

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

КВАЛІФІКАЦІЙНА РОБОТА

Тема: «Вебсайт інтернет-магазину з продажу гаджетів до мобільних пристроїв, з використанням багатокритеріальних фільтрів та рекомендаційної системи»

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

ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконав: здобувач 4 курсу  
групи КН-21  
Юрій ОМІЛЯНОВИЧ

Керівник: зав. кафедри комп'ютерних наук,  
к.е.н., с.н.с., доцент,  
Сергій МІЧКІВСЬКИЙ

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

Здобувач: \_\_\_\_\_  
(підпис)

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

ЗАТВЕРДЖУЮ:  
завідувач кафедри  
комп'ютерних наук  
\_\_\_\_\_Сергій МІЧКІВСЬКИЙ  
«\_\_\_\_\_»\_\_\_\_\_20\_\_\_р

ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Омільянович Юрій Сергійович

Тема роботи	Веб-сайт інтернет-магазину з продажу гаджетів до мобільних пристроїв, з використанням багатокритеріальних фільтрів та рекомендаційної системи
Номер та дата наказу про затвердження теми	№121-7 від 24 грудня 2024 року
Коротка постановка завдання	Розробка веб-сайту інтернет-магазину з продажу гаджетів для мобільних пристроїв з багатокритеріальними фільтрами та рекомендаційною системою
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	<p>Федько М.Р. Дослідження методів побудови рекомендаційних систем для вирішення задач в електронній комерції, методи колаборативної фільтрації. // International Electronic Scientific Journal “Science Online”, 2021. URL <a href="https://uk.legacy.reactjs.org/">https://uk.legacy.reactjs.org/</a> (дата звернення: 20.12.2024)</p> <p>Шаповалов С.М., Мічківський С.М. Розробка рекомендаційної системи для підбору співрозмовника в соціальній мережі // XII Міжнародна науково-технічна конференція «Проблеми Інформатизації» (м. Київ, 13 грудня 2018 року). – К.: Державний університет телекомунікацій, 2018. URL: <a href="https://dut.edu.ua/uploads/1_1701_65845854.pdf">https://dut.edu.ua/uploads/1_1701_65845854.pdf</a> (дата звернення: 20.12.2024)</p>
Вимоги до кваліфікаційної роботи	Кваліфікаційна робота має передбачити теоретичне, системотехнічне або експериментальне дослідження складного спеціалізованого завдання або практичної проблеми в галузі комп'ютерних наук, яке характеризується комплексністю та невизначеністю умов і потребує застосування теорій і методів інформаційних технологій.

Дата видачі завдання 27 грудня 2024 р.

Керівник

Сергій МІЧКІВСЬКИЙ

Здобувач освітнього ступеня бакалавра

Юрій ОМІЛЯНОВИЧ

## КАЛЕНДАРНИЙ ПЛАН

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

Керівник

Здобувач освітнього ступеня бакалавра

Сергій МІЧКІВСЬКИЙ

Юрій ОМІЛЯНОВИЧ

*Омілянович Ю.С. Вебсайт інтернет-магазину з продажу гаджетів до мобільних пристроїв, з використанням багатокритеріальних фільтрів та рекомендаційної системи.*

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

Описано розробку веб-сайту інтернет-магазину з продажу гаджетів до мобільних пристроїв, з використанням багатокритеріальних фільтрів та рекомендаційної системи.

Ключові слова: веб-сайт, гаджети, мобільні пристрої, інтернет-магазин, рекомендації.

Рис. 19. Бібліограф.: 17 найм.

*Omilianych Y.S. Website of an online store for the sale of gadgets for mobile devices, using multi-criteria filters and a recommendation system.*

Explanatory note of qualification work in specialty 122 - Computer Science (educational programme - Computer Science), Bachelor's degree - University of Economics and Law "KROK", Educational and Research Institute of Information and Communication Technologies, Department of Computer Science, Kyiv, 2025.

The development of an online store website for the sale of gadgets for mobile devices, using multi-criteria filters and a recommendation system.

Keywords: website, gadgets, mobile devices, online store, recommendations.

Fig. 19. Bibliography: 17 Items.

## ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ВЕБ-САЙТУ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ГАДЖЕТІВ ДО МОБІЛЬНИХ ПРИСТРОЇВ, З ВИКОРИСТАННЯМ БАГАТОКРИТЕРІАЛЬНИХ ФІЛЬТРІВ ТА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ.....	8
1.1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.2 АНАЛІЗ ПОТЕНЦІЙНИХ КОНКУРЕНТНИХ ПЕРЕВАГ ПРОГРАМНОГО ПРОДУКТУ ..	9
1.3 ПОСТАНОВКА ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ.....	13
Висновки до розділу 1 .....	15
РОЗДІЛ 2 ПРОЄКТУВАННЯ ВЕБ-САЙТ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ГАДЖЕТІВ ДО МОБІЛЬНИХ ПРИСТРОЇВ, З ВИКОРИСТАННЯМ БАГАТОКРИТЕРІАЛЬНИХ ФІЛЬТРІВ ТА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ .....	16
2.1 МОДЕЛЮВАННЯ ПОВЕДІНКИ ПРОДУКТУ .....	16
2.2 МОДЕЛЮВАННЯ СТРУКТУРИ ПРОДУКТУ .....	23
2.3 ОПИС АРХІТЕКТУРИ ПРОДУКТУ .....	28
Висновки до розділу 2 .....	32
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБ-САЙТ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ГАДЖЕТІВ ДО МОБІЛЬНИХ ПРИСТРОЇВ, З ВИКОРИСТАННЯМ БАГАТОКРИТЕРІАЛЬНИХ ФІЛЬТРІВ ТА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ.....	34
3.1 РЕАЛІЗАЦІЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	34
3.2 ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	38
3.3 ВИКОРИСТАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	39
Висновки до розділу 3 .....	42
ВИСНОВКИ .....	43
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	44
ДОДАТОК А ФРАГМЕНТИ ЛІСТИНГУ .....	46

## ВСТУП

**Актуальність теми.** В наш час буде важко знайти людину яка не має телефону, адже в сучасному світі в телефоні ми маємо абсолютно все що потрібно. В те й же час мати просто телефон теж не можливо. При купівлі будь-якого мобільного пристрою вам в комплекті зазвичай йде зарядка, яка вже є гаджетом. Але часто люди собі щось докуповують, наприклад: якщо ви любите слухати музику, то можете купити навушники, або просто колонку, якщо ви любите пограти в ігри, то я різні гаджети, які можуть спросити процес гри.

Таким чином, розробка веб-сайту інтернет-магазину з продажу гаджетів для мобільних пристроїв, з використанням багатокритеріальних фільтрів та рекомендаційною системою є достатньо актуальним та необхідним напрямком практичних досліджень.

**Мета дослідження** є розробка програмного забезпечення веб-сайту інтернет-магазину з продажу гаджетів для мобільних пристроїв з застосуванням сучасних технологій та реалізації функцій багатокритеріальних фільтрів і рекомендаційної системи підбору товарів.

Завдання дослідження полягає:

- розглянути існуючі аналоги та виявити їхні переваги та недоліки;
- розробити рекомендаційну систему підбору товарів, яка буде базуватися на нещодавно переглянутих гаджетах;
- спроектувати базу даних та інтерфейс користувача
- розробити програмне забезпечення вебсайту інтернет магазину, в якому реалізувати функцію багатокритеріальних фільтрів для зручного пошуку.

**Об'єктом дослідження** являється процес продажу гаджетів через інтернет-магазин та збір відгуків для подальшого покращення веб-сайту.

**Предметом дослідження** є продаж гаджетів для мобільних пристроїв з застосуванням сучасних технологій та реалізації функцій багатокритеріальних фільтрів і рекомендаційної системи підбору товарів.

**Методи дослідження:** методи та алгоритми розробки, а також впровадження багатокритеріальних фільтрів та рекомендаційної системи на веб-сайті інтернет-магазину з продажу гаджетів для мобільних пристроїв.

**Практичне значення.** Розробка веб-сайту інтернет-магазину з продажу гаджетів для мобільних пристроїв, з використанням багатокритеріальних фільтрів та рекомендаційної системи.

**Структура та обсяг** пояснювальної записки. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків та списку посилань (17 найменувань). Пояснювальна записка містить 19 рисунків. Загальний обсяг пояснювальної записки складає 46 сторінок, основний зміст викладено на 43 сторінках.

## РОЗДІЛ 1

# ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ВЕБ-САЙТУ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ГАДЖЕТІВ ДО МОБІЛЬНИХ ПРИСТРОЇВ, З ВИКОРИСТАННЯМ БАГАТОКРИТЕРІАЛЬНИХ ФІЛЬТРІВ ТА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

### 1.1 Опис предметної області

Електронна комерція(e-commerce)[1] - це сфера цифрової економіки, що включає всі фінансові та торгові транзакції, які проводяться за допомогою комп'ютерних мереж, та бізнес-процеси, пов'язані з проведенням цих транзакцій. В умовах стрімкого розвитку цифрових технологій та змін у споживчих звичках, онлайн-торгівля стає основним каналом реалізації товарів. Основними перевагами e-commerce є зручність для користувачів, широка аудиторія охоплення, автоматизація бізнес-процесів та зменшення витрат на утримання фізичних магазинів. Значною проблемою сучасних інтернет-магазинів є великий асортимент продукції, що ускладнює вибір для користувачів. Тому важливими елементами сайту стають багатокритеріальні фільтри, що дозволяють швидко відсортувати товари за певними параметрами (ціна, технічні характеристики тощо), та рекомендаційна система, яка формує індивідуальні пропозиції для кожного користувача на основі його інтересів, переглянутих товарів, історії покупок тощо.

Основні функції:

1. Каталог товарів - веб-сайт пропонує широкий асортимент гаджетів. Кожен товар має детальний опис, який включає технічні характеристики, фотографії, відгуки користувачів та рейтинги. Це дозволяє користувачам отримати повну інформацію про продукт перед його придбанням;
2. Багатокритеріальні фільтри - для зручності користувачів веб-сайт оснащений багатокритеріальними фільтрами, які дозволяють швидко знайти потрібний товар. Фільтри включають категорії товарів (наприклад, чохли,

зарядні пристрої), бренди, ціну, колір. Користувачі можуть комбінувати фільтри для точного пошуку, що значно спрощує процес вибору;

3. Рекомендаційна система - веб-сайт використовує рекомендаційну систему, яка персоналізує пропозиції на основі переглядів користувача. Це дозволяє користувачам отримувати рекомендації щодо популярних товарів та новинок, які можуть їх зацікавити.

