

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри, канд. техн. наук,
доцент _____ Є. О. Давиденко
«__»____2022 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

ВЕБЗАСТОСУНОК МАГАЗИНУ ОДЯГУ

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 408. 21810818

Студент

_____ О. М. Маркевич
«__»____2022 р.

Керівник завідувач кафедри ІПЗ, канд. техн. наук,
доцент

_____ Є. О. Давиденко
«__»____2022 р.

Консультант канд. техн. наук, доцент

_____ А. О. Алексєєва
«__»____2022 р.

Миколаїв 2022

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП.....	5
1 АНАЛІТИЧНИЙ ОГЛЯД ВЕБЗАСТОСУНКІВ МАГАЗИНУ ОДЯГУ. ОСНОВНІ ПРОБЛЕМИ ТА ЇХ ВИРІШЕННЯ.....	7
1.1 Огляд та аналіз існуючих вебзастосунків магазину одягу.....	7
1.2 Загальні відомості про створений модуль, висвітлення проблем та їх вирішення.....	14
1.3 Специфікації вимог до програмного забезпечення.....	15
Висновки до розділу 1	19
2 МОДЕЛЮВАННЯ ВЕБЗАСТОСУНКУ МАГАЗИНУ ОДЯГУ	20
2.1 Алгоритм роботи модулю.....	20
2.2 Робота вебзастосунку на прикладі UML діаграми.....	27
Висновки до розділу 2	33
3 ОГЛЯД ОСНОВНИХ МЕТОДІВ СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЙОГО РОЗРОБКА	34
3.1 Liquid.....	34
3.2 HTML	36
3.3 CSS	38
3.4 JavaScript та Vue	39
3.5 Файлова структура проєкту.....	41
Висновки до розділу 3	42
4 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	43
4.1 Огляд програмного забезпечення	43
4.2 Логіка об'єднання продуктів.....	59
Висновки до розділу 4	61
ВИСНОВКИ.....	62
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	63

ДОДАТОК А Ініціалізація Vue компоненту product	64
ДОДАТОК Б Допоміжні функції компонентів Vue.....	65
ДОДАТОК В Схема налаштувань секції header.....	67

ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ	– Програмне забезпечення
API	– Application Programming Interface
HTML	– Hyper Text Markup Language
CSS	– Cascading Style Sheets
JSON	– JavaScript Object Notation
UML	– Unified Modeling Language

ВСТУП

З появою Інтернет активно почали розвиватися веббраузери та вебтехнології програмування. Це дало змогу підприємцям, які вирішили зайнятися бізнесом, відкривати не традиційні торгові точки, а інтернет-магазини. Для такого вибору є кілька причин:

- 1) не потрібно сплачувати за оренду приміщення;
- 2) не потрібно сплачувати за комунальні послуги;
- 3) не потрібно наймати працівників для реалізації товару, забезпечувати їхню охорону праці та інше.

Таким чином можна зрозуміти, що витрати на утримання інтернет-магазину значно нижчі, ніж на відкриття торгові точки в будь-якому куточку населеного пункту, а прибуток за правильного розвитку та підходу він отримує більше.

Основні особливості інтернет-магазину, який буде створено в ході виконання бакалаврської кваліфікаційної роботи:

- 1) автоматизація продажу товарів;
- 2) зрозумілий та сучасний дизайн;
- 3) використання реактивної бібліотеки Vue та Vuex;
- 4) використання алгоритмів Shopify для рекомендацій щодо продуктів.

Тема роботи є *актуальною*, оскільки пандемія COVID-19 змусила багато людей залишатися вдома і не відвідувати багатолюдних місць, тому підприємці повинні були закрити фізичні магазини та перейти на онлайн-торгівлю.

Об'єкт кваліфікаційної роботи – процес онлайн-продажу товарів.

Предмет кваліфікаційної роботи – засоби створення інтернет-магазину на базі платформи електронної комерції Shopify.

Мета кваліфікаційної роботи – підвищення реалізації товарів в Інтернет шляхом створення автоматизованої системи продажу.

Для досягнення поставленої мети необхідно розв'язати наступні **завдання**:

– аналіз предметної області методологій розробки та тестування програмного забезпечення;

- розробка вимог до інтернет-магазину;
- розробка скелету або дизайну інтернет-магазину;
- розробка головного функціоналу та сторінок інтернет-магазину;
- розробка вторинних сторінок інтернет-магазину.

Для розв'язання поставлених завдань буде використані наступні технології:

- 1) Liquid;
- 2) HTML;
- 3) CSS/SCSS;
- 4) JavaScript.

Актуальність спеціального розділу про охорону праці зумовлена тим, що на сьогоднішній день широке використання та велике розповсюдження інформаційних технологій сприяло створенню та детальної розробки норм, а також вимог до використання комп'ютерної техніки на підприємствах. Всі представлені норми та вимоги регулюються законодавством України. Саме тому створення належних умов праці, виключення або мінімізування ризиків для життя та здоров'я робітників мають велике значення.

Метою спеціального розділу про охорону праці є дослідження питань охорони праці, які пов'язані з діяльністю робітника, розробника програмного забезпечення.

1 АНАЛІТИЧНИЙ ОГЛЯД ВЕБЗАСТОСУНКІВ МАГАЗИНУ ОДЯГУ

1.1 Огляд та аналіз існуючих вебзастосунків магазину одягу

На сьогоднішній день існує багато вебзастосунків, які відносяться до категорії інтернет-магазин одягу, де можна знайти та придбати будь-який одяг на свій смак не виходячи з дому. Майже кожний сучасний користувач Інтернету бачив або використовував магазин такої категорії, адже коли водиш запит у пошуковий рядок про одяг, одразу відображає подібні сайти. Інтернет-магазини з продажу одягу частіше за всього вміщують в собі інформацію про товари, рекламні банери акцій та інше. Але багато таких вебзастосунків мають простий, нудний та незрозумілий для користувачів дизайн, яким не тільки незрозуміло що можна зробити в тому або іншому випадку, але й мало інформують користувача.

Зараз створено дуже багато вебзастосунків, що мають великі обсяги продажу, такі як: Kasta, Arjen, Issaplus, Lamoda, Dressa тощо. В них міститься вся інформація про одяг, який можливо придбати. Також більшість мають систему знижок, інформацію про компанію, доставку, акції та інше. Великий плюс вищеперерахованих вебзастосунків, що вони мають достатньо зрозумілий, сучасний та приємний для користувача дизайн відносно інших конкурентів.

Великий мінус усіх сайтів, що вони мало інформують своїх покупців о своїй продукції, трендах. Якщо дивитись з точки зору власника інтернет-магазину, то також використовується мало методів для того, щоб зацікавити покупця в покупці або додатковій покупці будь-чого. З цього можна зробити висновок, що на даний момент не існує ідеально-спроєктованих вебзастосунків чи систем, які виконують базові задачі на всі сто відсотків.

Якщо у пошуку ввести: «купити одяг Україна», результат видасть наступний список сайтів:

– <https://issaplus.com/> – на сайті розміщені товари для продажу, рекламні банери з акціями, додаткова інформація про магазин (рис. 1.1). Також є можливість подивитись статті блогу (рис. 1.2) та карту сайта (рис. 1.3). На сторінці колекції є можливість відфільтрувати та відсортувати товари за потрібним

параметром (рис. 1.4).

– <https://arjen.com.ua/> – на сайті розміщено багато відео-контенту (рис. 1.5), товари для продажу, додаткова інформація про магазин, рекламні банери (рис. 1.6). Є можливість звернутись до підтримки сайту у реальному часі (рис. 1.7).

– <https://kasta.ua/> – на сайті розміщені товари для продажу, рекламні банери, додаткова інформація про магазин. Також є можливість подивитись додаткову інформацію про партнерів магазину (рис. 1.8), купити товар у кредит (рис. 1.9), отримати кешбек після покупки (рис. 1.10).

Такі сайти загального огляду мають свої недоліки та переваги. У відкритому доступі немає даних про кількість багів, відмов, швидкості обробки запиту та іншого. Тому висновки про переваги та недоліки даних систем можна робити лише з точки зору користувацького досвіду. Немає відчутної різниці у швидкості роботи – інтерфейс та функціонал цих систем майже однаковий. Потрібно відмітити, що з головних переваг сайтів варто виділити наступні: можливість оплати товару частинами, вільний доступ з будь-якого девайсу на якому є Інтернет, гарний та зручний інтерфейс. Критичних недоліків не було знайдено.

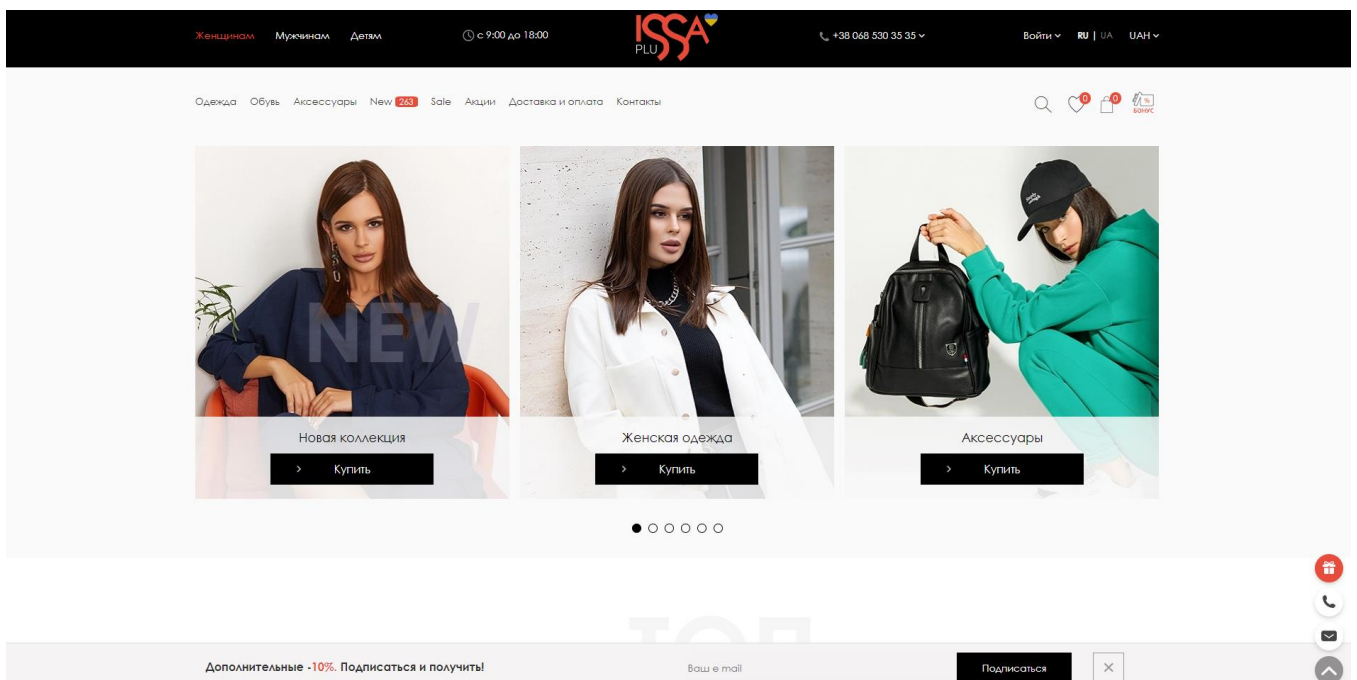


Рисунок 1.1 – Зовнішній вигляд головної сторінки сайту <https://issaplus.com/>

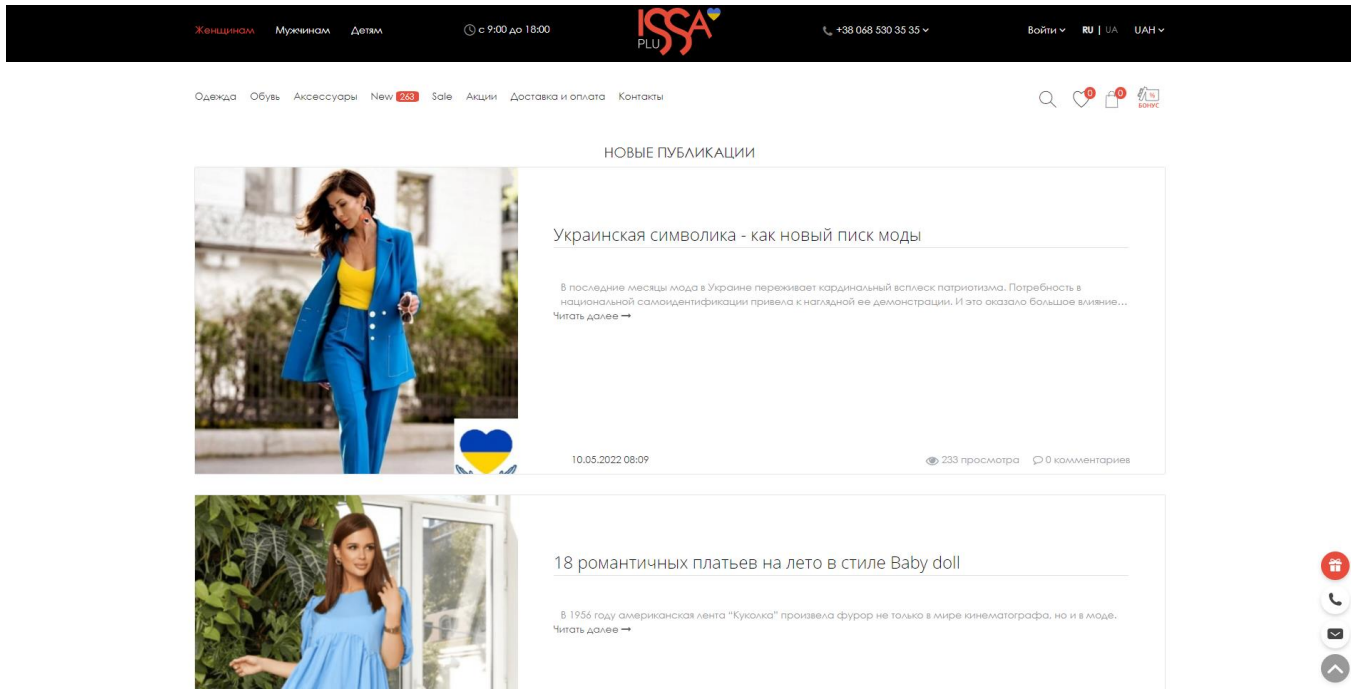


Рисунок 1.2 – Зовнішній вигляд сторінки блогу сайту <https://issaplus.com/>

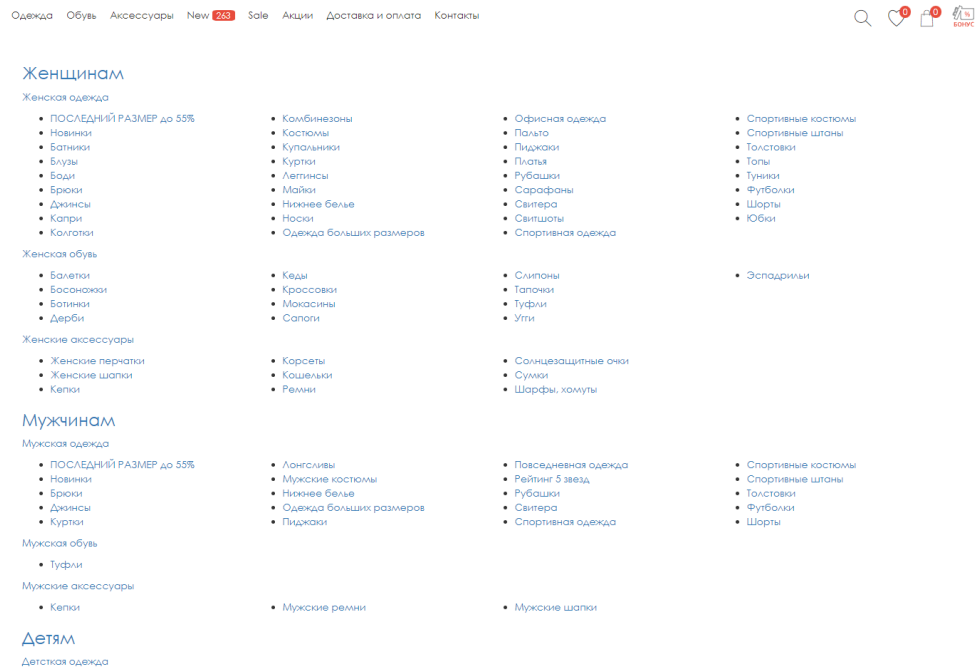


Рисунок 1.3 – Зовнішній вигляд карти сайту <https://issaplus.com/>

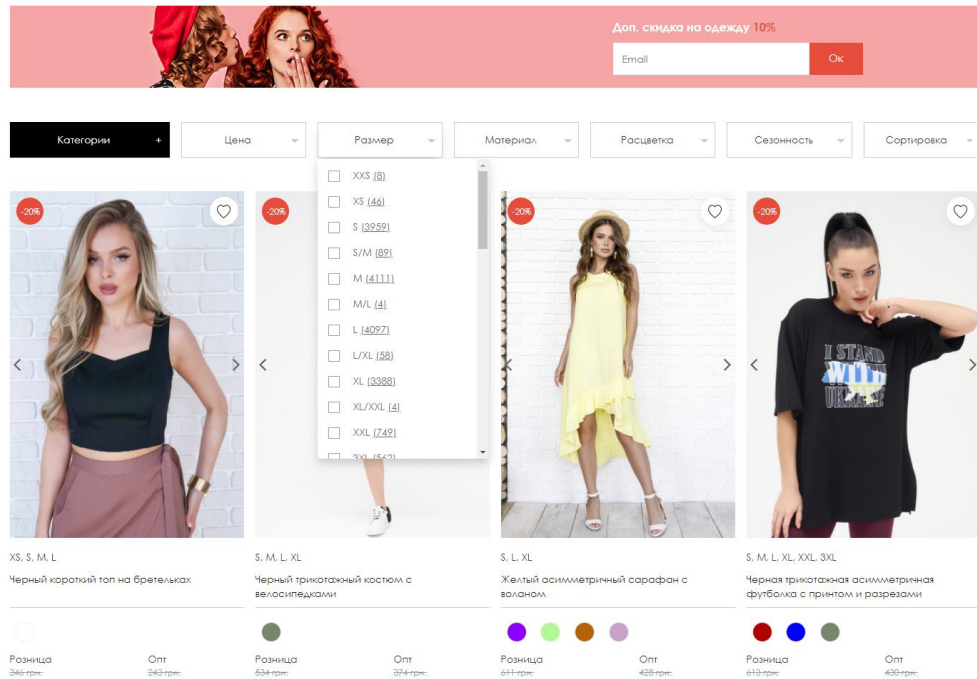


Рисунок 1.4 – Зовнішній вигляд сторінки колекції сайту <https://issaplus.com/>

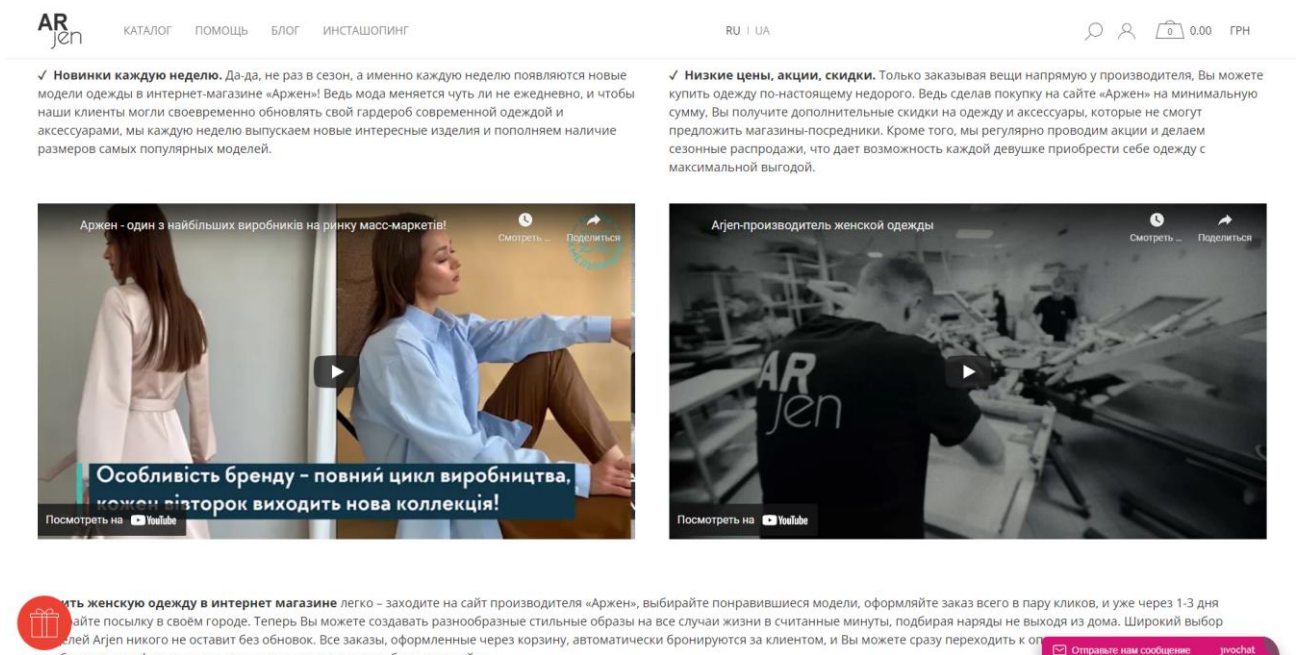


Рисунок 1.5 – Відео-контент сайту <https://arjen.com.ua/>

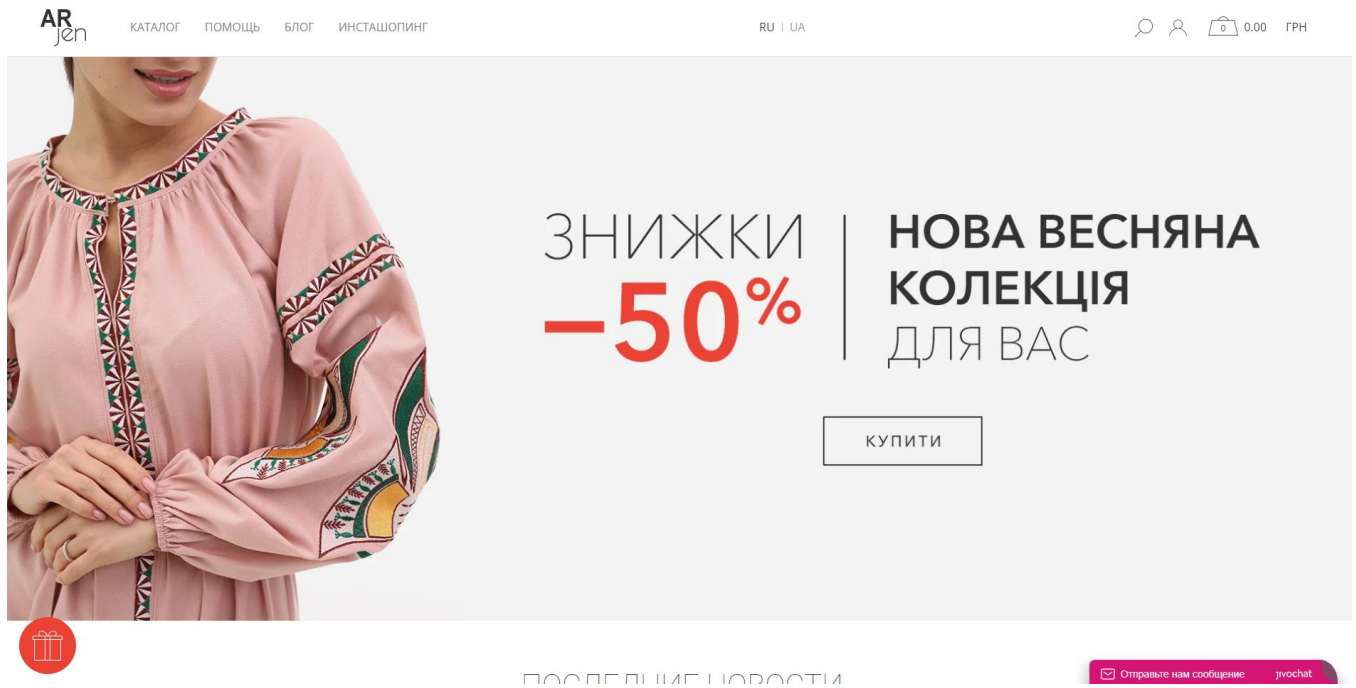


Рисунок 1.6 – Рекламний банер на головній сторінці сайту <https://arjen.com.ua/>

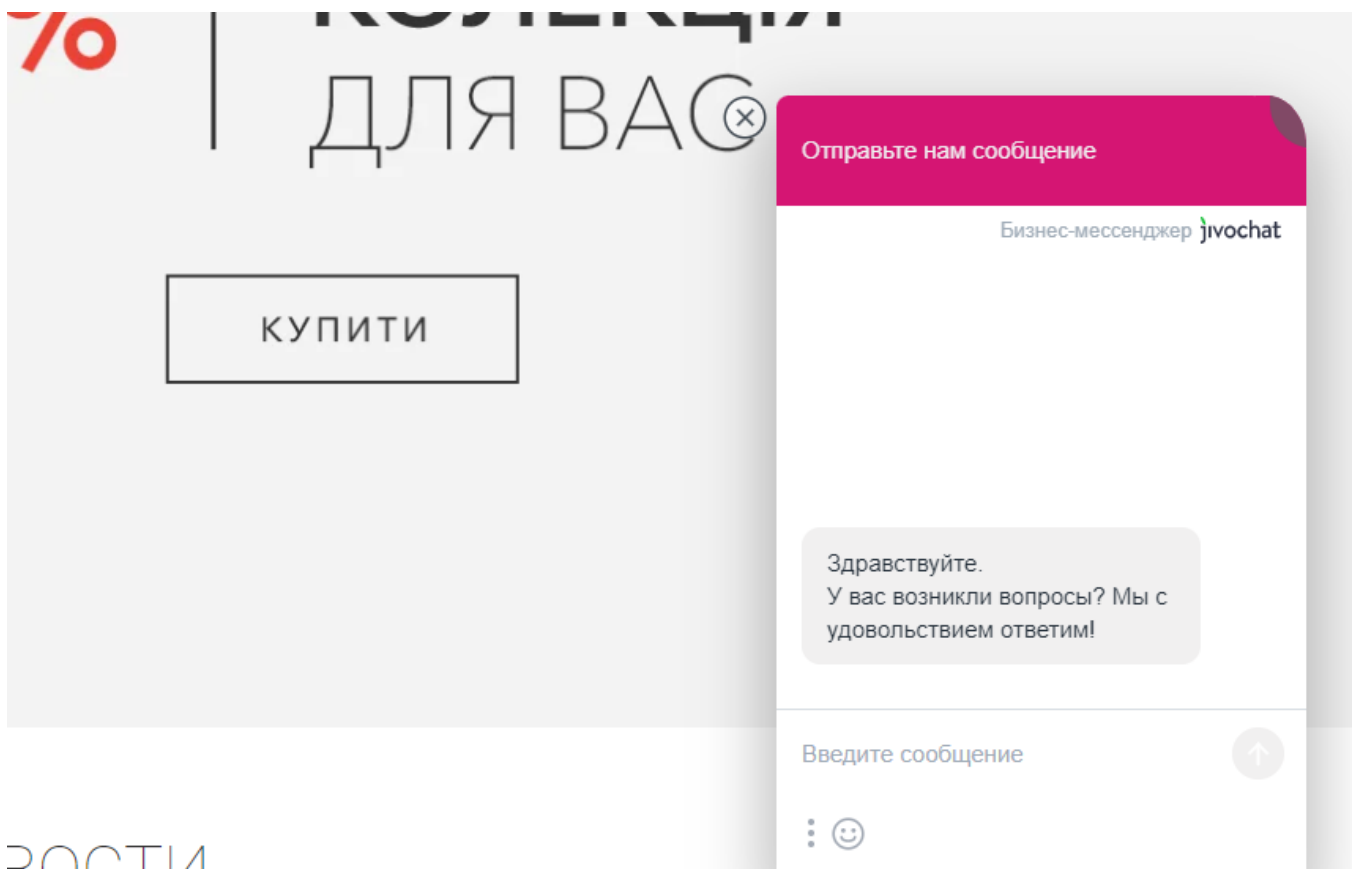


Рисунок 1.7 – Live-чат підтримки сайту <https://arjen.com.ua/>

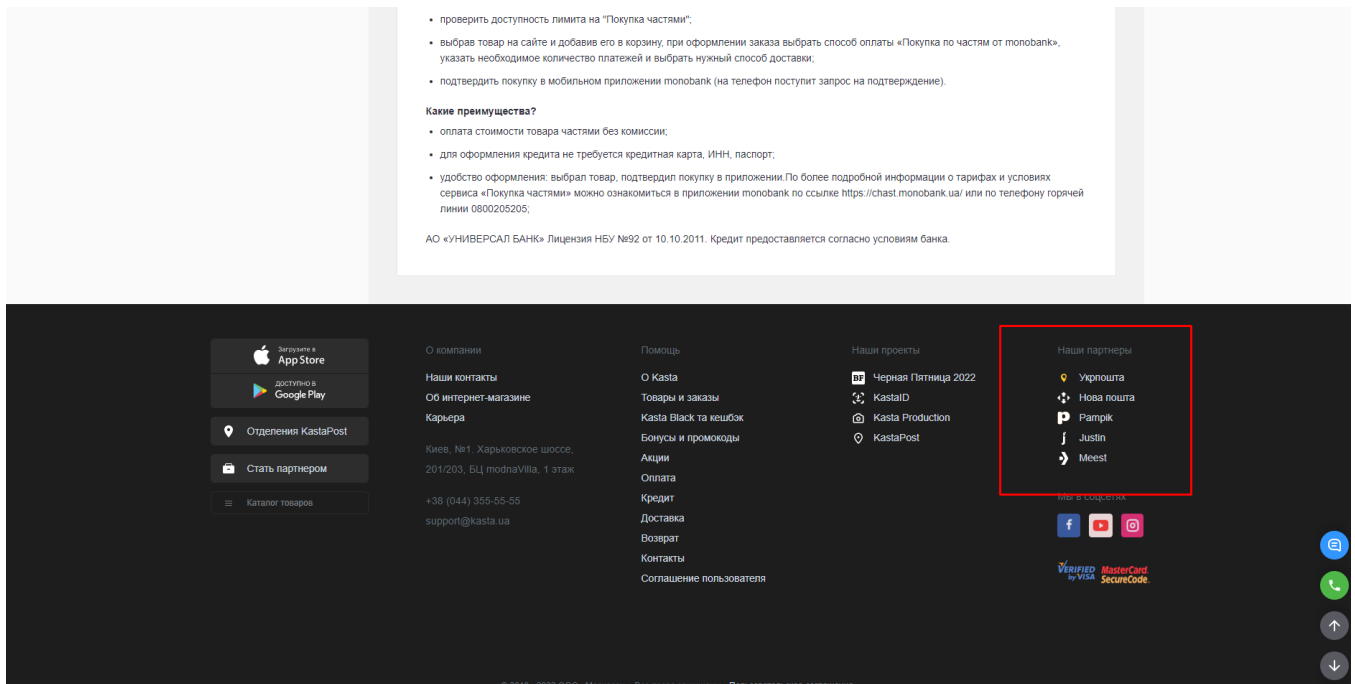


Рисунок 1.8 – Партнери сайту <https://kasta.ua/>

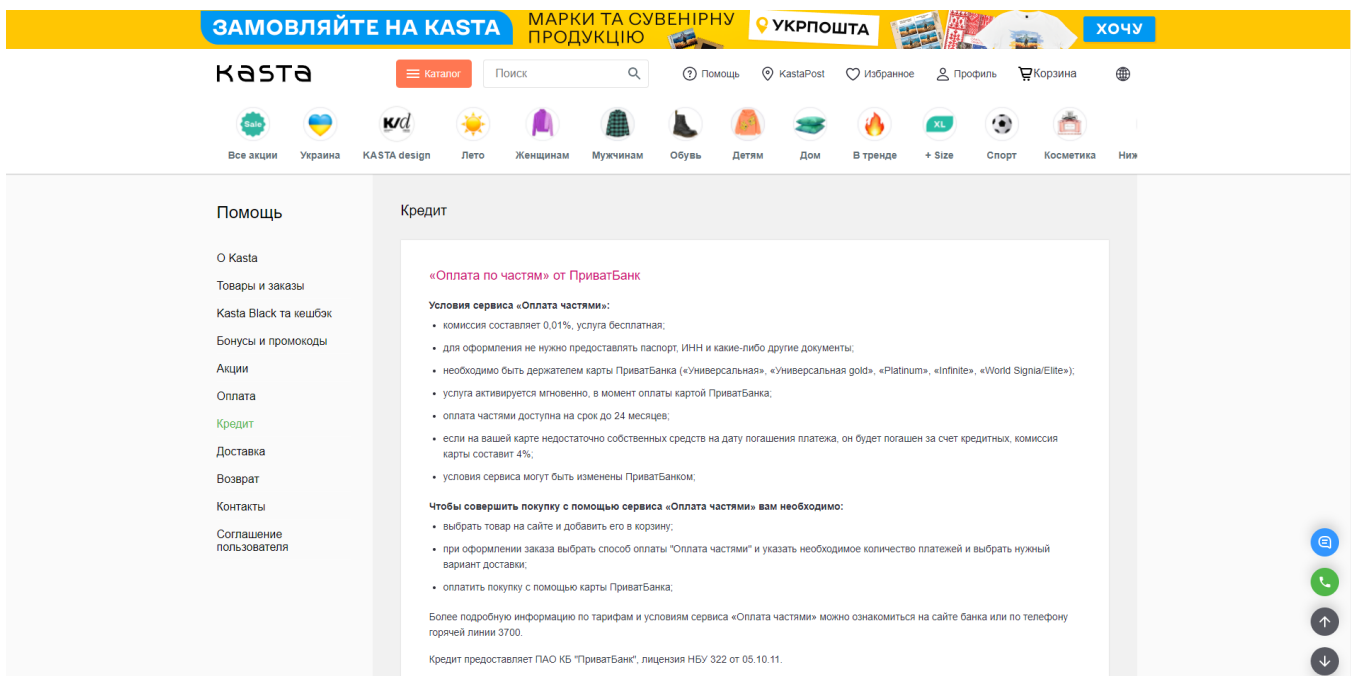


Рисунок 1.9 – Сторінка про купівлю товару у кредит на сайті <https://kasta.ua/>

Что такое Кешбэк

Кешбэк – это возврат части от стоимости покупки, оплаченной банковской картой на сайте, в виде скидок на Kasta или денежных средств.

[Полные правила программы Кешбэк](#)

Ставки Кешбэк

- 1% от суммы покупки – для всех покупателей;
- 3% от суммы покупки – если на момент оформления заказа у вас действует подписка Black;
- увеличенные ставки могут действовать в дни праздников от Kasta и указываются в карточке соответствующего товара;

Как получить Кешбэк

Программа распространяется на товар, у которого в карточке товара указана сумма кешбэка в гривнах.

Кешбэк начисляется при оплате товара банковской картой на сайте в момент оформления заказа.

Вы увидите сумму начисленного кешбэка в своем Профиле сразу после оплаты товара.

Начисленный кешбэк можно использовать через 15 дней после получения товара, за который он был начислен, для получения скидки при покупке товаров на Kasta или обменять на денежные средства.

В случае отказа от заказа или возврата товара кешбэк, начисленный за такую покупку, аннулируется.

Какой товар можно приобрести по Кешбэку

Кешбэк можно использовать при оплате любых товаров на сайте, кроме следующих:

- алкоголь;
- подписка на Black;
- услуги доставки;
- пакеты дополнительной гарантии и сервисного обслуживания;
- товары с государственными регулируемыми ценами;
- товар, у которого в карточке товара не указана возможность его использования;

Как использовать кешбэк при покупке

Если вы хотите оплатить товар кешбэком, просто включите отметку «Использовать кешбэк» в поле подтверждения заказа, прежде чем его оформить.

Размер кешбэка, потраченного на ваш заказ, будет определен следующим образом:

- к оплате товара автоматически применяется максимальное количество имеющегося у вас кешбэка, доступного к использованию, но не

Рисунок 1.10 – Сторінка з інформацією про кешбек на сайті <https://kasta.ua/>

Вище представлений основний список сайтів, які перші потрапляють на око користувачу, при введенні будь-якого запиту стосовно купівлі одягу в Україні. Представлені вище інтернет-магазини більш-менш схожі між собою за функціоналом та наявності тих або інших сторінок. Існує ще багата кількість інтернет-магазинів одягу і кожен наступний схожий з попереднім – цих трьох сайтів-прикладів буде достатньо для загального представлення функціоналу схожих сайтів.

ІНОВАЦІЇ

Функціонал магазинів, які були представлені вище, досить великий, але його можна поліпшити для збільшення продажів:

1. рекомендації товарів. Було б непогано рекомендувати продукти на основі вибраного, щоб зацікавити покупця в додатковій покупці, тим самим збільшивши середню вартість замовлення;

2. швидка покупка. Було б непогано реалізувати функцію швидкої покупки прямо зі сторінки колекції. Таким чином, покупцю не потрібно буде переходити на сторінку продукту, щоб додати товар у кошик;

3. блог та статті. Можна реалізувати сторінки блогу та статей, щоб детальніше знайомити покупця зі своїми продуктами та повідомляти про нові оновлення та поповнення колекцій.

1.2 Загальні відомості про створений модуль, висвітлення проблем та їх вирішення

У світі останніх подій в Україні, спочатку пандемія коронавірусу, а потім розгортання бойових дій на її території, змушують багатьох людей залишатися вдома і не виходити зайвий раз на вулицю. У зв'язку із цим фізичні магазини втратили свою актуальність. У такому разі найкраще відкрити інтернет-магазин, щоб продовжити свої продажі онлайн. Саме в інтернет-магазині одягу користувач зможе неквапливо купити потрібний товар.

Отже, для вирішення поставленої проблеми, нам необхідно реалізувати наступні функції:

- пошук інформації про товари;
- перегляд інформації про товари;
- порівняння товарів;
- занесення товарів у список обраного;
- занесення товарів у кошик;
- здійснення замовлення;
- наявність відеоогляду на товар;
- рекомендації щодо підбору аксесуарів до товару;
- написання відгуків;
- користування чатом на сайті;

- авторизація;
- реєстрація.

Вище представлений функціонал допоможе користувачу придбати потрібний йому одяг. Це дозволить користувачу зекономити час і не виходити зайвий раз на вулицю, ризикуючи своїм життям

1.3 Специфікації вимог до програмного забезпечення

ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

Призначення системи (застосунку), для якої розробляється програмне забезпечення

Дане програмне забезпечення створено для налагодження автоматизації продажу одягу.

Погодження, що ухвалені в програмній документації

Було погоджено, що для створення вебзастосунку та його злагодженої роботи буде використовуватися платформа електронної комерції Shopify. Деякі сторінки будуть використовувати Vue.js фреймворк для забезпечення більшої реактивності та відгуку на дії користувачів.

Межі проєкту ПЗ

Крайня дата завершення роботи над програмним забезпеченням – 10.06.2021р.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Представлений вебзастосунок не має обмежень у сферах його застосування.

Характеристики користувачів

Основні характеристики користувачів: наявність смартфона або комп'ютеру з доступом до Інтернет.

Загальна структура і склад системи

Основні частини для створення програмного забезпечення: вебзастосунок

Загальні обмеження

Єдине обмеження для роботи з програмним забезпеченням – це наявність

Інтернету

ФУНКЦІЇ ВЕБЗАСТОСУНКУ МАГАЗИНУ ОДЯГУ

- пошук інформації про товари;
- перегляд інформації про товари;
- порівняння товарів;
- занесення товарів у список обраного;
- занесення товарів у кошик;
- здійснення замовлення;
- наявність відеоогляду на товар;
- рекомендації щодо підбору аксесуарів до товару;
- написання відгуків;
- користування чатом на сайті;
- авторизація;
- реєстрація.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Особливих технічних вимог немає. Потрібен більш-менш справний комп'ютер або ноутбук з функціонуючим процесором та з достатньою оперативною пам'яттю, наприклад INTEL Core i5 7300, цей процесор підтримує необхідні для розробки програми (IDE PhpStorm).

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Системне програмне забезпечення

Для написання фронтенду було використано мову гіпертексту та розмітки HTML, мова шаблонів Liquid, каскадні таблиці стилів CSS та Javascript. Деякі сторінки було розроблено за допомогою JavaScript-фреймворку Vue.js з використанням Vuex. Розробка бекенду не потрібна так, як буде використана платформа Shopify. Вона має власний бекенд але доступ до нього отримати неможливо – можливо лише використовувати API-інтерфейси, які можуть отримувати доступ до продуктів, замовленням, клієнтам и т.д.

Мережне програмне забезпечення

Для створення ПЗ знадобиться будь-яка операційна система, HTML-

редактори: Visual Code, IDE PhpStorm та Node.js; будь-який браузер для зручного перегляду сторінок: Chrome, Opera, Firefox, Safari та інші.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Інтерфейс повинен задовольняти усім вимогам UX/UI, що дозволить користувачу затратити найменше часу на розуміння роботи системи. Шаблон вебзастосунку повинен мати багато сторінок (homepage, product page, collection page, article page, blog page, contact us page, etc).

Апаратний інтерфейс

Апаратний інтерфейс – це девайс користувача, яким він буде користуватися (смартфон, ноутбук, персональний комп'ютер)

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

Будь-який користувач, який бажає скористатися інтернет-магазином, може знайти його за допомогою браузера та пошукового ресурсу.

Переносимість

Програмне забезпечення може працювати на будь-якій операційній системі та будь-якому смартфоні.

Продуктивність

Продуктивність залежить від швидкості мережі Інтернет та від кількості контенту сторінки. Продуктивність застосунка повинна контролюватися часом виконання запитів, що не повинні перевищувати 3-4 секунд.

СЦЕНАРІЙ ВИКОРИСТАННЯ

Таблиця 1.1 – Сценарій використання (авторизація)

Діючі особи	Покупець, система
Мета	Переглянути всю доступну інформацію про певний товар.

Кінець таблиці 1.1

Успішний сценарій:	
1. Покупець переходить до сторінки з товаром.	
2. Обирає необхідні параметри товару.	
3. Система зберігає ці налаштування.	
4. Покупець дивиться інформацію в певних розділах (головне, характеристики, опис, відгуки та питання, аксесуари, наявність в магазинах).	
Результат	Обрано товар.

Таблиця 1.2 – Сценарій використання (реєстрація)

Діючі особи	Покупець, система
Мета	Створити обліковий запис.
Передумова	Покупець не авторизований.
Успішний сценарій:	
1. Покупець переходить до форми реєстрації.	
2. Заповнює контактні дані.	
3. Система здійснює валідацію даних.	
4. Система зберігає обліковий запис.	
5. Покупець отримує повідомлення про успішну реєстрацію.	
Результат	Збережено новий обліковий запис.

Таблиця 1.3 – Сценарій використання (здійснення покупки)

Діючі особи	Покупець, система
Мета	Оформити замовлення.
Передумова	К кошику є хоча б 1 товар.

Кінець таблиці 1.3

Успішний сценарій:	
1. Покупець натискає на кнопку “Оформити замовлення”.	
2. Система переводить покупця на форму з заповненням даних(адреса, піб, телефон та інше).	
3. Користувач вводить дані та відправляє їх.	
4. Система перевіряє коректність даних та приймає замовлення.	
Результат	Збережено новий обліковий запис.

ІНШІ ВИМОГИ

Усі вимоги описані вище.

Висновки до розділу 1

В першому розділі проведено огляд існуючих вебзастосунків магазину одягу. Розглянуті інтернет-магазини, які мають автоматизовану систему продажу товарів. Визначені основні переваги та недоліки оглянутих вебзастосунків. Визначено, що більшість являються комерційними розробками і недоступні ні у вигляді програмного коду, ні у вигляді формального опису алгоритмів. Сформульовані задачі досліджень дипломної роботи та основні вимоги щодо створення вебзастосунку.

2 МОДЕЛЮВАННЯ ВЕБЗАСТОСУНКУ МАГАЗИНУ ОДЯГУ

2.1 Алгоритм роботи сценаріїв

Вебзастосунок магазину одягу дуже простий та інтуїтивно зрозумілий для будь-якої людини. Даний вебзастосунок не матиме обмежень, він буде доступний для будь-якого користувача. Клієнт може користуватися ним через телефон, ноутбук чи персональний комп'ютер.

Вебзастосунок призначений для підвищення реалізації товарів в Інтернет. Основні функціональні можливості модулю:

- пошук інформації про товари;
- перегляд інформації про товари;
- занесення товарів у список обраного;
- занесення товарів у кошик;
- здійснення замовлення;
- наявність відеоогляду на товар;
- рекомендації щодо підбору аксесуарів до товару;
- написання відгуків;
- авторизація;
- реєстрація.

Блок-схема – це найбільш розповсюджений тип графічних моделей, що описують процеси або алгоритми, у яких різні кроки зображуються блоками різної форми та об'єднаних між собою лініями, які вказують напрям виконання. Наприклад, прямокутник – це відображення певної дії, ромб – умова або формула, паралелепіпед – введення або виведення даних та інші. Блок-схема допоможе схематично представити процес роботи модулю. Варто зауважити, що вебзастосунок безкоштовний, отже кожен бажаючий зможе зробити певні дії та отримати бажаний результат.

Алгоритм додавання товару до кошика та оформлення замовлення:

1. Переходимо на сторінку бажаного товару.

2. Обираємо потрібні нам опції кольору та розміру. Якщо даного варіанту з обраними опціями немає в наявності, кнопка “Add To Cart” стає неактивною і не дозволяє користувачу додати обраний варіант товару до кошику. В такому випадку потрібно вибрати інші опції, щоб отримати варіант товару який є в наявності – в такому випадку кнопка “Add To Cart” стає активною та дозволяє користувачу додати обраний варіант товару до кошику.

3. Додати товар у кошик, натиснувши кнопку “Add To Cart”.

4. Перейти у кошик. За бажанням є можливість залишити примітки до замовлення, після чого потрібно натиснути кнопку “Checkout”.

5. Після переходу до “Checkout” необхідно ввести контактні дані та інформацію про доставку. Також на цьому етапі є можливість ввести код для отримання знижки. Якщо введений код є недійсним, користувач отримає відповідне повідомлення про недійсність коду. В іншому випадку знижка буде застосована і ціна товару буде змінена. Якщо всі поля були заповнені, а інформація пройшла валідацію, користувача буде перенаправлено на етап оплати, в іншому випадку він отримає повідомлення про поля, які потрібно перезаповнити.



Рисунок 2.1 – Блок-схема алгоритму додавання товару до кошика та оформлення замовлення

Алгоритм створення облікового запису:

1. Переходимо на сторінку реєстрації.
2. Заповнюємо всі поля (ім'я, прізвище, e-mail, пароль).
3. Натискаємо кнопку "Create". Якщо всі поля заповнені вірно, на пошту користувача, яка була вказана, знаходить лист з активацією акаунту. Якщо хоча б одно з полів не заповнено, або заповнено некоректно, або існує акаунт з вказаною поштою, буде виведено відповідну помилку.

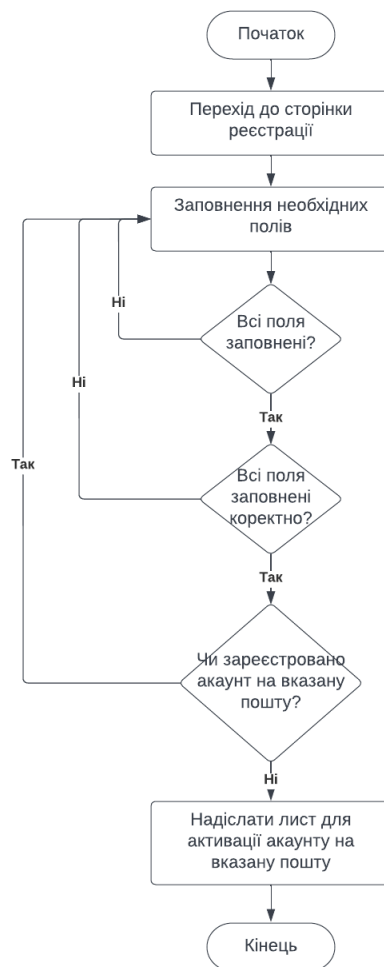


Рисунок 2.2 – Блок-схема алгоритму створення облікового запису

Алгоритм відновлення паролю акаунту:

1. Переходимо на сторінку для відновлення паролю.
2. Вказуємо пошту до якої прив'язано акаунт. Якщо поле заповнено некоректно або вказаної пошти немає в базі зареєстрованих користувачів, то виведеться відповідне повідомлення про помилку. Якщо вказана пошта існує у

базі зареєстрованих користувачів, тоді на пошту буде відправлено листа з посиланням на відновлення паролю.

3. Переходимо по посиланню у листі.
4. Вказуємо новий пароль та підтверджуємо його у відповідному полі.
5. Натискаємо кнопку “Reset password”. Якщо паролі в обох полях співпадають, то для цього акаунту буде змінено пароль. Якщо паролі в обох полях не будуть співпадати, тоді буде виведено відповідне повідомлення про помилку.

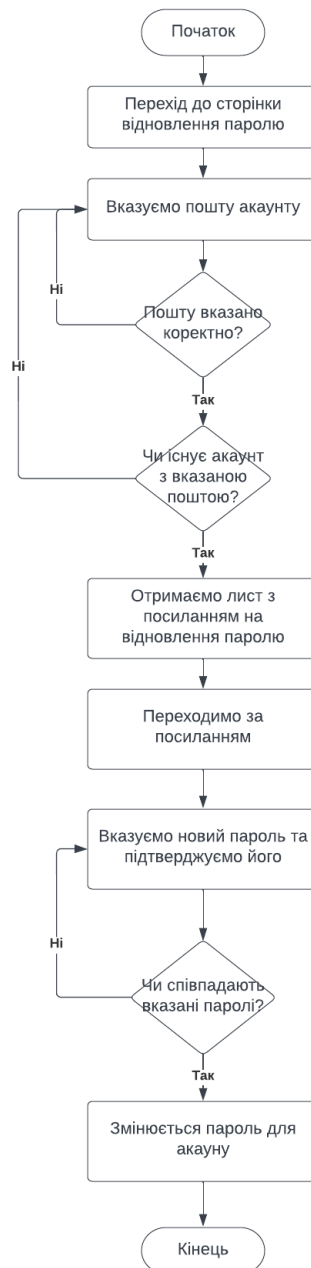


Рисунок 2.3 – Блок-схема алгоритму відновлення паролю для облікового запису

Алгоритм авторизації до акаунту:

1. Переходимо на сторінку авторизації.
2. Вказуємо пошту та пароль. Якщо всі поля заповнені, заповнені коректно, вказана пошта є у базі зареєстрованих користувачів та пароль до цього акаунту вказано вірно, тоді користувача буде авторизовано у систему. Інакше буде виведено відповідне повідомлення про помилку.

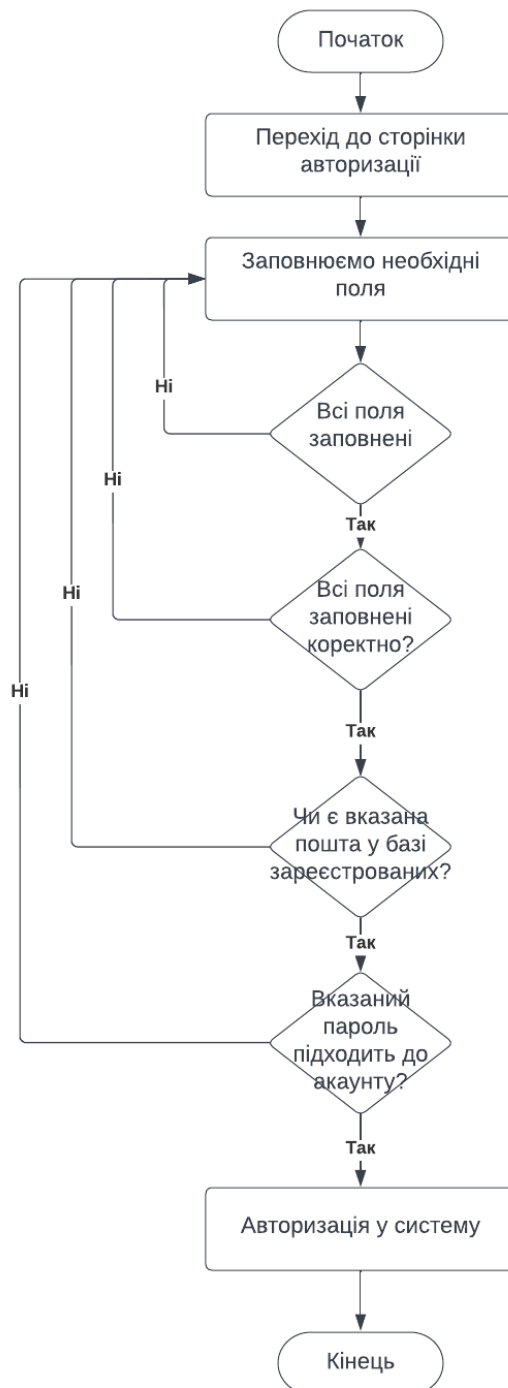


Рисунок 2.4 – Блок-схема алгоритму авторизації

Алгоритм пошуку товарів:

1. Натискаємо на значок пошуку.
2. Заповнюємо поле пошуку відповідним запитом. Якщо є товари, які підходять під запит, перші (за замовченням чотири) будуть відображені одразу (до натиснення кнопки пошуку).
3. Натискаємо на кнопку пошуку. Після пошуку будуть відображені товари, які підходять під запит. Якщо немає товарів, які підходять під запит, тоді буде виведено повідомлення про те, що товарів не було знайдено.

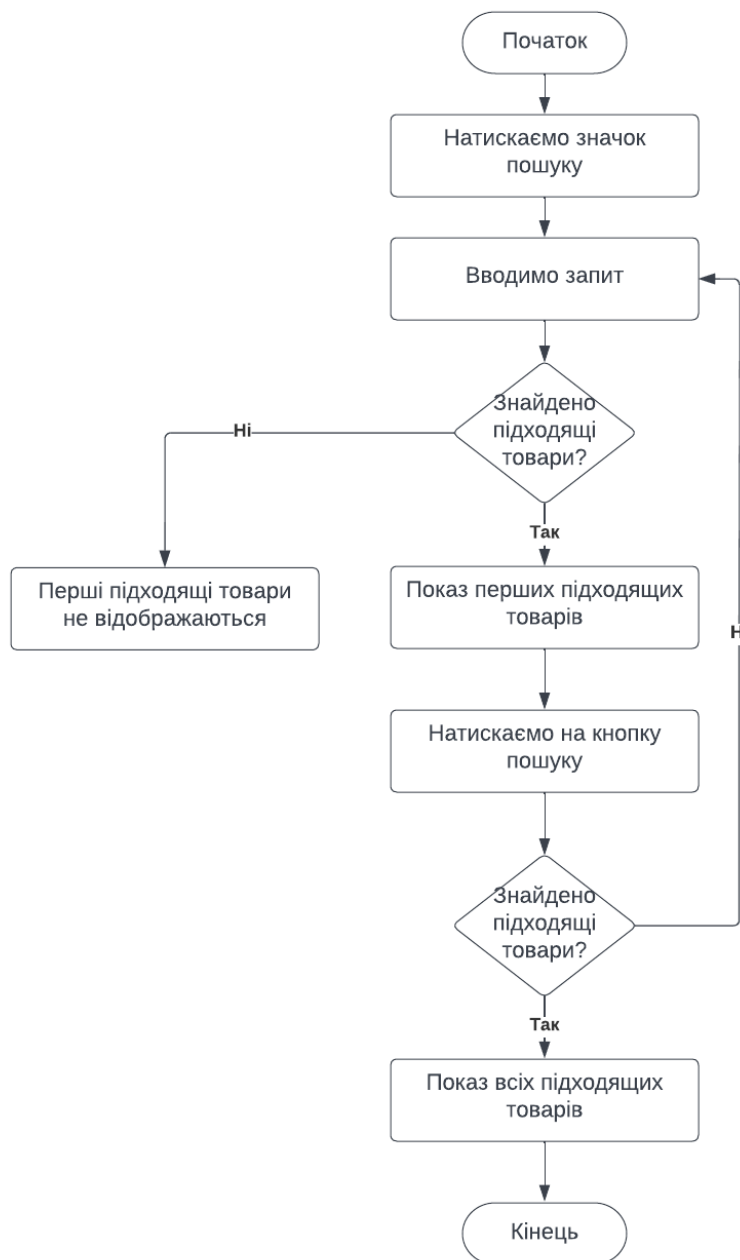


Рисунок 2.5 – Блок-схема алгоритму пошуку товарів

Розробка блок-схем зазвичай виконується, на етапі документування або розробки програмного забезпечення. Це робиться для уникнення великих ризиків, які пов'язані з основними помилками на етапі проектування та розробки програмного забезпечення. Іноді розробка програмного забезпечення не можлива без створення блок-схем, тому що існують візуальні мови програмування. Крім того блок-схеми використовуються для верифікації алгоритмів.

2.2 Робота вебзастосунок на прикладі UML діаграми

Unified Modeling Language (UML) – це уніфікована графічна мова для об'єктного моделювання в області розробки ПЗ. UML є мовою широкого профілю, вона використовується для системного проектування, відображення різних організаційних структур, а також для моделювання бізнес-процесів.

UML мова має власний синтаксис й правила оформлення моделей. За допомогою графічної нотації мови можна досконально візуалізувати систему, сприяти покращенню та уточненню моделі у процесі розробки, а також об'єднати всі компоненти в єдину структуру. Нотація UML містить основні типи елементів:

- написи;
- лінії;
- фігури;
- значки.

Відносини в UML – це посилання, яке описує, як дві або більше речей можуть ставитись один до одного під час виконання системи. Тобто відносини використовуються для уявлення зв'язку між структурними, групуючими або поведінковими речами. Існує декілька типів відношень, що показують взаємодію між актором та варіантами використання:

- Асоціації – це набір посилань, які пов'язують елементи моделі UML. Вони також визначають, скільки об'єктів беруть участь у відносинах.
- Залежності – демонструє як два або більше об'єктів залежать один від одного. При такому типі відносин, якщо ми вносимо зміни до одного об'єкта, ймовірно, що всі інші елементи також торкнуться цієї зміни.

- Узагальнення – переважно використовується для подання спадщини.
- Реалізація – цей зв'язок найчастіше зустрічається у разі інтерфейсів, коли одна сутність означає деяку відповідальність, яка не реалізована сама собою, а інша сутність реалізує її.

В мові UML є 12 типів діаграм [5]:

- 5 типів представляють поведінкові аспекти системи;
- 4 типи діаграм представляють статичну структуру додатку;
- 3 представляють фізичні аспекти функціонування системи (діаграми реалізації).

Деякі з видів діаграм специфічні для певної системи і додатку. Найбільш доступними з них є:

- Діаграма об'єктів (Object diagram) – демонструє частини об'єктів системи та зв'язки між ними;
- Діаграма класів (Class diagram) – демонструє логічну модель системи;
- Діаграма співробітництва (Collaboration diagram) – демонструє всі взаємозв'язки об'єктів;
- Діаграма активностей (Activity diagram) – демонструє бізнес-процеси об'єкта. Цей тип діаграм дозволяє показати послідовність, розгалуження та синхронізацію процесів.;
- Діаграма станів (Statechart diagram) – демонструє стан об'єктів системи;
- Діаграма послідовності (Sequence diagram) – демонструє послідовність передачу повідомлень між об'єктами, виділяючи послідовність прийому та передачі повідомлень;
- Діаграма розгортання (Deployment diagram) – демонструє декомпозицію системи на різних видах пристроїв;
- Діаграма прецедентів (Use-case diagram) – демонструє функціональне призначення системи, відображає об'єкти та завдання, які вони виконують.

Діаграма прецедентів – це множина акторів, варіантів використання, асоціації, відношення та узагальнення. Діаграма прецедентів використовує

наступні об'єкти:

– Актор – це сутність, що виконує будь-яку роль в одній даній системі. Це може бути зовнішня система, людина або організація. Як правило, зображується як скелет.

– Система – використовується для визначення сфери застосування та зображується у вигляді прямокутника. Це необов'язковий елемент, але корисний під час візуалізації великих систем

– Випадок використання – є функцією або дією всередині системи, зображується як овал із назвою функції.

– Пакет – ще один додатковий елемент, який корисний у складних діаграмах. Пакети використовуються для угруповання випадків використання.

Нижче представлена розроблена UML діаграма прецедентів, де відображена взаємодія вебзастосунку з усіма частинами модулю (рис. 2.4).

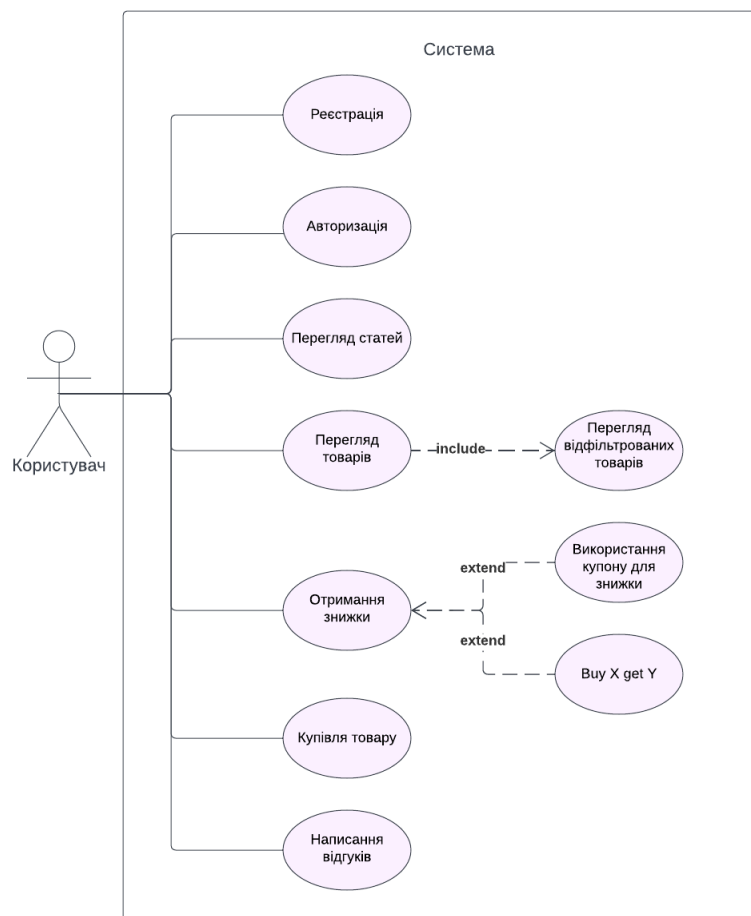


Рисунок 2.6 – UML діаграма прецедентів роботи модулю

На наступній UML діаграмі розгортання представлено архітектуру сервісів для системи електронної комерції Shopify. Існують окремі служби окремих модулів електронної комерції, які взаємодіють за допомогою REST API. Вебзастосунок використовує службу шлюзу для доступу до інших служб. Програма для смартфона підключена до модуля шлюзу мобільного API, який забезпечує зв'язок з іншими сервісами.

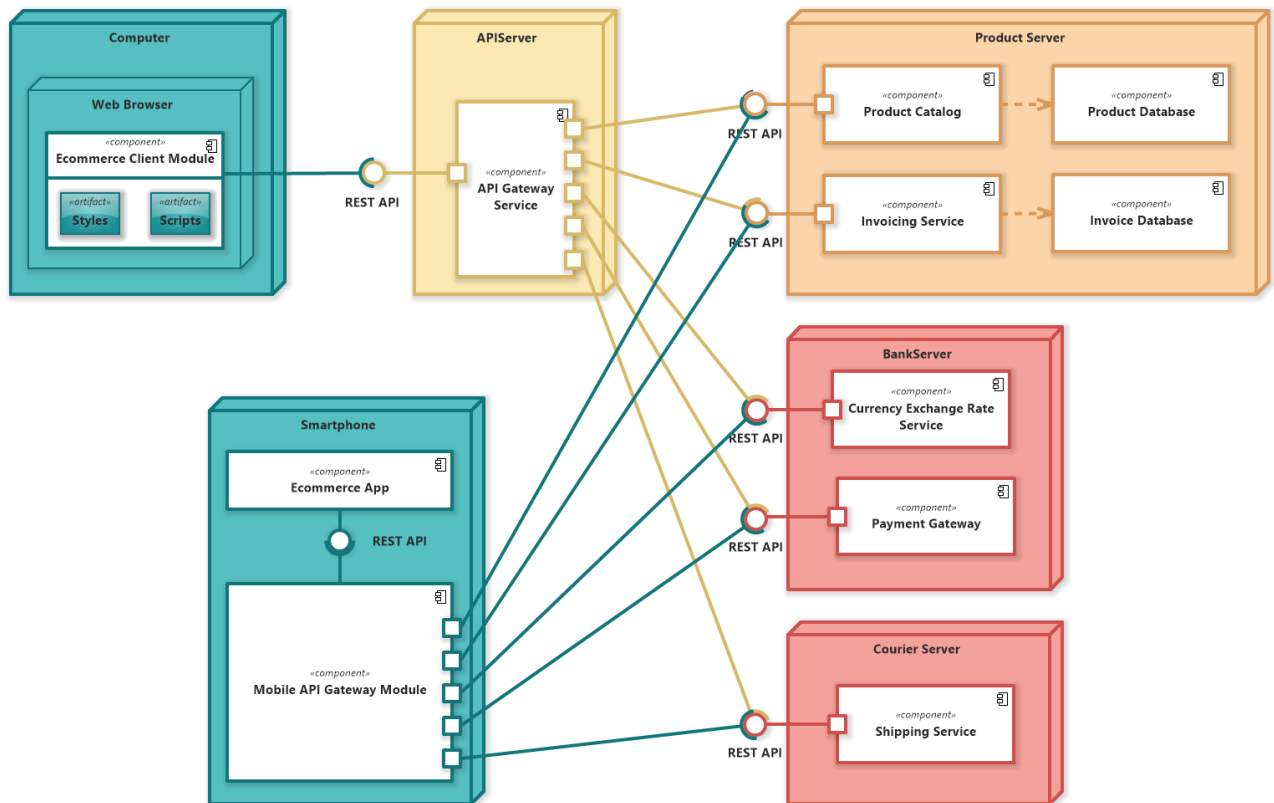


Рисунок 2.7 – UML діаграма розгортання

Платформа Shopify має деякі обмеження щодо створення варіантів продукту, можна створювати до 100 варіантів. І оскільки продукти вебзастосунку будуть мати кілька опцій, можливі ситуації, коли продукт може мати більше 100 варіантів. Тому було вирішено варіанти кольору додавати до магазину як окремі товари з іншим набором опцій. На сторінці продукту передбачена можливість перемикання не лише опцій, які має продукт, а й опції кольору.

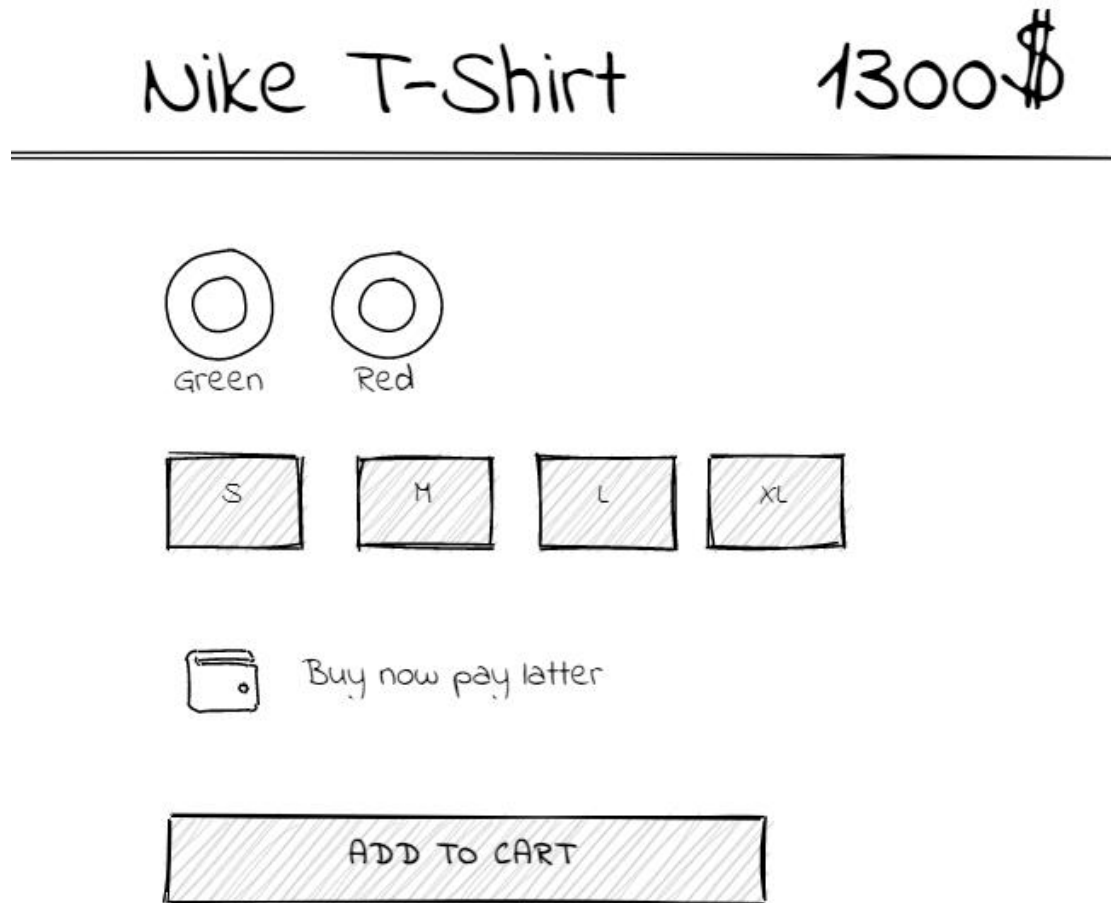


Рисунок 2.8 – Мокіп форми на сторінці продукту

При цьому галерея повинна відображати зображення вибраного кольору, а всі взаємодії з формою повинні бути зав'язані на продукті вибраного кольору (додавання в кошик, вибір опцій та інше). За допомогою Liquid це завдання реалізувати неможливо, оскільки потрібна реактивність. Тому було вирішено використати Vue.js (реалізація більш детально описана у третьому розділі). Vue.js - це прогресивний фреймворк для розробки інтерфейсів і односторінкових веб-додатків на мові JavaScript. Він вирішує завдання рівня представлення (view) та спрощує роботу з бібліотеками. Для кращого розуміння структури компонентів форми та галереї нижче представлена блок-схему успадкування компонентів.

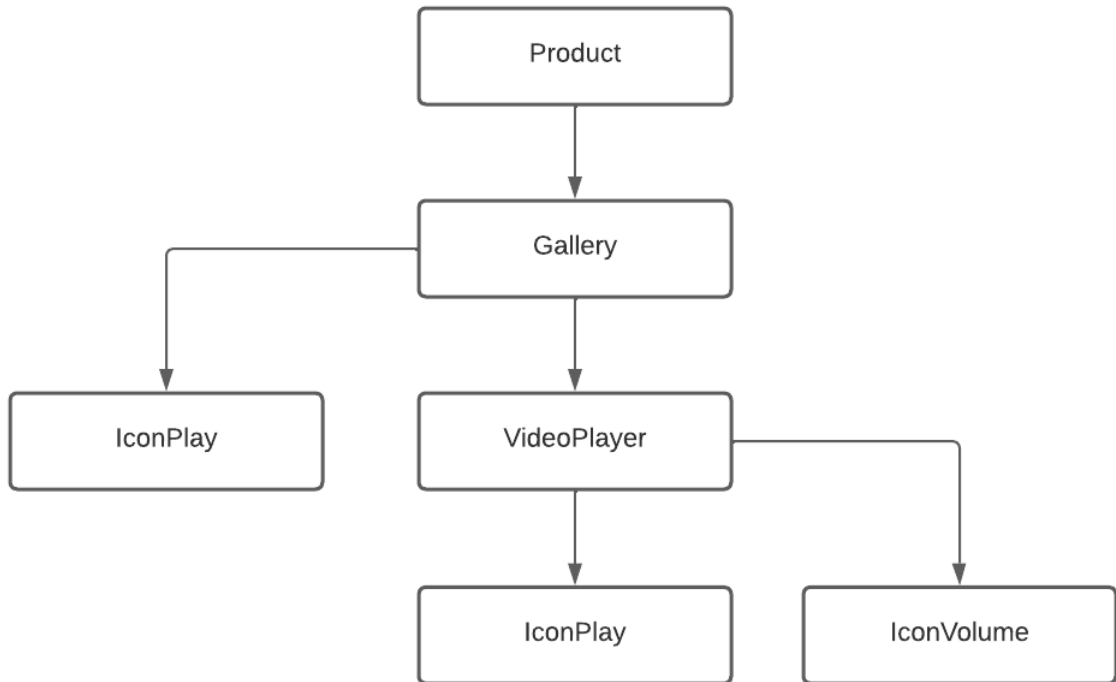


Рисунок 2.9 – Блок-схема успадкування компоненту Gallery

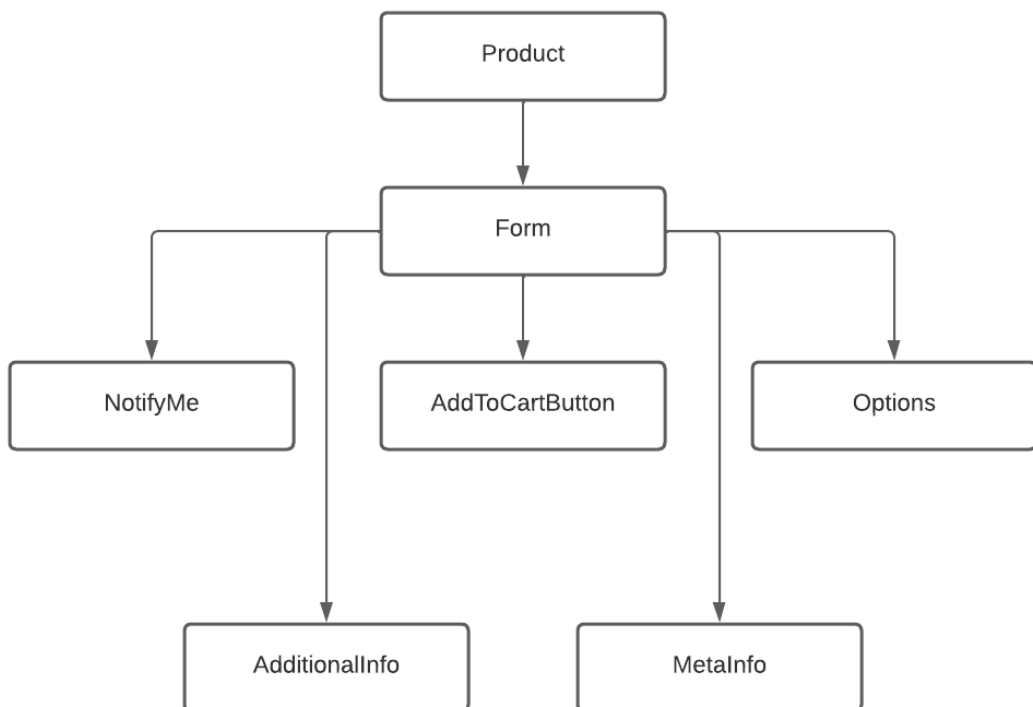


Рисунок 2.10 – Блок-схема успадкування компоненту Form

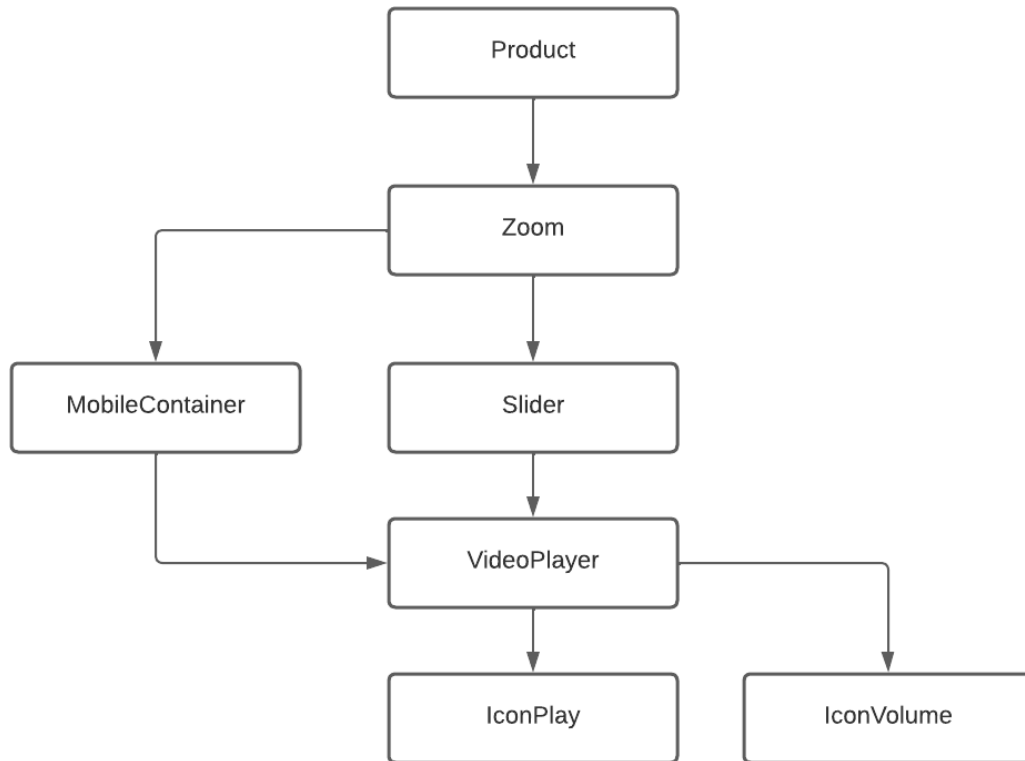


Рисунок 2.11 – Блок-схема успадкування компоненту zoom

Висновки до розділу 2

Отже, в другому розділі визначено основні функціональні можливості модулю, описано алгоритми роботи наступних сценаріїв: додавання товару до кошика з оформленням замовлення, створення облікового запису, процес авторизації, відновлення паролю до акаунту та пошуку товарів. Для більш точної та зрозумілої роботи системи було створено UML діаграми прецедентів. Також було продемонстровано блок-схеми успадкування компонентів (Gallery, Form, Zoom), що допомагає раціонально організувати компоненти.

3 ОГЛЯД ОСНОВНИХ МЕТОДІВ СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЙОГО РОЗРОБКА

Для вирішення поставлених задач з технічного боку, необхідно було знайти рішення, що є найбільш популярним у використанні та таким, що підходить за всіма вимогами системи. Окрім того необхідно враховувати недоліки існуючих систем, які були проаналізовані, щоб не допустити їх у системі, що розробляється. Також необхідно працювати над тим, щоб зменшити вплив існуючих систем на процес використання застосунку. Після проведеного аналізу аналогічних систем та ринку наявних вебтехнологій було прийнято рішення про використання HTML, CSS, Liquid, JavaScript як основної мови для розробки вебзастосунку.

3.1 Liquid

Liquid – це мова шаблонів з вихідним кодом, яка була створена Shopify і написана на Ruby. Liquid застосовується для динамічного накопичення вітрин. Shopify використовує її з 2006 року, а на сьогоднішній день і багатьма іншими розміщеними веб-програмами.



Рисунок 3.1 – Розміщення, які використовують Liquid

Для відображення динамічного контенту, Liquid використовує комбінацію об'єктів, тегів та фільтрів усередині файлів шаблонів [8; 10].

Об'єкти містять вміст Liquid, який відображається на сторінці, об'єкти та змінні відображаються, якщо вони укладені у подвійні фігурні дужки

```
Вхід
{{ page.title }}

Вихід
Introduction
```

Рисунок 3.2 – Приклад відображення об’єкту в Liquid

В цьому випадку Liquid відображує вміст title властивості page об’єкта, що містить текст “Introduction”.

Теги створюють потік та логіку керування для шаблонів. Теги не використовують для жодного видимого виводу при відображенні шаблону – це дозволяє створювати цикли, умови, а також призначати змінні, не відображаючи логіки Liquid на сторінці.

```
Вхід
{% if user %}
  Hello {{ user.name }}!
{% endif %}

Вихід
Hello Adam!
```

Рисунок 3.3 – Приклад використання тегів в Liquid

Фільтри додають зміни до того, як об’єкт або змінна буде виведена. Вони використовуються всередині подвійних фігурних дужок після привласнення змінних і розділяються вертикальною рисою.

```
Вход
{{ "/my/fancy/url" | append: ".html" }}

Виход
/my/fancy/url.html
```

Рисунок 3.4 – Приклад використання фільтрів в Liquid

Мови шаблонів також можна порівняти з функцією “Пошук та заміна” у тестових редакторах. Shopify визначає який шаблон надсилати до браузера, керуючись запитаною URL-адресою. Як тільки система обробляє відповідний шаблон для виведення, відбувається процес аналізу шаблону і зовнішнього файлу розмітки щодо пошуку об’єктів. Як тільки будуть знайдені відповідні елементи, Shopify замінює код Liquid необхідними даними магазину. Після цього фінальна версія документа відправляється браузеру у форматі HTML.

3.2 HTML

HTML (Hyper Text Markup Language) – це мова гіпертекстової розмітки сторінки. Вона використовується для того, щоб надати браузеру зрозуміти, як слід відображати завантажений сайт. Простими словами HTML це каркас сайту. У ньому можуть бути прописані:

- блоки;
- таблиці;
- зображення;
- абзаци;
- форми;
- посилання;
- заголовки тощо.

На сьогоднішній день представлено п’ять версій мови. Перша була випущена між 1986 і 1991 роками, а остання (версія 5.2) у 2017 році. Мова HTML

мала стати незалежною від будь-яких платформ – відображати скрізь однаково. Але цього не було досягнуто, тому що у користувачів зростали вимоги до мультимедіа.

HTML



Рисунок 3.5 – Логотип HTML5

Як результат, код інтерпретується по-різному не лише на пристроях з різною операційною системою, а й у різних браузерах. Це не є мінусом, адже що вимоги відрізняються. Користувач, який використовує телефон, щоб скористатися сайтом, готовий до мінімалістичного дизайну і до обмеженого функціоналу, а власник персонального комп'ютеру хоче більшого.

У п'ятій версії HTML став більш незалежним від інших мов. Наприклад, раніше перевірка правильного заповнення форм розроблялася виключно з використанням Javascript, тепер частину завдань HTML може обробляти самостійно. Також з'явився вбудований програвач, тобто тепер не потрібно підключати Adobe Flash Player або інші плагіни. Візуальна частина перетягування елементів зараз реалізується на HTML і CSS, якщо додати атрибут

draggable.

3.3 CSS

CSS (Cascading Style Sheets) – це каскадні таблиці стилів. По суті, це мова, яка відповідає за опис зовнішнього вигляду каркасу вебзастосунка, тобто HTML. Раніше веб-розробники не використовували дану технологію – так було на зорі розвитку інтернет-технологій. За допомогою HTML можна було виділити заголовок, параграф або змінити зображення тексту – більшого й не потрібно було. Згодом цього було недостатньо, тому і з'явилася технологія форматування без зміни вмісту та структури документа. CSS допомагає вирішити проблему “зоопарку” тегів форматування, коли різні браузері підтримували різні теги. Їх уніфікація та єдина база спростили роботу з веб-документами та полегшили життя веб-розробникам.



Рисунок 3.6 – Логотип CSS3

CSS використовується для визначення правил оформлення документів – включаючи верстку, дизайн та варіації макета для різних пристроїв та розмірів екрану. В цього способу форматування є кілька переваг:

- документ простіше обслуговувати;
- зовнішній вигляд всього сайту можна змінити централізовано, а не коригувати форматування кожної сторінки;
- теги не дублюються.

CSS стилі можна розмістити всередині тега <Head> або використовувати

окремий файл з розширенням “.css”.

3.4 Javascript та Vue.js

Javascript – це мова, яка дозволяє застосовувати складні речі на вебсторінці кожен раз, коли на вебсторінці відбувається щось більше, ніж її статичне відображення. Тобто це мова програмування, яка дозволяє створити динамічно-оновлюваний контент, анімувати будь що, керувати мультімедіа, втім, робить все, що завгодно.



Рисунок 3.7 – Логотип Javascript

Ядро мови Javascript дозволяє робити наступне:

- Зберігати дані усередині змінних.
- Операції з DOM-деревом. DOM – це об’єктна модель документа, котру браузер створює в пам’яті комп’ютера на основі коду HTML, отриманого ним від сервера.
 - Запускати код відповідно до певних подій на веб-сторінці. Наприклад, робити певну дію, коли користувач натискає на певну кнопку.
 - Використання функціональності інтерфейсів прикладного програмування (API), що надають додаткові надздібності для використання Javascript у власному коді.

API - це готові набір блоків коду, які дозволяють розробникам реалізувати програми, які інакше біло б важко або неможливо реалізувати. Якщо провести

аналогію з випадком із життя, то вони роблять те саме в сфері програмування, що готові комплекти меблів для домашнього будівництва. Тобто набагато простіше брати готові елементи меблів і скручувати їх разом, ніж створювати власноруч з нуля.

Як основний спосіб взаємодії програм з Shopify, використовується Shopify Admin API, що дозволяє додавати власні функції в інтерфейс користувача магазину. Shopify Admin API підтримує як Rest, так і GraphQL.

Vue.js - це прогресивний фреймворк для розробки інтерфейсів і односторінкових веб-додатків на мові JavaScript [3]. Він вирішує завдання рівня представлення (view) та спрощує роботу з бібліотеками.



Рисунок 3.8 – Логотип Javascript

Особливості Vue [2; 9]:

- Реактивність. Vue.js – це реактивний MVC-фреймворк, подання (view) змінюється у міру зміни моделі.
- Прогресивність. Ядро Vue.js досконально підходить для впровадження у існуючий проєкт. Таким чином фреймворк має можливість працювати з раніше використовуваними бібліотеками.

- Простота розуміння. Почати працювати з фреймворком можна без базових знань у веб-розробці але авжеж легше буде засвоїти нову технологію з базовими знаннями.
- Невеликий розмір. Фреймворк займає близько 20 кБ, тому реалізовані у ньому проєкти швидше краще ранжуються пошуковими роботами, а також швидше завантажуються.
- Швидка розробка. На сьогоднішній день існує велика кількість інструкцій, документації і готових рішень для Vue.js, які дозволяють вирішити будь-яку проблему, якщо вона виникає під час створення проєктів.
- Доступність. Фреймворк розповсюджується за ліцензією MIT, тому його можливо вільно використовувати як в особистих, так і в комерційних цілях.

3.5 Файлова структура проєкту

У процесі створення вебзастосунку магазину одягу на Shopify був використаний збирач на theplify, тому файлова структура складається з наступних каталогів:

- Assets. Каталог assets містить усі активи, що використовуються в темі, включаючи файли зображень, CSS та Javascript.
- Config. Каталог config містить файли конфігурації для теми. Файли конфігурації визначають установки в області налаштувань теми редактора тем, а також зберігають їх значення.
- Layout. Каталог layout містить файли макета для теми, через які рендеруються файли шаблонів. Макети – це рідкі файли, які дозволяють містити контент, який повинен повторюватися на декількох типах сторінок в одному місці.
- Locales. Каталог locales містить файли локалі для теми, які використовуються для надання перекладеного контенту. Файли локалі дозволяють надавати переклад у редакторі тем, забезпечувати переклади для інтернет-магазину та дозволяють продавцям налаштовувати текст в інтернет-магазині.

- **Sections.** Каталог sections містить розділи теми. Розділи – це рідкі файли, які дозволяють створювати повторні модулі контенту, які можуть бути налаштовані продавцями. Вони також можуть включати блоки, які дозволяють продавцям додавати, видаляти і змінювати продрок контенту в розділі.
- **Snippets.** Каталог snippets містить файли Liquid, в яких розміщуються невеликі повторно використовувані фрагменти коду.
- **Templates.** Каталог templates містить файли шаблонів теми, які управляють тим, що відображається на кожному типі сторінки.



Рисунок 3.9 – Схема файлової структури проекту

Висновки до розділу 3

Отже, в третьому розділі описано технології, методи та фреймворк що використовувались для розробки вебзастосунку магазину одягу:

- Liquid.
- HTML.
- CSS.
- JavaScript.
- Vue.js.

Кожну технологію було детально описано, наведено переваги у використанні. Вищеперераховані мови та технології це не від’ємна частина майбутнього вебзастосунку магазину одягу. Також описано файлову структуру проекту.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Огляд програмного забезпечення

Вебзастосунок магазину одягу має вебінтерфейс, що складається з наступних сторінок:

- homepage;
- collection page;
- contact page;
- product page;
- blog page;
- article page;
- register page;
- login page;
- cart page.

Зазвичай на головній сторінці розміщується найважливіша та найцікавіша інформація, щоб залучити покупця продовжити переходити на інші сторінки сайту. На головній сторінці можна розміщувати будь-які види секцій та у будь-якому порядку. Для неї були обрані наступні секції:

- slideshow (рис. 4.1) – в налаштуваннях секції та її блоків є можливість встановлювати декілька слайдів з зображеннями, текстом, посиланням та іншим (рис. 4.2);
- featured collection (рис. 4.3) – в налаштуваннях секції є можливість зміни заголовку, вибору колекції з якої будуть відображатись товари та інші налаштування (рис. 4.4);
- image banner (рис. 4.5) – в налаштуваннях секції та її блоків є можливість встановлювати зображення, текст, посилання та інше (рис. 4.6);
- video (рис. 4.7) – в налаштуваннях секції є можливість є можливість встановити текст, посилання на відео у YouTube та обрати placeholder (рис. 4.8) [8];

- multi column (рис. 4.9) – в налаштуваннях секції є можливість встановити текст, ширину блоків, зображення, форму зображення, посилання та інше (рис. 4.10);
- richtext (рис. 4.11) – в налаштуваннях секції є можливість встановити текст, колір фону, колір тексту секції та інше (рис. 4.12);
- newsletter (рис. 4.13) – в налаштуваннях секції є можливість встановити текст, колір тексту, колір фону секції, додати форму та інше (рис. 4.14).

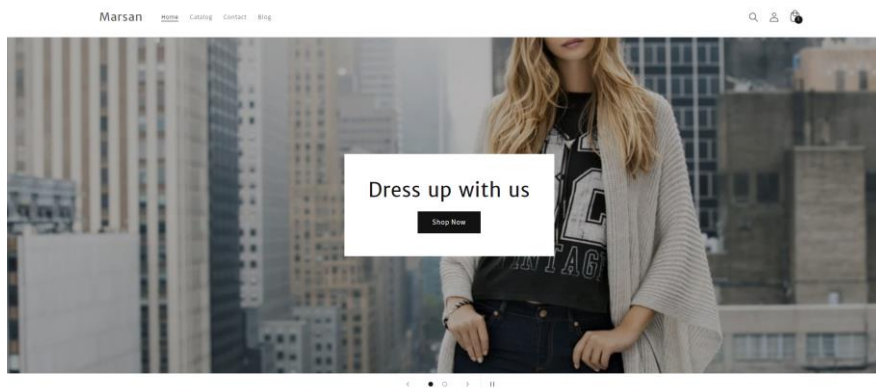


Рисунок 4.1 – Вигляд секції slideshow

Slideshow

Layout

Full width

Slide height

Large

Pagination style

Dots

Auto-rotate slides

Change slides every

3s

MOBILE LAYOUT

Show content below images on mobile

Рисунок 4.2 – Налаштування секції slideshow

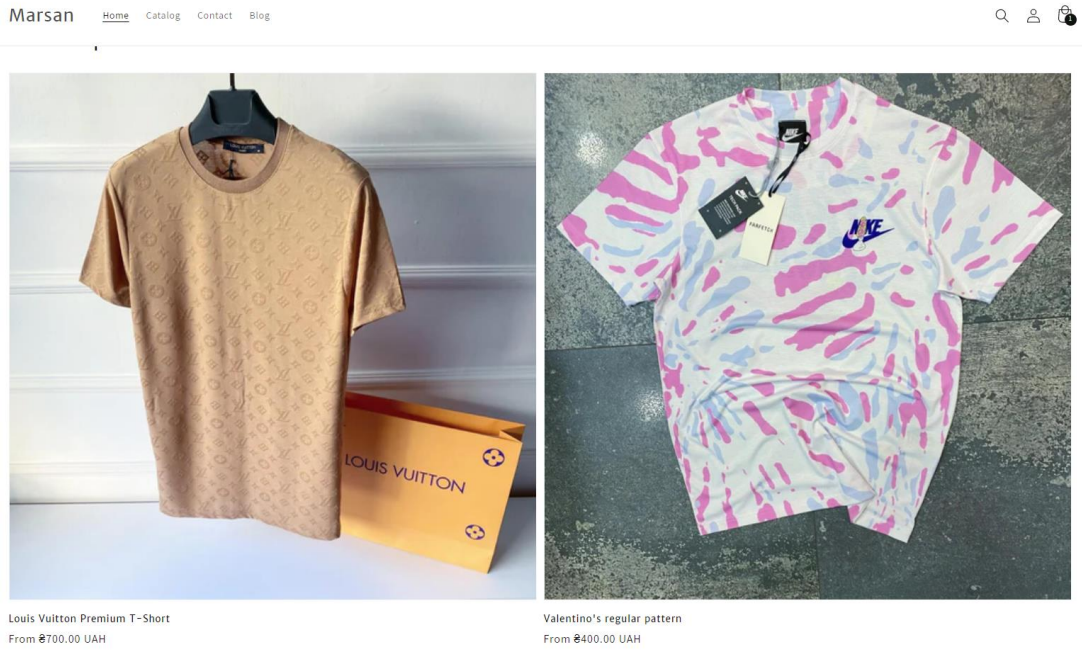


Рисунок 4.3 – Вигляд секції featured collection

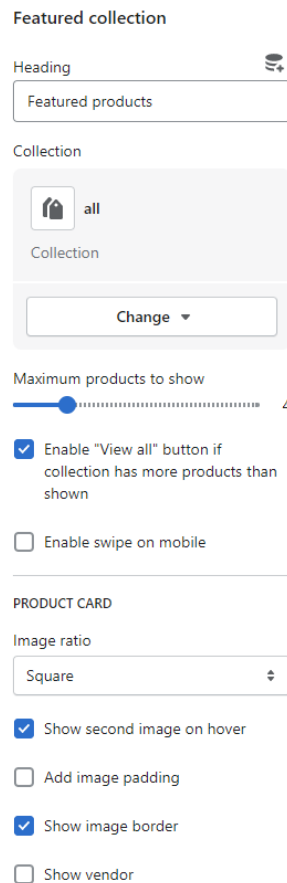


Рисунок 4.4 – Налаштування секції featured collection

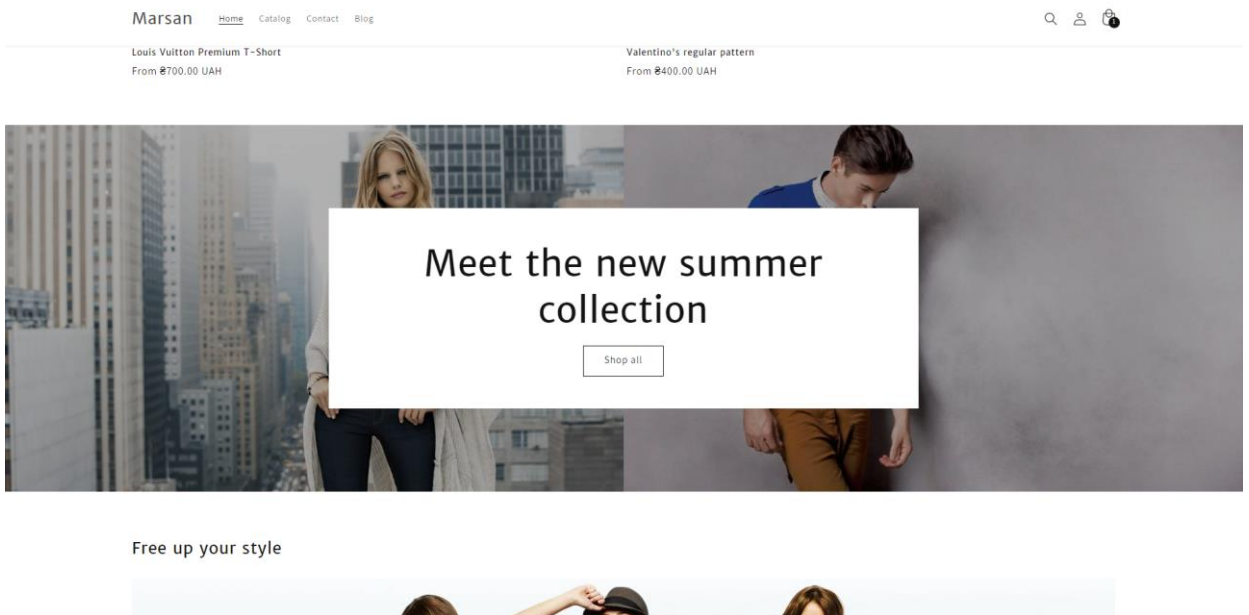


Рисунок 4.5 – Вигляд секції image banner

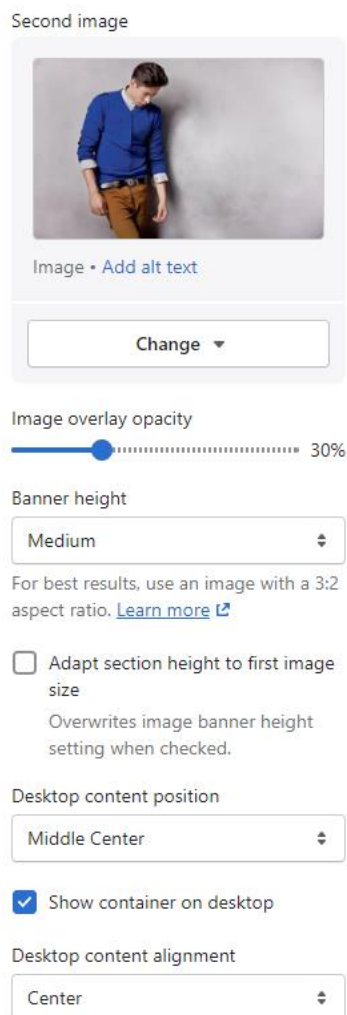


Рисунок 4.6 – Налаштування секції image banner

Free up your style

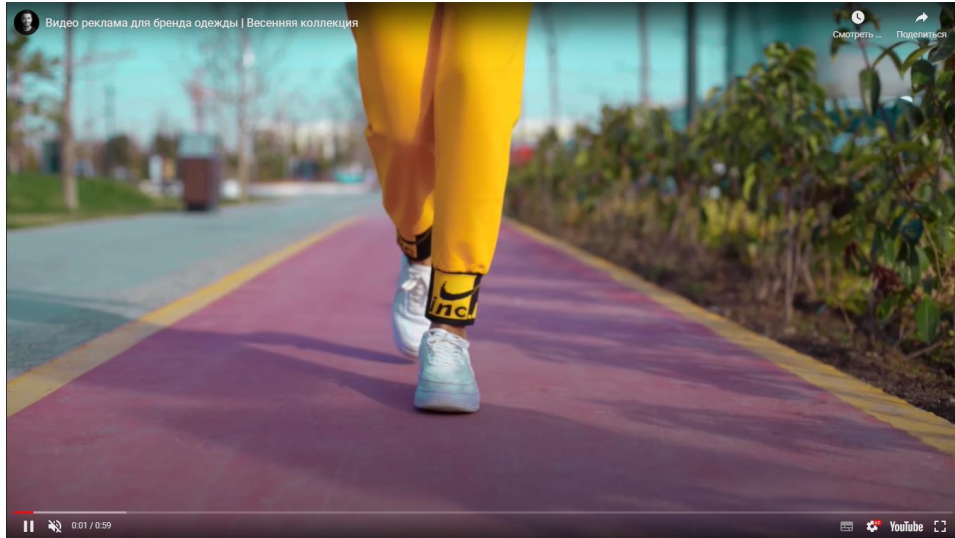


Рисунок 4.7 – Вигляд секції video

Video

Heading

Free up your style

Cover image





Image • Add alt text

Change ▾

URL

[https://www.youtube.com/watch?v=](https://www.youtube.com/watch?v=DBX7upvyrSM)



Видео реклама для бренда одежды
| Весенняя коллекция
[youtube.com/watch?v=DBX7upvyrSM](https://www.youtube.com/watch?v=DBX7upvyrSM)

Video plays in the page.

Video alt text

Describe the video to make it accessible for customers using screen readers.

Make section full width

Рисунок 4.8 – Налаштування секції video

New Summer Collection



Men's collection

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English.

[Shop Now](#) →



Women's collection

It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

[Shop Now](#) →



Unisex collection

Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

[Shop Now](#) →

Рисунок 4.9 – Вигляд секції multi column

Multicolumn

Heading

New Summer Collection

Image width

Full width of column

Image ratio

Adapt to image

Column alignment

Left

Secondary background

None

Button label

Button link

Paste a link or search

Enable swipe on mobile

Рисунок 4.10 – Налаштування секції multi column

Have questions? Write to us

There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure there isn't anything embarrassing hidden in the middle of text.

Contact Us

Рисунок 4.11 – Вигляд секції richtext

Heading

Heading

Have questions? Write to us

Heading size

Medium

Рисунок 4.12 – Налаштування секції richtext

Subscribe now so you don't miss out on updates

Be the first to know about new collections and exclusive offers.

Email



Рисунок 4.13 – Вигляд секції newsletter

Color scheme

Background 2



Make section full width

Each email subscription creates a customer account. [Learn more](#)

Рисунок 4.14 – Налаштування секції newsletter

На сторінці колекції відображаються продукти, які знаходяться у вказаній колекції (рис. 4.15).

Products

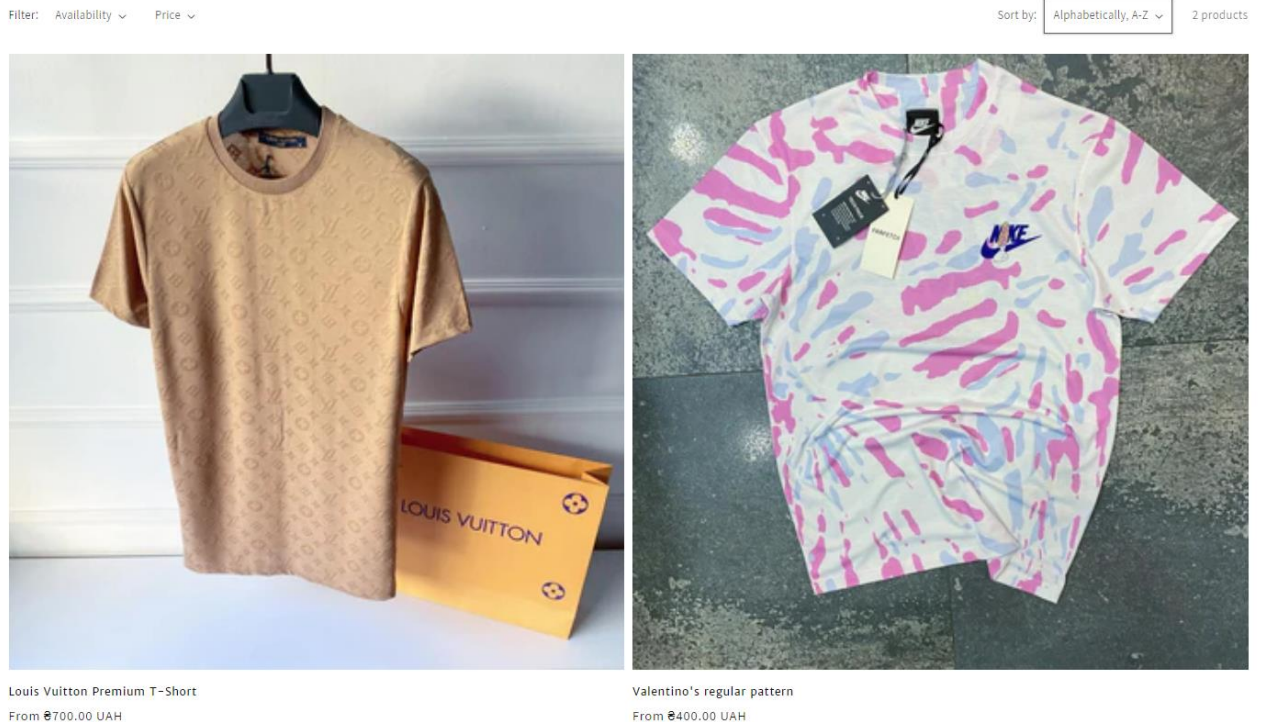


Рисунок 4.15 – Вигляд сторінки колекції

На сторінці колекції відсортувати продукти за наступними параметрами (рис. 4.16):

- featured;
- best selling;
- alphabetically, A-Z;
- alphabetically, Z-A;
- price, low to high;
- price, high to low;
- date, old to new;
- date, new to old.

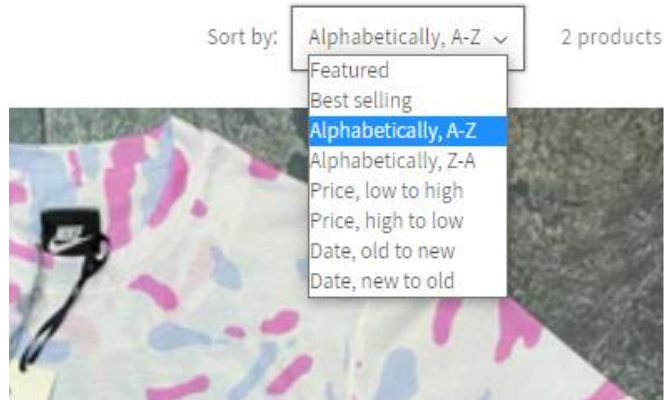


Рисунок 4.16 – Вигляд сортування на сторінці колекції

Також є можливість фільтрування товарів за наступними критеріями:

- availability (рис. 4.17);
- price (рис. 4.18).

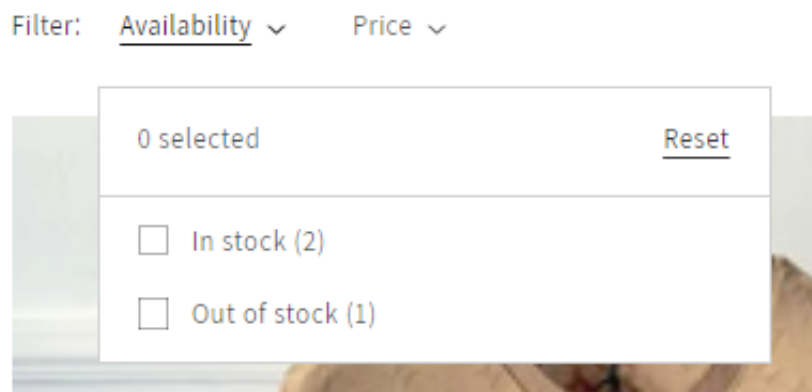


Рисунок 4.17 – Вигляд фільтру availability

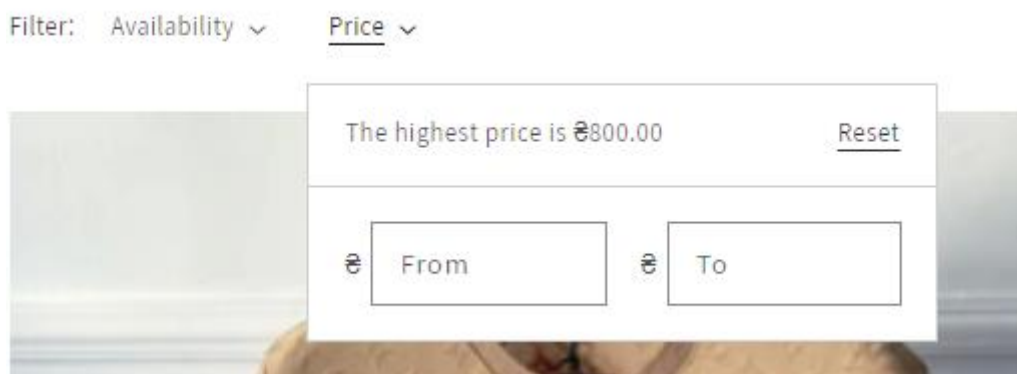


Рисунок 4.18 – Вигляд фільтру price

На сторінці продукту відображається докладна інформація щодо вибраного

продукту. На сторінці відображається графічний контент продукту, його ціна, назва, опис, кнопка додавання до кошика. Також якщо продукт має опції з варіантами, тоді відображаються опції, які можна переключити - при перемиканні опцій змінюється графічний контент варіанта. Здебільшого саме через цю сторінку додають продукт у кошик.

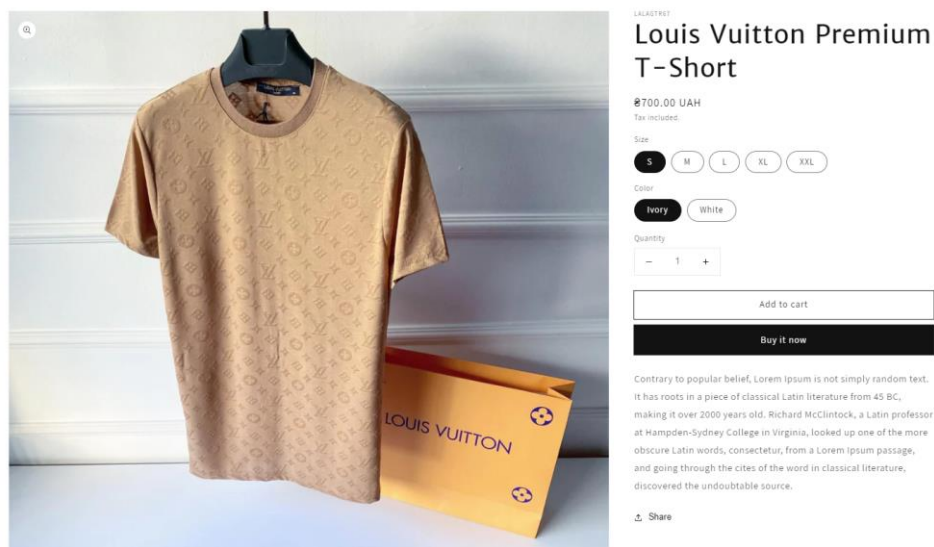


Рисунок 4.19 – Вигляд сторінки продукту

На сторінці кошика відображаються товари, додані до неї. Також є опис товарів, які були додані та загальна сума купівлі.

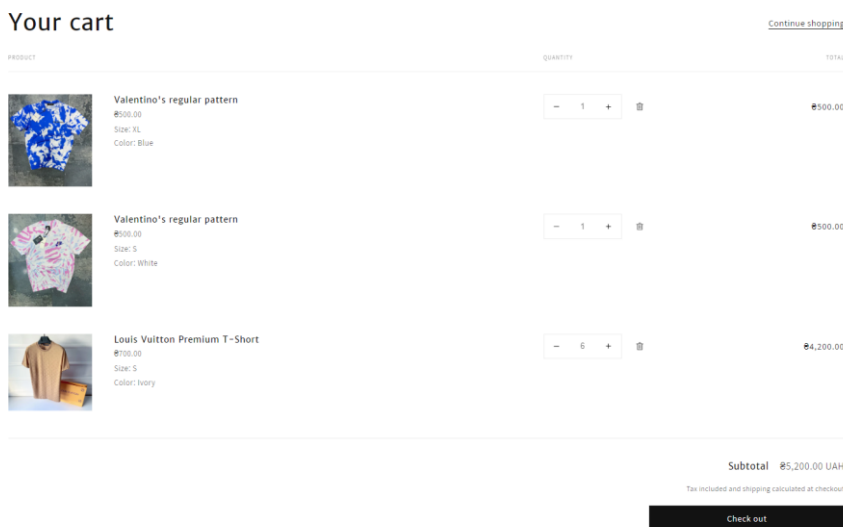


Рисунок 4.20 – Вигляд сторінки кошику

Вебзастосунок дає можливість власнику магазину публікувати новини та цікаву інформацію за допомогою блогу та його статей. На сторінці блогу відображається список існуючих статей. Сторінка статей відображає контент статтю, на цій же сторінці можна поділитися статей у соціальних мережах.

News



Post1

FEBRUARY 7, 2022

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Interdum consectetur libero id faucibus nisl. Platea dictumst vestibulum rhoncus est....



Post2

JANUARY 25, 2022

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Interdum consectetur libero id faucibus nisl. Platea dictumst vestibulum rhoncus est....

Рисунок 4.21 – Вигляд сторінки блогу



Post1

FEBRUARY 1, 2022

Share

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Interdum consectetur libero id faucibus nisl. Platea dictumst vestibulum rhoncus est. Turpis egestas pretium aenean pharetra magna ac placerat vestibulum. Sagittis eu volutpat odio facilisis mauris sit amet. Lectus mauris ultrices eros in cursus turpis massa tincidunt dui. Viverra suspendisse potenti nullam ac. Velit aliquet sagittis id consectetur. Neque gravida in fermentum et sollicitudin ac orci phasellus. Libero volutpat sed cras ornare arcu dui vivamus arcu.

Dui ut ornare lectus sit amet est placerat in. Eu volutpat odio facilisis mauris sit amet massa. Faucibus pulvinar elementum integer enim neque volutpat ac tincidunt vitae. Est velit egestas dui id ornare arcu odio ut. Sed libero enim sed faucibus. Ante metus dictum at tempor commodo ullamcorper a lacus. Id donec ultrices tincidunt arcu non sodales neque. Massa sapien faucibus et molestie ac feugiat sed lectus. Pellentesque adipiscing commodo elit at imperdiet dui accumsan sit amet. Non sodales neque sodales ut. Aliquam ultrices sagittis orci a scelerisque purus semper eget dui. Arcu vitae elementum curabitur vitae nunc sed. Facilisis magna etiam tempor orci eu lobortis elementum. Habitasse platea dictumst quisque sagittis purus sit.

Elit ut aliquam purus sit. Massa tempor nec feugiat nisl. Semper feugiat nibh sed pulvinar proin gravida hendrerit lectus a. Gravida quis blandit turpis cursus in hac habitasse. Nunc sed blandit libero volutpat sed. Turpis egestas pretium aenean pharetra magna. Ac tortor vitae purus faucibus ornare suspendisse sed. Sollicitudin nibh sit amet commodo nulla facilisi nullam vehicula ipsum. Sem viverra aliquet eget sit. Elementum sagittis vitae et leo dui ut diam. Cursus mattis molestie a

Рисунок 4.22 – Вигляд сторінки статті

У розробленому вебзастосунку за допомогою сторінок реєстрації та авторизації користувачі мають можливість зареєструвати обліковий запис і надалі використовувати його для входу в особистий кабінет, де відображається інформація поточним замовленням, історія замовлень та інше.

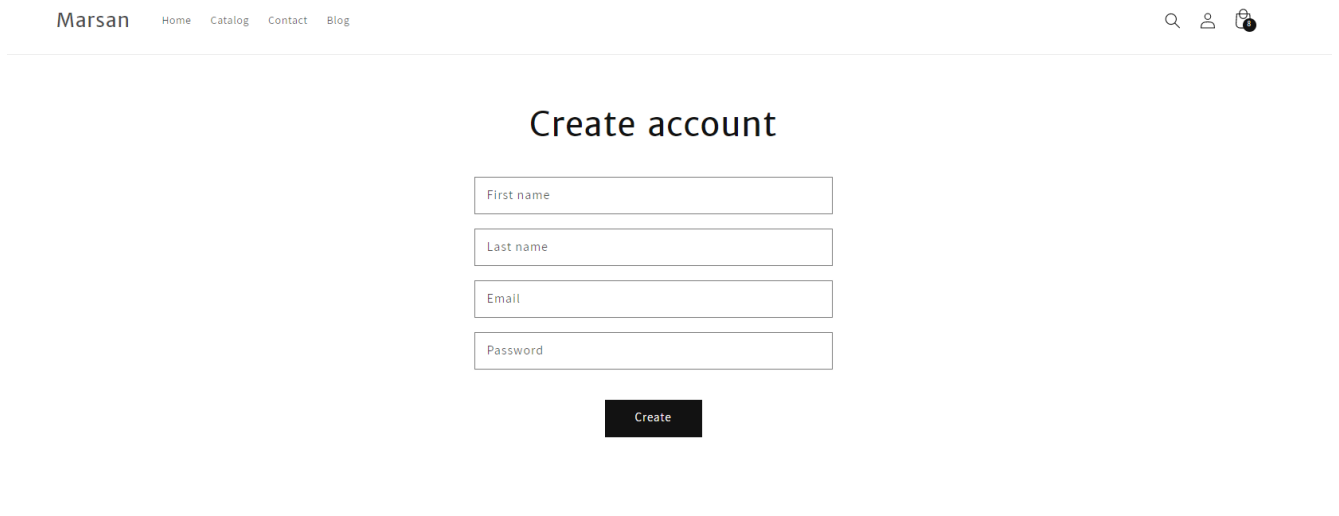


Рисунок 4.23 – Вигляд сторінки реєстрації

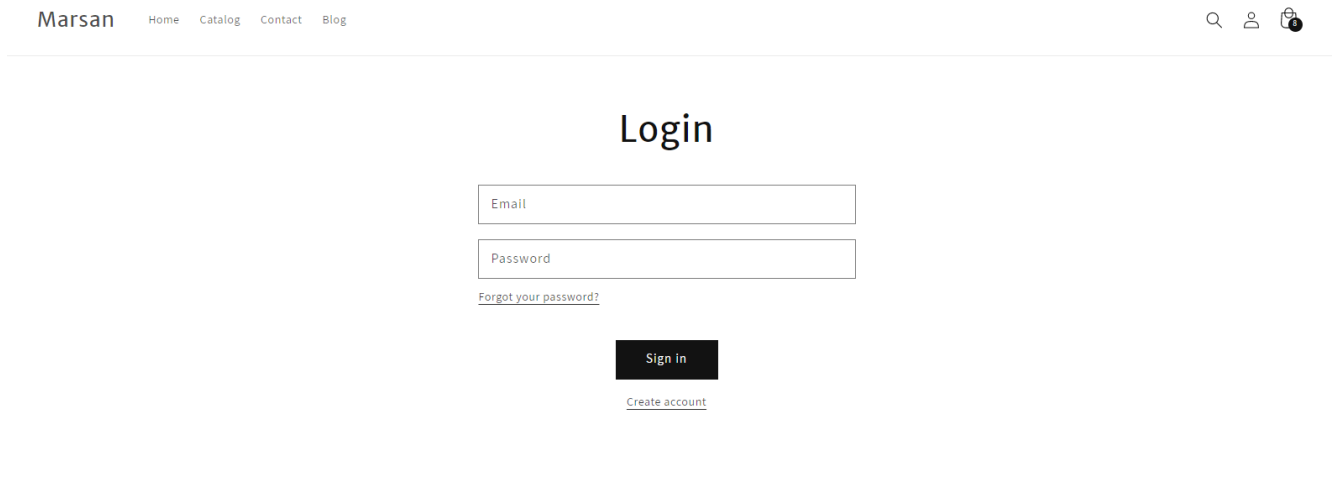


Рисунок 4.24 – Вигляд сторінки авторизації

Сторінка пошуку дуже схожа на сторінку колекцій. Вона має точно той же функціонал у вигляді фільтрів та сортування. На цій сторінці відображаються продукти, які відповідають параметрам запити, надісланому користувачем. На сторінці також можна повторно ввести запит для відображення нових результатів.

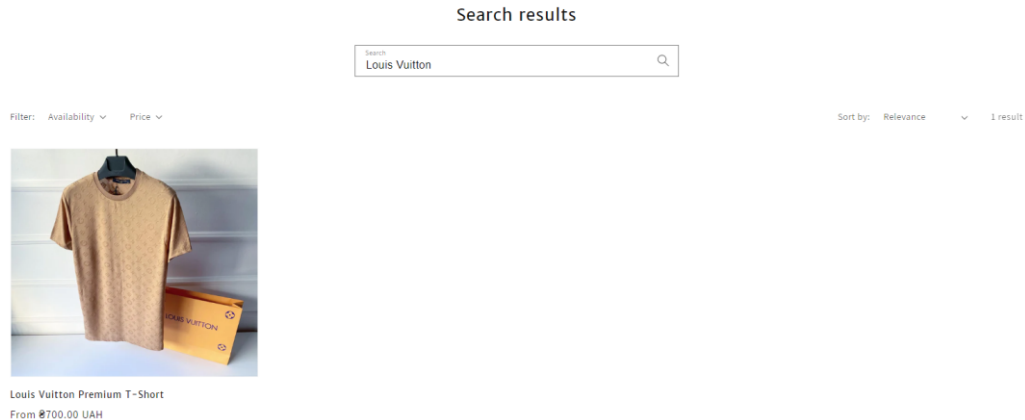


Рисунок 4.25 – Вигляд сторінки пошуку

На сайті є наступні унікальні секції, які відображаються на всіх сторінках сайту:

- announcement bar;
- header;
- footer.

Секція announcement bar служить для відображення найважливішої інформації. Налаштування секції має можливість змінити текст та посилання на потрібну сторінку.

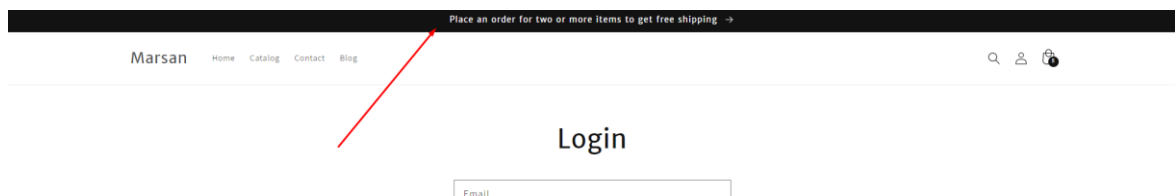


Рисунок 4.26 – Вигляд секції announcement bar

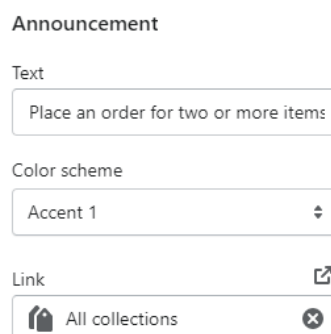


Рисунок 4.27 – Налаштування секції announcement bar

Шапка сайту містить посилання на основні сторінки сайту. В налаштуваннях секції є можливість обрати схему кольорів шапки, логотип, розмір логотипу, позиції логотипу, меню з навігації та інше.

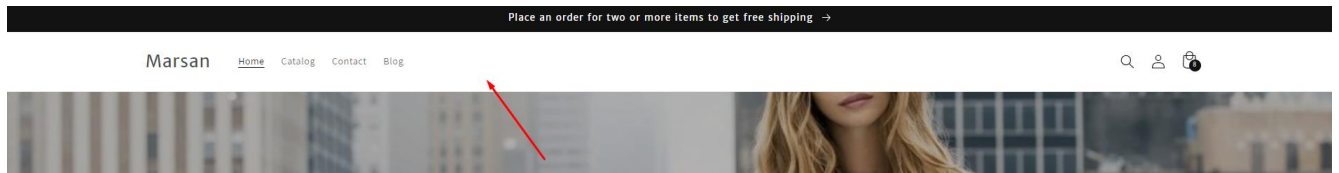


Рисунок 4.28 – Вигляд секції header

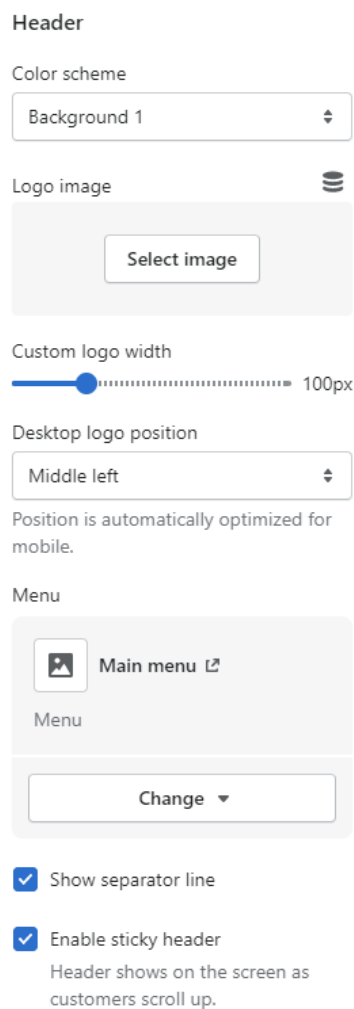


Рисунок 4.29 – Налаштування секції header

Підвал сайту також має посилання на головні сторінки сайту, текстову інформацію, інформацію про платіжні системи, а також можливість зміни регіону та валюти (якщо в налаштуваннях магазину є можливість оплати в іншій валюті). В налаштуваннях секції є можливість обрати меню з навігації, текст, відображення поштової форми та інше.

Quick links

[Home](#)
[Catalog](#)
[Contact](#)
[Blog](#)

Heading

All the Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making this the first true generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic words etc.

Subscribe to our emails

 →

Country/region

Ukraine (UAH) ▾



© 2022, laagtr67 Powered by Shopify

Рисунок 4.30 – Вигляд секції footer

Footer

Color scheme

Background 1

EMAIL SIGNUP

Subscribers added automatically to your "accepted marketing" customer list.

[Learn more](#)

Show email signup

Heading

Subscribe to our emails

SOCIAL MEDIA ICONS

To display your social media accounts, link them in your theme settings.

Show social media icons

COUNTRY/REGION SELECTOR

To add a country/region, go to your [payment settings](#).

Enable country/region selector

LANGUAGE SELECTOR

To add a language, go to your [language settings](#).

Enable language selector

PAYMENT METHODS

Show payment icons

Рисунок 4.31 – Налаштування секції footer

4.2 Логіка об'єднання продуктів

Платформа Shopify досить велика і надає досить багато можливостей для магазинів своїх клієнтів, але є деякі обмеження, обхід яких вимагає розробку нестандартних рішень. Одне з таких обмежень – це створення не більше 100 варіантів для одного продукту. Дане обмеження може ускладнити розробку для багатьох магазинів – не всі магазини мають продукти з однією або парою опцій. У випадку одягу, в одного товару може бути дуже багато опцій з великим набором значень, наприклад:

- розмір;
- колір;
- матеріал;
- узор та інші.

У такому випадку магазин одягу стикається з цим обмеженням і не зможе заповнити продукти так, як потрібно. Існують програми з магазину Shopify, які дозволяють створювати більше 100 варіантів продуктів, але майже всі вони платні та мають досить велику ціну за їх використання.

Для вирішення цієї проблеми без використання програми було вирішено створювати варіанти кольору як окремі продукти з однаковими наборами опцій [1]. Це дозволило зберегти всі варіанти продуктів та обійти обмеження Shopify у створенні до 100 варіантів. Але тоді у нас виникла проблема з їх поєднанням між собою на сторінці продукту. Магазин потребував функціоналу, який дозволить перемикати опції кольору, але без нестандартного вирішення цієї мети неможливо було досягти.

Тому було вирішено поєднувати продукти між собою за допомогою ручних колекцій. Shopify дозволяє створювати приблизно до 100.000 ручних колекцій – це лише припущення багатьох досвідчених розробників, оскільки точної інформації немає. Звичайно можна було б поєднувати продукти між собою за допомогою тегів або метатолів, але для менеджменту колекції набагато зручніші. Кожна колекція варіантів продукту створювалася за певним шаблоном – назва

колекції повинна включати "Variant-Collection". Приклад коду, який дозволяє отримати інформацію про варіанти кольору потрібного продукту:

```
{%- for collection in product.collections -%}
  {%- if collection.title contains "Variant-Collection" -%}
    {%- assign product_variants = collection.products -%}

    {%- capture product_colors -%}
      [
        {%- for product_variant in product_variants -%}
          {% for tag in product_variant.tags %}
            {% assign tag_down = tag | downcase %}

            {% if tag_down contains "color_" %}
              {% assign product_color = tag_down | split: "color_" | last %}
              {% break %}
            {% else %}
              {% assign product_color = "unset" %}
            {% endif %}
          {% endfor %}
        {%- endfor %}

        {
          "value": {{- product_color | json -}},
          "colorSwatchImage": {{-
product_variant.metafields["global"]["color_swatch_image"] | json -}},
          "disabled": {{- false | json -}}
        }{%- unless forloop.last == true -%},{%- endunless -%}
      ]
    {%- endcapture -%}

    {%- capture product_colors_map -%}
      [
        {%- for product_variant in product_variants -%}
          {% for tag in product_variant.tags %}
            {% assign tag_down = tag | downcase %}

            {% if tag_down contains "color_" %}
              {% assign product_color = tag_down | split: "color_" | last %}
              {% break %}
            {% else %}
              {% assign product_color = "unset" %}
            {% endif %}
          {% endfor %}

          {
            "id": {{- product_variant.id -}},
            "value": {{- product_color | json -}},
            "selected": {{- false -}}
          }{%- unless forloop.last == true -%},{%- endunless -%}
        {%- endfor %}
      ]
    {%- endcapture -%}

    {%- break -%}
  {%- endif -%}
{%- endfor -%}
```

Саме ця частина коду допомагає за допомогою Liquid знайти потрібну колекцію з варіантами кольору будь-якого продукту, записати у форматі JSON та передати цю інформацію до Vue.

Колір варіанта вказується у тегах продукту.

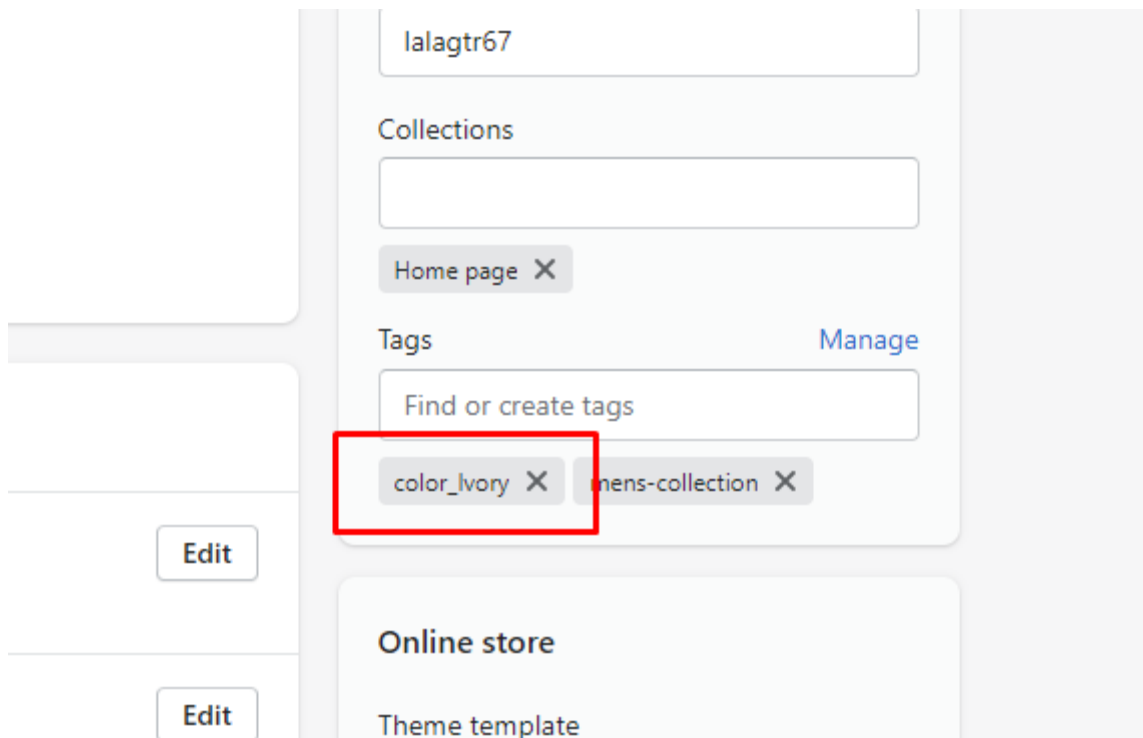


Рисунок 4.32 - Позначення кольору продукту за допомогою тега

Висновки до розділу 4

Отже, в четвертому розділі зображено вебінтерфейс розроблених сторінок вебзастосунок магазину одягу. Описано налаштування секцій та подано нестандартне вирішення проблеми з деяким обмеженням платформи Shopify. Показано результати роботи програмного забезпечення.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи розроблено вебзастосунок магазину одягу з виконанням усіх поставлених раніше завдань. Виконано аналіз предметної області, а також глибоко досліджено усі специфікації до вимог розробки програмного забезпечення. Виконано порівняння існуючих вебзастосунків магазину одягу, досліджені переваги та недоліки кожного з них. Реалізовано усі функції, що вирішують поставлені проблеми для користувачів та власника цього вебзастосунку. Для успішної реалізації виконано моделювання програмного забезпечення через блок-схеми та UML-діаграми. Описано усі засоби розробки ПЗ та застосовані методи. Отже, усі поставлені завдання виконані успішно з дотримання усіх поставлених вимог специфікації.

Використовуючи вебзастосунок магазину одягу, користувач може здійснити замовлення, не виходячи з дому. У сучасних реаліях це має велику перевагу перед фізичними магазинами одягу. Також, даний застосунок допоможе власнику магазину автоматизувати процес продажу товарів, що значно зекономить час на реалізацію товару та зменшити витрати на оренду приміщення.

Розроблений застосунок має зручний та інтуїтивно зрозумілий у використанні інтерфейс. Для розробки застосунка використано найсучасніші технології та методики, обрана ефективна архітектура для досягнення мети.

Тому створення представленого вебзастосунку магазину одягу зекономить час та допоможе користувачам здійснювати замовлення, не виходячи з дому, наражаючи себе на небезпеку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. How to group products into products with variants: веб-сайт. URL: <https://matrixify.app/tutorials/shopify-group-products-into-products-with-variants/> (дата звертання 01.12.2021);
2. Introduction to Vue.js with a Single Page Application (SPA): веб-сайт. URL: <https://www.red-gate.com/simple-talk/development/javascript/introduction-to-vue-js-with-a-single-page-application-spa-in-visual-studio/> (дата звертання 05.12.2021);
3. What is Vue?: веб-сайт. URL: <https://vuejs.org/guide/introduction.html> (дата звертання 30.01.2022);
4. What is Vuex?: веб-сайт. URL: <https://vuex.vuejs.org/> (дата звертання 06.02.2022);
5. UML Diagram Types Guide. Learn About All types of UML Diagrams with Examples: веб-сайт. URL: <https://creately.com/blog/diagrams/uml-diagram-types-examples/> (дата звертання 06.02.2022);
6. Shopify Admin API: веб-сайт. URL: <https://shopify.dev/api/admin> (дата звертання 22.02.2022);
7. Shopify Section Input Settings: веб-сайт. URL: <https://shopify.dev/themes/architecture/settings/input-settings> (дата звертання 28.03.2022);
8. Shopify Cheat Sheet: веб-сайт. URL: <https://www.shopify.com/partners/shopify-cheat-sheet> (дата звертання 05.04.2022);
9. Vue.js Tutorial – A Guide on Prototyping Web Apps: веб-сайт. URL: <https://www.shopify.com/partners/blog/vuejs-tutorial> (дата звертання 01.05.2022);
10. Shopify Liquid code examples: веб-сайт. URL: <https://shopify.github.io/liquid-code-examples/?category=navigation> (дата звертання 19.05.2022);

ДОДАТОК А

Ініціалізація Vue компоненту Product

```
import { createApp } from 'vue';
import Product from "../view/Product";
import { store } from "../stores/product";
import VueTippy, { TippyComponent } from "vue-tippy";

import { ACTION_INIT_SETTINGS } from '../stores/constants'

function ProductInit() {
  this.selectors = {
    productWrapper: "#product-wrapper",
  }

  this.productWrapper = document.querySelector(this.selectors.productWrapper);

  if (!this.productWrapper) {
    return null;
  }

  this.productInit();
}

ProductInit.prototype.productInit = function () {
  const productDataWrapper = document.querySelector('#ProductConfig');

  if (productDataWrapper) {
    this.productData = JSON.parse(productDataWrapper.innerHTML);
  }

  const app = createApp(Product);
  app.use(store);
  app.use(VueTippy, {
    theme: 'grey',
    interactive: true,
    zIndex: 14,
    allowHTML: true,
    offset: '0,12',
  });
  app.mount(this.productWrapper);

  store.dispatch(ACTION_INIT_SETTINGS, this.productData);
  app.config.devtools = true;
};

export default ProductInit;
```


ДОДАТОК Б

Допоміжні функції компонентів Vue

```
export function unescape(value) {
  const doc = new DOMParser().parseFromString(value, "text/html");

  return doc.documentElement.textContent;
}

export function imageSizedUrl(src, size) {
  if (!src) {
    if (theme.strings && theme.strings.hasOwnProperty('noImageLink')) {
      src = theme.strings.noImageLink;
    } else {
      return null;
    }
  }

  return window.slate.Image.getSizedImageUrl(src, size);
}

export function throttle(func, wait) {
  let args;
  let result;
  let thisArg;
  let timeoutId;
  let lastCalled = 0;

  function trailingCall() {
    lastCalled = new Date;
    timeoutId = null;
    result = func.apply(thisArg, args);
  }

  return function() {
    let now = new Date;
    lastCalled = lastCalled || new Date;
    let remain = wait - (now - lastCalled);

    args = arguments;
    thisArg = this;

    if (remain <= 0) {
      lastCalled = now;
      result = func.apply(thisArg, args);
    } else if (!timeoutId) {
      timeoutId = setTimeout(trailingCall, remain);
    }

    return result;
  };
}

export function setCookie(name, value, options = {}) {
  options = {
    path: '/',
    secure: false,
    ...options
  };

  if (options.expires instanceof Date) {
    options.expires = options.expires.toUTCString();
  }
}
```

```
let updatedCookie = encodeURIComponent(name) + "=" + encodeURIComponent(value);

for (let [key, value] of Object.entries(options)) {
  updatedCookie += ";" + key;

  if (value !== true) {
    updatedCookie += "=" + value;
  }
}

document.cookie = updatedCookie;
}

export function translate(value, params = {}) {
  try {
    const translation = value.split(".").reduce((translations, key) => {
      if (translations.hasOwnProperty(key)) {
        return translations[key];
      }

      throw new Error("Translation missed")
    }, theme.strings);

    return Object.keys(params).reduce((result= "", key) => {
      let regex = new RegExp(`{((\\s+)?(\\${key})(\\s+)?)} `, 'gm');

      return result.replace(regex, params[key]);
    }, unescape(translation));
  } catch (e) {}

  return `"${value}" translation missed`;
}
```

ДОДАТОК В

Схема налаштувань секції header

```
{% schema %}
{
  "name": "t:sections.header.name",
  "settings": [
    {
      "type": "select",
      "id": "color_scheme",
      "options": [
        {
          "value": "accent-1",
          "label": "t:sections.header.settings.color_scheme.options__1.label"
        },
        {
          "value": "accent-2",
          "label": "t:sections.header.settings.color_scheme.options__2.label"
        },
        {
          "value": "background-1",
          "label": "t:sections.header.settings.color_scheme.options__3.label"
        },
        {
          "value": "background-2",
          "label": "t:sections.header.settings.color_scheme.options__4.label"
        },
        {
          "value": "inverse",
          "label": "t:sections.header.settings.color_scheme.options__5.label"
        }
      ],
      "default": "background-1",
      "label": "t:sections.header.settings.color_scheme.label"
    },
    {
      "type": "image_picker",
      "id": "logo",
      "label": "t:sections.header.settings.logo.label"
    },
    {
      "type": "range",
      "id": "logo_width",
      "min": 50,
      "max": 250,
      "step": 10,
      "default": 100,
      "unit": "t:sections.header.settings.logo_width.unit",
      "label": "t:sections.header.settings.logo_width.label"
    },
    {
      "type": "select",
      "id": "logo_position",
      "options": [
        {
          "value": "middle-left",
          "label": "t:sections.header.settings.logo_position.options__1.label"
        },
        {
          "value": "top-left",
          "label": "t:sections.header.settings.logo_position.options__2.label"
        }
      ],
    }
  ],
}
```

```
    {
      "value": "top-center",
      "label": "t:sections.header.settings.logo_position.options__3.label"
    }
  ],
  "default": "middle-left",
  "label": "t:sections.header.settings.logo_position.label",
  "info": "t:sections.header.settings.logo_position.info"
},
{
  "type": "link_list",
  "id": "menu",
  "default": "main-menu",
  "label": "t:sections.header.settings.menu.label"
},
{
  "type": "checkbox",
  "id": "show_line_separator",
  "default": true,
  "label": "t:sections.header.settings.show_line_separator.label"
},
{
  "type": "checkbox",
  "id": "enable_sticky_header",
  "default": true,
  "label": "t:sections.header.settings.enable_sticky_header.label",
  "info": "t:sections.header.settings.enable_sticky_header.info"
}
]
}
{% endschema %}
```