

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Платформа продажу одягу з персоналізованою
системою рекомендацій»**

Виконав студент групи ЗПЗ-21-1
КОНВІСАР Андрій Сергійович

Керівник роботи:
НЕРОДА Сергій Іванович

Рецензент:
ЧИЖМОТРЯ Олексій Володимирович

Житомир – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення

Тетяна Вакалюк

«26» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **КОНВІСАР Андрій Сергійович**

Керівник роботи: **НЕРОДА Сергій Іванович**

Тема роботи: **«Платформа продажу одягу з персоналізованою системою рекомендацій»,**

затверджена Наказом закладу вищої освіти від 14.02.2025р., №62/с

Вихідні дані для роботи: Об'єктом дослідження є інформаційна система електронної комерції, що забезпечує функціонування інтернет-магазину одягу. Предметом дослідження є архітектура, функціональність і програмна реалізація інтернет-магазину, включаючи базу даних товарів, модулі управління кошиком, обліку користувачів, оформлення замовлень та впровадження персоналізованих рекомендацій на основі embedding-технологій і аналізу поведінки користувачів.

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Сергій НЕРОДА	27.02.2025р.	27.02.2025р.
2	Сергій НЕРОДА	27.02.2025р.	27.02.2025р.
3	Сергій НЕРОДА	27.02.2025р.	27.02.2025р.

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з вебдодатку платформи та пояснювальної записки. Пояснювальна записка до випускної роботи містить 69 сторінок, 27 ілюстрацій, 9 таблиць, 2 додатки та 35 джерел використаної літератури.

Метою випускної кваліфікаційної роботи бакалавра є розробка функціональної вебплатформи для онлайн-продажу одягу з інтеграцією персоналізованої системи рекомендацій, що базується на технологіях векторного представлення даних (ембедінгів).

Архітектура інтернет-магазину реалізована з використанням фреймворку Next.js, стилізації SCSS та бази даних MongoDB. Було створено модулі для каталогу товарів, кошика, особистого кабінету користувача, списку бажань та порівнянь товарів, та сторінки оформлення замовлення. Платіжна система інтегрована через сервіс Stripe.

Інтерфейс адаптовано для всіх типів пристроїв – мобільних, планшетів і десктопів. Проведено тестування продуктивності інтерфейсу та швидкості завантаження за допомогою сервісу GTMetrix.

КЛЮЧОВІ СЛОВА: ІНТЕРНЕТ-МАГАЗИН, РЕКОМЕНДАЦІЙНА СИСТЕМА, EMBEDDING, КОШИК, ЗАМОВЛЕННЯ, КОРИСТУВАЧ.

Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Конвісар А.С.			Платформа продажу одягу з персоналізованою системою рекомендацій	Літ.	Арк.
Перевір.		Нерода С.І.					
Реценз.						Житомирська політехніка, група ЗІПЗ-21-1	
Н. Контр.		Тичина О.П.					
Затверд.		Вакалюк Т.А.					

ABSTRACT

The bachelor's qualification thesis consists of a web application for an online store and an explanatory report. The explanatory report comprises 69 pages, 27 illustrations, 9 tables, 2 appendices, and references to 35 sources of literature.

The aim of the bachelor's qualification thesis is to develop a functional web platform for online clothing sales with the integration of a personalized recommendation system based on data vector representation technologies (embeddings).

The architecture of the online store is implemented using the Next.js framework, SCSS styling, and the MongoDB database. Modules were developed for the product catalog, shopping cart, user account, wishlist and product comparison, as well as the checkout page. The payment system is integrated via the Stripe service.

The interface is responsive and optimized for all types of devices – mobile, tablet, and desktop. Performance and page load speed were tested using the GTMetrix service.

KEYWORDS: ONLINE STORE, RECOMMENDATION SYSTEM, EMBEDDING, CART, ORDER, USER.

		<i>Конвісар А.С.</i>			ІПЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		<i>Нерода С.І.</i>				
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дат</i>		4

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ПЛАТФОРМИ ПРОДАЖУ ОДЯГУ	9
1.1. Постановка задачі	9
1.2. Аналіз існуючих аналогів програмного продукту: функціонал, переваги та недоліки	11
1.3. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	18
Висновки до першого розділу	22
РОЗДІЛ 2. ПРОЄКТУВАННЯ ПЛАТФОРМИ ПРОДАЖУ ОДЯГУ	23
2.1. Визначення функціональних і нефункціональних вимог до платформи	23
2.2 Розробка бази даних платформи	26
2.3 Проєктування та реалізація алгоритмів роботи системи	32
2.4 Реалізація функціоналу платформи	34
Висновки до другого розділу	43
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ПЛАТФОРМОЮ ..	44
ПРОДАЖУ ОДЯГУ	44
3.1 Користувацький інтерфейс і адаптивність верстки	44
3.2 Реалізація особистого кабінету, реєстрації та авторизації	51
3.3 Оформлення замовлення: кошик, оплата	54
3.4 Тестування платформи	58
Висновки до третього розділу	64
ВИСНОВКИ	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	66
ДОДАТКИ	69

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				5
Змн.	Арк.	№ докум.	Підпис	Дат		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API (Application Programming Interface) – інтерфейс прикладного програмування, який дозволяє обмінюватися даними між різними системами.

UI (User Interface) – інтерфейс користувача.

UX (User Experience) – користувацький досвід при взаємодії з інтерфейсом.

CRUD (Create, Read, Update, Delete) – основні операції з базами даних.

DB (Database) – база даних.

MongoDB – документно-орієнтована NoSQL база даних.

JSON (JavaScript Object Notation) – текстовий формат обміну даними між клієнтом і сервером.

SEO (Search Engine Optimization) – оптимізація вебсторінок для покращення їх видимості в пошукових системах.

REST API – стиль архітектури API, що ґрунтується на використанні HTTP-запитів.

JWT (JSON Web Token) – формат токенів для безпечної передачі інформації та автентифікації.

UX/UI – поєднання дизайну інтерфейсу користувача та досвіду користування.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		6

ВСТУП

Актуальність теми. В умовах стрімкого розвитку електронної комерції та високої конкуренції на ринку онлайн-продажу одягу особливої значущості набуває впровадження персоналізованих рекомендацій. Покупці очікують не лише швидкого доступу до товарів, а й індивідуального підходу до формування пропозицій відповідно до їхніх інтересів, поведінки та попередніх дій на платформі.

Рекомендаційні системи дозволяють значно підвищити конверсію, середній чек і рівень задоволеності користувача. У сфері онлайн-продажу одягу це набуває особливого значення, оскільки асортимент постійно оновлюється, а візуальні та стилістичні вподобання клієнтів мають динамічний характер. Сучасні технології, зокрема використання векторних ембедінгів і моделей штучного інтелекту, відкривають нові можливості для побудови інтелектуальних систем рекомендацій, які враховують не лише схожість назв чи категорій, а й глибший смисловий контекст товарів. Це дає змогу платформі виділятися серед конкурентів, формуючи у користувача відчуття індивідуального підходу й збільшуючи ймовірність повторних покупок.

Отже, розробка платформи з підтримкою персоналізованих рекомендацій є надзвичайно актуальною як з точки зору бізнес-ефективності, так і з позиції сучасного користувацького досвіду, що робить цю тему важливою як для практики електронної торгівлі, так і для подальших наукових досліджень у сфері інформаційних технологій.

Метою випускної кваліфікаційної роботи бакалавра є розробка функціональної вебплатформи для онлайн-продажу одягу з інтеграцією персоналізованої системи рекомендацій, що базується на технологіях векторного представлення даних (ембедінгів).

Для реалізації поставленої мети потрібно виконати наступні завдання:

1. Проаналізувати сучасні інтернет-магазини одягу, дослідити функціональні вимоги та технічні рішення, які використовуються у сучасних платформах;

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				7
Змн.	Арк.	№ докум.	Підпис	Дат		

2. Сформувати функціональні та нефункціональні вимоги до платформи та обґрунтувати вибір технологічного стеку, з урахуванням потреб у підтримці динамічних даних, ефективного рендерингу, управління станом платформи;

3. Розробити платформу інтернет-магазину.

Об'єктом дослідження є інформаційна система електронної комерції, що забезпечує функціонування інтернет-магазину одягу.

Предметом дослідження є архітектура, функціональність і програмна реалізація інтернет-магазину, включаючи базу даних товарів, модулі управління кошиком, обліку користувачів, оформлення замовлень та впровадження персоналізованих рекомендацій на основі embedding-технологій і аналізу поведінки користувачів.

Дослідження також охоплює аналіз технологій веброзробки, зокрема використання Next.js, React, MongoDB, Stripe, з метою забезпечення зручності користування, швидкодії, адаптивності інтерфейсу та безпеки обробки персональних і платіжних даних.

Окрему увагу в роботі приділено реалізації власної системи персоналізованих рекомендацій, яка побудована на основі embedding-технологій та аналізу поведінкових даних користувачів. За допомогою векторного представлення товарів (ембедінгів), що генеруються з урахуванням описів, назв, категорій і дій користувача, система формує динамічні й змістовно релевантні пропозиції, які значно покращують користувацький досвід і підвищують конверсію.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		8

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ПЛАТФОРМИ ПРОДАЖУ ОДЯГУ

1.1. Постановка задачі

Розробка платформи продажу одягу потребує ретельного визначення технічних і функціональних вимог, а також вибору сучасного інструментарію, який забезпечить високу продуктивність, масштабованість і персоналізовану взаємодію з користувачем. У межах цього дипломного проєкту реалізується сучасна Full-Stack веб-платформа для продажу одягу з інтегрованою системою рекомендацій на основі векторних ембедінгів.

1. Аналіз сучасних технологій та підходів:

- вивчення архітектури Full-Stack додатків із використанням Next.js, зокрема можливостей SSR (Server-Side Rendering) та API Routes;– використання React.js як основи клієнтської частини;
- застосування Redux Toolkit для централізованого керування станом додатку;
- аналіз підходів до інтеграції платіжних систем (Stripe) та зовнішніх API;
- дослідження можливостей персоналізованих систем рекомендацій на основі семантичних ембедінгів.

2. Формування функціональних та інформаційних вимог:

- реалізація головної сторінки з промо-контентом і навігацією;
- створення каталогу товарів з фільтрами за категоріями, брендами, кольорами, розмірами;
- детальна картка товару з описом, характеристиками, цінами, фотографіями та варіаціями;
- підтримка кошика, оформлення замовлень, перегляд історії покупок;
- створення профілю користувача, авторизація, реєстрація та редагування особистих даних;
- підключення онлайн-оплати через Stripe API;

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				9
Змн.	Арк.	№ докум.	Підпис	Дат		

– реалізація системи рекомендацій товарів із використанням векторних ембедінгів (Google Generative AI, MongoDB Vector Search).

3. Проєктування інтерфейсу користувача:

- адаптивна верстка з використанням SCSS modules для коректного відображення на різних пристроях;
- створення компонування сторінок за принципами UI/UX-дизайну;
- впровадження клієнтської логіки з використанням Next.js, React та Redux.

4. Побудова структури бази даних (MongoDB):

- створення колекцій: users, products, orders, payments, categories, embeddings;
- реалізація зв'язків між користувачами, товарами та замовленнями;
- налаштування векторного індексу в MongoDB Atlas для швидкого пошуку подібних товарів за ембедінгами.

5. Серверна логіка та API:

- реалізація backend'у на Node.js + Express через API маршрути;
- обробка запитів користувачів, генерація рекомендацій, оновлення товарів;
- валідація запитів, захист від несанкціонованого доступу;
- створення окремого маршруту для отримання рекомендованих товарів на основі ембедінгів;
- інтеграція з платіжною системою Stripe.

6. Тестування та перевірка працездатності

- функціональне тестування усіх ключових сценаріїв (додавання до кошика, оформлення замовлення, отримання рекомендацій);
- перевірка коректного виводу UI-компонентів та адаптивності дизайну;
- симуляція реальних кейсів покупки та тестування інтеграцій з Stripe і MongoDB.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		10

Результатом виконання плану дій є створення платформи, яка забезпечить ефективну взаємодію з користувачем, систему замовленнями товарів, а також інтеграцію з сучасними платіжними сервісами. Система рекомендацій на основі семантичних ембедінгів покликана значно підвищити рівень залученості користувача та конверсію платформи.

1.2. Аналіз існуючих аналогів програмного продукту: функціонал, переваги та недоліки

У процесі розробки сучасної веб-платформи для продажу одягу надзвичайно важливим етапом є аналіз існуючих рішень на ринку. Дослідження популярних аналогів дозволяє не лише визначити поточні тенденції у функціональності, дизайні та технологічному стеку, а й виявити недоліки, які можна уникнути під час реалізації власного продукту. Такий порівняльний аналіз допомагає сформулювати обґрунтовані вимоги до архітектури системи, користувацького досвіду (UX) та інструментів персоналізації.

Для аналізу сучасних аналогів програмного продукту було обрано низку відомих онлайн-магазинів, які спеціалізуються на продажу одягу та аксесуарів і мають масштабовану інфраструктуру, розвинену систему персоналізації й широкий асортимент продукції. До переліку включено як міжнародні, так і локальні рішення, а саме: ASOS, Zalando, H&M, Modnakasta. Вибір саме цих платформ обумовлений їхньою популярністю серед користувачів, сучасною реалізацією клієнтського інтерфейсу, підтримкою мобільних пристроїв та інтеграцією онлайн-оплати.

ASOS (As Seen On Screen) – це один із провідних світових онлайн-магазинів одягу, взуття та аксесуарів, орієнтований переважно на молодіжну аудиторію. Платформа забезпечує багатофункціональний інтерфейс, просту навігацію та зручну взаємодію з користувачем. На головній сторінці (рис. 1.1) представлено основні напрямки: “Shop Women” і “Shop Men”, а також промо-блоки зі знижками, мобільним застосунком, доставкою та поверненнями [18].

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		11

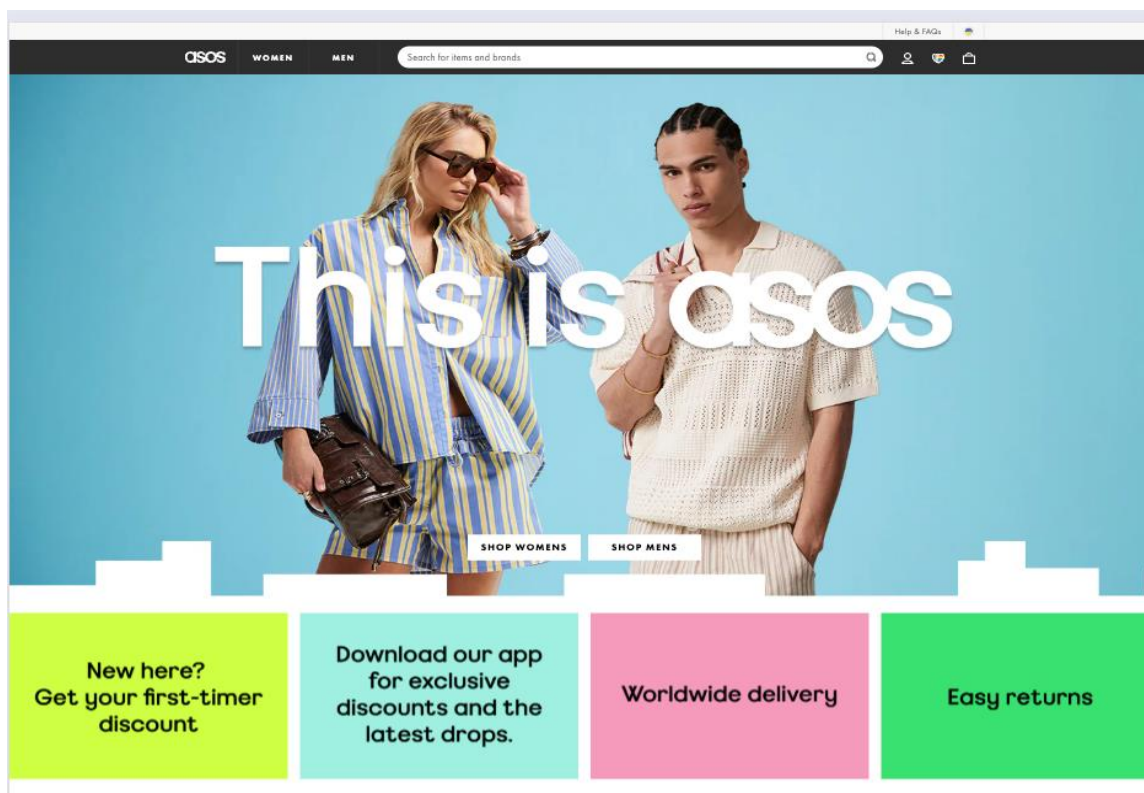


Рис. 1.1 – Головна сторінка платформи ASOS [18]

ASOS – це сучасна платформа електронної комерції, яка має низку важливих переваг. Її каталог містить розширені фільтри, що дозволяють користувачам шукати товари за категоріями, брендами, розмірами, стилем, а також навіть за етичними критеріями, як-от «сталій одяг». Сайт пропонує персоналізовані рекомендації, що формуються на основі історії переглядів, покупок та поведінки користувача. Інтерфейс повністю адаптований до різних пристроїв, забезпечуючи комфортний перегляд як на комп'ютерах, так і на планшетах або телефонах. Платформа підтримує онлайн-оплату та доставку в понад 190 країн світу. Крім того, вона пропонує зручну систему повернення товарів, що сприяє зростанню довіри з боку клієнтів.

Kasta.ua – один із найбільших онлайн-ритейлерів в Україні, що спеціалізується на продажу одягу, взуття, косметики, товарів для дому та електроніки. Платформа має чітко локалізований інтерфейс, підтримку української мови, національної валюти та широкий вибір товарів як від

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		12

міжнародних, так і вітчизняних брендів. Саме ці характеристики роблять Kasta зручним аналогом для порівняння у межах українського ринку (рис. 1.2) [24].

