

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Комп'ютерна інженерія _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Кулькову Вадиму Євгеновичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Веб-сайт інтернет-магазину

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 17 червня 2025 р.

3. Вхідні дані до роботи 1) Архітектура сайту: Headless комерція;

2) Стек технологій: Remix.js, React.js, Tailwind CSS; 3) CMS платформа: Shopify;

4) Протокол взаємодії: GraphQL; 5) Інтеграція: API сервісу доставки Нова Пошта.

4. Перелік питань, що потрібно опрацювати у роботі _____

1) аналіз предметної області;

2) дослідження використаних технологій;

3) розробка алгоритмічного забезпечення;

4) розробка та відлагодження програмного забезпечення;

5) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій _____

Слайд-презентація – 13 слайдів _____

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз предметної області	27.05.25-30.05.25	
2	Дослідження використаних технологій	31.05.25-02.06.25	
3	Розробка алгоритмічного забезпечення	03.06.25-05.06.25	
4	Розробка та відлагодження програмного забезпечення	06.06.25-09.06.25	
5	Оформлення матеріалів кваліфікаційної роботи	10.06.25-11.06.25	
6	Подання кваліфікаційної роботи керівникові та її попередній захист	12.06.25-13.06.25	
7	Подання кваліфікаційної роботи на рецензування	14.06.25-16.06.25	

Дата видачі завдання “ 26 ” травня 2025 р.

Здобувач _____

(підпис)

Керівник роботи _____

(підпис)

ст.викл. Налалія ЄРЬОМІНА

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 56 с., 13 рис., 1 дод., 13 джерел.

ВЕБ ТЕХНОЛОГІЇ, ЕЛЕКТРОННА КОМЕРЦІЯ, REMIX.JS, SHOPIFY, HEADLESS CMS, ІНТЕРНЕТ-МАГАЗИН, АДАПТИВНИЙ ДИЗАЙН, JAVASCRIPT, НОВА ПОШТА

Метою кваліфікаційної роботи є розробка веб-додатку для інтернет-магазину комп'ютерної периферії з використанням веб технологій та передових архітектурних рішень

У ході виконання кваліфікаційної роботи було проведено аналіз предметної області електронної комерції, досліджено сучасні технології веб-розробки та архітектурні підходи для створення високопродуктивних інтернет-магазинів.

Об'єктом дослідження є процеси створення сучасних веб-додатків для електронної комерції з використанням JavaScript. Особливу увагу приділено вивченню headless архітектури та її переваг у порівнянні з традиційними монолітними рішеннями.

У результаті виконання роботи було створено веб сайт «Grind Gear» – інтернет-магазину ігрової периферії. Сайт реалізовано з використанням Remix.js, React.js, Shopify headless CMS та Tailwind CSS. Система включає каталог товарів з фільтрацією, кошик, оформлення замовлень з інтеграцією Nova Poshta та адаптивний дизайн.

ABSTRACT

Bachelor's thesis: 56 pages, 13 figures, 1 appendices, 13 sources.

WEB TECHNOLOGIES, E-COMMERCE, REMIX.JS, SHOPIFY, HEADLESS CMS, ONLINE STORE, RESPONSIVE DESIGN, NOVA POST

The major goal of this thesis is develop a web application for an online computer peripherals store using web technologies and advanced architectural solutions.

The goal of the qualification work is to develop a web application for an online computer peripherals store using web technologies and advanced architectural solutions.

During the qualification work, an analysis of the e-commerce domain was conducted, and modern web development technologies and architectural approaches for creating high-performance online stores were studied.

The object of research is the processes of creating modern web applications for e-commerce using JavaScript. Special attention was paid to studying headless architecture and its advantages compared to traditional monolithic solutions.

As a result of the work, a fully functional web site «Grind Gear» was created – an online store for gaming peripherals. The application was built using Remix.js, React.js, Shopify headless CMS, and Tailwind CSS. The system includes a product catalog with filtering, shopping cart, order checkout with Nova Poshta delivery integration, and responsive design.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Загальне визначення електронної комерції	11
1.2 Аналіз провідних рішень та конкурентів	13
1.2.1 Інтернет-магазин «Зона51»	13
1.2.2 Інтернет-магазин «Telemart».....	14
1.2.3 Інтернет-магазин «WKey».....	16
1.3 Аналіз розроблюваного програмного рішення	17
2 ДОСЛІДЖЕННЯ ВИКОРИСТАНИХ ТЕХНОЛОГІЙ	19
2.1 Мова програмування JavaScript.....	19
2.2 Середовище виконання Node.js	20
2.3 Бібліотека React.js	21
2.4 Фреймворк Remix.....	21
2.5 CMS платформа Shopify	22
2.6 Мова запитів GraphQL API	22
2.7 CSS фреймворк Tailwind CSS	23
2.8 Середовище розробки Visual Studio Code	23
3 РОЗРОБКА АЛГОРИТМІЧНОГО ЗАБЕСПЕЧЕННЯ	25
3.1 Пошук та фільтрація товарів.....	25
3.1.1 Алгоритм текстового пошуку товарів.....	25
3.1.2 Фільтрація за категоріями та параметрами	26
3.1.3 Алгоритм сортування результатів	28
3.2 Управління кошиком та станом кошика.....	29
3.2.1 Додавання товарів до кошика	29
3.2.2 Оновлення кількості та видалення товарів.....	30
3.3 Обробка замовлень та доставки.....	31

3.3.1 Ведення даних користувача	31
4 РОЗРОБКА ТА ВІДЛАГОДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	33
4.1 Реалізація системи навігації та маршрутизації	33
4.2 Розробка системи пошуку товарів.....	35
4.3 Каталог товарів фільтрація та сортування.....	37
4.4 Розробка сторінки детального перегляду товару.....	39
4.5 Система оформлення замовлень.....	40
4.6 Результати тестування	42
ВИСНОВКИ.....	45
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	47
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	49

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

Адаптивний дизайн – підхід до веб-дизайну для коректного відображення на різних пристроях

Електронна комерція – Торгівля товарами та послугами через інтернет

API – Програмний інтерфейс додатку

CMS – Система управління контентом (англ. Content Management System)

DOM – об'єктна модель документа

GraphQL – Мова запитів для API

SSR – Серверний рендеринг (англ. Server-Side Rendering)

UI – Інтерфейс користувача

UX – Досвід користувача (англ. User Experience)

SEO – Оптимізація для пошукових систем

ВСТУП

В умовах невинної цифровізації суспільства та економіки, електронна комерція стала одна з ключових та невід'ємних складових продаж. Вона надає користувачам усе більш доступні, швидкі та ефективні інструменти для придбання широкого асортименту товарів, долаючи традиційні географічні та часові бар'єри.

Особливої актуальності в цьому контексті набуває сегмент онлайн-торгівлі спеціалізованими товарами, яскравим прикладом яких є комп'ютерна периферія. Попит на дану категорію продукції стабільно зростає через зростаючі потреби професіоналів різних галузей, так і постійним інтересом з боку широкого кола ентузіастів та домашніх користувачів, що прагнуть покращити свою взаємодію під час роботи чи геймінгу.

Застосування сучасних підходів до веб-розробки відіграє велику роль у створенні дійсно конкурентоспроможних та успішних рішень у сфері електронної комерції.

Успішність функціонування та довгострокова конкурентоспроможність таких онлайн платформ безпосередньо залежить від гармонійного поєднання низки ключових факторів. Серед них головне значення мають: висока швидкодія системи, що мінімізує час очікування користувача на всіх етапах взаємодії, продумана візуальна привабливість інтерфейсу, яка формує позитивне перше враження та сприяє впізнаваності бренду, а також забезпечення загального безшовного та інтуїтивного користувацького досвіду, що гарантує легкість навігації, простоту пошуку необхідної інформації.

Для досягнення поставленої мети в рамках даної роботи передбачається реалізація ключових функціональних модулів, що є характерними для більшості сучасних платформ електронної комерції. Це, зокрема, включає розробку структурованого каталогу товарів з гнучкими

можливостями фільтрації та сортування для зручного вибору; створення інформативних та деталізованих сторінок продуктів, ефективна система пошуку, розробка функціоналу кошика, а також проектування логічного та безпечного процесу оформлення замовлення. Невід'ємним та важливим аспектом розробки є забезпечення повної адаптивності створюваного веб-сайту, що гарантує його коректне та зручне відображення на різних типах пристроїв, від настільних комп'ютерів до мобільних телефонів, а також приділення належної уваги оптимізації продуктивності.

Метою даної передатестаційної роботи є розробка функціонального та конкурентоспроможного веб-сайту електронної комерції, що буде спеціалізуватися на продажу комп'ютерної периферії та аксесуарів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальне визначення електронної комерції

Електронна комерція (e-commerce) це сукупність процесів, пов'язаних із купівлею-продажем товарів та послуг а також здійсненням фінансових транзакцій за допомогою електронних мереж. Ця галузь охоплює широкий спектр діяльності: від онлайн продажів та маркетплейсів до інтернет-банкінгу, електронних платіжних систем та онлайн бронювання послуг. Розвиток електронної комерції тісно пов'язаний з розвитком інтернет-технології. Саме ці технології дали можливість продавати товари незалежно від відстані, робити багато процесів автоматичними та краще розуміти потреби кожного окремого покупця.

Електронна комерція почалася з 1960-х років. Тоді компанії почали використовувати електронний обмін даними, щоб передавати ділові документи, наприклад, замовлення товарів чи рахунки. Але справжній бум стався, коли на початку 1990-х років з'явився і став популярним Інтернет. Перші покупки через Інтернет, які започаткували сучасну електронну комерцію, відбулися в середині 1990-х. У 1994 році компанія NetMarket вперше продала CD диск онлайн. А в 1995 році з'явилися такі відомі компанії, як Amazon, що спочатку продавав книги, та eBay, який працював як інтернет-аукціон. Саме ці події стали основою для швидкого розвитку торгівлі в Інтернеті.

Розвитку електронної комерції наприкінці ХХ – на початку ХХІ століття сприяло декілька важливих речей. По-перше, популяризація інтернету. По-друге поява безпечних способів оплати. По-третє легка та швидка доставка товарів

Завдяки електронній комерції змінилися підходи до ведення бізнесу. Тепер навіть невеликі компанії можуть легко продавати свої товари та

послуги через Інтернет клієнтам по всьому світу, не витрачаючи гроші на оренду чи будівництво звичайних магазинів. Це допомагає зменшити витрати і більше уваги приділяти якості продукції. До того ж, багато процесів стали автоматичними, наприклад, приймання замовлень, спілкування з постачальниками та обслуговування клієнтів. Оскільки торгівля відбувається онлайн, веб-сайти та мобільні додатки стали дуже важливими. Вони є місцем, де компанії рекламують свої товари, продають їх та спілкуються з покупцями. Крім продажу, веб-сайти та додатки дозволяють збирати інформацію про те, як поведуться користувачі, що їм подобається та що їм потрібно. Це допомагає компаніям краще планувати свою рекламу та створювати пропозиції, які будуть цікаві конкретним людям.

