

**Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка**

**Факультет інформаційних технологій
Кафедра мережевих та інтернет технологій**

ЗАТВЕРДЖУЮ

завідувач кафедри
мережевих та інтернет технологій
_____ **Юрій КРАВЧЕНКО**
« ____ » _____ 2022 року

**КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА**

галузі знань 17 «Електроніка та телекомунікації»
за спеціальністю 172 «Телекомунікації та радіотехніка»
освітньо-професійна програма «Мережеві та інтернет технології»

на тему:

**РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ З ПІДТРИМКОЮ
МІЖНАРОДНИХ ПЛАТЕЖІВ**

_____ **Микита ШЕВЧЕНКО**

(ім'я та ПРИЗВИЩЕ)

_____ (підпис)

Виконав: студент групи МІТ-41

Керівник: доцент кафедри мережевих та інтернет технологій

_____ **к.т.н., доцент Оксана ГЕРАСИМЕНКО**

(науковий ступень, вчене звання, ім'я та ПРИЗВИЩЕ)

_____ (підпис)

Київ-2022

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ

завідувач кафедри

мережевих та інтернет технологій

_____ **Юрій КРАВЧЕНКО**

«_____» _____ 2022 року

Кафедра мережевих та інтернет технологій

**ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ**

Здобувачу вищої освіти _____ Шевченку Микиті Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи:

Розробка інтернет-магазину з підтримкою міжнародних платежів

затверджена на засіданні кафедри МІТ «24» грудня 2021 р. протокол №8

2. Термін здачі закінченої роботи «30» травня 2022 р.

3. Вихідні дані до проекту (роботи)

Технічне завдання замовника

4. Зміст пояснювальної записки (перелік питань, що їх потрібно розробити, обсяг – 35-40 стор.)

Вступ

1. Аналіз сучасних засобів реалізації міжнародних продажів в інтернет-магазині

1.1 Інтернет-продажі як явище сучасної економіки

1.2 Автоматизація процесів продажу через Інтернет

1.3 Аналіз засобів реалізації платформи інтернет-продажів

1.4 Вибір засобів реалізації інтернет-магазину

2. Проектування інтернет-магазину

2.1 Архітектура додатку в Shopify

2.2 Архітектура theme проекту

2.3 Дизайн веб-додатку

3. Розробка інтернет-магазину

3.1 HTML каркас додатку

3.2 Стилзація та адаптивність

3.3 Створення моделей налаштувань секцій та відображення даних

3.4 Програмування компонентів фільтрації, пошуку, сортування

3.5 Програмування продуктової сторінки та кошику

3.6 Процес оплати замовлення

Висновки

Дата видачі завдання _____

Керівник роботи _____

Оксана ГЕРАСИМЕНКО

(підпис)

(посада, прізвище, ім'я, по батькові)

Завдання прийняв до виконання _____

Микита ШЕВЧЕНКО

(підпис)

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ РОБОТИ

Номер	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Підготовчий	01.02.2022	
2	Розділ 1	01.03.2022	
3	Розділ 2	01.04.2022	
4	Розділ 3	15.05.2022	
5	Доповідь та слайди	30.05.2022	
6	Пояснювальна записка	30.05.2022	

Здобувач вищої освіти _____ Микита ШЕВЧЕНКО
(підпис)Керівник _____ Оксана ГЕРАСИМЕНКО
(підпис)

РЕФЕРАТ

Пояснювальна записка: 57 с., 36 рис., 15 джерел.

Об'єкт дослідження: онлайн платформи для міжнародних продажів.

Предмет дослідження: система керування контентом для платформи міжнародних продажів.

Мета роботи (проекту): розробка онлайн платформи для міжнародних продажів із використанням системи керування контентом, яка забезпечує підтримку різних валют розрахунку та обробку клієнтських запитів високої інтенсивності.

Методи дослідження: системний аналіз.

У роботі проведено аналіз сучасного стану платформ для організації міжнародних продажів, визначено їх особливості, переваги та недоліки.

Розроблено та побудовано інтернет-магазин з підтримкою міжнародних платежів.

Результати здійснених у дипломному проекті досліджень можуть бути використані при розробці інших інтернет-магазинів для міжнародних продажів.

Ключові слова: SHOPIFY, CMS, CRM, ІНТЕРНЕТ-ПРОДАЖІ, МУЛЬТИВАЛЮТНІСТЬ, ПЛАТІЖНА СИСТЕМА.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
1 АНАЛІЗ СУЧАСНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ МІЖНАРОДНИХ ПРОДАЖІВ В ІНТЕРНЕТ-МАГАЗИНІ	8
1.1 Інтернет-продаж як явище сучасної економіки	8
1.1.1 Аналіз сучасного стану інтернет-продажів	8
1.1.2 Проблеми організації міжнародних продажів через Інтернет.....	8
1.2 Автоматизація процесів продажу через Інтернет	10
1.2.1 Поняття системи управління інтернет-продажами та її функції.....	10
1.2.2 Складові системи керування інтернет-продажами	13
1.2.3 Порівняльний аналіз сучасних CMS (WCMS)	14
1.3 Аналіз засобів реалізації платформи інтернет-продажів.....	20
1.4 Вибір засобів реалізації інтернет-магазину	21
1.4.1 Вибір CMS.....	21
1.4.2 Вибір засобів реалізації серверної сторони	22
1.4.3 Вибір платіжної платформи	23
1.4.4 Вибір засобів забезпечення захисту конфіденційних даних	24
Висновки до розділу I	24
2 ПРОЕКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ	25
2.1 Архітектура додатку в Shopify	25
2.1.1 Парсинг .liquid файлів та компіляція html	25
2.1.2 Об'єкти Shopify.....	26
2.1.3 Файлова структура Shopify додатків	28
2.2 Архітектура theme проекту.....	30
2.2.1 Мапа інтернет магазину.....	30
2.2.2 Функціонал сторінок.....	31
2.2.3 Проектування theme	31
2.3 Дизайн веб-додатку.....	34
Висновки до розділу II.....	38

3 РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ	40
3.1 HTML каркас додатку	40
3.2 Стилiзацiя та адаптивнiсть.....	42
3.3 Створення моделей налаштувань секцiй та вiдображення даних.....	44
3.4 Програмування компонентiв фiльтрацiї, пошуку, сортування	46
3.5 Програмування продуктової сторiнки та кошику.....	48
3.6 Процес оплати замовлення.....	55
Висновки до роздiлу III	58
ВИСНОВКИ.....	59
ПЕРЕЛІК ПОСИЛАНЬ.....	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CMS – система керування вмістом

CRM – система керування клієнтами

ECM – система керування корпоративним контентом

WCM – система керування веб контентом

WP – WordPress

WC – WooCommerce

JS – JavaScript

SaaS – програмне забезпечення як сервіс

БД – база даних

MVC – архітектура веб-додатку model view controller

SPA – архітектура веб-додатку single page application

API – application programm interface інтерфейс програмного додатку, як правило, серверної сторони

ВСТУП

Перший інтернет-магазин був відкритий у 1992 році, коли Чарльз Стек створив інтернет-магазин книжкової продукції. Під час розвитку глобальної мережі, онлайн продаж зазнав глобальних метаморфоз.

У процесі еволюції цього явища було безліч подій, завдяки яким можна поділити цю еволюцію на деякі етапи. Але в даному контексті, найбільш сильно виділяється подія світової пандемії, яка поділила світ на до і після. COVID-19 створив тренд на дистанціювання як соціальне, так і робоче. Це здобуло свій відбиток у процесах інтернет продажів. Найголовнішою зміною стала потреба дистанційних покупок. Раніше це було просто зручно купити щось в інтернеті. Але після COVID-19 людям довелося купувати онлайн. Через два роки після пандемії, соціальне дистанціювання вже не є обов'язковим заходом. Але через її існування, під час боротьби держав із хворобою, безліч людей або звикли купувати онлайн, або просто зрозуміли, що це дуже зручно, так і продовжили. Все це призвело до великого зростання користувачів онлайн-магазинів. Крім того, розширення бізнесу вимагає створення інтернет-магазинів, доступних із різних куточків світу. У такому разі має бути забезпечена підтримка міжнародних платежів, враховуючи особливості ведення податкової та іншої документації у різних країнах. Ці задачі є надзвичайно актуальними і на даний час, оскільки все більше бізнесу переносить свою торгівлю в Інтернет. Таким чином, тема роботи є важливою і затребуваною на сьогоднішній день та має значні перспективи розвитку у майбутньому.

1 АНАЛІЗ СУЧАСНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ МІЖНАРОДНИХ ПРОДАЖІВ В ІНТЕРНЕТ-МАГАЗИНІ

1.1 Інтернет-продаж як явище сучасної економіки

1.1.1 Аналіз сучасного стану інтернет-продажів

Згідно зі статтею «Пандемія коронавірусу збільшить продажі електронної комерції в США на 219 мільярдів доларів у 2020-2021 роках» від April Berthene [1] COVID-19 прискорив зростання електронної комерції на роки вперед. Інші зміни, спричинені пандемією, також сприяли зростанню продажів електронної комерції, наприклад, інфляція. За новими даними Adobe, за останні два роки споживачі заплатили в Інтернеті на 32 мільярди доларів більше за ту саму кількість товарів.

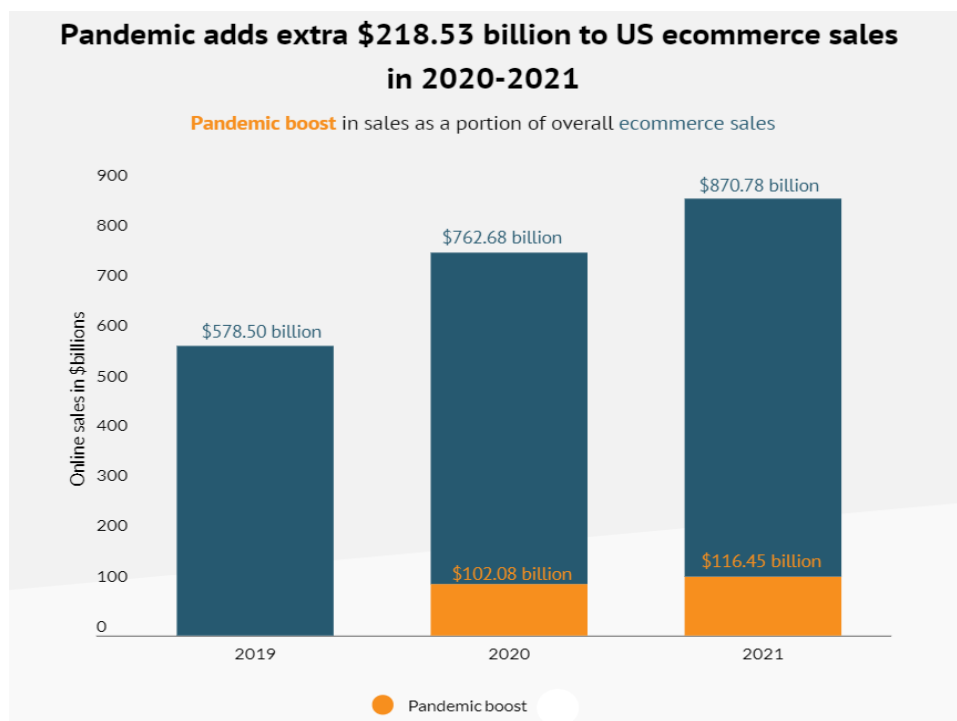


Рисунок 1.1 – Зростання електронної комерції [1]

1.1.2 Проблеми організації міжнародних продажів через Інтернет

Через поштовх до розвитку інтернет продажу, стало зручнішим створення міжнародних торгових майданчиків. Якщо раніше це було привілеєм для великого

бізнесу, то тепер така можливість є і для маленького бізнесу. Але існування можливості ще не означає відсутність проблем при налагодженні міжнародних продажів. Адже щоб налагодити бізнес процеси даної сфери, потрібні фахівці з кожної країни, де планується налагодити продажі. А в кожній країні, і всередині країни, є різне зведення документів про податки, різні формати квитанцій про оплату, і безліч вимог до платіжних систем і підтримка безпосередньо цих систем. Ще треба враховувати технічну інфраструктуру, налагоджувати логістику, підтримку для конкретного ринку, релевантні методи оплат. А ще потрібно розуміти відмінності трендів і потреб різних ринків, тому що в одній країні, якийсь продукт не матиме великого попиту, як в іншій, це теж потрібно враховувати. Для реалізації подібних вимог потрібно мати якийсь штат фахівців, наприклад, департамент логістики, який у свою чергу буде ділитися на штат співробітників, які займаються певним сектором, тобто списком країн, за які вони відповідають. Може бути розбиття складів по секторах продажів, у кожному секторі, тобто списку країн, є один або більше складів, які потрібно адмініструвати та налагоджувати доставки безпосередньо в ці склади, а також доставляти товари зі складу замовникам. Подібна потреба викликає значні складнощі при реалізації міжнародних продажів. Лише логістична проблема може відкласти на тривалий термін можливість масштабування магазину міжнародних продажів будь-якої продукції. Крім логістичної проблеми, має місце індивідуалізація контенту/продукції на продаж у певний сектор. Як приклад можна навести різницю між ринком комп'ютерних ігор європейського та азійського сектора. Через відмінності та особливості культур і переваг цих двох економічних секторів, попит на цифровий контент може сильно відрізнятись. Якщо виробляти цифрову продукцію без урахування вищезгаданих відмінностей, то прибуток може бути значно знижений. Вирішити цю проблему можна створенням департаменту адаптивного контенту, який, у свою чергу, буде ділитися на штати співробітників, які відповідатимуть за різні культурні сектори, щоб цифрова продукція, що випускається, створювалася з урахуванням переваг споживачів. Але це не вирішує повною мірою цю складову списку проблем міжнародних продажів, тому що звідси впливає питання про те, як здійснювати продаж між усіма

регіонами. Звичайно, можна для кожного регіону використовувати свою інформаційну систему продажів. Але відкривати в кожній країні новий філіал і постійно використовувати різні інструменти для продажу - не є ефективним в повній мірі. Набагато вигідніше використовувати одну єдину вичерпну платформу для адміністрування всіх ринкових секторів, з можливістю підлаштовуватися під геолокацію певного користувача, який зайшов на сайт компанії та пропонувати продукцію, спрямовану на його ринковий сектор. Набагато вигідніше використовувати єдину систему для відстеження товарів у наявності. Єдине, навіщо доведеться використовувати окремий інструмент, так це для складських приміщень, тому що там зовсім інші процеси. Найголовніше, щоб головний магазин завжди був у курсі того, в якому обсязі та кількості в нього є та чи інша продукція.

1.2 Автоматизація процесів продажу через Інтернет

1.2.1 Поняття системи управління інтернет-продажами та її функції

Говорячи про інтернет-продаж, завжди мається на увазі інтернет магазин. За стандартом, це є звичайним клієнт-серверним додатком, мінімальний функціонал якого обмежується авторизацією, демонстрацією продукту та його опцій, а також можливістю зробити замовлення та оплатити. Але з розвитком продажів, складно знайти успішний бізнес, який мав би тільки цей мінімальний набір функцій. Якщо взяти до уваги якісь відомі та популярні приклади майданчиків для інтернет продажів (Amazon, eBay; Розетка, відома українцям тощо) та проаналізувати набір функцій цих інформаційних додатків, то стане очевидним, що вони мають неймовірно великий обсяг можливостей. Зараз, якщо підприємець захоче створити свій інтернет магазин, то для підвищення ймовірності успіху йому доведеться додати на свій сайт, наприклад, форму зворотного зв'язку, можливо якісь рекурентні платежі, розстрочки, live-chat, авторизацію за допомогою сторонніх сайтів, наприклад facebook, gmail etc. І цей список може продовжуватися до безкінечності. Реалізувати вручну подібний обсяг функціоналу, тобто винаходити велосипед, просто щоб він був особисто вашого виробництва – це величезні

витрати. Мати свій штат розробників з огляду на послуги вартості програмістів, проджект-менеджерів, тімлідів, архітекторів тощо - коштуватиме мільйони гривень. Звичайний малий і навіть середній бізнес навряд чи має можливість дозволити собі це.

