

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»»

КВАЛІФІКАЦІЙНА РОБОТА

Тема: «Вебсайт інтернет-магазину комп'ютерних комплектуючих з використанням багатокритеріальних фільтрів та рекомендаційної системи (комплексна робота)»

Ступінь вищої освіти – бакалавр
Спеціальність – 122 «Комп'ютерні науки»
Освітня програма «Комп'ютерні науки»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконав: здобувач 4 курсу
групи КН-21

Максим ДАВИДЕНКО

Керівники: доцент кафедри комп'ютерних наук,
кандидат технічних наук

Олександр ПОЛІЩУК

асистент кафедри комп'ютерних наук

Микита ФЕШЕНКО

Засвідчую, що кваліфікаційна
робота оформлена відповідно до
ДСТУ 3008:2015 та не містить
запозичень з праць інших авторів
без відповідних посилань.

Здобувач: _____
(підпис)

м. Київ – 2025 рік

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»»

ЗАТВЕРДЖУЮ:
завідувач кафедри
комп'ютерних наук
_____Сергій МІЧКІВСЬКИЙ
« ____ » ____ 20 ____ р

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
Давиденко Максим Олександрович

Тема роботи	Вебсайт інтернет-магазину комп'ютерних комплектуючих з використанням багатокритеріальних фільтрів та рекомендаційної системи (комплексна робота)
Номер та дата наказу про затвердження теми	№121-7 від 24 грудня 2024 року
Коротка постановка завдання	Розробити функціональний вебсайт інтернет-магазину комп'ютерних комплектуючих, який забезпечує зручну навігацію для користувачів, реалізацію багатокритеріальних фільтрів для точного підбору товарів та впровадження рекомендаційної системи для персоналізованих пропозицій.
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	React. Бібліотека зі створення UI для вебу і нативних платформ. [Електронний ресурс]. URL: https://uk.react.dev (дата звернення: 07.04.2025). Next.js. Welcome to the Next.js documentation. [Електронний ресурс]. URL: https://nextjs.org/docs (дата звернення: 07.04.2025).
Вимоги до кваліфікаційної роботи	Кваліфікаційна робота має містити теоретичне, системотехнічне або експериментальне дослідження за темою роботи, яку слід розглядати як складне спеціалізоване завдання або практичну проблему в галузі комп'ютерних наук, яка характеризується комплексністю та невизначеністю умов і потребує застосування теорій і методів інформаційних технологій.

Дата видачі завдання «27» грудня 2024 р.

Керівник

Олександр ПОЛЩУК

Керівник

Микита ФЕЩЕНКО

Здобувач освітнього ступеня бакалавра

Максим ДАВИДЕНКО

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
Підготовчий етап			
1	Вибір напрямку дослідження	02.12.2024 р.	<i>виконано</i>
2	Формування теми та призначення керівника	16.12.2024 р.	<i>виконано</i>
3	Затвердження теми кваліфікаційної роботи	23.12.2024 р.	<i>виконано</i>
4	Затвердження завдання на кваліфікаційну роботу	27.12.2024 р.	<i>виконано</i>
Основний етап			
5	Розробка концепції кваліфікаційної роботи	13.01.2025 р.	<i>виконано</i>
6	Підбір та вивчення джерел інформації з напрямку дослідження. Огляд існуючих аналогів	20.01.2025 р.	<i>виконано</i>
7	Затвердження розширеної постановки завдання. Підготовка та подання керівникові розділу 1 кваліфікаційної роботи	10.03.2025 р.	<i>виконано</i>
8	Проектування. Підготовка та подання керівникові розділу 2 кваліфікаційної роботи	24.03.2025 р.	<i>виконано</i>
9	Підготовка доповіді для експертизи стану виконання кваліфікаційної роботи (проміжний контроль)	31.03-04.04.2025 р.	<i>виконано</i>
10	Реалізація. Підготовка та подання керівникові розділу 3 кваліфікаційної роботи	07.04.2025 р.	<i>виконано</i>
11	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи	14.04.2025 р.	<i>виконано</i>
12	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника та представлення керівникові доопрацьованого варіанту кваліфікаційної роботи	21.04.2025 р.	<i>виконано</i>
Завершальний етап			
13	Представлення рукопису для перевірки на плагіат	28.04-04.05.2025 р.	<i>виконано</i>
14	Підготовка презентації та доповіді на передзахист	05.05-11.05.2025 р.	<i>виконано</i>
15	Передзахист кваліфікаційної роботи	12.05-16.05.2025 р.	<i>виконано</i>
16	Доопрацювання роботи за результатами передзахисту	19.05-06.06.2025 р.	<i>виконано</i>
17	Експертиза роботи керівником та зовнішнім експертом	09.06-15.06.2025 р.	<i>виконано</i>
18	Доопрацювання доповіді та презентації для захисту	09.06-15.06.2025 р.	<i>виконано</i>
19	Захист кваліфікаційної роботи	16.06-22.06.2025 р.	<i>виконано</i>

Керівник

Керівник

Здобувач освітнього ступеня бакалавра

Олександр ПОЛЩУК

Микита ФЕЩЕНКО

Максим ДАВИДЕНКО

Давиденко М.О. Вебсайт інтернет-магазину комп'ютерних комплектуючих з використанням багатокритеріальних фільтрів та рекомендаційної системи (комплексна робота)

Пояснювальна записка кваліфікаційної роботи за спеціальністю 122 – Комп'ютерні науки (освітня програма – Комп'ютерні науки) СО Бакалавр. – ВНЗ «Університет економіки та права «КРОК», Навчально-науковий інститут інформаційних та комунікаційних технологій, кафедра комп'ютерних наук, Київ, 2025.

Досліджено процес створення функціонального вебсайту з електронної комерції комп'ютерних комплектуючих, який передбачає зручну навігацію для користувачів, реалізацію багатокритеріальних фільтрів для точного підбору товару та систему рекомендацій для персональних пропозицій.

Ключові слова: вебсайт, інтернет-магазин, комп'ютерні комплектуючі.

Табл. 2. Рис. 20. Бібліограф.: 30 найм.

Davydenko M.O. Website of an online store of computer components using multi-criteria filters and recommendation system (complex work)

Explanatory note of the qualification work in the specialty 122 – Computer Science (educational program – Computer Science) Bachelor's degree. – Higher Educational Institution "University of Economics and Law "KROK", Educational and Scientific Institute of Information and Communication Technologies, Department of Computer Science, Kyiv, 2025.

The article investigates the process of creating a functional website for e-commerce of computer components, which provides convenient navigation for users, implementation of multi-criteria filters for accurate product selection and a recommendation system of personal offers.

Keywords: website, online store, computer components.

Tabl. 2. Fig. 20. Bibliography: 30 Items.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	6
ВСТУП.....	7
РОЗДІЛ 1 ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ІНТЕРНЕТ- МАГАЗИНУ «TAGSYNC».....	9
1.1 ПРЕДМЕТНА ОБЛАСТЬ.....	9
1.2 ОГЛЯД АНАЛОГІВ.....	10
1.3 ПОСТАНОВКА ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ	15
Висновок до розділу 1.....	16
РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ «TAGSYNC».....	18
2.1 МОДЕЛЮВАННЯ ПОВЕДІНКИ ПРОДУКТУ.....	18
2.2 МОДЕЛЮВАННЯ UI/UX ДИЗАЙНУ.....	21
2.3 ПРОТОТИПУВАННЯ UI/UX ДИЗАЙНУ ПРОДУКТУ.....	24
Висновок до розділу 2.....	25
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ ІНТЕРНЕТ- МАГАЗИНУ «TAGSYNC».....	26
3.1 РЕАЛІЗАЦІЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ПРОДУКТУ	26
3.2 ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	37
3.3 ВИКОРИСТАННЯ ПРОГРАМНОГО ПРОДУКТУ	38
Висновки до розділу 3.....	44
ВИСНОВКИ	45
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	47
ДОДАТОК А ПОКАЗНИКИ ПРОДУКТИВНОСТІ.....	50

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Footer – так званий підвал сайту, його нижня частина, куди власники ресурсів заносять достатньо важливу інформацію, проте вона одночасно не відноситься до пункту першорядної важливості [1].

Header – найвища частина веб-сторінки. Header є областю, яка знаходиться над частиною веб-сторінки, де знаходиться контент. У шапці сайту зазвичай розміщують логотип, назву, основні контактні дані (адреса, телефон та електронна пошта), меню навігації та інші важливі елементи [2].

Бургер-меню – це іконка, яка зазвичай складається з трьох горизонтальних ліній. Під час кліку на цю іконку, користувачу відкривається меню з навігаційними посиланнями. Ця концепція особливо популярна на мобільних пристроях, де екранний простір обмежений [3].

Слайдер – це динамічний елемент дизайну, який показує варіативний контент, такий як зображення, текст та посилання. Ці елементи можуть змінюватися як автоматично, так і за допомогою ручного перемикання [4].

A11Y або accessibility (рівень доступності) у веб-розробці означає надання можливості якомога більшій кількості людей користуватися веб-сайтами, навіть якщо їхні можливості певним чином обмежені [5].

Рендер – це відмальовування коду вебдокумента в інтерактивну вебсторінку, яку остаточно видно у браузері [6].

ВСТУП

Актуальність теми знаходиться на перетині двох важливих сучасних сфер: електронної комерції та інформаційних технологій. Сьогодні інтернет-магазини є важливим інструментом для задоволення потреб споживачів у зручному та швидкому доступі до товарів. Зокрема, ринок комп'ютерних комплектуючих характеризується великою кількістю товарів, що ускладнює вибір для користувачів.

Багатокритеріальні фільтри є важливим інструментом, який дозволяє споживачам швидко знайти потрібні товари, враховуючи їхні потреби та вподобання.

Зважаючи на викладене вище, тема розробки функціонального вебсайту інтернет-магазину комп'ютерних комплектуючих є актуальною, оскільки спрямована на покращення користувацького досвіду та збільшення конкурентоспроможності на ринку електронної комерції.

Метою проєкту є розробка вебсайту інтернет-магазину комп'ютерних комплектуючих, який забезпечує зручну навігацію, реалізацію багатокритеріальних фільтрів для точного підбору товарів і впровадження рекомендаційної системи для персоналізованих пропозицій.

Завдання дослідження:

- сформулювати вимоги до функціоналу інтернет-магазину й очікування користувачів;
- виконати аналіз аналогів інтернет-магазинів у сфері комп'ютерних комплектуючих, на предмет переваг та недоліків;
- дослідити алгоритми та принципи роботи багатокритеріальних фільтрів, обрати оптимальні підходи для їх реалізації;
- розробити вебсайт із урахуванням високого рівня доступності, SEO та продуктивності.

Об'єктом дослідження є візуальна частина вебсайту для інтернет-магазину комп'ютерних комплектуючих.

Предметом дослідження є сфера електронної комерції з продажу комп'ютерних комплектуючих.

Практична цінність проєкту полягає у розробці візуальної частини сайту, з якою взаємодіють користувачі інтернет-магазину, що буде якісним і зручним у порівнянні з існуючими аналогами.

Структура та обсяг пояснювальної записки. Пояснювальна записка до проєкту складається зі вступу, трьох розділів, висновків, списку посилань (30 найменувань) та 1 додаток. Пояснювальна записка містить 20 рисунків, 2 таблиці. Загальний обсяг пояснювальної записки складає 50 сторінок, основний зміст викладено на 46 сторінках.

РОЗДІЛ 1

ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ІНТЕРНЕТ-МАГАЗИНУ «TAGSYNC»

1.1 Предметна область

Інтернет-магазин – онлайн-майданчик для продажу товарів через інтернет. Клієнти можуть переглядати асортимент, обирати те, що їм потрібно, замовляти і здійснювати оплату прямо на вебсайті. Додатково інтернет-магазини часто пропонують різні функції та інструменти, такі як особистий кабінет, відстеження статусу замовлення, історія покупок, накопичення бонусів тощо [7].

Інтернет-магазини є важливим сегментом сучасної електронної комерції, що забезпечують споживачам швидкий доступ до широкого спектра товарів, включаючи комп'ютерні комплектуючі. Ринок комп'ютерних комплектуючих є динамічним і конкурентним, оскільки технології постійно розвиваються, а споживачі прагнуть отримати найкраще обладнання для своїх потреб, зокрема для роботи, ігор та інших завдань.

