ЛЬВІВСЬКИЙ ТОРГОВЕЛЬНО-ЕКОНОМІ[ЧНИЙ УНІВЕРСИТЕТ

Факультет міжнародних економічних відносин та інформаційних технологій

Кафедра комп'ютерних наук, прикладної та вищої математики

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Розробка та реалізація інтерфейсу користувача для інтернет-магазину

Виконав: здобувач вищої освіти першого (бакалаврського) рівня
ОПП "Комп'ютерні науки"
спеціальності 122 "Комп'ютерні науки"
_____ Цяпало Назарій Іванович
(підпис)

Науковий керівник
_____ доктор філософії, доцент, Зацерковний Р.Г.
(підпис)

Львів – 2025

ЛЬВІВСЬКИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
Факультет міжнародних економічних відносин та інформаційних технологій
Кафедра комп'ютерних наук, прикладної та вищої математики
Перший (бакалаврський) рівень вищої освіти
Освітньо-професійна програма "Комп'ютерні науки"
Спеціальність 122 "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Завідувач кафедри комп'ютерних наук,
прикладної та вищої математики

З.Г. Новосад
(підпис) (прізвище та ініціали)

28.10.2025 р.

Завдання
до кваліфікаційної роботи
Цяпало Назарій Іванович
(прізвище, ім'я, по батькові)

1.Тема роботи Розробка та реалізація інтерфейсу користувача для інтернет-магазину

керівник роботи **Зацерковний Ростислав Григорович кандидат наук.**

затверджена наказом по Університету **№1041/03 від "28" жовтня 2024р.**

2.Термін здачі здобувачем вищої освіти роботи 5 червня 2025р.
3. Вхідні дані до роботи література, документація, матеріали та сайти Internet, що пов'язані з розробкою інтерфейсу користувача інтернет магазину.
4. Зміст розрахунково-пояснювальної записки (перелік підлягаючих розробці питань)
1) Дослідження теоретичних основ розробки інтерфейсів
2 Аналіз інтерфейсу інтернет-магазину
3) Реалізація інтерфейсу користувача
5. Перелік графічного матеріалу до 12 слайдів у форматі PowerPoint

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання 28.10.2025 р

Календарний план

№	Найменування етапів виконання КР	Терміни виконання етапів	Примітка
1	Отримання завдання і аналіз інформаційних джерел за темою роботи	до 11.12.2024	виконано
2	Дослідження теоретичних основ розробки інтерфейсів	до 13.01.2025	виконано
3	Аналіз інтерфейсу інтернет-магазину	до 07.02.2025	виконано
4	Побудова логічної структури сайту та сценаріїв взаємодії	до 01.04.2025	виконано
5	Початок реалізації інтерфейсу: верстка макету сайту	до 15.04.2025	виконано
6	Розробка товарних карток та фільтрації	до 22.04.2025	виконано
7	Оформлення кваліфікаційної роботи. Подання роботи на перевірку керівнику	до 16.05.2025	виконано
8	Доопрацювання кваліфікаційної роботи з врахуванням зауважень керівника. Подання роботи для перевірки на рівень запозичень	до 20.05.2025	виконано
9	Подання кваліфікаційної роботи керівнику для отримання відгуку	до 27.05.2025	виконано
10	Представлення кваліфікаційної роботи на зовнішнє рецензування. Подання кваліфікаційної роботи на кафедру комп'ютерних наук	до 5.06.2025	виконано
11	Попередній захист кваліфікаційної роботи	05.06.2025-06.06.2025	виконано
12	Захист кваліфікаційної роботи на засіданні екзаменаційної комісії	згідно графіку роботи ЕК	

Здобувач вищої освіти _____ Цяпало Н.І.

Керівник роботи _____ Зацерковний Р.Г.

Реферат

Кваліфікаційна робота присвячена створенню інтерфейсу інтернет-магазину для продажу комплектуючих ПК. Робота містить вступ, три розділи, висновки, список джерел та додатки. Основний зміст викладено на 60 сторінках, містить 12 рисунків і 3 таблиці.

Об’єкт дослідження — вебінтерфейси інтернет-магазинів.

Предмет дослідження — процес створення інтерфейсу користувача.

Мета роботи — реалізація функціонального та зручного UI для онлайн-магазину.

У першому розділі розглянуто основи UI/UX і сучасні технології веброзробки.

У другому — описано проєктування інтерфейсу, визначено структуру і функціонал.

У третьому — реалізовано інтерфейс із використанням HTML, CSS, JS і JSON.

Ключові слова: інтерфейс, інтернет-магазин, UX, HTML, CSS, JavaScript, фільтрація, картка товару.

Abstract

The qualification work is dedicated to creating an interface for an online store for selling PC components. The work contains an introduction, three chapters, conclusions, a list of sources and appendices. The main content is presented on 60 pages, contains 12 figures and 3 tables.

The object of research is web interfaces of online stores.

The subject of research is the process of creating a user interface.

The goal of the work is to implement a functional and convenient UI for an online store.

The first section covers the basics of UI/UX and modern web development technologies.

The second describes the interface design, defines the structure and functionality.

In the third, an interface is implemented using HTML, CSS, JS, and JSON.

ЗМІСТ

Вступ 6
1 Дослідження теоретичних основ розробки інтерфейсів 10
1.1 Поняття інтерфейсу користувача та його значення: 10
1.2. Огляд сучасних технологій для реалізації інтерфейсів 13
1.3 Особливості інтерфейсів для інтернет-магазинів 17
2 Аналіз інтерфейсу інтернет-магазину 22
2.1. Формування вимог до інтерфейсу 22
2.2. Визначення основного функціоналу магазину 25
2.3. Розробка структури сторінок сайту 29
2.4. Проєктування інтерфейсу користувача 33
2.5 Вибір технології реалізації інтерфейсу 37
3 Реалізація інтерфейсу користувача 40
3.1. Розробка головної сторінки з каталогом товарів 40
3.2. Реалізація карток товарів 43
3.3. Фільтрація товарів за категоріями, брендами та ціною 46
3.4. Адаптивність та кросбраузерність інтерфейсу 48
3.5. Впровадження функціональних елементів 52
Висновки 56
Список використаних джерел 60

ВСТУП

Актуальність теми:

У сучасному цифровому світі електронна комерція стрімко розвивається та займає дедалі більшу частку ринку продажів у всьому світі. Зростання популярності інтернет-магазинів обумовлене зручністю їх використання, широким вибором товарів, швидкістю обробки замовлень та можливістю здійснювати покупки незалежно від місця перебування покупця.

Із розвитком інтернет-технологій змінюються і вимоги до функціоналу онлайн-магазинів: користувачі очікують від веб-сайтів не лише швидкої роботи, а й інтуїтивно зрозумілого, зручного та привабливого інтерфейсу. Саме інтерфейс користувача є першим елементом взаємодії відвідувача з веб-ресурсом, і саме він формує перше враження про магазин. Якщо інтерфейс незручний, перевантажений або непродуманий, користувач, найімовірніше, покине сайт, не завершивши покупку, і звернеться до конкурентів.

У контексті продажу комп’ютерних комплектуючих вимоги до інтерфейсу ще вищі: покупець часто хоче швидко порівняти кілька моделей, відфільтрувати за технічними параметрами, переглянути зображення, дізнатися характеристики, наявність і ціну — і все це бажано в кілька кліків. Таким чином, створення ефективного, адаптивного та сучасного інтерфейсу інтернет-магазину є надзвичайно актуальним завданням як для комерційного сектору, так і для індивідуальних розробників, які прагнуть створити конкурентоспроможний продукт.

У даній дипломній роботі розглянуто повний цикл створення інтерфейсу користувача для інтернет-магазину з нуля: від теоретичних основ і проектування до безпосередньої реалізації та тестування.

Мета і завдання роботи:

Основною метою даної дипломної роботи є створення користувацького інтерфейсу для функціонального інтернет-магазину, який забезпечуватиме зручну взаємодію з каталогом товарів, їх переглядом, сортуванням та

фільтрацією. У фокусі — продаж комплектуючих до персональних комп’ютерів, що дозволяє враховувати специфіку таких товарів при розробці інтерфейсних рішень. Для досягнення поставленої мети було сформульовано такі завдання:

- Провести аналіз поняття інтерфейсу користувача, зокрема в контексті інтернет-магазинів;
- Ознайомитися з сучасними принципами UI/UX-дизайну та технологіями реалізації вебінтерфейсів;
- Розглянути приклади оформлення та структури популярних інтернет-магазинів;
- Визначити вимоги до функціоналу майбутнього інтерфейсу;
- Спроектувати інтерфейс за допомогою інструментів прототипування;
- Реалізувати основні елементи інтерфейсу (головна сторінка, картки товарів, фільтри, пошук тощо);
- Забезпечити адаптивність сайту для перегляду з різних пристроїв;
- Перевірити коректність роботи інтерфейсу та усунути можливі помилки.

Об’єкт і предмет дослідження:

Об’єктом дослідження в дипломній роботі є інтерфейси веб-застосунків, зокрема веб-інтерфейси інтернет-магазинів, які дозволяють здійснювати онлайн-торгівлю товарами.

Предмет дослідження — процес створення інтерфейсу користувача: від аналізу до практичного втілення, із врахуванням принципів зручності (usability), логіки структури, візуальної привабливості та технічної реалізації. Особлива увага приділяється інтерфейсам саме інтернет-магазинів, що працюють із комп’ютерною технікою, оскільки ці сайти мають власні функціональні особливості (фільтрація за технічними параметрами, багаторівневий каталог, підтримка порівняння товарів тощо).

Методи дослідження:

Під час виконання дипломної роботи застосовувалися як теоретичні, так і практичні методи дослідження. Зокрема:

- Метод аналізу літературних джерел та інтернет-ресурсів з тем UI/UX-дизайну, веброзробки та реалізації інтерфейсів;
- Метод порівняльного аналізу інтерфейсів популярних інтернет-магазинів (розгляд прикладів таких сайтів, як Rozetka, Comfy, Amazon, та інших);
- Метод моделювання — для створення прототипу інтерфейсу в середовищі Figma;
- Метод розробки — застосування HTML5, CSS3 та JavaScript для створення та стилізації сторінок сайту;
- Метод тестування — перевірка адаптивності, швидкодії та зручності інтерфейсу на різних пристроях і в різних браузерах.

Структура роботи:

Дипломна робота побудована логічно та послідовно й охоплює ключові етапи створення користувацького інтерфейсу. Структурно вона складається з:

- Вступу, у якому обґрунтовується актуальність теми, визначається мета, завдання, об'єкт і предмет дослідження, описуються методи, що використовувались у роботі, та надається загальна структура;
- Першого розділу, де розглядаються теоретичні основи: поняття інтерфейсу, вимоги до сучасних UI/UX-дизайнів, особливості інтерфейсів для онлайн-магазинів;
- Другого розділу, присвяченого проектуванню інтерфейсу: визначенню функціоналу, формуванню структури сторінок, створенню прототипів;
- Третього розділу, в якому описано реалізацію інтерфейсу: верстку головної сторінки, створення карток товарів, реалізацію фільтрації, адаптивності та інших елементів;

- Висновків, у яких підбиваються підсумки роботи, описуються досягнення поставленої мети та перспективи подальшого розвитку проєкту;
- Списку використаних джерел та додатків;

1 ДОСЛІДЖЕННЯ ТЕОРЕТИЧНИХ ОСНОВ РОЗРОБКИ ІНТЕРФЕЙСІВ

1.1 Поняття інтерфейсу користувача та його значення:

Інтерфейс користувача (англ. User Interface, UI) — це сукупність засобів, за допомогою яких користувач взаємодіє з інформаційною системою, програмним забезпеченням або вебресурсом. Інтерфейс виконує роль посередника між технічною складовою системи та її користувачем, полегшуючи виконання завдань і забезпечуючи зрозуміле відображення внутрішніх процесів. У контексті веброзробки під інтерфейсом розуміють насамперед графічні елементи, що забезпечують інтерактивну взаємодію з вебсайтом.

