

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА**  
**ПРИРОДОКОРИСТУВАННЯ**  
**Навчально-науковий інститут кібернетики,**  
**інформаційних технологій та інженерії**

"До захисту допущена"

Зав. кафедри комп'ютерних наук та  
прикладної математики

---

«\_\_» \_\_\_\_\_ 20\_\_ р.

**КВАЛІФІКАЦІЙНА РОБОТА**

**Проектування та розробка веб-додатку для маркетплейсу**

Виконав: **Божок Віктор Сергійович**

(прізвище, ім'я, по батькові)

група ПЗ-42інт.

---

(підпис)

Керівник: доцент, **к.т.н. Климюк Ю. Є.**

(науковий ступінь, вчене звання, посада, прізвище, ініціали)

---

(підпис)

## ЗМІСТ

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ	
РЕФЕРАТ	3
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП	5
РОЗДІЛ I. ТЕОРЕТИЧНА ЧАСТИНА	8
1.1. Складові компоненти системи	8
1.2. Опис вибраних технологій	16
РОЗДІЛ II. АНАЛІТИЧНА ЧАСТИНА	18
2.1. Технічне завдання	18
2.2. Порівняння з аналогами	30
2.3. Алгоритми та БД	32
РОЗДІЛ III. ПРИКЛАДНА ЧАСТИНА	36
3.1. Контейнеризація системи	36
3.2. Інструкція з користування	38
ВИСНОВКИ	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	48
ДОДАТКИ	49
Додаток А. Файл docker-compose.prod.yml	49

## РЕФЕРАТ

Кваліфікаційна робота: 54 сторінок, 39 малюнків, 2 таблиці, 1 додаток, 10 джерел.

**Метою кваліфікаційної роботи** є проектування та розробка веб-додатку для маркетплейсу з вбудованим ChatGPT.

**Об'єкт дослідження** – процес купівлі-продажу товарів під час використання маркетплейсу.

**Предметом дослідження** – методи та засоби купівлі-продажу під час використання маркетплейсу з вбудованим ChatGPT.

**Методи дослідження** – мови програмування Python й TypeScript, фреймворки Django та Next.js.

Оскільки покупцям доводиться витратити час у пошуках товару за найвигіднішою ціною у різних місцях, то виникає потреба у єдиному місці, де можна знайти товар за один день з відображенням усіх цін, умов покупки та рейтингу продавця.

**Ключові слова:** **МАРКЕТПЛЕЙС, ВЕБ-СЕРВІС, TYPESCRIPT, DJANGO.**

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- Django** — високорівневий відкритий Python-фреймворк для розробки вебсистем.
- FastAPI** — це високопродуктивний веб-фреймворк для створення API на основі Python.
- TypeScript** — це надмножина JavaScript, що означає, що ви можете використовувати весь ваш наявний JavaScript-код без змін.
- Next.js** — це вебфреймворк, що дозволяє розробляти веб-додатки на основі React із рендерингом на стороні сервера та генерацією статичних веб-сайтів.
- NGINX** — вільний вебсервер і проксі-сервер.
- Docker** — інструментарій для управління ізольованими Linux-контейнерами.
- PostgreSQL** — об'єктно-реляційна система керування базами даних.
- Redis** — розподілене сховище пар ключ-значення, які зберігаються в оперативній пам'яті, з можливістю забезпечувати довговічність зберігання на бажання користувача.

## ВСТУП

У сучасному світі процес купівлі товарів став невід'ємною частиною повсякденного життя, однак він часто залишається складним та часозатратним. Люди завжди враховують кілька ключових пріоритетів при виборі товарів: ціну, якість, умови покупки та витрачений час. Ці пріоритети можуть варіюватися залежно від конкретних потреб та обставин, але основний пріоритет завжди стоїть на першому місці. Наприклад, для когось ключовим фактором може бути ціна, тоді як інші надають перевагу якості або швидкості доставки. Таким чином, різні споживачі можуть витратити значно різну кількість часу на пошуки відповідного товару.

Пошук потрібного товару включає в себе ознайомлення з різними пропозиціями на ринку, аналіз умов покупки, оцінку надійності продавців, перевірку наявності товару та зручності доставки. Цей процес може займати значний час, особливо якщо пошук виконується вручну. Розглянемо, наприклад, ситуацію, коли споживач хоче придбати новий смартфон. Йому необхідно вивчити численні оголошення, порівняти ціни, прочитати відгуки, перевірити репутацію продавців і переконатися в наявності бажаного товару. У зв'язку з цим виникає потреба в автоматизації процесу купівлі, що дозволить споживачам швидше знаходити необхідні товари, зменшуючи витрати часу та зусиль.

Актуальність теми даної кваліфікаційної роботи полягає у необхідності розробки автоматизованої системи, яка б дозволила споживачам оперативно отримувати інформацію про наявні пропозиції на ринку, забезпечуючи зручний доступ до даних про товар, його вартість, умови покупки та доставку. Така система могла б значно покращити процес купівлі-продажу, зробивши його більш ефективним та менш стресовим. Наприклад, автоматизована система могла б надавати споживачам можливість отримувати оновлену інформацію про найкращі пропозиції у реальному часі, що значно зменшило б необхідність ручного пошуку та порівняння.

Наукова новизна даної роботи полягає у розробці інноваційних підходів до автоматизації процесу пошуку товарів, що враховують індивідуальні пріоритети споживачів. Вона включає розробку алгоритмів, які здатні ефективно обробляти великі обсяги даних про товари та продавців, надаючи користувачам найбільш релевантні результати. Практична значимість роботи визначається можливістю впровадження запропонованої системи в роботу інтернет-магазинів, що сприятиме покращенню якості обслуговування клієнтів та підвищенню конкурентоспроможності підприємств на ринку. Це дозволить скоротити час на пошук товарів, підвищити рівень задоволеності клієнтів та збільшити обсяги продажів.

Метою даної кваліфікаційної роботи є проєктування та розробка веб-додатку для маркетплейсу з вбудованим ChatGPT. Для досягнення цієї мети були поставлені завдання дослідити сучасний стан технологій автоматизації процесу купівлі-продажу, визначити основні пріоритети споживачів при виборі товарів, розробити алгоритми для автоматизованого пошуку товарів, що враховують ці пріоритети, створити прототип автоматизованої системи та провести його тестування, а також оцінити ефективність запропонованих рішень і розробити рекомендації щодо їх впровадження.

Основними методами дослідження, що використовувалися у роботі, були моделювання, порівняльний аналіз існуючих систем автоматизації процесу купівлі-продажу та експериментальне тестування розроблених алгоритмів на реальних покупцях і продавцях. Моделювання дозволило створити теоретичну основу для розробки алгоритмів, порівняльний аналіз надав змогу ідентифікувати сильні та слабкі сторони існуючих рішень, а тестування на реальних користувачах дозволило отримати об'єктивні результати щодо ефективності запропонованої системи. Використання реальних даних та практичних тестів забезпечило високий рівень достовірності та релевантності отриманих результатів.

Таким чином, дана кваліфікаційна робота спрямована на вирішення актуальної проблеми оптимізації процесу купівлі-продажу товарів за допомогою сучасних інформаційних технологій. Розроблені у рамках дослідження підходи та рішення можуть бути використані для покращення якості обслуговування клієнтів і підвищення ефективності роботи інтернет-магазинів, що сприятиме загальному розвитку електронної комерції. Використання автоматизованих систем для пошуку товарів дозволить не лише скоротити час, необхідний для прийняття рішення про покупку, але й забезпечить споживачів більш точною та релевантною інформацією, що підвищить рівень їх задоволеності та лояльності. Це дослідження має потенціал стати важливим внеском у розвиток інноваційних технологій у сфері електронної комерції, сприяючи створенню більш зручних і ефективних систем для споживачів та бізнесу.

## **РОЗДІЛ I. ТЕОРЕТИЧНА ЧАСТИНА**

### **1.1. Складові компоненти системи**

Система для автоматизації процесу купівлі-продажу товарів включає в себе декілька ключових компонентів: реєстрація, авторизація, кабінет, дані профілю, запити, чат.

#### **1. Реєстрація**

Користувач заповнює форму на рис. 1.1 з особистими даними, такими як ім'я, прізвище, електронна пошта, телефонний номер та інші необхідні дані. Після введення особистих даних, покупцю пропонується створити обліковий запис, вибравши унікальний логін та пароль для входу на платформу. Після надання необхідної інформації покупець натискає кнопку "Зареєструватися", і система надсилає підтвердження на вказану електронну пошту з посиланням для активації облікового запису. Покупець переходить за посиланням у електронному листі, після чого обліковий запис підтверджується та стає активним.

The image shows a registration form for a buyer (Покупець) within a system that also supports a seller (Продавець). The form is titled 'Покупець' and includes the following fields and elements:

- Имя (Name): Input field.
- Прізвище (Surname): Input field.
- По батькові (Patronymic): Input field.
- Електронна пошта (Email): Input field containing 'email@example.com'.
- Номер телефону (Phone number): Input field.
- Checkbox: A toggle switch is currently turned off.
- Text: Below the checkbox, it reads: 'Реєструючись, я ознайомлений(-а) та погоджуюсь з [Умовами публічної оферти](#), [Політики конфіденційності](#) та повідомленням про [файли cookie](#)'.
- Button: A blue button labeled 'Зареєструватися' (Register).

Рис. 1.1. Форма реєстрації покупця

Продавець заповнює форму на рис. 1.2, вводить інформацію про свою компанію, таку як тип юридичної особи, ЄДРПОУ, назву компанії, адресу реєстрації, контактні дані тощо. Після введення необхідної інформації продавець натискає кнопку "Зареєструватися", і система надсилає підтвердження на вказану електронну пошту з подальшими інструкціями. Продавець переходить за посиланням у електронному листі, після чого обліковий запис підтверджується та стає активним.

The image shows a registration form for a seller (Продавець) with the following fields and content:

- Таблиця навігації: Покупець | **Продавець**
- Тип юридичної особи:  (dropdown menu)
- ЄДРПОУ:
- Назва юр. особи:
- Електронна пошта:
- Номер телефону директора:
- Номер телефону менеджера:
- Населений пункт:
- Посилання на сайт або сторінку в соц. мережі:
- Адреса реєстрації компанії:
- Терми умов:  Реєструючись, я ознайомлений(-а) та погоджуюсь з [Умовами публічної оферти](#), [Політики конфіденційності](#) та повідомленням про [файли cookie](#)
- Кнопка: Зареєструватися

Рис. 1.2. Форма реєстрації продавця

## 2. Авторизація

Як і покупець, так і продавець для авторизація мають вказати свою електронну пошту та пароль у форму на рис. 1.3.

**Вхід в обліковий запис**

Електронна пошта

Пароль

[Забули пароль?](#)

Не маєте облікового запису? [Реєстрація](#)

Рис. 1.3. Форма авторизації

### 3. Дані профілю

На рис. 1.4 зображено вкладку «Дані профілю», де покупець може змінити свою аватарку, населений пункт, П.І.Б., номер телефону, електронну пошту та пароль.

**DUOSHOP**

- Кабінет
- Дані профілю**
- Мої замовлення
- Чати

**Сергій Пащенко**  
Покупець

**П.І.Б.**

Прізвище

Ім'я

По батькові

**Змінити електронну пошту**

**Змінити пароль**

**Населений пункт**

**Номер телефону**

Бажаний розмір зображення: 320px x 320px

Рис. 1.4. Дані профілю покупця

На рис. 1.5 зображено вкладку «Дані профілю», де продавець може змінити свою аватарку, населений пункт, номер телефону, електронну пошту, посилення на feed з інформацією про усі товари в його інтернет-магазині та пароль. Також продавець тут може оплатити доступ до сайту.

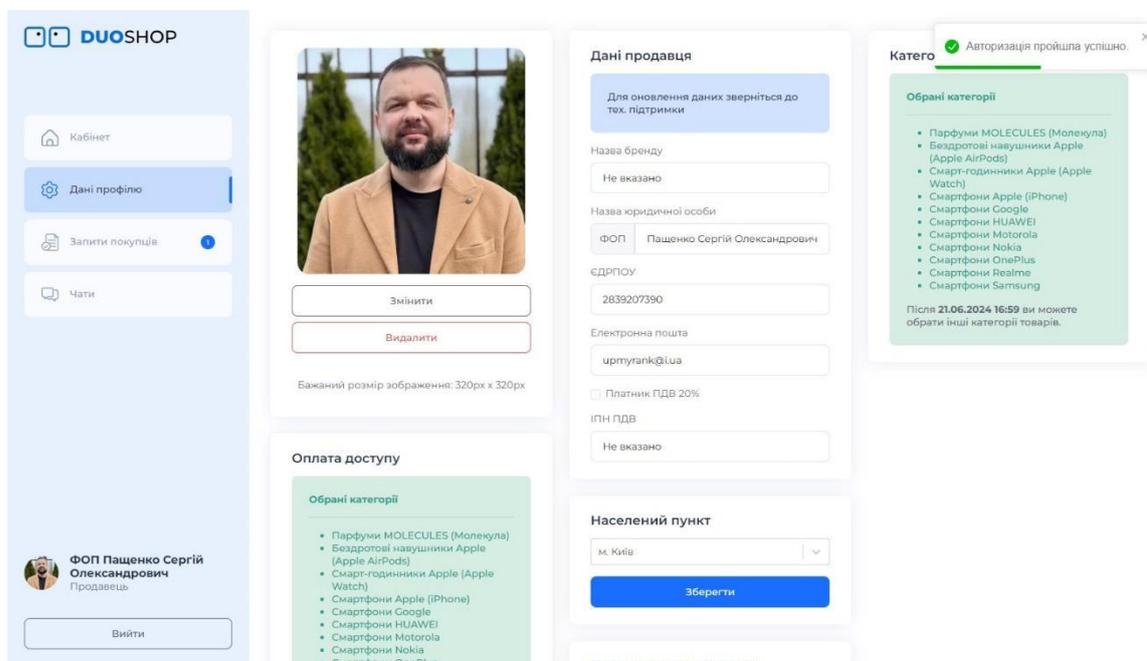


Рис. 1.5. Дані профілю продавця

#### 4. Кабінет

На рис. 1.6 зображено вкладку «Кабінет», де покупець може створити замовлення вказавши назву товару, його кількість, побажання, отримувача, спосіб доставки, оплати та де шукати продавця.

