

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

«Веб-орієнтований додаток приюту для тварин»

Студента 4 курсу, 6 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
освітньої програми «Інженерія
програмного забезпечення»

Компанець Денис
Олександрович

підпис студента

Науковий керівник
Кандидат педагогічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

Котенко Наталія
Олексіївна

підпис керівника

Гарант освітньої програми
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

Рзаєва Світлана
Леонідівна

підпис гаранта

Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Котенко Н. О.

«14» листопада 2022 р.

Завдання

на випускний кваліфікаційний проєкт студентів

Компанець Денис Олександрович

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Веб-орієнтований додаток
приюту для тварин»

Затверджена наказом ректора від «6» грудня 2022 р. № 3288

2. Строк здачі студентом закінченого проєкту 26 липня 2023

3. Цільова установка та вихідні дані до проєкту

Мета дослідження – аналіз можливостей, які надають веб-орієнтовані
додатки в контексті організації роботи притуків для тварин, виявлення їх
переваг та недоліків. Систематизація існуючих підходів до розробки веб-
додатків. Практична реалізація веб-додатку притулку для тварин.

Об'єктом дослідження є організація діяльності та потреби притулку для
тварин.

Предметом дослідження є методи та засоби проектування веб-додатку
притулку для тварин.

4. Консультанти проєкту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проєкту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Передумови створення веб-орієнтованого додатку

1.2. Вимоги до структури та функціонування системи

1.3. Огляд проблемної області

1.4. Постановка задачі

1.5. Технічне завдання

1.7. Висновок до розділу 1

РОЗДІЛ 2. ОГЛЯД ТЕХНОЛОГІЙ ТА ІНСТРУМЕНТІВ

2.1. Інструменти розробки

2.2 Front-end

2.3 Back-end

2.4 Висновок до розділу 2

РОЗДІЛ 3. РОЗРОБКА WEB-ОРІЄТОВАНОГО ДОДАТКУ ПРИТУЛКУ
ДЛЯ ТВАРИН

3.1. Концепція веб-додатку

3.2. Розробка веб-додатку

3.2.1 Макет та прототип веб-додатку засобами Figma

3.2.2 Використання технологій програмування

3.3 Проектування бази даних

3.4. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2022	21.09.2022
2.	<i>Розробка та затвердження завдання на проєкт</i>	14.11.2022	14.11.2022
3.	<i>Вступ та перелік літературних джерел</i>	23.12.2022	23.12.2022
4.	<i>Розділ 1. Огляд теми та існуючих рішень</i>	27.01.2023	27.01.2023
5.	<i>Розділ 2. Огляд технологій та інструментів для його розробки</i>	03.03.2023	03.03.2023
6.	<i>Розділ 3. Розробка web-орієнтованого додатку притулку для тварин</i>	14.04.2023	14.04.2023
7.	<i>Висновки</i>	28.04.2023	28.04.2023
8.	<i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	17.05.2023	17.05.2023
9.	<i>Підготовка автореферату та презентації доповіді</i>		
10.	<i>Попередній захист випускного кваліфікаційного проєкту</i>		
11.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>		
12.	<i>Здача прошого випускного кваліфікаційного проєкту на кафедрі</i>		
13.	<i>Публічний захист випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «14» листопада 2022 р.

8. Науковий керівник випускного кваліфікаційного проєкту Котенко Н.О.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Рзаєва С.Л.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Компанець Д. О.

(прізвище, ініціали, підпис)

АНОТАЦІЯ

У даній роботі було здійснено дослідження існуючих підходів та розроблено веб-орієнтований додаток притулку для тварин.

Метою дослідження є аналіз можливостей, які надають веборієнтовані додатки в контексті організації роботи притулків для тварин, виявлення їх переваг та недоліків. Систематизація існуючих підходів до розробки вебдодатків. Практична реалізація вебдодатку притулку для тварин.

Об'єктом дослідження є організація діяльності та потреби притулку для тварин.

Предметом дослідження є методи та засоби проектування вебдодатку притулку для тварин.

Ключові слова: веб-додаток, притулок для тварин

ABSTRACT

In this work, research was carried out on a web-based application of an animal shelter.

The purpose of the study is to analyze the opportunities provided by web-based applications in the context of organizing the work of animal shelters, identifying their advantages and disadvantages. Systematization of existing approaches to the development of web applications. Practical implementation of the animal shelter web application.

The object of the study is the organization of activities and needs of the shelter for animals.

The subject of research is methods and means of designing a web application of an animal shelter.

Keywords: web application, animal shelter

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

HTML (HyperText Markup Language) – мова гіпертекстової розмітки

CSS (Cascading Style Sheets) – каскадні таблиці стилів

PHP (*Hypertext Preprocessor*) – препроцесор гіпертексту



<i>ДТЕУ 121 06-15.БР</i>				
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
Зав. каф.		Криворучко О.В.		14.04.23
Керівник		Котенко Н. О.		14.04.23
Гарант		Рзасва С.Л.		14.04.23
Розробив		Компанець Д. О.		14.04.23
Веб-орієнтований додаток приюту для тварин				
<i>Перелік умовних скорочень</i>				
<i>Стадія</i>		<i>Аркуш</i>		<i>Аркушів</i>
<i>ПС</i>		2		50
<i>Факультет інформаційних технологій 4 курс, 6 група</i>				

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД ТА ПОСТАНОВКА ЗАДАЧІ	4
1.1. Передумови створення веб-орієнтованого додатку	4
1.2 Вимоги до структури та функціонування системи	8
1.3 Огляд проблемної області	9
1.4 Постановка задачі	13
1.5 Технічне завдання	14
1.6 Висновок до Розділу 1:	28
РОЗДІЛ 2 ОГЛЯД ТЕХНОЛОГІЙ ТА ІНСТРЕМНТІВ	30
2.1. Інструменти розробки	30
2.2 Front-end	31
2.3 Back-end.....	36
2.4 Висновок до Розділу 2:	40
РОЗДІЛ 3 РОЗРОБКА ВЕБ -ОРІЄНТОВАНОГО ДОДАТКУ ПРИТУЛКУ ДЛЯ ТВАРИН	42
3.1. Концепція веб-додатку	42
3.2 Розробка веб-додатку	42
3.2.1 Макет та прототип вебдодатку засобами Figma	42
3.2.2 Використані технології програмування	44
3.3 Проектування бази даних	44
3.4 Висновок до Розділу 3.....	45
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	48
ДОДАТКИ	49

					<i>ДТЕУ 121 06-16.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Веб-орієнтований додаток приноуту для тварин	<i>Стадія</i>	<i>Арку</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		23.12.22		3	3	50
Керівник		Котенко Н. О.		23.12.22		Факультет інформаційних технологій 4 курс, 6 група		
Гарант		Рзаєва С.Л.		23.12.22				
Розробив		Компанець Д. О.		23.12.22				
					<i>Зміст</i>			

ВСТУП

Актуальність Приют для тварин відіграють важливу роль у нашому суспільстві, забезпечуючи безпечний та турботливий притулок для тварин, які втратили своїх господарів або були кинуті на вулиці. Проте управління притулком для тварин може бути складним завданням, особливо коли йдеться про управління великою кількістю тварин, персоналом, волонтерами, фінансами та ресурсами.

У цій роботі буде розроблено веб-орієнтований додаток приюту для тварин, який може допомогти спростити управління притулком і підвищити ефективність роботи. Ми розглянемо різні функції та можливості, які можуть бути включені в додаток, включаючи управління тваринами, персоналом, волонтерами та ресурсами, а також пошук будинку для тварин та спілкування із спільнотою.

Мета дослідження – аналіз можливостей, які надають веб-орієнтовані додатки в контексті організації роботи приютів для тварин, виявлення їх переваг та недоліків. Систематизація існуючих підходів до розробки веб-додатків. Практична реалізація веб-додатку притулку для тварин.

Об'єктом дослідження є організація діяльності та потреби притулку для тварин.

Предметом дослідження є методи та засоби проектування веб-додатку притулку для тварин.

Завдання – розглянути існуючі веб-додатки притулків для тварин, проаналізувати їх сильні та слабкі сторони; обрати оптимальну технологію розробки; розробити веб-додаток притулку для тварин

					<i>ДТЕУ 121 06-15.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Веб-орієнтований додаток приюту для тварин	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		23.12.22		<i>В</i>	4	50
Керівник		Котенко Н. О.		23.12.22		Факультет інформаційних технологій 4 курс, 6 група		
Гарант		Рзаєва С.Л.		23.12.22				
Розробив		Компанець Д. О..		23.12.22	<i>Вступ</i>			

РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Передумови створення веб-орієнтованого додатку

Метою створення системи веб-орієнтованого додатку приюту для тварин є забезпечення ефективного управління та покращення якості життя тварин, що перебувають в приюті, а також сприяння у процесі їх усиновлення та забезпечення фінансової підтримки для розвитку приюту. Головні завдання системи веб-орієнтованого додатку для приюту для тварин включають в себе:

По-перше це забезпечення зручного та швидкого доступу користувачів до інформації про тварин, що перебувають в приюті.

По-друге я би виділив забезпечення можливості здійснення заявок на усиновлення тварин, що перебувають в приюті, та моніторингу стану заявки.

По-друге це забезпечення можливості здійснення онлайн-донатів та пожертвувань на розвиток та підтримку роботи приюту.

Ще забезпечення можливості контактування з представниками приюту та отримання консультацій щодо усиновлення тварин та сприяння їхньому благополуччю після усиновлення.

По-третє це забезпечення адміністраторів приюту зручним та ефективним інструментом для управління тваринами, заявками на усиновлення, пожертвуваннями та іншими аспектами роботи приюту.

Отже, створення системи веб-орієнтованого додатку для приюту для тварин дозволить покращити ефективність та зручність управління приютом, а також сприятиме підвищенню усвідомленості громадськості про проблеми безпритульних тварин.

Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		27.01.23	Веб-орієнтований додаток приюту для тварин	Стадія	Аркуш	Аркушів
Керівник		Котенко Н. О..		27.01.23		РІ	7	50
Гарант		Рзаєва С.Л.		27.01.23		Факультет інформаційних технологій		
Розробив		Компанець Д. О.		27.01.23		4 курс, 6 група		
					Огляд теми та існуючих рішень			

ДТЕУ 121 06-15.БР

1.2 Вимоги до структури та функціонування системи

Структура та функціонування системи повинна бути добре організованою та логічною. Для забезпечення ефективності та користувацької зручності, система повинна складатись з наступних підсистем:

а) Підсистема заявок на усиновлення. Ця підсистема повинна дозволяти користувачам заповнювати та відправляти заявки на усиновлення тварин. Підсистема повинна містити форми для введення даних про користувача та тварину, яку він хоче усиновити. Для забезпечення ефективності, підсистема повинна мати функції, що дозволяють відслідковувати статус заявки.

б) Підсистема донатів. Ця підсистема повинна дозволяти користувачам здійснювати пожертвування для підтримки роботи приюту. Підсистема повинна містити форму для введення даних про користувача та обрану суму пожертвування. Для забезпечення ефективності, підсистема повинна мати функції, що дозволяють відслідковувати здійснені пожертвування та відправляти дякувальні повідомлення.

в) Підсистема пошуку тварин. Ця підсистема повинна дозволяти користувачам шукати тварин, яких вони хочуть усиновити. Підсистема повинна мати фільтри для пошуку тварин за різними параметрами, такими як порода, вік, стать та інші.

г) Підсистема опису тварин. Ця підсистема повинна містити інформацію про кожну тварину в приюті, таку як порода, вік, стать, характер та інші. Інформація повинна бути представлена в зрозумілій та легкодоступній форм.

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

1.3 Огляд проблемної області

Притулки для тварин відіграють надзвичайно важливу роль у нашому суспільстві, проте вони стикаються з рядом складних проблем, пов'язаних з бездомними тваринами. Ця проблемна область потребує нашої уваги та негайних рішень.

Першою та найбільш помітною проблемою є бездомні тварини. У багатьох містах і сільських районах можна побачити безпритульних собак, котів та інших тварин, які мандрують по вулицях у пошуках їжі та притулку. Ці тварини часто стикаються з голодом, хворобами та травмами, а також піддаються небезпеці від вуличного життя та недобррозичливих людей. Надання притулку та догляду для бездомних тварин є критично важливим для їхнього благополуччя та безпеки.

Другою проблемою є переповненість притулків. Багато притулків мають обмежену кількість місць, але потік бездомних тварин надто великий, щоб їх всіх помістити. Це часто призводить до переповнення та недостатнього утримання тварин, що негативно впливає на їхній фізичний та психологічний стан. Бездомним тваринам потрібен додатковий простір, догляд та увага, але недостаток ресурсів у притулках ставить під загрозу їхнє благополуччя.

Третя проблема пов'язана з фінансуванням притулків для тварин.

Забезпечення необхідних ресурсів, таких як ветеринарна допомога, харчування, проживання та програми усиновлення, вимагає значних фінансових вкладень. Багато притулків для тварин залежать від благодійних внесків, грантів та пожертвувань, що не завжди є стабільними джерелами доходу. Недостатнє фінансування обмежує їхні можливості та якість надання догляду, що може негативно впливати на життя бездомних тварин.

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

Четверта проблема пов'язана з усиновленням та пошуком нових домівок для тварин. Притулки мають здійснювати активну роботу з просування тварин для усиновлення, але це часто є складним завданням. Необхідно розробляти ефективні механізми пошуку нових власників, проводити рекламні кампанії, брати участь у виставках та організовувати зустрічі з потенційними усиновлювачами. Важливо забезпечити, щоб кожна бездомна тварина мала шанс на знайомство з новою люблячою сім'єю.

Освіта та свідомість також відіграють важливу роль у проблемній області притулків для тварин. Громадська освіта про відповідальне утримання тварин, важливість стерилізації/кастрації, переваги усиновлення та правильного догляду має бути на високому рівні. Це сприятиме зменшенню кількості бездомних тварин та покращенню їхнього добробуту.

Отже, огляд проблемної області притулку для тварин вказує на необхідність зосередитися на забезпеченні притулків з достатніми ресурсами, розвитку програм усиновлення, посиленні фінансування та підвищенні свідомості громади. Лише шляхом спільних зусиль урядов, неприбуткових організацій, волонтерів та громадськості ми зможемо розв'язати ці проблеми і створити кращі умови для бездомних тварин.

Потрібно працювати над покращенням законодавства, забезпеченням фінансової підтримки таресурсів, впровадженням ефективних програм усиновлення та зростанням свідомості громади щодо відповідального ставлення до тварин. Тільки тоді ми зможемо забезпечити безпеку, захист та добробут для всіх тварин, що потребують допомоги та притулку.

В сучасному світі притулки для тварин є важливими установами, які забезпечують тимчасове проживання та догляд за бездомними тваринами. Вони виконують значну соціальну роль, забезпечуючи безпечні умови для тварин та допомагаючи їм знайти новий дім. Однак, з розвитком технологій,

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

все більше притулків починають використовувати web-орієнтовані додатки, щоб полегшити свою роботу та покращити взаємодію з громадськістю.

Огляд подібних рішень дає змогу розглянути й аналізувати існуючі веб-додатки, що використовуються притулками для тварин. Це дозволяє виявити кращі практики, інноваційні рішення та функціональні можливості, які можна впровадити в розробку власного додатка.

Одним з ключових аспектів огляду подібних рішень є аналіз функціональності. Веб-додатки притулків для тварин можуть містити такі модулі, як пошук тварин, реєстрація волонтерів, онлайн-адопція тварин, розміщення оголошень про знайдених тварин, спілкування зі спільнотою та багато іншого. Цей аналіз дозволяє визначити, які функції є найбільш важливими для притулку та його клієнтів.

Другим важливим аспектом є дизайн та користувацький досвід. Успішні веб-додатки притулків мають зручний та привабливий дизайн, який сприяє легкому навігації та взаємодії з користувачем. Вони мають інтуїтивні інтерфейси, що дозволяють швидко знаходити потрібну інформацію та виконувати необхідні дії. Аналізуючи подібні рішення, можна виявити успішні приклади дизайну та користувацького досвіду, які можна використати в розробці власного веб-додатка притулку для тварин.

Загалом, огляд подібних рішень є важливим етапом у розробці веб-додатка притулку для тварин. Він дозволяє поглиблено вивчити функціональність, дизайн та користувацький досвід існуючих додатків, що використовуються в цій сфері. Це дає можливість позичити кращі практики та ідеї, а також уникнути помилок та проблем, які вже були виявлені в інших рішеннях. Огляд подібних рішень є важливим кроком у створенні успішного та ефективного веб-додатка притулку для тварин, який задовольнятиме потреби притулку та користувачів, сприятиме збереженню тваринного

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

благополуччя та сприятиме їх швидкій адопції та знайомству з потенційними власниками.

Сучасний світ невпинно рухається вперед, а це ставить перед нами нові виклики та завдання. Як казав Білл Гейтс «If your business is not on the Internet then your business will be out of business», що в перекладі означає «Якщо ваш бізнес не існує в інтернеті, ваш бізнес не існує взагалі». У такому контексті створення веб-додатку є надзвичайно актуальним завданням, яке відповідає потребам суспільства і сучасних технологій. Для розуміння актуальності такого проекту необхідно розглянути декілька ключових аспектів.

По-перше, веб-додатки є потужним інструментом комунікації та обміну інформацією. У сучасному світі, де Інтернет є неодмінною частиною нашого повсякденного життя, веб-додатки забезпечують зручний та швидкий доступ до інформації та послуг. Вони дозволяють нам здійснювати покупки, отримувати консультації, знайомитися з новинами та багато іншого. Створення веб-додатки притулку для тварин дозволить забезпечити ефективний спосіб комунікації між потенційними власниками тварин та притулком, сприятиме швидкому знаходженню інформації про тварин для адопції та забезпечить зручну інтерактивну платформу для всіх зацікавлених осіб.

По-друге, створення веб-додатку для притулку для тварин сприятиме зменшенню кількості бездомних тварин та покращенню їхнього добробуту. На сьогоднішній день проблема безпритульних тварин стоїть гостро в багатьох країнах. Веб-додаток, який надає зручний спосіб пошуку та адопції тварин, зможе залучити більше зацікавлених осіб, які бажають дати тим безпритульним тваринам другий шанс на щасливе життя. Широкий доступ до інформації про тварин та зручність процесу адопції, що надає веб-додаток, може стимулювати більше людей взяти на себе відповідальність і

					<i>ДТЕУ 121 06-15.БР</i>	Аркуш
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		8

прийняти бездомну тварину у свою родину. Це дозволить зменшити навантаження на притулки та забезпечити кращі умови для тварин, що опинилися в скрутному становищі.

Таким чином, створення веб-додатку для притулку для тварин має велику актуальність у сучасному суспільстві. Це не тільки забезпечить зручний та швидкий доступ до інформації про тварин та можливість їх адопції, але й допоможе вирішити проблему бездомних тварин та поліпшити їхнє життя. Шляхом поєднання сучасних технологій та гуманітарних зусиль, веб-додаток може стати мостом між людьми та тваринами, допомагаючи створити щасливі та довгострокові партнерства.

1.4 Постановка задачі

В сучасному світі з розширенням цифрових технологій та зростанням популярності інтернету все більше сфер діяльності переходять у онлайн середовище. І не є винятком сфера притулків для тварин. Створення веб-додатку для таких притулків є актуальною та важливою задачею, що має на меті полегшити процес усиновлення безпритульних тварин, покращити управління та залучення громадськості до проблеми.

Однією з основних задач створення веб-додатку для притулку є створення централізованої бази даних про безпритульних тварин. Ця база даних має містити інформацію про кожну тварину, що перебуває у притулку, включаючи її характеристики, медичні дані, фотографії та іншу необхідну інформацію. Завдяки цьому, користувачі веб-додатку зможуть швидко знайти та переглянути інформацію про тварин, які доступні для усиновлення. Така централізована база даних значно спростить процес пошуку і вибору тварини

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

для усиновлення, що зробить його більш зручним та ефективним для користувачів.

