

Міністерство освіти і науки України  
Університет митної справи та фінансів

Факультет інноваційних технологій  
Кафедра комп'ютерних наук та інженерії програмного забезпечення

**Кваліфікаційна робота бакалавра**  
на тему: «Розробка web-додатку підтримки роботи квіткового магазину»

Виконав: студент групи ІПЗ 20-2  
Спеціальність 121 «Інженерія програмного  
забезпечення»  
Синенко Роман Віталійович  
(прізвище та ініціали)

Керівник: к.ф.-м.н., доц. Лебідь О. Ю.  
(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент: Спеціалізоване управління розробки  
та супроводження програмного забезпечення  
Департаменту з питань цифрового розвитку,  
цифрових трансформацій та цифровізації ДМСУ  
(місце роботи)

головний державний інспектор відділу розробки  
програмного забезпечення  
(посада)

Бахтін О. В.  
(науковий ступінь, вчене звання, прізвище та ініціали)

Дніпро – 2024

## АНОТАЦІЯ

Синенко Роман Віталійович. Розробка веб-додатку для підтримки роботи квіткового магазину.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за спеціальністю 121 «Інженерія програмного забезпечення». – Університет митної справи та фінансів, Дніпро, 2024.

Дана кваліфікаційна робота присвячена розробці веб-додатку для підтримки роботи квіткового магазину на платформі OpenCart. У роботі розглянуто сучасні методи та інструменти електронної комерції, проведено аналіз конкурентних програмних продуктів та визначено оптимальні технології для реалізації проєкту. Проведено проектування та розробку онлайн-магазину, включаючи дизайн інтерфейсу, інтеграцію бази даних, налаштування SEO та оптимізацію продуктивності сайту.

Отримані результати мають практичне значення, оскільки забезпечують зручний інструмент для управління квітковим бізнесом, покращують користувацький досвід та підвищують ефективність комерційної діяльності. Розроблений веб-додаток є функціональним, надійним та безпечним, що сприяє задоволенню потреб клієнтів та зростанню бізнесу.

Ключові слова: РОЗРОБКА, ЕЛЕКТРОННА КОМЕРЦІЯ, OPENCART, КВІТКОВИЙ МАГАЗИН, ВЕБ-ДОДАТОК.

## ABSTRACT

Syenko Roman Vitaliyovych. Development of a Web Application to Support the Operation of a Flower Shop.

Qualification work for the degree of Bachelor in the specialty 121 "Software Engineering". – University of Customs and Finance, Dnipro, 2024.

This qualification work is devoted to the development of a web application to support the operation of a flower shop on the OpenCart platform. The study considers modern methods and tools of e-commerce, analyzes competitive software products, and identifies optimal technologies for project implementation. The project involved designing and developing an online store, including interface design, database integration, SEO settings, and performance optimization.

The results obtained are of practical significance as they provide a convenient tool for managing a flower business, improve user experience, and increase commercial efficiency. The developed web application is functional, reliable, and secure, meeting the needs of customers and contributing to business growth.

Keywords: DEVELOPMENT, E-COMMERCE, OPENCART, FLOWER SHOP, WEB APPLICATION.

## ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ В УКРАЇНІ .....	8
1.1 Аналіз напрямків розвитку електронної комерції .....	8
1.2 Огляд і вибір оптимальних засобів реалізації.....	12
1.3 Постановка задачі.....	20
1.4 Висновки до першого розділу.....	22
РОЗДІЛ 2. РОЗРОБКА ТА ВПРОВАДЖЕННЯ ОНЛАЙН-МАГАЗИНУ .....	24
2.1 Планування та дизайн .....	24
2.2 Хостинг та SSL сертифікати .....	34
2.3 Підключення OpenCart та інтеграція бази даних.....	38
2.4 Висновки до другого розділу .....	43
РОЗДІЛ 3. УПРАВЛІННЯ ТА ОПТИМІЗАЦІЯ ОНЛАЙН-МАГАЗИНУ .....	45
3.1 Адміністративна панель OpenCart.....	45
3.2 Опис головних сторінок сайту .....	49
3.3 SEO налаштування та оптимізація продуктивності .....	57
3.4 Висновки до третього розділу .....	63
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	67
ДОДАТОК.....	70

## ВСТУП

Актуальність дослідження. У сучасному світі електронна комерція є одним із найважливіших напрямків розвитку бізнесу. Інтернет-магазини забезпечують зручність користувачам, дозволяючи здійснювати покупки в будь-який час та з будь-якого місця. Особливо важливо це для квіткових магазинів, де швидкість обробки замовлень та своєчасна доставка є критичними для задоволення потреб клієнтів. Розвиток інформаційних технологій та зростання популярності мобільних пристроїв сприяли збільшенню кількості онлайн-платформ, що спеціалізуються на продажу різноманітних товарів і послуг. Серед них особливе місце займають онлайн-магазини квітів, які дозволяють споживачам швидко та зручно вибрати букети для різних подій, оформляти замовлення з доставкою додому або до місця проведення заходу. Веб-додатки для квіткових магазинів значно спрощують процес вибору та замовлення квітів, надаючи користувачам широкий асортимент, детальні описи товарів, можливість залишати відгуки та оцінювати якість продукції.

Значення електронної комерції ще більше зросло у зв'язку з розвитком цифрової економіки та зростанням кількості користувачів Інтернету. Це підштовхнуло підприємства до активного розвитку та вдосконалення своїх онлайн-ресурсів. Для квіткових магазинів важливо мати інтуїтивно зрозумілий, функціональний та безпечний веб-додаток, який забезпечить високу продуктивність та задовольнить потреби клієнтів.

Метою даної роботи є розробка веб-додатку для підтримки роботи квітового магазину на платформі OpenCart, який забезпечить зручний інтерфейс для користувачів, можливість швидкого пошуку товарів, зручну систему кошика та оформлення замовлення, а також безпечні методи оплати і доставки. Розробка такого додатку включає в себе не лише технічні аспекти, але й врахування потреб кінцевих користувачів, створення привабливого дизайну та забезпечення високої продуктивності сайту.

Завдання роботи включають:

1. Аналіз сучасного стану електронної комерції в Україні та визначення основних вимог до квіткових онлайн-магазинів. Це включає вивчення ринку електронної комерції, визначення ключових гравців та аналіз їхніх сильних та слабких сторін.

2. Проведення аналізу конкурентних програмних продуктів та визначення оптимальних технологій для реалізації проекту. Це дозволить обрати найкращу платформу для розробки та забезпечити ефективне функціонування веб-додатку.

3. Проектування архітектури веб-додатку, розробка дизайну інтерфейсу та інтеграція бази даних. Важливими аспектами є зручність користування, швидкість завантаження сторінок та адаптивність дизайну.

4. Налаштування хостингу, встановлення SSL сертифіката та забезпечення безпеки даних користувачів. Безпека даних користувачів є критично важливою для успішної роботи будь-якого онлайн-магазину.

5. Оптимізація продуктивності сайту та налаштування SEO для підвищення видимості у пошукових системах. Це включає налаштування мета-тегів, оптимізацію контенту та забезпечення швидкого завантаження сторінок.

Об'єкт дослідження: сучасні технології розробки та підтримки веб-додатків. Предметом дослідження є веб-додаток, розроблений на платформі OpenCart для забезпечення ефективної роботи квітового магазину.

Методи дослідження включають методи аналізу та обробки інформації, проектування програмного забезпечення, проектування баз даних та методи забезпечення безпеки даних.

Результати продемонстровані в кваліфікаційній роботі відповідають наступним результатам навчання освітньої програми 121 «Інженерія програмного забезпечення»:

ПР1. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПР3. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПР6. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПР8. Вміти розробляти людино-машинний інтерфейс.

ПР11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.

ПР18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

ПР22. Знати та вміти застосовувати методи та засоби управління проектами.

Практичне значення одержаних результатів полягає у створенні зручного інструменту для управління квітковим бізнесом, що покращує користувацький досвід та підвищує ефективність комерційної діяльності. Розроблений веб-додаток забезпечує зручний інтерфейс для користувачів, швидкий доступ до товарів та послуг, ефективну систему управління замовленнями та безпечні методи оплати. Крім того, забезпечення високої продуктивності та SEO-оптимізації сайту сприяє залученню більшої кількості клієнтів та збільшенню обсягів продажів.

Структура кваліфікаційної роботи: вступ, 3 розділи, висновок, перелік використаних джерел, додаток. Обсяг роботи - 95 сторінок, робота містить 29 рисунків, перелік використаних джерел складається з 30 посилань, додаток розміщено на 29 сторінках.

## РОЗДІЛ 1. ДОСЛІДЖЕННЯ ЕЛЕКТРОННОЇ КОМЕРЦІЇ В УКРАЇНІ

### 1.1 Аналіз напрямків розвитку електронної комерції

У 2020 році галузь електронної комерції в Україні продемонструвала стрімке зростання. За даними дослідження, проведеного компаніями Soul Partners, Baker Tilly Україна та Aequo, обсяг ринку e-commerce збільшився на 41% порівняно з попереднім роком, ставши однією з найбільш динамічних галузей в країні. Вартість ринку електронної торгівлі досягла 4 мільярдів доларів США, що становить 88% від загального обсягу роздрібних продажів. Важливо відзначити, що за останні п'ять років ринок e-commerce в Україні майже потроївся [1].

Прогнози на наступні п'ять років свідчать про продовження зростання цього сектора - очікується, що обсяг електронної комерції подвоїться протягом цього періоду. Дослідники визначили кілька ключових факторів, які сприяли такому стрімкому розвитку галузі e-commerce в Україні. Серед них - збільшення рівня проникнення інтернету та поширення смартфонів серед населення, обмеження, пов'язані з пандемією COVID-19, що стимулювали перехід до онлайн-покупок, а також зростання довіри споживачів до цифрових платіжних систем.

Динаміка ринку електронної комерції та її частка у роздрібній торгівлі в Україні ілюструє (див. рис. 1.1) величезний потенціал цього сектора.



Рисунок 1.1 – Динаміка розвитку електронної комерції в Україні



Електронна комерція стрімко розвивається у всьому світі, ставши невід'ємною частиною сучасного бізнесу. Інтернет-магазини дозволяють компаніям охоплювати ширшу аудиторію, знижувати операційні витрати та підвищувати ефективність продажів. Розглянемо ключові напрямки розвитку електронної комерції та актуальні тренди галузі.

Мобільна комерція продовжує швидко розвиватися, стаючи важливим елементом сучасного бізнесу. Основні тенденції та статистичні дані про мобільну комерцію підкреслюють її вплив та перспективи. За даними дослідження, до 2026 року мобільні гаманці становитимуть понад 54% онлайн-платежів по всьому світу. У 2022 році цей метод став найпопулярнішим для онлайн-шопінгу, займаючи приблизно половину всіх транзакцій. Це пояснюється зручністю використання та швидким зростанням кількості смартфонів.

Присутність у соціальних мережах може збільшити продажі до 32% порівняно з традиційними методами. Активна стратегія в соціальних мережах допомагає досягти ширшої аудиторії, підвищити обізнаність про бренд і створити спільноту навколо продуктів. Очікується, що до 2024 року 80% голосових пошуків будуть розмовного характеру. Це означає, що компанії повинні адаптувати свої SEO-стратегії для відповідності природному мовленню та питанням користувачів. Доповнена реальність (AR) в онлайн-шопінгу також набуває популярності, оскільки 61% споживачів надають перевагу ритейлерам, які пропонують AR-досвід під час покупок [2]. Це робить онлайн-шопінг більш захоплюючим та персоналізованим, дозволяючи клієнтам візуалізувати продукти у своєму просторі.

Споживачі все частіше використовують мобільні додатки для покупок. 51% людей відкривають мобільний додаток до десяти разів на день, що значно підвищує ймовірність здійснення покупок [3].

У таблиці 1.1 представлено порівняння продажів мобільної електронної комерції у мільярдах доларів США за останні кілька років [4].

Таблиця 1.1

## Порівняння продажів мобільної електронної комерції

Рік	Продажі мобільної електронної комерції (у мільярдах доларів США)
2019	1,166
2020	1,530
2021	1,922
2022	1,945
2023	2,169
2024	2,522

Персоналізований маркетинг – це стратегія, що базується на використанні аналітики великих даних (Big Data) та штучного інтелекту (AI) для створення індивідуальних пропозицій для кожного клієнта. Завдяки цьому підходу компанії можуть значно підвищити ефективність своїх маркетингових кампаній [5].

Аналітичні інструменти збирають велику кількість інформації про поведінку користувачів, їхні уподобання, історію покупок та інші важливі дані. Ці дані аналізуються за допомогою AI, що дозволяє виявляти приховані закономірності та тренди. На основі цих аналізів компанії можуть створювати персоналізований контент, рекомендації та рекламні оголошення, які найбільше відповідають інтересам та потребам кожного клієнта [6].

Персоналізований маркетинг підвищує лояльність клієнтів та стимулює повторні покупки. Клієнти цінують індивідуальний підхід, що призводить до підвищення їхньої лояльності та стимулювання повторних покупок. Використання AI для аналізу даних допомагає компаніям швидко реагувати на

мінливі потреби ринку, що дозволяє залишатися конкурентоспроможними. Така персоналізація дає змогу створювати більш ефективні рекламні кампанії, орієнтовані на конкретні сегменти аудиторії, знижуючи витрати на рекламу та підвищуючи її результативність.

Багато успішних компаній вже використовують персоналізований маркетинг для досягнення своїх бізнес-цілей. Наприклад, великі онлайн-ритейлери, такі як Amazon та Netflix, активно використовують AI для створення персоналізованих рекомендацій товарів та контенту, що підвищує їхні продажі та залученість клієнтів [7][8].

Соціальна комерція в Україні набуває все більшого значення, адже інтеграція продажів через соціальні медіа платформи дозволяє компаніям безпосередньо взаємодіяти зі своїми клієнтами. Це створює зручний та інтерактивний досвід покупок, що значно підвищує залученість та лояльність клієнтів.

Основні тенденції соціальної комерції в Україні включають зростання ринку соціальної комерції, популярність соціальних платформ, інфлюенсер-маркетинг та впровадження нових функцій. Ринок соціальної комерції в Україні демонструє значний ріст, зокрема завдяки впровадженню нових технологій та зміні поведінки споживачів. За даними експертів, обсяг продажів через соціальні медіа платформи зростає, і все більше компаній інвестують у цей канал для збільшення продажів та підвищення впізнаваності бренду [9]. Найпопулярнішими платформами для соціальної комерції в Україні є Facebook, Instagram та TikTok. Вони пропонують різноманітні інструменти для бізнесу, такі як магазини на платформах, можливість продажу через живі трансляції, використання рекламних оголошень та інтеграцію з платформами електронної комерції [10].

Інфлюенсер-маркетинг відіграє важливу роль у соціальній комерції. Багато брендів співпрацюють з популярними блогерами та лідерами думок, які мають значний вплив на свою аудиторію. Це дозволяє ефективно просувати товари та

послуги, збільшуючи продажі та залученість клієнтів. Соціальні медіа платформи постійно впроваджують нові функції, які полегшують процес покупок. Наприклад, Instagram та Facebook мають функції для створення магазинів безпосередньо на платформі, що дозволяє користувачам переглядати та купувати товари без необхідності переходити на інші сайти.

Окрім цього, соціальна комерція дозволяє брендам створювати інноваційні маркетингові стратегії, що включають використання відеоконтенту, анімованих оголошень та інтерактивних історій, які залучають користувачів та утримують їх увагу. Ці стратегії не лише допомагають збільшити продажі, але й сприяють побудові міцного емоційного зв'язку з брендом, що в свою чергу підвищує лояльність клієнтів.

Соціальна комерція надає ряд переваг для українського ринку. Це включає зручність для клієнтів, які можуть здійснювати покупки безпосередньо через соціальні медіа, підвищення залученості клієнтів через інтерактивні інструменти та можливість створення персоналізованого контенту. Для українських компаній соціальна комерція є ефективним інструментом для досягнення своїх бізнес-цілей. Вона дозволяє залучати нових клієнтів, підвищувати впізнаваність бренду та збільшувати продажі завдяки використанню соціальних медіа платформ.

## 1.2 Огляд і вибір оптимальних засобів реалізації

Вибір платформи для електронної комерції є критичним для успішного запуску та розвитку онлайн-магазину. Основними критеріями вибору є функціональність, масштабованість, вартість, простота використання та підтримка платіжних систем. У цьому розділі буде проведено огляд та порівняння основних платформ, таких як OpenCart, Magento, WooCommerce та Shopify [11].

OpenCart – це популярна платформа для створення інтернет-магазинів, відома своєю відкритістю коду та широкими можливостями налаштування. Вона

забезпечує підтримку мультимовності, інтеграцію з багатьма платіжними системами та велику кількість модулів і тем для розширення функціональності.

Переваги OpenCart:

- безкоштовна та з відкритим вихідним кодом.
- простота у використанні та налаштуванні.
- велика кількість модулів та тем.
- підтримка мультимовності та багатовалютності.

Недоліки OpenCart:

- може вимагати технічних знань для складних налаштувань.
- менша спільнота підтримки порівняно з деякими іншими платформами.

Magento є однією з найпотужніших платформ для електронної комерції, що підходить для великих і середніх підприємств. Вона пропонує розширені можливості для налаштування та масштабування, підтримку інтеграції з численними платіжними системами та сторонніми сервісами.

Переваги Magento:

- висока масштабованість.
- широкі можливості налаштування.
- підтримка інтеграції з численними платіжними системами та сторонніми сервісами.

- велика спільнота та підтримка.

Недоліки Magento:

- висока вартість впровадження та підтримки.
- складність у використанні для новачків.
- вимоги до ресурсів серверу.

WooCommerce – це плагін для WordPress, який перетворює будь-який сайт на платформі WordPress у повноцінний інтернет-магазин. WooCommerce популярний серед малого та середнього бізнесу завдяки своїй простоті у використанні та інтеграції з WordPress.

Переваги WooCommerce:

- інтеграція з WordPress.
- простота у використанні та налаштуванні.
- велика кількість плагінів та тем.
- відкритий вихідний код та безкоштовний.

Недоліки WooCommerce:

- обмежена масштабованість для великих проектів.
- може вимагати додаткових плагінів для розширеної функціональності.
- залежність від WordPress.

Shopify є однією з найпопулярніших SaaS-платформ для створення інтернет-магазинів. Вона пропонує готові рішення для створення та управління онлайн-магазинами без необхідності в технічних знаннях. Shopify підтримує інтеграцію з численними платіжними системами та маркетинговими інструментами.

Переваги Shopify:

- простота у використанні.
- швидкий запуск інтернет-магазину.
- підтримка інтеграції з численними платіжними системами та маркетинговими інструментами.

- вбудовані інструменти для SEO та маркетингу.

Недоліки Shopify:

- щомісячна абонентська плата.
- обмежені можливості налаштування у порівнянні з платформами з відкритим вихідним кодом.
- вартість додаткових додатків та плагінів.

Порівняння основних платформ для електронної комерції наведено у таблиці 1.2

Таблиці 1.2

## Порівняння основних платформ для електронної комерції

Критерій	OpenCart	Magento	WooCommerce	Shopify
Простота використання	Середня	Складна	Висока	Висока
Вартість впровадження	Низька	Висока	Низька	Середня
Масштабованість	Висока	Висока	Середня	Середня
Функціональність	Висока	Дуже висока	Висока	Висока
Підтримка платіжних систем	Висока	Висока	Середня	Висока

На основі проведеного аналізу було прийнято рішення використовувати платформу OpenCart для квіткового онлайн-магазину. Основні переваги OpenCart включають відкритий вихідний код, безкоштовність використання та широку функціональність, що дозволяє легко адаптувати платформу під специфічні потреби бізнесу.

Для успішної реалізації квіткового онлайн-магазину на платформі OpenCart було ретельно підібрано низку технологій та інструментів, які забезпечать надійну, безперебійну та ефективну роботу веб-сайту. Правильний вибір технологій є критично важливим для створення привабливого та зручного для користувачів інтерфейсу, а також для забезпечення необхідної функціональності та продуктивності системи.

JavaScript – це одна з основних клієнтських мов програмування для веб-розробки. Вона забезпечує інтерактивність та динамічність веб-сторінок, що значно покращує досвід користувачів. JavaScript дозволяє створювати різноманітні інтерактивні елементи, такі як анімації, випадаючі меню, слайдери зображень, інтерактивні форми та багато іншого. Ці динамічні функції роблять

веб-сторінки більш привабливими та зручними для користувачів, а також допомагають залучати та утримувати відвідувачів на сайті. Крім того, JavaScript відіграє важливу роль у взаємодії з браузером та маніпулюванні веб-сторінками в режимі реального часу. Він дозволяє отримувати доступ до різних елементів DOM (Document Object Model) та змінювати їх вміст, стилі або поведінку, надаючи можливість створювати динамічні веб-додатки, які реагують на дії користувача. JavaScript також широко використовується для валідації форм, обробки подій та асинхронних запитів до серверів за допомогою технології AJAX [12].

HTML (HyperText Markup Language) та CSS (Cascading Style Sheets) є базовими технологіями для створення структури та візуального оформлення веб-сторінок. HTML використовується для визначення семантичної структури контенту, такого як заголовки, абзаци, списки, таблиці та інші елементи. CSS, у свою чергу, відповідає за візуальне оформлення цих елементів, включаючи налаштування кольорів, шрифтів, розмірів, відступів, фонів та багато іншого. Ці дві технології є невід'ємними складовими для створення зручного, привабливого та доступного інтерфейсу користувача, який відповідає сучасним стандартам веб-дизайну [13].

PHP (Hypertext Preprocessor) є серверною мовою програмування, яка широко використовується для розробки веб-додатків, таких як інтернет-магазини, блоги, форуми та інші динамічні веб-сайти. Вона забезпечує обробку даних на сервері, інтеграцію з базами даних та створення динамічних веб-сторінок [14]. Використання PHP в OpenCart дозволяє реалізувати всі необхідні функції для ефективного управління інтернет-магазином, такі як каталог товарів, кошик покупок, система замовлень, обробка платежів та багато іншого. PHP є однією з найпопулярніших мов для веб-розробки завдяки своїй простоті у вивченні, великій спільноті розробників та численним бібліотекам та фреймворкам, які значно полегшують процес розробки. Крім того, PHP



підтримується більшістю хостингових платформ, що забезпечує легку інтеграцію та розгортання веб-додатків.

SQL (Structured Query Language) є мовою програмування, яка використовується для роботи з реляційними базами даних. Вона дозволяє створювати, читати, оновлювати та видаляти дані з таблиць бази даних. У контексті квіткового онлайн-магазину SQL відіграє критично важливу роль для зберігання та управління інформацією про товари, категорії, замовлення, клієнтів та іншими даними, необхідними для функціонування інтернет-магазину. SQL забезпечує ефективні механізми для пошуку, фільтрації, сортування та агрегації даних, що дозволяє швидко отримувати необхідну інформацію та відображати її на веб-сторінках. Крім того, SQL підтримує складні запити та об'єднання даних з декількох таблиць, що є особливо важливим для систем електронної комерції з великою кількістю пов'язаних даних [15].

Для реалізації квіткового онлайн-магазину було обрано платформу OpenCart. OpenCart є потужною, гнучкою та масштабованою системою управління контентом (CMS) з відкритим вихідним кодом, яка спеціально розроблена для створення та керування інтернет-магазинами. Вона надає широкий спектр функцій та інструментів, необхідних для успішної реалізації онлайн-торгівлі [16].

Основні переваги OpenCart включають:

- мультимовність: Платформа підтримує кілька мовних версій, що дозволяє обслуговувати клієнтів з різних регіонів та країн, розширюючи потенційну аудиторію магазину.
- інтеграція з платіжними системами: OpenCart інтегрується з численними популярними платіжними системами, такими як PayPal, Stripe, Skrill та багатьма іншими. Це спрощує процес обробки платежів та забезпечує зручність для покупців під час оплати замовлень.
- розширюваність: Завдяки наявності великої кількості модулів, плагінів та тем оформлення, OpenCart легко розширюється та налаштовується відповідно

до специфічних потреб бізнесу. Це дозволяє додавати нові функції, інтегрувати додаткові сервіси та налаштовувати зовнішній вигляд магазину.

