

**Вищий навчальний заклад**  
**«Університет економіки та права «КРОК»**  
**Фаховий коледж**

Циклова комісія з інформаційних технологій

**Кваліфікаційна робота фахового молодшого  
бакалавра**

на тему Автоматизована система продажу мобільних телефонів через  
інтернет-магазин

Виконав \_\_\_\_\_

(Підпис)

Ромашко Владислав Юрійович

(прізвище, ім'я, по батькові)

Науковий керівник \_\_\_\_\_

Добришин Юрій Євгенович

(прізвище, ім'я, по батькові)

(Резолюція «До захисту»)

**Попередній захист:**

(Висновок: “До захисту в екзаменаційній комісії”)

**Голова циклової комісії**

\_\_\_\_\_ (Підпис)

\_\_\_\_\_ (Прізвище, ініціали)

\_\_\_\_\_ (Дата)

**Київ – 2025 року**

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД**  
**УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»**

Фаховий коледж  
Циклова комісія з інформаційних технологій  
Спеціальність 121 інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Голова циклової комісії \_\_\_\_\_ Леонід УВАРОВ  
(підпис)

«\_\_\_» \_\_\_\_\_ 2025 року

**ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Здобувач освіти Ромашко Владислав Юрійович

1. Тема роботи «Автоматизована система продажу мобільних телефонів через інтернет-магазин»

затверджена наказом по університету від «\_\_\_» \_\_\_\_\_ 202\_\_ р. № \_\_\_\_\_

2. Термін здачі закінченої роботи «30» травня 2025 року

3. Вихідні дані до роботи:

- а) цільова аудиторія – існуючі автоматизовані системи продажу мобільних телефонів, що застосовуються через інтернет магазин;
- б) функціональність – отримання повідомлень, їх обробка, надсилання відповідей, обробка інформації, тощо;
- в) технічні вимоги - платформа для розробки : сучасних автоматизованих систем продажу мобільних телефонів, мова програмування, база даних та інші технології;
- г) можливість інтеграції (взаємодії) з існуючими сучасними програмними додатками;
- д) існуючі рішення та кращі практики у сфері розробки сучасних автоматизованих систем продажу мобільних телефонів.

4. Зміст пояснювальної записки

- а) Розділ 1 Теоретична частина. Аналіз існуючих рішень та кращих практик у сфері розробки сучасних автоматизованих систем продажу мобільних телефонів, обґрунтування вибору платформи та технології для розроблення проекту.
- б) Розділ 2 Проектування та розробка. Створення зручного та інтуїтивно зрозумілого інтерфейсу для спілкування, розробка алгоритму, за яким буде працювати автоматизована система продажу мобільних телефонів, робота з користувачами.
- в) Розділ 3 Експериментальна частина. Перевірка роботи системи, виявлення та виправлення помилок, аналіз того, наскільки програмний продукт відповідає поставленим завданням та задовольняє потреби користувачів, інструкція користувачам (для технічного персоналу – опис процесу розробки автоматизованої системи, її архітектури та функціональності, для звичайних користувачів - як

користуватися інформацією та отримувати необхідну допомогу). Рекомендації щодо забезпечення конфіденційності інформації та дотримання цілісності інформації.

5. Перелік графічного матеріалу:

Скріншоти існуючих рішень

Скріншоти інтерфейсу продукту

Схеми і таблиці щодо візуалізації аналізу

Блок-схеми алгоритмів

Діаграми щодо проєктування продукту (наприклад, потоків даних, переходів станів, сутність-зв'язок, UML, бази даних тощо)

Дата видачі завдання «12» лютого 2025 року

Науковий керівник

\_\_\_\_\_

(підпис)

Юрій ДОБРИШИН

Завдання прийняв до виконання

\_\_\_\_\_

(підпис)

Владислав РОМАШКО

## РЕФЕРАТ

Пояснювальна записка: 59 сторінок, 16 рисунків, 4 таблиць, 5 додатків, 15 джерел.  
Об'єкт дослідження – процес автоматизації продажу мобільних телефонів через інтернет-магазин.  
Мета роботи – розробка веб-орієнтованої автоматизованої системи, що забезпечує управління товарами, обробку замовлень, реєстрацію користувачів та інтеграцію з платіжними сервісами.  
У кваліфікаційній роботі проаналізовано сучасні підходи до реалізації онлайн-торгівлі, вивчено популярні платформи та технології електронної комерції. На основі отриманих даних створено прототип інтернет-магазину з використанням мови програмування JavaScript, HTML, SCC. Інтерфейс реалізовано з використанням HTML, CSS та бібліотеки React.  
Система підтримує функціонал перегляду товарів, фільтрації, пошуку, реєстрації користувачів, оформлення замовлень. У роботі описано архітектуру веб-додатку, реалізовані модулі, структуру бази даних, інтерфейс адміністратора та користувача. Систему протестовано на різних пристроях, підготовлено технічну документацію та надано рекомендації щодо її подальшого розвитку.  
Результати розробки можуть бути використані для створення або вдосконалення веб-магазинів у сфері продажу мобільних пристроїв.  
Ключові слова: інтернет-магазин, автоматизація продажу, мобільні телефони, JavaScript, Node.js, Express, електронна комерція, управління товарами, замовлення.

## ABSTRACT

Explanatory note: 59 pages, 16 figures, 4 tables, 5 appendices, 15 sources.

The object of the study is the process of automating the sale of mobile phones through an online store. The goal of the robot is the development of a web-oriented automated system that provides product management, order processing, user registration and integration with payment services. The qualification work analyses modern approaches to the implementation of online trading, studied popular platforms and technologies of e-commerce. Based on the received data, a prototype of an online store was created using the programming language JavaScript, HTML, SCC. The interface is implemented using HTML, CSS and React libraries. The system supports the functionality of product viewing, filtering, search, user registration, ordering. The robot describes the architecture of the web application, the implemented modules, the database structure, the administrator and user interface. The system has been tested on various devices, technical documentation has been prepared and recommendations for its further development have been provided. The results of the development can be used to create or improve web stores in the field of mobile device sales.

Keywords: online store, sales automation, mobile phones, JavaScript, Node.js, Express, e-commerce, product management, order.

## ЗМІСТ

СКРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.....	7
ВСТУП.....	8
РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ВИБІР ТЕХНОЛОГІЙ.....	10
1.1. Огляд сучасних автоматизованих систем продажу.....	10
1.1.1. Характеристики популярних рішень.....	12
1.1.2. Порівняльний аналіз функціональних можливостей.....	14
1.2. Аналіз кращих практик у сфері онлайн-продажів мобільних телефонів..	16
1.3. Вибір платформи та технологій для реалізації системи.....	17
1.4. Обґрунтування архітектурного підходу.....	19
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ.....	21
2.1. Загальна архітектура системи.....	21
2.2. Розробка інтерфейсу користувача.....	22
2.2.1. Вимоги до UX/UI.....	24
2.2.2. Прототипування інтерфейсу.....	26
2.3. Алгоритми роботи системи.....	27
2.3.1. Алгоритм реєстрації користувача.....	28
2.3.2. Алгоритм входу користувача.....	28
2.3.3. Алгоритм додавання товару до кошика.....	28
2.3.4. Алгоритм оформлення замовлення.....	28
2.3.5. Алгоритм фільтрації товарів.....	29
2.3.6. Алгоритм збереження сесії користувача.....	29
2.4. Реалізація серверної та клієнтської частин.....	29
2.4.1. Клієнтська частина (frontend).....	30
2.4.2. Серверна частина (backend).....	30
2.4.4. Переваги обраної реалізації.....	31
РОЗДІЛ 3. ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА ТА ТЕСТУВАННЯ СИСТЕМИ.....	33
3.1. Методика тестування функціональності.....	33
3.1.1. Мета тестування.....	33

3.1.2. Методика тестування .....	33
3.1.3. Основні сценарії тестування .....	34
3.1.4. Середовище тестування .....	34
3.1.5. Виявлені помилки та їх усунення .....	34
3.2. Виявлення та усунення помилок .....	35
3.2.1. Класифікація виявлених помилок .....	35
3.2.2. Приклади виявлених помилок і рішень .....	35
3.2.3. Стратегії покращення стабільності .....	36
3.4. Інструкція користувача .....	36
3.5. Рекомендації щодо безпеки: .....	37
3.5.1. Захист персональних даних .....	37
3.5.2. Підтримка цілісності даних .....	38
ВИСНОВКИ .....	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	41
ДОДАТКИ .....	43
Додаток А Скріншоти існуючих рішень .....	43
Додаток Б. Скріншоти розробленого інтерфейсу .....	46
Додаток В. Фрагменти коду .....	49
Додаток Г. Блок-схеми алгоритмів .....	58
Додаток Е. Діаграми .....	60

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

**API** — Application Programming Interface (інтерфейс програмування застосунків)

**CSS** — Cascading Style Sheets (каскадні таблиці стилів)

**DB** — Database (база даних)

**GUI** — Graphical User Interface (графічний інтерфейс користувача)

**HTTP** — HyperText Transfer Protocol (протокол передачі гіпертексту)

**JSON** — JavaScript Object Notation (формат обміну даними)

**Node.js** — серверна платформа для виконання JavaScript

**REST** — Representational State Transfer (архітектурний стиль взаємодії в мережі)

**UI** — User Interface (користувацький інтерфейс)

**UX** — User Experience (користувацький досвід)

**HTML** — HyperText Markup Language (мова розмітки гіпертексту)

**ER** — Entity-Relationship (сутність-зв'язок)

**JS** — JavaScript (мова програмування)

## ВСТУП

**Актуальність завдання.** Сучасний ринок мобільних телефонів постійно зростає, що зумовлює високу конкуренцію серед компаній, які пропонують цей товар. Автоматизовані системи продажу стають необхідним інструментом для забезпечення зручного, швидкого та ефективного процесу покупки. Впровадження таких систем дозволяє підвищити рівень обслуговування клієнтів, оптимізувати бізнес-процеси та розширити канали збуту [1], [3].

**Мета роботи.** Розробити автоматизовану систему продажу мобільних телефонів, що забезпечує інтуїтивно зрозумілий інтерфейс користувача, функціональність для пошуку, вибору товарів, оформлення замовлення, а також надійний захист персональних даних [2], [14].

### **Завдання роботи.**

- Проаналізувати існуючі рішення та кращі практики у сфері онлайн-продажу мобільних телефонів [1], [3];
- Обрати технології та платформу для розробки системи [4], [5];
- Спроекувати архітектуру та інтерфейс користувача [2], [7];
- Розробити алгоритми роботи системи [6], [12];
- Реалізувати серверну і клієнтську частини системи [4], [8];
- Провести тестування та виправлення помилок [10];
- Підготувати інструкції для користувачів і технічного персоналу [13];
- Забезпечити безпеку та цілісність даних [14].

**Об'єкт дослідження.** Автоматизовані системи онлайн-продажу мобільних телефонів.

**Предмет дослідження.** Методи та технології розробки інтерактивних веб-додатків для реалізації функцій продажу мобільних телефонів із забезпеченням безпеки та зручності користування [2], [6].

**Методи дослідження.** Аналіз наукової літератури та існуючих рішень, порівняльний аналіз функціональності, моделювання системи, прототипування інтерфейсу, програмна реалізація, тестування та налагодження [1], [2], [3].

**Практичне значення одержаних результатів.** Розроблена система може бути використана для автоматизації продажів у невеликих та середніх інтернет-магазинах мобільних телефонів, що сприятиме підвищенню ефективності роботи, поліпшенню користувацького досвіду та зменшенню людського фактору у процесах продажу [1], [14].

**Структура роботи.** Дипломна робота складається зі вступу, трьох основних розділів, висновків, списку використаних джерел та додатків. У першому розділі наведено теоретичний аналіз існуючих рішень і вибір технологій. Другий розділ присвячено проєктуванню та розробці системи, а третій — експериментальній перевірці, тестуванню та оформленню документації [2].

## РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ВИБІР ТЕХНОЛОГІЙ

### 1.1. Огляд сучасних автоматизованих систем продажу

Сучасні автоматизовані системи продажу є невід’ємною частиною електронної комерції та активно використовуються для оптимізації процесу реалізації товарів і послуг [1], [2]. Вони дозволяють компаніям зменшити витрати на обслуговування клієнтів, прискорити обробку замовлень, підвищити точність виконання транзакцій та забезпечити кращу взаємодію з покупцями [1], [3].

