

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Відокремлений структурний підрозділ
«Тернопільський фаховий коледж
Тернопільського національного технічного університету імені Івана Пулюя»
Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

фахового молодшого бакалавра

на тему: Розробка вебзастосунку для обліку товарів магазину «SwagStore»

Виконав: студент IV курсу, групи КН-421

спеціальності: 122 Комп'ютерні науки

Максим МАЛИЙ

Керівник

Ігор КАПАЦІЛА

Рецензент

(ім'я та прізвище)

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
ІМЕНІ ІВАНА ПУЛЮЯ»

Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук
Освітньо-професійний ступінь «фаховий молодший бакалавр»
Спеціальність 122 Комп'ютерні науки
Галузь знань 12 Інформаційні технології

ЗАТВЕРДЖУЮ

Голова циклової комісії
комп'ютерних наук

_____ Галина МАРЦІЯШ

« 02 » березня 2026 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

_____ Малому Максиму Олеговичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка вебдодатку для обліку товарів магазину "SwagStore"

керівник роботи _____ Капаціла Ігор Богданович _____,
(прізвище, ім'я, по батькові)

затверджені наказом вищого навчального закладу № 4/9-132 від 27.02.2026 р.

2. Строк подання студенткою роботи: 19.06.2026 р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення, мови програмування: TypeScript, JavaScript; фреймворки та бібліотеки: Next.js, React, Node.js, Express.js, Tailwind CSS; база даних: MongoDB; ODM-бібліотека: Mongoose; засоби авторизації: JWT, стандарти IEEE 29148-2018, IEEE 29119, ДСТУ 8302:2015.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

- 1 Загальний розділ
 - 1.1 Аналітичний огляд існуючих рішень
 - 1.2 Технічне завдання
 - 1.2.1 Найменування та область застосування
 - 1.2.2 Призначення розробки
 - 1.2.3 Вимоги до функціоналу вебзастосунку
 - 1.2.4 Вимоги до програмної документації
 - 1.2.5 Техніко-економічні показники
 - 1.2.6 Стадії та етапи розробки
 - 1.2.7 Порядок тестування та прийому

- 2 Розробка технічного та робочого проекту

- 2.1 Розробка структури вебзастосунку
- 2.2 Створення та верстка сторінок вебзастосунку
- 2.3 Розробка структури бази даних
- 2.4 Програмування вебзастосунку
 - 2.4.1 Написання клієнтської частини
 - 2.4.2 Написання серверної частини
- 2.5 Тестування вебзастосунку
- 3 Спеціальний розділ
 - 3.1 Розміщення вебзастосунку та вимоги до середовища
 - 3.2 Інструкція адміністратора для роботи з вебзастосунком
 - 3.3 Підтримка, оновлення та подальший розвиток системи
- 4 Економічний розділ
 - 4.1 Визначення стадій технологічного процесу та загальної тривалості проведення робіт
 - 4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи
 - 4.3 Розрахунок витрат на електроенергію
 - 4.4 Розрахунок суми амортизаційних відрахувань
 - 4.5 Обчислення накладних витрат
 - 4.6 Складання кошторису витрат та визначення собівартості розробки
 - 4.7 Розрахунок ціни розробки
 - 4.8 Визначення економічної ефективності і терміну окупності
- 5 Охорона праці, техніка безпеки та екологічні вимоги
 - 5.1 Відповідальність за порушення законодавства про охорону праці
 - 5.2 Комплекс заходів по зменшенню шуму в джерелі його виникнення
- 6 Висновки

Додаткові вказівки: виконання кваліфікаційної роботи із розробкою програмного продукту – вебсайту.

5. Перелік графічного матеріалу:

1. Схема структурна клієнтської частини
2. UML-діаграма станів
3. ER-діаграма бази даних вебсайту
4. Таблиця техніко-економічних показників

6. Консультанти розділів роботи

| Розділ | Ім'я, прізвище та посада консультанта | Підпис, дата | |
|---|---------------------------------------|----------------|------------------|
| | | завдання видав | завдання прийняв |
| Економічний розділ | Любов КАЛУШКА | | |
| Охорона праці, техніка безпеки та екологічні вимоги | Генадій ГОРЯЧЕК | | |

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|-------|--|-------------------------------|----------|
| 1 | Отримання і аналіз технічного завдання | 05.03.2026 | |
| 2 | Збір і узагальнення інформації | 20.03.2026 | |
| 3 | Написання першого розділу | 28.04.2026 | |
| 4 | Розробка технічного та робочого проєкту | 07.05.2026 | |
| 5 | Написання спеціального розділу | 20.05.2026 | |
| 6 | Розрахунок економічної частини | 28.05.2026 | |
| 7 | Написання розділу охорони праці | 06.06.2026 | |
| 8 | Виконання графічної частини | 01.06.2026 | |
| 9 | Оформлення пояснювальної записки | 11.06.2026 | |
| 10 | Погодження нормоконтролю | 12.06.2026 | |
| 11 | Попередній захист кваліфікаційної роботи | .06.2026 | |
| 12 | Захист кваліфікаційної роботи | .06.2026 | |

7. Дата видачі завдання: 05 березня 2026 р.

Студент

_____ (підпис)

Максим МАЛИЙ

Керівник роботи

_____ (підпис)

Ігор КАПАЦІЛА

ЗМІСТ

| | |
|--|----|
| АНОТАЦІЯ..... | 7 |
| ВСТУП..... | 9 |
| 1 ЗАГАЛЬНИЙ РОЗДІЛ..... | 10 |
| 1.1 Аналітичний огляд існуючих рішень..... | 10 |
| 1.2 Технічне завдання..... | 11 |
| 2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ..... | 17 |
| 2.1 Розробка структури вебзастосунку..... | 17 |
| 2.2 Створення та верстка сторінок вебзастосунку..... | 19 |
| 2.3 Розробка структури бази даних..... | 20 |
| 2.4 Програмування вебзастосунку..... | 22 |
| 2.5 Тестування вебзастосунку..... | 33 |
| 3 СПЕЦІАЛЬНИЙ РОЗДІЛ..... | 46 |
| 3.1 Розміщення вебзастосунку та вимоги до середовища..... | 46 |
| 3.2 Інструкція адміністратора для роботи з вебзастосунком..... | 48 |
| 3.3 Підтримка, оновлення та подальший розвиток системи..... | 55 |
| 4 ЕКОНОМІЧНИЙ РОЗДІЛ..... | 57 |
| 4.1 Визначення стадій технологічного процесу та загальної тривалості проведення робіт..... | 57 |
| 4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи.. | 59 |
| 4.3 Розрахунок витрат на електроенергію..... | 60 |
| 4.4 Розрахунок суми амортизаційних відрахувань..... | 61 |
| 4.5 Обчислення накладних витрат..... | 61 |
| 4.6 Складання кошторису витрат та визначення собівартості розробки..... | 62 |
| 4.7 Розрахунок ціни розробки..... | 62 |
| 4.8 Визначення економічної ефективності і терміну окупності..... | 63 |

| | | | | | | | | |
|-----------|------|---------------|--------|------|--|---|------|---------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | | | |
| Зм. | Арк. | № докум. | Підпис | Дата | | | | |
| Розроб. | | Малий М.О. | | | «Розробка вебзастосунку для обліку товарів магазину «SwagStore»» Пояснювальна записка | Літ. | Арк. | Аркушів |
| Перевір. | | Капаціла І.Б. | | | | | 5 | 103 |
| Реценз. | | | | | | ВСП ТФК ТНТУ КН-421 м. Тернопіль | | |
| Н. Контр. | | Приймак В.А. | | | | | | |
| Затверд. | | | | | | | | |

| | |
|--|-----|
| 5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ..... | 65 |
| 5.1 Відповідальність за порушення законодавства про охорону праці..... | 65 |
| 5.2 Комплекс заходів по зменшенню шуму в джерелі його виникнення..... | 67 |
| ВИСНОВКИ..... | 70 |
| ПЕРЕЛІК ПОСИЛАНЬ..... | 72 |
| ДОДАТКИ..... | 74 |
| Додаток А. Лістинг файлу client/package.json..... | 74 |
| Додаток Б. Лістинг файлу server/package.json..... | 76 |
| Додаток В. Лістинг файлу моделі товару Goods.ts..... | 78 |
| Додаток Г. Лістинг файлу моделі користувача User.ts..... | 81 |
| Додаток Д. Лістинг файлу моделі замовлення Order.ts..... | 83 |
| Додаток Е. Лістинг файлу моделі складського обліку Inventory.ts..... | 86 |
| Додаток Ж. Лістинг файлу маршруту авторизації auth.ts..... | 88 |
| Додаток И. Лістинг файлу маршруту товарів goods.ts..... | 90 |
| Додаток К. Лістинг файлу маршруту замовлень orders.ts..... | 93 |
| Додаток Л. Лістинг файлу маршруту складського обліку inventory.ts..... | 96 |
| Додаток М. Лістинг файлу сторінки детального перегляду товару..... | 98 |
| Додаток Н. Лістинг файлу сторінки оформлення замовлення..... | 101 |

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 6 |

АНОТАЦІЯ

Тема дипломного проєкту: «Розробка вебзастосунку для інтернет-магазину одягу та обліку товарів “SwagStore”».

Метою дипломного проєкту є розробка вебзастосунку, який поєднує функції інтернет-магазину одягу та системи обліку товарів. Розроблений програмний продукт призначений для перегляду асортименту клієнтами, оформлення замовлень, керування товарами, контролю складських залишків та обробки звернень користувачів.

Пояснювальна записка складається з п'яти розділів. У загальному розділі розглянуто актуальність теми, виконано аналіз існуючих рішень і сформовано технічне завдання. У розділі розробки технічного та робочого проєкту описано структуру вебзастосунку, клієнтську і серверну частини, базу даних, адміністративну панель, каталог товарів, кошик, оформлення замовлень і складський облік. Для реалізації системи використано Next.js, React, TypeScript, Node.js, Express, MongoDB та Mongoose.

У спеціальному розділі наведено вимоги до розміщення вебзастосунку, інструкцію адміністратора та питання подальшої підтримки системи. В економічному розділі виконано розрахунок витрат на розробку, собівартості та економічної ефективності проєкту. У розділі охорони праці розглянуто питання безпечної роботи за комп'ютером і заходи щодо зменшення шуму на робочому місці.

Розроблений вебзастосунок «SwagStore» забезпечує перегляд каталогу товарів, фільтрацію за основними параметрами, вибір розміру, додавання товарів у кошик, оформлення замовлення, перегляд історії замовлень, а також керування товарами, категоріями, залишками, клієнтами та замовленнями через адміністративну частину. До складу дипломного проєкту також входить графічна частина зі схемами та техніко-економічними показниками.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| | | | | | | 7 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

ABSTRACT

The topic of the diploma project is “Development of a Web Application for an Online Clothing Store and Product Accounting System ‘SwagStore’”.

The purpose of the diploma project is to develop a web application that combines the functions of an online clothing store and a product accounting system. The developed software product is intended for viewing products by customers, placing orders, managing goods, controlling warehouse stock, and processing user requests.

The explanatory note consists of five sections. The general section describes the relevance of the topic, analyzes existing solutions, and formulates the technical specification. The technical and working project development section describes the structure of the web application, the client and server parts, the database, the administrative panel, the product catalog, the shopping cart, order processing, and warehouse accounting. The system was implemented using Next.js, React, TypeScript, Node.js, Express, MongoDB, and Mongoose.

The special section presents the requirements for deploying the web application, administrator instructions, and further system support. The economic section contains the calculation of development costs, cost price, and economic efficiency. The occupational safety section considers safe computer work and measures to reduce noise in the workplace.

The developed “SwagStore” web application provides product catalog browsing, filtering by main parameters, size selection, adding products to the cart, placing orders, viewing order history, and managing goods, categories, stock, customers, and orders through the administrative part. The diploma project also includes a graphical part with diagrams and technical and economic indicators.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 8 |

ВСТУП

У сучасному світі сфера електронної комерції є однією з найбільш динамічно зростаючих галузей, що потребує ефективного управління всіма ресурсами, зокрема товарами, замовленнями, персоналом та клієнтами. Для забезпечення стабільної роботи інтернет-магазину одягу важливо мати надійний механізм контролю за складськими залишками, їхнім обліком і швидкою обробкою замовлень. Традиційні методи ведення записів, такі як електронні таблиці або розрізнені системи для вітрини та складу, часто виявляються неефективними через велику ймовірність помилок, складність оновлення інформації в режимі реального часу та обмежену можливість аналізу даних. Це може призводити до продажу товарів, яких вже немає в наявності, перевитрат ресурсів і, як наслідок, до фінансових втрат та зниження рівня довіри покупців.

Впровадження сучасних вебтехнологій для автоматизації цих процесів є необхідністю для розвитку та підвищення конкурентоспроможності підприємств роздрібною торгівлі.

Метою даної дипломної роботи є розробка вебзастосунку для інтернет-магазину одягу та обліку товарів «SwagStore», який дозволить автоматизувати управління торговими процесами та значно спростити контроль за складськими ресурсами. Застосунок буде спрямований на вирішення низки важливих завдань, включаючи відстеження залишків за розмірами та кольорами, ведення історії замовлень і створення зручного інтерфейсу для покупців та персоналу. Його впровадження сприятиме підвищенню ефективності управління асортиментом, покращенню якості обслуговування клієнтів і загальному розвитку торгового майданчика.

Розробка такого вебзастосунку стане важливим кроком у цифровізації бізнес-процесів магазину, що дозволить забезпечити його стабільне функціонування та конкурентні переваги на ринку електронної комерції.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| | | | | | | 9 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

Сучасний ринок програмного забезпечення пропонує широкий вибір рішень для автоматизації процесів торгівлі та створення інтернет-магазинів. Зважаючи на специфічні потреби підприємств електронної комерції, можна виділити кілька основних підходів: готові хмарні платформи (SaaS), системи управління контентом (CMS), спеціалізовані системи управління запасами та самостійно розроблені вебзастосунки. Кожен з цих варіантів має свої переваги та обмеження, які необхідно враховувати при виборі оптимального рішення для конкретного бізнесу.

Готові програмні рішення, такі як Shopify або Wix, широко використовуються в індустрії та надають комплексний функціонал для швидкого запуску вітрини. Вони дозволяють розміщувати товари, приймати оплати та контролювати базові фінансові операції. Завдяки наявності хмарних технологій такі сервіси забезпечують зручний доступ до управління з будь-якого пристрою. Однак, більшість готових рішень працюють на підписній основі з додатковими комісіями за транзакції, що може стати значним фінансовим навантаженням для бізнесу. Додатковим недоліком є обмежена можливість глибокої адаптації логіки списання складських залишків під специфічні моделі (наприклад, складні варіації розмірів та кольорів одягу).

Системи управління контентом, такі як WordPress із модулем WooCommerce, орієнтовані на більший контроль над платформою та встановлюються на власний хостинг. Вони забезпечують автоматизацію каталогу та мають безліч плагінів. Однак використання такої архітектури для великих інтернет-магазинів із високим навантаженням часто призводить до проблем зі швидкодією. Крім того, ці системи потребують глибокої інтеграції з іншими програмами (наприклад, Zoho Inventory) для повноцінного складського обліку, що може стати додатковим викликом і вимагає залучення технічних фахівців.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 10 |

Самостійно розроблені вебзастосунки є альтернативним варіантом для тих, хто прагне створити систему, що повністю відповідатиме специфічним потребам бізнесу. Вони дозволяють отримати максимальну гнучкість у реалізації функціоналу, контроль над даними та незалежність від сторонніх постачальників програмного забезпечення. Завдяки використанню сучасних технологій, таких як екосистема JavaScript (фреймворки Next.js та React для клієнта, Node.js для сервера), створення власного вебзастосунку стає найбільш ефективним рішенням для автоматизації повного циклу інтернет-магазину одягу. Основними перевагами такого підходу є можливість налаштування атомарного списання інвентарю, інтеграція інтерактивних кошиків покупок та висока масштабованість у майбутньому.

Отже, аналіз існуючих рішень показав, що готові платформи електронної комерції та CMS-системи мають широкий функціонал, однак не завжди повністю відповідають потребам конкретного магазину одягу. Саме тому розробка власного вебзастосунку «SwagStore» є доцільною, оскільки дозволяє поєднати онлайн-каталог, кошик, оформлення замовлень, складський облік та адміністративне керування в одній системі. Такий підхід забезпечує гнучкість, зрозумілу структуру та можливість подальшого розширення функціоналу.

1.2 Технічне завдання

1.2.1 Найменування та область застосування

Тема дипломного проєкту — «Розробка вебзастосунку для інтернет-магазину одягу та обліку товарів “SwagStore”». Цей вебзастосунок призначений для автоматизації процесів електронної комерції та внутрішнього складського обліку товарів магазину одягу. Він дозволяє клієнтам зручно переглядати асортимент, обирати потрібні товари, додавати їх до кошика та оформлювати замовлення онлайн. Адміністратор, у свою чергу, має можливість вести облік

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 11 |

наявних товарних позицій, контролювати залишки на складі та своєчасно опрацьовувати замовлення клієнтів.

Система дає змогу додавати нові товари до каталогу, вказувати їх основні характеристики, зокрема назву, бренд, ціну, категорію, розділ, доступні кольори, розміри та зображення. Крім того, вебзастосунок забезпечує можливість відстеження руху товарів на складі, включаючи автоматичне списання відповідної кількості після успішного оформлення замовлення користувачем.

1.2.2 Призначення розробки

Призначенням розробки вебзастосунку «SwagStore» є створення ефективного інструменту для автоматизації процесів управління інтернет-магазином, що забезпечить оптимізацію роботи бізнесу. Вебзастосунок дозволить вирішити такі ключові завдання:

- онлайн-продажі: надання користувачам зручного інтерфейсу для пошуку одягу, додавання його до кошика та безпечного оформлення замовлень;
- облік товарів: забезпечення точного контролю за залишками продукції на складі, їх надходженням та списанням;
- управління запасами: надання адміністраторам інформації про поточний стан інвентарю та попередження про низький залишок для уникнення дефіциту популярних розмірів;
- зручність використання: створення інтуїтивного інтерфейсу для персоналу магазину, що спростить роботу з модерацією клієнтів, відгуками та замовленнями і мінімізує помилки.

Розробка орієнтована на поліпшення якості управління ресурсами магазину, забезпечення економії часу персоналу, підвищення рентабельності бізнесу, а також покращення рівня обслуговування клієнтів через гарантовану наявність усіх необхідних товарів.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 12 |

1.2.3 Вимоги до функціоналу вебзастосунку

Для ефективної роботи з вебзастосунком інтернет-магазину важливо забезпечити інтуїтивно зрозумілий інтерфейс та широкий функціонал, який дозволить здійснювати повноцінний контроль за торговими процесами. Основна мета цієї системи — спростити процеси купівлі для клієнтів та управління каталогом для адміністраторів.

Ключовою функцією клієнтської частини є перегляд інтерактивного каталогу товарів із можливістю динамічної фільтрації (за ціною, розміром, брендом, категорією). Користувачі повинні мати можливість додавати обрані товари до кошика, редагувати їх кількість та оформлювати замовлення із зазначенням даних для доставки. Авторизовані користувачі додатково отримують доступ до історії власних замовлень та можливості залишати відгуки до товарів.

Для адміністраторів головною функцією є зберігання та управління інформацією про товари. Кожен запис у базі даних міститиме такі атрибути, як назва, категорія (чоловічий, жіночий одяг, аксесуари), фотографії, ціна та варіації (розмірна сітка і кольори). Організація даних за різними категоріями допоможе персоналу швидко знаходити потрібні позиції. Додавання нових товарів, редагування існуючих записів та можливість приховування товарів, які більше не продаються, є невід’ємною складовою функціональності вебзастосунку.

Забезпечення безпеки даних та управління доступом також відіграє важливу роль. В системі передбачено розмежування прав доступу за допомогою JWT-токенів. Адміністратори матимуть повний доступ до закритої панелі (дашборду), де зможуть керувати статусами замовлень, опрацьовувати запити до служби підтримки та контролювати складські залишки. Такий підхід захистить дані від несанкціонованих змін та забезпечить надійний контроль.

Завдяки впровадженню цієї функціональності вебзастосунок забезпечить автоматизований контроль товарних залишків, допоможе уникнути нестач та перевитрат, а також оптимізує процеси продажу, що сприятиме підвищенню

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 13 |

ефективності роботи персоналу та забезпеченню високого рівня сервісу для клієнтів інтернет-магазину.

1.2.4 Вимоги до програмної документації

Для забезпечення якості програмної документації вебзастосунку важливо мати чіткий і докладний опис архітектури програми, її функціональних модулів, а також інструкції, які допоможуть адміністраторам правильно налаштувати та використовувати систему. Потрібно мати наступне:

- детальний опис архітектури вебзастосунку, включаючи серверну та клієнтську частини, базу даних і взаємодію між компонентами;
- інструкції з встановлення програми, включаючи вимоги до середовища, порядок налаштування сервера, бази даних і клієнтського інтерфейсу;
- посібник користувача з докладним описом усіх функцій, таких як додавання товарів, редагування, пошук тощо;
- технічну документацію, що описує процес розробки, включаючи використані технології, структуру коду, методологію тестування та впровадження;
- опис політики безпеки програми, що включає управління доступом, захист даних і резервне копіювання;
- журнал змін та оновлень, у якому відображені всі внесені модифікації для забезпечення прозорості подальшої підтримки.

Ця документація забезпечує зручність у роботі з програмою, допомагає швидко впровадити її в експлуатацію та підтримувати протягом усього життєвого циклу.

1.2.5 Техніко-економічні показники

Для забезпечення ефективної роботи вебзастосунку «SwagStore» важливо визначити техніко-економічні показники, які дозволяють оцінити витрати на розробку, експлуатацію та підтримку системи, а також її економічну ефективність.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 14 |

Визначення цих показників сприяє більш обґрунтованому прийняттю рішень на кожному етапі життєвого циклу проєкту. Детальний опис розрахунків наведено в економічному розділі пояснювальної записки.

1.2.6 Стадії та етапи розробки

Створення вебзастосунку «SwagStore» було організовано як послідовний інженерний цикл, що дозволив інтегрувати функції магазину та системи обліку в єдину екосистему. Весь процес розробки було розділено на п'ять ключових етапів.

На першій фазі, концептуальному плануванні, було визначено архітектурні обмеження проєкту. Основна увага приділялася вибору інструментів, що забезпечують високу швидкість взаємодії: було вирішено використовувати React для клієнтського інтерфейсу та Node.js для серверної логіки. Також на цьому етапі розроблялися сценарії користувацьких шляхів (User Flow) від моменту вибору товару до завершення транзакції.

Другий етап — проєктування структури даних. Замість використання стандартних реляційних схем, було прийнято рішення застосувати документ-орієнтовану модель MongoDB. Це дозволило гнучко налаштувати структуру об'єктів товарів, враховуючи варіативність одягу (розміри, кольори, артикули), що критично важливо для специфіки інтернет-магазину.

На етапі програмної імплементації реалізовувалася інтеграція клієнтської частини з API. Було розроблено систему автентифікації користувачів, впроваджено функціонал «розумного кошика» та налаштовано сервіси для автоматичного оновлення складських залишків. Основною задачею була синхронізація станів клієнтського інтерфейсу та серверних даних.

Фаза тестування та відладки передбачала перевірку цілісності даних. Особлива увага приділялася сценаріям одночасного доступу до товарних залишків, щоб уникнути виникнення ситуацій «race condition» при оформленні замовлень. Також проводилася перевірка коректності роботи фільтрів у каталозі та стабільності роботи системи при різній інтенсивності запитів.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 15 |

Заключний етап — інфраструктурне розгортання. Він включав вибір хостингу, налаштування середовища Node.js, конфігурацію веб-сервера та запуск механізмів постійного моніторингу працездатності системи. Після перевірки всіх параметрів застосунк було переведено в стан повної функціональної готовності.

