

**Міністерство освіти і науки України**  
**Сумський державний університет**  
**Навчально-науковий інститут бізнесу, економіки та менеджменту**  
**Кафедра економічної кібернетики**

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ Віталія КОЙБІЧУК  
(підпис) (Ім'я та ПРІЗВИЩЕ)

\_\_\_\_\_ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття освітнього ступеня бакалавр  
(бакалавр / магістр)

зі спеціальності \_\_\_\_\_ 051 Економіка \_\_\_\_\_  
(код та назва)

освітньо-професійної \_\_\_\_\_ програми \_\_\_\_\_ Економічна кібернетика \_\_\_\_\_  
(освітньо-професійної / освітньо-наукової) (назва програми)

на тему: Розробка веборієнтованої інформаційної системи для магазину одягу

Здобувача групи \_\_\_\_\_ ЕК-91a \_\_\_\_\_ Діденка Дмитра Олексійовича \_\_\_\_\_  
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

\_\_\_\_\_ Дмитро ДІДЕНКО \_\_\_\_\_  
(підпис) (Ім'я та ПРІЗВИЩЕ здобувача)

Керівник \_\_\_\_\_ доцент, к.ф.-м.н., Світлана КОЛОМІЄЦЬ \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ) (підпис)

Суми – 2023

Анотація  
кваліфікаційної роботи бакалавра на тему  
«РОЗРОБКА ВЕБОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ  
МАГАЗИНУ ОДЯГУ»

Студента Діденко Дмитро Олексійович  
(прізвище, ім'я, по батькові)

Актуальність теми, обраної для дослідження, визначається розвитком Інтернет-технологій та зростанням популярності онлайн-шопінгу. Все більше магазинів одягу переходять до віртуального простору. В зв'язку з цим, розробка веборієнтованої інформаційної системи для магазину одягу є актуальною та необхідною умовою конкурентоспроможності підприємства.

Метою роботи є розробка веборієнтованої інформаційної системи для магазину одягу.

Об'єктом дослідження є веборієнтована інформаційна система для магазину одягу.

Предметом дослідження є процес розробки веборієнтованої інформаційної системи для магазину одягу.

Задачами дослідження є: аналіз вимог клієнтів до веб-сайту магазину одягу; формулювання функціональних вимог до веборієнтованої інформаційної системи магазину одягу; розробка дизайну інформаційної системи; реалізація програмного коду інформаційної системи з використанням сучасних технологій веб-розробки; тестування та впровадження інформаційної системи в роботу магазину одягу.

Основним результатом кваліфікаційної роботи є: розробка та апробація прототипу веборієнтованої інформаційної системи для магазину одягу з використанням платформи WordPress. Отриманий результат може бути використаний для подальшого розвитку та вдосконалення інформаційних систем.

Ключові слова: веборієнтована інформаційна система, електронна комерція, магазин одягу, WordPress, Redux, WooCommerce.

Зміст кваліфікаційної роботи викладено на 56 сторінках. Список використаних джерел із 45 найменувань, розміщений на 4 сторінках. Робота містить 35 рисунків, а також 2 додатки, розміщений на 6 сторінках.

Рік виконання кваліфікаційної роботи – 2023 рік.

Рік захисту роботи – 2023 рік

Міністерство освіти і науки України  
Сумський державний університет  
Навчально-науковий інститут бізнесу, економіки та менеджменту  
Кафедра економічної кібернетики

ЗАТВЕРДЖУЮ

Завідувач кафедри

к.е.н., доцент

\_\_\_\_\_ Віталія КОЙБІЧУК

“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ  
(спеціальність 051 Економіка «Економічна кібернетика», «Бізнес аналітика»)

студенту 4 курсу, групи ЕК-91а

Діденку Дмитру Олексійовичу

(прізвище, ім'я, по батькові студента)

1. Тема роботи Розробка веборієнтованої інформаційної системи для магазину одягу /Development of a Web-oriented Snformation System for a Clothing Store

затверджена наказом по університету від « 23 » 05 2023 року № 0554-VI

2. Термін подання студентом закінченої роботи «16» червня 2023 року

3. Мета кваліфікаційної роботи розробка веборієнтованої інформаційної системи для магазину одягу

4. Об'єкт дослідження веборієнтована інформаційна система для магазину одягу.

5. Предмет дослідження процес розробки веборієнтованої інформаційної системи для магазину одягу

6. Кваліфікаційна робота виконується на матеріалах \_\_\_\_\_

7. Орієнтовний план кваліфікаційної роботи, терміни подання розділів керівникові та зміст завдань для виконання поставленої мети

Розділ 1 Аналіз сучасного стану та особливостей ринку продажу одягу

(назва – термін подання)

У розділі 1 провести огляд основних трендів у сфері онлайн-торгівлі; визначити переваги та недоліки наявних рішень; сформулювати технічні вимоги до веборієнтованої системи; визначити структуру програмування та функціональні можливості системи

Розділ 2 Проектування архітектури та інтерфейсу користувача

((назва – термін подання)

У розділі 2 розробити дизайн веб-сторінок у figma; визначити технології для реалізації системи; провести верстку та інтеграцію з CMS Wordpress; налаштувати плагіни redux, woocommerce; провести тестування веборієнтованої системи.

(зміст конкретних завдань до розділу, які має виконати студент)

### 8. Консультації з роботи:

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1			
2			
3			

### 9. Дата видачі завдання: «03»квітня 2023 року

Керівник кваліфікаційної роботи

\_\_\_\_\_

(підпис)

С.В. Коломієць

(ініціали, прізвище)

Завдання до виконання одержав

\_\_\_\_\_

(підпис)

Д.О. Діденко

(ініціали, прізвище)

## ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ТА ОСОБЛИВОСТЕЙ РИНКУ ПРОДАЖУ ОДЯГУ .....	8
1.1 Огляд основних трендів у сфері онлайн-торгівлі .....	8
1.2 Аналіз конкурентів на ринку .....	10
1.3 Визначення переваг та недоліків наявних рішень .....	13
1.4 Формулювання технічних вимог до системи .....	16
1.5 Структура програмування та визначення необхідних функціональних можливостей системи .....	19
РОЗДІЛ 2. ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА ІНТЕРФЕЙСУ КОРИСТУВАЧА	22
2.1 Розробка дизайну веб-сторінок у figma.....	22
2.2 Вибір технологій для реалізації системи .....	26
2.3 Верстка та інтеграція з CMS Wordpress .....	27
2.4 Налаштування плагінів redux, woocommerce .....	34
2.5 Тестування та налагодження системи .....	45
ВИСНОВКИ .....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	48
ДОДАТКИ .....	52

## ВСТУП

В сучасному світі поступовий розвиток інформаційних технологій перетворив їх на невід'ємну складову сфери торгівлі. Застосування веб-технологій у торгівлі набуває особливої актуальності, оскільки воно дозволяє збільшити обсяг продажів, привернути нових клієнтів та підвищити рівень обслуговування. У зв'язку з цим, розробка веборієнтованої інформаційної системи для магазину одягу стає надзвичайно важливою задачею.

Метою даного дослідження є розробка веборієнтованої інформаційної системи для магазину одягу.

Об'єктом дослідження є веборієнтована інформаційна система для магазину одягу.

Предметом дослідження є процес розробки веборієнтованої інформаційної системи для магазину одягу.

Мета дослідження обумовила завдання:

1. Проаналізувати потреби магазину одягу та визначити функціональні вимоги до веборієнтованої інформаційної системи для магазину одягу.
2. Розробити архітектуру та дизайн інформаційної системи.
3. Реалізувати програмний код інформаційної системи з використанням сучасних технологій веб-розробки.
4. Провести тестування та впровадження інформаційної системи в роботу магазину одягу.

Дослідження має велике значення для вирішення проблеми управління бізнес-процесами магазину одягу та поліпшення рівня обслуговування клієнтів. Шляхом розробки веборієнтованої інформаційної системи можна ефективно керувати всіма аспектами бізнесу та забезпечити найвищу якість обслуговування клієнтів. Це допоможе магазину зайняти сильну позицію на ринку та задовольнити потреби своїх клієнтів з високим рівнем задоволення.

## РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ТА ОСОБЛИВОСТЕЙ РИНКУ ПРОДАЖУ ОДЯГУ

### 1.1 Огляд основних трендів у сфері онлайн-торгівлі

Сучасний розвиток інтернет-торгівлі став однією з найбільш активно розвиваючихся галузей економіки. Згідно зі статистикою Hootsuite, кількість інтернет-користувачів постійно зростає, що сприяє розширенню онлайн-торгівлі (рисунок 1.1). Особливо зростає ринок онлайн-продажу одягу, що пов'язано зі зміною способу життя і зростанням кількості людей, які вибирають онлайн-шопінг. Технологічні інновації також сприяють цьому розвитку, забезпечуючи зручність та швидкість покупок в Інтернеті [29, с.64].

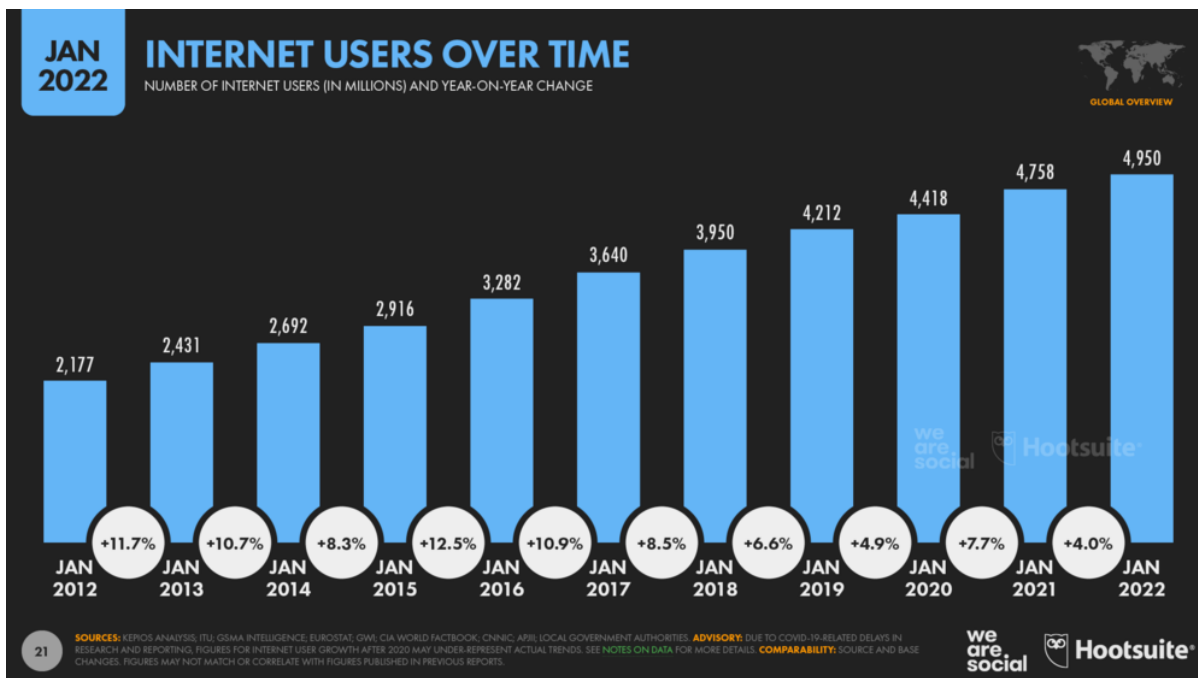


Рисунок 1.1 – Графік зростання інтернет-користувачів у останні роки.

Один з основних трендів у сфері онлайн-торгівлі - це перехід від комп'ютерів до мобільних пристроїв. За даними Meltwater, більше половини інтернет-користувачів



вже здійснюють покупки через мобільні пристрої. Тому компанії, які пропонують послуги онлайн-торгівлі, повинні адаптувати свої веб-сайти до мобільних пристроїв, щоб зберігати свої позиції на ринку (рисунок 1.2) [29, с.14]..

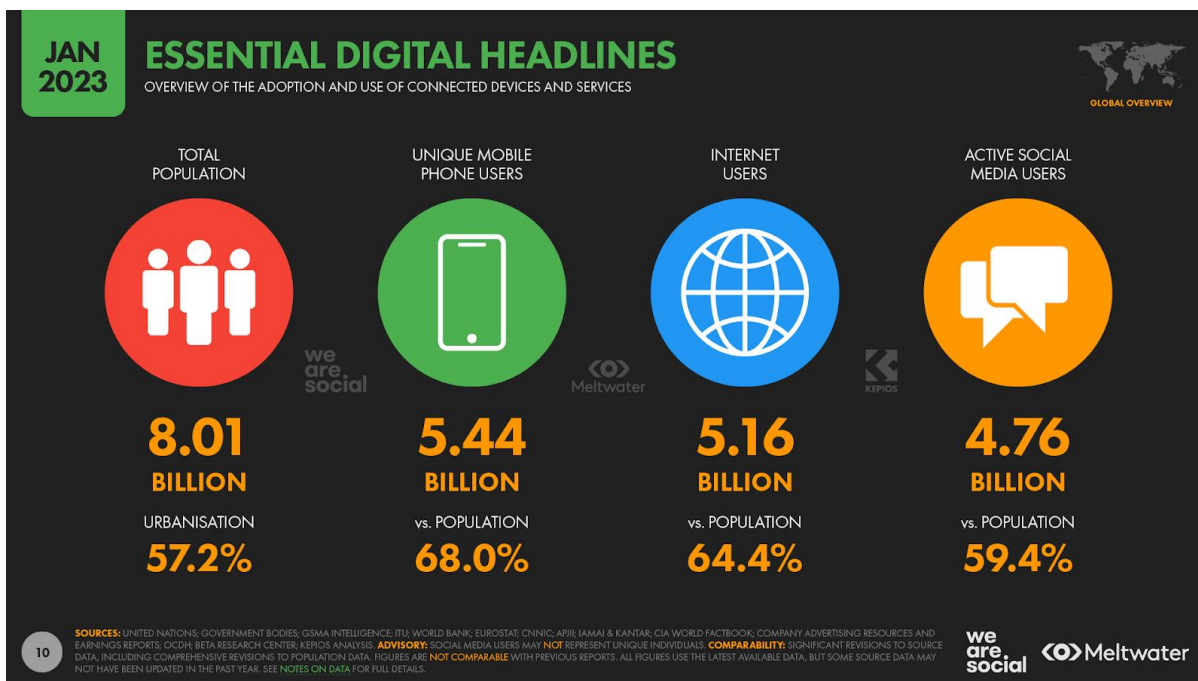


Рисунок 1.2 – Зростання кількості користувачів мобільних пристроїв

Для кращого розуміння сучасного стану ринку продажу одягу, важливо враховувати деякі ключові факти про світову популяцію та використання Інтернету та мобільних технологій:

1. Населення планети станом на початок 2023 року складає вже понад 8 мільярдів людей.
2. Понад 57% населення — жителі міст.
3. Мобільними телефонами сьогодні користуються близько 5,44 млрд осіб (68% всього населення).
4. Кількість користувачів інтернету — близько 5,16 мільярда осіб (64,4% населення планети).
5. За останні 12 місяців кількість інтернет-користувачів зросла на 1,9%.
6. Понад 4,76 мільярда людей користуються соціальними мережами на регулярній основі.

