

Міністерство освіти і науки України



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
«КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Кафедра мультимедійних систем

**Повний цикл створення сайту, використовуючи принципи дизайну та  
обробки зображень**

**Текстова частина до курсової роботи  
за спеціальністю 121 «Інженерія програмного забезпечення»**

Керівник курсової роботи  
с.в. Борозенний С.О.

\_\_\_\_\_

(підпис)

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

Виконала студентка  
Медовщук В.В.

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

Київ 2025

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Кафедра мультимедійних систем

ЗАТВЕРДЖУЮ  
Зав.кафедри мультимедійних систем,  
доцент, к.ф-м.н.  
\_\_\_\_\_ О. П. Жежерун (підпис)  
“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студентці Медовщук Валерії Віталіївни факультету інформатики 3 курсу

ТЕМА: Повний цикл створення сайту, використовуючи принципи дизайну та обробки зображень

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Вступ

1. Аналіз проблеми та дослідження користувачів
2. Дизайн сайту: планування та реалізація
3. Реалізація вебзастосунку
4. Оптимізація продуктивності: обробка зображень та швидкість завантаження

Висновки

Джерела

Додатки (за необхідністю)

Дата видачі “ \_\_\_\_ ” \_\_\_\_\_ 2023 р. Керівник \_\_\_\_\_ (підпис)

Завдання отримав \_\_\_\_\_ (підпис)

**Календарний план виконання роботи**

№	Назва етапу курсового проекту (роботи)	Термін виконання етапу
1	Отримання завдання на курсову роботу	09.10.2024
2	Аналіз матеріалів та визначення актуальності	26.12.2024
3	Розробка вебзастосунку	08.01.2025
4	Написання текстової частини до курсової роботи	12.04.2025
5	Коригування виконаної роботи	24.04.2025
6	Створення презентації	01.05.2025

Медовщук В.В. \_\_\_\_\_

Борозенний О. С. \_\_\_\_\_

“        ”  
\_\_\_\_\_

<b>Анотація.....</b>	<b>5</b>
<b>Вступ.....</b>	<b>6</b>
<b>РОЗДІЛ 1. Аналіз проблеми та дослідження користувачів.....</b>	<b>8</b>
1.1 Актуальність проблеми.....	8
1.2 Доступність вебсайту для людей з обмеженими можливостями.....	8
1.4 Визначення цільової аудиторії. Створення user persona.....	11
1.5 Формулювання цілей та задач проєкту.....	12
<b>РОЗДІЛ 2: Дизайн застосунку.....</b>	<b>14</b>
2.1 Принципи дизайну, що застосовувались у проєкті.....	14
2.2 Figma.....	15
2.3 Розробка макетів.....	16
2.4 Взаємозв'язок між UI-дизайном та user persona.....	17
<b>РОЗДІЛ 3: Реалізація вебзастосунку.....</b>	<b>18</b>
3.1 Вибір технологій.....	18
3.1.1 React.....	18
3.1.2 React Router Dom.....	18
3.1.3 SCSS.....	19
3.1.4 Vite.js.....	19
3.1.5 Firebase.....	20
3.2 SPA.....	20
3.3 Структура проєкту.....	21
3.4 Функціонал застосунку.....	22
3.4.1 Реєстрація та вхід.....	22
3.4.2 Головна сторінка.....	23
3.4.3 Сторінка категорії.....	23
3.4.4 Сторінка товару.....	23
3.4.5 Сторінка кошика.....	23
3.4.6 Сторінка акаунту.....	24
3.4.7 Панель доступності.....	24
<b>РОЗДІЛ 4: Обробка зображень.....</b>	<b>25</b>
4.1 Аналіз форматів зображень для вебу: вибір оптимального рішення.....	25
4.2 Оптимізація зображень у проєкті.....	26
4.3 Результати стискання та економія пам'яті.....	28
4.4 Порівняння швидкості завантаження до та після оптимізації зображень.....	29
<b>Висновки.....</b>	<b>30</b>
<b>Список використаних джерел.....</b>	<b>31</b>

## Анотація

Курсова робота присвячена повному циклу створення вебсайту з акцентом на дизайн, функціональність та оптимізацію. Було проведено аналіз конкурентних рішень. На основі дослідження потреб користувачів сформовано user persona, що дозволило розробити зрозумілу й ефективну візуальну концепцію.

Сайт було реалізовано для людей з обмеженими можливостями, за допомогою таких технологій: HTML, SCSS, React та Firebase.

У заключній частині роботи розглядається оптимізація завантаження сайту шляхом обробки зображень: їх стискання, застосування сучасних форматів та впровадження механізму Lazy Load.

## Вступ

Метою даної курсової роботи є створення інклюзивного вебзастосунку, який поєднує в собі функціональність, візуальну естетику, адаптацію до різних користувацьких потреб та ефективність. Продукт орієнтований на дві основні категорії користувачів: тих, хто цінує естетику, і тих, кому важливий комфорт і простота у використанні.

У процесі розробки було виконано повний цикл створення вебсайту: від дослідження цільової аудиторії та аналізу конкурентів до створення дизайн-макетів, реалізації функціоналу та оптимізації завантаження. Також було окремо приділено увагу забезпеченню доступності інтерфейсу для людей з інвалідністю.

