

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ПРОГРАМУВАННЯ

До захисту допускається.  
Завідувач кафедри \_\_\_\_\_ ІТП  
\_\_\_\_\_ Лифар В. \_\_\_\_\_  
«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

**ДИПЛОМНИЙ ПРОЄКТ (РОБОТА) БАКАЛАВРА  
ПОЯСНЮВАЛЬНА ЗАПИСКА**

НА ТЕМУ:

Веб-сайт інтернет магазину для малого бізнесу

---

---

---

Освітній ступінь – «бакалавр»

Спеціальність 121 «Інженерія програмного  
(шифр і назва спеціальності)

Керівник проєкту:

\_\_\_\_\_ (підпис)

Іванов В.Г.  
\_\_\_\_\_ (ініціали, прзвище)

Здобувач вищої освіти:

\_\_\_\_\_ (підпис)

Пупініна Є.Р.  
\_\_\_\_\_ (ініціали, прзвище)

Група:

ІПЗ-19д  
\_\_\_\_\_

Київ 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Інформаційних технологій та програмування

Освітній ступінь бакалавр

Спеціальність 121 Інженерія програмного забезпечення

(шифр і назва)

**ЗАТВЕРДЖУЮ:**

завідувач кафедри \_\_\_\_\_

ІТП \_\_\_\_\_

Лифар В.О.

« \_\_\_\_\_ » \_\_\_\_\_ 2023 р.

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

Пупініної Єлизавети Романівни

(прізвище, ім'я, по батькові)

1. Тема роботи: Веб-сайт інтернет магазину для малого бізнесу

Керівник проекту (роботи) \_\_\_\_\_

Іванов В. Г., к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "26" 04 2023 р. № 240/15.15-ОД

2. Термін подання роботи здобувача вищої освіти 19.06.2023

3. Вихідні дані до роботи матеріали переддипломної практики.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): а) аналіз предметної області;

б) існуючі реалізації

в) вибір програмних засобів для розробки продукту;

г) розробка продукту;

д) висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

## 6. Консультанти розділів проекту (роботи)

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|--------|---|----------------|------------------|
|        |   | завдання видав | завдання прийняв |
|        |   |                |                  |
|        |   |                |                  |
|        |   |                |                  |
|        |   |                |                  |

7. Дата видачі завдання 24.03.2023

Керівник

---

(підпис)

Завдання прийняв до виконання

---

(підпис)**КАЛЕНДАРНИЙ ПЛАН**

| № | Назва етапів дипломного проекту (роботи) | Строк виконання етапів проекту (роботи) | Примітка |
|---|--|---|----------|
| 1 | Отримання завдання                       | 24.03.2023                              |          |
| 2 | Пошук та вивчення літератури             | 27.03.2023-12.03.2023                   |          |
| 3 | Пошук та аналіз існуючих рішень          | 18.03.2023-21.03.2023                   |          |
| 4 | Розробка моделі веб-сайту                | 22.03.2023-15.04.2023                   |          |
| 5 | Реалізація веб-сайту                     | 16.04.2023-03.06.2023                   |          |
| 6 | Тестування фінальної версії продукту     | 05.06.2023-08.06.2023                   |          |
| 7 | Оформлення пояснювальної записки         | 08.06.2023-15.06.2023                   |          |

**Здобувач вищої освіти**

---

(підпис)**Керівник**

---

(підпис)**Пупініна Є.Р.**

---

(ініціали, прізвище)**Іванов В.Г.**

---

(ініціали, прізвище)

## РЕФЕРАТ

Пояснювальна записка складається з: 41 сторінок основного тексту, 14 рисунків, 1 додатку, 9 використаних джерел.

Метою даної бакалаврської дипломної роботи є веб-сайту для малого бізнесу з продажу одягу.

Об'єктом дослідження є розробка веб-сайту з використанням мов: HTML, CSS, Vue.js, SQL

### **Головні завдання роботи:**

1. Огляд предметної області.
2. Провести аналіз існуючих рішень.
3. Розробити архітектуру сайту.
4. Реалізувати архітектуру сайту
5. Провести перевірку створеного інструменту.

**Методи дослідження** – аналіз предметної області буде виконаний з акцентом на ієрархічне представлення об'єктів та сформованими логічними зв'язки між процесами.

**Ключові слова:** ПРОЄКТ, ВЕБ-САЙТ, ОДЯГ, VUE.JS, CSS, HTML, IDE, SQL

## ЗМІСТ

|   |    |
|---|----|
| ВСТУП.....  | 6  |
| РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ГАЛУЗІ .....                    | 8  |
| РОЗДІЛ 2 ІСНУЮЧІ РЕАЛІЗАЦІЇ.....                          | 10 |
| 2.1 Solmar .....  | 10 |
| 2.2 Cher17.....   | 11 |
| 2.3 19store.....  | 13 |
| 2.4 Parayashop.....                                       | 14 |
| РОЗДІЛ 3 АРХІТЕКТУРА ВЕБ-САЙТУ .....                      | 16 |
| 3.1 Модель сайту .....                                    | 17 |
| 3.1.1 Макет головної сторінки.....                        | 18 |
| 3.1.2 Макет каталогу .....                                | 19 |
| 3.1.3 Макет сторінки детальної інформації про товар ..... | 20 |
| 3.1.4 Макет сторінки оформлення замовлення.....           | 22 |
| 3.2 Архітектура сайту.....                                | 23 |
| 3.2.1 IDE IntelliJ IDEA.....                              | 23 |
| 3.2.2 HTML .....  | 24 |
| 3.2.3 CSS .....   | 25 |
| 3.2.4 Vue.js .....  | 26 |
| 3.2.5 SQL.....  | 27 |
| 3.3 Платіжна система LiqPay .....                         | 28 |
| 3.4 Просування веб-сайту .....                            | 31 |
| 3.5 Бізнес модель .....                                   | 34 |
| РОЗДІЛ 4 АПРОБАЦІЯ ВЕБ-ДОДАТКУ .....                      | 36 |
| 4.1 Головний екран.....                                   | 36 |
| 4.2 Сторінка каталогу веб-сайту.....                      | 38 |
| 4.3 Детальна інформація про товар .....                   | 39 |
| 4.4 Замовлення товару .....                               | 40 |
| ВИСНОВОК .....  | 42 |
| СПИСОК ЛІТЕРАТУРИ.....                                    | 44 |

## ВСТУП

У сучасному світі інтернет-технології стають необхідним інструментом для підтримки та розвитку бізнесу. Малий бізнес, зокрема, потребує ефективних та доступних рішень для просування своїх товарів і послуг на ринку. Веб-сайти інтернет-магазинів виявляються досить ефективним інструментом для залучення нових клієнтів та збільшення обсягів продажів.

Метою кваліфікаційної роботи є розробка та реалізація веб-сайту інтернет-магазину, спеціально адаптованого для потреб малого бізнесу. У роботі досліджуються різні аспекти веб-розробки, дизайну та функціоналу, це дозволить підвищити конкурентоспроможність малого бізнесу і полегшити процеси продажу через Інтернет.

У дипломній роботі будуть розглянуті основні етапи створення веб-сайту інтернет-магазину, включаючи аналіз вимог та потреб бізнесу, розробку дизайну, реалізацію функціоналу, інтеграцію платіжних систем, а також забезпечення безпеки та захисту персональних даних клієнтів. Крім того, будуть розглянуті інструменти та методи просування веб-сайту інтернет-магазину, щоб забезпечити максимальний потік відвідувачів та конвертацію їх у покупців.

Методологія дослідження включатиме збір і аналіз науково-технічної літератури, а також практичну реалізацію веб-сайту інтернет-магазину на базі сучасних веб-технологій. Результати дослідження й розробки будуть представлені у вигляді робочого веб-сайту, який буде оцінено з точки зору його ефективності, швидкодії та користуваельницької зручності.

Відповідно до отриманих результатів, робота над проектом інтернет-магазину може бути використана для подальшого розвитку та покращення малого бізнесу, забезпечуючи можливість розширення аудиторії клієнтів, збільшення обсягів продажів і підвищення конкурентоспроможності.

Основні завдання дипломної роботи включають:

1. Аналіз вимог та потреб малого бізнесу: Дослідження цільової аудиторії, аналіз конкурентів, визначення основних функцій і вимог до веб-сайту інтернет-магазину.

2. Розробка дизайну: Створення естетичного і привабливого дизайну, який відповідає бренду малого бізнесу, зручний для навігації та користування.

3. Реалізація функціоналу: Розробка інтерактивних елементів, таких як каталог товарів, кошик покупок, система фільтрації, оплата через платіжні системи.

4. Інтеграція платіжних систем: Забезпечення безпечного та зручного процесу оплати для клієнтів, включаючи інтеграцію з популярними платіжними шлюзами.

5. Продвиження та підтримка веб-сайту: Розробка стратегії просування, використання SEO-технік для покращення видимості сайту у пошукових системах, а також підтримка та оновлення веб-сайту після його запуску.

## РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ГАЛУЗІ

Веб-сайти з продажу одягу стали популярними платформами для електронної комерції, де підприємці можуть представляти свої модельні колекції, залучати клієнтів та збільшувати свої продажі. Огляд предметної галузі допоможе з'ясувати ключові аспекти та вимоги до розробки такого веб-сайту.

Веб-сайти з продажу одягу можна класифікувати на кілька категорій в залежності від їх специфіки та функцій. До них можна віднести:

1. Інформаційні сайти: Ці сайти надають основну інформацію про бренд, його історію, виробництво, філософію та особливості продукції.

2. Каталогіві сайти: Такі сайти містять широкий асортимент одягу, розділений на категорії та підкатегорії, з можливістю фільтрації за різними параметрами, такими як розмір, колір, стиль тощо.

3. Онлайн-магазини: Це повнофункціональні електронні платформи, де клієнти можуть шукати, обирати та купувати одяг. Онлайн-магазини зазвичай мають кошик покупок, систему оплати та доставки.

Вимоги до сучасних сайтів:

Сучасні веб-сайти для продажу одягу повинні відповідати ряду вимог, щоб забезпечити успішну і вигідну електронну комерцію:

1. Привабливий та зручний дизайн.

2. Зручний пошук та фільтрація: Веб-сайт повинен мати ефективну систему пошуку та фільтрації, щоб допомогти клієнтам швидко знаходити потрібний одяг.

3. Платіжні опції та безпека: Веб-сайт повинен підтримувати різні платіжні методи, такі як кредитні картки, електронні гаманці, а також забезпечувати безпеку транзакцій та захист персональних даних клієнтів.

Основний функціонал веб-сайту для продажу одягу повинен включати:

1. Перегляд каталогу товарів з можливістю фільтрації та сортування.

2. Детальну інформацію про кожен товар, включаючи фотографії, опис, розміри, кольори та ціни.



3. Додавання товарів до кошика та оформлення замовлення.

4. Інтеграція з платіжними шлюзами для прийому платежів.

Постановка задачі:

1. Аналіз вимог.

2. Проектування структури та інтерфейсу веб-сайту.

3. Розробка функціональності та інтеграція з базою даних.

4. Веб-дизайн та розробка стилів за допомогою CSS.

5. Тестування та налагодження веб-сайту.

## РОЗДІЛ 2 ІСНУЮЧІ РЕАЛІЗАЦІЇ

У цьому розділі буде проведено огляд існуючих веб-сайтів з продажу одягу. Аналіз існуючих рішень допоможе з'ясувати сильні та слабкі сторони різних веб-сайтів та визначити найкращі практики, які можна впровадити в розробці веб-сайту для продажу одягу.

Критерії аналізу веб-сайтів з продажу одягу:

1. Різноманітність: Вибірка повинна включати веб-сайти різних брендів, магазинів та платформ, щоб отримати широке уявлення про різні підходи та функціонал.

2. Популярність: Варто включити вибірку сайтів, які є популярними та мають велику кількість відвідувачів, оскільки це може свідчити про успішність та ефективність їхніх рішень.

3. Функціонал: Веб-сайти для аналізу повинні мати розширений функціонал, такий як пошук товарів, сортування, фільтрація, кошик покупок, система оплати та доставки.

4. Дизайн та зручність користування

Аналіз існуючих рішень дозволить отримати загальне уявлення сучасних веб-сайтів з продажу одягу та підкреслити ідеї для подальшої розробки веб-сайту для продажу одягу, а також допоможе визначити ті аспекти, які можна покращити або уникнути при розробці свого веб-сайту.

### 2.1 Solmar

Solmar - це популярний український бренд, який спеціалізується на виробництві базового трикотажного одягу. Сайт Solmar приваблює своїм красивим і лаконічним дизайном, де переважають біло-чорні кольори, доповнені акцентними зображеннями.