Автоматизовані системи продажу мобільних телефонів зазвичай включають такі компоненти:

- каталог товарів із можливістю фільтрації й сортування [4];
- систему обробки замовлень [5];
- інтеграцію з платіжними сервісами [5];
- базу даних клієнтів і їхніх замовлень (CRM) [11];
- аналітичний модуль для відстеження продажів [3];
- інструменти маркетингу (знижки, рекомендації, email-розсилки) [13];
- технічну підтримку користувачів через чат або форму зворотного зв’язку [10].

У процесі дослідження було проаналізовано низку популярних рішень, які активно застосовуються як на українському, так і на світовому ринку. Серед них варто виокремити такі системи, як:

- **Rozetka** — одна з найбільших онлайн-платформ в Україні, яка забезпечує повноцінний процес покупки мобільних пристроїв: від вибору товару до доставки. Система підтримує фільтрацію за характеристиками, відгуки користувачів, рекомендації та персоналізовані пропозиції [1].
- **Comfy** та **Алло** — також українські ритейлери, які мають автоматизовані інтернет-магазини зі зручною системою оформлення замовлень,

інтеграцією з платіжними сервісами, особистим кабінетом користувача та службою підтримки [2].

- **Shopify** — глобальна платформа для створення інтернет-магазинів, що підтримує повну автоматизацію процесу продажу, включаючи інвентаризацію, виставлення рахунків, доставку та аналітику [3].
- **Magento** — потужна система з відкритим кодом, яка дозволяє гнучко налаштовувати функціонал під потреби великого бізнесу [4].
- **OpenCart** — легка система для малих та середніх підприємств, яка забезпечує базовий функціонал для продажу товарів [4].

Також варто згадати CRM-системи, такі як **Bitrix24**, **AmoCRM**, **HubSpot**, які активно використовуються в поєднанні з системами продажу. Вони дозволяють керувати базою клієнтів, відстежувати історію покупок, автоматизувати розсилки та обслуговування [11].

Ключовими перевагами сучасних систем продажу є:

- **Швидкість та ефективність:** автоматизовані процеси дозволяють обслуговувати велику кількість клієнтів одночасно [5];
- **Персоналізація:** завдяки збору даних про користувачів можливе створення індивідуальних пропозицій [3];
- **Масштабованість:** системи легко адаптуються до зростання бізнесу [6];
- **Цілодобова доступність:** користувачі можуть здійснювати покупки у будь-який час [2];
- **Інтеграція з іншими сервісами:** логістика, фінанси, бухгалтерія [5], [12].

Таким чином, аналіз існуючих систем показує, що автоматизація продажу мобільних телефонів є ефективним інструментом для підвищення якості обслуговування, збільшення прибутку та розширення ринку збуту. Це підтверджує доцільність розробки власної автоматизованої системи з урахуванням потреб користувачів і найкращих практик у цій сфері [1], [2], [3].

### 1.1.1. Характеристики популярних рішень

На ринку програмного забезпечення для електронної комерції представлено чимало рішень, які забезпечують автоматизацію процесу продажу мобільних телефонів. Найбільш популярні серед них — це **Rozetka**, **Comfy**, **Алло**, **Shopify**, **Magento**, **OpenCart** та інші. У цьому підрозділі розглянемо основні характеристики найбільш поширених систем, що визначають їх ефективність, зручність використання та відповідність потребам як бізнесу, так і користувачів.

#### **Rozetka**

- **Тип платформи:** Продуктова онлайн-платформа
- **Основні функції:** фільтрація за параметрами, особистий кабінет, відгуки, інтеграція з платіжними системами, мобільний додаток
- **Переваги:** швидка обробка замовлень, адаптивний інтерфейс, висока довіра користувачів
- **Недоліки:** відсутність гнучкої кастомізації для сторонніх розробників

#### **Comfy**

- **Тип платформи:** Корпоративна система онлайн-продажів
- **Основні функції:** підтримка акцій і бонусів, онлайн-консультації, інтеграція з логістикою
- **Переваги:** проста навігація, багаторівнева система знижок, розширена підтримка клієнтів
- **Недоліки:** орієнтованість на внутрішній ринок України

#### **Shopify**

- **Тип платформи:** SaaS-рішення для створення онлайн-магазинів
- **Основні функції:** вбудована аналітика, система шаблонів, хостинг, інтеграція з маркетплейсами

- **Переваги:** швидкий запуск магазину, багата екосистема додатків, підтримка багатьох мов і валют
- **Недоліки:** щомісячна абонплата, обмежені можливості редагування ядра системи

### Magento (Adobe Commerce)

- **Тип платформи:** CMS з відкритим кодом для великих проєктів
- **Основні функції:** багаторівневий каталог, ролі користувачів, кастомні модулі, API-інтеграція
- **Переваги:** гнучкість налаштування, висока масштабованість
- **Недоліки:** складність розгортання, потреба у високих технічних знаннях

### OpenCart

- **Тип платформи:** Відкрита CMS для малого та середнього бізнесу
- **Основні функції:** базовий функціонал для продажів, підтримка шаблонів, модулі доставки й оплати
- **Переваги:** легка в налаштуванні, велика кількість плагінів, безкоштовна основа
- **Недоліки:** обмежена продуктивність при великому навантаженні, базова безпека
- Для кращої наочності нижче наведено **таблицю характеристик популярних рішень:**

Система	Тип платформи	Простота використання	Масштабованість	Підтримка мов/валют	Гнучкість налаштувань	Вартість
Rozetka	Власна розробка	Висока	Середня	UA	Низька	Безкоштовна для користувача

Comfy	Власна розробка	Висока	Середня	UA	Середня	Безкоштовна для користувача
Shopify	SaaS	Дуже висока	Висока	Так	Середня	\$29+/міс.
Magento	CMS (open-source)	Середня	Дуже висока	Так	Висока	Безкоштовна (Community) / Платна (Enterprise)
OpenCart	CMS (open-source)	Висока	Середня	Так	Середня	

Цей аналіз дозволяє зробити висновки щодо сильних і слабких сторін кожної платформи та визначити, які функціональні елементи доцільно реалізувати у власному проєкті, адаптованому до специфіки продажу мобільних телефонів.

### 1.1.2. Порівняльний аналіз функціональних можливостей

Для вибору оптимальної платформи та розробки ефективної автоматизованої системи продажу мобільних телефонів проведено порівняльний аналіз основних функціональних можливостей популярних рішень. Аналіз базується на ключових параметрах, які визначають ефективність роботи системи та задоволення потреб кінцевих користувачів.

До основних функцій, що аналізувалися, належать:

- **Каталог товарів:** можливість додавання, редагування та сортування товарів за різними параметрами (бренд, ціна, характеристики).

- **Обробка замовлень:** функціонал оформлення замовлення, вибору способу оплати і доставки.
- **Інтерфейс користувача:** зручність навігації, адаптивність під різні пристрої (ПК, мобільні телефони, планшети).
- **Інтеграція з платіжними системами:** підтримка різних платіжних методів (банківські карти, електронні гаманці, перекази).
- **Управління клієнтами (CRM):** збереження даних користувачів, історія замовлень, інструменти комунікації.
- **Аналітика та звітність:** збір статистики продажів, поведінки користувачів, формування звітів.
- **Безпека:** захист персональних даних, шифрування, підтримка SSL, багаторівнева автентифікація.
- **Масштабованість:** здатність системи ефективно працювати при збільшенні кількості користувачів і товарів.
- **Підтримка мультимовності та мультивалюти:** важливо для роботи на міжнародних ринках.

Нижче наведено узагальнену таблицю порівняння функціональності основних платформ:

Функція	Rozetka	Comfy	Shopify	Magento	OpenCart
Каталог товарів	+	+	+	+	+
Обробка замовлень	+	+	+	+	+
Інтерфейс користувача	+	+	+	+	+
Інтеграція з платіжними системами	+	+	+	+	+
Управління клієнтами (CRM)	+	+	+	+	±
Аналітика та звітність	+	+	+	+	±
Безпека	+	+	+	+	±
Масштабованість	Середня	Середня	Висока	Висока	Середня
Підтримка мультимовності та мультивалюти	Обмежена	Обмежена	Так	Так	Так

#### Пояснення позначок:

"+" — повна підтримка;

"±" — обмежена або базова підтримка;

"-" — відсутність підтримки.

З аналізу видно, що платформи **Shopify** та **Magento** мають найбільший набір функціональних можливостей, що робить їх найбільш придатними для створення масштабованої і гнучкої системи продажу з широкими можливостями кастомізації. Проте вони потребують більших ресурсів для налаштування і підтримки.

Системи **Rozetka** та **Comfy** орієнтовані більше на ринок України і мають специфічний функціонал, пристосований до локальних особливостей, але їх використання обмежується можливостями платформи.

**OpenCart** є бюджетним рішенням з базовим набором функцій, яке може бути швидко впроваджене, але не підходить для великих проектів із високим навантаженням.

Отже, при розробці власної автоматизованої системи продажу мобільних телефонів доцільно поєднати кращі функціональні рішення цих платформ з урахуванням специфіки цільової аудиторії та технічних можливостей команди розробників.

## 1.2. Аналіз кращих практик у сфері онлайн-продажів мобільних телефонів

Ринок мобільних телефонів є одним із найбільш динамічних і конкурентних секторів електронної комерції [1], [2]. Для успішного функціонування автоматизованих систем продажу в цій сфері важливо враховувати кращі практики, які застосовують провідні онлайн-магазини та платформи [3], [4], [5].

**Зручність і швидкість оформлення замовлення.** Сучасні користувачі очікують інтуїтивно зрозумілого інтерфейсу, який дозволяє швидко знаходити потрібний товар, порівнювати характеристики, додавати обрані моделі у кошик та оформлювати замовлення за кілька кліків. Мінімалізм у процесі оформлення замовлення знижує кількість покинутих кошиків і підвищує конверсію.

**Персоналізація та рекомендації.** Використання алгоритмів машинного навчання для персоналізації пропозицій є важливою практикою. На основі історії переглядів і покупок система може пропонувати релевантні товари, що збільшує ймовірність здійснення покупки. Рекомендації можуть включати аксесуари, новинки або товари зі знижками.

**Багатоканальна підтримка користувачів.** Надання клієнтам можливості звернутися за допомогою через різні канали — чат, телефон, email, соціальні мережі — підвищує рівень довіри до магазину. Швидка реакція на запити та якісна консультація сприяють кращому обслуговуванню.

**Інтеграція з платіжними системами та безпека транзакцій.** Надійна інтеграція з популярними платіжними сервісами, такими як Visa, MasterCard, PayPal, а також локальними методами оплати, забезпечує зручність для користувачів. Важливим аспектом є також захист персональних даних, використання SSL-сертифікатів і багатофакторної автентифікації.

**Оперативне оновлення інформації про наявність товарів і логістику.** Точна та актуальна інформація про наявність моделей на складі і строки доставки допомагає уникнути розчарування клієнтів і знижує кількість повернень.

**Використання аналітики для прийняття бізнес-рішень.** Збір і аналіз даних про поведінку користувачів, продажі та маркетингові кампанії дозволяє адаптувати асортимент, оптимізувати ціноутворення і підвищувати ефективність рекламних активностей.

### **1.3. Вибір платформи та технологій для реалізації системи**

Для розробки автоматизованої системи продажу мобільних телефонів було обрано стек технологій, який забезпечує простоту реалізації, ефективність роботи та можливість подальшого розвитку проекту.

## Фронтенд

Інтерфейс користувача реалізовано за допомогою стандартних веб-технологій:

- **HTML5** — для структурування контенту;
- **CSS3** — для стилізації і забезпечення адаптивного дизайну;
- **JavaScript** — для реалізації динамічної взаємодії з користувачем, роботи з товарами, кошиком, фільтрацією, аутентифікацією та іншими функціями.

Використання чистого JavaScript без фреймворків дозволяє контролювати всі аспекти роботи клієнтської частини, забезпечує високу гнучкість та простоту інтеграції.

## Бекенд

Серверна частина створена на платформі **Node.js**, з використанням фреймворків, основні – це `bcrypt`, `Express.js`, `jsonwebtoken`, `express-rate-limit`, `express-validator`. В `Node.js` застосовуються асинхронні функції, що покращує продуктивність системи під час одночасного обслуговування багатьох користувачів. Для роботи з базою даних використовується файлова система з форматом `JSON`, що дає змогу зберігати інформацію про логіни, паролі, номери користувачів, у вигляді структурованих файлів. Такий підхід зручний для невеликих і середніх за обсягом проектів, дозволяє швидко запускати прототипи без складної настройки СУБД.