У першому розділі розглянуто актуальність теми, проведено аналіз конкурентних рішень, сформовано user persona та визначено ключові цілі проєкту.

Другий розділ присвячений дизайну: описано підхід до створення візуальної концепції, розробку прототипу, представлена типографічна шкала, набір компонентів, виявлено зв'язок між дизайном та персонами уявних користувачів.

У третьому розділі наведено детальний опис реалізації функціоналу вебзастосунку, включно з вибором технологій (HTML, SCSS, React, Firebase).

Четвертий розділ описує методи оптимізації продуктивності сайту, зокрема обробку зображень, їх стискання, використання сучасних форматів для покращення швидкодії.

Таким чином, дана курсова робота демонструє створення вебзастосунку, який поєднує естетику, доступність і функціональність — ключові складові сучасного digital-продукту у сфері моди.

# РОЗДІЛ 1. Аналіз проблеми та дослідження користувачів

## *1.1 Актуальність проблеми*

У період повномасштабної війни кількість людей з обмеженими можливостями постійно зростає. Моя особиста відповідальність як свідомої громадянки — сприяти створенню максимально комфортного середовища для їхнього життя після повернення з фронту. Переконана, що будь-які нові ініціативи — чи то відкриття магазину, чи розробка цифрового застосунку — мають обов'язково враховувати потреби людей з інвалідністю.

## *1.2 Доступність вебсайту для людей з обмеженими можливостями*

Після ознайомлення з офіційним перекладом Настанов з доступності вебвмісту (WCAG) 2.1, я дійшла висновку, що доступність сайту — це не додатковий елемент, а необхідна умова для створення інклюзивного цифрового продукту. Ці настанови містять низку рекомендацій щодо того, як зробити вебвміст доступнішим і зрозумілішим для всіх користувачів, зокрема для людей з інвалідністю.

WCAG 2.1 орієнтований на потреби різних категорій користувачів: осіб зі слабким зором або повною втратою зору, людей з порушеннями слуху, моторики, мовлення або чутливістю до світла, а також користувачів із когнітивними або навчальними труднощами.

Під час аналізу я виділила кілька ключових аспектів, які мають бути враховані під час розробки сайту відповідно до WCAG 2.1:

1. Альтернативний текст для зображень забезпечує доступ до візуального контенту для користувачів зі слабким зором за допомогою скрінрідерів.
2. Навігація з клавіатури дозволяє людям, які не користуються мишею, повноцінно взаємодіяти з інтерфейсом лише за допомогою клавіш.
3. Чітка структура заголовків допомагає користувачам орієнтуватися в контенті, особливо тим, хто використовує допоміжні технології.
4. Контрастність кольорів має бути достатньою для забезпечення комфортного читання інформації людьми з порушеннями зору.
5. Підписи до відео та аудіо (субтитри або текстові транскрипції) особливо важливі для користувачів із вадами слуху.
6. ARIA-атрибути допомагають засобам доступності правильно розпізнавати складні елементи інтерфейсу, як-от вкладки чи модальні вікна.
7. Форми повинні містити чіткі мітки, які пояснюють, яку саме інформацію очікується ввести.
8. Автоматичне відтворення медіа слід уникати, оскільки це може ускладнити користування сайтом для людей з когнітивними особливостями.
9. Масштабування тексту має відбуватися без втрати структури та читабельності сторінки.

Я включила цей підрозділ до курсової роботи, оскільки аналіз конкурентних платформ показав: навіть великі ресурси часто нехтують питанням доступності. Це створює бар'єри для частини користувачів. Я переконана, що інклюзивність повинна бути врахована ще на етапі проектування, і доступність має стати обов'язковим стандартом сучасного веброзроблення.

### 1.3 Аналіз конкурентів та існуючих рішень

Перед початком створення вебзастосунку було здійснено дослідження конкурентного ринку у межах української fashion-індустрії. Основною метою аналізу було виявлення сильних сторін існуючих рішень, а також визначення їхніх недоліків, зокрема в контексті доступності для людей з інвалідністю. Для порівняльного аналізу були обрані два популярні інтернет-магазини: Answear.ua і Intertop.ua.

Answear.ua пропонує базову підтримку навігації за допомогою клавіатури — користувачі можуть переходити між елементами через клавішу Tab, тому що присутня фокус-навігація. Також було виявлено низький рівень контрасту між текстом і фоновими елементами, при цьому можливості змінити кольорову схему під потреби користувача відсутні.

Позитивним моментом є те, що більшість зображень містять alt-атрибути. Водночас іконки та кнопки у більшості випадків не мають додаткових атрибутів доступності, таких як aria-label, role або label for, що знижує ефективність використання сайту з допоміжними технологіями. Незважаючи на використання семантичних тегів HTML, структура заголовків на сайті порушена: відсутня логічна ієрархія, що ускладнює роботу скрінрідерів. Крім цього, розміри інтерактивних елементів не відповідають мінімальним рекомендованим значенням. Проте сайт зберігає функціональність навіть при масштабуванні до 200%.

У цілому Answear.ua частково враховує потреби людей з обмеженими можливостями, однак має значний потенціал для вдосконалення в частині навігації, контрастності та структури контенту.

Сайт Intertop.ua демонструє кращу структурну організацію, зокрема завдяки використанню семантичних елементів HTML, таких як nav, main і footer. Проте, як і у випадку з Answear.ua, структура заголовків є нечіткою і не дотримується логічної ієрархії, що створює складнощі при взаємодії з екранними читалками.

