюватися на стандартних мовах програмування (Java, C/C++/C#, PHP тощо).

3.1.2 Обґрунтування вибору сервера додатків

В дипломній роботі інформаційна система з надання допомоги тваринам розроблена з використанням сервера Open Server.

Це спеціальний сервер, призначений для web-розробки, тестування web-проектів. Він включає в себе комплекс таких програм:

1) Apache – надійний та гнучкий сервер з відкритим кодом.
2) PHP – популярна мова програмування, призначена для використання у web-розробках.
3) PHPMyAdmin – спеціальний інструмент, побудований на мові PHP.
4) Nginx– ще один гнучкий сервер.
5) MySQL – система, призначення якої – керування базами даних.
6) та інші.

OpenServer має хороший інтерфейс, широкі можливості та сферу застосування. Він є одним із найкращих та найбільш використовуваних серверів. Однією із особливостей являється саме простота використання – зручне меню, легкість у створенні хостів, портативність та безліч інших критеріїв, що надають істотні переваги. Основним призначенням, як згадувалось попередньо, є розробка сайтів та їх тестування у локальній мережі.

Платфора Open Server має в наявності графічний інтерфейс з трьома мовами (російська, українська, англійська) на вибір, портативність (немає прив'язки до певного комп'ютера), швидке переключення модулів і зміна конфігурації, що помітно збільшують зручність роботи з даними локальних веб сервером.

Зручність і простота управління забезпечують перевагу у використанні саме цього сервера. За час свого існування Open Server зарекомендував себе як якісний і надійний інструмент, необхідний кожному веб-майстру.

3.1.3 Обґрунтування вибору системи управління базами даних

База даних – упорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних

потреб користувачів. У технічному розумінні включно й система управління БД.

Система управління базами даних – це комплекс програмних і мовних засобів, необхідних для створення баз даних, підтримання їх в актуальному стані та організації пошуку в них необхідної інформації.

Централізований характер управління даними в базі даних передбачає необхідність існування деякої особи (групи осіб), на яку покладаються функції адміністрування даними, що зберігаються в базі.

Головним завданням БД є гарантоване збереження значних обсягів інформації та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин: збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних).

В дипломній роботі для розробки інформаційної системи з надання допомоги безпритульним тваринам використовується MySQL – система, призначення якої – керування базами даних.

В OpenServer за замовчуванням містяться утиліти для роботи з MySQL:

– PhpMyAdmin;
– MySQL менеджер;
– SQLite менеджер.

PhpMyAdmin – утиліта, написана на мові PHP для адміністрування виключно MySQL серверів, якій віддають перевагу користуватися як новачки, так і професіонали. Надає візуальний інтерфейс для роботи з сервером MySQL, через який дуже легко управляти базами, користувачами і їх правами.

Інструмент володіє великою кількістю цікавих можливостей і характеристик:

1) можливість створювати і видаляти бази даних без введення команд вручну;

2) працювати безпосередньо з таблицями, створювати, копіювати, видаляти, перейменовувати або ж змінювати дані;

3) здійснювати супровід таблиць;

4) редагувати та додавати нові поля окремо;

5) виконувати SQL-запити, в тому числі і пакетного типу;

6) швидко додавати дані з допомогою вбудованого інструменту імпорту текстових файлів в таблиці;

7) створювати і переглядати необхідні дані усіх представлених у базі таблиць;

8) підтримка великої кількості експортованих форматів;

9) управління та адміністрування декількома серверами одночасно;

10) дозволяти і забороняти користувачам виконувати певні дії або обмежити їх привілеї;

11) можливість відобразити схему всієї бази даних в форматі pdf шляхом графіки;

12) пошук по базі даних;

13) продукт локалізована більш ніж на 50 мов по всьому світу.

Продукт у наочній формі показує альтернативне застосування засобів для роботи з базою даних. Більше не потрібно вводити команди для керування з допомогою консолі MySQL вручну, де допустивши помилку в один символ можна було в самому гіршому випадку втратити всі дані.

MySQL менеджер при підключення до локального MySQL сервера чи до віддалених підключень, власне кажучи, відображає панель адміністратора для роботи з сервером MySQL, в якій можна управляти базами даних, їх таблицями, індексами, тригерами та іншими сутностями. Також використовується для того, щоб не встановлювати різне додаткове ПЗ.

SQLite менеджер – ще один графічний клієнт для роботи з MySQL в OpenServer. Дана утиліта єдина з усього комплекту OpenServer, яка дозволяє працювати з СКБД SQLite, яка все ще користується певним попитом у вузьких колах. Вона дозволяє підключатися не тільки до локального сервера MySQL, але і до віддалених, а також підтримує роботу не тільки з MySQL, але і з іншими СКБД. У даної програми є ще й можливість легко і просто міняти скіни, які оформлені звичайними CSS файлами, отже, правити їх може будь-хто, який хоч трохи володіє знаннями даної мови.

3.1.4 Обґрунтування вибору мови програмування

Для реалізації інформаційної системи з надання допомоги безпритульним тваринам була обрана об'єктно-орієнтована мова PHP.

PHP була створена для генерації HTML-сторінок на стороні веб-серверу. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок. PHP підтримується переважною більшістю хостинг-провайдерів.

PHP інтерпретується веб-сервером в HTML-код, який передається на сторону клієнта. На відміну від таких скриптових мов програмування, як JavaScript, користувач не має доступу до PHP-коду, що є перевагою з точки зору безпеки, але значно погіршує інтерактивність сторінок. Але ніщо не забороняє використовувати PHP для генерування і JavaScript-кодів які виконаються вже на стороні клієнта. PHP – це мова, яка може бути вбудованою безпосередньо в html-код сторінок, які, в свою чергу коректно будуть оброблені PHP-інтерпретатором.

Значною відзнакою PHP від якого-небудь коду, що виконується на стороні клієнта, наприклад, JavaScript, є те, що PHP-скрипти виконуються на стороні сервера. Ви навіть можете конфігурувати свій сервер таким чином, щоб HTML-файли оброблялися процесором PHP, так що клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання скрипта.

Головною особливістю мови PHP є її практичність. PHP надає програмісту інструмент для швидкого й ефективного вирішення поставлених завдань. Вона вирізняється винятковою гнучкістю до потреб розробника. Хоча PHP традиційно рекомендують використовувати у поєднанні з HTML-кодом, проте PHP з таким же успіхом інтегрується і в JavaScript, WML, XML та інші мови Інтернет-програмування [10-13].

Також при розробці системи була використана мова розмітки гіпертексту – Hyper Text Markup Language (HTML).

Мова HTML інтерпретується браузером і відображається у вигляді документа, зручному для людини.

HTML – це не мова програмування, це мова розмітки, яка каже браузеру, як відображати вміст веб-сторінки, яка переглядається. Вона може бути простою чи складною, залежно від бажання веб-дизайнера, який її створює.

HTML-документ є текстовим файлом розмічений за допомогою спеціальних (текстових) команд. Текстовий формат представлення веб-документів був вибраний виходячи з основних вимог до веб-документу: простота, можливість безпосередньої інтерпретації в будь-якій операційній системі, мінімальний розмір файлу, зручність редагування і інтерпретації.

Мова розмітки гіпертекстових документів HTML дозволяє визначити різні типи елементів, що забезпечують функціональність документа: текстові фрагменти із заданими параметрами форматування, списки, таблиці, зображення, гіперпосилання і т.д. Елементи HTML оголошуються за

допомогою команд розмітки, званих тегами. Всі HTML-теги, що зустрічаються в тексті документа інтерпретуються браузером при відображенні документа [14].