Підходом, який також спостерігається на ринку, є перехід від різних платформ до єдиного майданчика продажу. Більшість компаній, які продають товари в Інтернеті, пропонують їх на різних веб-сайтах, таких як, Amazon, eBay та інші. Однак, цей підхід вимагає значних затрат на підтримку і управління декількома майданчиками, що залежить від провайдерів послуг і може впливати на стабільність системи в цілому. Крім того, клієнтам необхідно реєструватись на кожному сайті окремо, що може знизити рівень їх задоволення від процесу покупки [19, с.42]..

Один з найбільш ефективних рішень в сфері онлайн-торгівлі - використання веборієнтованих інформаційних систем. Ці системи дозволяють об'єднати весь асортимент товарів на одному веб-сайті, що надає клієнтам легкий доступ до повної інформації про товари і можливість замовлення безпосередньо на сайті магазину. Важливою перевагою є також те, що всі дані про товари зберігаються в одній базі даних, що дозволяє їх зручну обробку та аналіз [5, с.14-79]..

## 1.2 Аналіз конкурентів на ринку

Під час розробки веб-орієнтованої інформаційної системи важливо провести аналіз конкурентів на ринку. Такий аналіз допоможе визначити основні переваги та недоліки конкурентів, а також знайти можливості для покращення нашого продукту. Один з ключових аспектів аналізу конкурентів - це вивчення їхньої присутності в онлайн-середовищі та рівня конкуренції, яку вони представляють.

На зображенні (рисунок 1.3) представлений графік, який ілюструє зростання обсягу електронної комерції в сфері продажу одягу в Україні. Графік показує позитивну динаміку росту продажів протягом останніх років, що свідчить про потенціал ринку. З урахуванням позитивних тенденцій росту електронної комерції і конкурентного середовища, мій проект має великі можливості для успішної діяльності та здобуття стабільної позиції [7, с.58]..

- Одяг – другий за обсягом і рівнем розвитку ринок електронної комерції в Україні після електроніки.
- Обсяг електронної комерції одягом в Україні зростає у середньому на 26% з 2016 року і досягнув \$291млн у 2020.
- Частка електронної комерції у роздрібній торгівлі одягом у 2020 році становила 6.8%.

ТОП-10 онлайн-магазинів та класифайдів у сфері одягу за обсягом трафіку в 2020 році:



Джерело: дані Semrus, аналіз Soul Partners  
\*Кількість візитів сайту; не враховує відвідуваність мобільних додатків



Джерело: дані Euromonitor, аналіз Soul Partners

Ключові показники електронної комерції одягом, 2020

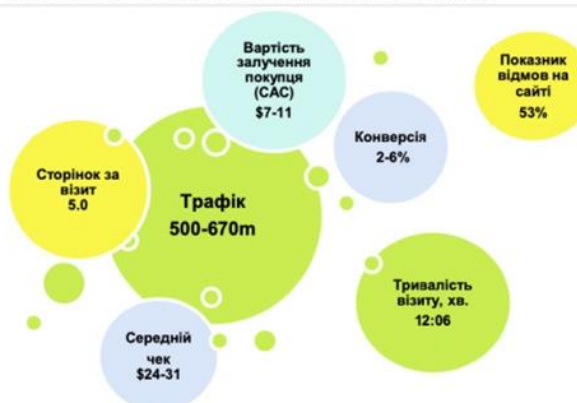


Рисунок 1.3 – Обсяг електронної комерції одягом в Україні

На сьогоднішній день на ринку продажу одягу діє багато конкурентів. Основними з них є такі відомі магазини, як Shafa, Zara, H&M, Mango, Kasta, та інші. Крім того, є багато малих та середніх магазинів, які працюють у різних сегментах ринку.

Давайте розглянемо детальніше кожного з них.

1. Rozetka - це один з найбільших українських інтернет-магазинів, який пропонує великий вибір одягу різних брендів та категорій. У Rozetka є можливість оглядати товари з різних кутів, що дозволяє більш детально розглянути деталі. Крім того, у магазині часто проводяться різні акції та знижки на товари. Однак, у Rozetka відсутня можливість придбання товарів в оптовому порядку та створення власного бренду.
2. ModnaKasta - це магазин, який пропонує ексклюзивні моделі одягу та аксесуарів за доступними цінами. Крім того, у ModnaKasta діє програма лояльності, що дозволяє отримувати знижки та подарунки за кожен придбаний товар. Однак, в

магазині недостатньо великий вибір товарів та відсутність можливості створення власного бренду.

3. Lamoda - це один з найбільших онлайн-магазинів одягу в Східній Європі з більш ніж 20 мільйонами зареєстрованих користувачів і більш ніж 3 мільйонами замовлень на місяць. Lamoda пропонує широкий асортимент товарів, який включає як місцеві бренди, так і відомі світові.

Щоб успішно конкурувати на ринку, наш магазин може скористатися наступними стратегіями:

1. Інноваційність: Введення нових інноваційних рішень та використання передових технологій допоможуть підвищити рівень задоволеності клієнтів і забезпечити конкурентну перевагу.
2. Широкий асортимент товарів: Підбір широкого асортименту товарів, включаючи різні бренди та категорії, дозволить привернути більше клієнтів і задовольнити різноманітні потреби споживачів.
3. Програми лояльності: Впровадження програм лояльності для клієнтів, де вони можуть отримувати знижки, бонуси або подарунки за покупки, сприятиме залученню нових клієнтів і збереженню існуючих.
4. Оптові закупівлі: Надання можливості оптових закупівель для клієнтів, які бажають придбати товари великими партіями, може привернути бізнес-клієнтів та забезпечити стабільні замовлення.
5. Власний бренд: Розгляньте можливість створення власного бренду, який буде ексклюзивним для нашого магазину. Це дозволить вирізнитися серед конкурентів і залучити клієнтів, які шукають унікальні товари.

Враховуючи аналіз конкурентів та ринкові можливості, важливо розробити унікальну стратегію та пропозицію магазину, які привернуть увагу клієнтів та дозволять зайняти стабільну позицію на ринку продажу одягу в Україні.

### 1.3 Визначення переваг та недоліків наявних рішень

При розробці також важливо визначити переваги та недоліки наявних рішень у даній сфері.

Розробка власного інтернет-магазину може надати повний контроль над процесом продажу та можливість використовувати власну брендovanу платформу. Проте, цей підхід може бути витратним і часомістким, а також вимагає наявності фахівців у галузі програмування та дизайну.

Використання готових платформ, таких як Shopify, WooCommerce або Magento, може бути швидшим варіантом, оскільки базовий функціонал вже наявний і може бути використаний з мінімальними змінами. Проте, використання готових платформ може обмежити можливості розширення функціоналу та розвитку індивідуальної стратегії продажу [29, с.22]..

Використання інтернет-майданчиків, таких як OLX, Prom.ua, Rozetka, також може бути зручним варіантом, оскільки вони надають доступ до великого клієнтського базису та маркетингової підтримки. Але варто враховувати, що це обмежує можливості індивідуальної настройки та розвитку бренду.

Враховуючи ці рішення, важливо аналізувати конкретні потреби бізнесу, його фінансові можливості, а також ринкові умови та характеристики продукту. Це допоможе визначити оптимальний підхід і зробити вибір, який задовольнятиме ваші потреби та сприятиме успіху вашого магазину.

Огляд існуючих рішень на ринку дозволяє виявити кілька переваг, які можуть бути використані при розробці:

1. Високий рівень автоматизації процесів: Більшість наявних систем мають високий рівень автоматизації, що дозволяє ефективно обробляти замовлення та знижувати витрати на робочу силу. Це сприяє швидкій обробці замовлень і забезпечує більш ефективне функціонування магазину.
2. Висока доступність: Багато існуючих систем працюють у режимі 24/7, що дозволяє клієнтам робити замовлення в будь-який зручний для них час. Це

забезпечує зручність та доступність обслуговування для клієнтів, незалежно від їх географічного розташування та режиму роботи.

3. Широкий асортимент товарів: Більшість існуючих систем пропонують широкий вибір товарів з різних категорій. Це дає змогу задовольнити різні потреби та вподобання різних груп клієнтів, що сприяє збільшенню продажів та розширенню клієнтської бази.
4. Зручність користування: Більшість існуючих систем мають простий та інтуїтивно зрозумілий інтерфейс, що сприяє легкості використання для клієнтів. Це робить процес замовлення та пошуку потрібного товару більш зручним та ефективним.

Незважаючи на переваги, існують також деякі недоліки, які можна врахувати при розробці нової системи:

1. Високі витрати на розробку та підтримку системи: Розробка та підтримка власної системи можуть бути дорогими. Це особливо може бути проблематичним для невеликих компаній з обмеженим бюджетом.
2. Невідповідність системи потребам клієнтів: Незважаючи на широкий асортимент товарів, існуючі системи не завжди мають всі товари, які потрібні конкретному клієнту. Це може обмежувати можливості продажу конкретних товарів або задоволення потреб певних сегментів клієнтів.
3. Висока конкуренція: Ринок онлайн-торгівлі одягом є насиченим, що створює значну конкуренцію для нових учасників. Щоб привернути та утримати клієнтів, потрібно пропонувати високоякісний продукт, відповідати сучасним тенденціям у дизайні та стилі, а також забезпечувати відмінний сервіс для клієнтів.

При розробці нової системи варто уважно вивчити ці переваги та недоліки, а також врахувати потреби та можливості вашого бізнесу для вибору оптимального шляху розвитку [36, с.32-90]..

Регулятивна база для онлайн-магазинів з продажу одягу є важливим аспектом, який потрібно враховувати для створення стійкої та безпечної електронної комерції. Наразі в Україні відсутня чітка регулятивна база, специфічно спрямована на онлайн-

магазини з продажу одягу. Однак, відбуваються постійні зусилля з боку уряду та регуляторних органів для розробки та впровадження відповідних нормативних актів.

Поки що, онлайн-магазини в Україні можуть використовувати загальні законодавчі акти, такі як Закон України "Про електронну комерцію", щоб регулювати свою діяльність. Цей закон визначає загальні принципи електронної комерції, включаючи права та обов'язки сторін угоди, захист споживачів та інші питання. Однак, він не має спеціальних положень, які б безпосередньо стосувалися онлайн-магазинів з продажу одягу [10, с.34]..

Для забезпечення законності та захисту прав споживачів у сфері онлайн-торгівлі одягом, рекомендується дотримуватися наступних принципів:

1. Чесність та прозорість: Онлайн-магазини повинні надавати точну та чітку інформацію про свої товари, включаючи ціни, розміри, склад тканини та інші важливі характеристики. Також вони повинні бути відкриті щодо політики повернення товару, обміну та гарантій.
2. Захист персональних даних: Онлайн-магазини повинні дотримуватися вимог щодо захисту персональних даних своїх клієнтів. Це означає, що вони повинні забезпечувати безпечне зберігання та обробку особистої інформації, а також повідомляти клієнтів про способи використання їх даних.
3. Гарантії якості: Онлайн-магазини повинні надавати якісні товари, що відповідають заявленій інформації. Якщо товар має дефекти або не відповідає опису, споживачі повинні мати можливість повернути його та отримати повернення коштів або обмін на новий товар.
4. Система зворотного зв'язку: Онлайн-магазини повинні мати доступну систему зворотного зв'язку, через яку клієнти можуть звертатися зі своїми питаннями, скаргами або пропозиціями. Швидка та професійна відповідь на звернення клієнтів є важливим елементом задоволення їх потреб.
5. Правила повернення та обміну: Онлайн-магазини повинні мати чіткі правила та процедури повернення та обміну товару. Споживачам потрібно бути повідомленими про умови повернення, строки та вимоги щодо повернення товару.

6. Безпека платежів: Онлайн-магазини повинні забезпечувати безпеку платежів та конфіденційність фінансової інформації своїх клієнтів. Вони повинні використовувати захищені платіжні системи та шифрування даних для запобігання шахрайству та крадіжкам.

Уряд та регуляторні органи в Україні активно працюють над розробкою та впровадженням спеціальних правил та стандартів для онлайн-торгівлі. Законодавча база постійно змінюється, доповнюється для забезпечення захисту прав споживачів та створення конкурентних умов на ринку. Це може включати визначення правил повернення товару, захисту персональних даних, відповідальності за продаж фальсифікованих товарів та інші питання, що стосуються онлайн-торгівлі [29, с.24-72].

#### 1.4 Формулювання технічних вимог до системи

Формулювання технічних вимог до системи є ключовим етапом у процесі розробки. Основні вимоги, які потрібно враховувати, включають функціональні вимоги, вимоги до безпеки, вимоги до продуктивності та вимоги до інтерфейсу користувача.

*Функціональні вимоги:*

1. Реєстрація користувачів: Система повинна надати можливість реєстрації нових користувачів, зберігання їх особистих даних та забезпечення авторизації.
2. Управління каталогом товарів: Система повинна забезпечувати можливість додавання, редагування та видалення товарів у каталозі, а також їх категоризацію.
3. Оформлення замовлення: Користувачі повинні мати можливість додавати товари до кошика, оформляти замовлення, вказувати адресу доставки та здійснювати оплату [40, с.24].