Види товарів:

1. Чохли: захисні чохла для мобільних телефонів, які можуть бути виготовлені з різних матеріалів, таких як силікон, пластик, шкіра. Чохли можуть мати різні дизайни та кольори, а також додаткові функції, такі як підставки;

2. Зарядні пристрої: різноманітні зарядні пристрої, включаючи кабелі, адаптери, бездротові зарядні станції та портативні зарядні пристрої (power banks). Зарядні пристрої можуть бути сумісні з різними моделями мобільних пристроїв та мати різні технічні характеристики;

3. Навушники: внутрішньо канальні, накладні та бездротові навушники, які забезпечують високоякісний звук та комфортне використання. Навушники можуть мати додаткові функції, такі як шумозаглушення та вбудований мікрофон;

4. Захисні екрани: захисні плівки та скло для екранів мобільних телефонів та планшетів, які забезпечують захист від подряпин, ударів та інших пошкоджень. Захисні екрани можуть мати різні рівні міцності та прозорості.

## **1.2 Аналіз потенційних конкурентних переваг програмного продукту**

На ринку існує велика кількість інтернет-магазинів з продажу гаджетів. Тому для того щоб визначити переваги інтернет-магазину “GadgetHub”, потрібно визначити плюси та мінуси існуючих аналогів. Із головних конкурентів були обрані наступні веб сайти – “Rozetka”, “Citrus” та “Eldorado”.

Rozetka (рис. 1.1) – один із найбільших інтернет-магазинів, що пропонує широкий вибір товарів, включаючи гаджети.

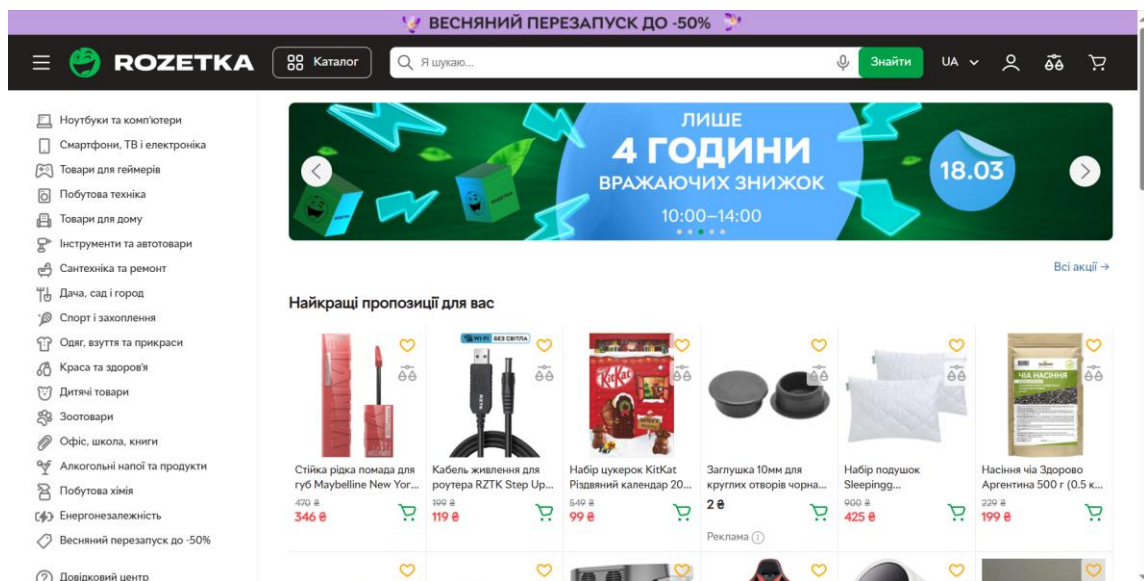


Рисунок 1.1 – Сайт магазину “Rozetka”

Джерело [2]

Переваги.

Великий асортимент – широкий вибір смартфонів, аксесуарів і гаджетів.

Зручна система фільтрів – можна сортувати за брендом, ціною, характеристиками, рейтингом.

Реальні відгуки користувачів – багато коментарів і оцінок від покупців.

Детальні описи та фото – більшість товарів мають хороші описи та реальні зображення.

Недоліки.

Перевантаженість інтерфейсу – Велика кількість банерів, акційних пропозицій та реклами може ускладнювати навігацію та відволікати користувача.

Не завжди в наявності – популярні гаджети можуть бути недоступні або їхній статус оновлюється із запізненням.

Довгий пошук потрібної моделі – через велику кількість товарів знайти конкретну річ може бути складно.

Citrus(рис. 1.2) – спеціалізований магазин гаджетів та аксесуарів.

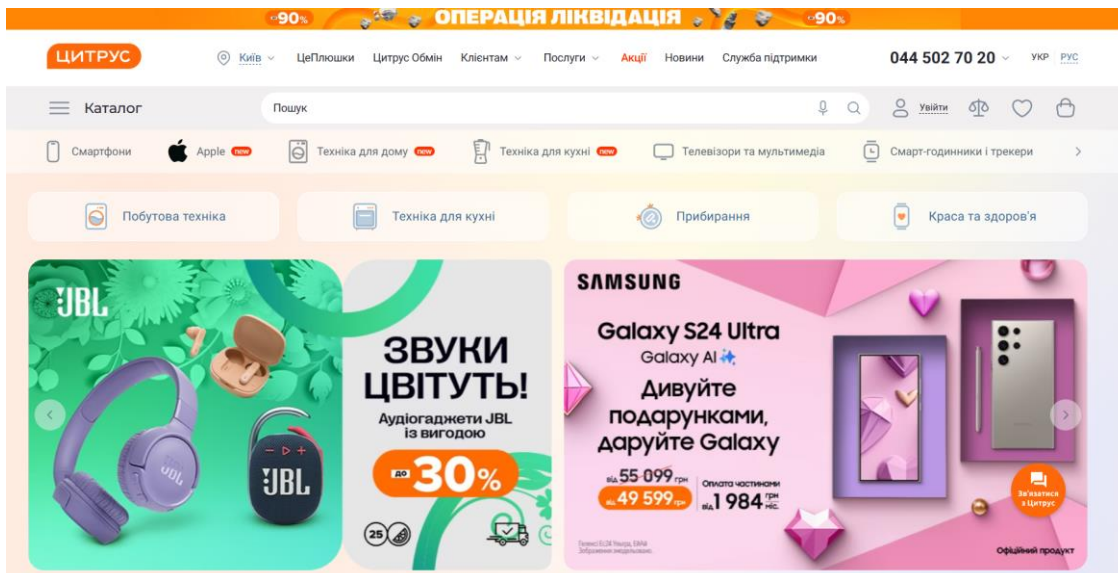


Рисунок 1.2 – Сайт магазину “Citrus”

Джерело [3]

Переваги.

Стильний та сучасний дизайн – Інтерфейс виглядає яскраво та сучасно, що створює позитивне враження та підкреслює фірмовий стиль магазину.

Зручний пошук та фільтрація – можна швидко знайти потрібний товар за параметрами.

Якісні фото та відео огляди – часто додають відео-огляди гаджетів.

Недоліки:

Перевантаженість рекламою на сайті – Велику частину сторінки займають банери та акційні пропозиції, що може відволікати від пошуку товару.

Фільтри працюють не завжди коректно – іноді показує товари, які не відповідають вибраним параметрам.

Часті проблеми з наявністю товару – може бути на сайті, але в реальності його немає.

Eldorado(рис 1.3) – міжнародна мережа магазинів електроніки з онлайн-платформою.

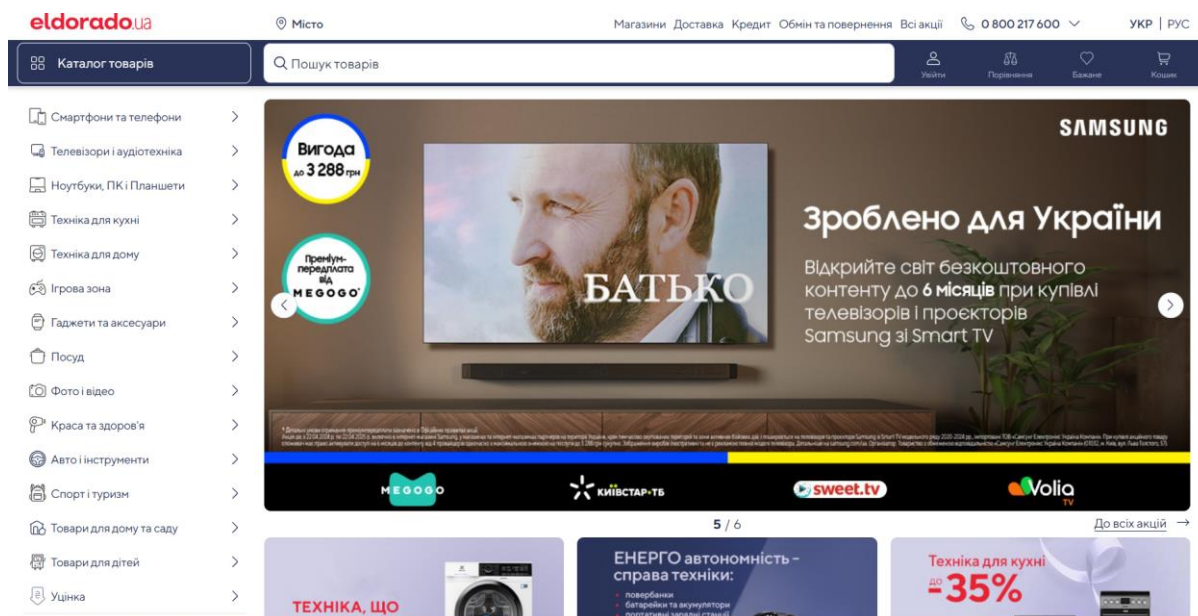


Рисунок 1.3 – Сайт магазину “Eldorado”

Джерело [4]

**Переваги.**

Зрозумілий інтерфейс – простий дизайн сайту, легкий пошук товарів.

Зручний фільтр та пошук товарів – Покупці можуть швидко знайти потрібний товар за брендом, ціною, характеристиками та іншими параметрами.

Можливість забронювати товар у магазині – можна знайти потрібний гаджет та забрати його офлайн.

**Недоліки.**

Перевантаженість сайту рекламними банерами – Велика кількість акційних банерів може відволікати та уповільнювати пошук потрібного товару.

Мало відгуків користувачів – складно оцінити товар без реальних відгуків.

Обмежена інформація про товари – короткі описи, мало реальних фото. Для того щоб розроблюваний інтернет-магазин “GadgetHub” мав конкурентні переваги перед його аналогами, потрібно впровадити наступні функціональні можливості.

Багатокритеріальна фільтрація - Пошук товарів за кількома параметрами одночасно.

Персоналізована рекомендаційна система - генерація індивідуальних пропозицій на основі вподобань.

Адаптивний дизайн та швидкість роботи - оптимізація сайту для мобільних пристроїв.