Основні виклики, з якими стикаються користувачі при виборі комп'ютерних комплектуючих, включають:

- велику кількість доступних товарів та їхню різноманітність;
- складність у порівнянні характеристик комплектуючих;
- відсутність ефективних механізмів пошуку та фільтрації;
- обмежені можливості персоналізованих рекомендацій.

Для розв'язання цих проблем сучасні інтернет-магазини впроваджують UI/UX рішення, які покращують користувацький досвід. Важливими елементами таких рішень є:

- інтуїтивно зрозумілий інтерфейс користувача (UI), що забезпечує зручну навігацію;
- багатокритеріальні фільтри, які допомагають швидко знайти потрібний товар;

- рекомендаційні системи, що адаптуються до потреб конкретного користувача;
- адаптивність інтерфейсу під мобільні пристрої;
- високий рівень доступності (A11Y) для всіх категорій користувачів.

1.2 Огляд аналогів

Для аналізу сучасних підходів до UI/UX-дизайну та функціональності інтернет-магазинів у сфері комп'ютерних комплектуючих було проведено огляд декількох конкурентних вебсайтів.

GameHall – це інтернет-магазин, що спеціалізується на продажі ігрової техніки та комп'ютерних комплектуючих (рис. 1.1).

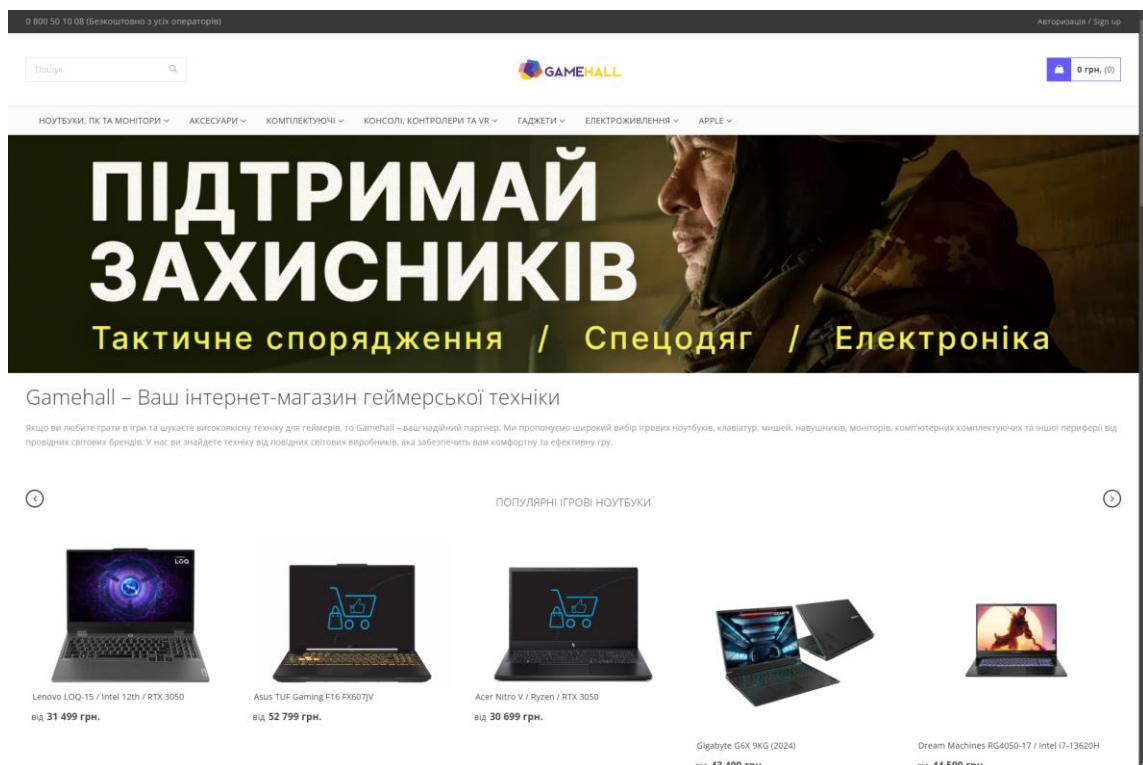


Рисунок 1.1 – Головна сторінка сайту GameHall.

Джерело: GameHall [8]

Недоліки:

- на сайті присутня лише світла тема;

- сайтом абсолютно не можливо користуватися за допомогою клавіатури;
- поганий контраст кольорів основного тексту;
- багато помилок на тест A11Y за допомогою спеціального розширення;
- розмір шрифту не адаптується під розмір шрифту браузера;
- header не є закріпленим у горі сторінки та під час гортання сторінки не прокручується разом з користувачем;
- якщо відкрити бургер-меню на мобільній версії, то на desktop-версії він залишиться відкритим;
- якщо відкрити фільтри на мобільній версії, то вони на desktop-версії з'являться зайві фільтри;
- текст елементів слайдера можуть знаходитись на різних рівнях;
- відсутність кнопки для скидання всіх фільтрів;
- неможливість додати товар у кошик, не переходячи на іншу сторінку;
- після зміни кількості товару, потрібно вручну оновити кошик;
- на сайті присутня лише одна мова (українська).

Переваги:

- фільтри та сортування зберігаються після оновлення сторінки;
- присутні анімації.

CompX – це інтернет-магазин комп'ютерної техніки з можливістю конфігурування ПК (рис. 1.2).

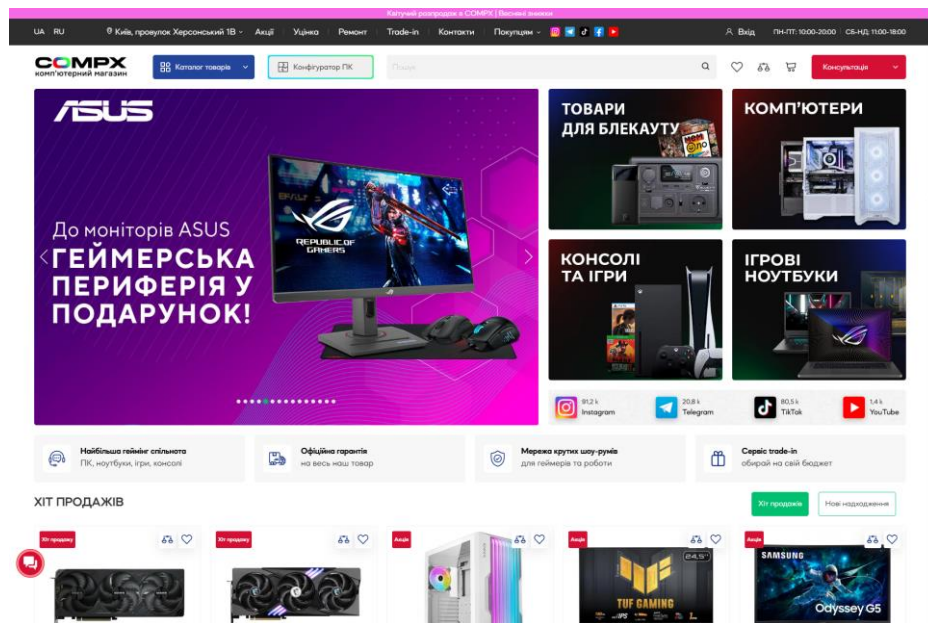


Рисунок 1.2 – Головна сторінка сайту Compx.

Джерело: Compx [9]

Недоліки:

- на сайті присутня лише світла тема
- слайдери не керуються клавіатурою;
- багато помилок на тест A11Y за допомогою спеціального розширення;
- у спливаючому вікні можна гортати сторінку клавішею Tab;
- на високих екранах footer не зафіксований внизу;
- якщо відкрити бургер-меню на мобільній версії, то на desktop-версії він залишиться відкритим;
- відсутність анімацій;
- відсутність відступів між деякими елементами;
- відсутня найпопулярніша мова – англійська.

Переваги:

- натискаючи клавішу Tab можна переміщатись по сайту та бачити на якому елементі наразі фокус;

- присутня кнопка «вгору», що автоматично після натискання гортає вгору сторінку;
- header є закріпленим у горі сторінки та під час гортання сторінки не прокручується разом з користувачем;
- розмір шрифту адаптується від розміру шрифту браузера;
- фільтри та сортування зберігаються після оновлення сторінки;
- можна обрати одразу декілька фільтрів і потім здійснити пошук;
- можливість додати товар у кошик, не переходячи на іншу сторінку;
- можливість змінити кількість товару декілька разів, не чекаючи кожен раз на оновлення;
- на сайті присутні дві мови.

LuckyLink – це інтернет-магазин ігрової техніки (рис. 1.3).

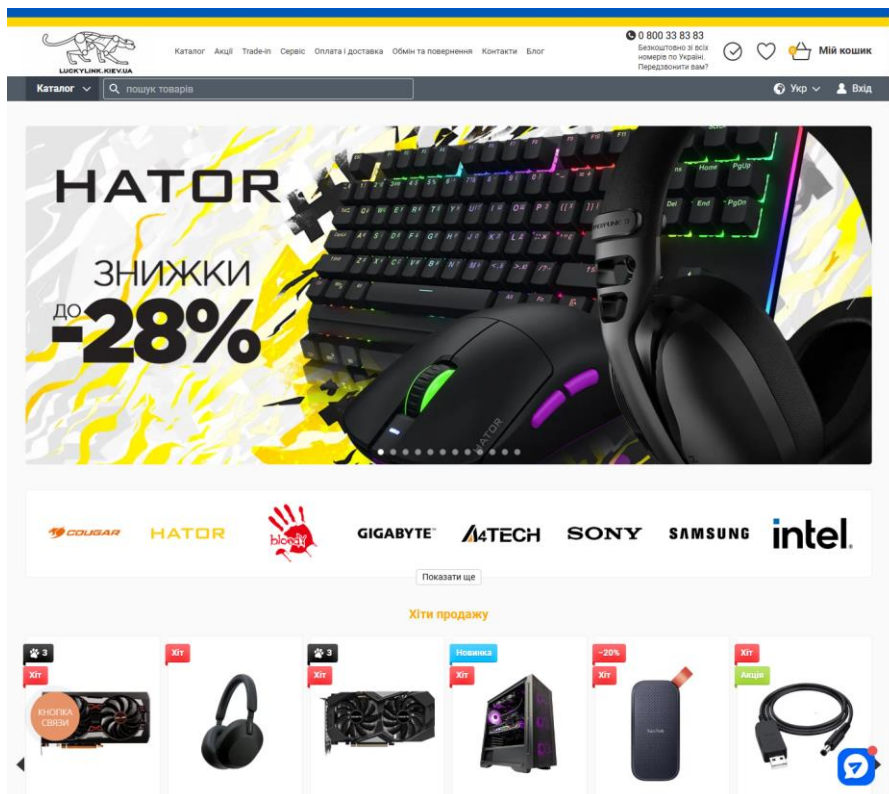


Рисунок 1.3 – Головна сторінка сайту LuckyLink.

Джерело: LuckyLink [10]

Недоліки:

- на сайті присутня лише світла тема;
- слайдери не керуються клавіатурою;
- багато помилок на тест A11Y за допомогою спеціального розширення;
- у спливаючому вікні можна сфокусуватись на елементах сторінки клавішею Tab;
- розмір шрифту не адаптується під розмір шрифту браузера;
- кнопка «Почати спілкування» перекриває кнопку «вгору»;
- header не є закріпленим у горі сторінки та під час гортання сторінки не прокручується разом з користувачем;
- неможливість обрати одразу декілька фільтрів: після встановлення фільтра йде запит, тимчасово блокуючи фільтри;
- деякі частини сайту мають одну мову;
- відсутня найпопулярніша мова – англійська.

Переваги:

- присутність анімацій;
- присутня кнопка «вгору»;
- фільтри та сортування зберігаються після оновлення сторінки;
- можливість додати товар у кошик, не переходячи на іншу сторінку;
- можливість змінити кількість товару декілька разів, не чекаючи кожен раз на оновлення;
- на сайті присутні дві мови.