Для задоволення цих потреб бізнесу, без великих витрат, вигадані такі рішення, як CMS. Система керування вмістом (CMS) — це комп'ютерне програмне забезпечення, яке використовується для керування створенням та модифікацією цифрового вмісту (управління вмістом). CMS зазвичай використовується для керування корпоративним вмістом (ECM) та керування веб-контентом (WCM). Якщо CMS - це більше про продукцію, а не клієнтів, то CRM є рішенням для клієнтів. Управління взаємовідносинами з клієнтами (CRM) — це процес, у якому компанія чи інша організація керує своєю взаємодією з клієнтами, зазвичай використовуючи аналіз даних для вивчення великих обсягів інформації.

Сучасні CMS системи мають у собі функції CRM, оскільки управління інтернет-продажами далеко не завжди відбувається без спілкування з клієнтами. CMS система має наступні функції:

- імпорт та експорт даних;
- зберігання, редагування, видалення даних;
- індексація, пошук, отримання даних: ключові слова або автор.;
- управління форматом: допомагає перетворити відскановані паперові документи та застарілі електронні документи на сучасні документи;
- ревізійний контроль: дозволяє оновлювати та редагувати вміст після початкової публікації. контроль версій також відстежує будь-які зміни, які користувачі вносять до файлів;
- управління доступом: розмежовує доступ між адміністраторами та користувачами. також можуть бути різні рівні доступу користувачів, наприклад вір функції;
- видавнича справа: дозволяє користувачам використовувати шаблон або набір шаблонів, схвалених організацією, а також майстрів та інших інструментів для створення та зміни контенту.

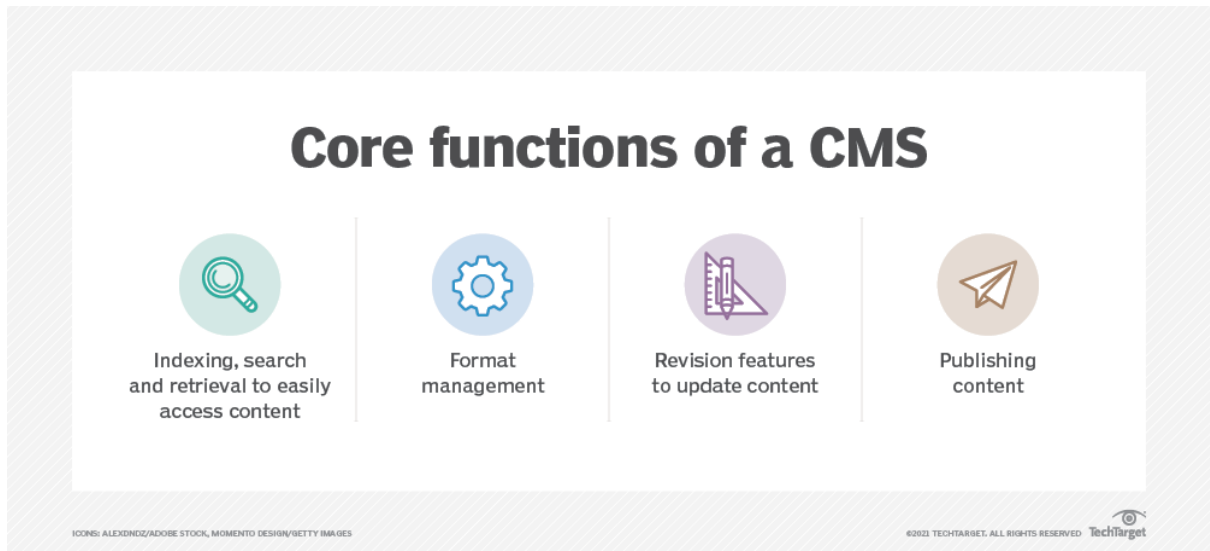


Рисунок 1.2 – Головні функції CMS

Інші популярні функції CMS включають:

- url-адреси, зручні для пошукової оптимізації;
- інтегрована та інтерактивна допомога, включаючи дошки обговорень;
- системи дозволів та безпеку на основі груп;
- повна підтримка шаблонів і налаштовуються шаблони;
- прості процедури встановлення та управління версіями за допомогою майстра;
- панель адміністратора з підтримкою кількох мов;
- мінімальні вимоги до сервера;
- вбудовані файлові менеджери;
- вбудовані журнали аудиту.

Набір цих функцій не є вичерпним, оскільки існує безліч інтернет магазинів, що потребують різної специфіки роботи, що впливає на функціонал.

1.2.2 Складові системи керування інтернет-продажами

CMS – дає можливість її користувачам створювати, видаляти, редагувати контент із веб-сайту без будь-яких навичок програмування та складається з двох частин:

1. CMA – графічний інтерфейс користувача, який дозволяє користувачам створювати, змінювати та видаляти контент із веб-сайту без навичок програмування;
2. CDA – надає серверні служби, які підтримують управління та доставку вмісту веб-сайту після того, як користувач створить його в CMA.

ECM vs. WCM

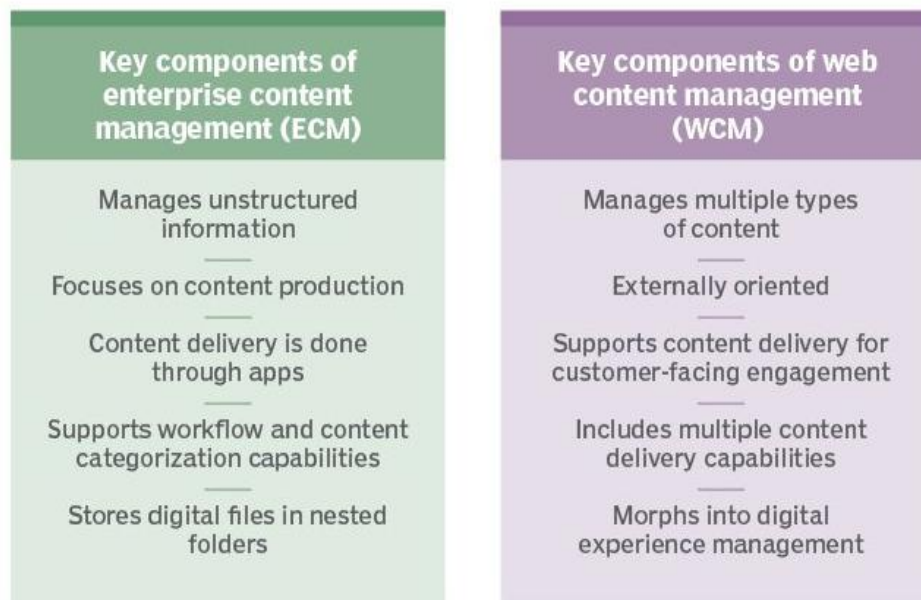


Рисунок 1.3 – Відмінності ECM та WCM

CMS є загальним поняттям, яке поділяється на більш предметні і вузькоспеціалізовані поняття у конкретних сферах – ECM, WCMS (рисунок 1.3). Якщо CMS є середовищем для створення, керування та зберігання цифрового контенту, і найкраще працює зі структурованим контентом, таким як документи та об'єкти бази даних. Тоді ECM є новим поняттям, яке спрямоване на корпоративний контент. Це не про інтернет-продаж. А також ECM може працювати і з

неструктурованим контентом, що включає програмне забезпечення, стратегії та засоби безпеки для ефективного управління контентом. Стосовно веб-контенту, WCMS є спеціальним рішенням для цього. Наприклад, можна визначити одну з функцій WCMS – це управління продуктами та їх сторінками в інтернет магазині, має інструменти публікації та полегшує спільну розробку, наприклад, інтернет магазинів.

1.2.3 Порівняльний аналіз сучасних CMS (WCMS)

При виборі CMS для бізнес-цілей, є сенс звернути увагу насамперед на 6 речей, а саме:

1. простота використання. WCMS має спрощувати створення та редагування контенту, повинен бути присутнім інтерфейс перетягування, щоб можна було змінювати розташування елементів на сторінці. Зміни у контенті інтернет-магазину мають бути зручними, не уповільнювати процес адміністрування магазину, а пришвидшувати;

2. варіативність дизайну. WCMS повинен мати безліч шаблонів дизайну веб-сайтів і їх компонентів, різних секцій тощо. Крім цього, має бути можливість створення цих самих компонентів, для фахівців, які мають достатньо навичок у програмуванні. Користувач повинен мати можливість використовувати та налаштовувати компоненти онлайн-магазину та його складових, без знань у програмуванні;

3. можливість імпорту та експорту даних. WCMS має надавати можливість імпорту та експорту даних. наприклад, якщо користувач вирішив мігрувати на обрану WCMS, йому доведеться імпортувати дані з його бази даних;

4. різні доповнення, сторонні додатки, розширення. Жодна CMS, WCMS, ECM система спочатку не містить вичерпний набір функцій для задоволення потреб усіх своїх клієнтів, оскільки існують інтернет магазини з абсолютно різною специфікою і дизайном. Так, щоб закрити умовну нестачу функціоналу, повинна бути можливість

створювати розширення, для обраної платформи. для цього, наприклад, платформа може надавати, розробникам і документацію з інструментів, які можуть використовувати для створення додатків;

5. наявність та зручність вивчення документації, а також підтримка. Незважаючи на мету CMS у спрощенні створення та адміністрування продуктів на її базі, у користувачів все одно виникатимуть питання про деталі реалізації чогось, у рамках CSM. Як відповідь на запитання користувачам, має бути документація, гайди та мануали з використання продукту. але часом цього не завжди вистачає, і на цей випадок має бути підтримка;

6. ціна. Цінова політика у кожній компанії, що має CMS – своя. є як і безкоштовні рішення, так і платні, із щомісячною підпискою. Як правило, безкоштовні рішення мають обмежений функціонал, і щоб користувач задовольнив усі потреби свого інтернет-магазину, доводиться використовувати додаткові розширення, які коштують грошей. цінове питання дуже індивідуальне для кожного бізнесу, все залежить від бюджету;

7. SEO – оптимізація цього процесу впливає на індексування сторінок пошуковими роботами. Впливає на місце у списку сторінок, що видаються за пошуковим запитом, наприклад, гуглом.

Відповідно до статті [12], у 2022 році найбільш актуальними платформами для електронної комерції є 15 варіантів: WordPress.org, WooCommerce, Shopify, BigCommerce, Drupal, HubSpot CMS hub, Joomla, Wix, WordPress.com, Ghost, Magento, Textpattern, Vitrix24, ТУРОЗ. В цілому, кожна з цих платформ можна використовувати для задоволення потреб будь-якого бізнесу електронної комерції. Єдиний пункт, який можна оцінити об'єктивно – це ціна, яку доведеться заплатити для початку свого бізнесу, чи масштабування існуючого, чи міграції існуючого. Такі пункти, як зручність та простота використання є суб'єктивними. Варіативність дизайну має кожна платформа, яка потрапила до топ платформ на 2022 рік. SEO оптимізація – невід'ємний пункт кожної з платформ, що входять до топу всього світу. Імпорт та експорт притаманні кожній з цих платформ. Різні розширення так само є в кожній із платформ.

Але кожна з платформ має свої особливості, які потрібно враховувати індивідуально для кожного бізнесу, так що на порівняльний аналіз буде взято перші 4 платформи, як найбільш популярні.

WordPress (Далі WP) – найпопулярніша у світі CMS та платформа для електронної комерції. Відповідно до цього сайту, на ньому працює 43% всіх веб-сайтів в Інтернеті. Його популярність обумовлена надзвичайною гнучкістю. Для новачків, малих бізнесів, звичайних лендингів та іншого є можливість швидко і легко створити сайт без навичок, використовуючи компоненти, які можна перетягувати на будь-яку частину сайту. Звичайно такий варіант має безліч обмежень і доводиться діяти в їх рамках, немає повної свободи дій. Але також є можливість створювати різні компоненти з будь-яким дизайном і майже будь-яким функціоналом під свої потреби, але це вже вимагає глибоких знань не тільки в програмуванні, але і вузькоспеціалізованих знань самої платформи.

Плюси:

- гнучкість;
- не потребує технічних навичок;
- зручна SEO-оптимізація;
- величезне ком'юніті;
- широкі можливості для розширення.

Мінуси:

- надмірна гнучкість, перенасиченість;
- XML формат;
- потрібно налаштовувати хостинг та доменне ім'я вручну, самому користувачеві;
- фокус клієнто-орієнтованості на користувачів без технічних навичок;
- не вузькоспеціалізований під одні лише платформи для інтернет продажів, на ньому можна робити що завгодно (є мінусом лише для тих, кому потрібен упор на продажі);
- багато зайвого у контексті створення онлайн магазину.

Особливості:

- завантаження контенту у форматі XML;
- орієнтований більше на користувачів без технічних навичок;
- основна частина функціоналу реалізується за допомогою плагінів;
- за безпеку та резервне копіювання відповідає користувач;
- потрібно використовувати сторонній хостинг та доменне ім'я.

Ціна:

- доменне ім'я (9-15 \$ / рік);
- хостинг (від 8 \$ / рік).

Підсумок: дешевий, переповнений функціоналом двигун, орієнтований більше на не технічних користувачів, не вузькоспеціалізований, чим, власне, і займає основну частку ринку.

WooCommerce (далі WC) - ще одна з найпопулярніших платформ електронної комерції, яка технічно не є платформою CMS, а лише плагіном для WP. Займає 5.8% ринку CMS. В цілому факт того, що це є плагіном для WP, означає, що для цієї платформи справедливі пункти аналізу, що і для WP. Але все одно варто виділити характеристики, безпосередньо для цього плагіна.

Плюси:

- є безкоштовний план;
- є безліч тем для сайтів;
- працює з будь-якою темою WP;
- плюси, які стосуються WP.

Мінуси:

- велика варіативність може стати відштовхуючою для не спеціалістів у цьому плагіні;

- потрібен WP;
- не самостійна платформа;
- мінуси, що стосуються WP.

Особливості:

- є плагіном, а не повноцінною платформою;
- потрібен WP;

- швидше за все, доведеться розширювати базовий функціонал за рахунок платних плагінів.

Підсумок: не повноцінна платформа, для роботи якої потрібен WP, є безкоштовною, але задовольнити потреби не новачків зможе лише за додаткову плату. Доведеться купувати хостинг та доменне ім'я, вся безпека та резервне копіювання припадає на користувачів.

Shopify – універсальна CMS платформа. Під універсальністю мається на увазі відсутність обмежень щодо типу товарів для продажу. Підтримує продаж у фізичному магазині, зручно, коли є фізичні та інтернет магазини, не потрібно використовувати платформи для обліку продукції у фізичному магазині та вручну співвідносити це з продукцією онлайн магазину. Одна з найзручніших платформ для міжнародних продажів. Використовує власну платіжну систему. Є хмарним рішенням (SaaS).

Плюси:

- синхронізація з фізичними магазинами;
- не потрібен хостинг;
- користувачеві не потрібно дбати про безпеку та резервне копіювання;
- орієнтована більше на технічних користувачів;
- зручність у міжнародних продажах;
- власне платіжне рішення Shopify Payments для кредитних чи дебетових карток;
- зручне створення тем для магазину, як і документація з нюансів розробки;
- не потрібно підвищувати рівень платіжного тарифу при досягненні певної суми заробітку, як, наприклад, у Bigcommerce;
- підтримка на зв'язку 24/7;
- однаково зручний для бізнесу будь-якого розміру.

Мінуси:

- відносно висока вартість платіжного плану для великого бізнесу (2000 \$ Shopify +);

- витрати можуть виявитися значними, якщо знадобиться безліч функцій, які надають додатки;

- щомісячна плата за використання;

- комісія до 2%, якщо не використовувати платіжний шлюз Shopify Payments.