The screenshot shows the 'Створення замовлення' (Create Order) page in the DuoShop application. On the left is a sidebar with navigation options: 'Кабінет', 'Дані профілю', 'Мої замовлення', and 'Чати'. The main content area is titled 'Створення замовлення' and includes a success notification: 'Авторизація пройшла успішно.' Below this are several sections: 'Товар' (Product) with fields for 'Назва товару' and 'Кількість' (set to 1), and a 'Підтвердити' button; 'Дані отримувача' (Recipient Data) with a warning about passport requirements and fields for 'Прізвище' (Пашенко), 'Ім'я' (Сергій), 'По батькові' (Олександрович), and 'Номер телефону' (+380988165784); 'Доставка' (Delivery) with radio buttons for 'Нова Пошта', 'Самовивіз' (selected), and 'Кур'єрська доставка'; 'Оплата' (Payment) with radio buttons for 'Онлайн' (selected) and 'При отриманні'; and 'Пошук продавця' (Seller Search) with radio buttons for 'Вся країна', 'В межах регіону', and 'В межах населеного пункту' (selected). A sidebar at the bottom left shows the user's profile: 'Сергій Пашенко, Покупець' with a 'Вийти' button.

Рис. 1.6. Кабінет покупця

На рис. 1.7 зображено вкладку «Кабінет», де продавець може спостерігати за своєю статистикою угод.

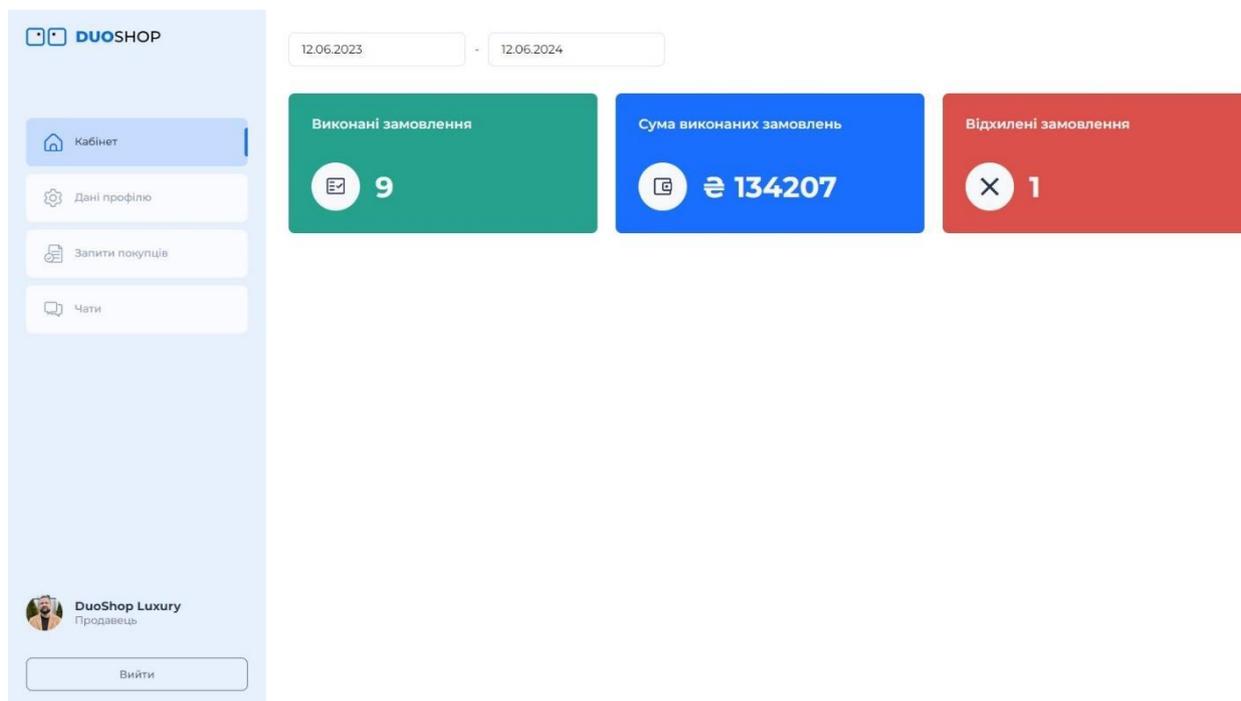


Рис. 1.7. Кабінет продавця

## 5. Запити

На рис. 1.8 зображено вкладку «Мої запити», де покупець може переглядати свої запити та пропозиції до них зі сторони продавців у реальному часі.

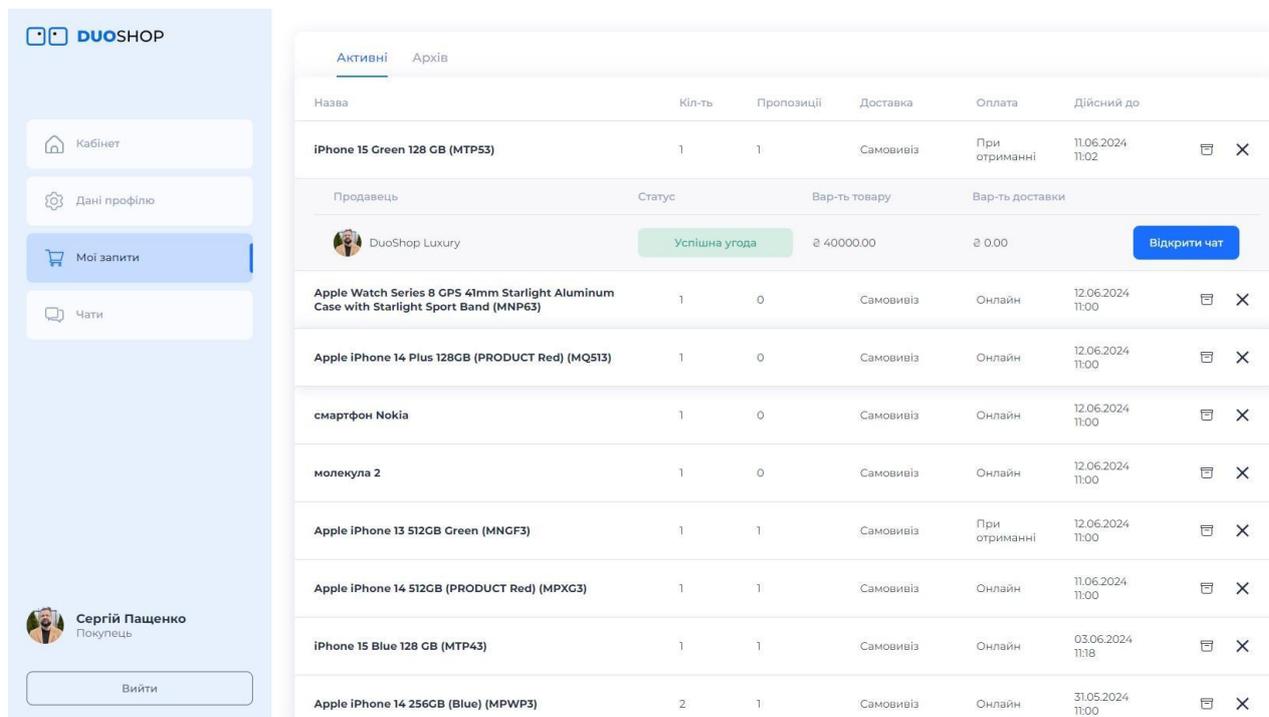


Рис. 1.8. Сторінка «Мої запити»

На рис. 1.9 зображено вкладку «Запити покупців», де продавець може переглядати доступні йому замовлення та надавати тут свої пропозиції.

Статус запиту	Назва	Кіл-ть	Доставка	Оплата	Дійсний до	Моя пропозиція
Пропозиція прийнята	iPhone 15 Pro Max Blue Titanium 256 GB (MU7A3)	1	Самовивіз	Онлайн	26.03.2024 19:06	Товар: € 1.00 Доставка: € 0.00
Успішна угода	iPhone 15 Pro Max Blue Titanium 256 GB (MU7A3)	1	Самовивіз	Онлайн	12.04.2024 11:36	Товар: € 1.00 Доставка: € 0.00
Пропозиція відхилена	Apple AirPods Max Sky Blue (MCVL3)	1	Кур'єрська доставка	При отриманні	12.04.2024 11:51	Товар: € 1.00 Доставка: € 1.00
Пропозиція прийнята	айфон	1	Самовивіз	Онлайн	12.04.2024 12:11	Товар: € 1.00 Доставка: € 0.00
Успішна угода	Apple AirPods Max Sky Blue (MCVL3)	1	Самовивіз	Онлайн	17.04.2024 13:47	Товар: € 1.00 Доставка: € 0.00
Успішна угода	Навушники Apple AirPods 3 (MME73)	1	Самовивіз	Онлайн	25.04.2024 14:58	Товар: € 1.00 Доставка: € 0.00

Рис. 1.9. Сторінка «Запити покупців»

## 6. Чат

На рис. 1.10 зображено вкладку «Чати», де продавець та покупець обговорюють деталі угоди. Після цього покупець підтверджує замовлення та оплачує його.

Рис. 1.10. Чат між покупцем та продавцем

## 1.2. Опис вибраних технологій

Для реалізації проєкту були задіяні наступні технології: Python, Django, FastAPI, Next.js, TypeScript, Docker, ChatGPT.

**Python** - це високорівнева інтерпретована об'єктно-орієнтована мова програмування з суворою динамічною типізацією. Вона приваблива для швидкої розробки завдяки своїм потужним структурам даних, динамічній семантиці та динамічному зв'язуванню, а також підходить для інтеграції існуючих компонентів. Python підтримує модулі та пакети, що сприяє модульності та повторному використанню коду. Інтерпретатор Python і стандартні бібліотеки доступні у скомпільованій та вихідній формі на всіх основних платформах[3].

**Django** – це високорівневий фреймворк з відкритим кодом на Python, призначений для розробки вебсистем. Сайти на Django складаються з однієї або кількох частин, які рекомендується робити модульними, що є однією з основних архітектурних особливостей цього фреймворку. Архітектура Django схожа на модель «Модель-Вигляд-Контролер» (MVC), але з деякими відмінностями в термінології: те, що в MVC називається «контролером», в Django називається «виглядом», а «вигляд» у класичному MVC відповідає «шаблону» в Django. Через це архітектурний підхід в Django називають MTV («Модель-Шаблон-Вигляд»)[4].

**FastAPI** - Це сучасна веб-платформа, вперше випущена у 2018 році, призначена для створення RESTful API на Python. Вона використовується для розробки API з Python 3.8+ на основі стандартних підказок типів. FastAPI побудований на основі Pydantic і використовує підказки типів для перевірки, серіалізації та десеріалізації даних[5].

**Next.js** - це вебфреймворк з відкритим вихідним кодом, який дозволяє створювати веб-додатки на основі React з рендерингом на стороні сервера та

генерацією статичних веб-сайтів. Документація React згадує Next.js серед «рекомендованих інструментів» і радить використовувати його для створення серверних веб-сайтів за допомогою Node.js. Хоча традиційні сайти на React можуть рендерити свій вміст лише на стороні клієнта в браузері, Next.js розширює цю функціональність, додаючи можливість серверної генерації веб-сайтів[7].

**CSS** - це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних[1].

**JavaScript** - це динамічна, об'єктно-орієнтована прототипна мова програмування, що реалізує стандарт ECMAScript. Вона найчастіше використовується для створення сценаріїв вебсторінок, дозволяючи взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, а також змінювати структуру та зовнішній вигляд вебсторінок на боці клієнта[2].

**TypeScript** - позиціонується як інструмент для розробки вебзастосунків, що розширює можливості JavaScript. Він повністю сумісний з JavaScript, і після компіляції програму на TypeScript можна виконувати в будь-якому сучасному браузері або на серверній платформі Node.js[6].

**Docker** - це інструмент для управління ізольованими Linux-контейнерами. Він доповнює інструменти LXC високорівневим API, що дозволяє керувати контейнерами на рівні ізоляції окремих процесів. Docker дозволяє запускати довільні процеси в ізольованому режимі без занепокоєння про вміст контейнера, а також переносити і клонувати ці контейнери на інші сервери, виконуючи всю роботу зі створення, обслуговування і підтримки контейнерів[8].

**ChatGPT** - чат-бот та віртуальний помічник з породжувальним штучним інтелектом.

## РОЗДІЛ II. АНАЛІТИЧНА ЧАСТИНА

### 2.1. Технічне завдання

#### 2.1.1. Головна сторінка сайту

Головна сторінка інформативна. На ній описана вся інформація щодо взаємодії з платформою як для покупців так і для продавців.

#### Сторінка складається з таких блоків:

##### 1. Хедер (Шапка):

Кнопка Live Chat (Help Desk) - Для візуалізації, цей функціонал у прототипі не працює.