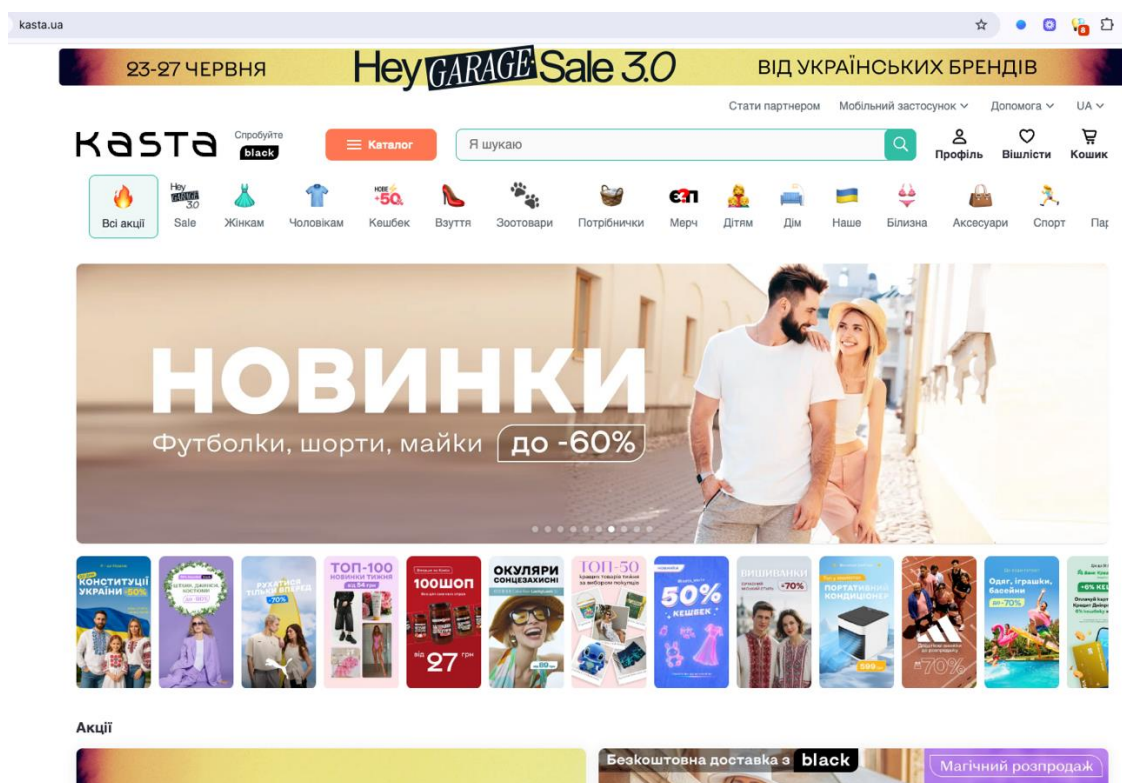


Рис. 1.2 – Головна сторінка української платформи Kasta.ua [24]

Платформа Kasta вирізняється зручністю використання та адаптацією під український ринок. Вона повністю підтримує українську мову, а основною валютою розрахунку є гривня. Для зручного пошуку товарів реалізовано розгалужену систему категорій та фільтрів: користувачі можуть швидко знаходити потрібне, відсортовуючи за брендом, розміром, кольором, ціною чи рейтингом. Персоналізовані рекомендації формуються на основі попередніх переглядів, вподобаних категорій і збережених товарів.

На головній сторінці часто розміщуються промо-слайдери та банери, які інформують про актуальні акції, знижки та розпродажі. Для мобільних користувачів передбачено зручний застосунок для iOS і Android, який підтримує push-сповіщення, завдяки чому можна миттєво отримувати інформацію про новинки чи вигідні пропозиції. Крім того, Kasta має власну логістичну службу – KastaPost, що забезпечує оперативну та доступну доставку замовлень по всій Україні.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				13
Змн.	Арк.	№ докум.	Підпис	Дат		

Після розгляду української мультибрендової платформи Kasta доцільно проаналізувати приклад європейського fashion-бренду, який активно працює на ринку України та підтримує онлайн-продаж – платформу Sinsay. Це приклад монобрендового магазину з міжнародною присутністю, який поєднує стильний дизайн, адаптивність інтерфейсу та чітку сегментацію за аудиторіями.

Sinsay (рис. 1.3) – це бренд компанії LPP S.A. (Польща), до якої також належать Reserved, Cropp і House. Онлайн-магазин Sinsay має офіційну версію для України, що повністю адаптована до локальних реалій: інтерфейс українською мовою, ціни в гривнях, доставка на території України та підтримка популярних платіжних методів [31].

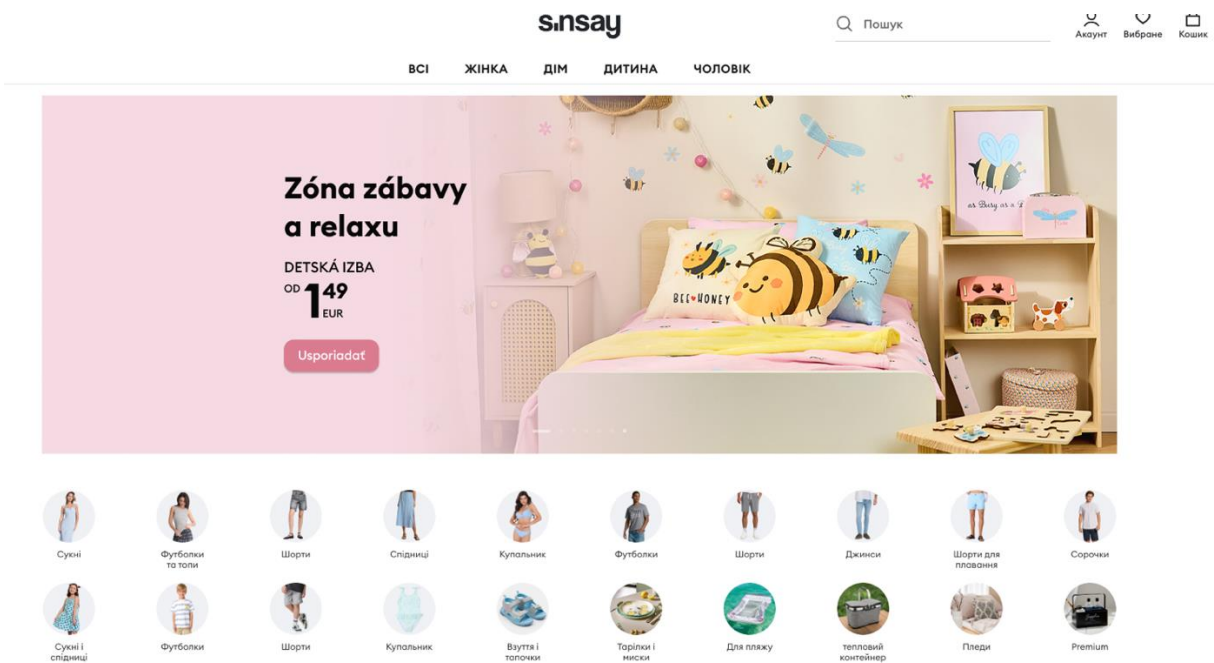


Рис. 1.3 – Головна сторінка української версії платформи Sinsay [31]

Платформа Sinsay вирізняється простим і мінімалістичним дизайном, який забезпечує зручність користування як на мобільних пристроях, так і на десктопах. Асортимент чітко розподілений за цільовими аудиторіями: окремі розділи призначені для жінок, чоловіків, дітей, а також є категорії з аксесуарами та товарами для дому.

У каталозі реалізовано базові, але функціональні фільтри — зокрема за типом товару, ціною та кольором. Головна сторінка регулярно оновлюється

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				14
Змн.	Арк.	№ докум.	Підпис	Дат		

акційними банерами та промо-блоками, які інформують про знижки, наприклад “-30% на другу одиницю”.

Для українських користувачів доступна локалізована доставка з можливістю обрати оплату онлайн або при отриманні товару. Крім того, Sinsay пропонує мобільний застосунок, який надає персоналізовані рекомендації, базуючись на історії переглядів користувача.

MODIVO – це міжнародний онлайн-магазин одягу, взуття та аксесуарів, який належить до польської групи eobuwie.pl S.A. (рис. 1.4). Платформа зосереджена на стильних брендах середнього та преміум-сегменту, має українську версію сайту, підтримує доставку по Україні, а також забезпечує персоналізований досвід покупок завдяки сучасним цифровим рішенням [27].

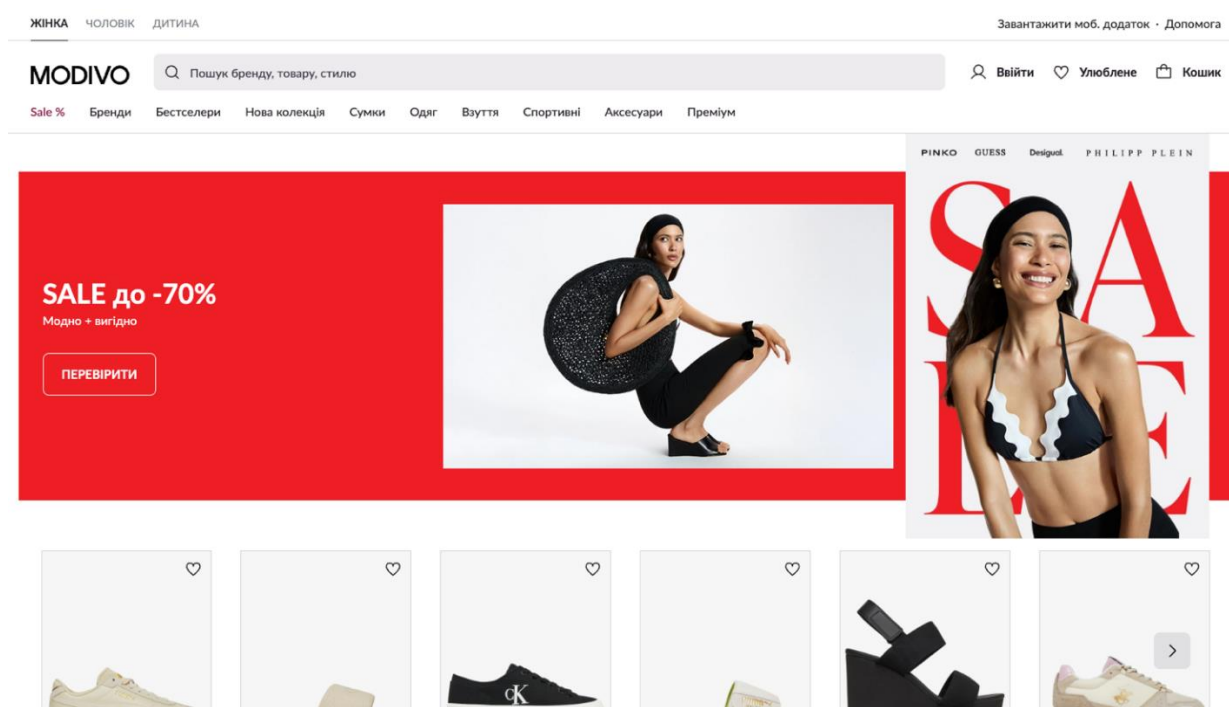


Рис. 1.4 – Головна сторінка української версії платформи MODIVO [27]

Платформа MODIVO пропонує зручний і добре структурований інтерфейс, де асортимент чітко розділено за категоріями: окремі секції для жінок, чоловіків, дітей, а також за брендами, розмірами, стилем, матеріалом і кольором. Для зручності користувача реалізовано систему фільтрів і тегів, яка дозволяє швидко звужити пошук і знайти потрібний товар.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				15
Змн.	Арк.	№ докум.	Підпис	Дат		

Дизайн сайту мінімалістичний і повністю адаптивний, що забезпечує комфортне користування з будь-якого пристрою. Система персоналізованих рекомендацій враховує попередні перегляди, обраний стиль і історію покупок, що дає змогу пропонувати користувачу найбільш релевантні товари.

MODIVO також має мобільний застосунок для iOS та Android, який підтримує push-сповіщення про акції, новинки та статус замовлень. Оплата здійснюється в національній валюті – гривні, а доставка покриває всю територію України. Окрім цього, передбачена зручна система повернення товарів, що підвищує лояльність користувачів до платформи.

Для кожної платформи здійснено оцінку переваг, недоліків та особливостей реалізації ключових функцій. З метою наочності проведено детальне порівняння функціональних характеристик платформ, яке представлено у вигляді таблиці в Додатку А.

Аналіз чотирьох платформ – ASOS, Kasta.ua, Sinsay, MODIVO – дозволив зробити низку ключових висновків. Попри високий рівень візуального оформлення та UX на міжнародних ресурсах (ASOS, MODIVO), лише Kasta.ua та Sinsay демонструють повноцінну локалізацію для українського користувача, включаючи інтерфейс українською мовою, ціни в гривнях і доставку по території України.

Серед усіх платформ лише ASOS реалізував просунуту систему рекомендацій, яка базується на поведінковій аналітиці, тоді як інші мають обмежені або базові алгоритми персоналізації. У свою чергу, Kasta.ua та MODIVO вирізняються розширеною фільтрацією, наявністю мобільних застосунків і акційними інструментами залучення користувачів.

Результати аналізу підкреслюють потребу в локалізованій, технологічно просунутій платформі, яка б поєднувала інтерфейсну зручність, адаптацію до українського ринку та сучасні засоби персоналізованих рекомендацій. Отримані висновки лягли в основу постановки вимог до власної розробки веб-платформи, орієнтованої на масового українського користувача.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				16
Змн.	Арк.	№ докум.	Підпис	Дат		

З метою виявлення конкурентних переваг та слабких сторін провідних гравців ринку онлайн-торгівлі одягом було проаналізовано чотири сучасні платформи: ASOS, Kasta.ua, Sinsay та MODIVO. Аналіз охоплює як міжнародні, так і локалізовані рішення, що дозволяє комплексно оцінити їх придатність до впровадження в українських умовах або подальшого наслідування в рамках розробки власного програмного продукту.

Табл. 1.1 відображає ключові переваги та недоліки кожної з платформ, орієнтуючись на такі аспекти, як персоналізація, підтримка локалізації, інтерфейсні рішення, функціональність мобільних застосунків та рівень адаптивності.

Таблиця 1.1

Порівняльна характеристика популярних платформ онлайн-торгівлі одягом

Платформа	Переваги	Недоліки
ASOS	Сильна персоналізація, інтернаціональний бренд, інтуїтивний UX	Обмеження по регіонах, інтерфейс перевантажений акціями
Kasta.ua	Повна локалізація, українська мова та валюта, власна доставка	Перевантаження рекламним контентом, базова персоналізація
Sinsay	Локалізований інтерфейс, сегментація аудиторій, стильний дизайн	Обмежені фільтри, проста система рекомендацій
MODIVO	Якісний UI, брендовий асортимент, доставка по Україні, адаптивність	Ціновий сегмент вище середнього, обмежений доступ без авторизації

На основі проведеного порівняння можна зробити висновок, що жодна з платформ не поєднує водночас повну локалізацію, високу якість персоналізації та гнучку архітектуру рекомендаційних механізмів. Найвищий рівень технологічного розвитку демонструє ASOS, однак платформа не орієнтована на український ринок. Kasta.ua та Sinsay, навпаки, повністю локалізовані, але обмежені за функціональністю інтелектуальних систем рекомендацій. MODIVO вирізняється балансом між якістю інтерфейсу та гнучкістю фільтрації, проте залишається орієнтованим переважно на брендовий сегмент.

У результаті проведеного аналізу існуючих платформ онлайн-торгівлі одягом – ASOS, Kasta.ua, Sinsay та MODIVO – було встановлено, що кожна з них має як унікальні сильні сторони, так і певні обмеження з точки зору

функціональності, локалізації та технологічного рівня. Міжнародні сервіси, такі як ASOS, пропонують розвинуті системи персоналізації та високий рівень UX/UI-дизайну, однак не адаптовані до українського ринку. Натомість локальні платформи (Kasta.ua, Sinsay, MODIVO) демонструють повну мовну та валютну підтримку, однак мають спрощені рекомендаційні механізми або обмежену гнучкість фільтрації.

Таке порівняння виявило незаповнену нішу на ринку – відсутність платформи, яка б поєднувала локалізовану архітектуру, глибоку персоналізацію, адаптивний дизайн і гнучкий пошук за допомогою сучасних AI-алгоритмів. Саме ці висновки стали основою для формування вимог до подальшої розробки власного веб-додатку, який має на меті забезпечити комфортну, персоналізовану та технічно сучасну взаємодію з українським споживачем.

1.3. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Розробка повнофункціональної веб-платформи для продажу одягу з інтегрованою системою персоналізованих рекомендацій потребує ретельного підходу до вибору інструментальних засобів, які забезпечать високу продуктивність, гнучкість та масштабованість системи. Обрані технології повинні підтримувати розподілену архітектуру, інтеграцію з зовнішніми API-сервісами (зокрема, платіжними системами та сервісами штучного інтелекту), мати активну спільноту, розвинену документацію та відповідати сучасним стандартам безпеки.

Крім того, не менш важливою є оцінка вимог до апаратного забезпечення як на етапі розробки, так і під час розгортання платформи. Це дозволяє забезпечити стабільну роботу як клієнтської, так і серверної частини додатку при зростанні навантаження та кількості користувачів [29].

Для реалізації клієнтської частини веб-додатку обрано зв'язку Next.js [10] + React.js, що є на сьогодні однією з найбільш популярних та технологічно просунутих комбінацій у середовищі JavaScript-розробки [8]. Next.js забезпечує

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		18

підтримку серверного рендерингу (SSR), що критично важливо для підвищення швидкості завантаження сторінок, SEO-оптимізації та покращення індексації у пошукових системах. Також цей фреймворк дозволяє реалізовувати динамічні маршрути, статичну генерацію сторінок (SSG) та обробку API-запитів безпосередньо на рівні фронтенду, що спрощує архітектуру.

Базовим фреймворком для створення інтерфейсних компонентів є React.js, який підтримує декларативний підхід, віртуальний DOM, повторне використання компонентів та інтеграцію з екосистемою Redux.

Управління глобальним станом застосунку здійснюється за допомогою Redux Toolkit [17], що є рекомендованим інструментом від розробників Redux. Його використання дозволяє мінімізувати boilerplate-код, централізувати логіку обробки асинхронних дій (через createAsyncThunk) та ефективно управляти потоками даних у масштабованих застосунках.

У якості серверного середовища обрано Node.js [15], що дозволяє використовувати JavaScript на стороні сервера та забезпечує асинхронну, подієво-орієнтовану модель обробки запитів. На базі Node.js реалізовано фреймворк Express.js, який відповідає за маршрутизацію, обробку HTTP-запитів, керування middleware-функціями та підключення зовнішніх сервісів.

Особливу увагу приділено реалізації механізмів аутентифікації та авторизації. Для цього застосовується JWT (JSON Web Token) – сучасний спосіб захисту даних у форматі шифрованих токенів.

Для забезпечення електронної комерції на платформі інтегровано Stripe API, який надає надійний, сертифікований інтерфейс для здійснення онлайн-оплат, обробки транзакцій та виставлення рахунків. Stripe підтримує вебхуки, які дозволяють в режимі реального часу отримувати повідомлення про статус платежів.

