

ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО «ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
Міжрегіональна академія управління персоналом

Факультет комп'ютерно-інформаційних технологій
Кафедра комп'ютерних інформаційних систем і технологій

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**Тема «Інтернет магазин із функцією рекомендацій для клієнтів на
базі штучного інтелекту»**

Виконала студентка групи ІК-9-22-Б1ПЗ(4.0д)

Кашиця Д.О.

Керівник: викл.

Яременко Д.М.

Рецензент:

Допущено до захисту

Завідувач кафедри

(підпис)

д.е.н., професор Кавун С.В.

Київ, 2026

РЕФЕРАТ

Пояснювальна записка: 63 с., 13 рис., 2 додатки, 22 джерела.

Мета роботи: створення мобільного інтернет-магазину з інтегрованою системою персоналізованих рекомендацій, здатною адаптувати товарний вміст під індивідуальні вподобання користувача.

Об'єкт дослідження: взаємодія користувачів з електронною комерцією в умовах персоналізованих сервісів на основі штучного інтелекту.

Предмет дослідження: методи реалізації гібридної системи персоналізованих рекомендацій на основі штучного інтелекту.

У кваліфікаційній роботі розглянуто процес проєктування та реалізації мобільного застосунку для електронної комерції, в якому реалізована персоналізація контенту за допомогою штучного інтелекту. Застосунок побудовано на базі Flutter, з використанням Firebase як основної хмарної платформи для зберігання даних, автентифікації та керування контентом. Особливу увагу приділено інтеграції мовної моделі GPT-4o від OpenAI, яка через API дозволяє генерувати релевантні рекомендації товарів на основі активності користувача.

Система включає в себе низку ключових компонентів: модулі авторизації, головну сторінку з динамічним контентом, окремі екрани для перегляду товару, рекомендацій, історії покупок, профілю користувача та оформлення замовлення. Основою логіки персоналізації виступає запис дій користувача (перегляди, покупки) у Firestore та формування запитів до GPT-4o для генерації рекомендацій, які згодом фільтруються та виводяться на головному екрані застосунку.

У межах дослідження здійснено аналіз сучасних архітектур систем рекомендацій, обґрунтовано вибір гібридного підходу, описано структуру бази даних та реалізовано повний цикл взаємодії між клієнтським інтерфейсом та

бекендом. Проведено базове тестування функціоналу, що підтвердило стабільність роботи та відповідність системи поставленим вимогам.

Ключові слова: ІНТЕРНЕТ-МАГАЗИН, РЕКОМЕНДАЦІЇ, ШТУЧНИЙ ІНТЕЛЕКТ, ГІБРИДНА МОДЕЛЬ, ПЕРСОНАЛІЗАЦІЯ, АІ, МАШИННЕ НАВЧАННЯ, Е-COMMERCE, СИСТЕМА РЕКОМЕНДАЦІЙ, КОРИСТУВАЦЬКИЙ ДОСВІД.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. АКТУАЛЬНІСТЬ ТЕМИ. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. ПОСТАНОВКА ЗАДАЧІ.....	9
1.1 Актуальність теми	9
1.2 Аналіз предметної області.....	11
1.3 Постановка задачі.....	14
РОЗДІЛ 2. АНАЛІЗ ТЕХНОЛОГІЙ. ВИБІР ШІ-МОДЕЛІ. ПОСТАНОВКА ЗАДАЧІ.....	17
2.1 Аналіз сучасних платформ.....	17
2.2 Дослідження моделей нейромереж.....	20
2.3.Проектування архітектури системи.....	23
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ.....	28
3.1 Архітектура системи.....	28
3.2 Розробка бази даних та структури зберігання.....	29
3.3 Реалізація гібридної системи рекомендацій.....	34
3.4 Проектування та реалізація інтерфейсу користувача (UI/UX)....	40
3.5 Тестування та відлагодження.....	43
ВИСНОВКИ.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48
ДОДАТОК А.....	51
ДОДАТОК Б.....	
.57	

ВСТУП

Сучасна електронна комерція стрімко трансформується під впливом новітніх інформаційних технологій, зокрема рішень на основі штучного інтелекту [1]. Зростання обсягів даних, динаміка користувацьких вподобань та конкуренція серед онлайн-платформ зумовлюють необхідність створення більш адаптивних та персоналізованих сервісів [2, 3]. У цьому контексті системи рекомендацій відіграють ключову роль, адже дозволяють не лише покращити досвід користувача, а й значно підвищити ефективність бізнес-процесів [4]

Актуальність теми обумовлена швидким розвитком технологій персоналізації у сфері e-commerce. За умов зростаючих очікувань користувачів та необхідності підвищення конверсії у продажах, застосування гібридних рекомендаційних систем на основі штучного інтелекту набуває дедалі більшого значення [5]. Особливої ваги набувають підходи, які дозволяють враховувати як історичну поведінку покупців, так і особливості самих товарів, формуючи релевантні пропозиції в реальному часі. Це дозволяє інтернет-магазинам не лише утримувати увагу клієнтів, але й суттєво знижувати витрати на маркетинг та покращувати цільове націлювання [6].

На відміну від традиційних рекомендаційних систем, сучасні інтелектуальні моделі, такі як трансформери, здатні обробляти дані в режимі реального часу, працювати з мультимодальними сигналами та взаємодіяти з користувачем у формі природної розмови. Це відкриває нові можливості для інтеграції таких моделей у мобільні додатки електронної комерції. Водночас, актуальність даного дослідження полягає і в тому, що воно орієнтоване на розв'язання практичної задачі – створення прототипу інтернет-магазину, який не лише виконує стандартні функції платформи онлайн-продажів, а й містить модуль рекомендацій на базі передових моделей штучного інтелекту.

Тематика дипломної роботи присвячена розробці інтернет-магазину з інтегрованим модулем рекомендацій, побудованим на базі гібридного підходу до персоналізації. Такий підхід передбачає поєднання контентного аналізу товарів із

врахуванням поведінкових патернів користувачів, що дозволяє досягти високої точності та релевантності пропозицій [7]. Особливістю системи є використання моделі GPT-4o – однієї з найсучасніших трансформерних архітектур, здатної ефективно обробляти складні запити, враховувати контекст і генерувати персоналізовані рекомендації на основі багатовимірних сигналів.

Дослідження спрямоване на вивчення методів впровадження інтелектуальних алгоритмів у структуру веб-додатку та створення повноцінного функціонального рішення, яке відповідає вимогам сучасного цифрового ринку. Система, що розробляється, демонструє приклад інтеграції новітніх AI-технологій у прикладну архітектуру мобільного застосунку з метою формування динамічного, персоналізованого та зручного користувацького досвіду.

РОЗДІЛ 1. АКТУАЛЬНІСТЬ ТЕМИ. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. ПОСТАНОВКА ЗАДАЧІ.

1.1 Актуальність теми

Сучасне впровадження нейронних мереж у бізнес-процеси, зокрема в електронну торгівлю, відкриває чимало нових можливостей для покращення взаємодії між користувачами та онлайн-платформами. Інтелектуальні алгоритми, що лежать в основі цих мереж, дозволяють не просто аналізувати поверхневі дії користувача, а й будувати глибокі поведінкові моделі, які враховують індивідуальні вподобання, часову динаміку дій, попередній досвід і контекст поточних пошуків. Здатність таких систем виявляти приховані закономірності у великому потоці даних сприяє формуванню персоналізованих рекомендацій у режимі реального часу. Наприклад, система може виявити, що користувач переглядає товари певного кольору або стилю, і автоматично адаптувати стрічку товарів відповідно до цих вподобань. Це значно знижує когнітивне навантаження на покупця, мінімізує кількість неактуальної інформації та створює комфортне середовище для здійснення покупок.

Важливою перевагою використання нейромереж є їхня здатність до безперервного навчання. У процесі взаємодії з користувачами системи самостійно вдосконалюють свої прогнози, накопичуючи новий досвід і модифікуючи внутрішні представлення даних. Це дозволяє платформам оперативно реагувати на зміну трендів, поведінкових сценаріїв і зовнішніх чинників (наприклад, сезонних коливань попиту). Така адаптивність робить електронну комерцію більш чутливою до потреб ринку й здатною задовольняти запити користувачів з високою точністю навіть у нестабільних умовах.

Окрім полегшення життя користувачів, нейронні мережі мають ще одну важливу функцію – вони виступають як інструмент стратегічного управління. Завдяки аналітиці, яка базується на інтелектуальній обробці даних, компанії отримують змогу приймати зважені рішення щодо формування асортименту,

динамічного ціноутворення, управління складськими запасами тощо. Наприклад, якщо система фіксує зростання інтересу до певного типу товарів серед певного сегмента клієнтів, це може слугувати підґрунтям для оперативного коригування закупівельної політики або маркетингової стратегії. Таким чином, штучний інтелект виконує функцію не лише рекомендаційного, а й прогностичного модуля.

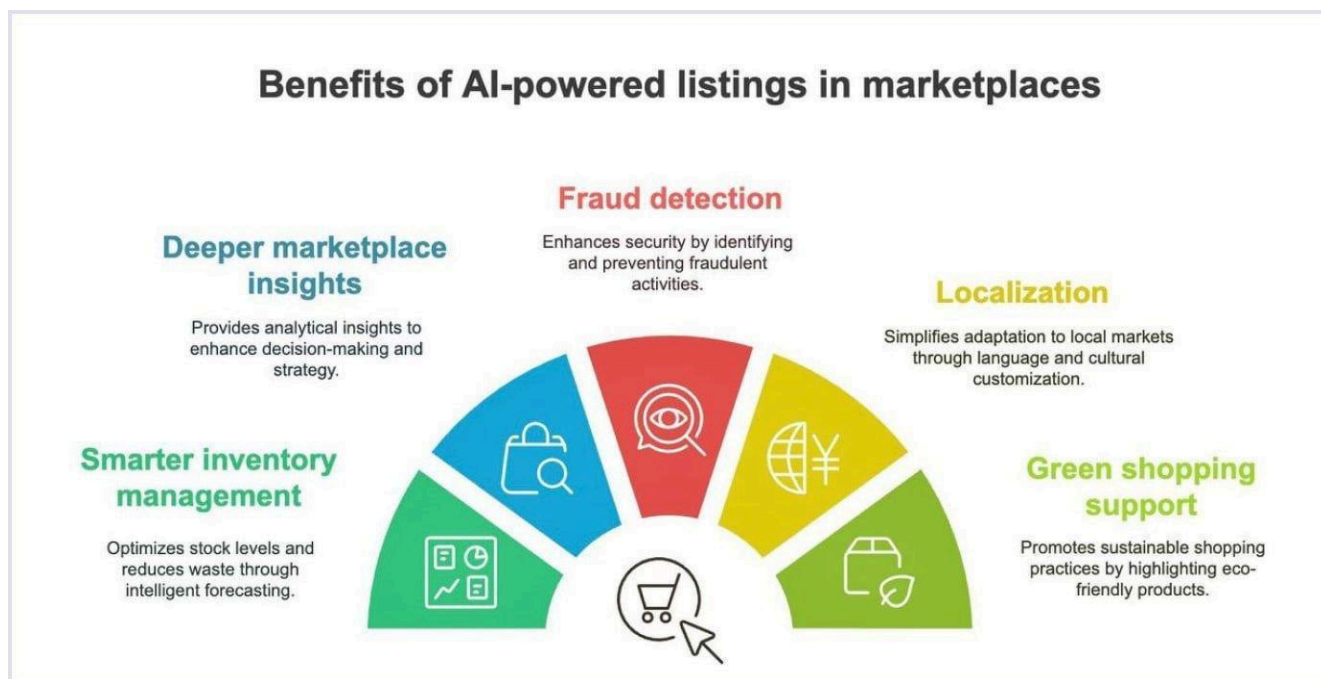


Рисунок 1.1 - переваги використання AI у маркетплейсах

На макрорівні такі технології сприяють трансформації електронної комерції з простої транзакційної системи в складну, соціально інтегровану екосистему. По-перше, нейронні мережі знижують бар'єр входу для нових користувачів, які, завдяки персоналізованому підходу, швидше адаптуються до цифрового середовища. По-друге, платформи стають більш інклюзивними, оскільки можуть пропонувати зручні рішення для людей з різними цифровими навичками, віком або мовними вподобаннями. По-третє, внаслідок підвищеної ефективності бізнес-процесів відбувається економія ресурсів – як людських, так і матеріальних – що позитивно впливає на стійкість усієї галузі.

Загалом, впровадження нейромереж в електронну торгівлю не обмежується лише підвищенням зручності або ефективності. Йдеться про фундаментальні зміни у способі мислення про взаємодію між бізнесом і споживачем, про перехід до нової парадигми, де дані, інтелект і персоналізація є ключовими рушіями

цифрової трансформації. У цьому контексті штучний інтелект стає не просто інструментом, а каталізатором еволюції цифрової комерції в сторону більш гнучкої, етичної й клієнтоорієнтованої моделі.

1.2 Аналіз предметної області

У сучасному цифровому середовищі, де конкуренція між електронними платформами невпинно зростає, персоналізація користувацького досвіду виходить на перший план [8]. Для забезпечення високого рівня лояльності клієнтів, збільшення коефіцієнта конверсії та підвищення загальної конкурентоспроможності, компанії все частіше звертаються до систем рекомендацій, побудованих на основі штучного інтелекту [9]. Саме такі системи дозволяють не просто відображати користувачеві релевантний контент, а й передбачати його наміри ще до моменту, коли він сам усвідомить свої потреби.

Штучний інтелект у сфері e-commerce стає не допоміжним, а центральним інструментом цифрової взаємодії. Завдяки розвитку глибокого навчання, обробки природної мови (NLP), аналізу зображень та трансформерів, сучасні алгоритми здатні формувати повноцінні когнітивні моделі користувача: розпізнавати його уподобання, стиль споживання, емоційний фон і навіть формувати передбачення щодо потенційних потреб.

Amazon по праву вважається світовим лідером з упровадження технологій штучного інтелекту у сфері електронної торгівлі [10]. Його рекомендаційна система побудована за гібридним принципом, що поєднує як колаборативну, так і контентну фільтрацію [11]. На практиці це означає, що алгоритм здатен не лише аналізувати особисту історію покупок користувача, але й порівнювати її з поведінковими шаблонами інших клієнтів, формуючи на цій основі персоналізовані пропозиції. Додатково Amazon активно впроваджує глибокі нейронні мережі для обробки голосових запитів через Alexa, аналізу текстових відгуків та розпізнавання зображень, що дозволяє забезпечити мультиканальний підхід до персоналізації.

Іншим яскравим прикладом є платформа Rozetka – один із лідерів українського ринку e-commerce. Попри більш обмежений обсяг впроваджених AI-технологій, компанія вже сьогодні активно використовує машинне навчання для оптимізації пошукових механізмів, автоматичного заповнення форм і формування блоків рекомендацій на основі історії переглядів і покупок. Інтерфейс платформи регулярно оновлюється з урахуванням A/B-тестування, що дозволяє адаптувати контент під поведінку конкретних сегментів користувачів.

Ще одним масштабним гравцем є AliExpress, частина екосистеми Alibaba Group, яка орієнтується на глибоку поведінкову аналітику [12]. Її рекомендаційна система обробляє величезні масиви даних про кліки, перегляди, час перебування на сторінці, геолокацію та інші сигнали. Особливої уваги заслуговує реалізований функціонал пошуку за зображенням, що базується на згорткових нейронних мережах (CNN), які здатні ідентифікувати візуальні патерни та знаходити аналогічні товари у каталозі [13]. Компанія також використовує AI для прогнозування попиту, управління логістикою та персоналізації цінових пропозицій.