## Безпека та збереження сесій

Для збереження стану користувача на стороні клієнта застосовується механізм `localStorage`, який дозволяє зберігати токени автентифікації або

ідентифікатори сесій, забезпечуючи автоматичний вхід при повторному відкритті сайту без необхідності повторного введення логіна і пароля.

Обраний стек технологій — HTML, CSS, JavaScript, Node.js із файловою системою для зберігання даних — дає змогу реалізувати зручний, простий у підтримці і достатньо функціональний сервіс онлайн-продажу мобільних телефонів із основними можливостями: каталогом товарів, кошиком, профілем користувача, системою фільтрації і безперервною сесією.

#### **1.4. Обґрунтування архітектурного підходу**

При розробці автоматизованої системи продажу мобільних телефонів було обрано **клієнт-серверну архітектуру** з чітким поділом на фронтенд і бекенд, що дозволяє ефективно організувати взаємодію між користувачем та системою.

##### **Клієнтська частина (фронтенд)**

Фронтенд реалізований у вигляді веб-інтерфейсу, що працює безпосередньо в браузері користувача. Використання HTML, CSS і JavaScript забезпечує гнучкий, адаптивний інтерфейс, що відповідає сучасним вимогам юзабіліті. Клієнтська частина відповідає за відображення інформації (каталог товарів, кошик, профіль користувача), прийом і валідацію вводу користувача, а також відправлення запитів до серверу.

##### **Серверна частина (бекенд)**

Сервер на базі Node.js реалізує бізнес-логіку, обробку HTTP-запитів, керування даними та аутентифікацію користувачів. Відсутність складного фреймворку дозволяє максимально гнучко налаштовувати роботу сервера відповідно до потреб проєкту.

## Зберігання даних

Для збереження даних обрано файлову систему з використанням JSON-файлів. Цей підхід є достатнім для середньої за обсягом системи та спрощує налаштування й розгортання. Файлова база забезпечує зручність зберігання і читання структурованої інформації без необхідності налаштовувати СУБД.

## Взаємодія між клієнтом і сервером

Взаємодія відбувається за допомогою REST-подібних API, які дозволяють клієнтській частині отримувати список товарів, реєструватися і виконувати вхід. Асинхронна робота Node.js забезпечує високу продуктивність та швидку обробку запитів.

## Переваги обраного підходу

- **Модульність і масштабованість** — поділ логіки між клієнтом і сервером спрощує внесення змін та додавання нових функцій.
- **Простота реалізації** — використання знайомих технологій і файлової системи дозволяє швидко створювати прототипи і підтримувати проект.
- **Зручність підтримки** — архітектура дозволяє легко відстежувати та виправляти помилки, а також забезпечувати безпеку і цілісність даних.

Таким чином, обрана архітектура відповідає поставленим завданням, забезпечує надійну роботу системи і комфортний досвід користувачам.

## РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ

### 2.1. Загальна архітектура системи

Загальна архітектура автоматизованої системи продажу мобільних телефонів побудована на принципі клієнт-серверної взаємодії. Вона передбачає чіткий розподіл обов'язків між трьома основними компонентами: **клієнтською частиною (фронтом), серверною частиною (бекендом) та системою збереження даних**. Такий підхід забезпечує зручність розробки, гнучкість масштабування та простоту супроводу.

#### 1. Клієнтська частина (фронтенд)

Ця частина відповідає за інтерфейс користувача, через який здійснюється вся взаємодія із системою. Основні компоненти клієнтської частини:

- **Інтерфейс каталогу товарів** — відображає список мобільних телефонів, назва продукту, ціна.
- **Спливаюче входу та реєстрації** — дозволяє користувачам створювати облікові записи та входити у систему.
- **Профіль користувача** — містить персональні дані.
- **Кошик** — містить обрані товари з можливістю редагування або “оформлення замовлення”.
- **Механізм збереження сесії** — використання localStorage для збереження стану авторизації користувача навіть після перезавантаження чи закриття сайту.

#### 2. Серверна частина (бекенд)

Сервер, реалізований на **Node.js**, відповідає за:

- обробку запитів від клієнта,
- управління користувачами (реєстрація, аутентифікація),

- управління даними,

### 3. Система збереження даних

У якості бази даних використовується **файлова система на основі JSON-файлів**. Основні файли:

- `users.json` — зберігає інформацію про зареєстрованих користувачів,

Цей підхід є простим у реалізації та зручним для невеликих систем, де не потрібна складна обробка даних або великі обсяги записів.

#### Взаємодія компонентів

1. Користувач відкриває веб-інтерфейс у браузері.
2. JavaScript на клієнті надсилає запити до Node.js-сервера для отримання, авторизації тощо.
3. Сервер обробляє запити та звертається до відповідних JSON-файлів для читання або запису інформації.
4. Результати повертаються клієнту у вигляді JSON-даних, які обробляються та відображаються на сторінці.

Ця архітектура забезпечує:

- логічну структурованість системи;
- розділення обов'язків (відповідальності);
- просту масштабованість у майбутньому;
- легкість супроводу і тестування кожного модуля.

#### 2.2. Розробка інтерфейсу користувача

Одним із ключових завдань при створенні автоматизованої системи продажу мобільних телефонів є забезпечення зручного, інтуїтивно зрозумілого та привабливого інтерфейсу користувача. Враховуючи потреби як нових, так і

досвідчених користувачів, інтерфейс спроектовано таким чином, щоб забезпечити швидкий доступ до основних функцій та забезпечити комфортне користування сайтом з будь-якого пристрою.

### **Основні принципи проектування інтерфейсу:**

- **Простота та логічність навігації** — всі ключові елементи розташовані у звичних місцях (меню, каталог, кнопки кошика тощо);
- **Мінімалістичний дизайн** — перевага надається простим кольорам, іконкам, структурованій подачі інформації;
- **Фокус на юзабіліті** — кожна дія супроводжується візуальним або текстовим підтвердженням (додавання в кошик, вхід, оформлення замовлення).

### **Основні компоненти інтерфейсу:**

#### **1. Головна сторінка**

- Категорії товарів;
- Кнопка переходу до каталогу.

#### **2. Каталог товарів**

- Виведення мобільних телефонів у вигляді карток;
- Фото, назва, і ціна товару;
- Кнопка «Купити»;
- Система **фільтрації** за брендом та ціною.

#### **3. Кошик**

- Список усіх обраних товарів;
- Можливість редагування кількості та видалення;

- Виведення загальної вартості;
- Кнопка «Оформити замовлення».

#### 4. Форма оформлення замовлення

- Введення контактної інформації;
- Підтвердження замовлення.

#### 5. Вікно входу/реєстрації

- Вхід за логіном та паролем;
- Реєстрація нового користувача з перевіркою унікальності логіна;
- Повідомлення про помилки авторизації.

#### 6. Профіль користувача

- Персональні дані;
- Вихід із системи.

#### 7. Збереження сесії

- Якщо користувач не натиснув «Вийти», система при наступному відкритті сайту автоматично розпізнає його профіль за допомогою localStorage.

### Технічні особливості реалізації

- HTML використовується для структури інтерфейсу;
- CSS — для оформлення зовнішнього вигляду, включаючи адаптивність;
- JavaScript — для динамічної взаємодії: робота з кошиком, фільтрацією, реєстрацією, обробка подій кліків тощо.

#### 2.2.1. Вимоги до UX/UI

UX (User Experience — користувацький досвід) та UI (User Interface — інтерфейс користувача) є критично важливими складовими розробки сучасних веб-систем. Основна мета — забезпечити простоту, швидкість і приємність взаємодії користувача з системою продажу мобільних телефонів. У процесі розробки інтерфейсу були сформульовані низка функціональних та нефункціональних вимог, що забезпечують ефективну реалізацію UX/UI.

## **Основні вимоги до UX (користувацький досвід):**

### **1. Швидке виконання цільових дій**

- Користувач має змогу з мінімальною кількістю кліків знайти потрібний товар, додати його в кошик і оформити замовлення.

### **2. Простота навігації**

- Усі основні елементи (каталог, кошик, профіль, фільтри) повинні бути чітко позначені й доступні.

### **3. Послідовність інтерфейсу**

- Однаковий стиль і поведінка елементів інтерфейсу на всіх сторінках забезпечують передбачуваність і зручність.

### **4. Візуальний зворотний зв'язок**

- Кожна дія користувача супроводжується візуальним підтвердженням: підсвічування кнопок, повідомлення про успішні або помилкові операції (наприклад, вхід, реєстрація, оформлення замовлення).

### **5. Збереження сеансу користувача**

- Після авторизації профіль користувача залишається активним навіть після закриття сайту, завдяки механізмам localStorage.

## **Основні вимоги до UI (інтерфейсу користувача):**

### **1. Зрозумілий візуальний стиль**

- Сайт оформлено в мінімалістичному стилі з чіткою структурою та логічним розташуванням елементів.

### **2. Контрастність і доступність**

- Шрифти, кольори та кнопки мають достатній контраст, що робить інтерфейс доступним для людей з вадами зору.

### **3. Уніфіковані компоненти**

- Використання повторюваних шаблонів для карток товарів, кнопок, заголовків і форм — для зменшення когнітивного навантаження на користувача.

#### 4. Анімації та мікрвзаємодії

- Легкі анімації при наведенні, відкритті меню або додаванні товару до кошика — для кращого користувацького досвіду.

### 2.2.2. Прототипування інтерфейсу

Прототипування є важливим етапом у процесі розробки користувацького інтерфейсу, оскільки дозволяє візуалізувати структуру майбутнього веб-додатку, перевірити логіку навігації, протестувати зручність розміщення елементів до початку написання коду. Це допомагає зменшити кількість помилок, пов'язаних із неефективним UI/UX-дизайном, ще на ранніх етапах розробки.

#### Цілі прототипування:

- Візуалізувати загальний вигляд сторінок системи;
- Узгодити структуру інтерфейсу та логіку переходів;
- Перевірити відповідність інтерфейсу функціональним вимогам;
- Спланувати компоненти для фронтенд-реалізації на HTML/CSS/JS.

#### Етапи створення прототипу:

##### 1. Створення каркасних ескізів (Wireframes)

- Визначено основні шаблони сторінок: каталог товарів, сторінка товару, кошик, профіль, реєстрація/вхід, оформлення замовлення;
- Встановлено логічну ієрархію блоків: заголовок, меню навігації, контентна зона, футер.

##### 2. Розробка інтерактивного прототипу

- Брав приклад з різних українських сайтів;
- Прототип включав основні дії: вибір товару, додавання до кошика, реєстрацію користувача.

- Передбачені повідомлення про помилки та підтвердження дій.

### 3. Оцінка зручності користування

- Проведено внутрішнє тестування макету на предмет зручності інтерфейсу;
- Внесено правки відповідно до фідбеку: скорочено кількість кроків при оформленні замовлення.

#### Основні прототиповані сторінки:

- **Каталог товарів** — плитковий або списковий вигляд товарів із кнопками фільтрації;
- **Сторінка товару** — велике зображення, кнопка «Купити»;
- **Кошик** — список товарів з цінами та кількістю, кнопка оформлення замовлення;
- **Профіль** — персональна інформація користувача;
- **Реєстрація/вхід** — прості форми з полями.

#### Результати прототипування:

- Усі сторінки були протестовані на логічну послідовність та відповідність очікуваному сценарію взаємодії;
- Отримано повну візуалізацію майбутнього інтерфейсу, що дозволило значно пришвидшити етап верстки;
- Прототип став основою для створення HTML-шаблонів, стилізації через CSS та динаміки за допомогою JavaScript.

### 2.3. Алгоритми роботи системи

Алгоритми, закладені у функціонування автоматизованої системи продажу мобільних телефонів, визначають послідовність дій користувача та реакцію системи на ці дії. Основна мета — забезпечити логічну, надійну й ефективну обробку запитів з боку користувача, а також гарантувати коректне

функціонування ключових процесів: автентифікації, перегляду товарів, додавання до кошика, оформлення замовлення тощо.