4. Управління замовленнями: Система повинна забезпечувати відстеження стану замовлень, оновлення їх статусу та комунікацію з клієнтами.
5. Адміністрування: Система повинна мати можливість керування ролями та правами користувачів, враховувати історію замовлень та особисті дані користувачів.

*Вимоги до безпеки:*

1. Захист особистих даних: Система повинна забезпечувати конфіденційність та цілісність особистих даних користувачів, а також захист від несанкціонованого доступу до цих даних.
2. Захист від атак: Система повинна мати вбудовані заходи безпеки, щоб захистити від хакерських атак, таких як SQL-ін'єкції, кросс-сайтові скрипти та інші вразливості.
3. Захист платежів: Система повинна забезпечувати безпечність платежних транзакцій та захист від шахрайських дій [3029, с.27].

*Вимоги до продуктивності:*

1. Швидкість відгуку: система повинна бути швидкою та ефективною в роботі, забезпечуючи швидке завантаження сторінок та операцій з базою даних.
2. Масштабованість: система повинна мати можливість масштабування для впорядкування зростаючого обсягу даних та трафіку.

*Вимоги до інтерфейсу користувача:*

1. Зручність використання: система повинна мати зрозумілий та зручний інтерфейс для користувачів з різним рівнем технічної компетенції.
2. Адаптивний дизайн: система повинна коректно відображатися на різних пристроях та розмірах екранів.

Формулювання цих технічних вимог є важливим для розробки інформаційної системи, яка задовольнятиме потреби магазину одягу та його клієнтів.

Однією з важливих технічних вимог є сумісність системи з різними операційними системами, такими як Windows, macOS та Linux, а також з різними веб-браузерами, такими як Chrome, Firefox, Safari та Internet Explorer. Це забезпечить

доступ користувачів до системи незалежно від їхньої платформи та веб-браузера, що сприятиме зручності та доступності [18, с.22]..

Також, важливим аспектом є масштабованість системи. Оскільки магазин одягу може мати велику кількість товарів та користувачів, система повинна бути здатною працювати з великим обсягом даних та обробляти одночасні запити високої навантаженості. Це допоможе запобігти можливим перебоям у роботі системи та забезпечити швидку та ефективну обробку даних.

Для забезпечення безпеки даних та конфіденційності, система повинна мати відповідні заходи захисту, такі як шифрування даних, контроль доступу, захист від несанкціонованого доступу та резервне копіювання даних. Це дозволить зберегти конфіденційну інформацію про клієнтів та забезпечити їхню довіру до системи [29, с.10]..

Наслідки витоку даних для компаній:

Витоки даних зачіпають не лише тих, чий дані було вкрадено, але й сторони, яким було довірено збереження цих даних. Компанії, які стали жертвами витоку даних, можуть зіткнутися з такими наслідками:

1. Юридичні проблеми: Витік даних може призвести до правових проблем для компанії, зокрема до порушень законодавства про захист персональних даних. Це може призвести до великих штрафів та погіршення репутації компанії.
2. Втрата довіри клієнтів: Клієнти можуть втратити довіру до компанії після витоку їхніх особистих даних. Це може призвести до втрати клієнтів та зменшення обсягу бізнесу.
3. Фінансові втрати: Витік даних може призвести до фінансових втрат, які пов'язані з відшкодуванням збитків клієнтам, судовими витратами та відновленням системи.
4. Втрата конкурентної переваги: Витік конфіденційної інформації, такої як плани розробки продуктів або бізнес-стратегії, може призвести до втрати конкурентної переваги перед іншими компаніями.

Для забезпечення стабільності та надійності роботи системи, важливо враховувати фактори, які можуть впливати на її функціонування. Наприклад, система

повинна мати механізми контролю помилок, обробки виняткових ситуацій та відновлення після відмови. Також, система повинна підтримувати резервне копіювання даних та можливість відновлення після випадкової втрати даних.

Простота використання та зручний інтерфейс користувача є ще одним важливим аспектом. Система повинна мати інтуїтивно зрозумілий інтерфейс, легку навігацію та зручність використання для різних категорій користувачів. Це дозволить забезпечити зручність покупок та позитивний досвід взаємодії з системою для користувачів [42, с.55-64]..

Отже, формулювання технічних вимог до системи допомагає забезпечити її ефективну та стабільну роботу, враховуючи потреби користувачів, вимоги проекту та сучасні технологічні стандарти.

## 1.5 Структура програмування та визначення необхідних функціональних можливостей системи

На етапі розробки структури програмування та визначення функціональних можливостей системи важливо дотримуватись кроків:

1. Аналіз вимог: Враховуйте технічні та функціональні вимоги, що були визначені на попередньому етапі. Це допоможе зрозуміти, які конкретні функції системи необхідно реалізувати.
2. Визначення архітектури системи: Виберіть загальну структуру системи та спосіб взаємодії компонентів. Розгляньте різні варіанти, такі як клієнт-серверна архітектура або мікросервісна архітектура, і оберіть найбільш підходящий для вашої системи.
3. Розподіл функціональності: Визначте, які функції та задачі будуть розподілені між різними компонентами системи. Наприклад, окремі модулі можуть

- відповідати за реєстрацію користувачів, керування каталогом товарів, оформлення замовлень та інші функціональні частини системи.
4. Визначення API та інтерфейсів: Залежно від обраної архітектури системи визначте, які API та інтерфейси будуть використовуватись для взаємодії між компонентами. Наприклад, RESTful API, SOAP-сервіси, WebSocket зв'язок та інші.
  5. Вибір технологій: Оберіть технології та інструменти, які найкраще підходять для розробки системи. Вибір мов програмування, фреймворків, баз даних та інших технологій залежить від потреб проекту та експертної оцінки.
  6. Розробка компонентів системи: Розпочніть розробку окремих компонентів системи відповідно до визначеної структури та використовуючи обрані технології. Кожен компонент має виконувати конкретну функцію або задачу системи.
  7. Інтеграція та тестування: Після розробки компонентів системи їх потрібно інтегрувати та протестувати, щоб переконатися, що вони правильно працюють та взаємодіють між собою [29, с.50]..

Важливо пам'ятати, що розробка структури програмування та визначення функціональних можливостей системи може бути ітеративним процесом. Після тестування та зворотного зв'язку можуть виникнути потреби у внесенні змін чи доповнень до структури та функціональності системи.

Під час розробки системи можуть виникати нові вимоги або змінюватися наявні. Наприклад, під час тестування можуть виявлятися недоліки або можливості для поліпшення функціоналу. Зворотний зв'язок від користувачів або клієнтів також може призвести до необхідності внесення змін у структуру програмування та функціональні можливості системи.

У такому ітеративному процесі розробки важливо планувати час і ресурси на внесення змін і доповнень, а також узгоджувати їх з зацікавленими сторонами та замовниками проекту. Регулярні зустрічі з командою розробників, тестувальниками та замовниками допоможуть виявляти нові потреби та ефективно управляти змінами.

Такий підхід дозволяє гнучко адаптуватися до змін, поліпшувати продукт протягом розробки та забезпечувати його відповідність потребам користувачів. В результаті, система буде більш функціональною, стабільною та задовольнятиме вимоги клієнтів у найкращому вигляді [12, с.64].

## РОЗДІЛ 2. ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА ІНТЕРФЕЙСУ КОРИСТУВАЧА

### 2.1 Розробка дизайну веб-сторінок у figma

Figma - це онлайн-інструмент для дизайну і прототипування, який дозволяє розробникам створювати інтерактивні прототипи, дизайнувати веб-сторінки та співпрацювати в команді. Цей інструмент дозволяє швидко створювати візуальні компоненти, вирівнювати їх, встановлювати зв'язки між сторінками та елементами, налаштовувати анімацію.

Процес розробки прототипів та дизайну веб-сторінок у Figma включає такі кроки:

1. Визначення вимог: Потрібно ретельно вивчити вимоги до інтерфейсу та веб-сторінок, зокрема, врахувати функціональні та нефункціональні вимоги, бізнес-потреби та очікування користувачів.
2. Створення скетчів: Першим кроком є створення простих малюнків або скетчів, де визначається загальна структура та розташування елементів на веб-сторінці. Це допомагає візуалізувати загальний вигляд та розміщення елементів.
3. Розробка прототипу: На основі скетчів розробляються прототипи у Figma. Створюються окремі сторінки та додаються елементи інтерфейсу, такі як кнопки, поля введення, меню, зображення тощо. Зв'язки між сторінками встановлюються для створення інтерактивності прототипу.
4. Дизайн веб-сторінок: Після розробки прототипів переходимо до детального дизайну веб-сторінок. Встановлюємо кольорову схему, шрифти, стилізуємо елементи інтерфейсу та вирівнюють їх. Важливо враховувати зручність використання та естетичний вигляд дизайну.

5. Анімація та взаємодія: У Figma можна налаштувати анімацію та взаємодію елементів, що дозволяє показати користувачам, як працює система. Наприклад, анімація кнопок, переходи між сторінками, показ повідомлень тощо.
6. Тестування та зворотній зв'язок: В кінці можна проводити тестування з користувачами, щоб отримати їх відгук. Зворотній зв'язок може вказати на можливі проблеми або поліпшення в інтерфейсі, що потребують уваги.

На основі інформації про майбутній сайт в Figma, який включає головну сторінку, магазин, про нас, контакти та корзину, формуємо наступні етапи процесу створення:

Аналіз та збір вимог:

Цей етап є важливим, оскільки він дозволяє отримати чітке розуміння вимог та очікувань замовника щодо функціональності та зовнішнього вигляду веб-сторінок. Він становить основу для подальшого проектування та розробки.

Проведений детальний аналіз ринку продажу одягу, досліджено конкурентів та їх діяльність, проаналізовано сильні та слабкі сторони, а також знайдено можливості для покращення нашого продукту.

Після цього сформовано чіткі вимоги до функціоналу сайту, врахувавши потреби та очікування клієнтів, їх зручність та задоволення під час користування. Вимоги включають такі функції, як пошук і фільтрація товарів, зручне оформлення замовлення, інформація про доставку та оплату, огляди товарів та інші важливі аспекти.

Інформаційна архітектура:

На цьому етапі розглядається, як буде організовано та представлено теоретичне завдання .

Створюємо у figma додаткові фрейми , щоб дотримуватись приблизної концепції під час розробки (Рисунок 2.1).

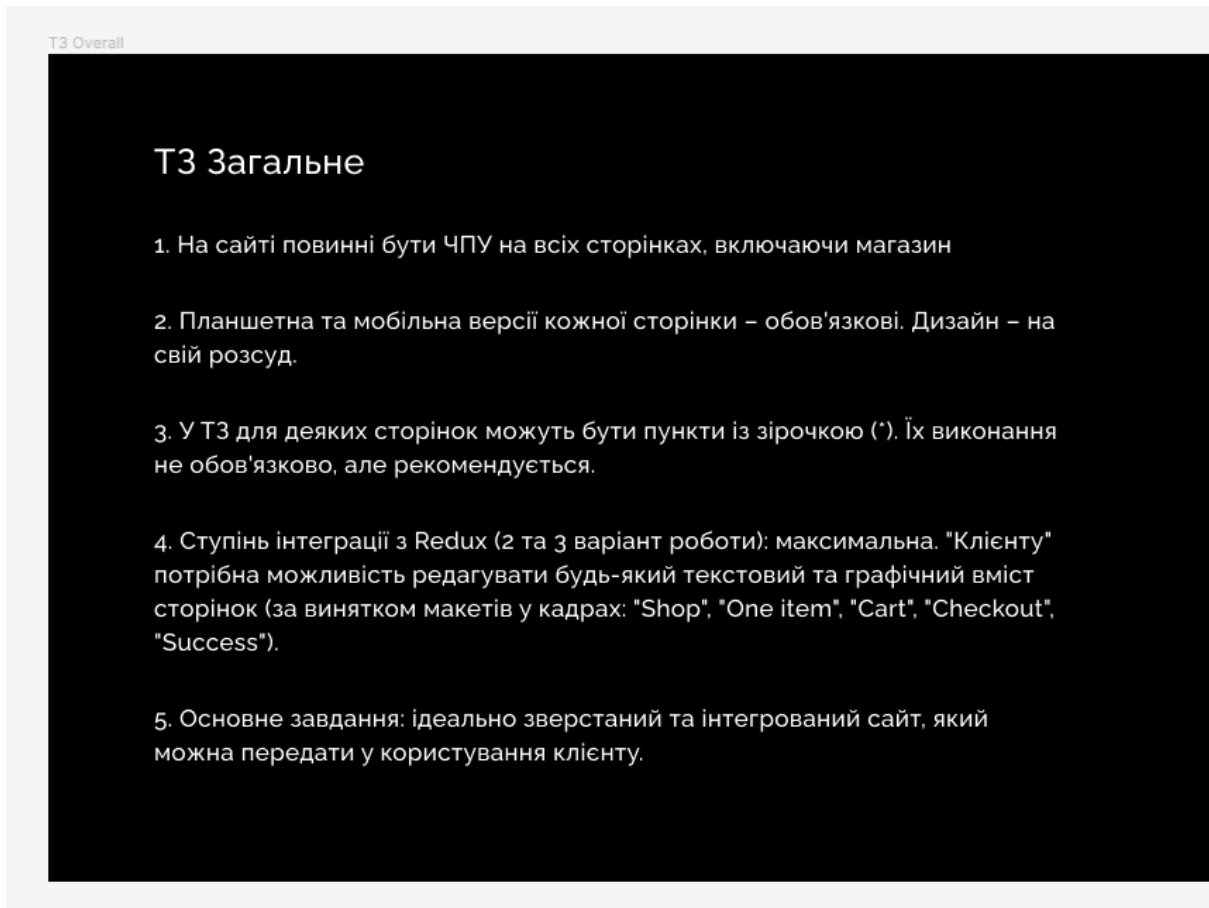


Рисунок 2.1 – ТЗ загальне

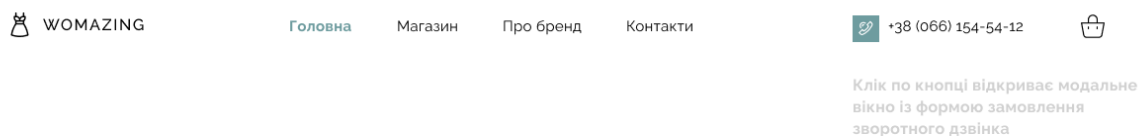
В результаті цього маємо чітке уявлення про інформаційну архітектуру майбутнього сайту і набір фреймів у figma, які слугують основою для подальшого розроблення компонентів та налаштування сайту. (Рисунок 2.2)



### У кошику щось є



### При наведенні на трубочку



### Звичайний вигляд При наводженні

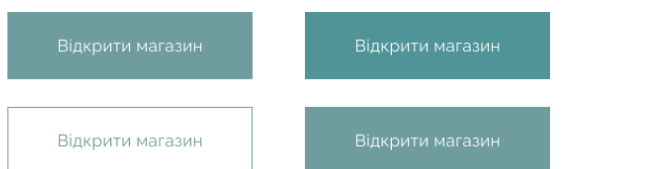


Рисунок 2.2 – Компоненти

### Створення прототипів:

Прототипи допомагають візуалізувати та перевірити концепцію веб-сторінок перед початком розробки. Вони дозволяють протестувати функціональність, навігацію та взаємодію з користувачами. Це важливий етап, щоб виявити можливі проблеми та зробити виправлення ще на ранній стадії розробки.