Також при розробці інформаційної системи була використана формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки – CSS (англ. Cascading Style Sheets – каскадні таблиці стилів). Стилем або CSS називається набір параметрів форматування, який застосовується до елементів документа, щоб змінити їх зовнішній вигляд. Можливість роботи зі стилями здавна включають в розвинені видавничі системи і текстові редактори, тим самим дозволяючи одним натисканням кнопки надати тексту заданий, заздалегідь встановлений вид. Тепер це доступно і творцям сайту, коли колір, розміри тексту і інші параметри зберігаються в певному місці і легко «прикручуються» до будь-якого тегу. Ще однією перевагою стилів є те, що вони пропонують набагато більше можливостей для форматування, ніж звичайний HTML.

CSS являє собою потужну систему, яка розширює можливості дизайну і верстки веб-сторінок.

Стилі є зручним, практичним і ефективним інструментом при верстці веб-сторінок і оформлення тексту, посилань, зображень та інших елементів. Незважаючи на явні плюси застосування стилів, розглянемо всі переваги CSS, в тому числі і непомітні на перший погляд:

1) Розмежування коду і оформлення. Ідея про те, щоб код HTML був вільний від елементів оформлення типу установки кольору, розміру шрифту та інших параметрів, стара як світ. В ідеалі, веб-сторінка повинна містити тільки теги логічного форматування, а вигляд елементів задається через стилі. При подібному поділі робота над дизайном і версткою сайту може вестися паралельно.

2) Різне оформлення для різних пристроїв. За допомогою стилів можна визначити вид веб-сторінки для різних пристроїв виводу: монітора, принтера, смартфона, планшета і ін. Наприклад, на екрані монітора відображати сторінку в одному оформленні, а при її друку – в іншому. Ця можливість також дозволяє приховувати або показувати деякі елементи документа при відображенні на різних пристроях.

3) Розширені в порівнянні з HTML способи оформлення елементів. На відміну від HTML стилі мають набагато більше можливостей по оформленню елементів веб-сторінок. Простими засобами можна змінити колір фону

елемента, додати рамку, встановити шрифт, визначити розміри, положення і багато іншого.

4) Прискорення завантаження сайту. При зберіганні стилів в окремому файлі, він кешується і при повторному зверненні до нього витягується з кешу браузера. За рахунок кешування і того, що стилі зберігаються в окремому файлі, зменшується код веб-сторінок і знижується час завантаження документів. Кешем називається спеціальне місце на локальному комп'ютері користувача, куди браузер зберігає файли при першому зверненні до сайту. При наступному зверненні до сайту ці файли вже не викачуються по мережі, а беруться з локального диска. Такий підхід дозволяє істотно підвищити швидкість завантаження веб-сторінок.

5) Єдине стильове оформлення безлічі документів. Сайт це не просто набір пов'язаних між собою документів, а й однакове розташування основних блоків, і їх вид. Застосування однакового оформлення заголовків, основного тексту та інших елементів створює спадкоємність між сторінками і полегшує користувачам роботу з сайтом і його сприйняття в цілому. Розробникам ж використання стилів істотно спрощує проектування дизайну.

6) Централізоване зберігання. Стилі, як правило, зберігаються в одному або декількох спеціальних файлах, посилання на які вказується в усіх документах сайту. Завдяки цьому зручно правити стиль в одному місці, при цьому оформлення елементів автоматично змінюється на всіх сторінках, які пов'язані із зазначеним файлом. Замість того щоб модифікувати десятки HTML-файлів, досить відредагувати один файл зі стилем і оформлення потрібних документів відразу ж зміниться.

3.2 Керівництво користувача

Інтерфейс користувачів має бути максимально простим та легкодоступним, зручним у використанні.

Для даної системи було розроблено шаблон з шапкою та розділами меню, які доступні як гостю, так і відвідувачу.

Метою головної сторінки є зацікавлення користувача у даній інформації, тому важливо, щоб вона була яскравою та примітною. За допомогою анімації заголовок змінюється тричі, і несе в собі заклики до дії:

1) «Мрієш мати домашнього улюбленця?»;
2) «Бажаєш допомогти безпритульному?»;
3) «Долучайся до порятунку чотирилапого…».

Саме ці лозунги спонукають користувачів використати представлений функціонал системи.

Важливо надати можливість відвідувачам зручно виконати ці дії, тому за допомогою кнопок «Взяти під опіку» та «Стати волонтером», не витрачаючи багато часу, можна перейти до наступних розділів системи (рис. 3.5).

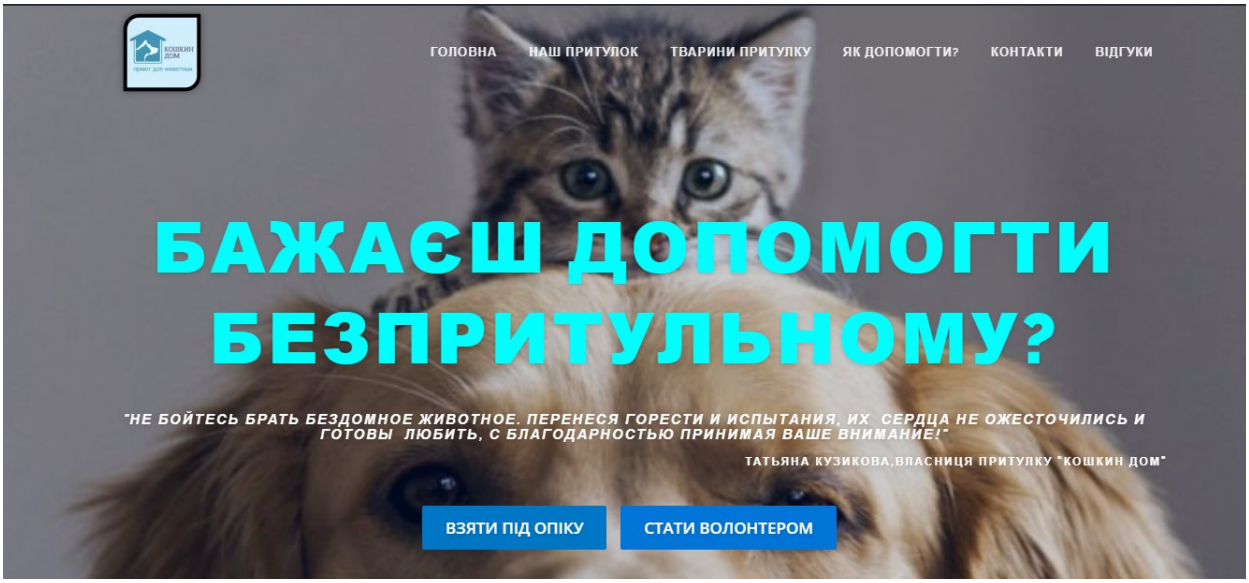


Рисунок 3.1 – Головний розділ інформаційної системи з надання допомоги безпритульним тваринам

Для переходу до наступних блоків головної сторінки відвідувачу достатньо декількох обертів колеса миші (рис. 3.2-3.4).