Аналізуючи конкуренти, можна зробити висновок, що сучасні інтернет-магазини мають різні рівні доступності та зручності навігації. Основні недоліки, які необхідно врахувати при розробці власного проєкту, це покращення A11Y, забезпечення інтуїтивної навігації та вдосконалення системи рекомендацій.

1.3 Постановка завдання на кваліфікаційну роботу

Беручи до уваги результати аналізу предметної області та конкурентних аналогів, основне завдання даної роботи полягає у створенні візуальної складової вебсайту для інтернет-магазину комп'ютерних комплектуючих. Сайт має забезпечувати:

- інтуїтивно зрозумілий UI/UX;
- потужну систему фільтрації товарів за різними критеріями;
- персоналізовані рекомендації на основі історії переглядів та покупок;
- адаптивність під мобільні пристрої;
- високий рівень доступності (A11Y);
- темну і світлу теми для зручності користувачів.

Перейдемо до функціональних та нефункціональних вимог (табл. 1.1) та вхідних та вихідних даних (табл. 1.2).

Таблиця 1.1 – Функціональні та нефункціональні вимоги

Функціональні вимоги	Нефункціональні вимоги
Пошук товарів за ключовими словами	UI має бути простим і зручним
Багатокритеріальні фільтри	Адаптивність для мобільних пристроїв
Рекомендаційна система	Підтримка темної та світлої тем
Реєстрація та авторизація користувачів	Високий рівень доступності (A11Y)
Кошик та оформлення замовлення	Висока швидкість завантаження сторінок

Таблиця 1.2 – Вхідні та вихідні дані

Вхідні дані	Вихідні дані
Інформація про товари	Готовий UI/UX прототип вебсайту
Категорії та фільтри товарів	Вихідні дані у вигляді функціональних блоків
Дані користувачів	Персоналізовані рекомендації

Висновок до розділу 1

В результаті проведеного аналізу предметної області та конкурентів, було виявлено кілька ключових проблем, з якими стикаються користувачі при виборі комп'ютерних комплектуючих у інтернет-магазинах. Основні складнощі в порівнянні характеристик товарів, не зовсім ефективні механізми пошуку та фільтрації, відсутність персоналізованих рекомендацій, а також різний рівень доступності (A11Y) й адаптивності сайтів.

Огляд різних рішень, таких як GameHall, CompX і LuckyLink, дав змогу виявити як сильні, так і слабкі сторони сучасних інтернет-магазинів. Серед основних недоліків можна відзначити проблеми з доступністю (A11Y), відсутність темної теми, складну навігацію та недостатню інтерактивність елементів. З іншого боку, позитивні моменти конкурентних платформ включають наявність кнопки для гортання вгору, збереження фільтрів після оновлення сторінки, можливість змінювати кількість товарів без перезавантаження та підтримку кількох мов.

На основі отриманих даних було сформульовано вимоги для розробки Frontend вебсайту інтернет-магазину комп'ютерних комплектуючих. Основна увага приділяється покращенню користувацького досвіду через впровадження:

- інтуїтивно зрозумілого UI/UX-дизайну;
- розширених можливостей пошуку та багатокритеріальної фільтрації товарів;

- персоналізованих рекомендацій, які враховують історію переглядів і покупок;
- адаптивного інтерфейсу для коректного відображення на мобільних пристроях;
- покращеної доступності для всіх категорій користувачів (A11Y);
- підтримки світлої та темної теми для зручності використання.

Функціональні та нефункціональні вимоги до майбутнього вебсайту були ретельно визначені. Серед них – підтримка пошуку за ключовими словами, авторизація користувачів, система фільтрації товарів, а також високий рівень зручності інтерфейсу та адаптивності. Впровадження цих рішень допоможе підвищити ефективність роботи платформи, залучити більше користувачів і покращити їхній досвід взаємодії з сайтом.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ «TAGSYNC»

2.1 Моделювання поведінки продукту

Для створення якісного дизайну інтернет-магазину необхідно чітко визначити портрет цільової аудиторії. Це дозволяє глибше зрозуміти потреби користувачів, їхні очікування та сценарії поведінки. Такий підхід забезпечує створення максимально зручного, логічного та ефективного інтерфейсу, що відповідає реальним потребам цільових користувачів.

Для глибшого розуміння поведінки користувачів і логіки взаємодії з системою було побудовано низку діаграм, які допомогли систематизувати вимоги та сформуванати архітектуру інтерфейсу.

Use case діаграма дозволила визначити основні сценарії використання системи, такі як перегляд товарів, застосування фільтрів, додавання до кошика, оформлення замовлення тощо (рис. 2.1). Завдяки цьому було зрозуміло, які функціональні можливості є пріоритетними для реалізації.

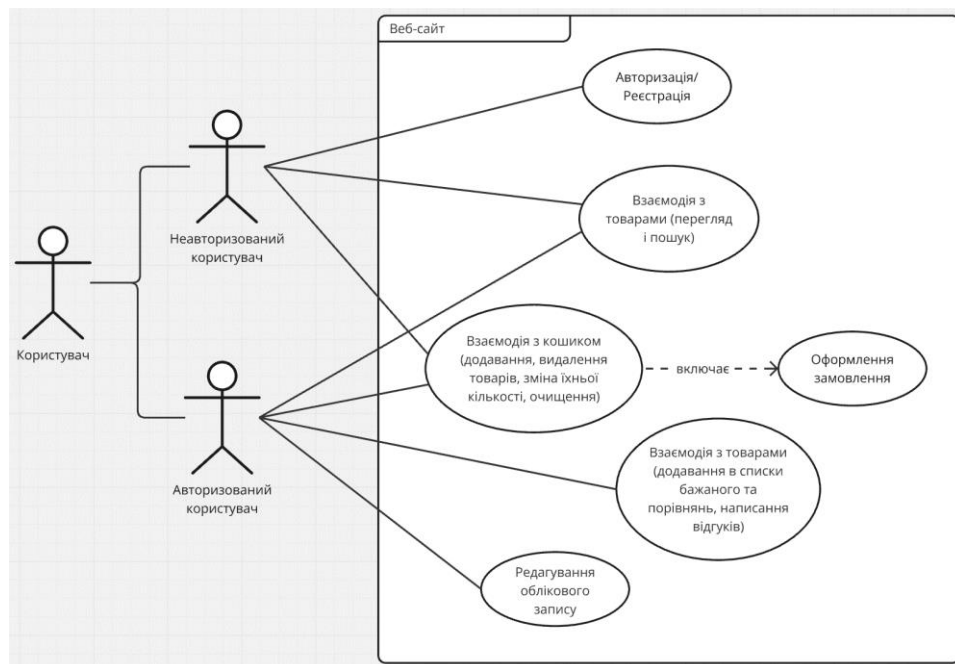


Рисунок 2.1 – Use case діаграма

Джерело: розроблено автором

Діаграма діяльності дала змогу візуалізувати процеси, які відбуваються під час взаємодії користувача з сайтом (рис. 2.2). Зокрема, були описані послідовності дій при пошуку та виборі товару, що допомогло оптимізувати навігацію та мінімізувати кількість кроків для досягнення мети.

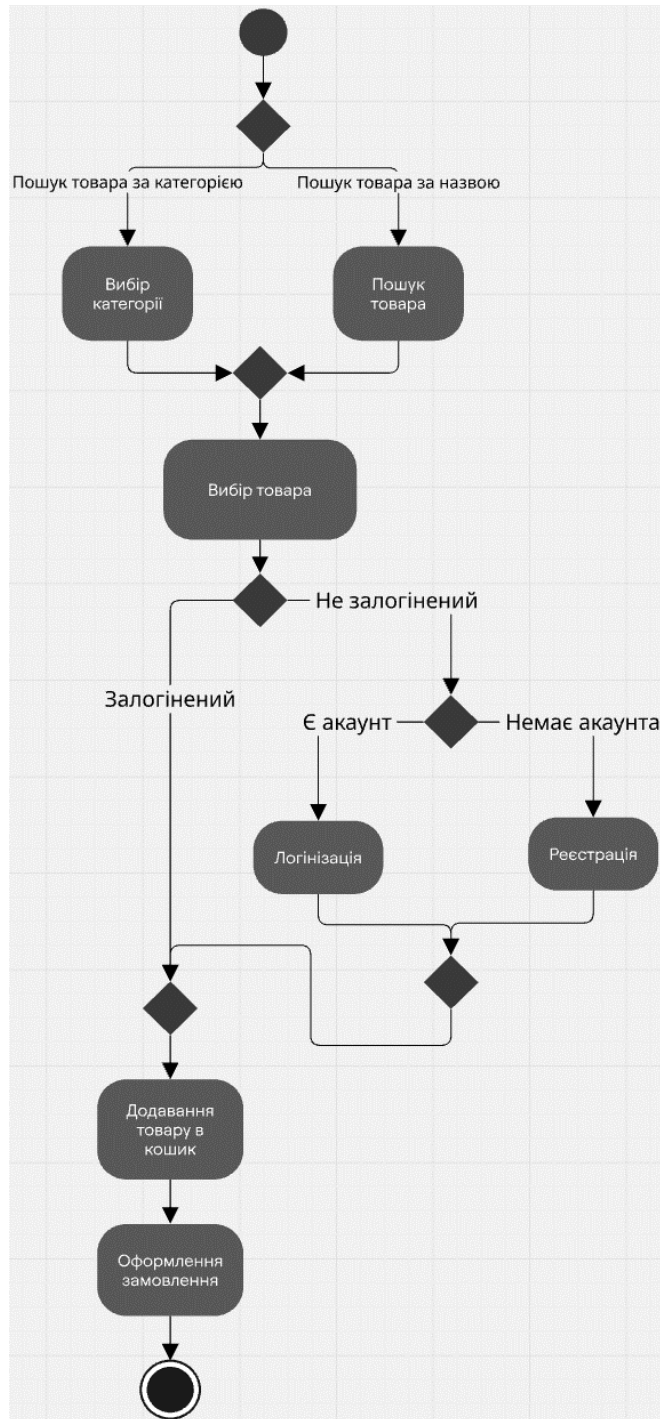


Рисунок 2.2 – Діаграма діяльності

Джерело: розроблено автором

Діаграма послідовності допомогла змодельовати послідовність взаємодії між користувачем і основними компонентами системи (рис. 2.3). Це дало змогу продумати логіку запитів до серверу, взаємодію з базою даних, обробку та відображення компонентів на клієнті.

