

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавра

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ
ЗООТОВАРИВ
SOFTWARE IMPLEMENTATION OF ONLINE PET STORE

Виконав: студент(ка) 4 курсу, групи КНТ-220сп

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

ГРИНЬ І.А.

(ПРІЗВИЩЕ та ініціали)

Керівник ФЕДОРОНЧАК Т.В.

(ПРІЗВИЩЕ та ініціали)

Рецензент ТЕРЕЩЕНКО Е.В.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2023 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

ГРИНЯ Іллі Андрійовича
(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація інтернет-магазину зоотоварів.
Software Implementation of Online Pet Store

керівник проєкту (роботи) к.т.н., доцент ФЕДОРОНЧАК Тетяна Василівна
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від « 4 » квітня 2023 року № 87

2. Строк подання студентом проєкту (роботи) 29 травня 2023

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1) Аналіз предметної області 2) Розробка архітектури вебзастосунку 3) Реалізація вебзастосунку 4) Експлуатація і тестування вебзастосунку

5). Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ФЕДОРОНЧАК Т.В., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю. ст. викладач		

7. Дата видачі завдання « 11 » квітня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2 тиждень	Розділ 1
3	Розробка архітектури програми	3 тиждень	Розділ 2
4	Розробка програми	4-5 тиждень	Розділ 3
5	Тестування та експериментальне дослідження програми	6 тиждень	Розділ 4
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	7 тиждень	Додатки
7	Захист роботи	8 тиждень	

Студент

_____ Ілля ГРИНЬ
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Тетяна ФЕДОРОНЧАК
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 103 с., 72 рис., 2 табл., 3 дод., 20 джерел.

ВЕБСАЙТ, INTERNET, BACKEND, FRONTEND, ВЕБЗАСТОСУНОК, JAVASCRIPT, EXPRESS, NODE, EXPRESS, HTML, CSS, SCSS, PUG, MySQL.

Об'єкт роботи – програмне забезпечення для продажу зоотоварів.

Предмет роботи – вебсайт для продажу зоотоварів.

Мета роботи – розробити інтернет магазин зоотоварів з можливістю їх замовлення.

Матеріали, методи та технічні засоби: JavaScript, HTML, CSS, Express, препроцесори: SCSS, PUG, база даних: MySQL.

Результати. Розробку програмного забезпечення магазину зоотоварів виконано за допомогою вебфреймворку Express з визначенням структури даних, вимог до програми, розробленням структури й алгоритму роботи програми на основі цих вимог, реалізацією відповідних одиниць. Програма дозволяє переглядати каталог товарів, сортувати товари за категоріями та ціною або використати пошук за назвою товару, додати бажаний товар до кошика, користувач може оформити замовлення заповнивши форму зворотного зв'язку.

Висновки. Розроблене програмне забезпечення дозволяє підвищити зручність підбору та покупки зоотоварів онлайн.

Галузь використання – Програмне забезпечення може використовуватися як корисний вебмагазин для підбору та купівлі товарів для тварин.

ABSTRACT

Explanatory note to the diploma qualification work bachelor's: 103 pages, 72 figures, 2 tables, 3 appendices, 20 sources.

WEBSITE, INTERNET, BACKEND, FRONTEND, WEB APPLICATION, JAVASCRIPT, EXPRESS, NODE, EXPRESS, HTML, CSS, SCSS, PUG, MySQL.

The object of work is software for the sale of pet products.

The subject of the work is websites for the sale of pet products. Materials, methods and technical means: JavaScript, HTML, CSS, Express, preprocessors: SCSS, PUG, database: MySQL.

The results. The software development of the pet store was carried out using the Express webframework with the definition of the data structure, the requirements for the program, the development of the structure and algorithm of the program based on these requirements, and the implementation of the corresponding units. The program allows you to view the product catalog, sort products by category and price or use the search by product name, the user can place an order by filling out the feedback form, add the desired product to the cart.

Conclusions. The developed software makes it possible to increase the convenience of selecting and buying pet products online.

Field of use. The software can be used as a useful webstore for selecting and buying pet products.

ЗМІСТ

Перелік скорочень та умовних познач	8
Вступ	10
1 Аналіз предметної області	11
1.1 Покупки в Інтернеті	11
1.2 Огляд програмних аналогів	12
1.3 Висновки за розділом 1	15
2 Розробка архітектури вебзастосунку	16
2.1 Вибір засобів проєктування і розробки	16
2.2 Визначення вимог до програмного забезпечення	17
2.3 Розробка інтерфейсу сайту інтернет-магазину зоотоварів	21
2.4 Проєктування бази даних	24
2.5 Визначення структури програми	29
2.6 Висновки за розділом 2	31
3 Реалізація вебзастосунку	32
3.1 Алгоритм функціонування програми	32
3.2 Алгоритм роботи кошика	39
3.3 Висновки за розділом 3	49
4 Експлуатація і тестування вебзастосунку	50
4.1 Опис програми	50
4.2 Експлуатація програми	50
4.3 Тестування вебзастосунку	53
4.4 Висновки за розділом 4	54
Перелік джерел посилання	55

Додаток А Технічне завдання	57
Додаток Б Текст програми	61
Додаток В Слайди презентації.....	95

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– (англ. Application Programming Interface, API) — набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення;
CSS	– (англ. Cascading Style Sheets «каскадні таблиці стилю») формальна мова оформлення та опис у зовнішньому вигляді документа (вебсторінки), написана з використанням мови розмітки (перш за все HTML або XHTML);
EXPRESS	– Express.js, або просто Express - програмний каркас розробка серверної частини вебзастосунків для Node.js;
HTML	– (англ. HyperText Markup Language - мова розмітки гіпертексту) стандартизована мова розмітки документів для перегляду вебсторінок у браузері;
HTTP	– (англ. Hyper Text Transfer Protocol) - протокол передачі даних, що використовується в комп'ютерних мережах;
JSON	– (англ. JavaScript Object Notation) - текстовий формат обміну даними, заснований на JavaScript;
MySQL	-вільна система керування реляційними базами даних, яка була розроблена для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом;
NODE.JS	– програмна платформа, заснована на двигуні V8, що перетворює JavaScript з вузькоспеціалізованої;
PUG	– це препроцесор HTML та шаблонизатор, який був написаний на JavaScript для Node;
REST	– це спосіб взаємодії сайтів та вебдодатків із сервером;

- SASS – (англ. Syntactically Awesome Stylesheets) - скриптова метамова, яка інтерпретується в каскадні таблиці стилів (CSS);
- WEBSITE – (англ. website, від web (веб) і site (місце)) сукупність вебсторінок та залежного вмісту, доступних у мережі Інтернет, які об'єднані як за змістом, так і за навігацією під єдиним доменним ім'ям;
- WEBSOCKET – протокол зв'язку поверх TCP-з'єднання, призначений для обміну повідомленнями між браузером та вебсервером у режимі реального часу;
- БД – база даних;
- ПЗ – програмне забезпечення.

ВСТУП

В наш час Інтернет стає все більш розвиненим середовищем для здійснення комунікацій. У зв'язку з глобальним розвитком мережі Інтернет та попиту на інформаційні сайти в програмуванні все більш різко почала виділятися окрема його галузь вебпрограмування.

Зараз, щоб привернути увагу клієнтів, покупців або партнерів, просто необхідно заявити про себе в інтернеті, шляхом створення вебсайту. Для цих цілей якраз і служить вебсайт, що містить основну інформацію про організацію, приватну особу, компанію, товари або послуги, контактні дані. Сайти дозволяють зберігати, передавати, продавати різні типи інформації, не відходячи від екрана комп'ютера.

Розроблений вебсайт в мережі Інтернет забезпечить приплив нових користувачів, оскільки добре зроблений сайт буде достатньо просто знайти за допомогою пошукових систем. За допомогою сайту можна домогтися швидкого реагування на побажання користувачів, а також вносити відповідні зміни до вебсайту. Крім цього, вебсайт дозволить знизити витрати на рекламу.

Дипломний проєкт повинен бути реалізований за допомогою сучасних програмних засобів. Основна задача – видати користувачеві зручний сайт з актуальними товарами. Повинен бути створений легкий в освоєнні інтерфейс для забезпечення комфортного перебування клієнта на сайті.

Ціллю дипломного проєкту є створення інтернет магазину зоотоварів. Реалізуватися це повинно за допомогою простого інтерфейсу, зрозумілого для рядових користувачів

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Покупки в Інтернеті

Зростання кількості користувачів Інтернету та зростання інтернет-магазинів тісно взаємопов'язані, оскільки збільшення кількості користувачів інтернету сприяє швидкому розширенню онлайн-покупок. Широка доступність високошвидкісного Інтернету та поширення мобільних пристроїв полегшили людям доступ до онлайн-магазинів з будь-якого місця та в будь-який час.

Огляд глобального цифрового стану на 2023 рік (дані взяті з сайту <https://datareportal.com/reports/digital-2023-global-overviewreport>) представлено на рис. 1.1.



Рисунок 1.1 – Глобальний цифровий стан

Розглянемо основні показники:

- населення світу 2023 року перевищило 8,01 мільярда;
- на початку 2023 року мобільними телефонами користувалися 5,44 мільярда людей, що дорівнює 68 відсоткам загального населення світу;

- сьогодні у світі налічується 5,16 мільярда користувачів Інтернету, тобто 64,4 відсотка всього населення планети зараз є онлайн;
- зараз у світі налічується 4,76 мільярда користувачів соціальних мереж, що становить трохи менш як 60 відсотків населення планети;
- оскільки все більше людей отримують доступ до Інтернету, потенційна клієнтська база онлайн-магазинів продовжує зростати. Крім того, легкість і зручність онлайн-покупок зробили його популярним вибором для споживачів, які все частіше звертаються до онлайн-магазинів.

1.2 Огляд програмних аналогів

Як було проаналізовано раніше, продаж товарів через інтернет є одним з найпопулярніших видів збуту товарів, багато вебсайтів які надають можливість купівлі товарів.

У цьому підрозділі вони будуть розглянуті в якості програмних аналогів і проаналізовані щодо можливостей які надають клієнту.

Першим інтернет-магазином для тварин буде masterzoo.ua представлено на рис. 1.2.

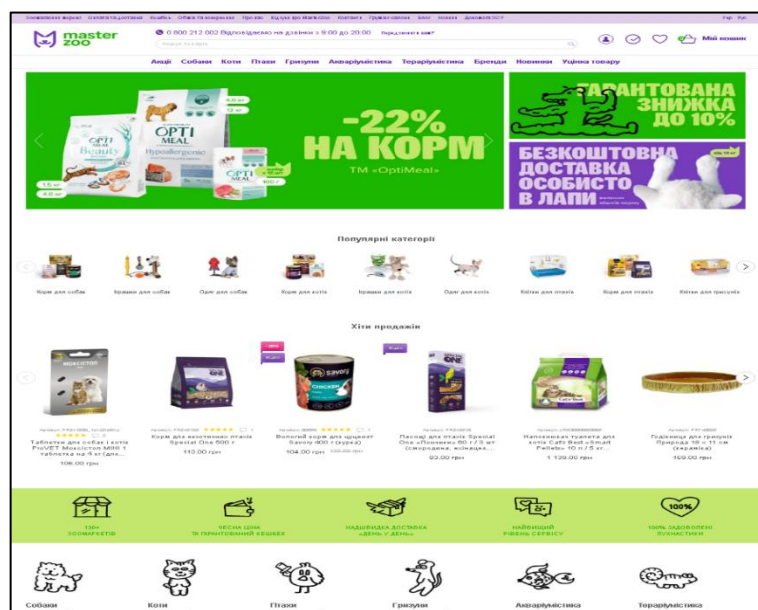


Рисунок 1.2 – Інтернет-магазин masterzoo

Інтернет магазин masterzoo має такий функціонал:

- можливість перегляду асортименту товарів;
- можливість реєстрації/авторизації користувача;
- додавання товарів у кошик;
- сортування та пошук товарів;
- зміна мови локалізації сайту;
- зв'язок онлайн з менеджером.

До мінусів слід віднести:

- не повну масштабованість сайту;
- відсутність інтерфейсу для мобільних телефонів.

Другим аналогом є вебсайт zoomark.com.ua представлено на рис. 1.3.

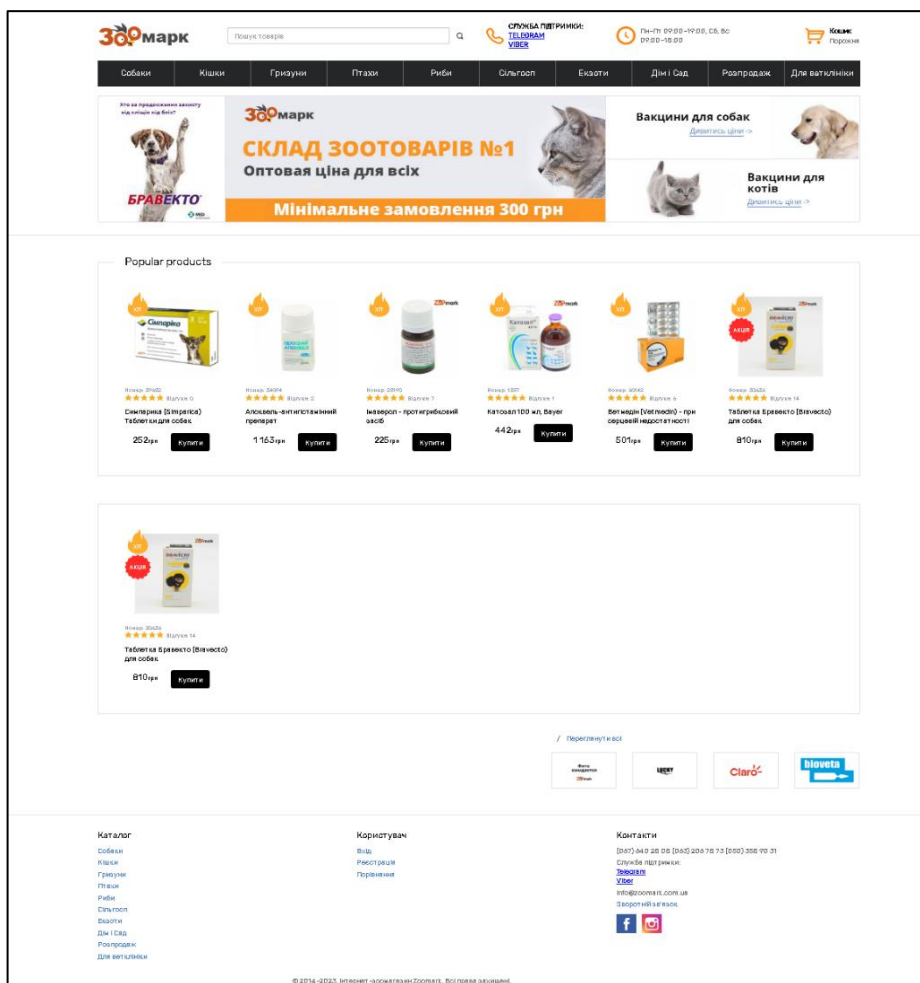


Рисунок 1.3 – Інтернет-магазин zoomark

Інтернет магазин zoomark має такий функціонал:

- можливість перегляду асортименту товарів;
- можливість реєстрації/авторизації користувача;
- додавання товарів у кошик;
- сортування та пошук товарів;
- зміна мови локалізації сайту.
- зв'язок онлайн з менеджером;

До мінусів інтернет-магазину:

- не повну масштабованість сайту;
- не сучасний дизайн.

Інший аналог подібного який можна розглянути типу інтернет-магазину для тварин є petmarket.ua представлено на рис. 1.4.

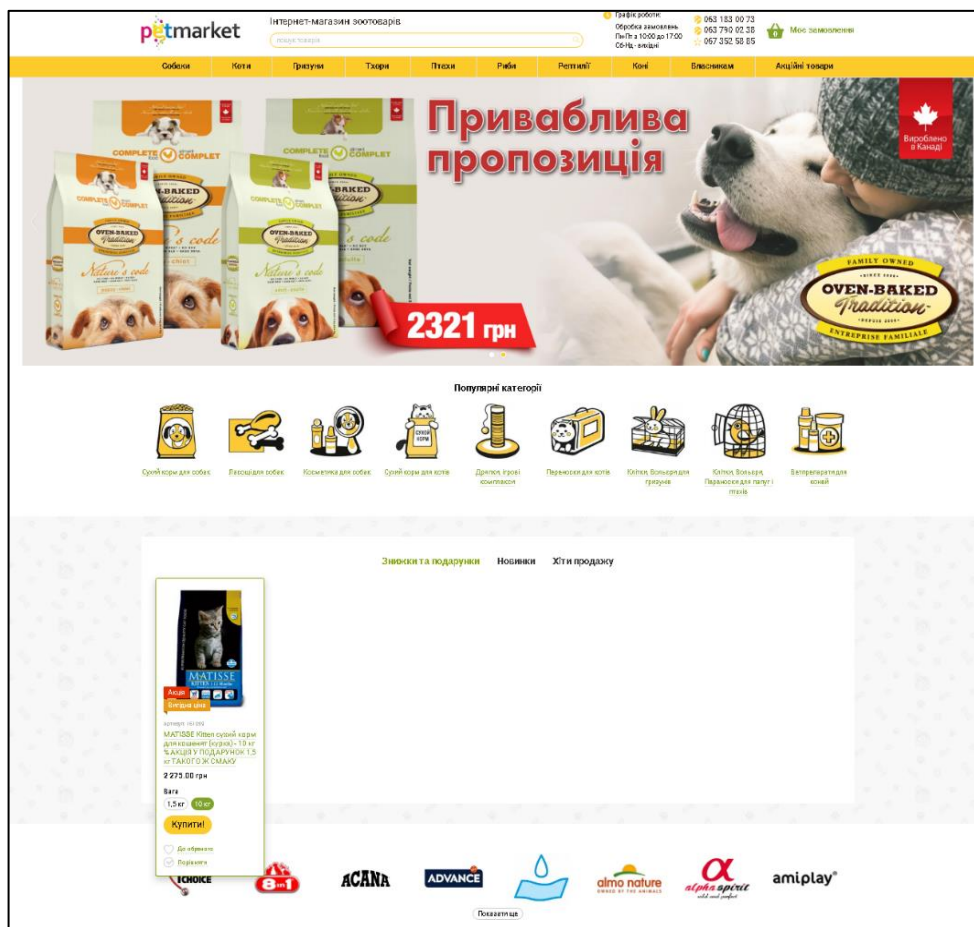


Рисунок 1.4 – Інтернет-магазин petmarket.ua

Інтернет магазин petmarket має такий функціонал:

- можливість перегляду асортименту товарів;
- можливість реєстрації/авторизації користувача;
- додавання товарів у кошик;
- сортування та пошук товарів;
- зміна мови локалізації сайту.

До недоліків цього магазину зоотоварів слід віднести:

- відсутність зв'язку з менеджером;
- не сучасний дизайн.

1.3 Висновки за розділом 1

Зростання кількості онлайн-магазинів і споживачів, які використовують Інтернет для покупок, призвело до значного розвитку електронної комерції. Електронна комерція стала однією з найшвидше зростаючих галузей бізнесу, яка надає споживачам можливість швидко та зручно здійснювати покупки в Інтернеті. Крім того, вона також дозволяє підприємствам зменшувати витрати на оренду та обслуговування фізичних магазинів, що забезпечує їм більшу прибутковість.

За даними звіту "Global E-commerce Report 2023" компанії RetailMeNot та Centre for Retail Research. Очікується, що зростання електронної комерції буде продовжуватися в майбутньому, оскільки все більше споживачів віддають перевагу зручності та доступності онлайн-покупок.

Було також здійснено перевірку програмних аналогів, проведений аналіз показав, що деякі з цих магазинів мають певні недоліки, такі як відсутність можливості повного масштабування, відсутність онлайн менеджера, застарілий або не зручний інтерфейс.

В результаті аналізу предметної області та аналізу аналогів було створено завдання для розробки програмних систем, що наведено в додатку А.

2 РОЗРОБКА АРХІТЕКТУРИ ВЕБЗАСТОСУНКУ

2.1 Вибір засобів проєктування і розробки

Так як створення цього застосунку має свої особливості, які включають роботу з невеликими наборами даних, такими як товари, замовлення та користувачі, з метою зберігання та обробки цих даних, розумним вибором для розробки основної мови програмування є Javascript, яка є дуже популярною та ефективною. JavaScript (JS) – це легковажна, інтерпретована або JIT-компільована, об'єктно-орієнтована мова з функціями першого класу [1].