Основною системою управління базами даних виступає MongoDB [9], обрана за її масштабованість, гнучку документно-орієнтовану модель та повну сумісність з JavaScript-екосистемою. Зберігання об'єктів у форматі BSON дозволяє зручно працювати з вкладеними структурами, а підтримка індексування – забезпечити швидкий доступ до необхідних записів.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				19
Змн.	Арк.	№ докум.	Підпис	Дат		

Хмарна інфраструктура MongoDB Atlas забезпечує резервне копіювання, автоматичне масштабування, високий рівень безпеки та можливість розгортання у географічно розподілених регіонах.

У межах реалізації системи рекомендацій використовується MongoDB Vector Search, який дозволяє створювати векторні індекси та здійснювати пошук схожих товарів за семантичною близькістю. Це особливо важливо для побудови персоналізованого досвіду на основі ембедінгів.

Для реалізації персоналізованої системи рекомендацій у даному проєкті використано підхід векторних ембедінгів. Цей підхід дає змогу перетворювати текстові описи товарів, їхні назви й категорії на багатовимірні числові вектори, які потім використовуються для обчислення подібності між товарами. Зокрема, для генерації таких векторів застосовуються зовнішні сервіси, як-от Google Generative AI Embedding API, що дозволяє створювати вектори на основі текстових характеристик, або Hugging Face Transformers API, який підтримує безліч попередньо натренованих моделей, зокрема sentence-transformers/all-MiniLM-L6-v2.

Згенеровані векторні представлення (ембедінги) зберігаються в базі даних MongoDB. Надалі вони використовуються для пошуку схожих товарів за допомогою алгоритму пошуку найближчих сусідів (nearest neighbor search), що забезпечує ефективну персоналізацію рекомендацій.

Для супроводу, тестування та перевірки працездатності проєкту активно використовувалися сучасні інструменти розробника. Наприклад, Postman дозволив вручну перевіряти API-запити, тестувати авторизацію, систему онлайн-платежів і отримання рекомендацій. Система контролю версій Git у поєднанні з GitHub забезпечувала керування кодовою базою, створення гілок і pull request'ів. Для зручної взаємодії з користувачем на клієнтському інтерфейсі використовувалася бібліотека react-toastify, яка відповідала за вивід інтерактивних повідомлень (наприклад, підтвердження додавання товару до кошика або повідомлення про помилки).

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		20

Успішна реалізація та подальше функціонування веб-платформи для продажу одягу вимагає відповідного апаратного забезпечення як на етапі розробки, так і під час експлуатації у хмарному середовищі. Вибір технічних характеристик повинен забезпечувати стабільність, надійність, швидку обробку запитів, а також безперебійну взаємодію з базами даних та зовнішніми API.

На етапі розробки проєкту оптимальним вважається використання персонального комп'ютера або ноутбука із процесором не нижче Intel Core i5 або AMD Ryzen 5, оперативною пам'яттю від 8 ГБ (рекомендовано 16 ГБ) та SSD-накопичувачем обсягом від 256 ГБ, що забезпечує швидке завантаження середовищ розробки, симуляцію серверного оточення та ефективну роботу з інструментами типу Docker, Postman, Git.

Для розгортання застосунку в реальному середовищі достатньо використання хмарних рішень, таких як Vercel, Render або Railway, які підтримують Node.js-сервер [20] та надають можливості горизонтального масштабування. У випадку розміщення на VPS або хмарному сервері, рекомендовані технічні параметри включають: 2 ГБ оперативної пам'яті, 1–2 ядра CPU, доступ до MongoDB Atlas, а також стабільне інтернет-з'єднання з високою пропускнуою здатністю [11].

Таким чином, вибір інструментальних засобів та визначення апаратних вимог для розробки платформи продажу одягу є критично важливим етапом, що безпосередньо впливає на її функціональність, масштабованість та користувацький досвід. Обраний стек технологій – Next.js, React, Redux Toolkit, Node.js, Express, MongoDB [28], Stripe API та модулі рекомендацій на основі векторних ембедінгів – забезпечує повноцінну реалізацію сучасної, адаптивної, персоналізованої системи з високою продуктивністю. Апаратні вимоги до середовища розробки й хостингу є помірними, що дозволяє ефективно впроваджувати платформу навіть у невеликих командах із мінімальними ресурсами.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				21
Змн.	Арк.	№ докум.	Підпис	Дат		

Висновки до першого розділу.

У першому розділі було проведено комплексний аналіз предметної області, в межах якого визначено основні функціональні, інформаційні та технологічні вимоги до сучасної веб-платформи з продажу одягу. Сформульовано цілісне бачення майбутнього продукту як повнофункціональної системи електронної комерції з підтримкою персоналізованих рекомендацій на основі векторних ембедінгів.

У процесі дослідження було здійснено порівняльний аналіз провідних аналогів (ASOS, Kasta.ua, Sinsay, MODIVO), що дозволило виявити сильні та слабкі сторони конкурентних рішень, окреслити типові моделі архітектури, оцінити рівень локалізації, інтерфейсних рішень та механізмів персоналізації. На основі цього сформульовано вимоги до функціональності, структури даних та UX-елементів цільової системи.

Також обґрунтовано вибір інструментального стеку, до якого увійшли Next.js, React, Redux Toolkit, Node.js, Express.js, MongoDB Atlas, Stripe API та сервіси генерації ембедінгів. З урахуванням обраних технологій наведено раціональні вимоги до апаратного забезпечення, необхідного для реалізації, тестування та розгортання платформи.

У підсумку, результати розділу створили необхідну теоретичну та методологічну базу для подальшого проєктування, моделювання та реалізації системи, що відповідає актуальним вимогам ринку онлайн-торгівлі та очікуванням сучасного користувача.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		22

РОЗДІЛ 2. ПРОЄКТУВАННЯ ПЛАТФОРМИ ПРОДАЖУ ОДЯГУ

2.1. Визначення функціональних і нефункціональних вимог до платформи

Процес створення програмного забезпечення, зокрема веб-платформи для електронної комерції, вимагає чіткого визначення вимог, що покладені в основу проєктування архітектури, реалізації логіки взаємодії з користувачем та забезпечення стабільної роботи системи. У цьому контексті особливу роль відіграє розмежування функціональних та нефункціональних вимог, які задають відповідно що саме повинна робити система і як саме вона має це реалізовувати.

Для забезпечення ефективного функціонування вебплатформи FitIQ були сформовані функціональні вимоги, які охоплюють усі ключові аспекти взаємодії користувача з системою. Основна мета – надати користувачу зручні інструменти для пошуку, перегляду, оцінки та придбання одягу, а також забезпечити індивідуальний досвід користування за рахунок фільтрації, збереження вподобаного, підписки на новинки тощо (табл.2.1).

Таблиця 2.1

Основні функціональні вимоги до вебплатформи FitIQ

Функціональна підсистема	Опис функціональності
Реєстрація та аутентифікація	Можливість створення облікового запису, входу та виходу зі збереженням авторизації (JWT)
Перегляд каталогу товарів	Вивід списку товарів на головній сторінці та в категоріях
Фільтрація за категоріями, кольорами	Механізми фільтрації на сторінці магазину для зручного пошуку
Детальна картка товару	Вивід повної інформації про товар: опис, ціна, розміри, фото
Додавання до кошика, редагування, оформлення замовлення	Збереження товарів у кошику, можливість змінювати кількість, оформлювати замовлення
Перегляд історії замовлень	Доступ до історії покупок зареєстрованого користувача
Редагування профілю користувача	Редагування імені, віку, статі, email, паролю в профілі
Онлайн-оплата через Stripe	Обробка платіжних транзакцій через Stripe API з підтвердженням

Персоналізовані рекомендації (на основі ембедінгів)	Вивід товарів, схожих за змістом або історією дій користувача
Пошук за назвою товару	Можливість пошуку товару за ключовим словом або назвою
Вивід новин моди	Відображення модних новин на окремій сторінці платформи
Підписка на оновлення колекцій через email	Форма підписки для отримання нових колекцій або знижок на email
Список бажань	Можливість додати товар до списку бажань, переглядати і переміщати до кошика
Порівняння товарів	Функціонал порівняння товарів за ціною, описом і характеристиками в таблиці

На основі аналізу предметної області, аналогів і очікувань цільової аудиторії було визначено перелік основних функціональних вимог до вебплатформи FitIQ. Запропонований набір функцій орієнтований на забезпечення повного циклу онлайн-покупок, від пошуку та вибору товару до його оформлення і подальшої взаємодії з користувачем.

Зокрема, серед ключових функціональних вимог виділено: можливість перегляду товарів із фільтрацією за категоріями, брендом, розміром, кольором, наявністю пошуку за назвою, кошика для покупок, списку бажань, системи порівняння товарів і форми підписки на нові колекції. Крім того, передбачено реалізацію інформаційного розділу з модними новинами для підвищення цінності платформи як джерела експертного контенту [1, с. 59].

Дані функції формують основу користувацького досвіду, сприяють персоналізації взаємодії та відповідають сучасним вимогам до e-commerce рішень. Зазначені функціональні компоненти є ключовими орієнтирами для подальшої технічної реалізації платформи.

Крім того, система платформи має відповідати сучасним стандартам веброзробки, що передбачає дотримання ряду нефункціональних вимог, які безпосередньо не визначають функціонал, однак істотно впливають на якість взаємодії користувача з платформою (табл.2.2). Зокрема, необхідно забезпечити високу швидкодію, що досягається через попередню генерацію сторінок, оптимізацію мережевих запитів та обмеження обсягу передаваних даних.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				24
Змн.	Арк.	№ докум.	Підпис	Дат		

Нефункціональні вимоги до платформи

№	Нефункціональний модуль	Опис вимог
1	Продуктивність	Оптимізоване завантаження контенту завдяки SSR/SSG, використання кешування, оптимізація запитів до бекенду.
2	Адаптивність	Коректне відображення інтерфейсу на смартфонах, планшетах і ПК, підтримка медіа-запитів.
3	Безпека	Захист даних користувача, шифрування паролів, захист від XSS, CSRF, використання HTTPS.
4	Надійність	Безперебійна робота серверної частини, система логування помилок, інформування користувача у разі збоїв.
5	SEO-оптимізація	Використання мета-тегів, генерація sitemap.xml, семантична структура коду, дружні URL.
6	Масштабованість	Можливість додавання нових функцій або сторінок без зміни архітектури, підтримка модульного підходу.
7	Зручність користування (UX)	Інтуїтивно зрозумілий інтерфейс, логічна навігація, адаптивна стилізація, підтримка user-friendly дизайну.

Не менш важливою є здатність інтерфейсу адаптуватися під різні розміри екранів, гарантуючи стабільну роботу платформи як на мобільних, так і на десктопних пристроях. З погляду інформаційної безпеки необхідно впровадити сучасні методи захисту даних: зокрема, шифрування автентифікаційної інформації, реалізацію протоколу HTTPS, а також захист від потенційно небезпечних запитів (XSS, CSRF тощо).

Важливими є також вимоги до стійкості системи: платформа має зберігати функціональність у разі часткових збоїв, а користувач має отримувати інформативні повідомлення про помилки. Підтримка стандартів пошукової оптимізації (SEO) забезпечується за рахунок генерації відповідної структури документу (мета-теги, семантична розмітка, sitemap), що сприяє видимості ресурсу у пошукових системах [16, с. 48].

Масштабованість архітектури платформи дозволяє у майбутньому безболісно розширювати функціонал та додавати нові модулі без зміни ядра. Також важливо забезпечити зручність інтерфейсу, що включає логічну навігацію, послідовну структуру сторінок і гнучку стилізацію з урахуванням принципів UX-дизайну.

		Ковісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				25
Змн.	Арк.	№ докум.	Підпис	Дат		

Таким чином, нефункціональні вимоги забезпечують технічну досконалість платформи, дозволяють досягти стабільної, безпечної, доступної та масштабованої роботи вебресурсу, сприяючи формуванню позитивного користувацького досвіду.

У процесі формування нефункціональних вимог до платформи інтернет-магазину FITIQ було визначено ключові характеристики, що забезпечують її якісну, стабільну та безпечну роботу в умовах реального навантаження. До них належать висока продуктивність, адаптивність інтерфейсу для різних пристроїв, захист даних користувачів, надійність серверної логіки, відповідність вимогам SEO, а також можливість масштабування проєкту у майбутньому. Особливу увагу приділено зручності користувача (UX) – простій навігації, логічному розташуванню елементів і візуальній структурі сторінок. Усі ці аспекти разом спрямовані на створення сучасного, конкурентоспроможного та технічно досконалого вебсервісу для масового українського користувача.

2.2 Розробка бази даних платформи

База даних відіграє ключову роль у функціонуванні вебплатформи, оскільки забезпечує збереження, цілісність та доступність інформації про користувачів, товари, замовлення, підписки, відгуки тощо. Для реалізації обраної архітектури інтернет-магазину було використано MongoDB, яка є документно-орієнтованою системою управління базами даних (СУБД) і добре підходить для динамічних, масштабованих застосунків. Гнучка структура документів дозволяє зберігати вкладені об'єкти, що є доцільним для зберігання пов'язаних сутностей, таких як товари з варіантами кольору, категорії з підкатегоріями, списки бажань тощо.

Кожен функціональний блок платформи підтримується окремою моделлю даних (колекцією), які визначені за допомогою Mongoose-схем. Це дозволяє суворо описати типи, зв'язки між об'єктами та додаткові валідації, що забезпечує надійність структури БД [5, с. 48].

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		26

У проєкті використовуються такі основні моделі:

1. Category.js – модель для категорій та підкатегорій товарів;
2. Coupon.js – модель для зберігання знижкових купонів;
3. Order.js – модель замовлень користувачів, включає інформацію про кошик, оплату та статус;
4. Products.js – основна модель товару (назва, опис, ціна, варіанти, кольори, зображення);
5. Subscription.js – модель для зберігання email-підписок на оновлення колекцій;
6. User.js – модель користувача, що містить дані профілю, статус, стать, вік, історію переглядів тощо.

Кожна з цих моделей має зв'язки з іншими сутностями через ObjectId, що забезпечує зв'язність та гнучкість у побудові запитів до БД [6, с. 29].

У структурі бази даних інтернет-магазину ключовим є зберігання повної та достовірної інформації про користувачів (табл.2.3).

Таблиця 2.3

Опис таблиці User

Поле	Тип	Призначення
name	varchar	Ім'я користувача
email	varchar	Email користувача (унікальний, перевіряється на валідність)
password	varchar	Пароль користувача (зашифрований)
role	enum	Роль користувача (user, admin)
contactNumber	varchar	Контактний номер (мобільний телефон)
shippingAddress	varchar	Адреса доставки
phone	varchar	Номер телефону (необов'язковий)
address	varchar	Адреса користувача (необов'язкова)
bio	text	Біографія (короткий опис, необов'язкова)
status	enum	Статус облікового запису (active, inactive, blocked)
gender	enum	Стать користувача (male, female, child)
age	number	Вік користувача (від 0 до 120)
viewedProducts	ObjectId[]	Список переглянутих товарів (Products)
passwordChangedAt	date	Дата зміни паролю

passwordResetToken	varchar	Токен для скидання паролю
passwordResetExpires	date	Термін дії токена скидання паролю
createdAt	timestamp	Дата створення запису
updatedAt	timestamp	Дата останнього оновлення

Модель User забезпечує комплексний підхід до збереження та обробки даних користувача, що дозволяє реалізувати функціонал реєстрації, входу, історії взаємодій із товарами, відновлення доступу та персоналізованого досвіду. Її структура оптимізована для розширюваності, безпеки й інтеграції з іншими частинами системи.

У структурі бази даних інтернет-магазину модель Products є ключовою, оскільки саме вона зберігає всю інформацію про товари, представлені на платформі. Завдяки цьому модель дозволяє повноцінно реалізувати функціонал перегляду, фільтрації, сортування, рекомендацій, персоналізації й обліку продажів.

Таблиця 2.4

Опис таблиці Products

Поле	Тип	Призначення
id	String	Унікальний ідентифікатор товару
sku	String	Артикул товару
img	String	Основне зображення товару
title	String	Назва товару
slug	String	URL-ідентифікатор товару
unit	String	Одиниця виміру товару
imageURLs	Array	Галерея зображень із кольорами та розмірами
parent	String	Головна категорія
children	String	Підкатегорія
price	Number	Ціна товару
discount	Number	Знижка на товар
quantity	Number	Кількість у наявності
brand	Object	Назва та ідентифікатор бренду
Category	Object	Назва та ідентифікатор категорії
status	String	Статус наявності
reviews	Array<ObjectId>	Список відгуків
productType	String	Тип товару
description	String	Опис товару
embedding	Array<Number>	Ембедінги для рекомендацій
videoId	String	Ідентифікатор відео

additionalInformation	Array<Object>	Додаткова інформація
tags	Array<String>	Теги товару
sizes	Array<String>	Наявні розміри
offerDate	Object	Період акційної пропозиції
featured	Boolean	Позначка рекомендованого товару
sellCount	Number	Кількість продажів
targetAudience	Array<String>	Цільова аудиторія
ageRange	Array<Object>	Віковий діапазон

Таким чином, модель Products охоплює всі необхідні характеристики товару для забезпечення ефективної роботи магазину, підтримуючи різноманітні маркетингові стратегії (знижки, акції, новинки), взаємодію з користувачем (через відгуки, вподобання, персоналізовані пропозиції) та масштабування асортименту. Гнучка структура моделі, наявність вкладених об'єктів (кольори, бренди, категорії) та підтримка системи ембедінгів відкривають можливість інтеграції з сучасними AI-рішеннями для підвищення ефективності продажів.

Модель Category використовується для класифікації товарів за ієрархічною структурою (табл. 2.5).

Таблиця 2.5

Опис таблиці Category

Поле	Тип	Призначення
img	String	Зображення категорії (необов'язкове)
parent	String	Назва головної (батьківської) категорії, унікальна
children	Array of Strings	Підкатегорії, що входять до головної
productType	String	Тип продукції в категорії (у нижньому регістрі)
description	String	Опис категорії (необов'язковий)
products	Array of ObjectId	Список товарів, які належать до категорії
status	String (enum)	Статус категорії: Show або Hide

Модель Category є основою для побудови логічної структури каталогу товарів, яка полегшує користувачам навігацію платформою. Завдяки зв'язку з товарами (products), підтримці вкладеності (через children) і можливості контролю видимості (status), ця модель забезпечує гнучкість у керуванні контентом і адаптацію до змін у продуктовій стратегії магазину.

