

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: «Модель e-commerce системи для продажу одягу з використанням сучасних інформаційних технологій»

на здобуття освітнього ступеня магістра  
зі спеціальності F3 Комп'ютерні науки  
(код, найменування спеціальності)  
освітньо-професійної програми Комп'ютерні науки  
(назва)

*Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело*

\_\_\_\_\_ Аліна КУЗІНА  
(підпис) (Ім'я, ПРИЗВИЩЕ здобувача)

Виконав:  
здобувач вищої освіти  
група КНДМ-62

Аліна КУЗІНА

Керівник:  
науковий ступінь,  
вчене звання

Юрій КАТКОВ  
Д.Т.Н., доцент

Рецензент:  
науковий ступінь,  
вчене звання

\_\_\_\_\_  
(Ім'я, ПРИЗВИЩЕ)

**Київ 2025**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

**Навчально-науковий інститут інформаційних технологій**

Кафедра Комп'ютерних наук  
Ступінь вищої освіти Магістр  
Спеціальність F3 Комп'ютерні науки  
Освітньо-професійна програма Комп'ютерні науки

**ЗАТВЕРДЖУЮ**

Завідувач кафедру Комп'ютерних наук

\_\_\_\_\_ Віктор ВИШНІВСЬКИЙ  
« \_\_\_\_\_ » \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

\_\_\_\_\_ Кузіної Аліні Ярославівні

*(прізвище, ім'я, по батькові здобувача)*

1. Тема кваліфікаційної роботи: «Модель e-commerce системи для продажу одягу з використанням сучасних інформаційних технологій»

керівник кваліфікаційної роботи Юрій КАТКОВ д.т.н., доцент,

*(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)*

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2025р. № 467

2. Строк подання кваліфікаційної роботи «26» грудня 2025р.

3. Вихідні дані до кваліфікаційної роботи:

3.1. Науково-технічна література з питань, пов'язаних із дослідженням e-commerce платформи для продажу одягу з використанням сучасних технологій;

3.2. Практичний досвід застосування e-commerce платформи для продажу одягу з використанням сучасних технологій;

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Аналіз ринку та аспектів створення e-commerce платформи для продажу одягу з використанням сучасних технологій.

4.2 Дослідження застосування інтелектуальних нових технологій для створення e-commerce платформи для продажу одягу.

4.3 Розробка практичних рекомендацій щодо впровадження інтелектуальних нових технологій для створення e-commerce платформи для продажу одягу.

5. Перелік графічного матеріалу: *презентація*

1. Тема дипломної роботи
2. Мета, об'єкт, предмет дипломної роботи
3. Постановка завдання дипломної роботи
4. Результати аналізу ринку та аспектів створення e-commerce платформи для продажу одягу з використанням сучасних технологій.
5. Результати дослідження застосування інтелектуальних технологій для створення e-commerce платформи для продажу одягу.
6. Практичні рекомендації щодо впровадження інтелектуальних технологій для створення e-commerce платформи для продажу одягу
7. Висновки
8. Апробація

6. Дата видачі завдання «15» вересня 2025 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури по темі «Створення e-commerce платформи для продажу одягу з використанням сучасних технологій»	15.09-05.10.24	Виконане
2	Аналіз специфіки і характеристик основ функціонування e-commerce платформи для продажу одягу	06.10-18.10.24	Виконане
3	Дослідження застосування інтелектуальних технологій для створення e-commerce платформи для продажу одягу	19.10-31.10.24	Виконане
4	Розробка практичних рекомендацій щодо впровадження інтелектуальних технологій для створення e-commerce платформи для продажу одягу	01.11-25.11.24	Виконане
5	Основні розділи.	27.11-03.12.24	Виконане
6	Розробка обов'язкових матеріалів.	04.12-10.12.24	Виконане
7	Попередній захист роботи.	11.12-14.12.24	Виконане
8	Пред'явлення роботи в деканат.	15.12-29.12.24	Виконане

Здобувач вищої освіти

Керівник  
кваліфікаційної роботи

(підпис)

(підпис)

Аліна КУЗІНА

(Ім'я, ПРІЗВИЩЕ)

Юрій КАТКОВ

(Ім'я, ПРІЗВИЩЕ)





## РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 113 с., 7 рис., 19 табл., 102 джерел.

*Мета роботи:* Підвищити ефективність застосування інтелектуальних нових технологій для створення Fashion E-commerce платформи.

*Об'єкт дослідження:* Процес застосування інтелектуальних нових технологій для створення Fashion E-commerce платформи.

*Предмет дослідження:* Інтелектуальні нові технології для створення Fashion E-commerce платформи.

*Короткий зміст роботи:*

*Наукове завдання* – обґрунтувати доцільність застосування інтелектуальних нових технологій для створення Fashion E-commerce платформи.

*Метод дослідження.* Метод дослідження заснований на відомих методах системного підходу, а саме: аналітично-розрахунковий, теоретичного аналізу з використанням моделювання, реконструювання і типологізації на основі аналізу емпіричних даних.

*Актуальність теми* – У сфері E-commerce (електронної комерції в галузі моди) особливу увагу приділено впровадженню цифрових технологій: штучного інтелекту, машинного навчання, Big Data, IoT/RFID, Blockchain та Smart Contracts, роботизації та автоматизації, що є вкрай актуальним.

*Галузь використання* – торгівля, надання послуг.

*Ключові слова:* Fashion E-commerce, інтелектуальні технології.

## ABSTRACT

**Text section of the qualification thesis for obtaining the Master's degree:** 113 pages, 7 figures, 19 tables, 102 sources.

**Research Objective:** To improve the effectiveness of applying intelligent emerging technologies for the development of a Fashion E-commerce platform.

**Object of Study:** The process of applying intelligent emerging technologies for the development of a Fashion E-commerce platform.

**Subject of Study:** Intelligent emerging technologies for the development of a Fashion E-commerce platform.

### **Brief Description of the Work:**

The scientific task is to justify the feasibility of applying intelligent emerging technologies for the development of a Fashion E-commerce platform.

### **Research Methodology:**

The research methodology is based on well-established systems approach methods, namely: analytical and computational methods, theoretical analysis using modeling, reconstruction, and typologization based on empirical data analysis.

### **Relevance of the Topic:**

In the field of Fashion E-commerce, special attention is given to the implementation of digital technologies such as artificial intelligence, machine learning, Big Data, IoT/RFID, Blockchain and Smart Contracts, robotics, and automation, which is of high current relevance.

### **Field of Application:**

Trade and service provision.

**Keywords:** Fashion E-commerce, intelligent technologies.

## ЗМІСТ

	Ст.
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....	10
ВСТУП.....	11
1 АНАЛІЗ РИНКУ ТА АСПЕКТІВ СТВОРЕННЯ E-COMMERCE ПЛАТФОРМИ ДЛЯ ПРОДАЖУ ОДЯГУ З ВИКОРИСТАННЯМ СУЧАСНИХ ТЕХНОЛОГІЙ.....	13
1.1 Поняття, сутність та основні типи Fashion E-commerce .....	13
1.2 Особливості ринку онлайн-продажу одягу .....	16
1.2.1 Огляд ринку e-commerce: глобальні та локальні тенденції....	16
1.2.2 Особливості продажу одягу онлайн.....	20
1.3 Наслідки застосування сучасних нових інформаційних технологій у Fashion E-commerce .....	23
2 ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ ІНТЕЛЕКТУАЛЬНИХ НОВИХ ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ E-COMMERCE ПЛАТФОРМИ ДЛЯ ПРОДАЖУ ОДЯГУ.....	31
2.1 Дослідження сучасних технологій для розробки e-commerce платформ:.....	31
2.1.1 Технології для розробки Frontend.....	31
2.1.2 Технології для розробки Backend.....	36
2.1.3 Технології для розробки бази даних.....	40
2.1.4 Технології для розробки хмарних сервісів.....	43
2.2 Обґрунтування вибору технологій для розробки e-commerce платформ.....	48
2.2.1 Мови програмування для e-commerce платформ .....	48
2.2.2 Фреймворки для e-commerce платформ .....	51
2.2.3 Бази даних для e-commerce платформ.....	54
2.2.4 Інструменти для інтеграції e-commerce платформ з платіжними системами .....	57

2.3	Визначення архітектури e-commerce платформи .....	61
2.3.1	Вибір архітектурного підходу до побудови e-commerce платформ.....	61
2.3.2	Функціональні вимоги .....	63
2.3.3	Нефункціональні вимоги до e-commerce платформи .....	66
3	РОЗРОБКА ПРАКТИЧНИХ РЕКОМЕНДАЦІЙ ЩОДО ВПРОВАДЖЕННЯ ІНТЕЛЕКТУАЛЬНИХ НОВИХ ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ E-COMMERCE ПЛАТФОРМИ ДЛЯ ПРОДАЖУ ОДЯГУ...	69
3.1	Загальні рекомендації щодо впровадження видів тестування e-commerce платформи.....	69
3.2	Практичні рекомендації щодо використання інструментів для тестування: Selenium, Jest, Postman.....	73
3.3	Практичні рекомендації щодо розробки інтерфейсу користувача (UI/UX дизайн).....	76
3.4	Практичні рекомендації щодо реалізація функціональності .....	78
3.5	Практичні рекомендації щодо Інтеграції зі сторонніми сервісами	81
	ВИСНОВКИ.....	85
	ПЕРЕЛІК ПОСИЛАНЬ.....	90
	Презентація .....	102

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- AI – Artificial Intelligence (Штучний інтелект);
- AR – доповнена реальність
- CRM – Customer Relationship Management (Управління відносинами з клієнтами);
- IT – Інформаційні технології
- NLP – Natural language processing (Натуральна обробка мови);
- SMS – Short Message Service
- TMS – Transportation Management Systems;
- VR – віртуальна реальність
- ШІ – штучний інтелект;

## ВСТУП

В атестаційній роботі магістра розглядається актуальна проблема застосування інтелектуальних нових технологій для створення e-commerce платформи для продажу одягу (Fashion E-commerce платформи - електронної комерції в галузі моди).

*Обґрунтування вибору теми та її актуальність.* E-commerce – це галузь економіки, що охоплює онлайн-продаж товарів та послуг. Інформаційні технології (ІТ) суттєво трансформують Fashion E-commerce, впливаючи на всі аспекти бізнесу - від дизайну та виробництва до маркетингу та логістики, а саме: персоналізацію та створення рекомендаційної системи за допомогою штучного інтелекту (AI); забезпечують створення віртуальної (VR) та доповненої реальності (AR); впроваджують цифровий сертифікат права власності (NFT) на цифровий одяг; створюють нові канали монетизації та оптимізують ланцюги поставок; забезпечують генеративний дизайн, 3D-рендеринг та швидке виробництво; впроваджують маркетинг на основі аналізу даних з соцмереж; забезпечують боротьбу з контрафактом; покращають управління поверненнями та багато іншого. Тобто ІТ-технології роблять Fashion E-commerce швидшим, індивідуалізованим та ефективним. Ті бренди, які інвестують у AI, AR та аналітику даних, отримують конкурентні переваги за рахунок гнучкості та точної відповіді на запити клієнтів. Але індустрія Fashion E-commerce під час впровадження інформаційних технологій стикається з унікальними викликами в управлінні запасами та логістикою через швидкі зміни тенденцій, сезонні коливання та високий рівень повернень. Тому дослідження впровадження цифрових технологій: штучного інтелекту, машинного навчання, Big Data, IoT/RFID, Blockchain та Smart Contracts, роботизації та автоматизації є вкрай актуальним.

*Формулювання завдання.* Fashion E-commerce перетворилася на один із найбільш динамічних і конкурентоспроможних секторів онлайн-торгівлі. Оскільки

цифрові платформи стають основною точкою взаємодії між бренда споживачами, керування внутрішніми процесами стало вирішальним для успіху. Сучасні клієнти очікують не лише різноманітності продуктів і доступності, але й зручність покупок, швидкої доставки та легкого повернення. Ці очікування створюють величезний тиск на компанії E-commerce з метою оновлення традиційних систем. Завдяки впровадженню ІТ-технологій компанії можуть трансформувати Fashion E-commerce, впливаючи на всі аспекти бізнесу - від дизайну та виробництва до маркетингу та логістики та надавати незмінну якість послуг. Зосередження на пошуку рішень, що дозволяють забезпечити гнучкість, масштабованість і економічну ефективність процесів Fashion E-commerce, швидко змінювані тренди та вимоги до обслуговування клієнтів є актуальною задачею..

*Ступінь вивчення проблеми.* Проблема використання інтелектуальних нових технологій для створення Fashion E-commerce платформ є частково вивченою. Ступінь проблеми використання інтелектуальних нових технологій для створення Fashion E-commerce платформ розглядається в багатьох роботах. Існує міцна основа для впровадження багатьох з цих технологій, проте постійно з'являються нові виклики та можливості, що вимагають подальших глибоких досліджень та розробок. Зокрема, області, які потребують найбільшої уваги, включають: подальше вдосконалення віртуальних примірочних, підвищення точності прогнозування трендів, поглиблену персоналізацію на основі глибинного аналізу стилю та емоцій, а також ефективну інтеграцію всіх цих елементів для створення дійсно безшовного та інтелектуального досвіду покупки.

*Специфіка джерельної бази.* Аналіз джерельної бази свідчить про її неоднорідність, що зумовлює необхідність систематизації джерел за визначеними критеріями, зокрема за рівнем важливості, якості, унікальності тощо, з метою виявлення наявних проблем і недоліків.

*Об'єкт дослідження:* Процес застосування інтелектуальних нових технологій для створення Fashion E-commerce платформи.

*Предмет дослідження:* Інтелектуальні нові технології для створення Fashion E-commerce платформи.

*Мета роботи:* Підвищити ефективність застосування інтелектуальних нових технологій для створення Fashion E-commerce платформи.

*Метод дослідження.* У магістерській роботі використано системний підхід та сукупність загальнонаукових і спеціалізованих методів дослідження. Зокрема, застосовано аналітично-розрахунковий метод для оцінювання ефективності функціонування e-commerce системи та її ключових показників. Метод теоретичного аналізу використано для вивчення сучасних інформаційних технологій, архітектурних рішень і моделей електронної комерції у сфері продажу одягу. Методи моделювання та проєктування застосовано для розроблення структурно-функціональної моделі e-commerce системи, а методи реконструювання й типологізації — для класифікації бізнес-процесів і компонентів системи на основі аналізу емпіричних даних та практичних прикладів.

*Наукове завдання* – обґрунтувати доцільність застосування інтелектуальних нових технологій для створення Fashion E-commerce платформи.

*Завдання роботи:*

1. Провести аналіз ринку та аспектів створення e-commerce платформи для продажу одягу з використанням сучасних технологій.
2. Обґрунтувати доцільність застосування інтелектуальних нових технологій для створення e-commerce платформи для продажу одягу.
3. Розробити практичні рекомендації щодо впровадження інтелектуальних нових технологій для створення e-commerce платформи для продажу одягу.

*Результати дослідження.* Підвищення ефективності застосування інтелектуальних нових технологій для створення e-commerce платформи для продажу одягу шляхом оптимізації процесу застосування інтелектуальних нових технологій для створення Fashion E-commerce платформи.

*Наукова новизна.* Наукова новизна цього дослідження полягає в підвищенні ефективності застосування інтелектуальних нових технологій для створення Fashion E-commerce платформи.

*Практична значущість результатів дослідження.* Практична значущість полягає в розробці практичних рекомендацій щодо застосування інтелектуальних нових технологій для створення Fashion E-commerce платформи..

*Апробація результатів* надається в 1 статті, 2 тезисах в рецензованих наукових журналах і виданнях.

В статті:

Kuzina.A. (2025) Prioritizing disruptive information technologies in Fashion E-Commerce using OSINT / International Science Journal of Engineering & Agriculture, Vol. 4 No. 5 (2025), p. 62-76/ DOI: <https://doi.org/10.46299/j.isjea.20250405> ISSN: 2720-6319, <https://isg-journal.com/isjea/article/view/1102> (date of access: 02.09.2025).

В тезах:

1. Kuzina A. Ya. E-commerce in the Fashion Industry: Digital Transformation, Challenges and Development Prospects (2025) стор 201./ Синергія інновацій: політика, економіка та менеджмент в цифровому світі: збірник матеріалів Міжнародної науково-практичної конференції (м. Київ, 18 квітня 2025 р.) [Електронне видання]. Київ : ДУІКТ, 2025. 364 с. [https://duikt.edu.ua/uploads/p\\_2779\\_26171712.pdf](https://duikt.edu.ua/uploads/p_2779_26171712.pdf)
2. Кузіна А.Я. ІНТЕГРАЦІЯ ІОТ-ТЕХНОЛОГІЙ В ІНТЕРНЕТ-МАГАЗИНІ ОДЯГУ ДЛЯ ПОКРАЩЕННЯ КОРИСТУВАЦЬКОГО ДОСВІДУ// VI Науково-технічна конференція «Сучасний стан та перспективи розвитку ІоТ». Збірник тез 15 квітня 2025. – К.: ДУІКТ, ст. 60-63. [https://duikt.edu.ua/uploads/p\\_2779\\_40288420.pdf](https://duikt.edu.ua/uploads/p_2779_40288420.pdf)

# 1 АНАЛІЗ РИНКУ ТА АСПЕКТІВ СТВОРЕННЯ E-COMMERCE ПЛАТФОРМИ ДЛЯ ПРОДАЖУ ОДЯГУ З ВИКОРИСТАННЯМ СУЧАСНИХ ТЕХНОЛОГІЙ

## 1.1 Поняття, сутність та основні типи Fashion E-commerce платформ

Електронна комерція (e-commerce) охоплює процеси купівлі та продажу товарів і послуг через мережу Інтернет і становить окремий сегмент цифрової економіки, у межах якого здійснюються транзакції, логістичне обслуговування, маркетингові комунікації та післяпродажна підтримка [1; 2]. Під e-commerce-платформою зазвичай розуміють програмно-технологічний комплекс, що забезпечує створення та експлуатацію інтернет-магазинів і маркетплейсів, включаючи засоби управління каталогом і запасами, оброблення замовлень, інтеграцію з платіжними провайдерами, модулі аналітики й персоналізації, а також інструменти взаємодії з клієнтами на всіх етапах воронки продажів [1]. Така платформа виконує роль операційного ядра цифрового ритейлу, поєднуючи зовнішні канали збуту з бекенд-службами та логістичною інфраструктурою.

Модний сегмент електронної комерції має специфіку, зумовлену високою варіативністю асортименту, частою змінюваністю колекцій та підвищеною часткою повернень, що ускладнює управління запасами, прогнозування попиту та планування «останньої милі» [2]. Це потребує гнучких інтеграцій із службами доставки та централізованого регулювання політик повернення. Важливим напрямом розвитку платформ є впровадження інтелектуальних технологій: систем рекомендацій і персоналізації на основі алгоритмів машинного навчання, візуального пошуку та комп'ютерного зору, а також AR/VR-модулів для віртуальної примірки, що підвищують інформованість користувача та впливають на конверсійні показники й частку повернень [2; 3]. У мобільному контексті особливого значення набувають безшовні сценарії взаємодії й мінімізація «тертя» в інтерфейсах, оскільки

мобільні пристрої формують істотну частку трафіку та транзакцій у fashion e-commerce [4].

Ступінь впровадження окремих інтелектуальних компонентів у модній електронній комерції є нерівномірним. Рекомендаційні системи та персоналізовані добірки товарів належать до найбільш досліджених і комерційно відпрацьованих напрямів; актуальні завдання пов'язані з підвищенням релевантності, урахуванням індивідуальних стилістичних уподобань і скороченням часу до першого справді відповідного результату [2; 7]. Рішення на основі AR/VR для віртуальної примірки перебувають у фазі активного розвитку: продемонстровано їхній позитивний вплив на залучення та впевненість покупця, однак триває робота над точністю візуалізації, добором розміру й зручністю інтеграції у типові сценарії перегляду та оформлення замовлення [3]. Технології комп'ютерного зору для розпізнавання елементів гардероба та пошуку схожих позицій за зображенням уже застосовуються провідними гравцями ринку, хоча підвищення якості розпізнавання текстур і фасонів залишається відкритою задачею [3; 5]. Використання чат-ботів і віртуальних асистентів для підтримки під час вибору товару та навігації каталогу поширюється паралельно з удосконаленням розуміння природної мови та інтеграцією з бекенд-процесами, зокрема з управлінням замовленнями й післяпродажним сервісом [4; 7]. На операційній стороні ключовими залишаються модулі прогнозування попиту, оптимізації запасів та роботизованої логістики складу, які дають змогу зменшити операційні витрати та час виконання замовлення у високосезонні періоди [1; 6].

Попри значний прогрес, дослідження виокремлюють низку викликів, релевантних для модної електронної комерції. Серед них – залежність якості алгоритмів від повноти та репрезентативності даних, складність побудови безперервних інтеграцій між різнорідними інтелектуальними компонентами в межах єдиної платформи, потреба у формальних гарантіях приватності та дотриманні етичних стандартів під час оброблення користувацьких даних, вимоги до UX щодо прозорості й керованості на мобільних пристроях, а також питання масштабованості й економічної доцільності доведення інновацій до промислової експлуатації [2; 7].

Сутнісна роль e-commerce платформи полягає в автоматизації наскрізних процесів онлайн-торгівлі з урахуванням чотирьох базових властивостей: зручності (наявність зрозумілих інтерфейсних патернів для покупців і адміністраторів), масштабованості (еластичне оброблення зростання трафіку й асортименту), безпеки (захист персональних даних і платіжних транзакцій) та інтегрованості (сумісність із CRM, ERP, логістичними та маркетинговими сервісами через стандартизовані API) [1; 7]. У модному сегменті додатково наголошується на потребі керування складними довідниками атрибутів товарів, які залежать від сезону, колекції, розміру, кольору та варіантів SKU, а також на підтримці політик повернення, чутливих до категорії продукції й поточного стану складу [2; 9]. Класифікація платформ зазвичай здійснюється за трьома вимірами: за способом розгортання, за типом бізнес-моделі та за форматом роботи. Узагальнення для основних типів наведено нижче (табл. 1.1)

Таблиця 1.1 - Основні типи e-commerce платформ [1; 2]

Ознака	За способом розгортання	
Тип	Опис	Приклади
SaaS (хмарні)	Готові рішення на основі підписки, без необхідності самостійного хостингу.	Shopify, BigCommerce, Wix
Open-Source	Платформи з відкритим кодом, що потребують налаштування та хостингу.	Magento, WooCommerce, OpenCart
Headless	Відокремлений фронтенд (наприклад, на React) та бекенд (API-орієнтований).	CommerceTools, Saleor
Ознака	За бізнес-моделлю	
B2C (Business-to-Consumer)	Продаж товарів кінцевим споживачам.	Amazon, Rozetka, Nike
B2B (Business-to-Business)	Оптові продажі між компаніями.	Alibaba, Prom.ua
C2C (Consumer-to-Consumer)	Платформи для продажу між приватними особами.	OLX, eBay, Facebook Marketplace
D2C (Direct-to-Consumer)	Виробники продають без посередників.	Tesla, Warby Parker
Ознака	За форматом роботи	
Онлайн-магазин	Класичний інтернет-магазин з власним асортиментом.	Allbirds, Zara
Маркетплейс	Багато продавців на одній платформі.	Amazon, Etsy, Allegro
Підписка (Subscription)	Регулярні поставки товарів (бокси, сервіси).	Glossybox, Dollar Shave Club
Dropshipping	Продаж без власних складів (товар відправляє постачальник).	Oberlo, Spocket

SaaS, Open-Source, headless-рішення, а також моделі B2C, B2B, C2C, D2C і різні формати роботи (класичний інтернет-магазин, маркетплейс, підписка, дропшипінг) задають можливі конфігурації технологічної та бізнес-архітектури платформи. У контексті fashion-ритейлу вибір конкретної комбінації залежить від масштабу бізнесу, цільових ринків, рівня контролю над брендом і глибини інтеграції з логістичними та маркетинговими сервісами.