### **1.3 Постановка завдання на кваліфікаційну роботу**

Розробка веб-сайту інтернет магазину “GadgetHub” являє собою реалізацію багатоетапного процесу в який входить: визначення потреб користувачів, розробка та впровадження багатокритеріальних фільтрів та рекомендаційної системи, створення зручного та приємного дизайну, а також забезпечення якості та ефективності платформи. Цей проєкт передбачає такі завдання:

1. Визначення цілей проєкту: Перший крок це визначення цілей при створенні веб-сайту інтернет магазину “GadgetHub”. Цілі включають в себе розробку веб-сайту, аналіз вимог клієнтів, впровадження багатокритеріальної фільтрації та рекомендаційної системи на веб-сайті.

2. Аналіз вимог користувача: Необхідно провести аналіз уподобань та очікувань клієнтів від “GadgetHub”. Це включає в себе оцінку ринку мобільних пристроїв, аналіз останніх покупок клієнтів та інших аспектів, які впливають на їхнє рішення при виборі гаджетів.

3. Розробка функціональних вимог: На основі вимог користувачів визначаються функціональні вимоги до сайту, такі як можливості пошуку та фільтрації товарів, сторінки товарів з детальним описом.

4. Проектування інтерфейсу користувача: Створення дизайну інтерфейсу веб-сайту, який буде зручним та комфортним у використанні.

5. Розробка та тестування: Проводиться програмування та створення сайту враховуючі попередні вимоги. Після цього проводиться тестування та виправлення помилок та недоліків

Вхідні дані:

- дані користувачів (ім'я, e-mail, пароль для входу)

*Необхідні для створення облікового запису користувача та обробки замовлень.*

- інформація про товари (назва, опис, ціна, зображення)

*Використовується для наповнення каталогу та відображення товарів на сайті.*

Вихідні дані:

- відображення списку товарів та їх деталей

*Забезпечує користувача інформацією про доступні товари;*

- повідомлення про помилки (наприклад, неправильний e-mail, нестача товару);

- інформує користувача про помилки при введенні або недоступність товару.

Функціональні вимоги:

- реєстрація та авторизація користувачів

*Дозволяє створювати персональні акаунти для здійснення покупок;*

- перегляд каталогу товарів із фільтрами

*Полегшує пошук потрібного товару за категоріями, ціною тощо;*

- перегляд сторінки товару (зображення, опис, ціна)

*Надає повну інформацію про конкретний товар;*

- додавання товарів у кошик та зміна кількості  
*Дозволяє користувачам формувати замовлення;*

Нефункціональні вимоги.

- оптимізація сайту для мобільних пристроїв  
*Забезпечує зручне користування з телефонів і планшетів;*

- простий та зрозумілий інтерфейс  
*Сприяє комфортному користуванню сайтом навіть недосвідченим користувачам;*

- можливість додавання нових категорій товарів  
*Забезпечує масштабованість системи для розширення асортименту.*

Автоматизовані функції:

- автоматична перевірка правильності введення даних (e-mail, номер телефону). *Допомагає уникнути помилок при реєстрації та оформленні замовлень;*

- автоматичне збереження вмісту кошика при перезавантаженні сторінки. *Покращує користувацький досвід, зберігаючи дані навіть при випадковому оновленні сторінки.*

## **Висновки до розділу 1**

Проведено дослідження предметної області в якому було сформовано та ціль застосування веб-сайту. Опис предметної області дав чітке визначення проблеми сучасних інтернет-магазинів, а також були вказані головні функції які повинен мати програмний продукт.

Проведено аналіз існуючих аналогів програмного продукту, були визначені плюси та мінуси, також були сформовані переваги веб-сайту “GadgetHub”.

Було визначено цілі проєкту, функціональні вимоги до сайту, вимоги до інтерфейсу.

## РОЗДІЛ 2

# ПРОЄКТУВАННЯ ВЕБ-САЙТ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ГАДЖЕТІВ ДО МОБІЛЬНИХ ПРИСТРОЇВ, З ВИКОРИСТАННЯМ БАГАТОКРИТЕРІАЛЬНИХ ФІЛЬТРІВ ТА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

### 2.1 Моделювання поведінки продукту

Проектування веб-сайту інтернет-магазину є важливим етапом, що визначає його логіку, функціональність, зручність у користуванні та візуальне сприйняття. У даному розділі розглянемо побудову структури сайту «GadgetHub», який спеціалізується на продажу гаджетів для мобільних пристроїв та має функціонал багатокритеріального фільтрування і рекомендацій.

Сайт повинен відповідати основним принципам побудови інформаційних систем: модульність, повторне використання компонентів, масштабованість та підтримка адаптивного дизайну. Це дозволить не лише реалізувати функціональні можливості в межах кваліфікаційної роботи, а й створити передумови для розширення системи в майбутньому.

Функціональна структура сайту включає:

Головну сторінку, яка слугує стартовою точкою та містить слайдер із популярними товарами та новинками;

Каталог товарів з можливістю багатокритеріального фільтрування за категоріями, брендом, ціною тощо;

Інформаційну сторінку кожного товару, що містить опис, зображення, характеристики, відгуки;

Сторінку кошик(обрані товари), куди користувачі можуть додавати товари для подальшого перегляду;

Модуль рекомендацій, який динамічно оновлюється на основі історії перегляду товарів;

Навігаційне меню, яке дозволяє легко перемикатися між основними розділами сайту.

В основі побудови структури сайту – принцип SPA (Single Page Application)[5], що забезпечується використанням JavaScript. Це означає, що перехід між розділами відбувається без повного перезавантаження сторінки, що забезпечує більшу швидкість та зручність користування.

Схематична структура компонентів вебсайту:

- Header (логотип, меню, пошук);
- Main (динамічний контент залежно від сторінки);
- Filters Sidebar (для сторінки каталогу);
- Product Cards (товари);
- Recommendation Block (на головній);
- Footer (контакти, політика конфіденційності).

Кожен з компонентів реалізований у вигляді HTML-блоку з відповідним CSS-оформленням та динамічною логікою на JavaScript.

Проектування навігації: меню навігації дозволяє переміщатись між головною сторінкою, каталогом, обраними товарами, а також містить пошукове поле. Меню є адаптивним – при зміні розміру екрана воно трансформується у випадаючий список (бургер-меню).

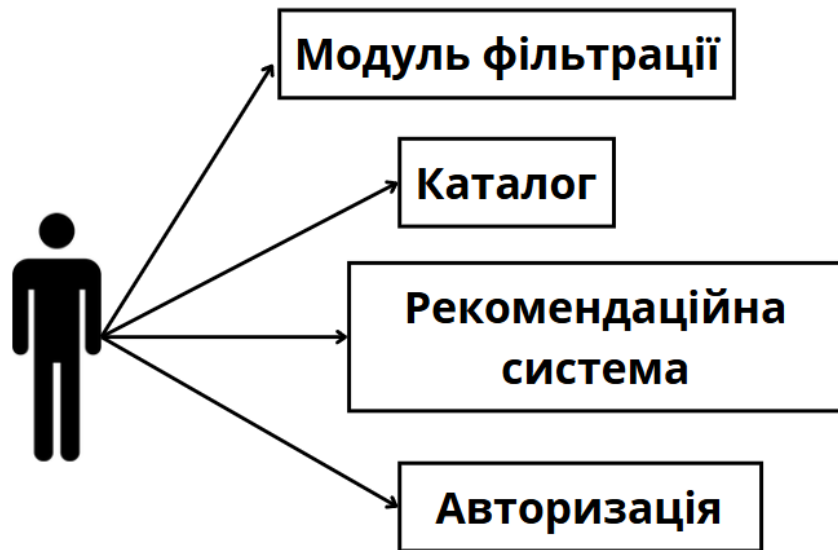
Проектування багатокритеріальних фільтрів: фільтри реалізовано у вигляді списку чекбоксів та повзунка для цінового діапазону. Після обрання параметрів товари миттєво оновлюються без перезавантаження сторінки.

Формування рекомендаційної системи. Блок рекомендованих товарів формується за такими критеріями:

- товари, які переглядав користувач останніми;
- товари з тієї ж категорії;
- популярні товари загалом.

Це реалізується через локальне збереження переглядів у LocalStorage і фільтрацію даних за категоріями та кількістю переглядів.

Діаграма прецедентів [6] (рис. 2.1) є важливим інструментом моделювання функціональності сайту інтернет-магазину GadgetHub, а також взаємодії користувачів із його основними модулями. Такий тип діаграм використовується для візуалізації ролей користувачів та їхніх можливих дій у межах системи.



*Рисунок 2.1 Діаграма прецедентів*

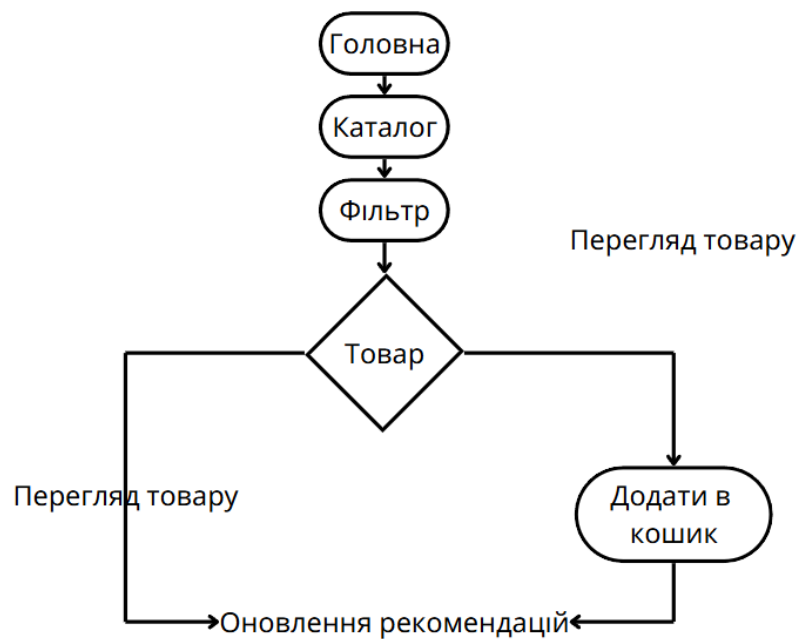
*Джерело створено автором*

Прецедент:

1. Модуль фільтрації – забезпечує можливість багатокритеріального фільтрування товарів у каталозі. Користувач надсилає запити для відбору потрібних товарів за категоріями, брендами, ціною тощо.
2. Каталог – відображає список товарів відповідно до запиту користувача (фільтрація, пошук, сортування) та дозволяє переглядати деталі кожного товару.

3. Рекомендаційна система – надає персоналізовані рекомендації, використовуючи збережену історію переглядів користувача. Вона отримує дані з локального сховища (LocalStorage).
4. Авторизація – використовується користувачем для входу в свій обліковий запис. Дані користувача зберігаються в базі даних на платформі Firebase

Діаграма діяльності [7] (рис. 2.2) використовується для візуалізації алгоритму виконання певної послідовності дій у межах системи. У контексті сайту GadgetHub вона демонструє логіку дій користувача під час пошуку та вибору товару за допомогою багатокритеріальних фільтрів або рекомендаційної системи.