### Дизайн веб-сторінок:

Дизайн веб-сторінок включає в себе вибір кольорової палітри, шрифтів, компоновання елементів та інші естетичні аспекти. Важливо створити привабливий та зручний дизайн, який відповідає бренду компанії і задовольняє потреби користувачів.

Основна увага на розробці і візуальному оформленні раніше обговорених сторінок, таких як "Головна", "Магазин", "Про бренд", "Контакти" і "Кошик". Кожна сторінка мала свою унікальну мету та функціонал.

У figma для кожної сторінки створено окремий фрейм, де відображається дизайн цієї сторінки. Крім основного зображення сторінки, також додано додаткові фрейми з приблизним технічним завданням для верстки. Це допоможе при розробці

краще розуміти, яким чином потрібно розташовувати елементи на сторінці та як забезпечити їх взаємодію. (рисунок 2.3)

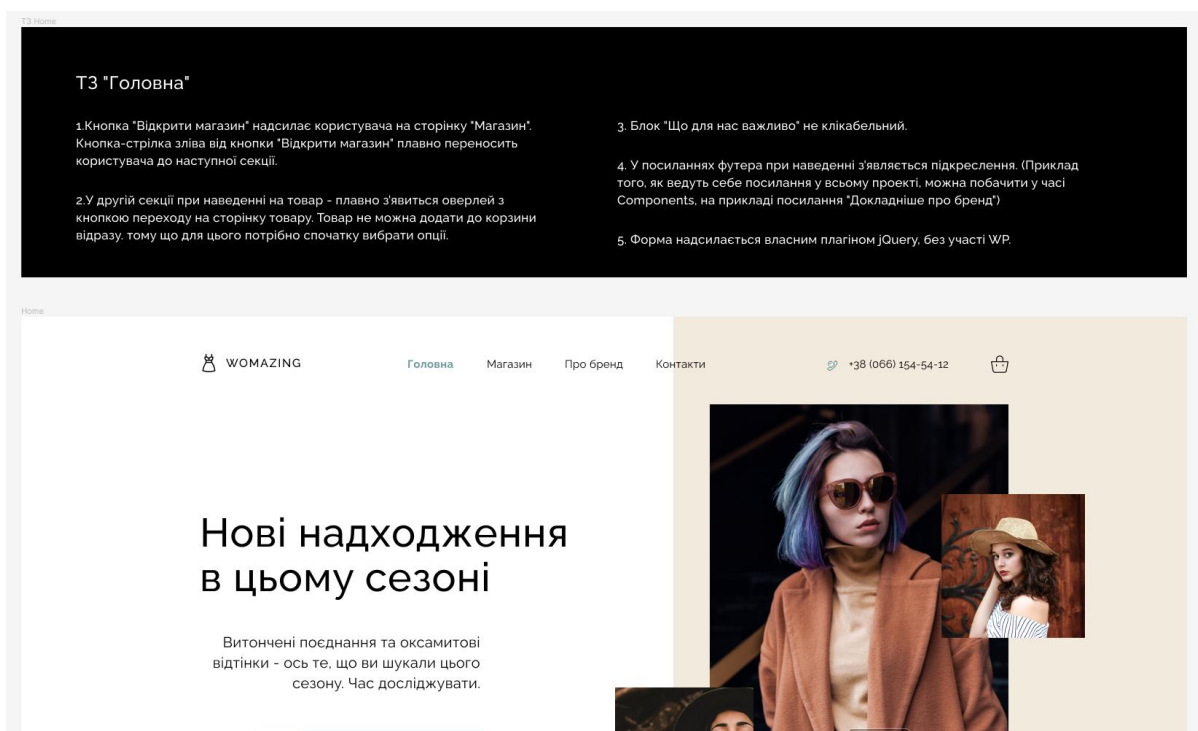


Рисунок 2.3 – Дизайн головної сторінки

## 2.2 Вибір технологій для реалізації системи

Враховуємо, як користувачі будуть спілкуватися з системою, які елементи і функції будуть доступні та як це буде організовано на веб-сторінках. Наприклад, можуть використовуватися такі методи взаємодії, як кнопки, посилання, форми, меню, фільтри тощо. Важливо створити зрозумілий та зручний інтерфейс.

Після визначення методів взаємодії з користувачем необхідно вибрати технології, які будуть використовуватись для реалізації системи. Це включає вибір мови програмування, фреймворків, бібліотек, баз даних та інших технологій, що відповідають технічному завданню. Важливо враховувати функціональні та нефункціональні вимоги до системи, а також розглянути ефективність,

масштабованість, безпеку та інші аспекти. Порівняння технологій допоможе знайти найкращий варіант, який задовольняє потреби проекту.

Врахування власних навичок та досвіду також має велике значення. Розробник повинен зрозуміти, наскільки він знайомий з певною технологією, яку він має компетенцію в цій сфері. Це допоможе забезпечити ефективну розробку та уникнути проблем, пов'язаних з нестачею знань або недостатнім ресурсним забезпеченням.

### 2.3 Верстка та інтеграція з CMS Wordpress

Графічний дизайн, створений на попередніх етапах, перетворено на реальні веб-сторінки, використовуючи HTML, CSS, JS та інші фронтенд технології.

Кожен елемент сторінок розміщений з максимальною точністю, форматування тексту налаштоване належним чином, зображення та медіа-елементи додані з метою створення привабливого вигляду. В роботі використовується редактор коду Visual Studio Code, який забезпечує зручність та продуктивність при програмуванні (рисунок 2.4).



Рисунок 2.4 – Visual Studio Code

Крім дизайну, було важливо встановити взаємозв'язки між сторінками, щоб забезпечити зручну навігацію для користувачів. Це включало створення посилань та кнопок, які дозволяли переходити між різними розділами та сторінками сайту. Щоб відвідувачі мали можливість з легкістю переходити до "Головної сторінки",

"Магазину", сторінки "Про бренд", "Контакти" та "Кошик" за допомогою інтуїтивно зрозумілих елементів навігації. (рисунок 2.5)

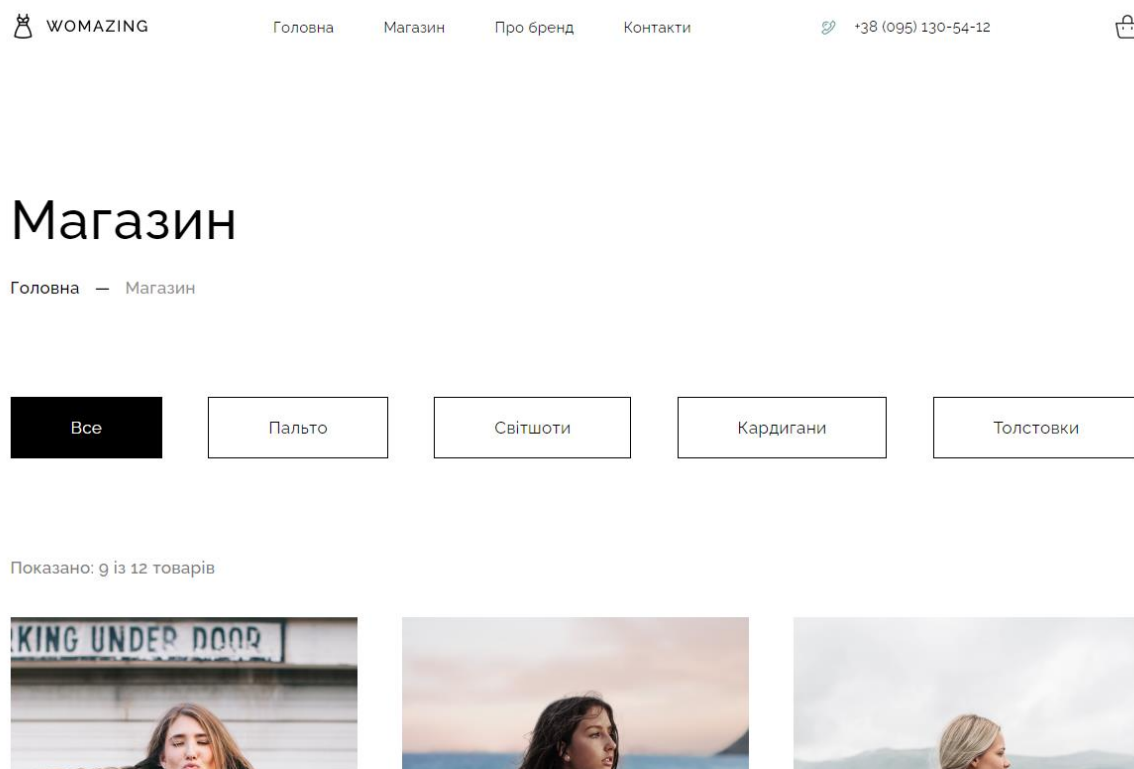


Рисунок 2.5 – Сторінка "Магазин"

Особлива увага приділяється адаптивності. Враховуються різні розміри екранів, включаючи планшетні і мобільні версії. Мета полягає в забезпеченні оптимального відображення контенту на будь-якому пристрої, зберігаючи при цьому дизайн і зручність використання. Таким чином, користувачі зможуть зручно переглядати і взаємодіяти з сайтом незалежно від пристрою, яким вони користуються.

Під час роботи з CMS WordPress приділяється особлива увага створенню та інтеграції теми, щоб забезпечити унікальний вигляд та функціональність веб-сайту. Для швидкої розробки теми користуємось шаблоном underscores (<https://underscores.me/>), що дозволяє швидше завантажувати тему та зменшити зусилля, які необхідно витратити на створення файлів з нуля. (рисунок 2.6)

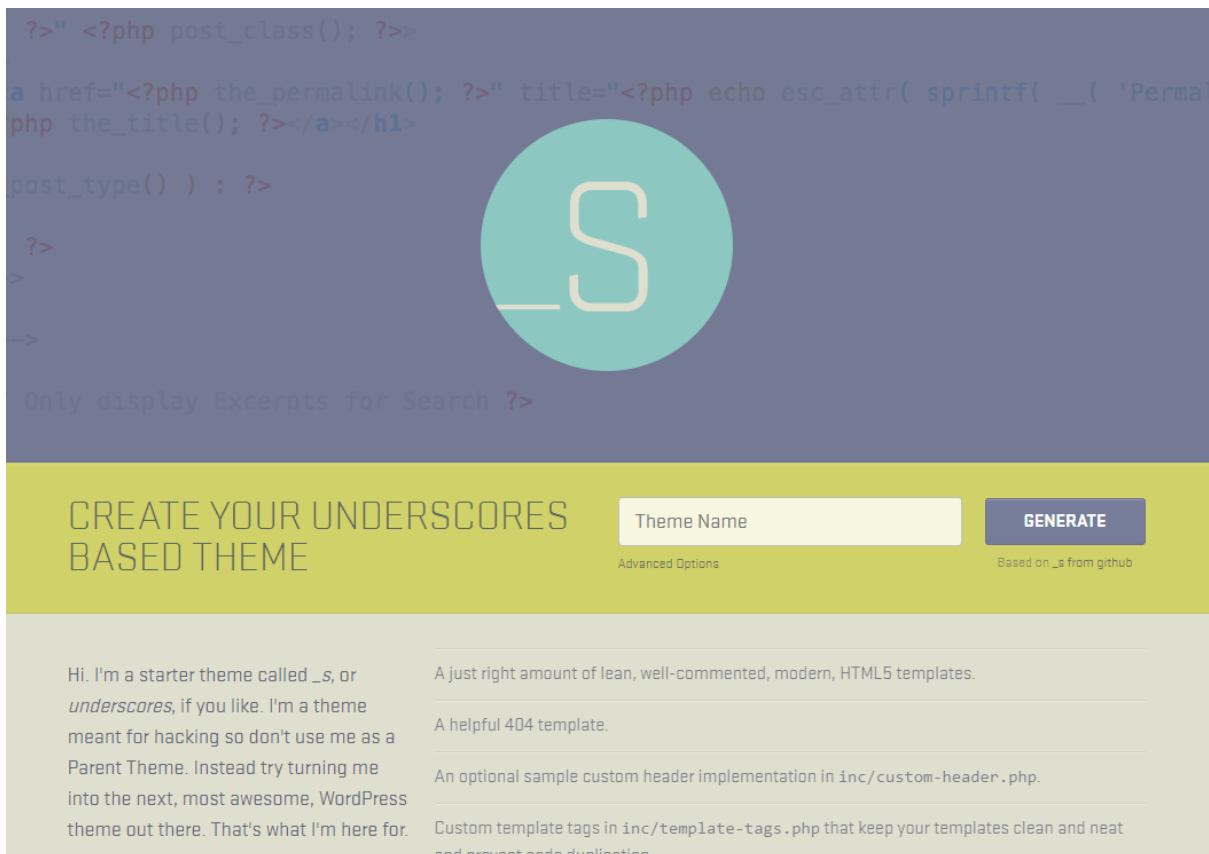


Рисунок 2.6 – Underscores

WordPress – це одна з найпопулярніших веб-платформ для управління веб-сайтами та контентом. Вона відома своєю простотою використання, широким спектром можливостей та гнучкістю. Надає широкі можливості для налаштування та розширення функціоналу сайту за допомогою різних плагінів та тем.