З розвитком цифрових технологій зросла й роль інтерфейсів: у сучасному світі користувачі очікують, що сайт або застосунок буде не лише функціональним, а й інтуїтивно зрозумілим, зручним і привабливим. Саме тому інтерфейс перестав бути лише технічним інструментом — він став важливою частиною загального користувацького досвіду.

Ключове значення UI у веброзробці

Якість інтерфейсу безпосередньо впливає на поведінку користувача на сайті. Якщо сайт має зручну навігацію, логічну структуру, читабельний шрифт, достатній контраст і естетичний вигляд — користувач із більшою ймовірністю залишиться на сайті, перегляне товари, скористається фільтрами, додасть позиції до кошика та завершить покупку. Натомість плутаний або перевантажений інтерфейс призводить до втрати відвідувачів, зменшення конверсії, погіршення SEO-показників і зниження лояльності до бренду.

Розробка хорошого інтерфейсу — це не лише про дизайн, а про розуміння цільової аудиторії, її потреб і очікувань. Інтерфейс повинен відповідати звичним сценаріям користувача, бути послідовним, логічним і передбачуваним.

Зв’язок інтерфейсу (UI) з користувацьким досвідом (UX)

Поняття UI часто розглядається разом із UX (User Experience). Ці два терміни взаємопов’язані, але мають різне значення:

- UI (User Interface) — це зовнішній вигляд і форма взаємодії. Тобто кнопки, меню, структура сторінок, розташування елементів.
- UX (User Experience) — це загальне враження користувача від використання системи. Включає швидкість, емоційність, задоволення, легкість виконання завдань.

Інакше кажучи, UI — це засіб, а UX — результат. Неможливо досягти хорошого UX без якісного UI. Саме тому веброзробники й дизайнери велику увагу приділяють вивченню поведінки користувача, тестуванню інтерфейсів і оптимізації під реальні сценарії використання.

Еволюція інтерфейсів у веброзробці

Розвиток інтерфейсів у веброзробці проходив кілька етапів:

- 1990–2000 рр. — перші HTML-сайти, практично без стилізації, із таблицями, простими формами, обмеженою інтерактивністю.
- 2000–2010 рр. — поява CSS, JavaScript і динамічних елементів, розвиток DOM, Ajax, застосування Flash, зростання важливості зовнішнього вигляду.
- 2010–2020 рр. — розквіт адаптивного дизайну, активне використання фреймворків (Bootstrap, jQuery, Vue, React), підвищення вимог до UX, швидкості завантаження, доступності.
- З 2020-х — мінімалізм, mobile-first-дизайн, акцент на продуктивність, спрощення інтерфейсів, інтеграція штучного інтелекту (наприклад, персоналізовані блоки товарів), використання анімацій і мікровзаємодій.

Сьогодні інтерфейс повинен не просто бути «гарним». Він має бути функціональним, швидким, інтуїтивним, доступним для людей з різними потребами (наприклад, користувачів з порушенням зору), адаптивним до різних розмірів екранів.

Типи інтерфейсів

Існує кілька видів інтерфейсів:

- Графічний інтерфейс користувача (GUI) — найпоширеніший, орієнтований на візуальну взаємодію.
- Командні інтерфейси (CLI) — використовуються професіоналами, працюють на основі введення команд.
- Вебінтерфейси — спеціальний підтип графічного інтерфейсу, що функціонує в браузерах.
- Мобільні інтерфейси — адаптовані для екранів смартфонів і планшетів.
- Голосові інтерфейси (VUI) — взаємодія голосом (наприклад, Siri, Google Assistant).
- Інтерфейси доповненої реальності (AR) — новітній тип, що активно розвивається.

У межах даної кваліфікаційної роботи нас цікавить саме вебінтерфейс, реалізований засобами HTML, CSS, JavaScript та JSON. Такий інтерфейс є універсальним, працює у всіх браузерах і легко адаптується до потреб користувача.

Значення інтерфейсу для інтернет-магазинів

Особливу роль інтерфейс відіграє в інтернет-магазинах. Тут він виконує функції не лише представлення інформації, а й спонукання до покупки, забезпечення довіри, впорядкування великої кількості товарів і зменшення кількості дій до оформлення замовлення.

Дослідження показують, що:

- понад 88% користувачів не повертаються на сайт після поганого досвіду;
- 70% користувачів залишають сайт, якщо їм складно знайти товар;
- 57% користувачів не рекомендують сайт, якщо він погано виглядає на мобільних пристроях.

Власне тому інтерфейс повинен бути:

- логічно структурованим;

- із простою навігацією (категорії, пошук, фільтри);
- із швидкою реакцією на дії (відкриття картки товару, додавання до кошика);
- з оптимізованою формою оформлення замовлення (мінімум кроків);
- адаптивним до різних екранів (від смартфона до широкого монітора).

У інтернет-магазині, створеному засобами HTML, CSS, JavaScript та JSON, я прагнув реалізувати простий, функціональний і легкий інтерфейс, що відповідає базовим вимогам до сучасного UI для e-commerce. Основний акцент було зроблено на категоріях товарів, фільтрації, читабельності карток товарів і легкості навігації. Візуальний стиль — стриманий, із мінімумом графічного шуму.

1.2. Огляд сучасних технологій для реалізації інтерфейсів

Розробка інтерфейсу користувача — це ключовий етап створення будь-якого вебзастосунку, зокрема інтернет-магазину. Від якості реалізації інтерфейсу залежить зручність використання сайту, ефективність взаємодії користувача з функціоналом, а також загальне враження від продукту. Для реалізації інтерфейсу застосовуються сучасні вебтехнології, серед яких центральне місце займають HTML, CSS, JavaScript, а також різноманітні фреймворки та бібліотеки, що базуються на них.

HTML, CSS і JavaScript є класичним «трикутником» фронтенд-розробки, на якому базуються усі інтерфейси вебзастосунків. Кожен з цих інструментів виконує власну функцію: HTML визначає структуру контенту, CSS відповідає за зовнішній вигляд, а JavaScript забезпечує динамічність та інтерактивність. Разом вони формують повноцінний користувацький інтерфейс.

HTML (HyperText Markup Language)

HTML є основою будь-якої вебсторінки. Це мова розмітки, яка використовується для визначення елементів документа: заголовків, абзаців,

посилань, списків, зображень, таблиць, форм тощо. HTML не є мовою програмування в повному сенсі цього слова, адже він не містить логіки, умов або циклів. Його завдання — описати, яка інформація має бути відображена на сторінці та в якій ієрархії.

Структура HTML-документа визначає, як саме контент буде організовано. Наприклад, у структурі інтернет-магазину важливо мати розділення на головне меню, каталог товарів, картки кожного товару, панель фільтрів, форму зворотного зв'язку, футер тощо. Усе це реалізується за допомогою HTML-елементів.

Особливої уваги заслуговує семантична верстка — використання елементів <header>, <main>, <nav>, <section>, <article>, <footer> тощо, що дозволяє зробити сторінку більш зрозумілою для пошукових систем, мобільних браузерів, а також допоміжних засобів (наприклад, скрінрідерів для людей з порушеннями зору). Семантична верстка підвищує доступність ресурсу, що є важливою вимогою сучасного вебу.

CSS (Cascading Style Sheets)

Мова стилів CSS використовується для візуального оформлення HTML-елементів. Якщо HTML визначає структуру, то CSS «оживлює» інтерфейс: додає кольори, шрифти, межі, тіні, фон, а також задає розміри, положення, відступи елементів тощо.

CSS дозволяє створювати як статичні, так і адаптивні макети. Адаптивний дизайн — це підхід, при якому інтерфейс автоматично підлаштовується під розмір екрана пристрою, з якого користувач переглядає сайт. У сучасному світі, де понад половину трафіку припадає на мобільні пристрої, адаптивність є обов'язковою вимогою до інтерфейсу. CSS підтримує медіа-запити (media queries), за допомогою яких можна задати різні стилі для різних розмірів екранів.

Існують кілька систем розташування елементів у CSS: класичне блочне позиціонування, flexbox (гнучкі контейнери) та grid layout (сіткова розмітка). Flexbox зручний для побудови одновимірних макетів (наприклад, горизонтальні

панелі навігації), а grid — для двовимірних структур (таблиці товарів, галереї тощо).

Також CSS підтримує анімації та переходи, що дозволяє робити інтерфейс більш привабливим і плавним у використанні. Наприклад, при наведенні курсора на картку товару може з’являтися тінь або змінюватися фон — усе це реалізується без використання JavaScript.

JavaScript

JavaScript — це мова програмування, яка дозволяє додати до вебінтерфейсу інтерактивність. JavaScript працює безпосередньо в браузері користувача (на стороні клієнта), завдяки чому вебсторінка може миттєво реагувати на дії користувача — без перезавантаження.

Основними можливостями JavaScript у контексті інтерфейсу інтернет-магазину є:

- обробка подій (click, change, input, hover тощо);
- динамічна зміна DOM-структури (додавання, приховування, заміна елементів);
- реалізація фільтрації товарів (зміна контенту при виборі категорії, бренду або ціни);
- валідація форм (перевірка коректності введених даних перед надсиланням);
- робота з локальним сховищем браузера (localStorage, sessionStorage);
- взаємодія з JSON-даними (наприклад, підвантаження карток товарів із зовнішнього JSON-файлу).

JavaScript забезпечує «живість» інтерфейсу. У межах дипломного проєкту, наприклад, при виборі категорії чи бренду товари фільтруються без оновлення всієї сторінки — саме завдяки JavaScript. Це значно покращує користувацький досвід, адже зменшує затримки та забезпечує швидку взаємодію.

Фреймворки та бібліотеки

Попри те, що HTML, CSS і JavaScript є достатніми для створення повноцінного інтерфейсу, на практиці часто використовують додаткові інструменти — фреймворки та бібліотеки. Вони прискорюють розробку, забезпечують кращу масштабованість проєкту та стандартизують код.

- Bootstrap — найпопулярніший CSS-фреймворк, який містить готові стилізовані компоненти (кнопки, форми, таблиці, модальні вікна) та сіткову систему. Ідеально підходить для швидкого створення адаптивного дизайну.
- React — JavaScript-бібліотека від компанії Facebook для створення інтерфейсів на основі компонентів. Використовується для складних, динамічних застосунків.
- Vue.js — легкий фреймворк із простою логікою та двостороннім зв’язком даних. Зручний для інтеграції у невеликі проєкти.
- Angular — потужний фреймворк від Google, який включає всі необхідні інструменти для побудови масштабних вебзастосунків (включаючи маршрутизацію, сервіси, шаблонізацію).

Фреймворки мають як переваги (швидкість, масштабованість, підтримка), так і недоліки (крива навчання, більший обсяг коду, залежність від екосистеми). У межах дипломного проєкту було прийнято свідоме рішення не використовувати фреймворки. Уся логіка реалізована за допомогою чистого HTML, CSS, JavaScript та структури даних у JSON-файлі. Такий підхід дозволив краще зрозуміти фундаментальні принципи фронтенд-розробки, а також зберіг простоту й контроль над усіма аспектами інтерфейсу.

