

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Проектування інформаційної системи для інтернет-магазину
мобільних телефонів Mobile-Fan»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_2021
Бортовов Дмитро Сергійович
Керівник: Уткін Юрій Вікторович
Рецензент: Муравльов Володимир
Вячеславович

Полтава – 2025 року

ВСТУП

Актуальність. У сучасному світі електронна комерція стала невід’ємною частиною економічної діяльності, забезпечуючи зручність купівлі-продажу товарів через Інтернет. За даними аналітичних звітів [1, 2], обсяг світового ринку e-commerce у 2024 р. перевищив 6 трлн доларів США, а темпи зростання онлайн-продажів в Україні становлять близько 20–25% щорічно. Особливо популярними стають інтернет-магазини мобільних телефонів, які дозволяють користувачам швидко знаходити товари, порівнювати технічні характеристики та оформлювати замовлення з будь-якої точки світу. В Україні, за оцінками Асоціації ритейлерів, у 2024 р. онлайн-продажі електроніки, включаючи мобільні телефони, склали понад 30% від загального обсягу ринку, що свідчить про зростання попиту на цифрові платформи для покупки техніки [3].

Однак, незважаючи на зростання популярності онлайн-магазинів, багато існуючих платформ мають обмеження: недостатня адаптація до потреб українського ринку, обмежені можливості фільтрації за технічними характеристиками, низька швидкість обробки запитів чи недостатній захист даних [4, 5]. Це створює потребу в розробці інформаційних систем, які б відповідали сучасним вимогам користувачів, забезпечували зручність, безпеку та економічну ефективність. Розробка інтернет-магазину мобільних телефонів, орієнтованого на український ринок, із підтримкою україномовного контенту, адаптивним дизайном, розширеними фільтрами та швидкою обробкою запитів, є актуальним завданням, що сприяє розвитку електронної комерції та популяризації цифрових платформ в Україні.

Метою кваліфікаційної роботи є розробка інформаційної системи інтернет-магазину «Mobile Fan» з використанням сучасних вебтехнологій, що забезпечує автоматизацію процесів продажу мобільних телефонів, зручність для користувачів і економічну ефективність.

Відповідно до мети роботи необхідно вирішити наступні завдання:

- 1) провести аналіз предметної галузі, сучасних інформаційних систем, вимог до вебдодатків для інтернет-магазинів та аналогів;
- 2) спроектувати архітектуру бази даних для зберігання даних про товари, користувачів, замовлення, відгуки та кошик;
- 3) реалізувати інформаційну систему, включаючи базу даних, клієнтську та серверну частини, а також провести техніко-економічний аналіз її ефективності.

Об'єкт дослідження – процеси проектування інформаційних систем для електронної комерції з клієнт-серверною архітектурою.

Предмет дослідження – проектування та розробка інформаційної системи інтернет-магазину мобільних телефонів «Mobile Fan» з використанням вебтехнологій і реляційних баз даних.

Методи досліджень – дослідження базуються на методах програмної інженерії, аналізу вимог, проектування баз даних (нотація Чена, нормалізація), розробки вебдодатків із використанням клієнт-серверної архітектури, тестування (функціонального, продуктивності, безпеки), а також техніко-економічного аналізу.

Інформаційна база – Інтернет-ресурси, наукові статті, довідкові матеріали про вебтехнології (HTML, CSS, JavaScript, PHP, MySQL, OpenServer), стандарти розробки програмного забезпечення (ISO 9241-11, ISO/IEC 27001, OWASP), а також статистичні дані про ринок електронної комерції в Україні та світі.

Практична значущість – розроблена інформаційна система «Mobile Fan» забезпечує зручність для користувачів, швидкий доступ до каталогу товарів із розширеними фільтрами та безпечну обробку замовлень. Запропонована архітектура бази даних, модульна структура проекту та адаптивний інтерфейс спрощують масштабування та підтримку системи, що робить систему привабливою для малого та середнього бізнесу в сфері e-commerce.

Результати роботи апробовані в рамках наукової конференції здобувачів вищої освіти за результатами науково-дослідної роботи у 2024-2025 рр.

Структура кваліфікаційної роботи логічно пов'язана з задачами досліджень і містить перелік умовних позначень, вступ, три розділи основної частини, висновки, список використаних джерел. Загальний обсяг текстової частини кваліфікаційної роботи складає 54 сторінки формату А4. Вона містить 17 рисунків і 13 таблиць.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ПРОЄКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ ІНТЕРНЕТ-МАГАЗИНІВ

1.1 Аналіз сучасних інформаційних систем для електронної комерції

Сучасний розвиток електронної комерції зумовлює потребу в ефективних інформаційних системах (ІС), які забезпечують управління онлайн-магазинами, обробку замовлень, взаємодію з клієнтами та аналіз даних. У 2024 р. в Україні частка онлайн-продажів у роздрібній торгівлі досягла 25%, що підкреслює актуальність створення зручних і функціональних платформ для електронної комерції [6]. Інформаційні системи для інтернет-магазинів повинні відповідати вимогам швидкості, безпеки, адаптивності та зручності використання. У цьому підрозділі розглянуто основні типи ІС для електронної комерції, їх функціональні можливості, переваги та недоліки.

Системи управління контентом (CMS) є найпоширенішим рішенням для створення інтернет-магазинів. Вони дозволяють швидко розгортати вебдодатки з мінімальними затратами часу на розробку. Основними представниками є Joomla, WordPress, Drupal та спеціалізовані платформи, такі як OpenCart і Magento. Кожна з них має свої особливості, які впливають на вибір для конкретного проєкту.

Joomla – це універсальна CMS, написана на PHP з використанням бази даних MySQL. Вона підтримує широкий набір модулів, таких як каталоги товарів, системи оплати та управління замовленнями. Joomla має інтуїтивно зрозумілу адміністративну панель, що полегшує керування контентом. Проте її недоліком є вразливість до атак через відкритий код, що вимагає регулярного оновлення та додаткових заходів безпеки [7].

WordPress із плагіном WooCommerce є популярним вибором для невеликих і середніх інтернет-магазинів. Завдяки простоті налаштування та великій кількості шаблонів, WordPress підходить для проєктів із відвідуваністю

до 1000 користувачів на добу. Однак при високому навантаженні система може працювати повільно, що обмежує її використання для великих платформ [8]. У 2024 р., за даними аналітичної компанії Statista, WordPress охоплює 43% ринку CMS для електронної комерції, що підтверджує її популярність [9].

Drupal вирізняється високою продуктивністю та гнучкістю, що робить його придатним для великих порталів. Ця CMS забезпечує надійний захист даних, але складність адміністрування та потреба у висококваліфікованих розробниках ускладнюють її використання для невеликих проєктів [10].

Самописні системи створюються з нуля під конкретні потреби замовника. Вони дозволяють оптимізувати функціонал, підвищити безпеку та адаптувати систему до унікальних бізнес-процесів. Основним недоліком є висока вартість і тривалість розробки. Наприклад, для інтернет-магазину мобільних телефонів самописна CMS може включати спеціалізовані модулі фільтрації за характеристиками пристроїв, що підвищує зручність для користувачів.

Для оцінки ефективності CMS використано порівняльний аналіз за критеріями: простота використання, продуктивність, безпека та масштабованість. Результати представлено в таблиці 1.1.

Таблиця 1.1 – Порівняння CMS для електронної комерції

CMS	Простота використання	Продуктивність	Безпека	Масштабованість
Joomla	Висока	Середня	Низька	Середня
WordPress	Висока	Низька	Середня	Низька
Drupal	Низька	Висока	Висока	Висока
Самописна CMS	Залежить від розробки	Висока	Висока	Висока

Примітка: оцінки базуються на середніх показниках, отриманих із літератури та практичного досвіду [2, 3].

Сучасні ІС для електронної комерції включають такі ключові модулі: каталог товарів, кошик, система фільтрації, адміністративна панель, інтеграція з платіжними системами та аналітика продажів. Наприклад, модуль фільтрації дозволяє користувачам швидко знаходити товари за параметрами, такими як

ціна, бренд чи технічні характеристики. На рисунку 1.1 зображено схему взаємодії основних модулів ІС.

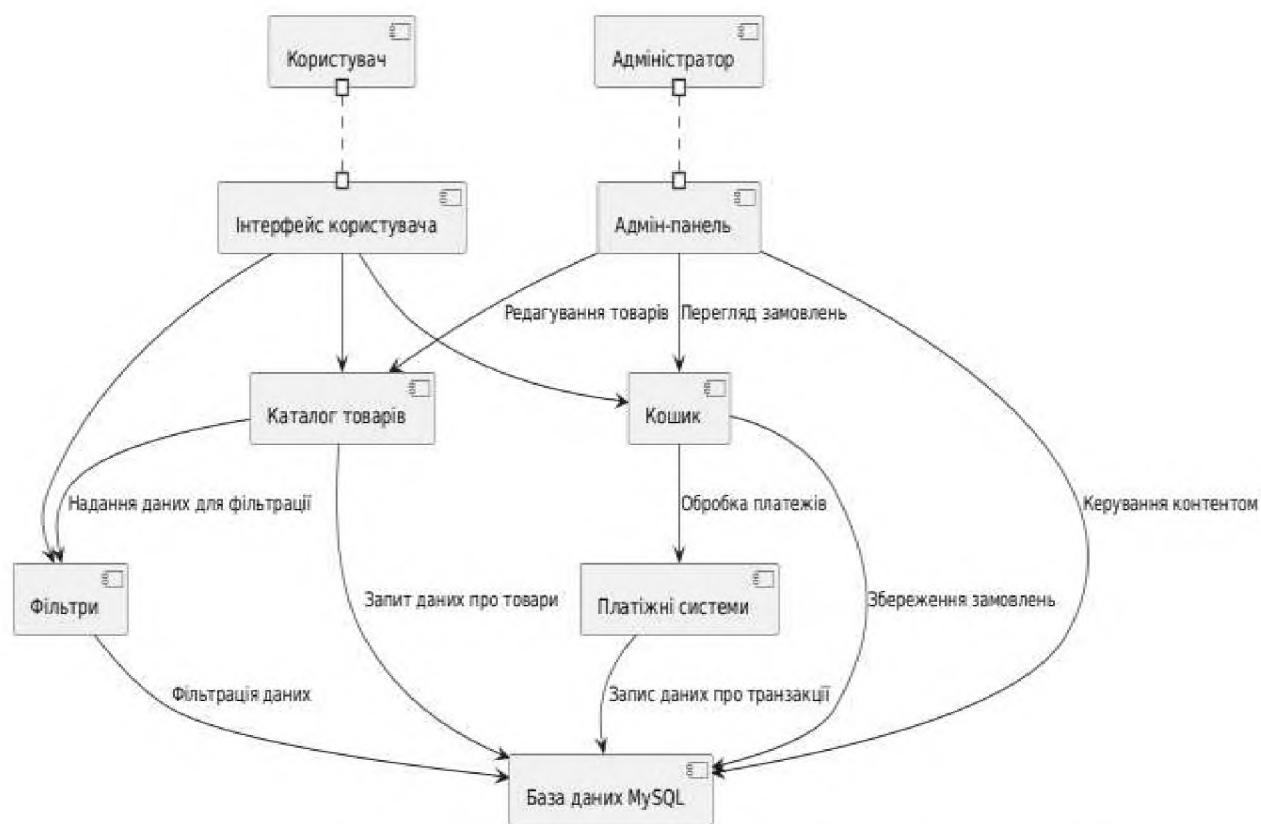


Рисунок 1.1 – Схема взаємодії модулів інформаційної системи

Тенденції розвитку ІС для електронної комерції включають інтеграцію штучного інтелекту для персоналізації пропозицій, використання хмарних технологій для підвищення масштабованості та впровадження адаптивного дизайну для мобільних пристроїв. За даними дослідження McKinsey, у 2024 р. 60% онлайн-покупок в Україні здійснюються через мобільні пристрої, що підкреслює важливість адаптивності [11].

Аналіз сучасних інформаційних систем показав, що вибір CMS залежить від масштабу проєкту, бюджету та вимог до функціональності. Для невеликих інтернет-магазинів, таких як «Mobile-Fan», WordPress або Joomla можуть бути економічно виправданими рішеннями, тоді як для складних систем із високими вимогами до безпеки та продуктивності доцільно розробляти самописну CMS.

1.2 Технологічні основи створення вебдодатків

Розвиток електронної комерції та зростання попиту на онлайн-платформи зумовлюють необхідність створення ефективних вебдодатків, які забезпечують зручність, швидкодію та безпеку для користувачів. У 2024 р. в Україні частка онлайн-продажів у роздрібній торгівлі досягла 25%, що підкреслює актуальність розробки вебдодатків для інтернет-магазинів [12]. Цей підрозділ присвячений аналізу технологічних основ створення вебдодатків, зокрема характеристики технологій PHP, MySQL, JavaScript та локального середовища OpenServer, їх перевагам, обмеженням та ролі у проєктуванні інформаційних систем для електронної комерції.

