

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 «магістр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Розробка системи електронної комерції із
застосуванням алгоритмів персоналізованих
рекомендацій»**

Виконав студент групи ПЗм-24-1
ЯСЕН Антон Євгенійович

Керівник роботи:
ГОДЛЕВСЬКИЙ Юрій Олександрович

Рецензент:
ЄФРЕМОВ Юрій Миколайович

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
«18» вересня 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: ЯСЕН Антон Євгенійович

Керівник роботи: ГОДЛЕВСЬКИЙ Юрій Олександрович

Тема роботи: **«Розробка системи електронної комерції із застосуванням алгоритмів персоналізованих рекомендацій»**,

затверджена Наказом закладу вищої освіти від **«18» вересня 2025 р. № 491/с**

Вихідні дані для роботи: **Процес функціонування систем електронної комерції, що забезпечують взаємодію користувача з онлайн-магазином, каталогом товарів, механізмами перегляду, вибору та покупки. Алгоритми персоналізованих рекомендацій та способи їх інтеграцій у веборієнтовану систему електронної комерції для підвищення ефективності взаємодії користувачів із товарним каталогом.**

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

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Годлевський Ю.О.	22.09.2025р.	22.09.2025р.
2	Годлевський Ю.О.	22.09.2025р.	22.09.2025р.
3	Годлевський Ю.О.	22.09.2025р.	22.09.2025р.

РЕФЕРАТ

Випускна кваліфікаційна робота ступеня «магістр» присвячена розробці системи електронної комерції з інтегрованим модулем персоналізованих рекомендацій товарів. Пояснювальна записка до випускної роботи містить текстову частину, викладену на 59 сторінках друкованого тексту, має 8 сторінок додатків. Список використаних джерел містить 34 найменувань і займає 4 сторінки. В роботі наведено 23 рисунків та 1 таблицю. Загальний обсяг роботи – 68 сторінок.

Метою роботи є проектування та реалізація системи електронної комерції з використанням алгоритмів персоналізованих рекомендацій, орієнтованих на аналіз поведінки користувачів, з метою підвищення релевантності товарних пропозицій, покращення користувацького досвіду та збільшення ефективності взаємодії з платформою.

У роботі описано структуру зберігання даних, орієнтовану на накопичення поведінкових подій користувачів, реалізацію рекомендаційної логіки без використання зовнішніх ML-платформ та інтеграцію рекомендацій у ключові елементи веб-застосунку. Проведено аналіз результатів роботи системи в умовах реального використання, що підтвердив адаптивність рекомендаційного модуля, його стабільну роботу за різних сценаріїв взаємодії та зменшення впливу проблеми холодного старту.

КЛЮЧОВІ СЛОВА: ЕЛЕКТРОННА КОМЕРЦІЯ, РЕКОМЕНДАЦІЙНІ СИСТЕМИ, ПЕРСОНАЛІЗАЦІЯ, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, ІТЕМ-ТО-ІТЕМ, ПОВЕДІНКОВІ ДАНІ, ВЕБ-ЗАСТОСУНОК, NEXT.JS, NODE.JS, MONGODB.

ABSTRACT

The final qualification work for the Master's degree is devoted to the development of an e-commerce system with an integrated module of personalized product recommendations. The explanatory note to the final work contains a text part, set out on 59 pages of printed text, has 8 pages of appendices. The list of sources used contains 34 names and occupies 4 pages. The work contains 23 figures and 1 table. The total volume of the work is 68 pages.

The purpose of the work is to design and implement an e-commerce system using personalized recommendation algorithms focused on analyzing user behavior, in order to increase the relevance of product offers, improve user experience and increase the efficiency of interaction with the platform.

The work describes a data storage structure focused on accumulating user behavioral events, implementing recommendation logic without using external ML platforms and integrating recommendations into key elements of the web application. An analysis of the system's performance results in real-world conditions was conducted, which confirmed the adaptability of the recommendation module, its stable operation under various interaction scenarios, and reduced the impact of the cold start problem.

KEYWORDS: E-COMMERCE, RECOMMENDATION SYSTEMS, PERSONALIZATION, COLLABORATIVE FILTERING, ITEM-TO-ITEM, WEB APPLICATION, NEXT.JS, NODE.JS, MONGODB.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ.....	10
1.1 АНАЛІЗ СУЧАСНОГО СТАНУ ТА ТЕНДЕНЦІЙ РОЗВИТКУ ЕЛЕКТРОННОЇ КОМЕРЦІЇ	10
1.2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ФОРМУВАННЯ ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ У СИСТЕМАХ ЕЛЕКТРОННОЇ КОМЕРЦІЇ	13
1.3 АНАЛІЗ НЕДОЛІКІВ СУЧАСНИХ РЕКОМЕНДАЦІЙНИХ СИСТЕМ ТА ВИКЛИКИ ЇХ ВПРОВАДЖЕННЯ.....	18
1.4 ОБҐРУНТУВАННЯ ВИБОРУ ГІБРИДНОГО ПІДХОДУ ТА НЕОБХІДНІСТЬ РОЗРОБКИ ВЛАСНОЇ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ.....	19
Висновки до розділу 1.....	21
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ІЗ ЗАСТОСУВАННЯМ АЛГОРИТМІВ ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ.....	24
2.1 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПОБУДОВИ СИСТЕМИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ІЗ АЛГОРИТМАМИ ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ.....	24
2.2 ПРОЄКТУВАННЯ СТРУКТУРИ ДАНИХ ТА ЗБЕРІГАННЯ ПОВЕДІНКОВИХ ПОДІЙ КОРИСТУВАЧІВ.....	26
2.3 АЛГОРИТМИ, СХЕМИ ТА ДІАГРАМИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	28
2.4 УЗАГАЛЬНЕННЯ ТА ОБҐРУНТУВАННЯ АРХІТЕКТУРНИХ ТА ПРОЄКТНИХ РІШЕНЬ СИСТЕМИ.	33
Висновки до розділу 2.....	35

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЙОГО	
ТЕСТУВАННЯ	37
3.1 РЕАЛІЗАЦІЯ ОСНОВНИХ КОМПОНЕНТІВ СИСТЕМИ.....	37
3.2 СТРУКТУРА ІНТЕРФЕЙСУ ВЕБ-ДОДАТКУ	41
3.3 АНАЛІЗ РОБОТИ СИСТЕМИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ З ІНТЕГРУВАННЯМ	
АЛГОРИТМІВ РЕКОМЕНДАЦІЇ.....	46
3.4 ТЕСТУВАННЯ ТА ПЕРЕВІРКА КОРЕКТНОСТІ РОБОТИ РЕКОМЕНДАЦІЙНОГО	
МОДУЛЯ.....	49
Висновки до розділу 3.....	52
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ДОДАТКИ.....	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

E-commerce – (англ. electronic commerce) – електронна комерція

CF – (англ. collaborative filtering) – колаборативна фільтрація

Item-to-item CF – (англ. item-to-item collaborative filtering) – колаборативна фільтрація на основі схожості товарів

API – (англ. application programming interface) – програмний інтерфейс прикладного програмування

ORM – (англ. object-relational mapping) – об'єктно-реляційне відображення

UI – (англ. user interface) – інтерфейс користувача

UX – (англ. user experience) – користувацький досвід

JSON – (англ. JavaScript Object Notation) – формат обміну даними

NoSQL – (англ. not only SQL) – нереляційна система керування базами даних

CRUD – (англ. create, read, update, delete) – базові операції роботи з даними

KPI – (англ. key performance indicators) – ключові показники ефективності

ML – (англ. machine learning) – машинне навчання

ВСТУП

Актуальність теми. Стрімкий розвиток електронної комерції та зростання конкуренції між онлайн-магазинами вимагають застосування інтелектуальних технологій для покращення взаємодії користувачів із сервісом. Персоналізовані рекомендації стали ключовим елементом сучасних торговельних платформ, оскільки дають змогу підвищити конверсію, збільшити середній чек, скоротити час пошуку товару та покращити задоволеність користувачів.

Більшість користувачів взаємодіють із великими обсягами інформації, тому системи рекомендацій стають основним інструментом навігації в каталозі. Відсутність персоналізації значно зменшує ефективність продажів. У зв'язку з цим актуальною є розробка гнучких, масштабованих рекомендаційних модулів, що враховують поведінку користувача, схожість товарів і загальні тенденції попиту.

Таким чином, створення сучасної системи електронної комерції з інтегрованими алгоритмами рекомендацій є важливим завданням, що відповідає потребам ринку, покращує комерційні показники та забезпечує індивідуальний користувацький досвід.

Отже, розробка системи електронної комерції із застосуванням алгоритмів персоналізованих рекомендацій є актуальним завданням, оскільки вона поєднує.

Мета дослідження є розробка та інтеграція системи персоналізованих рекомендацій у веборієнтовану платформу електронної комерції з метою підвищення ефективності взаємодії користувачів із товарним каталогом та збільшення результативності продажів.

Встановлена мета обумовлює наступні задачі:

- Проаналізувати існуючі системи;
- Дослідити основні методи побудови рекомендацій;
- Спроектувати структуру системи;
- Реалізувати модуль рекомендацій та інтегрувати його у застосунок.

Об'єктом дослідження є процеси функціонування систем електронної комерції, що забезпечують пошук, перегляд, вибір і придбання товарів користувачами в онлайн-середовищі.

Предметом дослідження є алгоритми персоналізованих рекомендацій та методи їх інтеграцій у систему електронної комерції, включно з поведінковими моделями, механізмами фільтрації та гібридними рекомендаційними підходами.

Наукова новизна роботи полягає у використанні динамічного профілю інтересів користувача, який формує рекомендації на основі актуальної історії взаємодій та створенні рекомендаційної системи, що потребує складної математичної моделі або ML-середовища, але забезпечує високу якість результату за рахунок оптимальної логіки.

Практичне значення роботи полягає в тому, що створена система може бути впроваджена у реальні онлайн-магазини незалежно від їх масштабу та дозволяє аналізувати поведінку користувачів завдяки механізму збору подій. Система може слугувати основою для впровадження персоналізованих маркетингових стратегій бо є легкокорозширюваною, модульною та адаптивною для подальшого використання зі складнішими алгоритмами.

Апробація. За темою кваліфікаційної роботи було опубліковано 1 тезу конференції[1].

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1 Аналіз сучасного стану та тенденцій розвитку електронної комерції

Електронна комерція протягом останніх двох десятиліть перетворилася на один із провідних секторів світової економіки. Активне зростання онлайн-торгівлі зумовлене сукупністю чинників, серед яких розвиток цифрових технологій, широке поширення мобільних пристроїв, підвищення швидкості та доступності інтернет-з'єднання, а також поступова зміна споживчої поведінки користувачів. Сучасні споживачі дедалі частіше віддають перевагу онлайн-покупкам завдяки їх зручності, швидкості та можливості порівняння товарів.

За даними аналітичної компанії Statista, обсяг світового ринку електронної комерції у 2023 році перевищив 5,5 трильйона доларів США, при цьому загальна динаміка розвитку галузі залишається стабільно позитивною [2]. Згідно з прогнозами, до 2027 року цей показник може зрости до 8 трильйонів доларів, що свідчить про подальше посилення ролі цифрових каналів у глобальній торгівлі

В Україні електронна комерція також демонструє високі темпи розвитку та входить до числа найбільш динамічних ринків у європейському регіоні. За інформацією Української асоціації електронної комерції (UAEC), кількість онлайн-покупців зростає щороку, незважаючи на складні економічні та соціальні умови [3] . Значний вплив на розвиток галузі мають активне впровадження мобільних фінансових сервісів, поширення безготівкових розрахунків, а також зростання рівня довіри споживачів до українських онлайн-магазинів і маркетплейсів, які за рівнем сервісу та функціональності наближаються до європейських аналогів.



Рис. 1.1 – Динаміка зростання обсягів світової електронної комерції за даними Statista

У міру зростання кількості товарів і користувачів інтернет-магазини дедалі частіше стикаються з проблемою інформаційного перевантаження. За таких умов користувачеві складно швидко знайти товар, який найбільш повно відповідає його потребам та очікуванням. Саме тому персоналізація відіграє ключову роль у сучасних системах електронної комерції, оскільки дозволяє формувати індивідуальний шлях покупця та надавати йому релевантні пропозиції в режимі реального часу.

Провідні світові ритейлери активно застосовують рекомендаційні системи як один із ключових інструментів персоналізації. За оцінками компанії Amazon, близько 35 % усіх покупок здійснюються завдяки рекомендаційним механізмам [4]. Аналогічна тенденція спостерігається і в сфері цифрового контенту: компанія Netflix зазначає, що до 75 % переглядів формуються на основі рекомендаційних алгоритмів [5]. Наведені показники свідчать про те, що персоналізація відіграє визначальну роль у підвищенні рівня продажів та утриманні користувачів.

Впровадження рекомендаційних систем дозволяє значно спростити процес пошуку товарів для користувачів, скоротити час прийняття рішення

про покупку та сформувати індивідуальний досвід взаємодії з вебінтерфейсом. Крім того, такі системи знижують навантаження на традиційні механізми пошуку й сприяють зростанню конверсії повторних покупок. Згідно з дослідженням консалтингової компанії McKinsey, ефективна персоналізація здатна підвищити прибутковість електронної комерції до 30 % у довгостроковій перспективі [6].

Попри численні переваги цифрових технологій, у сфері електронної комерції зберігається низка суттєвих проблем, що обумовлюють необхідність впровадження інтелектуальних систем персоналізації. Однією з ключових проблем є інформаційне перевантаження користувачів у великих товарних каталогах, де самостійний пошук оптимальної пропозиції стає складним і затратним за часом. Це, у свою чергу, негативно впливає на рівень конверсії та призводить до зростання відсотка відмов.

Більшість українських інтернет-магазинів і досі використовують базові механізми сортування товарів, зокрема за датою додавання, популярністю або ціною. Такі підходи є універсальними, проте не враховують індивідуальних уподобань конкретного користувача, що значно обмежує їх ефективність у персоналізованому середовищі.

Окремим викликом є технічна складність реалізації рекомендаційних алгоритмів. Подібні системи повинні обробляти десятки тисяч записів у режимі реального часу, що вимагає оптимізації структури бази даних, налаштування ефективних механізмів кешування та застосування швидкодіючих алгоритмів пошуку схожих товарів.

Поведінка покупця в онлайн-середовищі є багатофакторним процесом, який залежить від психологічних, економічних та візуальних чинників. Вона включає кілька ключових етапів, серед яких увага, інтерес, формування наміру та здійснення дії. Завдання рекомендаційної системи полягає у впливі на кожен із цих етапів, скорочуючи шлях користувача від першого контакту з товаром до здійснення покупки. Згідно з дослідженням Університету

Каліфорнії, коректно підібрані рекомендації можуть скоротити час прийняття рішення покупцем до 40 % [7] .

1.2 Аналіз існуючих методів формування персоналізованих рекомендацій у системах електронної комерції

Ефективність функціонування сучасних платформ електронної комерції значною мірою визначається здатністю системи точно ідентифікувати інтереси користувачів та пропонувати їм релевантні товари. У цьому контексті рекомендаційні алгоритми виступають одним із ключових інструментів персоналізації, оскільки дозволяють зменшити інформаційне перевантаження, підвищити рівень задоволеності покупців і, як наслідок, сприяти зростанню обсягів продажів.

Колаборативна фільтрація (Collaborative Filtering, CF) є одним із найбільш поширених підходів у побудові рекомендаційних систем. Її фундаментальна ідея ґрунтується на припущенні, що користувачі з подібними вподобаннями в минулому з високою ймовірністю демонструватимуть схожу поведінку в майбутньому [8].