Особливості:

- своя платіжна система;

- мало безкоштовних тем;

- немає доступу до robots.txt;

- обмеження функціональності можна обійти хитрощами у розробці, якщо немає бажання платити за готові рішення у Shopify Markets;

- можливість безкоштовного використання при створенні магазину розробником, пізніше його можна буде передати замовнику або перевести його у стан продажу.

Ціна: є три платіжні плани та Shopify +. Від 29 \$ / міс до 299 \$ Шопіфай + - 2000 \$ / міс. Якщо оплачувати за рік, то є знижка. Підсумок: перспективна платформа для електронної комерції зі своєю платіжною системою та відсутністю безкоштовного тарифного плану, з великою зручністю синхронізації фізичного та онлайн інтернет магазину.

BigCommerce (далі BC) – теж універсальне рішення для продажу, як і Shopify і є SaaS, тобто – програмне забезпечення, як послуга, але надає користувачеві базу даних та хостинг. Спочатку було створено як корпоративну CMS для великого бізнесу.

Плюси:

- пробний план;

- безкоштовне доменне ім'я з приставкою ПС;

- можливість використання разом із WP;

- розбиття клієнтів на групи та сегменти.

Мінуси:

- підвищення рівня тарифного плану, при досягненні значного рівня заробітку;

- річний поріг продажів;

- відсутність рекурентних платежів;
- відсутність мультимовних функцій та функцій мульти валют;
- мало сучасних тем для сайту, як у, наприклад, Shopify.

Особливості:

- не вдасться створити магазин із функціями підписки;
- не вдасться створити мультимовний магазин;
- не гнучкий у контексті дизайну.

Ціна: від 29,95 \$ / місяць до 300 \$, чим більше заробляє магазин, тим більше доводиться платити щомісяця.

Підсумки: початкова орієнтованість клієнта на корпоративний сегмент призвела до відсутності гнучкості дизайну. Навряд це хороший вибір для створення, наприклад малого/середнього міжнародного бізнесу.

1.3 Аналіз засобів реалізації платформи інтернет-продажів

Інтернет магазин являє собою клієнт-серверний додаток з будь-якою архітектурою, яка відповідає вимогам бізнесу. Воно може вміщати різні шари додатків, частина його може бути мікросервісною тощо. Але базова основа така, що є сервер, є клієнт, є база даних. База даних (далі БД) взаємодіє лише із сервером, сервер із БД і клієнтом, клієнт - лише із сервером. Крім цього, подібні веб-програми можуть мати на своїх сторінках будь-які iframe, наприклад, якщо використовувати live-chat plugin від Facebook. Можливості створення таких додатків обмежені лише бюджетом. І, зазвичай, розробники не мають технічних обмежень. По суті, даний варіант вимагає реалізації від одного full-stack інженера до, необмеженої розміром, команди, як наприклад в Amazon. Це залежить лише від потреб бізнесу - наскільки велику і функціональну програму потрібно розробити і в які терміни. Очевидно, що для створення інтернет-магазину, з нуля, для малого бізнесу, з адмін панеллю і можливість адміністрування контенту користувачем, одним розробником, знадобиться більше часу, ніж якщо це створюватиме команда розробників. Наприклад - створення інтернет-магазину квітів для середньостатистичного

підприємця в Україні, одним розробником, може зайняти до 4-5 місяців розробки. Середня зарплата middle full-stack інженера в Україні – від 2500\$ до 3600\$/місяць, залежно від стеку технологій.

Крім створення подібного продукту "своїми силами", можна вдатися до використання платформ електронної комерції. Вони задовольняють усі потреби будь-якого інтернет-магазину. Єдиний їхній мінус - обмеження в розробці. Кожна платформа диктує свої правила та специфікації з розробки інтернет-магазинів, це може створити деякі незручності та змусить шукати обхідні шляхи та лавірувати серед технічних обмежень обраної платформи. Але цей мінус перебиває те, що навіть підприємець, з наведеного вище прикладу про квітковий магазин, може з мінімальними вкладеннями створити свій власний інтернет магазин, навіть без навичок програмування. Хоча, все ж таки, для хоча б трохи серйозних проектів, без фахівців, все ж таки, не обійтися.

1.4 Вибір засобів реалізації інтернет-магазину

1.4.1 Вибір CMS

З метою економії часу та відсутності достатнього бюджету на команду розробників, при виборі чи робити магазин “власними силами” або ж на платформі для електронної комерції, перевага надається другому варіанту, згідно з пунктом 1.3 про аналіз засобів реалізації платформи інтернет-продажів.

Як бізнес, під який розроблятиметься ПЗ у даному дипломному проекті, буде невеликий інтернет магазин ювелірних виробів. Потрібна можливість адміністрування асортименту, відстеження кількості виробів, продаж виробів у будь-який куточок світу, де доступні запропоновані способи доставки. Оплата має бути зручною та швидкою. Очевидно, що потрібна можливість зручного управління продуктами та пакетами продуктів.

Після ретельного порівняльного аналізу пункту 1.2.3, як CMS і платформи для електронної комерції було обрано рішення Shopify. Його цінова політика щодо найдешевшого тарифного плану задовольняє потреби початкового бюджету для

старту і не є дорогою. Так само він дозволяє зручно адмініструвати асортимент продуктів і розробка своїх власних шаблонів (тем), під платформу, є зручною та має зрозумілу документацію. Крім цього, власне платіжне рішення Shopify не має обмежень для оплат у різних країнах, а також є підтримка багатьох інших платіжних рішень.

1.4.2 Вибір засобів реалізації серверної сторони

Вибір не робити все “своїми руками” обумовлений відсутністю достатньої кількості ресурсів, функціонал платформи задовольняє потреби розробки. Звичайно це не забороняє створити десь окремий сервер, написати API і використовувати його як додаток, наприклад, якщо якихось функцій немає. Але як правило – це рідкість. Для того, щоб виникла така необхідність, то специфіка і вузька спрямованість бізнесу, повинна бути дуже рідкісною, щоб платформа не подбала про створення такого функціоналу. Більш того, крім стандартного набору функцій ще існують додаткові програми, які додадуть ще більшу гнучкість у використанні. Таких додатків тисячі та десятки тисяч. Так що писати самотужки серверну сторону просто немає необхідності.

Платформа Shopify написана на Ruby on Rails (далі RoR), якщо раніше цей фреймворк мав лише трохи переваг, коли він тільки розвивався, то зараз, це саме те, що відмінно підходить для створення е-commerce платформ. При його використанні, час розробки дуже скорочується, а ще він дозволяє проводити надшвидке розгортання. Якби Shopify запускався в наші дні, можливо, він вибрав би інший стек технологій, але в той час RoR був дуже популярний. Вибір був зроблений з упором на гнучку і просту масштабованість, адже без цього платформа не змогла б вирости через технічні обмеження. А у RoR архітектура, якраз таки, дозволяє це робити, тому що там використовується абсолютна модульність і він загалом підходив для принципів дванадцяти факторного застосування. The Twelve-Factor App - методологія для створення SaaS додатків. Її принципи: 1. декларативний формат для опису процесу встановлення та налаштування; 2. зручна міграція між середовищами

виконання; 3. зручність у розгортанні на сучасних хмарних платформах; 4. безперервне розгортання; 5. масштабуватись. Крім цього, Shopify має архітектуру MVC, як OPM використовує Rails ORM.

1.4.3 Вибір платіжної платформи

Shopify має багато способів використання платежів, починаючи від свого рішення Shopify Payments, до 100+ інтеграцій з різними операторами платіжних систем. Крім цього набору варіантів, можна також підключити до свого магазину ручний спосіб оплати, сторонню платіжну систему або навіть створений розробниками платіжний шлюз якого-небудь банку. Але якщо використовувати сторонні рішення, то в адмін-панелі не буде видно інформацію про платежі. Так що в даному проекті будуть використовуватися стандартні рішення Shopify, з метою зручності адміністрування. Якщо використовувати Shopify Payments, то оплата буде здійснюватись на сторінці каси (чекаут) без перенаправлення на сторонні ресурси. А якщо використовувати наприклад PayPal або Google Pay, тоді користувач буде перенаправлений на сторінку оплати цих сервісів. Але перенаправлення відбувається лише у разі використання глобальних платіжних операторів, які підтримують Shopify:

- Apple Pay;
- Amazon Pay;
- Facebook Pay;
- Google Pay;
- PayPal;
- Shop Pay;
- Shop Pay Installments.

Ще є можливість використання сторонніх агрегаторів платіжних систем. У шопіфай є понад 100 налагоджених інтеграцій, у тому числі навіть з використанням

криптовалютних платіжних систем. Під час розробки та тестування даного дипломного проекту, як платіжна система буде використовуватися Bogus Gateway.

1.4.4 Вибір засобів забезпечення захисту конфіденційних даних

Shopify не надає доступу до своєї серверної сторони, з очевидних причин. Він бере функції захисту конфіденційності даних на себе.

Висновки до розділу I

Проаналізувавши які є способи реалізації платформи для інтернет-продажів, було прийнято рішення – використовувати платформу для електронної комерції, оскільки це швидко, зручно, дешево і не має якихось значних мінусів. В якості такої платформи було обрано рішення Shopify, тому що його функціонал та цінова політика задовольняє вимоги бізнес-проекту, який полягає у невеликому онлайн-магазині для продажу ювелірних виробів з можливістю продажу та покупок у різних країнах. А як основний платіжний шлюз використовуватиметься Bogus Gateway, оскільки магазин працюватиме в тестовому режимі для розробки. В подальшому, щоб, наприклад, продати магазин або запустити його в режим продакшну, можна буде експортувати тему в інший магазин, який буде створено для продажу, і дані, якщо вони потрібні, також можна буде експортувати і перенести на інший онлайн-магазин у рамках Shopify.

Технологічний стек обумовлений специфікою роботи з обраною CMS платформою – Liquid, html5, css + scss, JS, jQuery.

2 ПРОЕКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ

2.1 Архітектура додатку в Shopify

2.1.1 Парсинг .liquid файлів та компіляція html

Створення магазинів на базі Shopify передбачає тісну роботу з об'єктами цієї платформи, що потрібні для отримання даних від серверу. Для цього використовується технологія шаблонізації Liquid. Вона дозволяє відобразити дані в HTML шаблоні. На практиці - це набір додаткових тегів з, дещо, відмінним синтаксисом від базового HTML.

```
1  {{ product.title }} <!-- Output: Awesome T-Shirt-->
```

Рисунок 2.1 – Приклад використання Liquid

На рисунку 2.1 показаний приклад звернення до об'єкта product та його параметру title, тобто імені, що повертає результат “Awesome T-Shirt”. Такий код вставляється в будь яке місце html розмітки, але файли, що містять в собі такий код мають розширення .liquid. Працює це наступним чином:

1. клієнт робить запит на сервер Shopify, наприклад get index.html;
2. сервер отримує запит, знаходить аналог цього файлу, тільки із розширенням .liquid;

2.1 аналог цього файлу може містити в собі безліч інших файлів з розширенням .liquid. Тож сервер парсить головний, та вкладені файли, якщо бачить liquid теги, то обробляє їх. Наприклад щодо рисунку 2.1, коли liquid обробник побачить вусатий синтаксис, а в ньому зарезервований об'єкт product, та спробу отримати параметр title, тож обробник ліквід коду зробить запит на інший сервер, що займається обробкою запитів від liquid обробника, та поверне обробнику дані з сервера product.title;

2.2 після того, як обробник отримав дані, він замінив код liquid на отриманні дані. Таким чином відбувається парсинг всіх тегів та отримання й обробка даних;

2.3 виконується компіляція index.html файлу з підставленими даними й відправляється у відповідь на запит. Так працює шаблонізація.

Оскільки Shopify був створений в часи, коли MVC архітектура була найбільш популярною (зараз популярною є SPA), то цей підхід не мав більш вигідних аналогів. Якщо б Shopify розробляли зараз із використанням технології SPA, то більш зручніше було б не використовувати шаблонізатор, а користуватися звичайним API задля надання даних клієнту.

Звісно, зараз Shopify також має API, але якщо розмовляти лише про створення тем для магазину, то це API має функціонал лише для налагодження двигуна продуктової сторінки та корзини. Оскільки ми можемо отримувати дані, використовуючи шаблони, лише при оновленні сторінки (так працює MVC), то ми не можемо, наприклад, додати товар у корзину, без оновлення, тож потрібно використовувати API та створювати двигун, написаний на JavaScript.

2.1.2 Об'єкти Shopify

Shopify надає велику кількість об'єктів, для отримання даних клієнтом, щоб відобразити їх, та, якимось, працювати із ними через JavaScript.

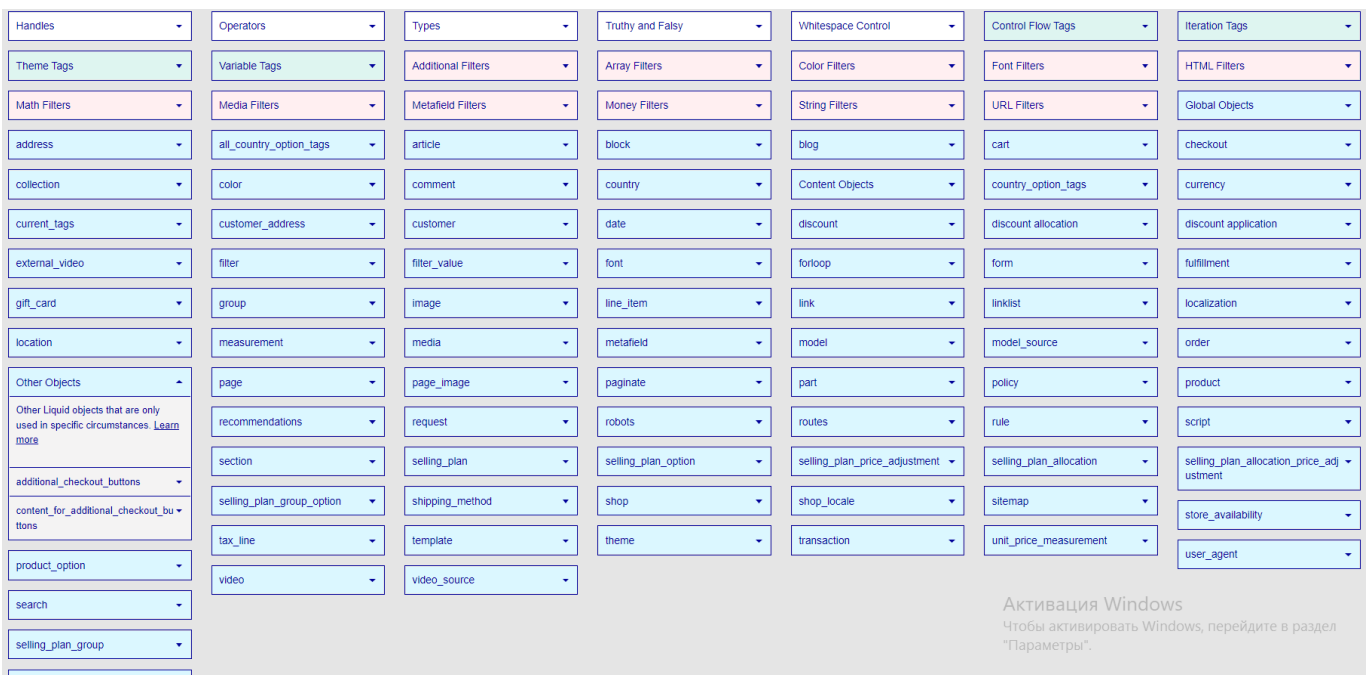


Рисунок 2.2 – Об'єкти Shopify.