Таким чином, e-commerce платформи виступають технологічною основою модного ритейлу, забезпечуючи інтегроване управління каталогом, замовленнями, оплатами та логістикою, а також створюючи умови для впровадження інтелектуальних сервісів персоналізації й візуальної взаємодії, що визначають конкурентоздатність у цифровому середовищі [2; 7].

## **1.2 Особливості ринку онлайн-продажу одягу**

**1.2.1 Огляд ринку E-commerce: глобальні та локальні тенденції.** Світовий ринок електронної комерції у 2024–2025 роках зберігає стійку тенденцію зростання під впливом розвитку інформаційних та інтелектуальних технологій, змін у споживчій поведінці та структурних перетворень у глобальній економіці. Узагальнений тренд полягає в посиленні персоналізації, підвищенні технологічної складності цифрових рішень і подальшій глобалізації каналів збуту. Компанії, які зволікають з упровадженням цифрових інновацій та моделей ухвалення рішень на основі даних, втрачають конкурентні позиції порівняно з гравцями, що раніше інтегрували такі підходи в операційну діяльність. На рівні щоденних процесів це означає зростання ролі аналітики в управлінні асортиментом, ціноутворенні та логістиці «останньої милі», а також переорієнтацію інтерфейсів на мобільні сценарії взаємодії [2; 7].

Глобальні тенденції 2024–2025 років. Галузеві огляди фіксують подальше зростання обсягів онлайн-продажів та посилення регіональної диференціації: найвищі темпи приросту демонструють ринки Азії, тоді як у більш зрілих економіках конкуренція зміщується до якості сервісу, швидкодії інтерфейсів і

гнучких промо-стратегій. Прогнози вказують на суттєве збільшення глобального обсягу e-commerce до 2027 року із паралельним зростанням частки мобільних транзакцій та крос-бордерних покупок [7].

Інтелектуальні технології стають наскрізним елементом ланцюга створення вартості: рекомендаційні системи, персоналізація на основі поведінкових та контентних сигналів, комп'ютерний зір, візуальний пошук, а також AR/VR-модулі віртуальної примірки демонструють помітний вплив на конверсію й частку повернень, що особливо важливо для категорій із домінуванням візуальної оцінки, насамперед у fashion-сегменті. Мобільний ритейл стимулює масове впровадження принципів *mobile-first* і прогресивних вебзастосунків, які зменшують «тертя» у взаємодії та підвищують частку завершених замовлень [1; 7].

У сфері виконання замовлень продовжується автоматизація складів, розвиток роботизованих комплексів сортування та впровадження стандартизованих процесів обробки повернень, що дає змогу знижувати витрати й підтримувати стабільний рівень сервісу за пікових навантажень. Зростає роль централізованих рішень для керування поверненнями й перерозподілом товарних потоків, зокрема на базі спеціалізованих сервісів та роботизованих сортувальних систем [3; 8]. Стратегічне управління, орієнтоване на дані – від мерчандайзингу в реальному часі до алгоритмічного ціноутворення й планування запасів, – розглядається як ключове джерело приросту вартості у моді та сегменті розкоші [7].

Технологічні драйвери трансформації. До інтерфейсних інновацій належать персоналізовані рекомендації, чат-боти й віртуальні асистенти, які зменшують інформаційне навантаження на користувача та прискорюють пошук релевантних товарів; у fashion-сегменті це тісно пов'язано з якістю візуального контенту та точністю розмірних підказок. Засоби доповненої та віртуальної реальності для примірки одягу підвищують поінформованість і впевненість покупця перед оплатою, хоча питання масштабованості таких рішень і їхньої інтеграції в стандартні сценарії перегляду та оформлення замовлення залишаються актуальними [2; 7].

На бекенд-рівні та в логістиці автоматизація складів, роботизоване сортування, інтегровані системи управління запасами й поверненнями розглядаються як необхідна передумова утримання конкурентної собівартості виконання замовлень і скорочення циклів обробки в умовах вираженої сезонності попиту. Паралельно розвиваються голосові інтерфейси та пошук із використанням природномовних запитів, що вимагає адаптації контенту, структури каталогів та SEO-стратегій під нові сценарії взаємодії [3; 6].

Еволюція бізнес-моделей. Ринок демонструє розширення моделей підписки з рекурентними доходами, коробкових сервісів і регулярних поставок у сегментах повсякденного попиту. Посилюється тренд на прямий продаж виробника кінцевому споживачеві (D2C), що дає змогу зберігати контроль над брендом, клієнтськими даними та маржинальністю.

Паралельно розвиваються моделі ресейлу й продажу відновлених товарів, які відповідають екологічним очікуванням та ідеям циркулярної економіки; у модній індустрії вони підтримуються культурою вторинного використання та апсайклінгу. Соціальна комерція й *live*-шопінг інтегрують контент та транзакції в єдиному середовищі, скорочуючи шлях до покупки та перерозподіляючи частину конверсій із класичних вебсайтів на платформи соціальних медіа. Подальший розвиток екосистем маркетплейсів, які поєднують логістику, платежі, маркетинг і аналітику в уніфікованих операційних моделях, підвищує ефективність масштабування бізнесу [7; 10].

Зміни у споживчій поведінці та виконанні. Домінування мобільних сесій формує підвищені вимоги до простоти навігації та фільтрації, швидкодії інтерфейсів і прозорості умов доставки та повернень. Активно розширюється мережа поштоматів і локальних пунктів видачі, упроваджуються безконтактні сценарії отримання замовлень; окремі гравці експериментують із безпілотними рішеннями та мікроскладами в густонаселених зонах, що дає змогу скорочувати час доставки та стабілізувати рівень сервісу.

Зворотна логістика дедалі частіше розглядається як окремий об'єкт аналітики: використання даних про причини повернень дозволяє мінімізувати втрати вартості, підвищувати оборотність запасів[2; 6].

Локальні тенденції: Україна та Східна Європа (2024–2025 роки). Регіональний сегмент e-commerce адаптується до зовнішніх шоків, водночас зберігаючи курс на розширення онлайн-каналів і впровадження омніканальних моделей. (табл..1.2).

Таблиця 1.2 - Аналіз локальних тенденції по категоріях за 2024-2025рокі в Україні [2; 7].

Категорія	Приклади	Чому зростає?
Продукти харчування	Zakaz.ua, Metro Market Online	Попит на доставку їжі під час блекаутів
Медикаменти та догляд	Tabletki.ua, Liki24	Дефіцит ліків у аптеках
Електроніка та powerbank	Rozetka, Citrus	Часті відключення світла
Військовий та кемпінговий одяг	Sturm, Prometheus	Запити від ЗСУ та волонтерів
Український крафт та handmade	Crafta.ua, Made in UA	Підтримка локальних виробників

Для українського ринку характерні переважання мобільних сценаріїв покупки, розвинена інфраструктура самовивозу й поштоматів, стандартизація процесів повернення, підвищені вимоги до прозорості відстеження доставки та рівня сервісних угод, а також підтримка локальних брендів і крафтових виробників. Платформи переважно спираються на промислові інтеграції з логістичними операторами та API-рішення для управління виконанням замовлень і зворотною логістикою. У fashion-категорії акцент зміщується до якості візуалізації, точності розмірних рекомендацій і швидкодії мобільних інтерфейсів, які формують більшу частину трафіку [7] .

Платіжні та маркетингові особливості локального ринку. Для українського сегмента типовою є множинність платіжних сценаріїв: оплата при отриманні, різні форми розстрочки, моделі «купити зараз – сплатити пізніше», а також окремі крос-

бордерні рішення, що доповнюють національну платіжну інфраструктуру. З маркетингового погляду помітне поширення швидких каналів запуску для малого та середнього бізнесу – продажі через месенджери, соціальні мережі та маркетплейси з локалізованим контентом, який спирається на національну ідентичність та ціннісні наративи. Такі практики узгоджуються з підходами цифрового маркетингу до сегментації аудиторії, автоматизації контактів і побудови тригерних ланцюжків у воронці продажів [7].

Таким чином, розвиток e-commerce у 2024–2025 роках визначається поєднанням персоналізації та інтелектуальних компонентів на стороні вітрини, автоматизації виконання й зворотної логістики, а також мобільно-першої архітектури взаємодії. Для платформ і брендів це означає пріоритет інвестицій у дані й аналітику, логістичну інтегрованість і якість користувацького досвіду, що безпосередньо корелює з їхньою конкурентоспроможністю на глобальних і локальних ринках [7].

### **1.2.2 Особливості продажу одягу онлайн (Fashion E-Commerce).**

Узагальнене поле конкуренції у fashion e-commerce формують багатобрендові мас-маркет платформи (на кшталт ASOS, Zalando), люксові агрегатори та бутикові ритейлери (Farfetch, Mytheresa), а також національні та регіональні рішення, включно з українськими аналогами. Оцінювання їхньої ефективності доцільно здійснювати за такими групами критеріїв: ширина та керованість асортименту, якість користувацького досвіду, рівень персоналізації та пошуку, логістика виконання й зворотної логістики, інтеграція інтелектуальних технологій, а також економіка клієнтського залучення і довгострокової лояльності [2].

Переваги. По-перше, провідні платформи забезпечують широку доступність товарної пропозиції завдяки мережам постачальників і стандартизованому онбордингу брендів та бутиків, що підтримує наявність «довгого хвоста» позицій і підвищує імовірність релевантної знахідки для різних бюджетів і стилістичних уподобань.

По-друге, інтерфейси характеризуються зрілою інформаційною архітектурою: багатовимірні фільтри, стабільні категоріальні таксономії, картки товарів з

високоякісними фото і відео, розширені розмірні гіді та контекстні підказки, які знижують когнітивне навантаження й зменшують тертя під час вибору.

По-третє, персоналізація та рекомендаційні алгоритми підсилюють відкриття асортименту через блоки суміжних і доповнювальних товарів, динамічні вітрини й індивідуальні промопропозиції.

По-четверте, мобільні застосунки і прогресивні вебзастосунки забезпечують високу швидкодію, офлайн-стійкість і канали ретаргетингу (push-повідомлення, персональні добірки), у тому числі з використанням візуального пошуку.

По-п'яте, стандартизовані політики повернення та розгалужені мережі пунктів приймання знижують бар'єр першої покупки та підтримують повторні транзакції. Нарешті, інтеграція з платіжними сервісами, антифрод-модулями та логістичними партнерами забезпечує передбачуваність виконання замовлення та прозорість відстеження [1; 2].

Недоліки. Ключовим обмеженням залишається відсутність фізичної взаємодії з товаром: невизначеність щодо посадки, фактури та тактильних властивостей матеріалу зумовлює підвищену частку повернень і витрати зворотної логістики. Навіть за наявності AR-/VR-модулів рівень реалізму та масштабованість таких рішень є неоднорідними, а покриття брендів і категорій — неповним. Додатковою проблемою виступають «втома від вибору» та навігаційні складнощі на каталогах із сотнями тисяч позицій: стандартні фільтри не завжди відображають справжній намір користувача, що потребує більш глибоких семантичних і природномовних пошукових механізмів.

Рекомендаційні системи можуть демонструвати ефект «холодного старту» для нових клієнтів і надмірну узагальненість для нестандартних стилів або фігур, знижуючи суб'єктивну релевантність. Нестандартизованість розмірних сіток між брендами зберігає ризики невідповідності, а автоматичні *size-matching*-алгоритми потребують кращих даних і калібрування. З операційного боку висока конкуренція підсилює залежність від платного залучення, тисне на маржу через часті промокампанії та програми безплатних повернень; водночас екологічні зовнішні ефекти інтенсивних доставок і зворотних відправлень стають додатковим

репутаційним викликом. Для брендів, що продають через маркетплейси, існує ризик розмивання історії бренду та часткової канібалізації власних каналів [2; 3].

Порівняльні спостереження за класами платформ. Багатобрендові мас-маркет агрегатори мають максимальну ширину асортименту, гнучкі промостратегії, розвинені мобільні інтерфейси та, як правило, найдемократичніші політики повернення; їхні обмеження — висока частка повернень, залежність від платного трафіку й тиск на валову маржу. Люксові платформи та бутикові ритейлери вирізняються кураторською добіркою, глибшим контентом (історії колекцій, редакційні лукбуки), сервісом стилістів і політиками автентифікації; компромісом є менша ширина складу, вища ціна володіння клієнтом і потенційно довші цикли доставки у крос-бордерних сценаріях. Національні та регіональні рішення переважають у локалізації платіжних методів, інтеграції з внутрішніми логістичними операторами, мовній та культурній адаптації; водночас якість даних каталогу й стандартизація розмірних атрибутів можуть бути менш однорідними через велику кількість продавців [1; 2].

Аналіз напрямів технологічного вдосконалення. Перспективними є інструменти детального моделювання посадки з використанням параметрів тіла та фізичного моделювання тканин; покращені візуальні пошуки, здатні розпізнавати фасони, лекала й текстури; багатоджерельні системи підбору розміру, що агрегують історію повернень, дані виробників і сигнали з поведінки користувачів; а також персоналізація, яка враховує контекст події, сезонність і бюджетні обмеження. На стороні операцій важливими стають прозора аналітика зворотної логістики, тарифи, чутливі до вуглецевого сліду, і сценарії «try-before-you-buy» з контрольованим ризиком [2].

Український контекст. Локальні платформи та маркетплейси мають перевагу в інтеграції з національними платіжними інструментами та службами доставки, у підтримці мовних і регуляторних вимог, а також у швидкості виконання замовлень у межах країни. Водночас для них особливо критичними є мобільна швидкодія інтерфейсів, точність розмірних рекомендацій і стандартизація товарних даних,

оскільки саме ці фактори найбільше впливають на конверсію й рівень повернень у модній категорії [7].

Одже популярні fashion e-commerce платформи досягли високих стандартів у зручності використання, ширині асортименту і персоналізації. Стратегічні обмеження зосереджуються навколо невизначеності посадки, навігації у надвеликих каталогах, вартості зворотної логістики та залежності від платного залучення.

Подальша конкурентна перевага формуватиметься завдяки впровадженню реалістичніших AR-/VR-примірок, семантичного пошуку й аналізу стилю, багатоджерельного підбору розміру, а також операційних практик, що зменшують частку повернень і екологічний слід без зниження якості клієнтського досвіду [2].

### **1.3 Наслідки застосування сучасних нових інформаційних технологій у Fashion E-commerce**

Сучасні IT-технології суттєво трансформують Fashion E-commerce, впливаючи практично на всі аспекти діяльності компаній – від проектування колекцій і планування виробництва до організації логістики, маркетингових кампаній і післяпродажного сервісу. До ключових технологічних напрямів, що визначають цифрову трансформацію модної електронної комерції, належать штучний інтелект і машинне навчання (Artificial Intelligence, AI; Machine Learning, ML), технології Big Data та аналітики даних, Інтернет речей (Internet of Things, IoT) і RFID, блокчейн і смарт-контракти, а також роботизація й автоматизація бізнес-процесів [2].

Використання технологій AI/ML у Fashion E-commerce охоплює увесь ланцюг створення цінності – від прогнозування попиту та формування асортименту до персоналізації взаємодії з клієнтом і оптимізації логістики. Узагальнено основні напрями впливу AI/ML на екосистему модної електронної комерції на рис. 1.1



AI/ML трансформують Fashion E-commerce, проте важливо повднувати автоматизацію з етикою, креативністю та контролем для досягнення сталого розвитку.

Рисунок 1.1 – Вплив AI/ML на Fashion E-commerce [11]

Застосування алгоритмів машинного навчання дає змогу реалізувати персоналізовані рекомендації товарів, візуальний пошук за зображенням, віртуальних стилістів і чат-ботів, а також сервіси віртуальної примірки на основі технологій доповненої та віртуальної реальності. AI-моделі використовуються для прогнозування трендів, динамічного ціноутворення, управління запасами, виявлення підробок, генерації контенту (фото- та текстових описів товарів) і підвищення загального рівня клієнтського досвіду. Разом із тим широке впровадження AI/ML супроводжується ризиками: можливою втратою «людського» виміру персоналізації, проблемами конфіденційності даних, посиленням алгоритмічних упереджень, зростанням енергоспоживання та соціальними наслідками для ринку праці, що

потребує поєднання технологічних інновацій з етичними стандартами та людським контролем [2; 3].

Технології Big Data у Fashion E-commerce дозволяють системно збирати та аналізувати великі масиви даних про поведінку користувачів, продажі, логістику та ринкові тренди. Подано узагальнену схему впливу Big Data на модну електронну комерцію з погляду їхніх переваг та недоліків на рис. 1.2



Рисунок 1.2 – Вплив Big Data у Fashion E-commerce: переваги та недоліки [12]

Завдяки Big Data підвищується точність персоналізованих рекомендацій, оптимізується асортимент і дизайн колекцій, вдосконалюється управління запасами та маршрутизація поставок, реалізується динамічне ціноутворення й таргетований маркетинг. Водночас концентрація значних обсягів чутливих даних створює загрози порушення конфіденційності, маніпуляцій поведінкою споживачів, посилення «фільтрувальних бульбашок», а також сприяє надмірному споживанню й додатковому екологічному навантаженню. Це потребує дотримання принципів

етичного використання даних та відповідності нормативним вимогам щодо їх захисту [1; 2].

IoT та RFID-технології у Fashion E-commerce забезпечують підвищення прозорості та керованості фізичних потоків товарів. Відображено основні наслідки застосування IoT/RFID для модної електронної комерції на рис. 1.3



Рисунок 1.3 – Наслідки застосування IoT/RFID у Fashion E-commerce: переваги та недоліки [13]

Використання RFID-міток і IoT-сенсорів дозволяє здійснювати розумне управління запасами, відстежувати рух продукції в режимі реального часу, контролювати умови транспортування, автоматизувати процеси приймання, відвантаження й повернення товарів, а також реалізовувати персоналізований шопінг у фізичних магазинах за допомогою «розумних» примірочних і вітрин. Разом із цим виникають питання кібербезпеки, конфіденційності даних про клієнтів,

високих капітальних витрат на впровадження, утворення електронних відходів та залежності бізнесу від технічної надійності інфраструктури [1; 7].

Блокчейн та смарт-контракти відкривають нові можливості для підвищення прозорості ланцюгів постачання, боротьби з контрафактною продукцією та створення нових форм взаємодії з клієнтами у Fashion E-commerce. Схематично подано основні переваги й недоліки застосування Blockchain та Smart Contracts у модній електронній комерції уа рис. 1.4



Рисунок 1.4 – Наслідки застосування Blockchain та Smart Contracts у Fashion E-commerce: переваги та недоліки [14]

Блокчейн-рішення дають змогу фіксувати походження сировини й готових виробів, формувати цифрові паспорти товарів, реалізовувати токеновані сертифікати автентичності (NFT), автоматизувати гарантійне обслуговування, створювати децентралізовані маркетплейси та програми лояльності на основі токенів, а також запроваджувати криптовалютні платежі й механізми автоматичної виплати роялті дизайнерам. Водночас блокчейн характеризується високими витратами на впровадження, обмеженою масштабованістю, значним

енергоспоживанням окремих мереж, правовими та регуляторними ризиками, а також складністю сприйняття для частини користувачів [2].

Роботизація й автоматизація процесів у Fashion E-commerce охоплюють роботизовані склади, автоматизовані системи сортування й пакування, використання мобільних роботів, чат-ботів, систем RPA для оброблення транзакцій та управління даними. Наведено узагальнення впливу роботизації та автоматизації на модну електронну комерцію на рис. 1.5.



Рисунок 1.5 – Наслідки застосування роботизації та автоматизації у Fashion E-commerce: переваги та недоліки [15]

Такі рішення сприяють прискоренню виконання замовлень, зменшенню операційних витрат, підвищенню точності обліку, розширенню можливостей персоналізації, зниженню частки повернень завдяки точнішому підбору розмірів і автоматизованому контролю якості. Разом з тим виникають ризики скорочення робочих місць, посилення залежності бізнесу від складних технічних систем, можливого зниження якості «живого» сервісу, а також додаткового екологічного впливу, пов'язаного з виробництвом та утилізацією обладнання [1; 6].

На основі проведеного аналізу можна зробити висновок, що впровадження сучасних ІТ-технологій у Fashion E-commerce істотно змінює бізнес-процеси та моделі взаємодії з клієнтами. Використання штучного інтелекту, Big Data, IoT/RFID, блокчейну та роботизації забезпечує персоналізацію покупок, появу віртуальних примірочних, автоматизацію ключових операцій, інтеграцію електронної комерції з соціальними мережами та формування нових форматів цифрового клієнтського досвіду. Водночас ці технології породжують низку викликів – від питань конфіденційності й етики використання даних до екологічної сталості та соціальних наслідків для ринку праці. Бренди, які поєднують технологічні інновації з відповідальним ставленням до споживачів і довкілля, отримують довгострокові конкурентні переваги на глобальному ринку модної електронної комерції [2]

## 2 ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ ІНТЕЛЕКТУАЛЬНИХ НОВИХ ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ Е-COMMERCE ПЛАТФОРМИ ДЛЯ ПРОДАЖУ ОДЯГУ

### 2.1 Дослідження сучасних технологій для розробки e-commerce платформ

**2.1.1 Технології для розробки Frontend.** Сучасні інтерфейси e-commerce платформ для продажу одягу у 2025 році спираються на еволюцію стандартів веб-платформи (HTML, CSS, ECMAScript/JavaScript), зрілі компонентні фреймворки (React/Next.js, Angular, Vue/Nuxt) та на формалізовані метрики якості взаємодії. Оновлене видання стандарту ECMAScript 2025 кодифікує низку удосконалень для роботи з модулями, колекціями та асинхронними операціями, що зменшує залежність від допоміжних бібліотек і підвищує надійність великих фронтенд-кодобаз модних маркетплейсів із насиченим мерчандайзингом, фільтрами та персоналізованими добірками [16]. Це створює більш передбачуваний фундамент для довготривалих проєктів і спрощує підтримку багатомовних та мультирегіональних вітрин.

У площині CSS спостерігається структурна зміна підходів до адаптивності. Замість жорсткої прив'язки лише до розміру вікна перегляду дедалі ширше застосовуються контейнерні запити, що дозволяють адаптувати компонент залежно від властивостей батьківського контейнера. Псевдоклас `:has()` забезпечує «реактивну» поведінку батьківських елементів залежно від стану вкладених вузлів без додаткового «клею» на JavaScript. Це дає змогу підтримувати єдину систему карток товарів і панелей фільтрів у сітках із різною щільністю та композицією, зменшуючи конфлікти специфічності та спрощуючи темізацію. Офіційний документ CSS Snapshot 2025 узагальнює актуальний стан модулів і слугує єдиною точкою відліку для браузерів та інструментів збірки, що є критично важливим для

масштабних команд, які розвивають дизайн-системи fashion-каталогів [17]. У результаті значна частина поведінки, що раніше вимагала скриптів, переноситься в декларативні стилі, а фронтенд стає простішим у валідації та супроводі.

Ключовою метрикою якості взаємодії у 2025 році є Interaction to Next Paint (INP), яка характеризує загальну чутливість інтерфейсу до дій користувача, включаючи затримки в обробленні подій і відмальовуванні наступного кадру. Методичні рекомендації пропонують орієнтуватися на INP не вище 200 мс (на 75-му перцентилі сесій) як «добрий» рівень, тоді як значення понад 500 мс розглядаються як «погані» [18]. Для модних каталогів із великою кількістю зображень високої якості це безпосередньо означає пріоритет серверного рендерингу та потокової видачі HTML, мінімізацію клієнтських пакетів JavaScript, винесення «важких» обчислень у веб-воркери або бекенд-сервіси, а також застосування адаптивних стратегій завантаження медіа залежно від умов мережі [18]. Наочні пороги INP відображено на діаграмі (рис. 2.1).