Згідно сукупності показників, після уважного аналізу, було прийняте рішення вибрати Express за сукупністю показників наведених у табл.2.1 серед альтернативних варіантів, таких як Koa і Vue.

Таблиця 2.1 – Вибір фреймворку розробки вебзастосунку

Критерії порівняння	Express	Koa	Vue
Зручність робота з даними	Так	Так	Так
Надає вам більше корисних інструментів з коробки	Так	Так	Ні
Величезне ком'юніті та підтримка фреймворка	Так	Так	Так
Просте налаштування роутингу	Так	Так	Так
Швидкість роботи	Так	Ні	Так

2.2 Визначення функціональних вимог до програмного забезпечення

Щоб забезпечити зручну роботу, програмне забезпечення для інтернет-магазину зоотоварів повинно мати широкий спектр функціональних можливостей, які мають використовуватися для зручної роботи:

- пошук товарів за найменуванням;
- перегляд усіх товарів;
- сортування товарів за категоріями;
- сортування товарів за ціною;
- детальний перегляд товару;
- додавання товарів до кошика;
- отримання повідомлення на електронну пошту при замовленні;
- додавання або редагування товарів Адміном;
- додавання або редагування товарів в кошику;
- можливість зв'язку з менеджером;
- авторизація та реєстрація користувача;
- можливість підтвердити замовлення;
- можливість відмінити замовлення;
- контактна інформація;
- видалення користувача.

Відповідно до специфіки функціональних можливостей визначаються сценарії дій користувача представлено на рис. 2.1:

- пошук товару за найменуванням;
- перегляд усіх товарів;
- сортування товарів за категоріями;
- сортування товарів за ціною;
- детальний перегляд товару;
- додавання товарів до кошика;
- додавання або редагування товарів в кошику;

- отримання повідомлення на електронну пошту при замовленні;
- можливість зв'язку з менеджером;
- авторизація та реєстрація користувача;
- можливість підтвердити замовлення;
- контактна інформація.

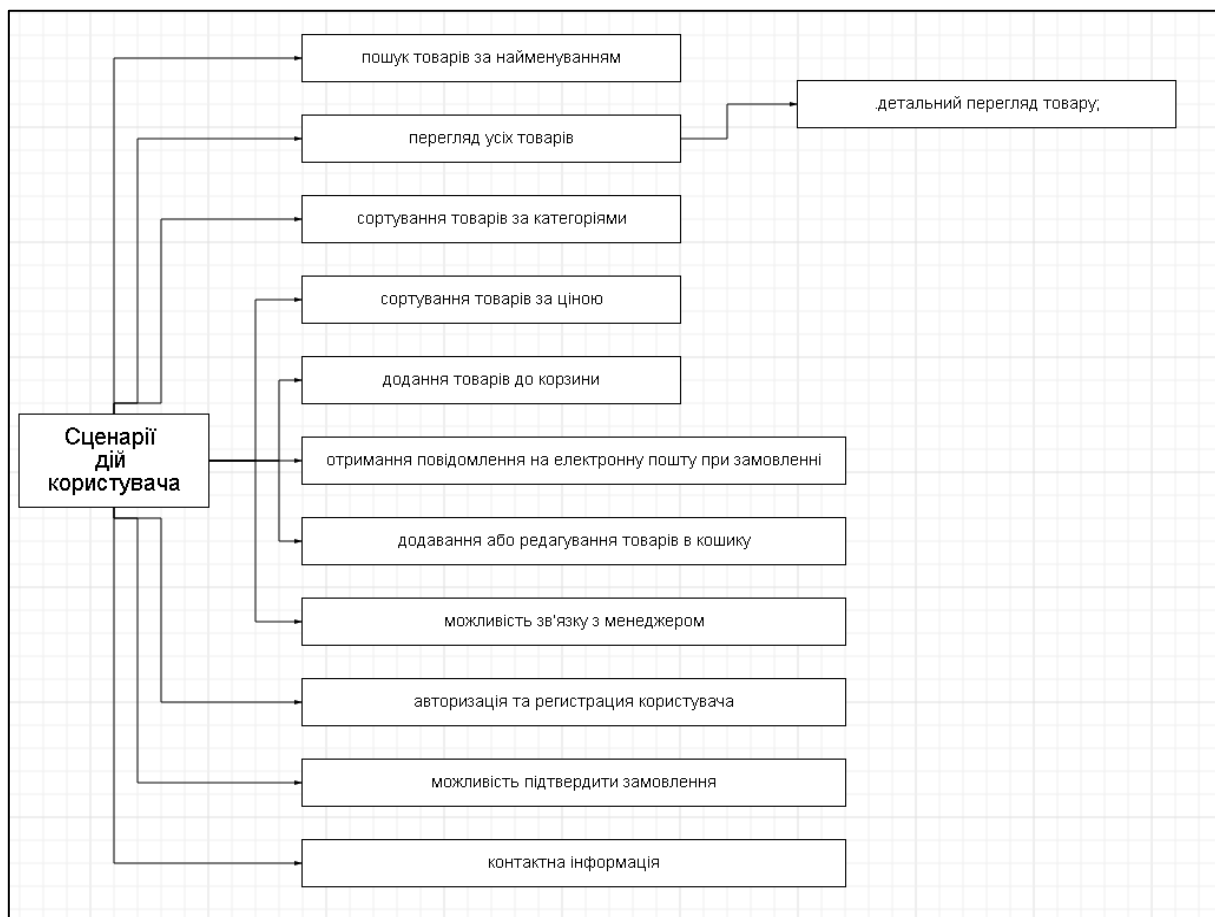


Рисунок 2.1 – Діаграма сценаріїв дій користувача

Сценарії дій менеджера представлено на рис. 2.2:

- пошук товару за найменуванням;
- перегляд усіх товарів;
- сортуння товарів за категоріями;
- сортуння товарів за ціною;
- детальний перегляд товару;
- додання товарів до коззини;

- додавання або редагування товарів в кошику;
- отримання повідомлення на електронну пошту при замовленні;
- авторизація та реєстрація користувача;
- можливість підтвердити замовлення;
- можливість відмінити замовлення;
- контактна інформація.

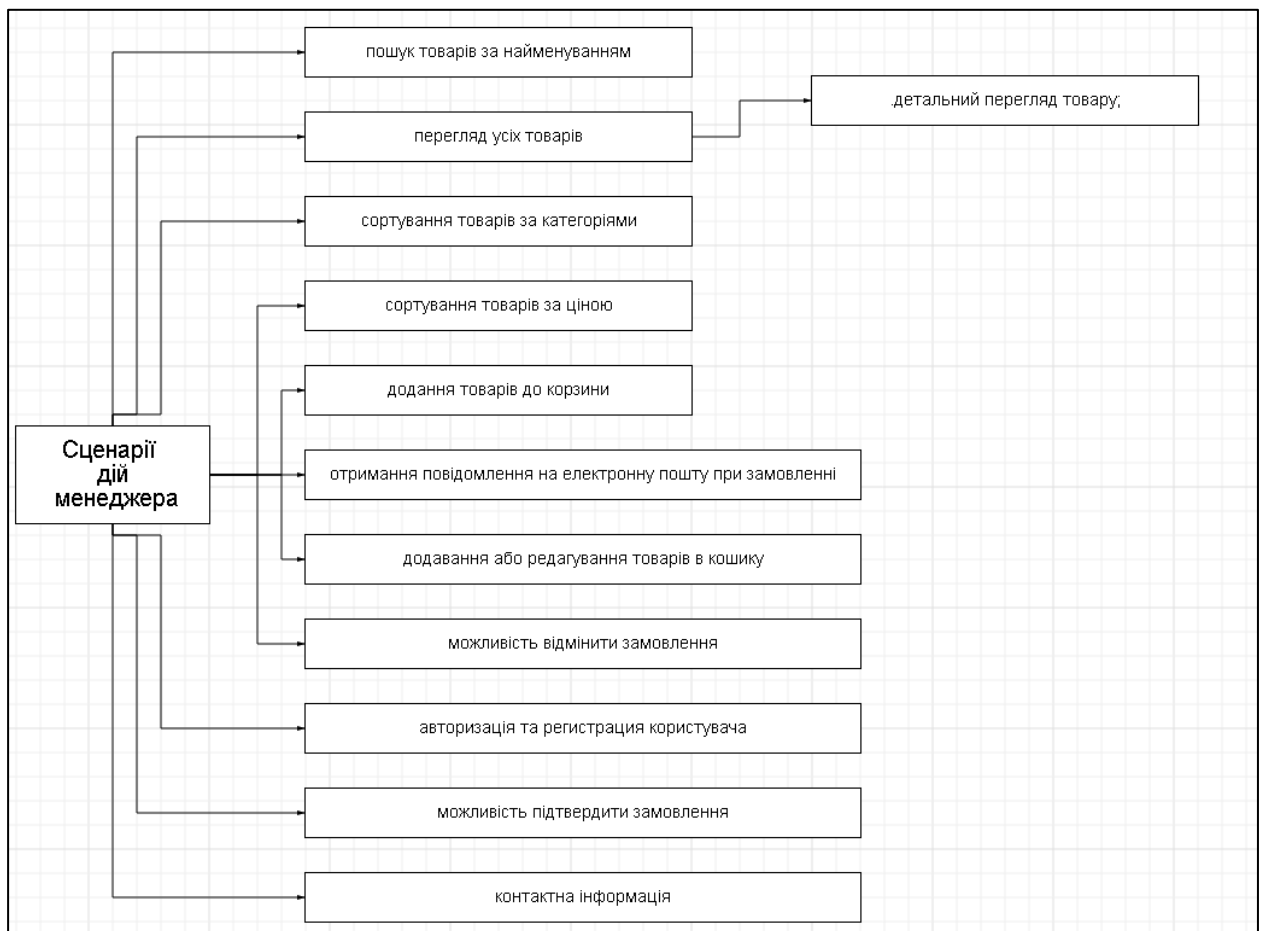


Рисунок 2.2 – Діаграма сценаріїв дій менеджера

Сценарії дій адміністратора представлено на рис. 2.3:

- пошук товару за найменуванням;
- перегляд усіх товарів;
- сортування товарів за категоріями;
- сортування товарів за ціною;
- детальний перегляд товару;

- додання товарів до корзини;
- додавання або редагування товарів в кошику;
- отримання повідомлення на електронну пошту при замовленні;
- авторизація та реєстрація користувача;
- можливість підтвердити замовлення;
- можливість відмінити замовлення;
- контактна інформація;
- видалення користувача;
- додавання або редагування товарів Адміном.

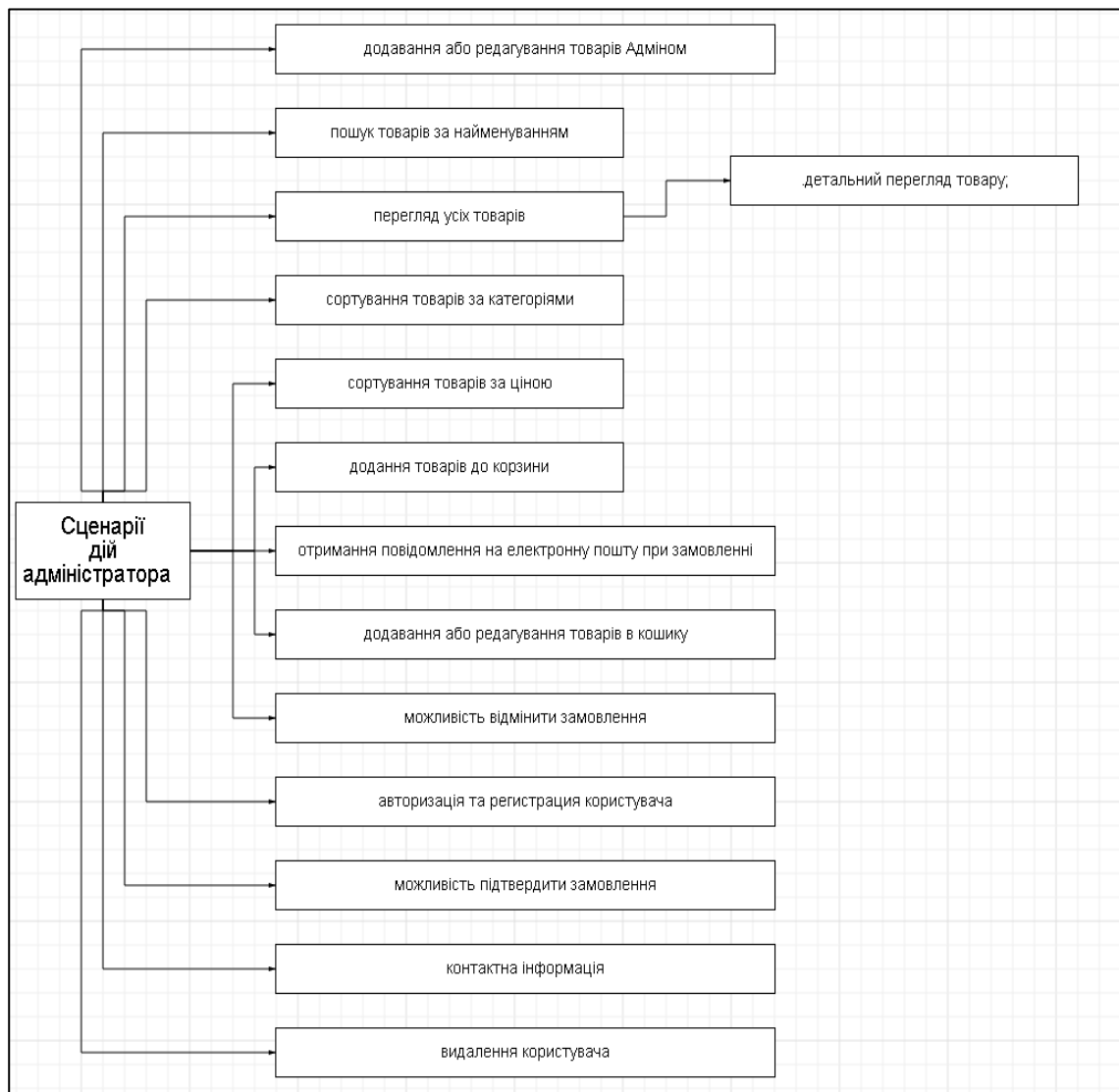


Рисунок 2.3 – Діаграма сценаріїв дій адміністратора

2.3 Розробка інтерфейсу сайту інтернет-магазину зоотоварів

Для розробки макету вебсайту було використано програму Figma. Програма Figma є популярним інструментом для дизайну та прототипування вебсайтів. Її використання має кілька переваг:

- хмарне зберігання та спільна робота: Figma дозволяє зберігати макети в хмарі, що дозволяє легко ділитися проектом з іншими учасниками;
- зручність використання: Figma має інтуїтивно зрозумілий інтерфейс, що спрощує роботу з ним;
- адаптивність та прототипування: Figma дозволяє створювати адаптивні макети, що відображаються на різних пристроях та екранах;
- інтеграція з іншими інструментами: Figma підтримує інтеграцію з іншими програмами та інструментами.

Загалом використання Figma для розробки макету вебсайту є популярним вибором серед дизайнерів та команд розробників завдяки його зручності, спільній роботі та функціональності.

Головним принципом при створенні вебсайту була зосередженість на інтуїтивно зрозумілому інтерфейсі з красивим оформленням та адаптивним дизайном. Візуальна привабливість та простота взаємодії з користувачем мають велике значення для створення позитивного враження і забезпечення зручності використання сайту. Крім того, адаптивний дизайн забезпечує оптимальний перегляд і функціональність сайту на різних пристроях і розмірах екранів, що є важливим у світі мобільних технологій. Комбінація цих факторів створює зручне та привабливе середовище для користувачів сайту, результат розробки головної сторінки представлений на рис. 2.4. – 2.5.

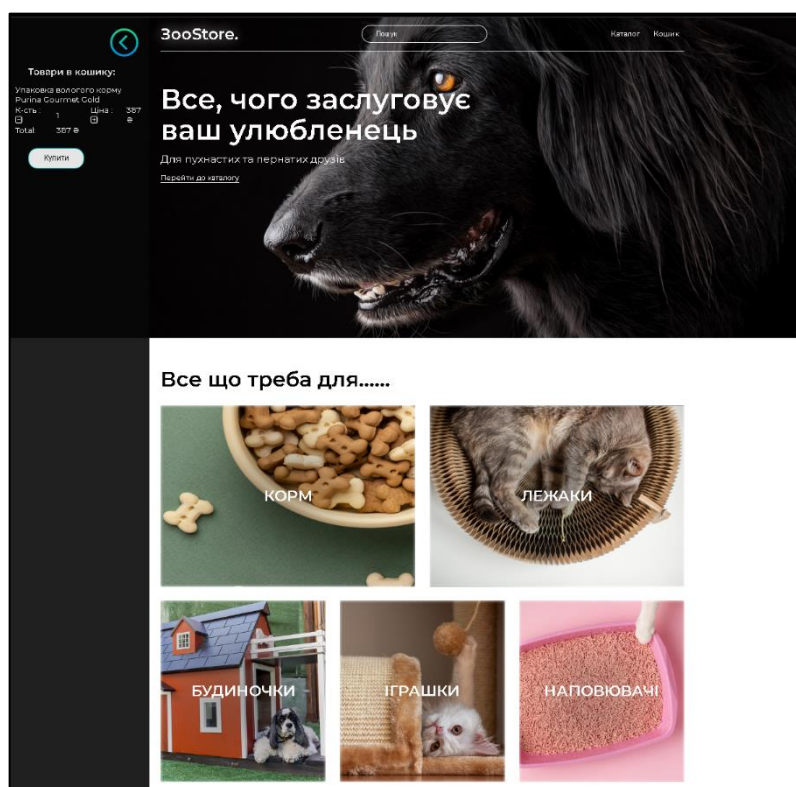


Рисунок 2.4 – Верхня частина головної сторінки

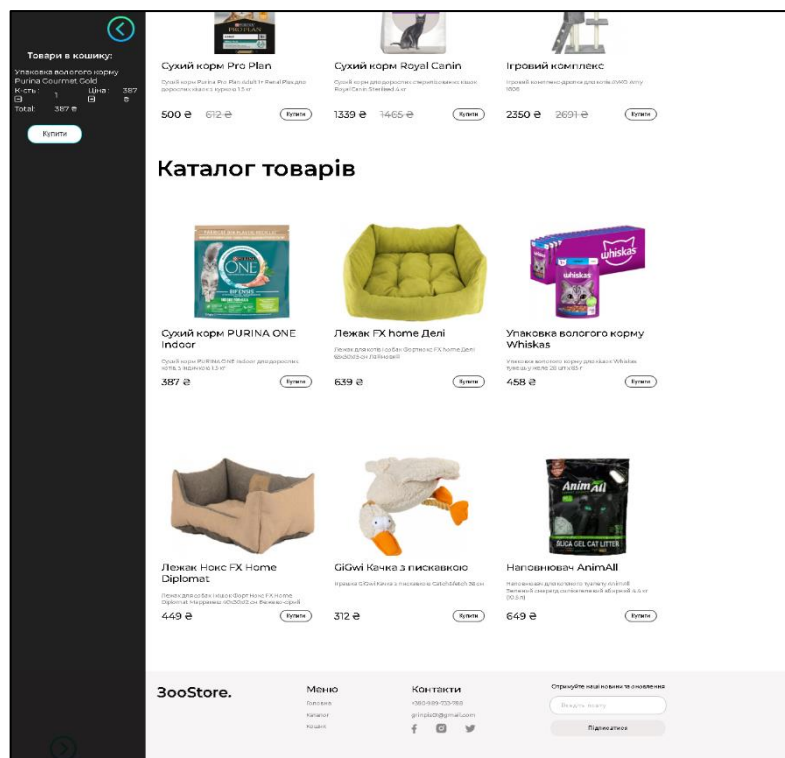


Рисунок 2.5 – Нижня частина головної сторінки

Сторінка каталогу є важливою частиною вебсайту, де представлений перелік товарів або послуг. Основна мета сторінки каталогу - надати користувачеві зручний спосіб ознайомитися з доступними продуктами або послугами та знайти потрібний товар чи інформацію, результат розробки сторінки каталогу представлений на рис. 2.6.