Окрему категорію представляють платформи типу TikTok, де рекомендаційна логіка є не допоміжним, а центральним елементом взаємодії [13, 14]. Її алгоритми, побудовані на основі трансформерів, аналізують сотні змінних – від тривалості перегляду до реакції на контент, звукову доріжку, кольорову гаму, стиль монтажу, ключові слова в описі відео та навіть емоції, що їх викликає контент [16, 17]. Алгоритм безперервно навчається, адаптуючи стрічку в реальному часі до вподобань конкретного користувача. Це дозволяє досягти неймовірного рівня залучення, коли кожен перегляд – це вже потенційна точка контакту з персоналізованим контентом.

Таким чином, аналіз провідних платформ демонструє, що застосування нейромереж та технологій штучного інтелекту стало визначальним чинником ефективності в електронній комерції. Вони не лише підвищують конверсію та утримання користувачів, а й формують нову парадигму цифрового споживання –

де кожен клієнт отримує саме той контент, який відповідає його потребам, цінностям і поточному стану [18, 19].



Рисунок 1.2 - Типовий процес доставки товарів із платформи Alibaba до Amazon FBA

Водночас слід враховувати, що ефективність рекомендаційної системи не обмежується лише якістю алгоритмів. Велику роль відіграє і логістична інфраструктура, яка забезпечує фізичне втілення цифрових рішень. Наприклад, у випадку співпраці між Alibaba та Amazon через модель FBA (Fulfillment by Amazon), критично важливо, щоб система рекомендацій синхронізувалась із

запасами на складах, враховувала час доставки, можливості перехресних продажів та навіть сезонні коливання у попиті. Така інтеграція цифрових і фізичних компонентів дозволяє досягти максимальної ефективності – коли не лише правильний товар пропонується правильному клієнту, але й доставляється швидко, зручно та з мінімальними витратами для продавця.

Більше того, із розвитком міжнародної торгівлі та глобальних ланцюгів постачання виникає потреба у ще тіснішій інтеграції рекомендаційних систем із системами управління запасами (inventory management systems) та транспортно-логістичними платформами. Завдяки цьому можливо не лише забезпечити високу релевантність пропозицій, а й уникнути ситуацій, коли рекомендований товар відсутній на складі або доставка займає надто багато часу. У перспективі такі злагоджені процеси відкривають шлях до реалізації концепції «розумної доставки» – коли алгоритми не тільки рекомендують продукт, а й самостійно підбирають оптимальний маршрут доставки, передбачають пікові навантаження на склади та навіть синхронізуються з графіками постачальників. Усе це сприяє підвищенню довіри користувачів до платформи, зниженню рівня відмов та покращенню загального клієнтського досвіду.

1.3 Постановка задачі

У зв'язку з активним впровадженням технологій штучного інтелекту в електронну комерцію та зростаючою потребою в персоналізованому сервісі, виникає гостра необхідність у створенні інтелектуального веб-застосунку, здатного адаптувати пропозиції товарів відповідно до вподобань, інтересів і контекстної поведінки користувачів. В епоху цифрової трансформації, коли користувачі очікують максимальної зручності, швидкості та індивідуального підходу, традиційні методи фільтрації контенту вже не забезпечують достатнього рівня релевантності. Саме тому інноваційні підходи до персоналізації, зокрема гібридні системи рекомендацій, набувають дедалі більшої актуальності.

Основним завданням цієї дипломної роботи є розробка повнофункціонального інтернет-магазину з інтегрованим модулем інтелектуальних рекомендацій, який базується на поєднанні контентної та поведінкової фільтрації. Така система має забезпечити високий рівень релевантності товарних пропозицій, підвищення конверсії, утримання клієнтів та загальне покращення взаємодії між платформою і користувачем. Особливістю запропонованого рішення є його реалізація у вигляді мобільного застосунку, побудованого на основі сучасного кросплатформного фреймворку Flutter, що дозволяє забезпечити єдину кодову базу для Android та iOS. Для зберігання, обробки та автентифікації даних обрано хмарні сервіси Firebase, а для генерації персоналізованих рекомендацій – модель GPT-4o від OpenAI, яка підтримує роботу з природною мовою, здатна до контекстного аналізу та працює у режимі реального часу.

Задачі, які необхідно вирішити в рамках цього дослідження:

- Аналіз сучасного стану розвитку систем рекомендацій у сфері електронної комерції: провести огляд наукових джерел, комерційних кейсів та технічних рішень, що використовуються в провідних e-commerce платформах; виявити актуальні тенденції та проблеми галузі.
- Дослідження гібридних моделей рекомендацій: розглянути архітектурні принципи побудови таких систем, вивчити алгоритмічну базу комбінованих підходів, проаналізувати їхню ефективність з точки зору точності та релевантності рекомендацій у динамічному користувацькому середовищі.
- Обґрунтування вибору технологічного стеку: визначити оптимальні інструменти для реалізації проєкту, зокрема мову програмування, фреймворк, базу даних, способи інтеграції AI-моделей та підходи до побудови UI/UX.
- Розробка застосунку та впровадження системи рекомендацій: створити структуру бази даних, реалізувати механізм збору поведінкових даних, побудувати гібридну систему рекомендацій і забезпечити її взаємодію з користувачем у режимі реального часу.

- Аналіз результатів і оцінка ефективності: провести тестування реалізованої системи, оцінити якість і релевантність рекомендацій, виявити слабкі сторони проєкту та запропонувати шляхи подальшого вдосконалення і масштабування.

Ця робота спрямована на створення функціональної, масштабованої та адаптивної системи, що демонструє практичне застосування інструментів штучного інтелекту для персоналізації користувацького досвіду у сфері електронної торгівлі.

РОЗДІЛ 2. АНАЛІЗ ТЕХНОЛОГІЙ. ВИБІР ШІ-МОДЕЛІ. ПОСТАНОВКА ЗАДАЧІ

2.1. Аналіз сучасних платформ

Сьогоднішні онлайн-платформи, які ми звикли бачити щодня – типу **Amazon**, **AliExpress** чи **TikTok** – використовують дуже складні технології. За простими на перший погляд рекомендаціями “вам також може сподобатися” стоять потужні системи, які аналізують дії мільйонів користувачів. Для мого проекту важливо було розібратись, як саме це працює, щоб вибрати правильну технічну основу.

Більшість передових платформ функціонують на основі хмарних обчислень, які забезпечують масштабованість та гнучкість обробки даних. Наприклад, завдяки інтеграції з такими сервісами, як Google Cloud AI або AWS Machine Learning, компанії можуть в реальному часі оновлювати моделі, перевіряти нові гіпотези та миттєво застосовувати зміни без зупинки платформи. Цей підхід дозволяє бути гнучким до змін у поведінці користувачів, що є критично важливим у динамічному середовищі електронної торгівлі.

Розглянемо **Amazon**. Це не просто інтернет-магазин, а справжній гігант, який впровадив систему **Amazon Personalize**. Вона дозволяє в реальному часі аналізувати, що саме цікавить кожного окремого користувача. Система враховує, що купували інші люди з подібними вподобаннями, а також розуміє, які товари схожі між собою (наприклад, через спеціальні математичні моделі). Щоб усе це працювало швидко, Amazon зберігає поведінкові дані у своїй системі зберігання (Redshift) та кешує частину результатів – інакше сервіс просто б не витримав навантаження [20]. Також вони тестують варіанти видачі через A/B-тестування – наприклад, показують різні рекомендації окремим групам людей і порівнюють, що краще працює.

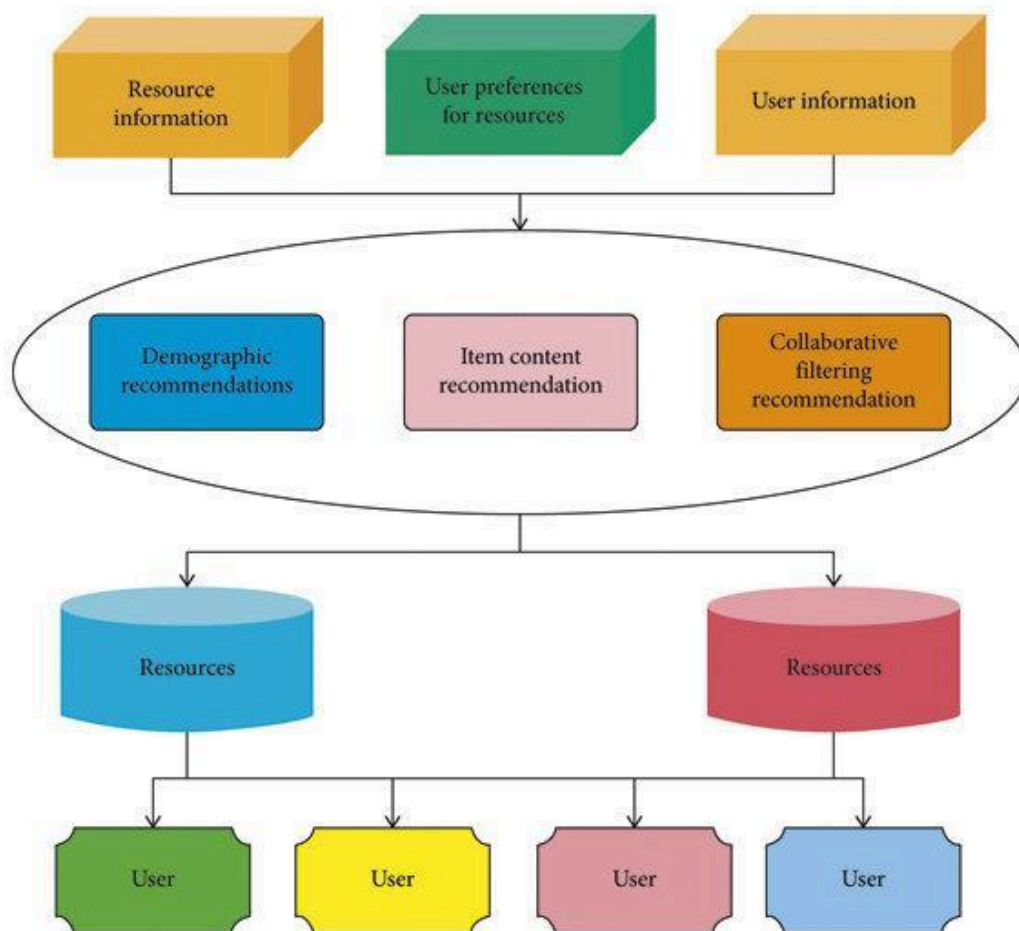


Рисунок 2.1 - Архітектура типової рекомендаційної системи

У **AliExpress** ситуація подібна, але є й свої особливості. Там персоналізація реалізована через платформу **PAI** від Alibaba. Вона орієнтована на поведінку користувачів: що шукають, як довго дивляться на товари, на що натискають. Є цікавий функціонал – пошук за картинкою. Завантажив фото – і система покаже подібні товари. Це стало можливим завдяки нейронним мережам, які “вміють бачити” зображення [21]. Всі ці дані зберігаються і обробляються в спеціальній розподіленій системі, яка справляється з величезними обсягами інформації.

Платформа **TikTok** працює інакше. Алгоритми TikTok не просто “радять відео”, а повністю формують стрічку контенту. Вони аналізують усе: як довго людина дивиться відео, чи лайкає, коментує, переглядає ще раз. Більше того, алгоритм враховує навіть те, що зображено на відео, які звуки звучать і навіть які емоції викликає контент. Усе це стало можливим завдяки складним неймережам, які працюють на основі трансформерів. Система навчається

постійно, буквально «на ходу», і вже за кілька хвилин може підлаштувати стрічку під конкретну людину.

Українська платформа **Rozetka** також демонструє прогресивний підхід до впровадження штучного інтелекту в електронну комерцію. Її інтерфейс не лише зручний для користувачів, а й гнучкий для інтеграції інтелектуальних компонентів. Однією з ключових функцій платформи є автоматичне формування персоналізованих блоків рекомендацій, які з'являються на головній сторінці, у каталозі та в картці товару. Ці блоки створюються на основі історії переглядів, доданих до кошика товарів, уподобаних позицій, а також з урахуванням даних з аналітики загальних трендів. Наприклад, якщо користувач не завершив покупку, Rozetka може надіслати e-mail із нагадуванням або новою добіркою аналогічних товарів – це елемент **retargeting-механіки**, яка підвищує конверсію.

Не менш важливим є той факт, що сучасні системи персоналізації дедалі частіше інтегруються з чат-ботами та віртуальними помічниками. Наприклад, платформи аналогічні eBay активно експериментують із генеративним ШІ, який може не лише рекомендувати товари, а й пояснювати, чому вони релевантні, створюючи більш “людську” взаємодію. Такі функції особливо корисні в мобільних застосунках, де швидкість та інтуїтивність взаємодії з користувачем мають вирішальне значення.

Окремий інтерес становить тенденція до універсалізації рекомендаційних підходів. Сучасні платформи дедалі частіше переходять від використання вузькоспеціалізованих алгоритмів (типу колаборативної фільтрації або окремих евристик) до інтегрованих гібридних моделей, які поєднують поведінкові патерни, вміст товарів, семантичну інформацію та контекст взаємодії. Це дозволяє створювати не лише точніші, а й гнучкіші системи, здатні підлаштовуватись під будь-які сценарії використання. Крім того, усе більше компаній починають розглядати мультимодальні підходи – тобто аналіз не лише тексту чи кліків, а й зображень, голосу, емоцій, що відкриває шлях до ще більш «людяного» цифрового досвіду.

Можемо зробити висновки:

По-перше, те, що всі великі платформи роблять ставку на персоналізацію.

По-друге, кожна система будується не лише на алгоритмах, а й на хорошій технічній інфраструктурі. І саме це – швидка обробка даних та тестування різних підходів – є обов’язковими умовами для створення конкурентної рекомендаційної системи.

2.2. Дослідження моделей нейромереж

В сучасному житті при активному використанні нейронних мереж у сучасних цифрових продуктах доцільно проаналізувати існуючі архітектури, що застосовуються для побудови рекомендаційних систем. Попередній аналіз довів ефективність таких систем у масштабних комерційних платформах, де швидкість, точність і гнучкість адаптації відіграють ключову роль у підвищенні якості взаємодії з користувачем.

Серед основних архітектур штучних нейронних мереж, що використовуються в сучасних рекомендаційних системах, варто детальніше розглянути чотири ключові підходи – згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN), автоенкодера та трансформери. Кожна з цих моделей має свої особливості, переваги та обмеження, що визначають сферу її ефективного застосування.

Згорткові нейронні мережі (CNN) застосовуються переважно для обробки візуальних даних. Їхня головна перевага – здатність «бачити» просторові зв’язки між елементами зображення, що дозволяє успішно вирішувати задачі класифікації, розпізнавання образів або пошуку товарів за фотографією. Наприклад, платформи аналогічні AliExpress використовують CNN для реалізації функції «знайти подібне», коли система аналізує завантажене користувачем зображення й знаходить схожі товари в каталозі. Однак для послідовностей даних, таких як історія дій користувача, CNN не є оптимальними, оскільки вони не враховують порядок подій.

Рекурентні нейронні мережі (RNN) та їхні розширення – **LSTM (Long Short-Term Memory)** – краще підходять для обробки часових рядів або даних, де критично важливо зберегти послідовність. Це можуть бути, зокрема, сесії перегляду товарів, кліки або покупки користувача в хронологічному порядку. LSTM-мережі здатні «запам'ятовувати» важливу інформацію з минулого й використовувати її для передбачення майбутніх дій, що робить їх ефективними для побудови рекомендацій, орієнтованих на динаміку поведінки.

Автоенкодери використовуються для виявлення прихованих закономірностей у даних шляхом стискання інформації в компактне векторне представлення. У контексті рекомендацій вони дозволяють моделювати уподобання користувачів та властивості товарів у спільному латентному просторі, де схожі об'єкти мають близькі координати. Це корисно для випадків, коли необхідно об'єднати дані з різних джерел або зменшити їхню розмірність для подальшої обробки. Проте автоенкодери рідко використовуються самостійно – зазвичай їх інтегрують у складніші гібридні архітектури.

