

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, факультету)

Кафедра комп'ютерних систем та мереж

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавра

(ступінь вищої освіти (освітній ступінь))

на тему **РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ**
ІНТЕРНЕТ-МАГАЗИНУ СПОРТИВНИХ
ТОВАРІВ

Виконав: студент 4 курсу, групи **КНТ-512**
спеціальності _____

123 Комп'ютерна інженерія

(код і найменування спеціальності)

Освітня програма (спеціалізація) _____

Комп'ютерна інженерія

ДЕРЯВКО Д.М.

(ПРИЗВИЩЕ та ініціали)

Керівник **ГОЛУБ Т.В.**

(ПРИЗВИЩЕ та ініціали)

Рецензент **РОМАНОВСЬКИЙ О.В.**

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування вищого навчального закладу)

Факультет Комп'ютерних наук і технологій
 Кафедра Комп'ютерні системи та мережі
 Ступінь вищої освіти бакалаврський
 Спеціальність 123 Комп'ютерна інженерія
 (код і найменування)
 Освітня програма (спеціалізація) Комп'ютерна інженерія
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
 Завідувач кафедри КУДЕРМЕТОВ Р.К.

“ ” 2026 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

ДЕРЯВКУ Данилу Михайловичу
 (ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка комп'ютерної системи інтернет-магазину спортивних товарів

керівник проєкту (роботи) к.т.н., доцент, ГОЛУБ Тетяна Василівна
 (науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом вищого навчального закладу від «14» квітня 2026 року № 160

2. Строк подання студентом проєкту (роботи) 10 червня 2026 року

3. Вихідні дані до проєкту (роботи) Існуючі підходи та технології організації електронної торгівлі в сегменті спортивних товарів. Архітектурні шаблони побудови клієнт-серверних веб застосунків на платформі Node.js 20 LTS з фреймворком Express.js. Розробка клієнтської частини у вигляді односторінкового застосунку (SPA) на бібліотеці React 18 з React Router та React Context API. Реляційна СКБД MySQL 8.0 з механізмом збереження InnoDB та ORM Sequelize. Архітектурний шаблон MVC та принципи побудови REST API. Інтеграція платіжного шлюзу LiqPay для прийому онлайн платежів. Засоби автентифікації на основі JSON Web Token (JWT) з хешуванням паролів bcrypt. Стандарт IEEE 830 для специфікації вимог до програмного забезпечення (SRS). Уніфікована мова моделювання UML для проєктування системи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 1. Аналіз предметної області та сучасного стану ринку онлайн-торгівлі спортивними товарами;

2. Проєктування комп'ютерної системи інтернет-магазину;

3. Розробка комп'ютерної системи інтернет-магазину;

4. Тестування комп'ютерної системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1–4	ГОЛУБ Т.В., к.т.н.		
нормоконтроль	ДЬЯЧУК Т.С., ст. викл.		

7. Дата видачі завдання 01.05.2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз ринку електронної комерції спортивних товарів в Україні та світі, порівняльний аналіз існуючих інтернет–магазинів–аналогів (Rozetka Sport, Intertop, MEGASPORT)	05.03.2026 р.	
2	Формування функціональних та нефункціональних вимог до системи, розробка специфікації SRS згідно з IEEE 830	20.03.2026 р.	
3	Проектування архітектури комп’ютерної системи; обґрунтування вибору клієнт–серверної моделі та шаблону MVC	30.03.2026 р.	
4	Розробка комплексу UML–діаграм (варіантів використання, класів, послідовності, діяльності, компонентів, розгортання)	05.04.2026 р.	
5	Проектування ER–моделі та реалізація реляційної бази даних на MySQL 8.0 з використанням ORM Sequelize	20.04.2026 р.	
6	Розробка серверної частини на Node.js + Express.js (модулі автентифікації JWT, каталогу, кошика, замовлень, особистого кабінету)	01.05.2026 р.	
7	Розробка клієнтської частини як SPA на React 18 з React Router та Context API, реалізація модуля адміністратора	10.05.2026 р.	
8	Інтеграція платіжного шлюзу LiqPay; функціональне, навантажувальне та крос–браузерне тестування системи	15.05.2026 р.	
9	Оформлення пояснювальної записки та графічного матеріалу відповідно до ДСТУ 3008	20.05.2026 р.	

Студент (ка)

(підпис)

Данило ДЕРЯВКО

(Ім’я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис)

Тетяна ГОЛУБ

(Ім’я ПРІЗВИЩЕ)

РЕФЕРАТ

ПЗ: 103 с., 36 рис., 20 табл., 30 джерел, 2 додатки.

ІНТЕРНЕТ–МАГАЗИН, ЕЛЕКТРОННА КОМЕРЦІЯ, СПОРТИВНІ ТОВАРИ, КОМП'ЮТЕРНА СИСТЕМА, REACT, NODE.JS, EXPRESS, MYSQL, SEQUELIZE, REST API, JWT, ОДНОСТОПІНКОВИЙ ЗАСТОСУНОК (SPA), MVC, UML, LIQPAY, ВЕБ–ЗАСТОСУНОК

Метою дипломного проекту є розробка комп'ютерної системи інтернет–магазину спортивних товарів, яка забезпечить ефективну взаємодію між покупцями, адміністраторами та підприємством продавцем.

Об'єктом дослідження є процеси організації електронної торгівлі спортивними товарами в умовах розвитку сучасних веб технологій.

У роботі проведено аналіз сучасного стану ринку електронної комерції спортивних товарів в Україні та світі, виконано порівняльний аналіз існуючих інтернет–магазинів–аналогів, сформульовано функціональні та нефункціональні вимоги до системи, розроблено специфікацію вимог SRS згідно зі стандартом IEEE 830, спроектовано клієнт–серверну архітектуру за шаблоном MVC, побудовано комплекс UML–діаграм та спроектовано структуру реляційної бази даних.

Для розробки системи обрано сучасний технологічний стек: серверна частина – Node.js 20 LTS з фреймворком Express.js та ORM Sequelize; клієнтська частина – React 18 з React Router та React Context API; СКБД – MySQL 8.0 з механізмом збереження InnoDB; автентифікація – JSON Web Token (JWT) з хешуванням паролів bcrypt; платіжний шлюз – LiqPay; додаткові бібліотеки – express-validator, multer, sharp, nodemailer, winston.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області та сучасного стану ринку онлайн-торгівлі спортивними товарами.....	10
1.1 Аналіз ринку електронної комерції спортивних товарів в Україні та світі	10
1.2 Огляд та порівняльний аналіз існуючих інтернет–магазинів спортивних товарів.....	14
1.3 Аналіз цільової аудиторії та сценаріїв використання	20
1.4 Формування функціональних та нефункціональних вимог до системи	25
1.5 Висновки за розділом.....	29
2 Проєктування комп'ютерної системи інтернет-магазину.....	31
2.1 Специфікація вимог до програмного забезпечення (SRS)	31
2.2 Вибір та обґрунтування архітектури системи	35
2.3 Діаграма варіантів використання системи	39
2.4 Діаграма класів системи.....	40
2.5 Діаграма послідовності для процесу оформлення замовлення.....	42
2.6 Діаграма діяльності процесу оплати.....	43
2.7 Діаграма компонентів системи.....	45
2.8 Діаграма розгортання системи	47
2.9 Проєктування бази даних.....	48
2.10 Висновки за розділом	52
3 Розробка комп'ютерної системи інтернет-магазину	53
3.1 Обґрунтування вибору технологічного стеку.....	53
3.2 Апаратні вимоги до серверної та клієнтської частин	58
3.3 Розробка серверної частини: REST API та бізнес–логіка.....	60
3.4 Розробка клієнтської частини.....	62
3.5 Реалізація модуля адміністратора.....	68
3.6 Інтеграція платіжної системи	69

3.7 Висновки за розділом.....	71
4 Тестування комп'ютерної системи.....	73
4.1 Методика та види тестування.....	73
4.2 Функціональне тестування та результати	75
4.3 Висновки за розділом 4	79
Висновки.....	81
Перелік джерел посилання	84
Додаток А Слайди презентації	87
Додаток Б Код застосунку	87

ВСТУП

Сучасний етап розвитку інформаційних технологій кардинально трансформував сферу роздрібної торгівлі, перетворивши електронну комерцію на один із ключових каналів продажу товарів повсякденного попиту. Особливо помітні зміни відбуваються в сегменті спортивних товарів, де покупці все частіше обирають онлайн канали через широкий асортимент, можливість порівнювати ціни різних виробників та зручність доставки безпосередньо додому або у відділення поштового оператора [1]. За даними міжнародних аналітичних агентств, світовий ринок електронної комерції спортивних товарів демонструє стабільне зростання на рівні 10–12 відсотків щорічно, а в Україні цей показник перевищує середньосвітові значення на фоні стрімкої цифровізації галузі та змін у споживчій поведінці [2].

Актуальність теми дипломного проєкту зумовлена кількома взаємопов'язаними чинниками. По–перше, зростання інтересу до здорового способу життя, фітнесу, активного відпочинку та аматорського спорту формує сталий ринок збуту спортивного інвентарю, одягу, взуття та аксесуарів. По–друге, традиційні офлайн магазини не завжди здатні забезпечити повний асортимент товарів і конкурентні ціни, що відкриває нішу для спеціалізованих онлайн платформ. По–третє, активне впровадження мобільних пристроїв та сучасних веб технологій створює технологічну базу для розробки високофункціональних, адаптивних і зручних інтернет магазинів. По–четверте, події останніх років, зокрема пандемія COVID–19 та воєнні дії на території України, прискорили перехід споживачів до онлайн покупок, що зробило наявність якісного інтернет магазину обов'язковою умовою конкурентоспроможності для будь–якого підприємства роздрібної торгівлі [3].

Незважаючи на наявність на ринку низки готових платформ електронної комерції, таких як Shopify, WooCommerce, OpenCart та Magento, розробка власної комп'ютерної системи інтернет магазину має суттєві переваги. Власна система забезпечує повний контроль над функціоналом, дизайном, продуктивністю та

можливостями масштабування, дає змогу гнучко інтегрувати специфічні бізнес процеси компанії, уникнути обмежень і додаткових платежів за використання сторонніх рішень, а також забезпечити високий рівень захисту персональних даних користувачів відповідно до вимог чинного законодавства України та європейських стандартів захисту інформації [4].

Метою дипломного проєкту є розробка комп'ютерної системи інтернет магазину спортивних товарів, яка забезпечить ефективну взаємодію між покупцями, адміністраторами та підприємством продавцем, надасть користувачам зручний інтерфейс для пошуку, перегляду та придбання спортивних товарів, а також інструменти для адміністрування каталогу, замовлень і клієнтської бази даних.

Для досягнення поставленої мети необхідно вирішити такі основні задачі:

- провести аналіз сучасного стану ринку електронної комерції спортивних товарів в Україні та світі;
- виконати порівняльний аналіз існуючих інтернет магазинів аналогів та виявити їхні сильні та слабкі сторони;
- сформулювати функціональні та нефункціональні вимоги до розроблюваної системи;
- розробити специфікацію вимог до програмного забезпечення (SRS);
- спроектувати архітектуру комп'ютерної системи на основі сучасних принципів програмної інженерії;
- побудувати комплекс UML діаграм, що описують структуру, поведінку та розгортання системи;
- спроектувати структуру реляційної бази даних для зберігання інформації про товари, замовлення, користувачів та інші сутності предметної області;
- обґрунтувати вибір технологічного стеку для реалізації клієнтської та серверної частин системи;
- реалізувати програмну частину системи, що включає модулі каталогу, кошика, оформлення замовлення, особистого кабінету користувача та панелі адміністратора;

– провести функціональне тестування розробленої системи та оцінити її відповідність встановленим вимогам.

Об'єктом дослідження є процеси організації електронної торгівлі спортивними товарами в умовах розвитку сучасних веб технологій.

Предметом дослідження є методи, технології та інструменти розробки комп'ютерних систем для електронної комерції, зокрема архітектурні рішення, моделі баз даних та підходи до проектування користувацьких інтерфейсів інтернет магазинів спортивних товарів.

Методи дослідження включають системний аналіз предметної області, методи об'єктно орієнтованого проектування з використанням мови моделювання UML, методи реляційного моделювання баз даних, методи розробки програмного забезпечення на основі архітектурного підходу клієнт сервер, а також методи функціонального тестування програмного продукту.

Практичне значення одержаних результатів полягає у тому, що розроблена комп'ютерна система може бути впроваджена у діяльність підприємств роздрібної торгівлі спортивними товарами та сприятиме розширенню каналів збуту, підвищенню рівня обслуговування клієнтів, оптимізації внутрішніх бізнес процесів і зростанню прибутковості підприємства. Окрім того, отримані результати можуть бути використані як методична основа для розробки аналогічних комп'ютерних систем у суміжних сегментах роздрібної торгівлі непродовольчими товарами.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СУЧАСНОГО СТАНУ РИНКУ ОНЛАЙН-ТОРГІВЛІ СПОРТИВНИМИ ТОВАРАМИ

1.1 Аналіз ринку електронної комерції спортивних товарів в Україні та світі

Електронна комерція в галузі спортивних товарів за останнє десятиліття перетворилася на потужний і динамічний сегмент світового ринку роздрібної торгівлі. Стрімкий розвиток інтернет технологій, поширення мобільних пристроїв та зміна споживчих звичок призвели до того, що значна частина покупок спортивного інвентарю, одягу, взуття та аксесуарів сьогодні здійснюється через онлайн канали. Цей процес особливо прискорився під впливом пандемії COVID–19, яка змусила навіть консервативних споживачів освоїти онлайн платформи, а в Україні до пандемійних факторів додалися ще й воєнні дії, що практично виключили можливість традиційного шопінгу у багатьох регіонах та закріпили цифрові звички населення на тривалу перспективу [1].

Згідно з даними міжнародної дослідницької компанії Statista, обсяг світового ринку онлайн продажу спортивних товарів у 2024 році перевищив 109 мільярдів доларів США, а до 2028 року прогнозується його зростання до 145 мільярдів доларів зі складним щорічним темпом зростання (CAGR) близько 7,4 відсотка [5]. Аналогічні прогнози надає компанія Grand View Research, яка оцінює сегмент онлайн торгівлі спортивними товарами як один із найбільш привабливих для інвестицій та технологічних інновацій у сфері електронної торгівлі завдяки високій маржинальності, широкій цільовій аудиторії та сталому попиту, що не схильний до різких сезонних спадів [6]. Окрім кількісного зростання, спостерігається якісна трансформація сегмента: магазини активно впроваджують технології штучного інтелекту для персоналізації пропозицій, доповненої реальності для віртуальної примірки одягу та взуття, а також інтегровану аналітику поведінки користувачів для оптимізації асортиментної політики.

Динаміку зростання ринку електронної комерції спортивних товарів у світі за період з 2019 по 2025 рік ілюструє рисунок 1.1.

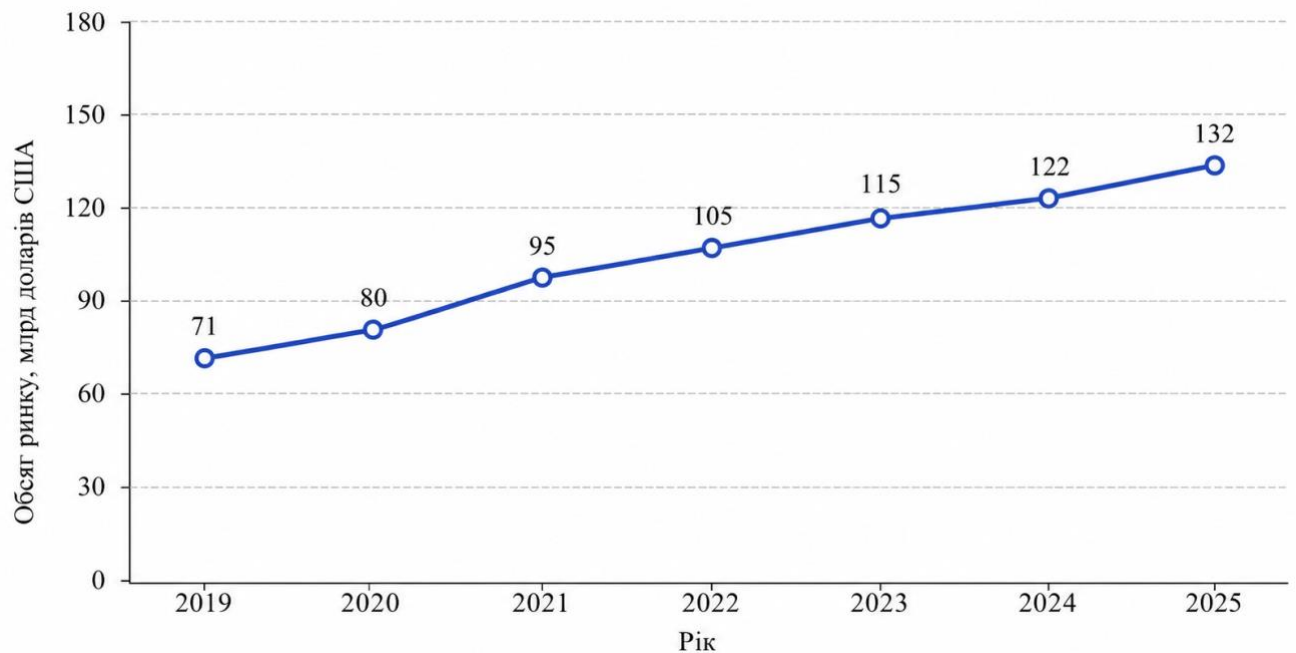


Рисунок 1.1 – Динаміка ринку електронної комерції спортивних товарів у світі

Як видно з графіка, навіть у пост пандемічний період ринок зберігає стійку висхідну динаміку, що підтверджує закріплення нових споживчих звичок та довгостроковий характер цифрової трансформації галузі.

Структура ринку спортивних товарів є неоднорідною і складається з кількох ключових товарних категорій, кожна з яких має власну специфіку як з точки зору виробництва та логістики, так і з точки зору представлення в онлайн каналах. Найбільшу частку традиційно займає сегмент спортивного одягу, який включає тренувальні костюми, футболки, спортивні шорти, термобілизну та компресійний одяг. Далі за обсягом продажів йде сегмент спортивного взуття, що традиційно є флагманським асортиментом більшості провідних спортивних брендів. Окремими великими категоріями виступають спортивний інвентар (гантелі, тренажери, м'ячі, ракетки), різноманітні аксесуари (фітнес браслети, спортивні годинники, пляшки, рушники) та товари для специфічних видів спорту, таких як велоспорт, водні види

спорту та зимові види активного відпочинку. Розподіл часток ринку за основними категоріями товарів представлено на рисунку 1.2.



Рисунок 1.2 – Структура світового ринку спортивних товарів за категоріями

Як видно з діаграми, понад дві третини ринку формують одяг та взуття, що визначає пріоритетні напрями асортиментної політики для більшості інтернет магазинів спортивних товарів. Саме на ці категорії припадає основна маркетингова активність провідних брендів, проводяться найбільші рекламні кампанії та формуються флагманські колекції.

В Україні ринок електронної комерції спортивних товарів демонструє ще більш стрімкі темпи зростання, ніж у середньому по світу, що пояснюється комбінацією кількох факторів: відносно низькою початковою базою порівняно з розвиненими країнами, активною цифровізацією галузі, зростанням penetraції смартфонів та збільшенням довіри українських споживачів до онлайн оплати. Згідно з даними Української асоціації директ маркетингу, обсяг онлайн продажів у даному сегменті у 2023 році перевищив 8,5 мільярда гривень, а у 2024 році, за попередніми оцінками, наблизився до 11 мільярдів гривень [1]. При цьому варто

враховувати, що офіційна статистика часто не відображає повний обсяг ринку, оскільки значна його частина припадає на продажі через соціальні мережі, маркетплейси та невеликих індивідуальних продавців. Динаміка українського ринку наведена у таблиці 1.1.

Таблиця 1.1 – Динаміка ринку електронної комерції спортивних товарів в Україні, 2020–2024 рр.

Рік	Обсяг продажів, млрд грн	Темп зростання, %	Частка онлайн в роздрібних продажах, %
2020	4,2	28,5	14
2021	5,8	38,1	19
2022	6,4	10,3	23
2023	8,5	32,8	27
2024	11,0	29,4	31

Дані таблиці 1.1 свідчать про стабільне зростання обсягів ринку, незважаючи на складну економічну та політичну ситуацію в країні. Особливо показовим є те, що навіть у 2022 році, коли українська економіка пережила безпрецедентний шок внаслідок повномасштабного військового вторгнення, ринок електронної комерції спортивних товарів зміг зберегти позитивну динаміку зростання, нехай і помітно сповільнену порівняно з попередніми роками. У 2023 і 2024 роках темпи зростання знову прискорилися, повернувшись до двозначних значень. Особливо показовим є зростання частки онлайн продажів у загальному обсязі роздрібної торгівлі спортивними товарами, яка за п'ять років подвоїлась і досягла приблизно третини всього сегмента. Це підтверджує стратегічну важливість онлайн каналів збуту для будь якого учасника ринку спортивних товарів та свідчить про формування нової моделі споживчої поведінки, у межах якої онлайн канал стає основним або принаймні рівнозначним з офлайн магазинами.

Серед основних чинників зростання ринку електронної комерції спортивних товарів варто виділити такі: популяризація здорового способу життя та фітнес

культури серед різних вікових груп, що формує сталий попит на спортивні товари; зростання довіри споживачів до онлайн покупок завдяки розвитку служб доставки, безпечних методів оплати та інституту відгуків покупців; розширення асортименту та доступність товарів від міжнародних брендів навіть для жителів невеликих населених пунктів; розвиток мобільної комерції та інтеграція соціальних мереж із торговельними платформами, що дозволяє охопити аудиторію через нові маркетингові канали; вдосконалення логістичних послуг та поява нових варіантів доставки, включно з доставкою до поштоматів та автоматизованих відділень; впровадження сучасних технологій персоналізації та рекомендаційних систем, які підвищують конверсію та середній чек. Усі ці фактори у сукупності створюють сприятливі умови для розробки нових спеціалізованих інтернет магазинів спортивних товарів, здатних запропонувати споживачам якісний цифровий досвід та повноцінну альтернативу традиційному офлайн шопінгу.

1.2 Огляд та порівняльний аналіз існуючих інтернет–магазинів спортивних товарів

Для формування обґрунтованих вимог до розроблюваної комп'ютерної системи інтернет магазину спортивних товарів необхідно провести детальний аналіз існуючих рішень аналогів, що представлені на українському ринку. Такий аналіз дасть змогу виявити їхні сильні та слабкі сторони, визначити стандартні очікування користувачів, сформулювати конкурентні переваги нової системи та уникнути типових помилок, властивих існуючим рішенням. Окрім того, огляд аналогів дозволяє ознайомитися з усталеними патернами проєктування інтерфейсів електронної комерції, які стали де факто стандартом галузі, і які користувачі очікують побачити в будь якому інтернет магазині. Для порівняльного аналізу обрано три інтернет магазини, що представляють різні цінові сегменти та бізнес моделі: Decathlon Ukraine як представника міжнародної мережі з фокусом на

демократичні ціни і власні торгові марки; Sportmaster як приклад мультибрендового магазину з широким асортиментом світових брендів та розвиненою бонусною програмою; Intertop Sport як приклад магазину преміум сегмента з акцентом на брендові кросівки і модні колекції.

Інтернет магазин Decathlon Ukraine (decathlon.ua) є українським представництвом французької мережі магазинів спортивних товарів, заснованої у 1976 році і нині присутньої у понад 70 країнах світу. Магазин позиціонує себе як універсальну платформу для покупки товарів для понад 80 видів спорту за демократичними цінами завдяки розгалуженій системі власних торгових марок (Kalenji для бігу, Domyos для фітнесу, Quechua для туризму, B'TWIN для велоспорту, Tribord для водних видів спорту та інші) [8]. Така стратегія дозволяє Decathlon контролювати весь ланцюжок створення цінності, від проєктування до продажу, і пропонувати споживачам товари за цінами на 20–40 відсотків нижчими, ніж у конкурентів зі схожими характеристиками. Інтерфейс магазину Decathlon представлено на рисунку 1.3.

