Державний торговельно-економічний університет

Кафедра комп’ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка дизайну в Figma та верстка Web-сайту для бренду одягу DEVIALT»

Студента 4 курсу, 5 групи,
спеціальності
122 «Комп’ютерні науки» _____ *підпис студента* Шевчук Микита Олегович

Науковий керівник
кандидат технічних наук, доцент _____ *підпис керівника* Демідов Павло Георгійович

Гарант освітньої програми
доктор фізико-математичних наук, професор _____ *підпис керівника* Пурський Олег Іванович

Київ 2024

Державний торговельно-економічний університет

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

Затверджую
Зав. кафедри Пурський О.І.
«20» грудня 2023р.

Завдання

на випускну кваліфікаційну роботу (проект) студенту

Шевчуку Микиті Олеговичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)

«Розробка дизайну в Figma та верстка Web-сайту для бренду одягу DEVIALT»

Затверджена наказом ректора від *«27» листопада 2023 р. № 4175*

2. Строк здачі студентом закінченої роботи *20 травня 2024 року*

3. Цільова установка та вихідні дані до роботи

Мета роботи: створення комплексного підходу до розробки дизайну в Figma та верстки веб-сайту для бренду одягу DEVIALT.

Об'єкт дослідження: процес розробки дизайну та верстки веб-сайту для бренду одягу DEVIALT.

Предмет дослідження: технології розробки дизайну в Figma та верстки веб-сайту для бренду одягу DEVIALT.

4. Перелік графічного матеріалу_____

_____

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

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Демідов П.Г.	22.12.2023 р.	22.12.2023 р.
2	Демідов П.Г.	22.12.2023 р.	22.12.2023 р.
3	Демідов П.Г.	22.12.2023 р.	22.12.2023 р.

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

ВСТУП

РОЗДІЛ 1. Теоретичне дослідження веб-сайтів у сфері брендового одягу.

1.1. Основи створення та показу брендового одягу на ринку продажу.

1.2. Моделі, методи та алгоритми побудови веб-сайтів показу одягу.

1.3. Характеристика сучасних сайтів одягу та їх програмне забезпечення.

РОЗДІЛ 2. Проєктування веб-сайту брендового одягу.

2.1. Архітектура багатосторінкового сайту та характеристика його компонентів.

2.2. Алгоритми роботи сайту та розрахунку вартості замовлення на придбання одягу.

2.3. Проєктування дизайну сайту одягу DEVIALT у середовищі Figma.

РОЗДІЛ 3. Розробка програмного забезпечення веб-сайту бренду одягу Devialt.

3.1. Створення та характеристика програмного коду окремих модулів веб-сайту одягу.

3.2. Характеристика віконного інтерфейсу взаємодії покупця з сайтом

продажу.

3.3. Опис прикладу вхідної бази одягу, її використання та тестування для роботи сайту одягу DEVIALT.

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	*Вибір теми випускної кваліфікаційної роботи*	10.10.2023	*10.10.2023*
2	*Розробка та затвердження завдання на випускну кваліфікаційну роботу*	22.12.2023	22.12.2023
3	*Вступ*	*05.02*.2024	
4	*РОЗДІЛ 1. Теоретичне дослідження веб-сайтів у сфері брендового одягу.*	*28.02.2024*	
5	*РОЗДІЛ 2. Проєктування веб-сайту брендового одягу.*	*06.04.2024*	
6	*РОЗДІЛ 3. Розробка програмного забезпечення веб-сайту бренду одягу Devialt.*	*13.05.2024*	
7	*Висновки*	*15.05.2024*	
8	*Здача випускної кваліфікаційної роботи на кафедру науковому керівнику*	*15.05.2024*	
9	*Попередній захист випускної кваліфікаційної роботи*	*16.05.2024*	
11	*Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи*	*17.05.2024*	
12	*Представлення готової зшитої випускної кваліфікаційної роботи на кафедру*	*20.05.2024*	

13	*Публічний захист випускної кваліфікаційної роботи*	*За розкладом роботи ЕК*	

8. Дата видачі завдання «22» грудня 2023 р.

9. Керівник випускної кваліфікаційної роботи (проекту)
Демідов П.Г.
(прізвище, ініціали, підпис)

10. Гарант освітньої програми Пурський О.І.
(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник
Шевчук М. О.
(прізвище, ініціали, підпис)

12. Відгук керівника випускної кваліфікаційної роботи (проекту)

Керівник випускної кваліфікаційної роботи (проекту)
_____ _____р.
(підпис, дата)

13. Висновок про випускну кваліфікаційну роботу (проект)

Випускна кваліфікаційна робота (проект) студента Шевчука М. О.
(прізвище, ініціали)
може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми_____ Демідов П.Г
підпис, прізвище, ініціали)

Завідувач кафедри _____ Пурський О.І.
(підпис, прізвище, ініціали)
«_____»_____2024 р.

АНОТАЦІЯ

У даній випускній кваліфікаційній роботі проведено комплексний аналіз процесу розробки дизайну та верстки веб-сайту для бренду одягу DEVIALT з використанням інструменту Figma. Обґрунтовано основні принципи створення дизайну та його вплив на користувачів на основі теоретичних підходів та практичних досліджень. Запропоновано концепцію дизайну відповідно до ідентичності бренду DEVIALT, зокрема щодо кольорової палітри, типографіки та композиції. Розроблено інтерфейс веб-сайту, який забезпечує ефективну навігацію та зручний користувацький досвід. Практична реалізація включає відтворення структури та дизайну на платформі HTML/CSS/JavaScript.

Ключові слова: розробка дизайну, Figma, верстка веб-сайту, брендування, DEVIALT.

ABSTRACT

This graduation qualification work provides a comprehensive analysis of the process of designing and developing a website for the clothing brand DEVIALT using the Figma tool. The main principles of design creation and its impact on users have been substantiated based on theoretical approaches and practical research. A design concept has been proposed in accordance with the identity of the DEVIALT brand, particularly regarding the color palette, typography, and composition. The website interface has been developed to ensure effective navigation and a convenient user experience. The practical implementation includes reproducing the structure and design on the HTML/CSS/JavaScript platform.

Keywords: design development, Figma, website layout, branding, DEVIALT.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CSS (Cascading Style Sheets) – каскадні таблиці стилів, мова опису зовнішнього вигляду веб-сторінок.

HTML (HyperText Markup Language) – стандартна мова розмітки документів для створення веб-сторінок.

JavaScript – мова програмування, яка використовується для додавання інтерактивності до веб-сторінок.

Figma – онлайн-інструмент для розробки інтерфейсів користувача та прототипування.

UX (User Experience) – користувацький досвід, сукупність емоцій і відчуттів користувача від використання продукту.

UI (User Interface) – користувацький інтерфейс, візуальна частина програми, з якою взаємодіє користувач.

SEO (Search Engine Optimization) – оптимізація для пошукових систем, процес поліпшення видимості веб-сторінок у пошукових системах.

UI компонент – незалежний блок користувацького інтерфейсу, який може бути повторно використаний у різних частинах веб-сайту або додатку.

ЗМІСТ

ВСТУП

Сучасний розвиток технологій та зростання конкуренції на ринку одягу вимагає від брендів не лише високої якості продукції, але й ефективної віртуальної присутності. Забезпечення стабільного успіху бренду DEVIALT у сфері моди потребує вивчення та використання передових інструментів дизайну та веб-розробки.

Мета роботи полягає в розробці комплексного підходу до дизайну та верстки веб-сайту для бренду одягу DEVIALT з використанням інструменту Figma. Для досягнення цієї мети вирішувалися наступні **завдання:**

- Аналіз сучасних тенденцій у дизайні веб-сайтів для модних брендів.
- Розробка унікального дизайну, що відповідає ідентичності бренду DEVIALT та забезпечує приємний користувацький досвід.
- Використання інструменту Figma для створення макетів веб-сайту та його елементів.
- Верстка створеного дизайну у веб-форматі з використанням HTML/CSS.

Дослідження та розробка веб-сайту для бренду DEVIALT актуальне, оскільки він дозволить підвищити його впізнаваність та конкурентоспроможність на ринку моди, залучити нових клієнтів та забезпечити зручний спосіб комунікації з аудиторією.

Об'єктом дослідження є процеси розробки дизайну та верстки веб-сайту для бренду одягу DEVIALT.

Предметом дослідження є методи та інструменти реалізації цих процесів з використанням інструменту Figma та веб-технологій.

Методи дослідження базуються на аналізі сучасних підходів до дизайну та веб-розробки, використанні системного підходу до розробки концепції сайту та елементів брендингу, а також на практичній реалізації цих концепцій у вигляді веб-сайту.

Випускна кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел та додатків.

РОЗДІЛ 1.
ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ ВЕБ-САЙТІВ У СФЕРІ БРЕНДОВОГО ОДЯГУ

1.1. Основи створення та показу брендового одягу на ринку продажу

Брендовий одяг займає важливе місце у сучасній модній індустрії, яка характеризується високою конкуренцією та швидкими змінами тенденцій. Процес створення та просування бренду одягу на ринку потребує комплексного підходу, що включає визначення цільової аудиторії, розробку унікального стилю та ефективну маркетингову стратегію. У цьому параграфі розглядаються основні аспекти створення брендового одягу та методи його показу на ринку.

Створення успішного бренду одягу потребує виконання наступних етапів:

- визначення цільової аудиторії. Це включає аналіз демографічних та психологічних характеристик потенційних клієнтів, їхніх потреб, уподобань та стилю життя. Наприклад, молодіжна аудиторія може бути зацікавлена у яскравих та сміливих дизайнах, тоді як старші клієнти віддають перевагу класичним та елегантним моделям. Визначення цільової аудиторії дозволяє створити продукцію, яка найкраще відповідає запитам споживачів і підвищує шанси на успіх на ринку;
- розробка унікального стилю бренду, який відрізнятиме його від конкурентів. Це включає вибір кольорової гами, типографіки, логотипу та інших візуальних елементів, що створюють цілісний образ бренду. Наприклад, бренд Chanel відомий своїм елегантним та розкішним стилем, тоді як Nike асоціюється з активністю та спортивним духом. Унікальний стиль допомагає бренду бути впізнаваним і запам'ятовуватися споживачам;

- важливим етапом є розробка маркетингової стратегії. Це включає вибір каналів комунікації з клієнтами, створення рекламних кампаній, участь у модних показах та співпраця з інфлюенсерами. Соціальні мережі, такі як Instagram та TikTok, стають все більш важливими інструментами для просування модних брендів, оскільки вони дозволяють безпосередньо взаємодіяти з цільовою аудиторією та отримувати зворотний зв'язок.

Використання соціальних мереж для просування бренду дозволяє значно збільшити охоплення аудиторії та залучити нових клієнтів. На рис. 1.1 наведено схематичне представлення показаних вище етапів.

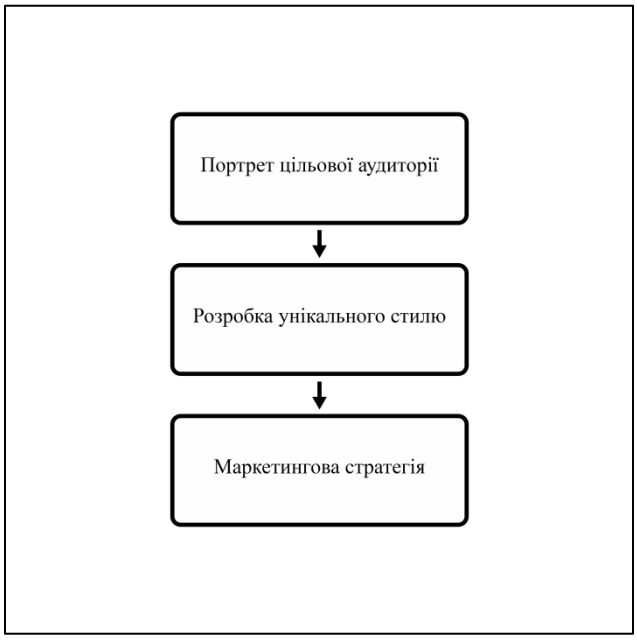


Рис. 1.1. Схематичне представлення важливих етапів створення бренду одягу

Веб-сайт бренду одягу є одним з основних інструментів для його просування. Він виконує кілька важливих функцій: інформаційну, комунікаційну та продажну.

Інформаційна функція забезпечує доступ до інформації про бренд, його історію, філософію та нові колекції. Важливо, щоб інформація була представлена зрозуміло та привабливо для відвідувачів.

Комунікаційна функція включає забезпечення зворотного зв'язку з клієнтами через форми контактів, соціальні мережі та блоги. Це допомагає зміцнювати відносини з клієнтами та отримувати відгуки про продукцію. Наприклад, блог на сайті може містити статті про нові тенденції в моді, поради щодо стилю та догляду за одягом, що залучає аудиторію та підвищує її інтерес до бренду.

Продажна функція веб-сайту реалізується через інтернет-магазин, який дозволяє здійснювати покупки безпосередньо на сайті. Для цього важливо забезпечити зручний та безпечний процес покупки, включаючи різні способи оплати та доставки. Інтернет-магазин має бути простим у використанні, з інтуїтивно зрозумілою навігацією та якісними фотографіями товарів. Додатково, важливо інтегрувати різні методи оплати, такі як кредитні картки, електронні гаманці та інші популярні платіжні системи. На рис 1.2 візуально зображена структура основних функцій сайту.

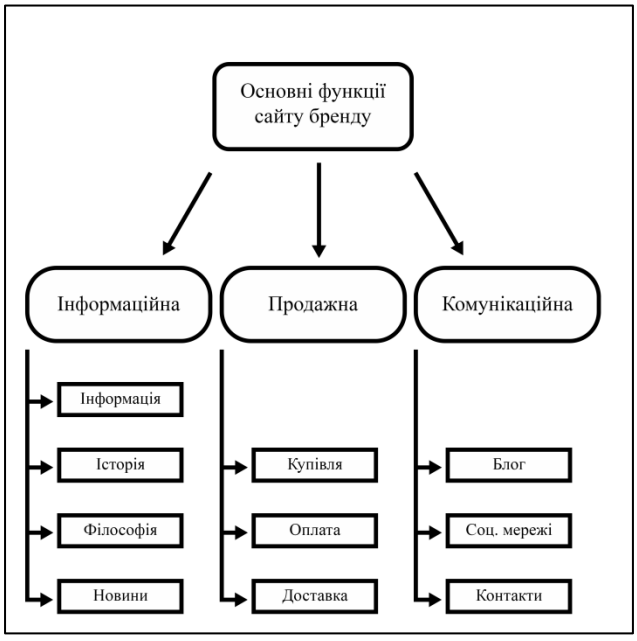


Рис. 1.2. Структура представлення основних функцій сайту бренду

Візуальна привабливість веб-сайту є ключовим фактором успіху. Дизайн у Figma дозволяє створювати високоякісні макети, які легко

адаптуються до потреб користувачів. Основні елементи UX/UI дизайну включають зручність навігації, візуальні елементи та адаптивний дизайн. Зручність навігації забезпечується логічною та інтуїтивною структурою сайту, яка дозволяє швидко знайти потрібну інформацію або товар.

Візуальні елементи, такі як фотографії високої якості, відео та інтерактивні елементи, допомагають краще представити продукцію та залучити увагу відвідувачів. Фотографії повинні бути професійними та показувати одяг з різних ракурсів, а відео можуть демонструвати колекції в русі або показувати процес створення продукції.

Адаптивний дизайн забезпечує коректне відображення сайту на різних пристроях, таких як смартфони, планшети та десктопи. Це важливо, оскільки все більше користувачів здійснюють покупки та переглядають сайти саме з мобільних пристроїв. Тому веб-сайт повинен автоматично підлаштовуватися під розміри екрану користувача, забезпечуючи зручність використання на будь-якому пристрої.