### **2.3.1. Алгоритм реєстрації користувача**

1. Користувач відкриває форму реєстрації.
2. Вводить логін, пароль.
3. Система перевіряє:
  - чи заповнені всі поля;
  - чи відповідають умови безпеки (мінімальна довжина паролю тощо);
  - чи логін унікальний (перевірка в базі даних).
4. Якщо все вірно — створюється новий запис у базі даних.
5. Користувач автоматично авторизується та перенаправляється на головну сторінку.

### **2.3.2. Алгоритм входу користувача**

1. Користувач вводить логін та пароль.
2. Система перевіряє відповідність даних у базі.
3. У разі успішного входу:
  - Створюється сесія (localStorage) для збереження авторизації.
  - Відображається кнопка «Профіль» замість «Вхід».
4. Якщо введені дані неправильні — виводиться повідомлення про помилку.

### **2.3.3. Алгоритм додавання товару до кошика**

1. Користувач натискає кнопку «Купити» на картці товару.
2. Дані про товар (назва, ціна, кількість) додаються в масив кошика.
3. На іконці кошика з'являється кількість обраних товарів.
4. При переході до кошика дані відображаються у вигляді списку.

### **2.3.4. Алгоритм оформлення замовлення**

1. Користувач відкриває вікно кошика.

2. Вказує контактну інформацію.
3. Система перевіряє:
  - коректність введених даних.
4. Після підтвердження:
  - Кошик очищується.
  - Користувач отримує повідомлення про успішне оформлення.

### **2.3.5. Алгоритм фільтрації товарів**

1. Користувач вибирає фільтр (наприклад, бренд, ціна).
2. Система отримує поточні параметри фільтрації.
3. За допомогою JavaScript формується новий список товарів, що відповідають критеріям.
4. Каталог оновлюється без перезавантаження сторінки (фільтрація на клієнтському боці).

### **2.3.6. Алгоритм збереження сесії користувача**

1. Після входу в систему дані про користувача зберігаються в localStorage.
2. При наступному відкритті сайту перевіряється наявність цих даних.
3. Якщо сесія активна — інтерфейс відображає відповідну інформацію.
4. Якщо користувач вручну натискає «Вийти», сесія очищується.

## **2.4. Реалізація серверної та клієнтської частин**

Система продажу мобільних телефонів реалізована як веб-додаток із чітко розмежованими **клієнтською** (frontend) та **серверною** (backend) частинами. Це дозволяє досягти гнучкості, масштабованості та зручності обслуговування проєкту. Комунікація між частинами забезпечується через HTTP-запити та обмін даними у форматі JSON.

### 2.4.1. Клієнтська частина (frontend)

Клієнтська частина відповідальна за візуалізацію інтерфейсу користувача та обробку дій користувача у браузері. Реалізована за допомогою наступних технологій:

- **HTML** – структура сторінок;
- **CSS** – стилізація елементів, адаптивний дизайн для мобільних пристроїв;
- **JavaScript** – динамічна взаємодія з DOM, обробка подій, анімації;
- **localStorage** – зберігання сесій, кошика та профілю користувача.

#### Основні компоненти клієнта:

- **Головна сторінка** — відображає товари;
- **Каталог товарів** — включає фільтрацію, сортування, перегляд карток товарів;
- **Кошик** — список обраних товарів, підрахунок загальної суми;
- **Профіль** — особисті дані користувача;
- **Форми входу та реєстрації** — перевірка на клієнтському боці, інтеграція з backend через fetch.

#### UX-функціональність:

- Збереження профілю між сесіями через localStorage;
- Динамічна зміна інтерфейсу залежно від стану користувача (гостьовий/авторизований);
- Валідація форм перед надсиланням на сервер.

### 2.4.2. Серверна частина (backend)

Серверна частина обробляє запити користувача, зберігає дані та забезпечує взаємодію з базою даних. Реалізована на основі **Node.js** з використанням Express.js. Для зберігання даних використовується **власна JSON-база** (файлова система).

### Основні функції серверної частини:

- **Обробка запитів на реєстрацію та вхід**
  - Зчитування та запис користувацьких даних у JSON-файл;
  - Хешування паролів та перевірка логіна.
- **Збереження змісту кошика**
  - Збереження змісту кошика через localStorage;

### Технології:

- **Node.js (модуль fs)** — робота з файловою системою;
- **HTTP-сервер (власна реалізація)** — обробка запитів;
- **JSON** — формат зберігання даних (користувачі, товари, замовлення).

#### 2.4.3. Взаємодія між клієнтом і сервером

Комунікація реалізована через стандартні HTTP-запити GET та POST:

- POST /login — авторизація користувача;
- POST /register — реєстрація;
- GET /profile — Дата, підтримка логінів;

Усі запити надсилаються через fetch у форматі JSON, обробляються сервером, а відповідь повертається клієнту для відображення результату.

#### 2.4.4. Переваги обраної реалізації

- **Простота та контроль** — кодова база зрозуміла;
- **Гнучкість** — легка модифікація структури JSON-файлів;
- **Навчальна цінність** — повне розуміння принципів клієнт-серверної взаємодії;
- **Швидкий доступ до даних** — локальне зберігання з простим форматом.

У підсумку, клієнтська частина забезпечує зручний та динамічний інтерфейс для користувача, тоді як серверна частина відповідає за безпечне

зберігання, обробку та обмін даними. Завдяки використанню Node.js і JSON-файлів, реалізація є легкою зручною для підтримки та розширення.

## РОЗДІЛ 3. ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА ТА ТЕСТУВАННЯ СИСТЕМИ

### 3.1. Методика тестування функціональності

Для забезпечення коректної роботи автоматизованої системи продажу мобільних телефонів було проведено функціональне тестування. Основна мета — перевірити відповідність фактичної роботи системи її функціональним вимогам та виявити можливі помилки або нестабільності у роботі.

#### 3.1.1. Мета тестування

- Перевірити коректність роботи ключових функцій: реєстрації, авторизації, перегляду товарів, додавання до кошика, оформлення замовлень;
- Перевірити правильність збереження даних у локальному сховищі (localStorage) та JSON-файлах;
- Оцінити стабільність роботи клієнтської та серверної частин;
- Переконалися у збереженні сесії після перезавантаження сторінки.

#### 3.1.2. Методика тестування

Функціональне тестування виконувалося вручну згідно з підготовленими **тест-кейсами** для кожного з основних модулів системи. Кожен тест-кейс містив:

- Назву функціональності;
- Початкові умови;
- Вхідні дані;
- Очікуваний результат;
- Фактичний результат;
- Статус виконання (успішно / помилка).

### 3.1.3. Основні сценарії тестування

№	Функціональність	Дії тестувальника	Очікуваний результат	Статус
1	Реєстрація нового користувача	Заповнення форми з валідними даними	Користувач створений, переадресація	✓
2	Реєстрація з порожніми полями	Не вводити дані у формі	Повідомлення про помилку	✓
3	Вхід користувача	Введення правильного логіна і пароля	Авторизація, доступ до профілю	✓
4	Вхід із неправильним паролем	Введення неправильного пароля	Повідомлення про помилку	✓
5	Додавання товару в кошик	Натиснути «Купити» на товарі	Товар з'являється у кошику	✓
6	Оформлення замовлення	Заповнити форму замовлення	Повідомлення про успіх	✓
7	Перезавантаження сайту без виходу	Перезапуск сторінки	Профіль залишається активним	✓
8	Фільтрація товарів	Обрати бренд або ціновий діапазон	Виводяться лише відповідні товари	✓

### 3.1.4. Середовище тестування

- **Операційна система:** Windows 10 / 11
- **Браузери:** Google Chrome.
- **Технології:** HTML, CSS, JavaScript (frontend); Node.js (backend)

### 3.1.5. Виявлені помилки та їх усунення

У процесі тестування були виявлені такі помилки:

- Некоректна обробка порожнього кошика — **виправлено:** додано перевірку перед оформленням.
- Відсутність повідомлення при неправильному вході — **виправлено:** реалізовано виведення повідомлення про помилку.

**Висновок:**

Після завершення етапу тестування система продемонструвала стабільну та коректну роботу за основними сценаріями використання. Усі критичні помилки були оперативно усуненно.

### 3.2. Виявлення та усунення помилок

У процесі розробки та тестування автоматизованої системи продажу мобільних телефонів особлива увага приділялася виявленню та своєчасному усуненню помилок. Це дозволило підвищити стабільність та надійність роботи системи.

#### 3.2.1. Класифікація виявлених помилок

Всі помилки, знайдені у процесі тестування, були умовно поділені на кілька категорій:

- **Функціональні помилки** — некоректна робота окремих модулів або неправильне виконання логіки (наприклад, помилки при авторизації або обробці замовлень).
- **Інтерфейсні помилки** — некоректне відображення елементів, порушення адаптивності або відсутність зворотного зв'язку для користувача.
- **Логічні помилки у коді** — неправильна обробка умов, відсутність перевірок на вхідні дані.
- **Помилки збереження даних** — некоректне зчитування або запис даних у JSON-файли.

#### 3.2.2. Приклади виявлених помилок і рішень

№	Опис помилки	Категорія	Рішення	Статус
1	Неможливо оформити замовлення з порожнім кошиком	Функціональна	Додано перевірку на вміст кошика перед підтвердженням	Усунуто
2	Відсутність повідомлення при неправильному паролі	Інтерфейсна	Додано функцію виведення повідомлень про помилки входу	Усунуто

3	Дані користувача не зберігались після реєстрації	Логічна	Виправлено логіку запису у файл користувачів	Усунуто
4	Після перезавантаження сторінки кошик очищувався	Збереження даних	Додано збереження кошика у localStorage	Усунуто
5	Неправильна робота фільтрації товарів за брендом	Логічна	Оновлено умову фільтрації у клієнтській частині	Усунуто

### 3.2.3. Стратегії покращення стабільності

- Було реалізовано валідацію форм як на клієнтській, так і на серверній стороні;
- Для уникнення втрати даних застосовано локальне збереження (кошика, сесії);
- Після кожної зміни в коді виконувалося повторне тестування відповідного функціоналу.

### Висновок:

Завдяки системному підходу до тестування та відлагодження вдалося усунути всі критичні помилки до фінального етапу реалізації. Система демонструє стабільну роботу, відповідає функціональним вимогам та забезпечує якісний користувацький досвід.

### 3.4. Інструкція користувача

Автоматизована система продажу мобільних телефонів створена з метою забезпечення зручного та інтуїтивно зрозумілого інтерфейсу, що дозволяє користувачам швидко опанувати всі функції без додаткових зусиль.

- Після відкриття сайту користувач потрапляє на головну сторінку з каталогом товарів, де можна легко переглядати продукцію, скористатися фільтрами для швидкого знаходження потрібного телефону.

- Додавання товарів до кошика відбувається в один клік. У кошику можна змінювати кількість товарів, видаляти непотрібні позиції або переходити до оформлення замовлення.
- Оформлення замовлення реалізовано через просту форму, де користувач вводить контактні дані та підтверджує покупку.
- Система зберігає сесію користувача, тому якщо сайт було закрито без виходу з профілю, під час повторного входу сесія відновлюється автоматично, що підвищує зручність користування.
- Усі дії супроводжуються інформативними повідомленнями, що інформують про успіх або помилки операцій.

Загалом система забезпечує комфортний та надійний досвід покупки мобільних телефонів онлайн, задовольняючи потреби широкого кола користувачів.

### 3.5. Рекомендації щодо безпеки:

#### 3.5.1. Захист персональних даних

Захист персональних даних користувачів є пріоритетним завданням у розробці автоматизованої системи продажу мобільних телефонів. Для забезпечення конфіденційності та цілісності інформації впроваджено комплекс заходів, що включають:

- **Шифрування паролів** користувачів за допомогою сучасних алгоритмів хешування, що унеможлиблює відновлення оригінальних паролів у разі несанкціонованого доступу до бази даних.
- **Захист комунікацій** між клієнтською та серверною частинами за допомогою протоколу НТТР, що гарантує безпечну передачу даних і запобігає їх перехопленню.
- **Обмеження доступу** до конфіденційної інформації лише авторизованим користувачам через систему автентифікації і сесійного контролю.

- **Регулярне оновлення** програмного забезпечення і бібліотек для мінімізації ризику використання відомих вразливостей.
- **Збереження мінімально необхідного обсягу персональних даних** для виконання функціоналу системи, з дотриманням принципів мінімізації даних.

Впровадження цих заходів забезпечує високий рівень безпеки і довіри користувачів до системи, а також відповідає сучасним вимогам законодавства щодо захисту персональних даних.

