КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: «Інформаційна система онлайн готелю для тварин»

Власюк Віталій Володимирович

(прізвище, ім’я та по батькові студента повністю)

Київ 2023 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТ
д.т.н., професор Цюцюра С.В.

«___» _____20___ року

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: «Інформаційна система онлайн готелю для тварин»

Виконав:студент 4-го курсу, групи КН-42с

Спеціальності: 122 «Комп`ютерні науки

.

Спеціалізація: _«Інформаційні управляючі системи та технології»

(шифр і назва напряму підготовки, спеціальності)

Власюк В.В. .

(прізвище та ініціали)

Керівник _д.т.н., проф. Цюцюра С.В.

(прізвище та ініціали)

Рецензент _к.т.н., доц. Шабала Є.Є.

(прізвище та ініціали)

Київ 2023 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

Факультет: __автоматизації і інформаційних технологій
Кафедра:__ __інформаційних технологій
Освітній рівень: «бакалавр» за ОПП
Спеціальність: _122 «Комп`ютерні науки»
Спеціалізація:__Інформаційні управляючі системи та технології

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТ
д.т.н., професор Цюцюра С.В.
_____
„___” _____2023 року

З А В Д А Н Н Я
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

Власюк Віталій Володимирович
(прізвище, ім’я, по батькові)

1. Тема роботи: Інформаційна система онлайн готелю для тварин
керівник роботи: Горда Олена Володимирівна, д.т.н, професор кафедри інформаційних технологій
затверджені наказом ректора КНУБА № 1811/2 від « 17» листопада 2022 р.
2. Термін подачі студентом роботи до захисту: 01 червня 2023 .
3. Вихідні дані до роботи _____
4. Зміст пояснювальної записки: Вступ 1. Аналіз та дослідження проблеми. 2. Проєктування програмного забезпечення. 3. Практична реалізація. 4. Техніко-економічне обгрунтування розробки підсистеми (Бізнес-план)
5. Перелік презентаційно-інформаційних слайдів: 1. Розробка інформаційної системи онлайн готелю для тварин. 2. Аналіз предметної області. 3. Дерево функцій. 4. Концептуальна модель. 5. База даних системи. 6. Алгоритм реєстрації користувача. 7. Алгоритм запису даних. 8. Вигляд сайту. 9. Запис інформації в базу даних. Тестовий приклад роботи програмного продукту.

6. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис
Техніко-економічне обґрунтування розробки підсистеми (Бізнес-план)	д.т.н. проф. Цюцюра С.В.		
Прийом програмного продукту	к.т.н., доц. Горда О.В.		

7. Дата видачі завдання: 15 лютого 2023 р.

КАЛЕНДАРНИЙ ПЛАН

Види робіт та їх зміст	Дата виконання
Р. 1. Аналіз предметної області та постановка задачі	Лютий 2023 р.
Р. 2. Розробка інформаційного забезпечення	Лютий 2023 р.
Р. 3. Розробка програмного забезпечення. Тестовий приклад програми	Березень 2023 р.
Р. 4. Техніко-економічне обґрунтування розробки підсистеми (Бізнес-план)	Квітень 2023 р.
Оформлення роботи	Травень 2023 р.
Направлення роботи на рецензування	Травень 2023 р.
Попередній захист роботи на кафедрі	Червень 2023 р.

Бакалавр _____ Власюк В.В.
(підпис) (прізвище та ініціали)

Керівник _____ Горда О.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Власюк В.В. «Інформаційна система онлайн готелю для тварин».

Атестаційна випускна робота бакалавра за спеціальністю: 122 «Комп'ютерні науки», спеціалізація: «Інформаційні технології». – Київський національний університет будівництва та архітектури. – Київ, 2023.

У атестаційній випускній роботі був проведений аналіз існуючих інформаційних онлайн систем готелів для тварин, обумовлене популярністю утримання тварин в домашніх умовах.

Була створена практична реалізація веб-частини та частини баз даних. Було проведено їх тестування, та перевірку на працездатність.

ANNOTATION

Vlasiuk V.V. "Information system of online hotel for animals".

Bachelor's thesis for a bachelor's degree in specialty: 122 "Computer Science", specialization: "Information Technology." - Kyiv National University of Construction and Architecture - Kyiv, 2023.

In the certification graduation work, an analysis of existing online information systems of hotels for animals was carried out due to the popularity of keeping animals at home.

A practical implementation of the web and database parts was created. They were tested and checked for performance.

ЗМІСТ

ВСТУП 8

1 АНАЛІЗ ТА ДОСЛІДЖЕННЯ ПРОБЛЕМИ 11

1.1 Аналіз проблеми 11

1.2 Формулювання проблеми 12

1.3 Дерево основних цілей 13

1.4 Аналіз існуючих рішень 15

1.5 Постановка задачі 17

1.6 Дерево функцій 18

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 20

2.1 Технічна платформа 20

2.2 Проєктування бази даних 25

2.2.1 Створення концептуальної моделі 26

2.2.2 Створення логічної моделі інформаційної системи онлайн готелю для тварин 28

2.2.3 Створення фізичної моделі інформаційної системи онлайн готелю для тварин 30

2.3 Проєктування серверної частини 31

2.4 Перевірка алгоритму 32

2.5 Розробка шаблону 35

2 ПРАКТИЧНА РЕАЛІЗАЦІЯ 37

3.1 Створення шаблону в Figma. 37

3.2 Розробка фронтенд частини 42

3.3 Розробка бекенд частини 50

3 БІЗНЕС- ПЛАН РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН-ГОТЕЛЮ ДЛЯ ТВАРИН 56

3.2 Резюме проєкту 56
3.3 Оцінка збуту ринку 56
3.4 Конкуренція 57
3.5 План розробки 58
3.6 Організаційний план 58
3.7 Календарний графік виконання проєкту 59
3.8 Сітьовий графік 63
3.9 WBS – модель проєкту 66
3.10 OBS – модель проєкту 67
3.11 План виконання проєкту 68
3.12 Конфлікти в проєкті 70
ВИСНОВОК 72
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 73

ВСТУП

Актуальність дослідження. Тема «Інформаційна система онлайн готелю для тварин» є дуже актуальною в наш час. В сучасному світі зростає популярність тримання домашніх тварин, включаючи собак, котів, гризунів та птахів. Разом з цим, збільшується потреба у комфортному і безпечному розміщенні цих тварин під час відсутності їх власників. У зв'язку з цим, розробка інформаційної системи, яка дозволить власникам тварин бронювати номери у готелі для тварин онлайн, є актуальною та потребує подальшого дослідження.

Мета дослідження. Метою даної дипломної роботи є розробка та впровадження інформаційної системи онлайн готелю для тварин. Основним завданням є створення зручного та надійного сервісу для власників тварин, який дозволить їм зручно та швидко бронювати номери для своїх улюбленців, переглядати інформацію про доступність номерів.

Об'єкт дослідження. Об'єктом дослідження є інформаційна система онлайн готелю для тварин. Дослідження охоплює аналіз та розробку функціональності системи, проектування бази даних, розробку інтерфейсу користувача.

Предмет дослідження. Розробки та впровадження інформаційної системи онлайн готелю для тварин. Включаючи аналіз вимог користувачів, розробку функціональності, створення бази даних, реалізацію інтерфейсу та тестування системи.

Методика та завдання дослідження. У даній роботі буде використана комбінація методів, таких як літературний огляд, аналіз вимог користувачів, проектування системи, реалізація програмного забезпечення та експериментальне тестування. Основними завданнями дослідження є:

- Аналізування вимог користувачів та функціональних можливостей системи;
- Проектування бази даних та розробка модулів системи;
- Реалізація інтерфейсу користувача;

- Проведення тестування та оцінка ефективності системи.

Практична значимість. Результатом дослідження буде розроблена інформаційна система онлайн готелю для тварин, яка забезпечує зручне та швидке бронювання номерів для власників тварин.

Ця система сприятиме покращенню обслуговування клієнтів, забезпечить їх зручність та надасть інноваційний продукт на ринку. Практична значимість полягає у вдосконаленні процесу бронювання номерів для тварин та забезпеченні зручності та надійності для власників та тварин.

Результати дослідження. Очікується, що результатом дослідження буде розроблена та впроваджена інформаційна система онлайн готелю для тварин. Система буде мати можливість бронювання номерів онлайн, перегляду інформації про доступність номерів. Також будуть проведені тестування та оцінка ефективності системи.

Короткий зміст розділів:

- Перший розділ вивчається проблема бронювання номерів для тварин у готелях, включаючи труднощі та недоліки існуючих систем. Встановлюється основна мета та завдання дослідження для вирішення проблеми бронювання номерів для тварин. Будується ієрархія цілей, які потрібно досягти в розробці інформаційної системи онлайн готелю для тварин. Аналізуються рішення у галузі баз даних та систем розумних речей, включаючи підходи, технології та платформи. Формулюється конкретна постановка задачі, включаючи опис функціоналу та вимог до системи онлайн готелю для тварин. Будується дерево функцій, яке відображає функціональні можливості інформаційної системи онлайн готелю для тварин.
- В другому розділі визначається технічна платформа, на якій буде реалізована інформаційна система. Розглядаються апаратне забезпечення, операційна система та мережеві протоколи. Здійснюється проєктування бази даних. Визначаються сутності,

атрибути та зв'язки між ними. Здійснюється проєктування серверної частини системи. Визначаються компоненти та протоколи комунікації. Відбувається перевірка алгоритмів системи. Виконуються тестування та аналіз результатів. Здійснюється розробка шаблону інтерфейсу системи. Встановлюються зовнішній вигляд, структура та функціонал елементів інтерфейсу.

- У третьому розділі створюється шаблон інтерфейсу системи в Figma, включаючи дизайн елементів та їх взаємодію. Розробляється фронтенд частина системи, використовуючи HTML, CSS та JavaScript для створення інтерфейсу та користувацької взаємодії. Проводиться розробка бекенд частини системи, використовуючи мови програмування PHP, для реалізації логіки бізнес-процесів та взаємодії з базою даних.
- Четвертий розділ описує бізнес-аспекти проєкту, включаючи створення резюме продукту, опис проектованого продукту та виду послуги, оцінку ринку збуту, конкуренцію, стратегію маркетингу, план виробництва та організаційний план, юридичні аспекти, оцінку ризиків та страхування, фінансовий план та стратегію фінансування.

1 АНАЛІЗ ТА ДОСЛІДЖЕННЯ ПРОБЛЕМИ

1.1 Аналіз проблеми

Актуальність поставленої задачі щодо розробки інформаційної системи онлайн готелю для тварин випливає з ряду факторів, які відображають сучасні тенденції та потреби суспільства. По-перше, утримання тварин у власних сім'ях стає все більш популярним явищем. Зростання усвідомлення важливості тварин як членів родини спонукає власників шукати надійні та зручні готельні послуги для своїх улюбленців під час відсутності. Це створює актуальну потребу у розвитку готельного бізнесу для тварин та підвищує значущість розробки інформаційної системи, що забезпечує ефективне бронювання та управління послугами для тварин.

По-друге, інтернет-технології та онлайн-сервіси здобувають все більшу популярність та широке використання в різних сферах життя. Це створює нові можливості для створення ефективної інформаційної системи для готелів для тварин. Завдяки цій системі, власники тварин зможуть зручно та оперативно здійснювати бронювання, отримувати доступ до інформації про готельні послуги та комунікувати з готелями.

Проаналізувавши існуючі системи та практики бронювання готелів для тварин, виявлено кілька проблем, з якими стикаються власники тварин та готелі. Недостатня кількість доступних місць у готелях для тварин, неефективне управління бронюванням, недостатній рівень зручностей та послуг для тварин, а також відсутність зручних засобів комунікації між власниками та готелями є лише кількома з них.

Отже, актуальність поставленої задачі полягає у необхідності розробки інформаційної системи, що вирішує вищезазначені проблеми та надає зручний та ефективний механізм бронювання та управління готельними послугами для тварин. Така система має потенціал сприяти розвитку готельного бізнесу для тварин та задовольнити потреби власників та їх

улюбленців, покращуючи їх досвід та забезпечуючи їх спокій під час розлуки.

1.2 Формулювання проблеми

Проблема, яку ми ставимо перед собою, полягає в недостатній ефективності та зручності існуючих систем бронювання та управління готельними послугами для тварин.

Власникам тварин складно знайти надійні готельні заклади, які забезпечують відповідний рівень комфорту та догляду для їх улюбленців. Часто готелі для тварин мають обмежену кількість доступних місць або недостатньо зручних послуг для задоволення потреб тварин.

Основна мета дипломної роботи полягає у розробці інформаційної системи, яка вирішить цю проблему шляхом створення зручного та ефективного механізму бронювання та управління готельними послугами для тварин.

Наша система буде сприяти власникам тварин у пошуку надійних готелів, забезпечувати зручну процедуру бронювання та отримання інформації про доступні послуги, а також забезпечувати зручний засіб комунікації між власниками та готелями.

Формулювання проблеми є важливим етапом нашого дослідження, оскільки воно допомагає чітко визначити завдання та цілі нашої роботи. Розробка ефективної інформаційної системи для готелів для тварин відповідатиме потребам сучасного ринку та сприятиме поліпшенню досвіду власників тварин.

Наша робота вирішує актуальну проблему у галузі готелів для тварин, що робить її значущою та важливою для подальшого вивчення, розвитку та покращення готельного бізнесу для тварин.

1.3 Дерево основних цілей

Метою дослідження в цій роботі є покращення простоти процесу оформлення замовлення для наших користувачів. Шлях до досягнення цієї мети представлений у вигляді дерева цілей, яке відображено на рисунку 1.1. Це дерево цілей визначає основні аспекти, які потрібно враховувати та покращувати, щоб забезпечити зручну та ефективну процедуру оформлення замовлення.

Досліджуючи різні аспекти, такі як інтуїтивний інтерфейс, мінімізація кроків, які потрібно виконати, та оптимізація взаємодії з користувачем, щоб забезпечити максимальну простоту та зручність процесу оформлення замовлення. Дослідження цих аспектів дозволить нам створити інтуїтивний та легко зрозумілий інтерфейс, який спростить процес оформлення замовлення для наших клієнтів.

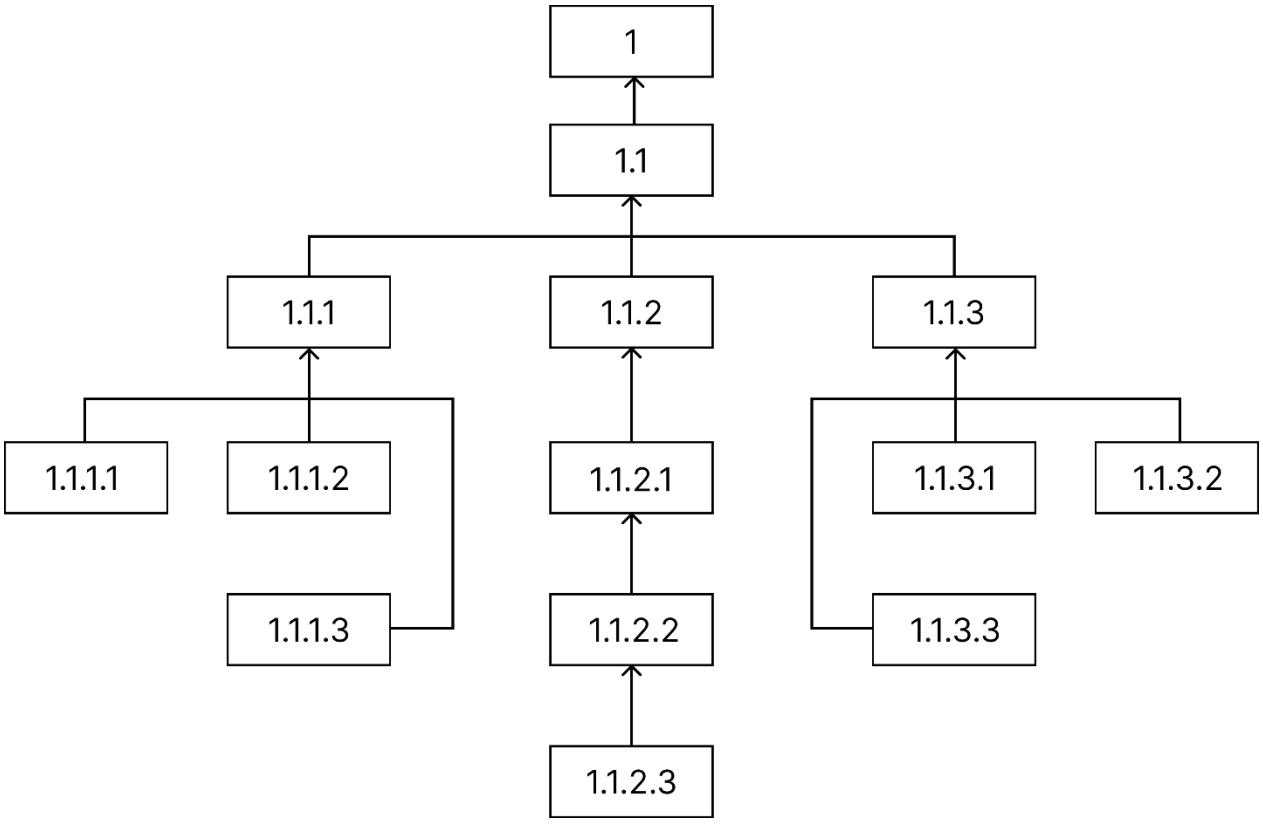


Рисунок 1.1 – Дерево цілей покращення простоти оформлення замовлення на сайті.

1. Основна ціль даного дерева, є зрозумілість та спрощення, оформлення замовлення для користувача, також доступна і зрозуміла подача інформації, для перегляду ключових пунктів сайту.

1.1. Наступною під ціллю є розробка самого сайту, який саме і покаже весь обсяг інформації. Він складається з трьох підпунктів:

1.1.1. Для розробки сайту потрібно зібрати вичерпну інформацію. Завдяки уважному збору даних з різних джерел можна краще виконати поставлену задачу та переконатися, що сайт відповідає потребам користувачів, які будуть його використовувати.

1.1.1.1. Збір інформації від команди розробників. Ціль зібрати інформацію, яка буде стандартизованою та буде відповідати всім поставленим вимогам.

1.1.1.2. Збір інформації від користувачів. Ціль зібрати інформацію, яка буде важливою для звичайних користувачів. Потрібно дізнатись основні тези та нюанси на які реагують звичайні користувачі.

1.1.1.3. Збір інформації від засновника. Потрібно заповнити певний перелік пунктів, які мають бути розташовані на сайті, перелік послуг та основна інформація про продукт.

1.1.2. Головна функція сайту полягає у забезпеченні користувачів необхідною інформацією. Для досягнення цієї мети потрібно визначити головний функціонал сайту, який повинен відповідати потребам користувачів. Основною метою є реалізація трьох підпунктів.

1.1.2.1. Оцінка швидкості зворотнього зв’зяку. Оцінити та зібрати всю інформацію по швидкості передачі особистих даних від клієнтів.

1.1.2.2. Тестування. Потрібно протестувати систему, та перевірити наявність всіх вказаних пунктів у розробці сайту.

1.1.2.3. Затвердження. Перегляд всієї зібраної інформації, для подальшого впровадження правок в систему.

1.1.3. Технічна реалізація пректу.

1.1.3.1. Розробка бек-енд частини. Мета полягає у розробці бекенд-частини з використанням сучасних технологій та забезпеченні її відповідності всім критеріям безпеки інформації, а також надання доступу за допомогою сучасних протоколів.

1.1.3.2. Розробка бази даних. Створення індивідуальної бази даних, для отримування інформації від користувача.