Одна з переваг використання WordPress для великих проєктів полягає в його масштабованості та зручності управління великим обсягом контенту. WordPress надає зручний інтерфейс для додавання, редагування та організації контенту на сайті. Крім того, він підтримує розподілену роботу над проєктом, що дозволяє різним користувачам працювати з різними ролями та правами доступу.

Інтеграція з CMS передбачає встановлення та налаштування WordPress на сервері, створення необхідних шаблонів та сторінок, а також інтеграцію верстки з функціональними можливостями, такими як можливість додавати, редагувати та видаляти контент, управляти меню, створювати форми зворотного зв'язку та багато іншого.

При інтеграції теми, важливо дотримуватись чіткого алгоритму. Кроки реалізації:

1. Створення власної бази даних в phpMyAdmin (рисунок 2.7).

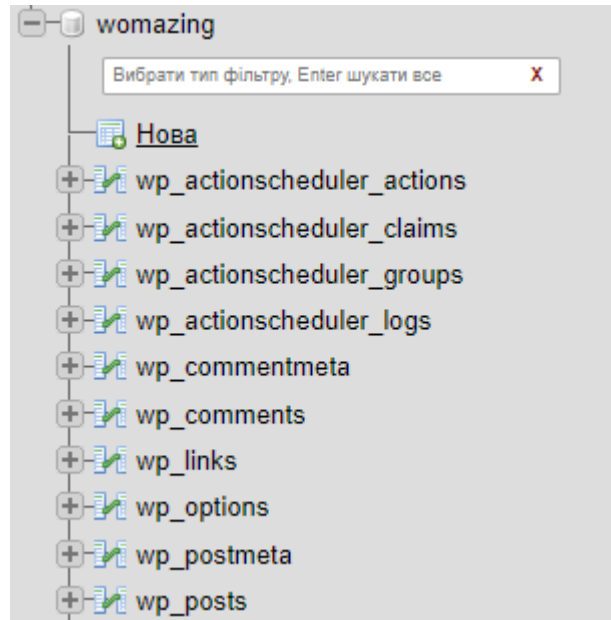


Рисунок 2.7 – phpMyAdmin

2. Завантаження WordPress з офіційного сайту. (<https://uk.wordpress.org/download/>) (рисунок 2.8)

Безцінно, але при цьому, безкоштовно  
Завантажте WordPress і використовуйте його на вашому сайті.

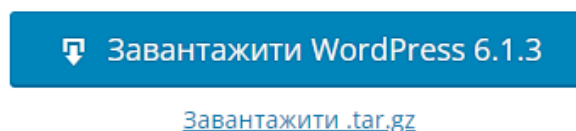


Рисунок 2.8 – Download WordPress

3. Підключення бази даних та додавання файлів, згенерованих у шаблоні underscores.

Після цих етапів можна починати перенос верстки до нашої теми. Перед початком переносу ми ознайомлюємось з адмін-панеллю (рисунок 2.9) WordPress та підключаємо нашу тему в розділі "Вигляд → Теми" (рисунок 2.10).

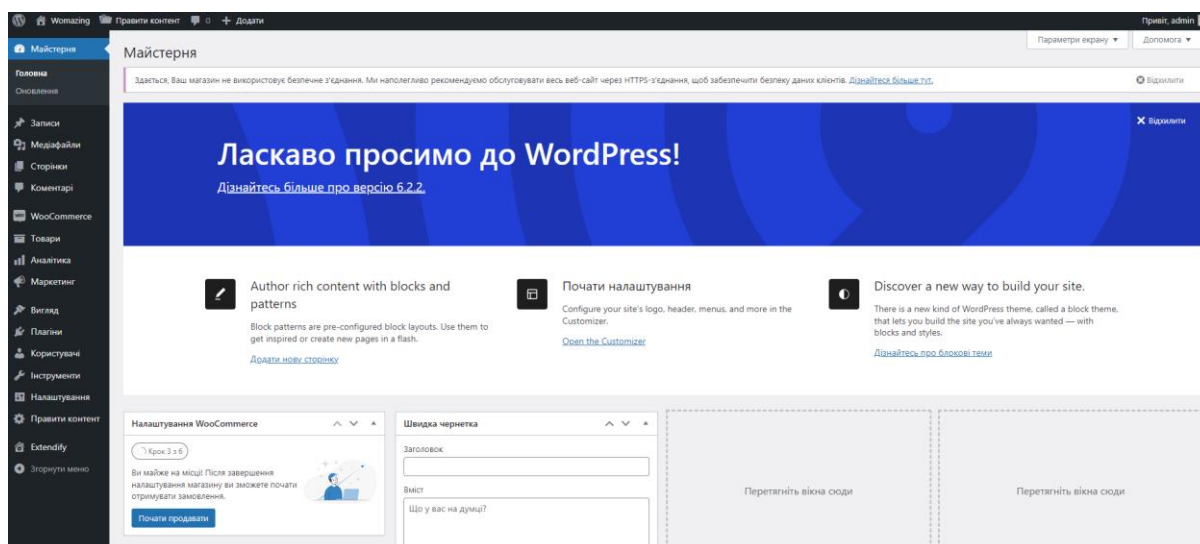


Рисунок 2.9 – Адмінка

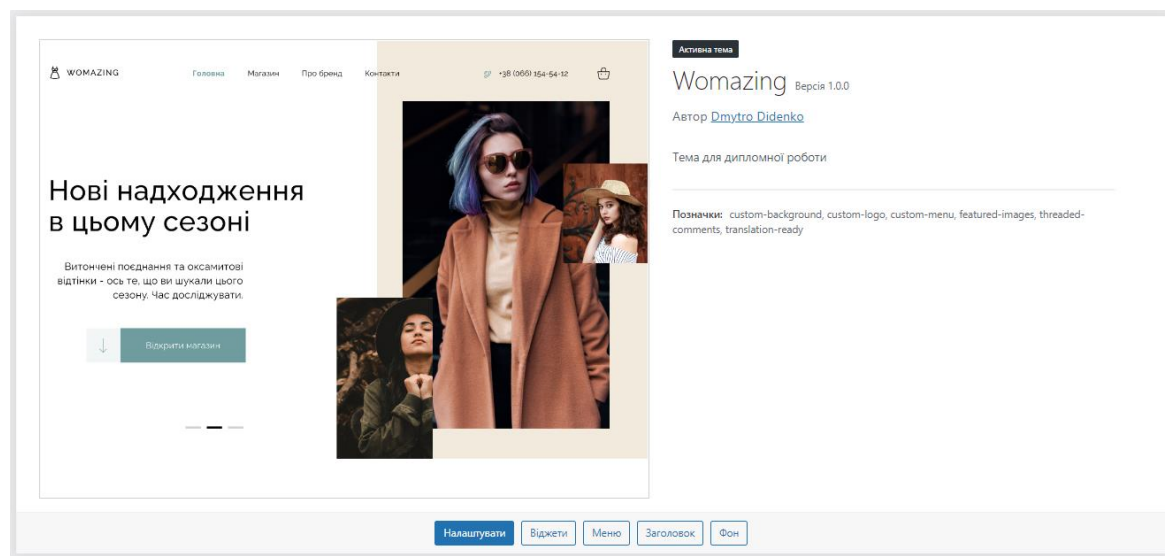


Рисунок 2.10 – Тема Womazing

Після підключення нової теми, відкриваємо редактор коду Visual Studio Code і продовжуємо інтеграцію.

Основні кроки :

1. В файлі `functions.php` підключаємо стилі CSS і JS.

2. Налаштовуємо файл header.php і footer.php.
3. Створюємо новий файл, шаблон для сторінок з назвою template-home.php та реєструємо його. (рисунки 2.11, 2.12).

```
wp-content > themes > womazing > template-home.php > s
1  <?php
2
3  /* Template name:home page */
4
5  get_header(2);
6
7  global $redux;
8  ?>
9
10 > <section class="season" id="season"> ...
146 <?php
147     get_footer();
148
149 ?>
```

Рисунок 2.11 – Template-home.php

Короткий зміст	^
Видимість	Публічно
Опублікувати	28.05.2023 01:02
Шаблон	home page
URL	womazing
АВТОР	
	<input type="text" value="admin"/>

Рисунок 2.12 – Шаблон home page

4. Відповідно до template-home.php створюємо та реєструємо всі інші сторінки (рисунки 2.13).



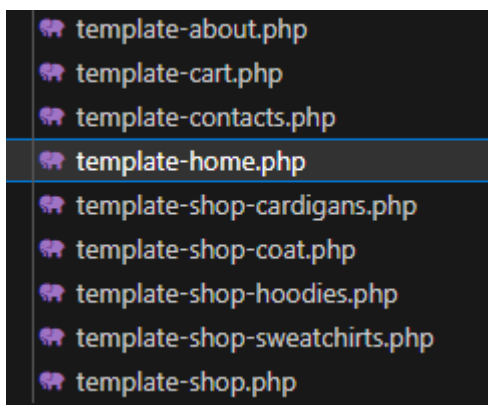


Рисунок 2.13 – Шаблони template

Після створення всіх сторінок (рисунок 2.14) та виконання вказаних кроків завершуємо інтеграцію нашої теми. Вона повністю функціонує на платформі WordPress. Проте варто зазначити, що всі налаштування здійснюються на локальному сервері (Open Server Panel), тому веб-сайт не буде доступний у мережі Інтернет.

 A screenshot of a WordPress page list table. The table has columns for checkboxes, page titles, authors, and dates. The page titles include 'about', 'contacts', 'home — Головна сторінка', 'shop — Сторінка магазину', 'shop-cardigans', 'shop-coat', 'shop-hoodies', 'shop-sweatshirts', 'Кошик — Сторінка "Кошик"', 'Мій обліковий запис — Сторінка Мій обліковий запис', and 'Оформлення замовлення — Сторінка "Оформлення замовлення"'. The authors are all 'admin'. The dates range from 28.05.2023 to 31.05.2023.
 

<input type="checkbox"/>	Заголовок	Автор	Дата
<input type="checkbox"/>	about	admin	Опубліковано 28.05.2023 о 01:42
<input type="checkbox"/>	contacts	admin	Опубліковано 28.05.2023 о 01:45
<input type="checkbox"/>	home — Головна сторінка	admin	Опубліковано 28.05.2023 о 01:02
<input type="checkbox"/>	shop — Сторінка магазину	admin	Опубліковано 28.05.2023 о 01:13
<input type="checkbox"/>	shop-cardigans	admin	Опубліковано 28.05.2023 о 01:38
<input type="checkbox"/>	shop-coat	admin	Опубліковано 28.05.2023 о 01:31
<input type="checkbox"/>	shop-hoodies	admin	Опубліковано 28.05.2023 о 01:40
<input type="checkbox"/>	shop-sweatshirts	admin	Опубліковано 28.05.2023 о 01:36
<input type="checkbox"/>	Кошик — Сторінка "Кошик"	admin	Опубліковано 31.05.2023 о 17:14
<input type="checkbox"/>	Мій обліковий запис — Сторінка Мій обліковий запис	admin	Опубліковано 31.05.2023 о 17:14
<input type="checkbox"/>	Оформлення замовлення — Сторінка "Оформлення замовлення"	admin	Опубліковано 31.05.2023 о 17:14
<input type="checkbox"/>	Політика повернення — Чернетка	admin	Остання зміна 31.05.2023 о 17:14
<input type="checkbox"/>	Заголовок	Автор	Дата

Рисунок 2.14 – Сторінки

Крім того, необхідно забезпечити належну підтримку та підготовку сайту до релізу. Грамотна верстка, відповідність дизайну до функціональності, адаптивність на різних пристроїв і забезпечення зручної навігації - усе це гарантує успішну інтеграцію та створення веб-сайту, який задовольнить потреби користувачів

## 2.4 Налаштування плагінів redux, woocommerce

Після завершення верстки та інтеграції з CMS, можемо перейти до налаштування плагінів Redux і WooCommerce.

Налаштування плагіна Redux є важливою частиною розробки веб-сайту. Redux – це потужна бібліотека для керування станом додатків, яка дозволяє ефективно управляти даними та забезпечувати їх однорідність у всьому додатку. Для успішного налаштування плагіна, необхідно виконати наступні кроки:

1. Встановлення та активація плагіна: Для встановлення та активації плагіна Redux, переходимо до розділу 'Плагіни' у панелі адміністрування WordPress. Обираємо 'Додати новий' у меню 'Плагіни'. З використанням пошукової функції, знаходимо плагін Redux і натискаємо кнопку 'Встановити зараз'. Після завершення встановлення, активуємо плагін (рисунок 2.15).