Хоча інтернет-торгівля має багато плюсів, існують і певні мінуси. Наприклад, купуючи онлайн, людина не може потримати товар в руках чи приміряти, що іноді призводить до розчарування, коли отримана річ виявляється не такою, як очікувалося. Ще одна складність – це велика кількість конкурентів. Розпочати бізнес в інтернеті простіше, ніж відкрити звичайний магазин, і це добре, але це також означає, що й інші можуть легко стати вашими конкурентами.

В залежності від того, хто і кому продає, електронну комерцію можна розділити на кілька типів. Найчастіше зустрічається торгівля між бізнесом та звичайним покупцем (це називається B2C), коли фірма продає свої товари людям через інтернет-магазин. Існує також торгівля між різними компаніями (B2B). Крім того, люди можуть продавати речі один одному (C2C), наприклад, через такі сайти, як OLX. А буває й так, що людина пропонує свої послуги або товари компанії (C2B). На сьогоднішній день інтернет-торгівля продовжує активно розвиватися: все більшу роль відіграють продажі через соціальні мережі, компанії використовують штучний інтелект для спілкування з клієнтами та покращення своєї роботи, а також поєднують онлайн та офлайн магазини для більшої зручності покупців.

1.2 Аналіз провідних рішень та конкурентів

1.2.1 Інтернет-магазин «Зона51»

Зона51 – це інтернет-магазин, який спеціалізується на продажу комп'ютерної периферії та є одним з найвідоміших в Україні спеціалізованих магазинів, орієнтованих на геймерів та ентузіастів кіберспорту та людей які хочуть мати більше комфорту за своїм робочим місцем. Асортимент магазину зосереджений на високоякісній периферії що покращує користувацький досвід

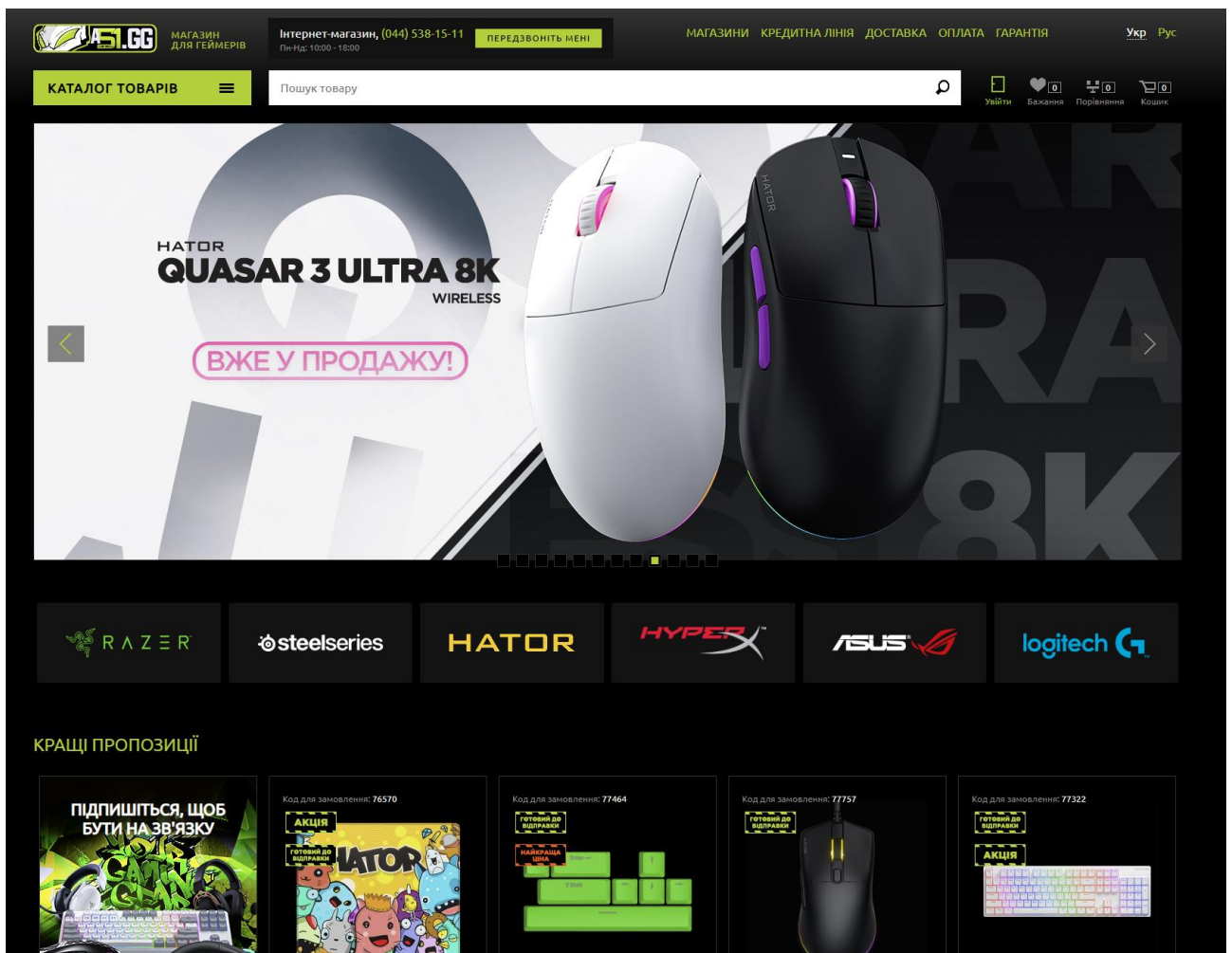


Рисунок 1.1 – Головна сторінка сайту Зона51

Інтернет магазин має низку переваг, особливо для цільової аудиторії.

Асортимент сайту дуже ретельно підібраний зазвичай це моделі від провідних та світових брендів. Навігація сайтом є інтуїтивно зрозумілою, категорії товарів чітко структуровані, що дозволяє легко знаходити потрібну периферію. Особливістю сайту є наявність блогу з оглядами новинок та інструкціями по вибору. Сторінки товарів містять детальну інформацію, якісні фотографії та, часто, відеоогляди. Функціонал «Порівняння товарів» реалізований зручно та можна детально порівняти чим товар відрізняється один від одного.

До недоліків сайту можна віднести те, що візуальний стиль сайту, хоч і відповідає геймерській тематиці, але може здатися що інтерфейс перевантажений кольорами там, багато користувачів би віддали перевагу більш мінімалістичному дизайну. Іноді швидкість завантаження сторінок з великою кількістю графічного контенту може бути не оптимальною. Хоча інформації про товари багато, відсутність розширених фільтрів за деякими специфічними технічними параметрами може ускладнити вибір для дуже вимогливих користувачів.

1.2.2 Інтернет-магазин «Telemart»

Цей інтернет-магазин спеціалізується на продажі комп'ютерних комплектуючих та периферії. Основний акцент робиться на ігрові пристрої та компоненти для збірки й покращенню комп'ютерів.

Інтернет магазин має функціональний дизайн, орієнтований на швидкий доступ до великого каталогу товарів. Структура меню та категорій добре продумана, що полегшує пошук комплектуючих та периферії.

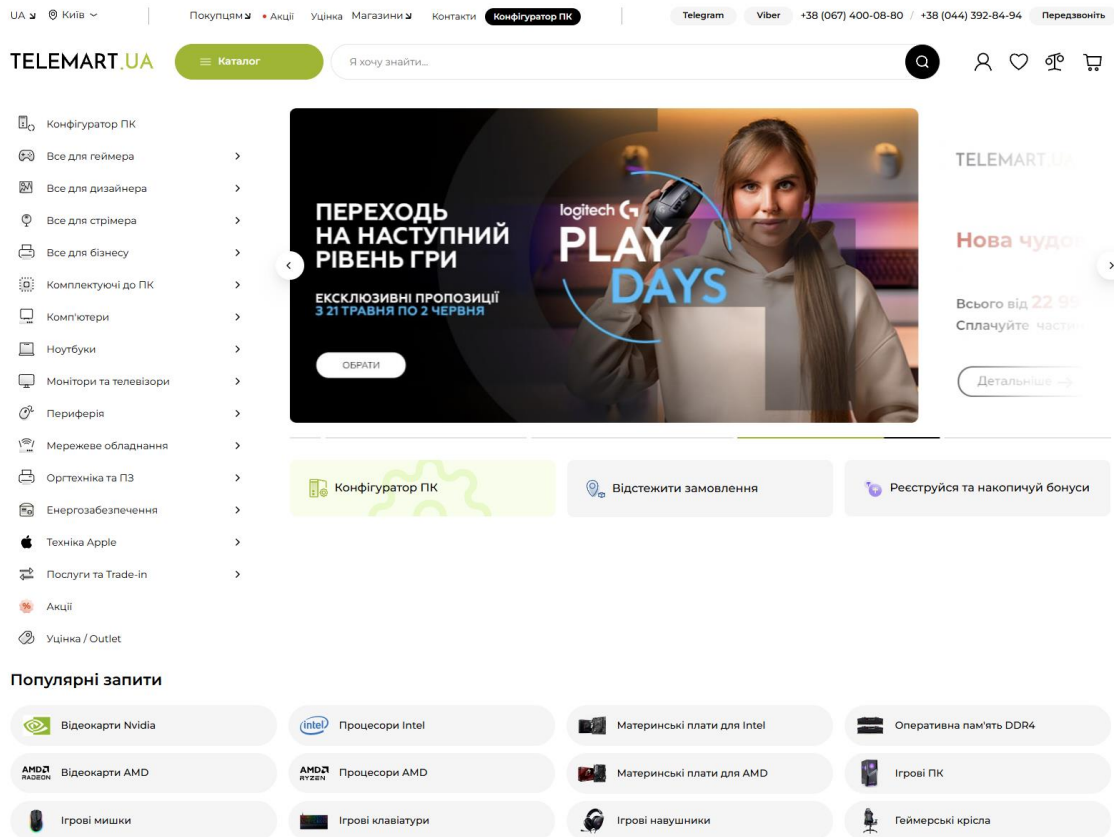


Рисунок 1.2 – Головна сторінка інтернет-магазину «Telemart»

Однією з ключових фішок сайту є деталізований конфігуратор ПК, який дозволяє користувачам самостійно підібрати компоненти для збірки комп'ютера, перевіряючи їх сумісність в режимі реального часу. Картки товарів містять вичерпні технічні характеристики, фотографії та відгуки покупців. Сайт також пропонує корисний контент у вигляді новин, оглядів та статей у власному блозі, що допомагає користувачам бути в курсі останніх тенденцій. Система фільтрації товарів досить розгалужена, дозволяючи точно налаштувати параметри пошуку.

Але також на сайті можна зустріти і недоліки. хоч сайт і функціональний, може здатися дещо застарілим та перевантаженим інформацією. Велика кількість текстових блоків та технічних деталей на сторінках може ускладнювати візуальне сприйняття. Мобільна адаптація сайту присутня, але користувацький досвід на смартфонах міг би бути більш плавним та інтуїтивним. Іноді навігація між численними підкатегоріями та

фільтрами може бути не найзручнішою. Хоча асортимент товарів широкий, візуальне представлення деяких товарів (кількість та якість фото) може поступатися конкурентам.

