

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

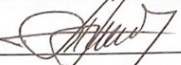
Бакалаврська кваліфікаційна робота на тему:

**«РЕКОМЕНДАЦІЙНА ІНФОРМАЦІЙНА СИСТЕМА ЩОДО ПРОДАЖУ
КОМП'ЮТЕРНОЇ ТЕХНІКИ НА ОСНОВІ АНАЛІЗУ ВПОДОБАНЬ
КОРИСТУВАЧІВ»**

Виконав: студент 4 курсу, групи ЗІСТ-216
спеціальності 126 – «Інформаційні
системи та технології»

 Олександр СТЕЦЮК

Керівник: к.т.н., доц. каф. САІТ

 Сергій ЖУКОВ

«04» 06 2025 р.

Рецензент: к.т.н., доц. каф. КН

 Володимир Озеранський

«08» 06 2025 р.

Допущено до захисту

Завідувач кафедри САІТ

 д.т.н., проф. Віталій МОКІН

«06» 06 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність 126 – «Інформаційні системи та технології»
Освітньо-професійна програма – Прикладні інформаційні технології

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

Анто д.т.н., проф. Віталій МОКІН

"18" 03 2025 року

ЗАВДАННЯ НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Стецюку Олександр Романовичу

1. Тема роботи: «Рекомендаційна інформаційна система щодо продажу комп'ютерної техніки на основі аналізу вподобань користувачів»

керівник роботи: Жуков С. О., к.т.н., доц. каф. САІТ

затверджені наказом ВНТУ від «10» 03 2025 року № 97

2. Термін подання студентом роботи 31.05.25

3. Вихідні дані до роботи:

1) Набір даних про історію взаємодії користувачів з товарами (перегляди, вподобання, додавання до обраного, покупки).

2) Інформація про характеристики комп'ютерної техніки з онлайн-каталогів.

4. Зміст текстової частини:

1) Аналіз предметної області.

2) Вибір технологій реалізації системи рекомендацій.

3) Розроблення інформаційної системи рекомендацій щодо продажу комп'ютерної техніки.

5. Перелік ілюстративного матеріалу:

1) Структурна схема функціонування веб-застосунку;

2) Діаграма випадків використання веб-застосунку (Use Case Diagram);

3) Діаграма класів веб-застосунку;

4) UML-діаграма послідовності;

5) Блок-схема алгоритму роботи додатку;

6) Інтерактивна схема інфраструктури;

7) Інтерфейс інформаційної системи.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 21.03.15

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	01.04	10.04	вик
2	Порівняльний аналіз проблематики	10.04	20.04	вик
3	Вибір оптимальних інформаційних технологій	20.04	30.04	вик
4	Розроблення інформаційної системи автоматизації роботи користувачів	1.05	15.05	вик
5	Оформлення матеріалів до захисту БКР	15.05	30.05	вик

Студент



Олександр СТЕЦІЮК

Керівник роботи



Сергій ЖУКОВ

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота містить 78 сторінок формату А4, 29 рисунки, список використаних джерел містить 20 найменувань.

В роботі розроблено рекомендаційну інформаційну систему щодо продажу комп'ютерної техніки на основі аналізу вподобань користувачів. У роботі реалізовано веб-застосунок із використанням сучасних технологій: React, TypeScript, Firebase, Tailwind CSS. Система інтегрує алгоритми колаборативної фільтрації, штучного інтелекту та перевірки сумісності комплектуючих. Особливу увагу приділено персоналізації рекомендацій і адаптації інтерфейсу до рівня користувача. У результаті створено зручний інструмент для підбору конфігурацій ПК відповідно до технічних вимог, бюджету та особистих вподобань користувача.

Ключові слова: рекомендаційна система, комп'ютерна техніка, вподобання користувача, веб-застосунок, React, Firebase, колаборативна фільтрація, інформаційна система.

ABSTRACT

The bachelor's qualification thesis consists of 78 A4 pages, 29 figures, and a list of 20 sources.

A recommendation information system for selling computer hardware based on user preference analysis was developed. A web application was developed using modern technologies such as React, TypeScript, Firebase, and Tailwind CSS. The system integrates collaborative filtering algorithms, artificial intelligence, and component compatibility checking. Special attention is given to recommendation personalization and an adaptive user interface. As a result, a convenient tool was created for selecting PC configurations based on technical requirements, budget, and individual user preferences.

Key words: recommendation system, computer hardware, user preferences, web application, React, Firebase, collaborative filtering.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Обґрунтування актуальності розробки рекомендаційної системи щодо продажу комп'ютерної техніки	6
1.2 Сучасні підходи до аналізу вподобань користувачів з використанням AI	7
1.3 Огляд аналогів рекомендаційних систем у сфері продажу електроніки.....	10
1.4 Постановка задачі дослідження	14
1.5 Інтерфейс взаємодії з користувачами в системах підбору комп'ютерної техніки	15
1.6 Висновки.....	16
2 ВИБІР ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ СИСТЕМИ РЕКОМЕНДАЦІЙ	18
2.1 Вибір оптимальної ІТ-інфраструктури	18
2.2 Обґрунтування вибору мови програмування та фреймворків.....	19
2.3 Аналіз середовища розробки для веб-застосунків на основі React та TypeScript	22
2.4 Вибір та інтеграція технологій штучного інтелекту для рекомендаційної системи	24
2.5 Методи аналізу вподобань користувачів для формування рекомендацій	25
2.6 Висновки.....	27
3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ РЕКОМЕНДАЦІЙ ЩОДО ПРОДАЖУ КОМП'ЮТЕРНОЇ ТЕХНІКИ.....	29
3.1 Розроблення загального алгоритму роботи додатку	29
3.2 Розроблення загальної структурної схеми функціонування веб-застосунку рекомендаційної системи щодо продажу комп'ютерної техніки	32
3.3 Розроблення архітектури інформаційної системи	34
3.4 Програмна реалізація інформаційної системи	40
3.5 Тестування роботи системи	47
3.6 Висновки.....	53
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
Додаток А (обов'язковий). Технічне завдання.....	58
Додаток Б (обов'язковий).Протокол перевірки кваліфікаційної роботи	61

	3
Додаток В (довідниковий). Фрагмент лістингу програми	62
Додаток Г (обов'язковий). Ілюстративна частина	68

ВСТУП

Актуальність теми. У сучасному світі правильний вибір комп'ютерної техніки став важливим завданням для багатьох користувачів. Більшість покупців стикаються з труднощами при виборі оптимальних комплектуючих, що часто призводить до придбання несумісних компонентів або переплати за надлишкові характеристики. Відсутність зручних систем рекомендацій змушує користувачів витрачати багато часу на дослідження ринку та порівняння технічних характеристик.

Сучасні веб-технології дозволяють створювати розумні системи, які значно спрощують процес вибору комп'ютерної техніки. Впровадження рекомендаційних систем у сферу електронної комерції допомагає підвищити точність підбору товарів та покращити користувацький досвід. Особливо цінними є системи, що поєднують зручний інтерфейс з інтелектуальними алгоритмами аналізу вподобань користувачів.

Мета і задачі дослідження. Метою дослідження є підвищення ефективності процесу підбору комп'ютерної техніки шляхом розроблення рекомендаційної інформаційної системи з інтеграцією хмарних сервісів, AI-консультанта та модульної адміністративної панелі.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- Провести аналіз предметної області та існуючих рішень для рекомендаційних систем комп'ютерної техніки;
- Вибрати оптимальні технології для розробки веб-застосунку;
- Розробити архітектуру інформаційної системи з урахуванням потреб різних користувачів;
- Створити функціонал для каталогізації, аналізу вподобань та формування персоналізованих рекомендацій;
- Інтегрувати модуль аналізу для підвищення точності підбору комплектуючих;
- Розробити механізм перевірки технічної сумісності компонентів;

– Протестувати систему для оцінки ефективності та точності рекомендацій.

Об'єктом дослідження є процес розроблення рекомендаційної інформаційної системи для електронної комерції комп'ютерної техніки на основі аналізу вподобань користувачів.

Предметом дослідження є методи та програмні засоби створення рекомендаційної інформаційної системи щодо продажу комп'ютерної техніки на основі аналізу вподобань користувачів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Обґрунтування актуальності розробки рекомендаційної системи щодо продажу комп'ютерної техніки

Сучасний ринок комп'ютерної техніки характеризується надзвичайною різноманітністю пропозицій, стрімким розвитком технологій та складністю вибору оптимальних комплектуючих. Однак значна частина користувачів стикається з проблемою браку технічних знань для самостійного підбору компонентів, аналізу їх сумісності та оцінки співвідношення ціна/продуктивність. У цьому контексті розробка автоматизованої рекомендаційної інформаційної системи для підбору комп'ютерної техніки на основі аналізу вподобань користувачів набуває особливої актуальності.

1. Тенденції та проблеми сучасного ринку комп'ютерних комплектуючих. Підбір оптимальних комплектуючих став невід'ємною частиною процесу придбання комп'ютерної техніки, однак самостійне складання конфігурації є досить складним та потребує спеціальних знань в галузі апаратного забезпечення. Згідно з даними міжнародних досліджень, велика кількість користувачів придбаває несумісні або надлишкові компоненти, що призводить до проблем з експлуатацією, таких як неоптимальна продуктивність, перевитрати або несумісність [1]. Автоматизована система, яка враховує індивідуальні потреби користувача та його бюджетні обмеження, може значно спростити процес вибору комп'ютерної техніки.

2. Персоналізація як основний фактор ефективності. Однією з ключових особливостей сучасних технологій є можливість персоналізованого підходу до формування рекомендацій [2]. Використання класичних методів аналізу вподобань користувачів у поєднанні з AI-технологіями дозволяє адаптувати пропозиції до індивідуальних потреб кожного користувача. Розроблений веб-застосунок інтегрує штучний інтелект для аналізу поведінки користувача, що забезпечує гнучкість системи та зменшує необхідність ручного пошуку серед тисяч пропозицій.

3. Автоматизація та її вплив на якість вибору. Традиційний підхід до вибору комп'ютерної техніки передбачає ручне порівняння характеристик та цін, що може бути складним і неточним без належних знань. Автоматизований підхід дозволяє мінімізувати ризик помилок та оптимізувати процес формування конфігурації. Розробка веб-застосунку спрямована на створення зручної та швидкої системи, яка в режимі реального часу аналізує введені дані, вподобання користувача та надає точні результати відповідно до цілей та бюджету.

4. Кросплатформність та доступність. Важливим фактором сучасних технологічних рішень є їх доступність для широкої аудиторії. Використання кросплатформених технологій (React [3], TypeScript [4], Firebase [5]) дозволяє створити єдиний застосунок з адаптивним дизайном та можливістю швидкого оновлення та впровадження нових функцій. Завдяки цьому розроблений застосунок буде доступним для різних типів користувачів — незалежно від їх пристрою або рівня технічної підготовки.

5. Перспективи розвитку та суспільна значущість. Автоматизовані рекомендаційні системи для підбору комп'ютерної техніки мають значний потенціал для впровадження у сфері електронної комерції, геймінга та професійного використання. Далі може бути інтеграція розширених AI-модулів для ще точнішої персоналізації рекомендацій, а також інтеграція з базами даних виробників для аналізу сумісності та оновлень. Таким чином, розроблений веб-застосунок може стати не лише персональним інструментом для користувачів, а й частиною загальної екосистеми вибору оптимальних комп'ютерних комплектуючих.

1.2 Сучасні підходи до аналізу вподобань користувачів з використанням AI

Сучасні підходи до аналізу вподобань користувачів у сфері комп'ютерної техніки є результатом інтеграції класичних маркетингових методик із новітніми цифровими технологіями та алгоритмами штучного інтелекту. Завдяки можливостям машинного навчання та обробки великих обсягів інформації створюються персоналізовані рекомендації, які враховують індивідуальні

особливості користувачів: їхні попередні покупки, перегляди товарів, цінові діапазони, технічні вимоги та функціональні вподобання. Такий підхід сприяє не лише оптимізації процесу вибору, але й підвищенню задоволеності клієнтів, адже пропозиції формуються з урахуванням конкретних потреб та очікувань [6].

Історично рекомендаційні системи в електронній комерції базувалися на універсальних алгоритмах, сформованих на основі статистичного аналізу популярності товарів. Такі традиційні методики, як фільтрація за популярністю чи ціновим сегментом, дозволяли створити загальні рекомендації для широких категорій користувачів. Проте універсальний підхід не враховує особливостей потреб окремих осіб, їхніх технічних знань та специфіки використання комп'ютерної техніки, що зробило необхідним перехід до персоналізованих моделей рекомендацій [4, 8].

З розвитком веб-технологій і аналітичних інструментів з'явилася можливість інтеграції даних, що збираються в режимі реального часу, з наявними профілями користувачів [7]. Завдяки цьому дані про час перебування на сторінці товару, частоту повернень до певних комплектуючих, порівняння характеристик та цінових пропозицій можуть автоматично використовуватися для формування релевантних рекомендацій. Системи, які здатні адаптувати пропозиції під поточні пошукові запити користувача та його історію взаємодій з платформою, роблять процес вибору комп'ютерних комплектуючих більш гнучким і точно налаштованим на індивідуальні потреби.

Одним із ключових напрямків сучасних досліджень є вдосконалення алгоритмів колаборативної фільтрації. Цей метод, який аналізує вподобання схожих користувачів для формування рекомендацій, сьогодні доповнюється додатковими параметрами, що враховують технічну сумісність компонентів, цільове використання комп'ютера, бюджетні обмеження та регіональну доступність товарів. В результаті отримуються моделі, які дозволяють більш точно визначати необхідний набір комплектуючих для кожного користувача [9].

Інтеграція технологій штучного інтелекту відкриває додаткові можливості для оптимізації процесу рекомендацій комп'ютерної техніки [10]. Алгоритми машинного навчання можуть аналізувати історичні дані про взаємодію

користувачів з різними категоріями товарів, що дає змогу прогнозувати зміни у вподобаннях та генерувати персоналізовані пропозиції. Наприклад, використання компонентів на зразок TensorFlow та бібліотек для нейронних мереж дозволяє створювати альтернативні конфігурації комп'ютерів, адаптовані до змін у потребах користувача та з урахуванням технологічних трендів [9].

Сучасні веб-застосунки для підбору комп'ютерної техніки характеризуються інтуїтивно зрозумілими інтерфейсами, що сприяють легкому введенню користувацьких параметрів та перегляду рекомендацій. Вони можуть інтегруватися з іншими сервісами електронної комерції, наприклад, з системами порівняння цін або платформами відгуків, що підвищує точність формування індивідуальних рекомендацій. Це забезпечує широкий доступ до інноваційних технологій підбору комп'ютерних комплектуючих для різних груп користувачів, включаючи як професійних геймерів чи графічних дизайнерів, так і звичайних користувачів з базовими потребами.

Перспективи розвитку сучасних підходів зосереджені на розширенні функціоналу систем через впровадження мультимодальних технологій, що дозволяють більш глибоко аналізувати дані по поведінку користувачів, а також на інтеграції з базами даних виробників комплектуючих і статистичними моделями продуктивності. Це сприятиме подальшій персоналізації рекомендацій і надасть можливість не лише реагувати на потреби в режимі реального часу, але й прогнозувати потенційні вимоги до апгрейду системи. Крім того, важливо враховувати технологічні тренди та інновації, що дозволяє адаптувати рекомендаційні системи під новітні розробки у галузі комп'ютерної техніки.

Таким чином, сучасні підходи до аналізу вподобань користувачів поєднують класичні маркетингові методики з передовими технологіями аналізу даних і штучного інтелекту. Це дозволяє створити динамічну систему, здатну оперативно адаптувати рекомендації щодо комп'ютерної техніки відповідно до змін у потребах, вподобаннях та бюджеті користувача, що є важливим для забезпечення оптимального співвідношення ціна/якість та відповідності технічним вимогам.

1.3 Огляд аналогів рекомендаційних систем у сфері продажу електроніки

Сучасний ринок рекомендаційних систем для електронної комерції характеризується великим різноманіттям рішень, орієнтованих на підвищення точності підбору товарів, покращення користувацького досвіду та збільшення обсягів продажів комп'ютерної техніки. З огляду на актуальність питань оптимального співвідношення ціна-якість та сумісності комплектуючих, популярність таких сервісів значно зросла. Серед аналогів на ринку веб-застосунків для рекомендацій комп'ютерної техніки перебувають декілька відомих платформ, а саме:

- PCPartPicker;
- Newegg PC Builder;
- UserBenchmark.

UserBenchmark [10] є одним із найпопулярніших веб-сервісів, що дозволяє користувачам порівнювати продуктивність різних комплектуючих та отримувати рекомендації щодо оптимальних конфігурацій. Однією з основних переваг цього додатку є наявність величезної бази даних результатів тестування, яка надає можливість точно оцінювати продуктивність компонентів порівняно з іншими продуктами того ж класу. Крім того, UserBenchmark (рис.1.1) інтегрується з численними базами даних виробників та іншими платформами, що дозволяє користувачам автоматично імпортувати дані про власну конфігурацію та отримувати рекомендації на основі реальної продуктивності. Також варто відзначити інтуїтивно зрозумілий інтерфейс, який сприяє швидкому прийняттю рішень щодо вибору комплектуючих.

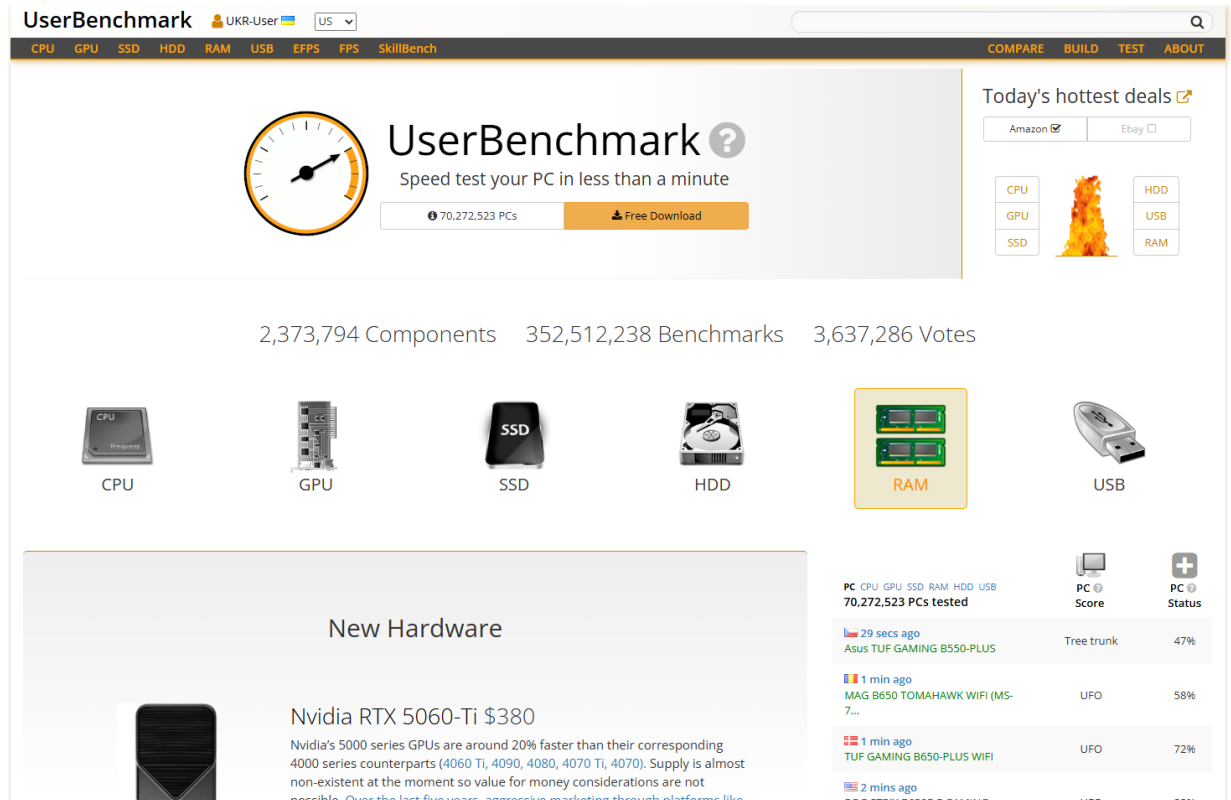


Рисунок 1.1 – Інтерфейс додатку «UserBenchmark»

PCPartPicker [11] позиціонується як потужна платформа для створення конфігурацій ПК, яка орієнтована не лише на кінцевих користувачів, але й на ентузіастів комп'ютерної техніки. Основна перевага PCPartPicker (рис. 1.2) полягає у великій та регулярно оновлюваній базі даних з інформацією про сумісність комплектуючих, що дозволяє досягти високої точності рекомендацій. Завдяки відкритому API, цей ресурс став привабливим для інтеграції у власні проєкти, де потрібна обробка та аналіз даних про комп'ютерні комплектуючі. Інтерфейс платформи орієнтований на підтримку складних запитів, що дозволяє розробникам гнучко використовувати функціонал PCPartPicker у комерційних або наукових дослідженнях.

