

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Веборієнтована інформаційна система інтернет-магазину електроніки

Назва теми

КВРІСТ 2101030.21.01.03 ПЗ

Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»

Шифр, назва

Освітня програма «Інформаційні системи та технології»

Назва

Виконав: студент IV курсу, група ІСТ-21-1


Підпис

Андрій ДЕРКАЧ

Ініціали, прізвище

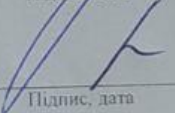
Керівник


Підпис, дата

Єлизавета ГНАТЧУК

Ініціали, прізвище

Нормоконтролер


Підпис, дата

Тетяна КИСІЛЬ

Ініціали, прізвище

До захисту допускаю:
зав. кафедри комп'ютерної
інженерії та інформаційних
систем


Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

«02» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

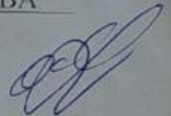
Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.



**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Деркач Андрій Олексійович

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Веборієнтована інформаційна система інтернет-магазину електроніки

Керівник проекту (роботи) Єлизавета ГНАТЧУК, д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Теоретичні основи досліджуваної теми

Вибір технологій та структурно-функціональне моделювання

Реалізація проєкту

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Архітектура магазину

Архітектура клієнтської частини

Контекстна

діаграма

та

її

декомпозиція

6. Консультанти розділів дипломного проекту (роботи)

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| Нормоконтроль | Тетяна КИСІЛЬ, доцент кафедри КПС | | |
| Антиплагіат | Андрій Нічепорук, доцент кафедри КПС | | |

7. Дата видачі завдання « 10 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

| №з/п | Назва етапів (розділів) дипломного проекту (роботи) | Термін виконання етапів проекту (роботи) | Примітка |
|------|--|--|----------|
| 1 | Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником | 10.01.2025 | виконано |
| 2 | Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження | 01.02.2025 | виконано |
| 3 | Робота над розділом 1 – дослідження предметної області та постановка задачі | 01.03.2025 | виконано |
| 4 | Робота над розділом 2 – вибір компонентів для проєктування веборієнтованої інформаційної системи інтернет-магазину електроніки | 01.04.2025 | виконано |
| 5 | Робота над розділом 3 – проєктування системи веборієнтованої інформаційної системи інтернет-магазину електроніки | 25.04.2025 | виконано |
| 6 | Оформлення пояснювальної записки згідно вимог | 20.05.2025 | виконано |
| 7 | Попередній захист ВКР | 22.05.2025 | виконано |
| 8 | Захист ВКР на засіданні ЕК | Червень 2025 року | |

Студент

Підпис

Андрій ДЕРКАЧ
Ініціали, прізвище

Керівник роботи

Підпис

Єлизавета ГНАТЧУК
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Веборієнтована інформаційна система інтернет магазину електроніки».

Автор роботи: Андрій Деркач.

Керівник роботи: Гнатчук Єлизавета Геннадіївна.

Пояснювальна записка: 66 с., 47 рис., 4 табл., 3 дод., 65джерел.

Графічна частина: 3 креслення.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, ІНТЕРНЕТ-МАГАЗИН, ЕЛЕКТРОННА КОМЕРЦІЯ, ІНТЕРНЕТ МАГАЗИН, БАЗА ДАНИХ.

Мета роботи – автоматизація та покращення існуючих рішень функціонування інтернет-магазину електроніки.

Об'єктом дослідження є сучасні інтернет-магазини, що спеціалізуються на продажу електронних товарів.

Предметом дослідження є методи та принципи побудови веборієнтованих інформаційних систем, спрямованих на автоматизацію процесів керування та взаємодії в інтернет-магазинах електронних товарів.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.







Підпис студента

30.05.2025

Дата

Зміст

| | |
|---|----|
| ВСТУП | 3 |
| 1 ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ТЕМИ | 5 |
| 1.1 Аналіз предметної області і виявлення наявних проблем і завдань | 5 |
| 1.2 Огляд та аналіз відомих інтернет-магазинів | 7 |
| 1.3 Аналітика онлайн продажів | 13 |
| 1.4 Архітектура та логіка взаємодії користувача в інтернет-магазині | 15 |
| 1.5 Аналіз основних вимог та постановка задачі..... | 16 |
| 1.6 Висновки до першого розділу..... | 17 |
| 2 ВИБІР ТЕХНОЛОГІЙ ТА СТРУКТУРНО-ФУНКЦІОНАЛЬНЕ МОДЕЛЮВАННЯ | 18 |
| 2.1 Структурно-функціональне моделювання інтернет-магазину електронних товарів..... | 18 |
| 2.2 Вибір редактора коду | 24 |
| 2.3 Вибір мови програмування | 26 |
| 2.4 База даних та серверна частина | 27 |
| 2.5 Менеджер пакетів, контроль версій, хостинг та хешування | 31 |
| 2.6 Висновки до другого розділу | 35 |
| 3 РЕАЛІЗАЦІЯ ПРОЄКТУ | 37 |
| 3.1 Структура проєкту | 37 |
| 3.2 Реалізація фронтенду | 39 |
| 3.3 Реалізація Бекенду та тестування..... | 60 |
| 3.4 Висновки до третього розділу..... | 68 |
| ВИСНОВКИ | 69 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ | 70 |
| ДОДАТОК А | 77 |
| ДОДАТОК Б | 78 |
| ДОДАТОК В | 79 |

| | | | | |
|---|------|---------------|---|--------|
| КвРІСТ.2101030.21.01.3 ПЗ | | | | |
| Зм. | Арк. | № докум. | Підпис | Дата |
| Виконав | | Андрій Деркач |  | |
| Перевір. | | Євген Гнатюк |  | |
| Н.контр. | | Тетяна КИСІЛЬ |  | 2021 |
| Затвер. | | Ольга ПАВЛОВА |  | 06.05 |
| Веборієнтована інформаційна система інтернет-магазину електроніки | | | Літера | Аркуші |
| | | | у | 2 |
| | | | ХНУ ICT-21-1 | |

ВСТУП

Сучасний світ характеризується визначальною роллю інформаційних технологій у забезпеченні функціонування комерційних структур. Електронна комерція стала невід'ємним елементом глобальної економіки, радикально трансформуючи традиційні моделі торгівлі та взаємодії між суб'єктами бізнесу та споживачами. Компанії різного розміру та галузевої приналежності дедалі активніше застосовують програмні рішення для автоматизації процесів, ефективного керування значними обсягами даних та безперебійної комунікації зі споживачами в реальному часі. У цьому контексті веборієнтовані інформаційні системи набули особливого значення, оскільки вони забезпечують безпрецедентну доступність сервісів цілодобово з будь-якої точки світу, гнучкість адаптації до зростаючих потреб ринку та виняткову зручність у використанні як для кінцевих користувачів, так і для адміністративного персоналу.

Дієве управління інформаційними ресурсами є фундаментальним чинником для досягнення стійкого успіху та забезпечення конкурентоспроможності будь-якої організації на сучасному ринку. Застарілі підходи до менеджменту, що ґрунтуються на паперовій документації, ручних операціях або розрізнених програмних засобах без належної інтеграції, часто демонструють низьку ефективність. Це спричиняє значні часові втрати, виникнення численних помилок через людський фактор, ускладнення процедур пошуку та консолідації інформації, а також уповільнює процес прийняття критично важливих управлінських рішень. Натомість, веборієнтовані інформаційні системи пропонують єдине централізоване сховище даних, забезпечують комплексну автоматизацію бізнес-процесів від обробки замовлень до управління складськими запасами та значно покращують взаємодію між усіма учасниками процесу, що в кінцевому підсумку позитивно впливає на задоволеність клієнтів та продуктивність роботи персоналу.

Особливої актуальності ця проблематика набуває у сфері торгівлі електронними товарами. Цей сегмент ринку вирізняється високою динамікою,

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|-----------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 3 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

постійним оновленням асортименту, швидким виходом нових моделей та високими очікуваннями споживачів щодо функціональності та зручності онлайн-платформ. Саме тому інтернет-магазинам критично важливо не тільки оперативно реагувати на ці зміни, але й пропонувати інноваційні рішення, що забезпечують досвід покупок. Це включає в себе швидку та безпечну обробку транзакцій, персоналізовані рекомендації та інтуїтивно зрозумілу навігацію. Конкуренція в цій галузі вимагає від інтернет-магазинів не просто присутності в мережі, а створення високотехнологічної, безпечної та зручної системи, здатної ефективно обробляти значні обсяги даних та забезпечувати користувацький досвід.

Дана бакалаврська робота присвячена процесу створення веборієнтованої інформаційної системи, яка слугуватиме основою для сучасного інтернет-магазину електронних товарів. В рамках дослідження буде детально проаналізовано предметну галузь, зокрема виявлення наявних проблем та завдань, що є характерними для інтернет-магазину електроніки. Особлива увага буде зосереджена на виборі оптимальних технологій розробки, проектуванні структури системи, що забезпечить її масштабованість та надійність, кодуванню основних функціональних модулів, а також етапам тестування для гарантування роботи. Розроблене рішення має на меті не лише оптимізувати інформаційні потоки та підвищити продуктивність праці персоналу завдяки автоматизації рутинних операцій, але й суттєво поліпшити якість обслуговування клієнтів, пропонуючи їм інтуїтивно зрозумілий інтерфейс та зручний функціонал. Успішна реалізація цього проєкту сприятиме посиленню конкурентоспроможності магазину на ринку електронної комерції.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|-----------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 4 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

1 ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ТЕМИ

1.1 Аналіз предметної області і виявлення наявних проблем і завдань

Сучасна електронна комерція, у своїй розвиненій формі, є підсумком тривалого процесу еволюції методів торгівлі, тісно переплетеного з прогресом інформаційних технологій. Від елементарних онлайн-каталогів до комплексних платформ, що забезпечують повний спектр операцій купівлі-продажу, веборієнтовані інформаційні системи стали визначальним фактором у кардинальній трансформації роздрібною торгівлі.

Широке розповсюдження мережі Інтернет ознаменувало появу перших інтернет-магазинів (рис. 1.1) . На початковому етапі це були переважно прості веб-сайти, що виконували функцію онлайн-каталогів товарів.

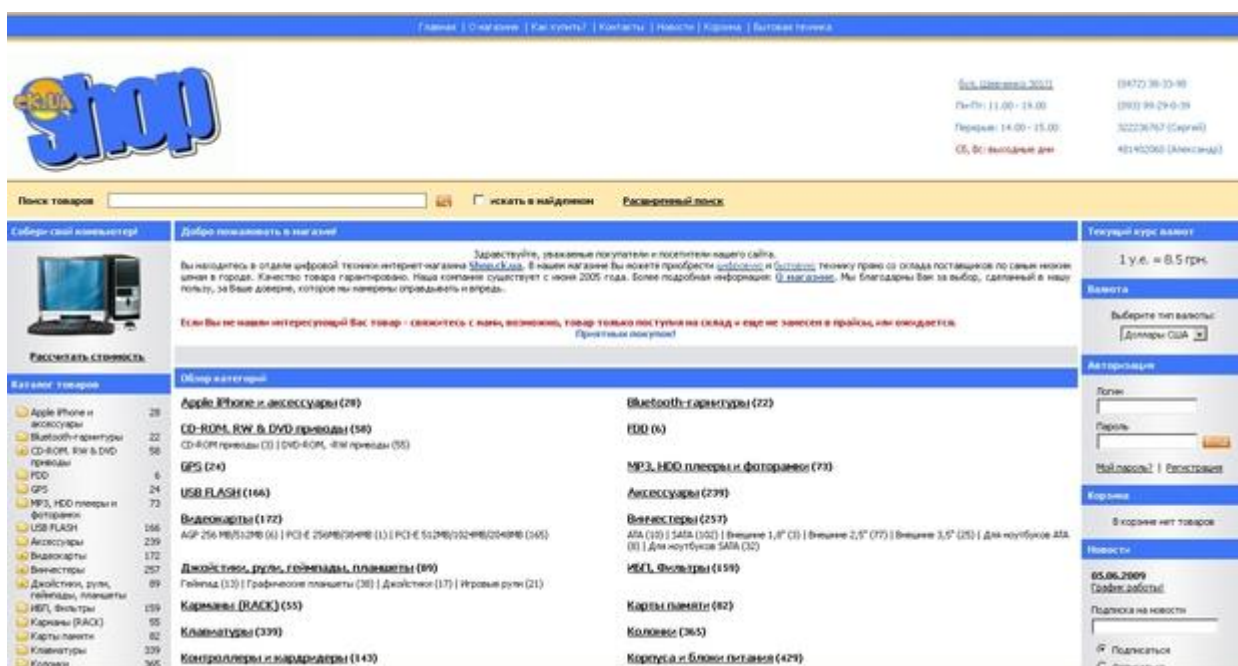


Рисунок 1.1 – Перший інтернет магазин в Черкасах [1]

Впровадження та популяризація таких платіжних систем, як PayPal, значно підвищили рівень безпеки та зручності онлайн-платежів, що стало імпульсом для зростання електронної комерції.

Зі стрімким зростанням використання смартфонів, мобільна комерція (m-commerce) набула статусу ключового сегменту онлайн-торгівлі. Інтернет-магазини активно оптимізуються під мобільні пристрої, розробляються спеціалізовані мобільні додатки для забезпечення більш зручного та інтегрованого досвіду покупок.

Персоналізація пропозицій, інтеграція штучного інтелекту, аналіз великих даних (Big Data), розвиток соціальної комерції (social commerce) та впровадження омніканальних стратегій є лише деякими з ключових тенденцій, що активно формують майбутній ландшафт електронної комерції.

Галузь електронної комерції, що спеціалізується на продажу електроніки, є однією з найбільш динамічних та висококонкурентних. Постійні технологічні інновації, часті випуски нових моделей електронних пристроїв та широкий асортимент товарів створюють специфічні завдання та виклики для функціонування інтернет-магазинів у цьому сегменті ринку.

Сьогодні значна частина комерційної діяльності перемістилася в онлайн-середовище. Електронна комерція відкриває широкі можливості як для продавців, так і для покупців, забезпечуючи зручність та доступність. У цьому контексті веборієнтовані інформаційні системи стали ключовим елементом успішного функціонування бізнесу в даній сфері.

Застосування традиційних підходів до керування магазином, таких як ведення обліку за допомогою електронних таблиць або використання відокремлених програмних засобів, часто виявляється неефективним і спричиняє низку проблем:

- ускладнене управління широким асортиментом;
- обмежені аналітичні можливості;
- недостатня автоматизація процесів;
- обмежена взаємодія з клієнтами.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|-----------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 6 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Розробка веборієнтованої інформаційної системи для магазину електронних товарів має на меті усунення зазначених проблем та забезпечення ефективного управління бізнес-процесами.

Основні цілі такої системи включають:

- керування товарним каталогом;
- опрацювання замовлень;
- керування клієнтською базою;
- аналіз даних;
- інтеграція з зовнішніми системами.

Для створення ефективної веборієнтованої інформаційної системи магазину електронних товарів необхідно:

- застосовувати актуальні веб-технології для розробки зручного та функціонального користувацького інтерфейсу;
- створити надійну базу даних для зберігання критично важливої інформації про товари, клієнтів та замовлення;
- реалізувати спеціалізовані алгоритми для автоматизації ключових бізнес-процесів;
- гарантувати високий рівень безпеки даних та фінансових транзакцій.

Аналіз предметної області чітко демонструє, що розробка веборієнтованої інформаційної системи є важливим та актуальним завданням для забезпечення успішного функціонування інтернет-магазину, що спеціалізується на продажу електронних товарів.

1.2 Огляд та аналіз відомих інтернет-магазинів

У даному розділі я розгляну українські та світові аналоги. Український онлайн-магазин Rozetka (рис. 1.2) – це провідний український маркетплейс, який пропонує широкий спектр товарів, що перевищує мільйон найменувань у різних категоріях, включаючи електроніку, побутову техніку, одяг, товари для дому,

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|-----------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 7 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

косметичні засоби тощо. Характеризується розвинуеною мережею пунктів видачі замовлень та якісною сервісною підтримкою клієнтів. Платформа активно використовує сучасні технології для аналізу поведінки користувачів.

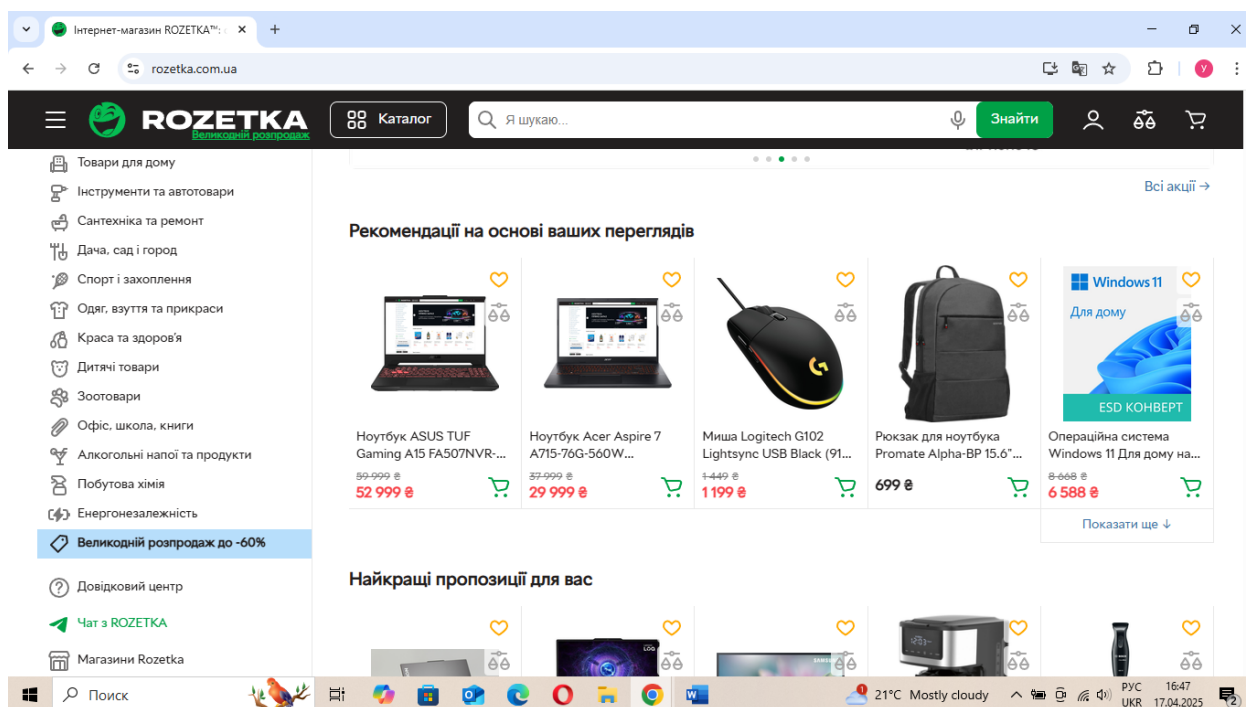


Рисунок 1.2 – Rozetka [2]

Foxtrot (рис. 1.3) – один із ключових гравців на українському ринку побутової техніки та електроніки. Даний онлайн-магазин активно впроваджує зручні варіанти доставки, вигідні кредитні пропозиції та програму лояльності для постійних покупців. Foxtrot використовує сучасні технології для оптимізації процесу онлайн-покупок, включаючи зручну навігацію по сайту, систему фільтрів для швидкого пошуку товарів, а також інтеграцію з платіжними системами. Компанія також активно розвиває мобільний додаток для забезпечення зручних покупок зі смартфонів. Окрім цього, магазин регулярно проводить акції та розпродажі, пропонуючи конкурентні ціни на популярні моделі електроніки, що приваблює нових клієнтів.

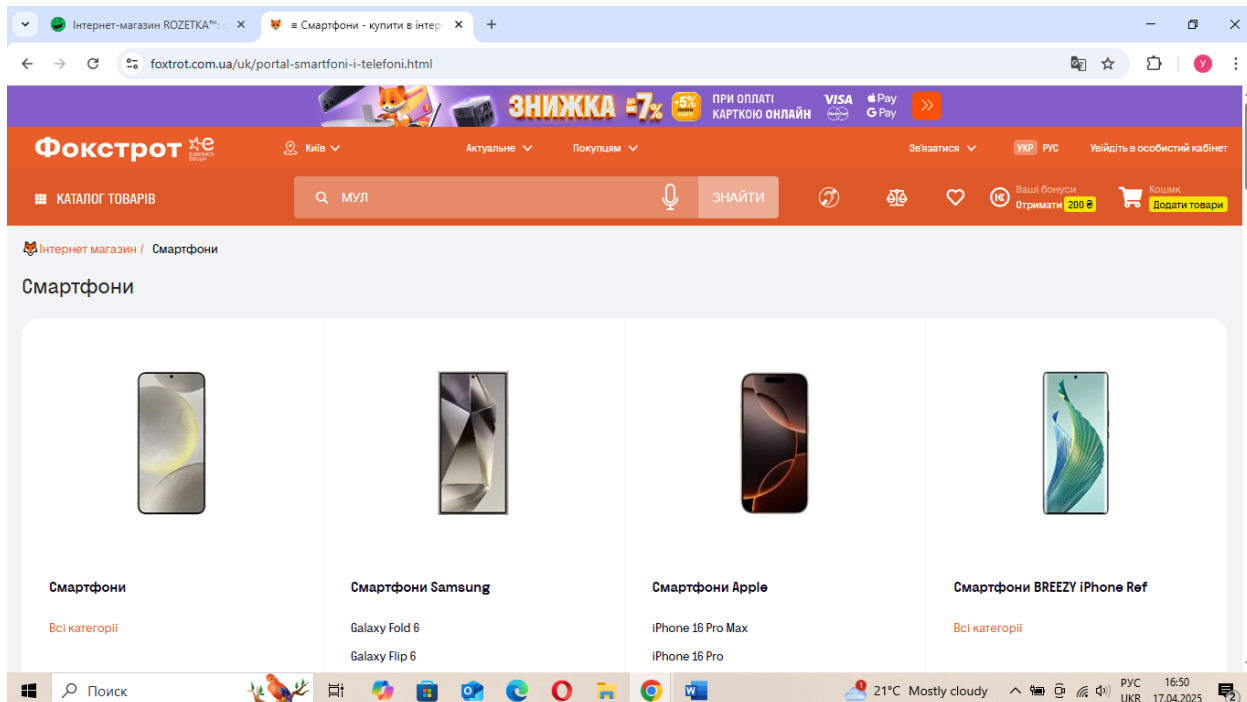


Рисунок 1.3 – Foxtrot [3]

Алло (рис. 1.4) – спеціалізується на реалізації мобільних пристроїв, гаджетів та різноманітних аксесуарів. Окрім того, функціонує як маркетплейс із широким вибором товарів та оперативною доставкою.

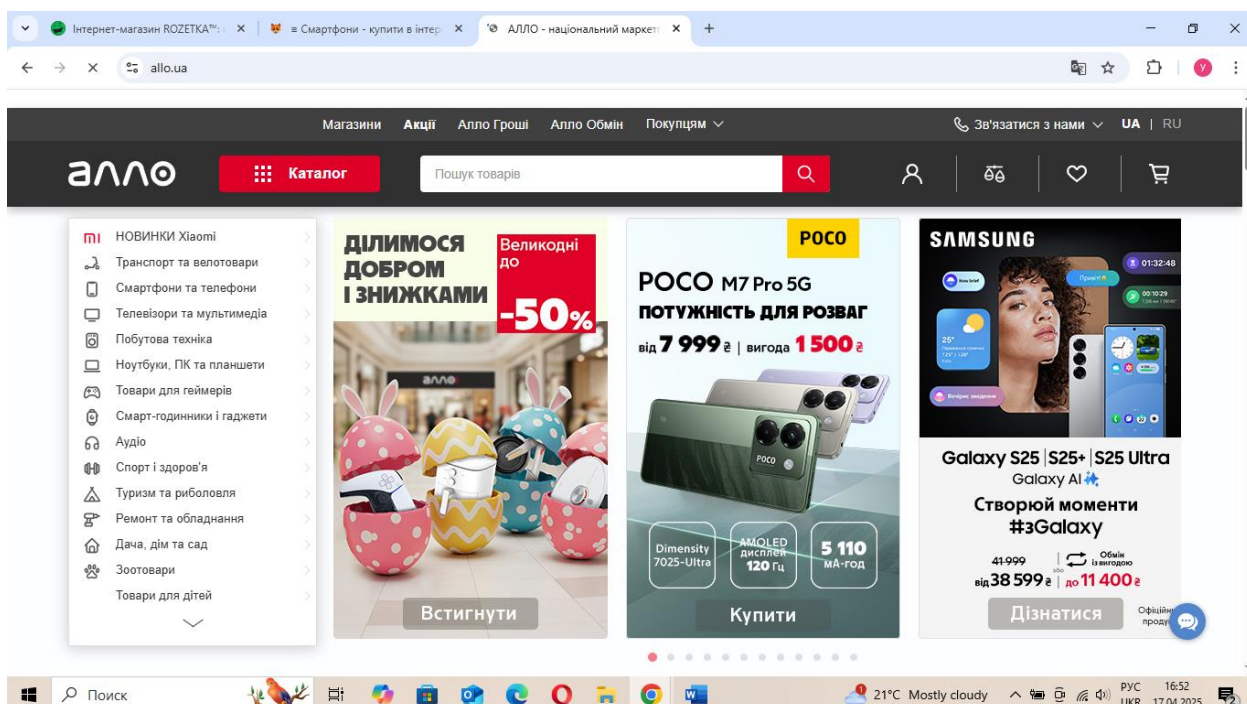


Рисунок 1.4 – Allo [4]

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

КВРІСТ.2101030.21.01.03 ПЗ

Арк.
9

Еpicentrk (рис. 1.5) – Онлайн-платформа відомої мережі гіпермаркетів, де представлено великий вибір товарів для дому, саду, будівництва, меблів та декору.