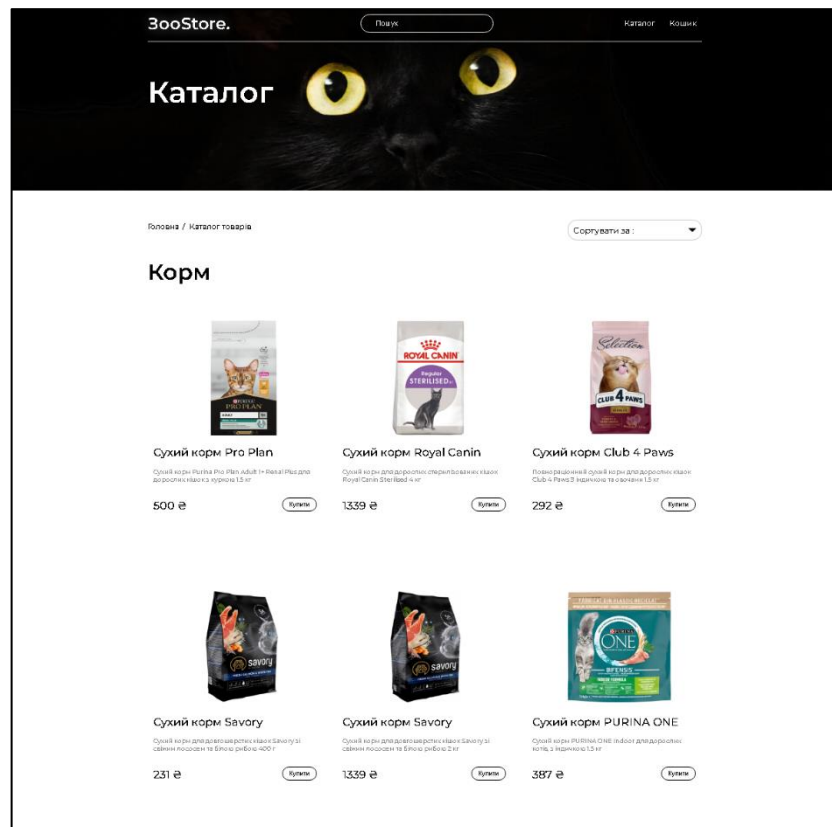
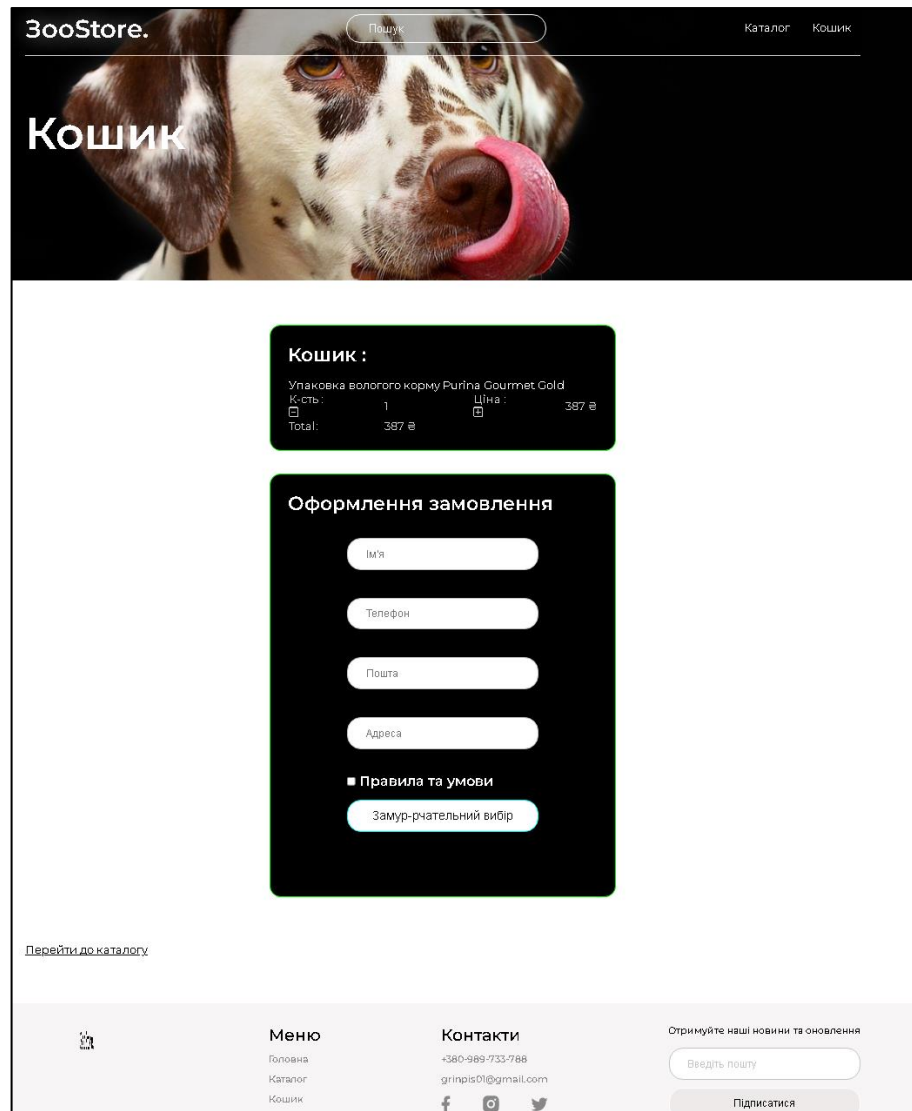


Рисунок 2.6 – Сторінка каталогу

Сторінка "Кошик" є важливою частиною магазину, де користувач може переглянути та керувати товарами, які він вибрав для покупки. Основна мета сторінки "Кошик" - забезпечити зручний спосіб оформлення замовлення та контролювання обраного вмісту кошика, результат розробки сторінки каталогу представлений на рис. 2.7.

В процесі розробки програми також було використано SCSS (Sassy CSS). SCSS є препроцесором CSS, що дозволяє розширити можливості звичайного CSS шляхом додавання змінних, вкладених стилів та інших функцій. SCSS надає більшу гнучкість та організованість у розробці стилів.

Весь код Sass / SCSS компілюється до стандартного CSS, щоб браузер насправді міг зрозуміти та відобразити результати [2].



The screenshot shows the 'ЗooStore.' website. At the top, there is a search bar with the text 'Пошук' and links for 'Каталог' and 'Кошик'. Below the header is a large image of a dog's face with the word 'Кошик' (Cart) overlaid. The main content area features a shopping cart summary and a checkout form.

Кошик :

Упаковка вологого корму Purina Gourmet Gold	
К-сть :	Ціна :
1	387 ₴
Total:	387 ₴

Оформлення замовлення

Ім'я

Телефон

Пошта

Адреса

☐ Правила та умови

[Перейти до каталогу](#)

Меню

- Головна
- Каталог
- Кошик

Контакти

+380-989-733-788
grinpis01@gmail.com

[f](#) [o](#) [w](#)

Отримуйте наші новини та оновлення

Рисунок 2.7 – Сторінка кошику

2.4 Проєктування бази даних

Для створення інтернет магазину потрібно вибрати БД де будуть зберігатися товари та інформація про користувачів, замовлення. Вибір баз даних був з дуже популярних і зручних рішень таких як вибір бази даних MySQL і PostgreSQL порівняння показників наведенні у табл.2.2.

Таблиця 2.2 – Вибір бази даних

Критерії порівняння	MySQL	PostgreSQL
Відкрите джерело	Відкритий вихідний код, але що належить Oracle	Відкрите джерело
Підтримка NoSQL/JSON	Підтримка деяких функцій	Тільки даних JSON
Декларативне секціонування	Підтримуються	Підтримуються
Логічна реплікація	Підтримуються	Підтримуються
Віконні функції	Підтримуються	Підтримуються
Вкладені вібірки	Так	Так
Транзакції	Так	Так
Тригери	Так	Так
Можливість зберігання лише у пам'яті	Так	Ні
Робота з простими запитам та невеликими обсягами даних	Дуже швидко	Швидко
Робота з складними запитам та складними операціями з даними.	Швидко	Дуже швидко

На основі наведених показників MySQL, можна зробити висновок, що MySQL є дуже добрим вибором для багатьох вебсайтів. З його простотою використання, великою спільнотою, високою продуктивністю та розширеними можливостями MySQL забезпечує зручну розробку та

ефективну роботу з базою даних. MySQL, найпопулярніша система керування базами даних SQL з відкритим кодом, розроблена, розповсюджена та підтримується Sun Microsystems, Inc [3]. На основі всіх факторів було обрано MySQL.

На основі дослідження функціональних вимог до програмного забезпечення, проведеного у розділі 2.2, було складено перелік основних вимог, що стосуються предметної області, і створено відповідні таблиці бази даних а також взаємодії в них. Взаємодія в базі даних налаштована наступним чином, товари зберігаються в таблиці goods та коли користувач вибрав товар та заповнив всі поля замовлення він надсилає дані у таблиці user_info та shop_order. Таблиця user_info містить інформацію о користувачі а також його ID. В таблицю shop_order входить id товару, ціна товару та присвоюється id користувача що представлено на рис. 2.8.

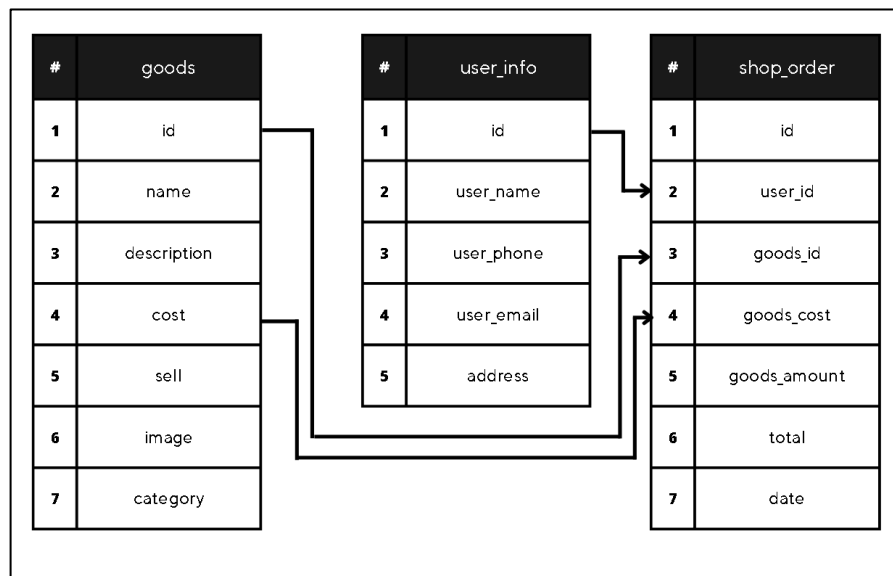
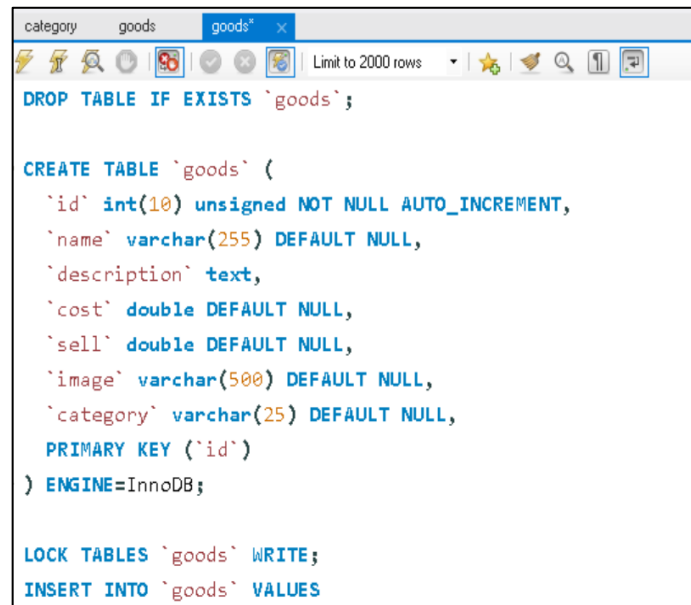


Рисунок 2.8 – Схема баз даних

Створення таблиці товару представлено на рис. 2.9:

- goods – назва БД;
- goods_id – унікальний ідентифікатор;
- goods_name – ціна товару;
- goods_description – категорія товару;

- goods_cost – ціна товару;
- goods_sell – знижка;
- goods_category – опис товару;
- goods_images – зображення товару;
- primary_key('id') – первинний ключ id.



```

DROP TABLE IF EXISTS `goods`;

CREATE TABLE `goods` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(255) DEFAULT NULL,
  `description` text,
  `cost` double DEFAULT NULL,
  `sell` double DEFAULT NULL,
  `image` varchar(500) DEFAULT NULL,
  `category` varchar(25) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;

LOCK TABLES `goods` WRITE;
INSERT INTO `goods` VALUES

```

Рисунок 2.9 – Таблиця товарів

Створення таблиці інформації про замовлення представлено на рис.

2.10:

- shop_order – назва БД;
- shop_order_id – унікальний ідентифікатор;
- shop_order _user_id – ідентифікатор користувача;
- shop_order _goods_id – ідентифікатор товару;
- shop_order _ goods_cost – ціна товару;
- shop_order _ goods_amount – кількість товару;
- shop_order _total – загальна ціна усього товару;
- shop_order _date – дата замовлення товару.

```

1 • use market;
2 • DROP TABLE IF EXISTS `shop_order`;
3
4 • CREATE TABLE `shop_order` (
5     `id` int(11) NOT NULL,
6     `user_id` int(11) NOT NULL,
7     `goods_id` int(11) NOT NULL,
8     `goods_cost` double NOT NULL,
9     `goods_amount` int(11) NOT NULL,
10    `total` double NOT NULL,
11    `date` int(11) NOT NULL
12 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

Рисунок 2.10 – Таблиця замовлень

Створення таблиці інформація про користувача представлено на рис.

2.11:

- user_info – назва БД;
- user_info _ id – ідентифікатор користувача;
- user_info _user_name – ім'я замовника;
- user_info _ user _phone – телефон;
- user_info _ user _email – пошта;
- user_info _adress – адреса доставки.

```

1 • use market;
2 • DROP TABLE IF EXISTS `user_info`;
3
4 • CREATE TABLE `user_info` (
5     `id` int(11) NOT NULL,
6     `user_name` varchar(100) NOT NULL,
7     `user_phone` varchar(100) NOT NULL,
8     `user_email` varchar(100) NOT NULL,
9     `address` varchar(1000) NOT NULL
10 );

```

Рисунок 2.11 – Таблиця інформація про користувача

2.5 Визначення структури програми

Структура програми побудована на основі архітектурного стилю REST API, що передбачає використання вебфреймворку Express. Застосування REST API дозволяє забезпечувати стандартизовану комунікацію між клієнтом і сервером за допомогою HTTP-протоколу. Клієнти можуть звертатися до сервера за різними ресурсами, використовуючи HTTP-методи, такі як GET, POST, PUT, DELETE і т.д. Сервер відповідає на запити клієнтів, надаючи їм необхідні дані або виконуючи певні дії [4]-[5].

Застосування вебфреймворку Express допомагає спростити процес розробки REST API. Express забезпечує шаблони для маршрутизації запитів, обробки запитів і відправки відповідей клієнтам. Використання Express дозволяє розробникам швидко створювати потужні та гнучкі REST API з мінімальними зусиллями [6].

У структурі розробки корзини використовується технологія AJAX (Asynchronous JavaScript and XML). AJAX дозволяє здійснювати асинхронні запити до сервера без перезавантаження сторінки. Це дозволяє динамічно оновлювати вміст вебсторінки без необхідності повного перезавантаження [7].

За допомогою AJAX можна взаємодіяти з REST API сервера, надсилаючи асинхронні запити й отримуючи відповіді у форматі JSON або інших форматах. Наприклад, при натисканні кнопки на вебсторінці, можна викликати AJAX-запит до сервера, представлено на рис. 2.12, який буде оброблювати цей запит і повертати необхідні дані [8].

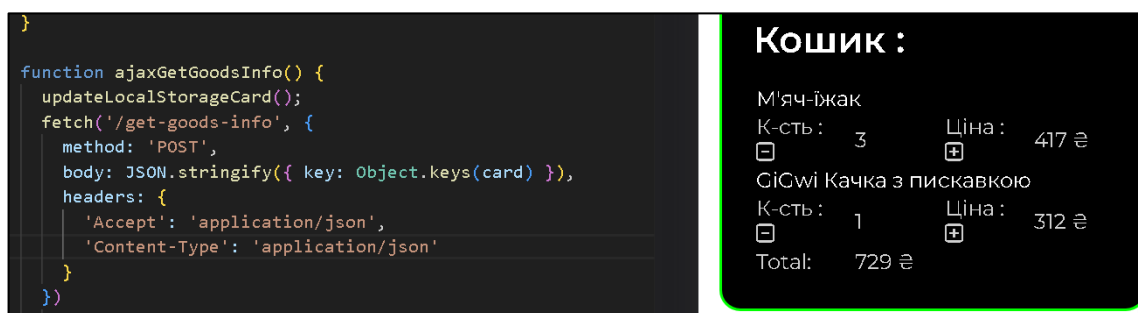


Рисунок 2.12 – AJAX-запит

У процесі розробки програми також було використано об'єкт Promise. Promise є механізмом у JavaScript, що дозволяє виконувати асинхронні операції та отримувати їх результати зручним способом [9].

Promise дозволяє виконувати операції, які займають багато часу, такі як запити до сервера представлено на рис. 2.13, завантаження файлів або обробка даних без блокування основного потоку виконання JavaScript. Вміст промісу може мати стан "очікування" (pending), "виконано" (fulfilled) або "відхилено" (rejected) [10].

Promise має два основні методи: then() та catch(). Метод then() викликається після успішного виконання промісу і отримує результат операції представлено на рис. 2.14. Метод catch() викликається у разі, якщо проміс відхилено, і обробляє помилку, якщо така виникла [11].

```
let goods = new Promise(function (resolve, reject) {
  con.query(
    'SELECT * FROM goods id',
    function (error, result) {
      if (error) reject(error);
      resolve(result);
    });
});

Promise.all([goods, nap, korm, game, homes, bed]).then(function (value) {
  res.render('index', {
    goods: JSON.parse(JSON.stringify(value[0])),
    nap: JSON.parse(JSON.stringify(value[1])),
    korm: JSON.parse(JSON.stringify(value[2])),
    game: JSON.parse(JSON.stringify(value[3])),
    homes: JSON.parse(JSON.stringify(value[4])),
    bed: JSON.parse(JSON.stringify(value[5]))
  });
});
```

Рисунок 2.13 – Запити до сервера

```

fetch('/get-goods-info', {
  method: 'POST',
  body: JSON.stringify({ key: Object.keys(card) }),
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json'
  }
})
.then(function (response) {
  return response.text();
})
.then(function (body) {
  console.log(body);
  showCard(JSON.parse(body));
})
}

```

Рисунок 2.14 – Promise методи

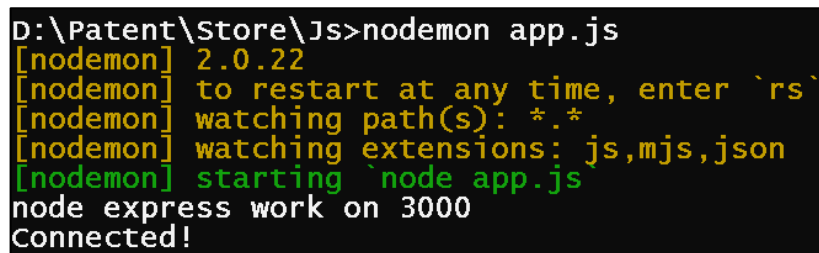
2.6 Висновки за розділом 2

Після порівняння фреймворків Express, Koa та Vue для розробки вебзастосунків на мові JavaScript, було прийнято рішення використати фреймворк Express. Також було прийнято рішення використовувати базу даних MySQL після порівняння з PostgreSQL. Обидві ці системи управління базами даних (СКБД) є популярними та мають свої особливості. Вибір MySQL був зроблений з огляду на його широку підтримку, швидкодію та велику спільноту розробників. Методологія REST була проаналізована та використовується для побудови вебзастосунку. Під час аналізу були виділені основні сценарії роботи користувачів, менеджера та адміністратора з вебзастосунком. Було побудовано таблиці бази даних та визначено структуру програми.

3 РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ

3.1 Алгоритм функціонування програми

Першим, що було зроблено встановлено nodemon – це інструмент, який допомагає розробляти програми на основі Node.js [12] представлено на рис. 3.1. Він стежить за змінами в файловій системі вашого проєкту і автоматично перезапускає сервер, щойно виявляється зміна в коді. Це зменшує необхідність вручну перезапускати сервер після кожної зміни коду і прискорює процес розробки. Одним із ключових переваг використання Nodemon є те, що він дозволяє зосередитися на написанні коду, не витрачаючи час на ручне перезапускання сервера після змін.