Рисунок 3.2 – Другий блок головного розділу інформаційної системи з надання допомоги безпритульним тваринам

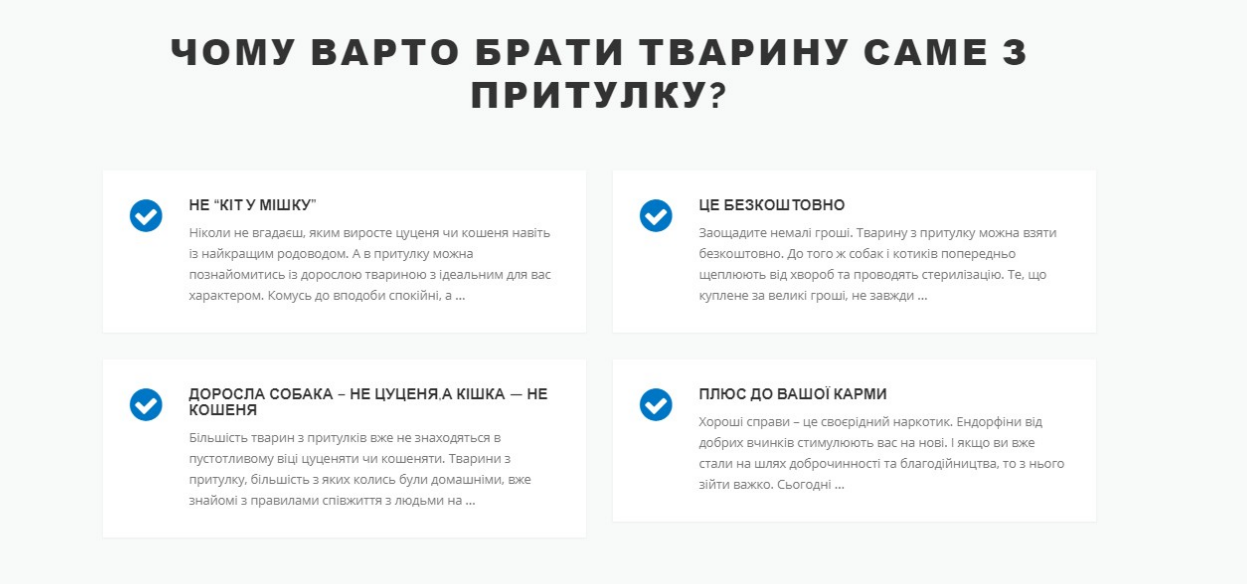


Рисунок 3.3 – Третій блок головного розділу інформаційної системи з надання допомоги безпритульним тваринам

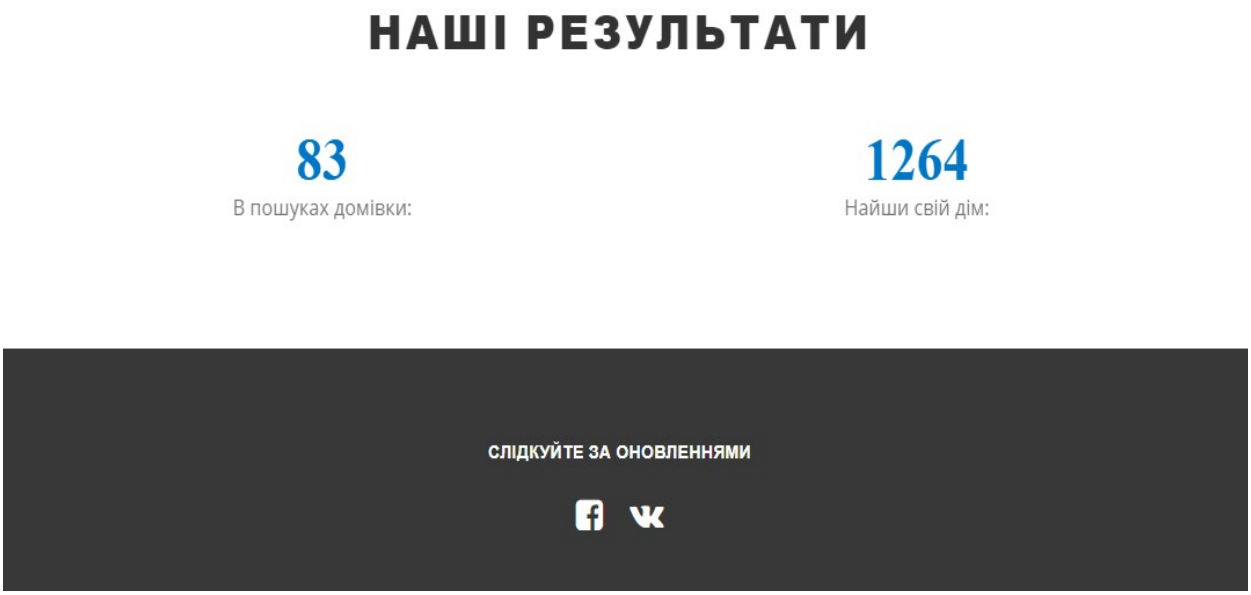


Рисунок 3.4 – Четвертий блок головного розділу інформаційної системи з надання допомоги безпритульним тваринам

Після розділу «Головна» представлено розділ «Наш притулок», в якому описується історія створення притулку, його діяльність. Розділ оформлено в нейтральних кольорах та заголовок виділено кольором (рис. 3.5).

Важливо надати можливість відвідувачам легко та зручно виконати дію, тому за допомогою кнопки «Взяти під опіку», яка розміщена на частині контекст-сторінки, є можливість перейти до наступного пункту меню «Тварини притулку». Даний пункт має дві категорії «Коти» та «Собаки» (рис. 3.6).

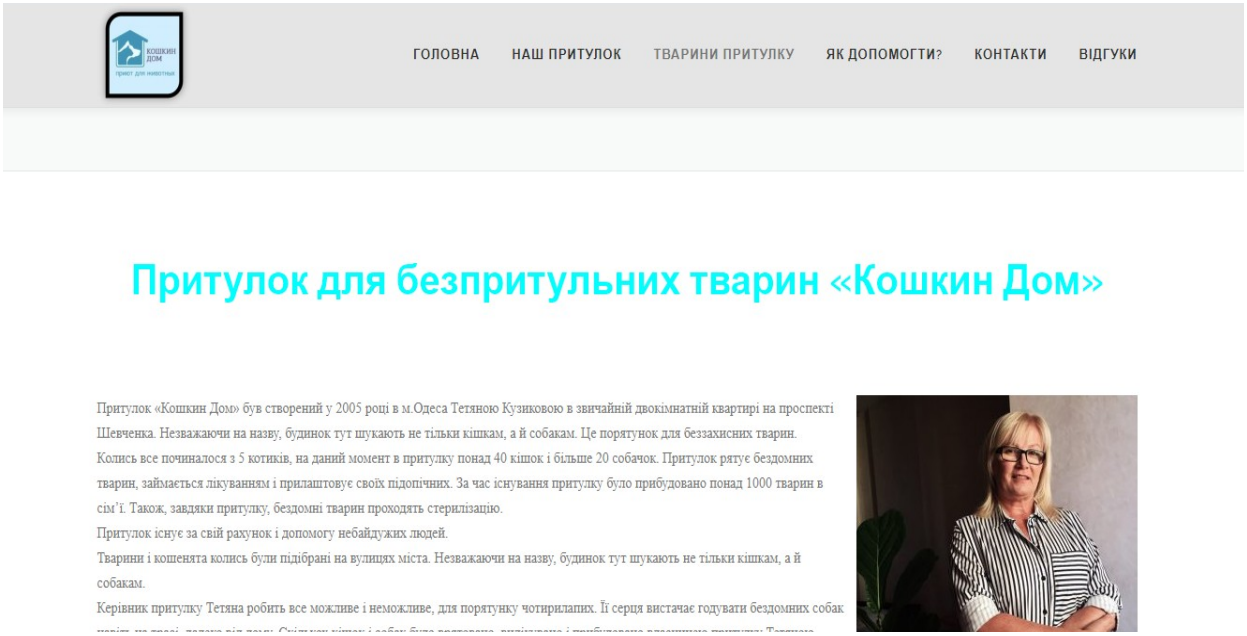


Рисунок 3.5 – Розділ «Наш притулок» для системи з надання допомоги безпритульним тваринам