Трансформери є найсучаснішим і найбільш універсальним підходом. Їхня ключова особливість – використання механізму «уваги» (self-attention), що дозволяє моделі зважати на взаємозв'язки між усіма елементами вхідної послідовності одночасно. Це забезпечує високу контекстуальну обізнаність, гнучкість і здатність обробляти як текст, так і мультимодальні дані. Моделі на базі трансформерів, зокрема GPT-4o, можуть не лише аналізувати вподобання користувача, а й формувати рекомендації у вигляді повноцінного діалогу, реагуючи на уточнення чи зміну інтересів у реальному часі. Вони також краще масштабуються та підтримують паралельну обробку, що критично важливо при роботі з великими наборами даних.

Таким чином, кожен з розглянутих підходів має свою нішу в екосистемі рекомендаційних систем. Вибір конкретної архітектури залежить від типу даних, вимог до точності, швидкості, гнучкості адаптації та цільового сценарію застосування.

Найбільш передовим класом моделей на сьогодні є трансформери. Вони дозволяють ефективно працювати з великими обсягами даних, зберігаючи контекст навіть на великих відстанях у послідовності. Особливої уваги заслуговує архітектура GPT (Generative Pre-trained Transformer), яка стала основою багатьох сучасних систем персоналізації. Її перевага полягає у здатності комбінувати знання, отримані з різних джерел, з генерацією змістовного контенту на основі проміжних сигналів. Зокрема, новітня модель ChatGPT-4o, має мультимодальні можливості – вона здатна працювати не лише з текстом, а й з візуальними та аудіо даними [22]. Завдяки цьому відкривається перспектива створення рекомендаційної системи, що не лише пропонує товари на основі минулих дій користувача, а й здатна проводити повноцінну діалогову взаємодію, уточнювати інтереси, адаптувати тон комунікації та формувати цілеспрямовані рекомендації в реальному часі.

Рішення щодо використання GPT-4o у даному проєкті продиктоване низкою технічних і практичних міркувань. Насамперед, модель надається через зручний API від OpenAI, що суттєво спрощує її інтеграцію в інфраструктуру мобільного додатку – не потребуючи локального розгортання чи обслуговування серверів. Це дозволяє зосередитися на бізнес-логіці, не витрачаючи ресурси на підтримку складної ML-архітектури. Також GPT-4o вирізняється високим рівнем контекстуального розуміння, що є особливо важливим для точного формування персоналізованих рекомендацій у сфері електронної комерції. Її здатність аналізувати як явні, так і неявні сигнали (наприклад, послідовність дій, історію покупок, поведінкові патерни) дозволяє створювати більш адаптивний і «людяний» користувацький досвід.

У порівнянні з альтернативними нейромережевими архітектурами GPT-4o демонструє значно вищу гнучкість та адаптивність. Так, згорткові нейронні мережі (CNN), хоча й ефективні у візуальній класифікації, не пристосовані до аналізу поведінкових шаблонів або роботи з послідовністю дій. Рекурентні моделі (RNN, LSTM) можуть працювати з часовими рядами, але мають труднощі з паралельною обробкою даних і схильні до втрати контексту на довгих відрізках

інформації. Автоенкодери, у свою чергу, добре зменшують розмірність даних і дозволяють побудувати спільний латентний простір для користувачів і товарів, однак слабо пристосовані до інтерактивного сценарію, де потрібна динамічна комунікація. Трансформери типу GPT-4o поєднують сильні сторони всіх вищезгаданих підходів і дозволяють реалізувати діалогові системи, що здатні адаптуватися до змін контексту в режимі реального часу, що є вкрай важливим для систем персоналізованих рекомендацій.

Окрім того, GPT-4o функціонує як генеративна модель, яка вміє не лише відповідати на запити, а й проактивно формувати рекомендації, базуючись на накопиченій інформації про користувача. Такий підхід дозволяє системі діяти на випередження, підказуючи товари чи акції ще до того, як користувач сам усвідомить свою потребу. Це відкриває шлях до нової якості взаємодії між платформою та клієнтом.

Окремої уваги заслуговує економічна модель OpenAI API – вона базується на принципі «pay-as-you-go», що означає оплату лише за фактично використані ресурси. У випадку проєктів, які перебувають на початковому етапі розвитку або мають обмежений трафік, це дозволяє істотно зекономити кошти порівняно з альтернативними рішеннями, що вимагають передоплати або виділення обчислювальної інфраструктури наперед. Таким чином, GPT-4o виявляється не лише технічно обґрунтованим, а й фінансово доцільним вибором для реалізації системи персоналізованих рекомендацій в межах даного інтернет-магазину.

2.3. Проектування архітектури системи

Архітектура створюваної системи базується на принципах модульності, масштабованості та інтегрованості зі сторонніми інтелектуальними сервісами. Основним завданням цього етапу є проектування логічної структури компонентів, які у взаємодії забезпечують реалізацію ключового функціоналу інтернет-магазину з персоналізованими рекомендаціями. Вибір архітектурної моделі був продиктований необхідністю забезпечення гнучкої клієнт-серверної

взаємодії, підтримки асинхронної роботи та можливості оперативного оновлення окремих елементів без втручання у цілісність системи.

Фронтенд-частина реалізована за допомогою кросплатформного фреймворку **Flutter**, що забезпечує єдиний кодовий базис для мобільних додатків на Android та iOS. Вибір цієї технології обґрунтований її високою продуктивністю, широкими можливостями UI-розробки та активною підтримкою спільноти. Flutter дозволяє створювати адаптивні інтерфейси, які швидко реагують на зміну даних у режимі реального часу, що є критично важливим для персоналізованих платформ.

Для зберігання та обробки даних використовується **Firestore Database** – хмарна NoSQL-база даних від Google, що забезпечує масштабовану, реактивну та структуровану роботу з даними. Контент, який включає як текстову інформацію (опис товарів, назви, категорії), так і мультимедійні ресурси (зображення продуктів), розміщується у **Firestore Storage**. Такий розподіл дає змогу оптимізувати швидкість доступу до критичних ресурсів та зменшити навантаження на канал передачі даних.

Аутентифікація користувачів реалізована за допомогою **Firestore Authentication**, де наразі впроваджено механізм реєстрації та входу через електронну пошту та пароль. Надалі система передбачає можливість розширення спектру методів авторизації, зокрема через соціальні мережі або біометричні дані. Наявність захищеного облікового запису дозволяє системі формувати персоналізовані рекомендації з урахуванням історії взаємодій кожного користувача.

Ключовим компонентом системи є **модуль рекомендацій**, що реалізується шляхом інтеграції з **OpenAI API**, зокрема моделлю **GPT-4o**. Вибір цієї моделі зумовлений її здатністю до високоточних контекстуальних аналізів, гнучкого оброблення запитів та підтримки мультимодального входу. За допомогою REST-запитів додаток надсилає до API агреговані дані про вподобання, перегляди та інші дії користувача, натомість отримуючи релевантні товарні рекомендації. Окрім базових функціональних модулів, система також передбачає реалізацію

аналітичного підмодуля, який відповідатиме за моніторинг поведінки користувачів у додатку. Зокрема, збиратимуться метрики взаємодії (час перегляду картки товару, частота натискань, глибина прокрутки), що дозволить не лише адаптувати рекомендації в режимі реального часу, а й будувати агреговану аналітику для адміністратора магазину. Збір таких даних здійснюється безпосередньо через події Flutter та зберігається у Firebase Analytics, звідки за потреби може бути експортовано до BigQuery для глибшого аналізу. Такий підхід забезпечує прозору та масштабовану систему збору поведінкової інформації, яка з часом може стати джерелом нових стратегій розвитку бізнесу. Важливою перевагою інтеграції з OpenAI є модель оплати **pay-as-you-go**, що дозволяє мінімізувати витрати на обчислювальні ресурси у випадку невеликого або нерегулярного трафіку, зберігаючи водночас якість та доступність інтелектуальної функціональності.

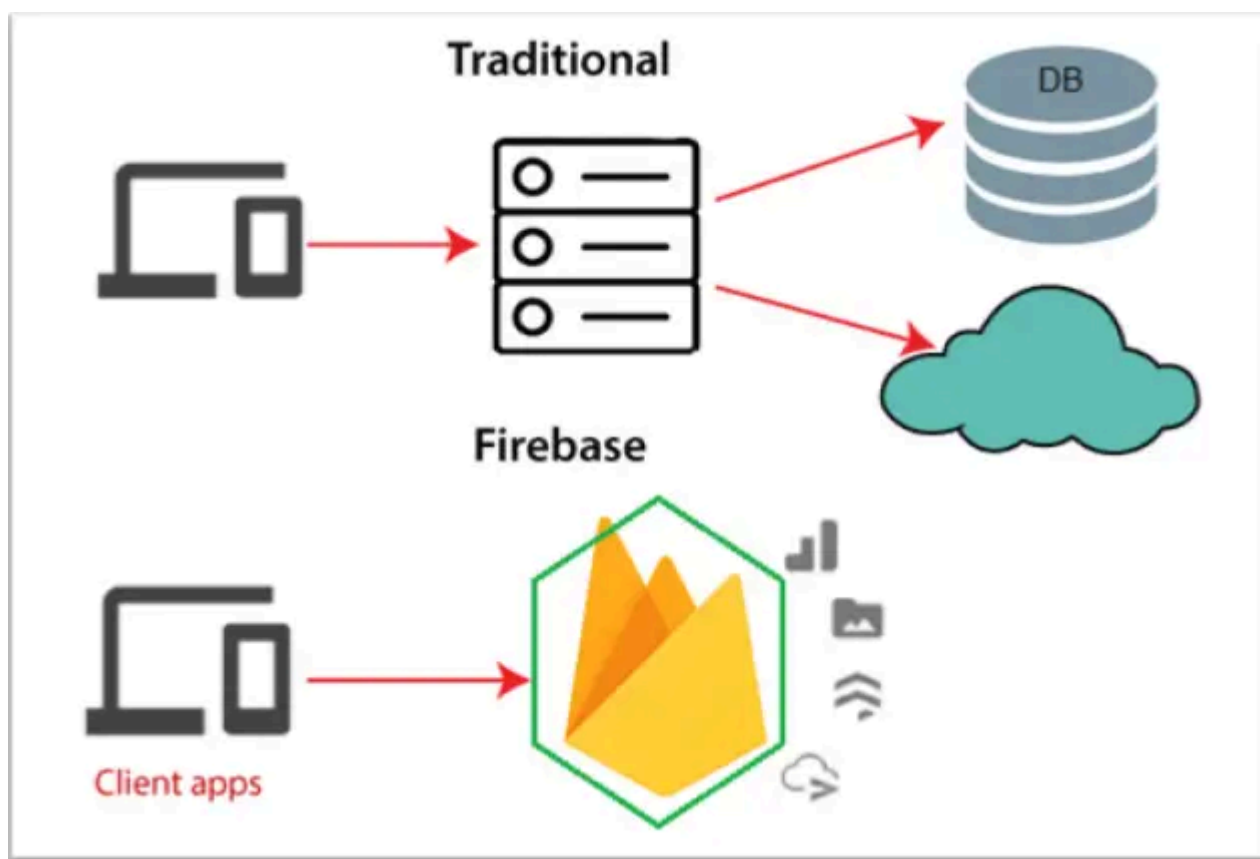


Рисунок 2.2 - Архітектура типової рекомендаційної системи

Окрему увагу під час проектування архітектури було приділено аспектам безпеки, стійкості до збоїв та обробки виняткових ситуацій. Firebase забезпечує базовий рівень захисту даних за допомогою правил доступу, що контролюються

на рівні користувача, а також шифрування інформації під час зберігання та передачі. Для обробки помилок у клієнтській частині запропоновано механізми логування та сповіщення, які дозволяють розробникам оперативно виявляти і усувати критичні помилки у роботі застосунку. Крім того, система передбачає використання локального кешування даних (через механізми Flutter та Firestore), що забезпечує функціонування базових сервісів навіть у разі тимчасового відключення від інтернету. Такий підхід гарантує стабільність роботи застосунку, покращує користувацький досвід і мінімізує ризики втрати даних. Ще одним важливим аспектом є підтримка офлайн-режиму роботи додатку. У багатьох користувачів може бути нестабільне інтернет-з'єднання, особливо при використанні мобільного додатку в умовах низької пропускну здатності мережі. Саме тому Flutter-застосунок реалізує кешування останніх отриманих рекомендацій, зображень та основного каталогу товарів. Це дозволяє зберегти мінімальну функціональність інтерфейсу – перегляд останніх рекомендацій, обране, історія покупок – навіть у разі тимчасової втрати доступу до мережі. Як тільки з'єднання буде відновлено, дані автоматично оновлюються, що забезпечує безперервність користувацького досвіду.

Усі компоненти системи взаємодіють через **асинхронні запити** та підтримують **реактивний обмін даними**, що забезпечує безперебійну роботу додатку в режимі реального часу. Архітектура також дозволяє легко масштабувати систему – додавати нові типи рекомендацій, розширювати базу продуктів, впроваджувати додаткові мовні моделі або змінювати структуру взаємодії з базою даних без потреби у повному перепроєктуванні.

Варто також зазначити, що запропонована архітектура легко адаптується до подальшого впровадження мультимодального введення. Наприклад, за допомогою майбутньої інтеграції з GPT-4o передбачається можливість запровадження голосового пошуку товарів або спілкування з віртуальним асистентом, що дозволить користувачам формулювати запити природною мовою. Така гнучкість є критично важливою для довгострокової перспективи розвитку системи в умовах

стрімкого прогресу у сфері штучного інтелекту та зростаючих очікувань кінцевих користувачів.

Таким чином, проектована архітектура поєднує сучасні хмарні сервіси, мобільні технології та засоби штучного інтелекту в єдину узгоджену інфраструктуру, здатну ефективно вирішувати завдання персоналізованої взаємодії у сфері електронної комерції.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ

1.1. Архітектура системи

Розроблена система інтернет-магазину, яку можна побачити на рисунку 3.1, з функцією персоналізованих рекомендацій ґрунтується на клієнт-серверній архітектурі з активною підтримкою хмарних технологій та зовнішніх API-сервісів. Такий підхід дозволяє досягти високої гнучкості, масштабованості та стабільності роботи навіть за умов збільшення навантаження. Всі компоненти системи розділені за функціональним призначенням, що спрощує модифікацію окремих частин проєкту та сприяє підтримці принципів модульності.кількості.