Таким чином, сучасні технології для розробки інтерфейсів охоплюють як базові мови HTML, CSS, JavaScript, так і численні фреймворки. Вибір інструментів залежить від розміру, складності та цілей проєкту. У дипломній роботі реалізовано інтерфейс на чистих технологіях — це дало змогу глибше зануритися в логіку побудови інтерфейсу, гнучко реалізувати фільтрацію товарів, структуру карток і навігацію, а також уникнути зайвої складності. Тим

не менше, ознайомлення з фреймворками дозволяє оцінити сучасні тенденції у веброзробці та потенційні напрямки розвитку проєкту.

1.3 Особливості інтерфейсів для інтернет-магазинів

Розробка сучасного інтерфейсу інтернет-магазину — це складне завдання, яке вимагає врахування великої кількості як технічних, так і психологічних, ергономічних і навіть маркетингових чинників. У час високої конкуренції в електронній комерції інтерфейс користувача стає не лише «вітриною» сайту, а й важливою частиною воронки продажів. У цьому розділі розглянемо ключові вимоги до інтерфейсів сучасних інтернет-магазинів з урахуванням стандартів UX/UI, поведінкових патернів користувачів та технічних особливостей реалізації.

Загальні принципи UX/UI-дизайну

UX (User Experience) та UI (User Interface) — це два взаємопов’язані поняття. UX охоплює загальний досвід користувача при взаємодії із сайтом, а UI — безпосередньо візуальне та інтерактивне оформлення елементів інтерфейсу.

Основні вимоги UX-дизайну полягають у забезпеченні:

- швидкої адаптації користувача до інтерфейсу (інтуїтивність);
- мінімізації кількості кроків до здійснення покупки;
- передбачуваності поведінки елементів;
- відсутності перевантаження контентом;
- чіткої ієрархії сторінок та елементів;
- послідовності інтерфейсу на всіх сторінках.

UI-дизайн, своєю чергою, повинен забезпечувати:

- візуальну привабливість (кольорова палітра, типографіка, сітка);
- контрастність і зручність читання;
- коректне відображення на різних пристроях;
- чіткість усіх інтерактивних елементів (кнопки, поля, меню тощо);
- логічну структуру карток товарів, фільтрів, кошика та форми замовлення.

Навігація та структура сторінок

Однією з ключових вимог до інтерфейсу є зручна навігація. Користувач повинен легко орієнтуватися у структурі сайту, швидко знаходити потрібний розділ, категорію або товар. Сучасні інтерфейси мають забезпечувати так звану «глибину у два кліки» — бажано, щоб користувач міг знайти товар або основну функцію максимум за два натискання миші або дотику до екрана.

Навігаційна система інтернет-магазину зазвичай включає:

- горизонтальне або вертикальне головне меню (категорії, сторінки);
- фільтри товарів (за категорією, брендом, ціною, популярністю);
- пошуковий рядок з автозаповненням або підказками;
- хлібні крихти (breadcrumbs) для відстеження шляху;
- видимий логотип із посиланням на головну сторінку;
- статичний або «липкий» хедер при скролі.

Правильно організована структура каталогу також має значення. Наприклад, у магазині комп’ютерних комплектуючих категорії повинні бути логічно згруповані (процесори, відеокарти, оперативна пам’ять, накопичувачі тощо). Важливо, щоб одна товарна позиція належала тільки до однієї основної категорії, що спрощує навігацію та реалізацію фільтрів.

Картка товару

Один із найважливіших елементів інтерфейсу інтернет-магазину — це картка товару. Саме вона відповідає за перше враження про продукт та є початковою точкою шляху до покупки. Вимоги до картки товару включають:

- наявність якісного зображення (або кількох);
- назву, бренд, короткий опис та ціну;
- кнопку додавання до кошика або обраного;

У межах кваліфікаційної роботи картки товарів реалізовані з використанням JavaScript та JSON-файлу, що містить дані про кожну позицію. Це забезпечує динамічне відображення контенту без оновлення сторінки, що відповідає вимогам сучасних SPA (Single Page Application) інтерфейсів.

Фільтрація та сортування

Ефективний інтерфейс інтернет-магазину неможливий без системи фільтрів. Користувачі очікують мати змогу швидко звузити вибір за допомогою параметрів. Основні вимоги до фільтрації:

- фільтрація без перезавантаження сторінки (JavaScript);
- можливість комбінованого фільтрування (категорія + бренд + ціна);
- зручні елементи керування (чекбокси, слайдери, випадаючі списки);
- видимість активних фільтрів та можливість їх скидання;
- сортування за популярністю, ціною, новинками тощо.

У реалізації кваліфікаційної роботи фільтрація реалізована вручну за допомогою JavaScript, що дозволило забезпечити швидку реакцію інтерфейсу на дії користувача, зберігаючи просту архітектуру.

Адаптивність та мобільна оптимізація

У сучасних умовах більшість користувачів заходять на сайти з мобільних пристроїв. Тому інтерфейс інтернет-магазину має бути повністю адаптивним. Основні вимоги:

- коректне масштабування елементів;
- сенсорна доступність усіх кнопок та меню;
- гнучке меню (бургер-меню);
- відсутність горизонтального скролу;
- оптимізація шрифтів та розмірів для читабельності.

Використання медіа-запитів у CSS, адаптивної сітки (flexbox, grid), а також перевірка макета у різних розмірах екрана — обов’язкові етапи верстки.

Швидкодія та доступність

Час завантаження сайту безпосередньо впливає на користувацький досвід і рівень конверсій. Тому інтерфейс має бути оптимізований:

- зображення — у форматах WebP або стислих JPG/PNG;
- мінімізація CSS та JavaScript-файлів;

- кешування ресурсів;
- відкладене завантаження зображень (lazy loading).зображення — у форматах WebP або стислих JPG/PNG;
- мінімізація CSS та JavaScript-файлів;
- кешування ресурсів;
- відкладене завантаження зображень (lazy loading).

Окрім швидкодії, інтерфейс повинен бути доступним для людей з обмеженими можливостями. Важливими є:

- відповідність WCAG-стандартам;
- контрастність;
- масштабування шрифтів;
- використання ARIA-атрибутів;
- навігація з клавіатури.

Психологічні аспекти дизайну

Успішний інтерфейс також враховує поведінкову психологію користувача. Наприклад:

- правило Фіттса — кнопки повинні бути великими та розташованими ближче до області взаємодії;
- ефект Зейгарник — незавершені дії (незаповнений кошик) спонукають до завершення;
- правило 3–5 секунд — користувач має зрозуміти, що пропонує сайт, протягом перших секунд.

Застосування колірної психології (наприклад, синій — довіра, червоний — акцент), візуальних ієрархій та принципу «від простого до складного» дозволяє покращити враження та зменшити кількість відмов.

Сучасний інтерфейс інтернет-магазину — це результат комплексного підходу, що поєднує технічні, дизайнерські та психологічні аспекти. Він повинен бути адаптивним, швидким, зручним, інтуїтивним та доступним. У межах дипломного проєкту ці вимоги були реалізовані через базові технології HTML,

CSS, JavaScript та власну логіку фільтрації й структурування даних. Відмова від сторонніх фреймворків дала змогу краще опрацювати фундаментальні принципи побудови інтерфейсу, зробивши акцент на базовій якості, простоті та ефективності.

2 АНАЛІЗ ІНТЕРФЕЙСУ ІНТЕРНЕТ-МАГАЗИНУ

2.1. Формування вимог до інтерфейсу

Одним із ключових етапів у розробці будь-якого вебзастосунку є формування вимог до його інтерфейсу користувача. Від якості інтерфейсу безпосередньо залежить ефективність взаємодії користувача з системою, ступінь задоволення очікувань аудиторії та навіть рівень комерційного успіху проєкту. У контексті інтернет-магазину вимоги до інтерфейсу охоплюють низку факторів — зручність, доступність, функціональність, а також естетичність і відповідність загальноприйнятим практикам UX/UI.

Зручність інтерфейсу

Під зручністю інтерфейсу зазвичай мається на увазі здатність користувача легко й швидко орієнтуватися в системі, виконувати цільові дії без необхідності вивчення інструкцій або підказок. Це поняття включає інтуїтивність навігації, послідовність елементів, логічну структуру сторінок.

- Інтуїтивність. Користувач повинен без зайвих дій розуміти, де знаходяться основні функції — каталог, пошук, фільтри, кошик, оформлення замовлення. Наприклад, значок кошика в правому верхньому куті — це загальноприйнятий стандарт, і відступ від нього без вагомої причини може викликати непорозуміння.
- Простота навігації. Кожен розділ повинен бути доступним за 1–2 кліки, а маршрути між сторінками — логічними. Важливо використовувати знайомі користувачам патерни, як-от: верхнє горизонтальне меню, «хлібні крихти», кнопки швидкого повернення до каталогу.
- Структурованість. Усі блоки на сторінці мають бути впорядковані: категорії — зверху або збоку, фільтри — у лівій частині, товари — по центру. Важливо уникати хаотичного розміщення елементів, що може збити користувача з пантелику.

- Послідовність стилю. Всі інтерфейсні елементи мають бути виконані в єдиному візуальному стилі. Це стосується кольорової палітри, типографіки, форм кнопок і блоків.

Також важливими показниками зручності є час виконання базових дій: наприклад, скільки секунд потрібно користувачу, щоб знайти потрібну категорію, додати товар у кошик або переглянути його характеристики. Усі ці аспекти піддаються оцінці в процесі тестування й безпосередньої взаємодії з макетами інтерфейсу.

Доступність інтерфейсу

Під доступністю слід розуміти здатність інтерфейсу бути зрозумілим і зручним для максимально широкого кола користувачів, незалежно від їхніх технічних знань чи досвіду онлайн-покупок. Хоча в рамках цього проєкту не передбачається реалізація адаптивного дизайну, базові принципи доступності мають залишатися в пріоритеті.

- Контрастність кольорів. Текст, фони та графічні елементи повинні мати достатній контраст для легкого зчитування. Наприклад, темно-сірий шрифт на світлому фоні — значно зручніший за світло-сірий на білому.
- Читабельність тексту. Обраний шрифт повинен бути сучасним, але водночас читабельним. Важливо також дотримуватись оптимального розміру шрифту (не менше 14px для основного тексту), достатніх міжрядкових інтервалів і логічної ієрархії заголовків.
- Уникнення перевантаження. Забагато візуальних елементів можуть відволікати користувача від цільової дії. Мінімалістичний, логічно організований інтерфейс значно краще сприяє концентрації уваги.

Загалом доступність — це не лише технічна, а й психологічна характеристика: інтерфейс має викликати в користувача почуття впевненості, а не плутанини чи тривоги.

Функціональність інтерфейсу

Інтерфейс інтернет-магазину є проміжною ланкою між користувачем і функціональними можливостями системи. Від того, наскільки ефективно організовано цю взаємодію, залежить загальне враження про сайт.

До основних функціональних можливостей, які повинен підтримувати інтерфейс, належать:

- Каталог товарів — має бути легко доступний і структурований за категоріями (наприклад, процесори, материнські плати, оперативна пам’ять). Кожна категорія повинна містити короткий опис або назву, а також список товарів у вигляді карток.
- Фільтрація — передбачає можливість вибору товарів за брендом, ціною, характеристиками. Фільтр має оновлювати список товарів динамічно, без перезавантаження сторінки.
- Пошук — текстове поле для швидкого знаходження товару за назвою, брендом чи характеристиками. Бажано передбачити автопідказки або випадаючий список із результатами.
- Картка товару — окрема сторінка або блок, де міститься вся детальна інформація: зображення, опис, технічні характеристики, ціна, кнопка «Додати в кошик».
- Кошик — дозволяє переглядати обрані товари, змінювати кількість, видаляти, переходити до оформлення замовлення.