1.2.7 Порядок тестування та прийому

Процес перевірки вебзастосунку «SwagStore» побудований таким чином, щоб гарантувати надійність системи на кожному етапі її створення. Ретельний контроль якості дозволяє виявити та усунути програмні помилки ще до того, як ресурс стане доступним для реальних покупців.

Тестування складається з кількох послідовних етапів. Спочатку перевіряється робота окремих функціональних модулів серверної та клієнтської частини. Далі проводиться комплексна перевірка, під час якої оцінюється взаємодія між клієнтською частиною та API, зокрема авторизація через JWT і обробка запитів до бази MongoDB. Також виконується наскрізна перевірка основних бізнес-процесів, зокрема оформлення замовлень та автоматичне оновлення залишків на складі. На завершення проводиться перевірка стабільності роботи основних сценаріїв вебзастосунку.

Окремо під час перевірки враховується послідовність виконання дій у системі. Це дає змогу переконатися, що користувач може пройти основний шлях від вибору товару до створення замовлення, а адміністратор — переглянути та опрацювати це замовлення в адміністративній панелі.

Приймання результатів розробки проводиться шляхом порівняння готового вебзастосунку з вимогами технічного завдання. Комісія перевіряє працездатність усіх модулів, зручність інтерфейсу та швидкість реакції системи. У разі відповідності розробленого продукту всім критеріям, він вважається придатним до використання, про що свідчить підписаний акт про введення в експлуатацію.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 16 |

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури вебзастосунку

Для створення вебзастосунку інтернет-магазину одягу «SwagStore» було спроектовано загальну структуру системи, яка поєднує публічну частину для покупців, адміністративну панель для керування магазином, серверну логіку та базу даних. На етапі проєктування було визначено основні ролі користувачів, функціональні модулі, порядок взаємодії між клієнтською та серверною частинами, а також спосіб збереження даних.

Вебзастосунок SwagStore складається з клієнтської частини, серверної частини та бази даних. Клієнтська частина реалізована з використанням Next.js, React і TypeScript та відповідає за відображення сторінок інтернет-магазину, каталогу товарів, пошуку, фільтрів, сторінки товару, кошика, оформлення замовлення, профілю користувача, історії замовлень, відгуків, форми підтримки та адміністративної панелі. Серверна частина побудована на основі Node.js, Express.js і TypeScript. Вона обробляє HTTP-запити, виконує авторизацію користувачів, перевіряє права доступу та забезпечує роботу з товарами, категоріями, складськими залишками, замовленнями, відгуками й зверненнями користувачів.

Збереження інформації здійснюється в базі даних MongoDB. Для взаємодії серверної частини з базою даних використовується бібліотека Mongoose. У базі даних зберігаються відомості про користувачів, товари, категорії, складські залишки, історію змін складу, замовлення, відгуки та звернення до підтримки. Для авторизації користувачів використовується JWT-токен, а дані кошика на стороні клієнта зберігаються в localStorage. Загальну структурну схему вебзастосунку SwagStore подано на рисунку 2.1.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| | | | | | | 17 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Структурна схема головної сторінки сайту

SwagStore

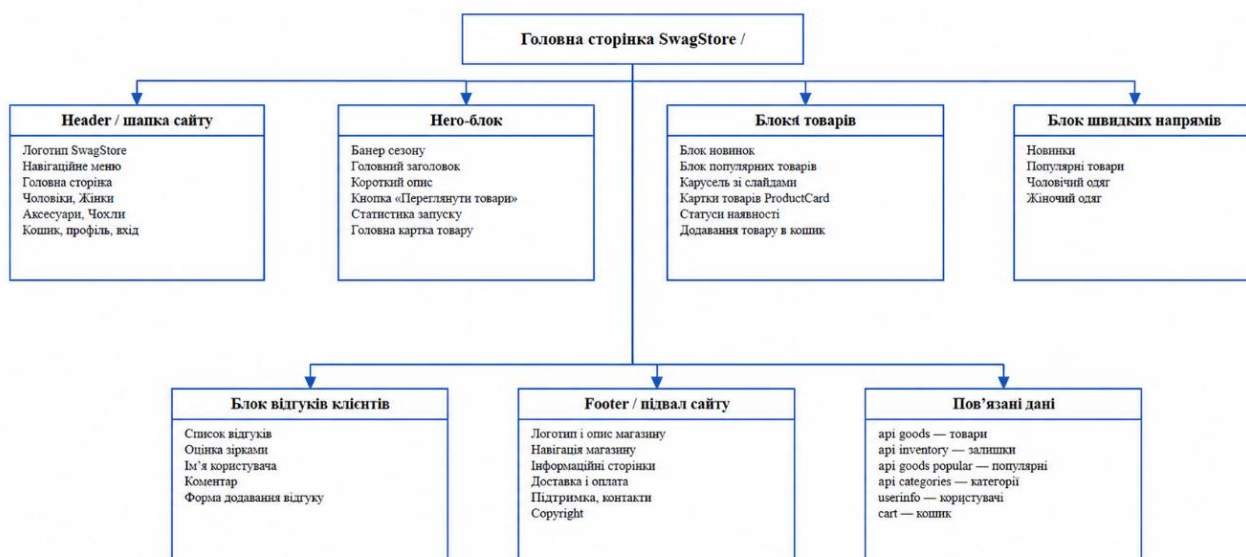


Рисунок 2.1 — Структурна схема головної сторінки сайту

Як видно з рисунка 2.1, користувач і адміністратор взаємодіють із системою через клієнтську частину вебзастосунку. Звичайний користувач працює з публічною частиною сайту, каталогом, пошуком, фільтрами, кошиком, оформленням замовлення, профілем та власними замовленнями. Адміністратор використовує адміністративну панель для керування товарами, категоріями, складськими залишками, замовленнями, клієнтами, відгуками та зверненнями користувачів.

Обмін даними між клієнтською та серверною частиною здійснюється через REST API за допомогою HTTP-запитів. Серверна частина приймає запити, виконує необхідну обробку даних, перевіряє авторизацію та звертається до бази даних MongoDB через Mongoose. Така структура дозволяє логічно розділити інтерфейс користувача, серверну логіку та збереження даних, що спрощує підтримку вебзастосунку і створює можливість для подальшого розширення його функціоналу.

Під час проєктування структури також було враховано зручність навігації для користувача. Основні сторінки вебзастосунку розміщені таким чином, щоб покупець міг швидко перейти від перегляду каталогу до сторінки товару, додати

товар до кошика та оформити замовлення. Адміністративна частина, у свою чергу, відокремлена від публічного інтерфейсу та доступна лише користувачам із відповідною роллю. Це забезпечує зрозумілу організацію системи та захист адміністративних функцій від несанкціонованого доступу.

2.2 Створення та верстка сторінок вебзастосунку

Процес створення та верстки інтерфейсу користувача вебзастосунку «SwagStore» базується на сучасних концепціях вебдизайну, забезпеченні високої швидкодії та повної адаптивності сторінок. Фронтенд-частину системи реалізовано з використанням бібліотеки React (версії 19) у середовищі фреймворку Next.js (версії 15.2.4), що дозволило застосувати компонентний підхід до побудови інтерфейсу. Таке рішення забезпечує декомпозицію складного графічного середовища інтернет-магазину на окремі, незалежні та повторно використовувані блоки (компоненти), що значно оптимізує структуру коду та спрощує його подальшу підтримку.

Розробка візуальної складової розпочалася з формування базового макета (Layout), який є наскрізним для всіх розділів вебзастосунку. До структури макета інтегровано спільні елементи керування: верхню навігаційну панель (Header) з категоріями товарів, пошуковим рядком та модулем кошика, а також інформаційний підвал (Footer). Використання механізму App Router у Next.js дозволило реалізувати оптимізоване завантаження контенту без повного перезапиту сторінок, що покращує користувацький досвід (UX).

Для стилізації інтерфейсу та побудови графічних елементів застосовано фреймворк Tailwind CSS (версії 3.4). Завдяки утилітарному підходу до верстки, стилі описуються безпосередньо всередині компонентів за допомогою зумовлених класів, що унеможливорює накопичення зайвого CSS-коду та прискорює рендеринг сторінок у браузері. Для забезпечення відповідності елементів інтерфейсу (кнопок, діалогових вікон, випадаючих списків, селекторів розмірів)

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 19 |

стандартам доступності (Accessibility) та сучасним вимогам дизайну, було інтегровано набір компонентів `shadcn/ui`, побудованих на базі примітивів Radix UI.

Важливим етапом розробки стало забезпечення повної адаптивності вебінтерфейсу (Responsive Web Design) під різні роздільні здатності екранів — від мобільних пристроїв до широкоформатних моніторів. Адаптивність реалізована за допомогою адаптивних модифікаторів Tailwind CSS (breakpoint modifiers). Наприклад, сітка товарів (Product Grid) динамічно змінює кількість колонок залежно від ширини екрана користувача, а головна навігаційна панель на мобільних пристроях трансформується у компактне висувне меню («гамбургер-меню»).

Для забезпечення динамічної взаємодії з користувачем на сторінках каталогу та оформлення замовлення впроваджено клієнтську валідацію форм. Вона базується на зв'язці бібліотек `react-hook-form` та `zod`, що дозволяє перевіряти коректність введених покупцем даних (формат номера телефону, наявність символу «@» в email тощо) в реальному часі до моменту відправки запиту на сервер. Для візуалізації стану завантаження товарів із бази даних (поки очікується відповідь від API) використовуються скелетони (Skeleton компоненти), які імітують структуру карток одягу, зменшуючи візуальний шум та підвищуючи комфорт взаємодії з вебзастосунком.

Зовнішній вигляд розроблених сторінок каталогу товарів та інтерфейсу користувача наведено у додатках.

2.3 Розробка структури бази даних

Для ефективного управління даними вебзастосунку «SwagStore» було спроектовано модель даних, що базується на документо-орієнтованому підході. Вибір системи управління базами даних MongoDB зумовлений потребою у високій гнучкості структури товарів, які мають складні характеристики, що важко реалізувати в межах жорстких реляційних схем.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 20 |

Проектування бази даних здійснювалося з урахуванням принципів мінімізації надмірності даних та забезпечення швидкого доступу до інформації. Для реалізації моделі даних використано ODM-бібліотеку Mongoose, що дозволяє описувати схеми документів на рівні коду, забезпечуючи автоматичну валідацію вхідної інформації перед її збереженням у систему. Це гарантує цілісність даних та унеможливорює потрапляння до бази некоректних записів.

Структура бази даних представлена колекціями, що логічно пов'язані між собою через ідентифікатори. Кожна колекція має первинний ключ `_id` типу `ObjectId`.

Колекція `users` зберігає дані про клієнтів та адміністраторів і містить наступні поля:

- `_id` – унікальний ідентифікатор;
- `name` – ім'я користувача;
- `email` – електронна пошта (унікальний індекс);
- `passwordHash` – хешований пароль;
- `role` – роль в системі (`user/admin`);
- `phone`, `city`, `deliveryAddress` – контактні дані та адреса;
- `createdAt` – дата реєстрації.

Колекція `goods` зберігає дані про асортимент одягу та містить поля:

- `_id` – ідентифікатор;
- `id` – бізнес-ідентифікатор товару;
- `name` – назва виробу;
- `section` – розділ (чоловіки/жінки/аксесуари);
- `category` – категорія товару;
- `brand` – бренд;
- `price` – ціна;
- `sizes`, `colors` – масиви характеристик;
- `image`, `images` – посилання на зображення.

Колекція `inventories` забезпечує складський облік і містить:

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 21 |

- `_id` – ідентифікатор;
- `goodId` – зовнішній ключ товару;
- `quantity` – кількість на складі;
- `warehouse` – назва складу;
- `color, size` – параметри варіанту товару.

Колекція `inventoryhistories` зберігає історію операцій зі складом (надходження/продаж/списання), що дозволяє відстежувати рух матеріальних цінностей.

Колекція `orders` об'єднує дані покупця та замовлені позиції:

- `_id` – ідентифікатор замовлення;
- `userId` – посилання на користувача;
- `items` – масив `OrderItem` (зріз цін та характеристик на момент покупки);
- `totalPrice` – загальна сума;
- `status` – статус (Активне/Доставляється/Реалізоване).

Колекція `categories` організовує каталог, містить назви та SEO-параметри (`slug`) для коректної навігації. Колекції `reviews` та `supportmessages` забезпечують зворотний зв'язок та підтримку користувачів.

ER-діаграму БД для даного вебзастосунку зображено в графічній частині 2026.КВР.122.421.14.00.00 БД.

2.4 Програмування вебзастосунку

Програмування вебзастосунку «SwagStore» виконувалося з використанням сучасного стеку вебтехнологій, який дозволяє розділити систему на клієнтську та серверну частини. Такий підхід забезпечує зручну організацію коду, спрощує тестування, дає змогу окремо розвивати інтерфейс користувача та серверну логіку, а також підвищує гнучкість подальшої підтримки проєкту.

Клієнтська частина вебзастосунку реалізована за допомогою Next.js, React та TypeScript. Вона відповідає за відображення сторінок інтернет-магазину, взаємодію користувача з каталогом товарів, пошуком, фільтрами, кошиком,

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| | | | | | | 22 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

профілем, оформленням замовлення та адміністративною панеллю. Використання компонентного підходу дало змогу поділити інтерфейс на окремі логічні блоки, які можна повторно використовувати на різних сторінках вебзастосунку.

Серверна частина розроблена з використанням Node.js, Express та TypeScript. Вона відповідає за обробку HTTP-запитів, взаємодію з базою даних MongoDB, авторизацію користувачів, роботу з товарами, категоріями, складськими залишками, замовленнями, відгуками та зверненнями в підтримку. Для обміну даними між клієнтською та серверною частинами використовується REST API, а інформація передається у форматі JSON.

Для збереження даних використовується MongoDB, а взаємодія з базою даних організована за допомогою бібліотеки Mongoose. У проєкті створено окремі моделі для користувачів, товарів, складських залишків, історії змін на складі, замовлень, категорій, відгуків та звернень користувачів. Така структура дозволяє логічно розділити дані за призначенням і забезпечити зручну роботу з ними на серверному рівні.

У процесі програмування було реалізовано механізм реєстрації та входу користувачів, розмежування доступу між звичайним користувачем і адміністратором, перегляд каталогу товарів, пошук і фільтрацію, додавання товарів до кошика, вибір розміру, оформлення замовлення та автоматичне оновлення залишків на складі. Окремо реалізовано адміністративну частину, яка надає можливість керувати товарами, категоріями, замовленнями, клієнтами, відгуками та зверненнями в підтримку.

Під час розробки також було враховано питання зручності користування та стабільності роботи системи. Клієнтська частина забезпечує зрозумілий інтерфейс для покупця, а серверна частина виконує перевірку даних і контролює коректність основних операцій. Завдяки цьому вебзастосунок «SwagStore» може виконувати роль як онлайн-вітрини магазину одягу, так і системи внутрішнього обліку товарів.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 23 |

2.4.1 Написання клієнтської частини

Клієнтська частина вебзастосунку «SwagStore» була реалізована з використанням фреймворку Next.js та бібліотеки React. Основним завданням клієнтської частини є забезпечення зручної взаємодії користувача з інтернет-магазином, відображення товарів, обробка дій покупця, передавання запитів до серверної частини та виведення отриманих даних у зрозумілому вигляді.

Для розробки інтерфейсу було використано компонентний підхід. Кожна частина сторінки представлена окремим компонентом, який відповідає за конкретну функцію або візуальний блок. Такий підхід дозволяє повторно використовувати однакові елементи на різних сторінках, спрощує підтримку коду та робить структуру проєкту більш зрозумілою. Наприклад, окремими компонентами реалізовано картки товарів, кнопки, форми, модальні вікна, елементи навігації, фільтри та блоки адміністративної панелі.

Основними сторінками клієнтської частини є головна сторінка, каталог товарів, сторінки категорій, сторінка пошуку, сторінка детального перегляду товару, кошик, сторінка оформлення замовлення, профіль користувача, сторінка історії замовлень та сторінки адміністративної частини. Головна сторінка виконує роль візитної сторінки магазину, де користувач може швидко перейти до каталогу, переглянути новинки та основні розділи магазину. Каталог товарів містить перелік позицій із можливістю фільтрації та сортування.

Сторінка товару містить детальну інформацію про обраний товар: назву, бренд, ціну, категорію, опис, доступні розміри, колір та зображення. На цій сторінці користувач може вибрати потрібний розмір і додати товар до кошика. Це дозволяє уникнути ситуацій, коли товар додається без необхідних характеристик, що є особливо важливим для магазину одягу.

Кошик реалізовано як окремий функціональний модуль клієнтської частини. У ньому користувач може переглянути вибрані товари, змінити їхню кількість, видалити непотрібні позиції або перейти до оформлення замовлення. Дані кошика зберігаються на стороні клієнта, що дозволяє зберігати вибрані

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 24 |

товари під час переходу між сторінками. При цьому під час оформлення замовлення серверна частина додатково перевіряє наявність товарів на складі та коректність підсумкової суми.

Сторінка оформлення замовлення містить форму для введення контактних даних і адреси доставки. Для підвищення зручності користування передбачено валідацію введених даних, зокрема перевірку електронної пошти, номера телефону та поштового індексу. Після успішного заповнення форми користувач може підтвердити замовлення, після чого дані передаються на серверну частину через API.

У клієнтській частині також реалізовано механізм авторизації користувачів. Після входу в систему користувач отримує доступ до профілю, історії власних замовлень і можливості залишати відгуки. Токен авторизації зберігається у локальному сховищі браузера та використовується для виконання запитів, які потребують підтвердження особи користувача. Якщо користувач має роль адміністратора, йому відкривається доступ до адміністративної панелі.

Адміністративна частина клієнтського інтерфейсу призначена для керування основними даними вебзастосунку. Через неї адміністратор може додавати та редагувати товари, керувати категоріями, переглядати залишки на складі, змінювати статуси замовлень, переглядати клієнтів, відгуки та звернення в підтримку. Інтерфейс адміністратора побудовано таким чином, щоб основні операції виконувалися швидко та не потребували роботи безпосередньо з базою даних.

Для стилізації сторінок було використано Tailwind CSS, що дозволило швидко створювати адаптивні та охайні елементи інтерфейсу. Завдяки адаптивній верстці вебзастосунок коректно відображається на різних пристроях, включаючи персональні комп'ютери, ноутбуки, планшети та смартфони. Це є важливим для інтернет-магазину, оскільки користувачі можуть переглядати товари та оформлювати замовлення з різних типів пристроїв.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 25 |

Обмін даними між клієнтською та серверною частинами здійснюється за допомогою HTTP-запитів до REST API. Клієнтська частина надсилає запити для отримання товарів, категорій, залишків, замовлень, відгуків та інших даних, а сервер повертає відповіді у форматі JSON. Такий формат є зручним для обробки у JavaScript/TypeScript і дозволяє швидко оновлювати інформацію на сторінках без ускладнення структури застосунку.

У межах дипломного проєкту реалізовано демонстраційний механізм вибору способу оплати. Такий підхід дозволяє перевірити повний сценарій оформлення замовлення без підключення реального платіжного сервісу. Користувач може обрати спосіб оплати під час оформлення замовлення, після чого інформація про вибраний варіант передається на серверну частину та зберігається разом із даними замовлення.

Таким чином, клієнтська частина вебзастосунку «SwagStore» забезпечує повноцінну взаємодію користувача з інтернет-магазином, поєднуючи зручний інтерфейс для покупців і функціональну адміністративну панель для керування магазином.

2.4.2 Написання серверної частини

Серверна частина вебзастосунку «SwagStore» реалізована з використанням Node.js, Express та TypeScript. Вона відповідає за обробку запитів від клієнтської частини, взаємодію з базою даних MongoDB, перевірку даних, авторизацію користувачів та виконання основної бізнес-логіки інтернет-магазину. Завдяки використанню серверної частини вебзастосунок може не лише відображати товари користувачам, а й забезпечувати збереження даних, обробку замовлень, контроль залишків і роботу адміністративної панелі.

Основою серверної частини є фреймворк Express, який використовується для створення REST API. За допомогою API клієнтська частина надсилає запити до сервера для отримання товарів, категорій, замовлень, відгуків, звернень у підтримку та іншої інформації. Відповіді сервера передаються у форматі JSON,

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 26 |

що є зручним для подальшої обробки на стороні клієнта. Такий підхід дозволяє чітко розділити логіку інтерфейсу та логіку обробки даних.

У серверній частині створено окремі маршрути для основних функціональних модулів вебзастосунку. Маршрут `/api/auth` відповідає за реєстрацію, вхід користувачів і отримання даних профілю. Маршрут `/api/goods` використовується для роботи з товарами, зокрема отримання списку товарів, додавання, редагування та видалення записів. Маршрут `/api/inventory` відповідає за складський облік і контроль залишків. Для роботи із замовленнями використовується маршрут `/api/orders`, а для категорій — `/api/categories`. Окремо реалізовано маршрути для відгуків, адміністративної статистики та звернень користувачів у підтримку.

Для збереження даних використовується база даних MongoDB. Взаємодія серверної частини з базою даних організована за допомогою бібліотеки Mongoose. У проєкті створено окремі моделі для користувачів, товарів, складських залишків, історії змін залишків, замовлень, категорій, відгуків та звернень у підтримку. Кожна модель описує структуру відповідної колекції, типи полів, обов'язкові значення та додаткові обмеження. Це дозволяє перевіряти дані ще до їх збереження в базу та зменшує ризик появи некоректної інформації.

Модуль авторизації забезпечує реєстрацію та вхід користувачів. Під час реєстрації сервер перевіряє правильність введених даних, унікальність електронної пошти та виконує хешування пароля за допомогою `bcrypt`. Збереження паролів у вигляді хешу підвищує безпеку системи, оскільки справжній пароль користувача не зберігається у відкритому вигляді. Після успішного входу користувач отримує JWT-токен, який використовується для підтвердження особи під час подальших запитів до захищених маршрутів.

У системі реалізовано розмежування доступу за ролями. Звичайний користувач має доступ до перегляду каталогу, кошика, оформлення замовлень, профілю, історії замовлень і відгуків. Адміністратор має розширені права та може керувати товарами, категоріями, залишками на складі, замовленнями, клієнтами,

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 27 |

відгуками та зверненнями в підтримку. Такий підхід дозволяє захистити адміністративні функції від несанкціонованого доступу.

Важливою частиною серверної логіки є робота з товарами. Для кожного товару зберігаються назва, бізнес-ідентифікатор, розділ, категорія, бренд, опис, ціна, доступні розміри, кольори, головне зображення та додаткові зображення. Це дозволяє формувати повноцінний каталог товарів для клієнтської частини та забезпечує адміністратору можливість гнучко керувати асортиментом магазину.

Окремий модуль відповідає за складський облік. Для кожного товару може зберігатися інформація про кількість на складі, склад розміщення, колір і розмір. Така структура є важливою саме для магазину одягу, оскільки один і той самий товар може мати різні варіанти за розміром і кольором. Крім поточного залишку, у системі передбачено збереження історії змін складських даних, що дозволяє відстежувати надходження, списання та інші операції.

Під час оформлення замовлення серверна частина виконує перевірку товарів, які користувач додав до кошика. Сервер повторно обчислює суму замовлення, перевіряє наявність потрібної кількості товарів на складі та лише після цього створює запис замовлення у базі даних. Такий підхід є важливим, оскільки дані на клієнтській стороні можуть змінюватися, тому остаточну перевірку необхідно виконувати саме на сервері.

Після успішного оформлення замовлення відбувається списання відповідної кількості товарів зі складу. Якщо під час перевірки виявляється, що потрібної кількості товару немає, сервер не дозволяє завершити оформлення замовлення та повертає повідомлення про помилку. Це дозволяє уникнути продажу товарів, яких фактично немає в наявності, і забезпечує коректність складського обліку.