Існує два основні типи колабораційної фільтрації:

- User-to-User CF сутність якого полягає у пошуку користувачів, поведінка та вподобання яких є подібними до поточного користувача. На основі оцінок і взаємодій таких користувачів система формує рекомендації товарів, які вони раніше оцінювали позитивно.

- Item-to-item CF – один із найбільш поширених підходів, що активно використовується великими маркетплейсами, зокрема компанією Amazon. У межах цього методу порівнюються не користувачі, а самі товари. Якщо користувач позитивно взаємодіяв з певним товаром, система рекомендує інші елементи, схожі за характеристиками або патернами спільних покупок, які з високою ймовірністю можуть його зацікавити [9].

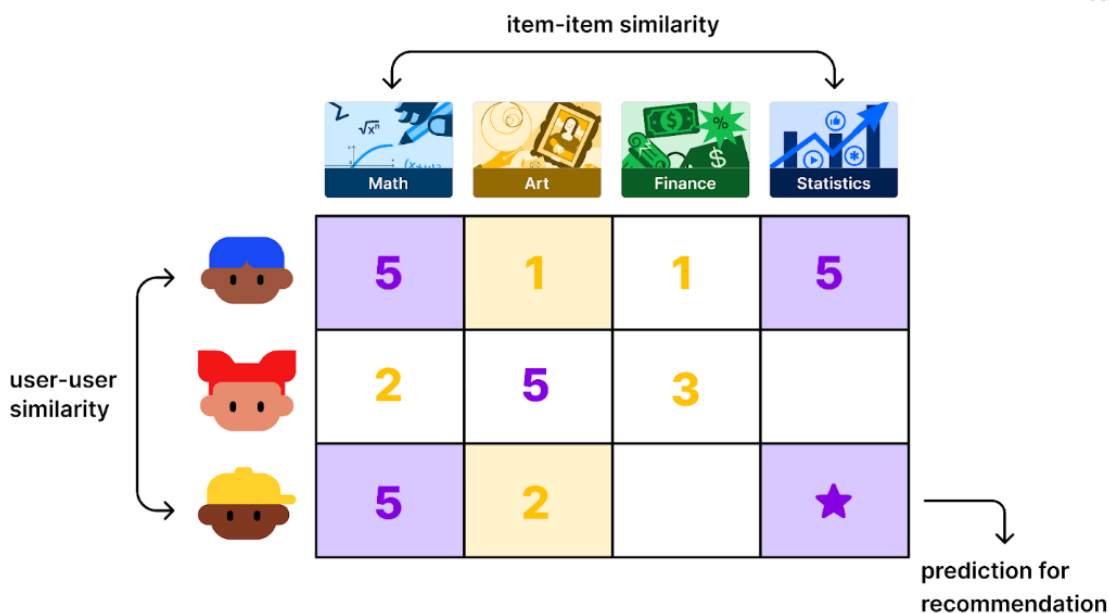


Рис. 1.2 – Принцип роботи item-to-item та user-to-user колаборативних фільтрацій

Серед основних переваг використання колаборативної фільтрації можна виокремити високу точність рекомендацій при великому обсягу даних, здатність виявляти приховані залежності між товарами, здатність працювати без глибокої інформації про характеристики товарів.

Водночас даний підхід має низку недоліків, зокрема, складність роботи у випадку нових товарів, необхідність великої історії поведінкових даних, погіршення якості при високій розрідженості матриці користувач-товар.

Серед українських відомих аналогів можна виділити маркетплейс «Rozetka», він застосовує елементи колаборативної фільтрації у блоках, на кшталт, «Клієнти, які переглядали цей товар, також цікавилися», «З цим товаром купують»(рис. 1.2) та «Вас можуть зацікавити»(рис.1.4).

Формування рекомендацій здійснюється на основі аналізу історії переглядів користувачів, їхньої взаємодії з товарними категоріями, а також рівня популярності товарів серед груп схожих користувачів. Оскільки маркетплейс «Rozetka» має значну аудиторію та великий обсяг трафіку, колаборативні моделі демонструють високу ефективність, адже система

володіє достатньою кількістю поведінкових даних для виявлення стабільних патернів спільної взаємодії з товарами.

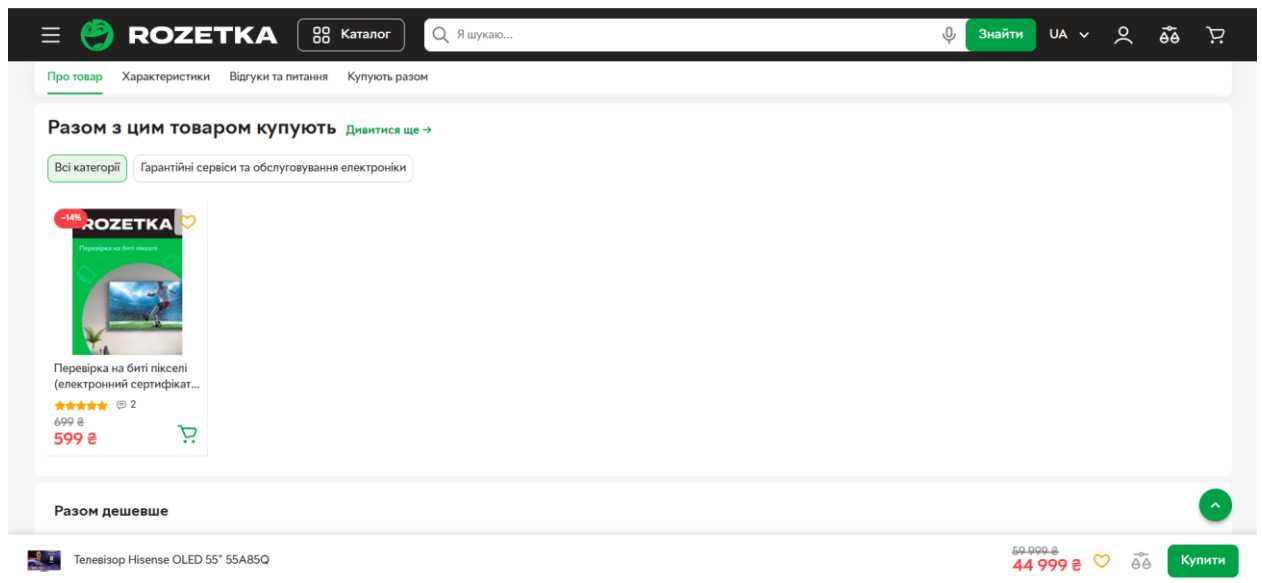


Рис. 1.3 Відображення «З цим товаром купують» на маркетплейсі «Rozetka»

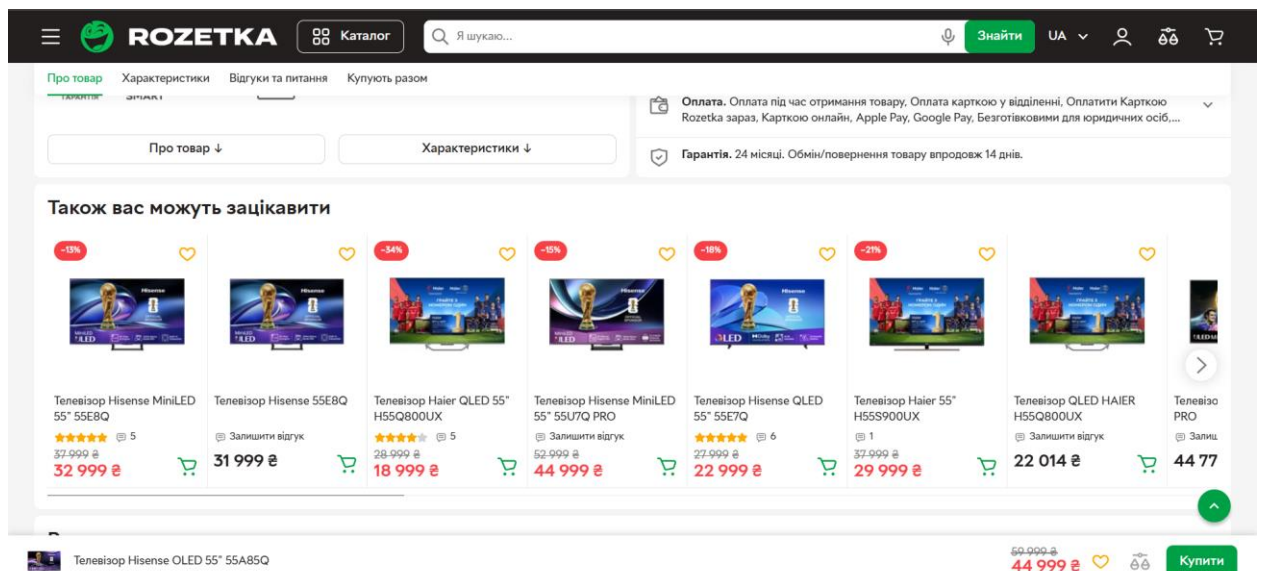


Рис. 1.4 Відображення «Вас можуть зацікавити» на маркетплейсі «Rozetka»

Серед зарубіжних аналогів доцільно розглянути маркетплейс «Amazon», який є однією з перших компаній, що впровадила гібридні рекомендаційні моделі у масштабних комерційних системах. Поєднання декількох підходів до формування рекомендацій дозволяє «Amazon» забезпечувати високий рівень

точності персоналізованих пропозицій для мільйонів користувачів у режимі реального часу. Система додатку включає в себе алгоритми Item-to-item колаборативної фільтрації, контентні методи аналізу товарів, алгоритми ранжування результатів, Моделі машинного навчання, а також компоненти глибокого навчання, що забезпечують адаптивність та масштабованість системи[10].

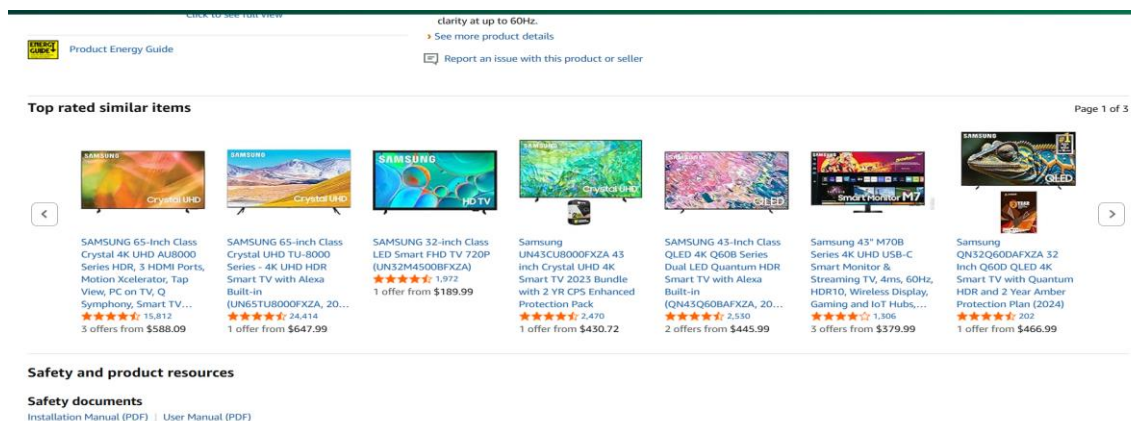


Рис. 1.5 Відображення «рекомендації найбільш оцінених схожих товарів» на маркетплейсі «Amazon»

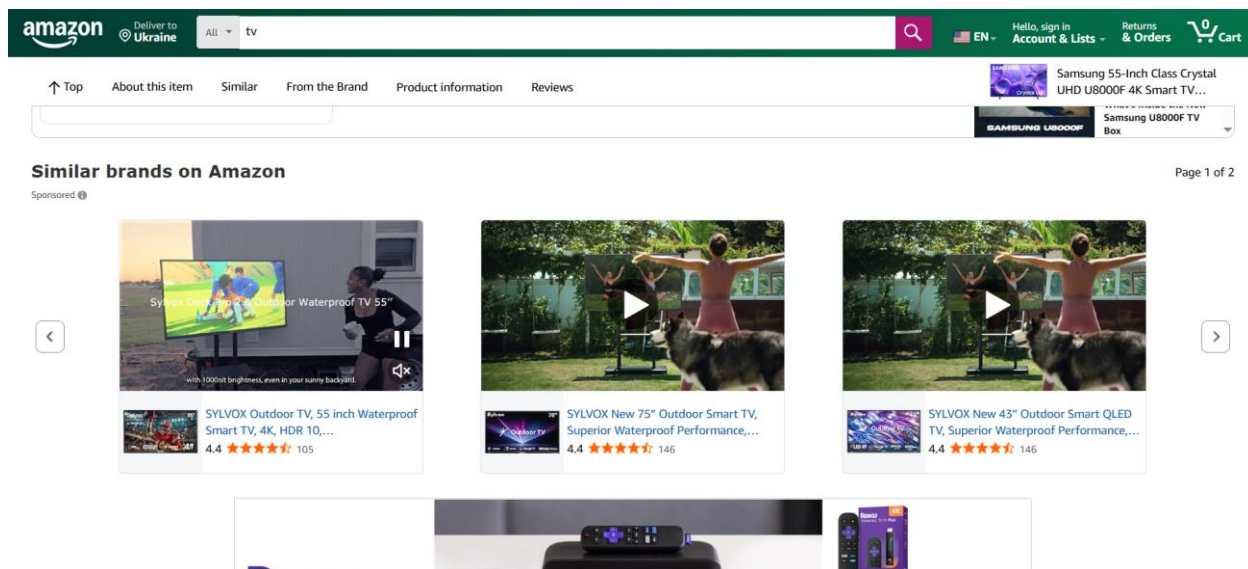


Рис. 1.6 Відображення «рекомендації подібних брендів» на маркетплейсі «Amazon»

Сучасні платформи електронної комерції активно впроваджують глибинні нейронні мережі. Основні типи моделей:

- Recurrent Neural Networks (RNN) – добре моделюють послідовні дії користувача;
- Convolutional Neural Networks (CNN) – Обробляються зображення товарів;
- Transformers та Berd-подібні моделі – застосовуються для аналізу пошукових запитів, оброки текстових описів та розуміння намірів користувача.
- Neural Collaborative Filtering (NCF) – поєднує CF з глибинним навчанням.

Таблиця 1.1

Порівняльна характеристика методів рекомендаційних систем

Метод	Переваги	Недоліки	Приклад використання
Колаборативна фільтрація	Висока точність, виявлення прихованих патернів	Проблема холодного старту	Amazon, Rozetka
SKонтента фільтрація	Працює для нових товарів, зрозуміла логіка	Менше різноманіття рекомендацій	Rozetka
Гібридні моделі	Найкраща точність, масштабованість	Складність реалізації	Amazon
Нейронні мережі	Глибоке розуміння поведінки, масштабованість	Потребують великих даних та ресурсів	Amazon

Аналіз існуючих методів формування рекомендацій показує, що сучасні платформи електронної комерції використовують різноманітні підходи для побудови індивідуального профілю користувача.

«Amazon» демонструє глобально найефективнішу модель item-to-item CF у поєднанні з DL.

«Rozetka», у свою чергу, застосовує змішані моделі для підвищення якості рекомендацій при високих обсягах трафіку.

1.3 Аналіз недоліків сучасних рекомендаційних систем та виклики їх впровадження.

Незважаючи на стрімкий розвиток алгоритмів персоналізації та їх активне впровадження у сучасних системах електронної комерції, рекомендаційні системи залишаються складними програмно-аналітичними компонентами, які мають низку суттєвих обмежень. Значна частина проблем пов'язана не лише з вибором алгоритмів, а й з природою даних, нерівномірною поведінкою користувачів, динамікою товарних каталогів та вимогами до масштабованості й швидкодії системи в реальних умовах експлуатації.