Очікування користувача

Користувачі інтернет-магазинів звикли до певних шаблонів поведінки сайтів. Відповідно, будь-які відхилення від цих шаблонів можуть викликати нерозуміння або дискомфорт. Наприклад, якщо кнопка «Купити» виглядає як звичайний текст або захована в нестандартному місці, користувач може її не знайти. Тому дуже важливо проєктувати інтерфейс, орієнтуючись на загальні стандарти e-commerce.

Також варто враховувати сучасні тренди: мінімалізм, яскраві акценти, інтерактивні елементи, легка анімація при наведенні. Проте ці елементи не повинні перевантажувати користувача або уповільнювати роботу сайту.

Таблиця 2.1 - Візуальне представлення вимог

Елемент інтерфейсу	**Вимоги**
Головна сторінка	Меню категорій, банери
Каталог	Структура категорій, сортування, фільтрація
Картка товару	Фото, опис, технічні характеристики, ціна, кнопка «Купити»
Кошик	Список товарів, можливість змінити кількість, кнопка «Оформити замовлення»
Пошук	Поле вводу з підказками, інтеграція з фільтрами
Футер	Контакти, політика конфіденційності, навігаційні посилання

Таким чином, сформовані вимоги до інтерфейсу інтернет-магазину стають основою для подальшого етапу — створення прототипу. Вони враховують як технічні, так і поведінкові аспекти взаємодії користувача із сайтом і спрямовані на забезпечення максимальної ефективності, простоти використання та відповідності очікуванням цільової аудиторії.

2.2. Визначення основного функціоналу магазину

Функціональні можливості інтернет-магазину визначають його здатність задовольняти потреби користувачів у повному циклі онлайн-покупки — від ознайомлення з товарами до завершення замовлення. Розуміння й чітке визначення основного функціоналу на початковому етапі проєктування забезпечує основу для логічного інтерфейсу, технічної реалізації та подальшої масштабованості системи. У цьому підрозділі буде розглянуто ключові функції інтернет-магазину, які обов’язково мають бути реалізовані, а також додаткові опції, що покращують користувацький досвід.

Основні групи функціоналу

Функціонал умовно можна поділити на три великі групи:

- Навігаційний функціонал — забезпечує доступ до категорій, підкатегорій, товарів, пошуку тощо.
- Операційний функціонал — дозволяє здійснювати конкретні дії з товарами: перегляд, додавання в кошик, застосування фільтрів.
- Системний функціонал — підтримує обробку замовлень, збереження даних користувача, взаємодію з базами даних, контроль за правильністю введених даних тощо.

У межах цього проєкту особлива увага приділяється навігаційно-операційному функціоналу, адже саме він найбільше пов’язаний із зовнішнім інтерфейсом користувача.

Ключові функції інтернет-магазину

1. Каталог товарів

Каталог — основний інструмент презентації товарів на сайті. Він є візуальним та структурним відображенням усіх наявних позицій, упорядкованих за категоріями. У межах проєкту каталог реалізується у вигляді динамічного списку карток товарів, які створюються на основі даних із JSON-файлу.

Кожна категорія (наприклад: процесори, відеокарти, оперативна пам’ять) має відображати тільки відповідні товари. Категорії формуються окремими кнопками або вкладками, розміщеними в інтерфейсі з лівого боку або у верхній частині сторінки. При виборі категорії користувач бачить лише релевантні картки.

2. Система фільтрації

Фільтри — один з найважливіших інструментів для зручного пошуку. Їх наявність дозволяє швидко скоротити кількість товарів до релевантних. У рамках реалізації передбачаються такі фільтри:

- за брендом (наприклад, Intel, AMD, MSI, Kingston);
- за ціною (за допомогою діапазону або повзунка);
- за категорією (обов’язкова прив’язка до логіки каталогу).

Фільтрація реалізується на фронтенді без перезавантаження сторінки, тобто зміни у фільтрах миттєво оновлюють виведений список товарів. Такий підхід підвищує зручність і відповідає сучасним стандартам UX.

3. Пошук товарів

Пошук є базовим елементом кожного сучасного e-commerce сайту. Користувач вводить назву товару або фрагмент ключових слів, а система автоматично фільтрує відповідні картки. У базовій реалізації заплановано простий текстовий пошук із фокусом на назвах товарів.

Удосконалення пошуку може включати:

- підказки при введенні;
- відображення кількості знайдених результатів;
- пошук за ключовими характеристиками.

Простота реалізації поєднується з великою користю: користувач швидко знаходить необхідне без переходів через десятки сторінок.

4. Картка товару

Картка товару — це компактний блок, у якому представлена візуальна та текстова інформація про товар. Основні елементи картки:

- зображення (або заглушка, якщо зображення відсутнє);
- назва моделі;
- бренд;
- ціна;
- кнопка «Додати до кошика».

Картка має бути компактною, але інформативною. Користувач має можливість ознайомитись з основною інформацією ще до переходу на окрему сторінку (деталізований перегляд можна реалізувати на пізніших етапах). Якщо користувач натискає на картку, може відкриватися модальне вікно або нова сторінка з повним описом (опціонально).

5. Кошик

Кошик — інтерактивний блок, у якому зберігаються обрані товари. У мінімальній реалізації має підтримувати:

- список доданих товарів;
- можливість змінювати кількість або видаляти позицію;
- розрахунок загальної суми;
- кнопку «Оформити замовлення».

Кошик має бути доступним з будь-якої сторінки сайту — як правило, у вигляді іконки в правому верхньому куті. Може відкриватися у вигляді випадаючого вікна або окремої сторінки. На цьому етапі система оформлення замовлення може бути спрощеною, наприклад, без збереження даних покупця, але основна логіка має бути витримана.

6. Сортування товарів

Крім фільтрів, користувачі часто очікують можливість сортування. Базові опції:

- за ціною (спочатку дешевші/дорожчі);
- за популярністю (може бути умовною);
- за назвою (за алфавітом).

Сортування полегшує огляд великої кількості позицій, особливо якщо користувач не має чіткого запиту.

7. Виведення повідомлень і зворотний зв’язок

Система має показувати сповіщення про основні дії: товар додано в кошик, товар уже є в кошику, фільтр не дав результатів тощо. Це забезпечує прозорість взаємодії та підвищує задоволення користувача від роботи з сайтом.

Додаткові функції (на перспективу)

Окрім основного функціоналу, можна запланувати й додаткові функції, які покращують взаємодію користувача:

- Історія переглядів;
- Обрані товари (вибране);
- Порівняння товарів;

- Сторінка з інформацією про компанію;
- Відгуки та рейтинги.

Таблиця 2.2 - Узагальнена таблиця функціоналу

Функції	**Призначення**
Каталог товарів	Виведення списку всіх товарів у структурованому вигляді
Фільтрація товарів	Зменшення кількості товарів до релевантних за вибраними параметрами
Пошук	Швидкий доступ до товару за назвою або фрагментом ключових слів
Картка товару	Відображення основної інформації, кнопка «Додати в кошик»
Кошик	Управління обраними товарами, розрахунок суми
Сортування	Організація товарів за зручними критеріями

Таким чином, визначення основного функціоналу є критичним для побудови логічної структури інтерфейсу. Кожен елемент інтерфейсу має відповідати конкретній функції та забезпечувати її ефективне виконання. Лише гармонійне поєднання зовнішнього вигляду й функціональності забезпечує високу якість взаємодії з користувачем, а отже — успіх інтернет-магазину в цілому.

2.3. Розробка структури сторінок сайту

Розробка структури сайту є критично важливим етапом створення будь-якого вебресурсу, а особливо інтернет-магазину. Від того, наскільки грамотно організована структура, залежить не лише ефективність навігації, але й зручність користування, швидкість прийняття рішень покупцем, а також загальна конкурентоспроможність проєкту. Якщо відвідувач не може інтуїтивно знайти потрібну йому інформацію або товар за кілька кліків — він, швидше за все, покине сайт, не здійснивши жодної взаємодії.

Поняття структури сайту

Під структурою вебсайту розуміється логічна ієрархія всіх його сторінок і функціональних блоків, об’єднаних у зрозумілу навігаційну систему. Для інтернет-магазину структура виконує не лише інформаційну, а й сервісну

функцію: допомагає здійснювати покупки, переглядати товари, формувати кошик, здійснювати пошук тощо.

Ідеальна структура повинна бути збалансованою між простотою (щоб не заплутати користувача) та гнучкістю (щоб легко розширювати проєкт у майбутньому). У структурі повинні бути передбачені всі основні елементи навігації, функціональні блоки, категорії товарів, і способи переходу між ними.

Принципи побудови структури інтернет-магазину:

- мінімальна кількість кліків для досягнення цілі (наприклад, знаходження товару або оформлення замовлення);
- повторюваність і логічність: подібні елементи інтерфейсу повинні працювати однаково;
- наявність інтуїтивно зрозумілих назв сторінок і кнопок;
- уникається дублювання сторінок або надмірне ускладнення навігації.

Типові сторінки інтернет-магазину

У типовому інтернет-магазині передбачаються такі сторінки:

1. Головна сторінка (index.html):

- Виконує роль «вітрини» — демонструє найпопулярніші товари, акції, навігаційне меню, бренди, банери.
- Спрямована на те, щоб затримати користувача на сайті та спрямувати його далі до перегляду категорій чи товарів.

2. Категорії товарів:

- Наприклад: Процесори, Відеокарти, Оперативна пам’ять тощо.
- Містять фільтри за брендом, ціною, характеристиками.
- Забезпечують зручне сортування товарів (за ціною, популярністю, новизною).

3. Картка товару:

- Зображення товару, детальний опис, технічні характеристики.
- Кнопка «Додати в кошик».

- Ціна, можливість вибору кількості.
- Рекомендації або пов’язані товари (опційно).

4. Кошик:

- Перелік вибраних товарів із можливістю редагування.
- Автоматичне оновлення підсумкової суми.
- Перехід до оформлення замовлення.
- Оформлення замовлення:
- Форма для введення контактних даних, способу доставки та оплати.
- Підтвердження замовлення.

5. Пошук:

- Результати пошуку за ключовими словами.
- Можливість додаткової фільтрації.
- Інформаційні сторінки:
- Про нас, Контакти, Політика конфіденційності, Умови повернення (за потреби).

Особливості реалізації структури в проєкті

У моєму випадку структура реалізована як односторінковий інтерфейс (single-page structure) з використанням HTML, CSS, JavaScript та JSON. Це дозволяє уникати зайвих завантажень сторінок та досягти швидкої взаємодії.

Основні особливості реалізації:

- Уся логіка розміщена на одній сторінці index.html.
- При натисканні на товар відкривається модальне вікно з деталями.
- Пошук і фільтрація здійснюються без перезавантаження сторінки (за допомогою JavaScript).
- Дані про товари беруться з файлу products.json, де вказані назва, категорія, бренд, ціна тощо.
- У верхній частині сайту — фільтри за категорією, брендом, ціновим діапазоном.

- Кошик реалізований як окрема динамічна секція або модальне вікно.

Інтерфейс не використовує складні бібліотеки або фреймворки, а базується на нативних технологіях, що робить структуру максимально прозорою для навчання та аналізу.

Навігаційні сценарії

Типові сценарії навігації включають:

- З головної сторінки → до категорії → перегляд товарів → картка товару → додати до кошика.
- Пошук за назвою → список результатів → вибір товару → картка товару → додати до кошика.
- Перегляд кошика → редагування товарів → оформлення замовлення.