1.2.3 Інтернет-магазин «WKey»

«WKey» представляє собою інтернет-магазин, що спеціалізується на продажу високоякісної та часто ексклюзивної геймерської периферії, з особливим акцентом на ігрові миші, килимки та аксесуари до них. Магазин орієнтований на вимогливих геймерів та ентузіастів, які шукають найкращі рішення для досягнення високих результатів в іграх та максимального комфорту.

Дизайн сайту «WKey.com.ua» сучасний, чистий та візуально привабливий, з акцентом на якісних зображеннях продукції. Навігація інтуїтивно зрозуміла, категорії товарів та фільтри за брендами дозволяють легко орієнтуватися в асортименті. Сайт має корисні інформаційні блоки на головній сторінці. Фішкою є розділ «Дивилися», що дозволяє швидко повернутися до переглянутих товарів. Процес оформлення замовлення виглядає простим та зрозумілим, з можливістю реєстрації або швидкого замовлення, вибору різних способів доставки та оплати. Наявність блогу та інформації про бренди свідчить про бажання надати користувачам більше деталей про бренди та моделі, які продає сайт.

Сайт також має свої недоліки. Також сайту не вистачає інтерактивних демонстрацій можливостей товарів або детальних відеооглядів прямо на сторінках продуктів. Функція порівняння товарів, яка важлива при виборі між дорогими моделями, повинна бути максимально детальною та легкою у використанні.

The screenshot displays the homepage of the WKEY internet store. At the top, there is a navigation bar with links for 'Про нас', 'Блог', 'Доставка і оплата', 'Гарантії', 'Бренди', and 'Акції'. Below this is the WKEY logo and a search bar. A phone number '050-220-30-50' and a contact time 'Пн-Пт: 10:00-18:00' are also visible. A navigation menu includes 'Мишки', 'Килимки', 'Аксесуари', 'Lamzu', 'Ninjutso', 'Odin Gaming', 'G-Wolves', and 'XTEN'. A 'Дивилися' button is on the right.

The main content area features two large promotional banners. The left banner is for the 'Ninjutso Sora V2' mouse, highlighting its weight (39 grams), 'NinjaForce' technology, 8000 Hz polling rate, PixArt 3395 sensor, and 26,000 DPI. The right banner is for the 'WLMouse Beast X Max' magnetic alloy mouse, priced at 6999 ₴.

Below the banners are six service icons: 'Відправка щодня' (Daily shipping), 'Служба турботи' (Care service), 'Оплата частинами' (Installment payment), 'Обмін і повернення' (Exchange and returns), 'Ексклюзивно в WKEY' (Exclusive to WKEY), and 'Програма лояльності' (Loyalty program).

The bottom section is titled 'Серія Lamzu Maya X' and displays five product cards for different mouse models: Charcoal Black, White, Cloud Gray, Aimlabs Maya X 8K, and Purple Shadow. Each card shows the mouse image, name, star rating, availability status, and price (5699 ₴).

Рисунок 1.3 – Головна сторінка інтернет-магазину «Wkey»

1.3 Аналіз розроблюваного програмного рішення

Проаналізувавши існуючі інтернет-магазини комп'ютерної периферії, виділивши їхні сильні та слабкі сторони, а також враховуючи специфіку цільової аудиторії, було прийнято рішення щодо розробки сайту «Grind Gear». Цей додаток орієнтований на забезпечення зручного та швидкого доступу до каталогу товарів як з десктопних комп'ютерів, так і з мобільних пристроїв через веб-браузери.

Для веб-сайта планується реалізувати наступні та ключові функції:

- перегляд товарів;
- сторінка товарів;

- пошук товарів;
- фільтрація товарів;
- додавання товарів до списку обраного;
- система авторизації;
- кошик товарів;
- оформлення замовлення;
- перегляд замовлень;
- порівняння товарів;
- розділ Блог.

Плануючи архітектуру для майбутнього веб-сайту «Grind Gear», наша головна мета – зробити його гнучким, щоб він міг легко розвиватися та змінюватися в майбутньому. Ми плануємо розділити сайт на дві основні частини. Перша частина – це те, що бачитимуть користувачі: привабливий дизайн, зручні кнопки та сторінки. Друга частина – це свого роду «мозок» сайту, який буде відповідати за управління товарами, обробку замовлень та всю іншу внутрішню роботу. Такий поділ дозволить нам покращувати кожен частину окремо, не впливаючи на іншу. Для зовнішнього вигляду сайту ми використаємо сучасні технології, щоб він працював швидко, виглядав сучасно, а окремі елементи дизайну можна було б легко використовувати повторно. Для управління самим магазином (товарами, замовленнями тощо) ми будемо використати вже існуючу надійну платформу для інтернет-торгівлі, яка дозволяє підключатися до її функцій через спеціальний програмний інтерфейс (API). Такий підхід, коли сайт складається з окремих функціональних блоків, значно спростить процес розробки, тестування та подальшої підтримки. Це також дозволить команді ефективніше працювати над різними аспектами сайту одночасно. Це зробить сайт легшим для підтримки та подальшого розвитку

2 ДОСЛІДЖЕННЯ ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

2.1 Мова програмування JavaScript

JavaScript – це ключова та динамічна мова програмування, що є невід’ємною частиною сучасного інтернету. Її історія почалася у 1995 році, коли Брендан Аїк в компанії Netscape створив її як інструмент для «оживлення» статичних HTML сторінок. З того часу JavaScript еволюціонувала з простої скриптової мови до потужної багатогранної технології. Стандартизація мови під назвою ECMAScript, що здійснюється ECMA International, забезпечує її послідовний розвиток та сумісність між різними браузерами та середовищами виконання.

Сучасний JavaScript пропонує розробникам широкий спектр інструментів для створення складних веб-додатків. Коли JavaScript працює у браузері користувача (на стороні клієнта), він дає змогу не лише змінювати вигляд сторінки та реагувати на дії користувача, але й виконувати складніші завдання. Це включає обмін даними з сервером без перезавантаження сторінки та збереження інформації безпосередньо у браузері[2].

Завдяки середовищам виконання, таким як Node.js, JavaScript вийшов за межі браузера і став повноцінною мовою для серверної розробки, дозволяючи створювати швидкі та масштабовані серверні додатки, API, працювати з файловою системою та базами даних. Екосистема JavaScript надзвичайно багата: тисячі бібліотек та фреймворків роблять розробку значно простішою та швидшою. У цьому проєкті JavaScript є головною технологією. Вона відповідає за те, щоб сайт був інтерактивним для користувача, а також за обробку запитів та взаємодію з іншими сервісами на сервері. Використання нових версій JavaScript допомагає писати код, який легше підтримувати та який працює ефективніше.

2.2 Середовище виконання Node.js

Node.js був вперше представлений Райаном Далом у 2009 році. Основною метою створення цього середовища було надання можливості виконувати JavaScript код поза межами веб-браузера, зокрема на сервері. Цей підхід кардинально відрізнявся від традиційних багатопотокових моделей, які використовувалися багатьма серверними технологіями того часу.

Node.js – це потужна програмна платформа, яка кардинально змінила роль JavaScript, перетворивши її з мови, орієнтованої переважно на браузер, на універсальну мову програмування загального призначення. Ключова перевага Node.js полягає в тому, що вона дозволяє розробникам використовувати JavaScript для написання серверного коду (backend), що раніше було прерогативою таких мов, як Java, Python або PHP. В основі Node.js лежить високопродуктивний рушій V8 JavaScript, розроблений Google для браузера Chrome[3].

Node.js працює за принципом обробки подій і не чекає, поки завершиться одна операція, щоб почати іншу (це називається неблокуючою моделлю). Це означає, що Node.js може дуже швидко обробляти багато запитів одночасно, не «зависаючи» в очікуванні, наприклад, відповіді від бази даних чи завершення читання файлу. Завдяки цьому Node.js чудово підходить для створення швидких веб-сайтів, API (інтерфейсів для взаємодії програм) та систем, що працюють у реальному часі.

У цьому проєкті Node.js використовується як основа для роботи фреймворку Remix на сервері. Тобто, саме Node.js приймає запити від користувачів, виконує логіку сайту (наприклад, завантажує дані про товари або обробляє додавання товару в кошик) та спілкується з іншими сервісами, зокрема з Shopify, щоб отримати інформацію про товари та керувати замовленнями.

2.3 Бібліотека React.js

React.js – це популярна JavaScript бібліотека для створення користувацьких інтерфейсів, розроблена компанією Facebook. Основна ідея React полягає у компонентному підході до розробки, де інтерфейс розбивається на незалежні, перевикористовувані частини – компоненти. React використовує концепцію віртуального DOM (Document Object Model), що дозволяє оптимізувати процес оновлення веб-сторінки, підвищуючи продуктивність. У проєкті фреймворк Remix використовує React для рендерингу всіх візуальних елементів сайту, таких як навігаційні панелі, картки товарів, сторінки категорій та окремих товарів, забезпечуючи динамічність та інтерактивність користувацького досвіду[4].

2.4 Фреймворк Remix

Це сучасний веб-фреймворк, який допомагає створювати веб-сайти та веб-додатки. Він працює «поверх» React.js і дозволяє розробляти як клієнтську, так і серверну частину сайту. Головна мета Remix – робити сайти швидкими, надійними та легкими в обслуговуванні. Він робить акцент на використанні стандартних веб-технологій, що робить його більш стабільним та передбачуваним. Важливою особливістю є серверний рендеринг (SSR), коли HTML сторінка генерується на сервері, а не в браузері, що прискорює перше завантаження сайту. Також Remix підтримує «поступове завантаження» – сайт буде працювати навіть з вимкненим JavaScript, а потім, коли JavaScript завантажиться, додасться більше інтерактивності. Remix значно спрощує роботу з даними: отримання інформації з сервера та її зміну завдяки спеціальним функціям loader та action. Ці функції тісно пов'язані з адресами сторінок. У нашому проєкті, Remix відповідає за загальну структуру сайту, переходи між сторінками, швидке завантаження даних про товари з платформи Shopify та обробку дій користувачів, наприклад, коли

вони додають товар до кошика[5].

2.5 CMS платформа Shopify

Shopify – це одна з найпопулярніших платформ для створення інтернет-магазинів та управління ними. Якщо розробники хочуть створити унікальний дизайн та повністю контролювати те, як виглядає і працює їхній магазин, Shopify дає таку можливість. Це називається підходом з відокремленою вітриною (часто використовують англійський термін «headless commerce»). Суть у тому, що Shopify використовується як надійна бекенд частина, де зберігаються товари, обробляються замовлення, ведеться облік клієнтів та інша важлива інформація для бізнесу. А фронтенд – те, що бачать покупці – створюється окремо, за допомогою інших інструментів. У цьому проєкті для створення такої індивідуальної вітрини магазину використовується фреймворк Remix. Він дозволяє розробити гнучкий, швидкий та зручний для користувачів інтерфейс, який «спілкується» з Shopify через API для обміну даними. Такий підхід дає повну свободу у створенні дизайну та функціоналу сайту[6].

2.6 Мова запитів GraphQL API

API – це набір правил, який дозволяє різним програмам обмінюватися даними. GraphQL є мовою запитів, спеціально створеною для таких API, а також серверним середовищем для їх обробки.