Кнопка реєстрації - Функціонал впроваджується. При натисканні відкривається вікно реєстрації Покупця/Продавця.

Кнопка входу в особисті кабінети - Функціонал впроваджується. При натисканні відкривається вікно введення логіна/пароля Покупця/Продавця

##### 2. Верхнє меню:

Опис сервісів платформи - Для візуалізації, цей функціонал на MVP не працює.

Оферта - Для візуалізації, цей функціонал на MVP не працює.

Про компанію - Для візуалізації, цей функціонал на MVP не працює.

Контакти - Для візуалізації, цей функціонал на MVP не працює.

##### 3. Основний контент:

Банер з формою входу в особисті кабінети і реєстрації - Функціонал впроваджується, дублюється функціонал кнопок з Хедера.

Блок з інфографікою переваги використання платформи - Функціонал впроваджується

Блок з відеороликами як користуватися платформою для покупців і для продавців - Функціонал впроваджується

##### 4. Футер (низ сторінки):

Посилання на соціальні мережі, посилання на сторінку "Умови та положення", "Політика конфіденційності", "Умови використання сайту", "FAQ", посилання на сторінку "Контакти" - Для візуалізації, цей функціонал у прототипі не працює.

### **2.1.2. Форма реєстрації**

Вікно складається з 2 закладок: реєстрація покупця і реєстрація продавця - Функціонал впроваджується.

Прив'язка акаунтів через номер телефону і підтвердження через введення коду з смс - Функціонал впроваджується.

Обов'язкові поля для Покупців: ПІБ, пароль доступу, дата народження, номер телефону, адреса електронної пошти, населений пункт, область, регіон, країна - Функціонал впроваджується.

Обов'язкові поля для Продавців: Найменування Юр. особи, ЄДРПОУ (номер у реєстрі), номер телефону директора, номер телефону менеджера, пароль доступу, адреса електронної пошти, населений пункт, область, регіон, країна - Функціонал впроваджується.

### **2.1.3. Вимоги до особистих кабінетів**

Особисті кабінети призначені для обміну інформацією (за принципом заявка-відповідь на заявку) між учасниками (Покупцями і Продавцями)

При появі в кабінеті одного з учасників нової відповіді, повідомлення, з'являється візуальне повідомлення в тій закладці, в якій є якесь нове повідомлення. Щойно нову інформацію прочитано/отримано, візуальне повідомлення зникає.

Вся інформація в закладках з'являється/зникає в режимі реального часу без перезавантаження сторінки.

## **2.1.4. Особистий кабінет покупця**

### **2.1.4.1. Загальний опис**

Після входу шляхом введення логіна і пароля, покупець потрапляє в особистий кабінет, який складається із закладок: Кабінет, Персональна інформація, Мої запити, Чати.

### **2.1.4.2. Закладка "Кабінет"**

Це закладка для подання заявок на купівлю шуканих товарів. Відкривається за замовчуванням. В ній є поле для введення запиту на покупку. Наприклад, покупець у поле пошуку вводить запит "Холодильник Bosch KGN39XW326", він визначився, що хоче купити саме цей холодильник, і йому потрібно отримати цінові пропозиції від продавців щодо цього холодильника.

Система платформи повинна за допомогою API семантичного розбору запиту має зрозуміти, що покупець шукає "Холодильник" і відіслати запит продавцям, які підписані на отримання заявок на категорію "Побутова техніка". API семантичного розбору має розпізнавати якщо вводять інформацію латиницею, кирилицею, кирилицею на латиниці.

Важливо: Під полем запиту, після семантичного розбору відображаються розпізнані категорії, до яких належить запит. Якщо запит належить до кількох категорій, то від покупця буде вимагатися вибір категорії, до якої належить запит, шляхом натискання на неї, без вибору заявка не відправиться до продавців.

Створення заявки. Покупець заповнює форму, яка аналогічна форми "Rozetka":

1. Ваші контактні дані - контактні дані покупця беруться із заповнених даних під час реєстрації. Дивитись рис. 2.1.

## 1 Ваши контактные данные

Фамилия	Имя
<input type="text" value="Сергеев"/>	<input type="text" value="Сергей"/>
Мобильный телефон	Электронная почта
<input type="text" value="+38 063 111 11 11"/>	<input type="text" value="sergeev@i.ua"/>

Рис. 2.1. Форма замовника товару

2. Поле товару, у нашому випадку запит товару на купівлю - у полі прописаний той товар, який покупець хоче купити. Форма редагується, покупець може змінити кількість. Ціна та вартість поки недоступні і розрахунок не ведеться, суму пропише продавець у цю форму, коли отримає заявку. Дивитись рис. 2.2.

1 Товары продавца Rozetka		<a href="#">Редактировать</a>	
	Цена	Количество	Сумма
 Холодильник SAMSUNG RB38T603FSA/JA	25 199 ₴	1	25 199 ₴

Рис. 2.2. Товари подавця

3. Доставка. Дуже бажано впровадити як на Розетці. Дивитись рис. 2.3, 2.4, 2.5.

## 2 Доставка

Ваш город

 Харьков

Харьковская обл.

>

Рис. 2.3. Вибір населеного пункту для доставки

Выберите свой город

Доставляем заказы по всей Украине!

Киев Харьков Одесса

Днепр Запорожье Львов

Введите населенный пункт Украины

Харьков

Например, Котюжины

Перейти на главную страницу Применить

Выбор города поможет предоставить актуальную информацию о наличии товара, его цены и способов доставки в вашем городе! Это поможет сохранить больше свободного времени для вас!

Рис. 2.4. Вибір населеного пункту для доставки

Самовывоз из наших магазинов Бесплатно

Курьер по вашему адресу Бесплатно  
 Доставка 22 марта с 98 ₴

Самовывоз из Новой Почты 449 ₴

Рис. 2.5. Вибір способу доставки

4. Оплата. Дуже бажано впровадити як на Розетці. Дивитись рис. 2.6.

## 3 Оплата

- Оплата при получении товара
- Оплатить сейчас
- Безналичными для юридических лиц  
Оплата для юридических лиц через расчетный счет
- Оплатить онлайн социальной картой "Паунок малюка"
- Безналичными для физических лиц  
Оплата для физических лиц через расчетный счет
- Кредит и оплата частями  
Оформление кредитов в банках партнеров

Рис. 2.6. Вибір способу оплати

5. Контактні дані одержувача замовлення. Якщо одержувач замовлення відрізняється від замовника, то тут потрібно прописати дані одержувача, якщо не відрізняється, то залишаються введені дані під час реєстрації. Дивитись рис. 2.7.

## 4 Контактные данные получателя заказа

Внимание! Получение заказа по паспорту. Введите фамилию, имя, отчество и мобильный номер телефона получателя заказа

Другой получатель ▼

Фамилия	Имя
<input type="text"/>	<input type="text"/>
Отчество	Мобильный телефон
<input type="text"/>	<input type="text" value="+38"/>

Рис. 2.7. Форма отримувача товару

Після натискання кнопки "Надіслати заявку" заявка відправляється в кабінети продавців. Продавці не бачать контактні дані покупця в заявці, тільки заповнені дані про товар/послугу та додаткові умови. Заявки отримують тільки ті продавці, які оплатили отримання заявок на дану категорію товарів/послуг. Якщо відправлено конкретний запит на покупку "Холодильник Bosch", то заявки отримують тільки ті продавці, які оплатили отримання заявок на категорію "Побутова техніка".

### 2.1.4.3. Закладка "Мої запити"

Відображає створені покупцем запити та пропозиції на них від продавців. Пропозиції від продавців виводяться одна за одною за часом приходу. Дивитись рис. 2.8.

Важливо! Пропозиції повинні приходити в реальному часі без перезавантаження сторінки.

№ 752 997 665 от 25.11.22 Выполнен	Сумма заказа 273 ₴	 ▾
№ 747 822 573 от 07.10.22 Выполнен	Сумма заказа 349 ₴	 ▾
№ 746 073 970 от 17.09.22 Выполнен	Сумма заказа 459 ₴	 ▾
№ 743 964 203 от 23.08.22 Выполнен	Сумма заказа 289 ₴	 ▾
№ 743 963 902 от 23.08.22 Выполнен	Сумма заказа 998 ₴	 ▾
№ 742 803 790 от 08.08.22 Выполнен	Сумма заказа 12 999 ₴	 ▾
№ 742 752 619 от 08.08.22 Выполнен	Сумма заказа 160 ₴	 ▾
№ 741 073 047 от 18.07.22 Выполнен	Сумма заказа 99 ₴	 ▾
№ 738 385 462 от 10.06.22 Выполнен	Сумма заказа 3 595 ₴	 ▾
№ 733 755 402 от 09.03.22 Отменен	Сумма заказа 1 492 ₴	 ▾

Рис. 2.8. Відображення запитів

1. Відповідь на запит має такий вигляд: Рядок таблиці з колонками (Найменування продавця, Рейтинг у системі, найменування

товару/послуги, вартість, яку запропонував продавець без урахування доставки, загальна вартість замовлення з доставкою.

- У таблиці відповідей для зручності користування можна вибудувати ранжування за всіма колонками для зручності відбору відповідей, що надходять на заявки.
- Кожна відповідь клікабельна, під час натискання відкривається вікно, в якому видно повністю відповідь продавця за заявкою з тими даними, які покупець заповнив у заявці. Дивитись рис. 2.9.

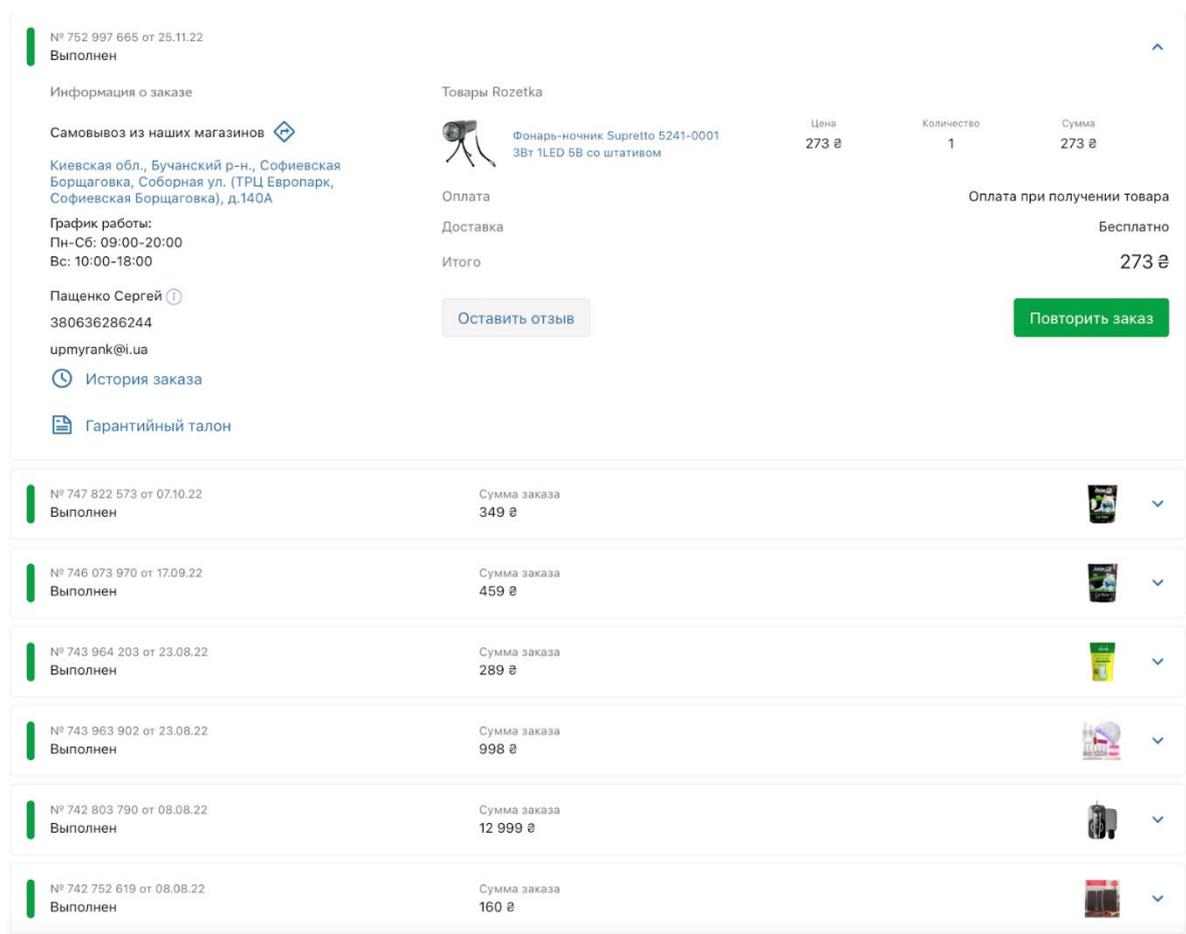


Рис. 2.9. Взаємодія з відповідями

- Вікно відповіді заявки містить:
  - Всю заповнену інформацію, яку покупець заповнював під час відправлення заявки, продавець тільки прописав вартість товару і вартість доставки.

- Вартість товару яку продавець прописав вручну, вартість автоматично перераховується (множиться) на кількість товару і виводиться загальна вартість
- Кнопка "Вибрати продавця"

Натискання кнопки "Обрати продавця" підтверджує факт вибору Продавця і його пропозиції, всі інші відповіді відхиляються автоматично і зникають зі сторінки покупця, на сторінці історії відповідей продавців набувають статусу "Відхилена".