- зручний інтерфейс адміністративної панелі: OpenCart має інтуїтивно зрозумілий інтерфейс адміністративної панелі, що дозволяє ефективно керувати магазином без необхідності мати спеціальні технічні знання. Це полегшує процес додавання товарів, керування замовленнями, налаштування магазину та відстеження статистики продажів.

Для створення макетів та інтерактивних прототипів веб-сторінок квіткового онлайн-магазину використовується потужний інструмент для дизайну Figma. Figma є хмарним додатком, який дозволяє створювати високоякісні макети, прототипи та дизайни веб-сайтів, мобільних додатків та інтерфейсів користувача [17].

Основні переваги використання Figma включають:

- інтерактивні прототипи: Figma дозволяє створювати інтерактивні прототипи з можливістю переходу між сторінками, анімаціями та взаємодією з різними елементами. Це допомагає оцінити та покращити користувацький досвід ще до початку розробки.

- співпраця в реальному часі: Кілька дизайнерів та членів команди можуть працювати над одним проектом одночасно, вносячи зміни та коментарі в режимі реального часу. Це значно полегшує процес узгодження та внесення змін до дизайну.

- бібліотеки компонентів: Figma дозволяє створювати та використовувати бібліотеки компонентів, що забезпечує узгодженість дизайну та прискорює процес розробки.

- експорт файлів: Готові макети та прототипи можна експортувати у різних форматах, таких як PNG, SVG та PDF, для використання в процесі розробки веб-сайту.

Для розміщення онлайн-магазину було обрано Hostiq.ua [18] через його надійність, швидкість і підтримку необхідних технологій. Хостинг забезпечує

стабільну роботу сайту та швидкий доступ до нього користувачам, що є критично важливим для успішної роботи онлайн-магазину (див. рис. 1.2).

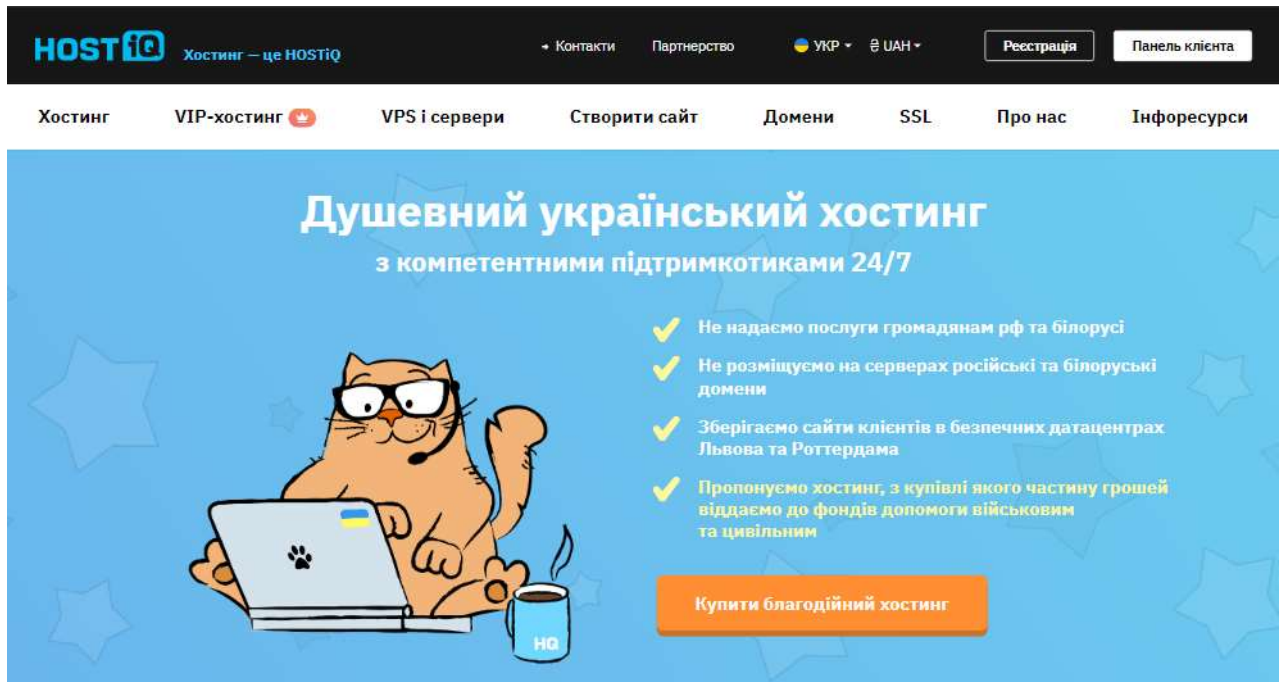


Рисунок 1.2 – Головна сторінка Hostiq.ua

Для захисту конфіденційної інформації використовуються SSL-сертифікати та інші засоби безпеки. SSL-сертифікат забезпечує шифрування даних, що передаються між браузером користувача та сервером, що підвищує довіру клієнтів до сайту. Крім того, OpenCart має вбудовані функції безпеки, такі як захист від крос-сайтового скриптингу (XSS), крос-сайтової підробки запитів (CSRF) та SQL-ін'єкцій, що допомагає запобігти злому та втраті даних.

Оптимізація продуктивності сайту включає використання інструментів для аналізу швидкості завантаження та забезпечення високої продуктивності. Основні методи оптимізації включають зменшення розміру зображень та інших ресурсів, використання кешування для зменшення навантаження на сервер та оптимізацію коду JavaScript та CSS. Ці заходи допомагають зробити сайт швидким та зручним для користувачів, що позитивно впливає на їхній досвід та показники конверсії.

SEO налаштування допомагають покращити видимість сайту у пошукових системах. Основні методи SEO оптимізації включають використання ключових слів у заголовках, описах та текстах, оптимізацію швидкості завантаження сайту, створення якісного та унікального контенту, а також використання внутрішніх та зовнішніх посилань. Ці заходи дозволяють залучити більше відвідувачів на сайт та підвищити його рейтинг у пошукових системах, що є критично важливим для успіху онлайн-магазину.

Система управління контентом OpenCart забезпечує широкі можливості для SEO оптимізації, включаючи генерацію SEO-дружніх URL, додавання мета-тегів, заголовків та описів для сторінок, створення сайтмапів та багато іншого.

Використання SSL-сертифікатів, оптимізація продуктивності та SEO налаштування разом з іншими обраними технологіями та інструментами забезпечать створення безпечного, швидкого та легко знайденого квіткового онлайн-магазину на платформі OpenCart.

### 1.3 Постановка задачі

Метою проекту є створення функціонального та зручного веб-сайту для продажу квітів на платформі OpenCart, який забезпечить зручний доступ до послуг флористики для широкого кола споживачів. Для досягнення цієї мети необхідно вирішити декілька ключових завдань.

Розробка макету та дизайну сайту включає створення макетів сторінок та прототипів для забезпечення зручного та привабливого інтерфейсу. Використання інструментів для дизайну, таких як Figma, дозволить створювати інтерактивні прототипи, що дозволять візуалізувати кінцевий продукт та внести необхідні корективи на ранніх етапах розробки.

Реалізація технічних аспектів проекту передбачає використання мов програмування (JavaScript, HTML, CSS, PHP, SQL) для створення динамічного та функціонального сайту, що забезпечить високу продуктивність і зручність

користування. Інтеграція з базами даних дозволить ефективно обробляти великі обсяги даних і забезпечувати безперебійну роботу сайту. Крім того, буде створена бекенд-логіка, яка оброблятиме запити користувачів, виконуватиме обчислення та забезпечуватиме належну роботу всіх функцій сайту.

Налаштування функціональних можливостей магазину включає реалізацію функцій авторизації та реєстрації користувачів, що забезпечить безпеку і зручність доступу до персоналізованих сервісів. Підтримка мультимовності та адаптивності забезпечить зручність користувачів з різних регіонів та пристроїв, що розширить аудиторію потенційних клієнтів. Інтеграція з платіжними системами забезпечить безпечні та зручні транзакції, що підвищить довіру користувачів до онлайн-магазину.

Оптимізація продуктивності сайту включає використання інструментів для аналізу швидкості завантаження та забезпечення високої продуктивності, що сприятиме покращенню користувацького досвіду. Зменшення розміру зображень та інших ресурсів, використання кешування для зменшення навантаження на сервер та оптимізація коду JavaScript та CSS забезпечать швидке завантаження сторінок і комфортне користування сайтом. Впровадження технологій стиснення даних прискорить передачу інформації між сервером та клієнтом.

Забезпечення безпеки є ключовим аспектом проекту. Використання SSL-сертифікатів та інших засобів безпеки для захисту конфіденційної інформації, що передається між браузером користувача та сервером, підвищить довіру клієнтів до онлайн-магазину. Реалізація багаторівневої системи безпеки забезпечить захист даних користувачів та запобігання несанкціонованого доступу до системи.

Впровадження стратегій електронного маркетингу включає оптимізацію сайту для пошукових систем (SEO) для покращення видимості у пошукових системах, що збільшить кількість відвідувачів і потенційних покупців. Використання інструментів електронного маркетингу, таких як контекстна

реклама, соціальні мережі, email-маркетинг та інші методи, допоможе залучити клієнтів та підвищити продажі. Створення контент-плану для регулярного оновлення блогу та новин на сайті сприятиме підвищенню інтересу користувачів та залученню нових відвідувачів.

Аналіз та тестування включають проведення тестування всіх функцій сайту для виявлення та виправлення можливих помилок перед запуском. Користувацьке тестування дозволить отримати зворотний зв'язок від потенційних клієнтів та внести відповідні зміни для покращення функціональності та зручності сайту.

Запуск та підтримка проекту передбачають проведення запуску сайту та моніторинг його роботи для виявлення та вирішення можливих проблем на початкових етапах. Регулярне оновлення контенту, системи безпеки та функціональних можливостей сайту забезпечить його актуальність та безпеку.

#### 1.4 Висновки до першого розділу

У першому розділі ми дослідили розвиток електронної комерції в Україні. Було виявлено, що за останні роки цей сектор демонструє стрімке зростання, що обумовлено низкою факторів, таких як збільшення проникнення інтернету, розвиток цифрової економіки та зростання популярності мобільних пристроїв. Зокрема, обсяг ринку e-commerce в Україні суттєво зріс завдяки пандемії COVID-19, яка стимулювала перехід до онлайн-покупок.

Ключові тенденції розвитку електронної комерції включають мобільну комерцію, яка стає все більш важливою складовою сучасного бізнесу, а також використання соціальних мереж для збільшення продажів. Дослідження показало, що мобільні гаманці і додатки стають основними інструментами для онлайн-платежів, а соціальні мережі значно підвищують залученість клієнтів та обізнаність про бренди.

Було проведено детальний аналіз конкурентних програмних продуктів для електронної комерції, таких як OpenCart, Magento, WooCommerce та Shopify. Кожна з цих платформ має свої переваги та недоліки, що дозволило визначити оптимальні технології для реалізації проекту квіткового онлайн-магазину. OpenCart був обраний як найбільш відповідна платформа завдяки своїй відкритості, функціональності та простоті використання.

Розглянуті сучасні методи та інструменти електронної комерції сприяють розвитку онлайн-бізнесу в Україні. Використання інноваційних технологій, таких як штучний інтелект для персоналізованого маркетингу, а також інтеграція з різними платіжними системами та соціальними мережами, дозволяють підвищити ефективність комерційної діяльності. Це забезпечує зручність для користувачів, покращує користувацький досвід та сприяє зростанню бізнесу.

## РОЗДІЛ 2.

### РОЗРОБКА ТА ВПРОВАДЖЕННЯ ОНЛАЙН-МАГАЗИНУ

#### 2.1 Планування та дизайн

На етапі планування та дизайну визначаються основні функціональні можливості сайту, створюються макети сторінок та інтерактивні прототипи. Визначення ключових функцій сайту, таких як пошук товарів, кошик, оформлення замовлення, реєстрація користувачів, є важливим аспектом цього етапу.

Перед початком проектування необхідно зібрати вимоги до системи та провести аналіз програмних продуктів-аналогів. Це включає визначення основних функцій сайту, потреб користувачів та кращих практик у галузі.

Основною метою проекту є створення квіткового онлайн-магазину, який забезпечить зручність користувачам при виборі, замовленні та доставці квітів. Сайт повинен мати інтуїтивно зрозумілий інтерфейс, швидке завантаження сторінок та адаптивний дизайн, щоб користувачі могли комфортно використовувати його як на комп'ютерах, так і на мобільних пристроях.

Для визначення конкурентних переваг та можливостей поліпшення, було проведено аналіз основних програмних продуктів-аналогів у Дніпрі.

Сайт Floristik.ua пропонує широкий асортимент квітів та букетів, різні способи оплати та доставки [19]. Інтерфейс сайту досить зручний, але можна покращити швидкість завантаження сторінок і зробити дизайн більш сучасним. Важливою перевагою цього сайту є можливість вибору доставки в різні міста України, а також можливість відправлення квітів за кордон. Сайт також має розділ акцій та спеціальних пропозицій, що дозволяє користувачам скористатися знижками та вигідними умовами покупки (див. рис. 2.1).



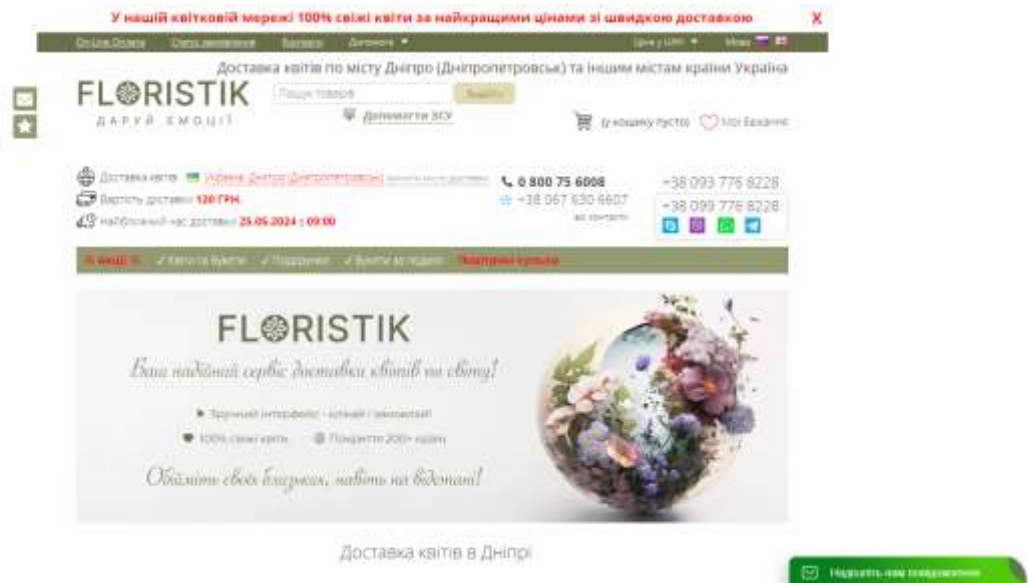


Рисунок 2.1 – Головна сторінка Floristik.ua

Сайт Buket24.dp.ua також пропонує різні варіанти квіткових композицій та подарунків [20]. Інтерфейс цього сайту зручний і зрозумілий, але є можливість оптимізації процесу замовлення та покращення мобільної версії. Однією з важливих особливостей сайту є наявність категорій товарів, які дозволяють швидко знайти потрібний продукт. Сайт також пропонує різні способи оплати, включаючи готівковий розрахунок, банківські картки та онлайн-платежі. Крім того, Buket24.dp.ua має інтеграцію з соціальними мережами, що дозволяє користувачам легко поділитися своїми враженнями про покупку (див. рис. 2.2).

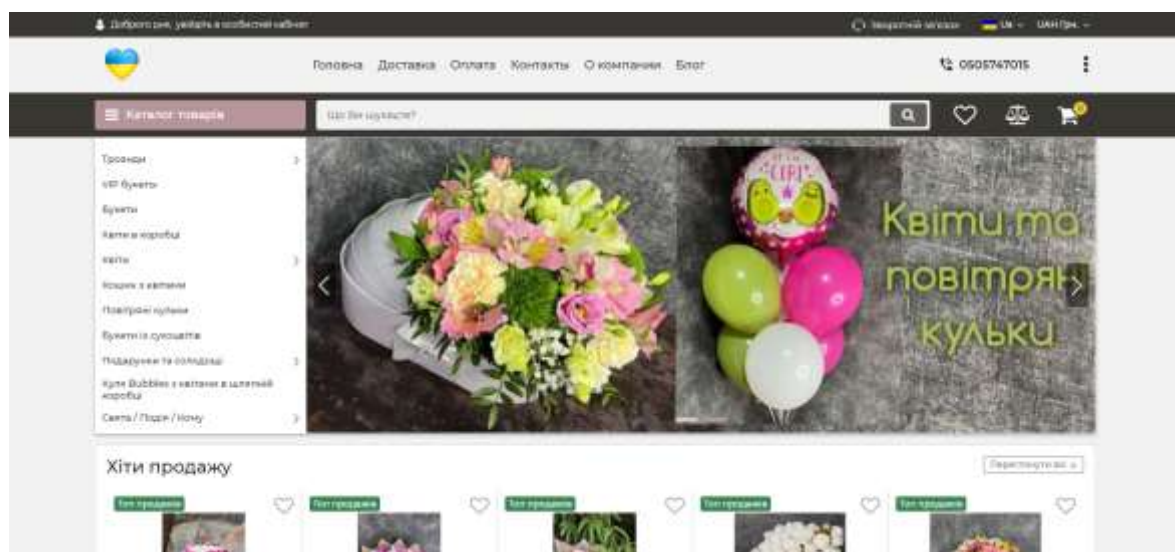


Рисунок 2.2 – Головна сторінка Buket24.dp.ua

Аналізуючи ці програмні продукти-аналогів, можна зробити висновки про основні переваги та недоліки кожного з них. Важливо враховувати ці аспекти при розробці власного онлайн-магазину, щоб забезпечити конкурентні переваги та задовольнити потреби користувачів. Для зручності було створено таблицю, що порівнює основні характеристики сайтів Floristik.ua та Buket24.dp.ua (див. табл. 2.1).

Таблиця 2.1

## Порівняльний аналіз сайтів Floristik.ua та Buket24.dp.ua

Критерії	Floristik.ua	Buket24.dp.ua
Асортимент товарів	Так	Так
Інтерфейс	Ні	Так
Швидкість завантаження	Ні	Ні
Зручність навігації	Так	Так
Способи оплати	Так	Ні
Доставка	Так	Так
Адаптивність	Ні	Ні
Акції та спеціальні пропозиції	Так	Так

На основі проведеного аналізу було визначено наступні функціональні можливості сайту. Важливо, щоб сайт мав можливість пошуку та фільтрації товарів за категоріями, ціною та рейтингом. Крім того, необхідно забезпечити докладні описи та зображення товарів, щоб користувачі могли отримати повну інформацію про продукцію. Кошик має бути зручним для додавання та перегляду вибраних товарів, а процес оформлення замовлення повинен включати можливість вибору способу доставки та оплати. Реєстрація та авторизація користувачів є обов'язковими функціями для забезпечення персоналізованого досвіду користувача. Також важливо, щоб користувачі могли

залишати відгуки та оцінювати товари. Інтеграція з платіжними системами забезпечить безпечні транзакції.

Створення інформаційної архітектури передбачає розробку структури сайту, яка включає основні розділи та підрозділи. Це забезпечує логічну та зручну навігацію для користувачів, сприяє легкому доступу до необхідної інформації та підвищує загальну ефективність використання сайту. Інформаційна архітектура визначає основні розділи сайту, такі як головна сторінка, каталог товарів, сторінка товару, кошик, оформлення замовлення та підтвердження замовлення [21].

Для ефективного планування архітектури сайту було проведено ретельний аналіз користувацьких сценаріїв та поведінки потенційних клієнтів. Цей аналіз дозволив визначити, які саме розділи та підрозділи є критично важливими для забезпечення зручності використання та задоволення потреб користувачів.

Основні розділи сайту включають:

- головна сторінка: містить основну інформацію про магазин, рекламні акції, нові надходження та популярні товари. Головна сторінка повинна бути інтуїтивно зрозумілою та привабливою для користувачів. Важливими елементами цієї сторінки є слайдер з акціями та новинками, що привертає увагу користувачів, блоки з популярними товарами та новинками, а також зручна навігація, яка дозволяє швидко переходити до інших розділів сайту.

- каталог товарів: включає всі доступні товари, організовані за категоріями, такими як квіти, букети, подарункові набори тощо. Кожна категорія повинна мати можливість фільтрації та сортування товарів за різними параметрами (ціна, популярність, рейтинг). Це забезпечує зручність пошуку та підвищує ефективність взаємодії з користувачами. Для кожної категорії варто створити підкатегорії, що додатково спрощують навігацію та дозволяють користувачам швидко знайти потрібний товар.

- сторінка товару: містить детальний опис товару, фотографії, ціну, відгуки користувачів та опції для додавання до кошика. Важливо забезпечити

можливість збільшення зображень товару для детального перегляду, що дозволяє користувачам краще ознайомитися з продуктом. Опис товару має бути інформативним і включати всі важливі характеристики та переваги. Відгуки користувачів допомагають потенційним покупцям прийняти рішення про покупку, тому їх також слід розміщувати на сторінці товару.

- кошик: дозволяє користувачам переглядати та редагувати товари, які вони додали до кошика. Кошик повинен включати можливість зміни кількості товарів, видалення товарів, перегляду загальної суми замовлення та продовження до оформлення замовлення. Інтерфейс кошика має бути простим та зручним, що забезпечує комфортне використання і мінімізує можливість помилок при оформленні замовлення.

- оформлення замовлення: процес має бути максимально простим та зручним. Він включає введення контактної інформації, вибір способу доставки та оплати, підтвердження замовлення. Важливо забезпечити користувачам можливість збереження інформації для швидкого оформлення замовлень у майбутньому. Для підвищення зручності можна впровадити можливість автозаповнення полів на основі раніше введеної інформації та підтримку різних методів оплати, включаючи онлайн-платежі, банківські картки та готівковий розрахунок.

- підтвердження замовлення: після успішного оформлення замовлення користувач повинен отримати підтвердження з усіма деталями замовлення. Це може бути сторінка підтвердження на сайті, а також електронний лист на вказану адресу електронної пошти. Важливо, щоб підтвердження містило всю необхідну інформацію, таку як список товарів, загальна сума, вибраний спосіб доставки та оплати, а також контактні дані для зв'язку з магазином у разі потреби.

Для створення інформаційної архітектури використовуються такі інструменти, як картографування сайту (sitemap) та прототипування. Картографування сайту допомагає визначити ієрархію сторінок і зв'язки між ними, що дозволяє побачити загальну структуру сайту та забезпечити логічну



Розробка макетів включає визначення розташування елементів, вибір кольорової гами, типографії та інших візуальних аспектів.

Прототип головної сторінки (див. рис. 2.4) забезпечує користувачів основною інформацією про магазин, контактні дані та можливість вибору букету. Сторінка має простий і зрозумілий інтерфейс, що дозволяє легко знайти необхідну інформацію.