Кожен із цих сценаріїв реалізовано без потреби переходу на нову HTML-сторінку, що зменшує час завантаження та покращує UX.

Візуальна схема структури:

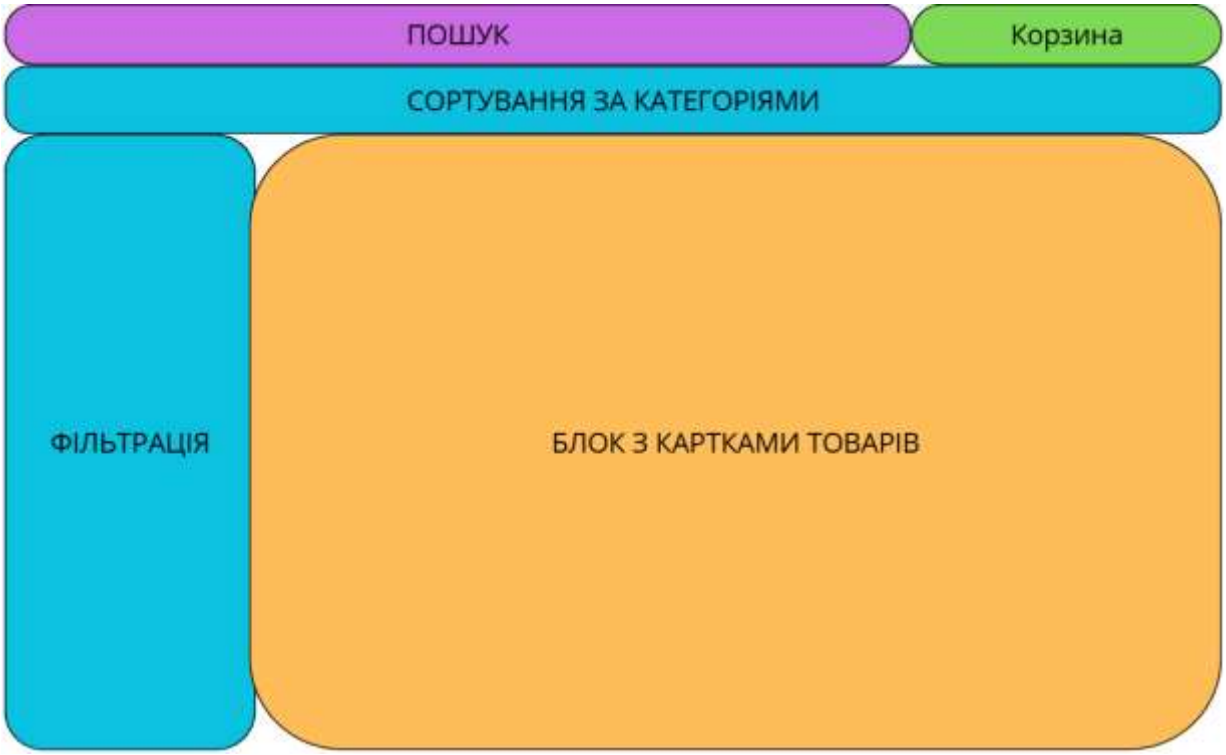

Рисунок 2.1 Переваги розробленої структури

- Мінімалістична, не перевантажена зайвими елементами.
- Швидка робота — за рахунок локального завантаження даних з JSON.
- Легка адаптація під інші категорії чи бренди.
- Зрозуміла структура для користувача — не потрібно вчитись, як працює сайт.

Розробка логічної, зручної та гнучкої структури є одним з основоположних факторів успішності інтернет-магазину. Вона визначає те, наскільки швидко та зручно користувач зможе виконати свою ціль: знайти товар, додати до кошика, оформити замовлення. У моєму випадку структура є спрощеною, проте повністю функціональною, і реалізована за допомогою доступних технологій — HTML, CSS, JS, JSON — що підкреслює її придатність для навчальних цілей.

2.4. Проєктування інтерфейсу користувача

Проєктування інтерфейсу користувача — це ключовий етап розробки інтернет-магазину, що визначає, як саме користувач взаємодіє з функціоналом,

які враження він отримує від сайту, як швидко та безперешкодно може досягти своєї мети — переглянути товари, порівняти, обрати, додати до кошика та оформити замовлення. Відсутність візуального прототипу не означає, що інтерфейс не було спроєктовано — у даному випадку проєктування здійснювалось безпосередньо у коді, з урахуванням логіки поведінки елементів, юзабіліті та принципів UI/UX-дизайну.

Загальний підхід до проєктування

У рамках кваліфікаційної роботи інтерфейс розроблявся за принципом послідовного логічного складання: від загального макету сторінки до реалізації окремих елементів — карток товарів, фільтрів, кнопок, кошика тощо. Всі елементи були зібрані в одному HTML-файлі, що дозволило реалізувати односторінкову логіку з динамічним оновленням вмісту за допомогою JavaScript. Це дозволяє обійтись без прототипів, оскільки візуальний вигляд проектувався безпосередньо «в браузері» — шляхом послідовного налаштування зовнішнього вигляду, відступів, кольорів, поведінки елементів.

Під час проєктування були враховані такі аспекти:

- Візуальна ієрархія: користувач легко відрізняє важливі елементи (заголовки, кнопки, ціни).
- Простота взаємодії: мінімальна кількість кліків для доступу до потрібної дії.
- Інтуїтивність: користувач без підказок розуміє, як користуватись сайтом.
- Узгодженість: однакове оформлення всіх карток товарів, однакова поведінка кнопок.
- Читабельність: використання контрастних кольорів, достатнього розміру шрифтів.

Структура інтерфейсу

Усі елементи інтерфейсу розташовані у вертикальній послідовності на головній сторінці:

1. Верхня частина: Назва магазину або логотип (умовний, текстовий).

2. Основна частина:

- Сітка товарів у вигляді карток.
- Панель фільтрації товарів (категорія, бренд, ціна).
- Кожна картка містить зображення, назву, категорію, бренд, ціну, кнопку "Додати до кошика".

3. Бокова панель або модальне вікно кошика (залежно від реалізації):

- Показує всі додані товари.
- Дозволяє видаляти товари або змінювати кількість.

4. Нижня частина: Контактна інформація (опційно).

Опис основних інтерфейсних елементів

Картки товарів:

- Розмір картки узгоджений для всіх товарів.
- Зображення адаптується до картки, навіть якщо товар не має фото (використовується заглушка).
- Текст вирівняний, є назва, категорія, бренд, ціна.
- Кнопка додавання до кошика має активну візуальну реакцію (зміна кольору при наведенні).

Фільтри:

- Фільтри виконані у вигляді списків чекбоксів або кнопок.
- Користувач може комбінувати фільтри за категорією, брендом, ціною.
- Після зміни фільтра результати оновлюються автоматично без перезавантаження сторінки.
- Змінюється лише вміст товарів — структура сторінки залишається сталою.

Кошик:

- Реалізований як окремий елемент, що відкривається при натисканні на відповідну кнопку.
- Містить перелік доданих товарів, загальну суму, кнопки для видалення товарів або очищення.
- Може містити кнопку переходу до оформлення замовлення (форма не обов’язкова у цьому варіанті реалізації).

Прототипування «в коді»

Замість традиційного графічного прототипу (у Figma, Adobe XD тощо), було реалізовано прототипування безпосередньо в HTML/CSS:

- Всі елементи додавались у DOM структуру поступово, з урахуванням поведінки.
- Стилі коригувались одразу після перегляду в браузері.
- Це дозволило одразу бачити взаємодію між елементами, що часто неможливо на етапі малювання прототипу.
- Такий підхід добре підходить для невеликих навчальних проєктів.

Особливості UX без графічного прототипу

Підхід без використання графічного інструменту має як переваги, так і обмеження:

Переваги:

- Швидкість: не потрібно витрачати час на відображення прототипів у Figma.
- Реалістичність: можна одразу бачити поведінку елементів у реальному середовищі.
- Гнучкість: одразу можна змінити структуру або логіку без оновлення візуального макету.

Недоліки:

- Відсутність етапу попереднього узгодження дизайну.
- Можливі труднощі з візуальним плануванням структури.

- Менше можливостей для колективної роботи (у випадку командної розробки).

Інтерфейс користувача магазину був спроєктований без графічного прототипування, однак із дотриманням основних принципів UI/UX. Такий підхід дозволяє досягти достатнього рівня якості та функціональності для навчального проєкту. Усі елементи інтерфейсу були створені вручну на HTML/CSS з використанням JavaScript для динамічної взаємодії. Це дозволило досягти інтуїтивної, логічної і візуально зрозумілої структури, що є головною метою у проєктуванні інтерфейсу інтернет-магазину.

2.5 Вибір технології реалізації інтерфейсу

У процесі розробки інтерфейсу інтернет-магазину було прийнято рішення використовувати набір технологій, який би одночасно забезпечував простоту реалізації, зручність для тестування та редагування, а також відповідав базовим вимогам до сучасного вебінтерфейсу. З огляду на поставлені завдання, а також формат реалізації проєкту в рамках дипломної роботи, було обрано базовий, але ефективний стек — HTML, CSS, JavaScript та JSON. Ці технології дозволяють створити повноцінний клієнтський вебінтерфейс без використання серверної логіки чи складних фреймворків.

Основна ідея полягала в тому, щоб реалізувати всі функції магазину — від виводу товарів і фільтрації до взаємодії з кошиком — виключно за допомогою клієнтських технологій. Такий підхід дозволяє зосередитися на розробці інтерфейсу та логіки користувацької взаємодії без необхідності налагоджувати сервер або базу даних. Усі дані зберігаються у форматі JSON, а сторінка динамічно змінюється через JavaScript. Таким чином, технології було підібрано не лише відповідно до технічних потреб, а й до навчальних цілей.

HTML (HyperText Markup Language) був використаний для створення структурної основи сайту. За його допомогою створено логічну розмітку: шапка сайту з навігацією, основна область з товарами, бокові панелі з фільтрами, а також модуль для відображення кошика. HTML забезпечив чітку ієрархію

елементів та дозволив підготувати сайт до подальшої стилізації й інтерактивної поведінки. Завдяки семантичній розмітці (зокрема, правильному використанню тегів header, main, section, article, footer тощо), структура сайту залишилась логічною та зрозумілою як для користувачів, так і для майбутніх розробників, які можуть аналізувати або модифікувати код.

CSS (Cascading Style Sheets) відповідав за візуальне оформлення інтерфейсу. За допомогою CSS було реалізовано стилізацію сторінки: кольорову палітру, сіткове компонування товарів, фонові зображення, тіні, шрифти, відступи, а також анімації при наведенні на елементи. Одним із принципів, яких дотримувались під час стилізації, була візуальна простота та однорідність — всі картки товарів виглядають однаково, кнопки мають схожий стиль, а кольори відповідають тематиці магазину техніки. Таким чином, CSS забезпечив створення привабливого, але ненавантаженого інтерфейсу.

JavaScript відіграв ключову роль у реалізації інтерактивної поведінки сайту. Завдяки йому інтерфейс отримав можливість реагувати на дії користувача — зміна фільтрів, пошук, перегляд карток товарів, додавання в кошик, оновлення його вмісту тощо. Оскільки сайт не має бекенду, саме JavaScript обробляє JSON-файл із товарами, динамічно створює HTML-елементи на основі цих даних і вставляє їх у DOM-структуру сторінки. Також з його допомогою реалізовано механізм фільтрації товарів без перезавантаження сторінки, що суттєво покращує зручність користування.