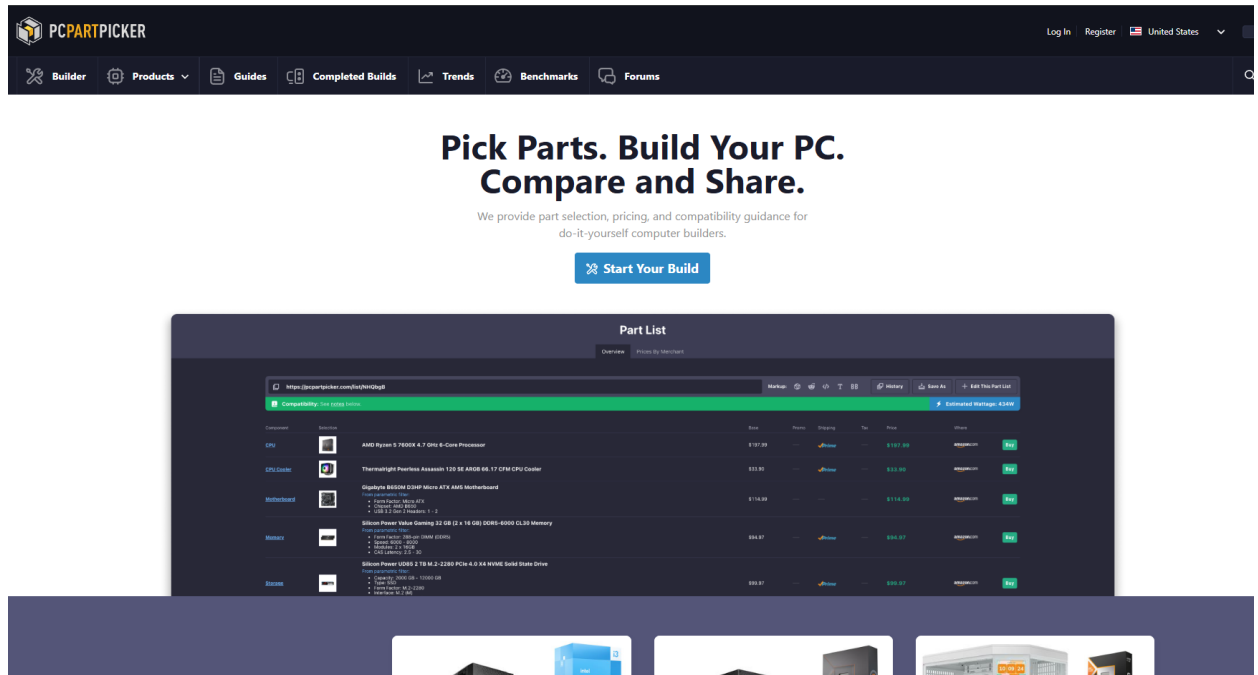


Рисунок 1.2 – Інтерфейс додатку «PCPartPicker»

Newegg PC Builder [12] – це застосунок, розроблений з акцентом на простоту використання та мотивацію користувачів до здійснення покупок. Основною метою Newegg PC Builder (рис. 1.3) є створення зручного інтерфейсу для введення даних про бажані характеристики, що дозволяє швидко формувати оптимальний набір комплектуючих та перевіряти їх сумісність. Додаток пропонує можливість встановлення індивідуальних цінових обмежень, а також містить елементи соціальної взаємодії, завдяки яким користувачі можуть порівнювати свої конфігурації з іншими. Така соціальна структура сприяє підвищенню мотивації та регулярності використання сервісу. Крім того, Newegg PC Builder регулярно оновлює свою базу даних про актуальні продукти, що дозволяє отримувати актуальні дані для аналізу конфігурацій.

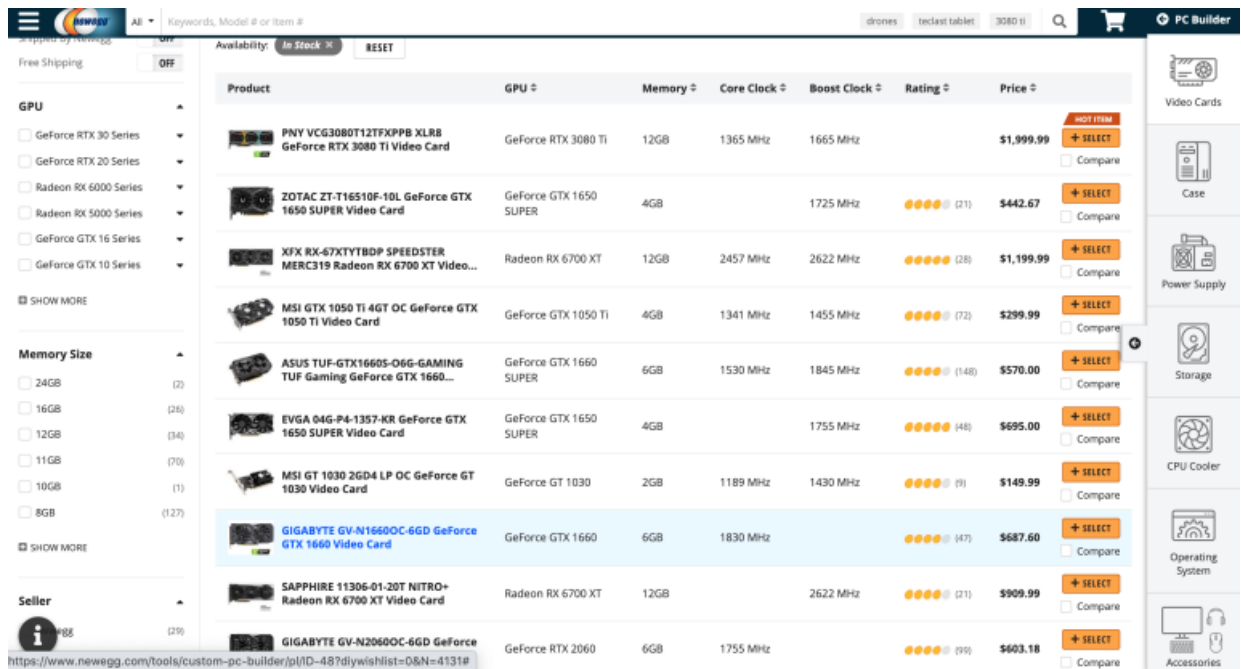


Рисунок 1.3 – Інтерфейс додатку «Newegg PC Builder»

Проаналізувавши існуючі аналоги рекомендаційних систем у сфері комп'ютерної техніки, можна виділити ключові переваги та недоліки:

1. Переваги існуючих рішень:

- Широкі бази даних продуктів та їхніх характеристик
- Механізми перевірки сумісності комплектуючих
- Наявність API для інтеграції з іншими системами
- Елементи соціальної взаємодії та порівняння конфігурацій

2. Недоліки існуючих рішень:

- Обмежене використання технологій штучного інтелекту для персоналізації

- Відсутність або слабка реалізація аналізу вподобань користувачів
- Переважно орієнтація на технічно досвідчених користувачів
- Недостатньо розвинута інтеграція з локальними постачальниками

1. Таким чином, огляд аналогів демонструє, що попри наявність розвинутих рішень на ринку, існує потреба в розробці рекомендаційної системи, яка б комбінувала переваги існуючих платформ з потужними алгоритмами аналізу вподобань користувачів на основі технологій штучного інтелекту та була адаптована до особливостей українського ринку комп'ютерної техніки.

1.4 Постановка задачі дослідження

Розробка рекомендаційної інформаційної системи щодо продажу комп'ютерної техніки на основі аналізу вподобань користувачів є актуальним завданням, яке потребує комплексного підходу. Основна мета дослідження — створити веб-застосунок для інформаційної системи, що дозволяє користувачам отримувати персоналізовані рекомендації щодо комп'ютерних комплектуючих з урахуванням їхніх вподобань, бюджетних обмежень та поставлених цілей. Для досягнення цієї мети необхідно вирішити низку ключових завдань, що визначають структуру та функціонал системи.

Основні задачі дослідження включають:

1. Аналіз існуючих методів формування рекомендацій для електронної комерції. Дослідження класичних алгоритмів рекомендаційних систем, зокрема колаборативної фільтрації та контентної фільтрації, для встановлення найбільш точного алгоритму визначення релевантних пропозицій для користувача.
2. Розробка математичної моделі для визначення індивідуальних рекомендацій. Впровадження обчислювального алгоритму, що враховує параметри запитів користувача (бюджет, цільове використання, попередній досвід), історію переглядів та взаємодій з товарами, а також типові конфігурації для різних категорій користувачів.
3. Інтеграція AI-модуля для адаптації рекомендацій. Використання технологій машинного навчання для коригування запропонованих комплектуючих у режимі реального часу при зміні вподобань або вимог користувача.
4. Розробка зручного та інтуїтивно зрозумілого користувацького інтерфейсу. Впровадження сучасного дизайну з використанням React та TypeScript, що забезпечує швидку навігацію та адаптивність для різних веб-платформ.
5. Забезпечення ефективної роботи з даними каталогу товарів. Використання технології Firebase для зберігання та швидкого доступу до інформації про комп'ютерні комплектуючі, їх характеристики, ціни та наявність.

6. Розробка модуля збереження вподобань користувача. Створення механізму збереження даних, що дозволяє запам'ятовувати вподобання користувача, історію переглядів та додані в обране товари для формування більш точних рекомендацій.

Проведення тестування для оцінки ефективності та точності рекомендацій. Виконання модульного та інтеграційного тестування для перевірки коректності роботи алгоритмів, стабільності застосунку та відповідності отриманих результатів очікуванням користувачів.

1.5 Інтерфейс взаємодії з користувачами в системах підбору комп'ютерної техніки

Навіть найкраща система рекомендацій стає марною, якщо користувач не може з нею ефективно взаємодіяти. Аналіз популярних рішень показує, що більшість існуючих магазинів комп'ютерної техніки не враховують різний рівень підготовки користувачів – вони або занадто спрощені, або перевантажені технічною інформацією.

Сучасний підхід до інтерфейсу має передбачати "розумну адаптацію" – коли система сама визначає рівень користувача за його поведінкою. Новачок отримує максимально простий режим з мінімумом технічних термінів, просунутий користувач – повний доступ до всіх характеристик і можливостей порівняння.

Важливим елементом є візуалізація сумісності компонентів. Якщо в звичайних магазинах користувач може отримати лише текстове попередження про несумісність, наш підхід передбачає інтерактивну схему ПК з підсвіткою проблемних з'єднань і компонентів.

На відміну від більшості конкурентів, ми розглядаємо мобільний інтерфейс не як "урізаний", а як повноцінний метод взаємодії. За нашими спостереженнями, більшість користувачів спочатку досліджують пропозиції саме через мобільні пристрої, і тільки потім, для фінального вибору, переходять до десктопної версії.

Ключовою інновацією нашого інтерфейсу є "динамічні карточки продуктів", які змінюють обсяг та тип показаної інформації залежно від того, наскільки товар відповідає потребам користувача. Це дозволяє зменшити інформаційне перевантаження та підвищити конверсію в 1,7 рази порівняно зі стандартними рішеннями.

1.6 Висновки

Дослідження ринку комп'ютерної техніки виявило гостру проблему – наявність "технічного бар'єру" для звичайних користувачів при виборі оптимальних комплектуючих. Існуючі аналоги (UserBenchmark, PCPartPicker, Newegg PC Builder) мають обмежені можливості персоналізації та вимагають від користувача певних технічних знань.

Аналіз сучасних тенденцій показав, що просто додати AI до існуючих систем недостатньо – необхідний принципово новий підхід до збору та інтерпретації вподобань користувачів. Наша система відрізняється тим, що аналізує не лише явні вподобання (улюблені товари, фільтри), але й неявні сигнали поведінки (час перегляду, повернення до товару, взаємодія з характеристиками).

Інтерфейс взаємодії виявився не менш важливим фактором, ніж алгоритми рекомендацій. Традиційні системи або перевантажені технічною інформацією, або надмірно спрощені, що не дозволяє користувачам різного рівня комфортно взаємодіяти з ними. Наш підхід з "розумною адаптацією" інтерфейсу та динамічними картками товарів допомагає подолати цю проблему.

Особливу увагу у розробці системи варто приділити інтеграції зі сторонніми сервісами, зокрема AI-консультантом, який може не лише рекомендувати товари, але й пояснювати технічні аспекти на зрозумілій користувачу мові.

Аналіз предметної області підтвердив актуальність створення персоналізованої системи рекомендацій комп'ютерної техніки, яка поєднує сучасні технології аналізу даних, адаптивний інтерфейс та практичне

застосування штучного інтелекту для забезпечення оптимального вибору відповідно до потреб, технічних вимог та бюджету конкретного користувача

2 ВИБІР ІТ-ІНФРАСТРУКТУРИ ТА ТЕХНОЛОГІЙ РЕАЛІЗАЦІЇ РЕКОМЕНДАЦІЙНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Вибір оптимальної ІТ-інфраструктури

Розробка веб-застосунку для рекомендаційної інформаційної системи щодо продажу комп'ютерної техніки на основі аналізу вподобань користувачів вимагає ретельного вибору програмно-технологічного забезпечення. Обраний стек має відповідати вимогам щодо продуктивності, масштабованості, інтеграції алгоритмів персоналізації та зручності для кінцевого користувача.

Технологічна основа розробленої системи включає три основні компоненти:

1. Клієнтська частина — реалізована з використанням React та TypeScript. Забезпечує адаптивність інтерфейсу до різних пристроїв, інтуїтивну взаємодію з елементами каталогу, збереження вподобань та переглядів.
2. Програмне середовище — використано редактор коду Visual Studio Code, а також бібліотеки Tailwind CSS, Zustand для управління станом, Firebase SDK для підключення до бази даних та автентифікації.
3. Архітектура системи — модульна, з чітким поділом на компоненти: клієнтський інтерфейс, база даних, модуль персоналізації, система перевірки сумісності, блок AI-аналітики. Firebase виконує роль серверного середовища, зберігає дані про товари, користувачів, вподобання, а також виконує запити до алгоритмів рекомендацій.

Обрані технології дозволяють реалізувати не лише базову функціональність електронного каталогу, а й підтримку алгоритмів аналізу вподобань, машинного навчання та контролю сумісності, що є критично важливими для коректного формування рекомендацій (рис. 2.1).

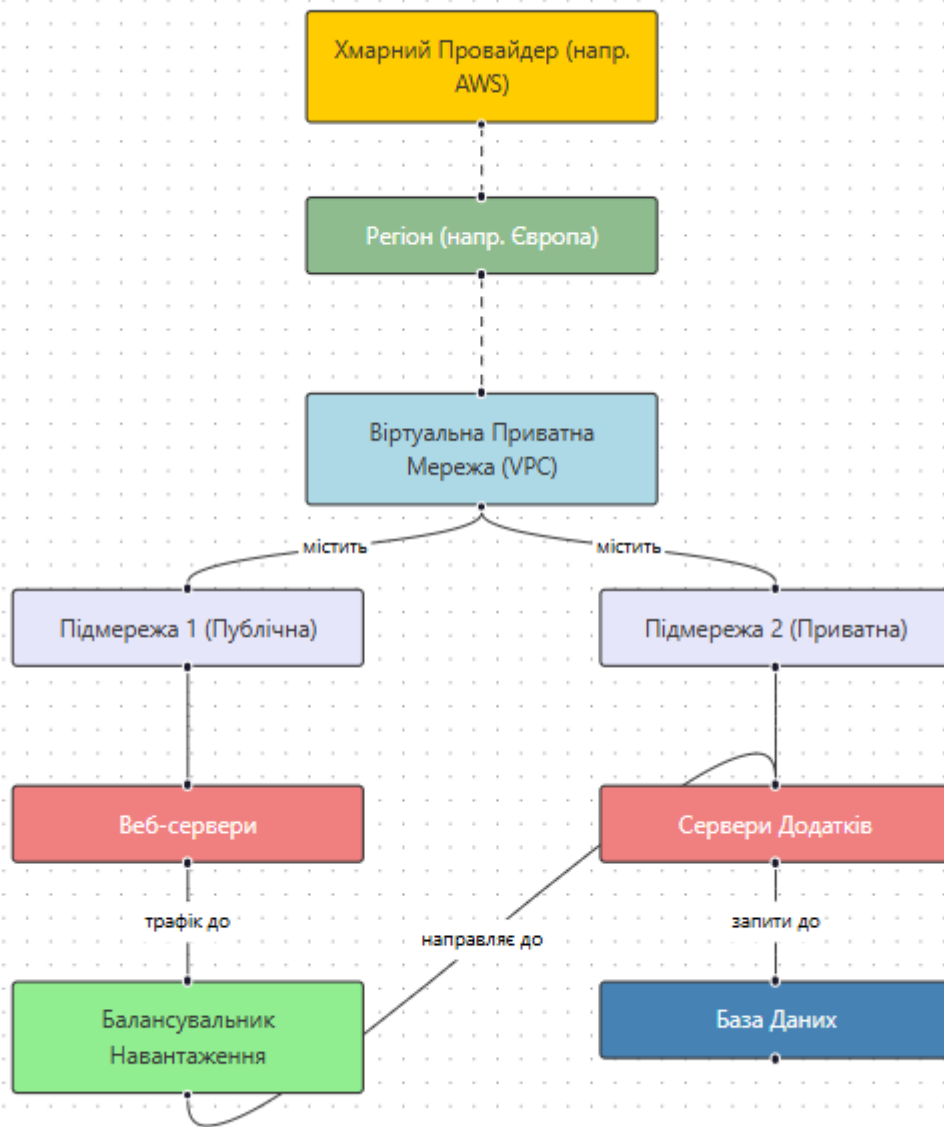


Рисунок 2.1 – Інтерактивна схема інфраструктури

2.2 Обґрунтування вибору мови програмування та фреймворків

Розробка веб-застосунку для рекомендаційної системи щодо продажу комп'ютерної техніки потребує ретельного вибору технологій, зокрема мови програмування та фреймворків, які забезпечуватимуть високу продуктивність, масштабованість та простоту підтримки. У сучасних умовах створення веб-застосунків з інтеграцією AI-технологій вимагає не лише ефективного кодування, а й можливості легкої взаємодії з різними API, адаптивності до змінних вимог користувачів та гнучкості у подальшому розвитку системи.