1.1.3.3. Мета - розробити фронтенд частину, яка забезпечить зручний та простий зовнішній доступ до інформації. Для цього використовуються актуальні технології.

1.4 Аналіз існуючих рішень

В сучасний час існує безліч сервісів та платформ, які допомагають аналізувати та вести метрику для веб-сайтів та бізнесів. Одним з найпопулярніших є система Google Analytics, яка є однією з найпоширеніших систем web-аналітики.

Ця система дозволяє користувачеві переглядати інформацію про трафік на сайті, а також основною її функцією є відстеження відвідувачів на сайті, що дозволяє детальніше вивчити різні аспекти в просуванні сайту. Google Analytics є ефективним інструментом для бізнесу, завдяки його функціям можна покращити та оптимізувати різні аспекти сайту.

Також варто звернути увагу на різноманітні сервіси, такі як Hotjar. Цей сервіс надає змогу переглядати поведінку користувачів на сайті.

Важливою функцією Hotjar є надання теплових карт, записів сеансів та опитувань на сайті. Це допомагає отримати детальну інформацію про поведінку відвідувачів і виявити можливі покращення, враховуючи надані дані.

Щодо сервісу SEMrush, його одним з найважливіших плюсів є можливість відстежування позицій в пошукових системах.

Сервіс також надає змогу аналізувати конкурентів і досліджувати ключові слова, що дозволяє покращити SEO просування в мережі. Використання SEMrush допомагає зрозуміти, як саме сайт виконується в пошукових системах.

Важливо пам'ятати, що різноманітні сервіси не заміняють повноцінного фахового знання провідних спеціаліст.

Нижче наведена таблиця 1.1 з порівнянням цих систем за деякими характеристиками:

Таблиця 1.1 – Порівняння сервісів

Характеристика	Google Analytics	Hotjar	SEMrush
Тип системи	Веб-аналітика	Теплові карти	SEO/SEM
Функціональність	Широкий спектр	Візуалізація	Ключові слова
Збір даних	Веб-сторінки	Поведінка	Конкуренти
Аналіз даних	Відвідувачі	Взаємодія	Трафік
Зручність використання	Висока	Середня	Висока
Інтеграція з іншими системами	Широкі можливості	Обмежена	Обмежена

1.5 Постановка задачі

Для виконання даної роботи, потрібно проаналізувати та розробити інформаційну систему онлайн готелю для тварин. Головною метою є розробка зрозумілого та зручного веб-сайту для користувачів.

Практична реалізація проєкту буде складатися з двох частин:

Розробка бази даних:

- Збір інформації для бронювання місць, включаючи дані про клієнтів, наявність кімнат, графік резервування та іншу необхідну інформацію.
- Створення таблиць та відношень для зберігання цих даних.

Розробка фронтенду (клієнтської частини) та бекенду (серверної частини):

- Фронтенд буде відповідати за відображення інтерфейсу користувача та взаємодію з ним. Він має забезпечити зрозумілий та привабливий дизайн, реалізувати функціональність для бронювання місць та відображення інформації про доступність готелю.
- Бекенд відповідатиме за обробку запитів від фронтенду, взаємодію з базою даних та бізнес-логіку додатку. Це включатиме розробку API для комунікації з фронтендом, обробку запитів користувачів, перевірку наявності місць для бронювання, зберігання та оновлення даних у базі даних та інші функції, необхідні для роботи системи [23].

У цьому проєкті використовуються декілька мов програмування для оптимізації та вирішення конкретних завдань. Основною мовою програмування буде JavaScript, яка забезпечує взаємодію з користувачем як на стороні клієнта, так і на стороні сервера. Крім того, для стилізації та оформлення веб-сторінок може бути використана мова CSS, а для розмітки структури сторінок - мова HTML.

Застосування JavaScript дозволить створити динамічні елементи інтерфейсу, валідувати дані, виконувати асинхронні запити до сервера та багато іншого. Крім того, JavaScript може використовуватись для здійснення валідації даних перед їх збереженням у базі даних та реалізації різних функцій, що покращують зручність використання сайту користувачами.

1.6 Дерево функцій

Вершина дерева функцій є головною метою роботи, а гілки дерева відображають функції або завдання, необхідні для досягнення цієї головної мети, а також підзавдання на нижчому рівні. Функціональна декомпозиція є методом декомпозиції, що базується на функціональних компонентах системи. Це означає, що кожен елемент системи розглядається як окрема одиниця, а також як складова частина більшої системи, де він виконує свою роль.

Дерево функцій, зображене на рисунку 1.2, демонструє ієрархічну структуру функцій і завдань, які потрібні для досягнення головної мети проєкту. Це допомагає визначити і розкрити всі необхідні функції та підзавдання, а також їх взаємозв'язки і взаємодію в межах системи. Дерево функцій є важливим інструментом для планування та управління проєктом, оскільки воно дозволяє краще організувати роботу, визначити пріоритети і забезпечити виконання всіх необхідних завдань для досягнення головної мети.

Це ієрархічне представлення функцій та завдань в дереві допомагає команді зрозуміти структуру проєкту, розподілити обов'язки та керувати виконанням завдань на різних рівнях. Крім того, це дозволяє зосередитися на конкретних функціях та їх виконанні, спрощує спілкування та співпрацю між учасниками проєкту.

Використання дерева функцій є ефективним інструментом для аналізу, планування та керування проєктами, оскільки воно дозволяє систематизувати

всі необхідні функції і завдання, забезпечити їх виконання та досягнення головної мети проєкту.

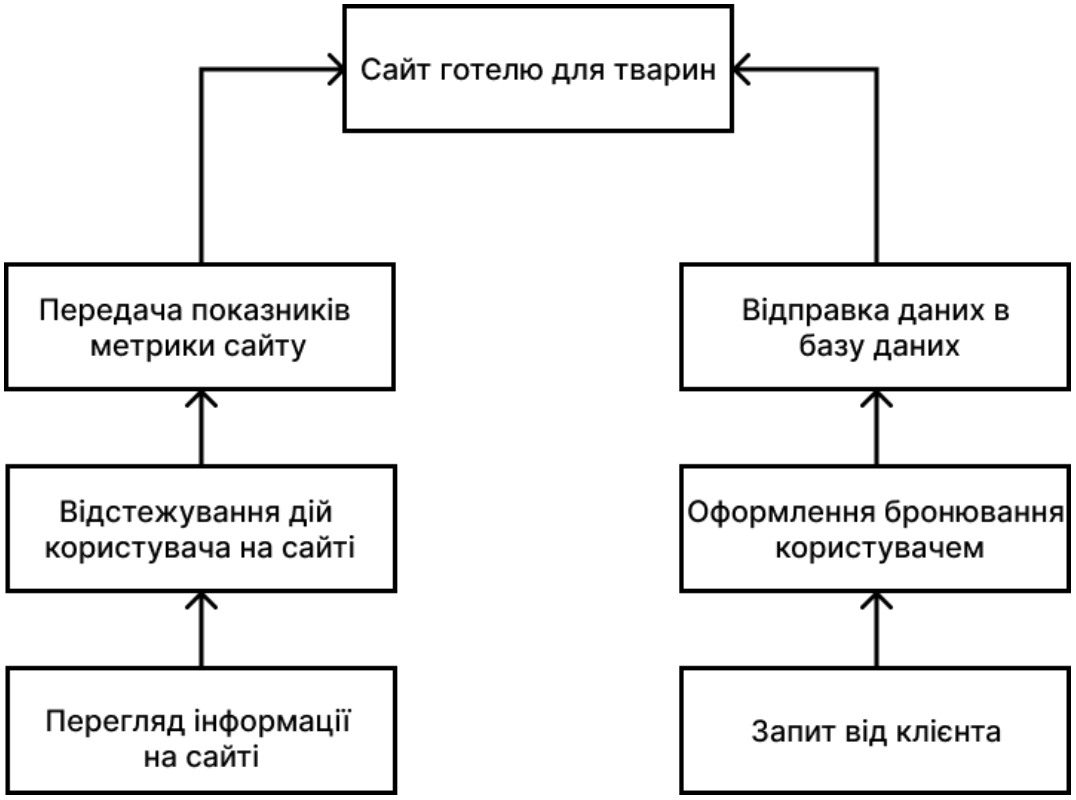


Рисунок 1.2 – Дерево функцій

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Технічна платформа

Для реалізації серверної частини, основною мовою програмування було обрано мову PHP[13]. Мова програмування PHP - є однією з найпопулярніших мов програмування для створення веб-додатків і має кілька значних переваг для використання в проєкті. Ось кілька переваг використання даної мови:

- PHP має простий і легко зрозумілий синтаксис, що робить його вивчення і розробку веб-додатків легкими. Ця мова має багато спільного з мовами, такими як C і Java, тому розробники з іншим досвідом можуть легко перейти до PHP.
- PHP має велику та активну спільноту розробників, що робить його особливо привабливим для програмістів. Це означає, що ви матимете доступ до багатьох ресурсів, документації, форумів та бібліотек, які допоможуть вам вирішити будь-які проблеми, з якими ви зіткнетесь під час розробки.
- PHP має велику розширюваність завдяки широкому спектру доступних розширень та бібліотек. Ці розширення і бібліотеки дозволяють розробникам з легкістю додавати нові функціональні можливості до своїх проєктів. Наприклад, вони можуть використовувати готові бібліотеки для роботи з базами даних, генерації PDF-файлів, маніпулювання зображеннями та багатьох інших операцій.
- У PHP є вбудована підтримка різних типів баз даних, включаючи MySQL, PostgreSQL, SQLite та інші. Це дозволяє розробникам зручно взаємодіяти з базами даних і виконувати різні операції, такі як вибірка, вставка, оновлення та видалення даних.
- Швидкодія: PHP часто використовується разом з веб-серверами Apache або Nginx для досягнення високої продуктивності.

Для зберігання інформації, в даному проєкті, було використано реляційну базу даних MySQL Server.

MySQL Server - є відкритою реляційною системою керування базами даних (РСКБД), яка надає надійні та ефективні засоби зберігання та керування даними[12]. Ось детальніше про деякі особливості та переваги MySQL:

- Надійність: MySQL володіє довірою спільноти розробників завдяки своїй стабільності та надійності. Вона використовує механізми транзакцій та журналування для забезпечення цілісності даних та відновлення після відмови. Багато великих організацій та веб-проєктів використовують MySQL у своїх продуктах через її високу надійність.
- Широка підтримка: MySQL підтримує багато операційних систем, таких як Windows, Linux, macOS, і FreeBSD. Вона також підтримує багато мов програмування, таких як PHP, Java, Python, Ruby і багато інших, що робить її універсальною для різних проєктів та стеків технологій.
- Висока продуктивність: MySQL відома своєю швидкодією та ефективністю. Вона має оптимізовані механізми запитів, індексування та кешування, що дозволяє швидко обробляти великі обсяги даних. Крім того, MySQL може бути налаштована для роботи на мультиядерних системах та оптимізована для використання пам'яті, що покращує продуктивність.
- Простота використання: MySQL має дружній та легкий у використанні інтерфейс. Вона пропонує різні інструменти управління базами даних, такі як командний рядок, графічний інтерфейс (наприклад, MySQL Workbench) та API для різних мов програмування.

Для того, щоб спросити ручну роботу з базою даних, було використано програму для доступу до бази даних MAMP 5.0.5.

MAMP (Mac, Apache, MySQL, PHP) - це програмне середовище, спеціально розроблене для розробки веб-додатків на платформі macOS та Windows [15].

MAMP поєднує в собі необхідні компоненти для розробки веб-додатків, такі як веб-сервер Apache, база даних MySQL та інтерпретатор PHP. Це забезпечує готове середовище для розробки і тестування додатків без необхідності окремого налаштування кожного компонента.

Програма надає зручний інтерфейс для управління веб-сервером Apache і сервером баз даних MySQL. Це дозволяє легко запускати, зупиняти та перезапускати сервери за допомогою простого інтерфейсу користувача.

MAMP дозволяє легко перемикатися між різними версіями PHP, що дозволяє розробникам тестувати свої додатки на різних версіях PHP без необхідності встановлення кожної версії окремо.

Перевагою є також можливість розробникам працювати локально на своєму комп'ютері без необхідності публікації коду на живому сервері. Це забезпечує зручність та безпеку розробки, оскільки можна випробувати та вдосконалити додаток до його публікації.

Сервер відправляє та отримує інформацію за допомогою технологій REST API (Representational State Transfer)[23]. REST API є архітектурним стилем, який дозволяє взаємодіяти з сервером шляхом передачі стандартних HTTP-запитів, таких як GET, POST, PUT та DELETE, для отримання або зміни ресурсів на сервері.

Одним з головних переваг REST API є його простота та легкість використання. Він використовує чітку структуру маршрутів (URL) та використовує HTTP-методи для визначення дій над ресурсами. Це робить його ідеальним вибором для розробки веб-додатків, оскільки REST API дозволяє зручну та ефективну комунікацію між клієнтом та сервером [3].

Крім того, REST API є незалежним від платформи та мови програмування, що означає, що можна використовувати різні технології на клієнтській та серверній сторонах. Це спрощує інтеграцію з різними

системами та дозволяє створювати гнучкі та масштабовані архітектури додатків.

REST API також підтримує кешування, що дозволяє зберігати копії ресурсів на клієнтській стороні та зменшує навантаження на сервер. Це сприяє покращенню продуктивності та зменшенню часу відповіді на запити.

Завдяки своїй популярності та широкій підтримці, REST API є стандартом для розробки веб-додатків та інтеграції сервісів, що дозволяє розробникам створювати сучасні та потужні додатки зі спрощеною комунікацією між компонентами.

Для розробки фронт-енд частини, було обрано мову програмування HTML, CSS, JavaScript.

HTML (HyperText Markup Language) - це стандартна мова розмітки для створення веб-сторінок та веб-документів. Вона використовується для структуризації та відображення вмісту в Інтернеті. HTML визначає структуру та семантику веб-сторінок, включаючи текстовий контент, зображення, посилання, таблиці, списки, форми та багато іншого.

HTML базується на тегах, які визначають різні елементи на веб-сторінці. Теги є розміткою, яка оточує контент і надає йому певний семантичний зміст.

Дана маова програмування також має багато інших тегів, які дозволяють додавати зображення, посилання, таблиці, форми, списки та багато іншого. Крім того, можна використовувати атрибути в тегах для надання додаткових властивостей елементам[6].

HTML є основою веб-розробки, і він зазвичай використовується разом з CSS і JavaScript для надання веб-сторінкам стильного вигляду та взаємодії з користувачем.

CSS (Cascading Style Sheets) - це мова стилізації, яка використовується для оформлення веб-сторінок та контролю їх зовнішнього вигляду. CSS дозволяє відокремити структуру веб-сторінки (визначену за допомогою

HTML) від її презентації, що дозволяє змінювати зовнішній вигляд сторінок без зміни їх структури [7].

Основна ідея CSS полягає у використанні набору правил, які встановлюють, які стилі та властивості повинні бути застосовані до елементів веб-сторінки. Кожне правило складається з селектора і блока оголошень. Селектор визначає, до яких елементів веб-сторінки будуть застосовуватись властивості, а блок оголошень містить список властивостей та їх значень.

JavaScript є однією з найпопулярніших мов програмування, яка використовується для створення динамічних інтерактивних веб-додатків.

JavaScript забезпечує можливість створювати багатий користувацький досвід шляхом додавання інтерактивності до веб-сторінок. Він дозволяє реагувати на події, взаємодіяти з користувачем, анімувати елементи та змінювати структуру сторінки без перезавантаження [21].

Дану мову програмування можна використовувати як на стороні клієнта (у браузері), так і на стороні сервера (за допомогою платформи, такої як Node.js). Це дозволяє розробникам створювати повноцінні веб-додатки, які працюють як на клієнтському, так і на серверному рівні.

Перевагою даної мови програмування є стандарт веб-розробки і підтримується всіма сучасними веб-браузерами. Це означає, що додатки, написані на JavaScript, будуть працювати на більшості пристроїв, які мають доступ до Інтернету.

JavaScript має велику кількість бібліотек, фреймворків та інструментів, що полегшують розробку веб-додатків. Наприклад, такі фреймворки, як React, Angular і Vue.js, дозволяють розробникам швидко створювати складні користувацькі інтерфейси та динамічні додатки.

Легка інтеграція, JavaScript може взаємодіяти з різними технологіями та сервісами. Він може спілкуватися з сервером за допомогою AJAX-запитів.

2.2 Проєктування бази даних

Як зазначено вище, база даних створюється з використанням MS SQL Server. Для забезпечення швидкої роботи бази даних необхідно виконати наступні кроки. Перш за все, необхідно належним чином налаштувати індекси для таблиць. Це допоможе зменшити кількість звернень до жорсткого диска та уникнути непотрібних операцій JOIN та інших важких операцій для системи.

Оптимізація індексів включає створення індексів на полях, які часто використовуються для пошуку або фільтрації даних. При цьому необхідно уникати надмірного використання індексів, оскільки це може призвести до збільшення розміру бази даних та зменшення швидкодії вставки та оновлення даних.

Для здійснення бронювання номеру для тварини важливо зберігати інформацію, необхідну для отримання особистих даних від користувача. Ця інформація включає, наприклад, ім'я та прізвище користувача, контактні дані, такі як номер телефону або адреса електронної пошти, а також інші відомості, що стосуються тварини, наприклад, порода, розмір, вік та особливі потреби.

Зберігання цих даних є важливим для ефективного оформлення бронювання та забезпечення комфортного перебування тварини під час її проживання. Крім того, наявність цих даних дозволяє персоналу готелю або спеціалістам з тваринного догляду надавати індивідуальний підхід до кожної тварини, враховуючи її потреби та вимоги.

З метою забезпечення конфіденційності та безпеки особистих даних користувачів, важливо використовувати надійні системи зберігання, які відповідають вимогам щодо захисту приватності. Компанії та установи повинні вживати заходів для запобігання несанкціонованому доступу до цих даних та забезпечення їх конфіденційності.

Дані, які будуть передаватися в базу даних:

- Особисті дані.
- Номер телефону.
- Дату та дні бронювання.
- Додаткова інформація.

Отримавши інформацію, можна приступити до розробки баз даних.

2.2.1 Створення концептуальної моделі

У процесі створення концептуальної моделі інформаційної системи онлайн готелю для тварин, спочатку необхідно ретельно проаналізувати ділові процеси, пов'язані з управлінням готельними послугами для тварин, такі як бронювання, реєстрація клієнтів, догляд за тваринами та облік платежів. Після аналізу ділових процесів визначаються вимоги до інформаційної системи, такі як зручне бронювання для власників тварин, відстеження наявності місць у готелі, зручний доступ до інформації про догляд та харчування для тварин, оплата послуг тощо.

Наступним кроком є визначення основних сутностей системи, таких як клієнти, готелі, тварини, бронювання, послуги. Потім встановлюються зв'язки між цими сутностями, наприклад, клієнт може здійснювати бронювання, готель може надавати різні послуги для тварин та інше.

На основі визначених сутностей та зв'язків розробляється схема бази даних, яка включає створення таблиць для збереження даних про клієнтів, готелі, тварин, бронювання, послуги, а також визначення відношень між ними.

При створенні концептуальної моделі важливо використовувати стандарти та методології, що допоможуть забезпечити її якість та зрозумілість.

Завершальним кроком є валідація та перевірка концептуальної моделі, яка включає аналіз моделі на відповідність вимогам, виявлення можливих

помилок та недоліків. Також може здійснюватися залучення фахових експертів для оцінки моделі.

Створення концептуальної моделі інформаційної системи онлайн готелю для тварин яку можна переглянути на рисунку 2.1. Є важливим етапом перед подальшою її розробкою. Вона допомагає визначити основні концепції та структуру системи, забезпечує ефективність та функціональність системи, а також покращує досвід користувачів.