JSON (JavaScript Object Notation) використовується як формат зберігання даних про товари. У спеціальному файлі products.json зберігається масив об’єктів, де кожен об’єкт описує один товар: його назва, категорія, бренд, ціна, опис, зображення тощо. JSON є зручним і зрозумілим форматом, який легко читати, редагувати та обробляти за допомогою JavaScript. Це дозволило організувати гнучку систему управління товарами — для додавання нового товару достатньо лише внести його дані у JSON-файл без потреби змінювати HTML-код вручну.

Вибір саме цього набору технологій був зумовлений не лише їх технічною доцільністю, але й педагогічною логікою. В рамках навчального проєкту було важливо не лише створити працездатний вебсайт, а й глибше зрозуміти принципи його роботи. HTML, CSS, JavaScript і JSON є базовими і незамінними складовими сучасного веброзроблення. Навчившись працювати з ними у взаємозв'язку, можна отримати фундаментальні знання, які стануть у нагоді при переході до більш складних інструментів — таких як React, Vue або серверні мови програмування.

Ще однією перевагою використаного підходу є його універсальність: створений сайт може працювати на будь-якому сучасному браузері без необхідності встановлення додаткових компонентів. Уся логіка працює на клієнтській стороні, а це означає, що сторінку можна просто відкрити у браузері — локально або з хостингу — і вона буде повністю функціональною. Це важливо для тестування, демонстрації проєкту та швидкого внесення змін.

Варто також зазначити, що обрані технології дають можливість у майбутньому легко розширити функціонал. Наприклад, у перспективі можна додати збереження товарів у кошику через локальне сховище браузера, реалізувати сортування, додати модуль авторизації або підключити простий сервер для обробки замовлень. Таким чином, цей стек технологій може виступати основою для подальшого розвитку проєкту.

У підсумку, вибір HTML, CSS, JavaScript і JSON як основних технологій реалізації інтерфейсу є виваженим і цілком обґрунтованим. Вони дозволили реалізувати повноцінний, динамічний і візуально привабливий інтерфейс інтернет-магазину з усіма базовими функціями: фільтрацією, переглядом товарів, взаємодією з кошиком. Цей набір технологій відповідає як технічним вимогам, так і навчальним цілям дипломної роботи, забезпечуючи прозорість, гнучкість і можливість подальшого розвитку.

3 РЕАЛІЗАЦІЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА

3.1. Розробка головної сторінки з каталогом товарів

Головна сторінка є основною точкою взаємодії користувача з інтерфейсом інтернет-магазину. Вона має бути максимально інформативною, зручною для навігації та функціональною. Основне призначення головної сторінки — представити користувачу доступні товари, дати змогу їх перегляду, фільтрації та пошуку без потреби переходу на інші сторінки. Під час реалізації було прийнято рішення побудувати головну сторінку на основі чистих технологій HTML, CSS, JavaScript та з використанням локального JSON-файлу для зберігання інформації про товари.

Структура головної сторінки була реалізована згідно з класичною схемою: заголовок (header), навігаційна панель, область з фільтрами, основний блок з товарами (каталог) та нижній колонтитул (footer). Основна частина екранного простору займається динамічним виводом товарів, які завантажуються через JavaScript із JSON-структури.

HTML-структура

Розмітка головної сторінки виконує каркасну роль і формує логічні блоки, які згодом заповнюються стилями та функціональністю. Основні елементи були оголошені як контейнери для подальшого динамічного наповнення.

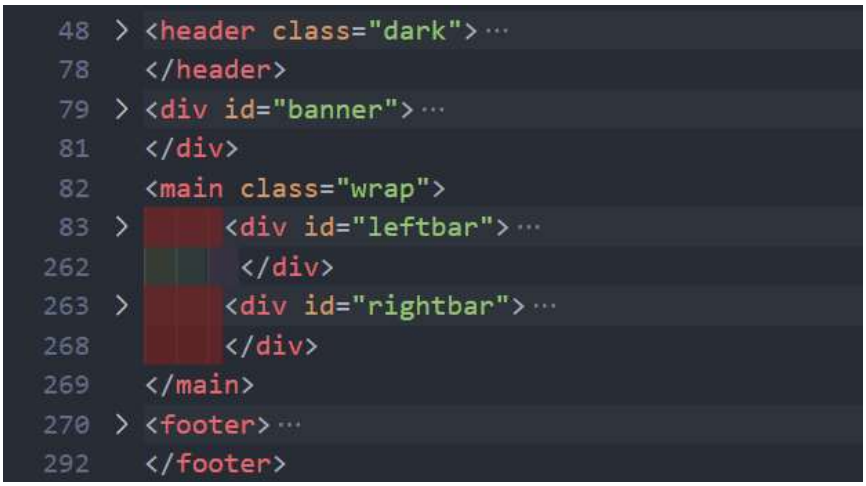

Рисунок 3. 1

У шапці сайту розташовано назву магазину, яка водночас виконує функцію повернення на головну сторінку. Також передбачено місце для навігаційного

меню, у майбутньому — з потенціалом для переходу між категоріями чи сторінками (наприклад, “Про нас”, “Контакти”).

Розділ з фільтрами включає випадаючі списки для вибору категорій і брендів, а також слайдер або два поля для задання діапазону цін. Ці елементи не лише додають зручності, але й виконують важливу функціональність — взаємодію з JavaScript-фільтрами.

Основна частина — каталог товарів — має динамічну природу. Статично в HTML вона містить лише контейнер, в який через JavaScript підвантажуються картки товарів. Це дозволяє зберігати легкість сторінки й відображати лише актуальні товари, згідно з фільтрами.

CSS-оформлення

Зовнішній вигляд сторінки оформлений через кастомні стилі, з акцентом на простоту, чистоту дизайну та читабельність. Основна кольорова схема — нейтральна, з використанням світлого фону, темного тексту, плавних тіней та заокруглених кутів для елементів.

Були застосовані стилі для вирівнювання блоків у сітку, оформлення карток, заголовків, кнопок, фільтрів. Усі стилі було організовано у зовнішньому CSS-файлі для зручності масштабування й підтримки.

```
#main_nav, #main_nav li, #main_nav a{
  display: flex;
  flex-flow: row wrap;
  height: 4.16vmax;
  align-items: center;
  justify-content: center;
}
#main_nav a{
  padding:0 1.5vmax;
  color: #3c3d41;
  text-transform: uppercase;
  font-weight: bold;
}
#main_nav a:hover{
  color: #fff;
  background-color: #3c3d41;
}
```

Рисунок 3. 2

Додатково для елементів навігації, кнопок та фільтрів реалізовано hover-ефекти, що покращують інтерактивність інтерфейсу. Враховано також однаковий відступ між товарами незалежно від їх кількості.

JavaScript та динамічне формування контенту

Найважливішим елементом у реалізації сторінки стало динамічне підвантаження товарів. Всі дані зберігаються у локальному файлі products.json, що містить масив товарів з їхніми властивостями: id, назва, категорія, бренд, ціна, зображення. Після завантаження сторінки JavaScript-функція зчитує цей файл, створює DOM-елементи карток товарів і вставляє їх у контейнер сторінки.

```
fetch('products.json')
  .then(response => response.json())
  .then(products => {
    const container = document.querySelector('.list_product');

    products.forEach(product => {
      const card = document.createElement('div');
      card.className = 'card';

      card.innerHTML = `
        <div class="top_part_card">
          <img src="${product.image}" alt="${product.name}">
        </div>
        <div class="bottom_part_card">
          <h2 class="product_title">${product.name}</h2>
          <p class="product_brand">Бренд: <i>${product.brand}</i></p>
          <p class="product_category">Категорія: ${product.category}</p>
          <p class="product_price">${product.price} грн</p>
          <button class="btn">Додати в кошик</button>
        </div>
      `;

      container.appendChild(card);
    });
  })
```

Рисунок 3. 3

Формування карток виконується за шаблоном: створення елемента div із класом card, вкладання зображення, заголовка, ціни, назви бренду. Якщо зображення для товару відсутнє — автоматично підставляється заглушка.

```
.catch(error => console.error('Помилка завантаження товарів:', error));
```

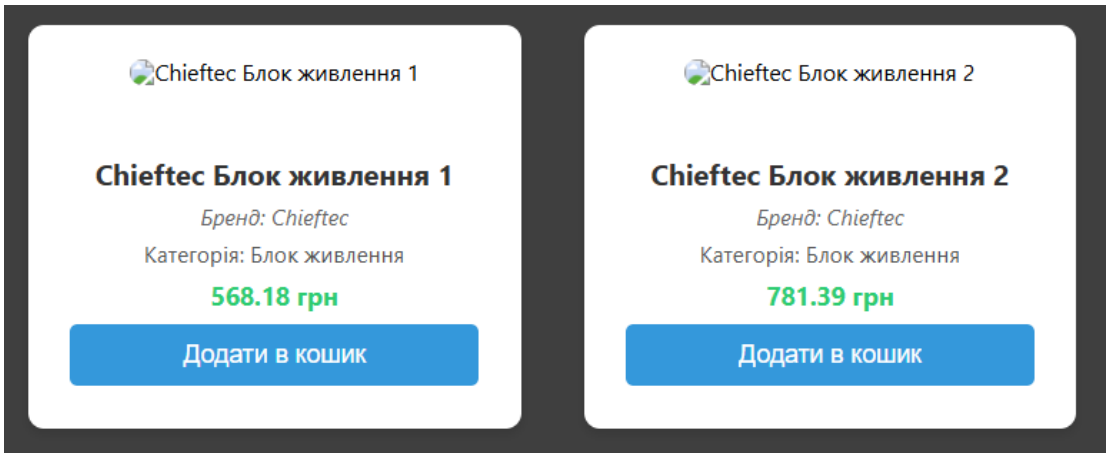

Рисунок 3. 4

Використання JavaScript дозволило реалізувати логіку фільтрації (описується в наступному підрозділі), забезпечити масштабованість (можливість додати більше товарів у JSON-файл), і загалом — знизити навантаження на сторінку при кожному оновленні.

Розробка головної сторінки стала основою для подальшого функціонування магазину. Було закладено базові логічні блоки, підключено стилі, реалізовано динамічне завантаження контенту, що значно підвищило гнучкість. Усі реалізації виконані вручну, без фреймворків, що демонструє розуміння базових вебтехнологій.

3.2. Реалізація карток товарів

Однією з ключових складових інтерфейсу інтернет-магазину є картки товарів. Саме вони відображають основну інформацію про кожну одиницю продукції та слугують точкою входу для взаємодії користувача з товаром. Якісне виконання карток безпосередньо впливає на зручність користування, швидкість прийняття рішення про покупку та загальне враження від сайту.

На початковому етапі були визначені базові компоненти картки товару:

- Зображення товару або заглушка.
- Назва товару.
- Категорія (наприклад, «Процесор», «Відеокарта», тощо).
- Виробник (бренд).
- Короткий опис або модель.

- Ціна.
- Кнопка дії («Додати до кошика» або «Детальніше»).

Усі ці елементи об’єднано в єдиний блок із семантично правильною HTML-структурою. Для організації блоків використовували div-контейнери з відповідними класами, що дозволило легко застосовувати стилізацію та логіку взаємодії.

Дані для заповнення карток беруться з зовнішнього файлу у форматі JSON. Кожен товар описано у вигляді об'єкта з відповідними полями: id, name, category, brand, model, price, image. Це дозволило централізовано управляти асортиментом і динамічно генерувати картки без дублювання коду.