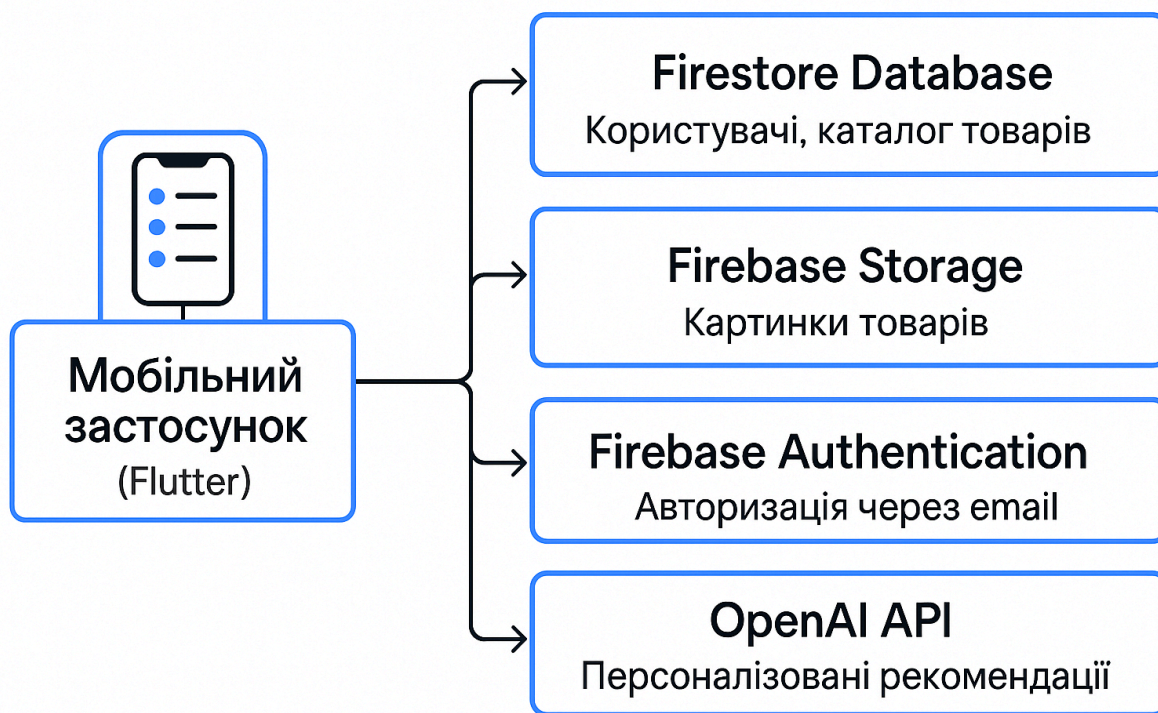


Рисунок 3.1 - архітектура системи

Клієнтська частина реалізована за допомогою фреймворку **Flutter**, який дозволяє створити єдиний кодовий базис для Android та iOS. Це значно зменшує часові витрати на розробку та забезпечує кросплатформну сумісність. Користувач взаємодіє з додатком через адаптивний інтерфейс, що відображає товари, рекомендації, профіль користувача, форму входу та інші компоненти. Усі запити

до бази даних, а також звернення до сторонніх сервісів, відбуваються безпосередньо з мобільного клієнта.

В якості бекенду використовується **Firestore** – комплекс хмарних сервісів від Google. Зокрема:

- **Firestore Database** – для зберігання структурованої інформації про товари, профілі користувачів, історію взаємодій.
- **Firestore Storage** – для зберігання мультимедійних файлів, зокрема зображень товарів.
- **Firestore Authentication** – для реалізації безпечної аутентифікації користувачів через email та пароль.

Особливу роль у системі відіграє **OpenAI API**, що забезпечує генерацію персоналізованих рекомендацій. На основі зібраної інформації про покупки користувача, додаток формує запит до моделі **GPT-4o**, яка аналізує вхідні дані й повертає текстові рекомендації у відповідь. Ці відповіді інтерпретуються клієнтською логікою, трансформуються у списки товарів і виводяться у відповідному інтерфейсі.

Комунікація між компонентами побудована за допомогою **REST API**, що дозволяє забезпечити асинхронну передачу даних, знизити навантаження на клієнт і покращити відгук системи. Завдяки **Firestore** функціональність синхронізується у реальному часі, а дані оновлюються без необхідності ручного перезавантаження додатку. Така архітектура сприяє швидкому реагуванню системи на дії користувача, що є ключовим чинником у контексті персоналізованої комерції.

Загалом проєктована архітектура демонструє поєднання мобільних технологій, хмарних сервісів та інтелектуальних алгоритмів, що забезпечує повноцінний функціонал для персоналізованої взаємодії у середовищі електронної торгівлі.

1.2. Розробка бази даних та структури зберігання

Для зберігання інформації в проєкті використовується хмарна база даних **Firebase Firestore**, яка дозволяє будувати гнучкі ієрархічні структури даних у вигляді колекцій та документів. Завдяки підтримці режиму реального часу, Firestore забезпечує миттєве оновлення вмісту мобільного додатку без потреби в ручному оновленні інтерфейсу.

На поточному етапі розробки система містить такі ключові колекції:

- **products** – зберігає інформацію про товари;
- **orders** – фіксує замовлення, що здійснюють користувачі;
- **users** – містить профілі зареєстрованих користувачів;
- **sellers** – описує облікові записи продавців;
- **reviews** – дозволяє зберігати відгуки про товари;
- **activity** – фіксує додаткову поведінкову активність або події в системі.

Колекція users зберігає профілі всіх зареєстрованих користувачів, які взаємодіють з платформою – як покупці, так і продавці. Дані використовуються для персоналізації, авторизації, відображення імен у замовленнях, відгуках тощо. Зокрема:

- **display_name** – ім'я, яке відображається у профілі користувача, коментарях, списках замовлень;
- **email** – електронна адреса, що використовується для входу через Firebase Authentication;
- **location** – географічне розташування користувача, використовується для аналітики або локалізованих рекомендацій;
- **bio** – текстовий опис профілю (необов'язковий), може застосовуватись для відгуків чи сторінки профілю;
- **uid** – унікальний ідентифікатор Firebase, зв'язує акаунт із системою авторизації;
- **photo_url** – посилання на аватар користувача, зберігається у Firebase Storage або сторонніх сервісах;
- **created_time** – мітка часу, коли було створено обліковий запис.

Колекція sellers пов'язана з обліковими записами продавців. Вона дозволяє розмежувати функціональність між звичайними користувачами й тими, хто має право додавати товари. Зокрема:

- `display_name` – ім'я продавця, яке може відображатися біля товарів;
- `email` – електронна адреса, зареєстрована в системі (може відрізнятися від контактної);
- `phone_number` – номер телефону для зв'язку, який може бути використаний підтримкою або покупцями;
- `userRef` – посилання на відповідного користувача з колекції `users`, що гарантує синхронізацію облікових записів;
- `created_at` – дата реєстрації продавця, може бути використана в аналітиці або перевірці активності.

Колекція products є ядром системи товарів. Вона містить усю необхідну інформацію про доступні для купівлі позиції. Зокрема:

- `name` – повна назва товару;
- `description` – короткий або детальний опис товару для сторінки продукту;
- `category` – категорія або підкатегорія, за якою товар може бути згрупований (наприклад: "shoes", "electronics");
- `price` – числове значення вартості, у форматі `double`;
- `pid` – унікальний внутрішній ідентифікатор продукту;
- `photo_url` – пряме посилання на картинку, яка зберігається у `Firebase Storage`;
- `seller_ref` – посилання на документ продавця, що додав товар;
- `userRef` – додаткове посилання на користувача, що виконав дію;
- `created_at` – мітка часу додавання, може використовуватись для фільтрації «нових» товарів.

Колекція orders зберігає інформацію про покупки, оформлені користувачами. Вона має гнучку структуру для підтримки кількох товарів, доставок і статусів. Зокрема:

- `amount` – загальна сума замовлення (включає ціну товарів і податки);
- `tax` – розраховане значення податку (якщо застосовується);

- `created_at` – дата й час оформлення замовлення;
- `itemsOrdered` – масив із посиланнями на конкретні продукти з колекції `products`;
- `shippingSelected` – вкладений об'єкт, який включає:
 - `description` – пояснення до типу доставки (наприклад, "доставка 10– 15 днів");
 - `shippingName` – коротка назва методу доставки (наприклад, "Free Delivery");
 - `price` – вартість обраного способу доставки;
- `status` – статус обробки замовлення ("Accepted", "Pending", "Delivered" тощо);
- `userPurchased` – посилання на користувача, який здійснив покупку;
- `vendor_name` – текстове поле з назвою магазину або бренду.

Колекція `reviews` дозволяє користувачам оцінювати товари, залишаючи коментарі. Ці дані не лише інформують інших покупців, а й використовуються як частина рекомендаційної системи. Зокрема:

- `reviewName` – короткий заголовок відгуку;
- `reviewDescription` – текстовий зміст, що відображає думку користувача;
- `rating` – числова оцінка від 1 до 5;
- `productRef` – посилання на товар, до якого залишено відгук;
- `userRef_Reviewer` – посилання на користувача, який написав коментар;
- `dateCreated` – мітка часу створення відгуку, корисна для сортування.

Колекція `activity` реалізує механізм активності користувачів, зокрема повідомлень, сповіщень або історії взаємодії із контентом платформи. Вона дозволяє зберігати й відображати події, пов'язані з конкретними діями – наприклад, переглядом товару, оновленням статусу замовлення або повідомленням від адміністрації. До її основних полів належать:

- `content` – текст повідомлення або опис події (наприклад, "Користувач переглянув товар X");
- `productRef` – посилання на документ із колекції `products`, з яким пов'язана подія;
- `title` – короткий заголовок повідомлення або опис категорії події;
- `sentAt` – мітка часу, коли було зафіксовано або відправлено активність;

- userList – масив посилань на користувачів, яких стосується подія (наприклад, усі отримувачі сповіщення або учасники взаємодії);

Структура Firestore, представлена на рисунку 3.2, реалізована в межах цього проєкту, і слугує прикладом продуманої та масштабованої моделі для платформи електронної комерції. Поділ даних на логічні колекції дозволяє не лише зручно зберігати й оновлювати інформацію, а й ефективно виконувати пошукові запити, проводити аналітику, та інтегрувати зовнішні сервіси, зокрема систему рекомендацій.

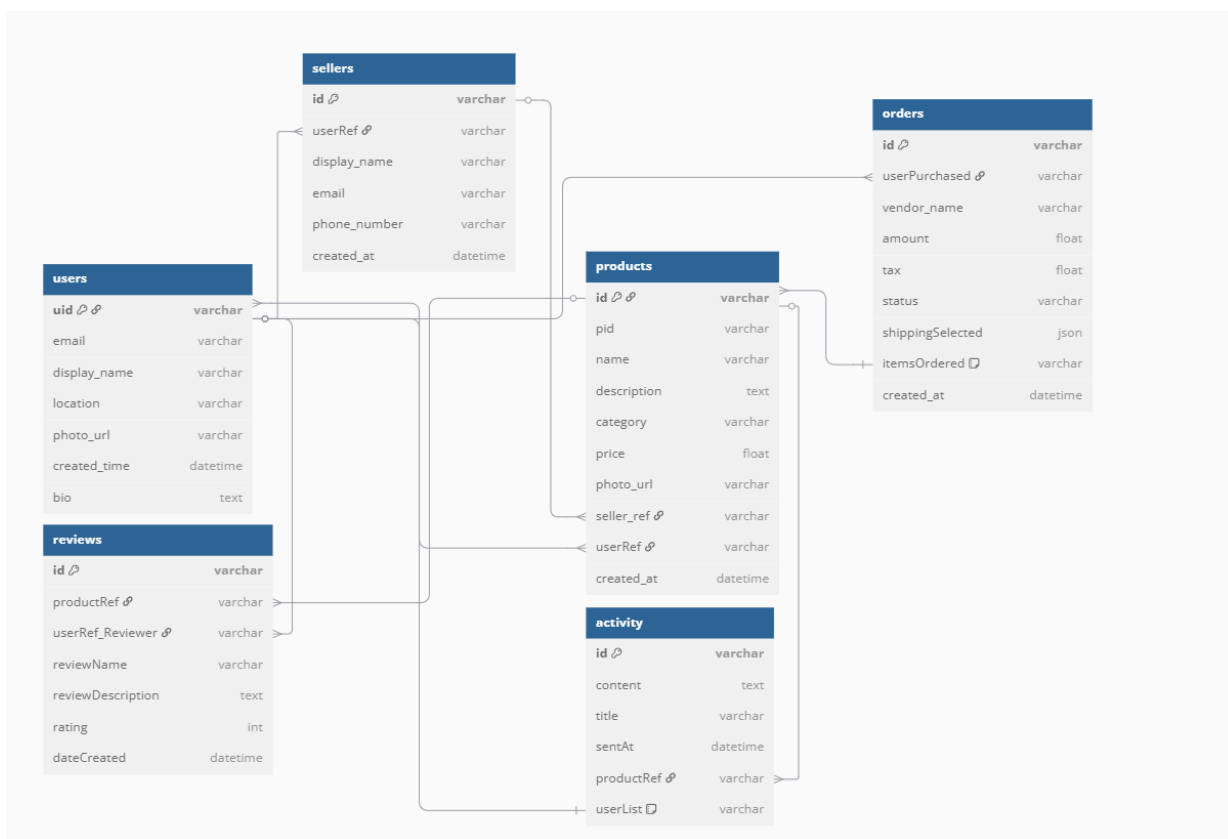


Рисунок 3.2 – ER-діаграма

Чітке розмежування між ролями користувача й продавця спрощує управління правами доступу, забезпечує безпечну роботу з даними та дозволяє надалі масштабувати платформу до рівня мультивендорного майданчика.

Інтеграція посилань між документами, таких як `productRef`, `userRef`, `seller_ref`, дозволяє зберігати реляційність у нереляційній базі даних, не втрачаючи при цьому гнучкості документа-орієнтованої моделі. Це суттєво спрощує

подальше формування запитів для системи рекомендацій, оскільки необхідні взаємозв'язки вже передбачені на рівні структури.

Зберігання зображень у Firebase Storage з посиланням у Firestore дозволяє зменшити навантаження на базу та розділити відповідальність за дані й медіа. Такий підхід покращує масштабованість та продуктивність системи, особливо в умовах великої кількості користувачів і товарів.

Вбудована підтримка індексів і можливість використання складених запитів відкривають додаткові перспективи для оптимізації роботи інтерфейсу та рекомендаційного ядра. Наприклад, рекомендації можуть фільтруватися за категоріями, рейтингом або географічним місцем розташування, що безпосередньо відображено у структурі даних.

Загалом запропонована модель є технічно обґрунтованою для реалізації інтернет-магазину з адаптивною персоналізацією. Вона гнучка, легко розширюється, підтримує гібридні підходи до рекомендацій, і забезпечує надійну основу для подальшого розвитку функціоналу додатку.

3.3. Реалізація гібридної системи рекомендацій

Одним із ключових компонентів реалізованого мобільного застосунку інтернет-магазину одягу є система персоналізованих рекомендацій. Її головна мета – автоматичне формування індивідуального списку товарів, які можуть зацікавити конкретного користувача. Такий функціонал дозволяє не лише покращити взаємодію з платформою, але й істотно підвищити ймовірність покупки завдяки релевантному контенту. У розрізі комерційної ефективності це сприяє зростанню конверсій, підвищенню середнього чека та зниженню часу, витраченого на пошук потрібного товару.

З технічної точки зору, система рекомендацій побудована на **гібридному принципі**, де поєднується спрощена колаборативна логіка, контентний аналіз описів товарів та можливості великої мовної моделі GPT-4o. Хоча система не використовує класичних моделей машинного навчання з навчанням на великих

масивах даних, вона успішно виконує функцію адаптивного підбору рекомендацій на основі обмеженої поведінкової інформації користувача. Важливою перевагою такого підходу є швидкість впровадження, масштабованість і мінімальні вимоги до локального зберігання або обчислень на клієнті.

Користувацька поведінка фіксується у спеціальній колекції *activity*, куди в реальному часі надходить інформація про перегляди товарів. Ці записи містять унікальний ідентифікатор користувача, ID товару, час перегляду та контекстну інформацію. Після завершення покупки відповідні дані фіксуються у колекції *orders*, де зберігається перелік придбаних позицій. Таким чином, система має доступ до двох ключових каналів даних – пасивної поведінки (перегляди) та активної взаємодії (купівлі).

На основі цих даних формується текстовий запит, що передається до OpenAI API. У ньому коротко описується, які саме товари користувач переглядав та що придбав. Також у запит включаються назви, категорії та описи відповідних товарів. GPT-4o, отримавши цей контекст, генерує список із декількох потенційно релевантних товарів, які система потім підтягує з колекції *products* та виводить на головному екрані застосунку.

Таким чином, попри обмежену складність реалізації, система успішно виконує функцію інтелектуального добору товарів, забезпечуючи індивідуальний підхід до кожного користувача без необхідності створення складної моделі навчання або великих обчислювальних витрат на боці розробника.

Формування персоналізованих рекомендацій у реалізованій системі відбувається шляхом взаємодії з мовною моделлю **GPT-4o**, яка надається компанією OpenAI у форматі зовнішнього **API**. Такий підхід дозволяє відмовитися від необхідності локального розгортання моделей або використання складної інфраструктури, що особливо важливо у контексті мобільного застосунку з обмеженим ресурсним бюджетом.