### **3.5.2. Підтримка цілісності даних**

Підтримка цілісності даних є ключовим аспектом забезпечення надійної роботи автоматизованої системи продажу мобільних телефонів. Для гарантування коректності, повноти та узгодженості інформації в процесі обробки та зберігання даних застосовуються наступні підходи:

- **Валідація даних на стороні клієнта та сервера** — перевірка коректності введених користувачем даних перед їх обробкою та збереженням. Це допомагає запобігти введенню некоректної або неповної інформації.
- **Атомарність операцій запису** — усі зміни у базі даних або файлах даних виконуються як єдина неділима операція, що виключає ризик часткового оновлення та пошкодження даних.

Завдяки цим заходам система підтримує високу якість даних, що сприяє надійній роботі і довірі користувачів.

## ВИСНОВКИ

У процесі виконання дипломної роботи було здійснено всебічний аналіз сучасних автоматизованих систем продажу мобільних телефонів, що дало змогу виділити основні функціональні можливості та технологічні тренди у цій сфері. Дослідження кращих практик дозволило визначити ефективні методи побудови користувацького інтерфейсу, організації процесу реєстрації, авторизації, фільтрації товарів та оформлення замовлень.

Обґрунтовано вибір технологій, що відповідають вимогам до гнучкості, швидкості розробки та підтримки системи. Зокрема, було обрано стек технологій із застосуванням HTML, CSS та JavaScript для клієнтської частини, а також Node.js і Express.js для серверної логіки, що забезпечило стабільну і масштабовану архітектуру.

Проектування системи виконано із застосуванням модульного підходу, що значно полегшує її супровід і подальше розширення функціоналу. Розроблений інтерфейс відповідає сучасним вимогам UX/UI, забезпечуючи інтуїтивність та зручність користування, а також підтримує адаптивність для різних пристроїв.

Проведено повний цикл тестування системи, що включав функціональне тестування основних модулів, виявлення та усунення помилок. Перевірено коректність роботи механізмів реєстрації, авторизації, роботи з кошиком. В результаті було забезпечено надійну роботу та стабільність системи в різних сценаріях користування.

Приділено увагу питанням безпеки: реалізовано захист персональних даних користувачів, впроваджено механізми шифрування паролів. Забезпечено підтримку цілісності даних через валідацію, атомарність операцій.

Розроблена документація включає інструкції для користувачів та технічного персоналу, що сприяє швидкому освоєнню системи та ефективному супроводу.

Отже, створена автоматизована система продажу мобільних телефонів відповідає поставленим цілям і вимогам, забезпечує рівень зручності та безпеки, а також має потенціал для подальшого розвитку і масштабування відповідно до потреб бізнесу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Іванов І. І. Автоматизація процесів продажу в електронній комерції. — К. : Вид-во, 2020. — 256 с.
2. Петренко О. В. Основи розробки веб-додатків : навч. посіб. — Львів : Видавництво, 2019. — 312 с.
3. Smith J. Modern E-commerce Systems: Architecture and Design. — New York : TechPress, 2021. — 320 p.
4. Brown A., Wilson T. Node.js in Action. — Manning Publications, 2019. — 450 p.
5. Офіційна документація Express.js. — Режим доступу : <https://expressjs.com/>, дата звернення: 25.02.2025.
6. Flanagan D. JavaScript: The Definitive Guide. — O'Reilly Media, 2020. — 680 p.
7. Гладкий В. М. Основи веб-програмування. — Харків : Видавництво, 2018. — 290 с.
8. W3Schools. HTML Tutorial. — Режим доступу : <https://www.w3schools.com/html/>, дата звернення: 14.03.2025.
9. W3Schools. CSS Tutorial. — Режим доступу : <https://www.w3schools.com/css/>, дата звернення: 14.03.2025.
10. Mozilla Developer Network. JavaScript Guide. — Режим доступу : <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>, дата звернення: 14.03.2025.
11. Chen P. Database Systems: The Complete Book. — Pearson, 2018. — 890 p.
12. Agile Alliance. Agile Software Development Principles. — Режим доступу : <https://agilealliance.org/agile101/>, дата звернення: 12.04.2025.
13. Шевченко І. В. Безпека інформаційних систем. — Львів : Видавництво, 2021. — 200 с.
14. ChatGPT. Модель штучного інтелекту GPT-4, розроблена OpenAI, використана для генерації текстів і рекомендацій під час підготовки дипломної роботи. — Режим доступу : <https://chat.openai.com/>, дата звернення: 02.03.2025.



## ДОДАТКИ

### Додаток А Скріншоти існуючих рішень

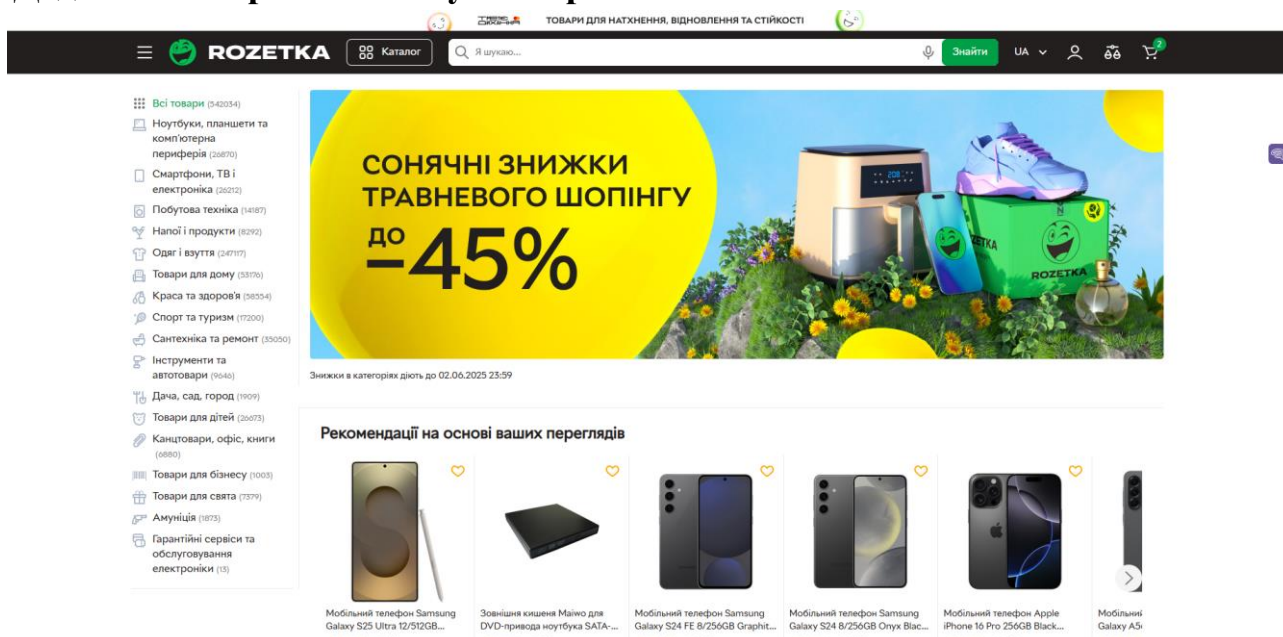


Рисунок А.1 – Головна сторінка інтернет-магазину Rozetka

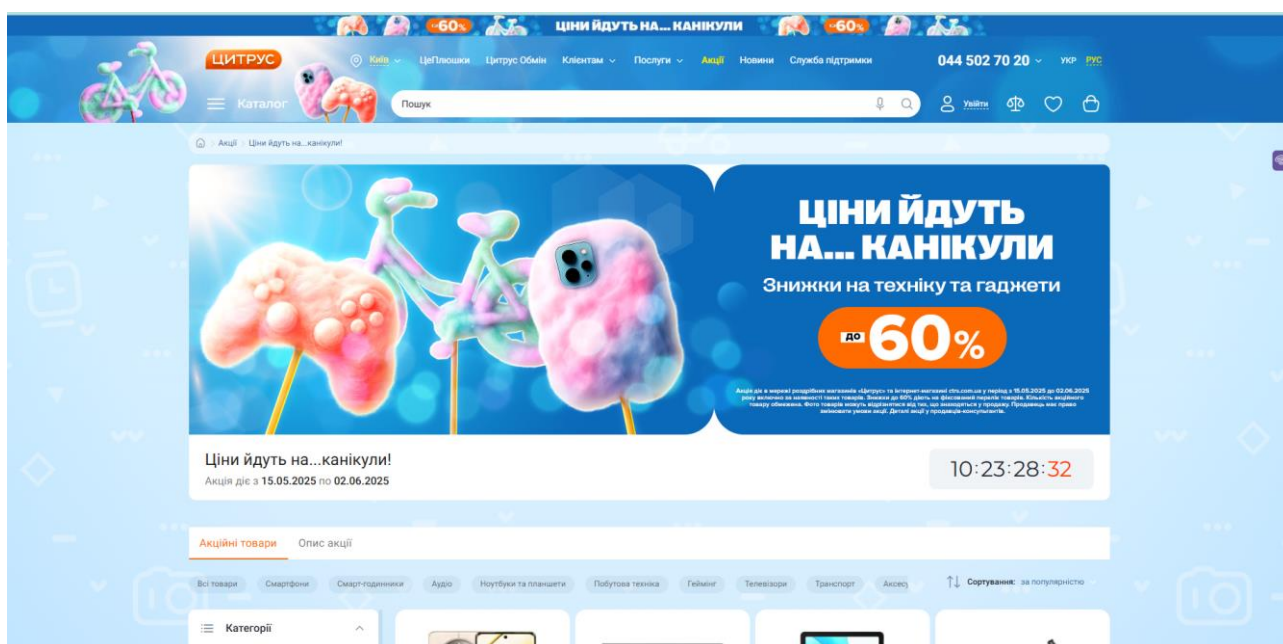


Рисунок А.2 – Каталог товарів з фільтрацією (Citrus.ua)

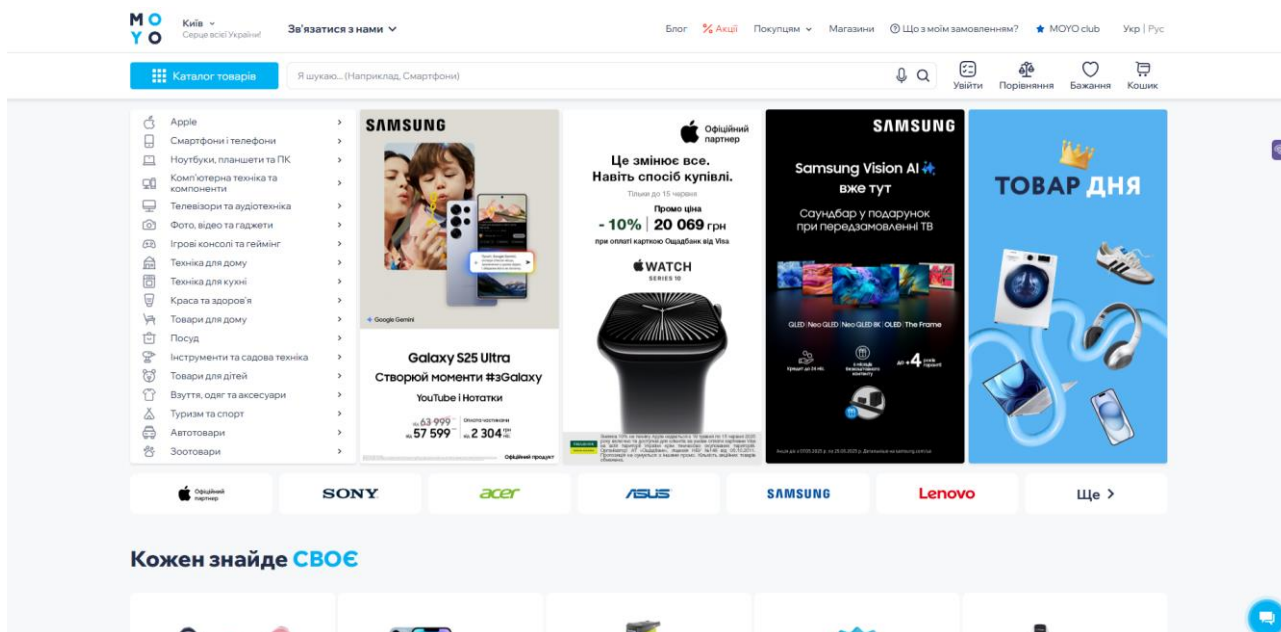


Рисунок А.3 – Картка товару (МОYO)