Модель Coupon (табл. 2.6) призначена для керування знижками, які можуть застосовуватися під час оформлення замовлення. Також модель дозволяє керувати статусом активності кожного купона.

Таблиця 2.6

Опис таблиці Coupon

Поле	Тип	Призначення
title	String	Назва купона
logo	String	Зображення або іконка, яка відображає купон
couponCode	String	Унікальний код купона для активації знижки
startTime	Date	Дата початку дії купона
endTime	Date	Дата завершення дії купона
discountPercentage	Number	Розмір знижки у відсотках
minimumAmount	Number	Мінімальна сума замовлення для використання купона
productType	String	Тип товару, до якого застосовується купон
status	String (enum)	Статус купона: активний або неактивний

Модель Coupon дозволяє ефективно реалізувати маркетингові стратегії на платформі шляхом надання персоналізованих або масових знижок. Завдяки структурованим полям та вбудованому контролю валідності даних, ця модель забезпечує гнучкість у налаштуванні акцій, стимулює продажі й покращує користувацький досвід.

Модель Order (табл. 2.7) є ключовою для управління процесом оформлення та обробки замовлень у системі електронної комерції. Вона фіксує повну інформацію про замовлення, включаючи дані покупця, вміст кошика, адресні та контактні реквізити, вартість замовлення, способи оплати й доставки. Крім того, модель підтримує автоматичну генерацію унікального номера рахунку-фактури (invoice), що є необхідним для обліку замовлень і формування документації.

Таблиця 2.7

Опис таблиці Order

Поле	Тип	Призначення
user	ObjectId (ref: User)	Посилання на користувача, що зробив замовлення
cart	Array	Масив товарів у кошику
name	String	Ім'я отримувача
address	String	Адреса доставки

Email	String	Електронна пошта користувача
contact	String	Контактний номер телефону
city	String	Місто доставки
country	String	Країна доставки
zipCode	String	Поштовий індекс
subTotal	Number	Проміжна сума замовлення
shippingCost	Number	Вартість доставки
discount	Number	Сума знижки
totalAmount	Number	Загальна сума до сплати
shippingOption	String	Обраний варіант доставки
cardInfo	Object	Інформація про картку (якщо збережена)
paymentIntent	Object	Об'єкт платіжного наміру (Stripe)
paymentMethod	String	Метод оплати (наприклад, картка)
orderNote	String	Додаткова інформація до замовлення
invoice	Number	Унікальний номер рахунку-фактури
status	String (enum)	Статус замовлення (наприклад: pending, delivered)

Завдяки моделі Order забезпечується зберігання структурованої та надійної інформації про кожне оформлене замовлення. Це дозволяє ефективно реалізувати функціонал замовлень, спростити інтеграцію з платіжними системами, а також формувати гнучку аналітику щодо продажів. Модель легко масштабовується та адаптується до змін у бізнес-логіці, що забезпечує її актуальність у рамках розвитку торговельної платформи.

Модель Subscription відповідає за зберігання адрес електронної пошти користувачів, які виявили бажання отримувати оновлення про нові колекції, акції або інші новини магазину (табл. 2.8).

Таблиця 2.8

Опис таблиці Subscription

Поле	Тип	Призначення
email	String	Email адреса підписника, використовується для надсилання новин та оновлень
timestamps	Date	Автоматичні мітки створення та оновлення запису

Незважаючи на просту структуру, модель Subscription виконує важливу роль у маркетинговій стратегії платформи. Вона дозволяє легко керувати базою підписників, забезпечуючи масштабованість розсилок та підвищення лояльності користувачів завдяки інформуванню про актуальні пропозиції.

Таким чином, база даних платформи створює надійне середовище для зберігання інформації про користувачів, товари, замовлення й інші компоненти магазину, що є фундаментом стабільної роботи системи й реалізації її бізнес-логіки.

2.3 Проєктування та реалізація алгоритмів роботи системи

Проєктування алгоритмів функціонування веб-платформи для продажу одягу передбачає створення чіткої, логічно завершеної схеми взаємодії користувача з інтерфейсом та серверною частиною на всіх етапах: від перегляду товарів і підписки на оновлення до оформлення замовлення і його обробки. Головна мета – забезпечення інтуїтивно зрозумілого, безпечного й безперебійного користування ресурсом.

Платформа реалізована на основі стеку MERN з Next.js як фреймворком для фронтенду та Express.js у зв'язці з MongoDB для бекенду [28]. Зберігання даних реалізовано через ODM Mongoose. Для керування станом використано Redux Toolkit [20] із збереженням даних у localStorage через Redux Persist. Серверна логіка обробляє запити REST API, зокрема для користувачів, товарів, замовлень, підписок, відгуків і новин [26].

У проєкті реалізовано ключові користувацькі сценарії, які забезпечують зручну та функціональну взаємодію з платформою.

Перш за все, система дозволяє проходити реєстрацію та авторизацію. Користувач може створити обліковий запис за допомогою електронної пошти та пароля. Всі введені дані перевіряються на сервері: перевіряється унікальність електронної адреси, а пароль надійно хешується за допомогою бібліотеки bcrypt. Після успішного входу система генерує JWT-токен, який зберігається у Redux Store. Це дозволяє реалізувати зручну функцію автологіна при повторному відвідуванні сайту.

Користувач може зручно переглядати каталог товарів, який поділений за категоріями, типами, кольорами тощо. Також реалізовано пошук за назвою

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				32
Змн.	Арк.	№ докум.	Підпис	Дат		

товару. Усі запити обробляються через API, а результати фільтруються згідно з вибраними параметрами.

Якщо користувач знаходить цікаві товари, він може додати їх до кошика або до списку бажань. Ця функція доступна лише після авторизації. Додані товари зберігаються в Redux Store та дублюються у localStorage. Крім того, кожна дія фіксується у спеціальній моделі переглядів, що в подальшому використовується для побудови персоналізованих рекомендацій.

Після вибору товарів користувач може перейти до оформлення замовлення (рис. 2.1). На сторінці оформлення /checkout він заповнює форму, де вказує свої контактні дані, адресу доставки, обирає спосіб доставки та метод оплати — або через систему Stripe, або післяплатою. Форма перевіряється за допомогою React Hook Form та Yup, а після підтвердження замовлення дані записуються до бази даних. Користувач перенаправляється на сторінку Stripe для завершення оплати.

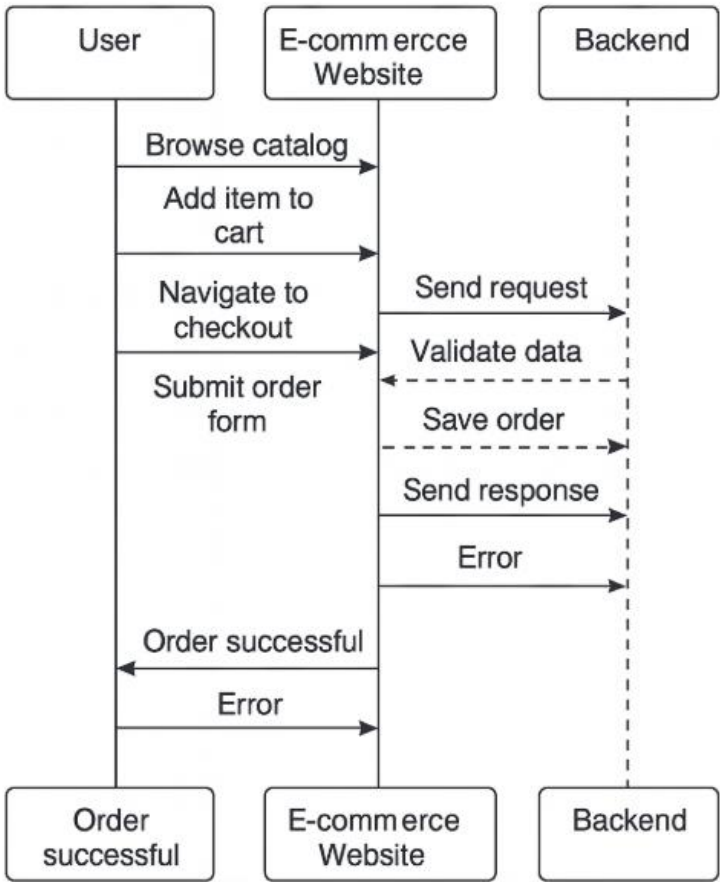


Рис.2.1 – Діаграма послідовності оформлення замовлення

Крім основного функціоналу магазину, на сайті також реалізовано форму підписки. Вона розташована внизу сторінки й дозволяє користувачеві залишити свою email-адресу. Після цього він буде отримувати листи з інформацією про нові колекції та надходження товарів.

Також на платформі створено окрему сторінку для перегляду модних новин. Новини отримуються через зовнішнє API – наприклад, з використанням RSS або власного бекенду. Вони відображаються у вигляді структурованих карточок із заголовками, зображеннями та коротким описом, що робить сторінку інформативною й привабливою для користувача.

Таким чином, платформа охоплює всі ключові етапи взаємодії користувача із системою: від першого відвідування сайту до оформлення покупки та отримання повідомлень про оновлення. Побудована логіка забезпечує гнучкість, захищеність і масштабованість, що дозволяє системі адаптуватися до зростання кількості користувачів та розширення функціоналу в майбутньому.

2.4 Реалізація функціоналу платформи

Одним із ключових елементів електронної комерції є відображення асортименту продукції. У розробленій платформі цей функціонал реалізовано за допомогою клієнтської логіки, яка динамічно отримує перелік товарів з бекенд-серверу через API-запити.

```
export const fetchAllProducts = createAsyncThunk(
  "products/fetchAll",
  async (_, { rejectWithValue }) => {
    try {
      const res = await fetch(`${BASE_URL}/api/product/all`);
      if (!res.ok) throw new Error("Failed to fetch products");
      return await res.json();
    } catch (error) {
      return rejectWithValue(error.message);
    }
  }
);
```

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		34

Отримані дані обробляються та виводяться у вигляді окремих карток товарів, де користувач бачить основну інформацію: назву, ціну, зображення, варіанти кольору.

Система кошика реалізована на клієнтській частині за допомогою Redux. Кожне натискання кнопки «Додати до кошика» викликає асинхронну функцію, яка перевіряє, чи товар вже присутній у кошику, й у разі повторного додавання змінює його кількість.

```
const handleAddProduct = (prd) => {  
  dispatch(add_cart_product(prd));  
  dispatch(trackViewedProduct(prd._id));  
};
```

Для підвищення персоналізації та покращення взаємодії користувача з платформою, у системі реалізовано базовий механізм формування рекомендаційних товарів на основі поведінкових даних. Зокрема, відстежуються товари, які користувач переглядав, додавав до кошика або зберігав у список бажань.

На клієнтській частині відслідковування відбувається за допомогою Redux-екшену:

```
export const apiTrackViewedProduct = async (productId) => {  
  const res = await fetch(`${BASE_URL}/api/user/cart-track`, {  
    method: "POST",  
    headers: {  
      "Content-Type": "application/json",  
      Authorization: `Bearer ${localStorage.getItem("token")} || ""`,  
    },  
    body: JSON.stringify({ productId }),  
  });  
  
  if (!res.ok) {  
    const errorData = await res.json();  
    throw new Error(errorData.message || "Failed to track viewed product");  
  }  
  
  return res.json();  
};
```

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		35

Цей код забезпечує авторизовану передачу ідентифікатора товару (productId) до бекенду, де інформація зберігається в спеціалізованій колекції Tracking. Це дозволяє фіксувати дії користувача щодо взаємодії з товаром, включаючи перегляди та додавання до кошика або бажаного.

На основі зібраної інформації формується підбірка рекомендованих товарів. Для цього бекенд аналізує:

- ідентифікатори переглянутих товарів;
- категорії, до яких належать ці товари;
- стать (gender) та віковий діапазон (ageRange) користувача, що зберігається у його профілі;
- частоту взаємодії з певними категоріями чи брендами.

Відповідний контролер (Додаток Б) виконує запит на підбір найбільш релевантних товарів, фільтруючи їх за статтю, віком та історією переглядів.

Рекомендаційні системи, що ґрунтуються на ембедінгах, є одними з найбільш ефективних у контексті семантичного узагальнення даних. На відміну від класичних колаборативних фільтрів чи простого порівняння категорій, метод ембедінгів дозволяє враховувати не лише поверхневі ознаки (колір, бренд, розмір), а й приховану семантичну структуру товарів. Для кожного товару формується вектор ознак – ембедінг, що узагальнює його семантичне представлення в багатовимірному просторі [30].

Для формування векторного подання кожного товару ми використовували API від Google Generative AI, зокрема модель embedding-001. Кожен товар представлено вектором із 768 значень з плаваючою комою, які кодують семантику назви й опису. Перевага даної моделі – універсальність і висока якість при роботі з текстами різної тематики (мода, одяг, стиль).

Спочатку потрібно налаштувати векторний індекс в MongoDB Atlas (рис. 2.2). У хмарній платформі MongoDB Atlas, ми активували розділ Atlas Search, що дозволяє створювати векторні індекси. Це є обов’язковою умовою для швидкого й ефективного пошуку за ембедінгами.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		36

У кластері Cluster0, на колекції products бази clothing-platform, був створений індекс vector_index типу vectorSearch. Він покриває поле embedding.

Розмір кожного вектора: 768 чисел (float). Всі вектори індексуються через спеціальну конфігурацію Atlas Search. Векторний індекс дозволяє виконувати запити kNN (найближчі сусіди).

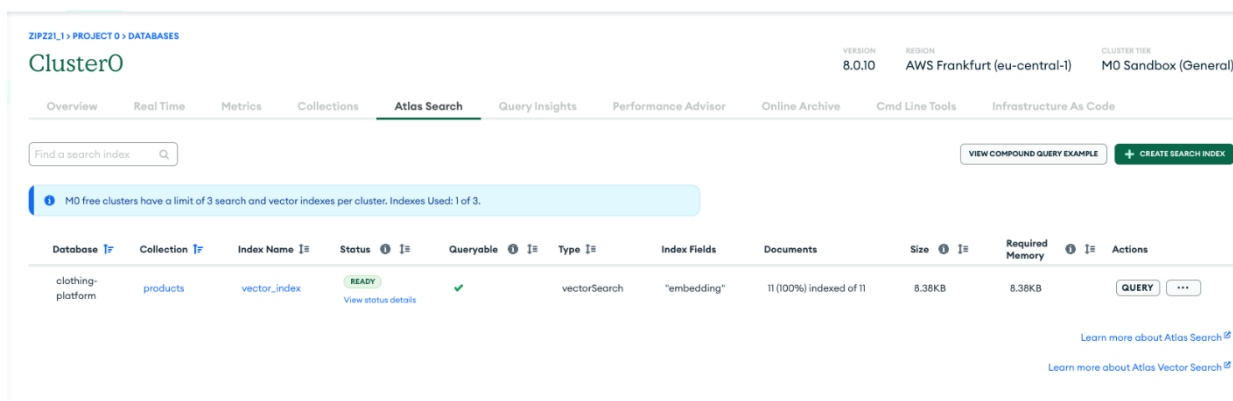


Рис. 2.2 – Налаштування векторного індексу для поля embedding у MongoDB Atlas

Для автоматичної генерації ембедінгів до всіх товарів, у яких це поле відсутнє, було реалізовано Node.js-скрипт updateEmbeddings.js, який: підключається до MongoDB [28]; витягує товари без ембедінгу; формує запит на отримання векторного подання через API; оновлює колекцію products, додаючи поле embedding.

```
async function updateEmbeddings() {
  try {
    await client.connect();
    const db = client.db("clothing-platform");
    const products = db.collection("products");

    const cursor = products.find({ embedding: { $exists: false } });

    let updatedCount = 0;

    for await (const product of cursor) {
      const title = product.title || "";
      const description = product.description || "";
      const text = `${title} ${description}`.trim();
```

```

if (!text) {

    console.warn(`Пропущено (порожній текст) _id=${product._id}`);
    continue;
}

const embedding = await getEmbedding(text);

await products.updateOne({ _id: product._id }, { $set: { embedding } });

updatedCount++;
}
console.log(`\n Готово! Оновлено ${updatedCount} продуктів.`);
} catch (err) {
    console.error("Помилка при оновленні:", err.message);
} finally {
    await client.close();
    process.exit(0);
}
}

```

Реалізація запиту рекомендацій для окремого товару відбувається через спеціальний маршрут REST API:

GET /api/products/:id/recommendations

Функціональна логіка реалізована у функції `getRecommendations`, що виконує наступну послідовність дій:

1. Завантаження цільового товару. Спочатку з бази даних витягується товар за вказаним `id`. Перевіряється наявність ембедінгу, адже без нього обчислення неможливе.

```
const currentProduct = await Product.findById(productId);
```

2. Вибірка всіх інших товарів із ембедінгами. Відбираються всі інші товари (крім поточного), які мають заповнене поле `embedding`.

```

const allProducts = await Product.find({
  _id: { $ne: productId },
  embedding: { $exists: true },
});

```

3. Обчислення косинусної подібності. Для кожної пари: поточний товар – інший товар, розраховується косинусна подібність між ембедінгами. Це значення у діапазоні $[0,1]$, де 1 означає максимальну схожість.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		38

```

export const cosineSimilarity = (vecA, vecB) => {
  if (
    !Array.isArray(vecA) ||
    !Array.isArray(vecB) ||
    vecA.length !== vecB.length
  ) {
    throw new Error("Embeddings must be arrays of the same length");
  }

  const dotProduct = vecA.reduce((sum, a, i) => sum + a * vecB[i], 0);
  const magnitudeA = Math.sqrt(vecA.reduce((sum, a) => sum + a * a, 0));
  const magnitudeB = Math.sqrt(vecB.reduce((sum, b) => sum + b * b, 0));

  if (magnitudeA === 0 || magnitudeB === 0) return 0;

  return dotProduct / (magnitudeA * magnitudeB);
};

```

4. Сортуювання та відбір топ-3 рекомендованих товарів. Результати сортуються за спаданням значення подібності, і обираються три товари з найвищою оцінкою.

```

const sorted = similarities.sort((a, b) => b.score - a.score);
const top = sorted.slice(0, 3).map((item) => item.product);

```

Результат надсилається у відповідь у форматі JSON (рис. 2.3).