Проблема «холодного старту» є однією з найважливіших у рекомендаційних системах і виникає у двох випадках [25]:

Новий користувач – платформа не має історії взаємодій (перегляд, покупок, пошукових запитів) тому алгоритм не може визначати його вподобання.

Новий товар – Проблема нового товару в тому, що його ще ніхто не купував, оглядав, оцінював. Тому алгоритм не може включити його в модель схожості.

Коли маркетплейс «Rozetka» додає нові товари, особливо помітно при додаванні маловідомих брендів, то вони тривалий час не потрапляють у блоки «Схожі товари» або «З цим товаром купують», оскільки бракує поведінкових даних.

«Amazon» частково вирішує «холодний старт» за допомогою конкретного аналізу описів товарів, попереднього ранжування за категоріями та прогнозування моделей на основі deep learning.

У платформ з великим каталогом товарів існує проблема низької взаємодії на одиницю товару. Наприклад, якщо магазин має 200 000 товарів, то в середньому користувач перегляне від 20 товарів, придбає 1-3, а залишає відгук для 1 товару зі 100. У результаті матриця “користувач-товар” має заповнення менше 0.1%. тому типу наслідки для цього це важкість знаходження схожих користувачів, багато товарів не отримують рекомендацій, знижується точність прогнозів та алгоритм переорієнтовується на популярні товари.

Також найбільшою проблемою є «проблема масштабованості». Збільшення кількості користувачів та товарів змушує систему обробляти більше поведінкових подій, виконувати більше запитів до бази даних та формувати рекомендації у мілісекундному режимі. Для великих платформ щодня генерується мільйони подій які повинні бути оброблені системою рекомендацій[24].

Сучасні алгоритми, особливо ті, що засновані на нейронних мережах, часто виступають як «чорний ящик». Для багатьох платформ важливо пояснити користувачеві чому товар було рекомендовано. Наявність зрозумілих пояснень дозволяє підвищити прозорість роботи рекомендаційної системи та зменшити відчуття нав'язування контенту з боку платформи. Це пов'язано з психологічними аспектами довіри. «Rozetka» іноді використовує такі повідомлення в категоріях техніки, а «Amazon» також застосовує пояснення рекомендацій у формі тегів.

Додатковим викликом є різноманітність товарних каталогів та складність їх структури. Товари можуть суттєво відрізнятися за характеристиками, ціною, категоріями та контекстом використання, що ускладнює побудову універсальної моделі рекомендацій.

Проведений аналіз дозволяє зробити висновок, що жоден із підходів до побудови рекомендаційних систем не є універсальним. Контентні алгоритми мають обмежену здатність до персоналізації, колаборативні методи залежать від якості та обсягу даних, а нейронні моделі потребують значних обчислювальних ресурсів і часто позбавлені прозорості.

Сукупність зазначених проблем підкреслює доцільність використання гібридних рекомендаційних моделей, які поєднують декілька підходів і дозволяють частково компенсувати недоліки кожного з них. Саме такий підхід є найбільш перспективним для реалізації в системах електронної комерції з великим асортиментом товарів і різноманітними сценаріями користувацької поведінки.

1.4 Обґрунтування вибору гібридного підходу та необхідність розробки власної рекомендаційної системи

На основі отриманих даних у даному підрозділі обґрунтується необхідність створення власної гібридної моделі рекомендацій, яка буде використовуватися у розробленій системі електронної комерції. Практичний досвід впровадження рекомендаційних систем у сфері електронної комерції свідчить, що жоден із поширених підходів не позбавлений суттєвих недоліків.

Контентні, колаборативні та нейронні методи по-різному проявляють свої обмеження, які стають особливо відчутними під час роботи з реальними користувачами та великими товарними каталогами. У таких умовах недоліки окремих підходів можуть призводити до зниження релевантності рекомендацій, що безпосередньо впливає на користувацький досвід.

Контентні алгоритми не потребують історії взаємодій і можуть застосовуватися незалежно від активності інших користувачів, що робить їх зручними з точки зору реалізації та пояснюваності результатів. Водночас їх ефективність значною мірою залежить від якості та повноти описів товарів, а обмежений набір ознак часто призводить до формування одноманітних рекомендацій, які слабо враховують приховані інтереси користувачів.

Однак ефективність таких підходів безпосередньо залежить від якості опису товарів. За відсутності повних і уніфікованих метаданих рекомендації часто зводяться до повторення однотипних позицій у межах однієї категорії та не відображають реальних сценаріїв споживчої поведінки, зокрема супутніх покупок.

У середовищі українських маркетплейсів ця проблема проявляється особливо помітно, оскільки інформація про товари нерідко подається фрагментарно або у довільному форматі. У таких умовах використання виключно контентного аналізу призводить до зниження якості рекомендацій і обмежує можливості персоналізації.

Колаборативні методи, на відміну від контентних, дозволяють виявляти приховані взаємозв'язки між товарами та користувачами, спираючись на реальні дії в системі. Вони добре масштабуються зі зростанням обсягу даних і здатні знаходити нетривіальні патерни поведінки.

Водночас ефективність колаборативної фільтрації різко знижується за відсутності достатньої історії взаємодій, а також у ситуаціях, коли матриця користувач–товар є сильно розрідженою. У великих каталогах, подібних до асортименту маркетплейсу «Rozetka», значна частина товарів має мінімальну кількість переглядів або покупок, що ускладнює формування стабільних і персоналізованих рекомендацій та сприяє домінуванню популярних позицій.

Застосування нейронних моделей і трансформерних архітектур відкриває можливості для глибшого аналізу поведінки користувачів і роботи з неструктурованими даними, забезпечуючи високу точність рекомендацій у великих системах. Проте такі рішення потребують значних обчислювальних ресурсів, великих навчальних вибірок і складної інфраструктури, що істотно ускладнює їх впровадження.

Хоча компанії на кшталт «Amazon» успішно використовують глибинне навчання у рекомендаційних механізмах. Для більшості малих і середніх проєктів електронної комерції, подібні підходи залишаються економічно недоцільними через високу вартість розробки, підтримки та масштабування.

Тож можна зробити висновок, що жоден підхід окремо не усуває всі проблеми. Саме тому у сучасних комерційних системах використовують гібридні моделі. Гібридні підходи поєднують сильні сторони кількох алгоритмів і компенсують їх слабкі місця, забезпечуючи більш стабільну якість рекомендацій за різних сценаріїв поведінки користувачів і обсягів доступних даних.

У розробленій системі застосовано практичний, інженерний гібридний алгоритм, який ефективний для середніх комерційних платформ. Платформа реєструє дії користувача такі як перегляд товару, додавання в кошик та купівля. На основі подій будується модель «товари, які переглядали разом». Цей підхід використовуються Amazon з 2003 року як основний.

Висновки до розділу 1

У першому розділі було проведено всебічний аналіз предметної області електронної комерції, який продемонстрував стрімке зростання обсягів онлайн-торговлі та відповідне підвищення вимог до персоналізації користувацького досвіду. Було встановлено, що рекомендаційні системи є ключовим елементом сучасних платформ електронної комерції, оскільки вони безпосередньо впливають на конверсію, середній чек, утримання користувачів та конкурентоспроможності сервісу.

Проведено огляд принципів роботи різних видів рекомендаційних алгоритмів – контентних, колаборативних та гібридних. На основі аналізу архітектури та практичного застосування рекомендаційних систем у провідних платформах, таких як «Rozetka» та «Amazon», визначено їхні переваги та обмеження, а також виявлено ключові технічні й організаційні виклики, пов'язані з впровадженням персоналізованих рекомендацій у нових та середніх за масштабом системах.

У результаті детального аналізу існуючих методів – контентних, колаборативних та нейронних – було обґрунтовано доцільність застосування гібридного підходу. Такий підхід дозволяє компенсувати слабкі сторони

окремих методів і забезпечує високу точність рекомендацій навіть у випадках браку даних, нерівномірної структури подій та слабкої категоризації товарів.

Проведений теоретичний аналіз дав змогу чітко сформулювати вимоги до майбутньої системи рекомендації: автономність, масштабованість, прозорість логіки, можливість інтеграції з наявною архітектурою платформи та здатність працювати з динамічним та розрідженими даними. З огляду на це було визначено, що оптимальним шляхом створення власної спеціалізованої гібридної рекомендаційної системи, адаптованої до технологічного стека застосунку та особливостей українського ринку електронної комерції.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ІЗ ЗАСТОСУВАННЯМ АЛГОРИТМІВ ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ

2.1 Вибір програмних засобів для побудови системи електронної комерції із алгоритмами персоналізованих рекомендацій.

Проектована система електронної комерції передбачає реалізацію складної багатокомпонентної архітектури, що включає вебінтерфейс, серверну бізнес-логіку, базу даних, підсистему автентифікації, а також модуль персоналізованих рекомендацій, який аналізує поведінку користувачів у режимі реального часу. З огляду на це, вибір програмних засобів здійснювався з урахуванням вимог до масштабованості, продуктивності, підтримованості та можливості гнучкої інтеграції рекомендаційних алгоритмів [24] .

Основною мовою програмування для реалізації системи було обрано TypeScript, що є строго типізованим надмножиною JavaScript. Доцільність вибору обумовлена необхідністю підтримки великої кодової бази, де помилки типізації можуть призводити до критичних збоїв у бізнес-логіці електронної комерції, зокрема при роботі з кошиком, замовленням та рекомендаційними вибірками. Завдяки використанню даної мови програмування можна забезпечити статичну перевірку типів на етапі компіляції, зменшити кількість runtime-помилки та покращити інтеграцію з сучасними IDE [26] .

Як середовище серверної частини було обрано Node.js, що реалізує подієво-орієнтовану, неблокуючу модель введення-виведення. Така модель є особливо ефективною для систем електронної комерції, які обслуговують велику кількість одночасних HTTP-запитів, таких як отримання списків товарів, формування рекомендацій, реєстрацію подій користувача та обробку транзакцій. Завдяки асинхронній природі Node.js система здатна ефективно масштабуватися без значного зростання апаратних витрат [27] .

Для побудови повноцінного full-stack вебзастосунку використано Next.js, фреймворк на базі React, який підтримує як клієнтський, так і серверний рендеринг, завдяки цьому сторінку з рекомендаціями можуть формуватися

безпосередньо на сервері, що дозволяє зменшити час першого відображення контенту та забезпечити більш плавний користувацький досвід [28] . Для зберігання даних системи обрано MongoDB, документоорієнтовану NoSQL систему керування базами даних.

Вибір MongoDB зумовлений специфікою предметної області електронної комерції, де дані мають вкладену структуру та часто змінюються. Для взаємодії з базою даних використано Prisma ORM, яка забезпечує типобезпечний доступ до даних та значно спрощує роботу з MongoDB [29] .

Рекомендаційний модуль системи реалізується без використання сторонніх ML-платформ або хмарних API, що обумовлено прагненням забезпечити повний контроль над логікою рекомендацій та зменшити експлуатаційні витрати. Основна логіка рекомендацій побудована з використанням аналізу поведінкових подій користувача, алгоритмів item-to-item та контентної фільтрації на основі категорій товарів. Завдяки такому підходу можливо формувати рекомендації в реальному часі, уникнути проблем з cold start та легко масштабувати систему.

Для збору даних про взаємодію користувачів з товарами реалізовано механізм фіксації подій, який зберігає різні типи активностей, такі як перегляд товару, додавання товару до кошика та оформлення замовлення. Кожна подія містить ідентифікатор користувача, ідентифікатор товару, тип події та часову помітку. Ці дані використовуються рекомендаційним модулем для побудови персоналізованих списків товарів.

Також було свідомо відмовлено від використання сторонніх сервісів рекомендацій. Наприклад Google Recommendations AI, оскільки вони працюють як «чорні скриньки» та потребують передачі поведінкових даних третім сторонам і обмежують гнучкість алгоритмів. Реалізація власного рекомендаційного модуля дозволяє повністю контролювати логіку персоналізації та адаптувати її під конкретні бізнес-вимоги.

2.2 Проєктування структури даних та зберігання поведінкових подій користувачів.

Ефективність алгоритмів персоналізованих рекомендацій у системах електронної комерції безпосередньо залежить від якості, повноти та структури вхідних даних. Основним джерелом інформації для формування рекомендацій у розроблювальній системі є поведінкові дані користувачів, які відображаються їхню взаємодію з товарами та функціональними елементами платформи. Тому ключовим етапом проєктування є побудова коректної моделі зберігання даних, яка забезпечує можливість ефективного аналізу дій користувачів у часовому та контекстному вимірах.

Структура даних повинна відповідати таким вимогам:

- Забезпечувати збереження історії дій користувачів;
- Підтримувати масштабування при зростанні кількості користувачів та товарів;
- Дозволяти виконання швидких аналітичних запитів для рекомендацій;
- Бути гнучкою для розширення новими типами подій;
- Гарантувати цілісність та узгодженість даних.

З огляду на ці вимоги було прийнято рішення використовувати документоорієнтований підхід до зберігання даних із чітким логічним поділом на основні сутності системи. У межах розроблювальної системи було визначено ключеві сутності, кожна з яких відіграє важливу роль у формуванні рекомендацій та взаємодії між компонентами системи.

Сутність користувача представляє зареєстрованого або автентифікованого відвідувача платформи. Користувач є центральним елементом персоналізації, оскільки всі рекомендації формуються відносно його поведінки. Зберігання користувачів у централізованій колекції дозволяє швидко агрегувати історію дій для кожного профілю та будувати персоналізовані рекомендації.

Модель товару є базовою одиницею контенту в системі. Саме товари рекомендуються користувачам, порівнюються між собою та аналізуються у контексті спільних покупок. Для формування структури використовуються назви товарів, унікальні ідентифікатори, ціна, категорія, зображення та рейтинги з відгуками. Така структура дозволяє використовувати як поведінкові, так і контентні ознаки товарів при формуванні рекомендацій.

Категорія товару є важливим атрибутом, який використовується в алгоритмах рекомендацій як fallback-механізм у випадках, коли недостатньо поведінкових даних. Вони дозволяють групувати товари за тематичною ознакою та виконувати рекомендації на основі подібності контенту.

Замовлення відображає факт завершеної покупки користувачем та є найціннішим сигналом для рекомендаційних алгоритмів. На відміну від переглядів, замовлення мають високий рівень значущості, оскільки свідчать про реальний намір придбати товар.

Ключовим компонентом рекомендаційної системи є модель поведінкових подій користувача. Саме вона фіксує всі дії, які користувач виконує в межах програми. Подію можна вважати будь-яку взаємодію користувача з товаром. Зберігання подій у вигляді окремої колекції дозволяє ефективно аналізувати послідовність дій користувачів, а також виконувати агрегацію подій для побудови рекомендаційних списків.

Проектована структура даних забезпечує логічні зв'язки між основними сутностями системи, зокрема між користувачами та їхніми подіями, користувачами та замовленнями, товарами та подіями, а також товарами і відповідними категоріями. Наявність таких зв'язків дозволяє виконувати складні аналітичні запити, необхідні для реалізації алгоритмів колаборативної та контентної фільтрації, а також для побудови гібридних рекомендаційних моделей.

Таким чином, спроектована структура даних створює надійний фундамент для роботи рекомендаційного модуля, забезпечуючи баланс між

гнучкістю, масштабованістю та ефективністю аналізу поведінкових даних у системі електронної комерції.

2.3 Алгоритми, схеми та діаграми програмного забезпечення

Функціонування систем електронної комерції з персоналізованими рекомендаціями ґрунтується на сукупності взаємопов'язаних алгоритмів, які забезпечують обробку дій користувачів, аналіз накопичених даних та формування релевантних рекомендацій у режимі реального часу. Саме алгоритмічне забезпечення визначає логіку взаємодії між користувачем, серверною частиною платформи та рекомендаційним модулем, а також безпосередньо впливає на якість користувацького досвіду.