Ключова особливість GraphQL полягає в тому, що клієнт (наприклад, веб-сайт або мобільний додаток) може точно вказати структуру даних, яку він очікує отримати. Це дозволяє уникнути проблеми надлишкового отримання даних (over-fetching), коли сервер повертає більше інформації, ніж потрібно клієнту, або недостатнього отримання (under-fetching), коли для отримання всіх необхідних даних доводиться робити декілька запитів. Таким

чином, GraphQL оптимізує обмін даними, роблячи його більш ефективним.

У даному проєкті GraphQL використовується для комунікації між фронтенд-додатком, створеним на Remix, та API платформи Shopify. Це дозволяє клієнтській частині сайту ефективно запитувати лише необхідну інформацію про товари, їх категорії, варіанти, ціни та інші супутні дані для коректного відображення користувачам[7].

2.7 CSS фреймворк Tailwind CSS

Tailwind CSS був вперше представлений Адамом Уетаном у 2017 році. Ідея полягала у відході від традиційних CSS фреймворків, які надавали готові компоненти, до підходу «utility first». Це означає, що замість використання попередньо стилізованих компонентів, розробники отримують великий набір маленьких, одноцільових CSS класів, які вони можуть комбінувати для створення будь-якого дизайну

Це дозволяє створювати унікальні та складні дизайни, не витрачаючи час на вигадкування назв для CSS класів і постійне перемикання між HTML та CSS файлами. Як результат, процес розробки значно прискорюється, а підтримка стилів стає простішою. Tailwind CSS відповідає за стилізацію всіх візуальних елементів сайту, забезпечуючи їх узгоджений вигляд та адаптивність на різних пристроях, від мобільних телефонів до настільних комп'ютерів

2.8 Середовище розробки Visual Studio Code

Visual Studio Code (VS Code) є передовим редактором коду, розробленим Microsoft, який здобув широке визнання у спільноті розробників. Його успіх зумовлений поєднанням простоти текстового редактора з потужними функціями, характерними для повноцінних інтегрованих середовищ розробки.

Однією з найважливіших переваг VS Code є його гнучкість та можливість розширення функціоналу. Величезний маркетплейс розширень дозволяє адаптувати середовище під специфічні потреби будь-якого проєкту. Розробники можуть встановлювати плагіни для підтримки різноманітних мов програмування, фреймворків, інструментів для форматування коду, лінтерів та інтеграції з системами контролю версій, такими як Git, що вбудований безпосередньо в редактор. Наявність інтегрованого терміналу дозволяє виконувати командні операції, не покидаючи середовища розробки.

VS Code надає потужні інструменти для відлагодження коду, включаючи можливість встановлення точок зупину, покрокового виконання та аналізу значень змінних. Редактор також добре інтегрується з хмарними платформами та сервісами, а функція Remote Development дозволяє комфортно працювати з проєктами, розміщеними на віддалених серверах або в контейнерах. Завдяки інтуїтивному інтерфейсу, можливостям кастомізації та регулярним оновленням, VS Code став незамінним інструментом для багатьох розробників.

3 РОЗРОБКА АЛГОРИТМІЧНОГО ЗАБЕСПЕЧЕННЯ

3.1 Пошук та фільтрація товарів

Обробка запитів включає декілька послідовних етапів: прийом пошукового запиту від користувача, попередню обробку та нормалізацію тексту, формування оптимізованого GraphQL запиту до Shopify API, виконання пошуку в базі даних товарів, постобробку результатів з урахуванням фільтрації та повернення відсортованих результатів користувачу.

3.1.1 Алгоритм текстового пошуку товарів

Алгоритм текстового пошуку товарів являє собою процес обробки користувацьких запитів для товарів в каталозі. Розробка алгоритму базувалася на балансуванні між повнотою результатів та продуктивністю системи з урахуванням особливостей інтеграції з GraphQL API Shopify. Ініціалізація пошукового процесу відбувається через прийом запиту від користувача, які передаються через URL структуру веб-додатку. Такий підхід забезпечує можливість збереження пошукових запитів у браузерні закладки та спрощує обмін результатами пошуку між користувачами.

Нормалізація текстового запиту включає видалення зайвих пробілів, приведення символів до нижнього регістру та фільтрацію спеціальних символів. Ці операції гарантують консистентність роботи пошукового механізму та знижують навантаження на сервер через усунення дублювання схожих запитів з незначними варіаціями форматування.

Встановлено мінімальну довжину запиту у два символи для уникнення надмірно широких результатів. GraphQL API Shopify має вбудований захист від спроб SQL ін'єкцій.

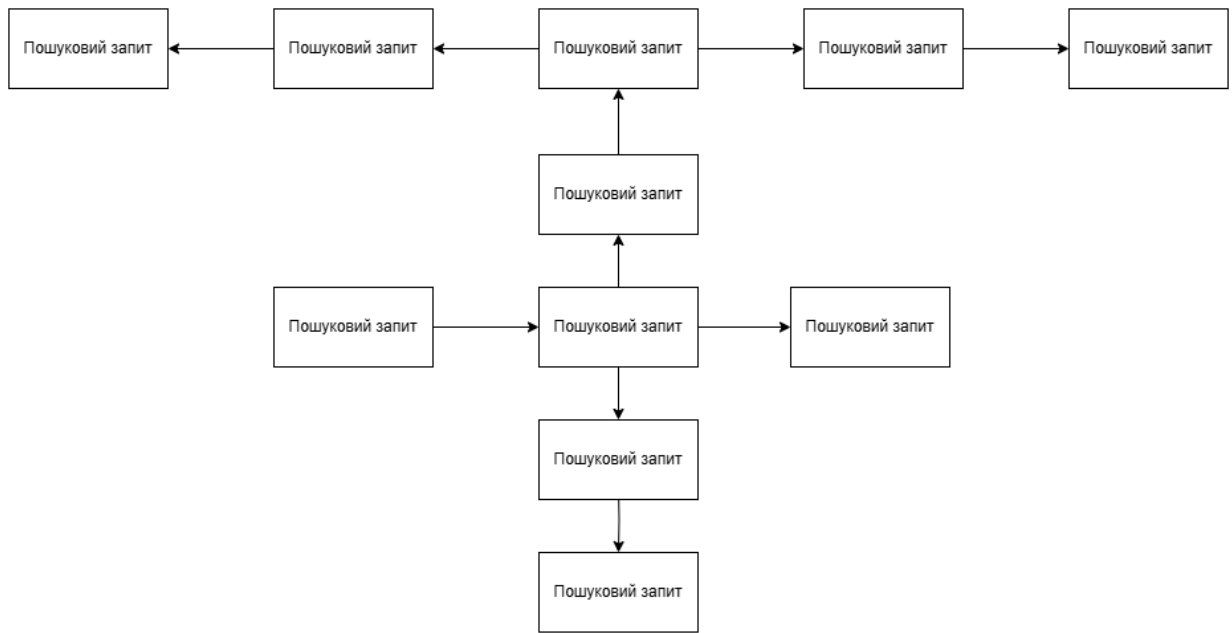


Рисунок 3.1 – Схема текстового пошуку

3.1.2 Фільтрація за категоріями та параметрами

Архітектура системи фільтрації базується на принципі створенні критеріїв, що забезпечує гнучкість у формуванні запитів різної складності – від елементарного вибору однієї категорії до вибору декількох параметрів. Така модульність дозволяє системі адаптуватися до різноманітних сценаріїв використання, забезпечуючи як простоту так і ефективність

Ініціалізація процесу фільтрації відбувається через аналіз параметрів URL адреси, які надходять до системи через маршрут каталогу товарів. Система підтримує набір параметрів фільтрації: `category` для визначення товарних категорій, `purpose` для специфікації призначення товару. Використання URL параметрів забезпечує не лише зручність навігації, але й можливість збереження та відтворення конкретних наборів фільтрів.

Формування GraphQL запиту є ключовим етапом фільтрації, де система створює запит на основі активних фільтрів. Для кожного типу фільтра генерується спеціалізована частина запиту: `collection queries` використовуються для фільтрації за категоріями товарів, `tag queries` застосовуються для відбору за призначенням. Ці компоненти інтегруються в

єдиний оптимізований запит, що мінімізує кількість звернень до API Shopify та забезпечує ефективне використання інтернету, це дуже ефективно, якщо у користувача повільний інтернет.

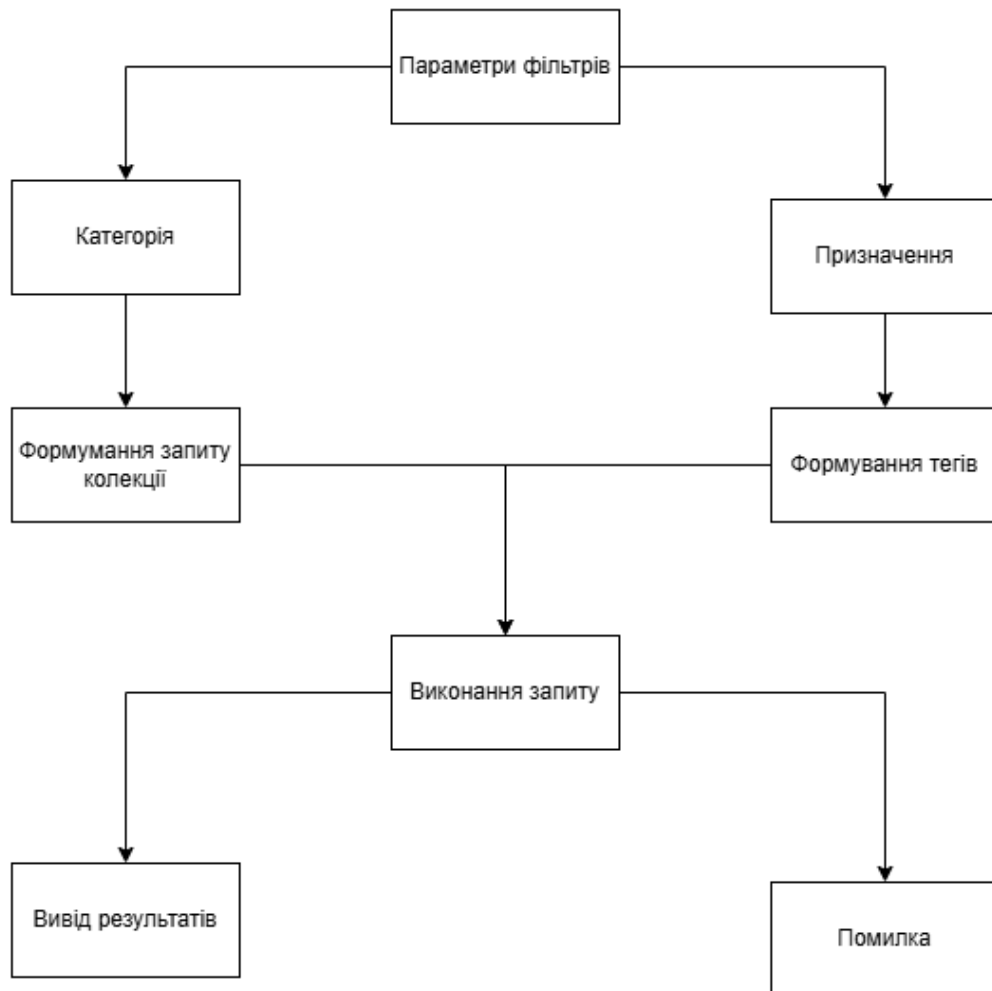


Рисунок 3.2 – Схема алгоритму фільтрації

Система включає надійний механізм обробки виняткових ситуацій, який гарантує стабільність роботи навіть при частковій недоступності окремих компонентів фільтрації. У випадку виникнення помилки при обробці конкретного фільтра, наприклад, через тимчасову недоступність певної категорії в базі даних, система автоматично виключає проблемний критерій та продовжує обробку решти параметрів. Такий підхід забезпечує максимальну доступність функціональності та запобігає повній відмові системи через локальні проблеми.