Дизайн сайту Solmar створений з акцентом на простоту та елегантність. Білі та чорні кольори створюють стильний контраст, що привертає увагу відвідувачів і підкреслює характер базового трикотажного одягу. Акцентні зображення додають виразності та привабливості до сайту, надаючи

користувачам візуальні точки прив'язки.

Однак, не лише дизайн, але й функціонал сайту Solmar вартий відзначення. Зручний та інтуїтивно зрозумілий інтерфейс допомагає користувачам легко зорієнтуватися на сайті та знайти потрібні товари. Функціональність сайту розроблена з урахуванням потреб клієнтів, дозволяючи зручно переглядати каталог продукції, додавати товари до кошика та здійснювати покупки з легкістю.

Solmar надає своїм клієнтам не тільки високоякісний базовий трикотажний одяг, але й зручну та естетичну онлайн-платформу для покупок. Красивий дизайн, акцентні зображення та інтуїтивно зрозумілий функціонал допомагають створити приємну та зручну атмосферу для клієнтів Solmar. Бренд поєднує естетику та функціональність, забезпечуючи задоволення та комфорт під час вибору та покупок базового трикотажного одягу.

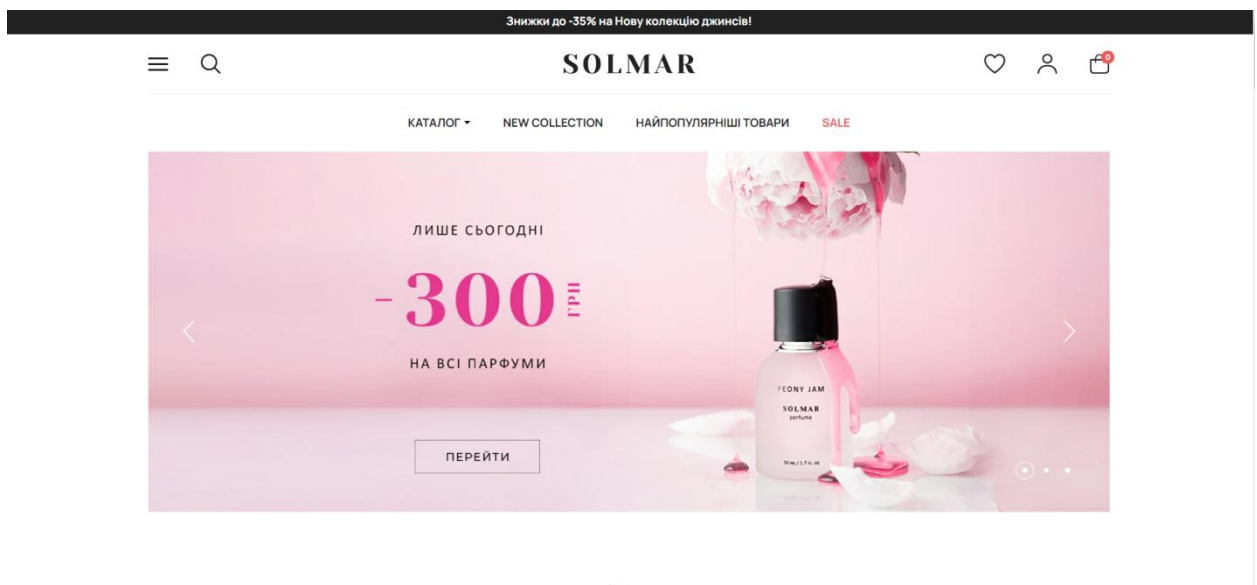


Рис. 2.1 – Головний екран Solmar

## 2.2 Cher17

Cher17 - український бренд одягу, який пропонує широкий вибір стильного та модного одягу для будь-яких випадків. Веб-сайт Cher17 виконаний у сучасному стилі з використанням анімованих динамічних вставок, що надає йому елегантності та виразності.

Головна сторінка сайту Cher17 ретельно структурована на різні сектори, що допомагає користувачам легко та швидко орієнтуватися на сайті. Кожен сектор відповідає конкретному аспекту бренду або типу одягу, що дозволяє користувачам без зусиль знайти інформацію або товар, який їх цікавить. Логічна структура головної сторінки сприяє ефективній навігації та забезпечує зручний досвід для користувачів.

Одним з основних елементів дизайну Cher17 є анімовані динамічні вставки, які додають жвавості та виразності на сайті. Ці анімації привертають увагу користувачів, створюють враження інтерактивності та додають стильного акценту до веб-сайту.

Cher17 надає своїм клієнтам не тільки широкий вибір модного одягу, але й елегантне та привабливе віртуальне середовище для покупок. Завдяки сучасному стилю дизайну, анімованим динамічним вставкам та логічній структурі головної сторінки, Cher17 створює приємний та зручний досвід покупок, дозволяючи користувачам знайти та придбати бажаний одяг з легкістю та задоволенням.

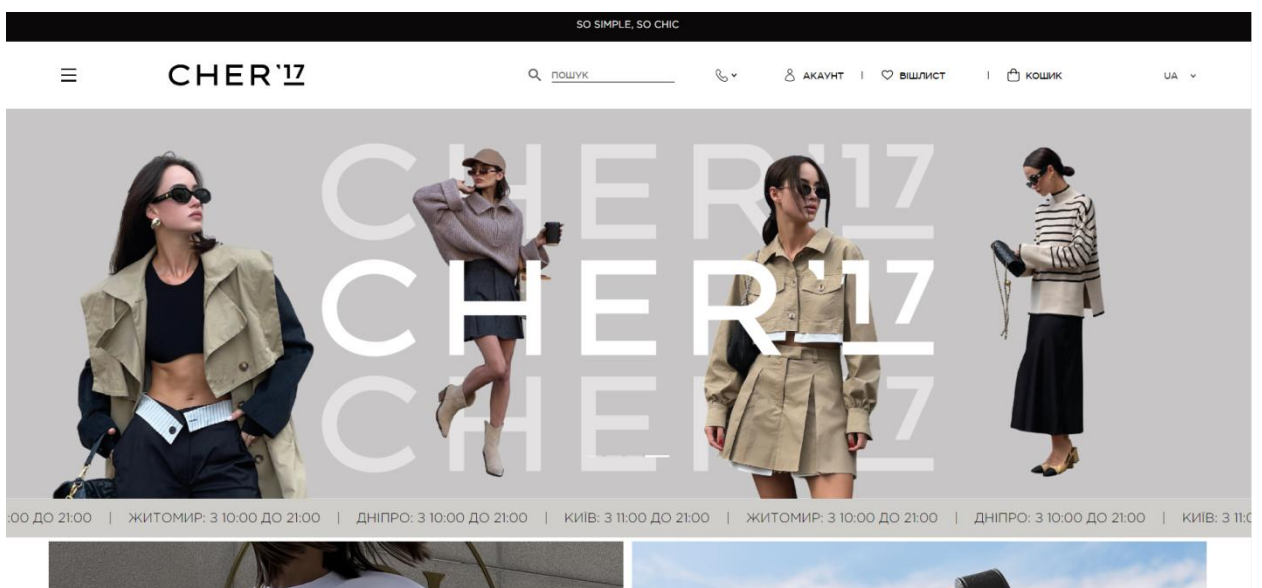


Рис 2.2 – Головний екран Cher17

### 2.3 19store

19store - це український бренд, спеціалізований у продажі топів та корсетів. Сайт 19store розроблений у вигляді односторінкового сайту, що забезпечує максимальну зручність та швидкість навігації для користувачів.

Головна особливість сайту 19store - його односторінкова структура. Весь контент та інформація розміщена на одній сторінці, що спрощує процес переходу між розділами та сприяє швидкому знайденню потрібного товару. Каталог товарів розділений на дві колонки, що дозволяє ефективно відображати широкий асортимент топів та корсетів.

Сучасний дизайн сайту 19store відображає стиль та імідж бренду. Логічно розташовані елементи та інтуїтивно зрозуміла навігація допомагають користувачам швидко зорієнтуватись на сторінці та знайти потрібний товар. Графічні елементи, колірна палітра та типографіка підкреслюють сучасний та стильний образ бренду.

19store надає своїм клієнтам зручне та естетичне віртуальне середовище для покупок топів та корсетів. Легка навігація, зрозумілий дизайн та зручний каталог допомагають користувачам ефективно переглядати та вибирати товари. Завдяки цим функціональним особливостям, 19store забезпечує приємний та зручний досвід покупок, задовольняючи потреби та очікування своїх клієнтів.

19store створює комфортну та естетичну віртуальну платформу для клієнтів, які шукають топи та корсети. Працюючи з простою навігацією, зрозумілим дизайном та зручним каталогом, користувачі можуть ефективно переглядати та вибирати товари. Ці функціональні особливості роблять покупки на 19store приємними та зручними, задовольняючи потреби та очікування клієнтів.

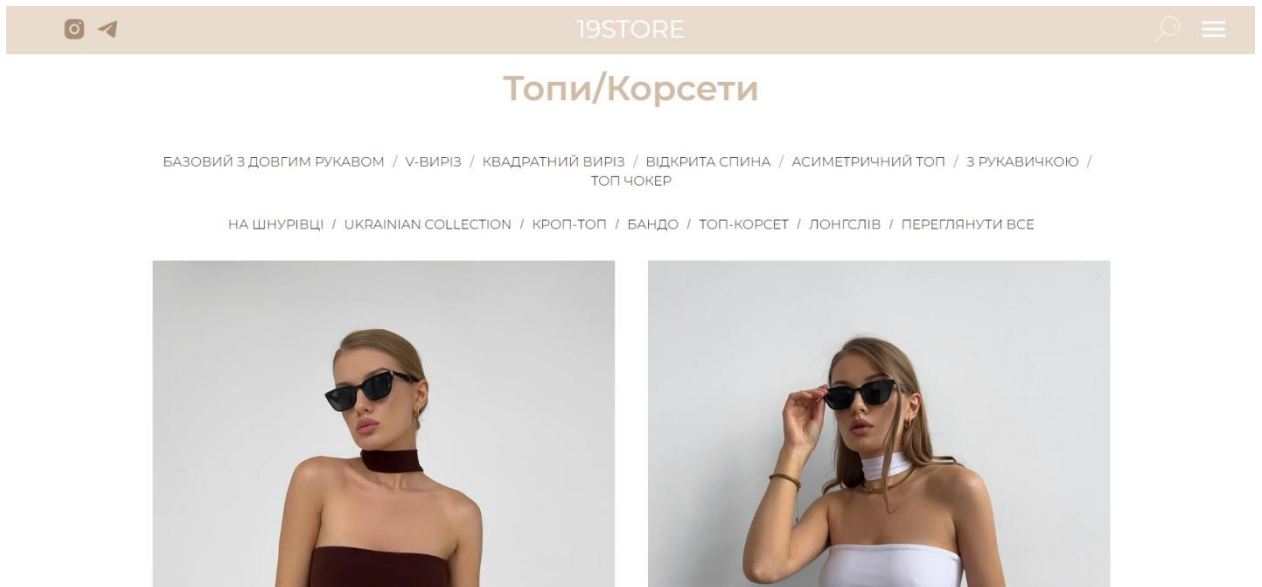


Рис. 2.3 – Головний екран 19store

## 2.4 Papayashop

Papayashop - це український бренд, який спеціалізується на продажі одягу для повсякденного використання та занять спортом. Веб-сайт Papayashop розроблений з метою надати користувачам зручну та ефективну платформу для замовлення високоякісного одягу.

Головний екран сайту Papayashop ретельно спроектований з урахуванням потреб користувачів. Його структура та навігаційне меню дозволяють легко та швидко знайти потрібні товари. Меню розділено на різні категорії одягу, що дозволяє користувачам зосередитись на своїх уподобаннях та швидко знайти необхідні речі.

Особлива увага приділена дизайну сайту Papayashop. Дизайн виконаний з акцентом на функціональність та зручність використання. Він поєднує естетичність та простоту, створюючи приємне візуальне враження. Користувачі зможуть легко орієнтуватись на сайті, знаходити потрібний товар і робити покупки без зайвих зусиль.

Завдяки зручному та привабливому дизайну, а також здійсненню широкого асортименту продукції, Papayashop створює приємне та задоволення від покупок віртуальне середовище для своїх клієнтів. Весь процес від вибору товару до оформлення замовлення є простим та зручним,

сприяючи задоволенню потреб та очікувань користувачів.

Особлива увага була приділена дизайну Parayashop, з фокусом на його функціональність та легкість використання. Дизайн поєднує естетику та простоту, створюючи приємне візуальне сприйняття. Користувачі зможуть з легкістю орієнтуватися на сайті, знаходити потрібний товар і здійснювати покупки без зайвих зусиль.

Завдяки зручному та привабливому дизайну, а також широкому асортименту продукції, Parayashop створює приємне та задоволення від покупок віртуальне середовище для своїх клієнтів. Весь процес, починаючи від вибору товару до оформлення замовлення, є простим та зручним, що сприяє задоволенню потреб та очікувань користувачів.

