

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем**

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.

_____ Ю. П. Кондратенко

« ____ » _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**ВЕБЗАСТОСУНОК ДЛЯ УПРАВЛІННЯ ІНТЕРНЕТ-
МАГАЗИНОМ ЕКОТОВАРІВ**

Спеціальність 122 «Комп'ютерні науки»

122 – КРБ – 402.22010201

Виконав студент 4-го курсу, групи 402

_____ *Є. Д. Богза*

«20» червня 2024 р.

Керівник: д-р пед. наук, професор

_____ *О. П. Мещанінов*

«20» червня 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
« ____ » _____ 2024 р.

З А В Д А Н Н Я
на виконання кваліфікаційної роботи

Видано студенту групи 402 факультету комп'ютерних наук Богзі Євгенію Дмитровичу.

1. Тема кваліфікаційної роботи «Вебзастосунок для управління інтернет-магазином екотоварів».

Керівник роботи Мещанінов Олександр Павлович, д-р пед. наук, професор.
Затв. наказом Ректора ЧНУ ім. Петра Могили від «28» грудня 2023 р. № 271

2. Строк представлення кваліфікаційної роботи студентом «20» червня 2024 р.

3. Вхідні (початкові) дані до роботи: сфера продажу екотоварів; існуючі вебзастосунки для продажу екотоварів; вебзастосунки управління інтернет-магазинами екотоварів.

Очікуваний результат: система управління інтернет-магазином екотоварів.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- аналіз сфери продажу екотоварів та особливостей вебзастосунків в цій сфері;
- огляд та аналіз існуючих вебзастосунків з продажу екотоварів;
- вибір та обґрунтування технологій розробки вебзастосунку;

– реалізація серверної частини та інтерфейсу вебзастосунку та використані технології для реалізації.

5. Перелік графічного матеріалу: презентація, 24 рисунки та 10 таблиць.

6. Завдання до спеціальної частини: «Опалення та вентиляція приміщень для працівників сфери інформаційної безпеки»

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Алексеева А. О., доцент кафедри екології	

Керівник роботи д-р пед. наук, проф. Мещанінов О. П.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Завдання прийнято до виконання Богза Є. Д.
(прізвище та ініціали)

_____ (підпис)

Дата видачі завдання « 14 » січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Вебзастосунок для управління інтернет-магазином екотоварів

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників КРБ	10.11.2023	15.11.2023	Виконано
2	Отримання завдання на виконання КРБ	10.01.2024	15.01.2024	Виконано
3	Складання календарного плану роботи на весь період виконання КРБ	16.01.2024	30.01.2024	Виконано
4	Отримання завдання на переддипломну практику	15.04.2024	29.04.2024	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до КРБ	29.04.2024	11.05.2024	Виконано
6	Розробка звіту з переддипломної практики	12.05.2024	15.05.2024	Виконано
7	Виконання КРБ: Аналіз сфери продажу екотоварів, огляд та аналіз існуючих вебзастосунків, розробка ПЗ	13.05.2024	22.06.2024	Виконано
8	Перший попередній захист КРБ на засіданні комісії кафедри	27.05.2024	27.05.2024	Виконано
9	Доробка та остаточне оформлення КРБ	28.05.2024	09.06.2024	Виконано
10	Другий попередній захист КРБ на засіданні комісії кафедри	10.06.2024	10.06.2024	Виконано
11	Подання КРБ рецензенту	13.06.2024	13.06.2024	Виконано
11	Подання КРБ, її електронної копії та інших документів (відгуку, рецензії) до захисту	17.06.2024	21.06.2024	Виконано
12	Захист КРБ перед екзаменаційною комісією (ЕК)	24.06.2024	28.06.2024	Виконано

Розробив студент Богза Є. Д.
(прізвище, ім'я, по батькові студента)

_____ (підпис)

Керівник роботи д-р пед. наук, проф. Мещанінов О. П.
(посада, прізвище, ім'я, по батькові)

_____ (підпис)

« 29 » _____ 01 _____ 2024 р

АНОТАЦІЯ

**кваліфікаційної роботи студента групи 402 ЧНУ ім. Петра Могили
Богзи Євгенія Дмитровича**

Тема: «Вебзастосунок для управління інтернет-магазином екотоварів»

Актуальність теми. Зростаючий інтерес до екологічно чистих товарів робить розробку вебзастосунків для управління інтернет-магазинами екотоварів особливо актуальною. Попит на натуральні, органічні продукти та товари, виготовлені з мінімальним негативним впливом на довкілля, стимулює створення функціональних вебплатформ для їх реалізації.

Об'єкт дослідження – процес розробки вебзастосунку для управління інтернет-магазином.

Предмет дослідження – технології та методи розробки функціонального вебзастосунку.

Метою кваліфікаційної роботи є розробка функціонального та зручного у використанні вебзастосунку для управління інтернет-магазином екотоварів, що забезпечує автоматизацію та оптимізацію ключових бізнес-процесів.

Пояснювальна записка складається зі вступу, трьох розділів, висновків та додатків.

У першому розділі проведено аналіз предметної області продажу екотоварів, огляд та аналіз існуючих систем управління, визначено мету та цільову аудиторію застосунку, окреслено основні вимоги до розробки.

У другому розділі розроблено архітектуру та структуру вебзастосунку, спроектовано модель бази даних, обрано та обґрунтовано технології та інструменти розробки.

У третьому розділі реалізовано серверну частину вебзастосунку на мові програмування PHP, розроблено клієнтську частину з використанням HTML, CSS та фреймворку Bootstrap, проведено тестування вебзастосунку для перевірки коректності роботи всіх функцій.

В результаті створено вебзастосунок для управління інтернет-магазином екотоварів, який забезпечує автоматизацію та оптимізацію бізнес-процесів, зручний інтерфейс та високу продуктивність.

Кваліфікаційна робота містить 91 сторінку, 28 рисунків, 10 таблиць, 38 використаних джерел та 5 додатків.

Ключові слова: вебзастосунок, екотовари, управління інтернет-магазином, електронна комерція, автоматизація бізнес-процесів, PHP, MySQL.

ABSTRACT

**to the qualification work by the student of group 402 of
Petro Mohyla Black Sea National University
Bohza Eugene
«Web application for managing an online eco-goods store»**

The relevance of the topic. The growing interest in eco-friendly products makes the development of web applications for managing online eco-goods stores particularly relevant. The demand for natural, organic products and goods with minimal environmental impact stimulates the creation of functional web platforms for their sale.

The object of study – process of developing a web application for managing an online eco-goods store.

Subject of study – technologies and methods of developing a web application.

The purpose of the qualification work is the development of a functional and easy-to-use web application for managing an eco-goods online store, which provides automation and optimization of key business processes.

The explanatory note consists of an introduction, three chapters, conclusion, and appendices.

In the first chapter, the analysis of the eco-goods sales domain, an overview and analysis of existing management systems, the determination of the application's goal and target audience, and the outlining of the main development requirements were conducted.

In the second chapter, the architecture and structure of the web application were developed, the database model was designed, and the technologies and development tools were selected and justified.

In the third chapter, the server-side part of the web application was implemented using PHP, the client-side part was developed using HTML, CSS, and the Bootstrap framework, and the application was tested to verify the correctness of all functions.

As a result, a web application for managing an online eco-goods store was created, providing business process automation and optimization, a user-friendly interface, and high performance.

The qualification work contains 91 pages, 28 figures, 10 tables, 38 references, and 5 appendices.

Keywords: web application, eco-goods, online store management, e-commerce, business process automation, PHP, MySQL.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ	7
1.1 Аналіз сфери продажу екотоварів та особливостей вебзастосунків	7
1.2 Огляд та аналіз існуючих вебзастосунків/систем управління	10
1.3 Постановка задачі на розробку вебзастосунку	20
Висновки до розділу 1	24
2 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ	27
2.1 Розробка структури та функціоналу вебзастосунку	27
2.2 Проектування моделі бази даних	31
2.3 Розробка дизайну користувацького інтерфейсу	37
Висновки до розділу 2	40
3 РОЗРОБКА ВЕБЗАСТОСУНКУ	42
3.1 Вибір та обґрунтування технологій розробки	42
3.2 Реалізація серверної частини вебзастосунку	48
3.3 Розробка інтерфейсу	52
3.4 Тестування вебзастосунку	66
Висновки до розділу 3	70
ВИСНОВКИ.....	72
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	75
ДОДАТОК А Програмний код styles.ccs	78
ДОДАТОК Б Програмний код header.php	81
ДОДАТОК В Програмний код footer.php	83
ДОДАТОК Г Програмний код index.php	84
ДОДАТОК Д Програмний код functions.php.....	86

ПЕРЕЛІК СКОРОЧЕНЬ

БД	–	База даних
ПЗ	–	Програмне Забезпечення
CAGR	–	Compound Annual Growth Rate
MySQL	–	My Structured Query Language
CMS	–	Content Management System
CRM	–	Customer Relationship Management
ERP	–	Enterprise Resource Planning
SEO	–	Search Engine Optimization
API	–	Application Programming Interface
XSS	–	Cross-Site Scripting
IDE	–	Integrated Development Environment
MVC	–	Model-View-Controller
AJAX	–	Asynchronous JavaScript and XML
DOM	–	Document Object Model
HTTPS	–	HyperText Transfer Protocol Secure

ВСТУП

Актуальність теми. В сучасних умовах стрімкого розвитку електронної комерції та зростаючого інтересу споживачів до екологічно чистих товарів, розробка вебзастосунків для продажу екотоварів та систем їх управління набуває особливої актуальності. Все більше людей прагнуть вести здоровий та екологічний спосіб життя, що стимулює попит на натуральні, органічні продукти та товари, виготовлені з мінімальним негативним впливом на навколишнє середовище. Відповідно, створення зручних та функціональних вебплатформ для реалізації екотоварів є перспективним напрямком розвитку електронної комерції.