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		39

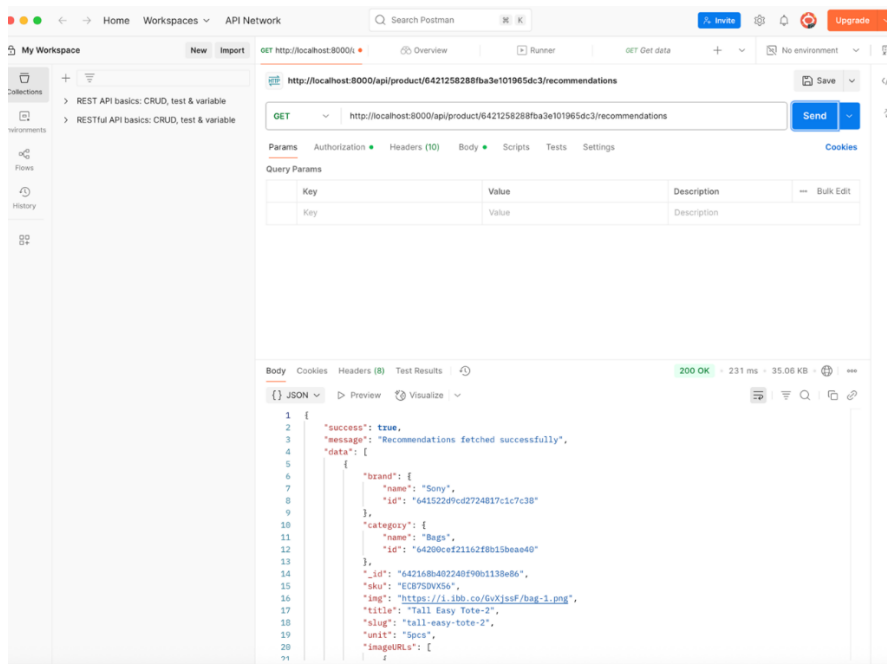


Рис. 2.3 – Виконання запиту на отримання рекомендованих товарів за ембедінгами через інструмент Postman

Запропонований підхід до побудови системи рекомендацій товарів на основі векторних представлень (ембедінгів) має низку суттєвих переваг, які забезпечують його ефективність у середовищі електронної комерції, зокрема в контексті магазину з продажу одягу.

По-перше, система забезпечує високий рівень семантичної релевантності результатів. На відміну від традиційних методів, що опираються виключно на категоріальні атрибути (тип, колір, бренд, тощо), реалізоване рішення аналізує глибинну змістовність товару. Це досягається шляхом побудови числових ембедінгів, які відображають комплексну семантику текстових описів, назв та інших характеристик. Таким чином, рекомендації формуються не лише на основі формальної схожості, а й за прихованими смисловими зв'язками між товарами.

По-друге, запропонована модель є гнучкою та масштабованою. Її універсальність полягає у здатності працювати з будь-якими наборами даних, незалежно від типу чи стилю товарів, кількості записів у базі даних або динаміки оновлення контенту. Зміни в асортименті не вимагають перебудови архітектури

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				40
Змн.	Арк.	№ докум.	Підпис	Дат		

– достатньо оновити ембедінг для нового товару, після чого система автоматично інтегрує його у векторний простір.

Третьою важливою перевагою є висока швидкодія обробки запитів. Завдяки тому, що обчислення ембедінгів виконується на етапі створення або оновлення товару, під час пошуку рекомендацій не виникає потреби у повторному семантичному аналізі. Система працює виключно з векторними представленнями, що зберігаються безпосередньо в базі даних MongoDB, використовуючи векторний індекс Atlas Vector Search. Це дозволяє оперативно знаходити найближчі товари за косинусною подібністю з мінімальним навантаженням на сервер.

У сукупності, зазначені переваги сприяють суттєвому підвищенню рівня персоналізації взаємодії користувача з платформою. Реалізована система рекомендацій забезпечує точне групування схожих товарів, що позитивно впливає на користувацький досвід, підвищує ймовірність здійснення покупки (конверсію).

Таким чином, описаний підхід повністю відповідає сучасним тенденціям у сфері штучного інтелекту та рекомендаційних технологій для e-commerce і може бути ефективно масштабований для потреб великих онлайн-платформ.

Оформлення замовлення реалізовано через сторінку /checkout, де користувач заповнює дані форми, включаючи адресу доставки, контактну інформацію та спосіб оплати.

Після успішної валідації, дані замовлення надсилаються до бекенду через асинхронний POST-запит:

```
export const apiSaveOrder = async (data) => {  
  const response = await axios.post(`${BASE_URL}/api/order/saveOrder`, data);  
  return response.data;  
};
```

На бекенді замовлення зберігається в колекції orders, і для кожного нового запису автоматично генерується унікальний номер рахунку (invoice):

```
exports.addOrder = async (req, res, next) => {  
  try {  
    const orderItems = await Order.create(req.body);
```

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		41

```

res.status(200).json({
  success: true,
  message: "Order added successfully",
  order: orderItems,
});
}
catch (error) {
  console.log(error);
  next(error)
}
};

```

Для прийому онлайн-платежів реалізована інтеграція зі Stripe [32]. Користувач обирає метод оплати на етапі оформлення замовлення. У разі вибору Stripe – фронтенд надсилає запит до бекенду, де створюється сесія оплати:

```

const stripe = require("stripe")(secret.stripe_key);
const Order = require("../model/Order");
exports.paymentIntent = async (req, res, next) => {
  try {
    const product = req.body;
    const price = Number(product.price);

    const amount = price * 100;
    const paymentIntent = await stripe.paymentIntents.create({
      currency: "usd",
      amount: amount,
      payment_method_types: ["card"],
    });
    res.send({
      clientSecret: paymentIntent.client_secret,
    });
  } catch (error) {
    console.log(error);
    next(error)
  }
};

```

У межах реалізації функціоналу платформи було розроблено повноцінний інтернет-магазин з повним циклом користувацької взаємодії – від перегляду товарів до здійснення замовлення. Особливу увагу приділено зручності інтерфейсу, безпеці операцій та адаптивності архітектури до майбутнього масштабування.

Користувач має змогу ознайомлюватися з товарами, які динамічно завантажуються з бази даних через API, фільтрувати їх за категоріями, кольором,

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				42
Змн.	Арк.	№ докум.	Підпис	Дат		

а також додавати до кошика. Реалізовано надійну систему авторизації та захисту даних [14], а замовлення оформлюються через сторінку /checkout із перевіркою введених даних та можливістю оплати як через Stripe API [33], так і офлайн (накладеним платежем).

Особливу цінність для користувача має впроваджена рекомендаційна система, що враховує історію переглядів, демографічні параметри (стать, вік), а також категорійну належність товарів. Це забезпечує персоналізований підхід до кожного клієнта та підвищує ймовірність повторних покупок.

Таким чином, реалізований функціонал платформи відповідає сучасним вимогам електронної комерції, поєднуючи ефективну архітектуру, гнучкість інтерфейсу та інтелектуальну логіку обробки даних.

Висновки до другого розділу.

У процесі розробки інтернет-платформи для продажу одягу було спроектовано архітектуру вебзастосунку, що поєднує клієнтську, серверну частину та базу даних у єдину функціональну систему. Було здійснено поетапне моделювання структури платформи, з урахуванням вимог до зручності, масштабованості та адаптивності інтерфейсу.

На основі аналізу функціональних вимог було побудовано схему бази даних, яка охоплює ключові сутності: користувачів, товари, категорії, замовлення, купони, підписки, а також логіку збереження інформації про перегляди користувачів. Особливу увагу приділено персоналізації – у моделях збережено вік, стать, історію переглянутих товарів, що є передумовою до побудови системи рекомендацій. Для реалізації цих сценаріїв було застосовано сучасні інструменти, зокрема React, Redux, Express.js, MongoDB та Stripe API. Такий підхід забезпечив комплексну реалізацію функціоналу, що відповідає сучасним вимогам e-commerce платформ.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		43

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ПЛАТФОРМОЮ ПРОДАЖУ ОДЯГУ

3.1 Користувацький інтерфейс і адаптивність верстки

Користувацький інтерфейс (UI) відіграє ключову роль у забезпеченні позитивного досвіду взаємодії відвідувачів із вебплатформою. У контексті електронної комерції інтерфейс повинен бути не лише естетично привабливим, але й інтуїтивно зрозумілим, функціональним і доступним на всіх типах пристроїв. Саме тому при розробці інтерфейсу інтернет-магазину було обрано сучасні підходи до верстки, побудовані на принципах UX/UI-дизайну та адаптивності.

Забезпечення адаптивної верстки є критично важливим для охоплення максимально широкої аудиторії користувачів. Сьогодні все більше покупок здійснюється зі смартфонів і планшетів, тому сайт повинен коректно відображатися на екранах з різною роздільною здатністю.

На рис.3.1 зображено приклад візуалізації головної сторінки вебплатформи FitIQ у двох режимах – для десктопних пристроїв та мобільних екранів. Як видно з ілюстрації, компонування елементів інтерфейсу змінюється відповідно до роздільної здатності екрана, зберігаючи при цьому функціональність та візуальну цілісність.

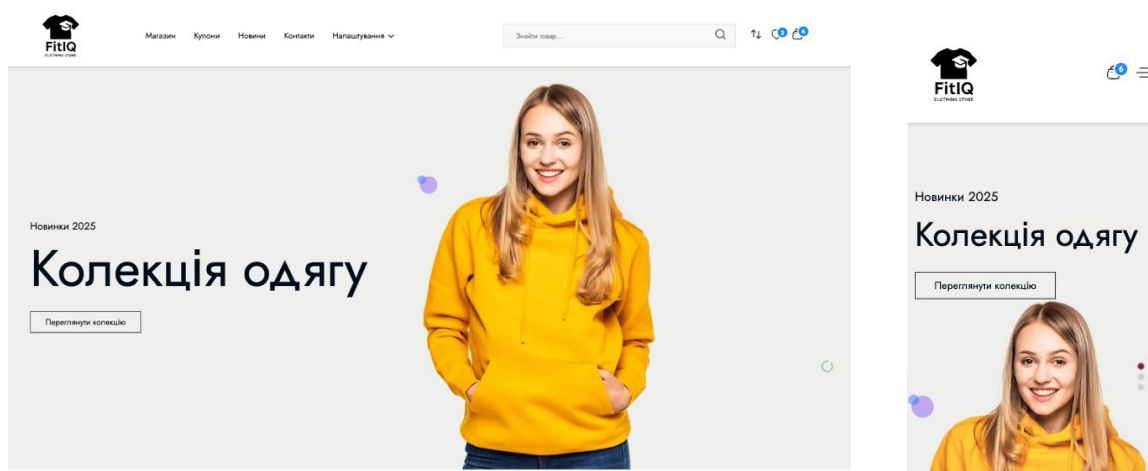


Рис. 3.1 – Головна сторінка платформи в десктопній версії та адаптивне відображення на мобільному пристрої

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		44

Верстка платформи реалізована з використанням технології SCSS, що дозволяє організувати стилізацію проєкту модульно та з урахуванням каскадної ієрархії. Для досягнення адаптивності було застосовано медіа-запити (@media) у SCSS-файлах, що забезпечують зміну властивостей стилів залежно від ширини вікна браузера. Наприклад, блок із заголовком, кнопкою та зображенням моделі в мобільному варіанті автоматично перекомпоновується у вертикальний формат, що значно покращує юзабіліті на смартфонах.

Адаптивність інтерфейсу також передбачає зміну розміру шрифтів, відступів, масштабування графічних елементів і перехід до мобільного меню, що видно на правій частині рисунку. Завдяки цьому забезпечується комфортне сприйняття контенту на різних типах пристроїв без необхідності горизонтального прокручування чи масштабування вручну.

Таким чином, реалізація адаптивного дизайну дозволила створити уніфіковану систему відображення сайту, яка задовольняє потреби як користувачів ПК, так і мобільних пристроїв. Такий підхід є критично важливим для сучасних e-commerce рішень, орієнтованих на максимальне охоплення аудиторії.

Сторінка магазину є ключовим елементом e-commerce платформи, оскільки саме тут користувач взаємодіє з основним контентом – товарами. Її структура та функціональність мають вирішальне значення для зручності навігації, швидкого пошуку потрібних позицій та формування позитивного досвіду користувача. У процесі розробки було враховано як технічні аспекти верстки й адаптивності, так і принципи UI/UX-дизайну.

На рис. 3.2 представлено реалізовану сторінку магазину, що містить динамічно сформований каталог товарів. Основну частину інтерфейсу займає сітка товарів, що складається з окремих карток, кожна з яких включає зображення продукту, його назву, короткий опис і вартість. Такий підхід дозволяє користувачеві швидко ознайомитись із доступними позиціями та перейти до детального перегляду.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		45

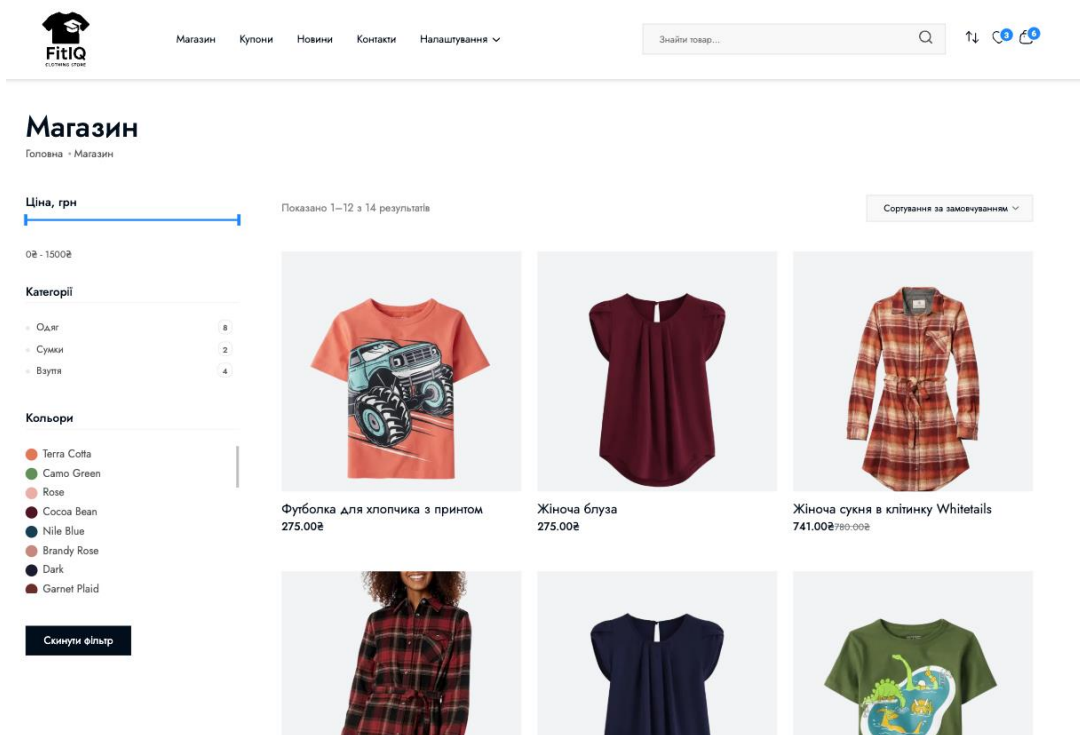


Рис. 3.2 – Сторінка магазину з каталогом товарів

Зліва відображено панель фільтрації, яка дозволяє здійснювати сортування за ціною (із застосуванням повзунка), категорією, кольором та іншими параметрами. Реалізація фільтрів відбувається шляхом обробки запитів до серверної частини з подальшою динамічною генерацією відповідного контенту на клієнтській стороні.

На рис. 3.3 продемонстровано функціональність інтерфейсу інтернет-магазину, що передбачає відображення інформації про вибраний товар у вигляді модального вікна.

Зображення ілюструє реалізацію UI-компонента, що дозволяє користувачеві швидко переглядати фото товару, дізнаватися ключову інформацію (назва, доступність, опис, ціна, кольори, кількість) та здійснювати дії («Додати до кошика», «Купити зараз») – без переходу на нову сторінку. Такий підхід підвищує зручність користування, скорочує шлях до покупки й покращує загальний UX (user experience) сайту.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				46
Змн.	Арк.	№ докум.	Підпис	Дат		

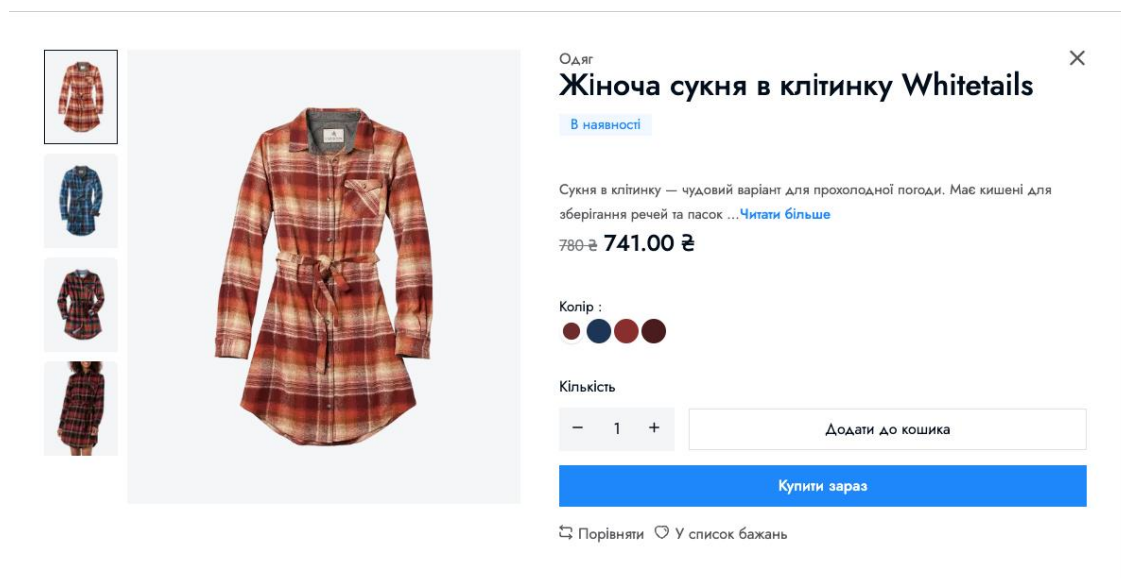


Рис. 3.3 – Відображення інформації про товар у модальному вікні платформи

На рис. 3.4 представлено приклад блоку рекомендацій, який з'являється при перегляді сторінки конкретного товару в інтернет-магазині. Такий блок автоматично формує добірку схожих товарів за допомогою алгоритму, що аналізує семантичну близькість текстових описів товарів у векторному просторі (embedding).