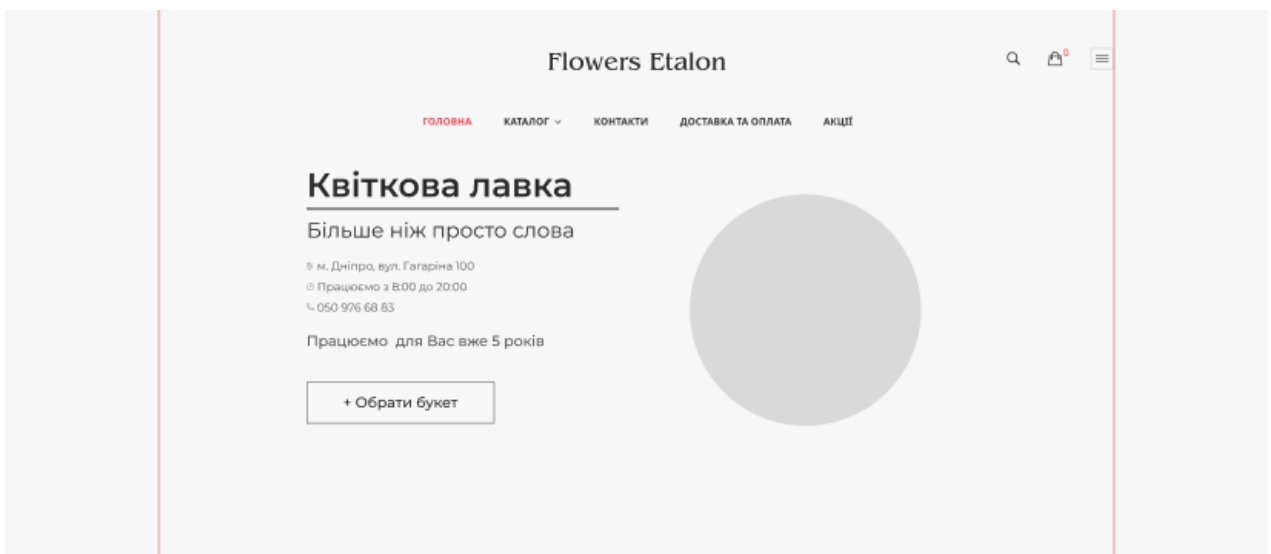


Рисунок 2.4 – Головна сторінка прототипу веб-сайту

Прототип каталогу товарів (див. рис. 2.5) представляє асортимент квітів і букетів, доступних для замовлення. На цій сторінці користувачі можуть переглядати товари за категоріями, оцінювати їх за рейтингами та переглядати ціни. Каталог товарів організований таким чином, щоб користувач міг легко знайти потрібний йому товар, використовуючи зручну навігацію та фільтри. Кожен товар у каталозі супроводжується якісним зображенням, детальним описом, ціною та інформацією про доступність.

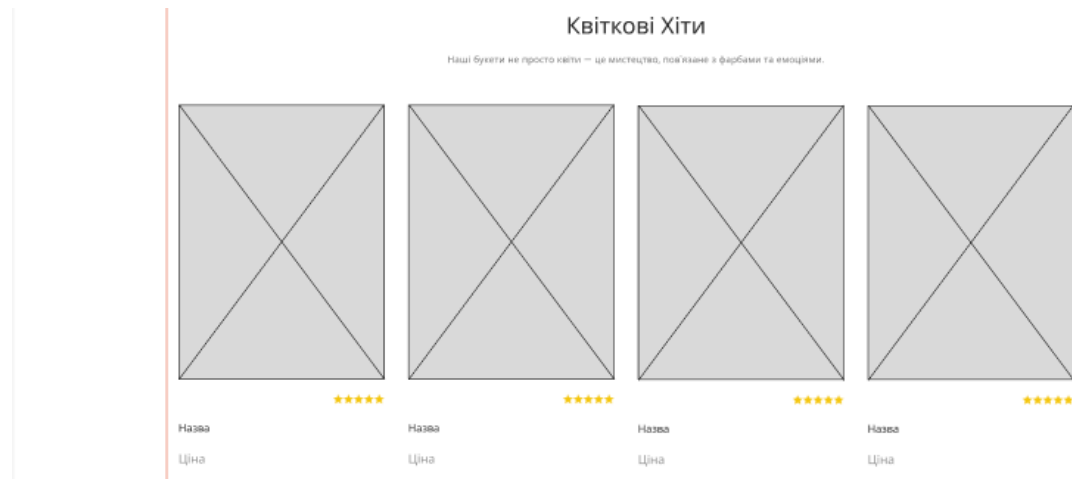


Рисунок 2.5 – Сторінка прототипу каталогу товарів

Прототип випадаючого списку каталогу із категоріями товарів (див. рис. 2.6) дозволяє користувачам швидко знаходити необхідні групи товарів, такі як квіти, букети, комплекти та декор. Це сприяє зручній навігації та полегшує процес пошуку товарів.

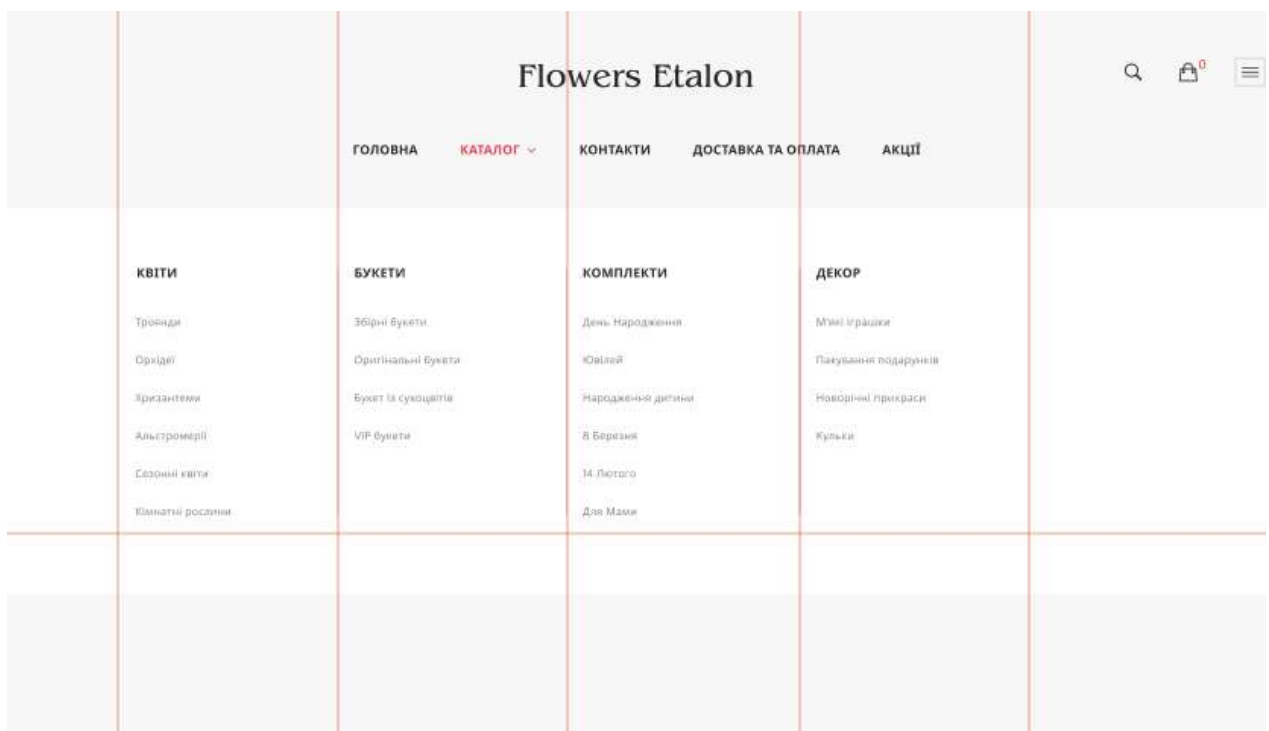


Рисунок 2.6 – Випадаючий список прототипу каталогу із категоріями товарів

Адаптивний дизайн є критично важливим для сучасних веб-сайтів, оскільки користувачі використовують різні пристрої для доступу до інтернету. Адаптивний дизайн дозволяє сайту автоматично підлаштовуватися під розміри екранів різних пристроїв, забезпечуючи комфортне переглядання та взаємодію з сайтом незалежно від того, чи користувач заходить з комп'ютера, планшета або смартфона. Це означає, що сайт завжди виглядає та функціонує оптимально, незалежно від роздільної здатності екрану чи його орієнтації [23]. До того ж, адаптивний дизайн сприяє покращенню SEO-показників сайту, оскільки пошукові системи віддають перевагу сайтам, які є зручними для мобільних користувачів. Це, в свою чергу, може підвищити видимість сайту в результатах пошуку та залучити більше відвідувачів.

На прикладах скріншотів головної сторінки видно, як сайт виглядає на різних пристроях (див. рис. 2.7 та рис. 2.8). Візуальні елементи та контент автоматично підлаштовуються під розміри екрану, забезпечуючи при цьому зручний доступ до всієї необхідної інформації. Такий підхід забезпечує не лише зручність у користуванні, але й значно покращує користувацький досвід, роблячи сайт доступним та функціональним на будь-якому пристрої.

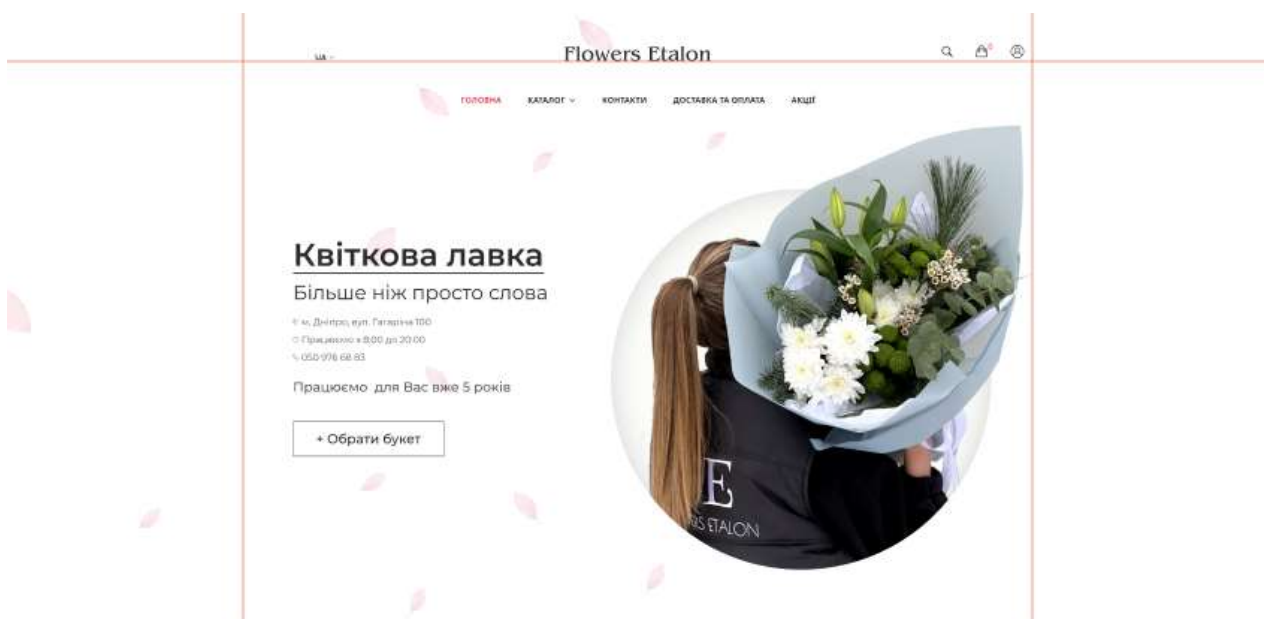


Рисунок 2.7 – Головна сторінка на десктопі





Рисунок 2.8 – Головна сторінка на мобільному пристрої

Адаптивний дизайн включає не лише зміну розміру елементів, але й їхнє розташування, щоб забезпечити зручність користувачів. Наприклад, на мобільних пристроях меню може бути перетворено на випадаюче для економії простору на екрані, а важливі елементи, такі як кнопки "Обрати букет", розміщуються на видному місці для зручного доступу. Крім того, зображення та відео можуть автоматично змінювати свій розмір або обрізатися, щоб відповідати екрану, забезпечуючи швидке завантаження та оптимальну якість перегляду.

Адаптивний дизайн також враховує такі аспекти, як розмір шрифтів та інтервалів між елементами, щоб текст залишався читабельним, а елементи інтерфейсу – легкодоступними навіть на найменших екранах. Важливими є також сенсорні елементи, такі як кнопки та посилання, які повинні бути достатньо великими для зручного натискання пальцем на мобільних пристроях. Це допомагає уникнути помилок під час взаємодії та покращує загальний користувацький досвід.

Такий підхід дозволяє забезпечити позитивний користувацький досвід незалежно від пристрою, що є важливим фактором для підвищення задоволеності клієнтів та збільшення конверсій у онлайн-магазині. Зручність використання та привабливий дизайн на всіх типах пристроїв сприяють тривалій взаємодії користувачів із сайтом, збільшенню часу перебування на ньому та, як наслідок, вищому рівню продажів. Адаптивний дизайн також позитивно впливає на SEO, оскільки пошукові системи віддають перевагу сайтам, які забезпечують хороший користувацький досвід на всіх пристроях.

## 2.2 Хостинг та SSL сертифікати

Надійність і продуктивність хостингу є критично важливими для успішної роботи онлайн-магазину. Вибір правильного хостинг-провайдера та налаштування серверних ресурсів забезпечують швидке завантаження сторінок, високу доступність сайту та захист даних користувачів.

Після ретельного аналізу було обрано хостинг-провайдера Hostiq.ua. Цей провайдер був обраний завдяки своїй надійності, високій продуктивності, масштабованості та підтримці. Важливими факторами при виборі також стали відгуки користувачів та рівень технічної підтримки, яка доступна 24/7. Hostiq.ua пропонує різноманітні тарифні плани, що дозволяють обрати оптимальне рішення для будь-якого бізнесу, з урахуванням його розміру та потреб. Важливо також зазначити, що Hostiq.ua забезпечує високу безпеку даних, що є ключовим аспектом для інтернет-магазинів, які обробляють особисту інформацію клієнтів та платіжні дані.

Налаштування хостингу включало кілька ключових етапів, які забезпечили коректну роботу сайту. Перш за все, було зареєстровано доменне ім'я (див. рис. 2.9), яке відповідає бренду магазину. Вибір доменного імені є важливим кроком, оскільки воно повинно бути легко запам'ятовуваним та відповідати тематиці магазину. Доменне ім'я є своєрідною візитівкою сайту, і правильний вибір може

значно вплинути на впізнаваність бренду та зручність користувачів у його запам'ятовуванні.

Мої домени	
Дата реєстрації:	24/01/2024
Домен:	<a href="https://flowersetalon.dp.ua">flowersetalon.dp.ua</a>
Захист WHOIS-інформації ?	Вимкнено
Сума першого платежу:	369.63грн.
Сума продовження:	399.60грн.
Дата наступної оплати:	24/01/2025
Період реєстрації:	1 рік/років
Статус:	Активний

Рисунок 2.9 – Доменне ім'я

Після реєстрації доменного імені було проведено налаштування серверних ресурсів. Цей етап включав оптимізацію налаштувань сервера для забезпечення максимальної продуктивності. Зокрема, було налаштовано кешування, що дозволяє зменшити навантаження на сервер та пришвидшити завантаження сторінок. Кешування є важливим аспектом оптимізації продуктивності, оскільки воно дозволяє зберігати часто використовувані дані у швидкодоступній пам'яті, що значно зменшує час завантаження сторінок.

Було також встановлено обмеження на кількість одночасних з'єднань, що дозволяє уникнути перевантаження сервера під час пікових навантажень. Це забезпечує стабільну роботу сайту навіть при великій кількості одночасних відвідувачів.

Встановлення SSL (Secure Sockets Layer) сертифіката забезпечує захищене з'єднання між веб-сервером та браузером користувача, шифруючи дані, що передаються. Це важливий крок для забезпечення безпеки даних користувачів і підвищення їхньої довіри до сайту. Використання SSL сертифіката гарантує, що будь-яка інформація, яка передається між сервером і клієнтом, захищена від

несанкціонованого доступу та може бути прочитана тільки призначеним отримувачем. Це особливо важливо для інтернет-магазинів, де передаються конфіденційні дані, такі як інформація про кредитні картки та персональні дані клієнтів.

Першим кроком у цьому процесі стало замовлення SSL сертифіката у постачальника. Після вибору та замовлення відповідного сертифіката, було проведено його встановлення на сервер. Для цього використовувалися інструменти, надані хостинг-провайдером, які значно спрощують процес інсталяції. Хостинг-провайдери зазвичай надають покрокові інструкції та технічну підтримку для забезпечення успішного встановлення сертифіката. Після встановлення сертифіката було проведено перевірку його роботи, щоб переконатися, що всі дані передаються по захищеному протоколу HTTPS. Перевірка включає тестування різних сторінок сайту та використання спеціальних інструментів для аналізу безпеки з'єднання.

SSL сертифікат працює за принципом шифрування даних між користувачем та сервером. Коли користувач відвідує сайт, захищений SSL сертифікатом, його браузер ініціює з'єднання з веб-сервером. Сервер відповідає, відправляючи свій SSL сертифікат, який містить публічний ключ. Браузер перевіряє сертифікат, щоб переконатися у його автентичності. Після успішної перевірки браузер використовує публічний ключ для шифрування симетричного сеансового ключа, який відправляється на сервер. Сервер розшифровує цей сеансовий ключ за допомогою свого приватного ключа. Відтепер всі дані, передані між браузером та сервером, шифруються за допомогою цього сеансового ключа [24]. Цей процес відбувається майже миттєво і непомітно для користувача, забезпечуючи безперервний захист даних.

Подано схему, що ілюструє процес роботи SSL сертифіката (див. рис. 2.10).

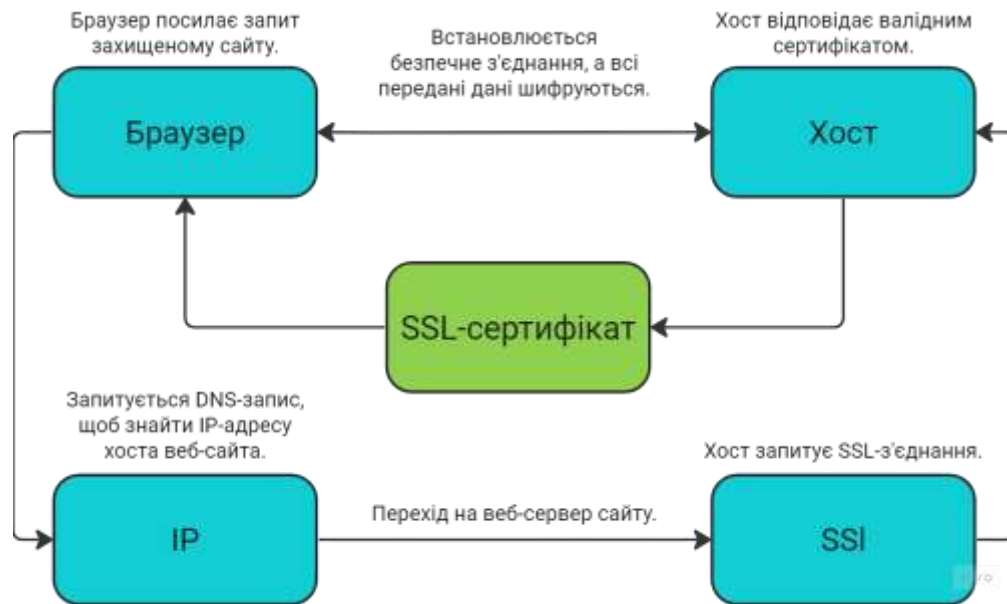


Рисунок 2.10 – Процес роботи SSL-сертифіката

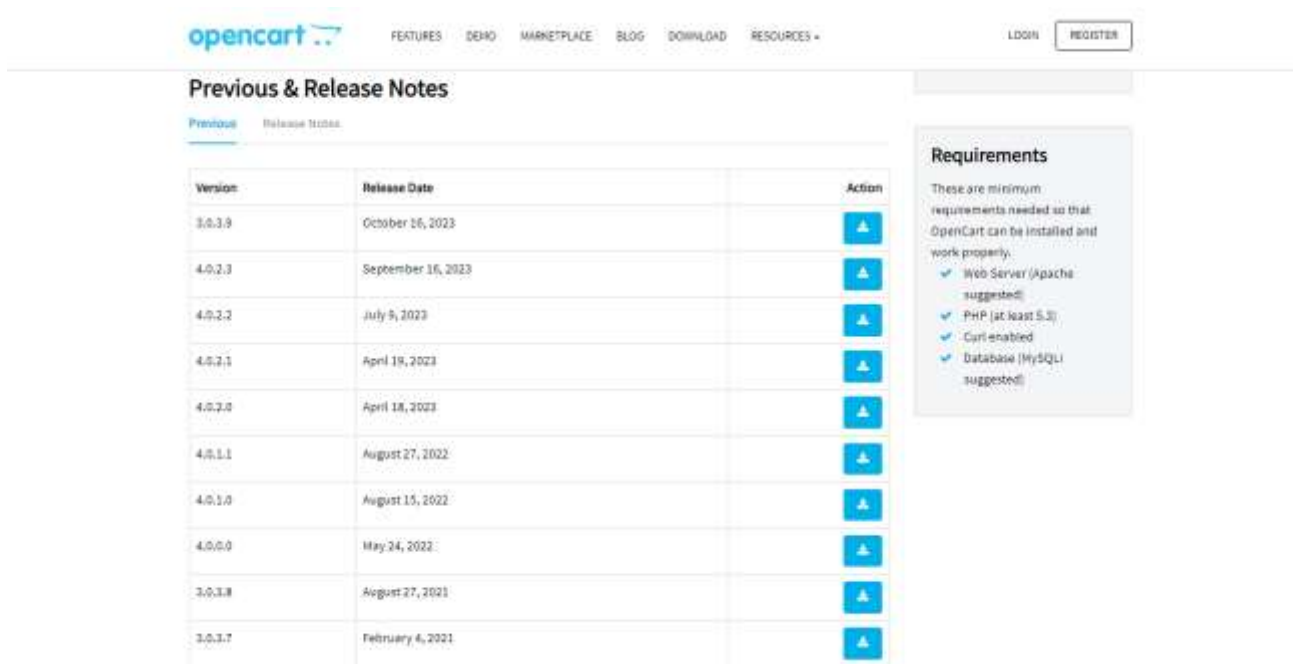
Встановлення SSL сертифіката має кілька важливих переваг. По-перше, це підвищує довіру користувачів до сайту, оскільки вони бачать, що їхні дані захищені. По-друге, це позитивно впливає на SEO, оскільки пошукові системи надають перевагу сайтам, які використовують HTTPS. Пошукові системи, такі як Google, враховують наявність SSL сертифіката при ранжуванні результатів пошуку, що може покращити видимість сайту та залучити більше відвідувачів. По-третє, це забезпечує відповідність вимогам законодавства щодо захисту персональних даних. У багатьох країнах законодавство вимагає використання захищених з'єднань для обробки конфіденційної інформації, і SSL сертифікат допомагає відповідати цим вимогам.

Завдяки належному вибору хостинг-провайдера, налаштуванню серверних ресурсів та встановленню SSL сертифіката, було забезпечено високу продуктивність та безпеку роботи онлайн-магазину. Ці кроки дозволяють надавати користувачам зручний та надійний сервіс, що сприяє їхній задоволеності та підвищенню рівня довіри до магазину. Крім того, забезпечення захищених з'єднань допомагає запобігти можливим атакам на дані користувачів, таким як перехоплення даних або фішинг. Це робить користування сайтом безпечнішим та сприяє довгостроковій лояльності клієнтів.

## 2.3 Підключення OpenCart та інтеграція бази даних

У цьому підрозділі ми детально розглянемо процес встановлення та налаштування OpenCart версії 3.0.3.7 для нашого проекту, що дозволить забезпечити функціональність, зручність використання та безпеку даних онлайн-магазину.

Першим кроком було завантаження останньої версії OpenCart 3.0.3.7 з офіційного сайту (див. рис. 2.11). Після цього архів з файлами OpenCart був завантажений на комп'ютер. Далі файли необхідно було завантажити на сервер через панель керування хостингом (cPanel). У розділі "Файловий менеджер" обрано кореневу директорію сайту (public\_html), де файли OpenCart були розпаковані.



The screenshot shows the OpenCart website's 'Previous & Release Notes' page. The page features a navigation menu at the top with links for FEATURES, DEMO, MARKETPLACE, BLOG, DOWNLOAD, and RESOURCES. There are also LOGIN and REGISTER buttons. The main content area is titled 'Previous & Release Notes' and includes a table of versions and a sidebar with system requirements.