Модуль замовлень зберігає інформацію про користувача, контактні дані, адресу доставки, коментар до замовлення, список товарів, підсумкову суму, вартість доставки, спосіб оплати, статус оплати та статус самого замовлення. Для адміністратора передбачена можливість переглядати замовлення та змінювати

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 28 |

їхній статус, наприклад «Активне», «Доставляється» або «Реалізоване». Це дозволяє контролювати процес виконання замовлення від моменту оформлення до завершення.

Також на серверній частині реалізовано роботу з відгуками та зверненнями користувачів. Відгуки дають можливість клієнтам залишати оцінку та коментар щодо роботи магазину, а звернення в підтримку дозволяють користувачам надсилати повідомлення адміністрації. Для таких звернень передбачено статуси обробки, що спрощує роботу адміністратора із запитами клієнтів.

Таким чином, серверна частина вебзастосунку «SwagStore» забезпечує основну логіку роботи системи, включаючи авторизацію користувачів, керування товарами, складський облік, оформлення замовлень, роботу з відгуками та підтримкою. Завдяки використанню Node.js, Express, MongoDB і Mongoose серверна частина є гнучкою, масштабованою та придатною для подальшого розширення функціональних можливостей вебзастосунку.

Для реалізації структури бази даних у серверній частині використано окремі моделі Mongoose, розміщені у каталозі server/src/models. Кожна модель відповідає окремій колекції бази даних MongoDB та описує структуру документів, типи полів і основні обмеження. Такий підхід дозволяє зберігати логіку роботи з даними в окремих файлах і не змішувати її з маршрутами або іншими частинами серверної логіки. Використання окремих моделей також спрощує перевірку даних перед їх збереженням у базі та зменшує ризик появи некоректних записів у системі. Завдяки такій організації серверна частина має зрозумілу структуру, а робота з користувачами, товарами, замовленнями та складськими залишками виконується через окремі логічні сутності. Завдяки цьому структура проєкту залишається зрозумілою, а внесення змін до окремих сутностей, наприклад товарів, замовлень або складських залишків, не впливає на інші модулі вебзастосунку. Крім того, така структура полегшує пошук помилок під час розробки та подальшого супроводу системи. Структуру моделей бази даних подано на рисунку 2.2.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 29 |

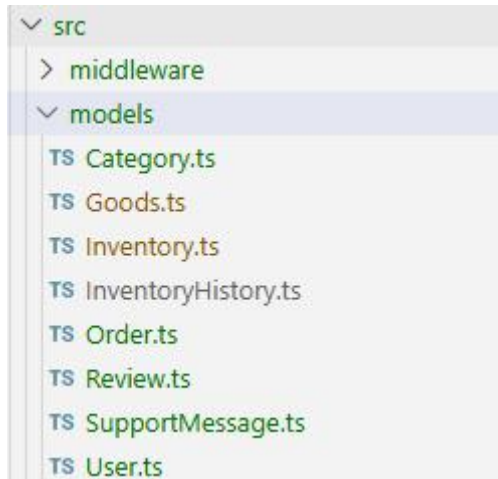


Рисунок 2.2 — Структура моделей бази даних

Для організації серверної логіки використано окремі файли маршрутів, розміщені у каталозі `server/src/routes`. Кожен маршрут відповідає за окремий функціональний модуль системи та обробляє відповідні HTTP-запити від клієнтської частини. Структуру API-маршрутів серверної частини вебзастосунку подано на рисунку 2.3.

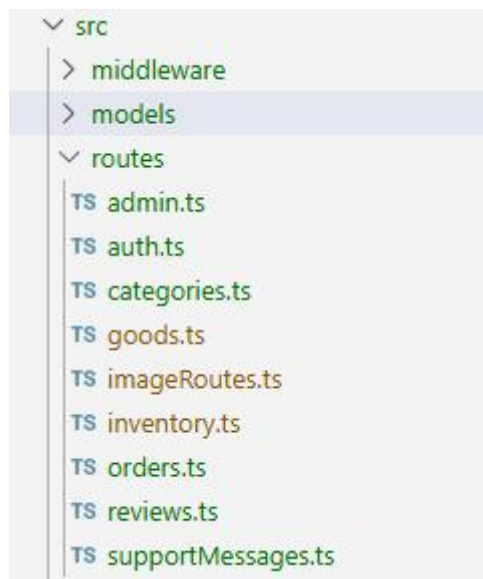


Рисунок 2.3 — Структура API-маршрутів серверної частини

Клієнтська частина вебзастосунку побудована на основі маршрутизації Next.js App Router. Основні сторінки та розділи розміщені у каталозі `client/app`, де

кожна папка відповідає окремому маршруту вебзастосунку. Така структура дозволяє логічно розділити публічні сторінки інтернет-магазину, кошик, оформлення замовлення, сторінки авторизації, профіль користувача та адміністративну частину. Структуру сторінок клієнтської частини подано на рисунку 2.4.

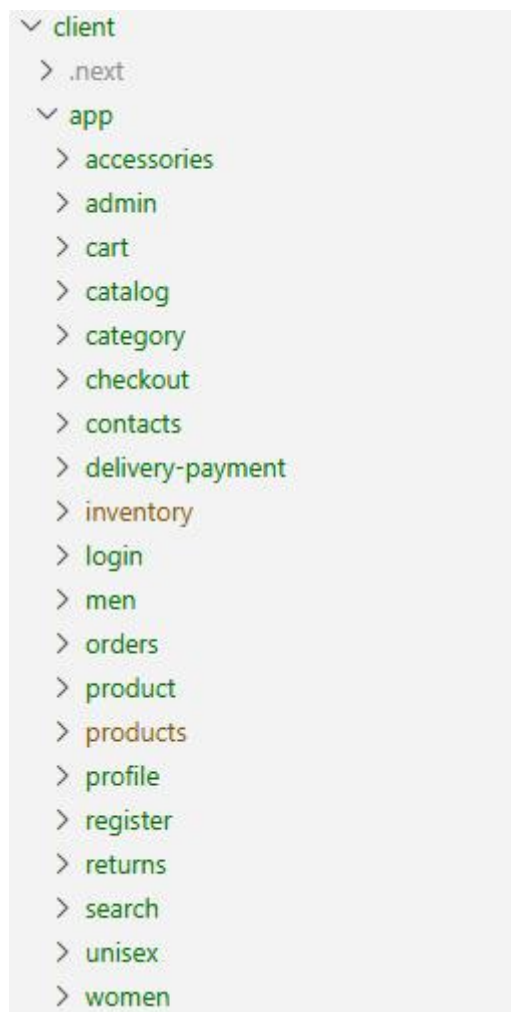


Рисунок 2.4 — Структура сторінок клієнтської частини

В адміністративній частині вебзастосунку реалізовано форму додавання нового товару. Вона дозволяє адміністратору вносити основну інформацію про товар, зокрема назву, ціну, розділ, категорію, опис, доступні розміри, колір та додаткові параметри відображення. Такий підхід спрощує наповнення каталогу та дає змогу керувати асортиментом магазину без безпосереднього редагування бази

даних. Форму додавання нового товару подано на рисунку 2.5.

Додати новий товар

Назва товару
Футболка Asics

Ціна
799

Розділ
Чоловіки

Категорія
Футболки

Бренд
Asics

Опис товару
Футболка ASICS — це універсальна модель, яка поєднує спортивний стиль, комфорт і функціональність для щоденної активності та легких тренувань

Розміри
S M L XL

Колір
Чорний

Показувати в новинках

Рисунок 2.5 — Форма додавання нового товару

Уся логіка вебзастосунку розподілена між клієнтською та серверною частинами. Клієнтська частина відповідає за відображення інтерфейсу, взаємодію користувача з каталогом, кошиком, формами та адміністративною панеллю. Серверна частина забезпечує обробку запитів, взаємодію з базою даних, авторизацію користувачів, роботу з товарами, категоріями, замовленнями, залишками, відгуками та зверненнями в підтримку.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 32 |

Файлова структура серверної частини побудована таким чином, щоб окремі елементи системи були логічно розділені за призначенням. Моделі бази даних зберігаються в окремому каталозі, маршрути API відповідають за обробку HTTP-запитів, а допоміжні файли забезпечують підключення до бази даних, перевірку авторизації та налаштування роботи сервера. Такий підхід спрощує підтримку проєкту, пошук потрібних файлів і подальше розширення функціональності вебзастосунку.

Загальну структуру серверної частини подано на рисунку 2.6.

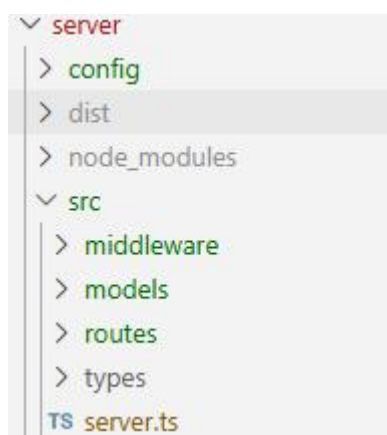


Рисунок 2.6 — Структура серверної частини

2.5 Тестування вебзастосунку

Після завершення основних етапів розробки було проведено перевірку працездатності вебзастосунку SwagStore. У процесі перевірки було проаналізовано основні сценарії роботи системи: відображення головної сторінки, роботу каталогу товарів, пошуку та фільтрації, перегляд сторінки товару, додавання товарів у кошик, оформлення замовлення, перегляд історії замовлень користувача та функціонування адміністративної панелі.

Окрему увагу було приділено перевірці адміністративних можливостей вебзастосунку, зокрема керуванню товарами, категоріями, складськими залишками, замовленнями, відгуками та зверненнями користувачів. Результати

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 33 |

перевірки основних сторінок і функціональних модулів вебзастосунку наведено на рисунках 2.7–2.21.

Головну сторінку вебзастосунку SwagStore подано на рисунку 2.7. Під час перевірки також було враховано коректність відображення основних блоків сторінки та переходів до каталогу товарів.

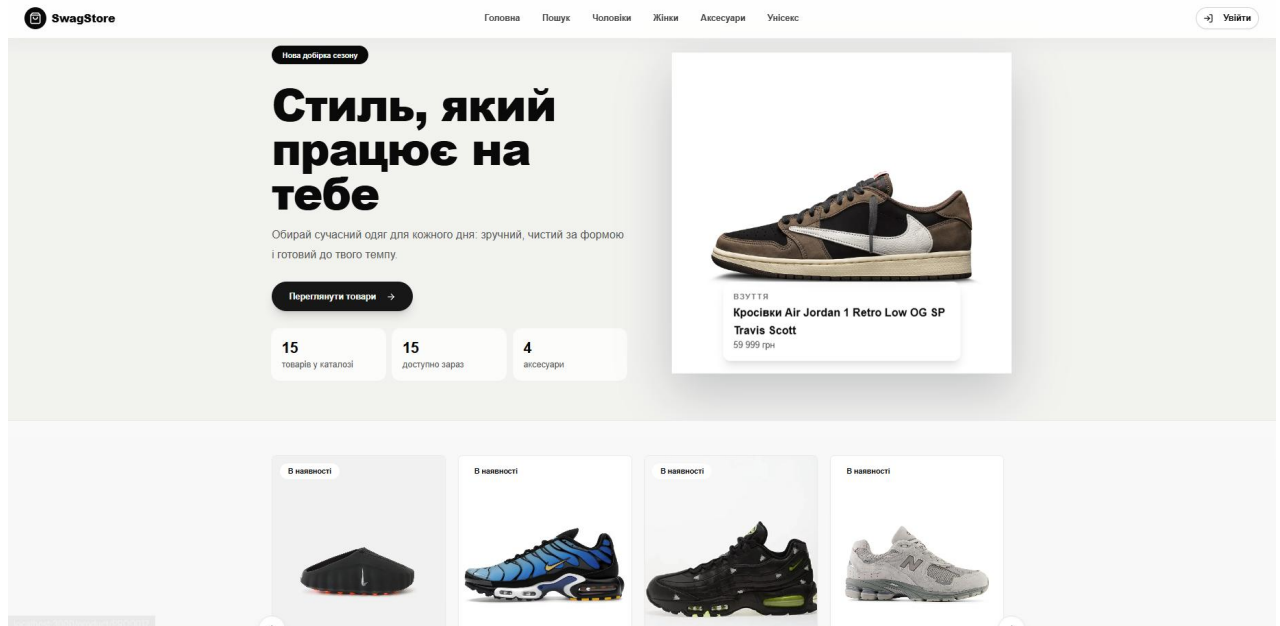


Рисунок 2.7 — Головна сторінка вебзастосунку SwagStore

Далі було протестовано каталог товарів. Перевірено відображення товарів, пошук, фільтрацію за основними параметрами, сортування та скидання фільтрів. Також було перевірено, що пошук і фільтри працюють узгоджено між собою та дозволяють швидко знайти потрібні товари в каталозі. Це підтверджує зручність використання сторінки каталогу для покупця. Окремо перевірено, що після зміни параметрів фільтрації список товарів оновлюється коректно та не порушує структуру карток. Крім того, перевірено збереження зрозумілого розташування елементів сторінки після оновлення списку товарів. Сторінку каталогу товарів подано на рисунку 2.8.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 34 |

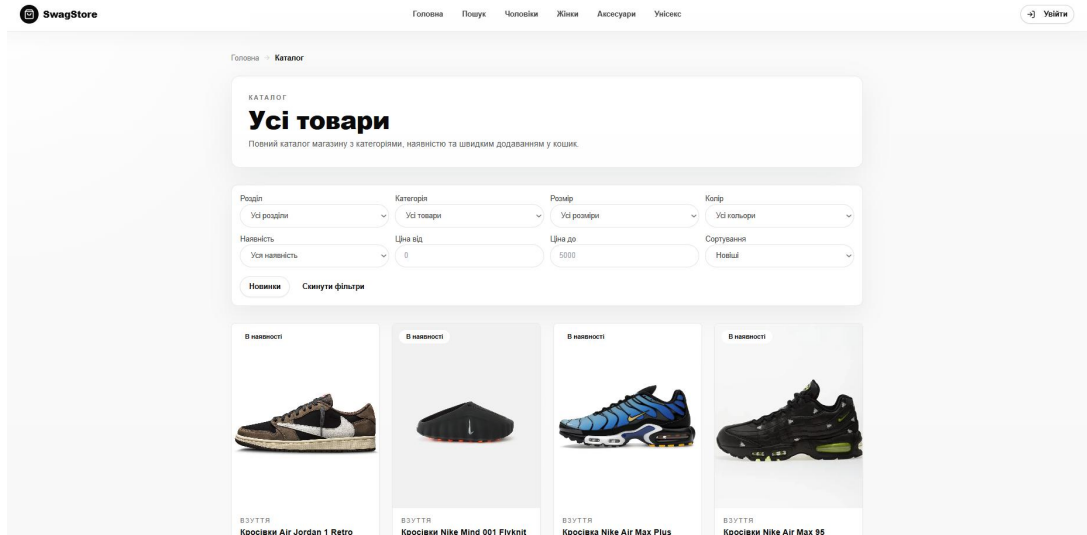


Рисунок 2.8 — Сторінка каталогу товарів

Окремо перевірено сторінку детального перегляду товару. Було протестовано відображення назви, бренду, ціни, опису, зображень, доступних розмірів, статусу наявності та додавання товару до кошика після вибору розміру. Додатково перевірено, що товар не додається до кошика без вибору необхідного розміру, оскільки для магазину одягу ця характеристика є обов'язковою. Така перевірка дозволила переконатися, що користувач отримує повну інформацію про товар перед його замовленням і може виконати дію покупки без помилок. Також було перевірено, що інформація на сторінці товару відповідає даним, які зберігаються в базі даних, зокрема назва, ціна, категорія, розміри та зображення. Сторінку детального перегляду товару подано на рисунку 2.9.

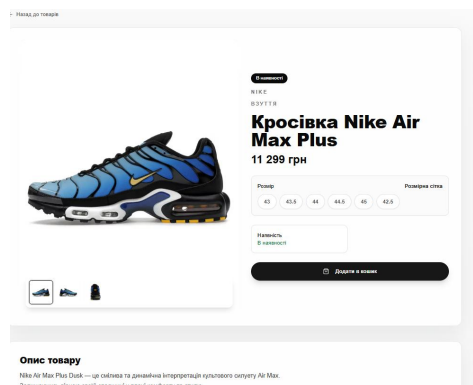


Рисунок 2.9 — Сторінка детального перегляду товару

Після цього перевірено роботу кошика: відображення доданих товарів, вибраного розміру, зміну кількості, видалення позицій і правильність підрахунку загальної суми. Додатково перевірено збереження даних кошика під час переходу між основними сторінками вебзастосунку, а також коректність оновлення підсумкової вартості після зміни кількості товарів. Це дозволило переконатися, що кошик працює стабільно та передає актуальні дані для подальшого оформлення замовлення. Окремо перевірено, що після видалення товару або зміни його кількості дані кошика оновлюються без перезавантаження сторінки та залишаються коректними для подальшого оформлення замовлення. Сторінку кошика користувача подано на рисунку 2.10.

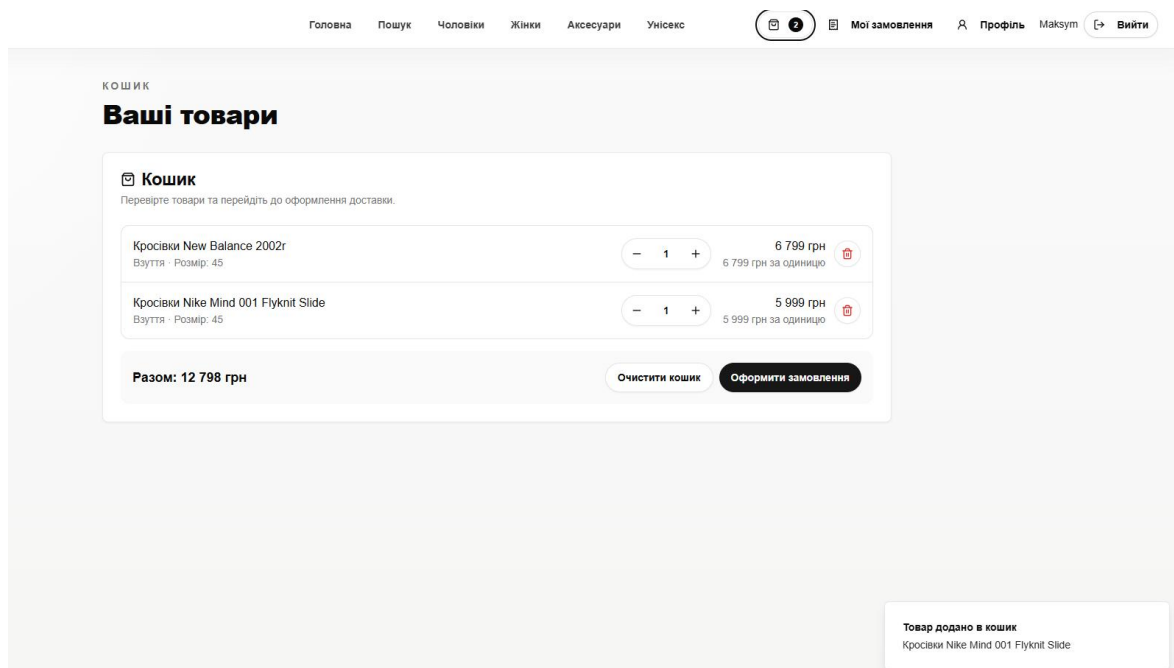


Рисунок 2.10 — Сторінка кошика користувача

На сторінці оформлення замовлення перевірено передавання товарів із кошика, відображення контактних даних, адреси доставки, вартості товарів і загальної суми. Також перевірено, що введені користувачем дані коректно використовуються під час формування замовлення. Сторінку оформлення замовлення подано на рисунку 2.11.

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 36 |

Дані для доставки
Заповніть контактні дані, щоб ми могли підготувати замовлення.

Контактні дані
Поля з іменем, телефоном та адресою є обов'язковими.

Ім'я отримувача
Максим

Номер телефону
+38073742045

Місто
Тернопіль

Адреса доставки
Коновальця 4, відділення 7

Поштовий індекс
46020

Коментар до замовлення
Побажання щодо доставки або товарів

Спосіб оплати

- Оплата при отриманні
Оплата під час отримання замовлення.
- Карткою онлайн (демо)
Демо-оплата буде позначена як оплачена.

Ваше замовлення
Перевірте склад замовлення перед підтвердженням.

| | |
|---|--------------------|
| Кросівки New Balance 2002r Ваутия - Розмір: 45 | 6 799 грн 1 шт. |
| Кросівки Nike Mind 001 Flyknit Slide Ваутия - Розмір: 45 | 5 999 грн 1 шт. |
| Товари | 12 798 грн |
| Доставка | 80 грн |
| До сплати | 12 878 грн |

Статус оплати буде "Очікує оплати" до моменту отримання.

Підтвердити замовлення

Рисунок 2.11 — Сторінка оформлення замовлення

У межах дипломного проекту реалізовано демонстраційний механізм вибору способу оплати. Такий підхід дозволяє перевірити повний сценарій оформлення замовлення без підключення реального платіжного сервісу. Додатково перевірено, що вибраний спосіб оплати зберігається разом із даними створеного замовлення. Демонстраційний вибір способу оплати подано на рисунку 2.12.

Коментар до замовлення
Побажання щодо доставки або товарів

Спосіб оплати

- Оплата при отриманні
Оплата під час отримання замовлення.
- Карткою онлайн (демо)
Демо-оплата буде позначена як оплачена.

Товари 12 798 грн

Доставка 80 грн

До сплати 12 878 грн

Статус оплати буде "Очікує оплати" до моменту отримання.

Підтвердити замовлення

Рисунок 2.12 — Демонстраційний вибір способу оплати

Після створення замовлення перевірено його збереження в базі даних і відображення в кабінеті користувача. Це дозволило переконатися, що користувач може переглядати актуальну інформацію про власні замовлення після завершення оформлення. Також перевірено, що в розділі замовлень відображаються основні дані про покупку: перелік товарів, підсумкова сума та поточний статус замовлення. Сторінку замовлень користувача подано на рисунку 2.13.

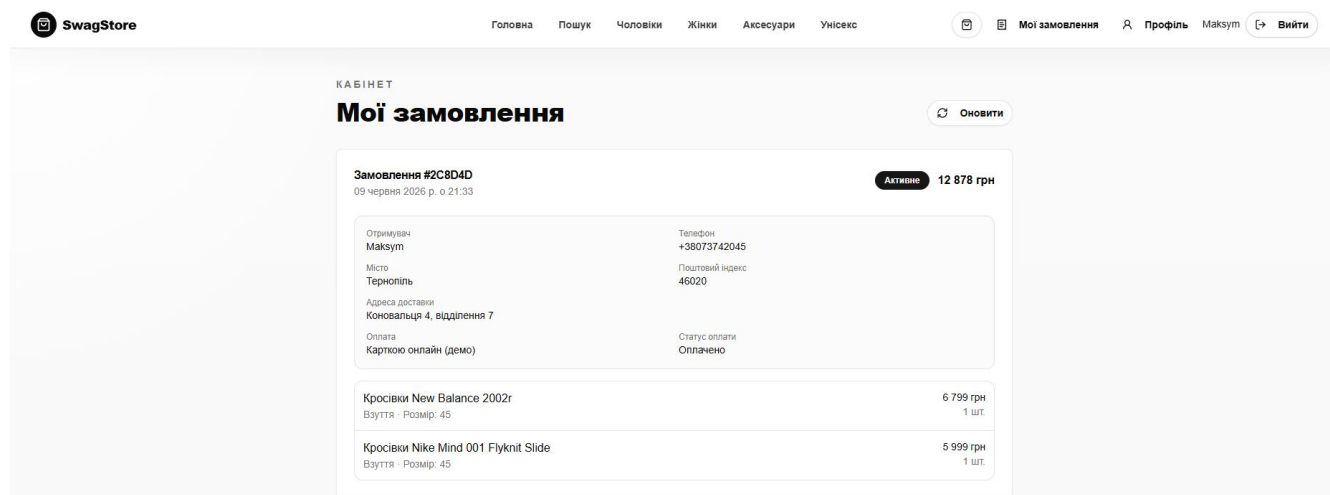


Рисунок 2.13 — Сторінка замовлень користувача

Адміністративну частину перевірено після входу користувача з роллю адміністратора. Було протестовано доступ до службових розділів і загальне відображення адміністративної панелі. Окремо перевірено, що звичайний користувач не має доступу до адміністративних функцій. Загальний вигляд адміністративної панелі подано на рисунку 2.14.