Деякі графічні елементи мають альтернативні описи, але більшість візуального контенту залишається без таких. Кнопки і подібні елементи також рідко забезпечуються атрибутами доступності. Рівень контрастності тексту місцями недостатній. Під час переміщення по сайту клавішею Tab не відображається фокус-навігація, через що користувач не бачить, який елемент є активним. Розміри інтерактивних зон також не відповідають вимогам стандартів WCAG, хоча сайт досить добре реагує на масштабування.

У результаті аналізу було встановлено, що навіть провідні онлайн-платформи у сфері моди в Україні не повною мірою відповідають актуальним вимогам інклюзивності. Виявлені проблеми з навігацією, візуальним сприйняттям і відсутністю описових атрибутів підкреслюють потребу у створенні нового вебзастосунку, що буде доступним, інтуїтивно зрозумілим і зручним для максимально широкого кола користувачів, включно з людьми з інвалідністю.

#### *1.4 Визначення цільової аудиторії. Створення user persona*

Для створення зручного, функціонального та водночас привабливого вебзастосунку було проаналізовано характерні очікування і поведінку потенційних користувачів. На основі отриманих результатів були сформовані два узагальнені образи, що репрезентують основні типи аудиторії. Це дозволило краще зрозуміти потреби користувачів, їхню мотивацію і сформулювати відповідні вимоги до інтерфейсу та функціоналу сайту.

Перший тип користувача — жінка на ім'я Анна. Вона нещодавно почала більше цікавитися стилем, проте для неї ключовими залишаються комфорт і швидкість. Вона надає перевагу речам, що не потребують складного догляду, легко знаходить те, що їй підходить, і не витрачає багато часу на пошук одягу.

Її вік становить 34 роки. Вона цінує зручність, якість і швидке прийняття рішень. Від сайту очікує простий у користуванні інтерфейс, можливість швидко підібрати одяг без тривалого пошуку за допомогою фільтрів.

Серед функціональних рішень, орієнтованих на такий тип користувача, передбачено систему фільтрів, підбір одягу за персональними даними та можливість залишити відгук.

Другий тип користувача — жінка на ім'я Аліса. Вона активно цікавиться модними тенденціями, веде сторінку в соціальних мережах, читає профільні блоги та ретельно підходить до створення власного стилю. Для неї важливі естетика, візуальне натхнення і самовираження.

Її вік становить 28 років. Вона очікує привабливого і сучасного дизайну сайту, великої кількості візуального контенту, читати статті від відомих стилістів та отримати одяг, який повністю дає змогу виражати свою індивідуальність.

### *1.5 Формулювання цілей та задач проекту*

Основною метою проекту є розробка інклюзивного вебзастосунку для онлайн-продажу одягу. Сайт повинен забезпечувати зручну, інтуїтивну та естетичну взаємодію для широкої аудиторії користувачів, у тому числі для людей з обмеженими можливостями. Його функціональність має відповідати сучасним стандартам доступності та враховувати потреби різних категорій користувачів.

Особлива увага приділяється двом ключовим сегментам цільової аудиторії: користувачам, які орієнтуються на стиль, тренди та візуальне натхнення, а також користувачам, для яких пріоритетними є зручність і

практичність, зокрема швидкий пошук, простота вибору та комфорт у щоденному використанні.

Для досягнення поставленої мети було визначено наступні цілі та задачі. Щодо функціоналу вебзастосунку, передбачається реалізація візуального інтерфейсу, який буде водночас естетично привабливим і адаптованим до різних стилістичних уподобань користувачів. Планується впровадження системи фільтрації товарів за категоріями. Кожен користувач зможе створити персональний профіль, де буде доступна історія покупок та сторінка акаунту. Також буде реалізовано функцію кошика для формування замовлень і можливість купувати одяг безпосередньо через сайт. Додатково передбачено додавання системи відгуків з можливістю перегляду та написання коментарів до товарів, а також розміщення блоку з останніми трендами та рекомендаціями від стилістів. Уся функціональність буде розроблена з урахуванням потреб людей з обмеженими можливостями.

Попри гарний візуал ти функціональність, дуже важливо також реалізувати сайт максимально ефективно, з чим допоможе оптимізація зображень.

Реалізація всіх перелічених цілей дозволить створити не лише сучасний онлайн-магазин, а й цифровий продукт, що відповідає принципам інклюзивного дизайну та враховує актуальні підходи до взаємодії з користувачами.

## РОЗДІЛ 2: Дизайн застосунку

### *2.1 Принципи дизайну, що застосовувались у проєкті*

Під час розробки вебзастосунку було враховано ключові принципи дизайну, що дозволяють створити цілісний, доречний і гармонійний візуальний образ.

**Доречність.** Кожен елемент інтерфейсу має функціональне навантаження та сприяє передачі головної ідеї. Усі рішення в дизайні приймалися з урахуванням цільової аудиторії, а графічні засоби використовувались лише там, де вони справді допомагають краще донести зміст.

**Пропорційність.** Всі елементи розміщені у відповідності до їхньої ваги та значущості. Заголовки, зображення та текстові блоки гармонійно співвідносяться один з одним і не викликають візуального перевантаження.

