

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента

Гончарова Максима Сергійовича
(ПІБ)

академічної групи

122-20-2
(шифр)

спеціальності

122 Комп'ютерні науки
(код і назва спеціальності)

освітньої програми

Комп'ютерні науки
(назва освітньої програми)

на тему:

Розробка веборієнтованого додатку для
продажу товарів, згенерованих штучним інтелектом
на базі платформи Shopify

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Ширін А.Л.			
розділів:				
спеціальний	доц. Ширін А.Л.			
економічний	доц. Касьяненко Л.В.			
Рецензент				
Нормоконтролер	доц. Гуліна І.Г.			

Дніпро
2024

РЕФЕРАТ

Пояснювальна записка: 75 с., 28 рис., 7 табл., 3 дод., 19 джерел.

Об'єкт розробки: веборієнтований додаток для продажу товарів, згенерованих штучним інтелектом.

Мета кваліфікаційної роботи: розробка веборієнтованого додатку для продажу товарів, згенерованих штучним інтелектом на основі технологічного стеку HTML/CSS/Python/Django framework.

У вступі розглядається необхідність та попит на онлайн-магазини, а також важливість їх швидкого створення, зручного керування та адміністрування, що сприятиме зростанню популярності вебсайту.

У першому розділі описується переваги використання платформи Shopify для створення свого онлайн-магазину, важливі функції та аспекти, які повинен в собі мати веб орієнтований додаток, а також принцип їх роботи.

У другому розділі розглядається функціонал додатку, створення візуальної частини та процес створення сервісу для імпорту товарів.

В економічному розділі оцінюють складність проєкту, визначають витрати на розробку веб платформи з урахуванням різних факторів (кількість рядків коду, використані ресурси, споживання електроенергії, Інтернет тощо), а також розраховують необхідний час для його реалізації.

Практичне значення роботи полягає в створенні платформи для продажу товарів, згенерованих штучним інтелектом, які можна оформлювати у окремому файлі та імпортувати на вебсайт, де можна зручно вести облік за допомогою платформи Shopify.

Актуальність платформи обумовлюється сучасним вибором більшості людей робити покупки не виходячи з дому, а також заохочування до використання штучного інтелекту як помічника та інструмент для покращення життя.

Список ключових слів: ВЕБ ОРІЄНТОВАНІСТЬ, HTML, CSS, PYTHON, DJANGO, FRAMEFORK, ВЕБСАЙТ, ВЕБ ПЛАТФОРМА, ШТУЧНИЙ ІНТЕЛЕКТ

ABSTRACT

Explanatory note: 75 p., 28 pic., 7 tables, 3 app., 19 sources.

The object of development: A web-oriented application for selling AI-generated products.

Purpose of the Qualification Work: To develop a web-oriented application for selling AI-generated products using the technological stack of HTML/CSS/Python/Django framework.

The introduction discusses the necessity and demand for online stores, as well as the importance of their rapid creation, easy management, and administration, which will contribute to the website's popularity growth.

The first chapter describes the advantages of using the Shopify platform to create an online store, the essential functions and aspects that a web-oriented application should have, and the principles of their operation.

The second chapter examines the application's functionality, the creation of the visual part, and the process of creating a service for importing products.

The economic chapter evaluates the project's complexity, determines the costs of developing the web platform considering various factors (number of lines of code, resources used, electricity consumption, Internet, etc.), and calculates the necessary time for its implementation.

The practical significance of the work lies in creating a platform for selling AI-generated products, which can be organized in a separate file and imported to the website, where it can be conveniently managed using the Shopify platform.

The relevance of the platform is driven by the modern trend of people preferring to shop from home and the encouragement to use artificial intelligence as an assistant and tool to improve life. List of keywords: WEB-ORIENTATION, HTML, CSS, PYTHON, DJANGO, FRAMEWORK, WEBSITE, WEB PLATFORM, ARTIFICIAL INTELLIGENCE

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

API – Application Programming Interface

БД – База даних

ОС – Операційна система

SQL – Structured Query Language

RAM – Random-Access Memory

HTTPS - Hypertext Transfer Protocol Secure

HTTP - Hypertext Transfer Protocol

JSON - JavaScript Object Notation

DOM - Document Object Model

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ ..	8
1.1. Загальні відомості з предметної галузі.....	8
1.2. Призначення розробки та галузь застосування.....	9
1.3. Підстави для розробки.....	9
1.4. Постановка завдання.....	10
1.5. Вимоги до програми або програмного виробу	12
1.5.1 Вимоги до функціональних характеристик	12
1.5.2 Вимоги до інформаційної безпеки	12
1.5.3 Вимоги до складу та параметрів технічних засобів	13
1.5.4 Вимоги до інформаційної та програмної сумісності	14
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ	15
2.1. Функціональне призначення системи.....	15
2.2. Опис застосованих математичних методів.....	15
2.3. Опис використаних технологій та мов програмування	16
2.4. Опис структури системи та алгоритмів її функціонування.....	24
2.5. Обґрунтування та організація вхідних та вихідних даних програми	33
2.6. Опис розробленої системи	34
2.6.1. Використані технічні засоби.....	34
2.6.2. Використані програмні засоби	35
2.6.3. Виклик та завантаження програми	35
2.6.4. Опис інтерфейсу користувача	35
РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ	50
3.1. Розрахунок трудомісткості розробки програмного забезпечення	50
3.2. Розрахунок витрат на створення програмного забезпечення.....	55
ВИСНОВКИ.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
ДОДАТОК А КОД ПРОГРАМИ.....	61
ДОДАТОК Б ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ	74
ДОДАТОК В ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ	755

ВСТУП

Вебсайти для сучасних магазинів – зручний спосіб для залучення більшої кількості клієнтів, що приводить до збільшення продажів та зростання магазину як бізнесу та комерційної одиниці. Під час нелегких часів пандемії COVID-19 у світі відбулося багато перетворень, людям треба було пристосуватись до реалій сьогодення. Тому виникає потреба в створенні якісних та легких в опрацюванні вебмагазинів за допомогою платформи Shopify, яка окрім того, що надає можливість повністю кастомізувати вебсайт під свої вимоги, а й дозволяє адміністратору вести облік товарів, покупців, бачити комплексну аналітику вебмагазину.

Shopify надає багато інструментів для побудови та управління інтернет-магазином. Зокрема, система дозволяє створювати унікальні дизайни вебсайтів без необхідності володіння глибокими знаннями програмування. Крім того, Shopify підтримує інтеграцію з різними платіжними системами, що значно спрощує процес покупки для клієнтів. Ця платформа також забезпечує високу безпеку даних, що є критично важливим аспектом для будь-якого онлайн-бізнесу.

Мета кваліфікаційної роботи: розробити веборієнтований додаток, який містить в собі вебсайт та сервіс імпортування товарів для продажу. Цей додаток буде включати функціонал для автоматичного завантаження товарів з інших платформ, що дозволить значно зекономити час та зусилля при веденні магазину. Окрім цього, додаток матиме зручний інтерфейс для адміністраторів, що дозволить легко керувати товарами, відстежувати продажі та аналізувати поведінку покупців. Таким чином, створення такого додатку сприятиме розвитку онлайн-комерції та допоможе підприємцям ефективніше управляти своїм бізнесом в інтернеті.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

У 21 столітті онлайн-торгівля набирає все більшої популярності, створюючи можливість продажу товарів будь-якого напрямку. З урахуванням сучасної динаміки життя люди все частіше обирають зручність покупок з власного дому або будь-якого іншого місця, що дозволяє їм економити час, раніше витрачений на дорогу до магазину.

Попит на фахівців, які здатні розробити та підтримувати такі інтернет-магазини, постійно зростає, оскільки збільшується кількість підприємств, які бажають привернути увагу клієнтів у віртуальному просторі. Це спонукає бізнес до пошуку способів ефективної конкуренції серед різноманітних конкурентів.

Проте для малих стартапів та підприємств з обмеженим бюджетом на розробку інтернет-магазину часто важко знайти достатньо ресурсів для створення сучасного та конкурентоспроможного проєкту.

Це саме розкриває потребу в доступних та ефективних інструментах для створення та управління електронними магазинами. Одним із таких інструментів є платформа Shopify, яка надає можливість створювати професійні та зручні у використанні інтернет-магазини з мінімальними витратами зусиль і коштів.

Shopify забезпечує широкий набір функцій, включаючи готові до використання шаблони дизайну, інтеграцію з різноманітними платіжними системами, інструменти для створення та управління запасами, а також аналітичні засоби для вивчення поведінки покупців. Використовуючи цю платформу, малі бізнеси можуть швидко запустити свій інтернет-магазин і почати конкурувати з більшими гравцями на ринку, не маючи значних витрат на розробку та підтримку вебпроєкту.

1.2. Призначення розробки та галузь застосування

В якості об'єкту розробки розглядається веборієнтований додаток, завдяки якому розробник може побудувати інтернет-магазин, інтегрувати додаткові необхідні сервіси та плагіни, а також використовувати платформу Shopify як систему керування вмістом, що дозволяє зручно обробляти замовлення.

Даний додаток повинен містити в собі сторінку з категоріями товарів, меню для швидкого доступу до сайту, сторінку товару з можливістю ознайомитись з характеристиками та додати товар до кошика, сторінку кошику, з якої користувач може придбати товар, зручну систему пошуку товарів по сайту, чат підтримки для зв'язка між магазином та користувачем, систему відгуків на товар для будівництва довіри до магазину, сторінки зі статтями блогу, в яких клієнт може дізнатись більше інформації про магазин та товару, мапа сайту.

Оскільки цей сервіс потрібен переважно для того, щоб швидко та зручно запустити свій онлайн-магазин, нам потрібно додати додаткову можливість створювати товари, а саме розробити систему імпорту, яка буде аналізувати .xls файл та автоматично заповнювати необхідні поля та метафілди в панелі адміністратора.

Крім того, важливою частиною розробки є забезпечення максимальної швидкості та ефективності роботи сайту. Оптимізація завантаження сторінок, використання кешування та інші техніки прискорення допоможуть забезпечити безперебійну роботу магазину навіть при великому потоці відвідувачів.

У зв'язку з постійним розвитком електронної комерції та змінами в потребах користувачів, важливо забезпечити масштабованість та гнучкість розроблюваного додатку. Постійне оновлення та підтримка забезпечать його актуальність та конкурентоспроможність на ринку електронної торгівлі.

1.3. Підстави для розробки

Підставами для розробки та виконання кваліфікаційної роботи) є:

- освітня програма 122 Комп'ютерні науки;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» № 469-с від 23.05.2024 р;
- завдання на кваліфікаційну роботу на тему «Розробка веборієнтованого додатку для продажу товарів, згенерованих штучним інтелектом на базі платформи Shopify».

1.4. Постановка завдання

Завдання: розробка веборієнтованого додатку для продажу товарів, згенерованих штучним інтелектом на базі платформи Shopify з використанням технологічного стеку HTML/CSS/Python/Django framework.

Цілі створення веборієнтованого додатку:

- покращення клієнтського досвіду шляхом впровадження інтуїтивно зрозумілого та привабливого інтерфейсу;
- створення конкурентоспроможного продукту, який сприяє зростанню бізнесу на ринку електронної комерції;
- впровадження нового погляду на створення веборієнтованих додатків за допомогою використання платформ для ведення бізнесу.

Веборієнтований додаток має відповідати таким вимогам:

- зручне користування сайтом для нового користувача;
- швидке завантаження сторінок;
- зрозумілість та візуальна привабливість.

Функціонал:

У табл. 1.1 наведено дозвіл користувачів (об'єктів) до певного функціоналу веборієнтованого додатку.

Також передбачається, що платформа налічуватиме певні об'єкти з певними властивостями (наведено у табл. 1.2):

Таблиця 1.1

Можливості користувачів (об'єктів) при роботі з вебплатформою.

Об'єкт	Можливості
Адміністратор	вхід до системи Shopify вхід до системи Django редагування документу та імпорт цього документу на сервіс редагування візуальної частини застосунку
Клієнт	перегляд товарів в застосунку користування пошуковою системою сервісу додавання товару в кошик придбання товару перегляд блогових статей

Таблиця 1.2

Об'єкти платформи та їх властивості

Об'єкт	Властивості
Користувач (клієнт, адміністратор)	електронна пошта
Товар	назва ціна зображення колір опис виробник відгуки
Колекція	назва товари сортування
Чат підтримки	назва поле для вводу тексту відображення повідомлень

Питання та відповіді	текстовий блок випадаючий список
Банери	зображення перемикання зображень
Меню	колекції лендінги

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Веборієнтований додаток для продажу товарів повинен бути реалізований на платформі Shopify за допомогою HTML та CSS; система керування обліковими даними - мовою програмування Python за допомогою фреймворку Django. Перейти до цих застосунків можна за допомогою браузера (Chrome, Mozilla, Microsoft Edge, Opera, Internet Explorer та ін.).

Для написання платформи передбачається використання наступних технологій: фреймворк Django, HTML, CSS.

Інформація, яка пов'язана з даними про товари, що розміщені на сайті, повинна знаходитися в базі даних PostgreSQL на самому сервері.

Веборієнтований додаток повинен давати можливість користувачам придбати товари, адміністраторам – вносити нові та редагувати вже завантажені товари.

1.5.2. Вимоги до інформаційної безпеки

Стандарт безпеки даних платіжних карток (PCI DSS) - це стандарт безпеки для організацій, які обробляють інформацію про кредитні та дебетові картки. Стандарт був створений для покращення управління платіжними даними та запобігання шахрайству.

За замовчуванням усі магазини Shopify, які використовують нашу платформу, автоматично відповідають вимогам PCI. Якщо користувач обирає Shopify для управління своїм магазином, то він може бути впевнений, що додаток отримав сертифікат PCI Level 1 і що цей сертифікат поширюється на ваш магазин, кошик для покупок та вебхостинг.

Відповідність стандарту PCI DSS приносить низку переваг, зокрема:

- Покращена безпека: відповідність PCI DSS сприяє безпечній обробці конфіденційних даних платіжних карток, знижуючи ризик витоку даних і шахрайства.
- Збільшення довіри клієнтів: демонстрація відповідності PCI DSS підвищує довіру клієнтів та впевненість у безпеці організації.
- Зменшення витрат: впровадження засобів контролю PCI DSS дозволяє зменшити ризик порушення безпеки та пов'язані з цим витрати, такі як штрафи, судові витрати та витрати на сповіщення клієнтів.
- Конкурентна перевага: відповідність PCI DSS може надати організації конкурентну перевагу, демонструючи прихильність до безпеки, що виділяє її серед конкурентів.

1.5.3. Вимоги до складу та параметрів технічних засобів

Щоб мати змогу користуватися додатком, треба обрати будь-який з сучасних пристроїв: ноутбук, планшет, персональний комп'ютер, смартфон. Мінімальні необхідні характеристики для доступу до сайту:

- версія браузера Chrome 81.0.4044.20 або вище;
- підтримка 32- або 64-розрядного процесора та операційної системи;
- операційна система Windows 7 чи вище;
- наявність маніпулятора-миші/тачпаду та клавіатури;
- оперативна пам'ять об'ємом 2 ГБ або вище
- процесор із частотою 1 ГГц або вище;
- жорсткий диск HDD чи SSD об'ємом 10 ГБ або вище.

1.5.4. Вимоги до інформаційної та програмної сумісності

Для ефективної роботи програми необхідно, щоб програмне забезпечення працювало на різних пристроях, таких як ноутбук, планшет, комп'ютер або телефон, і відповідало таким вимогам:

- Операційна система: Windows (7+) / Linux / MacOS / Android.
- Веббраузер: Google Chrome / Opera / Safari / Firefox / Microsoft Edge.

Застосунок має бути реалізовано за допомогою платформи Shopify, використовуючи HTML, CSS, мову програмування Python з фреймворком Django та базою даних PostgreSQL.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення системи

Даний веборієнтований додаток розробляється з переліком певних вимог, які він повинен виконувати:

- Доступ до сайту з будь-яких сучасних пристроїв та платформ, щоб кожна людина з доступом до Інтернету мала змогу відвідати онлайн-магазин
- Система якісних та своєчасних оновлень контенту, що охоплює в собі як оновлення блогових статей та візуального оформлення сайту, так і імпортування нових товарів на сайт
- Створення конкурентно спроможного додатку, який буде заохочувати потенційних покупців придбати товар саме в цьому онлайн-магазині
- Неординарний каталог товарів, який виводить всю онлайн торгівлю на новий рівень
- Веборієнтований додаток реалізує такі функції:
 - Зберігання товарів по колекціям, сортування товарів
 - Зручний та швидкий пошук товарів
 - Можливість додати товари в кошик та оплатити ці товари
 - Імпорт даних товарів з xlsx файлу на платформу Shopify

2.2. Опис застосованих математичних методів

Розроблений веборієнтований додаток складається з двох частин: безпосередньо вебсайт та додаток для імпортування товарів. Для створення вебсайту було використано знання з HTML та CSS. Для імпортування товарів з документу було розроблено додаток, який використовує Shopify API для запису metafields товару із Excel файлу.

Через це, даний веборієнтований додаток не має в собі потреби вирішувати математичні завдання, обчислювати математичні формули тощо.

2.3. Опис використаних технологій та мов програмування

Розроблений веборієнтований додаток для продажу товарів, згенерованих штучним інтелектом було реалізовано за допомогою HTML, CSS, а також Python з використанням фреймворка Django.

HTML (HyperText Markup Language) є основною мовою розмітки, яка використовується для створення та структурування вебсторінок. Вона є фундаментальною частиною веброзробки і відіграє ключову роль у формуванні контенту на вебсайтах. Завдяки HTML, розробники можуть організовувати контент за допомогою різноманітних тегів і атрибутів, які визначають текстові блоки, зображення, гіперпосилання та інші елементи вебсторінки. HTML дозволяє задавати заголовки, абзаци, списки, таблиці, форми та багато інших компонентів, що забезпечують змістовність і функціональність вебсторінок. Ця мова розмітки є основою для побудови будь-якого вебсайту, на яку накладаються стилі за допомогою CSS і функціональність за допомогою JavaScript.

HTML є надзвичайно важливим інструментом для розробників, оскільки без нього неможливо створити структуровану і добре організовану вебсторінку. Розглянемо докладніше деякі основні аспекти HTML та їх важливість у процесі веброзробки.

Переваги використання HTML:

- Простота використання - HTML легко вивчити та використовувати навіть для початківців. Основні теги та синтаксис зрозумілі та логічні, що робить процес створення вебсторінок доступним для широкого кола людей.
- Універсальність - HTML є універсальною мовою розмітки, яка підтримується всіма сучасними веббраузерами. Це означає, що

вебсторінки, створені на HTML, можуть бути переглянуті на будь-якому пристрої з доступом до Інтернету.

- Сумісність з іншими технологіями - HTML чудово інтегрується з іншими вебтехнологіями, такими як CSS і JavaScript. CSS дозволяє стилізувати HTML-контент, а JavaScript додає інтерактивність та динамічність.
- Пошукова оптимізація (SEO) - правильно структуровані HTML-документи сприяють кращому індексуванню та ранжуванню вебсторінок у пошукових системах. Використання семантичних тегів допомагає пошуковим системам краще розуміти зміст сторінки.
- Адаптивність - HTML дозволяє створювати адаптивні вебсторінки, які автоматично підлаштовуються під розмір екрану пристрою, будь то комп'ютер, планшет або смартфон. Це досягається за допомогою медіа-запитів та адаптивного дизайну.
- Можливість розширення - HTML5, остання версія мови, додала багато нових можливостей, включаючи підтримку мультимедіа, нові семантичні елементи та API для розширення функціональності вебдодатків.
- Відкритість та стандартизація - HTML є відкритим стандартом, який розробляється та підтримується консорціумом W3C. Це забезпечує прозорість і доступність мови для всіх користувачів та розробників.

CSS (аббревіатура від Cascading Style Sheets, що означає каскадні таблиці стилів) — це спеціальна мова стилів, яка використовується для визначення вигляду документів (де і як повинні відображатися елементи веб-сторінки), створених за допомогою мов розмітки. Найчастіше CSS застосовується до документів, розмічених мовами HTML, XHTML та XML.

В сучасних умовах HTML сам по собі має обмежений набір інструментів, що не дозволяє повною мірою реалізувати дизайнерські та функціональні ідеї веб-розробників. У таких випадках на допомогу приходить CSS, який вирішує більшість питань, пов'язаних з оформленням сторінки.