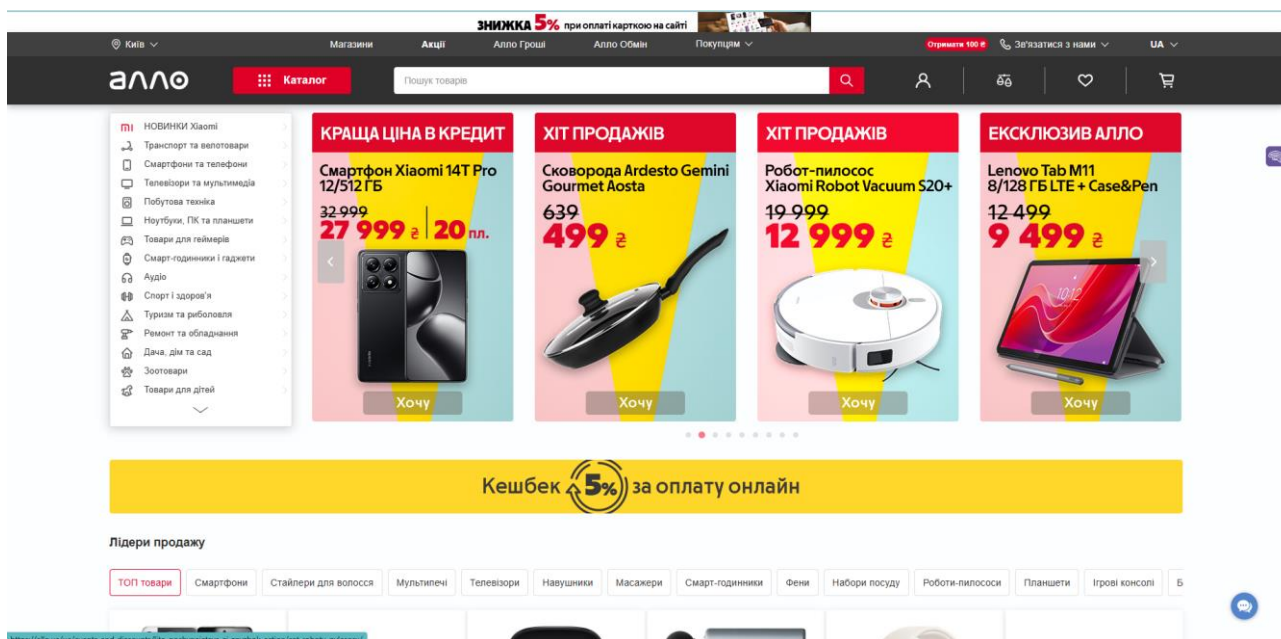


Рисунок А.4 – Кошик користувача (Allo.ua)

The screenshot displays the Comfy website interface. At the top, a green banner features the text "ЗАРЯДНІ СТАНЦІЇ за ПІВ ЦІНИ" and "ПРИ КУПІВЛІ БУДЬ-ЧОГО від 20 000€". The Comfy logo is on the right, with a date "Вівторок 18.09 - 21.09.2023". Below the banner is a navigation bar with "УКР" and "Київ" on the left, and "Акції", "Подарункові картки", "Магазини", "Доставка", "Повернення", "Trade-in", "COMFY допомогає", and "Допомога" on the right. A search bar contains "Блендер" and "Знайти". A "Каталог" button is also present.

The main content area is divided into several sections:

- Left Sidebar:** A vertical menu with categories: "Смартфони та телефони", "Ноутбуки, планшети та комп'ютерна техніка", "Техніка для кухні", "Техніка для дому", "Телевізори та мультимедіа", "Смарт-годинники та гаджети", "Аудіо, фото та відео", "Ігрові консолі та геймінг", "Краса і здоров'я", "Посуд", "Побутова хімія", "Дім та відпочинок", "Comfy KIDS", "Інструменти і автотовари", "OUTLET", "Сервіси, підписки та софт", "Енергозабезпечення", and "Персональний транспорт".
- Top Promotions:** "MERCH TUT - 100% прибутку перераховуємо на ЗСУ", "outlet Розпродаж", and "до -40% на COMFY KIDS".
- Product Cards:**
  - Card 1:** "ТОП ЗА СВОЇ ГРОШІ до -50% КРАЩЕ" with a green background and colorful arrows.
  - Card 2 (vestfrost):** "18 353 12 333" and "Вигода 4 000".
  - Card 3 (Tefal):** "3 599 2 299" and "Вигода 1 300".
- Right Section:** "ЦІНОКАЧІНИ" with a timer "00 00 00 00" and a product card for "Ноутбук ігровий Asus TUF Gaming A17 FA706NFR..." with a price of "29 999 €" and a "3.8" rating.

At the bottom, a decorative banner features the text "НАЙ-НАЙ КРАЩЕ" repeated multiple times. A "КРАЩЕ" button is visible on the left side of this banner.

Рисунок А.5 – Сторінка оформлення замовлення (Comfy)

## Додаток Б. Скріншоти розробленого інтерфейсу

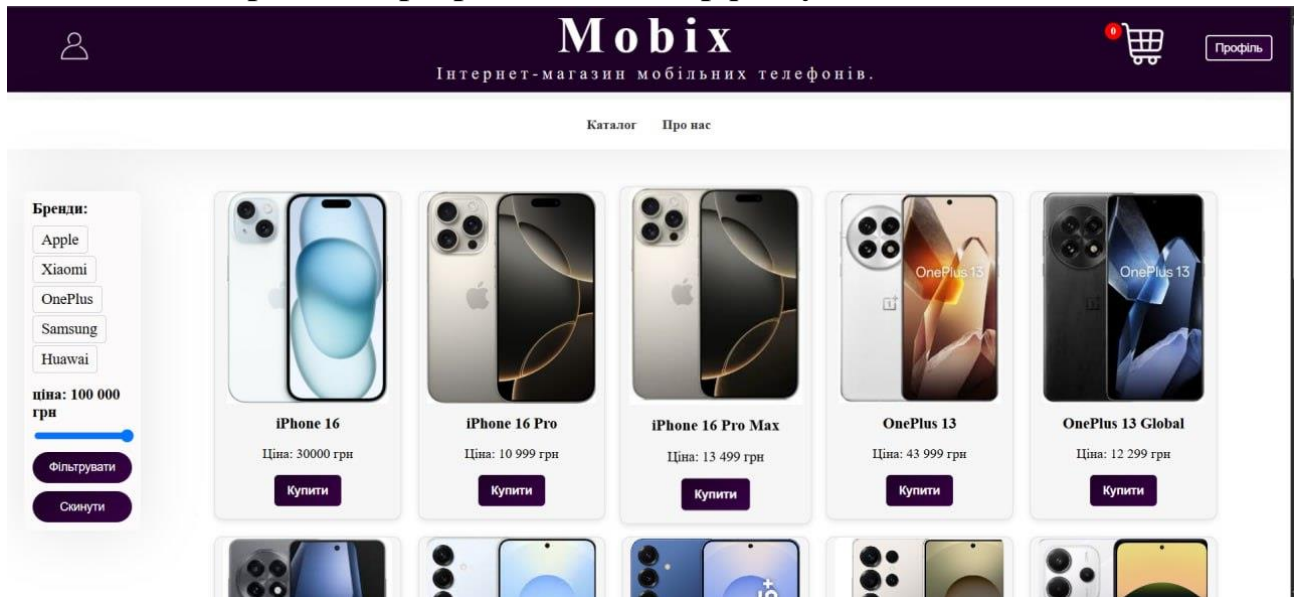


Рисунок Б.1 – Головна сторінка інтернет-магазину

```

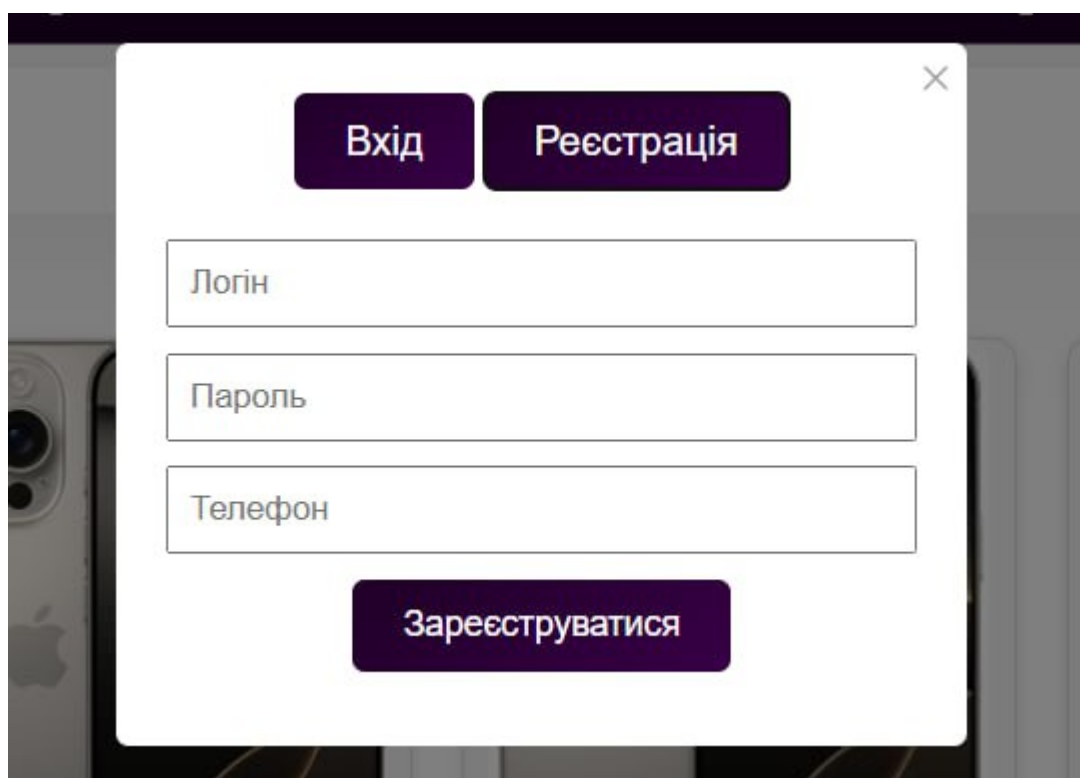
{} users.json ×
{} users.json > ...
1  [
2  {
3    "name": "test1",
4    "password": "$2b$10$aSjZe4n6iuzP4r/2Q5EtkOY6wSgG.PsfdXtpo7V2xml7ybmbBt.wG",
5    "phone": "3809988877"
6  },
7  {
8    "name": "test2",
9    "password": "$2b$10$hbMxHK/eDJQoRtTnGpxh1OmROAVkANxqToPcA93NRkJwRg0InRQLy",
10   "phone": "3809988877"
11  },
12  {
13   "name": "test3",
14   "password": "$2b$10$tZhYyKp6xkGvPNwsvPJY9.5d9l952YE6APLRIos9puhsFJXkj3AB.",
15   "phone": "3809988877"
16  }
17 ]

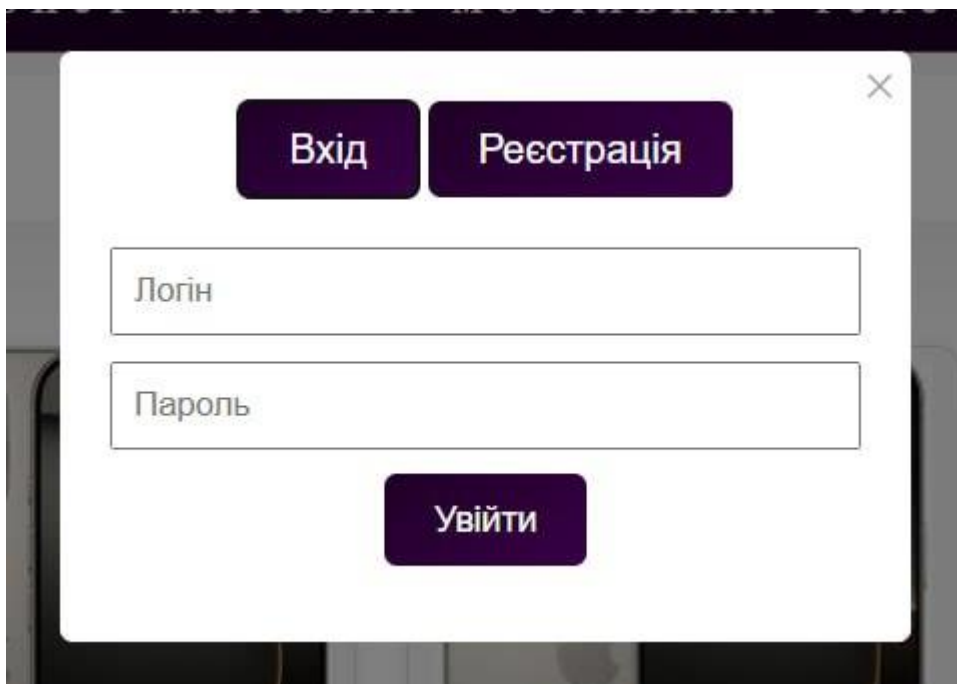
```