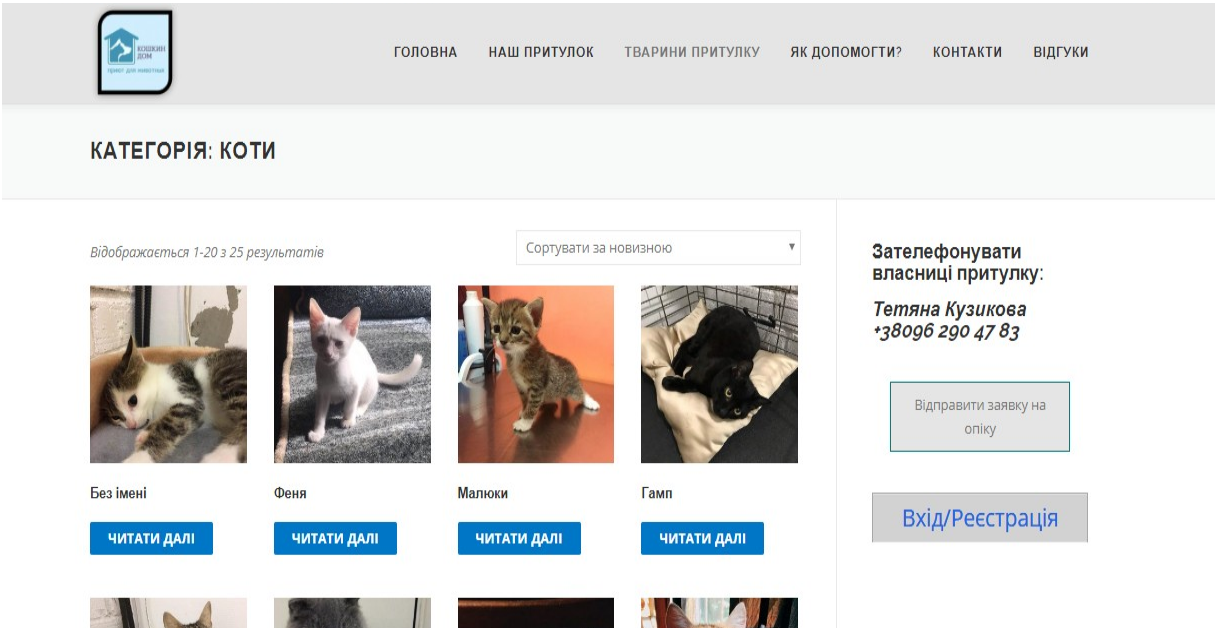


Рисунок 3.6 – Розділ «Тварини притулку», категорія «Коти» для системи з надання допомоги безпритульним тваринам

При виборі одної з них надається можливість перейти до каталогу тварин, де можна вибрати тварину та відправити заявку на опіку при умові реєстрації на сайті. В протилежному випадку дана опція не підтримується (рис. 3.7).

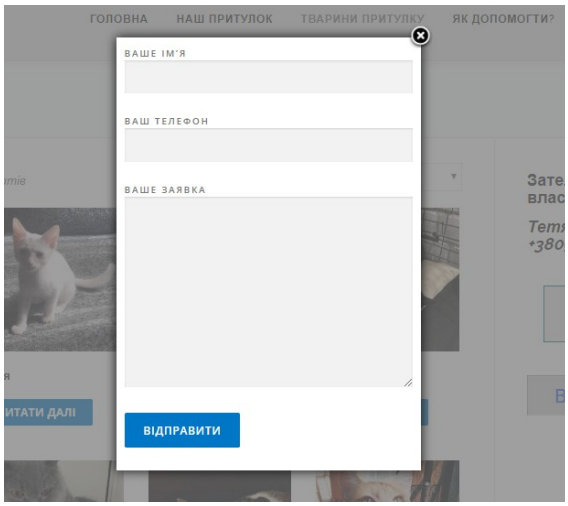


Рисунок 3.7 – Випливаюче вікно для оформлення заявки на опіку

Слідом за каталогом розміщено розділ «Як допомогти», до якого також можна перейти за допомогою кнопки «Стати волонтарем». Тут описується деяка інформація про волонтерський рух та їх напрямки роботи (рис. 3.8).

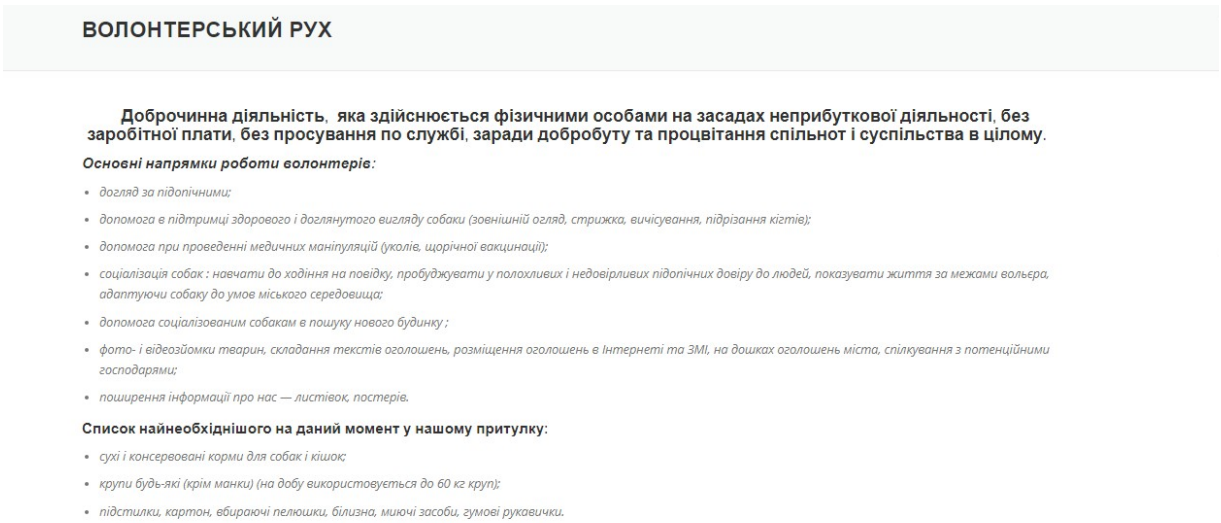


Рисунок 3.8 – Розділ «Як допомогти» для системи з надання допомоги безпритульним тваринам

Представлено також номер карточки власниці притулку для фінансової допомоги. Для зручності добавлено кнопки для переходу до веб-банкінгу деяких банківських установ нашої країни (рис. 3.9).

