

Державний торговельно-економічний університет
Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Web-орієнтована система ветеринарної клініки»

Студента 2 курсу, 4м групи
спеціальності
122 «Комп'ютерні науки»

Ластовка
Євген
Тарасович

*підпис
студента*

Науковий керівник
кандидат педагогічних наук, доцент

Дивак Володимир
Валерійович

*підпис
керівника*

Гарант освітньої програми
доктор фізико-математичних наук,
професор

Пурський Олег
Іванович

*підпис
керівника*

Київ 2023

Державний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 122 «Комп'ютерні науки»
Освітня програма «Комп'ютерні науки»

Зав. кафедри _____

Затверджую
Пурський О.І.
«9» грудня 2022р.

**Завдання
на випускню кваліфікаційну роботу студенту**

Ластовці Євгену Тарасовичу
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи
«Web-орієнтована система ветеринарної клініки»
Затверджена наказом ректора від «06» грудня 2022 р. № 3284
2. Строк здачі студентом закінченої роботи 24 листопада 2023 року
3. Цільова установка та вихідні дані до роботи
Мета роботи: теоретично обґрунтувати, розробити та перевірити web-орієнтовану систему ветеринарної клініки.
Об'єкт дослідження: розробка web-орієнтованої системи ветеринарної клініки.
Предмет дослідження: методи та технології розробки web-орієнтованої системи ветеринарної клініки.
4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Дивак В.В.	09.12.2023 р.	09.12.2023 р.
2	Дивак В.В.	09.12.2023 р.	09.12.2023 р.
3	Дивак В.В.	09.12.2023 р.	09.12.2023 р.

6. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ТЕОРЕТИЧНІ ПИТАННЯ РОЗРОБКИ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ВЕТЕРИНАРНОЇ КЛІНІКИ

1.1. Аналіз предметної області

1.2. Особливості розробки web-орієнтованої системи ветеринарної клініки

1.3. Аналіз методів та технологій розробки web-орієнтованої системи ветеринарної клініки

РОЗДІЛ 2. МОДЕЛЬ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ВЕТЕРИНАРНОЇ КЛІНІКИ

2.1. Модель web-орієнтованої системи ветеринарної клініки

2.2. Програмно технічні засоби розробки web-орієнтованої системи ветеринарної клініки

2.3. Опис програмного продукту

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ВЕТЕРИНАРНОЇ КЛІНІКИ

3.1. Тестування та аналіз результатів

3.2. Інструкція користувача

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ Пор	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	Вибір теми випускної кваліфікаційної роботи	01.11.2022	01.11.2022
2	Розробка та затвердження завдання на	09.12.2022	09.12.2022

	<i>випускнуну кваліфікаційну роботу</i>		
3	<i>Вступ</i>	<i>01.05.2023</i>	<i>01.05.2023</i>
4	<i>РОЗДІЛ 1. Теоретичні питання розробки web-орієнтованої системи ветеринарної клініки</i>	<i>14.06.2023</i>	<i>14.06.2023</i>
5	<i>Підготовка статті у збірник наукових статей магістрів</i>	<i>20.06.2023</i>	<i>20.06.2023</i>
6	<i>РОЗДІЛ 2. Модель web-орієнтованої системи ветеринарної клініки</i>	<i>08.09.2023</i>	<i>08.09.2023</i>
7	<i>РОЗДІЛ 3. Практична реалізація web-орієнтованої системи ветеринарної клініки</i>	<i>20.10.2023</i>	<i>20.10.2023</i>
8	<i>Висновки</i>	<i>02.11.2023</i>	<i>02.11.2023</i>
9	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	<i>22.11.2023</i>	<i>22.11.2023</i>
10	<i>Попередній захист випускної кваліфікаційної роботи</i>	<i>29.11.2023</i>	<i>29.11.2023</i>
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	<i>04.12.2023</i>	<i>04.12.2023</i>
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедру</i>	<i>06.12.2023</i>	<i>06.12.2023</i>
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	<i>За розкладом роботи ЕК</i>	

8. Дата видачі завдання «5» грудня 2021 р.

9. Керівник випускного кваліфікаційного проєкту Дивак В.В.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Пурський О.І

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент

Ластовка Є.Т.

(прізвище, ініціали, підпис)

Анотація

У випускній кваліфікаційній роботі здійснено розробку функціоналу та дизайну web-орієнтованої системи ветеринарної клініки. В даній роботі були проаналізовані та розглянуті аналогічні системи, інструментальні засоби розробки програмного продукту, матеріали для коректної розробки web-орієнтованої системи, різноманітні мови програмування, та проведено аналіз предметної області, логічної структури програми. Проведено практичну реалізацію web-орієнтованої системи ветеринарної клініки. Обрані інструментальні засоби розробки програмного продукту дадуть можливість розробити інтуїтивно зрозумілу, зручну web-орієнтовану систему для користувача.

Ключові слова: WEB, HTML, CSS, PHP, JavaScript, БД, СУБД, web-орієнтована система.

Anotation

In the final qualification work, the functionality and design of the web-oriented system of the veterinary clinic was developed. In this work, similar systems, software product development tools, materials for the correct development of a web-oriented system, various programming languages, and an analysis of the subject area and logical structure of the program were analyzed and considered. The practical implementation of the web-oriented system of the veterinary clinic was carried out. The selected software product development tools will provide an opportunity to develop an intuitive, convenient web-oriented system for the user.

Keywords: WEB, HTML, CSS, PHP, JavaScript, DB, DBMS, web-oriented system.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ПИТАННЯ РОЗРОБКИ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ВЕТЕРИНАРНОЇ КЛІНІКИ	10
1.1. Аналіз предметної області	10
1.2. Особливості розробки web-орієнтованої системи ветеринарної клініки ..	11
1.3 Аналіз методів та технологій розробки web-орієнтованої системи ветеринарної клініки.....	13
РОЗДІЛ 2. МОДЕЛЬ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ВЕТЕРИНАРНОЇ КЛІНІКИ	18
2.1. Модель web-орієнтованої системи ветеринарної клініки	18
2.2. Програмно технічні засоби розробки web-орієнтованої системи.....	22
2.3. Опис програмного продукту	27
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ВЕТЕРИНАРНОЇ КЛІНІКИ	29
3.1. Тестування та аналіз результатів.....	36
3.2. Інструкція користувача.....	36
ВИСНОВКИ	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	43
ДОДАТОК	44

ВСТУП

Актуальність дослідження зумовлена сучасними умовами отримання інформації. Web-системи або web-сайти надають доступ до великої кількості ресурсів та різного контенту. В даних умовах, що відбуваються в інформаційному просторі, все більшого значення набувають цифрові технології. Останнім часом з'явилося багато ветеринарних клінік з різними напрямками та послугами. Тому існує необхідність в автоматизації процесів на підприємствах, які надають різноманітні послуги.

Мета і завдання дослідження. Метою дослідження є теоретично обґрунтувати, розробити та перевірити web-орієнтовану систему ветеринарної клініки. Для досягнення даної мети було поставлено такі **завдання:**

1. Проаналізувати теоретичні та практичні підходи розробки web-орієнтованої системи ветеринарної клініки.
2. Побудувати модель web-орієнтованої системи ветеринарної клініки.
3. З'ясувати особливості використання програмно-апаратних засобів та розробити web-орієнтовану систему ветеринарної клініки.
4. Перевірити технологію та, у разі необхідності, внести зміни у web-орієнтовану систему ветеринарної клініки.

Об'єкт дослідження: процес розробки web-орієнтованої системи ветеринарної клініки.

Предмет дослідження: методи та технології розробки web-орієнтованої системи ветеринарної клініки.

Методи дослідження: теоретичний аналіз наукової літератури, порівняння, систематизація, класифікація дали змогу дослідити і узагальнити матеріали з питань розробки. Для вирішення поставленої задачі були використані такі методи:

- дослідження основних проблем предметної області (розділ 1, п.1.1, п.1.2);
- огляд існуючих рішень з розробки web-сайтів, web-додатків та web-орієнтованих систем (розділ 1, п.1.3);
- вибір мов програмування (розділ 2, п.2.2);
- вибір системи управління базою даних (СУБД) (розділ 2, п.2.1);
- детальний опис реалізації основних процедур та функцій web-орієнтованої системи з різними прикладами (розділ 3, п.3.1, п.3.2).

Наукова новизна результатів дослідження полягає в тому, що було теоретично обґрунтовано і розроблено технологію розробки та створення моделі web-орієнтованої системи. Сформовано метод розроблення дизайну web-орієнтованої системи, агрегація досліджених методів обробки даних та іншої інформації.

Практичне значення дослідження полягає в тому, що здійснення цих досліджень надає можливість: організувати ефективну роботу системи; автоматизувати процес зручного та швидкого керування даними; застосувати методи проектування інформаційних систем, більш детально ознайомитись, вивчити всі можливості вибраної СУБД та вдосконалити вміння написання SQL запитів; поглибити практичні та теоретичні знання.

Публікації. Результати дослідження опубліковано у збірнику наукових статей студентів, які здобувають освітній ступінь магістра за спеціальністю «Комп'ютерні науки» ДТЕУ. Web-система управління діяльністю он-лайн бібліотеки з рекомендаційним модулем // Прикладні комп'ютерні технології : зб. наук. ст. студ. / відп. ред.— Київ : Держ. торг.-екон. ун-т, 2023. – С. 73-77

Структура та обсяг випускної кваліфікаційної роботи. Випускна кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел із 11 найменувань, додатків і містить 44 сторінки основного тексту, 25 рисунків і 5 таблиць.

РОЗДІЛ 1.

ТЕОРЕТИЧНІ ПИТАННЯ РОЗРОБКИ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ВЕТЕРИНАРНОЇ КЛІНІКИ

1.1. Аналіз предметної області

На сьогоднішній день все більше з'являється організацій, які пропонують різноманітні медичні послуги для тварин. Ветеринарні клініки є перспективною і швидко розвиваючою галуззю, тому конкуренція в цій галузі тільки зростає. Тому, якщо організація має намір популяризувати свою клініку та, можливо, свою продукцію, вона має автоматизувати управлінські процеси, задля вивільнення часу на інші важливі роботи персоналу та організації в цілому.

Автоматизована система управління - інформаційна система, призначена для автоматизованого здійснення управлінських процесів.

Введення в дію АСУ повинен бути виправданий, тобто повинен призводити до корисним техніко-економічним, соціальним або інших результатів. Зокрема, використання автоматизованої системи управління дозволяє домогтися зниження чисельності управлінського персоналу, підвищення якості функціонування об'єкта управління і самого управління і т.д.

Будь-яка АСУ в процесі своєї роботи повинна виконувати наступні функції:

- збір, обробку та аналіз інформації (сигналів, повідомлень, документів тощо) про стан об'єкта управління;
- вироблення керуючих дій (програм, планів і т.д.);
- передачу керуючих впливів (сигналів, вказівок, документів) на виконання і контроль їх передачі;

- реалізацію і контроль виконання керівних впливів;
- обмін інформацією (документами, повідомленнями тощо) з іншими пов'язаними з нею автоматизованими системами. [1]

1.2. Особливості розробки web-орієнтованої системи ветеринарної клініки

Web-орієнтована система складається з деякого набору веб-сторінок і компонентів, які взаємодіють, утворюючи систему, яка виконується з використанням веб-серверів, мережі, протоколу прикладного рівня та браузера, і в якій дії користувача впливають на стан системи. Web-системи - це типові програми, які працюють в Інтернеті та взаємодіють з користувачем через Інтернет-браузер.