Важливо відзначити, що вебзастосунки для продажу екотоварів повинні не лише забезпечувати зручність здійснення покупок для клієнтів, але й надавати вичерпну інформацію про походження, склад та екологічність представлених товарів. Це вимагає розробки спеціалізованих функцій та інтеграції з базами даних екосертифікатів, що підтверджують якість та безпечність продукції. Таким чином, розробка вебзастосунку з продажу екотоварів є комплексною задачею, що потребує застосування сучасних технологій вебпрограмування та врахування специфічних вимог даної предметної області.

Об'єкт дослідження – процес розробки вебзастосунку для управління інтернет-магазином.

Предмет дослідження – технології та методи розробки функціонального вебзастосунку.

Метою даної кваліфікаційної роботи є розробка функціонального та зручного у використанні вебзастосунку для управління інтернет-магазином екотоварів, що забезпечує автоматизацію та оптимізацію ключових бізнес-процесів.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1) проаналізувати предметну область продажу екотоварів, визначити особливості та вимоги до вебзастосунків для управління інтернет-магазином;

- 2) провести огляд та аналіз існуючих систем управління, визначити їх переваги та недоліки;
- 3) здійснити постановку задачі на розробку вебзастосунку, сформулювати вимоги до функціоналу та дизайну;
- 4) розробити архітектуру та структуру проєкту, спроектувати модель бази даних;
- 5) обрати та обґрунтувати технології та інструменти розробки вебзастосунку (PHP, HTML, CSS, Bootstrap, MySQL);
- 6) реалізувати серверну частину вебзастосунку;
- 7) розробити клієнтську частину вебзастосунку;
- 8) провести тестування вебзастосунку, перевірити коректність роботи всіх функцій.

Аналіз останніх досліджень і публікацій. Питання розробки вебзастосунків для електронної комерції, в тому числі для продажу екотоварів, висвітлюються в працях таких науковців та практиків, як С. Мельник [1], О. Чубукова [2], В. Гужва [3] та ін. Автори розглядають особливості проєктування архітектури вебзастосунків, вибору технологій та інструментів розробки, забезпечення безпеки та оптимізації швидкодії. Проте, незважаючи на наявність теоретичних та практичних напрацювань у даній сфері, питання створення спеціалізованих вебзастосунків для продажу екотоварів потребує подальшого дослідження з урахуванням актуальних трендів ринку електронної комерції та вимог екологічно свідомих споживачів.

Методи дослідження. Для вирішення поставлених завдань були використані наступні методи: аналіз предметної області, порівняння, моделювання бізнес-процесів, проєктування архітектури, вебпрограмування, тестування програмного забезпечення.

Наукова новизна отриманих результатів полягає в розробці спеціалізованого вебзастосунку для управління інтернет-магазином екотоварів, що враховує

особливості даної предметної області та включає додаткові функції для інформування користувачів про екологічність представлених товарів.

Практичне значення отриманих результатів. Розроблений вебзастосунок може бути використаний реальними онлайн-магазинами екотоварів для організації продажів, залучення нових клієнтів та підвищення їх лояльності. Функції застосунку дозволяють надавати покупцям вичерпну інформацію про товари, спрощувати процес замовлення та оплати, забезпечувати високий рівень обслуговування.

Структура та обсяг роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, переліку джерел посилання (38 найменувань) та 5 додатків. Загальний обсяг роботи – 91 сторінка, з яких основна частина – 74 сторінки, перелік джерел посилання – 3 сторінки, додатки – 14 сторінок.

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз сфери продажу екотоварів та особливостей вебзастосунків

В останні роки спостерігається стрімке зростання популярності екотоварів, що зумовлено підвищенням екологічної свідомості споживачів та їх прагненням вести здоровий спосіб життя. Екотовари – це продукти, виготовлені з натуральних, органічних інгредієнтів без використання шкідливих хімічних речовин та з мінімальним негативним впливом на навколишнє середовище. До них відносяться органічні продукти харчування, натуральна косметика, екологічний одяг, предмети побуту з перероблених матеріалів тощо.

Ринок екотоварів демонструє стабільне зростання, що відкриває широкі перспективи для онлайн-продажів у цій сфері. За даними досліджень, обсяг світового ринку органічної продукції у 2021 році склав 129,2 млрд доларів США, а середньорічний темп росту (CAGR) у період з 2022 по 2027 рік прогнозується на рівні 14,3% [7]. Ця тенденція зумовлена низкою факторів, серед яких зростання обізнаності про переваги органічних продуктів для здоров'я, підвищення доходів населення в розвинених країнах, урядова підтримка виробництва органічної продукції та популяризація сталого способу життя.

Україна також демонструє позитивну динаміку розвитку ринку екотоварів. У 2020 році обсяг внутрішнього ринку органічної продукції в Україні оцінювався у 29,4 млн євро, що на 15% більше порівняно з попереднім роком [8]. За даними Федерації органічного руху України, площа сертифікованих органічних земель в Україні у 2020 році склала 462,5 тис. га, що робить нашу країну лідером у Європі за цим показником [9]. Все це свідчить про значний потенціал України як виробника та експортера органічної продукції.

В умовах пандемії COVID-19 та карантинних обмежень, електронна комерція набула особливого значення для реалізації екотоварів. Онлайн-продажі дозволяють виробникам та постачальникам екологічної продукції охопити ширшу аудиторію, забезпечити безперебійну роботу та доставку товарів до споживачів. За

даними досліджень, частка онлайн-продажів у загальному обсязі реалізації органічних продуктів у США зросла з 3% у 2019 році до 7% у 2020 році [4]. Схожі тенденції спостерігаються і на інших ринках, в тому числі в Україні.

Відповідно, розробка спеціалізованих вебзастосунків для продажу екотоварів стає важливим фактором успіху на цьому ринку. Такі застосунки дозволяють не лише здійснювати транзакції, але й надавати споживачам вичерпну інформацію про товари, їх екологічність та користь для здоров'я. Це особливо важливо в контексті екотоварів, адже довіра та обізнаність клієнтів є ключовими факторами при прийнятті рішення про покупку.

Вебзастосунки для продажу екотоварів мають низку особливостей та вимог, що відрізняють їх від традиційних інтернет-магазинів. Насамперед, вони повинні надавати детальну та достовірну інформацію про екологічність представлених товарів – склад, походження інгредієнтів, сертифікати якості, відсутність шкідливих домішок тощо. Це вимагає інтеграції з базами даних екосертифікатів, розробки спеціальних фільтрів та категорій товарів. Наприклад, вебзастосунок може дозволяти фільтрувати товари за такими критеріями, як "100% органічний", "веганський", "без глютену", "біорозкладне упакування" тощо.

Також важливою є наявність докладних описів товарів з акцентом на їх екологічні переваги. Покупці екотоварів хочуть знати не лише склад продукту, але й деталі його виробництва, вплив на довкілля, соціальну відповідальність виробника. Тому описи товарів повинні містити інформацію про методи органічного землеробства, відсутність пестицидів, гуманне ставлення до тварин, підтримку локальних фермерів, використання відновлюваної енергії на виробництві тощо.

Дизайн та контент вебзастосунків для продажу екотоварів також мають свої особливості. Вони повинні підкреслювати екологічність бренду та транслювати цінності сталого розвитку. Використання природних кольорів, мінімалістичного дизайну, зображень з природними текстурами – все це сприяє формуванню у відвідувачів відчуття єднання з природою та довіри до продуктів. При цьому,

дизайн має бути сучасним, естетичним та зручним у використанні, адже екоосвідомість не означає відмову від функціональності та стилю.

Контент вебзастосунку, окрім описів товарів, може включати також освітні матеріали про користь органічних продуктів, шкоду синтетичних речовин, проблеми забруднення довкілля, шляхи зменшення відходів тощо. Це не лише інформує клієнтів, але й формує спільноту однодумців навколо бренду. Інтеграція блогу, відео-контенту, інтерактивних елементів (наприклад, калькулятор екологічного сліду) підвищує залученість користувачів та їх лояльність до магазину.

Ще однією важливою функцією вебзастосунків для продажу екотоварів є надання рекомендацій щодо вибору екологічних альтернатив традиційним товарам, порад щодо відповідального споживання та утилізації відходів. Це може бути реалізовано у вигляді спеціальних міток для товарів ("замінює пластик", "багаторазового використання"), рекомендацій у картках товарів, окремих інформаційних розділів. Такий підхід не лише допомагає клієнтам зробити більш усвідомлений вибір, але й підкреслює експертність та відповідальність бренду.

Звичайно, як і будь-який інтернет-магазин, вебзастосунки для продажу екотоварів повинні забезпечувати зручність навігації, швидкість завантаження сторінок, безпечність здійснення транзакцій. Наявність зрозумілих фільтрів для пошуку товарів, можливість порівняння характеристик, відгуки та оцінки інших покупців – все це підвищує конверсію та стимулює здійснення покупок. Разом з тим, ці базові функції мають бути доповнені специфічними для екосфери можливостями, про які йшлося вище.

Підсумовуючи, можна сказати, що розробка вебзастосунків для продажу екотоварів є перспективним напрямком на стрімко зростаючому ринку екологічних продуктів. Такі застосунки повинні не лише забезпечувати зручність онлайн-покупок, але й надавати вичерпну інформацію про екологічність товарів, формувати довіру та лояльність клієнтів, просувати ідеї відповідального споживання.

Це вимагає комплексного підходу до проєктування та розробки, який включає ретельне опрацювання контенту, дизайну, користувацьких сценаріїв. Інтеграція з базами даних екосертифікатів, використання спеціальних фільтрів та категорій, надання рекомендацій та освітніх матеріалів – все це дозволяє створити унікальну цінність для клієнтів та виділитися серед конкурентів.