**ВАЖЛИВО!!!** До того моменту, поки покупець не вибрав продавця, для продавця покупець нонейм (не видно ні ПІБ, ні контактних даних), щойно вибрав продавця, у формі відображаються всі контактні дані покупця, і він уже не нонейм.

Покупець після вибору продавця може просто натиснути кнопку "Оформити угоду", а може підібрати ще додаткові товари до основного або обговорити всі нюанси угоди в особистому сокетному чаті (без перезавантаження сторінки). Вікно сокетного чату вбудовується в саму заявку для можливості спілкування між продавцем і покупцем у рамці цієї заявки.

#### **2.1.4.4. Закладка "Дані профілю"**

Мають бути такі можливості:

1. Зміна аватарки.
2. Заповнення особистої інформації про місце проживання, ПІБ, електронної пошти з можливістю змінювати інформацію.
3. Зміна пароля.
4. Зміна номеру телефону.
5. Зміна електронної пошти.

## **2.1.5. Особистий кабінет продавця**

### **2.1.5.1. Загальний опис**

Після входу шляхом введення логіна і пароля, Продавець потрапляє в особистий кабінет, який складається із закладок: Кабінет, Дані профілю, Запити покупців, Чати.

### **2.1.5.2. Закладка «Кабінет»**

Тут має відображатися статистика угод, а саме: Кількість виконаних замовлень, Сума виконаних замовлень, Кількість відхилених замовлень.

### **2.1.5.3. Закладка «Дані профілю»**

Мають бути такі можливості:

1. Зміна аватарки.
2. Вибір/додавання/зміна категорій за якими продавець буде отримувати заявки від покупців.

Наприклад, Продавець вирішив отримувати заявки за категоріями "Холодильники". Він вибирає зі списку категорію "Побутова техніка".

Приклад списку категорій:

- Авто і Мото
- Побутова техніка
- Все для дому
- Зоотовари
- Інструменти
- Комп'ютери, Мережі
- Краса і здоров'я
- Музичні інструменти
- Офіс
- Сантехніка, Будматеріали

- Смартфони, Телефони
- Телевізор, Аудіо, Відео, Фото
- Товари для дорослих
- Товари для геймерів
- Туризм, Риболовля

3. Зміна пароля

4. Зміна номеру телефону.

5. Зміна електронної пошти.

#### 2.1.5.4. Закладка «Запити покупців»

Це сторінка вхідних заявок на купівлю шуканих товарів/послуг. Відкривається за замовчуванням. Заявки, що надходять, ранжуються за часом надходження. Поки продавець не натиснув на неї і не відкрив, на ній відображається статус "Нова", щойно відкрив, статус "Нова" зникає. Кожна заявка клікабельна, під час натискання на неї заявка відкривається за принципом гармошки і можна побачити форму, яку заповнив покупець, без ціни на товар і без ціни на доставку. Дивитись рис. 2.10.

№ заявки	Дата	Статус	Сумма заказа
№ 752 997 665	от 25.11.22	Выполнен	273 ₴
№ 747 822 573	от 07.10.22	Выполнен	349 ₴
№ 746 073 970	от 17.09.22	Выполнен	459 ₴
№ 743 964 203	от 23.08.22	Выполнен	289 ₴
№ 743 963 902	от 23.08.22	Выполнен	998 ₴
№ 742 803 790	от 08.08.22	Выполнен	12 999 ₴

**Деталі заявки № 752 997 665:**

- Статус:** Выполнен
- Товари Rozetka:** Фонарь-ночник Suprette 5241-0001 3Вт 1LED 5В со штативом (Ціна: 273 ₴, Кількість: 1, Сума: 273 ₴)
- Оплата:** Оплата при получении товара (Бесплатно)
- Доставка:** Бесплатно
- Итого:** 273 ₴
- Дії:** Оставить отзыв, Повторить заказ

**Інформація про замовлення:**

- Самовывоз из наших магазинов:** Киевская обл., Бучанский р-н., Софиевская Борщаговка, Соборная ул. (ТРЦ Европарк, Софиевская Борщаговка), д.140А
- График работы:** Пн-Сб: 09:00-20:00, Вс: 10:00-18:00
- Контактна особа:** Пащенко Сергей (380636286244, urtugank@i.ua)
- Історія замовлення:** Істория заказа
- Гарантийний талон:** Гарантийный талон

Рис. 2.10. Взаємодія з відповідями

Наприклад, надійшла заявка на "Холодильник Bosch KGN39XW326". Продавець відкриває заявку, дивиться у себе за наявністю, якщо товар є, то прописує вартість товару за одиницю і прописує вартість доставки тим способом доставки, який вибрав покупець. Автоматично все підсумовується і виводиться загальна сума з урахуванням доставки. Продавець не бачить контактні дані продавця, поки покупець його не вибере. Натискає кнопку "Відповісти покупцеві".

Після заповнення відповіді на заявку і натисканні на кнопку відповіді, відповідь приходить до покупця в кабінет для подальшої взаємодії.

Якщо товару немає і продавець не може продати товар, то продавець натискає кнопку "Відхилити заявку" і заявка пропадає і з'являється на сторінці архіву заявок зі статусом "Відхилена".

Після того як покупець вибрав продавця, на сторінці заявки у продавця відображається статус заявки "Обраний покупцем", і тепер продавець може бачити контактні дані покупця, щоб отримати оплату або домовитися про продаж аксесуарів або додаткових товарів, а також закрити угоду.

Активується сокетний чат у цій заявці, в якому продавець і покупець зможуть домовитися про додаткові продажі або ще про щось.

Після того, як покупець вибрав продавця, з'являється можливість продавцю правити заявку, змінювати кількість товару, додавати нові позиції супутніх товарів у картці замовлення, які покупець хоче ще купити додатково.

Коли покупець здійснив оплату і отримав товар від продавця, покупець або продавець натискає кнопку "Угоду завершено" і закриває угоду, у продавця і покупця замовлення переходить у "Виконана заявка".

## 1.2. Порівняння з аналогами

Основними аналогами тут виступають Hotline (рис. 1.11) та eKatalog (рис. 1.12).

The screenshot shows the Hotline website interface for the Apple iPhone 15 Pro Max. At the top, there are navigation icons and a search bar. The main product image is on the left, with a detailed view of the phone's camera and back. To the right, the price range is shown as 48,058 to 67,200 UAH, with a 'ПОРІВНЯТИ' (Compare) button. Below this, there are links for 'Повідомити про зміну ціни, нові акції' and 'Дивитись схожі товари'. The 'Характеристики' (Specifications) section lists details like screen size (6.7"), processor (Apple A17 Pro), and storage options. The 'Опис' (Description) section provides more context about the phone's position in the market. At the bottom, there are tabs for 'ПРО ТОВАР', 'ДЕ КУПИТИ (130)', 'ДЕ ЗАБРАТИ (81)', 'ВІДГУКИ (6)', and 'КУПИТЬОМ РАЗОМ'. A filter sidebar on the left allows users to refine their search by delivery location and other criteria. The main content area shows a list of offers from retailers like 'Ябло' and 'Apple room', each with a price, warranty, and a 'КУПИТИ' button.

Рис. 1.11. Hotline

The screenshot shows the eKatalog website interface for the Apple iPhone 15 Pro Max. The top navigation bar includes the eKatalog logo, a search bar, and various filters. The main content area is titled 'Ціни на Apple iPhone 15 Pro Max 256 Гб'. It features a product image, a price range of 46,599 to 61,999 UAH, and a 'ДИНАМІКА ЦІН' (Price Dynamics) chart. Below this, there are tabs for 'ДЕ КУПИТИ (10)', 'ОПИС', 'ХАРАКТЕРИСТИКИ', 'ВІДГУКИ 6', 'ОБГОВОРЕННЯ', and 'АКСЕСУАРИ 10+'. The 'ХАРАКТЕРИСТИКИ' tab is active, showing detailed technical specifications. The 'ДЕ КУПИТИ' tab shows a list of offers from retailers like 'Ябло', 'PLATFORMA', and 'GROKHOLSKY', each with a price, warranty, and a 'Перейти в магазин' button. A sidebar on the right allows users to filter offers by brand and model. The bottom of the page has a navigation bar with 'ПОРІВНЯННЯ', 'ЗАКЛАДКИ', and 'ІСТОРІЯ 12'.

Рис. 1.12. eKatalog

Кожен з аналогів має як і свої плюси, так і мінуси. Детальніше описано у табл. 1.1.

Табл. 1.1. Порівняння з конкурентами

	DuoShop	Hotline	eKatalog
Можливість відфільтрувати продавців по способу доставки	+	-	-
Можливість поспілкуватися з продавцем не покидаючи сайт	+	-	-
Завжди актуальні ціни та наявність	+	-	-
Можливість переглянути динаміку цін	-	+	+
Можливість побачити відгуки на продавців	-	+	+
Можливість порівняти товари за характеристиками	-	+	+

На мою думку на ринку є досить потужні аналоги, але їх ключова проблема в тому, що ціни та інформація про наявність товару там не оновлюється в реальному часі. З цього випливає проблема, що обравши продавця, з часом, може з'ясуватися, що товар відсутній і треба шукати наступного.

Наявність можливості залишати та переглядати відгуки це прекрасний спосіб зрозуміти що за продавець перед вами. Розроблювана система не має системи відгуків, але це поки що.

Також можливість порівнювати ціни з колишніми для мене теж вважається плюсом, адже таким чином можна приблизно спрогнозувати коли буде дешевшою наступна модель товару. На рис. 1.13 зображений функціонал перегляду динаміки цін на платформі eKatalog.

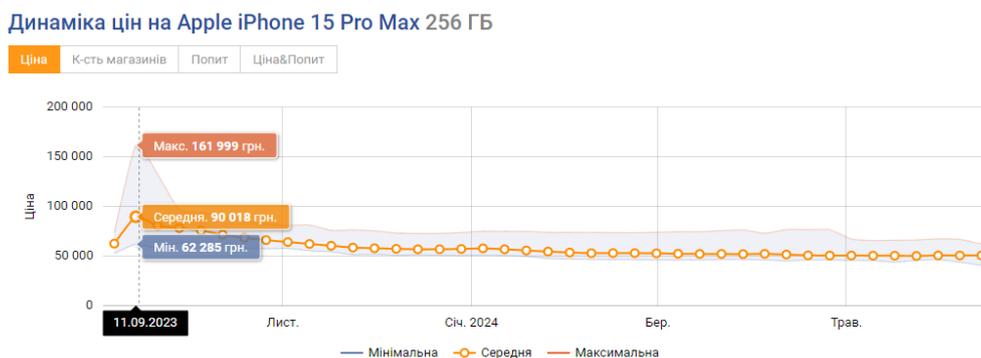


Рис. 1.13. Динаміка цін на eKatalog

### 2.3. Алгоритми та БД

Для визначення категорії лише за назвою товару я інтегрував ChatGPT використовуючи його API[9] у свою систему. Вона відправляє ШІ список усіх категорій, які адміністратор сайту вказав у адмін-панелі, та назву товару. Дивитись рис. 2.11.

```

2 usages  ± XFaost+1 *
@CriticalExceptLogger(chat_gpt_logger)
def get_product_category_ids_by_text(text: str, product_categories: List[ProductCategory]) -> List[str]:
    if not text:
        return []
    if DEBUG:
        return [product_categories[0].id]

    chat_gpt_logger.info(f'text: {text}')
    openai.api_key = get_chat_gpt_api_key()

    content = f'Є список категорій: [\n'
    for p in product_categories:
        p: ProductCategory
        if not p.has_children():
            content += f'{{ id: {p.id}, name: "{p.name}" }}\n'
    content += ']\n'
    content += f'Якому ID категорії належить товар "{text}"? ' \
        'Дай коротку відповідь, без уточнень та лишніх символів. ' \
        'ID відокремлюй одне від одного пробілами. Приклад відповіді: "12 72 94". ' \
        'Якщо не можливо визначити ID категорії за наданими даними, то дай мені таку відповідь: "0".'

    completion = openai.ChatCompletion.create(
        model='gpt-3.5-turbo',
        messages=[{
            'role': 'user',
            'content': content,
        }]
    )
    response = completion.choices[0].message.content

    chat_gpt_logger.info(f'response: {response}')

    if response == '0':
        result = []
    else:
        result = keep_numbers_and_spaces(response).split(' ')

    chat_gpt_logger.info(f'result: {result} {type(result)}')
    return result

```

Рис. 2.11. Функція get\_product\_category\_ids\_by\_text

Коли створюється заявка покупцем, то її треба надіслати тим продавцям, які продають товари за категорією, яку визначив ChatGPT по назві товару, який вказав покупець. Алгоритм на рис. 2.12 вирішує цю задачу.

```

@celery_app.task()
def create_requests_for_sellers(buyer_request_id: int):
    from orders.serializers.request_for_seller import RequestForSellerPreviewSerializer

    buyer_request = get_buyer_request_by_id_or_404(buyer_request_id)

    seller_ids = get_seller_ids_for_buyer_request(buyer_request)

    channel_layer = get_channel_layer()

    for seller_id in seller_ids:
        seller = get_seller_by_id_or_404(seller_id)
        request_for_seller = request_for_seller_create(
            buyer_request=buyer_request,
            seller=seller
        )
        async_to_sync(channel_layer.group_send)(
            f'seller_{seller.id}', {
                'type': 'new_request',
                'data': RequestForSellerPreviewSerializer(request_for_seller).data
            }
        )

    url = f'{get_website_url()}/dashboard/requests/active'
    email_text = (f'Вам надійшла заявка {buyer_request.product_name},'
                 f' {buyer_request.product_quantity} одиниць.'
                 f' Детальніше за посиланням: {url}')
    email_html_text = (f'Вам надійшла заявка {buyer_request.product_name},'
                      f' {buyer_request.product_quantity} одиниць.'
                      f' <a href="{url}">Детальніше тут</a>')

    send_mail_from_default_email(
        subject='Нова заявка від покупця!',
        message=email_text,
        html_message=email_html_text,
        recipient=seller.user.email
    )