Деякі об'єкти розбиті на декілька інших, аналогічно до процесу нормалізації таблиць в базах даних. Наприклад, коли відбувається робота з продуктами, то інтуїтивно, ми маємо глобальний масив продуктів, де кожен об'єкт, що міститься в такому масиві є об'єктом `product`, та коли відбувається додавання продуктів до

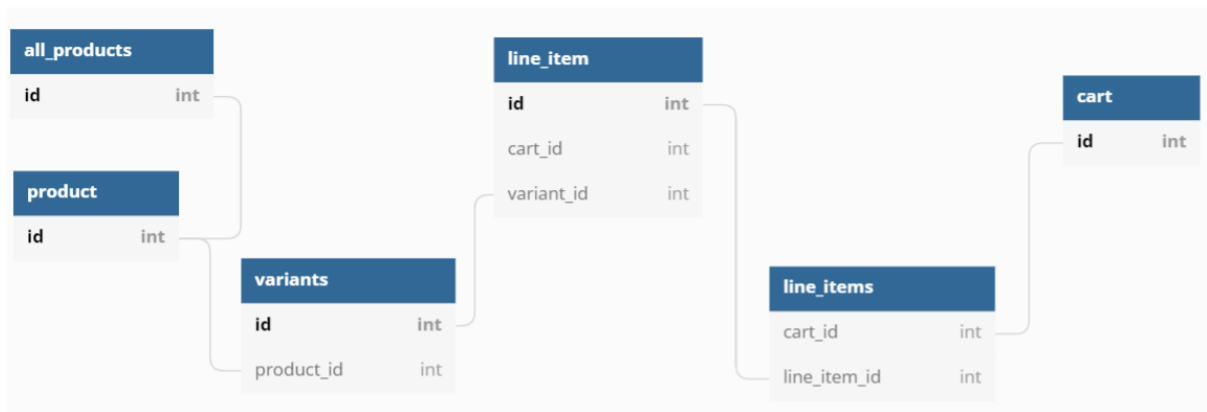


Рисунок 2.3 – Модель продукту

ко

рзин

и, то це відбувається із об'єктом типу `product`. Але це не зовсім так. Якщо оперувати таблицями з бази даних, тоді архітектура моделі продуктів буде як показано на рисунку 2.3.

Як можна побачити на рисунку 2.3, в кошику магазину не зберігаються продукти напряму, а там зберігається інший тип об'єкта, оскільки сам продукт, може мати декілька варіантів, що складаються на основі опцій. Наприклад є футболка із розмірами M, S, L і є два кольори - чорний та білий. Коли користувач робить свій вибір, він обирає комбінацію цих опцій, наприклад футболку розміра M білого кольору. Кількість цих варіантів буде добутком опцій, в даному випадку 6. Але при додаванні в корзину такого варіанту, потрібно ще враховувати кількість доданого варіанту, ціну після застосування знижки, та інші поля, які нелогічно застосовувати для об'єкта варіанту. Це здобуває свій відбиток в використуванні Shopify API, до кошику можна додати продукт, тільки використовуючи `id` варіанта, а не продукта, це різні `id`.

В Shopify є глобальні та локальні об'єкти. Звертання до глобальних об'єктів можливе з будь-якої сторінки сайту, та будь якого файлу liquid, будь то файл section чи snippet файл. Для локальних об'єктів потрібен контекст. Наприклад, якщо ми знаходимось на домашній сторінці і хочемо відобразити якийсь продукт. Ми можемо взяти цей продукт або з налаштувань секції: `section.settings.some_product`; або звернутися до глобального об'єкту, масиву продуктів, и по хендлу (унікальне ім'я продукта, без пробілів, маленькими літерами) отримати потрібний нам продукт: `all_products['awesome-t-shirt']`.

In the following example, `quantity` is the amount of the variant that you want to add and `id` is the variant ID of that variant. You can add multiple variants to the cart by appending more objects in the `items` array.

```

1 items: [
2   {
3     id: 36110175633573,
4     quantity: 2
5   }
6 ]

```

Рисунок 2.4 – Документація Shopify щодо додавання продуктів у кошик

2.1.3 Файлова структура Shopify додатків

Зовнішній вигляд магазину, його функціонал, робота з даними тощо – це описується словом тема (від англ. – Theme). Тема поділяється на три складові каталогів файлів:

1. Розмітка та функції - ця частина відповідає за шаблон сайту, його функціонал. Файли цієї частини використовують шаблонізатор для компіляції HTML.
 - a. (1) – `theme.liquid` є головним файлом шаблону, містить в собі `head`, `body`. Тут підключаються скрипти, файли стилей та інші ресурси, що потрібні відображатися на кожній сторінці сайту, наприклад хедер чи футер.
 - b. (2) – Контент сторінки, використовує шаблони json. Це може бути шаблон кошика, сторінки продукту, сторінки колекцій, блог,

пошук, обліковий запис користувача, сторінка логіну чи реєстрації, або будь-яка інша сторінка.

- c. (3) – Кожна сторінка веб-додатку на Shopify розбивається на секції, буквально – це `section.html` тег. Ці секції можна перевикористовувати. Кожна секція містить в собі, окрім розмітки, можливих скриптів чи файлів стилей, `schema` тег, де описуються налаштування схеми та описуються моделі даних, для того, щоб вносити та відображати дані, які власник магазину вводить через админ панель.
- d. (4) – Блоки, які буквально називаються `block`, описуються в `schema` тег секції, та відображаються в середині секції, де вони описані. Одна секція може містити в собі до 16 блоків.

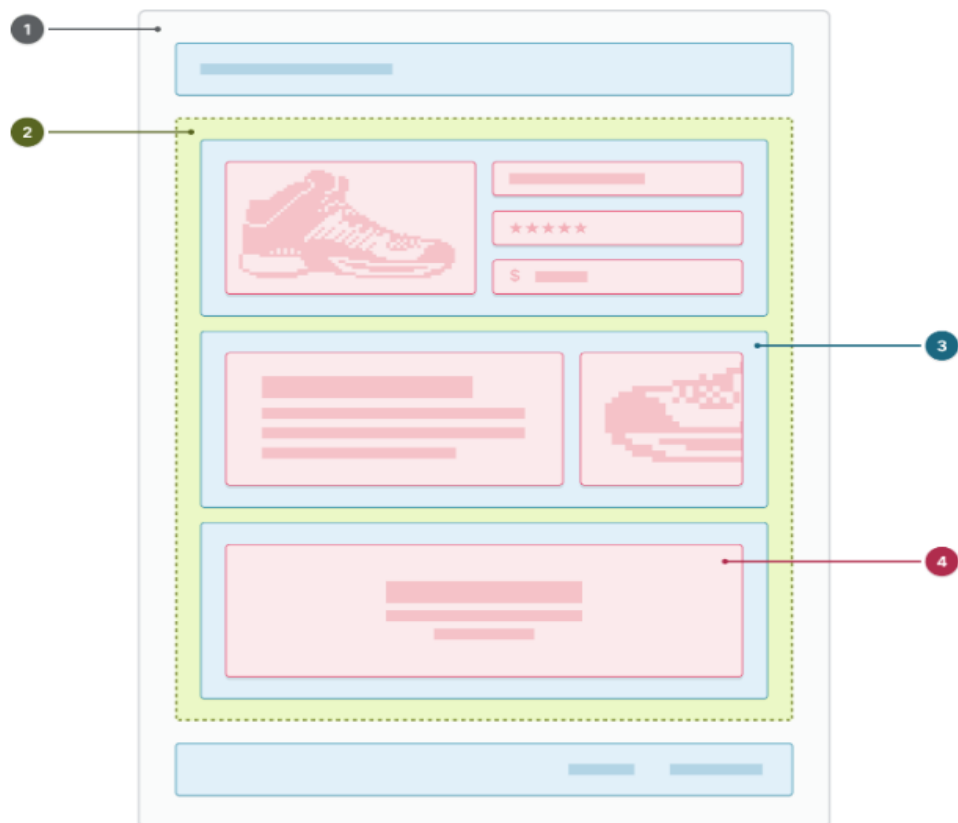


Рисунок 2.5 – Розмітка та функції

2. Допоміжні ресурси – всі файли, що використовуються іншими. Це такі файли, як: скрипти, файли стилей, медіа, шрифти, векторна графіка

тощо. Сюди входить 3 каталоги (папки) в якій зберігаються файли: assets зберігає js, (s)css, fonts, media; snippets - .liquid файли, що можна перевикористовувати будь де на сайті; папка locales, де зберігаються файли для перекладу на різні мови, використовується розширення .json.

3. Конфіги – ці файли потрібні для налаштувань магазину, потрібні, наприклад, для змінювання даних в розділі кастомізації (Customize) магазину. *settings_data.json* - зберігає дані, що власник магазину вніс у customize. *settings_schema.json* - описує ці дані.

Така архітектури підтримується чіткому слідуванні структури папок.

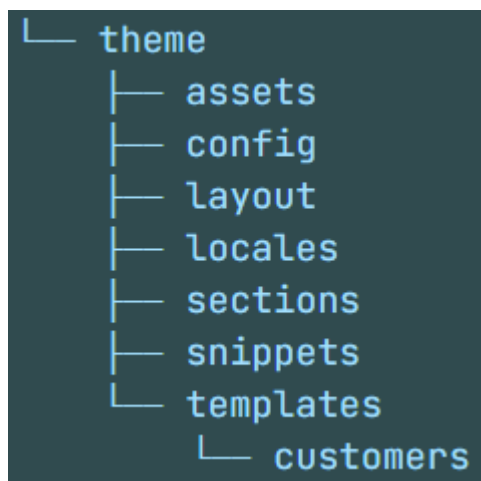


Рисунок 2.6 – Структура папок в Shopify проекті

2.2 Архітектура theme проекту

2.2.1 Мапа інтернет магазину

Перед створенням дизайну інтернет магазину потрібно визначитися зі сторінками та їх функціоналом.

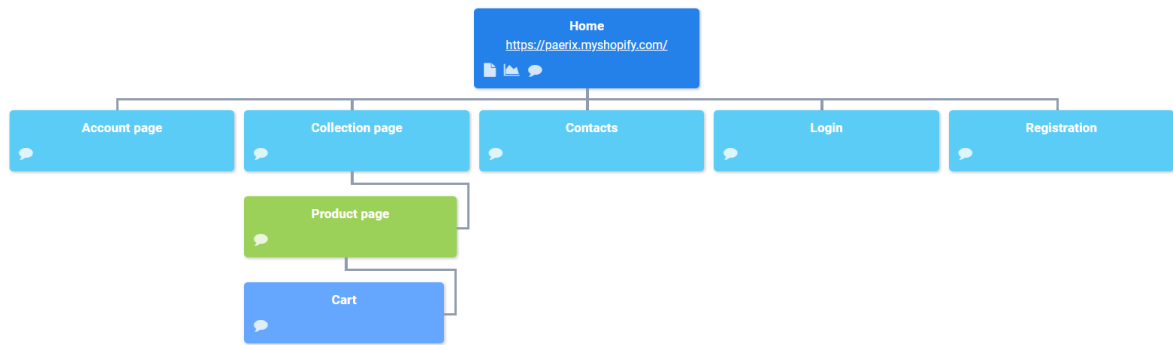


Рисунок 2.7 – Мапа інтернет магазину

2.2.2 Функціонал сторінок

Головна сторінка демонструє продукцію магазину та бренд. Сторінка колекцій відображає весь товар магазину, має різні фільтри для сортування та пошуку продуктів, й також пошук. Сторінка контактів дозволяє користувачам зв'язатися з власником. Сторінки логіну та реєстрації, відповідно, надають змогу користувачам зареєструватися та автентифікуватися на сайті, після чого, в них з'явиться можливість зайти у власний кабінет. Із сторінки колекцій користувач переходить на сторінку обраного продукту, щоб дізнатися більше про нього та додати його до кошику. На сторінці кошику він буде регулювати кількість товарів, видаляти, якщо потрібну, та далі він може бути перенаправлений на сторінку оплати.

2.2.3 Проектування theme

Хедер, тобто шапка всього сайту, що буде показана на кожній сторінці буде містити в собі логотип, навігацію по колекціям та сторінку “Про нас”. Також хедер повинен показувати іконку кошика та кількість товарів в ньому, й, відповідно, надавати змогу користувачу перейти у свій кошик. Над основною частиною повинний буде висвічуватися панель оголошень, наприклад, щоб сповістити користувачів про знижки - це допоможе підняти конверсію.

Футер, тобто підвал, також є спільним елементом до кожної сторінки інтернет-магазину. Повинен містити в собі логотип, навігацію по сторінкам й

контактну форму з функцією підписки на оновлення, посилання на соціальні мережі, а у самому низу - перелік підтримуваних платіжних систем.

Головна сторінка повинна поділятися на такі секції із наступними налаштуваннями для зміни даних через Customize в адмін панелі:

a) Відео банер із заголовком та кнопкою-посиланням: URL відео, зміна заголовку, зміна тексту посилання, зміна URL посилання.

b) Секція що відображає дві флагманські колекції виробів. Для кожної з двох колекцій повинні бути такі налаштування:

- Відображення назви колекції, але якщо не використано якийсь інший текст.
- Якщо не обрана картинка, то за замовчуванням буде відображена картинка колекції.
- Якщо кастомний опис відсутній, то буде відображено опис колекцій, якщо такий задан в адмін панелі.
- Кнопка-посилання за замовчуванням вестиме на сторінку колекції із фільтром цієї колекції, якщо не задан інший URL.
- Текст кнопки-посилання за замовчуванням - “customize”, його можна змінити.

c) Секція, що неявно призиває користувачів до покупки. Матиме такі налаштування:

- Заголовок змінний.
- Підзаголовок змінний.
- 6 текстових елементів, в якості параметрів налаштування секції, а не в якості блоків.
- Головна картинка по центру - змінна.

Налаштування наступних секцій будуть зрозумілі у ході розробки цих секцій:

- Секція-слайдер з відгуками.
- Секція з бенефітами (корисними елементами даного бізнесу).
- Слайдер з продуктами, що, нібито, найбільш продаються.

- Секція з постами із інстаграму.

Продуктова сторінка повинна поділятися на такі секції:

- “хлібні крихти”.
- Екран медіа продукту та інформаційна форма зо всіма опціями та кнопками покупки.
- Додаткова інформація у вигляді акордеону.
- Елементи, що дозволять користувачу додатково обрати ще якийсь продукт.
- Секція з відгуками.
- Слайдер з додатковими продуктами, щоб підвищити конверсію.

Сторінка колекцій - це сторінка де повинна відображатися продукція магазину з можливістю фільтрації, пошуку та сортування продукції, матиме такі елементи:

- “хлібні крихти”
- Заголовок.
- Підзаголовок.
- Кількість продуктів.
- Фільтраційна панель. Фільтрація по колекціям, пошук, сортування за відгуками, назвою, датою створення колекції, ціною.
- Екран товарів.

Сторінка “про нас” матиме наступні елементи: заголовок, підзаголовок, медіа банер, цитата засновника, пости з інстаграму.

Сторінка питань, тобто “FAQ” буде працювати наступним чином:

- Назва питання буде заголовком акордеону.
- Відповідь на запитання буде розкриватися по кліку на заголовок.
- Потрібна можливість вставки таблиці та медіа.

Сторінка контактів матиме лише форму зворотного зв'язку. А сторінки аккаунту, логіна та реєстрації матимуть мінімальний, стандартний функціонал. Сторінка кошику повинна показувати продукти, що добавленні із можливістю видаляти їх з кошику та змінювати кількість продукту, показувати загальну ціну товарів, ціну доставки та підсумкову ціну з подальшою можливістю

перенаправлення на сторінку оплати. Також в кошику потрібно демонструвати логотипи підтримуваних платіжних систем.

2.3 Дизайн веб-додатку

Згідно з вимогами до дизайну та спроектованому theme проекту, було розроблено дизайн наступних сторінок з описаними елементами. Дизайн додатку розроблений в Figma. Нижче показано дизайн основних сторінок інтернет-магазину рисунки 2.8 – 2.11.