Впровадження сучасних технологій також відіграє важливу роль у створенні та просуванні брендового одягу. Використання технологій штучного інтелекту (AI) та машинного навчання (ML) дозволяє покращити персоналізацію покупок, наприклад, рекомендувати продукти на основі попередніх покупок або переглядів користувачів. Це підвищує ймовірність покупки та задоволеність клієнтів.

Додатково, інтеграція доповненої реальності (AR) дозволяє створювати віртуальні примірочні, де клієнти можуть "приміряти" одяг, не виходячи з дому. Це може значно зменшити кількість повернень товарів та підвищити задоволеність клієнтів.

Соціальні мережі є важливим інструментом для просування брендового одягу. Вони дозволяють брендам взаємодіяти зі своїми клієнтами, ділитися

новинами, публікувати фотографії нових колекцій та отримувати зворотний зв'язок. Наприклад, Instagram дозволяє брендам публікувати високоякісні зображення продукції, проводити конкурси та співпрацювати з інфлюенсерами, які мають велику аудиторію підписників.

Контент-маркетинг також відіграє важливу роль у просуванні брендового одягу. Написання блогів, створення відео-контенту, електронні розсилки та інші форми контенту дозволяють залучити аудиторію та підвищити її інтерес до бренду. Наприклад, блог може містити статті про нові тенденції в моді, поради щодо стилю та догляду за одягом, що допомагає створити лояльну аудиторію.

1.2. Моделі, методи та алгоритми побудови веб-сайтів показу одягу

Побудова веб-сайтів для показу одягу потребує застосування різноманітних моделей, методів та алгоритмів, що забезпечують зручність використання, ефективність роботи та привабливий зовнішній вигляд. У цьому параграфі розглядаються основні підходи до створення веб-сайтів, які дозволяють брендам одягу досягати високих результатів у просуванні своєї продукції.

Моделі побудови веб-сайтів для показу одягу можна поділити на кілька основних категорій, серед яких слід виділити модель багатосторінкового сайту (MPA) та модель односторінкового сайту (SPA).

Багатосторінковий сайт (MPA) є традиційною моделлю побудови веб-сайтів, яка передбачає розподіл контенту на кілька сторінок. Кожна сторінка має свою URL-адресу і завантажується окремо. Ця модель зручна для великих сайтів, які містять багато інформації та різні розділи. Наприклад, інтернет-магазини одягу часто використовують MPA для організації каталогу товарів, де кожна категорія та підкатегорія мають окрему сторінку.

Переваги MPA включають можливість покращення SEO, оскільки кожна сторінка може бути оптимізована під окремі ключові слова. Крім того, MPA забезпечує зручність навігації для користувачів, які звикли до традиційної структури веб-сайтів. Однак MPA може мати повільніше завантаження сторінок у порівнянні з SPA через необхідність перезавантаження всієї сторінки при переході між розділами.

Односторінковий сайт (SPA) є сучасною моделлю, що передбачає завантаження всього контенту на одну сторінку. Перемикання між розділами здійснюється динамічно за допомогою JavaScript, що забезпечує більш швидку та плавну взаємодію з користувачем. SPA підходить для сайтів з інтерактивними елементами, такими як каталоги товарів з фільтрами та сортуванням, віртуальні примірочні та інші інтерактивні функції.

Переваги SPA включають швидку взаємодію з користувачем та зменшення навантаження на сервер завдяки меншій кількості запитів. Однак оптимізація SEO для SPA може бути складнішою, оскільки традиційні пошукові системи мають проблеми з індексацією динамічного контенту.

Для побудови веб-сайтів показу одягу використовуються різноманітні методи, які включають різні підходи до дизайну, верстки та програмування.

Метод водоспадного проектування передбачає послідовне виконання етапів розробки, починаючи з аналізу вимог і закінчуючи тестуванням та запуском. Кожен етап має чітко визначені завдання і завершення одного етапу є передумовою для початку наступного. Цей метод зручний для проектів з чітко визначеними вимогами та фіксованим обсягом робіт. Переваги водоспадного методу включають чітку структуру та прогнозованість процесу розробки. Однак цей метод може бути менш гнучким у випадках, коли вимоги до проекту змінюються під час розробки.

Гнучка розробка (Agile) є підходом, який передбачає ітеративний процес розробки з регулярними зворотними зв'язками та можливістю адаптації до змін. Agile дозволяє швидко реагувати на змінені вимоги та покращувати продукт на кожному етапі розробки. Цей метод є популярним для створення веб-сайтів, де важлива швидкість впровадження та постійне покращення функціоналу. Переваги Agile включають високу гнучкість та можливість швидко адаптуватися до змін, що є особливо важливим для динамічного ринку моди. Недоліки включають можливість перевищення термінів та бюджету через постійні зміни вимог.

Різні алгоритми використовуються для забезпечення ефективної роботи веб-сайтів показу одягу. Серед них слід виділити алгоритми рендерингу, кешування та оптимізації.

Алгоритми рендерингу відповідають за перетворення HTML-коду в візуальне відображення на екрані користувача. Основним завданням є забезпечення швидкого та коректного відображення сторінок. Використання технологій, таких як Virtual DOM у фреймворках React або Vue.js, дозволяє значно підвищити швидкість рендерингу за рахунок оптимізації оновлень лише тих частин сторінки, які змінюються.

Алгоритми кешування використовуються для зберігання часто використовуваних даних та зменшення навантаження на сервер. Використання кешу дозволяє значно покращити продуктивність сайту, особливо при високих навантаженнях. Наприклад, кешування зображень, стилів та скриптів дозволяє зменшити час завантаження сторінок.

Алгоритми оптимізації включають різноманітні підходи до покращення швидкості та ефективності роботи сайту. Серед них можна виділити мінімізацію та об'єднання файлів, оптимізацію зображень, використання

CDN (Content Delivery Network) та інші методи. Оптимізація дозволяє зменшити час завантаження сторінок та покращити користувацький досвід.

Для побудови веб-сайтів показу одягу використовуються різні інструменти та фреймворки, які спрощують процес розробки та забезпечують високу якість кінцевого продукту.

Figma є популярним інструментом для дизайну інтерфейсів, який дозволяє створювати макети та прототипи веб-сайтів. Figma підтримує спільну роботу в режимі реального часу, що дозволяє дизайнерам та розробникам ефективно співпрацювати. Основні переваги Figma включають зручність використання, підтримку адаптивного дизайну та можливість інтеграції з іншими інструментами.

HTML, CSS та JavaScript є основними технологіями для верстки та програмування веб-сайтів. HTML використовується для структурування контенту, CSS для стилізації, а JavaScript для додавання інтерактивності. Використання цих технологій дозволяє створювати зручні та функціональні веб-сайти.

React та Vue.js є популярними JavaScript фреймворками, які дозволяють створювати динамічні та інтерактивні інтерфейси. Вони забезпечують високу продуктивність та гнучкість розробки, що є важливим для створення сучасних веб-сайтів показу одягу.

WordPress та Shopify є системами управління контентом (CMS), які забезпечують зручне адміністрування веб-сайтів. WordPress дозволяє створювати різноманітні веб-сайти, включаючи інтернет-магазини, блоги та корпоративні сайти. Shopify спеціалізується на створенні інтернет-магазинів та пропонує вбудовані інструменти для управління товарами, замовленнями та оплатами.

1.3 Характеристика сучасних сайтів одягу та їх програмне забезпечення

Сучасні веб-сайти одягу відрізняються значною різноманітністю дизайнів, функціональних можливостей та технологій. Проте, можна виділити деякі загальні тенденції, які характеризують цю сферу. Наведемо ці тенденції та їх характеристики.

1. Дизайн. Візуальна привабливість: Сучасні веб-сайти одягу прагнуть бути візуально привабливими та естетично приємними. Використовуються високоякісні зображення одягу, стильні шрифти та кольорові палітри, які відповідають бренду.

Простота та чіткість: Дизайн сайту має бути простим та лаконічним, щоб користувачі могли легко знайти те, що шукають. Навігація має бути чіткою та інтуїтивно зрозумілою, а структура сайту – логічною.

Адаптивність: Сайти одягу мають бути адаптивними, щоб правильно відображатися на різних пристроях, включаючи смартфони, планшети та комп'ютери. Це важливо, оскільки все більше людей використовують мобільні пристрої для онлайн-покупок.

2. Функціональні можливості. Інтернет-магазин: Більшість сучасних веб-сайтів одягу мають вбудований інтернет-магазин, який дозволяє користувачам переглядати товари, додавати їх у кошик та оформляти замовлення.

Фільтри та пошук: Користувачі повинні мати можливість фільтрувати товари за категорією, розміром, кольором, ціною та іншими характеристиками. Також на сайті має бути зручний пошук, який допоможе користувачам знайти те, що вони шукають.

Відгуки та рейтинги: Відгуки та рейтинги інших покупців можуть допомогти користувачам прийняти рішення про покупку.

Персоналізація: Деякі веб-сайти одягу пропонують користувачам персоналізовані рекомендації товарів на основі їх історії покупок та переглядів.

3. Програмне забезпечення. CMS: Більшість веб-сайтів одягу створюються за допомогою системи управління контентом (CMS). CMS дозволяє легко оновлювати сайт без необхідності знати код.

Електронна комерція: Для роботи інтернет-магазину використовується програмне забезпечення для електронної комерції. Це програмне забезпечення дозволяє обробляти замовлення, приймати платежі та керувати запасами.

Аналітика: Веб-сайти одягу використовують аналітику, щоб відстежувати трафік, поведінку користувачів та конверсії. Ця інформація може бути використана для покращення сайту та збільшення продажів.

Крім того, сучасні веб-сайти одягу можуть використовувати такі функції:

Блог: Блог може використовуватися для публікації новин, статей та порад про моду.

Соціальні мережі: Веб-сайт може бути інтегрований з соціальними мережами, щоб користувачі могли легко ділитися товарами та відгуками.

Відео: Відео може використовуватися для демонстрації одягу в дії або для створення рекламних роликів.

Віртуальна примірочна: Деякі веб-сайти одягу пропонують віртуальні примірочні, які дозволяють користувачам бачити, як одяг виглядатиме на них, не приміряючи його.

Важливо зазначити, що не всі веб-сайти одягу мають всі ці функції. Функції, які включені на сайті, залежать від бюджету, цілей та цільової аудиторії сайту. Приклади сучасних веб-сайтів одягу:

1. ASOS: https://www.asos.com/
2. Zalando: https://zalando.com/
3. Net-a-Porter: https://www.net-a-porter.com/
4. Farfetch: https://www.farfetch.com/
5. SSENSE: https://www.ssense.com/en-us/men/sale

Ці веб-сайти є прикладами того, як можна використовувати різні функції та технології для створення успішного веб-сайту одягу. Вони надають користувачам приємний досвід та пропонують широкий асортимент товарів.

Одним з найбільш популярних інструментів для розробки дизайну веб-сайтів одягу є Figma. Ця платформа дозволяє створювати інтерактивні макети, що відображають усі елементи та функції майбутнього сайту, підтримує спільну роботу в режимі реального часу, що дозволяє дизайнерам, розробникам та іншим учасникам проєкту ефективно співпрацювати. Висока гнучкість та можливість швидкого внесення змін роблять Figma ідеальним інструментом для сучасних веб-дизайнерів.

РОЗДІЛ 2.
ПРОЄКТУВАННЯ ВЕБ-САЙТУ БРЕНДОВОГО ОДЯГУ

2.1. Архітектура багатосторінкового сайту та характеристика його компонентів

Для розробки ефективного веб-сайту бренду одягу DEVIALT необхідно визначити основні компоненти сайту та розробити їхню архітектуру. Це дозволить створити зручний і функціональний сайт, що забезпечить користувачам легкий доступ до інформації про товари та простий процес здійснення покупок. Правильне проєктування архітектури сайту дозволяє забезпечити високий рівень зручності для користувачів, оптимізувати навантаження на сервер і покращити загальну продуктивність. Сайт складається з наступних компонентів:

- основна сторінка. Вона одночасно виконує функцію каталогу товарів. Відображає всі доступні товари з їхніми основними характеристиками, такими як назва, ціна, короткий опис та зображення. Інтерфейс сторінки повинен бути інтуїтивно зрозумілим та забезпечувати можливість фільтрації та сортування товарів за різними критеріями (категорія, розмір, колір, ціна тощо). Розміщення товарів має бути організовано таким чином, щоб користувачі могли швидко знайти потрібний їм продукт. Зазвичай використовуються сіткові макети, які дозволяють представити товари у вигляді сітки з фотографіями та короткими описами, що значно спрощує навігацію;
- сторінка товару. Кожен товар має свою окрему сторінку, де представлена детальна інформація про нього. Це включає розширений опис, характеристики, кілька фотографій з різних ракурсів, а також можливість

вибору розміру перед додаванням до кошика. Важливо забезпечити зручний інтерфейс для вибору розміру та перегляду всіх деталей товару. На сторінці товару також можуть бути відгуки інших покупців, рекомендації схожих товарів та додаткові опції, такі як відеопрезентації чи 3D-моделі товару. Це допомагає користувачам прийняти рішення про покупку та підвищує рівень їхньої задоволеності;

- кошик. Кошик є ключовим компонентом для здійснення покупок. Він дозволяє користувачам переглядати обрані товари, змінювати їх кількість, видаляти товари та переходити до оформлення замовлення. Кошик повинен бути легко доступним з будь-якої сторінки сайту і відображати підсумкову вартість замовлення з урахуванням податків і знижок. Важливо, щоб інтерфейс кошика був інтуїтивно зрозумілим і зручним для користувачів. Візуальні елементи, такі як кнопки "Додати до кошика", "Оформити замовлення", мають бути добре видимими і легко доступними. Крім того, кошик повинен підтримувати функцію збереження вибраних товарів навіть після виходу з сайту, що дозволяє користувачам повернутися до своїх покупок пізніше;
- оформлення замовлення. Система оформлення замовлення включає форму для введення контактних даних, вибору способу оплати та доставки. Важливо забезпечити зручність і безпеку цього процесу, включаючи підтримку різних методів оплати (кредитні картки, електронні гаманці тощо). Після заповнення всіх полів користувач отримує підтвердження замовлення на електронну пошту. Процес оформлення замовлення має бути максимально простим і зрозумілим, щоб мінімізувати кількість покинутих кошиків. Важливо забезпечити користувачів можливістю відстежувати статус свого замовлення та отримувати оновлення щодо його обробки та доставки.

2.2. Алгоритми роботи сайту та розрахунку вартості замовлення на придбання одягу

Для забезпечення ефективної роботи сайту необхідно розробити алгоритми, що відповідають за обробку даних і взаємодію з користувачем. На рис. 2.1 наведена схема такого алгоритму.