```

Рис. 2.12. Функція create\_requests\_for\_sellers

Періодично треба перевіряти чи прийшли на рахунок гроші від якогось продавця. Реалізую це за допомогою АРІ ПриватБанку[10]. Дивитись рис. 2.13.

```

2 usages  ± XFaost
@CriticalExceptLogger(privatbank_logger)
def check_privat_bank_acc():
    core_settings = get_core_settings()
    pay_settings = get_pay_settings()

    if pay_settings.last_transaction_date:
        start_date = pay_settings.last_transaction_date
    else:
        start_date = date_now()

    start_date_str = date_to_str(start_date, join: '-')
    url = f'https://acp.privatbank.ua/api/statements/transactions?acc={core_settings.iban}&startDate={start_date_str}'

    follow_id = None

    while True:
        full_url = url
        if follow_id:
            full_url += f'&followId={follow_id}'

        r = requests.get(full_url, headers=get_header(core_settings))
        data = r.json()
        if r.status_code != 200:
            raise PrivatBankTransactionsBadRequestException(r.status_code, data)

        for tr in data['transactions']:
            if tr['TRANTYPE'] != 'C':
                continue

            transaction = Transaction(
                tr['REF'],
                tr['CCY'],
                tr['OSND'],
                tr['AUT_CNTR_ACC'],
                tr['AUT_CNTR_CRF'],
                tr['AUT_CNTR_NAM'],
                tr['SUM']
            )

            try:
                transaction.check()
            except (
                TransactionInvalidCurrencyException,
                TransactionInvalidPurposeException,
                TransactionInvalidCheckoutNumException,
                TransactionRepeatException,
                TransactionInvalidSumException,
                TransactionInvalidEDRPOUException,
                TransactionInvalidSellerException,
                TransactionOldException
            ) as error:
                privatbank_logger.info(error)
                continue

            seller_access_update(
                transaction.sellers_access_checkout.seller,
                transaction.ref
            )

        if not data['exist_next_page']:
            if data['transactions']:
                last_transaction = data['transactions'][-1]
                last_transaction_ref: str = last_transaction['REF']
                last_transaction_date = str_to_date(last_transaction['DATE_TIME_DAT_OD_TIM_P'].split()[0])
                update_last_transaction(pay_settings, last_transaction_ref, last_transaction_date)
            break

        follow_id = data['next_page_id']

```

Рис. 2.13. Функція check\_privat\_bank\_acc

На рис. 2.14 зображена схема бази даних.

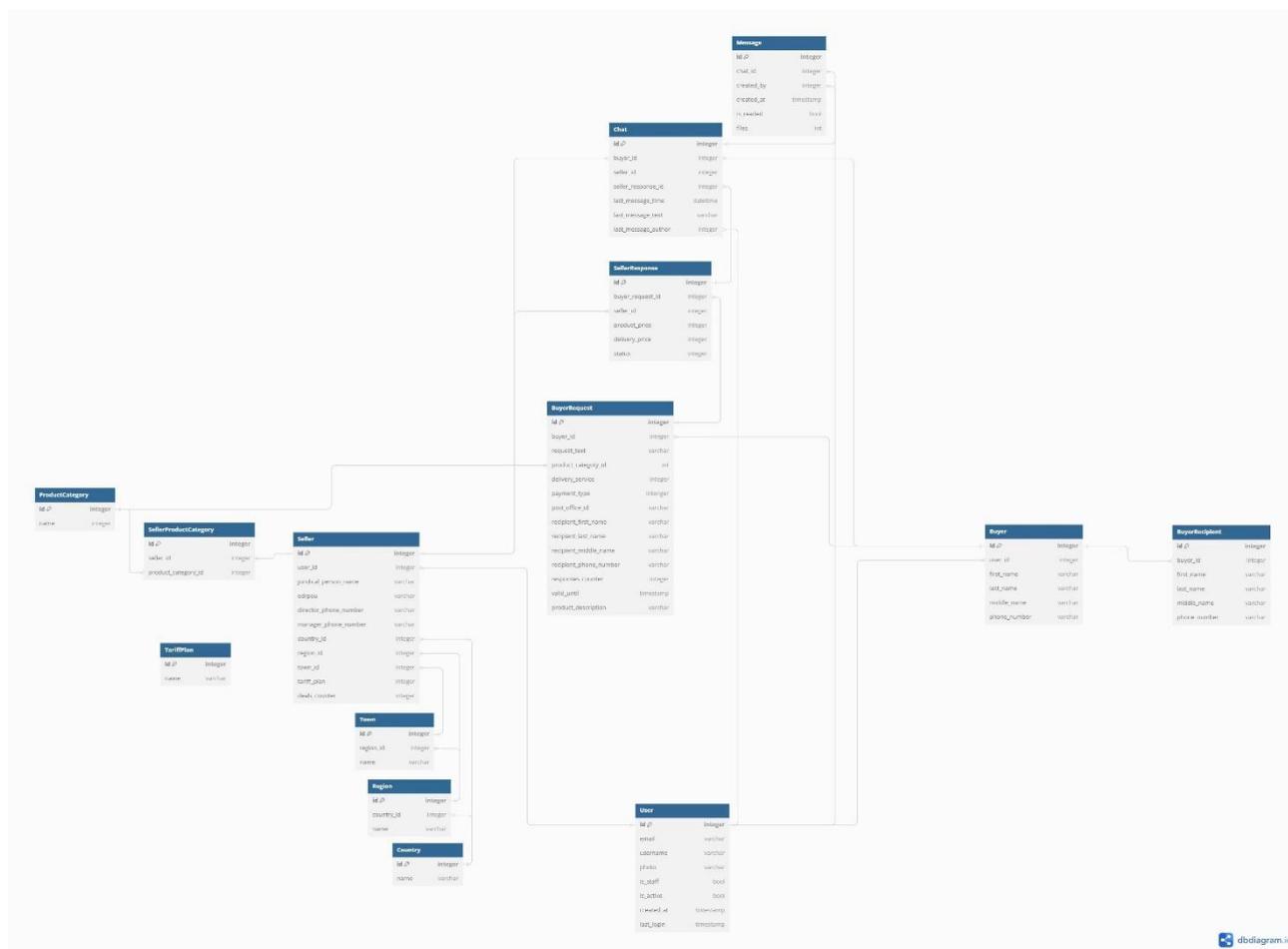


Рис. 2.14. Схема бази даних

## РОЗДІЛ ІІІ. ПРИКЛАДНА ЧАСТИНА

### 3.1. Контейнеризація системи

Контейнеризація є важливим аспектом сучасної розробки програмного забезпечення, що дозволяє ізолювати окремі компоненти системи в контейнери, забезпечуючи їхню незалежність та легкість керування. В даному проєкті використовується Docker Compose для управління контейнерами, що дозволяє створити та підтримувати багатокomпонентну архітектуру системи.

У проєкті використовується файл `docker-compose.yml` (Додаток А) версії 3.8, який описує сервіси, необхідні для функціонування системи. У таблиці 3.1. наведено структуру цього файлу та короткий опис кожного сервісу.

Табл. 3.1. Опис контейнерів

redis	Служить як брокер повідомлень для Celery
mongo_db	Нереляційна база даних для зберігання усіх відділень Нової Пошти
delivery_manager	Робить один запит на день до API Нової Пошти, який у відповідь повертає усі актуальні відділення сервісу доставки
delivery_manager_celery	Служить для планування наступних запитів до API Нової Пошти
postgres_db	Реляційна база даних для сайту. У ній зберігаються усі продавці, покупці, чати, запити покупців, пропозиції продавців, категорії товарів...
es (elasticsearch)	Служить для швидкого та розумного пошуку товару за назвою
backend	Найголовніший сервіс
backend_celery	Служить для планування завдань. Наприклад, кожен день робити запити до НБУ для актуалізації курсу долара
nginx	Це HTTP-сервер який тут запрограмований, щоб коректно розподіляти запити між різними сервісами

frontend	На ньому тримається візуальна частину сайту, яку бачить перед собою користувач
----------	--

### Переваги контейнеризації

- Кожен сервіс працює в окремому контейнері, що запобігає конфліктам залежностей та забезпечує стабільність системи.
- Легко масштабувати сервіси шляхом збільшення кількості контейнерів для окремих компонентів системи.
- Контейнери забезпечують однакове середовище для розробки, тестування та продакшн, що зменшує ризики непередбачених помилок.
- Docker Compose дозволяє керувати всіма контейнерами з одного файлу конфігурації, що спрощує розгортання та обслуговування системи.

Контейнеризація є ключовою технологією для створення сучасних веб-додатків, забезпечуючи їхню надійність, масштабованість та зручність у розгортанні. Використання Docker Compose в даному проєкті дозволяє легко керувати всіма компонентами системи, що складається з баз даних, бекенду, фронтенду та інших сервісів, необхідних для повноцінного функціонування маркетплейсу.

### 3.2. Інструкція з користування

Спочатку користувач має зареєструватися на сайті. Або як покупець, або як продавець. Відповідні форми зображені на рис. 1 та рис. 2. Користувачу прийде на електронну пошту лист для підтвердження реєстрації.

Далі користувач може авторизуватися ввівши електронну пошту та пароль. Це зображено на рис. 3. Після цього користувача перекине на вкладку «Кабінет». Якщо користувач авторизувався як покупець, то у нього кабінет буде виглядати як на рис. 6; якщо як продавець, то як на рис. 7.

Продавцю необхідно перейти на вкладку «Дані профілю», що на рис. 3.1, для вказання категорій товарів, які він продає та оплатити доступ.

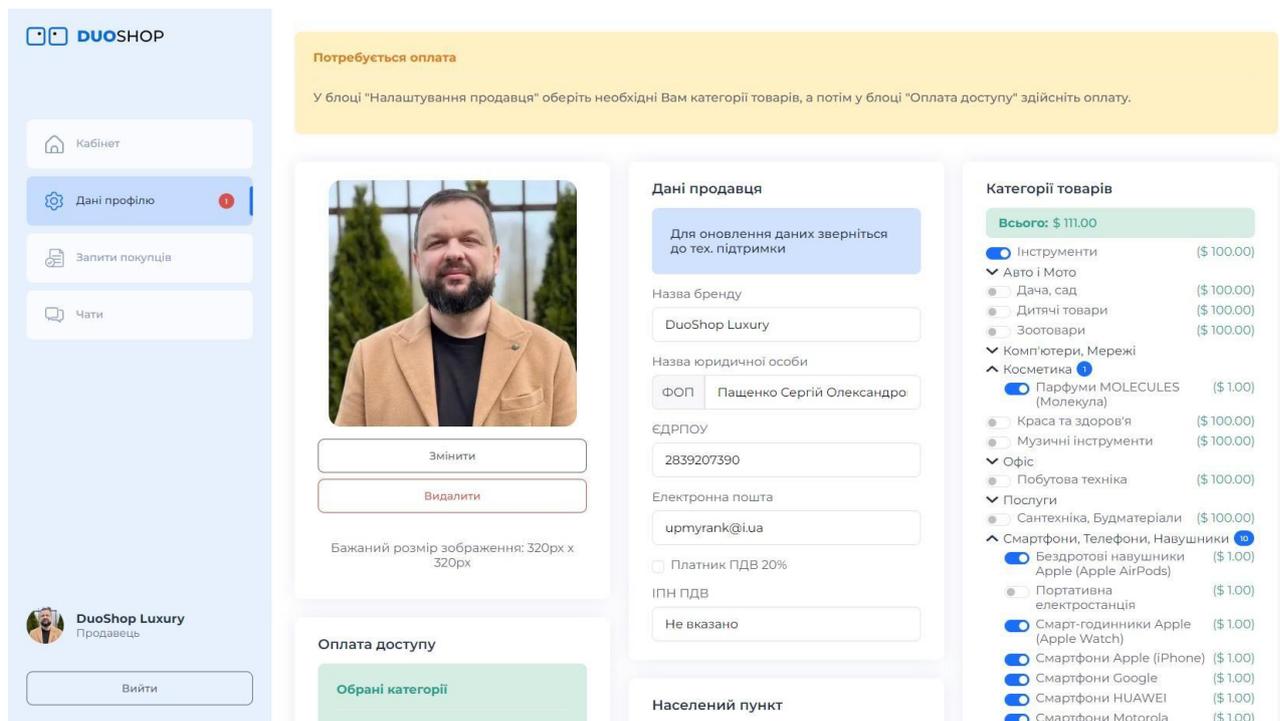


Рис. 3.1. Демонстрація вкладки «Дані профілю» у продавця

Коли покупець хоче оформити замовлення, йому треба перейти на вкладку «Кабінет» (рис. 3.2), заповнити усі дані та натиснути кнопку «Оформити замовлення».