Обробка та зберігання інформації в інтернеті відбувається за допомогою web-орієнтованих інформаційних систем. Дані системи можуть використовуватися в локальній мережі. Web-орієнтовані системи побудовані з використанням веб-додатків (Web-Application) - допоміжних програмних засобах. Вони призначені для автоматизованого виконання будь-яких дій, як на веб-серверах так і на стороні користувача.

Особливості web-орієнтованих інформаційних систем:

1. надійність системи;
2. багатокористувацька робота;
3. кросплатформенність.

Особливості розробки web-орієнтованої системи ветеринарної клініки:

- Аналіз брифу і написання ТЗ. Брифом називається документ або анкета, яку заповнює замовник. У даному документі клієнт вказує свої побажання щодо дизайну, функціоналу сайту, системи чи програми та

інші деталі проекту. Завдяки збору цих даних менеджери ІТ-компанії складають технічне завдання для розробників.

- Прототипування. Є одним з важливіших етапів при розробці великих проектів. Прототип – це схема всіх або декількох сторінок сайту у формі ескізу або HTML-документа, в яких розглядаються взаємодії та структурні елементи майбутнього сайту.
- Розробка дизайну. Цей етап характеризується тим, що дизайнер промальовує кожну сторінку ресурсу. На сьогоднішній день сайт має мати як десктопну версію сайту (для комп'ютерів) так і мобільну версію сайту. Дизайн майбутнього сайту має бути інтуїтивно зрозумілим для користувача, який буде взаємодіяти з інтерфейсом.
- Програмування. Front-end та back-end розробка. Front-end - перетворення макетів з дизайном в інтерактивні веб-сторінки за допомогою мови. Back-end інтеграція зверстаних веб-сторінок з базою даних та системою керування сайтом.
- Тестування. Програма чи система проходить перевірку на відповідність макетів дизайну, швидкість роботи, оптимізацію під різноманітні мобільні пристрої і відображення в різних браузерах. Оцінкою готового продукту (програми) з точки зору програміста і користувача є фінальною перевіркою якості проекту. Проводиться детальне тестування всієї системи на працездатність.
- Наповнення сайту. Цей етап характеризується тим, що на сайт додається велика кількість інформації про товари, послуги або інших даних для чого цей сайт буде використовуватися.
- Деплой сайту. На цьому етапі компанія-розробник допомагає клієнту вибрати хостинг (місце в інтернеті) і отримати домен (адрес ресурсу). Після чого переносять сайт на цей хостинг. Ресурс відкривається для

пошукових систем. І проходить етап фінального тестування після переносу.

- Просування ресурсу. SEO-фахівці, SMM-менеджери, таргетологи займаються комплексним просуванням ресурсу: аналізують його роботу, складають стратегію внутрішньої і зовнішньої оптимізації та втілюють її у життя. [2]

Щоб web-орієнтована система відповідала вимогам, необхідно:

- створити інтуїтивно зрозумілий інтерфейс, щоб користувач не витрачав багато часу на банальні та рутинні дії;
- для персоналу – неможливість роботи з даними без реєстрації від свого облікового запису створеного адміністратором системи;
- можливість створення облікового запису (реєстрація та вхід в особистий кабінет);
- швидкодія програмного продукту;
- відправлення відгуків та коментарів адміністратору системи;
- можливість пошуку запису по запиту;
- зручна панель навігації web-орієнтованої система;

Під час розробки web-орієнтованої системи ветеринарної клініки представлений перелік можливостей системи може бути модернізований та доповнений деякими функціями.

1.3. Аналіз методів та технологій розробки web-орієнтованої системи ветеринарної клініки

Web-орієнтована система ветеринарної клініки повинна виконувати функції по редагуванню, перегляду даних, запис на процедуру, облік про видачу заробітної плати (або заборгованість перед працівником), рівні

доступу для працівників (в залежності від їх повноважень та посад), налаштування особистого кабінету, керування даними реклами, користувачами та іншої інформації.

Найбільш популярні види сайтів:

Сайт-візитка. Є найпростішим сайтом, який містить велику кількість сторінок, де є можливість швидкого переходу на будь-яку іншу сторінку сайту. Дані сайти призначені для короткого опису фірми, знайомства користувача з послугами, а також надання контактної інформації.

Лендінг (Landing page). Односторінковий сайт, що використовуються для популяризування деякого товару або групи товарів, які об'єднані однаковими функціями. Ленд – сторінка, має містити тільки основну інформацію в якій не повинно бути багато текстової інформації та різноманітних посилань та блоків, тощо.

Сайт компанії з каталогом продукції. Діже схожий на сайт-візитку, але в додаток до цієї інформації, ще містить й каталог всіх послуг, що надає організація чи підприємство. Цей список може мати великий об'єм інформації. У такому випадку на сайті роблять зручну систему навігації з категоріями, фільтрами тощо. Це – сайт із великою кількістю сторінок, розділів та підрозділів.

Інтернет-магазин. На таких сайтах надається список продукції, яку користувач може замовити онлайн, не виходячи з дому, не витрачаючи час на відвідування магазину та на вибір необхідного товару.

Промо-сайт. Такі сайти використовуються для якісної презентації товару чи послуги, масштабної піар-акції. Промо-сайти, як правило, створюються напередодні конкретної події чи виходу нового товару. Характерною ознакою також є велика кількість графіки й мультимедіа-презентацій товару.

Інформаційний портал. Це сайт, який містить велику кількість статей з різноманітних тем. Також він може об'єднувати інформацію, зібрану з інших ресурсів. Інформаційний портал – досить складний сайт з безліччю сторінок й розгалуженою структурою, тобто характеризується великою глибиною сайту і добре продуманою навігацією.

Корпоративний сайт – сайт, який є найбільш вигідним варіантом для середнього і великого бізнесу, для великих підприємств з кількома напрямками діяльності, філіями, підрозділами. Характеризується тим, що має велику функціональну можливість і розгалужену структуру з великою кількістю розділів і підрозділів, що містять детальну інформацію про компанію та її послуги [3].

JetVet – програма для ветеринарів та додаток для ветклініки. За допомогою даної програми є можливість керувати записами, нагадуваннями та іншими функціями необхідними у діяльності ветеринарної клініки.

Їх програмний продукт має такі позитивні характеристики: сайт є сучасним, зручним та має інтуїтивно зрозумілий інтерфейс; достатній функціонал, серед яких керування записами, нагадуваннями, картками пацієнтів, збереження даних про тварину, планування раціону, різноманітні нотатки та планування витрат. Недоліки: щоб отримати весь цей функціонал, необхідно заплатити за роботу в даному програмному забезпеченні. (рис.1.1).

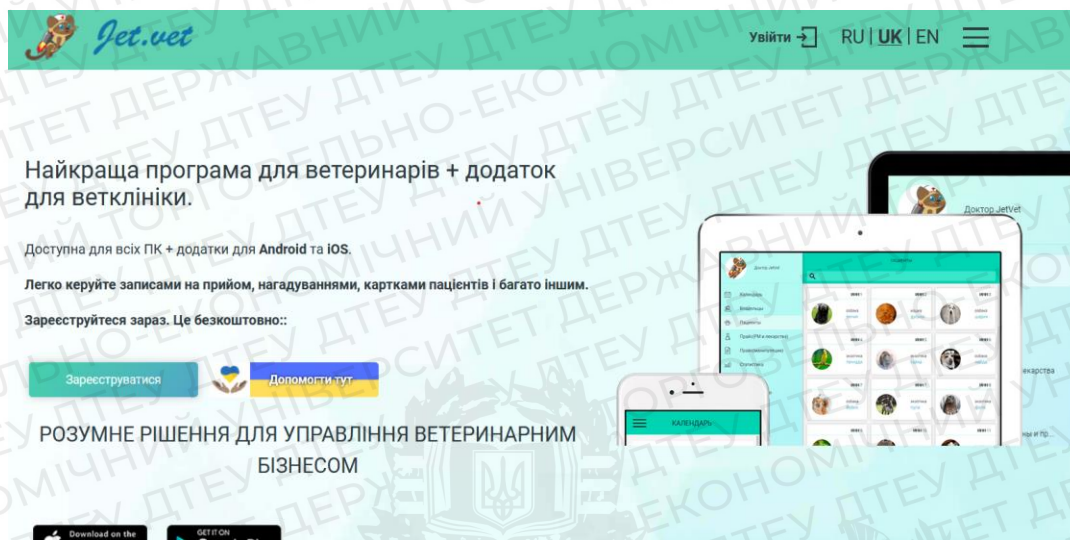


Рис.1.1. Програма для ветеринарів JetVet

Vetmanager – програма для керування ветеринарної клініки. Дана програма позиціонується як інструмент для керування базою даних клієнтів, фінансами та складом для ветеринарів. Даний сервіс постійно оновлюється та модернізується, є багато варіантів тарифів використання сервісу. Сервери сервісу знаходяться в Німеччині. Всі функції доступні одразу після купівлі тарифу. З недоліків – це те, що немає доступу ознайомитися з програмними продуктом до його купівлі; наявні тарифи платні, але є пробний період. (рис.1.2).

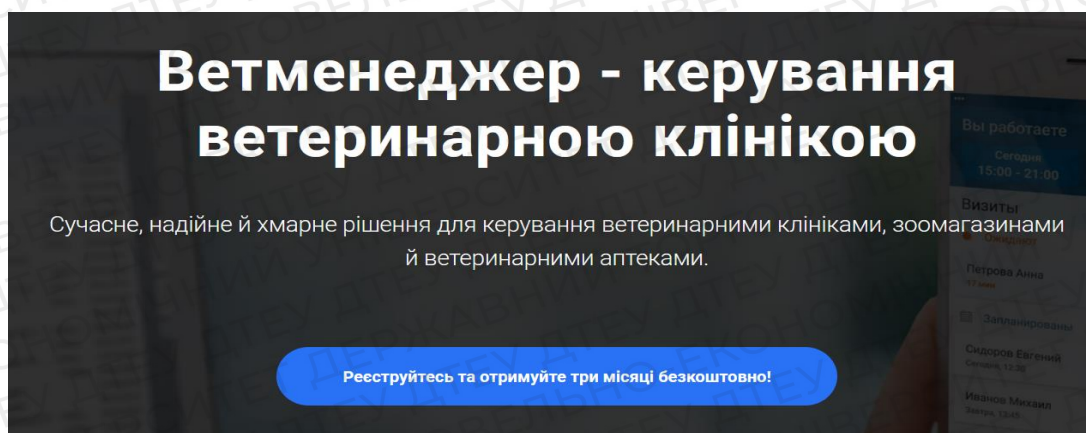


Рис.1.2. Ветменеджер - керування ветеринарною клінікою

В процесі дослідження, було зроблено аналіз предметної області, розроблені вимоги до майбутньої web-орієнтованої системи, проведено детальний аналіз та огляд подібних програмних рішень (сервісів, програм, сайтів) та виявлено їх переваги та недоліки, які в майбутньому будуть враховані під час розробки web-орієнтованої системи ветеринарної клініки.



РОЗДІЛ 2. МОДЕЛЬ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ВЕТЕРИНАРНОЇ КЛІНІКИ

2.1. Модель web-орієнтованої системи ветеринарної клініки

Web, або Всесвітня мережа (англ. World Wide Web) — глобальний інформаційний простір, заснований на фізичній інфраструктурі Інтернету і протоколі передачі даних HTTP.

Інтернет - всесвітня система добровільно об'єднаних комп'ютерних мереж, побудована на використанні протоколу IP і маршрутизації пакетів даних. Інтернет утворює глобальний інформаційний простір, слугує фізичною основою для Всесвітньої мережі і безлічі систем (протоколів) передачі даних.

Веб-сервіси - це технологія, яка дозволяє додаткам обмінюватися даними незалежно від платформи і мови програмування. Веб-сервіс обов'язково має програмний інтерфейс, який отримує через мережу команди і дані в задалегідь обумовленому форматі, виконує якісь операції і відправляє через мережу відповідь. Як протокол практично завжди використовується Тsp/іp, а якщо точніше, то HTTP або HTTPS.