У розділі товарів перевірено завантаження списку товарів, перегляд основної інформації та доступ до операцій додавання, редагування й видалення товарних позицій. Також перевірено, що зміни, внесені адміністратором, відображаються в каталозі вебзастосунку. Це підтверджує узгоджену роботу адміністративної частини з клієнтським каталогом товарів. Завдяки цьому адміністратор може швидко вносити зміни до асортименту та одразу перевіряти результат їх застосування в системі.

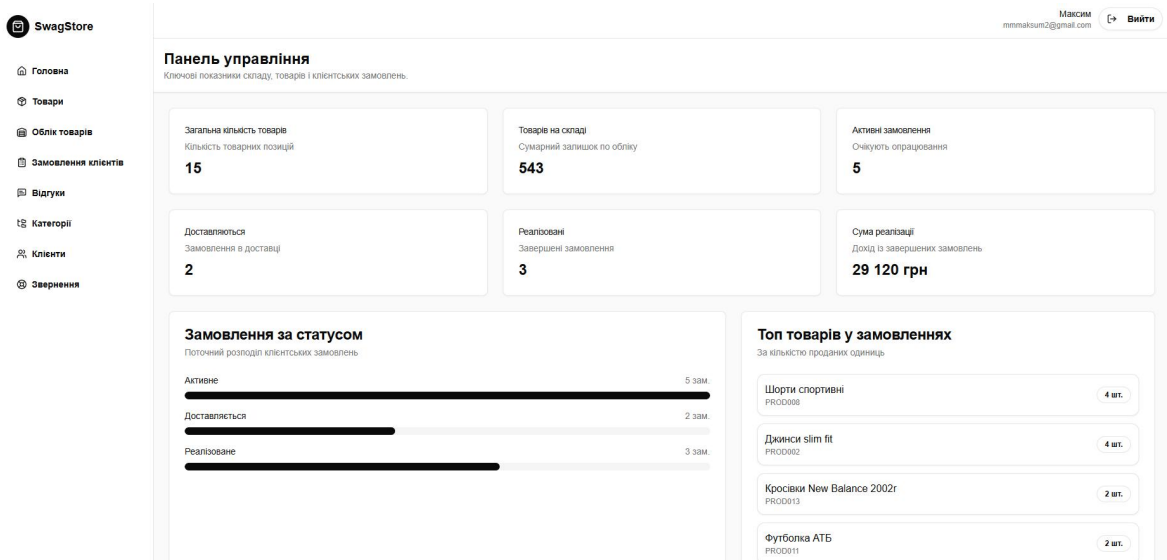


Рисунок 2.14 — Адміністративна панель SwagStore

Список товарів в адміністративній панелі подано на рисунку 2.15. На цій сторінці адміністратор може швидко переглянути основні дані товарів і перейти до керування окремими позиціями. Додатково перевірено, що товари відображаються у зрозумілому списку, а доступ до дій редагування та видалення розміщений безпосередньо біля відповідної товарної позиції. Це спрощує роботу адміністратора з каталогом і дозволяє швидко вносити зміни до асортименту магазину.

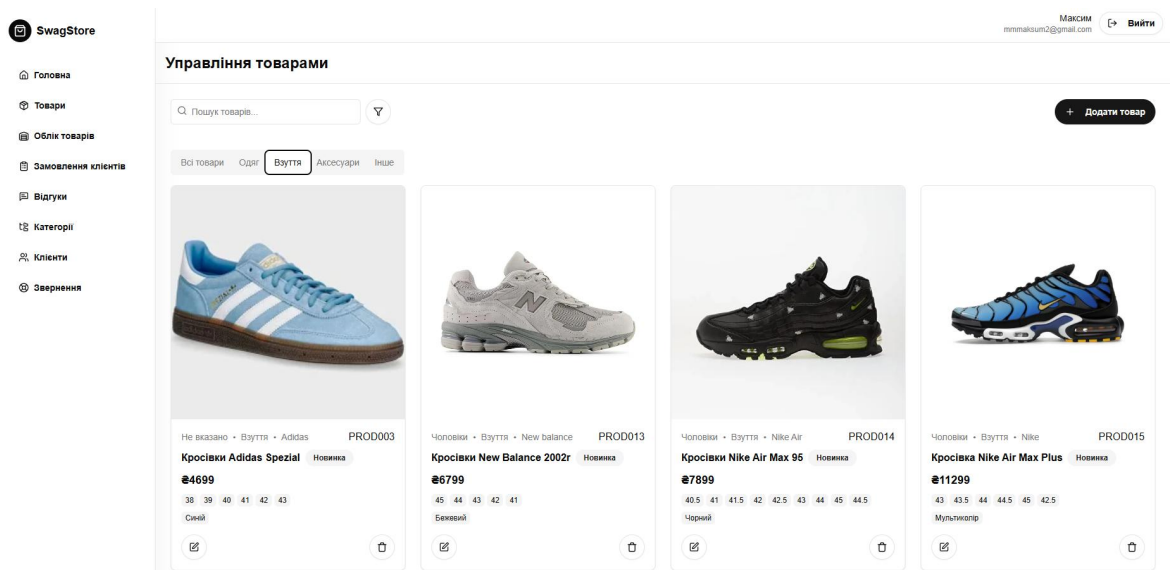


Рисунок 2.15 — Список товарів в адміністративній панелі

Також протестовано додавання нового товару через адміністративну панель. Було перевірено заповнення даних товару, збереження запису та його подальше відображення в адміністративній частині. Додатково перевірено коректність введення ціни, категорії, розміру, кольору та посилання на зображення товару. Результат додавання товару подано на рисунку 2.16.

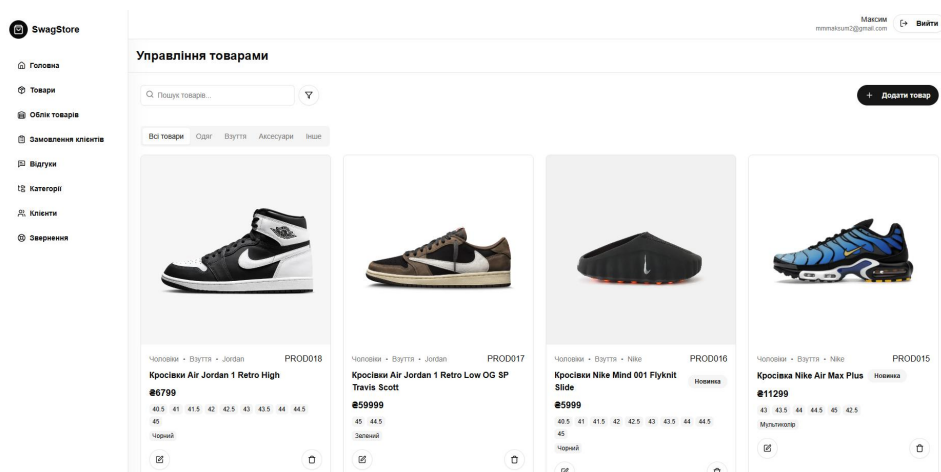


Рисунок 2.16 — Відображення доданого товару в адміністративній панелі

У розділі категорій перевірено перегляд, створення та редагування категорій, а також їх використання під час структурування каталогу. Це дало змогу переконатися, що категорії коректно застосовуються під час додавання товарів і фільтрації асортименту. Розділ керування категоріями подано на рисунку 2.17.

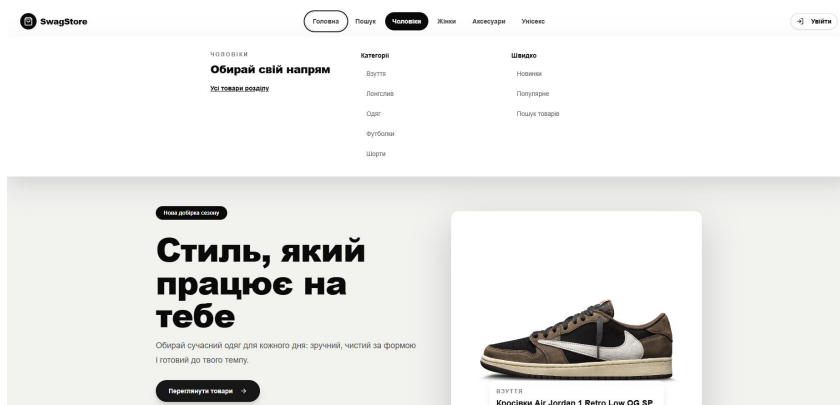


Рисунок 2.17 — Керування категоріями товарів

Наявність окремого розділу категорій спрощує подальше наповнення каталогу та дозволяє підтримувати зрозумілу структуру товарів. Адміністратор може швидко впорядковувати асортимент за розділами, а користувачі отримують зручнішу навігацію під час перегляду товарів. Також це важливо для роботи фільтрів, оскільки товари повинні коректно відображатися у відповідних категоріях після внесення змін в адміністративній панелі.

У складській частині перевірено редагування залишків за розмірами, кольорами та складським розміщенням. Також перевірено, що зміна кількості товару впливає на статус наявності в клієнтській частині. Особливу увагу приділено тому, щоб користувач не міг оформити замовлення на товар із недостатнім залишком. Така перевірка є важливою для уникнення ситуацій, коли користувач може замовити товар, якого фактично немає в наявності. Розділ керування складом подано на рисунку 2.18.

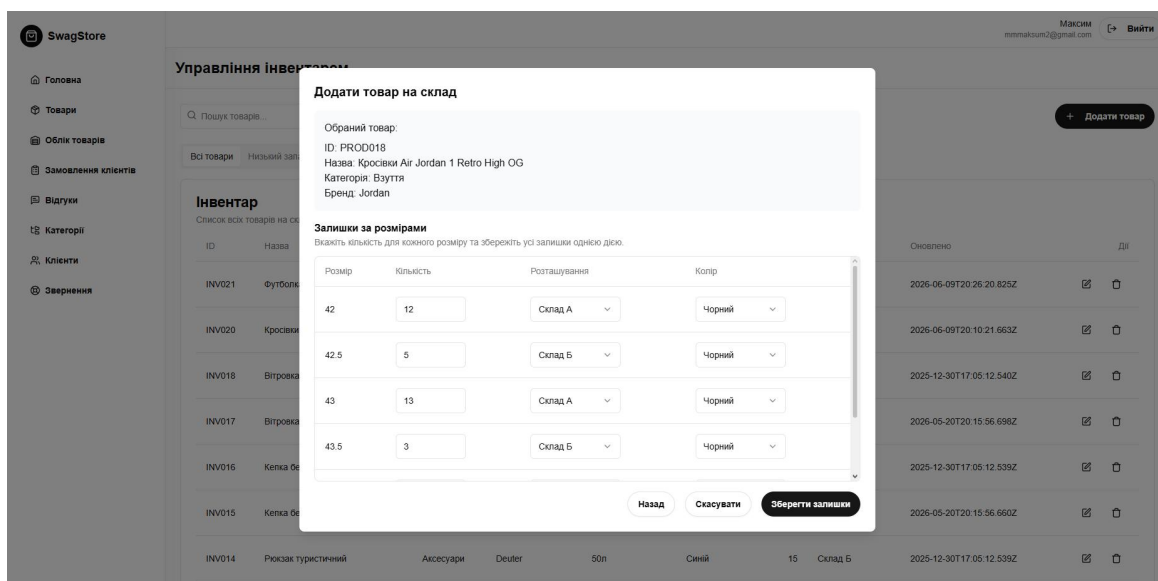


Рисунок 2.18 — Керування складом товарів

У розділі замовлень адміністратора перевірено перегляд оформлених замовлень, інформації про користувача, складу замовлення, суми та статусу. Також протестовано зміну статусу замовлення. Додатково перевірено, що

адміністратор може відстежувати етап виконання замовлення від моменту створення до завершення. Розділ керування замовленнями подано на рисунку 2.19.

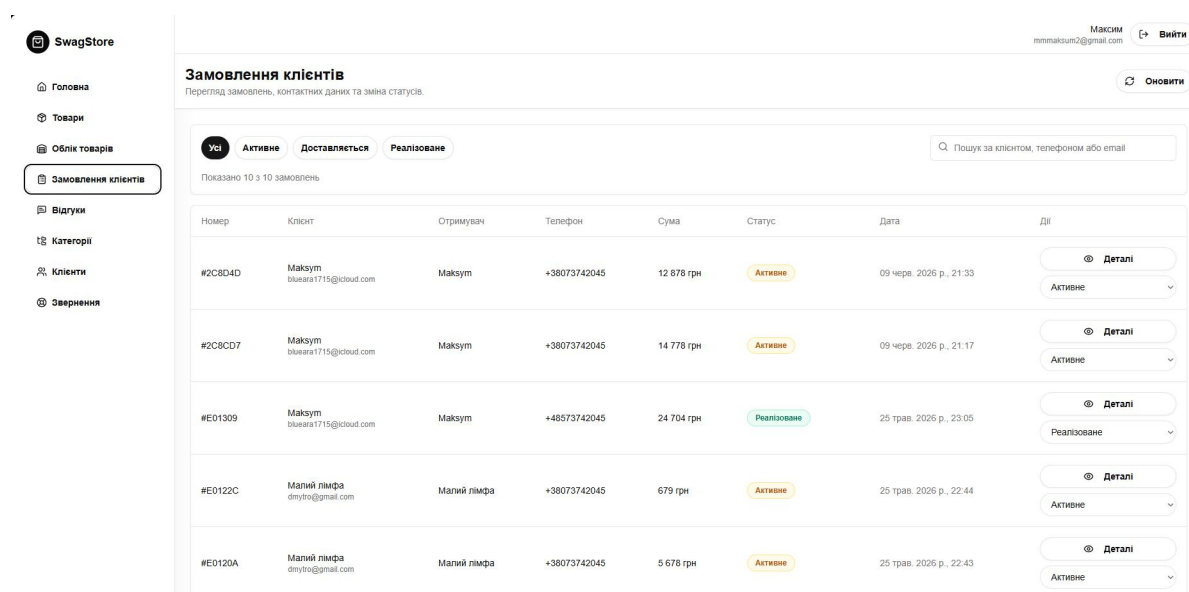


Рисунок 2.19 — Керування замовленнями в адміністративній панелі

Додатково перевірено розділ відгуків, у якому адміністратор може переглядати повідомлення користувачів і керувати їх відображенням у системі. Це дозволяє контролювати якість зворотного зв'язку та залишати на сайті лише актуальні повідомлення. Перегляд відгуків користувачів подано на рисунку 2.20.

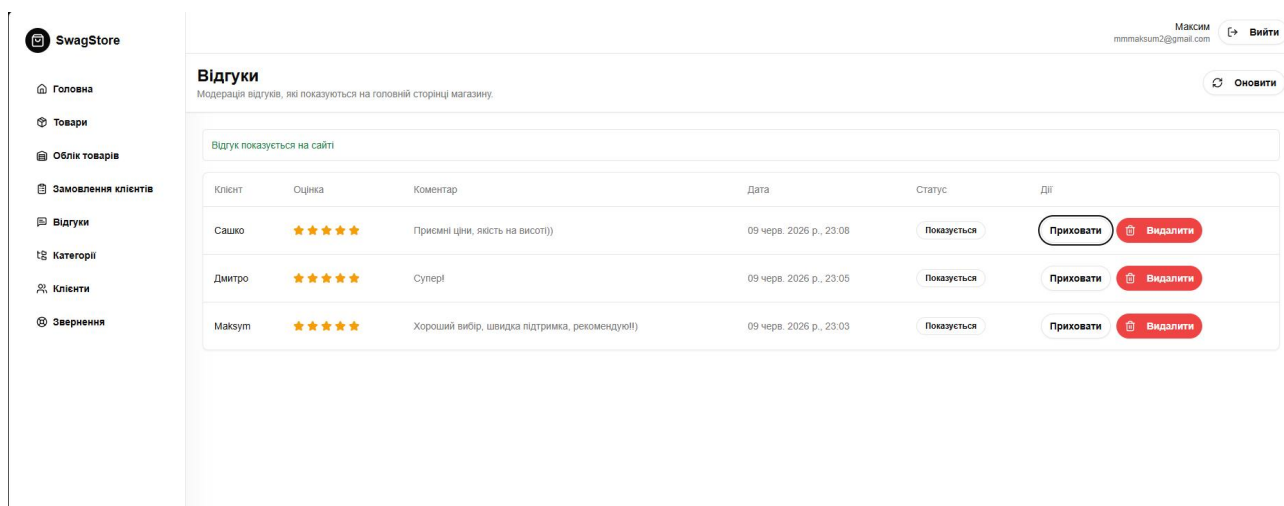


Рисунок 2.20 — Перегляд відгуків користувачів

Останнім етапом перевірено розділ підтримки. Було протестовано створення звернення користувачем, його збереження на сервері та відображення в адміністративній панелі. Додатково перевірено, що адміністратор може переглядати звернення та контролювати їх подальшу обробку. Розділ звернень користувачів подано на рисунку 2.21.

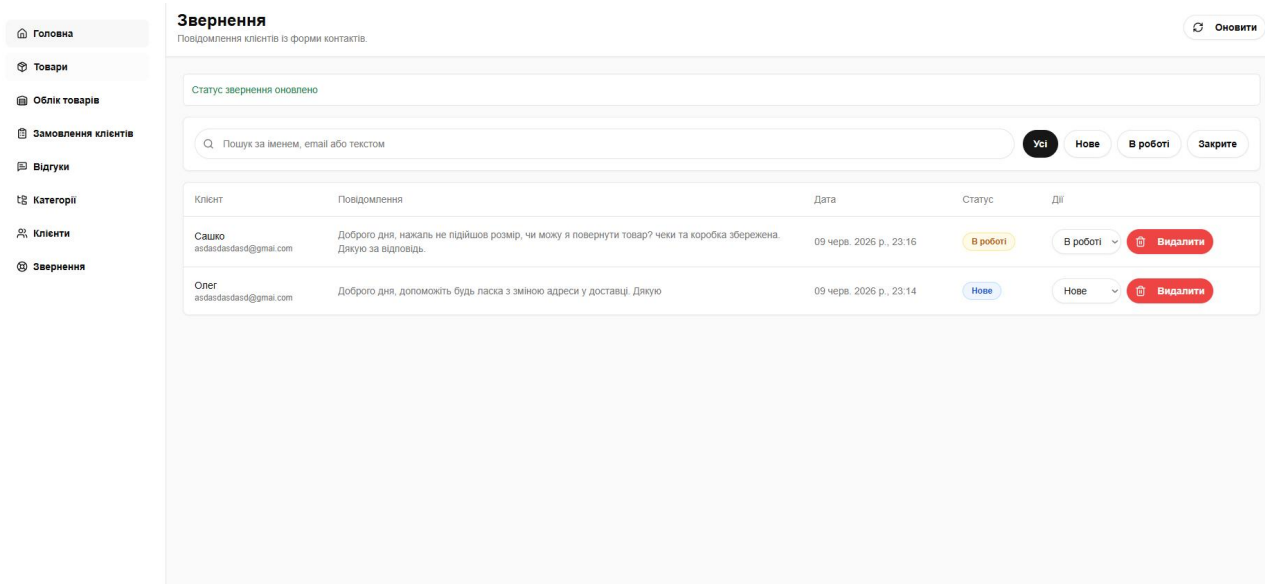


Рисунок 2.21 — Звернення користувачів до підтримки

Додатково було виконано статичний аналіз якості програмного коду за допомогою SonarQube Cloud. Такий аналіз дозволяє оцінити стан програмного коду за основними показниками якості, зокрема наявністю потенційних помилок, проблем підтримуваності, дублювання коду та можливих вразливостей. Використання цього інструмента дало змогу отримати незалежну оцінку програмної реалізації проєкту, виявити проблемні місця та визначити напрями його подальшого вдосконалення.

Результати статичного аналізу коду вебзастосунку SwagStore подано на рисунку 2.22.

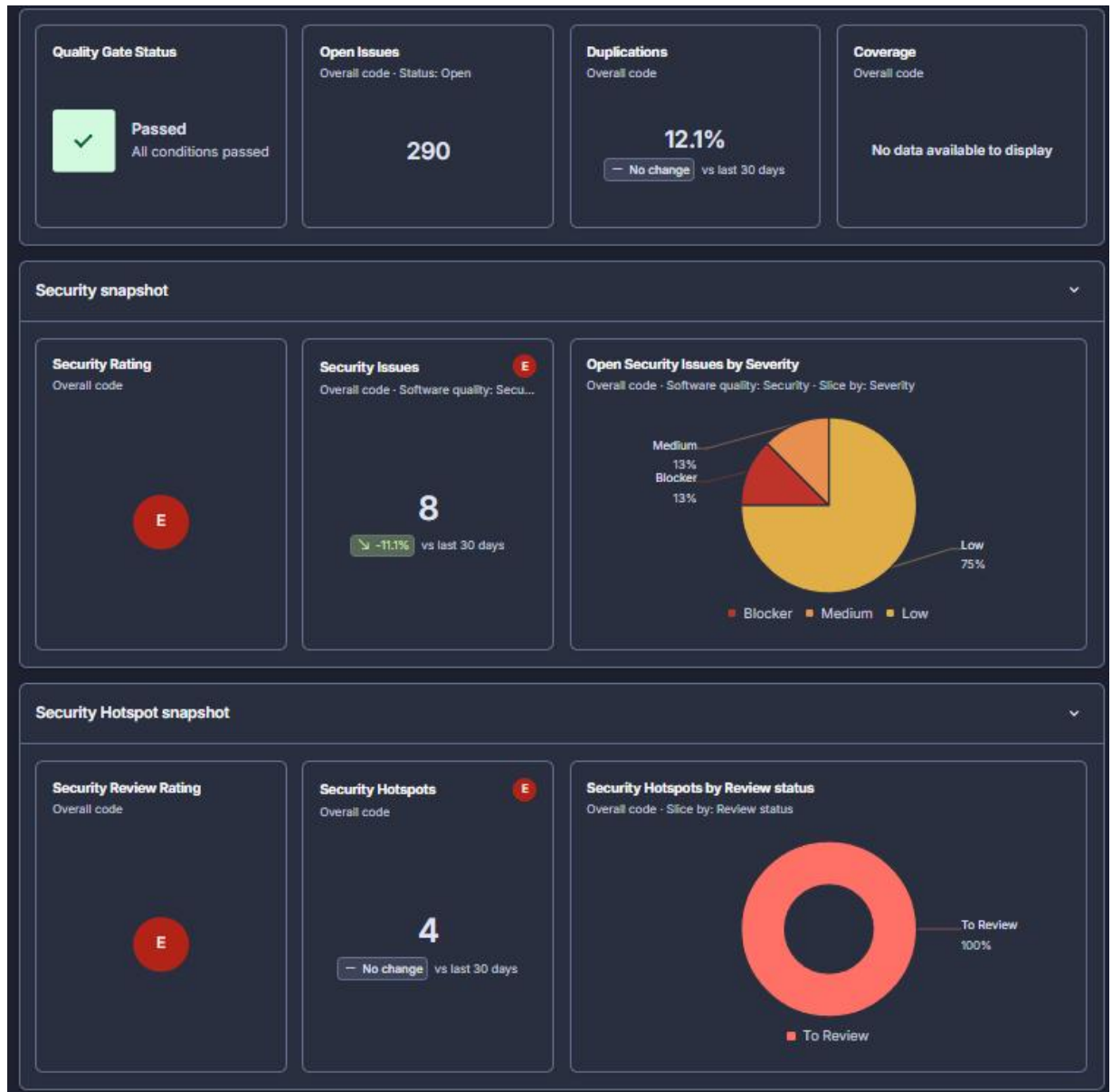


Рисунок 2.22 — Результати статичного аналізу коду в SonarQube Cloud

За результатами аналізу було отримано загальну оцінку стану програмного коду, зокрема показники якості, дублювання, безпеки та підтримуваності. Перевірка показала, що проєкт пройшов Quality Gate, а виявлені зауваження можуть бути використані для подальшого вдосконалення структури коду та підвищення якості програмної реалізації вебзастосунку. Це підтверджує доцільність використання засобів статичного аналізу під час супроводу й розвитку програмного продукту.