Кожного разу, коли користувач завершує певну активність – наприклад, здійснює покупку або переглядає кілька товарів поспіль – застосунок формує запит до **OpenAI API**. Цей запит включає кілька основних блоків інформації:

унікальний ідентифікатор користувача, список останніх переглянутих і куплених товарів (назва, категорія, короткий опис), а також, за можливості, історію попередніх рекомендацій. Усі ці дані об'єднуються в один структурований текстовий **prompt**, який подається мовній моделі.

У відповідь **GPT-4o** повертає згенерований список, який у подальшому аналізується клієнтом (або middleware-сервером, якщо буде реалізовано). На основі назви та категорії кожного запропонованого товару система виконує пошук у колекції **products** і формує візуальний блок рекомендацій.

Особливість цього підходу полягає у високій **контекстуальній адаптації** – модель здатна формувати не тільки точні, а й інтуїтивно зрозумілі рекомендації, враховуючи семантику товарів. **GPT-4o** працює з будь-якою мовою, що дозволяє легко адаптувати застосунок під україномовну або багатомовну аудиторію.

Важливою перевагою **OpenAI API** є модель оплати **pay-as-you-go**, що дозволяє уникати постійних витрат за невеликого трафіку. Це особливо вигідно на ранніх етапах розгортання проєкту, коли база користувачів ще не є масштабною. Процес побудови запиту до API включає три основні частини:

1. Системне повідомлення (**system**) – вказівка, яка задає загальну роль моделі. У цьому проєкті модель виконує функції персонального стиліста, що рекомендує одяг на основі поведінки користувача.
2. Історія взаємодії (**messages**) – включає:
 - список товарів з каталогу магазину (назви, категорії);
 - історію переглядів та покупок користувача.
3. Температура – параметр, що визначає ступінь варіативності відповідей. У нашому випадку він встановлений на рівні 0.7, що дозволяє моделі бути достатньо гнучкою, але залишатися в межах релевантного контексту.

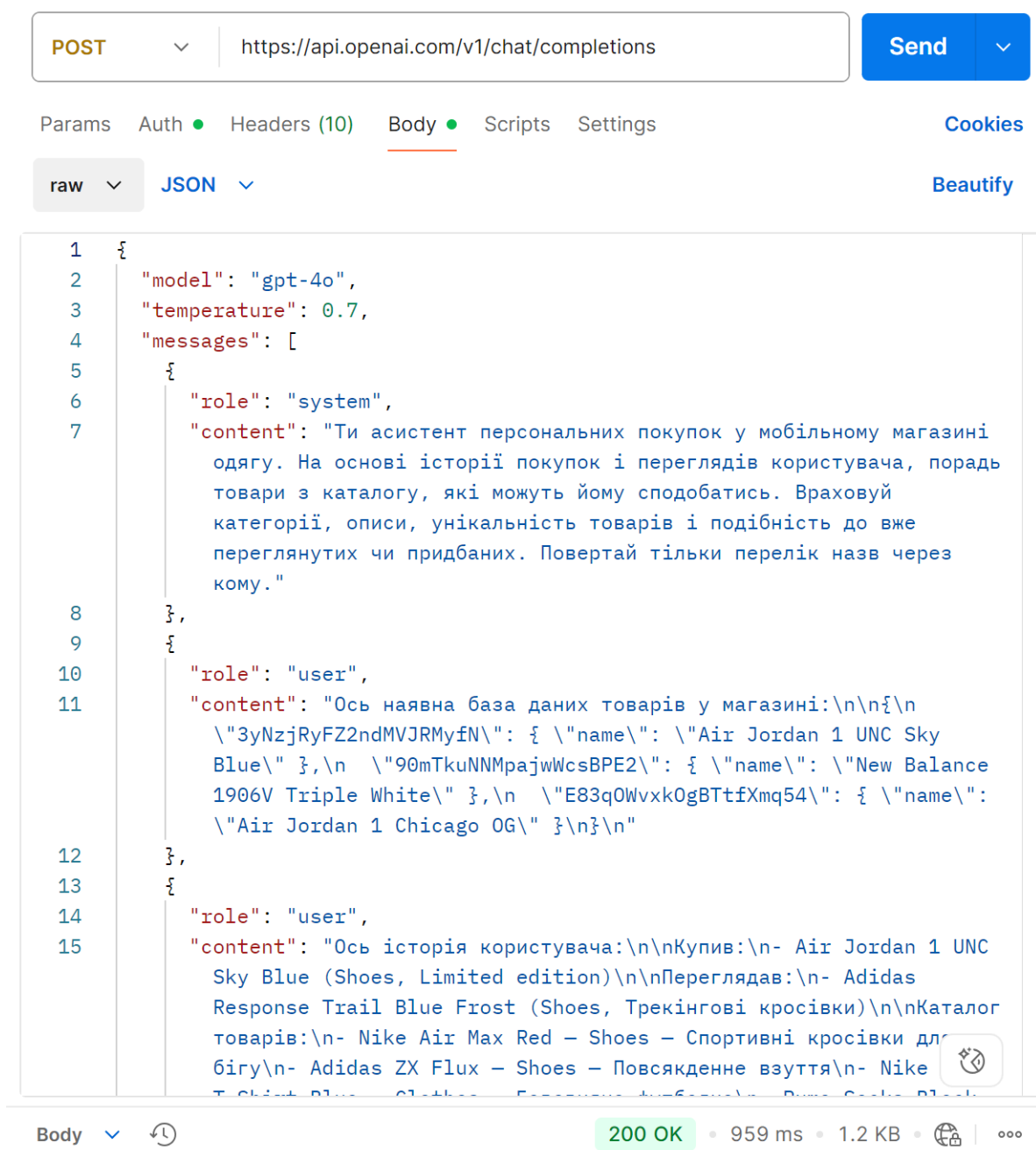


Рисунок 3.3 - приклад запиту в OpenAI API

На рисунку 3.3 продемонстровано приклад побудови запиту до OpenAI API через інтерфейс Postman. У цьому запиті можна побачити системне повідомлення, сформовану історію дій користувача та список доступного каталогу. Все передається у вигляді JSON-структури згідно з вимогами OpenAI.

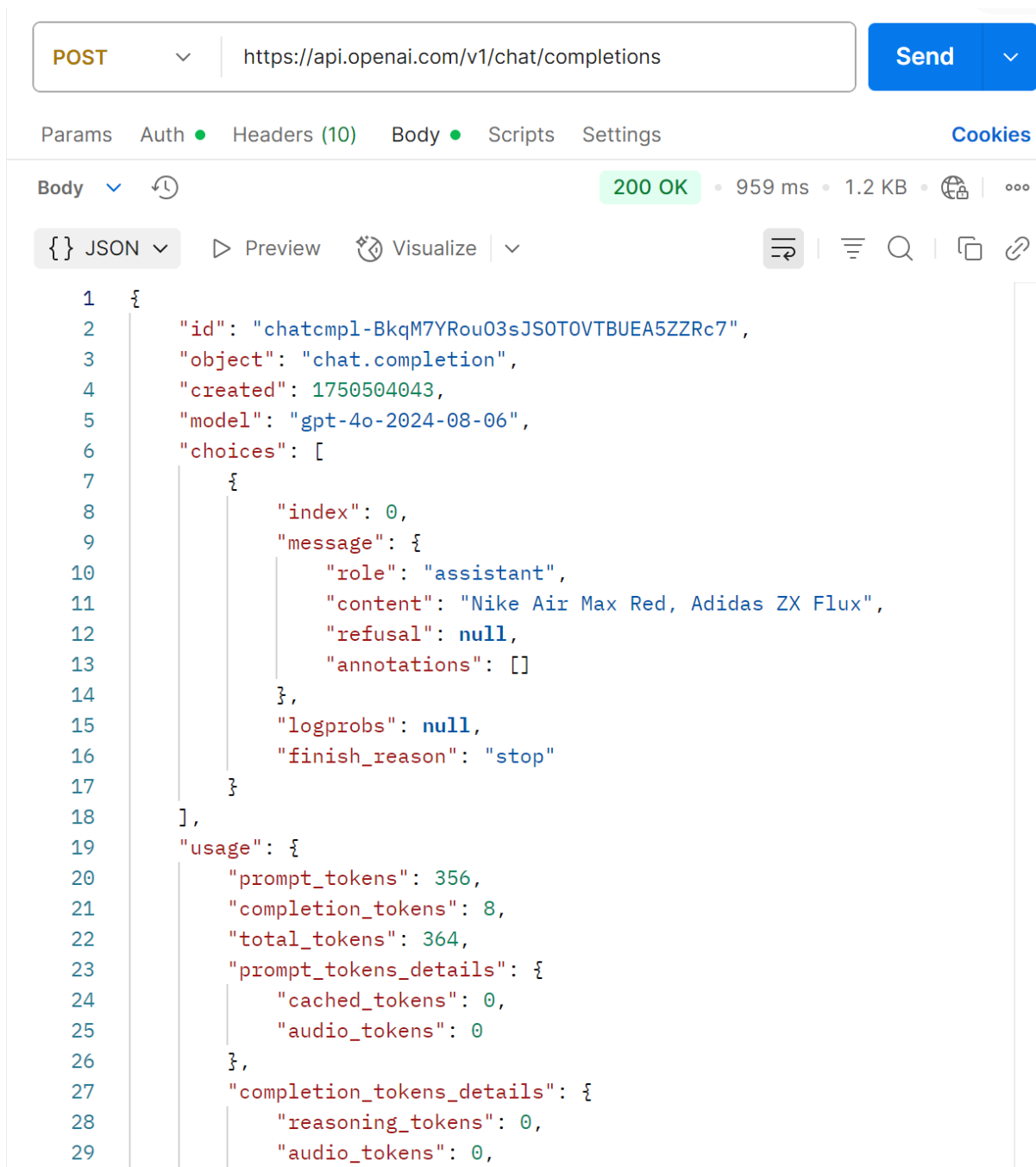


Рисунок 3.4 - приклад запиту в OpenAI API

Відповідь від GPT-4o представлена на рисунку 3.4. Модель повертає лише перелік рекомендованих товарів відповідно до наданого контексту. Ці назви форматуються (рис. 3.5) та використовуються клієнтською частиною застосунку для пошуку відповідних записів у базі products (рис. 3.6), після чого вони виводяться у вигляді візуального блоку рекомендацій (див. Додаток А).

```

1  import 'dart:convert';
2
3  List<String> parseRecommendations(String responseBody) {
4      final decoded = json.decode(responseBody);
5
6      final choices = decoded['choices'];
7      if (choices == null || choices.isEmpty) return [];
8
9      final content = choices[0]['message']['content'];
10     if (content == null || content.isEmpty) return [];
11
12     // Розбиваємо строку за комами і прибираємо зайві пробіли
13     final recommendations = content
14         .split(',')
15         .map((item) => item.trim())
16         .where((item) => item.isNotEmpty)
17         .toList();
18
19     return recommendations;
20 }

```

Рисунок 3.5 - парсинг отриманих від OpenAI даних

```

67  Future<List<Map<String, dynamic>>> fetchRecommendedProducts(List<String>
productNames) async {
68      final firestore = FirebaseFirestore.instance;
69      final List<Map<String, dynamic>> matchedProducts = [];
70
71      for (final name in productNames) {
72          final querySnapshot = await firestore
73              .collection('products')
74              .where('name', isEqualTo: name)
75              .get();
76
77          for (final doc in querySnapshot.docs) {
78              matchedProducts.add(doc.data());
79          }
80      }
81
82      return matchedProducts;
83  }
84

```

Рисунок 3.6 – пошук даних за назвами які повернув OpenAI

Перевагою описаної системи є її гнучкість і масштабованість. GPT-4o дозволяє легко змінювати стиль, глибину та формат рекомендацій без необхідності переписування алгоритмічної логіки на стороні клієнта. Достатньо змінити

формулювання `system prompt` – і результат зміниться відповідно до нових критеріїв. Крім того, система не вимагає великих обчислювальних ресурсів на стороні застосунку, що знижує вимоги до пристрою та збільшує потенційний пул користувачів.

Однак є і певні обмеження. Оскільки рекомендації формуються текстово, система не має прямого доступу до векторних представлень товарів або користувачів, що властиво класичним нейромережам. Це може знижувати точність у випадках великої кількості схожих позицій. Також можливе повернення товарів, які більше не доступні – що вимагає постійного актуалізування даних у базі. Важливо також враховувати, що API-запити до OpenAI є платними, тож їхня частота має бути оптимізована залежно від бюджету.

Попри зазначені обмеження, запропонована архітектура демонструє високу ефективність при малій вартості реалізації, що робить її практичним рішенням для проєктів малого та середнього масштабу. У подальшому система може бути легко розширена, наприклад, шляхом впровадження векторного пошуку або кешування відповідей для повторного використання.

3.4. Проєктування та реалізація інтерфейсу користувача (UI/UX)

Проєктування інтерфейсу користувача у рамках мобільного застосунку відбувалося з урахуванням принципів інтуїтивної навігації, функціональної зрозумілості та адаптації до широкого спектра мобільних пристроїв. Кожен елемент було реалізовано з метою забезпечити швидкий доступ до основних функцій та мінімізувати когнітивне навантаження на користувача.

Ключові екрани мобільного застосунку (див. Додаток Б):

- **Авторизація.** Дозволяє користувачеві увійти в систему через форму логіну за email/паролем, реалізовану за допомогою Firebase Authentication. У разі успішної авторизації ініціалізується сесія, а дані профілю підтягуються з бази `users`.

- **Головна сторінка (рис. 3.7).** Служить основним хабом взаємодії. Містить вітальний блок, список категорій товарів, акційні банери, а також динамічну секцію персоналізованих рекомендацій, які оновлюються після кожної активності користувача. Інформація підтягується з Firestore у режимі реального часу.

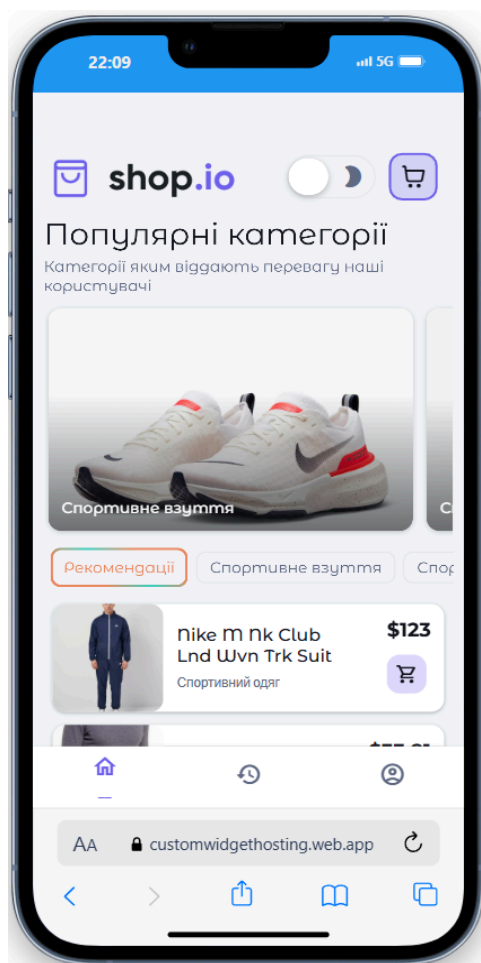


Рисунок 3.7 – Головна сторінка

- **Детальний перегляд товару.** Відображає зображення товару (з Firebase Storage), його опис, характеристики (розміри, кольори, доступність), рейтинг і відгуки. Також присутні кнопки для додавання до кошика або переходу до подібних товарів.