Одна з ключових переваг CSS полягає в можливості розділити зміст сторінки від її оформлення. Це розділення сприяло покращенню доступності та сприйняття змісту, дозволило забезпечити більшу гнучкість та контроль за відображенням у різних умовах, зробило зміст більш структурованим і зрозумілим, а також усунуло повторення. Саме це було основною метою створення CSS.

Що надає використання CSS:

- Відображення одного і того ж документа в різних стилях.
- Різноманітні дизайни сторінки для різних пристроїв - наприклад, на екрані дизайн може бути адаптованим для великої ширини, тоді як при друку меню не буде відображатися, а на смартфоні меню буде знаходитись під вмістом.
- Зменшення часу завантаження сторінок - завдяки перенесенню стилів в окремий CSS-файл, браузер завантажує лише структуру документа і зміст, а правила стилів завантажуються один раз і кешуються.
- Легкість у зміні дизайну - замість редагування кожної сторінки, достатньо змінити лише CSS-файл.
- Додаткові можливості оформлення - наприклад, за допомогою CSS можна зробити так, щоб меню залишалось видимим під час прокручування сторінки, або прибрати підкреслення у посилань.
- Створення складних і продуманих дизайнів.

Правила CSS працюють за принципом каскаду (пріоритету та ваги), що дозволяє отримати передбачуваний результат, коли до одного елемента застосовуються кілька стилів одночасно.

Python - це високорівнева мова програмування, яка є відомою своєю простотою, а також легким рівнем входу. Вона широко використовується в різних галузях, включаючи веброзробку, аналіз даних, штучний інтелект та автоматизацію завдань.

Відповідно до міжнародних рейтингів 2021-2022 років, мова програмування Python займає провідні позиції:

- PyPL, створеного на основі даних з Github;
- TIOBE, який визначає рейтинг за згадками на веб-сторінках;
- IEEE Spectrum — рейтинг відкритих проєктів, враховуючи частоту згадування у вакансіях.

У рейтингу популярності мов програмування серед програмістів, які працюють в Україні, Python посідає 4 місце з часткою програмістів 13%, демонструючи позитивну динаміку протягом останніх двох років.

Основні переваги Python:

- Python має простий синтаксис і структуру, що робить його легко читабельним і зрозумілим, ідеальним для початківців. Програми на Python коротші і швидші. Можна одразу написати і протестувати код без значних витрат часу.
- Python є безкоштовним, що дозволяє використовувати його у будь-яких проєктах без обмежень чи витрат. Його можна завантажити та одразу почати працювати. Крім того, Python має відкритий код, що дозволяє вносити зміни і коригування.
- Python має велику і активну спільноту розробників, яка постійно вдосконалює мову, створює нові бібліотеки та інструменти. Ця спільнота допомагає розробникам Python знаходити відповіді на будь-які питання і ділитися досвідом та знаннями.
- Python працює на будь-якій операційній системі, включаючи Windows, Mac OS та Linux. Це зробило Python дуже популярним у розробці програмного забезпечення та веб-додатків. Для роботи на Windows необхідно встановити безкоштовний дистрибутив Python Anaconda, а на інших ОС достатньо ввести "python" в терміналі.
- Python має багато бібліотек та інструментів, що дозволяють створювати різноманітні проєкти. Завдяки бібліотекам, таким як NumPy, Pandas, SciPy

та іншим, Python став дуже популярним у сфері аналізу даних та машинного навчання. Крім того, Python можна використовувати для розробки веб-додатків, графічного дизайну, наукових досліджень та інших сфер.

Основні сфери застосування Python:

- Автоматизація - незалежно від професії, якщо ви часто зіштовхуєтесь з процесами, які можна автоматизувати, Python ідеально підходить для цього. Можна просто написати скрипт або знайти відповідну бібліотеку.
- Веброботка - різні фреймворки, такі як Django, Flask та Pyramid, дозволяють швидко створити базовий бекенд. Можна працювати з базами даних, налаштовувати URL-адреси, відображати зображення на різних пристроях. Flask підходить для новачків завдяки простій структурі.
- Штучний інтелект - Python активно використовується у розробці технологій штучного інтелекту.
- Створення ігор - Python має багато бібліотек для створення ігор на різних платформах. Можна створювати прості ігри з використанням Pygame або використовувати Panda3D для складніших проєктів.
- Data Science - Python дозволяє проводити аналіз, обробку та візуалізацію великих обсягів даних. Бібліотеки, такі як NumPy, Pandas та Matplotlib, дозволяють легко аналізувати великі набори даних та візуалізувати їх. Вивчення основ NumPy та Pandas допоможе зрозуміти, як працювати з даними, а Matplotlib дозволить створювати візуалізації.
- Наукові дослідження - Python популярний у наукових дослідженнях, особливо у природничих науках та інженерії. SciPy, SymPy та Matplotlib — це лише деякі з бібліотек, що широко використовуються у цій сфері.

Django — це веб-фреймворк для Python, який пропонує стандартні методи для швидкої та ефективної розробки веб-сайтів. Основна мета цього високорівневого веб-фреймворку — створення комплексних веб-сайтів, які керуються базами даних. Він допомагає створювати та підтримувати якісні веб-

додатки, роблячи процес розробки плавним та економлячи час для швидкої розробки.

Основні переваги Django:

- Django легко налаштувати та запустити. Він пропонує багато варіантів для початку роботи.
- Забезпечує готовий до використання користувацький інтерфейс для адміністративних дій.
- Дозволяє створювати багатомовні веб-сайти завдяки вбудованій системі інтернаціоналізації.
- Допомагає швидко задовольняти потреби в високому трафіку.
- Використовується для створення різних типів систем управління контентом, соціальних мереж, а також платформ наукових обчислень.
- Допомагає забезпечити комплексне тестування додатків.
- Дозволяє документувати ваш API з допомогою виводу HTML.
- REST Framework підтримує кілька протоколів аутентифікації.
- Можливість обмеження швидкості запитів API від одного користувача.

Особливості Django:

- Допомагає визначати шаблони для URL-адрес у вашому додатку.
- Проста, але потужна система URL-адрес.
- Вбудована система аутентифікації.
- Об'єктно-орієнтована база даних забезпечує найкраще в своєму класі зберігання та отримання даних.
- Функція автоматичного адміністративного інтерфейсу дозволяє додавати, редагувати та видаляти елементи. Можна налаштувати панель адміністратора на свій розсуд.
- Платформа кешування з декількома механізмами кешування.

Основні характеристики Django:

- Слабозв'язаний — Django допомагає робити кожен елемент свого стека незалежним від інших.

- Мінімальна кількість коду забезпечує ефективну розробку.
- Не повторюваність. Все має бути розроблено в одному місці та не повторюватися знову.
- Швидка розробка. Django пропонує швидку та надійну розробку додатків.
- Послідовний дизайн. Django підтримує чистий дизайн та дозволяє легко дотримуватись найкращих практик веб-розробки.
- Надається діаграма (рис. 2.1.), яка показує, як взаємодіють компоненти шаблону для того, щоб обробити запит адміністратора під час імпортування

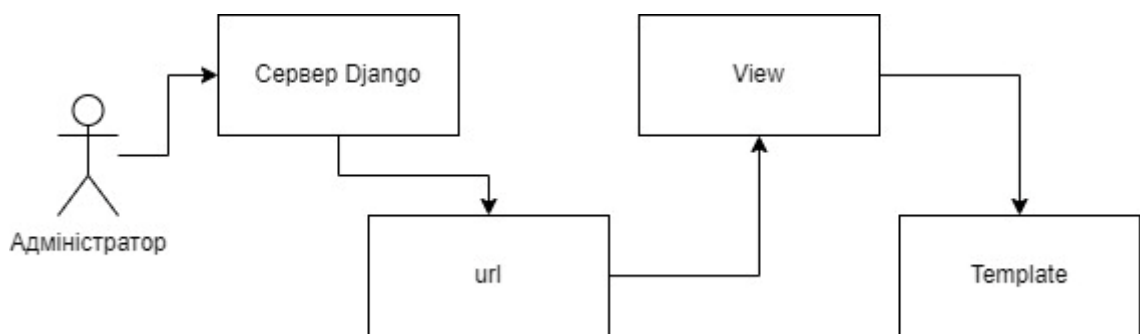


Рис. 2.1. Взаємодія компонентів шаблону, щоб обробити запит адміністратора під час імпортування

Для використання API Shopify для імпорту та експорту даних із магазину, треба виконати наступні загальні кроки:

- Налаштувати доступ до API:
 - Необхідно переконатися, що у вас є необхідні дозволи та облікові дані для доступу до API Shopify. Нам потрібен ключ API, секретний ключ API та, можливо, токен доступу залежно від API-ендпоінта, до якого ми звертаємось.
 - Згенерувати облікові дані API в адміністративній панелі Shopify у розділі «Додатки» > «Керування приватними додатками» або «Керування додатками» залежно від вашого налаштування.
- Вибрати API-ендпоінт:

- Треба визначити, який API-ендпоінт(и) потрібно використовувати для імпорту/експорту даних. API Shopify пропонує різні ендпоінти для різних типів даних, таких як продукти, клієнти, замовлення, колекції тощо.

- Ознайомитися з документацією API Shopify для списку доступних ендпоінтів та їх функціональностей.

- Розробити або використовувати існуючий клієнт API:

- Ми маємо можливість розробити власний клієнт API, використовуючи мови програмування, такі як Python, Node.js, PHP тощо, або скористатися існуючими бібліотеками та інструментами клієнтів API для Shopify.

- Популярні бібліотеки клієнтів API для Shopify включають shopify-api-node (Node.js), shopify_python_api (Python), shopify_api (Ruby) та інші.

- Встановити та налаштувати бібліотеку клієнта API у своєму середовищі розробки.

- Автентифікувати та робити запити до API:

- Автентифікуйте ваш клієнт API, використовуючи ключ API, секретний ключ API та токен доступу, отримані з адміністративної панелі Shopify.

- Використовувати клієнт API для надсилання запитів до відповідних API-ендпоінтів для імпорту/експорту даних.

- Приклади API-ендпоінтів, які можна використовувати:

- GET /admin/api/2023-10/products.json: Отримати дані про продукти.

- POST /admin/api/2023-10/products.json: Створити нові продукти.

- PUT /admin/api/2023-10/products/{product_id}.json: Оновити існуючі продукти.

- GET /admin/api/2023-10/customers.json: Отримати дані про клієнтів.

- POST /admin/api/2023-10/customers.json: Створити нових клієнтів.

- GET /admin/api/2023-10/orders.json: Отримати дані про замовлення.

- POST /admin/api/2023-10/orders.json: Створити нові замовлення.

2.4. Опис структури системи та алгоритмів її функціонування

У подробицях структуру проєкту можна описати таким чином:

- Frontend (Shopify):
 - Теми – Shopify використовує теми для візуального оформлення магазину. Теми складаються з шаблонів, CSS, JavaScript та інших ресурсів.
 - Шаблони – Основні шаблони для сторінок продуктів, колекцій, кошика, оформлення замовлення тощо.
 - Секції – Модульні компоненти для створення кастомних сторінок та налаштування тем.
 - Скрипти – JavaScript файли для динамічної взаємодії з користувачем, наприклад, AJAX-запити для додавання товарів у кошик без перезавантаження сторінки.
- Backend (Django):
 - account – Django-додаток, який відповідає за логіку користувача, автентифікацію, реєстрацію, управління профілями.
 - bin – .sh-файли, які використовуються в production розгортанні проєкту на сервері.
 - products – Django-додаток для управління товарами, їх імпортом та експортом до Shopify.
 - shopify_integration –Додаток для інтеграції з API Shopify, включає моделі, сервіси та задачі для імпорту/експорту даних.
 - settings.py – Основний файл налаштувань проєкту, де визначені ключові параметри, такі як бази даних, встановлені додатки, middleware та інші налаштування.
 - urls.py – Файл маршрутизації, де визначені всі шляхи до різних частин вебдодатку.
 - views.py – Логіка обробки запитів, повернення відповідей, включаючи рендеринг шаблонів та обробку даних з форм.

Алгоритм функціонування платформи:

– Реєстрація та автентифікація користувача:

○ Користувач реєструється або входить у систему через форму, розміщену на фронтенді (Shopify).

○ Дані з форми відправляються на бекенд (Django), де відбувається обробка запиту в додатку account.

○ Після успішної автентифікації, користувач отримує доступ до свого профілю та інших захищених частин сайту.

– Імпорт товарів на Shopify:

○ Адміністратор або автоматична задача запускає процес імпорту товарів у Django-додатку products.

○ Django-додаток `shopify_integration` використовує API Shopify для створення або оновлення товарів у магазині Shopify.

○ Дані про товари отримуються з локальної бази даних або зовнішніх джерел і формуються у відповідні запити до API Shopify.

○ За допомогою методів, таких як `POST /admin/api/2023-10/products.json` та `PUT /admin/api/2023-10/products/{product_id}.json`, товари імпортуються або оновлюються в магазині Shopify.

– Управління товарами:

○ Адміністратор може додавати, редагувати або видаляти товари через адміністративну панель Django.

○ Зміни в товарах зберігаються в локальній базі даних та синхронізуються з магазином Shopify через `shopify_integration`.

– Відображення товарів на фронтенді:

○ Товари, які були імпортовані або оновлені на Shopify, автоматично відображаються на фронтенді завдяки синхронізації даних.

○ Користувачі можуть переглядати, шукати та купувати товари через інтерфейс Shopify, який інтегрується з бекендом для обробки даних та транзакцій.

– Обробка замовлень:

- Коли користувач оформляє замовлення, дані передаються на бекенд для подальшої обробки.

- Django-додаток може обробляти замовлення, зберігати їх у базі даних, та відправляти підтвердження користувачу.

- Інформація про замовлення також може бути відправлена до Shopify для синхронізації стану замовлень.

Ця структура та алгоритми забезпечують ефективну роботу вебплатформи на Shopify з використанням Django для бекенду, що дозволяє легко імпортувати та експортувати товари, управляти даними та забезпечувати безперебійну роботу онлайн-магазину.

На проєкті, де дані з Excel файлу передаються на сервера Shopify для створення процесу імпортування даних про товари, парсинг відіграє ключову роль. У цьому процесі метафілди з Excel файлу використовуються для забезпечення правильного розміщення даних у системі Shopify.

Першим кроком є отримання Excel файлу, що містить дані про товари. Цей файл зазвичай містить інформацію про назви товарів, описи, ціни, кількість на складі, зображення та інші атрибути. Важливим аспектом є наявність метафілдів, які визначають, куди саме повинні бути занесені ці дані на платформі Shopify.

Далі відбувається процес парсингу. Спочатку Excel файл завантажується у програму або скрипт, написаний для цієї мети. Цей скрипт зчитує дані з файлу та аналізує їх, використовуючи метафілди для розуміння структури та призначення кожного поля. Метафілди є ключовими, оскільки вони допомагають визначити, які дані відповідають певним полям у базі даних Shopify.

Після аналізу даних відбувається їх передача на сервери Shopify за допомогою API. API Shopify дозволяє інтегрувати зовнішні програми та обмінюватися даними з платформою. Кожен товар з Excel файлу перетворюється у формат, зрозумілий для API, та відправляється на сервер. Тут важливо, щоб

кожне поле було правильно заповнене згідно з вимогами Shopify, що забезпечується метафілдами.

На сервері Shopify отримані дані обробляються та зберігаються у відповідних базах даних. В результаті, товари з'являються у панелі адміністрування Shopify, де їх можна редагувати, управляти їхніми властивостями та здійснювати інші операції.

Парсинг – це процес аналізу та обробки структурованих даних з метою їхнього розпізнавання та перетворення у формат, придатний для подальшого використання. Цей процес часто використовується у програмуванні для роботи з текстовими файлами, базами даних та іншими форматами даних. Основною метою парсингу є витягування необхідної інформації з даних та її впорядкування відповідно до заданих правил.

Парсинг може бути застосований у різних контекстах, включаючи веб-скрапінг, аналіз файлів журналу, обробку природної мови та багато іншого. У веб-розробці парсинг часто використовується для зчитування HTML-коду вебсторінок та витягування з нього конкретних даних, таких як заголовки, посилання або зображення.

У контексті роботи з файлами, такими як Excel, парсинг передбачає зчитування даних з файлів, розпізнавання їхньої структури та переведення у формат, придатний для подальшої обробки. Наприклад, при імпорті даних з Excel до бази даних, парсинг використовується для зчитування даних з таблиць Excel та збереження їх у відповідних таблицях бази даних.

Shopify – це платформа електронної комерції, яка дозволяє створювати та управляти інтернет-магазинами. Вона забезпечує користувачів інструментами для побудови вебсайту, управління товарами, обробки платежів та виконання інших завдань, пов'язаних з електронною комерцією. Shopify є однією з найпопулярніших платформ для створення інтернет-магазинів завдяки своїй зручності та широкому спектру можливостей.

Однією з ключових переваг Shopify є її гнучкість та можливість інтеграції з іншими сервісами та додатками через API. Це дозволяє розробникам створювати власні рішення для автоматизації бізнес-процесів, інтеграції з зовнішніми системами та розширення функціональності магазину.

Shopify пропонує широкий вибір шаблонів та тем, що дозволяють швидко створити привабливий дизайн магазину без необхідності володіння знаннями у вебдизайні. Крім того, платформа підтримує різні методи оплати та служби доставки, що робить її універсальним рішенням для бізнесів будь-якого розміру.

Excel – це програмне забезпечення для роботи з електронними таблицями, розроблене компанією Microsoft. Воно є частиною пакету офісних програм Microsoft Office. Excel дозволяє користувачам створювати, редагувати та аналізувати таблиці, виконувати складні обчислення, будувати графіки та діаграми, а також автоматизувати задачі за допомогою макросів.

Однією з ключових функцій Excel є можливість обробки великих обсягів даних. Програма підтримує різні типи даних, включаючи текст, числа, дати та формули, що дозволяє користувачам виконувати аналіз даних, створювати звіти та моделі. Excel широко використовується в бізнесі, освіті, науці та інших галузях для роботи з даними.

Excel також підтримує імпорт та експорт даних у різних форматах, що дозволяє інтегрувати його з іншими програмами та системами. Це робить його незамінним інструментом для аналізу та обробки даних, а також для створення звітів та презентацій.

Парсинг має багато переваг, що робить його важливим інструментом у процесі роботи з даними:

- Автоматизація - парсинг дозволяє автоматизувати процес зчитування та обробки даних, що значно зменшує час та зусилля, необхідні для виконання цих завдань вручну.
- Точність - автоматичний парсинг мінімізує ризик людських помилок, забезпечуючи високу точність обробки даних.

- Ефективність - парсинг дозволяє обробляти великі обсяги даних швидше та ефективніше, ніж це можливо вручну.
 - Гнучкість - парсинг може бути налаштований для роботи з різними форматами даних та адаптований до специфічних потреб проекту.
- Shopify пропонує численні переваги для користувачів:
- Простота використання - інтуїтивний інтерфейс та зручні інструменти роблять Shopify доступним навіть для користувачів без технічних знань.
 - Гнучкість та масштабованість - Shopify підходить для бізнесів будь-якого розміру, від малих магазинів до великих підприємств.
 - Підтримка багатьох платіжних систем - Shopify інтегрується з різними платіжними системами, забезпечуючи зручність для клієнтів.
 - Широкий вибір додатків та інтеграцій - користувачі можуть розширювати функціональність свого магазину за допомогою численних додатків та інтеграцій з іншими сервісами.
 - Надійність та безпека - Shopify забезпечує високу надійність та безпеку роботи магазинів, включаючи захист даних клієнтів та безперебійну роботу сайту.

Excel має численні переваги, що робить його популярним серед користувачів:

1. Можливості аналізу даних - Excel пропонує потужні інструменти для аналізу даних, включаючи функції, формули, фільтри та зведені таблиці.

2. Гнучкість у роботі з даними - користувачі можуть створювати та редагувати таблиці, будувати графіки та діаграми, а також виконувати складні обчислення.

3. Інтеграція з іншими програмами - Excel підтримує імпорт та експорт даних у різних форматах, що дозволяє легко інтегрувати його з іншими програмами та системами.