Рисунок 2.1 - Пороги Interaction to Next Paint (INP): «добре»  $\leq 200$  мс;  
«погано»  $> 500$  мс [18]

Архітектурно суттєві зміни пов'язані з чіткішим розмежуванням виконання між сервером і клієнтом у компонентних фреймворках. У React-екосистемі Next.js за замовчуванням застосовує Server Components для сторінок і макетів, що дозволяє виконувати завантаження даних та рендеринг на сервері, кешувати результат і потоково передавати HTML на клієнт; необхідна інтерактивність додається окремими клієнтськими компонентами [19]. Такий підхід особливо доцільний для fashion-каталогів: SEO-критичні сторінки з великою кількістю товарів і фільтрів відмальовуються на сервері, тоді як локальна взаємодія (модальні вікна, порівняння товарів, мікровзаємодії в кошику) реалізується в мінімальній кількості клієнтського

коду. Узагальнену схему розподілу відповідальності між браузером і сервером наведено на рис. 2.2.

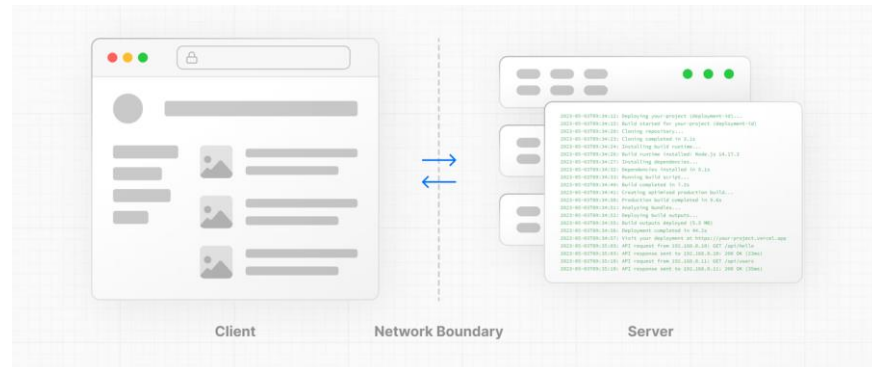


Рисунок 2.2 - Server and Client Environments (Next.js Learn, 2025) [19]

У світі Angular ключовими трендами 2024–2025 років є перехід до *zoneless*-підходу та закріплення `signals` як базового реактивного примітиву. Офіційна документація фіксує, що `signals` дають змогу явно описувати граф залежностей стану, зменшуючи кількість неконтрольованих перерендерів і полегшуючи профілювання складних інтерфейсів [20]. Для e-commerce-рішень це означає стабільнішу роботу форм реєстрації й оформлення замовлення, прогнозоване оновлення кошика та кращий контроль продуктивності у великій кодобазі. Усунення залежності від `Zone.js` також спрощує інтеграцію з сучасними веб-API й сторонніми бібліотеками [20].

Екосистема `Vue` консолідується навколо `Composition API`, що поєднується з `Nuxt 3` та `Vite` для забезпечення легковагих бандлів, серверного рендерингу, статичної генерації та потокової видачі. Офіційні посібники `Vue` і `Nuxt` підкреслюють, що така конфігурація особливо придатна для проєктів, де першорядними є швидкодія, керованість дизайн-системою та сумісність із `TypeScript` [21; 22]. Для fashion-платформ це дозволяє повторно використовувати однакові композиційні модулі (логіка вибору розміру, робота зі списком бажань, рекомендації «total look») у різних компонентах — від картки товару до сторінки колекції — без дублювання коду, забезпечуючи водночас стабільні показники завантаження на мобільних пристроях.

Для національного контексту важливо враховувати, що мобільні сценарії залишаються домінантними. Аналітичні огляди українського ринку e-commerce за перше півріччя 2025 року свідчать, що саме fashion-категорія демонструвала від’ємну динаміку виручки (близько  $-2\%$  у гривні порівняно з Н1 2024) на тлі скорочення кількості покупок приблизно на  $12\%$  і зниження конверсії на  $10\%$ , водночас середній чек зріс [23]. Це інтерпретується як зсув до залучення меншого, але якіснішого трафіку та концентрація продажів у брендів із вищою сприйнятою цінністю. З інженерної точки зору така конфігурація підтверджує актуальність пріоритетів: продуктивні мобільні інтерфейси, адаптивна подача зображень, мінімізація клієнтського JavaScript і досягнення цільових порогів INP, що безпосередньо корелюють із конверсією у сегменті з високою часткою мобільних сесій [18; 23]. Сумарно оптимальна фронтенд-архітектура для fashion e-commerce у 2025 році поєднує стандартні можливості HTML/CSS/JavaScript (із фокусом на контейнерних запитах та псевдокласі `:has()`), серверний рендеринг і стрімінг для каталожних та SEO-критичних сторінок, вибіркове застосування клієнтських компонентів для локальної інтерактивності, сучасні реактивні примітиви фреймворків (signals, Composition API) і дисципліну зменшення обсягу клієнтського коду [16; 17]. Узагальнені приклади того, як ці можливості веб-платформи впливають на інтерфейси продажу одягу, наведено в табл. 2.1.

Таблиця 2.1 - Ключові можливості веб-платформи (2025) та вплив на fashion e-commerce [16-17]

Компонент платформи	Ключова можливість	Вплив на інтерфейси продажу одягу
ECMAScript 2025	Оновлення специфікації; уніфікація модульності; надійніша робота з даними	Менше допоміжних бібліотек; стабільніша збірка великих вітрин
CSS Snapshot 2025	Консолідація модулів; єдина опора для реалізацій	Узгоджені дизайн-системи у великих командах
Контейнерні запити	Адаптація за властивостями контейнера	Єдині картки/фільтри у різних сітках без дублювання стилів
<code>:has()</code>	Селекція «вгору» від дочірніх структур	Реактивні стани без JS-клею
INP	Цільове значення $\leq 200$ мс (75-й перцентиль)	Пріоритет SSR/стрімінгу; мінімізація клієнтського JS

Порівняльні характеристики провідних фронтенд-фреймворків, які найчастіше застосовують для реалізації модних каталогів, систематизовано в табл. 2.2.

Таблиця 2.2 - Порівняння фреймворків для модних каталогів [19-21]

Критерій	React + Next.js	Angular 20.x	Vue 3 + Nuxt + Vite
Модель виконання за замовчуванням	Server Components, стримінг HTML; клієнтські компоненти для інтерактивності	Зонелес виконання; signals як базовий примітив реактивності	SSR із подальшою гідрацією; легковага збірка на Vite
Типові показники взаємодії (INP) на мобільних	Високі: RSC зменшує клієнтський JS і покращує INP за рахунок стримінгу	Високі: signals зменшують «зайві» перерендери у формах/кошиках	Високі: легкий бандл + ефективна гідрація; залежить від дисципліни компонентів
SSR/SSG/ISR/стримінг	SSR/SSG/ISR «з коробки», стримінг частинами інтерфейсу	SSR через Universal; стримінг опціональний, складніший у впровадженні	SSR/SSG «з коробки»; стримінг/partial hydration можливі через Nuxt-плагіни
Маршрутизація та код-сплітинг	Файлова маршрутизація, автоматичний сплітинг; зручні «layout-и»	Потужний роутер; детальний контроль зон і змін	Файлова маршрутизація в Nuxt; швидке налаштування сплітингу
Стан і реактивність	Локальний стан + сторонні рішення (Zustand/Redux); серверні мутації через маршрути/дії	Вбудовані signals, зрозумілий граф реактивності; сильні форми	Composition API (reactive/ref), Pinia як стандарт де-факто
Форми та валідація	Через бібліотеки (React Hook Form), багато ручних налаштувань	Сильна сторона: реактивні форми, валідатори, типізація	Через сторонні пакети (VeeValidate/Pinia Forms)
i18n/локалізація	next-intl/next-i18next; добре поєднується з file-based routing	Офіційні рішення + сувора типізація	nuxt-i18n; швидка інтеграція з SSG/SSR
Оптимізація зображень	Вбудований Image Optimization (адаптивні розміри, ладери)	Через сторонні білдери/плагіни; більше ручної роботи	Nuxt Image (модуль) з адаптивними пресетами
SEO/метадані	Вбудовані metadata/head; сильний SSR/ISR для індексації	SEO через Universal; потребує налаштування	useHead/app.config, SSR зручний для індексації

Профілювання/спостережність	Добрі девтулзи + екосистема перформанс-інструментів; трасування RSC	Офіційні DevTools для signals; зрозумілий профайлінг змін	Vue DevTools + Nuxt Analyze; простий перегляд реактивності
Крива навчання	Середня: треба розмежування сервер/клієнт і розуміння RSC	Вища: суворі структура, DI, реактивні форми, signals	Низька/середня: простий старт, далі - дисципліна в Composition API
Масштаб команд (50+ розробників)	Комфортно: модульні межі між RSC/клієнтом; потрібні правила	Дуже комфортно: «рамки» фреймворку мінімізують відхилення	Комфортно: потрібні гайдлайни по композиційному коду
Типові ризики	Межі сервер/клієнт, змішаний рендер; ризик «води» у бандлі	Міграції/строгіст; навчальна крива на великих формах	Роз'їзд стилів/патернів без гайдлайнів; непослідовна типізація
Коли обирати	Великі каталоги, сильний SEO, агресивні цілі INP/конверсії; швидкі дропи колекцій	Складні форми/кошки, корпоративна дисципліна, висока підтримуваність	Швидкі MVP → масштаб; багато UI-експериментів, легка SSR/SSG

Узагальнення зіставлення технологій (див. табл. 2.2) показує, що архітектури, які мінімізують обсяг клієнтського JavaScript, підтримують серверний рендеринг і потокову видачу, а також спираються на сучасні реактивні примітиви, є оптимальними для модних каталогів із високою часткою мобільних сесій [18]. Вибір стека має корелювати з домінуючими бізнес-КРІ: швидкістю відгуку на мобільних (INP), частотою оновлення колекцій, складністю форм і кошика, а також масштабом команди та вимогами до SEO. React + Next.js надає найбільше переваг для великих, динамічних вітрин із жорсткими вимогами до SEO/INP завдяки Server Components, стримінгу та вбудованій оптимізації зображень; Angular 20.x із signals і «zoneless»-підходом забезпечує передбачувану продуктивність у складних інтерфейсах з насиченими формами, де важлива інженерна дисципліна та підтримуваність; Vue 3 + Nuxt + Vite пропонує найнижчий поріг входу, швидку SSR/SSG і легкі бандли, що є зручним для швидких MVP із подальшим масштабуванням за наявності командних гайдлайнів [19; 21]. Практична інтерпретація така: якщо головні КРІ – мобільний INP і SEO для великого та часто оновлюваного каталогу, доцільно обрати React +

Next.js; якщо переважають складні бізнес-правила у формах/кошиках і потрібна суворе керуваність у великій команді – виправданим є Angular 20.x із signals; якщо критичні швидкість розробки та гнучкість (MVP → масштаб) з SSR/SSG «із коробки» та легкими бандлами – раціональним вибором буде зв'язка Vue 3 + Nuxt + Vite.

**2.1.2 Технології для розробки Backend.** Серверна архітектура сучасних e-commerce-рішень у fashion-категорії у 2025 році визначається поєднанням високопродуктивних середовищ виконання, зрілих веб-фреймворків і керованих сервісів розгортання. До ключових вимог належать здатність витримувати пікові навантаження під час «дропів» колекцій, низька латентність API для мобільних клієнтів, надійне оброблення операцій кошика й оплат, а також прозоре моделювання складних бізнес-процесів повернень і обмінів. У цьому контексті доцільно розглядати чотири поширені стеки: Node.js (екосистема JavaScript/TypeScript), Django (Python), Laravel (PHP) та Ruby on Rails (Ruby).

На рівні платформи Node.js у 2025 році підтримує довгострокову гілку LTS 22.x з регулярними безпековими оновленнями протягом усього життєвого циклу релізу. Це критично для тривалих інсталяцій із жорсткими вимогами до стабільності runtime та сумісності модулів у періоди різких сплесків трафіку під час маркетингових кампаній і запуску нових колекцій [24].

Django 5.2 поєднує класичну синхронну модель виконання із сучасними асинхронними шляхами (ASGI, Django Channels), що надає змогу гнучко розмежовувати CRUD-навантаження й подієві сценарії (чат-повідомлення, нотифікації, стрімінг статусу замовлень). У реліз-нотах 5.2 та подальших патч-версій окремо підкреслено посилення безпеки та політику зворотної сумісності, на яку спираються команди, що підтримують великі історичні кодові бази [25].

Екосистема Laravel (12.x) позиціонується не лише як фреймворк, а як частина цілісної платформи доставки: поєднання Laravel, керованих сервісів Laravel Forge та Laravel Cloud дає змогу зменшити витрати на налаштування інфраструктури, автоматизувати масштабування та спростити впровадження CI/CD-процесів для e-commerce-проектів малого й середнього бізнесу [26].

У світі Ruby on Rails триває розвиток у напрямі «продуктивність + зручність розробника»: лінійка 8.1 упроваджує, зокрема, механізми job continuations та структурування подій, орієнтовані на масштабні застосунки з інтенсивними фоновими процесами (обробка платежів, синхронізація залишків, нотифікації клієнтів) [27].

З погляду бізнес-інтеграцій для fashion e-commerce важливим трендом є перехід постачальників до «уніфікованих комерційних API», що охоплюють компоненти ERP, PIM, OMS та storefront-шару. Це стимулює запровадження подієвих взаємодій (webhooks), проміжних GraphQL-шарів і гібридних REST/GraphQL-архітектур. Показовим прикладом є перехід екосистеми Shopify до переважного використання GraphQL Admin API, який дає змогу виконувати складні вибірки з меншими затримками та природно інтегрується з сервісами персоналізації й аналітики в реальному часі [28]. Для українських гравців додатково актуальними є інтеграції з локальними платіжними сервісами, службами доставлення та поштоматами, що потребує стабільних SDK і можливості побудови надійних асинхронних фонових процесів (повторні списання, повернення, уточнення адреси тощо).

Узагальнюючи, до «ідеальних» вимог до бекенд-стека модного інтернет-магазину можна віднести чітку модель конкурентності (неблокувальне I/O або добре масштабовані воркери), зрілий фреймворк із вбудованими засобами безпеки, ORM та транзакційністю, модульну інтеграцію з headless-компонентами (PIM, OMS, платіжні шлюзи, служби доставлення), наявність інструментів профілювання й трасування як на рівні runtime, так і ORM, підтримку керованих сервісів для CI/CD, логування й автоматичного масштабування [24; 25].

У табл. 2.3 систематизовано, як основні бекенд-стеки задовольняють зазначені критерії.

Таблиця 2.3 - Порівняння бекенд-стеків для fashion e-commerce [24-27]

Критерій	Node.js (Express/Nest)	Django (5.2)	Laravel (12.x)	Ruby on Rails (8.1)
Модель конкурентності	Події, неблокувальне I/O; стабільні LTS-релізи 22.x у 2025 р.	Sync ядро + зрілі аsync-шляхи (ASGI, Channels); чіткі реліз-ноти 5.2	Sync запити + високопродуктивні воркери (Octane/Swoole/RoadRunner) і Reverb для WS	Threaded/процесна модель; 8.1 акцентує job continuations і структуровані події
ORM і транзакції	TypeORM/Prisma; флексибельність схем/міграцій	Django ORM (надійна транзакційність, міграції «з коробки»)	Eloquent (зручність моделей, міграції, кастомні касти)	Active Record (зрілі callback'и/міграції; оптимізація запитів)
API-шари	REST/GraphQL (Apollo, Mercurius), Server-Sent Events	Django REST Framework; Graphene-Django	API Resources; GraphQL (Lighthouse); WebSockets (Reverb)	Rails API mode; GraphQL-Ruby; Hotwire/Turbo для подій
Безпека	Часті LTS-патчі runtime; audit npm-ланцюга	Прозорі security-релізи (напр., 5.2.7)	Регулярні security-релізи фреймворку/екосистем	Регулярні патчі фреймворку; офіційні нотатки 8.1
Спостережність і перформанс	Профілювання event loop, APM; ефективний I/O	Профілювання ORM/кеш; зрілі middleware	Telescope, Pulse/Forge інтеграції; кеш/queues із Horizon	Bullet/Skylight; Active Support Notifications
DevOps/розгортання	Контейнери + Node LTS; serverless/edge сумісність	Класичні VM/контейнери; ASGI-стек	Керовані сервіси (Forge/Cloud) зі швидким автоскейлом [32; 33]	Класичні VM/контейнери; масштабовані worker-пули
Типові сильні сторони	Низька латентність API, WS-пуші, єдина мова з фронтом	Дисциплінований моноліт; швидка CRUD-розробка, адмінка	«Батарейки в комплекті», швидкий старт продакшен-інфр	Висока продуктивність команди, зріле CRUD, convention over configuration
Типові компроміси	Контроль якості прт-залежностей, ручне складання фреймворку	Python-екосистема важча для дуже високих RPS	PHP-перформанс залежить від runtime (FPM vs Octane)	Пам'ять/конкурентність потребують дисципліни; tuning для високих RPS

У табл. 2.4 наведено типові сценарії вибору бекенд-технології для модного інтернет-магазину та відповідні рекомендації.

Таблиця 2.4 - Сценарії вибору бекенд-технології для модного інтернет-магазину [24-28]

Сценарій	Рекомендований стек	Аргументація
Високочастотні мобільні запити, live-інвентар, push-події	Node.js (Nest/Express)	Низька латентність неблокувального I/O; зручні WS/SSE; єдина мова з фронтендом
Корпоративний моноліт із жорсткими правилами повернень, складні форми адміністрування	Django 5.2	Суворий ORM/транзакції, адмінка «з коробки», передбачувані релізи й security-патчі
Швидкий вихід у прод з «опініонованою» інфраю (керовані БД/деплой, автоскейл)	Laravel 12.x	Екосистема Forge/Cloud/Reverb; готові пакети оплати/UGC; швидке налаштування DevOps
Велика команда, висока швидкість CRUD-розробки, подієві фонові процеси	Rails 8.1	Convention over configuration, job continuations, багатий DX
Інтенсивні маркетингові інтеграції в headless-ландшафті, потреба в GraphQL	Будь-який із чотирьох + GraphQL	Shopify-подібні інтеграції (GraphQL Admin API, webhooks) скорочують латентність складних вибірок

Для fashion-каталогу, що активно «живе» у соціальних каналах і має високу частку мобільного трафіку, стек на основі Node.js зазвичай забезпечує найкращі часові характеристики API та подій завдяки неблокувальній моделі I/O [25]. Django є доцільним вибором для дисциплінованого доменного моноліту з потужним бек-офісом, де критичною є надійність транзакцій і цілісність даних [26]. Laravel мінімізує «тертя» DevOps завдяки керованим сервісам і дає змогу швидко довести продукт до промислової експлуатації [28]. Rails, з огляду на нові можливості 8.1 для роботи з подієвими потоками, зберігає високу продуктивність команд у великих інсталяціях, де важлива швидкість розробки й узгодженість архітектурних підходів [30]. Остаточний вибір бекенд-стека має базуватися на цільових KPI (латентність,

конверсія, вартість володіння), доступності компетенцій команди та сумісності з наявною headless-екосистемою платформи [24; 28].

**2.1.3 Технології для розробки бази даних.** Архітектура зберігання для fashion e-commerce у 2025 році спирається на поєднання ACID-транзакційних СУБД для критичних потоків (кошик, оплати, повернення), гнучких документно орієнтованих моделей для швидких змін каталогу й персоналізації, а також на механізми горизонтального масштабування та високої доступності. Практичні вимоги до бекенду модних вітрин – пікові «дропа» колекцій, масштабована аналітика атрибутів товарів і медіа, мінімальна затримка API для мобільних клієнтів – зумовлюють вибір архітектур, де реплікація, шардинг і онлайн-оновлення виконуються без простою. Нижче узагальнено сильні сторони MySQL, PostgreSQL та MongoDB станом на 2025 рік із фокусом на релевантні для e-commerce можливості.

MySQL 8.4 (LTS) у 2025 році підтримує послідовний цикл LTS-випусків, що є важливим для стабільності великих інсталяцій і планових безпекових оновлень у продакшені [29]. Для побудови HA-кластерів рекомендовано InnoDB Cluster (Group Replication + MySQL Router + MySQL Shell), який забезпечує автоматичне членство, відмовостійкість і керований фейловер; для міжрегіональної стійкості застосовується InnoDB ClusterSet з асинхронними каналами реплікації між кластерами [30]. Поєднання цих технологій дозволяє реалізовувати типову для e-commerce топологію «один primary для записів + кілька secondary для читань», а також сценарії disaster-tolerance із прозорим перенаправленням трафіку через Router.

PostgreSQL у 2025 році пропонує зрілий стек логічної реплікації як для оновлень без простою (zero-downtime upgrades), так і для складних сценаріїв міграції та декомпозиції – включно з фільтрацією рядків/стовпців, потоковою передачею транзакцій і BDR-візерунками [31; 32]. Актуальна документація «current» (PostgreSQL 18) систематизує архітектурні основи (клієнт/сервер, процеси backends, шлях обробки запитів) та деталі логічної реплікації (publication/subscription, replication slots), що дозволяє формувати гібридні топології – наприклад, відокремлювати «гарячі» таблиці замовлень у виділені вузли без зміни клієнтського

API [31]. Практичні кейси демонструють безперервні оновлення версій через логічну реплікацію та найкращі практики керування replication slots, щоб уникати надмірного зростання WAL [32].

MongoDB у 2025 році закріплює роль «операційного документного сховища з горизонтальним масштабуванням» для каталогів і персоналізації: штатний шардинг (розподіл даних на рівні колекції по шардах-репліка-сетах) та керовані сервіси Atlas з інтеграціями Data Federation та Online Archive забезпечують контроль вартості зберігання й запитів до «холодних» даних без зміни прикладної логіки [33; 34]. Оглядові матеріали MongoDB та програма модернізації застосунків (AMP) наводять приклади клієнтів, що консолідуєть «каталог + UGC + події» в єдиній документній моделі з горизонтальною масштабованістю [35]. Для e-commerce це означає можливість швидкого еволюціонування атрибутів товарів, варіантів і метаданих медіа без складних міграцій схем.

У підсумку типовий поділ обов'язків виглядає так: транзакційні потоки (кошик, оплати, повернення) – у реляційному ядрі (MySQL або PostgreSQL) з реплікацією й HA-кластерами; гнучкий каталог, персоналізація, події взаємодії – у документному шарі (MongoDB) з шардингом за ключем колекції; міжсистемні «мости» – через логічну реплікацію (PostgreSQL), канали подій/черги, а також ETL-процеси чи Data Federation для аналітики. Така композиція поєднує стабільність транзакцій із еволюційністю моделі даних, що особливо важливо для модних вітрин із високою частотою змін асортименту. Узагальнені характеристики СУБД наведено в табл. 2.5, а прикладні відповідності типових задач – у табл. 2.6.