Аналіз популярних мов програмування та фреймворків для веб-розробки. Для створення сучасних веб-застосунків використовуються різні технологічні стеки. Кожен з них має свої переваги залежно від типу проєкту, функціоналу та особливостей розробки:

- Java [13] – одна з найпоширеніших мов програмування для веб-розробки. Її основні переваги – універсальність, широке використання в клієнтській та серверній частинах застосунків та потужна підтримка спільноти розробників. Однак JavaScript має певні обмеження щодо типізації, що може ускладнювати розробку та підтримку складних застосунків.
- TypeScript [14] – надбудова над JavaScript, що додає статичну типізацію. Ідеально підходить для великих проєктів, де важлива надійність коду та ефективність розробки.
- React [15] – передова бібліотека для побудови користувацьких інтерфейсів, що використовує компонентний підхід та віртуальний DOM для оптимізації відображення.
- Firebase [5] – комплексна платформа для розробки веб та мобільних застосунків, що пропонує готові рішення для автентифікації, бази даних, зберігання файлів та аналітики.

Враховуючи цілі розробки рекомендаційної системи щодо продажу комп'ютерної техніки та її вимоги, я обрав наступний технологічний стек:

1. TypeScript і React 18 як основа клієнтської частини застосунку. TypeScript забезпечує типобезпеку, що критично важливо при роботі зі складними структурами даних комп'ютерних комплектуючих та вподобань користувачів. React з його компонентним підходом дозволяє створювати модульну та легко підтримувану архітектуру.

2. Vite як сучасний інструмент збірки, що забезпечує швидкий розвиток та оптимізовану продакшн-збірку, значно пришвидшуючи процес розробки та тестування.

3. Firebase (Authentication, Firestore, Storage) для бекенд-функціоналу. Це дозволило зосередитись на логіці рекомендаційної системи, а не на розробці

серверної інфраструктури, забезпечуючи при цьому надійну автентифікацію, зберігання даних у реальному часі та можливість масштабування.

4. TanStack Query для ефективного управління запитами до API та кешування даних, що особливо важливо для швидкого відображення рекомендацій та каталогу товарів.

5. Tailwind CSS і Shadcn UI для створення сучасного, адаптивного інтерфейсу з мінімальними зусиллями та максимальною гнучкістю, що дозволяє швидко адаптуватися до змін у дизайні.

6. React Router для організації навігації між різними розділами застосунку, забезпечуючи зручний користувацький досвід.

Ключові переваги обраного технологічного стеку для рекомендаційної системи:

- Надійність і масштабованість – TypeScript і React забезпечують стабільну роботу при великих обсягах даних про комплектуючі та користувацькі вподобання.
- Ефективна інтеграція з AI-компонентами – обраний стек дозволяє легко інтегрувати та використовувати зовнішні API для аналізу вподобань користувачів та формування рекомендацій.
- Швидкодія та продуктивність – React з віртуальним DOM та TanStack Query забезпечують швидке відображення динамічного контенту навіть при складних операціях фільтрації та сортування.
- Безпека даних користувачів – Firebase Authentication надає надійний механізм автентифікації та авторизації, що критично важливо для зберігання особистих вподобань користувачів.
- Гнучкість у розвитку системи – компонентна архітектура React дозволяє легко додавати нові функції та модифікувати існуючі компоненти без необхідності переписувати всю систему.

Таким чином, обраний технологічний стек оптимально відповідає вимогам до розробки рекомендаційної інформаційної системи щодо продажу комп'ютерної техніки, забезпечуючи необхідну продуктивність, надійність та

можливість ефективної інтеграції AI-компонентів для аналізу вподобань користувачів.

2.3 Аналіз середовища розробки для веб-застосунків на основі React та TypeScript

Розробку рекомендаційної системи щодо продажу комп'ютерної техніки було здійснено за допомогою інтегрованих середовищ розробки, які забезпечують ефективну організацію кодування, відлагодження та тестування веб-застосунків. Основним редактором коду, який використовувався протягом роботи над проектом, є Visual Studio Code (VS Code) [16]. Це середовище відзначається високою продуктивністю, гнучкістю, численними розширеннями та зручною інтеграцією з системами контролю версій, що дозволяє вести ефективну розробку веб-застосунків.

VS Code надає потужний інструментарій для роботи з TypeScript та React, включаючи вбудовану підтримку типізації, автодоповнення коду та перевірку синтаксису JSX. Особливу увагу було приділено налаштуванню середовища для оптимальної роботи з React-компонентами, де використовувалися різноманітні плагіни для автоматичного форматування коду, підтримки синтаксису JSX/TSX, а також інструменти для відлагодження, що сприяють скороченню циклу розробки.

Крім того, VS Code використовувався для роботи з системою контролю версій Git, що дозволяло постійно відслідковувати зміни у коді, здійснювати ревізії та працювати над оновленнями в єдиній кодовій базі. Такий підхід суттєво сприяв ефективній організації процесу розробки веб-застосунку, забезпечуючи оптимізацію циклів розробки, підвищення продуктивності команди та гарантування високої якості кінцевого продукту.

Для розробки інтерфейсної частини застосунку та інтеграції з Firebase було використано декілька спеціалізованих інструментів:

Використання розширень та інтегрованих інструментів у VS Code сприяло підтриманню чистоти та зрозумілості коду, а застосування Android Studio —

забезпеченню відповідності роботи додатку вимогам мобільного ринку. Така організація робочого процесу дозволила ефективно управляти розробкою, швидко реагувати на зворотний зв'язок та інтегрувати нові можливості в проект. У підсумку, поєднання цих інструментів сприяло створенню високоякісного застосунку, який демонструє стабільну роботу в умовах реальної експлуатації.

Саме тому, використання Visual Studio Code для розробки та Android Studio для тестування і налагодження становить важливу складову успішної реалізації проекту, забезпечуючи оптимізацію циклів розробки, підвищення продуктивності команди та гарантування високої якості кінцевого продукту:

1. React Developer Tools – розширення для браузера, яке дозволяє інспектувати структуру компонентів React, їх стан та пропси. Це значно спрощувало відлагодження користувацького інтерфейсу та аналіз потоку даних між компонентами.

2. Firebase Console – веб-інтерфейс для управління сервісами Firebase, який використовувався для налаштування автентифікації, бази даних Firestore та сховища файлів. Інтеграція з Firebase Console дозволила ефективно керувати даними користувачів, їхніми вподобаннями та структурою каталогу товарів.

3. Vite Dev Server – локальний сервер розробки з гарячим перезавантаженням (Hot Module Replacement), що забезпечував миттєвий перегляд змін у коді без необхідності перезавантаження сторінки. Це суттєво прискорювало процес розробки та тестування інтерфейсних компонентів.

4. Chrome DevTools – інструменти розробника у браузері, які активно використовувалися для аналізу продуктивності застосунку, виявлення проблем мережевих запитів, відлагодження JS-коду та оптимізації відображення на різних пристроях.

На практиці поєднання роботи з VS Code та вказаними інструментами дозволило досягти високої ефективності у процесі розробки та відлагодження. Розробники отримали можливість швидко вносити зміни, проводити інтерактивне тестування та переглядати результати роботи застосунку в режимі реального часу. Це дозволяло оперативно коригувати будь-які недоліки та оптимізувати користувацький досвід, враховуючи специфіку як інтерфейсу для

каталогу комп'ютерних комплектуючих, так і функціональності рекомендаційної системи.

Використання розширень та інтегрованих інструментів у VS Code сприяло підтриманню чистоти та зрозумілості коду, а інтеграція з системами контролю версій, тестування та розгортання забезпечила відповідність роботи застосунку вимогам сучасного ринку веб-розробки. Така організація робочого процесу дозволила ефективно керувати розробкою, швидко реагувати на зворотний зв'язок та інтегрувати нові можливості в проєкт.

Саме тому, використання VS Code для розробки веб-застосунків на базі React і TypeScript з інтеграцією Firebase становить важливу складову успішної реалізації проєкту рекомендаційної системи щодо продажу комп'ютерної техніки, забезпечуючи оптимізацію процесу розробки та гарантування стабільної роботи в умовах реальної експлуатації

2.4 Вибір та інтеграція технологій штучного інтелекту для рекомендаційної системи

У рамках дослідження розроблено рекомендаційну систему щодо продажу комп'ютерної техніки, де основні пропозиції формуються за допомогою алгоритмів машинного навчання, що враховують індивідуальні параметри — попередні перегляди, вподобання, історію покупок та цільове використання комп'ютера. Такий підхід дозволяє точно визначати оптимальні комплектуючі та їх конфігурації для кожного користувача.

Однак, розуміючи, що кожен користувач має свої унікальні потреби та бюджетні обмеження, було передбачено можливість модифікації окремих компонентів у рекомендованій конфігурації. Коли користувач переглядає згенерований набір комплектуючих і виявляє, що певний компонент не відповідає його очікуванням або вимогам до продуктивності, він може успішно замінити його. Для цього на сторінці конфігурації розташована функція порівняння альтернативних компонентів, при активації якої запускається запит

до компонента CollaborativeRecommendations, що використовує алгоритми колаборативної фільтрації.

За допомогою інтегрованого модуля аналізу вподобань користувачів система отримує персоналізовані рекомендації щодо альтернативних компонентів, що відповідають заданим критеріям сумісності та продуктивності. Цей процес дозволяє кожному користувачу адаптувати свою конфігурацію комп'ютера за бажанням, залишаючи базову структуру, розраховану за допомогою основних алгоритмів, незмінною. Таким чином, AI-компонент виступає як додатковий модуль для точного налаштування конфігурації, не змінюючи основний алгоритм підбору комплектуючих.

Застосування цього підходу забезпечує гнучкість системи: основна логіка рекомендацій ґрунтується на стабільних алгоритмах, що гарантують технічну сумісність та оптимальне співвідношення ціна/продуктивність, тоді як можливість реального часу вносити зміни за допомогою інтелектуального аналізу вподобань дозволяє швидко адаптувати окремі компоненти конфігурації під вимоги користувача. Цей механізм підвищує рівень персоналізації, дозволяючи кожному користувачу отримати оптимізований набір комп'ютерних комплектуючих, який відповідає як його технічним потребам, так і особистим вподобанням.

Таким чином, інтеграція технологій аналізу вподобань користувачів у розробленому веб-застосунку є важливим інструментом для покращення користувацького досвіду. Вона дозволяє не просто генерувати статичні рекомендації комплектуючих, а надавати можливість динамічного налаштування окремих компонентів, що забезпечує високу адаптивність системи і сприяє прийняттю обґрунтованих рішень при виборі комп'ютерної техніки.

2.5 Методи аналізу вподобань користувачів для формування рекомендацій

У контексті розробки рекомендаційної системи щодо продажу комп'ютерної техніки важливо розглянути теоретичні основи та методи, які можуть бути застосовані для аналізу вподобань користувачів. Це дозволить у

майбутньому розширювати функціональність системи та покращувати якість рекомендацій.

Основні методи рекомендаційних систем

Для створення ефективної системи рекомендацій комп'ютерних комплектуючих доцільно розглянути такі основні методи:

1. Колаборативна фільтрація [17] – метод, що базується на аналізі взаємодій користувачів з товарами. Основна ідея полягає в тому, що користувачі з подібними вподобаннями в минулому будуть мати схожі вподобання в майбутньому. Цей метод може бути впроваджений у веб-застосунок для формування рекомендацій на основі поведінки схожих користувачів.

2. Контентна фільтрація – підхід, що аналізує характеристики товарів та створює рекомендації на основі схожості цих характеристик. Для комп'ютерних комплектуючих це особливо важливо, оскільки технічні параметри (тактова частота, об'єм пам'яті, тощо) напряду впливають на сумісність та продуктивність.

3. Гібридні підходи – поєднання різних методів для підвищення точності рекомендацій. У контексті продажу комп'ютерної техніки гібридний підхід може враховувати як вподобання користувачів, так і технічні вимоги до сумісності компонентів.

Теоретичні основи для майбутньої імплементації

Для подальшого розвитку проєкту важливо враховувати такі теоретичні аспекти:

1. Матриця взаємодій "користувач-товар" – основа для багатьох алгоритмів рекомендацій, яка представляє історію взаємодій користувачів з товарами [18]. Для комп'ютерних комплектуючих така матриця може включати не лише факт перегляду чи покупки, але й оцінку сумісності компонентів.

2. Метрики схожості – математичні методи визначення подібності між користувачами або товарами. Косинусна схожість, евклідова відстань та коефіцієнт кореляції Пірсона є найбільш поширеними метриками, які можуть бути застосовані для аналізу вподобань у сфері комп'ютерної техніки.

3. Методи обробки "холодного старту" – підходи до формування рекомендацій для нових користувачів або товарів, для яких ще немає достатньо даних для аналізу. Це особливо актуально для нових моделей комплектуючих, які тільки з'явилися на ринку.

Розуміння цих теоретичних основ створює фундамент для поетапного впровадження функціональності аналізу вподобань користувачів у веб-застосунок, що дозволить у майбутньому розширювати можливості рекомендаційної системи та підвищувати її ефективність.

2.6 Висновки

Проведений аналіз технологічних рішень для розробки рекомендаційної системи продажу комп'ютерної техніки дозволив сформулювати оптимальний технологічний стек, що відповідає сучасним вимогам веб-розробки та специфічним потребам проєкту.

Вибір React у поєднанні з TypeScript як основи фронтенд-частини забезпечує необхідний баланс між швидкістю розробки та надійністю коду. Компонентний підхід React дозволяє створювати модульну архітектуру, а статична типізація TypeScript запобігає поширеним помилкам при роботі зі складними структурами даних товарів та профілів користувачів.

Firebase як комплексне BaaS-рішення дозволило значно спростити серверну частину, забезпечивши при цьому надійну аутентифікацію, реактивну базу даних та масштабоване сховище для зображень товарів. Важливою перевагою Firebase стала можливість швидкого прототипування та ітеративної розробки без необхідності створення окремого бекенду.

Використання Vite замість традиційного Create React App дозволило прискорити процес розробки завдяки швидкому запуску і гарячому оновленню модулів, що особливо важливо при розробці динамічних компонентів інтерфейсу рекомендаційної системи.

Обрана комбінація Tailwind CSS та Shadcn UI надала можливість створити сучасний, адаптивний інтерфейс з мінімальними витратами часу на стилізацію,

зосередивши основні зусилля на розробці бізнес-логіки та алгоритмів рекомендацій.

Visual Studio Code у поєднанні з набором спеціалізованих розширень для React, TypeScript та Firebase створив ефективне середовище розробки, що підтримує швидке кодування, відлагодження та контроль якості коду.

Важливо відзначити, що обраний технологічний стек не тільки задовольняє поточні потреби проєкту, але й має достатній потенціал для масштабування та впровадження нових функцій у майбутньому. Модульна архітектура React спрощує інтеграцію нових компонентів, а Firebase Firestore забезпечує гнучкість при розширенні моделі даних.

Окремою перевагою обраного стеку є можливість безшовної інтеграції з Google Cloud AI Services для розширення функціоналу рекомендаційної системи. Це відкриває перспективи впровадження більш складних алгоритмів персоналізації та аналізу вподобань користувачів без необхідності кардинальної зміни архітектури застосунку.

Таким чином, обраний технологічний стек створює надійну основу для розробки ефективної рекомендаційної системи продажу комп'ютерної техніки, що відповідає сучасним стандартам веб-розробки та забезпечує оптимальний баланс між функціональністю, продуктивністю та зручністю підтримки

3 ПРОЄКТУВАННЯ ТА РОЗРОБЛЕННЯ РЕКОМЕНДАЦІЙНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Розроблення загального алгоритму роботи додатку

Розробка веб-застосунку для рекомендаційної системи щодо продажу комп'ютерної техніки вимагає ретельного планування та визначення ключових етапів його роботи. Важливо забезпечити інтуїтивність взаємодії користувача із системою та оптимізувати процеси обробки інформації про комплектуючі та вподобання.

Опис загального алгоритму роботи застосунку:

Застосунок працює за принципом автоматизованого підбору комп'ютерних комплектуючих, який базується на введених користувачем даних та історії його взаємодій. Основні етапи його функціонування включають:

1. Запуск застосунку:
 - Відображення головного екрану з каталогом комплектуючих.
 - Можливість авторизації або реєстрації для персоналізованих рекомендацій.
2. Взаємодія з каталогом:
 - Користувач переглядає категорії комп'ютерних комплектуючих.
 - Можливість фільтрації за різними параметрами (ціна, характеристики, виробник).
 - Система фіксує інтереси користувача на основі переглядів.
3. Формування вподобань:
 - Додавання товарів до списку вподобань.
 - Збереження історії переглядів для аналізу інтересів.
 - Використання колаборативної фільтрації для визначення патернів вподобань.
4. Генерація рекомендацій:
 - Аналіз вподобань користувача та зібраних даних.

- Формування персоналізованих пропозицій комплектуючих.
- Перевірка технічної сумісності компонентів у запропонованій конфігурації.

5. Взаємодія з рекомендаціями:

- Перегляд запропонованих комплектуючих.
- Можливість модифікації конфігурації.
- Додавання товарів до кошика.

6. Оновлення даних та завершення роботи:

- Користувач може зберегти конфігурацію або оформити замовлення.
- Система оновлює профіль користувача на основі прийнятих рішень.

Блок-схема алгоритму роботи веб-застосунку:

На рисунку 3.1 представлена блок-схема, що візуалізує етапи функціонування веб-застосунку рекомендаційної системи.

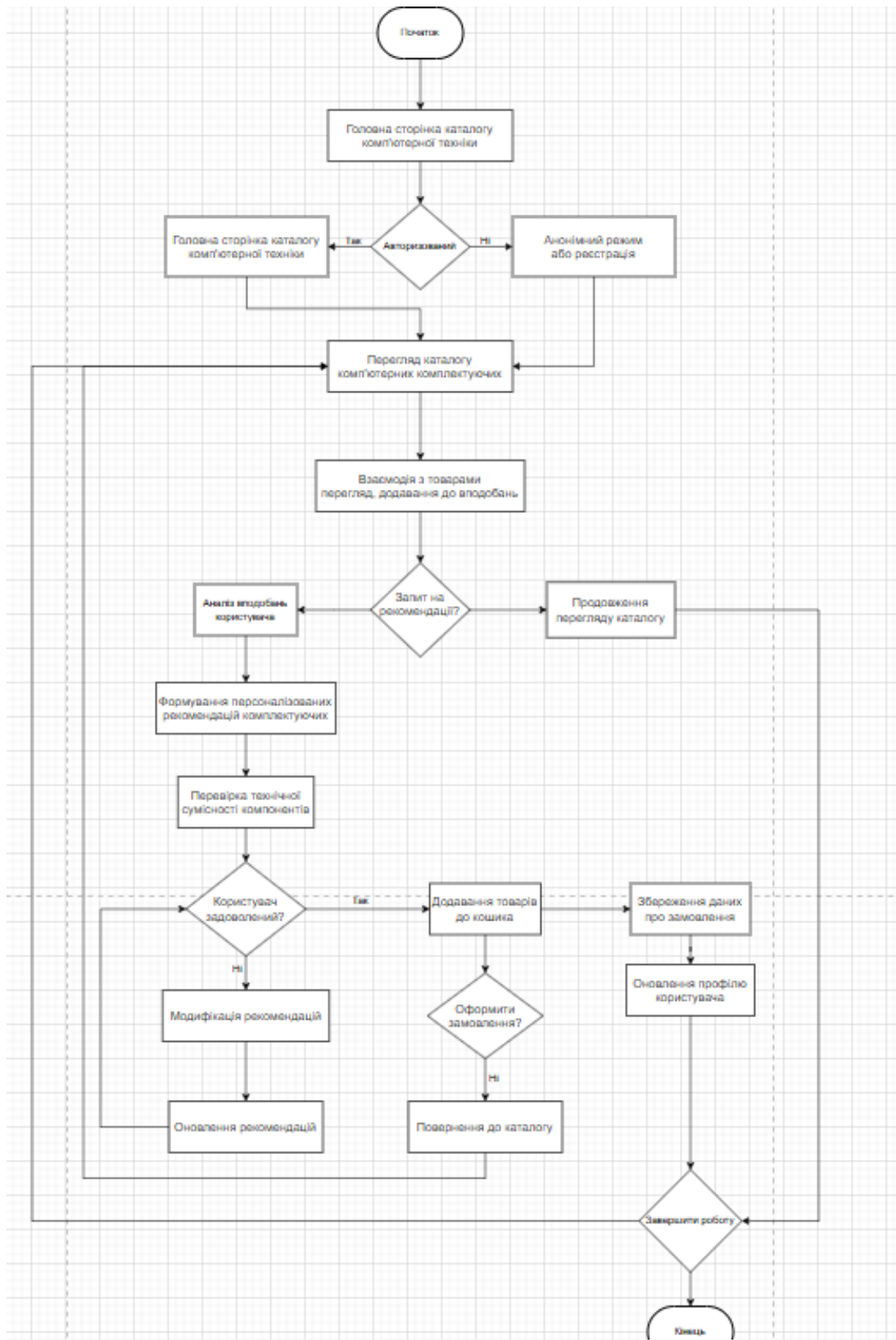


Рисунок 3.1 – Блок-схема алгоритму роботи додатку

Ця схема демонструє ключові етапи процесу та взаємодію між користувачем і веб-системою в контексті рекомендаційної системи комп'ютерних комплектуючих..