Version	Release Date	Action
3.0.3.9	October 16, 2023	Download
4.0.2.3	September 16, 2023	Download
4.0.2.2	July 9, 2023	Download
4.0.2.1	April 19, 2023	Download
4.0.2.0	April 18, 2023	Download
4.0.1.1	August 27, 2022	Download
4.0.1.0	August 15, 2022	Download
4.0.0.0	May 24, 2022	Download
3.0.3.8	August 27, 2021	Download
3.0.3.7	February 6, 2021	Download

**Requirements**  
These are minimum requirements needed so that OpenCart can be installed and work properly.

- Web Server (Apache suggested)
- PHP (at least 5.3)
- curl-enabled
- Database (MySQL suggested)

Рисунок 2.11 – Офіційна сторінка OpenCart

Для забезпечення коректної роботи OpenCart необхідно створити базу даних. У cPanel було відкрито розділ "MySQL Databases", де створено нову базу даних, а також користувача бази даних з усіма правами доступу до цієї бази.

Важливо було вибрати надійний пароль для користувача бази даних, щоб забезпечити захист даних від несанкціонованого доступу.

Після завантаження файлів на сервер і створення бази даних, у браузері було введено URL-адресу сайту, де розташовані файли OpenCart. Це автоматично відкриває сторінку інсталятора OpenCart. Інсталятор перевіряє системні вимоги і пропонує ввести дані для підключення до бази даних, а також створити обліковий запис адміністратора. Після успішного заповнення всіх полів інсталяція завершена.

Далі потрібно налаштувати адміністративну панель. Ввівши дані адміністратора, отримуємо доступ до панелі керування магазином, де можна керувати товарами, категоріями, замовленнями, користувачами та іншими аспектами магазину. Панель керування OpenCart є досить зручною та інтуїтивною, що дозволяє швидко освоїтися та ефективно керувати всіма процесами магазину.

Для створення привабливого інтерфейсу користувача було вирішено розробити унікальну тему на основі стандартної теми OpenCart. Процес створення та налаштування теми включав кілька важливих кроків:

- макети сторінок були налаштовані для забезпечення зручності користувачів. Розташування модулів та блоків було оптимізовано таким чином, щоб інформація на сторінках була логічно структурована та легко доступна. Це включало розміщення товарів, категорій, інформаційних блоків та інших елементів інтерфейсу. Наприклад, головна сторінка була оформлена з урахуванням потреб користувачів, щоб вони могли швидко знайти потрібну інформацію та перейти до відповідних розділів.

- стилі теми були створені відповідно до бренду квіткового магазину. Було обрано кольорову гамму, шрифти та інші елементи стилю, які відповідали загальній концепції магазину та підкреслювали його індивідуальність. Було також додано графічні елементи, що підсилюють візуальне враження від сайту та роблять його привабливішим для відвідувачів. Важливо було забезпечити

гармонійне поєднання всіх елементів дизайну, щоб створити єдиний стиль та підкреслити унікальність магазину.

- тема була адаптована для забезпечення необхідної функціональності квіткового магазину. Це включало налаштування кошика, системи замовлень, пошуку товарів та фільтрації. Були також додані спеціальні модулі для відображення акцій, новинок та популярних товарів, що сприяло покращенню користувацького досвіду та збільшенню продажів. Кожен елемент інтерфейсу був продуманий таким чином, щоб максимально полегшити процес покупок для користувачів.

Після цього я розпочав роботу над головною сторінкою сайту [Додаток]. Використовуючи HTML, CSS та JavaScript, я створив унікальний дизайн, що включав у себе слайдер із зображеннями продуктів, рекламні банери та інші візуальні елементи. Кожен блок головної сторінки був ретельно продуманий, щоб забезпечити привабливий вигляд та зручність для користувачів. Наприклад, для слайдера я використав Owl Carousel, що дозволило створити динамічну та інтерактивну презентацію товарів. Слайдер є важливим елементом головної сторінки, оскільки він привертає увагу користувачів до найважливіших пропозицій та акцій магазину.

Наступним кроком стало створення каталогу товарів. Я створив ієрархічну структуру категорій та підкатегорій, що дозволило зручно організувати продукцію. Кожен товар містив детальний опис, характеристики, ціни та фотографії високої якості. Для завантаження зображень та інформації про товари я використав стандартні інструменти OpenCart, що дозволило автоматизувати цей процес та зекономити час. Важливо було забезпечити високу якість фотографій товарів, щоб користувачі могли детально ознайомитися з продукцією перед покупкою.

Особливу увагу було приділено процесу оформлення замовлення. Для цієї мети я інтегрував модуль Simple, який значно спрощує процес оформлення замовлення та робить його більш інтуїтивним для користувачів. Я налаштував



модуль таким чином, щоб мінімізувати кількість кроків та зробити процес максимально простим для користувача. Simple дозволяє клієнтам легко додавати та видаляти товари, переглядати загальну вартість замовлення та переходити до оплати. Крім того, я інтегрував кілька платіжних шлюзів, включаючи оплату банківськими картами та електронними гаманцями, а також налаштував методи доставки з інтеграцією популярних служб доставки. Це дозволяє користувачам обирати найзручніший спосіб оплати та доставки, що підвищує їхню задоволеність від покупок.

Інші сторінки сайту були створені за тим же принципом, з урахуванням функціональних вимог та дизайну. Кожен елемент сайту був ретельно продуманий та реалізований з метою забезпечити найкращий користувацький досвід та ефективність роботи магазину. Всі сторінки були оптимізовані для швидкого завантаження та зручного перегляду на різних пристроях, що забезпечує комфортне користування сайтом незалежно від типу пристрою, який використовує користувач.

### Вибір СУБД

Для нашого проекту було обрано MySQL - систему керування реляційними базами даних (РСУБД), яка широко використовується для веб-додатків. Це рішення базувалося на її надійності, швидкості, масштабованості, підтримці SQL та простоті використання. MySQL забезпечує високу продуктивність та стабільність, що робить її ідеальною для інтернет-магазинів. Важливою перевагою MySQL є можливість зберігати неструктуровані дані завдяки підтримці нереляційних моделей даних, що забезпечує гнучкість та можливість масштабування в майбутньому. Це дозволяє системі адаптуватися до змінних вимог бізнесу, розширювати функціональність та інтегрувати нові типи даних без значних змін у структурі бази даних.

Ретельне проектування бази даних є критичним етапом, що забезпечує цілісність, узгодженість та ефективність даних. Розроблена схема бази даних для продуктів інтернет-магазину складається з нормалізованих таблиць у третій

нормальній формі (ЗНФ), що мінімізує надлишковість та аномалії при оновленні. Це дозволяє уникнути дублювання даних та забезпечує їх консистентність.

Для забезпечення належного функціонування інтернет-магазину, необхідно було визначити основні таблиці, що зберігатимуть інформацію про товари. Ці таблиці включають дані про основні характеристики товарів, їх описи, зображення, опції, атрибути, зв'язки з категоріями, відгуки та інформацію про виробників. Нижче наведено опис основних таблиць, що використовуються в базі даних:

- `os_product` - зберігає основні відомості про продукти.
- `os_product_description` - описи продуктів різними мовами.
- `os_product_image` - зображення продуктів та їх сортування.
- `os_product_option`, `os_product_option_value` - опції продуктів.
- `os_product_attribute` - додаткові атрибути продуктів.
- `os_product_to_category` - зв'язки продуктів з категоріями.
- `os_review` - відгуки про продукти.
- `os_manufacturer` - інформація про виробників.

Первинні та зовнішні ключі використовуються для зв'язування таблиць, що забезпечує цілісність та узгодженість даних. Наприклад, `os_product` пов'язана з `os_product_description` через `product_id`. Індeksi створені для оптимізації запитів, що дозволяє значно підвищити продуктивність бази даних при великій кількості запитів. Типи даних полів ретельно підібрані відповідно до обмежень та вимог до даних. Наприклад, `varchar` використовується для рядкових значень змінної довжини, а `decimal` для цін з фіксованою кількістю десяткових розрядів, що забезпечує точність та коректність фінансових розрахунків.

Для полегшення розуміння структури бази даних наведено ER-діаграму (діаграму сутність-зв'язок), яка ілюструє взаємозв'язки між таблицями (див. рис. 2.12). Ця діаграма допомагає візуалізувати структуру даних та зв'язки між різними сутностями, що спрощує процес розробки та підтримки бази даних.

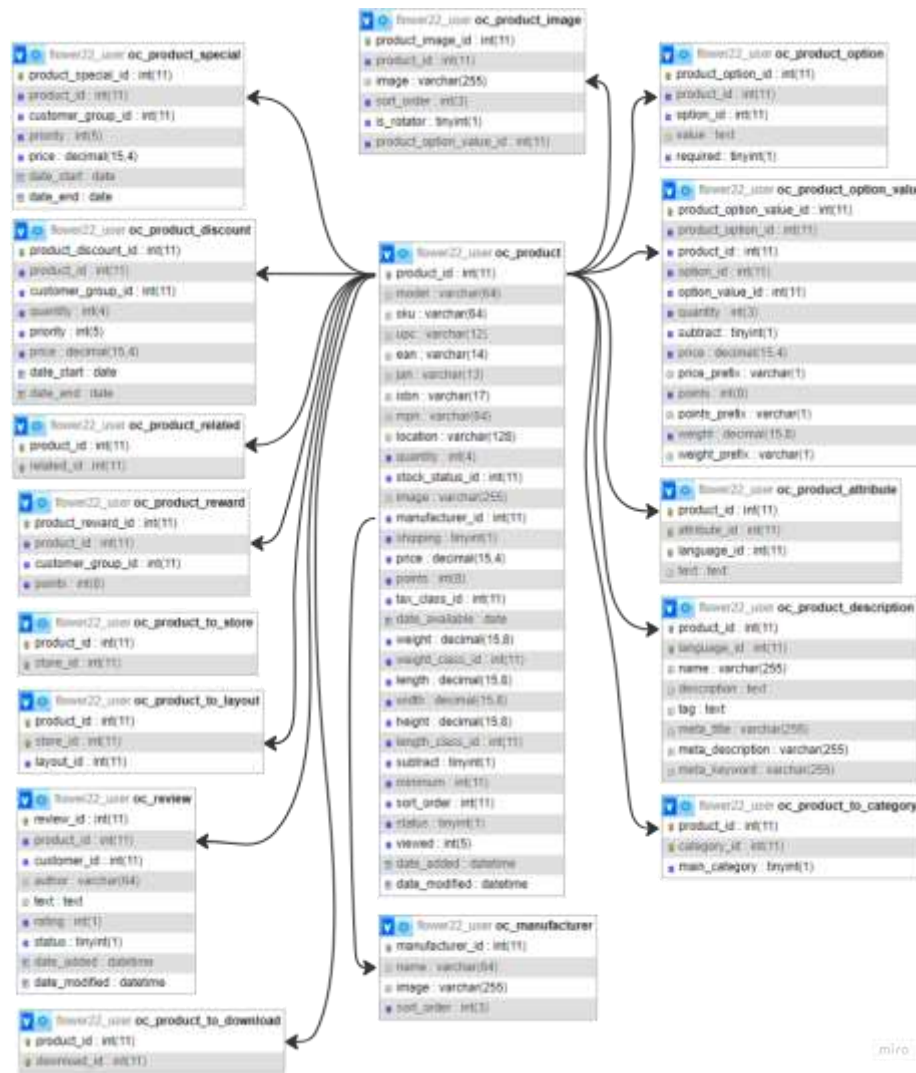


Рисунок 2.12 - ER-діаграма бази даних інтернет-магазину

Прикладами типових запитів є вибірка продуктів з категорії, додавання нового продукту, оновлення ціни та опису продукту.

Повна база даних також включає таблиці для керування замовленнями, клієнтами, налаштуваннями магазину та іншими аспектами електронної комерції.

## 2.4 Висновки до другого розділу

У другому розділі ми розглянули засоби розробки веб-додатку для підтримки роботи квіткового магазину. Основну увагу було приділено вибору та аналізу backend та frontend технологій, які забезпечують функціональність та

продуктивність веб-додатку. Було визначено, що для реалізації проекту оптимально використовувати JavaScript, HTML, CSS та PHP.

Backend технології, такі як PHP, забезпечують обробку даних на сервері, інтеграцію з базами даних та створення динамічних веб-сторінок. Використання PHP дозволяє реалізувати всі необхідні функції для ефективного управління інтернет-магазином, такі як каталог товарів, кошик покупок, система замовлень та обробка платежів. Крім того, PHP підтримується більшістю хостингових платформ, що забезпечує легку інтеграцію та розгортання веб-додатків.

Frontend технології, зокрема JavaScript, HTML та CSS, відповідають за створення інтерфейсу користувача та забезпечують динамічність веб-сторінок. JavaScript дозволяє створювати інтерактивні елементи, а HTML та CSS забезпечують структуру та візуальне оформлення контенту. Використання цих технологій сприяє створенню привабливого та зручного для користувачів інтерфейсу.

Було також обговорено важливість вибору правильного хостинг-провайдера та налаштування серверних ресурсів для забезпечення стабільної роботи сайту. Надійний хостинг, налаштування кешування та автоматичні резервні копії є критично важливими для забезпечення високої продуктивності та безпеки даних користувачів. Встановлення SSL сертифікатів забезпечує захищене з'єднання між сервером та браузером користувача, що підвищує довіру клієнтів до онлайн-магазину.

## РОЗДІЛ 3. УПРАВЛІННЯ ТА ОПТИМІЗАЦІЯ ОНЛАЙН-МАГАЗИНУ

### 3.1 Адміністративна панель OpenCart

Адміністративна панель OpenCart є потужним інструментом для управління онлайн-магазином. Вона дозволяє адміністратору керувати товарами, замовленнями, користувачами, налаштуваннями магазину та іншими аспектами роботи магазину. Розглянемо основні розділи адміністративної панелі та їх функціональні можливості.

Адміністративна панель складається з кількох основних розділів, які дозволяють адміністратору легко керувати всіма аспектами онлайн-магазину. Один з важливих розділів - це управління категоріями товарів. У цьому розділі адміністратор може додавати, редагувати та видаляти категорії товарів. Це включає створення нових категорій, визначення їхнього порядку сортування та зв'язування з відповідними товарами. Крім того, адміністратор може налаштовувати ієрархію категорій, створюючи підкатегорії, що допомагає у структуризації товарів і покращує навігацію для користувачів. На рисунку 3.1 показано, як виглядає розділ керування категоріями.

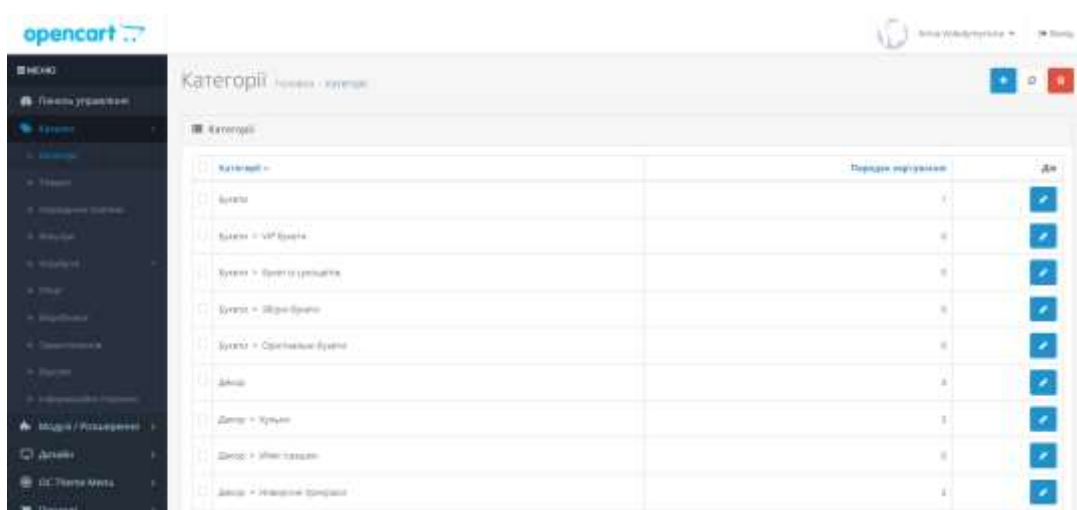


Рисунок 3.1 – Адміністративна панель OpenCart: управління категоріями

У розділі товарів адміністратор може бачити список усіх товарів, їхні зображення, назви, моделі, ціни, кількість та статуси. Це включає завантаження зображень товарів, додавання описів, встановлення цін та налаштування опцій товарів. Також у цьому розділі можна налаштовувати SEO-мета-теги для кожного товару, що покращує видимість товарів у пошукових системах. Адміністратор має можливість створювати акційні пропозиції та знижки, що відображаються на сторінках товарів, стимулюючи продажі. На рисунку 3.2 зображено, як виглядає розділ керування товарами.

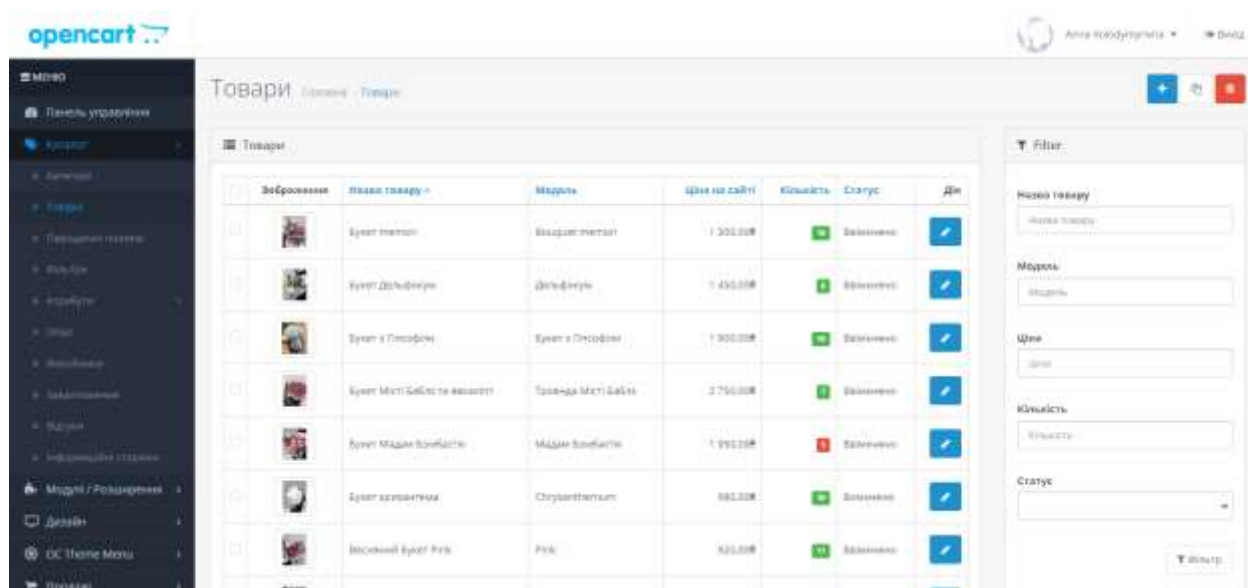


Рисунок 3.2 – Адміністративна панель OpenCart: управління товарами

Розділ керування клієнтами дозволяє адміністратору керувати обліковими записами користувачів, налаштовувати права доступу та ролі. Це включає можливість додавання нових користувачів, редагування інформації про існуючих користувачів та видалення облікових записів, які більше не потрібні. Крім того, адміністратор може аналізувати активність клієнтів, відстежуючи їхні замовлення, перегляди товарів та іншу взаємодію з сайтом. На рисунку 3.3 показано, як виглядає розділ керування клієнтами.

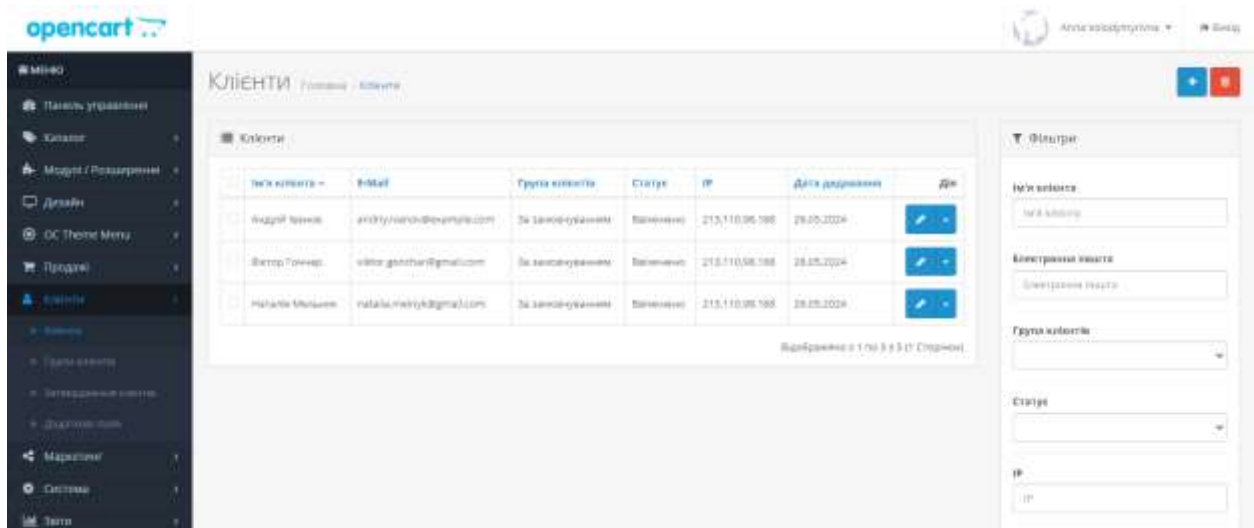


Рисунок 3.3 – Адміністративна панель OpenCart: управління клієнтами

Головна панель адміністратора надає зручний огляд всіх важливих показників і операцій магазину. Тут можна переглянути кількість замовлень, продажі, кількість клієнтів, а також людей, які зараз перебувають онлайн. На головній панелі також відображаються карти продажів та аналітика, що дозволяє адміністратору швидко оцінити стан магазину та приймати обґрунтовані рішення. Додатково на головній панелі можуть відображатися сповіщення про нові замовлення, повідомлення від клієнтів та інші важливі події. Це забезпечує оперативне реагування на зміни та події у магазині (див. рис. 3.4).

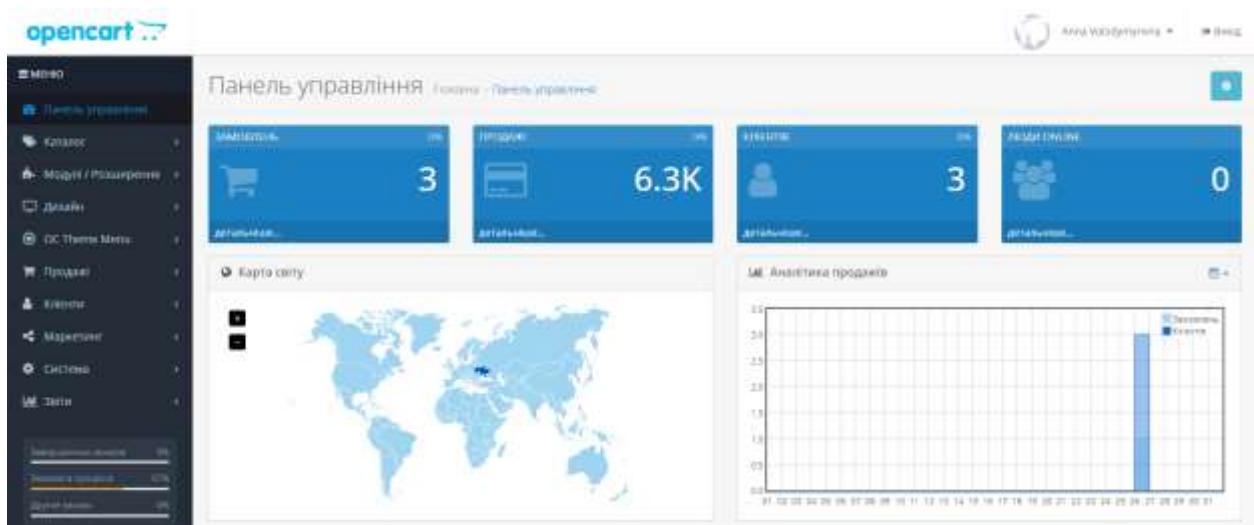


Рисунок 3.4 – Адміністративна панель OpenCart: головна панель

Адміністративна панель OpenCart надає зручні інструменти для обробки замовлень. Адміністратор може переглядати всі замовлення, змінювати їх статус, додавати коментарі та генерувати необхідні документи. Основні кроки для обробки замовлень включають відображення списку всіх замовлень з фільтрацією за статусом, датою та іншими параметрами, оновлення статусу замовлення (наприклад, "в обробці", "відправлено", "доставлено") та створення рахунків та накладних для замовлень. Крім того, адміністратор може налаштовувати автоматичні повідомлення для клієнтів про зміну статусу їхніх замовлень, що підвищує рівень обслуговування та задоволеності клієнтів.