4. Автоматизація задач - за допомогою макросів користувачі можуть автоматизувати рутинні задачі та зменшити час на їх виконання.

5. Універсальність використання - Excel використовується у різних галузях, включаючи бізнес, освіту, науку та дослідження, що робить його універсальним інструментом для роботи з даними.

Код програми міститься на сервері Shopify, через що розробник має можливість користуватися структурованою базою файлів для редагування та написання коду (рис. 2.1).

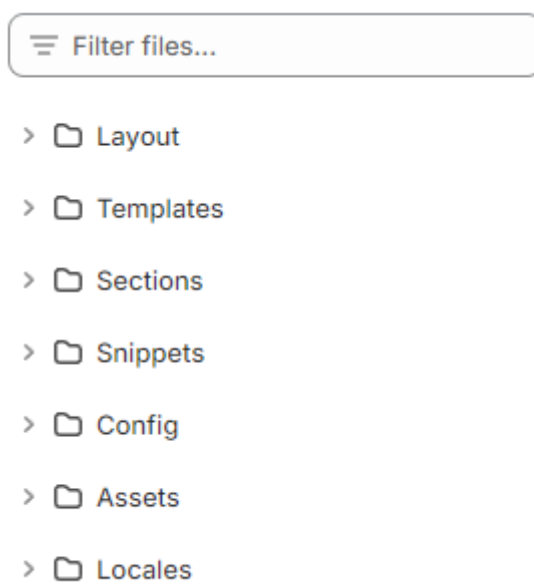


Рис. 2.1. Структура файлів

В кожній папці містяться файли, які відповідають за функціонування певних аспектів вебмагазину. Це дозволяє зручно знаходити шлях до потрібного файлу з повним розумінням, що треба відредагувати, бо назви файлів створюються на боці Shopify і відповідають саме за конкретний блок на сайті, наприклад: cart.js відповідає за кошик магазину (рис. 2.2). Така організація структури файлів спрощує процес розробки і підтримки вебмагазину, дозволяючи швидко знаходити потрібні компоненти для внесення змін і покращень.

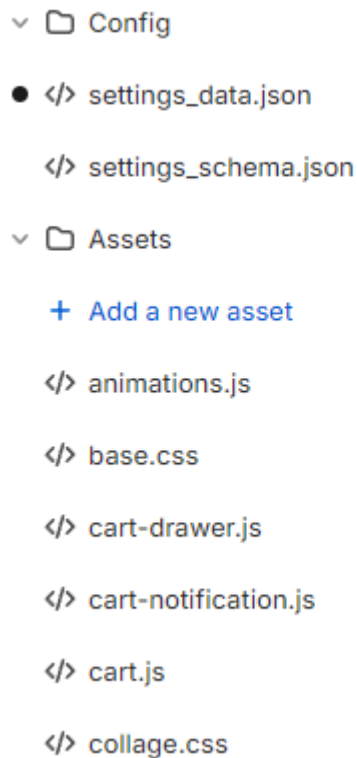


Рис. 2.2. Структура папок з файлами

Структура Shopify дозволяє редагувати та писати код не виходячи з панелі адміністратора сайту, при натисканні на певний файл відкриється текстовий редактор, в якому можна написати код, що буде відповідати за певний блок сайту (рис. 2.3). Текстовий редактор Shopify має схожу структуру, як і відомих застосунків для редагування коду. Розробник може табулювати код, код також автоматично підсвічує текст в залежності від його значення, розробник бачить кількість рядків в певному документі і має можливість вільно переміщуватись по рядками документу. Після внесення змін в коді треба натиснути на кнопку «Save» на тій самій сторінці, завдяки чому зміни в коді автоматично застосуються та будуть відображатися на вебсайті.

```
cart.js x
No recent changes
File does not have any other versions
1 class CartRemoveButton extends HTMLElement {
2   constructor() {
3     super();
4
5     this.addEventListener('click', (event) => {
6       event.preventDefault();
7       const cartItems = this.closest('cart-items') || this.closest('cart-drawer-items');
8       cartItems.updateQuantity(this.dataset.index, 0);
9     });
10  }
11 }
12
13 customElements.define('cart-remove-button', CartRemoveButton);
14
15 class CartItems extends HTMLElement {
16   constructor() {
17     super();
18     this.lineItemStatusElement =
19       document.getElementById('shopping-cart-line-item-status') || document.getElementById('CartDrawer-LineItemStatus');
20
21     const debouncedOnChange = debounce((event) => {
22       this.onChange(event);
23     }, ON_CHANGE_DEBOUNCE_TIMER);
24
25     this.addEventListener('change', debouncedOnChange.bind(this));
26   }
27 }
```

Рис. 2.3. Структура файлу з кодом

2.5. Обґрунтування та організація вхідних та вихідних даних програми

Даний веборієнтований додаток використовує систему імпортування товарів із Excel документу, які за допомогою Shopify API підтягуються в відповідні metafields. Ці дані мануально формуються в документі, після чого завантажуються на платформу.

Опис полів файлу для імпорту наведено в таблиці 2.1.

Опис полів таблиці Account

Назва поля	Тип даних поля	Ціль (значення)
sku	char	Артикул товару
title	char	Назва товару
description	char	Опис товару
price	integer	Вартість товару
vendor	char	Виробник товару
tag	char	Тег, за допомогою якого товар доєднується до колекції

2.6. Опис розробленої системи**2.6.1. Використані технічні засоби**

Для роботи веборієнтованого додатку не потрібні великі обчислювальні ресурси. Цю платформу було запущено на сервері з мінімальними характеристиками, які, незважаючи на свою обмеженість, все ж дозволяли обробляти певну кількість клієнтських запитів. Система, на якій проводилася розробка платформи, мала такі параметри:

- Операційна система: Windows;
- Процесор з тактовою частотою 1 ГГц або більше;
- Оперативна пам'ять (RAM) об'ємом від 1 ГБ;
- Жорсткий диск з об'ємом від 10 ГБ;
- Монітор, миша (тачпад) і клавіатура - для зручності розробки.

Попри невеликі технічні вимоги, ця конфігурація забезпечувала стабільну роботу і здатність відповідати на запити користувачів, що робило її досить ефективною для початкових етапів використання.

2.6.2. Використані програмні засоби

Під час створення платформи були використані наступні програмні засоби:

- браузер Chrome версії 125.0.6422.112;
- Microsoft Visual Studio Code;
- Shopify;

Для бекенду було використано мову Python версії 3.12, фреймворк Django версії 5.0.4.

Для фронтенду було використано HTML5 і каскадні таблиці стилів CSS3.

2.6.3. Виклик та завантаження програми

Увесь функціонал веб орієнтованого додатку міститься і працює завдяки серверам платформи Shopify, тому з боку розробника не відбувається ніякої дій для реалізації виклику та завантаження програми.

2.6.4. Опис інтерфейсу користувача

Щоб потрапити на онлайн-магазин, користувач повинен в будь-якому браузері відкрити пошукову строку, в яку треба буде ввести посилання на сайт: <https://3093c9-2b.myshopify.com/>. Після того, як в користувача завантажується сайт, він потрапляє на головну сторінку магазину. На цій сторінці (рис. 2.7) користувач може подивитись товари, почитати корисну інформацію про магазин, перейти на сторінку товару, відкрити пошукову строку, відкрити кошик, відкрити меню сайту, відкрити чат підтримки.

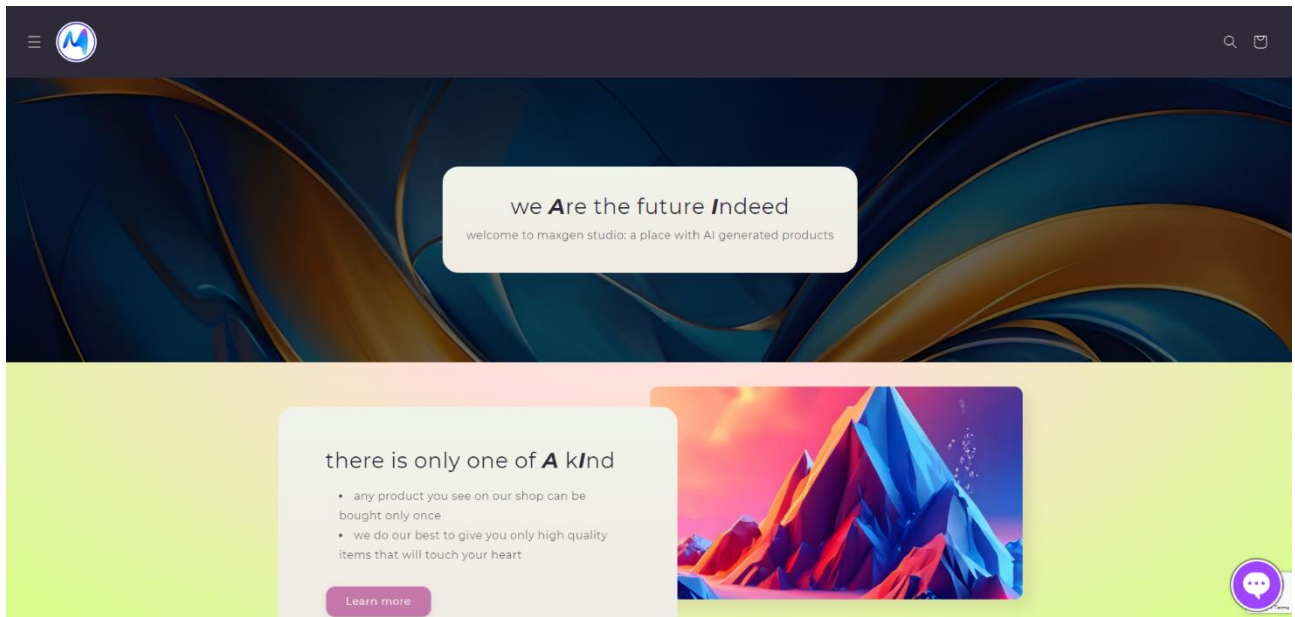


Рис. 2.7. Головна сторінка

На першому екрані завантажено банер з текстом, який позиціонує вагомість штучного інтелекту для цього магазину. Маркетингове рішення зробити літери А та І єдиними великими літерами на сайті повинно заохочити людей відвідати сайт. Для обґрунтування цього рішення було проведено опитування на 97 людей, студентів Національного технічного університету «Дніпровська політехніка» на платформі Telegram з приводу використання великих літер на сайтах (рис. 2.8). Більше 50 відсотків людей, які взяли участь в опитуванні обрали, що їм більше подобається, коли великі літери не використовуються в електронному вигляді (листування, онлайн-платформи, тощо).

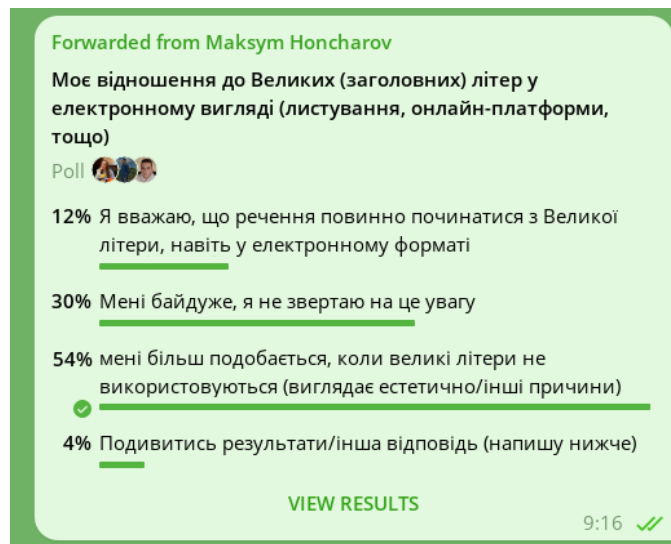


Рис. 2.8. Опитування з приводу використання великих літер в електронному вигляді

На другому екрані (рис. 2.9) сайту користувач бачить вибірку однієї з колекцій, обрати яку можна в налаштуваннях платформи Shopify. Також було налаштовано, що користувач може побачити перші 8 товарів з колекції, з яких перші 5 видно одразу, а інші появляються за допомогою використання спеціальних кнопок під товарами.

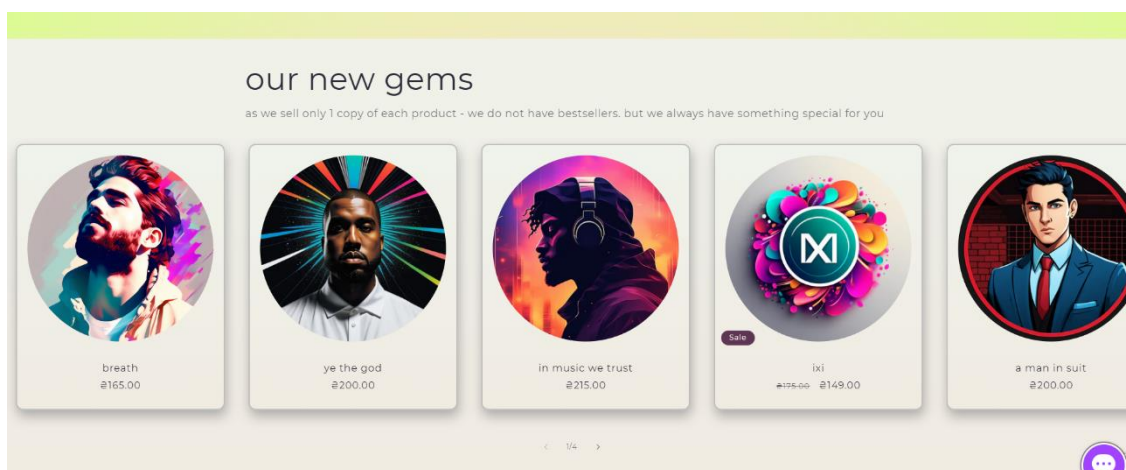


Рис. 2.9. Другий екран головної сторінки сайту

На третьому екрані (рис. 2.10) користувач може побачити розділ з банерами, який відрізняється від початкового тим, що їх можна перемикаати між собою. При натисканні кнопки під текстом користувач зможе зайти на сторінку колекції товарів, які вважаються новинками.

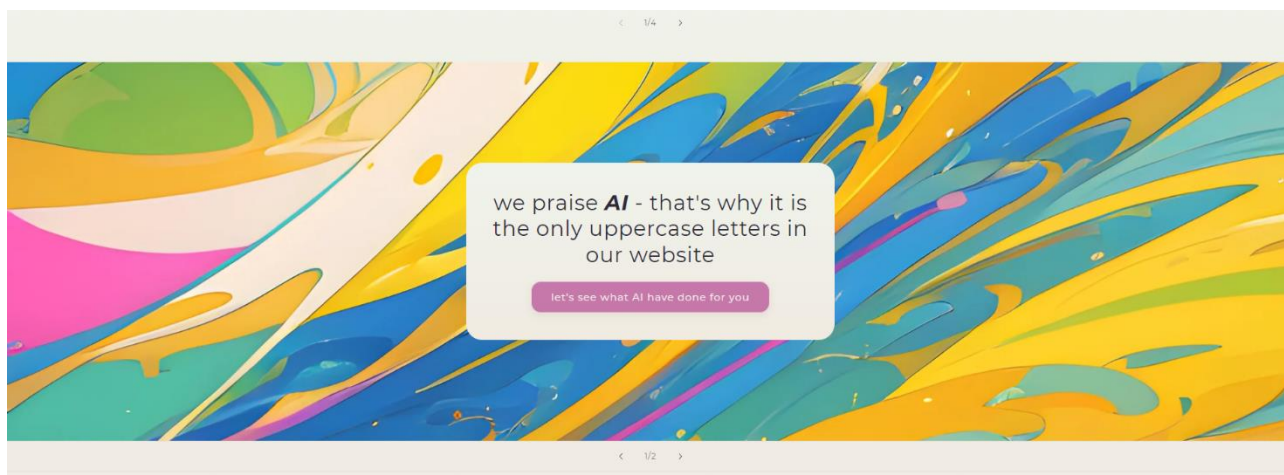


Рис. 2.10. Третій екран головної сторінки сайту

На четвертому екрані (рис. 2.11) користувач може побачити текстовий розділ з перевагами сайту. В блоці є заголовок, який допомагає користувачу зрозуміти, яку інформацію він зможе отримати з цього блоку, а також три міні-блоки з текстовим описом переваг цього онлайн-магазину.

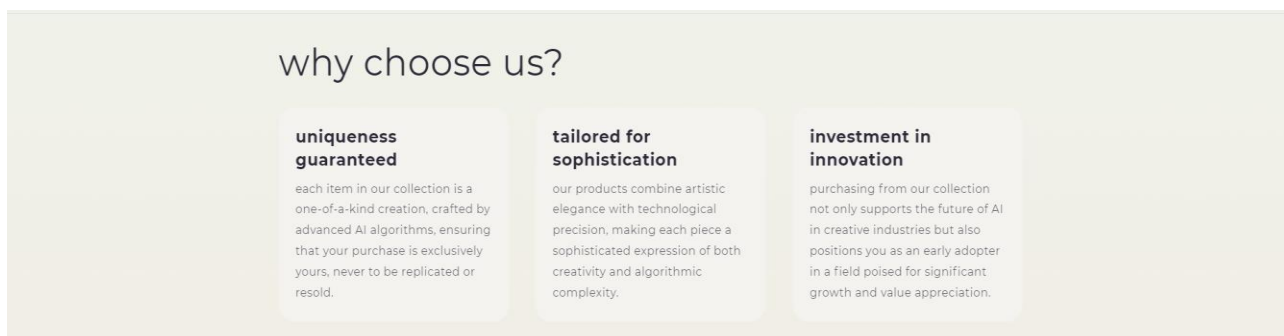


Рис. 2.11. Четвертий екран головної сторінки сайту

На п'ятому екрані (рис. 2.12) користувач може побачити фінальний текстовий розділ на головній розгортаються при натисканні на текстову частину питання.