На відміну від статичних інформаційних систем, платформи електронної комерції працюють із динамічними даними, що постійно оновлюються. До таких даних належать перегляди товарів, покупки, взаємодія з товарними категоріями, а також зміни асортименту. За таких умов алгоритми виконують роль адаптивного механізму, який дозволяє системі оперативно реагувати на зміну поведінки користувачів і формувати персоналізований контент.

Алгоритмічна модель рекомендаційної системи реалізується за модульним принципом. Кожен алгоритм відповідає за виконання окремої функціональної задачі, зокрема збір подій, аналіз поведінкових даних, обчислення релевантності товарів та агрегацію результатів. Такий підхід спрощує масштабування системи й дозволяє модифікувати або вдосконалювати рекомендаційні методи без втручання в базову логіку роботи інтернет-магазину.

Роботу системи персоналізованих рекомендацій умовно можна поділити на кілька ключових етапів, які виконуються послідовно та утворюють єдиний алгоритмічний процес. На першому етапі користувач взаємодіє з веб-інтерфейсом платформи електронної комерції. Система здійснює ідентифікацію користувача, після чого ініціалізує з'єднання з базою даних і рекомендаційним модулем для подальшої обробки інформації.

У процесі взаємодії користувача з платформою система реєструє ключові поведінкові події, зокрема перегляди товарів, додавання до кошика та здійснення покупок. Кожна подія зберігається у вигляді структурованого запису, який містить ідентифікатор користувача, ідентифікатор товару, тип події та часову мітку.

На основі накопичених подій формується поведінковий профіль користувача. Даний етап є критично важливим для персоналізації рекомендацій, оскільки дозволяє враховувати як короткострокові інтереси користувача, пов'язані з поточною сесією, так і довгострокові вподобання, сформовані за результатами попередніх взаємодій.

На основі результатів аналізу система запускає декілька незалежних алгоритмів:

- item-to-item колаборативну фільтрацію, яка знаходить товари, що часто переглядаються або купуються разом;
- персоналізовані рекомендації, засновані на історії конкретного користувача;
- контентні рекомендації, що використовуються як резервний механізм за відсутності достатніх поведінкових даних;
- алгоритм «користувачі часто купували разом», який аналізує склад замовлень.

Результати роботи окремих алгоритмів об'єднуються в єдиний список. Після завершення обчислень система передає готовий список рекомендацій на клієнтську частину.

Для формалізації описаної логіки у роботі використовується блок-схема алгоритму роботи рекомендаційної системи. Блок-схема дозволяє візуально представити послідовність обробки даних, точки прийняття рішень та взаємодію між основними модулями системи.

Блок-схема формування персоналізованих рекомендацій

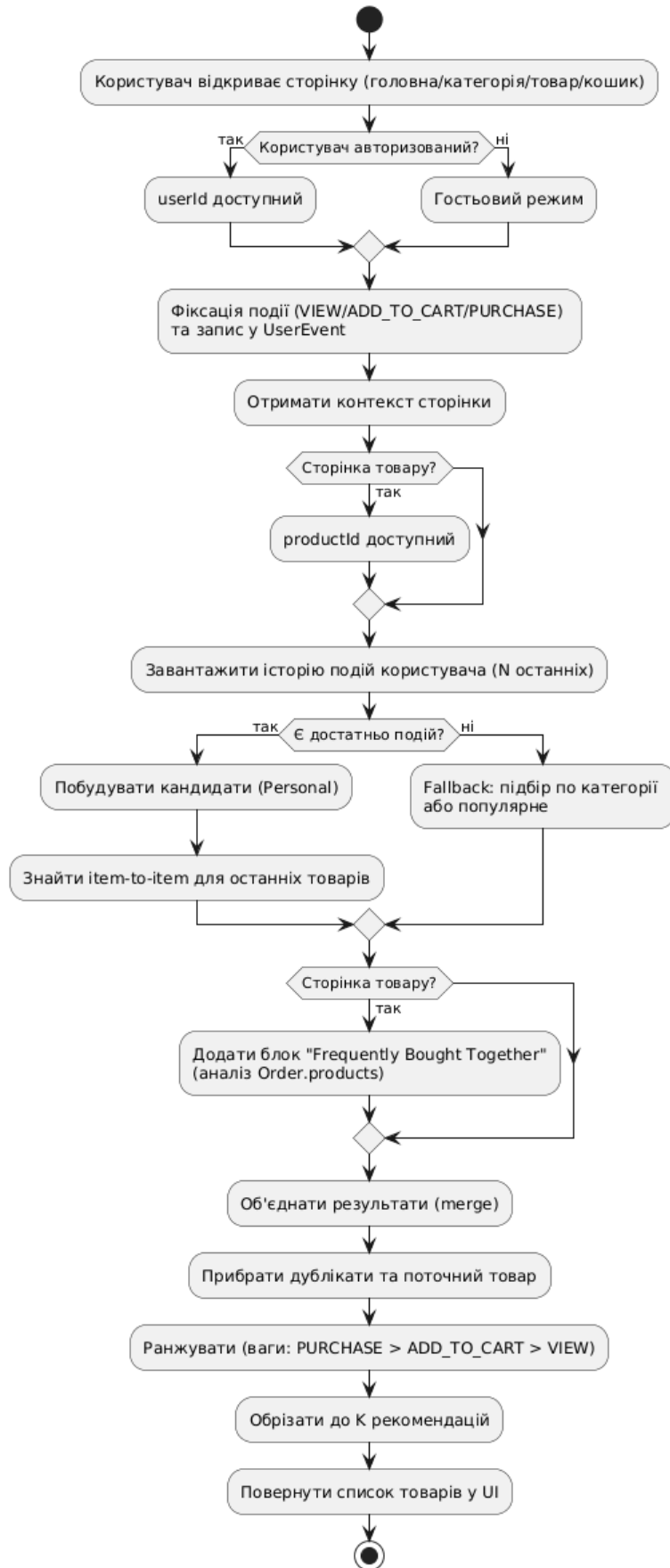


Рис.2.1 Блок-схема формування персоналізованих рекомендацій.

Блок-схема, наведена на рис. 2.1, відображає загальну логіку функціонування системи електронної комерції з інтегрованим модулем персоналізованих рекомендацій. Вона демонструє послідовність обробки взаємодії користувача з платформою — від відкриття сторінки до формування та відображення релевантних товарних пропозицій.

На початковому етапі система ініціалізує взаємодію з користувачем, визначає контекст сторінки та перевіряє статус авторизації. Для авторизованих користувачів використовується персоналізований профіль, тоді як для неавторизованих застосовуються загальні або контентні рекомендації.

У процесі роботи платформи фіксуються ключові поведінкові події користувача, зокрема перегляди товарів, додавання до кошика та покупки. На основі збережених подій формується поведінковий профіль, який використовується для подальшого аналізу. Для підвищення актуальності рекомендацій система враховує обмежену кількість останніх дій користувача.

За наявності достатнього обсягу даних запускається модуль персоналізованих рекомендацій, що формує список товарів-кандидатів із використанням алгоритмів колаборативної фільтрації типу *item-to-item*. У разі недостатньої кількості даних система переходить до резервного механізму, використовуючи рекомендації за категоріями або популярними товарами.

Отримані результати проходять етап агрегації та ранжування, після чого передаються на клієнтську частину системи для відображення користувачеві. Таким чином, блок-схема ілюструє повний цикл роботи рекомендаційної підсистеми та підтверджує модульний і адаптивний характер розробленої платформи.

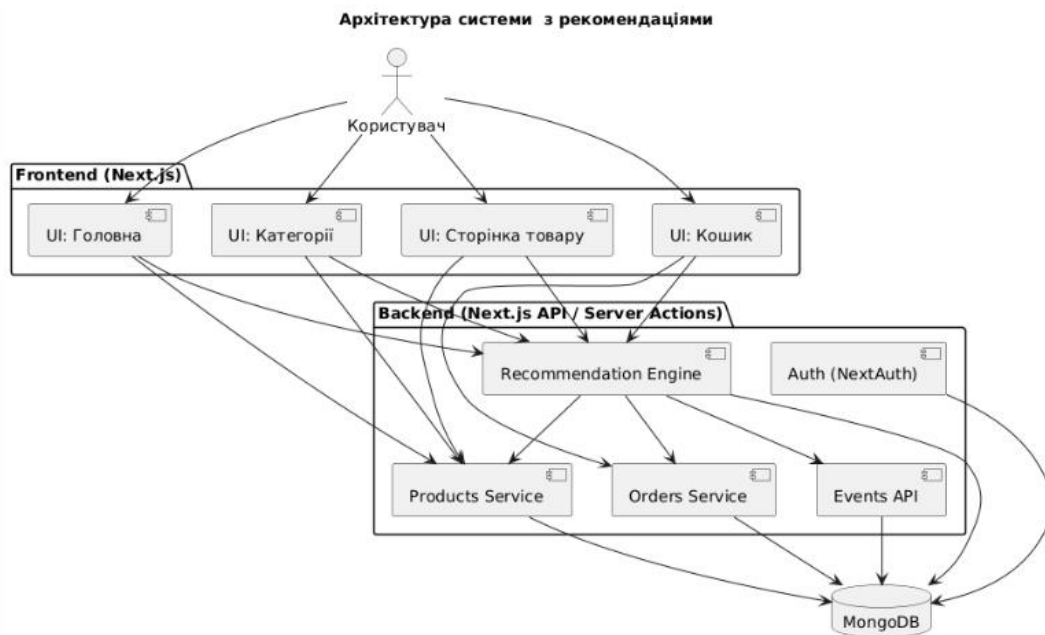


Рис. 2.2 Блок-схема архітектури системи з рекомендаціями.

Архітектурна блок-схема системи, наведена на рисунку 2.2, відображає загальну структуру програмного забезпечення електронної комерції з інтегрованим модулем персоналізованих рекомендацій та взаємодію між її основними компонентами. Архітектура побудована за принципом чіткого розділення відповідальностей між клієнтською частиною, серверною логікою, рекомендаційним модулем і базою даних.

Користувач взаємодіє з платформою через веб-інтерфейс, реалізований на основі фреймворку Next.js. Клієнтська частина відповідає за відображення сторінок магазину (головна сторінка, категорії, сторінка товару, кошик) та отримання рекомендацій, сформованих серверною частиною системи [28].

Обмін даними між клієнтською та серверною частинами здійснюється через API-запити. Серверна логіка відповідає за обробку запитів користувачів, реалізацію бізнес-логіки електронного магазину, авторизацію, роботу з товарами та замовленнями. Окремим компонентом серверної частини є сервіс збору поведінкових подій користувачів, який фіксує перегляди товарів, додавання до кошика та здійснення покупок.

Ключовим елементом архітектури є модуль персоналізованих рекомендацій який агрегує дані з сервісів товарів, замовлень і подій

користувачів. На основі отриманої інформації модуль формує рекомендації із застосуванням колаборативної фільтрації типу item-to-item та алгоритму «користувачі часто купували разом». Recommendation Engine функціонує як обчислювальний компонент і не зберігає власного стану, що забезпечує гнучкість та можливість розширення набору алгоритмів.

Усі серверні компоненти взаємодіють з централізованою базою даних MongoDB, яка використовується для зберігання інформації про користувачів, товари, замовлення та поведінкові події. Клієнтська частина не має прямого доступу до бази даних, а всі запити виконуються через серверну логіку, що підвищує рівень безпеки та контроль доступу до даних [29] .

Таким чином, архітектурна блок-схема демонструє багаторівневу структуру системи електронної комерції з чітким розмежуванням ролей між компонентами та підтверджує масштабованість і адаптивність розробленої платформи.

2.4 Узагальнення та обґрунтування архітектурних та проєктних рішень системи.

Проектування системи електронної комерції з використанням персоналізованих рекомендацій належить до складних інженерних завдань, оскільки вимагає узгодженого підходу до побудови архітектури, організації даних та взаємодії між окремими компонентами системи. На практиці такі платформи поєднують традиційну бізнес-логіку інтернет-магазину з аналітичними механізмами, що спираються на аналіз поведінки користувачів і працюють у режимі, наближеному до реального часу.

У другому розділі було розроблено архітектурну модель системи з урахуванням особливостей електронної комерції та вимог до персоналізації, масштабування й постійного накопичення даних. Запропоноване архітектурне рішення базується на модульному підході, що дає змогу розвивати окремі підсистеми незалежно одна від одної, не порушуючи загальної стабільності та логіки роботи платформи.

Одним із базових архітектурних рішень є поділ системи на клієнтську та серверну частини з чітким розмежуванням їхніх функцій. Клієнтська частина зосереджується на взаємодії з користувачем і відображенні інформації, тоді як серверна частина відповідає за реалізацію бізнес-логіки, роботу з базою даних і виконання рекомендаційних алгоритмів. Такий підхід спрощує контроль доступу до даних, підвищує рівень безпеки та дозволяє оптимізувати продуктивність за рахунок формування рекомендацій на сервері й зменшення обсягу даних, що передаються клієнтській частині.

Важливе місце в архітектурі системи займає рекомендаційна підсистема, яка спроектована як окремий логічно незалежний модуль. Таке рішення дає змогу реалізовувати рекомендаційні алгоритми без жорсткої прив'язки до конкретних сторінок або сценаріїв взаємодії з платформою. Сформовані рекомендації можуть використовуватися на різних етапах роботи користувача з системою — на головній сторінці, у категоріях, на сторінках товарів, у кошику або після завершення замовлення. Універсальність цього підходу спрощує повторне використання алгоритмів і підвищує практичну цінність рекомендаційного модуля для бізнесу.

Використання поведінкових подій як основного джерела даних для персоналізації є доцільним, оскільки такі дані відображають фактичну взаємодію користувачів із платформою. На відміну від задекларованих уподобань, події фіксують реальні дії — перегляди, переходи та покупки, що підвищує достовірність сформованих рекомендацій. Запропонована модель зберігання подій дає змогу аналізувати як поточні інтереси користувача в межах окремої сесії, так і стійкі поведінкові патерни, які формуються з часом у процесі використання системи.

Обрана стратегія формування рекомендацій базується на поєднанні кількох підходів, зокрема колаборативної фільтрації типу *item-to-item*, контентної фільтрації за категоріями товарів та аналізу спільних покупок. Використання такого гібридного підходу дозволяє зменшити вплив обмежень, притаманних окремим алгоритмам. Наприклад, проблему холодного старту

вдається частково нівелювати за рахунок контентних рекомендацій, тоді як підвищення точності персоналізації досягається поступово, у міру накопичення та аналізу поведінкових подій користувачів.

Під час проектування системи було прийнято рішення відмовитися від використання зовнішніх рекомендаційних сервісів. Попри те, що такі рішення дозволяють швидко отримати базову функціональність, їх застосування призводить до залежності від сторонніх провайдерів, обмежує можливості налаштування алгоритмів і потребує передачі поведінкових даних користувачів третім сторонам. Реалізація власного рекомендаційного модуля надає повний контроль над логікою персоналізації, дозволяє адаптувати алгоритми під конкретні бізнес-сценарії та забезпечити дотримання вимог щодо конфіденційності й захисту даних.

Запропонована архітектура передбачає можливість подальшого розвитку та розширення функціональності системи. Зокрема, до рекомендаційного модуля у майбутньому можуть бути інтегровані додаткові підходи, такі як вагове врахування поведінкових подій, часовий облік зміни інтересів користувачів або експериментальні моделі машинного навчання. Завдяки модульній побудові системи впровадження таких змін не потребує істотного перегляду загальної архітектури та може виконуватися поступово.