*Рисунок 2.2 Діаграма діяльності*

*Джерело створено автором*

Умовно весь процес взаємодії починається з ініціації користувачем доступу до головної сторінки сайту. Далі можливі дві основні гілки:

1. Перехід до каталогу товарів із наступним використанням фільтрів для точнішого вибору;
2. Ознайомлення з рекомендованими товарами на основі історії переглядів.

Після вибору товару можливий перегляд його детальної інформації або додавання до обраного. У разі наявності реєстрації – додатково виконується авторизація для збереження персоналізованого списку.

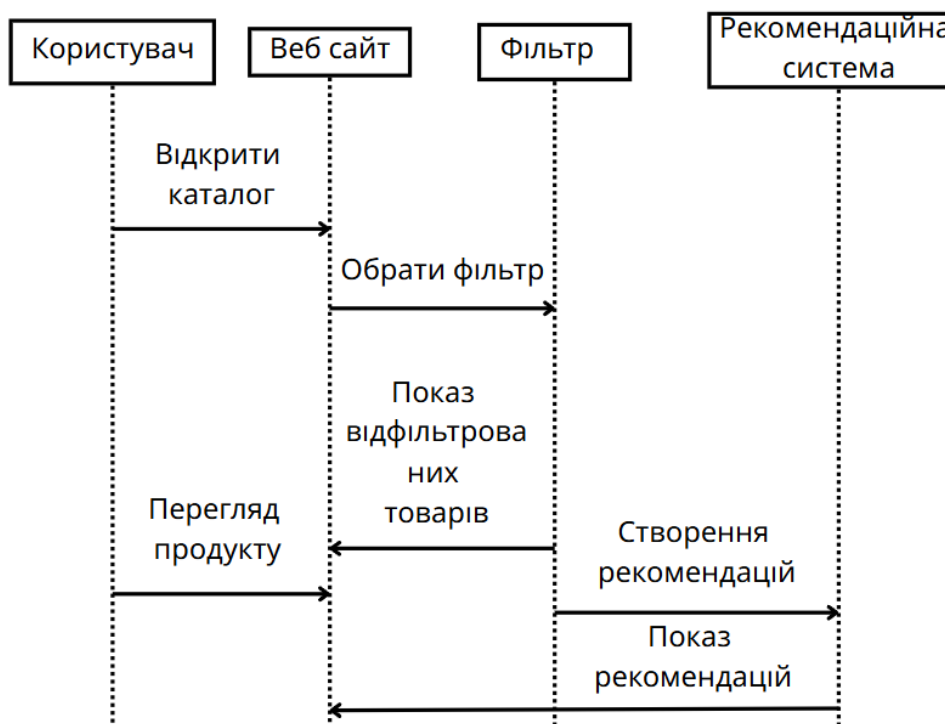
Діаграма діяльності дозволяє відобразити взаємозв'язки між різними сценаріями, що забезпечують гнучкість і логічність користувацького досвіду на сайті.

Діаграма послідовності [8] (рис. 2.3) демонструє послідовну взаємодію між основними компонентами системи сайту GadgetHub у процесі вибору товару через сторінку каталогу. Вона дозволяє відобразити обмін повідомленнями між об'єктами системи у часовій послідовності.

Сценарій виглядає наступним чином:

1. Користувач відкриває сторінку каталогу.
2. Інтерфейс сайту надсилає запит на завантаження списку товарів.
3. Користувач обирає параметри фільтрації (категорії, бренд, ціна тощо).
4. Сайт передає ці дані у модуль фільтрації, який виконує пошук релевантних товарів.
5. Відібрані товари повертаються до інтерфейсу і виводяться на сторінку.
6. Користувач відкриває детальну сторінку обраного товару.
7. Ця дія зберігається як перегляд у локальне сховище (LocalStorage).
8. Дані передаються до рекомендаційної системи, яка формує перелік товарів на основі останніх переглядів.
9. Інтерфейс сайту відображає блок з рекомендаціями.

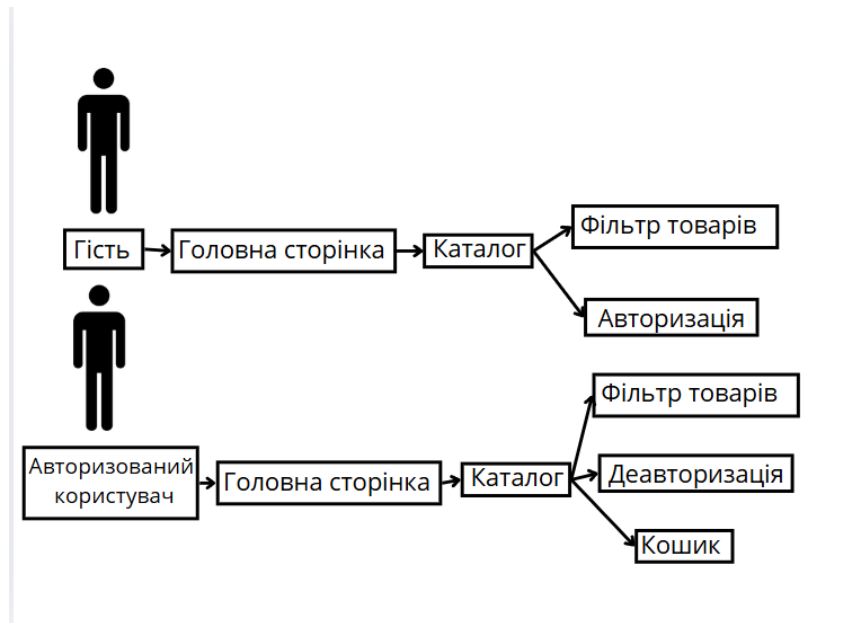
Діаграма послідовності дозволяє відстежити не лише порядок дій користувача, а й логіку обміну даними між модулями сайту. Це важливо для реалізації клієнтської логіки та подальшого розширення функціоналу.



*Рисунок 2.3 - Діаграма послідовності*

*Джерело розроблено автором*

На діаграмі комунікації [9] (рис. 2.4) зображено як різні ролі користувачів взаємодіють із функціональними компонентами сайту GadgetHub. Зокрема, представлено сценарії для гостя та авторизованого користувача. Зазвичай такі діаграми містять основні об'єкти, які беруть участь у процесах, а також стрілки або лінії зв'язку, що відображають способи обміну даними між ними. Це допомагає розробникам та аналітикам виявити потенційні проблеми у структурі взаємодії, оптимізувати процеси та покращити користувацький досвід.



*Рисунок 2.4 – діаграма комунікації*

*Джерело розроблено автором*

Гість:

1. Взаємодіє з головною сторінкою – може переглядати загальну інформацію про сайт.
2. Переходить до каталогу товарів – доступний перегляд усього асортименту.
3. Має доступ до фільтрації товарів – може застосовувати критерії (категорія, ціна, бренд).
4. У разі потреби авторизації – перенаправляється до відповідної сторінки для входу/реєстрації.
5. Авторизований користувач:
6. Також починає з головної сторінки – інтерфейс може бути розширеним (наприклад, з кнопкою «Кошик»).
7. Використовує каталог товарів – переглядає й відфільтровує позиції.
8. Має можливість додати товари в кошик – формується замовлення.
9. Може виконати деавторизацію – вихід із системи через відповідну функцію.

Кожна з дій у діаграмі представлена стрілкою, що ілюструє напрям комунікації користувача з окремими функціональними модулями системи. Така діаграма дозволяє наочно побачити відмінність у можливостях між гостем і зареєстрованим користувачем.

## 2.2 Моделювання структури продукту

Моделювання структури програмного продукту передбачає створення статичних моделей, які відображають основні сутності системи, їх атрибути, зв'язки та ієрархію. Цей підрозділ зосереджений на аналізі логіки побудови сайту GadgetHub з точки зору об'єктно-орієнтованого підходу до програмування.

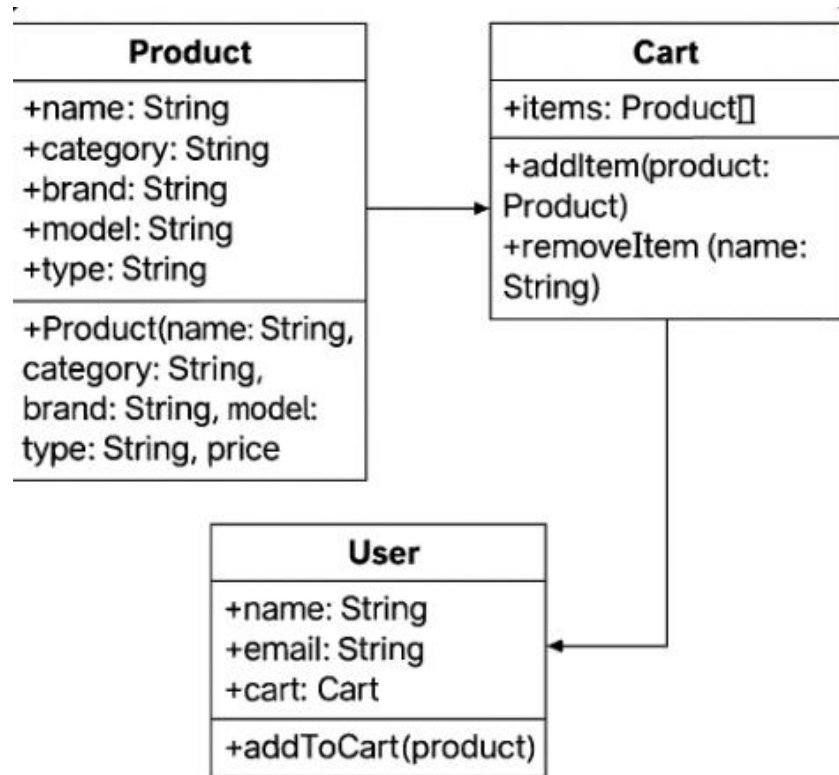
До основних діаграм, що дозволяють описати структуру програмного продукту, належать:

- діаграма класів;
- діаграма об'єктів;
- діаграма пакетів.

Кожна з них слугує інструментом для моделювання різних аспектів системи: класів та їхніх зв'язків, конкретних об'єктів у момент часу, структурних залежностей між модулями.

Діаграма класів (Class Diagram)[10](рис. 2.5)

Діаграма класів ілюструє об'єктно-орієнтовану структуру веб-сайту GadgetHub, що використовується для управління товарами, кошиком покупок та користувачами. Типові діаграми класів містять класи, позначені прямокутниками, які включають властивості та функції, що характеризують кожен об'єкт. Лінії між класами вказують на асоціації, наслідування чи залежності між ними. Це дозволяє чітко визначити, як елементи системи пов'язані між собою та як вони передають дані або викликають функції один одного.