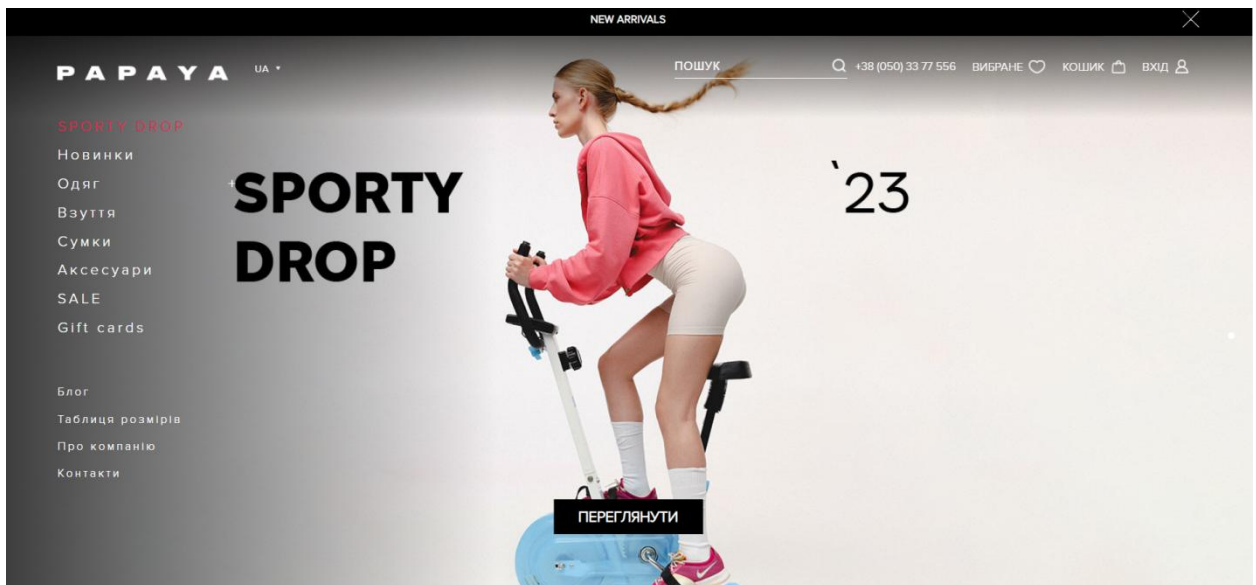


Рис. 2.4 – Головний екран Parayashop

### РОЗДІЛ 3 АРХІТЕКТУРА ВЕБ-САЙТУ

Бізнес-модель сайту з продажу одягу передбачає використання онлайн-платформи для приваблення клієнтів, представлення асортименту товарів і здійснення операцій купівлі-продажу. Основні компоненти такої бізнес-моделі включають:

1. Електронна комерція: Сайт діє як електронний магазин, де клієнти можуть переглядати каталог товарів, детально ознайомлюватися з продукцією і робити покупки відразу через сайт. Це забезпечує глобальний доступ до продукції і дозволяє клієнтам придбати одяг у зручний для них час.

2. Асортимент товарів: Бізнес-модель передбачає розширений вибір одягу, щоб задовольнити різноманітні потреби та вподобання клієнтів. Вона може включати жіночий, чоловічий та дитячий одяг, аксесуари, взуття тощо. Рознообразність асортименту сприяє залученню більш широкої аудиторії.

3. Маркетинг і реклама: Для привертання уваги потенційних клієнтів необхідна ефективна стратегія маркетингу та реклами. Це може включати використання соціальних медіа, контекстної реклами, електронних розсилок, співпрацю з блогерами та впливовими особистостями, проведення акцій та знижок тощо. Метою є залучення нових клієнтів і підтримка вірних покупців.

4. Інтеграція платіжних систем: Бізнес-модель передбачає забезпечення безпечних та зручних операцій оплати для клієнтів. Інтеграція платіжних систем, таких як кредитні картки, електронні гаманці та інші платіжні шлюзи, дозволяє здійснювати онлайн-транзакції безпосередньо на сайті.

5. Логістика та доставка: Однією з важливих складових бізнес-моделі є організація логістики та доставки. Система повинна забезпечувати швидку обробку замовлень, пакування товарів і вчасну доставку до клієнтів. Важливо також враховувати вартість доставки та повернення товару, щоб забезпечити задоволення клієнтів.

6. Обслуговування клієнтів: Важливо мати ефективну систему підтримки клієнтів, яка забезпечує швидке відповіді на запитання, допомогу з проблемами, повернення товарів та інші питання. Задоволені клієнти



частіше повертаються і рекомендують магазин своїм знайомим.

Загалом, бізнес-модель сайту з продажу одягу спрямована на створення зручного та задоволення при покупках середовища для клієнтів, залучення нових клієнтів та підтримку вірних покупців. Вона покликана забезпечити ефективну продаж одягу та збільшення доходів компанії.

### **3.1 Модель сайту**

Після огляду існуючих веб-сайтів з продажу одягу були підкреслені як плюси, так і мінуси цих рішень. Опираючись на це дослідження було розроблено модель веб-сайту, яка поєднує найкращі практики та враховує особливості малого бізнесу в сфері онлайн-торгівлі одягом.

Модель сайту пропонує зручну навігацію та чітку структуру, що дозволяє користувачам швидко знайти потрібний товар. Також було приділено особливу увагу естетиці та привабливості дизайну, забезпечуючи приємний імідж бренду та створюючи позитивний досвід для відвідувачів.

Однак, при розробці моделі сайту я також врахував недоліки, виявлені на існуючих веб-сайтах. Були вжиті заходи для поліпшення швидкості завантаження сторінок, оптимізації для пошукових систем, а також забезпечення простоти використання інтерфейсу та процесу оформлення замовлення.

Модель сайту ґрунтується на використанні сучасних технологій, таких як HTML, CSS, Vue.js, та включає інтеграцію платіжної системи LiqPay, що забезпечує зручність та безпеку оплати для користувачів.

В результаті, модель сайту інтернет-магазину для малого бізнесу з продажу одягу поєднує кращі практики, враховує попередній аналіз ринку та потреби цільової аудиторії.

Після аналізу наявних веб-сайтів, які продають одяг, було виявлено як позитивні, так і негативні аспекти цих рішень. Використовуючи це дослідження, була розроблена модель веб-сайту, яка поєднує найкращі практики та враховує особливості малого бізнесу в сфері онлайн-торгівлі

одягом.

Модель сайту пропонує зручну навігацію та чітку структуру, що дозволяє користувачам швидко знайти потрібний товар. Особлива увага була приділена естетиці та привабливості дизайну, створюючи приємний імідж бренду та позитивний досвід для відвідувачів.

При розробці моделі сайту були також враховані недоліки, виявлені на інших веб-сайтах. Були вжиті заходи для поліпшення швидкості завантаження сторінок, оптимізації для пошукових систем, а також забезпечення простоти використання інтерфейсу та процесу оформлення замовлення.

Модель сайту базується на використанні сучасних технологій, таких як HTML, CSS, Vue.js, і включає інтеграцію платіжної системи LiqPay, що забезпечує зручність та безпеку оплати для користувачів.

В результаті, модель сайту для інтернет-магазину малого бізнесу з продажу одягу поєднує найкращі практики, враховує аналіз ринку та потреби цільової аудиторії.

### **3.1.1 Макет головної сторінки**

На рисунку (рис 3.1) представлений макет головної сторінки сайту. Верхню частину сторінки займає логотип бренду, який візуально ідентифікує сайт. Також на цій частині розташовані п'ять вкладинок, які служать для навігації по різних розділах сайту, надаючи користувачеві доступ до різноманітного вмісту та категорій товарів.

Поруч з вкладинками розміщено пошукове вікно, яке дозволяє користувачеві швидко знайти потрібний товар або інформацію. Крім того, видно іконку кошика, яка вказує на наявність товарів у корзині, а також надає можливість переглянути вміст кошика та здійснити покупку.

У нижній частині головної сторінки знаходиться головне зображення, яке привертає увагу відвідувача і створює перший враження про сайт. Це зображення може бути презентацією нової колекції, рекламним зображенням

або іншим елементом, що відображає стиль та концепцію бренду.

Макет головної сторінки сайту добре збалансований та візуально привабливий. Він створений з метою надання зручності та зрозумілості для користувача, а також підсилення брендової ідентичності. Чітко визначені елементи, яскраві зображення та зручна навігація допомагають створити приємний досвід перегляду сайту і сприяють залученню користувачів до взаємодії з контентом та покупками.

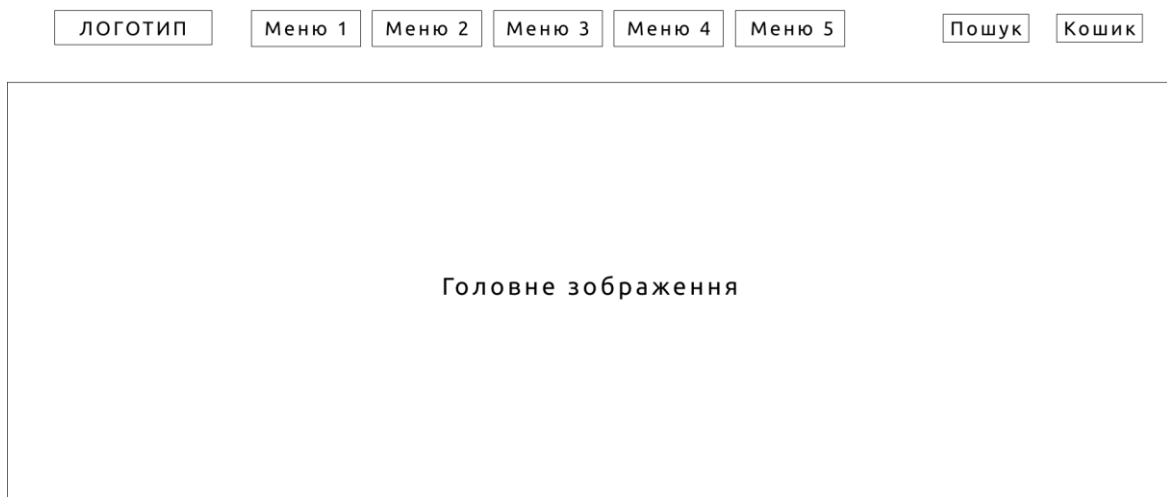


Рис. 3.1 – Макет головної сторінки

### 3.1.2 Макет каталогу

На рисунку (рис 3.2) показано макет каталогу сайту. У верхній частині сторінки знаходиться логотип бренду та п'ять вкладинок для навігації по різних розділах сайту. Також видно пошукове вікно та кошик, які залишаються статичними і доступними в будь-який час, щоб користувач мав зручний доступ до навігації та можливості перегляду вмісту кошика.

Нижче зліва розташований блок фільтрів, який дозволяє користувачеві здійснювати вибір товару за різними критеріями, такими як розмір, колір або тип продукту. Цей блок фільтрів спрощує процес пошуку та відбору товарів згідно з вподобаннями користувача.

З правої сторони макету розташований блок з товаром, де вказана назва та ціна. Цей блок відображає конкретний товар з каталогу, дозволяючи

користувачеві швидко ознайомитися з його характеристиками та ціною.

Макет каталогу сайту добре структурований та пропонує зручний спосіб перегляду товарів. Логотип та вкладинки забезпечують постійний доступ до навігації, що полегшує користувачам орієнтуватися на сайті. Блок фільтрів допомагає скоротити час пошуку, а блок з товаром забезпечує швидку оцінку доступних пропозицій. В цілому, макет каталогу сайту надає зручну та зрозумілу структуру, що допомагає користувачам ефективно переглядати і обирати товари.



Рис. 3.2 – Макет каталогу

### 3.1.3 Макет сторінки детальної інформації про товар

На рисунку (рис 3.3) представлений макет сторінки з детальною інформацією про товар. У верхній частині сторінки знаходяться логотип бренду та п'ять вкладинок для навігації по різних розділах сайту. Також видно пошукове вікно та кошик, які залишаються статичними та доступними на всій сторінці для зручності користувача.

Нижче зліва розташований колаж з фотографіями товару у різних ракурсах. Це дозволяє користувачу отримати більш детальне уявлення про вигляд товару з різних сторін. З правої сторони знаходиться назва товару, його ціна та розмірна сітка, що допомагає користувачеві швидко зорієнтуватися в характеристиках товару. Також поряд з ними розташована

кнопка "Купити", яка надає можливість додати товар до кошика з одного кліку.

Крім того, на макеті присутній блок з детальним описом товару. Цей блок містить інформацію про особливості, матеріал, розміри та інші характеристики товару, що допомагає користувачеві зробити інформований вибір.