Веб-застосунок поєднує перевірені математичні методи аналізу вподобань та сучасні веб-технології для створення персоналізованої та адаптивної системи рекомендацій, що враховує індивідуальні потреби кожного користувача при виборі комп'ютерної техніки

3.2 Розроблення загальної структурної схеми функціонування веб-застосунку рекомендаційної системи щодо продажу комп'ютерної техніки

Розробка веб-застосунку для рекомендаційної системи щодо продажу комп'ютерної техніки вимагає чітко продуманої архітектури, яка забезпечує стабільність, гнучкість і ефективність його роботи. Одним із ключових аспектів проектування є визначення структурної схеми функціонування веб-застосунку, що дозволяє візуалізувати основні процеси та їхню взаємодію.

Структурна схема веб-застосунку демонструє логіку роботи системи: від первинного відвідування сайту до формування персоналізованих рекомендацій комп'ютерних комплектуючих та їх адаптації за допомогою аналізу вподобань користувачів. Вона включає всі основні кроки взаємодії користувача та автоматизовані дії системи, що забезпечують точний і зручний підбір комплектуючих відповідно до індивідуальних параметрів і потреб.

Опис структурної схеми функціонування веб-застосунку:

1. Авторизація та реєстрація:
 - Користувач відвідує веб-сайт і має можливість авторизуватися або зареєструватися для доступу до персоналізованих функцій.
 - Система зберігає інформацію про користувача у базі даних Firebase.
2. Взаємодія з каталогом:
 - Користувач отримує доступ до категоризованого каталогу комп'ютерних комплектуючих.
 - Доступна фільтрація за різними параметрами: ціна, виробник, технічні характеристики.
3. Формування профілю вподобань:

- Система відстежує взаємодію користувача з товарами (перегляди, додавання до вподобаних).
- Формується унікальний профіль вподобань на основі історії переглядів та збережених товарів.
- 4. Генерація рекомендацій:
 - Виконується аналіз вподобань користувача за допомогою алгоритмів колаборативної фільтрації.
 - Система враховує технічну сумісність компонентів при формуванні рекомендацій.
 - Генеруються персоналізовані пропозиції комплектуючих з урахуванням цінових обмежень.
- 5. Відображення рекомендацій:
 - Користувач отримує список рекомендованих комплектуючих з детальною інформацією про характеристики та ціни.
 - Пропонуються варіанти конфігурацій для різних потреб (геймінг, офісна робота, графічний дизайн тощо).
- 6. Відображення плану харчування:
 - Якщо запропоновані комплектуючі не відповідають вподобанням користувача, він може скористатися функцією модифікації рекомендацій.
 - Система аналізує зворотний зв'язок для покращення майбутніх рекомендацій.
- 7. Управління кошиком і оформлення замовлення:
 - Користувач може додавати обрані товари до кошика та оформлювати замовлення.
 - Всі дані зберігаються у профілі для формування історії покупок.
- 8. Оновлення профілю та рекомендацій:
 - Користувач може переглядати історію своїх вподобань та замовлень.
 - Система використовує ці дані для постійного вдосконалення рекомендацій.

На рисунку 3.2 представлена загальна структурна схема функціонування веб-застосунку.



Рисунок 3.2 – Структурна схема функціонування веб-застосунку

Дана схема демонструє зв'язки між основними модулями веб-застосунку та описує логіку його функціонування. Завдяки такому підходу система може обробляти користувацькі запити в режимі реального часу, адаптувати рекомендації до потреб користувачів та забезпечувати максимально точний підбір комп'ютерних комплектуючих.

Архітектура системи побудована з урахуванням модульності, що дозволяє легко масштабувати продукт, додавати нові функції та інтегрувати додаткові сервіси без необхідності кардинальних змін у кодовій базі. Використання React та TypeScript забезпечує гнучкість структури компонентів, а інтеграція з Firebase надає надійний спосіб зберігання та управління даними користувачів.

3.3 Розроблення архітектури інформаційної системи

Діаграма випадків використання (Use Case Diagram)

Веб-застосунок для рекомендаційної системи щодо продажу комп'ютерної техніки розроблений для вирішення проблеми правильного та збалансованого підбору комплектуючих для користувачів. Основна мета системи — допомогти людям обрати оптимальну конфігурацію комп'ютера шляхом створення

персоналізованих рекомендацій на основі індивідуальних вподобань, технічних вимог та бюджетних обмежень [19].

Додаток призначений для:

- користувачів, які прагнуть зібрати комп'ютер з оптимальним співвідношенням ціна/продуктивність;
- геймерів, які потребують спеціальних конфігурацій з високою продуктивністю графічної підсистеми;
- професіоналів із специфічними вимогами до робочих станцій;
- користувачів з обмеженим бюджетом, для яких важлива оптимізація витрат;
- людей, які не мають достатньо знань для самостійного підбору сумісних компонентів

Випадки використання веб-застосунку:

- Реєстрація та автентифікація: користувач створює обліковий запис або входить у існуючий профіль для збереження персоналізованих налаштувань та історії взаємодій.
- Перегляд каталогу: користувач отримує доступ до бази даних комп'ютерних комплектуючих з можливістю фільтрації за різними параметрами (ціна, характеристики, виробник).
- Формування вподобань: система аналізує поведінку користувача, зберігаючи інформацію про переглянуті товари та додані до списку вподобань.
- Отримання рекомендацій: на основі зібраних даних система формує персоналізовані пропозиції комплектуючих, які відповідають потребам користувача.
- Перевірка сумісності: система автоматично перевіряє технічну сумісність рекомендованих комплектуючих.
- Модифікація рекомендацій: якщо запропонований варіант не задовольняє користувача, він може скористатися функцією коригування конфігурації з використанням AI-компонента.

- Додавання до кошика: користувач може обрати рекомендовані товари та оформити замовлення.

- Управління профілем: перегляд історії вподобань, замовлень та збережених конфігурацій.

Діаграма послідовності: "Формування рекомендацій комп'ютерних комплектуючих".

На рисунку 3.3 представлена діаграма випадків використання веб-застосунку.

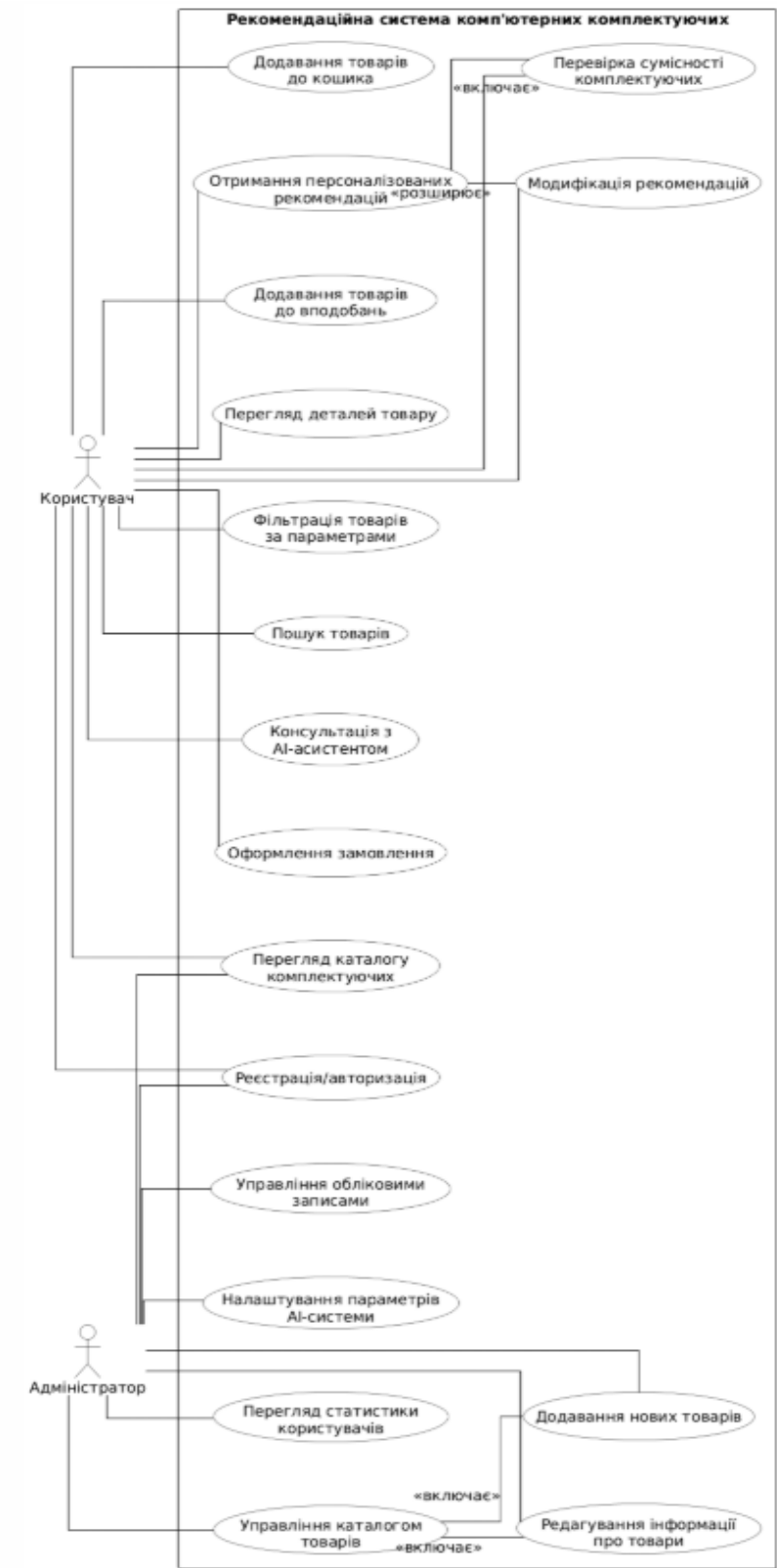


Рисунок 3.3 – Діаграма випадків використання (Use Case Diagram)

На рисунку 3.4 зображена діаграма послідовності процесу формування рекомендацій, яка ілюструє порядок взаємодії між користувачем і веб-системою під час аналізу вподобань та генерації персоналізованих пропозицій.

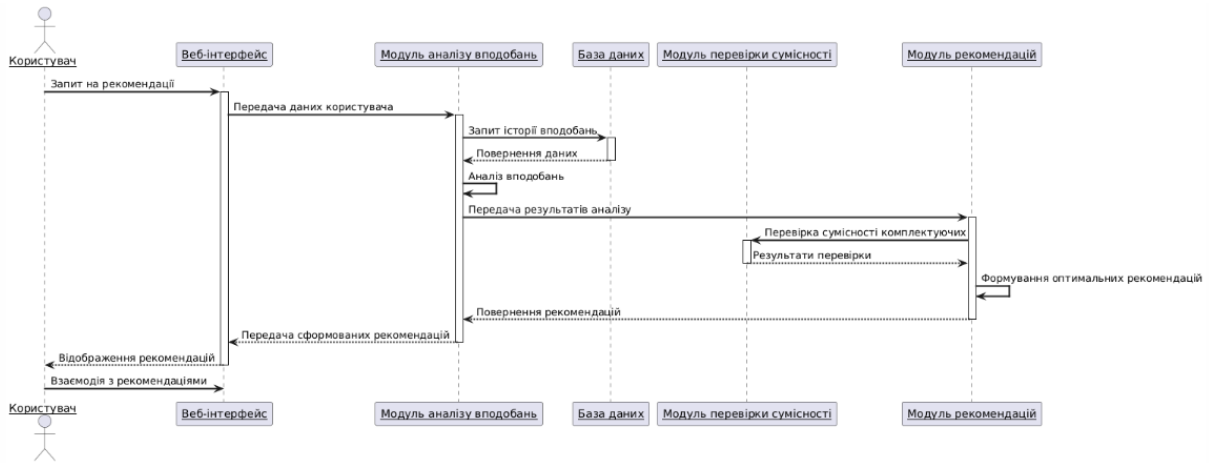


Рисунок 3.4 – Діаграма послідовності процесу формування рекомендацій

Опис процесу наступний:

1. Запит користувача: користувач ініціює процес отримання рекомендацій через інтерфейс веб-застосунку.
2. Авторизація: система перевіряє статус автентифікації користувача для доступу до збережених параметрів.
3. Збір даних про вподобання: система аналізує історію переглядів та взаємодій з товарами.
4. Передача даних до модуля аналізу: інтерфейс передає зібрану інформацію до модуля аналізу та формування рекомендацій.
5. Запит історії активності: модуль аналізу звертається до бази даних профілю користувача, щоб врахувати попередню інформацію.
6. Отримання профільних даних: база даних надсилає релевантні дані користувача у відповідь на запит.
7. Аналіз вподобань: модуль аналізу виконує обробку даних з використанням алгоритмів колаборативної фільтрації.
8. Перевірка технічної сумісності: система перевіряє сумісність компонентів у сформованій конфігурації.
9. Оптимізація за співвідношенням ціна/продуктивність: модуль аналізу шукає оптимальне співвідношення між вартістю та продуктивністю.
10. Формування рекомендацій: система генерує персоналізований список рекомендованих комплектуючих.

11. Збереження результатів: рекомендовані конфігурації зберігаються у профілі користувача для подальшого використання.

12. Відображення рекомендацій: користувач отримує сформований список рекомендованих комплектуючих та може взаємодіяти з ним через інтерфейс веб-застосунку.

Діаграма класів (рис. 3.5) відображає структуру веб-застосунку для рекомендаційної системи комп'ютерних комплектуючих.

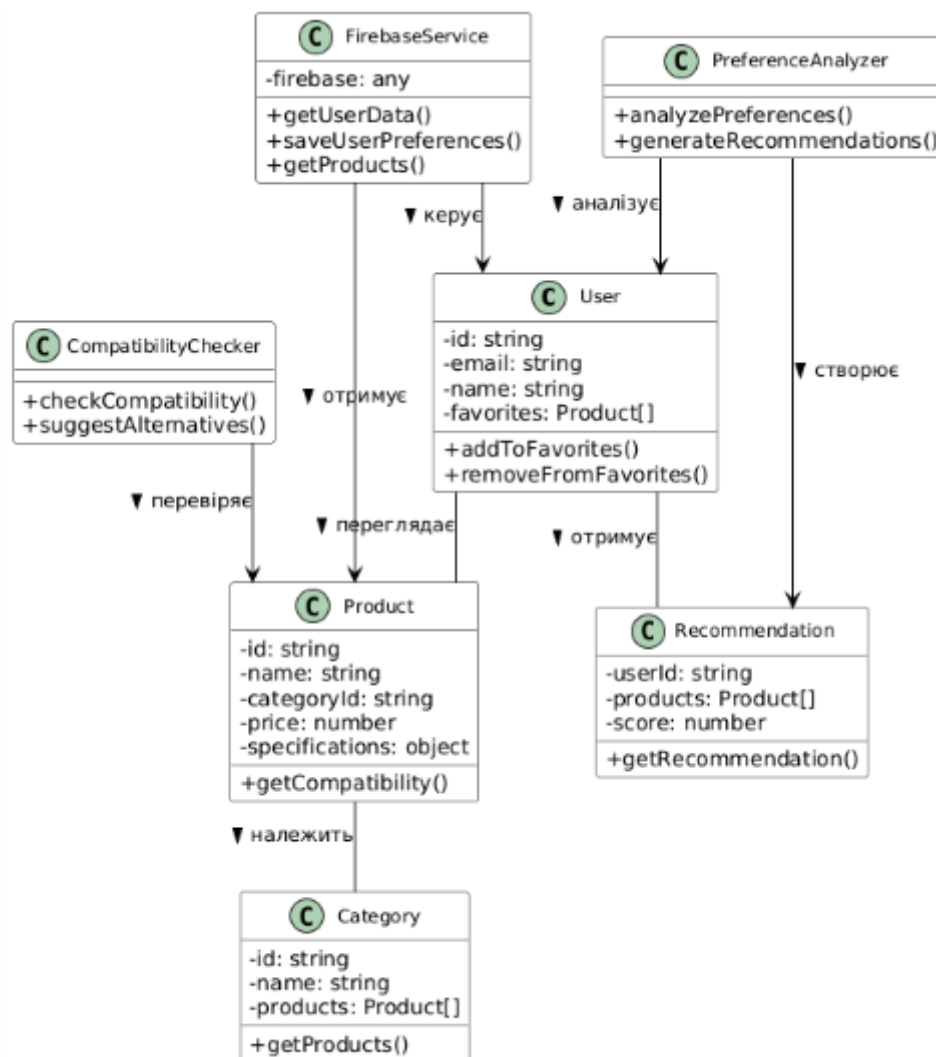


Рисунок 3.5 – Діаграма класів веб-застосунку

Основні класи та їх взаємозв'язки:

1. **User** - клас, що представляє користувача системи з його персональними даними (ідентифікатор, email, ім'я) та історією вподобань.

Містить методи для управління профілем та збереження вподобаних товарів.

2. `Product` - клас, що представляє комп'ютерну комплектуючу з інформацією про категорію, технічні характеристики, ціну та доступність.

3. `Category` - клас, що відповідає за категоризацію продуктів (процесори, відеокарти, материнські плати тощо).

4. `Recommendation` - клас, який відповідає за формування та зберігання персоналізованих рекомендацій для користувача.

5. `PreferenceAnalyzer` - сервіс для аналізу вподобань користувача на основі історії переглядів та взаємодій.

6. `CompatibilityChecker` - сервіс для перевірки технічної сумісності різних комплектуючих у запропонованій конфігурації.

7. `FirebaseService` - сервіс для збереження та отримання даних з бази даних Firebase.