Адміністратор також може керувати обліковими записами користувачів, налаштовувати права доступу та ролі. Основні кроки для управління користувачами включають створення нового облікового запису з необхідною інформацією, оновлення даних користувача, зміна прав доступу та ролей, а також видалення облікових записів, які більше не потрібні. Важливим аспектом є забезпечення безпеки даних користувачів, тому адміністратор може встановлювати складні паролі та двофакторну аутентифікацію для додаткового захисту облікових записів.

Налаштування магазину дозволяє адміністратору змінювати основні параметри магазину, такі як валюта, мова, платіжні та доставкові модулі. Основні налаштування магазину включають вибір основної валюти та мови магазину, налаштування методів оплати, таких як банківські картки, електронні гаманці та інші платіжні системи, налаштування методів доставки, включаючи кур'єрську доставку, самовивіз та поштові служби, а також вибір та налаштування теми магазину, налаштування макетів сторінок для забезпечення зручного та привабливого інтерфейсу. Адміністратор може також налаштовувати параметри безпеки, такі як SSL-сертифікати та захист від DDoS-атак.

Звіти є важливим інструментом для аналізу ефективності роботи магазину. Адміністратор може генерувати різні звіти про продажі, замовлення,



користувачів та товари. Основні типи звітів включають аналіз продажів за певний період часу, відстеження популярних товарів та категорій, відстеження кількості та статусів замовлень, аналіз ефективності обробки замовлень, аналіз активності користувачів, відстеження нових реєстрацій та активних користувачів, а також аналіз популярності товарів, відстеження залишків на складі та інших важливих показників. Адміністратор може використовувати ці звіти для прийняття обґрунтованих рішень щодо маркетингових стратегій, оптимізації асортименту товарів та покращення обслуговування клієнтів.

Таким чином, адміністративна панель OpenCart надає широкий спектр можливостей для ефективного управління онлайн-магазином. Вона забезпечує зручність та гнучкість у керуванні товарами, замовленнями, користувачами та налаштуваннями магазину, що дозволяє оптимізувати роботу магазину та забезпечити високий рівень обслуговування клієнтів. Завдяки широкому функціоналу, адміністратор може легко адаптувати магазин під специфічні потреби бізнесу, забезпечуючи стабільний розвиток та конкурентоспроможність на ринку електронної комерції.

### 3.2 Опис головних сторінок сайту

У цьому розділі розглянемо головні сторінки сайту, їхню структуру та функціональність. Основні сторінки включають головну сторінку, каталог товарів, сторінку товару, кошик, оформлення замовлення, контактну сторінку та сторінку акцій.

Головна сторінка є першою точкою контакту для відвідувачів сайту. Вона створена з метою привернути увагу клієнтів до основних пропозицій та новинок магазину, а також забезпечити зручний доступ до всіх розділів сайту (див. рис. 3.5). Основні елементи головної сторінки включають:

- слайдер з основними пропозиціями та новинками магазину: цей елемент дозволяє відвідувачам одразу побачити актуальні акції та спеціальні пропозиції.

Слайдер автоматично змінює зображення, привертаючи увагу до різних товарів і акцій.

- категорії товарів: цей блок допомагає швидко перейти до потрібної групи продуктів. Клієнти можуть легко знайти потрібні товари, обираючи відповідну категорію з меню, що розміщене у верхній частині сторінки.

- популярні товари або нові товари: цей розділ привертає увагу клієнтів до найпопулярніших або нових надходжень у магазині. Це сприяє збільшенню продажів та задоволеності клієнтів, надаючи їм актуальну інформацію про новинки та бестселери.

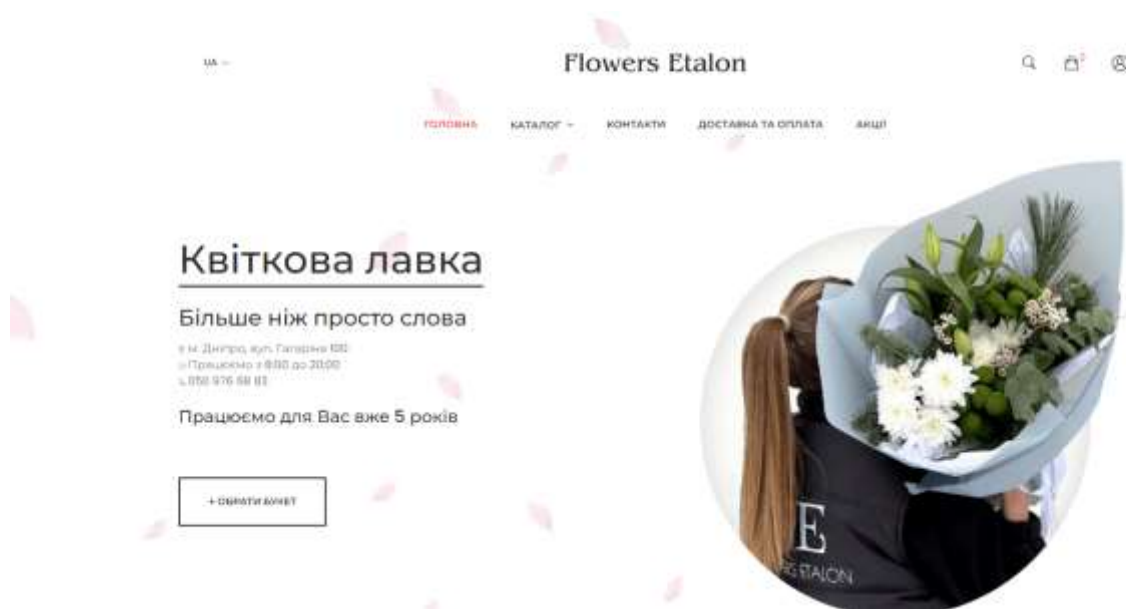


Рисунок 3.5 – Головна сторінка магазину

Каталог товарів є основним розділом сайту, де відвідувачі можуть переглядати всі доступні продукти магазину (див. рис. 3.6). Структура каталогу дозволяє сортувати товари за різними критеріями (ціна, популярність, новинки) та фільтрувати їх за категоріями. Кожна категорія має підкатегорії для зручної навігації, що забезпечує легкість пошуку потрібних товарів. Клієнти можуть швидко знаходити продукти, які їх цікавлять, використовуючи зручні фільтри та опції сортування.

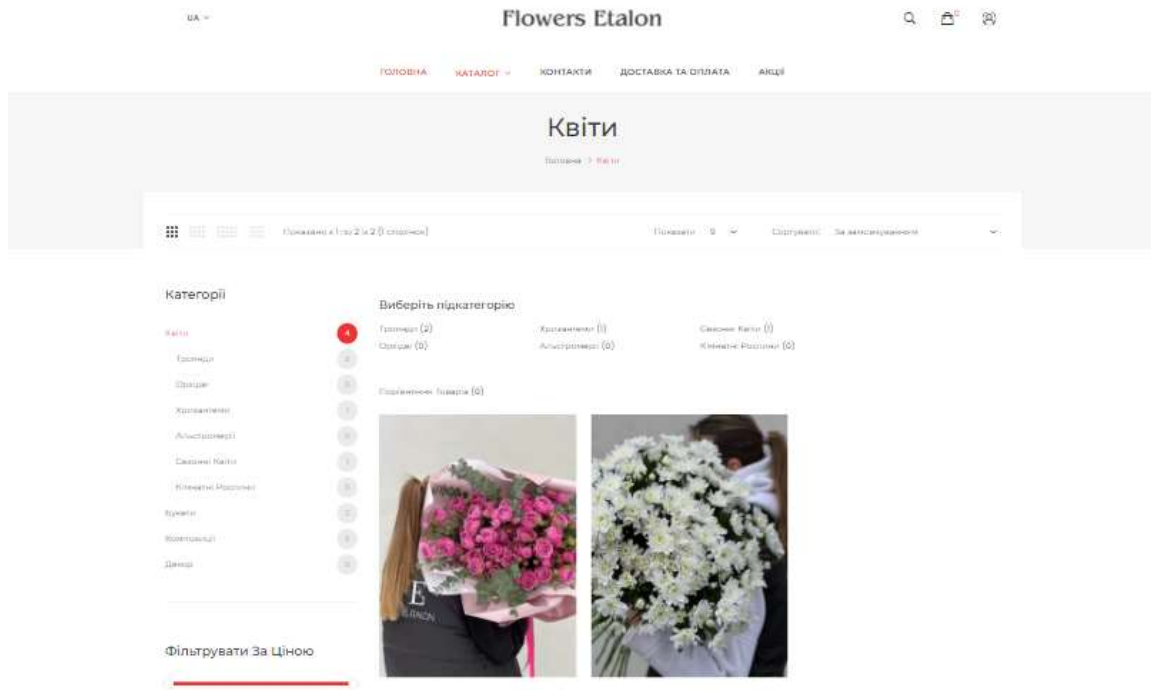


Рисунок 3.6 – Каталог товарів

Сторінка товару містить детальну інформацію про конкретний продукт. На прикладі сторінки товару "Букет Мадам Бомбастік" (див. рис. 3.7) можна побачити, що вона включає такі елементи:

- назва товару: відображає назву продукту, що дозволяє користувачам швидко ідентифікувати товар.
- ціна товару: показує вартість продукту, що дозволяє клієнтам оцінити витрати.
- опис товару: надає детальну інформацію про букет, включаючи його склад, особливості та можливі варіанти використання.
- зображення товару: висока якість фотографій допомагає клієнтам краще уявити товар.
- опції товару: якщо доступні різні варіанти (наприклад, розмір або колір), вони відображаються на сторінці.
- кнопка "Додати в кошик": дозволяє клієнтам швидко додати товар до свого замовлення.

- відгуки клієнтів: рейтинг і коментарі від інших покупців допомагають прийняти рішення про покупку.

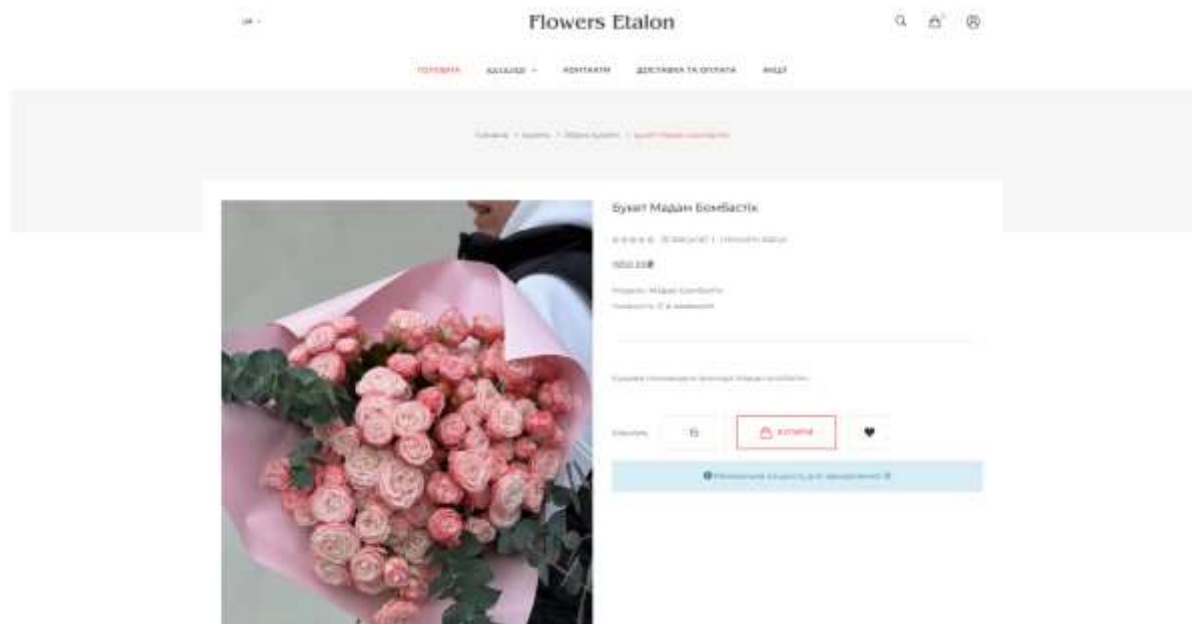


Рисунок 3.7 – Сторінка товару

Кошик є однією з найважливіших частин інтернет-магазину, оскільки він служить місцем, де клієнти можуть переглядати всі товари, які вони додали до свого замовлення під час шопінгу. Тут клієнти мають можливість не лише бачити перелік товарів, а й редагувати кількість кожного товару або видаляти ті, які більше не потрібні.

У кошику також відображається загальна сума замовлення, що є важливою інформацією для клієнта перед остаточним підтвердженням покупки. Це дозволяє користувачам легко контролювати свої витрати та, за потреби, змінювати склад замовлення, щоб відповідати своєму бюджету.

Кошик повинен бути інтуїтивно зрозумілим і зручним для користувачів, забезпечуючи їм простоту управління своїми покупками. Це включає в себе чітке відображення всіх важливих елементів, таких як назви товарів, їхні зображення, кількість, ціна за одиницю та загальна сума.

Зручність користування кошиком є ключовим фактором для забезпечення позитивного досвіду покупця, оскільки останній етап перед оформленням замовлення є вирішальним моментом у процесі покупки. Якщо клієнти зустрінуть труднощі або відчують незручності під час використання кошика, це може призвести до того, що вони залишать сайт без завершення покупки.

Таким чином, добре спроектований кошик сприяє підвищенню задоволеності клієнтів і збільшенню кількості успішно завершених транзакцій, що, в свою чергу, позитивно впливає на загальний успіх інтернет-магазину (див. рис. 3.8).

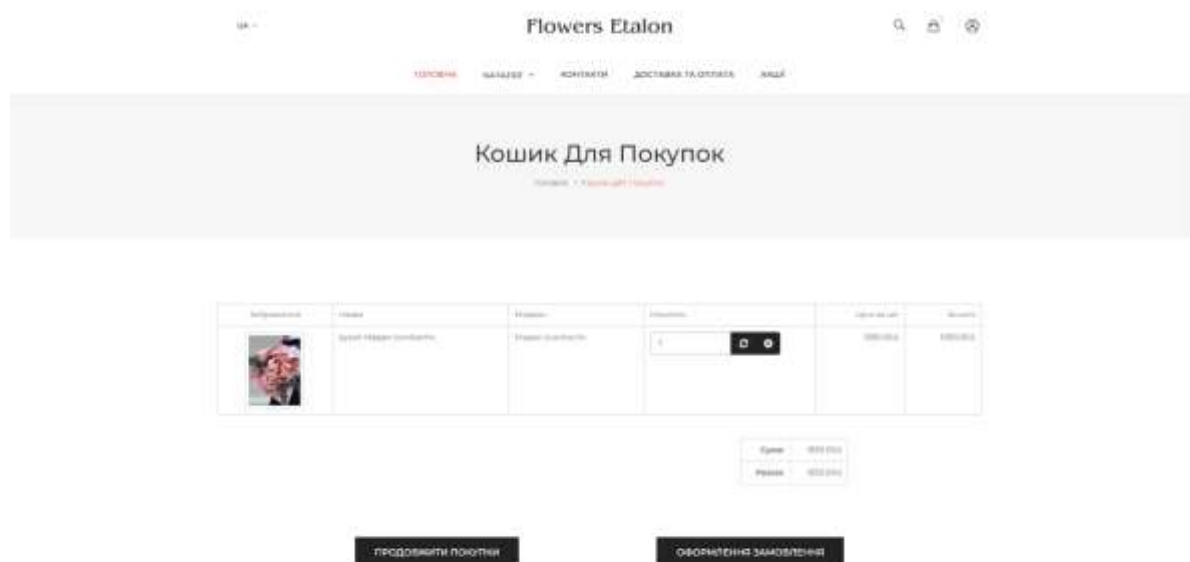


Рисунок 3.8 – Кошик

Сторінка оформлення замовлення є заключним етапом процесу покупки (див. рис. 3.9). Вона включає кілька важливих елементів:

- форма для заповнення контактної інформації: клієнт вводить своє ім'я, телефон та електронну пошту, що дозволяє магазину зв'язатися з ним у разі необхідності.

- адреса доставки: клієнт вказує адресу, на яку має бути доставлене замовлення. Це забезпечує точність та своєчасність доставки.
- способи доставки: клієнт обирає зручний для нього спосіб доставки, такий як кур'єрська доставка або самовивіз. Кожен спосіб має свої особливості та вартість.
- способи оплати: клієнт може обрати зручний для себе метод оплати, включаючи оплату банківською картою, готівкою при отриманні або через електронні платіжні системи.
- підтвердження замовлення: після заповнення всіх необхідних полів клієнт підтверджує замовлення, натискаючи відповідну кнопку. Це запускає процес обробки замовлення магазином.

На цій сторінці важливо забезпечити простоту та зрозумілість процесу для користувача. Кожен крок оформлення замовлення повинен бути інтуїтивно зрозумілим, щоб зменшити можливість помилок та забезпечити позитивний користувацький досвід.

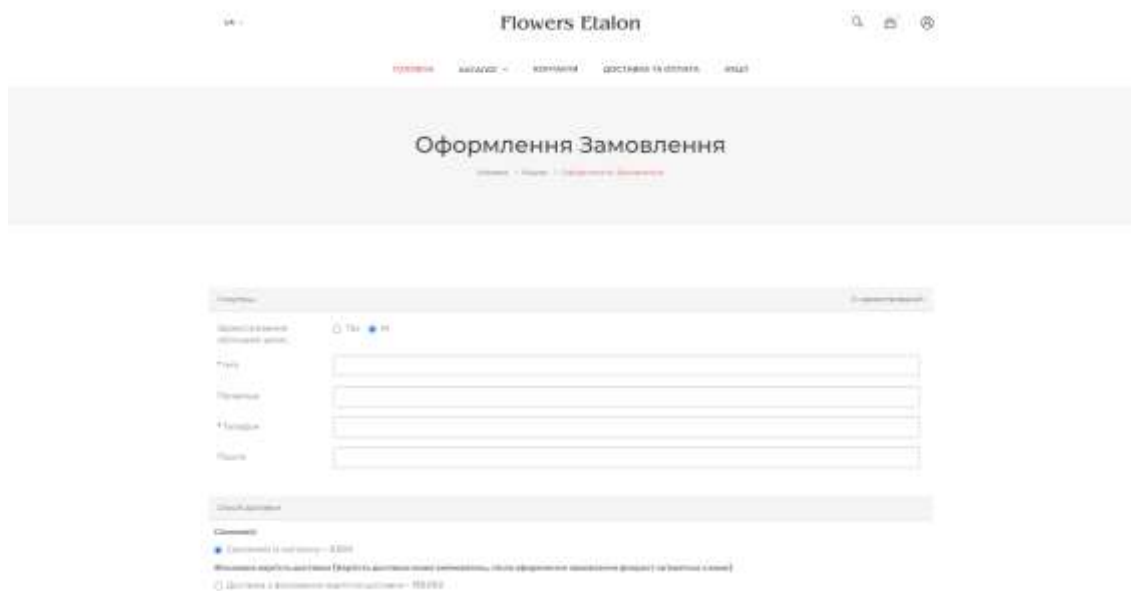


Рисунок 3.9 – Оформлення замовлення

Контактна сторінка містить інформацію про те, як зв'язатися з магазином (див. рис. 3.10). Вона включає адреси, телефони, електронну пошту, а також

форму для зворотного зв'язку. Це дозволяє клієнтам швидко знайти необхідну контактну інформацію та зв'язатися з підтримкою магазину. Наявність зручної контактної сторінки допомагає покращити обслуговування клієнтів та швидко вирішувати їхні питання або проблеми.

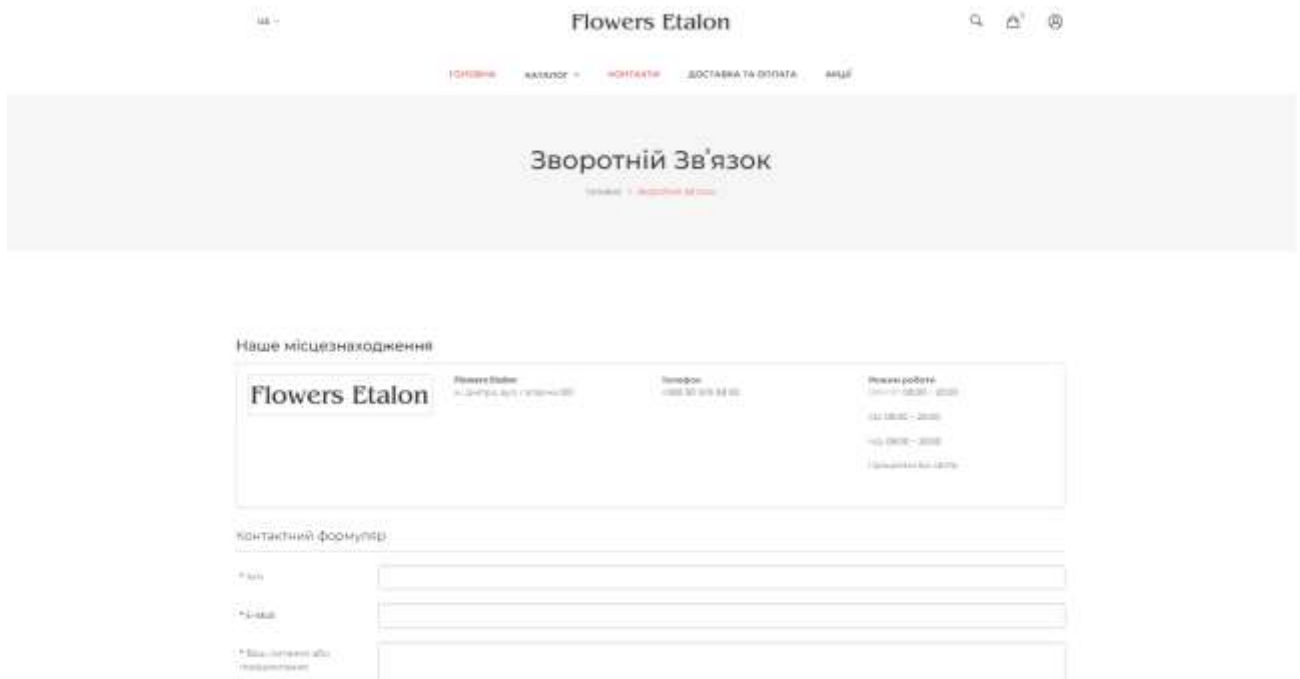


Рисунок 3.10 – Контактна сторінка

Сторінка акцій є важливим елементом інтернет-магазину, оскільки вона привертає увагу користувачів до спеціальних пропозицій та знижок. На цій сторінці відвідувачі можуть знайти акційні товари та скористатися вигідними пропозиціями, що сприяє підвищенню рівня продажів та залученню нових клієнтів.

Ця сторінка структурована таким чином, щоб відвідувачі могли легко знайти необхідну інформацію та зробити покупку. Всі акційні товари представлені в зручному форматі з чіткими зображеннями, описами та цінами. Користувачі можуть переглянути всі доступні знижки та спеціальні пропозиції, а також скористатися фільтрами для швидкого пошуку товарів за категоріями, ціною або іншими параметрами.

Загалом, сторінка акцій є ключовим інструментом для залучення клієнтів і збільшення обсягів продажів. Вона допомагає інтернет-магазину виділитися серед конкурентів, пропонуючи вигідні умови для покупок і забезпечуючи приємний користувацький досвід (див. рис. 3.11).