У результаті тестування підтверджено коректну роботу основних функцій вебзастосунку SwagStore. Перевірено перегляд товарів, роботу каталогу, пошуку й фільтрів, сторінки товару, кошика, оформлення замовлення, демонстраційного вибору способу оплати, кабінету користувача та адміністративної панелі. Проведена перевірка показала, що вебзастосунок відповідає поставленим функціональним вимогам і може бути використаний як демонстраційна система інтернет-магазину одягу. Також підтверджено, що основні сценарії взаємодії користувача із системою виконуються послідовно та без порушення логіки роботи. Це є важливим для інтернет-магазину, оскільки всі етапи — від перегляду товару до оформлення замовлення — мають працювати узгоджено.

Додаткове використання статичного аналізу коду дозволило не лише перевірити роботу вебзастосунку з боку користувача, а й оцінити якість програмної реалізації. Такий підхід є доцільним, оскільки під час подальшої підтримки системи важливо враховувати не тільки працездатність сторінок, а й зручність супроводу коду, наявність потенційних помилок та можливість безпечного розширення функціоналу. Це підвищує надійність проєкту та спрощує його подальше вдосконалення.

Отже, проведена перевірка підтвердила, що реалізовані модулі вебзастосунку SwagStore працюють узгоджено між собою. Клієнтська частина забезпечує зручну взаємодію користувача з інтернет-магазином, серверна частина виконує обробку запитів і перевірку даних, а база даних зберігає інформацію про товари, користувачів, замовлення, складські залишки, відгуки та звернення. Проведена перевірка дає підстави вважати розроблений вебзастосунок завершеним у межах поставленого технічного завдання, а також готовим до демонстрації та подальшого використання в межах дипломного проєкту. Отримані результати можуть бути використані як основа для подальшого розвитку системи.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| | | | | | | 45 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Розміщення вебзастосунку та вимоги до середовища

Розроблений вебзастосунок SwagStore може бути розміщений як локально для демонстрації та тестування, так і на віддаленому сервері для подальшого використання користувачами. Оскільки система складається з клієнтської частини, серверної частини та бази даних, для її стабільної роботи необхідно забезпечити правильне налаштування кожного з цих компонентів.

Клієнтська частина вебзастосунку реалізована з використанням Next.js, React та TypeScript. Вона відповідає за відображення сторінок інтернет-магазину, взаємодію користувача з каталогом, кошиком, формою оформлення замовлення, сторінкою товару та особистим кабінетом. Для запуску клієнтської частини необхідне середовище Node.js та менеджер пакетів npm. Під час локального тестування клієнтська частина може запускатися на порту 3000, що дозволяє перевіряти роботу інтерфейсу у браузері.

Серверна частина реалізована з використанням Node.js, Express та TypeScript. Вона відповідає за обробку API-запитів, авторизацію користувачів, роботу з товарами, категоріями, складом, замовленнями, відгуками та зверненнями до підтримки. Серверна частина взаємодіє з базою даних MongoDB через бібліотеку Mongoose. Під час локального запуску сервер може працювати на порту 5000, а клієнтська частина надсилає до нього запити через відповідні API-маршрути.

Для зберігання даних у вебзастосунку використовується MongoDB. У базі даних зберігаються відомості про користувачів, товари, категорії, залишки на складі, замовлення, відгуки та звернення користувачів. Завдяки використанню Mongoose структура даних описується у вигляді моделей, що полегшує роботу з документами бази даних та забезпечує зручну взаємодію між серверною частиною і MongoDB.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 46 |

Для локального розгортання вебзастосунку необхідно встановити Node.js, npm, підготувати базу даних MongoDB та налаштувати змінні середовища. У серверній частині до таких змінних можуть належати адреса підключення до бази даних, порт сервера та секретний ключ для роботи з JWT. У клієнтській частині необхідно вказати адресу API, до якого надсилатимуться запити. Це дозволяє гнучко змінювати налаштування залежно від того, де саме запускається система: локально або на віддаленому сервері.

Для демонстрації роботи проєкту достатньо локального запуску клієнтської та серверної частини. Спочатку запускається серверна частина, яка забезпечує роботу API та підключення до бази даних. Після цього запускається клієнтська частина, через яку користувач взаємодіє з вебзастосунком. Такий підхід дозволяє перевірити повний цикл роботи системи: від перегляду товарів до оформлення замовлення та його подальшої обробки адміністратором. Такий спосіб запуску також зручний для внесення змін і перевірки окремих функцій перед можливим розміщенням вебзастосунку на віддаленому сервері.

У разі подальшого розміщення вебзастосунку в мережі Інтернет клієнтську частину можна розгорнути на платформі, що підтримує Next.js, наприклад Vercel. Серверну частину можна розмістити на хостингу або платформі для Node.js-застосунків, наприклад Render. Базу даних доцільно розмістити у хмарному сервісі MongoDB Atlas. Такий варіант дозволяє відокремити клієнтську частину, сервер і базу даних, що спрощує підтримку та масштабування вебзастосунку.

Для зберігання даних вебзастосунку використовується MongoDB. Під час перевірки роботи бази даних було переглянуто створені колекції, у яких зберігається інформація про товари, категорії, складські залишки, історію змін складу, замовлення, відгуки, звернення користувачів та облікові записи. Перегляд структури бази даних дозволяє переконатися, що основні моделі вебзастосунку коректно представлені у вигляді окремих колекцій. Приклад відображення колекцій бази даних проєкту подано на рисунку 3.1.

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | <i>2026.KBP.122.421.14.00.00 ПЗ</i> | Арк. |
| | | | | | | 47 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

| Collection name | Properties | Storage size | Data size | Documents | Avg. document size | Indexes | Total index size |
|--------------------|------------|--------------|-----------|-----------|--------------------|---------|------------------|
| categories | - | 36.86 kB | 1.01 kB | 7 | 143.00 B | 2 | 73.73 kB |
| goods | - | 45.06 kB | 10.68 kB | 14 | 762.00 B | 2 | 73.73 kB |
| inventories | - | 36.86 kB | 10.80 kB | 68 | 186.00 B | 2 | 73.73 kB |
| inventoryhistories | - | 36.86 kB | 8.49 kB | 48 | 176.00 B | 1 | 36.86 kB |
| orders | - | 36.86 kB | 8.63 kB | 10 | 862.00 B | 2 | 73.73 kB |
| reviews | - | 36.86 kB | 602.00 B | 3 | 200.00 B | 1 | 36.86 kB |
| supportmessages | - | 36.86 kB | 649.00 B | 2 | 324.00 B | 1 | 36.86 kB |
| users | - | 36.86 kB | 1.92 kB | 7 | 274.00 B | 2 | 73.73 kB |

Рисунок 3.1 — Відображення колекцій бази даних проєкту в MongoDB Compass

Як видно з рисунка 3.1, база даних містить колекції categories, goods, inventories, inventoryhistories, orders, reviews, supportmessages та users. Така структура відповідає основним функціональним модулям вебзастосунку та забезпечує збереження даних, необхідних для роботи клієнтської й адміністративної частини системи.

Під час розміщення системи важливо правильно налаштувати взаємодію між клієнтом і сервером. Для цього необхідно вказати актуальну адресу API у змінних середовища клієнтської частини, а також дозволити серверу приймати запити з домену, на якому розміщено вебзастосунок. Крім цього, потрібно захистити конфіденційні дані, зокрема рядок підключення до бази даних і JWT-секрет, щоб вони не зберігалися у відкритому доступі.

Таким чином, вебзастосунок SwagStore може бути запущений як у локальному середовищі для тестування, так і на віддаленій інфраструктурі для демонстрації або подальшого використання. Обрана архітектура на основі Next.js, Node.js, Express і MongoDB забезпечує гнучкість розгортання, зручність підтримки та можливість подальшого розвитку системи.

3.2 Інструкція адміністратора для роботи з вебзастосунком

Адміністративна частина вебзастосунку SwagStore призначена для керування основними даними інтернет-магазину. Через неї адміністратор може додавати та редагувати товари, створювати категорії, контролювати залишки на

складі, переглядати замовлення користувачів, працювати з відгуками та зверненнями до підтримки. Наявність адміністративної панелі дозволяє керувати магазином без прямого доступу до бази даних і без редагування програмного коду.

Для початку роботи адміністратор повинен виконати вхід до системи, використовуючи обліковий запис із відповідною роллю. Після введення електронної пошти та пароля система перевіряє дані користувача. Якщо авторизація проходить успішно і користувач має права адміністратора, йому відкривається доступ до адміністративної панелі. Звичайні користувачі не мають доступу до службових розділів, що забезпечує базове розмежування прав у системі.

Після входу адміністратор потрапляє до панелі керування, де розміщені основні розділи для роботи з вебзастосунком: товари, категорії, склад, замовлення, відгуки та підтримка. Кожен із цих розділів відповідає за окрему частину функціональності магазину та дозволяє виконувати необхідні дії через графічний інтерфейс. Загальний вигляд адміністративної панелі SwagStore подано на рисунку 3.2.

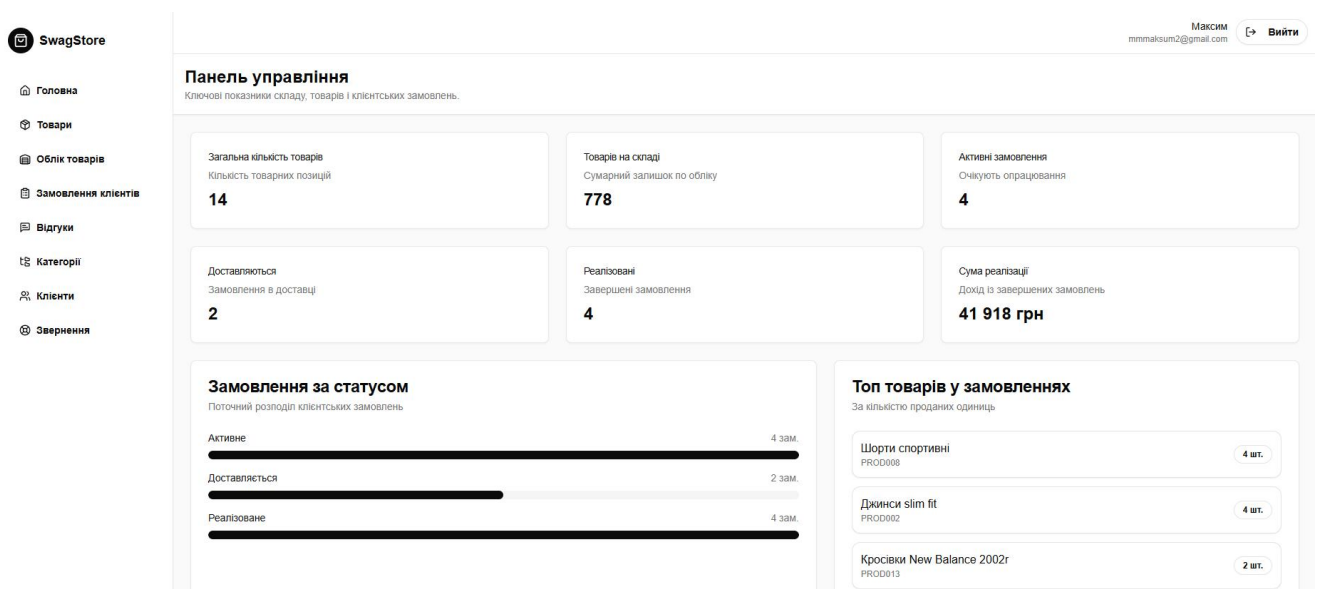


Рисунок 3.2 — Загальний вигляд адміністративної панелі SwagStore

Для наповнення каталогу адміністратор може додавати нові товари через спеціальну форму. У ній вказуються основні характеристики товару: назва, бренд, розділ, категорія, ціна, опис, доступні розміри, колір та посилання на зображення. Такий підхід дозволяє оновлювати асортимент магазину без безпосереднього редагування бази даних. Після збереження новий товар додається до загального списку позицій і може бути використаний у клієнтській частині вебзастосунку. Це спрощує наповнення каталогу та дозволяє швидко розширювати асортимент магазину. Форму додавання нового товару подано на рисунку 3.3.

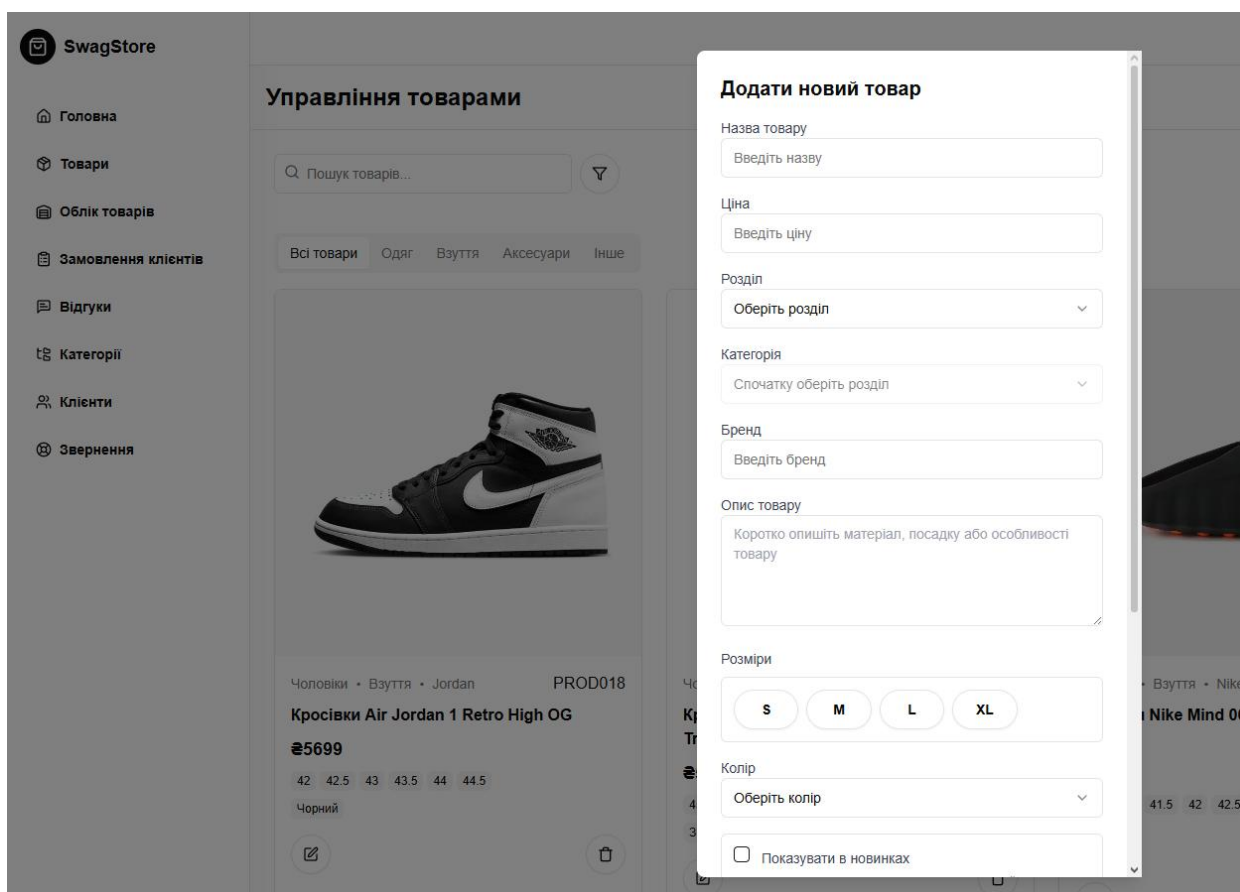


Рисунок 3.3 — Форма додавання нового товару в адміністративній панелі

У нижній частині форми адміністратор може вказати доступні розміри, колір, посилання на зображення та додаткові параметри відображення товару в каталозі. Додаткові параметри форми подано на рисунку 3.4.

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 50 |

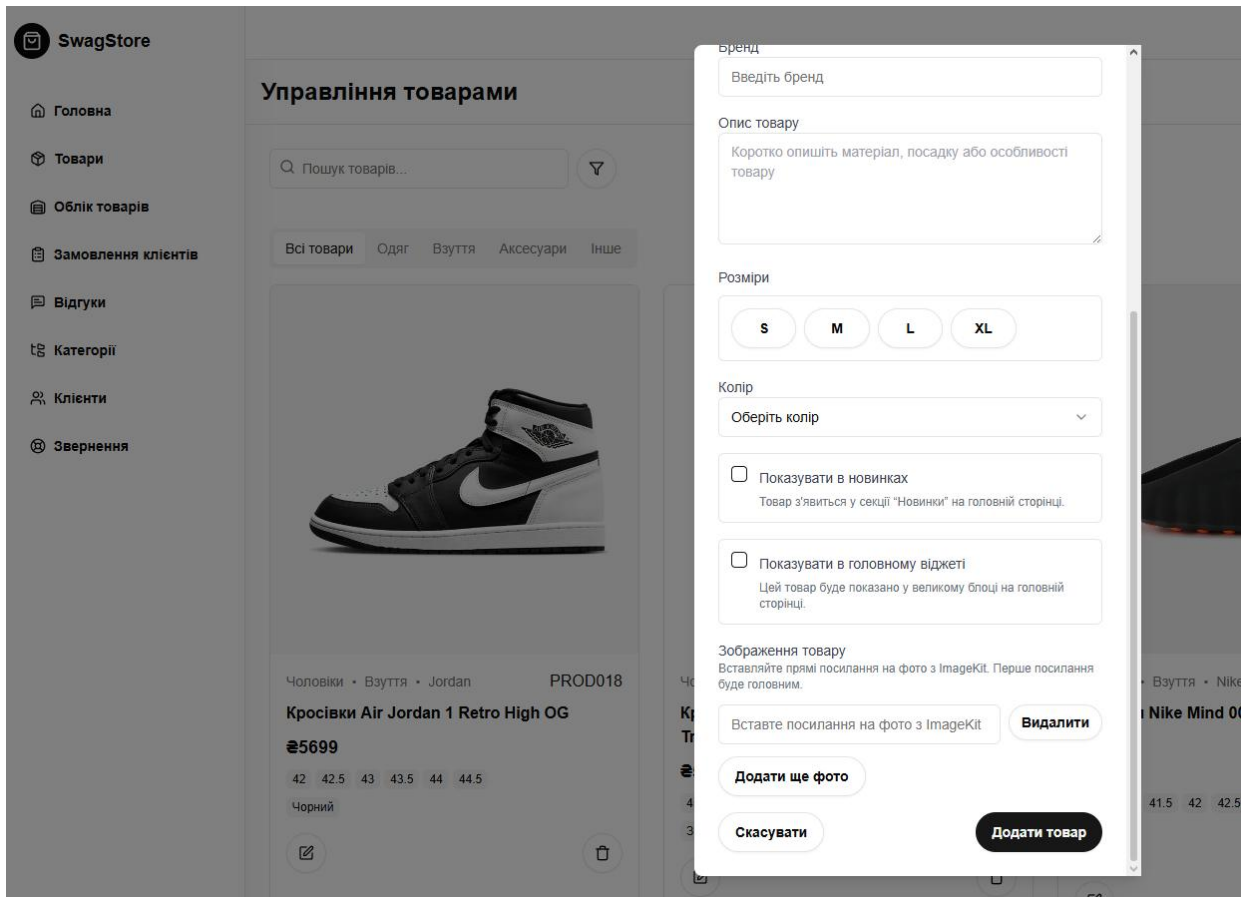


Рисунок 3.4 — Додаткові параметри форми додавання нового товару

Зміна статусу замовлення є важливою для інформування користувача про поточний етап обробки покупки. Завдяки цьому адміністратор може підтримувати актуальний стан замовлень і контролювати їх виконання без додаткового звернення до бази даних.

Окремий розділ адміністративної панелі призначений для керування складськими залишками товарів. Цей розділ дозволяє адміністратору швидко перевіряти актуальну кількість товарів без переходу безпосередньо до бази даних. У цьому розділі адміністратор може переглядати інформацію про наявність товарів, їх кількість, розмір, колір та склад розміщення. Така функціональність є важливою для інтернет-магазину одягу, оскільки один товар може мати кілька варіантів за розміром і кольором. Розділ керування складськими залишками подано на рисунку 3.5.

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 51 |

Управління інвентарем

Пошук товарів...

+ Додати товар

Всі товари Низький запас Немає в наявності

Інвентар

Список всіх товарів на складі.

| ID | Назва | Категорія | Бренд | Розмір | Колір | Кількість | Розташування | Оновлено | Дії |
|--------|--|-----------|--------|--------|---------|-----------|--------------|--------------------------|------|
| INV058 | Кросівки Air Jordan 1 Retro Low OG SP Travis Scott | Взуття | Jordan | 44.5 | Зелений | 2 | Склад А | 2026-06-09T20:49:12.612Z | 🔍 🗑️ |
| INV057 | Кросівки Air Jordan 1 Retro Low OG SP Travis Scott | Взуття | Jordan | 45 | Зелений | 3 | Склад А | 2026-06-09T20:49:12.482Z | 🔍 🗑️ |
| INV056 | Кросівки Nike Mind 001 Flyknit Slide | Взуття | Nike | 45 | Чорний | 7 | Склад А | 2026-06-09T20:49:03.437Z | 🔍 🗑️ |
| INV055 | Кросівки Nike Mind 001 Flyknit Slide | Взуття | Nike | 44.5 | Чорний | 7 | Склад А | 2026-06-09T20:49:03.317Z | 🔍 🗑️ |
| INV054 | Кросівки Nike Mind 001 Flyknit Slide | Взуття | Nike | 44 | Чорний | 6 | Склад А | 2026-06-09T20:49:03.197Z | 🔍 🗑️ |
| INV053 | Кросівки Nike Mind 001 Flyknit Slide | Взуття | Nike | 43.5 | Чорний | 6 | Склад А | 2026-06-09T20:49:03.072Z | 🔍 🗑️ |

Рисунок 3.5 — Розділ керування складськими залишками

Завдяки цьому розділу адміністратор може контролювати актуальну кількість товарів у системі та своєчасно оновлювати інформацію про залишки. Це зменшує ризик оформлення замовлення на товар, якого фактично немає в наявності, та забезпечує коректну роботу каталогу для користувачів.

Крім цього, адміністратор може використовувати цей розділ для оперативного оновлення залишків після надходження нових товарів або зміни кількості на складі. Це особливо важливо для магазину одягу, де один товар може мати кілька варіантів за розміром і кольором. Також це дає змогу швидко реагувати на зміну залишків і підтримувати актуальність інформації в каталозі товарів.