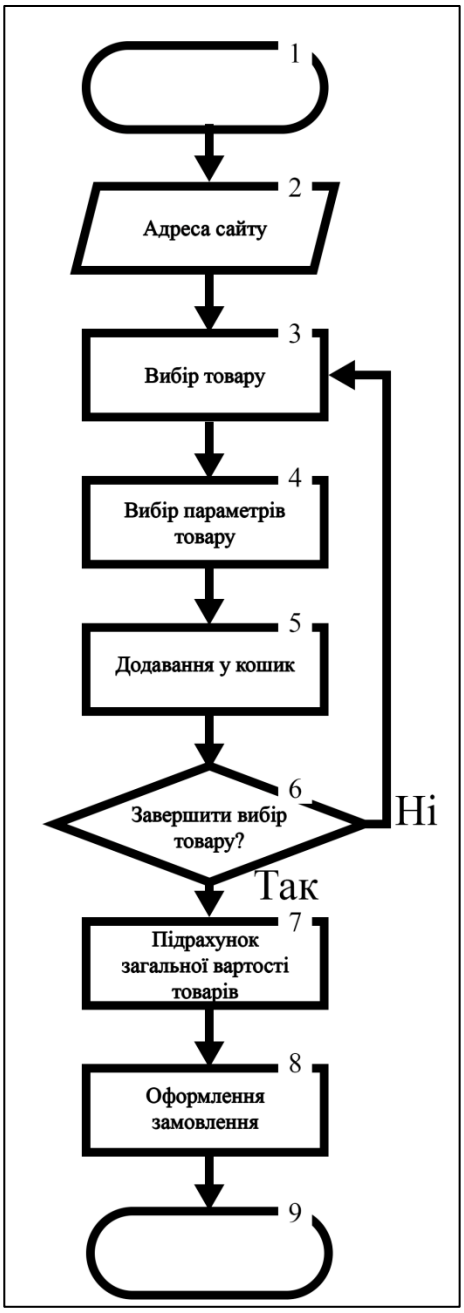


Рис. 2.1. Схема алгоритму взаємодії користувача з сайтом

Наведемо функціональні можливості, які реалізуються на сайтах продажу одягу (основні з яких задіяні на розробленому сайті):

- додавання товарів до кошика. Користувач обирає товар та розмір, додає його до кошика, де зберігається інформація про обрані товари. Важливо, щоб

цей процес був швидким і не вимагав перезавантаження сторінки, що покращує користувацький досвід;

- розрахунок загальної вартості замовлення. Система підсумовує вартість усіх товарів, додає податки та вартість доставки (якщо є), а також враховує застосовані знижки або промокоди. Важливо, щоб цей процес відбувався автоматично і відображався в реальному часі, дозволяючи користувачам бачити підсумкову вартість свого замовлення;
- генерація замовлення та відправка його до системи обробки замовлень. Після підтвердження замовлення користувачем система генерує унікальний номер замовлення та відправляє його на обробку. Це забезпечує точність та своєчасність обробки замовлень;
- підсумовування вартості обраних товарів. Вартість кожного товару додається до підсумкової суми замовлення. Це дозволяє користувачам бачити загальну вартість їхнього замовлення в режимі реального часу;
- застосування знижок. Якщо користувач має знижку або промокод, система автоматично віднімає знижку від підсумкової вартості. Це стимулює користувачів робити покупки та підвищує їхню задоволеність;
- додавання вартості доставки. Залежно від обраного способу доставки, до підсумкової суми додається вартість доставки. Це дозволяє користувачам бачити остаточну вартість свого замовлення з урахуванням усіх витрат.
- виведення підсумкової суми до оплати. Система відображає підсумкову суму, яку користувач повинен оплатити. Це забезпечує прозорість процесу та підвищує довіру користувачів до сайту.

Для розробки цих алгоритмів використовуються сучасні підходи та технології, що забезпечують високу продуктивність та зручність використання. Наприклад, використання JavaScript для динамічного оновлення кошика без перезавантаження сторінки покращує користувацький

досвід. Крім того, важливо забезпечити захист персональних даних користувачів, використовуючи надійні методи шифрування та безпечні протоколи передачі даних.

2.3. Проєктування дизайну сайту одягу DEVIALT у середовищі Figma

Проєктування дизайну веб-сайту в середовищі Figma включає кілька етапів, кожен з яких має своє значення для створення якісного та функціонального веб-сайту. Figma є потужним інструментом для створення макетів та прототипів, що дозволяє дизайнерам ефективно працювати над проектом. Використання Figma значно спрощує процес дизайну завдяки можливості спільної роботи в режимі реального часу, що дозволяє швидко вносити зміни та отримувати зворотний зв'язок. Наведемо характеристику основних етапів створення прототипів:

- низькорівневі прототипи. Спочатку створюються прості прототипи для визначення основної структури сторінок. Вони включають розміщення основних елементів, таких як меню, каталоги, кошики тощо. На цьому етапі важливо забезпечити логічну та зручну навігацію. Низькорівневі прототипи допомагають швидко оцінити загальну структуру сайту та виявити можливі проблеми на ранніх етапах розробки (рис. 2.2 та рис. 2.3);
- високоякісні макети. Після затвердження прототипів створюються детальні макети сторінок з урахуванням корпоративного стилю бренду. Використовуються кольори, шрифти та інші візуальні елементи, що відображають ідентичність бренду DEVIALT. Високоякісні макети дозволяють створити більш точне уявлення про те, як буде виглядати готовий сайт, і допомагають уникнути непорозумінь між дизайнерами та замовниками (рис. 2.4 та рис. 2.5);

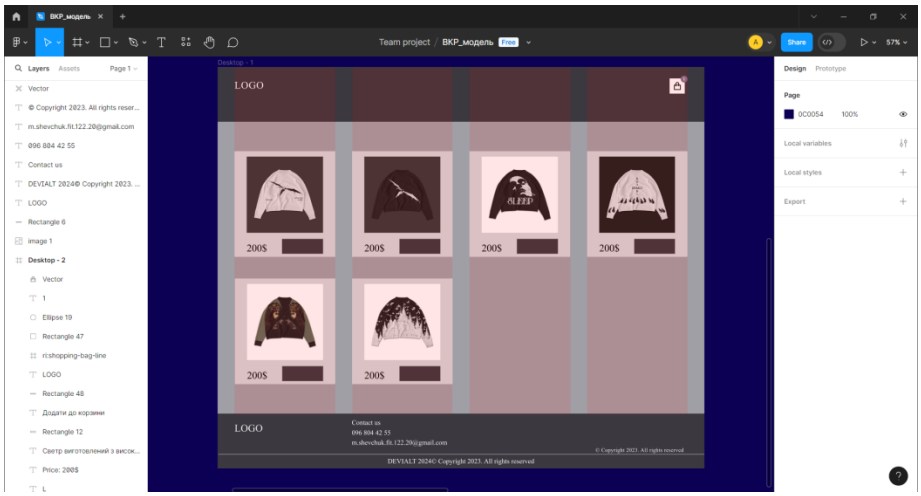


Рис. 2.2. Низькорівневий прототип основної сторінки

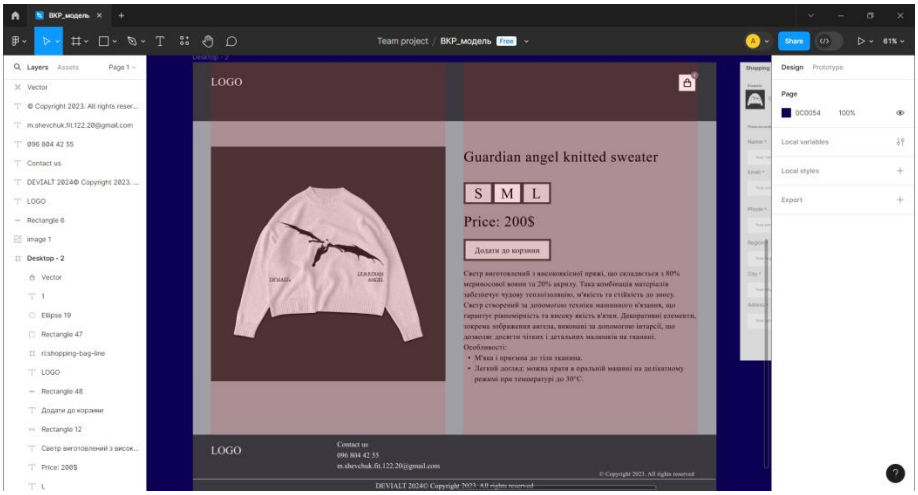


Рис. 2.3. Низькорівневий прототип сторінки товару

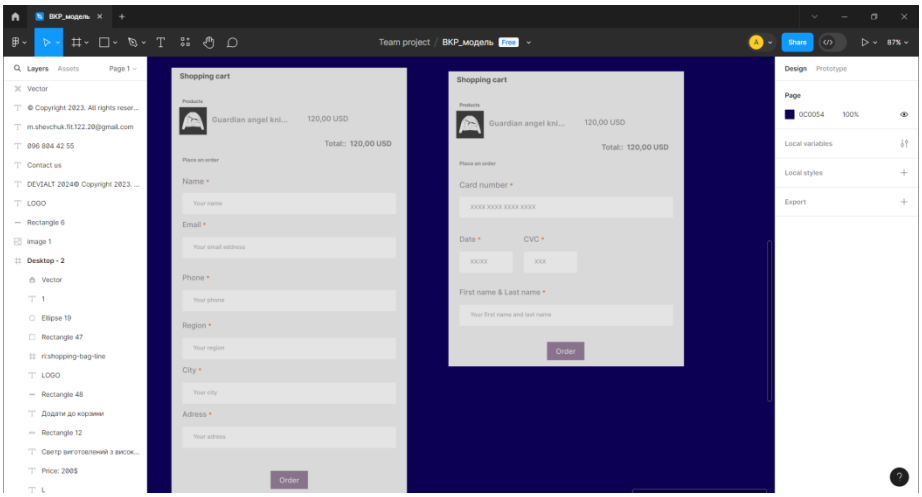


Рис. 2.4. Прототип корзини та оформлення замовлення

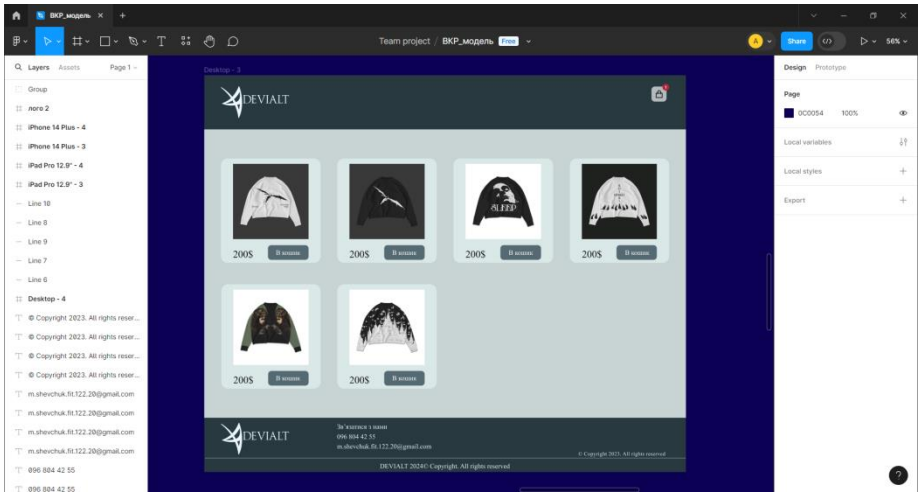


Рис. 2.5. Прототип основної сторінки

- додавання інтерактивності. У Figma можна створювати інтерактивні елементи, такі як кнопки, форми та переходи між сторінками. Це дозволяє моделювати поведінку користувача на сайті та тестувати різні сценарії взаємодії. Інтерактивні елементи допомагають користувачам зрозуміти, як працюватиме сайт, ще до його запуску, що дозволяє вчасно вносити корективи;

- тестування макетів. Важливо проводити тестування макетів з урахуванням зворотного зв'язку від потенційних користувачів. Це допомагає виявити недоліки та покращити дизайн до його остаточного затвердження. Тестування може включати різні методи, такі як юзабіліті-тести, опитування та аналіз поведінки користувачів.

- оптимізація макетів для різних пристроїв. Figma дозволяє легко створювати адаптивні макети, що коректно відображаються на різних пристроях, включаючи десктопи, планшети та смартфони. Адаптивний дизайн є важливим аспектом, оскільки все більше користувачів здійснюють покупки та переглядають сайти саме з мобільних пристроїв. Забезпечення адаптивного дизайну дозволяє підвищити зручність використання сайту та задовольнити потреби всіх користувачів (рис. 2.6);

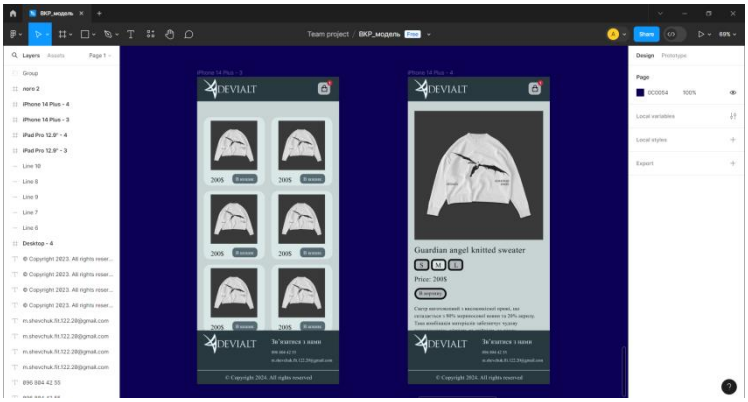


Рис. 2.6. Адаптивний дизайн сторінок для телефону

- забезпечення зручності використання. Веб-сайт повинен автоматично підлаштовуватися під розміри екрану користувача, забезпечуючи зручність використання на будь-якому пристрої. Це включає адаптацію шрифтів, зображень та розміщення елементів інтерфейсу. Адаптивний дизайн підвищує рівень задоволеності користувачів і сприяє збільшенню кількості повторних відвідувань сайту.

Таким чином, проєктування дизайну веб-сайту в середовищі Figma дозволяє створити якісний та функціональний продукт, який відповідає потребам користувачів та вимогам ринку. Використання сучасних технологій та методів забезпечує високу продуктивність і зручність використання сайту, що є ключовими факторами для досягнення успіху в сфері електронної комерції.

РОЗДІЛ 3.
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБ-САЙТУ БРЕНДУ ОДЯГУ DEVIALT

3.1. Створення та характеристика програмного коду окремих модулів веб-сайту одягу

У цьому розділі детально розглянемо розробку основних компонентів веб-сайту бренду одягу Devialt. Це включає клієнтську частину (HTML, CSS, JavaScript) і серверну частину (Node.js). Ми розглянемо ключові фрагменти коду, їх функції і взаємодію між собою. Наведемо характеристику HTML-файлів:

- вітальна сторінка "welcome.html". Цей HTML-файл (рис. 3.1) складається з кількох ключових частин.

1. Тег <head> містить метадані, такі як кодування, налаштування viewport і посилання на CSS-файл.

2. Тег <body> містить основний контент сторінки, включаючи шапку, головний контент і футер.

3. Шапка (<header>) містить логотип, який перенаправляє користувача на сторінку каталогу товарів.

4. Основний контент (<main>) містить вітальний текст і кнопку для переходу до каталогу.

5. Футер (<footer>) містить контактну інформацію і правові відомості.

Ця сторінка забезпечує користувачам можливість перейти до частини сайту з вибором товарів;

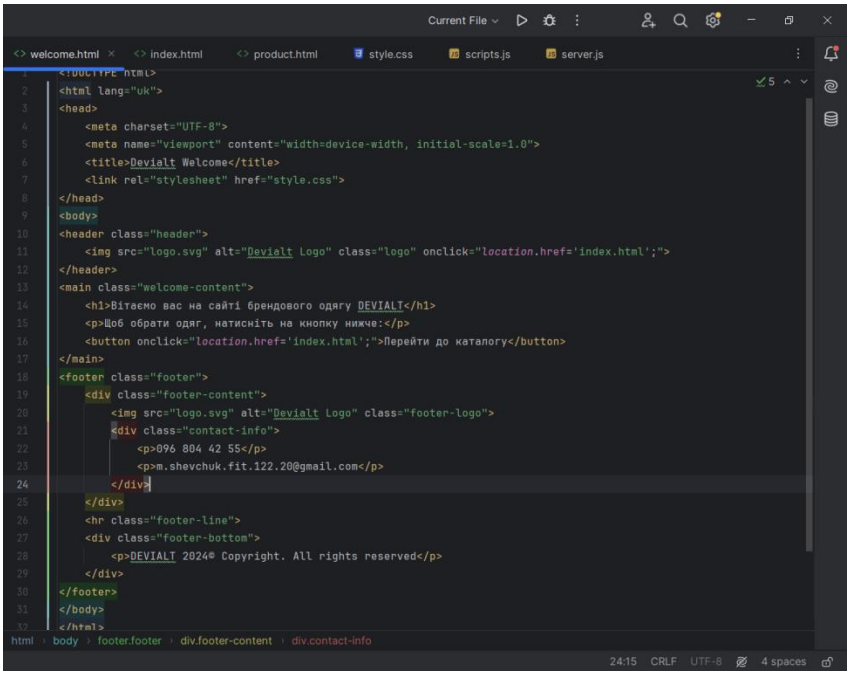


Рис. 3.1. Код відображає формування сторінки привітання (welcome.html)

- сторінка каталогу товарів "index.html". Цей HTML-файл (рис. 3.2-3.4) складається з кількох важливих розділів.