**Напрямок.** Структура сторінок побудована таким чином, щоб вести погляд користувача логічним шляхом — від основного до другорядного. Розміщення блоків враховує природний напрямок читання, що полегшує сприйняття.

**Єдність стилю.** Вебзастосунок витримано в одному стилі: дотримано єдиної кольорової палітри, типографіки, сітки та візуального ритму. Це дозволяє уникнути візуального хаосу та сприяє впізнаваності інтерфейсу.

**Контраст.** Для виділення важливих елементів використано контрастні кольори, розміри шрифтів та об'єми вільного простору. Це дозволяє створити динаміку та привернути увагу користувача до ключових частин інтерфейсу.

**Цілісний образ.** Усі сторінки вебзастосунку розроблені як частини єдиної композиції. Дизайн кожного екрану узгоджується із загальним стилем і логікою всієї системи, щоб створити цілісне користувацьке враження.

Стриманість. Дизайн побудовано без надмірних ефектів і декоративних деталей. Обмежено кількість шрифтів і стилів, зосереджено увагу на змісті й зручності.

Увага до деталей. Усі відступи, міжрядкові інтервали, положення тексту та зображень ретельно вивірено, щоб уникнути візуальних помилок і забезпечити професійний вигляд інтерфейсу.

Ритм. Вебзастосунок має візуальний ритм — повторювані елементи та модулі дозволяють користувачеві легше орієнтуватися в структурі сайту. Це створює звичну і комфортну взаємодію.

## 2.2 *Figma*

Під час етапу проектування інтерфейсу я обрала інструмент Figma, оскільки він забезпечує зручне середовище для створення сучасного UI-дизайну, яке повністю працює у браузері та не потребує встановлення додаткового програмного забезпечення. Цей інструмент дозволив швидко створити макети всіх сторінок вебзастосунку, протестувати структуру й візуальну логіку, а також передати готовий дизайн у розробку.

Переваги використання Figma полягають у її кросплатформеності, можливості командної роботи в режимі реального часу, а також у наявності вбудованих функцій для побудови компонентів, автолейаутів та створення адаптивних фреймів. Завдяки підтримці системи компонентів я мала змогу забезпечити візуальну узгодженість інтерфейсу. Також Figma дає можливість швидко створювати прототипи для демонстрації сценаріїв взаємодії, що особливо зручно при тестуванні логіки навігації.

Саме тому Figma стала оптимальним вибором для розробки дизайну проекту — як з точки зору функціональності, так і з позиції швидкості та зручності роботи.

### *2.3 Розробка макетів*

Процес створення дизайну сайту розпочався зі створення user persona. Визначення типових образів користувачів допомогло краще зрозуміти їхні очікування та потреби, що стало основою для подальших дизайнерських рішень.

Перед розробкою макетів я провела невелике дослідження існуючих модних платформ, щоб визначити найбільш вдалий візуальний стиль та функціональні елементи. Я виділила компоненти, які привертали увагу завдяки своїй простоті, елегантності та зручності використання, і врахувала ці спостереження у власному проекті.

Окрему увагу я приділила типографіці: було обрано один універсальний шрифт для всього сайту, а також розроблено градацію його розмірів відповідно до ієрархії заголовків і основного тексту. Це забезпечило послідовність і візуальну гармонію інтерфейсу.

Наступним кроком стало створення прототипу сайту. Для роботи я використала класичну 12-колонну сітку, що допомогло правильно організувати простір на сторінках і забезпечити адаптивність макетів. При розробці окремих елементів я дотримувалась принципу компонентизації: всі базові кнопки, поля форм, заголовки та картки були створені як окремі компоненти і організовані в єдиний UI kit. Це значно спростило подальшу роботу над макетами і зробило дизайн більш системним.

Таким чином, завдяки чітко спланованому процесу в Figma вдалося створити основу для естетичного, зручного та послідовного інтерфейсу вебзастосунку.

#### *2.4 Взаємозв'язок між UI-дизайном та user persona*

Створення макетів у Figma базувалося на чітко сформульованих user persona, що дозволило зробити дизайн сайту максимально наближеним до очікувань цільової аудиторії.

Відповідно до першої персони Аліси, яка цінує естетику, стиль і візуальну гармонію, був розроблений вишуканий мінімалістичний інтерфейс із великою кількістю білого простору, продуманою типографікою та привабливими візуальними акцентами.

Друга персона Андріана, яка орієнтується на зручність і швидкість у виборі, вплинула на створення максимально інтуїтивної структури сайту: прості фільтри, логічні групування товарів, зрозуміла навігація без зайвих елементів. Таким чином, дизайн проекту поєднав естетичність і практичність, відповідаючи одночасно потребам обох основних типів користувачів.

## РОЗДІЛ 3: Реалізація вебзастосунку

### 3.1 Вибір технологій

Для розробки вебзастосунку були обрані сучасні інструменти, які дозволили створити швидкий, адаптивний та функціональний сайт із підтримкою реального часу та зручною маршрутизацією. Нижче наведено детальний опис основних технологій, що використовувалися у проєкті.