3.1.3 Алгоритм сортування результатів

Система підтримує як прості критерії сортування (ціна, дата додавання, популярність) Основна логіка алгоритму сортування реалізована через систему пріоритетів, яка враховує різні критерії товарів залежно від обраного користувачем режиму сортування. Система підтримує чотири основні типи сортування: за популярністю (best-selling), за ціною у порядку зростання (price-asc), за ціною у порядку спадання (price-desc) та за новизною (created-desc). Кожен тип сортування використовує відповідні параметри Shopify API для оптимальної продуктивності та точності результатів.

Сортування за популярністю базується на вбудованих метриках Shopify, що відслідковують фактичні продажі товарів. Цінове сортування включає логіку яка використовує мінімальну ціну серед доступних варіантів для сортування за зростанням та спадання. Це забезпечує коректне відображення товарів у ціновому діапазоні незалежно від кількості та різноманітності варіантів. Сортування за новизною пріоритизує нещодавно додані товари яка враховує оновлення існуючих товарів, такі як додавання нових варіантів кольорів або зміни в описі.

Для оптимізації продуктивності сортування реалізовано гібридний підхід: первинна фільтрація та базове сортування виконуються на рівні GraphQL запиту до Shopify API, а додаткове ранжирування за специфічними критеріями проводиться на клієнтській стороні за допомогою оптимізованих JavaScript алгоритмів.

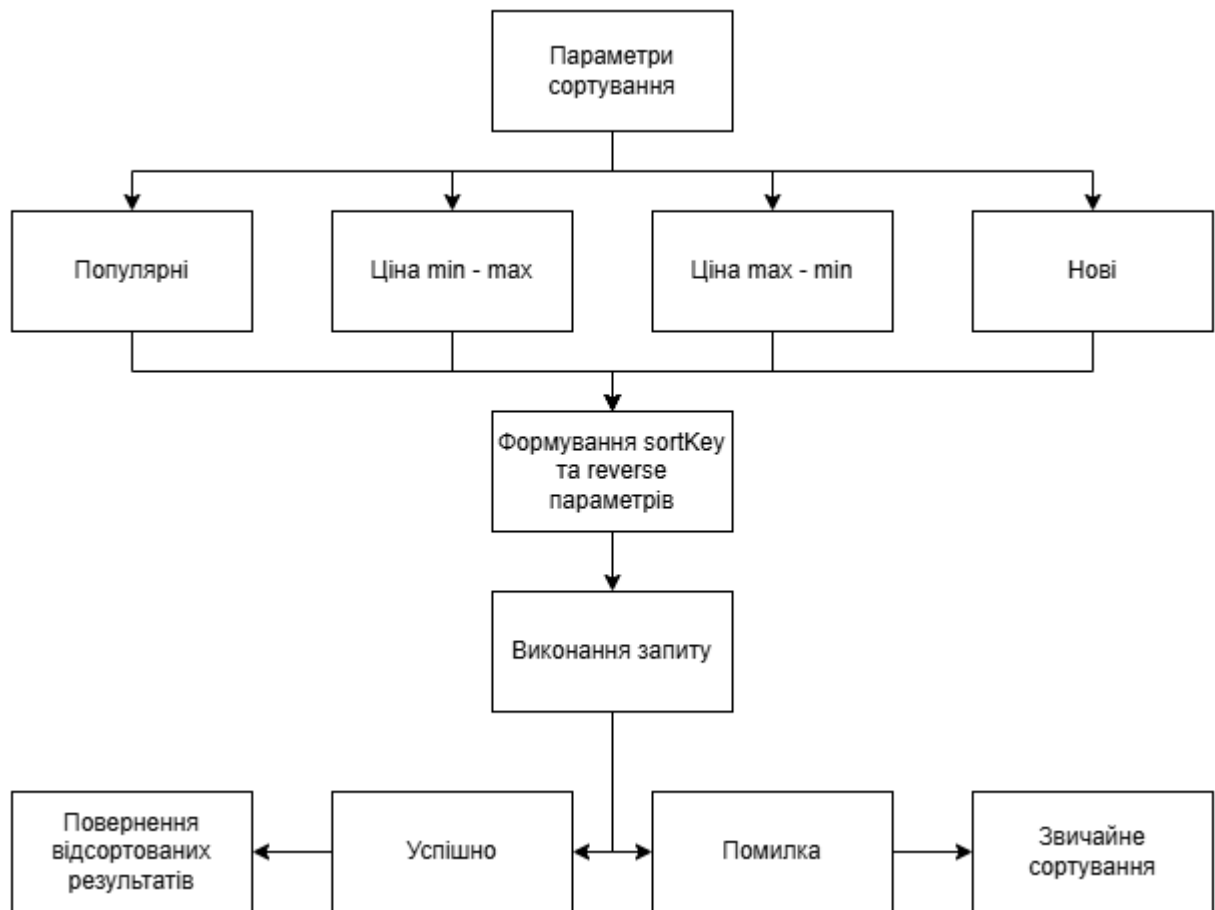


Рисунок 3.3 – Схема сортування

3.2 Управління кошиком та станом кошика

Система управління кошиком у веб-додатку представляє собою систему синхронізації стану між клієнтською частиною додатку та віддаленим Shopify Cart API. Основна складність полягає у забезпеченні консистентності даних при асинхронних операціях, мінімізації затримок в інтерфейсі користувача та надійній обробці помилок мережі.

3.2.1 Додавання товарів до кошика

Процес починається з валідації варіанту товару, що додається до кошика. Система перевіряє існування варіанту, його доступність для продажу, коректність кількості та відповідність обраних опцій (розмір, колір тощо) доступним комбінаціям. Після успішної валідації починається етап

оптимістичного оновлення інтерфейсу. Система створює тимчасовий запис товару з унікальним ідентифікатором та миттєво додає його до локального стану кошика. Це дозволяє користувачу негайно побачити результат своєї дії, не очікуючи відповіді сервера. Паралельно з оновленням UI запускається процес синхронізації з Shopify Cart API. Система перевіряє наявність активного кошика користувача. Якщо кошик не існує, створюється новий через відповідний API endpoint. Якщо кошик вже існує, товар додається до існуючого кошика.

Система також включає логіку обробки дублікатів – якщо товар з ідентичними параметрами вже існує в кошику, замість створення нового рядка збільшується кількість існуючого. Алгоритм включає механізм відкату змін у випадку помилок, що забезпечує коректність відображення стану кошика.

3.2.2 Оновлення кількості та видалення товарів

Система ефективно обробляє як поступові зміни кількості (збільшення/зменшення на одиницю), так і радикальні зміни (повне видалення товару).

Процес оновлення починається з валідації нової кількості товару. Система перевіряє, що кількість є додатним числом, не перевищує максимально дозволена кількість для товару та відповідає наявним залишкам на складі. Особливим випадком є обробка нульової кількості, що інтерпретується як запит на видалення товару з кошика. Аналогічно до алгоритму додавання, система використовує оптимістичні оновлення для забезпечення миттєвого відгуку інтерфейсу[9]. Зміни кількості негайно відображаються в локальному стані, а паралельно виконується синхронізація з сервером. Це створює враження швидкої та відгукливої системи навіть при повільному інтернет з'єднанні.

3.3 Обробка замовлень та доставки

Система замовлень побудована на основі багатоетапної валідації. Замість монолітної форми з усіма полями, користувач проходить через декілька кроків, який валідується незалежно, така система дає надійду структуру з можливістю модифікації без появи конфліктів. Центральною компонентою системи є *state machine*, що керує переходами між етапами оформлення замовлення. Кожен стан має свої правила валідації, доступні дії та умови переходу до наступного стану.

3.3.1 Ведення даних користувача

Валідація даних організована за принципом поступової перевірки. Контактні дані включають перевірку імені та прізвища на використання допустимих символів. Перевірку email на правильність написання, також перевірка введення номеру телефону, щоб користувач не міг вказати номер телефону іншої країни, також приступна валідація за довжиною номеру телефону. Система підтримує автоматичне форматування та нормалізацію номерів: перетворення номерів, що починаються з «0» на міжнародний формат «+380». Валідація адреси доставки інтегрована з API Nova Poshta та включає перевірку існування обраного міста та відділення в реальному часі[10]. Система не дозволяє користувачу продовжити оформлення замовлення з недійсною адресою доставки, що запобігає проблемам з доставкою на пізніших етапах.