1. Тег <head> містить метадані та посилання на CSS-стилі.
2. Тег <body> включає шапку, основний контент і футер.
3. Шапка містить логотип та іконку кошика із кількістю товарів.
4. Основний контент містить сітку товарів, де кожен товар має кнопку "В кошик".
5. Спливаючі вікна для кошика, оформлення замовлення, платіжної інформації та підтвердження замовлення.

Ця сторінка забезпечує користувачам можливість перегляду детальної інформації про товар, вибір розміру та додавання товару до кошика;

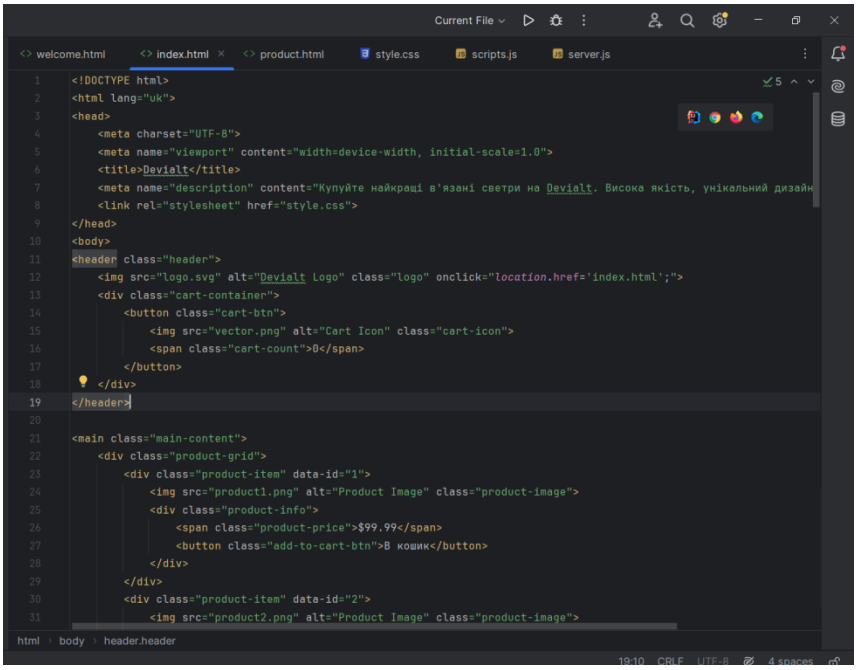


Рис. 3.2. Перша частина коду основної сторінки (index.html)

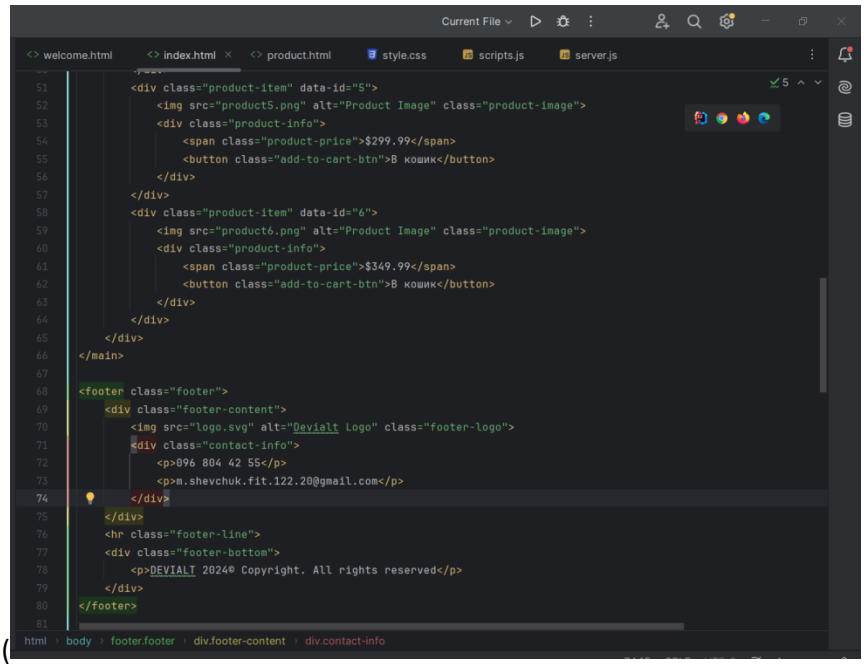


Рис. 3.3. Друга частина коду основної сторінки (index.html)

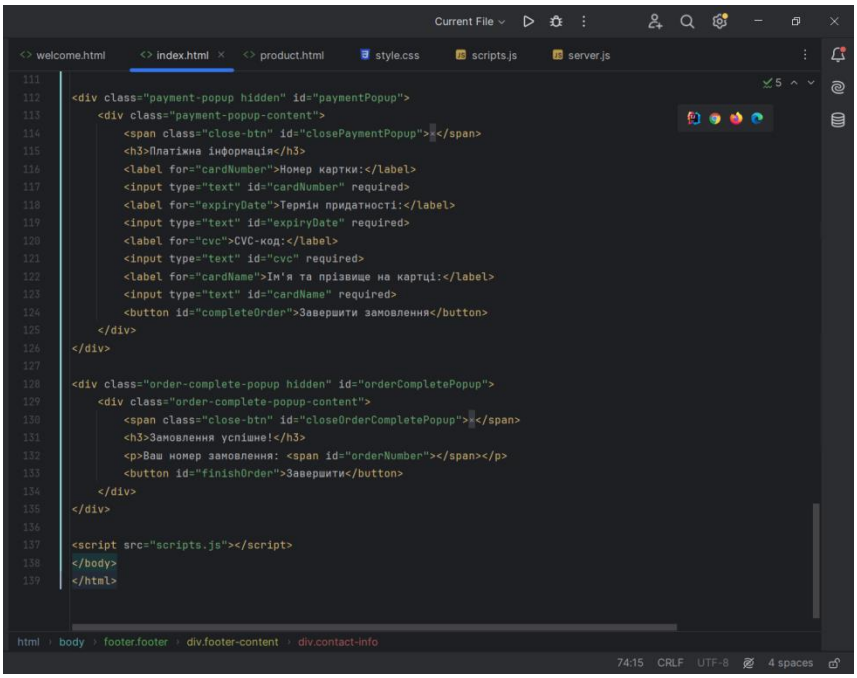


Рис. 3.4. Третя частина коду основної сторінки (index.html)

- розширена сторінка товару “product.html”. Цей HTML-файл (рис. 3.5) включає кілька основних компонентів.

1. Тег <head> містить метадані та посилання на CSS-стилі.

2. Тег <body> включає шапку, основний контент і футер.

3. Шапка містить логотип і кнопку кошика.

4. Основний контент (<main>) включає детальну інформацію про товар, вибір розміру та кнопку для додавання товару до кошика.

5. Футер містить контактну інформацію і правові відомості.

Ця сторінка забезпечує користувачам можливість перегляду детальної інформації про товар, вибір розміру та додавання товару до кошика.

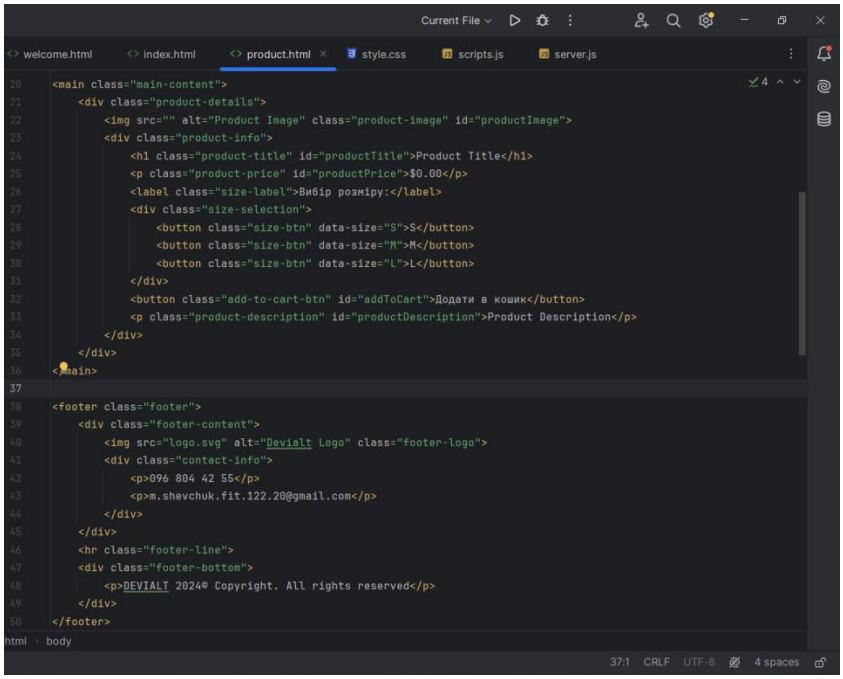


Рис. 3.5. Код для сторінки товару

Наведемо характеристику CSS-файлу. CSS використовується для стилізації HTML-елементів. Ось приклад ключових частин CSS-коду з файлу “style.css”:

- ці стилі визначають основні кольори та шрифти для веб-сайту. Вони також задають основні параметри для всіх елементів (*), такі як розмір коробки (box-sizing) (рис. 3.6);

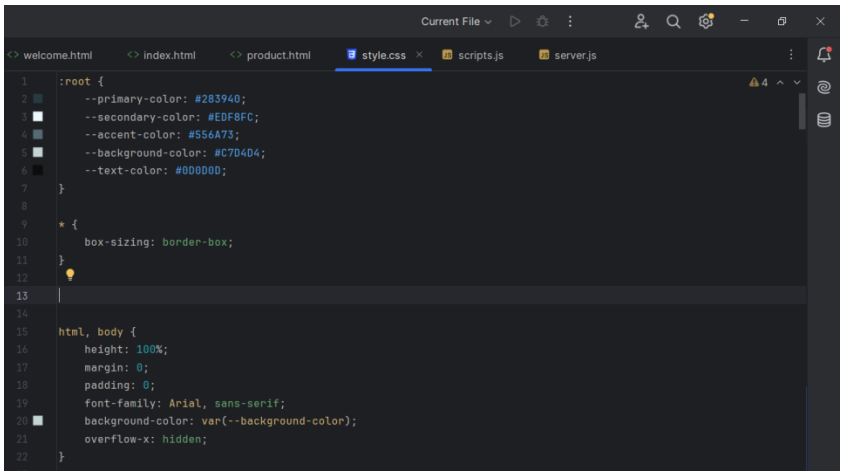


Рис. 3.6. Основні стилі сайту

- ці стилі визначають зовнішній вигляд шапки, включаючи фоновий колір, колір тексту, вирівнювання та розмір логотипу (рис. 3.7);

Рис. 3.7. Стилі для шапки сайту

- ці стилі визначають вигляд і поведінку кнопки кошика, включаючи позицію, розміри, кольори та кількість товарів у кошику (рис. 3.8);

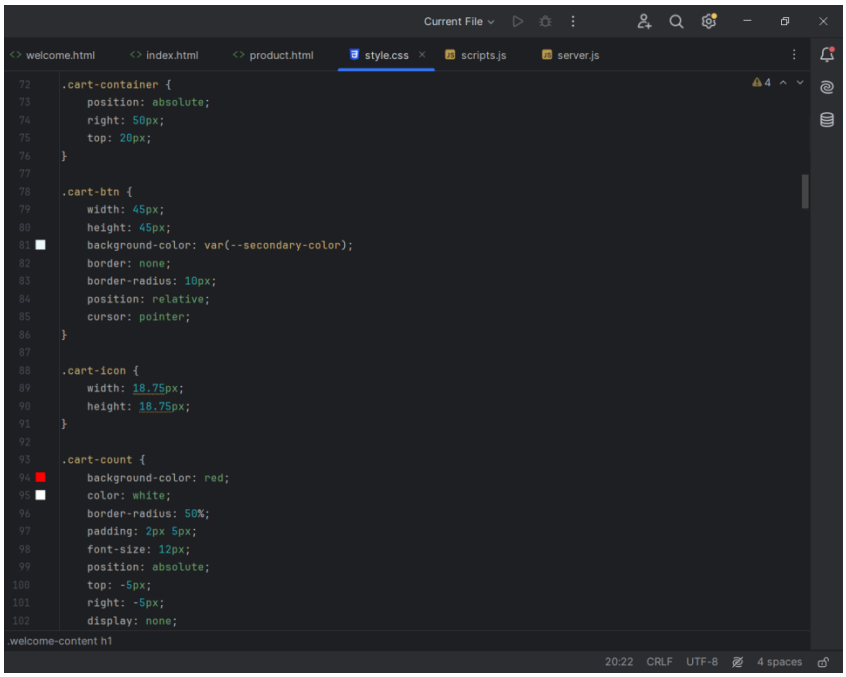


Рис. 3.8. Стилі для кошика

JavaScript забезпечує інтерактивність і динамічну поведінку веб-сайту. Наведемо характеристику JavaScript-файлів:

- Обробка подій для кошика "scripts.js". Цей файл (рис. 3.9-3.12) поділений на кілька важливих частин.

1. document.addEventListener('DOMContentLoaded', ...): забезпечує виконання коду після завантаження HTML-документа.

2. cartBtn.addEventListener('click', ...): обробляє подію натискання на кнопку кошика, показуючи вміст кошика.

3. renderCartItems(): відповідає за відображення товарів у кошику і додавання подій для видалення товарів.

4. updateCartCount() і updateTotalPrice(): оновлюють кількість товарів у кошику і загальну суму.

5. localStorage: використовується для збереження стану кошика між сеансами.