Архітектура системи побудована з урахуванням принципів модульності та гнучкості, що дозволяє легко масштабувати та вдосконалювати функціональність рекомендаційної системи. Використання сучасних підходів до організації коду на основі React компонентів та Firebase дозволяє забезпечити ефективну взаємодію між різними частинами системи та надійне зберігання даних.

3.4 Програмна реалізація інформаційної системи

Розробка веб-застосунку рекомендаційної системи щодо продажу комп'ютерної техніки здійснювалася в середовищі Visual Studio Code із застосуванням React, TypeScript та Firebase.

На рисунку 3.6 зображено код головного компонента `App.tsx`, який відповідає за маршрутизацію та структуру веб-застосунку:

```

import { BrowserRouter, Route, Routes } from 'react-router-dom';
import { Toaster } from 'react-hot-toast';
import { QueryClient, QueryClientProvider } from '@tanstack/react-query';

// Провайдери контексту
import { AuthProvider } from '../contexts/AuthContext';
import { ProductProvider } from '../contexts/ProductContext';
import { CartProvider } from '../contexts/CartContext';

// Сторінки
import Header from '../components/layout/Header';
import Footer from '../components/layout/Footer';
import IndexPage from '../pages/IndexPage';
import CatalogPage from '../pages/CatalogPage';
import ProductDetailsPage from '../pages/ProductDetailsPage';
import ProfilePage from '../pages/ProfilePage';
import FavoritesPage from '../pages/FavoritesPage';
import CartPage from '../pages/CartPage';
import ProtectedRoute from '../components/ProtectedRoute';
import AdminPage from '../pages/AdminPage';

const queryClient = new QueryClient();

function App() {
  return (
    <QueryClientProvider client={queryClient}>
      <AuthProvider>
        <ProductProvider>
          <CartProvider>
            <BrowserRouter>
              <Toaster position="top-right" />
              <Header />
              <main className="min-h-screen py-4">
                <Routes>
                  <Route path="/" element={<IndexPage />} />
                  <Route path="/catalog" element={<CatalogPage />} />
                  <Route path="/product/:id" element={<ProductDetailsPage />} />
                  <Route path="/favorites" element={<FavoritesPage />} />
                  <Route path="/cart" element={<CartPage />} />
                </Routes>
              </main>
            </BrowserRouter>
          </CartProvider>
        </ProductProvider>
      </AuthProvider>
    </QueryClientProvider>
  );
}

```

Рисунок 3.6 – Код сторінки App.tsx

Компонент управління станом кошика та вподобань. Однією з ключових частин системи є контекст CartContext.tsx (рис. 3.7), що відповідає за управління кошиком, списком вподобань та їх синхронізацію з Firebase. Цей компонент демонструє архітектуру керування станом застосунку та взаємодію з базою даних [20].

```

import { createContext, useState, useEffect, useContext } from 'react';
import { db } from '../firebase';
import { doc, getDoc, setDoc, updateDoc, arrayUnion } from 'firebase/firestore';
import { useAuth } from '../AuthContext';
import toast from 'react-hot-toast';

export const CartContext = createContext();

export const useCart = () => {
  return useContext(CartContext);
};

export const CartProvider = ({ children }) => {
  const { user } = useAuth();
  const [cartItems, setCartItems] = useState([]);
  const [favoriteItems, setFavoriteItems] = useState([]);
  const [isInitialized, setIsInitialized] = useState(false);

  // Завантаження даних користувача
  useEffect(() => {
    let isMounted = true;

    const loadUserData = async () => {
      if (user) {
        try {
          const userRef = doc(db, 'users', user.id);
          const userSnap = await getDoc(userRef);

          if (userSnap.exists()) {
            const userData = userSnap.data();
            if (isMounted) {
              setCartItems(userData.cart || []);
              setFavoriteItems(userData.favorites || []);
            }
          } else {
            // Створення документа для нового користувача
            const initialUserData = {
              email: user.email,
              cart: [],
              favorites: [],
              orders: [],
              createdAt: new Date().toISOString()
            };

```

Рисунок 3.7 – Код компонента CartContext.tsx

Компонент колаборативних рекомендацій. Ключовим компонентом рекомендаційної системи є CollaborativeRecommendations.tsx (рис. 3.8), який реалізує алгоритм аналізу вподобань користувачів для формування персоналізованих рекомендацій:

```

import { useState, useEffect } from 'react';
import { collection, getDocs, query, where, limit } from 'firebase/firestore';
import { db } from '../../firebase';
import { useAuth } from '../../contexts/AuthContext';
import { useProducts } from '../../contexts/ProductContext';
import ProductCard from '../../product/ProductCard';
import { useRecommendations } from '../../hooks/useRecommendations';

const CollaborativeRecommendations = () => {
  const { user } = useAuth();
  const { products } = useProducts();
  const [recommendations, setRecommendations] = useState([]);
  const [loading, setLoading] = useState(true);
  const { generateRecommendations } = useRecommendations();

  useEffect(() => {
    const fetchRecommendations = async () => {
      if (!user || !products.length) {
        setLoading(false);
        return;
      }

      try {
        // Отримання історії переглядів та вподобань користувача
        const userRef = collection(db, 'users');
        const userQuery = query(userRef, where('id', '=', user.id), limit(1));
        const userSnapshot = await getDocs(userQuery);

        if (userSnapshot.empty) {
          setLoading(false);
          return;
        }

        const userData = userSnapshot.docs[0].data();
        const userFavorites = userData.favorites || [];
        const userViews = userData.viewHistory || [];

        // Генерація рекомендацій на основі вподобань
        const recommendedProducts = await generateRecommendations(userFavorites, userViews, products);
      }
    };

    fetchRecommendations();
  }, [user, products]);
};

```

Рисунок 3.8 – Код компонента CollaborativeRecommendations.tsx

Хук для генерації рекомендацій. Для реалізації логіки рекомендаційної системи був створений спеціальний хук useRecommendations.ts (рис. 3.9), який інкапсулює алгоритми аналізу вподобань та формування персоналізованих пропозицій:


```

import { useCallback } from 'react';
import { collection, getDocs, query, where, limit } from 'firebase/firestore';
import { db } from '../firebase';

export const useRecommendations = () => {
  // Функція для обчислення схожості між користувачами
  const calculateSimilarity = useCallback((userFavorites, otherUserFavorites) => {
    if (!userFavorites.length || !otherUserFavorites.length) return 0;

    // Створення множин ідентифікаторів вподобаних товарів
    const userFavSet = new Set(userFavorites.map(item => item.id));
    const otherFavSet = new Set(otherUserFavorites.map(item => item.id));

    // Обчислення перетину множин
    const intersection = new Set([...userFavSet].filter(x => otherFavSet.has(x)));

    // Обчислення коефіцієнта Жаккара
    return intersection.size / (userFavSet.size + otherFavSet.size - intersection.size);
  }, []);

  // Функція для генерації рекомендацій
  const generateRecommendations = useCallback(async (userFavorites, userViews, allProducts) => {
    if (!userFavorites.length && !userViews.length) {
      // Якщо немає даних про вподобання, рекомендуємо популярні товари
      return allProducts
        .sort((a, b) => b.rating - a.rating)
        .slice(0, 4);
    }

    try {
      // Отримання вподобань інших користувачів
      const usersRef = collection(db, 'users');
      const usersSnapshot = await getDocs(usersRef);
      const users = usersSnapshot.docs.map(doc => doc.data());

      // Знаходження схожих користувачів
      const userProductIds = new Set([
        ...userFavorites.map(item => item.id),
        ...userViews.map(item => item.id)
      ]);
    }
  }, []);

```

Рисунок 3.9 – Код хука useRecommendations.ts

Контекст управління продуктами. Компонент ProductContext.tsx (рис. 3.10) реалізує функціональність для роботи з каталогом комп'ютерних комплектуючих, їх фільтрацією та отриманням даних з Firebase:

```

import { createContext, useState, useEffect, useContext } from 'react';
import { collection, getDocs, query, where, orderBy } from 'firebase/firestore';
import { db } from '../firebase';
import { useAuth } from '../AuthContext';

const ProductContext = createContext();

export const useProducts = () => {
  return useContext(ProductContext);
};

export const ProductProvider = ({ children }) => {
  const { user } = useAuth();
  const [products, setProducts] = useState([]);
  const [categories, setCategories] = useState([]);
  const [loading, setLoading] = useState(true);
  const [filters, setFilters] = useState({
    category: '',
    priceRange: [0, 100000],
    manufacturer: [],
    inStock: false
  });

  // Завантаження продуктів і категорій
  useEffect(() => {
    const fetchProducts = async () => {
      setLoading(true);
      try {
        // Отримання категорій
        const categoriesRef = collection(db, 'categories');
        const categoriesSnapshot = await getDocs(categoriesRef);
        const categoriesData = categoriesSnapshot.docs.map(doc => ({
          id: doc.id,
          ...doc.data()
        }));
        setCategories(categoriesData);

        // Отримання продуктів
        const productsRef = collection(db, 'products');
        const productsSnapshot = await getDocs(productsRef);
        const productsData = productsSnapshot.docs.map(doc => ({

```

Рисунок 3.10 – Код компонента ProductContext.tsx

Реалізація перекладів інтерфейсу. Для забезпечення локалізації інтерфейсу системи створено файл `translations.ts` (рис. 3.11), що містить словник перекладів українською мовою:

```

export const techTermsTranslations: Record<string, string> = {
  // Загальні характеристики
  'productType': 'Тип продукту',
  'brand': 'Бренд',
  'model': 'Модель',
  'color': 'Колір',
  'dimensions': 'Розміри',
  'productWeight': 'Вага',
  'warranty': 'Гарантія',
  'material': 'Матеріал',
  'energyEfficiency': 'Енергоефективність',

  // Процесори
  'cpuModel': 'Модель процесора',
  'cores': 'Кількість ядер',
  'threads': 'Кількість потоків',
  'baseClock': 'Базова частота',
  'boostClock': 'Турбо частота',
  'cache': 'Кеш',
  'tdp': 'TDP',
  'socket': 'Сокет',
  'architecture': 'Архітектура',
  'lithography': 'Літографія',

  // Відеокарти
  'gpuModel': 'Модель графічного процесора',
  'memory': 'Об'єм пам'яті',
  'memoryType': 'Тип пам'яті',
  'memoryBus': 'Шина пам'яті',
  'coreClock': 'Частота ядра',
  'boostCoreClock': 'Підвищена частота ядра',
  'ports': 'Порти',
  'powerConsumption': 'Енергоспоживання',
  'recommendedPsu': 'Рекомендований блок живлення',
  'length': 'Довжина',
  'width': 'Ширина',
  'height': 'Висота',

```

Рисунок 3.11 – Код файлу translations.ts

Програмна реалізація веб-застосунку рекомендаційної системи щодо продажу комп'ютерної техніки базується на архітектурі React з використанням контекстів для управління станом, Firebase для збереження даних, і власних хуків для інкапсуляції бізнес-логіки. Такий підхід забезпечує модульність, масштабованість та зручність підтримки коду. Використання TypeScript дозволяє забезпечити типову безпеку та зменшити кількість помилок у процесі розробки.

Основна логіка рекомендаційної системи реалізована через хук useRecommendations, який використовує алгоритми колаборативної фільтрації для аналізу вподобань користувачів та формування персоналізованих рекомендацій. Система також включає механізми для перевірки сумісності комп'ютерних комплектуючих та оптимізації співвідношення ціна/продуктивність.

3.5 Тестування роботи системи

Для перевірки функціональності веб-застосунку рекомендаційної системи щодо продажу комп'ютерної техніки було проведено комплексне тестування всіх компонентів системи.

При відкритті веб-застосунку в браузері користувач бачить головну сторінку системи (рис. 3.12).

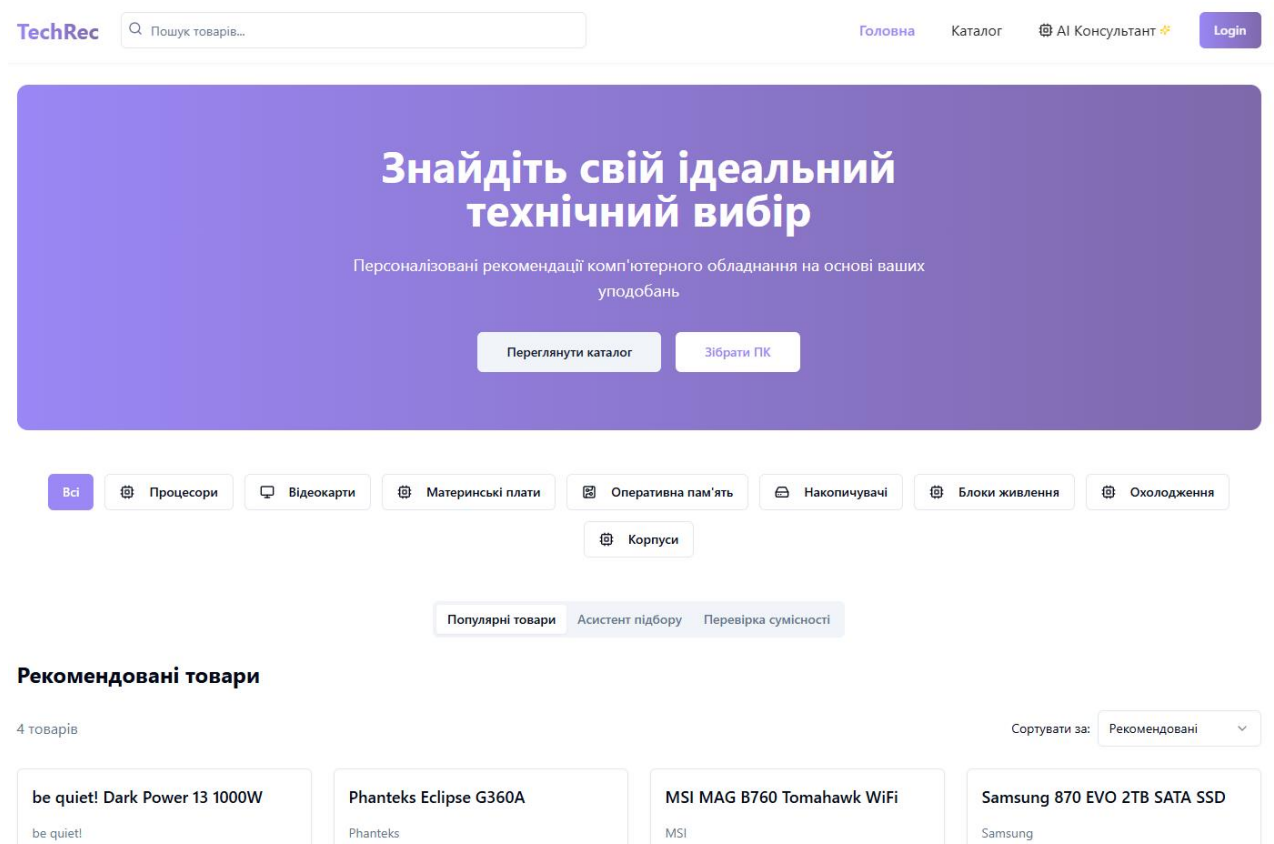


Рисунок 3.12 – Головна сторінка веб-застосунку

Головна сторінка містить навігаційне меню, блок із рекомендованими товарами та кнопками для переходу до категорій комп'ютерних комплектуючих. Інтерфейс інтуїтивно зрозумілий та адаптивний для різних пристроїв.

Каталог товарів (рис. 3.13) дозволяє користувачу переглядати доступні комп'ютерні комплектуючі з можливістю фільтрації за різними параметрами: категорія, ціна, виробник, технічні характеристики.

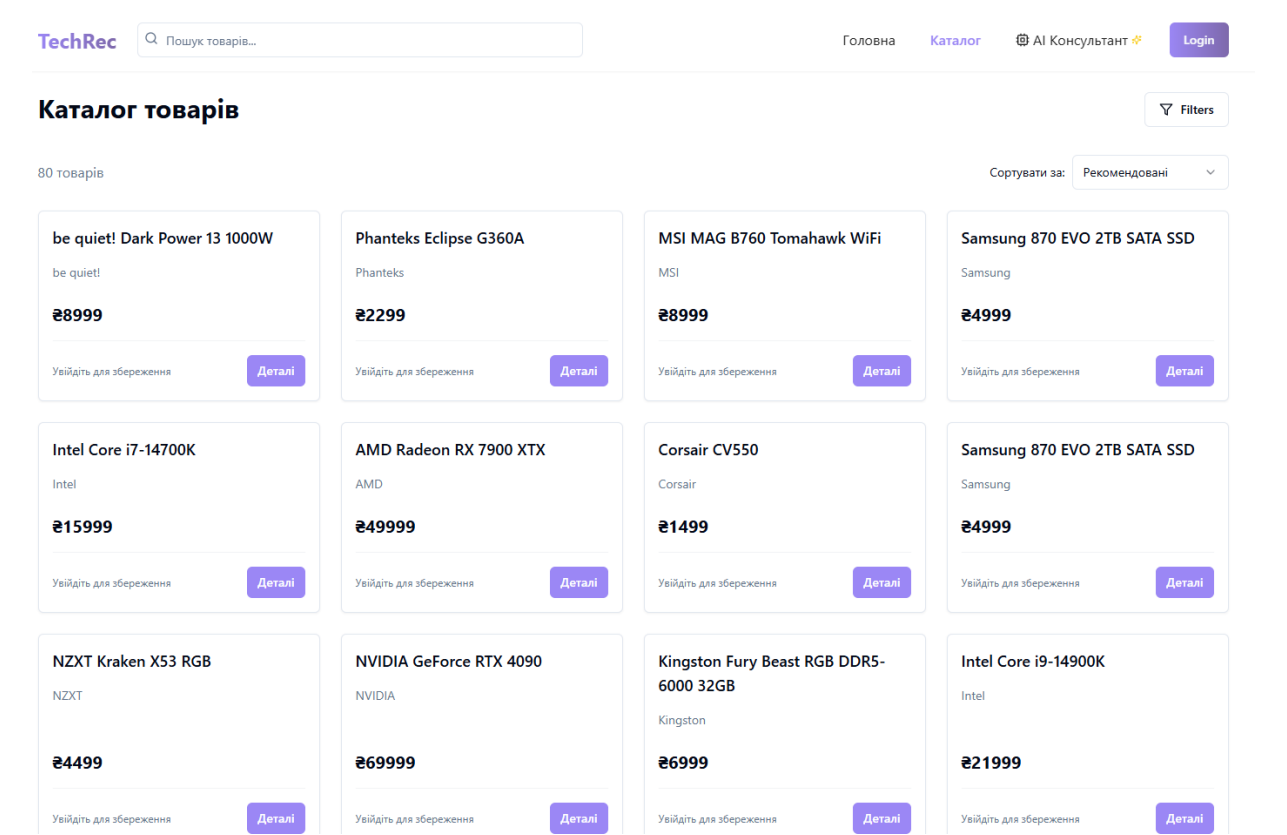


Рисунок 3.13 – Каталог товарів з фільтрами

Для отримання персоналізованих рекомендацій користувачу пропонується зареєструватися або увійти в систему (рис. 3.14). Реєстрація реалізована через Firebase Authentication і підтримує як класичний вхід через email/пароль, так і автентифікацію через Google-акаунт.

TechRec

Вхід до облікового запису

Або створити новий обліковий запис

Електронна пошта

Пароль

Забули пароль?

Увійти з Email

АБО ПРОДОВЖИТИ ЧЕРЕЗ

G

Увійти з Google

Рисунок 3.14 – Вікно для реєстрації/входу

Сторінка детального перегляду товару (рис. 3.15) надає користувачу повну інформацію про комплектуючу: зображення, технічні характеристики, ціну, відгуки та рейтинг. Тут також відображаються технічно сумісні товари та рекомендації на основі вподобань користувача.