#### 3.1.1 React

React — це сучасна JavaScript-бібліотека для створення інтерфейсів користувача, що базується на декларативному компонентному підході. У проєкті React використовувався для розробки всіх ключових частин застосунку, включаючи головну сторінку, картки товарів, обліковий запис користувача, кошик і панель навігації. Бібліотека дозволяє застосовувати повторно використовувані компоненти, що спрощує структуру та обслуговування інтерфейсу. Завдяки віртуальному DOM оновлення елементів відбувається ефективно і з мінімальним навантаженням на систему. React підтримує JSX — синтаксис, який поєднує структуру інтерфейсу з логікою, що спрощує написання коду. Функціональні компоненти можна доповнювати станом та обробкою подій за допомогою хуків. Велика екосистема та активна спільнота забезпечують розробників великою кількістю готових рішень і документації. Усе це дозволило організувати проєкт у вигляді незалежних модулів, що зробило застосунок масштабованим, зручним у підтримці й адаптованим до потреб користувачів.

#### 3.1.2 React Router Dom

React Router — це офіційна бібліотека для реалізації маршрутизації в React-застосунках. Вона дозволяє створювати односторінкові інтерфейси, у яких користувач може переміщуватись між різними представленнями без

повного перезавантаження сторінки. У проєкті ця бібліотека використовувалась для навігації між основними розділами: головною сторінкою, категоріями, окремими картками товарів, кошиком та профілем користувача. Основні переваги використання React Router полягають у можливості динамічного відображення компонентів залежно від поточного шляху в адресному рядку браузера, зручній організації переходів між сторінками без оновлення всієї сторінки та збереженні стану застосунку між переходами, що покращує користувацький досвід.

### 3.1.3 SCSS

SCSS є надбудовою над стандартним CSS, яка входить до складу препроцесора Sass і розширює його можливості за допомогою таких інструментів, як змінні, вкладеність, міксини, функції та модульність. Це дає змогу створювати більш структурований, масштабований і зручний у підтримці код. У межах проєкту SCSS використовувався для організації стилів у вигляді окремих модулів, а також для формування глобального файлу змінних, який містив ключові параметри — кольори, типографіку та відступи. Завдяки використанню змінних стало можливо швидко перемикає теми інтерфейсу, наприклад, реалізовувати контрастний або монохромний режим шляхом зміни лише одного значення.

### 3.1.4 Vite.js

Vite — це сучасний інструмент для фронтенд-розробки, який забезпечує надзвичайно швидкий запуск локального сервера завдяки попередньому білду з використанням нативних ES-модулів у браузері. У проєкті він використовувався для локальної розробки без затримок компіляції, підтримки сучасного синтаксису та швидкого оновлення змін через гарячу перезагрузку модулів. Також Vite застосовувався для фінального збирання оптимізованої версії

застосунку перед розгортанням, що скоротило час розробки і зробило роботу з проектом ефективнішою.

### 3.1.5 *Firebase*

Firebase — це платформа для розробників, яка надає інструменти для створення веб- і мобільних застосунків, включно з засобами автентифікації, хмарними базами даних, хостингом, зберіганням файлів і аналітикою. У проекті Idillio Firebase використовувався для реалізації авторизації користувачів через електронну пошту, зберігання даних у хмарній базі Firestore, підтримки функціональності кошика, історії замовлень і системи відгуків. Завдяки використанню оновлень у реальному часі через Firestore вдалося забезпечити динамічну взаємодію з інтерфейсом. Інтеграція Firebase дала змогу створити персоналізовану взаємодію для кожного користувача.

### 3.2 *SPA*

У проекті було використано архітектурний підхід SPA (Single Page Application), що означає односторінковий застосунок. Цей підхід передбачає завантаження одного HTML-документа, після чого оновлення контенту здійснюється динамічно — без перезавантаження сторінки. Завдяки цьому користувач бачить плавні переходи між розділами сайту, а взаємодія з інтерфейсом відбувається швидко та без затримок.

Переваги SPA полягають у тому, що значна частина логіки застосунку виконується на клієнтській стороні, що зменшує навантаження на сервер і забезпечує кращу продуктивність. Такий підхід дозволяє зберігати стан користувача між переходами, що особливо зручно при роботі з кошиком, профілем або фільтрами.

У випадку з Idillio SPA-архітектура забезпечила ефективну маршрутизацію між сторінками (головна, каталог, картка товару, профіль,

кошик), пришвидшила завантаження вмісту та створила більш інтерактивний досвід користувача. Це особливо важливо для модного застосунку, де естетика та зручність мають велике значення.

### 3.3 Структура проєкту

Під час розробки вебзастосунку Idillio була організована чітка структура файлів і папок для забезпечення зручності в навігації, масштабованості та простоти підтримки коду.

Проєкт має таку основну структуру:

1. Коренева директорія IDILLIO містить службові та конфігураційні файли, необхідні для запуску та збірки застосунку. Тут знаходяться `index.html` як основний HTML-файл, `vite.config.js` для налаштувань Vite, а також `package.json` і `package-lock.json` для опису залежностей і скриптів.
  2. Папка `.firebase` зберігає службові файли, створені при ініціалізації Firebase.
  3. Папка `node_modules` містить усі зовнішні бібліотеки, встановлені через `npm`.
  4. Папка `public` призначена для зберігання статичних файлів, таких як іконки або зображення, що не обробляються інструментом збірки.
  5. Папка `src` є основною для розміщення всього коду застосунку. Вона включає в себе декілька підпапок.
  6. Папка `assets` використовується для зберігання зображень та інших медіафайлів.
  7. Папка `components` містить окремі компоненти інтерфейсу, які можуть повторно використовуватись у різних частинах сайту.
- Папка `data` імовірно використовується для зберігання локальних даних або констант, що використовуються в логіці застосунку.