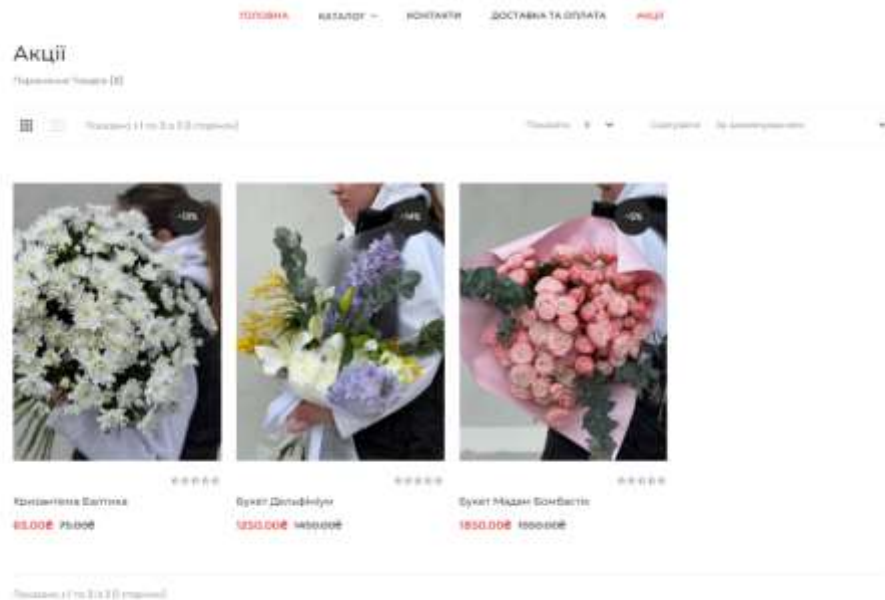


Рисунок 3.11 – Сторінка акцій

Основні елементи сторінки акцій включають:

- зображення товарів зі знижками: яскраві та привабливі фотографії акційних товарів допомагають виділити їх серед інших продуктів. Ці зображення розташовані таким чином, щоб користувач міг одразу побачити вигідні пропозиції.

- назва та опис товару: відображають основні характеристики та переваги товару, що перебуває на акції. Короткий і зрозумілий опис допомагає користувачам швидко зрозуміти, що пропонується.

- стара та нова ціна: показують розмір знижки, що дозволяє клієнтам бачити вигоду від покупки. Знижена ціна виділяється, щоб привернути увагу користувачів.



- відсоток знижки: вказує на розмір знижки, що додатково стимулює покупців скористатися пропозицією. Відсоток знижки розташований поруч із зображенням товару, що робить його добре помітним.

### 3.3 SEO налаштування та оптимізація продуктивності

SEO (Search Engine Optimization) є критично важливим для успішного функціонування будь-якого інтернет-магазину. Оптимізація сайту для пошукових систем допомагає підвищити його видимість у результатах пошуку, що сприяє залученню більшої кількості потенційних клієнтів і, як наслідок, збільшенню продажів. Висока позиція в результатах пошуку дозволяє вашому інтернет-магазину бути помітнішим серед конкурентів, що позитивно впливає на загальний обсяг трафіку та конверсії.

В процесі налаштування SEO для сайту було виконано комплексний аналіз ключових слів. Це включало визначення ключових слів, за якими потенційні клієнти можуть шукати товари, дослідження ринку, аналіз конкурентів та використання спеціалізованих інструментів для підбору ключових слів. Інструменти, такі як Google Keyword Planner, SEMrush і Ahrefs, допомогли ідентифікувати найбільш релевантні та ефективні ключові слова для нашого інтернет-магазину. Важливою частиною цього процесу було визначення довгохвостих ключових слів, які є менш конкурентними, але мають високу конверсійну здатність.

Контент сайту був оптимізований з урахуванням цих ключових слів. Було важливо використовувати визначені ключові слова у заголовках, описах товарів, мета-тегах та інших елементах сторінок. Особлива увага приділялася унікальності контенту та його релевантності для користувачів [25]. Наприклад, створювалися унікальні та інформативні мета-заголовки і мета-описи для кожної сторінки сайту (див. рис. 3.12). Мета-заголовки мають бути короткими та влучними, вони повинні містити основні ключові слова та відображати суть

сторінки. Мета-описи мають приваблювати користувачів і стимулювати їх перейти на сайт, тому вони повинні бути цікавими та інформативними. Окрім того, в процесі оптимізації було враховано швидкість завантаження сторінок та мобільну адаптивність, що є важливими факторами ранжування в пошукових системах.

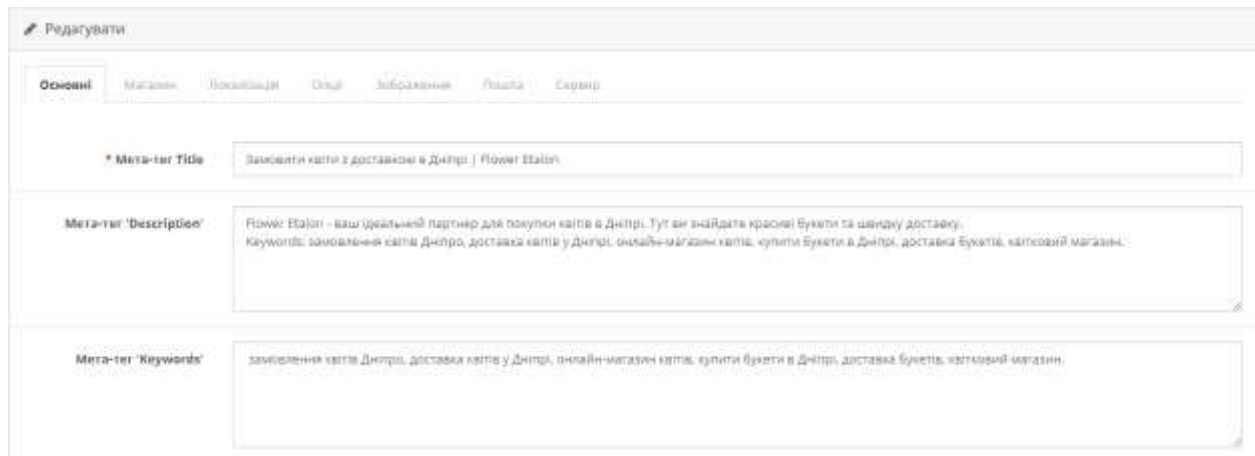


Рисунок 3.12 – Мета-теги в адміністративній панелі OpenCart

Також було налаштовано зрозумілі та короткі URL-адреси, що містять ключові слова. Це полегшує пошуковим системам індексацію сторінок та покращує їхню видимість. Крім того, створювалися зв'язки між різними сторінками сайту для полегшення навігації та покращення індексації пошуковими системами. Внутрішні посилання сприяють кращому розумінню структури сайту пошуковими системами та підвищують вагу важливих сторінок.

Окрема увага приділялася мобільній оптимізації. Забезпечувалося коректне відображення сайту на мобільних пристроях, що включало використання адаптивного дизайну та перевірку швидкості завантаження на різних типах пристроїв. Швидкість завантаження є одним з ключових факторів ранжування, тому оптимізація зображень, використання кешування та мінімізація коду є важливими кроками для покращення продуктивності сайту.

Особлива увага приділялася створенню та оптимізації файлу Sitemap, який є важливим елементом для покращення індексації сайту пошуковими системами.

Sitemap допомагає пошуковим системам швидше та точніше індексувати всі сторінки сайту, що сприяє їх кращій видимості в результатах пошуку. Це досягається за допомогою генерації XML-файлу, який містить перелік усіх сторінок сайту, а також додаткову інформацію про їх пріоритетність та частоту оновлення.

Важливим елементом SEO налаштувань є також аналіз та моніторинг ефективності. Для цього використовуються різні інструменти, серед яких Google Search Console [26]. Цей інструмент дозволяє відстежувати показники відвідуваності, кількість кліків, показів, середню позицію та середній CTR (див. рис. 3.13). Аналіз цих даних допомагає виявляти слабкі місця та можливості для покращення, а також оцінювати ефективність проведених SEO заходів. Наприклад, низький CTR може вказувати на те, що мета-описи потребують покращення, тоді як зниження позицій у пошукових системах може сигналізувати про проблеми з контентом або технічними аспектами сайту.

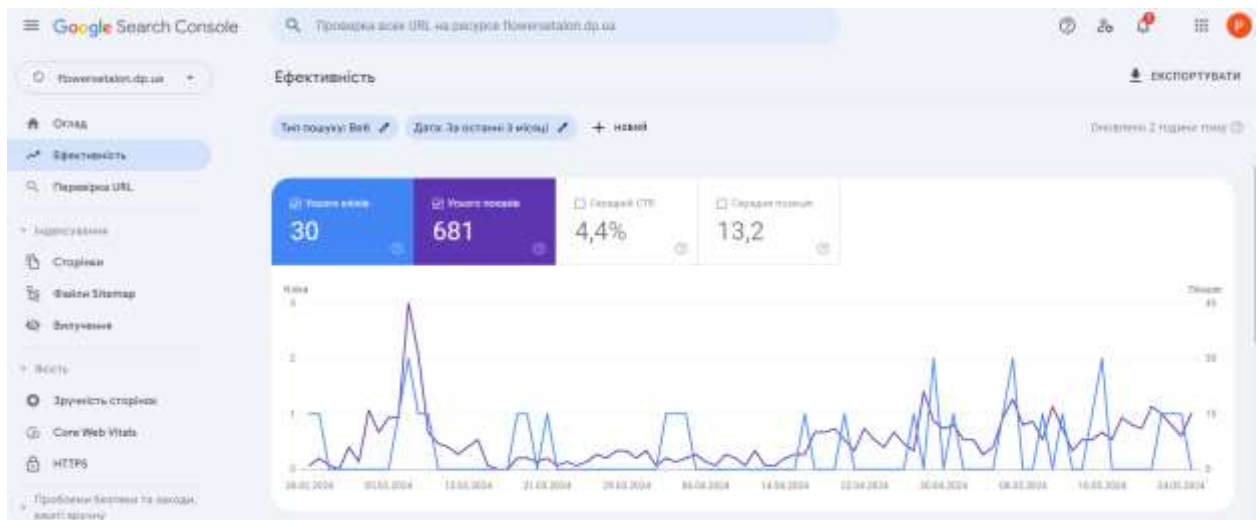


Рисунок 3.13 – Вкладка ефективності в Google Search Console

Таким чином, комплексний підхід до SEO налаштувань, що включає аналіз ключових слів, оптимізацію контенту, налаштування URL-адрес, мобільну оптимізацію, роботу з зовнішніми посиланнями, створення та оптимізацію

Sitemap, а також постійний моніторинг ефективності, забезпечує високі результати та сприяє успішному функціонуванню інтернет-магазину.

Для забезпечення швидкого завантаження сторінок та поліпшення користувацького досвіду була проведена ретельна оптимізація продуктивності сайту. Основним інструментом для оцінки та покращення продуктивності був Google PageSpeed Insights [27]. Цей інструмент дозволяє проводити детальний аналіз сайту та виявляти аспекти, які потребують оптимізації для покращення швидкості та ефективності.

Одним з перших кроків була оптимізація зображень. Всі зображення на сайті були переведені у формат .webp. Цей формат забезпечує значно менший розмір файлів без втрати якості, що значно скорочує час завантаження сторінок. Також було зменшено розмір зображень до оптимальних значень, що додатково знижує навантаження на сервер та підвищує швидкість завантаження. Оптимізація зображень є ключовим фактором для забезпечення швидкого завантаження сторінок, особливо на мобільних пристроях, де швидкість інтернет-з'єднання може бути обмеженою [28].

Наступним важливим кроком було використання кешування на стороні клієнта та сервера. Це дозволяє зберігати часто використовувані файли у кеші браузера користувача, що зменшує кількість запитів до сервера та прискорює завантаження сторінок при повторному відвідуванні сайту. Кешування дозволяє значно знизити навантаження на сервер і забезпечує швидкий доступ до контенту для користувачів [29].

Також було виконано оптимізацію CSS та JavaScript файлів. Усі CSS файли були об'єднані та мінімізовані, що зменшило їх розмір і кількість запитів до сервера. Аналогічно, JavaScript файли були мінімізовані та асинхронізовані, що дозволило зменшити час рендерингу сторінок та підвищити їхню інтерактивність. Мінімізація коду та зменшення кількості запитів до сервера є важливими кроками для підвищення продуктивності сайту та забезпечення швидкого завантаження. Важливим етапом було також впровадження критичних

CSS, що дозволяє завантажувати основні стилі на початку рендерингу сторінки, забезпечуючи швидке відображення основних елементів інтерфейсу.

Ще одним важливим кроком було налаштування lazy loading для зображень. Це означає, що зображення завантажуються лише тоді, коли вони стають видимими для користувача, що зменшує час початкового завантаження сторінки та економить трафік. Lazy loading дозволяє зменшити кількість завантажених ресурсів при першому відвідуванні сайту, що позитивно впливає на швидкість завантаження та користувацький досвід [30].

Аналіз сайту за допомогою Google PageSpeed Insights показав високий рівень ефективності для комп'ютерної версії сайту. Ефективність становила 97, оптимальні методи – 96, а оптимізація пошукових систем – 100. Це свідчить про високу швидкість завантаження сторінок та відмінну оптимізацію контенту (див. рис. 3.14).

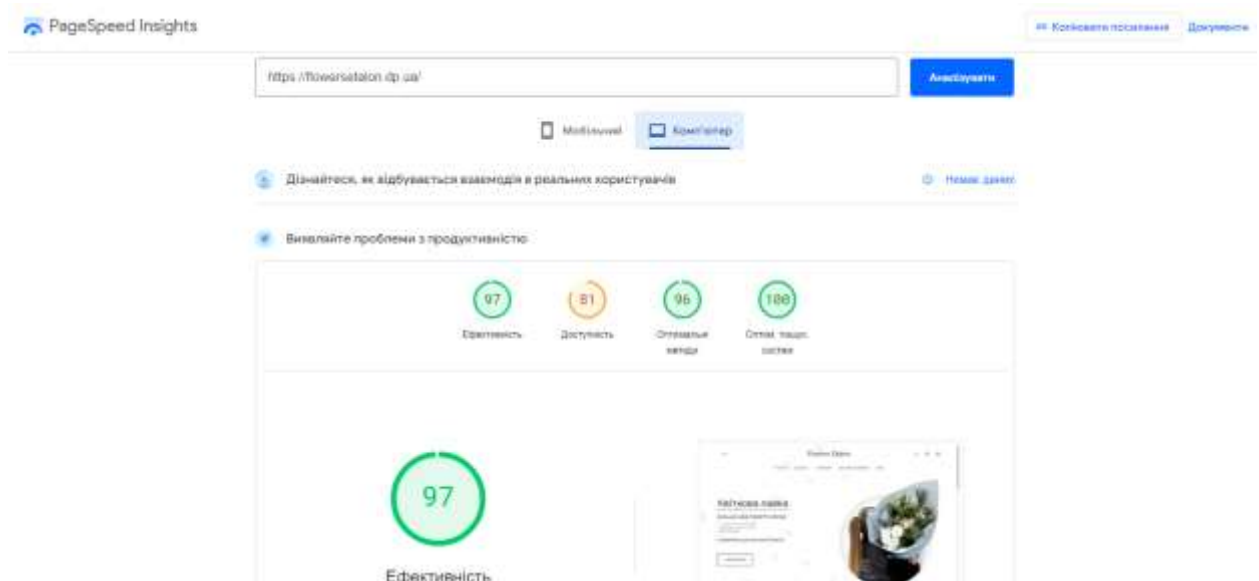


Рисунок 3.14 – Результати аналізу PageSpeed Insights для комп'ютерної версії сайту

Для мобільної версії сайту також було проведено аналіз, результати якого показали дещо нижчі показники ефективності. Ефективність становила 73, доступність – 81, оптимальні методи – 100. Це вказує на необхідність подальшої

оптимізації для мобільних пристроїв, щоб забезпечити користувачам ще кращий досвід (див. рис. 3.15). Показники для мобільних пристроїв можуть бути нижчими через обмеження пропускну здатності мобільних мереж та менші екрани, що потребують більш ретельної оптимізації зображень та інтерактивних елементів.

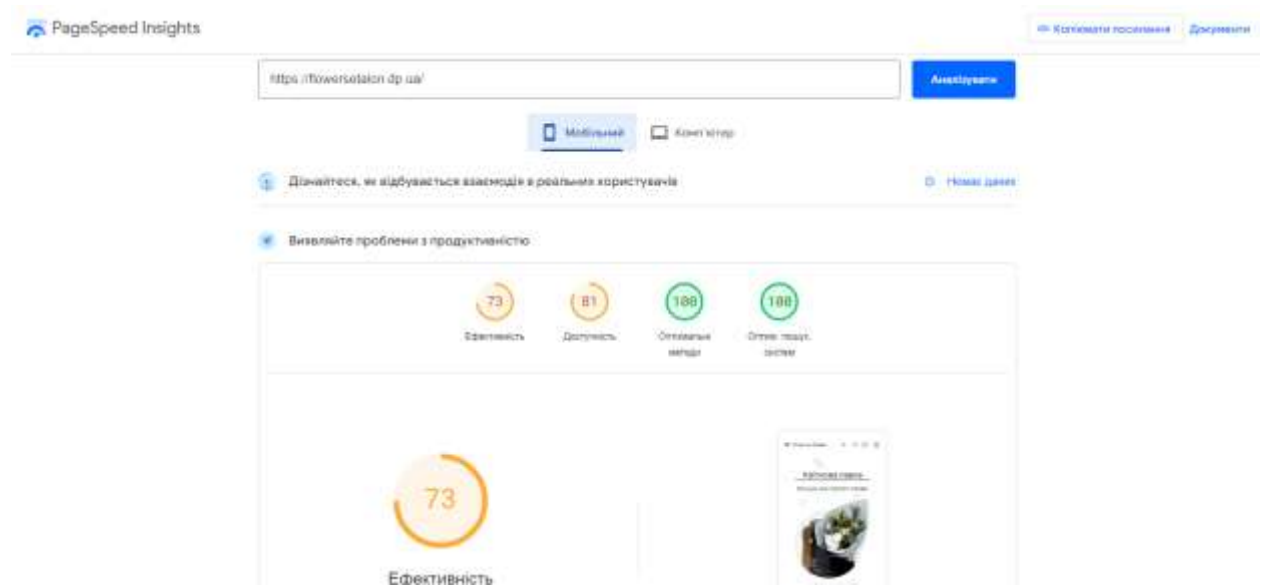


Рисунок 3.15 – Результати аналізу PageSpeed Insights для мобільної версії сайту

Ці заходи дозволили значно покращити продуктивність сайту та забезпечити користувачам швидкий доступ до контенту незалежно від пристрою, який вони використовують. Це сприяло підвищенню рівня задоволеності користувачів та покращенню показників конверсії на сайті. Оптимізація продуктивності є безперервним процесом, що включає регулярний аналіз та вдосконалення, щоб відповідати змінним вимогам користувачів та технологічним стандартам. Постійний моніторинг за допомогою інструментів, таких як Google Analytics та Google Search Console, дозволяє швидко виявляти і вирішувати проблеми, забезпечуючи стабільну і швидку роботу сайту.

### 3.4 Висновки до третього розділу

У третьому розділі було детально розглянуто управління та оптимізацію онлайн-магазину на платформі OpenCart. Основну увагу приділено адміністративній панелі OpenCart. Адміністративна панель дозволяє адміністратору ефективно керувати всіма процесами, що включає управління категоріями товарів, налаштування SEO-мета-тегів та створення акційних пропозицій.

Важливим аспектом розділу було описання головних сторінок сайту, таких як головна сторінка, сторінка товарів, кошик та оформлення замовлення. Детально описано функціональність кожної з цих сторінок та забезпечення їх зручності для користувачів. Окрему увагу приділено адаптивному дизайну, який дозволяє оптимізувати відображення сайту на різних пристроях, забезпечуючи комфортний користувацький досвід незалежно від типу пристрою.

SEO налаштування є критичним аспектом оптимізації онлайн-магазину, що включає використання ключових слів у заголовках, описах та текстах, оптимізацію контенту та швидкості завантаження сайту. Це сприяє покращенню видимості сайту у пошукових системах, що залучає більше клієнтів та підвищує обсяги продажів.

Оптимізація продуктивності сайту також була важливою частиною цього розділу. Використання інструментів для аналізу швидкості завантаження, таких як Google PageSpeed Insights, дозволило виявити та вирішити проблеми, пов'язані з продуктивністю.

Завдяки цим заходам вдалося забезпечити стабільну і швидку роботу сайту, підвищити рівень задоволеності користувачів та покращити показники конверсії. Оптимізація продуктивності є безперервним процесом, що включає регулярний аналіз та вдосконалення, щоб відповідати змінним вимогам користувачів та технологічним стандартам.

## ВИСНОВОКИ

Метою даної роботи було розробити веб-додаток для підтримки роботи квіткового магазину на платформі OpenCart, який забезпечить зручний інтерфейс для користувачів, можливість швидкого пошуку товарів, зручну систему кошика та оформлення замовлення, а також безпечні методи оплати і доставки. У процесі виконання роботи було досягнуто кілька важливих результатів:

1. Аналіз сучасного стану електронної комерції в Україні та визначення основних вимог до квіткових онлайн-магазинів. Було проведено детальне дослідження ринку електронної комерції в Україні, що дозволило визначити основні тенденції та особливості цього сектору. Також було проаналізовано потреби споживачів, які шукають зручні та безпечні способи здійснення покупок онлайн, що допомогло визначити основні вимоги до функціоналу веб-додатку.

2. Проведення аналізу конкурентних програмних продуктів та визначення оптимальних технологій для реалізації проєкту. На цьому етапі було проведено порівняльний аналіз різних платформ для розробки електронних магазинів, таких як OpenCart, Magento, WooCommerce та Shopify. Було виявлено, що кожна з цих платформ має свої переваги та недоліки. OpenCart було обрано завдяки її відкритому коду, широким можливостям для кастомізації та доступності численних модулів і розширень. Цей вибір дозволив забезпечити гнучкість та ефективність розробки, а також скоротити витрати на впровадження та підтримку проєкту.

3. Проектування архітектури веб-додатку, розробка дизайну інтерфейсу та інтеграція бази даних. Під час проектування архітектури було приділено особливу увагу зручності користування, швидкості завантаження сторінок та адаптивності дизайну. Були створені макети та прототипи основних сторінок сайту, що допомогло виявити можливі недоліки на ранніх етапах розробки. Інтерфейс було розроблено з урахуванням найкращих практик UX/UI дизайну, що забезпечило інтуїтивність та простоту користування для клієнтів. Інтеграція



бази даних включала налаштування таблиць для зберігання інформації про товари, замовлення та користувачів, що забезпечило ефективне управління даними та швидкий доступ до необхідної інформації.

4. Налаштування хостингу, встановлення SSL сертифіката та забезпечення безпеки даних користувачів. Було обрано надійний хостинг-провайдер, який забезпечує високу швидкість роботи серверів та безперебійну роботу сайту. Встановлення SSL сертифіката дозволило забезпечити захищене з'єднання між сервером та клієнтом, що є критично важливим для безпеки даних користувачів. Крім того, було налаштовано регулярне резервне копіювання даних, що забезпечує їх збереження у разі виникнення технічних проблем або збоїв у роботі сервера.

5. Оптимізація продуктивності сайту та налаштування SEO для підвищення видимості у пошукових системах. Було проведено детальний аналіз продуктивності сайту за допомогою інструментів, таких як Google PageSpeed Insights, що дозволило виявити та усунути вузькі місця, які уповільнювали завантаження сторінок. Впровадження технології lazy loading для зображень, налаштування кешування та оптимізація коду JavaScript та CSS значно покращили швидкість завантаження сторінок. Крім того, було налаштовано мета-теги, створено якісний контент та забезпечено його регулярне оновлення, що сприяло підвищенню позицій сайту у результатах пошукових систем та залученню більшої кількості клієнтів.

У результаті виконання кваліфікаційної роботи було досягнуто її мети – створено функціональний та зручний веб-додаток для підтримки роботи квіткового магазину. Розроблений додаток відповідає сучасним вимогам електронної комерції, забезпечує зручний інтерфейс для користувачів, швидкий доступ до товарів та послуг, ефективну систему управління замовленнями та безпечні методи оплати.