За допомогою JavaScript реалізовано цикл, який проходить масив об'єктів і на його основі створює відповідну HTML-структуру для кожного товару. Додавання нових товарів зводиться до розширення JSON-файлу, що є зручно для масштабування проєкту.

```
79  products.forEach(product => {
80    const card = document.createElement('div');
81    card.className = 'card';
82
83    card.innerHTML = `
84      <div class="top_part_card">
85        <img src="${product.image}" alt="${product.name}">
86      </div>
87      <div class="bottom_part_card">
88        <h2 class="product_title">${product.name}</h2>
89        <p class="product_brand">Бренд: <i>${product.brand}</i></p>
90        <p class="product_category">Категорія: ${product.category}</p>
91        <p class="product_price">${product.price} грн</p>
92        <button class="btn">Додати в кошик</button>
93      </div>
94    `;
95
96    container.appendChild(card);
97  });
```

Рисунок 3. 5

Особливу увагу приділено відображенню товарного зображення. У разі, якщо конкретне зображення відсутнє, виводиться заглушка, яка попереджає

користувача про відсутність фото, але водночас зберігає цілісність дизайну сторінки. Це реалізовано шляхом перевірки значення поля image: якщо воно пусте, вставляється стандартне зображення-заглушка.

Дизайн картки розроблявся відповідно до загального стилю інтерфейсу магазину. Для стилізації використовувались сучасні можливості CSS — flex-контейнери, тіні, заокруглення кутів, фонова заливка, кольори відповідно до тематики сайту. Всі стилі об’єднані в окремий CSS-файл, щоб забезпечити розділення логіки та презентації.

```css
/* Стилі для картки товару */
.card {
  width: 100%;
  background-color: #fff;
  border-radius: 10px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  overflow: hidden;
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  transition: transform 0.3s ease, box-shadow 0.3s ease;
}
```

Рисунок 3. 6

```css
/* Ціна товару */
.product_price {
  font-size: 16px;
  font-weight: bold;
  color: #2ecc71;
  margin-bottom: 15px;
}

/* Стилі для кнопки */
.btn {
  width: 100%;
  padding: 10px;
  background-color: #3498db;
  color: #fff;
  border: none;
  border-radius: 5px;
  font-size: 16px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}
```

Рисунок 3. 7

Картки містять базову інтерактивність. Наприклад, при наведенні змінюється тінь або колір рамки. Це досягається через ефекти hover у CSS. Також

кнопки мають змінений стиль при наведенні та натисканні, що дає користувачу зворотний зв’язок.

```
818   /* Ефект при наведенні на картку */
819   .card:hover {
820     transform: translateY(-10px);
821     box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15);
822   }
```

Рисунок 3. 8

Кожна картка є частиною сітки, що генерується автоматично залежно від кількості товарів. Це дозволило уникнути ручного розміщення блоків та забезпечити рівномірний розподіл на сторінці. Застосовано класичний grid-розподіл товарів на головній сторінці магазину.

Під час побудови карток враховувалася можливість подальшої інтеграції додаткових функцій. Наприклад:

- Виведення рейтингу (зірки).
- Мітки «Новинка», «Знижка».
- Кнопки «В обране», «Порівняти».
- Відображення наявності на складі.

Для цих можливостей у коді закладено відповідні місця та класи, навіть якщо зараз ці функції не використовуються. Такий підхід дозволяє розширювати функціонал без необхідності переписування існуючих блоків.

Підсумовуючи, можна сказати, що картки товарів у цьому інтернет-магазині реалізовані як гнучкий, масштабований і зручний інтерфейсний блок. Вони дозволяють динамічно відображати асортимент, забезпечують приємний візуальний досвід для користувача і підготовлені до розширення функціоналу. Саме з карток починається знайомство користувача з товаром, тому якість їх реалізації має вирішальне значення для успішності проєкту.

3.3. Фільтрація товарів за категоріями, брендами та ціною

Фільтрація є важливим елементом взаємодії користувача з інтерфейсом інтернет-магазину. Вона дозволяє швидко знайти необхідні товари серед

великого асортименту, використовуючи основні параметри: категорію, бренд і ціну. У межах реалізації інтерфейсу було створено базову функціональність фільтрації, яка працює на основі взаємодії з даними у форматі JSON.

Першим кроком стало визначення списку категорій та брендів. До категорій було віднесено: «Процесор», «Материнська плата», «Оперативна пам’ять», «Відеокарта», «Накопичувач», «Блок живлення», «Охолодження» та «Корпус». Для кожного товару в JSON передбачено поля category, brand і price, що дозволяє легко реалізувати перевірку умов фільтрації.

У візуальній частині фільтри реалізовано у вигляді чекбоксів для категорій та брендів, а також полів вводу для цінового діапазону. При зміні будь-якого з параметрів фільтрація застосовується миттєво, без перезавантаження сторінки. Це забезпечується використанням JavaScript, який динамічно оновлює вміст каталогу товарів.

```
152  // Обробка кліків по фільтрам бренду / категорії
153  document.addEventListener('click', function(e) {
154    const target = e.target;
155    if (target.matches('.anime')) {
156      e.preventDefault();
157
158      const category = target.getAttribute('data-category');
159      const brand = target.getAttribute('data-brand');
160
161      if (category !== null) activeCategory = category || null;
162      if (brand !== null) activeBrand = brand || null;
163
164      applyFilters();
165    }
166  });
```

Рисунок 3. 9

Особливу увагу було приділено сумісному використанню кількох фільтрів одночасно. Наприклад, користувач може обрати категорію «Відеокарта», бренд «MSI» і вказати ціновий діапазон від 10 000 до 20 000 грн. Система повинна виводити лише ті товари, які одночасно відповідають усім трьом умовам.

```
// Обробка фільтра по ціні
document.getElementById('search-btn').addEventListener('click', () => {
  activeMinPrice = parseInt(document.querySelector('input[name="minPrice"]').value) || 0;
  activeMaxPrice = parseInt(document.querySelector('input[name="maxPrice"]').value) || Infinity;
  applyFilters();
});
```

Рисунок 3. 10

У структурі HTML фільтри було винесено в окрему колонку зліва від списку товарів. Такий підхід відповідає типовим патернам розташування елементів фільтрації в e-commerce. Групування фільтрів за типом — окремо категорії, бренди та ціна — робить їх сприйняття зручним для користувача.

Функціональність фільтрації не тільки покращує юзабіліті інтерфейсу, а й дозволяє спростити процес прийняття рішення про покупку. Якщо користувач бачить лише ті товари, які відповідають його вимогам, це пришвидшує процес вибору.

```
// Основна функція фільтрації
function applyFilters() {
  const filtered = allProducts.filter(product => {
    const price = Number(product.price);
    const matchesPrice = price >= activeMinPrice && price <= activeMaxPrice;
    const matchesBrand = activeBrand ? product.brand === activeBrand : true;
    const matchesCategory = activeCategory ? product.category === activeCategory : true;
    return matchesPrice && matchesBrand && matchesCategory;
  });

  renderProducts(filtered);
}
```

Рисунок 3. 11

Загалом реалізація фільтрації є важливим етапом побудови інтерфейсу інтернет-магазину. Вона надає користувачу інструменти для ефективного керування каталогом товарів і суттєво підвищує якість взаємодії з системою. Хоча реалізовано базову фільтрацію, подальші етапи розробки можуть передбачати вдосконалення — наприклад, додавання фільтра за обсягом пам’яті, частотою або іншими технічними характеристиками товарів.

3.4. Адаптивність та кросбраузерність інтерфейсу

Під час розробки інтерфейсу інтернет-магазину особлива увага приділялася забезпеченню адаптивності та кросбраузерності, навіть у межах лише десктопної (ПК) версії. Це важливо для гарантування коректного

відображення і функціонування сайту на екранах різних розмірів та в різних браузерах, зокрема Google Chrome, Mozilla Firefox, Microsoft Edge та Opera.

Адаптивність у межах ПК-версії

Адаптивність інтерфейсу означає здатність елементів змінювати свої розміри, положення та пропорції відповідно до ширини вікна браузера. Навіть якщо сайт не орієнтований на мобільні пристрої, користувачі можуть мати різні розміри моніторів або відкривати сторінку у вікні меншого розміру.

У рамках цієї реалізації адаптивність забезпечувалася за рахунок використання відносних одиниць вимірювання, таких як відсотки (%), em, rem, vw/vh, а також за допомогою гнучкої верстки Flexbox.

Приклад стилів, де ширина елементів встановлюється у відсотках:

```
.product-card {
width: 30%;
margin: 1%;
display: inline-block;
}
```

Такий підхід дозволяє карткам товарів рівномірно заповнювати рядок та змінювати свою ширину при зміні розміру вікна браузера. У цьому випадку три картки поміщаються в одному рядку при ширині понад 1200 пікселів, але можуть автоматично перебудовуватись на дві або одну вужчу картку при зменшенні ширини.

Також важливою є властивість flex-wrap: wrap, яка дає змогу елементам переноситися на новий рядок при нестачі простору. У контейнері для товарів вона застосовується наступним чином:

```
.product-container {
display: flex;
flex-wrap: wrap;
justify-content: space-between;
}
```

Така структура дозволяє зберегти відступи між картками і забезпечити симетричне вирівнювання. Крім того, адаптивність була реалізована на рівні типографіки: шрифт може змінювати розмір залежно від розміру екрану, використовуючи одиниці rem або vw:

```
h1 {
font-size: 2.5vw;
}
```

Це гарантує, що заголовки залишаються пропорційними незалежно від ширини екрана, зберігаючи при цьому ієрархію елементів і візуальну логіку.

У разі надто вузьких екранів користувач ПК також може побачити перебудову інтерфейсу без порушення структури. Наприклад, фільтри можуть переходити в окремий вертикальний блок або плавно зменшуватися в розмірах, не перекриваючи основний контент.

Кросбраузерність

Кросбраузерність інтерфейсу означає його здатність однаково виглядати та працювати у різних браузерах, незалежно від особливостей їх рендерингу чи версії рушія. Це один із критичних аспектів зручності користувача, адже кожен браузер має власну реалізацію стандартів HTML/CSS.

Під час створення інтерфейсу були враховані типові особливості та відмінності браузерів. Для цього використовувалися такі підходи:

1. Уніфіковані стилі за замовчуванням: кожен браузер має свій набір стандартних стилів, які можуть відрізнятись (наприклад, для кнопок, форм, заголовків). Для нейтралізації цих відмінностей застосовувалося так зване "скидання стилів" — reset.css або normalize.css.

```
{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    }
```

Це дозволяє починати верстку з чистого аркуша та забезпечує передбачуваність стилів у різних браузерах.

2. Використання CSS-префіксів: певні властивості, такі як flexbox або градієнти, можуть мати альтернативні варіанти запису з префіксами для браузерів (-webkit-, -moz-, -ms-). Хоча більшість сучасних браузерів вже підтримують стандартизовані синтаксиси, іноді доцільно додавати префікси для сумісності.

```
.example {
    -webkit-user-select: none;
    -moz-user-select: none;
    user-select: none;
    }
```

3. Врахування різного рендерингу шрифтів та елементів у браузерах. Наприклад, у Firefox деякі елементи можуть мати більші відступи за замовчуванням, або в Edge — по-іншому обраховуватись ширина блоку. У таких випадках застосовується візуальне вирівнювання через додаткові правила.
4. Тестування: після реалізації інтерфейсу сайт тестувався у різних браузерах для виявлення можливих відмінностей. Зокрема, перевірялися:
    - відображення шрифтів і заголовків;
    - коректність розміщення товарних карток;
    - стабільність фільтрів;
    - відображення hover-ефектів та анімацій.
5. Обмеження використання нестабільних властивостей. Уникалися експериментальні чи рідко підтримувані CSS-функції, натомість використовували перевірені підходи, що мають високу підтримку за даними сервісів типу caniuse.com.