Висновки до розділу 2

У другому розділі дипломної роботи було виконано проектування системи електронної комерції з використанням персоналізованих рекомендацій. Основна увага приділялася формуванню архітектури програмного забезпечення, вибору технологічного стеку, побудові структури даних і визначенню алгоритмічної логіки, необхідної для коректної роботи інтернет-магазину та рекомендаційної підсистеми.

На початковому етапі обґрунтовано вибір програмних засобів для реалізації системи. Доцільність застосування full-stack підходу з використанням TypeScript, Node.js та фреймворку Next.js пояснюється потребою у масштабованому та продуктивному рішенні з підтримкою як

клієнтського, так і серверного рендерингу. Для зберігання даних було обрано базу даних MongoDB у поєднанні з ORM Prisma, що відповідає вимогам до роботи з динамічними структурами, характерними для електронної комерції.

У другому підрозділі було спроектовано структуру даних системи з акцентом на збереження та подальший аналіз поведінкових подій користувачів. Визначено основні сутності платформи, користувачів, товари, категорії, замовлення та події взаємодії, а також логічні зв'язки між ними. Запропонована модель дозволяє накопичувати історію дій користувачів, аналізувати зміну їхніх інтересів у часі та використовувати ці дані для персоналізації рекомендацій.

Алгоритмічні аспекти функціонування системи розглянуто у третьому підрозділі. У ньому описано послідовність роботи рекомендаційної підсистеми — від фіксації поведінкових подій до формування списків рекомендованих товарів. Застосування гібридного підходу, що поєднує item-to-item колаборативну фільтрацію, контентні рекомендації та аналіз спільних покупок, дозволяє підвищити релевантність результатів і зменшити вплив проблеми холодного старту.

Для наочного представлення логіки роботи системи та взаємодії між її компонентами використано блок-схеми, які відображають як алгоритм формування рекомендацій, так і загальну архітектуру платформи. Їх застосування полегшує розуміння структури програмного забезпечення та створює основу для подальшого розширення й оптимізації системи.

Опис архітектури продемонстрував принципи взаємодії між клієнтською частиною, серверною логікою, рекомендаційним модулем і базою даних. Запропоноване рішення забезпечує чіткий розподіл відповідальностей, підвищує рівень безпеки доступу до даних і дозволяє масштабувати систему зі зростанням кількості користувачів та товарів.

Таким чином, у другому розділі сформовано завершену проєктну основу для реалізації системи електронної комерції з персоналізованими рекомендаціями.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЙОГО ТЕСТУВАННЯ

3.1 Реалізація основних компонентів системи

У процесі розробки системи електронної комерції було реалізовано власний рекомендаційний модуль, який функціонує без залучення зовнішніх ML-платформ або сторонніх API. Модуль орієнтований на формування товарних рекомендацій на основі реальної поведінки користувачів, зокрема історії переглядів, покупок і базових контентних характеристик товарів.

Рекомендаційна логіка реалізована як окремий програмний компонент, що взаємодіє з базою даних через ORM Prisma та використовує накопичені поведінкові події, збережені у таблиці `userEvent`. Така організація дозволяє інтегрувати рекомендаційний механізм у наявну бізнес-логіку інтернет-магазину без змін базової архітектури та спрощує подальший розвиток модуля.

Одним з базових алгоритмів, реалізованих у системі, є `item-to-item` колаборативна фільтрація. Вона дозволяє знаходити товари, що часто переглядаються різними користувачами разом з поточним товаром.

```
async function getItemToItem(productId: string) {  
  const events = await prisma.userEvent.findMany({  
    where: { productId },  
  });
```

На першому етапі система отримує всі поведінкові події, пов'язані з конкретним товаром. Це дозволяє визначити перелік користувачів, які взаємодіяли з даним товаром.

```
const users = events.map((e) => e.userId);  
if (users.length === 0) return [];
```

У разі відсутності подій алгоритм припиняє роботу, що запобігає зайвим запитам до бази даних. Далі виконується пошук інших товарів, з якими взаємодіяли ті самі користувачі.

```
const relatedEvents = await prisma.userEvent.findMany({
  where: {
    userId: { in: users },
    productId: { not: productId },
  },
});
```

Отримані події агрегуються за частотою. Це дозволяє визначити найбільш пов'язані товари.

```
const freq: Record<string, number> = {};
relatedEvents.forEach((ev) => {
  freq[ev.productId] = (freq[ev.productId] || 0) + 1;
});
```

Після сортування результатів система повертає список найбільш релевантних товарів. Таким чином реалізується класичний item-to-item підхід, який широко застосовується у системах електронної комерції.

Для підвищення рівня персоналізації в системі реалізовано механізм побудови профілю інтересів користувача на основі його останніх дій.

```
async function getUserInterestProfile(userId: string) {
  const events = await prisma.userEvent.findMany({
    where: { userId },
    orderBy: { createdAt: "desc" },
    take: 50,
  });
```

Алгоритм аналізує останні 50 подій користувача. Такий підхід дозволяє враховувати актуальні інтереси та уникати впливу застарілих дій.

```
const products = await prisma.product.findMany({
  where: { id: { in: productIds } },
});
```

Отримані товари групуються за категоріями, після чого формується рейтинг категорій за частотою. Такий підхід дозволяє виділити найбільш

характерні напрями інтересів користувача та використати їх для подальшої персоналізації рекомендацій, орієнтованої на його актуальні вподобання.

```
const freq: Record<string, number> = {};  
relatedEvents.forEach((ev) => {  
  freq[ev.productId] = (freq[ev.productId] || 0) + 1;  
});
```

У результаті формується впорядкований список категорій, що найбільш відповідають інтересам конкретного користувача. Отриманий профіль використовується як один із складових елементів гібридної рекомендаційної моделі.

У випадках, коли поведінкових даних недостатньо, використовується контентна фільтрація за категорією товару. Це дозволяє формувати релевантні рекомендації на основі описових характеристик товарів і підтримувати коректну роботу системи навіть для нових користувачів або нових позицій каталогу.

```
async function getSimilarByCategory(category: string) {  
  return prisma.product.findMany({  
    where: { category },  
    take: 10,  
  });  
}
```

Даний механізм виконує роль резервного алгоритму та дозволяє зменшити проблему холодного старту для нових користувачів або нових товарів. Ключовим елементом рекомендаційного модуля є гібридний алгоритм, який поєднує декілька підходів. У разі відсутності поведінкових подій система повертає базовий список товарів.

```
if (events.length === 0) {  
  return prisma.product.findMany({ take: 10 });  
}
```

Для користувачів формується комбінований список рекомендацій:

- item-to-item рекомендації;
- рекомендації на основі інтересів;
- товари однієї категорії;
- популярні товари.

```
const combined = [
  ...itemBased.slice(0, 5),
  ...interestBased.slice(0, 5),
  ...similarCategory.slice(0, 3),
  ...popular.slice(0, 3),
];
```

Для сторінки конкретного товару реалізовано окремий сценарій рекомендацій. Алгоритм поєднує товари з однієї категорії та персональні рекомендації користувача.

```
const topCategory = sameCategory.slice(0, 4); // 4 категорійні
const topPersonal = personal.slice(0, 2); // 2 персональні
```

Це дозволяє зберегти контекст поточного товару та одночасно врахувати індивідуальні вподобання користувача. Для підвищення конверсії в системі реалізовано алгоритм аналізу спільних покупок

```
export async function getFrequentlyBoughtTogether(productId: string) {
  const orders = await prisma.order.findMany({
    where: {
      products: { some: { id: productId } },
    },
  });
}
```

Алгоритм аналізує замовлення, в яких присутній поточний товар, та визначає інші товари, що найчастіше купуються разом з ним. Це дозволяє формувати рекомендації, орієнтовані на завершення покупки.

3.2 Структура інтерфейсу веб-додатку

Під час відкриття платформи через браузер користувачеві відображається стартова сторінка застосунку, з якої розпочинається взаємодія з системою. Головна сторінка проєкту (рис 3.1).

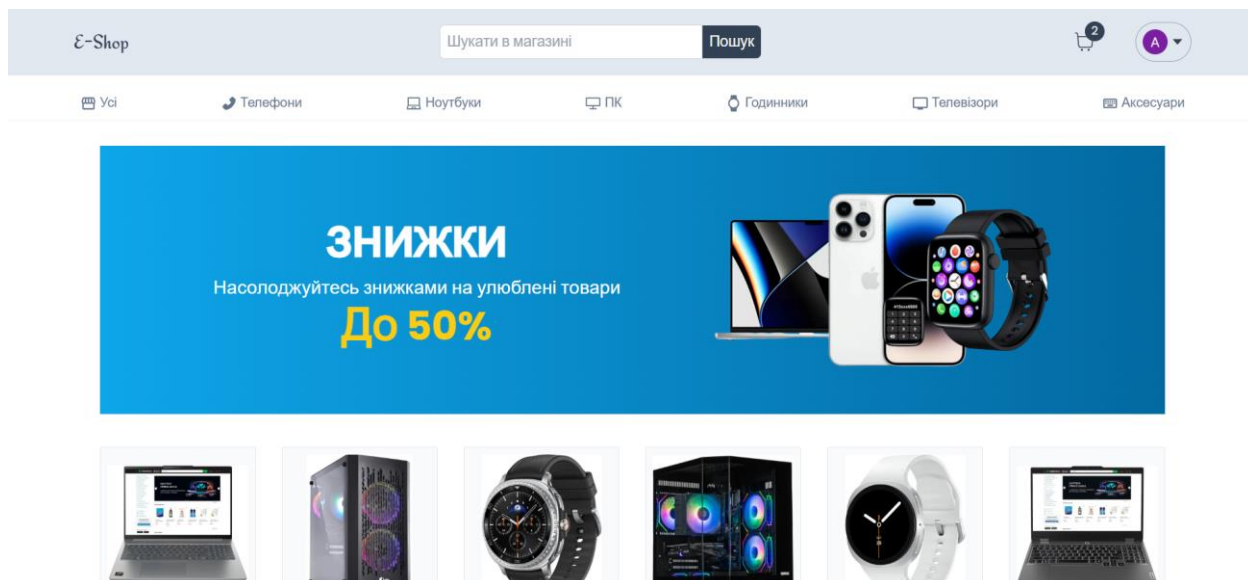


Рис 3.1 Головна сторінка проєкту

Після переходу до сторінки аутентифікації відкривається форма входу де потрібно ввести необхідні дані, такі як електронна пошта та пароль. Або увійти до системи за допомогою кнопки «Продовжити через Google». У випадку відсутності аккаунту, є кнопка «Зареєструватися»(рис 3.2).

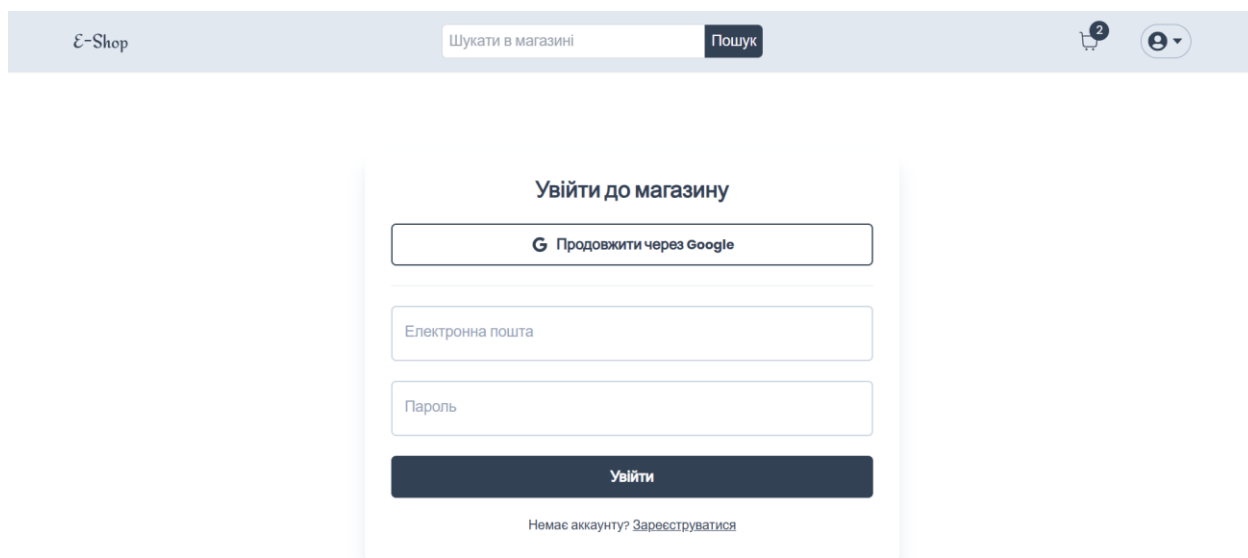


Рис 3.2 Форма входу у систему

Після успішної авторизації користувач перенаправляється до головної сторінки(рис.3.1). Інтерфейс системи буде поступово змінюватись в залежності від переглянутих користувачем товарів, відображаючи товари тієї категорії до яких він мав найбільший інтерес.(рис 3.3)

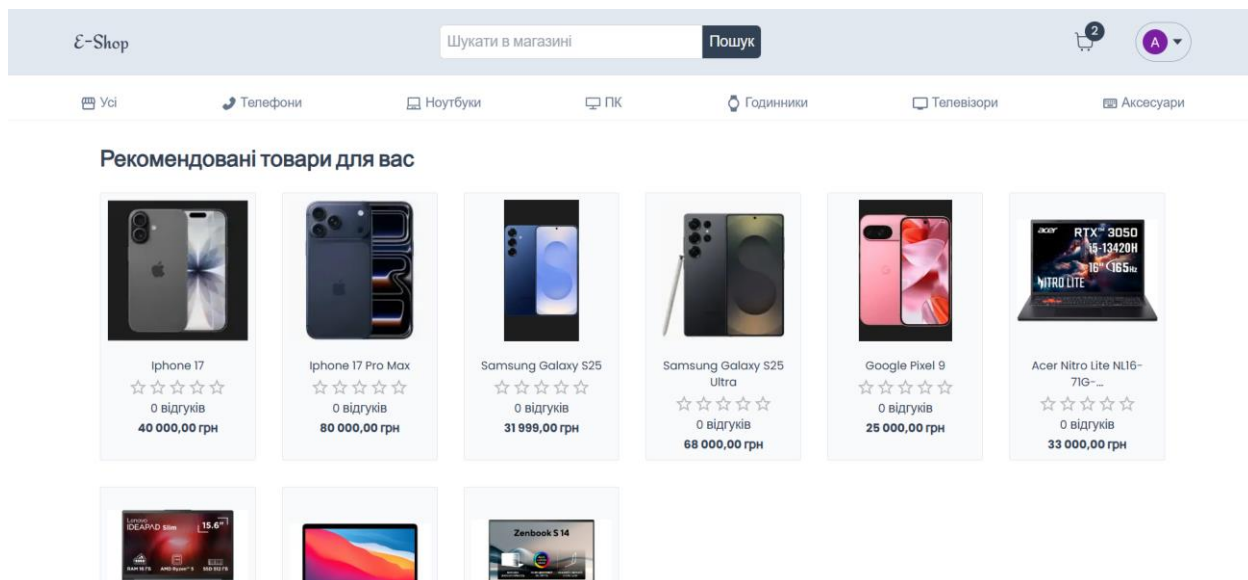


Рис 3.3 Відділ рекомендованих товарів для користувачів на головній сторінці.

Панель адміністрування надає доступ до отримання статистики, додавання товару, редагування товару та відстежування замовлення(рис 3.4). Додавання товару та редагування товару є важливим функціоналом для успішного та ефективного функціонування сайту.