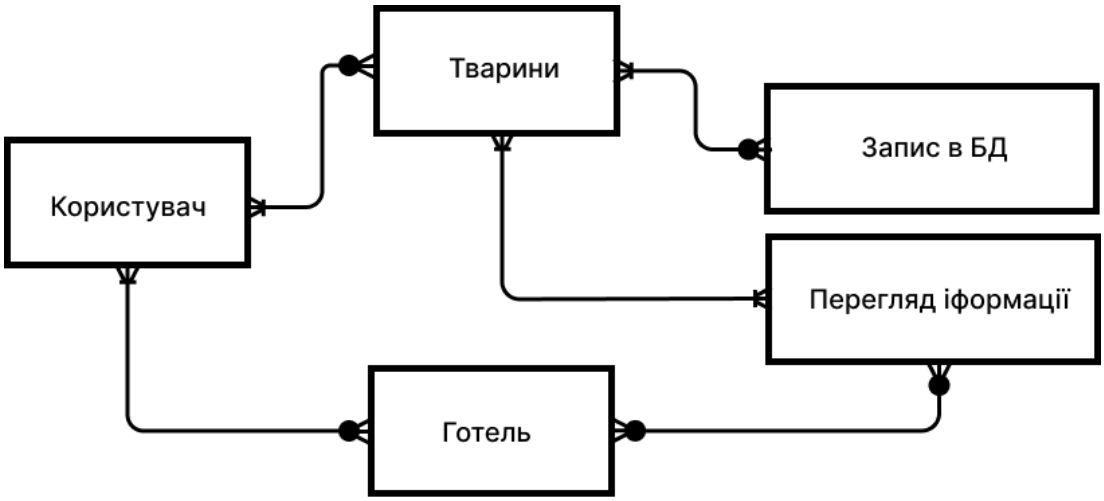


Рисунок 2.1 – Концептуальна модель опису зв’язків розроблювальної бази даних готелю для тварин

Крім того, важливо визначити ключові атрибути для кожного об'єкту. Наприклад, у випадку користувача це можуть бути атрибути, такі як унікальний ідентифікатор користувача, ім'я користувача, електронна пошта користувача, пароль користувача та рівень доступу користувача. Детальний опис кожного атрибуту можна знайти на рисунку 2.2.

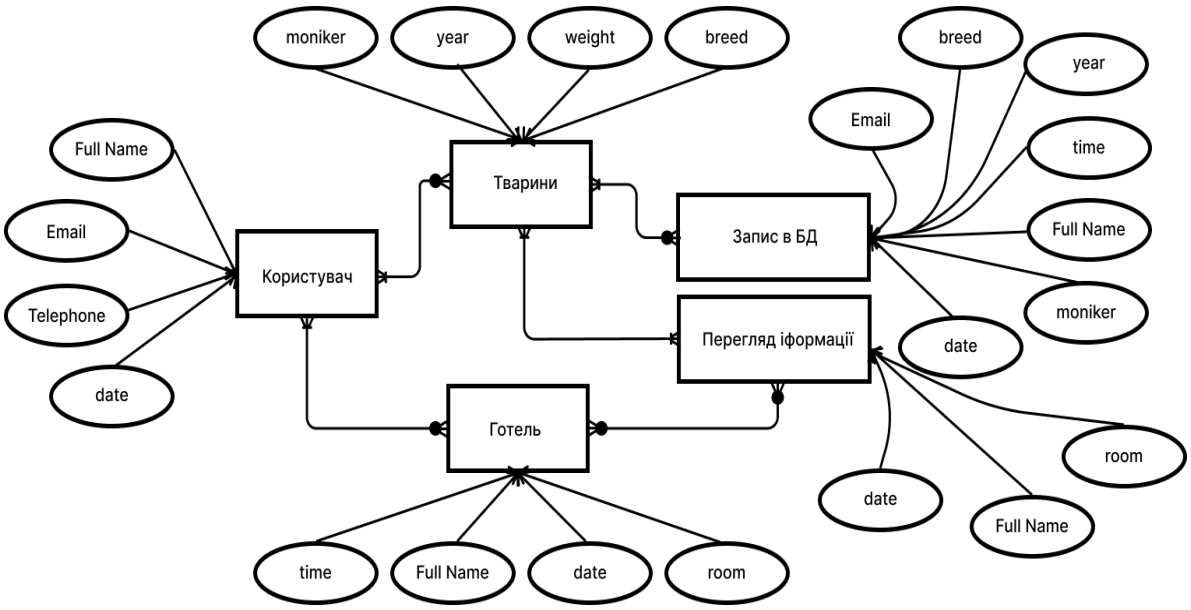


Рисунок 2.2 – Концептуальна модель атрибутів розроблювальної бази даних готелю для тварин

2.2.2 Створення логічної моделі інформаційної системи онлайн готелю для тварин

Створення логічної моделі інформаційної системи онлайн готелю для тварин є важливим етапом у розробці. Логічна модель визначає структуру та взаємозв'язки між сутностями системи, що допомагає уявити їх взаємодію.

На початку ідентифікуються основні сутності, які будуть присутні в системі онлайн готелю для тварин, такі як клієнти, готелі, тварини, бронювання, послуги тощо. Кожна сутність має свої атрибути, наприклад, для клієнта - ім'я, контактна інформація, адреса тощо.

Потім встановлюються зв'язки між сутностями. Наприклад, клієнт може здійснювати бронювання, готель надає різні послуги для тварин. Ці зв'язки можуть мати атрибути, такі як дата бронювання, тривалість перебування тощо.

На основі ідентифікованих сутностей та їх зв'язків створюється схема бази даних. Вона включає таблиці для збереження даних про клієнтів, готелі, тварин, бронювання, послуги та встановлює відношення між ними. Кожна таблиця має поля, що відповідають атрибутам сутностей, переглянути можна на рисунку 2.3.

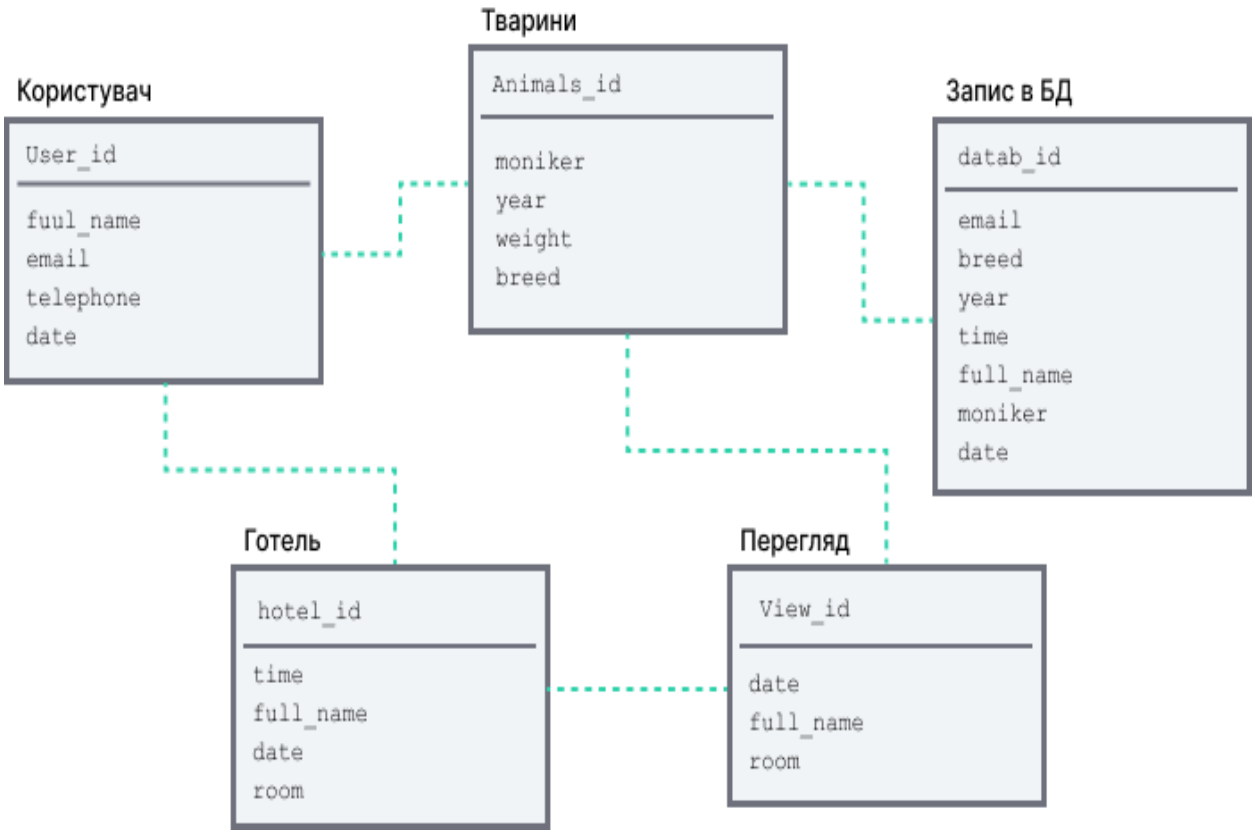


Рисунок 2.3 – Описана логічна модель з урахуванням взаємозв’язків.

Для досягнення якісної логічної моделі важливо дотримуватись стандартів та методологій, наприклад, нормалізації бази даних, що забезпечить її ефективність та якість.

У процесі створення логічної моделі слід враховувати вимоги до системи, забезпечуючи функціональність, безпеку та зручність використання. За потреби можна використовувати діаграми, наприклад, діаграми сутностей-зв'язків (ER-діаграми), для візуалізації моделі та полегшення розуміння її структури.

Валідація та перевірка логічної моделі є важливим етапом. Вона включає аналіз моделі на відповідність вимогам, виявлення можливих помилок та недоліків, а також залучення фахівців для оцінки моделі.

Створення логічної моделі інформаційної системи онлайн готелю для тварин допомагає уявити якісну структуру системи та визначити

взаємозв'язки між її компонентами. Це дозволяє ефективно переходити до фізичної реалізації системи та розробки програмного забезпечення.

2.2.3 Створення фізичної моделі інформаційної системи онлайн готелю для тварин

Створення фізичної моделі інформаційної системи онлайн готелю для тварин є одним з важливих етапів дипломної роботи. Фізична модель передбачає конкретну реалізацію системи, включаючи створення бази даних, програмного забезпечення та інфраструктури.

У процесі розробки фізичної моделі важливо враховувати ефективність та масштабованість системи, щоб забезпечити її оптимальну роботу навіть при великому навантаженні. Також, необхідно забезпечити захист даних і системи від несанкціонованого доступу та збоїв.

Фізичну модель бази даних, можна переглянути на рисунку 2.4.

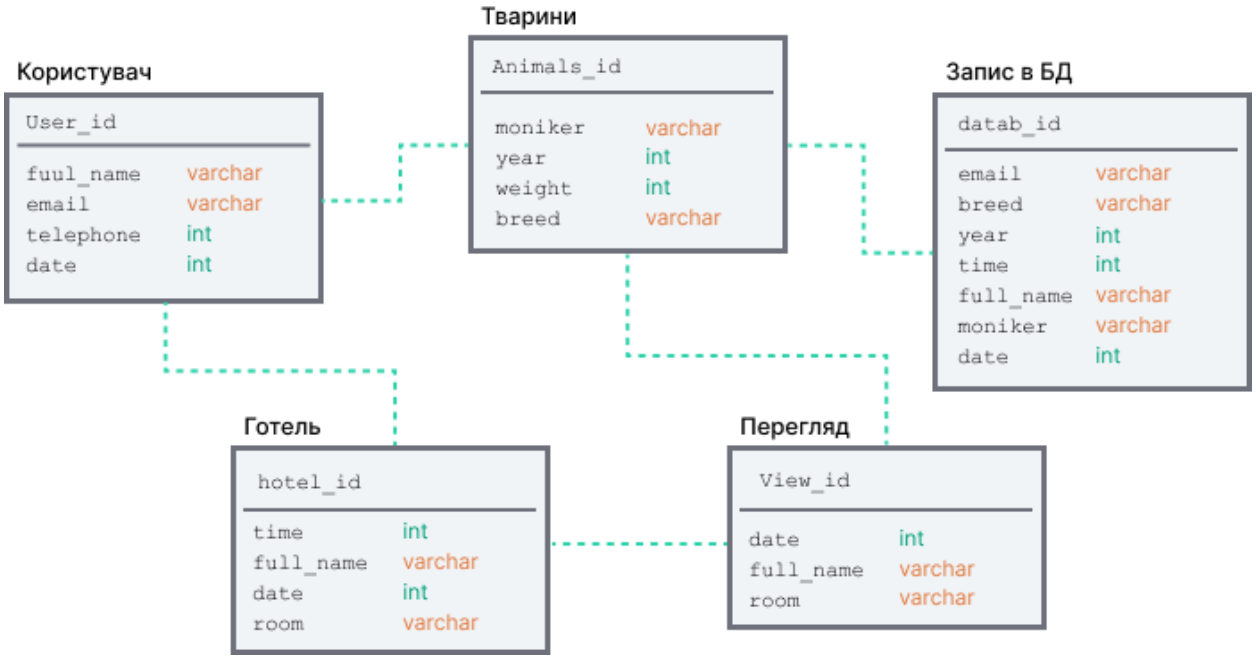


Рисунок 2.4 – фізична модель база даних

Опис фізичної моделі інформаційної системи онлайн готелю для тварин допомагає зрозуміти, як система буде функціонувати на практиці та які компоненти потрібно реалізувати. Це важливий крок у розробці системи та дозволяє перейти до наступних етапів, таких як реалізація програмного забезпечення та тестування системи.

2.3 Проєктування серверної частини

Для серверної частини було створена таблиця з назвою "tabledb", в якій зберігаються вхідні дані від користувачів. Для підключення до бази даних потрібно встановити параметри підключення, такі як сервер, користувач, пароль і назва бази даних [19].

Після встановлення з'єднання з базою даних за допомогою функції mysqli_connect() або конструктора new mysqli(), необхідно перевірити, чи вдалося успішно підключитися до бази даних. У разі невдалого з'єднання слід вивести повідомлення про помилку та припинити виконання скрипту.

Для вставки даних в таблицю необхідно створити SQL-запит, який складається з команди INSERT INTO, назви таблиці, переліку стовпців, до яких потрібно вставити дані, та значень, які треба вставити.

Після виконання запиту потрібно перевірити, чи він виконався успішно. У разі успішної вставки даних слід вивести повідомлення про успішну операцію. У випадку виникнення помилки під час виконання запиту слід вивести повідомлення про помилку та відобразити деталі помилки за допомогою методу error об'єкта з'єднання.

Після завершення роботи з базою даних важливо закрити з'єднання за допомогою методу close() об'єкта з'єднання.

Це загальний алгоритм для запису даних в таблицю MySQL з використанням PHP. Приклад запиту для запису в таблицю бази даних можна переглянути на рисунку 2.5.

```php
$name = filter_var(trim($_POST['name']),
FILTER_SANITIZE_STRING);
$telephone = filter_var(trim($_POST['telephone']),
FILTER_SANITIZE_STRING);
$date = filter_var(trim($_POST['date']),
FILTER_SANITIZE_STRING);
$textinf = filter_var(trim($_POST['textinf']),
FILTER_SANITIZE_STRING);
```

Рисунок 2.5 – Приклад запиту даних.

2.4 Перевірка алгоритму

При реєстрації алгоритм перевіряє введені дані, забезпечуючи їх достовірність та цілісність. Він перевіряє формат електронної пошти, пароль та інші важливі поля, щоб забезпечити відповідність вимогам безпеки та належності.

Алгоритм також враховує специфічні вимоги готелів для тварин, щоб забезпечити правильну інтеграцію даних. Він перевіряє, чи надана достатня інформація про тварину, зокрема вид, розмір, медичні вимоги та особливі потреби. Це допомагає нам забезпечити комфорт і безпеку для кожної тварини, яка перебуває в нашому готелі.

Крім того, алгоритм забезпечує швидку та ефективну обробку даних, що дозволяє нашим користувачам швидко залишати заявку на бронювання на нашому сайті. Ми прагнемо забезпечити зручний та безпроблемний досвід користувачів, гарантуючи, що процес реєстрації пройде гладко та без помилок.

Алгоритм, описаний у тексті, зображений на рисунку 2.6.

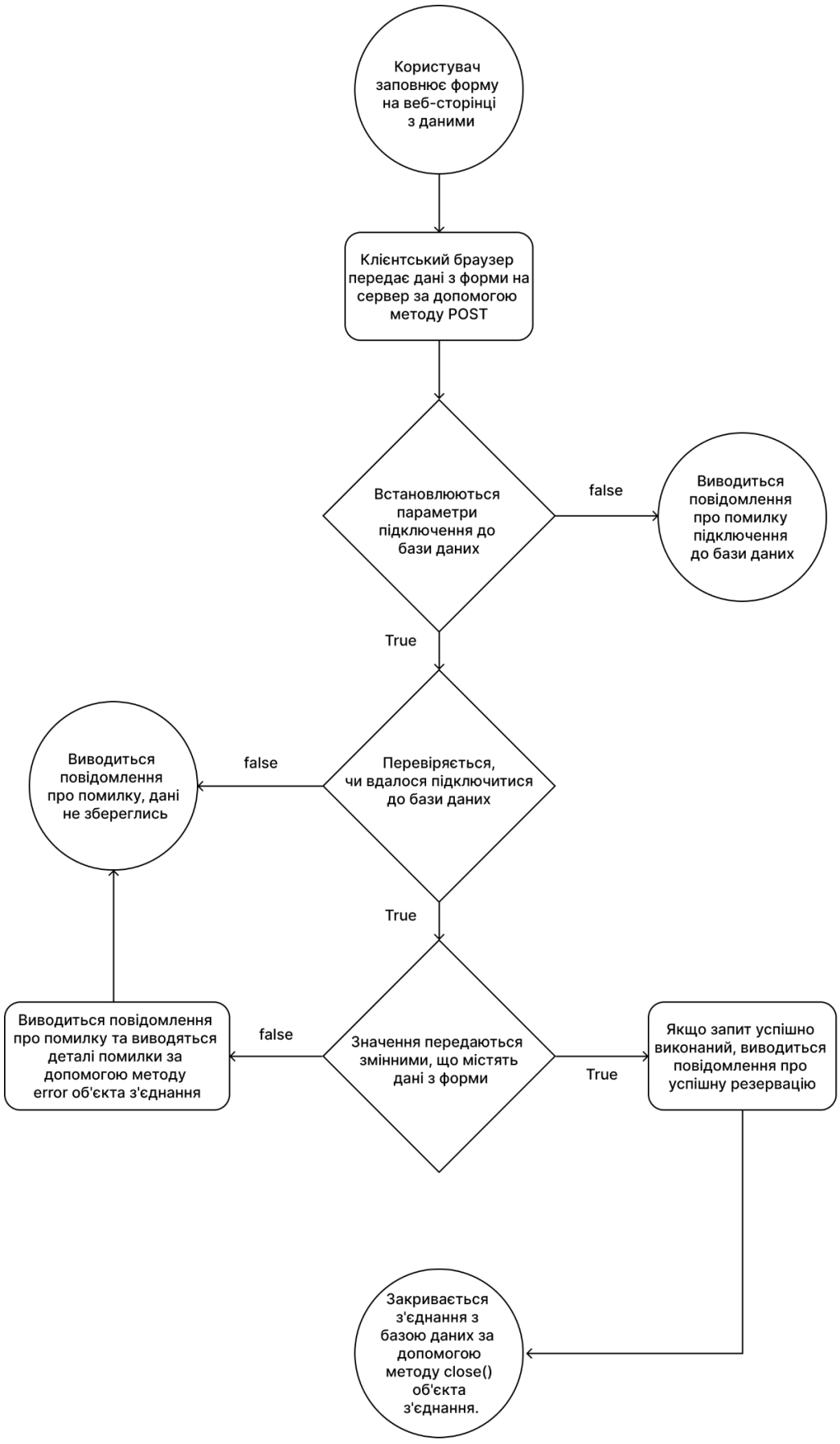


Рисунок 2.6 – Алгоритм реєстрації користувача.