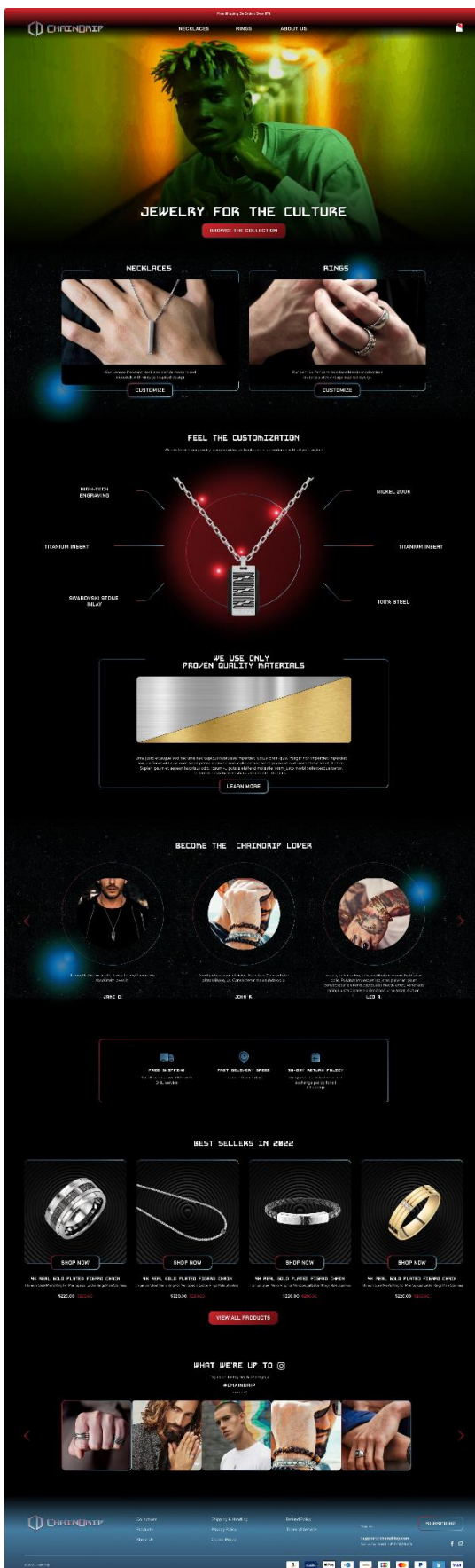


Рисунок 2.8 – Дизайн головної сторінки з хедером та футером

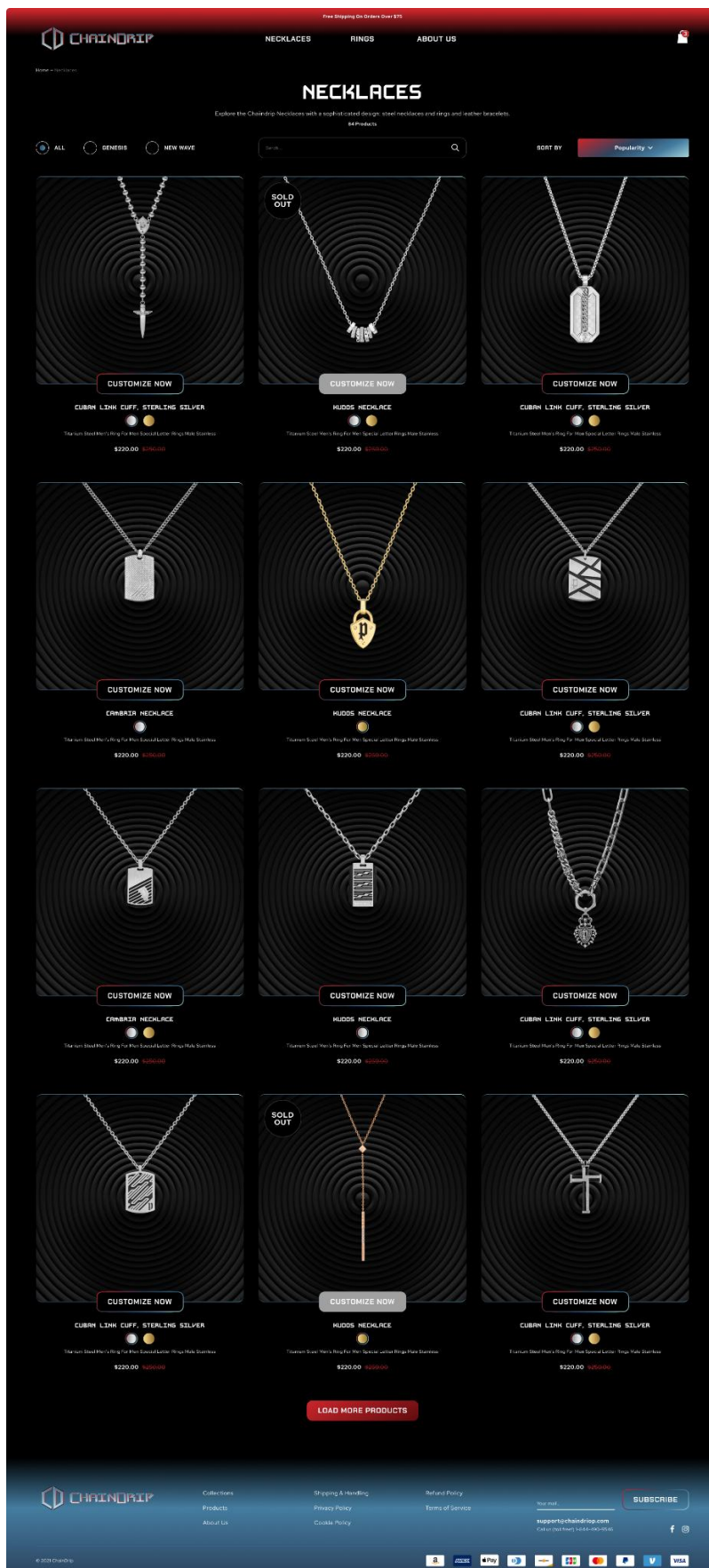


Рисунок 2.10 – Дизайн сторінки колекцій

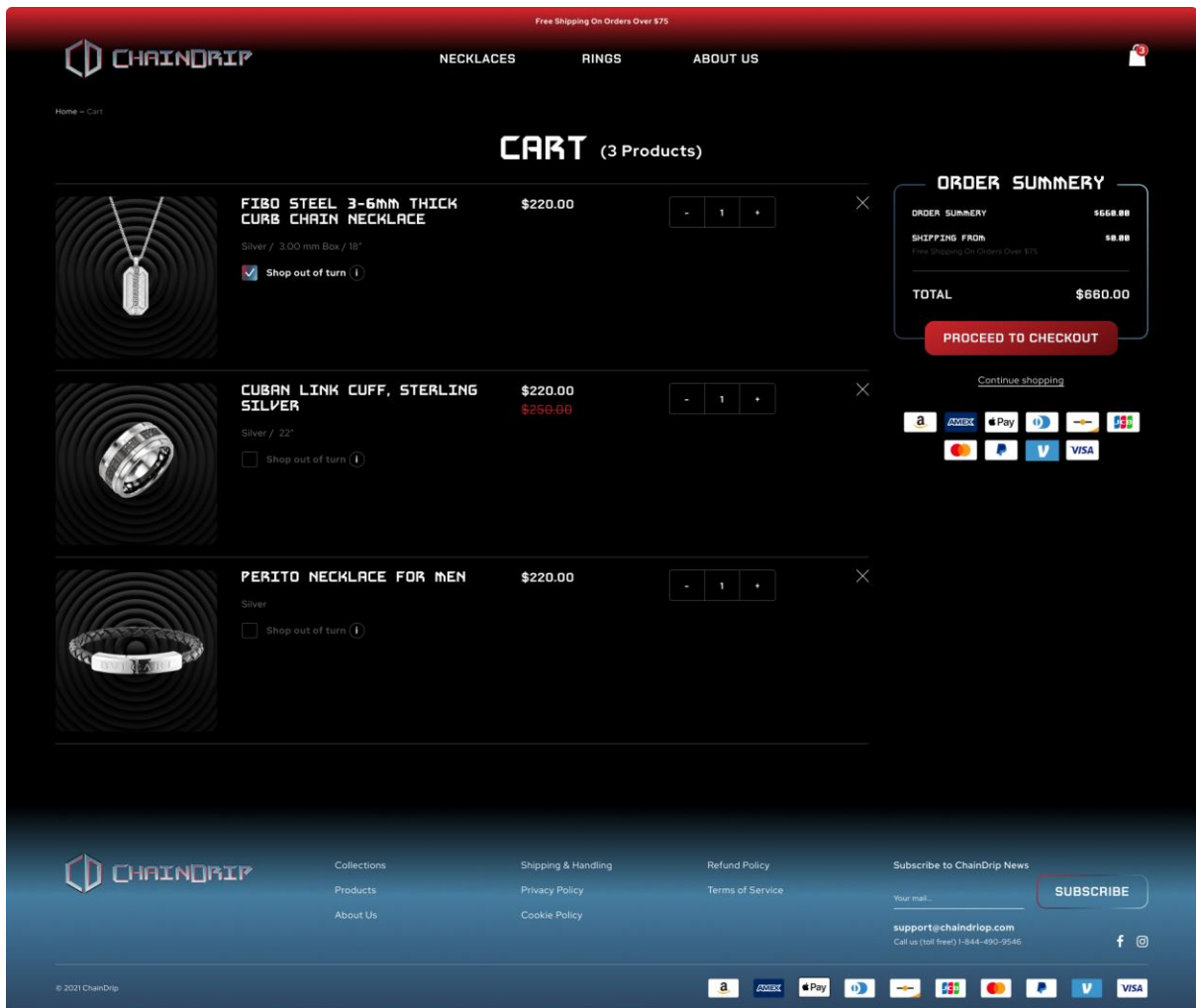


Рисунок 2.11 – Дизайн кошику

Висновки до розділу II

Під час проектування даного інтернет-магазину відбувалось чітке слідування проектній архітектурі Shopify, коли сторінки складаються з секцій, що мають налаштування та, можливо, містять в собі будь-які блоки у кількості - до 16 одиниць на секцію. Після розробки мапи сайту та після визначенням із тим, з яких сторінок складається даний веб-додаток, ці сторінки було розбито на секції. До деяких секцій були дописані певні налаштування, що вдалось прорахувати на етапі проектування. А також була повністю пропрацьована задача по створенню інтернет-магазину для міжнародного продажу. На основі мапи сайту, та подальшої детальної декомпозиції сторінок сайту, було створено дизайн для кожної сторінки та спільних елементів. Дизайн спирається на раніше вказані референси та повністю відповідає вимогам, що

були раніше описані. Після детального проектування та створеного дизайну, відбувся процес реалізації проектної частини.

3 РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ

3.1 HTML каркас додатку

Найголовніший файл із розмітки – це є `theme.liquid`. В ньому підключаються всі службові ресурси, потрібні для кожної сторінки, та елементи, що будуть відображатися на кожній сторінці. Скріншоти прикладу коду даного файла буде розбито на декілька частин.

```

layout > theme.liquid >
1 <!doctype html>
2 <html class="no-js" lang="{{ request.locale.iso_code }}">
3 <head>
4 <meta charset="utf-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width,initial-scale=1">
7 <meta name="theme-color" content="">
8 <link rel="canonical" href="{{ canonical_url }}">
9 <link rel="preconnect" href="https://cdn.shopify.com" crossorigin>
10 <link href="https://www.dafontfree.net/embed/YW1eCltdCwaHktc29saMQtcVndMkhc1ZkYXRhLzExL2EwMjA3MTg0QWxlCBM0XwaRkgU29saMQudRm" rel="stylesheet" type="text/css"/>
11 {% if settings.favicon != blank %}
12 <link rel="icon" type="image/png" href="{{ settings.favicon | img_url: '32x32' }}">
13 {% endif %}
14
15
16 {{ "jquery.js" | asset_url | script_tag }}
17 {% unless settings.type_header_font.system? and settings.type_body_font.system? %}
18 <link rel="preconnect" href="https://fonts.shopifycdn.com" crossorigin>
19 {% endunless %}
20
21 <title>
22 {{ page_title }}
23 {% if current_tags %} &ndash; tagged "{{ current_tags | join: ', ' }}" {% endif %}
24 {% if current_page != 1 %} &ndash; Page {{ current_page }} {% endif %}
25 {% unless page_title contains shop.name %} &ndash; {{ shop.name }} {% endunless %}
26 </title>
27
28 {% if page_description %}
29 <meta name="description" content="{{ page_description | escape }}">
30 {% endif %}
31
32 {% render 'meta-tags' %}
33
34 <script src="{{ 'global.js' | asset_url }}" defer="defer"></script>
35 {{ content_for_header }}
36
37 {% liquid
38 assign body_font_bold = settings.type_body_font | font_modify: 'weight', 'bold'
39 assign body_font_italic = settings.type_body_font | font_modify: 'style', 'italic'
40 assign body_font_bold_italic = body_font_bold | font_modify: 'style', 'italic'
41 %}
42
43 {% style %}
44 {{ settings.type_body_font | font_face: font_display: 'swap' }}

```

Рисунок 3.1 – Файл `theme.liquid`, частина 1

На скріншоті рисунок 3.1 можна побачити підключення головного CDN шопіфай, для роботи із об'єктами, бібліотеки скриптів, як наприклад jQuery, підключення мета тегів, потрібних для індексації сторінок та оголошення liquid змінних, що будуть використані в подальшому, для задання глобальних стилів шрифту.

```

44 {{ settings.type_body_font | font_face: font_display: 'swap' }}
45 {{ body_font_bold | font_face: font_display: 'swap' }}
46 {{ body_font_italic | font_face: font_display: 'swap' }}
47 {{ body_font_bold_italic | font_face: font_display: 'swap' }}
48 {{ settings.type_header_font | font_face: font_display: 'swap' }}
49
50 :root {
51 --font-body-family: {{ settings.type_body_font.family }}, {{ settings.type_body_font.fallback_families }};
52 --font-body-style: {{ settings.type_body_font.style }};
53 --font-body-weight: {{ settings.type_body_font.weight }};
54
55 --font-heading-family: {{ settings.type_header_font.family }}, {{ settings.type_header_font.fallback_families }};
56 --font-heading-style: {{ settings.type_header_font.style }};
57 --font-heading-weight: {{ settings.type_header_font.weight }};
58
59 --font-body-scale: {{ settings.body_scale | divided_by: 100.0 }};
60 --font-heading-scale: {{ settings.heading_scale | times: 1.0 | divided_by: settings.body_scale }};
61
62 --color-base-text: {{ settings.colors.text_red }}, {{ settings.colors.text_green }}, {{ settings.colors.text_blue }};
63 --color-base-background-1: {{ settings.colors.background_1_red }}, {{ settings.colors.background_1_green }}, {{ settings.colors.background_1_blue }};
64 --color-base-background-2: {{ settings.colors.background_2_red }}, {{ settings.colors.background_2_green }}, {{ settings.colors.background_2_blue }};
65 --color-base-solid-button-labels: {{ settings.colors.solid_button_labels_red }}, {{ settings.colors.solid_button_labels_green }}, {{ settings.colors.solid_button_labels_blue }};
66 --color-base-outline-button-labels: {{ settings.colors.outline_button_labels_red }}, {{ settings.colors.outline_button_labels_green }}, {{ settings.colors.outline_button_labels_blue }};
67 --color-base-accent-1: {{ settings.colors.accent_1_red }}, {{ settings.colors.accent_1_green }}, {{ settings.colors.accent_1_blue }};
68 --color-base-accent-2: {{ settings.colors.accent_2_red }}, {{ settings.colors.accent_2_green }}, {{ settings.colors.accent_2_blue }};
69 --payment-terms-background-color: {{ settings.colors.background_1 }};
70
71 --gradient-base-background-1: {% if settings.gradient_background_1 != blank %}{{ settings.gradient_background_1 }}{% else %}{{ settings.colors.background_1 }}{% endif %};
72 --gradient-base-background-2: {% if settings.gradient_background_2 != blank %}{{ settings.gradient_background_2 }}{% else %}{{ settings.colors.background_2 }}{% endif %};
73 --gradient-base-accent-1: {% if settings.gradient_accent_1 != blank %}{{ settings.gradient_accent_1 }}{% else %}{{ settings.colors.accent_1 }}{% endif %};
74 --gradient-base-accent-2: {% if settings.gradient_accent_2 != blank %}{{ settings.gradient_accent_2 }}{% else %}{{ settings.colors.accent_2 }}{% endif %};
75
76 --page-width: {{ settings.page_width | divided_by: 10 }}rem;
77 --page-width-margin: {% if settings.page_width == '1600' %}2{% else %}0{% endif %}rem;
78 }
79
80 *,
81 *:before,
82 *:after {
83 box-sizing: inherit;
84 }
85
86 html {
87 box-sizing: border-box;