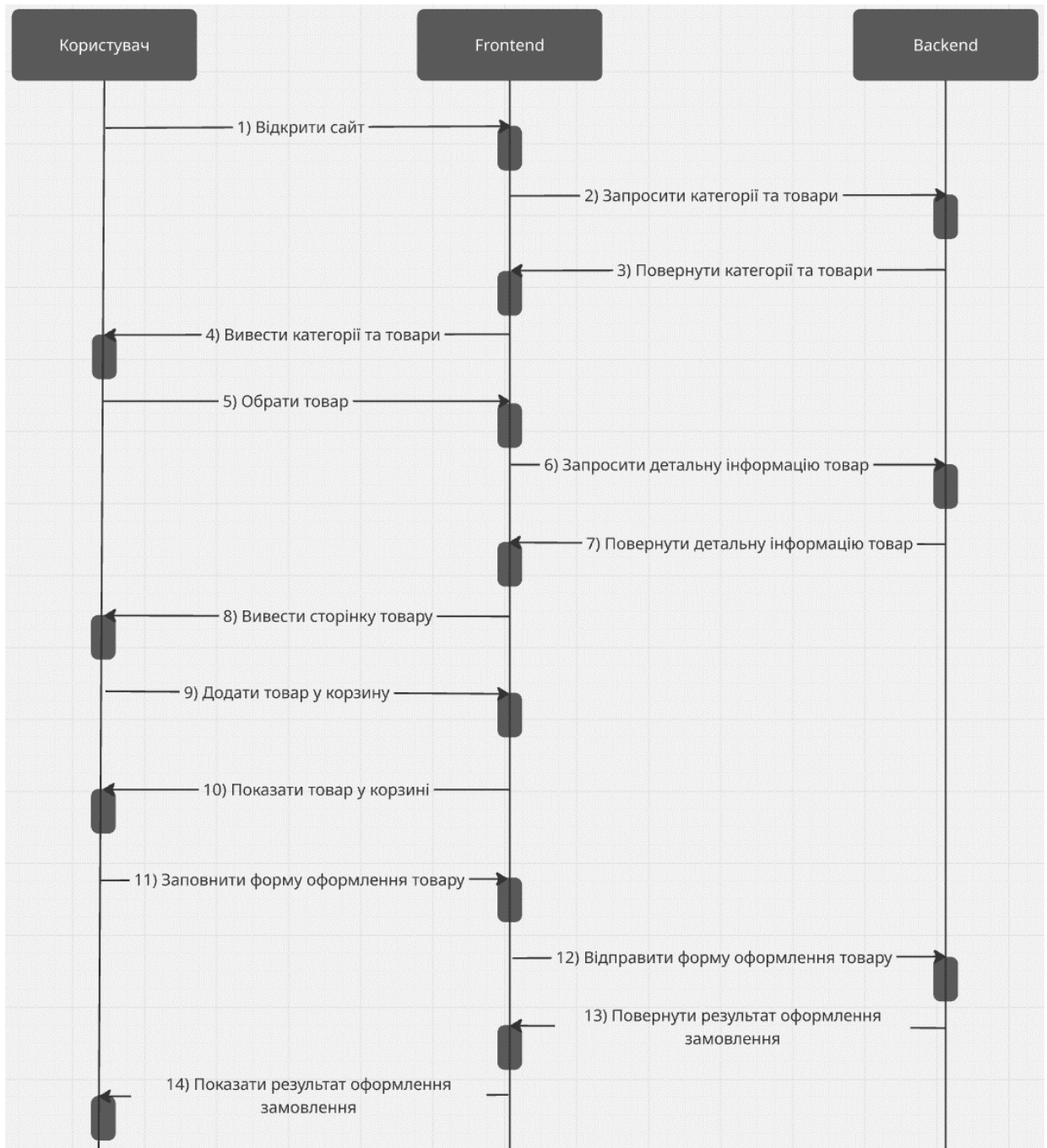


Рисунок 2.3 – Діаграма послідовності

Джерело: розроблено автором

Використання цих діаграм забезпечило структурований і системний підхід до проєктування інтерфейсу та логіки роботи системи. Завдяки їм вдалося не лише чітко окреслити функціональні можливості майбутнього продукту, але й забезпечити злагоджену взаємодію між усіма його складовими. Діаграми допомогли виявити потенційні проблемні місця ще на етапі планування, що дозволило уникнути помилок під час реалізації та зменшити витрати часу на подальші доопрацювання.

2.2 Моделювання UI/UX дизайну

Моделювання UI/UX дизайну є ключовим етапом у розробці інтерфейсу інтернет-магазину. Цей процес визначає, як саме користувач взаємодітиме з продуктом – від першого враження до виконання цільової дії, наприклад, покупки. Саме на цьому етапі формується не лише зовнішній вигляд інтерфейсу, а й логіка його роботи, з урахуванням потреб, очікувань та поведінки цільової аудиторії.

UI (User Interface) зосереджується на візуальних елементах: компонуванні сторінок, типографії, кольорах, іконках та графіці.

UX (User Experience) охоплює глибший рівень – це досвід користувача під час навігації сайтом, зручність пошуку інформації, швидкість виконання дій, задоволення від використання ресурсу.

Модель потенційного користувача, що дозволила визначити основні характеристики цільової аудиторії: вік, інтереси, технічну обізнаність, поведінку в інтернеті та очікування від взаємодії з сайтом (рис. 2.4). Це допомогло адаптувати функціональність і дизайн під потреби конкретних користувачів.

Мапа емпатії надала глибше розуміння емоційного стану користувачів (рис. 2.5). Вона показала, що користувачі думають, відчувають, роблять і говорять. Це дозволило врахувати їхні мотивації, страхи та очікування.

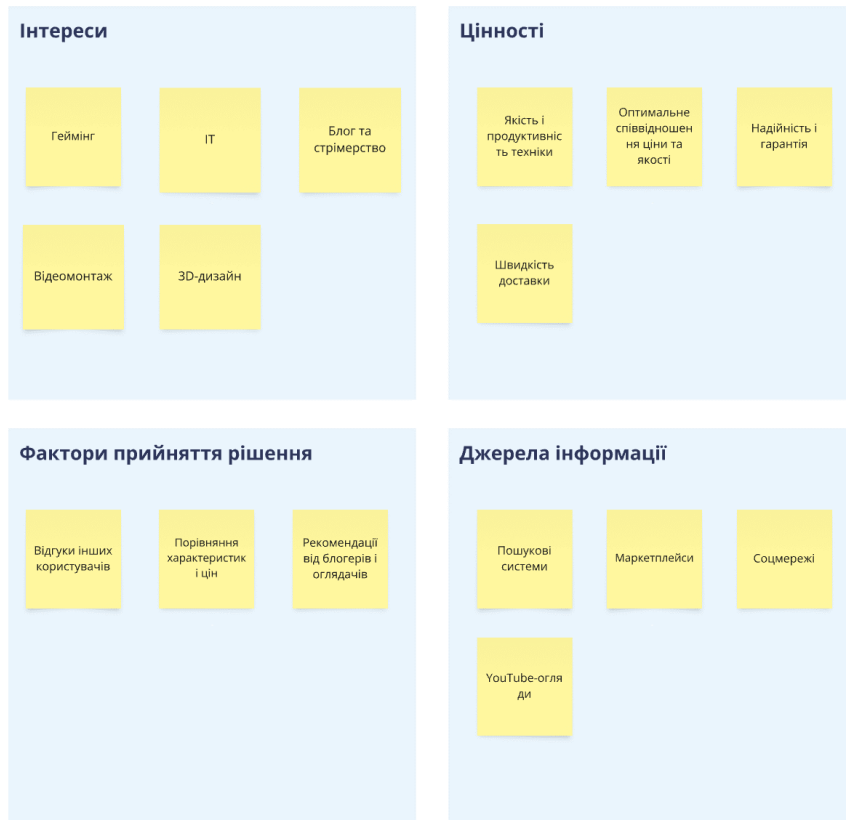


Рисунок 2.4 – Модель потенційного користувача

Джерело: розроблено автором

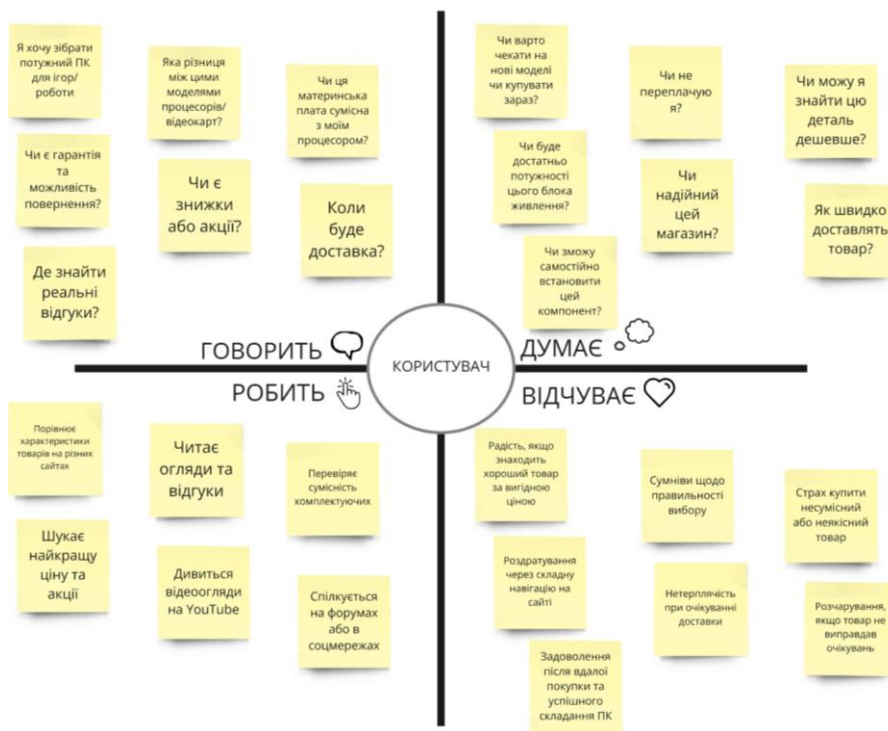


Рисунок 2.5 – Мапа емпатії

Джерело: розроблено автором

На мапі подорожі клієнта візуалізовано повний шлях користувача від першого контакту з сайтом до оформлення замовлення, що допомогло виявити ключові точки взаємодії, можливі проблеми та шляхи покращення досвіду користування (рис. 2.6).

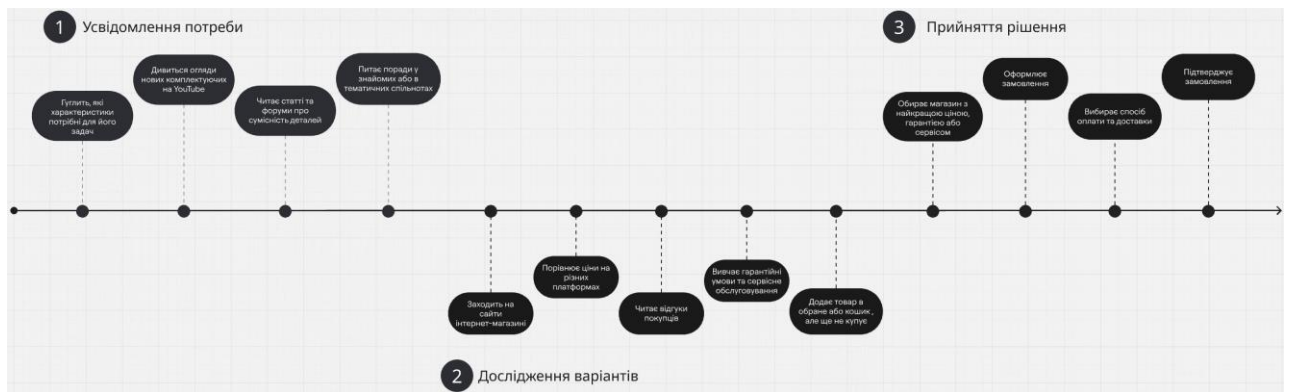


Рисунок 2.6 – Мапа подорожі клієнта

Джерело: розроблено автором

Карта сайту відобразила структуру сайту та ієрархію сторінок (рис. 2.7). Забезпечила логічну навігацію, зрозумілий розподіл інформації та зручний доступ до основних функцій.

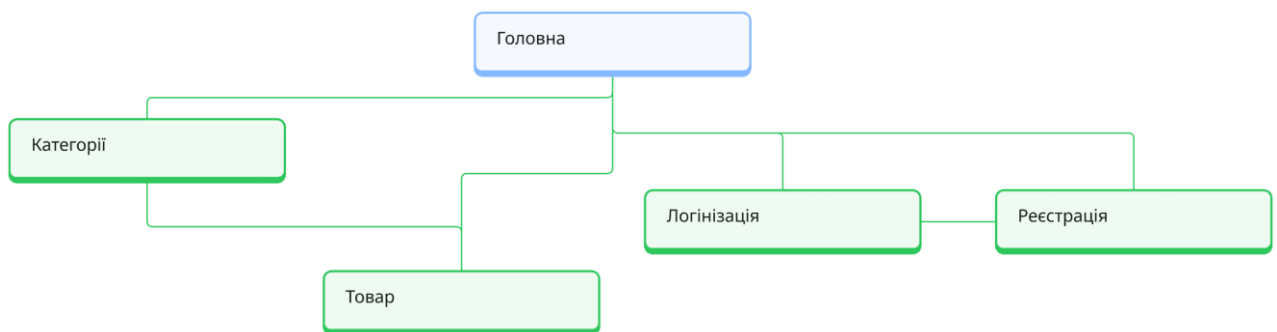


Рисунок 2.7 – Карта сайту

Джерело: розроблено автором

2.3 Прототипування UI/UX дизайну продукту

За допомогою Figma було розроблено прототип desktop-версій головної сторінки (рис 2.8), сторінки категорії (рис. 2.9) та сторінки товару (рис. 2.10).