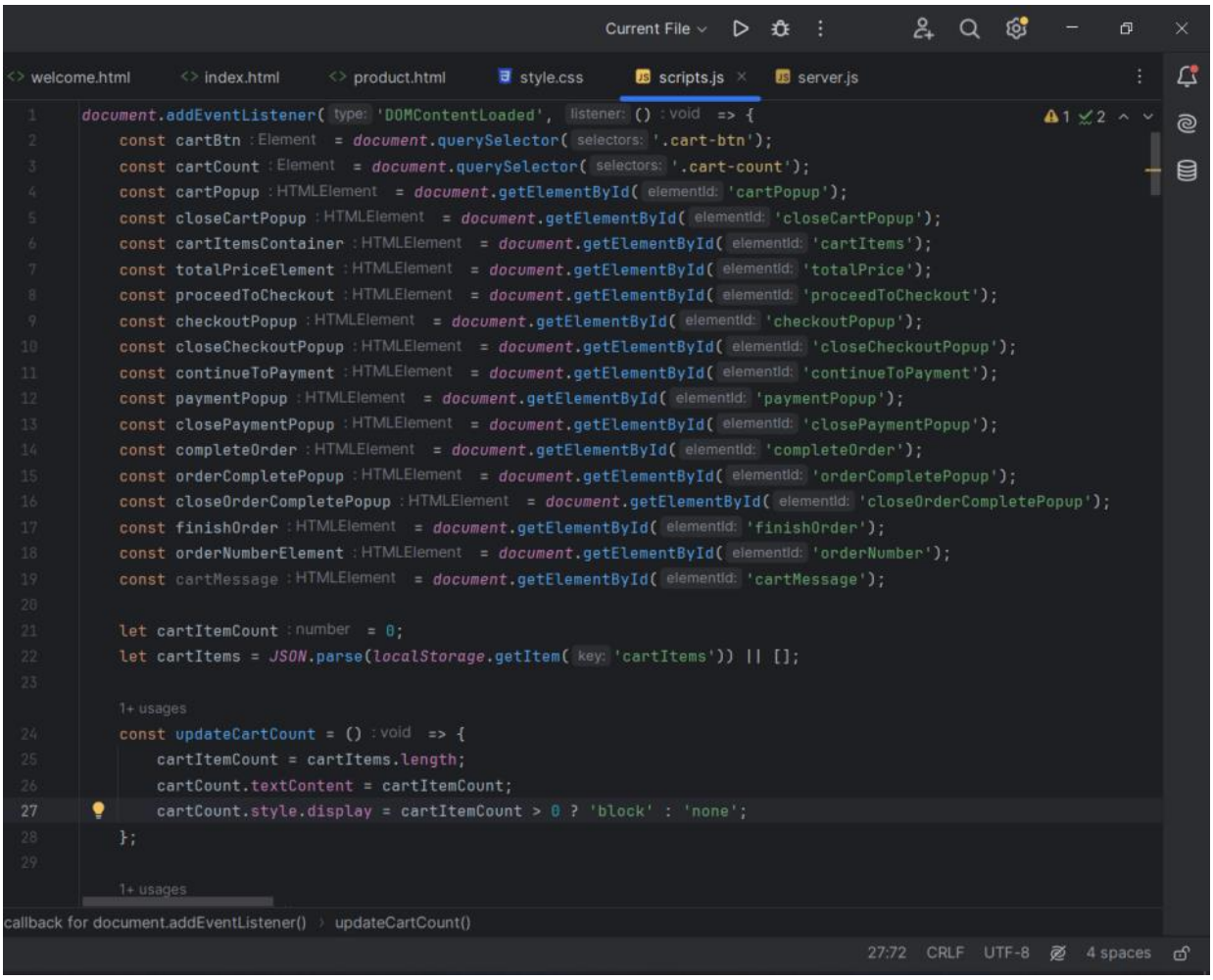


Рис. 3.9. Частина обробки подій для кошика

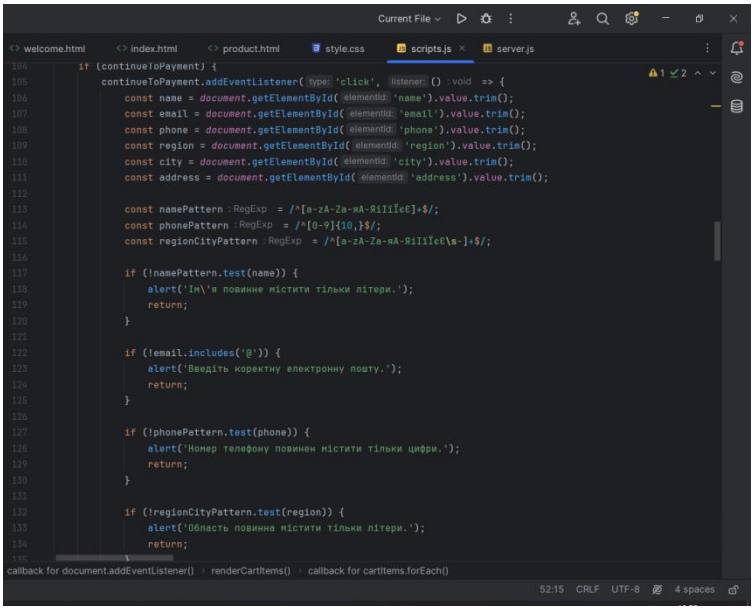
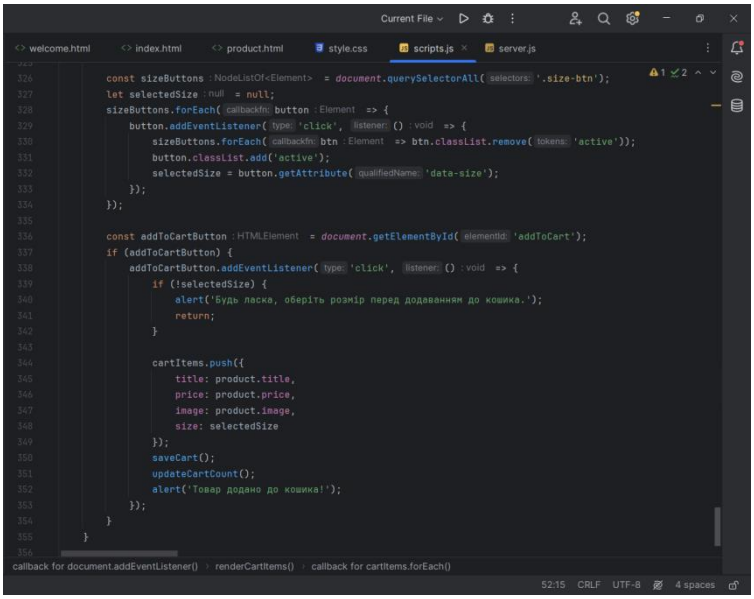


Рис. 3.10. Перевірка введених контактних даних

```
1+ usages
const generateOrderNumber = () : string => {
    const characters : string = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
    let orderNumber : string = '';
    for (let i : number = 0; i < 8; i++) {
        orderNumber += characters.charAt(Math.floor( x: Math.random() * characters.length));
    }
    return orderNumber;
};
```

Рис. 3.11. Формування унікального коду замовлення



```
Current File
welcome.html  index.html  product.html  style.css  scripts.js  server.js

const sizeButtons : NodeListOf<Element> = document.querySelectorAll( selectors: '.size-btn');
let selectedSize : null = null;
sizeButtons.forEach( callbackfn: button : Element => {
    button.addEventListener( type: 'click', listener: () : void => {
        sizeButtons.forEach( callbackfn: btn : Element => btn.classList.remove( tokens: 'active'));
        button.classList.add('active');
        selectedSize = button.getAttribute( qualifiedName: 'data-size');
    });
});

const addToCartButton : HTMLElement = document.getElementById( elementId: 'addToCart');
if (addToCartButton) {
    addToCartButton.addEventListener( type: 'click', listener: () : void => {
        if (!selectedSize) {
            alert('Будь ласка, оберіть розмір перед додаванням до кошика.');
            return;
        }

        cartItems.push({
            title: product.title,
            price: product.price,
            image: product.image,
            size: selectedSize
        });
        saveCart();
        updateCartCount();
        alert('Товар додано до кошика!');
    });
}
}

callback for document.addEventListener() › renderCartItems() › callback for cartItems.forEach()
52:15  CRLF  UTF-8  4 spaces
```

Рис. 3.12. Обробка та перевірка додавання товарів до кошику

- серверна частина (Node.js), файл "server.js". Для обробки серверних запитів і збереження замовлень використовується Node.js і Express.js. У цьому файлі (рис. 3.13) є кілька важливих частин.

1. express(): Ініціалізує додаток Express.
2. app.use(bodyParser.json()): Використовується для парсингу JSON-запитів.
3. app.use(express.static(publicDirectoryPath)): Налаштовує сервер для обслуговування статичних файлів.
4. app.get('/'): Визначає маршрут для головної сторінки.
5. app.post('/save-order'): Визначає маршрут для збереження замовлень, дані замовлення зберігаються у файл orders.txt.

6. app.listen(port, ...): Запускає сервер на вказаному порту.

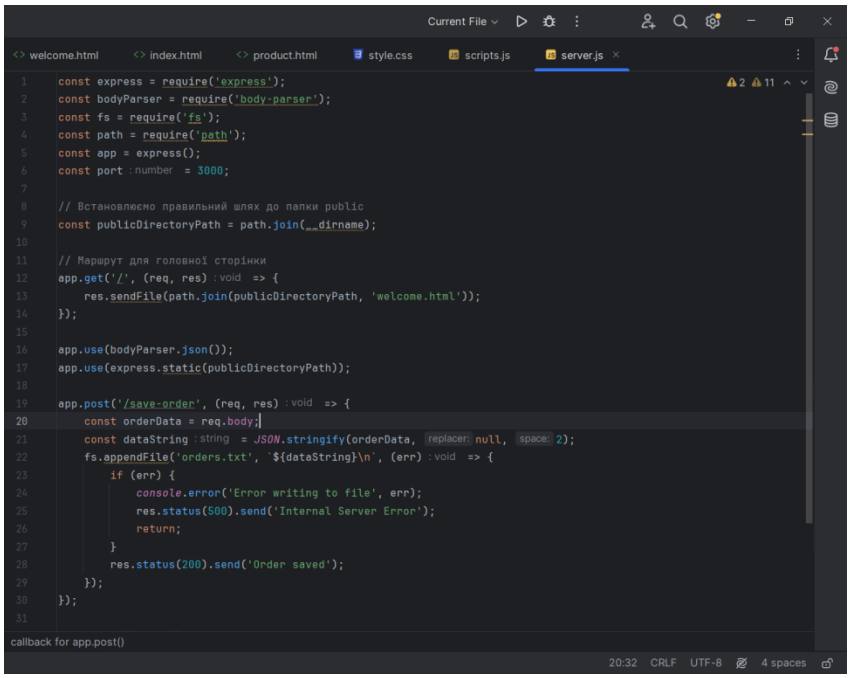


Рис. 3.13. Налаштування для серверної частини

Ця частина серверного коду забезпечує обробку запитів від клієнтів і збереження даних замовлень на сервері.

3.2. Характеристика віконного інтерфейсу взаємодії покупця з сайтом продажу

Інтерфейс користувача веб-сайту Devialt забезпечує інтуїтивно зрозумілу і зручну взаємодію для покупців. Основні етапи взаємодії включають вітальну сторінку, головну сторінку з каталогом товарів, сторінку детального перегляду товару, кошик та оформлення замовлення. Кожна сторінка має свій унікальний дизайн і функціональність, що забезпечує приємний користувацький досвід.

Вітальна сторінка є першою сторінкою, яку бачить користувач при вході на сайт. Вона має простий і привабливий дизайн, що включає логотип бренду, привітальне повідомлення та кнопку переходу до каталогу товарів (рис. 3.14).

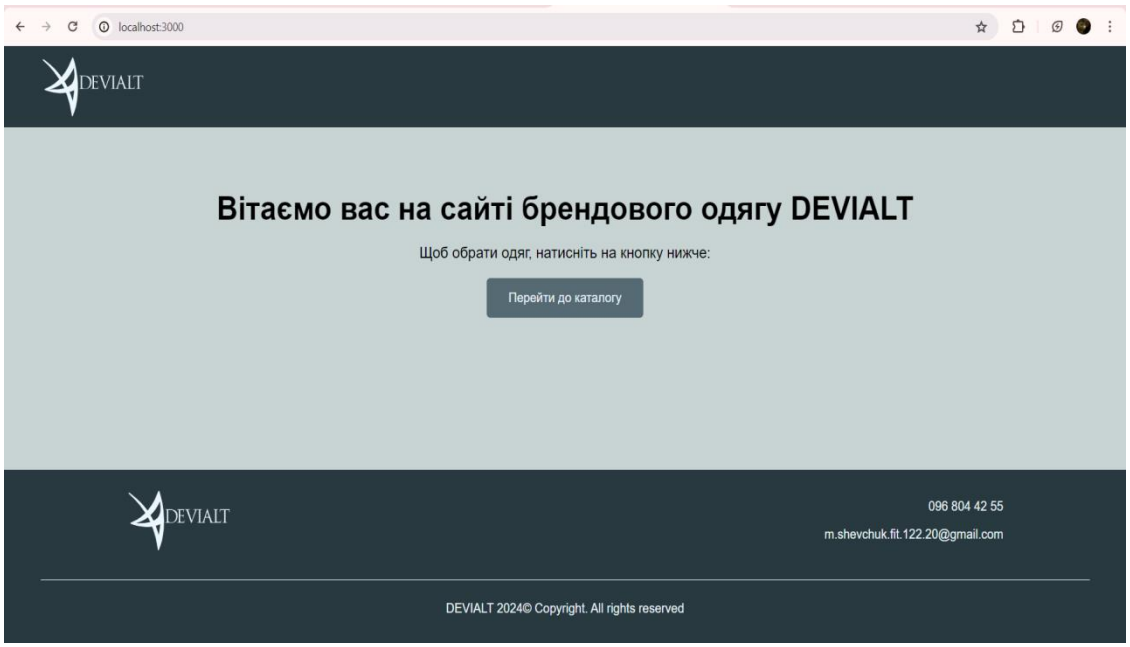


Рис. 3.14. Зовнішній вигляд вітальної сторінки

Основні елементи:

- логотип бренду, який також є посиланням на головну сторінку;
- привітальне повідомлення, що вітає користувача на сайті;
- кнопка "Перейти до каталогу", яка перенаправляє користувача до головної сторінки з каталогом товарів;
- контактна інформація у футері.

Головна сторінка з каталогом товарів містить каталог товарів, де користувачі можуть переглядати доступні продукти. Товари відображаються у вигляді сітки, кожен з яких має зображення, ціну і кнопку для додавання в кошик (рис. 3.15).

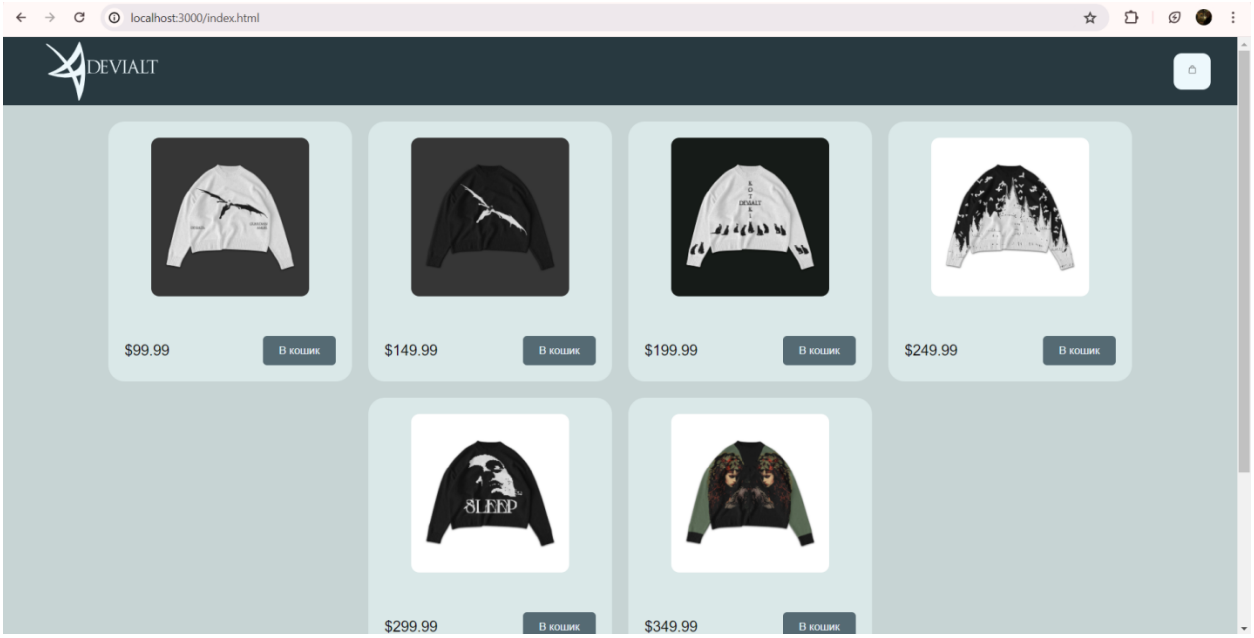


Рис. 3.15. Зовнішній вигляд сторінки з товарами

Основні елементи:

- логотип бренду, який є посиланням на головну сторінку;
- кнопка кошика з лічильником кількості товарів;
- сітка товарів, кожен товар має зображення, ціну та кнопку "В кошик";
- футер з контактною інформацією.