Для контролю процесу виконання покупок в адміністративній панелі передбачено розділ керування замовленнями клієнтів. У цьому розділі адміністратор може переглядати створені замовлення, перевіряти дані покупця, склад замовлення, загальну суму, спосіб оплати та поточний статус виконання. Це дозволяє контролювати замовлення від моменту його створення до завершення обробки. Розділ керування замовленнями клієнтів подано на рисунку 3.6.

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Арк. | № докум. | Підпис | Дата |

2026.КВР.122.421.14.00.00 ПЗ

Арк.

52

Замовлення клієнтів

Перегляд замовлень, контактних даних та зміна статусів.

Оновити

Усі Активне Доставляється Реалізоване

Пошук за клієнтом, телефоном або email

Показано 10 з 10 замовлень

| Номер | Клієнт | Отримувач | Телефон | Сума | Статус | Дата | Дії |
|---------|---------------------------------|-------------|--------------|------------|-------------|-------------------------|---------------------------------------|
| #2C8D4D | Maksym blueara1715@cloud.com | Maksym | +38073742045 | 12 878 грн | Реалізоване | 09 черв. 2026 р., 21:33 | Деталі Реалізоване |
| #2C8CD7 | Maksym blueara1715@cloud.com | Maksym | +38073742045 | 14 778 грн | Активне | 09 черв. 2026 р., 21:17 | Деталі Активне |
| #E01309 | Maksym blueara1715@cloud.com | Maksym | +48573742045 | 24 704 грн | Реалізоване | 25 трав. 2026 р., 23:05 | Деталі Реалізоване |
| #E0122C | Малий лімба dmytro@gmail.com | Малий лімба | +38073742045 | 679 грн | Активне | 25 трав. 2026 р., 22:44 | Деталі Активне |

Рисунок 3.6 — Розділ керування замовленнями клієнтів

Завдяки цьому розділу адміністратор може оперативно відстежувати стан замовлень і змінювати їхній статус відповідно до етапу обробки. Такий підхід спрощує контроль за виконанням замовлень і забезпечує узгоджену роботу клієнтської та адміністративної частини вебзастосунку.

Окремий розділ адміністративної панелі призначений для перегляду звернень користувачів. У ньому адміністратор може бачити повідомлення, які були надіслані через форму підтримки, переглядати контактні дані користувача, текст звернення та поточний статус його обробки. Це дозволяє швидко реагувати на запити клієнтів і підтримувати зворотний зв'язок із користувачами вебзастосунку. Розділ перегляду звернень користувачів подано на рисунку 3.7.

Звернення

Повідомлення клієнтів із форми контактів.

Оновити

Пошук за іменем, email або текстом

Усі Нове В роботі Закрите

| Клієнт | Повідомлення | Дата | Статус | Дії |
|------------------------------|---|-------------------------|----------|---|
| Сашко asdasdasd@gmail.com | Доброго дня, нажаль не підійшов розмір, чи можу я повернути товар? чеки та коробка збережена. Дякую за відповідь. | 09 черв. 2026 р., 23:16 | В роботі | В роботі Видалити |
| Олег asdasdasd@gmail.com | Доброго дня, допоможіть будь ласка з зміною адреси у доставці. Дякую | 09 черв. 2026 р., 23:14 | Нове | Нове Видалити |

Рисунок 3.7 — Розділ перегляду звернень користувачів

Наявність такого розділу спрощує роботу адміністратора із запитами користувачів, оскільки всі звернення зберігаються в одному місці та можуть бути опрацьовані без прямого доступу до бази даних. Це підвищує зручність адміністрування та забезпечує кращу взаємодію між магазином і клієнтами.

У розділі товарів адміністратор може переглядати список усіх товарних позицій, додавати нові товари, редагувати наявні або видаляти непотрібні. Під час додавання товару вказуються його назва, бренд, категорія, опис, ціна, розміри, колір, кількість на складі та посилання на зображення. Також може бути встановлено додаткові параметри, наприклад позначення товару як новинки або вибір товару для відображення в головному hero-блоці на головній сторінці. Після збереження новий товар з'являється у списку адміністративної панелі та стає доступним у відповідних розділах клієнтської частини сайту.

Розділ категорій використовується для структурування товарів у магазині. Адміністратор може створювати та редагувати категорії, які надалі використовуються під час додавання товарів і фільтрації в каталозі. Завдяки категоріям товари розподіляються між основними розділами сайту: чоловічий одяг, жіночий одяг, аксесуари та унісекс. Це полегшує навігацію для користувача та робить каталог більш зрозумілим.

Окреме значення має розділ складського обліку. У ньому адміністратор контролює залишки товарів, їх кількість, колір, розмір і розташування на складі. Для зручності роботи реалізовано можливість редагування залишків за всіма розмірами одного товару в одному вікні. Це дозволяє швидко вказати кількість для кожного розміру без необхідності створювати окремий складський запис вручну для кожної позиції. Дані складського обліку впливають на клієнтську частину: якщо товару немає в наявності, користувач не може додати його до кошика, а при низькому залишку система відображає позначку "Мало на складі".

У розділі замовлень адміністратор може переглядати всі оформлені користувачами замовлення. Для кожного замовлення відображається інформація про користувача, склад замовлення, загальна сума, спосіб оплати, дата створення

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 54 |

та поточний статус. Адміністратор може змінювати статус замовлення відповідно до етапу його обробки, наприклад “Активне”, “Доставляється” або “Реалізоване”. Це дозволяє використовувати систему не лише як каталог товарів, а й як інструмент для керування процесом продажу.

Розділ відгуків призначений для перегляду повідомлень, які користувачі залишають про магазин або товари. Адміністратор може аналізувати ці відгуки, щоб оцінити зручність користування сайтом і загальне враження клієнтів. Розділ підтримки використовується для перегляду звернень користувачів, надісланих через форму зв'язку. У таких зверненнях можуть міститися контактні дані, тема повідомлення та текст запиту.

Під час роботи з адміністративною панеллю адміністратор повинен уважно перевіряти введені дані перед збереженням. Особливо це стосується ціни товару, категорії, розмірів, кількості на складі та посилань на зображення. Для підтримання актуальності даних потрібно регулярно оновлювати залишки товарів, змінювати статуси замовлень і переглядати звернення користувачів.

Таким чином, адміністративна панель SwagStore забезпечує повний набір інструментів для керування інтернет-магазином. Вона дозволяє адміністратору працювати з товарами, категоріями, складськими залишками, замовленнями, відгуками та зверненнями користувачів через єдиний інтерфейс, що спрощує обслуговування вебзастосунку та робить систему придатною для демонстрації роботи інтернет-магазину одягу.

3.3 Підтримка, оновлення та подальший розвиток системи

Після розробки та тестування вебзастосунку SwagStore важливим етапом є його подальша підтримка. Вона передбачає контроль стабільної роботи клієнтської та серверної частини, актуальність даних у базі даних, оновлення товарного наповнення, перевірку коректності оформлення замовлень і своєчасне реагування на звернення користувачів.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 55 |

Адміністратор повинен регулярно оновлювати інформацію про товари: додавати нові позиції, редагувати описи, змінювати ціни, перевіряти категорії та контролювати залишки на складі. Особливу увагу потрібно приділяти складському обліку, оскільки від нього залежить можливість додавання товару до кошика та відображення статусів наявності в клієнтській частині сайту.

Важливою частиною підтримки є робота із замовленнями. Адміністратор повинен переглядати нові замовлення, змінювати їхні статуси відповідно до етапу обробки та контролювати правильність відображення інформації для користувача. Також необхідно переглядати відгуки та звернення до підтримки, оскільки вони можуть містити повідомлення про помилки або пропозиції щодо покращення роботи магазину.

З технічної точки зору підтримка вебзастосунку передбачає контроль роботи серверної частини, оновлення залежностей, резервне копіювання важливих даних та перевірку змін у локальному середовищі перед їх внесенням до основної версії системи. Це дозволяє зменшити ризик втрати даних і уникнути помилок у роботі кошика, оформлення замовлення, авторизації або адміністративної панелі після оновлення. У подальшому вебзастосунок SwagStore можна розширити додатковими можливостями. Наприклад, можна підключити реальний платіжний сервіс, додати систему промокодів, покращити аналітику продажів в адміністративній панелі та реалізувати автоматичні повідомлення користувачам про зміну статусу замовлення. Такі доповнення зроблять систему більш наближеною до повноцінного комерційного інтернет-магазину.

Таким чином, подальша підтримка SwagStore передбачає регулярне оновлення товарів, контроль складських залишків, обробку замовлень, перегляд звернень користувачів, технічний моніторинг серверної частини та захист даних. Реалізована структура вебзастосунку дозволяє підтримувати поточну функціональність і за потреби поступово розширювати систему новими можливостями.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 56 |

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини дипломного проєкту є проведення економічних розрахунків, спрямованих на визначення доцільності розробки та впровадження вебзастосунку «SwagStore». Розроблювана система поєднує функції інтернет-магазину одягу та внутрішнього обліку товарів, що дає змогу автоматизувати роботу з каталогом, залишками на складі, замовленнями клієнтів, відгуками та зверненнями в підтримку.

Об'єктом розробки є вебзастосунок «SwagStore», призначений для організації продажу одягу через інтернет та ведення обліку товарних позицій магазину. Система передбачає роботу з ролями користувача та адміністратора, оформлення замовлень, вибір розміру товару, фільтрацію каталогу, перегляд статистики клієнтів і керування товарами в адміністративній панелі.

Розрахунок вартості розробки виконується в декілька етапів: визначення трудомісткості робіт; розрахунок витрат на оплату праці та соціальні відрахування; визначення витрат на електроенергію; обчислення амортизаційних відрахувань; визначення накладних витрат; складання кошторису, розрахунок ціни робіт, економічної ефективності та терміну окупності.

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення робіт

У процесі створення вебзастосунку «SwagStore» було виділено основні етапи технологічного процесу: аналіз предметної області, підготовка технічного завдання, проєктування інтерфейсу, розробка клієнтської та серверної частин, налаштування бази даних, тестування, документування й підготовка проєкту до демонстрації.

Для визначення загальної тривалості робіт дані щодо витрат часу за окремими операціями технологічного процесу зведено у таблицю 4.1.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 57 |

Таблиця 4.1 – Середній час виконання робіт за стадіями технологічного процесу

| № п/п | Назва операції (стадії) | Виконавець | Середній час виконання, год. |
|-------|---|---------------------------------------|------------------------------|
| 1 | Планування та аналіз предметної області | Керівник проєкту (Pm) Інженер (I1) | 6 4 |
| 2 | Розробка технічного завдання | Керівник проєкту (Pm) Інженер (I1) | 5 4 |
| 3 | Проектування UI/UX та макетів сторінок | Інженер (I1) Інженер (I2) | 12 10 |
| 4 | Проектування архітектури та структури бази даних | Інженер (I1) Інженер (I2) | 6 10 |
| 5 | Розробка клієнтської частини, каталогу, кошика та оформлення замовлення | Інженер (I1) Інженер (I2) | 28 12 |
| 6 | Розробка серверної частини, API та адміністративних модулів | Інженер (I1) Інженер (I2) | 14 20 |
| 7 | Тестування, налагодження та виправлення помилок | Тестувальник Інженер (I1) | 18 8 |
| 8 | Підготовка технічної та звітної документації | Інженер (I1) | 10 |
| 9 | Управління проєктом і контроль виконання робіт | Керівник проєкту (Pm) | 13 |
| | Разом | | 180 |

Сумарний час виконання операцій технологічного процесу становить 180 годин. Такий обсяг робіт є обґрунтованим для навчального програмного продукту, який включає клієнтську частину, серверну логіку, базу даних, адміністративну панель та функціонал інтернет-магазину.

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

У цьому підрозділі проводиться розрахунок витрат, пов'язаних з оплатою праці спеціалістів, залучених до розробки вебзастосунку «SwagStore». До складу виконавців включено керівника проєкту, інженерів різного рівня та тестувальника.

Основна заробітна плата визначається за формулою:

$$Z_{осн} = T_c \cdot K_g, \quad (4.1)$$

де T_c – тарифна ставка, грн/год; K_g – кількість відпрацьованих годин. Для розрахунку прийнято такі тарифні ставки: керівник проєкту – 450 грн/год, інженер (12) – 272 грн/год, інженер (11) – 113 грн/год, тестувальник – 100 грн/год.

Основна заробітна плата становить: керівник проєкту – $24 \cdot 450 = 10\,800$ грн; інженер (12) – $52 \cdot 272 = 14\,144$ грн; інженер (11) – $86 \cdot 113 = 9\,718$ грн; тестувальник – $18 \cdot 100 = 1\,800$ грн. Сумарна основна заробітна плата дорівнює 36 462 грн.

Додаткова заробітна плата становить 10 % від основної заробітної плати:

$$Z_{дод} = Z_{осн} \cdot K_{допл}, \quad (4.2)$$

Загальна додаткова заробітна плата становить 3 646 грн. Загальні витрати на оплату праці без урахування ЄСВ дорівнюють:

$$V_{о.п} = Z_{осн} + Z_{дод} = 36\,462 + 3\,646 = 40\,108 \text{ грн.} \quad (4.3)$$

Єдиний соціальний внесок приймається у розмірі 22 %:

$$V_{єсв} = V_{о.п} \cdot 0,22 = 40\,108 \cdot 0,22 = 8\,824 \text{ грн.} \quad (4.4)$$

Зведені результати розрахунку основної заробітної плати, додаткової заробітної плати, ЄСВ та загальних витрат на оплату праці наведено в таблиці 4.2.

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| | | | | | | 59 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

| № | Категорія працівників | Тариф на ставка, грн | К-сть годин | Основна зарплата, грн | Додатков а зарплата, грн | ЄСВ, грн | Всього, грн |
|---|-----------------------|----------------------|-------------|-----------------------|--------------------------|--------------|---------------|
| 1 | Керівник проекту (Pm) | 450 | 24 | 10 800 | 1 080 | 2 614 | 14 494 |
| 2 | Інженер (12) | 272 | 52 | 14 144 | 1 414 | 3 423 | 18 981 |
| 3 | Інженер (11) | 113 | 86 | 9 718 | 972 | 2 352 | 13 042 |
| 4 | Тестувальник | 100 | 18 | 1 800 | 180 | 436 | 2 416 |
| | Разом | | 180 | 36 462 | 3 646 | 8 824 | 48 932 |

Отже, загальні витрати на оплату праці з урахуванням відрахувань на соціальні заходи становлять 48 932 грн.

4.3 Розрахунок витрат на електроенергію

Витрати на електроенергію визначаються за формулою:

$$Z_e = W \cdot T \cdot S, \quad (4.5)$$

де W – споживана потужність обладнання, кВт; T – кількість годин роботи обладнання; S – вартість 1 кВт·год електроенергії. Для виконання робіт використовується один комп'ютер із середньою споживаною потужністю 0,35 кВт. Тривалість роботи обладнання становить 180 годин, а вартість електроенергії приймається 15,94 грн за 1 кВт·год.

$$Z_e = 0,35 \cdot 180 \cdot 15,94 = 1\,004 \text{ грн.}$$

Витрати на електроенергію для розробки вебзастосунку «SwagStore» становлять 1 004 грн.

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 60 |

4.4 Розрахунок суми амортизаційних відрахувань

Амортизація – це процес поступового перенесення вартості основних засобів на собівартість створюваного продукту. Під час розробки вебзастосунку використовувалася комп’ютерна техніка та периферійне обладнання, необхідне для програмування, тестування і підготовки документації.

Сума амортизаційних відрахувань визначається за формулою:

$$A = BV \cdot NA \cdot T / (100 \cdot \Phi p), \quad (4.6)$$

Де BV – балансова вартість обладнання, грн; Na – річна норма амортизації, %; T – час використання обладнання під час розробки, год; Фр – річний фонд робочого часу обладнання, год.

Для розрахунку прийнято балансову вартість обладнання 53 000 грн, норму амортизації 25 %, час використання обладнання 180 годин та річний фонд робочого часу 2000 годин.

$$A = 53\,000 \cdot 25 \cdot 180 / (100 \cdot 2000) = 1\,193 \text{ грн.}$$

Отже, сума амортизаційних відрахувань становить 1 193 грн.

4.5 Обчислення накладних витрат

Накладні витрати пов’язані з організацією та обслуговуванням процесу розробки, оплатою адміністративних потреб, використанням програмних сервісів, забезпеченням робочих умов та іншими непрямими витратами. У межах розрахунку накладні витрати приймаються на рівні 40 % від суми основної та додаткової заробітної плати.

$$Nv = Vo.p \cdot 0,4, \quad (4.7)$$

$$Nv = 40\,108 \cdot 0,4 = 16\,043 \text{ грн.}$$

Отже, накладні витрати становлять 16 043 грн.

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 61 |

4.6 Складання кошторису витрат та визначення собівартості розробки

Для складання кошторису витрат та визначення собівартості розробки вебзастосунку «SwagStore» результати проведених вище розрахунків зведено у таблицю 4.3.

Таблиця 4.3 – Кошторис витрат на розробку вебзастосунку «SwagStore»

| № | Зміст витрат | Сума, грн | У % до загальної суми |
|---|-------------------------------|---------------|-----------------------|
| 1 | Витрати на оплату праці з ЄСВ | 48 932 | 72 |
| 2 | Витрати на електроенергію | 1 004 | 1 |
| 3 | Амортизаційні відрахування | 1 193 | 2 |
| 4 | Накладні витрати | 16 043 | 24 |
| 5 | Собівартість | 67 172 | 100 |

Собівартість розробки визначається за формулою:

$$Св = Во.п + Весв + Зе + А + Нв. \quad (4.8)$$

$$Св = 40\,108 + 8\,824 + 1\,004 + 1\,193 + 16\,043 = 67\,172 \text{ грн.}$$

Отже, собівартість розробки вебзастосунку «SwagStore» становить 67 172 грн.

4.7 Розрахунок ціни розробки

Розрахунок ціни робіт здійснюється з урахуванням собівартості, рівня рентабельності та податку на додану вартість. Для розрахунку приймається рівень

рентабельності 30 %, а ставка ПДВ – 20 %.

$$Ц = C_v \cdot (1 + P_{рен}) \cdot (1 + ПДВ), \quad (4.9)$$

$$Ц = 67\,172 \cdot (1 + 0,3) \cdot (1 + 0,2) = 104\,788 \text{ грн.}$$

Отже, розрахункова ціна розробки вебзастосунку «SwagStore» становить 104 788 грн.

4.8 Визначення економічної ефективності і терміну окупності

Економічна ефективність розробки визначається шляхом порівняння витрат на створення програмного продукту з очікуваним економічним результатом. Для вебзастосунку «SwagStore» економічний ефект може проявлятися у зменшенні часу на облік товарів, автоматизації формування замовлень, зниженні кількості помилок у складському обліку та підвищенні зручності роботи клієнтів і адміністратора.

Плановий прибуток або грошовий потік визначається як різниця між ціною розробки та собівартістю:

$$Г_{\Pi} = Ц - C_v = 104\,788 - 67\,172 = 37\,616 \text{ грн.} \quad (4.10)$$

Для оцінки ефективності використаємо чисту теперішню вартість за трирічний період із дисконтною ставкою 12 %. Розрахунок виконується за формулою:

$$ЧТВ = \sum (Г_{\Pi} / (1 + i)^t) - C_v. \quad (4.11)$$

$$ЧТВ = 37\,616 / 1,12 + 37\,616 / 1,12^2 + 37\,616 / 1,12^3 - 67\,172 = 23\,176 \text{ грн.}$$

Оскільки ЧТВ більше нуля, проєкт може бути рекомендований до впровадження. Термін окупності визначається як відношення собівартості до річного грошового потоку:

$$Ток = C_v / Г_{\Pi} = 67\,172 / 37\,616 = 1,8 \text{ року.} \quad (4.12)$$

Узагальнені результати економічних розрахунків та основні техніко-економічні показники розробки вебзастосунку «SwagStore» наведено в таблиці 4.4.

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 63 |

Таблиця 4.4 – Техніко-економічні показники розробки вебзастосунку

«SwagStore»

| № п/п | Показник | Значення |
|-------|--|----------|
| 1 | Собівартість, грн | 67 172 |
| 2 | Плановий прибуток / грошовий потік, грн | 37 616 |
| 3 | Ціна, грн | 104 788 |
| 4 | Чиста теперішня вартість, грн | 23 176 |
| 5 | Термін окупності, років | 1,8 |

Отримані техніко-економічні показники свідчать про доцільність розробки вебзастосунку «SwagStore». Проект має позитивну чисту теперішню вартість, а термін окупності становить приблизно 1,8 року. Це підтверджує економічну ефективність створення системи для інтернет-магазину одягу та обліку товарів, оскільки вона дозволяє автоматизувати ключові процеси, підвищити точність обліку, зменшити витрати часу адміністратора та покращити якість обслуговування клієнтів.

Крім того, впровадження такого вебзастосунку створює можливість для подальшого масштабування системи без значного збільшення витрат на обслуговування. У майбутньому до нього можна додати нові функції, зокрема розширену аналітику продажів, автоматичні повідомлення користувачам і підключення реального платіжного сервісу. Це підвищує практичну цінність розробки та підтверджує доцільність її використання для малого або середнього інтернет-магазину. Таким чином, отримані результати економічних розрахунків підтверджують практичну доцільність реалізації вебзастосунку SwagStore.

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 64 |

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

5.1 Відповідальність за порушення законодавства про охорону праці

Охорона праці є важливою складовою організації безпечної професійної діяльності, особливо у сфері інформаційних технологій, де працівники тривалий час працюють із комп'ютерною технікою, електрообладнанням, периферійними пристроями та програмними системами. Під час розробки вебзастосунку для інтернет-магазину одягу та обліку товарів «SwagStore» основна робота виконується за персональним комп'ютером або ноутбуком, тому важливими є дотримання вимог безпеки, правильна організація робочого місця, режим праці та відпочинку, а також відповідальне ставлення до використання обладнання.

Законодавство України про охорону праці встановлює обов'язки роботодавців і працівників щодо створення безпечних умов праці. Роботодавець зобов'язаний забезпечити належний технічний стан обладнання, провести інструктажі, організувати безпечне робоче місце, контролювати дотримання вимог безпеки та створити умови для збереження здоров'я працівників. Працівник, у свою чергу, повинен знати й виконувати інструкції з охорони праці, користуватися обладнанням за призначенням, не допускати самовільного втручання в електричні мережі або технічні засоби та повідомляти відповідальних осіб про виявлені несправності.

Порушення вимог охорони праці може призвести не лише до зниження ефективності роботи, а й до виникнення небезпечних ситуацій: ураження електричним струмом, перегрівання обладнання, пожежі, травмування працівника, погіршення зору, перевтоми та інших негативних наслідків. Саме тому законодавством передбачено різні види відповідальності за недотримання вимог охорони праці.

Дисциплінарна відповідальність застосовується до працівників у разі невиконання або неналежного виконання вимог інструкцій з охорони праці, правил внутрішнього трудового розпорядку чи розпоряджень керівника.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 65 |

Наприклад, працівник може порушити правила безпечної роботи з комп'ютерною технікою, використовувати несправні кабелі, самовільно підключати обладнання або ігнорувати встановлений режим праці та відпочинку. У таких випадках до нього можуть бути застосовані зауваження, догана або інші заходи дисциплінарного впливу відповідно до трудового законодавства.

Адміністративна відповідальність передбачена для посадових осіб і працівників у разі порушення нормативно-правових актів з охорони праці. Вона може проявлятися у вигляді штрафів за недотримання вимог безпеки, непроведення інструктажів, відсутність належної документації або неналежну організацію робочих місць. Для підприємств, які займаються розробкою програмного забезпечення, це може стосуватися недотримання вимог до приміщень, електробезпеки, освітлення, мікроклімату та ергономіки робочих місць.