Врахування специфічних вимог та особливостей сфери екотоварів при розробці вебзастосунків є запорукою успіху на цьому перспективному ринку. Адже саме через поєднання функціональності, зручності користування, інформативності та відповідності цінностям екоsvідомого споживання можна створити дійсно затребуваний та корисний продукт. А це, в свою чергу, сприятиме популяризації відповідального ставлення до природи та власного здоров'я серед широкого кола користувачів.

1.2 Огляд та аналіз існуючих вебзастосунків/систем управління

На сьогоднішній день існує достатньо інтернет-магазинів зі своїми перевагами та недоліками. Тож розглянемо кілька таких популярних магазинів та зробимо висновки щодо того, яким має бути вебзастосунок з управління такими інтернет-магазинами.

1. Еко-Лавка (ekolavka.ua) – один з найвідоміших українських онлайн-магазинів екологічних товарів, що працює з 2014 року. Він пропонує широкий асортимент органічних продуктів харчування, косметики, засобів догляду, товарів для дому та дітей (рис. 1.1).

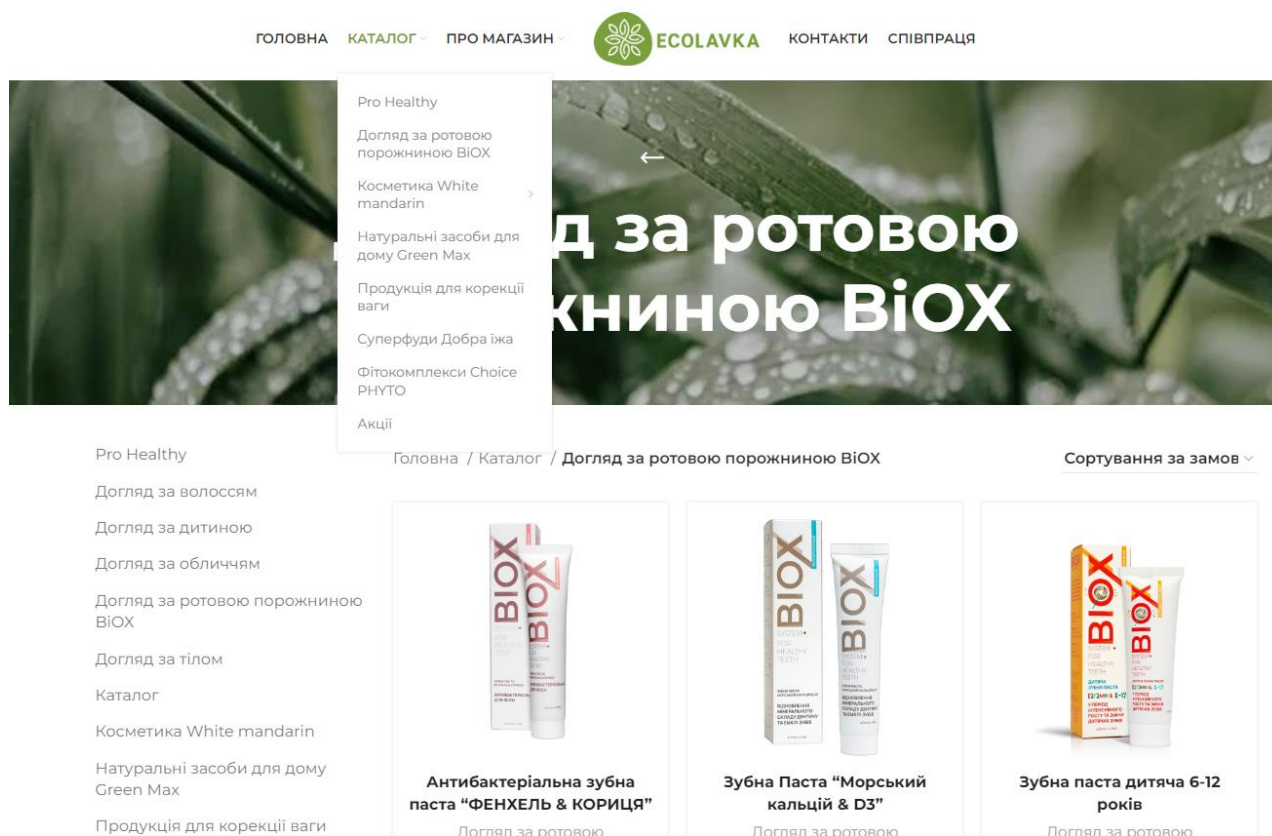


Рисунок 1.1 – Зручна навігація сайту Еко-Лавка

Переваги сайту Еко-Лавка:

- зручна навігація по категоріях товарів;
- детальна інформація про склад та сертифікацію продуктів;
- можливість фільтрації товарів за різними критеріями (виробник, ціна, призначення);
- швидка доставка по всій Україні;
- бонусна програма лояльності для постійних клієнтів;
- блог з корисними статтями про здоровий спосіб життя та екотренди.

Недоліки сайту Еко-Лавка:

- обмежений функціонал для управління бізнес-процесами компанії;
- відсутність інструментів для аналітики продажів та поведінки клієнтів;
- немає можливості інтеграції з зовнішніми системами та сервісами.

2. iHerb (iherb.com) – один з найбільших міжнародних онлайн-магазинів, що спеціалізується на продажу натуральних продуктів для здоров'я та краси (рис. 1.2).

Крім екотоварів, він пропонує також спортивне харчування, дієтичні добавки, товари для дітей та домашніх тварин.

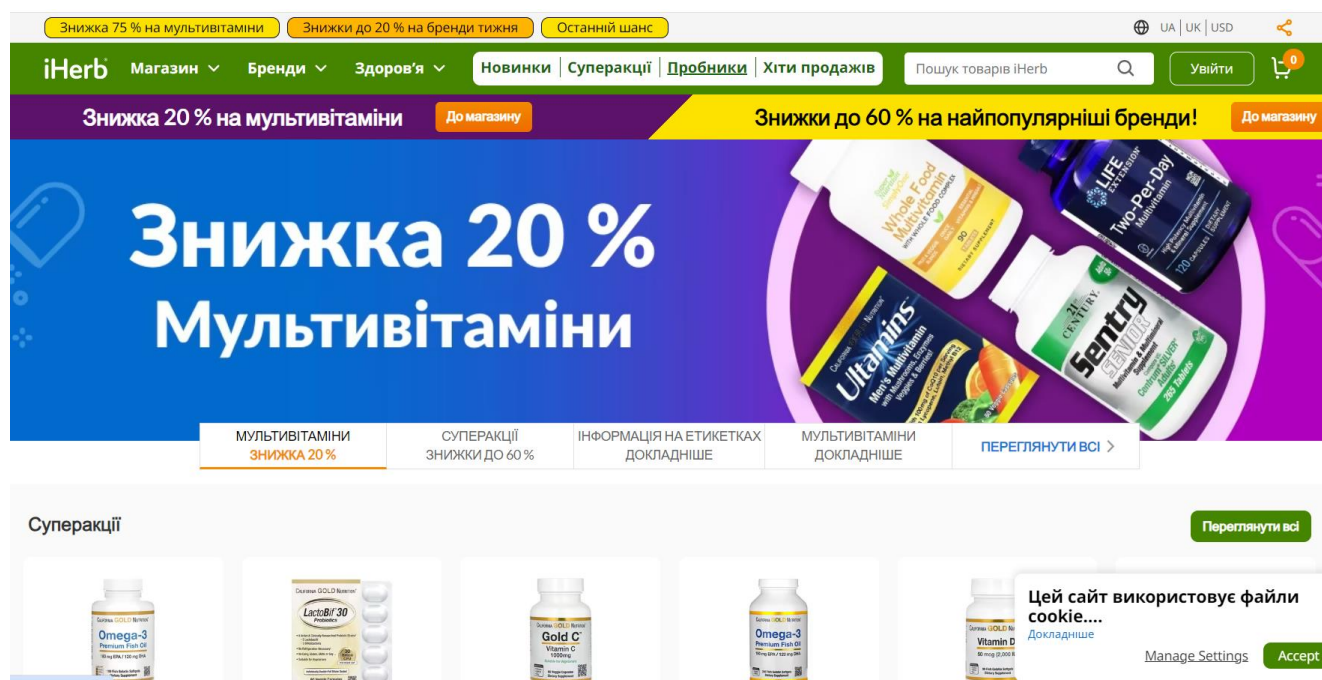


Рисунок 1.2 – Банер знижок на сайті iHerb

iHerb має наступні плюси:

- величезний вибір товарів від різних брендів (понад 30 тисяч найменувань);
- конкурентні ціни та регулярні знижки;
- безкоштовна доставка при замовленні від певної суми;
- локалізація сайту на різні мови, в тому числі українську;
- наявність мобільного додатку;
- відгуки та оцінки клієнтів для кожного товару;
- інформативні описи продуктів з переліком інгредієнтів, дозуванням, протипоказаннями.

iHerb має такі мінуси:

- відсутність функціоналу для управління складськими запасами та ціноутворенням;
- немає інструментів для аналізу ефективності маркетингових кампаній;

- складна процедура реєстрації та верифікації для нових клієнтів.

3. Organic&Natural (organic-natural.ua) – український вебзастосунок з продажу сертифікованих органічних та натуральних продуктів харчування, косметики, товарів для дому (рис. 1.3).