```
D:\Patent\Store\Js>nodemon app.js
[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
node express work on 3000
Connected!
```

Рисунок 3.1 – Nodemon

В ході розробки було зроблено низку необхідних підключень, шаблонізаторів, модулів представлено на рис. 3.2:

- підключення Express;
- посилання на папки де зберігаються статика;
- встановлення шаблонізатора pug;
- встановлення методів Express;
- підключення nodemailer;
- підключення та налаштування модулів mysql.

```

app.js > ...
1  let express = require('express');
2  let app = express();
3
4  app.use(express.static('public'));
5  app.use(express.static('views'));
6
7  app.set('view engine', 'pug');
8
9  let mysql = require('mysql');
10
11 app.use(express.json());
12 const nodemailer = require('nodemailer')
13
14 let con = mysql.createConnection({
15   host: 'localhost',
16   user: 'root',
17   password: 'Teamshotgotkill_1',
18   database: 'market'
19 });
20
21 con.connect(function(err) {
22   if (err) throw err;
23   console.log("Connected!");
24 });
25
26 app.listen(3000, function () {
27   console.log('node express work on 3000');
28 });
29

```

Рисунок 3.2 – Підключення

Структура проекту файлів визначає організацію та розміщення файлів у проекті. Вона визначає, де розміщувати різні типи файлів, такі як вихідний код, файли конфігурації, ресурси та інші компоненти проекту представлено на рис. 3.3. Це допомагає зберігати проект організованим, легким у використанні та розширенні. Перелік розроблених методів з їхніми призначеннями:

- `let express = require('express'), let app = express()` ds – підключення express;
- `app.use(express.static('#'))` – метод Node.js статичні файли;
- `app.use(express.json())` – метод який розбирає запити в об'єкт у форматі JSON. заснований на body-parser;
- `let con = mysql.createConnection` – метод підключення бази даних;
- `con.connect(function (err))` – метод перевірки підключення;
- `app.listen(3000, function (err))` – метод підключення на 3000 порт;
- `function getGoodsByCategory(category)` – метод функції повертає Promise який у свою чергу виводить дані з таблиці mysql;

- `app.get ('/catalog', function (req, res)` – метод запиту сторінки;
- `Promise.all([nap, korm, game, homes, bed])` – метод `Promise.all(iterable)` повертає `promise`, який виконається тоді, коли будуть виконані всі проміси;
- `res.render('catalog',{})` – метод використовується для рендерингу сторінок;
- `function saveOrder(data, result)` – метод `function` зберігає замовлення до бази даних;
- `async function sendMail(data, result)` – метод формування листа;
- `let info = await transporter.sendMail(mailOption)` – метод надсилання повідомлення про замовлення яке містить `id` повідомлення а також де можна подивитися це повідомлення за посланням;
- `function addToCard()` – метод додавання до кошику;
- `document.querySelectorAll('.add-to-card')` – метод при натисканні кнопки з класом `add-to-card` функція додає товар до кошика;
- `function ajaxGetGoodsInfo()` – метод відсилає запит на сервер і виводить інформацію it товарів що лежать у кошику;
- `function showCard()` – метод `showCard` формує дані у вигляді таблиці;
- `document.querySelector('#card-nav').innerHTML = out` – метод повертають усі знайдені статичні дані;
- `function cardPlus()` – метод додає один товар;
- `function cardMinus()` – метод віднімає один товар;
- `function updateLocalStorageCard()` – метод збереження кошика;
- `function showNav()` – метод розгортає кошик;
- `function closeNav()` – метод згортає кошик;
- `document.querySelector('.close-nav').onclick = closeNav` – метод при якому після натискання на кнопку з класом виконується функція закриття кошика;

- `document.querySelector('.show-nav').onclick = showNav` – метод при якому після натискання на кнопку з класом виконується функція відкриття кошика;
- `document.querySelector('#Shop-order').onsubmit = function (event)` – метод перевіряє та надсилає лист до БД.

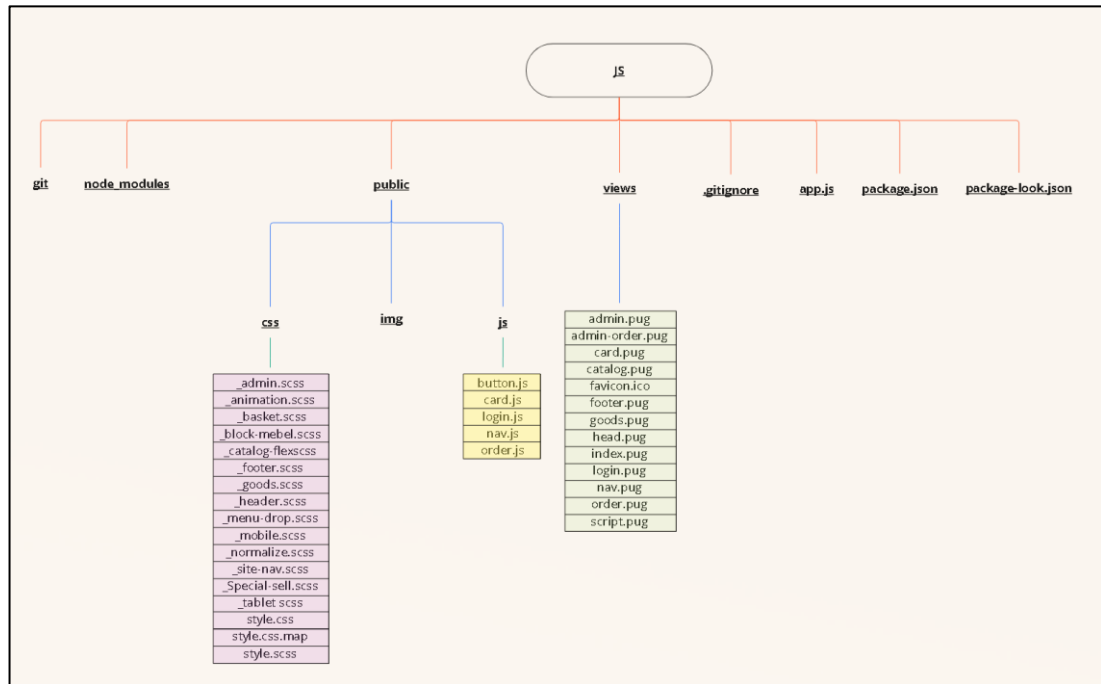


Рисунок 3.3 – Структура проекту

Вивід бази даних товарів з `mysql` я скористався запитом `SELECT`. Після виконання цього запиту, я отримав результат, який міститиме всю необхідну інформацію з БД що було спроектовано у розділі 2. Далі створив змінну і перебрав весь масив інформації і ідентифікував його. Виведення запиту до бази даних я привласнив новій змінній `Promise` представлено на рис. 3.4.

```

let game = new Promise(function (resolve, reject) {
  con.query("SELECT * FROM goods WHERE category = 'Іграшки'",
    function (error, result) {
      if (error) reject(error);
      resolve(result);

      let user = {};
      for (let i = 0; i < result.length; i++) {
        user[result[i]['id']] = result[i];
      }
      console.log(JSON.parse(JSON.stringify(user)))
    });
});

```

Рисунок 3.4 – Promise

Створення основної сторінки за допомогою препроцесора Pug. Pug - це препроцесор HTML, який дозволяє писати більш компактний і зрозумілий код HTML. Pug використовується для генерації HTML-сторінок на основі вхідних шаблонів представлено на рис. 3.5.

```

index.pug
doctype html
head
  meta(charset='utf-8')
  meta(http-equiv='X-UA-Compatible' content='IE=edge')
  meta(name='viewport' content='width=device-width, initial-scale=1.0')
  link(rel='preconnect' href='https://fonts.googleapis.com')
  link(rel='preconnect' href='https://fonts.gstatic.com' crossorigin='')
  link(href='https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600&display=swap' rel='stylesheet')
  link(rel='preconnect' href='https://fonts.googleapis.com')
  link(rel='preconnect' href='https://fonts.gstatic.com' crossorigin='')
  link(href='https://fonts.googleapis.com/css2?family=Poppins:wght@600&display=swap' rel='stylesheet')
  link(rel='stylesheet' href='css/style.css')
  link(rel='icon' href='favicon.ico' type='images/x-icon')
  title BookStore.

.top.center
  // Head-top-menu
  header
    .header
      a(href='index')
        p.header__logo-text BookStore.
      // <a href="index"></a>
      input.header__input(type='text' placeholder='Пошук')
      nav
        a.header__link(href='catalog') Каталог
        a.header__link(href='order') Кошик
    .phone
      .phone_menu
        a(href='index')
          p.header__logo-text BookStore.
      .phone_box
        a(href='#')
          img.phone__img(src='../img/f.png' alt='')
        a(href='catalog')
          img.phone__img(src='../img/fff.png' alt='')
        a(href='order')
          img.phone__img(src='../img/fff.png' alt='')
  // Head-top-menu
  // Head-under-text
  section
    .Under-text
      h1.Under-text_H1 Всі, чого заслуговує ваш улюбленець
      p.Under-text_p Для пухнастих та пернатих друзів
      a.Under-text_a.Under-text_catalog(href='catalog') Перейти до каталогу

```

Рисунок 3.5 – Індикація Pug

В препроцесорі Pug існує директива `each`, яка дозволяє виконувати цикл `for` для повторення блоку коду залежно від елементів масиву або об'єкту. Для перебору властивостей об'єкту використовується синтаксис `each value, key in object`. В цьому прикладі, `user` є об'єктом, а `key` та `value` представляють пари ключ-значення для кожної властивості об'єкту представлено на рис. 3.6.

```
.center.Mebel__style-h2
| .product-catalog.center
|   each val, index in homes
```

Рисунок 3.6 – Директива `each`

Для директиви `each` підключений `card.pug` представлено на рис. 3.7. в якому знаходиться реалізація відображення товару представлено на рис. 3.8.

```
.center.Mebel__style-h2
| .product-catalog.center
|   each val, index in game
|     include card.pug
```

Рисунок 3.7 – Підключення `card.pug`

```
JS app.js index.pug card.pug
views > card.pug
1 .center
2   .products
3     .product__sell
4     .product-catalog
5       .products__box
6         a(href="goods?id="+val['id'])
7           .products__box-img
8             img(
9               src='../img/'+val['image']
10            )
11         .h2-style
12           p #{val["name"]}
13         .text-p2
14           p #{val["description"]}
15         .price_cost
16           .price_cost-main
17             p(class="text-center") #{val['cost']} &#x20B4
18         .price_button
19           button(
20             data-goods_id = val['id']
21             class="button button-primary add-to-card"
22           ) Купити
23
24
25
26
27
28
```

Рисунок 3.8 – Код реалізації відображення товару

Формування сторінки одиночного товару реалізована в goods.pug. При натисканні на поля товару яке має тип посилання представлено на рис. 3.9, присвоюються значення id товару в значення `val['id']` та відкривається індивідуальна сторінка товару представлено на рис. 3.10.

```
a(href="goods?id="+val['id'])
```

Рисунок 3.9 – Присвоєння Id товару

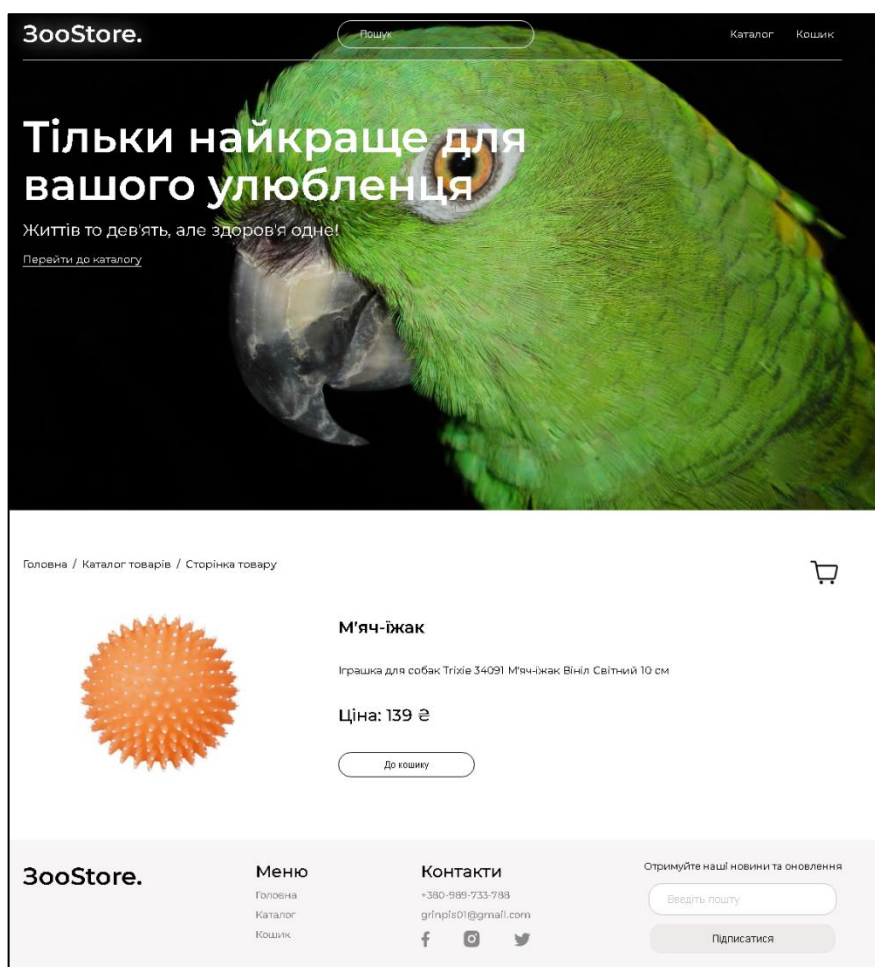


Рисунок 3.10 – Індивідуальна сторінка товару

За Ідентифікатором товару сторінка автоматично підтягує потрібні дані з БД, виведення даних проводиться через Promise представлено на рис. 3.11.

```

.center
.goods-box
  .goods-box__img
    img(
      src="img/"+goods[0]['image']
      class="goods-big-image"
    )
  .goods-box__info
    h2 #{goods[0]['name']}
    p #{goods[0]['description']}
  .price__button
    .price__cost
      .price__cost-main
        span(class="success") Ціна: #{goods[0]['cost']} ₾
  .goods-box__button
    button(
      class="button-success add-to-cart"
      data-goods_id=goods[0]['id']
    ) До кошику

```

Рисунок 3.11 – Індивідуальна сторінка товару

3.2 Алгоритм роботи кошика

Створено кошик для товарів з функціями додавання та видалення а також переходу до головного кошика представлено на рис. 3.12.

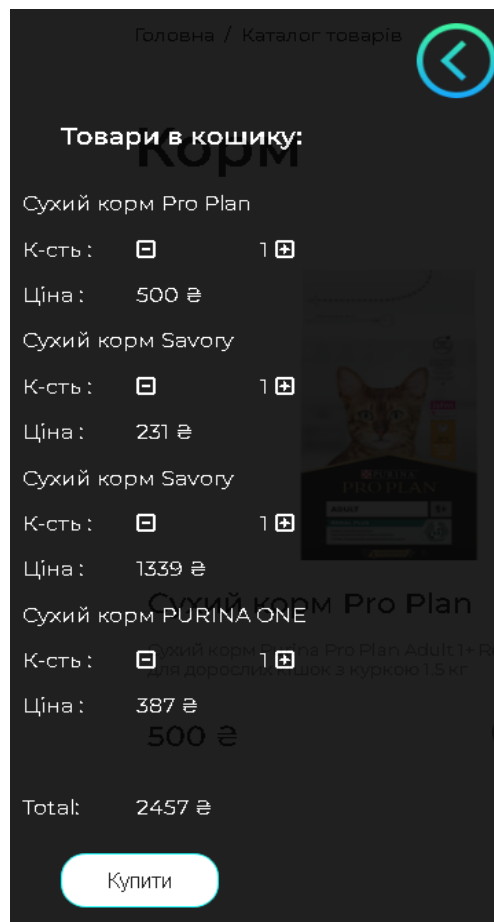


Рисунок 3.12 – Індивідуальна сторінка товару

Додавання в кошик реалізується через метод `querySelectorAll` клас `.add-to-card`, який обробляє функцію елемента `onclick` при натисканні на кнопку спрацьовує додавання в кошик представлено на рис. 3.13-3.14.

```
document.querySelectorAll('.add-to-card').forEach(function(element){
    element.onclick = addToCard;
});
```

Рисунок 3.13 – Document querySelector

```
function addToCard() {
    let goodsId = this.dataset.goods_id;
    if (card[goodsId]) {
        card[goodsId]++;
    }
    else {
        card[goodsId] = 1;
    }
    console.log(card);
    ajaxGetGoodsInfo();
}
```

Рисунок 3.14 – Функція додавання в кошик

Функція `ajax` відсилає запит на сервер і виводить інформацію it товарів що лежать у кошику. Відправляється запит на сервер методом `post`, `body` допомагає упакувати дані всередину `post` запиту представлено на рис. 3.15.

```
function ajaxGetGoodsInfo() {
    updateLocalStorageCard();
    fetch('/get-goods-info', {
        method: 'POST',
        body: JSON.stringify({ key: Object.keys(card) }),
        headers: {
            'Accept': 'application/json',
            'Content-Type': 'application/json'
        }
    })
    .then(function (response) {
        return response.text();
    })
    .then(function (body) {
        console.log(body);
        showCard(JSON.parse(body));
    })
}
```

Рисунок 3.15 – Функція `ajax`

Функція showCard формує дані у вигляді таблиці та присвоюється до id #card-nav представлено на рис. 3.16., а також є додавання та видалення товарів у кошику через метод querySelectorAll представлено на рис. 3.17.

```
function showCard(data) {
  let out = '<table class="table table-striped table-card"><tbody>';
  let total = 0;
  for (let key in card){
    out += '<tr><td colspan="4" class="movion"><a class="movion" href="/goods?id=${key}">${data[key]['name']}</a></tr>';
    out += '<tr><td><p class="movions">К-сть :</p></td><td class="movion"><i class="far fa-minus-square cart-minus" data-goods_id="${key}"></i></td>';
    out += '<td class="movion">${card[key]}</td>';
    out += '<td class="movion"><class="movion"><i class="far fa-plus-square cart-plus" data-goods_id="${key}"></i></td>';
    out += '</tr>';
    out += '<tr class="movion"><td><p class="movions">Ціна : </p></td><td class="movion">${formatPrice(data[key]['cost'] * card[key])} €</td>';
    out += '</tr>';
    total += card[key] * data[key]['cost'];
  }
  out += '<tr><td colspan="4" class="movion-total">Total: </td><td class="movion-total">${formatPrice(total)} €</td></tr>';
  out += '</tbody></table>';
  document.querySelector('#card-nav').innerHTML = out;
  document.querySelectorAll('.cart-minus').forEach(function (element) {
    element.onclick = cardMinus;
  });
  document.querySelectorAll('.cart-plus').forEach(function (element) {
    element.onclick = cardPlus;
  });
}
```

Рисунок 3.16 – Функція showCard

```
function cardPlus() {
  let goodsId = this.dataset.goods_id;
  card[goodsId]++;
  ajaxGetGoodsInfo();
}

function cardMinus() {
  let goodsId = this.dataset.goods_id;
  if (card[goodsId] - 1 > 0) {
    card[goodsId]--;
  }
  else {
    delete (card[goodsId]);
  }
  ajaxGetGoodsInfo();
}
```

Рисунок 3.17 – Додавання та видалення товарів

Функція збереження стану кошика в представлено на рис. 3.18.

```
function updateLocalStorageCard() {
  localStorage.setItem('card', JSON.stringify(card));
}
```

Рисунок 3.18 – Функція збереження кошика

Перевірка чи є щось у кошику представлено на рис. 3.19.

```
if (localStorage.getItem('card')) {
  card = JSON.parse(localStorage.getItem('card'));
  ajaxGetGoodsInfo();
}
```

Рисунок 3.19 – Перевірка на порожність кошика

Формування запиту /get-goods-info до бази даних mysql для отримання інформації о товарі представлено на рис. 3.20.

```
app.post('/get-goods-info', function (req, res) {
  console.log(req.body.key);
  if (req.body.key.length != 0) {
    con.query('SELECT id,name,cost FROM goods WHERE id IN (' + req.body.key.join(',') + ')', function (error, result, fields) {
      if (error) throw error;
      console.log(result);
      let goods = {};
      for (let i = 0; i < result.length; i++) {
        goods[result[i]['id']] = result[i];
      }
      res.json(goods);
    });
  }
  else {
    res.send('0');
  }
});
```

Рисунок 3.20 – Запит get-goods-info

Сторінка кошика є основним виведенням всіх товарів, вона має кошик з усіма товарами а також форму даних для здійснення покупки представлено на рис. 3.21.

Рисунок 3.21 – Сторінка кошика

Структура кошика розроблена у pug має клас card-nav завдяки якому підтягується усі вибрані товари, а також поля введення інформації представлено на рис. 3.22.