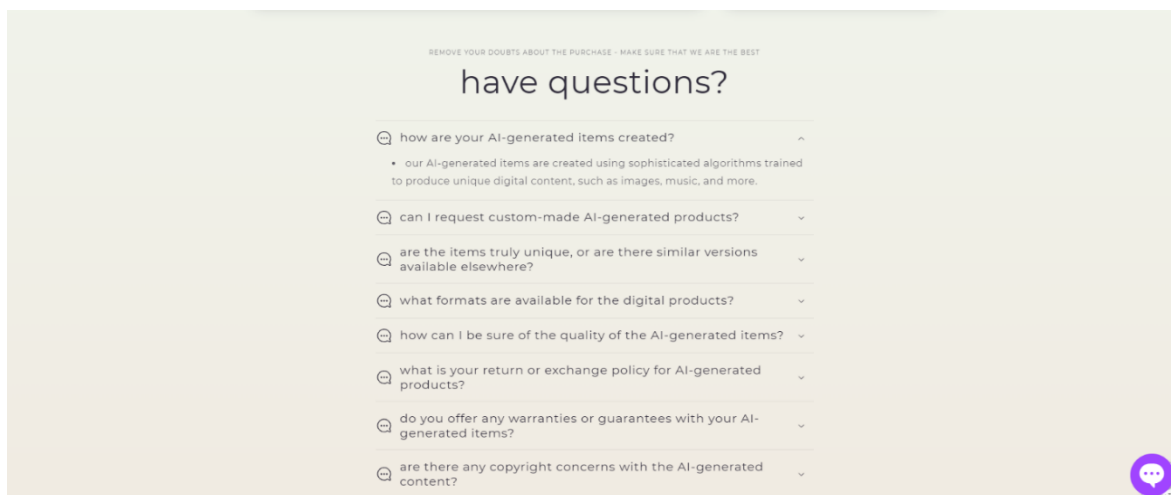


Рис. 2.12. П'ятий екран головної сторінки сайту

На будь-якій сторінці сайту (у тому числі і на головній) знизу користувач може відкрити чат-підтримки (рис. 2.13). Чат підтримки було встановлено за допомогою безкоштовного плагіну Chatway – Live Chat Support, який інтегрується напряму до магазину. Уся система комунікації розроблена командою Premіо, яка дозволяє додати та використовувати цю систему в своєму Shopify магазині. Основні функції, які були кастомізовані на боці адміністратора магазину:

- кольорова гама відображення вікна чату підтримки
- логотип магазину
- текстовий хедер, а також стандартні повідомлення, які відправляються за умови робочих годин магазину
- часи відображення віджету
- розмір віджету та його розташування

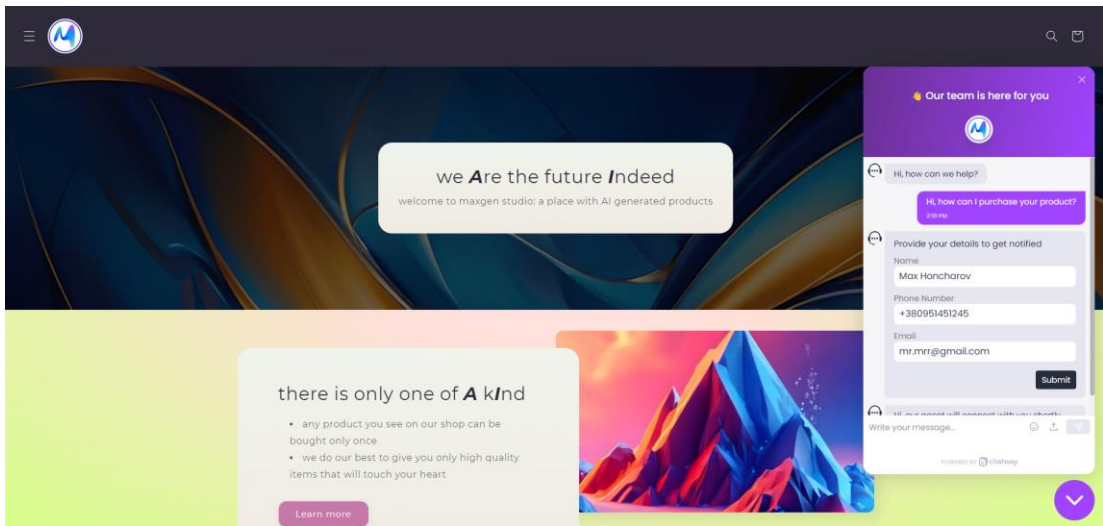


Рис. 2.13. Віджет чату підтримки

Усі чати з клієнтами зберігаються в спеціальній панелі адміністратора, яка знаходиться в одному місці з панеллю магазину Shopify (рис. 2.14). За допомогою цього можна вчасно побачити, коли клієнтам потрібна допомога, ставити теги та відмічати чати за статусом (не вирішені, вирішені, помічені, а також ділити за сегментами). Окрім того, усі дані, які клієнт вносить в input поля, можна побачити на окремій сторінці, же зберігається інформація про клієнтів, що дозволяє розширити свою базу.

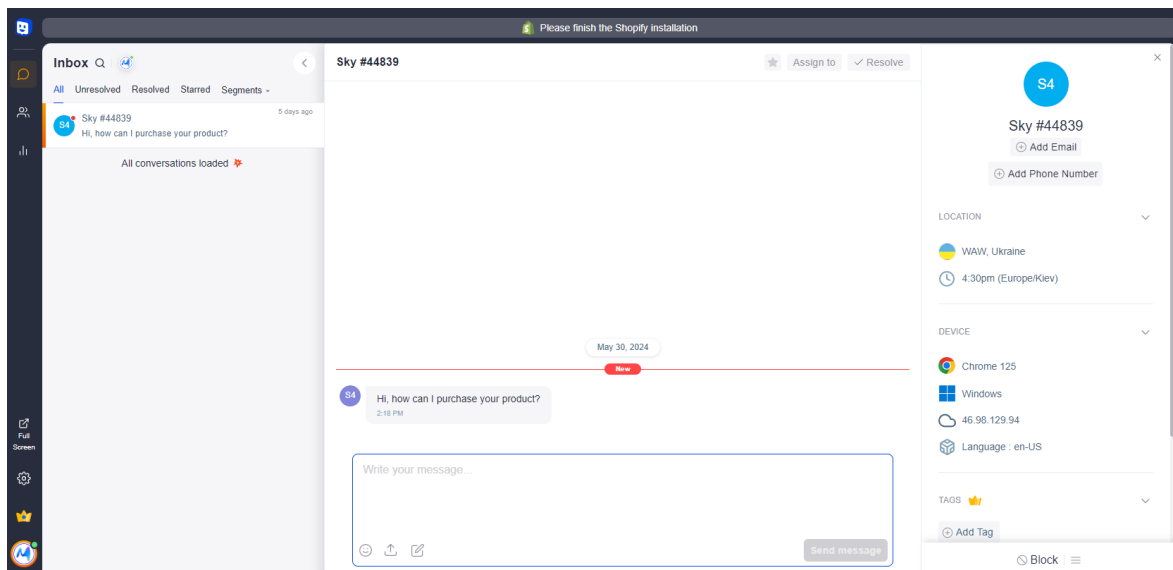


Рис. 2.14. Віджет чату підтримки

На сайті цього магазину можна перейти на сторінку товару (рис. 2.15), до якої можна потрапити як з головної сторінки, так і зі сторінки колекції, а також знайти потрібний товар, використовуючи пошук. Коли користувач переходить на сторінку товару, він може побачити:

- зображення товару, який пропонує придбати ця сторінка
- назва товару
- виробник товару
- ціна товару (звичайна або зі знижкою)
- опис товару

Після аналізу даних про товар, покупець може або одразу перейти до придбання, або додати товар до кошика.

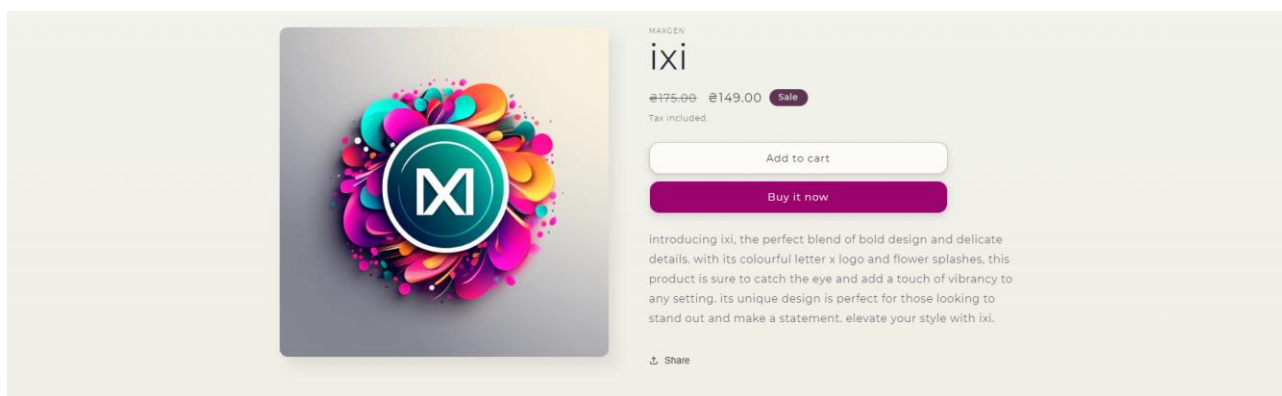


Рис. 2.15. Перший екран сторінки товару

При додаванні товару до кошика справа з'являється вікно, в якому можна буде побачити усі товари, які було додано до кошику (рис.2.16). На кожному товарі є перемикачі, які дозволяють обрати кількість товарів, яку покупець захоче придбати. Також є окреме поле, де покупець може залишити коментарі для магазину, які можна буде побачити в панелі адміністратора Shopify. Після того, як покупець впевнився, що його влаштовує зміст кошика, він може натиснути кнопку, яка відкриє сторінку чекаута, з якої можна здійснити оплату (рис. 2.17).

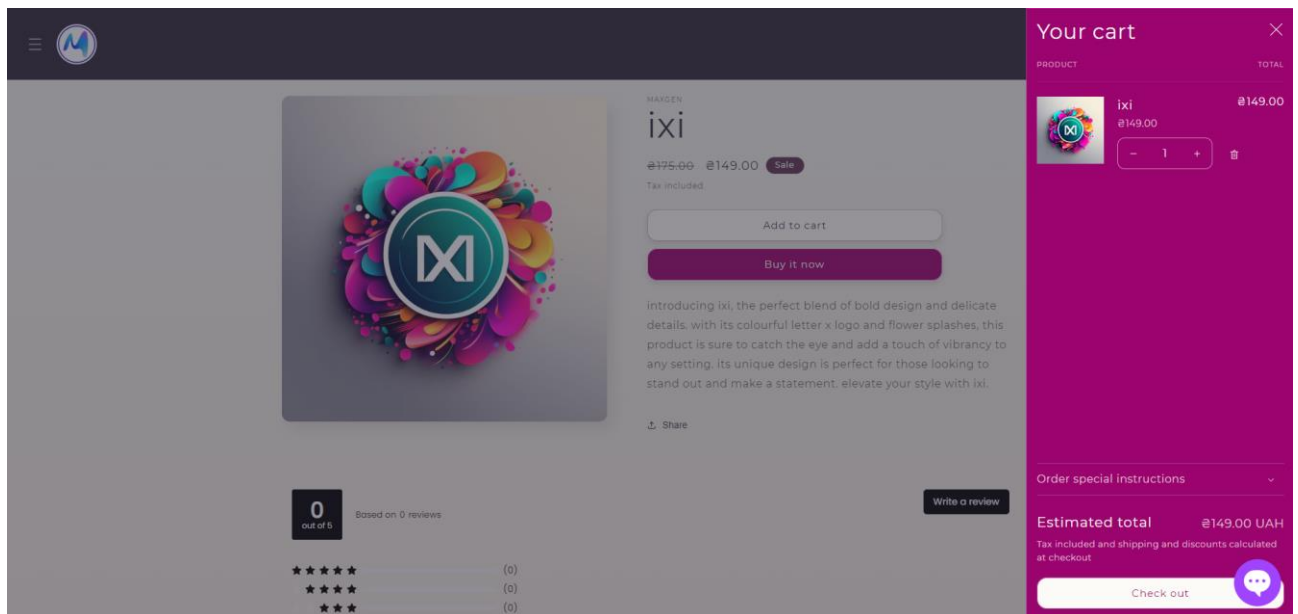


Рис. 2.16. Вікно кошику

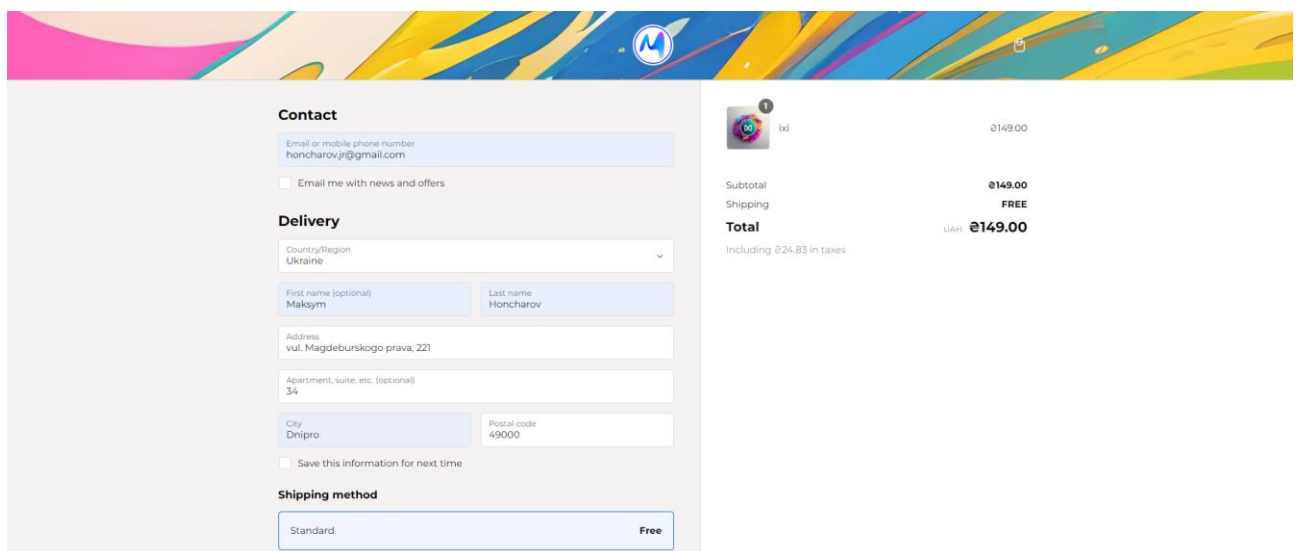


Рис. 2.17. Сторінка чекауту

Також на сторінці товару нижче опису можна знайти блок з відгуками, які залишили інші покупці, а також залишити свій (рис. 2.18). Віджет з відгуками було встановлено за допомогою безкоштовного плагіну Lai Product Reviews, який інтегрується напряму до магазину. Уся система комунікації розроблена командою Lai, яка дозволяє додати та використовувати цю систему в своєму Shopify магазині.

Основні функції, які були кастомізовані на боці адміністратора магазину:

- кольорова гама відображення відгуків
- розташування блоків відгуків

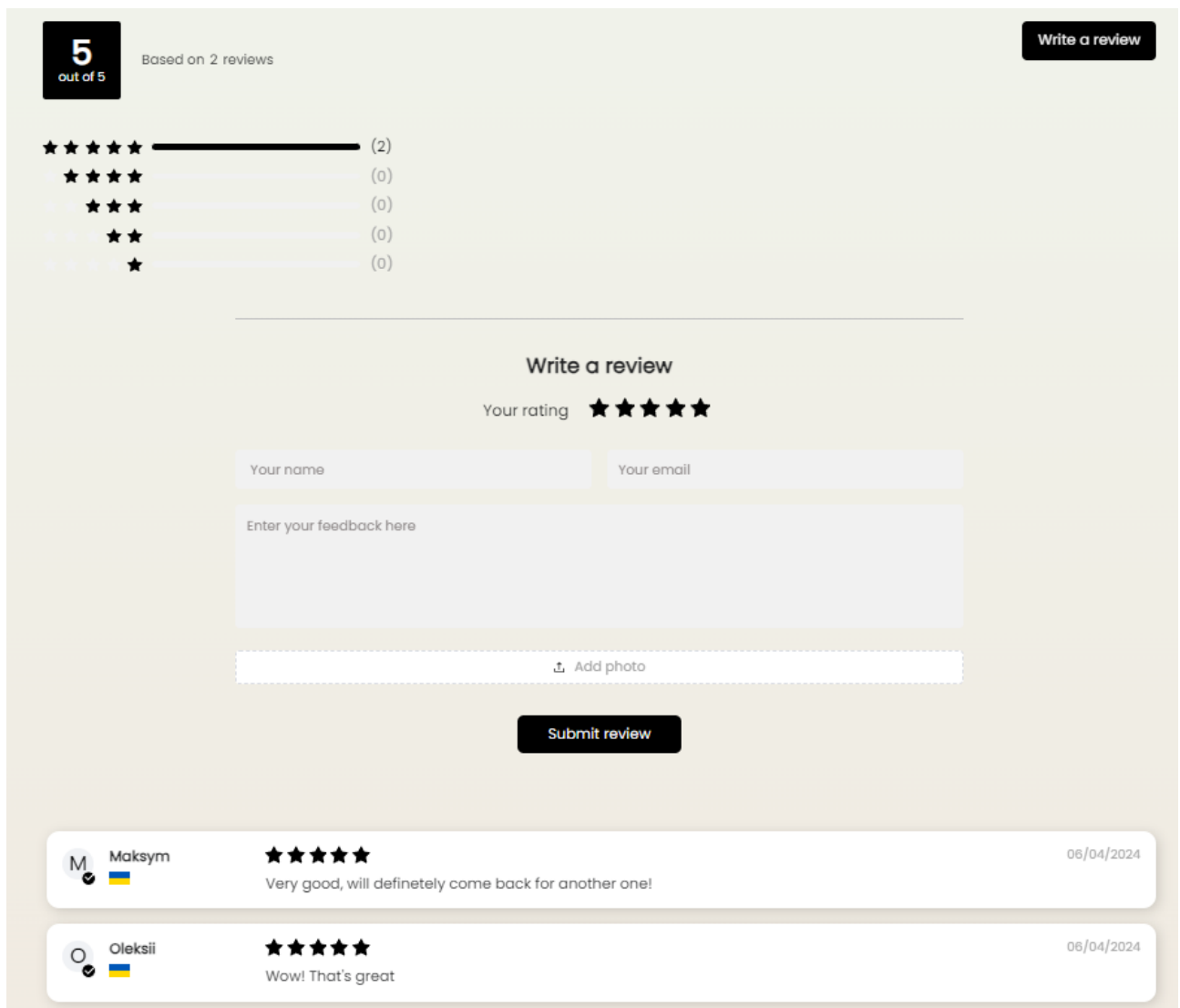


Рис. 2.18. Другий екран сторінки товару з відгуками

Останнім блоком сторінки товару є блок з відображенням схожих товарів, які покупець може також придбати (рис.2.19). Блок формується за допомогою завантаження туди товарів, які знаходяться в одній колекції з товарів, на сторінці якого цей блок відображається.

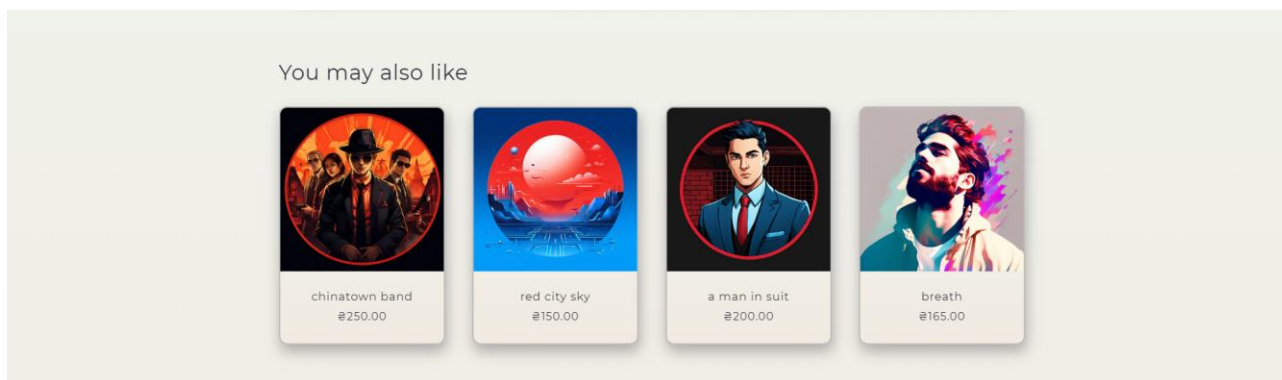


Рис. 2.19. Третій екран сторінки товару

З будь-якої сторінки сайту можна за допомогою хедера потрапити в меню, натиснувши на три стрічки зліва (рис. 2.20). З меню можна потрапити на головну сторінку сайту, на сторінку каталогу, на сторінку контактів, а також на ще одне меню зі списком та посиланнями на усі категорії магазину.

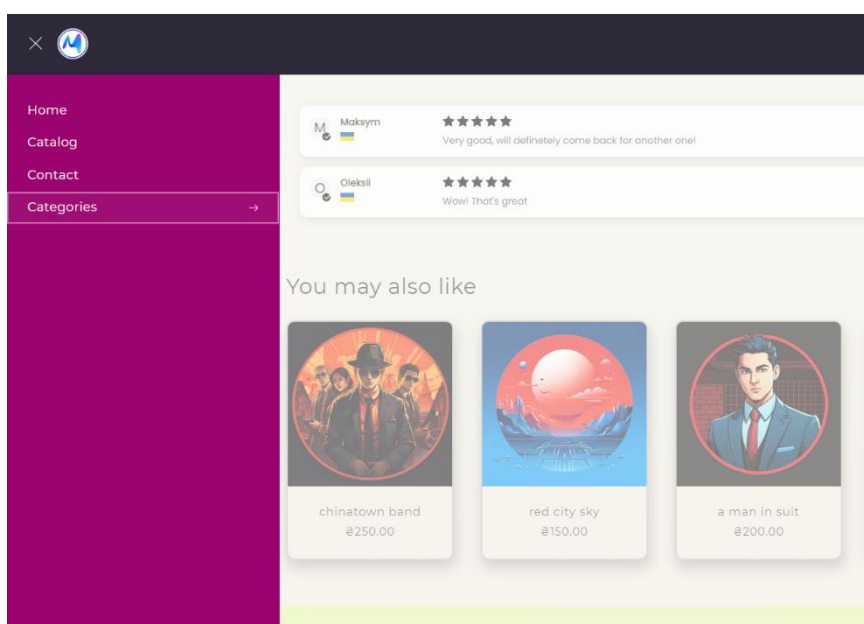


Рис. 2.20. Меню магазину

На сторінці каталогу користувач бачить усі товари, які доступні на сайті (рис. 2.21). Він може відсортувати їх за значеннями, ціною, а також поставити фільтри.