Рисунок Б.2 – Перегляд бази товарів (каталог)



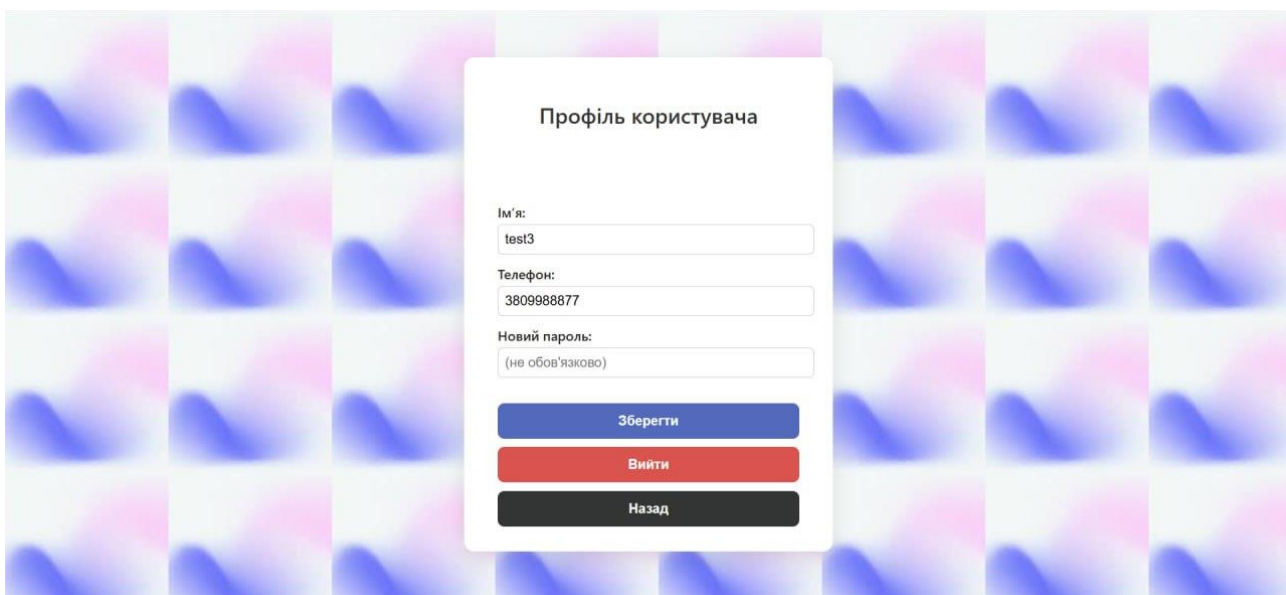
Рисунок Б.3 – Сторінка кошика користувача





A screenshot of a web application interface showing a login and registration form. The form is contained within a white modal window with a close button (X) in the top right corner. At the top of the modal, there are two dark purple buttons: "Вхід" (Login) on the left and "Реєстрація" (Registration) on the right. Below these buttons are two input fields: "Логін" (Login) and "Пароль" (Password). At the bottom of the modal is a dark purple button labeled "Увійти" (Login).

**Рисунок Б.4 – Форма реєстрації та входу**



A screenshot of a user profile form titled "Профіль користувача" (User Profile). The form is displayed in a white modal window over a background of a grid of blue and pink abstract shapes. The form contains three input fields: "Ім'я:" (Name) with the value "test3", "Телефон:" (Phone) with the value "3809988877", and "Новий пароль:" (New password) with the placeholder text "(не обов'язково)". Below the input fields are three buttons: "Зберегти" (Save) in blue, "Вийти" (Logout) in red, and "Назад" (Back) in black.

**Рисунок Б.5 – Особистий кабінет користувача**

## Додаток В. Фрагменти коду

### В.1 – Реєстрація користувача

#### HTML

```

<div id="authButtons">
<button class="bb" id="subtn">Вхід</button>
<div id="proDrpdwn">
<button onclick="goToProfile()">Профіль</button>
<button onclick="logout()">Вийти</button>
</div>
</div>

<div id="neokno" class="mdl">
<div class="mdlcon">
<span class="cl" onclick="closeModal()">&times;</span>
<div class="tab-buttons">
<button class="tabbtn act" onclick="showTab('login')">Вхід</button>
<button class="tabbtn" onclick="showTab('register')">Реєстрація</button>
</div>

<div id="loginTab" class="tab active">
<form id="loginForm">
<input name="login" placeholder="Логін" required /><br />
<input name="password" type="password" placeholder="Пароль" required /><br />
<button type="submit">Увійти</button>
</form>
</div>

<div id="registerTab" class="tab">
<form id="registerForm">
<input name="login" placeholder="Логін" required /><br />
<input name="password" type="password" placeholder="Пароль" required /><br />
<input name="phone" placeholder="Телефон" required /><br />
<button type="submit">Зареєструватися</button>
</form>
</div>
<div id="atMsg"></div>

```

#### JS

```

const loginForm = document.getElementById('loginForm');
const registerForm = document.getElementById('registerForm');
const atMsg = document.getElementById('atMsg');
const singUpBtn = document.getElementById('subtn');
loginForm.addEventListener('submit', async e => {
e.preventDefault();

const data = {name: loginForm.login.value,password: loginForm.password.value};
const res = await fetch('/login', {method: 'POST',headers: { 'Content-Type': 'application/json' },body: JSON.stringify(data)});
const result = await res.json().catch(() => null);if (res.ok) {
localStorage.setItem('token', result.token);
singUpBtn.textContent = 'Профіль';
singUpBtn.onclick = goToProfile;
singUpBtn.dataset.username = data.name;
atMsg.textContent = "";
closeModal();} else {atMsg.textContent = result?.error || result || 'Помилка входу'}});

window.onload = () => {
const token = localStorage.getItem('token');if (token) {try {

```

```

const payload = JSON.parse(atob(token.split('.')[1]));
singUpBtn.textContent = 'Профіль';
singUpBtn.onclick = goToProfile;
singUpBtn.dataset.username = payload.name;} catch {
localStorage.removeItem('token');}}};

function goToProfile() {
const token = localStorage.getItem('token');if (!token) {alert('Спочатку авторизуйтесь');return;}
window.location.href = 'profile.html';}

function logout() {
localStorage.removeItem('token');
singUpBtn.textContent = 'Вхід';
singUpBtn.onclick = openModal;}

const neoknomodal = document.getElementById("neokno");
const proDrpdwn = document.getElementById("proDrpdwn");
const singUpBtn = document.getElementById("subtn");
function openModal() {
neoknomodal.style.display = "block";}

function closeModal() {
neoknomodal.style.display = "none";}

function showTab(tab) {
document.querySelectorAll('.tab').forEach(t => t.classList.remove('active'));
document.querySelectorAll('.tabbtn').forEach(b => b.classList.remove('active'));
document.getElementById(tab + 'Tab').classList.add('active');
event.target.classList.add('active');}
singUpBtn.onclick = openModal;

window.onclick = function(event) {
if (event.target === neoknomodal) closeModal();
if (!proDrpdwn.contains(event.target) || event.target !== singUpBtn) {
proDrpdwn.style.display = "none";}}

document.getElementById('registerForm').addEventListener('submit', async e => {
e.preventDefault();
const form = e.target;
const data = {name: form.login.value,password: form.password.value,phone: form.phone.value};
const res = await fetch('/register', {method: 'POST',headers: { 'Content-Type': 'application/json' },body: JSON.stringify(data)});
const text = await res.text();
document.getElementById('atMsg').textContent = text;
});

document.getElementById('loginForm').addEventListener('submit', async e => {
e.preventDefault();
const form = e.target;
const data = {name: form.login.value,password: form.password.value};
const res = await fetch('/login', {method: 'POST',headers: { 'Content-Type': 'application/json' },body: JSON.stringify(data)});
const text = await res.text();if (res.ok) {closeModal();singUpBtn.textContent = 'Профіль';
singUpBtn.onclick = () => window.location.href = 'profile.html';
document.getElementById('atMsg').textContent = "}; else {
document.getElementById('atMsg').textContent = text;}}});

async function logout() {
await fetch('/logout', { method: 'POST' });
singUpBtn.textContent = 'Вхід';
singUpBtn.onclick = openModal;
proDrpdwn.style.display = 'none';}

```

```
function goToProfile() {
const username = singUpBtn.dataset.username;if (username) {window.location.href =
`profile.html?user=${encodeURIComponent(username)}`; } else {alert("Ім'я користувача не знайдено.");}}
```

## NODE

```
const express = require('express');
const fs = require('fs');
const bcrypt = require('bcrypt');
const jwt = require('jsonwebtoken');
const rateLimit = require('express-rate-limit');
const { body, validationResult } = require('express-validator');

require('dotenv').config({ path: '.env' });

const app = express();
const USERS_FILE = 'users.json';
const SALT_ROUNDS = 10;

const JWT_SECRET = process.env.JWT_SECRET;
const PORT = process.env.PORT || 3000;

app.use(express.static(__dirname));
app.use(express.json());

const loginLimiter = rateLimit({
  windowMs: 15 * 60 * 1000,
  max: 10,
  handler: (req, res) => {
    res.status(429).send("Занадто багато спроб входу. Спробуйте, будь ласка, пізніше.");
  }
});

function loadUsers() {
  if (fs.existsSync(USERS_FILE)) {
    const data = fs.readFileSync(USERS_FILE, 'utf-8');
    return JSON.parse(data);
  }
  return [];
}

function saveUsers(users) {
  fs.writeFileSync(USERS_FILE, JSON.stringify(users, null, 2));
}

app.post('/register', [
  body('name').trim().isLength({ min: 3 }).withMessage('Ім'я повинно містити щонайменше 3 символи'),
  body('password').isLength({ min: 6 }).withMessage('Пароль повинен містити щонайменше 6 символів'),
  body('phone').matches(/^\+?\d{7,15}$).withMessage('Телефон вказаний неправильно')
], async (req, res) => {
  const errors = validationResult(req);
  if (!errors.isEmpty()) {
    const messages = errors.array().map(err => err.msg).join(', ');
    return res.status(400).send(messages);
  }

  const { name, password, phone } = req.body;
  const users = loadUsers();
```

```

if (users.find(u => u.name === name)) {
  return res.status(400).send('Користувач із таким ім'ям вже існує');
}

const hash = await bcrypt.hash(password, SALT_ROUNDS);
users.push({ name, password: hash, phone });
saveUsers(users);

res.send('Ресстрація пройшла успішно!');
});

app.post('/login', loginLimiter, async (req, res) => {
  const { name, password } = req.body;
  const users = loadUsers();

  const user = users.find(u => u.name === name);
  if (!user) return res.status(400).send('Користувача не знайдено');

  const valid = await bcrypt.compare(password, user.password);
  if (!valid) return res.status(400).send('Невірний пароль');

  const token = jwt.sign({ name: user.name }, JWT_SECRET, { expiresIn: '7d' });

  res.json({ token });
});

function authenticateToken(req, res, next) {
  const authHeader = req.headers['authorization'];
  const token = authHeader && authHeader.split(' ')[1];
  if (!token) return res.status(401).send('Не авторизовано');

  jwt.verify(token, JWT_SECRET, (err, user) => {
    if (err) return res.status(403).send('Токен не валідний');
    req.user = user;
    next();
  });
}

app.get('/profile-data', authenticateToken, (req, res) => {
  const users = loadUsers();
  const user = users.find(u => u.name === req.user.name);
  if (!user) return res.status(404).send('Користувача не знайдено');

  res.json({ name: user.name, phone: user.phone });
});

app.put('/update-profile', authenticateToken, [
  body('name').trim().isLength({ min: 3 }).withMessage('Ім'я повинно містити щонайменше 3 символи'),
  body('phone').matches(/^\\+?\\d{7,15}$\\/).withMessage('Телефон вказаний неправильно'),
  body('password').optional().isLength({ min: 6 }).withMessage('Пароль повинен містити щонайменше 6 символів')
], async (req, res) => {
  const errors = validationResult(req);
  if (!errors.isEmpty()) {
    const messages = errors.array().map(err => err.msg).join(', ');
    return res.status(400).send(messages);
  }

  const { name, phone, password } = req.body;
  const users = loadUsers();