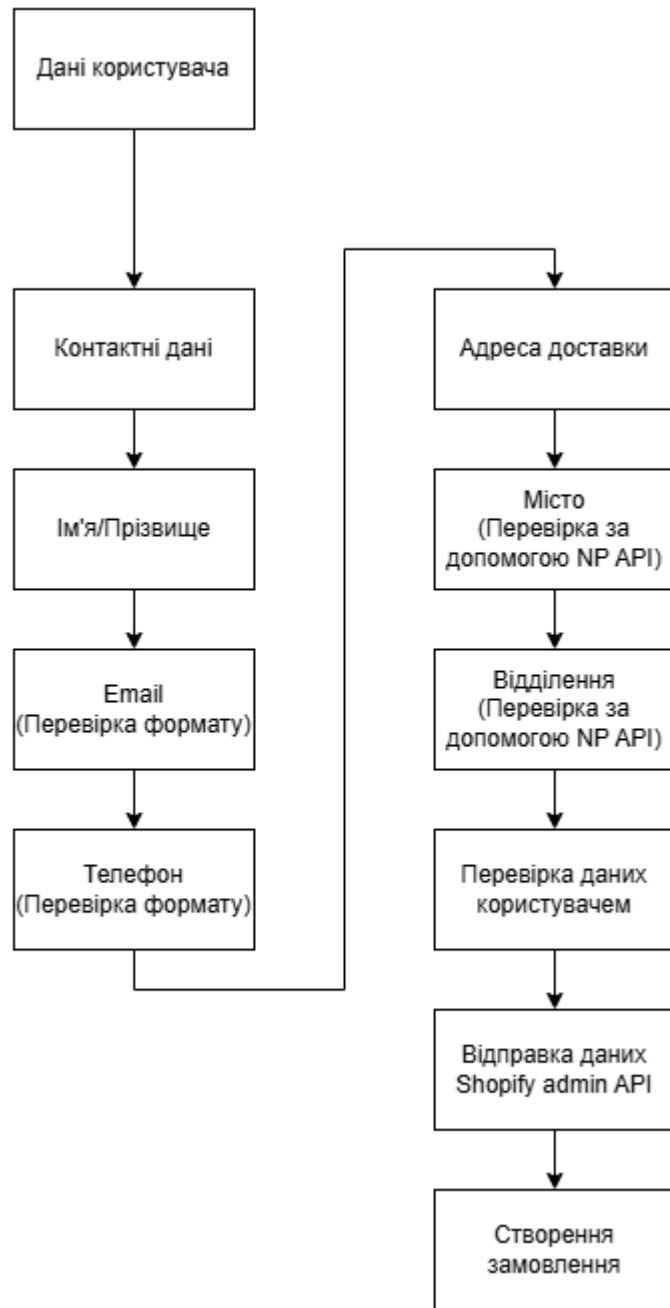


Рисунок 3.4 – Схема оформлення замовлення та валідації даних при оформленні

4 РОЗРОБКА ТА ВІДЛАГОДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура додатку побудована на основі модульності та розділення відповідальності. Кожен функціональний блок (навігація, пошук, каталог, кошик, оформлення замовлення) реалізований як незалежний модуль з власною логікою управління станом та API взаємодії. Така структура дозволяє легко масштабувати додаток, додавати нові функції та проводити перебудову функціоналу без впливу на інші частини системи. Процес розробки включав тісну співпрацю між frontend та backend компонентами, хоча основна логіка додатку виконується на клієнтській стороні з підтримкою серверного рендерингу. Тестування та відлагодження здійснювалося на всіх етапах розробки. Особлива увага приділялася кросбраузерній сумісності, адаптивності дизайну та продуктивності на різних типах пристроїв.

4.1 Реалізація системи навігації та маршрутизації

Система навігації та маршрутизації є основним компонентом веб-сайту, він повинен бути інтуїтивно зрозумілим, для зручного переміщення користувачів між різними розділами сайту. Розробка навігаційної системи базувалася на принципах файлової маршрутизації, що надається фреймворком Remix.js.

Архітектура маршрутизації включає структуру основних розділів сайту, де кожен маршрут відповідає конкретному функціональному модулю системи. Головна сторінка служить точкою входу та презентує користувачам ключові категорії товарів і популярні пропозиції. Каталог товарів реалізований через динамічний маршрут, який підтримує різноманітні параметри фільтрації та сортування через URL параметри, забезпечуючи можливість збереження та відтворення конкретних наборів фільтрів.

Адаптивність навігаційної системи досягається через використання Tailwind CSS [8] медіа запитів та JavaScript логіки для динамічної трансформації інтерфейсу залежно від характеристик пристрою користувача. На екранах настільних комп'ютерів навігація представлена у вигляді горизонтального меню з чітко структурованими категоріями товарів. При переході на мобільні пристрої навігаційна структура трансформується в компактне вертикальне меню, яке максимально ефективно використовує обмежений простір екрану.

Управління станом навігаційної системи реалізовано через набір React hooks, які забезпечують ефективне оновлення компонентів при зміні активних розділів, стану пошукової форми або кількості товарів у кошику. Синхронізація стану між різними частинами навігації досягається через контекст React, що забезпечує постійність відображення активних елементів незалежно від поточного маршруту.

Кожна категорія у панелі навігації має унікальне посилання що дозволяє користувачам легко ділитися посиланнями на конкретні розділи каталогу та повертатися до збережених пошукових результатів.

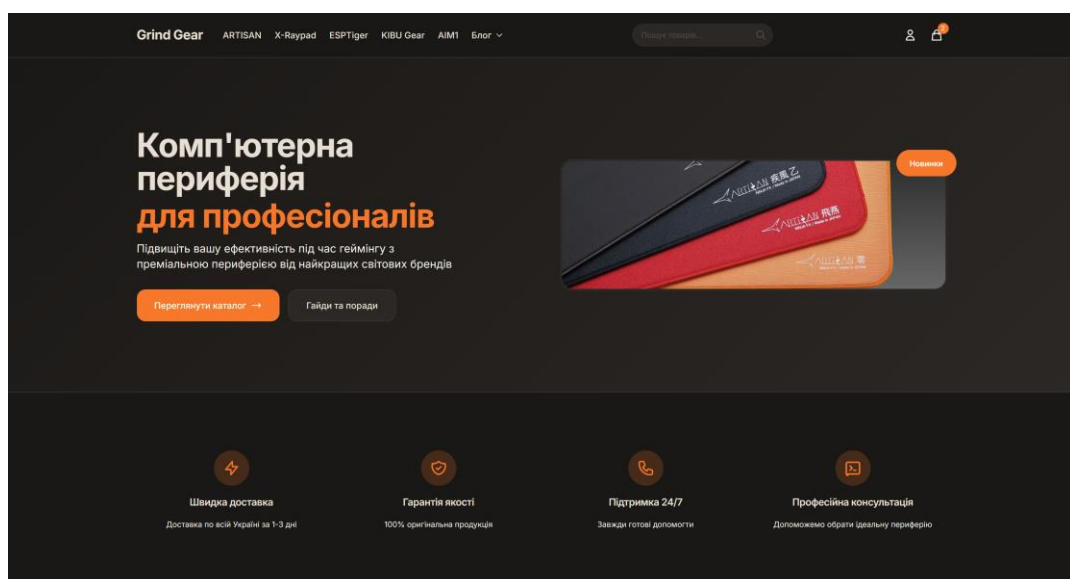


Рисунок 4.1 – Адаптивна навігація при комп'ютерному перегляді

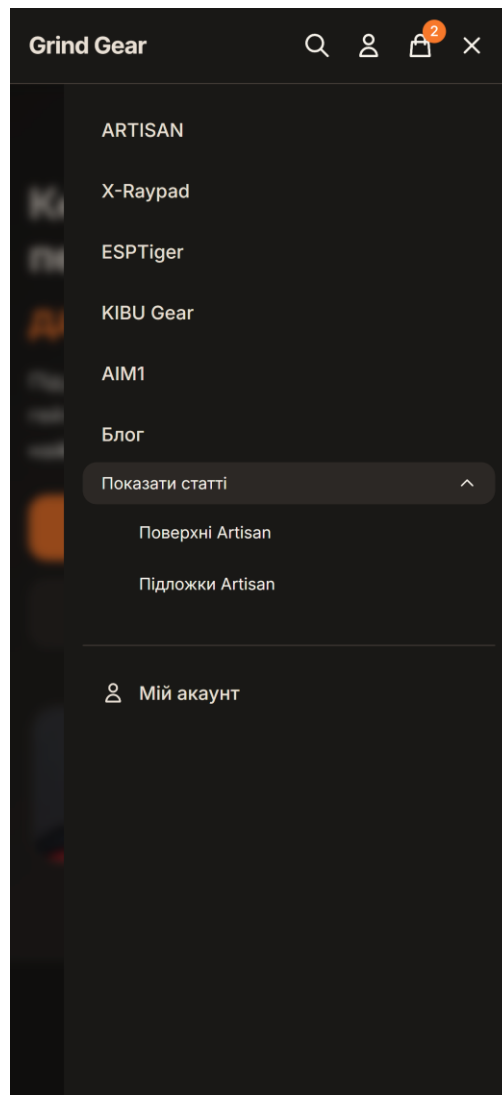


Рисунок 4.2 – Адаптивна навігація при мобільному перегляді

4.2 Розробка системи пошуку товарів

Розробка пошукової функціональності вимагала підходу, який би забезпечував не лише високу релевантність результатів, але й швидкість відгуку системи та зручність користування. Архітектура пошукової системи базується на інтеграції з Shopify Storefront API через використання спеціалізованих GraphQL запитів, що дозволяє ефективно отримувати та фільтрувати товари згідно з критеріями пошуку користувача.

Ключовим алгоритмічним рішенням стала реалізація пошуку, який одночасно аналізує декілька атрибутів товару для забезпечення максимального охоплення релевантних результатів. Система формує

пошуковий запит, використовуючи логічний оператор OR для об'єднання пошуку за різними полями: назвою товару, типом продукту, виробником та тегами. Такий підхід гарантує, що користувач знайде потрібний товар незалежно від того, як саме він сформулював запит – чи то за назвою конкретної моделі, брендом виробника або категорією товару. Структура GraphQL запиту оптимізована для отримання лише необхідної інформації, включаючи унікальний ідентифікатор товару, назву, handle для формування URL адрес, інформацію про виробника та категорію продукту.

Для покращення користувацького досвіду було впроваджено систему інтелектуальних підказок, яка працює в режимі реального часу під час введення пошукового запиту. Реалізація цієї функціональності потребувала створення окремого API endpoint, оптимізованого для швидкої обробки часткових запитів та повернення релевантних пропозицій. Система підказок аналізує введений текст та формує список можливих варіантів на основі наявних товарів, популярних запитів та категорій товарів. Механізм debouncing забезпечує оптимальний баланс між швидкістю реакції та навантаженням на сервер, затримуючи виконання запиту до моменту, коли користувач припинить введення на визначений проміжок часу[11].

Інтеграція пошукової функціональності в загальну архітектуру сайту реалізована через вбудовування пошукової форми безпосередньо в навігаційну панель, що забезпечує постійну доступність функції пошуку незалежно від поточної сторінки. На мобільних пристроях пошукова форма адаптивно трансформується, розгортаючись на всю ширину екрану при активації, що максимізує зручність введення тексту на сенсорних екранах. Візуальний дизайн пошукового інтерфейсу слідує принципам мінімалізму та функціональності.

Оптимізація продуктивності пошукової системи включала ретельне планування структури даних та мінімізацію обсягу інформації, що передається. Особливу увагу було приділено обробці зображень товарів у результатах пошуку – система запитує лише перше зображення кожного

товару, що достатньо для візуальної ідентифікації продукту та зменшує об'єм передаваних даних.

Система перевіряє мінімальну довжину запиту перед виконанням пошуку, запобігаючи надмірному навантаженню від занадто коротких та неспецифічних запитів.

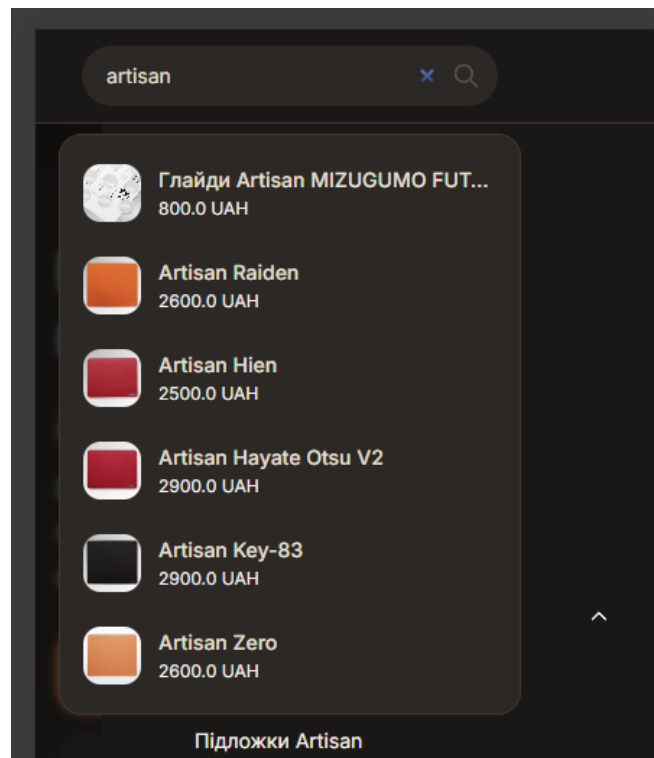


Рисунок 4.2 – Сторінка пошуку

4.3 Каталог товарів фільтрація та сортування

Система каталогу товарів із фільтрацією забезпечує ефективний перегляд, фільтрування та сортування великих обсягів товарних даних. Розробка каталогової системи потребувала створення гнучкої архітектури, здатної адаптуватися до різноманітних користувацьких сценаріїв відповідно до headless commerce [12].