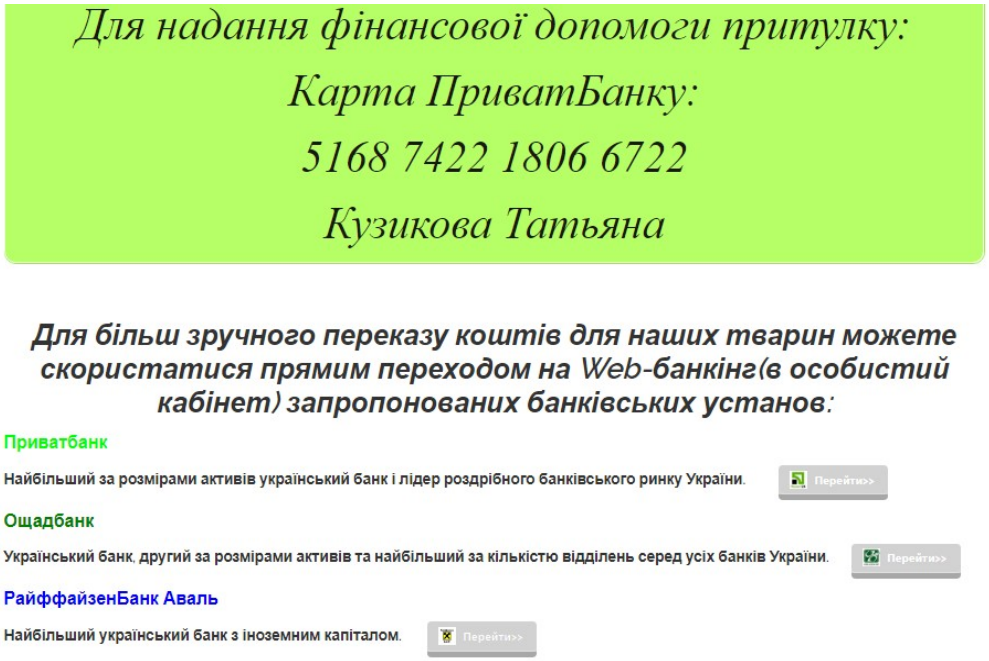


Рисунок 3.9 – Розділ «Як допомогти» для системи з надання допомоги безпритульним тваринам

Після перегляду розділу про допомогу, відвідувач може перейти до розділу «Відгуки». Тут надається можливість перегляду відгуків, залишених попередніми відвідувачами. В один момент на сторінці може відображатися декілька відгуків. Складаються вони безпосередньо із самого відгуку та ім`я людини, що залишає відгук.

Користувач-гість не може залишати відгуків, якщо він не пройшов авторизацію (рис. 3.10).

Користувач -відвідувач має можливість залишати відгуки. Йому достатньо в наданій формі ввести своє ім’я та сам відгук і натиснути на кнопку «Відправити». Через деякий час, після перевірки адміністратором на достовірність наданої інформації від відвідувача, відгук з’явиться поряд з іншими (рис. 3.11).

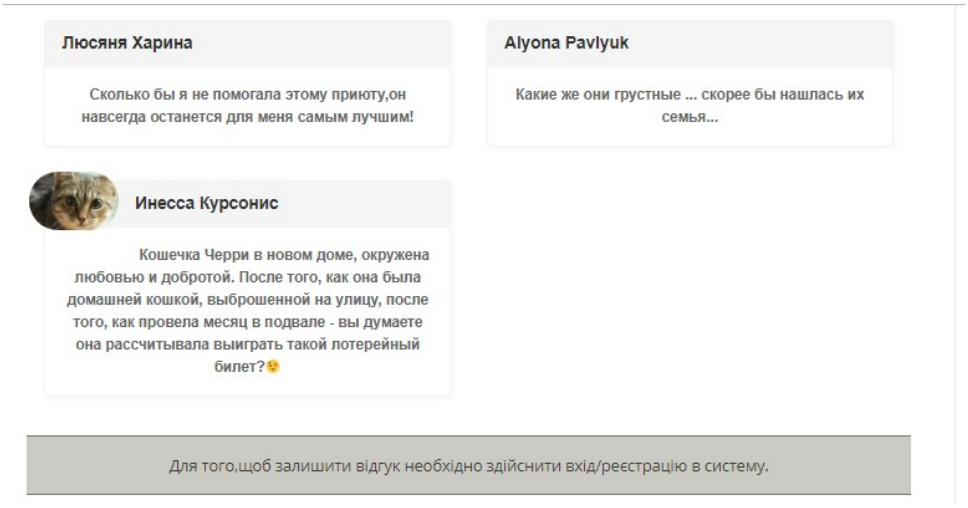


Рисунок 3.10 – Розділ «Відгуки» для системи з надання допомоги безпритульним тваринам

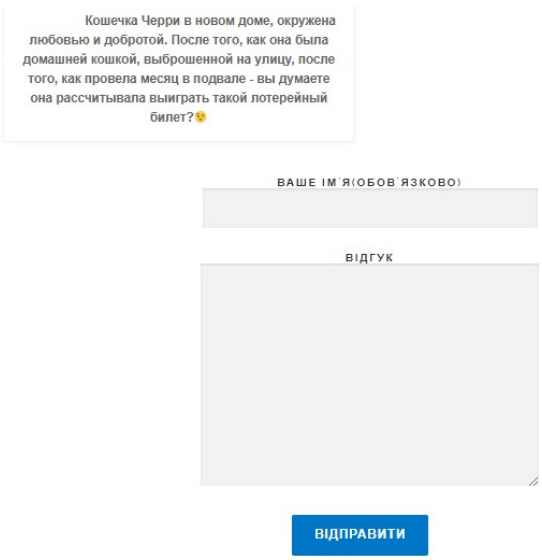


Рисунок 3.11 – Розділ «Відгуки» для системи з надання допомоги безпритульним тваринам

3.3 Керівництво адміністратора

Адміністратор – це найголовніша роль в WordPress. Вона призначається, коли виконується встановлення WordPress. Це дає повний контроль над усіма аспектами і настройками вашого сайту.

Роль адміністратора системи полягає в тому, щоб організовувати і підтримувати роботу кінцевих користувачів, тоді як забезпечення технічної

працездатності системи зазвичай забезпечується іншими людьми. Таким чином, адміністратор лише управляє роботою системи, наприклад:

– виконує певні дії з налаштування програми;
– керує обліковими записами користувачів;
– налаштовує права доступу;
– виправляє помилки в даних;
– забезпечує інформаційну підтримку.

Права адміністратора:

1) Записи: повний доступ: додавання, редагування, публікування і видалення будь-яких.

2) Сторінки: повний доступ: додавання, редагування та видалення будь-яких.

3) Управління коментарями: повний доступ.

4) Модулі: повний доступ: додавання, редагування та видалення будь-яких.

5) Теми: повний доступ: додавання, редагування та видалення будь-яких.

6) Користувачі: повний доступ: додавання, редагування та видалення будь-яких.

7) Налаштування WordPress: повний доступ.

Все управління системи здійснюється системним адміністратором через адмінпанель в CMS WordPress. Система управління вмістом (англ. Content management system, CMS) – комп'ютерна програма, яка використовується для управління вмістом чого-небудь (зазвичай це вміст розглядається як неструктуровані дані предметної задачі на противагу структурованих даних, які під управлінням СКБД). Зазвичай такі системи використовуються для зберігання і публікації великої кількості документів, зображень, музики або відео.

Інтерфейс системи управління WordPress дуже простий та інтуїтивно зрозумілий. Зліва розташована панель з усіма можливими пунктами, для редагування інформаційної системи з надання допомоги безпритульним тваринам.

Для того, щоб перейти в режим адміністрування сайтом, потрібно перейти на сторінку за адресою: http: // <адрес_сайта> / wp-admin. Відразу відкриється вікно входу в панель управління, де, ввівши логін і пароль, які були отримані при реєстрації, можна потрапити вже в саму панель управління (рис. 3.11).