Макет сторінки детальної інформації про товар використовує послідовне розташування елементів, забезпечуючи логічну структуру та зручну навігацію для користувачів. Наявність фотографій у різних ракурсах дозволяє детально розглянути товар, а додаткова інформація та кнопка "Купити" полегшують процес прийняття рішення щодо покупки. В цілому, макет сторінки детальної інформації про товар пропонує зручний спосіб ознайомитися з характеристиками товару та зробити покупку в одному місці.

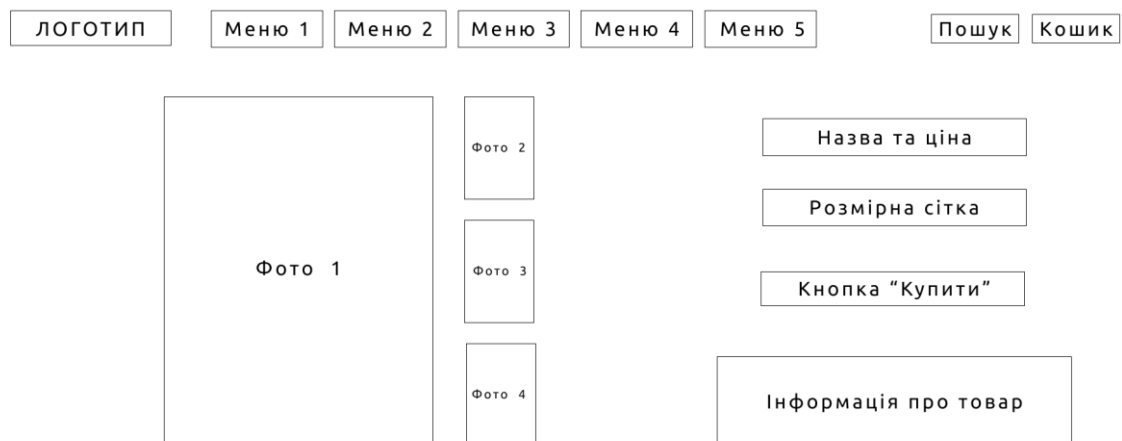


Рис. 3.3 – Макет сторінки детальної інформації про товар

### 3.1.4 Макет сторінки оформлення замовлення

На рисунку (рис 3.4) зображений макет сторінки оформлення замовлення. У верхній частині сторінки розташовані логотип бренду та п'ять вкладинок для навігації. Також видно пошукове вікно та кошик, які залишаються статичними, забезпечуючи зручний доступ до навігації на всій сторінці.

Нижче зліва розташований напис "Оформлення замовлення", що чітко вказує на призначення розділу. Нижче знаходяться поля для введення особистих даних покупця, які необхідні для доставки товару. Також є поле для введення коментаря покупця, яке дозволяє залишити додаткові вказівки або побажання стосовно замовлення.

З правої сторони розміщений блок з назвою "Кошик", який відображає перелік товарів, що знаходяться у кошику покупця. Далі присутнє фото товару, назва товару та обраний розмір, що надають детальну інформацію про замовлення. Під назвою товару розташована ціна, а поряд з нею розміщена кнопка "Купити", яка дозволяє підтвердити замовлення.

Макет сторінки оформлення замовлення пропонує чітку структуру та інтуїтивно зрозумілий інтерфейс для користувача. Він забезпечує зручний доступ до необхідних полів вводу, а також чітке відображення деталей замовлення.

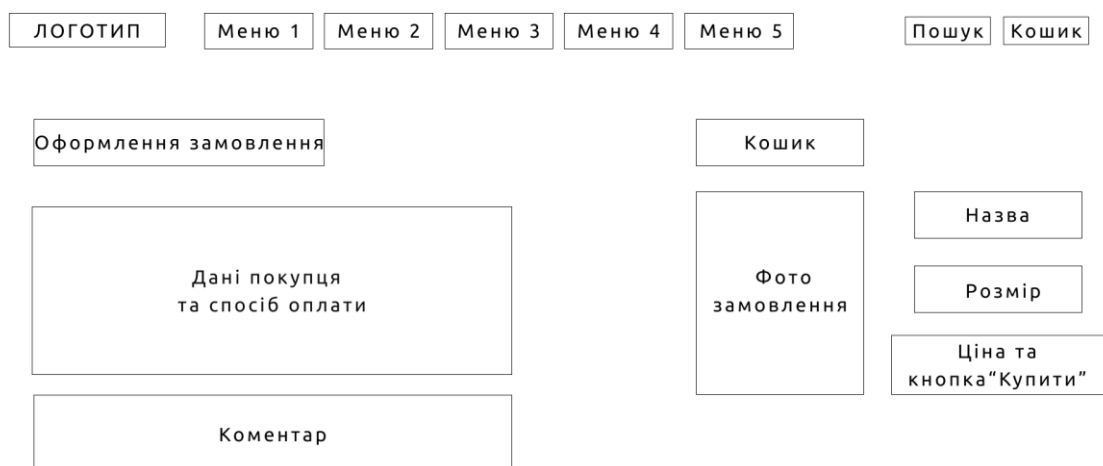


Рис. 3.4 – Макет оформлення замовлення

## 3.2 Архітектура сайту

### 3.2.1 IDE IntelliJ IDEA

IntelliJ IDEA є однією з найпопулярніших та потужних інтегрованих середовищ розробки (Integrated Development Environment - IDE) для програмування на мовах Java, Kotlin, JavaScript, інших мовах програмування та веб-розробки. Розроблена компанією JetBrains, IntelliJ IDEA надає широкий набір функціональності та інструментів для поліпшення продуктивності розробників.

Основні особливості та переваги IntelliJ IDEA:

1. Підтримка багатьох мов програмування: IntelliJ IDEA надає можливість розробки на різних мовах програмування, включаючи Java, Kotlin, JavaScript, TypeScript, SQL, HTML, CSS та інші. Це робить його універсальним середовищем для розробки різноманітних проектів.

2. Розширена підтримка фреймворків: IntelliJ IDEA має потужні інструменти для підтримки популярних фреймворків, таких як Spring, Hibernate, Android, Angular, React, Vue.js та багатьох інших. Це спрощує розробку застосунків на основі цих фреймворків та підвищує продуктивність розробників.

3. Розумний редактор коду: IntelliJ IDEA надає потужний редактор коду з автодоповненням, перевіркою синтаксису, рефакторингом, вбудованими інспекторами коду та багатьма іншими функціями. Це допомагає розробникам писати код швидше та з меншими помилками.

4. Інтеграція з системами контролю версій: IntelliJ IDEA підтримує популярні системи контролю версій, такі як Git, Subversion (SVN), Mercurial та інші. Це дозволяє легко працювати зі змінами в коді, використовувати гілки та співпрацювати з іншими розробниками.

5. Інструменти для тестування та налагодження: IntelliJ IDEA надає вбудовані інструменти для написання та виконання автоматизованих тестів, а також для налагодження програм. Це допомагає забезпечити якість коду та виявляти помилки швидше.

6. Екосистема плагінів: IntelliJ IDEA має широку екосистему плагінів, яка дозволяє розширювати його функціональність та відповідати конкретним потребам розробників.

IntelliJ IDEA є популярним вибором серед розробників завдяки своїм потужним можливостям, продуктивній роботі з кодом та дружньому інтерфейсу користувача. Воно допомагає прискорити розробку програмного забезпечення та полегшити рутинні завдання, що дозволяє розробникам більше уваги приділити креативному процесу створення програм.

### 3.2.2 HTML

HTML (HyperText Markup Language) є стандартною мовою розмітки, використовуваною для створення та структурування веб-сторінок. Вона використовується для опису вмісту сторінки, такого як текст, зображення, посилання, таблиці, форми, аудіо та відео елементи та багато іншого.

Основні особливості та концепції HTML:

1. Теги: HTML використовує теги для визначення структури та типу вмісту, який буде відображено на сторінці. Кожен тег починається зі знака `<` і закінчується `>`. Наприклад, `

` використовується для визначення абзацу, `` - для вставки зображення, `` - для створення посилання.

2. Елементи: Елементи є будівельними блоками HTML-документа і можуть містити інші елементи або текст. Вони визначаються за допомогою тегів і складаються з відкриваючого тега `<тег>` та закриваючого тега ``. Наприклад, `

# ` і `` визначають заголовок першого рівня.

3. Атрибути: Атрибути надають додаткову інформацію про елементи HTML. Вони вказуються всередині відкриваючого тега та додаються за допомогою імені атрибута та його значення. Наприклад, `href` - атрибут, використовуваний для вказання адреси посилання.

4. Структура: HTML має деревоподібну структуру, в якій елементи можуть бути вкладеними один в одного. Головний елемент сторінки є ``, а його безпосередніми дочірніми елементи - `` та ``. Елемент `` містить метадані та посилання на зовнішні ресурси, а



елемент ``<body>`` містить вміст, який буде відображено на сторінці.

5. Семантика: HTML надає набір семантичних тегів, які дозволяють вказувати смислову роль елементів на сторінці. Наприклад, ``<header>`` використовується для визначення верхньої частини сторінки, ``<nav>`` - для навігаційного меню, ``<section>`` - для логічної секції сторінки і т.д. Використання семантичних тегів поліпшує доступність та інтерпретацію сторінки пошуковими системами.

HTML є основою веб-розробки і використовується для створення структури та вмісту веб-сторінок. Разом з CSS та JavaScript він дозволяє створювати динамічні та привабливі інтерфейси, що допомагають залучити користувачів та забезпечити їм зручний досвід взаємодії з веб-сайтами.

### 3.2.3 CSS

CSS (Cascading Style Sheets) - це мова стилів, що використовується для визначення зовнішнього вигляду та форматування веб-сторінок, написаних з використанням HTML або інших мов розмітки.

Основні концепції та можливості CSS:

1. Відокремлення зовнішнього вигляду: CSS дозволяє відокремити стилі від HTML-структури сторінки. Це дозволяє розробникам ефективно керувати виглядом багатьох сторінок, змінюючи лише один CSS-файл.

2. Вибір елементів: CSS використовує селектори для вибору елементів, які мають бути стилізовані. Селектори можуть базуватися на типі елемента, класі, ідентифікаторі, атрибутах та інших характеристиках елементів.

3. Властивості та значення: CSS використовує властивості для встановлення конкретних стилів елементів. Властивості можуть охоплювати кольори, шрифти, розміри, відступи, фони, позиціонування та багато інших аспектів зовнішнього вигляду.

4. Каскадування та спадкування: CSS працює за принципом каскадування, де властивості можуть бути успадковані від батьківських елементів. Це дозволяє раціоналізувати стилі та зменшити дублювання коду.

5. Медіа-запити: CSS підтримує медіа-запити, які дозволяють

змінювати стилі в залежності від властивостей пристрою, на якому відображається сторінка. Це дає змогу створювати адаптивні та резинові дизайни, які працюють на різних пристроях.

6. Анімація та переходи: CSS дозволяє створювати анімацію та переходи, що надають сторінкам живості та динамічності. За допомогою CSS можна керувати зміною властивостей з плинними ефектами.

CSS використовується для створення стильного та привабливого зовнішнього вигляду веб-сторінок. Він співпрацює з HTML та JavaScript, допомагаючи реалізувати багатофункціональні та ефектні інтерфейси для користувачів.

### 3.2.4 Vue.js

Vue.js - це прогресивний фреймворк JavaScript для розробки користувацьких інтерфейсів. Він дозволяє розробникам побудувати ефективні та масштабовані односторінкові додатки (SPA) та інтерактивні компоненти веб-сайтів.

Основні особливості та переваги Vue.js:

1. Простота використання: Vue.js пропонує інтуїтивний синтаксис, що дозволяє швидко оволодіти його основами та почати розробку. Він також має документацію, яка детально пояснює його можливості та надає приклади використання.

2. Компонентна архітектура: Vue.js базується на компонентній архітектурі, що дозволяє розбити веб-інтерфейс на незалежні компоненти. Це спрощує розробку, тестування та підтримку коду. Компоненти можна повторно використовувати, що полегшує розширення та модульність додатків.

3. Реактивність: Vue.js використовує систему зв'язку даних з відслідковуванням змін (reactivity system), яка автоматично оновлює відображення даних, коли їх значення змінюються. Це дозволяє створювати динамічні та реактивні інтерфейси без необхідності вручну оновлювати стан

сторінки.

4. Директиви: Vue.js має набір вбудованих директив, які дозволяють маніпулювати DOM елементами та відображенням. Наприклад, директива `v-bind` використовується для зв'язування даних з властивостями елементів, а директива `v-for` - для повторення елементів списку.

5. Розширення за допомогою плагінів: Vue.js має велику екосистему плагінів, які дозволяють розширити його можливості та функціональність. За допомогою плагінів можна додати підтримку сторінок маршрутизації, керування станом додатка, інтеграцію з іншими бібліотеками та багато іншого.