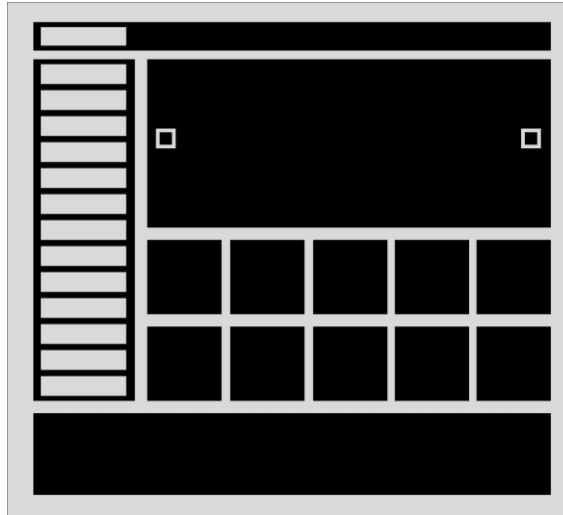


Рисунок 2.8 – Прототип головної сторінки

Джерело: розроблено автором

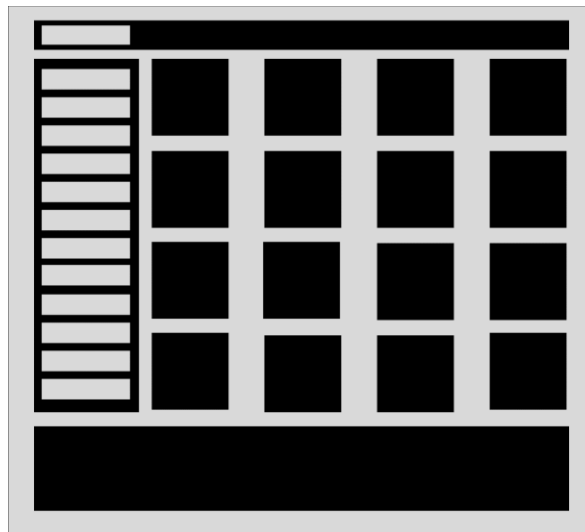


Рисунок 2.9 – Прототип сторінки категорії

Джерело: розроблено автором

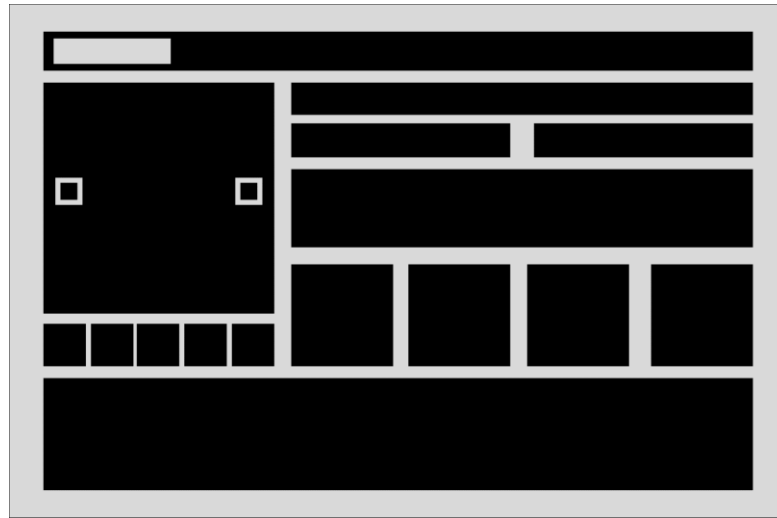


Рисунок 2.10 – Прототип сторінки товару

Джерело: розроблено автором

Висновок до розділу 2

У цьому розділі було проведено моделювання поведінки продукту та UI/UX дизайну інтернет-магазину. Визначення портрета цільової аудиторії та аналіз потреб користувачів дозволили створити зручний та ефективний інтерфейс, що сприятиме підвищенню зацікавленості відвідувачів і зростанню конверсії.

Було розглянуто основні етапи моделювання UI/UX дизайну, включаючи створення профілю потенційного клієнта, мапи емпатії, мапи подорожі клієнта та карти сайту. Використані методи та інструменти дозволили сформувати структурований підхід до розробки інтерфейсу, орієнтованого на зручність і потреби користувачів.

Також було розроблено прототипи основних сторінок вебсайту – головної, категорійної та сторінки товару. Це дозволило сформувати структуру майбутнього інтерфейсу, визначити логіку навігації та підготувати основу для подальшої реалізації функціоналу.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ ІНТЕРНЕТ-МАГАЗИНУ «TAGSYNC»

3.1 Реалізація та конструювання програмного продукту

React [11] – це JavaScript-бібліотека, що розроблена компанією Meta (раніше Facebook). React дозволяє ефективно будувати сучасні, інтерактивні вебзастосунки з використанням компонентного підходу.

React використовується для:

- створення односторінкових застосунків (SPA);
- динамічних веб-інтерфейсів, де змінюються лише частини сторінки;
- підтримки складної логіки у Frontend;
- розробки повторно використовуваних UI-компонентів.

Переваги React над аналогами (Angular, Vue):

- висока продуктивність завдяки Virtual DOM;
- компонентний підхід, що спрощує розробку та підтримку великих проєктів;
- гнучкість і свобода вибору сторонніх бібліотек для маршрутизації, стану, запитів тощо.;
- велика екосистема та спільнота, що дозволяє швидко знаходити рішення і використовувати готові компоненти;
- легка інтеграція з іншими технологіями (наприклад, Next.js для SSR або SEO-оптимізації);
- JSX-синтаксис, який дозволяє писати логіку та розмітку в одному місці, покращуючи читабельність і підтримку коду.

Next.js [12] – це фреймворк для React, розроблений компанією Vercel. Next.js дозволяє створювати повноцінні, швидкі та масштабовані вебзастосунки.

Next.js використовується для:

- статичної генерації сторінок (SSG), де генерація HTML відбувається на етапі build, для швидкого завантаження;
- серверного рендерингу (SSR), в якому рендеринг на сервері відбувається при кожному запиті, для динамічного контенту;
- інкрементної статичної регенерації, що є гібридом SSG і SSR і дозволяє оновлювати вже згенеровані статичні сторінки на фоні, без потреби повного перезапуску сайту.
- створення зручної системи маршрутизації без додаткових налаштувань;
- SEO-оптимізації та мета-тегів для кожної сторінки товару або категорії;
- вбудованої обробки динамічних маршрутів, що дозволяють створювати складні URL-структури без зайвого коду;
- інтеграції з API (REST і GraphQL).

Переваги Next.js:

- підтримка SSR та SSG з коробки, що покращує SEO й час завантаження;
- вбудована маршрутизація на основі структури файлів – без потреби в окремому налаштуванні;
- готова підтримка для імпорту зображень, CSS, шрифтів та інших ресурсів;
- оптимізації продуктивності для зменшення ваги сторінок;
- інтеграція з CDN та edge-функціями через Vercel або інші платформи;
- гнучкість: можна поєднувати SSR, SSG та клієнтський рендеринг у межах одного проєкту;
- хороша документація та активна спільнота.

TypeScript [13] – це надбудова над JavaScript, яка додає статичну

типізацію. Вона допомагає уникати багатьох помилок ще на етапі розробки та покращує масштабованість проєкту.

TypeScript використовується для:

- написання надійного та передбачуваного коду;
- зменшення кількості runtime-помилки;
- покращення автодоповнення та навігації в редакторі;
- роботи над великим проєктом у команді.

Переваги TypeScript над JavaScript:

- статична типізація, що дозволяє «ловити» помилки до запуску;
- кращий досвід розробника завдяки автодоповненню, рефакторингу, підказкам;
- підтримка сучасних функцій JavaScript, навіть у старіших браузерах (через компіляцію);
- краща структуризація коду завдяки інтерфейсам і типам;
- масштабованість: проєкт легше підтримувати й розширювати з часом.

SCSS (Sass) [14] – це препроцесор CSS, який додає змінні, вкладеність, міхін-функції, умови та інші зручні можливості, яких не вистачає в класичному CSS.

SCSS використовується для:

- створення стилів з використанням змінних та логіки;
- кращої організації коду завдяки вкладеності;
- побудови масштабованої системи стилів (наприклад, для великого магазину).

Переваги SCSS над CSS і LESS:

- вкладеність селекторів, яка робить код більш читабельним;
- змінні, міхін-функції, розширення, що зменшують дублювання коду;
- поширена підтримка у фреймворках та середовищах розробки;

- більше можливостей, ніж у CSS, і простіший синтаксис, ніж у Less;
- гнучка структура файлів (можна організувати код у частини та імпортувати їх).

Axios [15] – це популярна бібліотека для здійснення HTTP-запитів у JavaScript/TypeScript-застосунках.

Axios використовується для:

- запитів до API (отримання товарів, оформлення замовлень тощо);
- надсилання даних форм (наприклад, кошик або контактна форма);
- обробки помилок і відповідей сервера.

Переваги Axios над fetch API:

- простіший синтаксис та обробка запитів/відповідей;
- підтримка перехоплювачів (interceptors) для втручання в процес надсилання запиту або отримання відповіді **до того, як** запит буде надіслано на сервер або відповідь потрапить у викликаючий код;
- автоматична трансформація JSON;
- скасування запитів і конфігурацій за замовчуванням;
- краще зчитується і масштабується, особливо для великих проєктів.

TanStack Query [16] – це бібліотека для управління серверним станом у React-застосунках. Вона спрощує отримання, кешування, синхронізацію та оновлення даних з API.

TanStack Query використовується для:

- отримання даних з API;
- кешування відповідей, щоб уникати зайвих запитів;
- автоматичного повторного запиту при фокусі або зміні мережі;
- управління статусами запитів (loading, error, success);
- фонових оновлень даних.

Переваги TanStack Query над альтернативами (SWR, useEffect + useState):

- автоматичне кешування, що зменшує навантаження на API;

- зручне управління статусами та помилками, без зайвого коду;
- devtools для debug запитів у реальному часі;
- підтримка пагінації, нескінченне прокручування, мутацій із оптимістичним оновленням;
- менше коду та краща структуризація, ніж при використанні `useEffect/useState`.

React Hook Form [17] – це легка бібліотека для керування формами у React, яка базується на хуках. Вона забезпечує високопродуктивну валідацію та зручне управління формовими даними.

React Hook Form використовується для:

- створення форм будь-якої складності (реєстрація, замовлення, фільтри тощо);
- валідації полів у режимі реального часу;
- інтеграції з бібліотеками валідації;
- обробки відправлення форми та виведення помилок.

Переваги React Hook Form над альтернативами (Formik, useState):

- висока продуктивність: мінімальна кількість повторних рендерів;
- часто вимагає меншої кількості коду, ніж Formik або useState;
- проста інтеграція з бібліотеками валідації;
- зручна робота з елементами форми;
- гнучка структура форм без втрати типізації в TypeScript.

ESLint [18] – інструмент для аналізу коду на JavaScript/TypeScript, який допомагає дотримуватись стилю коду та знаходити потенційні помилки ще до запуску.

ESLint використовується для:

- виявлення помилок і антипатернів у коді;
- забезпечення єдиного стилю коду в команді;
- підтримки чистоти та читабельності коду;
- автоматичної перевірки під час commit або CI/CD.

Переваги ESLint:

- допомагає уникати багів ще до запуску застосунку;
- гнучка конфігурація, підтримка плагінів для React, TypeScript тощо;
- інтеграція з редакторами коду для миттєвих виправлень;
- покращує командну роботу через єдиний стиль.

Stylelint [19] – це інструмент для аналізу CSS/SCSS-коду, аналог ESLint для стилів.

Stylelint використовується для:

- перевірки коректності стилів;
- підтримки єдиного стилю написання;
- виявлення помилок або зайвого коду в стилях;

Переваги Stylelint:

- автоматично знаходить помилки в стилях;
- гнучка конфігурація;
- підтримка форматування (наприклад, через Prettier).

Prettier [20] – це інструмент для автоматичного форматування коду, Спирні переваги технологій.

Prettier використовується для:

- форматування коду;
- забезпечення єдиного стилю коду в команді;

Переваги Prettier:

- автоматичне форматування без необхідності в ручних втручань;
- підтримка численних мов та форматів (JavaScript, TypeScript, HTML, CSS, JSON тощо);
- інтеграція з редакторами коду для моментального форматування при збереженні;
- підвищує продуктивність, оскільки команда не витрачає час на форматування вручну.