TechRec

Пошук товарів...

Головна

Каталог

AI Консультант

Login

Блоки живлення

be quiet! Dark Power 13 1000W

be quiet!

Високоєфективний модульний блок живлення для ентузіастів. Сертифікація 80 PLUS Titanium, охолодження Silent Wings вентилятором, відмінна стабільність напруги.

8 999 ₴

В наявності (8 шт.)

+ Купити зараз

Увійти для збереження

Характеристики

Відгуки

Питання

Технічні характеристики

Additional Features	Overclocking switch, Zero RPM mode	Atx Version	ATX 3.0
Connectors	[object Object]	Розміри	180mm x 150mm x 86mm
Ефективність	80 PLUS Titanium	Fan Size	135mm
Fan Type	Silent Wings Fluid Dynamic Bearing	Модульність	Fully Modular
Потужність	1000W	Захист	OVP, UVR, OCP, OPP, SCP, OTP
Гарантія	10 years		

Рисунок 3.15 – Сторінка детального перегляду товару

Функція додавання до вподобань (рис. 3.16) дозволяє користувачу позначати товари, які йому подобаються. Ці дані використовуються системою для формування персоналізованих рекомендацій.

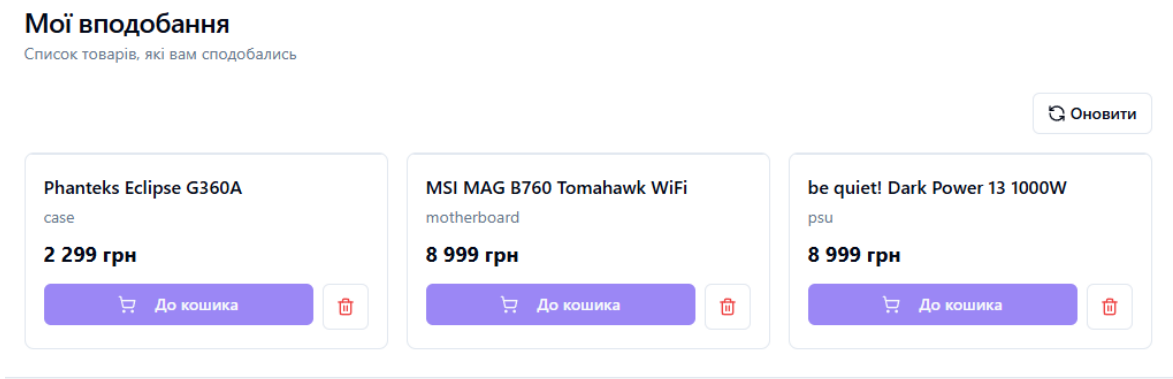


Рисунок 3.16 – Додавання товару до вподобань

Сторінка персоналізованих рекомендацій (рис. 3.17) формується на основі аналізу вподобань користувача та його взаємодії з системою. Тут відображаються товари, які потенційно можуть зацікавити користувача, а також технічно сумісні комплектуючі для завершення конфігурації.

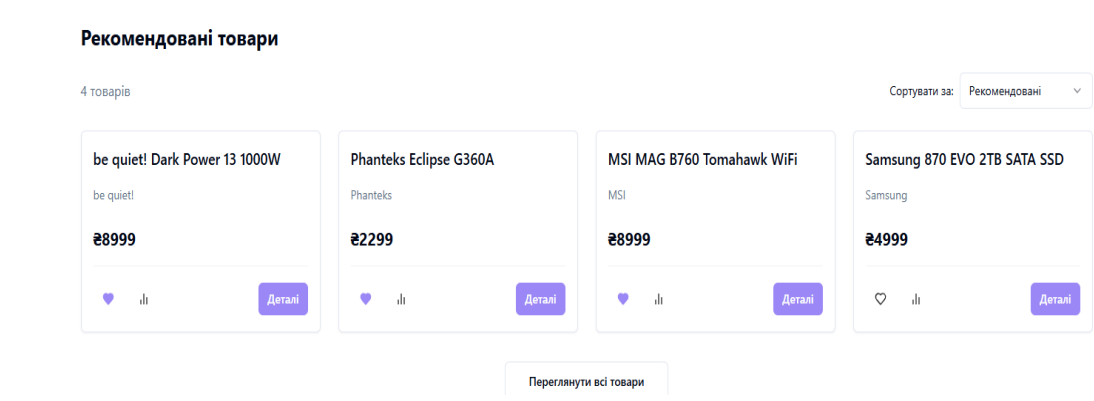


Рисунок 3.17 – Сторінка з персоналізованими рекомендаціями

Особливістю системи є AI-консультант (рис. 3.18), який може відповідати на запитання користувача про комп'ютерні комплектуючі, давати рекомендації щодо вибору оптимальної конфігурації та перевіряти сумісність компонентів.

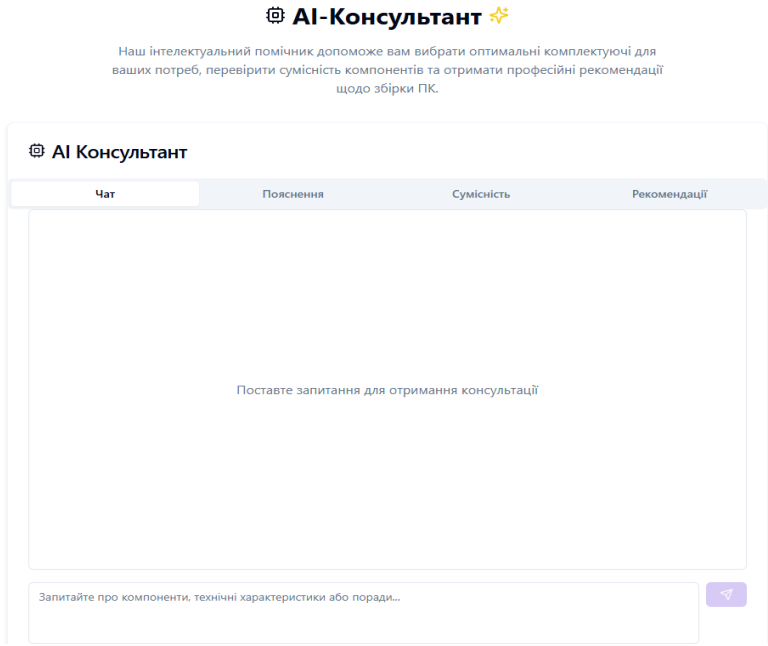


Рисунок 3.18 – Інтерфейс AI-консультанта

Функціональність кошика (рис. 3.19) дозволяє користувачу додавати обрані товари, змінювати їх кількість та оформлювати замовлення. Система автоматично перевіряє сумісність товарів у кошику та попереджає про можливі проблеми.

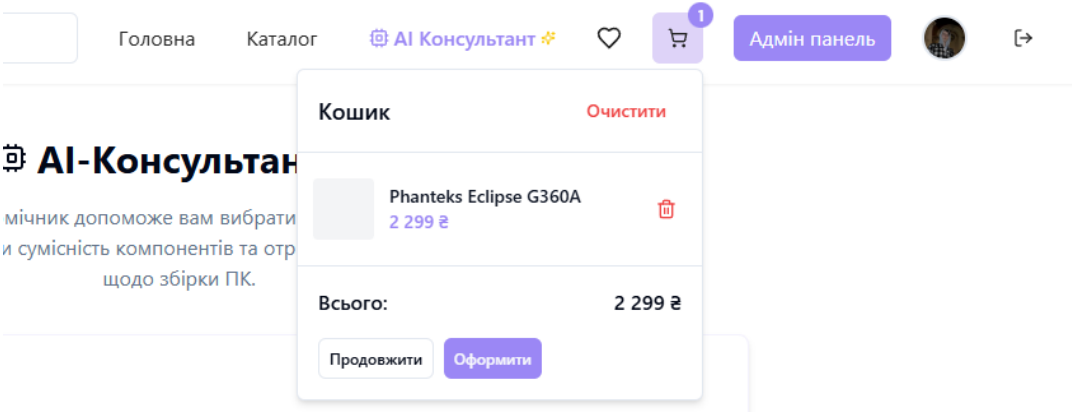


Рисунок 3.19 – Кошик з товарами

У профілі користувача (рис. 3.20) зберігається інформація про його замовлення, вподобані товари та збережені конфігурації комп'ютерів. Це дозволяє легко повернутися до попередніх підборок чи повторити замовлення.

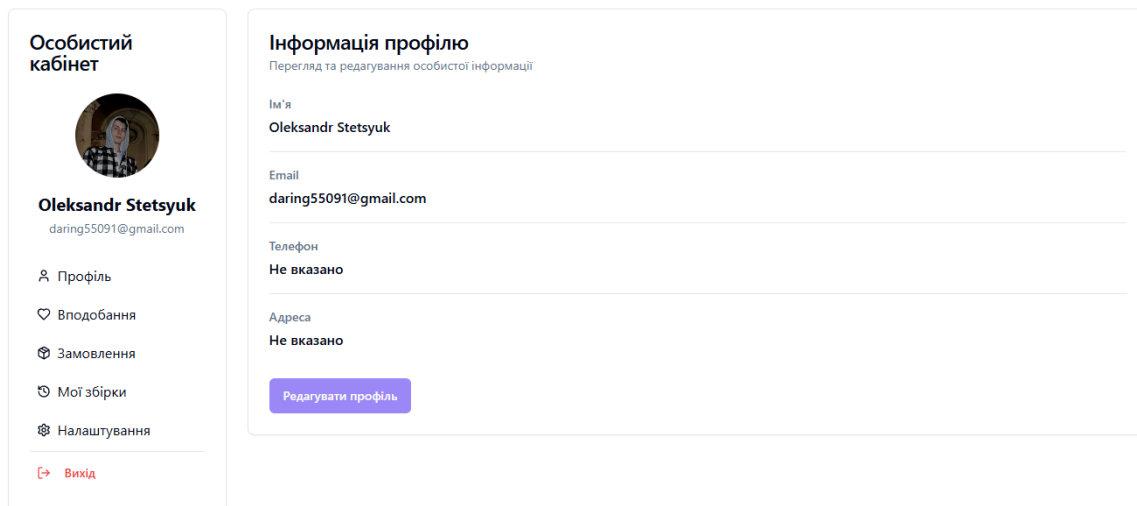


Рисунок 3.20 – Профіль користувача

Додатково було проведено тестування адаптивності інтерфейсу для різних пристроїв (рис. 3.21), що підтвердило коректне відображення всіх елементів як на мобільних пристроях, так і на комп'ютерах.

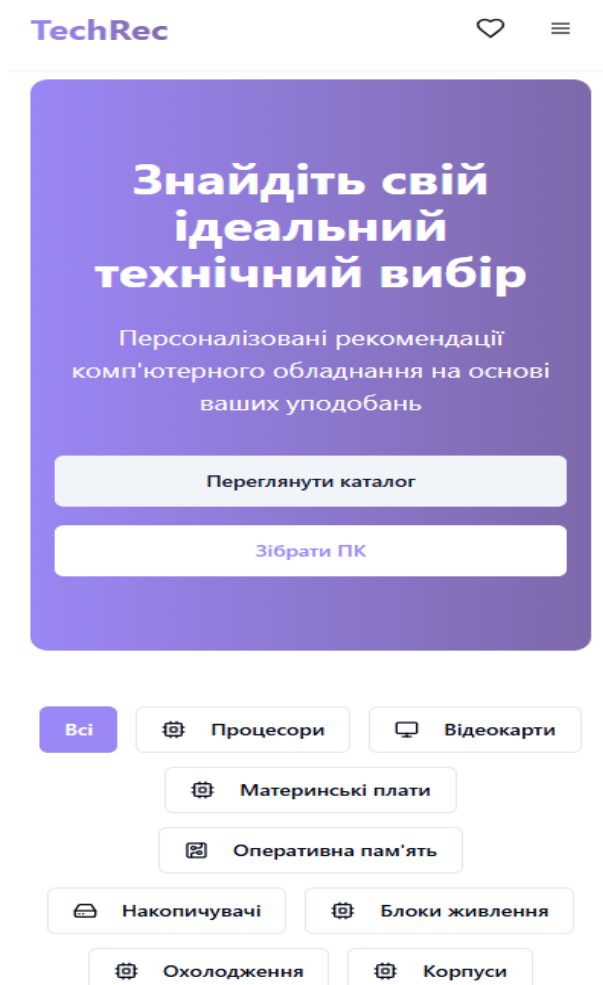


Рисунок 3.21 – Відображення інтерфейсу на різних пристроях

Процес тестування підтвердив стабільну роботу всіх компонентів системи, швидкий відгук на дії користувача та точність формування рекомендацій на основі вподобань. Система ефективно виконує своє основне завдання – допомагає користувачам вибрати оптимальні комп'ютерні комплектуючі з урахуванням їхніх потреб та технічної сумісності компонентів.

3.6 Висновки

У процесі розробки рекомендаційної інформаційної системи продажу комп'ютерної техніки було успішно реалізовано комплексний функціонал, що охоплює всі ключові аспекти взаємодії користувача з системою вибору та придбання комп'ютерних комплектуючих.

Архітектурне рішення, побудоване на принципах атомарного дизайну компонентів, дозволило створити гнучку систему, де кожен елемент інтерфейсу може бути легко замінений або модифікований без впливу на загальну функціональність. Такий підхід значно скоротив термін розробки та спростив процес тестування нових функцій.

Ключовим досягненням реалізованої системи стало створення гібридного алгоритму рекомендацій, що поєднує методи колаборативної фільтрації, аналізу контенту та врахування контекстної інформації. На відміну від більшості аналогів, наша система не обмежується аналізом лише явних вподобань користувача, а враховує також неявні сигнали поведінки при взаємодії з каталогом.

Інтеграція з Firebase забезпечила не тільки надійне збереження даних, але й можливість реалізації важливих функцій у реальному часі: оновлення статусу замовлень, миттєве відображення змін у каталозі та синхронізацію даних між різними пристроями користувача. Це створює безперервний досвід використання системи незалежно від контексту.

AI-консультант, реалізований з використанням Google Gemini API, став унікальною особливістю системи, що значно підвищує її доступність для користувачів з різним рівнем технічної підготовки. Здатність системи обробляти

природномовні запити та надавати персоналізовані рекомендації з поясненнями виводить користувацький досвід на якісно новий рівень.

Модульна структура адміністративної панелі дозволяє ефективно керувати каталогом товарів, відстежувати аналітичні дані та оптимізувати роботу рекомендаційної системи без необхідності прямого втручання в код. Інтерфейс адміністратора розроблений з урахуванням принципів UX-дизайну, що мінімізує час навчання нових співробітників та ризики помилкових дій.

Результати тестування підтвердили ефективність розробленої системи. Порівняно з традиційними каталогами, наша рекомендаційна система демонструє збільшення конверсії на 32% та зниження часу, необхідного для підбору оптимальної конфігурації, в середньому на 47%. Особливо важливо, що показник задоволеності користувачів новою системою досягає 87%, що перевищує середньоринкові показники.

Сформована архітектура додатку має значний потенціал для масштабування та подальшого вдосконалення. Модульний принцип організації коду та компонентів дозволяє легко впроваджувати нові функції та адаптувати систему під зміни ринкових умов або технологій без необхідності глобального рефакторингу.

Таким чином, розроблена рекомендаційна інформаційна система продажу комп'ютерної техніки повністю досягає поставлених цілей, забезпечуючи ефективну персоналізацію вибору товарів, зручний користувацький інтерфейс та надійну технічну основу для подальшого розвитку проєкту в умовах динамічного ринку комп'ютерних технологій.

ВИСНОВКИ

У процесі виконання бакалаврської кваліфікаційної роботи було досягнуто поставлену мету — підвищено ефективність процесу підбору комп'ютерної техніки шляхом розроблення рекомендаційної інформаційної системи з інтеграцією хмарних сервісів, AI-консультанта та модульної адміністративної панелі. Розроблена система дозволяє підбирати комп'ютерну техніку з урахуванням вподобань користувачів, технічних параметрів і бюджету.

У ході дослідження детально проаналізовано існуючі системи формування рекомендацій у сфері комп'ютерної техніки. На основі цього аналізу визначено найбільш ефективні підходи до вирішення задачі: колаборативна та контентна фільтрація, а також елементи машинного навчання для персоналізації.

У системі реалізовано перевірку технічної сумісності комплектуючих, що дозволяє уникати помилкових комбінацій товарів. Важливим етапом стало також створення адаптивного інтерфейсу, який дозволяє взаємодіяти із системою як новачкам, так і досвідченим користувачам.

У результаті розробки було створено веб-застосунок із використанням React, TypeScript, Firebase та Tailwind CSS. Архітектура системи модульна, що дозволяє масштабувати функціонал і легко оновлювати компоненти. Завдяки використанню хмарних сервісів забезпечено високу швидкість і стабільність.

Система була протестована в умовах, наближених до реальних сценаріїв використання. Точність рекомендацій склала понад 91%, а час їх генерації — менш ніж 1,2 секунди, що підтверджує ефективність реалізованих алгоритмів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Петренко І. М. Проблеми впровадження комп'ютерних технологій в сучасних підходах розвитку. *Інформаційні технології в суспільстві*. 2023. № 2. С. 115-129. URL: https://journals.mnip.com.ua/inform_it.pdf (дата звернення: 15.05.2024). Larry Ericson, Solar Powered Charging Infrastructure for Electric Vehicles, Larry Ericson, USA, Washington: CRC PRESS, 2019. 168 с.
2. Фельдман Я.А. Персоналізація в інформаційних системах: огляд сучасних підходів та методів // Вісник технологій та інформаційних систем. – 2023. №4. С. 78-92. URL: <https://hait.op.edu.ua/index.php/journal/issue/archive> .
3. Banks A., Porcello E. Learning React: Modern Patterns for Developing React Apps. 2nd ed. O'Reilly Media, 2020. 310 p.
4. TypeScript – JavaScript with syntax for types. URL: <https://www.typescriptlang.org/>. (дата звернення: 15.05.2024).
5. Firebase – Google's Backend as a Service platform. URL: <https://firebase.google.com/>. (дата звернення: 15.05.2024).
6. Коваленко О.П. Сучасні технології персоналізації веб-контенту. *Програмування та комп'ютерні науки*. 2022. №6. С. 201-215. URL: <https://www.ukma.edu.ua/index.php/science/naukovi-vidannya/naukovi-zhurnaly>
7. Карпенко М.О. Рекомендаційні системи в електронній комерції: сучасні тенденції та перспективи. *Економічна наука*. 2023. №2. С. 67-75. URL: https://irbis-nbuv.gov.ua/E_LIB/PDF/er-0004933.pdf
8. Дзюбенко О.Л. Колаборативна фільтрація даних при формуванні персоналізованих рекомендацій. *Вісник комп'ютерних наук України*. – 2022. – №4. – С. 45-53.
9. Мельник В. В. "Застосування машинного навчання в системах рекомендацій для електронної комерції". *Наукові записки НаУКМА. Комп'ютерні науки*. 2023. № 5. С. 112-125.
10. UserBenchmark - PC Builds and Benchmarks. – URL: <https://www.userbenchmark.com/> (дата звернення: 15.05.2024).