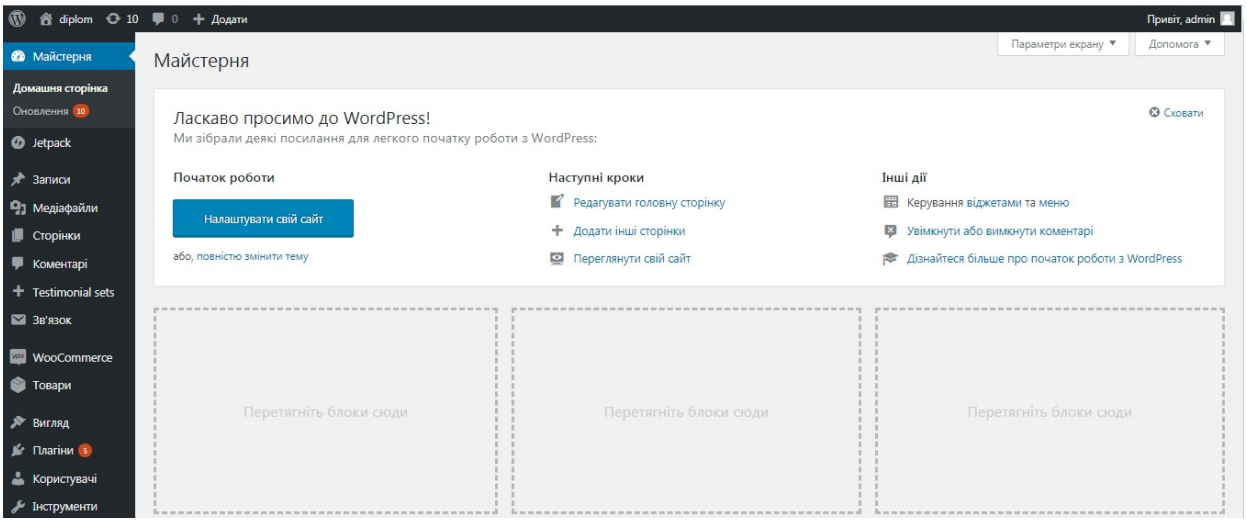


Рисунок 3.11 – Стартова сторінка панелі адміністрування CMS Wordpress

Наступний розділ меню редактора – «Записи». У записах можна переглянути всі наші записи, додати новий запис, а також рубрики і мітки, це якраз те місце, де створюються нові публікації. У редакторі, який знаходиться в цьому розділі, можна набрати текст, вставити текст або файл, вставити посилання, виділити шрифт або змінити його колір, можна вставити html-код і багато інших можливостей, які потрібні для опублікування матеріалів. Звідси є доступ до всіх публікацій в рубриках та до їх редагування.

Для зміни шаблону необхідно зайти в пункт «Вигляд» → «Теми» (рис. 3.12).

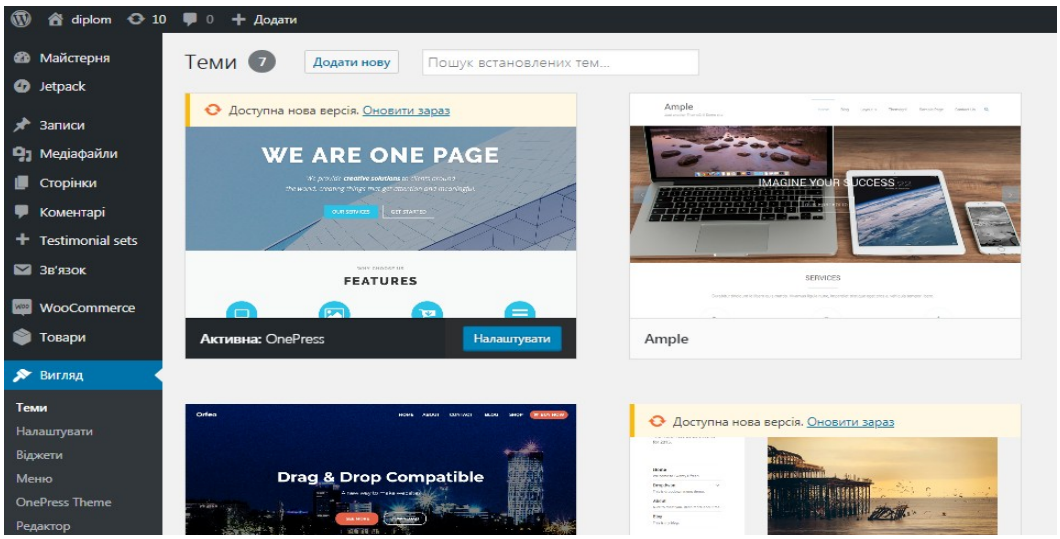


Рисунок 3.12 – Сторінка пункту «Вигляд», підрозділ «Теми» панелі адміністрування CMS Wordpress

Тут можна вибрати інший шаблон і встановити його. Або завантажити новий шаблон і налаштувати його на свій вибір і смак. Також цей розділ має ще декілька пунктів. У пункті «Налаштувати» є підрозділи. У підрозділі «Властивості сайту» можна змінити назву сайту, короткий опис і змінити значок сайту на вкладці браузера. У підрозділі «Кольори» можна змінити колір тексту заголовка і вибрати колір фону. У підрозділі «Зображення заголовка» можна змінити зображення заголовка встановивши індивідуальну картинку для організації. У підрозділі «Фонове зображення» можна скорегувати фон сайту, та в залежності від вибраної теми надаються інші підрозділи, за допомогою яких також можна змінювати вміст даних. У підрозділі «Меню» створюються майбутні пункти меню, також налаштовується вкладеність пунктів меню. У підрозділі «Віджети» налаштовуються позиції сайту (як відомо, будь-яка сторінка складається з деякого числа блоків, тобто позицій, де розташоване те чи інше, будь то місце для пунктів меню або місце для банерів).

В розділі «Плагіни» можна додавати новий функціонал для сайту (рис. 3.13).

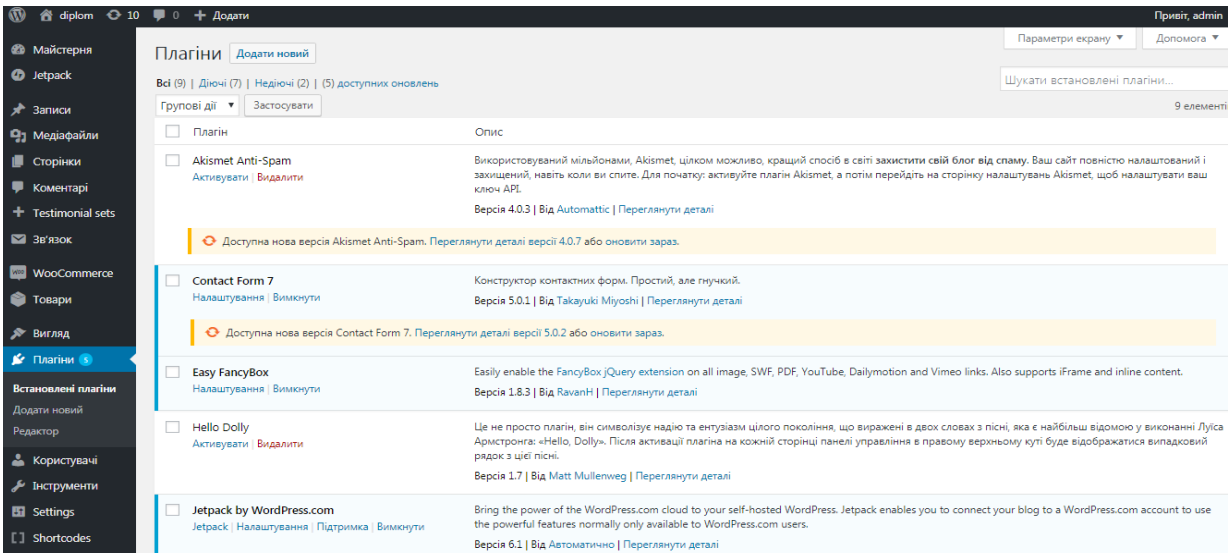


Рисунок 3.13 – Сторінка пункту «Плагіни» панелі адміністрування CMS Wordpress

У розділі «Сторінки» розташовується головна сторінка сайту. Тут її можна відредагувати. Також тут розташовані решта сторінок сайту (рис. 3.14).