Вебдодаток – це програмне забезпечення, яке працює через веббраузер і базується на клієнт-серверній архітектурі. Клієнтська частина відповідає за інтерфейс користувача (HTML, CSS, JavaScript), тоді як серверна частина обробляє запити, взаємодіє з базою даних та генерує динамічний контент (PHP, MySQL). На рисунку 1.2 зображено схему клієнт-серверної взаємодії вебдодатка.



Рисунок 1.2 – Схема клієнт-серверної архітектури вебдодатка

Клієнт надсилає HTTP-запит до сервера, який обробляє його за допомогою серверної мови програмування, наприклад PHP, і повертає відповідь у вигляді HTML-сторінки. База даних, така як MySQL, зберігає інформацію про товари, замовлення та користувачів. Локальне середовище, наприклад OpenServer, дозволяє розробникам тестувати додаток перед розгортанням на реальному сервері.

PHP (Hypertext Preprocessor) є однією з найпоширеніших мов програмування для створення вебдодатків завдяки простоті, кросплатформності та широкій підтримці. PHP використовує синтаксис, подібний до C та Perl, що полегшує його освоєння розробниками [13]. Основні переваги PHP включають:

- кросплатформність: PHP працює на різних операційних системах (Windows, Linux, macOS) і підтримує більшість вебсерверів, таких як Apache та Nginx;

- інтеграція з базами даних: PHP забезпечує зручну взаємодію з MySQL, PostgreSQL та іншими СУБД через вбудовані бібліотеки;

- відкритий код: Безкоштовний доступ до вихідного коду сприяє швидкому розвитку спільноти та появі нових бібліотек.

Недоліки PHP включають нижчу продуктивність порівняно з компільованими мовами, такими як Go, та потребу в ретельному управлінні безпекою через вразливості, пов'язані з відкритим кодом. У 2024 р., за даними W3Techs, PHP використовується у 77% вебсайтів із серверними технологіями, що підтверджує його домінування в веброзробці [16].

Приклад PHP-коду для вибірки даних із бази даних MySQL:

```
<?php
$conn = new mysqli(«localhost», «username», «password», «database»);
if ($conn->connect_error) {
    die(«Помилка підключення: « . $conn->connect_error);
}
$sql = «SELECT id, name, price FROM products»;
$result = $conn->query($sql);
```

```

while ($row = $result->fetch_assoc()) {
    echo «ID: « . $row[«id»] . « - Назва: « . $row[«name»] . « - Ціна: « .
    $row[«price»] . «<br>«;
}
$conn->close();
?>

```

Цей код демонструє базову операцію вибірки даних із таблиці товарів, що є типовим завданням для інтернет-магазину.

MySQL – це реляційна система управління базами даних (СУБД), яка широко застосовується у вебдодатках завдяки швидкості, надійності та підтримці складних запитів. Реляційна модель дозволяє організувати дані у таблиці, пов’язані між собою ключами, що забезпечує ефективне зберігання та пошук інформації [17]. Наприклад, для інтернет-магазину «Mobile-Fan» база даних може включати таблиці products (товари), orders (замовлення) та users (користувачі). Структура таблиці products наведена в табл. 1.2.

Таблиця 1.2 – Структура таблиці products у базі даних

Поле	Тип даних	Опис
id	INT	Унікальний ідентифікатор
name	VARCHAR(255)	Назва товару
price	DECIMAL(10,2)	Ціна товару
category_id	INT	Ідентифікатор категорії

Переваги MySQL включають високу швидкість обробки запитів, підтримку транзакцій та можливість масштабування. Однак при великих обсягах даних потрібна оптимізація запитів, щоб уникнути зниження продуктивності. У 2024 р. в Україні, за оцінками Асоціації ритейлерів, 65% інтернет-магазинів використовують MySQL як основну СУБД, що свідчить про її популярність у сфері електронної комерції [1].

JavaScript відповідає за динамічну поведінку вебдодатка на стороні клієнта, наприклад, обробку подій, валідацію форм чи асинхронні запити (AJAX). Основні можливості JavaScript включають:

- маніпуляція DOM: зміна структури HTML-сторінки в реальному часі;
- обробка подій: реакція на дії користувача, такі як кліки чи введення даних;
- асинхронні запити: завантаження даних без перезавантаження сторінки.

Приклад JavaScript-коду для додавання товару до кошика:

```
document.getElementById(«add-to-cart»).addEventListener(«click», function() {
  fetch(«/add-to-cart», {
    method: «POST»,
    body: JSON.stringify({ productId: 1, quantity: 1 }),
    headers: { «Content-Type»: «application/json» }
  })
  .then(response => response.json())
  .then(data => alert(«Товар додано до кошика!»));
});
```

Недоліком JavaScript є залежність від браузера, що може спричинити несумісність із застарілими версіями. Проте сучасні фреймворки, такі як React чи Vue.js, значно спрощують розробку складних інтерфейсів. За даними Stack Overflow, у 2024 р. JavaScript залишається найпопулярнішою мовою програмування, використовуваною 62% розробників [17].

OpenServer – це програмний комплекс для створення локального серверного середовища, який включає Apache, PHP, MySQL та інструменти адміністрування, такі як phpMyAdmin. OpenServer дозволяє розробникам тестувати вебдодатки без необхідності розгортання на віддаленому сервері. Основні переваги:

- простота налаштування: інтуїтивний інтерфейс для керування сервером і базами даних;
- підтримка різних версій PHP і MySQL: дозволяє тестувати додаток у різних конфігураціях;
- портативність: OpenServer можна запускати з USB-накопичувача.

Недоліком є обмеження для масштабних проєктів, оскільки OpenServer призначений переважно для локального тестування. На рисунку 1.3 зображено інтерфейс OpenServer.

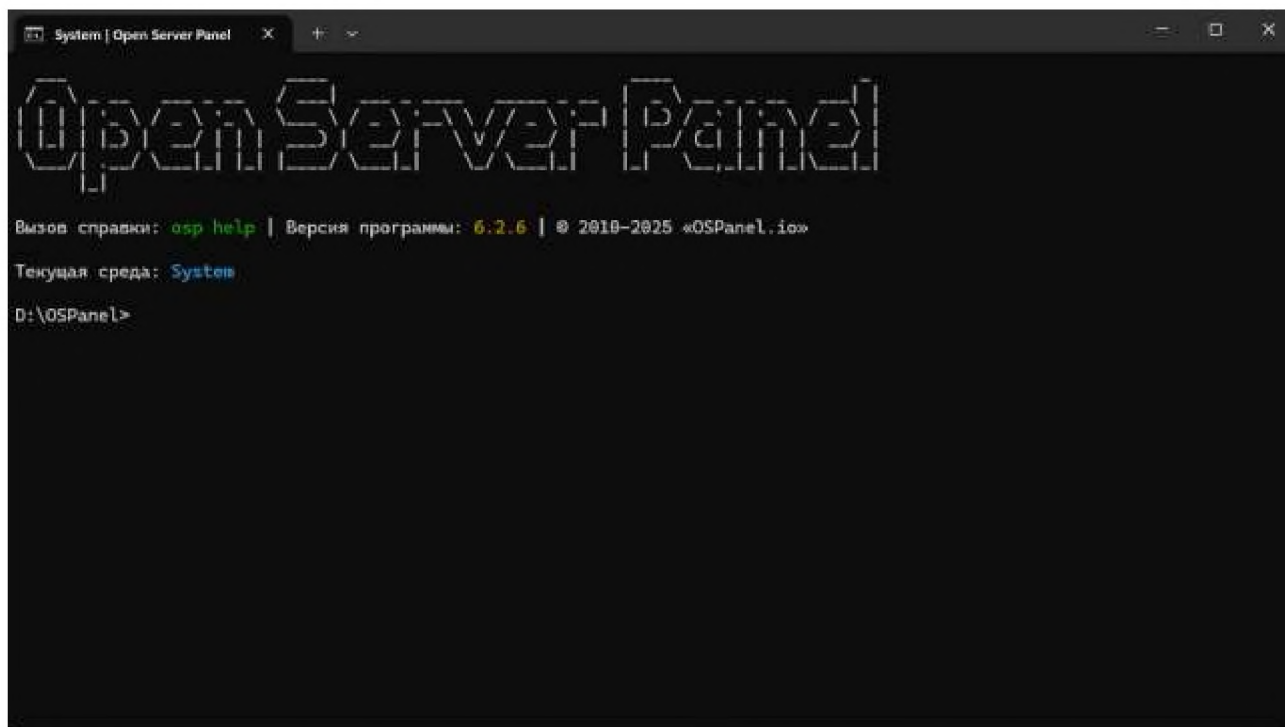


Рисунок 1.3 – Інтерфейс локального середовища OpenServer

Сучасні тенденції у веброзробці включають використання хмарних технологій, таких як AWS чи Google Cloud, для підвищення масштабованості, а також впровадження прогресивних вебдодатків (PWA), які поєднують переваги веб та мобільних додатків. За даними Gartner, у 2024 р. 30% нових вебдодатків для електронної комерції в Європі використовують PWA, що забезпечує швидший доступ для користувачів [18].

Технологічні основи створення вебдодатків базуються на комбінації серверних (PHP, MySQL) і клієнтських (JavaScript) технологій, які забезпечують функціональність, швидкодію та зручність. OpenServer є ефективним інструментом для локальної розробки, дозволяючи тестувати додатки в реальних умовах. У наступних підрозділах буде розглянуто практичну реалізацію інформаційної системи «Mobile-Fan» на основі цих технологій.

1.3 Аналіз вимог до інформаційних систем інтернет-магазинів

Інформаційні системи (ІС) для інтернет-магазинів відіграють ключову роль у забезпеченні ефективної взаємодії між продавцями та покупцями, сприяючи автоматизації бізнес-процесів і підвищенню рівня задоволеності клієнтів. У 2024 р. в Україні онлайн-продажі мобільних гаджетів склали понад 30% від загального обсягу ринку, що свідчить про зростання попиту на цифрові платформи для покупок [14]. Цей підрозділ присвячений аналізу функціональних і нефункціональних вимог до ІС інтернет-магазинів, зокрема для проєкту «Mobile-Fan», який спеціалізується на продажі мобільних телефонів. Вимоги охоплюють аспекти зручності використання, безпеки, швидкодії та адаптивності, що є критично важливими для сучасних вебдодатків.

Функціональні вимоги визначають набір можливостей, які ІС повинна надавати користувачам. Для інтернет-магазину «Mobile-Fan» ключовими модулями є каталог товарів, кошик, система фільтрації, оформлення замовлень та адміністративна панель. Кожен модуль виконує специфічні функції, спрямовані на забезпечення зручного досвіду для покупців і ефективного управління для адміністраторів.

Каталог товарів є основним інструментом для представлення асортименту. Він повинен забезпечувати швидкий доступ до інформації про товари, включаючи назву, ціну, технічні характеристики та зображення. Наприклад, для мобільних телефонів важливими є параметри, такі як обсяг пам'яті, діагональ екрана чи модель процесора. Каталог має підтримувати категоризацію товарів (наприклад, за брендами чи ціновими діапазонами) і можливість пошуку за ключовими словами.

Система фільтрації дозволяє користувачам звужувати вибір товарів за заданими параметрами. Для «Mobile-Fan» фільтри можуть включати бренд, ціну, тип операційної системи (Android, iOS) чи рік випуску. Ефективна система фільтрації зменшує час пошуку, що є важливим для користувацького досвіду. За

даними McKinsey, 70% покупців у 2024 р. використовують фільтри під час вибору товарів в інтернет-магазинах [19].

Кошик і оформлення замовлень дозволяє користувачам додавати товари, змінювати їх кількість або видаляти перед оформленням замовлення. Модуль оформлення замовлень включає введення даних про доставку, вибір способу оплати (банківська картка, накладений платіж) та підтвердження покупки. Важливо забезпечити інтеграцію з платіжними системами, такими як LiqPay чи WayForPay, для безпечної обробки транзакцій.

Адміністративна панель призначена для керування контентом і бізнес-процесами. Вона включає функції додавання/редагування товарів, перегляду замовлень, управління користувачами та аналізу продажів. Панель має бути захищеною від несанкціонованого доступу через авторизацію з використанням логіна та пароля.

Нефункціональні вимоги (табл. 1.3) визначають якісні характеристики ІС, такі як швидкодія, безпека, масштабованість та адаптивність. Вони є критично важливими для забезпечення конкурентоспроможності інтернет-магазину.

Таблиця 1.3 – Основні нефункціональні вимоги до ІС

Вимога	Опис	Показник
Швидкодія	Час завантаження сторінки	≤ 3 с
Безпека	Захист від SQL-ін'єкцій	100% запитів із підготовленими параметрами
Адаптивність	Підтримка мобільних пристроїв	100% сторінок адаптивні
Масштабованість	Максимальна кількість одночасних користувачів	≥ 1000

Швидкість завантаження сторінок є одним із ключових факторів, що впливають на задоволеність користувачів. За даними Google, затримка завантаження сторінки більше 3 секунд призводить до втрати 53% відвідувачів [20]. Для ІС «Mobile-Fan» необхідно оптимізувати запити до бази даних MySQL і використовувати кешування (наприклад, Memcached) для зменшення часу відгуку.