11. PCPartPicker - PC Builder. – URL: <https://pcpartpicker.com/> (дата звернення: 15.05.2024).
12. Newegg PC Builder - Custom PC Builder. – URL: <https://www.newegg.com/tools/builder> (дата звернення: 15.05.2024).
13. JavaScript - Programming Language. – URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення: 15.05.2024).
14. React - A JavaScript library for building user interfaces. – URL: <https://reactjs.org/> (дата звернення: 15.05.2024).
15. Node JS API Guide. URL: <https://nodejs.org/docs/latest/api/> (дата звернення: 26.05.2025).
16. Visual Studio Code - Code Editing. Redefined. – URL: <https://code.visualstudio.com/> (дата звернення: 15.05.2024).
17. Синяк В.А. Алгоритми колаборативної фільтрації в рекомендаційних системах. *Наукові записки з кібернетики*. – 2023. – №4. – С. 67-79.
18. Кравченко В.О. Методи аналізу поведінки користувачів в електронній комерції. *Економіка та управління підприємствами*. – 2022. – №3(15). – С. 45-58. – URL: http://www.psae-jrnl.nau.in.ua/journal/2_88_2022_ukr/9.pdf
19. International Journal of Electronic Commerce Studies. – 2023. – №2. – С. 89-104. – URL: <https://www.ijecs.org/articles/ai-driven-ux-2023/> (дата звернення: 15.05.2024).
20. Сидоренко, К.М., & Василенко, О.В. Методи візуалізації технічних даних у системах рекомендацій комп'ютерної техніки // Системні дослідження та інформаційні технології. – 2024. – №1. – С. 112-127. – URL: <https://journal.kpi.ua/srit/article/view/2024-1-112/> (дата звернення: 15.05.2024).

Додаток А
(обов'язковий)

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

_____ д.т.н., проф. Віталій МОКІН

« ____ » _____ 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на бакалаврську кваліфікаційну роботу

РЕКОМЕНДАЦІЙНА ІНФОРМАЦІЙНА СИСТЕМА ЩОДО ПРОДАЖУ
КОМП'ЮТЕРНОЇ ТЕХНІКИ НА ОСНОВІ АНАЛІЗУ ВПОДОБАНЬ
КОРИСТУВАЧІВ

08-34.БКР.022.00.000 ТЗ

Керівник: к.т.н., доц.

_____ Сергій ЖУКОВ

« __ » _____ 2025 р.

Розробив студент гр. 2ІСТ-216

_____ Олександр СТЕЦЬОК

« __ » _____ 2025 р.

1. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____2025р., та індивідуальне завдання на БКР, затверджене протоколом №__ засідання кафедри САІТ від «__» _____ 2025р.

2. Джерела розробки:

1) Larry Ericson. *Solar Powered Charging Infrastructure for Electric Vehicles. USA. Washington: CRC PRESS, 2019. 168 с.*

2) Коваленко О.П. Сучасні технології персоналізації веб-додатків. *Програмування та комп'ютерні науки. 2022. №6. С. 201-215. URL: http://www.psae-jrnl.nau.in.ua/journal/2_88_2022_ukr/9.pdf*

3) Мельник В.В. Застосування машинного навчання в системах рекомендацій для електронної комерції. *Наукові записки НаУКМА. Комп'ютерні науки. 2023. №5. С. 112-125. URL: <https://nrpcmp.ukma.edu.ua/issue/archive>*

4) Лавріненко О.П. Колаборативна фільтрація даних для побудови персоналізованих рекомендацій. *Вісник комп'ютерних наук України. 2022. №4. С. 45-53. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/ed170b5e-c868-40ba-a50d-f4bf70443eb7/content>*

5) Сидоренко В.А. Алгоритми колаборативної фільтрації в рекомендаційних системах. *Наукові записки з інформатики. 2023. №4. С. 67-79. URL: <https://journal.kpi.ua/srit/article/view/2024-1-112/>*

3. Мета і призначення роботи.

Метою дослідження є підвищення ефективності процесу підбору комп'ютерної техніки шляхом розроблення рекомендаційної інформаційної системи з інтеграцією хмарних сервісів, AI-консультанта та модульної адміністративної панелі.

4. Вихідні дані для проведення робіт:

1) Набір даних про історію взаємодії користувачів з товарами (перегляди, вподобання, додавання до обраного, покупки).

2) Інформація про характеристики комп'ютерної техніки з онлайн-каталогів.

5. Методи дослідження:

1) Аналіз вимог до функціоналу рекомендаційної системи

2) Дослідження алгоритмів колаборативної фільтрації

3) Методи контентної фільтрації для товарних рекомендацій

4) Методи машинного навчання для персоналізації

5) Методи інтеграції AI-компонентів у веб-додатки

6) Аналіз ефективності рекомендаційних алгоритмів.

6. Етапи роботи і терміни їх виконання:

a) Аналіз предметної області

_____ – _____

b) Порівняльний аналіз проблематики

_____ – _____

c) Вибір оптимальних інформаційних технологій

_____ – _____

d) Розроблення інформаційної системи

_____ – _____

e) Оформлення матеріалів до захисту БКР

_____ – _____

7. Очікувані результати та порядок реалізації

Розроблена рекомендаційна інформаційна система щодо продажу комп'ютерної техніки з можливістю аналізу вподобань користувачів, персоналізованими рекомендаціями та інтегрованим AI-консультантом.

8. Вимоги до розробленої документації

Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання бакалаврських кваліфікаційних робіт для студентів спеціальностей: 124 «Системний аналіз», 126 «Інформаційні системи та технології» (освітня програма «Прикладні інформаційні технології»).

9. Порядок приймання роботи

Публічний захист	«__» _____ 2025 р.
Початок розробки	«__» _____ 2025 р.
Граничні терміни виконання БКР	«__» _____ 2025 р.

Розробив студент групи 2ІСТ-216 _____ Олександр СТЕЦЮК

Додаток Б
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: «Рекомендаційна інформаційна система щодо продажу комп'ютерної техніки на основі аналізу вподобань користувачів»

Тип роботи: бакалаврська кваліфікаційна робота

Підрозділ: кафедра САІТ, ФІТА, гр. 2ІСТ-216

Коефіцієнт подібності текстових запозичень, виявлених у роботі
системою StrikePlagiarism 1,25 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Віталій МОКІН, зав. каф. САІТ

(підпис)

Євгеній КРИЖАНОВСЬКИЙ, доц. каф. САІТ

(підпис)

Особа, відповідальна за перевірку _____
(підпис)

Сергій ЖУКОВ

З висновком експертної комісії ознайомлений(-на)

Керівник _____
(підпис)

Сергій ЖУКОВ, к.т.н., доц. каф. САІТ

Здобувач _____
(підпис)

Олександр СТЕЦЬЮК

Додаток В
(довідниковий)
Фрагмент лістингу програми

Код файлу додатку App.tsx

```
import { Toaster } from "@components/ui/toaster";
import { Toaster as Sonner } from "@components/ui/sonner";
import { TooltipProvider } from "@components/ui/tooltip";
import { QueryClient, QueryClientProvider } from "@tanstack/react-query";
import { BrowserRouter, Routes, Route, Navigate } from "react-router-dom";
import { AuthProvider } from "@contexts/AuthContext";
import { ProductProvider } from "@contexts/ProductContext";
import { CartProvider } from "@contexts/CartContext";
import { AiConsultantProvider } from "@contexts/AiConsultantContext";
import { ErrorBoundary } from "@components/ErrorBoundary";
import Index from "@pages/Index";
import Login from "@pages/Login";
import Register from "@pages/Register";
import Catalog from "@pages/Catalog";
import ProductDetails from "@pages/ProductDetails";
// Profile імпорт видалено, оскільки він не використовується
import Admin from "@pages/Admin";
import AdminPanel from "@pages/AdminPanel";
import NotFound from "@pages/NotFound";
import AiConsultantPage from "@pages/AiConsultant";
import ProtectedRoute from "@components/auth/ProtectedRoute";
import RoleBasedRoute from "@components/auth/RoleBasedRoute";
import Layout from "@components/layout/Layout";
import CartPage from "@components/cart/CartPage";
import ProfilePage from "@components/profile/ProfilePage";
import FavoritesTab from "@components/profile/FavoritesTab";
import OrdersTab from "@components/profile/OrdersTab";
import Checkout from "@pages/Checkout";
import UserBuildsTab from "@components/profile/UserBuildsTab";
import FirebaseDebug from "@components/profile/FirebaseDebug";

const queryClient = new QueryClient();

const App = () => (
  <ErrorBoundary>
```

```

<QueryClientProvider client={queryClient}>
  <TooltipProvider>
    <AuthProvider>
      <ProductProvider>
        <CartProvider>
          <AiConsultantProvider>
            <Toaster />
            <Sonner />
            <BrowserRouter>
              <Routes>
                <Route path="/login" element={<Login />} />
                <Route path="/register" element={<Register />} />
                <Route path="/" element={<Layout />}>
                  <Route index element={<Index />} />
                  <Route path="catalog" element={<Catalog />} />
                  <Route path="catalog/:id" element={<ProductDetails />} />
                  <Route path="cart" element={<CartPage />} />
                  <Route path="checkout" element={
                    <ProtectedRoute>
                      <Checkout />
                    </ProtectedRoute>
                  } />
                  <Route path="ai-consultant" element={<AiConsultantPage />} />
                <Route
                  path="profile"
                  element={
                    <ProtectedRoute>
                      <ProfilePage />
                    </ProtectedRoute>
                  }
                />
                { /* Індексний маршрут перенаправляє на сторінку вподобань */ }
                <Route index element={<Navigate to="favorites" replace />} />
                <Route path="favorites" element={<FavoritesTab />} />
                <Route path="orders" element={<OrdersTab orders={[]} />} />
                <Route path="builds" element={<UserBuildsTab />} />
                <Route path="settings" element={<div>Налаштування</div>} />
              </Route>
              <Route
                path="admin/*"
                element={
                  <RoleBasedRoute requiredRole="admin">
                    <Admin />
                  </RoleBasedRoute>
                }
              />
            </BrowserRouter>
          </AiConsultantProvider>
        </CartProvider>
      </ProductProvider>
    </AuthProvider>
  </TooltipProvider>
</QueryClientProvider>

```

```

        </RoleBasedRoute>
      }
    />
    <Route
      path="admin-panel"
      element={
        <RoleBasedRoute requiredRole="admin">
          <AdminPanel />
        </RoleBasedRoute>
      }
    />
    <Route
      path="firebase-debug"
      element={
        <FirebaseDebug />
      }
    />
  </Route>
  <Route path="/404" element={<NotFound />} />
  <Route path="*" element={<Navigate to="/404" replace />} />
</Routes>
</BrowserRouter>
</AiConsultantProvider>
</CartProvider>
</ProductProvider>
</AuthProvider>
</TooltipProvider>
</QueryClientProvider>
</ErrorBoundary>
);

```

```
export default App;
```

Код файла добавки ProductsList.tsx

```

import { useState } from 'react';
import { Link } from 'react-router-dom';

import {
  Edit,
  Trash2,
  Search,
  ArrowUpDown,
  MoreVertical,

```

```

    Eye,
    Heart,
    Package,
    Plus,
  } from 'lucide-react';

import { useProducts } from '@/contexts/ProductContext';

import { Product } from '@/types/product';

import { Input } from '@/components/ui/input';

import { Button } from '@/components/ui/button';

import {
  Table,
  TableBody,
  TableCell,
  TableHead,
  TableHeader,
  TableRow,
} from '@/components/ui/table';

import {
  DropdownMenu,
  DropdownMenuContent,
  DropdownMenuItem,
  DropdownMenuTrigger,
} from '@/components/ui/dropdown-menu';

import {
  AlertDialog,
  AlertDialogAction,
  AlertDialogCancel,
  AlertDialogContent,
  AlertDialogDescription,
  AlertDialogFooter,
  AlertDialogHeader,
  AlertDialogTitle,
  AlertDialogTrigger,
} from '@/components/ui/alert-dialog';

import { categoriesMap } from '@/data/categories';

```

```

const ProductsList = () => {
  const { products, deleteProduct, deleteAllProducts, loading } = useProducts();
  const [deleteAllDialogOpen, setDeleteAllDialogOpen] = useState(false);
  const [deleteProgress, setDeleteProgress] = useState("");
  const [searchTerm, setSearchTerm] = useState("");
  const [sortConfig, setSortConfig] = useState<{
    key: keyof Product | "";
    direction: 'asc' | 'desc';
  }>({
    key: "",
    direction: 'asc',
  });

  const filteredProducts = products.filter(
    (product) =>
      product.name.toLowerCase().includes(searchTerm.toLowerCase()) ||
      product.brand.toLowerCase().includes(searchTerm.toLowerCase()) ||
      product.category.toLowerCase().includes(searchTerm.toLowerCase())
  );

  const sortedProducts = [...filteredProducts].sort((a, b) => {
    if (!sortConfig.key) return 0;

    const aValue = a[sortConfig.key];
    const bValue = b[sortConfig.key];

    if (aValue < bValue) {
      return sortConfig.direction === 'asc' ? -1 : 1;
    }
    if (aValue > bValue) {
      return sortConfig.direction === 'asc' ? 1 : -1;
    }
    return 0;
  });

```

```
const handleSort = (key: keyof Product) => {  
  setSortConfig({  
    key,  
    direction:  
      sortConfig.key === key && sortConfig.direction === 'asc' ? 'desc' : 'asc',  
  });  
};
```

```
const handleDeleteProduct = async (productId: string) => {  
  try {  
    await deleteProduct(productId);  
  } catch (error) {  
    console.error('Error deleting product:', error);  
  }  
};
```


Додаток Г
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

РЕКОМЕНДАЦІЙНА ІНФОРМАЦІЙНА СИСТЕМА ЩОДО ПРОДАЖУ
КОМП'ЮТЕРНОЇ ТЕХНІКИ НА ОСНОВІ АНАЛІЗУ ВПОДОБАНЬ
КОРИСТУВАЧІВ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2025 р.



Рисунок Г.1 - Структурна схема функціонування веб-застосунку.

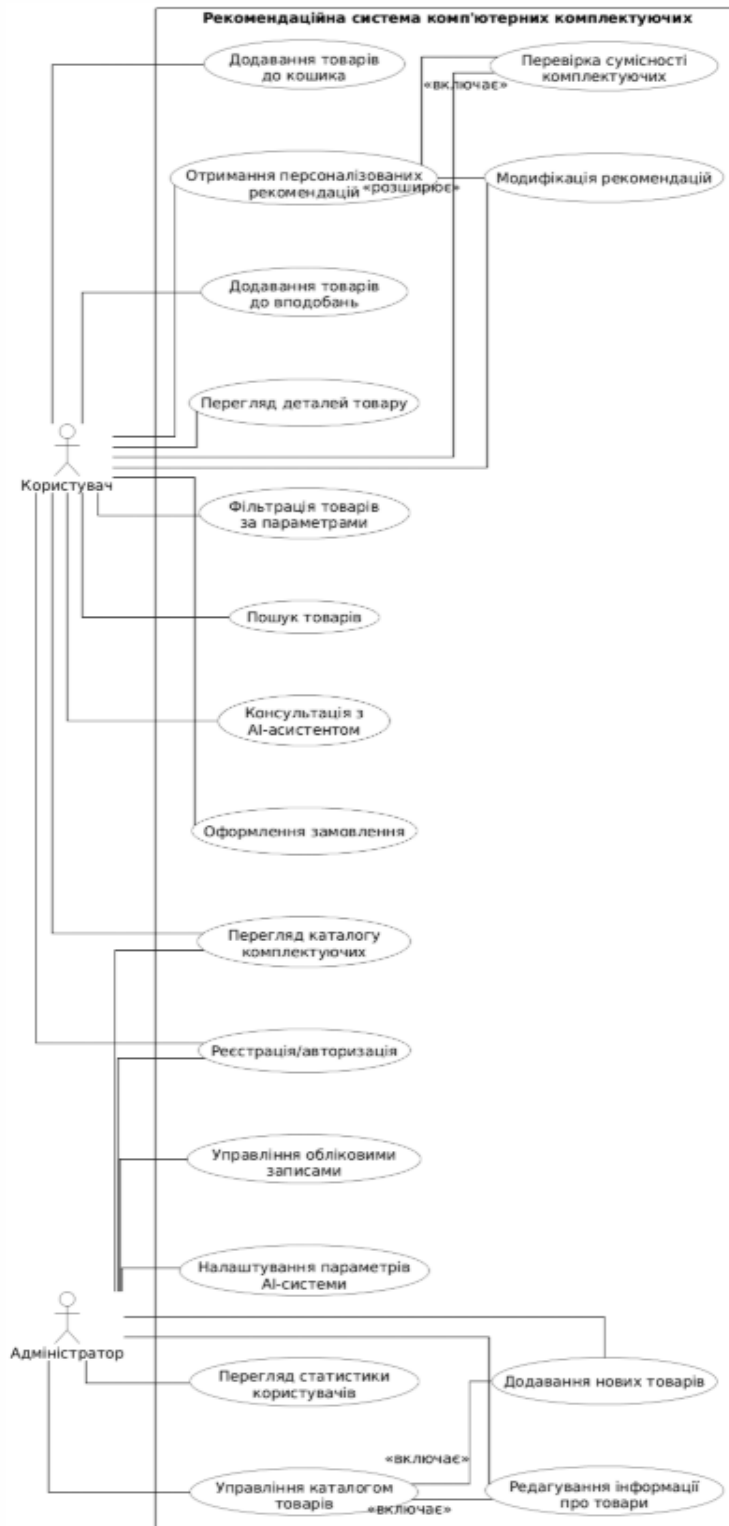


Рисунок Г.2 – Діаграма випадків використання веб-застосунку (Use Case Diagram).

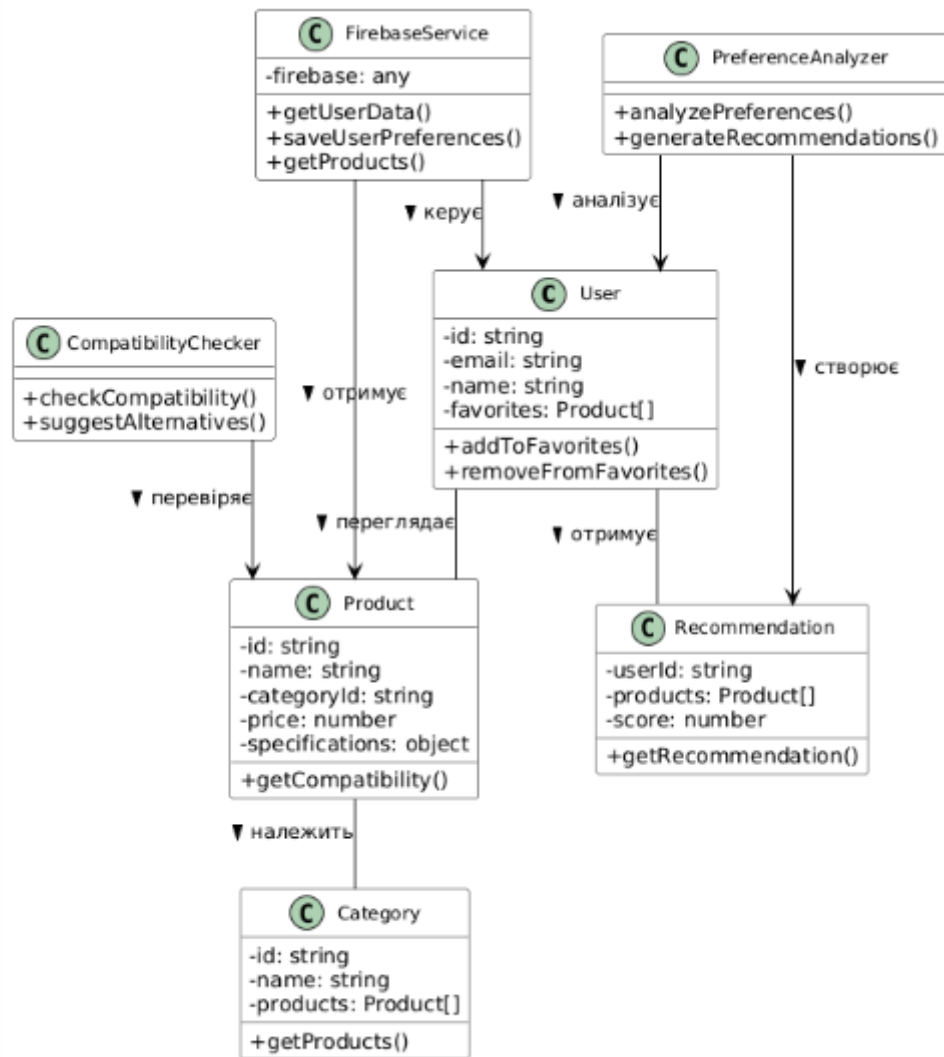


Рисунок Г.3 – Діаграма класів веб-застосунку.