Ще одна важлива задача полягає у розробці зручного інтерфейсу користувача, який дозволить легко та швидко ознайомитися з доступними тваринами, здійснювати пошук за різними критеріями (наприклад, вид тварини, вік, розмір) та здійснювати заявку на усиновлення. Інтерфейс повинен бути інтуїтивно зрозумілим та дружнім до користувача, щоб ти міг швидко зорієнтуватися на веб-сайті та знайти необхідну інформацію про тварину, а також здійснити заявку на усиновлення в кілька простих кроків. Крім того, веб-додаток може також надавати інші корисні функціональності, такі як підтримка онлайн-консультацій з фахівцями, організація благодійних акцій та збору пожертв для притулку, інформування про події та новини зі світу тваринництва.

Таким чином, створення веб-додатку для притулку для тварин є актуальною задачею, яка допоможе сприяти усиновленню безпритульних тварин, полегшить їхню пошуку та забезпечить зручний інтерфейс для взаємодії з користувачами. Реалізація такого проекту вимагатиме використання сучасних технологій та проектних підходів, але результат буде має велике соціальне значення, допомагаючи тваринам знайти новий дім та людям знайти вірного чотирилапого друга.

1.5 Технічне завдання

Мета. Метою створення системи веб-орієнтованого додатку притуку для тварин є забезпечення ефективного управління та покращення якості життя тварин, що перебувають в притуті, а також сприяння у процесі їх усиновлення

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

та забезпечення фінансової підтримки для розвитку притуку. Головні завдання системи веб-орієнтованого додатку для притуку для тварин включають в себе:

1. Забезпечення зручного та швидкого доступу користувачів до інформації про тварин, що перебувають в притуку.
2. Забезпечення можливості здійснення заявок на усиновлення тварин, що перебувають в притуку, та моніторингу стану заявки.
3. Забезпечення можливості здійснення онлайн-донатів та пожертвувань на розвиток та підтримку роботи притуку.
4. Забезпечення можливості отримання інформації про події та акції, що проводяться притуком.
5. Забезпечення можливості контактування з представниками притуку та отримання консультацій щодо усиновлення тварин та сприяння їхньому благополуччю після усиновлення.
6. Забезпечення адміністраторів притуку зручним та ефективним інструментом для управління тваринами, заявками на усиновлення, пожертвуваннями та іншими аспектами роботи притуку.

Створення системи веб-орієнтованого додатку для притуку для тварин дозволить покращити ефективність та зручність управління притуком, а також сприятиме підвищенню усвідомленості громадськості про проблеми безпритульних тварин.

Вимоги до системи в цілому. Для того, щоб система веб-орієнтованого додатку притуку для тварин була ефективною та корисною для користувачів, необхідно дотримуватись наступних вимог:

1. Система повинна мати зрозумілий та легкий інтерфейс так званий “User friendly”, що дозволяє користувачам швидко та зручно здійснювати потрібні дії та отримувати потрібну інформацію.

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

2. Для забезпечення безпеки користувачів та тварин система повинна мати високий рівень захисту даних, включаючи захист від зламів та незаконного доступу.

3. Система повинна мати повний набір функцій, що дозволяють користувачам здійснювати заявки на усиновлення, пожертвування та отримувати інформацію про тварин.

4. Система повинна бути адаптована для використання на різних мобільних пристроях, що дозволяє користувачам зручно користуватись додатком на будь-якому пристрої.

5. Система повинна бути швидкою та стабільною, що дозволяє користувачам швидко здійснювати необхідні дії та отримувати результати.

6. Система повинна бути легко модифікованою та розширюваною, що дозволяє додавати нові функції та можливості у майбутньому.

Вимоги до структури та функціонування системи, перелік підсистем. Структура та функціонування системи повинна бути добре організованою та логічною.

Для забезпечення ефективності та користувацької зручності, система повинна складатись з наступних підсистем:

1. Підсистема заявок на усиновлення

Ця підсистема повинна дозволяти користувачам заповнювати та відправляти заявки на усиновлення тварин. Підсистема повинна містити форми для введення даних про користувача та тварину, яку він хоче усиновити. Для забезпечення ефективності, підсистема повинна мати функції, що дозволяють відслідковувати статус заявки.

2. Підсистема донатів

Ця підсистема повинна дозволяти користувачам здійснювати пожертвування для підтримки роботи притуку. Підсистема повинна містити форму для введення даних про користувача та обрану суму пожертвування.

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

Для забезпечення ефективності, підсистема повинна мати функції, що дозволяють відслідковувати здійснені пожертвування та відправляти дякувальні повідомлення.

3. Підсистема пошуку тварин

Ця підсистема повинна дозволяти користувачам шукати тварин, яких вони хочуть усиновити. Підсистема повинна мати фільтри для пошуку тварин за різними параметрами, такими як порода, вік, стать та інші.

4. Підсистема опису тварин

Ця підсистема повинна містити інформацію про кожну тварину в притуці, таку як порода, вік, стать, характер та інші. Інформація повинна бути представлена в зрозумілій та легкодоступній форм

Вимоги до способів і засобів інформаційного обміну між компонентами системи

Одним з ключових аспектів веб-орієнтованого додатку притука для тварин є спосіб та засоби інформаційного обміну між його компонентами. Нижче наведені вимоги до способів та засобів інформаційного обміну між компонентами системи:

1. Комунікація між сервером та клієнтом повинна бути захищена протоколом HTTP або HTTPS.
2. Система повинна підтримувати REST API для отримання та оновлення даних.
 1. Для передачі структурованих даних використовуються формати, такі як JSON або XML.
 2. Для забезпечення безпеки обміну інформацією, використовується шифрування даних засобами SSL / TLS.
 3. Для зменшення навантаження на сервер, використовується кешування даних, що дозволяє уникнути повторних запитів до бази даних.

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

Для забезпечення ефективного інформаційного обміну між компонентами системи, необхідно враховувати вимоги до швидкості та надійності передачі даних. Для цього можуть використовуватись різні технології, такі як кешування даних, розподілена обробка запитів, масштабованість та інші.

Вимоги до режимів функціонування системи. Вимоги до режимів функціонування системи - це вимоги до того, як система повинна працювати в різних ситуаціях та умовах використання. Вони описують режими, у яких система повинна забезпечувати певний функціонал та виконувати певні дії. Нижче наведені деякі вимоги до режимів функціонування системи веб-орієнтованого додатку приюту для тварин:

1. Режим нормальної роботи - система повинна працювати без збоїв та забезпечувати коректну роботу всіх її компонентів та функціоналу.
2. Режим відновлення після збоїв - система повинна мати механізми відновлення після збоїв та забезпечувати безперервну роботу після відновлення.
3. Режим захисту від злому - система повинна мати механізми захисту від злому та забезпечувати безпеку користувачів та їхніх даних.
4. Режим резервного копіювання - система повинна мати механізми резервного копіювання даних та забезпечувати можливість відновлення після випадку втрати даних.
5. Режим обмеженого доступу - система повинна мати механізми обмеження доступу до певних функціональних можливостей та даних відповідно до прав користувачів.
6. Режим масштабованості - система повинна мати можливість масштабування своєї роботи при збільшенні обсягу даних та навантаження.

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

7. Режим тестування - система повинна мати механізми тестування для перевірки коректності та стійкості її роботи в різних умовах та ситуаціях.

8. Режим моніторингу - система повинна мати механ

Вимоги до режимів управління системою. Вимоги до режимів управління системою включають у себе наступні вимоги:

1. Можливість управління доступом до функцій системи, вона повинна мати можливість обмежувати доступ користувачів до окремих функцій залежно від їхніх прав доступу.
2. Можливість зміни налаштувань системи, вона повинна мати можливість змінювати налаштування в залежності від потреб користувачів та адміністраторів.
3. Можливість резервного копіювання та відновлення системи, вона повинна мати можливість створювати резервні копії даних та налаштувань, а також відновлювати їх у разі необхідності.
4. Можливість моніторингу роботи системи вона повинна мати можливість відстежувати роботу різних компонентів та виявляти та вирішувати проблеми, які виникають.
5. Можливість забезпечення безпеки системи вона повинна мати вбудовані механізми захисту від зловмисників, такі як захист від вторгнень, аутентифікація користувачів та шифрування даних.
6. Можливість оновлення системи вона повинна мати можливість оновлюватися до нових версій, які можуть включати нові функції та виправляти помилки.
7. Можливість відновлення системи після аварії вона повинна мати можливість відновлюватися після аварій, що можуть відбутися через системні помилки або зовнішні впливи.

					<i>ДТЕУ 121 06-15.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

8. Можливість масштабування системи вона повинна мати можливість масштабуватися для забезпечення потреб користувачів в залежності від збільшення навантаження на систему.

Показники призначення. Показники призначення - це вимоги та критерії, які повинна виконувати система для того, щоб вона була ефективною та задовольняла потреби користувачів та бізнесу. Основними показниками призначення можуть бути наступні:

1. Функціональність - система повинна мати необхідний функціонал для виконання поставлених перед нею завдань.
2. Надійність - система повинна бути стійкою до помилок та аварій.
3. Ефективність - система повинна працювати швидко та ефективно, відповідати на запити користувачів у максимально короткий час.
4. Масштабованість - система повинна мати можливість розширення та масштабування в разі збільшення навантаження на неї.
5. Безпека - система повинна бути захищена від несанкціонованого доступу та злому.
6. Зручність використання - система повинна бути простою та зрозумілою для користувачів.
7. Сумісність - система повинна бути сумісною з різними платформами та браузерами.
8. Підтримка - система повинна мати можливість швидко та ефективно вирішувати проблеми користувачів та надавати підтримку у разі потреби.
9. Відкритість - система повинна мати відкритий код та можливість розробки розширень.

Ці показники призначення допомагають забезпечити ефективну та якісну роботу системи, що задовольняє потреби користувачів та бізнесу.

				ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	8

Параметри, що характеризують ступінь відповідності системи призначенням. Параметри, що характеризують ступінь відповідності системи призначенням, можуть включати:

1. **Функціональність:** система повинна виконувати всі функції, необхідні для досягнення поставлених цілей. До функціональних вимог можуть відноситись такі показники, як точність виконання операцій, швидкість відповіді на запити, наявність необхідних функцій і можливостей для користувачів.
2. **Ефективність:** система повинна працювати з достатньою швидкістю та продуктивністю, щоб задовольняти потреби користувачів. Це можуть бути показники, такі як швидкість завантаження сторінок, розмір бази даних, кількість запитів на сервер, масштабованість системи.
3. **Надійність:** система повинна працювати без збоїв та помилок, щоб уникнути втрат даних або інших проблем. Це можуть бути показники, такі як кількість збоїв системи, час відновлення після збоїв, забезпечення доступності інформації.
4. **Безпека:** система повинна захищати конфіденційні дані користувачів та іншу важливу інформацію. Це можуть бути показники, такі як рівень шифрування даних, рівень автентифікації користувачів, захист від хакерських атак.
5. **Сумісність:** система повинна працювати з іншими програмними засобами та пристроями, щоб забезпечити широку сумісність інтерфейсів та форматів даних.
6. **Управління:** система повинна бути простою в управлінні та підтримці. Це можуть бути показники, такі як час на розгортання та налаштування системи, рівень складності

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

Вимоги до збереження працездатності системи в різних ймовірних умовах.

Вимоги до збереження працездатності системи в різних умовах можуть включати:

1. Вимоги до надійності: система повинна функціонувати стабільно та надійно в умовах, що можуть виникати під час роботи, наприклад, збої електропостачання, помилки в програмному забезпеченні, атаки з зовнішнього середовища.
2. Вимоги до резервування: система повинна мати механізми резервного копіювання та відновлення в разі виникнення збоїв або випадків аварійної зупинки.
3. Вимоги до масштабованості: система повинна бути здатною до масштабування для забезпечення необхідної продуктивності в разі збільшення обсягів оброблюваних даних або навантаження.
4. Вимоги до тестування: система повинна бути піддається тестуванню та перевірці працездатності в різних умовах, що можуть виникати під час роботи, наприклад, випадки, коли навантаження на систему збільшується.
5. Вимоги до реакції на аварійні ситуації: система повинна мати механізми автоматичної реакції на аварійні ситуації для запобігання втраті даних або іншим наслідкам.
6. Вимоги до підтримки: система повинна забезпечувати підтримку користувачів та адміністраторів, включаючи надання документації та вирішення проблем, що виникають під час роботи системи.
7. Вимоги до безпеки: система повинна забезпечувати безпеку даних та захист від зловмисних атак з зовнішнього середовища.

Вимоги до надійності. Зважаючи на те, що веб-орієнтований додаток прийому для тварин є важливим інструментом допомоги тваринам та людям,

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

які з ними працюють, надійність є ключовим фактором для його успішної реалізації. Ось деякі вимоги до надійності, які можуть бути включені в дипломну роботу:

1. **Доступність:** додаток повинен бути доступним для користувачів протягом визначеного часу. Для досягнення цього можуть бути використані різні стратегії, такі як резервне копіювання, резервне забезпечення енергопостачання, обмеження трафіку та т.д.
2. **Надійність транзакцій:** додаток повинен гарантувати, що транзакції (наприклад, процес оплати або бронювання) будуть оброблені правильно та в повному обсязі. Для цього можуть бути використані різні методи, такі як механізми перевірки на цілісність даних, відновлення збоїв тощо.
3. **Захист від кібератак:** додаток повинен бути захищеним від кібератак та інших загроз з боку злоумисників. Для досягнення цього можуть бути використані різні методи, такі як захист від DDOS-атак, шифрування даних та ідентифікації користувачів.
4. **Швидкість роботи:** додаток повинен працювати швидко та ефективно, щоб користувачі могли отримати швидкий доступ до необхідної інформації. Для досягнення цього можуть бути використані різні техніки, такі як кешування даних та оптимізація запитів до бази даних.

Перелік підсистем системи. В даній роботі існують наступні підсистеми системи:

1. Система управління базою даних (СУБД) - це програмне забезпечення, яке дозволяє створювати, зберігати, оновлювати та отримувати доступ до даних в базі даних. СУБД є ключовим елементом будь-якої системи, яка використовує базу даних, такої як веб-орієнтований додаток для приюту тварин.

					<i>ДТЕУ 121 06-15.БР</i>	Аркуш
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		8

СУБД має наступні функції:

- Створення бази даних та її таблиць;
- Зберігання даних в базі даних;
- Оновлення даних в базі даних;
- Видалення даних з бази даних;
- Організацію доступу до даних у базі даних;
- Запити до бази даних для отримання потрібної інформації;
- Захист бази даних від несанкціонованого доступу.

Для веб-орієнтованого додатку приюту тварин СУБД є ключовим компонентом, оскільки додаток має велику кількість даних, які потрібно зберігати та оновлювати. Наприклад, СУБД може використовуватися для зберігання інформації про тварин, які перебувають у приюті, їхнього стану здоров'я, вакцинації та взаємодії зі співробітниками приюту.

2. Веб-сервер - це програмне забезпечення, що надає можливість відображення веб-сторінок у веб-браузері. Веб-сервер виконує дві основні функції: прийняття запитів від клієнтів та надання їм відповідей у вигляді HTML-сторінок, зображень, аудіо- та відеофайлів, а також інших ресурсів; збереження та обробка даних, які надходять від клієнтів через форми, електронні листи та інші інтерактивні елементи на веб-сторінках.

Основні вимоги до веб-сервера полягають у надійності, швидкодії та безпеці. Веб-сервер повинен бути достатньо стійким до збоїв та забезпечувати високу швидкість обробки запитів, щоб забезпечити швидкий доступ до веб-сторінок та ресурсів для користувачів. Також веб-сервер повинен мати надійний захист від зловмисних атак, таких як

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

хакерські атаки та DDoS-атаки, і забезпечувати безпеку даних, які передаються між клієнтом та сервером.

3. Модулі взаємодії з соціальними мережами та іншими засобами масової комунікації є складовою частиною веб-орієнтованого додатку, яка дозволяє користувачам взаємодіяти з додатком за допомогою різних засобів комунікації. Ці модулі можуть включати функції інтеграції з різними соціальними мережами, такими як Facebook, Twitter, Instagram тощо, а також іншими засобами масової комунікації, наприклад, електронною поштою або месенджерами.

Основна мета модулів взаємодії з соціальними мережами та іншими засобами масової комунікації - розширення можливостей взаємодії користувачів з додатком та підвищення рівня зручності та ефективності взаємодії. Наприклад, можна додати можливість авторизації користувачів через обліковий запис у соціальній мережі, що дозволить їм швидко та зручно увійти до системи, без необхідності створювати окремий обліковий запис в додатку. Також можна додати функцію поширення інформації про додаток у соціальних мережах або направлення повідомлень через месенджери, що дозволить привернути більше уваги до додатку та збільшити його популярність.

Вимоги до видів забезпечення. Вимоги до видів забезпечення зазвичай включають вимоги до апаратного та програмного забезпечення, мережевих засобів, засобів збереження даних та інших компонентів системи. Основні вимоги до видів забезпечення веб-орієнтованого додатку приюту для тварин можуть включати наступне:

1. Апаратне забезпечення: Додаток повинен працювати на комп'ютерах, які відповідають певним технічним вимогам (наприклад, наявність

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

достатньої кількості оперативної пам'яті, потужного процесора та достатньої кількості місця на жорсткому диску).

2. Програмне забезпечення: Додаток повинен працювати на певному програмному забезпеченні, включаючи операційну систему, веб-сервер, СУБД та інші компоненти.
3. Мережеві засоби: Додаток повинен працювати на певному мережевому обладнанні та протоколах для забезпечення стабільного підключення до Інтернету та мережевої інфраструктури.
4. Засоби збереження даних: Додаток повинен забезпечувати збереження та безпеку даних, включаючи бекапи, резервне копіювання та захист від несанкціонованого доступу.
5. Інші компоненти: Додаток може включати інші компоненти, такі як засоби для збору даних, засоби аналізу даних та інші засоби, які необхідні для правильної роботи системи.

Важливо забезпечити сумісність всіх компонентів системи та їх відповідність технічним вимогам для забезпечення надійності та стабільності роботи додатку.

Вимоги до програмного забезпечення. Для ефективної реалізації веб-орієнтованого додатку необхідно враховувати наступні вимоги до програмного забезпечення:

1. Мова програмування: для розробки front-end використовується такі мови як HTML, CSS та JavaScript, а для розробки back-end можуть використовуватись Python, PHP та JavaScript(Node.js).

2. База даних: необхідно вибрати підходящу систему управління базами даних, найбільш популярні це MySQL, PostgreSQL, MongoDB, SQLite, в своєму проєкті я буду використовувати MySQL.

					<i>ДТЕУ 121 06-15.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

3. В моєму проєкті це не застосовується, але у великих є обов'язковим присутність фреймворків, такі як React, Vue та Angular.

Фреймворки в JavaScript є наборами зручних інструментів, бібліотек, методологій та стандартів, які допомагають розробникам швидко створювати складні веб-додатки з меншими зусиллями та з меншою кількістю помилок. Фреймворки надають готові рішення для часто виникаючих завдань, таких як маршрутизація, валідація даних, робота з базами даних, робота зі стилізацією, робота зі зображеннями та багато інших. Використання фреймворків дозволяє розробникам економити час на вирішенні повсякденних задач та зосереджуватися на реалізації основної логіки додатку.

Вимоги до технічного забезпечення. Основні вимоги до технічного забезпечення веб-орієнтованого додатку приюта для тварин можуть включати такі показники:

1. **Продуктивність:** додаток повинен працювати достатньо швидко, щоб користувачі могли взаємодіяти з ним без зайвої затримки. Показник продуктивності може включати швидкість завантаження сторінок, швидкість обробки запитів, максимальне число користувачів, яких може обслуговувати додаток одночасно і т.д.
2. **Масштабованість:** додаток повинен бути здатний збільшуватись у розмірах, щоб відповідати зростаючим потребам користувачів. Це може включати можливість додавання нових функцій, підтримку більшого числа користувачів, збільшення кількості серверів і т.д.
3. **Безпека:** додаток повинен бути захищений від зловмисників та небажаних вторгнень. Це може включати шифрування даних, захист від SQL-ін'єкцій та інших атак, контроль доступу до ресурсів, захист від вірусів і т.д.

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

4. Надійність: додаток повинен працювати безперебійно протягом тривалого часу. Це може включати контроль помилок, моніторинг працездатності, резервне копіювання даних, автоматичну відновлення після аварії і т.д.
5. Сумісність: додаток повинен бути здатний працювати на різних пристроях та платформах, таких як ПК, мобільні пристрої, планшети і т.д. Це може включати адаптивний дизайн, розширені можливості взаємодії з різними пристроями і т.д.
6. Доступність: додаток повинен бути доступн

1.6 Висновок до Розділу 1:

У данному розділі розглянуто актуальність створення веб-додатку для притулку для тварин та поставлено перед собою завдання розробити такий сервіс. Було обговорено основні проблеми, з якими зіштовхуються притулки тварин, такі як висока кількість безпритульних тварин, складність їх пошуку нових домів, та необхідність забезпечення зручного інтерфейсу для взаємодії з користувачами.

Було розглянуто можливі рішення, включаючи створення веб-додатку з базою даних про тварин та функціоналом для пошуку, резервування та усиновлення тварин. Такий веб-додаток забезпечить зручний доступ до інформації про тварин, допоможе їм знайти нові домівки та сприятиме їх усиновленню. Крім того, сервіс може включати різноманітні функціональні можливості, такі як онлайн-консультації, благодійні акції та інформування про події в світі тваринництва.

Створення веб-додатку для притулку для тварин є важливою задачею, яка вирішує проблеми безпритульних тварин та полегшує їх пошук нових

					<i>ДТЕУ 121 06-15.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

домівок. Реалізація такого проекту потребує використання сучасних технологій та проектних підходів, але вона має велике соціальне значення, допомагаючи тваринам знайти новий дім та людям знайти вірного чотирилапого друга.



					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

РОЗДІЛ 2

ОГЛЯД ТЕХНОЛОГІЙ ТА ІНСТРЕМНТІВ

2.1. Інструменти розробки

Проведемо огляд технологій та інструментів, які можуть бути використані для розробки web-орієнтованого додатка притулку для тварин. Нижче наведено короткий огляд деяких ключових аспектів, які слід розглянути:

а) Front-End технології: Для створення користувацького інтерфейсу додатка можна використовувати різні front-end технології, такі як HTML, CSS і JavaScript. Фреймворки, наприклад React або Angular, можуть також бути корисними для покращення ефективності розробки та взаємодії з користувачем.

б) Back-End технології: Для обробки логіки бізнес-процесів, зберігання даних та взаємодії з базою даних необхідно вибрати підходящі back-end технології. Популярні варіанти включають мови програмування, такі як Python, Java або Node.js, а також фреймворки, наприклад Django або Ruby on Rails.

в) Бази даних: Вибір підходящої бази даних є важливим кроком у розробці додатка. Реляційні бази даних, такі як MySQL або PostgreSQL, часто використовуються для зберігання структурованої інформації. Нереляційні бази даних, наприклад MongoDB або Redis, можуть бути корисними для зберігання неструктурованої або швидкодіючої інформації.

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 06-15.БР</i>			
Зав. каф.		Криворучко О.В.		03.03.23	Веб-орієнтований додаток притуку для тварин	Стадія	Аркуш	Аркушів
Керівник		Котенко Н. О..		03.03.23		P2	10	50
Гарант		Котенко Н.О.		03.03.23		Факультет інформаційних технологій		
Розробив		Компанець Д. О..		03.03.23		4 курс, 6 група		
					Вибір програмних засобів та проєктування мобільного додатку			

2.2 Front-end

Front-end частина web-орієнтованого додатку притулку для тварин, є ефективним та привабливим користувацьким інтерфейсом, який буде легким у використанні та надасть зручну взаємодію користувачам з притулком. Нижче наведено деякі ключові аспекти, які варто розглянути при розробці Front-end частина web-орієнтованого додатку притулку для тварин:

а) Дизайн та вигляд: Важливо створити привабливий та сучасний дизайн для мого додатка. Розгляньте використання естетичних кольорових схем, зручного шрифту, привабливих графічних елементів та зображень, щоб зробити мій додаток привабливим для користувачів.

б) Інтерфейс та навігація: Розробіть зрозумілий та інтуїтивний інтерфейс, який дозволить користувачам легко орієнтуватися в додатку та здійснювати необхідні дії. Використовуйте зрозумілі та логічні меню, кнопки та посилання, щоб допомогти користувачам знаходити потрібну інформацію та функціонал.

в) Функціонал для користувачів: Розгляньте, які функції та можливості ви хочете надати користувачам мого додатка. Наприклад, ви можете реалізувати можливість перегляду списку тварин, пошуку за різними критеріями, відображення деталей про тварин, можливість здійснення онлайн-донатів та інше. Забезпечте, щоб цей функціонал був зрозумілим, простим у використанні та зручним для користувачів.

г) Взаємодія з базою даних.

HTML. HTML (HyperText Markup Language — «мова гіпертекстової розмітки») – базовий будівельний блок Інтернету. Він визначає зміст та структуру веб-контенту. Інші технології, крім HTML, зазвичай використовуються для описузовнішнього вигляду/подання (CSS) або функціональності/поведінки (JavaScript) веб-сторінки. Під гіпертекстом

						Аркуш
						31
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-15.БР	

("hypertext") розуміються посилання, які з'єднують веб-сторінки один з одним або в межах одного веб-додатку або між веб-додатками. Посилання є фундаментальним аспектом Інтернету. Завантажуючи контент в Інтернет і пов'язуючи його зі сторінками, створеними іншими людьми, ви стаєте активним учасником Всесвітньої мережі.

HTML використовує розмітку ("markup") для відображення тексту, зображень та іншого контенту у веб-браузері. HTML-розмітка включає спеціальні "елементи", такі як <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, , , <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video> та багато інших.

HTML-елемент виділяється з іншого тексту в документі за допомогою "тегів", які складаються з імені елемента оточеного "<" та ">". Ім'я елемента всередині тега не чутливе до регістру. Тобто воно може бути написано у верхньому або нижньому регістрі, або змішане. Наприклад, тег <title> може бути записаний як <Title>, <TITLE> або будь-яким іншим способом.

CSS. CSS (Cascading Style Sheets) — це код, який ви використовуєте для стилізації веб-сторінки. Основи CSS допоможуть вам зрозуміти, що потрібно для початку роботи. Ми відповімо на такі питання як: Як зробити мій текст чорним чи червоним? Як зробити так, щоб контент з'являвся на певному місці на екрані? Як прикрасити мою веб-сторінку за допомогою фонових зображень та кольорів?

Так що таке CSS?

Як і HTML, CSS насправді є мовою програмування. Це не мова розмітки – це мова таблиці стилів. Це означає, що він дозволяє застосовувати стилі вибірково до елементів у документах HTML. Наприклад, щоб вибрати всі елементи абзацу на сторінці HTML і змінити текст всередині них з чорного на червоний, ви повинні написати цей CSS:

```
p {
```

						ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			32

color: red;

Анатомія набору правил CSS. Вся структура називається набором правил (але найчастіше для стислості "правило"). Зазначимо також імена окремих частин:

Селектор (Selector)

Ім'я HTML-елемента на початку набору правил. Він вибирає елемент(и) для застосування стилю (в даному випадку елементи `p`). Для стилізації іншого елемента просто змініть селектор.

Оголошення (Declaration)

Єдине правило, наприклад, `color: red;` вказує, які властивості елемента ви хочете стилізувати.

Властивості (Properties)

Способи, якими можна стилізувати певний HTML-елемент (у разі, `color` є властивістю для елементів `<p>`). У CSS ви вибираєте, які властивості ви хочете торкнутися у вашому правилі.

Значення якості (Property value)

Праворуч від властивості, після двокрапки, у нас є значення властивості, яке вибирає одну з безлічі можливих ознак для даної властивості (є безліч значень `color`, крім `red`).

					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		33

JavaScript. JavaScript® (часто просто JS) – це легковажна, інтерпретована або JIT-компільована, об'єктно-орієнтована мова з функціями першого класу. Найширше застосування знаходить мову сценаріїв веб-сторінок, але також використовується і в інших програмних продуктах, наприклад, node.js або Apache CouchDB. JavaScript це прототипно-орієнтована, мультипарадигменна мова з динамічною типізацією, яка підтримує об'єктно-орієнтовану, імперативну та декларативну (наприклад, функціональне програмування) стилі програмування. Докладніше про JavaScript.

Стандартом мови JavaScript є ECMAScript. Станом на 2012 рік усі сучасні браузерери повністю підтримують ECMAScript 5.1. Старі версії браузерів підтримують принаймні ECMAScript 3. 17 червня 2015 року відбувся випуск

шостої версії ECMAScript. Ця версія офіційно називається ECMAScript 2015, яку найчастіше називають ECMAScript 2015 чи просто ES2015. З недавнього часу стандарти ECMAScript випускаються щороку. Ця документація відноситься до останньої версії чернетки, якою є ECMAScript 2018.

Не слід плутати JavaScript з мовою програмування Java. І "Java", і "JavaScript" є торговими марками або зареєстрованими торговими марками Oracle у США та інших країнах. Однак, у обох мов різний синтаксис, семантика та застосування.

Що може JavaScript у браузері?

Сучасний JavaScript - це "безпечна" мова програмування. Він не надає низькорівневий доступ до пам'яті або процесора, тому що спочатку був створений для браузерів, які цього не вимагають. Можливості JavaScript сильно залежать від оточення, де він працює. Наприклад, Node.JS підтримує функції читання/запису довільних файлів, виконання мережевих запитів тощо.

						Аркуш
					ДТЕУ 121 06-15.БР	34
Зм.	Аркуш	№ докум	Підпис	Дата		

У браузері для JavaScript є все, що пов'язано з маніпулюванням веб-сторінками, взаємодією з користувачем і веб-сервером.

Наприклад, у браузері JavaScript може:

Додавати новий HTML-код на сторінку, змінювати існуючий вміст, модифікувати стилі.

Реагувати на дії користувача, натискання миші, переміщення вказівника, натискання клавіш.

Надсилати мережеві запити на віддалені сервери, завантажувати та завантажувати файли (технології AJAX та COMET).

Отримувати та встановлювати куки, ставити запитання відвідувачу, показувати повідомлення.

Запам'ятовувати дані за клієнта («local storage»).

Чого НЕ може JavaScript у браузері?

Можливості JavaScript у браузері обмежені задля безпеки користувача. Мета полягає в запобіганні доступу недобросовісної веб-сторінки до особистої інформації або нанесення шкоди користувачам.

Приклади таких обмежень включають:

JavaScript на веб-сторінці не може читати/записувати довільні файли на жорсткому диску, копіювати або запускати програми. Він не має прямого доступу до системних функцій ОС.

Сучасні браузери дозволяють йому працювати з файлами, але з обмеженим доступом, і надають його, тільки якщо користувач виконує певні дії, такі як перетягування файлу у вікно браузера або його вибір за допомогою тега <input>.

JavaScript може легко взаємодіяти з сервером, з якого надійшла поточна сторінка. Але його здатність отримувати дані з інших сайтів/доменів обмежена. Хоча це можливо в принципі, для чого потрібна явна згода

						Аркуш
					ДТЕУ 121 06-15.БР	35
Зм.	Аркуш	№ докум	Підпис	Дата		

(виражена в заголовках HTTP) із віддаленою стороною. Знову ж таки, це обмеження безпеки.

2.3 Back-end

Back-end розробка веб-додатків є невід'ємною частиною процесу створення комплексних та функціональних веб-систем. Вона займається розробкою та підтримкою серверної частини додатків, яка відповідає за обробку даних, взаємодію з базою даних, забезпечення безпеки та виконання бізнес-логіки. Основна мета back-end розробки полягає в створенні надійного, масштабованого та ефективного серверного середовища, яке забезпечує відповідну функціональність веб-додатку.

Для реалізації back-end частини веб-додатка використовуються різноманітні технології, мови програмування та фреймворки. Основні ключові аспекти back-end розробки включають:

а) Мови програмування: Існує багато мов програмування, які використовуються для back-end розробки. Деякі з найпопулярніших мов включають Java, Python, PHP, Ruby, C#, та Node.js (заснований на JavaScript). Кожна мова має свої особливості та переваги, і вибір мови залежить від вимог проекту та власних вподобань розробника.

б) Фреймворки: Фреймворки - це набір інструментів та бібліотек, які спрощують розробку back-end додатків. Вони надають готові рішення для роботи з базами даних, маршрутизації, обробки запитів та інше. Популярні фреймворки включають Django для Python, Ruby on Rails для Ruby, Spring для Java, Laravel для PHP, та Express.js для Node.js.

в) Бази даних: Back-end додатки зазвичай використовують бази даних для збереження та керування даними. Популярні бази даних включають MySQL, PostgreSQL, MongoDB, та SQLite. Вибір бази даних для back-end розробки залежить від вимог проекту та специфіки даних, які потрібно

зберігати та обробляти. Основні типи баз даних, що використовуються в back-end розробці, включають реляційні бази даних та нереляційні (NoSQL) бази даних.

Бази даних. Бази даних є важливою складовою будь-якого інформаційного системи і використовуються для зберігання, організації та управління великими

обсягами даних. Вони дозволяють ефективно зберігати і отримувати доступ до інформації, забезпечуючи структуру, цілісність і безпеку даних.

У розробці back-end додатків існує багато різних баз даних, які можна використовувати, і вибір підходящої бази даних залежить від вимог проекту, типу даних та потреб роботи з ними. Давайте розглянемо деякі з найпоширеніших типів баз даних:

а) Реляційні бази даних (Relational Databases): Це тип бази даних, де дані організовані у вигляді таблиць зі зв'язками між ними. Таблиці складаються з рядків і стовпців, і використовуються мови SQL (Structured Query Language) для роботи з даними. Деякі з найпопулярніших реляційних баз даних включають MySQL, PostgreSQL, Oracle та Microsoft SQL Server.

б) Нереляційні бази даних (NoSQL Databases): Цей тип бази даних використовується для зберігання та обробки великих обсягів неструктурованих даних. Вони не використовують таблиці зі зв'язками, а замість цього використовують інші структури даних, такі як ключ-значення, документи, графи або стовпчасті схеми. Деякі популярні нереляційні бази даних включають MongoDB, Cassandra, Redis та Elasticsearch.

в) Колоночні бази даних (Columnar Databases): Цей тип бази даних спеціалізується на роботі з великими обсягами даних і ефективному зберіганні колоночних даних. Вони орієнтовані на операції аналітики та звітності, де важливим є швидкий доступ до певних колонок даних. У таких базах даних дані зберігаються по колонах, а не по рядках, що

						ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			37

дозволяє ефективно виконувати операції агрегації, фільтрації та стискування даних. Прикладами колоночних баз даних є Apache Cassandra, Amazon Redshift та Apache HBase.

г) Інтернет-орієнтовані бази даних: Цей тип баз даних спеціалізується на забезпеченні швидкого доступу до даних з веб-додатків. Вони підтримують велику кількість одночасних підключень і оптимізовані для роботи з розподіленими системами. Інтернет-орієнтовані бази даних дозволяють ефективно обробляти запити на основі HTTP-протоколу, такі як CRUD (створення, читання, оновлення, видалення), і підтримують масштабування для забезпечення високої доступності та продуктивності. Приклади таких баз даних включають Firebase Realtime Database, CouchDB та Riak.

д) Об'єктно-реляційні бази даних (Object-Relational Databases): Цей тип баз даних комбінує можливості реляційних баз даних з об'єктно-орієнтованою моделлю програмування. Вони дозволяють зберігати складні об'єкти, такі як класи, об'єкти та спадковість, у базі даних. Об'єктно-реляційні бази даних часто використовуються в областях, де важлива нормалізація даних та забезпечення цілісності. Деякі приклади об'єктно-реляційних баз даних включають PostgreSQL, Oracle Database та MySQL (з використанням деяких розширень, таких як MySQL with InnoDB).

Вибір бази даних для вашого проекту залежить від специфіки ваших вимог та враховує фактори, такі як тип даних, масштаб проекту, швидкодія, безпека, розширюваність та вартість. Перед прийняттям рішення щодо вибору бази даних варто провести аналіз і ретельно зрозуміти особливості та переваги кожного типу бази даних.

PHP. PHP (рекурсивний акронім словосполучення PHP: Hypertext Preprocessor) - це поширена мова програмування загального призначення з відкритим кодом. PHP спеціально сконструйований для веб-розробок, і його код може впроваджуватися безпосередньо в HTML.

						ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			38

Проста відповідь, але що вона може означати? Ось приклад коду:

Приклад #1 Приклад програмування на PHP

```
<!DOCTYPE html>
<html>
<head>
<title>Приклад</title>
</head>
<body>
<?php
    echo "Привіт, я - скрипт PHP!";
?>
</body>
</html>
```

Замість рутинного виведення HTML-коду командами мови (як це відбувається, наприклад, Perl або C), скрипт PHP містить HTML з вбудованим кодом (у нашому випадку, це виведення тексту "Привіт, я - скрипт PHP!"). Код PHP відокремлюється спеціальними початковими та кінцевими тегами <?php і ?>, які дозволяють "перемикатися" в "PHP-режим" і виходити з нього.

PHP відрізняється від JavaScript тим, що PHP-скрипти виконуються на сервері та генерують HTML, який надсилається клієнту. Якби у вас на сервері був розміщений скрипт, подібний до вищенаведеного, клієнт отримав би тільки результат його виконання, але не зміг би з'ясувати, який саме код його зробив.

Ви навіть можете налаштувати свій сервер таким чином, щоб звичайні HTML-файли оброблялися процесором PHP, тому клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання скрипту.

PHP вкрай простий для освоєння, але водночас здатний задовольнити запити професійних програмістів. Не лякайтеся довгого списку можливостей PHP. Ви

можете швидко розпочати, і вже протягом кількох годин зможете створювати прості PHP-скрипти.

Хоча PHP, головним чином, призначений для роботи в середовищі веб-серверів, сфера його застосування не обмежується тільки цим. Читайте далі і не пропустіть розділ Можливості PHP або, почніть безпосередньо з вступного посібника, якщо вас цікавить виключно веб-програмування.

2.4 Висновок до Розділу 2:

У другому розділі проведено огляд технологій та інструментів, які можуть бути використані для розробки web-орієнтованого додатка притулку для тварин. Ми розглянули ключові аспекти, такі як front-end технології, back-end технології та вибір бази даних.

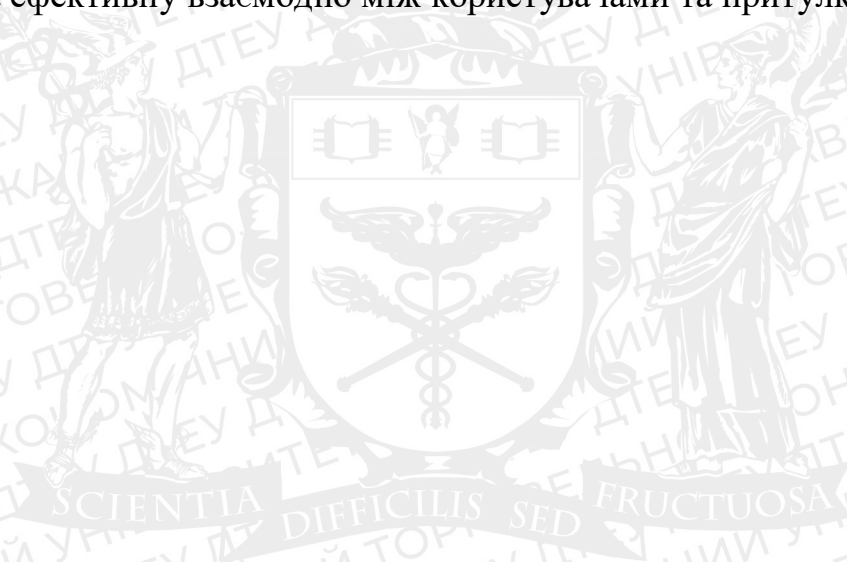
У контексті front-end технологій, визначено, що використання HTML, CSS і JavaScript є основою для створення користувацького інтерфейсу. Крім того, фреймворки, такі як React або Angular, можуть підвищити ефективність розробки та взаємодії з користувачем.

У відношенні back-end технологій, було вказано, що мови програмування, такі як Python, Java або Node.js, разом з фреймворками, наприклад Django або Ruby on Rails, можуть бути використані для обробки логіки бізнес-процесів, зберігання даних та взаємодії з базою даних.

Щодо вибору бази даних, зазначено, що реляційні бази даних, такі як MySQL або PostgreSQL, часто використовуються для зберігання структурованої інформації, тоді як нереляційні бази даних, такі як MongoDB або Redis, можуть бути корисними для зберігання неструктурованої або швидкодіючої інформації.

Загалом, огляд технологій та інструментів показав, що існує багато варіантів для розробки web-орієнтованого додатка притулку для тварин. Вибір

конкретних технологій та інструментів повинен базуватися на потребах проекту, масштабі та функціональних вимогах. Наступним кроком є подальше дослідження та конкретизація вибору технологій та інструментів у контексті розробки конкретного веб-додатка притулку для тварин. Правильно підібрані технології та інструменти допоможуть забезпечити ефективну розробку, високу продуктивність та зручний користувацький досвід. Враховуючи специфіку притулку для тварин і потреби його користувачів, важливо зробити обґрунтований вибір технологій та інструментів, що дозволить створити веб-сервіс, який буде забезпечувати зручну та ефективну взаємодію між користувачами та притулком для тварин.



					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		41

РОЗДІЛ 3

РОЗРОБКА ВЕБ -ОРІЄНТОВАНОГО ДОДАТКУ ПРИТУЛКУ ДЛЯ ТВАРИН

3.1. Концепція веб-додатку

Перш за все перед розробкою web-додатку треба сформулювати базові ідеї і гіпотези майбутнього сайту. Після цього йде моделювання сайту.

Першим кроком в розробці над сайтом це визначення цілей самого сайту.

Для свого сайту я визначив наступні цілі:

1. Представлення тварин. Додаток повинен надавати можливість притулку відображати інформацію про тварин, які знаходяться в притулку.

2. Усиновлення тварин. Головна мета притулку – знайти домівки для бездомних тварин. Додаток повинен надати можливість здійснити процес усиновлення через веб додаток

3. Просування сайту в мережі. Зараз люди знаходяться в інтернеті майже весь свій час і мета веб додатку, щоб люди сидячі в інтернеті натрапили на сторінку і побачили, що деяким тваринам потрібні домівки

3.2 Розробка веб-додатку

3.2.1 Макет та прототип вебдодатку засобами Figma

Після формування ідеї сайту можна переходити до розробки веб-додатку.

Почати слід з макету сайту. Найкращим додатком для створення макету є Figma і саме там я робив свій макет.Ось реалізація головної сторінки:

						ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			42

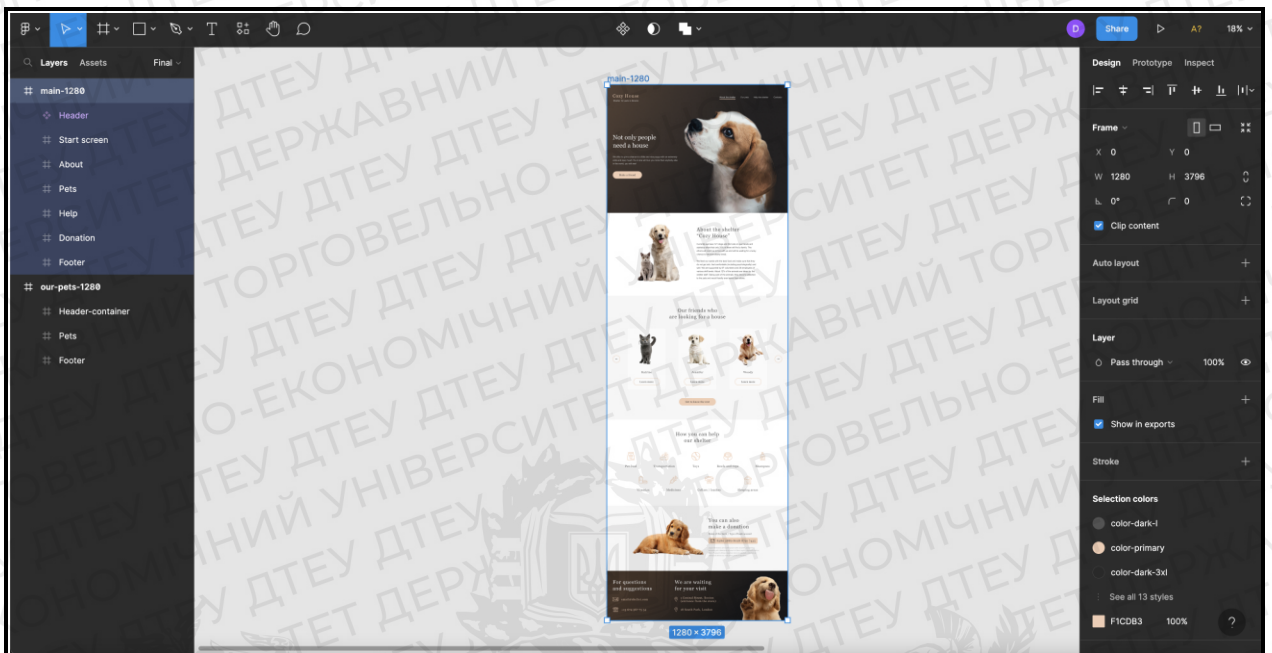


Рис.1 Фігма головна сторінка

А це реалізація сторінки Our pets:

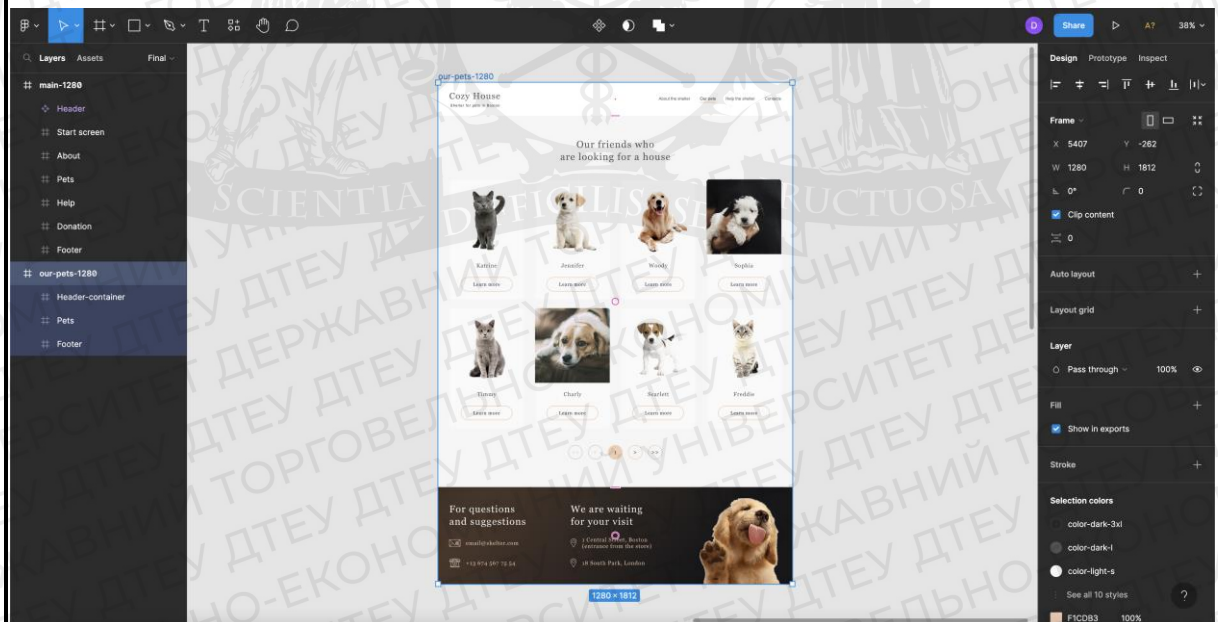


Рис.2 Фігма сторінка Our pets

						Аркуш
					ДТЕУ 121 06-15.БР	43
Зм.	Аркуш	№ докум	Підпис	Дата		

3.2.2 Використані технології програмування

Для розробки веб-додатку було використано наступні мови програмування – HTML/CSS, JavaScript та PHP. За допомогою HTML/CSS я реалізував верстку сайту, приклад коду реалізація header можна побачити в Додатку А, код форми зворотнього зв'язку в Додатку Б та код з інформацією про тварин в Додатку В.

3.3 Проектування бази даних

Першим і одним з найважливіших етапів в програмній реалізації є проектування бази даних.

База даних створювалась з допомогою phpMyAdmin це безкоштовний веб-орієнтований додаток з відкритим кодом, написаний на PHP. Він надає графічний інтерфейс користувача (GUI) для управління та адміністрування баз даних MySQL. За допомогою phpMyAdmin я зміг виконувати різноманітні завдання, пов'язані з управлінням баз даних, такі як створення і видалення баз даних, керування таблицями та виконання запитів.

В базі даних проекту визначені дві таблиці: animals_bd, в якій буде знаходитись інформація про тварин, а саме фото, ім'я, рік, порода, вага та характер; таблиця bd_form таблиця з формою зворотнього зв'язку в яку буде надходити інформація з сайту, а саме ім'я, пошта та телефон.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1	id	int(11)		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	2	photo	blob		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	3	name	varchar(50)	utf8_general_ci	Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	4	age	int(11)		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	5	breed	varchar(50)	utf8_general_ci	Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	6	weight	decimal(5,2)		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	7	character	varchar(255)	utf8_general_ci	Нет	Нет			Изменить Удалить Ещё

Рис.1 Таблица БД з інформацією про тварин

id	photo	name	age	breed	weight	character
1	[BLOB - 54.1 КиБ]	Ketrine	2	brazilian shorthair	4.50	Independent and playful
2	[BLOB - 50.9 КиБ]	Jennifer	0	labrador	3.50	very calm and obedient
3	[BLOB - 56.9 КиБ]	Woody	5	Golden retriever	27.00	intelligent reliable friendly kind
4	[BLOB - 59.4 КиБ]	Sophia	1	havanese bichon	5.00	playful sociable gentle intelligent

Рис.2 Таблица в якій видно інформацію

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(11)			Да	NULL			Изменить Удалить Ещё
2	name	text	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
3	email	text	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
4	phone	int(11)			Нет	Нет			Изменить Удалить Ещё

Рис.3 Таблица для форми зворотнього зв'язку

Додавання інформації з таблиці 1 реалізовано завдяки файлу submit.php. Саме цей файл відповідає за связування інформації з БД, який використовує серверну сторону програмування.

В Додатку Г наведено код цієї реалізації.

3.4 Висновок до Розділу 3

Було проведено детальне дослідження та описано процес розробки додатку, включаючи концепцію, розробку, макетування та прототипування веб-додатку, використані технології програмування та проектування бази даних.

Була розглянута загальна концепція та функціонал додатку. Розглянуті ключові функції та можливості додатку, які сприятимуть покращенню управління та комунікації притулку для тварин.

Були описані основні кроки та етапи розробки додатку, включаючи аналіз вимог, проектування архітектури, розробку функціоналу та інтерфейсу, тестування та впровадження. Розглянуті підходи та методики розробки, які сприяють успішному завершенню проекту в строк із високою якістю.

Було описано процес створення макетів та прототипів додатку за допомогою інструменту Figma. Використання макетування та прототипування дозволяє візуалізувати ідеї, перевірити їх функціональність та зручність використання перед переходом до фази роз



					ДТЕУ 121 06-15.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		46

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

З проведених досліджень можна зробити наступні висновки:

1. Доведено актуальність проблеми і вибору теми В якості операційних систем обрано, виконано детальний аналіз всіх її аспектів і можливих рішень.
2. Розглянуто інструменти для розробки веб-додатку, а саме: HTML/CSS, JavaScript, PHP, MySQL
3. Розроблено сучасний дизайн сайту
4. Відповідно до сформульованого технічного завдання програмно реалізовано і впроваджено проект для підтримки діяльності притулку для тварин



Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 06-15.БР</i>			
Зав. каф.		Криворучко О.В.		28.04.23				Веб-орієнтований додаток притуку для тварин
Керівник		Котенко Н. О.		28.04.23	ВП	47	50	
Гарант		Рзаєва С.Л.		28.04.23	Факультет інформаційних технологій			
Розробив		Компанець Д. О.		28.04.23	4 курс, 6 група			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. HTML [Електронний ресурс] – Режим доступу :
<https://developer.mozilla.org/ru/docs/Web/HTML>
2. Основи CSS [Електронний ресурс] – Режим доступу:
https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics
3. JavaScript [Електронний ресурс] – Режим доступу:
<https://developer.mozilla.org/ru/docs/Web/JavaScript>
4. Введення в JavaScript [Електронний ресурс] – Режим доступу:
<https://learn.javascript.ru/intro>
5. Що таке PHP [Електронний ресурс] – Режим доступу:
<https://www.php.net/manual/ru/intro-what-is.php>

<i>ДТЕУ 121 06-15.БР</i>								
					Веб-орієнтований додаток приюту для тварин <i>Список використаних джерел</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		23.12.22		<i>СВД</i>	48	50
Керівник		Котенко Н. О..		23.12.22		Факультет інформаційних технологій 4 курс, 6 група		
Гарант		Рзаєва С.Л.		23.12.22				
Розробив		Компанець Д. О.		23.12.22				

ДОДАТКИ

ДОДАТОК А

Код реалізації header

```
<header class="header container">
  <nav class="nav">
    <ul class="nav-list">
      <li class="logo">
        
      </li>
      <ul class="main-links">
        <li>
          <a class="nav_link" href="#">About the shelter</a>
        </li>
        <li>
          <a class="nav_link" href="shelter(our%20pets)/index.html">Our
pets</a>
        </li>
        <li>
          <a class="nav_link" href="#">Help the shelter</a>
        </li>
        <li>
          <a class="nav_link" href="#">Contacts</a>
        </li>
      </ul>
    </ul>
  </nav>
```


Форма зворотнього зв'язку

```
<div class="container">
```

```
<div class="frm">
```

```
<div class="sign">
```

```
<div class="sign_text">
```

```
<h1 class="title">ЗАПОВНІТЬ ФОРМУ</h1>
```

```
<p class="title_text">ЯКЩО ВИ ХОЧЕТЕ ДІЗНАТИСЯ БІЛЬШЕ  
ПРО НАШІ ПОСЛУГИ АБО ОБГОВОРИТИ,
```

```
ЯК МИ МОЖЕМО ДОПОМОГТИ ВАШОМУ БІЗНЕСУ,  
ЗВ'ЯЖІТЬСЯ З НАМИ! НАШІ ФАХІВЦІ ГОТОВІ ВІДПОВІСТИ
```

```
НА ВАШІ ЗАПИТАННЯ І ЗАПРОПОНУВАТИ ОПТИМАЛЬНЕ  
РІШЕННЯ, ЩО ВІДПОВІДАЄ ВАШИМ ПОТРЕБАМ.</p>
```

```
</div>
```

```
</div>
```

```
<form class="form">
```

```
<div class="form_main">
```

```
<div class="input-container1">
```

```
<input type="text" class="form-control" id="username"  
placeholder="Ваше ім'я">
```

```
</div>
```

```
<div class="input-container2">
```

```
<input type="email" class="form-control" id="email"  
placeholder="E-mail">
```

```
</div>
```

```
<div class="input-container3">
```

```
<input type="password" class="form-control" id="pwd"  
placeholder="Телефон">
```

```
</div>
```

```
<div class="form-group">
```

```
<button type="submit" class="btn-submit">Залишити  
заявку</button>
```

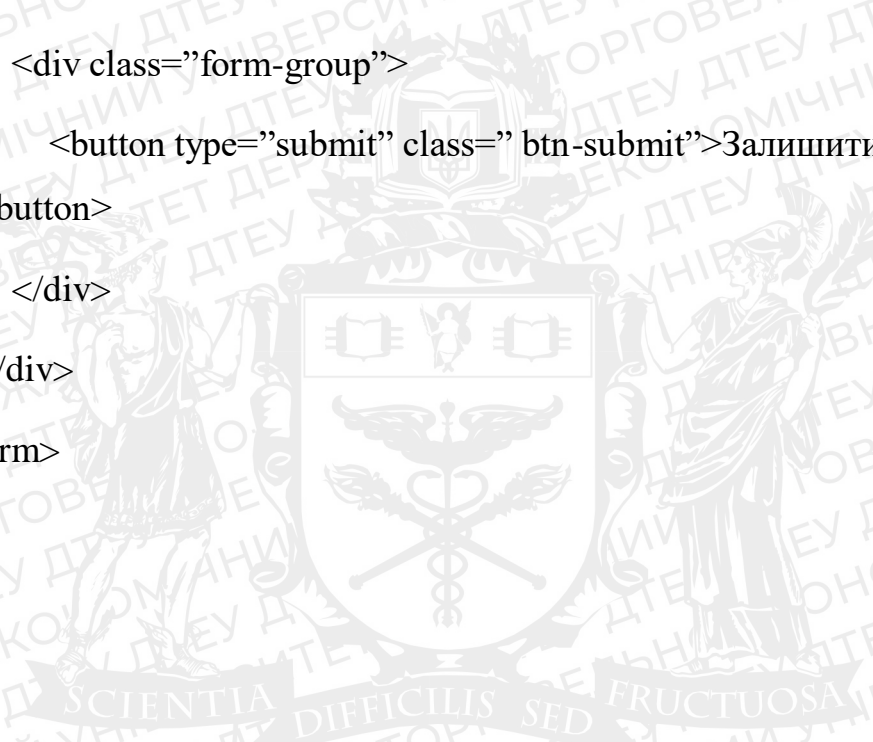
```
</div>
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</div>
```



Код з інформацією про тварин

```
<div class="container">  
  <h1 class="title">Тварина</h1>  
  <div class="animal">  
    <div class="animal-info">  
        
      <h2>Kit</h2>  
      <p>Вік: 2 роки</p>  
      <p>Порода: Британська короткошерста</p>  
      <p>Вага: 5 кг</p>  
      <p>Характер: Незалежний та грайливий</p>  
    </div>  
  </div>  
  <div class="popup-overlay" id="popupOverlay"></div>  
  <div class="popup-content" id="popupContent"></div>  
</div>
```

Код підключення сайту до бази даних

```
<?php
// Підключення до бази даних
$servername = «localhost»;
$username = «root»;
$password = «root»;
$dbname = «animals_bd»;

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die(«Помилка підключення до бази даних: « . $conn->connect_error);
}

// Отримання інформації про тварину з бази даних
$sql = «SELECT * FROM animals WHERE id = 1»;

$result = $conn->query($sql);
if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    $animalName = $row[«name»];
    $animalAge = $row[«age»];
    $animalBreed = $row[«breed»];
    $animalWeight = $row[«weight»];
    $animalCharacter = $row[«character»];
} else {
    echo «Тварина не знайдена»;
}
$conn->close();
?>
```


А за зворотній зв'язок відповідаю файл process_form.php з кодом:

```
<?php
$servername = «localhost»;
$username = «root»;
$password = «root»;
$dbname = «bd_form»;

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die(«Помилка з'єднання: « . $conn->connect_error);
}

$name = $_POST['username'];
$email = $_POST['email'];
$phone = $_POST['phone'];

$sql = «INSERT INTO your_table_name (name, email, phone) VALUES (?, ?,
?)»;

$stmt = $conn->prepare($sql);
$stmt->bind_param(«sss», $name, $email, $phone);

if ($stmt->execute()) {
    echo «Новий запис успішно створено»;
} else {
    echo «Помилка: « . $sql . «<br>» . $conn->error;
}
```



```
$stmt->close();
```

```
$conn->close();
```

```
?>
```