На сторінці розширеного перегляду товару користувач може детально ознайомитися з обраним товаром, включаючи його зображення, опис, ціну та вибрати розмір перед додаванням у кошик (рис. 3.16).

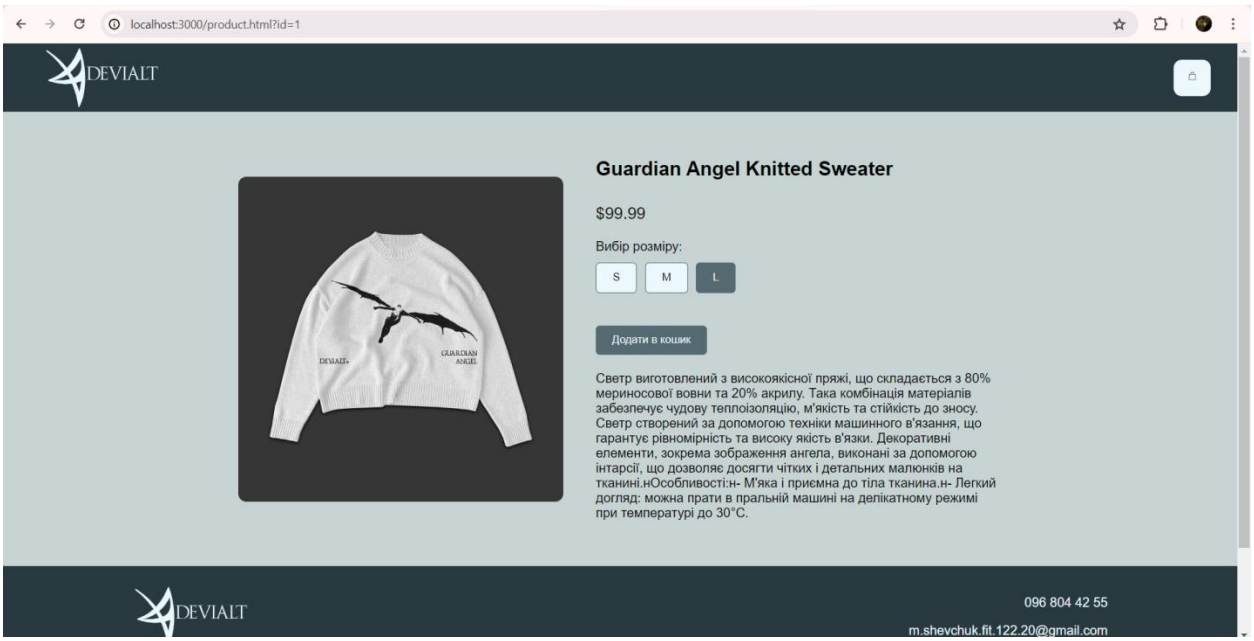


Рис. 3.16. Дизайн розширеної сторінки товару

Основні елементи:

- логотип бренду;
- кнопка кошика з лічильником кількості товарів;
- зображення товару;
- назва товару;
- ціна товару;
- опис товару;
- вибір розміру (кнопки для розмірів S, M, L);
- кнопка "Додати в кошик";
- футер з контактною інформацією;

Вікно кошика відображається, як спливаюче вікно, при натисканні на кнопку кошика на будь-якій сторінці. Користувач може переглянути товари, які він додав у кошик, змінити їх кількість або видалити товари (рис. 3.17).

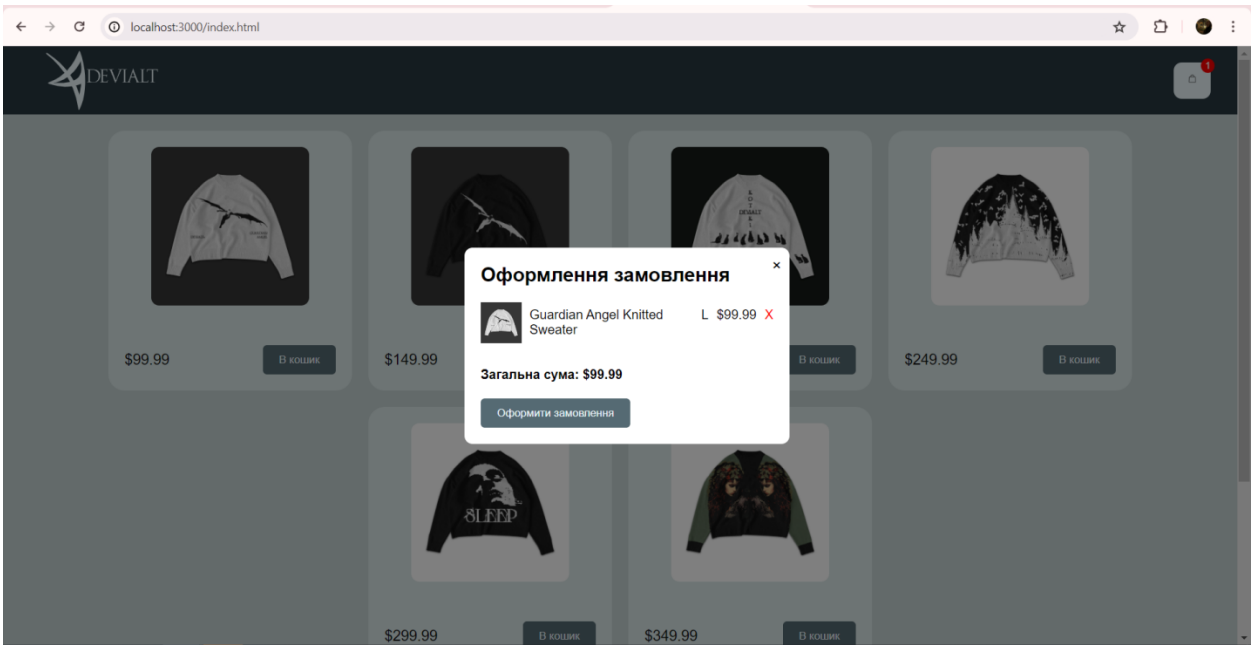


Рис. 3.17. Вікно кошика з товарами

Основні елементи:

- список товарів у кошику із зображенням, назвою, розміром, ціною та кнопкою для видалення товару;
- загальна сума замовлення;
- кнопка "Оформити замовлення" для переходу до сторінки оформлення замовлення;
- кнопка для закриття вікна кошика.

Оформлення замовлення. Після натискання кнопки "Оформити замовлення" користувач переходить до форми заповнення контактної інформації та платіжних даних.

Основні елементи:

- форма для введення контактної інформації (рис. 3.18) (ім'я, електронна пошта, номер телефону, область, місто, адреса);
- кнопка "Продовжити" для переходу до введення платіжної інформації;
- форма для введення платіжної інформації (рис. 3.18) (номер картки, термін придатності, CVC-код, ім'я на картці);

- кнопка "Завершити замовлення" для підтвердження покупки;
- кнопка для закриття вікна оформлення замовлення.

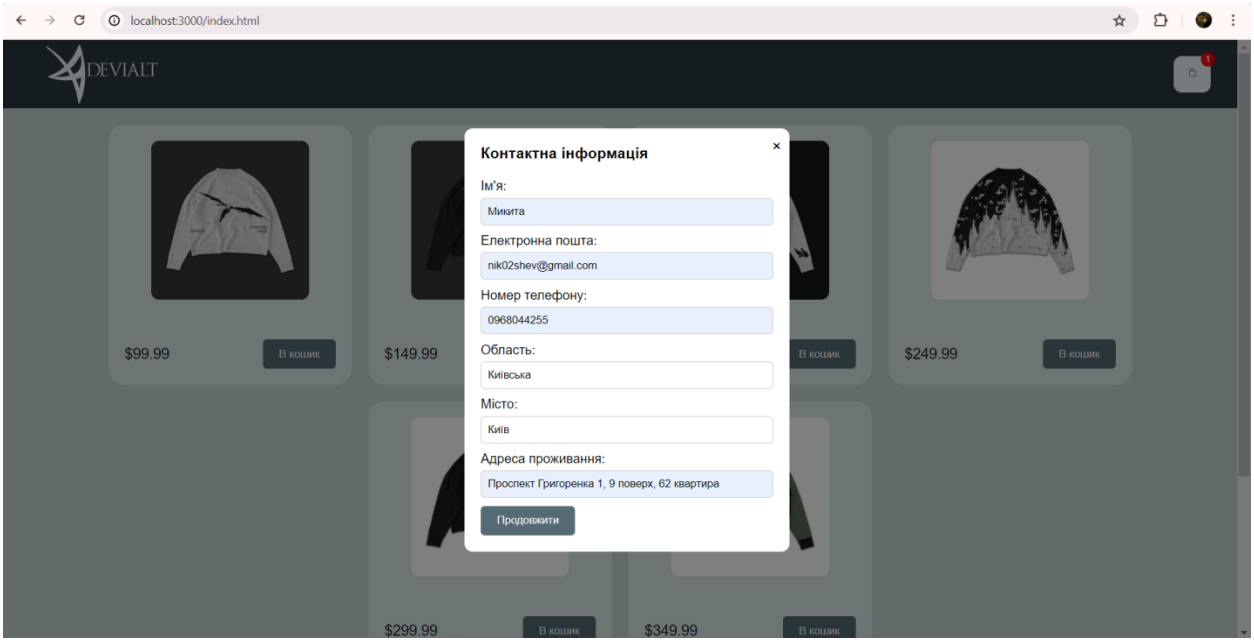


Рис. 3.18. Форма введення контактної інформації

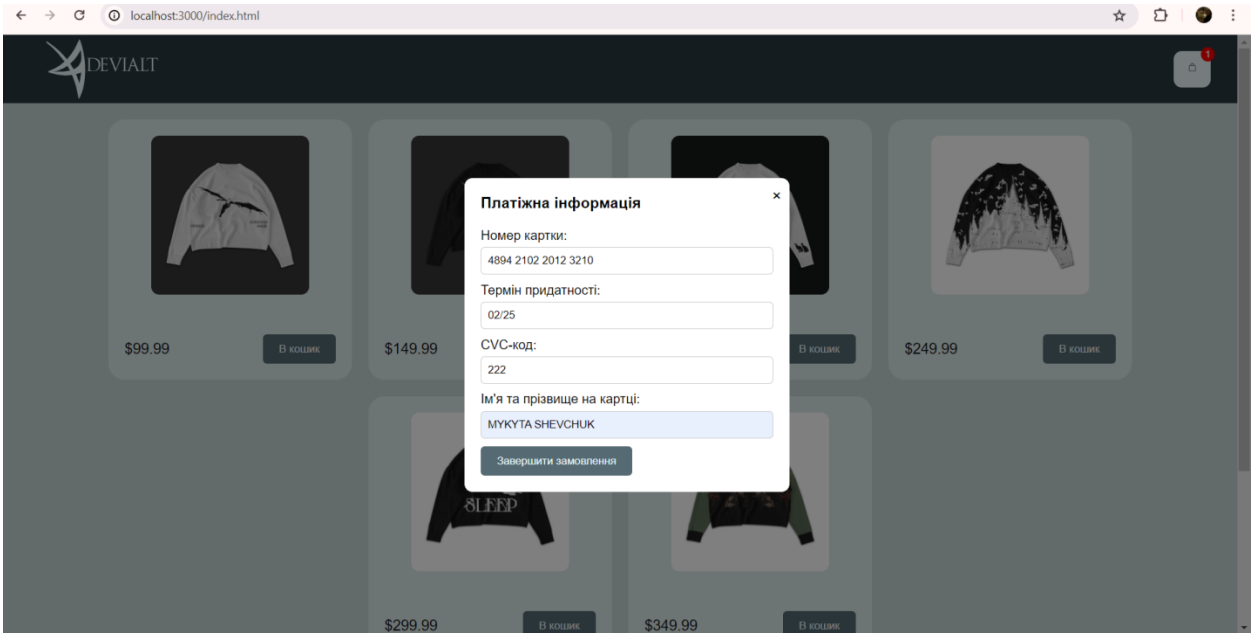


Рис. 3.19. Форма введення платіжної інформації

Підтвердження замовлення. Після успішного завершення замовлення користувач бачить повідомлення про підтвердження замовлення з номером замовлення (рис. 3.20).

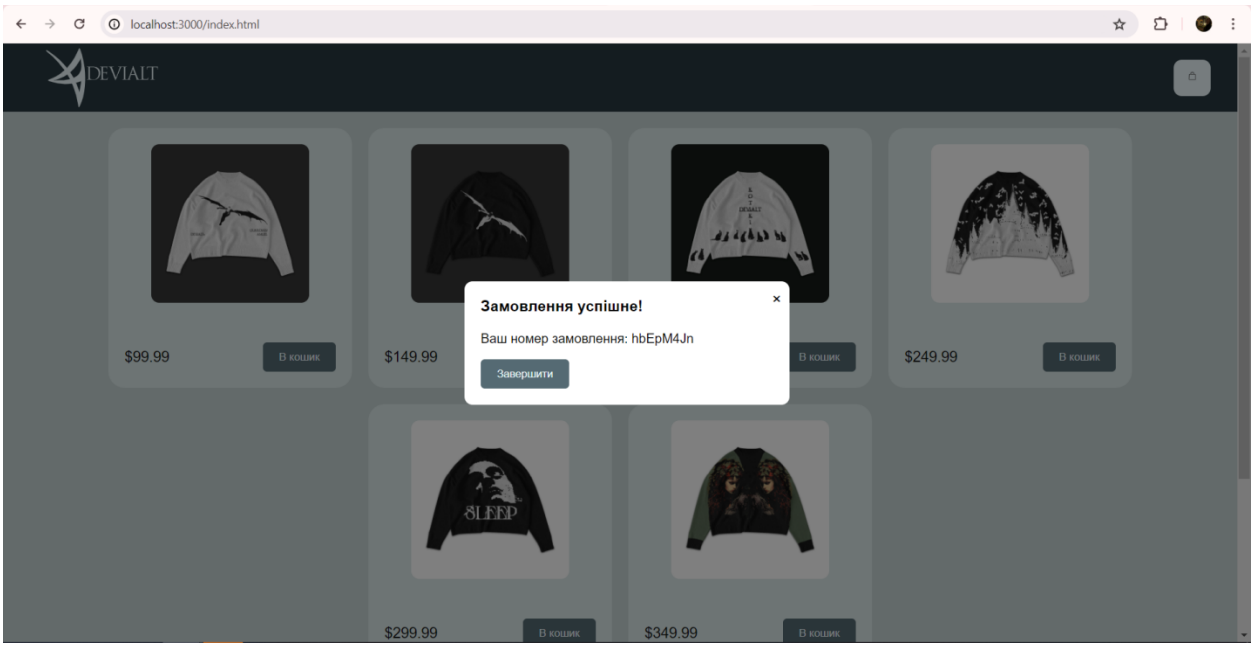


Рис. 3.20. Повідомлення про успішне замовлення з номером замовлення

Основні елементи:

- повідомлення про успішне замовлення;
- номер замовлення;
- кнопка "Завершити", яка повертає користувача на головну сторінку.