```

```

const index = users.findIndex(u => u.name === req.user.name);

if (index === -1) return res.status(404).send('Користувача не знайдено');

if (name !== req.user.name && users.find(u => u.name === name)) {
  return res.status(400).send('Інше ім'я вже зайнято');
}

users[index].name = name;
users[index].phone = phone;

if (password) {
  users[index].password = await bcrypt.hash(password, SALT_ROUNDS);
}

saveUsers(users);

const newToken = jwt.sign({ name }, JWT_SECRET, { expiresIn: '7d' });

res.json({ token: newToken });
});

app.listen(PORT, () => {
  console.log('Сервер запущено: http://localhost:${PORT}');
});

```

## В.2 – Фільтрація товарів

```

div id="content">
<div id="filtr">
<div>
<label>Бренди:</label>
<div id="brandOptions">
<span class="brnd" data-brand="iPhone">Apple</span>
<span class="brnd" data-brand="Xiaomi">Xiaomi</span>
<span class="brnd" data-brand="OnePlus">OnePlus</span>
<span class="brnd" data-brand="Samsung">Samsung</span>
<span class="brnd" data-brand="Huawai">Huawai</span>
</div>
</div>

<div style="margin-top: 15px;">
<label for="prcSlider" style="font-weight: bold; display: block; margin-bottom: 6px;">ціна: <span id="priceValue">0</span>
грн</label>
<input type="range" id="prcSlider" min="0" max="100000" step="500" value="100000" style="width: 100%;">
</div>
<button onclick="applyFilter()" style="width: 100%; margin-top: 10px;">Фільтрувати</button>
<button onclick="resetFilter()" style="width: 100%; margin-top: 10px;">Скинути</button>
</div>

```

```

const prcSlider = document.getElementById('prcSlider');
const priceValue = document.getElementById('priceValue');
priceValue.textContent = Number(prcSlider.value).toLocaleString('uk-UA');
prcSlider.addEventListener('input', () => {
  priceValue.textContent = Number(prcSlider.value).toLocaleString('uk-UA');});

document.querySelectorAll('.brnd').forEach(item => {
  item.addEventListener('click', () => {
    item.classList.toggle('sel');});});

```

```

function applyFilter() {
const selectedBrands = Array.from(document.querySelectorAll('.brnd.sel'))
.map(el => el.dataset.brand);
const maxPrice = parseFloat(prcSlider.value);
const showAllBrands = selectedBrands.length === 0;
document.querySelectorAll('.productrd').forEach(card => {
const brand = card.dataset.brand;
const price = parseFloat(card.dataset.price);
const brandMatch = showAllBrands || selectedBrands.includes(brand);
const priceMatch = !isNaN(maxPrice) ? price <= maxPrice : true;
if (brandMatch && priceMatch) {
card.style.display = 'block';} else {
card.style.display = 'none';}});

function resetFilter() {
document.querySelectorAll('.brnd.sel').forEach(el => {
el.classList.remove('sel');});
prcSlider.value = 100000;
priceValue.textContent = '100000';
document.querySelectorAll('.productrd').forEach(card => {
card.style.display = 'block';});}

```

### В.3 – Робота з кошиком

```

<div id="crtMdl">
<div class="crtcon">
<span id="cleCrtBtn" onclick="closeCart()">✕</span>
<h2>Кошик</h2>
<div id="crtItms"></div>
<p><strong>Всього:</strong><span id="cartTotal">0</span>€</p>
<button onclick="clearCart()">Очистити</button>
<button onclick="shwchkoutFrm()">Оформити замовлення</button>

<div id="chkoutFrm">
<h3>Дані для замовлення</h3>
<label>Ім'я:<br><input type="text" id="customerName" placeholder="Ваше ім'я"></label><br><br>
<label>Телефон:<br><input type="tel" id="customerPhone" placeholder="+380..."></label><br><br>
<label>Адреса доставки:<br><textarea id="customerAddress" rows="2" placeholder="Адреса"></textarea></label><br><br>

<button onclick="submitOrder()">Підтвердити замовлення</button>
<button onclick="cancelCheckout()">Скасувати</button>
<pre id="ordrsm"></pre>
</div>

```

```

let cart = JSON.parse(localStorage.getItem('cart')) || [];

function saveCart() {
localStorage.setItem('cart', JSON.stringify(cart));
}

function addToCart(name, price) {
let image = "";
const products = document.querySelectorAll(".productrd");
products.forEach(product => {
const title = product.querySelector("h3").innerText;
if (title === name) {
image = product.querySelector("img").src;
}
});
}

```

```

const item = cart.find(item => item.name === name);
if (item) {
  item.quantity += 1;
} else {
  cart.push({ name, price, quantity: 1, image });
}

saveCart();
updateCartUI();
updateProductButtons();
}

function updateProductButtons() {
  const products = document.querySelectorAll(".product");

  products.forEach(product => {
    const title = product.querySelector("h3").innerText;
    const button = product.querySelector("button");
    const itemInCart = cart.find(item => item.name === title);
    if (itemInCart) {
      button.textContent = "У кошкику";
    } else {
      button.textContent = "Купити";
    }
  });
}

function updateCartUI() {
  const crtItems = document.getElementById('cartItems');
  const crtTotal = document.getElementById('cartTotal');
  const cartCount = document.getElementById('cartCount');
  crtItems.innerHTML = "";
  let total = 0;
  let itemCount = 0;
  cart.forEach(item => {
    const itemTotal = item.price * item.quantity;
    itemCount += item.quantity;
    total += itemTotal;
    const itemElement = document.createElement('div');
    itemElement.innerHTML = `

<div>
<strong>${item.name}</strong><br>
${item.price} x ${item.quantity}<br>
<button onclick="changeQuantity('${item.name}', -1)"></button>
<button onclick="changeQuantity('${item.name}', 1)"></button>
<button onclick="removeFromCart('${item.name}')">Видалити</button>
</div>
`;
    crtItems.appendChild(itemElement);
  });
  crtTotal.textContent = total;
  cartCount.textContent = itemCount;
}

function changeQuantity(name, delta) {
  const item = cart.find(item => item.name === name);
  if (!item) return;
  item.quantity += delta;
}

```

```

if (item.quantity <= 0) {
removeFromCart(name);
} else {
saveCart();
updateCartUI();
}
}

function removeFromCart(name) {
cart = cart.filter(item => item.name !== name);
saveCart();
updateCartUI();
updateProductButtons();
}

function clearCart() {
cart = [];
saveCart();
updateCartUI();
updateProductButtons();
}

function openCart() {
document.getElementById('crtMdl').style.display = 'flex';
updateCartUI();
updateProductButtons();
document.getElementById('chkoutFrm').style.display = 'none';
document.getElementById('ordrsm').textContent = "";
}

function closeCart() {
document.getElementById('crtMdl').style.display = 'none';
}

function shwchkoutFrm() {
if (cart.length === 0) {
alert("Кошик порожній!");
return;
}
document.getElementById('chkoutFrm').style.display = 'block';
document.getElementById('ordrsm').textContent = "";
}

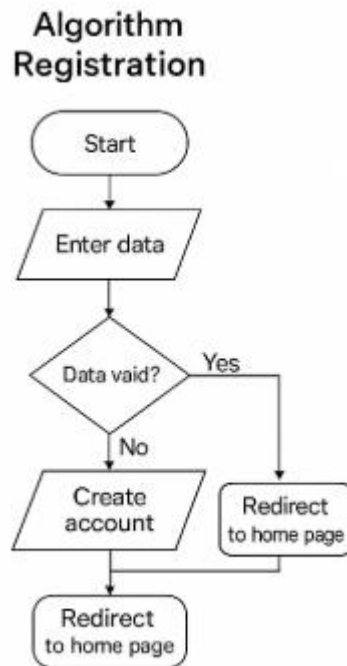
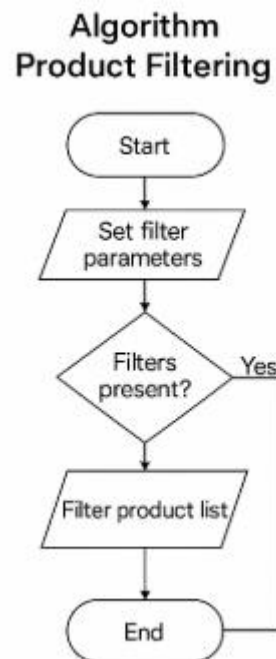
function cancelCheckout() {
document.getElementById('chkoutFrm').style.display = 'none';
document.getElementById('ordrsm').textContent = "";
}

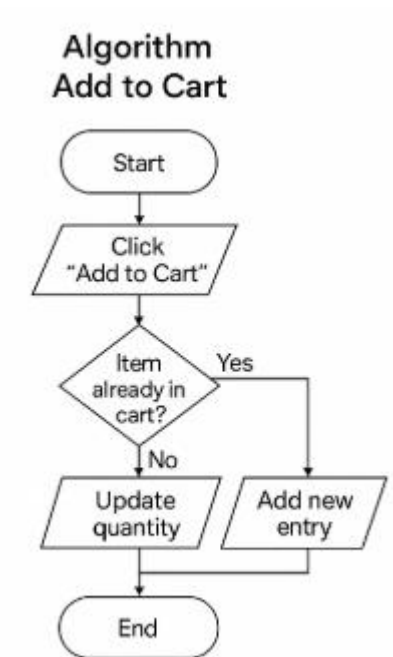
function submitOrder() {
const name = document.getElementById('customerName').value.trim();
const phone = document.getElementById('customerPhone').value.trim();
const address = document.getElementById('customerAddress').value.trim();

if (!name || !phone || !address) {
alert("Будь ласка заповніть всі поля для оформлення замовлення");
return;
}
let summary = `Дякуємо за замовлення, ${name}!\n\nВаше замовлення:\n`;
cart.forEach(item => {
summary += `${item.name} x ${item.quantity} = ${item.price * item.quantity}€\n`;
});
}

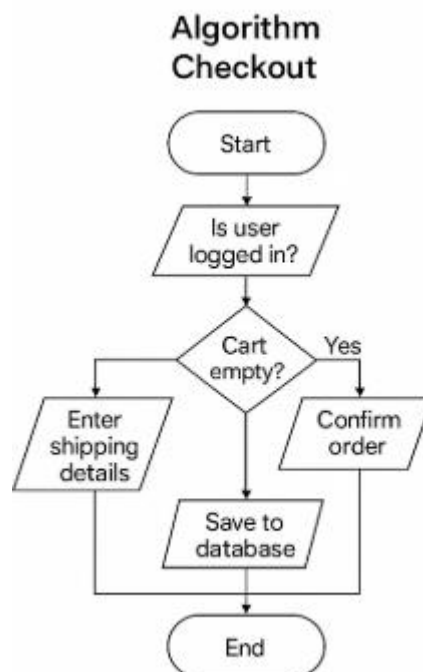
```

```
});  
summary += \nВсього: ${document.getElementById('cartTotal').textContent}€\n\n;  
summary += Доставка за адресою:\n${address}\nТелефон для зв'язку: ${phone};  
document.getElementById('ordrsm').textContent = summary;  
cart = [];  
saveCart();  
updateCartUI();  
updateProductButtons();  
alert("Замовлення оформлено Дякуємо!");  
}  
  
document.addEventListener('DOMContentLoaded', () => {  
updateCartUI();  
updateProductButtons();  
});
```

**Додаток Г. Блок-схеми алгоритмів****Рисунок Г.1 – Алгоритм реєстрації користувача****Рисунок Г.2 – Алгоритм фільтрації товарів**



**Рисунок Г.3 – Алгоритм додавання товару до кошика**



**Рисунок Г.4 – Алгоритм оформлення замовлення**

## Додаток Е. Діаграми