*Рисунок 2.5 – Діаграма класів*

*Джерело розроблено автором*

Основні класи:

- Product (Товар): Цей клас представляє окремий товар у каталозі. Він містить атрибути, такі як назва товару (name), категорія (category), бренд (brand), модель (model), тип (type), та ціна (price). Конструктор класу дозволяє створювати екземпляри товарів із зазначеними властивостями.
- Cart (Кошик): Клас для управління списком товарів у кошику користувача. Містить методи:
  - addItem(product): додає товар до кошика.
  - removeItem(productName): видаляє товар за його назвою;
- getTotalPrice(): розраховує загальну вартість товарів у кошику;
- User (Користувач): Клас, що моделює користувача сайту. Має атрибути name (ім'я) та email (електронна пошта). У цьому класі є об'єкт cart, який асоціюється з кошиком покупок користувача. Основні методи:

- addToCart(product): додає товар до кошика користувача.
- checkout(): виводить загальну суму вартості товарів у кошику.

Взаємодія класів:

1. Користувач (User) має власний кошик (Cart), який містить об'єкти товарів (Product).
2. При додаванні товарів до кошика, вони зберігаються у списку об'єктів items у класі Cart.

Діаграма класів допомагає зрозуміти зв'язки між основними компонентами системи та спрощує розробку логіки веб-застосунку.

Діаграма об'єктів (Object Diagram) [11] (рис. 2.6).

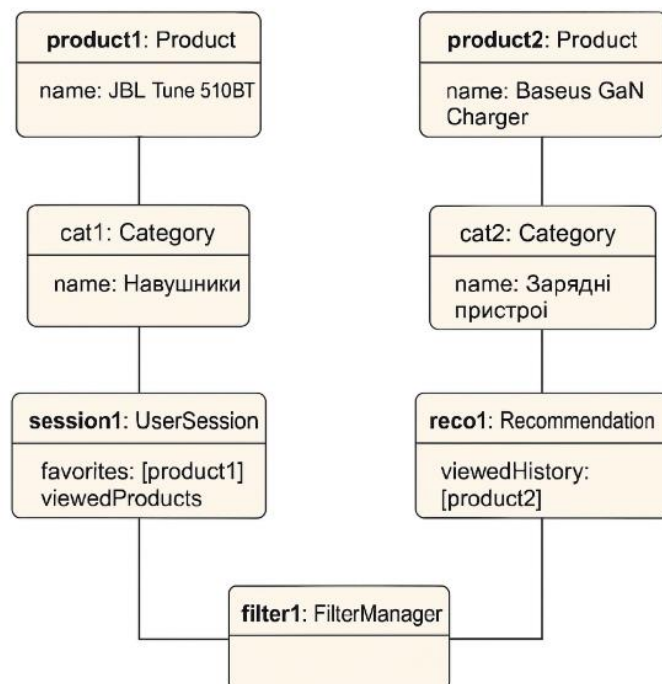


Рисунок 2.6 – Діаграма об'єктів

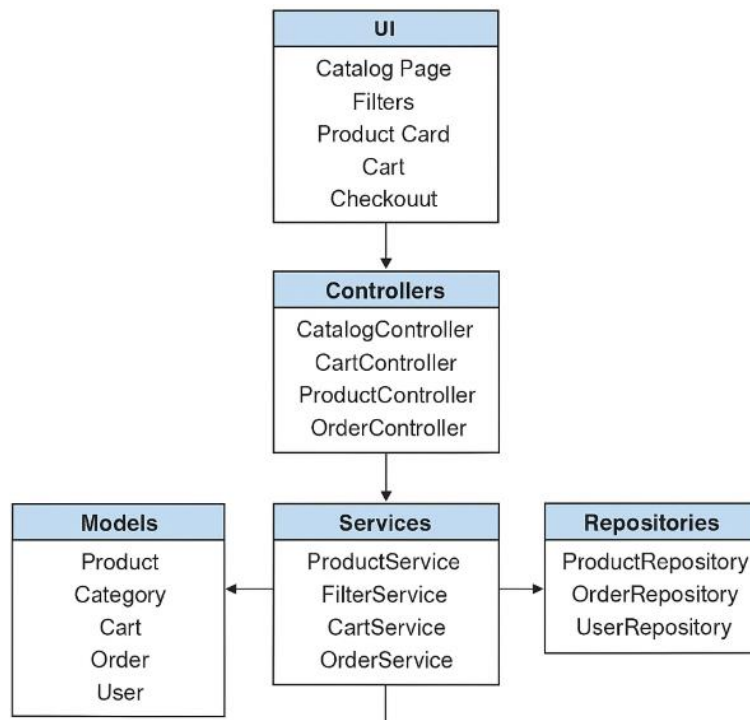
Джерело розроблено автором

Діаграма об'єктів є конкретним прикладом стану системи у певний момент часу, де на основі діаграми класів зображено екземпляри об'єктів з

їхніми поточними значеннями атрибутів. Для сайту GadgetHub така діаграма може демонструвати, наприклад, що:

- у категорії "Навушники" є об'єкти product1, product2, product3;
- користувач має відкриту сесію та переглядає товар product1;
- у локальному сховищі зберігається перелік обраного.

Діаграма пакетів[12] (Package Diagram, рис. 2.7)



*Рисунок 2.7 – Діаграма пакетів*

*Джерело розроблено автором*

Ця діаграма описує модульну структуру продукту – поділ системи на логічні блоки та залежності між ними. У випадку GadgetHub можна виокремити такі пакети:

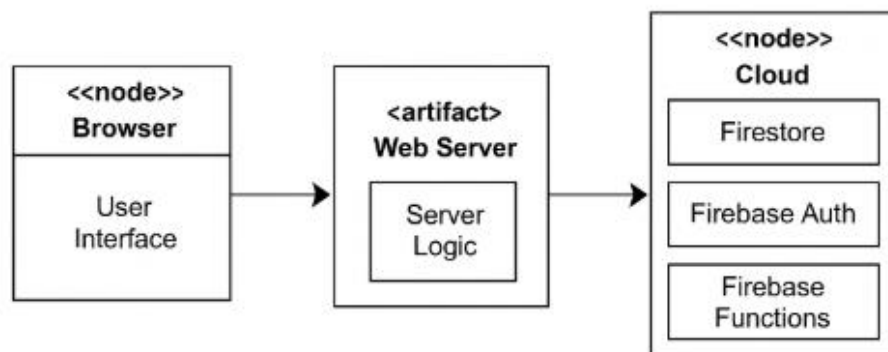
1. UI - Містить компоненти інтерфейсу користувача (каталог, фільтри, кошик, сторінка товару, тощо).

2. **Controllers**- Обробляє запити від користувача, керує логікою переходів між сторінками, ініціалізує дії.
3. **Services** - Містить бізнес-логіку: робота з кошиком, замовленнями, користувачами.
4. **Models** - Сюди входять сутності, що зберігають дані (Product, User, Order, CartItem тощо).
5. **Database/Repositories** - Робота з БД: запити, збереження даних, оновлення.

Діаграма дозволяє візуалізувати, які модулі залежать один від одного, і сприяє майбутньому масштабуванню системи.

Таким чином, структурне моделювання дозволяє формалізувати логіку побудови сайту, описати його архітектурні блоки, залежності між компонентами та типи даних, що використовуються. Це є основою для подальшого обслуговування, оптимізації та масштабування програмного продукту.

Діаграма розгортання (Deployment Diagram, рис. 2.8)



*Рисунок 2.8 – Діаграма розгортання*

*Джерело розроблено автором*

Ця діаграма показує фізичне розташування компонентів системи у середовищі виконання. У випадку GadgetHub: клієнтська частина (браузер

користувача) запускає інтерфейс, серверна логіка працює на хостингу (наприклад, Firebase Functions[13]), а дані зберігаються у хмарній базі Firestore. Також може використовуватись служба авторизації Firebase Auth. Ця діаграма дозволяє уявити, як система працює в реальних умовах.

Діаграма компонентів (Component Diagram, рис. 2.9)

Діаграма ілюструє, з яких логічних компонентів складається система та як вони взаємодіють. У проєкті GadgetHub основними компонентами є: Client (фронтенд-частина), BackendService (обробка логіки), FirebaseService (зв'язок із базою даних), AuthService (авторизація) та ProductService (робота з каталогом товарів). Така діаграма дозволяє чітко розмежувати відповідальність між частинами системи та забезпечити модульність, що спрощує тестування й підтримку.

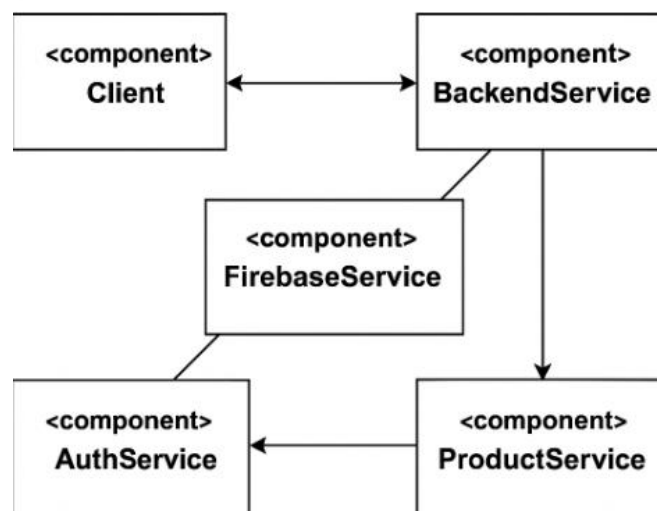


Рисунок 2.9 – діаграма компонентів

*Джерело розроблено автором*

### 2.3 Опис архітектури продукту

Архітектура вебзастосунку є фундаментальною основою його функціонування, забезпечуючи взаємодію між окремими компонентами, а

також визначаючи структуру, логіку обробки даних та доступ до ресурсів. В основі архітектури розробленого продукту лежить трирівнева (three-tier) модель, адаптована до сучасного підходу створення веб застосунків з використанням Firebase як бекенд-сервісу.

### Архітектурна модель

#### 1. Рівень представлення (Presentation Layer)

Цей рівень відповідає за взаємодію з користувачем. Інтерфейс користувача реалізований за допомогою HTML, CSS та JavaScript, що дозволяє створювати динамічні, інтерактивні веб сторінки. Саме тут користувачі взаємодіють із застосунком – переглядають каталог товарів, застосовують фільтри, переглядають характеристики тощо.

#### 2. Рівень логіки (Application Logic Layer)

У межах цього проєкту рівень логіки не реалізовано на стороні окремого сервера або традиційного бекенду. Натомість бізнес-логіка реалізується безпосередньо на клієнтській стороні за допомогою JavaScript. Усі обчислення, перевірки введених даних, фільтрація та сортування відбуваються на стороні клієнта.