Таблиця 2.5 - Ключові можливості СУБД та їх відповідність потребам fashion e-commerce [29-35]

Критерій	MySQL 8.4 (LTS)	PostgreSQL 18 (current)	MongoDB Atlas (2025)
Транзакційність (ACID)	InnoDB, XA; перевірене ACID-ядро	Повний ACID; зрілий MVCC	Транзакції на документ/мульти-документ
Реплікація/HA	Group Replication; InnoDB Cluster; ClusterSet для міжрегіональної стійкості	Логічна реплікація (publication/subscription), BDR-візерунки, zero-downtime upgrades	Репліка-сети; автоматичний фейловер у кластері
Шардинг	Вбудованого шардингу немає; шардинг на рівні додатка або NDB/проксі	Шардинг через розподіл на рівні додатка/FDW, логічна сегментація	Вбудований шардинг на рівні колекції
JSON/гнучка схема	JSON тип + індекси; загалом реляційна схема	JSONB з індексами, вирази, оператори	Документна модель; гнучка схема «без міграцій»
Онлайн-оновлення	LTS-патчі; зміна топологій у Cluster/ClusterSet	Zero-downtime upgrades через логічну реплікацію	Керовані оновлення Atlas, без простою
Аналітика/федерація	Зовнішні конектори/ETL; read-replicas	FDW/матеріалізовані представлення; логічні канали	Data Federation, Online Archive на GCP
Типові ролі у fashion	Замовлення/платежі, інвентар (OLTP)	Замовлення/взаємодії з логічним поділом, аудити	Каталог, персоналізація, UGC, події

Таблиця 2.6 - Відповідність шаблонів e-commerce задачам зберігання [29-35]

Завдання	Рекомендована технологія	Обґрунтування
Кошик/замовлення/оплати	MySQL InnoDB або PostgreSQL	ACID, справедливі блокування/ізоляції; проста реплікація читань; прозорий фейловер
Каталог (варіанти/атрибути/медіа)	MongoDB (шардований кластер)	Гнучка схема; горизонтальне масштабування; дешево архівування «холодних» даних
Повернення/обміни (RMA)	PostgreSQL + логічна реплікація	Безперервні зміни моделі/версій; виділення потоків у окремі вузли
Персоналізація/UGC	MongoDB Atlas	Документи з вкладеними об'єктами; прості моделі для рекомендацій/UGC
Аналітика подій/ETL	PostgreSQL (FDW) або Atlas Data Federation	Об'єднання джерел без міграцій; оптимізація вартості

**2.1.4 Технології для розробки хмарних сервісів.** У 2025 році архітектура хмарної інфраструктури для fashion e-commerce спирається на поєднання глобальних CDN і edge-виконання, керованих сервісів даних, засобів мережевої безпеки (WAF, захист від ботів), спостережності та інструментів контролю вартості. Для модних інтернет-магазинів із піковими навантаженнями під час «дропів» колекцій це означає потребу мінімізувати латентність першого байта (TTFB), забезпечити потокову видачу HTML та медіаконтенту, гнучко керувати політиками кешування на «краю» мережі, а також гарантувати передбачувану еластичність під час маркетингових кампаній.

На рівні мережевої доставки Amazon Web Services у складі AWS пропонує сервіс Amazon CloudFront як глобальний CDN із підтримкою edge-функцій, інтеграцією з AWS WAF та глибокою телеметрією запитів. Офіційний посібник для розробників описує життєвий цикл запиту на «краю», порядок застосування правил безпеки та рішень щодо кешування, що є критичним для точного налаштування

поведінки під час пікових розпродажів колекцій [36; 37]. У поєднанні з AWS WAF та іншими сервісами безпеки CloudFront дозволяє централізовано застосовувати правила фільтрації трафіку, захисту від DDoS/бот-трафіку й шифрування з'єднань, зберігаючи низьку затримку для мобільних клієнтів [36; 37; 43].

У Google Cloud роль глобального CDN виконує Cloud CDN, який інтегрується з глобальними HTTP(S)-балансувальниками та дозволяє кешувати як статичний, так і динамічно згенерований вміст максимально близько до користувача [38]. Для сценаріїв інтенсивного мультимедійного контенту (відео-лупи в картках товару, великі зображення колекцій, покази з подіуму) додатково застосовується Media CDN, оптимізований для потокової доставки відео й великих медіафайлів [39]. У практиці модних e-commerce-платформ це дає змогу винести «важкий» медіа-трафік із бекенд-сервісів, скоротити кількість HTTP-циклів для завантаження зображень і відео та зменшити загальну латентність взаємодії на мобільних пристроях [38; 39].

У Microsoft Azure для глобальної доставки контенту та маршрутизації застосовується Azure Front Door (Standard/Premium), який поєднує можливості глобального балансування навантаження, кешування на «краю» та інтеграції з Azure Web Application Firewall (WAF).

Оновлена документація підкреслює необхідність міграції з застарілого Front Door (Classic) на Standard/Premium, що забезпечує модернізовану модель безпеки та кращу керованість WAF-політиками [40]. Паралельні оновлення у сфері Application Gateway та Azure Firewall посилюють захист прикладного периметра для публічних веб-застосунків, зокрема й модних інтернет-магазинів з глобальним трафіком [40].

З бізнесової точки зору, великі хмарні провайдери розвивають «галузеві» ритейл-рішення. Google Cloud через портфель рішень для роздрібною торгівлі демонструє поєднання Cloud CDN/Media CDN з платформою Vertex AI для персоналізації, рекомендацій і retail-media-сценаріїв (банерні кампанії, промо-слоти на вітрині) [42].

AWS акцентує масштабованість своєї інфраструктури на прикладі Prime Day, коли сервіси обробляють історичні піки транзакцій та подій, зберігаючи стабільність і низьку латентність – це безпосередньо релевантно для модних e-commerce-

проектів із «дропами» капсульних колекцій та сезонними сейлами [41]. Узагальнені рекомендації AWS Well-Architected Framework (6 стовпів: операційна досконалість, безпека, надійність, продуктивність, оптимізація витрат, сталий розвиток) задають еталон для проєктування таких архітектур [43].

Для практичного проєктування хмарної складової fashion-платформи доцільно розміщувати статичний медіаконтент і HTML-відповіді у найближчих до користувача зонах за допомогою CDN та механізмів edge-обробки (CloudFront, Cloud CDN, Azure Front Door), одночасно застосовуючи формалізовані політики кешування (визначення часу життя, використання ETag, коректне формування cache-keys).

Навантаження на первинні джерела (origin) слід зменшувати за рахунок попереднього прогрівання кешу для нових колекцій, використання логіки на рівні edge для реалізації А/Б-експериментів і маршрутизації маркетингових кампаній, а також через чітке відокремлення статичного та медіа-трафіку від транзакційних API-запитів.

Важливим є вбудовування механізмів спостережності безпосередньо на рівні CDN/edge (спеціалізовані панелі моніторингу для відстеження помилок, показників cache-hit ratio, TTFB, географічного розподілу трафіку) з подальшою кореляцією цих метрик із показниками бекенду для проактивного виявлення деградацій продуктивності.

Хмарна архітектура має бути узгоджена з бізнес-KPI (наприклад, INP та TTFB на мобільних пристроях, конверсія на етапі кошика, вартість 1000 запитів або 100 ГБ переданого трафіку), а також відповідати вимогам щодо географії присутності та чинним регуляторним обмеженням. Узагальнений огляд ключових можливостей CDN/edge-рішень і їхнього впливу на модні e-commerce-платформи наведено в табл. 2.7.

Таблиця 2.7 - Ключові можливості CDN/edge та їхній вплив на fashion e-commerce [36-40]

Компонент	Ключова можливість (2025)	Вплив на модні вітрини
Amazon CloudFront	Оновлений інтерфейс та інтеграція з AWS WAF; детальний порядок обробки запиту на «краю»	Прозоре керування правилами кешу/безпеки; відтворювані профілі продуктивності під піки «дропів»
Google Cloud CDN	Service Extensions на «краю» (pre-cache) для глобального External HTTP(S) LB	Правила кампаній (перемикання бекендів, переписування) без навантаження на походження
Google Media CDN	Multipart range-запити (Preview)	Менше HTTP-циклів при завантаженні великих медіа/відео в картках товарів
Azure Front Door	Перехід із Classic на Standard/Premium (обмеження створення classic із 01.04.2025)	Міграція на сучасні SKU з новими політиками WAF; підвищення керованості безпеки

Порівняння трьох основних хмарних екосистем з погляду потреб fashion e-commerce подано в табл. 2.8.

Таблиця 2.8 - Порівняння хмарних сервісів для модних e-commerce [36; 38–40; 42; 43]

Категорія	AWS	Google Cloud	Microsoft Azure
CDN/edge-виконання	CloudFront + AWS WAF; функції на «краю», глибока телеметрія	Cloud CDN + Service Extensions (edge); Media CDN для відео/медіа	Azure Front Door (Standard/Premium) + WAF; глобальна маршрутизація
Потокова видача HTML/медіа	Перевірено на Prime Day-навантаженнях; звіти/метрики	Predefined dashboards для CDN; Next '25 - масові анонси спостережності	Оновлення мережевих сервісів (WAF/Firewall) для стабільності під піки
Персоналізація/AI для ритейлу	Інтегрується з власними сервісами AI; масштаб під глобальні піки	Vertex AI/Gemini; партнерська екосистема для retail media	Azure AI-сервіси; інтеграція з CDN/WAF і SIEM
Керованість/стандарти	Well-Architected (оновлене керівництво, 6 стовпів)	Розширені релізноти CDN/Media; індустриальні тренди	Офіційні оновлення/ретайри для Front Door/WAF/Firewall

Узагальнюючи, можна стверджувати, що хмарні сервіси екосистем AWS, Google Cloud та Microsoft Azure дають змогу проектувати модні e-commerce-платформи, в яких пікові навантаження під час «дропів» нових колекцій поглинаються CDN/edge-шаром завдяки агресивному кешуванню та глобальній маршрутизації запитів, а бекенд-API зберігають стабільну латентність, оскільки переважну частину статичного й медіаконтенту обробляють на «краю» мережі. Спостережність і безпека при цьому реалізуються безпосередньо на

інфраструктурному рівні (WAF, журнали доступу, метрики CDN), що спрощує дотримання вимог до захисту даних і репутації бренду. Додатково інструменти AI/ML хмарних провайдерів можуть бути тісно інтегровані з фронтом через персоналізовані вітрини, рекомендаційні модулі та динамічні промо-зони. Така архітектурна конфігурація є особливо придатною для fashion-ринку, що характеризується високою часткою мобільного трафіку, короткими життєвими циклами колекцій і підвищеними вимогами до швидкодії та візуальної якості інтерфейсу.

## **2.2 Обґрунтування вибору технологій для розробки e-commerce платформ**

Вибір технологічного стеку для e-commerce має ґрунтуватися на вимірюваних метриках користувацької якості (час до першого байта, Interaction to Next Paint), операційних показниках (пропускна здатність, надійність під піки попиту, частка повернень), вартості володіння та доступності компетенцій на ринку праці. У 2025 році в сегменті retail/fashion домінує триада мов, яка охоплює більшість виробничих інсталяцій: JavaScript (Node.js/TypeScript), Python (Django/FastAPI) та PHP (Laravel/Symfony). Професійні опитування розробників і галузеві огляди відзначають, що JavaScript зберігає найширшу крос-стекову спільноту; Python посилює позиції як універсальна мова для бекенду й аналітики; PHP залишається основою значної частки комерційних веб-сайтів, особливо там, де історично склалася відповідна інфраструктура [44; 45; 47; 48]. Рішення на JavaScript знижують когнітивну вартість проєкту завдяки єдиній мові на клієнті та сервері, що є доречним для швидких релізів каталогів і промокампаній. Python приваблює зрілими веб-фреймворками, потужною екосистемою бібліотек для аналітики й рекомендацій та активним розвитком інтерпретатора. PHP, попри конкуренцію, продовжує розвиватися у версіях 8.x, покращуючи продуктивність і модель безпеки та спираючись на широку базу розгорнутих систем [48; 49].

**2.2.1 Мови програмування для e-commerce платформ. JavaScript.** Стандарт ECMAScript 2025 фіксує чергове видання мови, яке стабілізує модульність, уточнює

роботу з базовими типами та асинхронними операціями і тим самим зменшує потребу в допоміжних бібліотеках як на клієнті, так і на сервері [16]. На рівні середовища виконання Node.js гілка 22.x перебуває у фазі довгострокової підтримки з плановою підтримкою до 2027 року та регулярними безпековими оновленнями [24]. Це дає можливість будувати низьколатентні API та подієві канали (Server-Sent Events, WebSocket) для live-інвентарю, нотифікацій під час «дропів» колекцій і керування чергами очікування. Єдина мова у фронтенді та бекенді спрощує розподіл завдань у командах, а також полегшує заміщення ролей; результати опитувань розробників підтверджують, що JavaScript залишається однією з найчастіше використовуваних мов у веб-проєктах 2025 року [44; 45].

Python.Актуальна релізна лінійка Python 3.13 упродовж 2025 року одержує регулярні оновлення, зокрема версію 3.13.9 від 14 жовтня 2025 р., що свідчить про стабільний цикл підтримки та активний розвиток інтерпретатора [46]. У центрі уваги цієї гілки — оптимізація продуктивності, зменшення накладних витрат виконання та поліпшення інструментів профілювання. Такі зміни напряду впливають на ефективність серверних процесів, які реалізують рекомендаційні системи, персоналізовані вітрини та ETL-пайплайни продуктової аналітики. Галузеві звіти та аналітика відкритого коду підкреслюють, що Python посилює статус основної мови для задач штучного інтелекту, аналітики даних і бекенд-сервісів, що робить його обґрунтованим вибором для модулів персоналізації, виявлення шахрайських транзакцій та безпосередньої інтеграції з ML-пайплайнами в межах одного технологічного стеку [44; 47].

PHP.Попри те, що частина нових проєктів обирає JavaScript або Python, PHP у 2025 році продовжує залишатися фундаментом великої кількості веб-сайтів з відомою серверною мовою. Статистика використання технологій для вебу свідчить, що PHP як і раніше домінує серед динамічних сайтів, причому частка інсталяцій на сучасних версіях 8.x неухильно зростає [48]. Розвиток гілки 8.4 супроводжується відчутними покращеннями продуктивності, розширенням мовних можливостей і посиленням безпеки, що детально відображено в офіційних релізних матеріалах та професійних оглядах [49]. Для e-commerce це означає прогнозовану експлуатацію

зрілих фреймворків на кшталт Laravel та Symfony, широку доступність готових інтеграцій із платіжними й логістичними провайдерами, а також значний пул спеціалістів, які підтримують уже розгорнуті системи.

Таблиця 2.9 узагальнює основні аргументи на користь кожної з трьох мов у контексті розробки e-commerce платформ.

Таблиця 2.9 - Обґрунтування вибору мови програмування для e-commerce

Критерій	JavaScript (Node.js/TypeScript)	Python (Django/FastAPI)	PHP (Laravel/Symfony)
Екосистема та кадри	Найбільша крос-стекова спільнота; легкий найм full-stack	Сильна база в AI/даних і бекенді; зростання частки розробників	Велика інстальована база в комерційному вебi; багато фахівців підтримки
Стандарт/платформа	ECMAScript 2025; Node 22.x LTS до 2027 р.	Регулярні оновлення 3.13.x у 2025 р.	Лінійка 8.x, активна експлуатація у продакшені
Продуктивність/латентність	Низькі затримки API; подієва модель I/O	Оптимізації інтерпретатора 3.13; стабільні серверні процеси	Відчутні покращення 8.4; зрілі кеші та опкоди
Архітектурні патерни	Edge/API-шлюзи, BFF, стрімінг подій	ML-сервіси, рекомендації, ETL/аналітика	Моноліти з чіткими DDD-модулями; headless via REST/GraphQL
Вартість володіння (TCO)	Єдина мова на клієнті/сервері знижує когнітивні витрати	Швидка розробка бек-офісу та аналітики	Багато готових пакетів для оплат/логістики
Типові ризики	Контроль якості прм-ланцюга; дисципліна перформансу	Версійність залежностей для ML/науки про дані	Спадщина старих інсталяцій; потреба в міграціях на 8.x

Практична інтерпретація полягає в тому, що для платформ, де головними показниками є швидкість розробки та низька латентність під часто оновлюваний каталог і live-механіки (дропа, віртуальні черги), доцільним є вибір JavaScript/TypeScript на базі Node.js. Якщо ключову роль відіграють персоналізація, протидія шахрайству й аналітика, об'єднані в єдиному стеку з ML-модулями, раціональним рішенням буде Python. Якщо ж планується використання вже наявної PHP-інфраструктури або робиться ставка на швидкий вихід у промислову експлуатацію з опорою на широку екосистему платіжних і логістичних інтеграцій, виправданим вибором є PHP-стек із сучасними версіями 8.x.

**2.2.2 Фреймворки для e-commerce платформ.** Архітектурний вибір фреймворку для модної e-commerce системи у 2025 році визначається вимогами до продуктивності мобільних інтерфейсів, можливостями потокової видачі HTML, зрілістю інструментів для роботи з формами та кошиком, наявністю галузевих компонентів для checkout і каталогів, а також відповідністю політикам безпеки. Для підходів на базі React у виробничій практиці характерним є поєднання з метарамками типу Next.js і спеціалізованими стеками для headless-вітрин, тоді як Angular пропонує інтегрований фреймворк з уніфікованою реактивністю на основі signals і режимом виконання без Zone.js [19; 20]. Django надає дисципліновану серверну основу з потужним ORM і регулярними безпековими оновленнями; у headless-сценаріях поширеним є використання платформ на кшталт Saleor з типізованим GraphQL-рівнем для storefront-ів [25; 51]. Такий ландшафт дозволяє поєднувати SSR і стримінг, сувору типізацію, керованість станом і транзакційність залежно від домінантних бізнес-показників каталогу.

У зв'язці React/Next.js/Hydrogen сторінки та макети в Next.js виконуються як серверні компоненти, що дає змогу виконувати завантаження даних і рендеринг на сервері з кешуванням і потоковою передачею HTML, а локальну інтерактивність забезпечують клієнтські компоненти [19; 50]. Це скорочує обсяг клієнтського JavaScript, покращує Interaction to Next Paint на мобільних пристроях і прискорює перший корисний рендер каталогу. Для комерційних вітрин застосовуються профільовані рішення на зразок Hydrogen, де пропонуються компоненти для

headless-storefront і пряма інтеграція зі Storefront API, що полегшує реалізацію кошика, PDP/PLP та промо-модулів [28; 50]. Узагальнене порівняння типових e-commerce-сценаріїв для основних фреймворків наведено в табл. 2.10.

Таблиця 2.10 - Порівняння фреймворків у типових e-commerce сценаріях [19; 20]

Критерій	React + Next.js / Hydrogen	Angular 20.x	Django 5.2 / Saleor
Візуальна продуктивність (мобільні)	RSC + стрімінг HTML, менші бандли; готові комерційні компоненти (Hydrogen)	Signals + zoneless зменшують перерендери; зручний контроль у DevTools	Серверний рендер Django, стабільна латентність API; фронтенд гнучко обирається
Форми/кошик	Бібліотеки на React; Hydrogen пропонує готові блоки кошика/checkout	Сильні реактивні форми; передбачувана зміна стану через signals	Django-форми для бек-офісу; storefront через GraphQL-шар Saleor
SEO/SSR	SSR/ISR «з коробки» у Next.js; тонке керування метаданими	SSR через Angular Universal; більша складність стрімінгу	SSR на стороні Django; headless-SSR реалізується у фронтенд-шарі
Headless-інтеграції	Hydrogen/Hydrogen React, Storefront API; легка побудова headless-вітрин	Офіційні SDK сторонніх сервісів; сувора структура модулів	Saleor GraphQL (публічні/приватні операції), кастомні адмін-додатки
Командна масштабованість	Чіткі межі Server/Client; велика спільнота, швидкий найм	Єдина парадигма для великої команди; «рамки» зменшують девіації	Дисциплінований бекенд; зрілий ORM/міграції, типізований GraphQL
Типові ризики	Дисципліна меж server/client; контроль прм-ланцюга	Міграції/навчальна крива; конфігурація SSR	Потрібен окремий фронтенд-шар; проектування схем GraphQL

В екосистемі Angular у 2025 році закріплено готовність до промислової експлуатації режиму без Zone.js і зміцнено роль signals як базового реактивного примітива. Це зменшує накладні витрати на відстеження змін, робить перерендеринг передбачуваним і спрощує профілювання реактивних графів у формах і кошику.

Додаткові інструменти на кшталт візуалізації роутера та signals у DevTools полегшують контроль продуктивності у складних вітринах, де затримки у фільтрах, сортуванні та виборі опцій SKU безпосередньо впливають на конверсію [20].

У напрямі Django і Django-базованих headless-платформ релізна лінійка 5.2 та подальші патчі 2025 року підтверджують стабільний цикл підтримки, що є критично важливим для транзакційних процесів, включно із замовленнями та поверненнями [25]. У headless-конфігураціях поширеним рішенням є Saleor як GraphQL-орієнтована платформа з API-only підходом, яка забезпечує типізований інтерфейс для storefront-ів та адміністративних застосунків і прискорює інтеграцію каталогу, кошика та промо-правил у гетерогенних фронтенд-стеках [51]. Евристика вибору фреймворку під ключові бізнес-КРІ наведена в табл. 2.11.

Таблиця 2.11 - Евристика вибору фреймворку під бізнес-КРІ [19; 20]

Домінантний КРІ / вимога	Рекомендація
Агресивні цілі INP/SEO у великому каталозі з частими «дропами»	React + Next.js; для Shopify-екосистеми - Hydrogen/Hydrogen React
Складні форми, кошики з правилами, корпоративна дисципліна	Angular 20.x із signals/zoneless
Суворі транзакційність, headless-операції через GraphQL, багатий бек-офіс	Django 5.2 як бекенд + Saleor як headless-ядро

За умов 2025 року та бізнес-вимог fashion-вітрин найбільш універсальною фронтенд-стратегією для динамічних каталогів є React з виробничим метарамком Next.js та, за необхідності, Hydrogen/Hydrogen React для headless-сценаріїв на Shopify [19; 50]. Angular доцільно обирати для інтерфейсів із високою насиченістю форм і жорсткими процесами у великій команді, де signals/zoneless забезпечують передбачувану продуктивність [20]. Django є раціональною основою бекенд-логіки та транзакцій з headless-проекцією через типізований GraphQL-шар Saleor; такий вибір спрощує інтеграцію з різними фронтендами, зберігаючи надійність і безпеку бекенду [25; 51].

За умов 2025 року та бізнес-вимог fashion-вітрин найбільш універсальною фронтенд-стратегією для динамічних каталогів є React з виробничим метарамком Next.js та, за необхідності, Hydrogen/Hydrogen React для headless-сценаріїв на

Shopify. Angular доцільно обирати для інтерфейсів із високою насиченістю форм і жорсткими процесами у великій команді, де signals/zoneless забезпечують передбачувану продуктивність. Django - раціональна основа бекенд-логіки та транзакцій з headless-проєкцією через типізований GraphQL-шар Saleor; такий вибір спрощує інтеграцію з різними фронтендами, зберігаючи надійність і безпеку бекенду.

**2.2.3 Бази даних для e-commerce платформ.** Архітектура зберігання даних у fashion e-commerce у 2025 році спирається на поєднання транзакційної надійності для замовлень і платежів, гнучких моделей для каталогу та медіа, а також можливостей горизонтального масштабування й федерації даних для аналітики та пікових подій («дропів» колекцій). Для цих вимог характерними є три зрілі СУБД: MySQL (гілка 8.4 LTS з удосконаленням HA-стеком InnoDB Cluster/ClusterSet), PostgreSQL (реліз 18 з поліпшеннями продуктивності та логічною реплікацією на рівні публікацій/підписок), MongoDB (кластер із багатодокументними ACID-транзакціями та сервісами Atlas на зразок Data Federation для доступу до об'єктного сховища) [29; 33]. Таке поєднання дає змогу розділяти навантаження між критично транзакційними потоками (кошик, оплата, повернення), змістовними довідниками SKU/варіантів і високочастотними читаннями каталогу з географічною близькістю до користувача.

MySQL 8.4 LTS застосовується як база для платежів, замовлень, інтеграцій з ERP і OMS завдяки зрілій реплікації, зрозумілим профілям продуктивності InnoDB та офіційним сценаріям високої доступності [29]. InnoDB Cluster забезпечує автоматичне членство в групі, детекцію відмов і фейловер, тоді як InnoDB ClusterSet додає сценарії катастрофостійкості завдяки пов'язаним кластерам у різних регіонах і окремому каналу реплікації між ними [30]. Для каталогу можливе поєднання реляційної схеми та зберігання атрибутів у JSON-стовпцях із індексацією окремих ключів, що корисно для багатовимірних фільтрів і фасетного пошуку без надмірної нормалізації.