Vue.js є дуже популярним фреймворком серед розробників, оскільки він поєднує простоту використання, потужність та розширюваність, дозволяючи швидко створювати високоякісні та інтерактивні веб-додатки.

### 3.2.5 SQL

SQL (Structured Query Language) - це мова запитів для керування реляційними базами даних. Вона використовується для створення, модифікації та отримання даних з бази даних.

Основні поняття та можливості SQL:

1. Структура бази даних: SQL дозволяє створювати структуру бази даних, включаючи таблиці для зберігання даних, відношення між таблицями та обмеження цілісності даних. Вона надає можливість описати схему бази даних з використанням таблиць, стовпців, ключів та інших елементів.

2. Запити до бази даних: SQL дозволяє виконувати різноманітні запити до бази даних для отримання потрібної інформації. Запити можуть бути простими (наприклад, вибірка всіх записів з таблиці) або складними (наприклад, комбінація декількох таблиць та умов для отримання певних даних).

3. Умови та фільтрація даних: SQL дозволяє використовувати умови для фільтрації даних, що отримуються з бази даних. За допомогою умов

можна вибирати записи, що відповідають певним критеріям (наприклад, десяткова кількість товару більше заданої величини) або виконують певні логічні умови.

4. Сортування та групування даних: SQL дозволяє сортувати отримані дані за певними критеріями, наприклад, за алфавітом або числовим значенням. Вона також підтримує групування даних для виконання агрегатних операцій, таких як сума, середнє значення або підрахунок кількості записів.

5. Модифікація даних: SQL дозволяє додавати, оновлювати та видаляти дані в базі даних. За допомогою відповідних запитів можна вставляти нові записи в таблиці, оновлювати значення певних полів або видаляти непотрібні записи.

SQL є стандартною мовою для роботи з реляційними базами даних і використовується в багатьох популярних системах керування базами даних, таких як MySQL, PostgreSQL, Oracle та інші. Вона є потужним інструментом для роботи з даними та забезпечує ефективну і зручну роботу з базами даних.

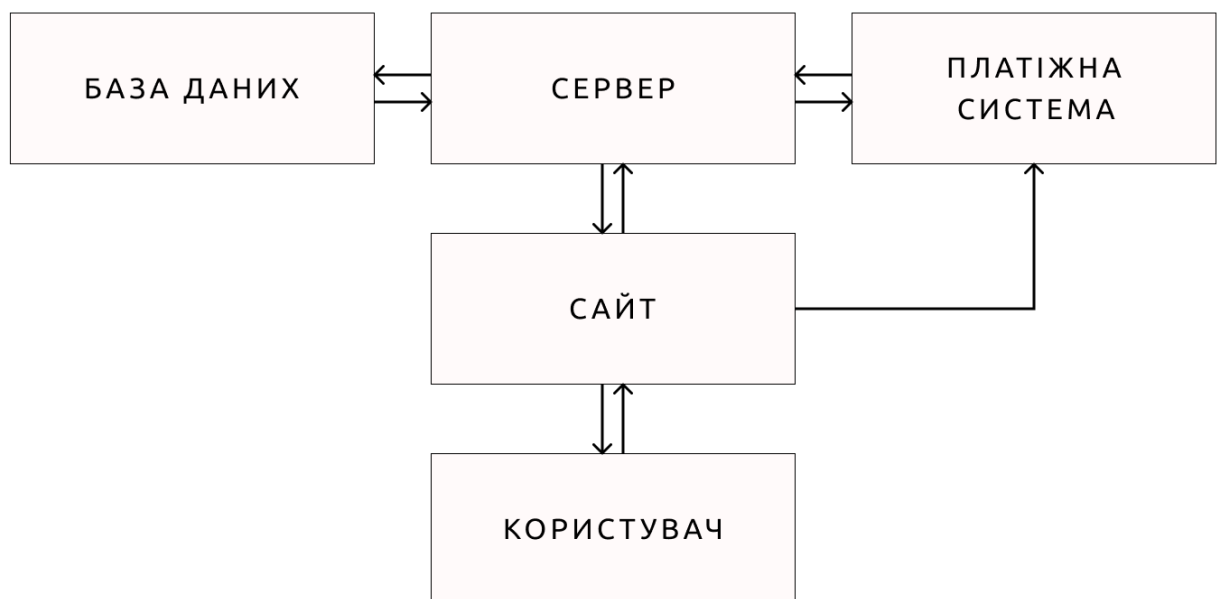


Рис. 3.5 – Схема роботи сайту

### 3.3 Платіжна система LiqPay

LiqPay є платіжною системою, яка забезпечує онлайн-платежі та

електронну комерцію. Розроблена українською компанією "ПриватБанк", LiqPay надає зручні та безпечні способи оплати на веб-сайтах та мобільних додатках.

Основні можливості LiqPay включають:

1. Електронні платежі: LiqPay дозволяє користувачам здійснювати платежі за допомогою банківських карт, електронних гаманців та інших способів оплати. Користувачі можуть швидко та зручно здійснювати покупки в інтернет-магазині без необхідності вводити багаторазово свої платіжні дані.

2. Перекази коштів: LiqPay дозволяє переказувати гроші між користувачами за допомогою мобільних номерів або електронних гаманців. Це зручний спосіб надсилання грошей родичам, друзям чи партнерам.

3. Оплата рахунків: Система LiqPay також надає можливість оплати комунальних послуг, мобільного зв'язку, інтернет-провайдерів та інших рахунків. Користувачі можуть легко та швидко оплачувати свої рахунки без відвідування офісів чи використання готівки.

4. API для розробників: LiqPay надає розробникам доступ до своєї API, що дозволяє інтегрувати платіжну систему у власні веб-сайти та додатки. Це дозволяє створювати різноманітні рішення, включаючи інтернет-магазини, платіжні ворота та інші сервіси.

5. Безпека: LiqPay використовує різні заходи для забезпечення безпеки платежів, включаючи шифрування даних, захист від шахрайства та виявлення підозрілих активностей.

Загалом, LiqPay є найпопулярнішим інструментом для забезпечення онлайн-платежів та електронної комерції. Вона дозволяє підприємствам та користувачам зручно, безпечно та ефективно здійснювати платежі та передавати кошти.

Також LiqPay надає можливість використовувати тестовий режим для інтеграції та тестування платіжної системи без реальних фінансових транзакцій. Це дозволяє вам перевірити правильність інтеграції, переконатися, що всі функціональні можливості працюють належним чином,

а також перевірити безпеку платіжних операцій перед запуском виробничого середовища.

У тестовому режимі LiqPay надає вам тестові дані, такі як тестові карткові дані, які ви можете використовувати для симуляції платежів. Це дозволяє вам виконати різні тестові сценарії, включаючи успішні та неуспішні транзакції, а також перевірити обробку помилок та винятків.

Під час роботи у тестовому режимі вам не будуть стягуватися реальні кошти з ваших карт або рахунків, оскільки всі платежі відбуватимуться у тестовому середовищі. Це дозволяє вам вільно експериментувати та вирішувати всі потенційні проблеми, що виникають під час інтеграції платіжної системи.

Щоб використовувати тестовий режим LiqPay, вам потрібно мати тестовий обліковий запис, який можна отримати через офіційний веб-сайт LiqPay. Вони нададуть вам тестові API-ключі та інструкції щодо налаштування тестового середовища.

Важливо пам'ятати, що перед переходом до виробничого середовища вам потрібно відключити тестовий режим та налаштувати роботу з реальними платежами, використовуючи правильні API-ключі та налаштування.

Використання тестового режиму LiqPay дозволяє вам максимально підготуватися до запуску платіжної системи на живому сайті, забезпечуючи безпеку, надійність та відповідність всім функціональним вимогам вашого веб-магазину.

LiqPay пропонує вам можливість використовувати тестовий режим, який надає тестові дані, включаючи тестові карткові дані, для симуляції платежів. Це дає вам можливість провести різні тестові сценарії, включаючи успішні та неуспішні транзакції, а також перевірити обробку помилок та винятків.

У тестовому режимі не будуть стягуватися реальні кошти з ваших карт або рахунків, оскільки всі платежі відбуватимуться у тестовому середовищі. Це дозволяє вам вільно експериментувати та вирішувати всі потенційні

проблеми, які виникають під час інтеграції платіжної системи.

Для використання тестового режиму LiqPay вам необхідно мати тестовий обліковий запис, який можна отримати через офіційний веб-сайт LiqPay. Вам будуть надані тестові API-ключі та інструкції щодо налаштування тестового середовища.

Важливо пам'ятати, що перед переходом до робочого середовища вам потрібно відключити тестовий режим та налаштувати роботу з реальними платежами, використовуючи правильні API-ключі та налаштування.

Використання тестового режиму LiqPay дозволяє вам максимально підготуватися до запуску платіжної системи на живому сайті, забезпечуючи безпеку, надійність та відповідність всім функціональним вимогам вашого веб-магазину.

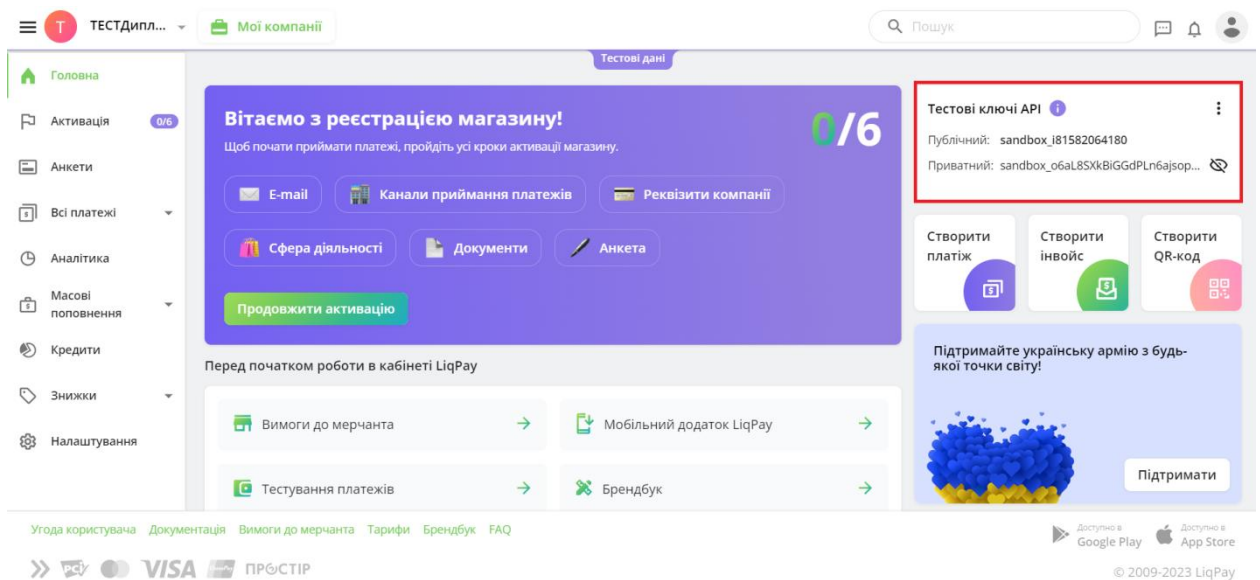


Рис. 3.6 – Тестовий режим LiqPay

### 3.4 Просування веб-сайту

Просування сайтів з продажу одягу є важливим етапом для залучення цільової аудиторії та збільшення продажів. Ось декілька стратегій, які можна використовувати для просування таких сайтів:

1. Цільове спрямування аудиторії: Визначте свою цільову аудиторію і розробіть маркетингові стратегії, спрямовані на їх потреби та інтереси.

Вивчіть їхній стиль, вподобання та покупний потенціал, щоб підібрати відповідні маркетингові канали та повідомлення.

2. Оптимізація пошукових систем (SEO): Виконайте пошукову оптимізацію вашого сайту, включаючи ключові слова, оптимізовані описи продуктів, унікальний контент та покращену швидкість завантаження сторінок. Це допоможе вам підвищити видимість вашого сайту у пошукових системах і залучити більше органічного трафіку.

3. Рекламні кампанії: Використовуйте різні форми реклами, такі як контекстна реклама, банерна реклама, соціальна реклама та ретаргетинг, щоб залучити увагу потенційних покупців. Рекламні кампанії можна налаштувати на платформах, таких як Google Ads, Facebook Ads, Instagram Ads та інші, з урахуванням вашої цільової аудиторії.

4. Соціальні медіа: Будуйте сильну присутність в соціальних медіа, створюючи привабливий контент і взаємодіючи зі своєю аудиторією. Публікуйте фотографії ваших товарів, організовуйте конкурси, запусайте розпродажі та залучайте відгуки і рекомендації від задоволених покупців. Соціальні медіа можуть допомогти вам побудувати лояльність клієнтів та збільшити свідомість про ваш бренд.