Матеріальна відповідальність настає у випадку, коли внаслідок порушення правил охорони праці підприємству або працівнику завдано матеріальної шкоди. Наприклад, неправильне підключення техніки, використання несправного обладнання або нехтування правилами експлуатації може призвести до пошкодження комп'ютерів, моніторів, мережевого обладнання або втрати даних. У такому випадку винна особа може відшкодувати завдані збитки у межах, визначених чинним законодавством.

Кримінальна відповідальність є найсуворішим видом відповідальності та може наставати у випадках, коли порушення правил охорони праці спричинило тяжкі наслідки, шкоду здоров'ю працівника або загрозу життю людей. Хоча робота техника-програміста зазвичай не пов'язана з підвищеною виробничою небезпекою, неправильна експлуатація електрообладнання, відсутність контролю за технічним станом мережі або грубе ігнорування вимог безпеки можуть створити небезпечні ситуації.

Під час виконання проекту «SwagStore» особливого значення набуває дотримання правил безпечної роботи з комп'ютерною технікою. Робоче місце має

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 66 |

бути організоване так, щоб монітор знаходився на безпечній відстані від очей, клавіатура і миша були розташовані зручно, а електричні кабелі не створювали перешкод і не мали пошкоджень. Працівник повинен регулярно робити перерви, уникати перевантаження електромережі, не використовувати обладнання з видимими несправностями та дотримуватися правил інформаційної безпеки.

Важливим профілактичним заходом є проведення інструктажів з охорони праці. Працівники повинні бути ознайомлені з правилами безпечної експлуатації комп'ютерної техніки, діями у разі аварійних ситуацій, правилами евакуації, пожежної безпеки та першої допомоги. Проведення інструктажів дозволяє зменшити ризик порушень і підвищує рівень відповідальності кожного працівника.

Отже, відповідальність за порушення законодавства про охорону праці є важливим механізмом забезпечення безпечних умов праці. У процесі розробки вебзастосунку «SwagStore» дотримання правил охорони праці сприяє збереженню здоров'я працівників, стабільній роботі обладнання, запобіганню аварійним ситуаціям і підвищенню загальної культури безпеки на робочому місці.

5.2 Комплекс заходів по зменшенню шуму в джерелі його виникнення

Шум є одним із чинників виробничого середовища, який може негативно впливати на працездатність, концентрацію уваги та загальне самопочуття працівника. У сфері інформаційних технологій джерелами шуму можуть бути системні блоки, вентилятори охолодження комп'ютерів і ноутбуків, серверне обладнання, принтери, сканери, кондиціонери, джерела безперебійного живлення та інші технічні пристрої. Під час розробки вебзастосунку «SwagStore» тривала робота з комп'ютерною технікою вимагає створення комфортного акустичного середовища, оскільки надмірний шум може знижувати продуктивність і викликати втому.

Зменшення шуму найбільш ефективно тоді, коли заходи застосовуються безпосередньо в джерелі його виникнення. Такий підхід дозволяє не лише

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 67 |

покращити умови праці, а й продовжити строк служби обладнання, оскільки підвищений шум часто свідчить про технічні несправності, забруднення систем охолодження або неправильну експлуатацію пристроїв.

Першим заходом зменшення шуму є використання сучасного малошумного обладнання. Під час вибору комп'ютерної техніки доцільно надавати перевагу ноутбукам, системним блокам, блокам живлення та вентиляторам із низьким рівнем шуму. Сучасні моделі обладнання мають ефективніші системи охолодження, які забезпечують достатню продуктивність без надмірного звукового навантаження. Для офісної роботи та веброзробки доцільно використовувати пристрої, що відповідають вимогам енергоефективності та мають оптимізовану систему вентиляції.

Важливим технічним заходом є регулярне очищення обладнання від пилу. Накопичення пилу на вентиляторах, радіаторах і вентиляційних отворах призводить до погіршення охолодження, підвищення температури компонентів і збільшення швидкості обертання вентиляторів. У результаті рівень шуму зростає. Регулярне технічне обслуговування комп'ютерної техніки дозволяє підтримувати її стабільну роботу, зменшити шум і запобігти перегріванню.

Ще одним способом зниження шуму є правильне розміщення обладнання. Системні блоки, принтери та інші пристрої не повинні розташовуватися безпосередньо біля робочого місця, якщо це не є необхідним. За можливості шумні пристрої доцільно розміщувати на окремих столах, у спеціальних нішах або в технічних зонах. Серверне обладнання, яке може створювати значний шум, бажано встановлювати в окремому приміщенні або спеціальній серверній шафі з вентиляцією.

Для зменшення вібрації та шуму від обладнання використовують гумові підкладки, антивібраційні кріплення та стійкі поверхні. Вібрація від вентиляторів, жорстких дисків або принтерів може передаватися на стіл чи підлогу, посилюючи рівень шуму. Використання спеціальних підставок і правильне закріплення обладнання дозволяє значно зменшити такі небажані ефекти.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 68 |

Ефективним заходом є налаштування режимів енергозбереження та охолодження. Для комп'ютерів і ноутбуків можна використовувати збалансований режим роботи, який знижує навантаження на процесор і відеоадаптер під час виконання звичайних офісних або програмістських задач. Це зменшує потребу в інтенсивній роботі вентиляторів і, відповідно, знижує рівень шуму. Водночас важливо не допускати перегрівання техніки та не блокувати вентиляційні отвори.

Окрему увагу слід приділяти роботі периферійних пристроїв. Принтери, сканери та багатофункціональні пристрої можуть створювати короткочасний, але помітний шум під час друку або сканування. Для зменшення впливу такого шуму доцільно виконувати друк великих обсягів документів у визначений час або розташовувати периферійні пристрої на відстані від основного робочого місця. Також варто використовувати справні картриджі, якісні витратні матеріали та своєчасно проводити технічне обслуговування пристроїв.

Організаційні заходи також мають значення. Під час роботи над вебзастосунком «SwagStore» потрібно підтримувати порядок на робочому місці, не розміщувати сторонні предмети біля вентиляційних отворів, не перевантажувати обладнання зайвими процесами та стежити за його технічним станом. У разі появи незвичного шуму, скреготу або сильного гулу працівник повинен припинити використання пристрою та повідомити відповідальну особу.

Додатково доцільно періодично перевіряти справність охолоджувальних елементів комп'ютерної техніки. Це допомагає своєчасно виявляти причини підвищеного шуму та підтримувати комфортні умови праці.

Отже, зменшення шуму в джерелі його виникнення є важливою складовою охорони праці в ІТ-сфері. Для забезпечення комфортної роботи під час розробки вебзастосунку «SwagStore» необхідно використовувати справне та малошумне обладнання, регулярно проводити його обслуговування, правильно розміщувати пристрої, зменшувати вібрацію та контролювати режими роботи техніки. Такі заходи сприяють безпечним і сприятливим умовам праці.

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 69 |

ВИСНОВКИ

У процесі виконання дипломного проєкту було розроблено вебзастосунок для інтернет-магазину одягу «SwagStore». Метою роботи було створення системи, яка поєднує клієнтську частину для перегляду та замовлення товарів і адміністративну панель для керування товарами, категоріями, складськими залишками, замовленнями, відгуками та зверненнями користувачів.

У першому розділі було розглянуто предметну область інтернет-магазинів одягу, визначено основні проблеми ручного ведення обліку товарів і замовлень, а також обґрунтовано доцільність створення вебзастосунку. Було сформовано основні вимоги до системи, зокрема можливість перегляду каталогу, пошуку та фільтрації товарів, додавання товарів до кошика, оформлення замовлення, вибору розміру та адміністрування товарного наповнення.

У другому розділі було описано практичну реалізацію вебзастосунку «SwagStore». Клієнтську частину реалізовано з використанням Next.js, React та TypeScript, серверну частину — на основі Node.js, Express та TypeScript, а для зберігання даних використано MongoDB і Mongoose. Було створено основні сторінки користувача, каталог товарів, кошик, сторінку оформлення замовлення, сторінку замовлень та адміністративну панель.

У системі передбачено розподіл ролей користувача та адміністратора. Користувач може переглядати товари, застосовувати фільтри, обирати розмір, додавати товари до кошика, оформлювати замовлення та переглядати їхню історію. Адміністратор має можливість додавати й редагувати товари, працювати з категоріями, контролювати складські залишки, змінювати статуси замовлень, переглядати відгуки та звернення до підтримки.

Окрему увагу було приділено складському обліку. У вебзастосунку реалізовано керування залишками товарів за розмірами, кольорами та розташуванням на складі. Це дозволяє точніше контролювати наявність товарів і

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 70 |

запобігати додаванню до кошика більшої кількості товару, ніж є в наявності. Також у клієнтській частині передбачено відображення статусів наявності товарів.

У третьому розділі було розглянуто питання розміщення, налаштування, адміністрування та подальшої підтримки вебзастосунку. Було описано вимоги до середовища запуску, можливості локального та віддаленого розгортання, порядок роботи адміністратора із системою та можливі напрями подальшого розвитку проєкту.

У четвертому розділі було виконано економічне обґрунтування доцільності розробки вебзастосунку. Проведено розрахунок трудомісткості робіт, витрат на оплату праці, електроенергію, амортизаційних і накладних витрат, а також визначено собівартість і розрахункову ціну розробки. Отримані показники підтверджують економічну доцільність створення системи.

У п'ятому розділі було розглянуто питання охорони праці під час роботи з комп'ютерною технікою. Було визначено основні вимоги до організації робочого місця, освітлення, мікроклімату, електробезпеки, пожежної безпеки та режиму праці користувача.

Проведене тестування підтвердило працездатність основних функцій вебзастосунку. Було перевірено роботу головної сторінки, каталогу, фільтрів, сторінки товару, кошика, оформлення замовлення, демонстраційного вибору способу оплати, сторінки замовлень користувача та адміністративної панелі.

Отже, у результаті виконання дипломного проєкту було створено вебзастосунок «SwagStore», який може бути використаний як демонстраційна система інтернет-магазину одягу з елементами складського обліку. Розроблена система має зрозумілий інтерфейс, підтримує основні сценарії роботи користувача й адміністратора та може бути розширена в майбутньому для використання в умовах реального комерційного проєкту.

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| | | | | | | 71 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

ПЕРЕЛІК ПОСИЛАНЬ

1) Методичні вказівки до виконання дипломного проєкту для здобувачів освіти спеціальності 122 «Комп'ютерні науки». Тернопіль : ВСП «ТФК ТНТУ ім. І. Пулюя».

2) Next.js Documentation. Офіційна документація фреймворку Next.js для розробки вебзастосунків на основі React. URL: <https://nextjs.org/docs> (дата звернення: 10.06.2026).

3) React Documentation. Офіційна документація бібліотеки React для створення користувацьких інтерфейсів. URL: <https://react.dev/learn> (дата звернення: 10.06.2026).

4) TypeScript Documentation. Офіційна документація мови TypeScript, яка використовується для типізованої розробки JavaScript-застосунків. URL: <https://www.typescriptlang.org/docs/> (дата звернення: 10.06.2026).

5) Tailwind CSS Documentation. Офіційна документація CSS-фреймворку Tailwind CSS для створення адаптивного інтерфейсу. URL: <https://tailwindcss.com/docs> (дата звернення: 10.06.2026).

6) PostCSS Documentation. Документація інструменту PostCSS для обробки CSS-файлів у вебзастосунках. URL: <https://postcss.org/docs/> (дата звернення: 10.06.2026).

7) Node.js Documentation. Офіційна документація середовища виконання Node.js, яке використовується для створення серверної частини вебзастосунку. URL: <https://nodejs.org/docs/latest/api/> (дата звернення: 10.06.2026).

8) Express.js Documentation. Офіційна документація фреймворку Express.js для створення API та серверної логіки вебзастосунків. URL: <https://expressjs.com/> (дата звернення: 10.06.2026).

9) MongoDB Documentation. Офіційна документація системи керування базами даних MongoDB. URL: <https://www.mongodb.com/docs/> (дата звернення: 10.06.2026).

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| | | | | | | 72 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

10) MongoDB Atlas Documentation. Документація хмарного сервісу MongoDB Atlas для розміщення та адміністрування баз даних. URL: <https://www.mongodb.com/docs/atlas/> (дата звернення: 10.06.2026).

11) Mongoose Documentation. Офіційна документація бібліотеки Mongoose для роботи з MongoDB у середовищі Node.js. URL: <https://mongoosejs.com/docs/> (дата звернення: 10.06.2026).

12) JSON Web Token Introduction. Опис принципів використання JSON Web Token для авторизації користувачів у вебзастосунках. URL: <https://jwt.io/introduction> (дата звернення: 10.06.2026).

13) bcrypt.js. Документація бібліотеки bcryptjs, яка використовується для хешування паролів користувачів. URL: <https://www.npmjs.com/package/bcryptjs> (дата звернення: 10.06.2026).

14) npm Documentation. Документація менеджера пакетів npm, який використовується для встановлення залежностей проєкту. URL: <https://docs.npmjs.com/> (дата звернення: 10.06.2026).

15) Git Documentation. Документація системи контролю версій Git, яка використовується для роботи з кодом проєкту. URL: <https://git-scm.com/doc> (дата звернення: 10.06.2026).

16) MDN Web Docs. Довідкові матеріали щодо використання HTML, CSS, JavaScript та HTTP у веброзробці. URL: <https://developer.mozilla.org/> (дата звернення: 10.06.2026).

17) REST API Tutorial. Основні принципи побудови REST API для взаємодії клієнтської та серверної частин вебзастосунку. URL: <https://restfulapi.net/> (дата звернення: 10.06.2026).

18) Web.dev. Матеріали щодо створення сучасних, адаптивних і продуктивних вебзастосунків. URL: <https://web.dev/learn> (дата звернення: 10.06.2026).

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 73 |

ДОДАТКИ

Додаток А. Лістинг файлу client/package.json

```
{  
  "name": "my-v0-project",  
  "version": "0.1.0",  
  "private": true,  
  "scripts": {  
    "dev": "next dev",  
    "build": "next build",  
    "start": "next start",  
    "lint": "next lint"  
  },  
  "dependencies": {  
    "@radix-ui/react-dialog": "1.1.4",  
    "@radix-ui/react-select": "2.1.4",  
    "@radix-ui/react-tabs": "1.1.2",  
    "@radix-ui/react-toast": "1.2.4",  
    "autoprefixer": "^10.4.20",  
    "class-variance-authority": "^0.7.1",  
    "clsx": "^2.1.1",  
    "date-fns": "4.1.0",  
    "embla-carousel-react": "8.5.1",  
    "lucide-react": "^0.454.0",
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 74 |

```
"next": "15.2.4",
"react": "^19",
"react-dom": "^19",
"react-hook-form": "^7.54.1",
"recharts": "latest",
"sonner": "^1.7.1",
"tailwind-merge": "^2.5.5",
"tailwindcss-animate": "^1.0.7",
"zod": "^3.24.1"
},
"devDependencies": {
  "@types/node": "^22",
  "@types/react": "^19",
  "@types/react-dom": "^19",
  "postcss": "^8",
  "tailwindcss": "^3.4.17",
  "typescript": "^5"
}
}
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 75 |

Додаток Б. Лістинг файлу server/package.json

```
{
  "name": "mongoose-express-ts",
  "version": "1.0.0",
  "scripts": {
    "tsc": "tsc",
    "watch-tsc": "tsc -w",
    "deploy": "node dist/src/server.js",
    "watch-deploy": "nodemon dist/src/server.js",
    "server": "concurrently \"npm run watch-tsc\" \"npm run watch-deploy\"",
    "start": "npm run deploy"
  },
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "config": "^3.3.8",
    "cors": "^2.8.5",
    "dotenv": "^16.5.0",
    "express": "^4.18.2",
    "express-validator": "^6.14.2",
    "http-status-codes": "^2.2.0",
    "jsonwebtoken": "^8.5.1",
    "mongoose": "^6.13.8",
    "multer": "^1.4.5-lts.2",
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 76 |

```
"typescript": "^4.8.4"
},
"devDependencies": {
  "@types/bcryptjs": "^2.4.2",
  "@types/express": "^4.17.21",
  "@types/jsonwebtoken": "^8.5.9",
  "@types/mongoose": "^5.11.97",
  "@types/multer": "^1.4.12",
  "concurrently": "^7.4.0",
  "nodemon": "^2.0.20"
}
}
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 77 |

Додаток В. Лістинг файлу моделі товару Goods.ts

```
import { Document, model, Schema } from "mongoose";

export type TGoods = {

  id: string;

  name: string;

  section?: string;

  category: string;

  brand: string;

  description?: string;

  price: number;

  sizes: string[];

  color?: string;

  colors: string[];

  image: string;

  images?: string[];

  isNew?: boolean;

  showOnHomeHero?: boolean;

};

export interface IGoods extends Document {

  id: string;

  name: string;

  section?: string;
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 78 |

```

category: string;

brand: string;

description: string;

price: number;

sizes: string[];

color?: string;

colors: string[];

image: string;

images: string[];

isNew: boolean;

showOnHomeHero: boolean;
}

```

```

export const GoodsSchema = new Schema({
  id: { type: String, required: true, unique: true },
  name: { type: String, required: true },
  section: { type: String, default: "" },
  category: { type: String, required: true },
  brand: { type: String, required: true },
  description: { type: String, default: "" },
  price: { type: Number, required: true },
  sizes: { type: [String], required: true },
  color: { type: String, default: "" },
  colors: { type: [String], required: true },

```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 79 |

```
image: { type: String, default: "" },
images: { type: [String], default: [] },
isNew: { type: Boolean, default: false },
showOnHomeHero: { type: Boolean, default: false },
}, { timestamps: true });

const Goods = model<IGoods>("Goods", GoodsSchema);
export default Goods;
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 80 |

Додаток Г. Лістинг файлу моделі користувача User.ts

```
import { Document, model, Schema } from "mongoose";
```

```
export type UserRole = "user" | "admin";
```

```
export type TUser = {  
  name: string;  
  email: string;  
  passwordHash: string;  
  role: UserRole;  
  phone?: string;  
  city?: string;  
  deliveryAddress?: string;  
  postalCode?: string;  
  createdAt: Date;  
};
```

```
export interface IUser extends Document {  
  name: string;  
  email: string;  
  passwordHash: string;  
  role: UserRole;  
  phone?: string;
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 81 |

```

    city?: string;

    deliveryAddress?: string;

    postalCode?: string;

    createdAt: Date;
}

export const UserSchema = new Schema({
  name: { type: String, required: true, trim: true },
  email: { type: String, required: true, unique: true, lowercase: true, trim: true },
  passwordHash: { type: String, required: true },
  role: { type: String, enum: ["user", "admin"], default: "user", required: true },
  phone: { type: String, trim: true },
  city: { type: String, trim: true },
  deliveryAddress: { type: String, trim: true },
  postalCode: { type: String, trim: true },
}, {
  timestamps: { createdAt: true, updatedAt: false },
});

const User = model<IUser>("User", UserSchema);

export default User;

```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 82 |

Додаток Д. Лістинг файлу моделі замовлення Order.ts

```
import { Document, model, Schema } from "mongoose";
```

```
export type TOrderItem = {
```

```
  productId: string;
```

```
  name: string;
```

```
  price: number;
```

```
  quantity: number;
```

```
  image?: string;
```

```
  size?: string;
```

```
  category?: string;
```

```
};
```

```
export interface IOrder extends Document {
```

```
  userId: string;
```

```
  customerName: string;
```

```
  customerEmail: string;
```

```
  recipientName?: string;
```

```
  phone?: string;
```

```
  city?: string;
```

```
  deliveryAddress?: string;
```

```
  postalCode?: string;
```

```
  comment?: string;
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 83 |

```
items: TOrderItem[];
itemsTotal?: number;
subtotal?: number;
deliveryPrice?: number;
totalPrice: number;
paymentMethod?: string;
paymentStatus?: string;
status: string;
createdAt: Date;
updatedAt: Date;
}
```

```
const OrderItemSchema = new Schema({
  productId: { type: String, required: true },
  name: { type: String, required: true },
  price: { type: Number, required: true },
  quantity: { type: Number, required: true },
  image: { type: String },
  size: { type: String },
  category: { type: String },
}, { _id: false });
```

```
export const OrderSchema = new Schema({
  userId: { type: String, required: true, index: true },
```

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | <i>2026.KBP.122.421.14.00.00 ПЗ</i> | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 84 |

```

customerName: { type: String, required: true },
customerEmail: { type: String, required: true },
recipientName: { type: String },
phone: { type: String },
city: { type: String },
deliveryAddress: { type: String },
postalCode: { type: String },
comment: { type: String },
items: { type: [OrderItemSchema], required: true },
itemsTotal: { type: Number },
subtotal: { type: Number },
deliveryPrice: { type: Number, default: 0 },
totalPrice: { type: Number, required: true },
paymentMethod: { type: String },
paymentStatus: { type: String },
status: { type: String, default: "Активне", required: true },
}, { timestamps: true });

```

```
const Order = model<IOrder>("Order", OrderSchema);
```

```
export default Order;
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 85 |

Додаток Е. Лістинг файлу моделі складського обліку Inventory.ts

```
import { Document, model, Schema } from "mongoose";
```

```
export type TInventory = {
```

```
  id: string;
```

```
  goodId: string;
```

```
  quantity: number;
```

```
  warehouse: string;
```

```
  color: string;
```

```
  size: string;
```

```
};
```

```
export interface IInventory extends Document {
```

```
  id: string;
```

```
  goodId: string;
```

```
  quantity: number;
```

```
  warehouse: string;
```

```
  color: string;
```

```
  size: string;
```

```
  createdAt: Date;
```

```
  updatedAt: Date;
```

```
}
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 86 |

```

export const InventorySchema = new Schema({
  id: { type: String, required: true, unique: true },
  goodId: { type: String, required: true },
  quantity: { type: Number, required: true },
  warehouse: { type: String, required: true },
  color: { type: String, required: true },
  size: { type: String, required: true },
}, {
  timestamps: true
});

const Inventory = model<IInventory>("Inventory", InventorySchema);

export default Inventory;

```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 87 |

Додаток Ж. Лістинг файлу маршруту авторизації auth.ts

```
const createToken = (user: IUser): string => {  
  return jwt.sign(  
    { userId: user._id.toString() },  
    getJwtSecret(),  
    { expiresIn: getJwtExpiration() }  
  );  
};  
  
router.post("/login", async (req: AuthRequest, res: Response) => {  
  try {  
    const { email, password } = req.body;  
  
    if (!email || !password) {  
      return res.status(400).json({ message: "Введіть email і пароль" });  
    }  
  
    const normalizedEmail = String(email).trim().toLowerCase();  
    const user = await User.findOne({ email: normalizedEmail });  
  
    if (!user) {  
      return res.status(401).json({ message: "Невірний email або пароль" });  
    }  
  }  
}
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 88 |

```
const isPasswordValid = await bcrypt.compare(String(password),
user.passwordHash);

if (!isPasswordValid) {
    return res.status(401).json({ message: "Невірний email або пароль" });
}

return res.json({
    token: createToken(user),
    user: toUserResponse(user),
});
} catch (error) {
    return res.status(500).json({ message: "Помилка при вході" });
}
});
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 89 |

Додаток И. Лістинг файлу маршруту товарів goods.ts

```
router.get("/", async (_req: Request, res: Response) => {  
  try {  
    const goods = await Goods.find().sort({ createdAt: -1, _id: -1 });  
    return res.status(HttpStatusCodes.OK).json({ goods });  
  } catch (error) {  
    console.error(error);  
    return res.status(HttpStatusCodes.INTERNAL_SERVER_ERROR).json({  
      error: "Помилка при отриманні товарів",  
    });  
  }  
});  
  
router.post("/", authMiddleware, adminMiddleware, async (req: Request, res: Response)  
=> {  
  try {  
    const {  
      name,  
      section,  
      category,  
      brand,  
      description,  
      price,  
      sizes,    }  
  }  
});
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 90 |