- **Профіль користувача.** Демонструє особисті дані користувача (ім'я, email, розташування, аватар), які зберігаються у відповідному документі в колекції users. Передбачена можливість редагування та виходу з акаунта.
- **Історія покупок.** Відображає перелік усіх замовлень користувача, які автоматично фіксуються у колекції orders. Кожне замовлення містить дату, перелік товарів, підсумкову вартість та статус виконання.
- **Панель рекомендацій (рис. 3.8).** Спеціальний екран, на якому користувач бачить сформовані на основі його дій пропозиції від моделі GPT-4o. Можна оновлювати список, переглядати деталі товарів і додавати їх до кошика.

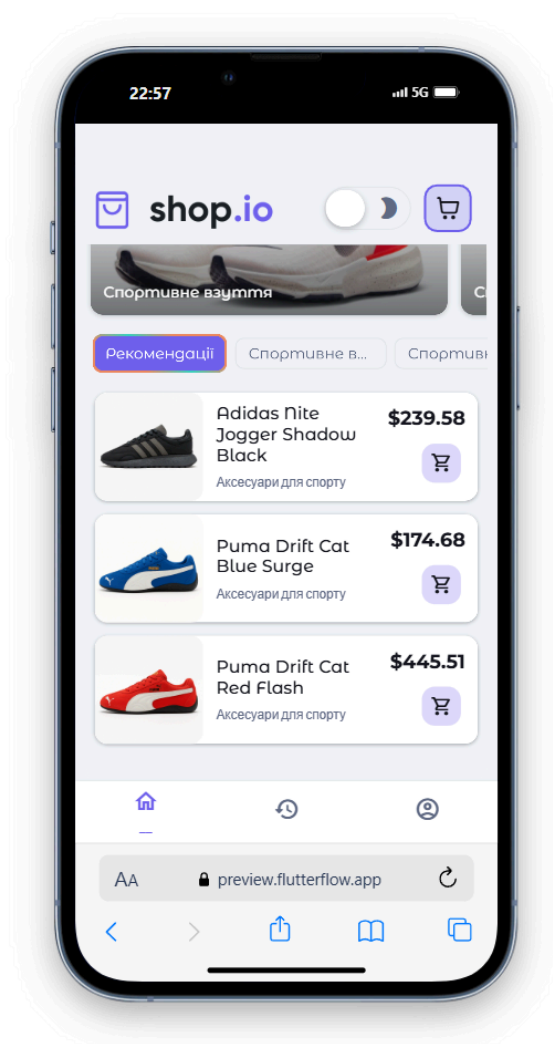


Рисунок 3.8 – Панель рекомендацій

- **Кошик.** Містить усі вибрані товари перед оформленням покупки. Користувач може змінювати кількість, видаляти позиції, обирати тип доставки та підтверджувати замовлення.

Інтерфейс було реалізовано з використанням **Flutter SDK** на основі **Material Design**, що забезпечує візуальну єдність і доступність. Для зберігання та управління станом застосовано патерн **Provider**, а для побудови динамічного контенту – **FutureBuilder** з асинхронною логікою завантаження даних.

Додаткові аспекти дизайну:

- **Мінімалістичний підхід.** Усі інтерфейси побудовані з акцентом на візуальність товарів: великі зображення, чиста типографіка, мінімум навігаційного шуму.
- **Гнучка адаптивність.** Кожен компонент верстається з урахуванням можливості адаптації до різних діагоналей екранів, що дозволяє комфортно працювати з додатком як на смартфонах, так і на планшетах.
- **Анімації та плавність.** Для покращення користувацького досвіду реалізовані переходи між екранами, fade-завантаження зображень і анімація елементів списку.
- **Темна тема.** Додаток підтримує перемикання між світлим і темним режимами, що дозволяє знизити навантаження на очі користувачів у різних умовах освітлення.
- **Інтерактивність.** Багато елементів (наприклад, списки рекомендацій, кнопки "Додати до кошика", фільтри) реалізовані як реактивні, що сприяє створенню живого та адаптивного досвіду взаємодії.

Загалом, дизайн інтерфейсу побудований так, щоб гармонійно поєднати естетику, функціональність і швидкодію, необхідні для успішної роботи в динамічному середовищі мобільної електронної комерції.

3.5. Тестування та відлагодження

Однією з найважливіших складових у розробці мобільного застосунку є перевірка стабільності та коректності взаємодії з зовнішніми сервісами –

насамперед Firebase та OpenAI API. Тестування цих компонентів дозволило переконатися в правильності передачі, обробки та збереження даних, а також у релевантності згенерованих рекомендацій.

Firebase API:

Було протестовано всі основні запити до Firestore та Firebase Authentication. Зокрема, перевірялися:

- Реєстрація та вхід користувача за допомогою email і пароля.
- Створення та оновлення документів у колекціях users, orders, activity.
- Отримання списку товарів з колекції products із використанням фільтрів за категорією та рейтингом.
- Слухачі Firestore (StreamBuilder) на головному екрані, які мають оновлювати контент у реальному часі.

Було підтверджено, що усі операції виконуються асинхронно, з обробкою помилок (наприклад, втрати з'єднання або відсутності прав доступу).

OpenAI API:

Тестування взаємодії з GPT-4o здійснювалося через інтерфейс Postman, а також безпосередньо у застосунку. Перевірялося:

- Формування prompt-запитів на основі поведінки користувача (перегляди, покупки).
- Обробка текстових відповідей моделі, зокрема – валідація JSON-структур або парсинг списків рекомендацій.
- Обробка виключень: перевищення ліміту (рис. 3.9), тайм-аути, некоректна відповідь.

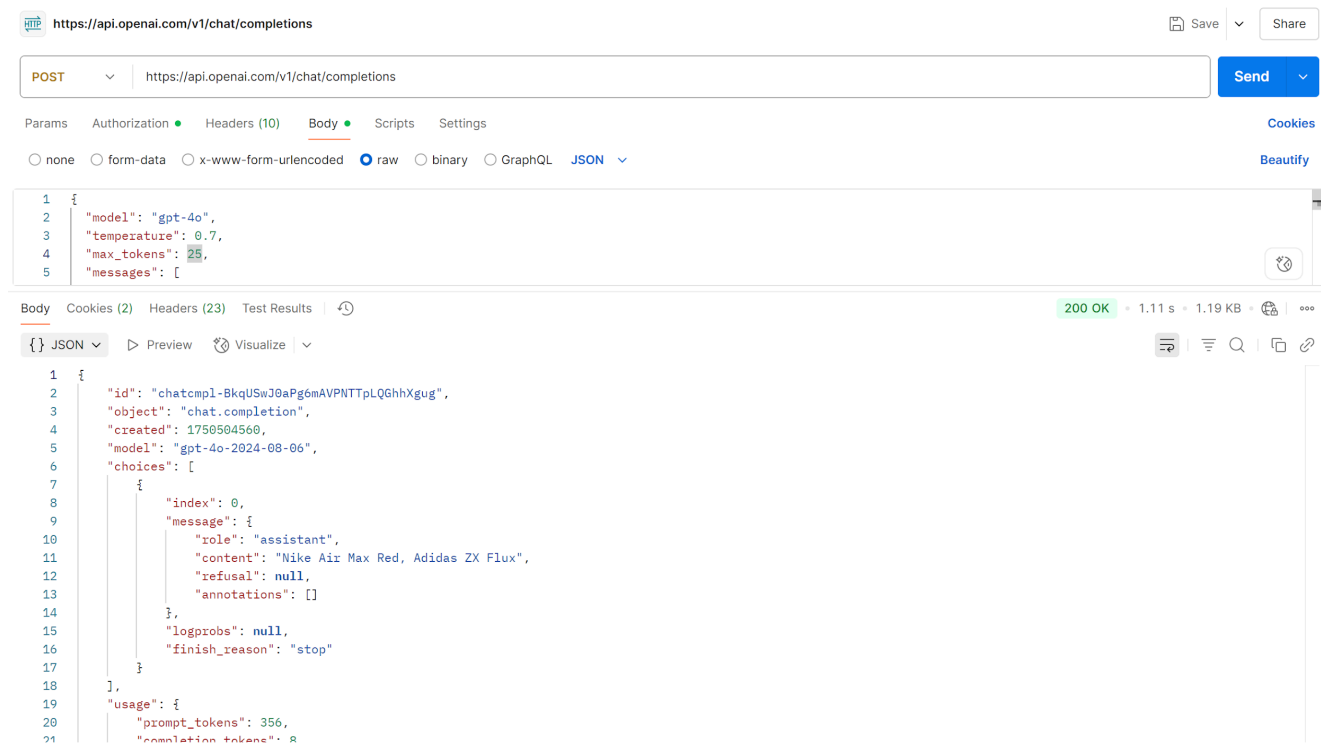


Рисунок 3.9 –Тестування обмеженої кількості кредитів

Результати показали, що GPT-4o успішно формує відповідні рекомендації навіть на основі обмеженої інформації, а структура відповіді є стабільною для подальшої обробки клієнтським кодом

ВИСНОВОК

1. У ході дослідження було здійснено аналіз сучасного стану розвитку систем рекомендацій в електронній комерції. Було розглянуто актуальні підходи до персоналізації, класифікацію рекомендаційних систем (контентні, колаборативні, гібридні) та ключові проблеми їхнього впровадження в e-commerce-середовище. Особливий акцент зроблено на гібридних моделях, які поєднують переваги різних підходів і, завдяки цьому, досягають вищої точності у формуванні персоналізованих рекомендацій.
2. Розглянуто архітектурні моделі, що поєднують нейромережеві алгоритми, машинне навчання та генеративний ШІ. Дослідження показало, що GPT-4o добре підходить для покращення персоналізованих рекомендацій і справляється з цим завданням дуже ефективно.
3. Обґрунтовано вибір технологічного стеку: Flutter як зручний кросплатформовий фреймворк для створення адаптивного інтерфейсу, Firebase як хмарна база даних з широким функціоналом, а також OpenAI API для реалізації рекомендаційної логіки. Рішення були підібрані відповідно до функціональних і нефункціональних вимог до системи.
4. Реалізовано гібридну систему персоналізованих рекомендацій, інтегровану в інтернет-магазин одягу. Така реалізація дозволяє враховувати як контентні характеристики товарів (категорія, колір, стиль, сезонність), так і поведінкові сигнали користувача (перегляди, додавання до обраного, покупки). З технічного боку гібридна система поєднує прості евристики з контентною фільтрацією та API GPT-4o, який використовується для уточнення інтересів і генерації рекомендацій у діалоговій формі. Це дає змогу не лише формувати релевантні товарні добірки, а й підтримувати динамічну взаємодію з користувачем, адаптуючи пропозиції в реальному часі.
5. Створено мобільну платформу інтернет-магазину з модулем рекомендацій, базою даних та структурою взаємодії з користувачем. Розроблено ключові екрани (авторизація, головна, товар, кошик, профіль, історія покупок,

рекомендації), реалізовано логіку збору поведінкових даних та запитів до GPT-4o, який формує текстові рекомендації.

6. Проведено базове тестування платформи. Тестування показало коректність роботи взаємодії з Firebase, ефективність генерації рекомендацій, адаптивність інтерфейсу та релевантність отриманих пропозицій. Результати підтверджують працездатність системи та її потенціал до масштабування.

Усі поставлені завдання дипломної роботи виконано. Мета дослідження досягнута. Результати розробки мають високу прикладну цінність і можуть бути основою для подальшого вдосконалення системи персоналізованих рекомендацій у сфері електронної комерції.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. How Businesses Are Using Artificial Intelligence [Electronic resource] // Forbes. – 2023. – Access: <https://www.forbes.com/advisor/business/software/ai-in-business/>
2. How Artificial Intelligence Is Transforming Business [Electronic resource] // The University of Arizona Global Campus. – 2024. – Access: <https://www.uagc.edu/blog/how-artificial-intelligence-transforming-business/>
3. Kapil Saini, Hybrid Recommender System for E-Commerce: A Comprehensive Review and Future Directions [Electronic resource] //ResearchGate. – 2023. – Access: https://www.researchgate.net/publication/373165499_Hybrid_Recommender_System_for_E-Commerce_A_Comprehensive_Review_and_Future_Directions/
4. Mohd Aarif, Leveraging GPT-4 Capabilities for Context-Aware Personalized Chatbots in E-commerce [Electronic resource] //ResearchGate. – 2024. – Access: https://www.researchgate.net/publication/381230601_Leveraging_GPT-4_Capabilities_for_Developing_Context-Aware_Personalized_Chatbot_Interfaces_in_E-commerce_Customer_Support_Systems
5. Hybrid Quality-Based Recommender Systems: A Systematic Review [Electronic resource] //MDPI. – 2024. – Access: https://www.mdpi.com/2313-433X/11/1/12?utm_source=chatgpt.com
6. E. Drachsler, H. Hummel, R. Koper. Personal Recommender Systems for Learners in Lifelong Learning Networks [Electronic resource] // ACM Digital Library. – 2008. – Access: <https://dl.acm.org/doi/10.1145/1363686.1363693>
7. Gediminas Adomavicius, Alexander Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions [Electronic resource] // IEEE Transactions on Knowledge and Data Engineering. – 2005. – Access: <https://ieeexplore.ieee.org/document/1423975>
8. Christoph Trattner, David Elsweiler. Food Recommender Systems: Important Contributions, Challenges and Future Research Directions [Electronic resource] // arXiv. – 2017. – Access: <https://arxiv.org/abs/1711.02760>

9. B. Gedik et al. Building a Hybrid Recommender System for E-commerce [Electronic resource] // IEEE Xplore. – 2020. – Access: <https://ieeexplore.ieee.org/document/9073039>
10. Microsoft Azure Blog. Leverage Large Language Models such as ChatGPT for your Ecommerce System [Electronic resource] – 2023. – Access: <https://techcommunity.microsoft.com/t5/ai-cognitive-services-blog/leverage-large-language-models-such-as-chatgpt-for-your/ba-p/3800412>
11. F. Ricci, L. Rokach, B. Shapira. Introduction to Recommender Systems Handbook [Electronic resource] // SpringerLink. – 2011. – Access: <https://link.springer.com/book/10.1007/978-0-387-85820-3>
12. OpenAI Community. Using GPT to build a Recommender System [Electronic resource] // OpenAI Forum. – 2023. – Access: <https://community.openai.com/t/using-gpt-to-build-a-recommender-system/250678>
13. R. Burke. Hybrid Recommender Systems: Survey and Experiments [Electronic resource] // User Modeling and User-Adapted Interaction. – 2002. – Access: <https://link.springer.com/article/10.1023/A:1021240730564>
14. Zhao, X. et al. Deep Reinforcement Learning for List-wise Recommendations [Electronic resource] // ACM Conference. – 2018. – Access: <https://dl.acm.org/doi/10.1145/3219819.3219901>
 Zhang, S. et al. Deep Learning based Recommender System: A Survey and New Perspectives [Electronic resource] // ACM Computing Surveys. – 2019. – Access: <https://dl.acm.org/doi/10.1145/3285029>
15. Cheng, H. et al. Wide & Deep Learning for Recommender Systems [Electronic resource] // arXiv. – 2016. – Access: <https://arxiv.org/abs/1606.07792>
16. P. Resnick, H. Varian. Recommender Systems [Electronic resource] // Communications of the ACM. – 1997. – Access: <https://dl.acm.org/doi/10.1145/245108.245121>
17. Medium. Mastering E-Commerce Product Recommendations in Python [Electronic resource] – 2023. – Access: <https://medium.com/data-science-at-microsoft/mastering-e-commerce-product-recommendations-in-python-1e2ec3310e1d>

- 18.Kumar, V. et al. A Personalized Product Recommender System Using Hybrid Filtering Technique [Electronic resource] // IEEE Access. – 2022. – Access: <https://ieeexplore.ieee.org/document/9874396>
- 19.Google Cloud Blog. How to Build a Recommendation Engine with Vertex AI and BigQuery ML [Electronic resource] – 2022. – Access: <https://cloud.google.com/blog/products/ai-machine-learning/building-a-recommendation-engine-vertex-ai-bigquery-ml>
- 20.Amazon Science. Personalized Recommendations at Amazon Scale [Electronic resource] – 2022. – Access: <https://www.amazon.science/latest-news/personalized-recommendations-at-amazon-scale>
- 21.Wikipedia. Hybrid recommender system [Electronic resource] – Access: https://en.wikipedia.org/wiki/Hybrid_recommender_system
- 22.ArXiv.org. A Survey of Recommender Systems Based on Deep Learning [Electronic resource] – 2020. – Access: <https://arxiv.org/abs/2004.11149>