#### 3. Рівень даних (Data Layer)

У якості засобу зберігання та обробки даних використовується Firebase – хмарна платформа від Google, яка забезпечує:

- зберігання даних у форматі JSON у Firebase Realtime Database;
- авторизацію користувачів;

Такий підхід дозволяє уникнути створення окремого сервера та зосередитись на розробці логіки застосунку на стороні клієнта.

### Простори імен, функції та структури

У структурі продукту виділено такі основні частини:

- `index.html` – основний файл, з якого починається завантаження інтерфейсу. Містить елементи інтерфейсу: заголовки, контейнер для каталогу, елементи фільтрації, тощо.

- style.css – таблиця стилів, відповідальна за зовнішній вигляд сторінок: сітку, кольори, шрифти, розміщення елементів.
- catalog.html – файл, що містить функції для динамічного формування вмісту каталогу на сторінці та реалізації багатокритеріальної фільтрації товарів,
- scripts/firebase-config.js – містить конфігурацію та ініціалізацію Firebase-з'єднання з базою даних.

Код логіки побудовано функціонально, без класів. Взаємодія з базою даних виконується через Firebase SDK, що дозволяє асинхронно отримувати дані та реагувати на їх зміни.

Модель Крухтена 4+1. Опис програмного продукту також може бути представлений через модель Крухтена 4+1, яка включає п'ять основних представлень:

1. Логічне представлення (Logical View) Містить опис основної функціональності системи: перегляд каталогу, застосування фільтрів, виведення інформації про товари.
2. Процесне представлення (Process View) Враховує взаємодію користувача з системою у вигляді сценаріїв використання: завантаження сторінки каталогу, введення фільтрів, оновлення списку товарів.
3. Фізичне представлення (Physical View) Застосунок розміщується у хмарному середовищі (Firebase Hosting). Всі файли зберігаються у статичному вигляді, а доступ до бази даних відбувається через API Firebase.
4. Розгортання (Development View) Розробка реалізована у вигляді набору HTML/CSS/JS файлів. Структура проекту є плоскою, логіка винесена у окремі скрипти. Дані завантажуються через Firebase SDK.
5. Сценарії використання (Use Case View) Включають основні взаємодії користувача з сайтом: відкриття каталогу, вибір фільтрів, перегляд деталей товару.

Модель С4 для архітектурного опису веб-сайту GadgetHub(рис. 2.10)

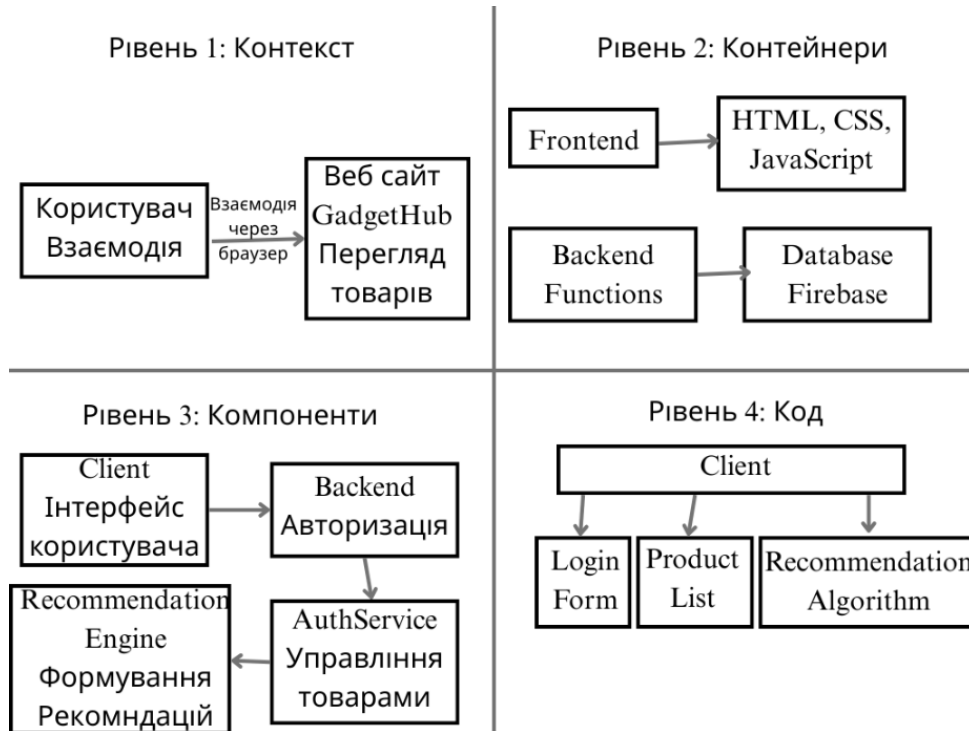


Рисунок 2.10 – Модель C4

Джерело розроблено автором

Модель C4 – це підхід до візуалізації архітектури програмних систем, що розбиває систему на чотири рівні представлення: Контекст (Context), Контейнери (Containers), Компоненти (Components) та Код (Code). Такий підхід дозволяє чітко структурувати архітектурні рішення та зрозуміти взаємозв'язки між елементами.

Рівень 1: Контекст. На найвищому рівні модель C4 демонструє загальний контекст роботи системи. У випадку GadgetHub ключовими учасниками є:

1. Користувач – взаємодіє із системою через браузер.
2. Веб-сайт GadgetHub – забезпечує доступ до каталогу товарів, пошуку, фільтрації та рекомендацій.
3. Firebase – хмарний бекенд для зберігання даних товарів, користувачів та історії покупок.

Рівень 2: Контейнери. Наступний рівень деталізує основні контейнери системи:

1. Клієнтська частина (Frontend) – веб-додаток, що виконується у браузері користувача (HTML, CSS, JavaScript).
2. Серверна логіка (Backend) – серверні функції, реалізовані у Firebase Functions.
3. База даних (Database) – хмарне сховище Firebase Firestore, де зберігаються дані про товари, користувачів та замовлення.

Рівень 3: Компоненти. На рівні компонентів розглядаються окремі логічні модулі, такі як:

1. Client (Інтерфейс користувача) – забезпечує взаємодію з користувачем.
2. AuthService – виконує авторизацію та управління обліковими записами.
3. ProductService – відповідає за роботу з товарами.
4. RecommendationEngine – формує персоналізовані рекомендації.

Рівень 4: Код. Останній рівень деталізує реалізацію кожного компонента через програмні модулі та класи.

Модель C4 допомагає чітко розуміти архітектуру продукту, розмежувати відповідальність між частинами системи та забезпечити гнучкість при розширенні функціоналу.

## **Висновки до розділу 2**

У другому розділі було проведено детальне моделювання та опис структури програмного продукту – веб застосунку для перегляду каталогу товарів із можливістю багатокритеріальної фільтрації.

На етапі моделювання поведінки продукту побудовано діаграми варіантів використання, діяльності, послідовності та комунікації. Вони відображають взаємодію користувача з системою, основні сценарії

користування, передачу повідомлень між об'єктами та загальний процес обробки даних у межах інтерфейсу.

Під час моделювання структури продукту розроблено діаграми класів, об'єктів та пакетів, які описують логічну і фізичну організацію програмного забезпечення, типи об'єктів, їхні зв'язки та взаємодії між частинами системи.

У підрозділі архітектурного опису застосовано модель Крухтена 4+1, яка дозволяє цілісно представити структуру, поведінку, фізичне розгортання та основні сценарії роботи продукту. Описано також ключові простори імен, скрипти та залежності проекту, серед яких особливе місце займає база даних Firebase, що використовується як основне джерело інформації.

Отримані моделі та діаграми створюють картину архітектури, логіки та функціонування веб застосунку, що є основою для його реалізації, тестування та подальшого удосконалення.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ВЕБ-САЙТ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ ГАДЖЕТІВ ДО МОБІЛЬНИХ ПРИСТРОЇВ, З ВИКОРИСТАННЯМ БАГАТОКРИТЕРІАЛЬНИХ ФІЛЬТРІВ ТА РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

#### 3.1 Реалізація та конструювання програмного продукту

Взаємодія між компонентами вебзастосунку. Реалізація вебсайту передбачає взаємодію між різними модулями системи, що відповідають за відображення інформації, управління користувацькими запитами та забезпечення доступу до даних. Для ефективної роботи інтернет-магазину необхідно налаштувати систему комунікації між клієнтом і сервером.

Зокрема, розробка GadgetHub включає такі механізми:

1. Обмін даними через API – застосунок виконує асинхронні запити до Firebase для отримання та оновлення інформації про товари.
2. Динамічне оновлення контенту – зміна даних на сторінці відбувається без перезавантаження завдяки механізму AJAX.
3. Система оновлення кешу – використання LocalStorage дозволяє зберігати тимчасові дані про переглянуті товари.
4. Інтерактивні події – застосування EventListeners у JavaScript дозволяє реагувати на дії користувачів у реальному часі.

Ці технології забезпечують гнучкість, швидкість роботи та комфортний користувацький досвід.

Для реалізації програмного продукту було обрано сучасні вебтехнології, що забезпечують гнучкість, масштабованість та зручність у розробці. Основною мовою програмування було обрано JavaScript, що дозволяє створювати інтерактивні елементи на стороні клієнта. Для побудови інтерфейсу використовувались HTML5 та CSS3, які забезпечують коректне відображення сторінок у сучасних браузерях.

Середовище розробки – Visual Studio Code[14] – було обрано через зручний інтерфейс, велику кількість розширень та підтримку багатьох мов програмування.

Для зберігання та обробки даних було використано Firebase Realtime Database – хмарне рішення, яке дозволяє зручно зберігати та синхронізувати дані між користувачами в реальному часі. Такий підхід зменшує потребу в налаштуванні власного сервера та забезпечує масштабованість застосунку.

### Фрагменти лістингу

#### Підключення до Firebase:

```
import { initializeApp } from "firebase/app";
import { getDatabase } from "firebase/database";

const firebaseConfig = {
  apiKey: "...",
  authDomain: "...",
  databaseURL: "https://your-project.firebaseio.com",
  projectId: "...",
  storageBucket: "...",
  messagingSenderId: "...",
  appId: "..."
};

const app = initializeApp(firebaseConfig);
const db = getDatabase(app);
```

### Опис реалізації сховища даних

Firebase Realtime Database є структурованою у вигляді JSON-дерева, де кожен вузол – це ключ-значення. Основні компоненти бази:

- products – інформація про товари (назва, опис, ціна, категорія, зображення);
- categories – список категорій товарів;

- users – дані користувачів.

Firebase забезпечує простий API, авторизацію, а також реальну синхронізацію між клієнтом і сервером без необхідності налаштування бекенду вручну.

У рекомендаційній системі для веб-сайту GadgetHub, розглядається метод який працює за принципом векторного представлення товарів. Основна ідея полягає у тому, щоб кожен товар на сайті було перетворено у математичний вектор на основі його характеристик.

Принцип роботи системи:

1. Перетворення товарів у вектори. Кожен товар описується набором характеристик, таких як category, brand, model, type, price, тощо. Ці характеристики переводяться у числове представлення. В результаті кожен товар отримує свій унікальний вектор у багатовимірному просторі.

2. Обчислення схожості товарів. Для підбору рекомендацій система визначає схожість між товарами за допомогою метрики, наприклад, косинусної відстані. Чим ближчий товар у векторному просторі до вже переглянутого або обраного користувачем, тим вищий його рейтинг у рекомендаціях.

3. Генерація персональних рекомендацій. Аналізуються товари, які користувач переглядав. Знаходяться найближчі товари за їхнім вектором. Система пропонує користувачеві список найбільш релевантних товарів.

Переваги такого підходу:

- висока точність рекомендацій, оскільки вони ґрунтуються на математичних розрахунках;
- гнучкість системи: можна легко додавати нові характеристики та змінювати алгоритм пошуку найближчих товарів;
- ефективність у великій базі даних, оскільки обчислення можна оптимізувати за допомогою спеціалізованих бібліотек машинного навчання.

Особливості проектування користувацького інтерфейсу. При розробці інтерфейсу веб-сайту важливим аспектом є забезпечення його зручності,

інтуїтивності та естетичної привабливості. Враховуючи, що GadgetHub є інтернет-магазином, особливий акцент зроблено на розташуванні елементів управління, логіці навігації та швидкості доступу до необхідної інформації.

Основні принципи, що були враховані під час проєктування:

1. Простота та мінімалізм – веб-сайт має чистий дизайн без зайвих візуальних елементів, що дозволяє користувачам зосередитися на товарах.
2. Адаптивність – сайт оптимізований для коректного відображення на мобільних пристроях та планшетах, використовуючи технологію CSS Media Queries.
3. Кольорова схема – вибір кольорів ґрунтується на контрастності та комфорті для очей, забезпечуючи легке сприйняття інформації.
4. Іконографіка – використання інтуїтивно зрозумілих іконок допомагає спростити навігацію та зменшити текстові навантаження.

Також було впроваджено покращене юзабіліті завдяки інтерактивним елементам: кнопки з плавною анімацією, спливаючі підказки та швидкий пошук товарів.

Оптимізація продуктивності веб-застосунку

Розробка веб-сайту GadgetHub передбачала не лише реалізацію основних функцій, а й забезпечення високої продуктивності для комфортної роботи користувачів. Основні методи оптимізації:

1. Мінімізація та об'єднання CSS і JavaScript-файлів – зменшує кількість HTTP-запитів та покращує швидкість завантаження сторінки.
2. Використання кешування – збереження статичних даних у браузері користувача, що дозволяє зменшити затримки при повторних запитах.
3. Ліниве завантаження (Lazy Loading) – відстрочка завантаження зображень та скриптів до моменту, коли вони дійсно потрібні користувачу.

4. Оптимізація запитів до Firebase – використання ефективного способу отримання даних, уникаючи зайвих запитів.

### **3.2 Тестування програмного продукту**

Тестування програмного продукту є невід'ємною частиною процесу розробки, що дозволяє виявити й усунути помилки, а також переконатися в коректності реалізації функціональності відповідно до поставлених вимог. Для забезпечення надійності й стабільності роботи застосунку було проведено як функціональне, так і модульне тестування.

#### **Функціональне тестування**

Функціональне тестування охоплювало перевірку основних сценаріїв використання веб сайту з позиції кінцевого користувача. Було протестовано такі функції:

1. Додавання та перегляд товарів – перевірено, що користувач може переглядати каталог товарів, відкривати сторінку з описом конкретного товару, фільтрувати товари за категоріями та іншими параметрами.
2. Кошик покупок – забезпечено додавання товарів у кошик, перегляд кошика, зміна кількості товару, видалення позицій та перехід до оформлення замовлення.
3. Сумісність з різними браузерами – перевірено коректну роботу сайту в Google Chrome, Firefox і Microsoft Edge.

Усі зазначені функції пройшли тестування успішно, без критичних помилок.

#### **Модульне тестування**

Модульне тестування було зосереджено на перевірці окремих частин логіки, написаної на JavaScript. Зокрема, були протестовані:

1. Функції взаємодії з Firebase – наприклад, `addProduct()`, `deleteProduct()`, `fetchProducts()` – шляхом перевірки коректного збереження, оновлення та видалення даних у базі.
2. Обробники подій – наприклад, дії при кліку на кнопки "Додати до кошика", "Очистити кошик".
3. Функції фільтрації й сортування товарів – перевірено правильність логіки сортування та фільтрації за категорією, ціною та іншими параметрами.

Тестування проводилось вручну.

### **3.3 Використання програмного продукту**

Розроблений вебсайт(рис. 3.1) призначений для зручного представлення та продажу товарів в онлайн-середовищі. Інтерфейс є інтуїтивно зрозумілим і не потребує спеціальних знань або навичок з боку користувача. Усі функції реалізовано таким чином, щоб забезпечити простоту навігації, швидкість доступу до необхідної інформації та зручність у взаємодії.

Основні можливості програмного продукту:

1. Перегляд каталогу товарів(рис. 3.2) - користувачі можуть ознайомитися з повним переліком товарів, розбитих за категоріями. Кожен товар має детальний опис, зображення, вказану ціну, доступність та інші характеристики(рис. 3.3).
2. Фільтрація та пошук - реалізовано фільтри за категорією, ціною, брендом або іншими параметрами, що значно полегшує пошук необхідного товару.
3. Кошик покупок(рис. 3.4) - користувачі можуть додавати товари до кошика, переглядати їх кількість, редагувати або видаляти позиції.
4. Адаптивність інтерфейсу - сайт коректно відображається як на комп'ютерах, так і на мобільних пристроях, що забезпечує універсальність його використання.

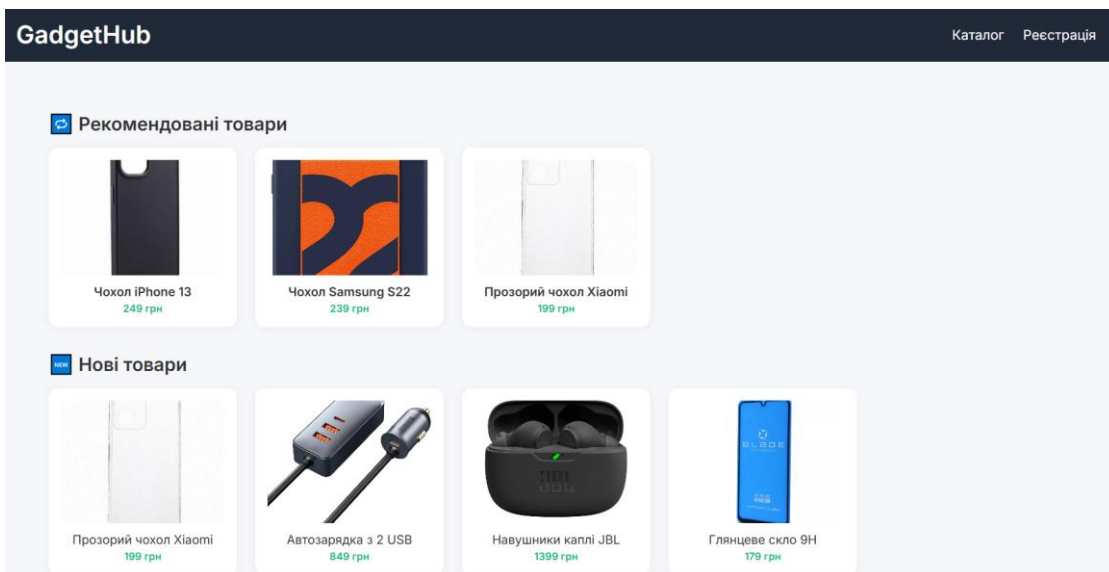


Рисунок 3.1 – Головна сторінка веб сайту  
Джерело розроблено автором

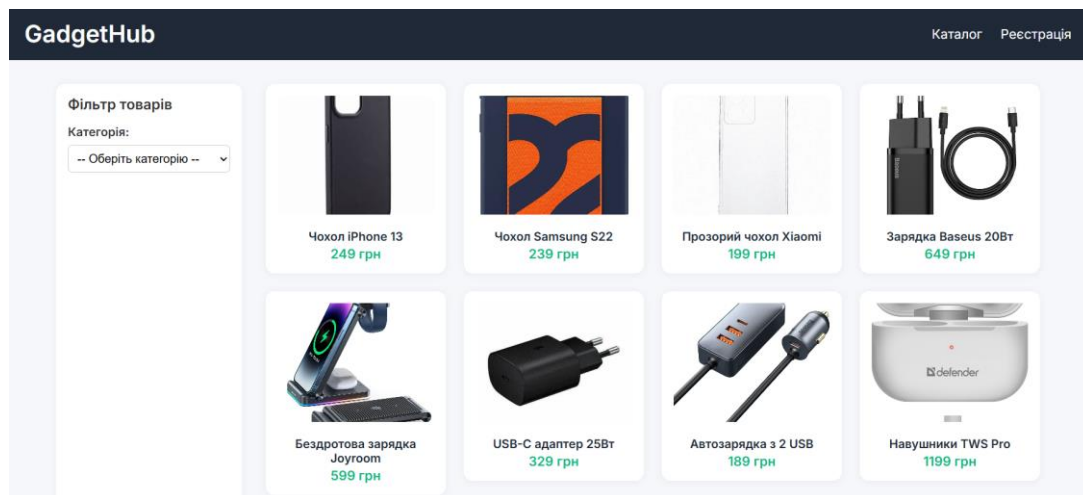
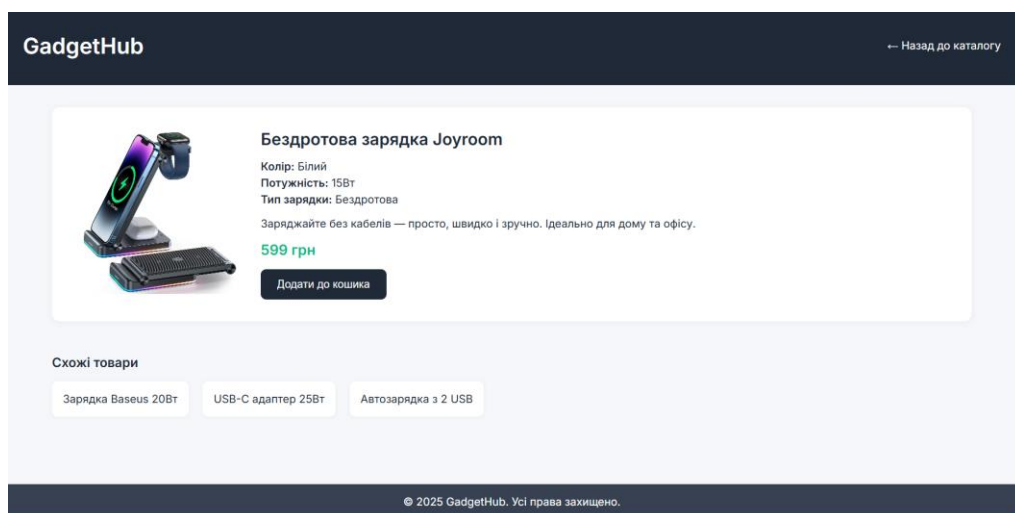
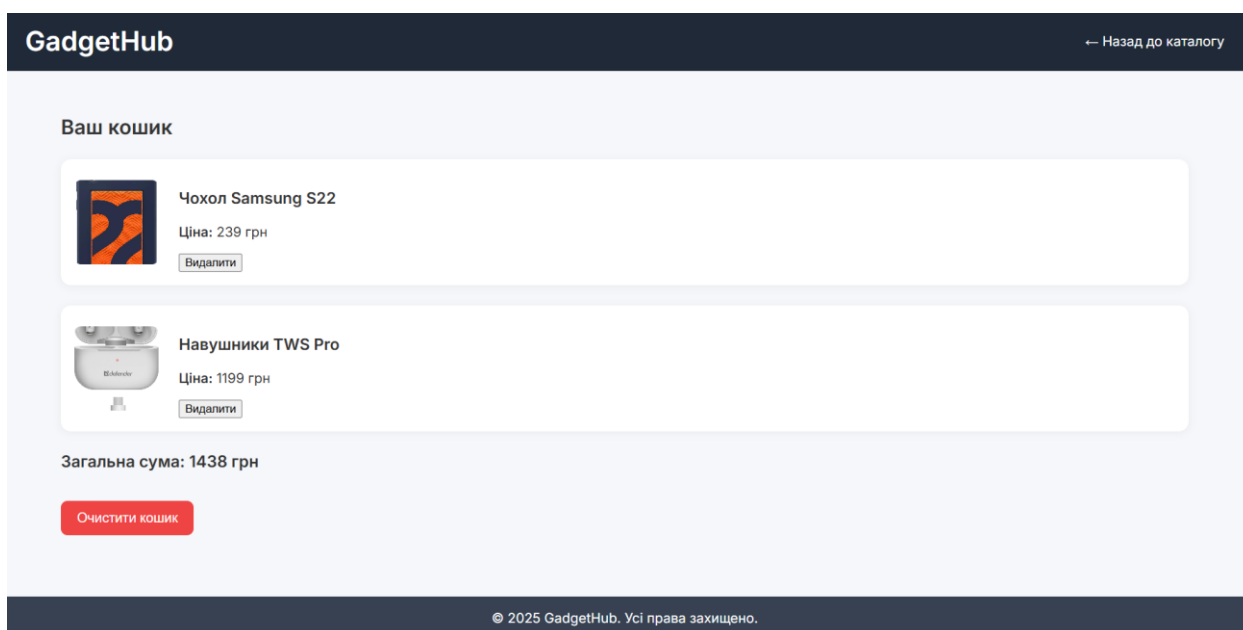


Рисунок 3.2 – Сторінка каталог веб сайту  
Джерело розроблено автором



*Рисунок 3.3 – Сторінка товару веб сайту  
Джерело розроблено автором*



*Рисунок 3.4 – Сторінка кошик веб сайту  
Джерело розроблено автором*

Таким чином, програмний продукт може використовуватись як основна платформа для ведення електронної комерції, підходить як для невеликого бізнесу, так і для особистих проєктів.

Перспективи розвитку та масштабування веб-сайту. Надалі платформа може бути розширена за рахунок впровадження нових функціональних можливостей:

1. Інтеграція платіжних систем – додавання можливості онлайн-оплати через PayPal, Stripe або банківські картки.
2. Впровадження AI-алгоритмів у рекомендаційну систему – машинне навчання може покращити точність персоналізованих пропозицій товарів.
3. Розширення товарного асортименту – адаптація сайту для продажу більш широкого спектра товарів.
4. Запуск мобільного застосунку – розробка мобільної версії на основі PWA (Progressive Web Application).

### **Висновки до розділу 3**

У цьому розділі було розглянуто процес реалізації програмного продукту, починаючи з вибору інструментів розробки, середовища виконання, бібліотек і технологій, які найбільш доцільно відповідали поставленим завданням. Надано фрагменти програмного коду, що демонструють основні механізми роботи системи, зокрема відображення товарів, обробку подій, роботу з базою даних.

Також було проведено функціональне і модульне тестування, результати якого засвідчили правильність роботи програмного продукту, відповідність заявленому функціоналу та ефективність окремих модулів. Це дало змогу переконатися в надійності та стабільності вебсайту.

Наведено приклади використання створеного веб-сайту, які демонструють його функціональність і адаптивність.

## ВИСНОВКИ

У процесі роботи було проведено аналіз предметної області та визначено основні функціональні вимоги до системи. На етапі проєктування побудовано UML-діаграми, які описують структуру, поведінку та архітектуру майбутнього програмного продукту. Для опису системи використано трирівневу клієнт-серверну архітектуру з використанням хмарної бази даних Firebase.

Реалізацію здійснено з використанням сучасних вебтехнологій: HTML, CSS, JavaScript, а також інструментів Firebase для збереження даних і авторизації користувачів. Сайт підтримує багатокритеріальну фільтрацію товарів, формування кошика та оформлення замовлення.

Було проведено функціональне і модульне тестування, результати якого підтвердили стабільність роботи сайту та відповідність заявленим вимогам. Розроблений продукт є готовим до використання в реальному середовищі, з можливістю подальшого вдосконалення, наприклад, через впровадження більш складної рекомендаційної системи чи додавання адміністративного кабінету.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Електронна комерція розширений опис. URL: [https://uk.wikipedia.org/wiki/Електронна\\_комерція](https://uk.wikipedia.org/wiki/Електронна_комерція)
2. Rozetka інтернет магазин. URL: <https://rozetka.com.ua/ua/>
3. Citrus інтернет магазин. URL: <https://www.ctrs.com.ua/>
4. Eldorado інтернет магазин. URL: <https://eldorado.ua/uk/>
5. Як працює SPA(Single Page Application) – Усе що потрібно знати.  
URL: [https://www.youtube.com/watch?v=7ySB3P3lZ\\_I](https://www.youtube.com/watch?v=7ySB3P3lZ_I)
6. Діаграма прецедентів. URL: [https://uk.wikipedia.org/wiki/Діаграма\\_прецедентів](https://uk.wikipedia.org/wiki/Діаграма_прецедентів)
7. Діаграма діяльності. URL: [https://uk.wikipedia.org/wiki/Діаграма\\_діяльності](https://uk.wikipedia.org/wiki/Діаграма_діяльності)
8. Діаграма послідовності. URL: [https://uk.wikipedia.org/wiki/Діаграма\\_послідовності](https://uk.wikipedia.org/wiki/Діаграма_послідовності)
9. Діаграмі комунікації. URL: [https://uk.wikipedia.org/wiki/Діаграма\\_комунікації](https://uk.wikipedia.org/wiki/Діаграма_комунікації)
10. Діаграма класів. URL: [https://uk.wikipedia.org/wiki/Діаграма\\_класів](https://uk.wikipedia.org/wiki/Діаграма_класів)
11. Діаграма об'єктів. URL: [https://uk.wikipedia.org/wiki/Діаграма\\_об%27єктів](https://uk.wikipedia.org/wiki/Діаграма_об%27єктів)
12. Діаграма пакетів. URL: [https://uk.wikipedia.org/wiki/Діаграма\\_пакетів](https://uk.wikipedia.org/wiki/Діаграма_пакетів)
13. База даних Firebase. URL: <https://firebase.google.com/>
14. Середовище розробки Visual studio. URL: <https://code.visualstudio.com/>
15. Мічківський С. Microsoft Office (Word, Excel, Outlook ...) : навч. посіб. / С. Мічківський, Д. Балдик, В. Головань; Східноукр. нац. ун-т ім. В. Даля, Аграр. ф-т. – Київ : [Вид-во Східноукр. нац. ун-т ім. В. Даля], 2023. – 128 с. – URL: <https://dspace.snu.edu.ua/handle/123456789/1723>
16. Шаповалов С.М., Мічківський С.М. Розробка рекомендаційної системи для підбору співрозмовника в соціальній мережі // XII Міжнародна

науково-технічна конференція «Проблеми Інформатизації» (м. Київ, 13 грудня 2018 року). – К.: Державний університет телекомунікацій, 2018. URL: [https://dut.edu.ua/uploads/1\\_1701\\_65845854.pdf](https://dut.edu.ua/uploads/1_1701_65845854.pdf)

17. Разиграєв В., Мічківський С. Рекомендаційна підсистема та багаторівнева архітектура в платформах інтернет-магазинів // Матеріали IV Міжнародної науково-практичної конференції «Бізнес-аналітика: моделі, інструменти та технології». 1-3 бер. 2023. – К.: НАУ, 2023. – 548 с. – С 357-359.

## ДОДАТОК А

### ФРАГМЕНТИ ЛІСТИНГУ

```

function loadRecommendations() {
  const category = localStorage.getItem("selectedCategory");
  const container = document.getElementById("recommended");
  container.innerHTML = "";

  if (!category || !productData[category]) {
    container.innerHTML = "<div class='product'>Немає рекомендацій – перейдіть до каталогу</div>";
    return;
  }

  productData[category].forEach(item => {
    const card = document.createElement("div");
    card.className = "product";
    card.style.cursor = "pointer";
    card.onclick = () => {
      window.location.href = item.link;
    };
    card.innerHTML = `
    
    <strong>${item.name}</strong><br><small>${item.price}</small>
    `;
    container.appendChild(card);
  });
}

```

*Рис А.1 – Фрагменти лістингу рекомендаційної системи*

*Джерело - розроблено автором*

```

function applyFilters() {
  const category = categorySelect.value;
  const inputs = filterContainer.querySelectorAll("input");
  const criteria = {};

  inputs.forEach(input => {
    if (input.value.trim() !== "") {
      criteria[input.dataset.filterKey] = input.value.trim().toLowerCase();
    }
  });

  productList.innerHTML = "";

  products
    .filter(p => p.category === category)
    .filter(p => {
      return Object.keys(criteria).every(key =>
        p[key]?.toLowerCase().includes(criteria[key])
      );
    })
    .forEach(p => {
      const div = document.createElement("div");
      div.textContent = p.name;
      productList.appendChild(div);
    });
}

```

*Рис А.2 – Фрагмент лістингу фільтрації*

*Джерело – розроблено автором*