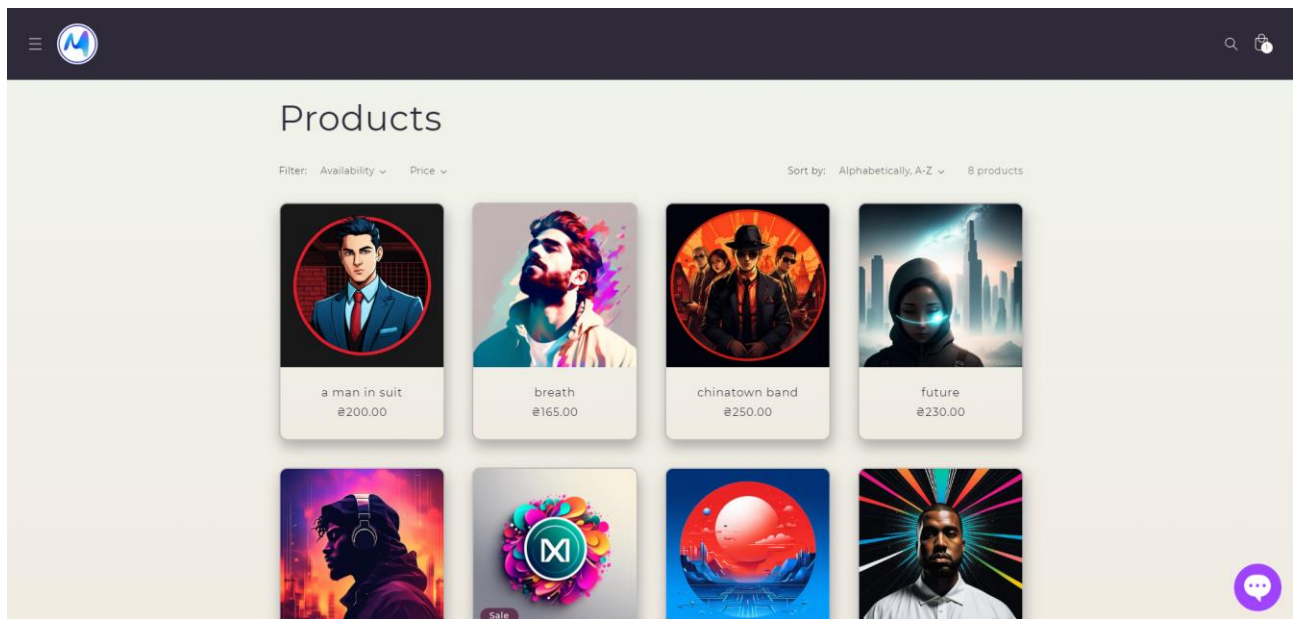


Рис. 2.21. Меню магазину

Для сервісу, який використовується для імпорту товарів в магазин, користувачу необхідно потрапити в Django фреймворк з налаштованою системою для імпортування (рис. 2.22):

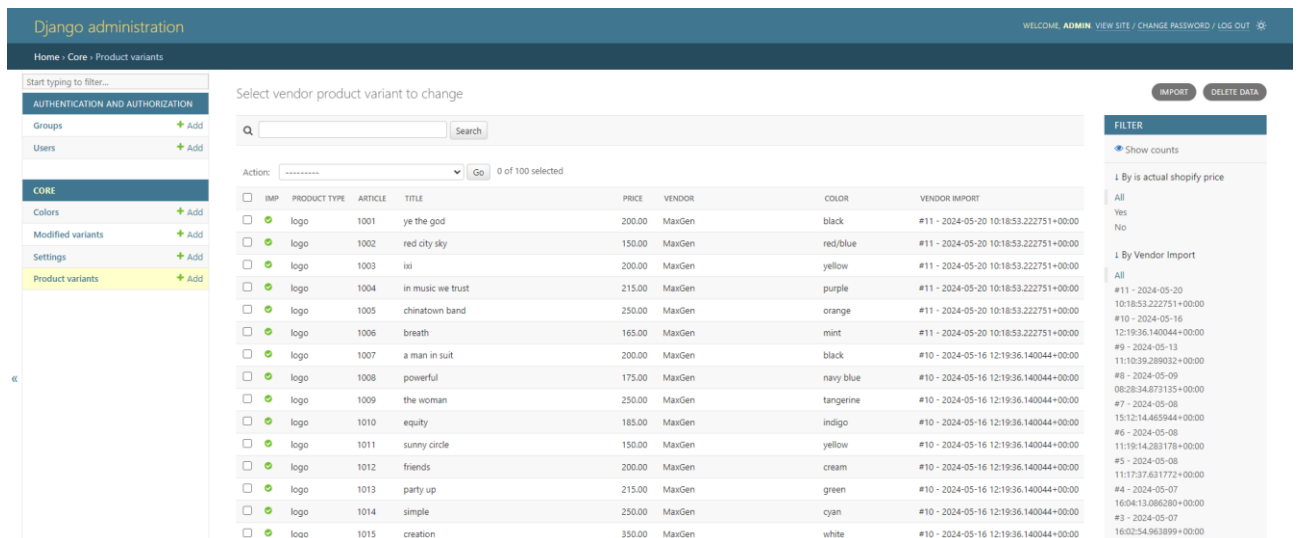


Рис. 2.22. Сервіс для імпортування товарів

На сервісі міститься сторінка з усіма товарами, які було заповнено в Excel файлі та завантажено у Django. Після того, як користувач додав усі бажані товари на сервіс, він має впевнитись, що усі кольори занесені в базу, з якої формується

liquid файл і передається на Shopify. Щоб це зробити, користувач повинен натиснути на кнопку меню «Colors» (рис. 2.23)

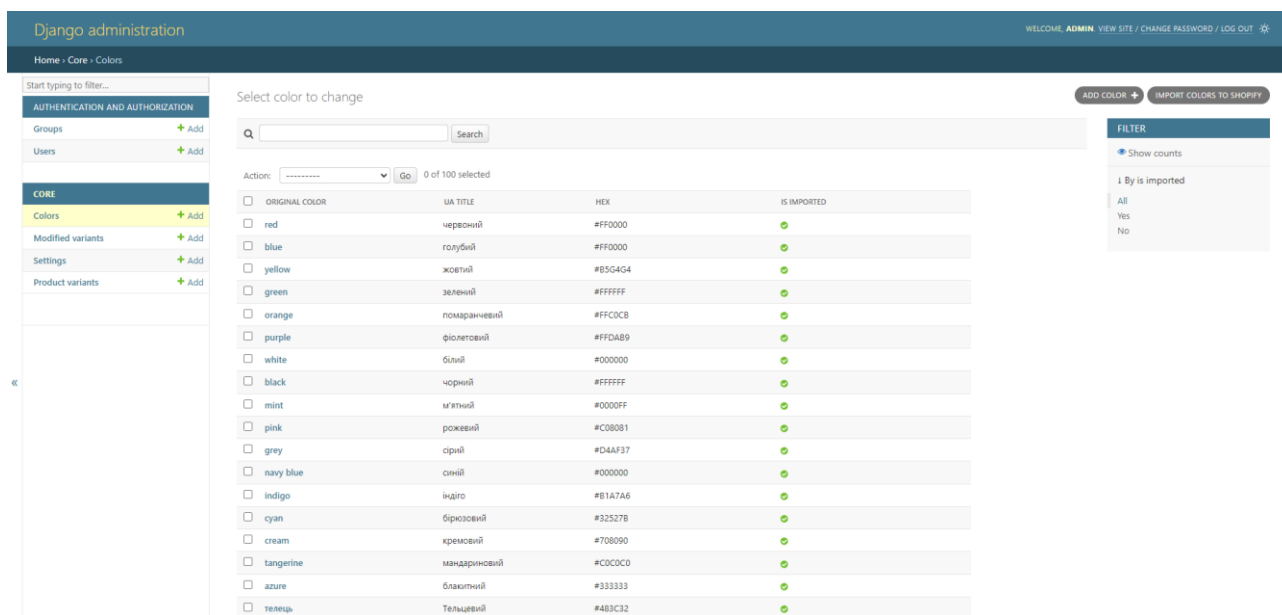


Рис. 2.23. Сторінка з усіма кольорами товарів

Для редагування будь-якого кольору, користувач повинен натиснути на значення «ORIGINAL COLOR» (рис. 2.24). Після натискання відкривається сторінка кольору, яка має такі поля:

- English color – назва кольору, яку приймає Shopify.
- Ua title – назва кольору, яка відображається в адмін панелі.
- Hex – код кольору, який відображається.
- Is imported – перевірка на те, чи є вже товар з цим кольором на Shopify.

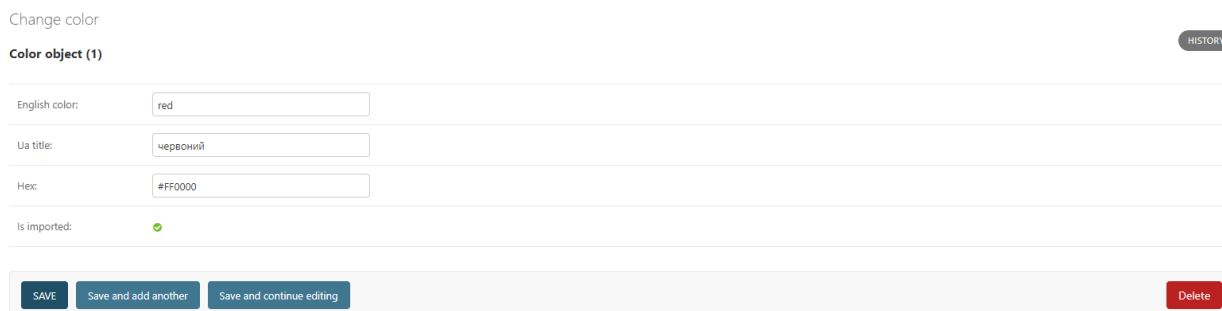


Рис. 2.24. Сторінка кольору

Використання сервісу Shopify надає певні переваги в обліку товарів та клієнтів. На головній сторінці адміністративної панелі вебмагазину розташоване навігаційне меню, на якому є сторінки з замовленнями, товарами, покупцями, маркетинговою аналітикою тощо. Також з цієї ж сторінки можна зручно потрапити до перегляду превью сайту, до редагування коду, а також до створення окремих сторінок і блогових статей.

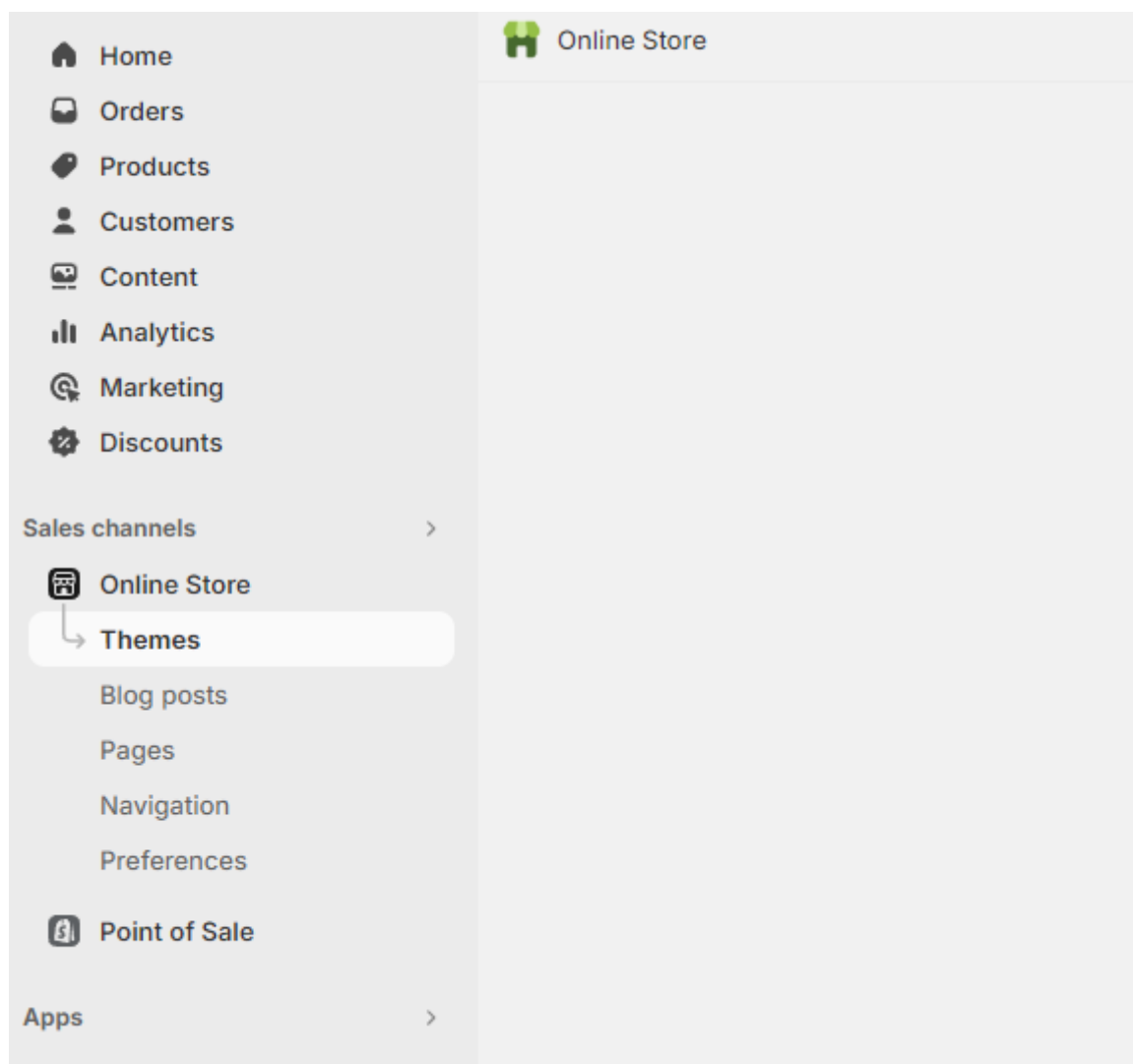
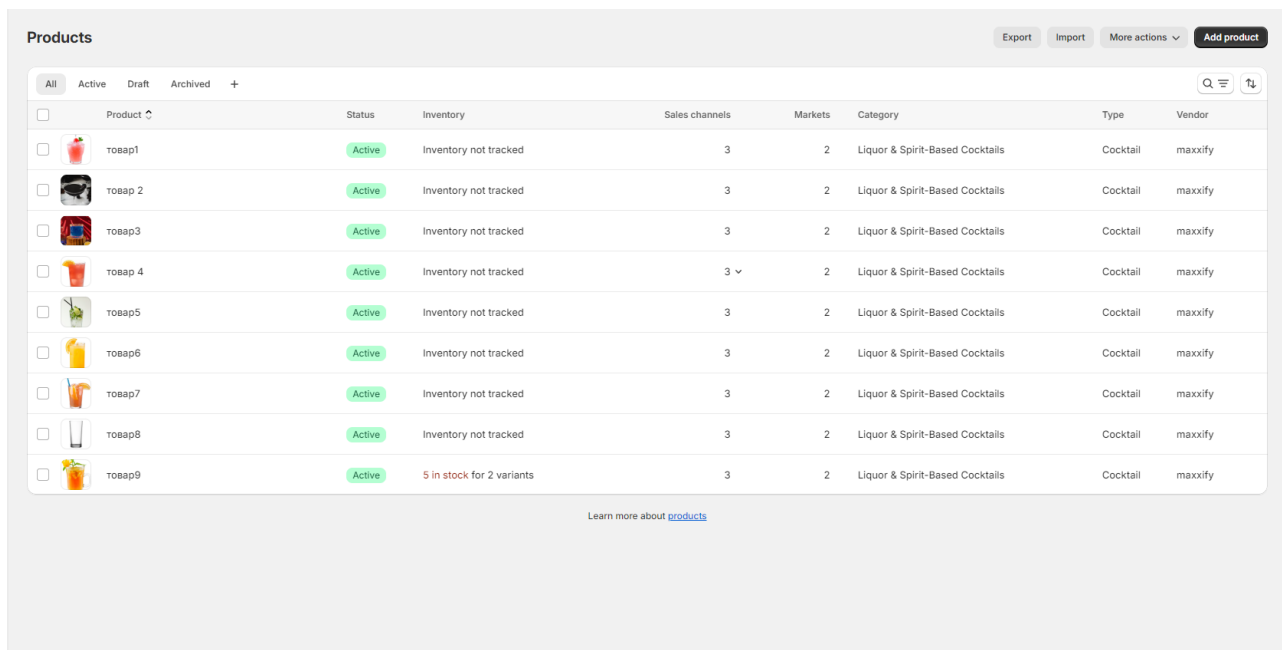


Рис. 2.25. Адміністративна панель Shopify

На сторінці з товарами розташований список з усіма товарами, які якийсь час були на сайті (рис. 2.26). Для того, щоб не плутати застарі товари з новими, їх можна зручно сортувати за групами, а саме обрати усі товари, або лише активні, або лише драфти товарі, або заархівовані. На цій сторінці також можна

створити продукт власноруч без імпортування цих даних за допомогою Django, але цей функціонал не є зручним, коли треба додати багато товарів. На кожному товарі також розписані його основні властивості, за якими також можна застосувати сортування.

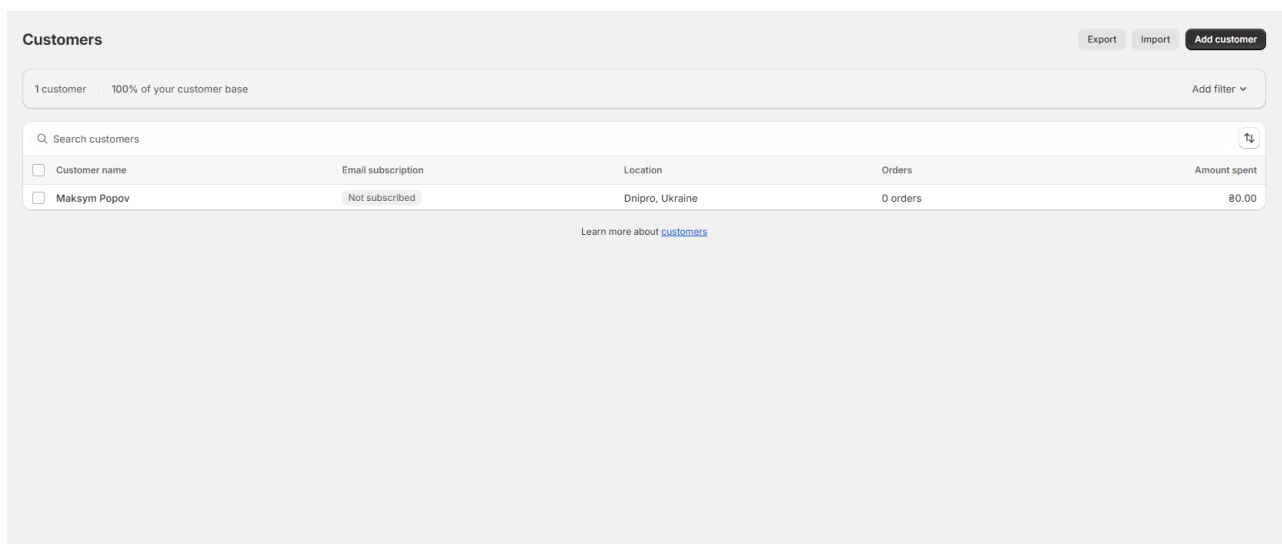


The screenshot shows a 'Products' management page with a table of 9 items. The table has columns for Product, Status, Inventory, Sales channels, Markets, Category, Type, and Vendor. All items are in an 'Active' status and have 'Inventory not tracked'. The categories are 'Liquor & Spirit-Based Cocktails' and the type is 'Cocktail'. The vendor for all items is 'maxxify'.