Алгоритм перевірки запису особистої інформації в базу даних зображений на рисунку 2.7.

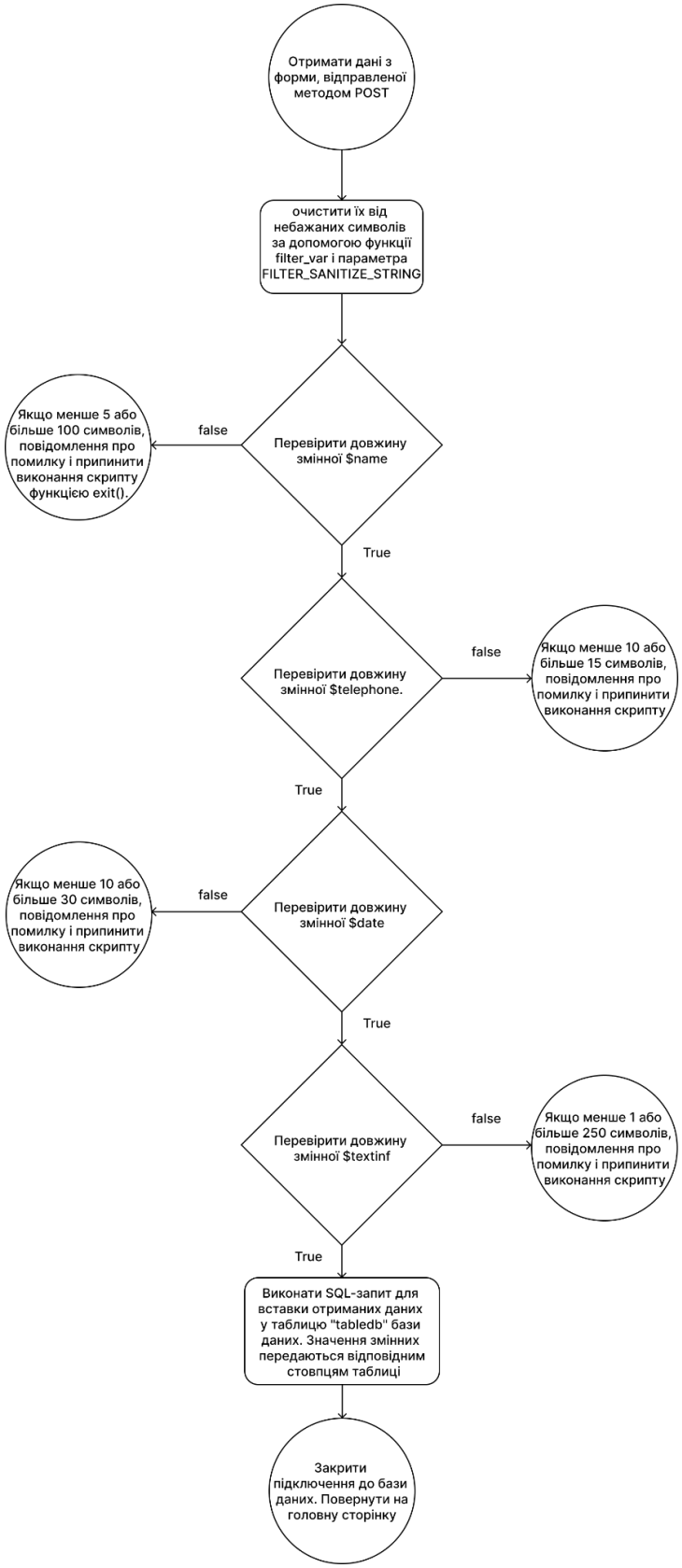


Рисунок 2.7 – Алгоритм запису даних.

2.5 Розробка шаблону

Для розробки зовнішнього вигляду сайту інформаційної системи онлайн готелю для тварин було обрано програму Figma. Figma є потужним інструментом, який дозволяє дизайнерам і розробникам спільно працювати над проєктом, втілюючи їхні ідеї в ефективні та привабливі інтерфейси.

Однією з переваг Figma є його зручний та інтуїтивно зрозумілий інтерфейс, що сприяє швидкому освоєнню програми. Користувачі можуть легко створювати, редагувати та оновлювати дизайн елементів інтерфейсу без необхідності переключатися між різними програмами або платформами.

Figma також пропонує широкі можливості для створення векторних графічних об'єктів та макетів, що сприяє реалізації творчих ідей дизайнерів. Крім того, програма підтримує додавання інтерактивності до прототипів, дозволяючи перевіряти функціональність та навігацію в процесі розробки.

Однією з головних переваг Figma є можливість спільної роботи та взаємодії між командами. Користувачі можуть запрошувати колег та клієнтів до проєкту, дозволяючи їм переглядати, редагувати та коментувати дизайн. Це забезпечує зручну комунікацію, спрощує процес затвердження змін та сприяє швидкій і ефективній роботі над проєктом.

Крім того, Figma забезпечує можливість експорту проєкту в різні формати, що спрощує подальшу інтеграцію з іншими інструментами розробки, такими як розробка веб-сайтів або мобільних додатків.

Загалом, використання програми Figma для розробки зовнішнього вигляду сайту інформаційної системи онлайн готелю для тварин є вигідним рішенням, оскільки вона надає потужні інструменти для творчого дизайну, спільної роботи та прототипування, спрощуючи процес розробки та підвищуючи якість результуючого інтерфейсу.

Порівняння Figma з конкурентами може бути залежне від індивідуальних потреб та вимог користувача. Однак, ось кілька основних відмінностей між Figma та деякими популярними конкурентами:

- Sketch: Sketch є нативною програмою для macOS, тоді як Figma працює у веб-браузері та підтримує різні операційні системи. Figma надає більш зручну спільну роботу та співпрацю, оскільки вона не вимагає спеціального програмного забезпечення та дозволяє працювати з різних платформ.
- Adobe XD: Якщо порівняти з Adobe XD, Figma має перевагу у спільній роботі та співпраці. Figma дозволяє більш зручно спільно редагувати та коментувати проєкти в реальному часі, що є важливим для колаборативної роботи. Крім того, Figma працює у веб-браузері, тоді як Adobe XD потребує встановлення спеціального програмного забезпечення.
- InVision Studio: InVision Studio також пропонує багато функцій для дизайну та прототипування, але його основна перевага полягає у високій якості прототипування та анімації. У той же час, Figma має більш потужні інструменти для спільної роботи та взаємодії між командами.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Створення шаблону в Figma.

Початок створення шаблону готелю для тварин у Figma можна розпочати зі створення загальної структури та компонентів шаблону. Після розробки та узгодження UX частини [25]. Почався процес створення UI дизайну та основних компонентів головної сторінки: заголовка або шапки сторінки, яка буде містити логотип готелю для тварин, назву та можливно додатковий елемент дизайну, який буде служити ідентифікатором готелю [26].

Наступним кроком може бути розміщення навігаційного меню або панелі зверху, де відвідувачі зможуть швидко знайти потрібну інформацію. Меню містить посилання на основні розділи, такі як "Головна", "Послуги", "Інформація","Галерея", "Контакти".

Результат даного блоку, можна переглянути на рисунку 3.1.

Рисунок 3.1 – Навігаційна панель.

Після навігаційної панелі, почався процес з додаванням блоків контенту на головну сторінку. Було додано, велике фото, що центральним елементом головної сторінки, що привертає увагу відвідувачів і створює позитивне перше враження.

Під основним фото розташовується невеликі блоки з коротким описом переваг готелю для тварин та актуальною інформацією. Також було додані компоненти кнопок, для оформлення заявки та зворотнього зв'язку.

Потрібно зазначити, що шрифт в проєкті відіграє важливу складову частину, зважаючи на концепцію та сучасні тренди дизайну в Україні, було обрано шрифт e-Ukraine.

Кожен окремий блок на шаблоні, було створено за стандартами сучаних тенденцій веб-дизайну. Було використано Auto Layout. Дана функція - це потужний інструмент, який дозволяє автоматично організовувати та змінювати розташування елементів у вашому дизайні.

Завдяки автолеяуту можна створити гнучкі компоненти, які адаптуються до різних розмірів та контенту. Головну сторінку можна переглянути на рисунку 3.2.

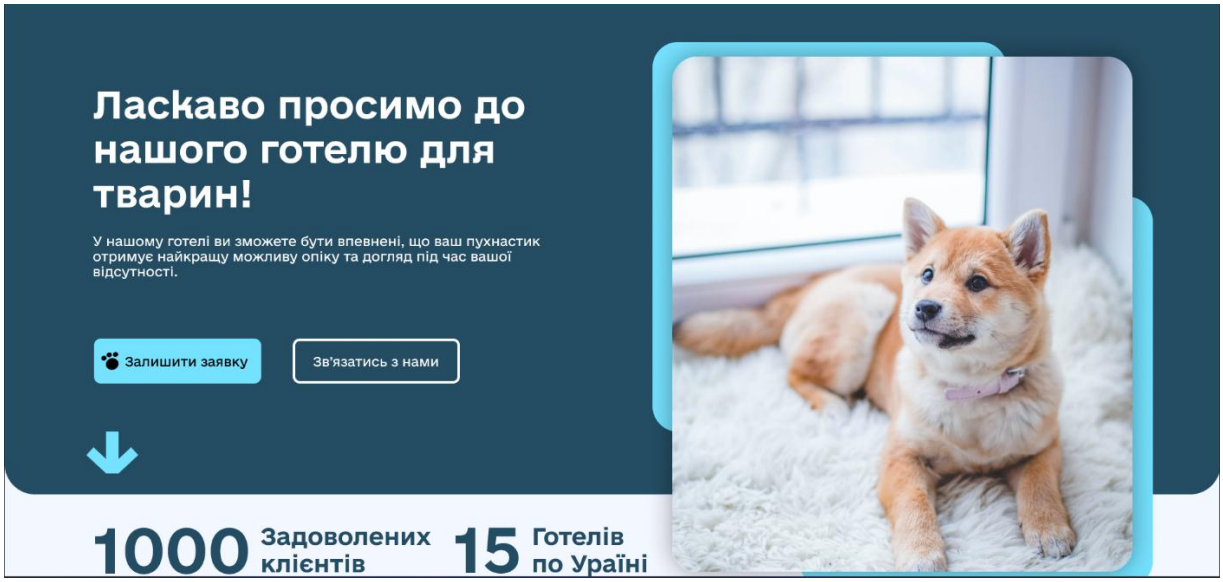


Рисунок 3.2 – Головна сторінка сайту.

Важливим елементом шаблону будуть блоки з інформацією про послуги та номери готелю для тварин. Для цього було створені блоки кожного типу номеру, їх опис, ціни та можливі додаткові переваги, такі як доступ до вольєрів, послуги грумінгу або прогулянки з твариною.

В прикладі дизайну, не було зазначено остаточних цін та послуг, їх було додано вже в процесі розробки фронтенд частини. Переглянути дані пункти можна на рисунку 3.3.

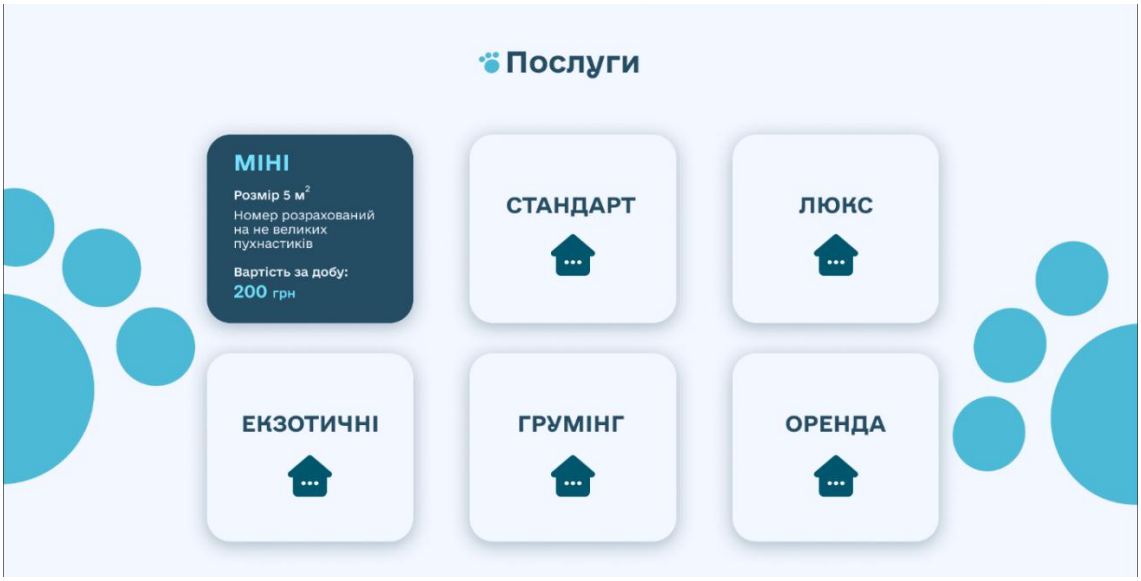


Рисунок 3.3 – Розділ послуг на сайті

Також, у рамках розробки шаблону готелю для тварин, було створено розділ з загальною інформацією про послуги та переваги нашого готелю, які виділяють його серед інших конкурентів. Цей розділ має велике значення для привертання уваги та зацікавлення відвідувачів сайту.

Щоб ознайомитися з даною інформацією, вам слід переглянути блок на рисунку 3.4. В цьому блоку будуть відображені основні послуги та переваги, які ми пропонуємо нашим клієнтам. Важливо викласти цю інформацію зрозуміло та привабливо, щоб вона була легко сприйнята відвідувачами сайту.

Рисунок 3.4 – Розділ переваг та загальної інформації.

Для наших користувачів є дуже важливим мати можливість ознайомитися з умовами та переглянути фотографії місць, де перебуватимуть їхні улюбленці. Ми розуміємо, що це допомагає забезпечити їм впевненість у якості та комфорті нашого готелю для тварин.

Тому, в рамках розробки шаблону, ми додали блок галереї, який дозволяє користувачам переглянути різноманітні фотографії самого готелю та інших тварин, які вже були у нашому догляді. Цей блок зображений на рисунку 3.5.

За допомогою слайдерів у галереї, користувачі можуть зручно перегортати фотографії та долучитися до віртуальної екскурсії по нашому готелю. Вони зможуть оцінити наші просторі та затишні приміщення, переглянути умови проживання та впевнитися, що ми надаємо комфортні умови для їхніх улюбленців.

Ми впевнені, що цей блок галереї допоможе нашим користувачам зробити інформований вибір та довіряти нам з перебуванням їхніх улюбленців.

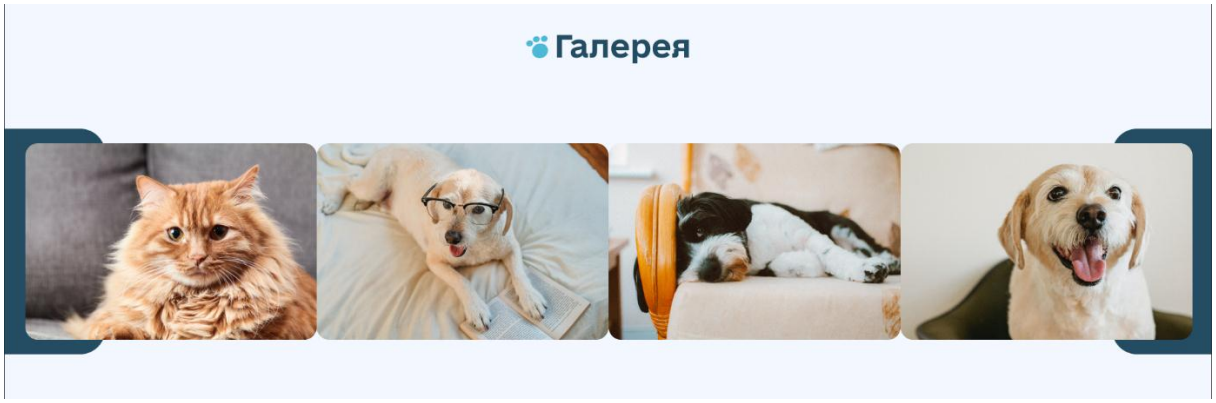


Рисунок 3.5 – Розділ галереї.

У даному проєкті обов'язковим елементом є форма реєстрації, яка дозволяє користувачам зареєструватися в нашому готелі для тварин. Ця форма має специфічні поля, що відповідають різним видам інформації, яку потрібно передати для обробки та зберігання в базі даних.

З метою ефективної передачі інформації нашій базі даних, ми створили декілька важливих полів вводу в цій формі реєстрації. Ці поля дозволяють користувачам ввести необхідні дані та забезпечити нам інформацію для подальшої обробки їхньої реєстрації. Форма реєстрації можна побачити на рисунку 3.6.

Серед полів вводу можуть бути такі дані, як ім'я та прізвище власника тварини, контактна інформація (телефон, електронна пошта), вид та ім'я тварини, медична інформація (наприклад, наявність алергій або хронічних захворювань). Ці поля дозволяють нам збирати необхідну інформацію для забезпечення комфорту та догляду за кожною твариною.

Створена нами форма реєстрації є ключовим елементом, який допоможе нам забезпечити належний рівень обслуговування та зручність для наших клієнтів.

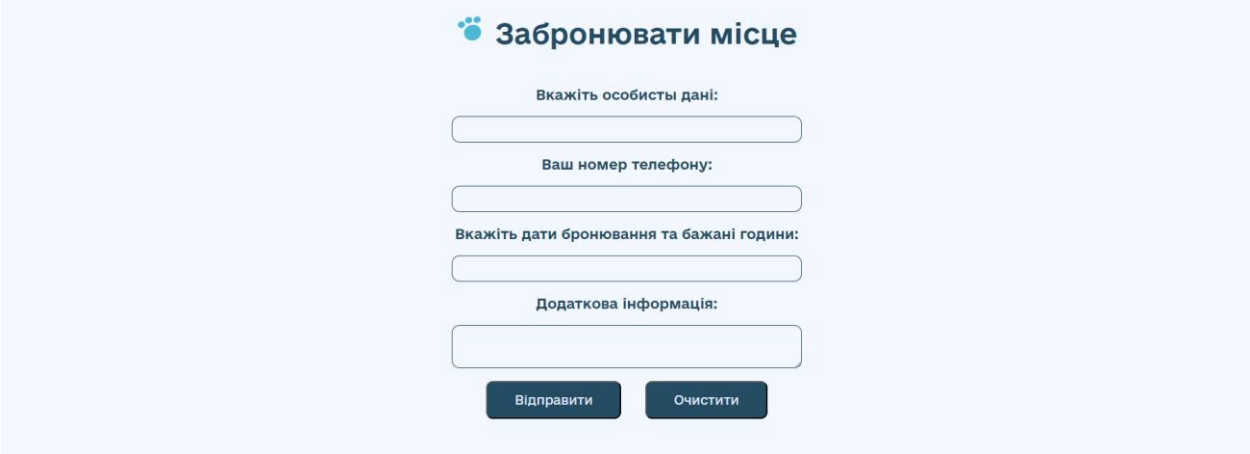


Рисунок 3.6 – Форма реєстрації.