ДОДАТОК А

Програмна реалізація алгоритму

```
import 'dart:convert';

import 'package:flutter/material.dart';

import 'package:cloud_firestore/cloud_firestore.dart';

import 'package:http/http.dart' as http;

import 'package:firebase_core/firebase_core.dart';

void main() async {

  WidgetsFlutterBinding.ensureInitialized();

  await Firebase.initializeApp();

  runApp(const MyApp());

}

class MyApp extends StatelessWidget {

  const MyApp({super.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: RecommendationScreen(),

    );

  }

}

class RecommendationScreen extends StatefulWidget {

  @override
```

```

State<RecommendationScreen> createState() => _RecommendationScreenState();

}

class _RecommendationScreenState extends State<RecommendationScreen> {

  String recommendations = "";

  bool loading = false;

  Future<void> fetchProductsAndAskGPT() async {

    setState() {

      loading = true;

    });

    final firestore = FirebaseFirestore.instance;

    // 1. Отримуємо товари з Firestore

    final snapshot = await firestore.collection("products").get();

    final products = snapshot.docs.map((doc) {

      final data = doc.data();

      return {

        "id": doc.id,

        "name": data['name'] ?? "",

      };

    }).toList();

    final firestoreJson = Map.fromEntries(

      products.map((item) => MapEntry(item['id'], {'name': item['name']})),

    );

    final firestoreJsonString = jsonEncode(firestoreJson);

```

```

const apiKey = 'YOUR_OPENAI_API_KEY';

const apiUrl = 'https://api.openai.com/v1/chat/completions';

final body = {

  "model": "gpt-4o",

  "temperature": 0.7,

  "messages": [

    {

      "role": "system",

      "content": "Ти асистент персональних покупок у мобільному магазині одягу. "

      "На основі історії покупок і переглядів користувача, порадь товари з каталогу, "

      "які можуть йому сподобатись. Враховуй категорії, описи, унікальність товарів і "

      "подібність до вже переглянутих чи придбаних. Повертай тільки перелік назв через кому."

    },

    {

      "role": "user",

      "content": "Ось наявна база даних товарів у магазині:\n\n$firestoreJsonString\n\n"

      "Врахуй цю БД як джерело для рекомендацій на основі дій користувача."

    },

    {

      "role": "user",

      "content": "Ось історія користувача:\n\n"

      "Купив:\n"

      "- Air Jordan 1 UNC Sky Blue (Shoes, Limited edition)\n\n"

```

"Переглядав:\n"

"- Adidas Response Trail Blue Frost (Shoes, Трекінгові кросівки)\n\n"

"Каталог товарів:\n"

"- Nike Air Max Red – Shoes – Спортивні кросівки для бігу\n"

"- Adidas ZX Flux – Shoes – Повсякденне взуття\n"

"- Nike T-Shirt Blue – Clothes – Бавовняна футболка\n"

"- Puma Socks Black – Accessories – Чорні базові шкарпетки\n\n"

"Які речі ти можеш порадити цьому користувачу?"

}

]

};

final response = await http.post(

Uri.parse(apiUrl),

headers: {

'Content-Type': 'application/json',

'Authorization': 'Bearer \$apiKey',

},

body: jsonEncode(body),

);

if (response.statusCode == 200) {

final data = jsonDecode(response.body);

final reply = data['choices'][0]['message']['content'];

setState() {

```

    recommendations = reply;

    loading = false;

  });

} else {

  setState(() {

    recommendations = 'Помилка: ${response.statusCode}\n${response.body}';

    loading = false;

  });

}

}

}

@override

Widget build(BuildContext context) {

  return Scaffold(

    appBar: AppBar(title: const Text('Рекомендації GPT')),

    body: Padding(

      padding: const EdgeInsets.all(16.0),

      child: Column(

        children: [

          ElevatedButton(

            onPressed: fetchProductsAndAskGPT,

            child: const Text('Отримати рекомендації'),

          ),

          const SizedBox(height: 20),

```

```

    if (loading)

      const CircularProgressIndicator()

    else if (recommendations.isNotEmpty)

      Expanded(

        child: SingleChildScrollView(

          child: Text(

            recommendations,

            style: const TextStyle(fontSize: 16),

          ),

        ),

      )

    else

      const Text('Натисни кнопку, щоб отримати поради.'),

  ],

),

),

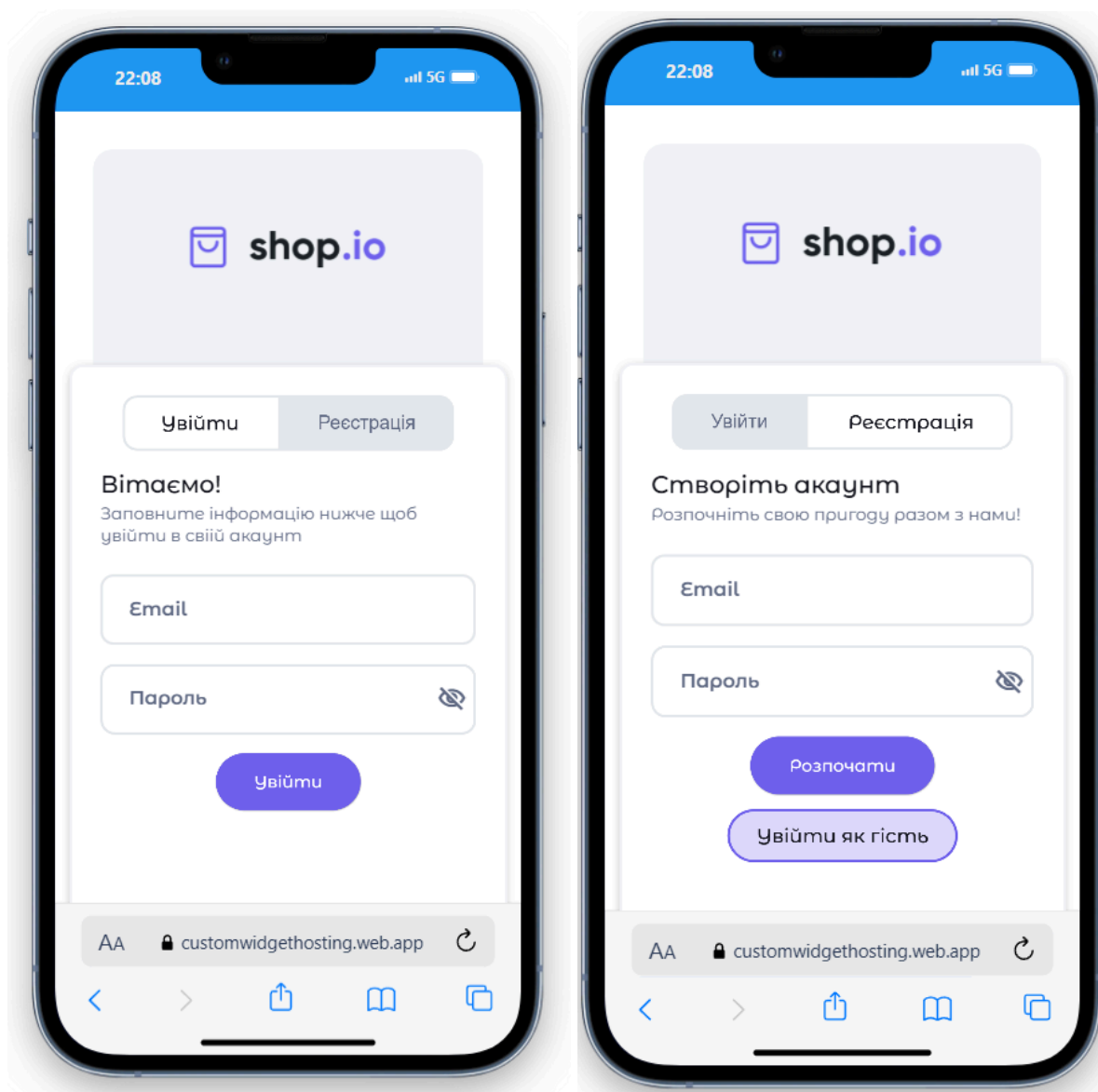
);

}

}

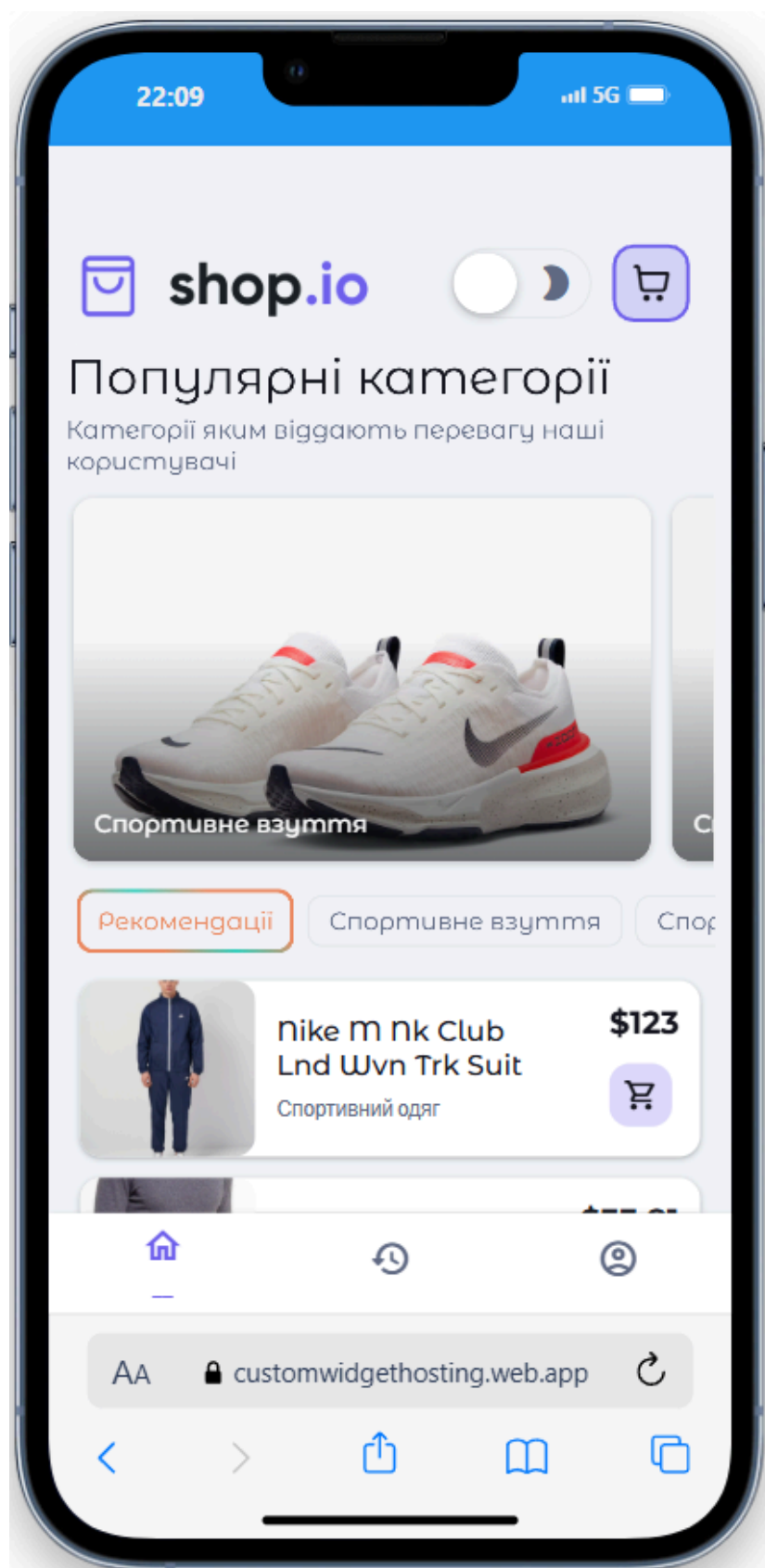
```