Загалом, кросбраузерна реалізація базувалася на принципі "мінімум нестабільності". Усі ключові стилі, а також скрипти, тестувалися у щонайменше 4 популярних браузерах для виявлення потенційних проблем.

У результаті розробки було досягнуто повноцінної адаптивності для всіх типових ПК-моніторів — від невеликих ноутбуків до широкоформатних екранів. Інтерфейс плавно змінює свою структуру, не втрачаючи читабельності, логіки та візуальної привабливості. Кросбраузерність забезпечує стабільну роботу незалежно від обраного браузера. Це особливо важливо у контексті інтернет-магазину, де комфорт і зручність взаємодії з сайтом безпосередньо впливають на користувацький досвід і потенційні продажі.

3.5. Впровадження функціональних елементів

Крім візуального відображення товарів та структури сайту, важливою складовою інтерфейсу інтернет-магазину є функціональні елементи, які забезпечують зручну взаємодію користувача з сайтом. У цьому підрозділі розглянуто реалізацію таких елементів, як кнопки, механізм пошуку, відкриття модальних вікон і навігація по категоріях. Їх функціональність реалізовано за допомогою JavaScript та jQuery, що дозволило забезпечити динамічну поведінку без перезавантаження сторінки.

Кнопки та інтерактивні блоки

Кнопки у межах даного інтерфейсу виконують різні функції: додавання товару до кошика, відкриття модального вікна авторизації, запуск фільтрації та завантаження додаткових товарів. Основна увага приділялася тому, щоб кожна кнопка:

- реагувала на події натискання;
- мала візуальний зворотний зв’язок (наприклад, hover-ефект);
- виконувала чітко визначену функцію.

Найважливішою функцією є додавання товарів до кошика. Кожна картка має кнопку, що дозволяє клонувати товар і додати його до блоку з кошиком.

```
btn.addEventListener('click', () => {
  const productClone = product.cloneNode(true);
  contentCart.appendChild(productClone);
  showData();
  productClone.style.height = '';
  const removeBtn = productClone.querySelector('button');
  removeBtn.textContent = 'Видалити з корзини';
  removeBtn.addEventListener('click', () => {
    productClone.remove();
    showData();
  });
});
```

Рисунок 3. 12

Модальне вікно для входу в кабінет користувача

На головній сторінці розміщено кнопку «Увійти». Натискання на неї викликає появу модального вікна з формою входу. Ця форма містить поля для логіну та паролю, а також кнопку надсилання даних.

Для реалізації відкриття та закриття модального вікна використовується додавання/видалення класу "open" та одночасне блокування прокрутки фону (body.style.overflow = 'hidden').

```
$('#sign').click(function(){
  $('#black_fill').toggleClass('open');
  $('#modal').toggleClass('open');
  document.body.style.overflow = 'hidden';
});
```

Рисунок 3. 13

Цей підхід дозволяє без додаткових бібліотек реалізувати плавне відкриття модального вікна з затемненням фону, забезпечуючи фокус уваги користувача на формі.

Пошук

Пошук є інтуїтивно зрозумілим функціоналом, доступним за допомогою кнопки з ID search_button. При натисканні на неї розкривається панель пошуку, яка, ймовірно, має бути доповнена функцією фільтрації за ключовими словами або назвою товару (у подальшому). Наразі логіка лише відповідає за відображення/приховування пошукового блоку.

```
61    $('#search_button').click(function(){ //
62      $('#search_panel').toggleClass('open');
```

Рисунок 3. 14

Таке рішення дозволяє зберігати компактність інтерфейсу, показуючи пошук лише тоді, коли він потрібен користувачу.

Завантаження додаткових товарів

У нижній частині головної сторінки розміщено кнопку "Завантажити ще". Вона додає нові товарні картки до блоку з продуктами без оновлення сторінки. Реалізовано через додавання HTML-елементів безпосередньо в DOM.

Ця функція слугує демонстраційною (у коді підвантажуються однакові товари), але технічно дозволяє реалізувати механізм lazy-loading або пагінацію у майбутньому.

Розкривні блоки категорій

У лівій панелі сайту категорії згруповано за допомогою блоків list_box. Кожна категорія має заголовок title_box, який можна розгортати/згортати. Це забезпечується за допомогою класів .open, які додаються або прибираються при кліку.

Цей механізм дозволяє зберігати компактність сторінки при великій кількості доступних категорій.

```
40  $('.title_box').click(function() //розкриття фільтру товарів
41  {
42    $(this).toggleClass('open');
43    $(this).next('.list_link').toggleClass('open');
44  });
```

Рисунок 3. 15

Кошик

Один із найважливіших функціональних елементів — кошик. Він розміщується в модальному вікні з ID #cart_box. Його відкривають кнопки з ID #cart, #order, або класом .title_cart. Вміст кошика формується динамічно, підраховується загальна сума та кількість товарів.

Ця функція також включає кнопку "Видалити з кошика", що прибирає відповідний товар із DOM та оновлює загальну суму.

У майбутньому кошик можна доповнити: збереженням у localStorage, переходом до оформлення замовлення або інтеграцією з серверною частиною.

ВИСНОВКИ

У процесі виконання дипломної роботи було здійснено повноцінний цикл розробки користувацького інтерфейсу для інтернет-магазину, призначеного для продажу комплектуючих до персональних комп’ютерів. З урахуванням актуальності теми та зростаючих вимог до якості вебінтерфейсів, основна увага приділялась створенню логічної, простої, інтуїтивно зрозумілої системи взаємодії між користувачем та електронним торговим ресурсом.

На першому етапі дослідження було проведено теоретичний аналіз сучасних підходів до побудови інтерфейсів користувача в контексті веброзробки. Розглянуто поняття UI та UX, визначено відмінності та взаємозв’язок між ними. Встановлено, що якісний інтерфейс значною мірою визначає ефективність комерційного сайту, впливає на показники конверсії, поведінку користувача, рівень довіри до ресурсу. Особливу увагу приділено історичному розвитку інтерфейсів та сучасним технологіям, які їх забезпечують: HTML, CSS, JavaScript, JSON. Також здійснено короткий огляд фреймворків, які використовуються у веброзробці, з аргументованим вибором відмови від них у межах цього проєкту для зосередження на базовій реалізації.

Другий етап був присвячений проєктуванню інтерфейсу. Було сформовано вимоги до структури, зовнішнього вигляду та функціональності майбутнього інтернет-магазину. На основі аналізу користувацьких очікувань, практик UI/UX-дизайну та специфіки онлайн-торгівлі комп’ютерною технікою було виділено ключові компоненти інтерфейсу: головна сторінка, каталог товарів, фільтри,

пошук, картки товарів, кошик, навігація тощо. Проєктування виконувалось безпосередньо в коді, що дало змогу швидко реалізовувати та перевіряти поведінку елементів, уникаючи етапу створення графічного прототипу. Такий підхід виправданий у межах навчального проєкту, оскільки забезпечив швидке тестування та гнучку адаптацію інтерфейсу до вимог користувача.

На третьому етапі відбулася реалізація інтерфейсу. Усі компоненти сайту були створені за допомогою HTML для структурної розмітки, CSS для стилізації та JavaScript для забезпечення динаміки й інтерактивності. Функціональність інтерфейсу включала динамічне завантаження товарів із JSON-файлу, фільтрацію за категорією, брендом і ціною, взаємодію з кошиком, пошук товарів тощо. Весь інтерфейс реалізовано у вигляді односторінкового застосунку (SPA), що дозволяє взаємодіяти з основним контентом без перезавантаження сторінки. Це значно покращує користувацький досвід та підвищує швидкодію ресурсу.

Особливістю розробки було те, що всі елементи були створені вручну, без застосування сторонніх бібліотек чи фреймворків. Це дозволило глибше зануритись у принципи веброзробки, краще зрозуміти логіку побудови інтерфейсів, опрацювати взаємодію елементів DOM-дерева та обробку подій через JavaScript. Окремо варто підкреслити, що використання JSON-файлу як джерела даних дозволило реалізувати поділ логіки та контенту — кожна товарна позиція описувалась окремим об’єктом, що забезпечило легкість у редагуванні та масштабуванні.

Ще одним важливим результатом стало досягнення базової кросбраузерності та масштабованості інтерфейсу. Інтерфейс відображається коректно в основних сучасних браузерах (Chrome, Firefox, Edge), а елементи інтерфейсу гнучко адаптуються до ширини вікна (у межах версії для ПК). Така гнучкість забезпечена завдяки застосуванню сучасних засобів CSS: flexbox, grid, відсоткових відступів, медіа-запитів тощо. Хоча у межах цього проєкту не реалізовано повної мобільної адаптації, структура та стилізація передбачають її подальше впровадження без необхідності значної перебудови коду.

Фінальною перевіркою якості стало тестування реалізованого інтерфейсу на відповідність поставленим вимогам. Було перевірено швидкість завантаження контенту, зручність фільтрації, коректність відображення карток товарів, роботу кошика, пошук, загальний вигляд сторінки. Оцінено також загальне враження користувача (UX) — логіка розміщення елементів, інтуїтивність навігації, простота взаємодії. Усі заплановані функції реалізовані відповідно до очікувань, що свідчить про досягнення поставленої мети.

Таким чином, у межах кваліфікаційної роботи було виконано:

- повний цикл розробки інтерфейсу інтернет-магазину: від теоретичного аналізу до практичного втілення;
- реалізацію динамічного інтерфейсу на чистому HTML, CSS, JavaScript з використанням JSON для зберігання даних;
- забезпечення основного функціоналу інтернет-магазину: каталог, фільтри, пошук, картки товарів, кошик;
- реалізацію основних принципів UI/UX без застосування графічних редакторів (через «прототипування в коді»);

- перевірку працездатності інтерфейсу та відповідності базовим вимогам кросбраузерності й гнучкості.

Результатом стала повноцінна фронтенд-реалізація функціонального інтерфейсу інтернет-магазину, що демонструє здатність застосовувати знання з веброзробки, UI/UX-дизайну та програмування на практиці. Робота може бути основою для подальшого розвитку проєкту: впровадження мобільної версії, серверної логіки, бази даних, розширеного пошуку, авторизації користувачів, системи оплати тощо.

Таким чином, поставлену мету досягнуто, а всі заплановані завдання виконано повною мірою. Отриманий результат має не лише навчальну, а й практичну цінність, що підкреслює успішність проведеного дослідження та реалізації проєкту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Nielsen, J. Usability Engineering. Morgan Kaufmann, 1994.Krug, S. Don't Make Me Think: A Common Sense Approach to Web Usability. New Riders, 2014.
2. W3C. HTML5 Specification. https://html.spec.whatwg.org/
3. MDN Web Docs. HTML, CSS, and JavaScript Reference. https://developer.mozilla.org/
4. Norman, D. A. The Design of Everyday Things. MIT Press, 2013.
5. Bootstrap. Official Documentation. https://getbootstrap.com/
6. UX Design Institute. Principles of User Interface Design. https://www.uxdesigninstitute.com/
7. Web.dev (Google). Learn Responsive Design. https://web.dev/learn/design/
8. ISO 9241-210:2019. Ergonomics of human-system interaction — Human-centred design for interactive systems.
9. Канцур О. Ю. Проектування інтерфейсів користувача: теорія і практика. Харків: НТУ "ХПІ", 2021.
10. UX Planet. Designing E-Commerce Experiences. https://uxplanet.org/
11. W3Schools. JavaScript Tutorial. https://www.w3schools.com/js/
12. Figma. User Interface Prototyping Guide. https://www.figma.com/resources/learn-ui-design/