Для сортування за брендом система використовує фільтрування за колекціями через GraphQL запити до Shopify API. Таким чином використовуються вбудовані можливості колекцій Shopify для

ефективнішого фільтрування та сортування товарів. Базові опції сортування включають відображення найпопулярніших товарів (best-selling) на основі метрик продажів Shopify, цінове сортування за зростанням і спаданням (price-asc та price-desc) з використанням значень minVariantPrice, а також демонстрацію найновіших надходжень (created-desc).

Систему управління станом фільтрів реалізовано через URL орієнтовану архітектуру з використанням хука useSearchParams із Remix.js. Це забезпечує синхронізацію між станом компонентів та навігацією браузера. Кожна зміна фільтрів автоматично оновлює URL через функцію setSearchParams із додатковими оптимізаціями: автоматичне очищення параметрів пагінації при зміні фільтрів та використання опції preventScrollReset для поліпшення користувацького досвіду. Логіка функції handleFilterChange включає валідацію нових значень фільтрів, логіку об'єднання для багатовибіркових фільтрів та автоматичне очищення порожніх або некоректних значень. Це забезпечує чистоту URL та коректність стану системи.

Візуальне відображення активних фільтрів реалізовано через інтерактивну систему мікс-фішок (filter chips), що дозволяє користувачам швидко переглядати поточні налаштування фільтрування та видаляти окремі фільтри одним кліком. Кожен активний фільтр відображається як стилізована фішка з назвою фільтра та кнопкою видалення, що запускає відповідну логіку видалення з автоматичним оновленням URL та перезавантаженням каталогу.

Для забезпечення відгуку системи при зміні фільтрів використовується техніка debouncing, яка затримує виконання запитів на 300 мілісекунд після останньої взаємодії користувача, що мінімізує кількість API викликів та покращує продуктивність. Візуальна частина каталогу адаптується під різні розміри екранів за допомогою responsive grid системи Tailwind CSS, забезпечуючи оптимальне відображення.

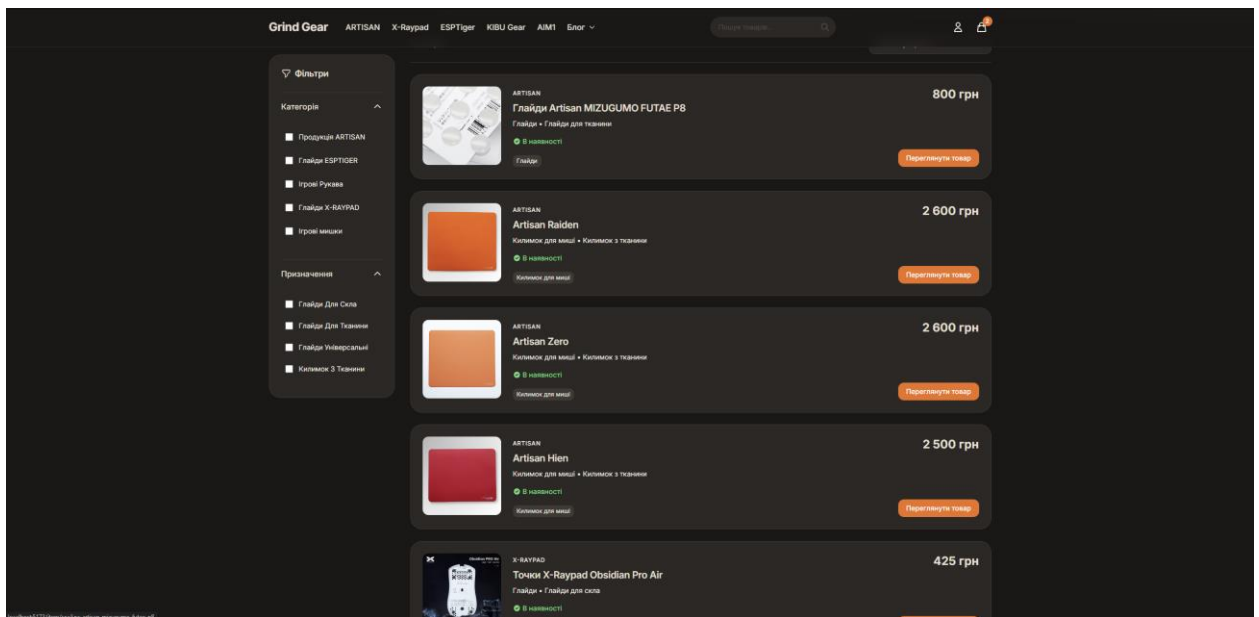


Рисунок 4.3 – Каталог товарів

4.4 Розробка сторінки детального перегляду товару

Сторінка детального перегляду товару є критично важливим компонентом інтернет-магазину, оскільки саме на ній користувач приймає остаточне рішення про покупку. Центральним елементом сторінки є галерея зображень товару, реалізована у вигляді інтерактивної карусельної галереї. Галерея підтримує декілька режимів перегляду: стандартний режим з мініатюрами для швидкої навігації між зображеннями та повноекранний режим для детального розгляду товару. Особливу увагу приділено оптимізації завантаження зображень – система використовує техніку lazy loading та адаптивні розміри зображень залежно від пристрою користувача.

При виборі користувачем певного параметру (наприклад, розміру або кольору), система автоматично оновлює доступність інших опцій, відображаючи лише ті комбінації, які є в наявності. Це реалізовано через алгоритм перевірки наявності кожної можливої комбінації варіантів у режимі реального часу. Недоступні варіанти візуально позначаються як неактивні, але залишаються видимими для інформування користувача про повний асортимент. При зміні варіанту автоматично оновлюється ціна, наявність та,

якщо доступно, специфічне зображення для обраного варіанту.

Функціонал додавання товару до кошика включає розширені можливості контролю кількості з валідацією доступних залишків. Система використовує оптимістичні оновлення для миттєвого відгуку на дії користувача, при цьому забезпечуючи надійну синхронізацію з серверною частиною. Реалізовано візуальну систему сповіщень, яка інформує користувача про успішне додавання товару до кошика або про виникнення помилок. Повідомлення автоматично зникають через визначений час, не відволікаючи користувача від подальшого перегляду.

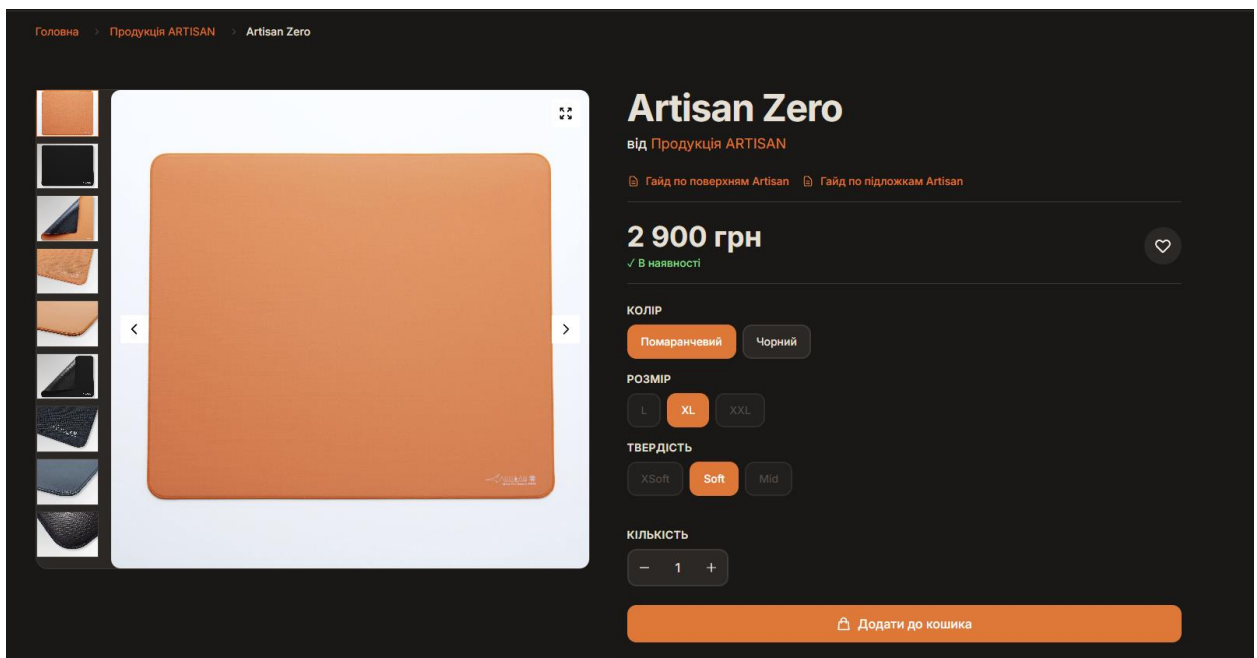


Рисунок 4.4 – Сторінка продукту

4.5 Система оформлення замовлень

Головним рішенням стало розділення функціональності на серверну та клієнтську частини через модульну структуру файлів. Клієнтська частина відповідає за інтерактивність користувацького інтерфейсу та візуальні компоненти, тоді як серверна логіка обробляє валідацію даних, взаємодію з API та безпосереднє створення замовлень у системі Shopify.

Замість традиційного підходу з монолітною формою, система реалізує чотирьох етапний процес оформлення замовлення, який розділяє збір інформації на логічні блоки: контактні дані, інформація про доставку, вибір способу оплати та фінальне підтвердження. Центральним елементом архітектури є state machine, який керує переходами між етапами, визначає доступні дії на кожному кроці та контролює умови переходу до наступного етапу. Така архітектура забезпечує надійну структуру з можливістю модифікації окремих етапів без впливу на загальну логіку системи.

Збір та валідація контактних даних реалізовані через систему перевірок, яка забезпечує коректність введеної інформації на різних рівнях. Валідація імені та прізвища включає перевірку мінімальної довжини та використання лише допустимих символів, що запобігає введенню некоректних даних. Особливу увагу приділено обробці телефонних номерів – система автоматично розпізнає та форматує українські номери, що можуть бути введені в різних форматах. Номери, що починаються з «0», автоматично перетворюються на міжнародний формат «+380», при цьому візуальне форматування з пробілами забезпечує легке сприйняття номера користувачем.