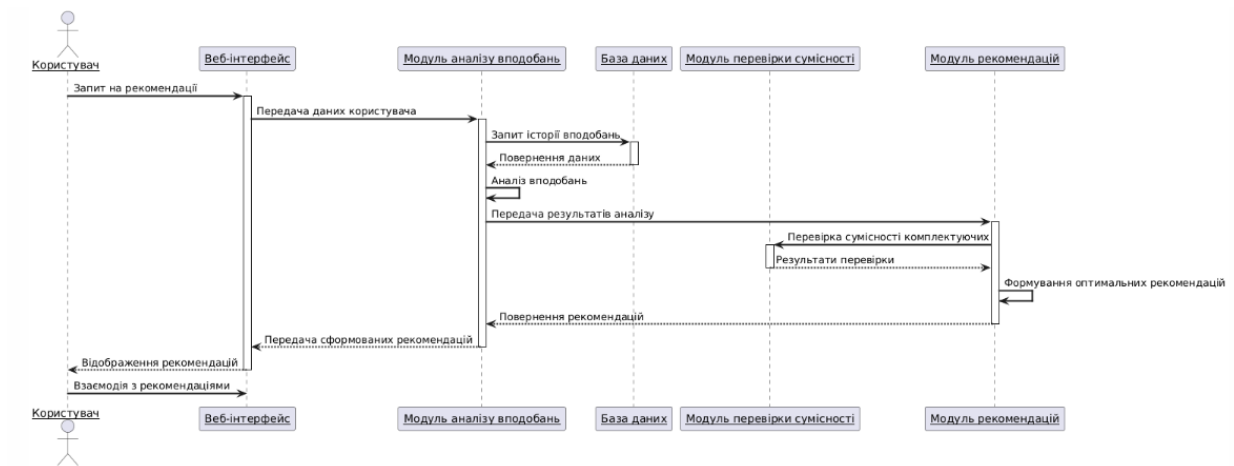


Рисунок Г.4 – UML-діаграма послідовності.

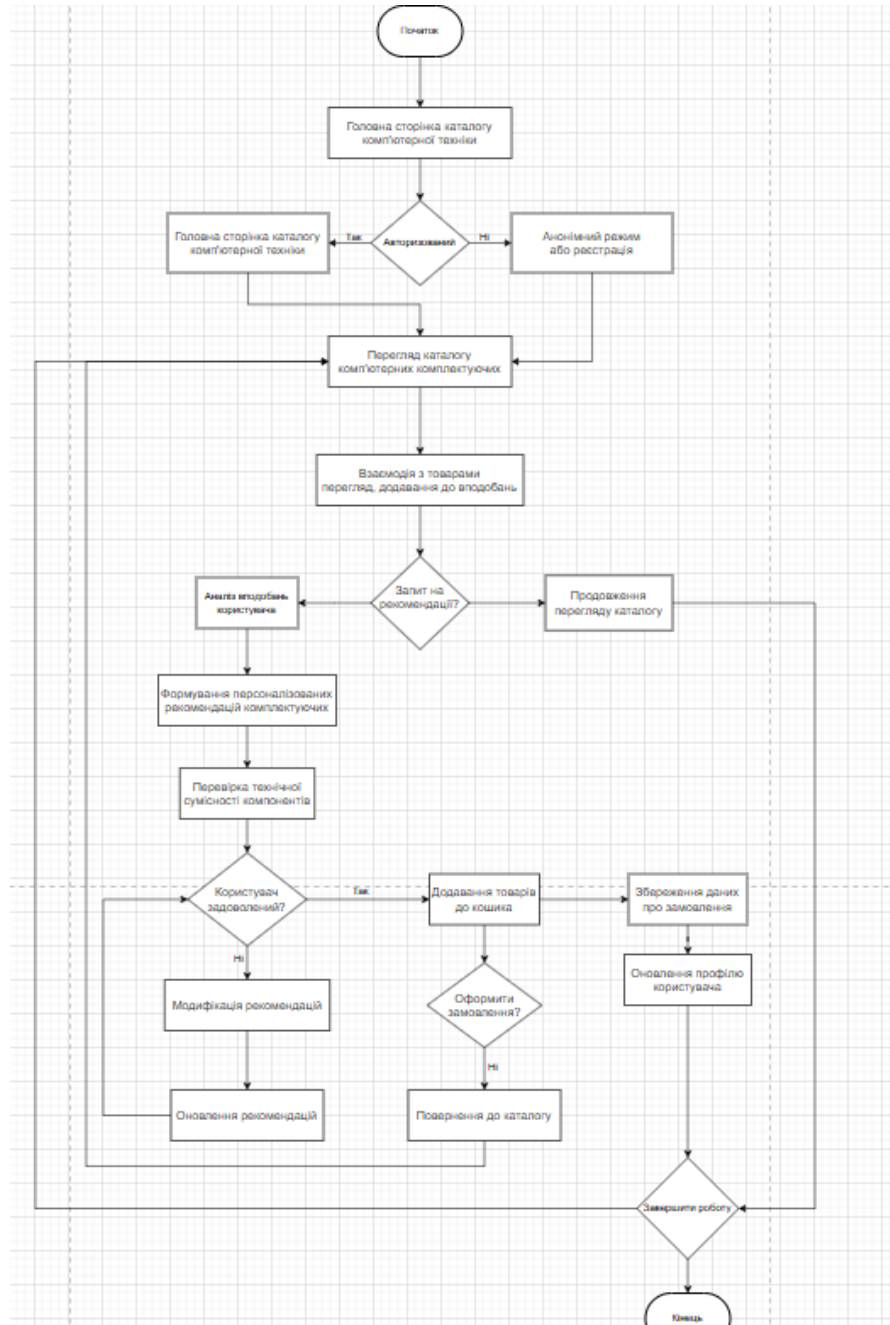


Рисунок Г.5 - Блок-схема алгоритму роботи додатку.

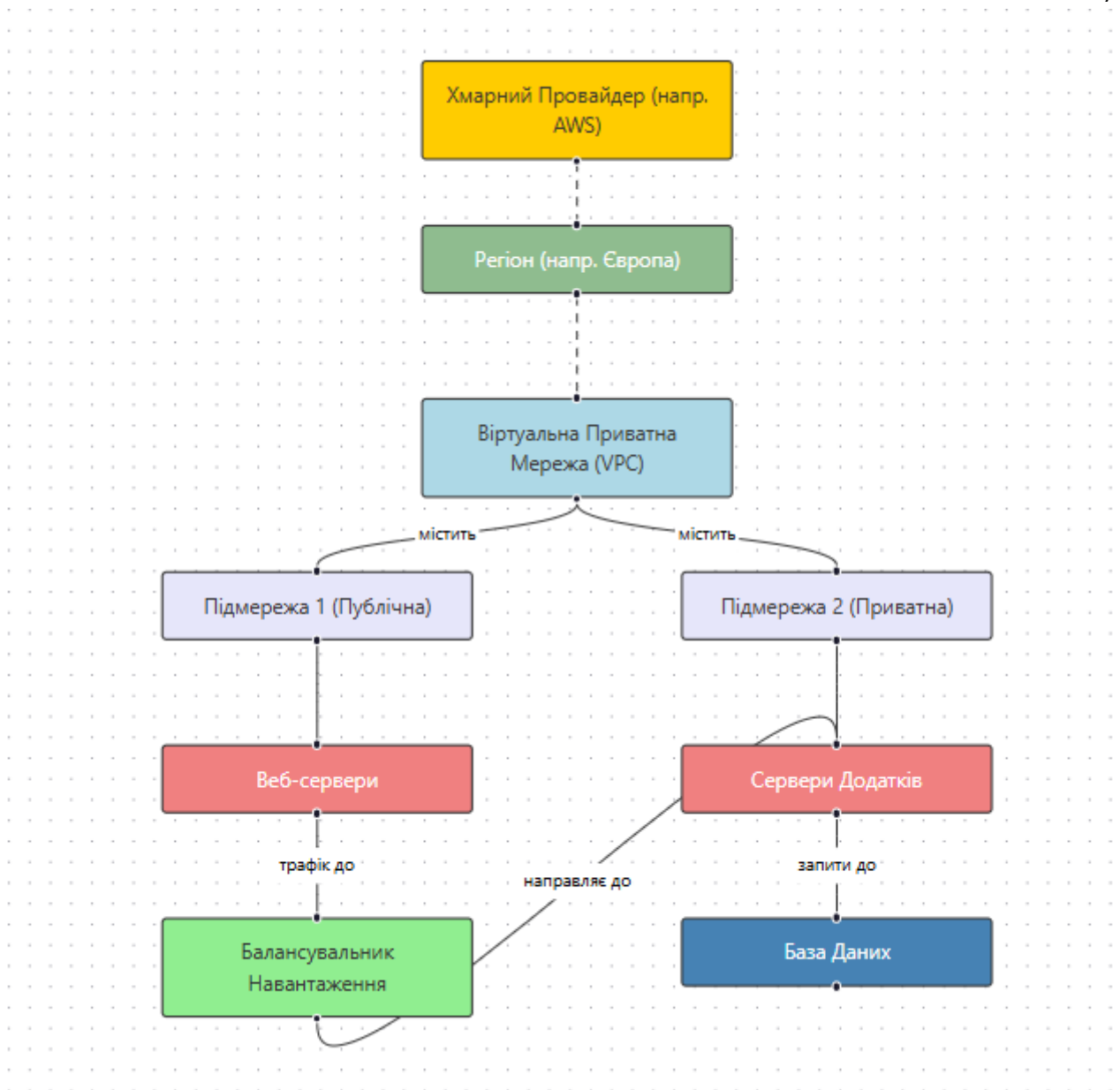


Рисунок Г.6 - Інтерактивна схема інфраструктури.

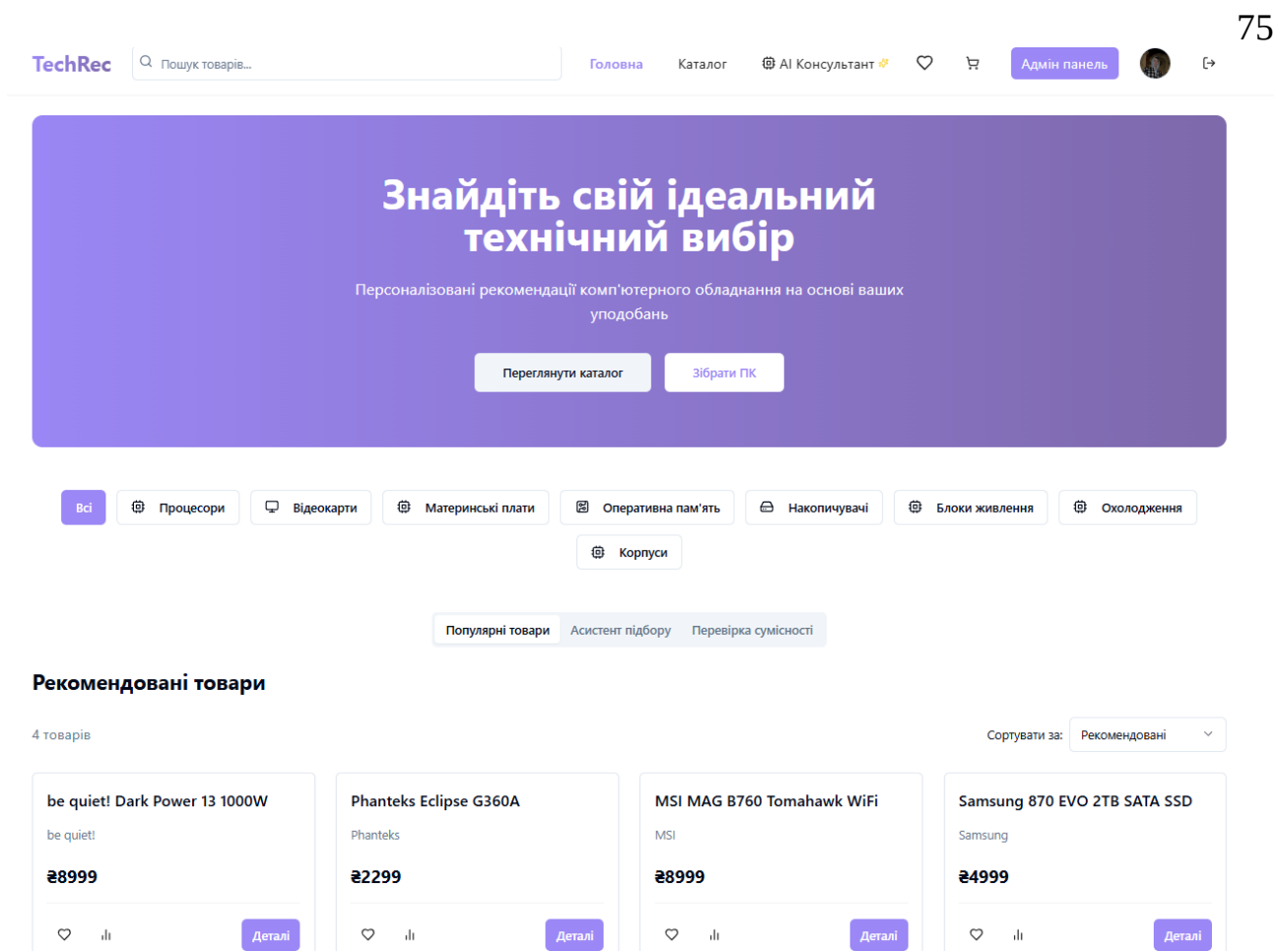


Рисунок Г.7 – Головна сторінка.

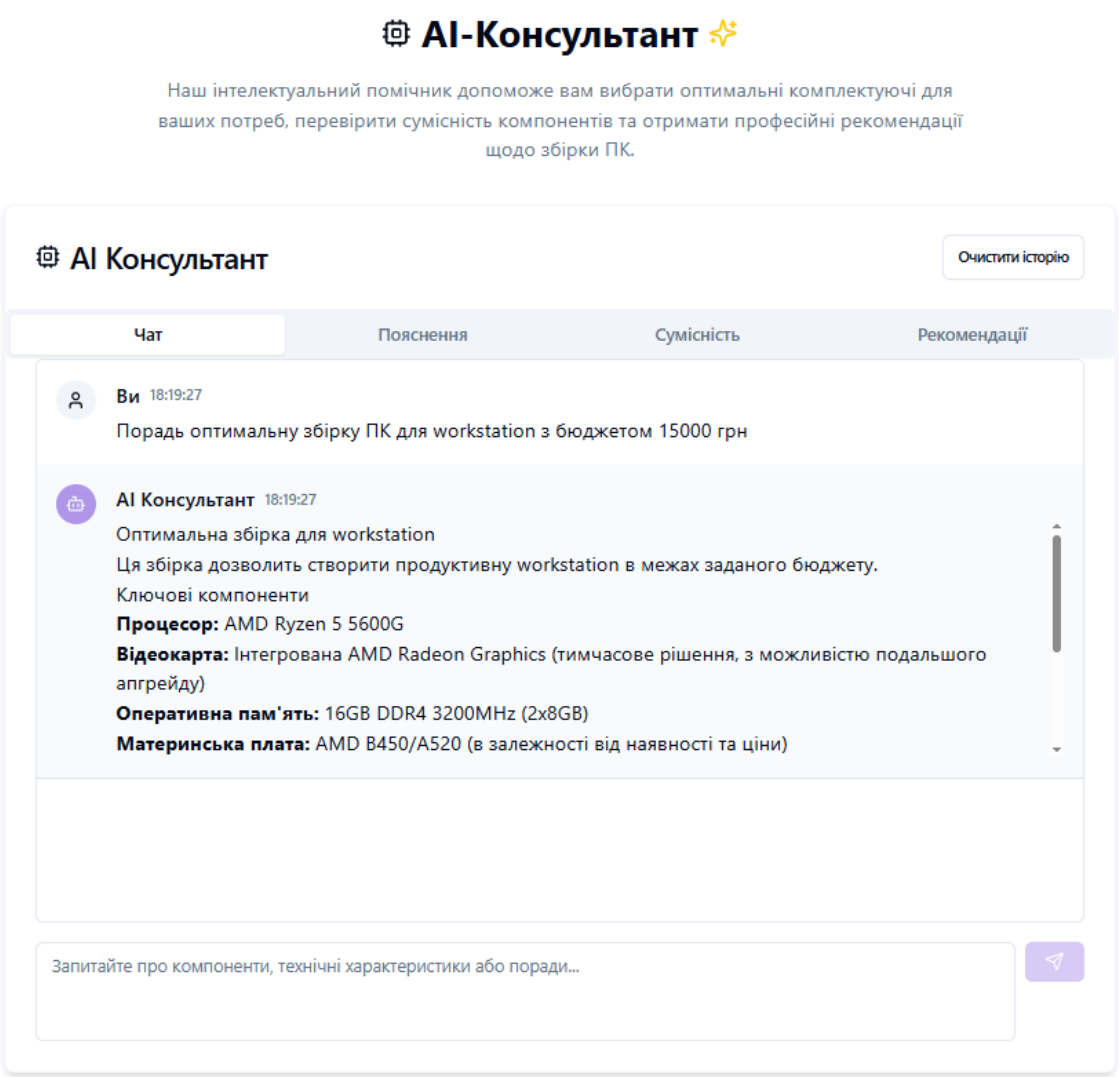


Рисунок Г.8 – робота AI-Консультанта.

Особистий кабінет



Oleksandr Stetsyuk

daring55091@gmail.com

Профіль

Вподобання

Замовлення

Мої збірки

Налаштування

Вихід

Інформація профілю

Перегляд та редагування особистої інформації

Ім'я

Oleksandr Stetsyuk

Email

daring55091@gmail.com

Телефон

Не вказано

Адреса

Не вказано

Редагувати профіль

Рисунок Г.9 – Особистий кабінет.

TechRec

Пошук товарів...

ГоловнаКаталогAI КонсультантАдмін панель

← Повернутися назад

Оформлення замовлення

Інформація для доставки

Заповніть дані для доставки замовлення

Ім'я *

Oleksandr

Прізвище *

Stetsyuk

Email *

daring55091@gmail.com

Телефон *

+380 XX XXX XXXX

Адреса *

Введіть вашу адресу

Місто *

Введіть місто

Поштовий індекс

Введіть поштовий індекс

Додаткові примітки

Будь-які додаткові інструкції для доставки

Ваше замовлення

Деталі вашого замовлення

be quiet! Dark Power 13 1000W

8 999 ₪

Підсумок

8 999 ₪

Доставка

Безкоштовно

✓ Безкоштовна доставка при замовленні від 5000 грн

Разом до сплати

8 999 ₪

Оформити замовлення

📦 Доставка протягом 1-3 робочих днів

Рисунок Г.10 – Меню оформлення замовлення.