Product	Status	Inventory	Sales channels	Markets	Category	Type	Vendor
товар1	Active	Inventory not tracked	3	2	Liquor & Spirit-Based Cocktails	Cocktail	maxxify
товар2	Active	Inventory not tracked	3	2	Liquor & Spirit-Based Cocktails	Cocktail	maxxify
товар3	Active	Inventory not tracked	3	2	Liquor & Spirit-Based Cocktails	Cocktail	maxxify
товар4	Active	Inventory not tracked	3	2	Liquor & Spirit-Based Cocktails	Cocktail	maxxify
товар5	Active	Inventory not tracked	3	2	Liquor & Spirit-Based Cocktails	Cocktail	maxxify
товар6	Active	Inventory not tracked	3	2	Liquor & Spirit-Based Cocktails	Cocktail	maxxify
товар7	Active	Inventory not tracked	3	2	Liquor & Spirit-Based Cocktails	Cocktail	maxxify
товар8	Active	Inventory not tracked	3	2	Liquor & Spirit-Based Cocktails	Cocktail	maxxify
товар9	Active	5 in stock for 2 variants	3	2	Liquor & Spirit-Based Cocktails	Cocktail	maxxify

Рис. 2.26. Сторінка усіх товарів вебмагазину

На сторінці з покупцями відображені усі акаунти, з яких колись відбулося придбання товару на цьому вебмагазині.



The screenshot shows a 'Customers' management page with a table of 1 customer. The table has columns for Customer name, Email subscription, Location, Orders, and Amount spent. The customer is 'Maksym Popov', who is 'Not subscribed', located in 'Dnipro, Ukraine', has '0 orders', and has spent '€0.00'.

Customer name	Email subscription	Location	Orders	Amount spent
Maksym Popov	Not subscribed	Dnipro, Ukraine	0 orders	€0.00

Рис. 2.27. Сторінка усіх покупців вебмагазину

Аналітична сторінка дозволяє відстежувати повну кількість продажів за обраний розробником час, бачити графіки замовлень та продажів, а також рейтинг коверсії (рис. 2.28).

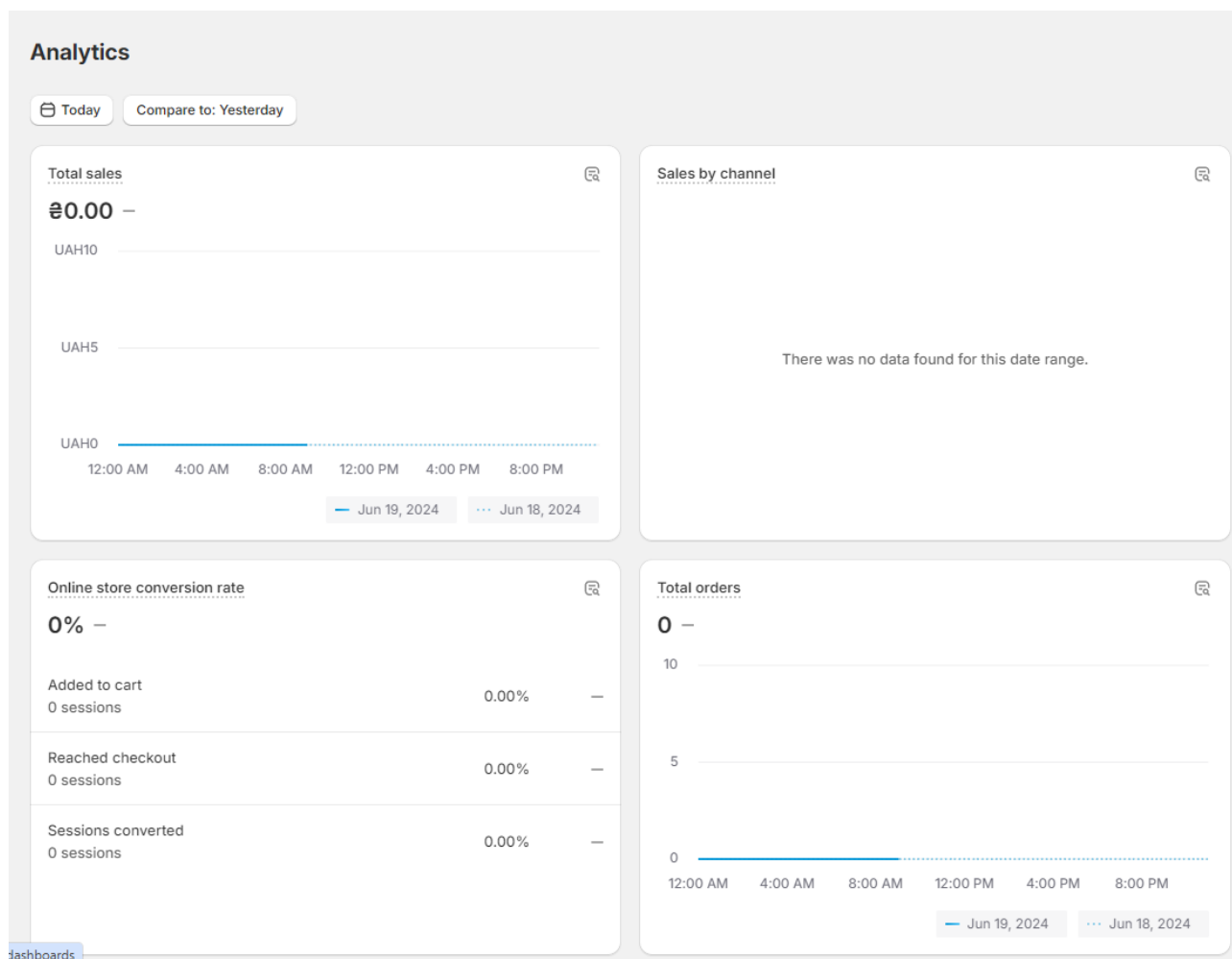


Рис. 2.28. Сторінка аналітики вебмагазину

РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості розробки програмного забезпечення

Вхідні дані:

q – передбачуване число операторів – 2500;

p – коефіцієнт кореляції програми в ході її розробки – 0,3;

C – коефіцієнт складності програми – 1,15.

годинна заробітна плата розробника – 240,96 грн/год; [6]

Відповідно до інформації на сайті [jooble.org](https://ua.jooble.org/) (<https://ua.jooble.org/>) і провівши аналіз даних, можна зробити висновок, що середня місячна заробітна плата web developer становить 1000,00 дол.США. На момент написання цієї записки, Офіційний курс гривні Національного банку України щодо долара становить 40, 4813 гривень за 1 Долар США [11](див. рис. 3.1.). Тобто, при заробітній платі в розмірі 1000 дол. США, це еквівалентно 40 481,3 грн. на місяць. Робочий графік розробника в середньому складається з 21 робочого дня по 8 годин на день. Отже, погодинна ставка становитиме (40 481,3 грн / 21 робочий день) / 8 годин = 240,96 грн. на годину.


Код цифровий	Код літерний	Кількість одиниць валюти	Назва валюти	Офіційний курс 
840	USD	1	Долар США	40,4813

Рис. 3.1. Курс долар/гривня

B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,4;

k – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,05;

вартість машино-години ЕОМ – 1, 79 грн/год.

При написанні кваліфікаційної роботи додаткове обладнання та приміщення не використовувалося.

Для розрахунку вартості машино-годин ПОЕМ були враховані витрати на електроенергію та домашній інтернет. Згідно з даними Мінфіну на момент написання роботи тариф на електроенергію для населення становила 4,32 грн/кВт·год. Домашній інтернет коштував 200,00 грн. на місяць, тобто 9,23 грн в день, 1,15 грн за годину. Ноутбук споживав 150 Вт. Таким чином вартість 1 години роботи на зазначеному ноутбуці становить $0,15 \cdot 4,32 = 0,648$ грн. За розрахунками $0,648 + 1,15 = 1,79$ (грн/год). Отже, вартість машино-години ЕОМ $C_{мч} = 1,79$ грн/год;

B_k – кількість розробників – 1.

Беручи до уваги, творчий характер роботи web розробника, нормування праці в процесі розробки програмного забезпечення не можна розглядати стандартно. Тому, трудомісткість розробки сайту може бути розрахована з використанням системи моделей, що пропонують різні рівні точності оцінки. Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_{и} + t_a + t_{п} + t_{отл} + t_d, \quad (3.1)$$

де t_o – витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

$t_{и}$ – витрати праці на дослідження алгоритму рішення задачі;

t_a – витрати праці на розробку блок-схеми алгоритму;

$t_{п}$ – витрати праці на програмування по готовій блок-схемі;

$t_{отл}$ – витрати праці на налагодження програми на ЕОМ;

t_d – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмному забезпеченні, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p), \quad (3.2)$$

де q – передбачуване число операторів (2500);

C – коефіцієнт складності програми (1,15);

p – коефіцієнт корекції програми в ході її розробки (0,3).

Звідси за формулою (3.2) умовне число операторів в програмі:

$$Q = 2500 \cdot 1,15 \cdot (1+0,3) = 3737,5 \text{ людино-годин.}$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \cdot 85) \cdot k}, \quad (3.3)$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності, стаж роботи 1 рік, тому коефіцієнт кваліфікації програміста = 1,05.

Визначимо збільшення витрат праці внаслідок недостатнього опису завдання не більше 50% ($B = 1,4$). Враховуючи коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності $k = 1,05$, отримуємо витрати праці на вивчення опису завдання за формулою (3.3):

Після підставлення значень маємо:

$$t_u = (3737,5 * 1,4) / (85 * 1,05) = 58,6 \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20..25) \cdot k}, \text{ людино-годин,} \quad (3.5)$$

де Q – умовне число операторів програми, людино-годин (3737, 5 людино-годин);

k – коефіцієнт кваліфікації програміста (1,05).

Підставивши відповідні значення в формулу (3.5), отримаємо:

$$t_a = 3737, 5 / (23 * 1,05) = 154,76 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_{\Pi} = \frac{Q}{(20..25) \cdot k}, \text{ людино-годин.} \quad (3.6)$$

За формулою (3.6) та значенням параметру Q , обчислюємо витрати праці на програмування по готовій блок-схемі:

$$t_n = 3737, 5 / (25 * 1,05) = 142,38 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{\text{отл}} = \frac{Q}{(4..5) \cdot k}, \text{ людино-годин.} \quad (3.7)$$

Витрати праці на налагодження програми на ЕОМ за умови автономного налагодження одного завдання за формулою (3.7):

Підставивши значення:

$$t_{\text{отл}} = 3737, 5 / (5 * 1,05) = 711,90 \text{ людино-годин.}$$

– за умови комплексного налагодження завдання:

$$t_{отл}^k = 1,5 \cdot t_{отл}, \text{ людино-годин.} \quad (3.8)$$

Витрати праці на налагодження програми на ЕОМ за умови комплексного налагодження завдання за формулою (3.8):

$$t_{отл}^k = 1,5 * 711,90 = 1067,85 \text{ людино-годин.}$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_d = t_{др} + t_{до}, \text{ людино-годин,} \quad (3.9)$$

де $t_{др}$ – трудомісткість підготовки матеріалів і рукопису:

$$t_{др} = \frac{q}{(15..20) \cdot k}, \text{ людино-годин,} \quad (3.10)$$

$t_{до}$ – трудомісткість редагування, печатки й оформлення документації:

$$t_{до} = 0,75 \cdot t_{др}, \text{ людино-годин.} \quad (3.11)$$

За формулами (3.9), (3.10) та (3.11) обчислюємо витрати праці на документацію:

$$t_{др} = 3737,5 / (20 \cdot 1,05) = 177,97 \text{ людино-годин,}$$

$$t_{до} = 0,75 \cdot 177,97 = 133,47 \text{ людино-годин,}$$

$$t_d = 177,97 + 133,47 = 311,47 \text{ людино-годин.}$$

Повертаючись до формули (3.1), отримаємо повну оцінку трудомісткості розробки програмного забезпечення:

$t = 50 + 58,6 + 154,76 + 142,38 + 711,90 + 311,47 = 1429,11$ людино-години.

3.2. Розрахунок витрат на створення програмного забезпечення

Витрати на створення web сайту Кпо включають витрати на заробітну плату виконавця програми Зз/п і витрат машинного часу, необхідного на налагодження програми на ЕОМ

$$K_{\text{ПО}} = Z_{\text{ЗП}} + Z_{\text{МВ}}, \text{ грн.} \quad (3.12)$$

Заробітна плата виконавців визначається за формулою:

$$Z_{\text{ЗП}} = t \cdot C_{\text{ЗП}}, \text{ грн,} \quad (3.13)$$

де: t - загальна трудомісткість, людино-годин;

$C_{\text{ЗП}}$ - середня годинна заробітна плата програміста, грн/година

Середня плата за одну годинну роботи web розробника становить 240,96 грн., тому:

$$Z_{\text{ЗП}} = 1429,11 \cdot 240,96 = 344\,358,3456 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$Z_{\text{МВ}} = t_{\text{отл}} \cdot C_{\text{мч}}, \text{ грн,} \quad (3.14)$$

де $t_{\text{отл}}$ - трудомісткість налагодження програми на ЕОМ, год (711,90 год);

$C_{\text{мч}}$ - вартість машино-години ЕОМ, грн/год (1,79 грн/год.).

Підставивши в формулу (3.14) відповідні значення, обчислюємо вартість необхідного для налагодження машинного часу:

$$Z_{\text{МВ}} = 711,90 \cdot 1,79 = 1274,301 \text{ грн.}$$

Звідси, за формулою (3.12), витрати на створення програмного продукту:

$$K_{\text{ПО}} = 344\,358,3456 + 1274,301 = 345\,632,6466 \text{ грн.}$$

Очікуваний період створення програмного застосунку:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс,} \quad (3.15)$$

де B_k – число виконавців (дорівнює 1);

F_p – місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

Звідси, за формулою (3.15), витрати на створення програмного продукту:

$$T = 1429,11 / (1 \cdot 176) = 8,2 \text{ місяців}$$

На основі вхідних даних, таких як передбачуване число операторів, коефіцієнти кореляції та складності програми, середня заробітна плата розробника, а також витрати на електроенергію та інтернет, було визначено загальні трудові витрати та витрати на створення програмного продукту.

Висновки: трудомісткість розробки програмного забезпечення, враховуючи всі етапи розробки, від підготовки та опису задачі до налагодження та підготовки документації, склала 1429,11 людино-годин. Ця оцінка включає в себе всі основні компоненти витрат праці, розраховані на основі кількості операторів та різних коефіцієнтів, що впливають на процес розробки.

Загальні витрати на створення програмного забезпечення склали 345 632,65 грн, з яких основну частину становила заробітна плата виконавця – 344 358,35 грн. Додаткові витрати включали вартість машинного часу для налагодження програми, що становила 1274,30 грн.

Очікуваний період створення програмного застосунку, враховуючи місячний фонд робочого часу, склав приблизно 8,2 місяця.

Таким чином, проведені розрахунки показали, що розробка програмного забезпечення є досить трудомістким і витратним процесом, що вимагає значних людських та матеріальних ресурсів. Отримані дані можуть бути використані для планування ресурсів і розрахунку бюджету при реалізації подібних проектів у майбутньому.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було розроблено веборієнтований додаток для вебмагазину, який займається продажем товарів, згенерованих штучним інтелектом. Цей додаток було реалізовано за допомогою HTML, CSS, Python, а також з використанням фреймворку Django та Shopify.

Веб орієнтований додаток виконує заданий функціонал:

- Забезпечити швидке та зручне користування вебмагазином для продажу товарів.
- Заохочити людей більше користуватися сервісами та купляти товари, які були створені за допомогою штучного інтелекту.
- Зробити зручне середовище для адміністратора сайту, який може керувати обліком товарів та покупців, а також керувати процесом імпортування товарів на Shopify.

Під час виконання даної кваліфікаційної роботи також було встановлено складність розробленої системи і здійснено розрахунок вартості роботи зі створення програми, використовуючи середню заробітну плату FullStack Python розробника.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Shopify Documentation. URL: <https://shopify.dev/docs> (дата звернення: 19.05.2024)
2. HTML.com. URL: <https://html.com/> (дата звернення: 15.04.2024)
3. Mozilla Developer Network (MDN). URL: <https://developer.mozilla.org/en-US/> (дата звернення: 19.06.2024)
4. Django Documentation. URL: <https://docs.djangoproject.com/en/stable/> (дата звернення: 19.06.2024)
5. Real Python. URL: <https://realpython.com/> (дата звернення: 19.06.2024)
6. Medium. URL: <https://medium.com/> (дата звернення: 19.06.2024)
7. Stack Overflow. URL: <https://stackoverflow.com/> (дата звернення: 19.06.2024)
8. Shopify Liquid Documentation.
URL: <https://shopify.dev/docs/themes/liquid/reference> (дата звернення: 19.06.2024)
9. Shopify API Documentation. URL: <https://shopify.dev/docs/admin-api/rest/reference> (дата звернення: 19.06.2024)
10. CSS-Tricks. URL: <https://css-tricks.com/> (дата звернення: 19.06.2024)
11. W3Schools. URL: <https://www.w3schools.com/> (дата звернення: 19.06.2024)
12. Python.org. URL: <https://www.python.org/> (дата звернення: 19.06.2024)
13. Django Girls Tutorial. URL: <https://tutorial.djangogirls.org/> (дата звернення: 19.06.2024)
14. Shopify Developers Community.
URL: <https://community.shopify.com/c/Shopify-APIs-SDKs/bd-p/shopify-apis-and-sdks> (дата звернення: 19.06.2024)
15. CSS Grid Layout. URL: <https://cssgrid.io/> (дата звернення: 19.06.2024)
16. Smashing Magazine.
URL: <https://www.smashingmagazine.com/2021/12/optimizing-shopify-ecommerce-website/> (дата звернення: 19.06.2024)

17. Envato Tuts+. URL: <https://tutsplus.com/> (дата звернення: 19.06.2024)
18. 77-а студентська науково-технічна конференція "Тиждень Студентської Науки" (16-20 травня 2022 р.), Секція 11 – Інформаційні та телекомунікаційні технології / Буслов. Д. Ю. Наукові керівники: к.т.н., доц. Спірінцев В. В. к.т.н., доц. Ширін А. Л.: НТУ «Дніпровська політехніка», 2022
19. Методичні рекомендації до виконання кваліфікаційних робіт здобувачів першого рівня вищої освіти спеціальності 121 Інженерія програмного забезпечення / В.В. Спірінцев, О.С. Шевцова, І.М. Удовик; Д : НТУ «Дніпровська політехніка», 2022. – 60 с.