Серед основних переваг розробленого веб-додатку слід відзначити зручний інтерфейс, оптимізовану швидкість завантаження сторінок, високу

безпеку даних користувачів, адаптивний дизайн для використання на різних пристроях, а також широкий функціонал, включаючи швидкий пошук товарів, зручну систему кошика та оформлення замовлення.

Водночас існують деякі недоліки та можливості для покращення. Зокрема, необхідність постійного оновлення контенту для підтримки актуальності інформації та залучення нових клієнтів, потреба у технічній підтримці для забезпечення стабільної роботи веб-додатку, а також можливе додаткове розширення функціоналу шляхом впровадження нових сервісів та можливостей, таких як автоматизація маркетингу або інтеграція з аналітичними системами.

Під час тестування веб-додатку було підтверджено його відповідність очікуванням щодо ефективності роботи, зручності користування та безпеки. Отримані результати мають практичне значення, оскільки сприяють автоматизації та покращенню управління квітковим бізнесом.

Отже, розроблений веб-додаток для підтримки роботи квітового магазину є важливим кроком у розвитку електронної комерції та забезпечує нові можливості для бізнесу в умовах цифрової економіки. Цей додаток використовує передові технології та забезпечує надійну та точну роботу, що сприяє задоволенню потреб клієнтів та зростанню бізнесу.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ринок електронної комерції в Україні. URL: <https://soulpartners.com.ua/news/tpost/x2dve03v71-rinok-elektronno-komerts-v-ukran-dosyagn>
2. Augmented Reality Stats. TechJury. URL: <https://techjury.net/blog/augmented-reality-stats/>
3. Mobile Commerce: The Future of Trends. Shopify. URL: <https://www.shopify.com/enterprise/blog/mobile-commerce-future-trends>
4. Продажі мобільної торгівлі. URL: <https://www.oberlo.com/statistics/mobile-commerce-sales>
5. Thevenot, G. (2017). Big Data and AI in Marketing: How Big Data and AI Technologies Are Transforming the Field of Marketing. HAL-SHS. URL: <https://shs.hal.science/halshs-01764594/document>
6. Gkikas, D. C., Theodoridis, P. K., & Beligiannis, G. N. (2022). Enhanced Marketing Decision Making for Consumer Behaviour Classification Using Binary Decision Trees and a Genetic Algorithm Wrapper. Informatics, 9(2), 45. URL: <https://doi.org/10.3390/informatics9020045>
7. Amazon Personalize – Real-Time Personalization and Recommendation for Everyone. (2023). AWS News Blog. URL: <https://aws.amazon.com/ru/blogs/aws/amazon-personalize-real-time-personalization-and-recommendation-for-everyone/>
8. Netflix TechBlog. (2023). RecSysOps: Best Practices for Operating a Large-Scale Recommender System. Retrieved from <https://netflixtechblog.medium.com/recsysops-best-practices-for-operating-a-large-scale-recommender-system-95bbe195a841>
9. Ковальчук К. (2019). Соціальна комерція: тенденції та перспективи в Україні. Бізнес Інформ, 3, 84-90. URL: [https://www.business-inform.net/export\\_pdf/business-inform-2019-3\\_0-pages-84\\_90.pdf](https://www.business-inform.net/export_pdf/business-inform-2019-3_0-pages-84_90.pdf)

10. Топ соціальних мереж в Україні. Wizeclub. URL: <https://wizeclub.education/blog/top-sotsialnih-merezh/>
11. Magento vs WooCommerce vs Shopify vs OpenCart: Best Comparison. Доступно на Emizentech. URL: <https://www.emizentech.com/blog/magento-vs-woocommerce-vs-shopify-vs-opencart.html>
12. Flanagan, D. JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language. 7th ed. O'Reilly Media, 2020. 706 с.
13. Meyer, E. CSS: The Definitive Guide. 4th ed. O'Reilly Media, 2017. 1096 с.
14. PHP and MySQL for Dynamic Web Sites. URL: <https://www.php.net/manual/en/book.mysql.php>
15. SQL Tutorial - W3Schools. URL: <https://www.w3schools.com/sql/>
16. OpenCart Documentation. URL: <https://docs.opencart.com/>
17. Figma: Design, Prototype, and Gather Feedback All in One Place. URL: <https://www.figma.com/>
18. HOSTIQ.ua. URL: <https://hostiq.ua/>
19. Floristik.ua. URL: <https://floristik.ua/>
20. Buket24.dp.ua. URL: <https://buket24.dp.ua/ua/>
21. Website Information Architecture 101: All You Need to Know. Userlytics. URL: <https://www.userlytics.com/website-information-architecture/>
22. Прототипування сайту. WebTune. URL: <https://webtune.com.ua/statti/dyzajn/prototypuvannya-sajtu/>
23. What is Responsive Web Design? - Mozilla Developer Network. URL: [https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Responsive\\_Design](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design)
24. What is SSL/TLS: An In-Depth Guide. SSL.com. URL: <https://www.ssl.com/article/what-is-ssl-tls-an-in-depth-guide/>
25. Унікальний контент та SEO. Bizzzdev. URL: <https://bizzzdev.com/unique-content-seo/>

26. Google search console. URL: <https://search.google.com/search-console/about>
27. Google PageSpeed Insights. URL: <https://pagespeed.web.dev/>
28. Оптимізація зображень для сайту. Elit-Web. URL: <https://elit-web.ua/ua/blog/optimizaciya-izobrazhenij-dlya-sajta>
29. Кирик, О. Аналіз роботи методу оптимізованого кешування даних в мережі доставки контенту. Національний університет «Львівська політехніка». URL: [https://pt.nure.ua/wp-content/uploads/2020/01/181\\_kyryk\\_cdn.pdf](https://pt.nure.ua/wp-content/uploads/2020/01/181_kyryk_cdn.pdf)
30. Lazy loading images. Web.dev. URL: <https://web.dev/browser-level-image-lazy-loading/>

## ДОДАТОК

### ЛІСТИНГИ ФАЙЛІВ, ЩО ВІДНОСЯТЬСЯ ДО СЕРВЕРНОЇ ЧАСТИНИ ДОДАТКУ

#### 1. Лістинг файлу « header.php »

```
1. <?php
2. class ControllerCommonHeader extends Controller {
3. public function index() {
4. // Analytics
5. $this->load->model('setting/extension');
6. $data['analytics'] = array();
7. $analytics = $this->model_setting_extension->getExtensions('analytics');
8. foreach ($analytics as $analytic) {
9. if ($this->config->get('analytics_' . $analytic['code'] . '_status')) {
10. $data['analytics'][] = $this->load->controller('extension/analytics/' .
    $analytic['code'], $this->config->get('analytics_' . $analytic['code'] .
    '_status'));
11. }
12. }
13. if ($this->request->server['HTTPS']) {
14. $server = $this->config->get('config_ssl');
15. } else {
16. $server = $this->config->get('config_url');
17. }
18. if (is_file(DIR_IMAGE . $this->config->get('config_icon'))) {
19. $this->document->addLink($server . 'image/' . $this->config-
    >get('config_icon'), 'icon');
20. }
21. $data['title'] = $this->document->getTitle();
22. $data['base'] = $server;
23. $data['description'] = $this->document->getDescription();
24. $data['keywords'] = $this->document->getKeywords();
25. $data['links'] = $this->document->getLinks();
26. $data['styles'] = $this->document->getStyles();
27. $data['scripts'] = $this->document->getScripts('header');
28. $data['lang'] = $this->language->get('code');
29. $data['direction'] = $this->language->get('direction');
```

```

30.  $data['name'] = $this->config->get('config_name');
31.  if (is_file(DIR_IMAGE . $this->config->get('config_logo'))) {
32.      $data['logo'] = $server . 'image/' . $this->config->get('config_logo');
33.  } else {
34.      $data['logo'] = '';
35.  }
36.  $this->load->language('common/header');
37.  // Wishlist
38.  if ($this->customer->isLoggedIn()) {
39.      $this->load->model('account/wishlist');
40.      $data['text_wishlist'] = sprintf($this->language->get('text_wishlist'),
          $this->model_account_wishlist->getTotalWishlist());
41.  } else {
42.      $data['text_wishlist'] = sprintf($this->language->get('text_wishlist'),
          (isset($this->session->data['wishlist']) ? count($this->session-
          >data['wishlist']) : 0));
43.  }
44.  $data['text_logged'] = sprintf($this->language->get('text_logged'), $this-
          >url->link('account/account', '', true), $this->customer->getFirstName(),
          $this->url->link('account/logout', '', true));
45.  $data['home'] = $this->url->link('common/home');
46.  $data['wishlist'] = $this->url->link('account/wishlist', '', true);
47.  $data['logged'] = $this->customer->isLoggedIn();
48.  $data['account'] = $this->url->link('account/account', '', true);
49.  $data['register'] = $this->url->link('account/register', '', true);
50.  $data['login'] = $this->url->link('account/login', '', true);
51.  $data['order'] = $this->url->link('account/order', '', true);
52.  $data['transaction'] = $this->url->link('account/transaction', '', true);
53.  $data['download'] = $this->url->link('account/download', '', true);
54.  $data['logout'] = $this->url->link('account/logout', '', true);
55.  $data['shopping_cart'] = $this->url->link('checkout/cart');
56.  $data['checkout'] = $this->url->link('checkout/checkout', '', true);
57.  $data['contact'] = $this->url->link('information/contact');
58.  $data['telephone'] = $this->config->get('config_telephone');
59.  $data['language'] = $this->load->controller('common/language');
60.  $data['currency'] = $this->load->controller('common/currency');
61.  $data['search'] = $this->load->controller('common/search');
62.  $data['cart'] = $this->load->controller('common/cart');
63.  $data['menu'] = $this->load->controller('common/menu');

```

```

64.     return $this->load->view('common/header', $data);
65.   }
66. }

```

## 2. Лістинг файлу « home.php »

```

1. <?php
2. class ControllerCommonFooter extends Controller {
3.     public function index() {
4.         $this->load->language('common/footer');
5.         $this->load->model('catalog/information');
6.         $data['informations'] = array();
7.         foreach ($this->model_catalog_information->getInformations() as $result) {
8.             if ($result['bottom']) {
9.                 $data['informations'][] = array(
10.                    'title' => $result['title'],
11.                    'href'  => $this->url->link('information/information', 'information_id=' .
                                $result['information_id'])
12.                );
13.            }
14.        }
15.        $data['contact'] = $this->url->link('information/contact');
16.        $data['return'] = $this->url->link('account/return/add', '', true);
17.        $data['sitemap'] = $this->url->link('information/sitemap');
18.        $data['tracking'] = $this->url->link('information/tracking');
19.        $data['manufacturer'] = $this->url->link('product/manufacturer');
20.        $data['voucher'] = $this->url->link('account/voucher', '', true);
21.        $data['affiliate'] = $this->url->link('affiliate/login', '', true);
22. <?php
23. class ControllerCommonHome extends Controller {
24.     public function index() {
25.         $this->document->setTitle($this->config->get('config_meta_title'));
26.         $this->document->setDescription($this->config-
                                >get('config_meta_description'));
27.         $this->document->setKeywords($this->config->get('config_meta_keyword'));
28.
29.         if (isset($this->request->get['route'])) {
30.             $this->document->addLink($this->config->get('config_url'), 'canonical');

```



```

31. }
32.
33. $data['column_left'] = $this->load->controller('common/column_left');
34. $data['column_right'] = $this->load->controller('common/column_right');
35. $data['content_top'] = $this->load->controller('common/content_top');
36. $data['content_bottom'] = $this->load->controller('common/content_bottom');
37. $data['footer'] = $this->load->controller('common/footer');
38. $data['header'] = $this->load->controller('common/header');
39.
40. $this->response->setOutput($this->load->view('common/home', $data));
41. }
42. }

```

### 3. Лістинг файлу « footer.php »

```

1. <?php
2. class ControllerCommonFooter extends Controller {
3. public function index() {
4. $this->load->language('common/footer');
5. $this->load->model('catalog/information');
6. $data['informations'] = array();
7. foreach ($this->model_catalog_information->getInformations() as $result) {
8. if ($result['bottom']) {
9. $data['informations'][] = array(
10. 'title' => $result['title'],
11. 'href' => $this->url->link('information/information', 'information_id=' .
    $result['information_id'])
12. );
13. }
14. }
15. $data['contact'] = $this->url->link('information/contact');
16. $data['return'] = $this->url->link('account/return/add', '', true);
17. $data['sitemap'] = $this->url->link('information/sitemap');
18. $data['tracking'] = $this->url->link('information/tracking');
19. $data['manufacturer'] = $this->url->link('product/manufacturer');
20. $data['voucher'] = $this->url->link('account/voucher', '', true);
21. $data['affiliate'] = $this->url->link('affiliate/login', '', true);
22. $data['special'] = $this->url->link('product/special');

```

```

23.  $data['account'] = $this->url->link('account/account', '', true);
24.  $data['order'] = $this->url->link('account/order', '', true);
25.  $data['wishlist'] = $this->url->link('account/wishlist', '', true);
26.  $data['newsletter'] = $this->url->link('account/newsletter', '', true);
27.  $data['powered'] = sprintf($this->language->get('text_powered'), $this-
    >config->get('config_name'), date('Y', time()));
28.  // Whos Online
29.  if ($this->config->get('config_customer_online')) {
30.  $this->load->model('tool/online');
31.  if (isset($this->request->server['REMOTE_ADDR'])) {
32.  $ip = $this->request->server['REMOTE_ADDR'];
33.  } else {
34.  $ip = '';
35.  }
36.  if (isset($this->request->server['HTTP_HOST']) && isset($this->request-
    >server['REQUEST_URI'])) {
37.  $url = ($this->request->server['HTTPS'] ? 'https://' : 'http://') . $this-
    >request->server['HTTP_HOST'] . $this->request->server['REQUEST_URI'];
38.  } else {
39.  $url = '';
40.  }
41.  if (isset($this->request->server['HTTP_REFERER'])) {
42.  $referer = $this->request->server['HTTP_REFERER'];
43.  } else {
44.  $referer = '';
45.  }
46.  $this->model_tool_online->addOnline($ip, $this->customer->getId(), $url,
    $referer);
47.  }
48.  $data['scripts'] = $this->document->getScripts('footer');
49.  $data['styles'] = $this->document->getStyles('footer');
50.
51.  return $this->load->view('common/footer', $data);
52.  }
53.  }

```

#### 4. Лістинг файлу « product.php »

```
1. <?php
2. class ControllerProductProduct extends Controller {
3.     private $error = array();
4.
5.     public function index() {
6.         $this->load->language('product/product');
7.
8.         $data['breadcrumbs'] = array();
9.
10.        $data['breadcrumbs'][] = array(
11.            'text' => $this->language->get('text_home'),
12.            'href' => $this->url->link('common/home')
13.        );
14.
15.        $this->load->model('catalog/category');
16.
17.        if (isset($this->request->get['path'])) {
18.            $path = '';
19.
20.            $parts = explode('_', (string)$this->request->get['path']);
21.
22.            $category_id = (int)array_pop($parts);
23.
24.            foreach ($parts as $path_id) {
25.                if (!$path) {
26.                    $path = $path_id;
27.                } else {
28.                    $path .= '_' . $path_id;
29.                }
30.
31.                $category_info = $this->model_catalog_category->getCategory($path_id);
32.
33.                if ($category_info) {
34.                    $data['breadcrumbs'][] = array(
35.                        'text' => $category_info['name'],
36.                        'href' => $this->url->link('product/category', 'path=' . $path)
37.                    );
38.                }
39.            }
```

```
40.
41. // Set the last category breadcrumb
42. $category_info = $this->model_catalog_category->getCategory($category_id);
43.
44. if ($category_info) {
45.     $url = '';
46.
47.     if (isset($this->request->get['sort'])) {
48.         $url .= '&sort=' . $this->request->get['sort'];
49.     }
50.
51.     if (isset($this->request->get['order'])) {
52.         $url .= '&order=' . $this->request->get['order'];
53.     }
54.
55.     if (isset($this->request->get['page'])) {
56.         $url .= '&page=' . $this->request->get['page'];
57.     }
58.
59.     if (isset($this->request->get['limit'])) {
60.         $url .= '&limit=' . $this->request->get['limit'];
61.     }
62.
63.     $data['breadcrumbs'][] = array(
64.         'text' => $category_info['name'],
65.         'href' => $this->url->link('product/category', 'path=' . $this->request-
        >get['path'] . $url)
66.     );
67. }
68. }
69.
70. $this->load->model('catalog/manufacturer');
71.
72. if (isset($this->request->get['manufacturer_id'])) {
73.     $data['breadcrumbs'][] = array(
74.         'text' => $this->language->get('text_brand'),
75.         'href' => $this->url->link('product/manufacturer')
76.     );
77.
```

```
78.     $url = '';
79.
80.     if (isset($this->request->get['sort'])) {
81.         $url .= '&sort=' . $this->request->get['sort'];
82.     }
83.
84.     if (isset($this->request->get['order'])) {
85.         $url .= '&order=' . $this->request->get['order'];
86.     }
87.
88.     if (isset($this->request->get['page'])) {
89.         $url .= '&page=' . $this->request->get['page'];
90.     }
91.
92.     if (isset($this->request->get['limit'])) {
93.         $url .= '&limit=' . $this->request->get['limit'];
94.     }
95.
96.     $manufacturer_info = $this->model_catalog_manufacturer-
>getManufacturer($this->request->get['manufacturer_id']);
97.
98.     if ($manufacturer_info) {
99.         $data['breadcrumbs'][] = array(
100.            'text' => $manufacturer_info['name'],
101.            'href' => $this->url->link('product/manufacturer/info', 'manufacturer_id='
                . $this->request->get['manufacturer_id'] . $url)
102.        );
103.     }
104. }
105.
106. if (isset($this->request->get['search']) || isset($this->request-
>get['tag'])) {
107.     $url = '';
108.
109.     if (isset($this->request->get['search'])) {
110.         $url .= '&search=' . $this->request->get['search'];
111.     }
112.
113.     if (isset($this->request->get['tag'])) {
```

```
114. $url .= '&tag=' . $this->request->get['tag'];
115. }
116.
117. if (isset($this->request->get['description'])) {
118. $url .= '&description=' . $this->request->get['description'];
119. }
120.
121. if (isset($this->request->get['category_id'])) {
122. $url .= '&category_id=' . $this->request->get['category_id'];
123. }
124.
125. if (isset($this->request->get['sub_category'])) {
126. $url .= '&sub_category=' . $this->request->get['sub_category'];
127. }
128.
129. if (isset($this->request->get['sort'])) {
130. $url .= '&sort=' . $this->request->get['sort'];
131. }
132.
133. if (isset($this->request->get['order'])) {
134. $url .= '&order=' . $this->request->get['order'];
135. }
136.
137. if (isset($this->request->get['page'])) {
138. $url .= '&page=' . $this->request->get['page'];
139. }
140.
141. if (isset($this->request->get['limit'])) {
142. $url .= '&limit=' . $this->request->get['limit'];
143. }
144.
145. $data['breadcrumbs'][] = array(
146. 'text' => $this->language->get('text_search'),
147. 'href' => $this->url->link('product/search', $url)
148. );
149. }
150.
151. if (isset($this->request->get['product_id'])) {
152. $product_id = (int)$this->request->get['product_id'];
```

```
153. } else {
154.     $product_id = 0;
155. }
156.
157. $this->load->model('catalog/product');
158.
159. $product_info = $this->model_catalog_product->getProduct($product_id);
160.
161. if ($product_info) {
162.     $url = '';
163.
164.     if (isset($this->request->get['path'])) {
165.         $url .= '&path=' . $this->request->get['path'];
166.     }
167.
168.     if (isset($this->request->get['filter'])) {
169.         $url .= '&filter=' . $this->request->get['filter'];
170.     }
171.
172.     if (isset($this->request->get['manufacturer_id'])) {
173.         $url .= '&manufacturer_id=' . $this->request->get['manufacturer_id'];
174.     }
175.
176.     if (isset($this->request->get['search'])) {
177.         $url .= '&search=' . $this->request->get['search'];
178.     }
179.
180.     if (isset($this->request->get['tag'])) {
181.         $url .= '&tag=' . $this->request->get['tag'];
182.     }
183.
184.     if (isset($this->request->get['description'])) {
185.         $url .= '&description=' . $this->request->get['description'];
186.     }
187.
188.     if (isset($this->request->get['category_id'])) {
189.         $url .= '&category_id=' . $this->request->get['category_id'];
190.     }
191.
```

```
192. if (isset($this->request->get['sub_category'])) {
193.     $url .= '&sub_category=' . $this->request->get['sub_category'];
194. }
195.
196. if (isset($this->request->get['sort'])) {
197.     $url .= '&sort=' . $this->request->get['sort'];
198. }
199.
200. if (isset($this->request->get['order'])) {
201.     $url .= '&order=' . $this->request->get['order'];
202. }
203.
204. if (isset($this->request->get['page'])) {
205.     $url .= '&page=' . $this->request->get['page'];
206. }
207.
208. if (isset($this->request->get['limit'])) {
209.     $url .= '&limit=' . $this->request->get['limit'];
210. }
211.
212. $data['breadcrumbs'][] = array(
213.     'text' => $product_info['name'],
214.     'href' => $this->url->link('product/product', $url . '&product_id=' .
        $this->request->get['product_id'])
215. );
216.
217. $this->document->setTitle($product_info['meta_title']);
218. $this->document->setDescription($product_info['meta_description']);
219. $this->document->setKeywords($product_info['meta_keyword']);
220. $this->document->addLink($this->url->link('product/product', 'product_id='
    . $this->request->get['product_id']), 'canonical');
221. $this->document-
    >addScript('catalog/view/javascript/jquery/magnific/jquery.magnific-
        popup.min.js');
222. $this->document-
    >addStyle('catalog/view/javascript/jquery/magnific/magnific-popup.css');
223. $this->document-
    >addScript('catalog/view/javascript/jquery/datettimepicker/moment/moment.min.j
        s');
```



```

224. $this->document-
    >addScript('catalog/view/javascript/jquery/datetangepicker/moment/moment-with-
    locales.min.js');
225. $this->document-
    >addScript('catalog/view/javascript/jquery/datetangepicker/bootstrap-
    datetangepicker.min.js');
226. $this->document-
    >addStyle('catalog/view/javascript/jquery/datetangepicker/bootstrap-
    datetangepicker.min.css');
227.
228. $data['heading_title'] = $product_info['name'];
229.
230. $data['text_minimum'] = sprintf($this->language->get('text_minimum'),
    $product_info['minimum']);
231. $data['text_login'] = sprintf($this->language->get('text_login'), $this-
    >url->link('account/login', '', true), $this->url->link('account/register',
    '', true));
232.
233. $this->load->model('catalog/review');
234.
235. $data['tab_review'] = sprintf($this->language->get('tab_review'),
    $product_info['reviews']);
236.
237. $data['product_id'] = (int)$this->request->get['product_id'];
238. $data['manufacturer'] = $product_info['manufacturer'];
239. $data['manufacturers'] = $this->url->link('product/manufacturer/info',
    'manufacturer_id=' . $product_info['manufacturer_id']);
240. $data['model'] = $product_info['model'];
241. $data['reward'] = $product_info['reward'];
242. $data['points'] = $product_info['points'];
243. $data['description'] = html_entity_decode($product_info['description'],
    ENT_QUOTES, 'UTF-8');
244. $data['short_description'] =
    utf8_substr(trim(strip_tags(html_entity_decode($product_info['description'],
    ENT_QUOTES, 'UTF-8'))),0,$this->config->get('theme_' . $this->config-
    >get('config_theme') . '_product_description_length')) . '...';
245.
246. if ($product_info['quantity'] <= 0) {
247. $data['stock'] = $product_info['stock_status'];

```

```

248. } elseif ($this->config->get('config_stock_display')) {
249.     $data['stock'] = $product_info['quantity'];
250. } else {
251.     $data['stock'] = $this->language->get('text_instock');
252. }
253.
254. $this->load->model('tool/image');
255.
256. if ($product_info['image']) {
257.     $data['popup'] = $this->model_tool_image->resize($product_info['image'],
        $this->config->get('theme_' . $this->config->get('config_theme') .
        '_image_popup_width'), $this->config->get('theme_' . $this->config-
        >get('config_theme') . '_image_popup_height'));
258. } else {
259.     $data['popup'] = '';
260. }
261.
262. if ($product_info['image']) {
263.     $data['thumb'] = $this->model_tool_image->resize($product_info['image'],
        $this->config->get('theme_' . $this->config->get('config_theme') .
        '_image_thumb_width'), $this->config->get('theme_' . $this->config-
        >get('config_theme') . '_image_thumb_height'));
264. } else {
265.     $data['thumb'] = '';
266. }
267.
268. $data['images'] = array();
269.
270. $results = $this->model_catalog_product->getProductImages($this->request-
        >get['product_id']);
271.
272. foreach ($results as $result) {
273.     $data['images'][] = array(
274.         'popup' => $this->model_tool_image->resize($result['image'], $this-
            >config->get('theme_' . $this->config->get('config_theme') .
            '_image_popup_width'), $this->config->get('theme_' . $this->config-
            >get('config_theme') . '_image_popup_height')),
275.         'thumb' => $this->model_tool_image->resize($result['image'], $this-
            >config->get('theme_' . $this->config->get('config_theme') .