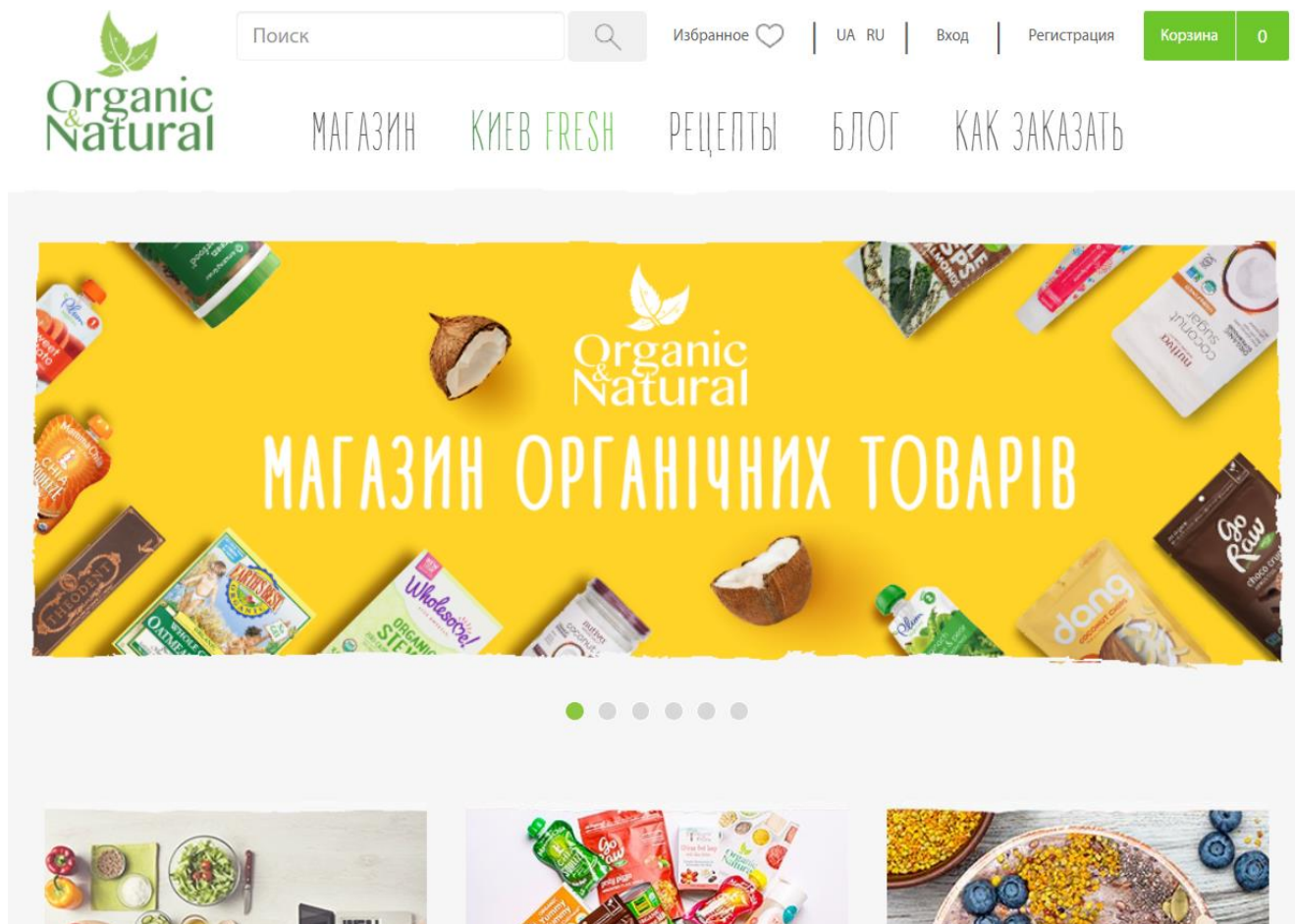


Рисунок 1.3 – Стильний дизайн Organic&Natural

Organic&Natural забезпечений нижченаведеними сильними сторонами:

- стильний мінімалістичний дизайн в еко стилі;
- зручні фільтри підбору товарів за різними параметрами;
- розділ з рецептами органічних страв;
- докладні описи кожного товару з інформацією про виробника та органічну сертифікацію;
- програма лояльності з накопичувальними балами;
- швидка доставка кур'єром або самовивіз з офлайн-магазину в Києві.

Organic&Natural має наступні недоліки:

- обмежені можливості для масштабування бізнесу та управління великою кількістю товарів;
- відсутність функціоналу для управління замовленнями та взаємодії з клієнтами;
- немає інтеграції з системами обліку та автоматизації бізнес-процесів.

4. Ecosclub (ecosclub.ua) позиціонує себе як онлайн-спільнота екологічно відповідальних споживачів. Окрім інтернет-магазину екотоварів, сайт містить багато інформаційних та навчальних матеріалів про сталий спосіб життя (рис. 1.4).

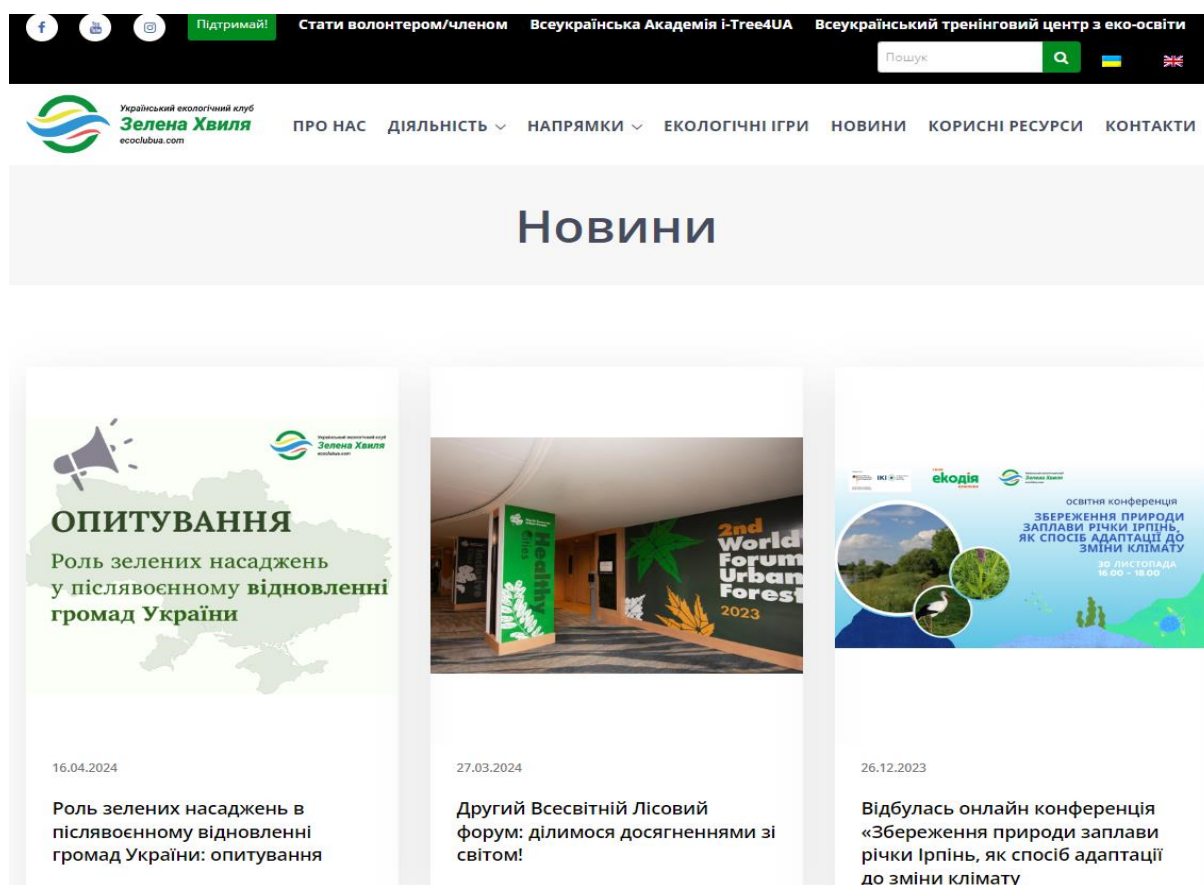


Рисунок 1.4 – Велика кількість статей, новин на сайті Ecosclub

Ecosclub забезпечений цими позитивними аспектами:

- величезна база корисних статей, відео, вебінарів на екологічну тематику;
- зручний інтерфейс з чітким поділом на категорії товарів;
- консультації експертів щодо підбору екотоварів;

- активна спільнота користувачів, можливість спілкування та обміну досвідом;
- прозора інформація про сертифікацію продукції, пояснення екомаркувань;
- регулярні акції та спеціальні пропозиції для членів клубу.

Слабкі сторони Ecosclub такі:

- відсутність повноцінного функціоналу для управління продажами та маркетингом;
- немає можливості налаштування бізнес-процесів під потреби конкретної компанії;
- обмежена масштабованість та гнучкість платформи.

Підсумовуючи, можна сказати, що ринок вебзастосунків з продажу екотоварів активно розвивається та має великий потенціал. Врахування кращих практик існуючих гравців та впровадження інноваційних підходів дозволить створити затребуваний та успішний застосунок в цій перспективній ніші. Далі детальніше розглянемо кілька прикладів систем керування магазинами.

На сьогоднішній день існуючі вебзастосунки з продажу екотоварів використовують різні системи керування продажами. Ці системи забезпечують базовий функціонал для управління бізнес-процесами, такими як облік товарів, управління замовленнями, інтеграція з платіжними системами.

1. Еко-Лавка (ekolavka.ua) використовує власну розроблену систему управління продажами, що базується на CMS Opencart. Ця система дозволяє такий функціонал:

- керувати каталогом товарів, додавати та редагувати описи, характеристики, фото;
- формувати замовлення, відстежувати їх статус та оплату;
- налаштовувати доставку та способи оплати;
- формувати звіти та аналітику продажів;
- керувати акціями та знижками;

- налаштувати розсилки та повідомлення клієнтам.

У цієї системи можна виділити такі переваги:

- гнучкість налаштувань під специфіку екотоварів;
- інтеграція з системами бухгалтерського обліку та складського управління;
- наявність модулів для SEO-оптимізації, інтеграції з соцмережами, формування XML-фідів.