Сторінки веб-сайтів — це файли в яких знаходиться деякий текст, розміченим на мові HTML. Ці файли завантажуються користувачем мережі на його власний комп'ютер, потім йде обробка програмою-браузером і, в результаті, виведення інформації на дисплей пристрою. [4]

Структура таблиць в СУБД MySQL та алгоритм web-орієнтованої системи (рис.2.1):

Таблиця 1 - Структура таблиці "Працівники".

Назва поля в	Тип поля	Опис поля	Обмеження
--------------	----------	-----------	-----------

СУБД			
id_emp	int	Код працівника	Primary key
name	varchar(255)	Ім'я	
lastname	varchar(255)	Прізвище	
email	varchar(255)	Пошта	
password	varchar(255)	Пароль	
id_position	int	Код посади	Foreign key
telephone	varchar(255)	Телефон	
date_birth	date	Дата народження	
img	mediumblob	Фото	
descrip	varchar(255)	Додаткова інформація	
acl	Int	Рівень доступу	
id_serv	Int	Код послуги	Foreign key
id_twork	Int	Код графіку роботи	Foreign key

Таблиця 2 - Структура таблиці "Тварини".

Назва поля в СУБД	Тип поля	Опис поля	Обмеження
id_pet	int	Код тварини	Primary key
name_p	varchar(100)	Ім'я	
years	int	Роки	
code_pet	int	ІДН тварини	
weight	float	Вага	
breed	varchar(100)	Порода	
type	varchar(100)	Тип тварини	

Таблиця 3 - Структура таблиці "Клієнт".

Назва поля в СУБД	Тип поля	Опис поля	Обмеження
id_client	int	Код клієнта	Primary key
name	varchar(255)	Ім'я	
lastname	varchar(255)	Прізвище	
email	varchar(255)	Пошта	
password	varchar(255)	Пароль	
date_birth	date	Дата народження	
telephone	varchar(255)	Телефон	
id_pet	int	Код тварини	Foreign key
acl	int	Рівень доступу	

Таблиця 4 - Структура таблиці "Заробітня плата".

Назва поля в СУБД	Тип поля	Опис поля	Обмеження
id_sal	int	Код зарплати	Primary key
id_emp	int	Код працівника	Foreign key
salary	int	Заробітня плата	
premium	int	Премія	
calculations	int	Вирахування	
to_pay	int	До виплати	
paid	int	Виплачено	
yes_not	tinyint(1)	Виплачено чи ні	
date	date	Дата видачі	

Таблиця 5 - Структура таблиці "Матеріали".

Назва поля в СУБД	Тип поля	Опис поля	Обмеження
id_material	int	Код запису користувача	Primary key
name	varchar(255)	Імя	
img	varchar(255)	Фото	
descrip_m	varchar(255)	Опис матеріалу	
id_cat	int	Код категорії	Foreign key

count	int	Кількість	
price	int	Ціна	
id_avail	int	Код наявності	Foreign key
id_provider	int	Код постачальника	Foreign key

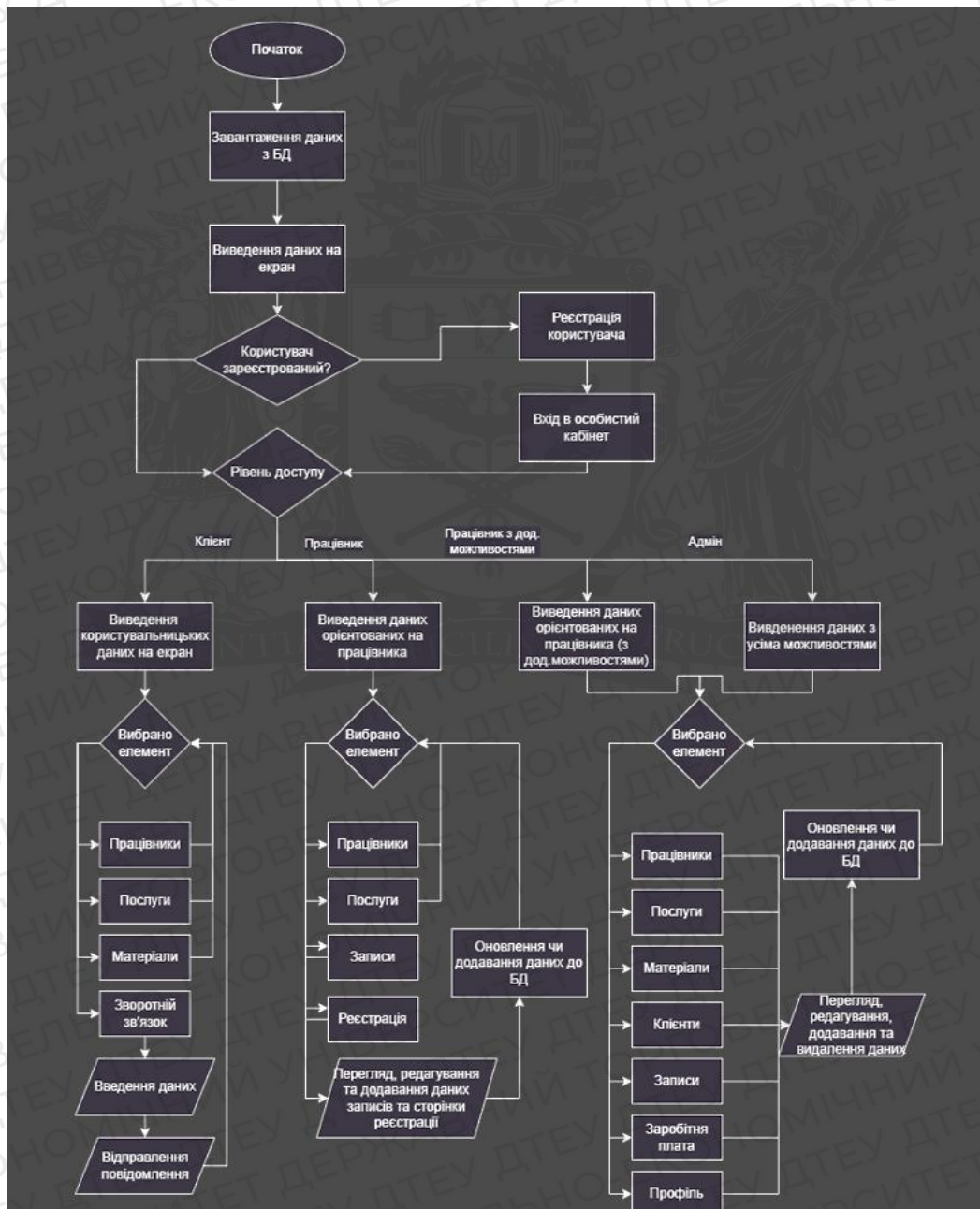


Рис.2.1. Алгоритм web-орієнтованої системи

2.2 Програмно технічні засоби розробки web-орієнтованої системи ветеринарної клініки

Архітектура клієнт – сервер (client-server architecture) – це концепція інформаційної мережі, яка визначає два типи компонентів: сервери і клієнти. Основна частина ресурсів даної архітектури, знаходиться в серверах.

Клієнт – комп'ютер на стороні користувача, який відправляє запит до сервера для надання інформації або виконання певних дій.

Сервер – це більш потужний комп'ютер або обладнання, призначене для вирішення конкретних завдань з виконання програмних кодів, виконання сервісних функцій за запитом клієнтів, надання користувачам доступу до певних ресурсів, зберігання інформації і баз даних.

Сервер працює за завданнями клієнтів і керує виконанням їх завдань. Після виконання кожного завдання сервер, в свою чергу, посилає отримані результати клієнту, який послав це завдання.

Сервісна функція в архітектурі клієнт – сервер описується комплексом прикладних програм, відповідно до якого виконуються різноманітні прикладні процеси.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення .

Архітектура клієнт-сервер передбачає взаємодію та обмін даними між ними та є концепцією у створенні розподілених мережних застосувань. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них (в найпростішому випадку — сервер один);
- набір клієнтів, які використовують сервіси, що надаються серверами;
- забезпечення взаємодії між клієнтами та серверами за допомогою мережі.

Функції, які реалізуються на сервері:

- зберігання, доступ, захист і резервне копіювання даних;
- обробка запиту клієнта;
- відправлення результату (відповіді) клієнту.

Функції, які реалізуються на стороні клієнта:

- надання інтерфейсу для користувача;
- формулювання і відправка запиту до сервера;
- отримання результатів запиту і відправка додаткових команд (запитів на додавання, оновлення або видалення даних).

За допомогою клієнт-серверної архітектури визначаються принципи спілкування між комп'ютерами, правила і взаємодії визначені в протоколі.

Існують концепції побудови системи клієнт-сервер:

1. Слабкий клієнт – це потужний сервер, де вся обробка інформації перенесена на сервер, а права доступу клієнта дуже обмежені. Сервер, в свою чергу, відправляє відповідь, яка не вимагає додаткової обробки. Клієнт взаємодіє з користувачем: складає та відправляє запит, приймає результат і виводить інформацію на екран.
2. Сильний клієнт – концепція, в якій частина обробки інформації надається клієнтові, а сам сервер виступає сховищем даних. Комп'ютер клієнта виконує роботу по обробці та поданню інформації. [5, 6, 7]

Після досліджень та аналізу літератури, для розробки web-орієнтованої системи ветеринарної клініки було прийнято рішення використовувати такі інструментальні засоби як:

- Open Server - це локальний сервер, завдяки якому можна керувати базою даних, конфігурацією підключень та створювати проекти. Сервер має багато керуючих програм та компонентів, які

полегшують взаємодію з програмою.

- HTML (HyperText Markup Language) - мова гіпертекстової розмітки.
- CSS – це мова таблиці стилів.
- PHP (рекурсивний акронім словосполучення PHP: Hypertext Preprocessor). Є однією з поширеніших мов програмування з відкритим вихідним кодом. PHP часто використовується для веб-розробки і його код може впроваджуватися безпосередньо в HTML.
- JavaScript - об'єктно-орієнтована, скриптова мова програмування, використовується для програмного доступу до об'єктів додатку.
- SQL - декларативна мова програмування. Вона використовується для створення, модифікації та управління даними в реляційній базі даних.
- MariaBD – відгалуження реляційної СУБД MySQL.

MariaDB – відгалуження реляційної СУБД MySQL, що розроблялася та підтримується спільнотою під ліцензією GPL. MariaDB є повністю сумісною з програмами, що використовують MySQL. Так як MySQL вже не дуже активно розвивається, то перехід на MariaDB є гарним вибором.

Крім зрозумілої політики розповсюдження продукту, дана СУБД відрізняється від MySQL вищою продуктивністю, новими можливостями та меншою кількістю помилок у коді (зокрема, за рахунок кращого покриття тестами та активнішого розвитку).

У MariaDB вбудований покращений оптимізатор запитів, безпечна та швидка реплікація, швидші індекси для механізму зберігання даних MEMORY(HEAP), більш висока продуктивність перекодування символів, використання пулу потоків, а також багато інших покращень, що позитивно впливають на продуктивність. У MariaDB відмова від підсистеми зберігання даних InnoDB та його заміна на XtraDB. Також крім MyISAM, Blackhole,

CSV, Memory та Archive включені підсистеми зберігання Aria, PBXT (до версії 5.5), FederatedX, OQGRAPH, SphinxSE, IBMDB2I (до версії 5.5), TokuDB, а також с версії 10.0 — Cassan SEQUENCE, Spider. Провідний розробник - Майкл Віденіус, автор оригінальної версії MySQL. Поштовхом до створення стала необхідність забезпечення вільного статусу СУБД (під ліцензією GPL) на противагу невизначеній політиці ліцензування MySQL компанією Oracle. MariaDB розробляється open source спільнотою.