PostgreSQL 18 у 2025 році підтверджує статус «policy-compliant» СУБД для фінансово чутливих процесів і складних транзакцій (повернення коштів, часткові

відвантаження, корекції) [31]. Для масштабування читань і міграцій схеми застосовується логічна реплікація за моделлю publication/subscription із можливістю каскадування, вибіркового тиражування таблиць і синхронізації на рівні підписника [32]. Тип jsonb і GIN-індекси забезпечують ефективний пошук за напівструктурованими атрибутами SKU та метаданими медіа, що особливо важливо для фільтрації великого каталогу.

MongoDB доречний для каталогу з частими змінами атрибутів, зберігання варіативних документів продуктів і швидкого рендерингу карток товарів, а також для потоків, де пріоритетом є швидке читання за ключами та агрегуваннями [33]. Наявність багатодокументних ACID-транзакцій робить можливим узгоджене оновлення кошика або інвентарю за кількома колекціями, а Atlas Data Federation дозволяє прозоро запитувати холодні шари даних у S3/Blob/GCS (наприклад, історію цін та архіви медіа), не дублюючи їх у «гарячому» кластері [34]. Це знижує ТСО у великих каталогах і пришвидшує побудову гібридних сховищ «операційні дані + архіви» [33; 34].

Узагальнене порівняння MySQL, PostgreSQL і MongoDB за ключовими функціональними вимогами fashion e-commerce наведено в табл. 2.12, а схемні патерни для модних каталогів і потоків замовлень – у табл. 2.13.

Таблиця 2.12 - Порівняння СУБД за функціональними вимогами fashion e-commerce [29; 33]

Вимога/функція	MySQL 8.4 (InnoDB)	PostgreSQL 18	MongoDB (Atlas)
Транзакційність і цілісність	Повна ACID; InnoDB; детерміновані профілі блокувань для замовлень/оплат	Повна ACID; розвинені рівні ізоляції; придатний для фінансових потоків	Багатоментні ACID-транзакції у кластері; зручно для кошика/інвентарю
Висока доступність (HA/DR)	InnoDB Cluster (автофейловер), ClusterSet для міжрегіональної DR	Фізична/логічна реплікація; гнучкі топології standby/subscriber	Replica-sets і sharding; керовані SLA у Atlas
Масштабування читань/записів	Read-replica; розподіл навантаження через роутер	Логічна реплікація publication/subscription; каскадування	Горизонтальний шардинг за ключем; auto-scaling у Atlas
Підтримка JSON/напівструктурованих атрибутів	JSON-стовпці з індексацією окремих полів; функціональні індекси	`jsonb` + GIN/виразні індекси; багаті оператори	Рідна документна модель; індекси за полями/масивами
Реплікація/міграції каталогу	Стандартна реплікація; безшовні оновлення через Group Replication	Тонке керування публікаціями/підписками для вибіркового тиражування	Глобальні кластери; online-архіви
Федерація/архіви/аналітика	Інтеграція з DWH через ETL	FDW / логічний стрім до DWH	Atlas Data Federation для S3/Blob/GCS без дублювання
Типові застосування	Замовлення, платежі, ERP/OMS інтеграції, транзакційні журнали	Фінансові корекції, повернення, rule-based промо, складні звіти	Каталог, атрибути SKU/варіантів, кеші карток, історія цін/медіа

Таблиця 2.13 - Схемні патерни для модних каталогів і потоків замовлень [29;

34]

Об'єкт/потік	Рекомендована модель	Аргументація	Примітки до реалізації
Каталог SKU/варіанти	Документна модель (MongoDB) або реляційна + JSON-атрибути	Гнучкі атрибути; швидкі читання карток; менше джоїнів	Індекси на фасетних атрибутах; TTL/архів для застарілих SKU
Кошик/чек-аут	Транзакційна реляційна (MySQL/PostgreSQL)	Узгодженість оплат; резерв інвентарю; PCI-процедури	Чіткі межі транзакцій; ідемпотентність webhook-ів
Повернення/корекції	Реляційна з історизацією (PostgreSQL)	Складні бізнес-правила; аудит	Логічна реплікація для аналітики без впливу на OLTP
Історія цін/медіа	Холодне сховище + федерація (Atlas Data Federation)	Дешеве зберігання; прозорий доступ без дублювання	Винос у S3/Blob/GCS із доступом через федерацію

**2.2.4 Інструменти для інтеграції e-commerce платформ з платіжними системами.** Інтеграція платіжної інфраструктури у модних e-commerce вітринах 2025 року має одночасно забезпечувати відповідність регуляторним вимогам (PCI DSS 4.0/4.0.1, SCA/3-D Secure 2 для ЄС), мінімізувати тертя в чек-ауті, підтримувати широкий спектр платіжних методів та надавати керовані засоби запобігання шахрайству. На практиці це реалізується двома взаємодоповнювальними підходами: використанням попередньо зібраних, хостованих платіжних форм (що дають змогу звузити зону відповідальності за PCI до профілю SAQ A за умови коректної інтеграції) та побудовою кастомних потоків із гнучким керуванням станами платежу, автентифікацією клієнта й постобробкою (повернення, часткові захоплення, рекурентні списання). Типовими інструментами виступають Stripe (модель Payment Intents та хостований Stripe Checkout) і PayPal (Orders API та Checkout), які покривають спектр сценаріїв від «low-code» впровадження до повністю керованої оркестрації платежів [52; 57].

Модель Stripe Payment Intents описує життєвий цикл платежу як явну машинку станів із подіями створення, підтвердження, автентифікації та захоплення, що дає змогу коректно реалізувати вимоги SCA/3-D Secure 2, а також складні потоки (часткове захоплення, повторні спроби, «off-session» списання) [52; 55]. Хостований Stripe Checkout надає попередньо зібрану платіжну сторінку з підтримкою карток, цифрових гаманців та локальних методів; він орієнтований на швидке впровадження з урахуванням PCI-вимог і готових UX-патернів для мобільних користувачів [53]. Оновлення 2025 року розширили можливості динамічної роботи з доставленням у Checkout (оновлення параметрів shipping options для вже створених сесій), що є критично важливим для fashion-каталогів із мінливими тарифами перевізників та складною географією відправлень [54]. Вимоги SCA забезпечуються завдяки вбудованій підтримці 3-D Secure 2 у платіжних потоках і Stripe Checkout, а також завдяки використанню гаманців із біометричною автентифікацією (Apple Pay, Google Pay), що знижує тертя в мобільних сценаріях [55]. Захист від шахрайства інтегровано через сервіс Stripe Radar, який застосовує машинне навчання, глобальні моделі ризику й настроювані правила для блокування та рев'ю високоризикових транзакцій [56].

У лінійці PayPal базовим інструментом є Orders API v2, який надає узагальнену абстракцію «замовлення» з операціями створення, авторизації та захоплення; інтеграція здійснюється через REST-інтерфейс із OAuth-автентифікацією застосунку та орієнтована на веб- і мобільні канали [57]. У межах продукту PayPal Checkout для продавців доступні сервіси Chargeback Protection та Effortless Chargeback Protection, які оцінюють ризик транзакції в реальному часі та, за виконання визначених критеріїв, покривають комісії й суму спору, що особливо важливо для категорій із високою частотою повернень та спірних замовлень (наприклад, преміальні модні товари) [58]. У гібридних вітринах PayPal доцільно комбінувати зі Stripe: платежі через брендований гаманець PayPal покривають аудиторію, що надає перевагу авторизації через акаунт, тоді як карткові та локальні платіжні методи, а також складні сценарії SCA можуть оброблятися через Stripe [52;

57]. Порівняльні характеристики Stripe і PayPal у типових сценаріях fashion e-commerce узагальнено в табл. 2.14.

Таблиця 2.14 - Порівняння Stripe і PayPal у типових сценаріях fashion e-commerce [52; 57]

Критерій	Stripe (Payment Intents / Checkout)	PayPal (Orders API / Checkout)
Тип інтеграції	Керована машинка станів платежу; хостований Checkout для «low-code»	Абстракція «Order» (create/authorize/capture); REST-інтеграція
Підтримка SCA/3-D Secure 2	Автоматичний виклик 3DS2 у Payment Intents; гаранті з біометрією	3DS2 в карткових потоках; авторизація через акаунт PayPal
Методи оплати	Картки, Apple Pay/Google Pay, локальні методи; Payment Links	PayPal Wallet, картки; альтернативні методи через акаунт
Захист від шахрайства	Radar (ML-правила, політики, аналітика)	Chargeback Protection / Effortless Chargeback Protection
Зміни в доставленні під час чек-ауту	Оновлення shipping options у вже створеній Checkout Session (03.2025)	Керування доставленням на стороні продавця/замовлення
PCI-контур	Хостований Checkout може звести зону до SAQ A за умов CSP/SRI	Хостовані віджети/редиректи також можуть підпадати під SAQ A
Операції після платежу	Часткове/відкладене захоплення; автоматизовані повернення; вебхуки	Часткове/повне захоплення; повернення; інструменти спорів

Відповідність PCI DSS 4.0/4.0.1. Для e-commerce-застосунків оновлені критерії SAQ A і SAQ A-EP деталізують умови, за яких використання хостованих платіжних сторінок або вбудованих форм стороннього провайдера (iframe) дійсно дозволяє звести зону відповідальності торговця до мінімуму [59; 60]. Неправильна класифікація інтеграції як SAQ A замість SAQ A-EP призводить до пропуску значної кількості контролів (регулярне сканування вразливостей, тестування на

проникнення, керування змінами, моніторинг скриптів на платіжній сторінці), що підвищує ризики компрометації платіжних даних [59; 60]. Для модних маркетплейсів із високою динамікою фронтенда доцільно поєднувати використання хостованих платіжних форм із технічними засобами контролю цілісності (CSP, SRI, інвентаризація скриптів) та чітким розмежуванням відповідальності між торговцем і платіжним провайдером. Типові патерни інтеграції та контролю ризиків наведено в табл. 2.15.

Таблиця 2.15 - Патерни інтеграції та контроль ризиків [53; 59]

Сценарій	Рекомендований патерн	Контроль ризиків і відповідності	Коментарі
Швидкий запуск колекції, мінімум розробки	Хостований Stripe Checkout або PayPal Checkout	SAQ A за умови CSP/SRI та інвентаризації скриптів; лог вебхуків	Підходить для MVP та промо-кампаній, особливо на мобільних
Мультирегіон з локальними методами і динамічною доставкою	Stripe Payment Intents + Checkout; оновлення shipping options	SCA/3DS2 за потреби; Radar-правила для піків попиту	Адаптація логістики під країну/службу доставки
Висока частка PayPal-користувачів і часті спори	PayPal Orders API з увімкненим Chargeback Protection	Політики диспутів, доказ доставки; журнали ідемпотентності	Зменшує фінансовий вплив чарджбеків у ризикових підкатегоріях
Єдиний чек-аут з вибором провайдера	Комбінований UX: PayPal кнопка + карткова форма Stripe	Розділення ключів, окремі вебхуки, централізований журнал подій	Оптимізація конверсії за уподобаннями клієнта

Підсумовуючи, для модних e-commerce-платформ у 2025 році доцільно комбінувати хостовані форми оплати для швидкого запуску колекцій та зменшення обсягу PCI-контролів із керованими потоками на базі машинки станів у сценаріях, де потрібні SCA/3-D Secure 2, часткові захоплення та рекурентні платежі. Stripe надає узгоджену модель Payment Intents і сервіс Radar, що знижує тертя в мобільних чек-аутах і забезпечує керованість ризиками [52; 56]. PayPal через Orders API та

продукти захисту від чарджбеків підвищує конверсію сегментів, які віддають перевагу брендованому гаманцю, та зменшує втрати від спорів [57; 58]. Вибір конкретної конфігурації має ґрунтуватися на домінантних KPI (INP/мобільні конверсії, частка чарджбеків, географія платіжних методів) і профілі відповідності (SAQ A проти SAQ A-EP) з урахуванням актуальних вимог PCI DSS 4.0/4.0.1 [59; 60].

## **2.3 Визначення архітектури e-commerce платформи**

Архітектурне проектування платформи електронної комерції у 2025 році ґрунтується на балансі між вимогами до продуктивності мобільних інтерфейсів, частотою оновлення каталогу, складністю транзакційних процесів (кошик, оплата, повернення), витратами на експлуатацію та зрілістю інженерної організації. Після періоду широкого ентузіазму навколо мікросервісних підходів у практиці впроваджень спостерігається зсув до більш поміркованих рішень: з акцентом на економічно доцільні компроміси, чіткі межі відповідальності доменів, стандартизовану спостережність і поетапну еволюцію від простіших конфігурацій до складніших лише за наявності обґрунтованої потреби [43; 61]. У модних вітринах, де трафік та навантаження мають виражену сезонність і суттєво залежать від «дропів» колекцій, архітектура повинна забезпечувати кероване масштабування «гарячих» доменів (пошук, каталог, оформлення замовлення), мінімізувати мережеві витрати в критичних транзакційних шляхах і водночас залишатися узгодженою з KPI конверсії та мобільними метриками взаємодії.

### **2.3.1 Вибір архітектурного підходу до побудови e-commerce платформ.**

Монолітна архітектура передбачає єдиний застосунок, у межах якого інтегровано користувацький інтерфейс, бізнес-логіку та доступ до даних. Її сильні сторони полягають у простішому розгортанні, коротшому циклі виведення продукту на ринок на стартових етапах, нижчих операційних витратах і відносно простому налагодженні. Ключові обмеження моноліту проявляються під час масштабування: виникає ризик «гарячих точок» у критичних модулях, формуються централізовані

залежності, що ускладнюють автономну еволюцію підсистем, а зі збільшенням доменної площини зростає складність підтримки [61; 63]. Для невеликих або середніх команд і сценаріїв із помірними піками попиту моноліт залишається раціональним вибором, особливо за умови модульної організації коду й дисципліни внутрішніх контрактів.

Мікросервісна архітектура розкладає систему на незалежні сервіси з чітко визначеними контрактами взаємодії. Переваги такого підходу полягають в автономному масштабуванні доменів (пошук, кошук, checkout), незалежних циклах релізів, локалізації збоїв у межах окремих сервісів і можливості встановлювати диференційовані SLO для різних підсистем [61; 62]. Водночас зростають витрати на мережеву координацію й узгодження даних, виникає потреба у сагах, транзакційних вихідних чергах та подійних шинах; підвищуються вимоги до спостережності, стандартизованих конвеєрів CI/CD і зрілої організації командної взаємодії. Економічна доцільність переходу до мікросервісів у ритейл-сценаріях обґрунтовується наявністю регулярних піків попиту, значної географічної присутності та потреби в незалежній еволюції ключових доменів [61; 62].

Гібридні конфігурації поєднують сильні сторони обох підходів. Модульний моноліт вводить явні внутрішні кордони між доменами в межах одного розгортання, зменшуючи ризики «великого моноліту без меж» і зберігаючи простоту операцій. Headless-підхід відокремлює вітрину від комерційного ядра, дозволяючи незалежно еволюціонувати веб-, мобільним та in-store інтерфейсам за стабільних API бекенду; у практиці впровадженнь headless часто слугує «мостом» між монолітними й мікросервісними стратегіями, зокрема у зв'язці з headless-платформами на кшталт Hydrogen або Saleor [50; 51]. Типовою траєкторією еволюції є старт із монолітом або модульним монолітом, поступовий перехід до headless для прискорення фронтенд-інновацій і вибіркове виділення сервісів у доменах із піковими навантаженнями, тоді як решта функціоналу зберігає переваги цілісного розгортання [61; 63].

Практичні критерії вибору можна підсумувати так. Якщо метою є швидке розгортання MVP або пілоту за умов помірних пікових навантажень, доцільними є монолітні або модульно-монолітні конфігурації. Якщо очікуються часті «дропаи»

колекцій, регіональні піки трафіку та наявні окремі «гарячі» домени (пошук, checkout, аналітика), виправданим стає гібридний варіант із headless-вітриною та виділеними сервісами в навантажених зонах. За наявності незалежних продуктових потоків роботи великої кількості інженерів і жорстких SLO для окремих доменів доречно застосовувати повноцінну мікросервісну архітектуру з формалізованими контрактами, механізмами узгодженості та платформеними практиками спостережності [61–63].

**2.3.2 Функціональні вимоги.** Функціональні вимоги до платформи електронної комерції у 2025 році визначаються взаємозв'язком між вимірюваною якістю взаємодії на мобільних пристроях, повнотою каталогів і атрибутів, гнучкістю платіжних сценаріїв та надійністю процесів доставлення і повернень. Базою технічної специфікації є сучасні можливості веб-платформи: зокрема інтерактивність за метрикою Interaction to Next Paint як стабільним Core Web Vital та узгоджені реалізації CSS-модулів, що забезпечують передбачуваність адаптивних інтерфейсів і дизайн-систем у великих командах [18; 17]. Виробничі рекомендації 2025 року для вітрин із високою частотою оновлення колекцій підкреслюють прямий зв'язок продуктивності фронтенда з конверсією й ефективністю рекламних витрат, що посилює пріоритет оптимізації візуального рендерингу та мережевих шляхів постачання медіаконтенту [41; 64].

Користувацький інтерфейс і взаємодія. Інтерфейс вважається придатним до промислової експлуатації, коли забезпечує стабільні пороги Interaction to Next Paint (не вище 200 мс на 75-му перцентилі сесій) у ключових сценаріях: пошук і фільтрування, перегляд картки товару, модифікація кошика, оформлення замовлення [18; 64]. Для досягнення цих порогів доцільними є серверний рендеринг і потокова видача HTML у каталозі та на сторінках товару, агресивне скорочення обсягу клієнтського JavaScript, використання сучасних можливостей CSS (контейнерні запити, селектор `:has()`) та дисципліна роботи із зображеннями (адаптивні розміри, прогресивні формати). З погляду UX, галузеві дослідження фіксують ключові причини залишення кошика: надто довгий або складний чек-аут,

примусова реєстрація, непрозорі витрати на доставлення й повернення, а також повільність інтерфейсу [64; 65].

Управління продуктами і пошук. Каталог має підтримувати багату атрибутизацію (розміри, склад, колірні варіанти, стан на складі), локалізацію контенту та мультирегіональні ціни. Фасетний пошук і фільтрування повинні забезпечувати сталу затримку відповіді за пікового навантаження, а інтерфейси — синхронізуватися зі станом URL для коректної індексації та спільного використання посилань. Рекомендаційні блоки та динамічні колекції доцільно кешувати на сервері й оновлювати подійно, щоб не погіршувати пороги взаємодії в мобільних сценаріях [36; 38].

Кошик і оформлення замовлення. Функціональні вимоги включають підтримку гостьового оформлення, автоматичне збереження стану кошика між сесіями, прозору калькуляцію податків і вартості доставлення, а також синхронну валідацію форм без блокуючих перезавантажень. Дослідження галузі показують, що скорочення кількості кроків оформлення, зниження обсягу обов'язкових полів і раннє відображення повної вартості замовлення зменшують частку залишених кошиків і прискорюють завершення транзакції [64; 65].

Оплата. Платіжні сценарії повинні підтримувати авторизацію та захоплення коштів, часткові повернення, повторні списання для підписок і Strong Customer Authentication у юрисдикціях, де це є обов'язковою вимогою. Модель з автоматом станів платежу (Payment Intents) зменшує кількість помилок і забезпечує коректну взаємодію з 3-D Secure 2, тоді як хостовані чек-аути дають змогу звузити контур відповідності PCI для сценаріїв швидкого запуску колекцій або пікових кампаній [52; 55]. Наявність цифрових гаманців (Apple Pay, Google Pay) та локальних методів оплати є необхідною умовою для підвищення мобільної конверсії у 2025 році [52; 55].

Доставлення та повернення. Функціональні вимоги включають відображення термінів і тарифів у реальному часі, вибір пунктів видачі, відкладене комплектування окремих позицій і прозору оркестрацію зворотної логістики. Аналітика ринку показує стійко високі частки повернень в онлайн-каналі, особливо

у fashion-категорії, що створює значний операційний та фінансовий тиск на рітейлерів [9; 66]. Платформа повинна підтримувати самостійне створення етикеток, трекінг статусів RMA, часткові повернення та автоматизоване вирахування комісій за повернення (де це застосовно). Інформаційні блоки щодо політик доставлення та повернення мають бути видимими до початку оформлення, що знижує ймовірність переривання процесу на пізніх стадіях [65; 66].

Облікові записи й приватність. Система має підтримувати гостей замовлення з можливістю перетворення їх на повноцінні акаунти, багатофакторну автентифікацію для кабінету користувача, перегляд історії замовлень і ініціювання запитів на видалення персональних даних. Зберігання платіжних методів делегується платіжним провайдером; на стороні платформи зберігаються лише токени, що мінімізує зону відповідальності за PCI та спрощує відповідність актуальним редакціям стандарту [52; 58].

Контент, локалізація та SEO. Платформа повинна забезпечувати багатомовний контент, локальні валюти та формати представлення цін, канонічні посилання для варіантів товарів і структуровані дані для каталогу. Пріоритет серверного рендерингу критичних сторінок (каталог, сторінка товару) та дисципліноване керування метаданими відповідають практичним рекомендаціям сучасних фреймворків для забезпечення стабільної індексації та органічного трафіку [19; 22].

Аналітика, експерименти та спостережність. Функціональні вимоги включають подієві журнали оформлення замовлення (кроки, причини відмов), можливість A/B-експериментів без деградації продуктивності, а також наскрізні кореляційні ідентифікатори між фронтендом, API-шлюзом і службами виконання замовлень для реконструкції користувацьких шляхів та швидкої локалізації збоїв [41; 43].

Безпека та відповідність. Серед обов'язкових вимог — захист від автоматизованих зловживань (боти, карт-тестинг), контроль зовнішніх скриптів на сторінках оформлення (через політики CSP і механізми SRI) та відокремлення платіжних форм у хостованих віджетах або іфреймах. Поєднання цих заходів із коректною класифікацією інтеграції за профілями SAQ A / SAQ A-EP відповідає

практикам зниження зони відповідальності торговця за PCI DSS 4.0/4.0.1 у 2025 році [37; 59].

**2.3.3 Нефункціональні вимоги до e-commerce платформи.** Нефункціональні вимоги у 2025 році визначають здатність платформи стабільно досягати бізнес-КРІ (конверсія, частота повторних покупок, вартість залучення клієнта) за умов пікових навантажень і високої частки мобільного трафіку. Основними напрямками є швидкодія (продуктивність і спостережність), безпека (відповідність вимогам і стійкість до зловживань) та масштабованість/надійність (еластичність і відновлюваність).

Швидкодія та спостережність. Базовою метрикою чутливості інтерфейсу для веб-додатків у 2025 році є Interaction to Next Paint (INP) як стабільний Core Web Vital, рекомендована ціль якого для торгових вітрин становить не більше 200 мс на 75-му перцентилі сесій; значення понад 500 мс вважаються незадовільними [18; 64]. На архітектурному рівні це зумовлює пріоритет серверного рендерингу та потокової видачі HTML для каталогу й сторінок товару, агресивне скорочення обсягу клієнтського JavaScript, використання edge-кешування повних сторінок і CDN для медіаконтенту, а також застосування контентних політик зображень з адаптивними розмірами та прогресивними форматами [36; 38]. Оглядові матеріали щодо продуктивності в 2025 році фіксують прямий позитивний вплив повносторінкового кешування та оптимізованих мережевих маршрутів на всі три Core Web Vitals (LCP, INP, CLS), що є критично важливим для мобільної конверсії у модних каталогах [41; 64]. Для операційного контролю доцільно впроваджувати RUM-моніторинг (дані реальних користувачів) з можливістю географічної сегментації та кореляції подій клієнтського й серверного трактів; сучасні платформи доставки й хостингу пропонують вбудовану аналітику з підходом privacy-by-design, що полегшує розгортання спостережності без суттєвого впливу на продуктивність [67; 43]. Практичні настанови хмарних провайдерів рекомендують систематично застосовувати шаблони продуктивності (правильний вибір обчислювальних профілів, керовані кеші, асинхронну обробку, граничні служби на «edge») і регулярно переглядати їх у межах Well-Architected-оцінювань [43; 70].