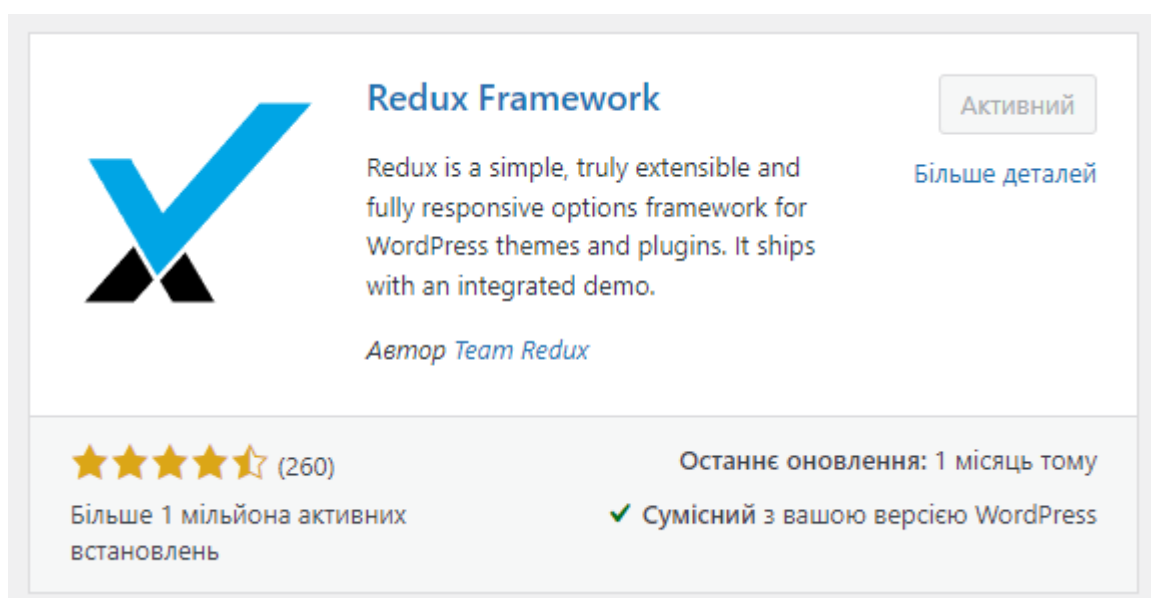


Рисунок 2.15 – Redux Framework

2. Підключення до теми: після підключення та активації плагіну, його файли зазвичай розміщуються у папці `wp-content/plugins/redux-framework/sample/`. Щоб успішно підключити плагін до нашої теми, необхідно скопіювати файл `sample-config.php` та перемістити його у папку `wp-content/themes/woomazing/inc/`. Важливо звернути увагу, що файл `sample-config.php` повинен бути перейменований на `redux-config.php`, щоб коректно працювати з нашою темою. (рисунок 2.16).

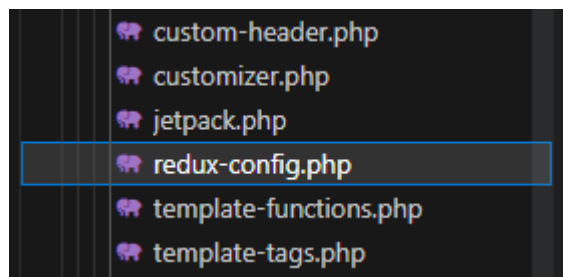


Рисунок 2.16 – `redux-config.php`

3. Після успішної інтеграції файла `redux-config.php`, в адміністративній панелі створюється новий розділ під назвою 'Правити контент' (рисунок 2.17). Кожен може встановити налаштування `redux-config.php` на свій розсуд, але головне — не забувати відключити автоматичне підключення стилів CSS, які надає сам плагін Redux. Для цього необхідно знайти у файлі `redux-config.php` рядок з налаштуванням `'output_tag'` та встановити йому значення `'false'` (рисунок 2.18). Це дозволить уникнути конфліктів стилів і надасть більш гнучкий контроль над зовнішнім виглядом теми.

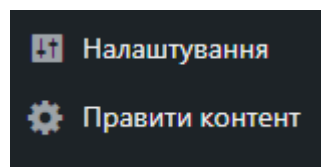


Рисунок 2.17 – Правити контент

```

159
160 // Allows dynamic CSS to be generated for customizer and google fonts,
161 // but stops the dynamic CSS from going to the page head.
162 'output_tag' => false,|

```

Рисунок 2.18 – output\_tag

4. Для розміщення наших компонентів необхідно створити папку з назвою 'components' за шляхом wp-content → themes → woomazing → inc → components (рисунок 2.19, 2.20). Саме в цій папці будуть знаходитись наші компоненти, які названі відповідно до назв сторінок. На рисунку 2.20 можна побачити приклад структури папки з компонентами.

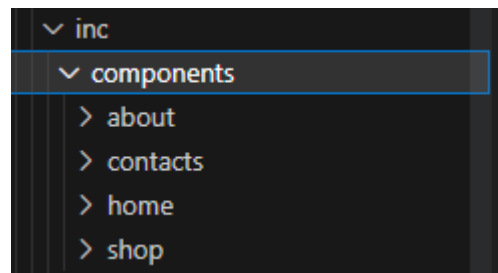


Рисунок 2.19 – Компоненти

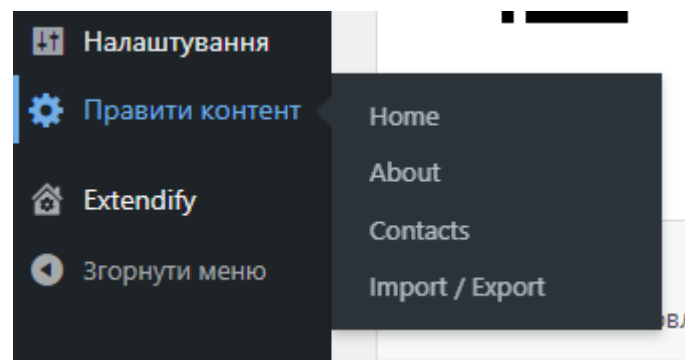


Рисунок 2.20 – Вигляд в адмінці

На прикладі сторінки Home перевіряємо детальніше налаштування. У папці components → home створюємо PHP-файли та називаємо їх відповідно до секцій у верстці (рисунок 2.21). Після цього в файлі redux-config.php реєструємо сторінку Home та файли collection-section.php, important-section.php, season-section.php, team-section.php.

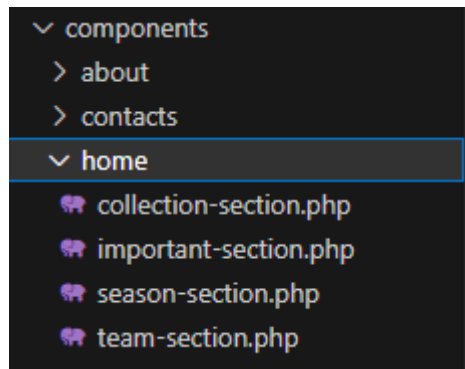


Рисунок 2.21 – Компоненти в VS

Детально перевіряємо першу секцію на сторінці Home, season-section.php. Основна задача полягає в тому, що створюється власна ієрархія по id. Для Секції 1 id = home-section, і далі можна додавати будь-яку назву.

Відповідно, якщо відкрити в адмінці "Правити контент" → "Home" → "Секція 1", можемо побачити, що створюється можливість вводити текст та фото в залежності від обраного типу (text, textarea, image) у файлі season-section.php (рисунок 2.22).

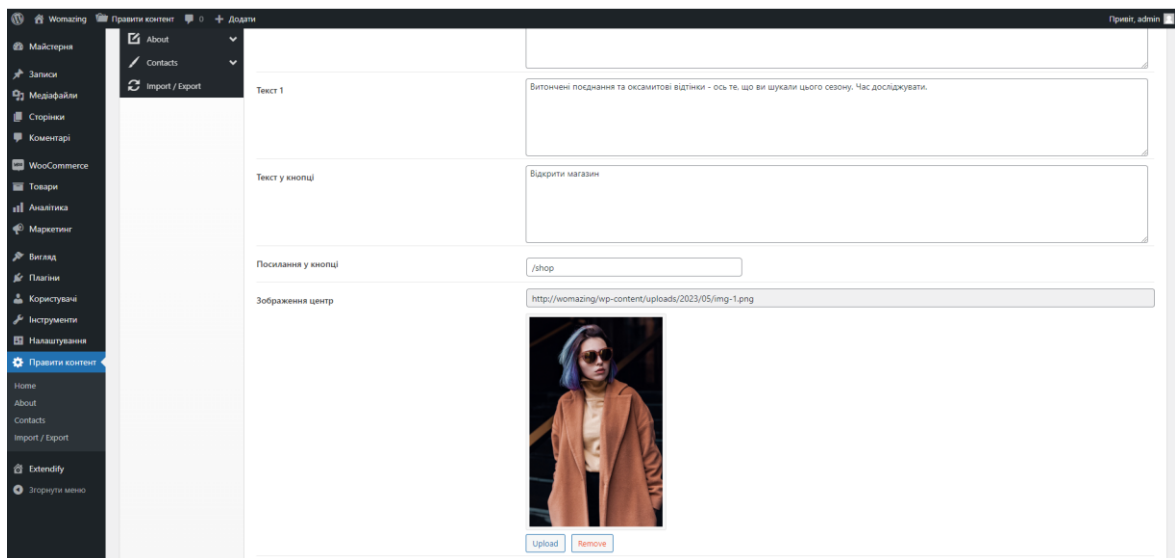


Рисунок 2.22 – Видяг в адмінці

Після успішної інтеграції Redux, переходимо до наступного етапу – підключення та налаштування плагіна WooCommerce. WooCommerce є надзвичайно важливим інструментом для розробки інтернет-магазину, який надає широкі

можливості для управління та продажу товарів. Для успішної інтеграції та налаштування плагіна WooCommerce, необхідно виконати наступні кроки:

1. Встановлення та активація плагіна: Цей етап є ключовим для створення функціонального та успішного інтернет-магазину, де зможемо здійснювати продажі, керувати інвентарем та надавати зручний досвід покупцям. (рисунок 2.23)

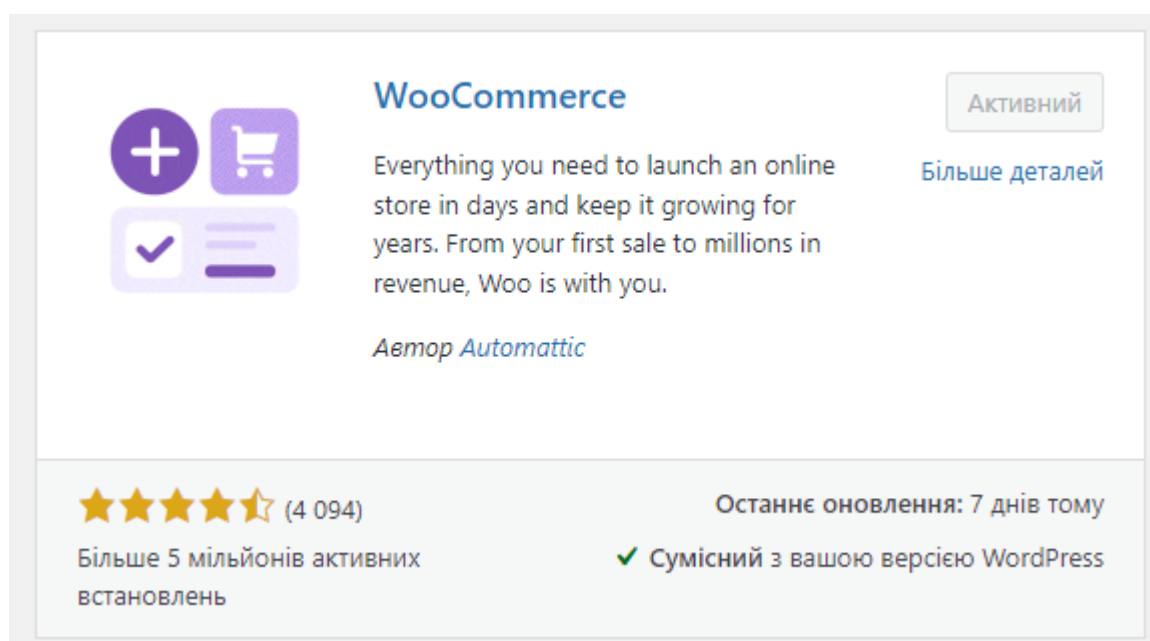


Рисунок 2.26 – Woocommerce

2. Після успішного підключення та активації плагіну, його файли зазвичай розміщуються у папці `wp-content/plugins/woocommerce`. Для того, щоб належним чином інтегрувати плагін у тему, створюємо файл `woo.php` в папці `wp-content/themes/woomazing/inc/`. (рисунок 2.24).

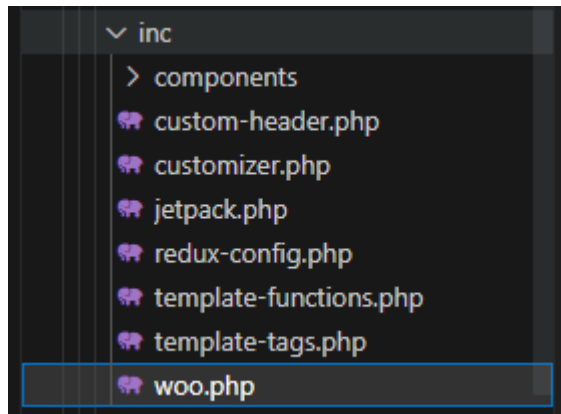


Рисунок 2.24 – Woocommerce в темі

3. Після створення файлу `woo.php` активуємо його та відключаємо деякі стандартні хуки. Це необхідно для забезпечення правильної роботи WooCommerce у нашій темі та можливості власного налаштування. (рисунок 2.25).

```

wp-content > themes > womazing > inc > woo.php > ...
1  <?php
2
3  if (in_array('woocommerce/woocommerce.php', apply_filters('active_plugins' , get_option('active_plugins')))) {
4
5      function womazing_add_woocommerce_support() {
6          add_theme_support( 'woocommerce' );
7      }
8
9      add_action( 'after_setup_theme', 'womazing_add_woocommerce_support' );
10
11     remove_action('woocommerce_before_main_content','woocommerce_breadcrumb' ,20);
12     remove_action('woocommerce_sidebar','woocommerce_get_sidebar' ,10);
13     /* remove_action('woocommerce_before_shop_loop','woocommerce_catalog_ordering' ,30); */
14     remove_action('woocommerce_before_single_product_summary','woocommerce_show_product_sale_flash' , 10);
15     remove_action('woocommerce_after_single_product_summary','woocommerce_output_product_data_tabs' , 10);
16
17
18 }

```

Рисунок 2.25 – Підключення woo.php

4. Створюємо папку `woocommerce` у `wp-content/themes/womazing/inc/` і копіюємо файли `archive-product.php` та `content-single-product.php` з розташування `wp-content/plugins/woocommerce/templates`. Будемо працювати з цими файлами, оскільки вони необхідні для налаштування сторінки магазину (`archive-product.php`) та відображення списку товарів.