PHP — це скриптова мова програмування, створена для генерації HTML-сторінок на стороні вебсервера. Офіційна назва PHP: Hypertext Preprocessor, але спочатку PHP розшифровувалася як Personal Home Page.

PHP є однією з найпоширеніших мов, яку використовують у сфері веброзробки, її підтримує більшість хостинг-провайдерів. PHP інтерпретує вебсервер у HTML-код, який передається на сторону клієнта. Користувач при перегляді веб-сторінки не бачить PHP-коду, тому що браузер отримує готовий HTML-код. Це є перевагою з погляду безпеки, але є і негативні наслідки - це погіршує інтерактивність сторінок.

PHP сьогодні використовують у багатьох сферах розробки:

- Веброзробка.
- Електронна комерція.
- Загальноорганізаційне програмне забезпечення.
- Створення баз даних.
- Мобільні застосунки.[8]

Популярність PHP мови обумовлена його перевагами:

- Гнучкість налаштування та сумісність з усіма відомими ОС та веб-серверами, легка інтеграція з MySQL.

- Детальна документація, що значно спрощує розробку та пошук спеціаліста.
- Зрозумілий синтаксис на базі мов C, Java та Perl.
- Масштабованість, яка дозволяє програмам на PHP витримувати великий потік трафіку.
- Гарна перспектива у майбутньому. На PHP пишуть CMS. Є багато популярних фреймворків, які створені на базі цієї мови. Тому фахівці PHP ще довгий час будуть затребувані, тому і популярність мови PHP буде, може не стрімко, але збільшуватися. [9]

SQL (Structured Query Language) застосовується практично в кожному додатку або на будь-якому сайті, які взаємодіють з базами даних. Це мова програмування, єдина функція якої – відправлення запитів до реляційних БД.

Під запитом мається на увазі звернення до бази даних, яке необхідне для управління інформацією в ній (видалення, додавання або зміна), здійснюване за допомогою системи управління базами даних (СКБД).

Мова виступає в ролі такого собі посередника між БД і проектом. Він дозволяє обробляти великі масиви даних, отримувати структуровану інформацію, а також адмініструвати сервер СУБД і редагувати таблиці.

SQL складається з набору операторів, які діляться на групи, кожна з яких має своє призначення.

Вони бувають наступними:

- DML – оператори маніпулювання даними, для управління вмістом БД та вилучення інформації з нього;
- DDL – оператори опису даних, для управління об'єктами БД;
- TCL – мова управління транзакціями;
- DCL – група операторів визначення доступу до даних.

Одні з найбільш популярних реляційних СУБД – MySQL, Oracle Database, PostgreSQL і MS SQL Server. SQL стандартизований, але в кожній з них є свої різновиди мови, які називаються діалектами.[10]

2.3. Опис програмного продукту

Дана web-орієнтована система ветеринарної клініки буде орієнтована в першу чергу на працівників самої клініки, щоб контролювати усі процеси, що відбуваються на підприємстві, маніпулювання великою кількістю інформації (плинність кадрів, облік матеріалів, замовлень, відвідувань та багато іншої інформації, яка важлива для функціонування клініки). Кожен працівник має свій рівень доступу, завдяки якому буде розподілено до яких даних працівник має доступ, а до яких не має.

Web-орієнтована система також буде орієнтована на користувача. Завдяки цьому сам же користувач може записатися на прийом до лікаря, подивитися інформацію про послуги, які надає клініка, ціни на ці послуги, спілкуватися з лікарем, написати відгуки, тощо.

Призначення даної web-орієнтованої системи полягає в автоматизації роботи працівників, для отримання достовірної та своєчасної інформації. Програма повинна бути зрозуміла та проста у використанні. Так як програма буде виконувати операції з базою даних, вона повинна бути оптимізована та працювати без помилок.

Інтерфейс програми повинен бути зручний у використанні, щоб користувач зміг за лічені хвилини знайти потрібну сторінку чи необхідну кнопку.

При роботі з програмою користувачу, у деяких випадках, не потрібно буде витратити час на заповнення усіх полів введення, бо ці поля вже з

заповненою інформацію, де користувачеві залишиться тільки обрати необхідні дані з випадуючого списку.

Вхідні та вихідні дані web-орієнтованої системи ветеринарної клініки.

Перелік вхідних даних:

- дані про працівника;
- дані про клієнта;
- дані про тварину;
- дані про посади;
- дані про послуги;
- дані про матеріали;
- дані запису домашніх улюбленців клієнтів на прийом до лікарів;
- заробітна плата кожного працівника.

В якості вихідних даних виступає наступна інформація:

- особова картка працівника;
- особова картка клієнта;
- особова картка домашнього улюбленця клієнта;
- заробітна плата працівників;
- статистична інформація;
- особиста електронна адреса для користувача;
- лист відправлений на локальний сервер.

РОЗДІЛ 3.

ПЕРЕВІРКА WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ВЕТЕРИНАРНОЇ КЛІНІКИ

3.1 Тестування та аналіз результатів

Тестування - оцінка якості ПЗ методом експериментальної перевірки - шляхом виконання тестів. Мета тестування - виявити наявність помилок/неузгодженостей. Тобто завданням тестування є знаходження помилок та досягнення відсутності критичних та інших помилок.

Кінцевою метою тестування програм чи ІТ-проектів є отримання сертифіката на розроблений програмний продукт.

Методи тестування й верифікації залежать від методів проектування та стадій, з яких починається перевірка правильності функціонування результатів проектування.

Основною метою програмного тестування є забезпечити повноту й узгодженість функцій та інтерфейсів, які були реалізовані програми. [11]

Тестування web-орієнтованої системи ветеринарної клініки:

Перед тим, як отримати доступ до потрібних функцій системи необхідно авторизуватися у web-системі, далі буде надано доступ до функцій, які відповідають рівню доступу облікового запису користувача.

Вхід в обліковий кабінет користувача з подальшою перевіркою каптчею (рис. 3.1.).



Рис. 3.1. Процес входу в особистий кабінет користувача

В процесі входу, проходить валідація введених даних (рис.3.2).

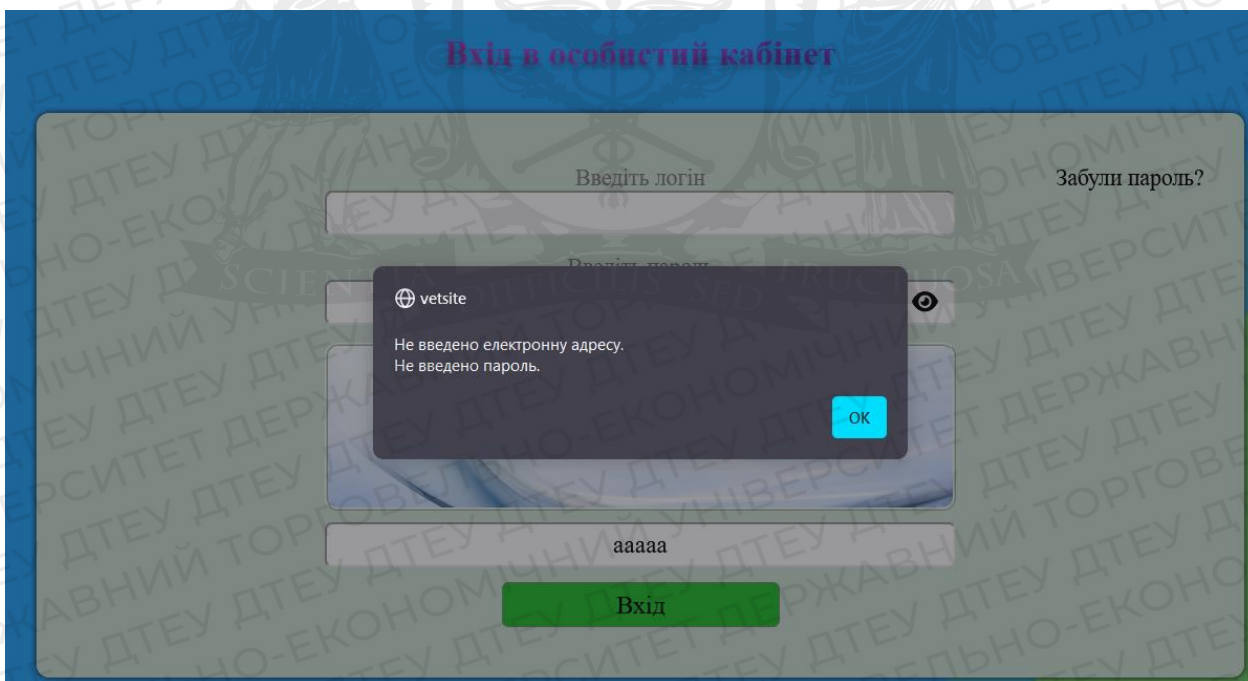


Рис. 3.2. Виведення повідомлення про помилки при введенні даних

Якщо користувач забув пароль від облікового запису, то є можливість натиснути кнопку «Забули пароль?» (ввівши електронну пошту) та

отримати тимчасовий пароль (який прийде на пошту), який в подальшому необхідно замінити на більш захищений. (рис.3.3, рис.3.4)

Відновлення паролю

Введіть адресу електронної пошти

roll@roll.com

Надіслати

Рис.3.3. Відновлення паролю

Ваш тимчасовий пароль: 5с3сс6
За можливості змініть пароль для забезпечення безпеки вашого особистого кабінету.

Рис.3.4. Тимчасовий пароль

Результат пошуку матеріалу у web-орієнтованій системі та виведення результату на екран (рис.3.5).

Ваше Працівники Послуги Клієнти Матеріали Записи Заробітня плата roll@roll.com

Вітамін Пошук

Результати пошуку

Вітамін
Код лікарського засобу: 11
Категорія товару
Вітамінні та імуностимулюючі препарати
Кількість
5
Ціна за одиницю
250

Рис.3.4. Результати пошуку

При додаванні даних про клієнта та домашнього улюбленця, працівник має заповнити всі необхідні поля, щоб введені дані збереглись в БД. (рис.3.6.)

• Ім'я: Рубін

• Роки: 5

• ІДН тварини: 1334522

• Вага: 12

• Порода: Терьер

• Вид: Собака

Додати

Рис.3.6. Створення запису в БД про домашнього улюбленця

Додавання даних про видачу заробітної плати працівникам (рис.3.7.). Розрахунок премії та відрахувань з заробітної плати працівника здійснюється автоматично в залежності від суми заробітної плати самого працівника.