Feature-Sliced Design (далі – FSD) [21] – архітектурний підхід для організації Frontend-проектів, орієнтований на масштабованість і модульність (рис. 3.).

FSD використовується для:

- організації структури коду за фічами;
- відокремлення логіки, UI та сторінок;
- зменшення зв'язності між модулями;
- кращої підтримки масштабованих проектів.

Переваги FSD:

- зручна структура, завдяки чому розробникам легко орієнтуватись у коді;
- feature не залежать одна від одної, через що менше багів;
- легко додавати нові можливості без впливу на інші частини;
- спрощене тестування та рефакторинг;
- офіційна документація;
- підходить як для маленьких, так і для великих проектів.

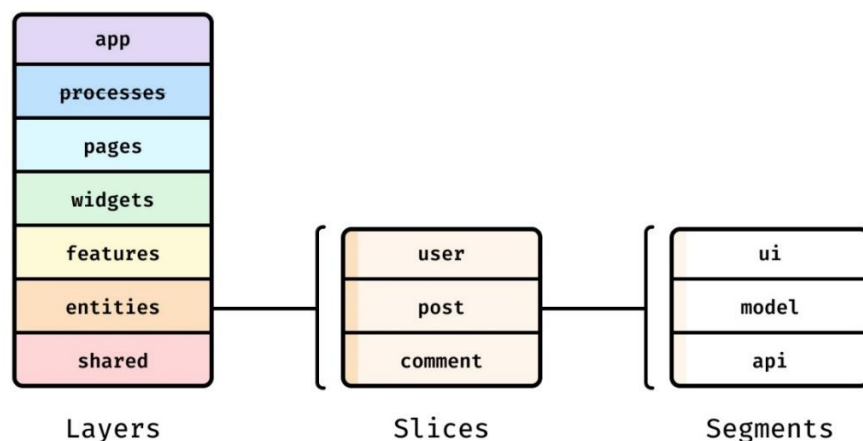


Рисунок 3.1 – Графічне представлення архітектурного підходу FSD

Джерело: FSD [21]

Оскільки під час розробки використовувався Next.js, у якому зарезервовані папки `app` та `pages`, було прийнято рішення перейменувати

відповідні рівні FSD:

- app на application;
- pages на views.

Yarn [22] – це сучасний менеджер пакетів для JavaScript/TypeScript, створений для більшої швидкості, стабільності та зручності роботи з залежностями.

Yarn використовується для:

- встановлення бібліотек і залежностей;
- запуск проєктних скриптів (yarn dev, yarn build тощо);
- керування версіями пакетів.

Переваги Yarn над аналогами (npm і pnpm):

- швидша установка, ніж у npm;
- краще читається і зрозуміліші повідомлення про помилки, ніж у npm;
- ширша підтримка інструментів і плагінів, ніж у pnpm (який новіший).

Postman [23] – це інструмент для тестування API та взаємодії з Backend під час розробки.

Postman використовується для:

- перевірки відповідей API та структури даних;
- роботи з авторизацією, заголовками, змінними середовища;
- імпортування/експортування колекцій API.

Переваги Postman:

- інтуїтивний інтерфейс для роботи з API;
- швидке налагодження запитів без написання коду;
- збереження запитів у колекції для спільного використання в команді.

GitHub [24] – це платформа для хостингу коду, командної розробки та контролю версій з використанням Git.

GitHub використовується для:

- зберігання та резервного копіювання коду;
- спільної роботи над проєктом (pull requests, code review);
- автоматизації через GitHub Actions (CI/CD, тестування, deploy);
- відкриття задач, багів, ведення документації та відстеження

витраченого часу.

Переваги GitHub:

- найпопулярніша платформа для open-source, командних і особистих проєктів;
- зручний веб-інтерфейс для роботи з гілками та змінами;
- потужна інтеграція з іншими сервісами;
- можливість командної роботи над кодом із контролем якості.

Vercel [25] – це платформа для хостингу Frontend-застосунків (особливо Next.js).

Vercel використовується для:

- автоматичного запуску сайту;
- хостингу production-версії проєкту;
- попереднього перегляду кожної гілки (preview deployments);

Переваги Vercel:

- ідеальна інтеграція з Next.js – не потребує додаткових налаштувань;
- автоматичний CI/CD після кожного пушу в GitHub;
- додаткові перевірки, такі як тести аналітики та продуктивності сайту;
- безкоштовний план.

Visual Studio Code (далі – VS Code) [26] – це сучасний редактор коду, який підтримує розширення, інтеграцію з Git і потужні інструменти для розробки.

VS Code використовується для:

- написання коду;
- роботи з Git та гілками безпосередньо з редактора;
- запуску терміналу та локального сервера;
- використання плагінів для мов, фреймворків, застосування набору правил написання коду і його автоматичного форматування.

Переваги VS Code:

- потужна інтеграція з TypeScript, React, ESLint, Stylelint, Prettier тощо;
- можливість налаштувати робоче середовище під себе (темами, гарячі клавіші, snippets тощо);
- величезна кількість розширень для будь-яких технологій;
- вбудований термінал, debugger, а також інтеграція з AI-асистентами;

Програмна частина вебсайту реалізована за допомогою ASP.NET Core. ASP.NET Core є крос-платформним, високо продуктивним фреймворком із відкритим вихідним кодом для створення сучасних хмарних застосунків, підключених до Інтернету. За допомогою ASP.NET Core можна створювати веб-застосунки і сервіси, застосунки IoT і мобільні Backend, використовувати інструменти розробки для Windows, macOS і Linux, розгортати в хмарі або локально, працювати на .NET Core або .NET Framework [27].

В якості бази даних застосовується Supabase. Supabase – це альтернатива Firebase з відкритим вихідним кодом. Це платформа backend-as-a-service, яка дозволяє розробникам створювати, масштабувати та керувати своїми додатками більш ефективно. Вона пропонує багатий набір функцій, включаючи бази даних в режимі реального часу, автентифікацію, зберігання, безсерверні функції та багато іншого [28].

Візуальна частина здебільшого взаємодіє із програмною через API-запити, однак у певних випадках – безпосередньо із Supabase, залежно від конкретного сценарію (рис. 3.2). Особливо це стосується авторизації та аутентифікації користувачів. Наприклад, при авторизації Supabase надає

розширені можливості для Next.js, серед яких – автоматичне керування сесією, збереження токенів доступу в cookies, а також синхронізація стану автентифікації між вкладками браузера.

Окрім авторизації, передбачено й повноцінну функціональність виходу з акаунту. Вбудована функція видаляє токен автентифікації та очищує всі дані, пов'язані з поточною сесією. Це дозволяє гарантувати безпечне завершення роботи користувача в системі. Після виходу з акаунту користувач автоматично перенаправляється на головну сторінку або на сторінку входу, що забезпечує зрозумілий і безперервний користувацький досвід.

Завдяки інтеграції з Supabase, процеси входу та виходу реалізовано з дотриманням найкращих практик безпеки та з урахуванням зручності для кінцевого користувача.

```
1 export const AuthService = {
2   addUserData: async (data: IUserData) => {
3     try {
4       const response: IResponse<IResult> = await axiosInstance.post(
5         "/UserRegistration/addUserData",
6         data
7       );
8
9       return response;
10    } catch (err) {
11      throw err;
12    }
13  },
14
15  login: async (data: ILogin) => {
16    try {
17      const response = await supabase.auth.signInWithPassword(data);
18
19      return response;
20    } catch (err) {
21      throw err;
22    }
23  },
24
25  logout: async () => {
26    try {
27      const response = await supabase.auth.signOut();
28
29      return response;
30    } catch (err) {
31      throw err;
32    }
33  },
34
35  register: async (data: IRegister) => {
36    try {
37      const response: IResponse<IResult> = await axiosInstance.post(
38        "/UserRegistration/register",
39        data
40      );
41
42      return response;
43    } catch (err) {
44      throw err;
45    }
46  },
47 };
48
```

Рисунок 3.2 – Фрагмент коду з сервісу для авторизації

Джерело: розроблено автором

3.2 Тестування програмного продукту

Після завершення основного етапу розробки інтернет-магазину було проведено тестування для перевірки коректності роботи верстки на різних браузерах та операційних системах. Основна мета тестування полягала у виявленні та усуненні можливих проблем з відображенням інтерфейсу користувача, що може впливати на зручність користування сайтом.

Було проведено перевірку верстки сайту в таких популярних браузерах:

- Google Chrome;
- Mozilla Firefox;
- Opera;
- Microsoft Edge;
- Safari.

Для кожного з цих браузерів було перевірено:

- коректне відображення всіх елементів інтерфейсу (слайдери, форми тощо);
- адаптивність верстки при зміні розмірів вікна браузера;
- ідентичні застосування стилів CSS.

Щоб забезпечити універсальність доступу до сайту, було здійснено тестування на таких операційних системах, таких як:

- Windows;
- macOS;
- Android;
- iOS.

У всіх вищевказаних середовищах сайт відображається коректно, верстка адаптується під різні екрани, інтерфейс залишається зручним і читабельним.

Для аналізу продуктивності, швидкості завантаження сторінок та оптимізації сайту було використано онлайн-інструмент PageSpeed Insights від Google.

За результатами тестування основної сторінки інтернет-магазину були отримані гарні показники для desktop- та мобільних версій (Додаток А).

Ці результати свідчать про високу оптимізацію сайту, правильну структуру HTML-коду, ефективне використання кешування, зображень та зовнішніх ресурсів.

3.3 Використання програмного продукту

Розроблений інтернет-магазин має широкий набір функціональних можливостей, які забезпечують зручний користувацький досвід на різних етапах взаємодії з сайтом – від реєстрації до оформлення замовлення.

Реєстрація та авторизація

Користувач може зареєструватися на сайті в два кроки:

1. Введення електронної пошти та паролю (рис. 3.3). Перед переходом до наступного кроку система перевіряє чи не зареєстрований вже користувач з вказаною електронною поштою.

The screenshot shows the registration page of the TagSync website. The page is divided into two main sections. The left section, titled "Створити обліковий запис" (Create account), features a progress indicator at the top, followed by input fields for "Email", "Пароль" (Password), and "Підтвердження паролю" (Confirm password). Below these fields are a "Підтвердити" (Confirm) button and a "Забули пароль?" (Forgot password?) link. The right section, titled "Уже є обліковий запис?" (Already have an account?), contains the text "Увійдіть, щоб отримати доступ до особистого кабінету та всіх можливостей сайту." (Log in to access your personal cabinet and all site features.) and a "Продовжити" (Continue) button. The footer includes the TagSync logo, a brief description of the store, a "Категорії" (Categories) section with links for "Накопичувачі", "Відеокарти", "Процесори", "Материнські плати", "Оперативна пам'ять", "Блок живлення", "Кулери", and "Корпуси", and a copyright notice "Усі права захищені © 2025".

Рисунок 3.3 – Перший крок реєстрації

Джерело: розроблено автором

2. Введення особистих даних: ім'я, прізвище, номер телефону, місто й адреса (рис. 3.4).

Рисунок 3.4 – Другий крок реєстрації

Джерело: розроблено автором

Також на сторінці реєстрації присутнє посилання на сторінку автентифікації.

На сторінці авторизації присутні форма з електронною поштою та паролем, посилання на сторінки реєстрації та скидання паролю (рис. 3.5). У разі втрати паролю користувач може вказати свою електронну пошту, на котрий буде надіслано посилання для встановлення нового паролю.