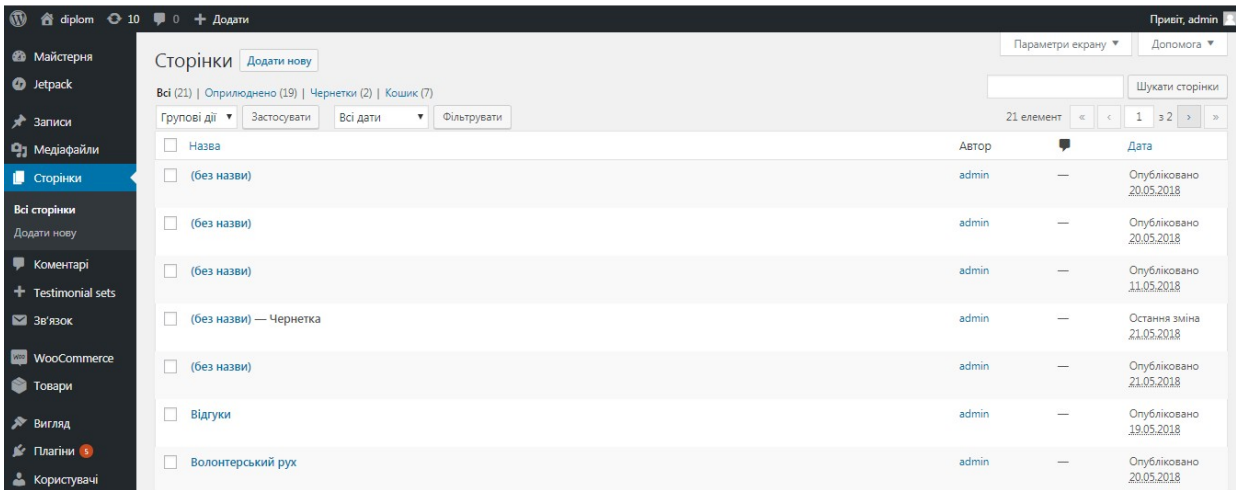


Рисунок 3.14 – Сторінка пункту «Сторінки» панелі адміністрування CMS Wordpress

Наступний розділ меню редактора – «Медіафайли» або бібліотека файлів. У ньому знаходяться всі файли, які завантажуються в бібліотеку. Це можуть бути картинки, медіа та відеофайли, з метою подальшої вставки в текст або у вигляді посилання для скачування читачами вашого ресурсу. При чому треба мати на увазі, що відображатися будуть тільки ті файли, які завантажені через панель адміністратора, завантажені безпосередньо на хостинг відображатися не будуть (рис. 3.15).

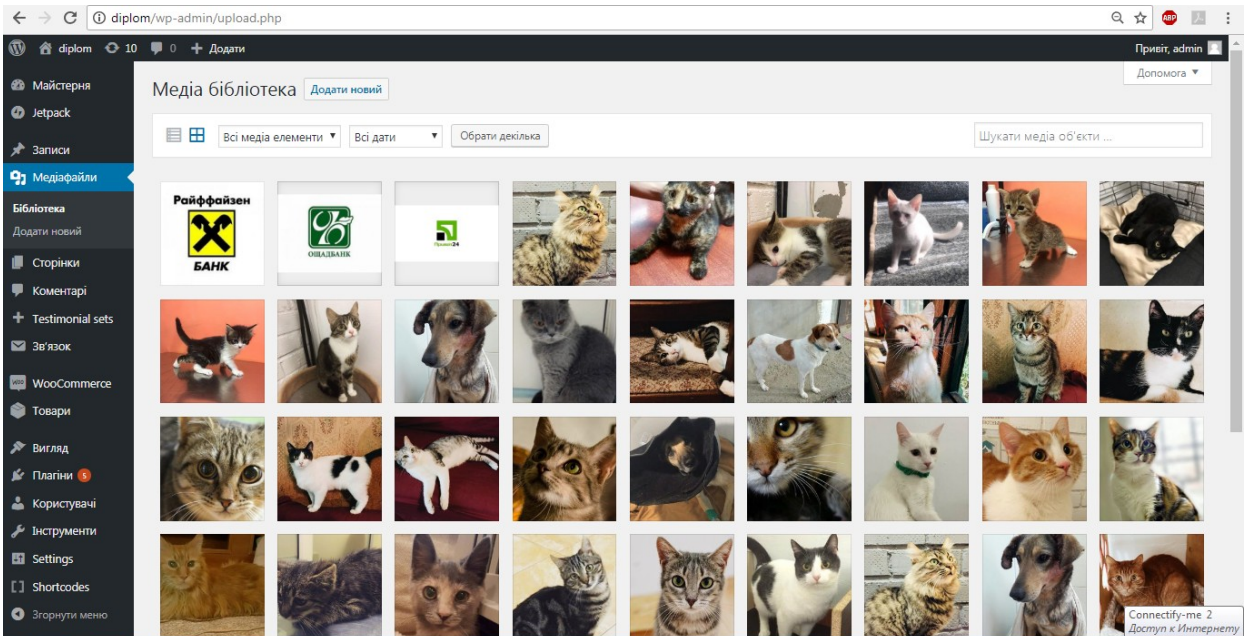


Рисунок 3.15 – Інтерфейс панелі адміністратора, при завантаженні зображень до системи

Адміністратор може поділитись посиланням на інформаційну систему у соціальних мережах, таким чином створивши можливість слідкувати за оновленнями не тільки на сайті (рис. 3.16).

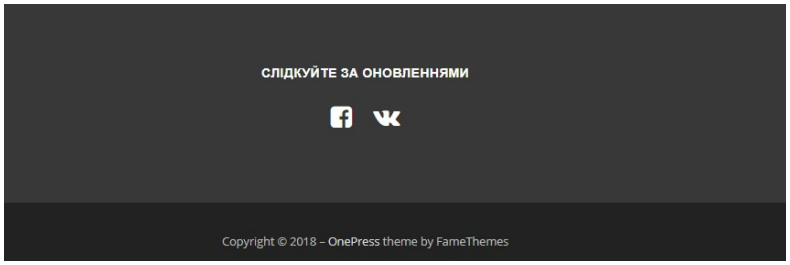


Рисунок 3.16 – Інтерфейс панелі адміністратора, при завантаженні зображень до системи

Системний адміністратор також має можливість додавати нових користувачів та надавати їм права доступу до інформації роботи в веб-системі. Тільки системний адміністратор може активувати, додати або змінити дані користувача. Кожному користувачеві відповідає група користувачів системи, яким роздаються права доступу (рис. 3.17).