8. Папка `firebase` містить конфігураційний файл `firebaseConfig.js`, який відповідає за підключення та ініціалізацію Firebase-сервісів.
9. Папка `pages` містить основні сторінки сайту, такі як головна сторінка, кошик, сторінка товару та профіль користувача. Кожна сторінка представлена окремим компонентом у форматі JSX.
10. Папка `styles` містить стилі, організовані для кожної сторінки або компонента окремо.
11. Файл `App.jsx` є основним компонентом, який об'єднує всі частини застосунку та підключає маршрутизацію.
12. Файл `main.jsx` є точкою входу в застосунок, де React-компонент `App` рендериться в DOM.
13. Файл `uploadProducts.js` використовується для завантаження товарів у базу даних, наприклад, у `Firestore`.
14. Додатково в кореневій директорії розташовані файли `.firebaserc` і `firebase.json` для налаштування Firebase-проєкту, а також файли `babel.config.json` і `eslint.config.js` для налаштування трансляції та перевірки коду.

### *3.4 Функціонал застосунку*

#### *3.4.1 Реєстрація та вхід*

У застосунку реалізована сторінка входу та реєстрації користувача. Без авторизації користувач не має доступу до основних дій, таких як додавання товарів до кошика, оформлення замовлення чи написання відгуків. Це дозволяє зберігати персоналізовані дані кожного користувача та контролювати процес купівлі.

### *3.4.2 Головна сторінка*

На головній сторінці розміщено промо-блок з поточними колекціями, а також реалізовано навігаційне меню і футер. Ця сторінка виконує функцію вітрини магазину й одразу знайомить користувача з пропозиціями сезону.

### *3.4.3 Сторінка категорії*

На сторінці з категоріями користувач може переглянути весь доступний асортимент у певній категорії (наприклад, "Взуття", "Акcesуари" тощо). Також доступна фільтрація за розміром, матеріалом, сезоном тощо і сортування за популярністю, новизною або ціною.

### *3.4.4 Сторінка товару*

На сторінці окремого товару користувач має можливість обрати розмір, кількість, ознайомитися з описом, залишити або прочитати відгуки інших покупців. Тут також знаходиться кнопка "Додати до кошика" (див. Рис. 5).

### *3.4.5 Сторінка кошика*

У кошику зібрані всі обрані користувачем товари. Можна обрати тільки певні товари для купівлі, видалити зайві або змінити кількість перед оформленням замовлення. Після натискання кнопки "Придбати" обрані товари переміщуються до історії замовлень користувача (див. Рис. 6).

### *3.4.6 Сторінка акаунту*

На сторінці акаунту користувач може переглянути свої попередні замовлення, редагувати персональні дані або вийти з облікового запису. Це забезпечує повний контроль над історією покупок.

### *3.5 Панель доступності та реалізація принципів інклюзивного дизайну*

У межах реалізації принципів інклюзивного дизайну було впроваджено окрему панель доступності, що дозволяє користувачам змінювати візуальні параметри інтерфейсу відповідно до власних потреб. Зокрема, реалізовано перемикання між стандартним, контрастним і монохромним режимами відображення.

Крім того, сайт відповідає базовим рекомендаціям WCAG щодо доступності. Всі зображення супроводжуються атрибутами alt, які надають текстовий опис візуального контенту. Для інтерактивних елементів, таких як кнопки та іконки, були використані aria-label, що забезпечує їх коректне озвучення скрінрідерами. Семантична розмітка HTML із використанням тегів main, nav, header, footer та логічною структурою заголовків (h1 → h2 → h3) полегшує навігацію для користувачів з допоміжними технологіями.

Особливу увагу приділено адаптивності: при збільшенні інтерфейсу на 200% у середовищі браузера всі елементи залишаються функціональними, не виходять за межі екрана, і структура сторінки зберігається. Реалізовано візуальне фокусування для клавіатурної навігації — підсвічування активного елемента під час натискання клавіші Tab, що дозволяє пересуватись по сайту без миші.

Таким чином, реалізовані технічні та візуальні рішення підвищили доступність вебзастосунку для широкого кола користувачів, включно з людьми з

порушеннями зору, моторики та користувачами, які покладаються на клавіатурну навігацію або скрінрідери.

## РОЗДІ 4: Обробка зображень

### *4.1 Аналіз форматів зображень для вебу: вибір оптимального рішення*

У процесі розробки вебзастосунку Idillio особливу увагу було приділено оптимізації графічного контенту, оскільки зображення значно впливають на обсяг передаваних даних і швидкість завантаження сайту. З цією метою було проаналізовано найбільш актуальні формати зображень, які сьогодні застосовуються в інтерфейсах вебдодатків.

JPEG застосовується для стискання фотографій із мінімально помітною втратою якості. Його перевага полягає у високому ступені стиснення, що дозволяє зменшити вагу файлу. Однак формат не підтримує прозорість і зберігає втрати якості при повторному збереженні, тому не підходить для зображень інтерфейсу.