Блок контактів на сайті готелю для тварин є невід'ємною та важливою частиною, оскільки він забезпечує можливість зв'язку з потенційними клієнтами. Цей блок надає необхідну контактну інформацію, що дозволяє людям звертатися до готелю з будь-якими питаннями або потребами.

В рамках проєкту, вся ключова інформація була розміщена в футері, що знаходиться у нижній частині сторінки. У цьому розділі користувач може швидко знайти контактні дані, такі як телефон, електронна пошта або адреса

готелю, щоб зв'язатися з нами. Крім того, в футері можуть бути додаткові пункти, які допомагають клієнтам ознайомитися з відгуками і враженнями інших клієнтів, а також прочитати політику конфіденційності.

Надання цієї інформації в футері спрощує процес зв'язку між клієнтами та готелем, дозволяючи їм швидко знайти потрібну інформацію та вирішити свої питання чи уточнити деталі. Крім того, доступ до відгуків та політики конфіденційності надає клієнтам додаткову впевненість у співпраці з нашим готелем для тварин. Даних блок можна переглянути на рисунку 3.7.

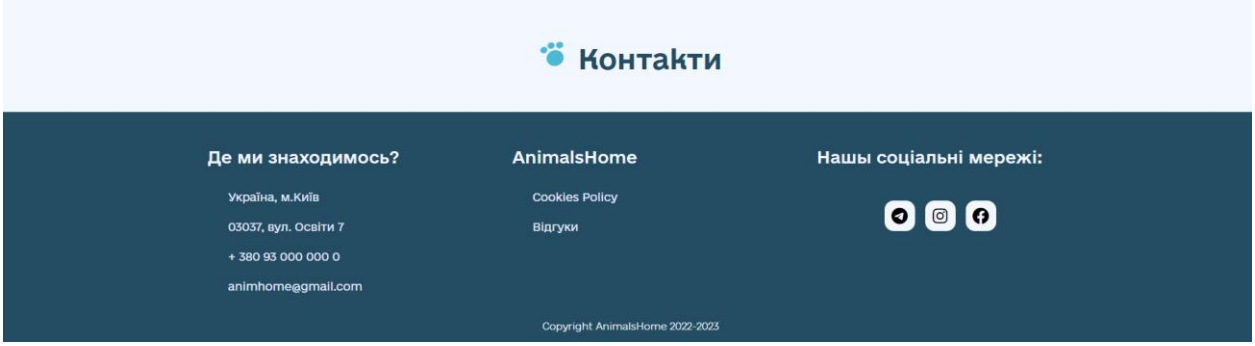


Рисунок 3.7 – Розділ контактної інформації.

3.2 Розробка фронтенд частини

При розробці першого блоку сайту, було використано структуру та вигляд заголовку (header) веб-сторінки. Основні елементи та їх функціональність включають:

`<header class="header" id="header">`: Визначає початок заголовку сторінки і надає йому клас "header" та ідентифікатор "header".

`<div class="container">`: Створює контейнер, який обмежує область заголовку сторінки.

`<div class="header__iner">`: Створює внутрішній контейнер для розміщення елементів заголовку.

`<div class="header__logo">`: Відображає логотип готелю. Зображення логотипу знаходиться у файлі "img/logo.png".

`<nav class="nav" id="nav">`: Створює навігаційне меню (навбар) з посиланнями на різні розділи сторінки.

`<a class="nav__link" href="#intro">Головна</a>`: Кожне посилання (<a>) в навігаційному меню має клас "nav__link" та вказує на відповідний розділ сторінки через атрибут href.

`<div class="header__telephone">`: Створює контейнер для відображення телефонного номера готелю.

`<h2 class="telephone">+380 93 327 327 7</h2>`: Відображає телефонний номер у заголовку, використовуючи заголовок другого рівня <h2> та клас "telephone".

Даний відрізок коду, можна переглянути на рисунку 3.8.

```
<body>
    <header class="header" id="header">
        <div class="container">
            <div class="header__iner">
                <div class="header__logo">
                    <img src="img/logo.png" alt="">
                </div>
                </button>
                <nav class="nav" id="nav">
                    <a class="nav__link" href="#intro">Головна</a>
                    <a class="nav__link" href="#bright">Послуги</a>
                    <a class="nav__link" href="#information">Інформація</a>
                    <a class="nav__link" href="#gal">Галерея</a>
                    <a class="nav__link" href="#cont">Контакти</a>
                </nav>
                <div class="header__telephone">
                    <h2 class="telephone">+380 93 327 327 7</h2>
                </div>
            </div>
        </div>
    </header>
```

Рисунок 3.8 – Вигляд коду першого блоку.

Для побудови зовнішнього вигляду сайту, було застосовано CSS стилі. Мета підключення CSS до HTML полягає в тому, щоб стилізувати та оформити веб-сторінку з використанням каскадних таблиць стилів (CSS).

CSS дозволяє змінювати вигляд та розташування елементів HTML, задавати кольори, шрифти, розміри, відступи, фонові зображення та інші атрибути для створення зовнішнього вигляду сторінки.

У проєкті було використано основні компоненти CSS, які можна оглянути на рисунку 3.9. Одним з важливих аспектів цього проєкту було підключення шрифту e-Ukraine, який, на відміну від шрифтів з бібліотеки Google Fonts, не був доступний напряму. Щоб мати можливість використовувати цей шрифт, необхідно було завантажити його на пристрій та правильно підключити до проєкту.

З метою підключення шрифту e-Ukraine було використано наступні компоненти CSS.

```css
@font-face{
    font-family: 'e-Ukraine Head';
    src: url('..fonts/pfdintextcondpro-regular.eot');
    src: local(''),url('..fonts/pfdintextcondpro-regular.woff')format('woff'), url('..fonts/pfdintextcondpro-regular.ttf')
        format('truetype'), url('..fonts/pfdintextcondpro-regular.svg') format('svg');
    font-weight: normal;
    font-style: normal;
}

@font-face{
    font-family: 'e-Ukraine';
    src: url('..fonts/pfdintextcondpro-regular.eot');
    src: local(''),url('..fonts/pfdintextcondpro-regular.woff')format('woff'), url('..fonts/pfdintextcondpro-regular.ttf')
        format('truetype'), url('..fonts/pfdintextcondpro-regular.svg') format('svg');
    font-weight: normal;
    font-style: normal;
}

@font-face{
    font-family: 'e-Ukraine';
    src: url('..fonts/pfdintextcondpro-regular.eot');
    src: local(''),url('..fonts/pfdintextcondpro-regular.woff')format('woff'), url('..fonts/pfdintextcondpro-regular.ttf')
        format('truetype'), url('..fonts/pfdintextcondpro-regular.svg') format('svg');
    font-weight: 500;
    font-style: normal;
}

@font-face{
    font-family: 'e-Ukraine';
    src: url('..fonts/pfdintextcondpro-regular.eot');
    src: local(''),url('..fonts/pfdintextcondpro-regular.woff')format('woff'), url('..fonts/pfdintextcondpro-regular.ttf')
        format('truetype'), url('..fonts/pfdintextcondpro-regular.svg') format('svg');
    font-weight: 700;
    font-style: normal;
}
@font-face{
    font-family: 'e-Ukraine';
    src: url('..fonts/pfdintextcondpro-regular.eot');
    src: local(''),url('..fonts/pfdintextcondpro-regular.woff')format('woff'), url('..fonts/pfdintextcondpro-regular.ttf')
        format('truetype'), url('..fonts/pfdintextcondpro-regular.svg') format('svg');
    font-weight: 300;
    font-style: normal;
}
```

Рисунок 3.9 – Підключення шрифту.

Складність даного коду полягає в необхідності прописувати окремий код для кожного розміру та товщини шрифта. Це вимагає використання різних файлів шрифтів, які зберігаються в окремій папці.

Кожен файл шрифту має власний шлях, який потрібно вказати в коді для його правильного підключення. Така організація стилів шрифтів може зробити код складним для підтримки та змін.

Кожен блок має стандартизовані розміри, що дозволяє легко копіювати та вставляти схожі стилі без зайвих зусиль. Це особливо помітно в блоці з послугами готелю зображений на рисунку 3.10, де використовуються однакові розміри для кожного елемента.

Такий підхід спрощує процес створення та зміни стилів, оскільки можна просто скопіювати та застосувати вже визначені розміри до нових елементів. Це забезпечує єдність та консистентність дизайну, що полегшує розробку та підтримку проєкту.

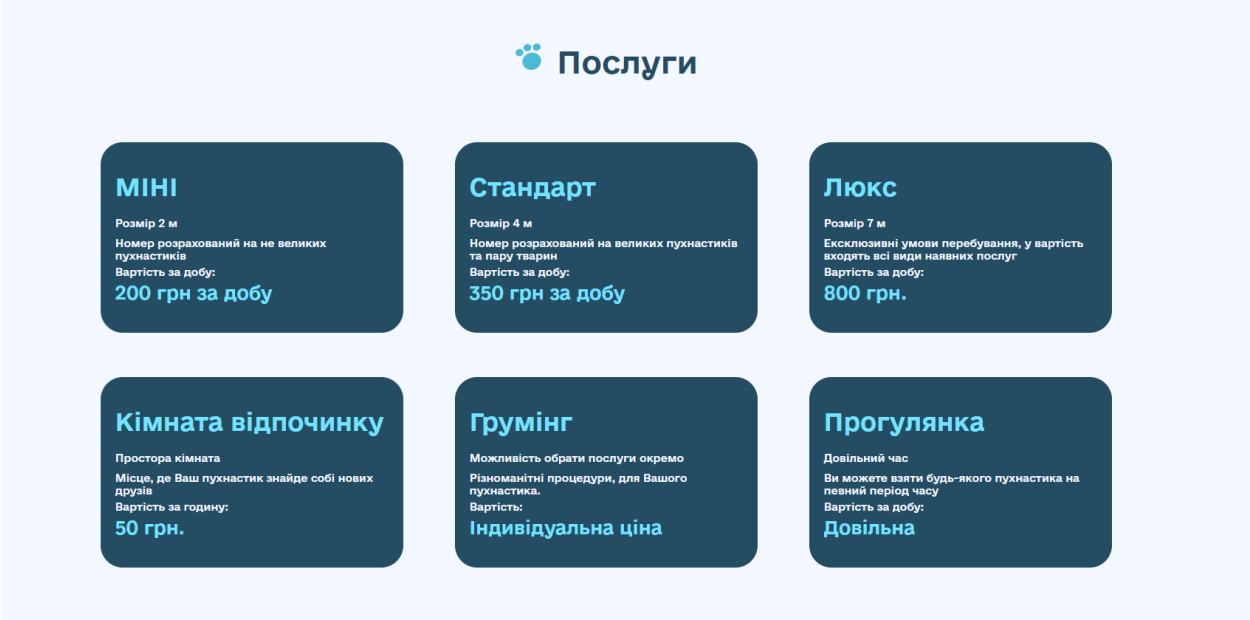


Рисунок 3.10 – Функціонал блоків.

Завдяки цьому підходу, ми отримуємо можливість легко створювати багато подібних блоків, які будуть розташовані впродовж один одного, не перекриваючись при цьому.

Це спрощує процес розміщення елементів на веб-сторінці та дозволяє забезпечити їх зручне відображення для користувачів. Вигляд коду, який ілюструє цей підхід, можна побачити на рисунку 3.11.

```
<div class="container">
    <div class="block__services__text" id="bright">
        <img src="img/lpblw.png" alt="">
        Послуги
    </div>
    <div class="block__services">
        <div class="block__card">
            <div class="movie__card__block">
                <div class="info__section__block">
                    <div class="content__block">
                        <h1>МІНІ</h1>
                    </div>
                    <div class="text__block__info">
                        Розмір 2 м
                    </div>
                    <div class="text__info__cost">
                        Номер розрахований на не великих пухнастиків
                    </div>
                    <div class="text__info__cost">
                        Вартість за добу:
                    </div>
                    <div class="text__cost">
                        200 грн за добу
                    </div>
                </div>
            </div>
        </div>
        <div class="block__card">
            <div class="movie__card__block">
                <div class="info__section__block">
                    <div class="content__block">
                        <h1>Стандарт</h1>
                    </div>
                    <div class="text__block__info">
                        Розмір 4 м
                    </div>
                    <div class="text__info__cost">
                        Номер розрахований на великих пухнастиків та пару тварин
                    </div>
                    <div class="text__info__cost">
                        Вартість за добу:
                    </div>
                    <div class="text__cost">
```

Рисунок 3.11 – Вигляд блоку Послуги в HTML

Також варто зазначити, що стилі, застосовані до CSS блоків, є однаковими. Це означає, що їх можна застосувати до будь-якого блоку з аналогічною структурою і отримати однаковий зовнішній вигляд.

Ця уніфікація спрощує розробку та забезпечує єдинообразний стиль для всіх блоків, що покращує загальний вигляд веб-сторінки. Деталі цих стилів можна переглянути на рисунку 3.12.

```css
.block__services{
    display: flex;
    background: url("1bg.png");
    flex-wrap: wrap;
    justify-content: center;
}
.block__card{
    cursor: grab;
}
.movie__card__block{
    cursor: grab;
    position: relative;
    display: block;
    max-width: 995px;
    margin: 5px;
    border-radius: 10px;
    transition: all 0.4s;
}
.movie__card__block:hover{
    transform: scale(1.02);
    transition: all 0.4s;
}
.movie__card__block .my_card{
    display: block;
    position: relative;
    float: right;
    height: 370px;
    margin-left: 15px;
}

.info__section__block{
    position: relative;
    display: block;
    max-width: 370px;
    padding: 20px;
    margin: 30px ;
    overflow: hidden;
    border-radius: 30px;
    transition: all 0.4s;
    background-color: #244D63;
}

.block__card .movie__card__block{
    flex-basis: calc(995px / 3);
}
```

Рисунок 3.12 – Стилі блоку послуги в CSS.

Важливою складовою цього коду є те, що всі компоненти розміщені всередині контейнера з класом "container". Цей контейнер створює блокову

структуру, яка дозволяє групувати та керувати компонентами окремо від інших елементів сторінки.

Клас "container" визначає обмежений простір, в межах якого розташовуються всі елементи, забезпечуючи їх взаємне розташування та візуальну єдність. Це дозволяє зручно керувати розмірами, положенням та стилями компонентів всередині контейнера і спрощує управління макетом сторінки в цілому.

Використання контейнера "container" допомагає створювати рівномірний та зручний вигляд сторінки, а також полегшує розширення та зміну дизайну в майбутньому. Вигляд даного класу можна переглянути на рисунку 3.13.

```
.container{
    font-family: 'e-Ukraine';
    width: 100%;
    max-width: 1500px;
    margin: 0 auto;
}
```

Рисунок 3.13 – Вигляд класу container.

Властивості даного коду:

`font-family: 'e-Ukraine';` - Встановлює шрифт для тексту всередині елементів з класом "container". В даному випадку використовується шрифт "e-Ukraine".

`width: 100%;` - Задає ширину елемента з класом "container" на 100% ширини його батьківського елемента. Це означає, що контейнер буде розтягуватися на всю доступну ширину.

`max-width: 1500px;` - Встановлює максимальну ширину для елемента з класом "container". В даному випадку обмеження становить 1500 пікселів. Це означає, що незалежно від розміру вікна браузера, ширина контейнера не перевищить 1500 пікселів.

`margin: 0 auto;` - Встановлює зовнішні відступи для елемента з класом "container". Значення "0" встановлює нульові відступи по верхньому і нижньому краях, а "auto" автоматично вирівнює контейнер горизонтально відносно його батьківського елемента. Це дозволяє центрувати контейнер по горизонталі на сторінці.

Загалом, ці властивості допомагають створити блокову структуру з широким контейнером, що розтягується на всю доступну ширину, з максимальною шириною 1500 пікселів і центрованим відносно горизонталі розташуванням.

Для створення галереї було важливо правильно згрупувати фотографії, щоб вони могли змінювати своє масштабування та мати однаковий розмір. Для цього було використано слайдер зі спеціально налаштованими властивостями, які вказані на рисунку 3.14.

Цей слайдер дозволяє відображати фотографії у вікні галереї і надає зручні функції перегляду та перемикання між фотографіями. Застосовані властивості дозволяють контролювати розмір фотографій, їх масштабування та інші параметри, що забезпечують комфортне використання галереї для користувача.

```
.gallery{
    display: flex;
    height: 300px;
    margin: 50px 0px 50px 0px;
}
.gallery__image{
    min-width: 0;
    flex: 1 1 10px;
    border-radius: 20px;
    object-fit: cover;
    opacity: .9;
    transition: .5s;
}
.gallery__image:hover{
    flex: 1 1 340px;
    opacity: 1;
}
```

Рисунок 3.14 – Вигляд коду слайдера.

Також, у проєкті була розроблена форма, яка використовується для збереження інформації в базі даних. Для цього була використана HTML-властивість "form", яка може бути переглянута на рисунку 3.15.

Коректне написання цієї форми є надзвичайно важливим аспектом у розробці бекенду. Вказання правильних атрибутів та відповідних налаштувань дозволяє формі збирати дані від користувача та передавати їх до бази даних для подальшої обробки.

Використання вірних властивостей та належна настройка форми є вирішальними факторами для успішної передачі та обробки даних у бекенді.

```
</div>
<div class="container">
    <div class="block__services__text">
        <img src="img/lpblw.png" alt="">
        Забронювати місце
    </div>
    <div class="form__registr" id = "form">
        <form action="inc/inc.php" method="post">
            <div>
                <p>Вкажіть особисты дані:</p>
                <input tabindex="1" name="name" type="text" />
            </div>
            <div>
                <p>Ваш номер телефону:</p>
                <input tabindex="2" name="telephone" type="text" />
            </div>
            <div>
                <p>Вкажіть дати бронювання та бажані години:</p>
                <input tabindex="3" name="date" type="text" />
            </div>
            <div>
                <p>Додаткова інформація:</p>
                <textarea  tabindex="4" name="textinf"></textarea>
            </div>
            <div>
                <button tabindex="5" type="submit">Відправити</button>
                <button tabindex="6" type="reset">Очистити</button>
            </div>
        </form>
    </div>
</div>
```

Рисунок 3.15 – Вигляд коду форми реєстрації

3.3 Розробка бекенд частини

Використовуючи мову програмування JavaScript, ми можемо додавати різноманітні властивості до блоків на веб-сторінці. Однією з таких

властивостей є "showHeader", яка була використана для полегшення взаємодії користувача з хедером (верхнім меню) на сайті.

Фрагмент коду, який зображений на рисунку 3.16, демонструє яким чином використовується ця властивість для керування видимістю та переходом між блоками на сторінці.

Завдяки цій властивості, користувач може легко переглядати різні розділи інформації, переходячи від одного блоку до іншого без зайвих труднощів. Це забезпечує швидкий та зручний спосіб взаємодії з веб-сторінкою та дозволяє ефективно відображати важливу інформацію для користувача.

```
VANTA.NET({
  el: "#intro",
  mouseControls: true,
  touchControls: true,
  gyroControls: false,
  minHeight: 200.00,
  minWidth: 200.00,
  scale: 1.00,
  scaleMobile: 1.00,
  color: 0x74bad2,
  backgroundColor: 0x111618,
  points: 11.00,
  maxDistance: 18.00,
  spacing: 17.00
})
window.onscroll = function showHeader() {

  var header = document.querySelector('#header');

  if(window.pageYOffset > 200 ){
    header.classList.add('header__fix');
  }else{
    header.classList.remove('header__fix');
  }
}
```

Рисунок 3.16 – Керування хедером.

Крім того, використання властивості "smooth" також додає елементу елегантності та привабливості до веб-сторінки. Плавні переходи між блоками здатні надати відчуття простоти та гармонії при взаємодії з сайтом.