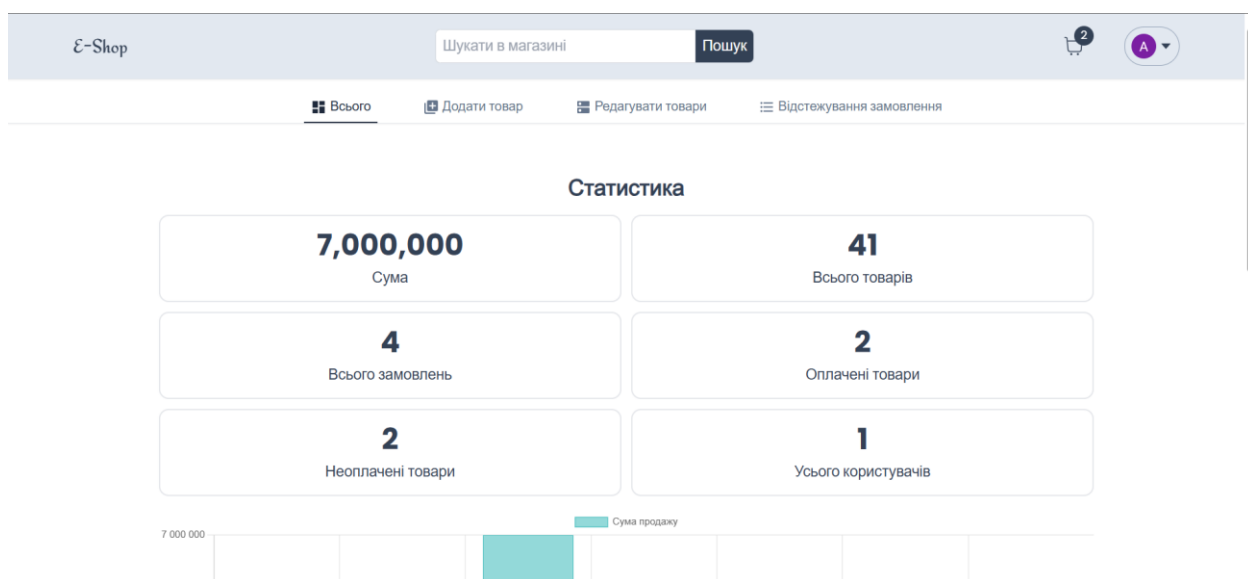


Рис 3.4 Сторінка адміністратора

Розділ «Додати товар» дозволяє додавати необхідний товар у систему. Для успішного додавання товару потрібно ввести назву, ціну, бренд товару та його опис(рис 3.5). Окрім цього потрібно обрати необхідну категорію товару та завантажити фотографії товару у розділі кольору(рис3.6).

The screenshot shows the 'Додати товар' (Add Item) form within the E-Shop interface. The header includes the 'E-Shop' logo, a search bar with the placeholder 'Шукати в магазині', and a 'Пошук' (Search) button. The navigation bar contains links for 'Всього' (All), 'Додати товар' (Add Item), 'Редагувати товари' (Edit Items), and 'Відстежування замовлення' (Order Tracking). The form itself is titled 'Додати товар' and contains four input fields: 'Назва' (Name), 'Ціна' (Price), 'Бренд' (Brand), and 'Опис товару' (Item Description).

Рис 3.5 Перша частина сторінки створення товару

The screenshot shows the 'Вибір товару та кольору' (Select Item and Color) section of the E-Shop interface. It features a grid of category buttons: 'Усі' (All), 'Телефони' (Phones), 'Ноутбуки' (Laptops), 'ПК' (PCs), 'Годинники' (Watches), 'Телевізори' (Televisions), and 'Акcesуари' (Accessories). Below the grid, a message states: 'Виберіть доступний товар та оберіть зображення. Ви поминні додати зображення інакше обрани вами колір буде проігноровано' (Select an available item and choose an image. You will not be able to add an image if you have already chosen a color, it will be ignored). There are six color selection options, each with a checkbox: 'Білий' (White), 'Чорний' (Black), 'Срібний' (Silver), 'Сірий' (Gray), 'червоний' (Red), and 'золотий' (Gold).

Рис 3.6 Друга частина сторінки створення товару

На сторінці «редагування товару» можна зміни дані товару , видалити товар або переглянути його. Такий підхід допомагає швидко реагувати на будь-які зміни які відбуваються з товаром.(рис 3.7).

Є-Shop

Шукати в магазині

Пошук

2

А

Всього

Додати товар

Редагувати товари

Відстежування замовлення

Редагування товару

☐	ID	Назва товару	Ціна	Категорія	Бренд	Наявність	Дії
☐	6939c4886d4d6053bd52e020	Iphone 17	40 000,00 грн	Телефони	Apple	В наявності ✓	
☐	6939c5c96d4d6053bd52e021	Iphone 17 Pro Max	80 000,00 грн	Телефони	Apple	В наявності ✓	
☐	6939c6216d4d6053bd52e022	Samsung Galaxy S25	31 999,00 грн	Телефони	Samsung	В наявності ✓	
☐	6939c6646d4d6053bd52e023	Samsung Galaxy S25 Ultra	68 000,00 грн	Телефони	Samsung	В наявності ✓	
☐	6939c6c96d4d6053bd52e024	Google Pixel 9	25 000,00 грн	Телефони	Google	В наявності ✓	
☐	6939caa56d4d6053bd52e027	Google Pixel 10	33 000,00 грн	Телефони	Google	В наявності ✓	
☐	6939ca36d4d6053bd52e028	Samsung Galaxy Flip 10 FE	33 000,00 грн	Телефони	Samsung	В наявності ✓	
☐	6939cb476d4d6053bd52e029	Xiaomi Redmi Note 14	8 000,00 грн	Телефони	Xiaomi	В наявності ✓	

Рис 3.7 Сторінка редагування товару

На сторінці «відстежування замовлення» адміністратор може змінювати статус відправки замовлень на «доставлено» або «відправлено». Завдяки цьому можна легко адмініструвати та слідкувати замовлення.

Є-Shop

Шукати в магазині

Пошук

2

А

Всього

Додати товар

Редагувати товари

Відстежування замовлення

Відстежування замовлення

☐	ID	Ім'я клієнта	Сума	Статус оплати	Статус відправки	Дата	дії
☐	693f1f52#007b92998a54a2	Anton Yasen	28 800,00 грн	очікування	відправлено	an hour ago	📄 ✓ 👁
☐	693f1f52#007b92998a54a1	Anton Yasen	28 800,00 грн	очікування	доставлено	an hour ago	📄 ✓ 👁
☐	6939ca136d4d6053bd52e025	Anton Yasen	25 000,00 грн	виконано	Очікується	4 days ago	📄 ✓ 👁
☐	6939ca136d4d6053bd52e026	Anton Yasen	45 000,00 грн	виконано	Очікується	4 days ago	📄 ✓ 👁

Рис 3.8 Сторінка відстежування замовлення

Після натискання на товар, користувач переходить на сторінку товару(рис 3.9). Тут він може ознайомитись з назвою товару, його ціною, переглянути його оцінки, категорію та бренд. Обрати кількість товару та додати товар до кошика.

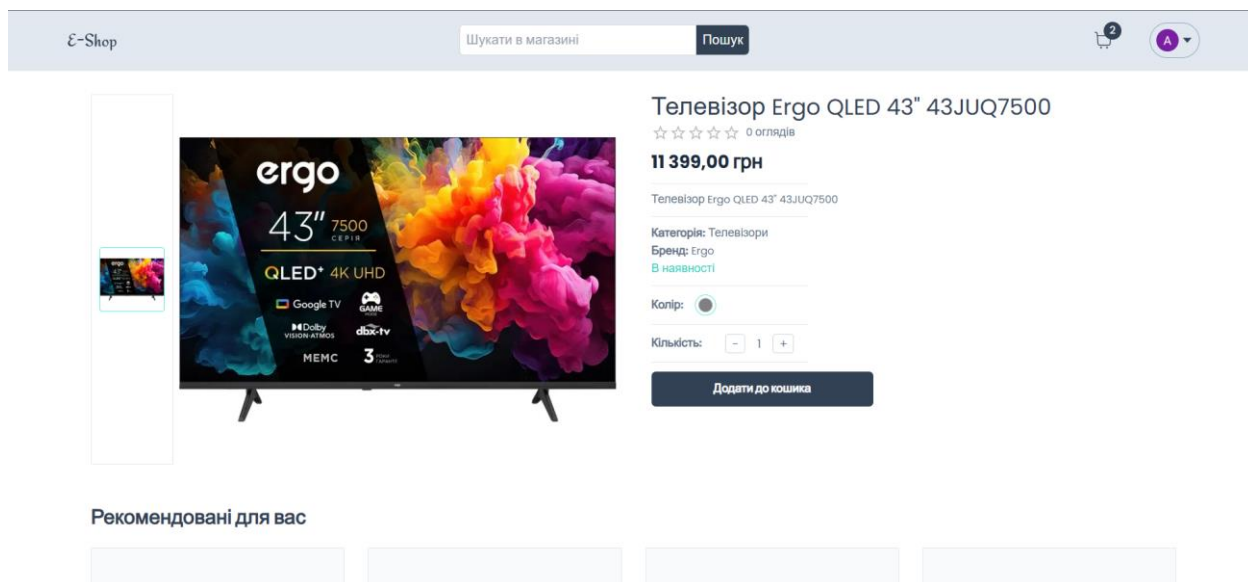


Рис 3.9 Сторінка товару

Нижче на сторінці товару(рис 3.9) відображаються рекомендовані товари з однокової категорії товару. Після перегляду попереднього перегляду інших товарів, товари з їх категорій також будуть з'являтися у рекомендаційному списку товару(рис 3.10).

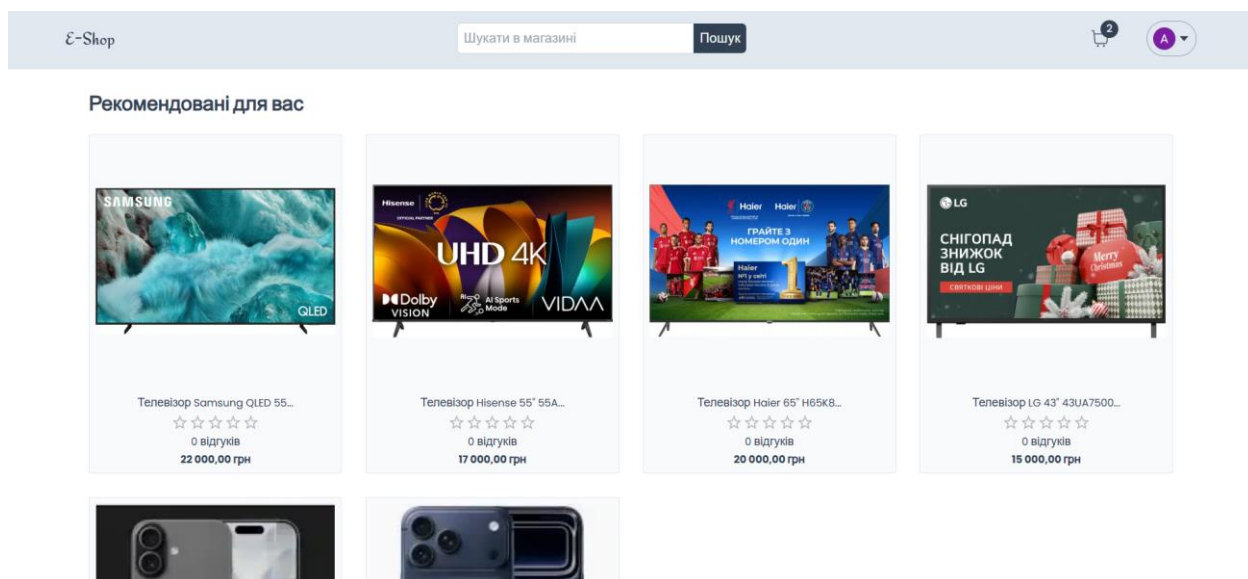


Рис 3.10 Рекомендаційний список товарів на сторінці товару.

Після додавання товару в кошик(рис. 3.9) він потрапляє у кошик. У кошику користувач може переглянути список доданих товарів та відредагувати кількість їх замовлень та оплатити їх. Також у кошику додана система рекомендацій яка відображає які товари часто беруть з обраними товарами користувача(рис 3.11).

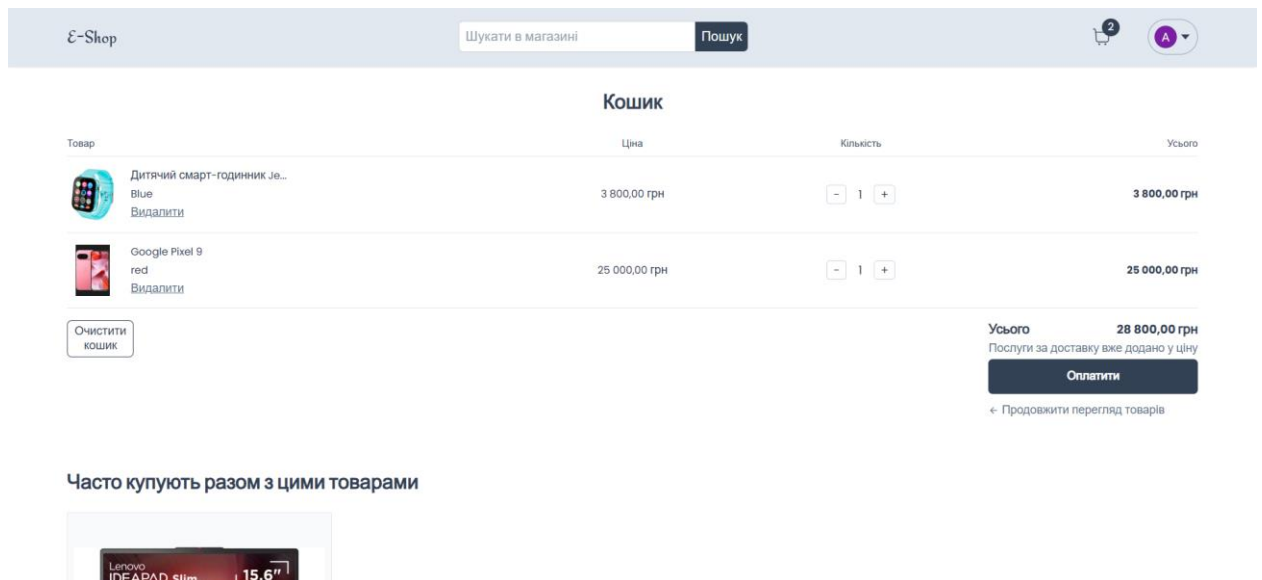


Рис 3.11 сторінка кошику

Після натискання кнопки «Оплатити» користувач перейде на сторінку оплати. Йому потрібно буде ввести необхідні дані для оплати та натиснути кнопку «Оплатити». (рис 3.12).