Рис. 3.4 – Рекомендації товарів на основі схожості ембедінгів

		Конвісар А.С.		
		Нерода С.І.		
Змн.	Арк.	№ докум.	Підпис	Дат

Зображення ілюструє реалізацію системи товарних рекомендацій, що працює за принципом пошуку схожих товарів на основі ембедінгів. При відкритті певного товару, його унікальний опис та ID використовуються для отримання embedding-вектора через модель embedding-001 бібліотеки Google Generative AI. Далі система обчислює відстані між векторами і виводить найбільш подібні товари. Це покращує користувацький досвід і підвищує ймовірність покупки за рахунок релевантних пропозицій.

Інтерфейс сторінки магазину розроблений із урахуванням сучасних вимог до зручності навігації та користувацького досвіду. Для ефективної взаємодії користувача з товарним асортиментом передбачено функціонал фільтрації продукції за різними параметрами (ціна, категорія, колір), можливість сортування за критеріями та пошуку товарів за ключовими словами (рис. 3.5). Завдяки цим функціональним можливостям користувач може швидко знайти релевантні товари, що підвищує загальну ефективність використання платформи.

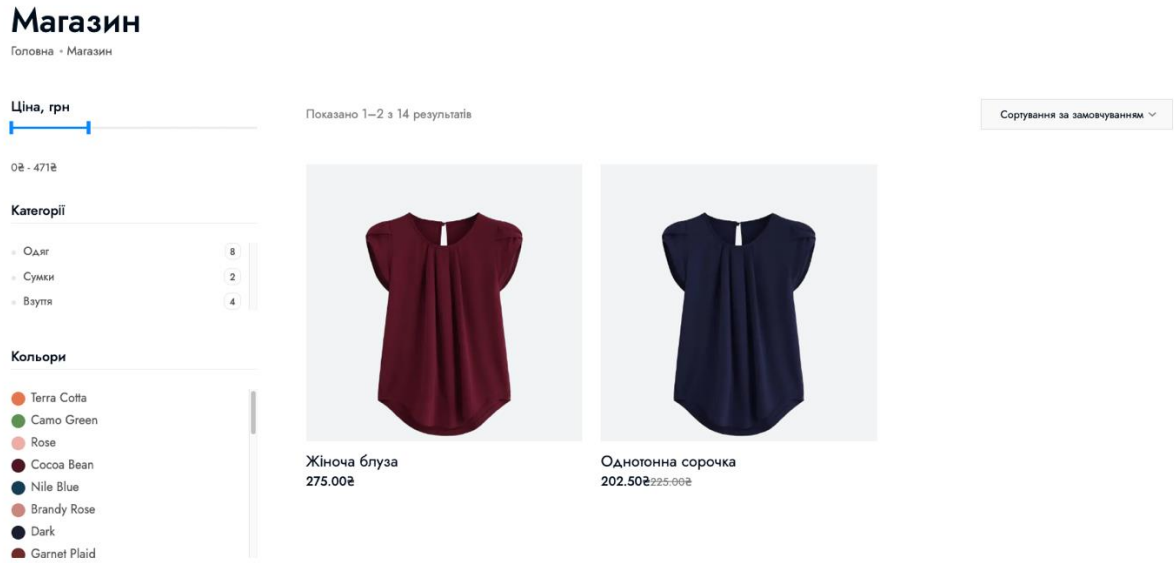


Рис. 3.5 – Інтерфейс сторінки магазину з фільтрами за категорією, кольором і ціною

Реалізований функціонал фільтрації дозволяє користувачу легко адаптувати відображення товарів під власні запити. Зокрема, завдяки повзунку ціни можливо задати діапазон вартості, а також обрати категорію чи колір

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		48

товарів. Це дозволяє зменшити кількість виведених результатів до лише релевантних, що сприяє зручності прийняття рішення про покупку.

Функція сортування дає змогу впорядковувати результати за логікою користувача: за ціною (від дешевих до дорогих або навпаки), за новизною товарів або за наявністю акцій (рис. 3.6). Це розширює можливості персоналізованого перегляду асортименту та покращує UX, особливо в умовах великої кількості товарів.

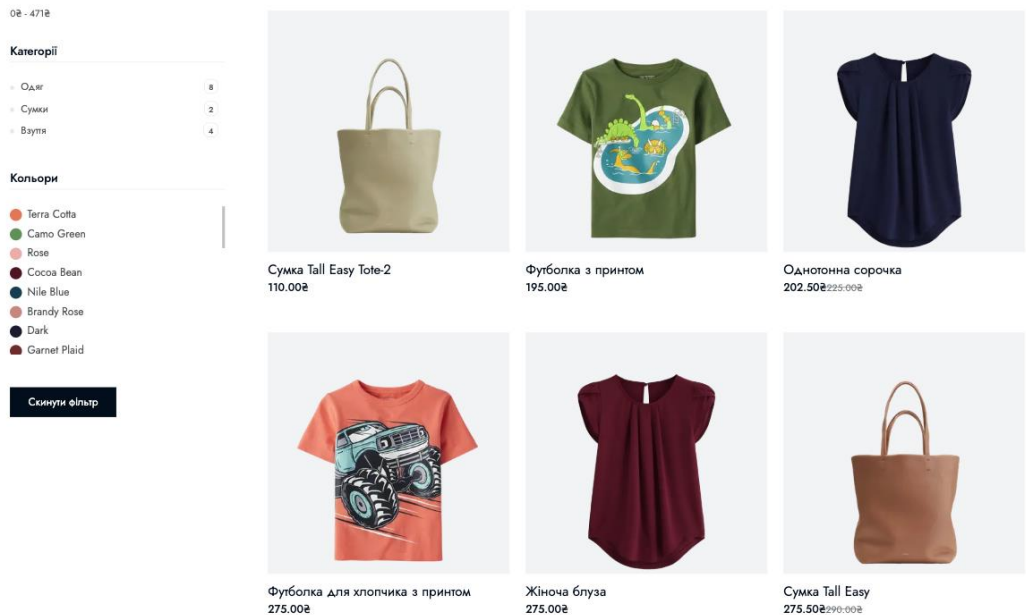


Рис. 3.6 – Механізм пошуку товарів за ключовими словами

Інтеграція поля пошуку забезпечує оперативний доступ до конкретного товару шляхом введення його назви або ключових слів. Пошукова система платформи адаптована для обробки користувацького запиту та швидкого оновлення результатів, що дозволяє мінімізувати час взаємодії з інтерфейсом у разі чітко сформульованого запиту користувача.

Інтеграція персоналізованих рекомендацій є ключовим елементом сучасних e-commerce платформ, що орієнтовані на підвищення конверсії та поліпшення взаємодії з користувачем. У платформі FitIQ передбачено відображення блоку «Рекомендовано для вас» (рис. 3.7), який стає доступним лише для зареєстрованих користувачів. Формування цих рекомендацій ґрунтується на аналізі вподобань

конкретного користувача, зокрема переглянутих, збережених або доданих до кошика товарів. Для точнішої персоналізації система також враховує вік і стать користувача, які мають бути попередньо заповнені у його профілі.

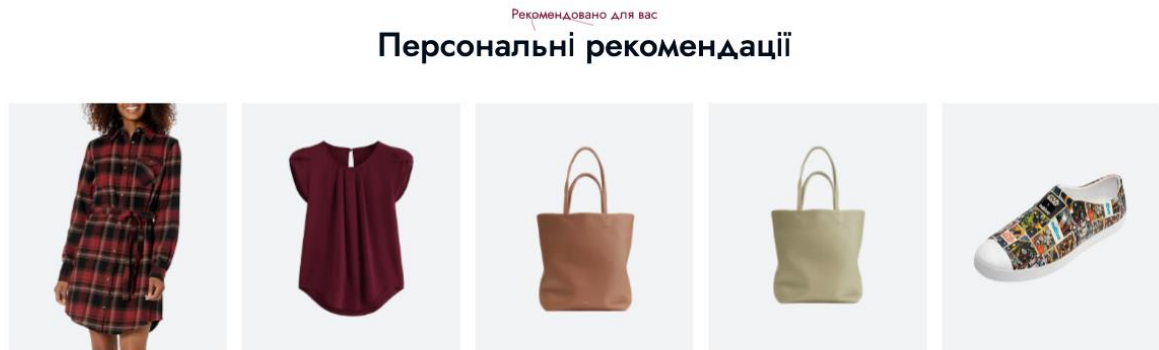


Рис. 3.7 – Блок персональних рекомендацій для авторизованого користувача на головній сторінці платформи

Персоналізовані рекомендації дозволяють підвищити релевантність запропонованих товарів і створити ефект «індивідуального підходу» до кожного користувача. Це сприяє кращому користувацькому досвіду, заохочує до повторних взаємодій із платформою та підвищує ймовірність здійснення покупки. Реалізація даного механізму у FitIQ демонструє використання сучасних підходів до персоналізації у сфері електронної комерції.

Отже, реалізація користувацького інтерфейсу платформи FitIQ продемонструвала високий рівень продуманості дизайну, зручності навігації та візуальної привабливості. Особлива увага приділена адаптивній верстці, що забезпечує коректне відображення контенту на пристроях із різними розмірами екранів – від настільних ПК до мобільних телефонів. Завдяки використанню SCSS було досягнуто гнучке стилізування компонентів інтерфейсу з урахуванням медіа-запитів, що дозволяє платформі відповідати сучасним вимогам до UX/UI-дизайну. Такий підхід створює комфортні умови для користувачів і сприяє підвищенню ефективності взаємодії з платформою незалежно від пристрою, що використовується.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		50

3.2 Реалізація особистого кабінету, реєстрації та авторизації

У сучасних вебплатформах особистий кабінет користувача відіграє ключову роль у забезпеченні персоналізованого досвіду взаємодії, а механізми реєстрації та авторизації виступають базовими інструментами для захисту доступу до конфіденційних даних. У процесі розробки платформи FitIQ особливу увагу було приділено створенню надійної та зручної системи автентифікації користувачів.

На платформі FitIQ реалізовано простий та інтуїтивно зрозумілий інтерфейс реєстрації нового користувача. Форма реєстрації (рис. 3.8) містить базові поля, необхідні для створення облікового запису: ім'я, електронна пошта та пароль. Крім того, користувач має можливість підтвердити свою згоду з політикою конфіденційності, що є обов'язковою умовою відповідно до чинного законодавства в галузі захисту персональних даних.

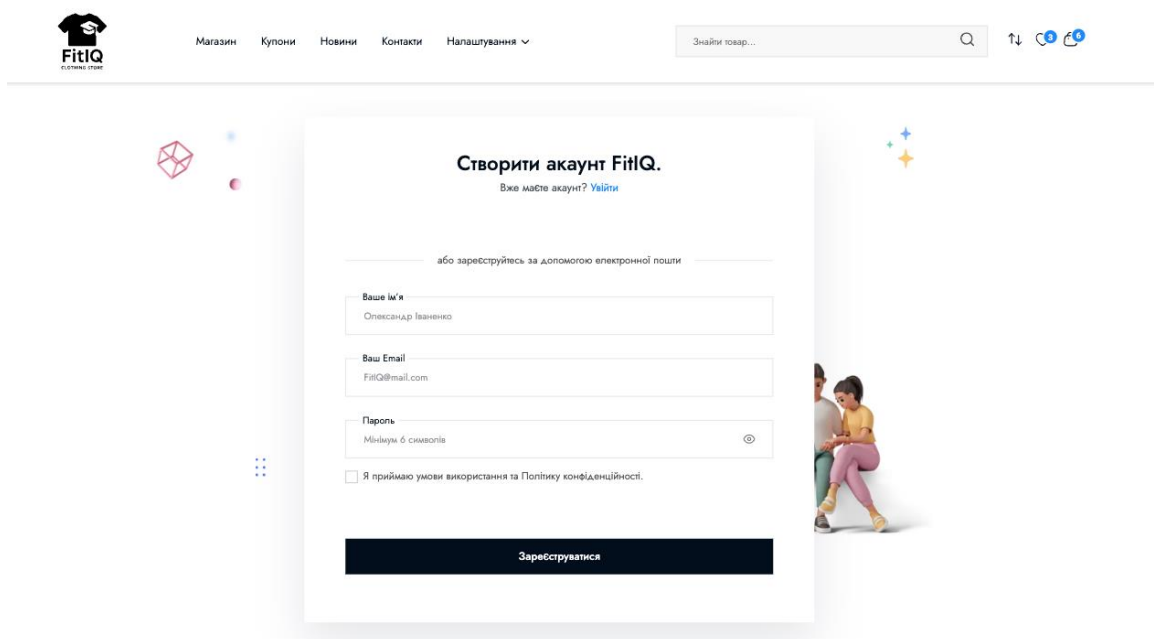


Рис. 3.8 – Форма реєстрації нового користувача на платформі FitIQ

Реалізована форма реєстрації на FitIQ відповідає сучасним вимогам зручності та безпеки: мінімалістичний дизайн, адаптивна верстка, обов'язкові валідаційні перевірки та можливість автентифікації через зовнішні сервіси. Це забезпечує ефективну початкову взаємодію користувача з платформою та формує позитивний перший досвід.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				51
Змн.	Арк.	№ докум.	Підпис	Дат		

Після успішного заповнення форми реєстрації, система автоматично додає нового користувача до бази даних. На рис. 3.9 представлено структуру документа, що зберігається в колекції користувачів. До об'єкта включено унікальний ідентифікатор `_id`, ім'я, електронну пошту, хешований пароль, роль користувача (за замовчуванням – “user”), статус облікового запису, а також службову інформацію щодо часу створення та останнього оновлення запису.

```
_id: ObjectId('685e3004b555d2dc50242c1b')
name: "Tester"
email: "tester@gmail.com"
password: "$2a$10$dkNNnM7pt3Xa8CIYj1Y5XeCRlPMMp2YF4WklLlAc7eYJs1dqdBweG"
role: "user"
status: "active"
▸ reviews: Array (empty)
▸ viewedProducts: Array (empty)
createdAt: 2025-06-27T05:45:40.821+00:00
updatedAt: 2025-06-27T05:45:40.873+00:00
__v: 0
```

Рис. 3.9 – Документ користувача у базі даних після реєстрації

Збереження користувача до бази даних після реєстрації є ключовим етапом формування його персонального простору на платформі FitIQ. Надійна обробка та зберігання даних, включно з хешуванням паролів, дозволяє забезпечити безпеку та цілісність інформації, а також закладає основу для персоналізованої взаємодії з користувачем у подальшому.

Однією з ключових функцій особистого кабінету є можливість користувача змінювати свої персональні дані (рис. 3.10). Це включає оновлення імені, електронної пошти, пароля, віку, статі, що може бути використано для персоналізації платформи.

Інформація подається в структурованому вигляді у вигляді форми, де користувач має змогу змінити основні поля, зокрема вік, стать та вподобання. Такий функціонал є важливим для побудови персоналізованих рекомендацій та покращення користувацького досвіду.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				52
Змн.	Арк.	№ докум.	Підпис	Дат		

FitIQ

Магазин Купони Новини Контакти Надашування

Знайти товар...

Профіль

Інформація

Мої замовлення

Особисті дані

Ім'я: test

Електронна пошта: test@gmail.com

Введіть номер телефону

Введіть адресу

Введіть місто

Оберіть стать

Коротко про себе

Зберегти профіль

Рис. 3.10 – Інтерфейс редагування персональних даних користувача

Важливим елементом особистого кабінету є можливість перегляду історії здійснених замовлень (рис. 3.11). Це дозволяє користувачам аналізувати власні покупки, а також за необхідності повторно оформлювати замовлення або звертатися до служби підтримки щодо конкретного товару.

Профіль

Інформація

Мої замовлення

ID замовлення	Дата замовлення	Статус
#2eb8	27 June 2025	pending

Мій кабінет
Профіль

Інформація
Контакти

Зв'язатися з нами

Питання? Телефонуйте:
+38 (098) 123 45 67

fitiq@gmail.com

Київ, вул. Боровського, 6

Рис. 3.11 – Сторінка з історією замовлень користувача

Такий функціонал забезпечує прозорість транзакцій і контроль за виконанням замовлень. Інтеграція історії замовлень у профіль користувача підвищує зручність взаємодії з платформою, дозволяючи користувачеві залишатися проінформованим про поточний статус покупок та формувати очікування щодо обслуговування. Це також сприяє підвищенню довіри до платформи.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				53
Змн.	Арк.	№ докум.	Підпис	Дат		

Отже, реалізація особистого кабінету на платформі FitIQ є ключовим компонентом взаємодії користувача з системою, що забезпечує персоналізований досвід користування. Завдяки впровадженню функціоналу реєстрації та авторизації користувачі мають змогу створювати облікові записи, захищені паролем, а також редагувати власні персональні дані. Платформа передбачає збереження даних користувача у базі даних, що забезпечує стійке функціонування і можливість використання цих даних для подальшої персоналізації – зокрема для формування індивідуальних товарних рекомендацій.

Крім того, особистий кабінет включає зручний інтерфейс редагування профілю, а також функцію перегляду історії замовлень, що сприяє підвищенню прозорості й контролю з боку користувача. Така комплексна реалізація не лише забезпечує зручність користувачів, а й формує основу для розвитку системи лояльності, рекомендаційних алгоритмів і маркетингових інструментів, орієнтованих на конкретну аудиторію.

3.3 Оформлення замовлення: кошик, оплата

Процес оформлення замовлення є критично важливим етапом функціонування будь-якої комерційної онлайн-платформи, оскільки саме він завершує цикл взаємодії користувача з товаром та безпосередньо впливає на конверсію і прибутковість ресурсу. У рамках реалізації проєкту FitIQ особлива увага приділялася інтуїтивності, швидкодії та надійності механізмів додавання товарів до кошика, їх перегляду, редагування кількості або видалення, а також подальшій оплаті замовлення.

Для підвищення зручності користування платформою реалізовано функціонал порівняння товарів (рис. 3.12). Цей інструмент дозволяє покупцеві одночасно переглядати характеристики кількох обраних позицій, що спрощує прийняття обґрунтованого рішення перед купівлею.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				54
Змн.	Арк.	№ докум.	Підпис	Дат		

Порівняти

Головна • Порівняти

Товар		
	Футболка для хлопчика з принтом	Футболка з принтом
Опис	Lorem ipsum dolor sit amet consectetur adipisicing elit. Ad, distinctio.	Lorem ipsum dolor sit amet consectetur adipisicing elit. Ad, distinctio.
Ціна	275.00₴	195.00₴
Додати до кошика	<button>Додати до кошика</button>	<button>Додати до кошика</button>
Видалити		

Рис. 3.12 – Сторінка порівняння товарів

Порівняльна таблиця дозволяє користувачам зручно аналізувати відмінності між схожими товарами за основними параметрами (назва, опис, ціна, наявність тощо). Такий підхід сприяє вдосконаленню користувацького досвіду та підвищує ймовірність успішного завершення покупки.