КОД ПРОГРАМИ

theme.liquid

```

<!doctype html>
<html class="js" lang="{{ request.locale.iso_code }}">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <meta name="theme-color" content="">
  <link rel="canonical" href="{{ canonical_url }}">

  {% if settings.favicon != blank %}
    <link rel="icon" type="image/png" href="{{ settings.favicon | image_url: width: 32, height: 32 }}">
  {% endif %}

  {% unless settings.type_header_font.system? and settings.type_body_font.system? %}
    <link rel="preconnect" href="https://fonts.shopifycdn.com" crossorigin>
  {% endunless %}

  <title>
    {{ page_title }}
    {% if current_tags %} &ndash; tagged "{{ current_tags | join: ', ' }}" {% endif %}
    {% if current_page != 1 %} &ndash; Page {{ current_page }} {% endif %}
    {% unless page_title contains shop.name %} &ndash; {{ shop.name }} {% endunless %}
  </title>

  {% if page_description %}
    <meta name="description" content="{{ page_description | escape }}">
  {% endif %}

  {% render 'meta-tags' %}

  <script src="{{ 'constants.js' | asset_url }}" defer="defer"></script>
  <script src="{{ 'pubsub.js' | asset_url }}" defer="defer"></script>
  <script src="{{ 'global.js' | asset_url }}" defer="defer"></script>
  {% if settings.animations_reveal_on_scroll %}
    <script src="{{ 'animations.js' | asset_url }}" defer="defer"></script>
  {% endif %}

  {{ content_for_header }}

  {%- liquid
    assign body_font_bold = settings.type_body_font | font_modify: 'weight', 'bold'
    assign body_font_italic = settings.type_body_font | font_modify: 'style', 'italic'
    assign body_font_bold_italic = body_font_bold | font_modify: 'style', 'italic'
  %}

  {% style %}
    {{ settings.type_body_font | font_face: font_display: 'swap' }}
    {{ body_font_bold | font_face: font_display: 'swap' }}
    {{ body_font_italic | font_face: font_display: 'swap' }}
    {{ body_font_bold_italic | font_face: font_display: 'swap' }}
    {{ settings.type_header_font | font_face: font_display: 'swap' }}

  {% for scheme in settings.color_schemes %}
    {% assign scheme_classes = scheme_classes | append: ', .color-' | append: scheme.id %}
    {% if forloop.index == 1 %}
      :root,

```

```

    {%- endif %}
    .color-{{ scheme.id }} {
      --color-background: {{ scheme.settings.background.red }},{{ scheme.settings.background.green }},{{
scheme.settings.background.blue }};
      {% if scheme.settings.background_gradient != empty %}
      --gradient-background: {{ scheme.settings.background_gradient }};
      {% else %}
      --gradient-background: {{ scheme.settings.background }};
      {% endif %}

      {% liquid
      assign background_color = scheme.settings.background
      assign background_color_brightness = background_color | color_brightness
      if background_color_brightness <= 26
        assign background_color_contrast = background_color | color_lighten: 50
      elsif background_color_brightness <= 65
        assign background_color_contrast = background_color | color_lighten: 5
      else
        assign background_color_contrast = background_color | color_darken: 25
      endif
      %}

      --color-foreground: {{ scheme.settings.text.red }},{{ scheme.settings.text.green }},{{ scheme.settings.text.blue
}};
      --color-background-contrast: {{ background_color_contrast.red }},{{ background_color_contrast.green }},{{
background_color_contrast.blue }};
      --color-shadow: {{ scheme.settings.shadow.red }},{{ scheme.settings.shadow.green }},{{
scheme.settings.shadow.blue }};
      --color-button: {{ scheme.settings.button.red }},{{ scheme.settings.button.green }},{{ scheme.settings.button.blue
}};
      --color-button-text: {{ scheme.settings.button_label.red }},{{ scheme.settings.button_label.green }},{{
scheme.settings.button_label.blue }};
      --color-secondary-button: {{ scheme.settings.background.red }},{{ scheme.settings.background.green }},{{
scheme.settings.background.blue }};
      --color-secondary-button-text: {{ scheme.settings.secondary_button_label.red }},{{
scheme.settings.secondary_button_label.green }},{{ scheme.settings.secondary_button_label.blue }};
      --color-link: {{ scheme.settings.secondary_button_label.red }},{{ scheme.settings.secondary_button_label.green
}},{{ scheme.settings.secondary_button_label.blue }};
      --color-badge-foreground: {{ scheme.settings.text.red }},{{ scheme.settings.text.green }},{{
scheme.settings.text.blue }};
      --color-badge-background: {{ scheme.settings.background.red }},{{ scheme.settings.background.green }},{{
scheme.settings.background.blue }};
      --color-badge-border: {{ scheme.settings.text.red }},{{ scheme.settings.text.green }},{{ scheme.settings.text.blue
}};
      --payment-terms-background-color: rgb({{ scheme.settings.background.rgb }});
    }
    {% endfor %}

    {{ scheme_classes | prepend: 'body' }} {
      color: rgba(var(--color-foreground), 0.75);
      background-color: rgb(var(--color-background));
    }

:root {
  --font-body-family: {{ settings.type_body_font.family }}, {{ settings.type_body_font.fallback_families }};
  --font-body-style: {{ settings.type_body_font.style }};
  --font-body-weight: {{ settings.type_body_font.weight }};
  --font-body-weight-bold: {{ settings.type_body_font.weight | plus: 300 | at_most: 1000 }};

  --font-heading-family: {{ settings.type_header_font.family }}, {{ settings.type_header_font.fallback_families }};
  --font-heading-style: {{ settings.type_header_font.style }};

```

```

--font-heading-weight: {{ settings.type_header_font.weight }};

--font-body-scale: {{ settings.body_scale | divided_by: 100.0 }};
--font-heading-scale: {{ settings.heading_scale | times: 1.0 | divided_by: settings.body_scale }};

--media-padding: {{ settings.media_padding }}px;
--media-border-opacity: {{ settings.media_border_opacity | divided_by: 100.0 }};
--media-border-width: {{ settings.media_border_thickness }}px;
--media-radius: {{ settings.media_radius }}px;
--media-shadow-opacity: {{ settings.media_shadow_opacity | divided_by: 100.0 }};
--media-shadow-horizontal-offset: {{ settings.media_shadow_horizontal_offset }}px;
--media-shadow-vertical-offset: {{ settings.media_shadow_vertical_offset }}px;
--media-shadow-blur-radius: {{ settings.media_shadow_blur }}px;
--media-shadow-visible: {% if settings.media_shadow_opacity > 0 %}1{% else %}0{% endif %};

--page-width: {{ settings.page_width | divided_by: 10 }}rem;
--page-width-margin: {% if settings.page_width == '1600' %}2{% else %}0{% endif %}rem;

--product-card-image-padding: {{ settings.card_image_padding | divided_by: 10.0 }}rem;
--product-card-corner-radius: {{ settings.card_corner_radius | divided_by: 10.0 }}rem;
--product-card-text-alignment: {{ settings.card_text_alignment }};
--product-card-border-width: {{ settings.card_border_thickness | divided_by: 10.0 }}rem;
--product-card-border-opacity: {{ settings.card_border_opacity | divided_by: 100.0 }};
--product-card-shadow-opacity: {{ settings.card_shadow_opacity | divided_by: 100.0 }};
--product-card-shadow-visible: {% if settings.card_shadow_opacity > 0 %}1{% else %}0{% endif %};
--product-card-shadow-horizontal-offset: {{ settings.card_shadow_horizontal_offset | divided_by: 10.0 }}rem;
--product-card-shadow-vertical-offset: {{ settings.card_shadow_vertical_offset | divided_by: 10.0 }}rem;
--product-card-shadow-blur-radius: {{ settings.card_shadow_blur | divided_by: 10.0 }}rem;

--collection-card-image-padding: {{ settings.collection_card_image_padding | divided_by: 10.0 }}rem;
--collection-card-corner-radius: {{ settings.collection_card_corner_radius | divided_by: 10.0 }}rem;
--collection-card-text-alignment: {{ settings.collection_card_text_alignment }};
--collection-card-border-width: {{ settings.collection_card_border_thickness | divided_by: 10.0 }}rem;
--collection-card-border-opacity: {{ settings.collection_card_border_opacity | divided_by: 100.0 }};
--collection-card-shadow-opacity: {{ settings.collection_card_shadow_opacity | divided_by: 100.0 }};
--collection-card-shadow-visible: {% if settings.collection_card_shadow_opacity > 0 %}1{% else %}0{% endif
% }};
--collection-card-shadow-horizontal-offset: {{ settings.collection_card_shadow_horizontal_offset | divided_by:
10.0 }}rem;
--collection-card-shadow-vertical-offset: {{ settings.collection_card_shadow_vertical_offset | divided_by: 10.0
}}rem;
--collection-card-shadow-blur-radius: {{ settings.collection_card_shadow_blur | divided_by: 10.0 }}rem;

--blog-card-image-padding: {{ settings.blog_card_image_padding | divided_by: 10.0 }}rem;
--blog-card-corner-radius: {{ settings.blog_card_corner_radius | divided_by: 10.0 }}rem;
--blog-card-text-alignment: {{ settings.blog_card_text_alignment }};
--blog-card-border-width: {{ settings.blog_card_border_thickness | divided_by: 10.0 }}rem;
--blog-card-border-opacity: {{ settings.blog_card_border_opacity | divided_by: 100.0 }};
--blog-card-shadow-opacity: {{ settings.blog_card_shadow_opacity | divided_by: 100.0 }};
--blog-card-shadow-visible: {% if settings.blog_card_shadow_opacity > 0 %}1{% else %}0{% endif %};
--blog-card-shadow-horizontal-offset: {{ settings.blog_card_shadow_horizontal_offset | divided_by: 10.0 }}rem;
--blog-card-shadow-vertical-offset: {{ settings.blog_card_shadow_vertical_offset | divided_by: 10.0 }}rem;
--blog-card-shadow-blur-radius: {{ settings.blog_card_shadow_blur | divided_by: 10.0 }}rem;

--badge-corner-radius: {{ settings.badge_corner_radius | divided_by: 10.0 }}rem;

--popup-border-width: {{ settings.popup_border_thickness }}px;
--popup-border-opacity: {{ settings.popup_border_opacity | divided_by: 100.0 }};
--popup-corner-radius: {{ settings.popup_corner_radius }}px;
--popup-shadow-opacity: {{ settings.popup_shadow_opacity | divided_by: 100.0 }};
--popup-shadow-horizontal-offset: {{ settings.popup_shadow_horizontal_offset }}px;

```

```

--popup-shadow-vertical-offset: {{ settings.popup_shadow_vertical_offset }}px;
--popup-shadow-blur-radius: {{ settings.popup_shadow_blur }}px;

--drawer-border-width: {{ settings.drawer_border_thickness }}px;
--drawer-border-opacity: {{ settings.drawer_border_opacity | divided_by: 100.0 }};
--drawer-shadow-opacity: {{ settings.drawer_shadow_opacity | divided_by: 100.0 }};
--drawer-shadow-horizontal-offset: {{ settings.drawer_shadow_horizontal_offset }}px;
--drawer-shadow-vertical-offset: {{ settings.drawer_shadow_vertical_offset }}px;
--drawer-shadow-blur-radius: {{ settings.drawer_shadow_blur }}px;

--spacing-sections-desktop: {{ settings.spacing_sections }}px;
--spacing-sections-mobile: {% if settings.spacing_sections < 24 %}{{ settings.spacing_sections }}{% else %}{{
settings.spacing_sections | times: 0.7 | round | at_least: 20 }}{% endif %}px;

--grid-desktop-vertical-spacing: {{ settings.spacing_grid_vertical }}px;
--grid-desktop-horizontal-spacing: {{ settings.spacing_grid_horizontal }}px;
--grid-mobile-vertical-spacing: {{ settings.spacing_grid_vertical | divided_by: 2 }}px;
--grid-mobile-horizontal-spacing: {{ settings.spacing_grid_horizontal | divided_by: 2 }}px;

--text-boxes-border-opacity: {{ settings.text_boxes_border_opacity | divided_by: 100.0 }};
--text-boxes-border-width: {{ settings.text_boxes_border_thickness }}px;
--text-boxes-radius: {{ settings.text_boxes_radius }}px;
--text-boxes-shadow-opacity: {{ settings.text_boxes_shadow_opacity | divided_by: 100.0 }};
--text-boxes-shadow-visible: {% if settings.text_boxes_shadow_opacity > 0 %}1{% else %}0{% endif %};
--text-boxes-shadow-horizontal-offset: {{ settings.text_boxes_shadow_horizontal_offset }}px;
--text-boxes-shadow-vertical-offset: {{ settings.text_boxes_shadow_vertical_offset }}px;
--text-boxes-shadow-blur-radius: {{ settings.text_boxes_shadow_blur }}px;

--buttons-radius: {{ settings.buttons_radius }}px;
--buttons-radius-outset: {% if settings.buttons_radius > 0 %}{{ settings.buttons_radius | plus:
settings.buttons_border_thickness }}{% else %}0{% endif %}px;
--buttons-border-width: {% if settings.buttons_border_opacity > 0 %}{{ settings.buttons_border_thickness }}{%
else %}0{% endif %}px;
--buttons-border-opacity: {{ settings.buttons_border_opacity | divided_by: 100.0 }};
--buttons-shadow-opacity: {{ settings.buttons_shadow_opacity | divided_by: 100.0 }};
--buttons-shadow-visible: {% if settings.buttons_shadow_opacity > 0 %}1{% else %}0{% endif %};
--buttons-shadow-horizontal-offset: {{ settings.buttons_shadow_horizontal_offset }}px;
--buttons-shadow-vertical-offset: {{ settings.buttons_shadow_vertical_offset }}px;
--buttons-shadow-blur-radius: {{ settings.buttons_shadow_blur }}px;
--buttons-border-offset: {% if settings.buttons_radius > 0 or settings.buttons_shadow_opacity > 0 %}0.3{% else
%}0{% endif %}px;

--inputs-radius: {{ settings.inputs_radius }}px;
--inputs-border-width: {{ settings.inputs_border_thickness }}px;
--inputs-border-opacity: {{ settings.inputs_border_opacity | divided_by: 100.0 }};
--inputs-shadow-opacity: {{ settings.inputs_shadow_opacity | divided_by: 100.0 }};
--inputs-shadow-horizontal-offset: {{ settings.inputs_shadow_horizontal_offset }}px;
--inputs-margin-offset: {% if settings.inputs_shadow_vertical_offset != 0 and settings.inputs_shadow_opacity > 0
%}{{ settings.inputs_shadow_vertical_offset | abs }}{% else %}0{% endif %}px;
--inputs-shadow-vertical-offset: {{ settings.inputs_shadow_vertical_offset }}px;
--inputs-shadow-blur-radius: {{ settings.inputs_shadow_blur }}px;
--inputs-radius-outset: {% if settings.inputs_radius > 0 %}{{ settings.inputs_radius | plus:
settings.inputs_border_thickness }}{% else %}0{% endif %}px;

--variant-pills-radius: {{ settings.variant_pills_radius }}px;
--variant-pills-border-width: {{ settings.variant_pills_border_thickness }}px;
--variant-pills-border-opacity: {{ settings.variant_pills_border_opacity | divided_by: 100.0 }};
--variant-pills-shadow-opacity: {{ settings.variant_pills_shadow_opacity | divided_by: 100.0 }};
--variant-pills-shadow-horizontal-offset: {{ settings.variant_pills_shadow_horizontal_offset }}px;
--variant-pills-shadow-vertical-offset: {{ settings.variant_pills_shadow_vertical_offset }}px;
--variant-pills-shadow-blur-radius: {{ settings.variant_pills_shadow_blur }}px;

```

```

}

*,
*::before,
*::after {
  box-sizing: inherit;
}

html {
  box-sizing: border-box;
  font-size: calc(var(--font-body-scale) * 62.5%);
  height: 100%;
}

body {
  display: grid;
  grid-template-rows: auto auto 1fr auto;
  grid-template-columns: 100%;
  min-height: 100%;
  margin: 0;
  font-size: 1.5rem;
  letter-spacing: 0.06rem;
  line-height: calc(1 + 0.8 / var(--font-body-scale));
  font-family: var(--font-body-family);
  font-style: var(--font-body-style);
  font-weight: var(--font-body-weight);
}

@media screen and (min-width: 750px) {
  body {
    font-size: 1.6rem;
  }
}
{% endstyle % }

{{ 'base.css' | asset_url | stylesheet_tag }}

{%- unless settings.type_body_font.system? -% }
  {% comment % }theme-check-disable AssetPreload{% endcomment % }
  <link rel="preload" as="font" href="{{ settings.type_body_font | font_url }}" type="font/woff2" crossorigin>
  {% comment % }theme-check-enable AssetPreload{% endcomment % }
{%- endunless -% }

{%- unless settings.type_header_font.system? -% }
  {% comment % }theme-check-disable AssetPreload{% endcomment % }
  <link rel="preload" as="font" href="{{ settings.type_header_font | font_url }}" type="font/woff2" crossorigin>
  {% comment % }theme-check-enable AssetPreload{% endcomment % }
{%- endunless -% }

{%- if localization.available_countries.size > 1 or localization.available_languages.size > 1 -% }
  {{ 'component-localization-form.css' | asset_url | stylesheet_tag: preload: true }}
  <script src="{{ 'localization-form.js' | asset_url }}" defer="defer"></script>
{%- endif -% }

{%- if settings.predictive_search_enabled -% }
<link
  rel="stylesheet"
  href="{{ 'component-predictive-search.css' | asset_url }}"
  media="print"
  onload="this.media='all'"
>
{%- endif -% }

```



```

<script>
  if (Shopify.designMode) {
    document.documentElement.classList.add('shopify-design-mode');
  }
</script>
</head>

<body class="gradient{% if settings.animations_hover_elements != 'none' %} animate--hover-{{
settings.animations_hover_elements }}{% endif %}">
  <a class="skip-to-content-link button visually-hidden" href="#MainContent">
    {{ 'accessibility.skip_to_text' | t }}
  </a>

  {%- if settings.cart_type == 'drawer' -%}
    {%- render 'cart-drawer' -%}
  {%- endif -%}

  {% sections 'header-group' %}

  <main id="MainContent" class="content-for-layout focus-none" role="main" tabindex="-1">
    {{ content_for_layout }}
  </main>

  {% sections 'footer-group' %}

  <ul hidden>
    <li id="ally-refresh-page-message">{{ 'accessibility.refresh_page' | t }}</li>
    <li id="ally-new-window-message">{{ 'accessibility.link_messages.new_window' | t }}</li>
  </ul>

  <script>
    window.shopUrl = '{{ request.origin }}';
    window.routes = {
      cart_add_url: '{{ routes.cart_add_url }}',
      cart_change_url: '{{ routes.cart_change_url }}',
      cart_update_url: '{{ routes.cart_update_url }}',
      cart_url: '{{ routes.cart_url }}',
      predictive_search_url: '{{ routes.predictive_search_url }}',
    };

    window.cartStrings = {
      error: `{{ 'sections.cart.cart_error' | t }}`,
      quantityError: `{{ 'sections.cart.cart_quantity_error_html' | t: quantity: '[quantity]' }}`,
    };

    window.variantStrings = {
      addToCart: `{{ 'products.product.add_to_cart' | t }}`,
      soldOut: `{{ 'products.product.sold_out' | t }}`,
      unavailable: `{{ 'products.product.unavailable' | t }}`,
      unavailable_with_option: `{{ 'products.product.value_unavailable' | t: option_value: '[value]' }}`,
    };

    window.quickOrderListStrings = {
      itemsAdded: `{{ 'sections.quick_order_list.items_added.other' | t: quantity: '[quantity]' }}`,
      itemAdded: `{{ 'sections.quick_order_list.items_added.one' | t: quantity: '[quantity]' }}`,
      itemsRemoved: `{{ 'sections.quick_order_list.items_removed.other' | t: quantity: '[quantity]' }}`,
      itemRemoved: `{{ 'sections.quick_order_list.items_removed.one' | t: quantity: '[quantity]' }}`,
      viewCart: `{{- 'sections.quick_order_list.view_cart' | t -}}`,
      each: `{{- 'sections.quick_order_list.each' | t: money: '[money]' }}`,
      min_error: `{{- 'sections.quick_order_list.min_error' | t: min: '[min]' }}`,
    };
  </script>

```

```

    max_error: `{{- 'sections.quick_order_list.max_error' | t: max: '[max]' }}`,
    step_error: `{{- 'sections.quick_order_list.step_error' | t: step: '[step]' }}`,
  };

  window.accessibilityStrings = {
    imageAvailable: `{{ 'products.product.media.image_available' | t: index: '[index]' }}`,
    shareSuccess: `{{ 'general.share.success_message' | t }}`,
    pauseSlideshow: `{{ 'sections.slideshow.pause_slideshow' | t }}`,
    playSlideshow: `{{ 'sections.slideshow.play_slideshow' | t }}`,
    recipientFormExpanded: `{{ 'recipient.form.expanded' | t }}`,
    recipientFormCollapsed: `{{ 'recipient.form.collapsed' | t }}`,
    countrySelectorSearchCount: `{{ 'localization.country_results_count' | t: count: '[count]' }}`,
  };
</script>

{% - if settings.predictive_search_enabled -% }
  <script src="{{ 'predictive-search.js' | asset_url }}" defer="defer"></script>
{% - endif -% }
</body>
</html>