Формула для оцінки часу відгуку системи:

$$T_{response} = T_{query} + T_{processing} + T_{network}; \quad (1.1)$$

де T_{query} – час виконання запиту до бази даних,

$T_{processing}$ – час обробки на сервері,

$T_{network}$ – час передачі даних через мережу.

Безпека є критично важливою для захисту даних користувачів і транзакцій. Основні заходи включають:

- використання HTTPS для шифрування даних;
- захист від SQL-ін'єкцій шляхом використання підготовлених запитів у PHP;
- обмеження доступу до адмін-панелі через двофакторну автентифікацію.

ІС «Mobile-Fan» повинна використовувати технології CSS (наприклад, Flexbox або Grid) для адаптації інтерфейсу до різних розмірів екранів. Система повинна витримувати зростання кількості користувачів і транзакцій. Для цього необхідно використовувати горизонтальне масштабування бази даних (реплікація MySQL) та балансування навантаження на сервері (наприклад, через Nginx).

Юзабіліті (зручність використання) є важливим аспектом, який впливає на утримання користувачів. Основні принципи юзабіліті включають:

- простота навігації: меню та фільтри мають бути інтуїтивно зрозумілими;
- чіткість інформації: описи товарів повинні бути лаконічними та інформативними;
- швидкий доступ до кошика: кнопка кошика має бути завжди видимою.

Для оцінки юзабіліті можна використовувати метрику System Usability Scale (SUS), яка базується на опитуванні користувачів. Формула для обчислення SUS:

$$SUS = \sum_{i=1}^{10} (x_i - 1) \cdot 2,5; \quad (1.2)$$

де x_i – оцінка користувача за кожним із 10 питань (від 1 до 5).

Система вважається зручною, якщо $SUS \geq 68$ [25]. На рисунку 1.4 зображено приклад адаптивного інтерфейсу каталогу товарів.

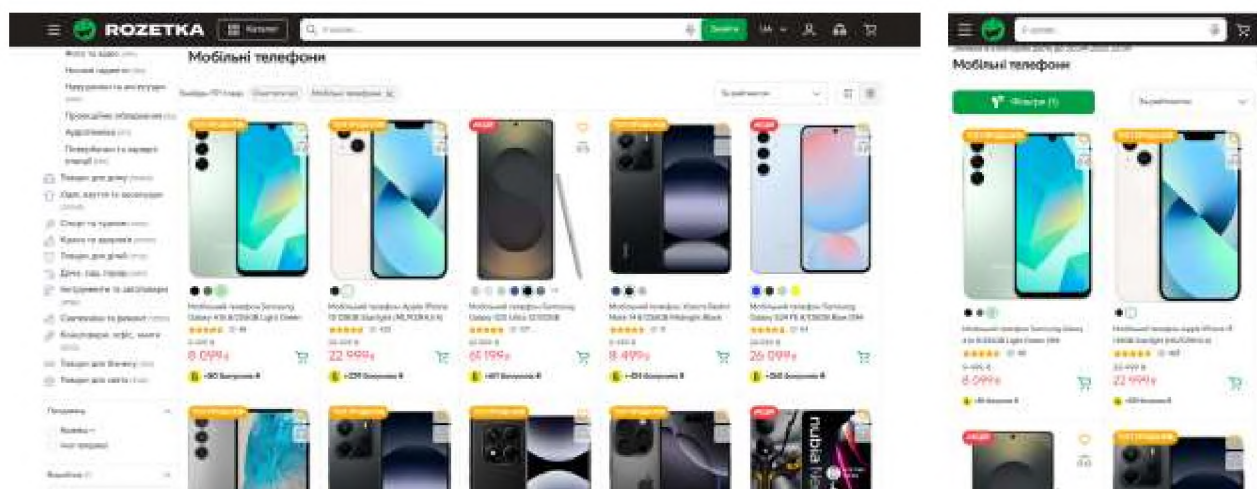


Рисунок 1.4 – Приклад адаптивного інтерфейсу каталогу товарів

Функціональні вимоги до ІС «Mobile-Fan» включають створення каталогу товарів, системи фільтрації, кошика, оформлення замовлень та адмін-панелі, що забезпечують основний функціонал інтернет-магазину. Нефункціональні вимоги, такі як швидкодія, безпека, адаптивність і масштабованість, гарантують якісну роботу системи в умовах зростання навантаження.

1.4 Аналіз предметної області

Предметна область інтернет-магазину мобільних телефонів охоплює бізнес-процеси, пов'язані з продажем товарів, управлінням асортиментом, обробкою замовлень і взаємодією з клієнтами. Аналіз предметної області дозволяє встановити функціональні вимоги до системи та забезпечити її відповідність потребам бізнесу. Бізнес-процеси інтернет-магазину «Mobile-Fan» можна поділити на три основні категорії: управління асортиментом, обробка замовлень і взаємодія з клієнтами. Кожен процес включає низку операцій, які виконуються користувачами системи та підтримуються інформаційною

системою. Опис основних бізнес-процесів інтернет-магазину наведено в табл. 1.4.

Таблиця 1.4 – Основні бізнес-процеси інтернет-магазину

Процес	Опис	Відповідальний	Вимоги до системи
Управління асортиментом	Додавання/редагування товарів	Адміністратор	Інтерфейс для керування каталогом, інтеграція з базою даних
Обробка замовлень	Оформлення, підтвердження, доставка	Покупець, адміністратор	Кошик, інтеграція з платіжними системами
Взаємодія з клієнтами	Обробка запитів, повідомлення	Адміністратор	Форма зворотного зв'язку, автоматичні повідомлення

Процес управління асортиментом охоплює додавання, редагування та видалення товарів у каталозі. Адміністратор відповідає за створення карток товарів, які містять назву, опис, ціну, технічні характеристики (наприклад, обсяг пам'яті, тип процесора) і зображення. Каталог структурується за категоріями (наприклад, бренди чи цінові діапазони) для полегшення навігації. Для підтримки актуальності асортименту система має забезпечувати швидке оновлення даних про наявність товарів. За даними Statista, у 2024 р. 60% покупців в Україні обирають товари на основі детальних описів і характеристик, що підкреслює важливість якісного управління асортиментом [24].

Процес обробки замовлень починається з вибору товару покупцем і додавання його до кошика. Далі користувач оформляє замовлення, вказуючи дані про доставку (адреса, спосіб доставки) і спосіб оплати (банківська картка, накладений платіж). Система генерує замовлення, яке надходить до адміністратора для обробки. Адміністратор перевіряє наявність товару, підтверджує замовлення та організовує доставку. Після виконання замовлення інформація про нього зберігається в базі даних для подальшого аналізу. Ефективність цього процесу залежить від швидкості обробки запитів і інтеграції з платіжними системами, такими як LiqPay чи WayForPay.

Процес взаємодії з клієнтами включає надання інформації про товари, обробку запитів і скарг, а також підтримку післяпродажного обслуговування. Покупці можуть звертатися до служби підтримки через форму зворотного зв'язку або чат. Система має забезпечувати автоматичне надсилання повідомлень про статус замовлення (наприклад, «Замовлення відправлено»). За даними McKinsey, 65% покупців у 2024 р. очікують оперативної відповіді на свої запити протягом 1 години, що вимагає автоматизації комунікаційних процесів [26].

Для оцінки ефективності бізнес-процесів можна використовувати метрику часу виконання операції $T_{\text{operation}}$. Оптимізація цього показника підвищує задоволеність користувачів.

$$T_{\text{operation}} = T_{\text{user}} + T_{\text{system}} + T_{\text{admin}}; \quad (1.3)$$

де T_{user} – час, витрачений покупцем на вибір товару та оформлення замовлення,

T_{system} – час обробки запиту системою,

T_{admin} – час, потрібний адміністратору для підтвердження та виконання замовлення.

Інформаційна система «Mobile-Fan» має двох основних типів користувачів: покупців і адміністраторів. Кожен тип має свої завдання, вимоги до системи та сценарії взаємодії. Покупці – це кінцеві користувачі, які відвідують інтернет-магазин для пошуку, вибору та придбання мобільних телефонів. Їхні основні дії включають:

- перегляд каталогу товарів і використання фільтрів (наприклад, за брендом чи ціною);
- додавання товарів до кошика та оформлення замовлень;
- звернення до служби підтримки для отримання консультацій.

Покупці очікують простого та інтуїтивно зрозумілого інтерфейсу, швидкого завантаження сторінок і безпечної обробки платежів. За даними

Google, 53% користувачів залишають сайт, якщо сторінка завантажується довше 3 секунд, що підкреслює важливість швидкодії для цього типу користувачів [27].

Адміністратори відповідають за управління контентом і бізнес-процесами. Для адміністраторів система має надавати зручну адміністративну панель із захищеним доступом через авторизацію. Важливим є також забезпечення можливості експорту даних (наприклад, звітів про продажі) у формати CSV або Excel. Їхні завдання включають:

- додавання та редагування товарів у каталозі;
- обробку замовлень (підтвердження, організація доставки);
- аналіз даних про продажі та взаємодію з клієнтами.

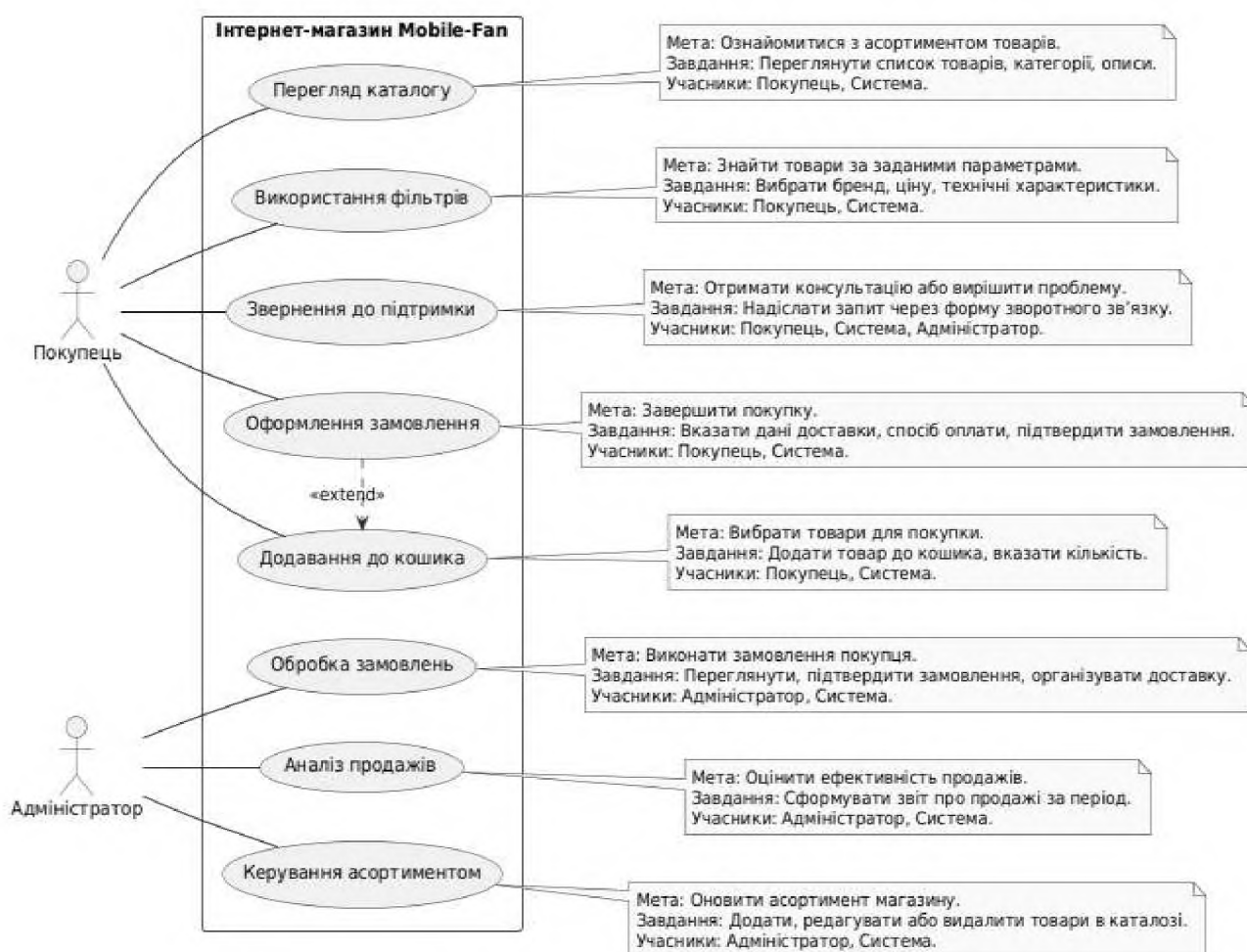


Рисунок 1.5 – Use-Case діаграма інформаційної системи «Mobile-Fan»

На рисунку 1.5 зображено Use-Case діаграму системи «Mobile-Fan». Use-Case діаграма відображає функціональні можливості системи з точки зору користувачів. Основні сценарії для покупця:

1. Перегляд каталогу: покупець відкриває сторінку каталогу, де відображаються товари з можливістю сортування за категоріями.
2. Використання фільтрів: покупець задає параметри (наприклад, ціна чи бренд) для звуження вибору.
3. Додавання до кошика: покупець додає товари до кошика, вказуючи кількість.
4. Оформлення замовлення: покупець заповнює форму з даними про доставку та оплату.
5. Звернення до підтримки: покупець надсилає запит через форму зворотного зв'язку.