```

Рисунок 3.2 – Файл theme.liquid, частина 2

Тут лише деякі глобальні стилі та подальші CSS

```

87 box-sizing: border-box;
88 font-size: calc(var(--font-body-scale) * 62.5%);
89 height: 100%;
90 }
91
92 body {
93 display: grid;
94 grid-template-rows: auto auto 1fr auto;
95 grid-template-columns: 100%;
96 min-height: 100%;
97 margin: 0;
98 font-size: 1.5rem;
99 letter-spacing: 0.06rem;
100 line-height: calc(1 + 0.8 / var(--font-body-scale));
101 font-family: var(--font-body-family);
102 font-style: var(--font-body-style);
103 font-weight: var(--font-body-weight);
104 }
105
106 @media screen and (min-width: 750px) {
107 body {
108 font-size: 1.6rem;
109 }
110 }
111 {% endstyle %}
112
113 {{ 'base.css' | asset_url | stylesheet_tag }}
114
115 {% unless settings.type_body_font.system? %}
116 <link rel="preload" as="font" href="{{ settings.type_body_font | font_url }}" type="font/woff2" crossorigin>
117 {% endunless %}
118 {% unless settings.type_header_font.system? %}
119 <link rel="preload" as="font" href="{{ settings.type_header_font | font_url }}" type="font/woff2" crossorigin>
120 {% endunless %}
121
122 {% if settings.predictive_search_enabled %}
123 <link rel="stylesheet" href="{{ 'component-predictive-search.css' | asset_url }}" media="print" onload="this.media='all'">
124 {% endif %}
125
126 <script>document.documentElement.className = document.documentElement.className.replace('no-js', 'js');
127 if (Shopify.designMode) {
128 document.documentElement.classList.add('shopify-design-mode');
129 }
130 </script>

```

Рисунок 3.3 – Файл theme.liquid, частина 3

```

130 </script>
131 <script src="https://unpkg.com/vue@3.2.31/dist/vue.global.prod.js"></script>
132 <link rel="stylesheet" href="{{ 'fonts.css' | asset_url }}">
133 <link rel="stylesheet" href="{{ 'custom.css' | asset_url }}">
134 <link rel="stylesheet" href="{{ 'product-snippet.css' | asset_url }}">
135 <link rel="stylesheet" href="{{ 'bread-crumbs.css' | asset_url }}">
136 </head>
137
138 <body class="gradient">
139 <a class="skip-to-content-link button visually-hidden" href="#MainContent">
140   {{{ "accessibility.skip_to_text" | t }}}
141 </a>
142 <!--
143   For delete:
144   sections:
145     announcement-bar
146 -->
147
148 {{{ section 'custom-header' %}}
149
150 <main id="MainContent" class="content-for-layout focus-none" role="main" tabindex="-1">
151   {{{ content_for_layout }}}
152 </main>
153
154 {{{ section 'custom-footer' %}}
155
156 <ul hidden>
157 <li id="ally-refresh-page-message">{{{ "accessibility.refresh_page" | t }}}</li>
158 </ul>
159
160 <script>
161 window.shopUrl = '{{ shop_url }}';
162 window.routes = {
163   cart_add_url: '{{ routes.cart_add_url }}',
164   cart_change_url: '{{ routes.cart_change_url }}',
165   cart_update_url: '{{ routes.cart_update_url }}',
166   predictive_search_url: '{{ routes.predictive_search_url }}'
167 };
168
169 window.cartStrings = {
170   error: '{{ "sections.cart.cart_error" | t }}',
171   quantityError: '{{ "sections.cart.cart_quantity_error_html" | t: quantity: "[quantity]" }}'
172 }
173

```

Рисунок 3.4 – Файл theme.liquid, частина 3

```

174 window.variantStrings = {
175   addToCart: `{{{ 'products.product.add_to_cart' | t }}}`,
176   soldOut: `{{{ 'products.product.sold_out' | t }}}`,
177   unavailable: `{{{ 'products.product.unavailable' | t }}}`,
178 }
179
180 window.accessibilityStrings = {
181   imageAvailable: `{{{ 'products.product.media.image_available' | t: index: '[index]' }}}`,
182   shareSuccess: `{{{ 'general.share.success_message' | t }}}`,
183   pauseSlideshow: `{{{ 'sections.slideshow.pause_slideshow' | t }}}`,
184   playSlideshow: `{{{ 'sections.slideshow.play_slideshow' | t }}}`,
185 }
186 </script>
187
188 {%- if settings.predictive_search_enabled -%}
189 <script src="{{ 'predictive-search.js' | asset_url }}" defer="defer"></script>
190 {%- endif -%}
191 {{{ "slick.min.js" | asset_url | script_tag }}}
192 </body>
193 </html>

```

Рисунок 3.5 – Файл theme.liquid, частина 4

3.2 Стилiзацiя та адаптивнiсть

Для надання структурi HTML вiдповiдностi дизайну, не було використану препроцесори для CSS чи будь-якi фреймворки, як наприклад TailWind. Стандартного набору функцiй CSS достатньо. А для адаптивностi та responsive сайту, використовувався пiдхiд iз поєднанням використання вiдносних одиниць вимiрювання, таких як vw/vh та медiа запитiв, оскiльки на мобiльних девайсах довелось в деяких мiсцях використовувати фiксований розмiр деяких елементiв, тому що ширина вьюпорта дуже маленька для правильного вiдображення.

Нижче наведені приклади коду стилізації додатку:

```

1 @charset "UTF-8";
2
3 .cart {
4   width: 91.67%;
5   margin: 0 auto;
6   padding-top: 150px;
7 }
8
9 .cart_header {
10  width: 100%;
11  display: -webkit-box;
12  display: -ms-flexbox;
13  display: flex;
14  -webkit-box-pack: center;
15  -ms-flex-pack: center;
16  justify-content: center;
17  -webkit-box-align: end;
18  -ms-flex-align: end;
19  align-items: flex-end;
20  margin-bottom: 40px;
21 }
22
23 .cart_heading {
24  margin: 0;
25  margin-right: 20px;
26 }
27
28 .cart_main {
29  width: 100%;
30  display: -webkit-box;
31  display: -ms-flexbox;
32  display: flex;
33  -webkit-box-pack: justify;
34  -ms-flex-pack: justify;
35  justify-content: space-between;
36 }
37
38 .cart_line-items_container {
39  width: 74.43%;
40 }
41
42 .line-item {
43  width: 100%;
44  padding: 20px 0;
45 }
46
47 .line-item_image_wrapper {
48  width: 100%;
49  padding: 20px 0;
50  display: -webkit-box;
51  display: flex;
52  -webkit-box-pack: justify;
53  -ms-flex-pack: justify;
54  justify-content: space-between;
55 }
56
57 .line-item_image {
58  width: 13.55%;
59  height: 13.55%;
60  display: -webkit-box;
61  display: flex;
62  -webkit-box-pack: center;
63  -ms-flex-pack: center;
64  justify-content: center;
65  background-image: url("product-bcg.jpg");
66  background-repeat: no-repeat;
67  background-size: cover;
68  background-position: center;
69  border-radius: 15px;
70  -webkit-border-radius: 15px;
71  -moz-border-radius: 15px;
72  -ms-border-radius: 15px;
73  -o-border-radius: 15px;
74 }
75
76 .line-item_image {
77  width: 100%;
78  height: 100%;
79  -o-object-fit: cover;
80  object-fit: cover;
81 }
82
83 .line-item_general-info {
84  width: 21.35%;
85 }
86
87 .line-item_title {
88  width: 100%;
89 }
90
91 .line-item_title {
92  margin: 0;
93  text-align: left;
94 }
95
96 .line-item_options {
97  opacity: 0.5;
98  margin: 20px 0;
99 }
100
101 .out-of-turn_wrapper {
102  width: 87.32%;
103  margin: 0;
104  padding: 0;
105  display: -webkit-box;
106  display: -ms-flexbox;
107  display: flex;
108  -webkit-box-pack: start;
109  -ms-flex-pack: start;
110  justify-content: flex-start;
111  -webkit-box-align: center;
112  -ms-flex-align: center;
113  align-items: center;
114 }
115
116 .out-of-turn_label {
117  margin-left: 15px;
118  margin-right: 40px;
119  display: -webkit-box;
120  display: -ms-flexbox;
121  display: flex;
122  -webkit-box-align: center;
123  -ms-flex-align: center;
124  align-items: center;
125  -webkit-border-radius: 3px;
126  -ms-border-radius: 3px;
127 }
128
129 input[type="checkbox"].custom-checkbox:after {
130  display: block;
131  width: 10px;
132  height: 10px;
133  border: 1px solid #444;
134  border-radius: 3px;
135  -webkit-border-radius: 3px;
136  -ms-border-radius: 3px;
137 }

```

Рисунок 3.6 – Частина коду стилізації додатку 1

```

1 .main-header {
2   position: absolute;
3   z-index: 3;
4   top: 0;
5   left: 0;
6   width: 100%;
7   background: -webkit-gradient(linear, left top, left bottom,
8     background: linear-gradient(180deg, #fcb262 4.55%, #fcb262 4.55%, #fcb262 4.55%, #fcb262 4.55%);
9 }
10
11 .main-header_wrapper {
12  display: -webkit-box;
13  display: -ms-flexbox;
14  display: flex;
15  margin: 0 auto;
16  width: 91.5%;
17  -webkit-box-pack: justify;
18  -ms-flex-pack: justify;
19  justify-content: space-between;
20  -webkit-box-align: center;
21  -ms-flex-align: center;
22  align-items: center;
23 }
24
25 .main-header_cart-wrapper {
26  margin-bottom: 15px;
27  position: relative;
28 }
29
30 .main-header_cart-wrapper .visually-hidden {
31  display: none;
32 }
33
34 .cart-count-bubble {
35  height: 20px;
36  width: 20px;
37  background: #fcb262;
38  top: 0 !important;
39  right: -10px;
40  border-radius: 50%;
41  display: flex;
42  justify-content: center;
43  align-items: center;
44  position: absolute;
45 }
46
47 .cart-count-bubble span {
48  line-height: 10px;
49 }
50
51 .main-header_mid-column {
52  -webkit-box-pack: center;
53  -ms-flex-pack: center;
54  justify-content: center;
55  padding-right: 12.5%;
56 }
57
58 .main-header_links-container {
59  padding: 0;
60 }
61
62 .main-header_link-wrapper:not(:last-child) {
63  margin-right: 50px;
64 }
65
66 .main-header_link:after {
67  content: "";
68  opacity: 0;
69  display: inline-block;
70  width: 95%;
71  border: 2px solid #fcb262;
72  border-image-slice: 1;
73  transition: 0.35s;
74  -webkit-transition: 0.35s;
75  -moz-transition: 0.35s;
76  -ms-transition: 0.35s;
77  -o-transition: 0.35s;
78  -transition: 0.35s;
79 }
80
81 .main-header_link:hover:after {
82  opacity: 1;
83 }
84
85 .main-header_links-container {
86  display: -webkit-box;
87  display: -ms-flexbox;
88  display: flex;
89 }
90
91 .main-header_burger {
92  display: none;
93 }
94
95 .main-header_mobile {
96  position: absolute;
97  top: 0;
98  left: 100%;
99  width: 100%;
100  height: 100%;
101  z-index: 10;
102  display: -webkit-box;
103  display: -ms-flexbox;
104  display: flex;
105  -webkit-box-pack: justify;
106  -ms-flex-pack: justify;
107  justify-content: space-between;
108  background-color: #444;
109  opacity: 0;
110  transition: 0.35s;
111  -webkit-transition: 0.35s;
112  -ms-transition: 0.35s;
113  -o-transition: 0.35s;
114  -transition: 0.35s;
115 }
116
117 @media (max-width: 769px) {
118  .main-header_logo-wrapper {
119    display: none;
120  }
121  .main-header_links-container {
122    display: none;
123  }
124  .main-header_links-container .padding-top: 10px;
125  .main-header_links-container .padding-bottom: 10px;
126  .main-header_links-container .margin-bottom: 10px;
127 }

```

Рисунок 3.7 – Частина коду стилізації додатку 2

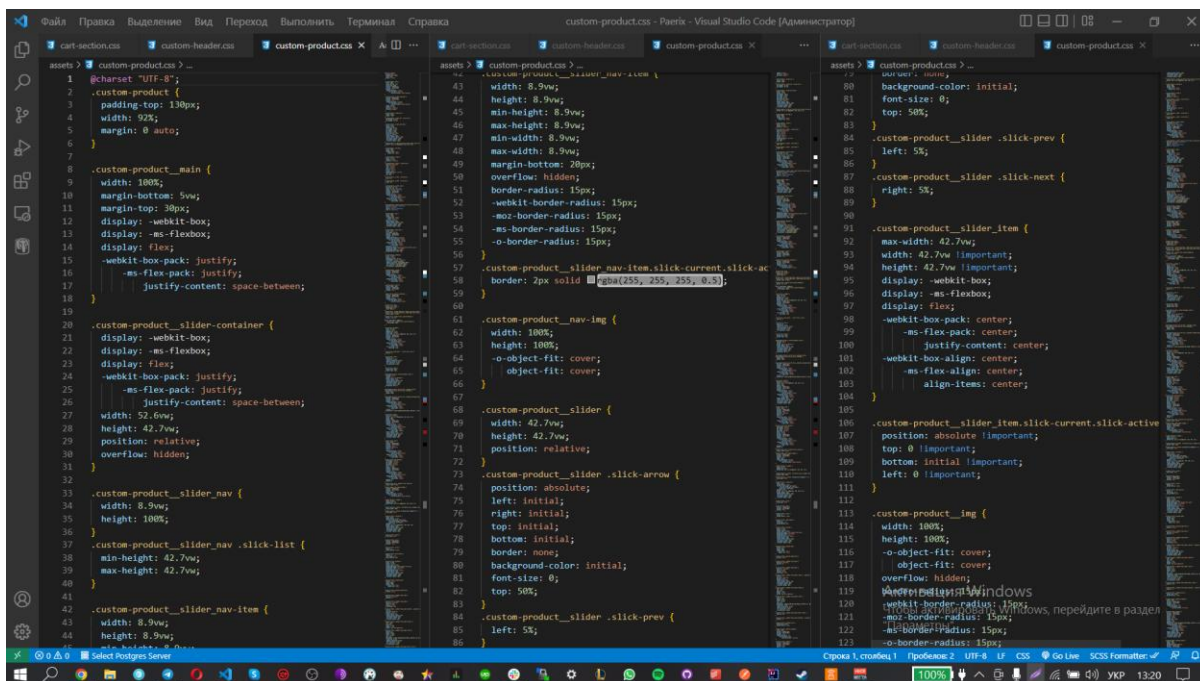


Рисунок 3.8 – Частина коду стилізації додатку 3

3.3 Створення моделей налаштувань секцій та відображення даних

Для того, щоб у власника магазину була змога змінювати дані через кастомізаційну панель, використовується Schema тег, де описуються дані, їх тип, айді, та лейбл, тобто назва, чи опис у панелі. В розмітці виконується звернення до об'єкта і його налаштувань, по айді елемента. Сервер замість цього візьме дані з панелі та добавить їх на місце розташування.