## Створення замовлення

**Товар**

Назва товару: Apple iPhone 13 256Gb Midni; Кількість: 1

Смартфони Apple (iPhone)

Побажання до замовлення: новий

Підтвердити

**Доставка**

Оберіть способи доставки, якими би ви хотіли отримати обраний товар або послугу

- Нова Пошта
- Самовивіз
- Кур'єрська доставка

**Оплата**

Оберіть способи оплати, якими би ви хотіли оплатити обраний товар або послугу

- Онлайн
- При отриманні

**Дані отримувача**

Увага! Отримання замовлення за паспортом. Введіть прізвище, ім'я, по батькові як зазначено у документі та мобільний номер телефону отримувача замовлення

Пащенко Сергій Олександрович

Прізвище: Пащенко; Ім'я: Сергій

По батькові: ; Номер телефону:

**Пошук продавця**

Оберіть межі пошуку обраного товару або послуги

- Вся країна
- В межах регіону
- В межах населеного пункту

Рис. 3.2. Заповнена форма для створення замовлення

Якщо назва товару буде введена не валідно, то виведеться помилка на рис.

3.3.

## Створення замовлення

**Товар**

Назва товару: аааааааааа; Кількість: 1

Будь-ласка, вкажіть назву конкретніше

Побажання до замовлення: новий

Підтвердити

Рис. 3.3. Демонстрація помилки при вводиті назви товару

Важливо вводити назву товару коректно, адже система на основі назви визначає його категорію.

Створивши замовлення, покупець може його переглянути у вкладці «Мої запити». Дивитись рис. 3.4.

Назва	Кіл-ть	Пропозиції	Доставка	Оплата	Дійсний до		
iPhone 15 Green 128 GB (MTP53)	1	1	Самовивіз	При отриманні	11.06.2024 11:02	🗑️	✕
Apple Watch Series 8 GPS 41mm Starlight Aluminum Case with Starlight Sport Band (MNP63)	1	0	Самовивіз	Онлайн	12.06.2024 11:00	🗑️	✕
Apple iPhone 14 Plus 128GB (PRODUCT Red) (MQ513)	1	0	Самовивіз	Онлайн	12.06.2024 11:00	🗑️	✕
смартфон Nokia	1	0	Самовивіз	Онлайн	12.06.2024 11:00	🗑️	✕
молекула 2	1	0	Самовивіз	Онлайн	12.06.2024 11:00	🗑️	✕
Apple iPhone 13 512GB Green (MNGF3)	1	1	Самовивіз	При отриманні	12.06.2024 11:00	🗑️	✕
Apple iPhone 14 512GB (PRODUCT Red) (MPXG3)	1	1	Самовивіз	Онлайн	11.06.2024 11:00	🗑️	✕
iPhone 15 Blue 128 GB (MTP43)	1	1	Самовивіз	Онлайн	03.06.2024 11:18	🗑️	✕
Apple iPhone 14 256GB (Blue) (MPWP3)	2	1	Самовивіз	Онлайн	31.05.2024 11:00	🗑️	✕

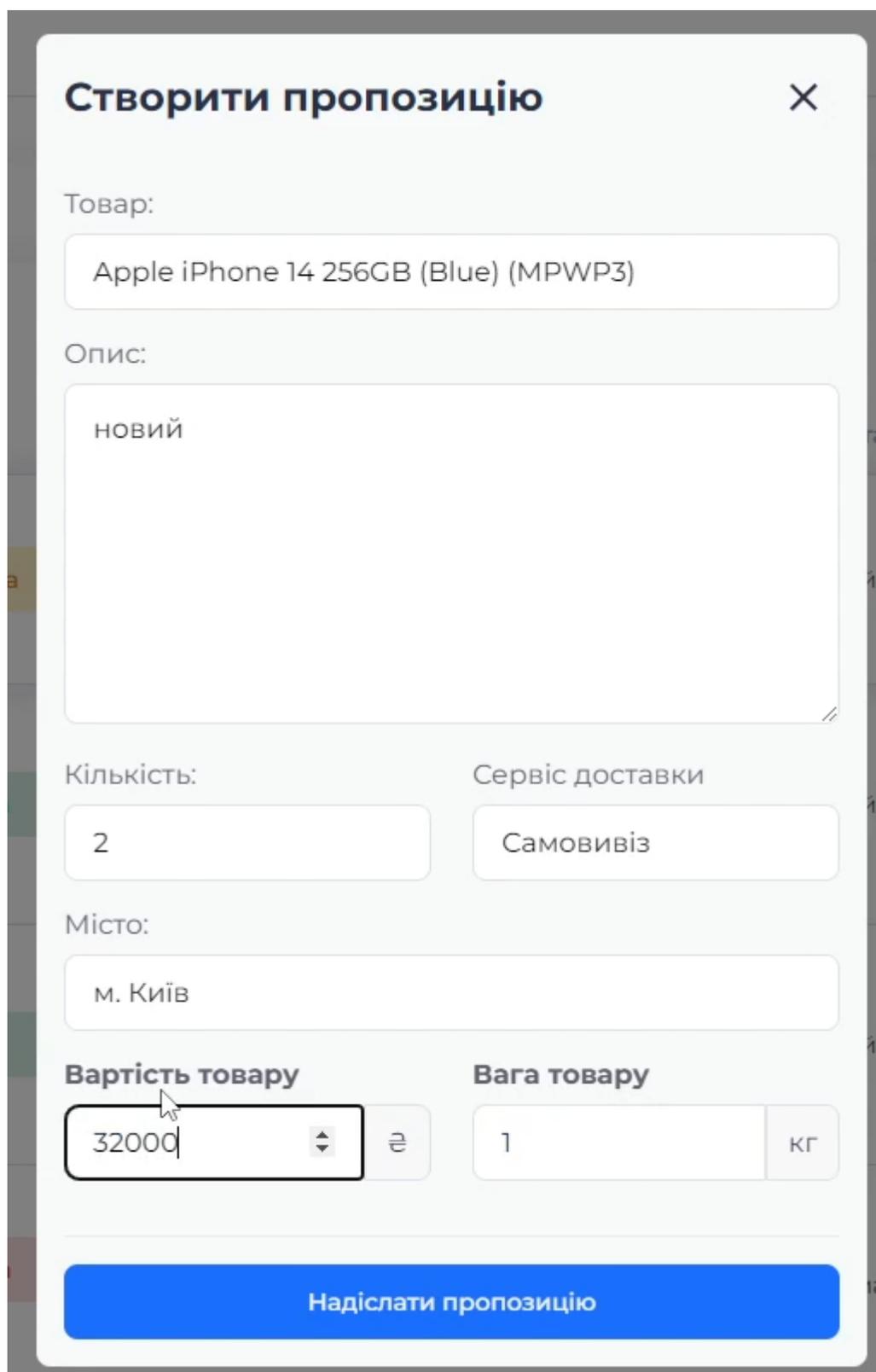
Рис. 3.4. Демонстрація вкладки «Мої запити»

Також замовлення буде доступне продавцям, які продають товари за категорією, яка визначила система. Дивитись рис. 3.5.

Статус запиту	Назва	Кіл-ть	Доставка	Оплата	Дійсний до	Моя пропозиція	
Відповідь вами не дана	Apple iPhone 14 256GB (Blue) (MPWP3)	2	Самовивіз	Онлайн	31.05.2024 11:00	Запропонувати	Відкинути
Пропозиція прийнята	iPhone 15 Pro Max Blue Titanium 256 GB (MU7A3)	1	Самовивіз	Онлайн	26.03.2024 19:06	Товар: € 1.00 Доставка: € 0.00	🗑️
Успішна угода	iPhone 15 Pro Max Blue Titanium 256 GB (MU7A3)	1	Самовивіз	Онлайн	12.04.2024 11:36	Товар: € 1.00 Доставка: € 0.00	🗑️
Пропозиція відхилена	Apple AirPods Max Sky Blue (MGYL3)	1	Кур'єрська доставка	При отриманні	12.04.2024 11:51	Товар: € 1.00 Доставка: € 1.00	🗑️
Пропозиція прийнята	айфон	1	Самовивіз	Онлайн	12.04.2024 12:11	Товар: € 1.00 Доставка: € 0.00	🗑️
Успішна угода	Apple AirPods Max Sky Blue (MGYL3)	1	Самовивіз	Онлайн	17.04.2024 13:47	Товар: € 1.00 Доставка: € 0.00	🗑️
Успішна угода	Навушники Apple AirPods 3 (MME73)	1	Самовивіз	Онлайн	25.04.2024 14:58	Товар: € 1.00 Доставка: € 0.00	🗑️
Успішна угода	Навушники Apple AirPods 3 (MME74)	1	Самовивіз	При отриманні	26.04.2024 13:37	Товар: € 1.00 Доставка: € 0.00	🗑️
Успішна угода	Apple Watch Series 9 GPS 45mm Silver Aluminum Case w. Winter Blue S. Loop (MR9F3)	1	Самовивіз	При отриманні	02.05.2024 11:00	Товар: € 9000.00 Доставка: € 0.00	🗑️

Рис. 3.5. Демонстрація вкладки «Запити покупців»

Далі продавці мають або підтвердити, або відхилити замовлення. При підтвердженні вони мають запропонувати свою ціну за товар та доставку. Дивитись рис. 3.6.



The image shows a modal window titled "Створити пропозицію" (Create Offer) with a close button (X) in the top right corner. The form contains the following fields:

- Товар:** Apple iPhone 14 256GB (Blue) (MPWP3)
- Опис:** A text area containing the word "НОВИЙ" (New).
- Кількість:** A numeric input field with the value "2".
- Сервіс доставки:** A dropdown menu with "Самовивіз" (Self-pickup) selected.
- Місто:** A text field with "м. Київ" (Kyiv).
- Вартість товару:** A numeric input field with "32000" and a Euro symbol (€) to its right. A mouse cursor is pointing at the input.
- Вага товару:** A numeric input field with "1" and a Kilogram symbol (КГ) to its right.

At the bottom of the modal is a large blue button labeled "Надіслати пропозицію" (Send Offer).

Рис. 3.6. Демонстрація модального вікна «Створити пропозицію»

Покупець може спостерігати за усіма пропозиціями у реальному часі.  
Дивитись рис. 3.7.

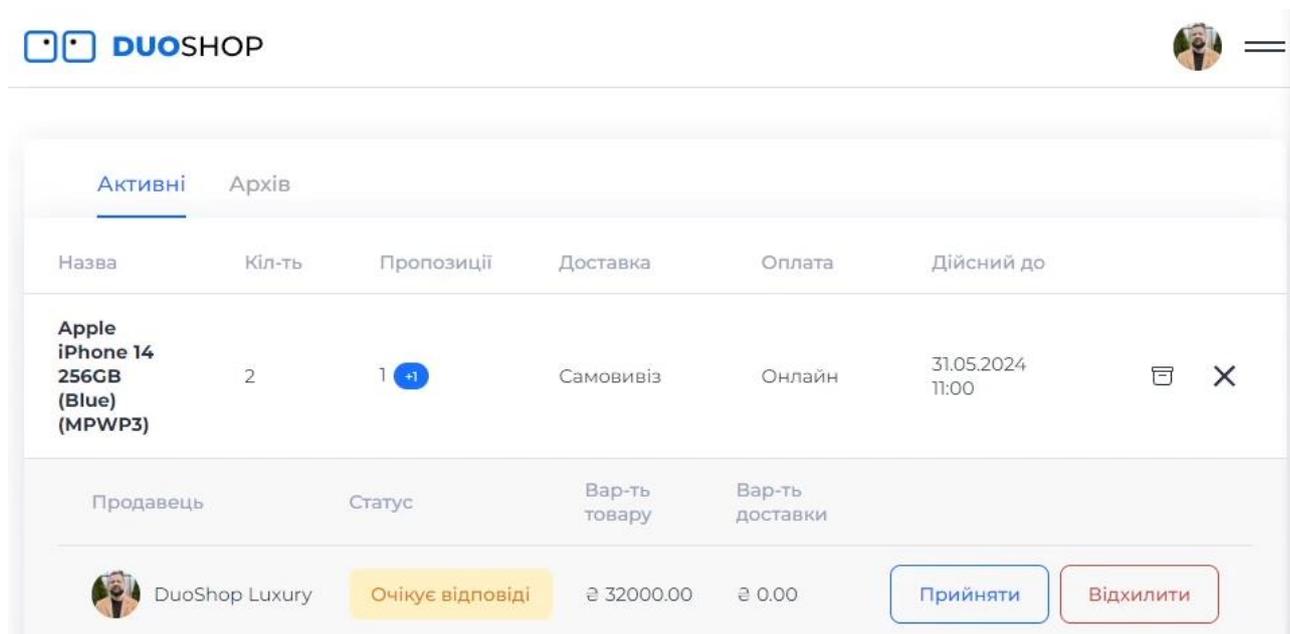


Рис. 3.7. Демонстрація нової пропозиції від продавця

Покупець має або прийняти, або відхилити пропозицію. При підтвердженні між продавцем та покупцем створиться чат. Дивитись рис. 3.8 та 3.9.