Розвинена ІТ-інфраструктура є критично важливим елементом для забезпечення стабільної та ефективної роботи онлайн-платформи та всіх пов'язаних сервісів компанії Еpicentrk.ua. Постійна модернізація цієї інфраструктури свідчить про стратегічний підхід компанії до технологічного розвитку, спрямований на підтримку зростаючих потреб бізнесу та забезпечення високого рівня обслуговування клієнтів.

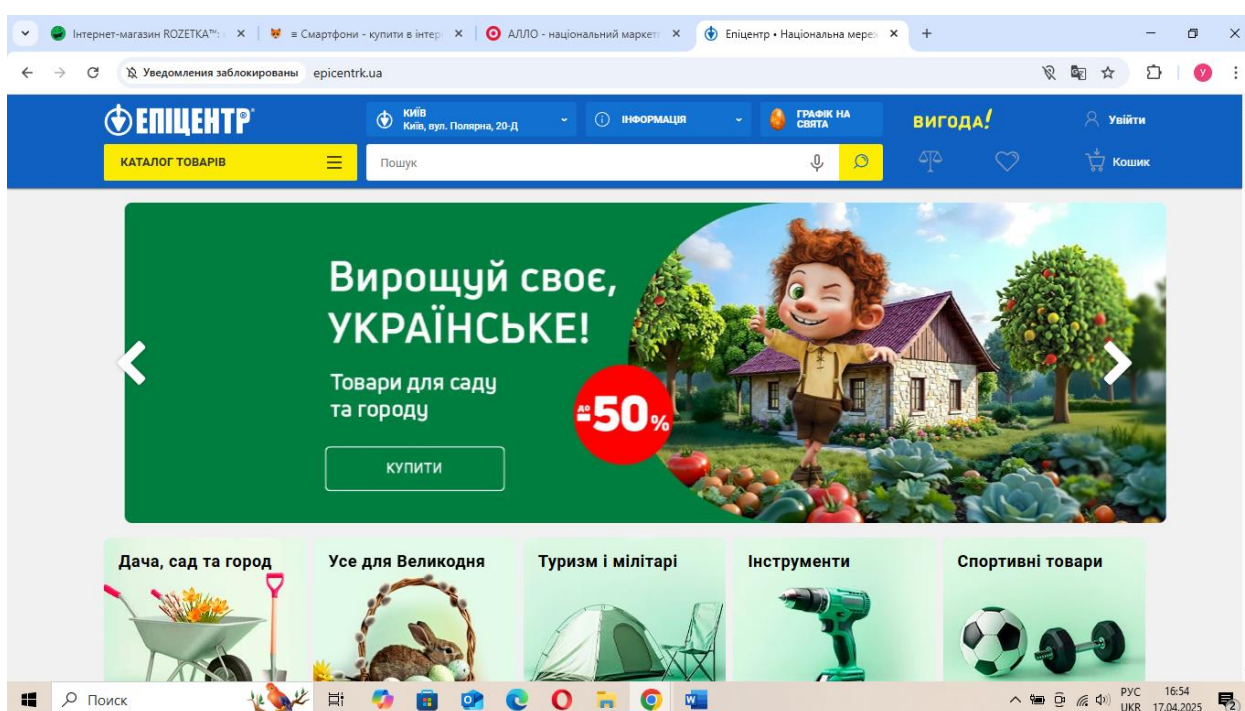


Рисунок 1.5 – Еpicentrk [5]

Світові інтернет-магазини США Amazon (рис. 1.6) – найбільша у світі платформа електронної комерції з величезним асортиментом товарів. Відомий швидкою доставкою, системою рейтингу товарів та інноваційними технологіями.

Німеччина Zalando (рис. 1.8) – європейський лідер з продажу модного одягу та аксесуарів. З безкоштовною доставкою та поверненням товарів. Однією з ключових особливостей Zalando є безкоштовна доставка та повернення товарів, що значно підвищує зручність для покупців та сприяє зростанню їхньої лояльності. Ця політика усуває один з основних бар'єрів для онлайн-покупок одягу та взуття, дозволяючи клієнтам приміряти товари вдома та без зайвих витрат повернути ті, що не підійшли. Таким чином, Zalando.com є яскравим прикладом успішної європейської компанії в сфері електронної комерції моди, яка зробила ставку на зручність для клієнтів, широку пропозицію товарів та постійний розвиток своєї технологічної та логістичної інфраструктури.

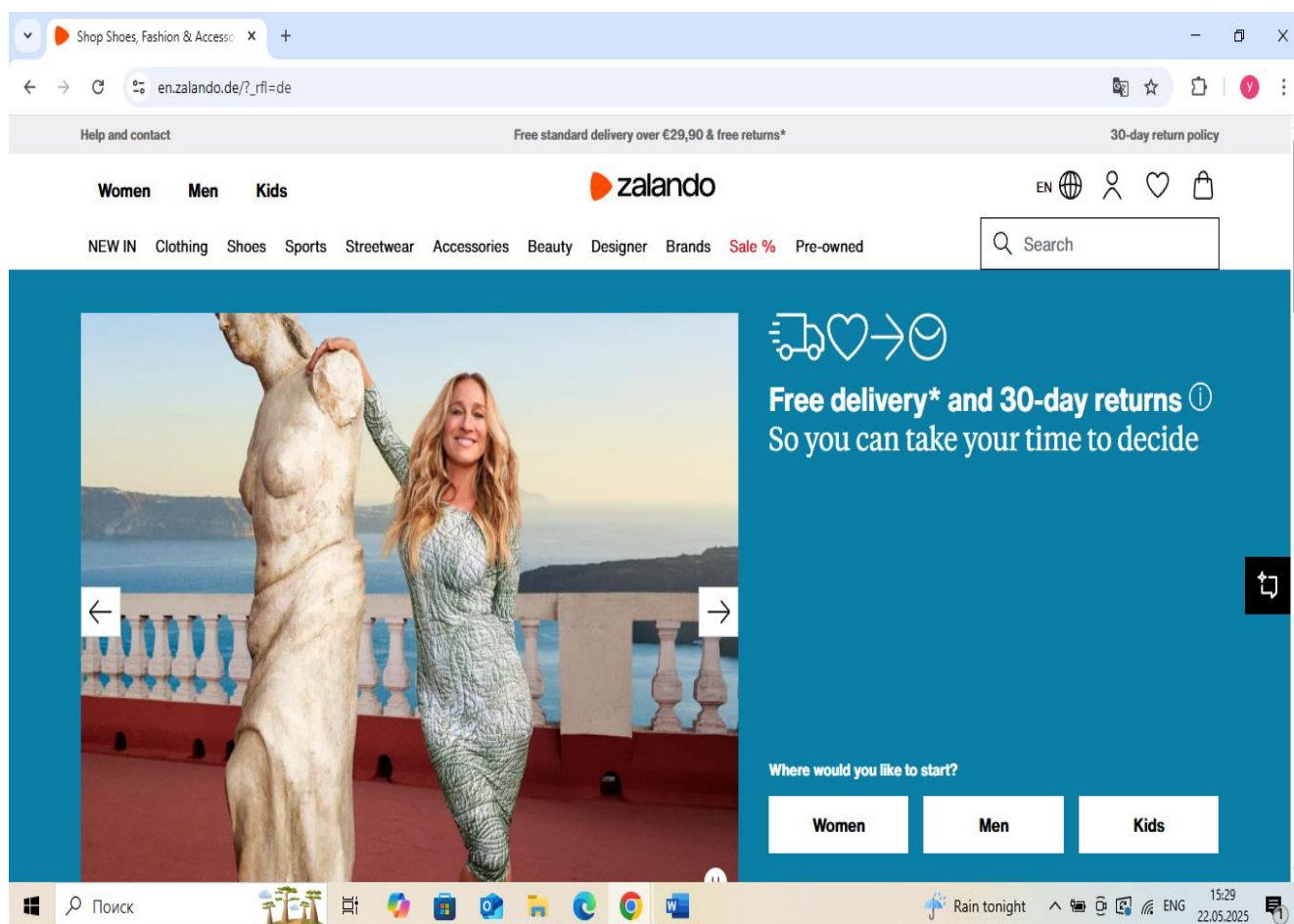


Рисунок 1.8 – Zalando [8]

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 12 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Проаналізувавши провідні українські та міжнародні онлайн-магазини, щоб виявити ключові тенденції та характеристики успішної електронної комерції, зокрема в сегменті електронних товарів можемо зробити висновок.

Українські приклади включають Rozetka та Allo, що демонструють ефективність моделі маркетплейсу з широким вибором товарів, Foxtrot, що спеціалізується на побутовій техніці та електроніці та робить акцент на програмах лояльності, та Epicentrk, що підкреслює важливість розвиненої IT-інфраструктури та поєднує онлайн-продажі з фізичними точками видачі.

Міжнародні аналоги представлені Amazon, який є еталоном глобальної електронної комерції з величезним асортиментом та інноваціями, AliExpress, що пропонує низькі ціни завдяки прямим продажам, та Zalando, що робить акцент на зручності клієнтів, зокрема безкоштовній доставці та поверненню.

Аналіз підкреслює важливість широкого асортименту, зручного інтерфейсу, ефективної системи пошуку, різноманітних варіантів доставки та оплати, якісної підтримки клієнтів, надійної IT-інфраструктури, інновацій та програм лояльності для успіху онлайн-магазину.

Отримані знання будуть використані для розробки конкурентоспроможної та клієнтоорієнтованої платформи інтернет-магазину електронних товарів.

1.3 Аналітика онлайн продажів

Загальна тенденція розвитку електронної комерції в Україні протягом останнього п'ятиріччя демонструє стійке зростання обсягів онлайн-продажів, хоча конкретні річні показники можуть варіюватися. Особливо інтенсивний приріст спостерігався у період пандемії COVID-19, що зумовило значну переорієнтацію споживачів на здійснення покупок через інтернет. У 2024 році зафіксовано збільшення обсягу продажів через мобільні пристрої більш ніж на 25% у порівнянні з попереднім роком (рис. 1.9).

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 13 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

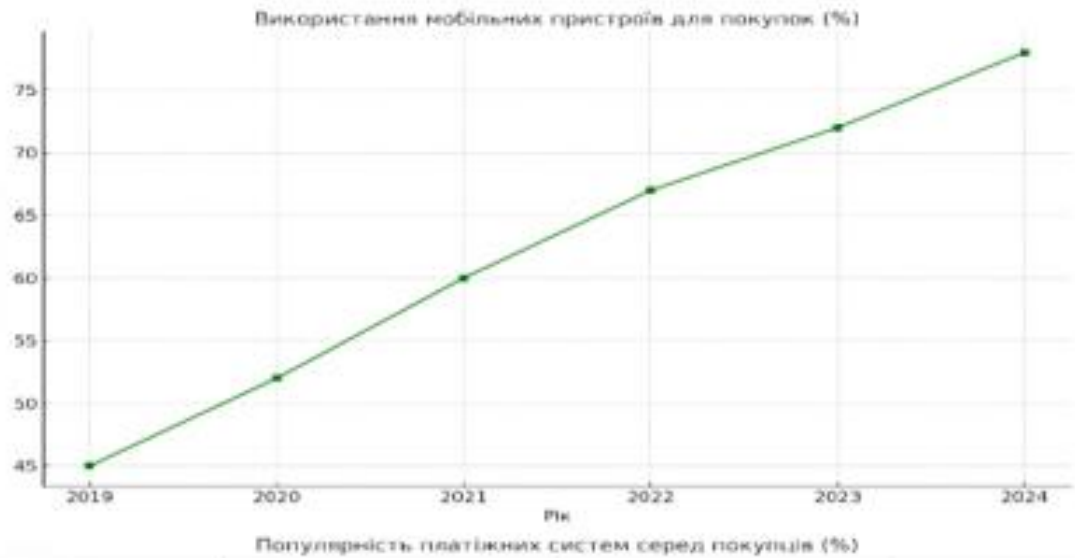


Рисунок 1.9 – Статистика використання мобільних пристроїв для покупки

У 2024 році кількість покупок, здійснених з використанням мобільних пристроїв в Україні, зросла на понад 25%. Середній розмір замовлення, оформленого через мобільні пристрої, сягнув \$33, що на 17% перевищує аналогічний показник попереднього року. Найбільш затребуваними категоріями товарів при покупках зі смартфонів є: електроніка 23%, мода 19%, а також товари для краси та здоров'я 15%. На рисунку 1.10 зображено динаміку онлайн продажів.



Рисунок 1.10 – Обсяг онлайн продажів в Україні у мільярдах гривень

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

Згідно з результатами дослідження Mastercard Digitalization Trust Survey 2019 (рис. 1.11), 87% українських споживачів віддають перевагу здійсненню оплат за допомогою смартфонів. Серед різноманітних способів оплати в інтернет-магазинах найбільшу популярність серед покупців мають онлайн-розрахунки банківськими картками 72%, оплата платіжною карткою при отриманні замовлення 48% та використання мобільних платіжних гаманців 32%.

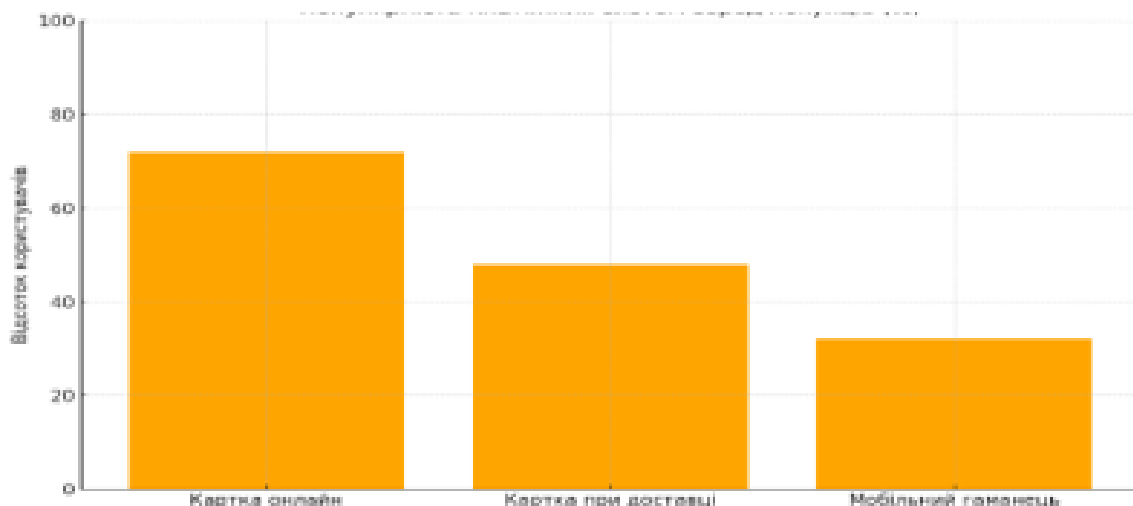


Рисунок 1.11 – Вибір оплати серед користувачів в Україні

1.4 Архітектура та логіка взаємодії користувача в інтернет-магазині

Архітектура інтернет-магазину являє собою впорядковану структуру веб-ресурсу, що визначає траєкторію навігації користувачів між різними сторінками та організацію процесу здійснення покупок. Ретельно розроблена архітектура є запорукою зручної навігації, ефективної взаємодії з відвідувачами та, як наслідок, високої конверсії.

Логіка взаємодії користувача описує послідовність кроків, які проходить потенційний покупець на веб-сайті інтернет-магазину, починаючи з моменту пошуку бажаного товару і завершуючи етапом післяпродажного обслуговування. Глибоке розуміння цієї послідовності дій дозволяє оптимізувати структуру та функціональність сайту відповідно до потреб цільової аудиторії, мінімізувати

кількість незавершених замовлень (покинутих кошиків) та підвищити загальний рівень задоволеності клієнтів, перегляну основні етапи логіки взаємодії користувача.

Користувач заходить на сайт з конкретною потребою, наприклад, знайти певний тип смартфона або телевізор. Для задоволення цієї потреби він використовує пошукову панель, вводячи відповідний запит, або переглядає доступні категорії товарів. Якість функціоналу пошуку, включаючи автозаповнення та надання релевантних підказок, відіграє важливу роль у залученні та утриманні уваги користувача. Після переходу до каталогу товарів користувач застосовує різноманітні фільтри, щоб звужити діапазон пошуку та швидше знайти потрібний варіант. Зручність та швидкість роботи інструментів фільтрації є одним із ключових факторів ефективної навігації по сайту. На цьому етапі користувач детально вивчає картки окремих товарів, порівнює їхні характеристики, аналізує візуальний контент та ознайомлюється з відгуками інших покупців. Забезпечення привабливого візуального представлення товарів та надання вичерпного і зрозумілого опису є критично важливим для прийняття рішення про покупку. Якщо обраний товар відповідає вимогам користувача, він додає його до віртуального кошика. Потім користувач переходить до процесу оформлення замовлення, який включає вибір способу доставки, методу оплати та введення необхідної контактної інформації.

1.5 Аналіз основних вимог та постановка задачі

Аналіз основних вимог до створення інтернет-магазину, створення успішного інтернет-магазину вимагає ретельного аналізу та врахування широкого спектру вимог, що охоплюють як технічні аспекти, так і потреби бізнесу та кінцевих користувачів. Ці вимоги можна умовно поділити на кілька ключових категорій які і є нашою задачею.

Завданням роботи є забезпечення інструментів для ефективного керування асортиментом, включаючи додавання нових позицій, редагування існуючих,

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 16 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

видалення застарілих, завантаження якісних візуальних матеріалів створення детальних описів з характеристиками, класифікацію товарів за категоріями, встановлення цін, надання знижок та проведення акцій, відстеження наявності на складі, а також надання користувачам можливості порівнювати товари та здійснювати пошук за різними критеріями. Забезпечення високої швидкості завантаження веб-сторінок, стабільної роботи навіть при значному навантаженні, а також оптимальної роботи на різних типах пристроїв (настільні комп'ютери, планшети, смартфони).

Створення інтуїтивно зрозумілого та легкого у навігації інтерфейсу, забезпечення адаптивного дизайну. Забезпечення стабільної роботи інтернет-магазину без збоїв та помилок, впровадження системи резервного копіювання даних та плану відновлення після можливих аварій. Розробка системи з чітким та добре задокументованим кодом, що полегшує подальше внесення змін, оновлень та технічну підтримку.

Проведення аналізу діяльності основних конкурентів для виявлення їхніх сильних та слабких сторін та визначення власних конкурентних переваг та встановлення часових рамок для запуску проєкту.

1.6 Висновки до першого розділу

У межах розділу 1 було проведено аналіз сфери електронної комерції. Визначено ключові цілі майбутньої системи такі як ефективне адміністрування каталогу, обробка замовлень, управління клієнтами та аналіз даних. Проаналізовано досвід провідних українських та міжнародних онлайн-магазинів з метою виявлення факторів, що сприяють їхньому успіху. Проаналізовано тенденції ринку онлайн-продажів в Україні та важливість зручності для користувачів. На основі проведеного аналізу сформульовано основні вимоги та визначено завдання для створення ефективної та конкурентоспроможної веборієнтованої інформаційної системи для інтернет-магазину електронних товарів.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 17 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

2 ВИБІР ТЕХНОЛОГІЙ ТА СТРУКТУРНО-ФУНКЦІОНАЛЬНЕ МОДЕЛЮВАННЯ

2.1 Структурно-функціональне моделювання інтернет-магазину електронних товарів

Для здійснення структурно-функціонального моделювання діяльності інтернет-магазину, що спеціалізується на електронних товарах, необхідно розробити контекстну діаграму (рис. 2.1). Ця діаграма повинна відображати ключові вхідні та вихідні дані, елементи управління та механізми, задіяні в процесі «Онлайн продаж електронних товарів». Декомпозиція цієї діаграми сприятиме підвищенню продуктивності та ефективності системи шляхом ідентифікації та усунення можливих проблемних зон.

Вхідні дані:

- намір придбати електронний товар;
- перегляд товарного каталогу;
- додавання товарів у кошик;
- внесення даних для оформлення замовлення;
- вибір способу доставки;
- вибір способу оплати;
- підтвердження замовлення;
- запит на інформацію про товар.

Вихідні дані:

- прибуток;
- електронний товар;
- підтвердження здійснення замовлення;
- інформація про товар.

Управління:

- законодавство України «Про захист прав споживачів»;
- умови, що стосуються доставки та оплати;

- інформація про акційні пропозиції та знижки.

Механізми:

- інтернет-магазин;
- вебсервер;
- система управління базою даних;
- адміністратор інтернет-магазину;
- покупець.

З метою підвищення продуктивності та ефективності, блок «Онлайн продаж електронних товарів» може бути розкладений на такі основні підпроцеси:

- відображення каталогу та функціональність пошуку товарів;
- управління кошиком покупця;
- оформлення замовлення;
- обробка замовлення адміністратором;
- оплата замовлення;
- доставка замовлення.

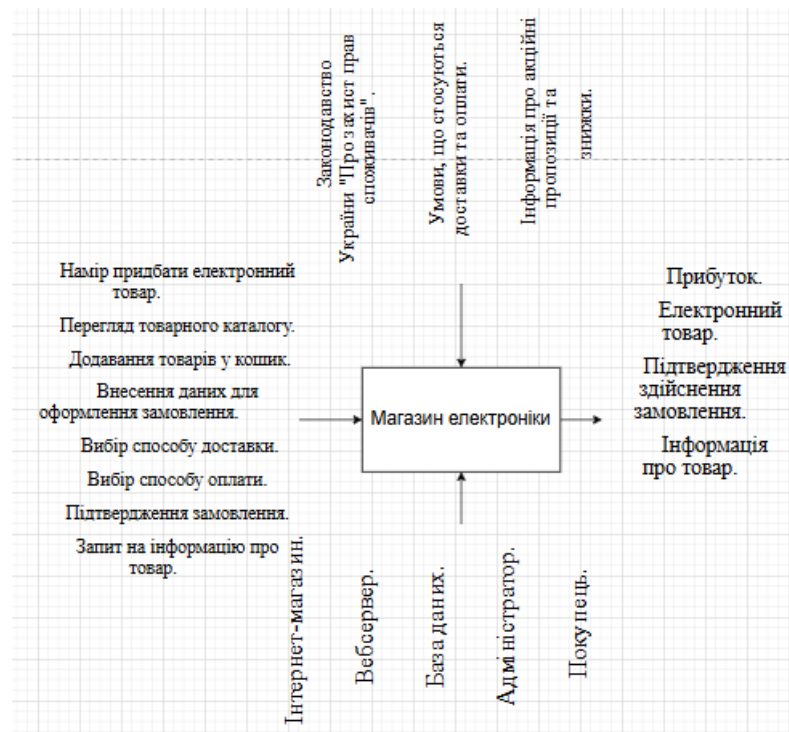


Рисунок 2.1 – Контекстна діаграма діяльності майбутнього інтернет-магазину електроніки

Подальшим етапом декомпозиції контекстної діаграми є розробка функціональної моделі. З метою спрощення розробки було прийнято рішення про її поділ на три окремі функції. Опис зазначених функцій представлено у вигляді окремих функціональних блоків (рис .2.2).