Безпека та відповідність. Для веб-платформ із картковими платежами у 2025 році актуальними є вимоги PCI DSS v4.0/4.0.1 і профілі самооцінювання SAQ A/SAQ A-EP: якщо сторінка містить вбудовану платіжну форму постачальника (у тому числі через iframe), торговець має підтвердити відсутність вразливостей, що дозволяють змінювати поведінку платіжного контенту; сценарії редиректів повного аутсорсингу оплати класифікуються окремо [59; 60]. На технічному рівні це вимагає суворих політик Content Security Policy і Subresource Integrity для сторінок оформлення замовлення, інвентаризації сторонніх скриптів, розділення доменів довіри та зберігання платіжних даних виключно у провайдера (на стороні платформи – лише токени) [37; 52]. Додатково необхідно забезпечити захист від автоматизованих зловживань (боти, карт-тестинг), централізований секрет-менеджмент і шифрування даних у спокої та під час передавання, єдиний аудит доступу та немодифіковані журнали подій. Для профілактики типових уразливостей доцільно орієнтуватися на сучасні контрольні списки тестування веб-безпеки та узагальнені огляди пріоритетних класів ризиків (наприклад, OWASP Web Security Testing Guide і OWASP Top 10), адаптуючи їх до домену електронної комерції [68; 69].

Масштабованість, надійність і відновлюваність. Платформа повинна еластично масштабуватися по горизонталі під регулярні «дропи» колекцій і сезонні піки попиту, застосовуючи авто-скейлінг за багатофакторними сигналами (CPU, латентність, глибина черг) із захистом від «гойдання» (thrashing) за допомогою гістерезису та «warm pools». Галузеві рекомендації 2025 року для надійного масштабування радять комбінувати шаблони «queue-based load leveling», «circuit breaker», «bulkhead isolation», використовувати керовані кеші й репліки для розвантаження баз даних та чітко розділяти синхронні й асинхронні шляхи в процесах оформлення [70; 43]. Документація Well-Architected для провідних хмарних платформ доповнена прикладами для SaaS-навантажень: акцент робиться на формулюванні вимірюваних SLO (латентність, error budget), автоматизованому тестуванню відновлюваності (chaos-практики), багатозонному розміщенні та регіональному відмовостійкому дизайні з чітко визначеними RTO/RPO [43; 71]. Для

високочутливих потоків (платежі, підтвердження замовлення) виправданими є ідемпотентні API й гарантії «at-least-once» з дедуплікацією подій, а для каталогу та пошуку – агресивне використання CDN/edge-кешів і попереднього обчислення агрегатів, щоб обмежити «хвіст» латентності під час пікових навантажень [36; 38].

Сумарно нефункціональні вимоги фіксують цільові пороги взаємодії на мобільних ( $INP \leq 200$  мс на P75 у ключових сценаріях), вимоги до безпечного оформлення відповідно до PCI DSS v4.0/4.0.1 із коректною класифікацією SAQ A/SAQ A-EP та застосуванням CSP/SRI й токенизації, а також обов'язковість еластичного масштабування з формалізованими SLO і розгорнутою спостережністю. Узгодження цих вимог з рекомендаціями Well-Architected-підходів і сучасними практиками веб-продуктивності у 2025 році дозволяє підтримувати стабільну конверсію та керовану собівартість під змінні навантаження модних вітрин [18; 43].

## 3 РОЗРОБКА ПРАКТИЧНИХ РЕКОМЕНДАЦІЙ ЩОДО ВПРОВАДЖЕННЯ ІНТЕЛЕКТУАЛЬНИХ НОВИХ ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ E-COMMERCE ПЛАТФОРМИ ДЛЯ ПРОДАЖУ ОДЯГУ

### 3.1 Загальні рекомендації щодо впровадження видів тестування e-commerce платформи

Організація тестування має виходити з вимірюваних показників якості користувацької взаємодії на мобільних пристроях, передусім із фокусу на Interaction to Next Paint як стабільному Core Web Vital, а також урахувати пікові навантаження під час «дропів» колекцій, сезонність попиту та підвищені вимоги до безпеки платежів [18; 64]. Комплексна стратегія тестування для fashion e-commerce доцільно базується на чотирьох рівнях — модульному, інтеграційному, системному та навантажувальному — із чітко визначеними критеріями входу/виходу, трасованістю дефектів і поєднанням лабораторних та польових вимірювань продуктивності [72]. Узагальнений огляд рівнів тестування, їхніх цілей та типових кейсів наведено в табл. 3.1.

Таблиця 3.1 - Матриця видів тестування для fashion e-commerce [72; 75]

Рівень тестування	Мета	Обсяг і типові кейси	Критерії виходу	Інструменти/практики
Модульне	Коректність функцій і компонентів без зовнішніх залежностей	Валідація форм; логіка кошика; UI-події; відображення цін/знижок	Покриття критичних гілок; відсутність регресій у формах/кошиках	Компонентні тести у реальному браузері; ізоляція залежностей
Інтеграційне	Узгодженість модулів і API (фронтенд ↔ шлюз ↔ сервіси)	Повторні спроби платежу; зміна доставки; конфлікти інвентарю; ідемпотентність	Сталі SLO відповіді; коректні компенсації; відсутність незахоплених винятків	Контракти, стабли/моки; журналювання кореляційних ID

Системне	Перевірка повного користувацького шляху в умовах, наближених до реальних	Пошук → PDP → кошик → чек-аут → повернення; багатомовність/мультивалютність	INP ≤ 200 мс (P75); SEO-готовність; відсутність блокерів у чек-ауті	Поєднання лабораторних і польових вимірювань Core Web Vitals
Навантажувальне	Стійкість та еластичність під піки попиту та «дропи»	«Спайки»; довгі плато; деградація сервісів; профілі з високою мобільною часткою	Дотримання SLO; контрольований хвіст латентності; безпомилкові транзакції	кб-скрипти; інтеграція з observability-платформами

Модульне тестування. Метою модульного тестування є верифікація коректності окремих функцій, сервісів і компонентів інтерфейсу в ізоляції від зовнішніх залежностей, що зменшує вартість дефекту та прискорює цикл змін. Для фронтенда доцільно застосовувати компонентні тести у реальному браузері, які перевіряють поведінку UI-елементів (кнопки, селектори, форми кошика, відображення цін і знижок) зі справжнім рендерингом та обробкою подій, а не лише в абстрагованих DOM-середовищах [72; 73]. Такий підхід знижує ризики регресій у сценаріях фільтрації, перемикання варіантів товару та модифікації кошика і підвищує передбачуваність релізів. Критеріями виходу є повне покриття критичних гілок бізнес-логіки кошика та валідації форм, а також відсутність відомих регресій.

Інтеграційне тестування. Завдання інтеграційного тестування полягає у перевірці взаємодії модулів «фронтенд ↔ API-шлюз ↔ служби кошика/каталогу/платежів» із акцентом на ідемпотентність, коректний облік помилок і відновлюваність. Для таких сценаріїв доцільно застосовувати стабільні тестові подвійники (mock/stub) зовнішніх сервісів і мінімально необхідну кількість «живих» інтеграцій в ізольованому середовищі, наближеному до продуктивного. Обов'язковими є кейси повторної відправки платежу після таймауту, оновлення вартості доставлення на пізніх кроках оформлення, а також конфлікти інвентарю під піковим попитом. Критеріями виходу слугують відсутність незахоплених винятків, коректна компенсація помилок (retries, скасування, повернення коштів) і сталі часи відповіді в межах погоджених SLO [62; 72].

Системне тестування. На системному рівні перевіряється повний користувацький шлях — від пошуку та фасетного фільтрування до оформлення замовлення й ініціювання повернення — в умовах, максимально наближених до реальних. Необхідне поєднання лабораторних вимірювань (інструменти аудиту продуктивності) та даних реальних користувачів (RUM), оскільки перші пояснюють причини деградацій, а другі відображають фактичний досвід у різних мережевих та девайсних умовах [18; 67]. У 2025 році практики тестування рекомендують розглядати Core Web Vitals як ключові контрольні метрики, при цьому особливу увагу приділяти INP. Критеріями виходу є стабільні пороги INP не вище 200 мс (P75) для основних сценаріїв (пошук, перегляд картки товару, робота з кошиком, чек-аут), коректна індексація SEO-критичних сторінок і відсутність блокувальних дефектів у процесі оформлення замовлення.

Навантажувальне (продуктивнісне) тестування. Метою навантажувального тестування є підтвердження еластичності й стабільності системи під очікуваними та граничними навантаженнями, включно зі «спайками» під час дропів колекцій. Рекомендовано моделювати профілі трафіку, що віддзеркалюють типові піки для сегмента «Мода»: короткі інтенсивні сплески з високою часткою мобільних сеансів і значною кількістю одночасних операцій фільтрування та перегляду карток товарів. Інструменти класу k6 дозволяють описувати сценарії у вигляді коду, масштабувати їх до розподіленого запуску та інтегрувати з платформами спостережності, що спрощує кореляцію показників застосунку й інфраструктури [75; 76]. Критеріями виходу є дотримання SLO щодо затримки та частки помилок, відсутність неконтрольованого зростання черг, контрольований «хвіст» латентності та передбачувана деградація без втрати транзакційності. Формалізовані критерії входу/виходу та цільові SLO для кожного рівня тестування узагальнені в табл. 3.2.

Таблиця 3.2 - Критерії входу/виходу та SLO-цілі для тестування e-commerce платформи [72; 75]

Рівень	Критерії входу	Критерії виходу	SLO / Метрики
Модульне	Завершені специфікації компонентів; ізольовані залежності; тестові дані	100% покриття критичних гілок; відсутність відомих регресій	Час прогону < 5 хв на PR; стабільність тестів $\geq 99\%$
Інтеграційне	Опубліковані контракти API; стабільність зовнішніх сервісів	Ідемпотентність підтверджена; коректна компенсація помилок	P50 латентності < 300 мс; відсоток помилок < 0.5%
Системне	Оточення, наближене до продакшену; наповнені каталоги та акаунти	INP $\leq 200$ мс (P75) у ключових сценаріях; SEO-готовність	Core Web Vitals: LCP $\leq 2.5$ с, INP $\leq 200$ мс, CLS $\leq 0.1$
Навантажувальне	Моделі трафіку та цільові піки; інтеграція з моніторингом	Дотримання SLO під піком; відсутність неконтрольованих черг	P95 латентності < 800 мс; відсоток помилок < 1%; відсутність втрати даних

Практичні рекомендації щодо процесу.

По-перше, доцільно уніфікувати критерії входу/виходу на кожному рівні тестування з урахуванням бізнес-KPI та SLO, а також включити моніторинг Core Web Vitals у конвеєр релізів.

По-друге, слід забезпечити трасованість дефектів від модульного рівня до польових метрик, використовуючи кореляційні ідентифікатори в логах браузера, API та бекенд-сервісів.

По-третє, профілі навантаження необхідно будувати на підставі аналітики реальної поведінки користувачів (розподіл пристроїв, географія, типові сценарії) і регулярно переглядати їх за результатами пікових кампаній та «дропів» колекцій [64; 76].

### 3.2 Практичні рекомендації щодо використання інструментів для тестування

Узгоджена стратегія тестування e-commerce платформи повинна розподіляти відповідальність між рівнями перевірок та інструментами так, щоб мінімізувати вартість дефекту й забезпечити стабільність релізів під пікові навантаження. Практичний поділ ролей може бути таким: Jest покриває модульні й компонентні перевірки інтерфейсів і бізнес-логіки на рівні коду; Postman забезпечує функціональне, регресійне та контрактне тестування API з керованими середовищами та колекціями; Selenium використовується для наскрізних користувацьких сценаріїв у реальних браузерах, включно з перевіркою інтеграції фронтенд/бекенд/платіжні потоки [72; 75]. Нижче наведено рекомендації щодо кожного інструмента та організації їх спільної роботи в конвеєрі збірки й постачання.

Selenium: наскрізні браузерні сценарії та регресійні «рейки». Доцільно будувати набір стабільних end-to-end сценаріїв, що віддзеркалюють критичні для конверсії шляхи: пошук і фасетне фільтрування, перегляд картки товару, додавання до кошика, оформлення замовлення з вибором параметрів доставлення й оплати, ініціація повернення. Для підвищення відтворюваності тестів рекомендовано уніфікувати стратегії локаторів (data-атрибути замість селекторів класів), застосовувати явні очікування замість глобальних затримок, фіксувати початковий стан (вхідні дані, куки, локаль, валюта) через фікстури. Масштабування під різні браузери й паралельний прогін досягається за допомогою Selenium Grid, ізоляції сесій та пулу попередньо «підігрітих» екземплярів браузерів [80; 81]. Антифрагментність забезпечується через дрібні кроки сценарію (один намір користувача — одна перевірка), централізацію утиліт взаємодії з інтерфейсом і збирання артефактів на дефекті (скріншоти, логи консолі, мережеві трейси). У регресійні «рейки» доцільно включати лише сценарії з максимальною бізнес-цінністю та мінімальною флаксістю, тоді як експлораторні або рідкісні кейси варто виносити в окремі періодичні прогони.

Jest: модульні та компонентні перевірки, що знімають тиск з end-to-end. Для фронтенда доцільно комбінувати модульні тести бізнес-логіки (перетворення цін, правила вибору варіантів, розрахунок промо) і компонентні тести інтерфейсу в середовищі, наближеному до реального DOM. Рекомендовано ізолювати мережеві виклики через моки або тестові подвійники, перевіряючи лише функціональність, за яку відповідає конкретний модуль чи компонент; побічні ефекти та часові залежності слід робити керованими (fake timers, контроль часу) [72; 77]. Покриття має фокусуватися на критичних гілках кошика, валідаторах форм і логіці відображення варіантів товарів; суто косметичні аспекти розмітки не повинні домінувати у звітності. Для великих кодових баз доцільно застосовувати інкрементальні прогони за змінами, кешування результатів і розпаралелювання, що скорочує час зворотного зв'язку під час активних «дропів» колекцій [77; 74]. На рівні типів бажано спиратися на статичну перевірку (TypeScript) і тестувати публічні API модулів, а не внутрішні деталі реалізації.

Postman: функціональні, регресійні та контрактні перевірки API. Практичний мінімум включає структуру колекцій за доменами (каталог, кошик, оплати, доставлення), окремі середовища для кожного контуру (локальне, тестове, передпродакшн, продакшн-моніторинг), параметризацію змінних доступу, даних та локалей. Скрипти на етапах pre-request і tests мають перевіряти коди статусу, схеми відповідей (JSON Schema), ідемпотентність у сценаріях повторної відправки, коректність заголовків кешування й обмежень швидкості, а також дотримання бізнес-інваріантів (узгодженість сум у підсумках замовлення) [78]. Для регресії важливо включати кейси граничних станів (відсутність товару, конфлікт кошика, помилки платіжних постачальників) і повернення. У CI/CD середовищі колекції доцільно запускати через Newman з машинними звітами для подальшого агрегування в системах спостережності [79]. Контрактну узгодженість API із клієнтами слід підтримувати перевітками відповідності відповідей опублікованим схемам (OpenAPI/JSON Schema), щоб зміни в API не порушували роботу фронтенда й мобільних застосунків.

Оркестрація в CI/CD і поділ відповідальності. Доцільно впроваджувати три «пояси якості». Перший пояс — швидкі Jest-прогони для кожного коміту й запиту на злиття; вони забезпечують миттєвий зворотний зв'язок і блокують очевидні регресії. Другий пояс — запуск Postman/Newman-колекцій для кожного злиття в релізну гілку та за нічними розкладами на передпродакшні середовища, що підтверджує контрактну й функціональну цілісність доменів [72; 78]. Третій пояс — Selenium-рейки як обов'язковий етап перед релізом і за розкладом, із паралельним прогоном у кількох браузерях і регіонах. Усі пояси мають публікувати артефакти (звіти, скриншоти, логи, метрики часу відповіді) з кореляційними ідентифікаторами, що прискорює аналіз причин дефектів.

Дані, середовища та стабільність тестів. Для Selenium і Postman важливими є керований тестовий каталог і підхід «дані як код»: фабрики тестових сутностей, які створюють продукти, користувачів і замовлення у передбачуваних станах, ізоляція тестів через унікальні префікси або namespace-и, регламент очищення артефактів та життєвий цикл секретів і токенів доступу [75; 76]. Для зменшення флакисності в Selenium слід застосовувати прості патерни очікувань, уникати залежності від нестабільних анімацій і відкладених рендерів та переносити нестабільні перевірки на нижчі рівні, якщо вони не додають цінності на e2e-рівні. У Postman варто уніфікувати тайм-аути, ретраї та політики «fail-fast», щоб не маскувати деградацію продуктивності.

Метрики якості та пороги. Рекомендовано фіксувати цільові пороги часу прогону для Jest і частки нестабільних тестів у Selenium, а також показники відповідності схемам і середні часи відповіді для Postman-колекцій. Для e-commerce із високою мобільною часткою корисно включати у звітність кореляцію між лабораторними показниками (латентність API, стабільність e2e-тестів) і польовими метриками взаємодії інтерфейсу (Core Web Vitals), щоб ранжувати дефекти за впливом на конверсію [18; 64].

Сумарно Jest знімає основну частину ризиків на рівні коду й компонентів, Postman гарантує цілісність і стабільність API-контрактів та бізнес-правил, а Selenium підтверджує, що критичні користувацькі шляхи працюють у реальних

браузерах за різних умов мережі та локалей. Така стратифікація забезпечує передбачуваність релізів, скорочує час реакції на інциденти та знижує довгострокову вартість забезпечення якості [72; 75].

### **3.3 Практичні рекомендації щодо розробки інтерфейсу користувача (UI/UX дизайн)**

Розробка інтерфейсу модної e-commerce платформи повинна спиратися на поєднання вимірюваної швидкодії (насамперед Interaction to Next Paint як стабільного Core Web Vital), перевірених галузевих патернів мобільного UX, сучасних принципів доступності та локалізації під ключові регіони збуту (ЄС та Україна) [18; 64]. Практичні настанови нижче систематизовано за основними ділянками користувацького шляху.

Навігація та пошук. Верхня навігація повинна забезпечувати безперервний перехід між категоріями, доступ до пошуку «з першого торкання» та явну індикацію активного розділу. На мобільних пристроях доцільна комбінація постійно видимого поля пошуку й жестів доступу до меню, тоді як каталог має підтримувати фасетні фільтри з прогресивним розкриттям (від пріоритетних до другорядних атрибутів). Дослідження UX для e-commerce фіксують, що чітка ієрархія навігації, «липкі» панелі дій і видимий стан фільтрів зменшують когнітивне навантаження та підвищують глибину перегляду каталогу на смартфонах [82; 83].

Каталог та керування фільтрами. Для списків товарів доцільно використовувати інкрементальне підвантаження (infinite scroll) із помітними «якорями» повернення до початку й збереженням стану фільтрів у URL, що полегшує обмін посиланнями та індексацію. Групування фільтрів за суттєвими ознаками (розмір, колір, наявність, склад тканини) у поєднанні з видимим лічильником застосованих критеріїв і швидким скиданням підвищує контроль користувача над результатами та зменшує кількість «порожніх» сторінок результатів [82; 83].

Сторінка товару (PDP). Оптимальною є послідовність «головне фото — назва/ціна — ключові атрибути — кнопка додавання в кошик», із ранньою видимістю розмірної сітки та інформації про наявність. Рекомендовано реалізувати галерею з жєстами, збільшенням деталей, швидкими перемикачами кольорів і варіантів, а також показом на моделі із зазначенням параметрів тіла. Важливу роль відіграють довірчі сигнали: короткі підсумки відгуків і оглядів, у тому числі у форматі II-підсумків із посиланням на повні тексти, що прискорює оцінку придатності товару без втрати прозорості джерел [84; 42].

Розмір та «примірка». Інтерактивні розмірні гіди (запит базових параметрів, типу фігури), лаконічні підказки («модель зросту 175 см у розмірі M», «посадка oversize») і впізнавані піктограми посадки знижують невизначеність та ймовірність повернень. UX-дослідження показують, що винесення ключових підказок над «fold» та доступ до гідів без залишення PDP статистично підвищують частку додавань у кошик для модних категорій [83; 84].

Кошик і оформлення замовлення (checkout). Доцільно забезпечити можливість гостьового оформлення, мінімальну кількість кроків, синхронну валідацію полів і раннє відображення загальної вартості з доставленням і податками. Для мобільних інтерфейсів рекомендовано «липку» панель підсумку та кнопку продовження, а також підтримку нативних гаманців (Apple Pay, Google Pay) і локальних методів оплати. Аналітичні дослідження 2025 року підкреслюють прямий зв'язок між спрощенням форми, скороченням часу до візуального відгуку та зменшенням відмов у чек-ауті [64; 65].

Швидкодія й миттєвий відгук. Рекомендовані пороги Interaction to Next Paint для торгових фронтендів становлять не більше 200 мс на 75-му перцентилі сесій; значення понад 500 мс розглядаються як незадовільні [18]. Для їх досягнення слід мінімізувати обсяг клієнтського JavaScript, виконувати рендеринг і завантаження даних на сервері з потоковою передачею HTML, застосовувати суворе керування пріоритетами завантаження медіа та забезпечувати негайний візуальний фідбек на дії користувача (зміна стану кнопок, «skeleton»-заглушки, optimistic UI) [18; 64].

Доступність і універсність дизайну. У 2025 році доцільно орієнтуватися як на усталені критерії WCAG 2.1, так і на «живі» напрацювання WCAG 3.0, що розширюють спектр врахованих потреб користувачів [86; 87]. Практичний мінімум включає достатній контраст елементів керування, логічну послідовність фокуса, видимі стани навігації, коректні підписи для допоміжних технологій, зрозумілу індикацію помилок із підказками щодо виправлення та підтримку масштабування інтерфейсу без втрати функціональності.

Контент, локалізація і мікрокопі. Для ЄС та України необхідні багатомовний інтерфейс, коректні формати валют і дат, локальні правові повідомлення (умови повернень, обробка персональних даних). На рівні мікрокопі доцільно уникати неоднозначних назв атрибутів, а інформацію про доставлення та повернення подавати компактними блоками без вимушеного виходу з основного потоку оформлення [82; 86].

Експерименти та спостережність. Впровадження A/B-експериментів слід поєднувати з польовим моніторингом Core Web Vitals, інакше позитивний ефект від зміни дизайну може бути нівельований деградацією чутливості інтерфейсу. Рекомендовано зберігати причинно-наслідкові зв'язки між варіантами експериментів, бізнес-подіями (додавання в кошик, успішні платежі) та метриками швидкодії, щоб ранжувати варіанти за впливом на конверсію [88; 64].

Сумарно оптимальний UI/UX-профіль для fashion e-commerce у 2025 році поєднує мобільно-перший дизайн із дисципліною швидкодії за INP, керовані фасети й зрозумілу сторінку товару з ранньою видимістю варіантів і розмірів, мінімалістичний гостьовий чек-аут із нативними гаманцями, доступний і локалізований контент, а також безперервні експерименти, підкріплені польовими метриками [18; 64].