Користувачі цінують такий підхід, оскільки він створює враження високоякісного та добре продуманого дизайну.

Комбінація властивостей "smooth" та "showHeader" дозволяє нам створити стильний та функціональний веб-сайт, який привертає увагу користувачів та надає їм зручну навігацію та позитивний досвід використання.

Вигляд даного коду, можна переглянути на рисунку 3.17.

```
window.onscroll = function showHeader() {

  var header = document.querySelector('#header');

  if(window.pageYOffset > 200 ){
    header.classList.add('header__fix');
  }else{
    header.classList.remove('header__fix');
  }
}
const anims = document.querySelectorAll('a[href*="#"]')

for (let anim of anims){
  anims.addEventListener("click", function(e){
    e.preventDefault();
    const blockID = anim.getAttribute('href')
    document.querySelector('  '+ blockID).scrollIntoView({
      behavior: "smooth",
      block:"start"
    })
  })
}
```

Рисунок 3.17 – Взаємодія властивостей анімації.

Для забезпечення запису даних в базу даних в нашому проєкті, необхідно налаштувати підключення PHP. Цей процес включає вказівку посилання до відповідного файлу у формі запису, яке можна знайти на рисунку 3.18.

Після успішного підключення PHP, ми повинні правильно вказати всі необхідні дані у PHP-скрипті та в таблиці бази даних. Це включає передачу та обробку даних, введених користувачем, збереження їх у відповідних полях бази даних і налагодження взаємодії між PHP-скриптом та базою даних.

Коректна настройка підключення PHP та правильне вказання даних у PHP-скрипті та базі даних є важливими етапами, що забезпечують успішну передачу та збереження даних у базі даних.

```
<form action="inc/inc.php" method="post">
    <div>
        <p>Вкажіть особисты дані:</p>
        <input tabindex="1" name="name" type="text" />
    </div>
    <div>
        <p>Ваш номер телефону:</p>
        <input tabindex="2" name="telephone" type="text" />
    </div>
    <div>
        <p>Вкажіть дати бронювання та бажані години:</p>
        <input tabindex="3" name="date" type="text" />
    </div>
    <div>
        <p>Додаткова інформація:</p>
        <textarea  tabindex="4" name="textinf"></textarea>
    </div>
    <div>
        <button tabindex="5" type="submit">Відправити</button>
        <button tabindex="6" type="reset">Очистити</button>
    </div>
</form>
```

Рисунок 3.18 – Форма для роботи атрибутів в PHP.

Для забезпечення надійної обробки та взаємодії з даними в нашому проєкті, використовується ряд різноманітних функцій у PHP. Ці функції гарантують правильне збереження та обробку інформації. Крім того, для забезпечення запису коректних даних, ми можемо скористатись фільтрами, які можна знайти на рисунку 3.19.

У випадку, коли зустрічаються помилкові дані, ми можемо використовувати оператори if та else if для їх відображення користувачу. Ці

оператори дозволяють нам встановлювати умови та здійснювати відповідні дії, в залежності від того, чи виконуються умови чи ні.

Застосування різних функцій PHP для перевірки, фільтрації та обробки даних, а також використання операторів if та else if для відображення помилок, гарантує якісну тестування та обробку даних у нашому проєкті. Це допомагає нам підтримувати надійність та якість нашого програмного забезпечення.

```php
<?php
    $name = filter_var(trim($_POST['name']),
    FILTER_SANITIZE_STRING);
    $telephone = filter_var(trim($_POST['telephone']),
    FILTER_SANITIZE_STRING);
    $date = filter_var(trim($_POST['date']),
    FILTER_SANITIZE_STRING);
    $textinf = filter_var(trim($_POST['textinf']),
    FILTER_SANITIZE_STRING);

    if (mb_strlen($name) < 5 || mb_strlen($name) > 100 ) {
        echo " Халепа, перегляньте правильність написання даних ";
        exit();
    } else if (mb_strlen($telephone) < 10 || mb_strlen($telephone) > 15 ) {
        echo " Халепа, перегляньте правильність написання даних ";
        exit();
    } else if (mb_strlen($date) < 10 || mb_strlen($date) > 30 ) {
        echo " Халепа, перегляньте правильність написання даних ";
        exit();
    }
    else if (mb_strlen($textinf) < 1 || mb_strlen($textinf) > 250 ) {
        echo " Халепа, перегляньте правильність написання даних ";
        exit();
    }

    $mysql = new mysqli ('localhost','root','root','animalshome-db');
    $mysql->query("INSERT INTO `tabledb` (name, telephone, date, textinf)
    VALUES ('$name','$telephone','$date','$textinf')");

    $mysql->close();

    header('Location: /');
?>
```

Рисунок 3.19 – Код запису в базу даних.

Після того, як користувач здійснює запис на нашому сайті, введені ним дані зберігаються в спеціальній таблиці бази даних. Ця таблиця містить інформацію, яку користувач надав під час запису.

Працівник, який відповідає за обробку цих даних, має можливість переглянути записи в таблиці, що дозволяє йому отримати доступ до вказаної

інформації користувача. Завдяки цьому працівник може зв'язатися з користувачем, уточнити деталі запиту і отримати повну та точну інформацію, необхідну для подальшої обробки.

Така взаємодія між працівником та користувачем дозволяє ефективно обробляти запити, вирішувати можливі непорозуміння та забезпечувати якісний сервіс для користувачів нашого сайту.

Рисунок 3.20 – Запис в базі даних.

4 БІЗНЕС- ПЛАН РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОНЛАЙН-ГОТЕЛЮ ДЛЯ ТВАРИН

4.1 Резюме проєкту

Назва проєкту: «Розробки інформаційної системи On-line готелю для тварин».

Місцезнаходження проєкту: Україна, Київ.

Ціль проєкту: ціллю проєкту являється зручна інформаційна система для бронювання номеру для домашніх тварин.

Суть проету: Створення інформаційної безпеки, яка буде зручною як і для користувача так і для власника. Швидка обробка запису та загальної інформацію про бронювання номеру.

Передплата. За для безпеки та закупівлі харчів для улюбленця, власник отримує передплату за бронювання номеру заздалегіть. Це дає змогу ефективніше розпланувати час та послуги для власника перед повною оплатою від клієнта.

4.2 Оцінка збуту ринку

На просторах інтернету є велика кількість різноманітних веб-сайтів, які надають різні послуги. Проте, більшість конкурентних сайтів обмежуються лише деяким обсягом послуг для користувачів. Швидкість та зручність оформлення запитів на сайті стали ключовими факторами в наш час.

Наша бізнес-структура є ідеальним варіантом для малих та середніх підприємств. Для реалізації цієї структури необхідно мати просторе приміщення з відповідною кількістю кімнат. Декілька спеціалістів будуть достатньо для забезпечення догляду за тваринами.

Наша команда фахівців забезпечує високий рівень професійності та індивідуальний підхід до кожного клієнта. Ми гарантуємо надання повного спектру послуг для догляду за тваринами, що задовольнять навіть найвимогливіших користувачів. Наша форма бізнесу є ефективним та перспективним рішенням для тих, хто цінує швидкість, зручність та якість обслуговування..

4.3 Конкуренція

В сучасний час спостерігається активне відкриття схожих моделей готелів по всьому світу. Цей бізнес-підхід, як правило, є дуже популярним у Європі. Проте на українському ринку пропозицій з подібним підходом до клієнта виявляється досить мало.

Для успішного розпочатку такого бізнесу необхідно детально вивчити потреби та запити клієнтів на ринку. Розуміння їхніх очікувань та пріоритетів є ключовим етапом у розробці власної концепції готелю. Додатковою вагомою складовою є розгортання бізнесу в Інтернеті.

Для досягнення більшої ефективності необхідно належним чином налаштовувати рекламні кампанії та оптимізувати SEO-продвиження. Додатково можна розглянути можливість закупки реклами на різноманітних платформах та активне управління сторінками в соціальних мережах.

Варто підкреслити, що успішність такого бізнесу значно залежить від ретельного аналізу ринку та активних заходів щодо просування та залучення клієнтів.

Правильно вибрана стратегія маркетингу та належне управління онлайн-присутністю можуть суттєво збільшити його популярність та успіх.

4.4 План розробки

Створення сайту такого типу є складною та вибагливою роботою, яка потребує залучення значної кількості фахівців з різних сфер. Це необхідно для того, щоб врахувати всі можливі ризики розробки, розробити ефективну бізнес-модель та створити просту та надійну систему, задовольняючи потреби користувачів.

У процесі розробки такого сайту, зазвичай залучаються програмісти, дизайнери, копірайтери та спеціалісти з веб-розробки. Кожен з цих фахівців вносить свій внесок у розробку, забезпечуючи якість та функціональність сайту.

Окрім того, розробка такого сайту вимагає детального аналізу та планування. Необхідно враховувати потреби цільової аудиторії, конкурентний ринок та забезпечувати високу взаємодію з користувачами. Важливо створити зручний інтерфейс, надійну систему безпеки та ефективну систему управління контентом.

Загальна мета створення такого сайту полягає в тому, щоб забезпечити користувачам зручний та надійний інструмент для отримання послуг та задоволення їхніх потреб. Це вимагає багато зусиль, але при правильному підході та врахуванні всіх аспектів розробки, можна створити успішний та популярний веб-сайт.

4.5 Організаційний план

Для забезпечення ефективної та якісної роботи над проєктом було розроблено декілька важливих компонентів, які сприяють організації та контролю процесу розробки. Одним з таких компонентів є організаційна структура (ОС) проєкту.

Організаційна структура (ОС) визначає розподіл ролей, відповідальностей та повноважень серед учасників проєкту. Вона допомагає створити чітку ієрархію команди проєкту, забезпечує взаємодію між різними відділами та спеціалістами, а також визначає ланцюжок комунікації.

Крім того, для розподілу та контролю робіт у проєкті використовується структура робіт (WBS) - це ієрархічна декомпозиція проєкту на окремі етапи, фази, завдання та підзавдання. WBS допомагає управляти роботою, розподіляти завдання між учасниками команди, а також встановлювати пріоритети та контролювати прогрес виконання.

План-графік проєкту є ще одним важливим інструментом, який допомагає управляти часом та ресурсами проєкту. Він визначає послідовність виконання робіт, встановлює терміни та крайні дати завершення етапів та задач, а також відображає залежності між різними елементами проєкту.

Використання організаційної структури, структури робіт та план-графіка проєкту допомагає забезпечити систематичний підхід до розробки та керування проєктом. Ці інструменти сприяють ефективному розподілу ресурсів, плануванню та контролю робіт, а також забезпечують відповідну координацію між учасниками команди проєкту.

4.6 Календарний графік виконання проєкту

Для забезпечення ефективної роботи всіх учасників проєкту було розроблено графік виконання, який включав в себе визначення кількості завдань та призначення відповідальних осіб для їх виконання. Цей графік допомагає встановити послідовність та тривалість робіт, необхідних для успішної реалізації проєкту.

Додатковою інформацією про тривалість виконання робіт є Таблиця 1, яка є одним із вхідних даних проєкту. У цій таблиці вказана інформація

щодо часових рамок для кожного завдання, що допомагає управляти графіком виконання та контролювати прогрес проєкту.

Застосування графіка виконання та врахування інформації з Таблиці 1 є важливими етапами у плануванні та керуванні проєктом. Вони допомагають визначити оптимальну послідовність робіт, розподілити ресурси та встановити реалістичні терміни виконання завдань.

Ці інструменти дозволяють забезпечити ефективне керування проєктом та досягти поставлених цілей у відповідних термінах.

Таблиця 4.1 - Вхідні дані.

№	Назва робіт	Склад виконавців	Тривалість(дні)
1	Початок проєкту		0
2	Аналіз задачі	Керівник Проєктний менеджер Аналітик Головний програміст Головний дизайнер	5
3	Розробка ТЗ	Проєктний менеджер Головний дизайнер Головний програміст	4
4	Розробка прототипу	Дизайнер	3
5	Створення макету	Дизайнер	7
6	Затведження макату	Головний дизайнер	1
7	Верстка	Верстальник	15
8	Написання Unit-тестів	Головний програміст	10
9	Розробка БД	Back-end програміст	15
10	Розробка REST API	Back-end програміст 2	12
11	Розробка головної сторінки сайту	Front-end програміст	15
12	Тестування	Тестувальник Головний програміст	7

13	Документування	Головний програміст	5
14	Наповнення контентом	Копірайтер	5
15	Затвердження фінального виду продукту	Керівник Головний програміст Проєктний менеджер	2
16	Встановлення метрик та аналітики	SEO спеціаліст	3
17	SEO просування	SEO спеціаліст	15
18	Рекламна компанія	Маркетогол	15
19	Ввелення в експлуатацію	Головний програміст Проєктний менеджер	2
20	Супровід	Головний програміст Front-end програміст Back-end програміст Тестувальник Проєктний менеджер Дизайнер	30
21	Кінець проєкту		1

Для візуалізації календарного графіка проєкту була використана Діаграма Ганта, яка була побудована за допомогою додатку Microsoft Project. Цей інструмент дозволяє представити послідовність робіт та їх тривалість у вигляді графічної діаграми з використанням позначок, проміжків та залежностей між завданнями [11].

Використання Діаграми Ганта дозволяє візуалізувати графічно всі етапи проєкту, показати залежності між завданнями та визначити критичні шляхи. Це допомагає керівникам проєкту та учасникам команди краще розуміти хронологію та ресурси, необхідні для успішного виконання проєкту.

Додаток Microsoft Project є потужним інструментом управління проєктами, який дозволяє планувати, відстежувати та керувати різними

аспектами проєкту. Будівництво календарного графіка з використанням цього додатку забезпечує точність та надійність у плануванні та виконанні проєкту. Діагарма Ганта представлена на рисунку 4.1 – 4.5.

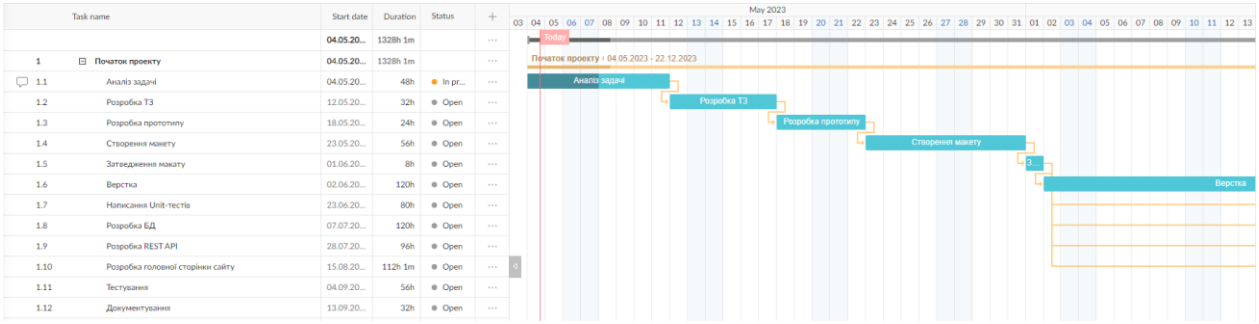


Рисунок 4.1 – Календарний графік у вигляді діаграми Ганта

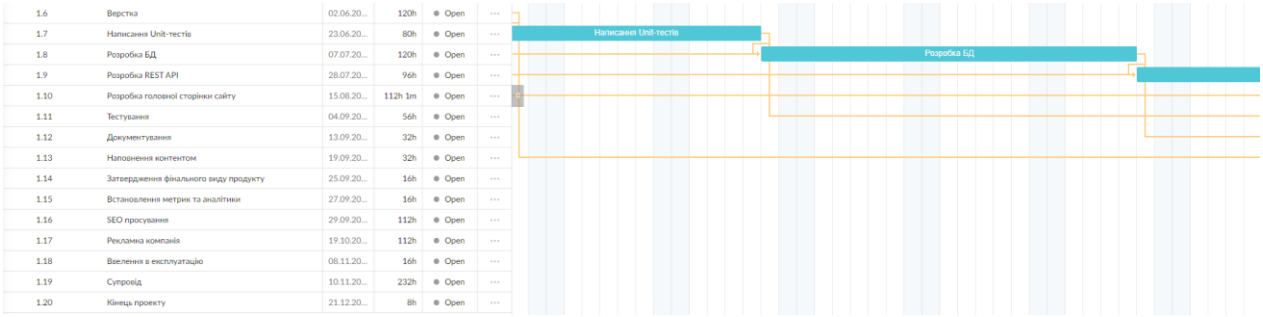


Рисунок 4.2 – Продовження графіку у вигляді діаграми Ганта

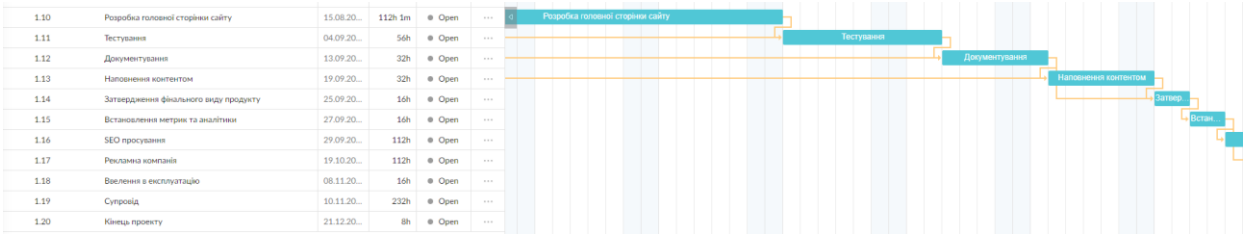


Рисунок 4.3 – Продовження графіку у вигляді діаграми Ганта

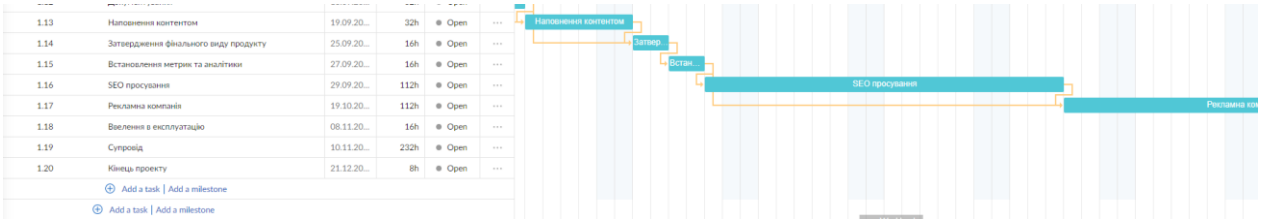


Рисунок 4.4 – Продовження графіку у вигляді діаграми Ганта

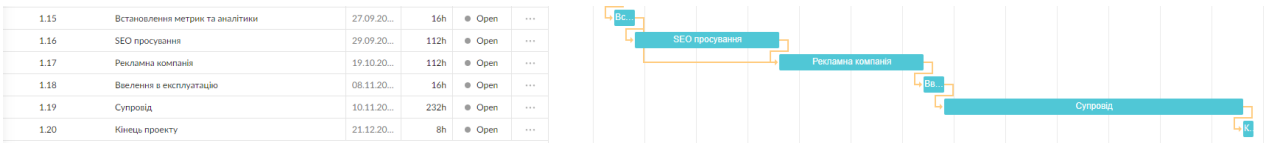


Рисунок 4.5 – Продовження графіку у вигляді діаграми Ганта

4.7 Сітьовий графік

На основі переліку робіт, що наведений у Таблиці 1, було побудовано сіткову діаграму за допомогою програмного забезпечення Microsoft Project. Ця діаграма детально відображає послідовність та залежності між різними завданнями проєкту.