```

order.pug
.cart_display-interface
  h2(class="cart-headers") Кошик :
  div(class="category-list")
    div(id="card-nav")

.cart_display-interface.cart_input_date.center
  h2.cart_head Оформлення замовлення
  .cart_head-move

  form#lite-shop-order(action="")
    div(class="form-control")
      label(for="username")
      input(
        type="text",
        id="username",
        placeholder="Ім'я",
      )
    div(class="form-control")
      label(for="phone")
      input(
        type="text",
        id="phone",
        placeholder="Телефон",
      )
    div(class="form-control")
      label(for="email")
      input(
        type="text",
        id="email",
        placeholder="Пошта",
      )
    div(class="form-control form-controls")
      label(for="address")
      input(
        type="text",
        id="address",
        placeholder="Адреса",
      )
    div(class="form-control")
      label(for="rule")
      input(
        type="checkbox",
        id="rule",
        value="1",
      )
      span Правила та умови
    div(class="form-control text-rights")
      button(class="button-primary", type="submit") Замур-рчателний вибір

```

Рисунок 3.22 – Структура кошика

Nodemailer — це модуль для програм Node.js, що дозволяє легко надсилати електронні листи [13]. Купуючи товар необхідно заповнити поля даних, перевірку та відправлення та відправлення замовлення було реалізовано через nodemailer представлено на рис. 3.23-3.25.

```

document.querySelector('#lite-shop-order').onsubmit = function (event) {
  event.preventDefault();
  let username = document.querySelector('#username').value.trim();
  let phone = document.querySelector('#phone').value.trim();
  let email = document.querySelector('#email').value.trim();
  let address = document.querySelector('#address').value.trim();

  if (!document.querySelector('#rule').checked) {
    //с правилами не согласен
    Swal.fire({
      title: 'Помилка !',
      text: 'Прочитайте та прийміть правила',
      type: 'info',
      confirmButtonText: 'OK'
    });
    return false;
  }

  if (username == '' || phone == '' || email == '' || address == '') {
    //не заповнені поля
    Swal.fire({
      title: 'Помилка !',
      text: 'Заповніть всі поля',
      type: 'info',
      confirmButtonText: 'OK'
    });
    return false;
  }
}

```

Рисунок 3.23 – Перевірка даних

```

fetch('/finish-order', {
  method: 'POST',
  body: JSON.stringify({
    'username': username,
    'phone': phone,
    'address': address,
    'email': email,
    'key': JSON.parse(localStorage.getItem('card'))
  }),
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json'
  }
})
.then(function (response) {
  return response.text();
})
.then(function (body) {
  if (body == 1) {
    Swal.fire({
      title: 'Готово',
      text: 'OK',
      type: 'info',
      confirmButtonText: 'OK'
    });
  } else {
    Swal.fire({
      title: 'Помилка відправки',
      text: 'Помилка !',
      type: 'помилка',
      confirmButtonText: 'OK'
    });
  }
})
}

```

Рисунок 3.24 – Формування запиту

```

app.post('/get-goods-info', function (req, res) {
  console.log(req.body.key);
  if (req.body.key.length != 0) {
    con.query('SELECT id,name,cost FROM goods WHERE id IN (' + req.body.key.join(',') + ')', function (error, result, fields) {
      if (error) throw error;
      console.log(result);
      let goods = {};
      for (let i = 0; i < result.length; i++) {
        goods[result[i]['id']] = result[i];
      }
      res.json(goods);
    });
  } else {
    res.send('0');
  }
});

```

Рисунок 3.25 – Відправка запиту

Результатом роботи nodemailer є тестовий виведення даних на тестову пошту представлено на рис. 3.26.

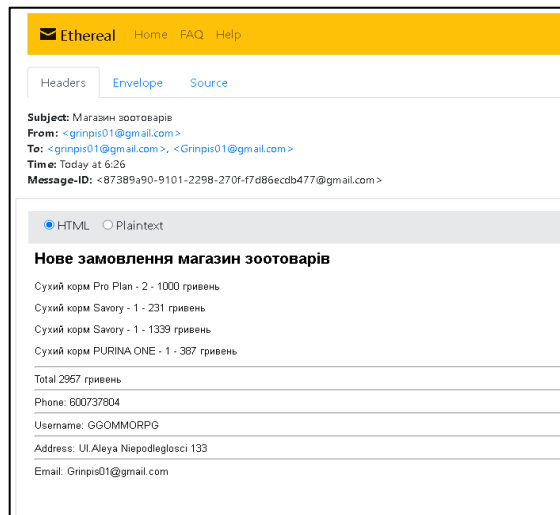


Рисунок 3.26 – Отриманий лист

Реалізації збереження замовлень у базі даних реалізована таким чином: функція `saveOrder` приймає значення `data` (інформація про користувача), `result` (відомості про товар). Вона викликається у `post` запиті `/finish-order` представлено на рис. 3.27.

```
app.post('/finish-order', function (req, res) {
  console.log(req.body);
  if (req.body.key.length != 0) {
    let key = Object.keys(req.body.key);
    con.query(
      "SELECT id,name,cost FROM goods WHERE id IN ('" + key.join(',') + "')",
      function (error, result, fields) {
        if (error) throw error;
        console.log(result);
        sendmail(req.body, result, catch(console.error));
        saveOrder(req.body, result);
        res.send('1');
      }
    );
  } else {
    res.send('0');
  }
});

function saveOrder(data, result) {
  let sql;
  sql = "INSERT INTO user_info (user_name, user_phone, user_email, address) VALUES ('" + data.username + "', '" + data.phone + "', '" + data.email + "', '" + data.address + "')";
  con.query(sql, function (error, resultQuery) {
    if (error) throw error;
    console.log('user save');
    console.log(resultQuery);
    let userId = resultQuery.insertId;
    date = new Date() / 1000;
    for (let i = 0; i < result.length; i++) {
      sql = "INSERT INTO shop_order (date, user_id, goods_id, goods_cost, goods_amount, total) VALUES ('" + date + "', '" + userId + "', '" + result[i]['id'] + "', '" + result[i]['cost'] + "', '" + data.key[result[i]['id']] + "')";
      con.query(sql, function (error, resultQuery) {
        if (error) throw error;
        console.log('1 saved');
      });
    }
  });
}
```

Рисунок 3.27 – Функція `saveOrder`

Після того, як користувач вибрав товари і заповнив всі форми інформації дані відправляються на сервера `mysql`, перший, який зберігає інформацію про користувача, а другий, який зберігає перелік замовлень представлено на рис. 3.28 - 3.30.

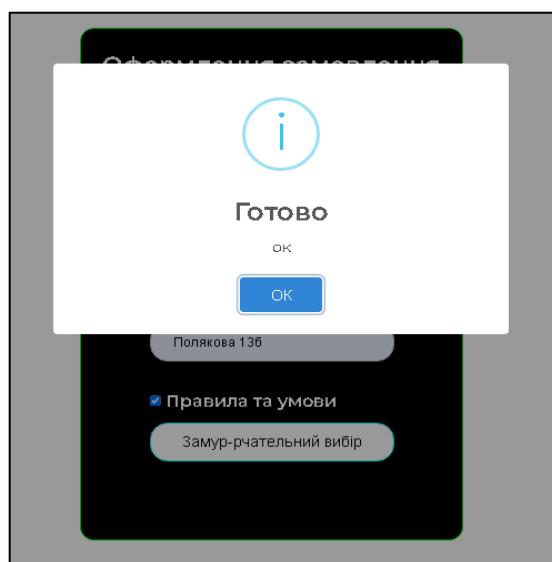


Рисунок 3.28 – Відправка замовлення

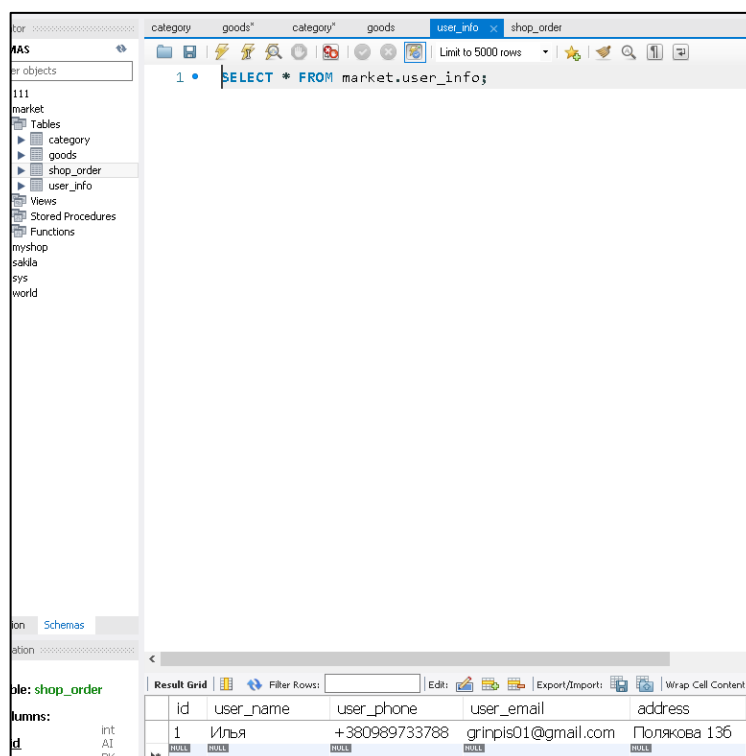


Рисунок 3.29 – Збережена інформація про користувача

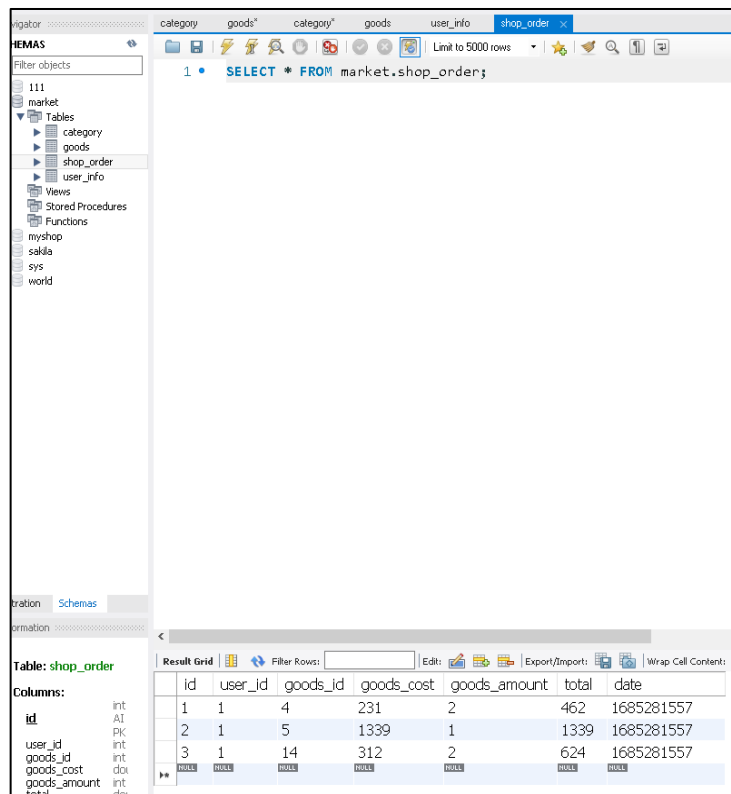


Рисунок 3.30 – Збережена інформація про замовлення

phpMyAdmin дозволяє адміністраторам зручно та ефективно керувати різними аспектами сайту без необхідності втручатися в код або базу даних напямучу [14]. Вона забезпечує безпеку, контроль та зручність управління сайтом з боку адміністраторів. Розроблена таблиця PHP для адмінів та менеджерів представлено на рис. 3.31. PHP - це мова програмування загального призначення, яка використовується для розробки вебдодатків і динамічних вебсайтів. розроблена база даних для адмін панелі [15].

Структура таблиці Вид відносин

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(5)			Hi	Немає		AUTO_INCREMENT	      
2	login	varchar(100)	utf8_general_ci		Hi	Немає			      
3	password	varchar(100)	utf8_general_ci		Hi	Немає			      
4	hash	varchar(100)	utf8_general_ci		Hi	Немає			      

Перевірити все Вибрані:       

Рисунок 3.31 – Таблиця PHP

Запит на виведення адмін панелі представлено на рис. 3.32.

```
app.get('/admin-order', function (req, res) {
  con.query('SELECT
    shop_order.id as id,
    shop_order.user_id as user_id,
    shop_order.goods_id as goods_id,
    shop_order.goods_cost as goods_cost,
    shop_order.goods_amount as goods_amount,
    shop_order.total as total,
    from_unixtime(date,"%Y-%m-%d %h:%m") as human_date,
    user_info.user_name as user,
    user_info.user_phone as phone,
    user_info.address as address
  FROM
    shop_order
  LEFT JOIN
    user_info
  ON shop_order.user_id = user_info.id ORDER BY id DESC', function (error, result, fields) {
    if (error) throw error;
    console.log(result);
    res.render('admin-order', { order: JSON.parse(JSON.stringify(result)) });
  });
});

/**
 * login form =====
 */
app.get('/login', function (req, res) {...
```

Рисунок 3.32 – Admin-order

Проміжна перевірка для перевірки доступу або приховування представлено на рис. 3.33.

```
JS adminjs > exports > con.query() callback
1 module.exports = function (req, res, con, callback) {
2   console.log(req.cookies);
3   console.log(req.cookies.hash);
4   console.log(req.cookies.id);
5   if (req.cookies.hash == undefined || req.cookies.id == undefined) {
6     res.redirect('/login');
7     return false;
8   }
9   con.query(
10    'SELECT * FROM user WHERE id=' + req.cookies.id + ' and hash="' + req.cookies.hash + "'",
11    function (error, result) {
12      if (error) reject(error);
13      console.log(result);
14      if (result.length == 0) {
15        console.log('error user not found');
16        res.redirect('/login');
17      }
18      else {
19        next();
20      }
21    });
22 }
```

Рисунок 3.33 – Функція перевірки доступу

3.3 Висновки за розділом 3

Відповідно до вимог і архітектури, був реалізований застосунок, який має такі основні частини: сторінки, роботу з товарами і логіку корзини, а також адмінпанель. Кожна частина була ініціалізована згідно з вимогами.

В додатку використовується MySQL база даних. Архітектура бази даних була розроблена з урахуванням потреб додатку. Кожен модуль кожної частини був реалізований з урахуванням функціоналу та деталей, що були описані.

4 ЕКСПЛУАТАЦІЯ І ТЕСТУВАННЯ ВЕБЗАСТОСУНКУ

4.1 Опис програми

Був спроектований та реалізований застосунок, який надає користувачам можливість переглядати та купувати товари для тварин. Цей спеціалізований застосунок для продажу зоотоварів пропонує численні переваги як для покупців, так і для продавців.

Для покупців, вебзастосунок забезпечує зручний спосіб перегляду широкого асортименту товарів для тварин, таких як корм, аксесуари, іграшки та інше. Користувачі можуть швидко шукати та фільтрувати товари за категоріями, цінами і іншими параметрами.

Для продавців, вебзастосунок створює можливість просто та ефективно представити свої товари для тварин онлайн. Застосунок також може надавати функції звітності, які допомагають продавцям відстежувати продажі, аналізувати популярність товарів та приймати відповідні рішення для покращення бізнесу.

Загалом, цей вебзастосунок створює зручну та ефективну платформу для здійснення покупок зоотоварів, сприяючи як покупцям, так і продавцям у досягненні своїх цілей.

4.2 Експлуатація програми

Для використання програми вам потрібно перейти за URL-адресою, яка відкриє вебсторінку, на якій користувачі можуть взаємодіяти з програмою представлено на рис. 4.1.

Ця вебсторінка була спроектована з урахуванням зручного та привабливого дизайну, зручної навігації та відповідності вимогам користувачів. Вона може містити різні елементи інтерфейсу, такі як меню, кнопки, поля введення, списки, таблиці, анімації, зображення тощо, які сприяють зручній та ефективній взаємодії з програмою.

Важливо, щоб вебсторінка була розроблена з урахуванням кращих практик у сфері вебдизайну та веброзробки, забезпечуючи користувачам зручне та збалансоване середовище для використання сайту та досягнення їх мети.

При відкритті вебсторінки користувачам можуть бути доступні різні опції, такі як перегляд товарів, додавання та віднімання в кошику, здійснення покупок, перегляд індивідуальної сторінки товару та оформлення замовлення представлено на рис. 4.2 – 4.4.

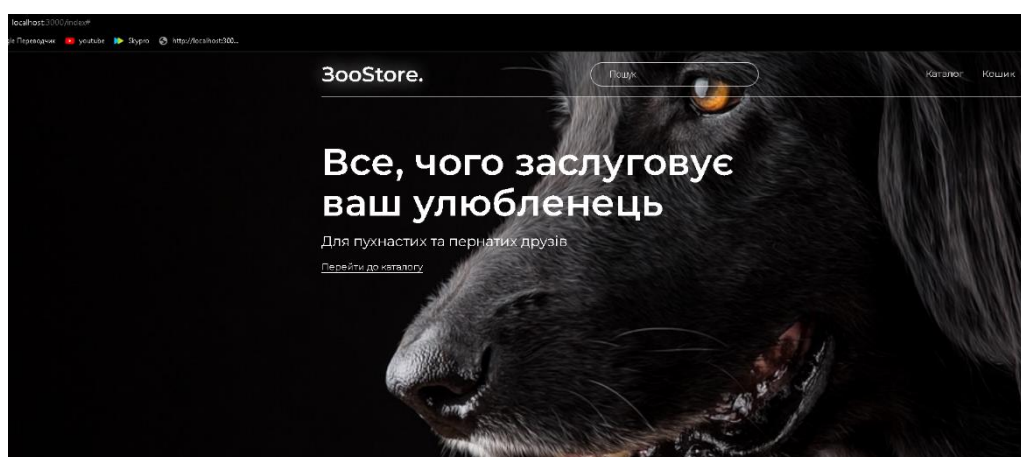


Рисунок 4.1– URL-адреса

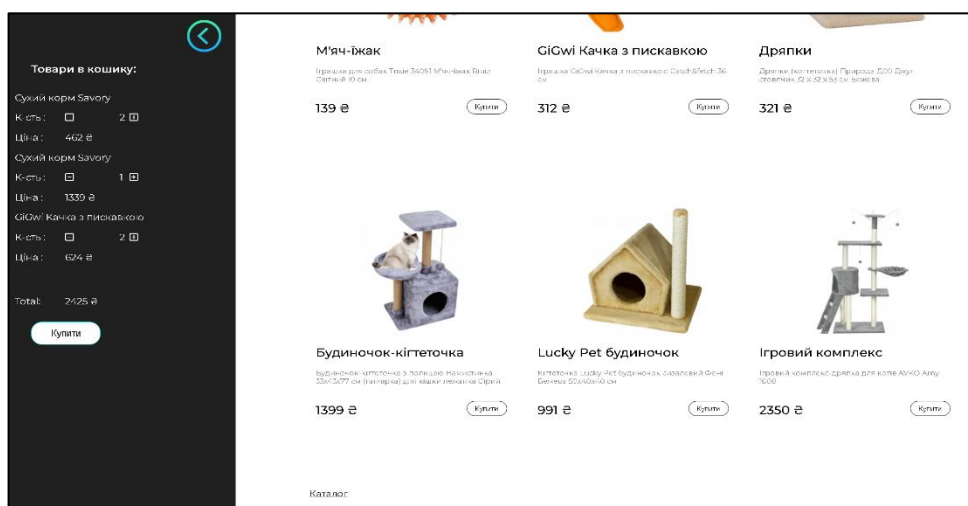


Рисунок 4.2– Кошик

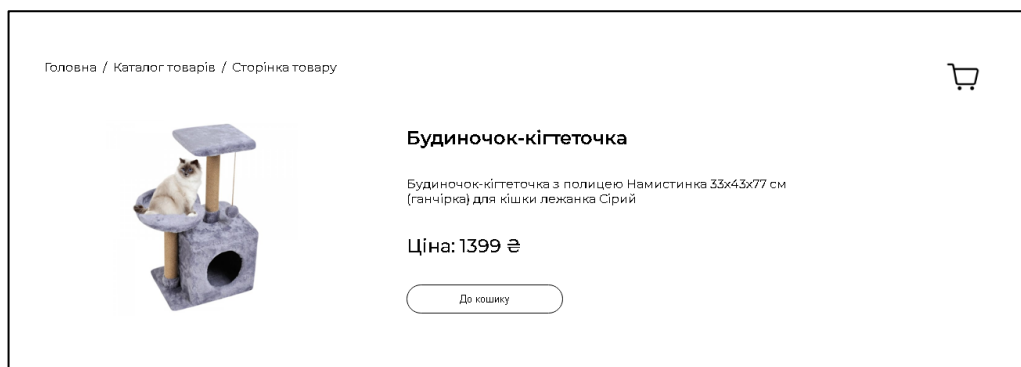


Рисунок 4.3– Сторінка товару

Кошик :

Сухий корм Savory
К-сть: 2
Ціна: 462 ₴

Сухий корм Savory
К-сть: 1
Ціна: 1339 ₴

О'Ом! Качка з пшавкою
К-сть: 2
Ціна: 624 ₴

Total: 2425 ₴

Оформлення замовлення

Ім'я

Телефон

Пошта

Адреса

☒ Правила та умови

[Перейти до каталогу](#)

ZooStore.

Меню
Головна
Каталог
Кошки

Контакти
+380-969-733-788
grinpis01@gmail.com
f o t

Отримуйте наші новини та оновлення

Рисунок 4.4 – Оформлення замовлення

В результаті оформлення замовлення прийде лист на пошту представлено на рис. 4.5.

Нове замовлення магазин зоотоварів
Сухий корм Savory - 2 - 462 гривень
Сухий корм Savory - 1 - 1339 гривень
GiGwi Качка з пискавкою - 2 - 624 гривень
Total 2425 гривень
Phone: +380989733788
Username: SonRayS
Address: Полякова 136
Email: grinpis01@gmail.com

Рисунок 4.5 – Повідомлення про замовлення

1.3 Тестування вебзастосунку

Для перевірки коректності реалізації вебдодатку був використаний метод ручного тестування, що включає проходження всіх сценаріїв користувача вручну.

Цим підходом було перевірено правильність виконання наступних сценаріїв:

- перехід на URL-адресу магазину;
- перегляд товарів;
- сторінка товару;
- додавання товарів до кошика;
- редагування товарів в кошику;
- перехід на сторінку каталогу;
- сортування товарів;
- перехід по хлібних крихтах;
- оформлення товару;
- отримання листа замовлення.

Було проведено перевірку адаптивності дизайну шляхом оцінки відображення елементів керування вебдодатку на різних роздільній здатності дисплея.

Цей етап тестування дозволив перевірити, чи належним чином вебзастосунок реагує на зміну розмірів екрана та пристосовує свій дизайн для забезпечення зручності використання незалежно від типу пристрою, на якому він відображається.

4.4 Висновки за розділом 4

В процесі розробки був зроблений інтернет-магазин для тварин, він розроблений з урахуванням сучасного дизайну, що відображає його спеціалізацію та привабливість для відвідувачів.

Дизайн сайту включає гармонійну комбінацію кольорів, привабливі шрифти та графіку, які допомагають створити привітну та привабливу атмосферу для користувачів. Зображення товарів високої якості та привертають увагу відвідувачів.

Інтерфейс сайту розроблений з урахуванням зручності навігації та взаємодії з користувачем. Меню та розділи сайту розташовані логічно, що дозволяє легко знайти необхідну інформацію та перейти до відповідних розділів.

Дизайн сайту також пристосований до різних пристроїв і роздільних здатностей екранів. Це забезпечує користувачам зручну інтеракцію з сайтом незалежно від пристрою, на якому вони його відкривають.

Після описування сценаріїв користувача та основних елементів вебінтерфейсу вебдодатку, було проведено тестування розробленого вебзастосунку, включаючи перевірку функціональності та адаптивності дизайну.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. HTML and CSS: Design and Build Websites – by Jon Duckett [Electronic resource]. – Access mode: <https://wtf.tw/ref/duckett.pdf>.
2. Relational vs. Non-Relational Database: Pros & Cons [Electronic resource]. – Access mode: <https://aloo.co/blog/relational-vs-non-relational-database-pros-cons>.
3. Documentation jQuery API [Electronic resource]. – Режим доступу <https://api.jquery.com/>.
4. Comparing Database Management Systems: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch, and others [Electronic resource]. – Access mode: <https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others>.
5. Instruction for nodemon [Electronic resource]. – Access mode: <https://www.npmjs.com/package/nodemon?activeTab=readme>.
6. Introduction to JSON Web Tokens [Electronic resource]. – Access mode: <https://jwt.io/introduction>.
7. Documentation MySQL [Electronic resource]. – Access mode: <https://dev.mysql.com/>.
8. Sequelize is a modern TypeScript and Node.js ORM for Postgres, MySQL, MariaDB, SQLite and SQL Server, and more. Featuring solid transaction support, relations, eager and lazy loading, read replication and more. [Electronic resource]. – Access mode: <https://sequelize.org/>.
9. Node.js HTTP Documentation [Electronic resource]. – Access mode: <https://nodejs.org/api/http.html>.
10. Express Docs [Electronic resource]. – Access mode: <https://expressjs.com/>.
11. Webpack Typescript plugin [Electronic resource]. – Access mode: <https://webpack.js.org/guides/typescript/>.

12. Express cors middleware [Electronic resource]. – Access mode: <https://expressjs.com/en/resources/middleware/cors.html>.
13. Color Mode – Chakra UI [Electronic resource]. – Access mode: <https://chakra-ui.com/docs/styled-system/color-mode>.
14. Nodemailer is a module for Node.js applications to allow easy as cake email sending Documentation [Electronic resource]. – Access mode: <https://nodemailer.com/about/>.
15. JavaScript electronic modern resource [Electronic resource]. – Access mode: <https://developer.mozilla.org/>.
16. Express Fast, unopinionated, minimalist web framework for Node.js [Electronic resource]. – Access mode: <https://expressjs.com/>.
17. Мельник В. М. Дослідження швидкодії виконання запиту в СУБД MySQL та MariaDB засобами Python та PHP [Текст] / В. М. Мельник, А. І. Нагорнюк, К. Я. Бортник // Комп'ютерно–інтегровані технології: освіта, наука, виробництво. – 2019. – № 36. – С. 123–127.
18. Documentation PHP [Electronic resource]. – Access mode: <https://php.net/>.
19. Інструкція до SCSS / SASS [Електронний ресурс] – Режим доступу: <https://ivanmalaniak.pp.ua/it-blog/sass-scss/povna-instruktsiia-do-scss-sass/>.
20. Офіційний сайт Pug [Електронний ресурс] – Режим доступу: <https://pugjs.org/api/getting-started.html>.

ДОДАТОК А
Технічне завдання

Вступ

Створення онлайн-магазину є чудовим способом розширення аудиторії та збільшення продажі. Одним з ключових елементів буде розробка зручного та інтуїтивно зрозумілого інтерфейсу, що дозволить покупцям швидко знайти потрібні їм товари та оформити замовлення. Метою проєкту є реалізація інтернет-магазину зоотоварів, що дозволить покупати через мережу інтернет. Розроблений застосунок буде надавати функції реєстрації, покупки, сортування товарів а також додавання.

A.1 Підстави для розробки

Дипломна кваліфікаційна робота виконується на основі завдання, виданого за темою «Програмна реалізація інтернет-магазину зоотоварів» у відповідності з наказом № 87 від 4 квітня 2023 року затверджено Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Програма має реалізувати функції отримання каталогу товарів, їх сортування, пошуку та замовлення, а також можливістю отримання зворотнього зв'язку з користувачем.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Набір функціональних вимог, які має реалізувати програма, включає:

- пошук товарів;
- сортування товарів;
- додавання товарів у кошик

- авторизація адміністратора;
- можливість зворотного зв'язку;
- додавання товарів; – видалення товарів;
- редагування замовлення.

A.3.2 Умови експлуатації

Розроблений сайт необхідно розмістити в мережі Інтернет, повинна працювати кросбраузерність. Кінцевими користувачами тут можуть бути як клієнти інтернет-магазину так і обслуговуючий персонал.

Щоб скористатися інтернет-магазин як клієнт або адміністратор необхідно мати персональний комп'ютер з доступом до Інтернету. Браузер і доступ до Інтернету. Сайт повинен мати вебсервер і базу даних з Інформація про продукт та клієнтів.

A.3.3 Вимоги до складу та параметрів технічних засобів

Мінімальні системні вимоги та ПЗ, що необхідні для функціонування

- сервер та хостинг;
- сервер (IBM-сумісна архітектура);
- серверна ОС: Windows Server;
- MySQL-сервер для управління користувачами та організації
- доступу до БД.
- редактор коду Visual Studio Code.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- ПК користувача (IBM-сумісна архітектура);
- процесор із тактовою частотою від 2 GHz;
- обсяг ОЗП – від 4GB (для безперебійної роботи нових версій браузерів);

- підключення до мережі Internet;
- установлений браузер.

A.4 Порядок контролю та приймання

Виконання дипломної кваліфікаційної роботи повинно відбуватися відповідно до календарного плану, згідно встановлених завдань. Їх термінів виконання і результатів виконання.

ДОДАТОК Б
Текст програми

```
let express = require('express');
let app = express();

app.use(express.static('public'));
app.use(express.static('views'));

app.set('view engine', 'pug');

let mysql = require('mysql');

app.use(express.json());
const nodemailer = require('nodemailer')

let con = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'Teamshotgotkill_1',
  database: 'market'
});

con.connect(function (err) {
  if (err) throw err;
  console.log("Connected!");
});

process.env["NODE_TLS_REJECT_UNAUTHORIZED"] = 0; //off search autoriz

app.listen(3000, function () {
  console.log('node express work on 3000');
});
```

```

function getGoodsByCategory(category) {
  return new Promise(function (resolve, reject) {
    con.query(`SELECT * FROM goods WHERE category = '${category}'`,
      function (error, result) {
        if (error) reject(error);
        resolve(result);

        let user = {};
        for (let i = 0; i < result.length; i++) {
          user[result[i]['id']] = result[i];
        }
        console.log(JSON.parse(JSON.stringify(user)))
      });
  });
}

```

```

app.get('/catalog', function (req, res) {
  console.log('load /catalog');
  console.log("Connected!");

```

```

let nap = getGoodsByCategory('Наповнювачі');

```

```

let korm = getGoodsByCategory('Корм');

```

```

let game = getGoodsByCategory('Іграшки');

```

```

let homes = getGoodsByCategory('Будиночки');

```

```

let bed = getGoodsByCategory('Лежаки');

```

```

Promise.all([nap, korm, game, homes, bed]).then(function (value) {
  res.render('catalog', {
    nap: JSON.parse(JSON.stringify(value[0])),
    korm: JSON.parse(JSON.stringify(value[1])),
    game: JSON.parse(JSON.stringify(value[2])),
    homes: JSON.parse(JSON.stringify(value[3])),
    bed: JSON.parse(JSON.stringify(value[4]))
  });
});
});
});

```

```

app.get('/index', function (req, res) {
  console.log(req.query.id);

```

```

let nap = getGoodsByCategory('Наповнювачі');

```

```

let korm = getGoodsByCategory('Корм');

```

```

let game = getGoodsByCategory('Іграшки');

```

```

let homes = getGoodsByCategory('Будиночки');

```

```

let bed = getGoodsByCategory('Лежаки');

```

```

let goods = new Promise(function (resolve, reject) {
  con.query(
    'SELECT * FROM goods id',
    function (error, result) {
      if (error) reject(error);

```

```

    resolve(result);
  });
});

```

```

Promise.all([goods, nap, korm, game, homes, bed]).then(function (value) {
  res.render('index', {
    goods: JSON.parse(JSON.stringify(value[0])),
    nap: JSON.parse(JSON.stringify(value[1])),
    korm: JSON.parse(JSON.stringify(value[2])),
    game: JSON.parse(JSON.stringify(value[3])),
    homes: JSON.parse(JSON.stringify(value[4])),
    bed: JSON.parse(JSON.stringify(value[5]))
  });
})
});

```

```

app.get('/goods', function (req, res) {
  console.log(req.query.id);
  con.query('SELECT * FROM goods WHERE id=' + req.query.id, function (error,
result, fields) {
    if (error) throw error;
    res.render('goods', { goods: JSON.parse(JSON.stringify(result)) });
  });
});

```

```

app.post('/get-goods-info', function (req, res) {
  console.log(req.body.key);
  if (req.body.key.length != 0) {
    con.query('SELECT id,name,cost FROM goods WHERE id IN (' +
req.body.key.join(',') + ')', function (error, result, fields) {

```

```

    if (error) throw error;
    console.log(result);
    let goods = {};
    for (let i = 0; i < result.length; i++) {
        goods[result[i]['id']] = result[i];
    }
    res.json(goods);
});
}
else {
    res.send('0');
}
});

app.get('/order', function (req, res) {
    res.render('order');
});

app.post('/get-category-list', function (req, res) {
    // console.log(req.body);
    con.query('SELECT id, category FROM category', function (error, result, fields)
    {
        if (error) throw error;
        console.log(result)
        res.json(result);
    });
});

app.post('/finish-order', function (req, res) {
    console.log(req.body);

```

```

if (req.body.key.length != 0) {
  let key = Object.keys(req.body.key);
  con.query(
    'SELECT id,name,cost FROM goods WHERE id IN (' + key.join(',') + ')',
    function (error, result, fields) {
      if (error) throw error;
      console.log(result);
      sendMail(req.body, result).catch(console.error);
      saveOrder(req.body, result);
      res.send('1');
    });
}
else {
  res.send('0');
}
});

```

```

function saveOrder(data, result) {
  let sql;
  sql = "INSERT INTO user_info (user_name, user_phone, user_email,address)
VALUES ('" + data.username + "', '" + data.phone + "', '" + data.email + "', '" +
data.address + "')";
  con.query(sql, function (error, resultQuery) {
    if (error) throw error;
    console.log("1 user save");
    console.log(resultQuery);
    let userId = resultQuery.insertId;
    date = new Date() / 1000;
    for (let i = 0; i < result.length; i++) {

```

```

    sql = "INSERT INTO shop_order (date, user_id, goods_id, goods_cost,
goods_amount, total) VALUES (" + date + "," + userId + "," + result[i]['id'] + ", "
+ result[i]['cost'] + "," + data.key[result[i]['id']] + ", " + data.key[result[i]['id']] *
result[i]['cost'] + ")";

    con.query(sql, function (error, resultQuery) {
        if (error) throw error;
        console.log("1 saved")
    })
}
});
}

```

```

async function sendMail(data, result) {
    let res = '<h2>Нове замовлення магазин зоотоварів</h2>';
    let total = 0;
    for (let i = 0; i < result.length; i++) {
        res += `<p>${result[i]['name']} - ${data.key[result[i]['id']]} -
            ${result[i]['cost'] * data.key[result[i]['id']]} гривень</p>`;
        total += result[i]['cost'] * data.key[result[i]['id']];
    }
    console.log(res);
    res += '<hr>';
    res += `Total ${total} гривень`;
    res += `<hr>Phone: ${data.phone}`;
    res += `<hr>Username: ${data.username}`;
    res += `<hr>Address: ${data.address}`;
    res += `<hr>Email: ${data.email}`;

    let testAccount = await nodemailer.createTestAccount();

```

```

let transporter = nodemailer.createTransport({
  host: "smtp.ethereal.email",
  port: 587,
  secure: false, // true, false for other ports
  auth: {
    user: testAccount.user,
    pass: testAccount.pass
  }
});

```

```

let mailOption = {
  from: '<grinpis01@gmail.com>',
  to: "grinpis01@gmail.com," + data.email,
  subject: "Магазин зоотоварів",
  text: 'Работает!',
  html: res
};

```

```

let info = await transporter.sendMail(mailOption);
console.log("MessageSent: %s", info.messageId);
console.log("PreviewSent: %s", nodemailer.getTestMessageUrl(info));
return true;
}

document.querySelector('.close-nav').onclick = closeNav;
document.querySelector('.show-nav').onclick = showNav;

function closeNav() {
  document.querySelector('.site-nav').style.left = '-425px';
}

function showNav() {

```

```

    document.querySelector('.site-nav').style.left = '0';
}

window.onscroll = function() {scrollFunction()};

function scrollFunction() {
    if (document.body.scrollTop > 2 || document.documentElement.scrollTop > 2) {
        document.getElementById("myBtn").style.display = "block";
    } else {
        document.getElementById("myBtn").style.display = "none";
    }
}

// Работа с {}
let card = {};

document.querySelectorAll('.add-to-card').forEach(function(element){
    element.onclick = addToCard;
});

if (localStorage.getItem('card')) {
    card = JSON.parse(localStorage.getItem('card'));
    ajaxGetGoodsInfo();
}

function addToCard() {
    let goodsId = this.dataset.goods_id;
    if (card[goodsId]) {
        card[goodsId]++;
    }
    else {
        card[goodsId] = 1;
    }
}

```

```

    }
    console.log(card);
    ajaxGetGoodsInfo();
  }

```

```

function ajaxGetGoodsInfo() {
  updateLocalStorageCard();
  fetch('/get-goods-info', {
    method: 'POST',
    body: JSON.stringify({ key: Object.keys(card) }),
    headers: {
      'Accept': 'application/json',
      'Content-Type': 'application/json'
    }
  })
  .then(function (response) {
    return response.text();
  })
  .then(function (body) {
    console.log(body);
    showCard(JSON.parse(body));
  })
}

```

```

function showCard(data) {
  let out = '<table class="table table-striped table-card"><tbody>';
  let total = 0;
  for (let key in card){
    out +=`<tr><td colspan="4" class='movion'><a class='movion'
href="/goods?id=${key}">${data[key]['name']}</a></tr>`;
  }
}

```

```

    out += `<tr><td><p class='movions'>К-сть :</p></td><td class='movion'><i
class="far fa-minus-square cart-minus" data-goods_id="${key}"></i></td>`;
    out += `<td class='movion'>${card[key]}</td>`;
    out += `<td class='movion'><class='movion'><i class="far fa-plus-square cart-
plus" data-goods_id="${key}"></i></td>`;
    out += '</tr>';
    out += `<tr class='movion'><td><p class='movions'>Ціна : </p></td><td
class='movion'>${formatPrice(data[key]['cost'] * card[key])} ₴</td>`
    out += '</tr>';
    total += card[key] * data[key]['cost'];
}

    out += `<tr><td colspan="1" class='movion-total'>Total: </td><td
class='movion-total'>${formatPrice(total)} ₴</td></tr>`;
    out += '</tbody></table>';
    document.querySelector('#card-nav').innerHTML = out;
    document.querySelectorAll('.cart-minus').forEach(function (element) {
        element.onclick = cardMinus;
    });
    document.querySelectorAll('.cart-plus').forEach(function (element) {
        element.onclick = cardPlus;
    });
}

function cardPlus() {
    let goodsId = this.dataset.goods_id;
    card[goodsId]++;
    ajaxGetGoodsInfo();
}

function cardMinus() {

```

```

let goodsId = this.dataset.goods_id;
if (card[goodsId] - 1 > 0) {
    card[goodsId]--;
}
else {
    delete (card[goodsId]);
}
ajaxGetGoodsInfo();
}

function formatPrice(price) {
    return price.toFixed().replace(/\d(?:=(\d{3})+\.)/g, '$& ');
}

function updateLocalStorageCard() {
    localStorage.setItem('card', JSON.stringify(card));
}

document.querySelector('#lite-shop-order').onsubmit = function (event) {
    event.preventDefault();
    let username = document.querySelector('#username').value.trim();
    let phone = document.querySelector('#phone').value.trim();
    let email = document.querySelector('#email').value.trim();
    let address = document.querySelector('#address').value.trim();

    if (!document.querySelector('#rule').checked) {
        //с правилами не согласен
        Swal.fire({
            title: 'Помилка !',
            text: 'Прочитайте та прийміть правила',
            type: 'info',

```

```

        confirmButtonText: 'OK'
    });
    return false;
}

if (username == " || phone == " || email == " || address == ") {
    //не заполнены поля
    Swal.fire({
        title: 'Помилка !',
        text: 'Заповніть всі поля',
        type: 'info',
        confirmButtonText: 'OK'
    });
    return false;
}

fetch('/finish-order', {
    method: 'POST',
    body: JSON.stringify({
        'username': username,
        'phone': phone,
        'address': address,
        'email': email,
        'key': JSON.parse(localStorage.getItem('card'))
    }),
    headers: {
        'Accept': 'application/json',
        'Content-Type': 'application/json'
    }
}

```