Ще одним важливим елементом інтерфейсу є сторінка списку бажань (рис. 3.13), що дозволяє зберігати вподобані товари для подальшого розгляду або купівлі. Це особливо корисно для відкладених покупок.

Список бажань

Головна • Список бажань

Товар	Ціна	Кількість	Дія
 Жіноча блуза	275.00₴	1	<button>Додати до кошику</button>  Видалити
 Жіноча сукня в клітинку Whitetails	780.00₴	0	<button>Додати до кошику</button>  Видалити
 Однотонна сорочка	225.00₴	0	<button>Додати до кошику</button>  Видалити
<button>Перейти до кошика</button>			

Рис. 3.13 – Список бажань користувача

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		55

Наявність списку бажань позитивно впливає на залучення клієнта та підвищує ймовірність повторного повернення до платформи. Користувачі мають можливість зберігати товари, які їх зацікавили, та легко додавати їх до кошика в зручний момент.

Сторінка кошика (рис. 3.14) є важливим етапом процесу оформлення замовлення. Вона дає змогу користувачеві переглянути обрані товари, змінити їхню кількість або видалити позиції перед переходом до оформлення. Також тут реалізовано вибір типу доставки та динамічний підрахунок загальної вартості покупки.

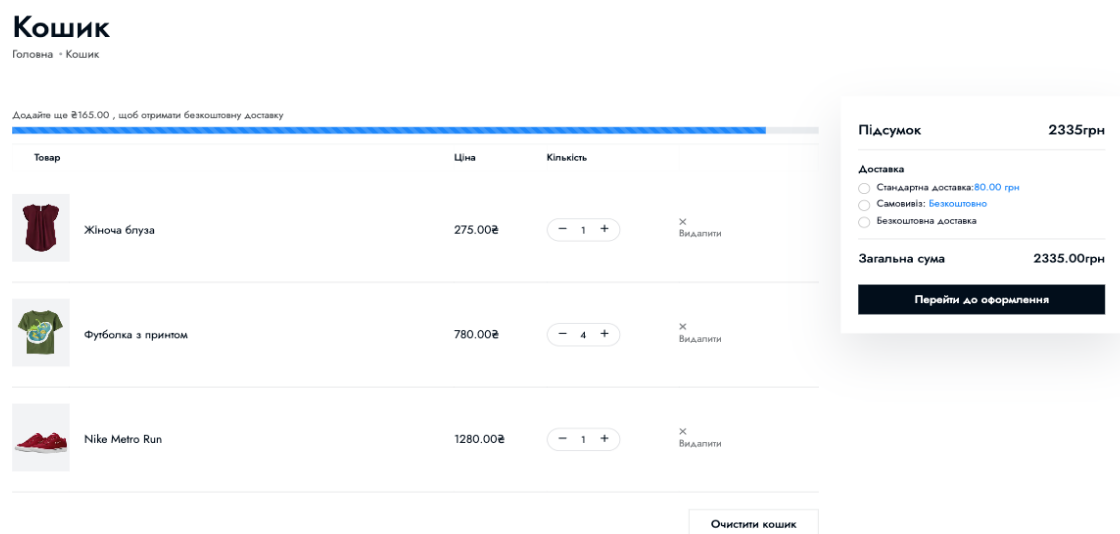


Рис. 3.14 – Інтерфейс сторінки кошика

Інтерфейс кошика надає користувачеві повний контроль над фінальним замовленням, дозволяючи здійснювати гнучке редагування замовлення та миттєво оцінювати вартість з урахуванням обраної доставки.

Після переходу до етапу оформлення замовлення користувач заповнює деталі доставки, обирає тип доставки та переглядає фінальний підсумок замовлення. Форма (рис. 3.15) містить поля для імені, прізвища, країни, адреси, міста, поштового індексу, номера телефону та електронної пошти. Праворуч представлений блок з переліком товарів, підрахунком суми та кнопкою підтвердження замовлення.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		56

Деталі замовлення

Ім'я *

Андрій

Прізвище *

Конвісар

Країна *

Україна

Адреса

Бориса Тена 19, кв 85

Місто

Житомир

Поштовий індекс

10003

Телефон *

0637654321

Електронна пошта*

test@gmail.com

Нотатки до замовлення (необов'язково)

Коментар до замовлення або доставки.

Ваше замовлення

Товар	Сума
Жіноча блуза x 1	275.00₴
Футболка з принтом x 4	195.00₴
Nike Metro Run x 1	1280.00₴
Доставка	Кур'єром сьогодні: 60.00₴ ☒ Доставка протягом 7 днів: 20.00₴ ☐
Проміжна сума	2335.00₴
Вартість доставки	60.00₴
Знижка	0.00₴
Загальна сума	2395.00₴
☒ Банківська картка visa 4242 4242 4242 4242 05 / 44 233 10003	
☐ Оплата при отриманні	

Оформити замовлення

Рис. 3.15 – Форма оформлення замовлення з деталями доставки та підсумком

Форма оформлення замовлення реалізована у зручному та логічному вигляді, що спрощує процес взаємодії користувача з платформою. Завдяки чітко структурованим полям та інтерактивному підсумку, користувач може впевнено завершити покупку, що позитивно впливає на рівень задоволеності клієнта та конверсію платформи.

Після підтвердження користувачем оформлення покупки, дані про замовлення автоматично фіксуються у базі даних (рис. 3.16). Це дозволяє не лише здійснювати подальшу обробку покупки (доставка, звітність, аналітика), а й формувати історію замовлень у профілі клієнта.

```

{
  _id: ObjectId('685e372cb555d2dc50242eb8')
  user: ObjectId('685d57dc01fa46e975cf997c')
  cart: Array (3)
    name: "Андрій Конвісар"
    address: "Бориса Тена 19, кв 85"
    email: "test@gmail.com"
    contact: "0637654321"
    city: "Житомир"
    country: "Україна"
    zipCode: "10003"
    subTotal: 2335
    shippingCost: 60
    discount: 0
    totalAmount: 2395
    shippingOption: "on"
  cardInfo: Object
  paymentIntent: Object
    paymentMethod: "Card"
    orderNote: ""
    status: "pending"
    createdAt: 2025-06-27T06:16:12.069+00:00
    updatedAt: 2025-06-27T06:16:12.069+00:00
    invoice: 1011
  __v: 0
}

```

Рис. 3.16 – Збереження даних про замовлення в базі даних

Формування запису в базі даних після здійснення замовлення є ключовим етапом у роботі платформи. У структурі збережених даних передбачені всі необхідні поля: інформація про користувача, товари, способи оплати, доставка, дата та статус. Такий підхід забезпечує надійність, прозорість та ефективність управління процесами електронної комерції на платформі.

Таким чином, у рамках реалізації функціоналу оформлення замовлення на платформі було впроваджено повноцінний процес покупки, що охоплює всі ключові етапи: формування кошика, додавання та видалення товарів, вибір способу доставки, оформлення даних покупця та підтвердження замовлення. Важливою перевагою є можливість попереднього порівняння товарів і додавання їх до списку бажань, що сприяє прийняттю зваженого рішення перед покупкою.

Завдяки інтеграції системи обробки замовлень дані кожної покупки надійно зберігаються в базі даних, включаючи повну інформацію про покупця, обрані товари, способи доставки та оплати. Це забезпечує простежуваність, контроль і подальший аналіз транзакцій.

3.4 Тестування платформи

У процесі розробки сучасної вебплатформи, особливо у сфері електронної комерції, важливим етапом є комплексне тестування як клієнтської, так і серверної частини системи. Основна мета тестування полягає у виявленні можливих вузьких місць продуктивності, затримок у завантаженні контенту, некоректної побудови структури сторінок або неоптимізованого рендерингу ресурсів. З огляду на це, для оцінки продуктивності, ефективності завантаження та структурної цілісності платформи FitIQ було обрано інструмент GTMetrix.

GTMetrix є одним із найбільш популярних сервісів для аналізу швидкодії вебзастосунків, оскільки він поєднує в собі технології Lighthouse і Web Vitals, а також дозволяє детально аналізувати процеси завантаження ресурсів за допомогою Waterfall-графіка. Платформа надає об'єктивну оцінку ключових

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		58

метрик продуктивності, таких як Largest Contentful Paint (LCP), Time to Interactive (TTI), Total Blocking Time (TBT) та Cumulative Layout Shift (CLS), що є критично важливими показниками в контексті UX та SEO-оптимізації.

Крім того, GTMetrix дозволяє виконувати структурний аудит HTML/CSS/JS-розмітки, визначати ресурси, які блокують відображення вмісту, а також аналізувати використання зовнішніх бібліотек, кешування, оптимізацію шрифтів та зображень. Таким чином, використання GTMetrix є цілком обґрунтованим вибором, що дозволяє отримати всебічну картину як технічного стану платформи, так і зручності взаємодії з нею кінцевого користувача.

Рис. 3.17 демонструє результати аудиту структури сторінки вебплатформи FitIQ, виконаного за допомогою сервісу GTMetrix. Цей аудит дає змогу оцінити ступінь оптимізації HTML-, CSS- та JavaScript-компонентів сторінки, а також ефективність завантаження основних ресурсів. Показники, отримані в результаті цього аудиту, дозволяють розробникам виявити потенційні проблеми й впровадити зміни для покращення продуктивності платформи.

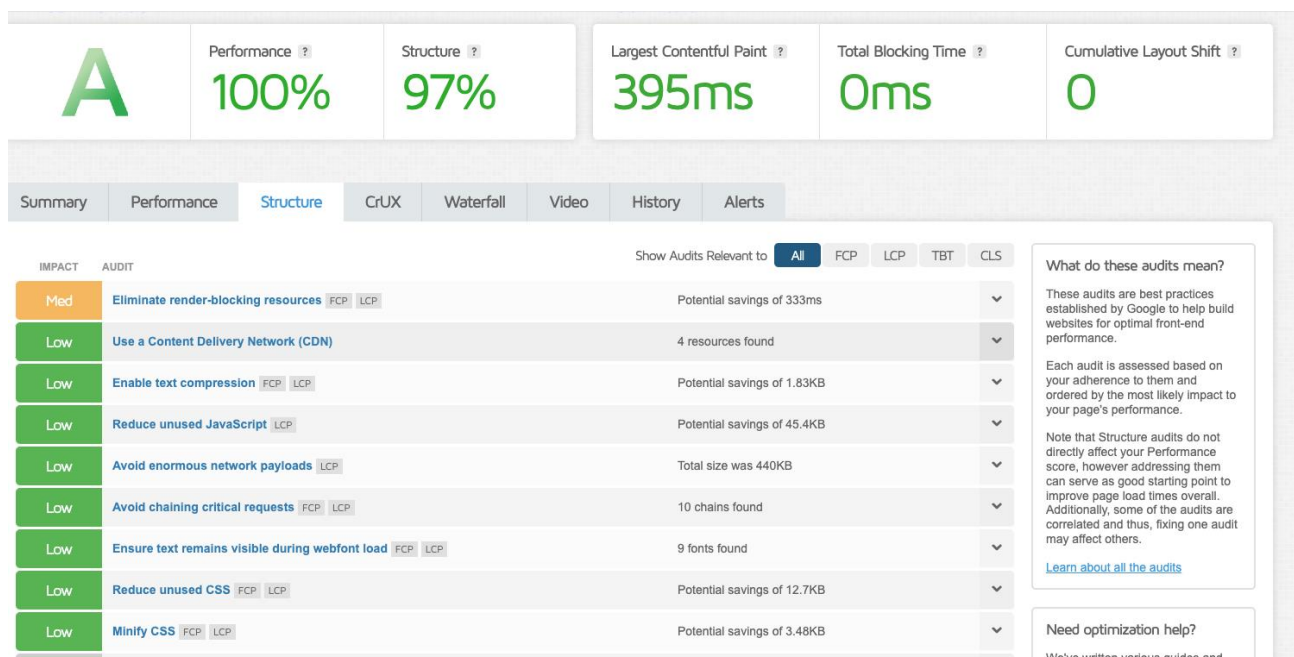


Рис. 3.17 – Аудит структури сторінки за результатами GTMetrix

Структура сторінки платформи FitIQ демонструє високий рівень оптимізації: загальний показник структури становить 97%, а більшість

виявлених проблем мають низький рівень впливу. Єдиним пунктом із середнім пріоритетом є необхідність усунення блокуючих ресурсів рендерингу, що може бути оптимізовано шляхом застосування атрибутів `async` або `defer` до скриптів. Інші рекомендації, як-от використання CDN, зменшення розміру JavaScript або CSS, вже реалізовані або мають незначний вплив на загальну продуктивність. Це свідчить про високу якість реалізації структури сторінки та відповідність платформи сучасним вимогам веброзробки.

Рис. 3.18 ілюструє графік завантаження ресурсів (Waterfall Chart), який є важливим інструментом для аналізу продуктивності вебсторінки. Цей графік наочно демонструє, як саме браузер завантажує ресурси вебсайту FitIQ, починаючи від початкового запиту до HTML-документа і завершуючи завантаженням зображень, скриптів, стилів та шрифтів. Кожен запит візуалізується як окремий рядок із часовими показниками: DNS-запит, встановлення з'єднання, отримання відповіді та обробка. Такий підхід дозволяє оцінити послідовність, час завантаження та пріоритетність ресурсів, виявити вузькі місця, що можуть уповільнювати рендеринг сторінки.

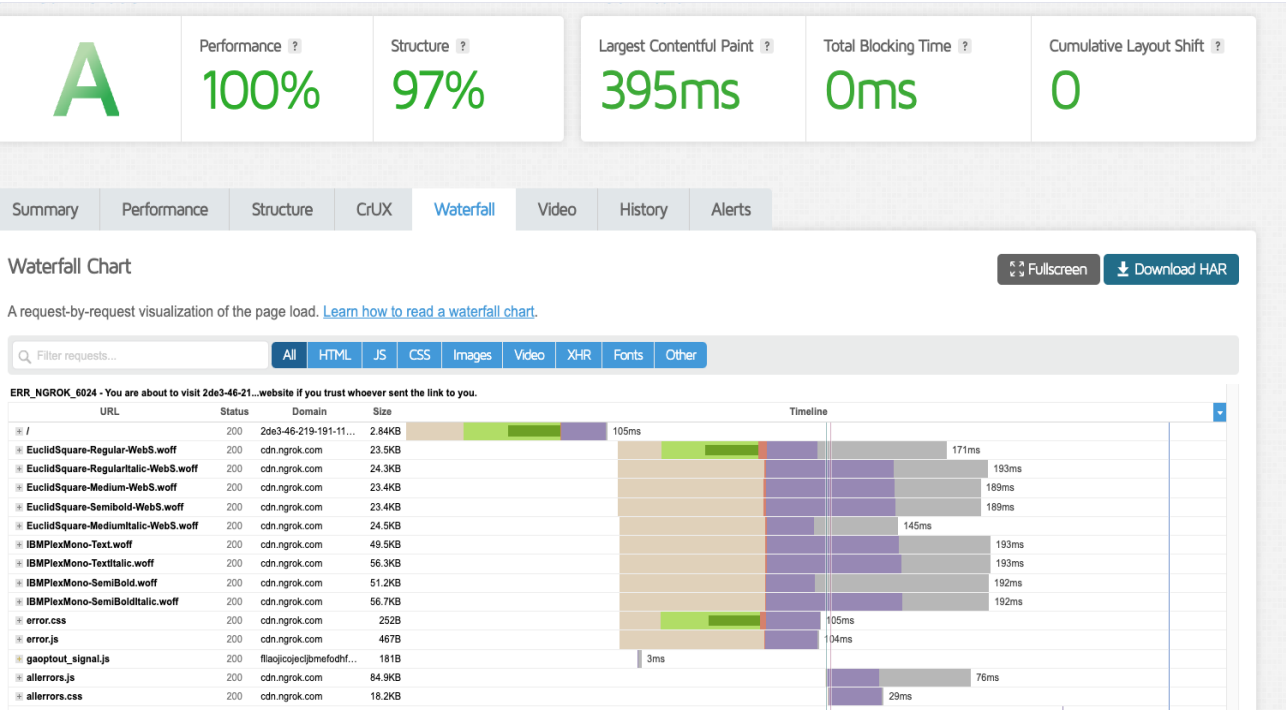


Рис. 3.18 – Графік завантаження ресурсів (Waterfall Chart)

Згідно з наведеним графіком Waterfall, завантаження ресурсоємних елементів платформи FitIQ відбувається синхронно, без затримок та блокувань, що забезпечує відмінні показники продуктивності. Усі ключові ресурси, зокрема шрифти, стилі та JavaScript-файли, обробляються з мінімальною затримкою і не спричиняють блокування візуального контенту. Це свідчить про високий рівень оптимізації фронтенду, правильне використання HTTP-запитів, кешування та асинхронного завантаження. У підсумку, така структура значно покращує UX (досвід користувача), скорочуючи час першого рендеру та підвищуючи швидкість взаємодії з платформою.

Рис. 3.19 демонструє результати оцінки продуктивності вебплатформи FitIQ на основі інструменту Lighthouse, інтегрованого в систему GTMetrix. Lighthouse – це сучасний автоматизований засіб аудиту вебсторінок, який вимірює низку ключових показників, що впливають на реальний користувацький досвід. Ці метрики дозволяють оцінити швидкість завантаження, плавність інтерфейсу, реактивність та стабільність макету під час візуалізації сторінки.