Але у зазначеній системи є і недоліки:

- необхідність самостійної технічної підтримки та доопрацювань системи;
- відсутність деяких функцій, що є в більш просунутих комерційних CMS.

2. iHerb (iherb.com) використовує комерційну систему управління продажами Oracle ATG Web Commerce (рис. 1.5). Це потужна платформа для управління великими обсягами онлайн-продажів, яка забезпечує наступний функціонал:

- персоналізація пропозицій та реклами на основі поведінки користувачів;
- розширені можливості промо-акцій, знижок, кросс-сейлу та апсейлу;
- потужна система пошуку та фільтрації товарів;
- масштабованість для обробки великої кількості транзакцій;
- багатомовність та мультивалютність;
- можливості A/B-тестування сторінок для підвищення конверсії.

Oracle ATG Web Commerce 10.2 Server Architecture

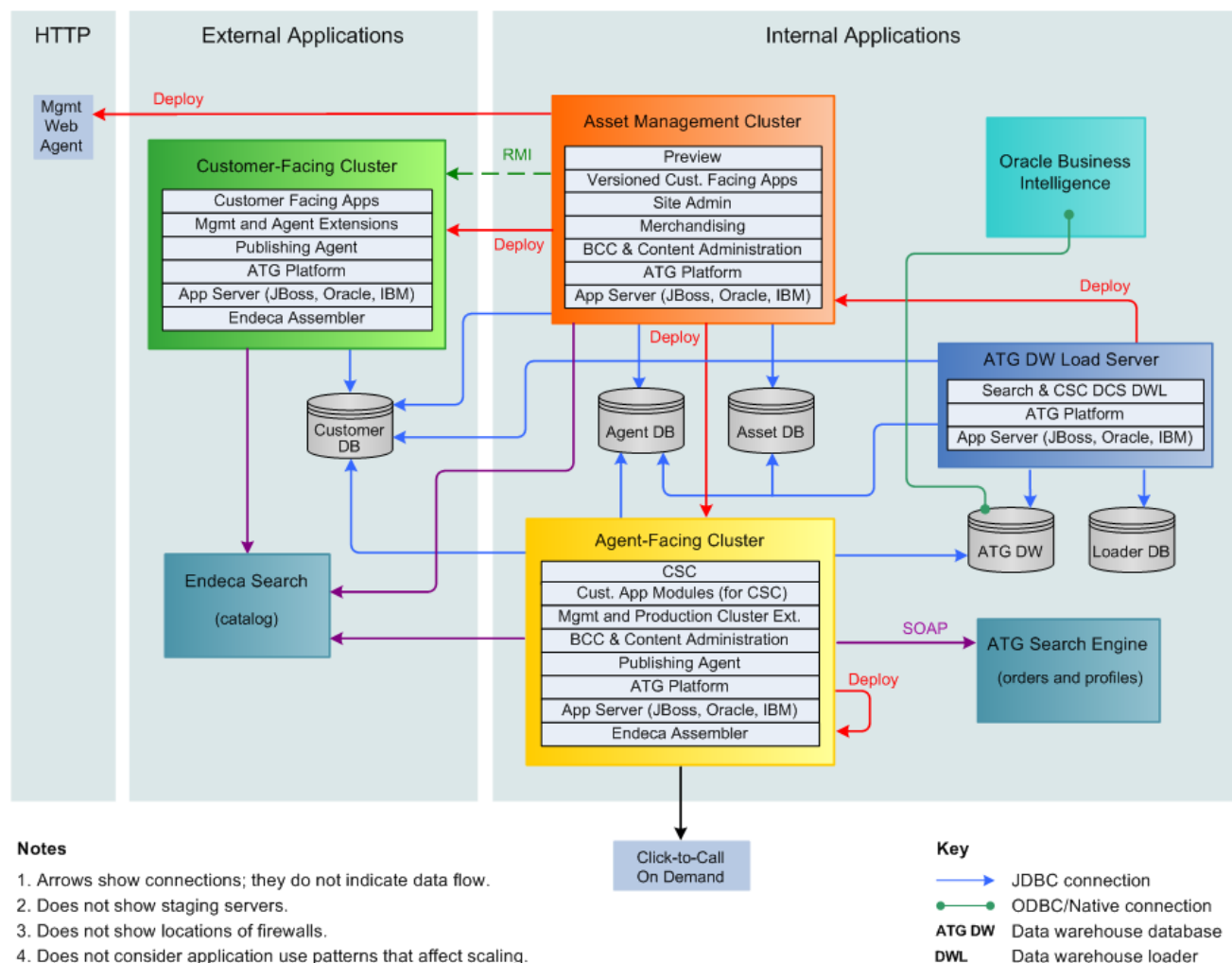


Рисунок 1.5 – Архітектура Oracle ATG Web Commerce

У зазначеній системі можна виділити наступні плюси:

- широкі можливості для маркетингу та персоналізації;
- надійність та продуктивність при високих навантаженнях;
- інтеграція з різними ERP, CRM, платіжними та логістичними системами.

Але у цій системі є свої мінуси:

- висока вартість ліцензії та впровадження;
- складність налаштування та освоєння функціоналу;
- надмірність функцій для невеликих та середніх магазинів.

3. Organic&Natural (organic-natural.ua) використовує популярну opensource CMS для інтернет-магазинів WooCommerce на базі WordPress (рис. 1.6). Це досить гнучка та зручна система, що дозволяє наступне застосування:

- швидко створити каталог товарів з різними варіаціями;
- керувати замовленнями та платежами;
- налаштовувати способи доставки та оплати;
- формувати звіти та аналізувати продажі;
- підключати різні плагіни та розширення для додаткових функцій.

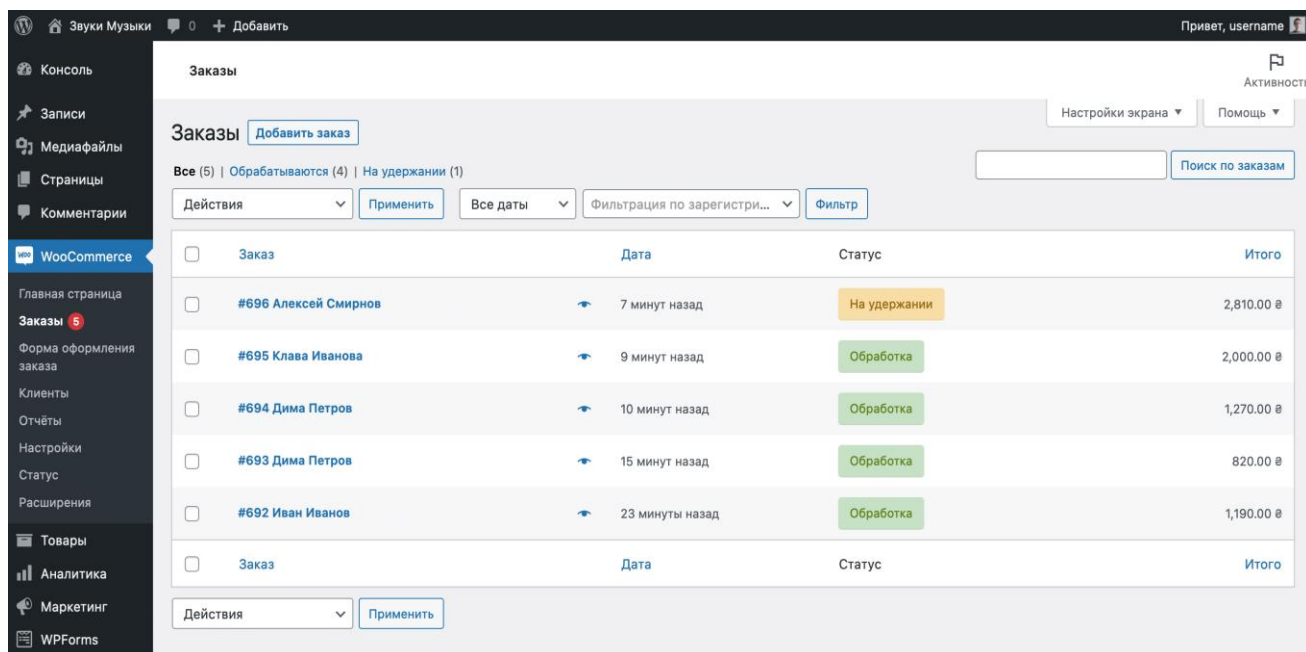


Рисунок 1.6 – Приклад панелі управління WooCommerce

Ця система має нижченаведені сильні сторони:

- безкоштовність базової версії та плагінів;
- простота освоєння та адміністрування;
- велика спільнота розробників та користувачів, багато готових рішень;
- інтеграція з українськими платіжними системами та службами доставки.

Слабкі сторони системи такі:

- обмежена масштабованість при значному зростанні кількості товарів та замовлень;
- необхідність самостійного налаштування та технічної підтримки;
- ризики безпеки через популярність системи у зловмисників.

4. Ecoclub (ecoclub.ua) Ecoclub використовує SaaS-платформу Shopify для управління продажами (рис. 1.7). Це хмарний сервіс, що дозволяє швидко

розгорнути інтернет-магазин без необхідності власної розробки та підтримки.

Основні можливості Shopify:

- конструктор сторінок з готовими дизайнами та шаблонами;
- управління каталогом товарів, замовленнями, клієнтами;
- вбудовані платіжні системи та інтеграції зі службами доставки;
- автоматичні податкові обчислення та генерація рахунків-фактур;
- базова аналітика продажів та відвідуваності магазину;
- додаткові додатки для розширення функціональності.