ДОДАТОК Б



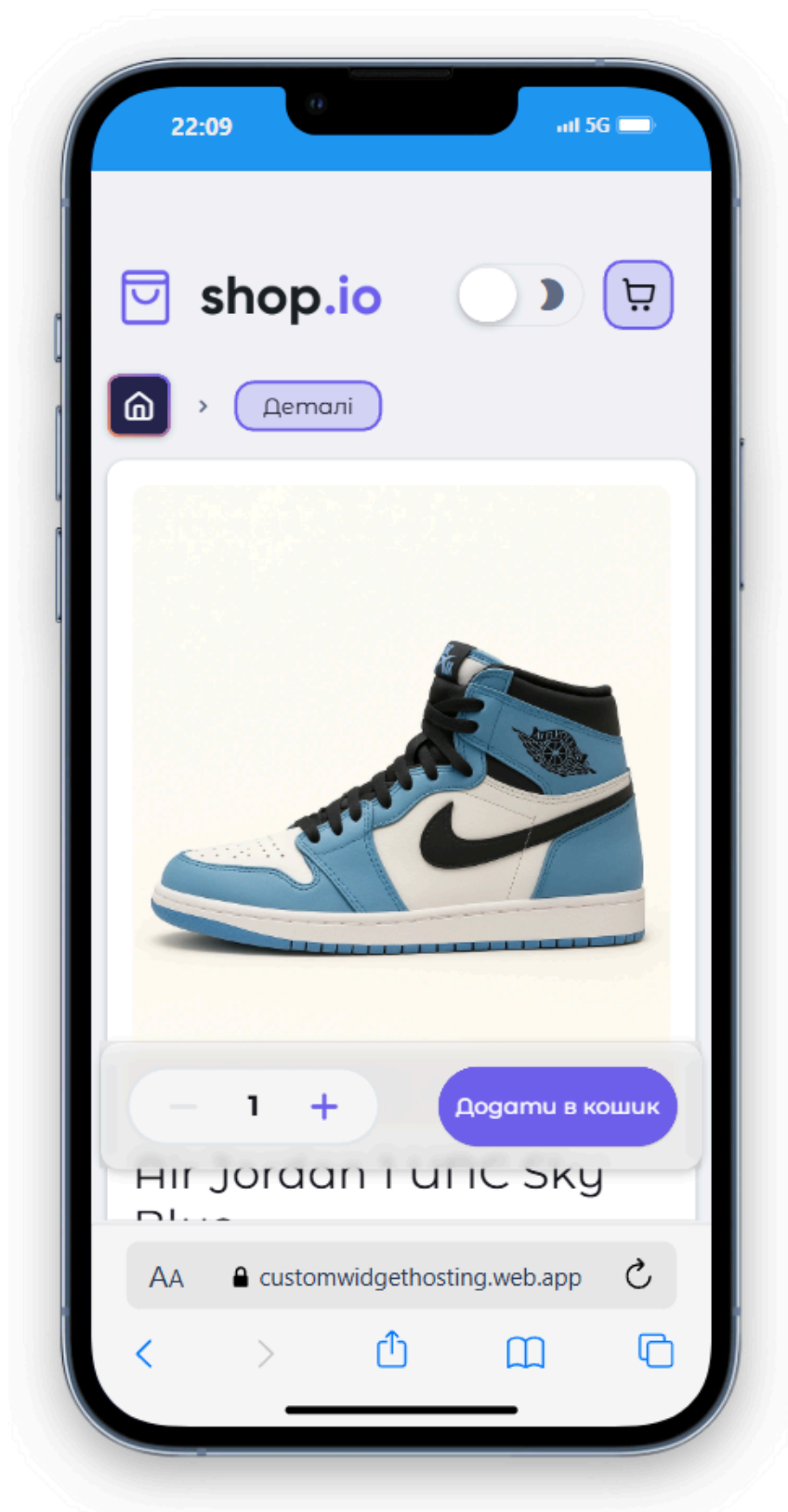
Сторінка авторизації

Продовження додатку Б

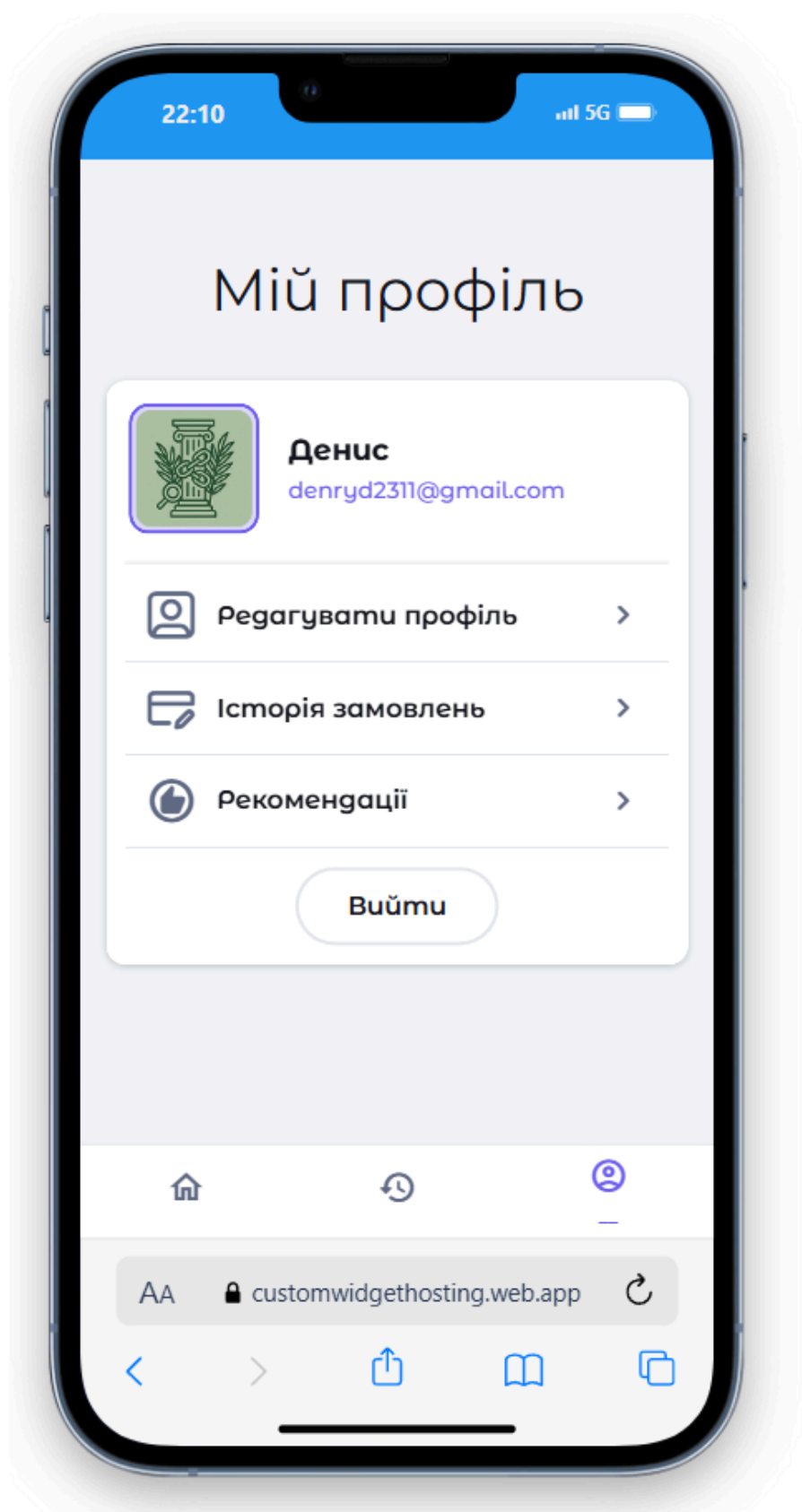


Головна сторінка мобільного застосунку

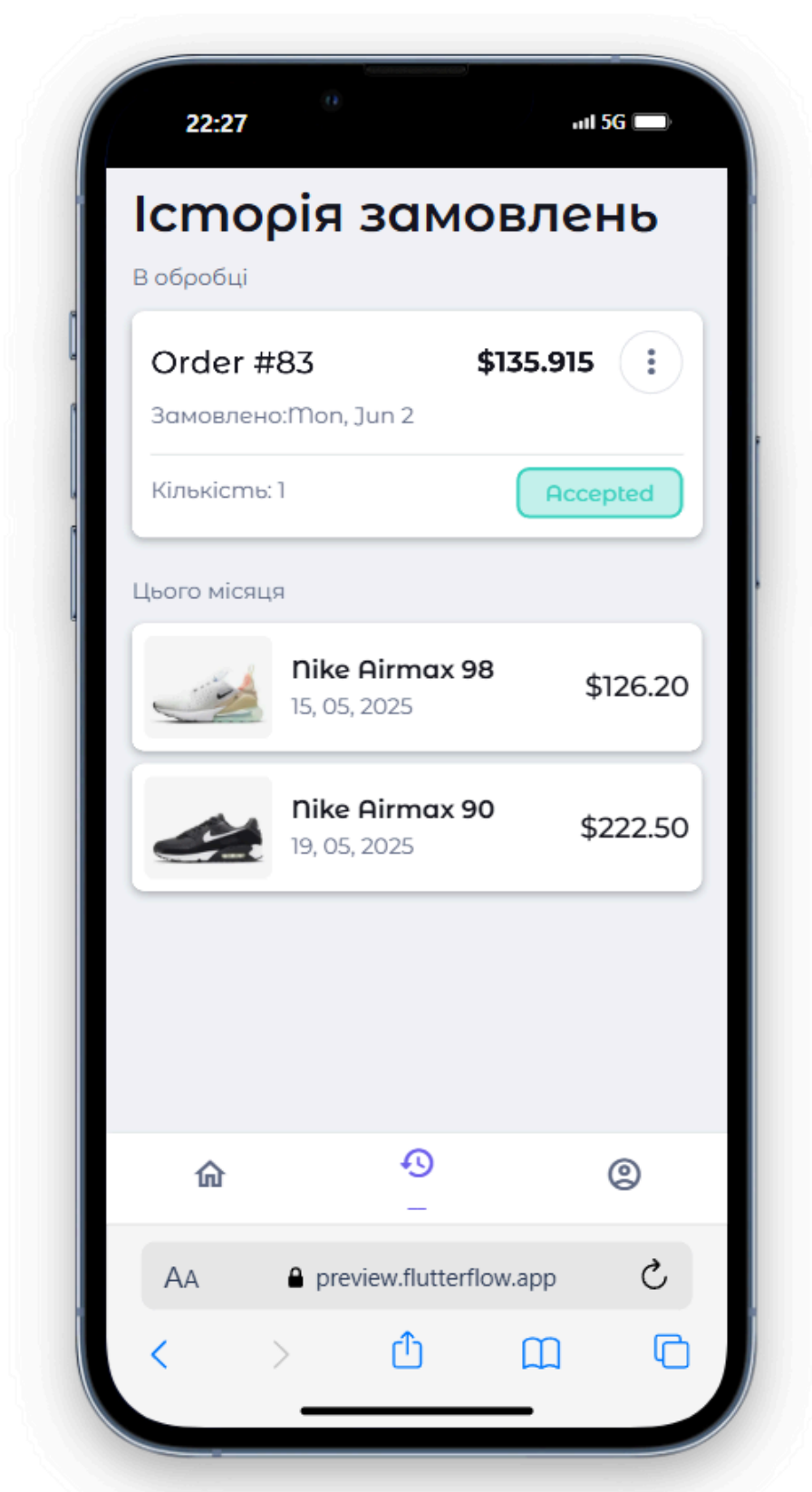
Продовження додатку Б



Сторінка детального перегляду товару

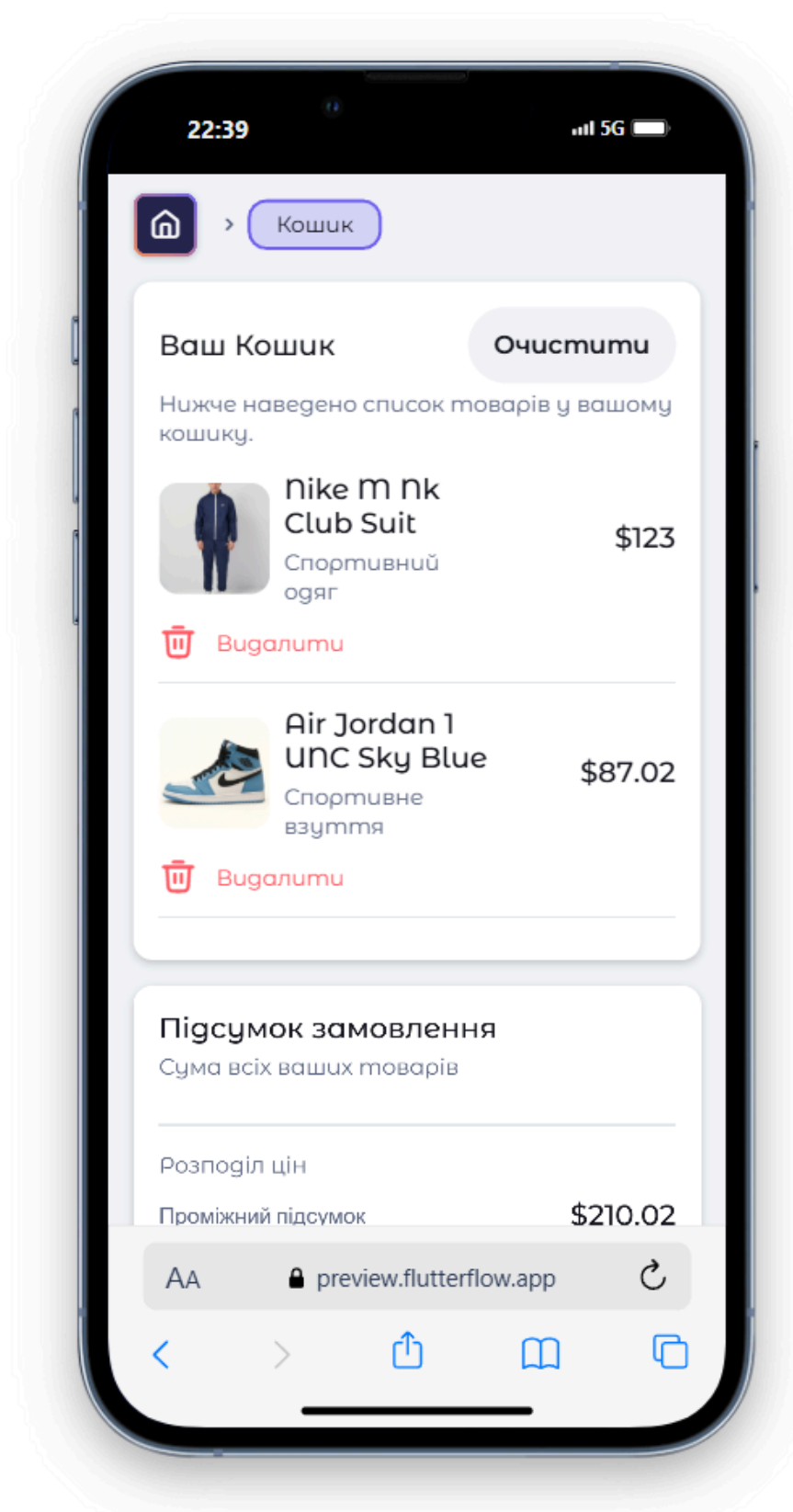


Сторінка профілю



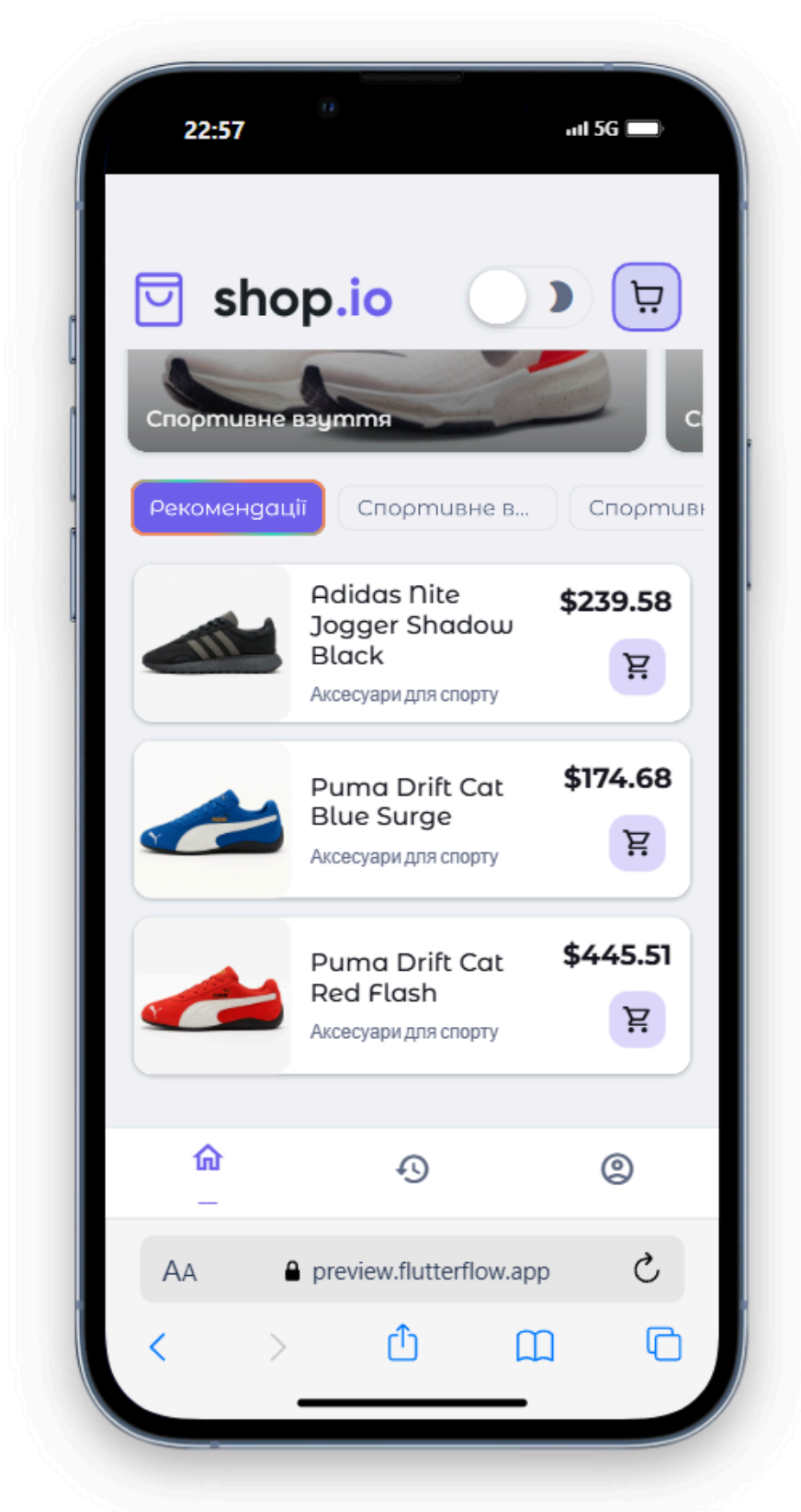
Сторінка історії покупок

Продовження додатку Б



Кошик

Продовження додатку Б



Панель рекомендацій