5. Впливові особистості: Співпрацюйте з впливовими особистостями і блогерами, які мають аудиторію, зацікавлену в моді та одязі. Вони можуть допомогти просунути ваш бренд через спонсорвані публікації, огляди продуктів або спільні проекти, привертаючи увагу своїх прихильників до вашого сайту.

6. Електронна пошта: Збирайте адреси електронної пошти своїх клієнтів та відвідувачів сайту, щоб розпочати електронний маркетинг. Відправляйте персоналізовані розсилки, пропозиції та акції, щоб привернути увагу та зберегти зацікавленість покупців.

7. Відгуки та рекомендації: Надайте можливість клієнтам залишати відгуки про свої покупки на вашому сайті або на сторінках відгуків. Позитивні відгуки та рекомендації можуть значно збільшити довіру до



вашого бренду та підвищити конверсію.

Ці стратегії просування можна комбінувати та адаптувати залежно від вашої цільової аудиторії, бюджету та ресурсів. Важливо тримати руку на пульсі модних тенденцій та постійно працювати над покращенням користувацького досвіду та задоволення потреб клієнтів.

Ефективне просування сайтів з продажу одягу є невід'ємною складовою успішної стратегії, яка допомагає залучати цільову аудиторію та значно збільшувати обсяги продажів. У сучасному конкурентному світі електронної торгівлі, де кількість онлайн-магазинів зростає з кожним днем, важливо використовувати ефективні методи та стратегії для забезпечення видимості вашого сайту і привертання уваги потенційних покупців.

Одяг є однією з найпопулярніших категорій товарів, тому конкуренція в цій галузі ще більш жорстка. Для того, щоб виділитися серед безлічі інших магазинів і привернути увагу своєї цільової аудиторії, варто використовувати комплексний підхід до просування.

Початковим кроком є оптимізація вашого сайту для пошукових систем (SEO). Це включає в себе розробку унікального та цільованого контенту, оптимізацію мета-тегів та заголовків сторінок, побудову якісної внутрішньої структури сайту, а також отримання якісних зовнішніх посилань. Це допоможе покращити позиції вашого сайту в пошукових системах і забезпечить більше органічного трафіку.

Паралельно з SEO, варто активно використовувати контекстну рекламу, таку як реклама на пошукових системах або соціальних медіа. Це дасть можливість вам з'явитися на першій сторінці пошукових систем за ключовими запитами, а також розмістити рекламні оголошення перед вашою цільовою аудиторією на популярних платформах соціальних медіа.

Не потрібно забувати і про важливість контент-маркетингу. Створюйте цікавий, корисний та релевантний контент, який привертає увагу вашої аудиторії. Публікуйте блоги, статті, інфографіку, відео та інші формати контенту, які стимулюють відвідування вашого сайту та сприяють

розповсюдженню інформації про ваш бренд.

Також, важливо використовувати соціальні медіа для просування вашого сайту з продажу одягу. Залучайте свою аудиторію через публікації зображень, конкурси, знижки та акції. Будуйте спільноту навколо вашого бренду і взаємодійте зі своїми клієнтами, щоб підтримувати відносини та забезпечувати лояльність.

Не забувайте і про важливість аналізу та вимірювання результатів. Використовуйте аналітичні інструменти для відстеження трафіку, конверсій, поведінки користувачів та інших метрик, щоб постійно вдосконалювати свою стратегію просування та досягати кращих результатів.

Узагалі, просування сайту з продажу одягу вимагає постійного вдосконалення, експериментів та аналізу. Будьте відкритими до нових можливостей та тенденцій в електронній комерції, і завжди підлаштовуйте свою стратегію відповідно до потреб своєї цільової аудиторії.

### **3.5 Бізнес модель**

Для створення веб-сайту інтернет-магазину з продажу одягу були використані передові інструменти та технології, які забезпечують ефективну розробку і функціональність сайту. Розробка веб-сайту проводилась у потужному середовищі розробки IDE IntelliJ IDEA, яке дозволяє комфортно працювати з кодом і забезпечує продуктивність розробки.

Для створення структури та розмітки сторінок сайту була використана мова програмування HTML. CSS використовувався для стилізації та візуального оформлення елементів сайту, що сприяє привабливому зовнішньому вигляду. Для реалізації динамічності та інтерактивності веб-сайту був використаний Vue.js, що дозволяє реалізувати компонентний підхід до розробки та покращити користувацький досвід.

Мова структурованого запиту до баз даних SQL була використана для зберігання та керування даними, пов'язаними з інтернет-магазином. Це дозволяє ефективно організувати та отримувати необхідну інформацію про

товари, замовлення та інші деталі, що стосуються функціонування магазину.

Завдяки цим потужним інструментам і технологіям був створений веб-сайт інтернет-магазину, який надає багатофункціональне та зручне середовище для клієнтів. Це допомагає їм здійснювати покупки онлайн з комфортом, надійністю та задоволенням.

## РОЗДІЛ 4 АПРОБАЦІЯ ВЕБ-ДОДАТКУ

У процесі розробки веб-сайту з продажу одягу для малого бізнесу було створено сайт під назвою "For you". Цей проект був оснований на ретельному аналізі існуючих рішень на ринку, з урахуванням їх позитивних та негативних аспектів. Аналіз допоміг зрозуміти потреби цільової аудиторії та визначити ключові функції, які повинен мати веб-сайт.

На основі цього аналізу був розроблений макет веб-сайту, який передбачав логічну структуру та ефективне розміщення елементів на сторінках. Макет включав головну сторінку, каталог товарів, сторінку детальної інформації про товар, а також сторінку оформлення замовлення. Важливим аспектом було забезпечення простоти навігації, естетичного дизайну та зручності використання для користувачів.

Після розробки макету, результати аналізу та вимог до сайту були втілені в повноцінний дизайн веб-сайту "For you". У процесі розробки були враховані принципи ефективності, естетичності та функціональності. Дизайн сайту був створений з урахуванням потреб цільової аудиторії та бренду, з фокусом на зручність використання та привабливий зовнішній вигляд.

Результатом роботи над проектом став веб-сайт "For you", який поєднує естетичний дизайн зі зручною навігацією та високою функціональністю. Цей веб-сайт пропонує користувачам зручну та привабливу платформу для вибору та придбання одягу, задовольняючи їхні потреби і сприяючи розвитку малого бізнесу в сфері моди.

### 4.1 Головний екран

На рисунку (рис. 4.1) зображено головну сторінку веб-сайту "For you". При першому відвідуванні сайту, користувача вітають п'ять вкладинок, розташованих у верхній частині екрана. Кожна вкладка представляє різні категорії товарів, дозволяючи швидко перейти до бажаного розділу.

Зверху також розташована кнопка пошуку, яка дозволяє користувачеві здійснити пошук конкретного товару за назвою, категорією або іншими параметрами. Кнопка кошика виділяється окремо, нагадуючи про наявність

товарів, які користувач додав до свого кошика. Це забезпечує легкий доступ до навігації та контролю над обраними товарами.

Нижче на головній сторінці розташоване головне зображення, яке привертає увагу відвідувача. Воно розтягнуте на всю ширину екрана та зосереджене посередині, привертаючи увагу до колекції "LIMITED EDITION". Це підкреслює унікальність та ексклюзивність товарів, які доступні на сайті.

Такий розташунок елементів на головній сторінці сприяє зручності використання сайту та швидкому знаходженню необхідних товарів. Він створює привабливе та інформативне перше враження, що спонукає користувача продовжити перегляд асортименту і зробити покупку.

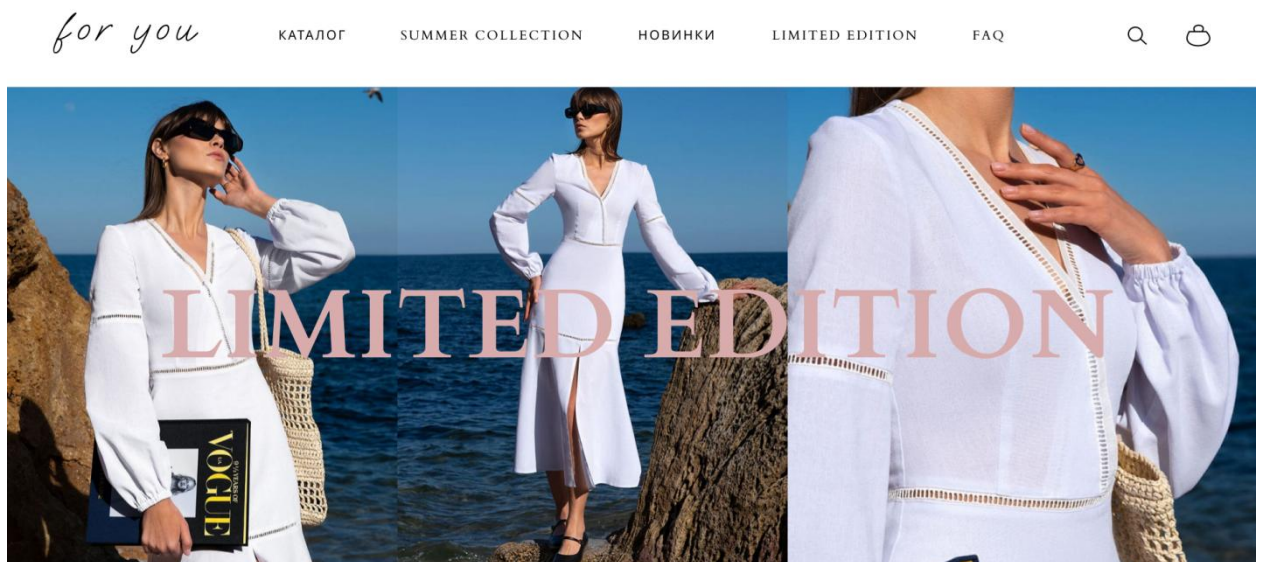


Рис. 4.1 – Головний екран веб-сайту

## 4.2 Сторінка каталогу веб-сайту

На рисунку (рис. 4.2) зображено сторінку каталогу веб-сайту "For you". Як і на головній сторінці, у верхній частині сторінки розташовані п'ять вкладинок з різними категоріями товарів, а також кнопка пошуку та кошик. Ці елементи залишаються статичними, що забезпечує зручну навігацію по сайту незалежно від місця перебування користувача на сторінці.

Знизу, ліворуч розташований блок фільтрів, який допомагає користувачу звужувати вибір товарів за певними критеріями. Цей блок містить два відкриті чекбокси "В наявності" та "Під замовлення", що дозволяють обрати товари відповідно до їх доступності. Нижче розташовані три випадаючі меню "Тип одягу", "Тип взуття" та "Розмір", які дозволяють користувачеві швидко фільтрувати товари за вказаними параметрами. Це створює зручний і ефективний спосіб пошуку і зменшує завантаження сторінки візуальним шумом.

З правої сторони сторінки розміщені самі товари, розташовані по три в ряд. Це дозволяє показати достатньо товарів без перенавантаження сторінки і забезпечує оптимальний розмір фотографій. Під кожним товаром розміщена його назва та ціна, що надає користувачу додаткову інформацію про товар.

Таке розташування елементів на сторінці каталогу дозволяє користувачу зручно переглядати товари, фільтрувати їх за різними критеріями та знайти бажаний продукт. Він також забезпечує приємний візуальний досвід та сприяє легкому здійсненню покупки.

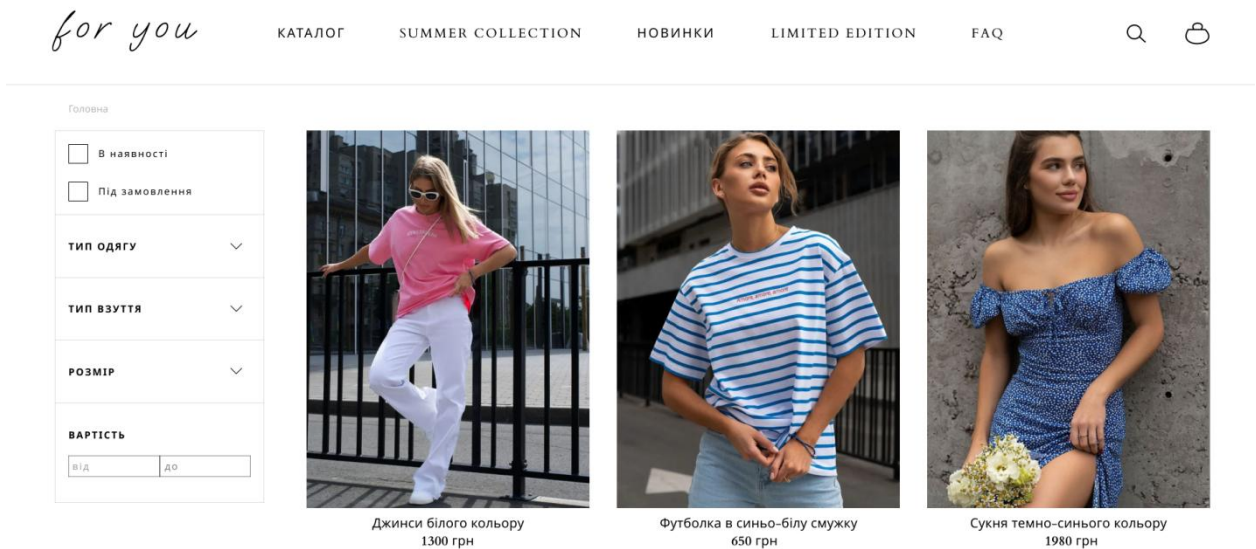


Рис. 4.2 – Сторінка каталогу веб-сайту

### 4.3 Детальна інформація про товар

На рисунку (рис. 4.3) зображено сторінку детальної інформації про товар на веб-сайті "For you". Зверху сторінки користувача зустрічають п'ять вкладинок з різними товарами, кнопка пошуку та кошик. Ці елементи залишаються статичними на всій сторінці, що дозволяє користувачу зручно орієнтуватися та повертатися до них незалежно від прокрутки.