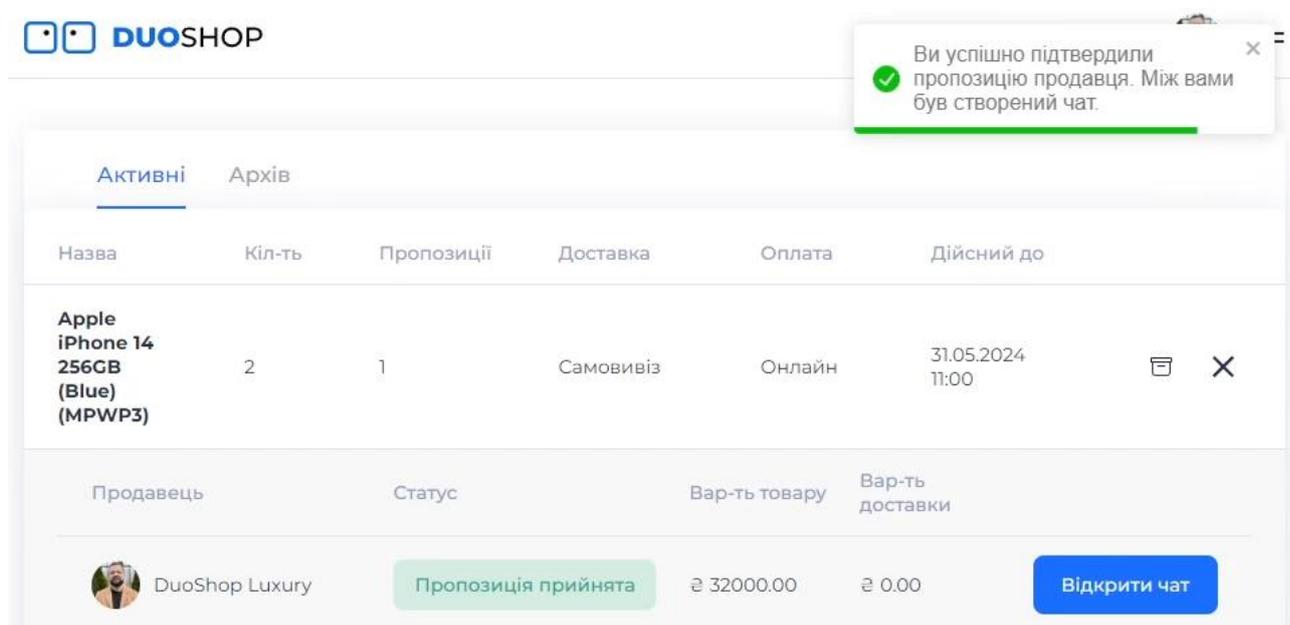


Рис. 3.8. Момент після підтвердження пропозиції

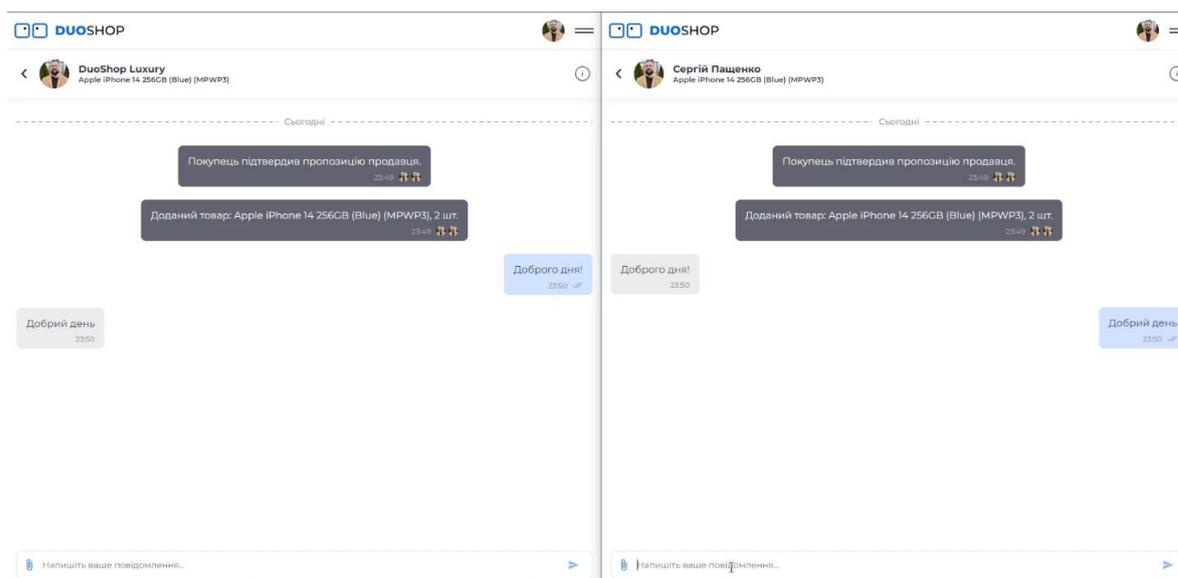


Рис. 3.9. Демонстрація чату

В чаті покупець та продавець можуть обговорити усі деталі угоди. При необхідності, можуть переглянути та змінити дані натиснувши на кнопку зверху праворуч, що під бургер меню. Дивитись рис. 3.10.

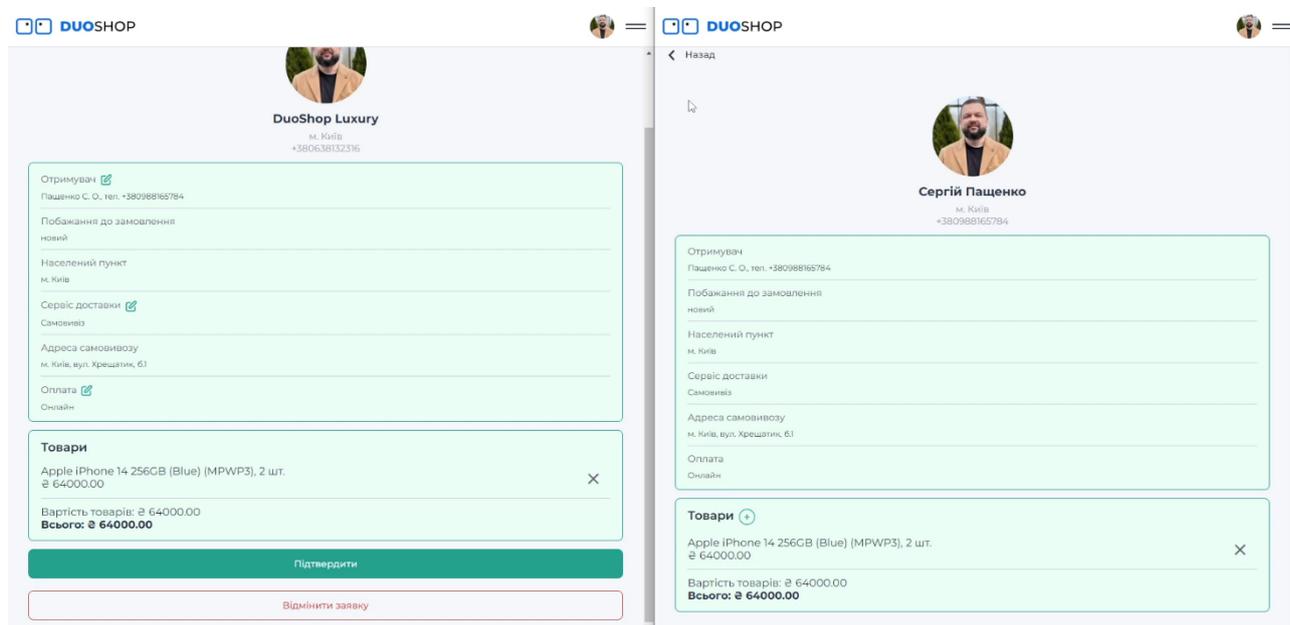


Рис. 3.10. Інформаційна панель чату

Тут можна побачити усі дані, які вказував покупець та продавець. Спробуємо додати товар. Але це може робити лише продавець. Дивитись рис. 3.11.

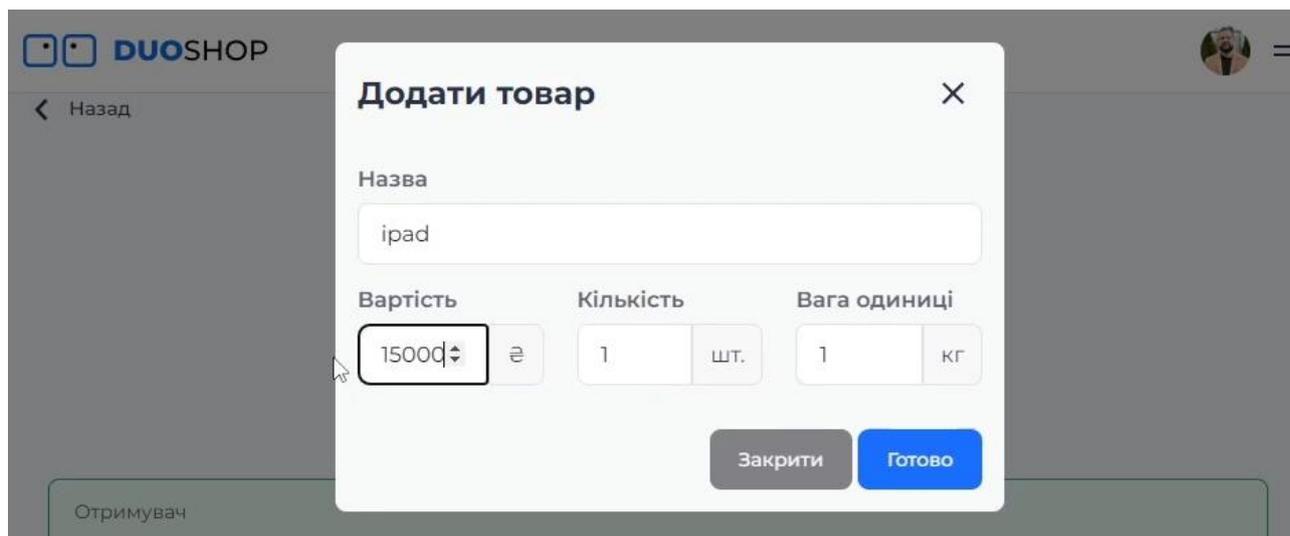


Рис. 3.11. Модальне вікно «Додати товар»

Тут продавцю необхідно вказати назву товару, його вартість, кількість, вагу одиниці та натиснути кнопку «Готово». Після чого обидва учасники отримають відповідне сповіщення. Дивитись рис. 3.12.

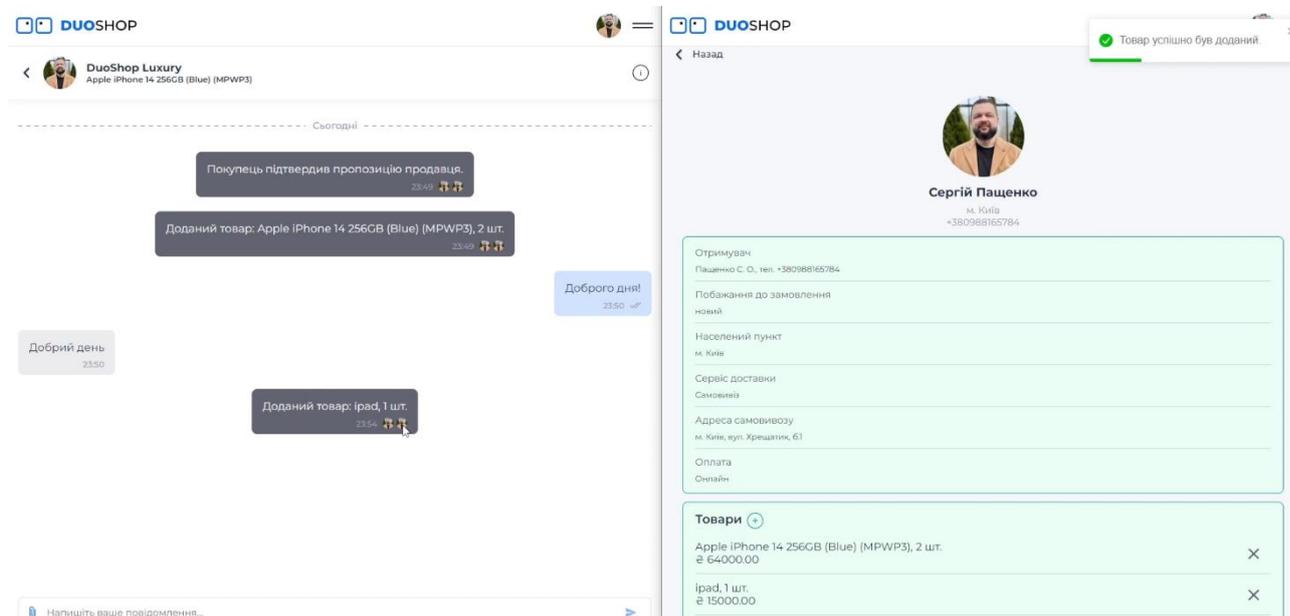


Рис. 3.12. Момент після того, як додали новий товар

В чаті можна обмінюватися не тільки текстовими повідомленнями, але й файлами. Дивитись рис. 3.13.