○ Заробітна плата для: Максим Ролінов

○ Видано: 5000

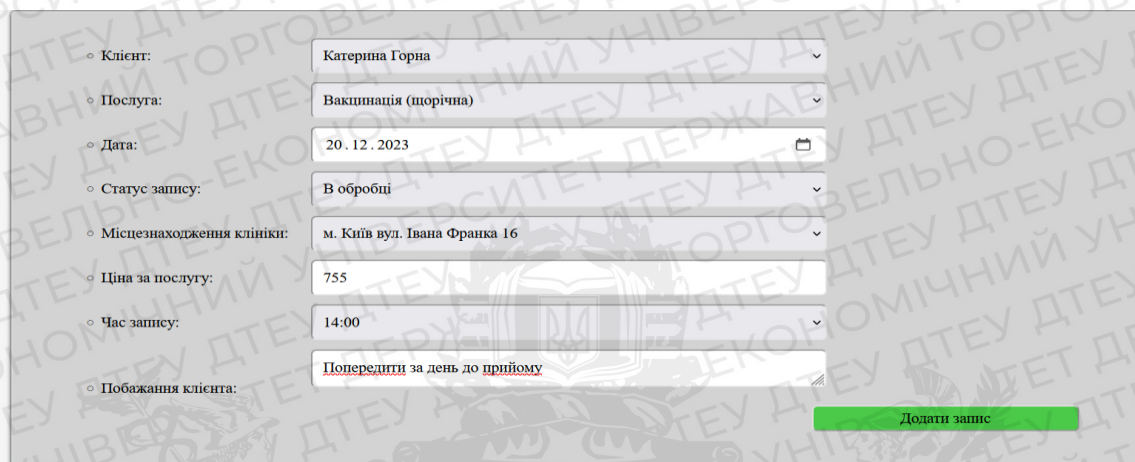
Зарплата без премії та відрахувань: 23000

○ Дата видачі: 19.10.2023

Додати запис

Рис.3.7. Додавання даних про видачу заробітної плати працівникам

При створенні запису прийому до лікаря, більшість полів вже мають дані, які необхідно обрати, та додати запис до БД. (рис.3.8)



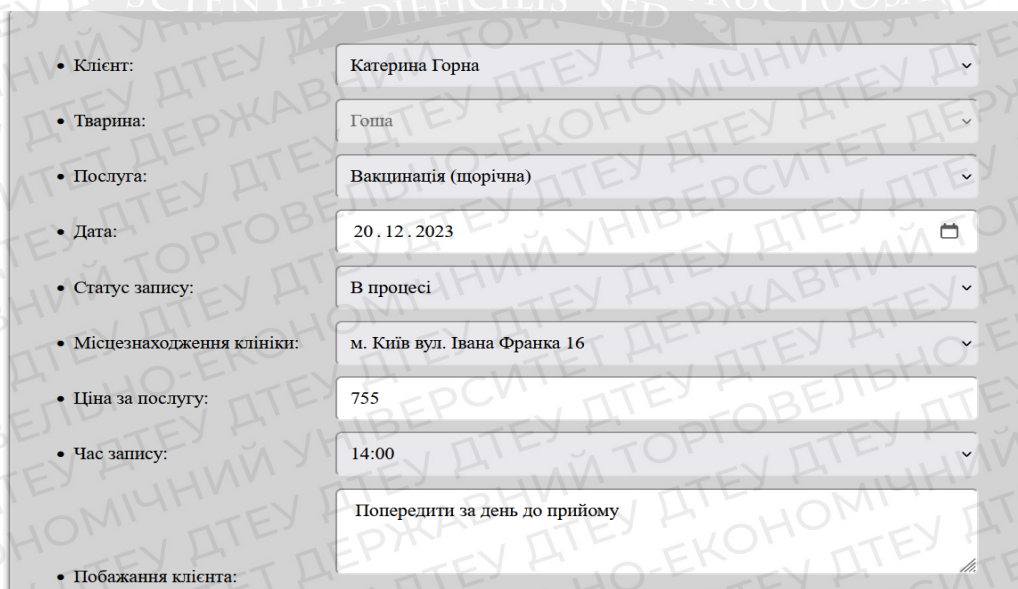
The screenshot shows a form for creating a medical appointment. The fields are as follows:

Клієнт:	Катерина Горна
Послуга:	Вакцинація (щорічна)
Дата:	20. 12. 2023
Статус запису:	В обробці
Місцезнаходження клініки:	м. Київ вул. Івана Франка 16
Ціна за послугу:	755
Час запису:	14:00
Побажання клієнта:	Попередити за день до прийому

A green button labeled "Додати запис" is located at the bottom right of the form.

Рис.3.8. Створення запису на прийом до лікаря

Зміна даних запису на прийом до лікаря, відбувається таким же чином, що і створення запису. При зміні або перегляді даних, з'являється поле, де вказується домашній улюбленець клієнта, який прив'язаний до запису клієнта.(рис.3.9)



The screenshot shows the same form as in Figure 3.8, but with an additional field for the pet:

Клієнт:	Катерина Горна
Тварина:	Гоша
Послуга:	Вакцинація (щорічна)
Дата:	20. 12. 2023
Статус запису:	В процесі
Місцезнаходження клініки:	м. Київ вул. Івана Франка 16
Ціна за послугу:	755
Час запису:	14:00
Побажання клієнта:	Попередити за день до прийому

Рис.3.9. Зміна даних запису на прийом до лікаря

Одні й ті ж самі дії необхідно виконати при створенні нового облікового запису про клієнта. Після заповнення полів введення дані будуть перевірені та виведено відповідне повідомлення (рис.3.10).

The image shows a registration form for a client. It consists of several input fields and a submit button. The fields are labeled as follows:

- Ім'я: Микита
- Прізвище: Прикін
- Логін: login@gmail.com
- Пароль: Password (with an eye icon for visibility)
- Телефон: +38 (848) 888-88-88
- Дата народження: 28.10.1979 (with a calendar icon)
- Домашній улюбленець: Рей (with a dropdown arrow)

At the bottom right of the form is a green button labeled "Додати клієнта". To the right of the form is a cartoon illustration of a man with a blue banner that says "CLIENT".

Рис.3.10. Реєстрація клієнта

Якщо користувач має рівень доступу, який не відповідає налаштуванням системи, то при деякій дії користувачу буде виведено повідомлення, де буде відображено інформацію, що в нього не має необхідних прав для редагування цієї інформації. (рис.3.11)

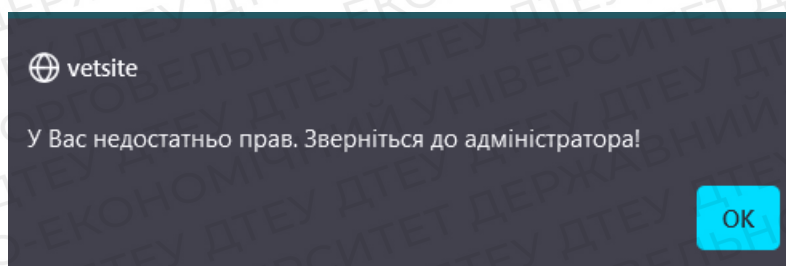
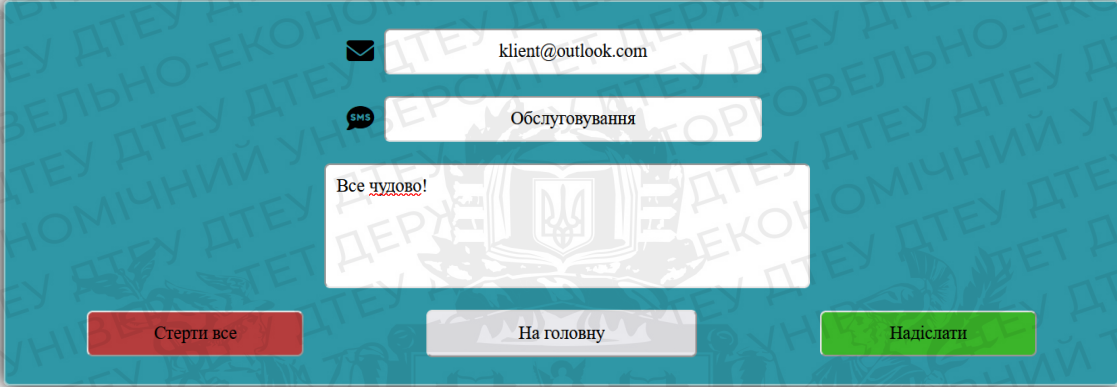


Рис.3.11. Повідомлення про недоступність деякої дії

Кожен відвідувач сайту має можливість залишити повідомлення чи коментар для організації, яке прийде на пошту компанії. Це зроблено з

метою подальшої модернізації web-орієнтованої системи, задля зручного користування системою, як працівникам так і клієнтам.(рис.3.12)

Залишити коментар чи побажання



✉ klient@outlook.com

💬 Обслуговування

Все чудово!

Стерти все На головну Надіслати

Рис.3.12. Відправлення коментаря чи повідомлення

Якщо працівник має найнижчий рівень доступу, то функціонал системи буде обмежений, та деякі дії чи функції будуть відрізнятись від дій працівника з вищим рівнем доступу. Додавання даних до БД про клієнта та домашнього улюбленця здійснюється доволі просто, а якщо при введенні даних були допущені помилки буде виведено повідомлення (рис.3.13).

Якщо на комп'ютер не підтримує JavaScript, то виведення повідомлення про помилки буде інакшим (рис.3.14).

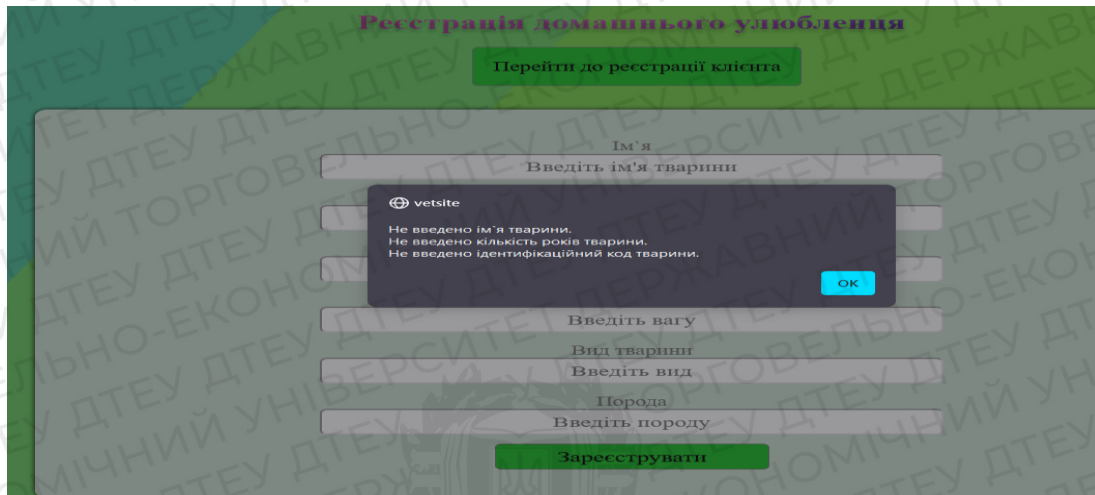


Рис.3.13. JavaScript повідомлення про некоректно заповнені поля



Рис.3.14. Повідомлення про некоректно заповнені поля (без JavaScript)

3.2 Інструкція користувача

Після створення веб-сайту, додатку чи web-орієнтованої системи необхідно описати функціонал багатьох кнопок, як користуватися функціями, які присутні у системі, для того щоб зрозуміти що і як працює та зробити знайомство користувача з системою простішим.

1. Головна сторінка web-орієнтованої системи ветеринарної клініки (рис. 3.15.)

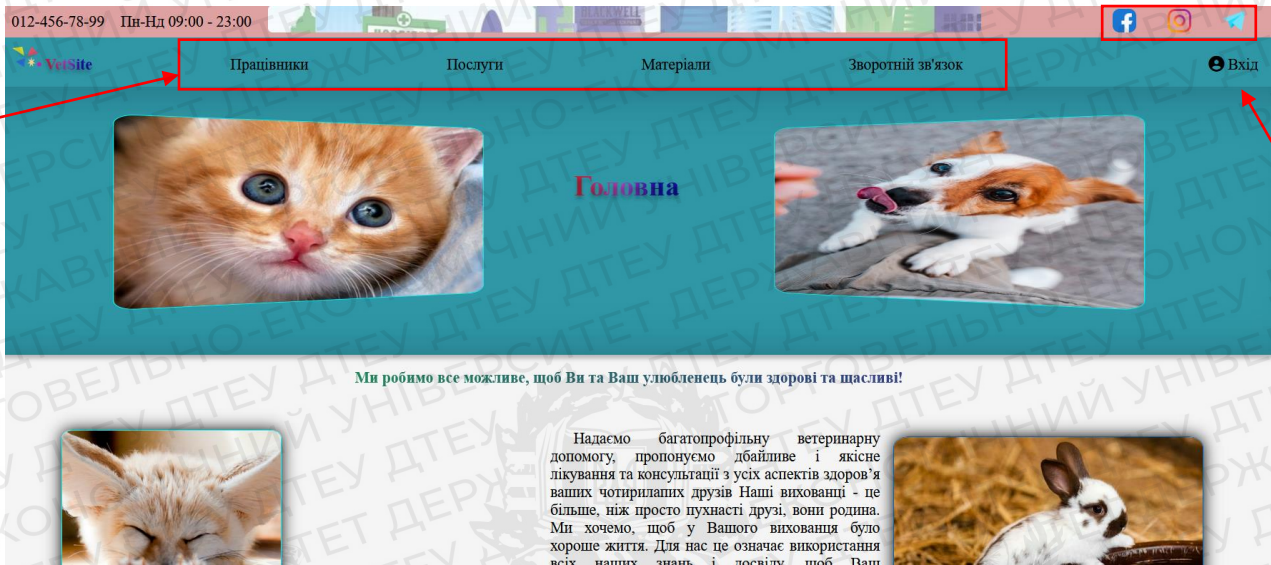


Рис.3.15. Головна сторінка

1. Посилання на відповідні сторінки, де буде відображено деяку інформацію.
 2. Посилання на соціальні мережі web-орієнтованої системи ветеринарної клініки.
 3. Вхід в особистий кабінет користувача.
2. Сторінка входу в особистий кабінет працівника ().

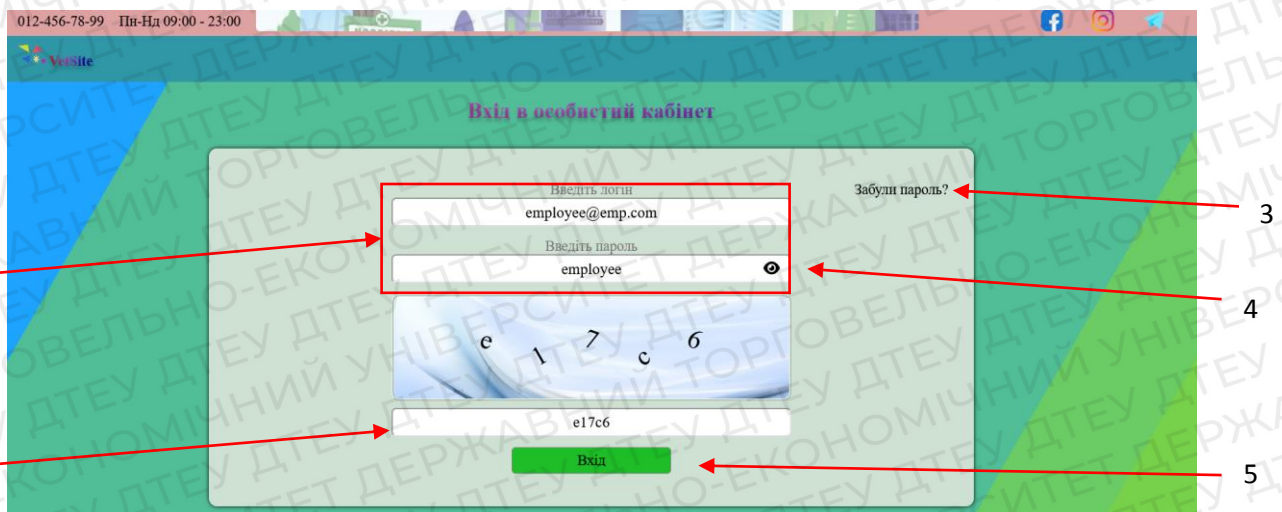


Рис.3.16.

1. Поля введення даних (логін та пароль).

2. Перевірка каптчею.
 3. Посилання на сторінку відновлення паролю.
 4. Перегляд паролю (за замовчуванням позначається точками).
 5. Кнопка входу в обліковий запис.
3. Головна сторінка web-орієнтованої системи ветеринарної клініки (рис. 3.17.)

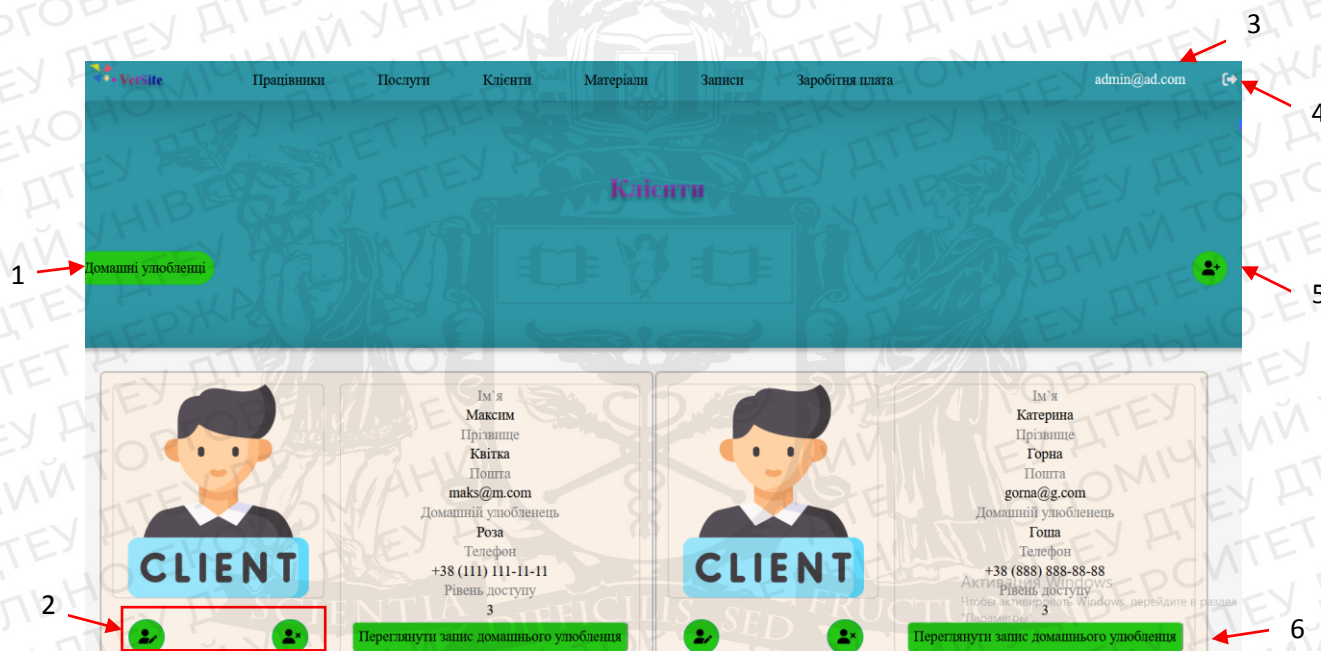


Рис.3.17.

1. Перехід на сторінку, де знаходяться дані про домашніх улюбленців клієнтів клініки.
2. Редагування та видалення запису.
3. Логін користувача, є також можливість переходу у власний профіль, де можна ознайомитись з даними.
4. Вихід з облікового запису.
5. Додавання запису про клієнта.
6. Перехід на сторінку, де знаходяться дані про домашнього улюбленця відповідного клієнта.

4. Додавання запису про клієнта в БД (рис. 3.18.)

The screenshot shows a registration form for a client. On the left, a list of labels is followed by input fields. A red box highlights the first seven fields: Name (Петро), Surname (Введіть прізвище), Email (petro@gmail.com), Password (masked with dots), Phone (+38 (578) 921-46-31), Birth Date (08.11.1999), and Pet (Кеша). A dropdown menu is visible for the Pet field. To the right is a cartoon character with a blue 'CLIENT' badge. At the bottom right is a green 'Додати клієнта' button. Red arrows and numbers 1, 2, and 3 point to the first field, the dropdown menu, and the button respectively.

• Ім'я:	<input type="text" value="Петро"/>
• Прізвище:	<input type="text" value="Введіть прізвище"/>
• Логін:	<input type="text" value="petro@gmail.com"/>
• Пароль:	<input type="password" value="•••••"/>
• Телефон:	<input type="text" value="+38 (578) 921-46-31"/>
• Дата народження:	<input type="text" value="08.11.1999"/>
• Домашній улюбленець:	<input type="text" value="Кеша"/>

Додати клієнта

Рис.3.18. Додавання запису про клієнта

1. Обов'язкові поля для введення даних.
2. Випадаючий список.
3. Кнопка додавання даних в БД.

5. Зміна даних про домашнього улюбленця (рис. 3.19.)

The screenshot shows a form for pet information. On the left, a list of labels is followed by input fields. A red arrow and number 1 point to the 'Ім'я:' label and its corresponding field (Роза). The other fields are: Owner (Максим), Age (2), ID number (14422312), Weight (2), Breed (Тер'єр), and Type (Собака). On the right is a photo of a dog. At the bottom right is a green 'Зберегти зміни' button.

• Ім'я:	<input type="text" value="Роза"/>
• Власник:	<input type="text" value="Максим"/>
• Роки:	<input type="text" value="2"/>
• ІДН тварини:	<input type="text" value="14422312"/>
• Вага:	<input type="text" value="2"/>
• Порода:	<input type="text" value="Тер'єр"/>
• Вид:	<input type="text" value="Собака"/>

Зберегти зміни

Рис.3.19. Зміна даних про домашнього улюбленця

1. Поле, яке неможливо змінити, бо дані прив'язані до

відповідного клієнта.

6. Сторінка записів прийомів до лікарів (рис. 3.20.)

Записи на прийом до лікарів

№ / Клієнт	Послуга	Час запису	Додаткова інформація		Ціна	Тварина	Дата	Дія
			Місцезаходження	Статус				
15 / Петро Лоновий	Обробка ротової порожнини	15:00	м. Київ вул. Почайна 2а	В обробці	450	Гарфілд	2023-11-24	
14 / Олександра Сова	Консультація	11:00	м. Київ вул. Івана Франка 16	Готово	300	Аїя	2023-11-07	
13 / Анастасія Квітчана	Лікар вузької спеціальності	17:00	м. Київ вул. Почайна 2а	В процесі	1000	Мальфа	2023-11-07	
12 / Віталій Озорний	Глибокий зіскріб зі шкіри	19:00	м. Київ вул. Івана Франка 16	В обробці	550	Рей	2023-11-10	
11 / Віталій Озорний	Глибокий зіскріб зі шкіри	09:00	м. Київ вул. Почайна 2а	Відмінено	550	Рей	2023-11-07	
10 / Спізавета Вінчар	Видалення зламаних коренів зубів	10:00	м. Київ вул. Почайна 2а	В обробці	1500	Кеша	2023-11-30	
9 / Анастасія Квітчана	Лікар вузької спеціальності	18:00	м. Київ вул. Івана Франка 16	В процесі	850	Мальфа	2023-11-07	
8 / Катерина Горна	Вакцинація (щорічна)	14:00	м. Київ вул. Івана Франка 16	В обробці	755	Гоша	2023-12-20	
6 / Катерина Горна	Вакцинація	16:00	м. Київ вул. Почайна 2а	В процесі	1234	Гоша	2023-08-30	
4 / Максим Квітка	Вакцинація (щорічна)	13:00	м. Київ вул. Коцюбинського 24	Готово	800	Роза	2023-06-19	

Рис.3.20. Записи на прийом до лікаря

1. Кнопка швидкого повернення на початок сторінки.
2. Додавання запису про запис клієнта на прийом.
3. Видалення відповідного запису.
4. Редагування відповідного запису.
7. Додавання звіту про отримання заробітної плати за працю (рис. 3.21.)