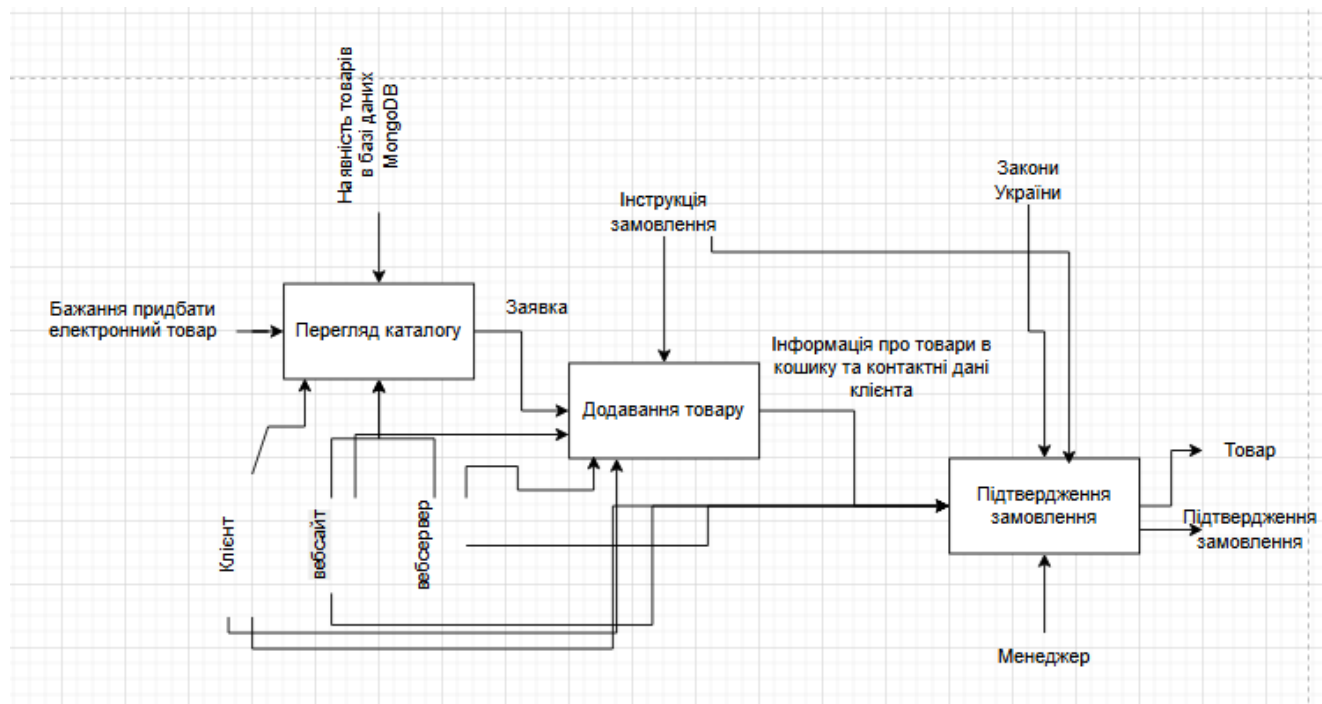


Рисунок 2.2 – Декомпозиція контекстної діаграми діяльності майбутнього інтернет-магазину електроніки

Процес онлайн-продажу в інтернет-магазині електроніки включає три ключові стадії такі як перегляд каталогу, додавання товару в кошик та підтвердження замовлення. На стадії перегляду каталогу покупець взаємодіє з веб-сайтом та сервером для перегляду товарів з бази даних, маючи намір придбати товар. На етапі додавання товару в кошик покупець обирає товар, веб-сайт передає запит на сервер для перевірки наявності та додавання інформації в кошик, а результатом є інформація про товари в кошику та дані покупця. На стадії підтвердження замовлення покупець вводить дані доставки/оплати, сервер перевіряє їх, розраховує вартість та реєструє замовлення, а результатом є прибуток та підготовлений до відправки товар, при цьому покупець отримує підтвердження.

Хедер є верхньою частиною кожної сторінки веб-сайту та відповідає за навігацію, ідентифікацію магазину та надання швидкого доступу до ключових функцій. Для інтернет-магазину електронних товарів хедер містить такі елементи як меню навігації та кошик .

Футер розташовується в нижній частині кожної сторінки та містить важливу інформацію, яка не є першочерговою для процесу покупки, але є корисною для користувачів. Для інтернет-магазину електронних товарів футер буде містити інформацію про магазин контактну інформацію та посилання на інші ресурси.

Структура меню онлайн магазину з продажу електронних товарів наведена на рисунку 2.3.

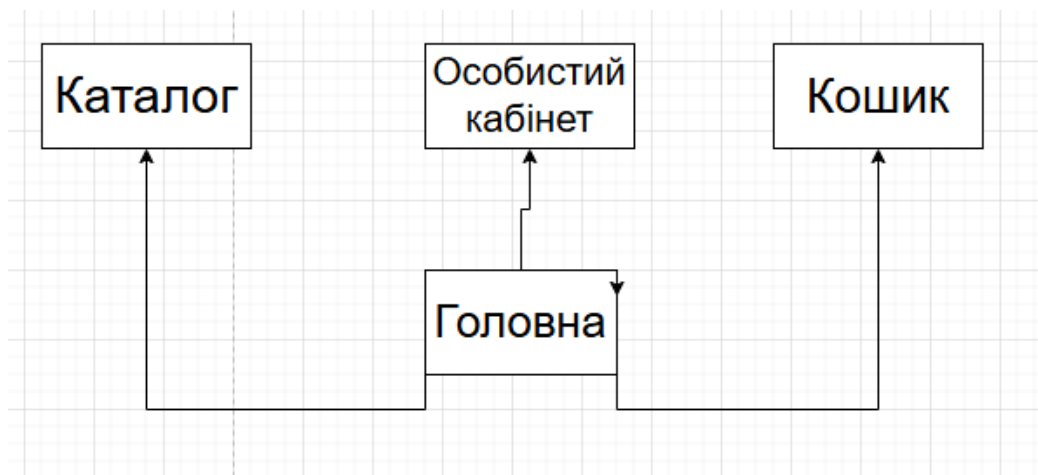


Рисунок 2.3 – Меню магазину

Онлайн магазин з продажу електроніки повинен мати такі сторінки:

- головна сторінка;
- каталог;
- кошик;
- особистий кабінет;
- вхід та реєстрація;
- оформлення замовлення;
- адмін сторінка.

Головна сторінка інтернет-магазину електронних товарів є відправною точкою для користувачів та має на меті надати загальне уявлення про асортимент, ключові пропозиції та полегшити навігацію до основних розділів також містить банер часто містить банери.

Сторінка каталогу є ключовим розділом онлайн-магазину, де користувачі можуть переглядати повний спектр доступних електронних товарів. Її головна функція полягає у забезпеченні зручного інтерфейсу для огляду, фільтрування та вибору необхідних товарів містить всі дані про товар фото ,ціну та характеристики.

Сторінка особистого кабінету розроблена для авторизованих користувачів інтернет-магазину. Вона надає можливість доступу до особистих даних та історії взаємодії з магазином.

Перед тим, як перейти до оформлення замовлення, користувачі потрапляють на сторінку кошик, де вони мають можливість переглянути вибрані товари, відредагувати їх кількість переглянути ціну і ухвалити рішення про придбання.

Сторінка для формлення замовлення є останнім кроком перед підтвердженням придбання, де користувач вводить необхідні дані для організації доставки та здійснення оплати вибраних товарів.

Сторінка входу та реєстрації призначена як для нових, так і для вже зареєстрованих користувачів інтернет-магазину. Вона поділяється на дві основні частини розділ «Вхід» для авторизації існуючих користувачів та розділ «Реєстрація» для створення нового облікового запису.

Сторінка адміністратора надає інструменти для управління товарним каталогом, включаючи додавання нового товару із зазначенням назви, ціни, опису та завантаженням зображень, редагування існуючих товарів з можливістю оновлення будь-яких їхніх даних та видалення товарів з каталогу з підтвердженням дії.

Структурна схема клієнтської частини представлена на рисунку 2.4.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 22 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

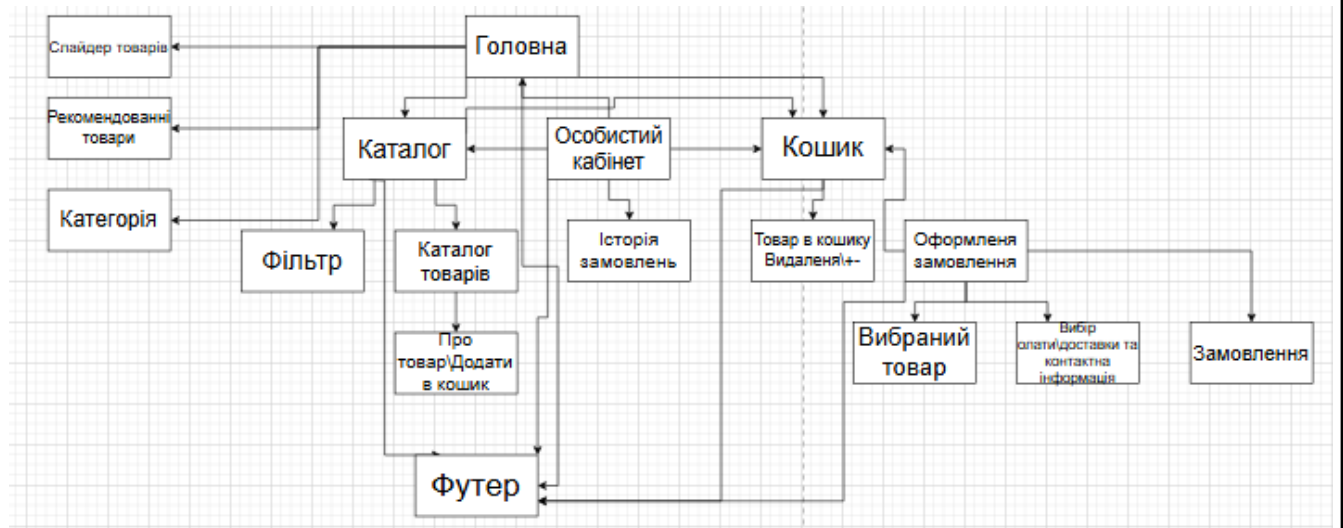


Рисунок 2.4 – Навігація клієнтської частини проекту

Веборієнтована інформаційна система з продажу електроніки повинна дозволити клієнту виконувати наступні дії:

- переглядати список усіх доступних товарів;
- фільтрувати товари за різними критеріями та параметрами;
- переглядати зображення товару та детальну інформацію про кожен товар;
- додавати обрані товари до кошика;
- переглядати вміст кошика;
- змінювати кількість одиниць кожного товару в кошику;
- видаляти товари з кошика;
- переглядати проміжну суму замовлення;
- вводити необхідну контактну інформацію;
- вводити адресу доставки;
- обирати спосіб доставки;
- обирати спосіб оплати;
- переглядати підсумкову інформацію про замовлення;
- підтверджувати замовлення;
- реєструватися в системі;
- авторизуватися в системі;
- переглядати історію своїх замовлень.

Для адміністратора інтернет-магазину повинні бути доступні наступні дії:

- введення назви аксесуара;
- завантаження зображень;
- введення детального опису;
- вказівка ціни;
- вибір категорії;
- зміна будь-якої інформації про товар;
- видалення окремих товарів.

При перегляді каталогу аксесуарів, процес замовлення, отримання додаткової інформації та зв'язок з оператором інтернет-магазину має бути інтуїтивно зрозумілим для клієнта.

Процес ведення обліку товарів, замовлень та перегляд детальної інформації не повинен створювати труднощів для адміністратора, що сприятиме прискоренню обробки заявки.

2.2 Вибір редактора коду

Visual Studio Code (VS Code) – це безкоштовний, розроблений Microsoft та надзвичайно популярний редактор коду, що вирізняється своєю легкістю та швидкістю роботи, забезпечуючи швидкий запуск та економне використання системних ресурсів. Він пропонує інтуїтивно зрозумілий та гнучко налаштовуваний інтерфейс, а також інструменти для редагування коду, включаючи підсвічування синтаксису для широкого спектра мов програмування (HTML, CSS, JavaScript, PHP та інші), функцію автозавершення коду, що сприяє прискоренню розробки та зменшенню кількості помилок, підтримку Emmet для швидкої генерації HTML-структур та можливості рефакторингу коду для безпечного перейменування змінних і функцій.

Однією з ключових переваг VS Code (рис. 2.5) є його розгалужена екосистема розширень, які значно розширюють базову функціональність редактора, надаючи підтримку для роботи з Git, інструменти для налагодження коду, інтеграцію з

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 24 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

конкретними мовами програмування та базами даних. Редактор також включає вбудований термінал, що дозволяє запускати команди безпосередньо з робочого середовища, та інтегровану підтримку Git для зручної взаємодії з системою контролю версій.

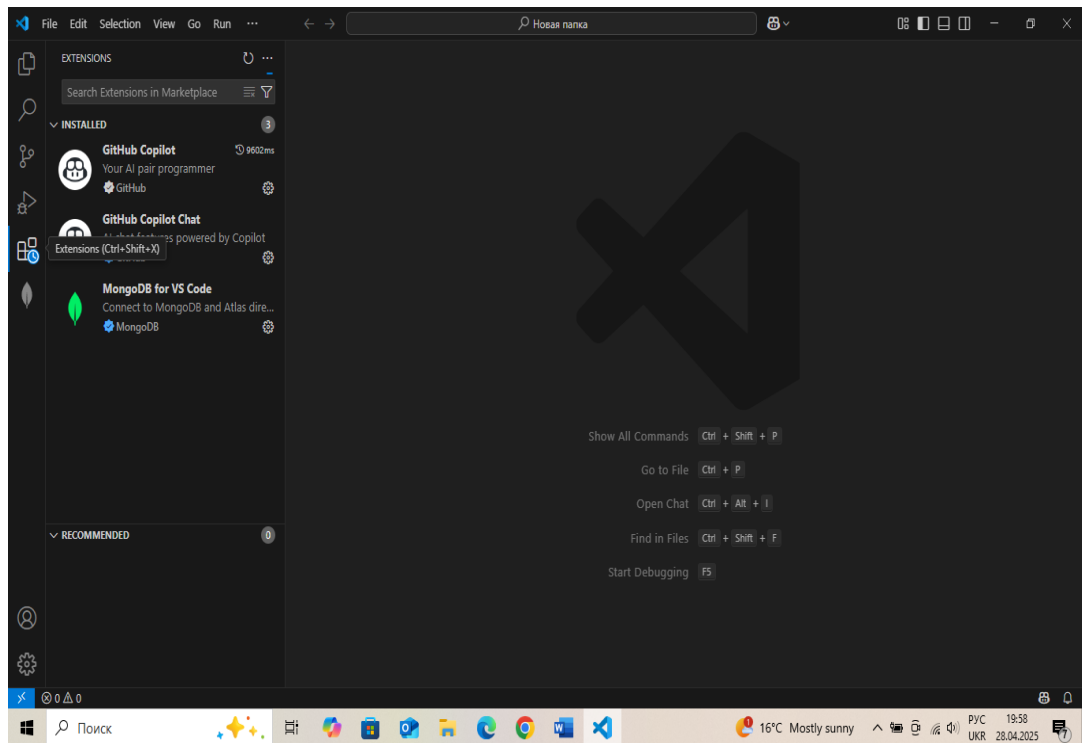


Рисунок 2.5 – Середовище VSC

VS Code є кросплатформним інструментом, що забезпечує його безперерйну роботу на операційних системах Windows, macOS та Linux. Активна спільнота розробників постійно працює над підтримкою редактора, розробкою нових розширень та наданням допомоги користувачам. VS Code є особливо цінним інструментом для розробки фронтенду (з використанням HTML, CSS, JavaScript), бекенду (наприклад, на PHP), редагування конфігураційних файлів, налагодження JavaScript-коду та ефективної роботи з Git.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 25 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

2.3 Вибір мови програмування

HTML (HyperText Markup Language) CSS (Cascading Style Sheets) JS(JavaScript). Ці технології є фундаментом будь-якого сучасного веб-сайту, зокрема й інтернет-магазину. Вони відповідають за структуру, візуальне оформлення та інтерактивність клієнтської частини, або фронтенду.

HTML використовувався для створення семантичної структури кожної сторінки. Це включає розмітку заголовків, параграфів, списків, навігаційних елементів, форм для взаємодії з користувачем, а також відображення товарних карток. Завдяки використанню HTML5, забезпечено широку сумісність з різними веб-браузерами та підтримку нових структурних елементів, що покращило як організацію, так і доступність контенту. HTML є загальноприйнятим стандартом для розмітки веб-сторінок, а HTML5 значно підвищив рівень їхньої структурованості та доступності для всіх користувачів.

CSS відповідав за візуальне представлення веб-сайту. Він визначав кольорову гаму, використовувані шрифти, розміри елементів, відступи, загальні макети сторінок та стилі для різних елементів інтерфейсу. Застосування CSS дозволило створити привабливий та зручний для користувача інтерфейс, який також є адаптованим для коректного відображення на різних типах пристроїв завдяки використанню медіа-запитів. Крім того, CSS використовувався для реалізації плавної анімації та візуальних переходів між елементами. Однією з ключових переваг CSS є відокремлення структури сторінки (заданої HTML) від її візуального оформлення, що значно спрощує подальшу підтримку та модифікацію дизайну. Використання CSS3, зокрема таких можливостей як Flexbox та Grid, дало змогу створювати складні та водночас гнучкі макети сторінок, а адаптивний дизайн забезпечив зручність перегляду контенту на екранах різних розмірів.

JavaScript відповідав за інтерактивність та динамічність інтернет-магазину. За його допомогою обробляються різноманітні події, що ініціюються користувачем (наприклад, кліки миші), здійснюється валідація введених у форми даних, виконуються асинхронні запити до сервера для отримання необхідної інформації

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 26 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

без повного перезавантаження сторінки, реалізується маніпулювання структурою веб-сторінки DOM., JavaScript значно покращив зручність користування сайтом завдяки таким функціям, як автозаповнення полів, надання контекстних підказок та обробка помилок введення даних. JavaScript є невід'ємною частиною сучасних веб-додатків, дозволяючи створювати багаті та інтерактивні інтерфейси. Застосування асинхронних запитів (AJAX) забезпечило високу швидкість реагування та чуйність інтерфейсу.

2.4 База даних та серверна частина

NoSQL (Not Only SQL) бази даних являють собою різноманітний клас систем управління базами даних, що відрізняються від традиційних реляційних баз даних. Ключовою відмінністю є відмова від організації даних у вигляді таблиць з жорстко визначеними схемами. Натомість, NoSQL бази даних (рис. 2.6) використовують широкий спектр моделей даних для ефективного зберігання та швидкого отримання інформації, серед яких:

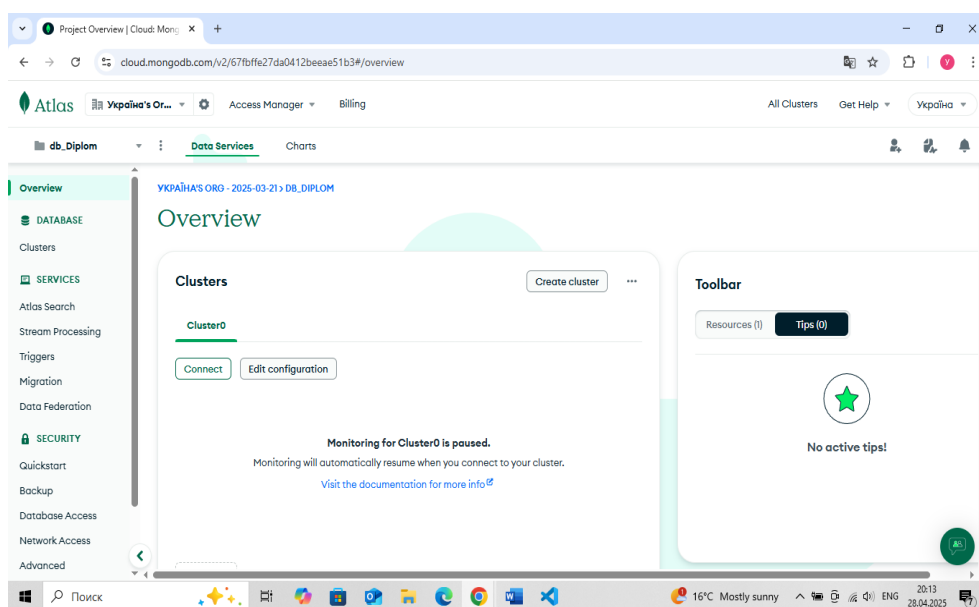


Рисунок 2.6 – NoSQL база даних mongodb [9]

MongoDB – це NoSQL база даних, яку використовуватимемо для зберігання всієї необхідної інформації для функціонування інтернет-магазину. Вона надійно зберігає дані про товари (включаючи назву, детальний опис, ціну, зображення та категорію), інформацію про користувачів (їхні профілі та історію замовлень), дані про самі замовлення (склад замовлення, поточний статус та адреси доставки), інформацію про кошики покупців та перелік категорій товарів.

Однією з ключових переваг MongoDB є гнучкість зберігання даних, що реалізується за допомогою JSON-подібних документів. Це дозволяє легко додавати нові поля до документів та змінювати структуру даних без необхідності попереднього визначення жорсткої схеми бази даних. Крім того, MongoDB забезпечує високу масштабованість, що є критично важливим для великих інтернет-магазинів з великим обсягом даних та високою відвідуваністю.

Вибір NoSQL (конкретно MongoDB) замість SQL виявився більш доцільним для проєкту з наступних причин. Структура даних товарів та замовлень в інтернет-магазині може часто змінюватися залежно від нових вимог або додавання нових функцій. Гнучка структура документів MongoDB ідеально підходить для таких динамічних даних до того ж його можна підключити відразу до VSC (рис. 2.7).

Для забезпечення стабільної роботи інтернет-магазину при зростанні обсягів даних та кількості користувачів необхідна можливість легкого горизонтального масштабування, яку добре підтримує MongoDB. Відсутність необхідності попереднього визначення жорсткої схеми в MongoDB значно прискорює процес розробки, оскільки розробники можуть швидше адаптуватися до змін та додавати нові функції. Таким чином, використання MongoDB забезпечує надійне, ефективне та гнучке зберігання даних, що є критично важливим для успішного функціонування та подальшого розвитку інтернет-магазину.

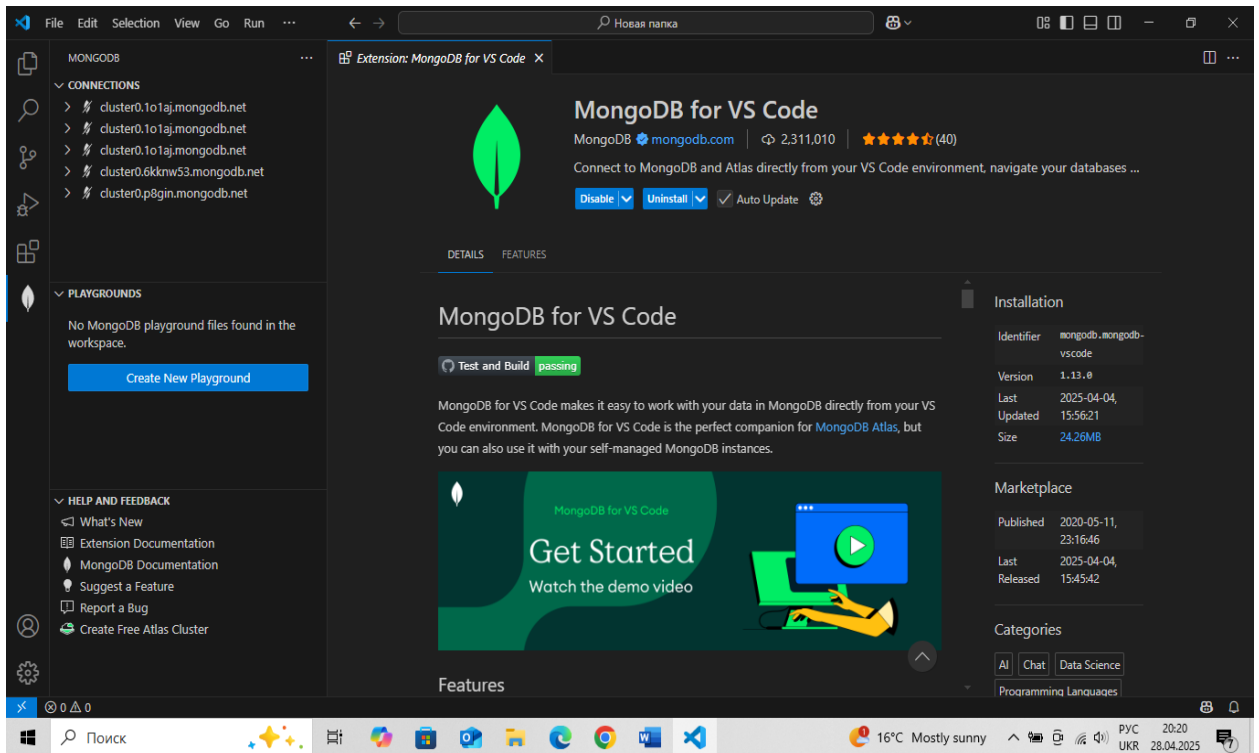


Рисунок 2.7 – Mongoddb в VSC

Для створення серверної частини (бекенду) інтернет-магазину було обрано Node.js як середовище виконання JavaScript на сервері та Express.js як мінімалістичний фреймворк для Node.js.

Node.js надає можливість виконувати код JavaScript на серверній стороні. Це стало ключовою перевагою, оскільки дозволило використовувати єдину мову програмування (JavaScript) як для розробки клієнтської частини (фронтенду), код якої виконується у веб-браузері користувача, так і для розробки серверної частини, код якої виконується на сервері. Node.js використовувався для таких важливих завдань:

- обробка вхідних HTTP-запитів від клієнтських додатків (наприклад, коли користувач запитує сторінку з описом товару або додає товар до свого віртуального кошика);
- взаємодія з базою даних MongoDB (здійснення операцій читання та запису даних, що стосуються товарів, користувачів та їхніх замовлень);

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 29 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

- реалізація основної бізнес-логіки інтернет-магазину (включаючи процеси обробки замовлень, автентифікації та авторизації користувачів, а також управління вмістом кошика покупця);

- створення API (інтерфейсу прикладного програмування) для забезпечення обміну даними між фронтендом та бекендом. Створений API визначає чіткі правила, за якими фронтенд може надсилати запити до сервера та отримувати від нього необхідні дані.

Node.js вирізняється своєю неблокуючою, керованою подіями архітектурою. Це означає, що він здатен ефективно обробляти велику кількість одночасних запитів без необхідності очікування завершення кожної окремої операції. Така архітектура робить Node.js надзвичайно продуктивним для розробки застосунків, які потребують обробки значної кількості паралельних з'єднань, що є критично важливим для забезпечення належної роботи інтернет-магазину.

Використання JavaScript як на клієнті, так і на сервері значно спростило процес розробки, оскільки можливо зосередитися на одній мові програмування, уникаючи необхідності контекстного перемикання між різними технологіями.

Node.js має велику та активну спільноту розробників, а також доступ до величезної екосистеми готових до використання модулів (пакетів), розміщених в npm (менеджері пакетів Node). Це значно прискорило процес розробки та дозволило використовувати вже перевірені рішення для багатьох типових завдань.

Express.js є мінімалістичним та водночас надзвичайно гнучким веб-фреймворком для Node.js. Він надає розробникам набір необхідних інструментів та функцій, які значно спрощують процес створення веб-застосунків та API. У нашому проєкті Express.js використовувався для:

- визначення маршрутів (кінцевих точок API). Маршрут являє собою спосіб відображення вхідних HTTP-запитів на певні функції обробки на сервері (наприклад, GET-запит на URL-адресу /products може бути оброблений функцією, яка повертає список усіх доступних товарів);

- обробки різних типів HTTP-запитів (таких як GET для отримання даних, POST для надсилання даних на сервер, PUT для оновлення існуючих даних та DELETE для видалення даних). Express.js надає зручні методи для обробки кожного з цих типів запитів;

- керування процесом обробки запитів та формування відповідей. Express.js значно полегшує отримання даних, що передаються в запиті, та надсилання відповідних даних назад клієнту;

- використання проміжного програмного забезпечення являють собою функції, які можуть бути виконані на різних етапах обробки запиту, наприклад, для здійснення автентифікації користувача, керування сесіями, логування вхідних запитів тощо;

- express.js значно спрощує розробку RESTful API, які є основою для обміну даними між сучасними фронтенд та бекенд частинами веб-застосунків. Він надає зручні засоби для визначення маршрутів, обробки запитів та маніпулювання даними.

Express.js є одним з найпопулярніших та добре документованих фреймворків для Node.js. Велика спільнота та наявність вичерпної документації значно полегшують процес розробки, налагодження та подальшої підтримки проєкту.

Фреймворк є надзвичайно гнучким та дозволяє легко інтегрувати різноманітні сторонні модулі та бібліотеки для розширення його функціональності відповідно до потреб проєкту.

Таким чином, ефективна взаємодія Node.js та Express.js забезпечила створення продуктивної серверної частини інтернет-магазину, характеризуючись високою швидкістю роботи, масштабованістю та зручністю.

2.5 Менеджер пакетів, контроль версій, хостинг та хешування

Менеджер пакетів - це програмний інструмент, який автоматизує процес встановлення, оновлення, налаштування та видалення програмних бібліотек та інших необхідних інструментів у межах розробницького проєкту. Їхня ключова

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 31 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

роль полягає у значному спрощенні керування залежностями проєкту та підвищенні загальної ефективності процесу розробки. Замість того, щоб вручну шукати, завантажувати та інтегрувати потрібні бібліотеки, розробники можуть використовувати менеджери пакетів для автоматизації цих рутинних завдань..

NPM (Node Package Manager) є стандартним та найбільш поширеним менеджером пакетів для екосистеми Node.js. Він являє собою не лише інструмент командного рядка, але й величезний онлайн-реєстр, що містить сотні тисяч готових до використання пакунків (бібліотек, інструментів, фреймворків тощо). Завдяки своїй широкій популярності та великій спільноті, npm став де-факто стандартом у JavaScript-розробці, активно використовуючись як для бекенд-розробки (з Node.js), так і для фронтенд-розробки (для управління залежностями в таких фреймворках, як React, Angular, Vue.js, а також для встановлення різноманітних інструментів збірки та оптимізації)..

Працюючи над проєктом інтернет-магазину, бекенд якого розроблявся на Node.js, основним інструментом для керування залежностями та пакетами було обрано npm, або Node Package Manager.

Git – це розподілена система контролю версій, яка відіграла ключову роль у відстеженні всіх змін у коді проєкту, забезпеченні історії цих змін, можливості повернення до попередніх версій та організації паралельної роботи декількох розробників. Git використовувався для керування всією кодовою базою: кожна значна зміна фіксувалася як окремий коміт, що дозволяло детально відстежувати еволюцію коду та легко виправляти можливі помилки.

GitHub (рис. 2.8) – це веб-платформа, яка надала не лише хостинг для репозиторію Git, але інструменти для спільної розробки. Вона забезпечила графічний інтерфейс для зручної роботи з Git та розширила його функціональність наступними можливостями:

GitHub став централізованим місцем для зберігання копій усього коду проєкту, забезпечуючи таким чином його резервне копіювання, постійну доступність для всіх членів команди та ефективну спільну роботу над проєктом.

Для розміщення статичних файлів фронтенду інтернет-магазину було використано безкоштовну послугу GitHub Pages, яка дозволила легко опублікувати веб-сайт без необхідності налаштування окремого хостингу.

Основні команди Git, які активно використовувалися в процесі розробки, включали `git init` (для ініціалізації нового репозиторію), `git add` (для додавання файлів під контроль версій), `git commit` (для фіксації зроблених змін), `git branch` (для створення та перемикання між гілками) та `git merge` (для об'єднання змін з різних гілок). Завдяки своїй популярності та гнучкості, Git став незамінним інструментом для ефективною розробки.

Для розміщення фронтенд частини інтернет-магазину було використано послугу хостингу GitHub Pages, яка надала зручний та безкоштовний спосіб розміщення статичних веб-сайтів безпосередньо з репозиторію Git. Процес розробки бекенд частини відбувався локально на моєму комп'ютері з використанням Node.js та локального сервера (`localhost`), що є стандартною практикою для веб-розробки.

GitHub Pages – це сервіс хостингу статичних веб-сайтів від GitHub, що дозволяє безкоштовно розміщувати веб-сайти безпосередньо з репозиторіїв. Для цього користувач створює репозиторій, додає файли веб-сайту (HTML, CSS, JavaScript, зображення) та налаштовує GitHub Pages для цього репозиторію, після чого файли стають доступними за згенерованою URL-адресою. GitHub Pages є простим, безкоштовним сервісом з інтеграцією Git, ідеально підходить для статичних сайтів, тому я використав його для фронтенду. Однак, він не підтримує серверні мови (Node.js, PHP, Python), не надає баз даних та можливостей для серверної логіки, необхідних для бекенду інтернет-магазину (обробка замовлень, аутентифікація). Тому для повноцінного розгортання бекенду з Node.js та MongoDB розробка та тестування проводились локально на комп'ютері з використанням Node.js та `localhost`.

Для безпечного зберігання паролів користувачів буде використовуватися бібліотеку `bcrypt`. Хешування – це процес перетворення пароля на незворотний

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 34 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

рядок фіксованої довжини (хеш), що є важливим для безпеки, оскільки замість самих паролів у базі даних зберігаються їхні хеші, що унеможлиблює їх відновлення у випадку злому. bcrypt вважається безпечним завдяки використанню випадкових даних для кожного пароля та адаптивному фактору вартості, що ускладнює злам. У моєму проєкті bcrypt використовувався при реєстрації користувача для хешування введеного пароля перед його збереженням у базі даних MongoDB, а при вході в систему введений пароль знову хешувався та порівнювався зі збереженим хешем для аутентифікації.

2.6 Висновки до другого розділу

У межах розділу 2 було обґрунтовано кожне ключове технологічне рішення, що стало фундаментом для створення інтернет-магазину. Кожен вибір був продиктований специфікою проєкту, вимогами до функціональності, продуктивності, масштабованості та зручності розробки, а також базувався на глибокому розумінні основних веб-технологій.

Для клієнтської частини було обрано HTML, CSS та JavaScript як базові будівельні блоки. Це рішення на початковому етапі надало нам повний контроль над структурою, стилізацією та інтерактивністю користувацького інтерфейсу без необхідності використання додаткових бібліотек або фреймворків.

На серверній стороні перевагу було віддано Node.js завдяки його ефективній обробці великої кількості паралельних запитів та можливості використання єдиної мови програмування (JavaScript) на всьому стеку. Фреймворк Express.js значно спростив процес створення API. Для зберігання даних обрано гнучку та масштабовану NoSQL базу даних MongoDB.

Вибір Visual Studio Code як редактора коду, npm для управління залежностями, Git та GitHub для контролю версій та хостингу статичного фронтенду (через GitHub Pages), а також бібліотеки bcrypt для забезпечення

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 35 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

безпечного хешування паролів, сформував надійну та зручну інфраструктуру для розробки та управління проектом.

Прийняті технологічні рішення, попри фокус на фундаментальних веб-технологіях для фронтенду, у поєднанні з бекендом на Node.js та гнучкою базою даних MongoDB, заклали основу для розробки функціонального та ефективного інтернет-магазину.

Підведення підсумків щодо ролі менеджерів пакетів у сучасній веб-розробці та обґрунтування вибору npm для керування залежностями бекенду інтернет-магазину, враховуючи його екосистему та інтеграцію з Node.js.

Проведене структурно-функціональне моделювання є важливим етапом проєктування веборієнтованої інформаційної системи, оскільки воно дозволяє чітко визначити межі системи, її основні функції, потоки даних та взаємодію між користувачами та адміністраторами. Результати цього моделювання слугуватимуть основою для подальшої розробки архітектури системи, проєктування бази даних та реалізації функціональних модулів, спрямованих на забезпечення ефективного та зручного процесу онлайн-продажу електронних товарів. Визначення потенційних проблемних зон на етапі моделювання сприятиме підвищенню продуктивності та ефективності майбутньої системи. Ці технології та методи розробки забезпечили реалізацію необхідного функціоналу, та дозволили створити систему, яка є гнучкою для майбутніх розширень та адаптацій.

3 РЕАЛІЗАЦІЯ ПРОЄКТУ

3.1 Структура проєкту

Структура (рис. 3.1-3.3) інтернет-магазину базується на класичній трирівневій моделі.

1. Клієнтський рівень (рис. 3.1).

Виконує такі функції як відображення інтерфейсу користувача (каталог, сторінки товарів, кошик, форми, особистий кабінет), забезпечення інтерактивності (обробка дій, валідація форм, динамічне оновлення), взаємодія з бекендом через API (HTTP-запити/відповіді JSON).

```
# admin.css
<> admin.html
JS admin.js
JS auth-routes.js
# auth.css
<> auth.html
JS auth.js
<> cart.html
JS cart.js
<> catalog.html
JS catalog.js
# check.css
JS checkout.js
<> checkout.html
<> index.html
JS index.js
{} products.json
# profile.css
<> profile.html
JS profile.js
JS script.js
# style.css
```

Рисунок 3.1 – Файли ФронЕнду

2. Серверний рівень(рис. 3.2).

Виконує такі функції як прийом/обробка запитів від фронтенду, реалізація бізнес-логіки (замовлення, кошик, аутентифікація), взаємодія з базою даних (отримання або збереження даних), забезпечення безпеки або валідації, надання API для фронтенду.

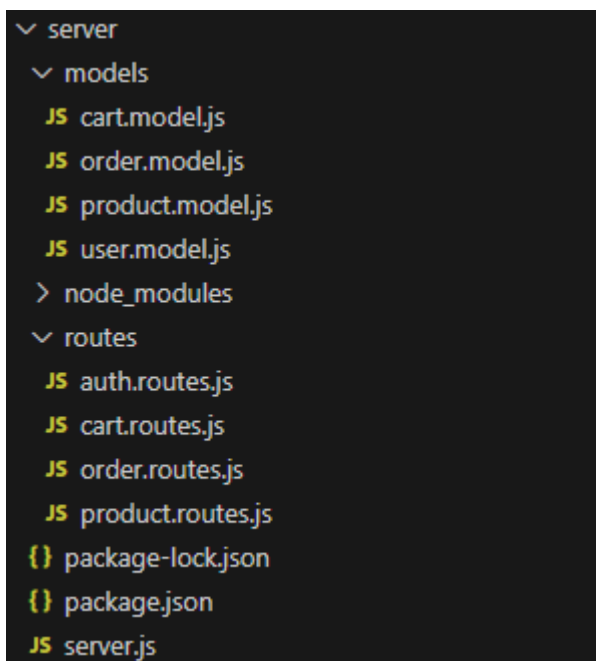


Рисунок 3.2 – Файли Бекенду

3. Рівень даних .

Зберігання даних магазину (товари, користувачі, замовлення, категорії, кошики), забезпечення ефективного доступу для бекенду. Фронтенд надсилає HTTP-запити до бекенду для отримання даних або виконання дій. Бекенд обробляє запити, взаємодіє з базою даних, отримує/зберігає інформацію, формує відповідь у форматі JSON та надсилає її назад на фронтенд для оновлення інтерфейсу.

Додаткові компоненти:

- npm;
- git;
- github;

- bcrypt;
- localhost.

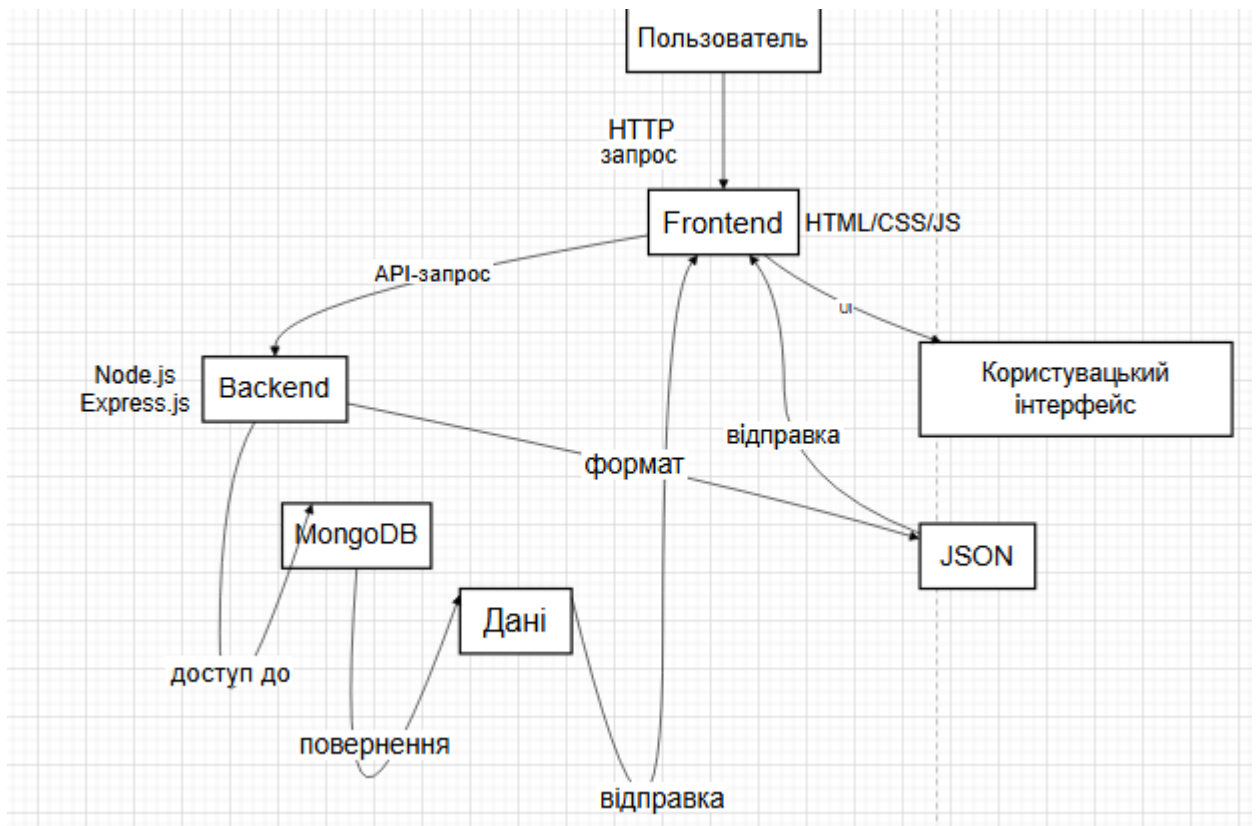


Рисунок 3.3 – Схема проекту

Ця трирівнева архітектура забезпечила чітке розділення відповідальностей, спростивши розробку, тестування та підтримку, а використання RESTful API зробило систему гнучкою та масштабованою.

3.2 Реалізація фронтенду

Фронтенд інтернет-магазину побудовано з використанням трьох основних веб-технологій. HTML для створення структури сторінок, CSS для їхньої стилізації та візуального оформлення, JavaScript для забезпечення інтерактивності та динамічної поведінки.

Для кожної ключової секції магазину були розроблені відповідні HTML-структури, включаючи головну сторінку, сторінку каталогу товарів, детальні

сторінки окремих продуктів, кошик покупця, сторінки реєстрації та входу користувачів, особистий кабінет та сторінку оформлення замовлення. При цьому активно використовувалися семантичні HTML5 теги, що покращило структуру та доступність контенту для пошукових систем та допоміжних технологій.

За візуальне представлення відповідав CSS, за допомогою якого було визначено загальну кольорову схему веб-сайту, обрано типографіку (шрифти), розроблено гнучкі макети сторінок з використанням сучасних CSS-технологій, таких як Flexbox та Grid. Для забезпечення зручного перегляду сайту на різних типах пристроїв (від настільних комп'ютерів до мобільних телефонів) було реалізовано адаптивний дизайн з використанням медіа-запитів. Крім того, CSS застосовувався для стилізації окремих елементів інтерфейсу та додавання базових візуальних анімацій.

Інтерактивність та динамічність фронтенду забезпечувалися за допомогою JavaScript. Він використовувався для обробки різноманітних дій користувача (наприклад, кліків, наведення курсора), валідації даних, що вводяться у форми, реалізації функціональності кошика покупця з використанням локального сховища браузера (localStorage), здійснення асинхронних запитів до API бекенду для отримання та надсилання даних, а також для динамічного оновлення вмісту веб-сторінок та окремих елементів інтерфейсу, таких як модальні вікна, сповіщення користувачів та слайдери зображень.

Загалом, фронтенд інтернет-магазину був розроблений з особливою увагою до зручності використання, швидкості завантаження сторінок та плавності взаємодії з усіма елементами інтерфейсу.

Головна сторінка проєкту, під назвою index.html підключений до style.css в якому додано всі стилі крім останніх сторінок потім почав працювати оскільки обсяг коду (рис. 3.4) становив приблизно 1500 рядків кода.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 40 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

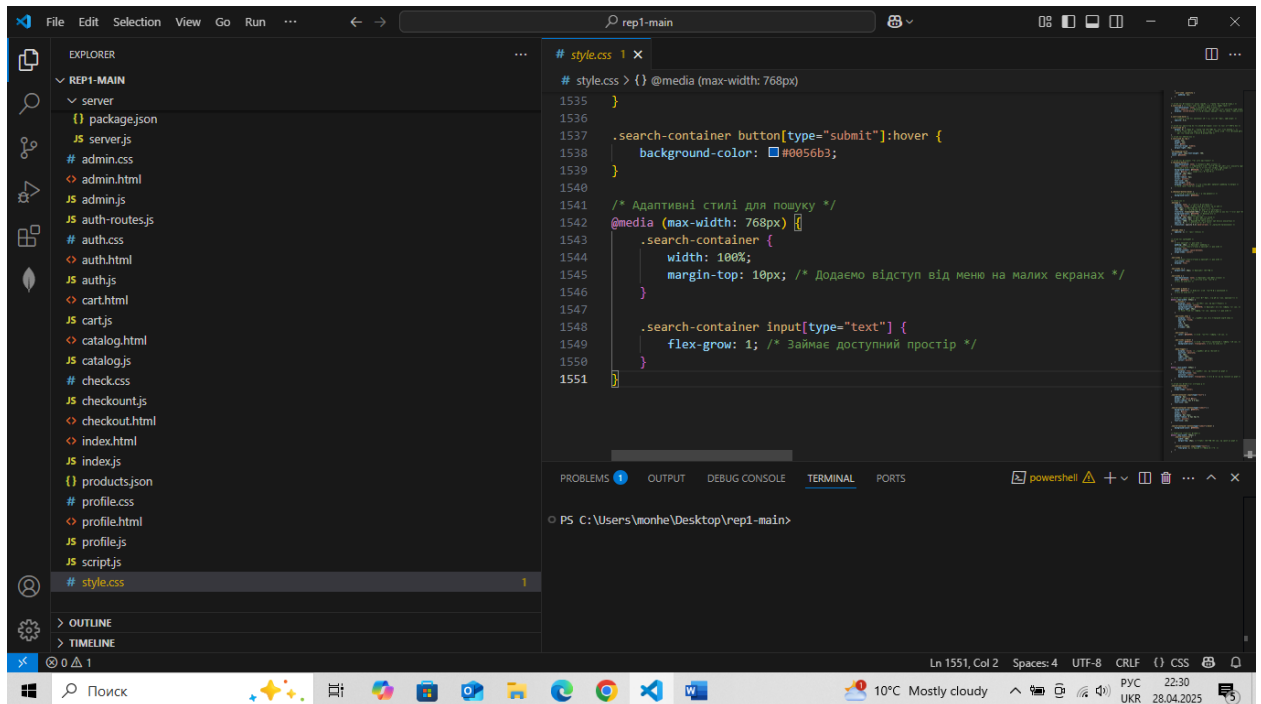


Рисунок 3.4 – Розмір файлу style.css

Головна сторінка «Магазину Електроніки» розроблена з акцентом на швидке ознайомлення користувача з ключовими аспектами асортименту та забезпечення зручної навігації.

У верхній частині сторінки (рис. 3.5) розміщено шапку, яка містить логотип магазину для миттєвої ідентифікації бренду, основну навігаційну панель з посиланнями на ключові розділи сайту («Головна», «Каталог», «Особистий кабінет») та інформацію про кошик покупця, де наразі знаходиться один обраний товар.

Безпосередньо під шапкою розташовано великий банер із зображенням ноутбука, що, ймовірно, є частиною акційної пропозиції, призначеної привернути увагу відвідувачів до вигідних умов придбання.

Далі розміщено блок рекомендованих товарів, позначений відповідним заголовком, який представляє три ключові товарні позиції. Кожен товар у цьому блоці супроводжується назвою, актуальною ціною, а також двома важливими кнопками, детальніше для отримання повної інформації про продукт та додати в кошик для швидкого додавання товару до замовлення.

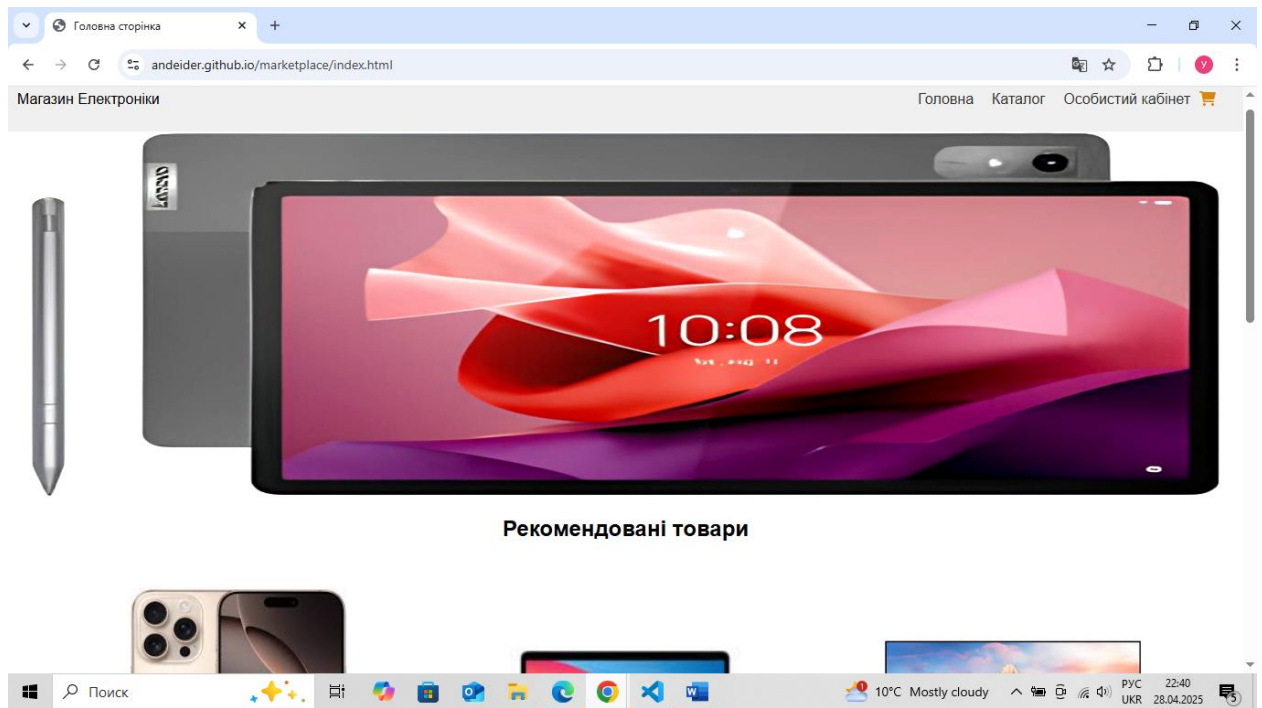


Рисунок 3.5 – Верхня частина головного екрану

Нижче (рис. 3.6) розташовано блок категорій товарів, представлений у вигляді чотирьох візуально привабливих плиток. Кожна плитка містить назву основної категорії (таких як «Смартфони», «Ноутбуки», «Телевізори» та «Планшети») та відповідне тематичне зображення для полегшення візуальної ідентифікації та швидкого переходу до потрібного розділу каталогу.

У нижній частині сторінки розміщено футер, який містить важливу контактну інформацію, посилання на сторінки магазину в соціальних мережах, адресу електронної пошти для зв'язку, а також коротку інформацію про наш магазин. Футер та хедер є на всіх сторінках крім входу замовлення тому при обзорі інших сторінок вони не буду учитуватися.

Основною метою головної сторінки є надання відвідувачам швидкого уявлення про асортимент товарів, виділення головних пропозицій та забезпечення інтуїтивно зрозумілої навігації по ключових розділах веб-сайту.

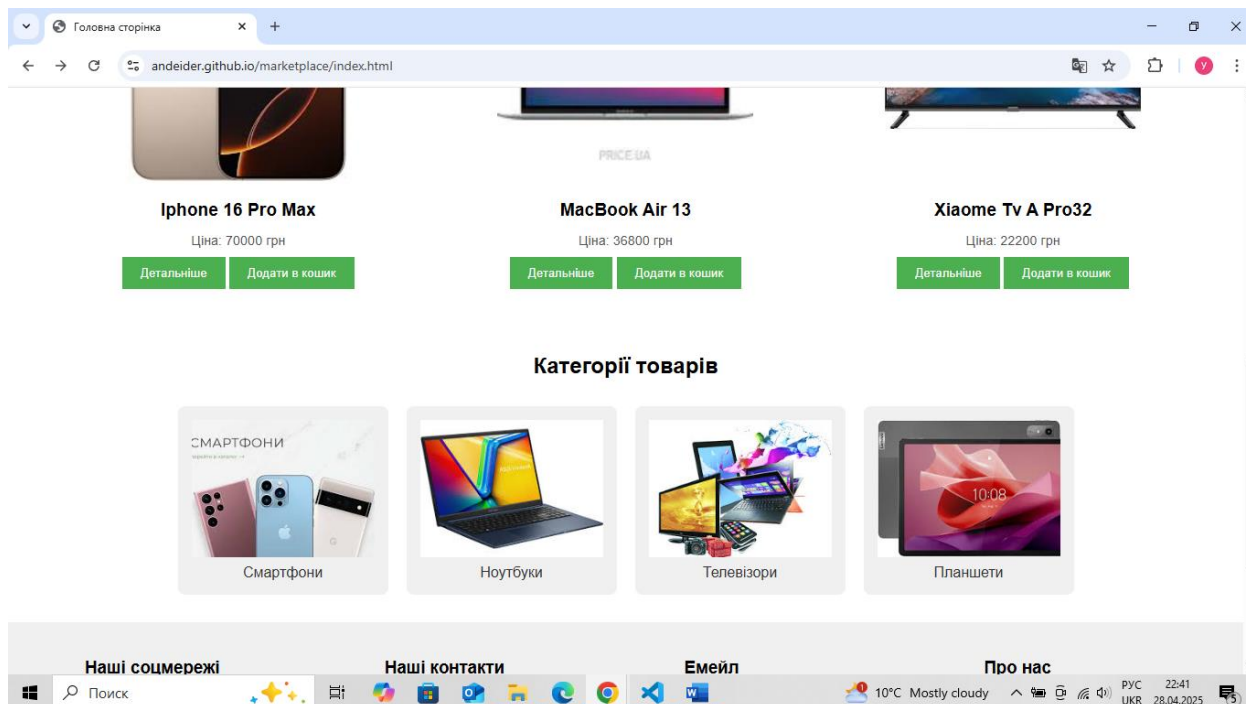


Рисунок – 3.6 Нижня частина головного екрану

Далі при переході через категорії товарів наприклад при натиск на любую вкладку в категоріях перейдемо до сторінки каталог товарів (рис. 3.7). В проєкті файл catalog.html підключено до style.css та catalog.js . Та зразу фільтрується якщо заходили саме з категорії.

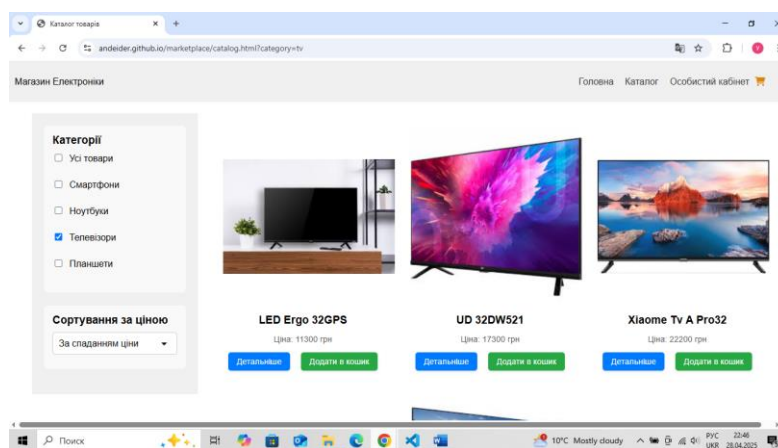


Рисунок 3.7 – Каталог товарів

Сторінка каталогу товарів (catalog.html) була розроблена як центральний елемент для навігації користувачів у широкому асортименті. Вона поєднує в собі

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

HTML для структури, CSS для візуального оформлення та JavaScript для інтерактивності, зокрема для динамічної фільтрації та сортування товарів.

Основу сторінки складала такі ключові секції:

- навігаційним меню («Головна», «Каталог», «Особистий кабінет») та іконкою кошика з лічильником кількості товарів;
- бічна панель фільтрів розташована зліва, містила елементи для фільтрації товарів за різними параметрами;
- категорії товарів представлені у вигляді списку з чекбоксами. Кожен чекбокс відповідав певній категорії («Усі товари», «Смартфони», «Ноутбуки», «Телевізори», «Планшети»);
- сортування за ціною реалізовано через випадаючий список з опціями вибору методу сортування («За спаданням ціни»). У майбутньому планувалося розширення критеріїв сортування.

Основна область сторінки, де відображалися картки товарів, що відповідали застосованим фільтрам та вибраному критерію сортування. Кожна картка товару включає назву товару, ціну товару, кнопки дій, детальніше (рис. 3.8) для відкриття модального вікна.

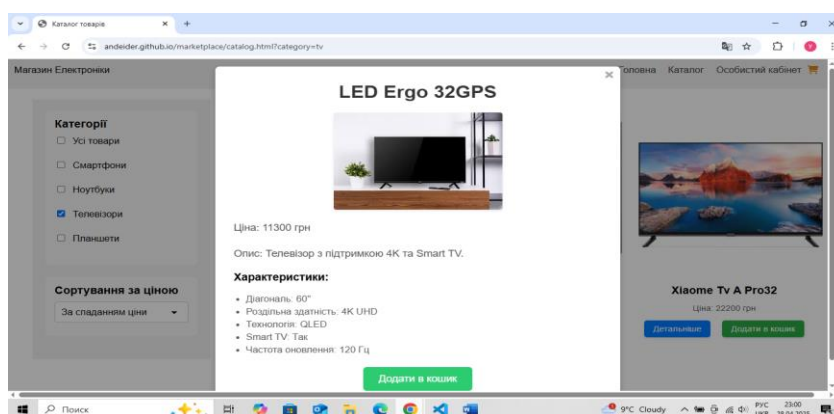


Рисунок 3.8 – Модальне вікно

Та додати в кошик (рис. 3.9) для швидкого додавання товару до кошика користувача.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 44 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |



Рисунок 3.9 – Показ що в кошик товар доданий

Стилізація сторінки каталогу за допомогою CSS була спрямована на створення зручного та візуально привабливого інтерфейсу:

- застосовувалися сучасні методи компоювання (Flexbox або CSS Grid) для створення гнучкої двоколонної структури, що оптимально розміщувала бічну панель фільтрів та основний контент зі списком товарів;
- елементи бічної панелі були стилізовані для забезпечення зручної навігації та інтуїтивного вибору критеріїв фільтрації. Візуальне виділення активної категорії допомагало користувачеві орієнтуватися;
- оформлення карток товарів було розроблено таким чином, щоб чітко відображати ключову інформацію (зображення, назва, ціна) та забезпечувати легкий доступ до кнопок взаємодії;
- завдяки медіа-запитам забезпечувалася коректне відображення сторінки на різних розмірах екранів, гарантуючи зручність перегляду як на настільних комп'ютерах, так і на мобільних пристроях;

JavaScript відігравав ключову роль у наданні сторінці каталогу динамічної функціональності:

- реалізовано обробку подій зміни стану чекбоксів категорій. При виборі або скасуванні вибору категорії, JavaScript виконував асинхронний запит до бекенду для отримання оновленого списку товарів, що відповідають вибраним фільтрам. Отримані дані динамічно відображались в основній частині сторінки без необхідності повного перезавантаження;
- оброблялася подія зміни значення у випадаючому списку «Сортування за ціною». При виборі іншого критерію сортування, JavaScript надсилав відповідний

запит на сервер, отримував відсортований список товарів та оновлював відображення в основній області;

- при натисканні на кнопку «Додати в кошик» для конкретного товару, JavaScript ініціював асинхронний запит до бекенду для додавання цього товару до кошика поточного користувача. Після успішного виконання цієї операції міг оновлюватися лічильник товарів, відображений у шапці сторінки.

Реалізація сторінки каталогу була націлена на надання користувачам ефективних та зручних інструментів для пошуку та вибору необхідних товарів, забезпечуючи швидку та інтуїтивно зрозумілу взаємодію з інтерфейсом.

Сторінка кошика (cart.html) дозволяла користувачам переглядати та редагувати обрані товари (рис. 3.10) перед оформленням замовлення.

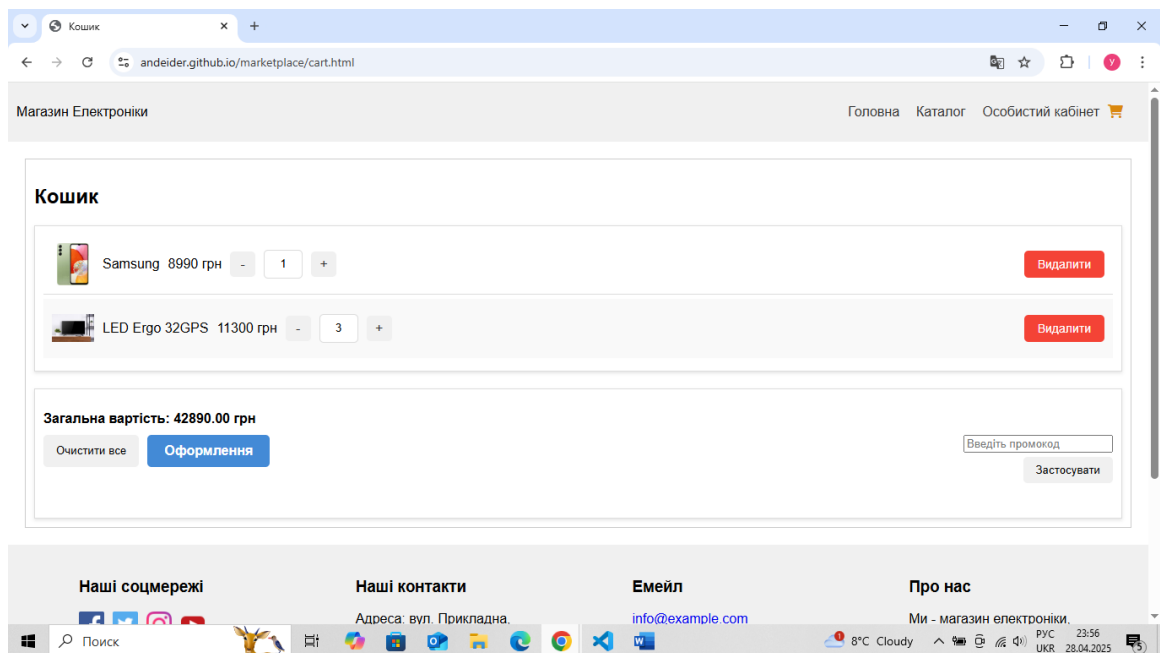


Рисунок 3.10 – Сторінка кошика

Відображення товарів (дані з localStorage), зміна кількості (+/- з динамічним перерахунком вартості), видалення товару (оновлення відображення, localStorage, загальної вартості), очищення (рис. 3.13) всього кошика (видалення з відображення, localStorage, обнулення загальної вартості), перехід на (рис. 3.14)

checkout.html (кнопка «Оформлення»), базова імітація застосування промокоду (рис. 3.11-3.12).

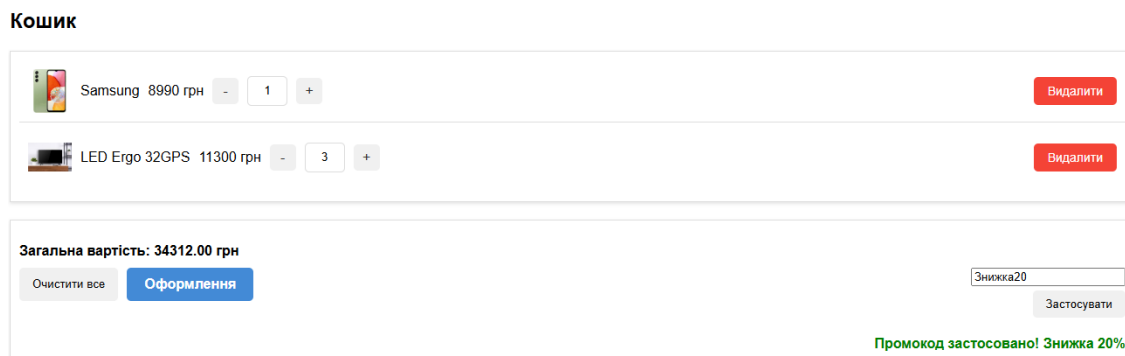


Рисунок 3.11 – Застосування промокоду

Після введення система перевіряє валідність промокоду чи він існує, чи дійсний, чи не вичерпано його ліміт використання. У разі успішної перевірки, сума замовлення автоматично перераховується з урахуванням наданої промокодом знижки. Ця функціональність забезпечує гнучкість у проведенні маркетингових акцій та підвищує лояльність клієнтів.

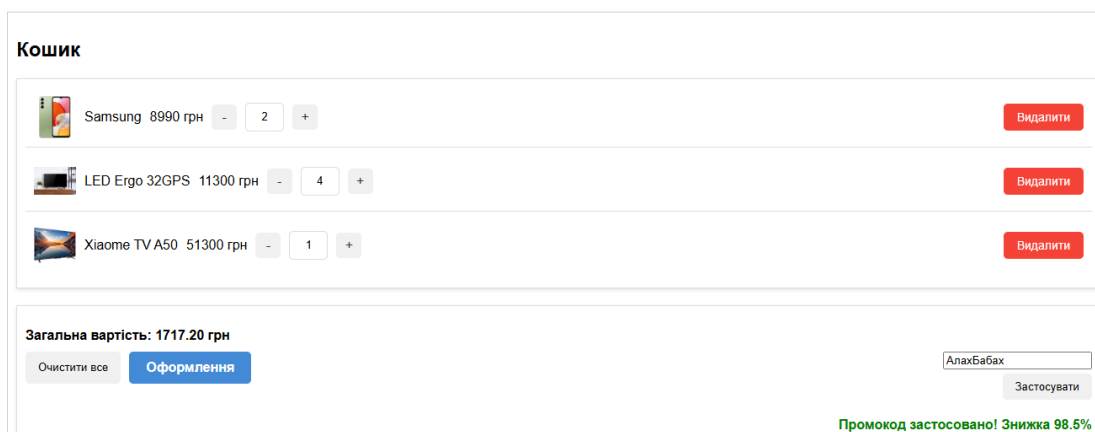


Рисунок 3.12 – Додавання товару та застосування іншого промокоду

Функціонал додавання товару дозволяє користувачеві помістити обрані позиції з каталогу у свій віртуальний кошик. Ця дія ініціюється натисканням

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

відповідної кнопки на сторінці товару і супроводжується оновленням інформації про вміст кошика, наприклад, відображенням загальної кількості товарів або суми.

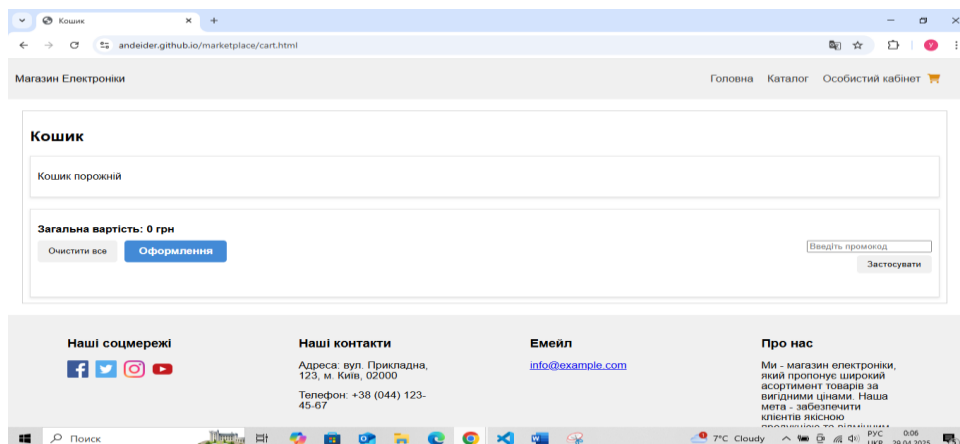


Рисунок 3.13 – Показ функції очищення кошика

Сторінка кошика забезпечувала основні можливості керування вмістом перед оформленням замовлення, використовуючи клієнтські технології для динамічного оновлення інтерфейсу та тимчасового зберігання даних. При оформленні переходить на сторінку оформлення (рис. 3.14) замовлення сторінка знаходиться в файлі checkout.html він підключений до файлу checkout.js та через великий обсяг файлу style.css для цієї сторінки створив інший стильовий файл check.css .

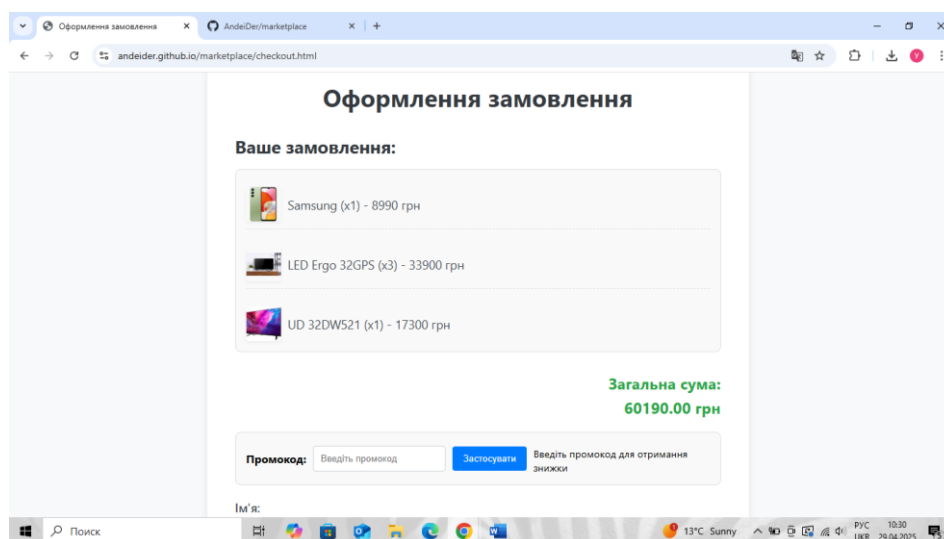


Рисунок 3.14 – Сторінка оформлення замовлення

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

Сторінка оформлення замовлення (checkout.html) була розроблена як завершальний крок перед підтвердженням покупки, де користувач вводив дані для доставки та оплати. На початковому етапі розробки її функціонал обмежувався збором інформації та імітацією оформлення замовлення.

Основна структура включала:

- короткий огляд товарів до замовлення (кількість, ціна);
- обов'язкові поля для імені, прізвища, телефону, email;
- поля для країни, області, міста, індексу, адреси (рис. 3.15);
- радіо-кнопки зі статичними варіантами. Вартість доставки при виборі були відсутні при виборі самовивозі або фіксована при іншому виборі;
- радіо-кнопки зі статичними варіантами (наприклад, «При отриманні»).

Інтеграція з платіжними системами відсутня;

- загальна вартість товарів, можливо, з фіксованою вартістю доставки
- імітація відправки даних замовлення.

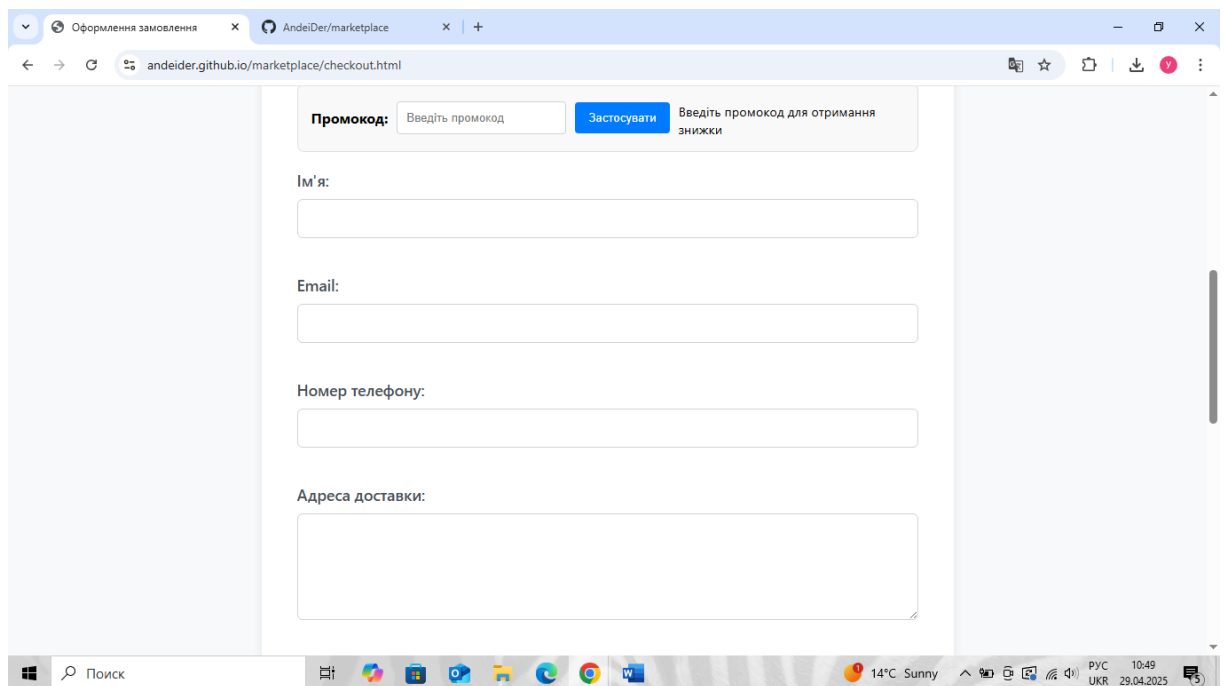


Рисунок 3.15 – Частина сторінки оформлення замовлення для вводу даних

Функціональність (JavaScript):

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 49 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

- інформація про товари з localStorage відображалася при завантаженні сторінки;
- базова перевірка обов'язкових полів контактних даних та адреси на клієнті;
- вибір доставки та оплати (рис. 3.16) щоб користувач міг вибрати один із статичних варіантів в відображенні підсумкової суми сума доставки додається до вартості обраних товарів та при обранні способу оплати при отриманні (Рис. 3.16) частину сторінки, а саме вводу даних карти скривається;
- відображення підсумкової суми, розраховувалася на основі товарів у кошику та фіксованої доставки (рис. 3.16);
- імітація оформлення, при натисканні «Підтвердити замовлення» виводилося повідомлення про успішне оформлення або відбувався перехід на тимчасову сторінку особистого кабінету(рис. 3.17). Реальна обробка замовлення на не була реалізована на цьому етапі розробки.

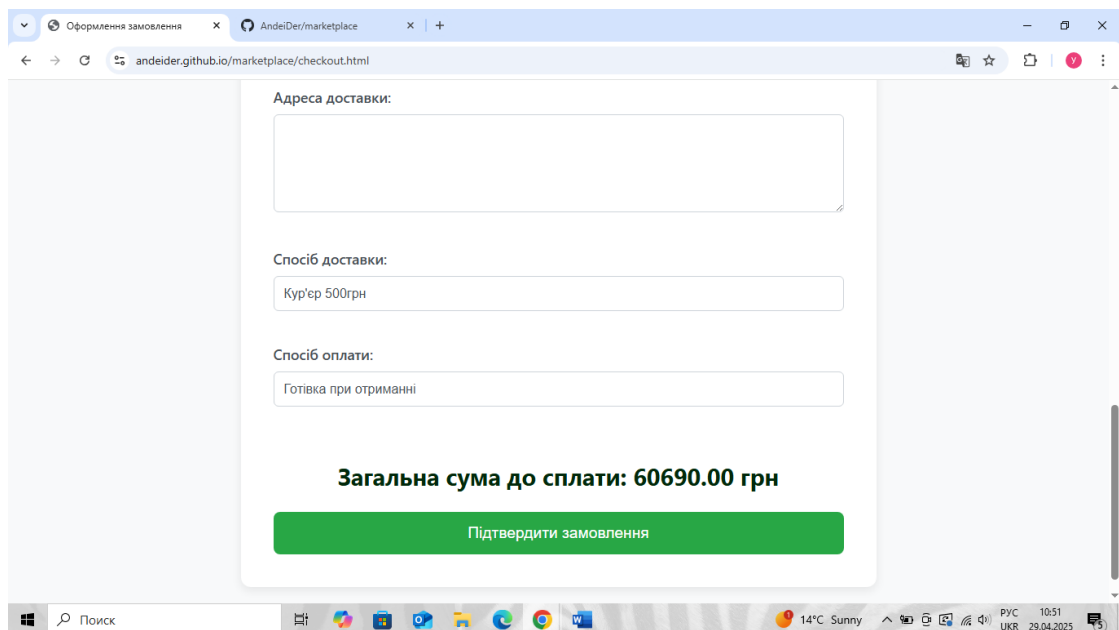


Рисунок 3.16 – Вибір способу доставки та оплати та загальна сума

Сторінка оформлення замовлення на цьому етапі слугувала для збору даних та імітації завершення покупки, готуючи основу для інтеграції з бекендом та платіжними системами.

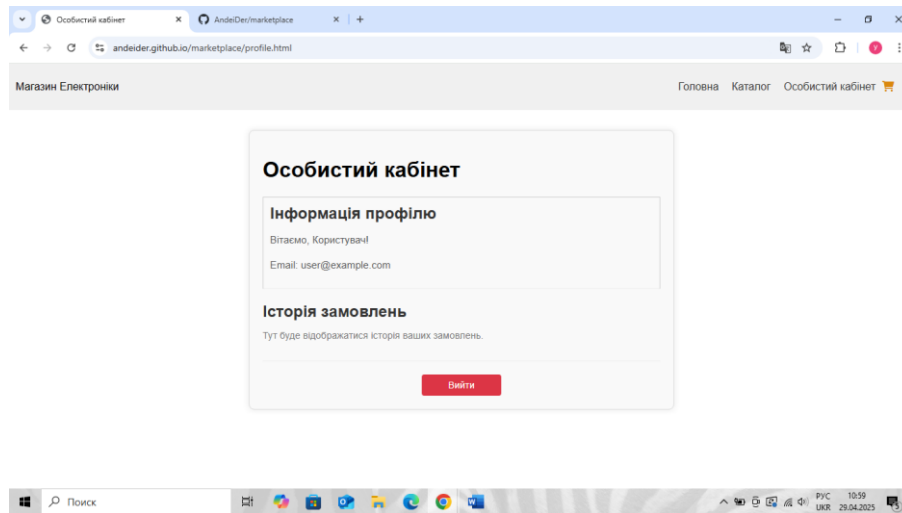


Рисунок 3.17 – Особистий кабінет

Сторінка особистого кабінету (profile.html) була розроблена для надання зареєстрованим користувачам доступу до їхньої основної інформації та базової історії взаємодії з магазином. Також для нього також інший файл стилів profile.css. На початковому етапі розробки функціональність цієї сторінки була досить обмеженою. Перед переходом на цю сторінку потрібно пройти авторизація та вхід (рис. 3.18-3.19).

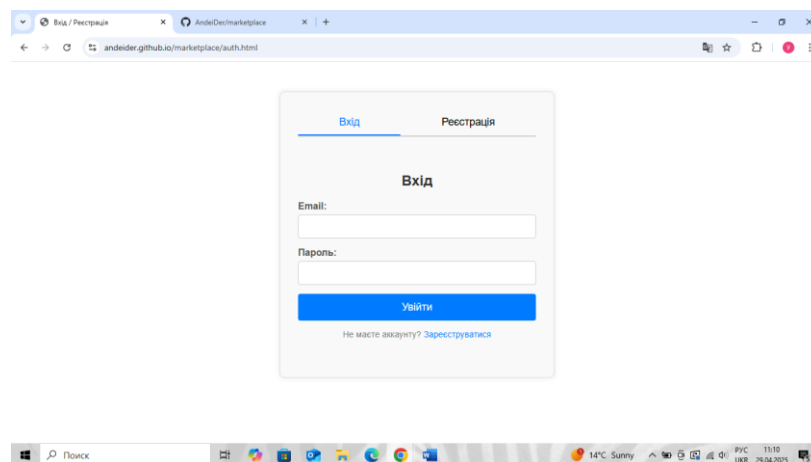


Рисунок 3.18 – Вхід

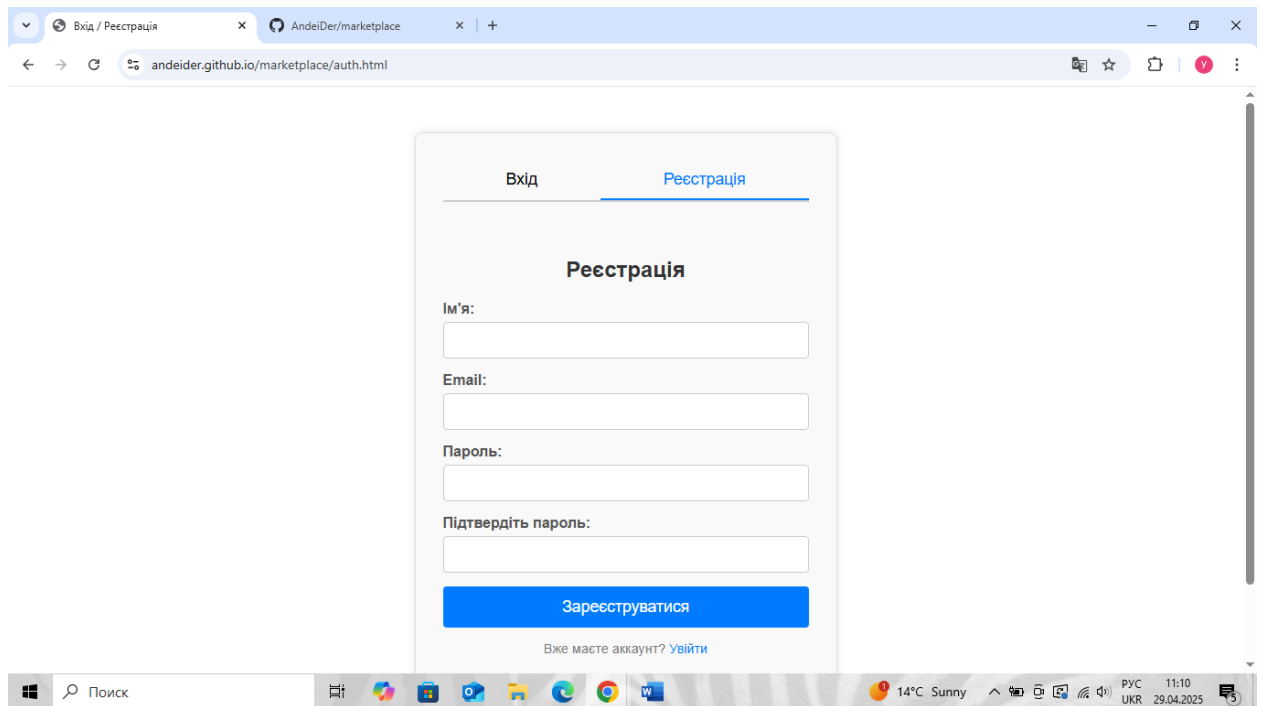


Рисунок 3.19 – Реєстрація

Для забезпечення зручного користування інтернет-магазином на різних пристроях (десктопи, планшети, смартфони) на всіх основних сторінках було реалізовано адаптивний дизайн. Це досяглося за допомогою комбінації HTML, CSS та мета-тегів viewport.

Використання відносних одиниць вимірювання (відсотки, em, rem) замість фіксованих пікселів для визначення ширини та висоти елементів дозволяло елементам гнучко змінювати свої розміри залежно від розміру екрана. Технології Flexbox та CSS Grid активно використовувалися для створення адаптивних та гнучких макетів сторінок.

Для оптимізації завантаження сторінок на різних пристроях використовувалися адаптивні зображення. Це могло включати використання атрибута <picture> або атрибута srcset тега для надання браузеру кількох варіантів зображення різної роздільної здатності, з яких браузер вибирав найбільш підходящий для поточного пристрою.

Для поліпшення навігації на мобільних пристроях на всіх основних сторінках (головна, каталог, особистий кабінет, кошик, оформлення замовлення) було реалізовано бургер-меню (рис. 3.20).

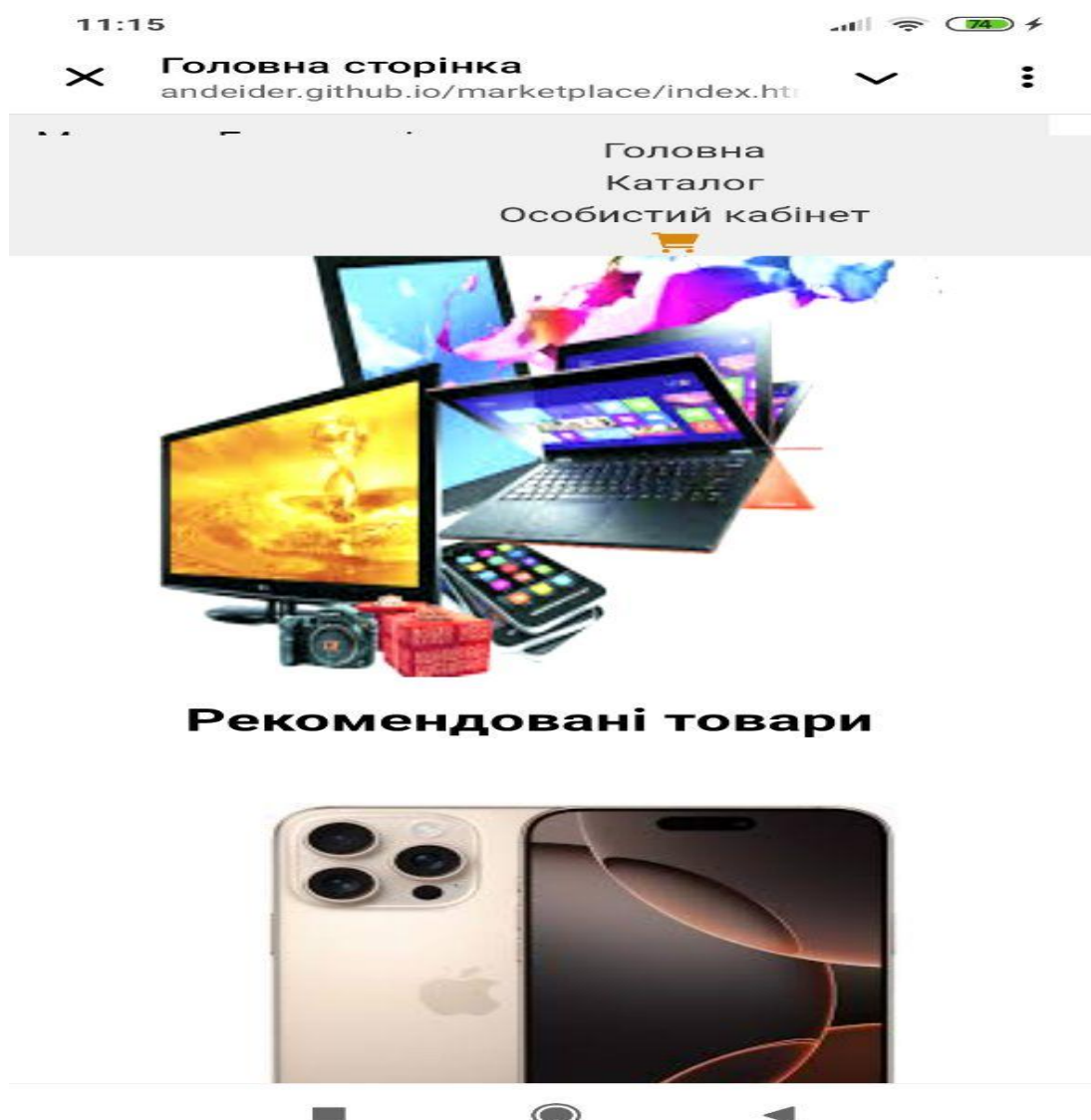


Рисунок 3.20 – Реалізація бургер меню

Головна сторінка (рис. 3.21). Адаптувалися блоки з головними пропозиціями та категоріями товарів, змінюючи їхнє розташування та кількість колонок залежно від ширини екрана. Банер міг змінювати розміри або відображатися по-різному.



Xiaome Tv A Pro32

Ціна: 22200 грн

Детальніше Додати в кошик

Категорії товарів



Рисунок 3.21 – Головна сторінка з телефону

Каталог товарів (рис. 3.22-3.23) бічна панель фільтрів трансформувалася для мобільних пристроїв .Список товарів змінював кількість колонок.

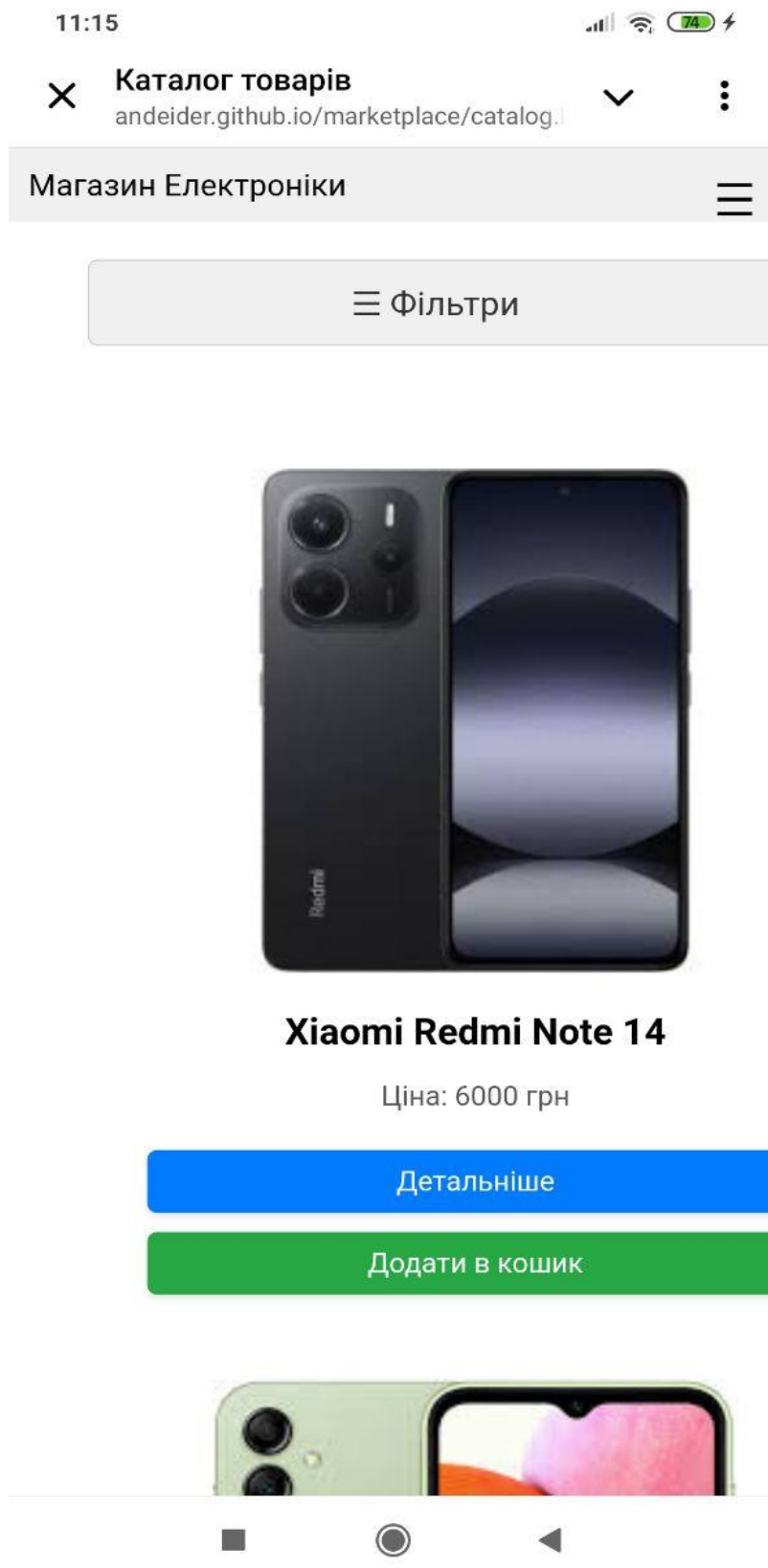


Рисунок 3.22 – Каталог для мобільних пристроїв

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

Каталог для мобільних пристроїв розроблений з урахуванням принципів адаптивного дизайну, що забезпечує коректне та зручне відображення товарів на екранах різного розміру від смартфонів до планшетів.

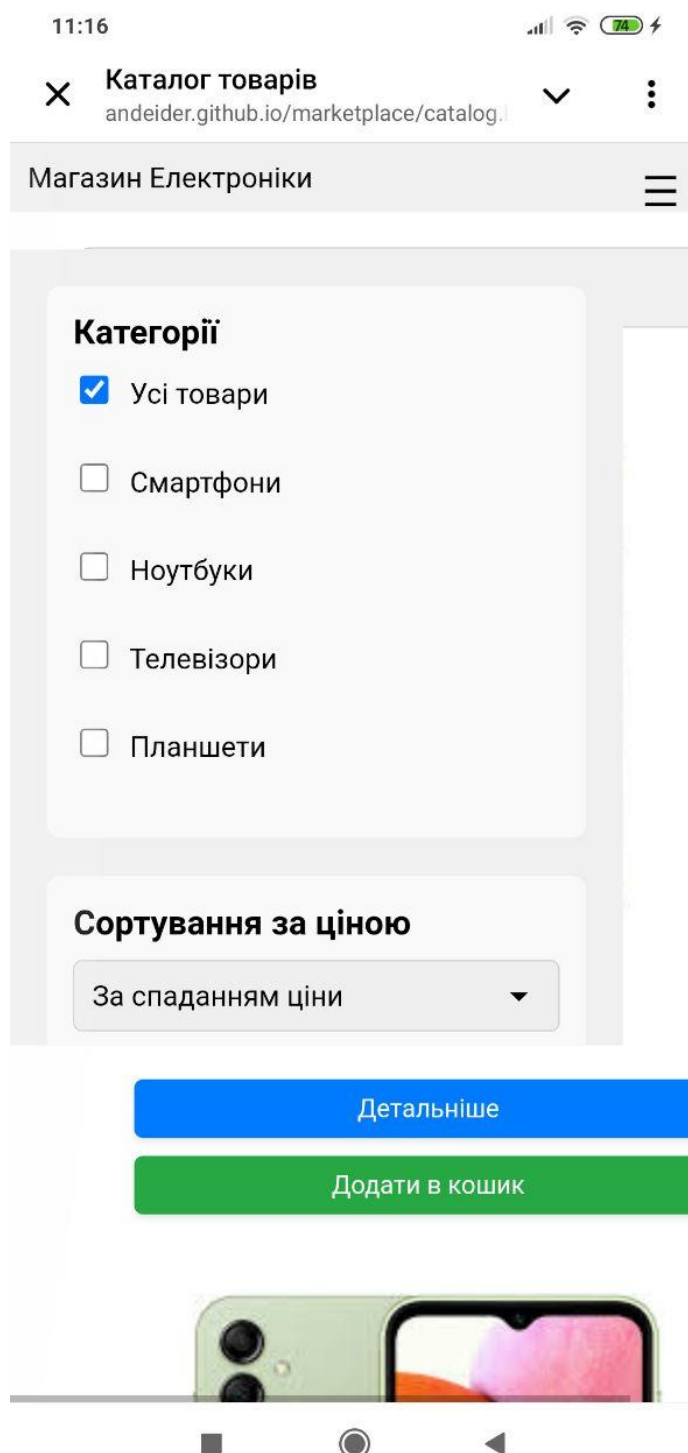


Рисунок 3.23 – Фільтрація для товарів в каталозі з мобільного пристрою

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

Таблиця з товарами могла трансформуватися у вертикальний список на мобільних пристроях для кращого відображення інформації. Кнопки дій розташовувалися таким чином, щоб бути легкодоступними.

Кошик інтернет-магазину (рис. 3.24), оптимізований для мобільних пристроїв, надає користувачам зручний та інтуїтивно зрозумілий інтерфейс для керування вибраними товарами перед оформленням замовлення. На сторінці кошика товари відображаються у компактному, але інформативному вигляді, включаючи назву товару, його мініатюрне зображення, ціну за одиницю та загальну суму

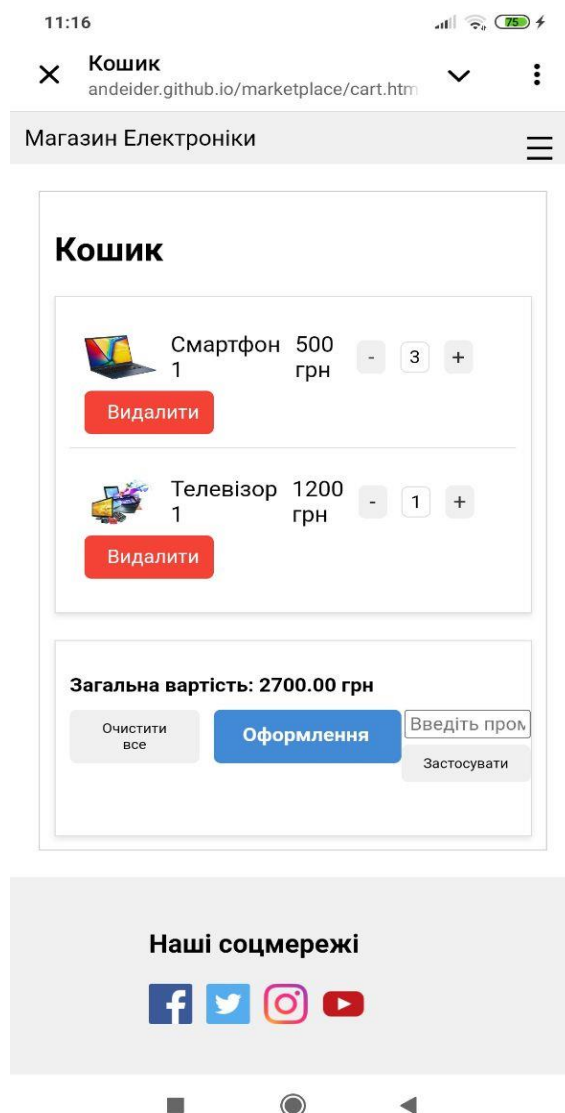


Рисунок 3.24 – Кошик з мобільного застосунку

Адаптована сторінка особистого кабінету зображена на рисунку 3.25.

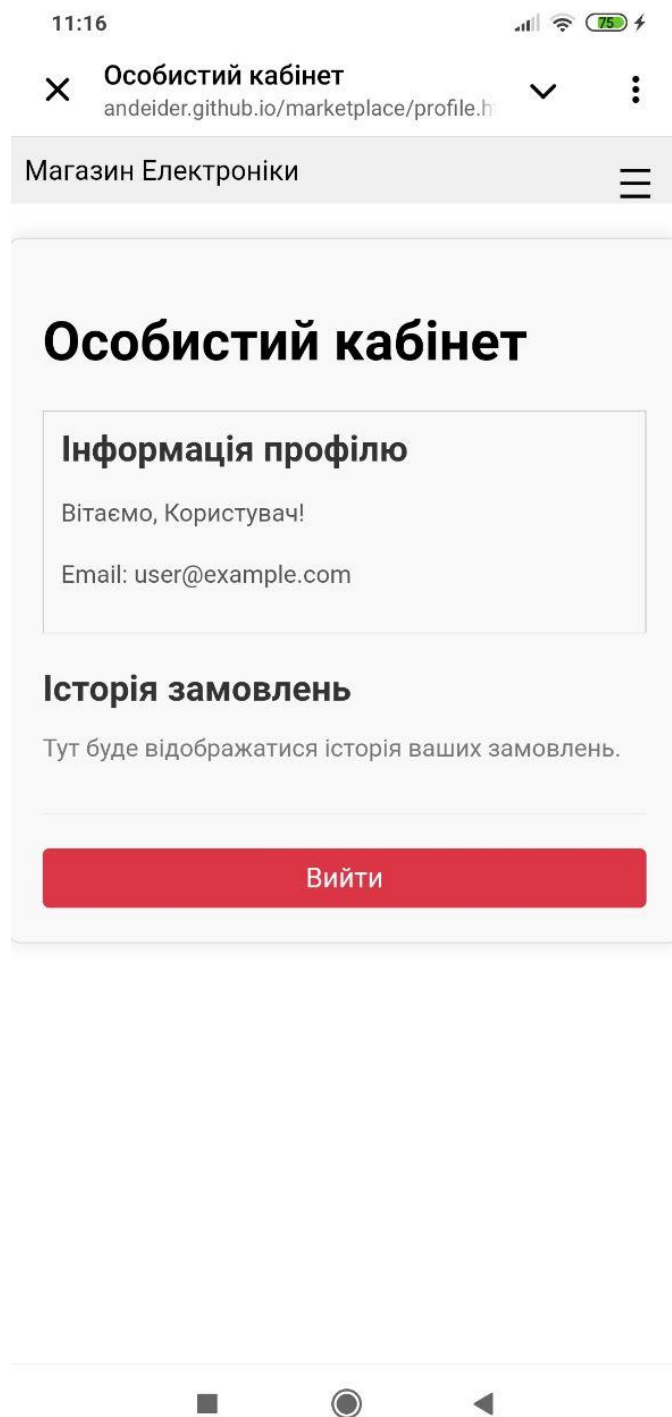


Рисунок 3.25 – Особистий кабінет з мобільного пристрою

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 58 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Форми для введення контактних даних та адреси доставки адаптувалися за шириною. Етапи відображалися зручно на різних екранах реалізація на рисунку 3.26.

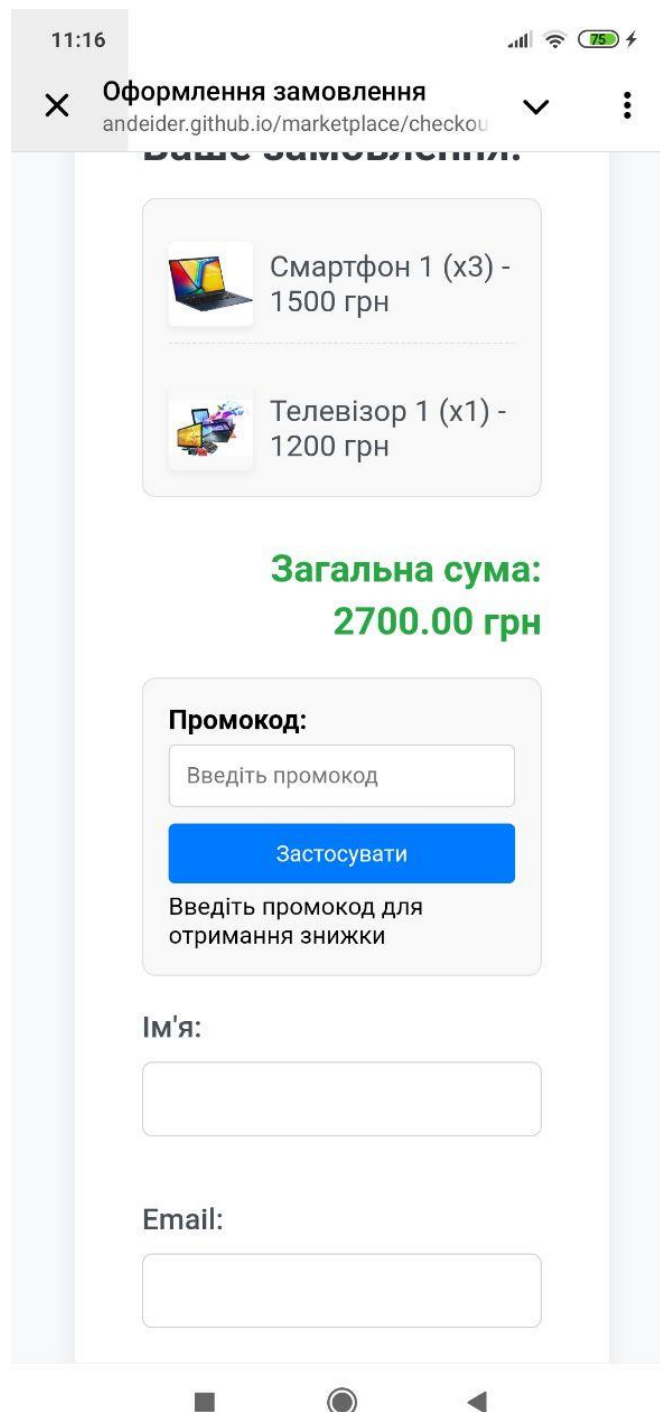


Рисунок 3.26 – Сторінка оформлення з мобільного пристрою

Отже, застосування цих підходів дозволило моєму інтернет-магазину досягти базового рівня зручності для користувачів, незалежно від типу пристрою, який вони використовують для перегляду.

Всю фронтенд частина виставлена в github pages (рис. 3.27) та доступна в інтернеті за посиланням <https://andeider.github.io/marketplace/>.

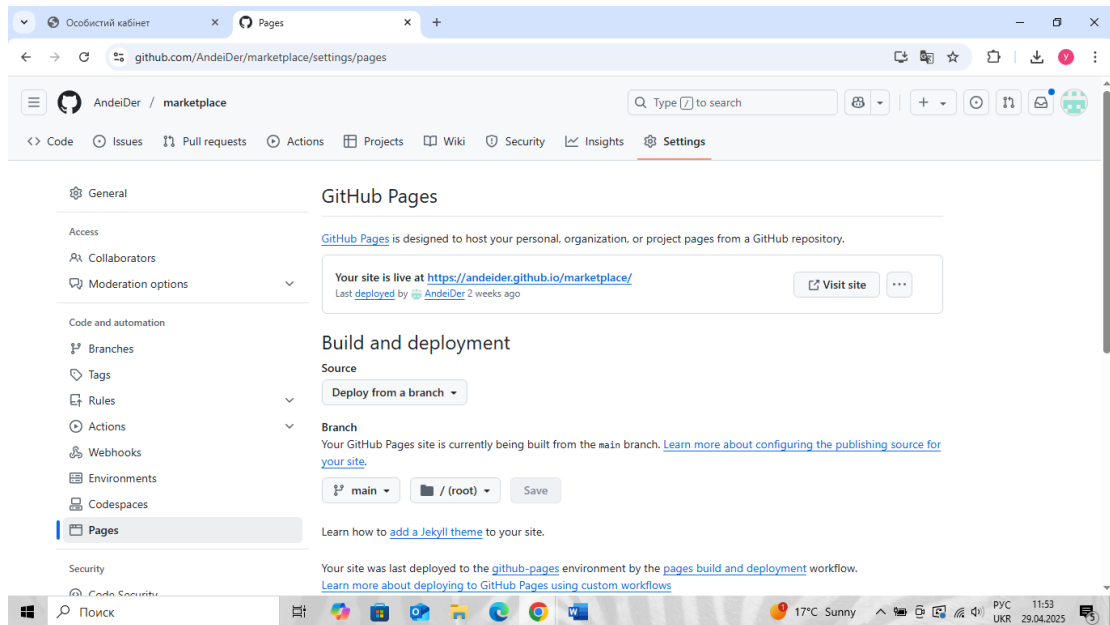


Рисунок 3.27 – GitHub Pages та посилання сайту

3.3 Реалізація Бекенду та тестування

Бекенд інтернет-магазину був розроблений з використанням Node.js та фреймворку Express.js. Для зберігання даних використовувалася база даних MongoDB. Бекенд відповідав за обробку запитів від фронтенду, реалізацію бізнес-логіки та взаємодію з базою даних.

Для початку розробки серверної частини інтернет-магазину першим кроком була ініціалізація Node.js проєкту та встановлення всіх необхідних бібліотек (пакунків) за допомогою npm.

1. Ініціалізація проєкту Node.js:

Проєкт ще не був ініціалізований, тому відкривши термінал у кореневій директорії бекенду та запускав наступну команду npm init -y.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 60 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

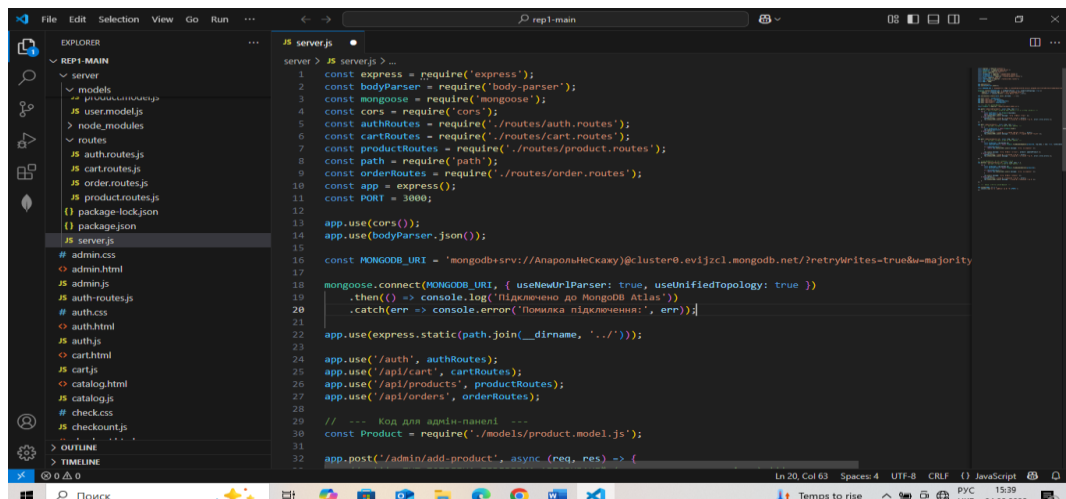
Ця команда автоматично створювала файл package.json з базовими налаштуваннями проєкту, використовуючи значення за замовчуванням. Файл package.json є важливим для відстеження залежностей проєкту, скриптів для запуску та інших метаданих.

Для розробки серверної частини інтернет-магазину на Node.js було використано менеджер пакунків npm для встановлення всіх необхідних бібліотек. Після первинної ініціалізації проєкту за допомогою команди npm init -у встановлено основні залежності, виконавши наступну команду в терміналі npm install express mongoose bcrypt.

Ця команда автоматично завантажила та встановила три ключові пакунки:

- express;
- mongoose;
- bcrypt.

Ці три пакунки склали основу функціональності бекенду, забезпечуючи можливість обробки HTTP-запитів від клієнтської частини, взаємодії з базою даних MongoDB для зберігання та отримання даних, а також безпечне керування обліковими даними користувачів (рис. 3.29). На рисунку 3.28 зображено код з підключенням до бази даних, паролі від бази даних було змінено.



```
server > JS server.js > ...
1  const express = require('express');
2  const bodyParser = require('body-parser');
3  const mongoose = require('mongoose');
4  const cors = require('cors');
5  const authRoutes = require('./routes/auth.routes');
6  const cartRoutes = require('./routes/cart.routes');
7  const productRoutes = require('./routes/product.routes');
8  const path = require('path');
9  const orderRoutes = require('./routes/order.routes');
10 const app = express();
11 const PORT = 3000;
12
13 app.use(cors());
14 app.use(bodyParser.json());
15
16 const MONGODB_URI = 'mongodb+srv://AnapolyeCkaxy@cluster0.evijzcl.mongodb.net/?retryWrites=true&majority
17
18 mongoose.connect(MONGODB_URI, { useNewUrlParser: true, useUnifiedTopology: true })
19   .then(() => console.log('Підключено до MongoDB Atlas'))
20   .catch(err => console.error('Помилка підключення:', err));
21
22 app.use(express.static(path.join(__dirname, '..')));
23
24 app.use('/auth', authRoutes);
25 app.use('/api/cart', cartRoutes);
26 app.use('/api/products', productRoutes);
27 app.use('/api/orders', orderRoutes);
28
29 // --- Код для admin-панелі ---
30 const Product = require('./models/product.model.js');
31
32 app.post('/admin/add-product', async (req, res) => {
```

Рисунок 3.28 – Підключення до бази даних

Після успішного встановлення з'єднання отримано можливість використовувати моделі Mongoose для взаємодії з колекціями в базі даних MongoDB, виконувати операції CRUD. Цей підхід значно спростив розробку, оскільки моделі Mongoose забезпечують валідацію даних перед їх збереженням, дозволяють визначати схеми для забезпечення консистентності інформації та надають зручний API для виконання складних запитів.

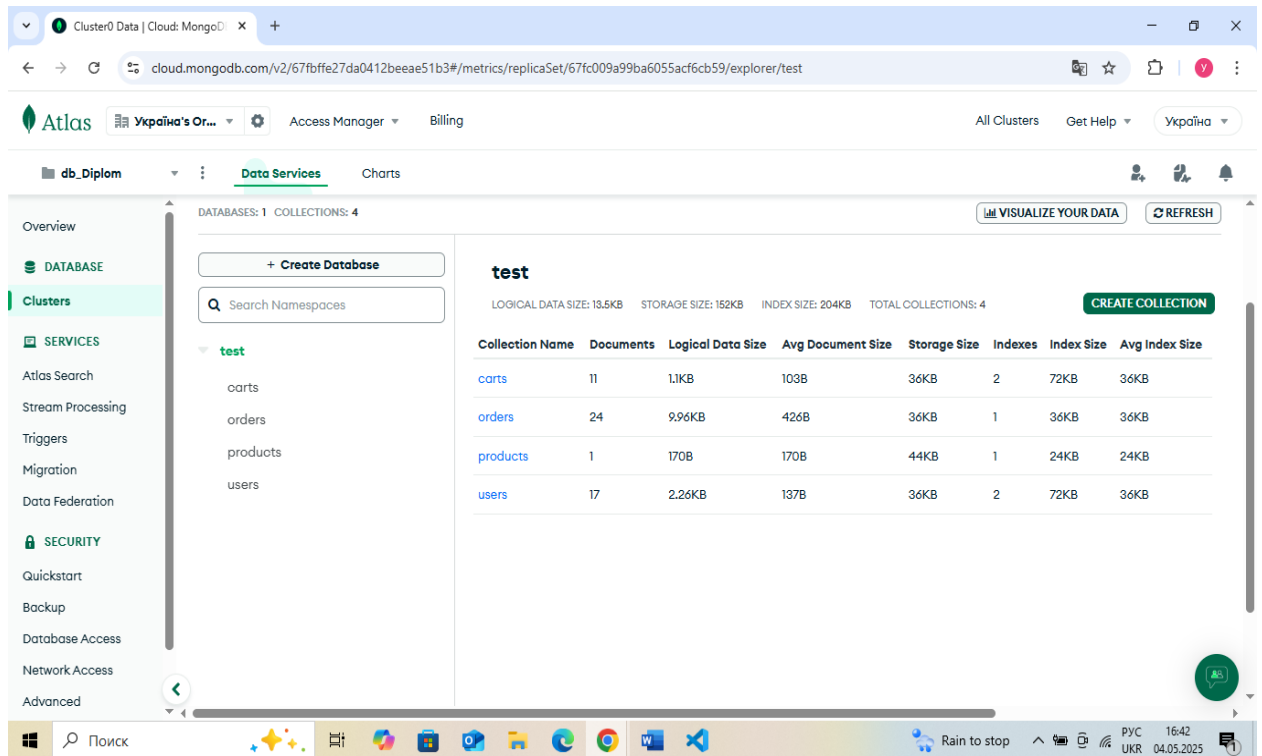


Рисунок 3.29 – Реалізована база даних магазину

Структура бази даних була ретельно розроблена для ефективного зберігання та управління даними інтернет-магазину, враховуючи основні сутності: користувачів, товари, кошики та замовлення. Кожна з цих колекцій спроектована з урахуванням специфічних потреб відповідних даних та взаємозв'язків між ними, що забезпечує гнучкість та масштабованість системи. Зокрема, використання нереляційної моделі MongoDB дозволяє легко адаптувати схему даних до майбутніх змін у вимогах бізнесу та додавання нових функціональностей без значних перебудов.

Таблиця 3.1 – Структура та опис таблиці користувача

| Назва поля | Тип даних | Короткий опис |
|------------|------------------------|---|
| id | ObjectId | Унікальний ідентифікатор користувача |
| email | String | Електронна адреса користувача (унікальна) |
| password | String | Хешований пароль користувача |
| cart | ObjectId (ref: 'Cart') | Посилання на кошик користувача |

Колекція users у базі даних (рис. 3.30) MongoDB інтернет-магазину є центральною для зберігання всієї інформації, пов'язаної із зареєстрованими користувачами та їхніми обліковими записами. Кожен документ у цій колекції представляє окремого користувача.

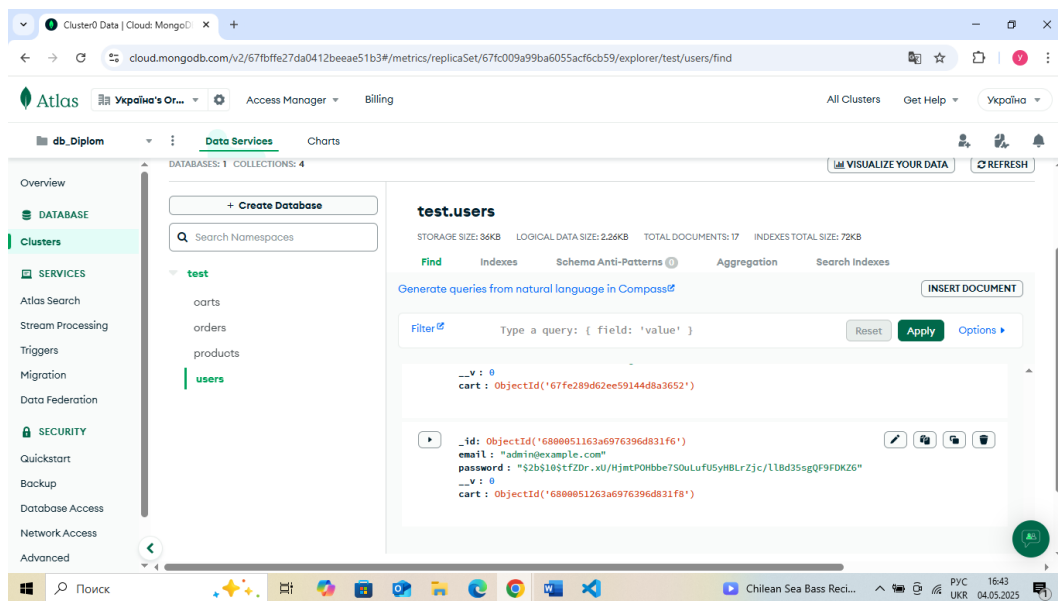


Рисунок – 3.30 дані з колекції users в MongoDB

Таблиця 3.2 – Структура та опис таблиці Товари

| Назва поля | Тип даних | Короткий опис |
|-------------|-----------|---------------------------------|
| _id | ObjectId | Унікальний ідентифікатор товару |
| name | String | Назва товару |
| description | String | Опис товару |
| price | Number | Ціна товару |
| category | String | Категорія товару |
| image | String | URL-адреса зображення товару |

Товари знаходяться не в базі даних ,а в форматі .JSON, прописані в файлі product, а цей кластер бази даних створений для сторінки адмін панелі яку було реалізовано після встановлення всього фронтенту та бекенду до того ж адмін панель це також робота з Бекендом тому фронтенд частину для сторінки адміна (рис 3.31) добавлено в цей розід роботи.

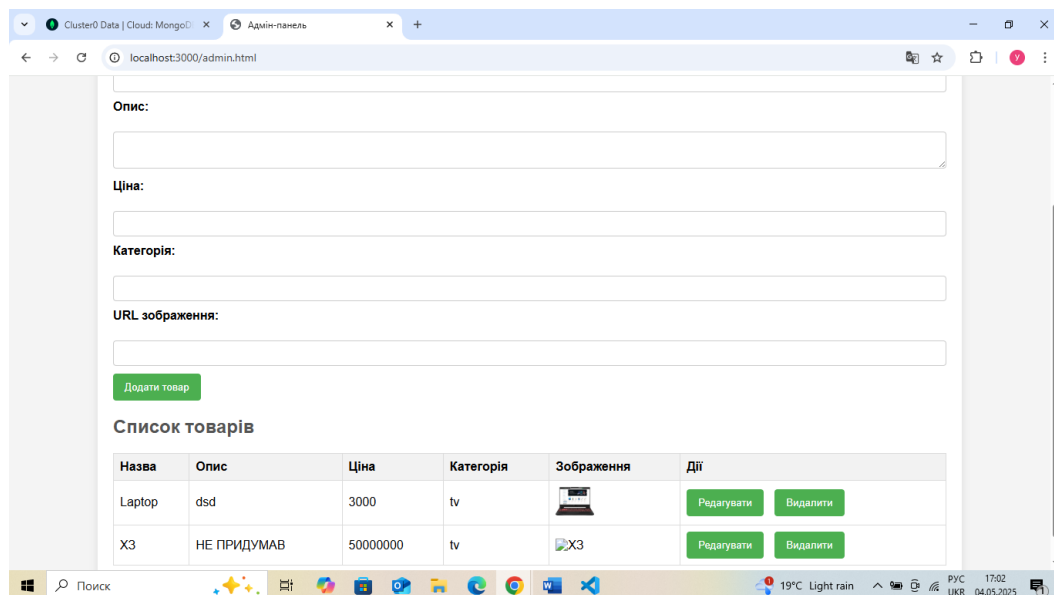


Рисунок 3.31 – Адмін панель

Як вона працює, при додані, видалені чи редагувані в адмін панелі дані надсилаються на сервер. Далі розробник додає вже в файл product якщо це потрібно та погоджено з менеджером.

Таблиця 3.3 – «Замовлення» зберігає інформацію про оформлені замовлення.

| Назва поля | Тип даних | Короткий опис |
|-----------------|------------------------|--|
| _id | ObjectId | Унікальний ідентифікатор замовлення |
| userId | ObjectId (ref: 'User') | Ідентифікатор користувача, який зробив замовлення |
| items | Array (Object) | Масив товарів у замовленні (productId, name, quantity, price, image) |
| totalAmount | Number | Загальна сума замовлення |
| shippingAddress | String | Адреса доставки |
| paymentMethod | String | Спосіб оплати |
| orderDate | Date | Дата та час оформлення замовлення |
| shippingPrice | Number | Вартість доставки |
| discountAmount | Number | Сума знижки |

У базі даних, колекція orders (рис. 3.32) є однією з ключових, оскільки вона зберігає всю інформацію, що стосується здійснених покупок в інтернет-магазині. Кожен документ у цій колекції представляє собою окреме замовлення і

спроектований таким чином, щоб містити повний набір даних, необхідних для його обробки, відстеження та аналізу.

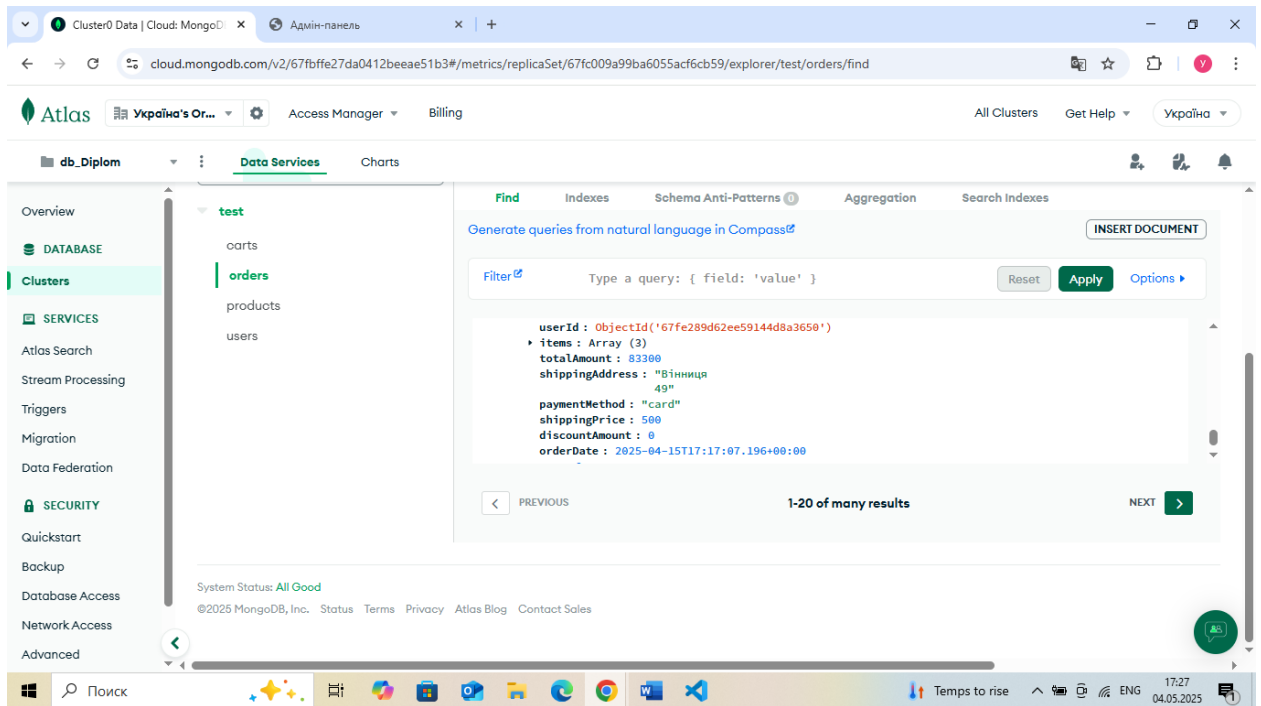


Рисунок 3.32 – Кластер orders в mongodb

Таблиця 3.4 – Кошик зберігає інформацію для кошика

| Назва поля | Тип даних | Короткий опис |
|------------|------------------------|--|
| _id | ObjectId | Унікальний ідентифікатор кошика |
| userId | ObjectId (ref: 'User') | Ідентифікатор користувача, якому належить кошик (унікальний) |
| items | Array (Object) | Масив товарів у кошику |

Описана структура бази даних MongoDB забезпечила гнучке та ефективно зберігання даних інтернет-магазину та дозволила встановити необхідні зв'язки між основними сутностями. Використання Mongoose спростило моделювання даних та взаємодію з базою даних у бекенді на Node.js.

У процесі розробки застосовувалося тестування одразу після написання коду, що включало кілька ключових етапів. Насамперед, здійснювалося ручне тестування бекенду, під час якого перевірялася працездатність API, моделей та сервісів. Це відбувалося шляхом надсилання тестових запитів для детального аналізу обробки даних, логіки функціонування, взаємодії з базою даних та відповідності формату відповідей.

Окрім того, проводилася ретельна перевірка моделей Mongoose. Для цього використовувалися скрипти або консоль Node.js, що дозволяло тестувати операції CRUD (створення, читання, оновлення, видалення) з базою даних, забезпечуючи їх відповідність визначеним моделям.

Щодо фронтенд-компонентів, то їх тестування реалізовувалося через візуальну перевірку. Оцінювалося відображення React-компонентів з різними властивостями (пропсами) та станами за допомогою інструментів розробника браузера. Активно застосовувалася функція console.log для відстеження обробки подій та змін стану, що сприяло виявленню потенційних проблем.

Важливою частиною були швидкі інтеграційні перевірки, які передбачали ручну верифікацію базової взаємодії між невеликими інтегрованими частинами фронтенду та відповідними API-ендпоінтами бекенду. Прикладами таких перевірок були відображення отриманих даних або коректна відправка даних форм. Активне застосування console.log у коді виявилось корисним інструментом для відстеження потоку виконання, значень змінних та оперативного виявлення будь-якої неочікуваної поведінки системи, такий підхід до тестування забезпечував швидке виявлення та виправлення помилок ще на початкових етапах розробки, що заощаджувало час. Крім того, це сприяло поглибленому розумінню власної логіки коду в процесі тестування, зменшуючи час на відладку в майбутньому завдяки ранньому виправленню помилок та формувало звичку одразу перевіряти свій код.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 67 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

3.4 Висновки до третього розділу

У цьому розділі розглянуто шлях створення інтернет-магазину, починаючи від проектування архітектури фронтенду та бекенду, через етапи їхньої безпосередньої розробки, і до застосування стратегій тестування.

Особливу увагу було приділено реалізації адаптивного дизайну для забезпечення безперебійної роботи на різних пристроях шляхом використання медіа-запитів та бургер-меню для мобільних версій.

Серверна частина розроблялася на Node.js з використанням фреймворку Express.js. Для зберігання даних обрано NoSQL базу даних MongoDB, а взаємодія з нею здійснювалася через бібліотеку Mongoose. Розглянуто процес встановлення залежностей за допомогою npm, підключення до бази даних та структуру основних колекцій (користувачі, товари, замовлення, кошики) з описом їхніх полів. Бекенд надавав RESTful API для комунікації з фронтендом, забезпечуючи обробку запитів, реалізацію бізнес-логіки та управління даними.

Для забезпечення високої якості та надійності інтернет-магазину було застосовано комплексний підхід до тестування на різних етапах розробки. Особливий акцент було зроблено на тестуванні одразу після написання коду для оперативного виявлення та усунення помилок.

Процес розробки інтернет-магазину був комплексним та включав проектування архітектури, реалізацію функціональності як клієнтської, так і серверної частин, а також забезпечення якості через систематичне тестування на кожному етапі що забезпечило реалізацію необхідних функцій та стабільне функціонування.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 68 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

ВИСНОВКИ

У цій роботі було розроблено функціональний інтернет-магазин з використанням обраного стеку технологій.

У першому розділі було здійснено аналіз існуючих інтернет-магазинів, виявлено їхні сильні та слабкі сторони, а також сформульовано основні функціональні вимоги до розроблюваного проєкту. Було підкреслено ключові аспекти, що визначають успіх платформи електронної комерції, такі як зручність користувацького інтерфейсу, ефективність навігації, безпека транзакцій та потенціал масштабування.

Другий розділ був присвячений детальному огляду та обґрунтуванню вибору технологій розробки. Описано основні етапи процесу онлайн-продажу, структуру клієнтського інтерфейсу (хедер, футер, меню) та перелік ключових сторінок інтернет-магазину, включаючи головну, каталог, кошик, особистий кабінет, сторінки входу чи реєстрації та оформлення замовлення, а також адміністративну панель.

У третьому розділі було представлено опис процесу розробки інтернет-магазину, включаючи архітектуру фронтенду та бекенду, структуру бази даних MongoDB та ключові етапи реалізації основної функціональності. Було наведено схеми моделей даних MongoDB та описано підходи до тестування окремих компонентів та системи в цілому.

Проведений аналіз та розроблений інтернет-магазин демонструють практичне застосування сучасних веб-технологій для створення повноцінної платформи електронної комерції. Результати цієї роботи можуть слугувати прикладом ефективного використання обраного стеку технологій для розробки веб-додатків.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 69 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Перший інтернет магазин. URL: <https://www.vikka.ua/novini/pershij-internet-magazin-v-cherkasax-zyavivsvya-13-rokiv-tomu/> (дата звернення:04.03.2025).
2. Електронний магазин Rozetka. URL: <https://rozetka.com.ua/> (дата звернення: 04.03.2025).
3. Електронний магазин Foxtrot. URL: <https://www.foxtrot.com.ua/> (дата звернення: 04.03.2025).
4. Магазин електронних товарів Allo. URL: <https://allo.ua/> (дата звернення: 05.03.2025).
5. Електронний магазин EpicentrK. URL: <https://epicentrk.ua/> (дата звернення: 05.03.2025).
6. Amazon. URL: <https://www.amazon.com/> (дата звернення: 05.03.2025).
7. AliExpress. URL: https://best.aliexpress.com/?browser_redirect=true (дата звернення: 05.03.2025).
- 8 Zalando. URL: https://en.zalando.de/?_rfl=de (дата звернення: 05.03.2025).
9. MongoDB. URL: <https://www.mongodb.com/> (дата звернення: 29.05.2025).
10. Продажі через мобільні пристрої в Україні зросли на 25% у першому півріччі 2024 року. URL: <https://marketer.ua/ua/sales-through-mobile-devices-in-ukraine-increased-by-25-in-the-first-half-of-2024/> (дата звернення: 02.03.2025) .
11. Пристрої та сервіси для платежів. URL: <https://futuremagazine.io/ua/87-ukrayintsiv-nadayut-perevagu-oplatam-smartfonom/> (дата звернення:05.03.2025).
12. Продажі через пристрої. URL: <https://eba.com.ua/prodazhi-cherez-mobilni-prystroyi-v-ukrayini-zrosly-na-25-u-pershomu-pivrichchi-2024-roku/>(дата звернення: 02.03.2025).
13. Зростання продажів з мобільних девайсів. URL: <https://rau.ua/novuni/prodazhi-mobilni-2024-admitad/> (дата звернення:02.03.2025).
14. Історія інтернет магазинів. URL: <https://nashkraj.ua/uk/blog/istoriya-internet-magazyniv-z-chogo-use-pochalosya/> (дата звернення:03.03.2025.)

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 70 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

15. Syafrizal M. Web-Based SME Online Marketing System (E-Commerce). *International Journal Software Engineering and Computer Science (IJSECS)*. 2021. Vol. 1, No 2. P. 75-79.
16. Wiratama J., & Desanti R. I. Analysis and design of web-based information system for church congregations case study: church BNKP Pewarta. *Ultima InfoSys: Jurnal Ilmu Sistem Informasi*. 2021. Vol. 12, No 2. P. 115-120.
17. Zamil A., Abu-AlSondos I.A., & Salameh A.A. Encouraging consumers to make online purchases using mobile applications, how to keep them in touch with e-services providers? *International Journal of Interactive Mobile Technologies*. 2020. Vol. 14, No 17. P. 56-65.
18. Zamil A., & Salameh A. The effects of reward systems and organizational structure on tacit knowledge sharing. *Management Science Letters*. 2020. Vol. 10, No 10. P. 2229-2236.
19. Zamil A.M. Electronic shopping based on internet motivations and barriers for jordanian customers (field study). *Arab Journal of Administration*. 29(1).
20. Permatahati I., & Muqorobin M. Computer Sales Forecasting System Application Using Web-Based Single Moving Average Method. *International Journal of Computer and Information System*. 2022. Vol. 3, No 2. P. 56-63.
21. Walek B., & Fajmon P. A hybrid recommender system for an online store using a fuzzy expert system. *Expert Systems with Applications*. 2023. 212, 118565.
22. Nurninawati E., Effendy M. Y., & Rianputra, A. M. Web-Based Product Marketing Information System Design at Definier Store. *International Journal of Cyber and IT Service Management*. 2023. Vol. 3, No 1. P. 1-11.
23. Goodman D., *Dynamic HTML: The definitive reference: A comprehensive resource for HTML, CSS, DOM & JavaScript*. Sebastopol, CA : O'Reilly Media, Inc., 200.
24. Shah H., & Soomro, T. R. Node. js challenges in implementation. *Global Journal of Computer Science and Technology*. 2017. Vol. 17, No 2. P. 73-83.

25. Hahn E., *Express in Action: Writing, building, and testing Node. js applications*. Shelter Island, NY : Manning Publications Co., 2016.
26. Saundariya K., Abirami M., Senthil K. R., Prabakaran D., Srimathi B., & Nagarajan G. Webapp service for booking handyman using mongodb, express JS, react JS, node JS. *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*. 2021. P. 180-183.
27. Patil A., Patil M., Tondare S., Naregalkar P., Nigade A., Pawar S., & Morbale, J. Harnessing the MERN stack for scalable E-commerce website design: A full-stack approach with MongoDB, Node. js, Express. js, and React. js. *Journal of Integrated Science and Technology*. 2025. Vol. 13, No 5. P. 1116-1116.
28. Rappl F. *Modern Frontend Development with Node. js: A compendium for modern JavaScript web development within the Node. js ecosystem*. Birmingham : Packt Publishing Ltd., 2022.
29. De Oliveira V. F., Pessoa M. A. D. O., Junqueira F., & Miyagi P. E. SQL and NoSQL Databases in the Context of Industry 4.0. *Machines*. 2021. Vol. 10, № 1. P. 20.
30. Khan W., Kumar T., Zhang C., Raj K., Roy A. M., & Luo B. SQL and NoSQL database software architecture performance analysis and assessments-a systematic literature review. *Big Data and Cognitive Computing*. 2023. Vol. 7, № 2. P. 97.
31. Mendu M., Krishna B., CH S., Mahesh G., & Pallavi J. Development of real time data analytics based web applications using NoSQL databases // *AIP Conference Proceedings*. - 2022. - Vol. 2418, No. 1. - P. 020038.
32. Skanda C., Srivatsa B., & Premananda B. S. Secure Hashing using BCrypt for Cryptographic Applications. *2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon)*, 2022. P. 1-5.
33. Sreehari C. A., & Francis N. Password security using BCrypt // *Proceedings of the National Conference on Emerging Computer Applications (NCECA)-2021*. - 2021. - Vol. 3, No. 1.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 72 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

34. Sari I. P., Azzahrah A., Qathrunada I. F., Lubis N., & Anggraini T. Perancangan sistem absensi pegawai kantoran secara online pada website berbasis HTML dan CSS. *Blend sains jurnal teknik*. 2022. Vol. 1, No 1. P. 8–15..

35. Gor V. Media Queries. *Creating Responsive Websites Using HTML5 and CSS3: A Perfect Reference for Web Designers*. Berkeley, CA: Apress, 2023. P. 105–119..

36. Thakur A., & Dhiman K. Chat Room Using HTML, PHP, CSS, JS, AJAX. arXiv preprint arXiv:2106.14704, 2021. URL: <http://arxiv.org/abs/2106.14704> (дата звернення: 1.05.2025).

37. Mohd T. K., Thompson J., Carmine A., & Reuter G. Comparative Analysis on Various CSS and JavaScript Frameworks. *J. Softw.* - 2022. - Vol. 17, No 6. - P. 282-291.

38. Aryawan I. K. B. M. Development of Lecturer Data Management System Using Asynchronous JavaScript & XML (AJAX) Concepts . *JIKA (Jurnal Informatika)*. - 2022. - Vol. 6, No 2. - P. 212-219.

39. Zhang M., & Meng W. JSISOLATE: lightweight in-browser JavaScript isolation . *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. - 2021. - P. 193-204.

40. Ye G., Tang Z., Tan S. H., Huang S., Fang D., Sun X., ... & Wang Z. Automated conformance testing for JavaScript engines via deep compiler fuzzing . *Proceedings of the 42nd ACM SIGPLAN international conference on programming language design and implementation*. -2021. - P. 435-450.

41. Odeniran Q., Wimmer H., & Rebman C. Node. js or PhP? Determining the better website server backend scripting language . *Issues in Information Systems*. - 2023. - Vol. 24, No 1. - P. 328.

42. Dean J. *Web Programming with HTML, CSS, and JavaScript*. Jones & Bartlett Learning, 2025.

43. Kusumawardani D. M., Astiti S., Fathoni M. Y., Sunardi D., & Fernandez S. Web Dasar Menggunakan HTML, CSS, JS, PHP dan Studi Kasus. PT. Sonpedia Publishing Indonesia, 2023.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 73 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

44. Sadriddinovich J. T., & Muhiddinovna, M. M. WEB PROGRAMMING INFORMATION . *SUSTAINABILITY OF EDUCATION, SOCIO-ECONOMIC SCIENCE THEORY*. 2024. Vol. 2, No 19. P. 232–234.

45. Frain, B. Responsive Web Design with HTML5 and CSS: *Build future-proof responsive websites using the latest HTML5 and CSS techniques*. Packt Publishing Ltd., 2022.

46. Lakshmi Y. H., Sangeetha Y., Vyshnavi K. P. S., & Fyzulla S. A website for a consultancy using mern stack . 2023 *3rd International Conference on Smart Data Intelligence (ICSMDI)*. 2023. P. 195–200. IEEE.

47. Lazuardy M. F. S., & Anggraini D. Modern front end web architectures with react. js and next. js . *Research Journal of Advanced Engineering and Science*. – 2022. – Vol. 7, No 1. – P. 132–141.

48. Wilson D., Hassan, S. U., Aljohani N. R., Visvizi A., & Nawaz R. Demonstrating and negotiating the adoption of web design technologies: Cascading Style Sheets and the CSS Zen Garden . *Internet Histories*. 2023. Vol. 7, No 1. P. 27–46.

49. Ai K. H., Funabiki N., Mon K. T., Shwe S. H. M., Kyaw H. H. S., & Lin K. S. A proposal of code modification problem for web client programming using JavaScript . 2021 *Ninth International Symposium on Computing and Networking (CANDAR)*. – 2021. – P. 196–202. IEEE.

50. Dutonde P. D., Mamidwar S. S., Korvate M. S., Bafna S., & Shirbhate D. D. Website developemnt technologies: A review . *Int. J. Res. Appl. Sci. Eng. Technol.* 2022. Vol. 10, No 1. P. 359–366.

51. Prasetyo S. M., Nugroho M. I. P., Putri R. L., & Fauzi O. Pembahasan Mengenai Front-End Web Developer dalam Ruang Lingkup Web Development . *BULLET: Jurnal Multidisiplin Ilmu*. 2022. Vol. 1, No 06. P. 1015–1020.

52 Somasundar A. V. S. S., Chilakarao M., Raju B. R. K., Behera S. K., Ramana C. V., & Sethy P. K. MongoDB integration with Python and Node. js, Express. js . 2024 *Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*. 2024. P. 1–5. IEEE

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 74 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

53. Subbulakshmi N., Begum S. A., Vignesh R. V., & Chandru R. Asynchronous Event Driven Brain Teaser Using Node. js . *2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC)*. 2024. P. 113–118. IEEE.

54. Valladares Poncela A. Analysis, design and implementation of a Node. js Web Application for personal link management using a single link in social networks. 2021.

55. Fiechter A., Minelli R., Nagy C., & Lanza M. Visualizing github issues . 2021 *Working Conference on Software Visualization (VISSOFT)*. 2021. P. 155–159. IEEE.

56. Braga P. H. P., Hébert K., Hudgins E. J., Scott E. R., Edwards B. P., Sánchez Reyes L. L., ... & Crystal-Ornelas R. Not just for programmers: How GitHub can accelerate collaborative and reproducible research in ecology and evolution . *Methods in Ecology and Evolution*. 2023. Vol. 14, No 6. P. 1364–1380.

57. Mare B. S. Perancangan Sistem Informasi Berbasis Web Pada Koperasi Simpan Pinjam Sejahtera Bersama . *Indonesian Journal of Networking and Security (IJNS)*. – 2022. Vol. 11, No 2.

58. Lamothe M., Guéhéneuc Y. G., & Shang W. A systematic review of API evolution literature. *ACM Computing Surveys (CSUR)*. 2021. Vol. 54, No 8. P. 1–36

59. Diyasa G. S. M., Budiwitjaksono G. S., Ma'rufi H. A., & Sampurno I. A. W. Comparative analysis of rest and graphql technology on nodejs-based api development. *Nusantara Science and Technology Proceedings*. 2021. P. 43–52.

60. Qaisa R. S., Putri A., & Maghfirah H. Perancangan Aplikasi ToDo Menggunakan Node. js dan REST API. *The Indonesian Journal of Computer Science*. 2023. Vol. 12, No 6..

61. Azkarin V., Guntara R. G., & Herdiana O. Development of a REST API for Human Resource Information System for Employee Referral Management Domain Using the Express JS Framework and Node. js. *Journal of Scientific Research, Education, and Technology (JSRET)*. 2023. Vol. 2, No 3. P. 1085–1094..

62. Safitri R. K., & Putro H. P. Implementasi REST API untuk Komunikasi Antara ReactJS dan NodeJS (Studi Kasus: Modul Manajemen User Solusi247). *Automata*. 2021. Vol. 2, No 1..

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. 75 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

63. Anh V. Real-time backend architecture using Node. js, Express and Google Cloud Platform. 2021.

64. Arteca E., Harner S., Pradel M., & Tip F. Nessie: Automatically testing javascript apis with asynchronous callbacks. *Proceedings of the 44th International Conference on Software Engineering*. 2022. P. 1494–1505..

65. Quintero L. M. P., Barona D. J. R., León K., & Torres F. C. Desarrollo de un aplicativo móvil con Node. js para la venta de productos agrícolas en MiPymes . *Revista Temario Científico*. 2024. Vol. 4, No 2. P. 1–12.

| | | | | | | |
|-----|------|----------|--------|------|----------------------------|------|
| | | | | | КВРІСТ.2101030.21.01.03 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 76 |

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Андрій ДЕРКАЧ

Співавтор:

Назва: Деркач_Веборієнтована інформаційна система інтернет-магазину електроніки

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:8.3%

Коефіцієнт подібності 2:4.3%

Мікропробіли: 0

Заміна букв: 9

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-05-29 13:16:44.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-05-29

Дата



Доцент Андрій Нічепорук

експерт

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 1.0%

Dictionaries check: en_US, ru_RU, ua_UA. Errors in the documents: 13%

| | | | | |
|---|----------|---------|---------------------------|---------|
| ID: 242415 Title: БКР Веборієнтована інформаційна система інтернет-магазину електроніки Added in a DB: 2025-05-29 Authors: Андрій ДЕРКАЧ Heads: Єлизавета ГНАТЧУК Consultants: Opponents: | Document | | Sum coincidence on the DB | |
| | Symbols | Lexemes | Symbols | Lexemes |
| | 67687 | 1022 | 1932 (3%) | 32 (3%) |

Plagiarism sources

| ID | Description | Plagiarism presence in the document | |
|----|-------------|-------------------------------------|---------|
| | | Symbols | Lexemes |

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник: Деркач Андрій Олексійович

Тема: Веборієнтована інформаційна система інтернет-магазину електроніки

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг дипломної роботи:

Кількість листів креслень 3; кількість сторінок записки 66

1. Короткий зміст роботи та прийнятих рішень Запропонована автоматизація та покращення існуючих рішень функціонування інтернет-магазину електроніки. Розглянуто методи та принципи побудови веборієнтованих інформаційних систем, спрямованих на автоматизацію процесів керування та взаємодії в інтернет-магазинах електронних товарів. Практична значимість отриманих результатів полягає у розробленому програмному забезпеченні веборієнтованої інформаційної системи інтернет-магазину електроніки, що дозволяє забезпечити автоматизацію існуючих рішень.

2. Висновок про відповідність роботи дипломному завданню Дипломна робота відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі подано об'єкт та предмет дослідження, мету, а також характеристику структури роботи.

У межах першого розділу було проведено аналіз сфери електронної комерції. Визначено ключові цілі майбутньої системи такі як ефективне адміністрування каталогу, обробка замовлень, управління клієнтами та аналіз даних. На основі проведеного аналізу сформульовано основні вимоги та визначено завдання для створення ефективної та конкурентоспроможної веборієнтованої інформаційної системи для інтернет-магазину електронних товарів.

У другому розділі було обгрунтовані ключові технологічні рішення, що стало фундаментом для створення інтернет-магазину.

У третьому розділі розглянуто створення інтернет-магазину, починаючи від проєктування архітектури фронтенду та бекенду, через етапи їхньої безпосередньої розробки, і до застосування стратегій тестування. Особливу увагу було приділено реалізації адаптивного дизайну для забезпечення безперебійної роботи на різних пристроях шляхом використання медіа-запитів та бургер-меню для мобільних версій.

4. Позитивні сторони роботи: Розроблена веборієнтована інформаційна система є корисним інструментом для користувачів та власників інтернет-магазинів електроніки. Вона не лише дозволяє автоматизувати процес функціонування, але й підтримує ухвалення рішень щодо оптимізації процесів вибору та доставки електроніки кінцевим користувачам

5. Негативні сторони роботи: В роботі не приділено достатньої уваги огляду існуючих технічних рішень створення веборієнтованих інформаційних систем

6. Оцінка графічного оформлення та пояснювальної записки роботи: —

7. Відгук про роботу в цілому: В загальному робота виконана на достатньому професійному рівні

8. Інші зауваження: —

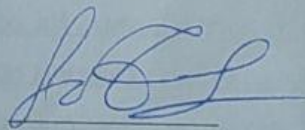
9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «добре» 4,00 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)

Дарина Арнольд Володарівна доцент кафедри ІІІЗ

"2" серпня 2025р.



Завідувачу кафедри КПС
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Андрія ДЕРКАЧА

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи ІСТ-22-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

10.02 2025 року



**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованою системою виявлення текстових збігів/ідентичності/схожості:
Назва: Веборієнтована інформаційна система інтернет-магазину електроніки

Автор: Андрій Деркач

Спеціальність: 126 – Інформаційні системи та технології

Освітня програма: освітньо-професійна Інформаційні системи та технології

Науковий керівник: Єлизавета Гнатчук, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|--|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи. | |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |

Підтвердження:

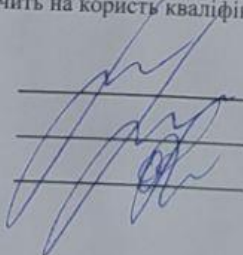
Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
 - 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
 - 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту. (Тут текст можна модифікувати)
- Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 8.3% і адресується до 45 першоджерела; та системою Anti-Plagiarism складає 3%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС



Єлизавета ГНАТЧУК

Єлизавета ГНАТЧУК

Ольга ПАВЛОВА