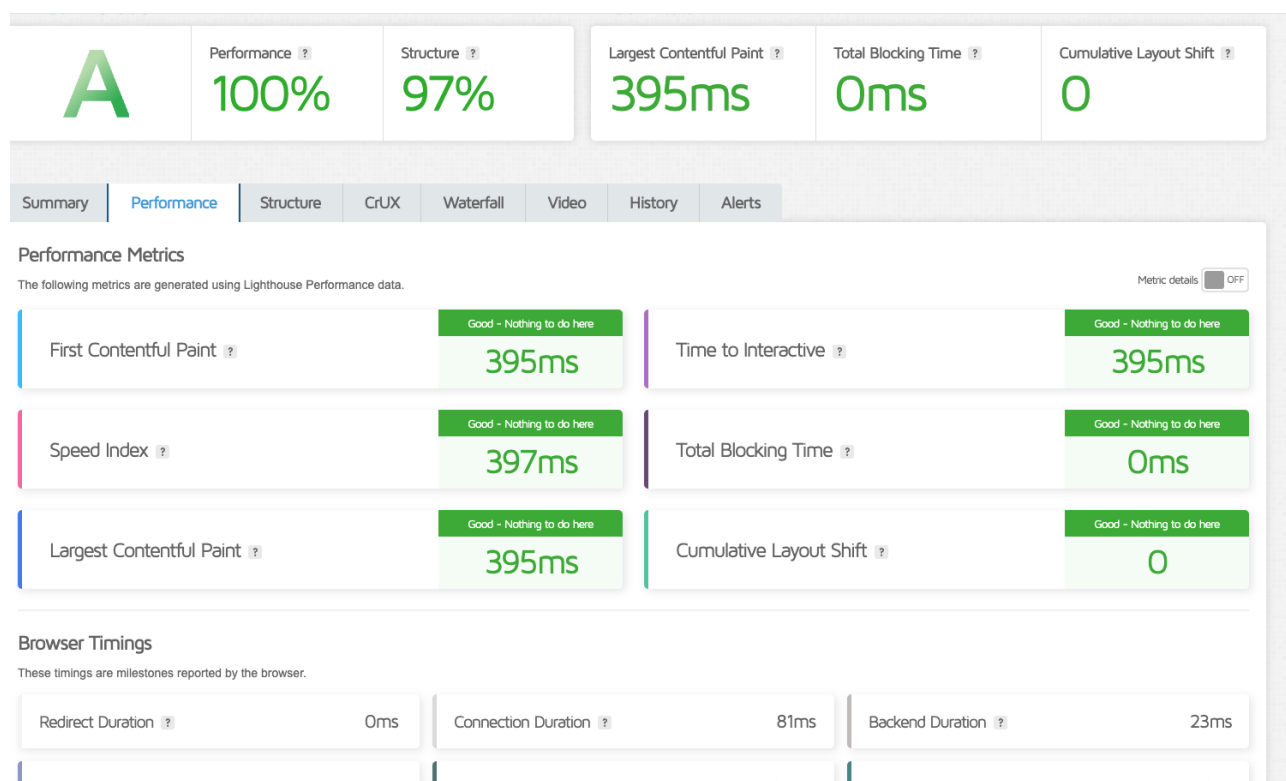


Рис. 3.19 – Оцінка продуктивності на основі Lighthouse в GTMetrix

Аналіз результатів Lighthouse показує, що платформа FitIQ демонструє виключно високий рівень продуктивності. Зокрема, показники First Contentful Paint (395 мс), Largest Contentful Paint (395 мс) та Time to Interactive (395 мс) свідчать про надзвичайно швидку готовність інтерфейсу до взаємодії, що є критично важливим для користувацького досвіду. Відсутність блокувань (Total Blocking Time = 0 мс) та повна стабільність візуального макету (Cumulative Layout Shift = 0) вказують на ефективне використання ресурсів і дотримання принципів сучасної веброзробки. Ці результати підтверджують, що вебплатформа оптимізована відповідно до кращих практик Lighthouse і готова до роботи з великою кількістю одночасних користувачів без втрати продуктивності.

Проведено аудит платформи за допомогою розширення Lighthouse у браузері Chrome (рис. 3.20).

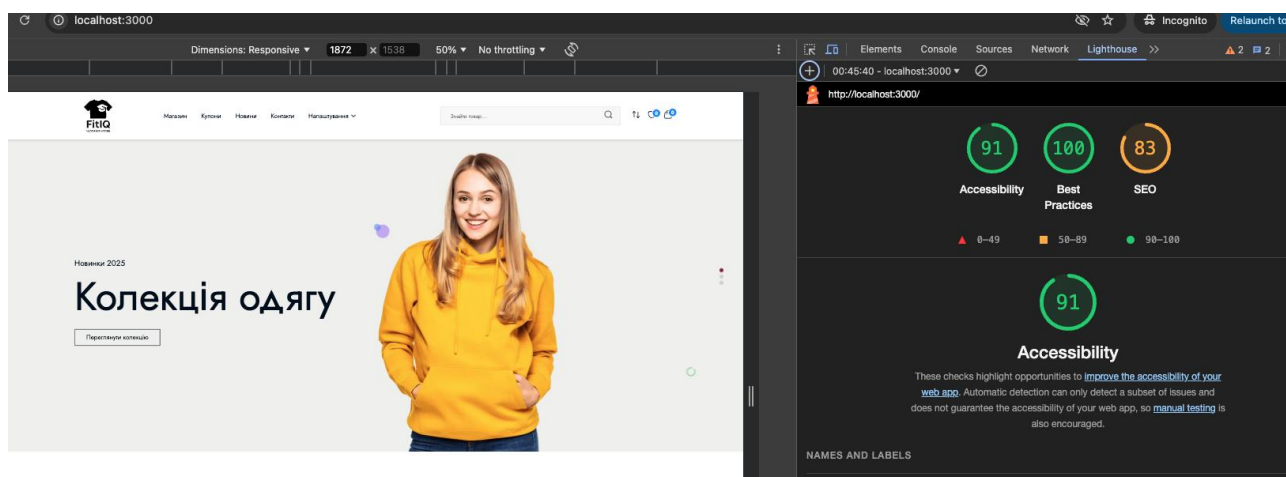


Рис. 3.20 – Аудит платформи FitIQ за допомогою Lighthouse

Згідно з результатами Lighthouse, вебплатформа FitIQ демонструє високі показники за більшістю критеріїв: Best Practices – 100, Accessibility – 91, SEO – 83. Така оцінка свідчить про якісну реалізацію інтерфейсу, дотримання стандартів безпеки, коректне використання тегів HTML, доступність елементів керування і коректне заповнення метаданих. У підсумку, аудит підтверджує, що FitIQ відповідає вимогам сучасних вебстандартів і забезпечує стабільну, доступну та ефективну взаємодію з користувачами.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		62

У процесі тестування вебплатформи FitIQ було здійснено комплексну оцінку її продуктивності, структури та зручності використання за допомогою сучасного аналітичного інструменту – GTMetrix, а також локального аудиту Lighthouse. Ці інструменти дозволили виявити як технічні характеристики завантаження сторінки, так і потенційні вузькі місця, які можуть вплинути на взаємодію користувача із системою.

У межах тестування платформи було використано онлайн-інструмент GTMetrix, що дозволив здійснити комплексну оцінку продуктивності, структури та швидкодії вебзастосунку. Даний сервіс надає об'єктивні метрики щодо часу завантаження сторінки, взаємодії користувача з інтерфейсом та ефективності використання ресурсів.

Результати тестування підтвердили високий рівень оптимізації платформи: усі ключові показники продуктивності набрали найвищі бали, а виявлені рекомендації мають незначний вплив і не потребують термінового втручання. Це свідчить про те, що розроблена система є стабільною, швидкою у роботі та технічно готовою до реального використання.

Загалом, тестування платформи за допомогою GTMetrix дозволило переконатися у відповідності функціоналу сучасним вебстандартам і забезпечити високу якість користувацького досвіду. Окремо було проведено локальне тестування за допомогою Lighthouse, яке також продемонструвало високі оцінки: Accessibility – 91, Best Practices – 100, SEO – 83. Це підтверджує, що інтерфейс відповідає стандартам доступності, використовує безпечні та перевірені практики, а контент добре оптимізований для пошукових систем.

Загалом, результати тестування засвідчили, що вебплатформа FitIQ має високу продуктивність, оптимізовану структуру та доступний користувацький інтерфейс. Всі функціональні елементи працюють коректно, що забезпечує стабільність, масштабованість та ефективність взаємодії з кінцевим користувачем.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		63

Висновки до третього розділу.

У третьому розділі було розглянуто практичні аспекти реалізації користувацького інтерфейсу та функціоналу вебплатформи з продажу одягу. Проаналізовано структуру та адаптивність верстки, що забезпечує коректне відображення контенту на різних типах пристроїв. Реалізовано особистий кабінет користувача, де передбачено функції реєстрації, авторизації та перегляду персоналізованої інформації.

Окрему увагу приділено модулю оформлення замовлення, включно з додаванням товарів до кошика, заповненням платіжних даних та підтвердженням покупки. Проведено тестування основних функціональних елементів інтерфейсу, що дозволило виявити та усунути типові помилки в логіці взаємодії користувача з системою.

Таким чином, реалізований інтерфейс платформи є інтуїтивно зрозумілим, функціонально повним та готовим до практичного використання для забезпечення зручного онлайн-шопінгу.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		64

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було спроектовано та реалізовано функціональний вебдодаток інтернет-магазину одягу з інтегрованою системою персоналізованих рекомендацій, що відповідає сучасним тенденціям розвитку e-commerce платформ. Розроблена система охоплює повний цикл взаємодії користувача: від перегляду товарів до формування замовлення й оплати, враховуючи індивідуальні вподобання покупців.

Основу архітектури платформи становить стек сучасних вебтехнологій: Next.js, SCSS-модулі та база даних MongoDB, що забезпечило високу продуктивність, масштабованість і підтримку серверного рендерингу. Для інтеграції рекомендаційної системи використано векторні представлення даних (ембедінги), які формуються на основі поведінки користувача, зокрема переглядів і доданих у кошик товарів.

Реалізовано ключові компоненти системи: каталог, кошик, сторінку оформлення замовлення, особистий кабінет із можливістю редагування даних, авторизацію, підписку на новинки, інтеграцію з платіжною системою Stripe.

Особливу увагу приділено оптимізації інтерфейсу: проведено тестування продуктивності за допомогою GTMetrix, результати якого засвідчили високий рівень завантаження та зручності використання застосунку на різних пристроях.

Таким чином, розроблений інтернет-магазин є сучасною, технічно вдосконаленою платформою, яка може бути масштабована, доповнена новими функціями – зокрема, вдосконаленням системи рекомендацій, аналітикою користувацьких дій та маркетинговими інструментами – для забезпечення конкурентоспроможності на ринку онлайн-торгівлі.

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		65

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Алекс Маккоу. Веб-застосунки на JavaScript, 2012. – 283 с.
2. Березін О.Є. Комп'ютерні мережі: захист інформації: навчальний посібник / О.Є. Березін, Л.Є. Гордієнко, Є.О. Іванченко. – Київ: Видавництво “Кондор”, 2017. – 352 с.
3. Бондаренко С.П. Інформаційні системи і технології в економіці: навч. посіб / С.П. Бондаренко, В.І. Коваль. К.: Центр учбової літератури, 2021. – 352 с.
4. Борисенко В.О. Захист інформації на комп'ютерах і мережах: підручник / В.О. Борисенко, О.І. Бондаренко, І.О. Железняк. Київ: Видавництво “Наукова думка”, 2019 – 416 с.
5. Гайна Г.А. Організація баз даних і знань. Мови баз даних: Конспект лекцій / Г.А. Гайна. – Київ: КНУБА, 2002. – 64 с.
6. Гайна Г.А. Організація баз даних і знань. Організація реляційних баз даних: Конспект лекцій / Г.А. Гайна, Н.Л. Попович; К.: КНУБА, 2000. – 76с.
7. Грінченко С.В. Методи та засоби захисту інформації в комп'ютерних системах: навчальний посібник / С.В. Грінченко, Ю.В. Мартинюк, С.О. Щербак. – Київ: Видавничий центр КНЕУ, 2018. – 142 с.
8. Гурман Д. JavaScript. Біблія користувача. 2002. – 958 с.
9. Документація MongoDB [Електронний ресурс] – Режим доступу: <https://www.mongodb.com/docs>
10. Документація Next.js [Електронний ресурс] – Режим доступу: <https://nextjs.org/docs>
11. Сидоренко М.В. Проєктування інформаційних систем: навч. посіб. / М.В. Сидоренко. – Харків: ФОП Бровін О.В., 2020. – 295 с.
12. Ткаченко А.А. Практика розробки Full-Stack додатків на JavaScript / А.А. Ткаченко. – Харків: СумДУ, 2023. – 216 с.
13. Шилова Ю.А. Захист інформації в комп'ютерних системах: підручник /

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		66

Ю.А. Шилова. Київ: Видавничо-поліграфічний центр “Київський університет”, –2016. – 492 с.

14. Що таке Node JS. [Електронний ресурс] – Режим доступу: <https://netology.ru/blog/node>

15. Ярощук І.О. Розробка вебзастосунків з використанням фреймворку Next.js. / І.О. Ярощук // Вісник НТУУ «КПІ». Інформатика та обчислювальна техніка. – 2022. – № 68. – С. 47-53.

16. Andrew Mead The Complete React Developer Course (Hooks and Redux) / М. Andrew. Udemy, 2020.

17. ASOS. Офіційний сайт. [Electronic resource]. – Mode of access: https://www.asos.com/client_side_language

18. Clientsidetechnologies. [Electronic resource]. – Mode of access: <https://w3techs.com/technologies/overview/>

19. David Herron. “Node.js Web Development: Server-side development with Node 10 made easy”/ Herron David. Packt Publishing, 2018.

20. Ganore P. What Is A Web Server And How Does It Function? / Ganore Pravin. 2017. [Electronic resource]. – Mode of access: <https://www.milesweb.com/blog/hosting/web-server-function/>

21. HTML&CSS. [Electronic resource]. – Mode of access: <https://www.w3.org/standards/webdesign/htmlcss>

22. ISO/IEC 25010:2011. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.

23. Kasta.ua. Офіційний сайт. [Electronic resource]. – Mode of access: <https://kasta.ua>

24. Krug S. Don't Make Me Think: A Common Sense Approach to Web Usability/ S. Krug. 3rd ed. New Riders, 2014.

25. MDN Web Docs. JavaScript Guid. [Electronic resource]. – Mode of access: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Нерода С.І.				67
Змн.	Арк.	№ докум.	Підпис	Дат		

26. MODIVO Україна. Офіційний сайт. [Electronic resource]. – Mode of access: <https://www.modivo.ua>
27. MongoDB. [Electronic resource]. – Mode of access: <https://proselyte.net/tutorials/mongodb/introduction/>
28. Most popular JS stacks. [Electronic resource]. – Mode of access: <https://medium.com/datadriveninvestor/most-popular-technology-stack-to-choose-from-full-stack-vs-mean-stack-vs-mern-stack-in-2019-d12c0a17439a>
29. Oba, J. (2024, September 17). Building smart recommendation systems with Node.js and Google Gemini. Afrodev [Electronic resource] / J. Oba. – Mode of access: <https://afrodev.space/building-smart-recommendation-system-with-embeddings>
30. Sinsay Україна. [Electronic resource]. – Mode of access: <https://www.sinsay.com/ua/uk>
31. Stripe API Documentation [Electronic resource]. – Mode of access: <https://stripe.com/docs/api>
32. Stripe. Checkout Integration Guide. Stripe, 2024.
33. Trends in JS development. [Electronic resource]. – Mode of access: <https://trends.builtwith.com/docinfo/Javascript>
34. Web Server and its Types of Attacks 2012. [Electronic resource]. – Mode of access: <https://www.greycampus.com/open-campus/ethical-hacking/web-server-and-its-types-of-attacks>.

ДОДАТКИ

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		69

Порівняльна таблиця функціональності популярних платформ онлайн-торгівлі
одягом: ASOS, Kasta.ua, Sinsay, MODIVO

Функція / Платформа	ASOS	<u>Kasta.ua</u>	Sinsay	MODIVO
Локалізація для України	Не підтримується	Повна локалізація	Підтримується	Підтримується
Мова інтерфейсу	Англійська	Українська	Українська	Українська
Ціни у гривнях	У валюті GBP/USD	Так	Так	Так
Онлайн-оплата	Так	Так	Так	Так
Мобільний застосунок	Так	Так	Так	Так
Сегментація товарів	Жінки, чоловіки, аксесуари	Жінки, чоловіки, діти, техніка	Жінки, чоловіки, діти, дім	Жінки, чоловіки, діти, бренди
Система рекомендацій	Потужна, базується на поведінці	Базова, але наявна	Обмежена	Середня (на основі переглядів)
Акції та промо-пропозиції	Часті акції, банери	Промо, персоналізовані добірки	Акції на головній сторінці	Акції, сезонні розпродажі
Фільтрація товарів	Розширена	Розширена	Базова	Розширена
Збереження улюбленого	Так	Так	Так	Так
Адаптивність до мобільних	Так	Так	Так	Так
Особистий кабінет	Так	Так	Так	Так
Доставка по Україні	Ні	Так	Так	Так
Міжнародна доставка	Так	Ні	Ні	Ні
Пошук за категоріями	Так	Так	Так	Так
Пошук за ключовими словами	Так	Так	Так	Так
Зв'язок з користувачем	Push, email	Push, email	Push	Push, email
Оцінки і відгуки	Так	Так	Ні	Так
Повернення товару	Так	Так	Так	Так
Рівень дизайну (UI/UX)	Високий	Середній	Високий	Високий

Джерело: створено автором на основі аналізу створено автором на основі [18, 24, 27, 31]

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Нерода С.І.				70
Змн.	Арк.	№ докум.	Підпис	Дат		

Реалізація API-ендпоінту для отримання персоналізованих рекомендацій
користувачеві на основі ембедінгів

```

exports.getPersonalizedRecommendations = async (req, res) => {
  try {
    const { userId } = req.params;

    const user = await User.findById(userId);
    if (!user || !user.viewedProducts || !user.viewedProducts.length) {
      return res.status(200).json([]);
    }

    const viewed = await Product.find({
      _id: { $in: user.viewedProducts },
      embedding: { $exists: true },
    });

    const embeddings = viewed.map((p) => p.embedding);
    if (!embeddings.length) return res.status(200).json([]);

    const vectorSize = embeddings[0].length;
    const profileVector = Array(vectorSize).fill(0);

    embeddings.forEach((vec) => {
      for (let i = 0; i < vectorSize; i++) {
        profileVector[i] += vec[i];
      }
    });

    for (let i = 0; i < vectorSize; i++) {
      profileVector[i] /= embeddings.length;
    }

    const demographicFilter = {};

    if (user.gender) {
      demographicFilter.targetAudience = { $in: [user.gender, "unisex"] };
    }

    if (user.age) {
      demographicFilter.ageRange = {

```

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		71

```

    $elemMatch: {
      min: { $lte: user.age },
      max: { $gte: user.age },
    },
  };
}

const allProducts = await Product.find({
  _id: { $nin: user.viewedProducts },
  embedding: { $exists: true },
  ...demographicFilter,
});

const recommendations = allProducts.map((product) => {
  const score = cosineSimilarity(profileVector, product.embedding);
  return { product, score };
});

const top = recommendations
  .sort((a, b) => b.score - a.score)
  .slice(0, 5)
  .map((r) => r.product);

res.status(200).json(top);
} catch (err) {
  console.error("Personalized recommendation error:", err.message);
  res
    .status(500)
    .json({ message: "Failed to generate personalized recommendations" });
}
};

```

		Конвісар А.С.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Нерода С.І.				
Змн.	Арк.	№ докум.	Підпис	Дат		72