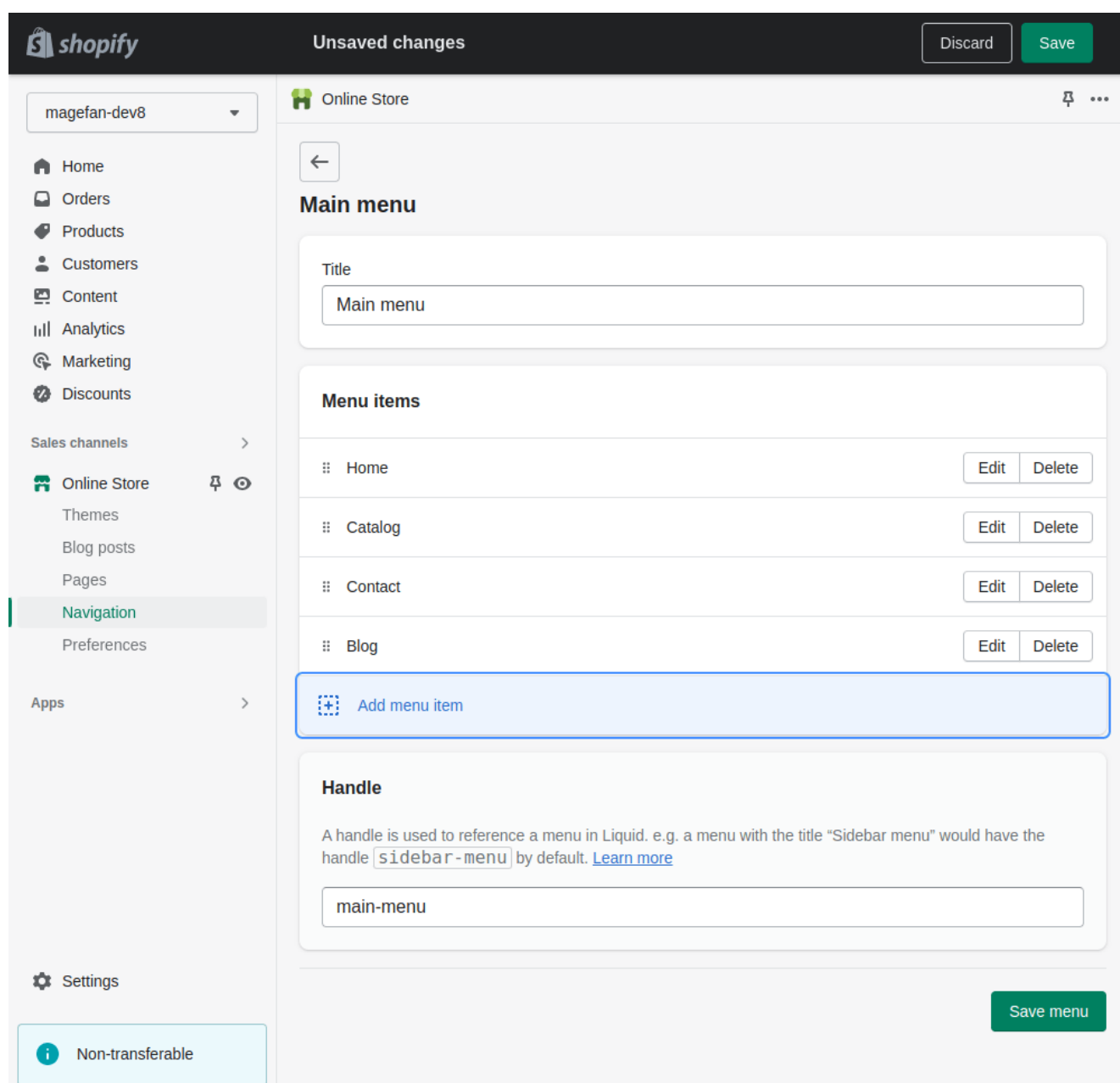


Рисунок 1.7 – Панель управління Shopify

Переваги системи наступні:

- швидкість та простота запуску онлайн-продажів;
- надійність та безпека завдяки хмарній інфраструктурі;
- цілодобова технічна підтримка;
- масштабованість при зростанні бізнесу.

Недоліки панелі управління такі:

- обмежені можливості кастомізації дизайну та функцій;
- прив'язка до екосистеми Shopify, складність міграції на інші платформи;
- додаткові витрати на оренду сервісу та розширення функціоналу.

Підсумовуючи, можна сказати, що вибір системи управління продажами для інтернет-магазину екотоварів залежить від масштабу бізнесу, бюджету на розробку та підтримку, необхідного функціоналу. Для невеликих та нішевих проєктів підійдуть гнучкі opensource-рішення як Opencart та WooCommerce. Для середнього бізнесу варто розглянути SaaS-платформи на кшталт Shopify. А для великих інтернет-магазинів з високим навантаженням та потребами в персоналізації можуть бути виправдані комерційні системи рівня Oracle ATG Web Commerce [22].

Але в будь-якому випадку, система управління продажами повинна відповідати специфіці екотоварів, забезпечувати зручність адміністрування каталогу, оформлення замовлень, інтеграцію з маркетинговими інструментами. А також надавати необхідну аналітику для прийняття бізнес-рішень та оптимізації продажів.

1.3 Постановка задачі на розробку вебзастосунку

На основі проведеного аналізу предметної сфери та огляду існуючих рішень, можна сформулювати наступну постановку задачі на розробку вебзастосунку:

Мета розробки: створення функціонального та зручного у використанні вебзастосунку для управління інтернет-магазином екотоварів, що дозволить

автоматизувати та оптимізувати ключові бізнес-процеси, підвищити ефективність роботи персоналу та забезпечити високу якість обслуговування клієнтів.

Цільова аудиторія вебзастосунку буде наступна:

- власники та адміністратори інтернет-магазинів екотоварів;
- менеджери з продажу та маркетингу;
- співробітники складу та служби доставки;
- бухгалтери та фінансові менеджери.

Тепер можемо зробити перелік основних аспектів роботи системи, на який будемо надалі орієнтуватись увесь проєкт:

1) Управління каталогом екотоварів:

- додавання, редагування та видалення товарів;
- групування товарів за категоріями та колекціями;
- управління атрибутами та варіаціями товарів;
- масове завантаження та оновлення даних про товари.

2) Управління замовленнями:

- формування та обробка замовлень клієнтів;
- редагування та відстеження статусу замовлень;
- генерація рахунків та товарно-транспортних накладних;
- обробка повернень та рекламаций.

3) Управління клієнтами:

- збереження та редагування інформації про клієнтів;
- історія замовлень та взаємодії з клієнтами;
- сегментація клієнтів за різними критеріями;
- інтеграція з системами розсилки email та SMS.

4) Управління контентом:

- створення та редагування сторінок з описом товарів;
- управління блогом та публікація статей;
- додавання та оновлення промо-матеріалів (банери, відео тощо);
- SEO-оптимізація контенту для пошукових систем.

- 5) Маркетингові інструменти:
 - створення та управління акціями та знижками;
 - генерація промокодів та керування програмою лояльності;
 - інтеграція з системами вебаналітики (Google Analytics, Facebook Pixel);
 - відстеження ефективності рекламних кампаній.
 - 6) Управління складом та доставкою:
 - облік наявності товарів на складі;
 - резервування та списання товарів;
 - формування маршрутів доставки та відстеження відправлень;
 - інтеграція зі службами доставки (Нова Пошта, Укрпошта тощо).
 - 7) Фінансовий облік:
 - формування та відправка рахунків-фактур;
 - облік оплат та відстеження заборгованостей;
 - розрахунок комісій та податків;
 - інтеграція з системами онлайн-платежів та бухгалтерськими програмами.
 - 8) Звітність та аналітика:
 - формування звітів з продажів та фінансових показників;
 - аналітика поведінки відвідувачів на сайті;
 - розрахунок ключових показників ефективності (KPI) бізнесу;
 - візуалізація даних у вигляді графіків та діаграм.
 - 9) Адміністрування та безпека:
 - розмежування прав доступу для різних категорій користувачів;
 - аудит дій користувачів та ведення журналу змін;
 - резервне копіювання та відновлення даних;
 - забезпечення безпеки даних та захист від несанкціонованого доступу.
- Також можна зазначити наступні вимоги до інтерфейсу:
- інтуїтивно зрозумілий та user-friendly інтерфейс;

- адаптивний дизайн для коректного відображення на різних пристроях;
- швидке завантаження сторінок та оптимізація продуктивності;
- кастомізація інтерфейсу під фірмовий стиль інтернет-магазину;
- локалізація інтерфейсу українською та англійською мовами.

Вимоги до технологій при розробці будуть наступними:

- використання сучасного стеку вебтехнологій для розробки серверної та клієнтської частини;
- розробка на базі популярного PHP-фреймворку (Laravel, Symfony, CodeIgniter тощо);
- дотримання принципів SOLID, MVC та інших best practices веброботи;
- вибір СУБД MySQL або PostgreSQL для зберігання та обробки даних;
- забезпечення безпеки даних (захист від SQL-ін'єкцій, XSS-атак, використання HTTPS);
- оптимізація швидкодії та масштабованості вебпрограми;
- використання системи контролю версій (Git) для командної роботи над проектом;
- розгортання застосунку на надійному хостингу з можливістю масштабування.

Таким чином, розроблюваний вебзастосунок для управління інтернет-магазином екотоварів повинен забезпечувати автоматизацію та оптимізацію ключових бізнес-процесів, пов'язаних з онлайн-продажами. Він має надавати зручні інструменти для управління каталогом товарів, обробки замовлень, взаємодії з клієнтами, ведення складського та фінансового обліку, аналізу даних та прийняття управлінських рішень.

Використання сучасних вебтехнологій, дотримання кращих практик розробки, забезпечення безпеки даних та оптимізація продуктивності дозволять створити надійну та ефективну систему, готову до інтенсивної експлуатації в умовах реального бізнесу.