Для адміністратора:

1. Керування асортиментом: додавання, редагування або видалення товарів у базі даних.
2. Обробка замовлень: перегляд, підтвердження та організація доставки замовлень.
3. Аналіз продажів: формування звітів про продажі за період.

Діаграма допомагає чітко визначити межі системи та її взаємодію з користувачами, що є основою для подальшого проєктування. Аналіз предметної області інтернет-магазину «Mobile-Fan» показав, що основними бізнес-процесами є управління асортиментом, обробка замовлень і взаємодія з клієнтами. Ключові користувачі – покупці та адміністратори – мають різні завдання, які система повинна підтримувати через відповідні функціональні модулі. Use-Case діаграма ілюструє основні сценарії використання, що слугуватиме основою для розробки системи.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ MOBILE-FAN

2.1 Вибір та обґрунтування засобів розробки

Вибір засобів розробки інформаційної системи (ІС) для інтернет-магазину «Mobile-Fan» є ключовим етапом, який визначає її функціональність, безпеку та ефективність. Порівняння готових систем управління контентом (CMS) із самописними рішеннями дозволяє обґрунтувати вибір самописної CMS на основі PHP, MySQL і OpenServer, а також проаналізувати їх відповідності вимогам проєкту.

Для створення ІС розглянуто два підходи: використання готових CMS (наприклад, Joomla, WordPress із плагіном WooCommerce, OpenCart) і розробка самописної системи. Готові CMS забезпечують швидке розгортання завдяки модульній структурі та широкому вибору шаблонів. Наприклад, OpenCart пропонує базові функції для інтернет-магазинів, такі як каталог товарів і обробка замовлень. Однак такі системи мають обмежену гнучкість для реалізації специфічних вимог, наприклад, розширених фільтрів за технічними характеристиками мобільних телефонів. Крім того, готові CMS можуть бути вразливими до кібератак через відкритий код, що потребує постійного оновлення [28].

Таблиця 2.1 – Порівняння готових CMS і самописної системи

Критерій	Готові CMS	Самописна CMS
Час розробки	Низький	Високий
Гнучкість	Обмежена	Висока
Безпека	Середня	Висока
Витрати	Низькі	Високі

Самописна CMS дозволяє створювати систему, точно адаптовану до потреб «Mobile-Fan», включаючи унікальні модулі, такі як детальна фільтрація за параметрами пристроїв. Хоча розробка самописної системи є більш

трудомісткою і потребує більших початкових витрат, вона забезпечує вищий рівень безпеки, продуктивності та гнучкості. З огляду на вимоги до спеціалізованого функціоналу та безпеки, для «Mobile-Fan» обрано самописну CMS.

Для реалізації самописної CMS обрано PHP як серверну мову програмування, MySQL як систему управління базами даних і OpenServer як локальне середовище тестування. PHP забезпечує ефективну обробку запитів і зручну інтеграцію з MySQL, що необхідно для створення динамічного каталогу товарів і управління замовленнями. MySQL дозволяє організувати реляційні таблиці для зберігання даних про товари, користувачів і транзакції, забезпечуючи швидкий доступ і масштабованість. OpenServer, завдяки вбудованим інструментам, таким як Apache і phpMyAdmin, спрощує локальне тестування та налагодження системи.

Для клієнтської частини використовується JavaScript, який забезпечує асинхронне оновлення даних і покращення взаємодії з інтерфейсом, наприклад, через динамічне завантаження фільтрів. За даними Stack Overflow, у 2024 р. JavaScript використовується 62% розробників для створення інтерактивних вебдодатків, що підтверджує його актуальність [29]. На рисунку 2.1 зображено схему технологічного стеку системи.

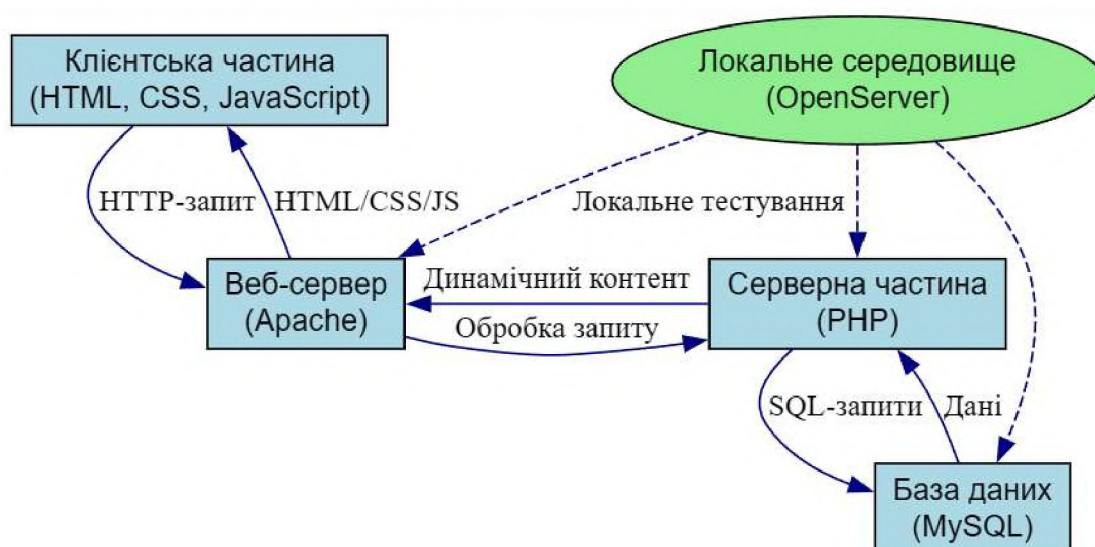


Рисунок 2.1 – Схема технологічного стеку інформаційної системи «Mobile-Fan»

Вибір самописної CMS для «Mobile-Fan» обґрунтовано необхідністю гнучкості, безпеки та реалізації специфічних функцій, таких як розширені фільтри. PHP, MySQL, JavaScript і OpenServer забезпечують оптимальний технологічний стек для створення ефективної ІС.

2.2 Розробка обмежень цілісності даних, індексів та тригерів

Розробка структури бази даних (БД) є ключовим етапом проєктування інформаційної системи (ІС) інтернет-магазину «Mobile-Fan», оскільки від правильної організації даних залежить ефективність роботи системи, швидкість обробки запитів і зручність управління інформацією. Структура БД має бути повною, нормалізованою, уникати надмірності даних і забезпечувати швидкий доступ до інформації. У цьому підрозділі описано процес створення інфологічної та концептуальної моделі БД, визначено сутності, атрибути, зв'язки, а також представлено діаграму «сутність-зв'язок» у нотації Чена з урахуванням принципів нормалізації.

Інформаційна система «Mobile-Fan» призначена для автоматизації процесів продажу мобільних телефонів через Інтернет, включаючи управління каталогом, обробку замовлень, авторизацію користувачів і підтримку зворотного зв'язку. Основна мета БД – забезпечити зберігання та швидкий доступ до даних про товари, користувачів, замовлення, категорії та відгуки, а також підтримувати аналітичні запити для аналізу продажів. БД має відповідати вимогам ефективності, безпеки та масштабованості, щоб підтримувати зростання кількості користувачів і товарів. Для визначення структури БД сформульовано основні прецеденти використання системи:

1. Реєстрація користувача – занесення даних нового користувача (логін, пароль, email).
2. Оформлення замовлення – створення замовлення з інформацією про товари, користувача, дату та статус.

3. Перегляд каталогу – відображення товарів із фільтрацією за брендом, ціною чи характеристиками.

4. Додавання товару до кошика – занесення даних про вибрані товари для певного користувача.

5. Залишення відгуку – додавання оцінки та коментаря до товару.

6. Аналіз продажів – генерація звіту про найпопулярніші товари за кількістю замовлень.

7. Оновлення статусу замовлення – зміна статусу (наприклад, «обробляється», «відправлено») адміністратором.

8. Управління каталогом – додавання, редагування або видалення товарів адміністратором.

Додатковий опис прецедентів БД наступний:

1. Реєстрація користувача: мета – додати нового користувача до системи. Завдання: ввести логін, email, зашифрований пароль; перевірити унікальність email; зберегти дані. Учасники: користувач, система.

2. Оформлення замовлення: мета – зафіксувати покупку. Завдання: обрати товари з кошика, вказати спосіб доставки, зберегти дату та статус замовлення. Учасники: користувач, система.

3. Перегляд каталогу: мета – надати доступ до товарів. Завдання: відобразити список товарів із фільтрами (бренд, ціна) або результатами пошуку. Учасники: користувач, система.

4. Додавання товару до кошика: мета – зберегти вибір користувача. Завдання: додати ідентифікатор товару та користувача до кошика. Учасники: користувач, система.

5. Залишення відгуку: мета – зберегти оцінку товару. Завдання: ввести рейтинг (1-5), коментар, пов'язати з товаром і користувачем. Учасники: користувач, система.

Для визначення сутностей, атрибутів і зв'язків проведено морфологічний аналіз вимог, ідентифікуючи іменники як кандидати в сутності та дієслова як зв'язки.

Іменники – сутності та атрибути (таблиця 2.2):

- користувач: людина, яка реєструється та робить замовлення; атрибути: ідентифікатор, логін, пароль, email;
- товар: мобільний телефон, доступний для покупки; атрибути: ідентифікатор, назва, бренд, опис, ціна, зображення, категорія;
- замовлення: процес покупки; атрибути: ідентифікатор, дата, статус, загальна сума;
- кошик: тимчасовий набір товарів користувача; атрибути: ідентифікатор товару, ідентифікатор користувача, кількість;
- категорія: класифікація товарів (наприклад, бренд чи тип); атрибути: ідентифікатор, назва;
- відгук: оцінка товару користувачем; атрибути: ідентифікатор, рейтинг, коментар, дата.

Таблиця 2.2 – Сутності та їх атрибути

Сутність	Атрибути
Користувач	id (INT, PK), login (VARCHAR), password (VARCHAR), email (VARCHAR)
Товар	id (INT, PK), name (VARCHAR), brand (VARCHAR), description (TEXT), price (DECIMAL), image (VARCHAR), category_id (INT, FK)
Категорія	id (INT, PK), name (VARCHAR)
Замовлення	id (INT, PK), user_id (INT, FK), date (DATETIME), status (VARCHAR), total (DECIMAL)
Кошик	product_id (INT, FK), user_id (INT, FK), quantity (INT)
Відгук	id (INT, PK), product_id (INT, FK), user_id (INT, FK), rating (INT), comment (TEXT), date (DATETIME)

Дієслова – зв'язки (таблиця 2.3):

- користувач оформляє замовлення;
- користувач додає товар до кошика;
- товар належить до категорії;
- користувач залишає відгук про товар;
- замовлення містить товари.

Таблиця 2.3 – Зв'язки між сутностями

Сутність 1	Зв'язок	Сутність 2	Опис
Користувач	оформляє	Замовлення	Один користувач може мати багато замовлень
Користувач	додає до	Кошик	Користувач додає товари до кошика
Товар	належить до	Категорія	Один товар належить до однієї категорії
Користувач	залишає	Відгук	Користувач може залишити відгук про товар
Замовлення	містить	Товар	Замовлення включає один або більше товарів

На основі морфологічного аналізу розроблено концептуальну модель БД у нотації Чена, яка відображає сутності, їх атрибути та зв'язки з урахуванням кардинальності. На рисунку 2.2 зображено діаграму «сутність-зв'язок» для БД.

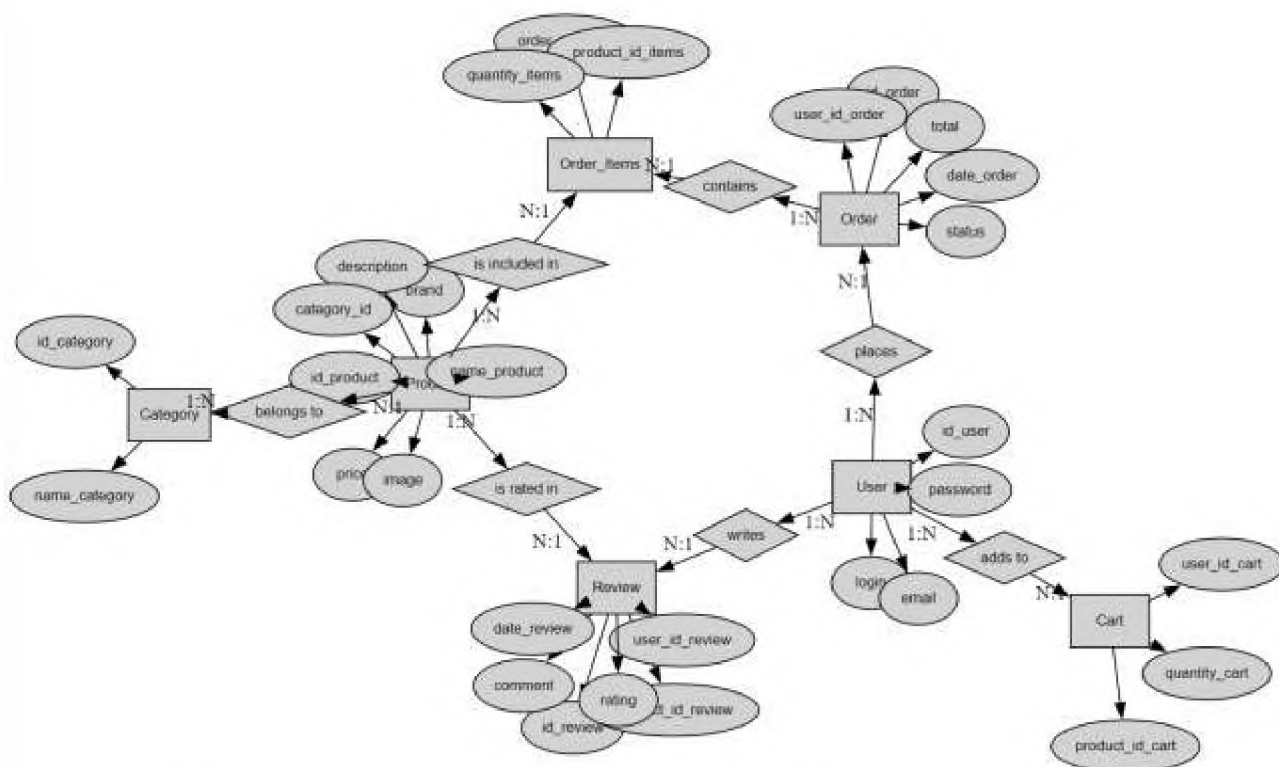


Рисунок 2.2 – Діаграма «сутність-зв'язок» для БД інтернет-магазину «Mobile-Fan»

Основні зв'язки:

- User – Order: один до багатьох (1:N); один користувач може мати багато замовлень;
- Product – Category: багато до одного (N:1); товар належить до однієї категорії;

- User – Cart: один до багатьох (1:N); користувач має один кошик із кількома записами;
- Order – Product: багато до багатьох (N:M); реалізовано через проміжну таблицю Order_Items.

Для усунення надмірності даних і забезпечення цілісності проведено нормалізацію до третьої нормальної форми (3NF):

1. Перша нормальна форма (1NF): усі атрибути атомарні, таблиці не містять повторюваних груп. Наприклад, таблиця Product має окремі поля для назви, бренду та ціни.

2. Друга нормальна форма (2NF): усунуто часткову залежність. Наприклад, у таблиці Order атрибут total залежить від повного ключа (order_id).

3. Третя нормальна форма (3NF): усунуто транзитивні залежності. Наприклад, поле category_id у таблиці Product посилається на таблицю Category.

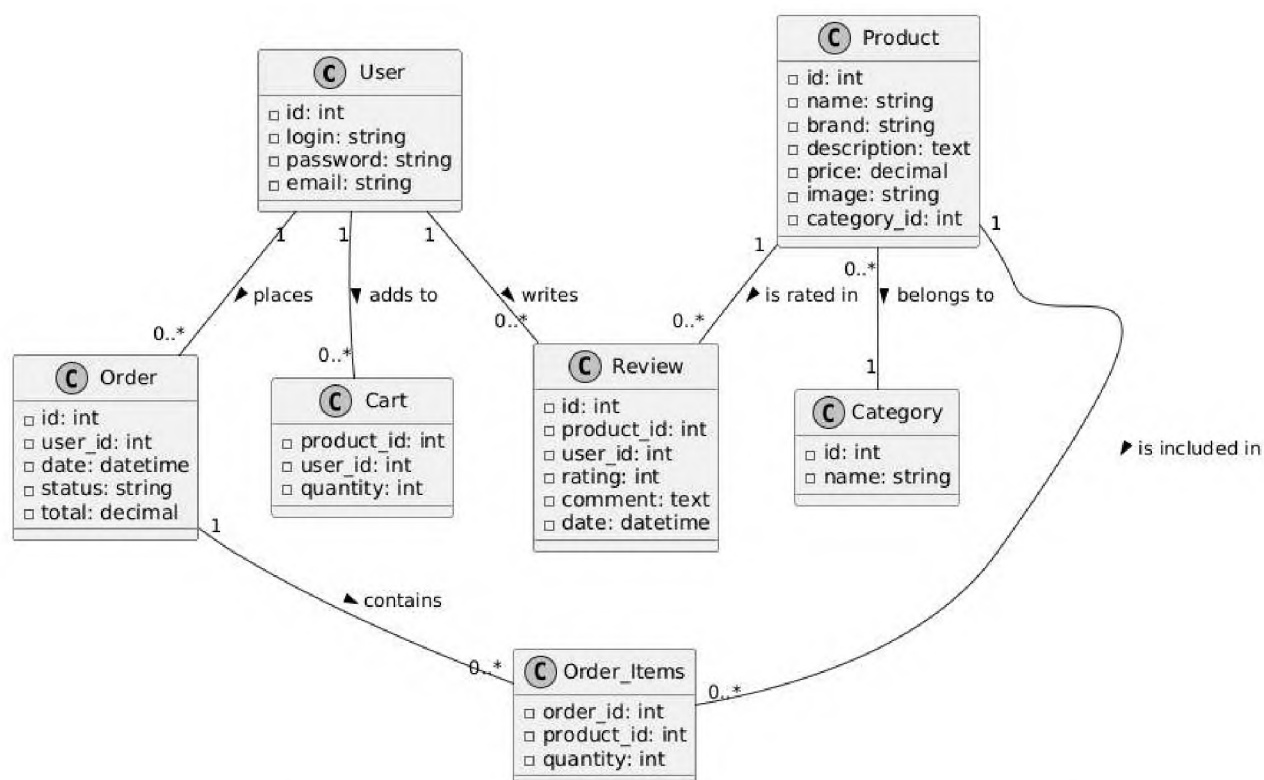


Рисунок 2.3 – EER модель БД інтернет-магазину «Mobile-Fan»

Для підвищення продуктивності передбачено індексацію полів, таких як name і brand у таблиці Product, а також використання зовнішніх ключів для референційної цілісності.

На основі діаграми «сутність-зв'язок» розроблено EER модель, яка використовує англійські назви сутностей (User, Product, Category, Order, Cart, Review, Order_Items) і зв'язків (places, adds to, belongs to, writes, contains). На рисунку 2.3 зображено EER модель БД. Розроблена структура БД включає шість основних сутностей і проміжну таблицю Order_Items для реалізації зв'язку N:M між Order і Product. Діаграма «сутність-зв'язок» і EER модель відображають зв'язки з урахуванням кардинальності, а нормалізація до 3NF забезпечує ефективність і цілісність даних. Структура відповідає вимогам ІС «Mobile-Fan», підтримуючи швидкий доступ до даних і аналітичні функції.

2.3 Проектування структури сайту і опис функціональних модулів системи

Проектування структури сайту та функціональних модулів ІС інтернет-магазину «Mobile-Fan» є важливим етапом розробки, який визначає зручність використання, ефективність взаємодії користувачів із системою та її відповідність бізнес-вимогам. Для забезпечення зручності користувачів структура сайту розроблена з урахуванням принципів юзабіліті, адаптивності та швидкодії.

Структура сайту «Mobile-Fan» розроблена для забезпечення простої навігації та швидкого доступу до ключових функцій. Сайт складається з основних сторінок, кожна з яких виконує певну роль у взаємодії з користувачами. Основні сторінки включають:

1. Головна сторінка – вітрина магазину, яка презентує популярні товари, акції та категорії. Мета – залучити користувачів і спрямувати їх до каталогу.

2. Каталог товарів – сторінка з переліком мобільних телефонів, що підтримує фільтрацію за брендом, ціною чи технічними характеристиками.

3. Сторінка товару – детальна інформація про конкретний товар, включаючи опис, характеристики, ціну, зображення та відгуки.

4. Кошик – сторінка для перегляду обраних товарів, зміни їх кількості та переходу до оформлення замовлення.

5. Оформлення замовлення – форма для введення даних про доставку та оплату.

6. Особистий кабінет – сторінка для зареєстрованих користувачів, де відображаються історія замовлень, налаштування профілю та збережені товари.

7. Сторінка авторизації/реєстрації – форми для входу в систему або створення нового облікового запису.

8. Сторінка зворотного зв'язку – форма для надсилання запитів до служби підтримки.

9. Адміністративна панель – інтерфейс для адміністраторів, доступний лише після авторизації, для управління товарами, замовленнями та користувачами.

На рисунку 2.4 зображено структурну схему сайту.



Рисунок 2.4 – Структурна схема сайту «Mobile-Fan»

Структура сайту розроблена з урахуванням принципів юзабіліті, описаних у стандарті ISO 9241-11, які передбачають ефективність, результативність і задоволеність користувачів [30]. Для забезпечення адаптивності використано CSS-технології, такі як Flexbox і Grid, що дозволяють коректно відображати сайт на різних пристроях. За даними Statista, у 2024 р. 60% онлайн-покупок в Україні здійснюються через смартфони, що підкреслює важливість адаптивного дизайну [4].

Функціональні модулі ІС «Mobile-Fan» розроблені для підтримки основних бізнес-процесів, таких як продаж товарів, обробка замовлень і взаємодія з клієнтами. Кожен модуль реалізує певний набір функцій, які відповідають прецедентам використання, описаним у попередніх підрозділах.

Модуль авторизації та реєстрації забезпечує доступ користувачів до системи через авторизацію та створення нових облікових записів. Основні функції:

- реєстрація: введення логіна, email і пароля. пароль шифрується за допомогою алгоритму bcrypt для забезпечення безпеки;
- авторизація: перевірка логіна та пароля, створення сесії для авторизованого користувача;
- відновлення пароля: надсилання листа з посиланням для скидання пароля.

Модуль використовує PHP для серверної обробки та JavaScript для валідації форм на клієнтській стороні. Для захисту від SQL-ін'єкцій застосовуються підготовлені запити, як показано нижче:

```
$stmt = $conn->prepare(«SELECT id, password FROM users WHERE email = ?»);
$stmt->bind_param(«s», $email);
$stmt->execute();
```

За даними OWASP, у 2024 р. 15% кібератак на вебдодатки пов'язані з помилками авторизації, що підкреслює важливість безпечної реалізації цього модуля [31].

Модуль каталогу товарів дозволяє користувачам переглядати асортимент мобільних телефонів. Основні функції:

- відображення товарів у вигляді сітки або списку з інформацією про назву, бренд, ціну та зображення;
- пагінація для обмеження кількості товарів на сторінці (наприклад, 12 товарів на сторінку);
- пошук за ключовими словами (наприклад, «Samsung» або «iPhone»).

Модуль використовує SQL-запити для вибірки даних із таблиці Product, наприклад:

```
SELECT id, name, brand, price, image FROM Product WHERE name LIKE ? LIMIT 12 OFFSET ?;
```

Для оптимізації швидкодії застосовано індексацію полів name і brand. Модуль також підтримує кешування даних за допомогою Memcached, що зменшує час відгуку системи.

Модуль фільтрації дозволяє звужувати вибір товарів за параметрами, такими як бренд, ціна, обсяг пам'яті чи операційна система. Основні функції:

- вибір кількох значень фільтрів (наприклад, бренди Samsung і Apple);
- динамічне оновлення результатів за допомогою AJAX-запитів;
- збереження обраних фільтрів у сесії користувача.

Фільтрація реалізується через складні SQL-запити з умовами, наприклад:

```
SELECT * FROM Product WHERE brand IN (?, ?) AND price BETWEEN ? AND ?;
```

За даними McKinsey, 70% покупців у 2024 р. використовують фільтри під час вибору товарів в інтернет-магазинах, що підтверджує важливість цього модуля [9].

Модуль кошика дозволяє користувачам додавати товари, змінювати їх кількість і видаляти перед оформленням замовлення. Основні функції:

- додавання товару до кошика через асинхронний запит (JavaScript Fetch API);
- відображення загальної суми та кількості товарів;
- збереження даних кошика в таблиці Cart.

Приклад JavaScript-коду для додавання товару:

```
fetch('/add-to-cart', {
  method: 'POST',
  body: JSON.stringify({ productId: 1, quantity: 1 }),
  headers: { 'Content-Type': 'application/json' }
})
.then(response => response.json())
.then(data => alert('Товар додано до кошика!'));
```

Модуль оформлення замовлень дозволяє користувачам завершувати покупку, вводячи дані про доставку та оплату. Основні функції:

- вибір способу доставки (наприклад, Нова Пошта, Укрпошта);
- вибір способу оплати (банківська картка, накладений платіж);
- збереження замовлення в таблиці Order і Order_Items.

Модуль інтегрується з платіжними системами, такими як LiqPay, через API. Для забезпечення безпеки транзакцій використовується HTTPS.

Модуль відгуків дозволяє користувачам залишати оцінки та коментарі до товарів. Основні функції:

- додавання відгуку з рейтингом (1–5) і текстом;
- відображення середнього рейтингу товару;
- модерація відгуків адміністратором.

Дані зберігаються в таблиці Review. Для підрахунку середнього рейтингу використовується SQL-запит:

```
SELECT AVG(rating) FROM Review WHERE product_id = ?;
```

Модуль зворотного зв'язку забезпечує взаємодію користувачів із службою підтримки. Основні функції:

- форма для надсилання запитів (тема, повідомлення, email);
- автоматичне надсилання підтвердження користувачу;
- збереження запитів у базі даних для обробки адміністратором.

Модуль використовує PHPMailer для надсилання листів. За даними Google, 65% користувачів очікують відповіді на запит протягом 1 години, що вимагає автоматизації цього процесу [10].

Адміністративна панель призначена для управління контентом і бізнес-процесами. Основні функції:

- додавання, редагування та видалення товарів;
- перегляд і обробка замовлень (зміна статусу, наприклад, «відправлено»);
- аналіз продажів через звіти (наприклад, найпопулярніші товари);
- управління користувачами (блокування, зміна ролей).

Панель захищена двофакторною автентифікацією. Для генерації звітів використовується SQL-запит:

```
SELECT p.name, COUNT(oi.product_id) as order_count
FROM Product p
JOIN Order_Items oi ON p.id = oi.product_id
GROUP BY p.id
ORDER BY order_count DESC;
```

На рисунку 2.5 зображено схему взаємодії функціональних модулів.

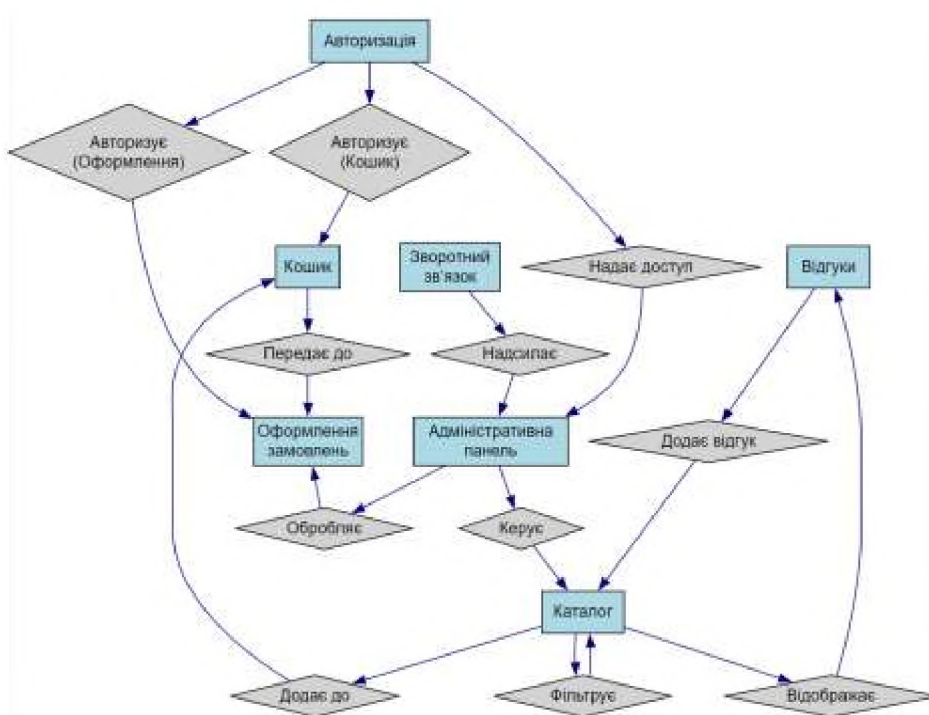


Рисунок 2.5 – Схема взаємодії функціональних модулів системи «Mobile-Fan»

Структура сайту «Mobile-Fan» включає основні сторінки, які забезпечують зручну навігацію та доступ до ключових функцій. Функціональні модулі – авторизація, каталог, фільтрація, кошик, оформлення замовлень, відгуки, зворотний зв'язок і адміністративна панель – підтримують основні бізнес-процеси магазину. Використання PHP, MySQL, JavaScript і принципів юзабіліті забезпечує ефективність і зручність системи.

2.4 Аналіз вимог до інтерфейсу та безпеки

Розробка ІС інтернет-магазину «Mobile-Fan» вимагає ретельного аналізу вимог до інтерфейсу користувача та безпеки, оскільки ці аспекти безпосередньо впливають на зручність використання, довіру клієнтів і захист даних. Інтерфейс користувача ІС «Mobile-Fan» має відповідати принципам юзабіліті, визначеним стандартом ISO 9241-11, які включають ефективність, результативність і задоволеність користувачів [13].

Таблиця 2.4 – Вимоги до інтерфейсу

Вимога	Опис	Технологія
Навігація	Чітке меню, логічні переходи	HTML, CSS
Адаптивність	Коректне відображення на всіх пристроях	Flexbox, Grid
Швидкодія	Час завантаження < 3 с	Кешування, оптимізація зображень
Доступність	Підтримка WCAG 2.1	Альтернативний текст, клавіатурна навігація

Основні вимоги до інтерфейсу наведені у таблиці 2.4 і включають:

1. Простота навігації – меню має бути логічно структурованим, із чіткими назвами пунктів (наприклад, «Каталог», «Кошик», «Особистий кабінет»). За даними Nielsen, 80% користувачів залишають сайт, якщо навігація є незручною [11].

2. Адаптивність – інтерфейс повинен коректно відображатися на пристроях із різними розмірами екранів (смартфони, планшети, ПК). Для цього використовуються CSS-технології, такі як Flexbox і Grid. У 2024 р. 60% онлайн-покупок в Україні здійснюються через мобільні пристрої, що підтверджує необхідність адаптивного дизайну [4].

3. Швидкодія – час завантаження сторінок не повинен перевищувати 3 секунди, оскільки, за даними Google, 53% користувачів залишають сайт із повільним завантаженням [10].

4. Візуальна привабливість – інтерфейс має використовувати сучасний дизайн із гармонійною кольоровою палітрою та читабельними шрифтами (наприклад, Arial, розмір 16 пікселів для основного тексту).

5. Доступність – система повинна відповідати стандартам WCAG 2.1, включаючи підтримку клавіатурної навігації та альтернативний текст для зображень [21].

На рисунку 2.6 зображено макет головної сторінки сайту.



Рисунок 2.6 – Макет головної сторінки «Mobile-Fan»

Безпека ІС «Mobile-Fan» є критично важливою через обробку конфіденційних даних, таких як персональна інформація користувачів і платіжні дані. Основні вимоги до безпеки базуються на стандарті ISO/IEC 27001 і рекомендаціях OWASP [12, 14] і показані у таблиці 2.5.

Таблиця 2.5 – Вимоги до безпеки

Вимога	Опис	Технологія
Авторизація	Шифрування паролів, 2FA	bcrypt, OTP
SQL-ін'єкції	Підготовлені запити	PDO, MySQLi
Шифрування	HTTPS, PCI DSS	SSL/TLS, LiqPay
XSS-захист	Екранування, CSP	htmlspecialchars, CSP

Для забезпечення безпеки застосовується надійна авторизація з шифруванням паролів за допомогою алгоритму bcrypt, двофакторною автентифікацією (2FA) для адміністраторів і захистом від атак типу brute-force шляхом обмеження кількості спроб входу, наприклад, до п'яти за десять хвилин. Захист від SQL-ін'єкцій досягається використанням підготовлених запитів у PHP, що запобігає введенню шкідливого коду. Усі з'єднання захищені протоколом HTTPS із сертифікатом SSL/TLS, а платіжні дані обробляються через інтегровані платіжні шлюзи, такі як LiqPay, що відповідають стандарту PCI DSS. Для захисту від XSS-атак вхідні дані користувачів, наприклад відгуки, екрануються за допомогою функцій PHP, таких як htmlspecialchars(), а також застосовується Content Security Policy (CSP) для обмеження джерел скриптів. Усі дії користувачів і адміністраторів, зокрема авторизація та редагування товарів, записуються в лог-файли, а система виявлення вторгнень (IDS) забезпечує моніторинг підозрілих активностей.

Інтерфейс ІС «Mobile-Fan» має бути простим, адаптивним і швидкодійним, відповідаючи стандартам юзабіліті та доступності. Вимоги до безпеки включають захист даних, безпечну авторизацію та запобігання кібератакам. Реалізація цих вимог із використанням сучасних технологій забезпечить зручність і довіру користувачів. У наступному розділі буде розглянуто практичну реалізацію інтерфейсу та захисних механізмів.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ MOBILE-FAN

3.1 Розробка вебдодатка

Розробка вебдодатка є ключовим етапом створення сучасних інформаційних систем, що забезпечують зручний доступ до послуг через Інтернет. У даному підрозділі розглядається процес налаштування середовища розробки з використанням OpenServer та кодування основних модулів системи на мовах програмування PHP і JavaScript. Ці технології обрано через їхню поширеність, гнучкість та широку підтримку в розробці вебдодатків для електронної комерції, зокрема для платформ продажу мобільних телефонів, як у проекті «Mobile-fan» [1].

OpenServer є популярним програмним комплексом для створення локального серверного середовища, яке включає вебсервер Apache, інтерпретатор PHP, СУБД MySQL та інші інструменти, необхідні для розробки вебдодатків. Налаштування OpenServer дозволяє розробникам імітувати реальне серверне середовище на локальному комп'ютері, що значно спрощує тестування та відлагодження коду.

Для початку необхідно завантажити останню версію OpenServer з офіційного сайту [2]. У 2025 р. рекомендованою є версія 6.2.4 або новіша, яка підтримує PHP 8.4 та MySQL 8.4. Після завантаження інсталятор запускається, і користувач обирає директорію для встановлення, наприклад, D:\OSPanel. Важливо переконатися, що на диску достатньо вільного місця (не менше 5 ГБ).

Після встановлення OpenServer запускається через ярлик на робочому столі. Основні налаштування виконуються через панель керування OpenServer:

1. Вибір версії PHP – для проекту «Mobile-fan» обрано PHP 8.4 через його покращену продуктивність і підтримку сучасних функцій, таких як JIT-

компіляція та типізовані властивості [3]. У меню налаштувань OpenServer у вкладці «Модулі» обирається PHP 8.4.

2. Налаштування MySQL – використовується MySQL 8.4 для забезпечення сумісності з PHP. У вкладці «Сервер» вказується порт (зазвичай 3306) та перевіряється підключення до СУБД.

Для зберігання даних про товари, користувачів і замовлення створюється база даних mobile_fan у MySQL. Використовується SQL-запит для ініціалізації:

```
CREATE DATABASE mobile_fan CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

Далі імпортується SQL-дамп (mobile_fan.sql), який містить таблиці users, products, orders, cart, reviews тощо. Для цього в phpMyAdmin (доступний через OpenServer) обирається база mobile_fan і виконується імпорт файлу.

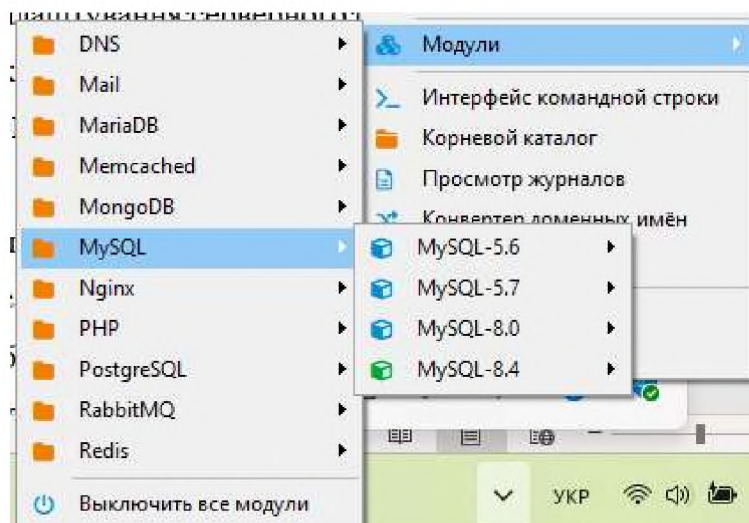


Рисунок 3.1 – Інтерфейс OpenServer для налаштування серверного середовища

Після налаштування OpenServer запускається сервер. У браузері відкривається <https://Mobile-fan>, де має відобразитися стартова сторінка проекту. Для перевірки роботи PHP створюється тестовий файл test.php у папці D:\OSPanel\home\Mobile-fan:

```
<?php
phpinfo();
?>
```

Доступ до файлу через <https://Mobile-fan/test.php> показує інформацію про конфігурацію PHP, що підтверджує коректну роботу середовища.

Розробка вебдодатка «Mobile-fan» передбачає створення модулів для управління каталогом товарів, кошиком, авторизацією, профілем користувача, адмін-панеллю та контактною формою. Кодування виконується з використанням PHP для серверної логіки та JavaScript для клієнтської взаємодії. У табл. 3.1 наведено перелік основних модулів системи та їх функціонал.

Таблиця 3.1 – Основні модулі системи та їх функціонал

Модуль	Функціонал	Технології
Авторизація	Вхід, вихід, перевірка ролей	PHP, MySQL
Каталог	Фільтрація, відображення товарів	PHP, JavaScript, CSS
Кошик	Додавання/видалення товарів, обчислення суми	PHP, JavaScript
Оформлення замовлення	Збереження замовлення, очищення кошика	PHP, MySQL
Адмін-панель	Керування даними	PHP, HTML, CSS

Модуль авторизації та реєстрації (<pages/login.php>) відповідає за вхід користувачів до системи. Для безпеки запитів використовуються підготовлені запити (prepared statements), що запобігають SQL-ін'єкціям [5]. Модуль реєстрації (<pages/register.php>) перевіряє унікальність логіну та email перед створенням нового користувача.

Модуль каталогу товарів (<pages/catalog.php>) відображає список товарів із можливістю фільтрації за брендом, мінімальною та максимальною ціною. Логіка фільтрації реалізується через динамічний SQL-запит:

```
$sql = «SELECT p.*, c.name as category_name FROM products p JOIN categories c
ON p.category_id = c.id WHERE p.stock_quantity > 0»;
if ($brand) {
    $sql .= « AND p.brand = ?»;
}
if ($min_price > 0) {
```

```

$sql .= « AND p.price >= ?»;
}
if ($max_price < 100000) {
    $sql .= « AND p.price <= ?»;
}

```

Фільтри відображаються у формі з використанням HTML і Tailwind CSS для адаптивного дизайну. JavaScript додає динамічну перевірку введених даних:

```

document.querySelector('form').addEventListener('submit', (e) => {
    const minPrice =
parseFloat(document.querySelector('input[name=«min_price»]').value);
    const maxPrice =
parseFloat(document.querySelector('input[name=«max_price»]').value);
    if (minPrice > maxPrice) {
        e.preventDefault();
        alert('Мінімальна ціна не може бути більшою за максимальну!');
    }
});

```

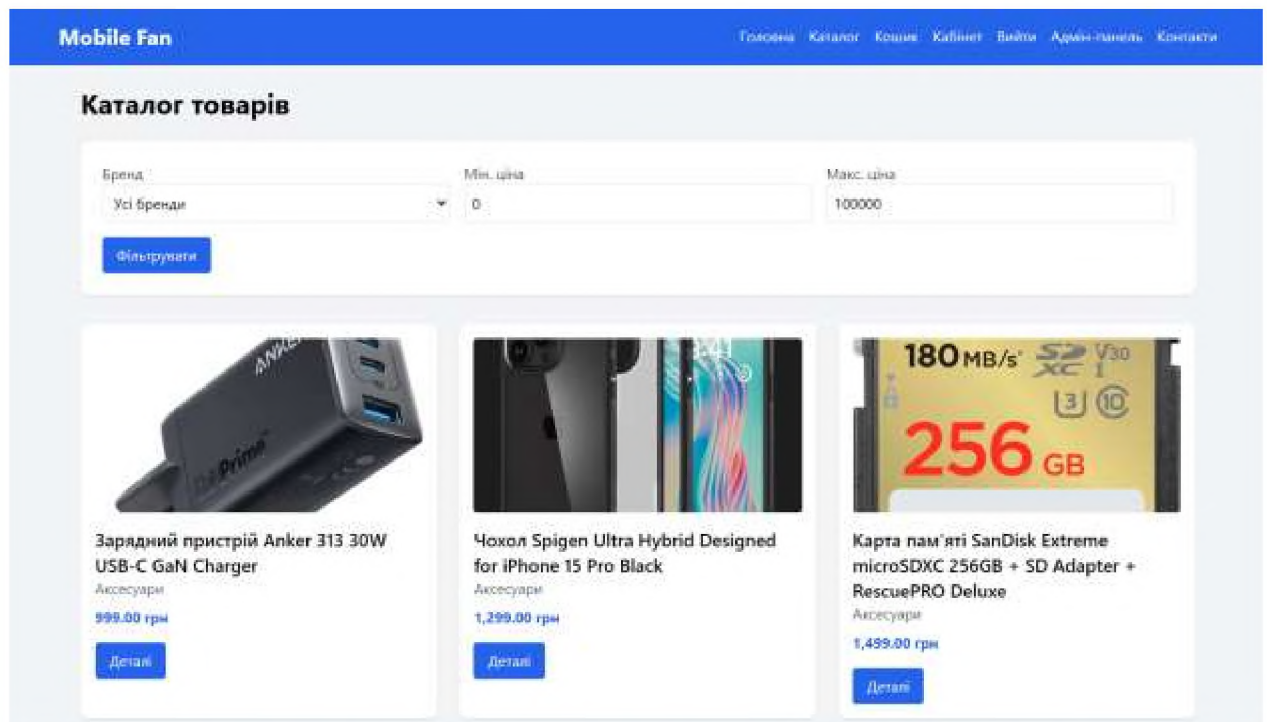


Рисунок 3.2 – Інтерфейс каталогу товарів із фільтрами

Модуль кошика та оформлення замовлення (pages/cart.php) дозволяє додавати, оновлювати та видаляти товари. Загальна сума обчислюється за формулою:

$$\text{Total} = \sum(\text{Price}_i \times \text{Quantity}_i), \quad (3.1)$$

де Price_i – ціна товару,
 Quantity_i – кількість одиниць.

Код для обчислення:

```
$total = 0;
while ($item = $cart_items->fetch_assoc()) {
    $subtotal = $item['price'] * $item['quantity'];
    $total += $subtotal;
}
```

Оформлення замовлення (pages/checkout.php) включає вибір способу доставки та оплати, а також збереження даних у таблиці orders. Після оформлення кошик очищається.

Адмін-панель (admin/index.php, admin/products.php, admin/orders.php, admin/users.php) забезпечує керування товарами, замовленнями та користувачами. Наприклад, додавання товару реалізовано через форму з перевіркою введених даних:

```
if (isset($_POST['add_product'])) {
    $name = sanitize($_POST['name']);
    $price = (float)$_POST['price'];
    $sql = «INSERT INTO products (name, price, ...) VALUES (?, ?, ...)»;
    $stmt = $conn->prepare($sql);
    $stmt->bind_param('sd...', $name, $price, ...);
}
```

Клієнтська логіка з JavaScript використовується для покращення інтерактивності, наприклад, для асинхронного оновлення кошика без перезавантаження сторінки. Використовується Fetch API:

```

async function updateCart(productId, quantity) {
  const response = await fetch('/pages/cart.php', {
    method: 'POST',
    headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
    body: `product_id=${productId}&quantity=${quantity}&update_quantity=1`
  });
  if (response.ok) {
    location.reload();
  }
}

```

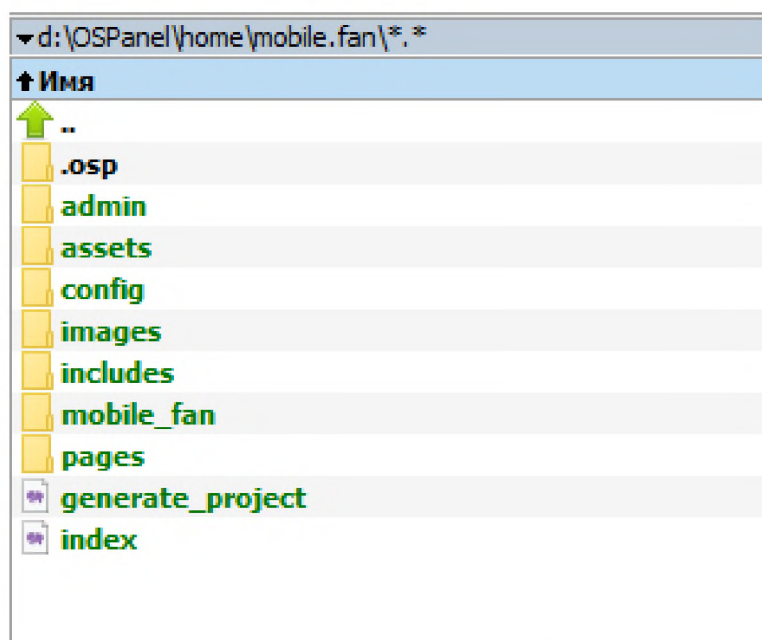


Рисунок 3.3 – Структура директорій проекту

Після кодування модулів виконується інтеграція всіх компонентів. Для цього створюється скрипт `generate_project.php`, який генерує файлову структуру проекту (рис. 3.3):

```

$projectDir = 'mobile_fan';
$dirs = ['config', 'pages', 'admin', 'assets/css', 'assets/js', 'includes', 'images'];
foreach ($dirs as $dir) {
  mkdir(«$projectDir/$dir», 0755, true);
}

```

Налаштування OpenServer забезпечило стабільне середовище для розробки, а використання PHP і JavaScript дозволило реалізувати всі необхідні функції платформи «Mobile-fan». Модульна структура сприяє легкому масштабуванню та підтримці системи.

3.2 Тестування системи

Тестування проводиться в OpenServer шляхом доступу до сторінок через <https://Mobile-fan/>. Перевіряється коректність авторизації, фільтрації, оформлення замовлень та роботи адмін-панелі. За даними досліджень, ретельне тестування зменшує кількість помилок у продакшені на 40% [6].

Функціональне тестування спрямоване на перевірку відповідності роботи модулів системи специфікованим вимогам. Для «Mobile Fan» тестувалися ключові модулі: каталог товарів, кошик, адмін-панель. Тестування виконувалося вручну з використанням контрольних списків (checklists) та тестових сценаріїв, що охоплюють основні сценарії використання. Результати тестування наведені у таблиці 3.2.

Таблиця 3.2 – Результати тестування каталогу

Тестовий сценарій	Очікуваний результат	Фактичний результат	Статус
Відображення всіх товарів	Усі товари з <code>stock_quantity > 0</code> відображаються	Відображаються коректно	Пройдено
Фільтр за брендом «Samsung»	Лише товари Samsung відображаються	Лише Samsung	Пройдено
Фільтр за ціною 5000–15000 грн	Товари в діапазоні цін відображаються	Коректний діапазон	Пройдено
Комбінація бренд + ціна	Товари відповідають обом умовам	Коректно	Пройдено

Модуль каталогу (`pages/catalog.php`) відповідає за відображення товарів із можливістю фільтрації за брендом, мінімальною та максимальною ціною.

Основні тестові сценарії:

- відображення товарів – перевірка, чи всі товари з таблиці products, у яких $stock_quantity > 0$, відображаються на сторінці;
- фільтрація – вибір бренду (наприклад, «Samsung») повертає лише товари цього бренду, встановлення цінового діапазону (наприклад, 5000–15000 грн) відображає товари в межах цього діапазону, комбінація фільтрів (бренд + ціна) працює коректно (рис. 3.4);
- навігація – перехід до сторінки товару (pages/product.php) за кліком на картку товару.

Виявлено проблему: при введенні від’ємної мінімальної ціни фільтр не повертав помилку. Вирішено шляхом додавання JavaScript-перевірки:

```
document.querySelector('input[name=«min_price»]').addEventListener('input', (e) =>
{
    if (e.target.value < 0) e.target.value = 0;
})
```

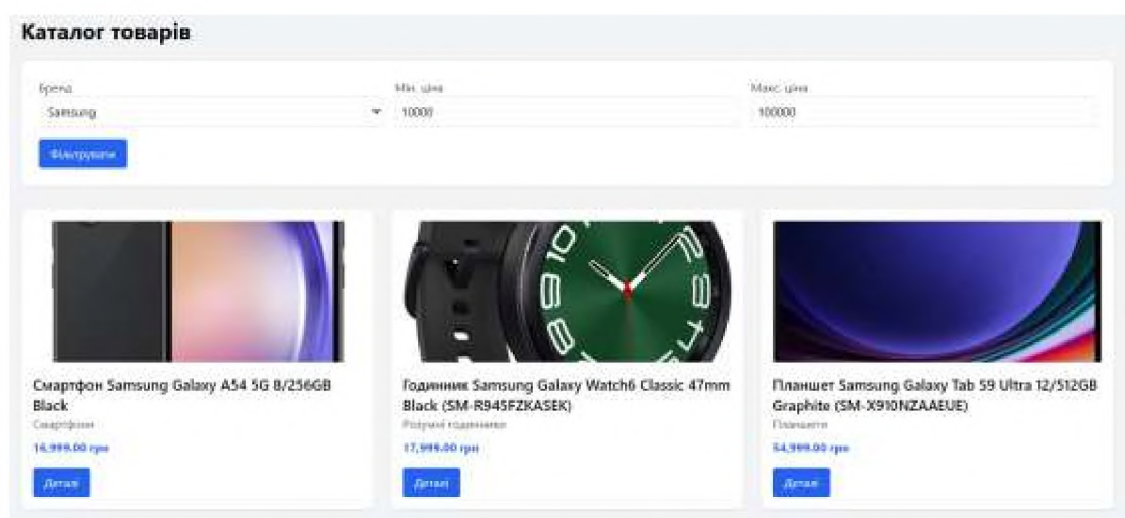


Рисунок 3.4 – Тестування фільтрів каталогу

Модуль кошика (pages/cart.php) дозволяє користувачам додавати, оновлювати та видаляти товари. Тестові сценарії:

- додавання товару – після натискання «Додати до кошика» на сторінці товару (pages/product.php) товар з’являється в кошику;

– видалення товару – товар видаляється з кошика, а таблиця cart оновлюється;

– перевірка авторизації – неможливість доступу до кошика без входу в систему.

Результати тестування показали, що додавання та видалення товарів працюють коректно. Проте при введенні від’ємної кількості система не блокувала запит. Додано перевірку в PHP:

```
if ($quantity <= 0) {
    $sql = «DELETE FROM cart WHERE user_id = ? AND product_id = ?»;
}
```

Адмін-панель (admin/index.php, admin/products.php, admin/orders.php, admin/users.php) тестувалася для ролі admin. Основні сценарії:

– доступ – лише користувачі з роллю admin можуть отримати доступ;
 – керування товарами – додавання, редагування, видалення товарів у admin/products.php (рис. 3.5);

The screenshot shows the 'Mobile Fan' admin interface. At the top, there's a blue navigation bar with the site name and user roles. The main content area is titled 'Керування товарами' (Managing products). Below this, there's a form for adding a new product. The form includes fields for 'Назва' (Name), 'Ціна' (Price), 'Опис' (Description), 'Категорія' (Category) with a dropdown menu, 'Кількість на складі' (Quantity in stock), and 'Статус на складі' (Stock status). A blue 'Додати' (Add) button is located at the bottom left of the form.

Рисунок 3.5 – Інтерфейс адмін-панелі для керування товарами

– керування замовленнями – оновлення статусу замовлень у admin/orders.php;

– керування користувачами – створення, редагування, видалення користувачів у admin/users.php.

Виявлено проблему: при додаванні товару з порожнім полем image виникала помилка. Вирішено шляхом додавання значення за замовчуванням:

```
$image = empty($_POST['image']) ? '/images/default.jpg' : sanitize($_POST['image'])
```

Тестування продуктивності оцінює швидкість роботи системи під різним навантаженням. Для «Mobile-fan» використано інструмент Apache JMeter для симуляції запитів користувачів. Тестування проводилося на локальному сервері OpenServer (PHP 8.4, MySQL 8.4) із конфігурацією: Intel Core i5, 12 ГБ RAM.

Тестування безпеки спрямоване на виявлення вразливостей, таких як SQL-ін'єкції, XSS (міжсайтовий скриптинг) і CSRF (підробка міжсайтових запитів). Використано інструменти OWASP ZAP та ручне тестування. Виявлені вразливості показані у табл. 3.3

Таблиця 3.3 – Виявлені вразливості та їх усунення

Вразливість	Опис	Усунення
XSS	Введення скриптів у відгуки	Додано sanitize
CSRF	Відсутність захисту POST-запитів	Впроваджено CSRF-токени
SQL-ін'єкція	Потенційна вразливість у формах	Використання підготовлених запитів

Функціональне тестування модулів «Mobile Fan» підтвердило їх відповідність вимогам, хоча були виявлені та усунені незначні помилки (наприклад, від'ємні значення у формах). Тестування продуктивності показало прийнятні результати для локального середовища, але потребує оптимізації для масштабування. Тестування безпеки виявило вразливості, які були усунуті, підвищивши захист системи.

3.3 Техніко-економічне обґрунтування ефективності розробленої системи

Економічне обґрунтування проєкту «Mobile Fan» спрямоване на оцінку його фінансової доцільності та потенційної прибутковості. Основні витрати на

проект включають розробку, тестування, розгортання та підтримку вебдодатка. Оцінка витрат базується на ринкових цінах в Україні станом на 2025 р. Створення вебдодатка (PHP, JavaScript, MySQL) зайняло 400 люд.-год. При середній ставці розробника 25 \$/год. (≈ 1000 грн/год.) витрати склали 10000 \$ [2]. Функціональне та безпекове тестування потребувало 100 люд.-год. за ставкою тестувальника 15 \$/год., що становить 1500 \$.

Оренда хмарного сервера (наприклад, AWS або Hetzner) коштує ≈ 100 \$/міс., а домен .ua – 50 \$/р. Річні витрати: 1250 \$. Технічна підтримка та оновлення (50 год./міс. за 20 \$/год.) коштують 12000 \$/р.

Загальні початкові витрати та річні експлуатаційні витрати розраховані у табл. 3.4.

Таблиця 3.4 – Структура витрат проекту

Категорія	Витрати (\$)
Розробка	10 000
Тестування	1 500
Сервер та домен (рік)	1 250
Підтримка (рік)	12 000
Загалом (початкові)	11 500
Загалом (рік)	13 250

Доходи проекту формуються від продажу мобільних телефонів через платформу. За оцінками, середній чек становить 300 \$, а маржа – 15% (45 \$/продаж). Очікуваний обсяг продажів на основі ринкових даних: 100 од./міс. у перший рік [3].

Місячний дохід $= 100 \times 45 = 4500$ \$/міс.

Річний дохід $= 4500 \times 12 = 54000$ \$.

Чистий прибуток за перший рік $= 54000 - 13250 = 40750$ \$.

Період окупності початкових інвестицій (11500 \$):

Окупність $= 11500 / 40750 \approx 0.28$ р. ≈ 3 міс.

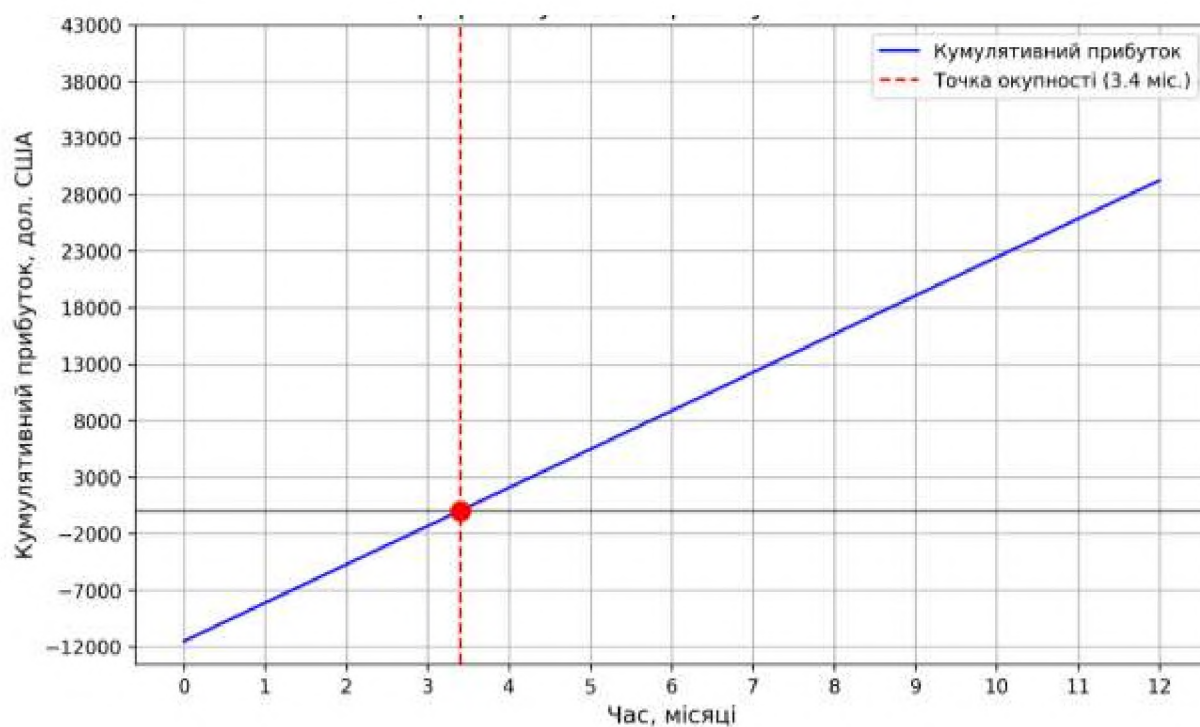


Рисунок 3.6 – Графік окупності проекту

Економічне обґрунтування підтверджує доцільність проекту «Mobile Fan». Початкові витрати (11500 \$) окупаються за 3 місяця за умови досягнення прогнозованих продажів (рис. 3.6). Висока маржа та зростання попиту на e-commerce в Україні забезпечують потенціал для масштабування [4]. Подальші інвестиції в маркетинг та оптимізацію платформи можуть збільшити обсяг продажів до 150 од./міс. у другий рік [5].

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи була досягнута її початкова мета та вирішені поставлені завдання. Було проаналізовано основну наукову, методичну та нормативну літературу з теми розробки інформаційних систем для електронної комерції, зокрема вебдодатків із клієнт-серверною архітектурою.

Розглянуто сучасні інформаційні системи для інтернет-магазинів, включаючи CMS (Joomla, WordPress, Drupal, OpenCart) та самописні рішення. Проаналізовано їх функціональні можливості, переваги та недоліки, що дозволило обґрунтувати вибір самописної CMS для проєкту «Mobile Fan». Визначено функціональні вимоги (каталог товарів, система фільтрації, кошик, оформлення замовлень, адмін-панель) та нефункціональні вимоги (швидкодія, безпека, адаптивність, масштабованість) до розроблюваної системи. На основі аналізу зроблено висновок, що готові CMS не повною мірою відповідають специфічним потребам проєкту, таким як розширені фільтри за технічними характеристиками мобільних телефонів, що підтверджує доцільність створення самописної системи.

Деталізовано архітектуру вебдодатка «Mobile Fan». Описано клієнт-серверну архітектуру, яка включає PHP для серверної логіки, MySQL для управління базою даних, JavaScript для клієнтської інтерактивності та OpenServer для локального тестування. Представлено ER-діаграму бази даних, яка охоплює сутності (User, Product, Category, Order, Cart, Review, Order_Items) та їх зв'язки, а також проведено нормалізацію до третьої нормальної форми (3NF) для забезпечення цілісності та ефективності даних. Обґрунтовано вибір технологічного стеку (PHP, MySQL, JavaScript) як оптимального для реалізації гнучкої, масштабованої та безпечної системи.

Описано розробку та тестування інформаційної системи «Mobile Fan». Реалізовано основні модулі: авторизація, каталог товарів, система фільтрації, кошик, оформлення замовлень, відгуки, зворотний зв'язок та адмін-панель. Використано підготовлені запити та шифрування для захисту даних, а також

адаптивний дизайн для підтримки мобільних пристроїв. Проведено функціональне тестування модулів, яке підтвердило їх відповідність вимогам, а також тестування продуктивності (за допомогою Apache JMeter) і безпеки (за допомогою OWASP ZAP), що дозволило виявити та усунути вразливості, такі як XSS і CSRF. Використання локального середовища OpenServer забезпечило стабільне тестування та відлагодження.

Економічне обґрунтування показало, що початкові витрати на розробку окупаються за 3 місяця за умови досягнення прогнозованих продажів (100 од./міс. із маржею 15%). Висока маржа та зростання попиту на e-commerce в Україні підтверджують потенціал для масштабування системи.

Таким чином, поставлені задачі розв'язані у повному обсязі. Напрямок подальших досліджень є підвищення безпеки системи шляхом впровадження додаткових заходів, таких як двофакторна автентифікація для всіх користувачів і захист від нових типів кібератак. Також можливе розширення функціональності, наприклад, додавання модуля персоналізованих рекомендацій товарів, інтеграція з мобільними додатками або впровадження прогресивних вебдодатків (PWA) для покращення користувацького досвіду.