Кожен етап взаємодії користувача з веб-сайтом Devialt ретельно спланований і реалізований з урахуванням зручності та інтуїтивності. Візуальний дизайн і функціональність забезпечують позитивний користувацький досвід, що є критично важливим для залучення і утримання клієнтів.

3.3. Опис прикладу вхідної бази одягу, її використання та тестування для роботи сайту одягу DEVIALT

Вхідна база даних одягу є ключовим елементом для функціонування веб-сайту Devialt. Вона містить інформацію про всі доступні товари, включаючи їх назви, ціни, описи, зображення та доступні розміри. Ця база

даних забезпечує зберігання і доступ до інформації, необхідної для відображення товарів на сайті, їх додавання до кошика та обробки замовлень.

Структура вхідної бази даних: вхідна база даних одягу представлена у вигляді JavaScript-об'єкта у файлі scripts.js (рис. 3.21). Цей об'єкт містить усі необхідні дані про товари і використовується на сторінці детального перегляду товару (product.html) та на головній сторінці (index.html).

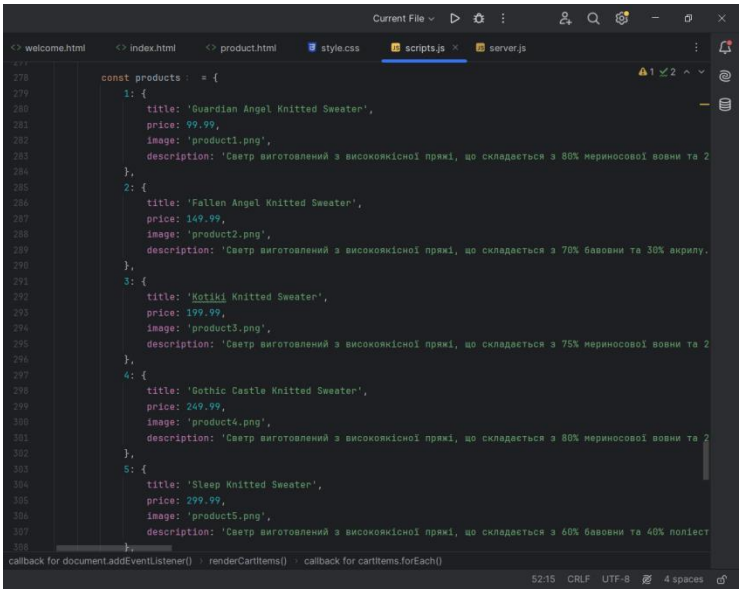


Рис. 3.21. Приклад структури бази даних товарів

Цей об'єкт містить шість товарів, кожен з яких має унікальний ідентифікатор. Для кожного товару вказані наступні дані:

- title (назва товару);
- price (ціна товару);
- Image (шлях до зображення товару);
- description (опис товару).

Використання бази даних. На головній сторінці використовується база даних для відображення каталогу товарів. Функція renderProducts() відповідає за рендеринг товарів з бази даних на сторінку.

Сторінка детального перегляду товару (product.html). на сторінці детального перегляду товару база даних використовується для відображення інформації про конкретний товар, обраний користувачем. На рис. 3.22 зображено код для виконання цих дій.

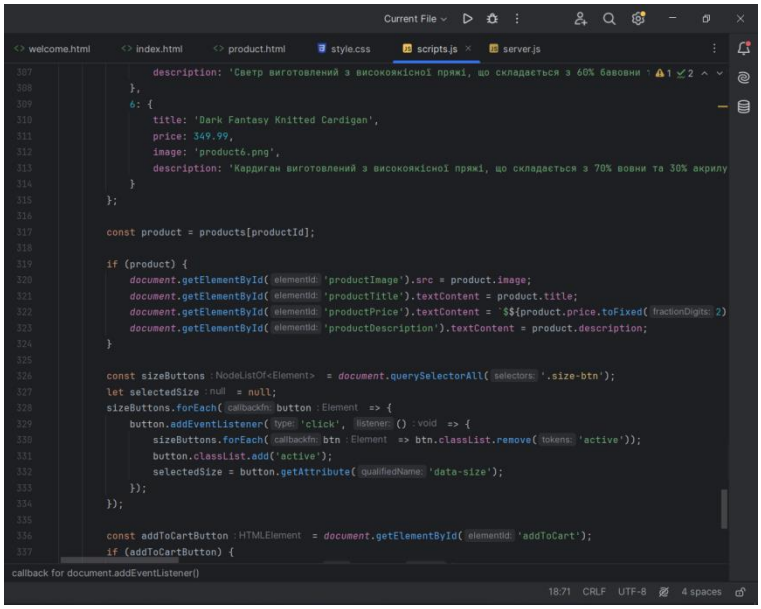


Рис. 3.22. Читання системою параметру «Ідентифікатор одягу» (id) з URL-адреси, пошук відповідного товару у базі даних і відображення на сторінці.

Тестування бази даних включає перевірку коректності відображення товарів на головній сторінці та на сторінці детального перегляду товару, а також перевірку функціональності додавання товарів до кошика та оформлення замовлень.

Спершу, перевірка відображення товарів на головній сторінці. Відкриваємо сторінку веб-сайту (index.html), щоб переконатися, що всі товари з бази даних відображаються у вигляді сітки. Важливо переконатися, що для кожного товару відображається правильне зображення та ціна. Для цього кожен товар повинен мати відповідні дані, які збігаються з тими, що зберігаються у базі даних (рис. 3.23).

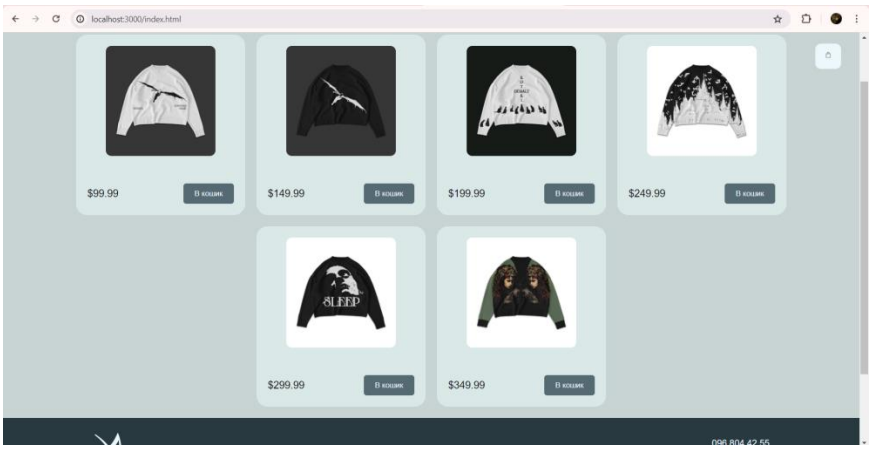


Рис. 3.23. Перевірка відображення товарів на головній сторінці

Далі перевірка відображення товару на сторінці детального перегляду. Використовуючи параметр id у URL-адресі (наприклад, http://localhost:3000/product.html?id=2). Треба переконатися, що сторінка завантажується з правильним товаром відповідно до переданого id. На рис. 3.24 показано, що кожен товар відображається без помилок.

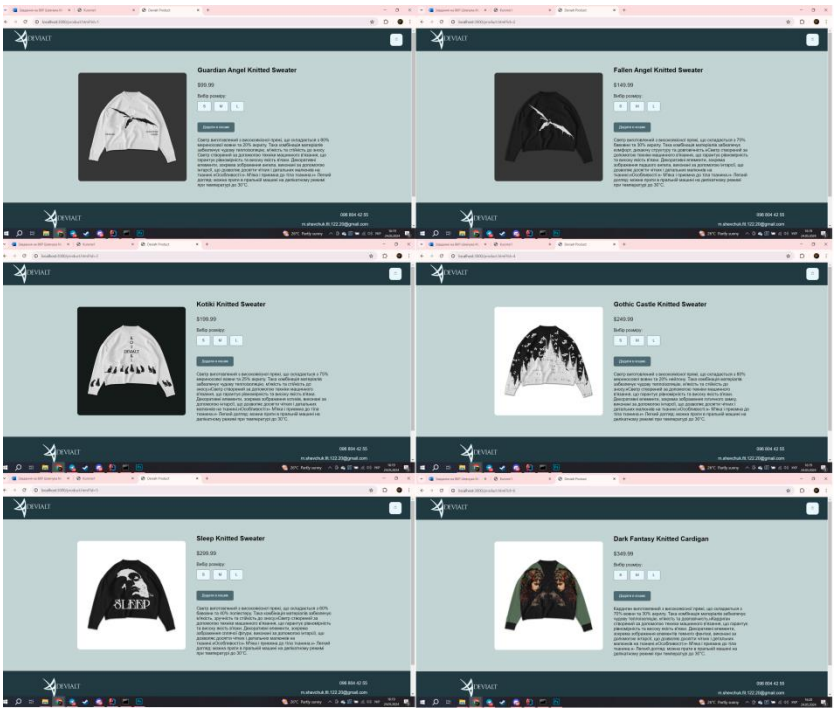


Рис. 3.24. Перевірка відображення кожного товару на сторінці детального перегляду

Перевірка, що всі дані товару (зображення, назва, ціна, опис) правильно відображаються на сторінці і відповідають інформації з бази даних. Далі оберемо розмір товару, натиснувши на відповідну кнопку розміру (S, M, L). Після цього натиснемо кнопку "Додати в кошик" і переконаємося, що товар правильно додається до кошика з вибраним розміром. Треба переконатись, що кожен товар має правильні дані (назва, зображення, розмір, ціна) (рис. 3.25).

Перевірка функціональність видалення товарів з кошика. Натисканням на кнопку видалення, можна переконатись, що товар успішно видаляється з кошика, а загальна сума оновлюється відповідно (рис. 3.26).

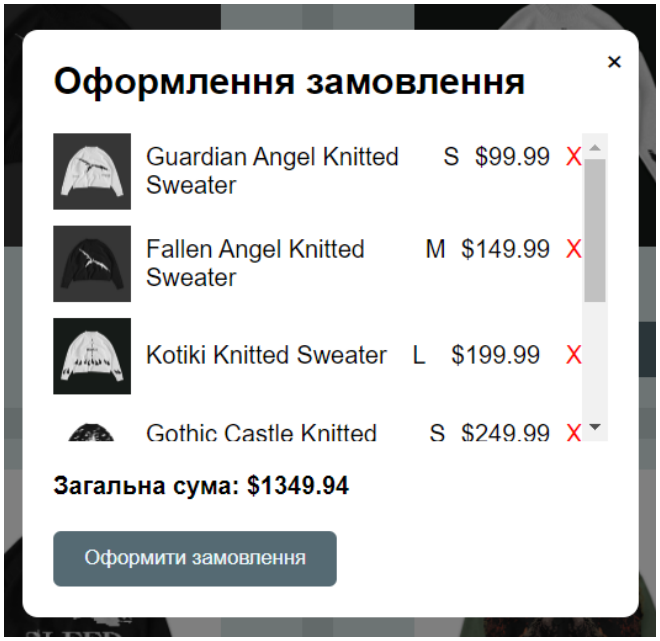


Рис. 3.25. Перевірка додавання товарів до кошика

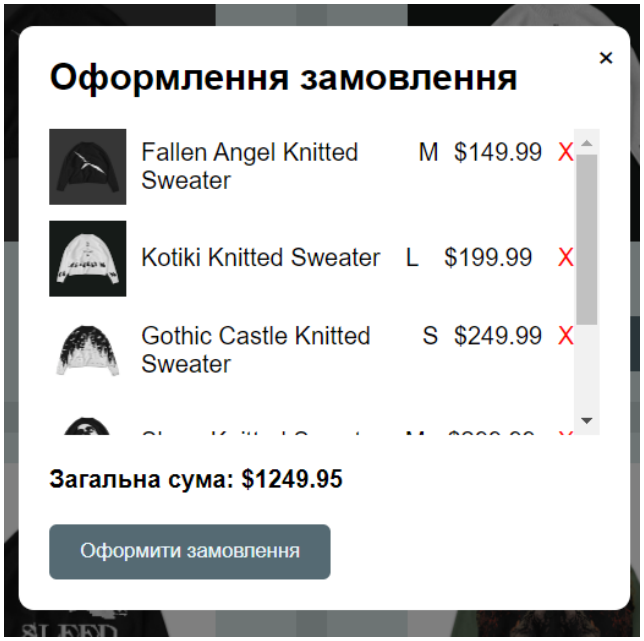


Рис. 3.26. Видалення товарів з кошика

Далі перехід до оформлення замовлення, натиснувши на кнопку "Оформити замовлення". Треба правильно заповнити всі поля контактної інформації (ім'я, електронна пошта, номер телефону, область, місто, адреса) і натиснути кнопку "Продовжити". Якщо якісь дані будуть введені неправильно, то повинне бути повідомлення про це (рис. 3.27).

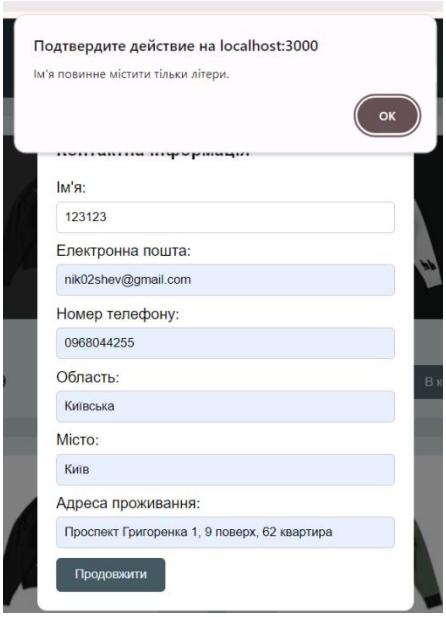


Рис. 3.27. Спроба неправильного вводу імені

На наступному етапі введення платіжних даних (номер картки, термін придатності, CVC-код, ім'я на картці). Треба переконатись, що всі поля заповнені коректно (рис. 3.28).

Після завершення замовлення треба перевірити, що з'являється повідомлення про успішне замовлення з номером замовлення (рис. 3.29).

Останнім етапом тестування є перевірка занесення замовлень у файл orders.txt. Цей файл містить усі дані про замовлення, збережені на сервері.
Після завершення замовлення, серверний код (server.js) зберігає дані замовлення у файл orders.txt (рис. 3.30).