```

base.css

```

:root {
  --alpha-button-background: 1;
  --alpha-button-border: 1;
  --alpha-link: 0.85;
  --alpha-badge-border: 0.1;
}

.product-card-wrapper .card,
.contains-card--product {
  --border-radius: var(--product-card-corner-radius);
  --border-width: var(--product-card-border-width);
  --border-opacity: var(--product-card-border-opacity);
  --shadow-horizontal-offset: var(--product-card-shadow-horizontal-offset);
  --shadow-vertical-offset: var(--product-card-shadow-vertical-offset);
  --shadow-blur-radius: var(--product-card-shadow-blur-radius);
  --shadow-opacity: var(--product-card-shadow-opacity);
  --shadow-visible: var(--product-card-shadow-visible);
  --image-padding: var(--product-card-image-padding);
  --text-alignment: var(--product-card-text-alignment);
}

.collection-card-wrapper .card,
.contains-card--collection {
  --border-radius: var(--collection-card-corner-radius);
  --border-width: var(--collection-card-border-width);
  --border-opacity: var(--collection-card-border-opacity);
  --shadow-horizontal-offset: var(--collection-card-shadow-horizontal-offset);
  --shadow-vertical-offset: var(--collection-card-shadow-vertical-offset);
  --shadow-blur-radius: var(--collection-card-shadow-blur-radius);
  --shadow-opacity: var(--collection-card-shadow-opacity);
  --shadow-visible: var(--collection-card-shadow-visible);
  --image-padding: var(--collection-card-image-padding);
  --text-alignment: var(--collection-card-text-alignment);
}

.article-card-wrapper .card,
.contains-card--article {
  --border-radius: var(--blog-card-corner-radius);
  --border-width: var(--blog-card-border-width);
}

```

```

--border-opacity: var(--blog-card-border-opacity);
--shadow-horizontal-offset: var(--blog-card-shadow-horizontal-offset);
--shadow-vertical-offset: var(--blog-card-shadow-vertical-offset);
--shadow-blur-radius: var(--blog-card-shadow-blur-radius);
--shadow-opacity: var(--blog-card-shadow-opacity);
--shadow-visible: var(--blog-card-shadow-visible);
--image-padding: var(--blog-card-image-padding);
--text-alignment: var(--blog-card-text-alignment);
}

.contains-content-container,
.content-container {
--border-radius: var(--text-boxes-radius);
--border-width: var(--text-boxes-border-width);
--border-opacity: var(--text-boxes-border-opacity);
--shadow-horizontal-offset: var(--text-boxes-shadow-horizontal-offset);
--shadow-vertical-offset: var(--text-boxes-shadow-vertical-offset);
--shadow-blur-radius: var(--text-boxes-shadow-blur-radius);
--shadow-opacity: var(--text-boxes-shadow-opacity);
--shadow-visible: var(--text-boxes-shadow-visible);
}

.contains-media,
.global-media-settings {
--border-radius: var(--media-radius);
--border-width: var(--media-border-width);
--border-opacity: var(--media-border-opacity);
--shadow-horizontal-offset: var(--media-shadow-horizontal-offset);
--shadow-vertical-offset: var(--media-shadow-vertical-offset);
--shadow-blur-radius: var(--media-shadow-blur-radius);
--shadow-opacity: var(--media-shadow-opacity);
--shadow-visible: var(--media-shadow-visible);
}

/* base */

.page-width {
max-width: var(--page-width);
margin: 0 auto;
padding: 0 1.5rem;
}

body:has(.section-header .drawer-menu) .announcement-bar-section .page-width {
max-width: 100%;
}

.page-width.drawer-menu {
max-width: 100%;
}

.page-width-desktop {
padding: 0;
margin: 0 auto;
}

.utility-bar__grid.page-width {
padding-left: 3rem;
padding-right: 3rem;
}

@media screen and (min-width: 750px) {

```

```

.page-width {
  padding: 0 5rem;
}

.header.page-width,
.utility-bar__grid.page-width {
  padding-left: 3.2rem;
  padding-right: 3.2rem;
}

.page-width--narrow {
  padding: 0 9rem;
}

.page-width-desktop {
  padding: 0;
}

.page-width-tablet {
  padding: 0 5rem;
}
}

@media screen and (min-width: 990px) {
.header:not(.drawer-menu).page-width {
  padding-left: 5rem;
  padding-right: 5rem;
}

.page-width--narrow {
  max-width: 72.6rem;
  padding: 0;
}

.page-width-desktop {
  max-width: var(--page-width);
  padding: 0 5rem;
}
}

.isolate {
  position: relative;
  z-index: 0;
}

.section + .section {
  margin-top: var(--spacing-sections-mobile);
}

@media screen and (min-width: 750px) {
.section + .section {
  margin-top: var(--spacing-sections-desktop);
}
}

.element-margin-top {
  margin-top: 5rem;
}

@media screen and (min-width: 750px) {
.element-margin {

```

```

    margin-top: calc(5rem + var(--page-width-margin));
  }
}

.background-secondary {
  background-color: rgba(var(--color-foreground), 0.04);
}

.grid-auto-flow {
  display: grid;
  grid-auto-flow: column;
}

.page-margin,
.shopify-challenge__container {
  margin: 7rem auto;
}

.rte-width {
  max-width: 82rem;
  margin: 0 auto 2rem;
}

.list-unstyled {
  margin: 0;
  padding: 0;
  list-style: none;
}

.hidden {
  display: none !important;
}

```

password.liquid

```

<!doctype html>
<html class="js full-height" lang="{{ request.locale.iso_code }}">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <meta name="theme-color" content="">
  <link rel="canonical" href="{{ canonical_url }}">

  {% if settings.favicon != blank -% }
    <link rel="icon" type="image/png" href="{{ settings.favicon | image_url: width: 32, height: 32 }}">
  {% endif -% }

  {% unless settings.type_header_font.system? -% }
    <link rel="preconnect" href="https://fonts.shopifycdn.com" crossorigin>
  {% endunless -% }

  <title>{{ shop.name }}</title>

  <meta name="description" content="{{ page_description | escape }}">

  {% render 'meta-tags' % }

  {{ content_for_header }}

  {% liquid

```

```

assign body_font_bold = settings.type_body_font | font_modify: 'weight', 'bold'
assign body_font_italic = settings.type_body_font | font_modify: 'style', 'italic'
assign body_font_bold_italic = body_font_bold | font_modify: 'style', 'italic'
% }

{% style %}
{{ settings.type_body_font | font_face: font_display: 'swap' }}
{{ body_font_bold | font_face: font_display: 'swap' }}
{{ body_font_italic | font_face: font_display: 'swap' }}
{{ body_font_bold_italic | font_face: font_display: 'swap' }}
{{ settings.type_header_font | font_face: font_display: 'swap' }}

{% for scheme in settings.color_schemes -%}
  {% assign scheme_classes = scheme_classes | append: ', .color-' | append: scheme.id %}
  {% if forloop.index == 1 -%}
    :root,
  {%- endif %}
  .color-{{ scheme.id }} {
    --color-background: {{ scheme.settings.background.red }},{{ scheme.settings.background.green }},{{
scheme.settings.background.blue }};
    {% if scheme.settings.background_gradient != empty %}
      --gradient-background: {{ scheme.settings.background_gradient }};
    {% else %}
      --gradient-background: {{ scheme.settings.background }};
    {% endif %}
    --color-foreground: {{ scheme.settings.text.red }},{{ scheme.settings.text.green }},{{ scheme.settings.text.blue
}}};
    --color-shadow: {{ scheme.settings.shadow.red }},{{ scheme.settings.shadow.green }},{{
scheme.settings.shadow.blue }};
    --color-button: {{ scheme.settings.button.red }},{{ scheme.settings.button.green }},{{
scheme.settings.button.blue }};
    --color-button-text: {{ scheme.settings.button_label.red }},{{ scheme.settings.button_label.green }},{{
scheme.settings.button_label.blue }};
    --color-secondary-button: {{ scheme.settings.background.red }},{{ scheme.settings.background.green }},{{
scheme.settings.background.blue }};
    --color-secondary-button-text: {{ scheme.settings.secondary_button_label.red }},{{
scheme.settings.secondary_button_label.green }},{{ scheme.settings.secondary_button_label.blue }};
    --color-link: {{ scheme.settings.secondary_button_label.red }},{{ scheme.settings.secondary_button_label.green
}},{{ scheme.settings.secondary_button_label.blue }};
    --color-badge-foreground: {{ scheme.settings.text.red }},{{ scheme.settings.text.green }},{{
scheme.settings.text.blue }};
    --color-badge-background: {{ scheme.settings.background.red }},{{ scheme.settings.background.green }},{{
scheme.settings.background.blue }};
    --color-badge-border: {{ scheme.settings.text.red }},{{ scheme.settings.text.green }},{{ scheme.settings.text.blue
}}};
    --payment-terms-background-color: rgb({{ scheme.settings.background.rgb }});
  }
{% endfor %}

{{ scheme_classes | prepend: 'body' }} {
  color: rgba(var(--color-foreground), 0.75);
  background-color: rgb(var(--color-background));
}

:root {
  --font-body-family: {{ settings.type_body_font.family }}, {{ settings.type_body_font.fallback_families }};
  --font-body-style: {{ settings.type_body_font.style }};
  --font-body-weight: {{ settings.type_body_font.weight }};

  --font-heading-family: {{ settings.type_header_font.family }}, {{ settings.type_header_font.fallback_families }};
  --font-heading-style: {{ settings.type_header_font.style }};

```

```

--font-heading-weight: {{ settings.type_header_font.weight }};

--font-body-scale: {{ settings.body_scale | divided_by: 100.0 }};
--font-heading-scale: {{ settings.heading_scale | times: 1.0 | divided_by: settings.body_scale }};
--media-padding: {{ settings.media_padding }}px;
--media-border-opacity: {{ settings.media_border_opacity | divided_by: 100.0 }};
--media-border-width: {{ settings.media_border_thickness }}px;
--media-radius: {{ settings.media_radius }}px;
--media-shadow-opacity: {{ settings.media_shadow_opacity | divided_by: 100.0 }};
--media-shadow-horizontal-offset: {{ settings.media_shadow_horizontal_offset }}px;
--media-shadow-vertical-offset: {{ settings.media_shadow_vertical_offset }}px;
--media-shadow-blur-radius: {{ settings.media_shadow_blur }}px;
--media-shadow-visible: {% if settings.media_shadow_opacity > 0 %}1{% else %}0{% endif %};

--page-width: {{ settings.page_width | divided_by: 10 }}rem;
--page-width-margin: {% if settings.page_width == '1600' %}2{% else %}0{% endif %}rem;

--product-card-image-padding: {{ settings.card_image_padding | divided_by: 10.0 }}rem;
--product-card-corner-radius: {{ settings.card_corner_radius | divided_by: 10.0 }}rem;
--product-card-text-alignment: {{ settings.card_text_alignment }};
--product-card-border-width: {{ settings.card_border_thickness | divided_by: 10.0 }}rem;
--product-card-border-opacity: {{ settings.card_border_opacity | divided_by: 100.0 }};
--product-card-shadow-opacity: {{ settings.card_shadow_opacity | divided_by: 100.0 }};
--product-card-shadow-visible: {% if settings.card_shadow_opacity > 0 %}1{% else %}0{% endif %};
--product-card-shadow-horizontal-offset: {{ settings.card_shadow_horizontal_offset | divided_by: 10.0 }}rem;
--product-card-shadow-vertical-offset: {{ settings.card_shadow_vertical_offset | divided_by: 10.0 }}rem;
--product-card-shadow-blur-radius: {{ settings.card_shadow_blur | divided_by: 10.0 }}rem;

--collection-card-image-padding: {{ settings.collection_card_image_padding | divided_by: 10.0 }}rem;
--collection-card-corner-radius: {{ settings.collection_card_corner_radius | divided_by: 10.0 }}rem;
--collection-card-text-alignment: {{ settings.collection_card_text_alignment }};
--collection-card-border-width: {{ settings.collection_card_border_thickness | divided_by: 10.0 }}rem;
--collection-card-border-opacity: {{ settings.collection_card_border_opacity | divided_by: 100.0 }};
--collection-card-shadow-opacity: {{ settings.collection_card_shadow_opacity | divided_by: 100.0 }};
--collection-card-shadow-visible: {% if settings.collection_card_shadow_opacity > 0 %}1{% else %}0{% endif
% };
--collection-card-shadow-horizontal-offset: {{ settings.collection_card_shadow_horizontal_offset | divided_by:
10.0 }}rem;
--collection-card-shadow-vertical-offset: {{ settings.collection_card_shadow_vertical_offset | divided_by: 10.0
}}rem;
--collection-card-shadow-blur-radius: {{ settings.collection_card_shadow_blur | divided_by: 10.0 }}rem;

--blog-card-image-padding: {{ settings.blog_card_image_padding | divided_by: 10.0 }}rem;
--blog-card-corner-radius: {{ settings.blog_card_corner_radius | divided_by: 10.0 }}rem;
--blog-card-text-alignment: {{ settings.blog_card_text_alignment }};
--blog-card-border-width: {{ settings.blog_card_border_thickness | divided_by: 10.0 }}rem;
--blog-card-border-opacity: {{ settings.blog_card_border_opacity | divided_by: 100.0 }};
--blog-card-shadow-opacity: {{ settings.blog_card_shadow_opacity | divided_by: 100.0 }};
--blog-card-shadow-visible: {% if settings.blog_card_shadow_opacity > 0 %}1{% else %}0{% endif %};
--blog-card-shadow-horizontal-offset: {{ settings.blog_card_shadow_horizontal_offset | divided_by: 10.0 }}rem;
--blog-card-shadow-vertical-offset: {{ settings.blog_card_shadow_vertical_offset | divided_by: 10.0 }}rem;
--blog-card-shadow-blur-radius: {{ settings.blog_card_shadow_blur | divided_by: 10.0 }}rem;

--badge-corner-radius: {{ settings.badge_corner_radius | divided_by: 10.0 }}rem;

--spacing-sections-desktop: {{ settings.spacing_sections }}px;
--spacing-sections-mobile: {% if settings.spacing_sections < 24 %}{{ settings.spacing_sections }}{% else %}{{
settings.spacing_sections | times: 0.7 | round | at_least: 20 }}{% endif %}px;

--grid-desktop-vertical-spacing: {{ settings.spacing_grid_vertical }}px;
--grid-desktop-horizontal-spacing: {{ settings.spacing_grid_horizontal }}px;

```

```

--grid-mobile-vertical-spacing: {{ settings.spacing_grid_vertical | divided_by: 2 }}px;
--grid-mobile-horizontal-spacing: {{ settings.spacing_grid_horizontal | divided_by: 2 }}px;

--text-boxes-border-opacity: {{ settings.text_boxes_border_opacity | divided_by: 100.0 }};
--text-boxes-border-width: {{ settings.text_boxes_border_thickness }}px;
--text-boxes-radius: {{ settings.text_boxes_radius }}px;
--text-boxes-shadow-opacity: {{ settings.text_boxes_shadow_opacity | divided_by: 100.0 }};
--text-boxes-shadow-visible: {% if settings.text_boxes_shadow_opacity > 0 %}1{% else %}0{% endif %};
--text-boxes-shadow-horizontal-offset: {{ settings.text_boxes_shadow_horizontal_offset }}px;
--text-boxes-shadow-vertical-offset: {{ settings.text_boxes_shadow_vertical_offset }}px;
--text-boxes-shadow-blur-radius: {{ settings.text_boxes_shadow_blur }}px;

--buttons-radius: {{ settings.buttons_radius }}px;
--buttons-radius-outset: {% if settings.buttons_radius > 0 %}{{ settings.buttons_radius | plus:
settings.buttons_border_thickness }}{% else %}0{% endif %}px;
--buttons-border-width: {% if settings.buttons_border_opacity > 0 %}{{ settings.buttons_border_thickness }}{%
else %}0{% endif %}px;
--buttons-border-opacity: {{ settings.buttons_border_opacity | divided_by: 100.0 }};
--buttons-shadow-opacity: {{ settings.buttons_shadow_opacity | divided_by: 100.0 }};
--buttons-shadow-visible: {% if settings.buttons_shadow_opacity > 0 %}1{% else %}0{% endif %};
--buttons-shadow-horizontal-offset: {{ settings.buttons_shadow_horizontal_offset }}px;
--buttons-shadow-vertical-offset: {{ settings.buttons_shadow_vertical_offset }}px;
--buttons-shadow-blur-radius: {{ settings.buttons_shadow_blur }}px;
--buttons-border-offset: {% if settings.buttons_radius > 0 or settings.buttons_shadow_opacity > 0 %}0.3{% else
%}0{% endif %}px;

--inputs-radius: {{ settings.inputs_radius }}px;
--inputs-border-width: {{ settings.inputs_border_thickness }}px;
--inputs-border-opacity: {{ settings.inputs_border_opacity | divided_by: 100.0 }};
--inputs-shadow-opacity: {{ settings.inputs_shadow_opacity | divided_by: 100.0 }};
--inputs-shadow-horizontal-offset: {{ settings.inputs_shadow_horizontal_offset }}px;
--inputs-margin-offset: {% if settings.inputs_shadow_vertical_offset != 0 and settings.inputs_shadow_opacity > 0
%}{{ settings.inputs_shadow_vertical_offset | abs }}{% else %}0{% endif %}px;
--inputs-shadow-vertical-offset: {{ settings.inputs_shadow_vertical_offset }}px;
--inputs-shadow-blur-radius: {{ settings.inputs_shadow_blur }}px;
--inputs-radius-outset: {% if settings.inputs_radius > 0 %}{{ settings.inputs_radius | plus:
settings.inputs_border_thickness }}{% else %}0{% endif %}px;

--variant-pills-radius: {{ settings.variant_pills_radius }}px;
--variant-pills-border-width: {{ settings.variant_pills_border_thickness }}px;
--variant-pills-border-opacity: {{ settings.variant_pills_border_opacity | divided_by: 100.0 }};
--variant-pills-shadow-opacity: {{ settings.variant_pills_shadow_opacity | divided_by: 100.0 }};
--variant-pills-shadow-horizontal-offset: {{ settings.variant_pills_shadow_horizontal_offset }}px;
--variant-pills-shadow-vertical-offset: {{ settings.variant_pills_shadow_vertical_offset }}px;
--variant-pills-shadow-blur-radius: {{ settings.variant_pills_shadow_blur }}px;
}
{% endstyle % }

{%- unless settings.type_body_font.system? -% }
{% comment % }theme-check-disable AssetPreload{% endcomment % }
<link rel="preload" as="font" href="{{ settings.type_body_font | font_url }}" type="font/woff2" crossorigin>
{% comment % }theme-check-enable AssetPreload{% endcomment % }
{%- endunless -% }
{%- unless settings.type_header_font.system? -% }
{% comment % }theme-check-disable AssetPreload{% endcomment % }
<link rel="preload" as="font" href="{{ settings.type_header_font | font_url }}" type="font/woff2" crossorigin>
{% comment % }theme-check-enable AssetPreload{% endcomment % }
{%- endunless -% }

{{ 'section-password.css' | asset_url | stylesheet_tag }}
{{ 'base.css' | asset_url | stylesheet_tag }}

```



```

{{ 'component-list-social.css' | asset_url | stylesheet_tag }}

<script src="{{ 'global.js' | asset_url }}" defer="defer"></script>
<script src="{{ 'details-modal.js' | asset_url }}" defer="defer"></script>
<script src="{{ 'password-modal.js' | asset_url }}" defer="defer"></script>
</head>

<body class="password gradient">
  <a class="skip-to-content-link button visually-hidden" href="#MainContent">
    {{ 'accessibility.skip_to_text' | t }}
  </a>

  {% section 'main-password-header' %}

  <main id="MainContent" class="password-main">
    {{ content_for_layout }}
  </main>
  <footer>
    {% section 'main-password-footer' %}
  </footer>
  <ul hidden>
    <li id="ally-new-window-message">{{ 'accessibility.link_messages.new_window' | t }}</li>
  </ul>
</body>
</html>

```

Решта коду додається окремо на CD-диску.

ВІДГУК

керівника економічного розділу

на кваліфікаційну роботу бакалавра на тему:

«Розробка веборієнтованого додатку для продажу товарів, згенерованих

штучним інтелектом на базі платформи Shopify»

студента групи 122-20-2 Гончарова Максима Сергійовича

**Керівник економічного розділу доц.
каф. ПЕП та ПУ, к.е.н**

Л.В. Касьяненко

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
ГончаровМС_диплом.docx	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
ГончаровМС_диплом.pdf	Пояснювальна записка до кваліфікаційної роботи у форматі PDF
Програма	
ГончаровМС.rar	Архів. Містить коди програми і скомпільовану програму
Презентація	
ГончаровМС_диплом.ppt	Презентація кваліфікаційної роботи