Сіткова діаграма, яку можна побачити на рисунках 4.6-4.10, допомагає визначити послідовність виконання робіт, визначити критичні шляхи та розподілити ресурси. Цей графічний інструмент дає змогу краще розуміти хронологію та логіку виконання завдань у проєкті.

Використання програмного забезпечення Microsoft Project дозволяє ефективно планувати, керувати та контролювати проєкти будь-якої складності. За допомогою сіткової діаграми, побудованої в цьому програмному забезпеченні, команда проєкту може візуалізувати та керувати всіма аспектами проєкту, що сприяє ефективному та успішному його виконанню.

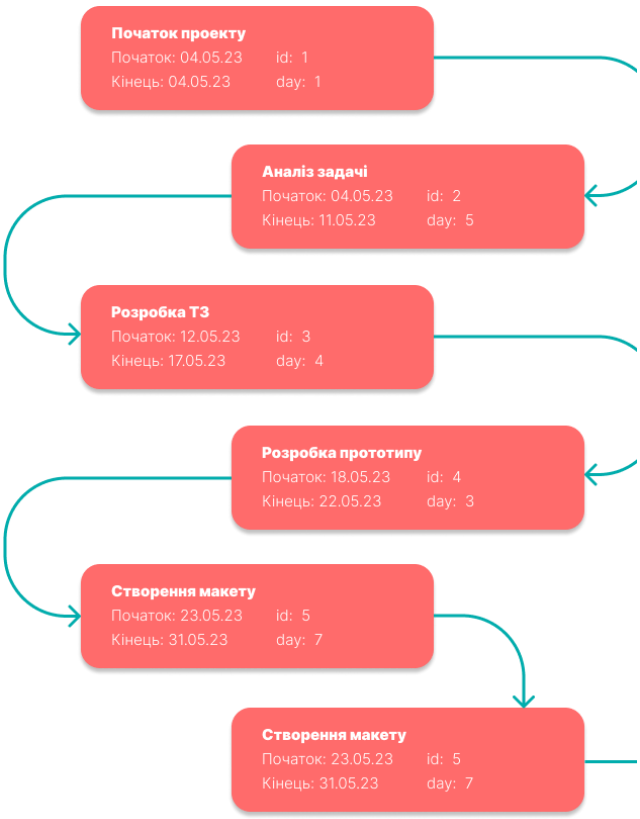


Рисунок 4.6 – Початок сітьового графіку

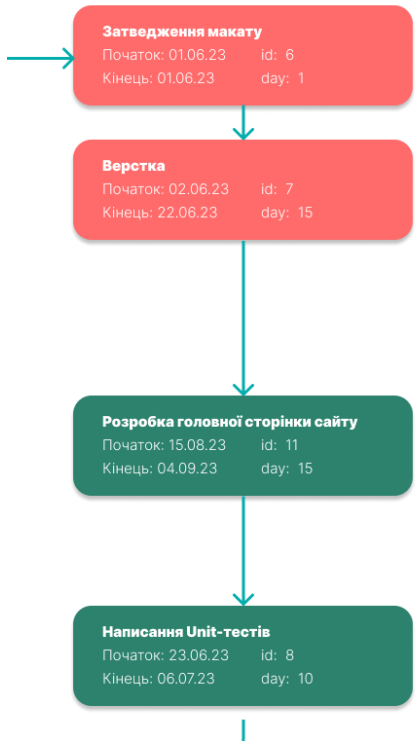


Рисунок 4.7 – Продовження сітьового графіку

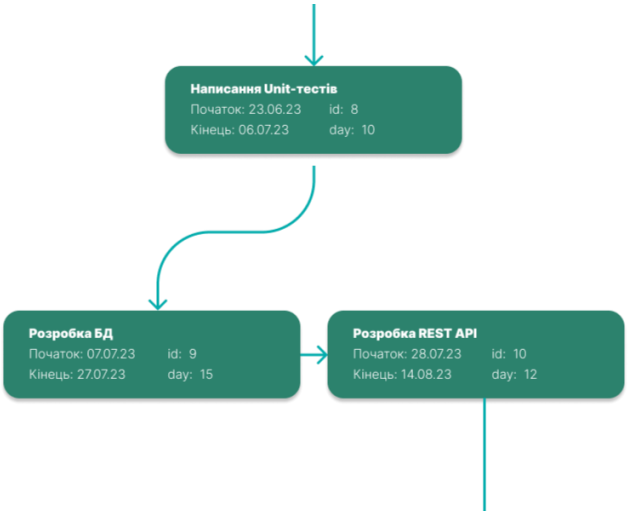


Рисунок 4.8 – Продовження сітьового графіку

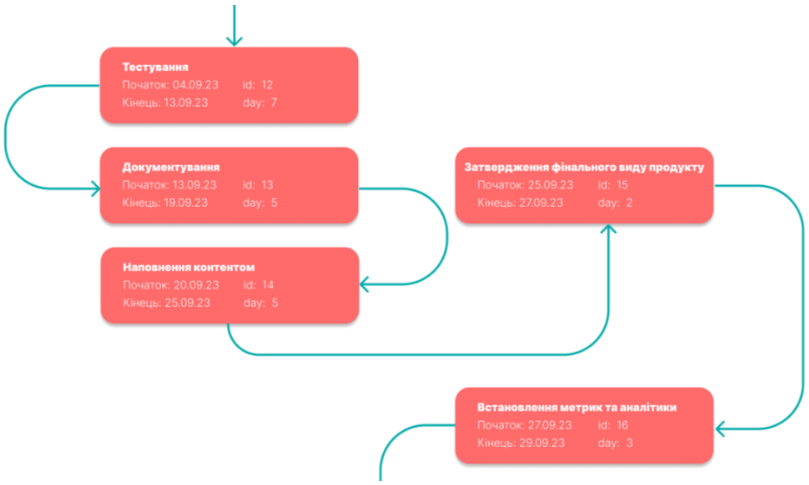


Рисунок 4.9 – Продовження сітьового графіку

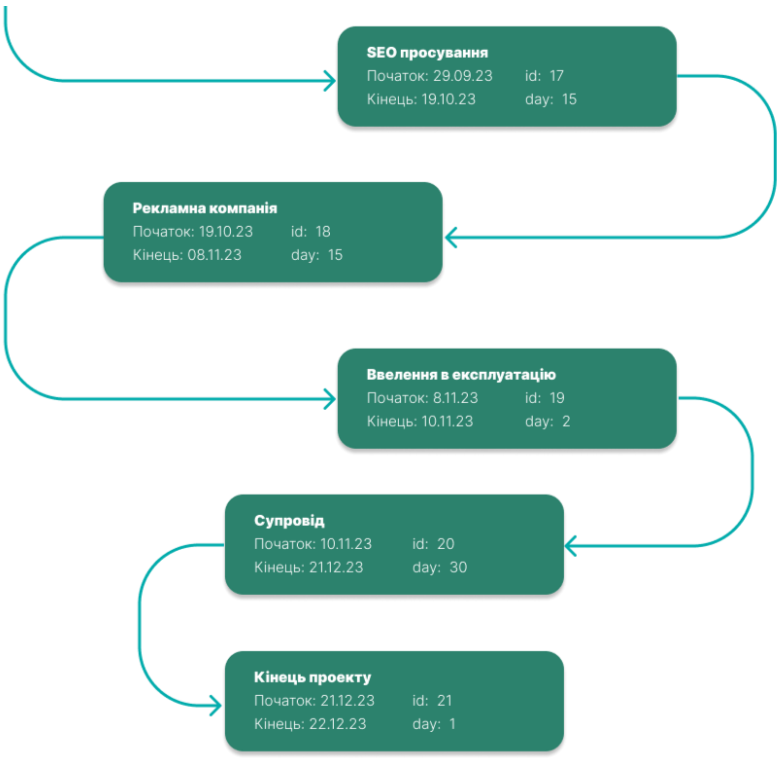


Рисунок 4.10 – Продовження сітьового графіку

4.8 WBS – модель проєкту

Згідно з темою роботи, була проведена декомпозиція проєкту, що дозволило розбити його на менші, більш управліні етапи та завдання. У загальному вигляді ця декомпозиція проєкту зображена на рисунку 4.34.

Декомпозиція є важливим етапом в управлінні проєктом, оскільки дозволяє розбити складний проєкт на більш керовані та керовані елементи. Це допомагає уточнити обсяг робіт, визначити послідовність виконання та розподілити ресурси.

Зображена на рисунку 4.11 декомпозиція проєкту надає загальний огляд структури та організації проєкту, демонструючи відносини між його основними елементами.

Це важливий інструмент для керівників проєкту та всіх учасників команди, щоб мати чітке уявлення про структуру та обсяг робіт, необхідних для успішного виконання проєкту.

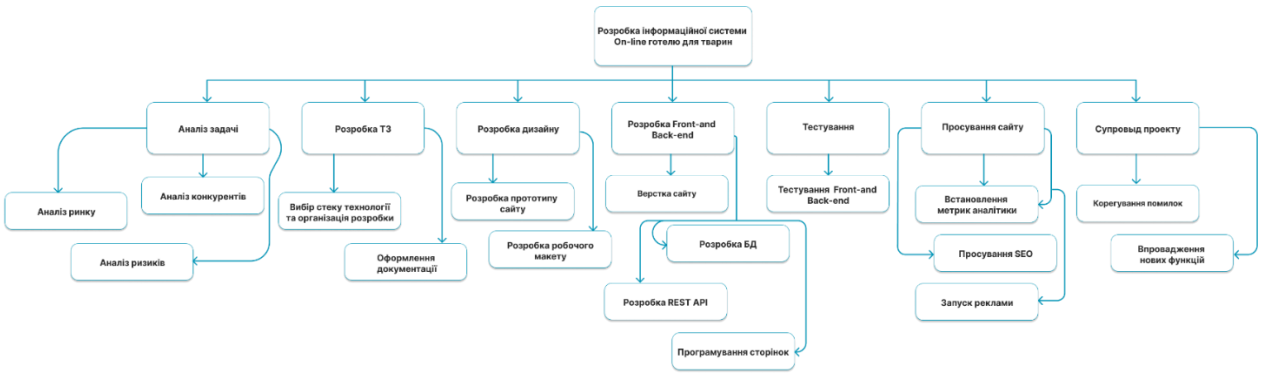


Рисунок 4.11 – WBS модель проєкту

4.9 OBS – модель проєкту

На рисунку 4.12 представлена організаційна структура проєкту, відома також як організаційна модель проєкту. Організаційна структура проєкту визначає ієрархію та структуру ролей, відповідальностей та комунікації всередині проєкту.

Ця модель відображає всі учасників проєкту, включаючи керівників, команди, підрядники та інших зацікавлених сторін. Вона надає візуальну репрезентацію структури організації та взаємозв'язків між різними ролями та функціональними групами.

Організаційна структура проєкту допомагає встановити чіткі лінії комунікації та визначити ролі та відповідальності кожного учасника проєкту. Це сприяє ефективній комунікації, координації робіт та досягненню поставлених цілей проєкту.

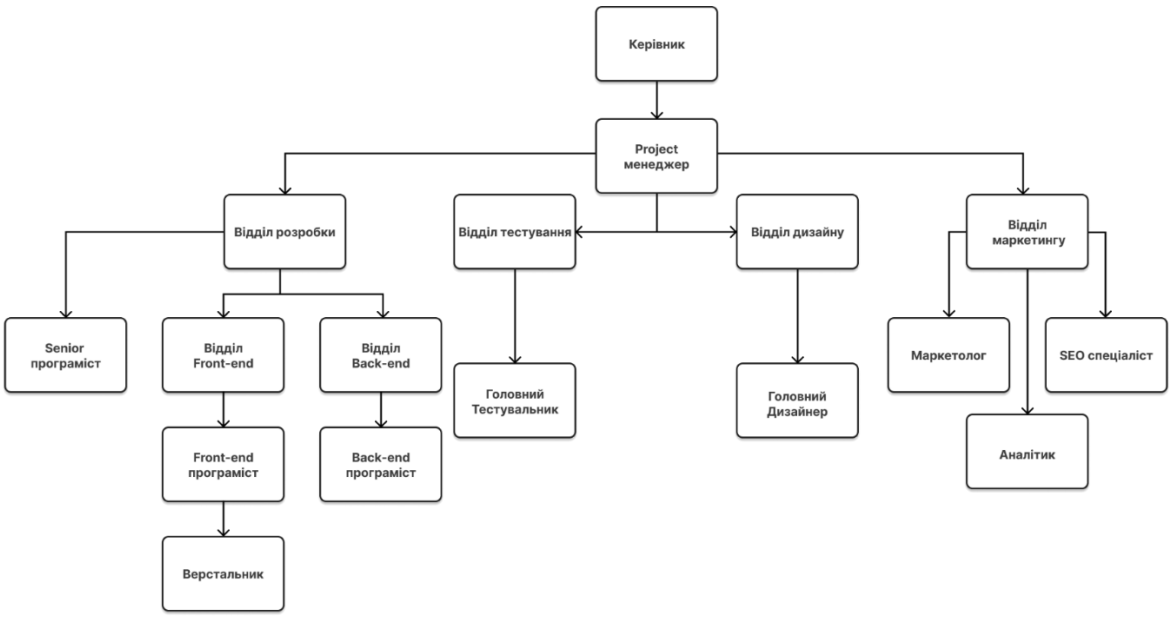


Рисунок 4.12 – OBS модель проєкту

4.10 План виконання проєкту

Створення плану виконання проєкту було успішно здійснено з використанням програмного забезпечення Microsoft Project. На рисунках 4.13 і 4.14 наведений детальний план проєкту, що включає послідовність робіт, їх тривалість, залежності між роботами та виділені ресурси.

Цей план виконання дозволяє управляти ходом проєкту, розподіляти завдання між учасниками команди, визначати терміни та критичні шляхи. Він також допомагає виявляти можливі затримки, перекриття та ресурсні конфлікти, що дозволяє вчасно приймати відповідні корективні заходи для забезпечення успішного виконання проєкту.

Створений план виконання проєкту є важливим інструментом для керування проєктом, контролю прогресу та досягнення поставлених цілей. Він служить основою для планування та координації робіт, а також дозволяє учасникам проєкту мати чітке уявлення про терміни та очікувані результати.

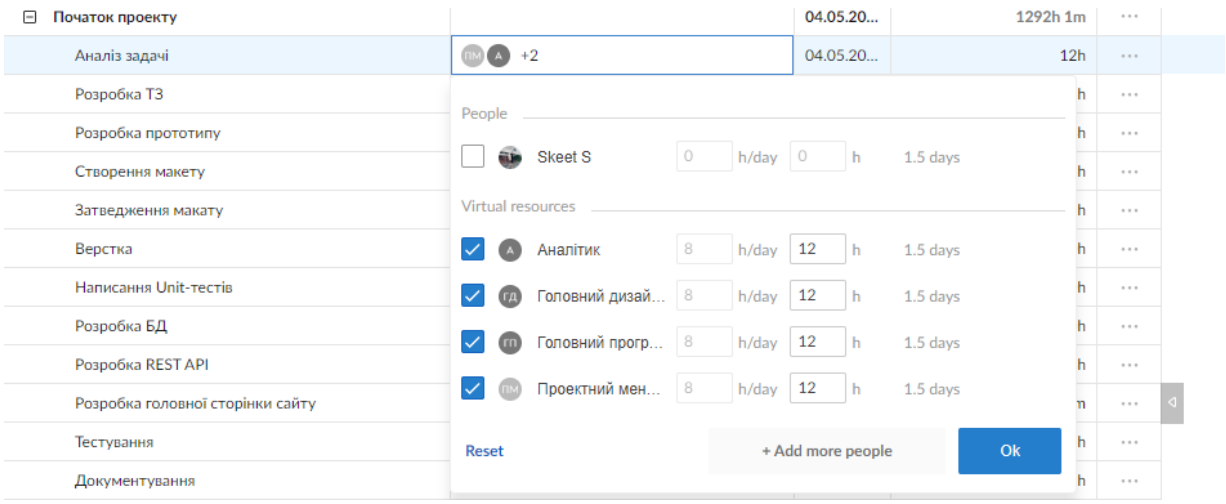


Рисунок 4.13 – План виконання проєкту

	04.05.20...	1292h 1m
Початок проекту	04.05.20...	1292h 1m
Аналіз задачі	04.05.20...	12h
Розробка ТЗ	05.05.20...	32h
Розробка прототипу	11.05.20...	24h
Створення макету	16.05.20...	56h
Затвердження макату	25.05.20...	8h
Верстка	26.05.20...	120h
Написання Unit-тестів	16.06.20...	80h
Розробка БД	30.06.20...	120h
Розробка REST API	21.07.20...	96h
Розробка головної сторінки сайту	08.08.20...	112h 1m
Тестування	28.08.20...	56h
Документування	06.09.20...	32h
Наповнення контентом	12.09.20...	32h
Затвердження фінального виду продукту	18.09.20...	16h
Встановлення метрик та аналітики	20.09.20...	16h
SEO просування	22.09.20...	112h
Рекламна компанія	12.10.20...	112h
Ввелення в експлуатацію	01.11.20...	16h
Супровід	03.11.20...	232h
Кінець проекту	14.12.20...	8h

Рисунок 4.14 – План виконання проєкту

4.11 Конфлікти в проєкті

Виробничі конфлікти можуть виявитися джерелом розвитку та стимулом до кращої продуктивності. Ці конфлікти можуть посприяти розбудові інтелектуальної конкуренції, яка в свою чергу спонукає членів команди працювати на максимум своїх можливостей. Вони можуть збільшити творчий потенціал, оскільки конфлікти можуть змусити нас знайти нові та нестандартні рішення, виходячи за межі усталених шаблонів.

Результатом таких конфліктів може стати створення більш "креативного" продукту або здійснення більш ефективних рішень. Крім того, конфлікти можуть сприяти згуртованості команди та підвищенню її ефективності у досягненні кінцевої мети проєкту.

Важливо зазначити, що в контексті виробничих конфліктів мова йде про суперечки, пов'язані з професійною діяльністю, а не особистими суперечками. Особистісні конфлікти не є конструктивними для ефективної команди.

У разі, коли особистісні проблеми стають джерелом конфлікту, важливо вжити заходи щодо видалення або ізоляції особи, яка становить загрозу співпраці та колективній роботі команди. Часто навіть один негативний елемент може негативно позначитися на роботі всієї команди.

Щодо вирішення конфліктів, можливим підходом є застосування принципу "програвай/вигравай". Наприклад, можна зробити компроміс та пожертвувати своїми інтересами на користь інтересів іншої сторони, або погодитися з рішенням начальника, коли не вдається досягти взаєморозуміння. Проте, важливо зберігати свій авторитет і не зловживати цим підходом.

Інший підхід до вирішення конфліктів - це ухилення. У деяких випадках, якщо часові рамки дозволяють, можна відкласти вирішення проблеми. Іноді конфлікт стає менш актуальним з плином часу або сторони набувають додаткової інформації, яка може допомогти знайти більш

ефективне рішення. Однак, варто пам'ятати, що з часом конфлікт може загостритися та стати ще складнішим для вирішення.

ВИСНОВОК

В результаті проведеного дослідження та розробки інформаційної системи онлайн готелю для тварин було досягнуто основну мету проекту - поліпшення обслуговування клієнтів і забезпечення зручності бронювання номерів для власників тварин. Застосування системи дозволяє користувачам здійснювати бронювання номерів онлайн, переглядати інформацію про доступність номерів та проводити платежі через Інтернет.

В ході аналізу вимог користувачів та функціональних можливостей системи було виявлено, що цілком відповідає потребам клієнтів. Проектування програмного забезпечення та його практична реалізація були проведені з урахуванням цих вимог, що сприяло досягненню успішного результату.