### **3.4 Практичні рекомендації щодо реалізації функціональності**

Реєстрація та автентифікація користувачів. Для зниження тертя під час входу й одночасного підвищення захищеності від фішингу доцільно впроваджувати

пароль-less-підхід на основі passkeys (WebAuthn/FIDO2), які вже підтримуються сучасними браузером та мобільними ОС. Реєстрація відбувається через створення пари ключів і прив'язку публічного ключа до облікового запису; автентифікація – через криптографічний виклик із перевіркою підпису на сервері. Рекомендовано забезпечити резервні канали (пароль + одноразові коди, e-mail/OTP) і поступову міграцію користувачів на passkeys за моделлю «opt-in». Для пароля як резервного механізму слід застосовувати сучасні підходи до збереження (Argon2id, унікальна сіль, налаштовані параметри пам'яті та ітерацій, контроль витоків) і дисципліну керування сесіями (короткоживучі токени, відокремлення анонімної та автентикованої сесій, регенерація ідентифікаторів після входу, обмеження одночасних сесій, політики idle-timeout). Для захисту від автоматизованих атак додаються ліміти на спроби входу, оцінка ризику за пристроєм/геолокацією та моніторинг підозрілої активності. З погляду інтерфейсу важливо забезпечити «миттєвий» візуальний відгук (стани кнопок, індикатори процесу, підказки виправлення помилок) без виходу за цільові пороги Interaction to Next Paint на мобільних. Практичні орієнтири щодо passkeys і керування сесіями наведено в актуальних рекомендаціях 2025 року [89–92].

Управління каталогом товарів. Для модної категорії доцільно чітко розмежувати сутності «товар» (Product), «варіант» (Variant) і «SKU», забезпечивши окремий канонічний ідентифікатор моделі та інваріантні атрибути (бренд, склад тканини, категорія) на відміну від варіативних (розмір, колір, довжина). Рекомендовано підтримувати повний життєвий цикл атрибутів, перелік обов'язкових полів для кожного каналу збуту (зображення, GTIN/MPN за наявності, таблиця розмірів), а також придатність даних до зовнішніх фідів (Merchant Center) з коректною нормалізацією варіантів apparel. Для пошуку й фільтрації доцільно застосовувати фасетну модель з індексацією варіативних атрибутів, збереженням стану фільтрів у URL, дедуплікацією за артикулом і періодичною валідацією результатів. У складних каталогах варто поєднувати ручну таксономію з ML-евристиками для динамічного формування підказок і фасетів. Оновлені на 2025 рік

вимоги до фідів apparel та рекомендації щодо фасетного пошуку детально окреслено у відповідних матеріалах Google і Elastic Labs [93–94].

Кошик і оформлення замовлення. Кошик повинен виступати «серверним джерелом істини» з ідемпотентними операціями додавання/видалення, рекалькуляцією цін і промо-правил на сервері та синхронізацією між пристроями. Для забезпечення швидкодії слід мінімізувати обсяг клієнтського JavaScript, застосовувати потокову видачу HTML та оптимістичний UI, що дає миттєвий відгук на мобільних. Потік оформлення замовлення доцільно будувати максимально коротким, із підтримкою гостьового режиму, нативних гаманців, синхронної валідації полів, раннього розрахунку вартості доставки/податків і чітких повідомлень про помилки. На рівні API критичною є ідемпотентність (наприклад, через ключі ідемпотентності у запитах створення платежів/замовлень), що гарантує безпечні повторні спроби за умов збоїв мережі [61; 70].

Інтеграція платіжних систем. Для карткових платежів у ЄС необхідно підтримувати Strong Customer Authentication з автоматичним маршрутом через 3-D Secure 2, використовуючи готові потоки платіжних провайдерів із машинним визначенням винятків і мінімізацією тертя. У Stripe практичним вибором є Payment Intents/Checkout із вбудованою підтримкою SCA, у PayPal – модель Orders API з режимами «authorize→capture» або «capture» залежно від логістичних сценаріїв (перевірка наявності, відвантаження з відстрочкою). В обох випадках обов'язковими є ідемпотентні запити, надійна обробка вебхуків і прозора синхронізація статусів платежів зі статусами замовлень, включно з частковими поверненнями. Для періодів пікового навантаження потрібні політики повторних спроб для 5xx/таймаутів, детерміновані state-машини на стороні бекенду та моніторинг аномалій конверсії за VIN/країною [52–60].

Система доставки та відстеження замовлень. З огляду на крос-бордерні сценарії та потребу в прозорості «останньої милі» доцільно інтегруватися з уніфікованими API перевізників для генерації ярликів, розрахунку тарифів, отримання webhooks і відстеження подій (створено, у транзиті, невдала спроба, вручено). З експлуатаційної точки зору важливими є нормалізація трекінг-статусів

до єдиного словника, push-повідомлення для користувачів, SLA-ескалації при «застряганні» відправлень, інтеграція з процесами повернень і автоматичне оновлення прогнозованого часу доставки (ETA). Бажано підтримувати кількох перевізників для маршрутизації за країною/зоною та резервування на випадок збоїв окремого постачальника. Актуальні настанови 2025 року наведено у документації DHL, UPS та FedEx щодо API для електронної комерції та трекінгу відправлень [7; 101–102].

Підсумок. Практична реалізація перелічених функцій у 2025 році спирається на поєднання пароль-less-автентифікації з сучасним керуванням сесіями, структурованого каталогу з фасетним пошуком, ідемпотентного кошика та чек-ауту, інтегрованих SCA-сумісних платежів і уніфікованого трекінгу доставки. Така композиція мінімізує тертя у ключових точках воронки, знижує ризики подвійних операцій і забезпечує прозорий ланцюг «оплата → відвантаження → доставка», що безпосередньо корелює з конверсійними та ретеншн-метриками у fashion-сегменті [7; 52–60; 61; 70; 89–94; 101–102].

### **3.5 Практичні рекомендації щодо інтеграції зі сторонніми сервісами**

Інтеграція комунікаційних сервісів у модній електронній комерції у 2025 році повинна підпорядковуватися двом взаємопов'язаним цілям: підвищенню конверсій і довгостроковій лояльності та дотриманню вимог надійності, приватності й галузевих правил доставності повідомлень. Технічно це зводиться до дисципліни згоди користувача, автентифікації відправника, ідемпотентного обміну з постачальниками та наскрізної спостережності за доставленням повідомлень. У цьому підрозділі систематизовано практичні настанови для двох ключових каналів – SMS і електронної пошти – з огляду на актуальні вимоги 2025 року.

SMS як транзакційний і маркетинговий канал. Для надсилання A2P-трафіку в ключових ринках, зокрема у США, обов'язковою є реєстрація брэнда і кампаній у 10DLC із наданням зразків контенту, опису процесу згоди та відписки, а також посилення на умови сервісу; нерегістровний трафік блокується або штрафується, що

напряму впливає на доставність і собівартість комунікацій. Незалежно від юрисдикції згода користувача має бути явною, відтворюваною та журналюваною: фіксуються час і джерело opt-in, ідентифікатор пристрою або IP, канал отримання згоди (форма, чек-аут, ключове слово). Повідомлення повинні підтримувати стандартні ключові слова для відписки й довідки (STOP, HELP), а тексти підписок – містити ідентифікацію бренду, орієнтовну частоту надсилання і застереження щодо тарифікації. На рівні технічної реалізації доцільно розділяти потоки: транзакційні повідомлення (OTP, підтвердження замовлення, зміна статусу доставки) надсилаються з виділених ідентифікаторів і мають пріоритет у чергах, тоді як маркетингові кампанії – з окремих номерів. Дотримання throughput-лімітів, «тихих годин» для немиттєвих розсилок, уникнення контенту-тригерів і використання власних доменів для скорочених URL підвищують репутацію відправника й зменшують частку відмов кар'єрів. Аналітика має охоплювати частки доставлення, причини hard/soft-відмов, відгук на рівні сесій сайту, а також гармонізацію з іншими каналами через уніфіковані ідентифікатори кампаній і ліміти частоти, щоб уникати дублювання комунікацій. Усі ці вимоги узгоджуються з актуальними рекомендаціями щодо A2P-меседжингу та 10DLC і галузевими «Messaging Principles & Best Practices» [95–96].

Електронна пошта як канал транзакційних сповіщень і життєвих циклів клієнта. У 2025 році провідні поштові провайдери висувають консолідовані вимоги до технічної автентифікації та якості трафіку: для відправників зі значними обсягами обов'язкова наявність SPF і DKIM для домену From, запис DMARC із доменною узгодженістю, механізм «one-click unsubscribe», контроль рівня скарг на спам і дотримання допустимих порогів відмов; політика DMARC у режимі спостереження (p=none) застосовується для аудиту джерел і поступового переходу до quarantine/reject [97–99]. Одночасно з цим рекомендовано впроваджувати BIMI після досягнення режиму DMARC-enforce, що покращує впізнаваність бренду в інбоксі й позитивно корелює з інбокс-плейсментом [100]. З погляду експлуатації ефективність каналу визначається не лише контентом, а й «гігієною» бази: регулярним видаленням мертвих адрес, сегментацією за стадіями життєвого циклу

(new, at-risk, lapsed), тригерними ланцюжками (abandoned cart/browse, back-in-stock, post-purchase) і політиками приглушення після скарг, щоб зменшити ризик втрати доменної репутації [97–98]. Технічна інтеграція з ESP повинна забезпечувати ідемпотентні виклики API, явне оброблення 4xx/5xx, збір і реакцію на події доставлення, відкриттів, кліків, скарг і відмов, а також синхронізацію suppression-списків між середовищами [97–99]. Налаштування автентифікації слід виконувати на власному домені з коректними DNS-записами, а не довільними піддоменами постачальника, що підвищує контроль і прогнозованість доставності [97–99].

Архітектурні засади інтеграції. Для узгодженої роботи каналів рекомендовано застосовувати подійно-орієнтовану інтеграцію з операційними системами (OMS, платіжні шлюзи, служби доставки). Події рівня «order.created», «payment.captured», «shipment.in\_transit», «refund.issued» породжують транзакційні повідомлення в email або SMS; усі виклики до провайдерів здійснюються з ключами ідемпотентності, а зміни статусів відстежуються через вебхуки постачальника з надійним повторенням у разі таймаутів. З метою керованості комунікацій доцільно вводити автомати станів на боці бекенду для кожного «ланцюжка» (наприклад, від відправлення до вручення), що уникає дублювання або втрати повідомлень. Крос-канальна оркестрація має спиратися на єдину модель пріоритетів: транзакційні події переважають маркетинг; для часово чутливих сповіщень доцільно використовувати SMS як резервний канал, але лише за наявності згоди й у рамках обмежень частоти, щоб не знижувати довіру користувачів і операторів [61; 70; 95–97].

Безпека, приватність і відповідність. Оброблення персональних даних для розсилок здійснюється за принципом мінімізації: у комунікаційних системах зберігаються лише ідентифікатори користувача, токени для взаємодії з ESP і агреговані показники; платіжні реквізити не потрапляють до каналу розсилок, а секрети інтеграцій зберігаються у менеджері секретів із ротацією і аудитом доступу. Для email обов'язкові SPF, DKIM і DMARC; для SMS – ведення журналів згоди і відписки, відповідність політикам операторів, недопущення «snowshoeing» і підміни ідентифікаторів. Усі сторінки, де збирається згода, повинні містити прозорі повідомлення про політику приватності та мету обробки даних, а також надавати

простий механізм відкликання згоди в один клік; це одночасно покращує регуляторну відповідність і доставки повідомлень завдяки зниженню скарг [95–99].

Спостережність і вимірювання ефекту. Впровадження наскрізних ідентифікаторів кампаній і UTM-міток у посиланнях дозволяє пов'язувати події каналу з поведінкою на сайті та бізнес-результатами. На рівні операційної звітності канали відстежують базові показники доставлення, відмов і скарг; на рівні маркетингу – відгук, конверсії та внесок у дохід із урахуванням атрибуції. Практики 2025 року кладуть у основу експериментального дизайну контрольні групи-«холдбеки» та суворий облік накладання каналів, щоб уникати переоцінювання впливу одного каналу за рахунок іншого. У публічних поштових провайдерів і операторів зв'язку регулярні огляди й рекомендації підкреслюють важливість підтримання низьких complaint-rate, правильних заголовків і технічної автентифікації як ключових факторів довготривалої доставності [88; 96–98].

Сумарно, інтеграція сторонніх сервісів комунікацій у модній електронній комерції має спиратися на документовану згоду, технічну автентифікацію відправника та стандартизовані подійні інтерфейси з ідемпотентністю й обробкою вебхуків. Дисципліна розділення транзакційних і маркетингових потоків, уніфіковані обмеження частоти, «гігієна» списків і контроль доменної репутації забезпечують передбачувану доставність, тоді як подійна оркестрація і наскрізна аналітика пов'язують канал із реальними конверсіями. Такий підхід дозволяє одночасно підвищувати дохід і знижувати ризики регуляторних порушень та деградації доставності, що є критично важливим на ринках із високою мобільною часткою та сезонними піками попиту [61; 70; 88; 95–100].

## ВИСНОВКИ

Проведене дослідження підтвердило, що конкурентоспроможність fashion e-commerce у 2025 році визначається поєднанням трьох чинників: технологічної адекватності обраного стеку, архітектурної дисципліни та операційної зрілості процесів. Ринок демонструє стійке зростання, водночас ускладнюються вимоги споживачів до персоналізації, швидкості взаємодії, прозорості логістики та якості сервісу. Для українського та східноєвропейського контексту додаткового значення набувають гнучкість ланцюгів поставок, готовність до пікових навантажень і використання омніканальних форматів продажів.

*Стек технологій.* Триада мов програмування JavaScript, Python та PHP залишається практично достатньою для побудови сучасних e-commerce систем. Вибір має ґрунтуватися на бізнес-KPI, доступності кадрових ресурсів і вимогах до інтеграцій. JavaScript доцільний для швидких релізів і низької латентності на одному стеку клієнт-сервер; Python є оптимальним для модулів аналітики, персоналізації та інтеграції з ML-пайплайнами; PHP виправданий у середовищах із наявною інфраструктурою і широкою екосистемою комерційних інтеграцій.

*Фронтенд-стратегія.* Для каталожних і SEO-критичних сторінок пріоритетними є серверний рендеринг та потокова передача HTML; локальна інтерактивність реалізується вибірковими клієнтськими компонентами. Застосування сучасних можливостей CSS, включно з контейнерними запитами та селектором `has`, у поєднанні з дисципліною скорочення клієнтського JavaScript, є необхідною передумовою досягнення цільових значень показників користувацької якості на мобільних пристроях. Практичними метарамками для виробничого застосування є React з Next.js та спеціалізовані стекові рішення для headless-сценаріїв.

*Архітектурний підхід.* Для нових продуктів доцільно стартувати з модульного моноліту з чіткими доменними межами та контрактами, що знижує складність і TCO

на ранніх етапах. Еволюція до headless-моделі або мікросервісів виправдана за наявності відокремлюваних доменів, підвищеної частоти релізів та необхідності незалежного масштабування. Рекомендовано використовувати підхід API-first, шар BFF для клієнтів і подієві інтеграції там, де це спрощує зв'язність.

*Бекенд-фреймворки.* Поєднання React/Next.js зі спеціалізованими комерційними стековими рішеннями забезпечує швидку реалізацію вітрин, кошика та checkout. Для сценаріїв з високими вимогами до транзакційності і керованості даних виправдане застосування Django як бекенд-основи та headless-рішень з типізованими API.

*Дані та сховища.* Реляційні СУБД доцільні для транзакційних контурів і фінансового обліку; документоорієнтовані бази забезпечують гнучкість для каталогів, атрибутів та контенту. Вибір між MySQL, PostgreSQL та MongoDB має враховувати вимоги до цілісності, аналітичних запитів, реплікації, шардінгу та безперервних оновлень. Рекомендовано проектувати змішані схеми з чіткими SLA для кожного контуру даних.

*Хмара та доставка контенту.* Використання керованих сервісів основних провайдерів у поєднанні з CDN та edge-кешуванням підвищує стійкість до пікових навантажень і скорочує час до першого байта. Орієнтиром має бути відповідність принципам добре спроектованих архітектур щодо продуктивності, надійності та безпеки.

*Платежі та відповідність.* Інтеграції зі Stripe і PayPal доцільно реалізовувати через сучасні API з дотриманням принципів ідемпотентності, підтримкою багатофакторної автентифікації та 3-D Secure. Мінімізація зони відповідності стандартам безпеки платіжних даних досягається через токенізацію та делегування зберігання чутливих даних платіжним провайдерам.

*Логістика та повернення.* Інтеграції зі службами доставки і відстеження забезпечують прозорість ланцюга поставок, а спрощені політики повернень знижують бар'єри до покупки. Для fashion-категорії управління поверненнями є ключовим фактором економіки замовлення, тому моделі прогнозування та диференційовані правила мають бути складовою продуктового ядра.

*Тестування та якість.* Повний контур якості включає модульні, інтеграційні, системні та навантажувальні випробування з автоматизацією у конвеєрі CI/CD. Інструменти для UI-, API- та перфоманс-тестування слід поєднувати з польовою аналітикою користувачьких показників. Якісні ворота мають опиратися на цільові пороги Core Web Vitals і ділові KPI конверсій.

*UI/UX та доступність.* Мобільний сценарій, зрозумілий пошук і фільтри, якісний візуальний контент, прогнозування розміру і прозорий checkout без зайвих кроків істотно впливають на конверсію. Дотримання міжнародних рекомендацій з доступності має розглядатися як необхідна умова якості інтерфейсу.

*Безпека та ідентифікація.* Актуальні практики включають застосування сучасних методів автентифікації, захист сесій, контроль доступу на рівні доменів, шифрування даних, виявлення ботів і захист від типових уразливостей. Регулярні аудити і тестування на проникнення мають бути складовою життєвого циклу.

*Штучний інтелект.* Найвищу віддачу сьогодні забезпечують персоналізація, візуальний пошук, прогнозування попиту та покращення якості контенту. Водночас необхідно враховувати ризики приватності, упередженості моделей і вплив на операційні витрати, запроваджуючи прозорі політики даних і контроль якості моделей.

### **Практичні рекомендації**

*Архітектура і поетапне впровадження.* Розпочати з модульного моноліту з виділеними доменними модулями та шаром API-first. Передбачити можливість еволюції до headless-моделі та відокремлення сервісів, критичних до масштабування. Запровадити шар BFF для мобільного та веб-клієнтів.

*Фронтенд і продуктивність.* Для SEO-критичних сторінок застосовувати серверний рендеринг зі стрімінгом. Для інтерфейсів з високою взаємодією використовувати реактивні примітиви фреймворків. Стандартизувати вимоги до Core Web Vitals і включити їх до критеріїв приймання релізів. Забезпечити edge-кешування та оптимізацію графіки і шрифтів.

*Дані та інтеграції.* Розділити транзакційний контур і контент-каталог. Для транзакцій застосовувати реляційну базу з надійною реплікацією, для каталогу та

пошуку — документоорієнтоване або гібридне сховище з індексуванням і кешуванням. Усі інтеграції з платіжними і логістичними провайдерами реалізовувати через ідемпотентні API з чіткими SLA та моніторингом.

*Платежі та комплаєнс.* Вибудувати процеси обробки платежів із підтримкою 3-D Secure і багатофакторної автентифікації. Мінімізувати зберігання платіжних даних у власній системі. Регулярно перевіряти відповідність актуальним вимогам галузевих стандартів і банківських регуляцій.

*Тестування і нагляд.* Створити стратегію тестування, що охоплює модульні, інтеграційні, системні та навантажувальні випробування, а також моніторинг у реальному часі. Упровадити автоматизовані регресійні набори для кошика і checkout. Встановити пороги якості й блокери релізів за результатами тестів і польових метрик.

*UX і доступність.* Стандартизувати шаблони сторінок каталогу і картки товару. Забезпечити коректні фільтри, варіативність розмірів і підказки щодо посадки. Скоротити кількість кроків у процесі оформлення замовлення. Гарантувати відповідність вимогам доступності та локалізації.

*Логістика та повернення.* Інтегрувати відстеження відправлень, автоматизувати повідомлення клієнтів, впровадити прозору політику повернень. Застосувати прогнозні моделі для управління запасами та зниження частки повернень.

*Безпека.* Регулярно проводити аналіз ризиків, перевірки коду та тестування на проникнення. Упровадити сучасні механізми автентифікації, контроль сесій і журналювання безпеки. Організувати процес швидкого встановлення виправлень уразливостей.

*Дані та AI.* Вибудувати етичну і правову рамку використання даних, включно з механізмами згоди та прозорими налаштуваннями приватності. Для моделей персоналізації забезпечити контроль якості, спостережність і переоцінювання на основі актуальних даних.

*Операційна зрілість.* Формалізувати процеси розробки і релізів, прийняти підхід trunk-based із короткими ітераціями, запровадити канаркові або поетапні

викати. Забезпечити безперервне навчання команди щодо нових технологій і галузевих практик.

**Ключові показники оцінювання.** Рекомендовано формально закріпити перелік цільових показників: час до першого байта, показники Core Web Vitals, доступність сервісу, конверсія в покупку, середній чек, частка повернень, частка успішних платежів, швидкість оформлення замовлення, точність прогнозів попиту та задоволеність клієнтів.

### **Обмеження та напрями подальших досліджень**

Застосування інтелектуальних технологій у fashion-категорії стикається з ризиками упередженості даних, чутливістю приватності, витратами на інфраструктуру і підготовку даних. Перспективними напрямками подальших робіт є поєднання моделей персоналізації з віртуальними примірочними, підвищення точності розмірних рекомендацій, зменшення частки повернень завдяки покращенню візуалізації і якості контенту, а також подальша автоматизація логістики на основі подієвих архітектур.