Приклад коду:

```

1  {{ 'main-video-banner.css' | asset_url | stylesheet_tag }}
2
3  <section class="main-video-banner">
4    <div class="video-wrapper">
5      <video class="main-video-banner_background" autoplay muted loop 1
6        <source src="{{ section.settings.video }}" type="video/mp4">
7      </video>
8    </div>
9    <div class="main-video-banner_text-content-wrapper">
10     <h1 class="main-video-banner_text-content_heading main-heading">
11       {{ section.settings.heading }}
12     </h1>
13     <a href="{{ section.settings.url }}" class="main-video-banner_text-content_link">
14       {{ section.settings.link_text }}
15     </a>
16   </div>
17 </section>
18
19 {% schema %}
20 {
21   "name": "Main video banner",
22   "presets": [
23     {
24       "category": "home-page",
25       "name": "Main video banner"
26     }
27   ],
28   "settings": [
29     {
30       "type": "url",
31       "id": "video",
32       "label": "Video background"
33     },
34     {
35       "type": "text",
36       "id": "heading",
37       "label": "Heading",
38       "default": "The future of jewelry"
39     },
40     {
41       "type": "text",
42       "id": "link_text",
43       "label": "link_text",
44       "default": "Current collection: Genesis"
45     }
46   ],
47   "sections": [
48     {
49       "type": "url",
50       "id": "url",
51       "label": "URL"
52     }
53   ]
54 }
55 {% endschema %}

```

Рисунок 3.9 – Код секції відео банеру головної сторінки

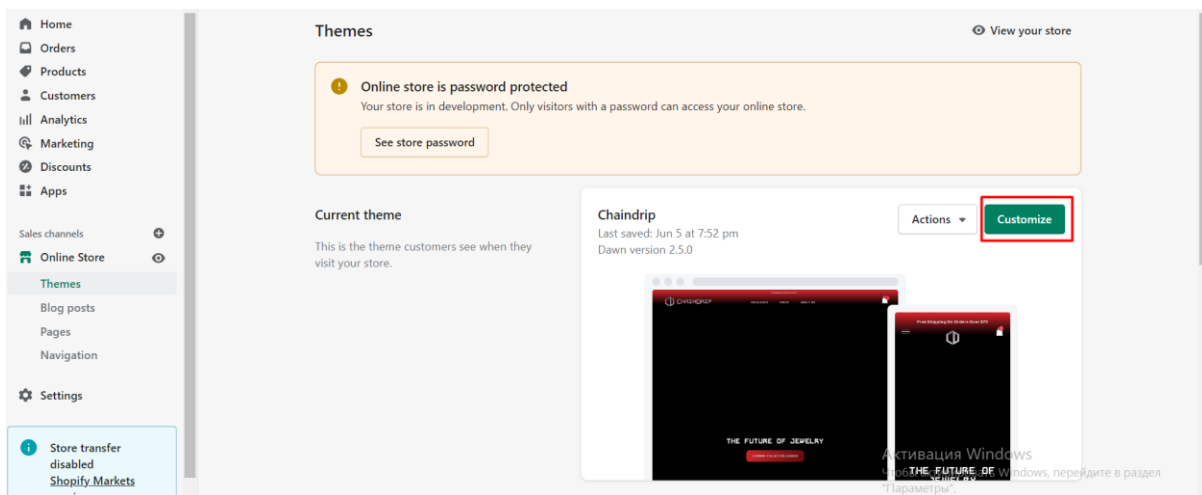


Рисунок 3.10 – Точка входу в кастомізаційний режим

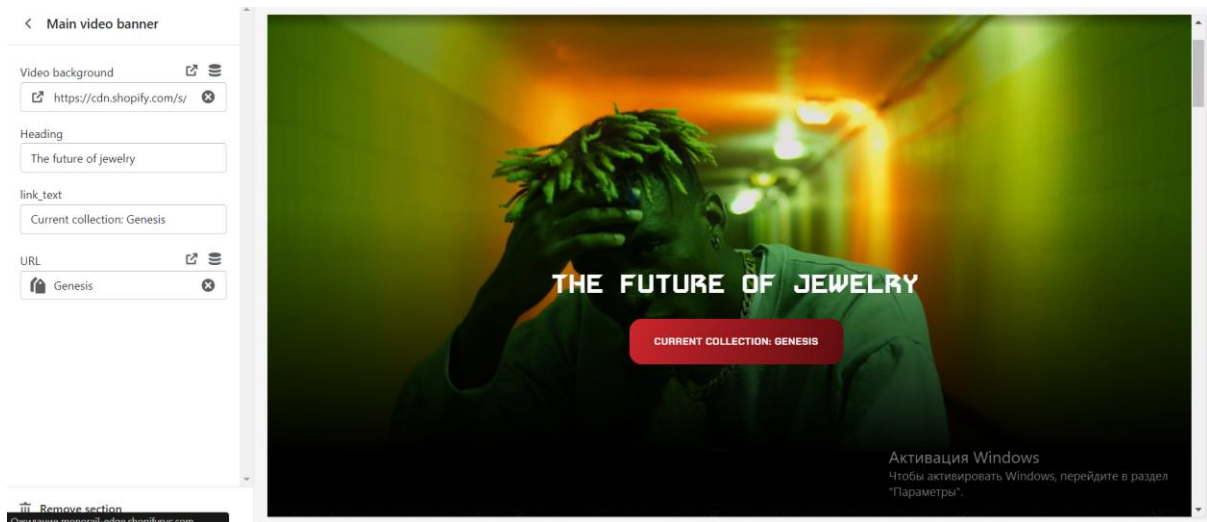


Рисунок 3.11 – Внесення даних секції

Як можна побачити на рисунку 3.10, є кнопка налаштувань теми, далі власник магазину може зайти в цей режим, обрати сторінку та секцію для налаштувань. Далі, на рисунку 3.11 можна побачити створену секцію, та дані, що були внесені зліва, відображаються на сайті.

3.4 Програмування компонентів фільтрації, пошуку, сортування

Для реалізації функціоналу фільтрації на сторінці колекцій було використано фреймворк Vue.js задля реактивності даного процесу. Якщо описувати логіку роботи функціоналу скорочено, то принцип виглядає наступним чином: спочатку відбувається опис об'єкту фреймворку, під час його створення. Для опису він приймає в якості параметра об'єкт, в якому описується назва компонента, синтаксис роздільника, який використовується в розмітці. За замовчуванням фреймворк використовує вусатий синтаксис, але задля запобігання конфліктів з шаблонізатором, який використовує такий самий роздільник, тож було прийнято рішення змінити роздільник, який можна побачити на рисунку 3.12. Далі виконується оголошення та ініціалізація полів цього об'єкта. Потім опис та реалізація методів, потрібних для роботи далі. Ініціалізується поле-спостерігач за рядком фільтра. І, використовуючи хук початкового створення компонента, викликається ланцюжок виклику методів для ініціалізації подальшого функціоналу.

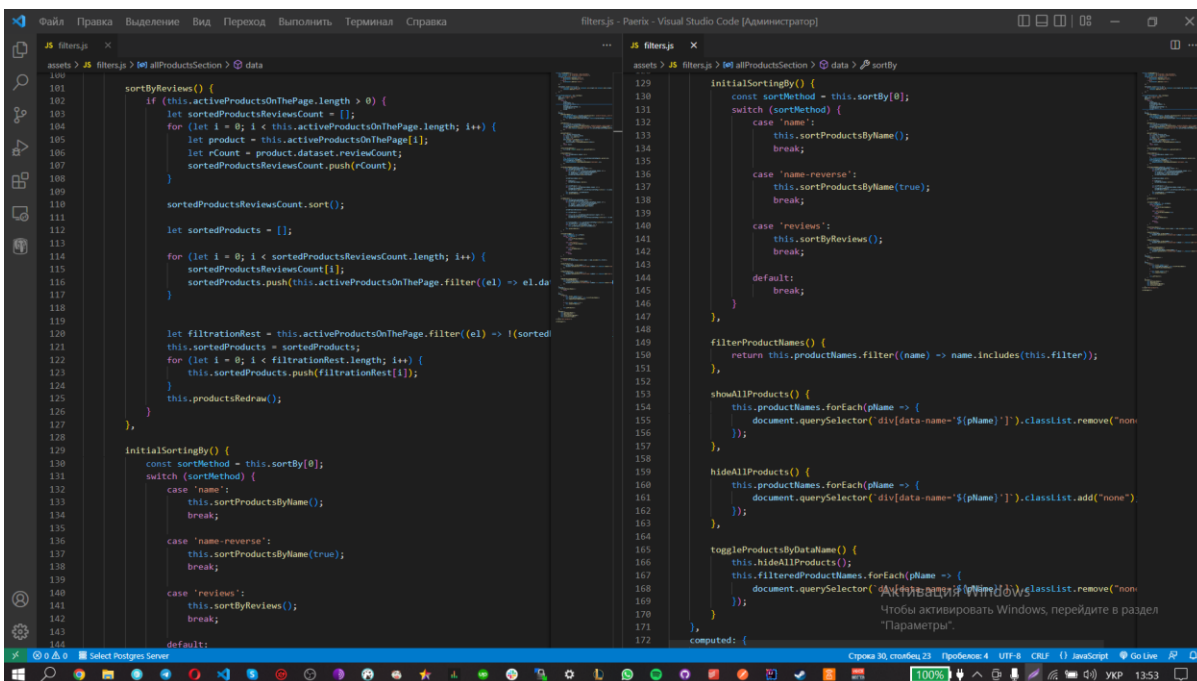


Рисунок 3.14 – Деталі реалізації методів компонента, частина 2

3.5 Програмування продуктової сторінки та кошику

Функціонал файлу cart.js є стандартним набору CRUD функцій для елементів кошику, таких як варіанти продуктів. В даній ситуації використовується Ajax Shopify Арі для відправки запитів на сервер. Логіка полягає в тому, що коли спрацьовують події кліку чи зміни на елементах, що прослуховуються, викликаються обробники подій, що викликають раніше описані функції для відправки даних на сервер. Нижче наведений код реалізації функціоналу кошика:

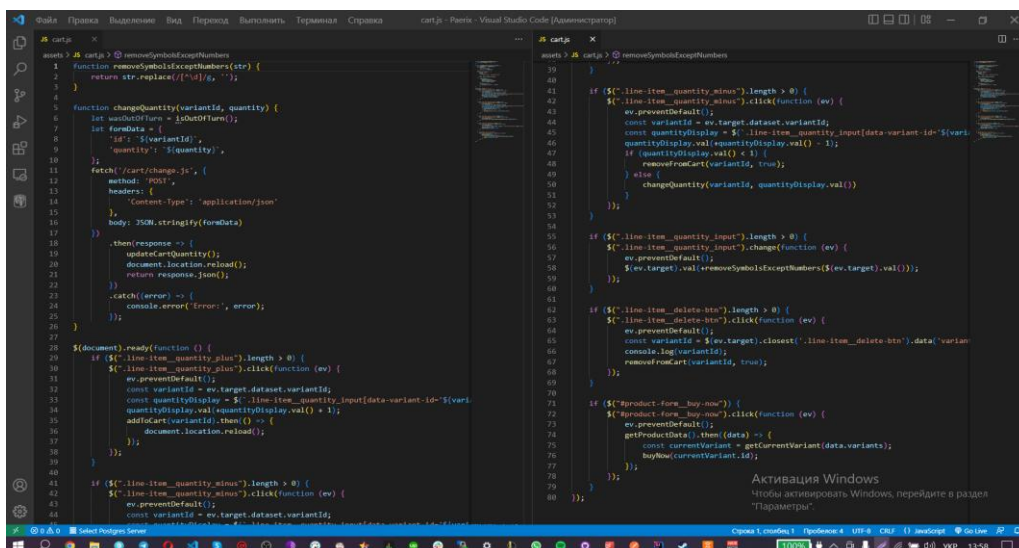


Рисунок 3.15 – Код реалізації функціоналу кошика

Нижче наведений повний код реалізації product page source.

```
function isOutOfTurn() {
  if ($("#out-of-turn").is(":checked")) {
    return true;
  } else {
    return false;
  }
}

function getLocation() {
  const location = window.location.href.split('/');
  return location[location.length - 1];
}

async function getProductsFromCart() {
  let productData;
  await $.getJSON('/cart.js')
    .done(function (json) {
      productData = json;
    })
    .fail(function (jqxhr, textStatus, error) {
      var err = textStatus + ", " + error;
      console.log("Request Failed: " + err);
    });

  return productData;
}

function updateCartQuantity() {
  getProductsFromCart().then(products => {
    let quantity = 0;
    if (isOutOfTurn()) {
      quantity = products.item_count - 1;
    } else {
      quantity = products.item_count;
    }
    $("#cart-quantity")[0].innerText = quantity + " items";
    $("#cart-quantity__hidden")[0].innerText = quantity;
  });
}

function addedProductMessage() {
  $('product-added-message').css({
    'display': 'flex'
  })
  $('product-added-message__text').addClass('fadeIn');
  setTimeout(() => {
    $('product-added-message__text').removeClass('fadeIn');
  });
}
```

```

}, 1500);
setTimeout(() => {
  $('.product-added-message').css({
    'display': 'none'
  })
}, 2000);
}

```

```

function addToCart(variantId, amount = 1) {
  if (!variantId) {
    return;
  }
  let formData = {
    'items': [{
      'id': variantId,
      'quantity': amount
    }]
  };
  fetch('/cart/add.js', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(formData)
  })
  .then(response => {
    updateCartQuantity();
    addedProductMessage();
    if (getLocation() === 'cart') {
      document.location.reload();
    }
    return response.json();
  })
  .catch((error) => {
    console.error('Error:', error);
  });
}

```

```

function removeFromCart(variantId, isReload = false) {
  let wasOutOfTurn = isOutOfTurn();
  let formData = {
    'id': `${variantId}`,
    'quantity': '0'
  };
  fetch('/cart/change.js', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },

```

```

    body: JSON.stringify(formData)
  })
  .then(response => {
    if (getLocation() === 'cart') {
      document.location.reload();
    }
    if (!wasOutOfTurn) {
      updateCartQuantity();
    }
    if (isReload) {
      location.reload();
    }
    return response.json();
  })
  .catch((error) => {
    console.error('Error:', error);
  });
}

async function getProductData() {
  if ($('#product-handle').length > 0) {
    let productData;
    const productHandle = $("#product-handle").val();
    await $.getJSON(`/products/${productHandle}.js`)
      .done(function (json) {
        productData = json;
      })
      .fail(function (jqxhr, textStatus, error) {
        var err = textStatus + ", " + error;
        console.log("Request Failed: " + err);
      });

    return productData;
  }
}

function getSelectedValues() {
  const checkedInputs = $('input:checked');
  let valuesList = [];
  checkedInputs.each(function (idx) {
    valuesList.push(this.value);
  });
  return valuesList;
}

function getCurrentVariant(variantsList) {
  const selectedValues = getSelectedValues();
  if (variantsList.length > 1) {
    const currentVariant = variantsList.filter(variant => {

```

```

    let coincidences = 0;
    variant.options.forEach(option => {
      if (selectedValues.includes(option)) {
        coincidences++;
      };
    });
    if (coincidences === 3) {
      return variant;
    }
  });

  return currentVariant[0];
} else {
  return variantsList[0];
}
}

function changePrice(price, priceCompare = 0) {
  if (priceCompare !== 0 && priceCompare !== price) {
    document.querySelector(".custom-product__price-compare").innerText = "$" + priceCompare;
  }
  document.querySelector(".custom-product__price").innerText = "$" + price;
}

function buyNow(productId) {
  addToCart(productId);
  setTimeout(function () {
    window.location.href = '/checkout';
  }, 50);
}

function getReviewsSection() {
  const shopifySections = $(".shopify-section");
  const reviewsContainer = $("#shopify-product-reviews");
  let reviewsSection;
  shopifySections.each(function (idx) {
    if ($(this).find(reviewsContainer)[0]) {
      reviewsSection = shopifySections[idx];
    }
  });
  return reviewsSection;
}

function customizeReviewsSection() {
  const reviewsSection = getReviewsSection();
  const revCount = $(' .spr-summary-actions-togglereviews');