• Заробітня плата для: Вікторія Молода

• Видано: 20000

Зарплата з урахуванням податків: 21000

• Дата видачі: 07.11.2023

Змінити запис

Рис.3.21. Створення запису про отримання заробітної плати працівником

1. Поле, яке неможливо змінити, а дані, які знаходяться в полі введення, вираховуються автоматично.

2. Зміна запису в БД.

8. Додавання запису до БД, про запис на прийом до лікаря (рис.3.22.)

1

- Клієнт: Максим Квітка
- Послуга: Вакцинація
- Дата: дд. мм. гггг
- Статус запису: Готово
- Місцезнаходження клініки: м. Київ вул. Кошобинського 24
- Ціна за послугу:
- Час запису: 09:00

2

Побажання клієнта:

3

Додати запис

Рис.3.22. Створення запису про отримання заробітної плати працівником

1. Випадаючі списки, де необхідно обрати необхідний варіант.
2. Текстове поле для побажань (можна не вводити).
3. Додавання запису до БД.

ВИСНОВКИ

Узагальнення результатів теоретичної та дослідно-експериментальної роботи дає підстави для таких висновків:

1. Аналіз теоретичних та практичних підходів розробки web-орієнтованих систем дозволили визначити, що web-орієнтована система - це система, яка має візуальну складову (унікальний дизайн) та набір певних функцій, які автоматизують, тим самим пришвидшать, взаємодію працівника з оперуванням великої кількості даних.

2. Модель роботи web-орієнтованої системи ветеринарної клініки складається з структур таблиць, опису послідовності роботи системи та алгоритму її роботи.

3. Використання програмно-апаратних засобів сприяло розробці web-орієнтованої системи, а саме: заповнення даними бази даних, написання коду для автоматизації процесів маніпулювання даними, написання скриптів для відображення даних.

4. Побудова моделі web-орієнтованої системи ветеринарної клініки та використання програмно-апаратних засобів допоможуть в подальшій модернізації системи та забезпечити більш якісної автоматизації процесів.

Перспективами подальших досліджень є аналіз клієнтських систем, тобто створення функціонального облікового запису для користувача та написання нових модулів та функцій програми. В роботі вже є заготовки для подальшого розвитку системи, бути доповнена функціями, які пришвидшать роботу персоналу з даними та зроблять можливим клієнту користуватися додатком створеним саме для нього.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Автоматизовані системи [Електронний ресурс]: https://stud.com.ua/21186/informatika/ponyattu_a_automatizovani_sistemi_upravlinnya.
2. Основні етапи веб-розробки [Електронний ресурс]: <https://web-systems.solutions/blog/web-rozrobka-etapy-i-standarty/>.
3. Види сайтів [Електронний ресурс]: <https://web-systems.solutions/blog/english-yaki-vydy-sajtiv-ta-internet-magazyniv-isnuyut/>.
4. Веб-технології. Їх різновиди та функції [Електронний ресурс]: <http://sites.znu.edu.ua/webprog/lect/1170.ukr.html>.
5. Архітектура клієнт-сервер [Електронний ресурс]: <http://educational.mariroz.com/InformTechVInfrastrRynku/lect/lect8.pdf>
6. Архітектура клієнт-сервер [Електронний ресурс]: <http://inter.ptngu.com/kompyuterni-merezhi/arhitektura-kliiyent-server>.
7. Клієнт-серверна архітектура: <https://training.qatestlab.com/blog/technical-articles/client-server-architecture/>.
8. PHP [Електронний ресурс]: <https://dou.ua/lenta/articles/how-to-learn-php/>.
9. Переваги PHP [Електронний ресурс]: <https://hyperhost.ua/uk/wiki/chto-takoe-php>.
10. SQL [Електронний ресурс]: <https://avada-media.ua/ua/sql/>.
11. Тестування [Електронний ресурс]: https://pidru4niki.com/1628011847733/informatika/testuvannya_program_sistem

ДОДАТОК

Програмний код реалізації Web-системи

login.php

```
<?php
    session_start();
    include_once 'top.php';
    $symbols = ['a','z','c','o','e','k',1,2,3,4,5,6,7,8,9,0];
    $is_validate_cap=null;

    $rand_s = [];
    for($i=0;$i<5;$i++)
    {
        $rand_s[] = $symbols[rand(0,count($symbols)-1)];
    }
    $str_s = implode(" ", $rand_s);

    $error_message = "";
    $is_validate_cap = false;
    $is_validate_login = false;
    if(isset($_POST['mail']) && isset($_POST['password']))
    {
        if(isset($_POST['cap']))
        {
            $cap=$_POST['cap'];
            $str_chk=$_POST['str_s'];

            for($i=0;$i<5;$i++)
            {
                if($cap[$i] !=$str_chk[$i]){
                    $is_validate_cap = false;
                    break;
                }
                else
                {
                    //var_dump($cap[$i]);
                    $is_validate_cap = true;
                }
            }
        }
    }
}
```

```

    }
}

if ($is_validate_cap)
{
    $sql = mysqli_query(dbconnect(), "SELECT count(*), email, acl FROM
`employee`
    WHERE `email` = " . $_POST['mail'] . "
    AND `password` = ".MD5($_POST['password'])." LIMIT 1;")
    or die(mysqli_connect_error());
    $row = mysqli_fetch_assoc($sql);
    if($row['count(*)']>0){
        $_SESSION['mail'] = $_POST['mail'];
        $_SESSION['acl'] = $row['acl'];
        $_SESSION["authenticated"] = 'true';
        header("Location: /index.php");exit;
    }
    else{
        echo '<script>alert("Помилка при введенні даних!");</script>';
        session_destroy();
    }
}
else
{
    echo '<script>alert("Помилка при введенні даних!");</script>';
    session_destroy();
}
}
?>
<script>

```

```

function validate(form) {
    fail = validateMail(form.mail.value)
    fail += validatePassword(form.password.value)
    if (fail == "") return true
    else { alert(fail); return false }
}

```

```

    }
    function validateMail(field)
    {
    if (field == "") return "Не введено електронну адресу.\n"
    else if (!((field.indexOf(".") > 0) && (field.indexOf("@") > 0)) || /^[^a-
zA-Z0-9.@_-.]/.test(field)) return "Електронна адреса має неправильний
формат.\n"
    return ""
    }
    function validatePassword(field)
    {
    if (field == "") return "Не введено пароль.\n"
    return ""
    }
</script>
</head>
<body>
    <div class="bg"></div>
    <div class="bg bg2"></div>
    <div class="bg bg3"></div>
    <div class="content">
    <header>
    <div class = "bluetop">
        <a href = "index.php" class = "logo"><img style = "margin-top:1%;
height: 30px;"src = "img/Component 71.png"></img>
        <a href = "index.php" class = "textLogo">VetSite</a>
    </div>
    </header>
    <h1 class="textLogo" id="tcf2">Вхід в особистий кабінет</h1>
    <div class="logcont">
        <form action="login.php" method="post" onSubmit="return
validate(this)">
            <a href="resetpass.php" class="resetpass">Забули
пароль?</a>
            <div class="logrow">
                <label>Введіть логін</label>
                <input id="" class="inputlog" type="email" name="mail">

```

```

</div>
<div class="logrow" id="dpass">
  <label>Введіть пароль<span id="text" style="display:
none;">Caps lock ввімкнено!</span></label>
  <i class="fas fa-eye icon-pass-r" id="icon-pass"
style="position: absolute; left: 93%; top: 48%; cursor: pointer;"></i>
  <input class="inputlog" type="password" id="myInput"
maxlength="12" name="password">
</div>
  <div class="logrow" id="dpass">
    <div class="imagecap">
      <?php foreach($rand_s as $s): ?>
        <span class="s"><?php echo $s;?></span>
      <?php endforeach; ?>
    </div>
  </div>
  <div class="logrow" id="dpass">
    <input class="inputlog" type="text" placeholder="Символи з
картинки" name="cap" required minlength="5" maxlength="5"
autocomplete="off">
    <input type="hidden" name="str_s" value="<?php echo $str_s;
?>">
  </div>
  <div class="logrow textcenter">
    <input class="button" type="submit" value="Вхід">
  </div>
</form>
</div>
</body>
</html>
<script>
var input = document.getElementById("myInput");
var text = document.getElementById("text");

input.addEventListener("keyup", function(event) {
  if (event.getModifierState("CapsLock")) {
    text.style.display = "inline-block";
  }
});

```

```
text.style.color = "red";
text.style.margin = "0 0 0 10%";
} else {
text.style.display = "none"
}
});
const inputPass = document.getElementById('myInput');
const iconPass = document.getElementById('icon-pass');
iconPass.addEventListener('click', () =>{
    if(inputPass.getAttribute('type') === 'password'){
        inputPass.setAttribute('type', 'text');
    }
    else{
        inputPass.setAttribute('type', 'password');
    }
});
</script>
```

emp.php

```
<?php
include 'top.php';
//if(($_SESSION['acl'] == '0') || ($_SESSION['acl'] == 1)){
$sql = "SELECT *, `employee`.`descrip` as `description` FROM `employee`
JOIN `tworkemp` ON `employee`.`id_twork` = `tworkemp`.`id_twork`
JOIN `service` ON `employee`.`id_serv` = `service`.`id_serv`
JOIN `position` ON `employee`.`id_position` = `position`.`id_position` ORDER
BY `id_emp` ASC";
$result = mysqli_query(dbconnect(), $sql);
$data = [];

if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {
        $data[] = $row;
    }
} else {
    echo "0 results";
}
```



```

mysqli_close(dbconnect());
?>
<div class="container">
<?php foreach ($data as $semp): ?>
<div class="contleft">
<div class="left">
<div class="contimg">

</div>
<?php
if(($_SESSION['acl'] == '0') || ($_SESSION['acl'] == 1)){
echo '<a href="edit_emp.php?id='.$semp["id_emp"].'" class="ai edit">
<i class="fas fa-regular fa-user-pen acti" title="Змінити"></i>
</a>
<a href="delete_emp.php?id='.$semp["id_emp"].'" class="ai del">
<i class="fas fa-solid fa-user-xmark acti" title="Видалити"></i>
</a>';
}else{
}
?>
<p>Код працівника: <?php echo $semp['id_emp']; ?></p>
</div>
<div class="conttext">
<p class="txtspan" id="txtspan">Ім`я</p>
<p class="txtspan"><?php echo $semp['name']; ?></p>
<p class="txtspan" id="txtspan">Прізвище</p>
<p class="txtspan"><?php echo $semp['lastname']; ?></p>
<p class="txtspan" id="txtspan">Пошта</p>
<p class="txtspan"><?php echo $semp['email']; ?></p>
<p class="txtspan" id="txtspan">Посада</p>
<p class="txtspan"><?php echo $semp['position']; ?></p>
<p class="txtspan" id="txtspan">Телефон</p>
<p class="txtspan"><?php echo $semp['telephone']; ?></p>
<p class="txtspan" id="txtspan">Послуга</p>
<p class="txtspan"><?php echo $semp['name_serv']; ?></p>
<p class="txtspan" id="txtspan">Приймальні дні</p>
<p class="txtspan"><?php echo $semp['name_twork']; ?></p>
<p class="txtspan" id="txtspan">Про працівника</p>