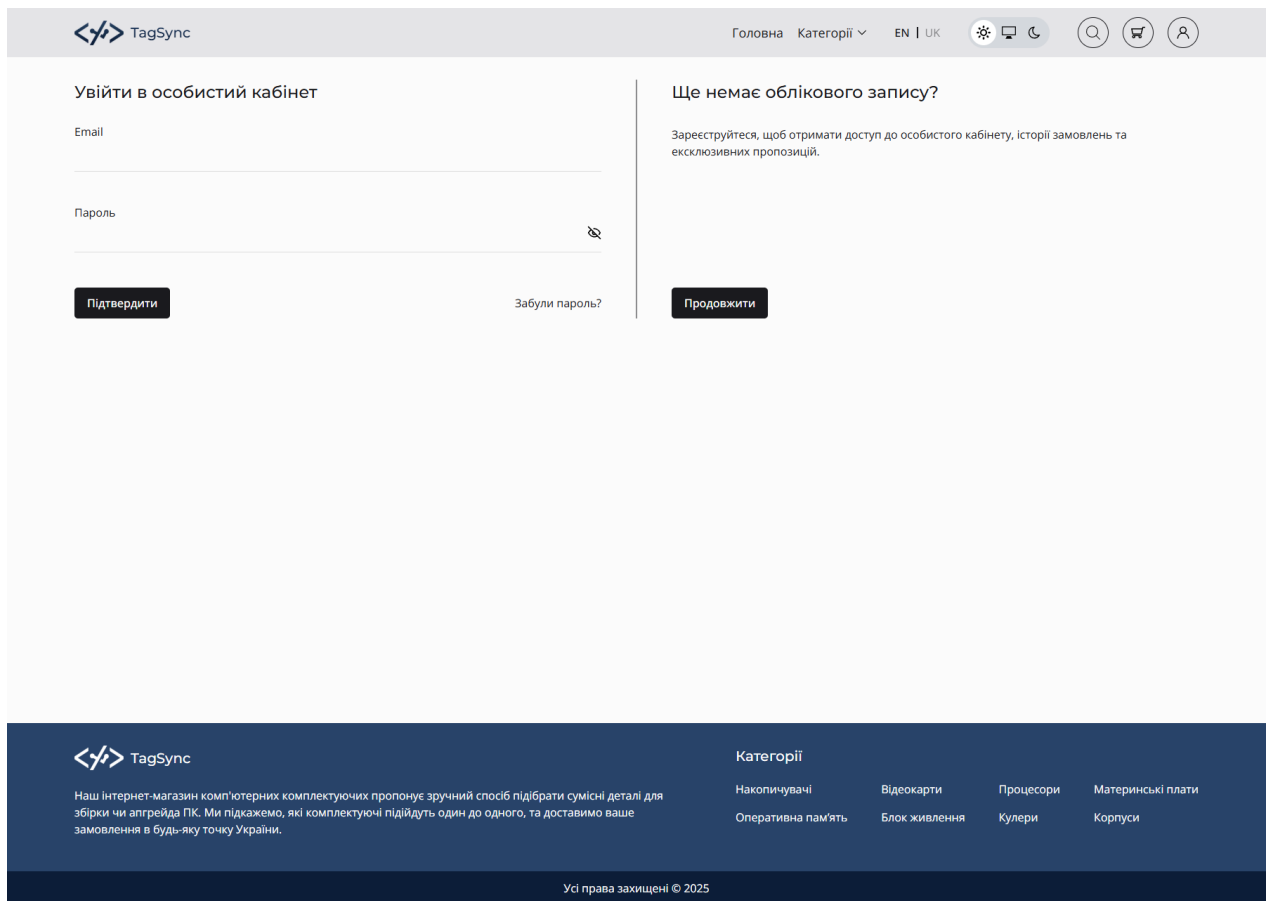


Рисунок 3.5 – Сторінка входу в обліковий запис

Джерело: розроблено автором

Після успішного входу користувач автоматично перенаправляється на останню відвідувану сторінку, якщо вона була та не пов'язана з акаунтом або помилкою, в інших випадках - на головну сторінку сайту.

При виході з акаунту користувач перенаправляється на сторінку входу в обліковий запис.

Головне меню та користувацький інтерфейс

У header доступні наступні можливості:

- перехід на головну сторінку та категорії товарів;
- перехід на сторінки особистого кабінету (редагування особистих даних, список бажаного, список порівнянь, замовлення), для авторизованих користувачів;

- вихід з акаунту, а якщо користувач не авторизований, то посилання на сторінку авторизації;
- зміна теми (світла, темна, системна);
- зміна мови інтерфейсу (українська й англійська);
- пошук товарів за назвою;
- відкриття модального вікна з кошиком.

У футері користувач бачить:

- загальний опис сайту;
- список усіх категорій товарів.

Головна сторінка

На головній сторінці розміщені:

- вертикальна панель, яка розташована збоку веб-сторінки з усіма категоріями товарів;
- слайдер із основною інформацією;
- слайдери з рекомендованими та нещодавно переглянутими товарами.

Сторінка категорії (рис. 3.6)

Функціонал включає:

- відображення «хлібних крихт» (елемент інтерфейсу сайту, що показує шлях від початкового елемента до поточного рівня. Наприклад: головна – каталог – техніка – телефони.);
- сортування за ціною, рейтингом або переглядами;
- динамічне застосування фільтрів з додатковими кнопками для скидання;
- пагінацію з обмеженням кількості відображених товарів;
- збереження всіх параметрів (фільтри, сортування, сторінка) в URL для можливості поділитися чи оновити сторінку без втрати налаштувань;

- картки товарів із кнопками додавання до списку бажаного, порівняння та кошика.

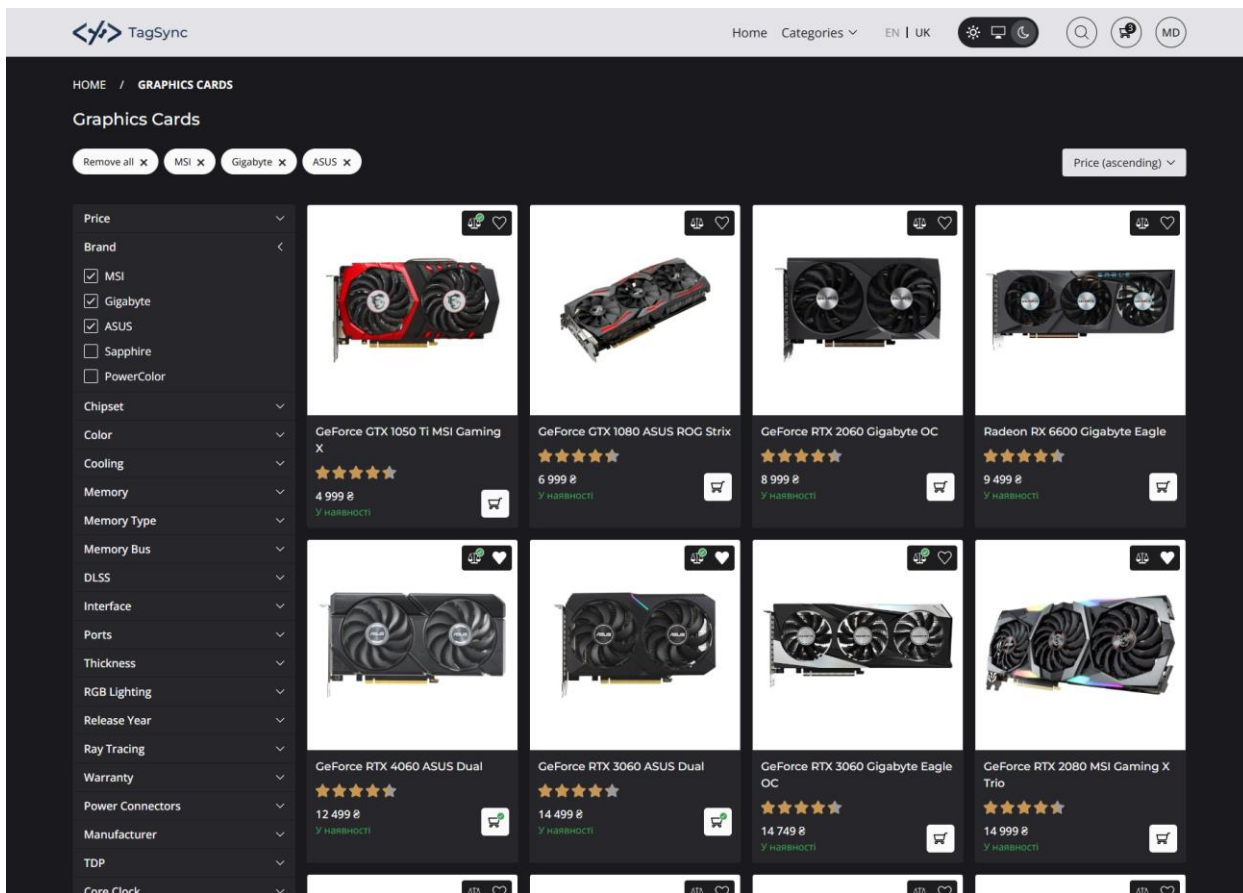


Рисунок 3.6 – Сторінка з відеокартами

Джерело: розроблено автором

Сторінка товару

На сторінці окремого товару користувач бачить:

- слайдери з зображеннями;
- назву товару, рейтинг, кількість написаних відгуків, наявність, ціну;
- кнопки додавання до списку бажаного, порівняння та кошика;
- таблицю характеристик;
- слайдери із рекомендаціями (переглядають також, схожі, сумісні, за ціною);

- відгуки, з можливістю залишити новий (лише для користувачів, які придбали товар).

Кошик

У модальному вікні кошика користувач може:

- переглянути товари, їхню кількість та ціну;
- змінити кількість товарів або видалити їх;
- очистити кошик;
- перейти до оформлення замовлення.

Оформлення замовлення

Функціонал сторінки оформлення включає:

- список товарів у кошику з цінами та кількістю;
- змінити кількості товарів або видалити їх;
- загальну вартість;
- форму з особистими даними, яка автоматично заповнена особистими даними користувача, проте її можна редагувати.

Після успішного оформлення відбувається перенаправлення на сторінку замовлень.

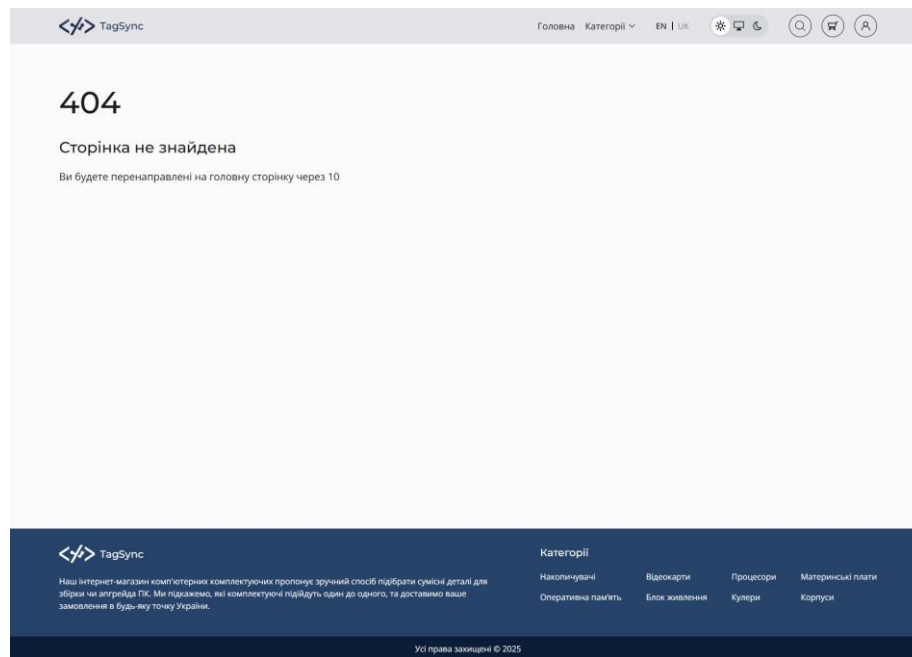
Особистий кабінет

На сторінках особистого кабінету перейти до сторінок:

- списку замовлень;
- списку бажаних товарів
- списку порівнянь;
- редагування особистих даних;
- виходу з акаунту.

Інші можливості

При спробі потрапити на неіснуючу сторінку користувач побачить відповідне повідомлення про помилку та таймер зворотного відліку на перенаправлення на головну (рис. 3.7).



*Рисунок 3.7 – Сторінка 404
Джерело: розроблено автором*

Висновки до розділу 3

Особливу увагу було приділено вибору технологічного стеку. Основою клієнтської частини став Next.js – сучасний React-фреймворк, який забезпечує ефективну обробку та відображення, зручну маршрутизацію, високу продуктивність і хорошу SEO-оптимізацію. Це дало змогу створити швидкий, адаптивний та зручний у користуванні інтерфейс.