Створена інформаційна система є інноваційним продуктом на ринку готелів для тварин. Вона надає користувачам зручність та швидкість процесу бронювання, а також можливість отримання актуальної інформації про доступність номерів. Це сприяє поліпшенню досвіду користувачів та задоволенню їх потреб у затишному перебуванні їх тварин у готелі.

У майбутньому, система може розширюватися та вдосконалюватися шляхом додавання нових функцій та можливостей. Наприклад, впровадження відстеження здоров'я та стану тварин. Це дозволить системі ще більше задовольняти потреби користувачів та збільшить її конкурентоспроможність на ринку.

У цілому, розробка інформаційної системи онлайн готелю для тварин була успішною та відповідає вимогам проекту. Її впровадження дозволить покращити якість обслуговування клієнтів та зайняти важливу нішу на ринку готелів для тварин.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

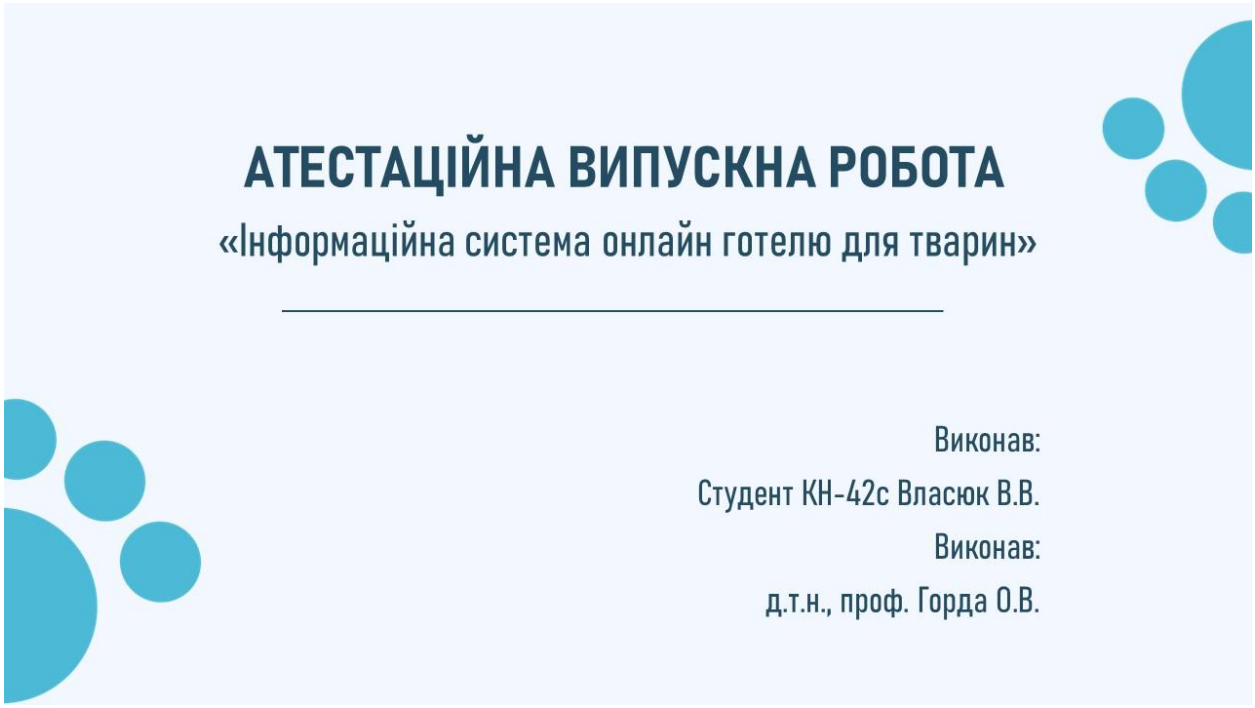
1. Цюцюра С. В. «Методичні вказівки до виконання дипломної роботи» – // КНУБА – 2018 р.
2. Microsoft SQL Server [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Microsoft_SQL_Server
3. REST [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/REST
4. CASE [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/CASE
5. Google Forms [Електронний ресурс] // Google – 2023. – Режим доступу до ресурсу: https://www.google.com/intl/ua/forms/about/
6. HTML [Електронний ресурс] // Mc.today – 2023. – Режим доступу до ресурсу: https://mc.today/uk/shho-take-html-ta-yak-za-dopomogoyu-nogo-uvijti-do-it/
7. CSS [Електронний ресурс] // Css.in.ua – 2023. – Режим доступу до ресурсу: https://css.in.ua/article/shcho-take-html_10
8. UML діаграми [Електронний ресурс] // Docs.kde.org – 2023. – Режим доступу до ресурсу: https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml-basics.html
9. Побудова дерева цілей [Електронний ресурс] // Pidru4niki.com – 2023. – Режим доступу до ресурсу: https://pidru4niki.com/18180520/ekonomika/pobudova_dereva_tsiley
10. Роберт Мартін. Чистий код. Правильність написання програмного коду для продукту 2021. – 371с.
11. Діаграма Ганта [Електронний ресурс] // Nachasi.com – 2023. – Режим доступу до ресурсу: https://nachasi.com/creative/2020/09/03/gantt-chart/

12. Java Skript програмування [Електронний ресурс] // Uk.javascript.info – 2023. – Режим доступу до ресурсу: https://uk.javascript.info
13. PHP [Електронний ресурс] // Programming.in.ua – 2023. – Режим доступу до ресурсу: http://programming.in.ua/web-design/allphp/30-about-php.html
14. Що таке Figma? [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2023. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Figma
15. Для чого потрібен MAMP? [Електронний ресурс] // Biblprog.org.ua – 2023. – Режим доступу до ресурсу: https://biblprog.org.ua/ua/mamp/
16. SQL посібник [Електронний ресурс] // Sqlstyle.guide – 2023. – Режим доступу до ресурсу: https://www.sqlstyle.guide/ua/
17. Ицик Бен-Ган. Microsoft SQL Server 2012. Основи T-SQL. // Microsoft Press – 2012. – 982 с.
18. Декларативна мова програмування SQL [Електронний ресурс] // Wikibooks – 2023. – Режим доступу до ресурсу: https://uk.wikibooks.org/wiki/SQL
19. Бази даних [Електронний ресурс] // Hostkoss.com – 2023. – Режим доступу до ресурсу: https://hostkoss.com/b/uk/database/
20. HTTP [Електронний ресурс] // Матеріал з Вікіпедії — вільної енциклопедії. – 2022. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/HTTP
21. Що таке Java Skript? [Електронний ресурс] // Javascript.info – 2023. – Режим доступу до ресурсу: https://uk.javascript.info/intro
22. Чому корисно знати про Localhost? [Електронний ресурс] // Hostenko.com – 2023. – Режим доступу до ресурсу: https://hostenko.com/wpcafe/tutorials/shho-take-localhost-i-chomu-mozhe-znadobitisja-stvoriti-lokalnij-veb-sajt-wordpress/
23. Що таке API? [Електронний ресурс] // Hostiq.ua – 2023. – Режим доступу до ресурсу: https://hostiq.ua/blog/ukr/what-is-api/

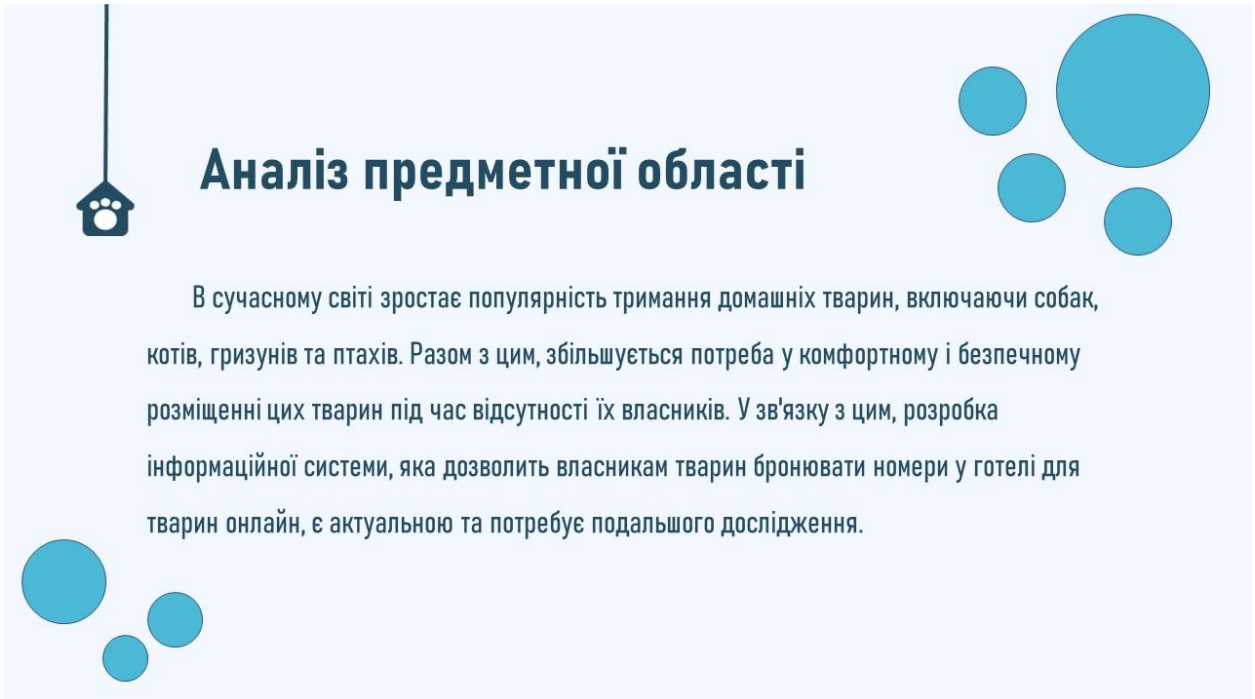
24. Етапи розробки сайту [Електронний ресурс] // Websait.uz.ua – 2023. – Режим доступу до ресурсу: https://websait.uz.ua/website-development/etapy-rozrobky-sajtu/
25. Джон Маеда. Закони простоти. Основи UI/UX дизайну – 2016. – 673с.
26. Основи UI/UX [Електронний ресурс] // Dou.ua – 2023. – Режим доступу до ресурсу: https://dou.ua/forums/topic/42880/.

Додаток А

Презентація Атестаційної випускної роботи

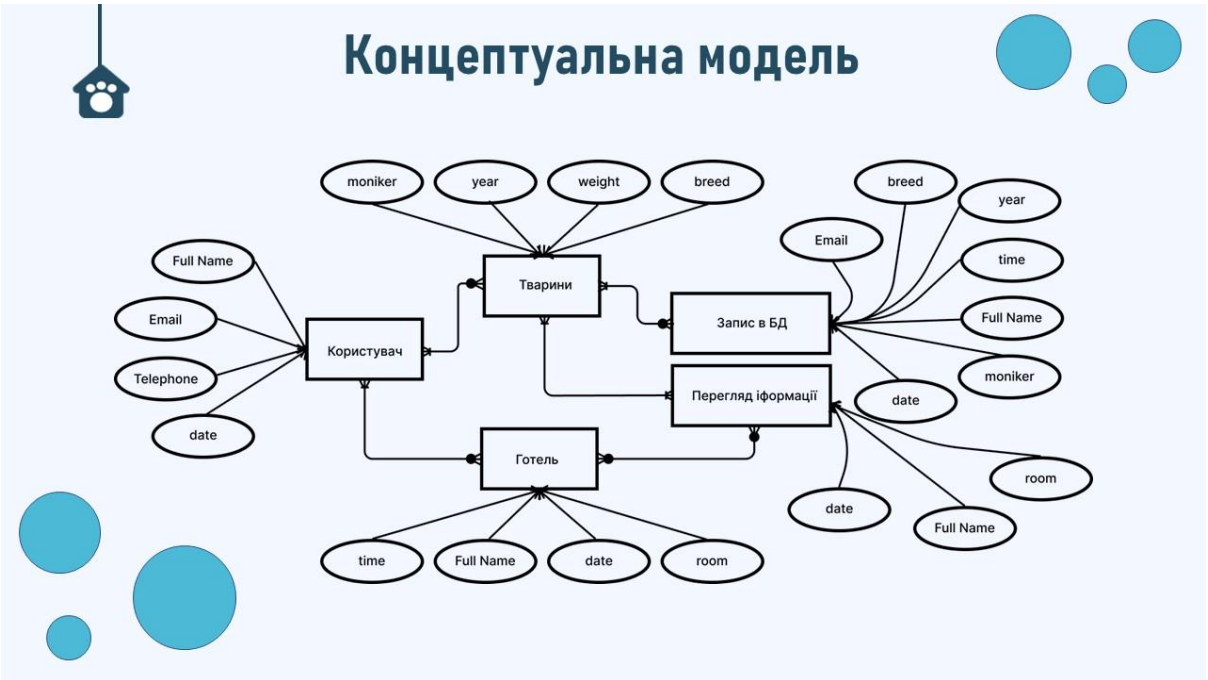


Слайд 1 – Розробка інформаційної системи онлайн готелю для тварин



Слайд 2 – Аналіз предметної області

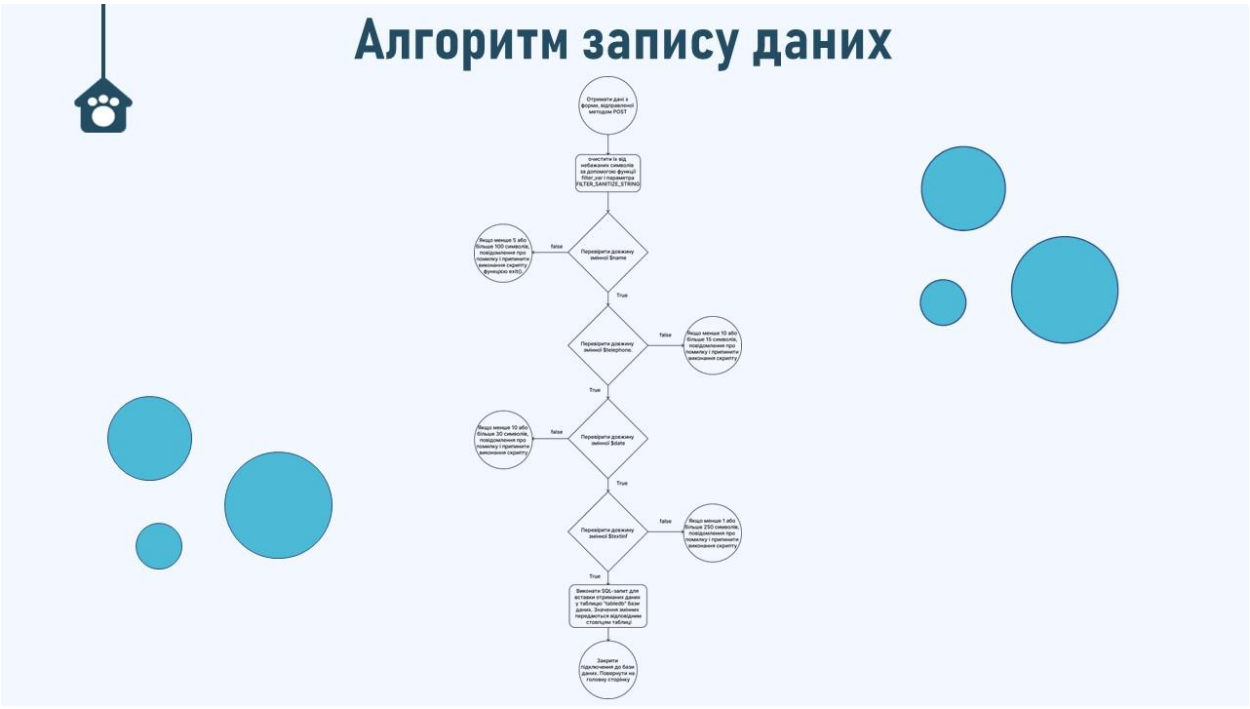
Слайд 3 – Дерево функцій



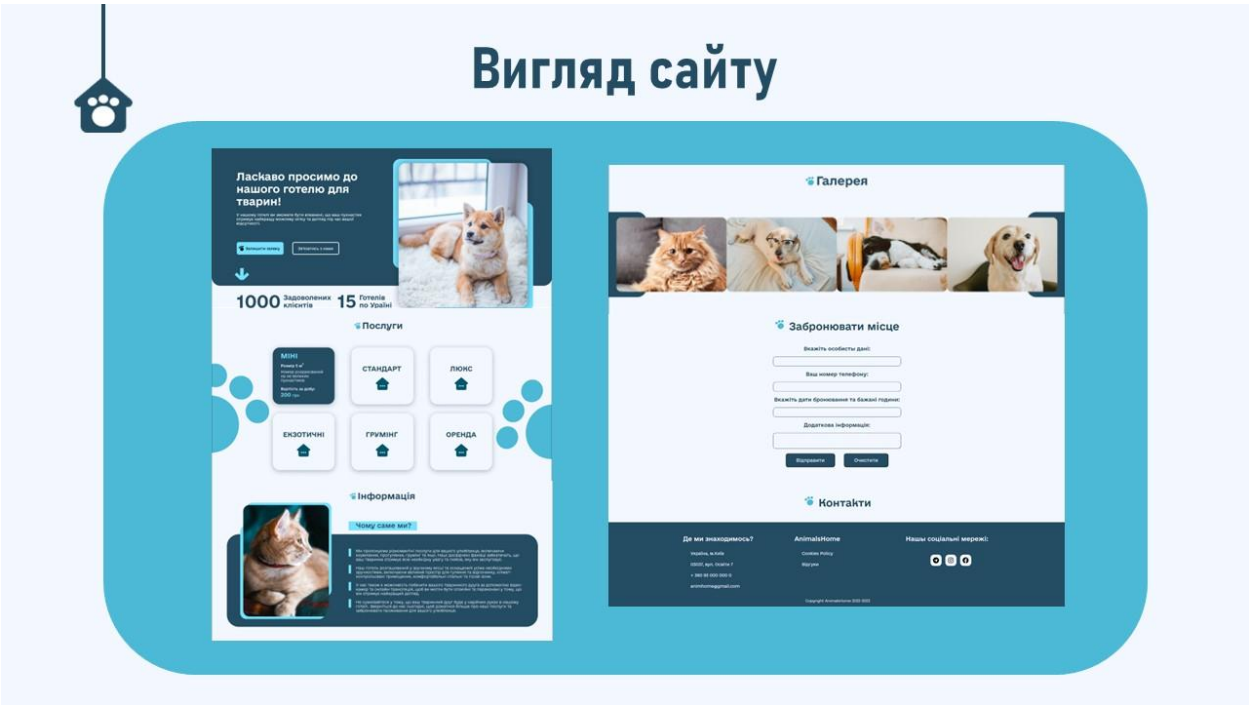
Слайд 4 – Концептуальна модель

Слайд 5 – База даних системи

Слайд 6 – Алгоритм реєстрації користувача



Слайд 7 – Алгоритм запису даних



Слайд 8 – Вигляд сайту

Приклад запису даних

Слайд 9 – Запис інформації в базу даних

ВИСНОВКИ

Проект інформаційної системи онлайн готелю для тварин був успішно реалізований. Вимоги користувачів та функціональні можливості системи були визначені під час аналізу та дослідження проблеми. Використовуючи JavaScript та PHP з базою даних MySQL, було розроблено функціональний сайт. Результатом є створена інформаційна система, яка забезпечує зручне бронювання номерів для тварин. Бізнес-план виявив потенціал проекту та визначив можливості розвитку, включаючи додаткові функції. Цей інноваційний продукт поліпшує обслуговування клієнтів та забезпечує зручність для власників тварин.

Слайд 10 – Висновки