5. Товари, які будуть виводитись, потрібно завантажити через адміністративний розділ (рисунок 2.26). Крім того, одразу ж створюємо для них атрибути - 'розмір' та 'колір' (рисунок 2.27).

Рисунок 2.26 – Товари

Рисунок 2.27 – Атрибути

6. Працюємо з файлом `archive-product.php`. В нього переносимо нашу верстку та додаємо код для виводу створених товарів (рисунок 2.28). Для отримання детальної інформації про те, як виводити зображення, заголовок, ціну та інші дані, можемо скористатися документацією WooCommerce: Get Product Info (<https://www.businessbloomer.com>).



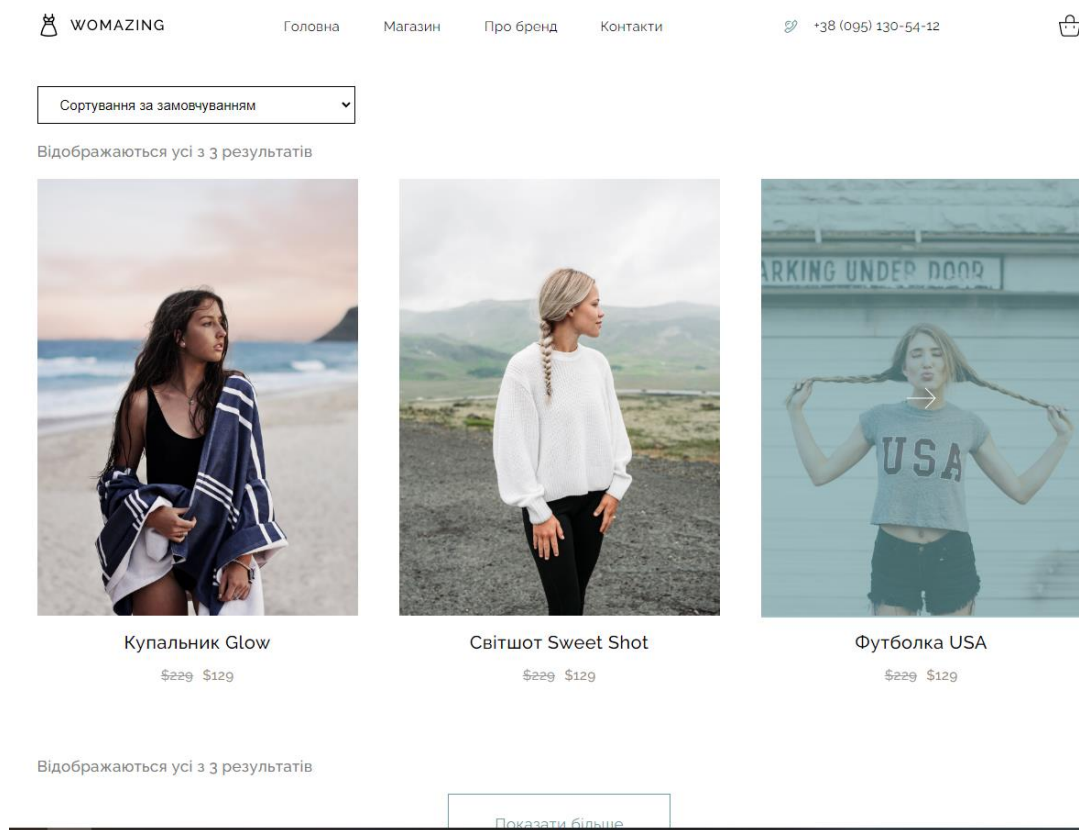


Рисунок 2.28 – Сторінка "shop"

7. Працюємо з файлом `content-single-product.php`, який відповідає за відображення зображення товару на який перейшов користувач. В цей файл переносимо нашу розроблену верстку та додаємо код для виводу зображення одного товару, створених атрибутів, таких як 'розмір' та 'колір' (рисунок 2.29).

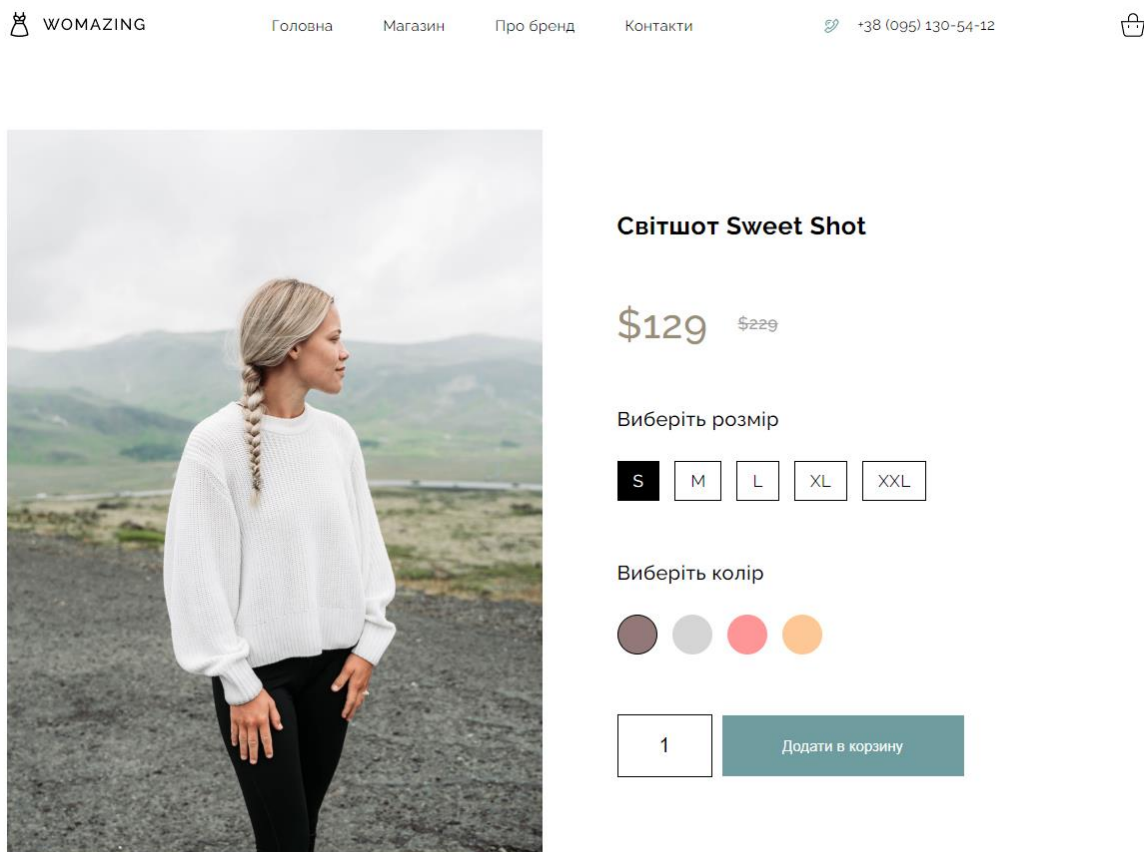


Рисунок 2.29 – Сторінка "item"

- Для налаштування сторінки 'Кошик' створюємо файл `template-cart.php` в шляху `wp-content/themes/woomazing/template-cart.php`. Код з файлу `wp-content/plugins/woocommerce/templates/cart/cart.php` копіюється та переноситься до створеного файлу `template-cart.php`. За допомогою файлу `main.css`, який підключений у `functions.php`, кастомізуємо відображення товару. Результат додавання товару на сторінці 'Кошик' можна побачити на рисунку 2.31.

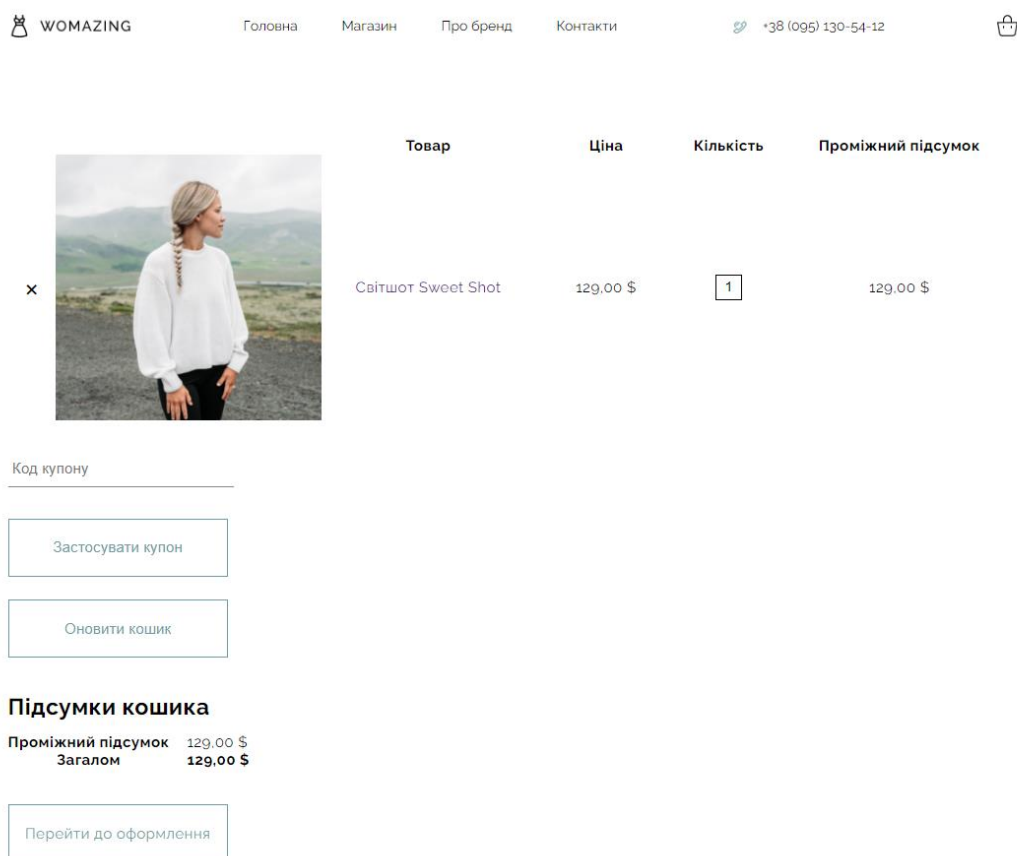


Рисунок 2.31 – Товар у кошику

9. Після вибору товару, користувач натискає кнопку 'Перейти до оформлення' (рисунок 2.32) і заповнює всю необхідну інформацію. Сторінка 'Оформлення замовлення' може бути створена за аналогічним принципом, як сторінка 'Кошик' у пункті 8.

# Оформлення замовлення

Головна — Оформлення заказу

## Дані покупця

Ім'я  
\_\_\_\_\_

E-mail  
\_\_\_\_\_

Телефон  
\_\_\_\_\_

## Адреса одержувача

Країна  
\_\_\_\_\_

Місто  
\_\_\_\_\_

Вулиця  
\_\_\_\_\_

## Ваше замовлення

Товар	Усього
Футболка USA	\$129
Підсумок	\$129
<b>Разом</b>	<b>\$129</b>

## Способи оплати

Оплата готівкою

Замовити

Рисунок 2.32 – Сторінка замовлення

Підключення та налаштування плагінів Redux і WooCommerce є важливим етапом у розробці веб-сайту. Redux надає потужні інструменти для керування станом додатків, забезпечуючи однорідність даних. WooCommerce, зі своїм широким набором функціональності, дозволяє створювати і управляти інтернет-магазином. Правильна інтеграція допомагає створити потужний та функціональний веб-сайт з можливістю продажу товарів онлайн

## 2.5 Тестування та налагодження системи

Тестування та налагодження системи є завершальним етапом в розробці веб-сайту. Під час цього процесу здійснюються наступні дії:

1. Функціональне тестування: перевіряємо, чи працюють всі функціональні можливості системи відповідно до вимог та специфікацій. Переконаємося, що коректно відображаються всі сторінки, елементи взаємодії з користувачем, форми, функції магазину, опції пошуку тощо.
2. Компонентне тестування: Перевіряємо окремі компоненти системи, такі як розмітка, кнопки, меню, слайдери, форми, елементи навігації тощо. Впевнюємось, що ці компоненти працюють правильно та відповідають заданим вимогам.
3. Кросс-браузерне тестування: Перевіряємо, як система відображається та працює на різних веб-браузерах, таких як Google Chrome, Mozilla Firefox, Safari, Microsoft Edge тощо. Переконаємося, що система має однаковий вигляд та функціонал на всіх популярних браузерах.
4. Відповідність до мобільних пристроїв: Перевіряємо, як система відображається та працює на різних мобільних пристроях, таких як смартфони та планшети. Забезпечуємо, щоб система була адаптивною та коректно працювала на різних розмірах екранів.
5. Тестування функціональності магазину: У випадку, якщо система включає функції магазину, перевіряємо процес додавання товарів до кошика, оформлення замовлення, розрахунок вартості, відстеження стану замовлення тощо.
6. Оптимізація продуктивності: Перевіряємо продуктивність системи та виконують необхідні оптимізації для забезпечення швидкої роботи та низької завантаженості сторінок.
7. Тестування безпеки: Перевіряють систему на наявність потенційних уразливостей та проводимо тестування безпеки, щоб забезпечити захист від

можливих атак. Використовують різні методики, такі як перевірка на наявність XSS атак, SQL-ін'єкцій, вразливостей аутентифікації та авторизації.

8. Завершення налагодження: Завершуємо останні налагодження та виправлення, щоб система була готовою до впровадження. Перевіряємо, що всі функції та компоненти працюють належним чином, і система готова до релізу.

Після успішного тестування та налагодження системи, веб-сайт готовий для впровадження та використання користувачами. Можемо переходити до фінальних етапів, таких як розгортання системи на сервері, надання доступу користувачам, підтримка та обслуговування системи після впровадження.