```

```

<p class="txtspan"><?php echo $semp['description']; ?></p>
<p class="txtspan" id="txtspan">Рівень доступу</p>
<p class="txtspan"><?php echo $semp['acl']; ?></p>
</div>
</div>
<?php endforeach; ?>
</div>

<?php
include_once 'footer.php';
//}else{
//    header("Location: /login.php");exit;
// }
?>

<?php
session_start();
include_once 'config.php';
include 'functions.php';
$sql = "SELECT * FROM `employee`";
$result = mysqli_query(dbconnect(), $sql);
$semp1 = [];
if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {
        $semp1[] = $row;
    }
} else {
    echo "0 results";
}
if(isset($_POST['search_s'])){
    $text = $_POST['search_s'];
    $text = trim(strip_tags(strip_slashes(htmlspecialchars($text))));

    $sql = "SELECT * FROM `service`
        JOIN `material` ON `service`.`id_material` =
        `material`.`id_material`";

```

```
WHERE `name_serv` LIKE '%$text%' OR `name` LIKE  
 '%$text%';";
```

```
$result = mysqli_query(dbconnect(), $sql);
```

```
$data = [];
```

```
if (mysqli_num_rows($result) > 0) {
```

```
    while($row = mysqli_fetch_assoc($result)) {
```

```
        $data[] = $row;
```

```
    }
```

```
}
```

```
}
```

```
if(isset($_POST['search_m'])){
```

```
    $text = $_POST['search_m'];
```

```
    $text = trim(strip_tags(stripslashes(htmlspecialchars($text))));
```

```
$sql = "SELECT * FROM `material`
```

```
JOIN `category` ON `material`.`id_cat` = `category`.`id_cat`
```

```
JOIN `avail` ON `material`.`id_avail` = `avail`.`id_avail`
```

```
JOIN `provider` ON `material`.`id_provider` =
```

```
`provider`.`id_provider` WHERE `material`.`name` LIKE '%$text%' OR
```

```
`material`.`descrip_m` LIKE '%$text%' OR `price` LIKE '%$text%'
```

```
OR `provider`.`name_p` LIKE '%$text%';";
```

```
$result = mysqli_query(dbconnect(), $sql);
```

```
$data = [];
```

```
if (mysqli_num_rows($result) > 0) {
```

```
    while($row = mysqli_fetch_assoc($result)) {
```

```
        $data[] = $row;
```

```
    }
```

```
}
```

```
}
```

```
if(isset($_POST['search'])){
```

```
    $text = $_POST['search'];
```

```
    $text = trim(strip_tags(stripslashes(htmlspecialchars($text))));
```

```

        $sql = "SELECT *, `employee`.`descrip` as `description` FROM
`employee`
        JOIN `tworkemp` ON `employee`.`id_twork` =
`tworkemp`.`id_twork`
        JOIN `service` ON `employee`.`id_serv` = `service`.`id_serv`
        JOIN `position` ON `employee`.`id_position` =
`position`.`id_position` WHERE `employee`.`name` LIKE '%$text%' OR
`employee`.`lastname` LIKE '%$text%' OR `email` LIKE '%$text%'
        OR `position`.`position` LIKE '%$text%';";
        $result = mysqli_query(dbconnect(), $sql);
        $data = [];
        if (mysqli_num_rows($result) > 0) {
            while($row = mysqli_fetch_assoc($result)) {
                $data[] = $row;
            }
        }
    }
    ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>VetSite</title>
    <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet" href="fontawesome-free-6.3.0-web/fontawesome-
free-6.3.0-web/css/all.min.css">
    <!--<link rel="stylesheet" href="bootstrap-5.3.0-
alpha1/dist/css/bootstrap.min.css"-->
    <!--<script rel="stylesheet" src="bootstrap-5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js"></script-->
    <script defer src="js/simple-adaptive-slider.js"></script>
    <script type="text/javascript" src="js/jquery-3.6.0.min.js"></script>
    <script type="text/javascript" src="js/script.js"></script>
    <script src="js/jquery-3.2.1.js"></script>
    <script src="js/jquery.mask.js"></script>
    <script src="js/custom.js"></script>
</head>

```

<body>

profile.php

```
<?php
include_once "top.php";
$sql = "SELECT * FROM `reception` JOIN `client` ON `reception`.`id_client` =
`client`.`id_client`
JOIN `clientspet` ON `clientspet`.`id_pet` = `client`.`id_pet`
JOIN `service` ON `reception`.`id_serv` = `service`.`id_serv`
INNER JOIN `status` ON `reception`.`id_status` = `status`.`id_status`
JOIN `local` ON `reception`.`id_loc` = `local`.`id_loc`
JOIN `timentry` ON `reception`.`id_zap` = `timentry`.`id_zap`
WHERE `reception`.`id_rec` LIMIT 10";
$result = mysqli_query(dbconnect(), $sql);
$recept = [];

if(isset($_SESSION['mail'])){
    $sql = mysqli_query(dbconnect(),"
        SELECT * FROM `employee`
        WHERE `email` = '".$_SESSION['mail']."'";
    ") or die(mysqli_connect_error());
    $row = mysqli_fetch_assoc($sql);
    if (mysqli_num_rows($result) > 0) {
        while($row = mysqli_fetch_assoc($result)) {
            $recept[] = $row;
        }
    } else {
        echo "0 results";
    }

    $sql = "SELECT * FROM `timework`";
    $result = mysqli_query(dbconnect(), $sql);
    $tw = [];

    if (mysqli_num_rows($result) > 0) {
        while($row = mysqli_fetch_assoc($result)) {
            $tw[] = $row;
        }
    }
}
```

```

}
} else {
    echo "0 results";
}

if (isset($_GET['id']) and !empty($_GET['id']))
{
    $sid = (int)trim($_GET['id']);
    $sql = "SELECT * FROM `employee`
JOIN `tworkemp` ON `employee`.`id_twork` = `tworkemp`.`id_twork`
JOIN `position` ON `employee`.`id_position` = `position`.`id_position`
WHERE `id_emp`=" . $sid;
    $result = mysqli_query(dbconnect(), $sql);
    $data = mysqli_fetch_assoc($result);
    mysqli_close(dbconnect());

    if (mysqli_num_rows($result) > 0) {
        while($row = mysqli_fetch_assoc($result)) {
            $data[] = $row;
        }
    } else {
        echo "0 results";
    }
    mysqli_close(dbconnect());
}
?>
<div class="container">
<div class="empcont">
<aside id="asideprofile">

<div class="profcont">
<div class="prdiv">
<p id="rrr">Про себе: <?php echo $data['descrip'] ?></p>
</div>
<div class="prdiv">
<ul style="display: flex; text-align:center;">
<li style="width:33.3%; list-style: none;">

```

```

<label>Ім`я:</label>
<p><?php echo $data['name'];?></p>
</li>
<li style="width:33.3%; list-style: none;">
<label>Прізвище:</label>
<p><?php echo $data['lastname'];?></p>
</li>
<li style="width:33.3%; list-style: none;">
<label>Дата народження:</label>
<p><?php echo $data['date_birth'];?></p>
</li>
</ul>
</div>
<div class="prdiv">
<ul style="display: flex; text-align:center;">
<li style="width:33.3%; list-style: none;">
<label>Електронна пошта:</label>
<p><?php echo $data['email'];?></p>
</li>
<li style="width:33.3%; list-style: none;">
<label>Номер телефону:</label>
<p><?php echo $data['telephone'];?></p>
</li>
<li style="width:33.3%; list-style: none;">
<label>Посада:</label>
<p><?php echo $data['position'];?></p>
</li>
</ul>
</div>
</div>
<input type="hidden" name="id" value="<?php echo $data['id_user']; ?>">
</aside>
<main id="mainprofile">
<div class="profedim">
<figure>


```

```

<figcaption>Ідентифікаційний код працівника: <?=$data['id_emp']?></figcaption>
</figure>
</div>
<p class="graph">Графік роботи</p>
<table style="margin: 2%;">
  <tr>
    <td>
      <span>День тижня</span>
    </td>
    <td>
      <span>Перша зміна</span>
    </td>
    <td>
      <span>Друга зміна</span>
    </td>
  </tr>
  <? foreach($tw as $twk): ?>
    <tr>
      <td>
        <span><?=$twk['name_day'] ?></span>
      </td>
      <td>
        <span><?=$twk['first'] ?></span>
      </td>
      <td>
        <span><?=$twk['second'] ?></span>
      </td>
    </tr>
  <? endforeach; ?>
</table>
</main>
</div>
<p class="graph">Усі записи клієнтів на прийом до спеціалістів</p>
<table>
<caption></caption>
<thead>
<tr>

```



```

<th rowspan="2" class="first">№ / Клієнт</th>
<th rowspan="2">Послуга</th>
<th colspan="3">Додаткова інформація</th>
<th rowspan="2">Ціна</th>
<th rowspan="2">Тварина</th>
<th rowspan="2">Дата</th>
</tr>
<tr>
<td class="first">Час запису</td>
<td class="first">Місцезнаходження</td>
<td class="first">Статус</td>
</tr>
</thead>
<tbody>
<?php foreach ($receipt as $d):?>
<tr>
<td><?= $d['id_rec'];?> / <?= $d['name'], " ", $d['lastname'];?>
<td><?= $d['name_serv'];?></td>
<td><?= substr($d['time'],0,-10);?></td>
<td><?= $d['adress'];?></td>
<td><?= $d['name_st'];?></td>
<td><?= $d['price'];?></td>
<td><?= $d['name_p'];?></td>
<td><?= $d['date'];?></td>
</tr>
<?php endforeach;?>
</tbody>
</table>
</div>
<?php
include_once 'footer.php';
?>
<?php }else{
    header("Location: /login.php");exit;
}
?>

```

e-mail.php

```
<?php
session_start();
include_once 'config.php';
include_once 'top.php';

if (isset($_POST['submit'])){

    $to = 'evgenlastovka@gmail.com';
    $subject = $_POST['topic'];
    $message = $headers . $_POST['message'] . "\r\n";
    $message = htmlspecialchars($_POST['message']);
    $message = urldecode($message);
    $message = trim($message);
    $headers = 'Повідомлення від користувача /->'. $_POST['email'] . "\r\n".
'Tекст повідомлення:';

    mail($to, $subject, $message, $headers);
}
?>

<body>
<script>
function quest (f) {
    if(confirm("Ви впевнені, що хочете надіслати
цього листа?"))
        f.submit();
}
function del (d){
    if(confirm("Ви впевнені, що хочете стерти дані з
усіх полей?"))
        d.reset();
}
</script>
```

```
<p class="pname textLogo" id="pnamemail">Залишити коментар чи побажання</p>
```

```
<div class="mailCont">
```

```
<form action="e_mail_user.php" method="post" onsubmit="quest(this); return false;" onreset="del(this); return false;">
```

```
<div class="inpMail">
```

```
<i class="fas fa-envelope" data-tooltip="<?=/ *$ _SESSION['mail']; */ "Введіть електронну пошту?">" ></i> <!--/*еще подумать*/-->
```

```
<input name="email" type="email" autocomplete="off" placeholder="Введіть логін" required >
```

```
</div>
```

```
<div class="inpMail">
```

```
<i class="fas fa-sms"></i>
```

```
<input name="topic" type="text" autocomplete="off" placeholder="Введіть тему повідомлення" required >
```

```
</div>
```

```
<textarea name="message" type="text" autocomplete="off" placeholder="Напишіть Ваші побажання, а ми залюбки прочитаємо їх та прийmemo до уваги!"></textarea>
```

```
<div class="inpMail">
```

```
<input class="res" type="reset" name="reset" value="Стерти все">
```

```
<input class="tsbut" type="button" value="На головну" onClick='location.href="/index.php"'>
```

```
<input class="subm" type="submit" name="submit" value="Надіслати">
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</body>
```

```
</html>
```

```
<?php
```

```
include 'footer.php';//}else{
```

```
header("Location: /login.php");exit;
?>
//
```