Знизу, ліворуч розташований колаж з чотирьох фотографій товару, що дозволяє покупцеві роздивитися товар з різних ракурсів та побачити його деталі. Справа від фотографій розміщено назву товару, а під нею вказана ціна. Далі слідує "таблиця розмірів", яка надає користувачу інформацію про різні розміри товару. Після таблиці розмірів розташовано вибір розміру для покупки, де користувач може обрати підходящий розмір товару, а поряд з цим полем розташована кнопка "Купити". Ця кнопка надає зручний спосіб оформлення замовлення безпосередньо на сторінці товару.

Під кнопкою "Купити" розташований детальний опис товару, який містить додаткову інформацію про його характеристики, матеріали, колір або будь-які інші важливі деталі. Цей опис допомагає покупцеві краще зрозуміти товар перед прийняттям рішення про покупку.

Такий розташунок елементів на сторінці детальної інформації про

товар дозволяє користувачу легко переглядати фотографії товару, знаходити інформацію про його розміри, вибрати потрібний розмір та здійснювати покупку зручним способом. Детальний опис товару надає додаткову інформацію, що сприяє зробленню обґрунтованого вибору покупцем.

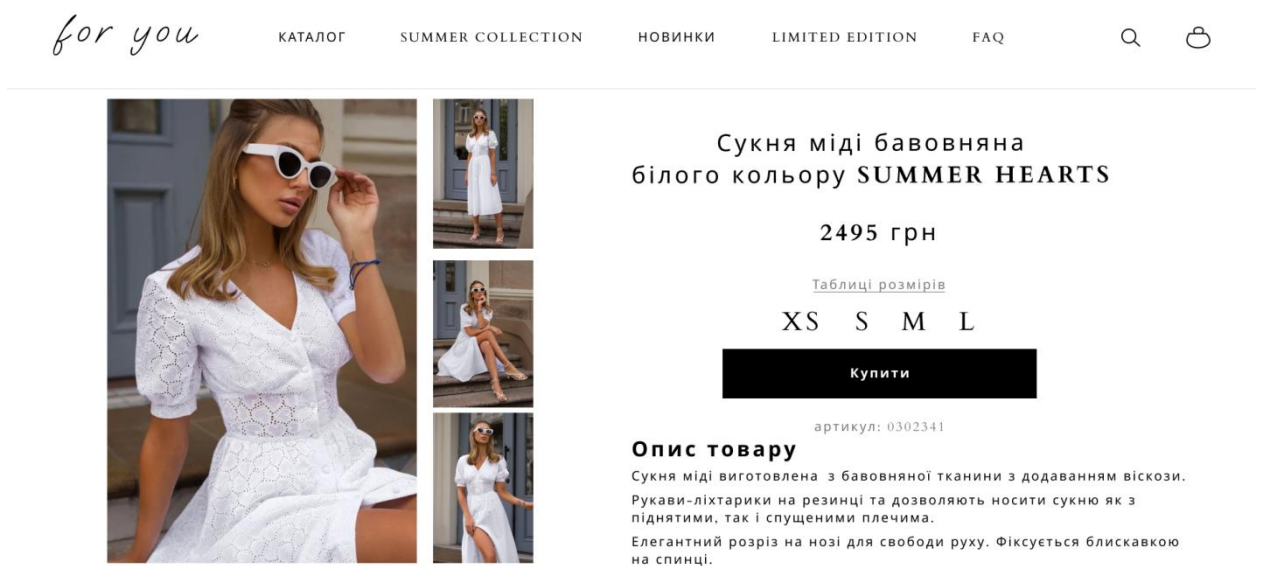


Рис. 4.3 – Детальна інформація про товар

#### 4.4 Замовлення товару

На рисунку (рис. 4.1) зображено сторінку детальної інформації про товар на веб-сайті "For you". Зверху сторінки користувача зустрічають п'ять вкладинок з різними товарами, кнопка пошуку та кошик. Ці елементи залишаються статичними на всій сторінці, що дозволяє користувачу зручно орієнтуватися та повертатися до них незалежно від прокрутки.

Знизу, ліворуч розташований колаж з чотирьох фотографій товару, що дозволяє покупцеві роздивитися товар з різних ракурсів та побачити його деталі. Справа від фотографій розміщено назву товару, а під нею вказана ціна. Далі слідує "таблиця розмірів", яка надає користувачу інформацію про різні розміри товару. Після таблиці розмірів розташовано вибір розміру для покупки, де користувач може обрати підходящий розмір товару, а поряд з цим полем розташована кнопка "Купити". Ця кнопка надає зручний спосіб оформлення замовлення безпосередньо на сторінці товару.

Під кнопкою "Купити" розташований детальний опис товару, який



містить додаткову інформацію про його характеристики, матеріали, колір або будь-які інші важливі деталі. Цей опис допомагає покупцеві краще зрозуміти товар перед прийняттям рішення про покупку.

Такий розташунок елементів на сторінці детальної інформації про товар дозволяє користувачу легко переглядати фотографії товару, знаходити інформацію про його розміри, вибирати потрібний розмір та здійснювати покупку зручним способом. Детальний опис товару надає додаткову інформацію, що сприяє зробленню обґрунтованого вибору покупцем.

The screenshot shows a checkout page for the brand 'for you'. The page is divided into two main sections: 'ОФОРМЛЕННЯ ЗАМОВЛЕННЯ' (Checkout) and 'КОШИК' (Cart).

**ОФОРМЛЕННЯ ЗАМОВЛЕННЯ**

**Дані отримувача**

Ім'я:  Прізвище:

тел.:  email:

Місто:


Номер відділення НП:

Коментар до замовлення:

Карта Visa і MasterCard (LiqPay)

Оплата при отриманні

**КОШИК**



Сукня міді бавовняна білого кольору SUMMER HEARTS

Розмір: S

Артикул: 0302341

Не телефонувати мені

2495 грн

Рис. 4.4 – Замовлення товару

## ВИСНОВОК

У рамках дипломної роботи був розроблений веб-сайт інтернет-магазину для малого бізнесу з продажу одягу. Веб-сайт був реалізований з використанням HTML, CSS та Vue.js, а також була підключена платіжна система LiqPay для забезпечення безпечних та зручних онлайн-платежів.

Під час процесу розробки веб-сайту були враховані ключові аспекти, які впливають на успішну роботу інтернет-магазину. Були досліджені існуючі веб-сайти з продажу одягу, аналізовані їх переваги та недоліки, що дало змогу розробити модель сайту, відповідну потребам та очікуванням цільової аудиторії.

Розроблена архітектура back-end забезпечує ефективну обробку запитів користувачів, збереження та керування даними, а також безпеку та захист інформації, включаючи використання платіжної системи LiqPay для безпечних онлайн-транзакцій.

Проведена апробація сайту дозволила перевірити його функціональність, сумісність з різними браузерами та пристроями, а також впевнитися в його належній роботі перед запуском.

В цілому, розроблений веб-сайт інтернет-магазину з продажу одягу має потенціал стати ефективним інструментом для малого бізнесу, дозволяючи залучати клієнтів, проводити онлайн-продажі та забезпечувати зручний та безпечний досвід покупок.

Завдяки використанню сучасних технологій розробки веб-сайту та підключенню платіжної системи LiqPay, можливо забезпечити високий рівень безпеки, швидкості та зручності для користувачів, що сприятиме позитивному враженню від взаємодії з інтернет-магазином.

Ураховуючи всі аспекти розробки та апробації сайту, можна зробити висновок, що веб-сайт інтернет-магазину для малого бізнесу з продажу одягу є готовим до експлуатації та може стати ефективним інструментом для розвитку бізнесу, забезпечуючи задоволення потреб клієнтів та досягнення поставлених цілей.

Після ретельного розгляду всіх аспектів розробки та тестування сайту, можна зробити висновок, що веб-сайт інтернет-магазину для малого бізнесу з продажу одягу є готовим до використання та може стати ефективним інструментом для розвитку бізнесу. Цей сайт готовий задовольняти потреби клієнтів та допомагати досягати поставлених цілей.

## СПИСОК ЛІТЕРАТУРИ

1. Package Search Official Website IntelliJ idea— 2023 — URL:  
<https://www.jetbrains.com/ru-ru/idea/>
2. Package Search Official Website Vue.js— 2023 — URL: <https://vuejs.org/>
3. Package Search Official Website Vuex — 2023 — URL:  
<https://vuex.vuejs.org/>
4. Official Website 19store — 2023 — URL: <https://19store.com.ua/clothes/tops>
5. Official Website papayashop — 2023 — URL: <https://papayashop.com.ua/>
6. Official Website cher17— 2023 — URL: <https://cher17.com/>
7. Official Website solmar — 2023 — URL:  
[https://solmar.com.ua/ua/?gclid=Cj0KCQjw1rqkBhCTARIsAAHz7K3i8jOAVvCNUhVIILDctDQnnoOnyASkmg\\_1687AybNgQJ3j9-CloesaAj\\_CEALw\\_wcB](https://solmar.com.ua/ua/?gclid=Cj0KCQjw1rqkBhCTARIsAAHz7K3i8jOAVvCNUhVIILDctDQnnoOnyASkmg_1687AybNgQJ3j9-CloesaAj_CEALw_wcB)
8. Designing Websites That Sell: Principles of Successful Internet Projects.  
Michael Landin.2011.c.76— 145.
9. Web Analytics 2.0: Analyzing Website Data That Drives Revenue" (author:  
Avinash Kaushik,2014.c.85 — 124.

## ДОДАТОК А

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel="icon" href="<%= BASE_URL %>favicon.ico">
    <title>tutorial_online_shop</title>
  </head>
  <body>
    <noscript>
      <strong>We're sorry but tutorial_online_shop doesn't work properly without
JavaScript enabled. Please enable it to continue.</strong>
    </noscript>
    <div id="app"></div>
    <!-- built files will be auto injected -->
  </body>
</html>

<template>
  <div class="v-main-wrapper">
    <v-header />
    <keep-alive>
      <router-view></router-view>
    </keep-alive>
  </div>
</template>

<script>
import vHeader from './layouts/v-header'

export default {
  name: 'v-main-wrapper',
  components: {
    vHeader
  },
  props: {},
  data() {
    return {
      title: 'Main wrapper'
    }
  },
}

```

```

    computed: {},
    methods: {},
    watch: {}
  }
</script>

<style>
.v-main-wrapper {
  margin: 0 auto;
}
</style>

<template>
<div class='v-select'>
  <p
    class="title"
    @click="areOptionsVisible = !areOptionsVisible"
  >{{ selected }}</p>
  <div
    class="options"
    v-if="areOptionsVisible || isExpanded"
  >
    <p
      v-for="option in options"
      :key="option.value"
      @click="selectOption(option)"
    >
      {{ option.name }}
    </p>
  </div>
</div>
</template>

<script>
export default {
  name: "v-select",
  props: {
    options: {
      type: Array,
      default() {
        return []
      }
    },
  },
  selected: {

```

```

    type: String,
    default: "
  },
  isExpanded: {
    type: Boolean,
    default: false
  }
},
data() {
  return {
    areOptionsVisible: false
  }
},
methods: {
  selectOption(option) {
    this.$emit('select', option)
    this.areOptionsVisible = false;
  },
  hideSelect() {
    this.areOptionsVisible = false;
  }
},
mounted() {
  document.addEventListener('click', this.hideSelect.bind(this), true)
},
beforeDestroy() {
  document.removeEventListener('click', this.hideSelect)
}
}
</script>

<style>
.v-select {
  position: relative;
  width: 200px;
  cursor: pointer;
  text-align: left;
}
.title {
  border: solid 1px #aeaeae;
  padding: 8px;
}
.v-select p {

```

```

    margin: 0;
  }

.options {
  border: solid 1px #aeaeae;
  background: #ffffff;
  position: absolute;
  top: 30px;
  left: 0;
  width: 100%;
  padding: 8px;
}

.options p:hover {
  background: #e7e7e7;
}
</style>

<template>
  <div class='v-notification'>
    <transition-group
      name="v-transition-animate"
      class="messages_list"
    >
      <div
        class="v-notification__content"
        v-for="message in messages"
        :key="message.id"
        :class="message.icon"
      >
        <div class="content__text">
          <span>{{ message.name }}</span>
          <i class="material-icons">{{ message.icon }}</i>
        </div>
        <div class="content_buttons">
          <button v-if="rightButton.length">{{ rightButton }}</button>
          <button v-if="leftButton.length">{{ leftButton }}</button>
        </div>
      </div>
    </transition-group>
  </div>
</template>

<script>

```



```
export default {
  name: "v-notification",
  props: {
    messages: {
      type: Array,
      default: () => {
        return []
      }
    },
    rightButton: {
      type: String,
      default: ""
    },
    leftButton: {
      type: String,
      default: ""
    },
    timeout: {
      type: Number,
      default: 3000
    }
  },
  data() {
    return {}
  },
  methods: {
    hideNotification() {
      let vm = this;
      if (this.messages.length) {
        setTimeout(function () {
          vm.messages.splice(vm.messages.length - 1, 1)
        }, vm.timeout)
      }
    }
  },
  watch: {
    messages() {
      this.hideNotification()
    }
  },
  mounted() {
    this.hideNotification()
  }
}
```

```
</script>

<style lang="scss">
.v-notification {
  position: fixed;
  top: 80px;
  right: 16px;
  z-index: 10;
  &__messages_list {
    display: flex;
    flex-direction: column-reverse;
  }
  &__content {
    padding: 16px;
    border-radius: 4px;
    color: #ffffff;
    display: flex;
    justify-content: space-between;
    align-items: center;
    height: 50px;
    margin-bottom: 16px;
    background: green;
    &.error {
      background: red;
    }
    &.warning {
      background: orange;
    }
    &.check_circle {
      background: green;
    }
  }
}
.content {
  &__text {
    display: flex;
    align-items: center;
    justify-content: space-between;
  }
}
.material-icons {
  margin-left: 16px;
}
}
```

```

.v-transition-animate {
  &-enter {
    transform: translateX(120px);
    opacity: 0;
  }
  &-enter-active {
    transition: all .6s ease;
  }
  &-enter-to {
    opacity: 1;
  }

  &-leave {
    height: 50px;
    opacity: 1;
  }
  &-leave-active {
    transition: transform .6s ease, opacity .6s, height .6s .2s;
  }
  &-leave-to {
    height: 0;
    transform: translateX(120px);
    opacity: 0;
  }

  &-move {
    transition: transform .6s ease;
  }
}
</style>

<template>
<div class='v-header'>
  <router-link :to="{ name: 'mainPage'}">
    
  </router-link>
  <div class="search-field">
    <input
      type="text"
      v-model="searchValue"
    >
    <button class="search_btn">
      <i class="material-icons" @click="search(searchValue)">search</i>

```

```

    </button>
    <button class="search_btn">
      <i class="material-icons" @click="clearSearchField">cancel</i>
    </button>
  </div>
</div>
</template>

<script>
import { mapActions, mapGetters } from 'vuex'

export default {
  name: "v-header",
  props: {},
  data() {
    return {
      searchValue: ""
    }
  },
  computed: {
    ...mapGetters([
      'SEARCH_VALUE'
    ])
  },
  methods: {
    ...mapActions([
      'GET_SEARCH_VALUE_TO_VUEX'
    ]),
    search(value) {
      this.GET_SEARCH_VALUE_TO_VUEX(value);
      if (this.$route.path !== '/catalog') {
        this.$router.push('/catalog')
      }
    },
    clearSearchField() {
      this.searchValue = ""
      this.GET_SEARCH_VALUE_TO_VUEX();
      if (this.$route.path !== '/catalog') {
        this.$router.push('/catalog')
      }
    }
  }
}
</script>

```

```

<style lang="scss">
.v-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  background: $green-bg;
  padding: 16px;
  position: fixed;
  width: 100%;
  top: 0;
  left: 0;

  img {
    width: 50px;
  }

  .search-field {
    padding: 16px;
    position: relative;
    right: 200px;
    display: flex;
    justify-content: center;
    align-items: center;
  }

  .search_btn {
    margin-left: 16px;
    background: transparent;
    border: none;
  }
}
</style>

<template>
<div class='v-catalog'>

  <v-notification
    :messages="messages"
  />

  <router-link :to="{name: 'cart', params: { cart_data: CART }}">
    <div class="v-catalog__link_to_cart">Cart: {{ CART.length }}</div>

```

```

</router-link>
<h1>Catalog</h1>
<div class="filters">
  <v-select
    :selected="selected"
    :options="categories"
    @select="sortByCategories"
  />
  <div class="range-slider">
    <input
      type="range"
      min="0"
      max="1000"
      step="10"
      v-model.number="minPrice"
      @change="setRangeSlider"
    >
    <input
      type="range"
      min="0"
      max="10000"
      step="10"
      v-model.number="maxPrice"
      @change="setRangeSlider"
    >
  </div>
  <div class="range-values">
    <p>Min: {{minPrice}}</p>
    <p>Max: {{maxPrice}}</p>
  </div>
</div>
<div class="v-catalog__list">
  <v-catalog-item
    v-for="product in filteredProducts"
    :key="product.article"
    :product_data="product"
    @addToCart="addToCart"
    @productClick="productClick"
  />
</div>
</div>
</template>

<script>

```

```

import vCatalogItem from './v-catalog-item'
import {mapActions, mapGetters} from 'vuex'
import vSelect from './v-select'
import vNotification from '../notifications/v-notification'

export default {
  name: "v-catalog",
  components: {
    vCatalogItem,
    vSelect,
    vNotification
  },
  props: {},
  data() {
    return {
      categories: [
        {name: 'Все', value: 'ALL'},
        {name: 'Мужские', value: 'м'},
        {name: 'Женские', value: 'ж'},
      ],
      selected: 'Все',
      sortedProducts: [],
      minPrice: 0,
      maxPrice: 10000,
      messages: []
    }
  },
  computed: {
    ...mapGetters([
      'PRODUCTS',
      'CART',
      'IS_MOBILE',
      'IS_DESKTOP',
      'SEARCH_VALUE'
    ]),
    filteredProducts() {
      if (this.sortedProducts.length) {
        return this.sortedProducts
      } else {
        return this.PRODUCTS
      }
    }
  },
  methods: {

```

```

...mapActions([
  'GET_PRODUCTS_FROM_API',
  'ADD_TO_CART'
]),
productClick(article) {
  this.$router.push( {name: 'product', query: { 'product': article }})
},
setRangeSlider() {
  if (this.minPrice > this.maxPrice) {
    let tmp = this.maxPrice;
    this.maxPrice = this.minPrice;
    this.minPrice = tmp;
  }
  this.sortByCategories()
},
sortByCategories(category) {
  let vm = this;
  this.sortedProducts = [...this.PRODUCTS]
  this.sortedProducts = this.sortedProducts.filter(function (item) {
    return item.price >= vm.minPrice && item.price <= vm.maxPrice
  })
  if (category) {
    this.sortedProducts = this.sortedProducts.filter(function (e) {
      vm.selected === category.name;
      return e.category === category.name
    })
  }
},
addToCart(data) {
  this.ADD_TO_CART(data)
  .then(() => {
    let timeStamp = Date.now().toLocaleString();
    this.messages.unshift(
      {name: 'Товар добавлен в корзину', icon: 'check_circle', id: timeStamp}
    )
  })
},
sortProductsBySearchValue(value) {
  this.sortedProducts = [...this.PRODUCTS]
  if (value) {
    this.sortedProducts = this.sortedProducts.filter(function (item) {
      return item.name.toLowerCase().includes(value.toLowerCase())
    })
  } else {

```



```

        this.sortedProducts = this.PRODUCTS;
    }
}
},
watch: {
    SEARCH_VALUE() {
        this.sortProductsBySearchValue(this.SEARCH_VALUE);
    }
},
mounted() {
    this.GET_PRODUCTS_FROM_API()
        .then((response) => {
            if (response.data) {
                this.sortByCategories()
                this.sortProductsBySearchValue(this.SEARCH_VALUE)
            }
        })
}
}
</script>

```

```

<style lang="scss">
.v-catalog {
    &__list {
        display: flex;
        flex-wrap: wrap;
        justify-content: space-between;
        align-items: center;
    }

    &__link_to_cart {
        position: fixed;
        top: 80px;
        right: 10px;
        padding: $padding*2;
        border: solid 1px #aeaeae;
        background: #ffffff;
    }
}

.filters {
    display: flex;
    justify-content: space-between;
    align-items: center;

```

```

}

.range-slider {
  width: 200px;
  margin: auto 16px;
  text-align: center;
  position: relative;
}

.range-slider svg, .range-slider input[type=range] {
  position: absolute;
  left: 0;
  bottom: 0;
}

input[type=range]::-webkit-slider-thumb {
  z-index: 2;
  position: relative;
  top: 2px;
  margin-top: -7px;
}
</style>

<template>
<div class='v-catalog-item'>

  <v-popup
    v-if="isInfoPopupVisible"
    rightBtnTitle="Add to cart"
    :popupTitle="product_data.name"
    @closePopup="closeInfoPopup"
    @rightBtnAction="addToCart"
  >
    
    <div>
      <p class="v-catalog-item__name">{{ product_data.name }}</p>
      <p class="v-catalog-item__price">Price: {{ product_data.price | toFix |
formattedPrice }} P.</p>
      <p class="v-catalog-item__price">{{ product_data.category }}</p>
    </div>
  </v-popup>

```

```

    
    <p class="v-catalog-item__name">{{product_data.name}}</p>
    <p class="v-catalog-item__price">Price: {{product_data.price | toFix |
formattedPrice}}</p>
    <button
      class="v-catalog-item__show-info"
      @click="showPopupInfo"
    >
      Show info
    </button>
    <br>
    <button
      class="v-catalog-item__add_to_cart_btn btn"
      @click="addToCart"
    >Add to cart
    </button>
  </div>
</template>

```

```

<script>
import vPopup from '../popup/v-popup'
import toFix from '../filters/toFix'
import formattedPrice from "../filters/price-format";

export default {
  name: "v-catalog-item",
  components: {
    vPopup
  },
  props: {
    product_data: {
      type: Object,
      default() {
        return {}
      }
    }
  },
  data() {
    return {
      isInfoPopupVisible: false
    }
  },
  filters: {

```

```

    toFix,
    formattedPrice
  },
  computed: {},
  methods: {
    productClick() {
      this.$emit('productClick', this.product_data.article)
    },
    showPopupInfo() {
      this.isInfoPopupVisible = true;
    },
    closeInfoPopup() {
      this.isInfoPopupVisible = false;
    },
    addToCart() {
      this.$emit('addToCart', this.product_data);
    }
  },
  mounted() {
    this.$set(this.product_data, 'quantity', 1)
  }
}
</script>

```

```

<style lang="scss">
.v-catalog-item {
  flex-basis: 25%;
  box-shadow: 0 0 8px 0 #e0e0e0;
  padding: $padding*2;
  margin-bottom: $margin*2;

  &__image {
    width: 100px;
  }
}
</style>

```

```

<template>
<div class='v-product-page'>
  
  <p>Product name: {{product.name}}</p>
  <p>Article: {{product.article}}</p>

```

```

<p>Price: {{product.price | toFix | formattedPrice}}</p>
<button
  class="v-catalog-item__add_to_cart_btn btn"
  @click="addToCart"
>Add to cart
</button>
</div>
</template>

```

```

<script>
import {mapGetters, mapActions} from 'vuex'
import toFix from "../../filters/toFix";
import formattedPrice from "../../filters/price-format";

export default {
  name: "v-product-page",
  props: {},
  data() {
    return {}
  },
  filters: {
    formattedPrice,
    toFix
  },
  computed: {
    ...mapGetters([
      'PRODUCTS'
    ]),
    product() {
      let result = {}
      let vm = this;
      this.PRODUCTS.find(function (item) {
        if (item.article === vm.$route.query.product) {
          result = item;
        }
      })
      return result;
    }
  },
  methods: {
    ...mapActions([
      'GET_PRODUCTS_FROM_API',
      'ADD_TO_CART'
    ]),

```

```
    addToCart() {
      this.ADD_TO_CART(this.product)
    },
  },
  mounted() {
    if (!this.PRODUCTS.length) {
      this.GET_PRODUCTS_FROM_API()
    }
  }
}
</script>
```

```
<style scoped>
```

```
</style>
```