```

    })
    .then(function (response) {
        return response.text();
    })
    .then(function (body) {
        if (body == 1) {
            Swal.fire({
                title: 'Готово',
                text: 'ок',
                type: 'info',
                confirmButtonText: 'ОК'
            });
        }
        else {
            Swal.fire({
                title: 'Помилка відправки',
                text: 'Помилка !',
                type: 'помилка',
                confirmButtonText: 'ОК'
            });
        }
    })
}

doctype html
head
    meta(charset='utf-8')
    meta(http-equiv='X-UA-Compatible' content='IE=edge')
    meta(name='viewport' content='width=device-width, initial-scale=1.0')
    link(rel='preconnect' href='https://fonts.googleapis.com')
    link(rel='preconnect' href='https://fonts.gstatic.com' crossorigin="")

```

```

link(href='https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600&display=swap' rel='stylesheet')
link(rel='preconnect' href='https://fonts.googleapis.com')
link(rel='preconnect' href='https://fonts.gstatic.com' crossorigin='')
link(href='https://fonts.googleapis.com/css2?family=Poppins:wght@600&display=swap' rel='stylesheet')
link(rel='stylesheet' href='css/style.css')
link(rel='icon' href='favicon.ico' type='images/x-icon')
title 3ooStore.

```

```
.top.center
```

```
// Head-top-menu
```

```
header
```

```
.header
```

```
a(href='index')
```

```
p.header__logo-text 3ooStore.
```

```
// <a href="index"></a>
```

```
input.header__input(type='text' placeholder='Пошук')
```

```
nav
```

```
a.header__link(href='catalog') Каталог
```

```
a.header__link(href='order') Кошик
```

```
.phone
```

```
.phone__menu
```

```
a(href='index')
```

```
p.header__logo-text 3ooStore.
```

```
.phone__box
```

```
a(href='#')
```

```
img.phone__img(src='../img/f.png' alt='')
```

```
a(href='catalog')
```

```

        img.phone__img(src='../img/fff.png' alt='')
        a(href='order')
        img.phone__img(src='../img/ff.png' alt='')
// Head-top-menu
// Head-under-text
section
    .Under-text
        h1.Under-text__H1 Все, чого заслуговує ваш улюбленець
        p.Under-text__p Для пухнастих та пернатих друзів
        a.Under-text__a.Under-text__catalog(href='catalog') Перейти до каталогу
// Head-under-text
button(class="show-nav" id="myBtn")
a(href='#' class='show-nav')
    img(src='../img/chevron.png' alt='')
// Block-mebel
section
    .Mebel.center
        h2.Mebel__style-h2 Все що треба для.....
        .Mebel__for
            .Mebel__foto.Mebel__foto-0.Mebel__foto-big
                a(href='catalog#FEED') КОРМ
            .Mebel__foto.Mebel__foto-1.Mebel__foto-big
                a(href='catalog#Sunbeds') ЛЕЖАКИ
            .Mebel__foto.Mebel__foto-2
                a(href='catalog#home') БУДИНОЧКИ
            .Mebel__foto.Mebel__foto-3
                a(href='catalog#play') ІГРАШКИ
            .Mebel__foto.Mebel__foto-4
                a(href='catalog#Napovnyuvachi') НАПОВНЮВАЧІ
// Block-mebel

```

```

// Special-sells

include nav.pug

.center.products__style-h2.Special-sell
  h3 Товары
.center

.center.Mebel__style-h2
  .product-catalog.center
    each val, index in game
      include card.pug

.center.Mebel__style-h2
  .product-catalog.center
    each val, index in homes
      include card.pug

.center
  .link-catalog__text
    a.link-catalog__text(href='catalog')
      | Каталог

// footer
footer
  .center
    .footer
      .footer__logo
        // 
        a(href='index')

```

```

    p.footer__logo ZooStore.
.footer__block
    h4 Меню
.footer__downbar
    a(href='index') Головна
    a(href='catalog') Каталог
    a(href='order') Кошик
.footer__block
    h4 Контакти
.footer__downbar
    a(href='#') +380-989-733-788
    a(href='https://www.google.com/search?q=help%40interier.com&oq=help%
40interier.com&aqs=chrome..69i57j69i58.714j0j7&sourceid=chrome&ie=UTF-8')
grinpis01@gmail.com
.footer__block-links
    a(href='https://www.facebook.com/')
    img(src='../img/facebook.svg' alt='facebook')
    a(href='https://www.instagram.com/')
    img(src='../img/inst.svg' alt='instagram')
    a(href='https://twitter.com/?lang=ru')
    img(src='../img/twitter.svg' alt='twitter')
.footer__block.footer__button
    p.footer__text-р Отримуйте наші новини та оновлення
    input(type='email' placeholder='Введіть пошту')
    button Підписатися
// footer

script(src='js/card.js')
script(src='js/nav.js')
script(src='js/button.js')

```

```

script(src='js/order.js')
script(src="https://kit.fontawesome.com/0748402994.js"
crossorigin="anonymous")
script(src='https://cdn.jsdelivr.net/npm/sweetalert2@8')

doctype html
head
  meta(charset='utf-8')
  meta(http-equiv='X-UA-Compatible' content='IE=edge')
  meta(name='viewport' content='width=device-width, initial-scale=1.0')
  link(rel='preconnect' href='https://fonts.googleapis.com')
  link(rel='preconnect' href='https://fonts.gstatic.com' crossorigin='')
  link(href='https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600&display=swap' rel='stylesheet')
  link(rel='preconnect' href='https://fonts.googleapis.com')
  link(rel='preconnect' href='https://fonts.gstatic.com' crossorigin='')
  link(href='https://fonts.googleapis.com/css2?family=Poppins:wght@600&display=swap' rel='stylesheet')
  link(rel='stylesheet' href='css/style.css')
  link(rel='icon' href='favicon.ico' type='images/x-icon')
  title 3ooStore.
.top-basket.center
// Head-top-menu
header
  .header
    a(href='index')
      p.header__logo-text 3ooStore.
      // <a href="index"></a>
      input.header__input(type='text' placeholder='Поиск')

```

```

nav
  a.header__link(href='catalog') Каталог
  a.header__link(href='order') Кошик
.phone
  .phone__menu
    a(href='index')
      p.header__logo-text ZooStore.
  .phone__box
    a(href='#')
      img.phone__img(src='../img/f.png' alt='')
    a(href='catalog')
      img.phone__img(src='../img/fff.png' alt='')
    a(href='order')
      img.phone__img(src='../img/ff.png' alt='')
// Head-top-menu
// Head-under-text
section
  .Under-text
    h1.Under-text__H1 &Kcy;&ocy;&shcy;&icy;&kcy;
// Head-under-text
// BASKET

.center.cart-move
  .cart__display-interface
    h2(class="cart-headers") Кошик :
    div(class=".category-list")
      div(id="card-nav")

  .cart__display-interface.cart__input-date.center
    h2.cart__head Оформлення замовлення

```

.cart__head-move

```
form#lite-shop-order(action="")
```

```
  div(class='form-control')
```

```
    label(for='username')
```

```
    input(
```

```
      type='text',
```

```
      id='username',
```

```
      placeholder="Имя",
```

```
    )
```

```
  div(class='form-control')
```

```
    label(for='phone')
```

```
    input(
```

```
      type='text',
```

```
      id='phone',
```

```
      placeholder='Телефон',
```

```
    )
```

```
  div(class='form-control')
```

```
    label(for='email')
```

```
    input(
```

```
      type='text',
```

```
      id='email',
```

```
      placeholder='Почта',
```

```
    )
```

```
  div(class='form-control form-controls')
```

```
    label(for='address')
```

```
    input(
```

```
      type='text',
```

```
      id='address',
```

```
      placeholder='Адреса',
```

```
    )
```

```

div(class='form-control')
  label(for='rule')
    input(
      type='checkbox',
      id='rule',
      value='1'
    )
    span Правила та умови
div(class='form-control text-rights')
  button(class='button-primary', type='submit') Замов-
рчателльний вибір

// Supecial-sells
.center
.link-catalog__text
  a.link-catalog__text(href='catalog') &Pcy;&iecy;&rcy;&iecy;&jcy;&tcy;&icy;
&dcy;&ocy; &kcy;&acy;&tcy;&acy;&lcy;&ocy;&gcy;&ucy;

script(src='js/card.js')
script(src='js/nav.js')
script(src='js/order.js')
script(src="https://kit.fontawesome.com/0748402994.js"
crossorigin="anonymous")
script(src='https://cdn.jsdelivr.net/npm/sweetalert2@8')

// footer
.footer
  .center
  .footer
  .footer__logo

```

```

// 
a(href='index')
  p.footer__logo ЗооStore.
.footer__block
  h4 Меню
  .footer__downbar
    a(href='index') Головна
    a(href='catalog') Каталог
    a(href='order') Кошик
  .footer__block
    h4 Контакти
    .footer__downbar
      a(href='#') +380-989-733-788
      a(href='https://www.google.com/search?q=help%40interier.com&oq=help%
40interier.com&aqs=chrome..69i57j69i58.714j0j7&sourceid=chrome&ie=UTF-8')
grinpis01@gmail.com
    .footer__block-links
      a(href='https://www.facebook.com/')
        img(src='../img/facebook.svg' alt='facebook')
      a(href='https://www.instagram.com/')
        img(src='../img/inst.svg' alt='instagram')
      a(href='https://twitter.com/?lang=ru')
        img(src='../img/twitter.svg' alt='twitter')
    .footer__block.footer__button
      p.footer__text-р Отримуйте наші новини та оновлення
      input(type='email' placeholder='Введіть пошту')
      button Підписатися

// footer
doctype html

```

head

```

meta(charset='utf-8')
meta(http-equiv='X-UA-Compatible' content='IE=edge')
meta(name='viewport' content='width=device-width, initial-scale=1.0')
link(rel='preconnect' href='https://fonts.googleapis.com')
link(rel='preconnect' href='https://fonts.gstatic.com' crossorigin='')
link(href='https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600&display=swap' rel='stylesheet')
link(rel='preconnect' href='https://fonts.googleapis.com')
link(rel='preconnect' href='https://fonts.gstatic.com' crossorigin='')
link(href='https://fonts.googleapis.com/css2?family=Poppins:wght@600&display=swap' rel='stylesheet')
link(rel='stylesheet' href='css/style.css')
link(rel='icon' href='favicon.ico' type='images/x-icon')
title 3ooStore.

```

.top-catalog.center

// Head-top-menu

header

.header

a(href='index')

p.header__logo-text 3ooStore.

//

input.header__input(type='text' placeholder='Пошук')

nav

a.header__link(href='catalog') Каталог

a.header__link(href='order') Кошик

.phone

.phone__menu

```

a(href='index')
  p.header__logo-text 3ooStore.
.phone__box
  a(href='#')
    img.phone__img(src='../img/f.png' alt='')
  a(href='catalog')
    img.phone__img(src='../img/fff.png' alt='')
  a(href='order')
    img.phone__img(src='../img/ff.png' alt='')
// Head-top-menu
// Head-under-text
section
  .Under-text
    h1.Under-text__H1 &Kcy;&acy;&tcy;&acy;&lcy;&ocy;&gcy;
// Head-under-text
// Catalog-menu
.block.center
div
  ul.block-pack__ul
    a.block-pack__li(href='index') &Gcy;&ocy;&lcy;&ocy;&vcy;&ncy;&acy;
    a.block-pack__li(href='catalog') &Kcy;&acy;&tcy;&acy;&lcy;&ocy;&gcy;
&tcy;&ocy;&vcy;&acy;&rcy;&iukcy;&vcy;
.menu-drop
  ul.menu-drop__ul
    li.menu-drop__list
      a.menu-drop__a.menu-drop__cost(href='#')
        p &Scy;&ocy;&rcy;&tcy;&ucy;&vcy;&acy;&tcy;&icy; &zcy;&acy; :
      img.menu-drop__img(src='../img/Polygon1.png' alt='')
      ul.menu-drop__box
        li

```

```

        a.menu-drop__cost.menu-drop__cost-1(href='#')
&Pcy;&ocy;&rcy;&yacy;&dcy;&ocy;&kcy;:
&scy;&pcy;&iecy;&rcy;&shcy;&ucy; &ncy;&ocy;&vcy;&iukcy;
    li
        a.menu-drop__cost.menu-drop__cost-2(href='#')
&Pcy;&ocy;&rcy;&yacy;&dcy;&ocy;&kcy;:
&scy;&pcy;&iecy;&rcy;&shcy;&ucy; &zcy;&ncy;&icy;&zncy;&kcy;&icy;
    li
        a.menu-drop__cost.menu-drop__cost-3(href='#')
&Pcy;&ocy;&rcy;&yacy;&dcy;&ocy;&kcy;:
&scy;&pcy;&iecy;&rcy;&shcy;&ucy; &dcy;&iecy;&shcy;&iecy;&vcy;&iukcy;
    li
        a.menu-drop__cost.menu-drop__cost-4(href='#')
&Pcy;&ocy;&rcy;&yacy;&dcy;&ocy;&kcy;:
&scy;&pcy;&iecy;&rcy;&shcy;&ucy;
&pcy;&ocy;&pcy;&ucy;&lcy;&yacy;&rcy;&ncy;&iukcy;

button(class="show-nav" id="myBtn")
  a(href="#" class='show-nav')
    img(src='../img/chevron.png' alt="

include nav.pug

// Catalog-menu
.center.Mebel__style-h2
  h4#FEED #{korm[0]['category']}
  .product-catalog.center
    each val, index in korm
      include card.pug

```

```

.center.Mebel__style-h2
h4#play #{game[0]['category']}
.product-catalog.center
  each val, index in game
    include card.pug

.center.Mebel__style-h2
h4#home #{homes[0]['category']}
.product-catalog.center
  each val, index in homes
    include card.pug

.center.Mebel__style-h2
h4#Sunbeds #{bed[0]['category']}
.product-catalog.center
  each val, index in bed
    include card.pug

.center.products__style-h2
h4#Napovyuvachi #{nap[0]['category']}
.center.products__sell
  each val, index in nap
    include card.pug

// Catalog-flex-poz
.center
.link-catalog__text
  a.link-catalog__text(href='catalog')
    img(src='../img/up-arrow.png' alt=" height='24px' width='24px'
padding='24px')

```

| Наропу

```
// footer
```

```
footer
```

```
.center
```

```
.footer
```

```
.footer__logo
```

```
// 
```

```
a(href='index')
```

```
p.footer__logo 3ooStore.
```

```
.footer__block
```

```
h4 Меню
```

```
.footer__downbar
```

```
a(href='index') Головна
```

```
a(href='catalog') Каталог
```

```
a(href='order') Кошик
```

```
.footer__block
```

```
h4 Контакты
```

```
.footer__downbar
```

```
a(href='#') +380-989-733-788
```

```
a(href='https://www.google.com/search?q=help%40interier.com&oq=help%40interier.com&aqs=chrome..69i57j69i58.714j0j7&sourceid=chrome&ie=UTF-8')
grinpis01@gmail.com
```

```
.footer__block-links
```

```
a(href='https://www.facebook.com/')

```

```
img(src='../img/facebook.svg' alt='facebook')
```

```
a(href='https://www.instagram.com/')

```

```
img(src='../img/inst.svg' alt='instagram')
```

```
a(href='https://twitter.com/?lang=ru')
```

```

        img(src='../img/twitter.svg' alt='twitter')
    .footer__block.footer__button
    p.footer__text-р Отримуйте наші новини та оновлення
    input(type='email' placeholder='Введіть пошту')
    button Підписатися
// footer

script(src='js/card.js')
script(src='js/nav.js')
script(src='js/button.js')
script(src='js/order.js')
script(src="https://kit.fontawesome.com/0748402994.js"
crossorigin="anonymous")
script(src='https://cdn.jsdelivr.net/npm/sweetalert2@8')

doctype html
head
    meta(charset='utf-8')
    meta(http-equiv='X-UA-Compatible' content='IE=edge')
    meta(name='viewport' content='width=device-width, initial-scale=1.0')
    link(rel='preconnect' href='https://fonts.googleapis.com')
    link(rel='preconnect' href='https://fonts.gstatic.com' crossorigin='')
    link(href='https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600&display=swap' rel='stylesheet')
    link(rel='preconnect' href='https://fonts.googleapis.com')
    link(rel='preconnect' href='https://fonts.gstatic.com' crossorigin='')
    link(href='https://fonts.googleapis.com/css2?family=Poppins:wght@600&display=swap' rel='stylesheet')
    link(rel='stylesheet' href='css/style.css')
    link(rel='icon' href='favicon.ico' type='images/x-icon')

```

title 3ooStore.

.top-id.center

// Head-top-menu

header

.header

a(href='index')

p.header__logo-text 3ooStore.

//

input.header__input(type='text' placeholder='Пошук')

nav

a.header__link(href='catalog') Каталог

a.header__link(href='basket') Кошик

.phone

.phone__menu

a(href='index')

p.header__logo-text 3ooStore.

.phone__box

a(href='#')

img.phone__img(src='../img/f.png' alt="")

a(href='catalog')

img.phone__img(src='../img/fff.png' alt="")

a(href='basket')

img.phone__img(src='../img/ff.png' alt="")

// Head-top-menu

// Head-under-text

section

.Under-text

h1.Under-text__H1 Тільки найкраще для вашого улюбленця

```

    p.Under-text__p Життів то дев'ять, але здоров'я одне!
    a.Under-text__a.Under-text__catalog(href='catalog') Перейти до каталогу
// Head-under-text

.center.block
div
    ul.block-pack__ul
        a.block-pack__li(href='index') Головна
        a.block-pack__li(href='catalog') Каталог товарів
        a.block-pack__li(href="#") Сторінка товару
    .block__images
        a(href='basket')
        img(src="../img/cart.png", alt="")
    .center.block-swap
    .block__images
        a(href='catalog')
        img(src="../img/category.png", alt="")
    .block__images
        a(href='basket')
        img(src="../img/cart.png", alt="")
    .price__cost-main

.center
    .goods-box
    .goods-box__img
        img(
            src="img/"+goods[0]['image']
            class="goods-big-image"
        )
    .goods-box__info

```

```

h2 #{goods[0]['name']}
p #{goods[0]['description']}
.price__button
  .price__cost
    .price__cost-main
      span(class="success") Ціна: #{goods[0]['cost']} ₴
.goods-box__button
  button(
    class="button-success add-to-cart"
    data-goods_id=goods[0]['id']
  ) До кошику

```

```

// footer
footer
  .center
    .footer
      .footer__logo
        // 
        a(href='index')
          p.footer__logo ZooStore.
      .footer__block
        h4 Меню
        .footer__downbar
          a(href='index') Головна
          a(href='catalog') Каталог
          a(href='basket') Кошик
      .footer__block
        h4 Контакти
        .footer__downbar
          a(href='#') +380-989-733-788

```

```

a(href='https://www.google.com/search?q=help%40interier.com&oq=help%
40interier.com&aqs=chrome..69i57j69i58.714j0j7&sourceid=chrome&ie=UTF-8')
grinpis01@gmail.com

```