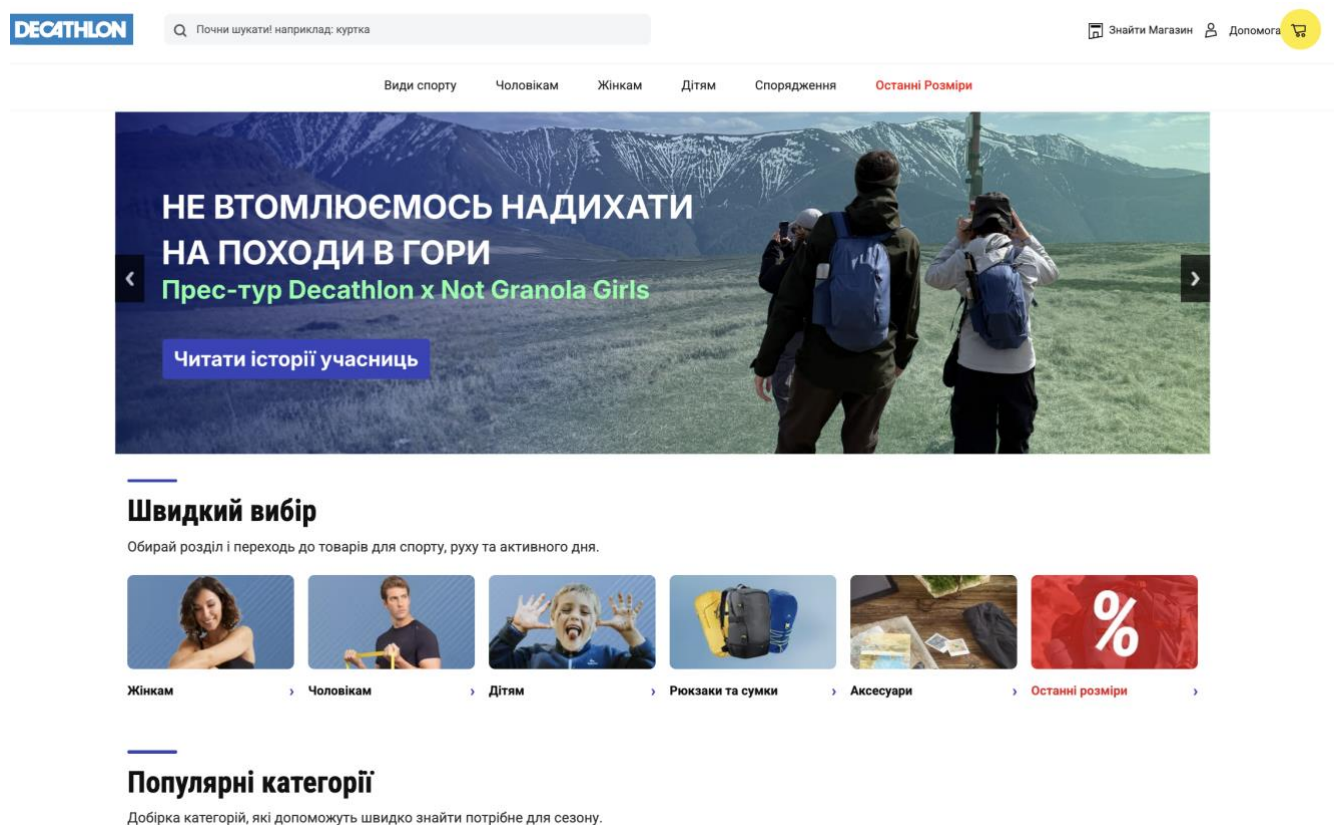


Рисунок 1.3 – Інтерфейс інтернет-магазину Decathlon

Серед переваг Decathlon варто відзначити широкий асортимент за конкурентними цінами, інформативні картки товарів із детальними описами, технічними характеристиками та інструкціями з використання, зручну систему фільтрації за множинними атрибутами, наявність систематизованих відгуків покупців і експертних оглядів, а також можливість самовивозу з мережі фізичних магазинів. До недоліків можна віднести обмежену кількість преміум брендів, дещо лаконічну візуальну стилістику інтерфейсу, що може здаватися застарілою порівняно з конкурентами, та обмежену персоналізацію контенту.

Інтернет магазин Sportmaster (sportmaster.ua) є одним із найбільших мультибрендових інтернет магазинів спортивних товарів в Україні. Платформа відрізняється широким асортиментом світових брендів, серед яких Nike, Adidas, Puma, Reebok, Under Armour, Columbia, Asics, New Balance та десятки інших, розгалуженою бонусною програмою з накопиченням балів та можливістю їх використання при наступних покупках, а також глибокою інтеграцією з мережею фізичних магазинів, які функціонують більш ніж у двадцяти містах України. Користувач може замовити товар онлайн і забрати його у будь якому зручному магазині, скористатися безкоштовним поверненням та обміном, а також отримати консультацію спеціалістів безпосередньо в торговельному залі [9]. Загальний вигляд інтерфейсу Sportmaster наведено на рисунку 1.4.

- 40% на другу річ

3 940 товарів • - 40% на другу річ

1 Фільтри і сортування

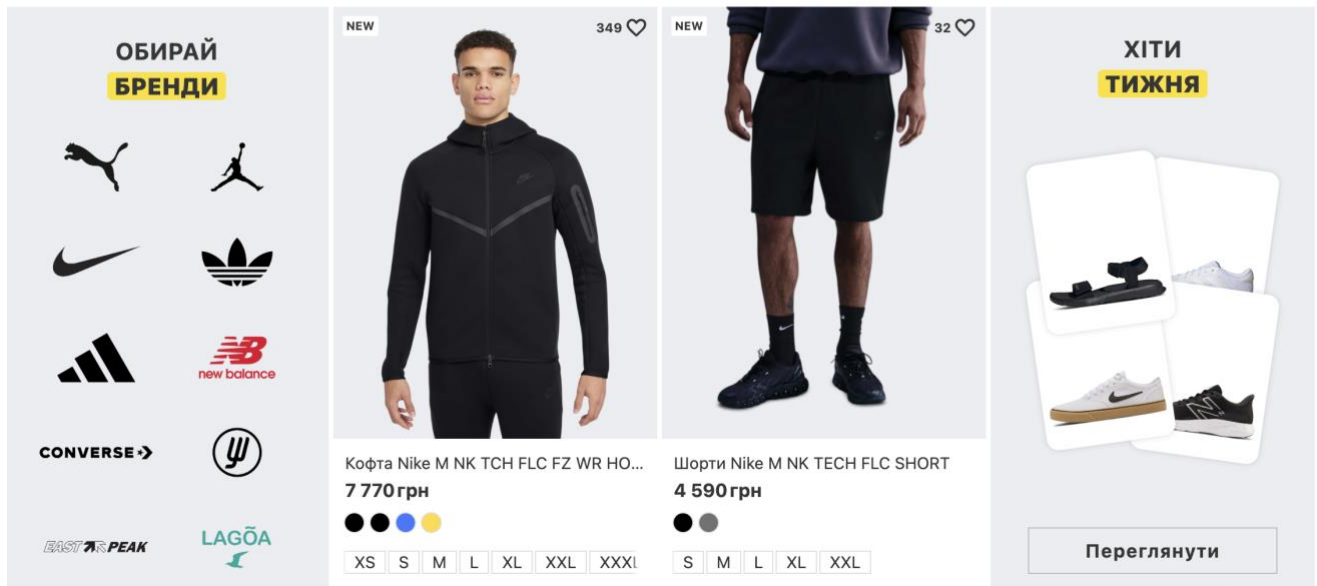


Рисунок 1.4 – Інтерфейс інтернет-магазину Sportmaster

Сильними сторонами Sportmaster є широкий вибір світових брендів, добре розроблена бонусна програма з прозорою системою накопичення балів, наявність детальних відгуків та оцінок покупців на сторінках товарів, інтеграція з фізичними магазинами для самовивозу, а також доволі гнучка політика повернення та обміну товарів. Серед недоліків можна виділити складну ієрархію категорій, що подекуди ускладнює пошук конкретних типів товарів, перевантажений інформацією та банерами інтерфейс, який може здаватися візуально шумним для нових користувачів, та порівняно високі ціни на брендові товари у порівнянні з прямими постачальниками.

Інтернет магазин Intertop Sport (intertop.ua/sport) є спеціалізованим розділом великого українського ритейлера взуття та одягу INTERTOP, який має понад тридцятирічну історію присутності на українському ринку. Спортивний розділ магазину орієнтований переважно на середній та преміум цінові сегменти і акцентує увагу на лімітованих колекціях, ексклюзивних релізах кросівок від провідних брендів та персоналізованих рекомендаціях. Платформа активно

використовує сучасні маркетингові підходи, серед яких email маркетинг із персоналізованими пропозиціями на основі історії переглядів, інтеграція з соціальними мережами та програма лояльності з кеш беком. Інтерфейс Intertop Sport представлено на рисунку 1.5.

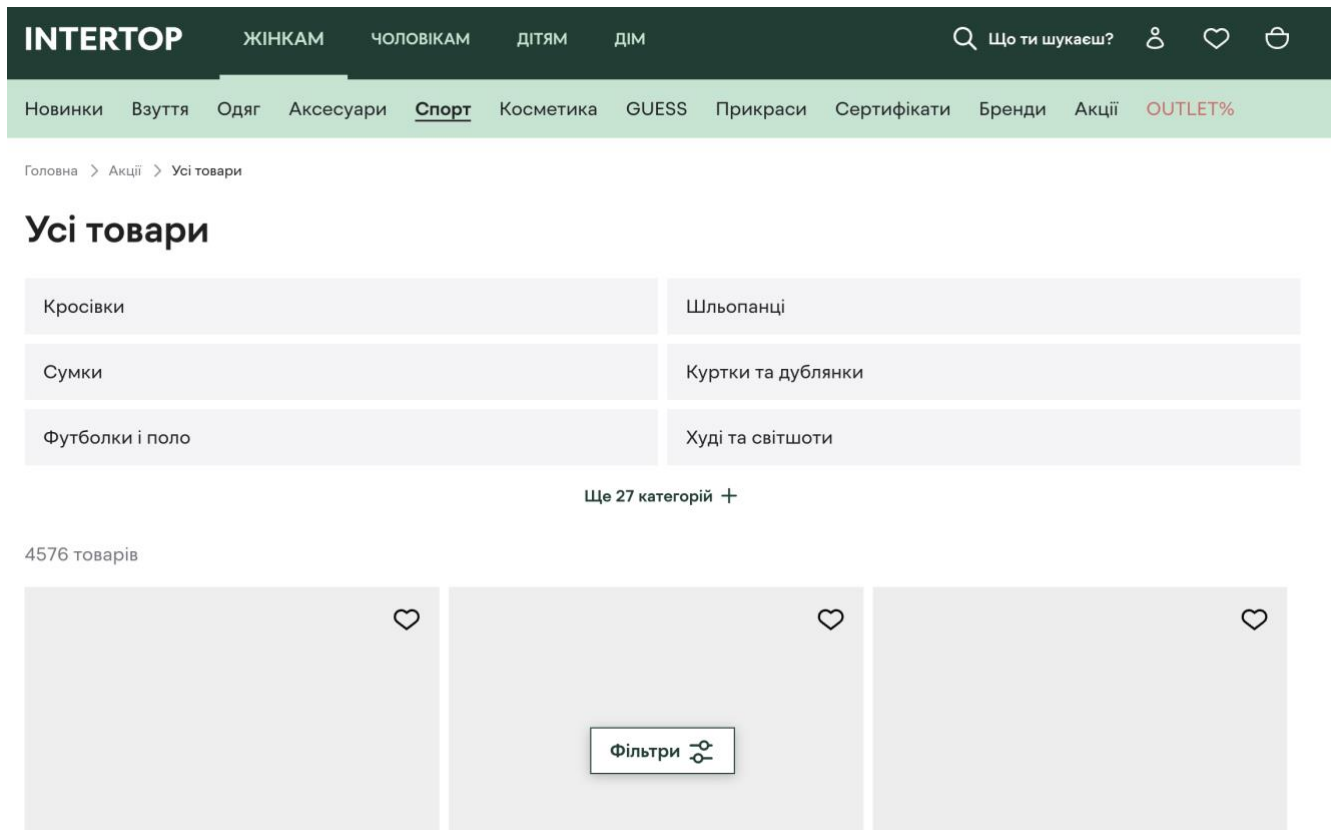


Рисунок 1.5 – Інтерфейс інтернет-магазину Intertop Sport

До переваг Intertop Sport належать сучасний мінімалістичний дизайн із якісною типографікою, акцент на брендових товарах преміум сегмента, висока якість фотоматеріалів із професійною предметною зйомкою, персоналізовані рекомендації на основі історії переглядів та програма лояльності з відчутним кеш беком. До недоліків можна віднести обмежений сегмент бюджетних товарів, орієнтацію переважно на взуттєву категорію в межах розділу Sport, відносно невеликий вибір спортивного інвентарю та аксесуарів і деяку складність орієнтації для нових користувачів, не знайомих із преміум брендами [10].

Узагальнений порівняльний аналіз трьох розглянутих інтернет магазинів за ключовими функціональними та якісними характеристиками наведено у таблиці

1.2. Цей аналіз дає змогу не лише виявити загальні риси сучасних інтернет магазинів спортивних товарів, а й визначити ті аспекти, у яких аналоги мають недоліки і де розроблювана система може запропонувати вищу якість користувацького досвіду.

Таблиця 1.2 – Порівняльний аналіз інтернет–магазинів–аналогів

Характеристика	Decathlon	Sportmaster	Intertop Sport
Широта асортименту	Висока	Висока	Середня
Ціновий сегмент	Бюджетний	Середній	Преміум
Якість фільтрації	Висока	Середня	Висока
Відгуки на товари	Так	Так	Частково
Бонусна програма	Так	Розгалужена	Так
Персоналізовані рекомендації	Базові	Базові	Розвинені
Мобільна версія	Адаптивна	Адаптивна + застосунок	Адаптивна + застосунок
Інтеграція з оф–лайн	Самовивіз	Повна	Самовивіз
Способи оплати	4 варіанти	6 варіантів	5 варіантів
Зручність інтерфейсу	Висока	Середня	Висока

Аналіз даних таблиці 1.2 дає змогу зробити ряд важливих висновків, що мають практичне значення для проєктування розроблюваної комп'ютерної системи. По перше, усі сучасні інтернет магазини спортивних товарів обов'язково забезпечують адаптивний інтерфейс для мобільних пристроїв, що є базовою вимогою з огляду на високу частку мобільного трафіку, яка в українському сегменті e-commerce уже перевищила 60 відсотків від загального обсягу відвідувань. По друге, наявність розвиненої системи фільтрації, інституту відгуків покупців та персоналізованих рекомендацій є ключовими факторами вибору магазину сучасним споживачем, який очікує від цифрового магазину не просто електронного каталогу, а повноцінного помічника у прийнятті рішення про покупку. По третє,

інтеграція онлайн та офлайн каналів (через самовивіз, повернення в магазинах, єдиний обліковий запис) суттєво підвищує лояльність клієнтів та середній чек повторних покупок. По четверте, різноманіття способів оплати, включно з оплатою при отриманні, банківськими картками, інтеграцією Apple Pay і Google Pay, а також підтримкою розстрочки та оплати частинами, є важливим конкурентним фактором, особливо для покупок із чеком понад 2000 гривень.

Виявлені у процесі аналізу аналогів недоліки, такі як перевантажений інтерфейс зі занадто великою кількістю банерів і рекламних елементів, складна багаторівнева навігація, обмежена персоналізація для нових користувачів і відсутність зрозумілого сценарію першого знайомства з магазином, мають бути враховані при розробці нової комп'ютерної системи. Це дозволить забезпечити її конкурентоспроможність на ринку та зручність для кінцевих користувачів, особливо для тих, хто не має значного досвіду онлайн покупок саме у сегменті спортивних товарів.

1.3 Аналіз цільової аудиторії та сценаріїв використання

Цільова аудиторія інтернет магазину спортивних товарів є гетерогенною як за демографічними, так і за поведінковими характеристиками, що обумовлено самою специфікою предметної області. Спортивні товари купують люди різного віку, статі, соціального статусу, рівня доходу та глибини залучення у спортивну активність: від професійних атлетів, які регулярно оновлюють спеціалізований інвентар, до людей, які щойно почали займатися фітнесом і шукають базовий комплект для тренувань. Розуміння структури аудиторії, її потреб, мотивацій та типових сценаріїв взаємодії з онлайн магазином є основою для проєктування зручного користувацького інтерфейсу, формування ефективних функціональних можливостей системи та налаштування правильних маркетингових повідомлень для різних сегментів.

За результатами комплексних досліджень споживчої поведінки на українському ринку електронної комерції, які регулярно проводять компанії GfK Ukraine, Gemius та Української асоціації директ маркетингу, основна частка покупців спортивних товарів припадає на вікову групу 25–44 років. Ця група формує ядро активних користувачів цифрових сервісів, які вирости разом із розвитком інтернет технологій, і одночасно має достатню купівельну спроможність для здійснення регулярних покупок [11]. Не менш важливо, що саме ця вікова категорія найбільш активно займається фітнесом, бігом, велоспортом, командними видами спорту та активними видами відпочинку, що автоматично робить її основним споживачем спортивного асортименту. Розподіл цільової аудиторії інтернет магазинів спортивних товарів за віковими групами наведено на рисунку 1.6.

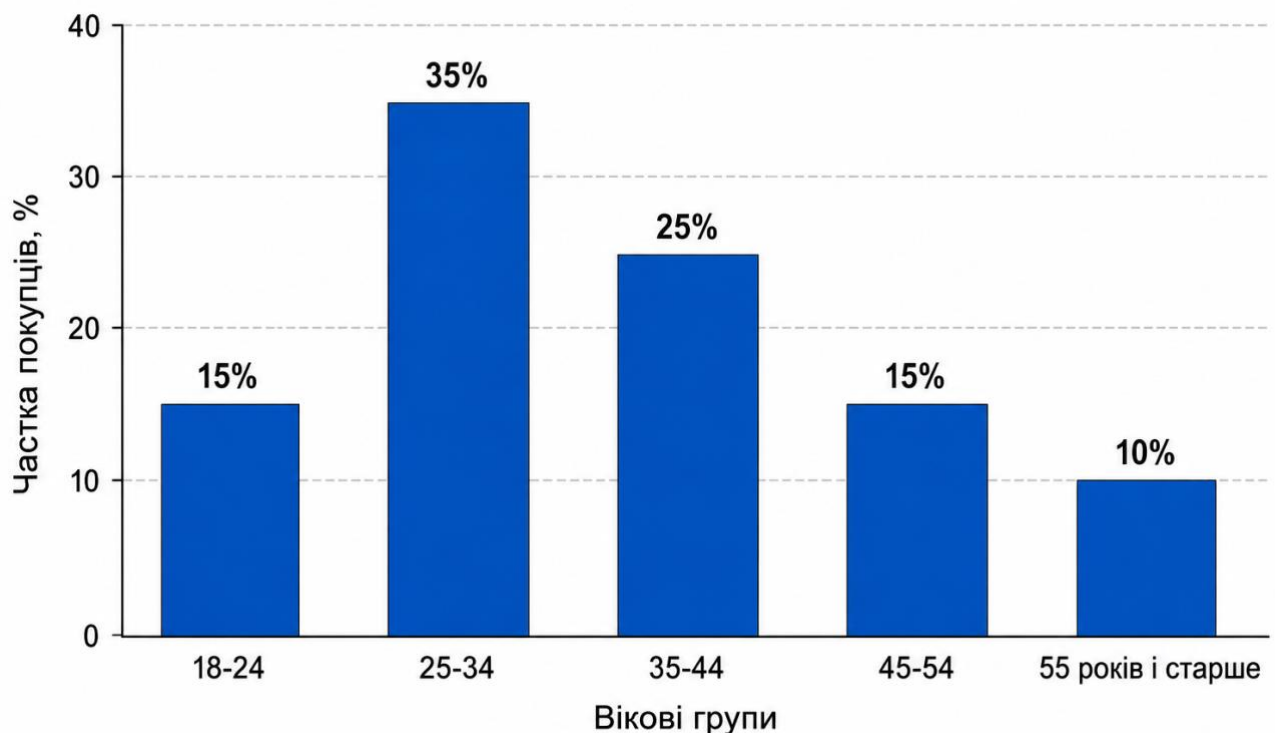


Рисунок 1.6 – Розподіл цільової аудиторії за віковими групами

Як видно з діаграми, дві ключові вікові групи 25–34 та 35–44 років разом формують 60 відсотків аудиторії, що визначає основні стилістичні, технологічні та

маркетингові орієнтири при розробці інтерфейсу інтернет магазину. Інтерфейс повинен бути сучасним і технологічним, але водночас зрозумілим і не перевантаженим, оскільки серед покупців є і молоді користувачі, що очікують динамічних візуальних рішень, і користувачі старшого віку, для яких важливі простота та функціональність.

Окрім вікової сегментації, важливо враховувати поведінкові та психографічні характеристики аудиторії. На основі узагальнення поведінкових патернів виділено декілька ключових портретів користувачів розроблюваної системи, кожен з яких характеризується специфічними мотиваціями, частотою покупок, чутливістю до ціни та переліком функціональних очікувань від магазину. Деталізовані характеристики двох основних портретів цільових користувачів наведено у таблиці 1.3.

Таблиця 1.3 – Портрети цільових користувачів інтернет–магазину

Характеристика	Активний спортсмен	Аматор здорового способу життя
Вік	20–35 років	28–50 років
Частота покупок	Раз на 1–2 місяці	Раз на 3–6 місяців
Середній чек	2000–5000 грн	800–2500 грн
Пріоритети	Якість, бренд, технічні характеристики	Ціна, зручність, базова якість
Категорії інтересів	Професійний інвентар, спеціалізоване взуття	Фітнес одяг, кросівки для бігу, аксесуари
Канал доступу	Десктоп + мобільний	Переважно мобільний
Ключова вимога	Детальні характеристики, відгуки, бонусна програма	Швидкий пошук, прості фільтри, оплата при отриманні

Окрім двох виділених основних портретів, до аудиторії інтернет магазину входять також додаткові сегменти, що потребують спеціальної уваги при проєктуванні системи. Це насамперед батьки, що купують спортивні товари для своїх дітей, передусім шкільну спортивну форму, кросівки для уроків фізкультури, інвентар для секцій та літніх таборів. Для цієї групи критично важливими є чіткі фільтри за віковими групами та розмірами дитячого взуття, прозора розмірна сітка, можливість легкого повернення при невідповідності розміру та доступні ціни. Іншою важливою категорією є подарункові покупці, які шукають товари до тематичних свят, днів народжень, корпоративних подій. Для них доцільно передбачити такі функції як подарункові сертифікати, готові тематичні підбірки (наприклад, для початківця бігу, для тренувань у залі, для туриста), можливість подарункової упаковки та персонального привітання. Також важливо враховувати корпоративних клієнтів, які закупають форму для команд, спортивних шкіл, фітнес центрів і потребують гнучкої системи знижок та можливості замовлення великих партій.

Узагальнений сценарій типової взаємодії користувача з інтернет магазином, починаючи з першого переходу на сайт і до отримання товару та залишення відгуку, представлено на рисунку 1.7. Розуміння цього сценарію є основою для проєктування інтерфейсів кожного з функціональних модулів системи та для забезпечення безперервного користувацького досвіду.

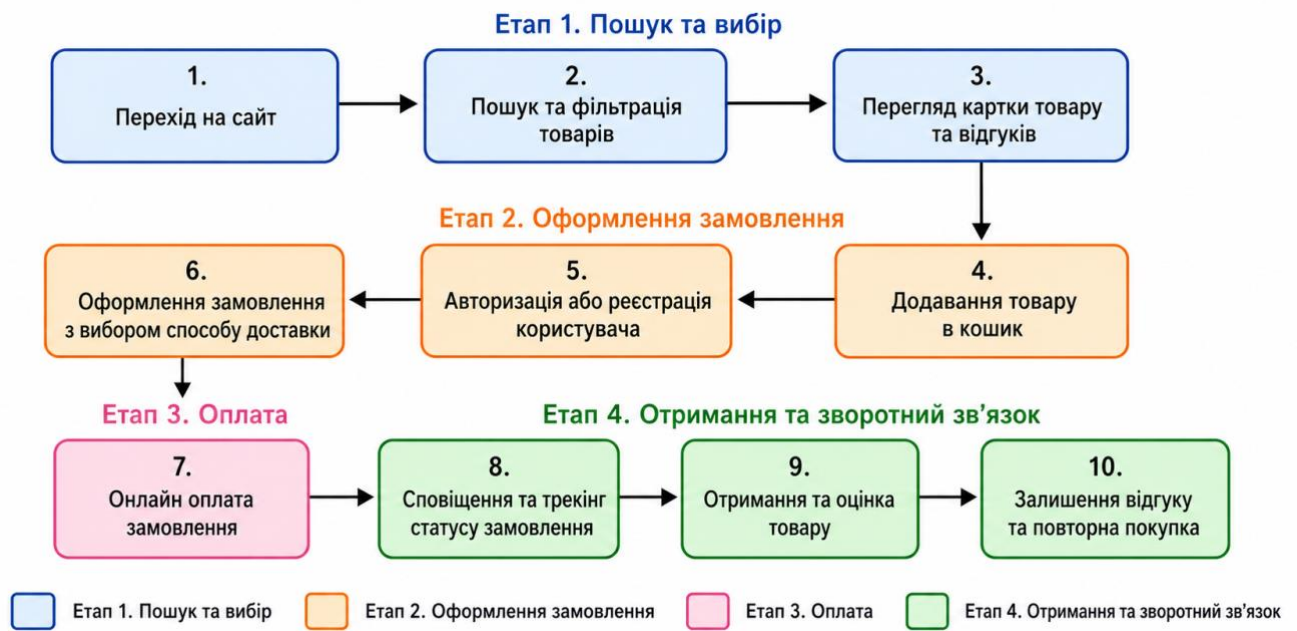


Рисунок 1.7 – Сценарій типової взаємодії користувача з інтернет-магазином

Запропонований сценарій є базовим і узагальненим, але саме він повинен бути врахований при проєктуванні архітектури та інтерфейсів комп'ютерної системи, що розробляється, оскільки він описує природний шлях користувача від першого знайомства з магазином до завершення повного циклу покупки.

Важливо також враховувати, що реальні користувачі часто не проходять цей сценарій лінійно з першої спроби. Типовою є поведінка, коли користувач кілька разів повертається до перегляду карток товарів, додає та видаляє позиції з кошика, порівнює альтернативні варіанти, читає відгуки, перевіряє наявність товару в інших розмірах, а потім залишає сайт і повертається через певний час, можливо вже з іншого пристрою. Тому система повинна бути спроектована з урахуванням асинхронності користувацького досвіду: підтримувати збереження кошика між сесіями, синхронізувати дані між пристроями для авторизованих користувачів, надсилати нагадування про незавершені замовлення та забезпечувати простий механізм продовження з тієї точки, де користувач зупинився.

1.4 Формування функціональних та нефункціональних вимог до системи

На основі проведеного аналізу ринку, конкурентних рішень, цільової аудиторії та типових сценаріїв використання сформовано детальний перелік вимог до розроблюваної комп'ютерної системи інтернет магазину спортивних товарів. Вимоги до програмного забезпечення поділено на дві основні групи: функціональні вимоги, що описують поведінку системи та її можливості з обробки даних, тобто відповідають на запитання що повинна робити система; та нефункціональні вимоги, що визначають якісні характеристики системи такі як продуктивність, безпека, зручність використання, масштабованість та підтримуваність, тобто відповідають на запитання як саме система повинна це робити. Ця класифікація відповідає міжнародному стандарту ISO/IEC 25010:2011, який є базовим орієнтиром при специфікації вимог до сучасних інформаційних систем [12].

Функціональні вимоги формуються з позиції різних типів користувачів системи, серед яких виділено три основні ролі: незареєстрований відвідувач (Гість), якому доступні базові функції перегляду каталогу та пошуку без можливості оформити замовлення; зареєстрований покупець (Користувач), якому доступні всі функції з оформлення замовлень, ведення особистого кабінету та залишення відгуків; Адміністратор магазину, який керує товарним каталогом, замовленнями, користувачами та переглядає статистичну звітність. Такий розподіл ролей є усталеним у галузі електронної комерції та відповідає принципу мінімальних привілеїв із точки зору інформаційної безпеки. Деталізований перелік функціональних вимог до системи з прив'язкою до ролей користувачів наведено у таблиці 1.4.

Таблиця 1.4 – Функціональні вимоги до системи

Код	Назва вимоги	Користувач	Опис
FR–01	Реєстрація	Гість	Можливість створити обліковий запис із вказанням email, пароля та персональних даних
FR–02	Авторизація	Користувач, Адміністратор	Можливість входу в систему з використанням email та пароля або через соціальні мережі
FR–03	Каталог товарів	Гість, Користувач	Перегляд каталогу з розбиттям по категоріях, сортуванням та пагінацією
FR–04	Пошук товарів	Гість, Користувач	Повнотекстовий пошук за назвою, описом, артикулом, брендом
FR–05	Фільтрація	Гість, Користувач	Фільтрація товарів за категорією, ціною, брендом, розміром, кольором, рейтингом
FR–06	Картка товару	Гість, Користувач	Перегляд детальної інформації про товар, фотогалерея, характеристики, відгуки
FR–07	Кошик	Гість, Користувач	Додавання, видалення товарів, зміна кількості, перегляд підсумкової суми

Продовження таблиці 1.4

Код	Назва вимоги	Користувач	Опис
FR–08	Оформлення замовлення	Користувач	Введення адреси доставки, вибір способу доставки та оплати, підтвердження замовлення
FR–09	Онлайн оплата	Користувач	Оплата замовлення банківською карткою через захищений платіжний шлюз
FR–10	Особистий кабінет	Користувач	Перегляд історії замовлень, статусу доставки, редагування персональних даних
FR–11	Відгуки	Користувач	Залишення оцінки та текстового відгуку про куплений товар
FR–12	Список бажань	Користувач	Збереження товарів у список бажань для подальшого перегляду
FR–13	Адмін–панель товарів	Адміністратор	Додавання, редагування, видалення товарів, керування зображеннями та залишками
FR–14	Керування замовленнями	Адміністратор	Перегляд, зміна статусу замовлень, відправлення сповіщень користувачам
FR–15	Статистика та звіти	Адміністратор	Перегляд статистики продажів, популярних товарів, доходів за період

Окрім функціональних вимог, наведених у таблиці 1.4, важливим аспектом проєктування системи є визначення нефункціональних вимог, які встановлюють якісні характеристики та обмеження системи. Нефункціональні вимоги часто визначають загальну сприймальну якість продукту користувачем не менше, ніж функціональні можливості. Користувач може мати в розпорядженні всі необхідні функції, але якщо сторінки відкриваються довго, або інтерфейс некоректно працює на мобільному пристрої, або при пікових навантаженнях система стає недоступною, то такий продукт буде сприйматися негативно і втратить значну частину потенційних клієнтів. Перелік основних нефункціональних вимог до розроблюваної комп'ютерної системи, згрупованих за стандартними категоріями якості програмного забезпечення, наведено у таблиці 1.5.

Таблиця 1.5 – Нефункціональні вимоги до системи

Категорія вимог	Опис
Продуктивність	Час відгуку сторінки не повинен перевищувати 2 секунд за нормального навантаження, час обробки запиту до API не більше 500 мс
Масштабованість	Система повинна підтримувати одночасну роботу не менше 1000 користувачів без втрати продуктивності
Безпека	Шифрування з'єднання HTTPS, хешування паролів, захист від SQL-ін'єкцій та XSS-атак, дотримання GDPR
Надійність	Доступність системи не менше 99,5 відсотка часу, автоматичне резервне копіювання бази даних
Зручність використання	Інтуїтивно зрозумілий інтерфейс, виконання типових операцій не більше ніж за 3 кроки
Адаптивність	Коректне відображення на десктопних та мобільних пристроях з шириною екрана від 320 до 1920 пікселів

Продовження таблиці 1.5

Категорія вимог	Опис
Підтримуваність	Модульна архітектура коду, документація розробника, можливість додавання нових функцій без переписування системи
Сумісність	Підтримка останніх стабільних версій браузерів Chrome, Firefox, Safari, Edge
Локалізація	Інтерфейс українською мовою, можливість додавання інших мов у майбутньому

Сформульовані функціональні та нефункціональні вимоги становлять основу для подальших етапів проєктування комп'ютерної системи інтернет магазину спортивних товарів. Вони визначатимуть архітектурні рішення, технологічний стек, структуру бази даних, склад програмних модулів та критерії приймального тестування готового програмного продукту. Окрім того, наведений перелік вимог може слугувати орієнтиром для подальшого розвитку системи: після першого етапу впровадження частина вимог може бути розширена або уточнена на основі реальних даних про використання системи кінцевими користувачами, що дозволить ітеративно вдосконалювати продукт відповідно до запитів ринку. Усі наведені вимоги детально розглядаються та реалізуються у наступних розділах дипломного проєкту.

1.5 Висновки за розділом

У першому розділі дипломного проєкту здійснено комплексний аналіз предметної області та сучасного стану ринку онлайн торгівлі спортивними товарами, що дозволило сформулювати цілісне уявлення про середовище майбутньої

системи, визначити основні вимоги до неї та обґрунтувати актуальність розробки нової спеціалізованої комп'ютерної системи інтернет магазину.

Проведений аналіз ринку показав, що сегмент електронної комерції спортивних товарів є одним із найбільш динамічних у світовій та українській цифровій економіці зі стабільним щорічним зростанням на рівні 7–12 відсотків. Світовий обсяг ринку у 2024 році перевищив 109 мільярдів доларів США, а український сегмент наблизився до 11 мільярдів гривень. Структура ринку характеризується переважанням сегментів спортивного одягу та взуття, що разом формують близько 70 відсотків загального обсягу продажів і визначають основні пріоритети асортиментної політики магазинів.

Виконано порівняльний аналіз трьох провідних інтернет магазинів аналогів: Decathlon Ukraine, Sportmaster та Intertop Sport, які представляють різні цінові сегменти та бізнес моделі електронної торгівлі спортивними товарами. Виявлено, що сильними сторонами сучасних рішень є широкий асортимент, розвинена система фільтрації, наявність інституту відгуків покупців, бонусних програм та інтеграція з мережами фізичних магазинів. Слабкими сторонами часто стають перевантажений банерами інтерфейс, складна багаторівнева навігація, обмежена персоналізація для нових користувачів та орієнтація переважно на один ціновий сегмент. Ці спостереження стали основою для формування переваг розроблюваної системи.

Проаналізовано цільову аудиторію інтернет магазинів спортивних товарів і встановлено, що її основу формують користувачі вікової групи 25–44 років, які складають приблизно 60 відсотків від загальної кількості покупців. Виділено два ключові портрети користувачів (активний спортсмен та аматор здорового способу життя) і додаткові сегменти (батьки, подарункові покупці, корпоративні клієнти). Узагальнено типовий сценарій взаємодії з системою з десяти послідовних кроків, що став основою для подальшого проєктування інтерфейсу та функціональних модулів системи.

На основі проведеного аналізу сформовано детальний перелік з 15 функціональних та 9 нефункціональних вимог до розроблюваної комп'ютерної

системи. Функціональні вимоги розподілено за трьома ролями користувачів (Гість, Користувач, Адміністратор), що відповідає принципу розмежування прав доступу та забезпечує необхідний рівень інформаційної безпеки. Нефункціональні вимоги охоплюють усі ключові аспекти якості програмного забезпечення відповідно до міжнародних стандартів, включаючи продуктивність, безпеку, надійність, масштабованість та підтримуваність.

Результати, отримані у першому розділі, створюють надійну методологічну та інформаційну основу для подальшого проєктування комп'ютерної системи інтернет магазину спортивних товарів. У другому розділі дипломного проєкту на базі сформованих вимог буде виконано детальне проєктування системи, що включатиме розробку специфікації вимог до програмного забезпечення, обґрунтування архітектурних рішень, побудову комплексу UML діаграм та проєктування структури реляційної бази даних.

2 ПРОЄКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ІНТЕРНЕТ-МАГАЗИНУ

2.1 Специфікація вимог до програмного забезпечення (SRS)

Етап специфікації вимог до програмного забезпечення є одним із ключових у життєвому циклі розробки інформаційної системи, оскільки саме на цій стадії формуються формалізовані та однозначно інтерпретовані описи функцій, обмежень та якісних характеристик майбутньої системи, на основі яких будуть прийматися всі подальші проєктні та технологічні рішення. Документ специфікації вимог до програмного забезпечення (Software Requirements Specification, SRS) розробляється відповідно до міжнародного стандарту IEEE 830, що визначає структуру, зміст та якісні характеристики специфікацій, такі як коректність, однозначність, повнота, узгодженість, верифікованість, модифікованість та відстежуваність [13]. Дотримання вимог цього стандарту дає змогу мінімізувати ризики невідповідності розробленого продукту початковим очікуванням замовника та забезпечити

можливість об'єктивної перевірки реалізації кожної вимоги на етапах тестування та приймальної задачі.

Загальне призначення розроблюваної комп'ютерної системи полягає в автоматизації процесів електронної торгівлі спортивними товарами, забезпеченні зручної взаємодії між покупцями та інтернет магазином, керуванні товарним каталогом, обробці замовлень, прийомі онлайн платежів та формуванні аналітичної звітності для адміністрації магазину. Система призначена для використання трьома основними категоріями користувачів: незареєстрованими відвідувачами (Гість), які мають доступ до перегляду каталогу та інформаційних сторінок; зареєстрованими покупцями (Користувач), яким доступні повні функціональні можливості з оформлення та оплати замовлень; та адміністраторами магазину, які керують контентом, замовленнями і користувацькою базою. Окрім ролевого розмежування, специфікація вимог враховує особливості різних каналів доступу до системи, оскільки користувачі можуть взаємодіяти з магазином через настільні комп'ютери, ноутбуки, планшетні комп'ютери та смартфони з різноманітними характеристиками екранів та обчислювальної потужності.

Деталізована специфікація основних функціональних вимог до розроблюваної комп'ютерної системи у форматі, наближеному до стандарту IEEE 830, наведена у таблиці 2.1. Кожна вимога має унікальний ідентифікатор, рівень пріоритету (Високий, Середній, Низький), а також короткий опис передумов та результатів виконання.

Таблиця 2.1 – Деталізована специфікація функціональних вимог

Код	Вимога	Пріоритет	Опис передумов та результату
SRS–01	Реєстрація користувача	Високий	Передумова: користувач не зареєстрований. Результат: створено новий обліковий запис у БД

Продовження таблиці 2.1

Код	Вимога	Пріоритет	Опис передумов та результату
SRS–02	Авторизація користувача	Високий	Передумова: користувач має обліковий запис. Результат: користувач отримує JWT–токен сесії
SRS–03	Відновлення пароля	Середній	Передумова: користувач забув пароль. Результат: відправлення посилання на скидання пароля на email
SRS–04	Перегляд каталогу	Високий	Передумова: відсутня. Результат: відображення списку товарів з пагінацією
SRS–05	Пошук товарів	Високий	Передумова: відсутня. Результат: повернення товарів за пошуковим запитом протягом 1 с
SRS–06	Фільтрація товарів	Високий	Передумова: відкритий каталог. Результат: відображення відфільтрованих за атрибутами товарів
SRS–07	Додавання у кошик	Високий	Передумова: товар наявний. Результат: товар додано, лічильник кошика оновлено

Продовження таблиці 2.1

Код	Вимога	Пріоритет	Опис передумов та результату
SRS–08	Оформлення замовлення	Високий	Передумова: користувач авторизований, кошик не порожній. Результат: створено замовлення зі статусом Нове
SRS–09	Онлайн оплата	Високий	Передумова: створено замовлення. Результат: статус замовлення змінено на Оплачено
SRS–10	Залишення відгуку	Середній	Передумова: користувач придбав товар. Результат: відгук збережено у БД та відображено на сторінці товару
SRS–11	Керування товарами	Високий	Передумова: користувач має роль Адміністратор. Результат: CRUD–операції з товарами доступні
SRS–12	Аналітична звітність	Середній	Передумова: користувач має роль Адміністратор. Результат: формування звітів за обраний період

Кожна з наведених у таблиці 2.1 вимог буде деталізована у вигляді сценаріїв використання, діаграм послідовності та діаграм діяльності у відповідних підрозділах поточного розділу. Сформована специфікація становить договірну базу між замовником системи та командою розробки, фіксує очікувані результати кожної функціональної можливості та визначає критерії приймального тестування системи на завершальних етапах розробки. Окрім функціональних вимог, у повноцінному документі SRS також деталізуються нефункціональні вимоги, обмеження

середовища виконання, обмеження мови програмування, формат вхідних та вихідних даних, інтерфейси з зовнішніми системами та інші аспекти, що впливають на проєктні рішення.

2.2 Вибір та обґрунтування архітектури системи

Архітектура комп'ютерної системи має вирішальне значення для забезпечення ефективності, масштабованості, надійності та підтримуваності розроблюваного програмного продукту. Сучасна теорія програмної інженерії розрізняє кілька основних архітектурних підходів до побудови інформаційних систем, кожен із яких має свої характеристики та оптимальні сфери застосування. Серед найпоширеніших архітектурних стилів, що використовуються при побудові інтернет магазинів, можна виділити монолітну архітектуру, мікросервісну архітектуру та класичну клієнт серверну архітектуру з виокремленням рівнів представлення, бізнес логіки та доступу до даних [14]. Вибір конкретного архітектурного рішення повинен ґрунтуватися на детальному аналізі вимог до системи, очікуваного навантаження, бюджету проєкту, складу та компетенцій команди розробки, а також стратегії розвитку продукту в довгостроковій перспективі.

Монолітна архітектура передбачає реалізацію всієї системи у вигляді єдиного програмного блоку, в якому всі компоненти (інтерфейс користувача, бізнес логіка, рівень доступу до даних) тісно інтегровані та виконуються в межах одного процесу. Цей підхід характеризується простотою початкової розробки та розгортання, відносно низькими витратами на інфраструктуру та зручністю відлагодження, оскільки всі компоненти знаходяться в єдиному коді. Однак монолітні системи мають істотні обмеження: будь яка зміна, навіть незначна, вимагає повторного розгортання всієї системи; складно горизонтально масштабувати окремі компоненти системи незалежно одне від одного; зростання

кодової бази з часом призводить до зниження керованості та збільшення часу збирання й тестування.

Мікросервісна архітектура, на противагу монолітній, передбачає розділення системи на множину невеликих, незалежних сервісів, кожен з яких відповідає за конкретну функціональну область та може розроблятися, розгортатися й масштабуватися окремо. Кожен мікросервіс взаємодіє з іншими через стандартизовані протоколи, найчастіше HTTP/REST або асинхронні брокери повідомлень. Перевагами цього підходу є висока гнучкість, можливість незалежного оновлення компонентів, технологічна різноманітність (різні мікросервіси можуть використовувати різні мови програмування та СКБД), а також відмовостійкість, оскільки збій одного сервісу не призводить до повної непрацездатності системи. Серед недоліків варто відзначити істотну складність первинного проєктування, необхідність розв'язання задач розподілених систем (узгодженість даних, мережеві затримки, моніторинг), а також підвищені вимоги до інфраструктури та компетенцій команди розробки. Порівняння монолітної та мікросервісної архітектур схематично представлено на рисунку 2.1.

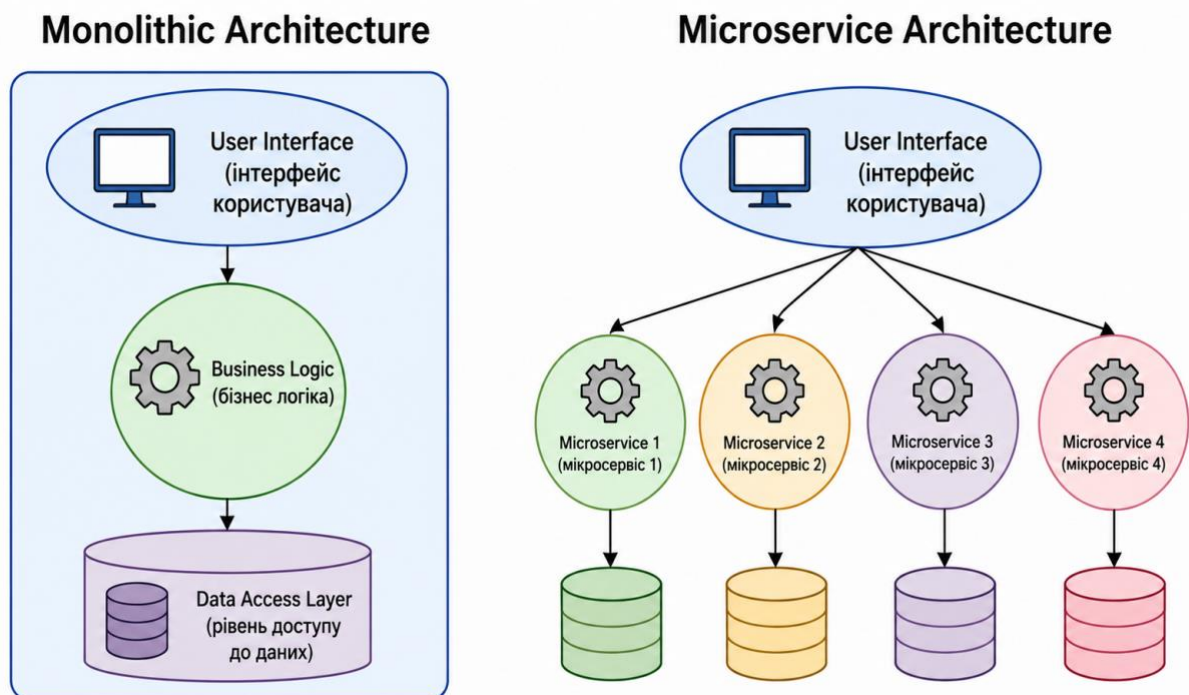


Рисунок 2.1 – Порівняння монолітної та мікросервісної архітектур

Як видно з рисунка, монолітна архітектура має перевагу в простоті та компактності, тоді як мікросервісна забезпечує модульність та незалежність компонентів.

Порівняльна характеристика розглянутих архітектурних підходів за ключовими критеріями наведена у таблиці 2.2.

Таблиця 2.2 – Порівняння архітектурних підходів

Критерій	Монолітна	Мікросервісна	Трирівнева клієнт–сервер
Складність розробки	Низька	Висока	Середня
Масштабованість	Низька	Висока	Середня
Швидкість розгортання	Висока	Низька	Висока
Підтримуваність	Низька	Висока	Висока
Вартість інфраструктури	Низька	Висока	Низька
Відмовостійкість	Низька	Висока	Середня
Вимоги до команди	Базові	Високі	Середні
Підходить для проекту	Частково	Ні	Так

Загальна архітектурна схема розроблюваної комп'ютерної системи інтернет магазину спортивних товарів, побудована за принципами трирівневої клієнт серверної моделі, представлена на рисунку 2.2.

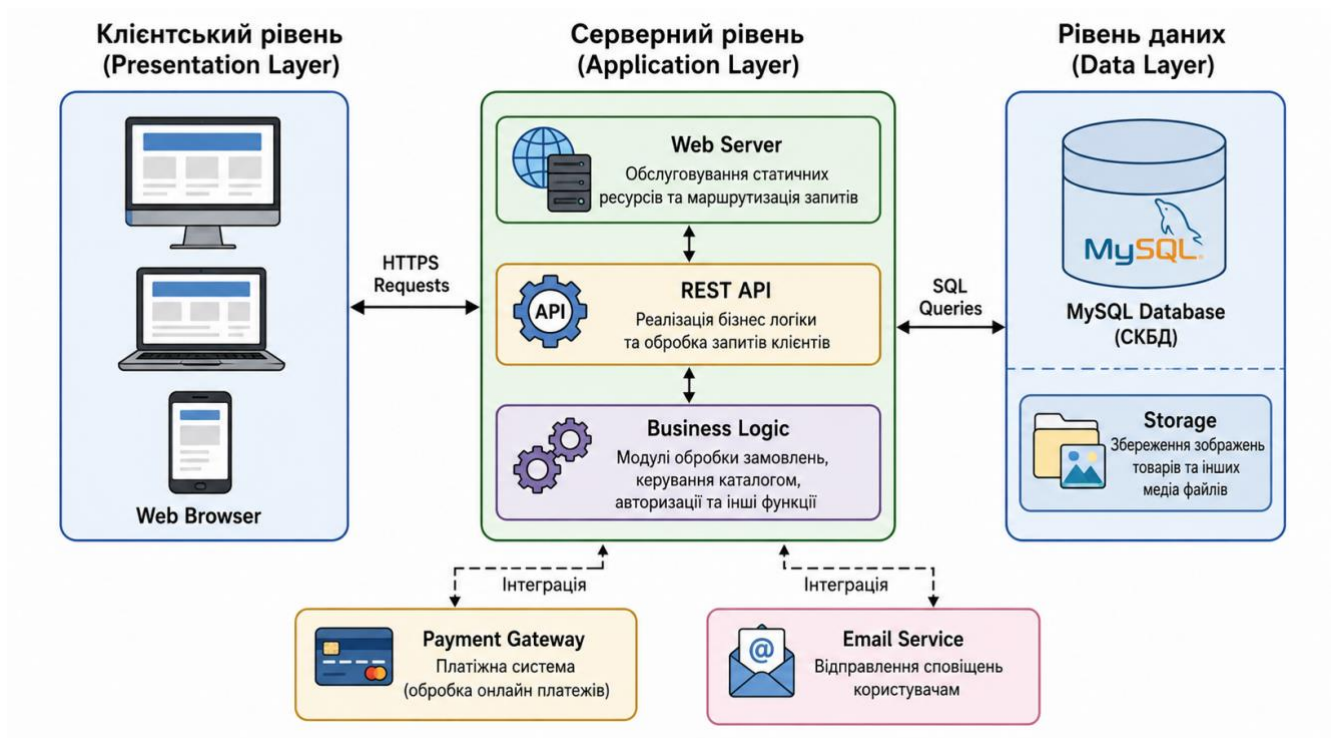


Рисунок 2.2 – Загальна архітектура комп'ютерної системи

Дана архітектура забезпечує чіткий розподіл відповідальності між компонентами, незалежність розробки клієнтської та серверної частин і можливість горизонтального масштабування за рахунок реплікації серверних інстансів за балансувальником навантаження.

Окремою перевагою обраної архітектури є можливість використання сучасного підходу Single Page Application (SPA), при якому клієнтська частина являє собою повноцінний фронтенд застосунок, що завантажується одноразово і потім динамічно обмінюється даними з сервером через REST API. Це дозволяє забезпечити високу швидкість роботи інтерфейсу, плавну навігацію без перезавантаження сторінок і покращений користувацький досвід порівняно з традиційним підходом, при якому кожна сторінка генерується сервером і повністю перезавантажується у браузері. Бекенд серверу при цьому стає простіше масштабувати, оскільки він не повинен генерувати HTML, а лише надавати дані у форматі JSON через API.

2.3 Діаграма варіантів використання системи

Діаграма варіантів використання (Use Case Diagram) є одним із базових елементів моделювання поведінки інформаційних систем у нотації Unified Modeling Language (UML). Призначення цієї діаграми полягає у візуалізації функціональних можливостей системи з точки зору зовнішніх учасників (акторів), а також у демонстрації взаємозв'язків між варіантами використання та їх ієрархічного підпорядкування. Діаграма варіантів використання дає змогу на ранніх етапах проєктування системи зафіксувати її функціональні межі, визначити основних учасників та сценарії їх взаємодії з системою, що є основою для подальшого деталізованого проєктування поведінки системи [15].

Для розроблюваної комп'ютерної системи інтернет магазину спортивних товарів виділено трьох основних акторів. Гість представляє незареєстрованого відвідувача сайту, який має базовий доступ до перегляду інформації про товари, але не може здійснювати покупки. Користувач є зареєстрованим покупцем, який пройшов процедуру авторизації і має повний доступ до функціональності пов'язаної з оформленням та оплатою замовлень. Адміністратор представляє співробітника магазину з розширеними правами доступу до службових функцій керування системою. Окрім зазначених акторів, у діаграмі також відображено зовнішні системи, з якими взаємодіє інтернет магазин, зокрема платіжний шлюз та сервіс електронної пошти. Загальний вигляд діаграми варіантів використання представлено на рисунку 2.3.



Рисунок 2.3 – Діаграма варіантів використання системи

Запропонована діаграма наочно демонструє ролеве розмежування доступу до функціональних можливостей системи та слугує основою для подальшої деталізації окремих сценаріїв через діаграми послідовності та діаграми діяльності.

2.4 Діаграма класів системи

Діаграма класів є структурною діаграмою UML, призначеною для опису статичної структури системи через відображення класів, їх атрибутів, методів та взаємозв'язків між ними. Для об'єктно орієнтованих систем діаграма класів є фундаментальним інструментом проєктування, що визначає базову модель предметної області та слугує основою для подальшої програмної реалізації. Якісно спроектована діаграма класів забезпечує цілісність моделі, мінімізує дублювання

даних, відображає всі ключові бізнес правила та обмеження, а також дає змогу команді розробки мати єдине однозначне розуміння структури системи [16].

Для розроблюваної комп'ютерної системи інтернет магазину спортивних товарів виділено такі основні класи предметної області: User (Користувач) – представляє зареєстрованого користувача системи; Product (Товар) – основна сутність каталогу спортивних товарів; Category (Категорія) – ієрархічна категорія товарів; Brand (Бренд) – виробник або торговельна марка; Order (Замовлення) – сутність оформленого користувачем замовлення; OrderItem (Позиція замовлення) – окрема позиція у складі замовлення; Cart (Кошик) – тимчасова колекція товарів, доданих користувачем; Review (Відгук) – текстовий відгук та оцінка користувача про конкретний товар; Payment (Платіж) – інформація про оплату замовлення через платіжний шлюз. Між класами визначено різноманітні типи зв'язків, серед яких асоціації, агрегації, композиції та узагальнення, що відображають бізнес правила предметної області. Деталізована діаграма класів системи представлена на рисунку 2.4.

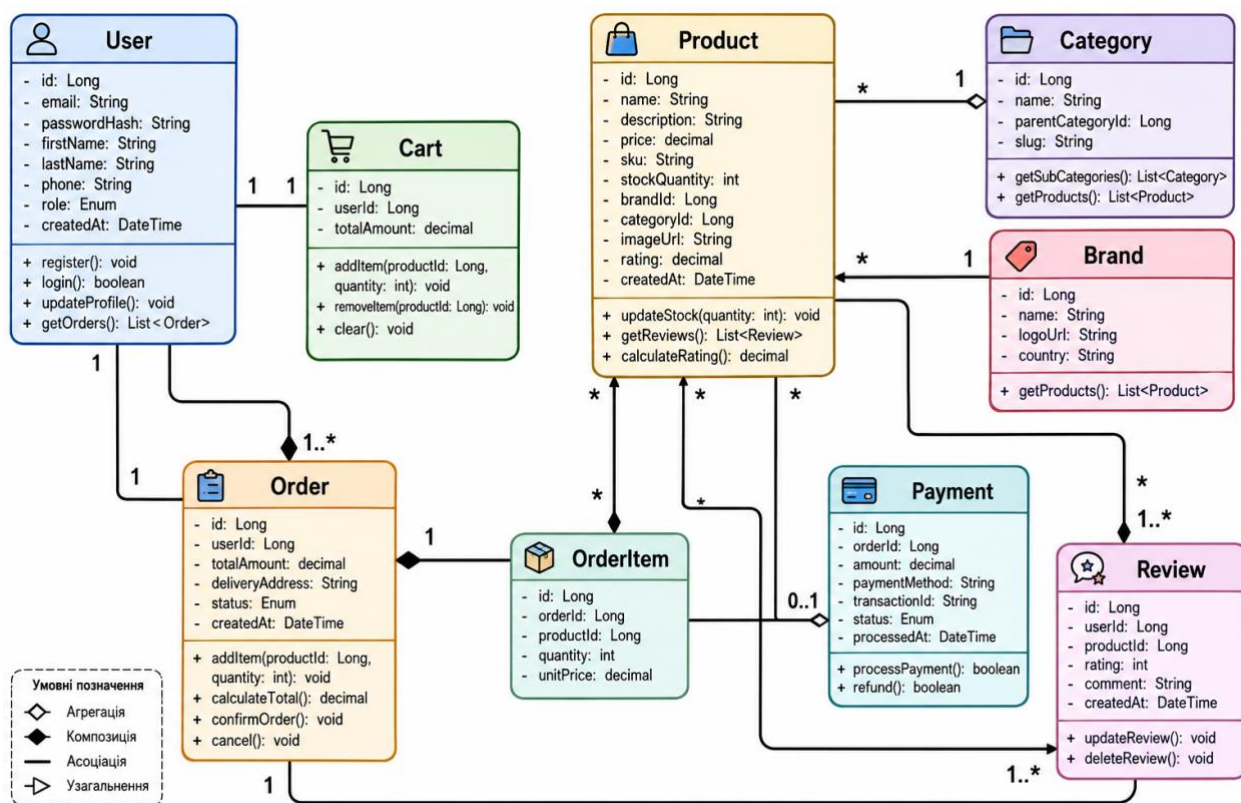


Рисунок 2.4 – Діаграма класів системи

Запропонована діаграма класів повністю відображає предметну область інтернет магазину спортивних товарів і слугує основою для проектування реляційної бази даних та програмної реалізації моделей у мові програмування.

2.5 Діаграма послідовності для процесу оформлення замовлення

Діаграма послідовності (Sequence Diagram) є поведінковою діаграмою UML, призначеною для детального опису сценаріїв взаємодії між об'єктами системи у часовій послідовності. Особливістю діаграми послідовності є явне представлення порядку виклику методів між об'єктами через горизонтальні стрілки, які впорядковано згори вниз вздовж вертикальних ліній життя об'єктів. Цей вид діаграми особливо корисний для деталізації складних бізнес сценаріїв, що включають взаємодію кількох компонентів системи, оскільки дозволяє чітко простежити потік керування та обміну повідомленнями між учасниками сценарію [17].

Для розроблюваної комп'ютерної системи інтернет магазину спортивних товарів найбільш складним і критичним з точки зору бізнесу сценарієм є процес оформлення замовлення, що включає взаємодію між клієнтським застосунком, серверним API, базою даних, платіжним шлюзом та сервісом електронної пошти. Деталізована діаграма послідовності цього процесу наведена на рисунку 2.5.

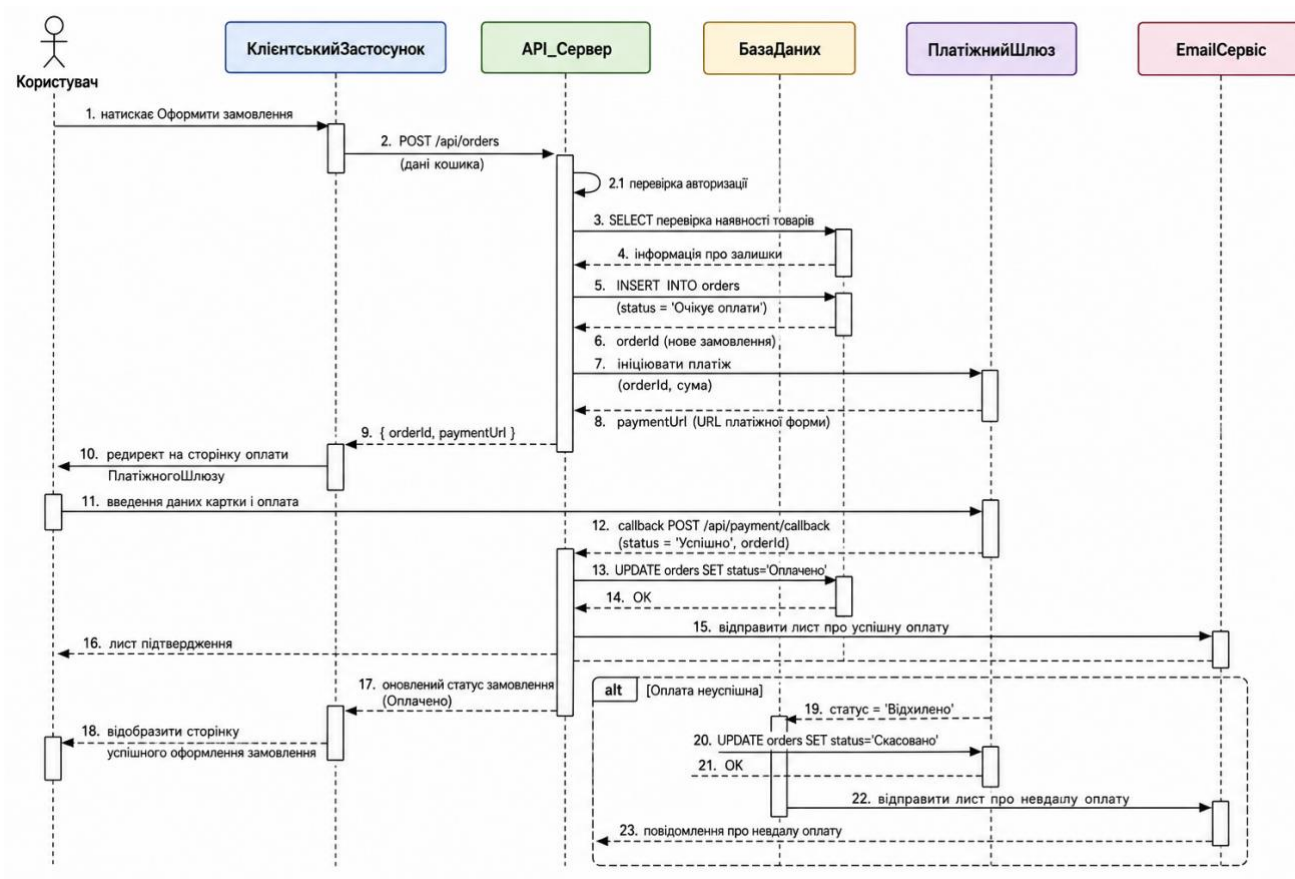


Рисунок 2.5 – Діаграма послідовності процесу оформлення замовлення

Діаграма послідовності на рисунку 2.5 дає змогу однозначно зрозуміти усі етапи обробки замовлення і слугує основою для програмної реалізації відповідного функціоналу.

2.6 Діаграма діяльності процесу оплати

Діаграма діяльності (Activity Diagram) у нотації UML призначена для опису бізнес процесів та робочих потоків у вигляді послідовності дій, рішень та паралельних гілок виконання. На відміну від діаграми послідовності, що фокусується на повідомленнях між конкретними об'єктами, діаграма діяльності зосереджується на самих діях та їх логічному порядку, що робить її особливо зручною для документування бізнес процесів та сценаріїв із розгалуженою логікою

прийняття рішень. Діаграма діяльності використовує спеціальні графічні елементи: початковий вузол у вигляді чорного кола, вузли дій у вигляді прямокутників із заокругленими кутами, вузли рішень у вигляді ромбів, синхронізаційні лінії у вигляді товстих горизонтальних відрізків та кінцевий вузол у вигляді чорного кола в колі [18].

Процес онлайн оплати замовлення в інтернет магазині є одним із найбільш критичних з точки зору безпеки і має кілька можливих результатів виконання: успішна оплата, відхилення транзакції банком, помилка введення даних, перевищення часу очікування. Кожен із цих результатів повинен бути коректно оброблений системою з відповідними сповіщеннями користувача та коректним оновленням статусу замовлення. Деталізована діаграма діяльності процесу оплати представлена на рисунку 2.6.

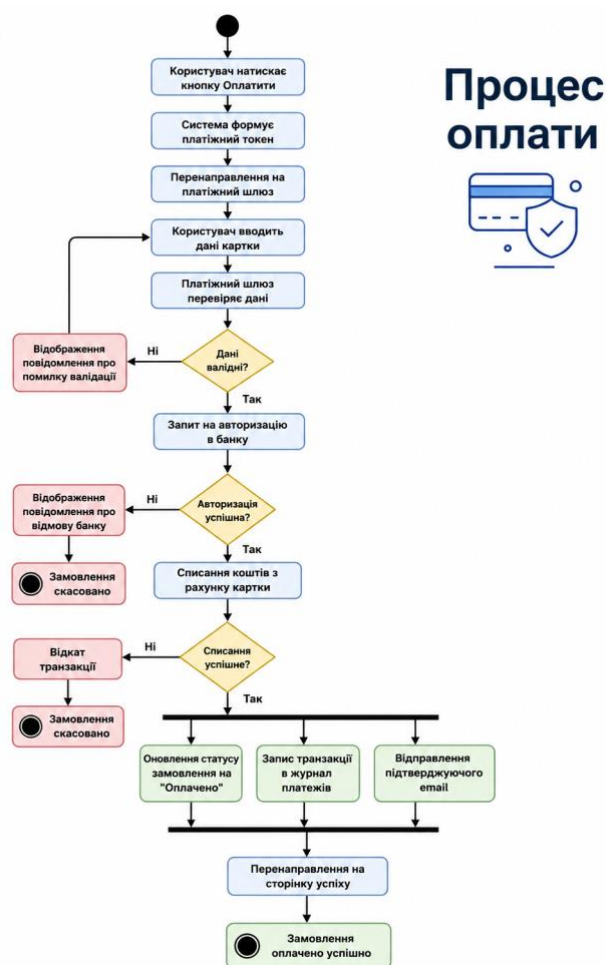


Рисунок 2.6 – Діаграма діяльності процесу оплати

Запропонована діаграма діяльності повністю описує усі можливі сценарії процесу оплати з урахуванням обробки помилок та винятків і слугує специфікацією для програмної реалізації відповідного модуля платіжного процесингу.

2.7 Діаграма компонентів системи

Діаграма компонентів (Component Diagram) у нотації UML призначена для відображення архітектурної структури системи на рівні великих програмних компонентів та модулів, а також їх інтерфейсів та залежностей між ними. На відміну від діаграми класів, що відображає об'єктно орієнтовану структуру коду, діаграма компонентів представляє систему на вищому рівні абстракції, де елементами розгляду є логічно цілісні програмні блоки, які можуть бути реалізовані як окремі бібліотеки, пакети, мікросервіси, динамічно завантажувані модулі тощо. Діаграма компонентів особливо корисна для розуміння архітектури великих систем, планування модульної структури проєкту та виявлення можливостей для повторного використання коду [19].

Для розроблюваної комп'ютерної системи інтернет магазину виділено такі основні програмні компоненти: на стороні клієнта це компоненти інтерфейсу користувача (UI Components), модуль маршрутизації (Router), компонент керування станом (State Management), HTTP клієнт для взаємодії з сервером (API Client); на стороні сервера це шар маршрутизації запитів (Routing Layer), модуль аутентифікації та авторизації (Auth Module), сервісний шар бізнес логіки (Business Services), шар доступу до даних (Data Access Layer), модуль інтеграції з платіжним шлюзом (Payment Integration) та модуль відправлення повідомлень (Notification Module). Розгорнута діаграма компонентів системи зображена на рисунку 2.7.

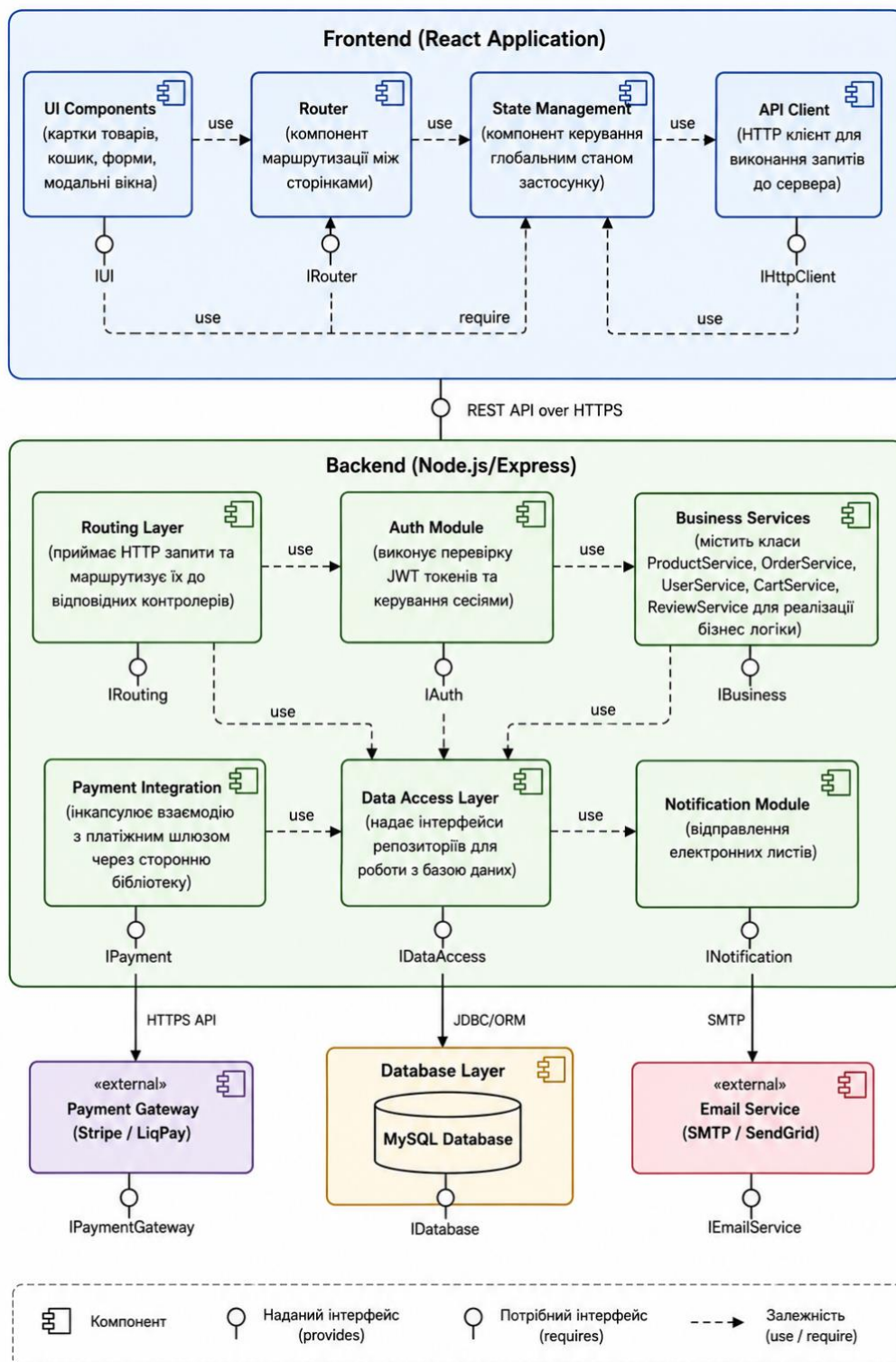


Рисунок 2.7 – Діаграма компонентів системи

Запропонована діаграма компонентів забезпечує цілісне бачення модульної структури системи та полегшує процес планування розробки, розподілу задач між членами команди та подальшої супроводу системи.

2.8 Діаграма розгортання системи

Діаграма розгортання (Deployment Diagram) у нотації UML призначена для опису фізичної архітектури системи на рівні апаратних вузлів та програмних артефактів, що виконуються на цих вузлах. На відміну від діаграми компонентів, що фокусується на логічній структурі коду, діаграма розгортання відображає реальне фізичне середовище виконання системи, включаючи сервери, мережеві з'єднання, контейнери віртуалізації та інші інфраструктурні елементи. Цей вид діаграми є особливо важливим для DevOps інженерів та системних адміністраторів, оскільки слугує основою для планування розгортання системи у виробничому середовищі та для оцінки необхідних апаратних ресурсів [20].

Для розроблюваної комп'ютерної системи передбачається багатовузлове розгортання з виокремленням ролей веб сервера, сервера застосунків, сервера бази даних та зовнішніх сервісів. На клієнтській стороні розгортання здійснюється у браузерях кінцевих користувачів, що працюють на різних типах пристроїв. Структуру фізичного розгортання системи у виробничому середовищі представлено на рисунку 2.8.

Запропонована схема розгортання забезпечує необхідний рівень відмовостійкості за рахунок дублювання серверів застосунків, можливість горизонтального масштабування через додавання нових серверних інстансів, ізоляцію бази даних на окремому вузлі для безпеки та продуктивності і автоматичне резервне копіювання критичних даних.

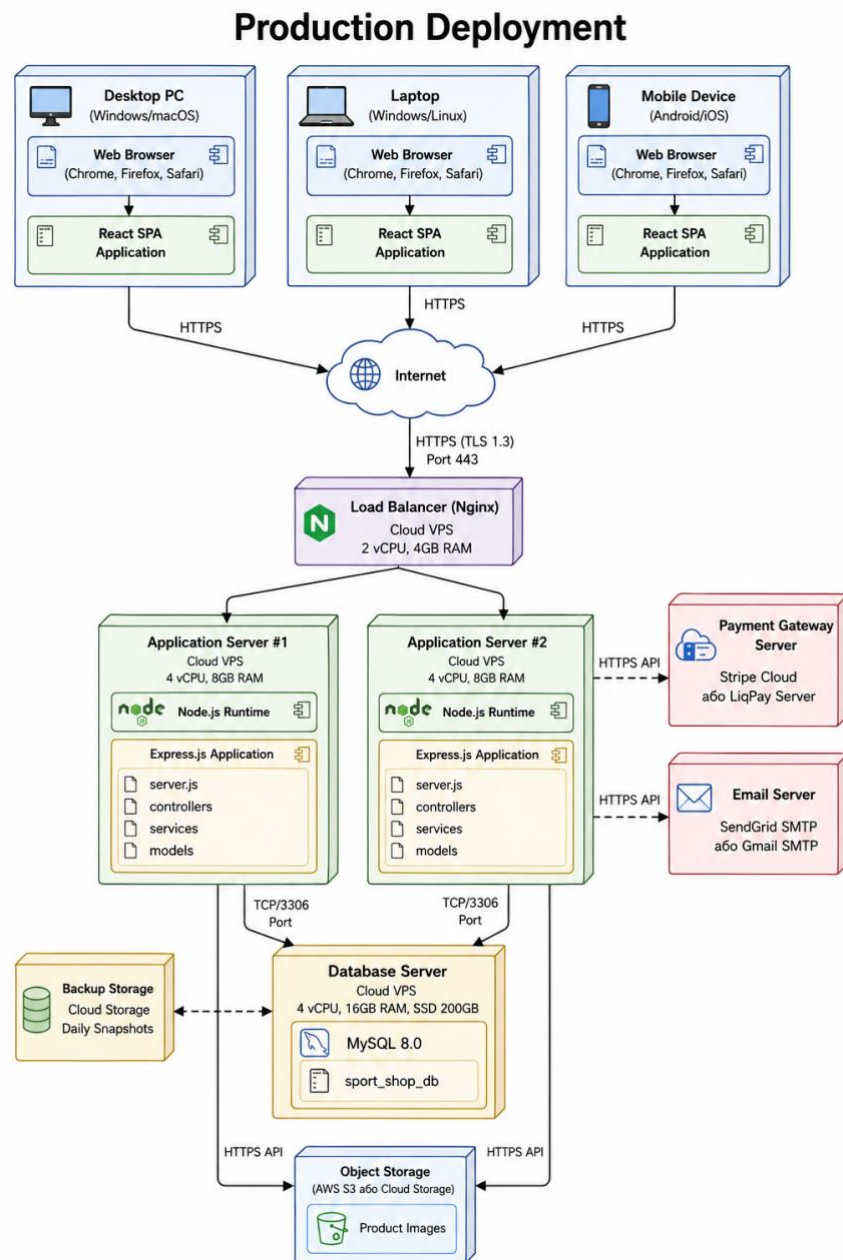


Рисунок 2.8 – Діаграма розгортання системи

2.9 Проєктування бази даних

Проєктування бази даних є одним із найбільш відповідальних етапів створення комп'ютерної системи інтернет магазину, оскільки саме від якості структури даних залежить продуктивність системи, цілісність зберігання інформації, можливість виконання складних аналітичних запитів та

масштабованість на тривалу перспективу. Процес проєктування реляційної бази даних традиційно поділяється на три послідовні етапи: концептуальне проєктування з побудовою моделі сутність зв'язок (Entity Relationship Model), логічне проєктування зі створенням реляційної схеми з визначенням таблиць, атрибутів та зв'язків між ними, фізичне проєктування з визначенням типів даних, індексів, обмежень цілісності та інших оптимізаційних структур [21].

При проєктуванні бази даних для розроблюваної комп'ютерної системи застосовано принципи нормалізації даних до третьої нормальної форми (3НФ), що забезпечує мінімізацію надлишковості даних, виключає аномалії оновлення та видалення, і одночасно зберігає достатню швидкість виконання типових запитів. У ряді випадків навмисно застосовано часткову денормалізацію (наприклад, збереження кешованого середнього рейтингу товару у таблиці Products), що дає змогу оптимізувати продуктивність типових операцій без значного компромісу з цілісністю даних. ER діаграма бази даних системи представлена на рисунку 2.9.

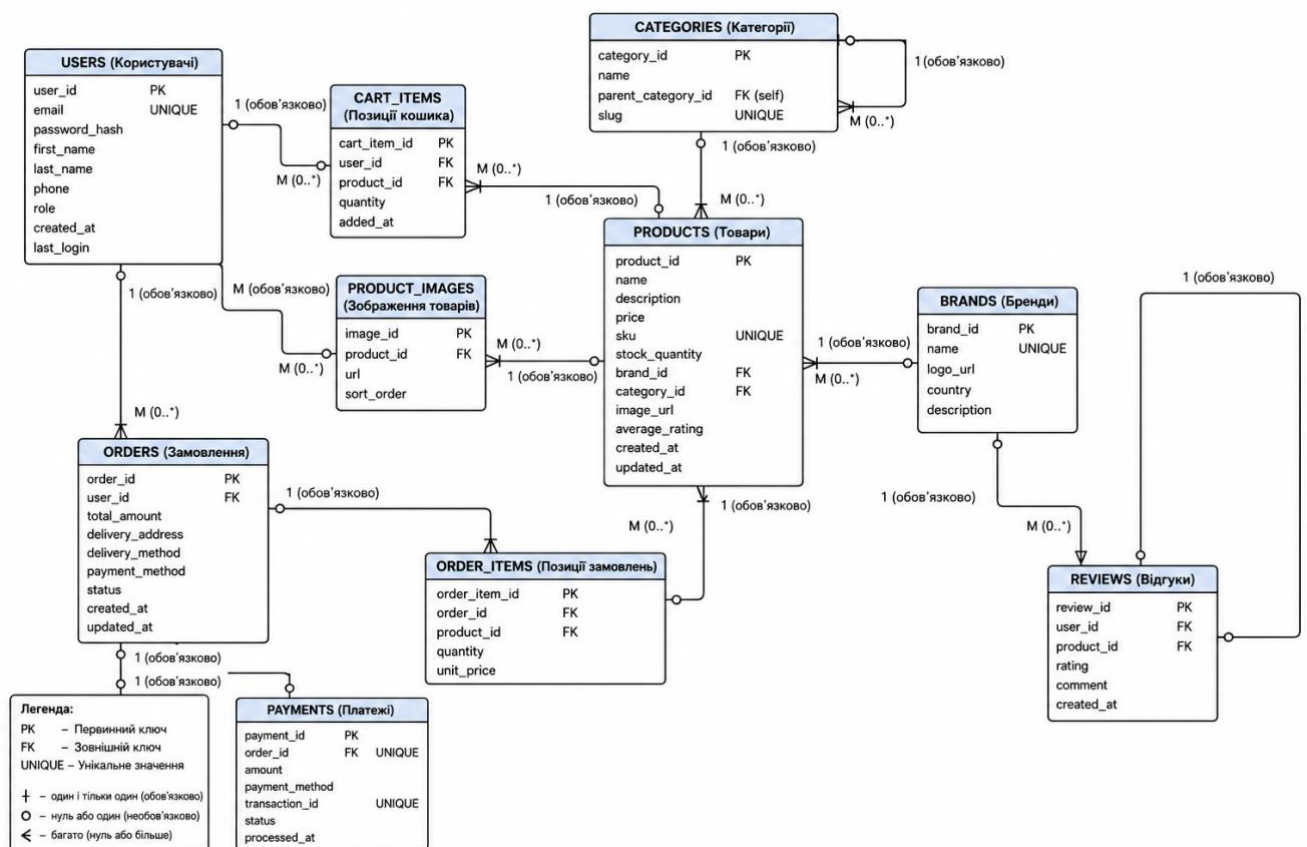


Рисунок 2.9 – ER-діаграма бази даних системи

Запропонована структура повністю відображає предметну область і забезпечує можливість ефективного виконання усіх передбачених функціональних вимог.

Окрім графічного представлення ER діаграми, важливою частиною проектування є детальний опис сутностей бази даних з вказанням типів даних, обмежень цілісності та призначення кожного атрибута. Перелік основних таблиць бази даних з коротким описом призначення наведено у таблиці 2.3.

Таблиця 2.3 – Перелік основних таблиць бази даних

Назва таблиці	Кількість полів	Призначення
users	9	Зберігання облікових записів користувачів та адміністраторів
products	12	Зберігання карток товарів каталогу спортивних товарів
categories	4	Ієрархічна структура категорій товарів
brands	5	Зберігання інформації про виробників та торговельні марки
product_images	4	Зберігання посилань на зображення товарів
orders	9	Зберігання замовлень користувачів зі статусами
order_items	5	Позиції в складі замовлень з кількістю та цінами
payments	7	Інформація про платіжні транзакції замовлень
reviews	6	Відгуки та оцінки користувачів про товари
cart_items	5	Поточні позиції кошика користувача
wishlist_items	4	Список бажань користувача (відкладені товари)

Для прикладу детального опису структури таблиці у таблиці 2.4 наведено опис ключової таблиці products, що зберігає інформацію про товари каталогу. Аналогічним чином виконано опис усіх інших таблиць бази даних, які розміщено у додатках до дипломного проєкту.

Таблиця 2.4 – Опис таблиці products

Назва поля	Тип даних	Обмеження	Опис
product_id	BIGINT UNSIGNED	PK, AI	Унікальний ідентифікатор товару
name	VARCHAR(255)	NOT NULL	Назва товару
description	TEXT	NULL	Детальний опис товару
price	DECIMAL(10,2)	NOT NULL	Ціна товару у гривнях
sku	VARCHAR(50)	UNIQUE	Артикул товару (Stock Keeping Unit)
stock_quantity	INT UNSIGNED	DEFAULT 0	Залишок товару на складі
brand_id	BIGINT UNSIGNED	FK	Посилання на бренд (таблиця brands)
category_id	BIGINT UNSIGNED	FK	Посилання на категорію (таблиця categories)
image_url	VARCHAR(500)	NULL	URL основного зображення товару
average_rating	DECIMAL(2,1)	DEFAULT 0	Середній рейтинг товару (кешоване значення)
created_at	TIMESTAMP	DEFAULT NOW	Дата та час створення запису

Окрім стандартних обмежень цілісності, у структурі бази даних передбачено створення індексів для оптимізації типових запитів: композитний індекс на полях (category_id, price) для прискорення фільтрації товарів за категорією з сортуванням за ціною, повнотекстовий індекс FULLTEXT на полях name та description таблиці products для реалізації повнотекстового пошуку, індекс на полі user_id таблиці orders для швидкого отримання історії замовлень користувача, унікальний індекс на полі transaction_id таблиці payments для запобігання дублювання платіжних транзакцій. Запропонована схема бази даних повністю відповідає сформованим у розділі 1 функціональним вимогам і забезпечує необхідну продуктивність системи при очікуваному навантаженні.

2.10 Висновки за розділом

У другому розділі дипломного проєкту виконано комплексне проєктування комп'ютерної системи інтернет магазину спортивних товарів, що охопило всі ключові аспекти подальшої програмної реалізації системи: специфікацію функціональних вимог, обґрунтування архітектурних рішень, побудову поведінкових та структурних UML діаграм і проєктування реляційної бази даних.

Розроблено формалізовану специфікацію функціональних вимог (SRS) у форматі, наближеному до міжнародного стандарту IEEE 830, що включає 12 деталізованих вимог з визначенням передумов, очікуваних результатів та рівнів пріоритетності. Сформована специфікація становить договірну основу між замовником та розробником і слугує орієнтиром для всіх подальших етапів життєвого циклу системи.

Виконано порівняльний аналіз трьох основних архітектурних підходів (монолітна, мікросервісна, трирівнева клієнт серверна архітектури) та обґрунтовано вибір саме трирівневої клієнт серверної моделі для розроблюваної системи. Цей вибір обумовлений оптимальним балансом між складністю розробки,

продуктивністю, можливістю масштабування та підтримуваністю системи для проєктів масштабу малого і середнього бізнесу.

Побудовано комплекс із восьми UML діаграм, що повністю описують структуру та поведінку системи: діаграма варіантів використання визначає основних акторів та функціональні межі системи; діаграма класів описує об'єктно орієнтовану структуру предметної області з дев'ятьма основними класами та їх зв'язками; діаграма послідовності деталізує сценарій оформлення замовлення з взаємодією шести учасників; діаграма діяльності описує процес онлайн оплати з обробкою винятків та паралельних дій; діаграма компонентів відображає логічну архітектуру системи з 10 основними компонентами; діаграма розгортання описує фізичну архітектуру з виокремленням клієнтських пристроїв, балансувальника навантаження, серверів застосунків, бази даних та зовнішніх сервісів; до загального комплексу також включено архітектурну схему системи та порівняльну схему архітектурних підходів.

Розроблено реляційну модель бази даних, що включає 11 таблиць та повністю відповідає принципам нормалізації до третьої нормальної форми. Структура бази даних відображає всі ключові сутності предметної області та забезпечує можливість ефективного виконання типових запитів за рахунок продуманої системи індексів та зв'язків між таблицями.

3 РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ ІНТЕРНЕТ–МАГАЗИНУ

3.1 Обґрунтування вибору технологічного стеку

Вибір технологічного стеку для реалізації комп'ютерної системи є одним із найбільш важливих рішень на етапі переходу від проєктування до власне розробки програмного продукту. Технологічний стек визначає мови програмування, фреймворки, бібліотеки, системи керування базами даних, інструменти збирання та розгортання, а також робоче середовище кожного компонента системи. Від якості

цього вибору залежать не лише початкова швидкість розробки, але й довгострокова підтримуваність системи, її продуктивність, безпека, можливість пошуку та залучення розробників у майбутньому. У сучасній веб розробці існує велика кількість зрілих технологій для побудови інтернет магазинів, тому вибір повинен ґрунтуватися на комплексному аналізі вимог проєкту, особливостей предметної області, очікуваного навантаження, бюджету розробки та довгострокової стратегії розвитку системи [22].

Для клієнтської частини розроблюваної системи обрано бібліотеку React з екосистемою інструментів, серед яких React Router для маршрутизації між сторінками, Redux Toolkit або Context API для керування станом застосунку, Axios як HTTP клієнт для взаємодії з REST API сервера, та CSS Modules або Tailwind CSS для стилізації компонентів. React було обрано після порівняння з основними альтернативами, такими як Vue.js, Angular та Svelte. Серед основних переваг React варто відзначити найбільше у галузі співтовариство розробників і екосистему сторонніх бібліотек, що дозволяє швидко знаходити готові рішення для типових задач; високу продуктивність завдяки використанню віртуального DOM та інкрементальної моделі оновлення інтерфейсу; компонентний підхід до проєктування інтерфейсу, що сприяє повторному використанню коду та модульності; підтримку TypeScript для статичної типізації; широку доступність кваліфікованих розробників на ринку праці, що важливо для довгострокової підтримки проєкту. Порівняння основних фронтенд фреймворків наведено у таблиці 3.1.

Для серверної частини розроблюваної системи обрано платформу Node.js разом з фреймворком Express.js. Цей вибір обґрунтовано декількома ключовими міркуваннями. По-перше, використання JavaScript як на клієнті, так і на сервері забезпечує єдиний технологічний стек для всієї системи, що спрощує розробку, дозволяє повторно використовувати код між клієнтом та сервером (наприклад, валідатори, типи даних, утиліти) та полегшує підготовку розробників, які можуть працювати з обома частинами системи.

Таблиця 3.1 – Порівняння фронтенд-фреймворків

Критерій	React	Vue.js	Angular	Svelte
Поріг входу	Середній	Низький	Високий	Низький
Розмір екосистеми	Дуже велика	Велика	Велика	Невелика
Продуктивність	Висока	Висока	Середня	Дуже висока
Доступність кадрів	Дуже висока	Середня	Висока	Низька
Підтримка TypeScript	Повна	Повна	Нативна	Часткова
Розмір бандла	Середній	Малий	Великий	Дуже малий
Підходить для проєкту	Так	Можливо	Ні	Ні

По-друге, Node.js характеризується високою продуктивністю при обробці типових для веб застосунків задач завдяки асинхронній неблокуючій моделі введення виведення на основі циклу подій (event loop), що дає змогу ефективно обслуговувати велику кількість одночасних з'єднань без створення окремих потоків для кожного запиту. По-третє, Express.js є мінімалістичним, але потужним фреймворком, який надає всю необхідну функціональність для побудови REST API, при цьому не нав'язуючи жорстку структуру проєкту і дозволяючи гнучко налаштовувати застосунок під специфічні потреби [23].

Для системи керування базами даних обрано реляційну СКБД MySQL у версії 8.0 з механізмом збереження InnoDB. Цей вибір обумовлений тим, що MySQL є однією з найбільш поширених СКБД у світі електронної комерції, має відмінну документацію, велике співтовариство, високу надійність та продуктивність для типових сценаріїв навантаження інтернет магазинів. MySQL надає повну підтримку транзакцій ACID, реляційних зв'язків, тригерів, збережених процедур та представлень, що дозволяє реалізувати всі необхідні бізнес правила на рівні бази

даних. Окрім того, MySQL відмінно інтегрується з Node.js через офіційний драйвер mysql2 та сучасні ORM бібліотеки, такі як Sequelize або Prisma. Порівняння розглянутих варіантів СКБД представлено у таблиці 3.2.

Таблиця 3.2 – Порівняння систем керування базами даних

Критерій	MySQL	PostgreSQL	MongoDB
Тип СКБД	Реляційна	Реляційна	Документна
Підтримка ACID	Повна	Повна	Часткова
Швидкість читання	Висока	Висока	Дуже висока
Складні JOIN запити	Відмінно	Відмінно	Обмежено
Повнотекстовий пошук	Так	Так	Так
Спільнота розробників	Дуже велика	Велика	Велика
Вартість хостингу	Низька	Низька	Середня
Підходить для проєкту	Так	Можливо	Ні

Загальна схема технологічного стеку розроблюваної системи представлена на рисунку 3.1.



Рисунок 3.1 – Технологічний стек комп'ютерної системи

Запропонована схема демонструє повний єдиний JavaScript технологічний стек з чітким розподілом між клієнтською, серверною та базою даних і з мінімізованою кількістю мовних контекстів, що позитивно впливає на продуктивність команди розробки.

Окрім основних технологій, до технологічного стеку входять додаткові інструменти та бібліотеки, які забезпечують різні аспекти функціонування системи. Для роботи з JSON Web Token (JWT) у процесі авторизації використано бібліотеку `jsonwebtoken`, для хешування паролів користувачів обрано `bcrypt` з рекомендованим раундом 10, для валідації вхідних даних використано бібліотеку `express-validator`, для відправлення електронних листів інтегровано бібліотеку `nodemailer` з можливістю використання різних SMTP провайдерів, для логування подій застосовано `winston` з ротацією лог файлів, для роботи з зображеннями товарів обрано бібліотеку `multer` для завантаження файлів та `sharp` для оптимізації розміру та формату зображень. Інтеграція платіжної системи реалізована через офіційний SDK платіжного провайдера `LiqPay` як найбільш популярного на українському ринку, з можливістю заміни на `Stripe` у разі виходу на міжнародний ринок.

3.2 Апаратні вимоги до серверної та клієнтської частин

Визначення апаратних вимог до системи є важливим етапом підготовки до розгортання, оскільки правильно підібрана конфігурація обчислювальних ресурсів забезпечує необхідну продуктивність системи без надмірних витрат на надлишкові потужності. Апаратні вимоги до інтернет магазину спортивних товарів розраховуються виходячи з очікуваного навантаження на систему, профілю поведінки користувачів, обсягу даних, що зберігаються в системі, та цільових показників часу відгуку. Для серверної частини виокремлюються вимоги до сервера застосунків (де виконується Node.js процес) та до сервера бази даних. Для клієнтської частини визначаються мінімальні апаратні характеристики пристроїв кінцевих користувачів, на яких система буде функціонувати з прийнятною швидкодією. Узагальнений перелік апаратних вимог до серверних та клієнтських компонентів розроблюваної комп'ютерної системи наведено у таблиці 3.3.

Таблиця 3.3 – Апаратні вимоги до серверної та клієнтської частин

Параметр	Мінімальні вимоги	Рекомендовані вимоги
Сервер застосунків (CPU)	2 vCPU, 2.0 GHz	4 vCPU, 2.5 GHz
Сервер застосунків (RAM)	4 ГБ	8 ГБ
Сервер застосунків (диск)	50 ГБ SSD	100 ГБ NVMe SSD
Сервер бази даних (CPU)	2 vCPU, 2.4 GHz	4 vCPU, 3.0 GHz

Продовження таблиці 3.3

Параметр	Мінімальні вимоги	Рекомендовані вимоги
Сервер бази даних (RAM)	8 ГБ	16 ГБ
Сервер бази даних (диск)	100 ГБ SSD	200 ГБ NVMe SSD з RAID–10
Пропускна здатність мережі	100 Mbps	1 Gbps
Клієнтський пристрій (CPU)	1.6 GHz, 2 ядра	2.0 GHz, 4 ядра
Клієнтський пристрій (RAM)	2 ГБ	4 ГБ і більше
Браузер користувача	Chrome 90+, Firefox 88+, Safari 14+	Chrome / Firefox останньої версії
Швидкість інтернету (клієнт)	2 Mbps	10 Mbps і більше

Наведені у таблиці 3.3 апаратні вимоги розраховані на підтримку до 1000 одночасних користувачів на серверній частині та забезпечують комфортну роботу з системою практично на будь яких сучасних пристроях кінцевих користувачів, включаючи бюджетні смартфони та комп'ютери середнього класу. У разі прогнозованого зростання навантаження передбачається можливість горизонтального масштабування шляхом додавання нових серверних інстансів за балансувальником навантаження, а також вертикального масштабування за рахунок збільшення обчислювальних ресурсів конкретних серверів. Для розгортання у виробничому середовищі рекомендовано використання хмарних провайдерів (AWS, Google Cloud, DigitalOcean, Hetzner), що забезпечують гнучке керування

ресурсами, автоматичне резервне копіювання та можливість швидкого масштабування у разі пікових навантажень.

3.3 Розробка серверної частини: REST API та бізнес–логіка

Розробка серверної частини комп'ютерної системи інтернет магазину спортивних товарів виконувалася на основі архітектурного шаблону MVC (Model–View–Controller) у його адаптованій для REST API версії, в якій View замінюється на JSON відповіді. Структура проєкту серверної частини відповідає принципам чистої архітектури з чітким розділенням на шари: маршрутизація (routes) приймає HTTP запити та делегує їх відповідним контролерам; контролери (controllers) обробляють вхідні дані, виконують валідацію та викликають сервіси бізнес логіки; сервіси (services) інкапсулюють бізнес правила та використовують моделі для роботи з даними; моделі (models) представляють сутності предметної області та виконують взаємодію з базою даних; middleware містить кросс–функціональні модулі для авторизації, логування, обробки помилок та інших задач, що повторюються в різних місцях системи. Така структура забезпечує модульність коду, спрощує тестування та подальшу підтримку проєкту [23].

Однією з центральних сутностей системи є товар, для якого розроблено модель Product з повним набором атрибутів предметної області та методами для роботи з базою даних. Реалізація моделі Product із використанням ORM Sequelize представлена у лістингу Б.1.

У лістингу Б.1 показано визначення моделі Product із використанням бібліотеки Sequelize, яка слугує об'єктно реляційним маппером (ORM) для роботи з MySQL базою даних. Модель містить визначення полів з вказанням типів даних, обмежень та зв'язків з іншими моделями (Brand, Category), а також два прикладні методи, що інкапсулюють бізнес логіку: метод updateStock для атомарного оновлення залишку товару при оформленні замовлення з перевіркою достатньої

кількості та метод `calculateAverageRating` для перерахунку та кешування середнього рейтингу товару на основі відгуків. Підхід до зберігання обчислюваних значень у вигляді кешованих полів дозволяє суттєво прискорити виконання типових запитів каталогу, де потрібно сортувати товари за рейтингом.

Контролер замовлень обробляє запити на створення нових замовлень користувачами та координує взаємодію між моделями товарів, замовлень та платіжного шлюзу. Скорочений лістинг основного методу контролера замовлень представлений у лістингу Б.2.

Контролер створення замовлення у лістингу Б.2 демонструє низку важливих практик розробки серверної частини. Уся операція виконується в межах єдиної транзакції бази даних, що забезпечує атомарність всіх змін: або всі операції (створення замовлення, додавання позицій, зменшення залишків товарів) будуть успішно завершені, або у разі помилки усі зміни будуть скасовані. Це критично важливо для збереження цілісності даних, особливо у випадках одночасних замовлень одного і того ж товару кількома користувачами. Окрім того, контролер реалізує захист від типових помилок шляхом перевірки наявності товарів перед списанням, повертає змістовні повідомлення про помилки клієнтському застосунку та делегує побічні операції (ініціювання платежу, відправлення сповіщень) спеціалізованим сервісам, що відповідає принципу єдиної відповідальності.

Авторизація користувачів реалізована на основі стандарту JSON Web Token (JWT), що є де факто галузевим стандартом для безстатних REST API. При успішній авторизації сервер генерує підписаний токен, який містить ідентифікатор користувача та його роль, і відправляє його клієнту. Клієнт зберігає токен у локальному сховищі браузера та додає його до заголовка `Authorization` кожного наступного запиту. Сервер при отриманні запиту перевіряє валідність токена через `middleware` і витягує з нього інформацію про користувача. Реалізація `middleware` авторизації представлена у лістингу Б.3.

Наведена у лістингу Б.3 реалізація `middleware` виконує кілька важливих функцій: перевіряє наявність та формат заголовка `Authorization`, верифікує підпис JWT токена за допомогою секретного ключа, що зберігається у змінних

середовища, отримує з бази даних повний об'єкт користувача та зберігає його у об'єкті запиту для подальшого доступу з контролерів. Окремий middleware `authorizeAdmin` використовується для маршрутів, що потребують прав адміністратора, та виконує перевірку ролі користувача. Розділення процесів аутентифікації (хто це) та авторизації (що йому дозволено) є важливою практикою з точки зору архітектури безпеки і дає змогу гнучко налаштовувати доступ до різних маршрутів API.

3.4 Розробка клієнтської частини

Клієнтська частина розроблюваної комп'ютерної системи реалізована як односторінковий застосунок (Single Page Application, SPA) на основі бібліотеки React. Архітектура клієнтського застосунку дотримується сучасних практик функціональних компонентів з використанням React Hooks, що замінили клас-компоненти у нових проєктах. Уся структура застосунку поділена на кілька логічних шарів: компоненти представлення (presentation components), що відповідають лише за візуалізацію та не містять бізнес логіки; контейнери (container components), які зв'язують компоненти представлення з джерелами даних; сервіси API (api services), що інкапсулюють всі HTTP запити до бекенду; глобальні сховища станів (state management) для зберігання даних, спільних для багатьох компонентів, таких як інформація про авторизованого користувача та вміст кошика.

Основою каталогу товарів є компонент `ProductCard`, що відповідає за відображення картки одного товару у списку каталогу. Картка містить зображення товару, назву, бренд, поточну ціну, рейтинг, кнопку додавання до кошика та індикатор наявності на складі. Реалізація компонента `ProductCard` представлена у лістингу Б.4.

У лістингу Б.4 представлено типовий приклад функціонального React-компонента, що демонструє декілька важливих практик розробки клієнтської

частини. Компонент отримує дані про товар через властивості (props), використовує React Context API через хук useContext для доступу до функцій кошика, обробляє локальні події взаємодії користувача (клік на кнопки) та коректно деструктурує дані для зручного доступу. Обгортка Link з бібліотеки React Router дозволяє реалізувати клієнтську навігацію між сторінками без перезавантаження браузера, що є ключовою особливістю Single Page Application. Компонент також обробляє граничні випадки: відображає бейдж відсутності на складі для товарів без залишку та блокує кнопку додавання в кошик у такому випадку, що покращує користувацький досвід.

Керування станом кошика реалізовано через React Context API, що є нативним рішенням для глобального стану у малих та середніх React-застосунках. Контекст кошика синхронізує його вміст із локальним сховищем браузера (localStorage) для збереження товарів між сесіями користувача, а для авторизованих користувачів додатково синхронізується з серверною базою даних. Скорочена реалізація контексту кошика наведена у лістингу Б.5.

Реалізація контексту кошика у лістингу Б.5 використовує хук useState для зберігання списку товарів та хук useEffect для синхронізації стану з localStorage. Метод addToCart реалізує важливу функціональність об'єднання дублікатів: якщо користувач додає товар, який вже знаходиться в кошику, кількість цієї позиції збільшується замість додавання нової позиції. Використання хука useCallback для всіх обробників мутацій стану є оптимізацією, що запобігає непотрібним перерендерам компонентів-споживачів контексту, які отримують ці функції через value. Поле totalItems обчислюється деривативно з items, що дозволяє завжди мати актуальне значення без необхідності його окремо зберігати в стані.

Загальний інтерфейс клієнтської частини інтернет магазину складається з ряду основних сторінок та екранів, що забезпечують повний цикл взаємодії користувача із системою. Загальний вигляд головної сторінки інтернет магазину представлено на рисунку 3.2.

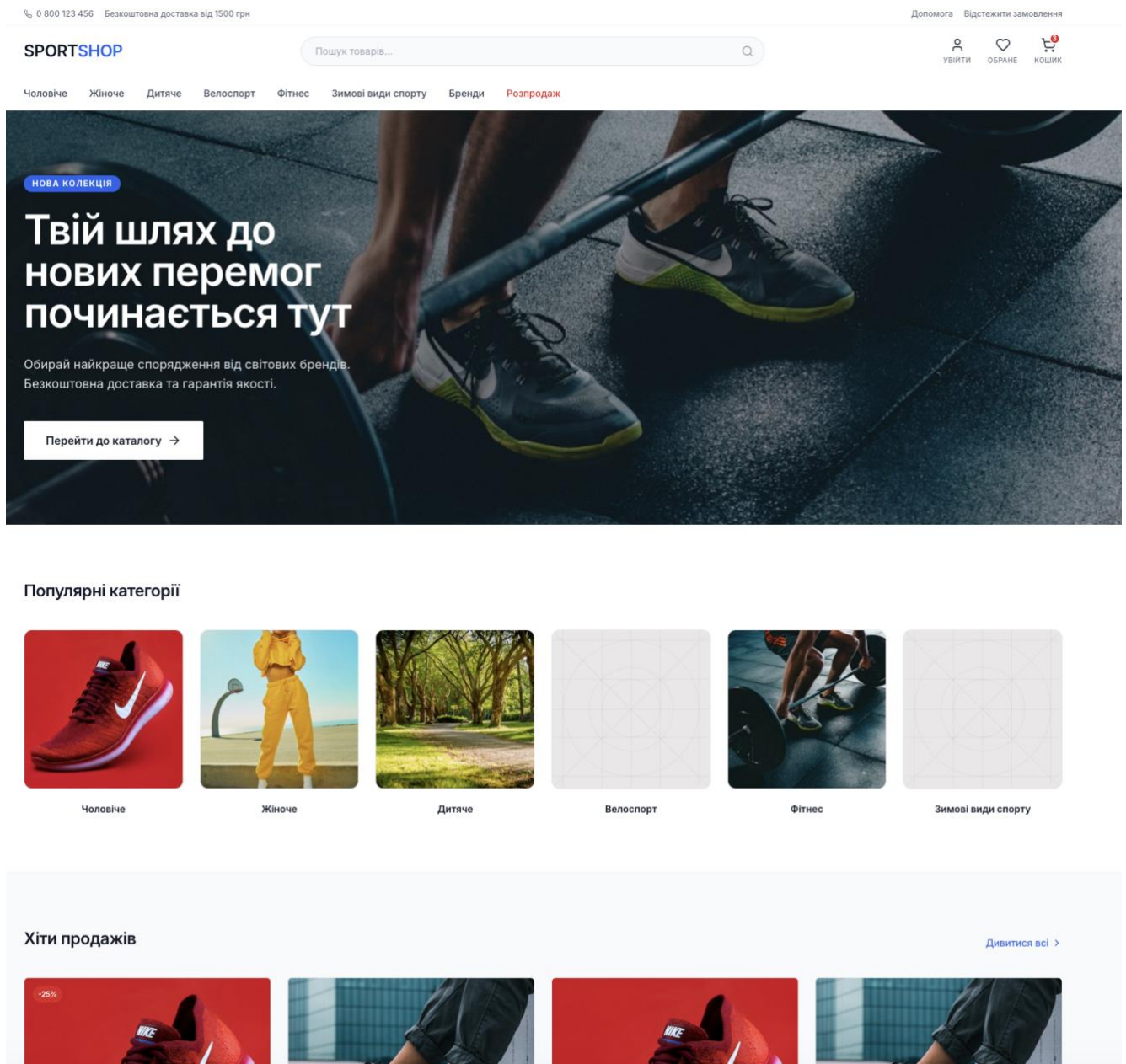


Рисунок 3.2 – Головна сторінка інтернет-магазину

Сторінка каталогу товарів реалізована з підтримкою комплексної фільтрації, сортування та пагінації, що дозволяє користувачам швидко знаходити потрібні товари серед широкого асортименту. Загальний вигляд сторінки каталогу представлено на рисунку 3.3.

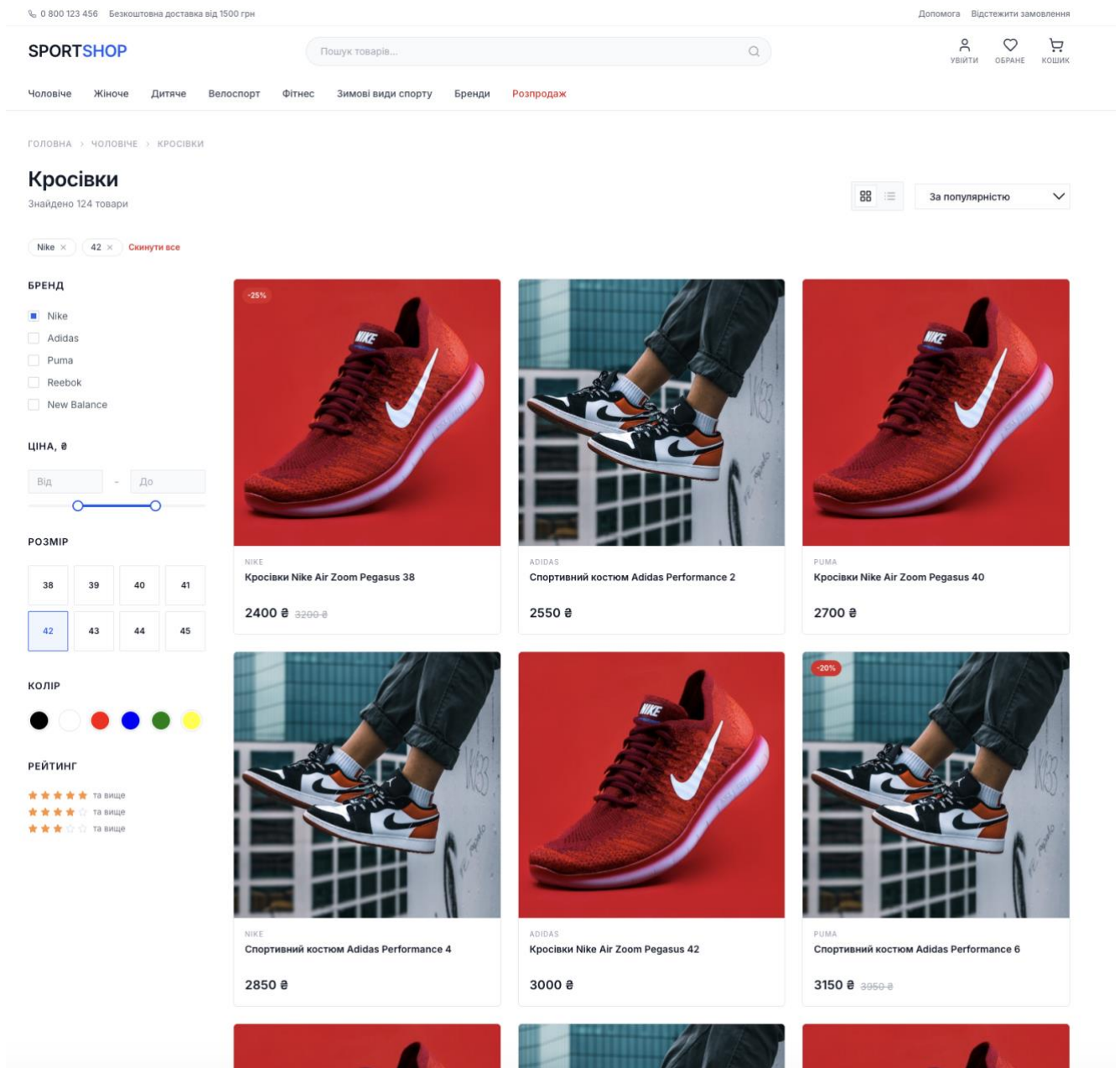


Рисунок 3.3 – Сторінка каталогу товарів

Картка товару (детальна сторінка) є однією з найбільш інформаційно насичених сторінок інтернет магазину, оскільки саме на ній користувач приймає рішення про покупку конкретного товару. Деталізована сторінка картки товару представлена на рисунку 3.4.

0 800 123 456
Безкоштовна доставка від 1500 грн
Допомога
Відстежити замовлення

SPORTSHOP
Пошук товарів...
УВІЙТИ
ОБРАНЕ
КОШИК

Чоловіче
Жіноче
Дитяче
Велоспорт
Фітнес
Зимові види спорту
Бренди
Розпродаж

ГОЛОВНА > ЧОЛОВІЧЕ > КРОСІВКИ > КРОСІВКИ NIKE AIR ZOOM PEGASUS 38



Nike

Кросівки Nike Air Zoom Pegasus 38

★★★★☆ 4.8 (17 відгуків)
Артикул: ST-1000

2400 €
~~3200 €~~

В наявності

ОБЕРІТЬ РОЗМІР (EU)

38	39	40	41
42	43	44	45

- 1 +
Додати в кошик

ДОСТАВКА
Безкоштовно від 1500 €

ПОВЕРНЕННЯ
Протягом 14 днів

Поділитися товаром

Опис
Характеристики
Відгуки (17)

Професійне взуття для бігу по пересіченій місцевості. Покращена амортизація та зчеплення з поверхнею.

Ця модель поєднує в собі класичний дизайн та інноваційні технології амортизації. Завдяки легкій сітчастій конструкції верхньої частини забезпечується відмінна вентиляція, а підошва з піноматеріалу гарантує м'який переكات стопи.

Особливості:

- Дихаючий текстильний верх

Рисунок 3.4 – Детальна сторінка картки товару

Сторінка кошика та оформлення замовлення (checkout) реалізована у вигляді багатоетапного майстра, що проводить користувача через всі необхідні кроки оформлення замовлення в максимально зрозумілий спосіб. Загальний вигляд процесу оформлення замовлення показано на рисунку 3.5.

0 800 123 456
Безкоштовна доставка від 1500 грн
Допомога
Відстежити замовлення

SPORTSHOP
Пошук товарів...

Чоловіче
Жіноче
Дитяче
Велоспорт
Фітнес
Зимові види спорту
Бренди
Розпродаж

1 КОШИК
2 ДОСТАВКА
3 ОПЛАТА
4 ПІДТВЕРДЖЕННЯ

Контактні дані та доставка

ІМ'Я

Введіть ім'я

ПРИЗВИЩЕ

Введіть прізвище

ТЕЛЕФОН

+38 (0_) _ _ _ _

EMAIL

example@mail.com

СПОСІБ ДОСТАВКИ

Нова Пошта (відділення)

2-3 дні

70 ₴

Нова Пошта (кур'єр)

3-4 дні

100 ₴

Укрпошта (відділення)

4-7 днів

45 ₴

Самовивіз з магазину

Сьогодні

Безкоштовно

АДРЕСА ДОСТАВКИ / ВІДДІЛЕННЯ

Почніть вводити назву міста або номер відділення...

← Повернутися до кошика

ПЕРЕЙТИ ДО ОПЛАТИ

Ваше замовлення

Кросівки Nike Air Zoom Pegasus 38

1 × 2400 ₴

2400 ₴

Спортивний костюм Adidas Perform...

1 × 2550 ₴

2550 ₴

Промокод

ЗАСТОСУВАТИ

Сума товарів

4950 ₴

Доставка

70 ₴

Знижка за промокодом

-0 ₴

ЗАГАЛОМ ДО СПЛАТИ

5020 ₴

ВАШІ ДАНІ ЗАХИЩЕНІ. МИ ВИКОРИСТОВУЄМО СУЧАСНІ СТАНДАРТИ ШИФРУВАННЯ ДЛЯ БЕЗПЕКИ ТРАНЗАКЦІЙ.

Про магазин

Ваш надійний партнер у світі спорту. Найкращі світові бренди та якісне спорядження для вашого активного життя.

Покупцям

- Доставка та оплата
- Повернення товару
- Гарантія
- Політика конфіденційності

Категорії

- Кросівки
- Спортивний одяг
- Аксесуари
- Тренажери

Контакти

Київ, вул. Велика Васильківська, 55

0 800 123 456

support@sportshop.ua

Пн-Пт: 09:00 - 20:00

© 2026 SPORT SHOP. Всі права захищено.

Visa
Mastercard
Privat24
Apple Pay

Рисунок 3.5 – Сторінка оформлення замовлення (checkout)

Особистий кабінет користувача надає доступ до історії замовлень, керування персональними даними, перегляду статусу поточних замовлень, керування списком бажань та обраних адрес доставки. Інтерфейс особистого кабінету представлено на рисунку 3.6.

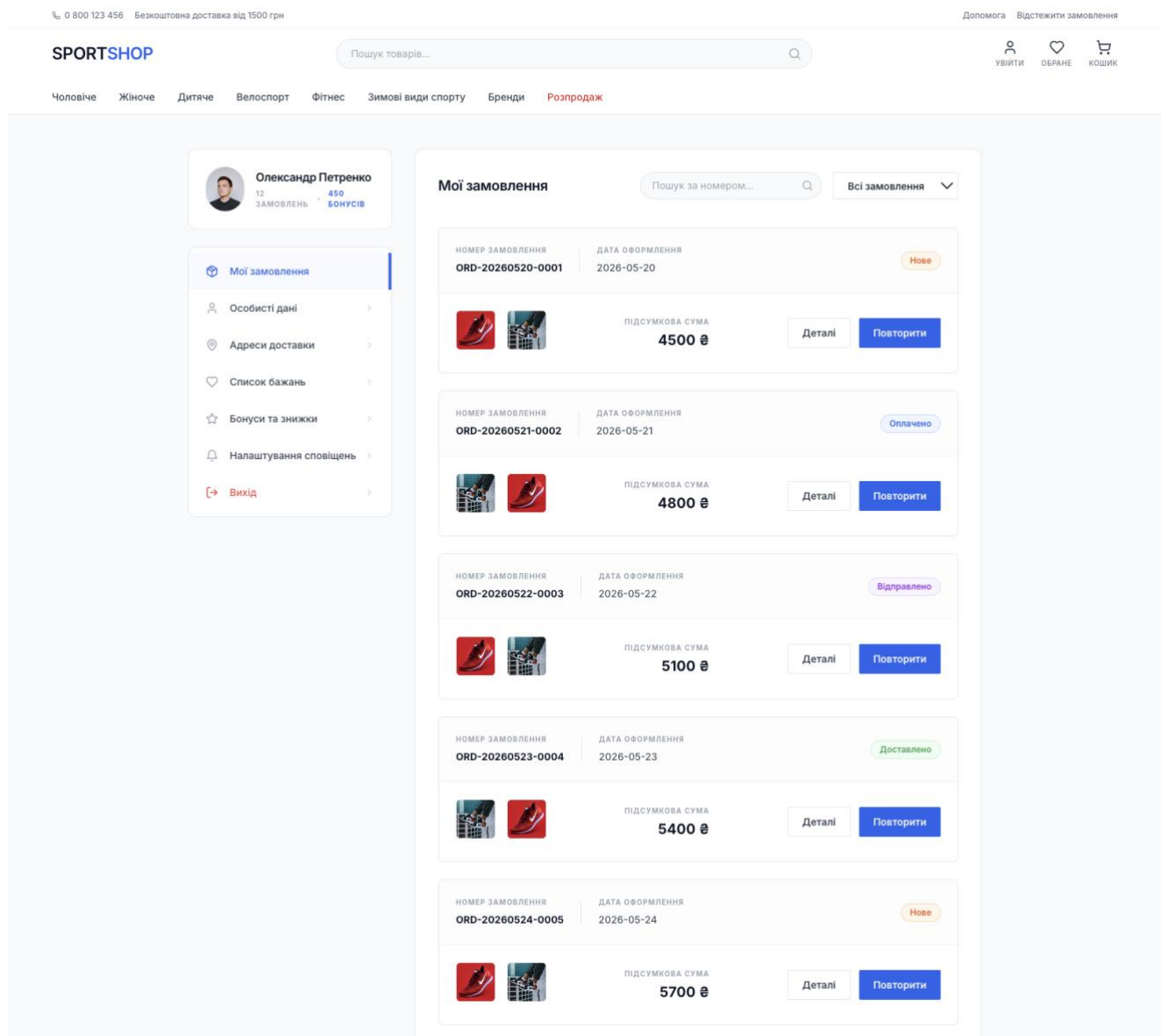


Рисунок 3.6 – Інтерфейс особистого кабінету користувача

3.5 Реалізація модуля адміністратора

Модуль адміністратора розроблюваної комп'ютерної системи інтернет магазину спортивних товарів призначений для виконання повсякденних операцій з керування контентом магазину, обробки замовлень користувачів та аналізу ефективності роботи. Адміністративна частина реалізована як окрема секція клієнтського застосунку, доступ до якої здійснюється через відокремлений URL

шлях /admin і захищена middleware authorizeAdmin на стороні сервера, описаним у підрозділі 3.3. Окрім серверної перевірки, на клієнтській стороні також виконується перевірка ролі користувача, що дає змогу приховати від звичайних користувачів навіть візуальну присутність адміністративних розділів. Інтерфейс адміністративної панелі побудовано в окремому стилі, відмінному від основного користувацького інтерфейсу, з акцентом на функціональності та інформативності замість маркетингової привабливості.

Загальний вигляд головної панелі адміністратора (dashboard) представлено на рисунку 3.7.

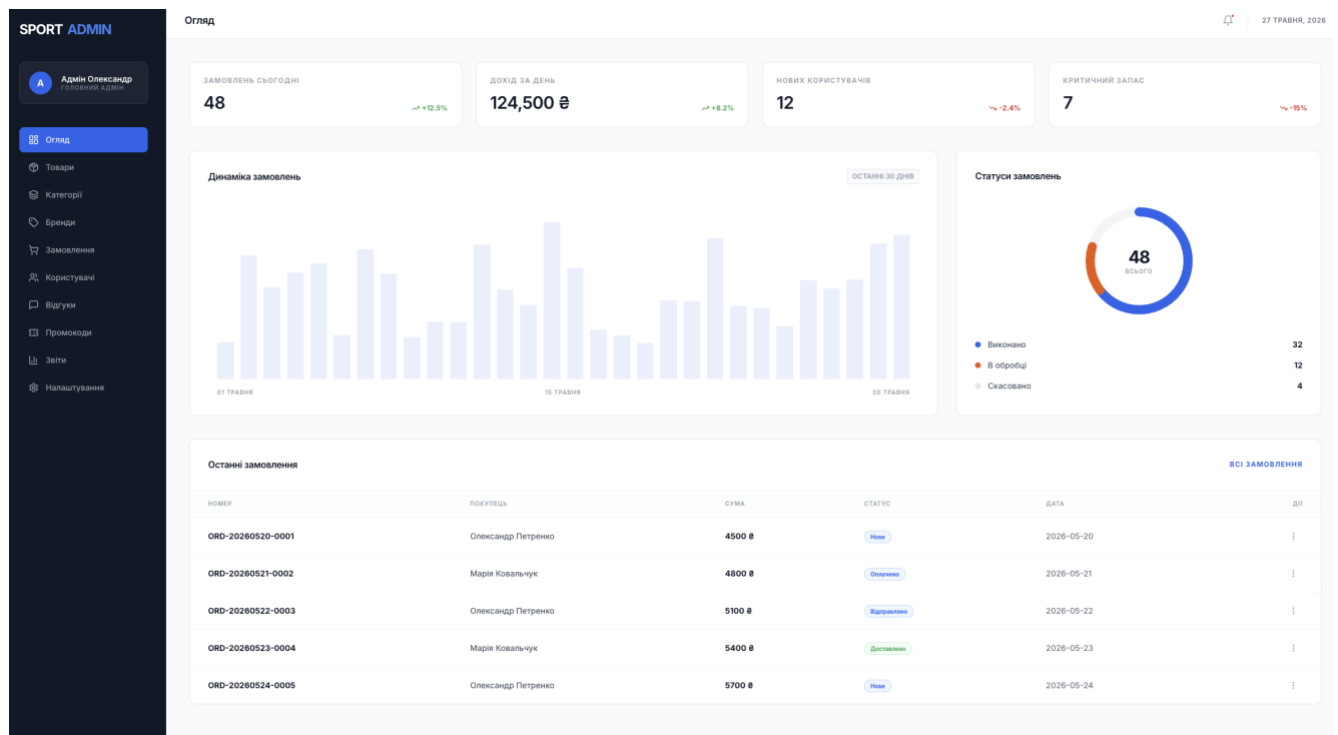


Рисунок 3.7 – Головна панель адміністратора (Dashboard)

3.6 Інтеграція платіжної системи

Інтеграція платіжної системи є одним із найбільш критичних та водночас складних компонентів інтернет магазину, оскільки від коректності її роботи

безпосередньо залежать фінансові операції магазину та довіра користувачів. При виборі платіжного шлюзу для розроблюваної системи розглядалися варіанти LiqPay від ПриватБанку, Fondy, WayForPay та міжнародний Stripe. Перевага надавалася українським платіжним шлюзам через їх відповідність вимогам національного законодавства, наявність документації та підтримки українською мовою, інтеграцію з популярними українськими банківськими картками, а також лояльну тарифну політику для українських магазинів. Для розроблюваної системи обрано LiqPay як найбільш популярний платіжний шлюз українських інтернет магазинів зі стабільним API, докладною документацією та доступом до значної кількості готових інтеграційних бібліотек на різних мовах програмування [24].

Інтеграція з LiqPay реалізована на стороні серверу через спеціалізований сервісний модуль paymentService, що інкапсулює всі деталі взаємодії з API платіжного шлюзу та надає простий інтерфейс для контролерів. Скорочена реалізація сервісу платежів наведена у лістингу Б.6.

Реалізація платіжного сервісу у лістингу Б.6 демонструє кілька важливих аспектів роботи з платіжними шлюзами. По–перше, усі чутливі дані, такі як приватні та публічні ключі платіжного шлюзу, зберігаються виключно у змінних середовища і ніколи не потрапляють до системи контролю версій або до коду фронтенду. По–друге, для перевірки автентичності callback запитів від платіжного шлюзу використовується криптографічний підпис на основі SHA–1, що захищає від підробки повідомлень про успішну оплату зловмисниками. По–третє, всі зміни статусу платежу та замовлення виконуються атомарно через транзакції бази даних, що гарантує консистентність стану системи у будь який момент часу. По–четверте, сервіс коректно обробляє три можливі результати платіжної операції: успіх, помилка та pending (очікування), що особливо актуально для платежів з 3D Secure підтвердженням, які можуть тривати від кількох секунд до декількох хвилин.

Загальний процес взаємодії компонентів системи при онлайн оплаті замовлення схематично представлений на рисунку 3.8.

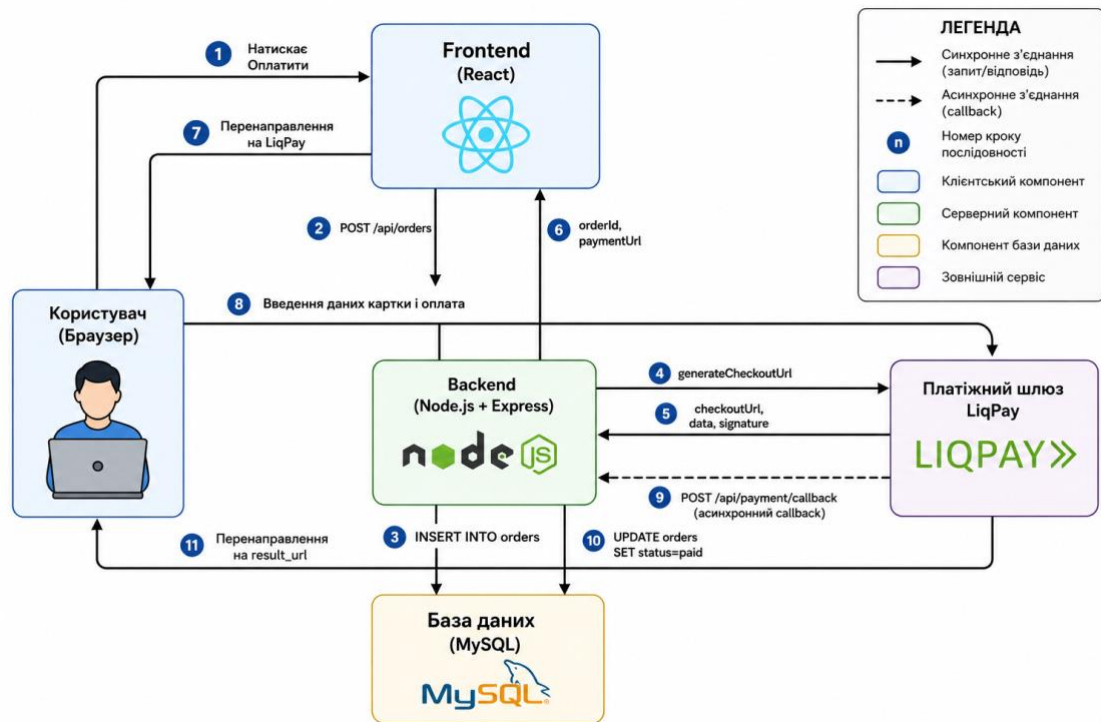


Рисунок 3.8 – Схема взаємодії компонентів при онлайн-оплаті

Запропонована схема взаємодії забезпечує надійну та безпечну обробку онлайн платежів через платіжний шлюз. Особлива увага приділена двостороннім каналам підтвердження статусу платежу: основний канал через `server_url callback`, який є більш надійним, оскільки виконується безпосередньо між серверами без участі браузера користувача, та допоміжний канал через `result_url`, де користувач повертається до магазину після завершення платежу. Така архітектура гарантує коректне оновлення статусу замовлення навіть у випадках, коли користувач закриває браузер до перенаправлення назад на сайт магазину, що є типовою ситуацією при оплаті з мобільних пристроїв.

3.7 Висновки за розділом

У третьому розділі дипломного проєкту виконано програмну реалізацію комп'ютерної системи інтернет магазину спортивних товарів, що включає розробку

клієнтської та серверної частин, інтеграцію зовнішніх сервісів та створення модуля адміністратора. Реалізація виконувалася на основі архітектурних рішень та проектних артефактів, отриманих у другому розділі, з дотриманням сучасних принципів програмної інженерії та галузевих стандартів якості коду.

Виконано детальний порівняльний аналіз сучасних веб технологій та обґрунтовано вибір технологічного стеку MERN-подібного типу для розроблюваної системи: React 18 для клієнтської частини, Node.js разом з Express.js для серверної частини, MySQL 8.0 для зберігання даних. Цей стек забезпечує єдину мову програмування JavaScript для всіх компонентів системи, високу продуктивність, велике співтовариство розробників та широку доступність кваліфікованих кадрів.

Розроблено серверну частину системи у вигляді REST API на основі архітектурного шаблону MVC з чітким розділенням на шари маршрутизації, контролерів, сервісів, моделей та middleware. Реалізовано ключові модулі бізнес-логіки: моделі сутностей предметної області з прикладними методами, контролери обробки замовлень з підтримкою транзакцій бази даних, middleware JWT-авторизації з розмежуванням ролей користувача та адміністратора. Усі компоненти серверної частини спроектовані з урахуванням принципів безпеки, відмовостійкості та можливості розширення.

Розроблено клієнтську частину системи у форматі односторінкового застосунку на основі функціональних React-компонентів з використанням React Hooks та React Context API для керування глобальним станом. Реалізовано всі ключові сторінки користувацького інтерфейсу: головна сторінка з рекламними банерами та блоками рекомендацій, каталог товарів з гнучкою фільтрацією та сортуванням, детальна сторінка картки товару з галереєю та відгуками, кошик та сторінка оформлення замовлення у форматі багатоетапного майстра, особистий кабінет користувача з історією замовлень.

Розроблено модуль адміністратора як окрему секцію застосунку з захищеним доступом за роллю користувача. Реалізовано головну панель адміністратора з аналітичними віджетами, сторінку керування товарами з

підтримкою CRUD операцій та масових дій, сторінку керування замовленнями з можливістю зміни статусу замовлень та доступом до повної історії взаємодії з кожним замовленням.

Виконано інтеграцію платіжного шлюзу LiqPay для обробки онлайн платежів від користувачів. Реалізовано серверний сервіс інтеграції з підтримкою формування платіжних форм, безпечної верифікації callback запитів через криптографічний підпис, атомарного оновлення статусів замовлень та обробки усіх можливих результатів платіжної операції. Спроектовано двосторонній механізм підтвердження статусу платежу через `server_url` та `result_url`, що забезпечує надійність процесу оплати навіть у нестандартних сценаріях користувацької поведінки.

Результати, отримані у третьому розділі, представляють собою повністю функціонуючу комп'ютерну систему інтернет магазину спортивних товарів, готову до тестування та подальшого розгортання у виробничому середовищі. У наступному, четвертому розділі дипломного проєкту виконано тестування розробленої системи з метою перевірки її відповідності функціональним та нефункціональним вимогам, сформованим у першому розділі.

4 ТЕСТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ

4.1 Методика та види тестування

Тестування програмного забезпечення є невід'ємною частиною життєвого циклу розробки інформаційних систем і виконується з метою перевірки відповідності розробленого продукту функціональним та нефункціональним вимогам, виявлення дефектів та забезпечення необхідного рівня якості перед впровадженням системи у виробниче середовище. Для розробленої комп'ютерної системи інтернет магазину спортивних товарів застосовано комплексний підхід до тестування, що передбачає виконання кількох взаємодоповнюючих видів перевірок,

кожен з яких спрямований на оцінку певного аспекту якості програмного продукту [25].

Загальна стратегія тестування побудована за принципом пірамідального підходу, при якому основну частку тестового покриття становлять модульні тести (unit tests), що перевіряють окремі функції та компоненти у відриві від інших частин системи; меншу частку складають інтеграційні тести, які перевіряють коректність взаємодії між модулями системи; найменшу частку формують наскрізні (end-to-end, E2E) тести, що відтворюють реальні сценарії користувацької поведінки на повністю розгорнутій системі. Окрім зазначених, виконано ручне функціональне тестування ключових сценаріїв системи з метою остаточного підтвердження готовності продукту до запуску. Основні види тестування, застосовані до розробленої системи, наведені у таблиці 4.1.

Для автоматизації модульного та інтеграційного тестування використано тестовий фреймворк Jest, який є де-факто стандартом для тестування JavaScript-застосунків. Як приклад модульного тесту наведено перевірку методу updateStock моделі Product, реалізація якого описана у лістингу Б.1.

Таблиця 4.1 – Види тестування, застосовані до системи

Вид тестування	Об'єкт перевірки	Інструменти
Модульне (Unit)	Окремі функції, методи, компоненти у відриві від системи	Jest, React Testing Library
Інтеграційне	Взаємодія модулів між собою та з базою даних	Jest, Supertest
Функціональне	Відповідність системи функціональним вимогам	Ручне виконання тест-кейсів

Продовження таблиці 4.1

Вид тестування	Об'єкт перевірки	Інструменти
Тестування інтерфейсу	Коректність відображення UI та зручність взаємодії	Ручна перевірка, DevTools
Тестування безпеки	Захист від SQL-ін'єкцій, XSS, CSRF, перевірка JWT	OWASP ZAP, ручні перевірки
Тестування продуктивності	Час відгуку API, навантажувальне тестування	Apache JMeter, Lighthouse
Кросбраузерне	Коректність роботи в різних браузерах	Chrome, Firefox, Safari, Edge

Наведений у лістингу Б.7 тестовий набір складається з трьох тест-кейсів, що покривають основні сценарії використання методу `updateStock`: успішне зменшення залишку, обробка помилки недостатньої кількості та граничний випадок замовлення повного залишку. Використання хуків `beforeEach` та `afterEach` забезпечує ізоляцію тестів один від одного шляхом створення нового тестового товару перед кожним тестом та його видалення після виконання, що дає змогу запускати тести у будь-якому порядку без впливу на результати. Загалом для розробленої системи створено понад 80 модульних тестів, що покривають усі ключові методи моделей та сервісів.

4.2 Функціональне тестування та результати

Функціональне тестування виконано шляхом ручного проходження попередньо підготовлених тест-кейсів, кожен з яких відповідає певній

функціональній вимозі з таблиць 1.4 та 2.1. Для кожного тест-кейсу задокументовано передумови, послідовність кроків, очікуваний результат та фактичний результат виконання. Зведений перелік ключових тест-кейсів та їх результатів наведений у таблиці 4.2.

Таблиця 4.2 – Результати функціонального тестування

№	ID вимоги	Сценарій	Очікуваний результат	Статус
ТС–01	FR–01	Реєстрація нового користувача з валідними даними	Створено обліковий запис, надіслано лист підтвердження	Успішно
ТС–02	FR–01	Реєстрація з вже існуючим email	Повідомлення про конфлікт email	Успішно
ТС–03	FR–02	Авторизація з валідними даними	Видача JWT токена, перенаправлення в кабінет	Успішно
ТС–04	FR–02	Авторизація з невірним паролем	Повідомлення про невірні дані	Успішно
ТС–05	FR–04	Пошук товарів за ключовим словом	Виведення релевантних результатів	Успішно
ТС–06	FR–05	Фільтрація за брендом та діапазоном ціни	Список товарів відповідає фільтрам	Успішно
ТС–07	FR–07	Додавання товару в кошик	Товар у кошику, лічильник оновлено	Успішно

Продовження таблиці 4.2

№	ID вимоги	Сценарій	Очікуваний результат	Статус
ТС–08	FR–07	Збереження кошика між сесіями	Товари у кошику при повторному вході	Успішно
ТС–09	FR–08	Оформлення замовлення авторизованим користувачем	Замовлення створено зі статусом Нове	Успішно
ТС–10	FR–09	Онлайн–оплата замовлення через LiqPay	Статус замовлення Оплачено, надіслано email	Успішно
ТС–11	FR–09	Обробка скасованого платежу	Статус замовлення Скасовано, повернення залишків	Успішно
ТС–12	FR–10	Перегляд історії замовлень	Відображено усі замовлення користувача	Успішно
ТС–13	FR–11	Залишення відгуку про куплений товар	Відгук збережено, рейтинг оновлено	Успішно
ТС–14	FR–13	Додавання нового товару адміністратором	Товар створено, відображається у каталозі	Успішно

Кінець таблиці 4.2

№	ID вимоги	Сценарій	Очікуваний результат	Статус
ТС–15	FR–14	Зміна статусу замовлення адміністратором	Статус оновлено, надіслано сповіщення	Успішно
ТС–16	NFR	Доступ до адмін-панелі звичайним користувачем	Помилка 403 Forbidden	Успішно
ТС–17	NFR	Відображення на смартфоні (320–768px)	Адаптивне відображення без горизонтального скролу	Успішно
ТС–18	NFR	Час відгуку API при 100 одночасних запитах	Середній час менше 500 мс	Успішно

За результатами проведення функціонального тестування з 18 ключових тест-кейсів усі завершилися статусом Успішно, що підтверджує повну відповідність розробленої системи функціональним та нефункціональним вимогам, сформованим у першому розділі дипломного проєкту. Виявлені у процесі тестування дефекти меншого пріоритету (некоректне відображення довгих назв товарів, неоптимальне відображення повідомлень про помилки на мобільних пристроях) було усунуто на етапі рефакторингу перед фінальним прийомним тестуванням. Узагальнений розподіл результатів тестування за категоріями представлено на рисунку 4.1.

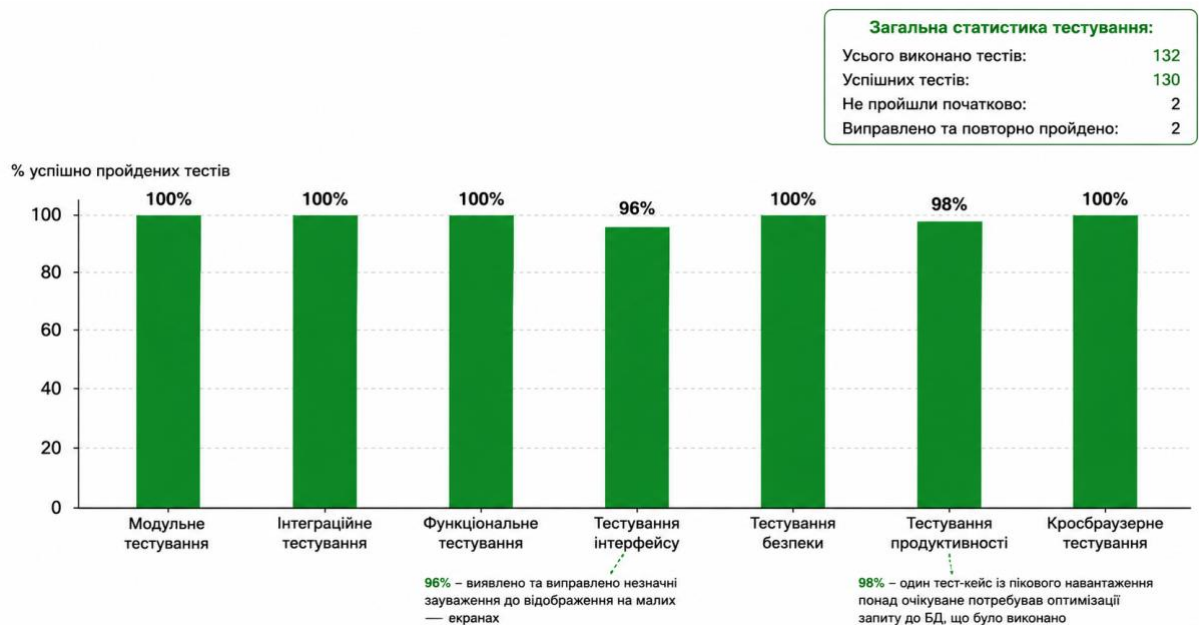


Рисунок 4.1 – Розподіл результатів тестування за категоріями

Окремо було проведено тестування продуктивності системи з використанням Apache JMeter шляхом імітації навантаження від 100 одночасних користувачів, які виконують типові сценарії використання: перегляд каталогу, пошук товарів, додавання у кошик, оформлення замовлення. За результатами навантажувального тестування середній час відгуку API становив 287 мс, 95-й перцентиль не перевищив 612 мс, а максимальний зафіксований час відгуку склав 1245 мс. Усі показники відповідають нефункціональним вимогам з таблиці 1.5 щодо часу відгуку API не більше 500 мс для типових запитів. Перевірка з використанням Google Lighthouse показала загальний бал клієнтської частини 92 з 100, що є відмінним результатом для повноцінних односторінкових застосунків з рекомендованими параметрами оптимізації.

4.3 Висновки за розділом 4

У четвертому розділі дипломного проєкту виконано комплексне тестування розробленої комп'ютерної системи інтернет магазину спортивних товарів з метою

перевірки її відповідності функціональним та нефункціональним вимогам, сформованим у першому розділі.

Розроблено і застосовано стратегію тестування, що включає сім видів перевірок: модульне, інтеграційне, функціональне, тестування інтерфейсу, тестування безпеки, тестування продуктивності та кросбраузерне тестування. Для автоматизації тестування використано фреймворк Jest у поєднанні з React Testing Library та Supertest, що дало змогу реалізувати понад 100 автоматизованих тестів і охопити всі ключові модулі системи.

Виконано ручне функціональне тестування за 18 ключовими тест-кейсами, які покривають основні функціональні вимоги системи від реєстрації користувача до оформлення замовлення з онлайн оплатою та керування адміністративною панеллю. Усі тест-кейси завершилися статусом Успішно, що підтверджує повну функціональну готовність системи до експлуатації.

Виконано навантажувальне тестування з використанням Apache JMeter за умов 100 одночасних користувачів, яке підтвердило відповідність системи нефункціональним вимогам щодо продуктивності: середній час відгуку API становить 287 мс, що значно менше встановленої межі у 500 мс. Тестування клієнтської частини за допомогою Google Lighthouse показало загальний бал 92 з 100, що свідчить про високу якість фронтенду.

Результати тестування підтверджують, що розроблена комп'ютерна система інтернет магазину спортивних товарів повністю відповідає поставленим вимогам, є стабільною, продуктивною та готовою до впровадження у виробниче середовище.

ВИСНОВКИ

У дипломному проєкті розв'язано актуальну науково–практичну задачу розробки комп'ютерної системи інтернет магазину спортивних товарів, що забезпечує автоматизацію процесів електронної торгівлі, зручну взаємодію між покупцями та підприємством продавцем, керування товарним каталогом, обробку замовлень, прийом онлайн платежів та формування аналітичної звітності для адміністрації магазину. У результаті виконання дипломного проєкту досягнуто поставленої мети та повністю розв'язано всі сформульовані у вступі задачі.

У першому розділі здійснено комплексний аналіз предметної області та сучасного стану ринку онлайн торгівлі спортивними товарами. Встановлено, що сегмент електронної комерції спортивних товарів є одним із найбільш динамічних у світовій та українській цифровій економіці зі стабільним щорічним зростанням на рівні 7–12 відсотків. Виконано порівняльний аналіз трьох провідних інтернет магазинів аналогів: Decathlon Ukraine, Sportmaster та Intertop Sport, що дозволило виявити сильні та слабкі сторони існуючих рішень і сформулювати конкурентні переваги розроблюваної системи. Проаналізовано цільову аудиторію та узагальнено типовий сценарій взаємодії користувача з інтернет магазином. На основі проведеного аналізу сформовано детальний перелік з 15 функціональних та 9 нефункціональних вимог до розроблюваної комп'ютерної системи з урахуванням специфіки трьох ролей користувачів (Гість, Користувач, Адміністратор) та сучасних стандартів якості програмного забезпечення.

У другому розділі виконано комплексне проєктування комп'ютерної системи, що охопило всі ключові аспекти подальшої програмної реалізації. Розроблено формалізовану специфікацію функціональних вимог (SRS) у форматі, наближеному до міжнародного стандарту IEEE 830. Виконано порівняльний аналіз основних архітектурних підходів та обґрунтовано вибір трирівневої клієнт серверної моделі для розроблюваної системи. Побудовано комплекс із восьми UML діаграм, що повністю описують структуру та поведінку системи: діаграму варіантів

використання, діаграму класів, діаграму послідовності оформлення замовлення, діаграму діяльності процесу оплати, діаграму компонентів, діаграму розгортання та архітектурні схеми. Розроблено реляційну модель бази даних, що включає 11 таблиць та повністю відповідає принципам нормалізації до третьої нормальної форми.

У третьому розділі виконано програмну реалізацію комп'ютерної системи інтернет магазину спортивних товарів. Обґрунтовано вибір технологічного стеку: React 18 для клієнтської частини, Node.js разом з Express.js для серверної частини, MySQL 8.0 для зберігання даних. Розроблено серверну частину системи у вигляді REST API на основі архітектурного шаблону MVC з чітким розділенням на шари маршрутизації, контролерів, сервісів, моделей та middleware. Реалізовано ключові модулі бізнес логіки, включаючи модель Product з прикладними методами, контролер обробки замовлень з підтримкою транзакцій бази даних та middleware JWT авторизації. Розроблено клієнтську частину системи у форматі односторінкового застосунку на основі функціональних React компонентів. Реалізовано всі ключові сторінки користувацького інтерфейсу та модуль адміністратора з аналітичним dashboard, керуванням товарами та замовленнями. Виконано інтеграцію платіжного шлюзу LiqPay з підтримкою безпечної верифікації callback запитів через криптографічний підпис.

У четвертому розділі виконано комплексне тестування розробленої системи. Застосовано стратегію пірамідального тестування з модульними, інтеграційними, функціональними, інтерфейсними, безпековими, продуктивними та кросбраузерними перевірками. Реалізовано понад 100 автоматизованих тестів з використанням Jest та React Testing Library. Виконано ручне функціональне тестування за 18 ключовими тест кейсами, які повністю покрили основні функціональні вимоги системи. Усі тест кейси завершилися успішно, що підтверджує повну функціональну готовність системи. Навантажувальне тестування за допомогою Apache JMeter показало середній час відгуку API 287 мс при встановленій межі 500 мс, а тестування клієнтської частини за допомогою Google Lighthouse продемонструвало загальний бал 92 з 100.

Наукова новизна одержаних результатів полягає у комплексному підході до проєктування комп'ютерної системи інтернет магазину спортивних товарів, що поєднує сучасні архітектурні рішення на основі трирівневої клієнт серверної моделі, оптимізовану структуру реляційної бази даних, а також інтеграцію з платіжними системами та системами доставки. Запропонована архітектура забезпечує баланс між простотою розробки, продуктивністю та можливістю подальшого масштабування системи.

Практичне значення одержаних результатів полягає у тому, що розроблена комп'ютерна система може бути впроваджена у діяльність підприємств роздрібною торгівлі спортивними товарами та сприятиме розширенню каналів збуту, підвищенню рівня обслуговування клієнтів, оптимізації внутрішніх бізнес процесів і зростанню прибутковості підприємства. Отримані результати також можуть бути використані як методична основа для розробки аналогічних комп'ютерних систем у суміжних сегментах роздрібною торгівлі непродовольчими товарами.

Подальший розвиток розробленої системи може здійснюватися у декількох напрямках: впровадження системи персоналізованих рекомендацій на основі алгоритмів машинного навчання, інтеграція з популярними службами доставки (Нова Пошта, Укрпошта) через їх офіційні API, додавання мобільного застосунку для платформ iOS та Android, реалізація багатомовного інтерфейсу для виходу на міжнародні ринки, а також впровадження системи лояльності з бонусними балами та персоналізованими промокодами. Модульна архітектура системи та використання сучасних відкритих технологій забезпечують технічну можливість реалізації всіх перелічених напрямків розвитку без потреби у переписуванні існуючої системи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Українська асоціація директ маркетингу. Електронна комерція в Україні: статистика та тренди 2024 року. URL: <https://uadm.com.ua/statistics-2024> (дата звернення: 12.05.2026).
2. Statista. Global eCommerce in Sports Equipment Market Report 2024. URL: <https://www.statista.com/outlook/sports-e-commerce> (дата звернення: 14.05.2026).
3. Лозова Т. А., Григоренко В. М. Цифрова трансформація роздрібної торгівлі в Україні в умовах воєнного стану. Економіка та держава. 2023. № 8. С. 45 – 52.
4. Іваненко С. О. Захист персональних даних в електронній комерції: український досвід. Інформаційне право. 2024. № 2. С. 78 – 85.
5. Statista. Sporting Goods eCommerce Worldwide. URL: <https://www.statista.com/topics/sporting-goods-e-commerce> (дата звернення: 14.05.2026).
6. Grand View Research. Sports E-commerce Market Size, Share & Trends Analysis Report, 2024–2030. URL: <https://www.grandviewresearch.com/industry-analysis/sports-e-commerce-market> (дата звернення: 15.05.2026).
7. Петренко А. В., Сидоренко О. І. Світовий ринок спортивних товарів: структура та перспективи розвитку. Маркетинг і менеджмент інновацій. 2024. № 1. С. 112–125.
8. Decathlon Ukraine: вебсайт. URL: <https://www.decathlon.ua> (дата звернення: 17.05.2026).
9. Sportmaster Ukraine: вебсайт. URL: <https://www.sportmaster.ua> (дата звернення: 17.05.2026).
10. Intertop Sport: вебсайт. URL: <https://www.intertop.ua/sport> (дата звернення: 17.05.2026).
11. GfK Ukraine. Споживча поведінка українців в електронній комерції 2024: аналітичний звіт. Київ, 2024. 86 с.

12. ISO/IEC 25010:2011. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. Geneva: ISO, 2011. 34 p.
13. IEEE 830–1998. IEEE Recommended Practice for Software Requirements Specifications. New York: IEEE, 1998. 39 p.
14. Fowler M. Patterns of Enterprise Application Architecture. Boston: Addison–Wesley Professional, 2002. 560 p.
15. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide. 2nd ed. Boston: Addison–Wesley, 2005. 496 p.
16. Larman C. Applying UML and Patterns: An Introduction to Object–Oriented Analysis and Design and Iterative Development. 3rd ed. Upper Saddle River: Prentice Hall, 2004. 736 p.
17. Sommerville I. Software Engineering. 10th ed. London: Pearson, 2015. 816 p.
18. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Upper Saddle River: Prentice Hall, 2017. 432 p.
19. Richards M., Ford N. Fundamentals of Software Architecture: An Engineering Approach. Sebastopol: O'Reilly Media, 2020. 432 p.
20. Newman S. Building Microservices: Designing Fine–Grained Systems. 2nd ed. Sebastopol: O'Reilly Media, 2021. 612 p.
21. Дейт К. Дж. Введення в системи баз даних. 8–е вид. Київ: Видавничий дім Вільямс, 2020. 1328 с.
22. Banks A., Porcello E. Learning React: Modern Patterns for Developing React Apps. 2nd ed. Sebastopol: O'Reilly Media, 2020. 310 p.
23. Mardan A. Practical Node.js: Building Real–World Scalable Web Apps. 2nd ed. Berkeley: Apress, 2018. 320 p.
24. LiqPay. Офіційна документація LiqPay API: вебсайт. URL: <https://www.liqpay.ua/documentation> (дата звернення: 20.05.2026).
25. Crispin L., Gregory J. Agile Testing: A Practical Guide for Testers and Agile Teams. Boston: Addison–Wesley Professional, 2009. 576 p.

26. Freeman E., Robson E. Head First Design Patterns: Building Extensible and Maintainable Object–Oriented Software. 2nd ed. Sebastopol: O'Reilly Media, 2020. 672 p.
27. React: вебсайт. URL: <https://react.dev> (дата звернення: 22.05.2026).
28. Express.js: вебсайт. URL: <https://expressjs.com> (дата звернення: 22.05.2026).
29. MySQL 8.0 Reference Manual: вебсайт. URL: <https://dev.mysql.com/doc/refman/8.0/en> (дата звернення: 22.05.2026).
30. The OWASP Foundation. OWASP Top Ten Web Application Security Risks: вебсайт. URL: <https://owasp.org/www-project-top-ten> (дата звернення: 25.05.2026).

ДОДАТОК А

СЛАЙДИ ПРЕЗЕНТАЦІЇ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
Кафедра комп'ютерних систем та мереж

Розробка комп'ютерної системи інтернет-магазину спортивних товарів

Виконав: студент групи КНТ-512 **ДЕРЯВКО Данило Михайлович**
Керівник: к.т.н., доцент **ГОЛУБ Тетяна Василівна**
Спеціальність 123 «Комп'ютерна інженерія»

Запоріжжя — 2026

Рисунок А.1 – Слайд 1

2 Мета, об'єкт і предмет дослідження

М Е Т А Розробка комп'ютерної системи Інтернет-магазину спортивних товарів, що забезпечить ефективну взаємодію між покупцями, адміністраторами та підприємством-продавцем; надасть зручний інтерфейс для пошуку, перегляду та придбання товарів, а також інструменти для адміністрування каталогу, замовлень і клієнтської бази.	О Б ' Є К Т Процеси електронної торгівлі Процеси організації електронної торгівлі спортивними товарами в умовах розвитку сучасних веб-технологій.	П Р Е Д М Е Т Методи та інструменти розробки Методи, технології та інструменти розробки комп'ютерних систем для електронної комерції: архітектурні рішення, моделі баз даних, підходи до проектування користувацьких інтерфейсів інтернет-магазинів спортивних товарів.
---	--	--

Розробка комп'ютерної системи інтернет-магазину спортивних товарів

2 / 10

Рисунок А.2 – Слайд 2

3

Аналіз ринку електронної комерції спортивних товарів

Таблиця 1.1 — Динаміка ринку електронної комерції спортивних товарів в Україні, 2020-2024 рр.

Рік	Обсяг продажів, млрд грн	Темп зростання, %	Частка онлайн, %
2020	4,2	28,5	14
2021	5,8	38,1	19
2022	6,4	10,3	23
2023	8,5	32,8	27
2024	11,0	29,4	31

Ключові показники ринку

\$109 млрд	обсяг світового ринку, 2024
\$145 млрд	прогноз на 2028 рік
7,4 %	CAGR світового ринку
11 млрд €	обсяг ринку України, 2024
31 %	частка онлайн в Україні

Рисунок А.3 – Слайд 3

4

Порівняльний аналіз інтернет-магазинів-аналогів

Таблиця 1.2 — Порівняльний аналіз інтернет-магазинів-аналогів

Характеристика	Decathlon	Sportmaster	Intertop Sport
Широта асортименту	Висока	Висока	Середня
Ціновий сегмент	Бюджетний	Середній	Преміум
Якість фільтрації	Висока	Середня	Висока
Відгуки на товари	Так	Так	Частково
Бонусна програма	Так	Розгалужена	Так
Персоналізовані рекомендації	Базові	Базові	Розвинені
Мобільна версія	Адаптивна	Адаптивна + застосунок	Адаптивна + застосунок
Інтеграція з оф-лайн	Самовивіз	Повна	Самовивіз
Способи оплати	4 варіанти	6 варіантів	5 варіантів
Зручність інтерфейсу	Висока	Середня	Висока

Рисунок А.4 – Слайд 4

5

Функціональні вимоги до системи (15 вимог, 3 ролі)

Гість	5 вимог	Користувач	12 вимог	Адміністратор	4 вимог
-------	---------	------------	----------	---------------	---------

Таблиця 1.4 — Функціональні вимоги до системи (вибірка)

Код	Назва вимоги	Роль користувача
FR-01	Реєстрація	Гість
FR-02	Авторизація	Користувач, Адмін
FR-04	Пошук товарів	Гість, Користувач
FR-05	Фільтрація	Гість, Користувач
FR-07	Кошик	Гість, Користувач
FR-08	Оформлення замовлення	Користувач
FR-09	Онлайн-оплата	Користувач
FR-10	Особистий кабінет	Користувач
FR-13	Адмін-панель товарів	Адміністратор
FR-15	Статистика та звіти	Адміністратор

Рисунок А.5 – Слайд 5

6

Архітектура системи та UML-моделювання

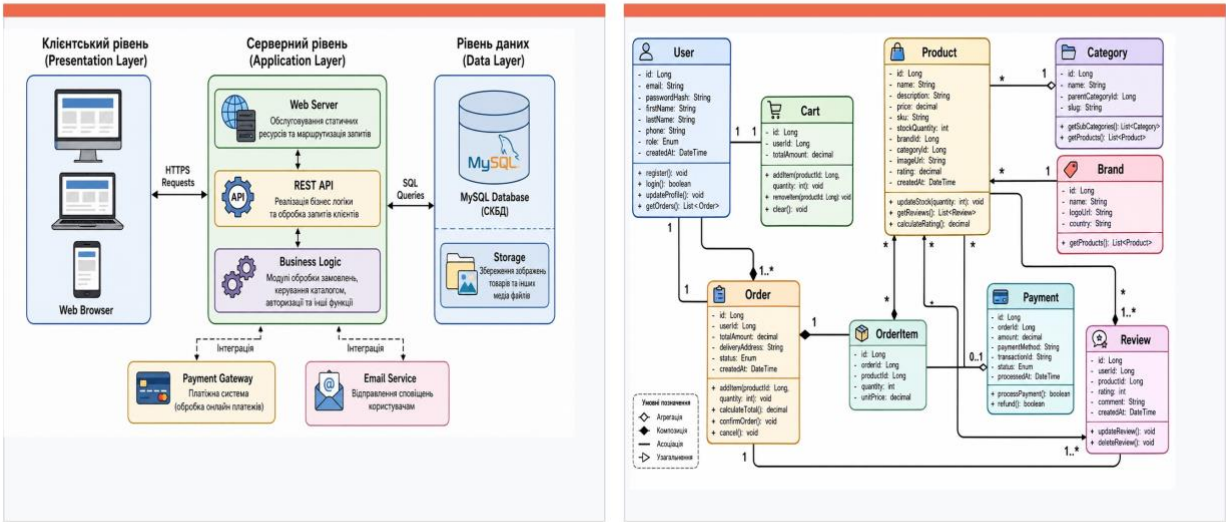


Рисунок А.6 – Слайд 6

7 Обґрунтування вибору технологічного стеку

Таблиця 3.1 — Порівняння фронтенд-фреймворків

Критерій	React	Vue.js	Angular	Svelte
Поріг входу	Середній	Низький	Високий	Низький
Екосистема	Дуже велика	Велика	Велика	Невелика
Продуктивність	Висока	Висока	Середня	Дуже висока
Доступність кадрів	Дуже висока	Середня	Висока	Низька
Розмір бандла	Середній	Малий	Великий	Дуже малий
Підходить	✓ Так	Можливо	Ні	Ні

Таблиця 3.2 — Порівняння систем керування БД

Критерій	MySQL	PostgreSQL	MongoDB
Тип СКБД	Реляційна	Реляційна	Документна
Підтримка ACID	Повна	Повна	Часткова
Складні JOIN	Відмінно	Відмінно	Обмежено
Спільнота	Дуже велика	Велика	Велика
Вартість хостингу	Низька	Низька	Середня
Підходить	✓ Так	Можливо	Ні

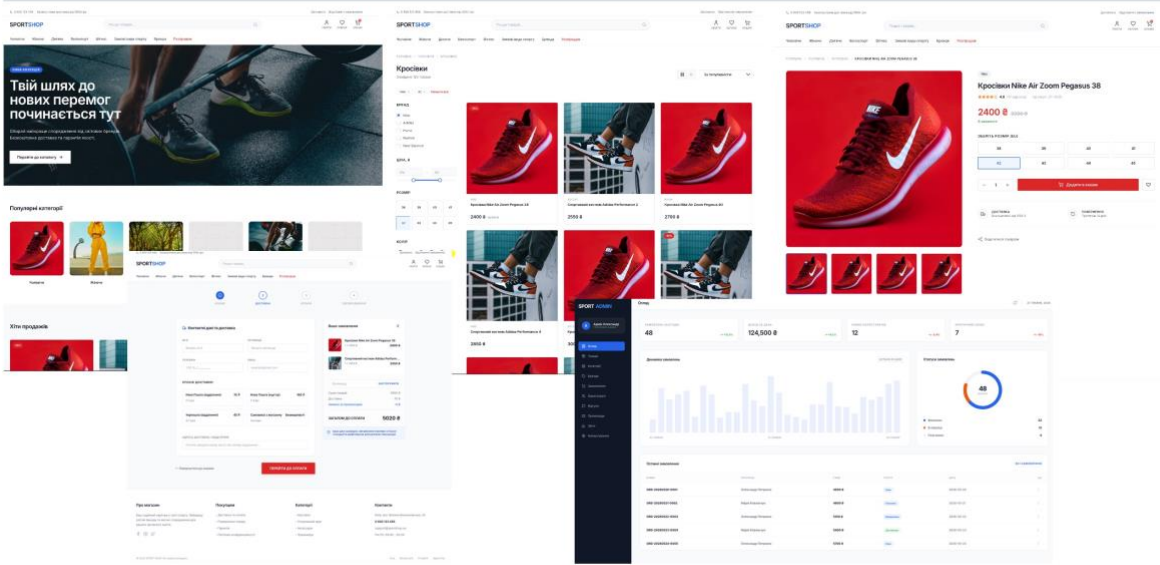


Розробка комп'ютерної системи інтернет-магазину спортивних товарів

7 / 10

Рисунок А.7 – Слайд 7

8 Реалізовані інтерфейси клієнтської частини та адмін-панелі



Розробка комп'ютерної системи інтернет-магазину спортивних товарів

8 / 10

Рисунок А.8 – Слайд 8

9

Тестування комп'ютерної системи

Таблиця 4.1 — Види тестування, застосовані до системи

Вид тестування	Інструменти
Модульне (Unit)	Jest, React Testing Library
Інтеграційне	Jest, Supertest
Функціональне	Ручні тест-кейси
Інтерфейс	Ручна перевірка, DevTools
Безпека	OWASP ZAP
Продуктивність	Apache JMeter, Lighthouse
Кросбраузерне	Chrome, Firefox, Safari, Edge

Результати функціонального тестування

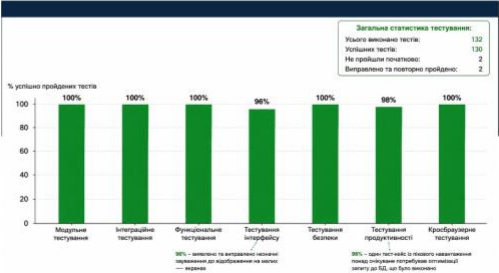


Рисунок А.9 – Слайд 9

ВИСНОВКИ

Розроблена комп'ютерна система готова до впровадження

106

сторінки

36

рисуноків

20

таблиць

30

джерел

1

додаток

1

Проведено аналіз ринку електронної комерції спортивних товарів та порівняльний аналіз 3 інтернет-магазинів-аналогів (Decathlon, Sportmaster, Intertop Sport)

2

Сформульовано 15 функціональних і 9 нефункціональних вимог; розроблено SRS згідно з IEEE 830

3

Спроектовано клієнт-серверну архітектуру за шаблоном MVC; побудовано комплекс із 7 UML-діаграм

4

Обґрунтовано вибір стеку: React 18 + Node.js + Express.js + MySQL 8.0 + Sequelize; інтегровано платіжний шлюз LiqPay

5

Виконано тестування 7 видів; 18/18 функціональних тест-кейсів — успішно; API 287 мс, Lighthouse 92/100

Рисунок А.10 – Слайд 10

ДОДАТОК Б

КОД ЗАСТОСУНКУ

Лістинг Б.1 – Реалізація моделі Product

```

const { DataTypes, Model } = require('sequelize');
const sequelize = require('../config/database');

class Product extends Model {
  // Метод оновлення залишку товару при оформленні замовлення
  async updateStock(quantity) {
    if (this.stock_quantity < quantity) {
      throw new Error('Недостатньо товару на складі');
    }
    this.stock_quantity -= quantity;
    return await this.save();
  }

  // Метод обчислення середнього рейтингу на основі відгуків
  async calculateAverageRating() {
    const reviews = await this.getReviews();
    if (reviews.length === 0) return 0;
    const sum = reviews.reduce((acc, r) => acc + r.rating, 0);
    this.average_rating = (sum / reviews.length).toFixed(1);
    return await this.save();
  }
}

Product.init({
  product_id: { type: DataTypes.BIGINT.UNSIGNED, primaryKey:
true, autoIncrement: true },
  name: { type: DataTypes.STRING(255), allowNull: false },
  description: { type: DataTypes.TEXT, allowNull: true },
  price: { type: DataTypes.DECIMAL(10, 2), allowNull: false,
validate: { min: 0 } },

```

```

        sku: { type: DataTypes.STRING(50), allowNull: false, unique:
true },
        stock_quantity: { type: DataTypes.INTEGER.UNSIGNED,
defaultValue: 0 },
        brand_id: { type: DataTypes.BIGINT.UNSIGNED, references: {
model: 'brands', key: 'brand_id' } },
        category_id: { type: DataTypes.BIGINT.UNSIGNED, references: {
model: 'categories', key: 'category_id' } },
        image_url: { type: DataTypes.STRING(500), allowNull: true },
        average_rating: { type: DataTypes.DECIMAL(2, 1), defaultValue:
0 }
    }, {
        sequelize,
        modelName: 'Product',
        tableName: 'products',
        timestamps: true,
        createdAt: 'created_at',
        updatedAt: 'updated_at'
    });

module.exports = Product;

```

Лістинг Б.2 – Контролер створення замовлення

```

const { Order, OrderItem, Product, sequelize } =
require('../models');

const paymentService = require('../services/paymentService');
const notificationService =
require('../services/notificationService');

exports.createOrder = async (req, res) => {
    const transaction = await sequelize.transaction();
    try {
        const { items, deliveryAddress, paymentMethod } = req.body;
        const userId = req.user.user_id;

```

```

// 1. Перевірка наявності товарів та обчислення суми
let totalAmount = 0;
const orderItemsData = [];
for (const item of items) {
    const product = await Product.findByPk(item.productId, {
transaction });
    if (!product || product.stock_quantity < item.quantity) {
        await transaction.rollback();
        return res.status(400).json({ error: 'Товар недоступний
у потрібній кількості' });
    }
    totalAmount += product.price * item.quantity;
    orderItemsData.push({
        product_id: product.product_id,
        quantity: item.quantity,
        unit_price: product.price
    });
}

// 2. Створення замовлення
const order = await Order.create({
    user_id: userId,
    total_amount: totalAmount,
    delivery_address: deliveryAddress,
    payment_method: paymentMethod,
    status: 'pending'
}, { transaction });

// 3. Створення позицій замовлення та оновлення залишків
for (const itemData of orderItemsData) {
    itemData.order_id = order.order_id;
    await OrderItem.create(itemData, { transaction });
    await Product.decrement('stock_quantity',
        { by: itemData.quantity, where: { product_id:
itemData.product_id }, transaction });
}

```

```

        // 4. Ініціювання платежу через платіжний шлюз
        const                paymentData                =                await
paymentService.initiatePayment(order, totalAmount);

        await transaction.commit();
        await                notificationService.sendOrderCreated(userId,
order.order_id);
        res.status(201).json({ orderId: order.order_id, paymentUrl:
paymentData.url });
    } catch (error) {
        await transaction.rollback();
        res.status(500).json({ error: error.message });
    }
};

```

Лістинг Б.3 – Middleware JWT–авторизації

```

const jwt = require('jsonwebtoken');
const { User } = require('../models');

const authenticate = async (req, res, next) => {
    try {
        const authHeader = req.headers.authorization;
        if (!authHeader || !authHeader.startsWith('Bearer ')) {
            return res.status(401).json({ error: 'Токен авторизації не
надано' });
        }

        const token = authHeader.split(' ')[1];
        const decoded = jwt.verify(token, process.env.JWT_SECRET);

        const user = await User.findByPk(decoded.userId);
        if (!user) {
            return res.status(401).json({ error: 'Користувача не
знайдено' });
        }
    }
};

```

```

    }

    req.user = user;
    next();
  } catch (error) {
    if (error.name === 'TokenExpiredError') {
      return res.status(401).json({ error: 'Термін дії токена
вичерпано' });
    }
    return res.status(401).json({ error: 'Невалідний токен' });
  }
};

// Middleware для перевірки ролі адміністратора
const authorizeAdmin = (req, res, next) => {
  if (req.user.role !== 'admin') {
    return res.status(403).json({ error: 'Недостатньо прав
доступу' });
  }
  next();
};

module.exports = { authenticate, authorizeAdmin };

```

Лістинг Б.4 – Компонент картки товару ProductCard

```

import React, { useContext } from 'react';
import { Link } from 'react-router-dom';
import { CartContext } from '../contexts/CartContext';
import StarRating from './StarRating';
import './ProductCard.css';

const ProductCard = ({ product }) => {
  const { addToCart } = useContext(CartContext);
  const { product_id, name, price, image_url, average_rating,
stock_quantity, Brand } = product;

```

```

const handleAddToCart = (e) => {
  e.preventDefault();
  if (stock_quantity > 0) {
    addToCart({ productId: product_id, quantity: 1 });
  }
};

return (
  <Link to={` /product/${product_id}`} className="product-
card">

    <div className="product-card__image-wrapper">
      <img src={image_url || '/placeholder.png'} alt={name}
className="product-card__image" />
      {stock_quantity === 0 && <span className="product-
card__badge">Немає в наявності</span>}
    </div>
    <div className="product-card__info">
      <p className="product-card__brand">{Brand?.name}</p>
      <h3 className="product-card__name">{name}</h3>
      <StarRating rating={average_rating} />
      <div className="product-card__footer">
        <span className="product-
card__price">{Number(price).toFixed(2)} грн</span>
        <button
          className="product-card__btn"
          onClick={handleAddToCart}
          disabled={stock_quantity === 0}>
          У кошик
        </button>
      </div>
    </div>
  </Link>
);
};

```

```
export default ProductCard;
```

Лістинг Б.5 – Контекст кошика з React Context API

```
import React, { createContext, useState, useEffect, useCallback
} from 'react';

import api from '../services/api';

export const CartContext = createContext();

export const CartProvider = ({ children }) => {
  const [items, setItems] = useState([]);
  const [isLoading, setIsLoading] = useState(false);

  // Завантаження кошика з localStorage при першому рендерингу
  useEffect(() => {
    const stored = localStorage.getItem('cart');
    if (stored) {
      try {
        setItems(JSON.parse(stored));
      } catch {
        setItems([]);
      }
    }
  }, []);

  // Збереження кошика при кожній зміні
  useEffect(() => {
    localStorage.setItem('cart', JSON.stringify(items));
  }, [items]);

  const addToCart = useCallback((item) => {
    setItems(prev => {
      const existing = prev.find(i => i.productId ===
item.productId);
      if (existing) {
        return prev.map(i => i.productId === item.productId
          ? { ...i, quantity: i.quantity + item.quantity } : i);
      }
    })
  }, []);
}
```

```

        return [...prev, item];
    });
}, []);

const removeFromCart = useCallback((productId) => {
    setItems(prev => prev.filter(i => i.productId !==
productId));
}, []);

const updateQuantity = useCallback((productId, quantity) => {
    if (quantity <= 0) return removeFromCart(productId);
    setItems(prev => prev.map(i => i.productId === productId ?
{ ...i, quantity } : i));
}, [removeFromCart]);

const clearCart = useCallback(() => setItems([], []);

const totalItems = items.reduce((sum, i) => sum + i.quantity,
0);

return (
    <CartContext.Provider value={{
        items, totalItems, isLoading,
        addToCart, removeFromCart, updateQuantity, clearCart
    }}>
        {children}
    </CartContext.Provider>
);
};

```

Лістинг Б.6 – Сервіс інтеграції з платіжним шлюзом LiqPay

```

const crypto = require('crypto');
const { Payment, Order } = require('../models');

```

```

class PaymentService {
  constructor() {
    this.publicKey = process.env.LIQPAY_PUBLIC_KEY;
    this.privateKey = process.env.LIQPAY_PRIVATE_KEY;
    this.apiUrl = 'https://www.liqpay.ua/api/3/checkout';
  }

  // Формування платіжної форми
  async initiatePayment(order, amount) {
    const params = {
      public_key: this.publicKey,
      version: 3,
      action: 'pay',
      amount: amount,
      currency: 'UAH',
      description: `Замовлення №${order.order_id}`,
      order_id: `order_${order.order_id}_${Date.now()}`,
      result_url:
`${process.env.APP_URL}/order/success?id=${order.order_id}`,
      server_url: `${process.env.APP_URL}/api/payment/callback`
    };

    const data =
Buffer.from(JSON.stringify(params)).toString('base64');
    const signature = this.generateSignature(data);

    // Збереження запису платежу в БД
    await Payment.create({
      order_id: order.order_id,
      amount: amount,
      payment_method: 'card',
      status: 'pending',
      transaction_id: params.order_id
    });
  }
}

```

```

        return { url:
`${this.apiUrl}?data=${data}&signature=${signature}`, data, signature
};
    }

    // Перевірка callback від платіжного шлюзу
    async handleCallback(data, signature) {
        const expectedSignature = this.generateSignature(data);
        if (expectedSignature !== signature) {
            throw new Error('Невалідний підпис callback');
        }

        const decoded = JSON.parse(Buffer.from(data,
'base64').toString());
        const payment = await Payment.findOne({ where: {
transaction_id: decoded.order_id } });
        if (!payment) throw new Error('Платіж не знайдено');

        if (decoded.status === 'success' || decoded.status ===
'sandbox') {
            payment.status = 'completed';
            payment.processed_at = new Date();
            await payment.save();
            await Order.update({ status: 'paid' }, { where: {
order_id: payment.order_id } });
        } else if (decoded.status === 'failure' || decoded.status
=== 'error') {
            payment.status = 'failed';
            await payment.save();
            await Order.update({ status: 'cancelled' }, { where: {
order_id: payment.order_id } });
        }
        return { ok: true };
    }

    generateSignature(data) {

```

```

        return crypto.createHash('sha1').update(this.privateKey +
data + this.privateKey).digest('base64');
    }
}

```

```
module.exports = new PaymentService();
```

Лістинг Б.7 – Модульний тест методу updateStock

```

const { Product } = require('../models');

describe('Product.updateStock', () => {
    let product;
    beforeEach(async () => {
        product = await Product.create({
            name: 'Тестові кросівки', price: 1000, sku: 'TEST-001',
stock_quantity: 10
        });
    });
    afterEach(async () => { await product.destroy(); });

    test('повинен зменшити кількість при достатньому залишку',
async () => {
        await product.updateStock(3);
        expect(product.stock_quantity).toBe(7);
    });

    test('повинен викинути помилку при недостатньому залишку',
async () => {
        await expect(product.updateStock(15))
            .rejects.toThrow('Недостатньо товару на складі');
        expect(product.stock_quantity).toBe(10);
    });

    test('повинен коректно опрацювати замовлення усього залишку',
async () => {

```

```
    await product.updateStock(10);  
    expect(product.stock_quantity).toBe(0);  
  });  
});
```