При розробці продукту особливу увагу слід приділити зручності використання, адаптивності дизайну під різні пристрої та швидкості роботи. Можливість кастомізації інтерфейсу під фірмовий стиль та локалізація різними мовами дозволять адаптувати під потреби конкретного інтернет-магазину та його цільової аудиторії.

В процесі розробки важливо забезпечити можливість інтеграції вебзастосунку з іншими системами та сервісами, такими як служби доставки, платіжні шлюзи, системи вебаналітики, облікові програми тощо. Це дозволить створити цілісну екосистему для ефективного управління інтернет-магазином екотоварів.

Реалізація поставлених вимог та використання передових практик веброзробки дозволить створити потужний та гнучкий інструмент для управління онлайн-бізнесом у сфері екотоварів. Вебзастосунок стане надійним фундаментом для зростання та розвитку інтернет-магазину, забезпечуючи високу ефективність операційних процесів та відмінний рівень обслуговування клієнтів.

Висновки до розділу 1

У першому розділі кваліфікаційної роботи було проведено ґрунтовний аналіз предметної сфери продажу екотоварів та особливостей вебзастосунків в цій галузі. Розглянуто поточний стан та перспективи розвитку ринку екологічних товарів, відзначено зростання попиту на органічну та натуральну продукцію серед споживачів. Це свідчить про актуальність та своєчасність розробки спеціалізованих систем для управління онлайн-продажами у даному сегменті.

Аналіз існуючих вебзастосунків з продажу екотоварів дозволив виявити їх сильні сторони:

- інформативність опису товарів;
- зручність навігації;
- наявність додаткових функцій (блог, рецепти, поради експертів).

Водночас, було відзначено і типові недоліки, зокрема:

- застарілий дизайн;
- обмежений асортимент;
- недостатню оптимізацію під мобільні пристрої.

Окрему увагу було приділено огляду систем управління продажами, що використовуються в розглянутих вебзастосунках. Проаналізовано переваги та недоліки різних підходів – від власних розробок на базі opensource CMS до використання комерційних платформ електронної комерції.

Такі системи мають наступні переваги:

- можливість налаштування системи під специфічні потреби бізнесу;
- забезпечують широкий спектр можливостей для управління продажами та маркетингом;
- висока продуктивність при роботі з великими обсягами даних і високим навантаженням;
- наявність технічної підтримки та регулярних оновлень.

До недоліків систем віднесемо наступне:

- ліцензійні витрати та витрати на впровадження можуть бути значними;
- потребує значних зусиль для налаштування та освоєння функціоналу;
- деякі системи можуть бути складно масштабувати при значному зростанні кількості товарів та замовлень;
- необхідність самостійної технічної підтримки та регулярних оновлень для opensource рішень.

Це дало розуміння та допомогло визначити ключові критерії та підходи для створення власної системи управління продажами, яка відповідатиме специфічним потребам проекту та забезпечуватиме максимальну ефективність роботи інтернет-магазину екотоварів.

На основі проведеного аналізу було сформульовано постановку задачі на розробку вебзастосунку. Визначено мету та цільову аудиторію застосунку, окреслено основні вимоги до функціональності, дизайну та технологій розробки.

Серед ключових завдань виділено:

- створення інформативного каталогу товарів;
- забезпечення зручних інструментів для оформлення та відстеження замовлень;
- інтеграцію з платіжними та логістичними сервісами;
- реалізацію функцій для залучення та утримання клієнтів.

Таким чином, у першому розділі закладено теоретичний фундамент для подальшої практичної розробки продукту. Визначено актуальність та затребуваність подібних рішень на ринку, проаналізовано досвід конкурентів, сформовано власне бачення оптимального функціоналу та дизайну. Це дозволяє перейти до наступного етапу – безпосереднього проектування архітектури та інтерфейсу вебзастосунку.

2 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ

2.1 Розробка структури та функціоналу вебзастосунку

Проаналізувавши предметну область та вимоги до системи, можна приступити до розробки її структури та функціоналу. Цей етап є ключовим у процесі створення застосунку, оскільки він закладає основу для подальшої технічної реалізації та визначає, наскільки зручним та ефективним буде користування системою.

Перш за все, необхідно визначити основні модулі та компоненти вебзастосунку. Базуючись на файлах, наданих у попередньому запиті, можна виділити такі ключові елементи системи:

1) головна сторінка (`index.php`): відправна точка для користувачів, де вони можуть ознайомитися з основною інформацією про магазин, побачити рекламні банери, акційні пропозиції тощо. Також на головній сторінці доцільно розмістити графік продажів за місяць, щоб наочно продемонструвати динаміку роботи магазину;

2) каталог товарів (`products.php`): один з центральних модулів застосунку, де відображається перелік усіх доступних екотоварів. Для зручності користувачів товари мають бути структуровані за категоріями та підкатегоріями, з можливістю фільтрації та сортування за різними параметрами (ціна, популярність, новизна тощо). При переході на сторінку конкретного товару (`details_product.php`) користувач повинен бачити детальну інформацію про нього: опис, фото, характеристики, відгуки покупців, рекомендовані товари тощо;

3) модуль роботи з категоріями (`edit_categories.php`): для ефективного управління асортиментом магазину менеджерам необхідний зручний інструмент для роботи з категоріями товарів. Цей модуль повинен надавати можливості створення, редагування та видалення категорій і підкатегорій, а також управління їх ієрархічною структурою;

4) модуль роботи з характеристиками товарів (`edit_characteristics.php`, `edit_units.php`): кожен товар має набір властивостей та характеристик, за якими його можна описати та порівняти з іншими товарами. Для цього в системі передбачені окремі сторінки для управління характеристиками та одиницями їх виміру. Менеджери магазину повинні мати можливість гнучко налаштувати перелік характеристик під кожну категорію товарів;

5) модуль роботи з виробниками (`manufacturers.php`): оскільки магазин спеціалізується на екотоварах, дуже важливо надавати покупцям інформацію про виробників представленої продукції. На сторінці списку виробників повинна бути можливість переглянути інформацію про кожного з них, а також додати, відредагувати або видалити виробника;

6) модуль роботи з клієнтами (`edit_client.php`): для ефективної комунікації з покупцями та побудови довгострокових відносин необхідно мати базу даних клієнтів магазину. Цей модуль повинен надавати функціонал для додавання нових клієнтів, редагування їх даних, перегляду історії замовлень тощо;

7) модуль роботи з замовленнями (`orders.php`): ще один ключовий компонент системи, який відповідає за обробку замовлень покупців. Менеджери магазину повинні мати можливість переглядати список усіх замовлень, змінювати їх статус, додавати коментарі тощо. Також цей модуль має бути інтегрований з системою обліку товарів, щоб автоматично списувати продані товари зі складу;

8) модуль роботи з екосертифікатами (`eco_certificates.php`): відповідає за управління екологічними сертифікатами, які асоціюються з товарами в інтернет-магазині екотоварів. Цей модуль забезпечує функціонал для додавання, редагування та видалення сертифікатів, а також їх перегляд. Це дозволяє менеджерам магазину гнучко керувати екологічними сертифікатами та підтримувати їх актуальність;

9) адмін-панель (`admin.php`): для керування всіма аспектами роботи магазину розробляється окремий розділ – адмін-панель. Тут адміністратори системи можуть налаштувати глобальні параметри магазину, управляти

користувачами та їх правами доступу, переглядати статистику продажів та інші звіти.

Окрім основних модулів, у файлі `functions.php` можна побачити додатковий функціонал, який забезпечує наступну роботу системи:

- функції для роботи з користувачами: авторизація, редагування профілю тощо. Вони дозволяють розмежувати доступ до різних частин системи та персоналізувати досвід користувачів;

- функції для роботи з товарами, категоріями, виробниками: додавання, редагування, видалення записів у відповідних таблицях бази даних. Вони інкапсулюють роботу з БД і дозволяють уніфікувати операції з даними в різних модулях;

- функції для роботи з замовленнями та клієнтами: створення нових замовлень, зміна їх статусу, додавання клієнтів у базу тощо;

- функції сортування: дозволяє впорядковувати товари за різними критеріями, такими як назва, вартість, наявність, забезпечуючи швидкий доступ до необхідної інформації;

- функції фільтрування: допомагає фільтрувати товари за параметрами, такими як матеріал, упаковка, місце походження, термін служби, звужує результати пошуку до релевантних товарів;

- допоміжні функції: наприклад, функція для відображення графіку продажів за місяць, яка використовується на головній сторінці.

Варто відзначити, що ключовим аспектом при розробці структури вебзастосунку для управління бізнесом є забезпечення зручної та інтуїтивно зрозумілої навігації між сторінками. Користувачі, такі як менеджери та адміністратори, повинні легко орієнтуватися в системі та швидко знаходити потрібну інформацію. Для цього доцільно використовувати різні UX-патерни: меню категорій, хлібні крихти, пошук, фільтри тощо [4].

Не менш важливим є адаптивний дизайн застосунку, який забезпечує коректне відображення контенту на різних пристроях: комп'ютерах, планшетах,

смартфонах. Сьогодні більшість користувачів заходять у системи управління бізнесом зі своїх мобільних девайсів, тому адаптивність напряму впливає на зручність використання та продуктивність роботи.

Також при розробці структури застосунку можна передбачити інтеграцію зі сторонніми сервісами та системами, які є важливими для автоматизації бізнес-процесів, наприклад:

- інтеграція з поштовими сервісами для автоматичної відправки транзакційних листів (підтвердження замовлення, зміна статусу тощо);
- підключення сервісів доставки для автоматичного розрахунку вартості доставки та відстеження посилок;
- інтеграція з системами обліку товарів на складі для синхронізації даних про наявність та резервування товарів;
- інтеграція з системами аналітики для аналізу ефективності маркетингових кампаній та поведінки клієнтів.