PNG забезпечує безвтратне стиснення і підтримує прозорість. Цей формат ідеальний для інтерфейсної графіки, логотипів та зображень з чіткими лініями. Основним його недоліком є велика вага файлів, що може негативно впливати на продуктивність сторінки.

SVG є векторним форматом і підходить для відображення логотипів, іконок та графіки, що потребує масштабування без втрати якості. Він підтримується більшістю браузерів і легко змінюється за допомогою CSS, але не призначений для складної растрової графіки або фотографій.

WebP розроблений спеціально для використання в інтернеті. Він дозволяє отримати суттєве зменшення ваги зображень порівняно з JPEG та PNG, підтримує прозорість, безвтратне та втратне стиснення. У тестах для Idillio WebP показав кращу швидкість завантаження, що позитивно вплинуло на зручність використання сайту та його SEO-показники.

Для забезпечення сумісності з усіма браузерами було реалізовано адаптивну підгрузку зображень за допомогою тегу `picture`, де як основний формат задається WebP, а як резервний — JPEG або PNG. Такий підхід дозволив поєднати переваги сучасного формату з максимальною доступністю для користувачів.

У підсумку WebP став основним форматом для зображень у проєкті Idillio завдяки своїй ефективності, гнучкості та високій продуктивності.

## *4.2 Оптимізація зображень у проєкті*

Оптимізація графічного контенту є невід’ємною частиною розробки швидкого та зручного вебзастосунку. У проєкті Idillio я реалізувала низку практичних заходів, спрямованих на зменшення ваги зображень, прискорення завантаження сторінок і покращення загального користувацького досвіду. Нижче наведено перелік застосованих технік:

### 1. Використання формату WebP

Усі зображення товарів та графічних блоків на сайті були переведені у формат WebP — сучасний формат, який забезпечує високу якість зображення при значно меншому обсязі файлу порівняно з JPEG або PNG. Це дозволило зменшити розмір більшості зображень на 30–40% без помітної втрати якості.

Для забезпечення сумісності з усіма браузерами я використовую тег `<picture>`, у якому передбачено варіант для старіших браузерів:

```
html
```

```
CopyEdit
```

```
<picture>
```

```
<source srcset="product.webp" type="image/webp">
```

```
  
</picture>
```

## 2. Видалення метаданих

Перед публікацією зображення проходили очищення від зайвих метаданих. Ці дані не мають жодної практичної цінності для кінцевого користувача, але можуть займати до 10% розміру файлу. Для автоматичного видалення метаданих я використовувала сервіси на кшталт [tinypng.com](http://tinypng.com) та вбудовані інструменти обробки в редакторах.

## 3. Стиснення без значної втрати якості

Кожне зображення проходило додаткове стиснення з урахуванням допустимого балансу між якістю та вагою. У результаті середній розмір однієї фотографії був зменшений із 500–700 КБ до 100–250 КБ. При цьому чіткість деталей залишалася достатньою для якісного візуального сприйняття.

## 4. Відкладене завантаження зображень

У проєкті було впроваджено техніку відкладеного завантаження, яка полягає у використанні атрибута `loading="lazy"` для всіх зображень. Такий підхід дозволяє завантажувати зображення лише тоді, коли вони потрапляють у зону видимості користувача. Це знижує початкове навантаження на мережу, пришвидшує рендеринг сторінки та покращує загальний користувацький досвід.

## 5. Адаптивне завантаження візуального контенту

Для оптимізації великих банерів і фонових зображень було використано атрибути `srcset` та `sizes`. Вони дозволяють браузеру автоматично обирати зображення відповідного розміру залежно від ширини екрана пристрою. Така реалізація зменшує обсяг завантажуваних даних, особливо для мобільних

користувачів, і запобігає використанню зайвих ресурсів. Це сприяє підвищенню швидкодії сайту на різних пристроях та відповідає сучасним вимогам до адаптивного дизайну

## 6. Використання SVG для елементів інтерфейсу

Усі піктограми, іконки та логотипи на сайті були реалізовані у форматі SVG. Це дозволило зберегти чіткість і масштабованість графіки незалежно від роздільної здатності екрана, а також зменшити загальний обсяг файлів завдяки векторному підходу.

### *4.3 Результати стискання та економія пам'яті*

Для перевірки ефективності обраних технік було проведено практичний експеримент на прикладі одного зображення товару. Початковий файл мав розмір 1.3 МБ у форматі JPEG. Після проходження всіх етапів оптимізації його розмір суттєво зменшився. Спочатку за допомогою сервісу tinypng.com вдалося зменшити обсяг до 547 КБ завдяки видаленню метаданих і базовому стисканню. Наступним кроком було використання інструмента Squoosh, де вручну зменшено фізичний розмір зображення до 800 пікселів по ширині, обрано формат WebP і застосовано додаткове стискання без значної втрати якості.

У результаті оптимізоване зображення мало обсяг лише 5 КБ, що означає зменшення на понад 99,6 відсотка. Це стало можливим завдяки тому, що початкове фото містило велику кількість порожнього білого простору, не потребувало високої деталізації, а також завдяки ефективному стисканню формату WebP для таких типів зображень.

Попри суттєве зменшення розміру, візуальна якість зображення залишилася задовільною: структура одягу збережена, зображення виглядає

чітко та добре відображається у картці товару. Така оптимізація дозволила значно скоротити час завантаження сторінки без негативного впливу на користувацький досвід.