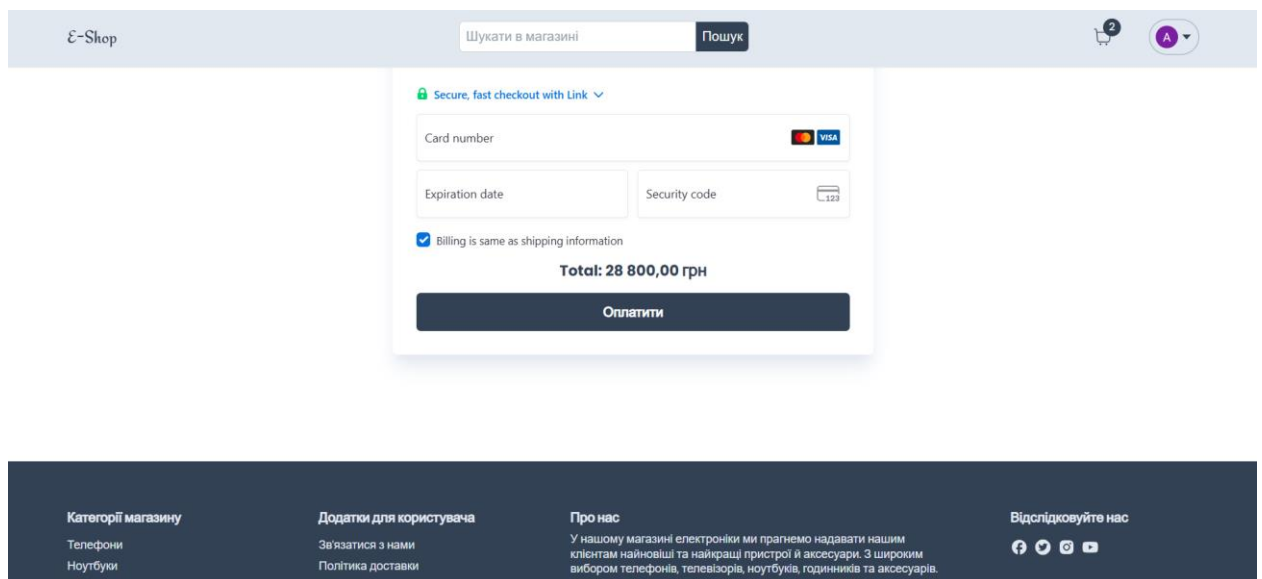


Рис 3.12 Сторінка оплати товару

3.3 Аналіз роботи системи електронної комерції з інтегруванням алгоритмів рекомендації.

Оцінювання ефективності розробленої системи електронної комерції з модулем персоналізованих рекомендацій здійснювалося на основі аналізу її роботи в умовах практичного використання платформи. На відміну від підходів, орієнтованих виключно на формальні офлайн-метрики машинного

навчання, у межах даної роботи основний акцент було зроблено на прикладній цінності рекомендацій, їх здатності адаптуватися до дій користувачів та впливі на загальний користувацький досвід.

Під час аналізу роботи головної сторінки платформи було встановлено, що рекомендаційний модуль формує персоналізовані добірки товарів для авторизованих користувачів з урахуванням їхніх нещодавніх дій. Уже після кількох взаємодій із товарами система починає пропонувати позиції, які логічно пов'язані з поточними інтересами користувача. Це свідчить про коректне накопичення та інтерпретацію поведінкових подій.

Зокрема, у ситуаціях, коли користувач послідовно переглядав товари з однієї категорії, рекомендаційний блок поступово зміщувався у бік цієї категорії або суміжних товарних груп. Подібна поведінка системи підтверджує правильність формування профілю інтересів та ефективність алгоритмів аналізу взаємодій.

Разом із тим, у разі зміни сценарію користувацької поведінки — наприклад, переходу від перегляду смартфонів до ноутбуків або аксесуарів — рекомендаційний список оновлювався відповідним чином. Це демонструє здатність системи враховувати не лише довгострокові вподобання, а й короткострокові інтереси, пов'язані з поточною сесією.

Окремо було проаналізовано роботу рекомендаційного модуля на сторінках окремих товарів, де вплив рекомендацій на рішення про покупку є найбільш відчутним. У цьому контексті використаний підхід поєднує персоналізовані та категорійні рекомендації, що дозволяє зберігати баланс між контекстною доречністю та індивідуальними інтересами користувача.

Практичні спостереження показали, що більшість рекомендованих товарів мають спільну категорійну належність із поточним товаром, що відповідає очікуванням користувачів. Водночас додаткові персоналізовані позиції розширюють перелік рекомендацій і дозволяють звернути увагу на товари, які могли бути пропущені під час стандартного перегляду каталогу.

Такий підхід зменшує відчуття одноманітності рекомендацій, зберігаючи при цьому їх релевантність.

Оцінювання системи здійснювалося на основі різних сценаріїв взаємодії користувачів із платформою, включаючи перегляд товарів, навігацію між категоріями, взаємодію зі сторінками товарів і додавання позицій до кошика. Для аналізу використовувалися дані, які не входили до початкового наповнення системи, що дозволило перевірити здатність рекомендаційного модуля адаптуватися до нових дій користувачів у режимі, наближеному до реального часу.

Окрему увагу приділено аналізу роботи системи в умовах обмеженого обсягу поведінкових даних, зокрема для нових або неавторизованих користувачів. У таких випадках система коректно використовує резервні механізми, формуючи рекомендації на основі популярних товарів або поточного контексту сторінки. Навіть за відсутності персоналізованих даних рекомендаційні блоки залишаються наповненими та корисними для користувача, що дозволяє зменшити негативний вплив проблеми холодного старту.

Крім кількісних спостережень, було виконано якісний аналіз роботи рекомендаційного модуля на окремих прикладах. Так, для користувачів, які переглядали кілька моделей смартфонів, система пропонувала альтернативні пристрої зі схожими характеристиками та ціновим діапазоном. Аналогічно, у випадку перегляду ноутбуків рекомендації включали моделі з близькими технічними параметрами та брендами.

Під час змішаних сценаріїв взаємодії, наприклад одночасного перегляду смартфонів і аксесуарів, система демонструвала здатність поєднувати рекомендації з різних категорій без порушення загальної логіки персоналізації. Це підтверджує ефективність гібридного підходу, використаного у рекомендаційному модулі.

Загалом результати аналізу свідчать про практичну придатність розробленого модуля персоналізованих рекомендацій для використання в

реальній системі електронної комерції. Система стабільно працює за різних сценаріїв взаємодії, коректно реагує на зміну інтересів користувачів і зберігає функціональність навіть за обмеженої кількості вхідних даних.

Найвищий рівень релевантності рекомендацій досягається у випадках активної взаємодії користувача з платформою, коли накопичується достатній обсяг поведінкових подій. Водночас наявність резервних механізмів дозволяє підтримувати прийнятну якість рекомендацій і для нових користувачів. Отримані результати підтверджують обґрунтованість обраної архітектури та алгоритмічного підходу й створюють основу для подальшого вдосконалення системи, зокрема через урахування ваги різних типів подій та часової динаміки інтересів користувачів.

3.4 Тестування та перевірка коректності роботи рекомендаційного модуля.

Завершальним кроком у реалізації програмного продукту стало комплексне тестування розробленої системи електронної комерції з метою перевірки працездатності інтегрованого рекомендаційного механізму. Основна увага на цьому етапі приділялася оцінці загальної стабільності платформи, коректності фіксації дій користувачів та відповідності сформованих рекомендацій різним сценаріям взаємодії з веб-застосунком.

Перевірка проводилася в середовищі, максимально наближеному до реальної експлуатації системи. У ході тестування аналізувалася взаємодія рекомендаційної логіки з ключовими функціональними компонентами платформи, зокрема модулями перегляду товарів, формування кошика та оформлення замовлень. Було перевірено, що події взаємодії користувачів з товарами коректно реєструються, передаються між компонентами системи та надійно зберігаються у базі даних без втрати або спотворення інформації.

Окрему увагу приділено роботі рекомендаційної підсистеми за різних рівнів активності користувачів. Результати тестування засвідчили, що система стабільно формує рекомендації як для користувачів із накопиченою історією

переглядів і покупок, так і для нових або неавторизованих відвідувачів(рис.3.13).

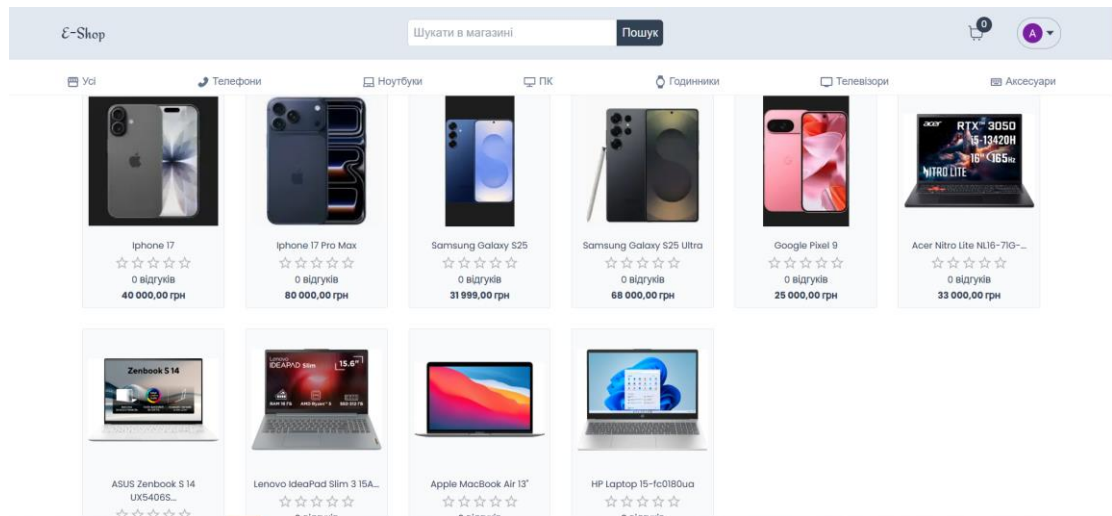


Рис.3.13 Тестування роботи рекомендаційної підсистеми на головній сторінці після тривалого перегляду різних товарів.

У ситуаціях, коли персональних даних недостатньо, рекомендаційний модуль автоматично використовує альтернативні сценарії, орієнтуючись на контекст сторінки або загальну популярність товарів. Це підтверджує коректну реалізацію механізму обробки проблеми холодного старту(рис 3.14).

Рекомендовані товари для вас

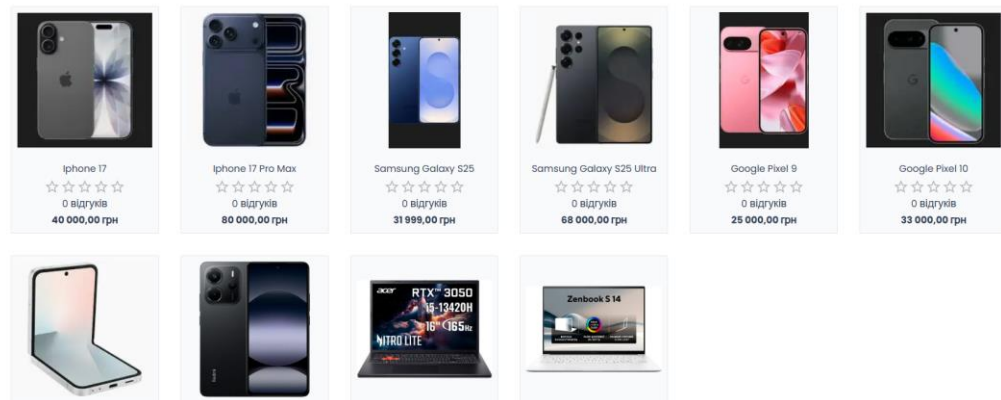


Рис 3.14 Тестування роботи рекомендаційної підсистеми на головній сторінці для нового користувача який не виконував ніяких дій.

Також було перевірено роботу рекомендацій у межах сторінки конкретного товару та кошика. Під час тестування встановлено, що на сторінці товару система пропонує позиції, логічно пов'язані з поточним вибором користувача, водночас доповнюючи їх персоналізованими варіантами на основі попередніх дій. У кошику рекомендації коректно відображають товари,

які зазвичай купуються разом, що створює додаткові умови для завершення замовлення та збільшення середньої вартості покупки.

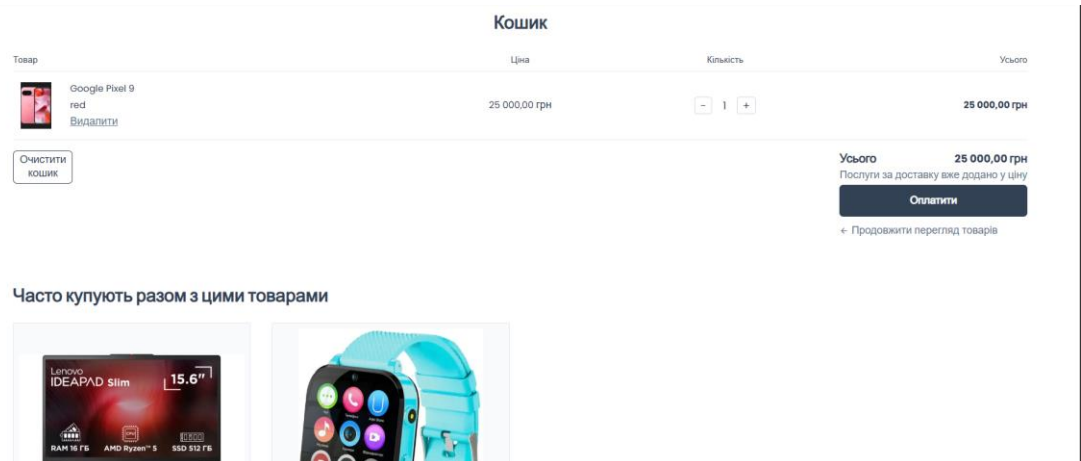


Рис 3.15 Тестування рекомендаційного модуля “часто купують разом” від лица нового користувача який не створював замовлень.

Технічна частина тестування включала перевірку виконання запитів до бази даних і обробки результатів рекомендаційних обчислень. У процесі перевірки було встановлено, що використання ORM Prisma забезпечує надійну та стабільну взаємодію з базою даних, а виконання рекомендаційних запитів не призводить до критичного навантаження на систему. Навіть за зростання кількості зафіксованих поведінкових подій час формування рекомендацій залишається прийнятним для інтерактивного веб-застосунку.

У ході тестування також було виявлено незначні технічні неточності, пов’язані з передачею параметрів між клієнтською та серверною частинами системи. Після їх усунення загальна узгодженість роботи компонентів платформи була покращена, а рекомендаційний модуль почав функціонувати стабільно у всіх перевірених сценаріях.

Отримані результати тестування свідчать про те, що розроблена система електронної комерції відповідає вимогам щодо надійності, коректності та практичної придатності. Реалізований рекомендаційний механізм демонструє передбачувану поведінку, адекватно реагує на дії користувачів і може бути використаний у реальних умовах без необхідності залучення зовнішніх сервісів або складних машинних моделей.

Висновки до розділу 3

У третьому розділі дипломної роботи було зосереджено увагу на практичній реалізації системи електронної комерції з модулем персоналізованих рекомендацій, а також на аналізі результатів її функціонування. Основним завданням цього розділу стало впровадження рекомендаційної логіки у веб-застосунок, перевірка коректності її інтеграції з іншими компонентами платформи та оцінювання ефективності роботи системи в умовах наближених до реального використання.

У підрозділі 3.1 розглянуто реалізацію ключових елементів рекомендаційного модуля. Запропоноване рішення ґрунтується на аналізі поведінкових подій користувачів і не потребує використання зовнішніх ML-платформ або сторонніх сервісів. Поєднання алгоритмів item-to-item колаборативної фільтрації, формування профілю інтересів користувача, контентної фільтрації за категоріями та аналізу спільних покупок дозволило реалізувати гібридний механізм рекомендацій, здатний підвищувати релевантність товарних пропозицій у різних сценаріях взаємодії з платформою.

Важливою особливістю реалізованого підходу є модульна побудова рекомендаційної логіки та її інтеграція з бізнес-логікою інтернет-магазину за допомогою ORM Prisma. Така організація спрощує масштабування системи, полегшує подальше вдосконалення алгоритмів персоналізації та дозволяє вносити зміни без необхідності перегляду базової архітектури застосунку.