```

.footer__block-links

```

```

a(href='https://www.facebook.com/')

```

```

img(src='../img/facebook.svg' alt='facebook')

```

```

a(href='https://www.instagram.com/')

```

```

img(src='../img/inst.svg' alt='instagram')

```

```

a(href='https://twitter.com/?lang=ru')

```

```

img(src='../img/twitter.svg' alt='twitter')

```

```

.footer__block.footer__button

```

```

p.footer__text-р Отримуйте наші новини та оновлення

```

```

input(type='email' placeholder='Введіть пошту')

```

```

button Підписатися

```

```

// footer

```

```

@import 'normalize';

```

```

@import 'header';

```

```

@import 'under-header-text';

```

```

@import 'menu-drop';

```

```

@import 'block-mebel';

```

```

@import 'Special-sell';

```

```

@import 'catalog-items';

```

```

@import 'catalog-flex';

```

```

@import 'basket';

```

```

@import 'footer';

```

```

@import 'goods';

```

```

@import 'site-nav';

```

```

@import 'tablet';

```

```

@import 'mobile';

```

```

@import 'animation';

```

ДОДАТОК В
Слайди презентації

Національний університет “Запорізька політехніка”

Дипломна кваліфікаційна робота

ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ МАГАЗИНУ
ЗООТОВАРІВ
SOFTWARE IMPLEMENTATION OF ONLINE PET STORE

Виконав :

Ілля ГРИНЬ

Студент групи КНТ-220СП

Керівник :

Тетяна ФЕДОРОНЧАК

Рисунок В.1 – Слайд №1

Об’єкт роботи – програмне забезпечення для продажу зоотоварів.

Предмет роботи – вебсайти для продажу зоотоварів.

Мета роботи – розробити інтернет магазин зоотоварів з можливістю їх замовлення.

Завдання

1. Огляд аналогів;
2. Визначення вимог до програмного продукту;
3. Проектування архітектури вебзастосунку;
4. Визначення засобів реалізації;
5. Програмна реалізація в застосунку;

Рисунок В.2 – Слайд №2

Огляд аналогів			
Критерій	Інтернет-магазин masterzoo	Інтернет-магазин zoomark	Інтернет-магазин petmarket.ua
Можливість перегляду асортименту товарів	ТАК	ТАК	ТАК
Можливість реєстрації/авторизації користувача	ТАК	ТАК	ТАК
Додавання товарів у кошик	ТАК	ТАК	ТАК
Сортування та пошук товарів	ТАК	ТАК	ТАК
Зміна мови локалізації сайту	ТАК	ТАК	ТАК
Зв'язок онлайн з менеджером	ТАК	ТАК	НІ
Повна масштабованість сайту	Не повна	НІ	ТАК
Індивідуальні інтерфейси для всіх типів пристроїв	НІ	ТАК	ТАК
Сучасний дизайн	ТАК	НІ	НІ

Рисунок В.3 – Слайд №3



Рисунок В.4 – Слайд №4

Вибір мови програмування

Критерій	JavaScript	Python
Тип мови	Динамічна	Динамічна
Підхід до ООП	Прототипний	Клас-об'єктний
Синтаксис	Схожий на С	Простий та зрозумілий
Сфера застосування	Веб-розробка (клієнтська та серверна), інтерактивні елементи на веб-сторінках	Веб-розробка, наукові обчислення, аналіз даних, штучний інтелект, автоматизація завдань
Фреймворки та бібліотеки	ТАК	ТАК
Підтримка платформ	Робота у браузері, на сервері (Node.js)	Робота на різних платформах (Windows, macOS, Linux)
Асинхронне програмування	ТАК	ТАК
Робота з потоками	Не повна	ТАК
Продуктивність	ТАК	НІ
Обробка виключень	ТАК	ТАК
Спільнота та підтримка	ТАК	ТАК

Рисунок В.5 – Слайд №5

Вибір взаємодії клієнту з сервером

Критерій	SOAP	REST
Стандарт	SOAP є протоколом обміну повідомленнями на основі XML	REST є архітектурним стилем для побудови веб-сервісів
Протоколи	ТАК	ТАК
Формат обміну даними	Використовується XML для обміну даними	Використовується різні формати, такі як JSON, XML, HTML тощо
Стиль комунікації	Формалізований та жорсткий підхід з визначеними правилами і схемами	Більш гнучкий підхід без жорстких правил, заснований на принципах ресурсів і стану
Продуктивність	НІ	ТАК
Підтримка кешування	ТАК	ТАК
Складність реалізації	ТАК	НІ
Розширюваність	ТАК	ТАК
Сумісність з браузерами	Не повна	ТАК
Спільнота та підтримка	Має певну кількість прихильників та підтримку, але менше активна порівняно з REST	Має широку спільноту та активну підтримку, оскільки є популярним стилем архітектури

Рисунок В.6 – Слайд №6

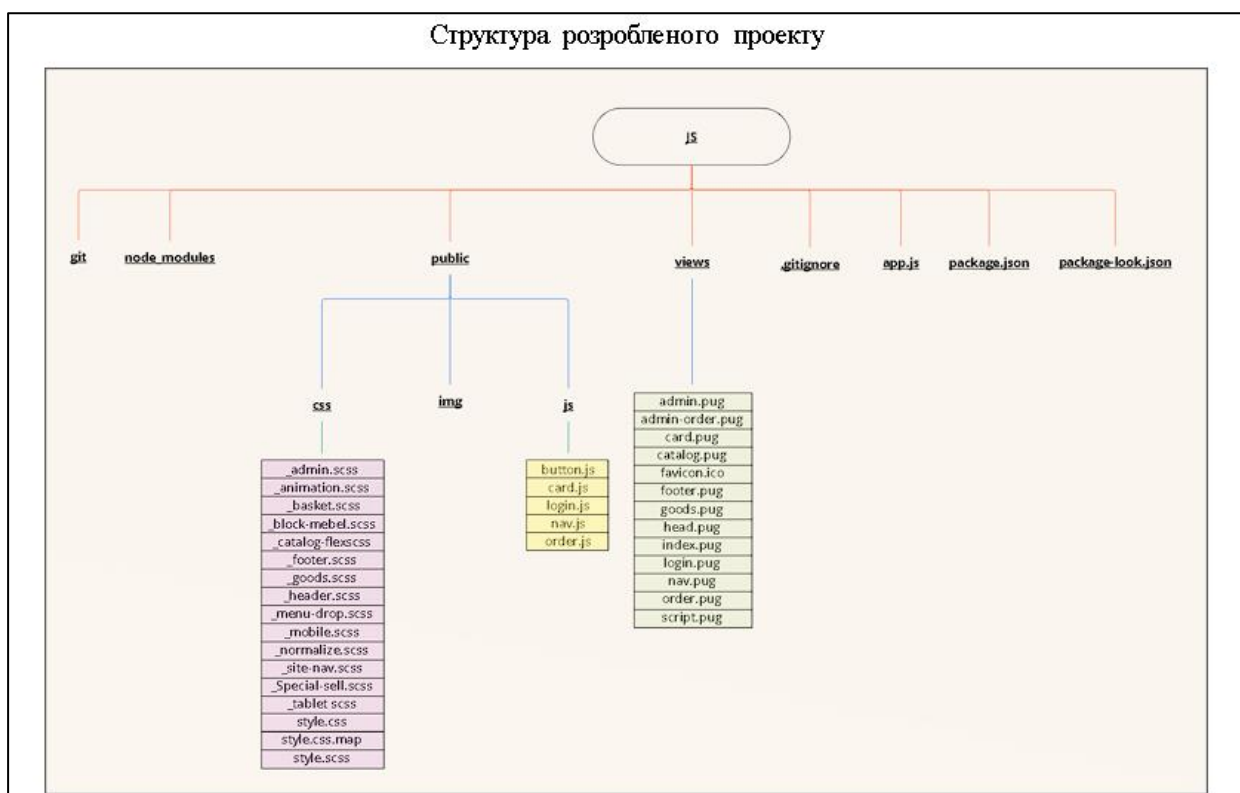


Рисунок В.7 – Слайд №7

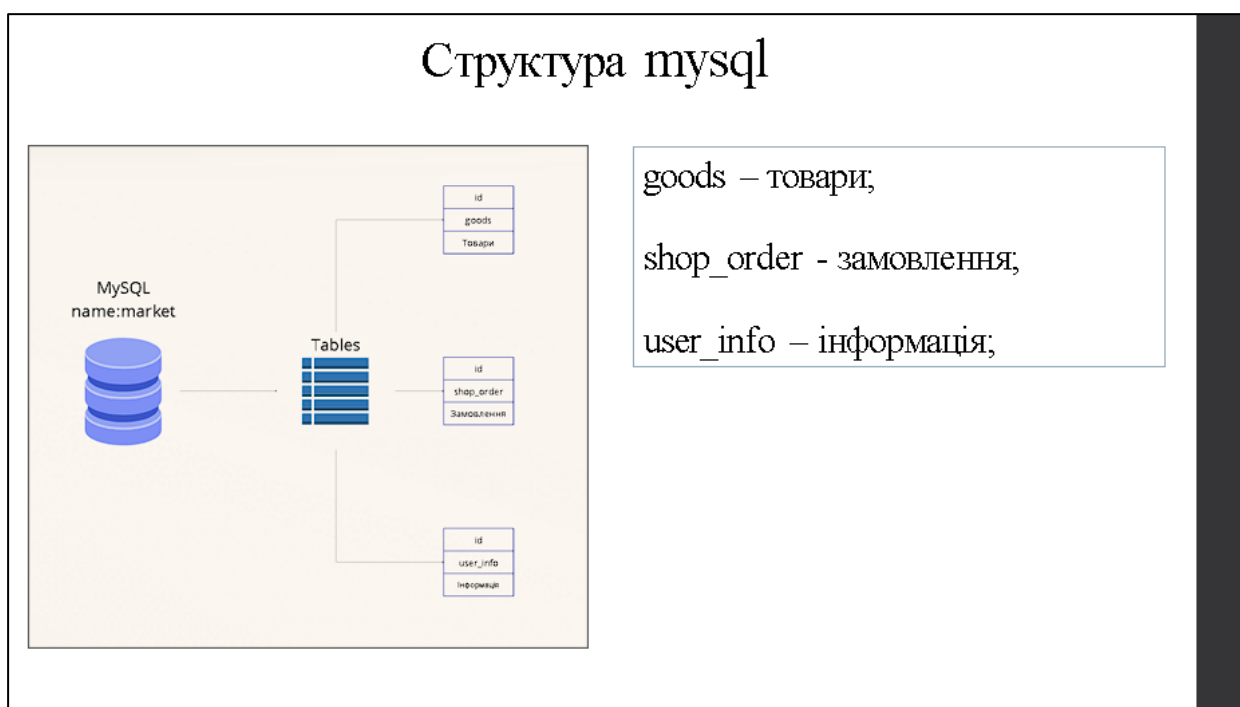


Рисунок В.8 – Слайд №8

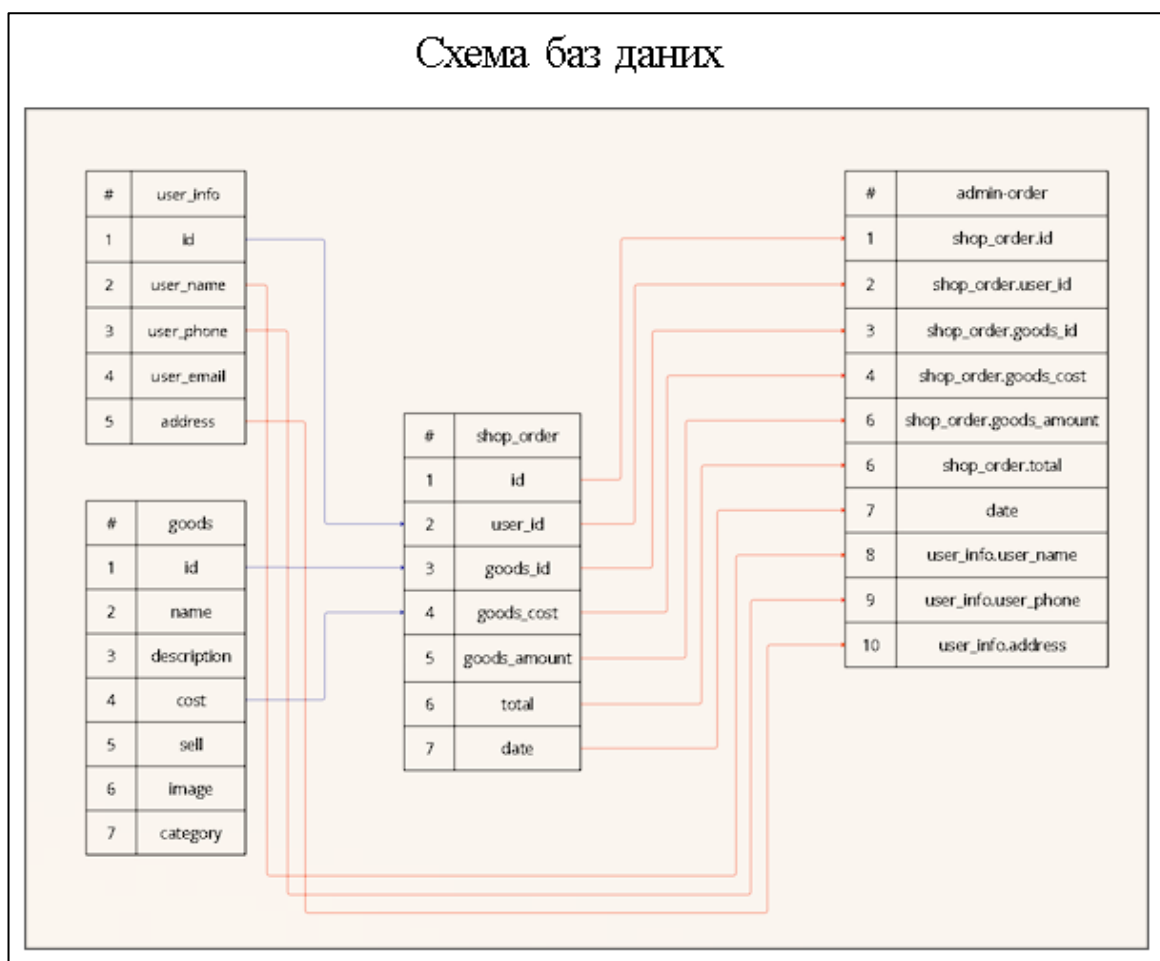


Рисунок В.9 – Слайд №9

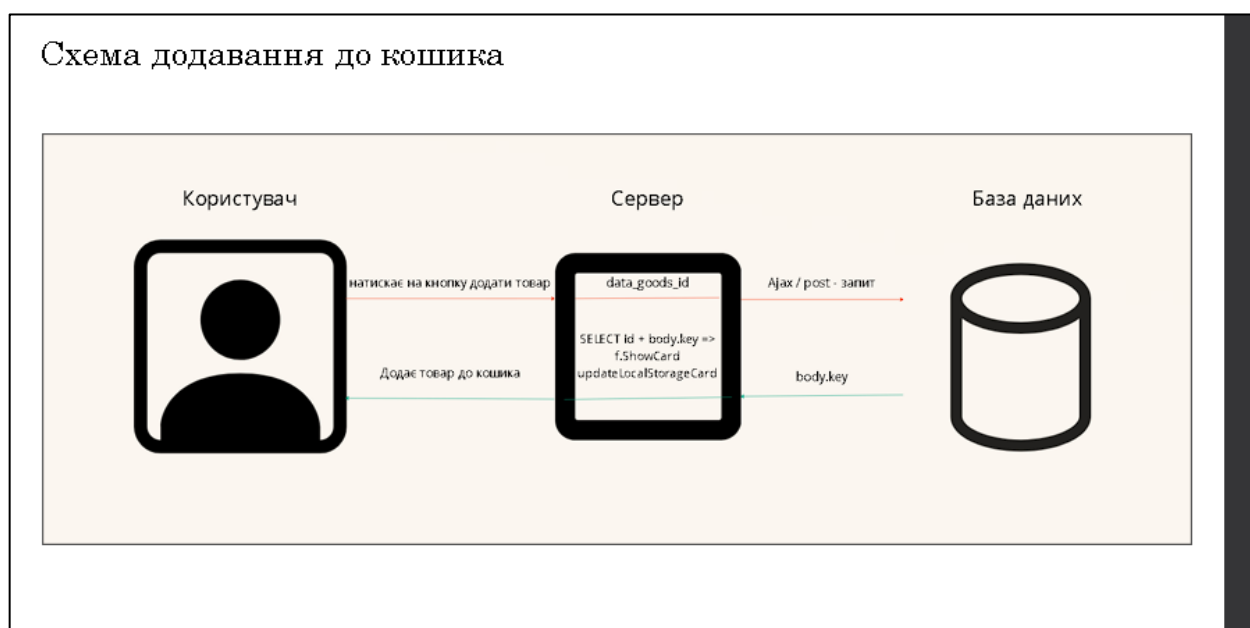


Рисунок В.10 – Слайд №10

Оформлення замовлення

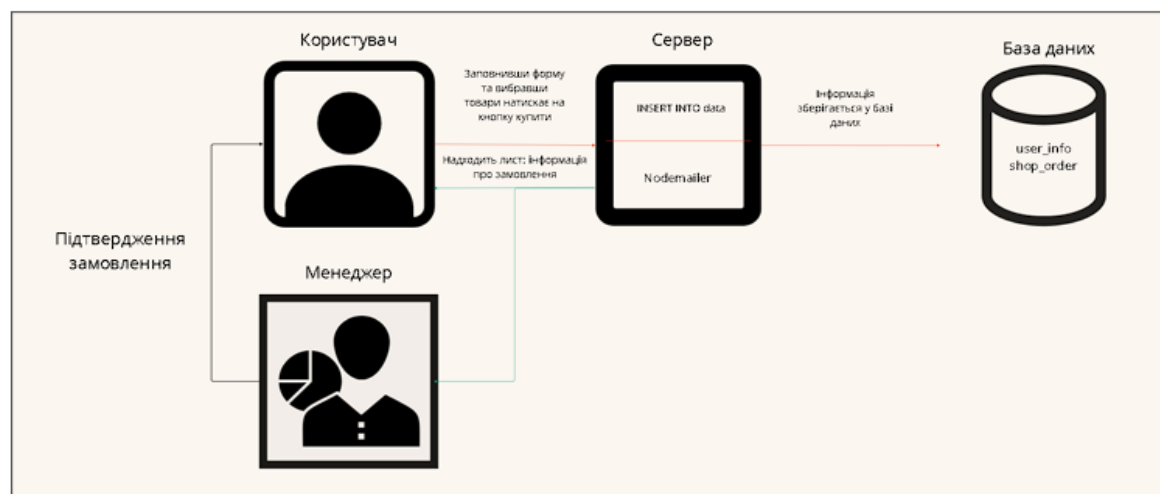
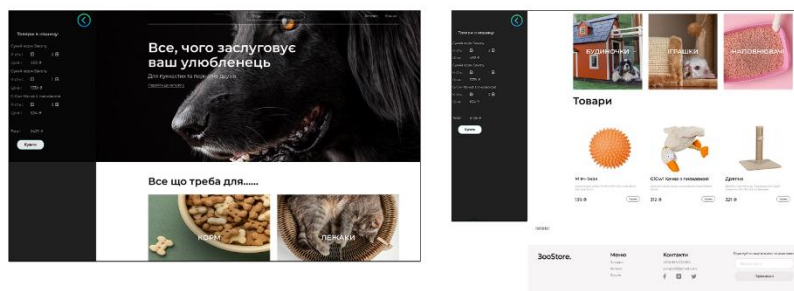


Рисунок В.11 – Слайд №11

Перегляд спроектованого інтернет-магазину



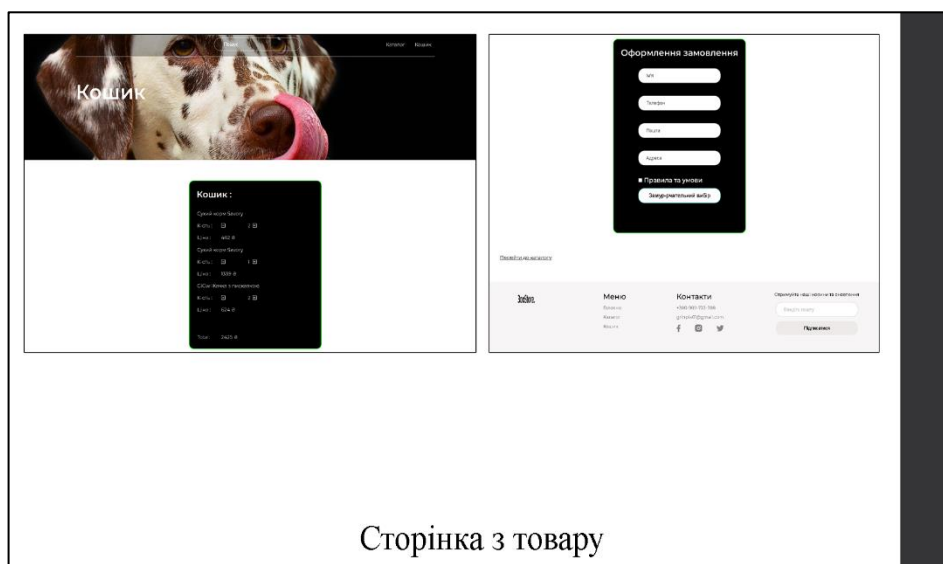
Головна сторінка

Рисунок В.12 – Слайд №12



Сторінка каталогу

Рисунок В.13 – Слайд №13



Сторінка з товару

Рисунок В.14 – Слайд №14

Висновки

У ході розробки дипломної кваліфікаційної роботи було розроблено інтернет магазин зоотоварів, а саме :

1. Було сформульовано технічне завдання;
2. Спроековано модель бази даних;
3. Спроековано архітектуру програми;
4. Визначено ряд функціоналу на основі сучасних потреб та вимог;
5. Був реалізований магазин зоотоварів що відповідає встановленим вимогам;

Рисунок В.15 – Слайд №15

Дякую за увагу !

Рисунок В.16 – Слайд №16