#### *4.4 Порівняння швидкості завантаження до та після оптимізації зображень*

Щоб на практиці переконалися в ефективності впровадженої оптимізації, було проведено власне дослідження продуктивності. Метою було виміряти час завантаження сторінки з товарами до та після стискання зображень і переходу на формат WebP. Для цього було реалізовано React-хук `useLoadTime`, який фіксує час від моменту монтування компонента `ProductGrid` до завершення його відображення в інтерфейсі.

Було виконано два тести на одній і тій самій сторінці. До оптимізації, коли зображення мали формат PNG із середньою вагою приблизно 1.3 мегабайта, час завантаження компонента складав близько 1500 мілісекунд. Після переходу на формат WebP із очищенням від метаданих і стисненням до 5 кілобайт, час завантаження скоротився до приблизно 900 мілісекунд.

Це свідчить про зменшення часу завантаження приблизно на 40 відсотків. Такий результат є суттєвим покращенням для загального комфорту користувача. Менше навантаження на мережу дозволяє швидше відображати вміст, що особливо важливо для мобільних користувачів і в умовах повільного інтернет-з'єднання.

Отже, навіть просте зменшення ваги зображень без втрати помітної якості позитивно впливає на продуктивність інтерфейсу та виправдане з точки зору UX.

## Висновки

У результаті виконання курсової роботи було реалізовано повний цикл створення вебзастосунку для онлайн-продажу одягу, який поєднує естетику, функціональність та інклюзивність. На основі аналізу цільової аудиторії були розроблені user persona, що дозволило сформулювати вимоги до інтерфейсу і структури сайту відповідно до очікувань реальних користувачів. Ретельно підібрані технології — React, SCSS, Vite, Firebase — забезпечили високу продуктивність, масштабованість і зручність роботи з даними.

Під час розробки особливу увагу приділено принципам сучасного дизайну, що дозволило досягти візуальної узгодженості та зручної навігації. Було впроваджено оптимізацію зображень із використанням форматів WebP, методів стиснення та відкладеного завантаження, що суттєво покращило продуктивність і швидкість завантаження сторінок.

Завдяки впровадженню SPA-архітектури користувачі можуть взаємодіяти із сайтом без перезавантаження сторінок, що покращує UX. Враховано також вимоги доступності, що робить проєкт придатним для ширшої аудиторії, включно з людьми з порушеннями зору чи моторики.

Отже, в межах курсової було успішно реалізовано сучасний, адаптивний та зручний вебзастосунок, який відповідає технічним і дизайнерським вимогам, а також демонструє застосування комплексного підходу до розробки сайтів на практиці.

## Список використаних джерел

1. Web Content Accessibility Guidelines (WCAG) 2.1:  
<https://www.w3.org/Translations/WCAG21-ua/>
2. Принципи дизайну:  
[https://articles.basov.com.ua/printsipy-dizayna.html#google\\_vignette](https://articles.basov.com.ua/printsipy-dizayna.html#google_vignette)
3. React.js documentation: <https://react.dev/learn>
4. Sass documentation: <https://sass-lang.com/documentation/>
5. Vite documentation: <https://vite.dev/guide/>
6. Firebase developer documentation: <https://firebase.google.com/docs>
7. SPA (Single-page application):  
<https://developer.mozilla.org/en-US/docs/Glossary/SPA>
8. Формати зображень для вебу - які краще використовувати та для чого:  
<https://dementor.org/blog/03-formaty-zobrazhen-dlya-webu-jpeg-png-svg-webp-avif-yaki-krashche>
9. Figma dev mode documentation:  
<https://help.figma.com/hc/en-us/sections/15023066873239>
10. Як оптимізувати зображення для сайту:  
<https://voll.com.ua/uk/blog/yak-optimizuvati-zobrazhennya-dlya-sajtu>
11. Intertop:  
[https://intertop.ua/uk-ua/?srsltid=AfmBOoqsN0a-\\_RH5IrCEOLY0BurO0fckTjIMhllA-U7u1h0DSWsPV2y5](https://intertop.ua/uk-ua/?srsltid=AfmBOoqsN0a-_RH5IrCEOLY0BurO0fckTjIMhllA-U7u1h0DSWsPV2y5)
12. Answer:  
[https://answer.ua/?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=search\\_brand\\_Answear%20%5BMaxClicks%5D&utm\\_id=21646667206&gad\\_source=1&gbraid=0AAAAAB8Djs0C7lB6YlvOaEgZWAE36lxAK&gclid=Cj0KCQjww-HABhCGARIsALLO6XwReboorFcUB5CU-IH2PyhQ\\_m6wBulT-SDX7-Q5lIBPVlywjznJMZMaAkuBEALw\\_wcB](https://answer.ua/?utm_source=google&utm_medium=cpc&utm_campaign=search_brand_Answear%20%5BMaxClicks%5D&utm_id=21646667206&gad_source=1&gbraid=0AAAAAB8Djs0C7lB6YlvOaEgZWAE36lxAK&gclid=Cj0KCQjww-HABhCGARIsALLO6XwReboorFcUB5CU-IH2PyhQ_m6wBulT-SDX7-Q5lIBPVlywjznJMZMaAkuBEALw_wcB)