У підрозділі 3.2 проаналізовано структуру користувацького інтерфейсу веб-застосунку. Описано основні сторінки платформи, зокрема головну сторінку, сторінки автентифікації, перегляду товарів, кошика та адміністративного керування. Показано, що рекомендаційний модуль інтегровано в різні частини інтерфейсу, включно з головною сторінкою, сторінкою товару та кошиком, що забезпечує послідовну персоналізацію користувацького досвіду на всіх етапах роботи з платформою.

У підрозділі 3.3 виконано аналіз роботи системи з урахуванням інтегрованих рекомендаційних алгоритмів. Результати тестування засвідчили, що система коректно реагує на зміну поведінки користувачів, враховує як поточні, так і накопичені інтереси, а також стабільно функціонує за умов обмеженого обсягу поведінкових даних. Застосування резервних механізмів формування рекомендацій дозволяє зменшити вплив проблеми холодного старту та забезпечити працездатність рекомендаційного модуля для нових користувачів.

Загалом результати, отримані у третьому розділі, свідчать про те, що розроблене програмне забезпечення є цілісним, логічно організованим і придатним для використання в реальній системі електронної комерції. Реалізований рекомендаційний модуль підвищує рівень персоналізації платформи, позитивно впливає на користувацький досвід і створює умови для зростання залученості та конверсії. Отримані результати можуть слугувати основою для подальшого розвитку системи, зокрема шляхом розширення набору алгоритмів персоналізації та поглиблення аналізу поведінкових даних.

ВИСНОВКИ

У межах дипломної роботи було виконано розробку та дослідження системи електронної комерції з модулем персоналізованих рекомендацій, спрямованої на покращення взаємодії користувачів із товарним каталогом і підвищення ефективності роботи платформи загалом.

У процесі дослідження проаналізовано сучасні підходи до персоналізації в електронній комерції та визначено їхню роль у формуванні користувацького досвіду. Встановлено, що рекомендаційні механізми є важливим інструментом зменшення інформаційного перевантаження, підвищення залученості користувачів і стимулювання повторних покупок, особливо в умовах великої кількості товарів і динамічної поведінки споживачів.

У першому розділі здійснено аналіз основних типів рекомендаційних систем, зокрема колаборативних, контентних і гібридних підходів. Розглянуто їхні переваги та обмеження, такі як проблема холодного старту, залежність від обсягу даних і складність масштабування. На основі проведеного аналізу зроблено висновок про доцільність використання гібридного підходу, який дозволяє поєднати сильні сторони різних алгоритмів.

Другий розділ було присвячено проєктуванню архітектури системи. Обґрунтовано вибір технологічного стеку, що включає TypeScript, Node.js, Next.js, MongoDB та ORM Prisma, а також розроблено модульну архітектуру з чітким розмежуванням відповідальностей між компонентами. Спроектована структура даних орієнтована на збереження та аналіз поведінкових подій користувачів, що створює основу для ефективної персоналізації рекомендацій.

У третьому розділі реалізовано програмне забезпечення системи електронної комерції та власний рекомендаційний модуль без використання зовнішніх ML-сервісів. Реалізований механізм поєднує item-to-item колаборативну фільтрацію, аналіз інтересів користувачів, контентні рекомендації та алгоритм спільних покупок. Результати тестування показали, що система коректно реагує на зміну поведінки користувачів, враховує як

короткострокові, так і накопичені інтереси та стабільно працює навіть за обмеженого обсягу даних.

Загалом отримані результати свідчать про те, що розроблена система є працездатною, логічно організованою та придатною для використання в реальних умовах електронної комерції. Запропонований рекомендаційний модуль забезпечує релевантні товарні пропозиції, покращує користувацький досвід і створює передумови для зростання конверсії та залученості користувачів.

Практична цінність роботи полягає у можливості застосування розробленої архітектури та рекомендаційного модуля в інтернет-магазинах різного масштабу. Запропоноване рішення також має потенціал для подальшого розвитку, зокрема шляхом урахування ваги різних типів поведінкових подій, часової динаміки інтересів користувачів або впровадження більш складних моделей аналізу даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ясен А.Є., Годлевський Ю.О інтеграція хмарних рішень у сучасну розробку програмного забезпечення. VIII Всеукраїнської науково-технічної конференції "Комп'ютерні технології: інновації, проблеми, рішення", 02-03 грудня 2025 року. Житомир: "Житомирська політехніка", 2025.
2. Statista. Retail e-commerce sales worldwide from 2014 to 2027 (in trillion U.S. dollars). 2023. URL: <https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/> .
3. Promodo. Український e-commerce 2025: дослідження та підсумки за 1 півріччя 2025. 2025. URL: <https://www.promodo.ua/ukrayinskiy-ecommerce-2025-1> .
4. Rejoiner. The Amazon Recommendations Secret to Selling More Online: 35% of Amazon.com's revenue is generated by its recommendation engine. 2025. URL: <https://www.rejoiner.com/resources/amazon-recommendations-secret-selling-online> .
5. Business Insider. Netflix's Recommendation Engine Drives 75% of Viewership. 2012. URL: <https://www.businessinsider.com/netflixs-recommendation-engine-drives-75-of-viewership-2012-4> .
6. McKinsey & Company. What is personalization? 2023. URL: <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-personalization> .
7. Peng, M., Qin, Y., Zhang, W. та ін. How Does Information Overload Affect Consumers' Online Decision Processes?. 2021. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8567038/> .
8. Adomavicius, G., Tuzhilin, A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering. 2005. Т. 17(6). С. 734–749. URL: <https://doi.org/10.1109/TKDE.2005.99> .
9. Sarwar, B., Karypis, G., Konstan, J., Riedl, J. Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th International Conference on

World Wide Web (WWW). 2001. C. 285–295. URL: <https://doi.org/10.1145/371920.372071> .

10. Burke, R. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction. 2002. T. 12. C. 331–370. URL: <https://doi.org/10.1023/A:1021240730564> .

11. Schafer, J. B., Konstan, J. A., Riedl, J. E-Commerce Recommendation Applications. Data Mining and Knowledge Discovery. 2001. T. 5. C. 115–153. URL: <https://doi.org/10.1023/A:1009804230409> .

12. Herlocker, J. L., Konstan, J. A., Terveen, L. G., Riedl, J. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems. 2004. T. 22(1). C. 5–53. URL: <https://doi.org/10.1145/963770.963772> .

13. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A. Recommender systems survey. Knowledge-Based Systems. 2013. T. 46. C. 109–132. URL: <https://doi.org/10.1016/j.knosys.2013.03.012> .

14. Ricci, F., Rokach, L., Shapira, B. (eds.). Recommender Systems Handbook (2nd ed.). Springer. 2015. URL: <https://doi.org/10.1007/978-1-4899-7637-6> .

15. Aggarwal, C. C. Recommender Systems: The Textbook. Springer. 2016. URL: <https://doi.org/10.1007/978-3-319-29659-3> .

16. Koren, Y., Bell, R., Volinsky, C. Matrix factorization techniques for recommender systems. Computer. 2009. T. 42(8). C. 30–37. URL: <https://doi.org/10.1109/MC.2009.263> .

17. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L. BPR: Bayesian Personalized Ranking from Implicit Feedback. UAI. 2009. URL: <https://arxiv.org/abs/1205.2618> .

18. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S. Neural Collaborative Filtering. Proceedings of WWW. 2017. URL: <https://arxiv.org/abs/1708.05031> .

19. Zhang, S., Yao, L., Sun, A., Tay, Y. Deep Learning based Recommender System: A Survey and New Perspectives. ACM Computing Surveys. 2019. T. 52(1). Article 5. URL: <https://doi.org/10.1145/3285029> .

20. Konstan, J. A., Riedl, J. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*. 2012. T. 22. C. 101–123. URL: <https://doi.org/10.1007/s11257-011-9112-x> .
21. Beel, J., Langer, S., Genzmehr, M., et al. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*. 2016. T. 17. C. 305–338. URL: <https://doi.org/10.1007/s00799-015-0156-0> .
22. Beel, J., et al. Research paper recommender system evaluation: a quantitative literature survey. *Proceedings of RecSys 2013*. 2013. URL: <https://doi.org/10.1145/2532508.2532512> .
23. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D. Session-based Recommendations with Recurrent Neural Networks. *ICLR*. 2016. URL: <https://arxiv.org/abs/1511.06939> .
24. Linden, G., Smith, B., York, J. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing*. 2003. T. 7(1). C. 76–80. URL: <https://doi.org/10.1109/MIC.2003.1167344> .
25. Lika, B., Kolomvatsos, K., Hadjiefthymiades, S. Facing the cold start problem in recommender systems. *Expert Systems with Applications*. 2014. T. 41(4). C. 2065–2073. URL: <https://doi.org/10.1016/j.eswa.2013.09.005> .
26. TypeScript Documentation. *The TypeScript Handbook*. Microsoft. URL: <https://www.typescriptlang.org/docs/handbook/intro.html> .
27. Casciaro, M., Mammino, L. *Node.js Design Patterns* (3rd ed.). Packt Publishing. 2020. ISBN: 978-1839214110 .
28. Next.js Documentation. Vercel. URL: <https://nextjs.org/docs> .
29. Brazil, E., Chodorow, K. *MongoDB: The Definitive Guide* (3rd ed.). O'Reilly Media. 2019. ISBN: 978-1491954461 .
30. Prisma Documentation: Next-generation ORM for Node.js. Prisma. URL: <https://www.prisma.io/docs> .
31. Fielding, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. PhD dissertation, University of California, Irvine. 2000.

URL: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

.

32. Quadrana, M., Karatzoglou, A., Hidasi, B., Cremonesi, P. Sequence-Aware Recommender Systems. ACM Computing Surveys. 2018. URL: <https://doi.org/10.1145/3190616> .

33. Kang, W.-C., McAuley, J. Self-Attentive Sequential Recommendation. IEEE ICDM. 2018. URL: <https://arxiv.org/abs/1808.09781> .

34. Covington, P., Adams, J., Sargin, E. Deep Neural Networks for YouTube Recommendations. Proceedings of RecSys 2016. 2016. URL: <https://doi.org/10.1145/2959100.2959190>

ДОДАТКИ

ДОДАТОК А

Лістинг що відповідає роботі системі рекомендації

```
import { prisma } from "@libs/prismadb";

async function getItemToItem(productId: string) {
  const events = await prisma.userEvent.findMany({
    where: { productId },
  });

  const users = events.map((e) => e.userId);
  if (users.length === 0) return [];

  const relatedEvents = await prisma.userEvent.findMany({
    where: {
      userId: { in: users },
      productId: { not: productId },
    },
  });

  const freq: Record<string, number> = {};
  relatedEvents.forEach((ev) => {
    freq[ev.productId] = (freq[ev.productId] || 0) + 1;
  });

  const sorted = Object.entries(freq)
    .sort((a, b) => b[1] - a[1])
    .map(([id]) => id);

  return prisma.product.findMany({
    where: { id: { in: sorted } },
    take: 10,
  });
}
```

```
}
```

```
async function getUserInterestProfile(userId: string) {  
  const events = await prisma.userEvent.findMany({  
    where: { userId },  
    orderBy: { createdAt: "desc" },  
    take: 50,  
  });
```

```
  if (events.length === 0) return [];
```

```
  const productIds = events.map((e) => e.productId);
```

```
  const products = await prisma.product.findMany({  
    where: { id: { in: productIds } },  
  });
```

```
  const freq: Record<string, number> = {};
```

```
  products.forEach((p) => {  
    freq[p.category] = (freq[p.category] || 0) + 1;  
  });
```

```
  return Object.entries(freq)  
    .sort((a, b) => b[1] - a[1])  
    .map(([cat]) => cat);  
}
```

```
async function getSimilarByCategory(category: string) {  
  return prisma.product.findMany({  
    where: { category },  
    take: 10,
```

```

    });
}

export async function recommendForUser(userId: string) {
  const events = await prisma.userEvent.findMany({
    where: { userId },
    orderBy: { createdAt: "desc" },
    take: 10,
  });

  if (events.length === 0) {
    return prisma.product.findMany({ take: 10 });
  }

  const lastViewedId = events[0].productId;

  const itemBased = await getItemToItem(lastViewedId);

  const categories = await getUserInterestProfile(userId);
  let interestBased: any[] = [];

  if (categories.length > 0) {
    interestBased = await prisma.product.findMany({
      where: { category: { in: categories.slice(0, 2) } },
      take: 10,
    });
  }

  const lastProduct = await prisma.product.findUnique({
    where: { id: lastViewedId },
  });

```

```

let similarCategory: any[] = [];
if (lastProduct) {
  similarCategory = await prisma.product.findMany({
    where: {
      category: lastProduct.category,
      id: { not: lastViewedId },
    },
    take: 5,
  });
}
const popular = await prisma.product.findMany({
  take: 10,
  orderBy: { price: "desc" },
});

const combined = [
  ...itemBased.slice(0, 5),
  ...interestBased.slice(0, 5),
  ...similarCategory.slice(0, 3),
  ...popular.slice(0, 3),
];

const unique = Array.from(new Map(combined.map((p) => [p.id, p])).values());

return unique.slice(0, 10);
}

export async function getFrequentlyBoughtTogether(productId: string) {
  const orders = await prisma.order.findMany({
    where: {

```

```

    products: { some: { id: productId } },
  },
});

if (orders.length === 0) return [];

const freq: Record<string, number> = {};

for (const order of orders) {
  for (const item of order.products) {
    if (item.id === productId) continue;
    freq[item.id] = (freq[item.id] || 0) + 1;
  }
}

const sortedIds = Object.entries(freq)
  .sort((a, b) => b[1] - a[1])
  .map(([id]) => id);

return prisma.product.findMany({
  where: { id: { in: sortedIds } },
  take: 6,
});
}

export async function recommendForUserOnProductPage(
  userId: string,
  productId: string
) {
  const personal = await recommendForUser(userId);

  const current = await prisma.product.findUnique({

```

```
    where: { id: productId },
  });
```

```
let sameCategory: any[] = [];
```

```
if (current) {
  sameCategory = await prisma.product.findMany({
    where: {
      category: current.category,
      id: { not: productId },
    },
    take: 20,
  });
}
```

```
const topCategory = sameCategory.slice(0, 4);
```

```
const topPersonal = personal.slice(0, 2);
```

```
const combined = [...topCategory, ...topPersonal];
```

```
const uniq = Array.from(new Map(combined.map((p) => [p.id, p])).values());
```

```
return uniq.slice(0, 6);
```

```
}
```

Лістинг що відповідає відображенню рекомендацій на сторінці товару
"use client";

```
import { useEffect, useState } from "react";
```

```
import ProductCard from "../components/products/ProductCard";
```

```
export default function CartRecommendations({
```

```

    productIds,
  }: {
    productIds: string[];
  }) {
    const [products, setProducts] = useState<any[]>([]);
    const [loading, setLoading] = useState(true);

    useEffect(() => {
      async function load() {
        try {
          const res = await fetch("/api/recommend-for-cart", {
            method: "POST",
            headers: { "Content-Type": "application/json" },
            body: JSON.stringify({ productIds }),
          });

          const data = await res.json();
          setProducts(data);
        } catch (e) {
          console.error(e);
        } finally {
          setLoading(false);
        }
      }

      load();
    }, [productIds]);

    if (loading || products.length === 0) return null;
  }
}

```

```
return (  
  <div className="mt-14">  
    <h2 className="text-2xl font-bold mb-6">  
      Часто покупают разом с этими товарами  
    </h2>  
  
    <div className="grid grid-cols-2 sm:grid-cols-3 lg:grid-cols-4 gap-8">  
      {products.map(p => (  
        <ProductCard key={p.id} data={p} />  
      ))}  
    </div>  
  </div>  
);  
}
```