```

```

$('.spr-summary-actions').addClass("write-review__btn-wrapper");
$('.spr-summary-actions').find("a").addClass("write-review__btn");
$('.spr-review-header-title').addClass("red-hat__20-20_700");
$('.spr-review-header-byline strong').addClass("red-hat__15-20_700");
if ($(".spr-pagination-page").length > 0) {
  $(".spr-pagination-page").addClass("red-hat__15-20_700");
}
$(".spr-review-content-body").addClass("red-hat__15-20_400");
$(".spr-review-footer.rte").addClass("none");
[$(".spr-form-input-text"), $(".spr-form-input-email"), $(".spr-form-input-textarea")].forEach(el => {
  $(el).addClass("custom-input");
  $(el).addClass("red-hat__12-15_400");
});
$(".spr-form-label").addClass("red-hat__12-15_400");
revCount.addClass('red-hat__12-15_700');
}

function moveReviewsToMain() {
  const reviewsSection = getReviewsSection();
  $(reviewsSection).find(".page-width").removeClass('page-width');
  $(reviewsSection).addClass('shopify-product-reviews__section');
  $(reviewsSection).clone().appendTo('.product-page__left');
  $(reviewsSection).remove();
}

function setHaveTheChoiseLink(newLink) {
  $('#have-the-choice__link')[0].href = newLink;
}

function haveTheChoise() {
  if ($('#have-the-choice .products-offer__sidebar-item_input').length < 1) {
    return;
  }

  $('#have-the-choice .products-offer__sidebar-item_input').change(function() {
    const productId = this.id;
    $('#have-the-choice .products-offer__display_img-active').removeClass('products-offer__display_img-active');
    $('#have-the-choice .products-offer__display_img[data-product-id='${productId}']').addClass('products-offer__display_img-active');
    const productLink = $('#have-the-choice .products-offer__display_img[data-product-id='${productId}']')[0].dataset.productUrl;
    setHaveTheChoiseLink(productLink);
  });
}

$(document).ready(function () {
  if ($('#out-of-turn').length > 0) {
    $('#out-of-turn').change(function () {

```

```

    let ischecked = $(this).is(':checked');
    const id = $(this).data('product-id');
    if (!ischecked) {
      removeFromCart(id)
    } else {
      addToCart(id)
    }
  });
}

$(".product-form__add").click(function (ev) {
  ev.preventDefault();
  getProductData().then((data) => {
    const currentVariant = getCurrentVariant(data.variants);
    addToCart(currentVariant.id);
  });
});

$("#product-form__buy-now").click(function (ev) {
  ev.preventDefault();
  getProductData().then((data) => {
    const currentVariant = getCurrentVariant(data.variants);
    if (!(currentVariant)) {
      buyNow(ev.target.dataset.id);
    } else {
      buyNow(currentVariant.id);
    }
  });
});

moveReviewsToMain();
setTimeout(function () {
  customizeReviewsSection();
}, 1500);

$('.custom-radio-btn__round').click(function () {
  getProductData().then((data) => {
    const currentVariant = getCurrentVariant(data.variants);
    const price = currentVariant.price / 100;
    const compare = currentVariant.compare_at_price / 100;
    changePrice(price, compare);
  })
});

$('.custom-radio-btn__label').click(function () {
  getProductData().then((data) => {
    const currentVariant = getCurrentVariant(data.variants);
    const price = currentVariant.price / 100;
    const compare = currentVariant.compare_at_price / 100;

```

```

    changePrice(price, compare);
  })
});

haveTheChoise();
});

```

3.6 Процес оплати замовлення

В якості тестового платіжного шлюзу використовується bogus gateway згідно з документацією Shopify для тестування оплат магазинів.

Field name	Details to enter
Name on card	Enter <code>Bogus Gateway</code>
Credit card number	Enter each of these in turn to simulate different types of transaction: <ul style="list-style-type: none"> • Enter <code>1</code> to simulate a successful transaction • Enter <code>2</code> to simulate a failed transaction • Enter <code>3</code> to simulate an exception (this generates a message indicating that an error has occurred with the provider)
CVV	Enter any 3-digit number (for example, <code>111</code> .)
Expiry Date	Enter any date in the future.

Рисунок 3.16 – Документація Bogus Gateway

Після переходу до оплати замовлення, відбувається ввід контактних даних користувача. В якості країни доставки буде використовуватися Багами.

Рисунок 3.17 – Контактна інформація замовлення

Далі відбувається процес оплати, де користувач вводить дані своєї картки. На рисунку 3.17 дані введені для тестування, згідно з документацією.

Cart > Information > Payment

Contact [Change](#)

Billing [Change](#)

Payment
All transactions are secure and encrypted.

⚠ This store can't accept real orders or real payments.

Credit card B

Card number 🔒

Name on card

Expiration date (MM / YY) Security code 🔒

[Pay now](#) [Return to information](#)

Total UAH **2100.00**

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Рисунок 3.18 – Дані для оплати замовлення

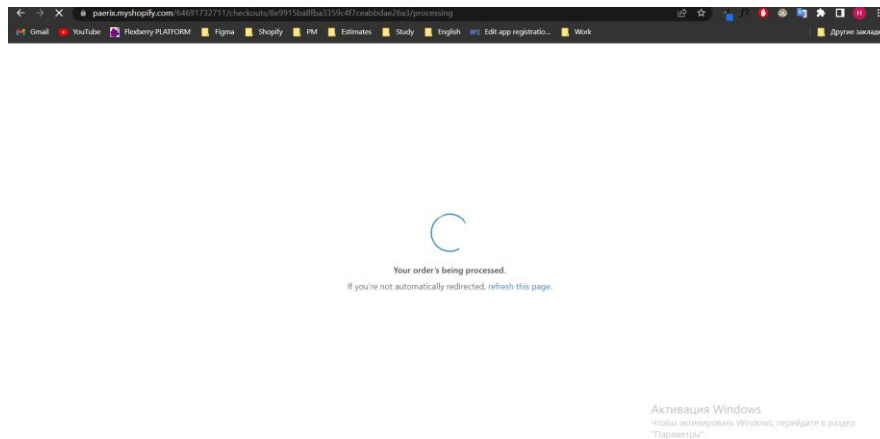


Рисунок 3.19 – Очікування на результат транзакції

Після того, як транзакція була виконана успішно, то користувач отримує наступне повідомлення, яке показано на рисунку 3.20.

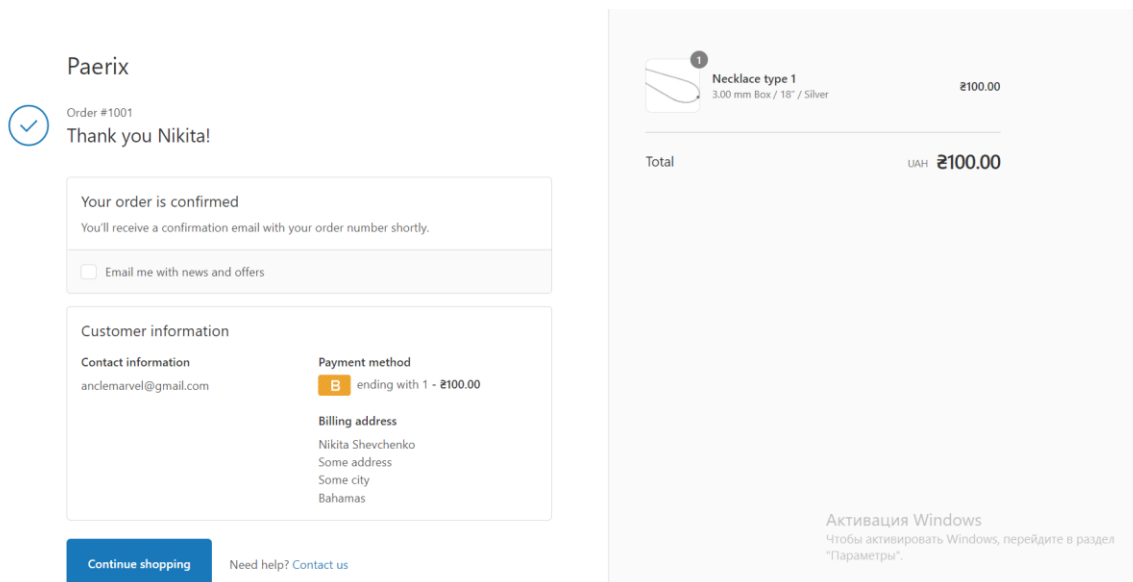


Рисунок 3.20 – Успішна оплата замовлення

В той час, як власник може побачити в адмін панелі дані, щодо замовлення та його статусу:

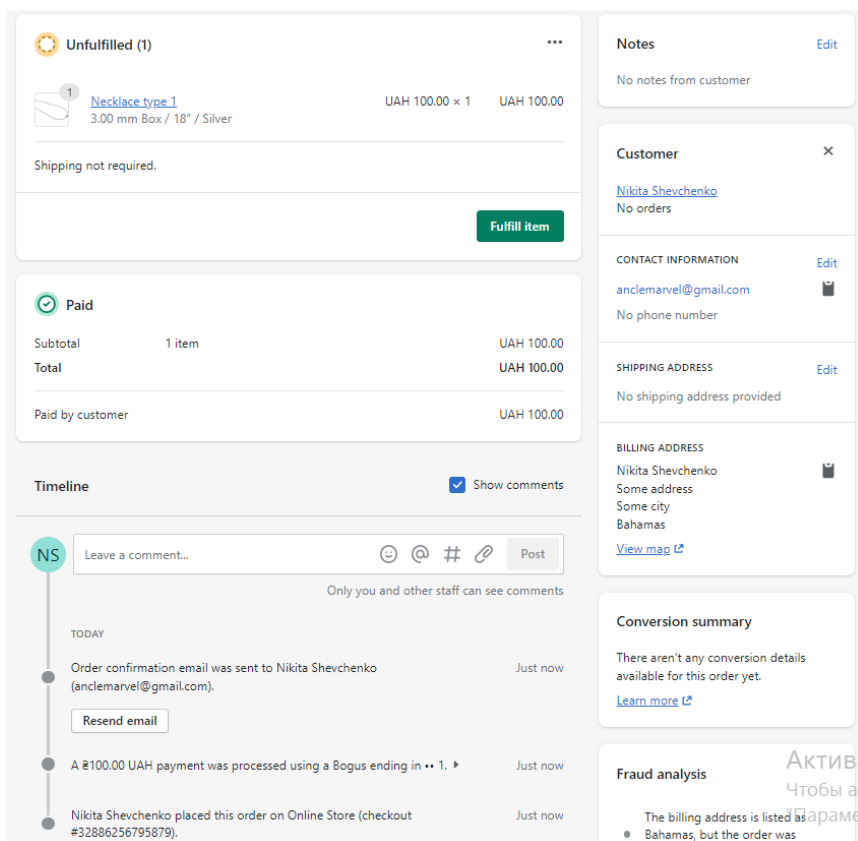


Рисунок 3.21 – Замовлення в адмін панелі

Висновки до розділу III

Завдяки детальному проектуванню додатку, було розроблено повний обсяг функціоналу. У роботі було використано фреймворк Vue.js, а якості сучасної практики для розробки стандартних рішень по реактивному сортуванню, фільтрації та пошуку елементів на сторінці. Також було використано Ajax Shopify API. Розмітка та стилізація сайту відповідає розробленому дизайну.

Під час тестування платіжної системи не було виявлено будь-яких помилок, замовлення відображається в адмін панелі зі статусом оплаченого.

ВИСНОВКИ

Проаналізувавши сучасний стан інтернет-продажів, було виявлено, що актуальність інтернет-продажів з кожним роком зростає, а пандемія, що спричинила кризу у майже всіх галузях економіки, навпаки дала поштовх для збільшення капіталізації електронної комерції. Актуальність теми підтверджується цифрами аналітиків.

Задля створення та налагодження інтернет-магазину для міжнародного продажу було обрано платформу Shopify, оскільки вона задовольняє всі поставлені вимоги як розробки, в технічному плані, так і бюджет бізнесу. В якості інструментів та технологій розробки було обрано наступний стек технологій: html, liquid, css, js, jq, vue.js.

Коли ціль та мета була чітко поставлена, було виконано проектування додатку, обрано архітектуру проекту в якості MVC, створено мапу сайту та розбито сторінки на секції та елементи з описом налаштувань даних. Дизайн додатку було розроблено на основі референсів, додаток цілком відповідає сучасним вимогам дизайну.

На основі спроектованого додатку була виконана реалізація кодової частини інтернет-магазину. Весь спроектований функціонал був написаний та реалізований так, як передбачалось, весь план був виконаний. Додаток було протестовано із використанням тестової платіжної системи.

Практична цінність роботи полягає у тому, що розроблений інтернет-магазин для продажів товарів з можливістю міжнародних платежів для малого та середнього бізнесу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Coronavirus pandemic adds \$219 billion to US ecommerce sales in 2020-2021. [Електронний ресурс] / Berthene, A. 15 March 2022 p. <https://www.digitalcommerce360.com/article/coronavirus-impact-online-retail/>
2. How to Work With Shopify Theme Blocks. [Електронний ресурс] / Griffin, L. 13 Oct 2021 p. <https://www.shopify.com/partners/blog/theme-blocks>
3. How to Test your Theme Before Submitting to the Shopify Theme Store. [Електронний ресурс] / Griffin, L. 3 Jan 2022 p. <https://www.shopify.com/partners/blog/test-shopify-theme>
4. How to work with Metafields when building Shopify themes. [Електронний ресурс] / Griffin, L. 2 Mar 2022 p. <https://www.shopify.com/partners/blog/metafields>
5. Tips for an Efficient Shopify Theme Review Process. [Електронний ресурс] / Gunawan, P. 9 Aug 2021 p. <https://www.shopify.com/partners/blog/theme-review-process>
6. Tapping the Huge Market for Mobile-Optimized Apps. [Електронний ресурс] / Kelley, A. 22 May 2022 p. <https://www.shopify.com/partners/blog/mobile-best-practices>
7. Content management systems. [Електронний ресурс] / www.nibusinessinfo.co.uk: <https://www.nibusinessinfo.co.uk/content/important-cms-features-and-functions>
8. Content management system (CMS). [Електронний ресурс] / Sarah Amstler, F. C. Feb 2021 p. <https://www.techtarget.com/searchcontentmanagement/definition/content-management-system-CMS>
9. Shopify cheat sheet. [Електронний ресурс] / www.shopify.com: <https://www.shopify.com/partners/shopify-cheat-sheet>
10. Liquid references. [Електронний ресурс] / shopify.dev: https://shopify.dev/api/liquid#liquid_basics

- 11.Shopify Ajax API. [Электронный ресурс] / shopify.dev:
<https://shopify.dev/api/ajax>
- 12.15 Best and Most Popular CMS Platforms in 2022 (Compared). [Электронный ресурс] / Staff, E. 1 Jan 2022 p. <https://www.wpbeginner.com/showcase/best-cms-platforms-compared/>
- 13.Content management system. [Электронный ресурс] / 2 June 2022 p. en.wikipedia.org: https://en.wikipedia.org/wiki/Content_management_system
- 14.Customer relationship management. [Электронный ресурс] / 21 June 2022 p. en.wikipedia.org: https://en.wikipedia.org/wiki/Customer_relationship_management
- 15.Top Tips for Building Themes That Are Fully Featured, Like Customizing Shopify's Best Free Themes. [Электронный ресурс] / Williamson, S. 25 Jan 2022 p. <https://www.shopify.com/partners/blog/best-free-themes>