Підсумовуючи, запропонована сукупність технологічних і процесних рішень створює передумови для побудови масштабованої, безпечної та економічно ефективної платформи продажу одягу. Дотримання викладених рекомендацій дозволить прискорити виведення продукту на ринок, забезпечити стабільну якість взаємодії користувачів і підвищити бізнес-результати за рахунок зростання конверсії, зниження операційних ризиків і сталого управління витратами.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Jacobs, F. R., & Chase, R. B. (2023). Operations and Supply Chain Management (16th ed.). New York: McGraw-Hill Education.
2. McKinsey & Company. (2021). Jumpstarting Value Creation with Data and Analytics in Fashion and Luxury. [Електронний ресурс]. Режим доступу: <https://www.mckinsey.com/industries/retail/our-insights/jumpstarting-value-creation-with-data-and-analytics-in-fashion-and-luxury>
3. Kim, J., & Kim, J. (2020). The Impact of Augmented Reality on Online Fashion Shopping: A Consumer Perspective. Journal of Retailing and Consumer Services, 55, 102073.
4. Pantano, E., & Priporas, C. V. (2016). The Effect of Mobile Retailing on Consumers' Purchasing Experiences: A Dynamic Perspective. Computers in Human Behavior, 61, 548–555.
5. The Wall Street Journal. (2024). How Tech Helped Levi's Ride the 'Baggy Jeans' Trend. [Електронний ресурс]. Режим доступу: <https://www.wsj.com/articles/how-tech-helped-levis-ride-the-baggy-jeans-trend-f290721d>
6. Amazon. (2023). How Amazon Deploys Robots in Its Operations Facilities. [Електронний ресурс]. Режим доступу: <https://www.aboutamazon.com/news/operations/how-amazon-deploys-robots-in-its-operations-facilities>
7. DHL. (2022). DHL eCommerce Solutions API Integration. [Електронний ресурс]. Режим доступу: <https://www.dhl.com/us-en/home/ecommerce-solutions.html>
8. Supply Chain Dive. (2024). UPS' Happy Returns taps into Geek+ sorting robotics. [Електронний ресурс]. Режим доступу: <https://www.supplychaindive.com/news/ups-happy-returns-uses-geek-plus-robotics/730957/>

9. UPS. (2023). Ecommerce Returns Management with Happy Returns. [Електронний ресурс]. Режим доступу: <https://www.ups.com/us/en/ups-happyreturns>
10. Business Insider. (2025). How Amazon Uses Robots to Sort and Transport Packages in Warehouses. [Електронний ресурс]. Режим доступу: <https://www.businessinsider.com/how-amazon-uses-robots-sort-transport-packages-warehouses-2025-2>
11. Авторська розробка, створено за допомогою програми Microsoft PowerPoint (схема / ілюстрація №1).
12. Авторська розробка, створено за допомогою програми Microsoft PowerPoint (схема / ілюстрація №2).
13. Авторська розробка, створено за допомогою програми Microsoft PowerPoint (схема / ілюстрація №3).
14. Авторська розробка, створено за допомогою програми Microsoft PowerPoint (схема / ілюстрація №4).
15. Авторська розробка, створено за допомогою програми Microsoft PowerPoint (схема / ілюстрація №5).
16. Ecma International. (2025). ECMAScript® 2025 Language Specification (16th ed.). [Електронний ресурс]. Режим доступу: <https://ecma-international.org/publications-and-standards/standards/ecma-262/>
17. World Wide Web Consortium (W3C). (2025). CSS Snapshot 2025. [Електронний ресурс]. Режим доступу: <https://www.w3.org/TR/css-2025/>
18. Web.dev. (2024). Optimize Interaction to Next Paint (INP). [Електронний ресурс]. Режим доступу: <https://web.dev/articles/optimize-inp>
19. Vercel / Next.js. (2025). React Foundations: Server and Client Components. [Електронний ресурс]. Режим доступу: <https://nextjs.org/learn/react-foundations/server-and-client-components>
20. Angular Team. (2025). Angular Documentation: Signals and Zoneless Change Detection. [Електронний ресурс]. Режим доступу: <https://angular.dev>
21. Vue.js Team. (2025). Vue 3 Documentation: Composition API Guide. [Електронний ресурс]. Режим доступу: <https://vuejs.org/guide>

- 22.Nuxt Team. (2025). Nuxt 3 Documentation. [Електронний ресурс]. Режим доступу: <https://nuxt.com/docs>
- 23.Аналітичний центр e-commerce України. (2025). Огляд ринку електронної комерції України за перше півріччя 2025 року. [Електронний ресурс]. Режим доступу: [https://pro-consulting.ua/ua/base/analiz-rynka-ukrainy?level1=retail\\_trading&level2=electronnaya\\_kommerciya&stat=1](https://pro-consulting.ua/ua/base/analiz-rynka-ukrainy?level1=retail_trading&level2=electronnaya_kommerciya&stat=1)
- 24.Node.js. (2025). Node.js Release Schedule (LTS 22.x). [Електронний ресурс]. Режим доступу: <https://nodejs.org/en/about/releases>
- 25.Django Software Foundation. (2024). Django 5.1 release notes. [Електронний ресурс]. Режим доступу: <https://docs.djangoproject.com/en/5.1/releases/5.1/>
- 26.Laravel. (2025). Laravel Forge – Server Management for PHP Applications. [Електронний ресурс]. Режим доступу: <https://forge.laravel.com>
- 27.Ruby on Rails. (2024). Rails 8.0.0 alpha1 Release Notes. [Електронний ресурс]. Режим доступу: <https://github.com/rails/rails/releases/tag/v8.0.0.alpha1>
- 28.Shopify. (2024). GraphQL Admin API Reference. [Електронний ресурс]. Режим доступу: <https://shopify.dev/docs/api/admin-graphql>
- 29.Oracle / MySQL. (2025). MySQL 8.4 LTS Release Notes. [Електронний ресурс]. Режим доступу: <https://dev.mysql.com/doc/relnotes/mysql/8.4/en/>
- 30.Oracle / MySQL. (2025). MySQL InnoDB Cluster and ClusterSet Overview. [Електронний ресурс]. Режим доступу: <https://dev.mysql.com/doc/mysql-shell/8.4/en/mysql-innodb-cluster.html>
- 31.PostgreSQL Global Development Group. (2025). PostgreSQL 18 Documentation: Overview and Architecture. [Електронний ресурс]. Режим доступу: <https://www.postgresql.org/docs/current/index.html>
- 32.PostgreSQL Global Development Group. (2025). PostgreSQL 18 Documentation: Logical Replication. [Електронний ресурс]. Режим доступу: <https://www.postgresql.org/docs/current/logical-replication.html>
- 33.MongoDB Inc. (2025). MongoDB Atlas – Overview. [Електронний ресурс]. Режим доступу: <https://www.mongodb.com/atlas>

34. MongoDB Inc. (2025). MongoDB Atlas: Data Federation and Online Archive. [Электронный ресурс]. Режим доступа: <https://www.mongodb.com/docs/atlas/data-federation/>
35. MongoDB Inc. (2025). Application Modernization Program (AMP) – Customer Case Studies for Retail and Ecommerce. [Электронный ресурс]. Режим доступа: <https://www.mongodb.com/modernization>
36. Amazon Web Services. (2024). Amazon CloudFront Developer Guide: Introduction to CloudFront. [Электронный ресурс]. Режим доступа: <https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html>
37. Amazon Web Services. (2023). AWS WAF Developer Guide. [Электронный ресурс]. Режим доступа: <https://docs.aws.amazon.com/waf/latest/developerguide/>
38. Google Cloud. (2024). Cloud CDN Overview. [Электронный ресурс]. Режим доступа: <https://cloud.google.com/cdn/docs/overview>
39. Google Cloud. (2024). Media CDN Overview. [Электронный ресурс]. Режим доступа: <https://cloud.google.com/media-cdn/docs/overview>
40. Microsoft Corporation. (2024). Azure Front Door Standard and Premium Documentation. [Электронный ресурс]. Режим доступа: <https://learn.microsoft.com/azure/frontdoor/>
41. Amazon Web Services. (2023). Prime Day 2023 Powered by AWS – All the Numbers. [Электронный ресурс]. Режим доступа: <https://aws.amazon.com/blogs/aws/prime-day-2023-powered-by-aws-all-the-numbers/>
42. Google Cloud. (2024). Retail Solutions with Vertex AI. [Электронный ресурс]. Режим доступа: <https://cloud.google.com/solutions/retail>
43. Amazon Web Services. (2024). AWS Well-Architected Framework. [Электронный ресурс]. Режим доступа: <https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html>
44. Stack Overflow. (2024). 2024 Stack Overflow Developer Survey. [Электронный ресурс]. Режим доступа: <https://survey.stackoverflow.co/2024/>

45. JetBrains. (2024). The State of Developer Ecosystem 2024. [Электронный ресурс].  
Режим доступа: <https://www.jetbrains.com/lp/devecosystem-2024/>
46. Python Software Foundation. (2025). Python Release Python 3.13.9. [Электронный ресурс]. Режим доступа: <https://www.python.org/downloads/release/python-3139/>
47. GitHub. (2024). The State of the Octoverse 2024. [Электронный ресурс]. Режим доступа: <https://octoverse.github.com/>
48. W3Techs. (2025). Usage Statistics and Market Share of PHP for Websites. [Электронный ресурс]. Режим доступа: <https://w3techs.com/technologies/details/pl-php>
49. PHP Group. (2024). PHP 8.4 Release Announcement. [Электронный ресурс].  
Режим доступа: <https://www.php.net/releases/8.4/en.php>
50. Shopify. (2025). Hydrogen – React-based Framework for Headless Storefronts. [Электронный ресурс]. Режим доступа: <https://hydrogen.shopify.dev>
51. Saleor Commerce. (2025). Saleor Documentation: GraphQL-First Headless Commerce Platform. [Электронный ресурс]. Режим доступа: <https://docs.saleor.io>
52. Stripe. (2025). The Payment Intents API. [Электронный ресурс]. Режим доступа: <https://docs.stripe.com/payments/payment-intents>
53. Stripe. (2025). Use a prebuilt Stripe-hosted Checkout page. [Электронный ресурс].  
Режим доступа: <https://docs.stripe.com/payments/checkout>
54. Stripe. (2025). Update shipping options for Stripe Checkout Sessions. [Электронный ресурс]. Режим доступа: [https://docs.stripe.com/api/checkout/sessions/object#checkout\\_session\\_object-shipping\\_options](https://docs.stripe.com/api/checkout/sessions/object#checkout_session_object-shipping_options)
55. Stripe. (2025). 3D Secure authentication. [Электронный ресурс]. Режим доступа: <https://docs.stripe.com/payments/3d-secure>
56. Stripe. (2025). Stripe Radar – Protect your business against fraud. [Электронный ресурс]. Режим доступа: <https://docs.stripe.com/radar>
57. PayPal. (2025). Orders API v2 – Reference. [Электронный ресурс]. Режим доступа: <https://developer.paypal.com/docs/api/orders/v2/>

58. PayPal. (2025). Integrate Chargeback Protection. [Электронный ресурс]. Режим доступа:  
<https://developer.paypal.com/docs/checkout/advanced/customize/chargeback-protection/integrate/>
59. PCI Security Standards Council. (2022). PCI DSS v4.0 SAQ A – Card-not-present only, e-commerce merchants. [Электронный ресурс]. Режим доступа:  
<https://listings.pcisecuritystandards.org/documents/PCI-DSS-v4-0-SAQ-A.pdf>
60. PCI Security Standards Council. (2022). PCI DSS v4.0 SAQ A-EP – E-commerce merchants affecting payment transaction security. [Электронный ресурс]. Режим доступа: <https://listings.pcisecuritystandards.org/documents/PCI-DSS-v4-0-SAQ-A-EP.pdf>
61. Newman, S. (2021). Building Microservices. Designing Fine-Grained Systems (2nd ed.). O'Reilly Media.
62. Fowler, M., & Lewis, J. (2014). Microservices. [Электронный ресурс]. Режим доступа: <https://martinfowler.com/articles/microservices.html>
63. Fowler, M. (2015). MonolithFirst. [Электронный ресурс]. Режим доступа: <https://martinfowler.com/bliki/MonolithFirst.html>
64. Think with Google. (2019). Milliseconds Make Millions: A Study of Mobile Site Speed and Conversions. [Электронный ресурс]. Режим доступа: <https://www.thinkwithgoogle.com/marketing-strategies/app-and-mobile/milliseconds-make-millions>
65. Baymard Institute. (2025). Cart Abandonment Rate Statistics 2025 – Cart & Checkout. [Электронный ресурс]. Режим доступа: <https://baymard.com/lists/cart-abandonment-rate>
66. International Logistics Group (ILG). (2025). E-Commerce Fashion Returns: How to Tackle the Rising Volume of Online Returns. [Электронный ресурс]. Режим доступа: <https://www.international-logistics-group.com/us-en/insights/e-commerce-fashion-returns-161632/>

- 67.Vercel. (2024). Web Analytics – Privacy-Friendly Real-User Monitoring for Modern Web Apps. [Электронный ресурс]. Режим доступа: <https://vercel.com/analytics>
- 68.OWASP Foundation. (2023). OWASP Web Security Testing Guide v5.0. [Электронный ресурс]. Режим доступа: <https://owasp.org/www-project-web-security-testing-guide/>
- 69.OWASP Foundation. (2021). OWASP Top 10 – 2021: The Ten Most Critical Web Application Security Risks. [Электронный ресурс]. Режим доступа: <https://owasp.org/www-project-top-ten/>
- 70.Microsoft. (2023). Cloud Design Patterns: Queue-Based Load Leveling, Circuit Breaker, Bulkhead. [Электронный ресурс]. Режим доступа: <https://learn.microsoft.com/azure/architecture/patterns>
- 71.Principles of Chaos Engineering. (2020). Chaos Engineering – System Resiliency Through Experimentation. [Электронный ресурс]. Режим доступа: <https://principlesofchaos.org/>
- 72.Cypress.io. (2025). Testing Types – End-to-End vs Component Testing in Cypress. [Электронный ресурс]. Режим доступа: <https://docs.cypress.io/app/core-concepts/testing-types>
- 73.Cypress.io. (2025). Component Testing – Get Started with Cypress Component Testing. [Электронный ресурс]. Режим доступа: <https://docs.cypress.io/app/component-testing/get-started>
- 74.Playwright. (2025). Writing Tests with Playwright Test. [Электронный ресурс]. Режим доступа: <https://playwright.dev/docs/writing-tests>
- 75.Grafana Labs. (2025). Grafana k6 Documentation – Overview. [Электронный ресурс]. Режим доступа: <https://grafana.com/docs/k6/latest/>
- 76.Grafana Labs. (2025). Load Testing Websites with Grafana k6. [Электронный ресурс]. Режим доступа: <https://grafana.com/docs/k6/latest/testing-guides/load-testing-websites/>
- 77.Jest Community. (2024). Getting Started with Jest. [Электронный ресурс]. Режим доступа: <https://jestjs.io/docs/getting-started>

78. Postman. (2024). Introduction to API Testing in Postman. [Электронный ресурс]. Режим доступа: <https://learning.postman.com/docs/designing-and-developing-your-api/testing-an-api>
79. Postman. (2024). Newman Command-line Collection Runner. [Электронный ресурс]. Режим доступа: <https://www.npmjs.com/package/newman>
80. Selenium Project. (2024). Selenium WebDriver – Getting Started. [Электронный ресурс]. Режим доступа: <https://www.selenium.dev/documentation/webdriver>
81. Selenium Project. (2024). Selenium Grid – Scalable Browser Testing Infrastructure. [Электронный ресурс]. Режим доступа: <https://www.selenium.dev/documentation/grid>
82. Baymard Institute. (2023). E-Commerce UX: Main Navigation & Category Design. [Электронный ресурс]. Режим доступа: <https://baymard.com/blog/ecommerce-main-navigation>
83. Baymard Institute. (2023). Product List & Filtering UX: How Users Filter and Sort Products. [Электронный ресурс]. Режим доступа: <https://baymard.com/blog/product-list-filtering>
84. Baymard Institute. (2023). Product Page UX: Essential Elements of a High-Converting PDP. [Электронный ресурс]. Режим доступа: <https://baymard.com/blog/product-page-ux>
85. Google. (2023). Material Design for E-commerce Experiences. [Электронный ресурс]. Режим доступа: <https://m3.material.io>
86. World Wide Web Consortium (W3C). (2018). Web Content Accessibility Guidelines (WCAG) 2.1. [Электронный ресурс]. Режим доступа: <https://www.w3.org/TR/WCAG21>
87. World Wide Web Consortium (W3C). (2023). W3C Accessibility Guidelines (WCAG) 3.0 – Editor’s Draft. [Электронный ресурс]. Режим доступа: <https://www.w3.org/TR/wcag-3.0>
88. Web.dev. (2023). A/B Testing and Core Web Vitals. [Электронный ресурс]. Режим доступа: <https://web.dev/articles/ab-testing-and-core-web-vitals>

89. World Wide Web Consortium (W3C). (2023). Web Authentication: An API for accessing Public Key Credentials Level 3. [Электронный ресурс]. Режим доступа: <https://www.w3.org/TR/webauthn-3/>
90. FIDO Alliance. (2023). What are passkeys? [Электронный ресурс]. Режим доступа: <https://fidoalliance.org/passkeys/>
91. OWASP Foundation. (2024). Session Management Cheat Sheet. [Электронный ресурс]. Режим доступа: [https://cheatsheetseries.owasp.org/cheatsheets/Session\\_Management\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html)
92. Biryukov, A., Dinu, D., & Khovratovich, D. (2021). RFC 9106: Argon2 Memory-Hard Function for Password Hashing and Proof-of-Work Applications. IETF. [Электронный ресурс]. Режим доступа: <https://www.rfc-editor.org/rfc/rfc9106>
93. Google. (2024). Product data specification – Apparel & accessories. Google Merchant Center Help. [Электронный ресурс]. Режим доступа: <https://support.google.com/merchants/answer/7052112>
94. Elastic. (2023). Faceted search and filtering for e-commerce. Elastic Search Solutions. [Электронный ресурс]. Режим доступа: <https://www.elastic.co/blog/faceted-search-ecommerce>
95. Twilio. (2024). A2P 10DLC for US Messaging – Getting Started. [Электронный ресурс]. Режим доступа: <https://www.twilio.com/docs/usage/a2p-10dlc>
96. CTIA. (2023). Messaging Principles and Best Practices. [Электронный ресурс]. Режим доступа: <https://www.ctia.org/the-wireless-industry/industry-commitments/messaging-principles-and-best-practices>
97. Google. (2024). Gmail Bulk Sender Guidelines. [Электронный ресурс]. Режим доступа: <https://support.google.com/a/answer/81126>
98. Yahoo. (2024). Yahoo Mail Sender Best Practices. [Электронный ресурс]. Режим доступа: <https://senders.yahoo.com/senders/>
99. DMARC.org. (2023). Overview of DMARC. [Электронный ресурс]. Режим доступа: <https://dmarc.org/overview/>

100. BIMI Group. (2023). Brand Indicators for Message Identification (BIMI) Implementation Guide. [Электронный ресурс]. Режим доступа: <https://bimigroup.org/>
101. UPS. (2024). UPS Developer API – Tracking. [Электронный ресурс]. Режим доступа: <https://developer.ups.com/api/reference/tracking>
102. FedEx. (2025). FedEx Web Services and API for Shipping and Tracking. [Электронный ресурс]. Режим доступа: <https://www.fedex.com/en-us/developer/web-services.html>

# Презентація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

## ДИПЛОМНА РОБОТА

на ступінь вищої освіти магістр  
із спеціальності 122 Комп'ютерні науки

Модель e-commerce системи для продажу одягу  
з використанням сучасних інформаційних технологій

Виконала: студентка 6 курсу, групи КНДМ-62  
Кузіна Аліна Ярославівна

Керівник: д.т.н., професор, Катков Ю.І.

Київ – 2025

## Загальна характеристика дипломної роботи

- **Мета роботи**  
Підвищити ефективність застосування інтелектуальних нових технологій для створення Fashion E-commerce платформи.
- **Об'єкт дослідження**  
Процес застосування інтелектуальних нових технологій для створення Fashion E-commerce платформи.
- **Предмет дослідження**  
Інтелектуальні нові технології для створення Fashion E-commerce платформи.
- **Наукове завдання**  
Обґрунтувати доцільність застосування інтелектуальних нових технологій для створення Fashion E-commerce платформи.

# Постановка завдання дипломної роботи

## **Проблема:**

Забезпечити гнучкість, масштабованість та економічну ефективність процесів Fashion E-commerce в умовах швидкої зміни трендів та очікувань клієнтів.

## **Завдання роботи:**

1. Провести аналіз ринку та аспектів створення e-commerce платформи для продажу одягу з використанням сучасних технологій.
2. Обґрунтувати доцільність застосування інтелектуальних нових технологій для створення e-commerce платформи для продажу одягу.
3. Розробити практичні рекомендації щодо впровадження інтелектуальних нових технологій для створення e-commerce платформи для продажу одягу.

# Результати аналізу ринку Fashion E-commerce

1. Глобальні тенденції
  - Стале зростання онлайн-продажів у 2024–2025 рр.
  - Посилення персоналізації та аналітики на основі даних.
  - Інтеграція омніканальних моделей і соціальної комерції.
2. Локальні тенденції (Україна, Східна Європа)
  - Переважання мобільних сценаріїв купівлі.
  - Розвинена інфраструктура доставки та повернень.
  - Підтримка локальних брендів і крафтових виробників.
3. Бізнес-моделі
  - B2C, D2C, маркетплейси, підписка, дропшипінг.

# Наслідки застосування сучасних IT у Fashion E-commerce

1. Ключові технології
  - Штучний інтелект та машинне навчання (персоналізація, рекомендації).
  - Big Data (аналітика попиту, управління асортиментом).
  - IoT/RFID (відстеження товарів та логістики).
  - Blockchain і Smart Contracts (прозорість, антифрод).
  - Роботизація та автоматизація складів і логістики.
2. Позитивні ефекти
  - Персоналізований досвід покупця.
  - Віртуальні примірочні та точні розмірні рекомендації.
  - Автоматизація операцій і прискорення виконання замовлень.
3. Виклики
  - Приватність і етика використання даних.
  - Вартість інфраструктури та складність впровадження.

## Технології для розробки e-commerce платформи

1. Стек мов програмування
  - JavaScript/TypeScript – швидкі релізи, низька латентність, єдиний стек.
  - Python – аналітика, ML, персоналізація та протидія шахрайству.
  - PHP – використання наявної інфраструктури та широкої екосистеми інтеграцій.
2. Фронтенд-фреймворки
  - React + Next.js – SEO, стримінг HTML, мобільна продуктивність.
  - Angular – інтегрований фреймворк, signals, корпоративна дисципліна.
  - Vue + Nuxt – швидкі MVP, гнучкі UI-експерименти.

## Хмарна інфраструктура для Fashion E-commerce

1. Хмарні платформи
  - AWS, Google Cloud, Microsoft Azure.
2. Ключові можливості
  - CDN та edge-виконання для «дропів» колекцій.
  - Глобальна маршрутизація запитів і кешування медіаконтенту.
  - Вбудована безпека (WAF, журнали доступу, метрики).
  - Інтеграція AI/ML сервісів з фронтендом (персоналізовані вітрини, рекомендаційні блоки).

## Сховище даних та бази даних

1. Реляційні СУБД
  - PostgreSQL – транзакційні операції, складні транзакції, jsonb + GIN-індекси для пошуку за атрибутами SKU.
  - MySQL – класичні e-commerce сценарії з високою швидкістю читання.
2. Документоорієнтована СУБД
  - MongoDB – каталог з варіативними атрибутами, швидкий рендер карток товарів, гібридне сховище «операційні дані + архіви».
3. Висновок
  - Комбінація СУБД вибирається залежно від вимог до транзакційності, гнучкості схеми та вартості володіння.

## Рекомендації щодо тестування та якості

1. Цілі тестування
  - Забезпечити стабільність під пікові навантаження («дропа» колекцій).
  - Досягти цільових значень Core Web Vitals (INP тощо).
2. Рівні тестування
  - Модульне, інтеграційне, системне, навантажувальне.
3. Інструменти
  - Jest – модульні й компонентні тести.
  - Postman – функціональне та контрактне тестування API.
  - Selenium – наскрізні сценарії в реальних браузерах.
  - k6 – навантажувальне тестування та перевірка SLO.

## Рекомендації щодо безпеки та комунікацій

1. Безпека та приватність
  - Мінімізація персональних даних у комунікаційних системах.
  - Зберігання секретів у менеджері секретів з ротацією та аудитом.
2. Email-комунікації
  - SPF, DKIM, DMARC – обов'язкові політики автентифікації.
  - Поділ транзакційних і маркетингових розсилок.
3. SMS та інші канали
  - Ведення журналів згоди й відписок.
  - Дотримання обмежень частоти та політик операторів.

# Апробація результатів

## В статті:

- Kuzina.A. (2025) Prioritizing disruptive information technologies in Fashion E-Commerce using OSINT / International Science Journal of Engineering & Agriculture, Vol. 4 No. 5 (2025), p. 62-76/ DOI: <https://doi.org/10.46299/j.isjia.20250405> ISSN: 2720-6319, <https://isg-journal.com/isjia/article/view/1102> (date of access: 02.09.2025).

## В тезах:

- Kuzina A.Ya. E-commerce in the Fashion Industry: Digital Transformation, Challenges and Development Prospects (2025) стор 201./ Синергія інновацій: політика, економіка та менеджмент в цифровому світі: збірник матеріалів Міжнародної науково-практичної конференції (м. Київ, 18 квітня 2025 р.) [Електронне видання]. Київ : ДУІКТ, 2025. 364 с. [https://duikt.edu.ua/uploads/p\\_2779\\_26171712.pdf](https://duikt.edu.ua/uploads/p_2779_26171712.pdf)
- Кузіна А.Я. ІНТЕГРАЦІЯ ІОТ-ТЕХНОЛОГІЙ В ІНТЕРНЕТ-МАГАЗИНИ ОДЯГУ ДЛЯ ПОКРАЩЕННЯ КОРИСТУВАЦЬКОГО ДОСВІДУ// VI Науково-технічна конференція «Сучасний стан та перспективи розвитку ІоТ». Збірник тез 15 квітня 2025. – К.: ДУІКТ, ст. 60-63. [https://duikt.edu.ua/uploads/p\\_2779\\_40288420.pdf](https://duikt.edu.ua/uploads/p_2779_40288420.pdf)