Ретельна типізація, структурований підхід до стилізації та ефективна організація роботи з формами сприяли подальшій підтримці проекту й масштабованості застосунку. Крім того, були впроваджені практики, які підвищують якість коду, забезпечують його стабільність і полегшують командну розробку.

Загалом, розроблений функціональний прототип на базі Next.js став міцним фундаментом для подальшої розробки повноцінного вебзастосунку, що відповідає сучасним вимогам до продуктивності, якості та зручності користування.

ВИСНОВКИ

У межах цього проєкту було розроблено візуальну частину вебсайту інтернет-магазину комп'ютерних комплектуючих, основна мета якого – покращити користувацький досвід, зробити навігацію інтуїтивною, а процес пошуку товарів – швидким і ефективним. Робота над проєктом охоплювала повний цикл створення клієнтської частини вебзастосунку – від постановки завдання до реалізації функціонального прототипу.

На етапі аналізу були проведені вивчення предметної області, визначено ключові проблеми існуючих платформ і сформовано вимоги до майбутнього рішення. Було виявлено низку бар'єрів, з якими стикаються користувачі: складність у порівнянні характеристик, обмежені можливості фільтрації, нестача персоналізації та недоступність інтерфейсів для всіх категорій користувачів. Вивчення конкурентів дало змогу окреслити сильні сторони сучасних рішень і врахувати їх під час розробки власного продукту.

На етапі проєктування було змодельовано поведінку цільової аудиторії. Це дозволило сформулювати інтерфейс, максимально орієнтований на потреби відвідувачів. Візуалізація користувацьких сценаріїв і структуризація навігації сприяли створенню логічного й зручного інтерфейсу, який підвищує зацікавленість користувачів.

У процесі реалізації було створено прототипи основних сторінок застосунку: головної, сторінки категорій і сторінки товару. Ключовим інструментом став фреймворк Next.js, який забезпечив високу продуктивність, ефективну обробку та відображення, хорошу SEO-оптимізацію та зручну маршрутизацію. Особливу увагу приділено типізації, стилізації та роботі з формами, що забезпечує стабільність та гнучкість проєкту.

Загалом, реалізований функціональний прототип став надійною основою для подальшого розширення функціоналу інтернет-магазину. Усі рішення приймалися з урахуванням потреб реальних користувачів, що дозволяє проєкту ефективно конкурувати на ринку електронної комерції,

забезпечуючи при цьому високу якість взаємодії та позитивний досвід користувача.

Протягом навчання проводились дослідження, зокрема була проведена апробація дослідження на Міжнародній V Науковій конференції «Сучасний менеджмент організації: витоки, реалії та перспективи розвитку» [29].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Project SEO. Footer. [Електронний ресурс]. URL: <https://project-seo.net/terms-uk/footer-2> (дата звернення: 24.04.2025).
2. Tonyline. Header. [Електронний ресурс]. URL: <https://tonyline.com.ua/glossary/header> (дата звернення: 24.04.2025).
3. Run IT. Як зробити бургер меню CSS? [Електронний ресурс]. URL: <https://www.run-it.com.ua/iak-zrobyty-burher-meniu-css> (дата звернення: 24.04.2025).
4. Webtop. Слайдер на сайті. [Електронний ресурс]. URL: <https://web24.pro/rozrobka-sajtiv-blog/slajder-na-sajti> (дата звернення: 24.04.2025).
5. MDN. Accessibility. [Електронний ресурс]. URL: <https://developer.mozilla.org/en-US/docs/Web/Accessibility> (дата звернення: 24.04.2025).
6. Promodo. Чому SEO не працює: як перевірити рендеринг сторінок на сайті? [Електронний ресурс]. URL: <https://www.promodo.ua/blog/chomu-seo-ne-pracyuie-yak-pereviriti-rendering-storinok-na-sayti#:~:text=Рендеринг> (дата звернення: 24.04.2025).
7. Fondy. Як відкрити інтернет-магазин: покрокова інструкція створення бізнесу з нуля. [Електронний ресурс]. URL: <https://fondy.ua/uk/knowledge/online-store-creation> (дата звернення: 10.03.2025).
8. Інтернет-магазин GameHall [Електронний ресурс]. URL: <https://gamehall.com.ua> (дата звернення: 10.03.2025).
9. Інтернет-магазин CompX [Електронний ресурс]. URL: <https://compx.ua> (дата звернення: 10.03.2025).
10. Інтернет-магазин LuckyLink [Електронний ресурс]. URL: <https://luckylink.kiev.ua> (дата звернення: 10.03.2025).
11. React. Бібліотека зі створення UI для вебу і нативних платформ. [Електронний ресурс]. URL: <https://uk.react.dev> (дата звернення: 07.04.2025).
12. Next.js. Welcome to the Next.js documentation! [Електронний ресурс]. URL: <https://nextjs.org/docs> (дата звернення: 07.04.2025).
13. TypeScript. TypeScript is JavaScript with syntax for types. [Електронний ресурс]. URL: <https://www.typescriptlang.org> (дата звернення: 07.04.2025).

14. Sass. CSS with superpowers. [Електронний ресурс]. URL: <https://sass-lang.com> (дата звернення: 07.04.2025).
15. Axios. HTTP-клієнт для браузера та node.js на основі Promise. [Електронний ресурс]. URL: <https://axios-http.com/uk> (дата звернення: 09.04.2025).
16. Tanstack Query. Powerful asynchronous state management for TS/JS, React, Solid, Vue, Svelte and Angular. [Електронний ресурс]. URL: <https://tanstack.com/query/latest> (дата звернення: 09.04.2025).
17. React Hook Form. Performant, flexible and extensible forms with easy-to-use validation. [Електронний ресурс]. URL: <https://react-hook-form.com> (дата звернення: 09.04.2025).
18. ESLint. Find and fix problems in your JavaScript code. [Електронний ресурс]. URL: <https://eslint.org> (дата звернення: 12.04.2025).
19. Stylelint. A mighty CSS linter that helps you avoid errors and enforce conventions. [Електронний ресурс]. URL: <https://stylelint.io> (дата звернення: 12.04.2025).
20. Prettier. What is Prettier? [Електронний ресурс]. URL: <https://prettier.io/docs> (дата звернення: 12.04.2025).
21. FSD. Documentation. [Електронний ресурс]. URL: <https://feature-sliced.github.io/documentation/docs> (дата звернення: 12.04.2025).
22. Yarn. Safe, stable, reproducible projects. [Електронний ресурс]. URL: <https://yarnpkg.com> (дата звернення: 12.04.2025).
23. Postman. Your Complete API Platform, From Design to Delivery. [Електронний ресурс]. URL: <https://www.postman.com> (дата звернення: 12.04.2025).
24. GitHub. GitHub documentation. [Електронний ресурс]. URL: <https://docs.github.com/en/get-started> (дата звернення: 15.04.2025).
25. Vercel. Vercel Documentation. [Електронний ресурс]. URL: <https://vercel.com/docs> (дата звернення: 15.04.2025).
26. Visual Studio Code. Your code editor. [Електронний ресурс]. URL: <https://code.visualstudio.com> (дата звернення: 15.04.2025).
27. Brights. Розробка ASP.NET. [Електронний ресурс]. URL: <https://brights.io/ua/technology/asp-net> (дата звернення: 23.04.2025).
28. Stape. Supabase Writer tag для server Google Tag Manager. [Електронний ресурс]. URL: <https://stape.io/ua/blog/supabase-writer-tag-for-server-google-tag-manager> (дата звернення: 23.04.2025).
29. Феценко М.І., Давиденко М.О., Косенко С.М. АНАЛІЗ «КРАНЧУ» ЯК ЯВИЩА В ІНДУСТРІЇ ВІДЕОІГОР // «КРОК» Конференції, Сучасний

менеджмент організації: витоки, реалії та перспективи розвитку 2025 – URL: <https://conf.krok.edu.ua/ММО/ММО-2025/paper/view/2856> (дата звернення: 23.04.2025).

30. Pagespeed Insights [Електронний ресурс]. URL: <https://pagespeed.web.dev> (дата звернення: 23.04.2025).

ДОДАТОК А

ПОКАЗНИКИ ПРОДУКТИВНОСТІ

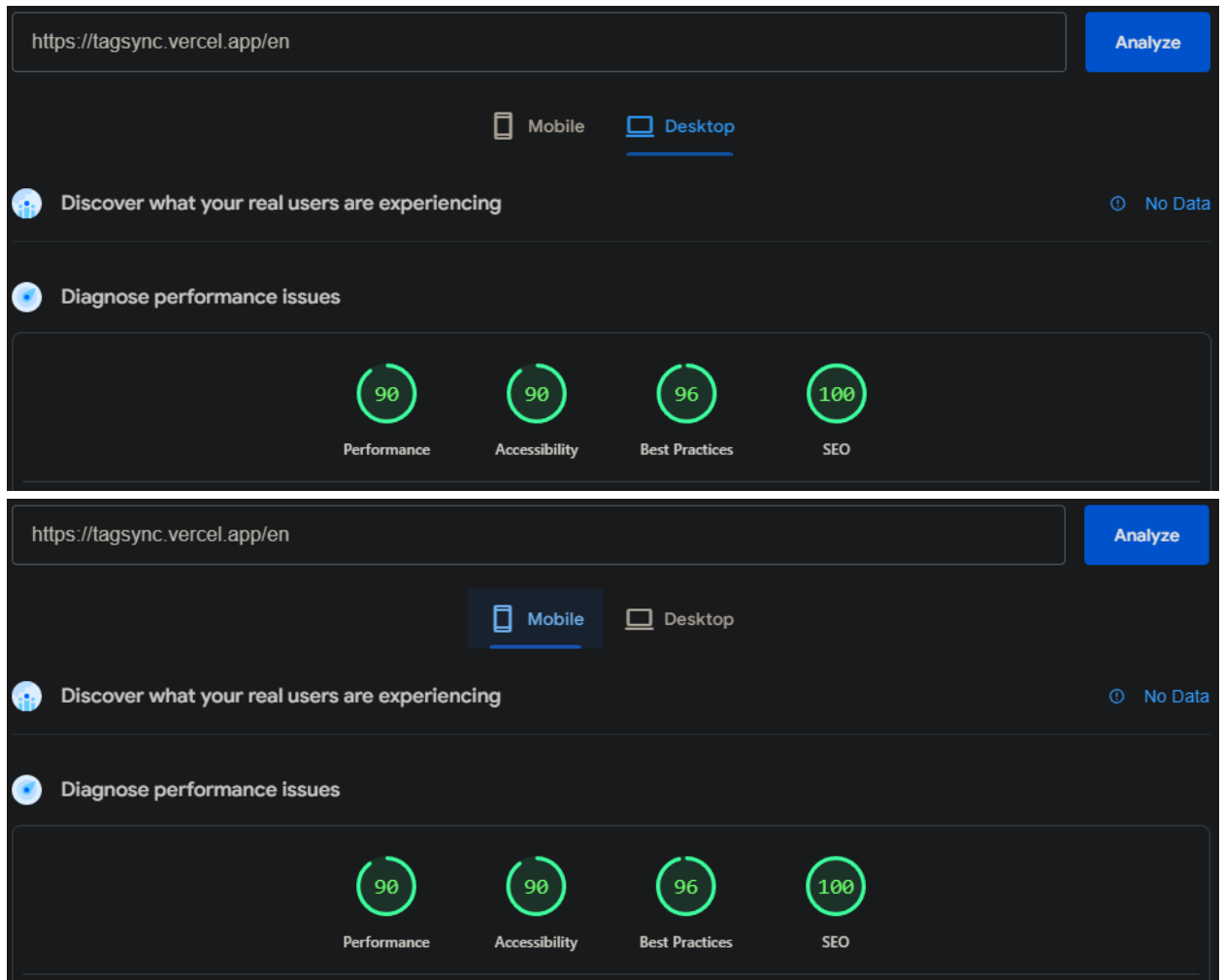


Рисунок А.1 – Зверху показники продуктивності desktop-версії, а знизу – показники мобільної версії.

Джерело: Pagespeed Insights [30]