```

```

    '_image_additional_width'), $this->config->get('theme_' . $this->config-
    >get('config_theme') . '_image_additional_height'))
276. );
277. }
278.
279. if ($this->customer->isLoggedIn() || !$this->config-
    >get('config_customer_price')) {
280. $data['price'] = $this->currency->format($this->tax-
    >calculate($product_info['price'], $product_info['tax_class_id'], $this-
    >config->get('config_tax')), $this->session->data['currency']);
281. } else {
282. $data['price'] = false;
283. }
284.
285. if ((float)$product_info['special']) {
286. $data['special'] = $this->currency->format($this->tax-
    >calculate($product_info['special'], $product_info['tax_class_id'], $this-
    >config->get('config_tax')), $this->session->data['currency']);
287.
288. } else {
289. $data['special'] = false;
290.
291. }
292.
293. if ($this->config->get('config_tax')) {
294. $data['tax'] = $this->currency->format((float)$product_info['special'] ?
    $product_info['special'] : $product_info['price'], $this->session-
    >data['currency']);
295. } else {
296. $data['tax'] = false;
297. }
298.
299. $discounts = $this->model_catalog_product->getProductDiscounts($this-
    >request->get['product_id']);
300.
301. $data['discounts'] = array();
302.
303. foreach ($discounts as $discount) {
304. $data['discounts'][] = array(

```

```

305. 'quantity' => $discount['quantity'],
306. 'price'     => $this->currency->format($this->tax-
    >calculate($discount['price'], $product_info['tax_class_id'], $this->config-
    >get('config_tax')), $this->session->data['currency'])
307. );
308. }
309.
310. $data['options'] = array();
311.
312. foreach ($this->model_catalog_product->getProductOptions($this->request-
    >get['product_id']) as $option) {
313.     $product_option_value_data = array();
314.
315.     foreach ($option['product_option_value'] as $option_value) {
316.         if (!$option_value['subtract'] || ($option_value['quantity'] > 0)) {
317.             if (((($this->config->get('config_customer_price') && $this->customer-
                >isLoggedIn()) || !$this->config->get('config_customer_price')) &&
                (float)$option_value['price'])) {
318.                 $price = $this->currency->format($this->tax-
                    >calculate($option_value['price'], $product_info['tax_class_id'], $this-
                    >config->get('config_tax') ? 'P' : false), $this->session->data['currency']);
319.             } else {
320.                 $price = false;
321.             }
322.
323.             $product_option_value_data[] = array(
324.                 'product_option_value_id' => $option_value['product_option_value_id'],
325.                 'option_value_id'       => $option_value['option_value_id'],
326.                 'name'                   => $option_value['name'],
327.                 'image'                   => $this->model_tool_image-
                    >resize($option_value['image'], 50, 50),
328.                 'price'                   => $price,
329.                 'price_prefix'           => $option_value['price_prefix']
330.             );
331.         }
332.     }
333.
334.     $data['options'][] = array(
335.         'product_option_id' => $option['product_option_id'],

```

```

336. 'product_option_value' => $product_option_value_data,
337. 'option_id'           => $option['option_id'],
338. 'name'                => $option['name'],
339. 'type'                 => $option['type'],
340. 'value'                => $option['value'],
341. 'required'             => $option['required']
342. );
343. }
344.
345. if ($product_info['minimum']) {
346. $data['minimum'] = $product_info['minimum'];
347. } else {
348. $data['minimum'] = 1;
349. }
350.
351. $data['review_status'] = $this->config->get('config_review_status');
352.
353. if ($this->config->get('config_review_guest') || $this->customer-
    >isLoggedIn()) {
354. $data['review_guest'] = true;
355. } else {
356. $data['review_guest'] = false;
357. }
358.
359. if ($this->customer->isLoggedIn()) {
360. $data['customer_name'] = $this->customer->getFirstName() . '&nbsp;' .
    $this->customer->getLastName();
361. } else {
362. $data['customer_name'] = '';
363. }
364.
365. $data['reviews'] = sprintf($this->language->get('text_reviews'),
    (int)$product_info['reviews']);
366. $data['rating'] = (int)$product_info['rating'];
367.
368. // Captcha
369. if ($this->config->get('captcha_' . $this->config->get('config_captcha') .
    '_status') && in_array('review', (array)$this->config-
    >get('config_captcha_page'))) {

```

```

370. $data['captcha'] = $this->load->controller('extension/captcha/' . $this-
    >config->get('config_captcha'));
371. } else {
372. $data['captcha'] = '';
373. }
374.
375. $data['share'] = $this->url->link('product/product', 'product_id=' .
    (int)$this->request->get['product_id']);
376.
377. $data['attribute_groups'] = $this->model_catalog_product-
    >getProductAttributes($this->request->get['product_id']);
378.
379. $data['products'] = array();
380.
381. $results = $this->model_catalog_product->getProductRelated($this->request-
    >get['product_id']);
382.
383. foreach ($results as $result) {
384. if ($result['image']) {
385. $image = $this->model_tool_image->resize($result['image'], $this->config-
    >get('theme_' . $this->config->get('config_theme') . '_image_related_width'),
    $this->config->get('theme_' . $this->config->get('config_theme') .
    '_image_related_height'));
386. } else {
387. $image = $this->model_tool_image->resize('placeholder.png', $this->config-
    >get('theme_' . $this->config->get('config_theme') . '_image_related_width'),
    $this->config->get('theme_' . $this->config->get('config_theme') .
    '_image_related_height'));
388. }
389.
390. if ($this->customer->isLoggedIn() || !$this->config-
    >get('config_customer_price')) {
391. $price = $this->currency->format($this->tax->calculate($result['price'],
    $result['tax_class_id'], $this->config->get('config_tax')), $this->session-
    >data['currency']);
392. } else {
393. $price = false;
394. }
395.

```

```

396. if ((float)$result['special']) {
397.   $special = $this->currency->format($this->tax-
      >calculate($result['special'], $result['tax_class_id'], $this->config-
      >get('config_tax')), $this->session->data['currency']);
398.   $rate_special = round(100 - ((float)$result['special'] / $result['price']
      * 100));
399. } else {
400.   $special = false;
401.   $rate_special = false;
402. }
403.
404. if ($this->config->get('config_tax')) {
405.   $tax = $this->currency->format((float)$result['special'] ?
      $result['special'] : $result['price'], $this->session->data['currency']);
406. } else {
407.   $tax = false;
408. }
409.
410. if ($this->config->get('config_review_status')) {
411.   $rating = (int)$result['rating'];
412. } else {
413.   $rating = false;
414. }
415. $model = $result['model'];
416. $data['tags'] = array();
417. if ($result['tag']) {
418.   $tags = explode(',', $result['tag']);
419.   foreach ($tags as $tag) {
420.     $data['tags'][] = array(
421.       'tag' => trim($tag),
422.       'href' => $this->url->link('product/search', 'tag=' . trim($tag))
423.     );
424.   }
425. }
426. $data['products'][] = array(
427.   'product_id' => $result['product_id'],
428.   'thumb' => $image,
429.   'model' => $model,
430.   'name' => $result['name'],

```

```

431. 'description' =>
    utf8_substr(trim(strip_tags(html_entity_decode($result['description'],
    ENT_QUOTES, 'UTF-8'))), 0, $this->config->get('theme_' . $this->config-
    >get('config_theme') . '_product_description_length')) . '..',
432. 'price'      => $price,
433. 'special'    => $special,
434. 'rate_special' => $rate_special,
435. 'tax'        => $tax,
436. 'tags'       => $data['tags'],
437. 'minimum'    => $result['minimum'] > 0 ? $result['minimum'] : 1,
438. 'rating'     => $rating,
439. 'href'       => $this->url->link('product/product', 'product_id=' .
    $result['product_id'])
440. );
441. }
442.
443. $data['tags'] = array();
444.
445. if ($product_info['tag']) {
446.     $tags = explode(',', $product_info['tag']);
447.
448.     foreach ($tags as $tag) {
449.         $data['tags'][] = array(
450.             'tag' => trim($tag),
451.             'href' => $this->url->link('product/search', 'tag=' . trim($tag))
452.         );
453.     }
454. }
455.
456. $data['recurrings'] = $this->model_catalog_product->getProfiles($this-
    >request->get['product_id']);
457.
458. $this->model_catalog_product->updateViewed($this->request-
    >get['product_id']);
459.
460. $data['column_left'] = $this->load->controller('common/column_left');
461. $data['column_right'] = $this->load->controller('common/column_right');
462. $data['content_top'] = $this->load->controller('common/content_top');

```



```
463. $data['content_bottom'] = $this->load-
    >controller('common/content_bottom');
464. $data['footer'] = $this->load->controller('common/footer');
465. $data['header'] = $this->load->controller('common/header');
466.
467. $this->response->setOutput($this->load->view('product/product', $data));
468. } else {
469. $url = '';
470.
471. if (isset($this->request->get['path'])) {
472. $url .= '&path=' . $this->request->get['path'];
473. }
474.
475. if (isset($this->request->get['filter'])) {
476. $url .= '&filter=' . $this->request->get['filter'];
477. }
478.
479. if (isset($this->request->get['manufacturer_id'])) {
480. $url .= '&manufacturer_id=' . $this->request->get['manufacturer_id'];
481. }
482.
483. if (isset($this->request->get['search'])) {
484. $url .= '&search=' . $this->request->get['search'];
485. }
486.
487. if (isset($this->request->get['tag'])) {
488. $url .= '&tag=' . $this->request->get['tag'];
489. }
490.
491. if (isset($this->request->get['description'])) {
492. $url .= '&description=' . $this->request->get['description'];
493. }
494.
495. if (isset($this->request->get['category_id'])) {
496. $url .= '&category_id=' . $this->request->get['category_id'];
497. }
498.
499. if (isset($this->request->get['sub_category'])) {
500. $url .= '&sub_category=' . $this->request->get['sub_category'];
```

```
501. }
502.
503. if (isset($this->request->get['sort'])) {
504.     $url .= '&sort=' . $this->request->get['sort'];
505. }
506.
507. if (isset($this->request->get['order'])) {
508.     $url .= '&order=' . $this->request->get['order'];
509. }
510.
511. if (isset($this->request->get['page'])) {
512.     $url .= '&page=' . $this->request->get['page'];
513. }
514.
515. if (isset($this->request->get['limit'])) {
516.     $url .= '&limit=' . $this->request->get['limit'];
517. }
518.
519. $data['breadcrumbs'][] = array(
520.     'text' => $this->language->get('text_error'),
521.     'href' => $this->url->link('product/product', $url . '&product_id=' .
        $product_id)
522. );
523.
524. $this->document->setTitle($this->language->get('text_error'));
525.
526. $data['continue'] = $this->url->link('common/home');
527.
528. $this->response->addHeader($this->request->server['SERVER_PROTOCOL'] . '
        404 Not Found');
529.
530. $data['column_left'] = $this->load->controller('common/column_left');
531. $data['column_right'] = $this->load->controller('common/column_right');
532. $data['content_top'] = $this->load->controller('common/content_top');
533. $data['content_bottom'] = $this->load->
        >controller('common/content_bottom');
534. $data['footer'] = $this->load->controller('common/footer');
535. $data['header'] = $this->load->controller('common/header');
536.
```

```
537. $this->response->setOutput($this->load->view('error/not_found', $data));
538. }
539. }
540.
541. public function review() {
542. $this->load->language('product/product');
543.
544. $this->load->model('catalog/review');
545.
546. if (isset($this->request->get['page'])) {
547. $page = $this->request->get['page'];
548. } else {
549. $page = 1;
550. }
551.
552. $data['reviews'] = array();
553.
554. $review_total = $this->model_catalog_review-
    >getTotalReviewsByProductId($this->request->get['product_id']);
555.
556. $results = $this->model_catalog_review->getReviewsByProductId($this-
    >request->get['product_id'], ($page - 1) * 5, 5);
557.
558. foreach ($results as $result) {
559. $data['reviews'][] = array(
560. 'author' => $result['author'],
561. 'text' => nl2br($result['text']),
562. 'rating' => (int)$result['rating'],
563. 'date_added' => date($this->language->get('date_format_short'),
    strtotime($result['date_added']))
564. );
565. }
566.
567. $pagination = new Pagination();
568. $pagination->total = $review_total;
569. $pagination->page = $page;
570. $pagination->limit = 5;
571. $pagination->url = $this->url->link('product/product/review',
    'product_id=' . $this->request->get['product_id'] . '&page={page}');
```

```

572.
573. $data['pagination'] = $pagination->render();
574.
575. $data['results'] = sprintf($this->language->get('text_pagination'),
    ($review_total) ? (($page - 1) * 5) + 1 : 0, (((($page - 1) * 5) >
    ($review_total - 5)) ? $review_total : (((($page - 1) * 5) + 5),
    $review_total, ceil($review_total / 5));
576.
577. $this->response->setOutput($this->load->view('product/review', $data));
578. }
579.
580. public function write() {
581. $this->load->language('product/product');
582.
583. $json = array();
584.
585. if ($this->request->server['REQUEST_METHOD'] == 'POST') {
586. if ((utf8_strlen($this->request->post['name']) < 3) || (utf8_strlen($this-
    >request->post['name']) > 25)) {
587. $json['error'] = $this->language->get('error_name');
588. }
589.
590. if ((utf8_strlen($this->request->post['text']) < 25) ||
    (utf8_strlen($this->request->post['text']) > 1000)) {
591. $json['error'] = $this->language->get('error_text');
592. }
593.
594. if (empty($this->request->post['rating']) || $this->request-
    >post['rating'] < 0 || $this->request->post['rating'] > 5) {
595. $json['error'] = $this->language->get('error_rating');
596. }
597.
598. // Captcha
599. if ($this->config->get('captcha_' . $this->config->get('config_captcha')) .
    '_status') && in_array('review', (array)$this->config-
    >get('config_captcha_page')) {
600. $captcha = $this->load->controller('extension/captcha/' . $this->config-
    >get('config_captcha') . '/validate');
601.

```

```
602. if ($captcha) {
603.     $json['error'] = $captcha;
604. }
605. }
606.
607. if (!isset($json['error'])) {
608.     $this->load->model('catalog/review');
609.
610.     $this->model_catalog_review->addReview($this->request->get['product_id'],
        $this->request->post);
611.
612.     $json['success'] = $this->language->get('text_success');
613. }
614. }
615.
616. $this->response->addHeader('Content-Type: application/json');
617. $this->response->setOutput(json_encode($json));
618. }
619.
620. public function getRecurringDescription() {
621.     $this->load->language('product/product');
622.     $this->load->model('catalog/product');
623.
624.     if (isset($this->request->post['product_id'])) {
625.         $product_id = $this->request->post['product_id'];
626.     } else {
627.         $product_id = 0;
628.     }
629.
630.     if (isset($this->request->post['recurring_id'])) {
631.         $recurring_id = $this->request->post['recurring_id'];
632.     } else {
633.         $recurring_id = 0;
634.     }
635.
636.     if (isset($this->request->post['quantity'])) {
637.         $quantity = $this->request->post['quantity'];
638.     } else {
639.         $quantity = 1;
```

```
640. }
641.
642. $product_info = $this->model_catalog_product->getProduct($product_id);
643.
644. $recurring_info = $this->model_catalog_product->getProfile($product_id,
    $recurring_id);
645.
646. $json = array();
647.
648. if ($product_info && $recurring_info) {
649.     if (!$json) {
650.         $frequencies = array(
651.             'day'          => $this->language->get('text_day'),
652.             'week'        => $this->language->get('text_week'),
653.             'semi_month' => $this->language->get('text_semi_month'),
654.             'month'       => $this->language->get('text_month'),
655.             'year'        => $this->language->get('text_year'),
656.         );
657.
658.         if ($recurring_info['trial_status'] == 1) {
659.             $price = $this->currency->format($this->tax-
                >calculate($recurring_info['trial_price'] * $quantity,
                $product_info['tax_class_id'], $this->config->get('config_tax')), $this-
                >session->data['currency']);
660.             $trial_text = sprintf($this->language->get('text_trial_description'),
                $price, $recurring_info['trial_cycle'],
                $frequencies[$recurring_info['trial_frequency']],
                $recurring_info['trial_duration']) . ' ';
661.         } else {
662.             $trial_text = '';
663.         }
664.
665.         $price = $this->currency->format($this->tax-
                >calculate($recurring_info['price'] * $quantity,
                $product_info['tax_class_id'], $this->config->get('config_tax')), $this-
                >session->data['currency']);
666.
667.         if ($recurring_info['duration']) {
```

```

668. $text = $trial_text . sprintf($this->language-
    >get('text_payment_description'), $price, $recurring_info['cycle'],
    $frequencies[$recurring_info['frequency']], $recurring_info['duration']);
669. } else {
670. $text = $trial_text . sprintf($this->language->get('text_payment_cancel'),
    $price, $recurring_info['cycle'], $frequencies[$recurring_info['frequency']],
    $recurring_info['duration']);
671. }
672.
673. $json['success'] = $text;
674. }
675. }
676.
677. $this->response->addHeader('Content-Type: application/json');
678. $this->response->setOutput(json_encode($json));
679. }
680. }

```

## 5. Лістинг файлу « checkout.php »

```

1. <?php
2. class ControllerCheckoutCheckout extends Controller {
3. public function index() {
4. // Validate cart has products and has stock.
5. if ((!$this->cart->hasProducts() && empty($this->session->data['vouchers']))
    || (!$this->cart->hasStock() && !$this->config-
    >get('config_stock_checkout'))) {
6. $this->response->redirect($this->url->link('checkout/cart'));
7. }
8. // Validate minimum quantity requirements.
9. $products = $this->cart->getProducts();
10. foreach ($products as $product) {
11. $product_total = 0;
12. foreach ($products as $product_2) {
13. if ($product_2['product_id'] == $product['product_id']) {
14. $product_total += $product_2['quantity'];
15. }
16. }

```

```

17.   if ($product['minimum'] > $product_total) {
18.     $this->response->redirect($this->url->link('checkout/cart'));
19.   }
20. }
21. $this->load->language('checkout/checkout');
22. $this->document->setTitle($this->language->get('heading_title'));
23. $this->document-
    >addScript('catalog/view/javascript/jquery/datetimepicker/moment/moment.min.js');
24. $this->document-
    >addScript('catalog/view/javascript/jquery/datetimepicker/moment/moment-with-
    locales.min.js');
25. $this->document-
    >addScript('catalog/view/javascript/jquery/datetimepicker/bootstrap-
    datetimepicker.min.js');
26. $this->document-
    >addStyle('catalog/view/javascript/jquery/datetimepicker/bootstrap-
    datetimepicker.min.css');
27.   if ($this->config->get('payment_klarna_account') || $this->config-
    >get('payment_klarna_invoice')) {
28.     $this->document-
    >addScript('http://cdn.klarna.com/public/kitt/toc/v1.0/js/klarna.terms.min.js
    ');
29.   }
30. $data['breadcrumbs'] = array();
31. $data['breadcrumbs'][] = array(
32.   'text' => $this->language->get('text_home'),
33.   'href' => $this->url->link('common/home')
34. );
35. $data['breadcrumbs'][] = array(
36.   'text' => $this->language->get('text_cart'),
37.   'href' => $this->url->link('checkout/cart')
38. );
39. $data['breadcrumbs'][] = array(
40.   'text' => $this->language->get('heading_title'),
41.   'href' => $this->url->link('checkout/checkout', '', true)
42. );
43. $data['text_checkout_option'] = sprintf($this->language-
    >get('text_checkout_option'), 1);

```



```
44.     $data['text_checkout_account'] = sprintf($this->language-
        >get('text_checkout_account'), 2);
45.     $data['text_checkout_payment_address'] = sprintf($this->language-
        >get('text_checkout_payment_address'), 2);
46.     $data['text_checkout_shipping_address'] = sprintf($this->language-
        >get('text_checkout_shipping_address'), 3);
47.     $data['text_checkout_shipping_method'] = sprintf($this->language-
        >get('text_checkout_shipping_method'), 4);
48.
49.     if ($this->cart->hasShipping()) {
50.         $data['text_checkout_payment_method'] = sprintf($this->language-
            >get('text_checkout_payment_method'), 5);
51.         $data['text_checkout_confirm'] = sprintf($this->language-
            >get('text_checkout_confirm'), 6);
52.     } else {
53.         $data['text_checkout_payment_method'] = sprintf($this->language-
            >get('text_checkout_payment_method'), 3);
54.         $data['text_checkout_confirm'] = sprintf($this->language-
            >get('text_checkout_confirm'), 4);
55.     }
56.     if (isset($this->session->data['error'])) {
57.         $data['error_warning'] = $this->session->data['error'];
58.         unset($this->session->data['error']);
59.     } else {
60.         $data['error_warning'] = '';
61.     }
62.     $data['logged'] = $this->customer->isLoggedIn();
63.     if (isset($this->session->data['account'])) {
64.         $data['account'] = $this->session->data['account'];
65.     } else {
66.         $data['account'] = '';
67.     }
68.     $data['shipping_required'] = $this->cart->hasShipping();
69.     $data['column_left'] = $this->load->controller('common/column_left');
70.     $data['column_right'] = $this->load->controller('common/column_right');
71.     $data['content_top'] = $this->load->controller('common/content_top');
72.     $data['content_bottom'] = $this->load-
        >controller('common/content_bottom');
73.     $data['footer'] = $this->load->controller('common/footer');
```

```

74.     $data['header'] = $this->load->controller('common/header');
75.     $this->response->setOutput($this->load->view('checkout/checkout', $data));
76. }
77. public function country() {
78.     $json = array();
79.     $this->load->model('localisation/country');
80.     $country_info = $this->model_localisation_country->getCountry($this-
    >request->get['country_id']);
81.     if ($country_info) {
82.         $this->load->model('localisation/zone');
83.         $json = array(
84.             'country_id'      => $country_info['country_id'],
85.             'name'           => $country_info['name'],
86.             'iso_code_2'     => $country_info['iso_code_2'],
87.             'iso_code_3'     => $country_info['iso_code_3'],
88.             'address_format' => $country_info['address_format'],
89.             'postcode_required' => $country_info['postcode_required'],
90.             'zone'           => $this->model_localisation_zone-
    >getZonesByCountryId($this->request->get['country_id']),
91.             'status'        => $country_info['status']
92.         );
93.     }
94.     $this->response->addHeader('Content-Type: application/json');
95.     $this->response->setOutput(json_encode($json));
96. }
97. public function customfield() {
98.     $json = array();
99.     $this->load->model('account/custom_field');
100.    // Customer Group
101.    if (isset($this->request->get['customer_group_id']) && is_array($this-
    >config->get('config_customer_group_display')) && in_array($this->request-
    >get['customer_group_id'], $this->config-
    >get('config_customer_group_display'))) {
102.        $customer_group_id = $this->request->get['customer_group_id'];
103.    } else {
104.        $customer_group_id = $this->config->get('config_customer_group_id');
105.    }
106.    $custom_fields = $this->model_account_custom_field-
    >getCustomFields($customer_group_id);

```

```
107. foreach ($custom_fields as $custom_field) {  
108.   $json[] = array(  
109.     'custom_field_id' => $custom_field['custom_field_id'],  
110.     'required'       => $custom_field['required']  
111.   );  
112. }  
113. $this->response->addHeader('Content-Type: application/json');  
114. $this->response->setOutput(json_encode($json));  
115. }  
116. }
```