Інтеграція з сервісом доставки «Нова Пошта» реалізована через спеціалізовану серверну бібліотеку, яка дозволяє працювати з API перевізника. Пошук відділень підтримує множинні формати запитів – користувач може вводити як повний номер відділення у форматі «Відділення №1», так і просто число «1», або використовувати ключові слова типу «поштомат». Складні регулярні вирази аналізують введений текст та автоматично визначають тип пошукового запиту, забезпечуючи релевантні результати.

Фінальний етап оформлення замовлення включає перевірку всіх введених даних перед відправкою на сервер. Система формує структурований об'єкт замовлення, який включає інформацію про товари з кошика, контактні дані клієнта, обрану адресу доставки та спосіб оплати.

Перед створенням замовлення в системі Shopify виконується додаткова серверна валідація для забезпечення цілісності даних та запобігання можливим помилкам. Після успішного створення замовлення користувач отримує підтвердження з унікальним номером замовлення та детальною інформацією про подальші кроки. Система також автоматично очищає кошик та перенаправляє користувача на сторінку підтвердження, де відображається вся інформація про оформлене замовлення.

The image shows two side-by-side panels from a mobile application. The left panel, titled 'Контактні дані' (Contact information), asks for contact details for communication and delivery. It includes fields for 'Ім'я *' (Name), 'Прізвище *' (Surname), 'Email *' (Email), and 'Телефон *' (Phone). Below these fields is a note: 'Ми надішлемо підтвердження замовлення на цю адресу' (We will send the order confirmation to this address). At the bottom, there is a section 'Важлива інформація' (Important information) with a note: 'Всі поля позначені * є обов'язковими для заповнення' (All fields marked * are mandatory for completion). The right panel, titled 'Доставка' (Delivery), shows the delivery method as 'У відділення/поштомат' (To branch/postbox) via 'Нова Пошта' (Nova Poshta). It includes a note: 'Доставка оплачується при отриманні' (Delivery is paid upon receipt) and 'Вартість доставки розраховується оператором Нової Пошти та сплачується при отриманні товару' (Delivery cost is calculated by Nova Poshta operator and paid upon receipt of the goods). Below this, there are dropdown menus for 'Місто / Населений пункт *' (City / populated area) and 'Номер відділення або поштоводу *' (Branch or post office number). The selected city is 'м. Львів, Львівська обл.' (Lviv, Lviv region) and the branch is 'Відділення №1: вул. Городоцька, 359' (Branch No. 1: Hrushevska St, 359). At the bottom, there are 'Назад' (Back) and 'Далі >' (Next) buttons.

Рисунок 4.4 – Форми оформлення замовлення

4.6 Результати тестування

Комплексне тестування розробленого веб-сайту проводилося з метою верифікації функціональних вимог та забезпечення стабільної роботи всіх компонентів системи.

Тестування навігаційної системи продемонструвало надійну роботу навігаційної панелі на різноманітних пристроях та розмірах екранів. Особлива увага приділялася перевірці коректності відображення на популярних мобільних пристроях, включаючи iPhone 12 та Samsung Galaxy S21, де навігаційне меню трансформується у вертикальний формат з випаданим списком. Динамічне розширення пошукової форми на мобільних пристроях працює бездоганно, забезпечуючи повне використання

доступного простору екрану при активації пошуку. Автоматичне оновлення індикатора кількості товарів у кошику відбувається миттєво при додаванні нових позицій, що було підтверджено через моніторинг стану компонентів у React Developer Tools та аналіз мережевих запитів у консолі браузера.

Функціональність пошукової системи пройшла ретельну перевірку з різними типами запитів та граничними випадками. Валідація мінімальної довжини запиту працює коректно – при введенні менше двох символів система не відображає запит, через що не здійснюються виконання зайвих запитів до API Shopify, що оптимізує використання ресурсів. Механізм сортування результатів демонструє стабільну роботу з усіма доступними опціями: за релевантністю, ціною в обох напрямках та датою створення товару. Зміна параметрів сортування правильно оновлює URL адресу через механізм `useSearchParams`, забезпечуючи можливість збереження та відтворення результатів пошуку через посилання.

Модуль управління кошиком продемонстрував відмінні результати при тестуванні оптимістичних оновлень інтерфейсу. При додаванні товару система миттєво створює тимчасовий запис з унікальним ідентифікатором, забезпечуючи негайний візуальний відгук для користувача, навіть при повільному інтернет з'єднанні.

Інтеграція з API служби доставки «Нова Пошта» показала стабільні результати при різних сценаріях використання. Пошук населених пунктів коректно обробляє запити, включаючи написання різних міст та сел. Пошук відділень демонструє високу точність при роботі з різними форматами запитів: числові запити коректно фільтрують відділення за номерами, текстові запити знаходять відділення за ключовими словами, а комбіновані запити правильно інтерпретуються системою розпізнавання.

Система валідації форм пройшла всебічне тестування з використанням різноманітних тестових даних та граничних випадків. Валідація телефонних номерів демонструє надійну роботу з усіма поширеними форматами введення українських номерів. Механізм автоматичного форматування

працює в реальному часі, додаючи пробіли для покращення читабельності номера без втручання в процес введення користувача. Валідація електронної пошти використовує вираз, який точно визначає коректність формату адреси та успішно пройшов тестування.

Кросбраузерне тестування підтвердило повну сумісність додатку з усіма сучасними браузерами, включаючи останні версії Chrome, Firefox та Edge. Всі функціональні можливості працюють ідентично незалежно від браузера, з мінімальними візуальними відмінностями в рендерингу CSS стилів, які не впливають на користувацький досвід. Стрес тестування під високим навантаженням включало симуляцію інтенсивних дій користувача: швидке послідовне додавання товарів до кошика, одночасну роботу з множинними вкладками каталогу, інтенсивний пошук з частою зміною запитів. Система продемонструвала стабільну роботу без втрати даних або збоїв у функціональності, підтверджуючи високий рівень надійності реалізованого рішення та його готовність до використання в реальних умовах.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було проведено комплексне дослідження сучасних підходів до розробки веб-додатків для електронної комерції та створено повнофункціональний інтернет-магазин. Робота дозволила поглибити теоретичні знання у сфері веб-технологій та застосувати їх на практиці для вирішення реальних завдань електронної торгівлі. Аналіз існуючих рішень на ринку, включаючи такі популярні платформи як Зона51, Telemart та WKey. Спостереження за цими популярними платформами стали основою для прийняття рішення про використання інноваційної headless архітектури, яка забезпечує розділення презентаційного шару від бізнес логіки та управління даними.

Ключовим технологічним рішенням стало використання фреймворку Remix.js, який представляє новий підхід до створення React додатків з акцентом на продуктивність та оптимізацію. На відміну від традиційних Single Page Applications, Remix забезпечує серверний рендеринг, що суттєво покращує початкову швидкість завантаження сторінок. Інтеграція з Shopify Storefront API через GraphQL забезпечила гнучке та ефективне управління каталогом товарів, при цьому дозволяючи запитувати лише необхідні дані, що мінімізує навантаження на мережу та покращує загальну продуктивність системи.

Значну увагу було приділено розробці алгоритмічного забезпечення для критичних компонентів системи. Реалізація функціоналу кошика включала створення механізму оптимістичних оновлень, який забезпечує миттєвий відгук інтерфейсу на дії користувача з подальшою фоновією синхронізацією з сервером. Це рішення дозволило досягти високого рівня відгуку системи навіть при нестабільному інтернет з'єднанні. Алгоритм обробки замовлень було побудовано на основі багатоетапної валідації з використанням state machine для управління переходами між кроками

оформлення. Інтеграція з API Нової Пошти потребувала розробки системи вибору міст та відділень.

Використання CSS фреймворку Tailwind CSS дозволило значно прискорити процес розробки інтерфейсу користувача. Це рішення також позитивно вплинуло на продуктивність розробки та спростило підтримку стилів проєкті. Особлива увага була приділена забезпеченню повної адаптивності інтерфейсу для різних типів пристроїв – від мобільних телефонів до настільних комп'ютерів з великими екранами.

Результатом виконаної роботи став сучасний веб-сайт електронної комерції, який демонструє високі показники продуктивності, забезпечує зручний користувацький досвід та відповідає актуальним вимогам ринку. Система включає повний цикл функціональності інтернет-магазину: від перегляду каталогу товарів з розширеними можливостями фільтрації та пошуку до оформлення замовлень. Проведене тестування підтвердило стабільну роботу всіх модулів системи та їх відповідність поставленим вимогам. Досвід, отриманий під час розробки даного проєкту, продемонстрував важливість правильного вибору технологічного стеку та архітектурних рішень на початкових етапах проектування. Використання сучасних інструментів та підходів дозволило створити рішення, яке може бути розширене додатковим функціоналом відповідно до бізнес потреб.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Електронна комерція [Електронний ресурс] // Вікіпедія : вільна енциклопедія. URL: https://uk.wikipedia.org/wiki/Електронна_комерція (дата звернення: 23.05.2025).
2. Eloquent JavaScript [Електронний ресурс] / Marijn Haverbeke. URL: <https://eloquentjavascript.net/> (дата звернення: 23.05.2025).
3. Документація Node.js [Електронний ресурс] / Node.js Foundation. URL: <https://nodejs.org/uk/docs/> (дата звернення: 23.05.2025).
4. Документація React [Електронний ресурс] / Meta Platforms, Inc. URL: <https://react.dev/> (дата звернення: 23.05.2025).
5. Документація Remix [Електронний ресурс] / Remix Team. URL: <https://remix.run/docs> (дата звернення: 23.05.2025).
6. Storefront API [Електронний ресурс] / Shopify. URL: <https://shopify.dev/docs/api/storefront> (дата звернення: 23.05.2025).
7. Вступ до GraphQL [Електронний ресурс] / GraphQL Foundation. URL: <https://graphql.org/learn/> (дата звернення: 23.05.2025).
8. Tailwind CSS Documentation [Електронний ресурс] / Tailwind Labs. URL: <https://tailwindcss.com/docs> (дата звернення: 17.06.2025).
9. Concurrent Optimistic Updates in React Query [Електронний ресурс] / Tkdodo's blog. URL: <https://tkdodo.eu/blog/concurrent-optimistic-updates-in-react-query> (дата звернення: 17.06.2025).
10. API Документація Нова Пошта [Електронний ресурс] / ТОВ «Нова Пошта». URL: <https://developers.novaposhta.ua/documentation> (дата звернення: 17.06.2025).
11. Debouncing and Throttling Explained Through Examples [Електронний ресурс] / David Corbacho. URL: <https://css-tricks.com/debouncing-throttling-explained-examples/> (дата звернення: 17.06.2025).
12. What is Headless Commerce? [Електронний ресурс] / BigCommerce.

URL: <https://www.bigcommerce.com/articles/headless-commerce/> (дата звернення: 17.06.2025).

13. Єр'оміна Н. С., Кульков В. Є. Розробка веб сайту інтернет магазину. Scientific Research: Modern Challenges and Future Prospects : матеріали міжнародної науково-практичної конференції (9-11 червня 2025 р.). 2025. С. 297-300.