```

    color,
    colors,
    image,
    images,
    isNew,
    showOnHomeHero,
  }: Omit<TGoods, "id"> = req.body;

const lastItem = await Goods.findOne({})
  .sort({ id: -1 })
  .collation({ locale: "en_US", numericOrdering: true });

let newId: string;

if (lastItem && lastItem.id) {
  const lastNumber = parseInt(lastItem.id.replace("PROD", ""), 10);
  const nextNumber = lastNumber + 1;
  newId = "PROD" + nextNumber.toString().padStart(3, "0");
} else {
  newId = "PROD001";
}

const normalizedImages = normalizeImages(image, images);
const normalizedColor = String(color || colors?.[0] || "").trim();

```

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 91 |

```
const newGoods = new Goods({
    id: newId,
    name,
    section: String(section || "").trim(),
    category,
    brand,
    description: String(description || "").trim(),
    price,
    sizes,
    color: normalizedColor,
    colors: normalizedColor ? [normalizedColor] : colors,
    image: normalizedImages[0] || image || "",
    images: normalizedImages,
    isNew: Boolean(isNew),
    showOnHomeHero: Boolean(showOnHomeHero),
});
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 92 |

Додаток К. Лістинг файлу маршруту замовлень orders.ts

```
router.post("/", authMiddleware, async (req: AuthRequest, res: Response) => {  
  try {  
    if (!req.user) {  
      return res.status(401).json({ message: "Потрібно увійти в акаунт" });  
    }  
  
    const incomingItems = req.body.items as IncomingOrderItem[] | undefined;  
    const requestedTotal = Number(req.body.totalPrice);  
    const recipientName = String(req.body.recipientName || "").trim();  
    const phone = String(req.body.phone || "").trim();  
    const city = String(req.body.city || "").trim();  
  
    const deliveryAddress = normalizeDeliveryAddress(city,  
String(req.body.deliveryAddress || ""));  
  
    const postalCode = String(req.body.postalCode || "").trim();  
    const comment = String(req.body.comment || "").trim();  
    const requestedDeliveryPrice = Number(req.body.deliveryPrice || 0);  
    const paymentMethod = String(req.body.paymentMethod ||  
PAYMENT_METHODS[0]).trim();  
  
    if (!Array.isArray(incomingItems) || incomingItems.length === 0) {  
      return res.status(400).json({ message: "Кошик порожній" });  
    }  
  }  
}
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 93 |

```

const items: TOrderItem[] = [];

for (const item of incomingItems) {
    const productId = String(item.productId || item.id || "").trim();
    const quantity = normalizeQuantity(item.quantity);
    const product = await Goods.findOne({ id: productId });

    if (!product) {
        return res.status(404).json({ message: `Товар ${productId} не
знайдено` });
    }

    items.push({
        productId: product.id,
        name: product.name,
        price: Number(product.price),
        quantity,
        image: product.image || item.image,
        size: item.size,
        category: product.category || item.category,
    });
}

const computedItemsTotal = roundMoney(items.reduce((sum, item) => {

```

| | | | | | | |
|-----|------|----------|--------|------|-------------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 94 |

```
return sum + item.price * item.quantity;  
}, 0));
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 95 |

Додаток Л. Лістинг файлу маршруту складського обліку `inventory.ts`

```
router.post('/bulk', authMiddleware, adminMiddleware, async (req, res) => {  
  try {  
    const goodId = String(req.body.goodId || "").trim();  
    const items = Array.isArray(req.body.items) ? req.body.items : [];  
  
    if (!goodId) {  
      return res.status(400).json({ message: 'Product is required' });  
    }  
  
    if (items.length === 0) {  
      return res.status(400).json({ message: 'Inventory rows are required' });  
    }  
  
    const savedItems = [];  
  
    for (const item of items) {  
      const size = String(item.size || 'Без розміру').trim();  
      const color = String(item.color || 'Без кольору').trim();  
      const warehouse = String(item.warehouse || 'Склад А').trim();  
      const quantity = Math.max(0, Math.floor(Number(item.quantity || 0)));  
  
      const existingInventory = await Inventory.findOne({
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 96 |

```

        goodId,
        size,
        color,
        warehouse,
    });

    if (existingInventory) {
        const previousQuantity = Number(existingInventory.quantity || 0);
        existingInventory.quantity = quantity;
        await existingInventory.save();

        if (previousQuantity !== quantity) {
            await InventoryHistory.create({
                inventoryId: existingInventory.id,
                goodId: existingInventory.goodId,
                previousQuantity,
                newQuantity: quantity,
                type: quantity >= previousQuantity ? 'incoming' :
'outgoing',
                warehouse: existingInventory.warehouse,
            });
        }
    }

```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 97 |

Додаток М. Лістинг файлу сторінки детального перегляду товару

```
const stockByProduct = useMemo(() => {  
  return inventory.reduce<Record<string, number>>((totals, item) => {  
    totals[item.goodId] = (totals[item.goodId] || 0) + Number(item.quantity || 0);  
    return totals;  
  }, {});  
}, [inventory]);  
  
const stockByProductSize = useMemo(() => {  
  return inventory.reduce<Record<string, number>>((totals, item) => {  
    const key = `${item.goodId}::${item.size || ""}`;  
    totals[key] = (totals[key] || 0) + Number(item.quantity || 0);  
    return totals;  
  }, {});  
}, [inventory]);  
  
const stock = product ? stockByProduct[getProductId(product)] : undefined;  
const displayedStock = product && selectedSize  
  ? stockByProductSize[`${getProductId(product)}::${selectedSize}`] || 0  
  : stock;  
  
const handleAddToCart = () => {  
  if (!product || outOfStock) return;  
}
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 98 |

```

if (!user) {
    setLoginPromptOpen(true);
    return;
}

if (!selectedSize) {
    setSizeError("Оберіть розмір перед додаванням у кошик");
    return;
}

const selectedStock = selectedSize
    ? stockByProductSize[`${getProductId(product)}::${selectedSize}`] || 0
    : stock;

if (typeof selectedStock === "number") {
    const currentQuantity = readCartItems()
        .filter((item) =>
            item.productId === getProductId(product)
            && (item.size || "") === (selectedSize || ""))
        .reduce((sum, item) => sum + item.quantity, 0);

    if (currentQuantity + 1 > selectedStock) {

```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 99 |

```
setSizeError("Недостатньо товару на складі");  
  
toast({ title: "Недостатньо товару на складі", variant: "destructive" });  
  
return;  
  
}  
  
}  
  
};
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 100 |

Додаток Н. Лістинг файлу сторінки оформлення замовлення

```
const validateForm = () => {  
  
  const nextErrors: CheckoutErrors = {};  
  
  if (!form.recipientName.trim()) {  
    nextErrors.recipientName = "Вкажіть ім'я отримувача";  
  }  
  
  if (!form.phone.trim()) {  
    nextErrors.phone = "Вкажіть номер телефону";  
  }  
  
  if (!form.city.trim()) {  
    nextErrors.city = "Вкажіть місто";  
  }  
  
  if (!form.deliveryAddress.trim()) {  
    nextErrors.deliveryAddress = "Вкажіть адресу доставки";  
  }  
  
  if (form.postalCode.trim() && !/^\d{5}$/.test(form.postalCode.trim())) {  
    nextErrors.postalCode = "Поштовий індекс має містити рівно 5 цифр";  
  }  
}
```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 101 |

```

setErrors(nextErrors);

return Object.keys(nextErrors).length === 0;
};

const handleSubmit = async (event: React.FormEvent) => {
  event.preventDefault();

  if (!user) {
    setSubmitError("Щоб оформити замовлення, увійдіть в акаунт.");
    return;
  }

  if (cartItems.length === 0) {
    setSubmitError("Кошик порожній");
    return;
  }

  if (!validateForm()) {
    return;
  }

  const quantitiesByProductSize = cartItems.reduce<Record<string, CartItem>>((totals,
item) => {

```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.КВР.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 102 |

```

const key = `${item.productId}::${item.size || ""}`;

totals[key] = {
  ...item,
  quantity: (totals[key]?.quantity || 0) + item.quantity,
};

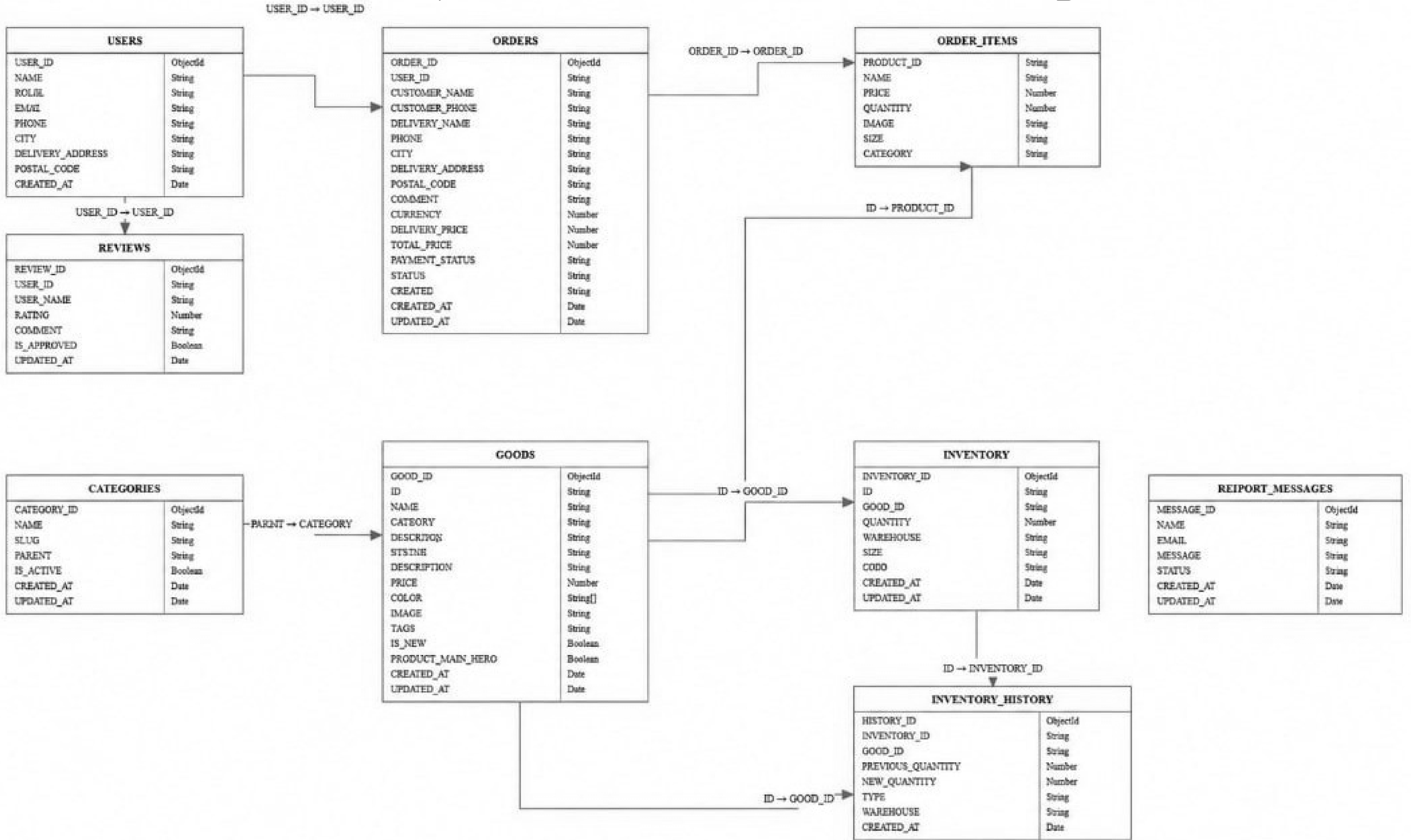
return totals;
}, {});

const hasStockIssue = Object.values(quantitiesByProductSize).some((item) => {
  const stock = getStockForCartItem(item);
  return typeof stock === "number" && item.quantity > stock;
});
};

```

| | | | | | | |
|-----|------|----------|--------|------|------------------------------|------|
| | | | | | 2026.KBP.122.421.14.00.00 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 103 |

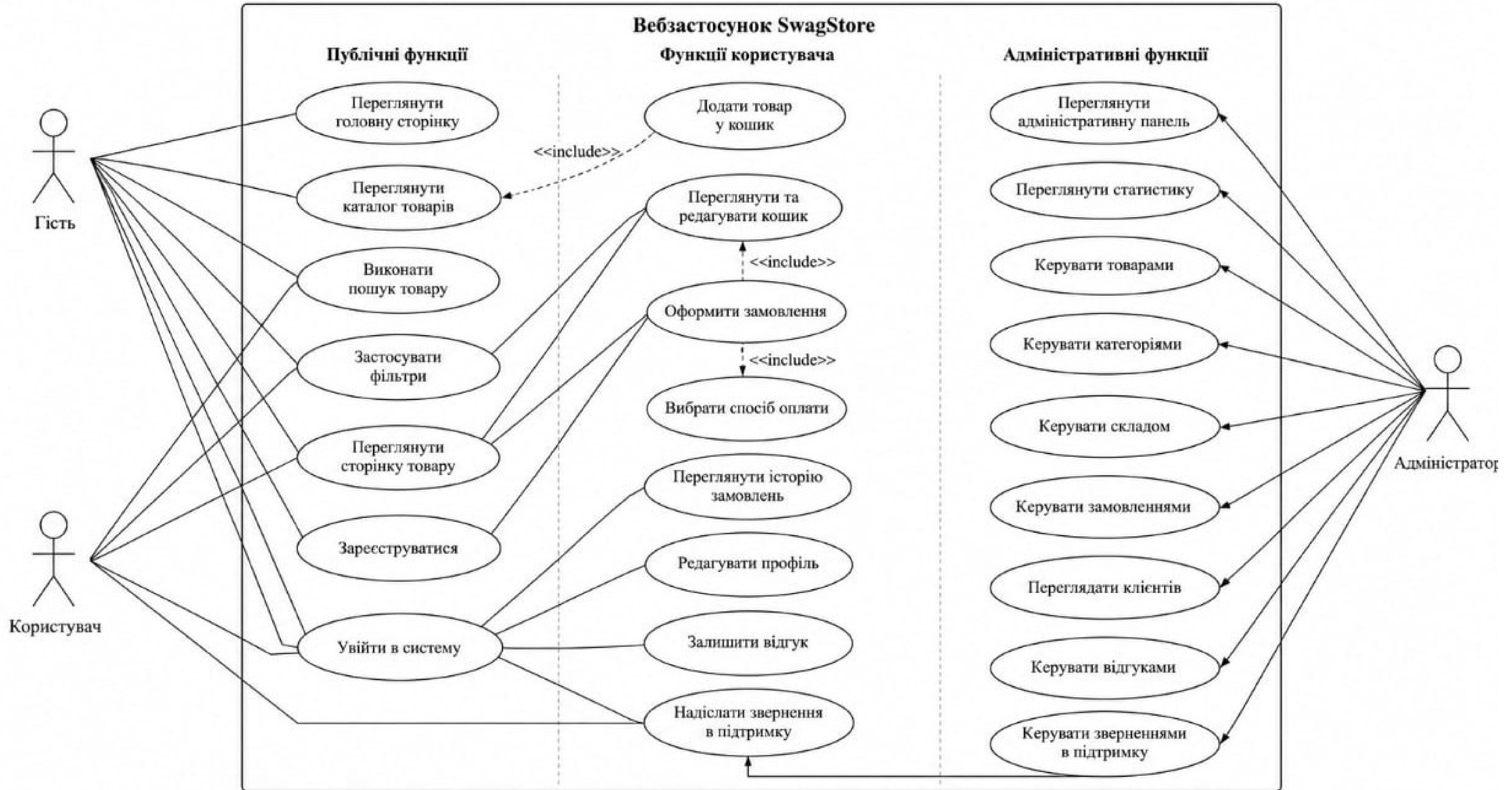
ER-діаграма бази даних вебсайту



| | | | | | | | | |
|------------------------------|--------------|----------|-------|------|---|--------------------|-------|---------|
| 2026.KBP.122.421.14.00.00 БД | | | | | | | | |
| Зм. | Арх. | № докум. | Підп. | Дата | Розробка вебдодатку для обліку товарів магазину одягу "SwagStore" | Лист | Маса | Масштаб |
| Розроб. | Молоди М.О. | | | | | | | |
| Перев. | Корнати І.Б. | | | | | | | |
| Текст. | | | | | | Архив | Архив | 1 |
| Рецензент | Прішнік В.А. | | | | | ВП ТФК ТНТУ КН-421 | | |
| Мод. № опрац. | | | | | м. Тернопіль | | | |
| Мод. і дата | | | | | Формат А1 | | | |
| Мод. № опрац. | | | | | Затв. | | | |

Лист. заголовок
Лист. №
Лист. і дата
Лист. № опрац.

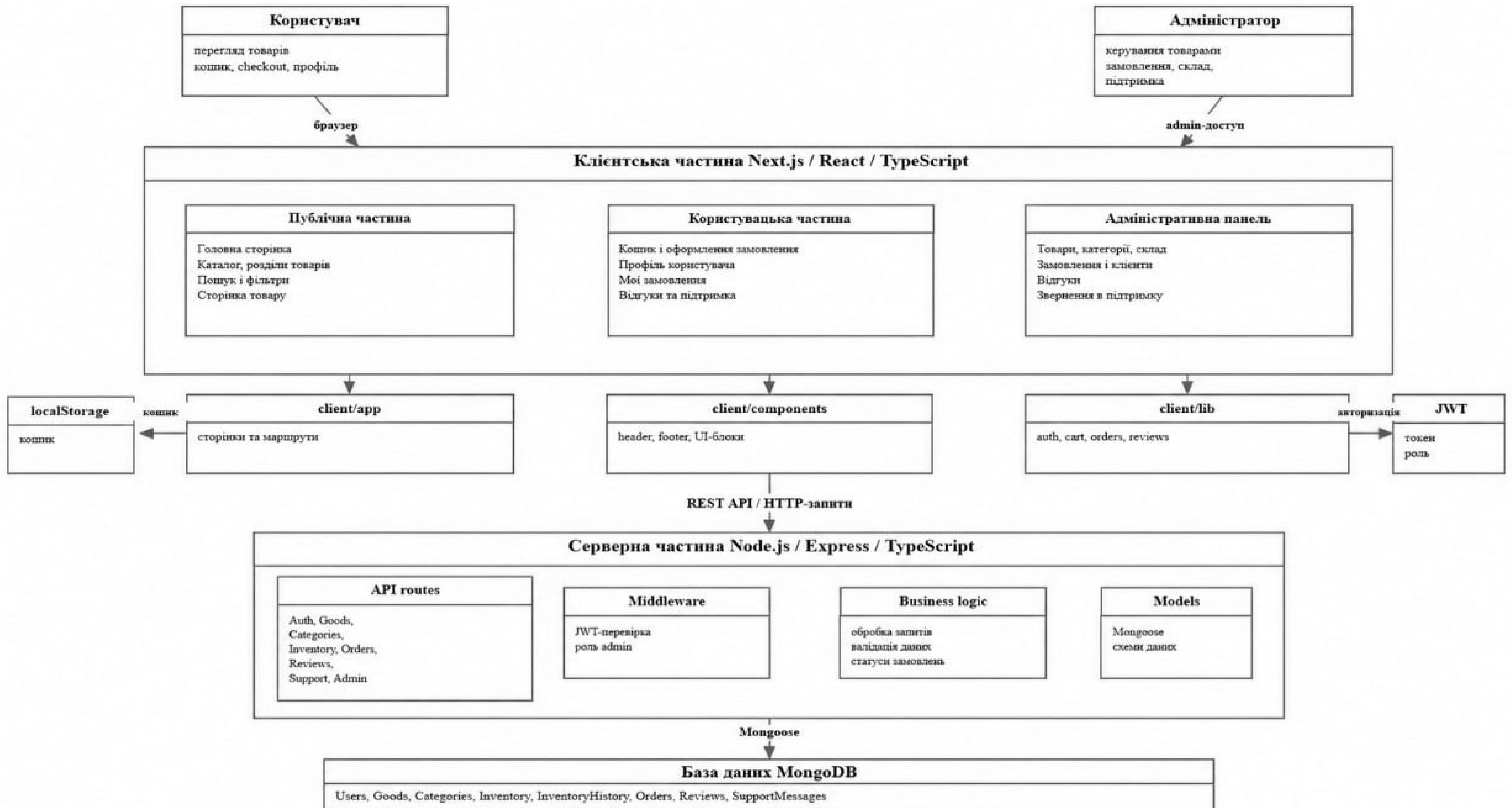
UML-діаграма варіантів використання програми



Перш. застос. _____
 Стор. № _____
 Підп. і дата _____
 Зам. № _____
 Підп. і дата _____
 № арк. _____

| | | | | | | | |
|-----------|---------------|--------|-------|------|---|-------|---------|
| | | | | | 2026.KBP.122.421.14.00.00 ДВ | | |
| Зм. | Арк. | № док. | Підп. | Дата | Розробка вебдодатку для обліку товарів магазину одягу "SwagStore" | | |
| Розроб. | Малиш М. О. | | | | Лист | Маса | Масштаб |
| Перев. | Корочка І. Б. | | | | | | |
| І.контр. | | | | | Архив | Архив | 1 |
| Рецензент | | | | | ВП ТФК ТНТУ КН-421 | | |
| І.контр. | Приймак В.А. | | | | м. Тернопіль | | |
| Затв. | | | | | Формат А1 | | |

Структурна схема клієнтської частини сайту



Перш. запис.
Стор. №
Ліст. і дата.
Зач. № і дата.
№ і дата.

| | | | | | | | |
|------------------------------|----------------|--------|-------|------|---|-------|---------|
| 2026.KBP.122.421.14.00.00 CC | | | | | Лист | Маса | Масштаб |
| Зм. | Арх. | № док. | Підп. | Дата | Розробка вебдодатку для обліку товарів магазину одягу "SwagStore" | | |
| Розроб. | Молоди М. О. | | | | | | |
| Перев. | Корачула І. Б. | | | | Архив | Архив | 1 |
| І. контр. | | | | | ВП ТФК ТНТУ КН-421 | | |
| Рецензент | Приміч В. А. | | | | м. Тернопіль | | |
| Н. контр. | | | | | | | |
| Затв. | | | | | Формат А1 | | |

Таблиця техніко-економічних показників

| № | Показник | Одиниці вимірювання | Значення |
|----|---|---------------------|---|
| 1 | Платформа програмного продукту | – | Вебзастосунок |
| 2 | Тип інтерфейсу | – | Графічний вебінтерфейс |
| 3 | Мови програмування | – | TypeScript, JavaScript |
| 4 | Основні технології | – | Next.js, React, Node.js, Express.js, Tailwind CSS |
| 5 | База даних | – | MongoDB / Mongoose |
| 6 | Кількість файлів програмної частини | шт | 150 |
| 7 | Розмір програмної частини без службових папок | МБ | 0,86 |
| 8 | Собівартість | грн | 67 172 |
| 9 | Плановий прибуток | грн | 37 616 |
| 10 | Ціна | грн | 104 788 |
| 11 | Чиста теперішня вартість | грн | 23 176 |
| 12 | Термін окупності | рік | 1,8 |

| | | | | | | | |
|-----------|--------------|--------|-------|------|---|-------|---------|
| | | | | | 2026.KBP.122.421.14.00.00 ТБ | | |
| Зм. | Арк. | № док. | Підп. | Дата | Розробка введодатку для обліку товарів магазину одягу "SwagStore" | | |
| Розроб. | Малиш М.О. | | | | Лист | Маса | Масштаб |
| Перев. | Капача І.Б. | | | | | | |
| І.контр. | | | | | Аркци | Аркци | 1 |
| Рецензент | | | | | ВП ТФК ТНТУ КН-421 | | |
| І.контр. | Пріймак В.А. | | | | м. Тернопіль | | |
| Затв. | | | | | Формат А1 | | |