## ВИСНОВКИ

Внаслідок аналізу сучасного стану та особливостей ринку продажу одягу, було виявлено, що веборієнтована інформаційна система є необхідним інструментом для підтримки бізнесу в цій галузі. Розвиток онлайн-шопінгу та зміни в способах споживання модного одягу призводять до необхідності створення інформаційної системи, яка повинна бути зручною та функціональною для споживачів.

Розробка веборієнтованої інформаційної системи для магазину одягу є важливим кроком у модернізації та удосконаленні бізнес-процесів. Веборієнтована система дозволить підвищити ефективність роботи магазину, забезпечити зручне користування для клієнтів, а також підвищити конкурентоспроможність на ринку продажу одягу.

Внаслідок проведення дослідження з розробки веборієнтованої інформаційної системи для магазину одягу :

1. Проаналізовано потреби магазину одягу, визначено функціональні вимоги до веборієнтованої інформаційної системи магазину одягу.
2. Розроблено архітектуру та дизайн інформаційної системи.
3. Реалізовано програмний код інформаційної системи з використанням сучасних технологій веб-розробки.
4. Проведено тестування та впровадження веборієнтованої інформаційної системи в роботу магазину одягу.

Веборієнтована система, розроблена в роботі, сприяє автоматизації багатьох рутинних завдань, забезпечує можливість збільшення обсягів продажів шляхом привертання нових клієнтів та покращення взаємодії з наявними клієнтами через персоналізовані пропозиції, акції та знижки тощо.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Котлер Ф., Армстронг Г. Принципи маркетингу. Київ: Видавництво Ін Юре, 2022.
2. Левітт Т. Ефективність маркетингу. Київ: Наш Формат, 2023 (Дата звернення: 25.04.2023).
3. Портер М. Конкурентна перевага: створення та забезпечення вищої продуктивності. Київ: Наш Формат, 2022 (Дата звернення: 25.04.2023).
4. Дейтон Т., Котлер Ф. Маркетингове дослідження: стратегічне планування, збір та аналіз даних, тлумачення результатів. Київ: Видавництво Ін Юре, 2017
5. Чернов В. Електронна комерція: технології, системи, моделі. Київ: Кондор, 2021 (Дата звернення: 27.04.2023).
6. Laudon K., Laudon, D. Management Information Systems: Managing the Digital Firm. Kyiv: In Yure Publishing House, 2022 (Last accessed: 28.04.2023).
7. O'Brien J., Marakas G. Management Information Systems: Practice and Strategies. Kyiv: Kondor, 2023 (Last accessed: 26.04.2023).
8. Turban E., Volonino L., Wood D. Information Technology Management: Digital Strategies for Insight, Action, and Sustainable Performance. Kyiv: Kondor, 2022 (Last accessed: 28.04.2023).
9. Steinberg R., Wickers, A. Analysis and Design of Information Systems: A Conceptual Approach. Kyiv: Nash Format, 2021 (Last accessed: 25.04.2023).
10. Ross D., Weber R. Database Management Systems: Tasks, Methods, and Tools. Kyiv: In Yure Publishing House, 2021 (Last accessed: 27.04.2023).
11. Kolisnyk O. Business Process Modeling: Concepts, Methods, Tools. Kyiv: Kondor, 2021 (Last accessed: 28.04.2023).
12. Isayeva O. Information Technologies in Business: Theory and Practice. Kyiv: Nash Format, 2022 (Last accessed: 28.04.2023).



13. Dick P., McDonald M. Marketing Management: Analysis, Planning, Implementation, and Control. Kyiv: In Yure Publishing House, 2019 (Last accessed: 28.04.2023).
14. Humboldt V. Lectures on the Philosophy of Education. Kyiv: Nash Format, 2021 (Last accessed: 28.04.2023).
15. Meyer L. History of Psychology: From Antiquity to Modernity. Kyiv: In Yure Publishing House, 2020. № 19(2). P. 147–254.
16. Kaplan R., Norton D. The Balanced Scorecard: Translating Strategy into Action. Kyiv: Kondor., 2021. № 11(4). P. 152–170 (Last accessed: 29.04.2023).
17. Deming W. Quality, Productivity, and Competitive Position: Selected Works. Kyiv: Nash Format., 2023. № 18(1). P. 99–155 (Last accessed: 21.04.2023).
18. Drucker P. The Practice of Management: Concepts and Principles. Kyiv: In Yure Publishing House, 2021. № 20(1). P. 247–297. (Last accessed: 28.04.2023).
19. Jackson P. Managing Organizational Change: A Systems Thinking Approach. Kyiv: Nash Format, 2018.
20. Davenport T. Big Data Analytics: Making Smarter Decisions and Getting Better Results. Kyiv: Kondor, 2022 (Last accessed: 28.04.2023).
21. Senou D., Davis, G. Knowledge Management: Technologies and Methods. Kyiv: In Yure Publishing House, 2021 (Last accessed: 28.04.2023).
22. Goldratt E. The Goal: A Process of Ongoing Improvement. Kyiv: Nash Format., 2022. № 8(2). P. 159–170. (Last accessed: 30.04.2023).
23. Rummler A., Brache D. Improving Performance: How to Manage the White Space on the Organization Chart. Kyiv: Kondor, 2023. № 12(2). (Last accessed: 06.05.2023).
24. Kernighan B., Ritchie D. The C Programming Language. Kyiv: In Yure Publishing House., 2020.
25. Martin R. Clean Code: A Handbook of Agile Software Craftsmanship. Kyiv: Nash Format., 2021. № 12(4). P. 5–8. (Last accessed: 23.04.2023).

26. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Kyiv: Kondor, 2022. № 12(10). (Last accessed: 28.05.2023).
27. Beck K. Test-Driven Development: By Example. Kyiv: In Yure Publishing House, 2023. № 11(5) (Last accessed: 10.05.2023).
28. McConnell S. Code Complete: A Practical Handbook of Software Construction. Kyiv: Nash Format, 2022. № 12(5). (Last accessed: 25.04.2023).
29. Brooks F. The Mythical Man-Month: Essays on Software Engineering. Kyiv: In Yure Publishing House, 2020. № 12(4), P. 64–112. (Last accessed: 15.04.2023).
30. Sutherland, J. Scrum: The Art of Doing Twice the Work in Half the Time. Kyiv: Nash Format, 2023. № 12(3). P. 219–333. (Last accessed: 25.04.2023).
31. Cleland D., Ireland L. Project Management: Strategic Design and Implementation. Kyiv: In Yure Publishing House, 2023. (Last accessed: 29.04.2023).
32. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. Kyiv: Kondor, 2023. № 18(3 March). (Last accessed: 28.04.2023).
33. Gido J., Clements J. Successful Project Management. Kyiv: In Yure Publishing House, 2022. № 2(1). (Last accessed: 30.04.2023).
34. Schwalbe K. Information Technology Project Management. Kyiv: Nash Format., 2019.
35. Roberts T. Applied Software Measurement: Global Analysis of Productivity and Quality. Kyiv: Kondor, 2021. № 30(4). P. 187–211. (Last accessed: 30.04.2023).
36. Pressman R. Software Engineering: A Practitioner's Approach. Kyiv: In Yure Publishing House, 2020. № 11(4). P. 159–172. (Last accessed: 30.04.2023).
37. Sommerville I. Software Engineering. Kyiv: Nash Format, 2019
38. Larman C. Agile and Iterative Development: A Manager's Guide. Kyiv: Kondor, 2020. (Last accessed: 29.04.2023).
39. McConnell S. (2022). Rapid Development: Taming Wild Software Schedules. Kyiv: Nash Format, 2022. № 14(3-4). P. 260–297. (Last accessed: 30.04.2023).
40. Fowler M. Refactoring: Improving the Design of Existing Code. Kyiv: In Yure Publishing House., 2022. № 22(2). P. 79–198. (Last accessed: 04.05.2023).

41. Davis A. Software Requirements: Objects, Functions, and States. Kyiv: Nash Format, 2023. № 10(3). P. 24–64. (Last accessed: 06.05.2023).

42. Cockburn A. Writing Effective Use Cases. Kyiv: In Yure Publishing House, 2023. P. 25–83. (Last accessed: 05.05.2023).

43. Ambler S. Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process. Kyiv: Kondor, 2021. № 12(3). P. 160–307. (Last accessed: 05.05.2023).

44. Hunt A., Thomas D. The Pragmatic Programmer: Your Journey to Mastery. Kyiv: Nash Format, 2020.

45. McConnell S. Software Estimation: Demystifying the Black Art. Kyiv: In Yure Publishing House., 2022. № 5. P. 29–87.

# ДОДАТКИ

ДОДАТОК А  
(обов'язковий)

SUMMARY

Didenko D.O. Development of a Web-oriented Information System for a Clothing Store – Bachelor's degree thesis. Sumy State University, Sumy, 2023.

The functional requirements of the web-oriented information system for a clothing store have been defined in the work. The architecture and design of the information system have been developed. The software code of the web-oriented information system for the clothing store has been implemented using modern web development technologies. Testing and implementation of the information system in the operation of the clothing store have been conducted.

Keywords: web-oriented information system, electronic commerce, clothing store, WordPress, Redux, WooCommerce.

АНОТАЦІЯ

Діденко Д.О. Розробка веборієнтованої інформаційної системи для магазину одягу – Кваліфікаційна робота бакалавра. Сумський державний університет, Суми, 2023.

В роботі визначено функціональні вимоги до веборієнтованої інформаційної системи магазину одягу. Розроблено архітектуру та дизайн інформаційної системи. Реалізовано програмний код веборієнтованої інформаційної системи магазину одягу з використанням сучасних технологій веб-розробки. Проведено тестування та впровадження інформаційної системи в роботу магазину одягу.

Ключові слова: веборієнтована інформаційна система, електронна комерція, магазин одягу, WordPress, Redux, WooCommerce.

## ДОДАТОК Б

```

137  /**
138   * Enqueue scripts and styles.
139   */
140  function womazing_scripts() {
141
142      wp_enqueue_style( 'womazing-main-css', get_template_directory_uri() . '/assets/css/main.css' );
143
144      wp_enqueue_script( 'womazing-jquery', get_template_directory_uri() . '/assets/js/jquery-3.6.4.min.js', array(), _S_VERSION, true );
145      wp_enqueue_script( 'womazing-main-js', get_template_directory_uri() . '/assets/js/main.js', array(), _S_VERSION, true );
146
147
148      if ( is_singular() && comments_open() && get_option( 'thread_comments' ) ) {
149          wp_enqueue_script( 'comment-reply' );
150      }
151  }
152  add_action( 'wp_enqueue_scripts', 'womazing_scripts' );

```

Рисунок Б.1 – Підключення CSS і JS

```

313  /*
314   * ---> START SECTIONS
315   */
316
317  // -> Home
318  Redux::set_section(
319      $opt_name,
320      array(
321          'title'           => esc_html__( 'Home', 'your-textdomain-here' ),
322          'id'              => 'home',
323          'desc'           => esc_html__( 'Головна сторінка', 'your-textdomain-here' ),
324          'customizer_width' => '400px',
325          'icon'           => 'el el-home',
326      )
327  );
328  require_once get_template_directory() . "/inc/components/home/season-section.php";
329  require_once get_template_directory() . "/inc/components/home/collection-section.php";
330  require_once get_template_directory() . "/inc/components/home/important-section.php";
331  require_once get_template_directory() . "/inc/components/home/team-section.php";
332

```

Рисунок Б.2 – Фрагмент коду з файла redux-config.php

```

1 <?php
2 /**
3  * Redux Framework text config.
4  * For full documentation, please visit: http://devs.redux.io/
5  *
6  * @package Redux Framework
7  */
8
9 defined( 'ABSPATH' ) || exit;
10
11 Redux::set_section(
12     $opt_name,
13     array(
14         'title'      => esc_html__( 'Секція 1', 'your-textdomain-here' ),
15         'id'         => 'home-season',
16         'subsection' => true,
17         'customizer_width' => '700px',
18         'fields'     => array(
19             array(
20                 'id'      => 'home-season-title',
21                 'type'    => 'textarea',
22                 'title'   => esc_html__( 'Заголовок', 'your-textdomain-here' ),
23                 'default' => 'Default Text',
24             ),
25             array(
26                 'id'      => 'home-season-text-1',
27                 'type'    => 'textarea',
28                 'title'   => esc_html__( 'Текст 1', 'your-textdomain-here' ),
29                 'default' => 'Default Text',
30             ),
31             array(
32                 'id'      => 'home-season-text-2',
33                 'type'    => 'textarea',
34                 'title'   => esc_html__( 'Текст у кнопці', 'your-textdomain-here' ),
35                 'default' => 'Default Text',
36             ),
37             array(
38                 'id'      => 'home-season-link',
39                 'type'    => 'text',
40                 'title'   => esc_html__( 'Посилання у кнопці', 'your-textdomain-here' ),
41                 'default' => 'Default Text',
42             ),
43         ),
44     ),
45 );

```

Рисунок Б.3 – Фрагмент коду з файла season-section.php

```

73 <section class="stuff" id="stuff">
74 <div class="container">
75 <?php woocommerce_catalog_ordering(); ?>
76 <?php woocommerce_result_count(); ?>
77 <div class="things_set">
78 <?php
79
80     if ( wc_get_loop_prop( 'total' ) ) {
81         while ( have_posts() ) {
82             the_post();
83
84             global $product;
85
86             >
87             <a href="<?>= get_permalink( $product->get_id() ); ?>" class="things_wrap">
88                 <div class="things_img">
89                     <?php echo wp_get_attachment_image( $product->get_image_id(), array( 350, 478 ) ); ?>
90                 </div>
91
92                 <h4 class="things_title">
93                     <?>= $product->get_name() ?>
94                 </h4>
95                 <div class="price_set">
96                     <p class="price_old">
97                         <?>= $product->get_regular_price(); ?>
98                     </p>
99                     <p class="price_new">
100                        <?>= $product->get_price(); ?>
101                    </p>
102                </div>
103            </a>
104
105            <?php
106
107            }
108        }
109
110    ?>
111
112 </div>
113 <?php woocommerce_result_count(); ?>
114 <a href="/shop" class="things_btn">Показати більше</a>
115 </div>
116 </section>

```

Рисунок Б.4 – Код виведення товарів