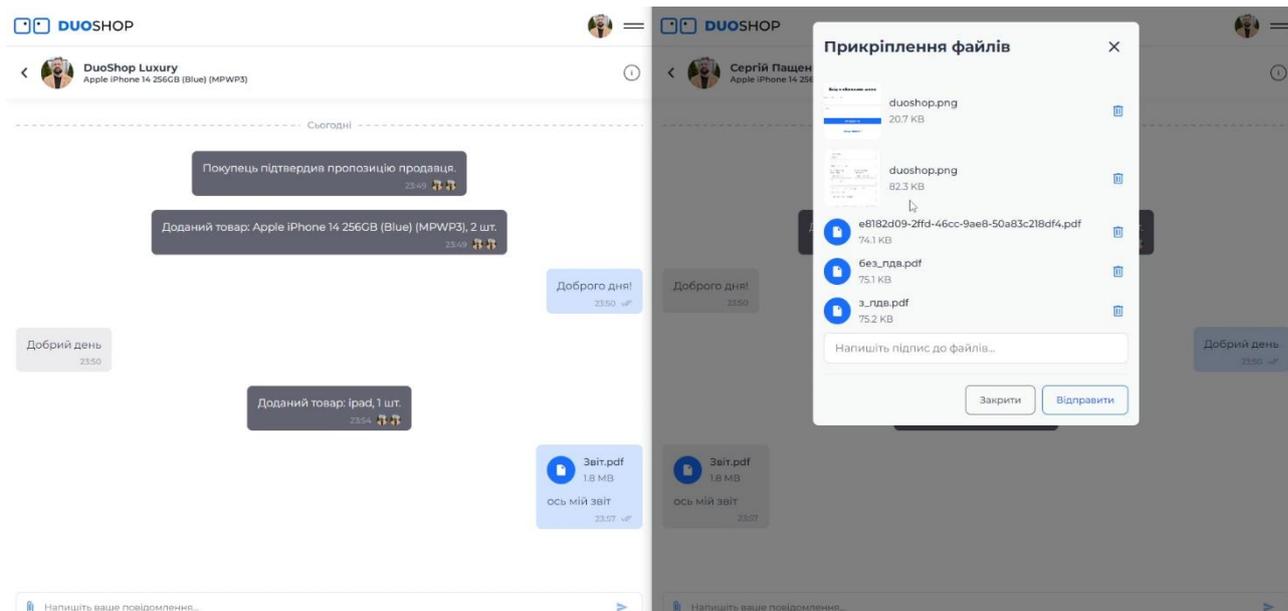


Рис. 3.13. Демонстрація можливості обміну файлами

Але файл не може важити більше 12МБ. І за один раз не можна відправити більше десяти файлів. Інакше буде відповідна помилка. Це зображено на рис. 3.14.

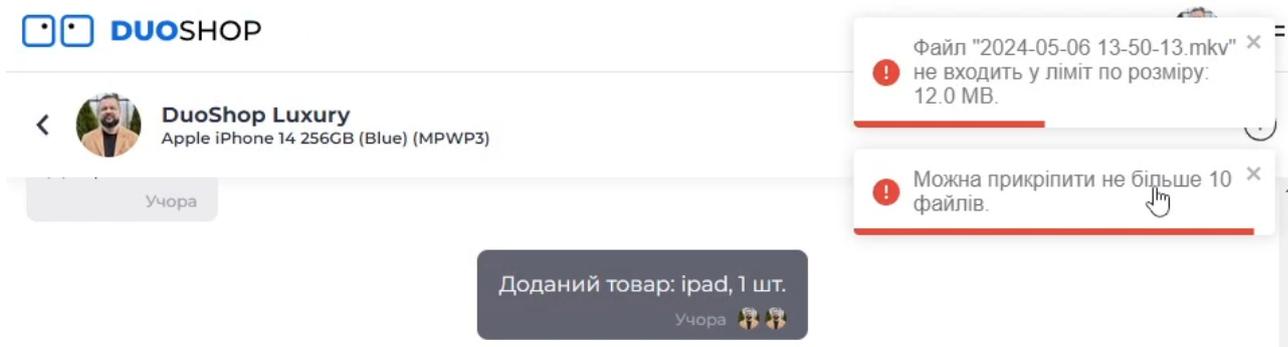


Рис. 3.14. Помилки про перевищення лімітів при спробі відправити файли

Коли усі деталі були обговорені, то покупець може підтвердити та оплатити замовлення. Відповідну кнопку можна побачити на рис. 25.

## ВИСНОВКИ

В даній кваліфікаційній роботі було розроблено веб-додаток для маркетплейсу з використанням сучасних технологій, таких як Python, TypeScript, Django та Next.js. Цей додаток спрямований на автоматизацію процесу купівлі-продажу товарів, що значно полегшує життя споживачам та продавцям.

Основні досягнення роботи включають аналіз ринку та потреб користувачів, де визначено основні пріоритети споживачів при виборі товарів: ціна, якість, умови покупки та витрачений час. Було проаналізовано існуючі рішення (Hotline та eKatalog) та визначено їхні недоліки. У процесі проектування та розробки веб-додатку створено архітектуру системи, що включає реєстрацію, авторизацію, профілі користувачів, кабінет, запити та чат. Також розроблено алгоритми для автоматизованого пошуку товарів з врахуванням індивідуальних пріоритетів споживачів, забезпечено актуальність цін та наявності товарів у реальному часі, що є ключовою перевагою даного додатку.

В рамках інтеграції інноваційних технологій було використано фреймворки та технології, такі як Django, FastAPI, Next.js та Docker, а також впроваджено можливість інтеграції ChatGPT для полегшення взаємодії з користувачами та надання їм релевантної інформації.

Автоматизована система дозволяє споживачам оперативно отримувати інформацію про наявні пропозиції на ринку, що значно покращує якість обслуговування клієнтів. Платформа забезпечує зручний доступ до даних про товар, його вартість, умови покупки та доставку. Інтернет-магазини, що впровадять дану систему, зможуть значно підвищити свою конкурентоспроможність за рахунок покращення обслуговування клієнтів та збільшення обсягів продажів. Використання актуальних даних про товари та їх наявність дозволяє споживачам робити більш обґрунтовані та швидкі рішення щодо покупки.

Система має потенціал для подальшого розвитку, зокрема, впровадження системи відгуків. Рекомендовано розширити функціонал платформи, додавши можливість інтеграції з іншими сервісами. Таким чином, розроблений веб-додаток для маркетплейсу має значний потенціал для вдосконалення процесу купівлі-продажу товарів, підвищення рівня задоволеності клієнтів та збільшення ефективності роботи інтернет-магазинів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Український веб-довідник – Доступно за посиланням: <https://css.in.ua/> (Дата звернення: 10 червня 2024)
2. The Modern JavaScript Tutorial – Доступно за посиланням: <https://javascript.info/> (Дата звернення: 10 червня 2024)
3. Python Documentation. Python Docs – Доступно за посиланням: <https://docs.python.org/> (Дата звернення: 10 червня 2024).
4. Django Documentation. Django Docs – Доступно за посиланням: <https://docs.djangoproject.com/> (Дата звернення: 10 червня 2024).
5. FastAPI Documentation. FastAPI Docs – Доступно за посиланням: <https://fastapi.tiangolo.com/> (Дата звернення: 10 червня 2024).
6. TypeScript Documentation. TypeScript Docs – Доступно за посиланням: <https://www.typescriptlang.org/docs/> (Дата звернення: 10 червня 2024).
7. Next.js Documentation. Next.js Docs – Доступно за посиланням: <https://nextjs.org/docs> (Дата звернення: 10 червня 2024).
8. Docker Documentation. Docker Docs – Доступно за посиланням: <https://docs.docker.com/> (Дата звернення: 10 червня 2024).
9. ChatGPT Integration Guide. OpenAI Documentation – Доступно за посиланням: <https://platform.openai.com/docs/> (Дата звернення: 10 червня 2024).
10. Опис API для взаємодії із серверною частиною додатка «Автоклієнт» - Доступно за посиланням: <https://docs.google.com/document/d/e/2PACX-1vTtKvGa3P4E-1DqLg3bHRF6Wi9S7GIjSMFEFхII5qQZBGхuTXs25hQNiUU1hMZQhOух6BNvIZ1bVKSr/pub> (Дата звернення: 10 червня 2024).

## ДОДАТКИ

### Додаток А. Файл `docker-compose.prod.yml`

```
version: '3.8'
```

```
services:
```

```
  redis:
```

```
    image: redis:7.2-alpine
```

```
  mongo_db:
```

```
    image: mongo:4.4.6
```

```
    volumes:
```

```
      - mongo_data:/var/lib/mongo/data/
```

```
    environment:
```

```
      - MONGO_INITDB_DATABASE=${COMMON_MONGO_DATABASE}
```

```
      - MONGO_INITDB_ROOT_USERNAME=${COMMON_MONGO_USER}
```

```
      -
```

```
      MONGO_INITDB_ROOT_PASSWORD=${COMMON_MONGO_PASSWORD}
```

```
  delivery_manager:
```

```
    build:
```

```
      context: .
```

```
      dockerfile: ./compose/production/delivery-manager/Dockerfile
```

```
    command: ./start-delivery-manager.sh
```

```
    env_file:
```

```
      - ./compose/production/delivery-manager/.env
```

```
    environment:
```

```
      - DM_MONGO_HOST=${COMMON_MONGO_HOST}
```

- DM\_MONGO\_PORT=\${COMMON\_MONGO\_PORT}
- DM\_MONGO\_DATABASE=\${COMMON\_MONGO\_DATABASE}
- DM\_MONGO\_USER=\${COMMON\_MONGO\_USER}
- DM\_MONGO\_PASSWORD=\${COMMON\_MONGO\_PASSWORD}
- DM\_CELERY\_BROKER\_URL=\${COMMON\_CELERY\_BROKER\_URL}

depends\_on:

- mongo\_db
- redis

delivery\_manager\_celery:

build:

context: .

dockerfile: ./compose/production/delivery-manager/Dockerfile

command: ./start-delivery-manager-celery.sh

env\_file:

- ./compose/production/delivery-manager/.env

environment:

- DM\_MONGO\_HOST=\${COMMON\_MONGO\_HOST}
- DM\_MONGO\_PORT=\${COMMON\_MONGO\_PORT}
- DM\_MONGO\_DATABASE=\${COMMON\_MONGO\_DATABASE}
- DM\_MONGO\_USER=\${COMMON\_MONGO\_USER}
- DM\_MONGO\_PASSWORD=\${COMMON\_MONGO\_PASSWORD}
- DM\_CELERY\_BROKER\_URL=\${COMMON\_CELERY\_BROKER\_URL}

depends\_on:

- delivery\_manager

postgres\_db:

image: postgres:15.4-alpine

volumes:

- postgres\_data:/var/lib/postgresql/data/

environment:

- POSTGRES\_DB=\${COMMON\_POSTGRES\_DATABASE}
- POSTGRES\_USER=\${COMMON\_POSTGRES\_USER}
- POSTGRES\_PASSWORD=\${COMMON\_POSTGRES\_PASSWORD}

es:

image: elasticsearch:8.10.2

build:

context: .

dockerfile: ./compose/dev/es/Dockerfile

volumes:

- elasticsearch:/usr/share/elasticsearch/data

environment:

- discovery.type=single-node
- xpack.security.enabled=false

backend:

build:

context: .

dockerfile: ./compose/production/backend/Dockerfile

command: ./start-backend.sh

volumes:

- static\_volume:/home/app/backend/public\_files/static
- media\_volume:/home/app/backend/public\_files/media
- ./apps/backend/src/logs:/home/app/backend/logs

env\_file:

- ./compose/production/backend/.env

environment:

```

- BACKEND_DOMAIN=${COMMON_ENTRYPOINT_DOMAIN}
- BACKEND_FONDY_IP=${COMMON_FONDY_IP}
-
BACKEND_ALLOWED_HOSTS=${COMMON_ENTRYPOINT_DOMAIN},nginx
x
-
BACKEND_TRUSTED_ORIGINS=https://${COMMON_ENTRYPOINT_DOMAIN}
N}
-
BACKEND_ELASTICSEARCH_DSL_HOSTS=${COMMON_ELASTICSEARCH
_DSL_HOSTS}
- BACKEND_SQL_HOST=${COMMON_POSTGRES_HOST}
- BACKEND_SQL_PORT=${COMMON_POSTGRES_PORT}
- BACKEND_SQL_DATABASE=${COMMON_POSTGRES_DATABASE}
- BACKEND_SQL_USER=${COMMON_POSTGRES_USER}
- BACKEND_SQL_PASSWORD=${COMMON_POSTGRES_PASSWORD}
-
BACKEND_CELERY_BROKER_URL=${COMMON_CELERY_BROKER_URL}
depends_on:
- redis
- postgres_db
- es

backend_celery:
build:
context: .
dockerfile: ./compose/production/backend/Dockerfile
command: ./start-backend-celery.sh
env_file:
- ./compose/production/backend/.env
environment:

```

```

- BACKEND_DOMAIN=${COMMON_ENTRYPOINT_DOMAIN}
- BACKEND_FONDY_IP=${COMMON_FONDY_IP}
-
BACKEND_ALLOWED_HOSTS=${COMMON_ENTRYPOINT_DOMAIN},ngin
x
-
BACKEND_TRUSTED_ORIGINS=https://${COMMON_ENTRYPOINT_DOMAI
N}
-
BACKEND_ELASTICSEARCH_DSL_HOSTS=${COMMON_ELASTICSEARCH
_DSL_HOSTS}
- BACKEND_SQL_HOST=${COMMON_POSTGRES_HOST}
- BACKEND_SQL_PORT=${COMMON_POSTGRES_PORT}
- BACKEND_SQL_DATABASE=${COMMON_POSTGRES_DATABASE}
- BACKEND_SQL_USER=${COMMON_POSTGRES_USER}
- BACKEND_SQL_PASSWORD=${COMMON_POSTGRES_PASSWORD}
-
BACKEND_CELERY_BROKER_URL=${COMMON_CELERY_BROKER_URL}
volumes:
- ./apps/backend/src/logs:/home/app/backend/logs
depends_on:
- backend

nginx:
build:
context: .
dockerfile: ./compose/production/nginx/Dockerfile
environment:
- NGINX_DOMAIN=${COMMON_ENTRYPOINT_DOMAIN}
volumes:
- static_volume:/home/app/backend/public_files/static

```

- media\_volume:/home/app/backend/public\_files/media
- ./compose/production/nginx/certs:/etc/nginx/certs

ports:

- 80:80
- 443:443

depends\_on:

- backend

frontend:

build:

context: .

dockerfile: ./compose/production/frontend/Dockerfile

depends\_on:

- nginx

volumes:

mongo\_data:

postgres\_data:

static\_volume:

media\_volume:

elasticsearch: