

МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ
ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ УПРАВЛІННЯ, ПСИХОЛОГІЇ
ТА БЕЗПЕКИ

Кафедра інформаційних технологій

РОЗРОБКА ВЕБ-САЙТУ ЗА ДОПОМОГОЮ СИСТЕМИ
КЕРУВАННЯ КОНТЕНТОМ DRUPAL 10

Кваліфікаційна робота
здобувача вищої освіти
4 курсу заочної форми навчання
Івана ТЕРЕМТИ

Науковий керівник:

Сергій КУТАЄВ

Рецензент:

вчене звання, науковий ступінь

(Ім'я ПРИЗВИЩЕ рецензента)

Кваліфікаційна робота допущена до захисту

«___» _____ 2026 р., протокол № _____

Завідувач кафедри інформаційних технологій

_____ Олег ЗАЧЕК

(підпис)

Львів
2026

АНОТАЦІЯ

ТЕРЕМТА І. Розробка веб-сайту за допомогою системи керування контентом Drupal 10. – Рукопис.

Дослідження на здобуття освітнього ступеня «бакалавр» за спеціальністю 126 «Інформаційні системи та технології». – Львівський державний університет внутрішніх справ, МВС України, Львів, 2026.

У роботі розглянуто сучасні підходи до створення інтернет-магазинів на базі систем керування контентом. Проведено аналіз предметної області, конкурентів та популярних систем керування контентом. Розроблено інтернет-магазин з продажу комп'ютерної периферії. Виконано структурно-функціональне моделювання, спроектовано архітектуру системи та базу даних. Реалізовано сайт з використанням модуля Drupal, Views, Layout Builder, Webform та інших сучасних інструментів.

Ключові слова: веб сайт, системи керування контентом, тестування, база даних, адміністрування, інтерфейс.

ABSTRACT

TEREMTA I. Website development using the Drupal 10 content management system. – Manuscript.

Research for obtaining a bachelor's degree in specialty 126 «Information systems and technologies». – Lviv State University of Internal Affairs, MIA of Ukraine, Lviv, 2026.

The paper considers modern approaches to creating online stores based on content management systems. The subject area, competitors, and popular content management systems are analyzed. An online store selling computer peripherals has been developed. Structural and functional modeling has been performed, the system architecture and database have been designed. The site has been implemented using the Drupal module, Views, Layout Builder, Webform, and other modern tools.

Keywords: website, content management systems, testing, database, administration, interface.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СИСТЕМ КУРЕВАННЯ КОНТЕНТОМ.....	9
1.1. Актуальність розробки інтернет-магазинів у сучасних умовах.....	9
1.2. Аналіз конкурентів ринку.....	10
1.3. Роль систем керування контентом у розвитку електронної комерції....	12
1.4. Аналіз існуючих система керування контентом CMS.....	14
1.4.1. Система керування контентом WordPress.....	15
1.4.2. Система керування контентом Joomla.....	19
1.4.3. Система керування контентом Drupal.....	21
РОЗДІЛ 2 ПРОЄКТУВАННЯ ВЕБ-САЙТУ.....	23
2.1. Принципи роботи та архітектура системи.....	23
2.2. Структурно-функціональне моделювання.....	26
2.2.1. Функціональне моделювання в нотації IDEF0.....	26
2.2.2. Моделювання варіантів використання (Use Case).....	28
2.3. Проєктування бази даних веб-додатку.....	30
2.3.1. Інформаційна архітектура.....	31
2.3.2. Модель «сутність-зв'язок» (ER-діаграма).....	32
2.3.3. Основні типи контенту та поля.....	33
2.3.4. Заходи щодо оптимізації бази даних.....	33
2.4. Вибір технологічного стеку та інструментів розробки.....	34
2.4.1. Обґрунтування вибору основної платформи.....	34
2.4.2. Інструменти розробки та розгортання.....	34

2.5. Дизайн інтерфейсу та користувацького досвіду (UX/UI).....	36
РОЗДІЛ 3 Методи розробки веб-сайту.....	41
3.1. Розробка та наповнення веб-сайту.....	49
3.2. Використання веб-сайту.....	54
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API	Application Programming Interface (інтерфейс програмування додатків)
CRUD	Create, Read, Update, Delete (операції для роботи з даними)
DHTML X	Dynamic HTML eXtensions (бібліотека для створення інтерактивних діаграм)
HTTP	Hypertext Transfer Protocol (протокол передачі даних для веб-запитів)
JSON	JavaScript Object Notation (формат обміну даними)
LAMP	Linux, Apache, MySQL/MariaDB, PHP (технологічний стек розгортання серверної частини)
OAuth	Open Authorization (механізм аутентифікації і авторизації)
PHP	Hypertext Preprocessor (мова програмування для веб-розробки)
SOAP	Simple Object Access Protocol (протокол обміну структурованими повідомленнями)
SQL	Structured Query Language (мова для роботи з базами даних)
SSL	Secure Sockets Layer (протокол захищеного з'єднання)
SSO	Single Sign-On (єдина система входу)
UI	User Interface (користувацький інтерфейс)
URL	Uniform Resource Locator (уніфікований покажчик ресурсу)
XML	eXtensible Markup Language (розширювана мова розмітки)

ВСТУП

У сучасних умовах стрімкого розвитку цифрових технологій та електронної комерції створення якісного веб-сайту є одним із ключових чинників успішного ведення бізнесу. Зростання популярності онлайн-торгівлі зумовлює потребу у зручних, надійних і функціональних інтернет-магазинах, які дозволяють розширювати клієнтську базу, оптимізувати процеси продажу та підвищувати конкурентоспроможність підприємства. Значну допомогу в цьому може надати використання сучасних систем керування контентом (CMS), зокрема Drupal 10, яка забезпечує гнучкість, безпеку та масштабованість веб-ресурсів.

Інтернет-магазин комп'ютерної периферії, реалізований у вигляді спеціалізованого веб-сайту, дозволяє ефективно презентувати широкий асортимент товарів, забезпечувати зручний пошук, оформлення замовлень та взаємодію з клієнтами в режимі онлайн. Такий програмний продукт є актуальним інструментом для підприємств, які займаються продажем комп'ютерної техніки, комплектуючих та периферійних пристроїв, сприяючи автоматизації бізнес-процесів і підвищенню рівня обслуговування споживачів.

Необхідність розробки такого веб-сайту обумовлюється активним переходом торгівлі в онлайн-середовище, зростанням вимог споживачів до швидкості та зручності покупок, а також потребою у зниженні витрат на утримання фізичних торгових точок. Подібні рішення можуть використовуватися як невеликими приватними підприємцями, так і середніми компаніями, що працюють у сфері ІТ.

Володіючи сучасним набором функцій (ієрархічний каталог товарів, кошик, оформлення замовлень, система управління контентом, адаптивний дизайн), розроблюваний веб-сайт покликаний забезпечити ефективну

онлайн-торгівлю комп'ютерною периферією та покращити взаємодію між продавцем і покупцем.

Із викладеного випливає **актуальність теми** кваліфікаційної роботи – зростаюча потреба бізнесу в сучасних, зручних і функціональних веб-рішеннях для організації електронної комерції на основі потужних систем керування контентом.

Аналіз останніх досліджень і публікацій. Питанням проєктування та розробки інтернет-магазинів на базі систем керування контентом присвячено значну кількість наукових праць українських та зарубіжних авторів. Серед них можна виділити роботи, що розглядають особливості Drupal [6–9], порівняльний аналіз CMS [1–5], а також дослідження у сфері електронної комерції. Однак швидкий розвиток веб-технологій, поява нових версій CMS, зміна підходів до фронтенд-розробки та підвищення вимог до безпеки й продуктивності вимагають подальшого удосконалення методів створення таких систем. У зв'язку з цим питання розробки інтернет-магазину на базі Drupal 10 залишається актуальним.

Метою кваліфікаційної роботи є розробка функціонального інтернет-магазину з продажу комп'ютерної периферії за допомогою системи керування контентом Drupal 10.

Для досягнення поставленої мети необхідно виконати такі **завдання**:

- проаналізувати предметну область та існуючі аналоги інтернет-магазинів;
- провести структурно-функціональне моделювання системи;
- спроектувати архітектуру веб-сайту та базу даних;
- розробити інтернет-магазин на платформі Drupal 10;
- здійснити тестування та аналіз отриманого програмного продукту.

Об'єктом дослідження є процес розробки веб-сайту за допомогою системи керування контентом.

Предметом дослідження виступають методи та засоби створення веб-сайтів на базі системи керування контентом Drupal 10.

Методи дослідження. У роботі застосовано загальнонаукові методи: аналіз і синтез, порівняння, системний підхід, моделювання, а також спеціальні методи веб-розробки, включаючи використання Composer, Git та хостинг-платформи Pantheon.

Структура роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел. Обсяг основного тексту роботи становить 50 сторінок, містить 23 рисунки, 2 таблиці.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СИСТЕМ КУРЕВАННЯ КОНТЕНТОМ

1.1. Актуальність розробки інтернет-магазинів у сучасних умовах

Сучасний світ характеризується стрімким переходом бізнесу в цифрове середовище. Електронна комерція стала одним із найдинамічніших секторів економіки. За даними досліджень, у 2025 році обсяг ринку електронної комерції в Україні сягнув приблизно 256 млрд грн, що на 7% більше, ніж у попередньому році. Частка онлайн-покупок у загальному роздрібному товарообороті становить близько 10%, тоді як у світі цей показник перевищує 20%. Глобальний ринок e-commerce у 2026 році, за прогнозами, перевищить 7,4 трлн доларів США.

Особливо актуальним є розвиток онлайн-торгівлі комп'ютерною периферією (монітори, клавіатури, миші, навушники, веб-камери тощо). Зростання віддаленої роботи, геймінгу, навчання онлайн та постійне оновлення техніки формує стабільний попит на цю категорію товарів. Підприємства, які не представлені в інтернеті, втрачають значну частку потенційних клієнтів, оскільки сучасний споживач віддає перевагу швидкому пошуку, порівнянню характеристик і зручному оформленню замовлення без відвідування фізичного магазину.

Основними перевагами інтернет-магазину порівняно з традиційною торгівлею є:

- Зменшення операційних витрат (оренда приміщень, персонал);
- Розширення географії продажів;
- Можливість роботи 24/7;
- Зручність для клієнта (фільтри, порівняння товарів, відгуки);
- Збір даних про поведінку покупців для подальшого маркетингу.

1.2. Аналіз конкурентів ринку

Для оцінки конкурентного середовища та визначення позиціонування розроблюваного інтернет-магазину було проведено аналіз основних компаній ринку продажу комп'ютерної техніки та периферії в Україні станом на 2026 рік. Основні групи конкурентів:

1. Великі універсальні маркетплейси та мережі:

Rozetka.ua – безумовний лідер ринку. Має найбільший асортимент, потужну логістику, високу впізнаваність і довіру клієнтів. Переваги: величезний вибір, швидка доставка, зручний інтерфейс, відгуки, порівняння товарів. Недоліки: через великий обсяг іноді нижча спеціалізація в окремих категоріях і менш професійні консультації щодо складних комплектуючих.

Comfy.ua, MOYO.ua, Foxtrot, Allo – потужні мережі з офлайн-присутністю. Сильні сторони: омніканальність (можливість самовивозу з магазинів), акції, кредитування, гарантійне обслуговування. Вони добре працюють з масовим сегментом.

2. Спеціалізовані магазини комп'ютерної техніки та периферії.

Brain.com.ua – один із найсильніших спеціалізованих магазинів. Добре представлений каталог комплектуючих і периферії, конкурентні ціни, акції для геймерів і збірників ПК. Telemart.ua – спеціалізується на комплектуючих для ПК, ігровій периферії, має шоуруми в кількох містах. Сильна сторона – професійна консультація та орієнтація на ентузіастів. Exe.ua – великий асортимент (понад 50 тис. товарів), орієнтація на опт і роздріб, довга історія на ринку. Artline.ua, CompX, Microtron – спеціалізовані магазини з акцентом на комп'ютерну техніку та периферію. PowerUp.ua, GamingPC.com.ua – орієнтуються на геймерські збірки та високопродуктивну периферію.

Порівняльний аналіз конкурентів

Параметр	Rozetka	Brain	Telemart	MOYO / Allo	Розроблюваний магазин
Спеціалізація на периферії	Середня	Висока	Висока	Середня	Висока
Зручність каталогу та фільтрів	Висока	Висока	Висока	Висока	Висока
Адаптивність та UX	Відмінна	Добра	Добра	Добра	Відмінна
Ціноутворення	Конкурентне	Конкурентне	Конкурентне	Конкурентне	Гнучке
Консультація та контент	Середня	Висока	Висока	Середня	Висока
Швидкість розробки нових функцій	Середня	Середня	Середня	Середня	Висока (Drupal)
Фокус на ніші	Низький	Високий	Високий	Низький	Високий

Більшість великих компаній (Rozetka, MOYO, Allo) є універсальними магазинами, де комп'ютерна периферія – лише одна з багатьох категорій. Це створює можливість для спеціалізованого інтернет-магазину, який може виграти за рахунок:

- Глибшої експертизи в категорії (детальніші описи, порівняльні таблиці, гайди);
- Кращої персоналізації пропозицій;

- Швидшого впровадження нових функцій (наприклад, конфігуратор ПК, розширений фільтр за характеристиками периферії);
- Фокусу на користувацькому досвіді саме для цієї товарної групи.

Розроблюваний інтернет-магазин на базі Drupal 10 має потенціал зайняти нішу професійного спеціалізованого ресурсу, який поєднує зручність великих маркетплейсів з глибокою спеціалізацією, характерною для Brain чи Telemart, але з сучаснішим інтерфейсом і гнучкішою архітектурою.

1.3. Роль систем керування контентом у розвитку електронної комерції

Для ефективної реалізації інтернет-магазину критично важливим є вибір інструментів розробки. Сучасні системи керування контентом (CMS) дозволяють значно прискорити створення та підтримку веб-ресурсу, знизити залежність від дорогих розробників і забезпечити зручне управління.

Серед існуючих CMS особливе місце посідає Drupal 10 – потужна, гнучка та безпечна open-source система, яка ідеально підходить для створення складних комерційних проєктів. Drupal вирізняється високим рівнем безпеки, модульною архітектурою, широкими можливостями масштабування та вбудованою підтримкою SEO. Станом на 2026 рік Drupal 10 залишається актуальною версією, що дозволяє реалізовувати сучасні рішення, водночас вимагаючи подальшого планування переходу на Drupal 11.

Використання системи керування контентом на кшталт Drupal дає можливість:

- Швидко створювати та керувати каталогом товарів;
- Налаштовувати складні бізнес-процеси (кошик, оформлення замовлення, інтеграції з платежами та доставкою);
- Забезпечувати адаптивність під мобільні пристрої;
- Підтримувати високий рівень захисту даних користувачів.

Незважаючи на стрімке зростання e-commerce, багато українських компаній, особливо середнього та малого бізнесу, досі стикаються з проблемами:

- Відсутність сучасного, зручного онлайн-представництва;
- Низька якість існуючих сайтів (повільне завантаження, погана мобільна версія, складна навігація);
- Складнощі в управлінні великим асортиментом товарів;
- Недостатня інтеграція з системами обліку та аналітики.
- Розробка спеціалізованого інтернет-магазину комп'ютерної периферії на базі Drupal 10 дозволяє вирішити ці проблеми, забезпечивши:
- Інтуїтивно зрозумілий інтерфейс для користувачів;
- Ефективне управління контентом для адміністратора;
- Можливість подальшого масштабування проекту;
- Відповідність сучасним стандартам безпеки та продуктивності.

CMS відіграють ключову роль у веб-розробці. Вони дозволяють користувачам без глибоких знань програмування створювати функціональні сайти, що відповідають сучасним стандартам, і водночас надають розробникам інструменти для реалізації складних проєктів. Розуміння ролі CMS у цьому процесі є важливим для оцінки їхнього впливу та визначення напрямків удосконалення, зокрема створення простих систем.

Однією з основних функцій CMS у сучасній веб-розробці є спрощення роботи з контентом. Завдяки інтуїтивним інтерфейсам користувачі можуть додавати тексти, зображення, відео чи посилання, не звертаючись до коду. Це особливо важливо в умовах зростання попиту на швидке оновлення інформації: бізнеси потребують регулярно публікувати новини, блогери – ділитися новими постами, а магазини – оновлювати каталоги товарів. Наприклад, такі системи, як Drupal, дозволяють за лічені хвилини створити сторінку чи пост, що робить їх незамінними для динамічних вебсайтів.

CMS також відіграють важливу роль у підвищенні ефективності розробки. Завдяки модульній структурі та готовим шаблонам вони

скорочують час, необхідний для запуску сайту, порівняно з розробкою з нуля. Однак ця ефективність має свою ціну: популярні CMS часто перевантажені функціями, які не завжди потрібні конкретному користувачу, що може призводити до зниження швидкодії чи ускладнення освоєння системи.

Безпека є ще одним аспектом, де роль CMS у веб-розробці має двоїстий характер. З одного боку, розвинені системи, такі як WordPress, пропонують регулярні оновлення та плагіни для захисту від атак. З іншого боку, їхня популярність робить їх привабливими цілями для

Роль CMS у сучасній веб-розробці можна узагальнити як забезпечення доступності, ефективності та гнучкості у створенні вебсайтів. Вони дозволяють як новачкам, так і професіоналам швидко реалізовувати ідеї, адаптуватися до змін та відповідати вимогам аудиторії.

Таким чином, актуальність теми кваліфікаційної роботи зумовлена стрімким розвитком ринку електронної комерції в Україні та світі, зростанням попиту на комп'ютерну периферію та необхідністю створення сучасних, зручних і конкурентоспроможних веб-рішень на базі потужних систем керування контентом.

1.4. Аналіз існуючих система керування контентом CMS

Система керування контентом (Content Management System, CMS) – це спеціалізоване програмне забезпечення, призначене для створення, редагування, публікації та управління веб-контентом без необхідності глибоких знань програмування. CMS відіграє ключову роль у сучасному цифровому середовищі, оскільки веб-сайти стали основним інструментом комунікації, бізнесу, поширення інформації та самовираження. Такі системи використовуються для створення сайтів різного типу і складності – від особистих блогів і корпоративних сторінок до великих інформаційних порталів та інтернет-магазинів.

Головною метою CMS є суттєве спрощення роботи з контентом. Користувач отримує інтуїтивно зрозумілий інтерфейс, через який може додавати та редагувати тексти, зображення, відео, налаштовувати мета-дані для SEO, керувати структурою сайту без написання коду.

Сучасні системи керування контентом можна умовно поділити на дві основні категорії:

Готові рішення (ready-made CMS) – як з відкритим кодом (open-source), так і комерційні (proprietary);

Самостійно розроблені системи (custom-built CMS), створені під конкретні потреби проекту.

До найпопулярніших готових open-source CMS належать WordPress, Drupal, Joomla та інші. Вони пропонують великий набір готового функціоналу, розвинену екосистему модулів і тем, а також активну спільноту розробників і користувачів.

Перевагами готових CMS є швидке розгортання сайту, наявність детальної документації та відносно низький поріг входу. Однак із зростанням складності проекту проявляються й суттєві недоліки. Додавання нестандартного функціоналу часто вимагає встановлення великої кількості плагінів, що негативно впливає на продуктивність і безпеку сайту. Крім того, через високу популярність такі системи стають частою мішенню кібератак. Великий обсяг коду також ускладнює оптимізацію, особливо для невеликих проєктів, де важливі швидкодія та мінімалізм.

1.4.1. Система керування контентом WordPress

WordPress (рис. 1.1) є найбільш поширеною у світі системою керування контентом (Content Management System, CMS). Вона була розроблена на мові програмування PHP та традиційно використовує реляційну базу даних

MySQL.

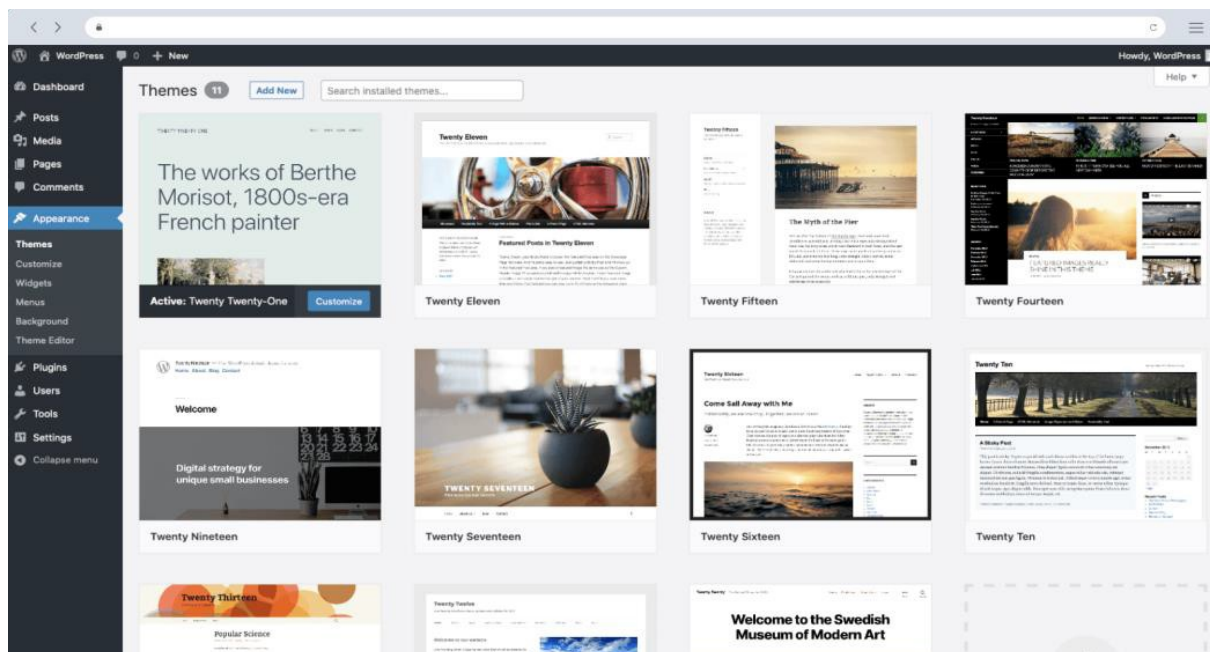


Рис.1.1. Система керування контентом WordPress

Створена у 2003 році як проста платформа для ведення блогів, WordPress з часом еволюціонував у потужний універсальний інструмент, який застосовується для створення різноманітних типів веб-сайтів: від особистих сторінок і портфоліо до складних корпоративних порталів, новинних ресурсів та повноцінних інтернет-магазинів.

Популярність системи пояснюється наявністю величезної екосистеми плагінів і тем, а також активною міжнародною спільнотою, яка налічує мільйони користувачів і розробників [1, 2]. WordPress пропонує інтуїтивно зрозумілий інтерфейс адміністративної панелі, що дозволяє користувачам без спеціальних знань програмування ефективно додавати та редагувати контент, змінювати дизайн сайту та налаштовувати базові функціональні можливості. Водночас система забезпечує широкі перспективи для глибокої кастомізації для розробників, які володіють технічними навичками, включаючи редагування коду тем, створення власних плагінів та інтеграцію зі сторонніми сервісами.

Однією з ключових сильних сторін WordPress є його висока функціональність. Базова версія системи надає можливість створювати сторінки та записи, додавати різноманітні типи контенту (текст, зображення,

відео, аудіо), налаштовувати меню, віджети та таксономії. Завдяки розширенню функціоналу за допомогою спеціалізованих плагінів, таких як WooCommerce (для створення інтернет-магазину), Yoast SEO (для оптимізації під пошукові системи) та Elementor (для візуального конструювання сторінок за принципом drag-and-drop), користувачі отримують можливість трансформувати базовий сайт у потужний комерційний ресурс. Зокрема, WooCommerce додає повноцінну систему управління товарами, кошиком, оформленням замовлень та інтеграціями з платіжними системами, що робить WordPress одним із найпопулярніших рішень серед представників малого та середнього бізнесу.

Проте висока модульність системи має і суттєві недоліки. Встановлення великої кількості плагінів часто призводить до ускладнення структури сайту, виникнення конфліктів між ними та необхідності додаткового часу на налаштування й оптимізацію. У багатьох випадках функціональні можливості WordPress виявляються надлишковими для простих проєктів, що призводить до невиправданого підвищення складності підтримки ресурсу.

Простота використання WordPress значною мірою залежить від рівня технічної підготовки користувача. Для новачків адміністративна панель із сучасним блочним редактором (Gutenberg) є досить інтуїтивною, дозволяючи легко додавати контент, змінювати його порядок і застосовувати стилі. Тисячі безкоштовних і преміум-тем суттєво спрощують початковий етап створення сайту. Однак для користувачів без досвіду веб-розробки процес освоєння додаткових плагінів, вирішення конфліктів сумісності та оптимізації сайту може становити значні труднощі. Наприклад, встановлення та правильне налаштування плагінів кешування потребує розуміння принципів роботи веб-серверів і баз даних.

Питання продуктивності WordPress традиційно є предметом дискусій. У чистому вигляді, без додаткових модулів, система демонструє прийнятну швидкість роботи. Однак активне використання плагінів і зростання обсягу

контенту призводить до збільшення часу завантаження сторінок. Згідно з даними сучасних тестів, середній сайт на WordPress із 5–10 активними плагінами має час завантаження від 1,8 до 3,5 секунд, що в умовах сучасних вимог до Core Web Vitals вважається недостатньо оптимальним. Крім того, традиційне використання MySQL вимагає значних серверних ресурсів, і на бюджетних хостингах продуктивність системи може суттєво знижуватися під час пікових навантажень.

Безпека WordPress характеризується двоїтим характером. З одного боку, регулярні оновлення ядра системи та наявність потужних захисних плагінів (наприклад, Wordfence, Sucuri, Solid Security) дозволяють ефективно протидіяти більшості типових загроз. З іншого боку, висока популярність платформи робить її привабливою мішенню для кібератак. Значна частина вразливостей виникає не в ядрі, а в застарілих або неправильно налаштованих плагінах і темах.

Гнучкість системи є однією з її найважливіших переваг. Завдяки повністю відкритому коду та активній підтримці PHP розробники мають можливість створювати власні теми, плагіни, а за потреби – вносити зміни навіть у ядро системи. Так, редагування файлу `functions.php` дозволяє додавати кастомні функції, віджети та фільтри. Однак така гнучкість вимагає ґрунтовних знань програмування, що суттєво обмежує можливості її використання новачками без залучення спеціалістів.

Вимоги до інфраструктури WordPress є стандартними для PHP-орієнтованих CMS: необхідний хостинг з підтримкою PHP (рекомендовано версію 8.1 і вище) та MySQL/MariaDB. Для локального тестування зазвичай застосовуються програмні стеки на кшталт XAMPP, LocalWP, Laragon або Docker-середовища.

Підсумовуючи, WordPress є потужною, універсальною та високоадаптивною системою керування контентом, яка ідеально підходить для реалізації складних проєктів із великою аудиторією та активним розвитком. Водночас її висока ресурсомісткість, залежність від великої

кількості плагінів, потенційні проблеми з безпекою та продуктивністю роблять її не завжди оптимальним вибором для простих сайтів або проєктів з обмеженими технічними ресурсами. У таких випадках альтернативні рішення, орієнтовані на мінімалізм, локальне зберігання даних та спрощену архітектуру, можуть забезпечити кращий баланс між функціональністю та ефективністю.

1.4.2. Система керування контентом Joomla

Однією з провідних CMS з відкритим кодом є Joomla (рис. 1.2), створена у 2005 році на основі Mambo. Розроблена з використанням PHP та MySQL, вона поєднує в собі простоту WordPress та функціональність Drupal. Joomla орієнтована на створення корпоративних сайтів, порталів, спільнот та вебдодатків, пропонуючи модульну архітектуру та розширені можливості управління контентом. Її популярність пояснюється балансом між гнучкістю для розробників і доступністю для користувачів із базовими технічними навичками, а також активною спільнотою, яка підтримує тисячі розширень і шаблонів [3-5].

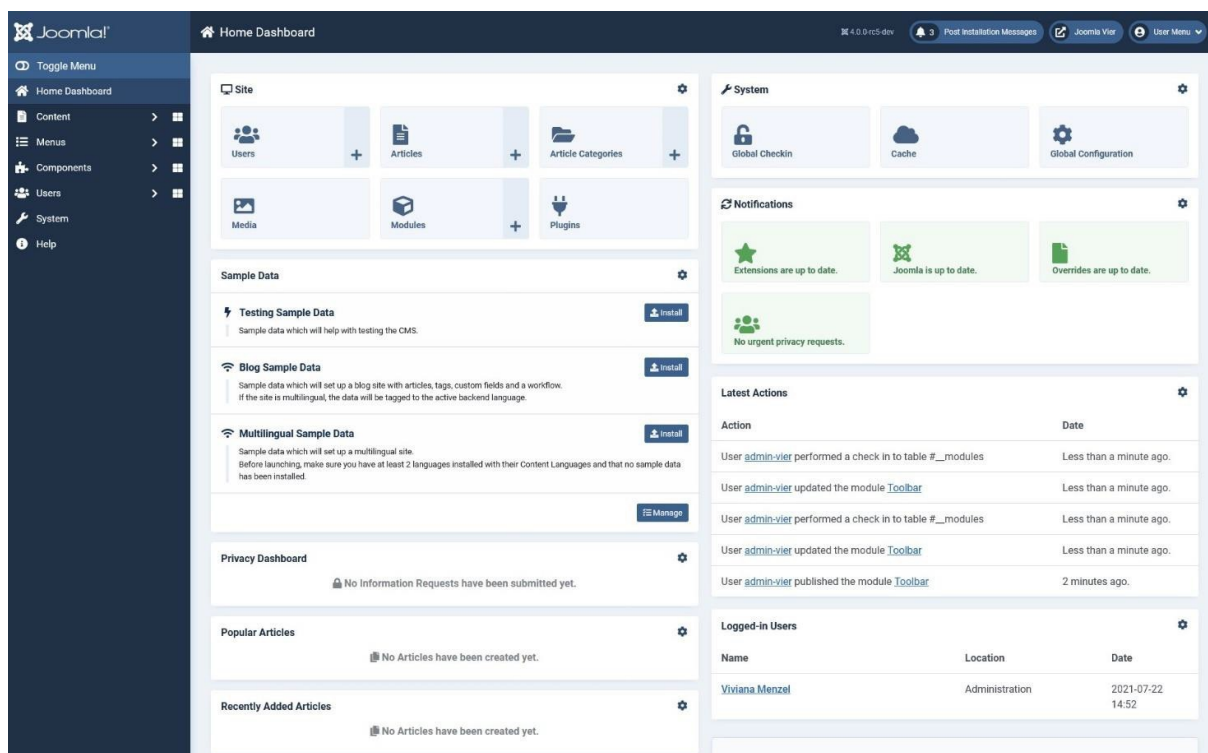


Рис. 1.2. Система керування контентом Joomla

Функціональність Joomla охоплює широкий спектр задач. Базова версія підтримує створення сторінок, статей, меню, категорій, а також управління користувачами з різними рівнями доступу. Через розширення, такі як K2 для контенту чи VirtueMart для e-commerce, система може бути адаптована до складних проєктів, таких як інтернет-магазини чи форуми. Наприклад, VirtueMart додає каталог товарів та систему оплати, що робить Joomla конкурентом WordPress із WooCommerce. Проте Joomla пропонує широкий функціонал, що може бути надлишковим для простих сайтів типу блогів чи лендінгів. Хоч адмін-панель Joomla і має модульний дизайн для швидкого управління контентом та структурою, проте навчитися користуватися нею складніше, ніж у випадку з WordPress. Наприклад, створення меню чи категорій потребує розуміння ієрархії, а встановлення розширень може заплутати новачків через різноманітність опцій. Joomla вимагає більше часу на навчання, особливо для тих, хто не має досвіду в вебадмініструванні, тому мінімалістичний та зрозумілий дизайн має значну перевагу [3-5].

Продуктивність Joomla залежить від конфігурації та хостингу. У базовому вигляді система швидша за WordPress із великою кількістю плагінів, але додавання модулів, наприклад, VirtueMart збільшує навантаження на сервер. Такі великі системи значно поступаються легким flat-file CMS[3-5].

Безпека Joomla підтримується регулярними оновленнями та вбудованими функціями, такі як двофакторна автентифікація. Проте, як і WordPress, вона вразлива через популярність, Joomla також часто атакується через застарілі розширення. Особиста CMS із її простою архітектурою та відсутністю зовнішніх залежностей на початковому етапі, може бути менш схильною до таких ризиків, хоча й не підтримує складні функції безпеки, як Joomla.

Гнучкість Joomla забезпечується її модульною структурою та підтримкою PHP. Розробники можуть створювати власні компоненти, модулі

чи шаблони, редагуючи код через вбудований редактор або файли на сервері. Наприклад, додавання кастомного модуля для відображення новин потребує базового знання PHP та Joomla API [3-5].

Joomla – потужна CMS для середніх та великих сайтів із розвиненим функціоналом, але її складність та залежність від інфраструктури роблять її менш оптимальною для простих задач.

1.4.3. Система керування контентом Drupal

Як CMS із відкритим кодом, Drupal (рис. 1.3) позиціонується як платформа для масштабних та складних веб-проектів з великим навантаженням. Запущена в 2001 році, вона базується на PHP і використовує MySQL або інші БД, як PostgreSQL для зберігання даних. Її аудиторія – це професійні розробники та організації, які потребують гнучкості та безпеки. Drupal підходить для створення корпоративних порталів, урядових сайтів, освітніх платформ та складних вебдодатків, завдяки модульній архітектурі та акценту на кастомізацію [6-8].

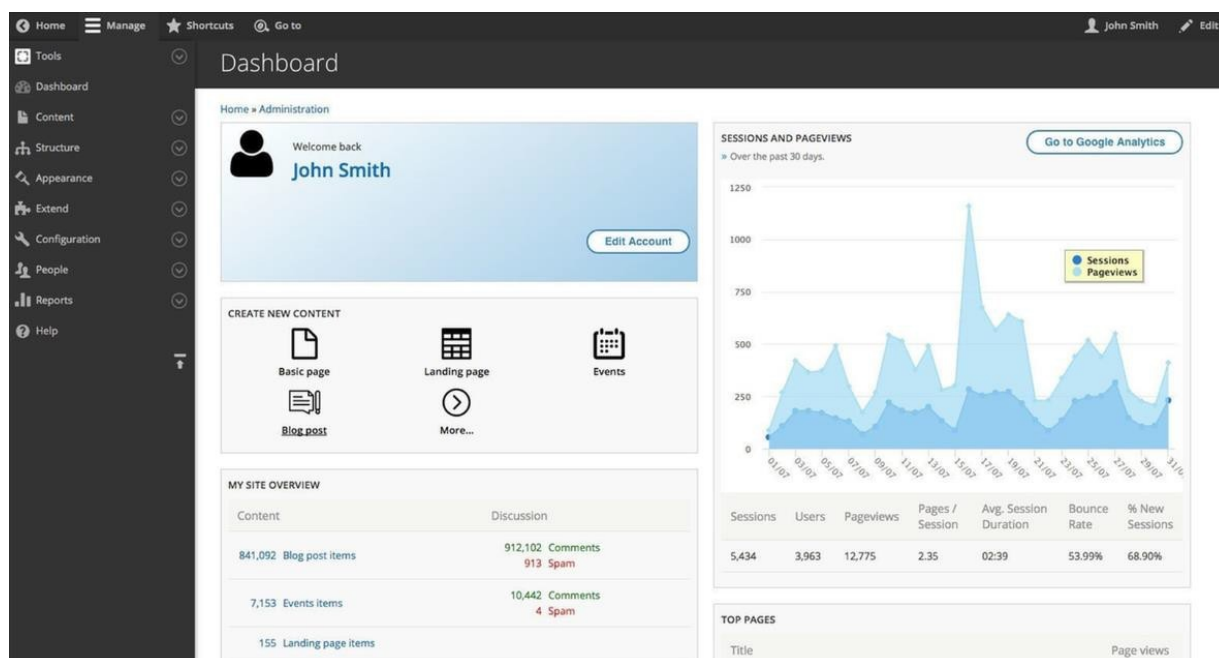


Рис. 1.3. Система керування контентом Drupal

Функціональність Drupal є однією з найширших серед CMS. Базова версія підтримує створення контенту, управління користувачами,

категоризацію та багатомовність. Модулі, такі як Views для створення кастомних сторінок чи Commerce для e-commerce, дозволяють адаптувати систему до специфічних задач, наприклад, платіжних систем чи інтерактивних дашбордів. Drupal не проста в використанні: адмін-панель нових версій стала інтуїтивнішою завдяки редактору Layout Builder, але все одно вимагає технічних знань. Наприклад, створення сторінки через Views потребує розуміння фільтрів та зв'язків, що складно для новачків.

Продуктивність Drupal вища за WordPress та Joomla у базовій конфігурації завдяки оптимізованому ядру. Але додавання модулів таких як Commerce збільшує вимоги до сервера. Drupal ефективна для високонавантажених проєктів, але потребує потужного хостингу.

Безпека – одна з найсильніших сторін Drupal. Її ядро вважається найзахищенішим серед популярних CMS: команда безпеки випускає оновлення щомісяця, а вразливості модулів рідші, ніж у WordPress. Проте безпека залежить від правильного налаштування, що складно для нетехнічних користувачів.

Гнучкість Drupal неперевершена для професійних розробників. Вона дозволяє створювати кастомні типи контенту, інтеграції з API чи навіть власні модулі через PHP та Drupal API. Наприклад, модуль Views дає змогу будувати складні запити без коду, але вимагає базових знань.

Вимоги до інфраструктури Drupal найвищі серед масштабних CMS. Потрібен хостинг із PHP 8.0 або вище, MySQL або PostgreSQL і бажано SSD для швидкості. Це робить її дорожчою в розгортанні в порівнянні з особистою CMS, яка буде налаштована для потреб власного сайту, що спрощує старт для користувачів із мінімальними ресурсами. Drupal ідеальна для складних проєктів із високими вимогами до безпеки та гнучкості, але її складність та ресурсомісткість роблять її непрактичною для простих сайтів.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ВЕБ-САЙТУ

2.1. Принципи роботи та архітектура системи

Суть системи управління контентом полягає у зборі, управлінні та публікації контенту. Система керування розроблена, щоб максимально спростити створення та керування веб-сайтом, зберігаючи при цьому гнучкість у налаштуваннях і контролі. Оскільки створенню систем управління контентом стало приділятися все більше уваги, з'явилися системи з відкритим кодом. Найбільш типовою перевагою систем керування вмістом з відкритим кодом є те, що велика група розробників щодня покращує ці системи, роблячи їх зручнішими у використанні [11-13].

Drupal 10 є однією з найпотужніших і найбільш гнучких open-source систем керування контентом, яка особливо добре підходить для створення складних веб-додатків, зокрема інтернет-магазинів. Основна філософія Drupal базується на принципах модульності, стандартизації даних, API-First та розширюваності, що дозволяє розробникам створювати рішення, які легко адаптуються під змінювані бізнес-вимоги.

Основні принципи роботи Drupal 10:

1. Модульна архітектура Вся функціональність системи реалізована у вигляді незалежних модулів. Ядро Drupal містить лише базовий набір функцій, а необхідний функціонал (каталог товарів, кошик, оформлення замовлень, SEO тощо) підключається через додаткові модулі. Такий підхід забезпечує високу гнучкість і дозволяє уникати зайвого навантаження на систему.

2. Entity-Oriented підхід Усі основні об'єкти системи (товари, користувачі, замовлення, статті, блоки) розглядаються як сутності (Entities). Це дозволяє використовувати єдиний механізм для створення, редагування, зберігання та відображення різних типів контенту.

3. API-First та Decoupled Architecture Drupal 10 активно підтримує headless-підхід завдяки вбудованому модулю JSON:API та GraphQL. Це означає, що система може працювати не тільки як традиційний моноліт, але й як backend, який віддає дані через API для зовнішнього фронтенду (наприклад, Next.js, Nuxt.js).

4. Контентно-орієнтований дизайн Контент повністю відокремлений від його візуального представлення. Це дозволяє змінювати дизайн сайту без втручання в сам контент.

Багатошарова архітектура Drupal 10. Архітектура системи побудована за класичним багатошаровим принципом:

Клієнтський шар (Presentation Layer) Користувач взаємодіє з сайтом через веб-браузер. У роботі було використано класичний підхід з Twig-шаблонами та Layout Builder. Для підвищення продуктивності та сучасного UX можливе використання headless-архітектури.

Application Layer (Drupal Core + Модулі) Центральний шар системи, який обробляє бізнес-логіку. Основні модулі, використані в проєкті:

Drupal Commerce – ключовий модуль для реалізації функціоналу інтернет-магазину (товари, варіації товарів, кошик, замовлення, платежі, знижки);

Views – для створення динамічних каталогів, фільтрів і сторінок товарів;

Layout Builder – сучасний інструмент для візуального компонентного побудування сторінок;

Webform – для створення складних форм (зворотній зв'язок, оформлення замовлення);

Pathauto, Metatag, Redirect – для SEO-оптимізації;

JSON:API – для можливого підключення зовнішнього фронтенду.

Data Access Layer Drupal 10 підтримує реляційні бази даних: MySQL, MariaDB та PostgreSQL. У проєкті використовувалася MySQL. Для

підвищення продуктивності рекомендується підключення Redis (для кешування та сесій) та Varnish або Nginx FastCGI Cache.

Інфраструктурний шар Для розгортання та тестування проєкту використовувалася хмарна платформа Pantheon, побудована на Google Cloud. Вона забезпечує автоматичне масштабування, створення тестових середовищ (Multidev), резервне копіювання та інструменти моніторингу.

Схема роботи системи при обробці запиту користувача

5. Користувач надсилає HTTP-запит через браузер.
6. Веб-сервер (Nginx/Apache) передає запит PHP-інтерпретатору.
7. Drupal виконує bootstrap – завантажує конфігурацію, підключає необхідні модулі.
8. Система виконує маршрутизацію та викликає відповідний контролер.
9. Формується відповідь (HTML-сторінка або JSON для API).
10. Результат кешується на кількох рівнях (Dynamic Page Cache, Render Cache, Redis).
11. Така архітектура забезпечує високу масштабованість, безпеку та продуктивність навіть при значному навантаженні.

Використання Drupal 10 з модульною архітектурою, Entity API та підтримкою сучасних підходів (API-First, Layout Builder, Commerce) дозволило реалізувати інтернет-магазин комп'ютерної периферії з урахуванням вимог до гнучкості, безпеки та зручності адміністрування. Обрана архітектура забезпечує можливість подальшого розвитку проєкту, зокрема переходу до headless-режиму або інтеграції з зовнішніми сервісами.

2.2. Структурно-функціональне моделювання

Структурно-функціональне моделювання є ключовим етапом проєктування, який дозволяє формалізувати бізнес-процеси, визначити функціональні вимоги системи та встановити взаємозв'язки між її компонентами. У рамках роботи для моделювання інтернет-магазину комп'ютерної периферії були застосовані два стандартизовані методи: функціональне моделювання в нотації IDEF0 та діаграми варіантів використання (Use Case) мови UML.

2.2.1. Функціональне моделювання в нотації IDEF0

Методологія IDEF0 використовується для наочного представлення функцій системи та потоків даних. Вона дозволяє описати діяльність у вигляді ієрархічної системи діаграм. На рис. 2.1 зображена контекстна діаграма A-0.



Рис. 2.1. Контекстна діаграма IDEF0 «Функціонування інтернет-магазину комп'ютерної периферії»

Контекстна діаграма представляє систему як єдиний функціональний блок і показує її взаємодію із зовнішнім середовищем. Це найвищий рівень абстракції, який дає загальне уявлення про призначення системи.

У центрі діаграми розташований головний функціональний блок «Інтернет-магазин комп'ютерної периферії».

Вхідні стрілки (Input) зліва демонструють дані, які надходять у систему: запити клієнтів (пошук, перегляд), дані про товари від постачальників, інформацію про користувачів.

Вихідні стрілки (Output) справа відображають результати роботи системи: оформлені замовлення, підтвердження оплати, статус доставки, звіти для адміністратора.

Стрілки управління (Control) зверху показують обмеження та правила, за якими працює система: законодавство України у сфері електронної комерції, Політика конфіденційності, правила обробки персональних даних, бізнес-політика компанії.

Механізми виконання (Mechanism) знизу вказують на ресурси, які забезпечують функціонування: користувачі (клієнти та адміністратори), система керування контентом Drupal 10, база даних, інтеграції зі службами доставки («Нова Пошта», Justin) та платіжними системами.

Дана діаграма виконує роль «чорної скриньки» і є фундаментом для подальшої декомпозиції. Вона допомагає зрозуміти межі системи та її місце в бізнес-процесах підприємства.

На рис. 2.2 представлена декомпозиція контекстної діаграми. Вона містить шість основних блоків:



Рис. 2.2. Діаграма декомпозиції першого рівня

- A1 – Формування та управління каталогом товарів;
- A2 – Пошук, фільтрація та перегляд товарів;
- A3 – Робота з кошиком та оформлення замовлення;
- A4 – Обробка, оплата та виконання замовлень;
- A5 – Адміністрування сайту та управління контентом;
- A6 – Аналіз продажів і формування звітності.)

Декомпозиція дозволяє деталізувати кожен процес та забезпечити повне розуміння функціонування системи. Кожен блок має свої входи, виходи, управління та механізми

2.2.2. Моделювання варіантів використання (Use Case)

Діаграми Use Case призначені для опису взаємодії зовнішніх акторів із системою. Вони допомагають чітко визначити функціональні вимоги з точки зору користувача. На рисунку 2.3 зображена Use Case діаграма.

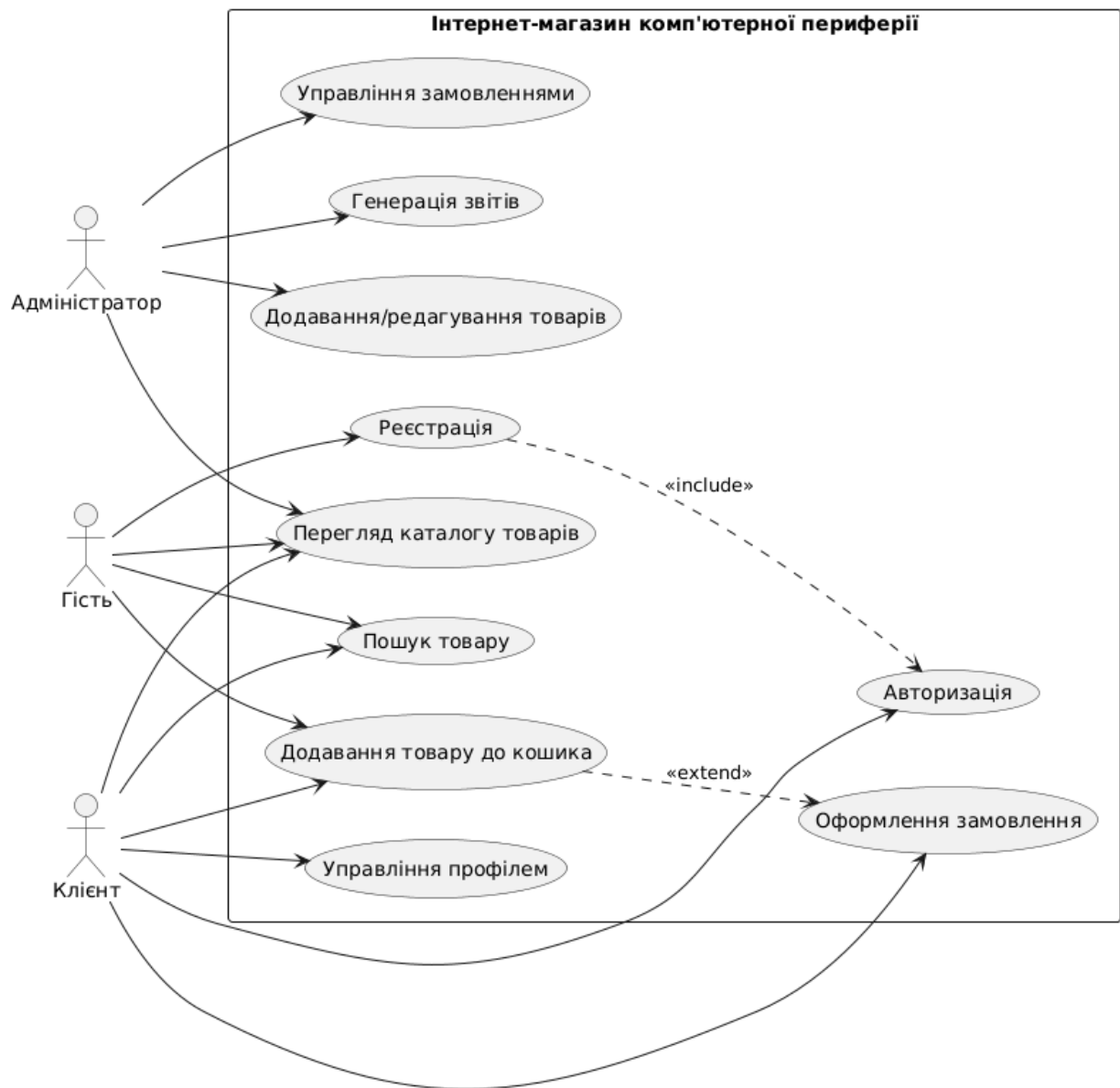


Рис. 2.3. Діаграма варіантів використання (Use Case)

На діаграмі зображено прямокутник системи «Інтернет-магазин комп'ютерної периферії», всередині якого розташовані основні варіанти використання. Зовні прямокутника розміщені актори.

Актори:

- Гість
- Зареєстрований клієнт
- Адміністратор
- Менеджер замовлень

Основні варіанти використання:

- Перегляд каталогу товарів
- Пошук товару за характеристиками
- Додавання товару до кошика
- Оформлення замовлення
- Реєстрація / Авторизація
- Управління особистим кабінетом
- Додавання / редагування товарів (адміністратор)
- Управління замовленнями
- Налаштування сайту
- Генерація звітів)

Особливо детально було промодельовано сценарій «Оформлення замовлення», який включає розгалуження залежно від способу оплати та доставки.

- Проведене структурно-функціональне моделювання дозволило:
- Чітко сформулювати функціональні вимоги до інтернет-магазину;
- Визначити ролі користувачів та їхні взаємодії з системою;
- Виявити ключові бізнес-процеси та їхню послідовність;
- Створити надійну основу для проектування бази даних та безпосередньої програмної реалізації.

Отримані моделі IDEF0 та Use Case стали фундаментом для подальших етапів розробки та забезпечили системний підхід до створення інтернет-магазину на платформі Drupal 10.

2.3. Проектування бази даних веб-додатку

Проектування бази даних є одним із найважливіших етапів розробки веб-додатку, оскільки від його якості залежить продуктивність, масштабованість, цілісність даних та зручність подальшої підтримки системи. У даній роботі для інтернет-магазину комп'ютерної периферії була спроектована база даних з урахуванням особливостей Drupal 10 та модуля **Drupal Commerce**.

2.3.1. Інформаційна архітектура

Інформаційна архітектура системи побудована на **Entity-Based підході**, що є стандартом для Drupal. Усі основні об'єкти системи представлені як сутності (Entities), що забезпечує уніфікований спосіб роботи з даними.

Основні сутності системи:

- User – зареєстровані користувачі (клієнти, адміністратори, менеджери).
- Commerce Product – основна сутність товару.
- Commerce Product Variation – варіації товару (різні кольори, конфігурації).
- Commerce Order – замовлення клієнта.
- Commerce Order Item – позиція в замовленні.
- Customer Profile – профіль клієнта (адреса доставки, контактні дані).
- Taxonomy Term – категорії, бренди, типи пристроїв.
- Media (Image) – зображення товарів.

2.3.2. Модель «сутність-зв'язок» (ER-діаграма)

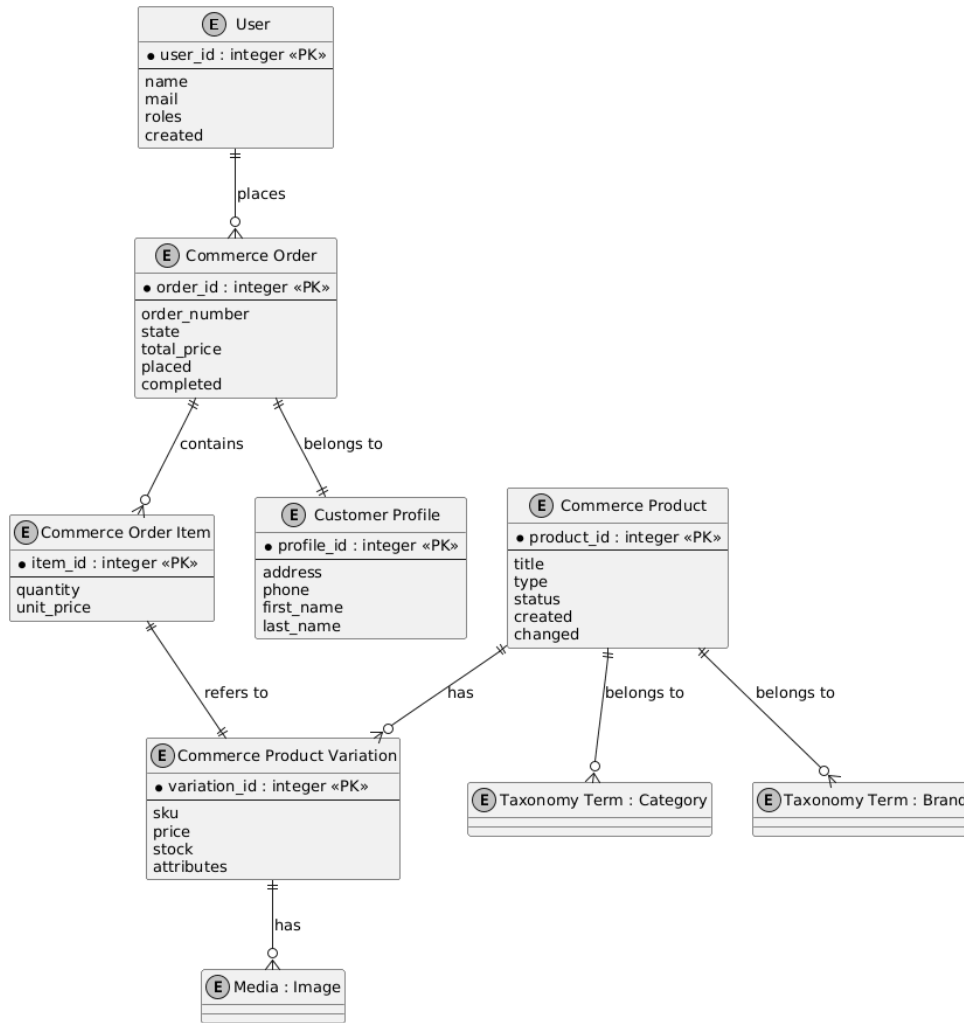


Рис. 2.4. Діаграма «сутність-зв'язок» (Entity-Relationship Diagram) інтернет-магазину

Детальний опис ER-діаграми:

- User пов'язаний з Order (один користувач може мати багато замовлень).
- Commerce Product є основною сутністю товару і має одну або декілька Variation (варіацій).
- Order складається з Order Item, кожен з яких посилається на конкретну варіацію товару.
- Таксоному використовується для категоризації товарів (ієрархічні категорії, бренди).
- Media забезпечує зберігання зображень товарів.

2.3.3. Основні типи контенту та поля

У Drupal 10 були створені такі основні типи контенту:

- Product (Commerce) – основний тип для товарів.
- Page – статичні сторінки (Про нас, Доставка, Контакти).
- Blog – статті блогу.

Ключові поля для типу Commerce Product:

- Назва товару
- Артикул (SKU)
- Ціна
- Короткий та повний опис
- Технічні характеристики (поля типу List, Number, Text)
- Зображення (Media Reference)
- Наявність на складі
- Категорія (Entity Reference)
- Бренд (Entity Reference)

2.3.4. Заходи щодо оптимізації бази даних

- Використання індексів на часто використовуваних полях (sku, price, status).
- Застосування Redis для кешування сутностей та Views.
- Використання PostgreSQL (рекомендовано) або оптимізованого MySQL/MariaDB.
- Регулярне очищення кешу та логів.

Спроектвана база даних на основі Entity API Drupal 10 забезпечує високу гнучкість, цілісність даних та можливість подальшого масштабування інтернет-магазину. Використання стандартних сутностей модуля Commerce дозволило значно прискорити розробку та забезпечити надійність роботи системи.

2.4. Вибір технологічного стеку та інструментів розробки

Вибір технологічного стеку є критично важливим етапом проєктування, оскільки від нього залежить швидкість розробки, масштабованість, безпека, продуктивність та вартість підтримки майбутнього веб-додатку. У даній роботі технологічний стек обирався з урахуванням вимог до функціональності інтернет-магазину, сучасних практик веб-розробки та необхідності забезпечення зручності адміністрування.

2.4.1. Обґрунтування вибору основної платформи

Система керування контентом Drupal 10 була обрана як основа проєкту з таких причин:

- Висока гнучкість і модульна архітектура, що дозволяє реалізовувати складні бізнес-процеси електронної комерції.
- Потужний модуль Drupal Commerce, який є одним із найбільш функціональних рішень для створення інтернет-магазинів серед open-source CMS.
- Високий рівень безпеки – Drupal традиційно вважається однією з найзахищеніших CMS.
- Відмінна підтримка SEO, багатосторінковості та багатомовності.
- Можливість використання API-First підходу (JSON:API), що забезпечує перспективу переходу до headless-архітектури.
- Активна спільнота та регулярні оновлення

Порівняно з WordPress Drupal забезпечує кращу структурованість даних і масштабованість для середнього та великого асортименту товарів.

2.4.2. Інструменти розробки та розгортання

Для реалізації проєкту використовувався наступний технологічний стек:

1. Мова програмування та фреймворк
PHP 8.2+ – основна серверна мова.

Symfony 6.4 (входить до складу Drupal 10) – сучасний компонентний фреймворк, який забезпечує високу якість коду та безпеку.

2. Менеджер залежностей

Composer – стандарт де-факто для управління залежностями у PHP-проєктах. Використовувався для встановлення Drupal, модулів Commerce, Views, Layout Builder та інших компонентів. Composer забезпечує відтворюваність середовища та зручне оновлення пакетів.

3. Система контролю версій

Git – для ведення історії змін коду. Використовувалася модель Git Flow з гілками main, develop та feature-гілками. Репозиторій розміщено на платформі Pantheon.

4. Хостинг та середовище розгортання

Pantheon – спеціалізована хмарна платформа для Drupal та WordPress.

Переваги вибору:

Автоматичне створення тестових середовищ (Multidev);

- Вбудована підтримка Composer, Git, Drush;
- Висока продуктивність на базі Google Cloud;
- Інструменти моніторингу, бекапів та оновлень;
- Можливість безкоштовного використання для розробки.
- 5. Локальне середовище розробки
- Локальний сервер на базі DDEV або Lando (рекомендовано для Drupal).
- Для простоти також використовувався XAMPP / Laragon.

5. Додаткові інструменти та модулі Drupal (табл. 2.1.)

Додаткові інструменти та модулі Drupal

Категорія	Інструмент / Модуль	Призначення
Електронна комерція	Drupal Commerce 2.x	Товари, замовлення, кошик, платежі
Відображення даних	Views + Contextual Filters	Каталог, фільтри, сторінки товарів
Дизайн сторінок	Layout Builder	Компонентне компонування сторінок
Форми	Webform	Зворотній зв'язок, оформлення замовлень
SEO	Metatag, Pathauto, Redirect	Оптимізація для пошукових систем
Продуктивність	Redis, Internal Page Cache	Кешування
Зображення	Image API, Focal Point	Оптимізація медіа
API	JSON:API	Перспектива headless

Загальна архітектура технологічного стеку.

Проект реалізовано за монолітною архітектурою з елементами headless-підходу. Drupal 10 виконує роль backend + frontend, з можливістю відокремлення фронтенду в майбутньому.

Обраний технологічний стек (Drupal 10 + Composer + Git + Pantheon) є оптимальним балансом між швидкістю розробки, функціональністю, безпекою та вартістю підтримки. Він повністю відповідає вимогам сучасного інтернет-магазину комп'ютерної периферії та забезпечує значний потенціал для подальшого розвитку системи.

2.5. Дизайн інтерфейсу та користувацького досвіду (UX/UI)

Дизайн інтерфейсу та користувацького досвіду є одним із ключових факторів успіху інтернет-магазину, оскільки безпосередньо впливає на конверсію відвідувачів у покупців, рівень задоволеності клієнтів та лояльність до бренду. При проектуванні інтерфейсу інтернет-магазину комп'ютерної периферії основна увага приділялася принципам користувацько-орієнтованого дизайну (User-Centered Design), мінімалізму, інтуїтивності та адаптивності.

При розробці дизайну були застосовані такі принципи:

- Мінімалізм і чистота інтерфейсу – уникнення зайвих елементів.
- Адаптивність (Responsive Design) – коректне відображення на всіх пристроях (мобільні телефони, планшети, десктопи).
- Швидкість доступу до ключових дій – пошук, додавання в кошик, оформлення замовлення.
- Візуальна ієрархія – акцент на товарах і цінах.
- Доступність (Accessibility) – відповідність стандартам WCAG 2.1.
- Єдиний стиль бренду – сучасний технологічний вигляд, що асоціюється з IT та геймінгом.

Структура основних сторінок:

На рис. 2.5. подано схема макету головної сторінки (Wireframe)

На головній сторінці розташовані: великий hero-банер з акціями, блок категорій у вигляді карток, секція популярних товарів з кнопками «Купити», блок переваг та відгуків клієнтів.

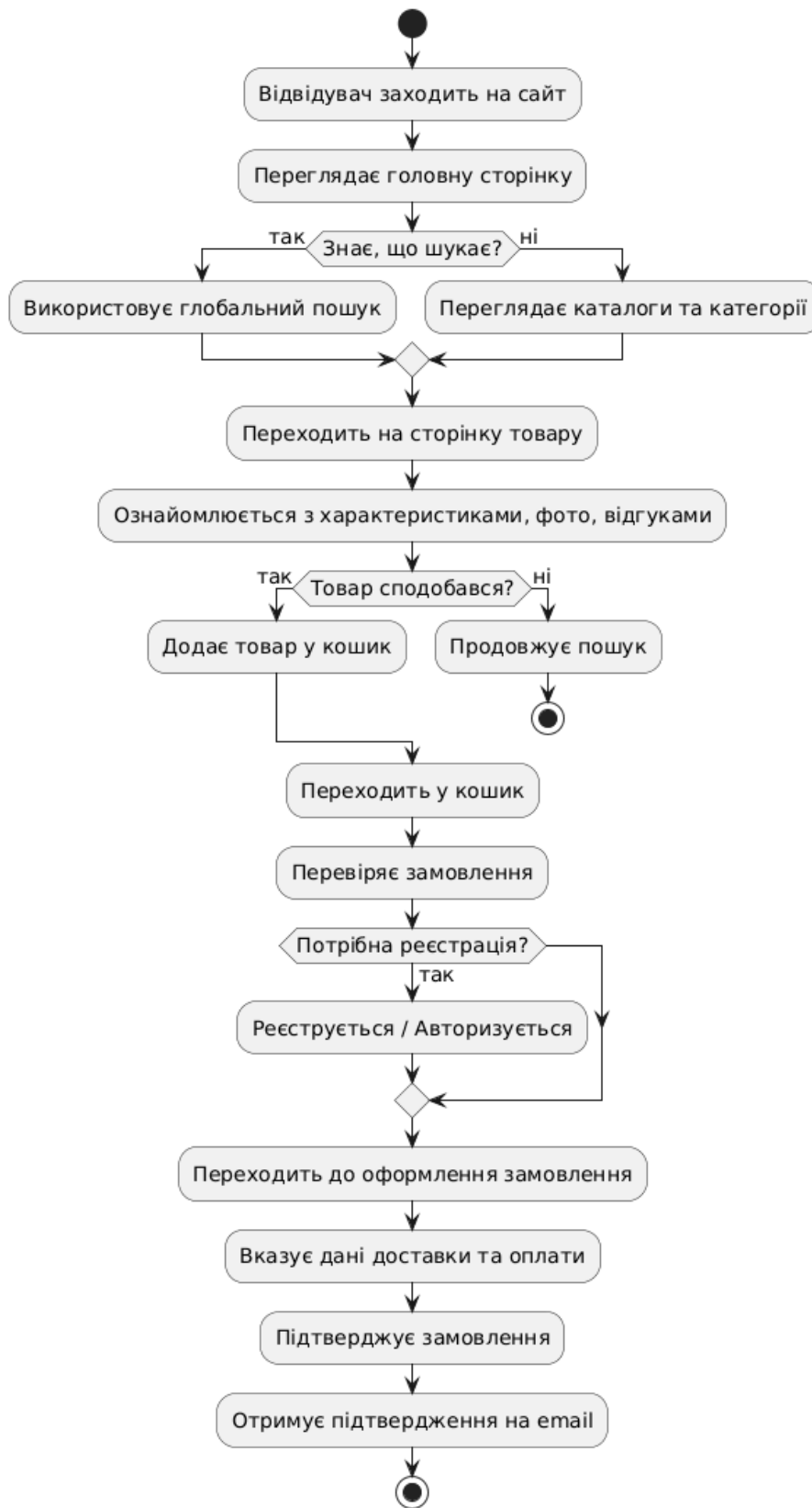


Рис. 2.5. Схема макету головної сторінки (Wireframe)

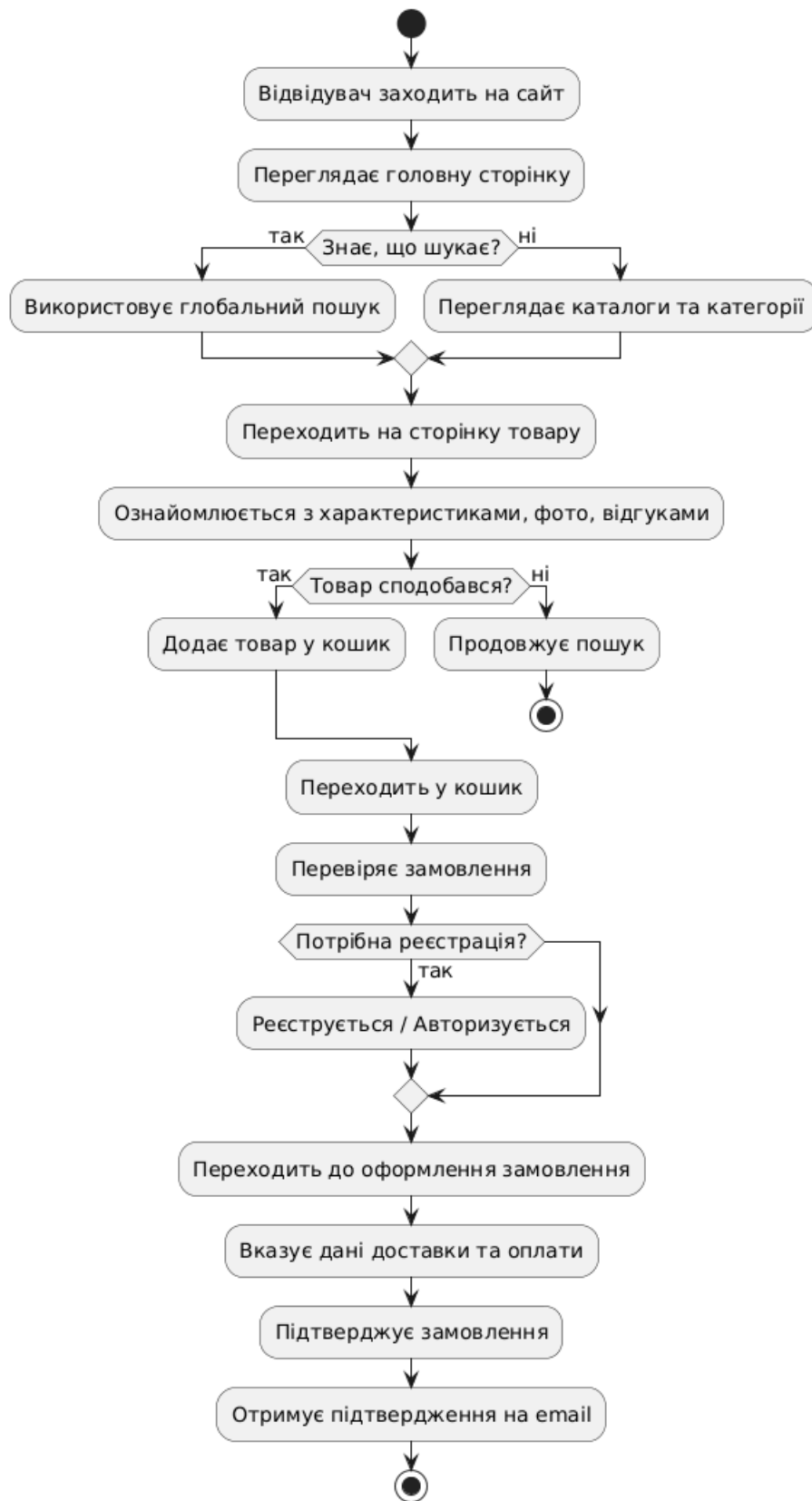


Рис. 2.5. Діаграма User Flow «Купівля товару»

Діаграма ілюструє типовий шлях користувача від заходу на сайт до успішного оформлення замовлення. Особлива увага приділена зменшенню кількості кроків та усуненню бар'єрів на шляху до покупки.

Ключові сторінки та їх особливості:

Сторінка каталогу – потужна система фільтрів (ціна, бренд, характеристики), сортування, сітка/список товарів.

Картка товару – детальні характеристики, галерея зображень, порівняння, рекомендації «Разом з цим купують».

Кошик та оформлення замовлення – покроковий процес, збереження даних, можливість редагування.

Особистий кабінет – історія замовлень, улюблені товари, персональні знижки. **Інструменти дизайну**

Figma – створення wireframes, прототипів та фінального дизайну.

Layout Builder (Drupal) – реалізація компонентного дизайну.

Twig + CSS (Bootstrap 5 + кастомні стилі).

Розроблений інтерфейс поєднує сучасні тенденції UX/UI, високий рівень зручності та функціональності. Адаптивний дизайн, інтуїтивна навігація та продуманий user flow забезпечують позитивний користувацький досвід і сприяють підвищенню конверсії інтернет-магазину комп'ютерної периферії.

РОЗДІЛ 3

МЕТОДИ РОЗРОБКИ ВЕБ-САЙТУ

Етап реалізації є практичною частиною кваліфікаційної роботи, під час якої відбувається безпосереднє створення функціонального інтернет-магазину на основі розробленої раніше архітектури, моделі бази даних та дизайну.

Реалізація проводилася з дотриманням принципів agile-підходу: ітераційна розробка, регулярне тестування та поступове наповнення функціоналом. Основною платформою реалізації стала Drupal 10, що дозволило поєднати швидкість розробки з високою гнучкістю системи.

Для розробки та розгортання інтернет-магазину використовувався два середовища:

Локальне середовище – для основної розробки.

Віддалене середовище – хмарна платформа Pantheon для тестування та демонстрації.

Для розробки та розгортання проєкту було створено такий стек:

- Drupal 10.2 – ядро системи;
- PHP 8.2 – серверна мова програмування;
- MySQL 8.0 – база даних;
- Composer 2 – менеджер залежностей;
- Git – система контролю версій;
- Pantheon – хмарна платформа для розгортання та тестування;
- Drush – командний рядок для Drupal (управління сайтом, оновлення, кеш тощо);
- Figma – для створення та імпорту дизайну;
- VS Code + розширення для Drupal – середовище розробки.

Локальне середовище було налаштоване за допомогою DDEV – сучасного інструменту для розробки Drupal-проектів. Це дозволило швидко підняти повноцінне середовище з підтримкою Redis, Mailhog та XDebug.

Розробка проводилася поетапно:

Встановлення та базова конфігурація Drupal 10 Створення проекту через Composer, налаштування конфігураційного файлу settings.php, підключення до бази даних.

Встановлення та налаштування ключових модулів:

- Drupal Commerce (основний функціонал магазину);
- Views, Layout Builder, Webform;
- Metatag, Pathauto, Redirect;
- Redis (для кешування).

Створення типів контенту та сутностей:

- Commerce Product та Commerce Product Variation;
- Кастомні поля для характеристик периферії (розмір, тип підключення, сумісність тощо);
- Таксономії: «Категорії», «Бренди», «Тип пристрою».

Розробка дизайну та фронтенду:

- Встановлення базової теми;
- Кастомізація через Layout Builder;
- Адаптивна верстка з використанням Bootstrap 5 та кастомного CSS.

Реалізація основного функціоналу:

- Каталог товарів з розширеними фільтрами;
- Кошик та оформлення замовлення;
- Особистий кабінет користувача;
- Система управління замовленнями в адмін-панелі.

Drupal Commerce став основою комерційної частини сайту.

В результаті виконано:

- Встановлено та налаштовано модуль Commerce 2.x.
- Створено тип товару «Периферія».
- Налаштовано Product Variations для підтримки різних характеристик (колір, тип підключення, розмір).
- Реалізовано систему атрибутів товарів.
- Налаштовано типи замовлень та стани замовлень (draft → placed → processing → completed → cancelled).
- Налаштовано Customer Profile для зберігання даних доставки.

Приклад команди встановлення:

Bash

```
composer require drupal/commerce
drush en commerce commerce_product commerce_cart commerce_checkout -y
composer require "drupal/commerce:^2.0"
```

Для розгортання веб-сайту віддалено використовувався сервіс Pantheon (pantheon.io).

Pantheon Systems, Inc. - це приватна корпорація в Сан-Франциско, для веб-сайтів, що працюють на базі Drupal і WordPress CMS з відкритим кодом. Це спеціальний постачальник PaaS для конкретного додатка.

Технологічно програми Pantheon працюють як програмне забезпечення як послуга, а не на власних серверах користувачів. Сервіс Pantheon створений на основі Google Cloud Platform.

Pantheon – це поєднання інструментів веб-розробки в хмарі та послуг веб-хостингу та керування. Pantheon робить для веб-сайту компанії те ж саме, що програмно визначені центри обробки даних роблять для її інфраструктури. Кожен член команди може використовувати всі сервіси, які забезпечують роботу сайту, за запитом, розробляючи на єдиній платформі. Як наслідок, великі веб-сайти запускаються швидше, а керування ними коштує менше.

Базуючись на мультитенантній контейнерній архітектурі Pantheon, запускається за лічені хвилини, не потребуючи додаткової інфраструктури чи віртуальних серверів.

Pantheon полегшує розгортання різних версій сайту на тестових серверах і в реальному часі, а також на резервних серверах. [7]

Для розгортання проекту локально і інсталяції модулів drupal, використовувався менеджер залежностей PHP Composer.

Composer – менеджер пакетів прикладного рівня для мови програмування PHP що забезпечує стандартний формат для управління залежностями у програмному забезпеченні та необхідними бібліотеками.

Composer працює з командного рядка і встановлює залежності (наприклад, бібліотек) для застосунку. Він також дозволяє користувачам встановлювати PHP пакети, доступні на «Packagist», який є його основним сховищем, яке містить доступні пакети. Він також реалізує автозавантажувач класів, для встановлених бібліотек і це полегшує використання коду від сторонніх розробників.

Composer використовується як складова частина декількох популярних PHP проектів з відкритим вихідним кодом, наприклад: Laravel, Symfony.

Приклад команди встановлення:

Bash

```
composer require drupal/commerce  
drush en commerce commerce_product commerce_cart commerce_checkout -y
```

Приклад розгортання Drupal локально за допомогою команди composer (рис 3.1).

A screenshot of a terminal window with a black background. The title bar shows 'MINGW64:/d/ospanel543/domains/cms-kn51m'. The prompt is 'Admin@DESKTOP-E1B6L5R MINGW64 /d/ospanel543/domains/cms-kn51m (master)'. The command entered is '\$ git clone ssh://codeserver.dev.1207fa1d-df26-441a-a129-0604323abccb@codeserver.dev.1207fa1d-df26-441a-a129-0604323abccb.drush.in:2222/~/.repository.git -b master cms-kn51m-kandia'.

```
MINGW64:/d/ospanel543/domains/cms-kn51m  
Admin@DESKTOP-E1B6L5R MINGW64 /d/ospanel543/domains/cms-kn51m (master)  
$ git clone ssh://codeserver.dev.1207fa1d-df26-441a-a129-0604323abccb@codeserver  
.dev.1207fa1d-df26-441a-a129-0604323abccb.drush.in:2222/~/.repository.git -b mast  
er cms-kn51m-kandia
```

Рис 3.1. Приклад інсталяції модуля за допомогою команди composer.

```
MINGW64:/d/ospanel543/domains/cms-kn51m
Admin@DESKTOP-E1B6L5R MINGW64 /d/ospanel543/domains/cms-kn51m (master)
$ composer require 'drupal/panels:^4.7'
./composer.json has been updated
Running composer update drupal/panels
No patches supplied.
> DrupalComposerManaged\ComposerScripts::preUpdate
Using version 'dev-main' for path repositories.
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Package doctrine/reflection is abandoned, you should avoid using it. Use roave/better-reflection instead.
Package symfony/debug is abandoned, you should avoid using it. Use symfony/error-handler instead.
Package webmozart/path-util is abandoned, you should avoid using it. Use symfony/filesystem instead.
Generating autoload files
89 packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
Scaffolding files for pantheon-systems/drupal-integrations:
 - Skip [web-root]/sites/default/settings.php because it already exists and overwrite is false.
> DrupalComposerManaged\ComposerScripts::postUpdate
Found 2 security vulnerability advisories affecting 2 packages.
Run composer audit for a full list of advisories.

Admin@DESKTOP-E1B6L5R MINGW64 /d/ospanel543/domains/cms-kn51m (master)
$ |
```

Рис 3.2. Для роботи з віддаленим сервером використовувалась система контролю версій GIT.

Git – розподілена система керування версіями файлів та спільної роботи. Git є однією з найефективніших, надійних і високопродуктивних систем керування версіями, що надає гнучкі засоби

нелінійної розробки, що базуються на відгалуженні і злитті гілок. Для забезпечення цілісності історії та стійкості до змін заднім числом використовуються криптографічні методи, також можлива прив'язка цифрових підписів розробників до тегів і комітів.

Прикладами проєктів, що використовують Git, є ядро Linux, Android, Debian DragonFly BSD, Perl, Eclipse, GNOME, KDE, Qt, Ruby on Rails, PostgreSQL, VideoLAN, PHP, One Laptop Per Child (OLPC), АБІС Koha, GNU LilyPond та ELinks і деякі дистрибутиви GNU/Linux.

Програма є вільною і випущена під ліцензією GNU GPL версії 2.

Система спроектована як набір програм, спеціально розроблених з врахуванням їхнього використання у скриптах. Це дозволяє зручно створювати спеціалізовані системи керування версіями на базі Git або користувацькі інтерфейси.

Віддалений доступ до репозиторіїв Git забезпечується git-демоном, SSH або HTTP сервером. TCP-сервіс git-daemon входить у дистрибутив Git і є разом з SSH найпоширенішим і надійним методом доступу. Метод доступу HTTP, хоч має низку обмежень, дуже популярний в контрольованих мережах, тому що дозволяє використання наявних конфігурацій мережевих фільтрів. [9]

Приклад заливання даних з локального репозиторію на віддалений.

Для початку оновлююся до актуальної версії гілки master за допомогою команди – `git pull origin master` (рис. 3.3.).

```
MINGW64:/d/ospanel543/domains/cms-kn51m
Admin@DESKTOP-E1B6L5R MINGW64 /d/ospanel543/domains/cms-kn51m (master)
$ git pull origin master
From ssh://codeserver.dev.1207fa1d-df26-441a-a129-0604323abccb.drush.in:2222/~/.r
epository
 * branch          master      -> FETCH_HEAD
Already up to date.

Admin@DESKTOP-E1B6L5R MINGW64 /d/ospanel543/domains/cms-kn51m (master)
$ |
```

Рис 3.3. Оновлюення до актуальної версії гілки master

Тепер додаю файли і заливаю зміни (рис. 3.4).

```
MINGW64:/d/ospanel543/domains/cms-kn51m
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
       modified:   composer.json
       modified:   composer.lock
       modified:   web/sites/default/settings.php
       modified:   web/themes/custom/recipe/css/style.css

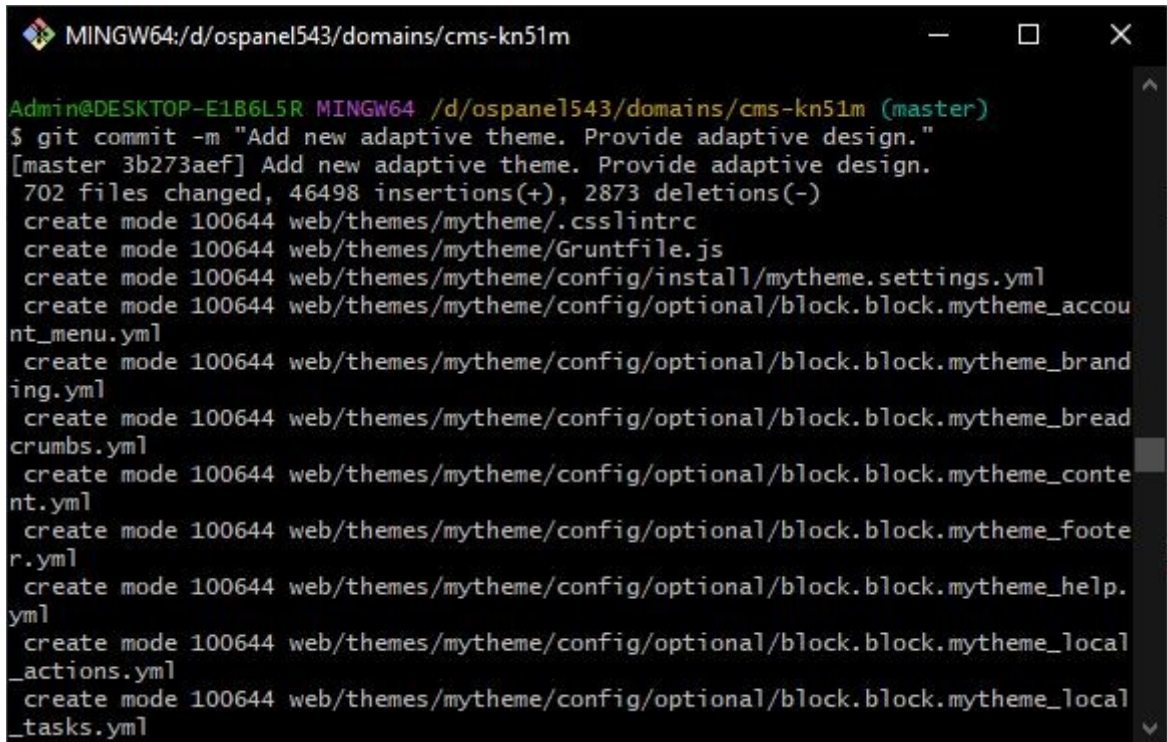
Untracked files:
  (use "git add <file>..." to include in what will be committed)
       web/themes/mytheme/

no changes added to commit (use "git add" and/or "git commit -a")

Admin@DESKTOP-E1B6L5R MINGW64 /d/ospanel543/domains/cms-kn51m (master)
$ git add composer.json composer.lock web/themes/mytheme
warning: in the working copy of 'web/themes/mytheme/.csslintrc', LF will be repl
aced by CRLF the next time Git touches it
warning: in the working copy of 'web/themes/mytheme/layout/page-layout/sass/_pag
e-layout-base.scss', LF will be replaced by CRLF the next time Git touches it
```

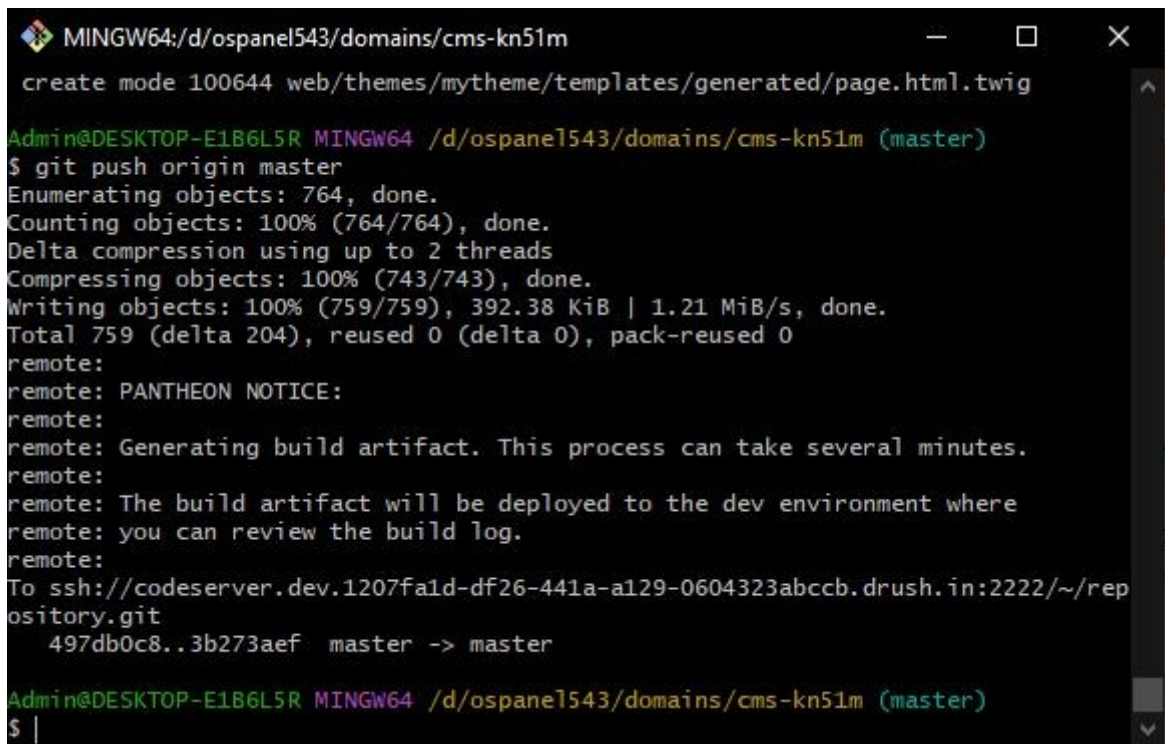
Рис 3.4. Додано файли

Додаю повідомлення commit і заливаю на віддалений репозиторій (рис. 3.5 3.5).



```
MINGW64:/d/ospanel543/domains/cms-kn51m
Admin@DESKTOP-E1B6L5R MINGW64 /d/ospanel543/domains/cms-kn51m (master)
$ git commit -m "Add new adaptive theme. Provide adaptive design."
[master 3b273aef] Add new adaptive theme. Provide adaptive design.
702 files changed, 46498 insertions(+), 2873 deletions(-)
create mode 100644 web/themes/mytheme/.csslintrc
create mode 100644 web/themes/mytheme/Gruntfile.js
create mode 100644 web/themes/mytheme/config/install/mytheme.settings.yml
create mode 100644 web/themes/mytheme/config/optional/block.block.mytheme_account_menu.yml
create mode 100644 web/themes/mytheme/config/optional/block.block.mytheme_branding.yml
create mode 100644 web/themes/mytheme/config/optional/block.block.mytheme_breadcrumbs.yml
create mode 100644 web/themes/mytheme/config/optional/block.block.mytheme_content.yml
create mode 100644 web/themes/mytheme/config/optional/block.block.mytheme_footer.yml
create mode 100644 web/themes/mytheme/config/optional/block.block.mytheme_help.yml
create mode 100644 web/themes/mytheme/config/optional/block.block.mytheme_local_actions.yml
create mode 100644 web/themes/mytheme/config/optional/block.block.mytheme_local_tasks.yml
```

Рис. 3.5. Додано повідомлення commit



```
MINGW64:/d/ospanel543/domains/cms-kn51m
Admin@DESKTOP-E1B6L5R MINGW64 /d/ospanel543/domains/cms-kn51m (master)
$ git push origin master
Enumerating objects: 764, done.
Counting objects: 100% (764/764), done.
Delta compression using up to 2 threads
Compressing objects: 100% (743/743), done.
Writing objects: 100% (759/759), 392.38 KiB | 1.21 MiB/s, done.
Total 759 (delta 204), reused 0 (delta 0), pack-reused 0
remote:
remote: PANTHEON NOTICE:
remote:
remote: Generating build artifact. This process can take several minutes.
remote:
remote: The build artifact will be deployed to the dev environment where
remote: you can review the build log.
remote:
To ssh://codeserver.dev.1207fa1d-df26-441a-a129-0604323abccb.drush.in:2222/~/.repository.git
 497db0c8..3b273aef  master -> master
Admin@DESKTOP-E1B6L5R MINGW64 /d/ospanel543/domains/cms-kn51m (master)
$ |
```

Рис. 3.6. Віддалений репозиторій

Як видно з рис. 3.6 залити зміни на віддалений репозиторій вдалося без введення паролю і логіну. Це тому, що попередньо на pantheon.io був доданий публічний ключ ssh (рис 3.7).

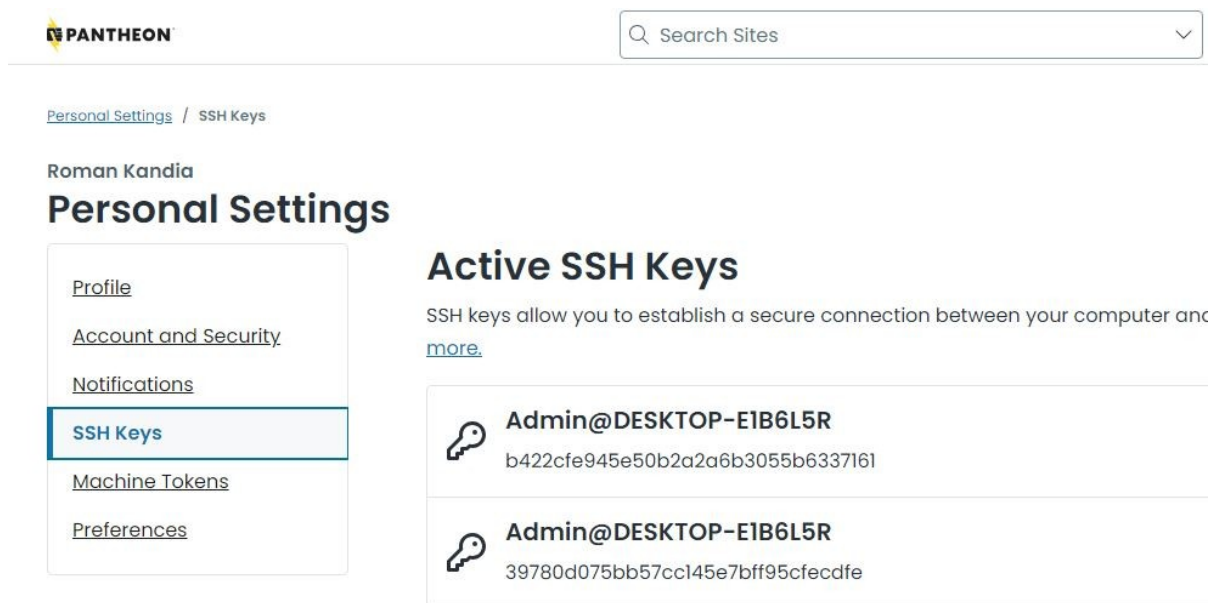


Рис 3.7. Додавання публічного ключа ssh

3.1 Розробка та наповнення веб-сайту

Розробка веб-сайту для продажу комп'ютерної периферії проходила відповідно до завдання та з посиланням на різний рівень володіння інформаційними технологіями користувачами.

Для реалізації проекту було зроблено вибір в бік використання системи CMS « Drupal».

Для початку роботи з певною розробкою проекту, тестуванням проекту, налагодженням проекту, найперше потрібно було зареєструватися на сайті [https:// Drupal.com](https://Drupal.com).

Після закінчення, реєстрації, перейшли до самого процесу створення сайту. На цьому етапі пробно було провести аналіз існуючих шаблонів та обрати підходящий для нашого сайту.

Створивши кабінет адміністратора з логіном і паролем, користувач може починати роботу з розробленням сайту. Він потрапляє на початкову сторінку кабінету.

Ліворуч є меню з різними налаштуваннями, шаблонами і т.д.

Вибравши опцію «ВИГЛЯД», знаходимо підходящу тему, використовуючи «ПОШУК» у правому верхньому куті ввівши слово «Shop» (рис. 3.8).

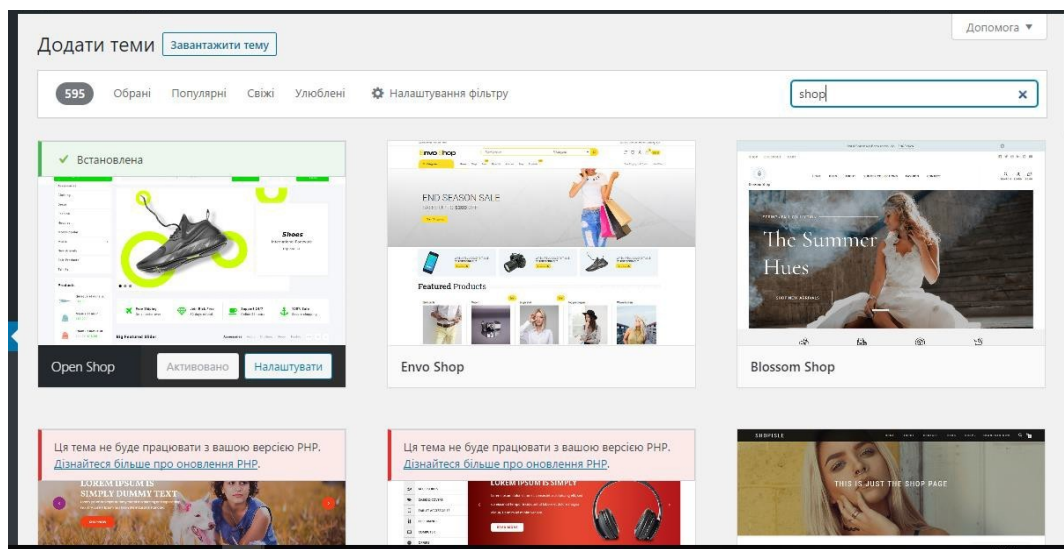


Рис. 3.8. Підбір теми

Після цього, переходимо до встановлення та активації цієї теми. Потім продовжуємо редагувати сайт і задаємо потрібну кількість сторінок сайту.

Далі було створено такі сторінки веб-сайту:

- Акаунт;
- Блог;
- Головна сторінка;
- Зразок сторінок;
- Кошик;
- Магазин;
- Напишіть нам;
- Оформлення замовлення;
- Політика конфіденційності;
- Список бажань.

Кожна сторінка веб-додатку має логотип веб-сайту та перелік відповідної інформації, що відповідає назви сторінки.

На сторінці «ГОЛОВНА» міститься загальна інформація про магазин. Ліворуч є меню магазину, які продукти є в асортименті. По середині є пошук по сайту, праворуч міститься контактна інформація (рис. 3.9).

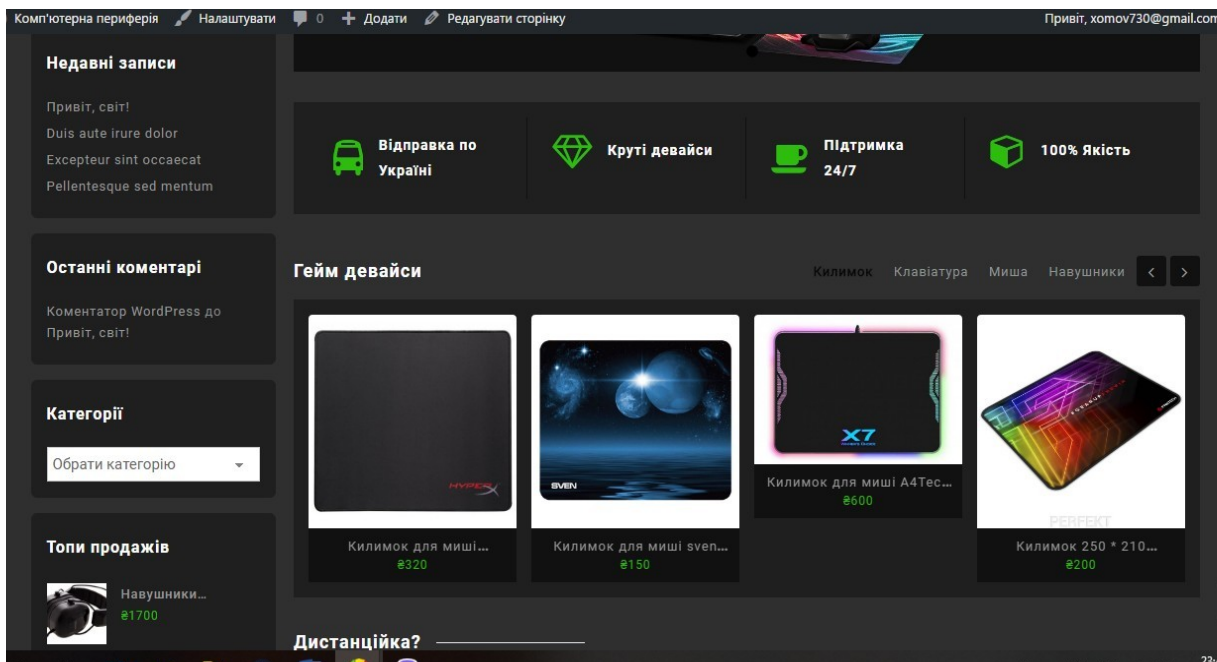


Рис. 3.9 Сторінка «ГОЛОВНА»

Сторінка «БЛОГ» містить в собі інформацію про історію життя сайту, цікаві новини та пропозиції (рис. 3.10).

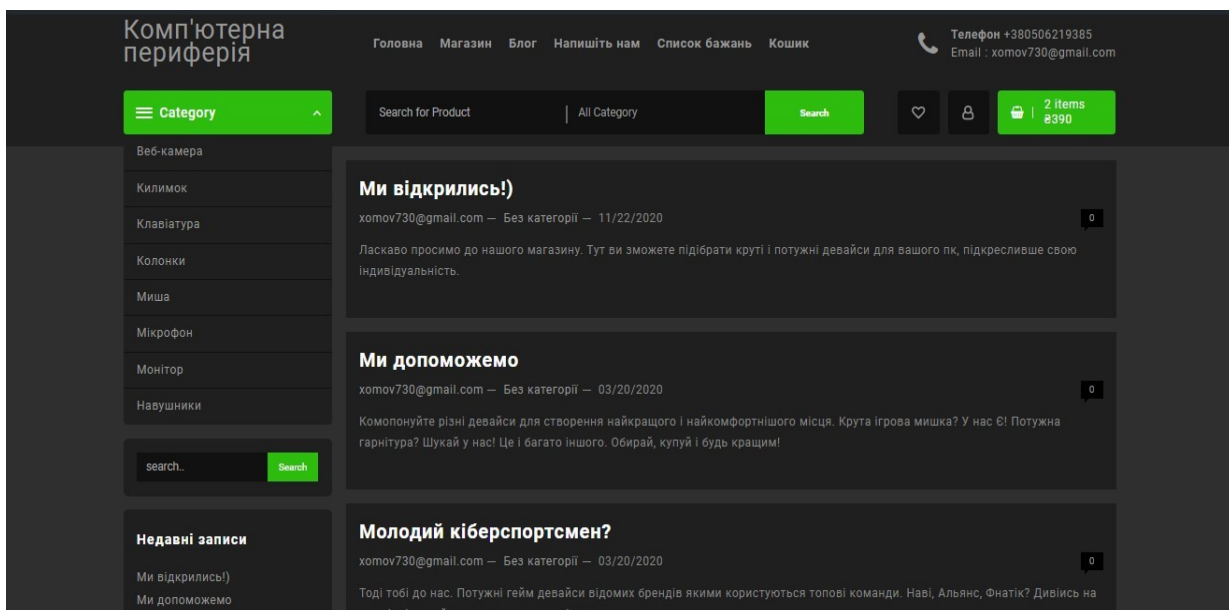


Рис. 3.10 Сторінка «БЛОГ»

Сторінка «МАГАЗИН» має посилання на різні категорії товарів, для зручності пошуку потрібного товару для клієнта (рис. 3.11 - 3.13).

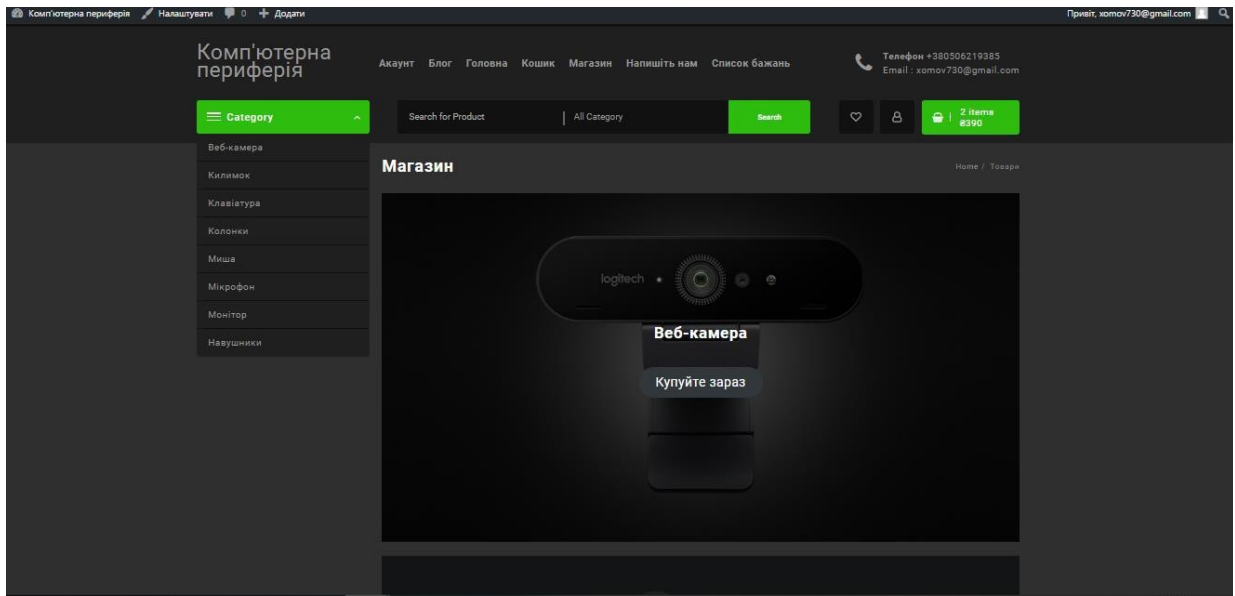


Рис. 3.11. Сторінка «МАГАЗИН»

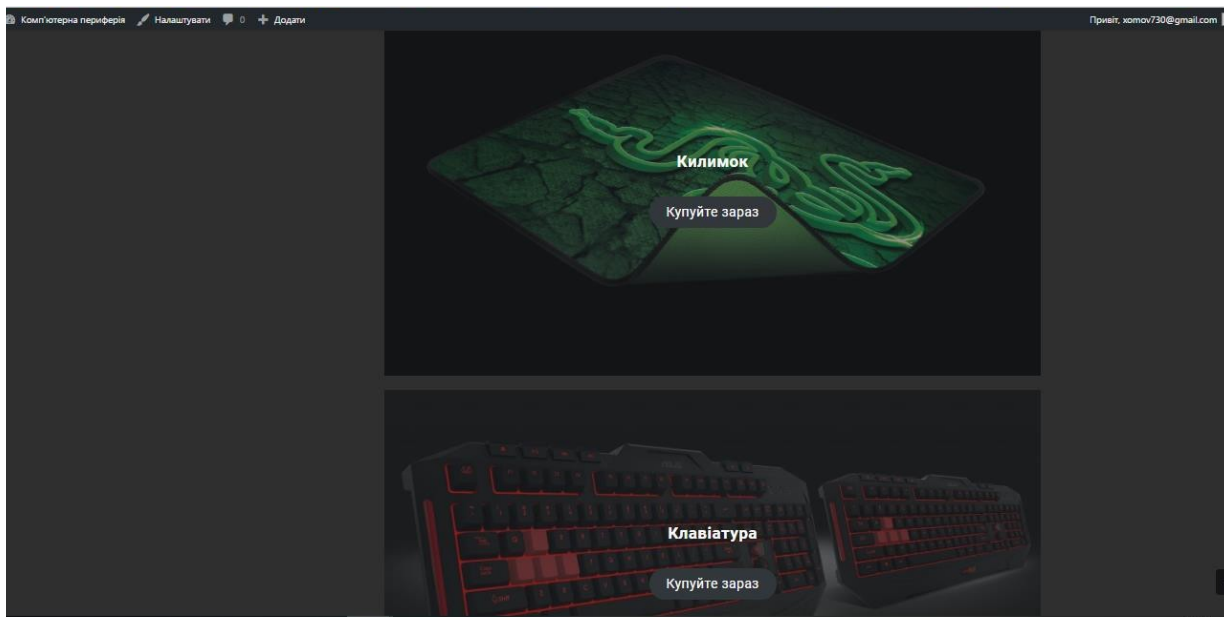


Рис. 3.12. Сторінка «МАГАЗИН»

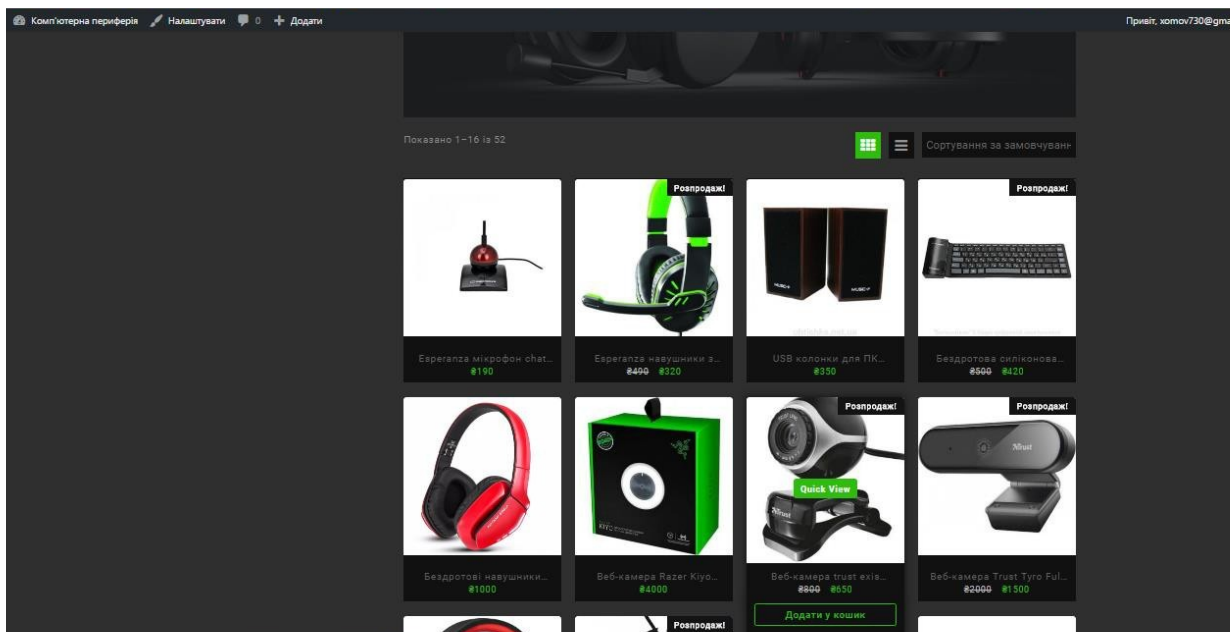


Рис. 3.13 Сторінка «МАГАЗИН»

Сторінка «КОШИК» є важливою для покупця, адже саме сюди відправляються вибрані для покупок товари клієнта (рис. 3.14).

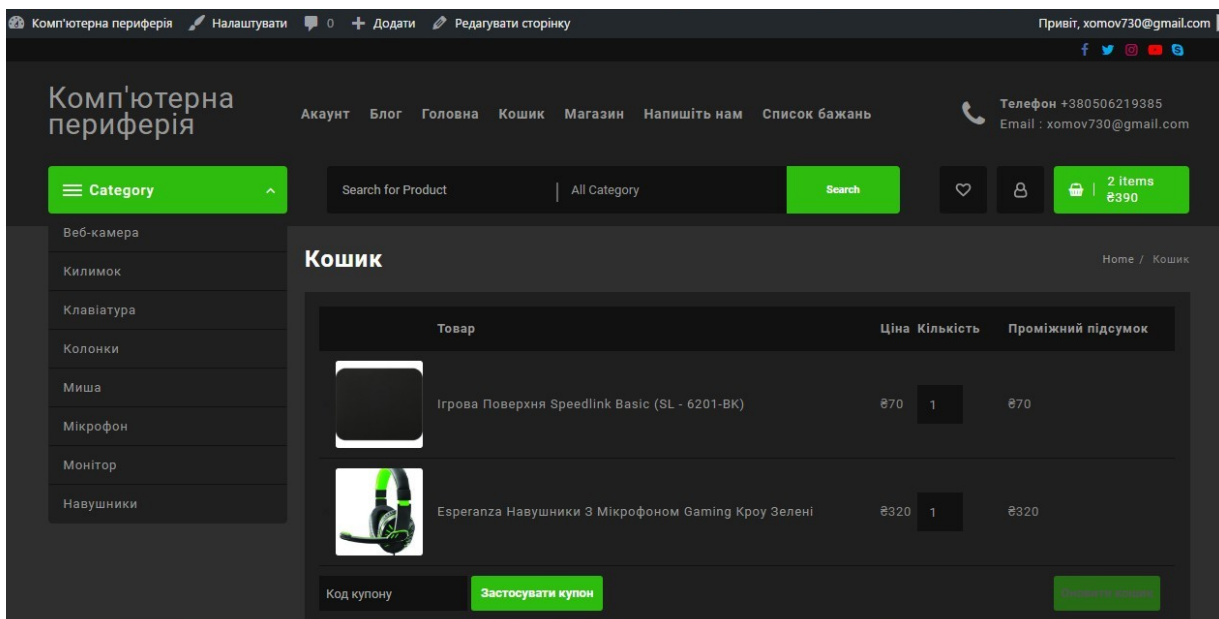


Рис. 3.14 Сторінка «КОШИК»

Відкриваючи сторінку «НАПИШІТЬ НАМ» клієнт має можливість контактувати з інформаційними менеджерами, задати питання, написати пропозиції щодо роботи сайту та товарів (рис. 3.15).

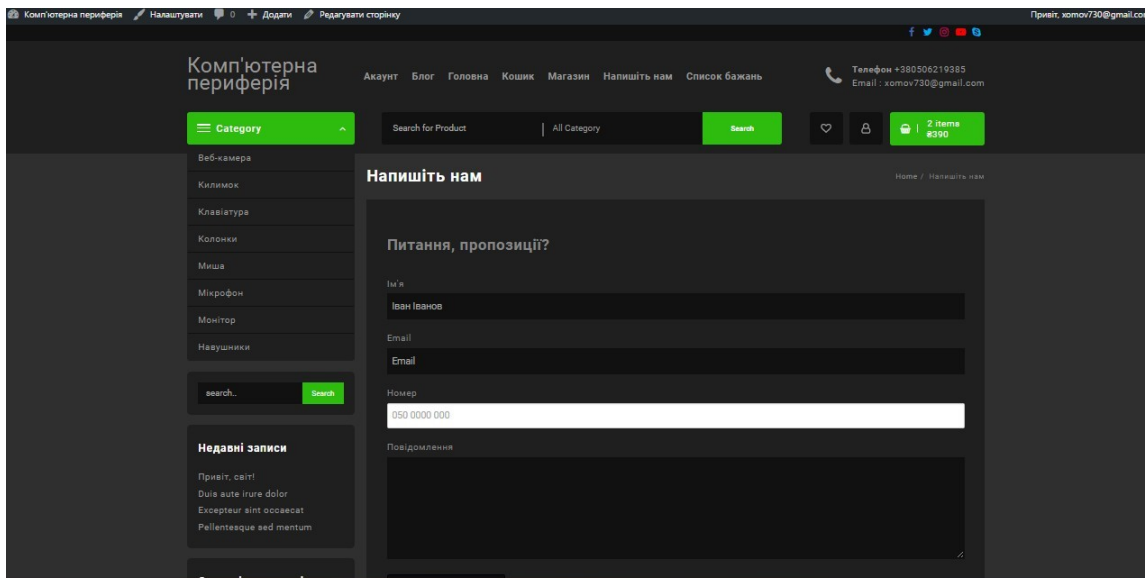


Рис. 3.15 Сторінка «НАПИШІТЬ НАМ»

Ведення веб-сайту з продажу комп'ютерної периферії вимагає відповідального ставлення до своїх обов'язків, підтримки всіх діючих сторінок і т.д.

3.2. Використання веб-сайту

Для тестування веб-сайту і використання його можливостей та опцій в повній мірі, потрібно придбати домен на хостинг-платформі, пробна версія безкоштовного використання домену становить два місяці. Покупку Інтернет- домену, щоб спробувати використання та застосування сайту на практиці, а потім уже при оптимальній роботі та детальному тесті можливостей сайту оформлювати покупку не було оформлено. Хостинг дає доступ до використання фізичного місця з метою зберігання даних на віддаленому сервері. З допомогою хостингу веб-сайт знаходить у відкритому режимі для користувачів і їх відвідування сайту.

Якість хостингу впливає на коректну роботу сайту та можливість реалізації його серед користувачів.

Безкоштовні хостингові системи мають такі недоліки:

- мало пам'яті;
- ненадійність;
- відсутність деяких важливих функцій.

Проплативши хостинг, було отримано низку переваг та додаткових функцій, за оплату, від 1 долара в місяць.

- Тестуючи веб-сайт, маємо враховувати такі параметри:
- правильна робота всіх обов'язкових функцій сайту;
- можливість функцій зворотнього зв'язку, при різних умовах використання сайту користувачем (відкриття з різних браузерів, з різним розширенням екрану);
- коректна робота пошуку;
- відповідність контенту та наповнень усіх сторінок.

Перевіривши веб-сайт у тестовому режимі, можна запускати його у повноцінне використання, користувачі можуть заходити на початкову головну сторінку сайту, та починати свої покупки в Інтернет-магазині.

ВИСНОВКИ

У сучасних умовах стрімкого розвитку електронної комерції створення якісного спеціалізованого інтернет-магазину є важливим конкурентним інструментом. Виконана кваліфікаційна робота підтверджує актуальність обраної теми.

У першому розділі проведено аналіз ринку комп'ютерної периферії в Україні, досліджено конкурентне середовище та порівняно популярні системи керування контентом. Встановлено, що Drupal 10 є одним з найкращих рішень для складних комерційних проєктів завдяки модульній архітектурі, високому рівню безпеки та потужним інструментам електронної комерції.

Другий розділ присвячено проєктуванню. Розроблено структурно-функціональну модель системи, спроектовано базу даних на основі Entity API, визначено технологічний стек (PHP 8.2+, Symfony, Composer, Git, Pantheon).

У третьому розділі реалізовано функціональний інтернет-магазин. Впроваджено каталог товарів з розширеними фільтрами, систему кошика та оформлення замовлень на базі Drupal Commerce, адаптивний інтерфейс, особистий кабінет та інструменти адміністрування. Проведено тестування основних сценаріїв використання.

Розроблений програмний продукт повністю відповідає поставленим завданням, демонструє практичне застосування сучасних веб-технологій і може бути використаний як реальний комерційний ресурс або основа для подальшого розвитку (перехід на Drupal 11, headless-архітектура, інтеграція з ERP-системами).

Отримані результати підтверджують ефективність використання Drupal для створення сучасних інтернет-магазинів спеціалізованої ніші.

Перспективи подальших досліджень: оновлення до Drupal 11,

впровадження ШІ-рекомендацій товарів, розширення інтеграцій з платіжними та логістичними сервісами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. WordPress Codex. WordPress Documentation URL: <https://wordpress.org/documentation>.
2. WordPress URL: <https://en.wikipedia.org/wiki/WordPress>
3. Joomla. URL: <https://en.wikipedia.org/wiki/Joomla>
4. Joomla. URL: <https://www.joomla.org/>
5. Joomla Documentation. Official Guide. URL: <https://docs.joomla.org/Portal:Beginners/en>
6. Drupal 10 Module Development: Develop and Deliver Engaging and Intuitive Enterprise-Level Apps. – de Gruyter GmbH, Walter, 2023.
7. Quillen K. Drupal 10 Development Cookbook: Practical Recipes to Harness the Power of Drupal for Building Digital Experiences and Dynamic Websites / K. Quillen. – de Gruyter GmbH, Walter, 2023.
8. Drupal. URL: <https://en.wikipedia.org/wiki/Drupal>
9. Drupal Developer Documentation URL: <https://www.drupal.org/docs>
10. Darlack J. M. Diktuon: drupal - CMS and beyond. J. M. Darlack. Theological librarianship. 2013. Vol. 6, no. 2. P. 1–3.
11. Chevance R. J. Server Architectures: Multiprocessors, Clusters, Parallel Systems, Web Servers, Storage Solutions .R. J. Chevance. Elsevier Science & Technology Books, 2004.
12. Darlack J. M. Diktuon: drupal. CMS and beyond. J. M. Darlack. Theological librarianship. 2013. Vol. 6, no. 2. P. 1–3.
13. Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, Incorporated, 2020. 250 p.
14. Платформа Drupal. Overview. Drupal URL: https://www.drupal.org/docs/user_guide/en/index.html
15. UML Diagrams. URL: <https://www.uml-diagrams.org/>

16. Li P. Jira 8 Essentials: Effective project tracking and issue management with enhanced Jira 8.21 and Data Center features, 6th Edition. P. Li. Packt Publishing, 2022. 412 p.
17. Newman S. Building Microservices: Designing Fine-Grained Systems. Sam Newman. Beijing; Boston : O'Reilly Media, 2015. 280 p.
18. PHP: Hypertext Preprocessor. URL: <https://www.php.net/>
19. Документація MySQL. URL: <https://dev.mysql.com/doc/>
20. Working with coders: A guide to software development for the perplexed non-techie. Apress. 2017. 220 p.