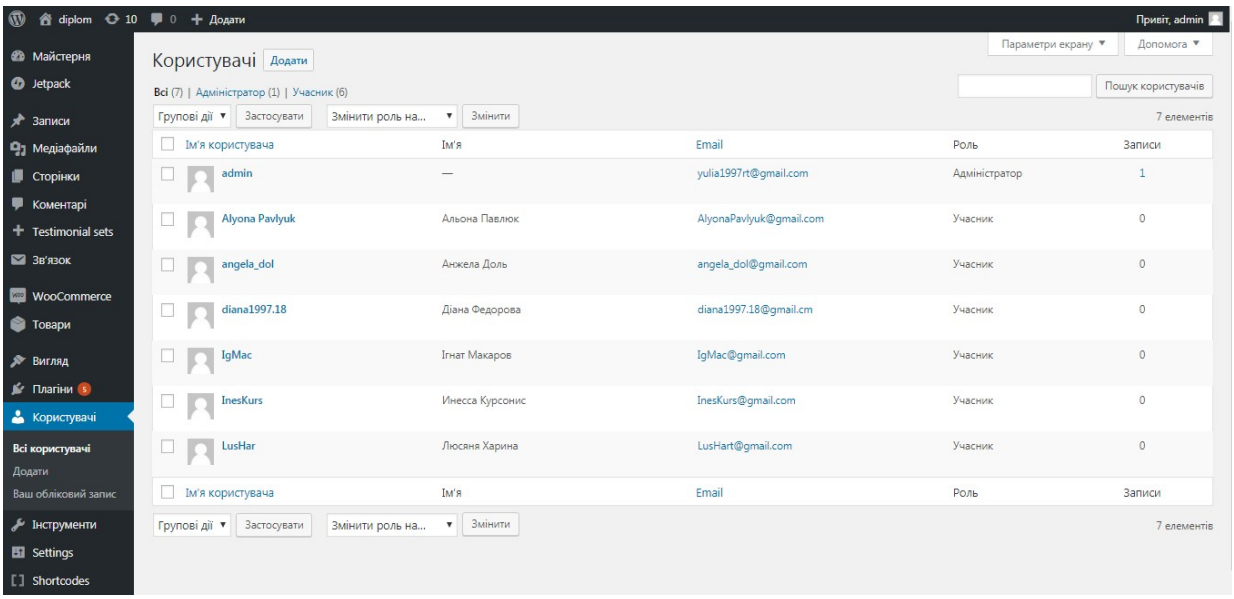


Рисунок 3.17 – Інтерфейс панелі адміністратору при управлінні користувачами

У CMS WordPress визначені такі групи користувачів: учасник, помічник, автор, редактор та адміністратор (рис. 3.18).

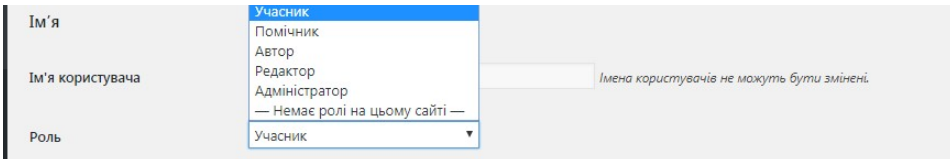


Рисунок 3.18 – Налаштування прав користувача

Для того, щоб мати можливість залишати відгуки та відправити заявку на опіку, необхідно увійти в систему, тому кожен користувач вносився в групу Учасник. Користувач-гість, який не зареєстрований в системі і не занесений до бази даних не має на це право.

ВИСНОВКИ

У результаті виконання дипломного проекту була розроблена інформаційна система з надання допомоги бездомним тваринам для притулку «Кошкин Дом».

Необхідність реалізації даної інформаційної системи полягає в поширенні допомоги такого виду в країні та, одночасно, впроваджуючи її в мережу Інтернет, врятувати хоч малу частину життів та здоров'я тварин.

Система дозволяє встановити зручний зв'язок з притулком та надасть можливість допомогти йому. Система представляє користувачам інформацію про даний заклад, дає можливість перегляду каталогів тварин, які на даний час шукають родину та потребують допомоги.

Для забезпечення зручного користування веб-системою виконано наступне:

1) розроблений зрозумілий інтерфейс та зручна форма перегляду представленої інформації;
2) надана можливість легкого доступу до потрібної інформації;
3) представлена докладна інформація щодо притулку;
4) в системі є каталог тварин та можливість відправити заявку на опіку;
5) є перелік послуг, що надаються (стати волонтером, стати опікуном);
6) представлений оновлений звіт про роботу притулку у вигляді новин тижня;
7) можливість додавати та дивитись відгуки відвідувачів.

Для реалізації інформаційної системи було вибрано архітектуру системи, яка представляє собою трирівневу, також проведено вибір програмних засобів реалізації. Здійснено розмежування прав доступу певних користувачів системи: користувач – гість, користувач – відвідувач, користувач – адміністратор.

Функціями для користувача-відвідувача є можливість перегляду каталогу тварин, стати опікуном (відправити заявку на її опіку), додати відгук, ознайомлення з наданою інформацією, стати волонтером.

Функціями для користувача-гостя є можливість перегляду каталогу тварин та відгуків відвідувачів, стати волонтером та ознайомлення з наданою інформацією.

Функціями для користувача-адміністратора є можливість управління авторизацією в системі, додавання, видалення, редагування будь-якої

інформації, керування правами користувачів, повний контроль над працездатністю системи.

Завдяки використанню сучасних Web-технологій вдалося створити максимально простий та легкодоступний інтерфейс для різних категорій користувачів.

Система реалізована з використанням сучасних програмних засобів: CMS WordPress, спеціального сервера Open Server, який включає в себе комплекс різних програм, мова програмування PHP, CSS, мова розмітки HTML.

Отже, розроблена інформаційна система з надання допомоги безпритульним тваринам є закінченою Web-системою з достатньо широким функціоналом для комфортної роботи з системою.

ПЕРЕЛІК ПОСИЛАНЬ

1. Веб-система київського благодійного фонду «Happy Paw». [Електронний ресурс] – Режим доступу: http://happypaw.ua/ua

2. Веб-система притулку для тварин «Сіріус». [Електронний ресурс] – Режим доступу: http://www.dogcat.com.ua/

3. Веб-системою буде БФ «Благомай». [Електронний ресурс] – Режим доступу: http://charitymay.com/uk/we-help-uk1/help-animals_uk

4. Веб-система «GladPet». [Електронний ресурс] – Режим доступу: https://gladpet.org/

5. Гайна Г.А. Основи проектування баз даних: Навчальний посібник. − К.: КНУБА, 2005. – 204 с.

6. Коннолли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика. Уч. пос. – М.: Издательский дом «Вильямс», 2000 - 761 с.

7. Басс Л., Клементс П., Кацман Р. Архитектура програмного обеспечения на практике. – СПб.: Питер, 2006 – 575 с.

8. Що таке архітектура програмного забезпечення. [Електронний ресурс] – Режим доступу: http://easy-code.com.ua/2012/07/shho-take-arxitektura-programnogozabezpechennya-rizne-programuvannya-statti/

9. Рыбальченко М.В. Архитектура информационных систем. Уч. пос. – М.: Издательство «Юрайт», 2016. – 91 с.

10. Бенкен Е. MySQL, XML. Программирование для Интернета. – М.:, БХВ-Петербург, 2007. – 336 с.

11. Мазуркевич А., Еловой Д. MB PHP: настольная книга программиста. – Минск: Новое знание, 2003. – 480 с.: ил.

12. Ульман Л. Основы программирования на PHP:/Ларри Ульман. Пер. с англ. – М.: ДМК Пресс, 2001. – 288 с.: ил. (Самоучитель).

13. Стивен Хольцнер. PHP в примерах. – М.: «Бином-Пресс», 2007. – 352 с.

14. А. Петюшкин. HTML в Web-дизайне. – М.:, БХВ – Петербург, 2004.– 400 с.