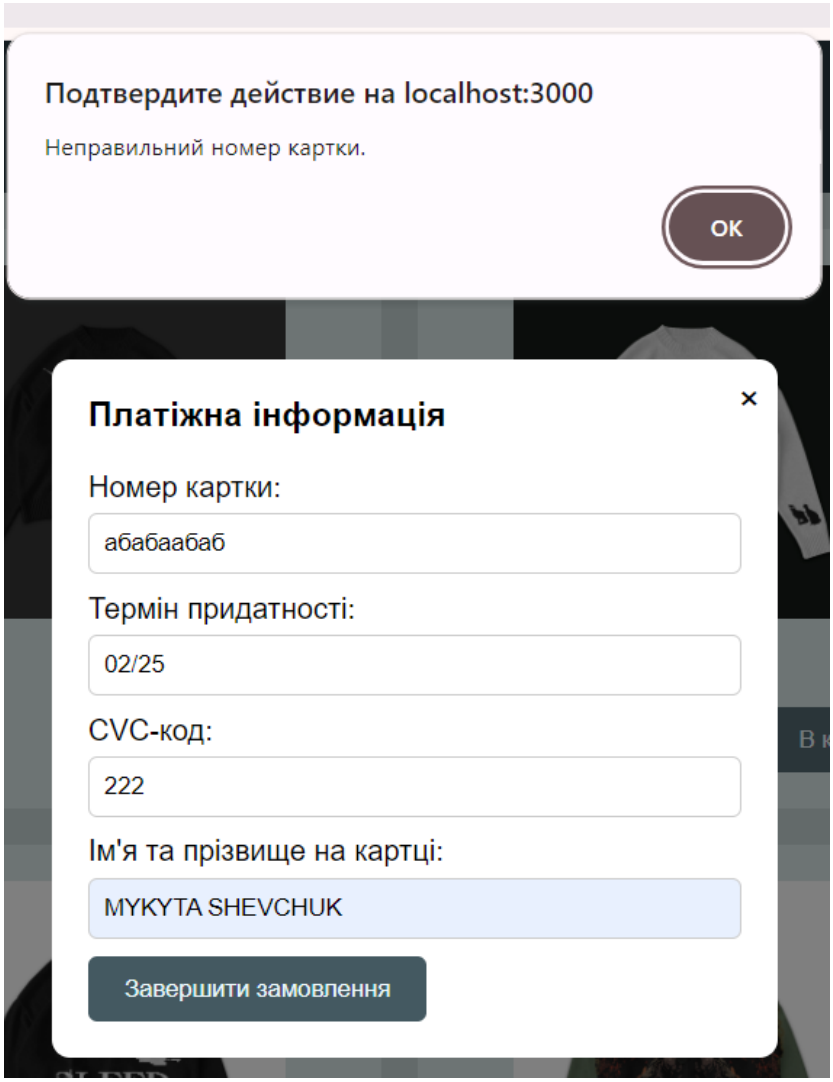


Рис. 3.28. Тестування неправильного вводу у номері картки

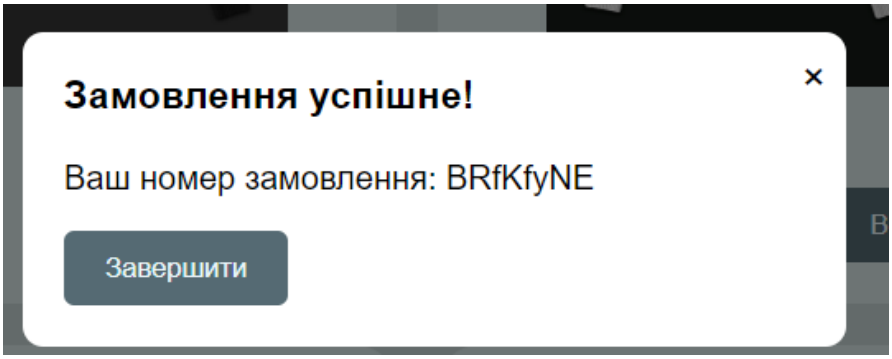


Рис. 3.29. Формування унікального номера замовлення

*orders.txt – Блокнот

Файл Правка Формат Вид Справка


```
{
  "orderNumber": "BRfKfyNE",
  "contactInfo": {
    "name": "Микита",
    "email": "nik02shev@gmail.com",
    "phone": "0968044255",
    "region": "Київська",
    "city": "Київ",
    "address": "Проспект Григоренка 1, 9 поверх, 62 квартира"
  },
  "paymentInfo": {
    "cardNumber": "3242234242344234",
    "expiryDate": "02/25",
    "cvc": "222",
    "cardName": "MYKYTA SHEVCHUK"
  },
  "cartItems": [
    {
      "title": "Fallen Angel Knitted Sweater",
      "price": 149.99,
      "image": "product2.png",
      "size": "M"
    },
    {
      "title": "Kotiki Knitted Sweater",
      "price": 199.99,
      "image": "product3.png",
      "size": "L"
    },
    {
      "title": "Gothic Castle Knitted Sweater",
      "price": 249.99,
      "image": "product4.png",
      "size": "S"
    },
    {
      "title": "Sleep Knitted Sweater",
      "price": 299.99,
      "image": "product5.png",
      "size": "M"
    },
    {
      "title": "Dark Fantasy Knitted Cardigan",
      "price": 349.99,
      "image": "product6.png",
      "size": "L"
    }
  ]
}
```


Рис. 3.30. Занесення всієї інформації про замовлення до orders.txt

Процес тестування бази даних включає перевірку всіх етапів взаємодії користувача з веб-сайтом, починаючи від відображення товарів на головній сторінці та сторінці детального перегляду, до додавання товарів у кошик, оформлення замовлень і перевірки занесення замовлень у файл orders.txt. Усі функції повинні працювати без помилок, забезпечуючи позитивний досвід для користувачів.

ВИСНОВКИ

У випускній кваліфікаційній роботі представлено результати теоретичних і прикладних досліджень, що полягають у розробці веб-сайту для бренду одягу DEVIALT з використанням інструменту Figma. Результати прикладних досліджень стали основою для створення функціонального та привабливого веб-сайту, що відповідає вимогам сучасного ринку. В результаті проведених досліджень були отримані такі **висновки:**

Використання Figma для розробки дизайну веб-сайту є ефективним засобом для створення візуально привабливого та функціонального інтерфейсу. Це забезпечує високу якість макетів і дозволяє швидко адаптуватися до змін вимог.

Проведений аналіз сучасних тенденцій у сфері веб-дизайну та їх впливу на користувацький досвід дозволив розробити концепцію дизайну, яка відповідає ідентичності бренду DEVIALT. Це включає вибір кольорової палітри, типографіки та композиційних рішень, що сприяють створенню впізнаваного та привабливого образу бренду.

Створено архітектуру багатосторінкового веб-сайту, яка забезпечує ефективну навігацію та зручний користувацький досвід. Основні компоненти

сайту включають головну сторінку, каталог товарів, сторінки детального перегляду товарів, кошик та систему оформлення замовлень.

Практична реалізація дизайну веб-сайту на платформі HTML/CSS/JavaScript показала високу ефективність використання цих технологій для створення сучасних, інтерактивних та адаптивних веб-сайтів. Усі розроблені модулі функціонують коректно та взаємодіють між собою, забезпечуючи безперебійне користування сайтом.

Ретельне тестування бази даних товарів підтвердило коректність відображення товарів на головній сторінці та сторінці детального перегляду, а також функціональність додавання товарів до кошика та оформлення замовлень. Усі замовлення успішно заносяться до файлу orders.txt, що свідчить про правильність реалізації серверної частини проекту.

Розроблений інтерфейс користувача забезпечує інтуїтивно зрозумілу навігацію та позитивний користувацький досвід. Всі елементи сайту були протестовані на зручність використання та відповідність сучасним стандартам веб-дизайну.

Отримані результати підтверджують ефективність застосованих методів та технологій для розробки дизайну та верстки веб-сайту. Впроваджені рішення можуть бути використані для створення інших веб-сайтів у сфері електронної комерції, що сприятиме покращенню якості взаємодії користувачів з брендом та підвищенню конкурентоспроможності компанії.

Рекомендації відносно удосконалення та розвитку розробленого веб-сайту.

1. Продовжити дослідження у напрямку вдосконалення користувацького досвіду, зокрема, шляхом впровадження нових інструментів та технологій.

2. Розглянути можливість використання штучного інтелекту для персоналізації пропозицій та покращення взаємодії з клієнтами.

3. Здійснювати регулярний аналіз поведінки користувачів на сайті для своєчасного виявлення проблем та впровадження відповідних покращень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ахвердова, І. Г. Теорія та практика веб-дизайну. Київ: Видавництво Ліра-К, 2018. С. 34-56, 120-145.

2. Буряк, І. В. Основи проектування веб-сайтів: навчальний посібник. Київ: Центр учбової літератури, 2019. С. 65-98, 172-196.

3. Глушак, О. С. Основи HTML та CSS: Веб-дизайн і розробка. Харків: Фоліо, 2020. С. 80-125, 200-235.

4. Гончаренко, Л. П. Основи JavaScript: від початківця до професіонала. Львів: Видавництво «Світ», 2017. С. 112-148, 250-285.

5. Гребенюк, Н. П. Веб-дизайн: сучасні тренди та перспективи. Київ: Наукова думка, 2021. С. 45-87, 150-180.

6. Зайцева, А. В., Ковальчук, М. В. Основи дизайну веб-сторінок. Дніпро: Ліра, 2016. С. 53-95, 145-175.

7. Карпенко, О. В. Технології створення веб-додатків. Київ: Видавництво КНУ, 2022. С. 130-160, 210-240.

8. Коваль, О. М. Верстка веб-сторінок за допомогою HTML та CSS. Київ: Політехніка, 2018. С. 95-120, 180-210.

9. Куценко, І. В. Веб-дизайн та розробка інтерфейсів. Київ: Видавництво НАУ, 2019. С. 110-135, 190-220.

10. Мороз, В. І. Основи проектування інформаційних систем. Одеса: Видавництво ОНУ, 2020. С. 75-105, 160-190.

11. Сидоренко, О. В. Основи UX/UI дизайну: від теорії до практики. Київ: Алерта, 2021. С. 90-120, 170-200.

12. Тарасов, Д. О. Практичний посібник з JavaScript. Львів: Видавництво «Львівська Політехніка», 2017. С. 105-145, 210-240.

13. Чалий, С. В. Розробка сучасних веб-додатків. Київ: Вид. дім «Києво-Могилянська академія», 2018. С. 80-110, 170-205.

14. Шевченко, М. В. Основи проектування та розробки веб-ресурсів. Київ: Видавництво КУБГ, 2021. С. 120-150, 200-230.

15. Бойко, Ю. О. Figma для початківців. Дніпро: Середняк, 2020. С. 50-80, 140-170.

16. Web Design Trends 2022. Retrieved from https://www.smashingmagazine.com/web-design-trends-2022

17. Figma: Collaborative Interface Design Tool. Retrieved from https://www.figma.com

18. JavaScript Documentation. Retrieved from https://developer.mozilla.org/en-US/docs/Web/JavaScript

19. HTML & CSS Documentation. Retrieved from https://developer.mozilla.org/en-US/docs/Web/HTML

20. Node.js Documentation. Retrieved from https://nodejs.org/en/docs/

21. React Documentation. Retrieved from https://reactjs.org/docs/getting-started.html

22. Bootstrap Documentation. Retrieved from https://getbootstrap.com/docs/5.0/getting-started/introduction/

23. UX/UI Design Principles. Retrieved from https://www.interaction-design.org/literature/topics/ux-design

24. Responsive Web Design Basics. Retrieved from https://developers.google.com/web/fundamentals/design-and-ux/responsive/

25. W3C Standards. Retrieved from https://www.w3.org/standards/

ДОДАТКИ

Додаток А. Формування основної сторінки

Додаток А.1. HTML-код головної сторінки (index.html)

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Devialt</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<header class="header">
    <img src="logo.svg" alt="Devialt Logo" class="logo"
onclick="location.href='index.html';">
```

```
    <div class="cart-container">
        <button class="cart-btn">
            <img src="vector.png" alt="Cart Icon" class="cart-icon">
            <span class="cart-count">0</span>
        </button>
    </div>
</header>
<main class="main-content">
    <div class="product-grid">
        <!-- Продукти будуть відображені тут -->
    </div>
</main>
<footer class="footer">
    <div class="footer-content">
        <img src="logo.svg" alt="Devialt Logo" class="footer-logo">
        <div class="contact-info">
            <p>096 804 42 55</p>
            <p>m.shevchuk.fit.122.20@gmail.com</p>
        </div>
    </div>
    <hr class="footer-line">
    <div class="footer-bottom">
        <p>DEVIALT 2024© Copyright. All rights reserved</p>
    </div>
</footer>
<script src="scripts.js"></script>
</body>
```

```
</html>
```

Додаток А.2. CSS-код головної сторінки (style.css)

```
:root {
    --primary-color: #283940;
    --secondary-color: #EDF8FC;
    --accent-color: #556A73;
    --background-color: #C7D4D4;
    --text-color: #0D0D0D;
}
 html, body {
    height: 100%;
    margin: 0;
    padding: 0;
    font-family: Arial, sans-serif;
    background-color: var(--background-color);
    overflow-x: hidden;
}
.header {
    background-color: var(--primary-color);
    color: var(--secondary-color);
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 0 50px;
    height: 100px;
}
```

```
.header .logo {
    width: 150px;
    cursor: pointer;
}
```

Додаток Б. Фрагмент основних видів одягу

Додаток Б.1. HTML-код для відображення товарів (index.html)

```
<main class="main-content">
    <div class="product-grid">
        <div class="product-item" data-id="1">
            <img src="product1.png" alt="Product Image" class="product-image">
            <div class="product-info">
                <span class="product-price">$99.99</span>
                <button class="add-to-cart-btn">В кошик</button>
            </div>
        </div>
        <!-- Інші продукти тут -->
    </div>
</main>
```

Додаток Б.2. JavaScript для відображення товарів (scripts.js)

```
const addToCartButtons = document.querySelectorAll('.add-to-cart-btn');
if (addToCartButtons.length) {
    addToCartButtons.forEach(button => {
        button.addEventListener('click', (event) => {
            const productItem = event.target.closest('.product-item');
            const productId = productItem.getAttribute('data-id');
```

```
            window.location.href = `product.html?id=${productId}`;
        });
    });
}
```

Додаток В. Фрагмент розрахунку загальної вартості

Додаток В.1. JavaScript для розрахунку загальної вартості кошика (scripts.js)

```
const updateTotalPrice = () => {
    const total = cartItems.reduce((sum, item) => sum + item.price, 0);
    totalPriceElement.textContent = total.toFixed(2);
};
```

Додаток В.2. HTML-код відображення загальної вартості кошика (index.html)

```
<div class="cart-popup-content">
    <p class="total-price">Загальна сума: $<span
id="totalPrice">0.00</span></p>
    <button id="proceedToCheckout">Оформити замовлення</button>
</div>
```

Додаток Г. Код серверної частини

Додаток Г.1. Серверний код для збереження замовлень (server.js)

```
const express = require('express');
const bodyParser = require('body-parser');
const fs = require('fs');
const path = require('path');
const app = express();
const port = 3000;
```

```
app.use(bodyParser.json());
app.use(express.static(path.join(__dirname)));
app.post('/save-order', (req, res) => {
    const orderData = req.body;
    const dataString = JSON.stringify(orderData, null, 2);
    fs.appendFile('orders.txt', `${dataString}\n`, (err) => {
        if (err) {
            console.error('Error writing to file', err);
            res.status(500).send('Internal Server Error');
            return;
        }
        res.status(200).send('Order saved');
    });
});
app.listen(port, () => {
    console.log(`Server running at http://localhost:${port}`);
});
```

Додаток Г.2. HTML-код для відправки замовлення (product.html)

```
<form id="orderForm">
    <label for="name">Ім'я:</label>
    <input type="text" id="name" required>
    <label for="email">Електронна пошта:</label>
    <input type="email" id="email" required>
    <!-- Інші поля форми -->
    <button type="submit">Завершити замовлення</button>
</form>
```

Додаток Г.3. JavaScript для обробки замовлення (scripts.js)

```
document.getElementById('orderForm').addEventListener('submit', (e) => {
    e.preventDefault();
    const orderData = {
        name: document.getElementById('name').value,
        email: document.getElementById('email').value,
        // Інші поля форми
    };
    fetch('/save-order', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify(orderData)
    })
    .then(response => response.text())
    .then(data => {
        console.log(data);
        alert('Замовлення успішно збережено');
    })
    .catch(error => {
        console.error('Error:', error);
        alert('Помилка збереження замовлення');
    });
}); document.getElementById('orderForm').addEventListener('submit', (e) => {
    e.preventDefault();
```

```
    const orderData = {
        name: document.getElementById('name').value,
        email: document.getElementById('email').value,
        // Інші поля форми
    };
    fetch('/save-order', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify(orderData)
    })
    .then(response => response.text())
    .then(data => {
        console.log(data);
        alert('Замовлення успішно збережено');
    })
    .catch(error => {
        console.error('Error:', error);
        alert('Помилка збереження замовлення');
    });
});
```