Всі ці зовнішні сервіси потрібно врахувати при проектуванні архітектури застосунку та передбачити відповідні API для обміну даними.

Оскільки магазин екотоварів регулярно оновлює свій асортимент, додає нові категорії та виробників, необхідно надати менеджерам зручні інструменти для управління контентом системи. При цьому система повинна бути захищеною від неавторизованого доступу, SQL-ін'єкцій, XSS-атак та інших типових вразливостей.

Підсумовуючи, можна сказати, що розробка структури та функціоналу вебзастосунку для управління інтернет-магазином екотоварів – це комплексний процес, який вимагає врахування багатьох факторів: зручності користування, адаптивності дизайну, безпеки даних, інтеграції зі сторонніми сервісами тощо. Завдяки ретельному плануванню та проектуванню архітектури застосунку можна створити надійну та ефективну систему, яка дозволить бізнесу успішно розвиватися та масштабуватися.

2.2 Проєктування моделі бази даних

Для реалізації застосунку було спроектовано базу даних, яка складається з декількох взаємопов'язаних таблиць. Ця база даних дозволяє зберігати інформацію про товари та послуги, контрагентів, угоди, фінансові операції, користувачів системи та співробітників компанії. Розглянемо детальніше структуру та призначення кожної таблиці [33].

Таблиця "products_and_services" призначена для зберігання інформації про товари та послуги, які пропонує компанія (табл. 2.1). Кожен запис у цій таблиці містить унікальний ідентифікатор (id), назву (name), опис (description), тип (type) – товар або послуга, ціну (price), кількість на складі (quantity_in_stock), матеріал виготовлення (material), тип упаковки (packaging), походження товару (place_of_origin) та термін його дії/служби (lifecycle). Ця таблиця є центральною в базі даних, оскільки вона пов'язана з таблицями угод та фінансових операцій.

Таблиця 2.1 – Структура таблиці "products_and_services"

Колонка	Тип даних	Опис
id	int(11)	Унікальний ідентифікатор товару або послуги (PRIMARY KEY)
name	varchar(255)	Назва товару або послуги
description	text	Опис товару або послуги
type	enum	Тип: 'Product' (товар) або 'Service' (послуга)
price	decimal(10,2)	Ціна товару або послуги
quantity_in_stock	int(11)	Кількість товару на складі
material	varchar(255)	Матеріал, з якого виготовлений товар
packaging	varchar(255)	Тип упаковки, в якій товар продається
place_of_origin	varchar(255)	Походження товару
lifecycle	varchar(50)	Термін служби товару

Таблиця "contractors" зберігає дані про контрагентів компанії – постачальників та клієнтів (табл. 2.2). Кожен запис містить унікальний ідентифікатор (id), назву (name), коментар (comment) та тип контрагента (type) – юридична або фізична особа. Ця таблиця пов'язана з таблицями угод та фінансових операцій, що дозволяє відстежувати взаємодію з контрагентами.

Таблиця 2.2 – Структура таблиці "contractors"

Колонка	Тип даних	Опис
id	int(11)	Унікальний ідентифікатор контрагента (PRIMARY KEY)
name	varchar(255)	Назва контрагента
comment	text	Коментар щодо контрагента
type	enum	Тип контрагента: 'legal' (юридична особа) або 'individual' (фізична особа)

Таблиця "deals" призначена для зберігання інформації про угоди, укладені між компанією та контрагентами (табл. 2.3). Кожен запис містить унікальний ідентифікатор (id), дату угоди (deal_date), коментар (comment), ідентифікатор контрагента (contractor_id), ідентифікатор користувача, який уклав угоду (user_id), та загальну суму угоди (total_amount). Ця таблиця пов'язана з таблицями контрагентів, користувачів та фінансових операцій.

Таблиця 2.3 – Структура таблиці "deals"

Колонка	Тип даних	Опис
id	int(11)	Унікальний ідентифікатор угоди (PRIMARY KEY)
deal_date	date	Дата укладення угоди
comment	text	Коментар щодо угоди

Закінчення таблиці 2.3

Колонка	Тип даних	Опис
contractor_id	int(11)	Ідентифікатор контрагента (FOREIGN KEY до таблиці "contractors")
user_id	int(11)	Ідентифікатор користувача (FOREIGN KEY до таблиці "users")
total_amount	decimal(10,2)	Загальна сума угоди

Таблиця "deals_products_services" є зв'язуючою таблицею між таблицями "deals" та "products_and_services" (табл. 2.4). Вона зберігає інформацію про те, які товари або послуги були включені до кожної угоди та в якій кількості. Кожен запис містить ідентифікатор угоди (deal_id), ідентифікатор товару або послуги (product_service_id) та кількість (quantity).

Таблиця 2.4 – Структура таблиці "deals_products_services"

Колонка	Тип даних	Опис
deal_id	int(11)	Ідентифікатор угоди (PRIMARY KEY, FOREIGN KEY до таблиці "deals")
product_service_id	int(11)	Ідентифікатор товару або послуги (PRIMARY KEY, FOREIGN KEY до таблиці "products_and_services")
quantity	int(11)	Кількість товару або послуги в угоді

Таблиця "financial_operations" призначена для зберігання інформації про фінансові операції компанії (табл. 2.5). Кожен запис містить унікальний ідентифікатор (id), дату операції (operation_date), ідентифікатор пов'язаної угоди (deal_id), ідентифікатор контрагента (contractor_id), тип операції (operation_type) – надходження або витрата, коментар (comment) та суму операції (amount).

Таблиця 2.5 – Структура таблиці "financial_operations"

Колонка	Тип даних	Опис
id	int(11)	Унікальний ідентифікатор фінансової операції (PRIMARY KEY)
operation_date	date	Дата здійснення фінансової операції
deal_id	int(11)	Ідентифікатор пов'язаної угоди (FOREIGN KEY до таблиці "deals")
contractor_id	int(11)	Ідентифікатор контрагента (FOREIGN KEY до таблиці "contractors")
operation_type	enum	Тип операції: 'Income' (надходження) або 'Expense' (витрата)
comment	text	Коментар щодо фінансової операції
amount	decimal(10,2)	Сума фінансової операції

Таблиця "company_account" призначена для зберігання інформації про баланс компанії (табл. 2.6). Вона містить лише один запис з унікальним ідентифікатором (id) та полем балансу (balance).

Таблиця 2.6 – Структура таблиці "company_account"

Колонка	Тип даних	Опис
id	int(11)	Унікальний ідентифікатор запису (PRIMARY KEY)
balance	decimal(15,2)	Баланс компанії

Таблиця "chat_messages" зберігає повідомлення чату між користувачами системи (табл. 2.7). Кожен запис містить унікальний ідентифікатор (id), ідентифікатор користувача, який відправив повідомлення (user_id), текст повідомлення (message), час виконання задачі (task_time) та мітку часу створення повідомлення (timestamp).

Таблиця 2.7 – Структура таблиці "chat_messages"

Колонка	Тип даних	Опис
id	int(11)	Унікальний ідентифікатор повідомлення (PRIMARY KEY)
user_id	int(11)	Ідентифікатор користувача, який відправив повідомлення
message	text	Текст повідомлення
task_time	int(11)	Час виконання задачі
timestamp	timestamp	Мітка часу створення повідомлення (DEFAULT CURRENT_TIMESTAMP)

Таблиця "eco_certificates" призначена для зберігання інформації про екосертифікати (табл. 2.8). Кожен запис у цій таблиці містить унікальний ідентифікатор (id), назву сертифіката (name) та його опис (description). Ця таблиця дозволяє менеджерам магазину додавати, редагувати та видаляти сертифікати, що забезпечує гнучкість у управлінні екологічною інформацією про товари.

Таблиця 2.8 – Структура таблиці "eco_certificates"

Колонка	Тип даних	Опис
id	int(11)	Унікальний ідентифікатор сертифіката (PRIMARY KEY)
name	varchar(255)	Назва сертифіката
description	text	Опис сертифіката

Таблиця "product_certificates" є зв'язуючою таблицею між таблицями "products_and_services" та "eco_certificates" (табл. 2.9). Вона зберігає інформацію про те, які сертифікати були присвоєні кожному товару або послугі. Кожен запис містить ідентифікатор товару або послуги (product_id) та ідентифікатор сертифіката (certificate_id).

Таблиця 2.9 – Структура таблиці "product_certificates"

Колонка	Тип даних	Опис
product_id	int(11)	Ідентифікатор товару або послуги (FOREIGN KEY до таблиці "products_and_services")
certificate_id	int(11)	Ідентифікатор сертифіката (FOREIGN KEY до таблиці "eco_certificates")

На діаграмі бази даних зображено взаємозв'язки між таблицями (рис. 2.1). contractors має зв'язок один-до-багатьох з financial_operations і deals. Це дає змогу відстежувати всі фінансові операції, пов'язані з конкретним контрагентом та всі угоди, в яких бере участь конкретний контрагент. deals має зв'язок один-до-багатьох з financial_operations і deals_products_services. Завдяки цьому можна відстежувати фінансові операції, що стосуються конкретної угоди і пов'язувати угоди з конкретними продуктами та послугами. users має зв'язок один-до-багатьох з deals, через це ми можемо відстежувати всі угоди, в яких бере участь конкретний користувач. products_and_services має зв'язок один-до-багатьох з deals_products_services і product_certificates. Це дає можливість визначати, які продукти чи послуги входять до складу конкретної угоди та пов'язувати продукти з відповідними екосертифікатами. eco_certificates має зв'язок один-до-багатьох з product_certificates, що забезпечує відстеження того, які продукти мають конкретний екосертифікат.

Така структура бази даних дозволяє ефективно зберігати та обробляти дані, необхідні для функціонування проєкту. Вона забезпечує цілісність даних, мінімізує дублювання інформації та дозволяє легко виконувати операції вибірки, фільтрації та агрегації даних.

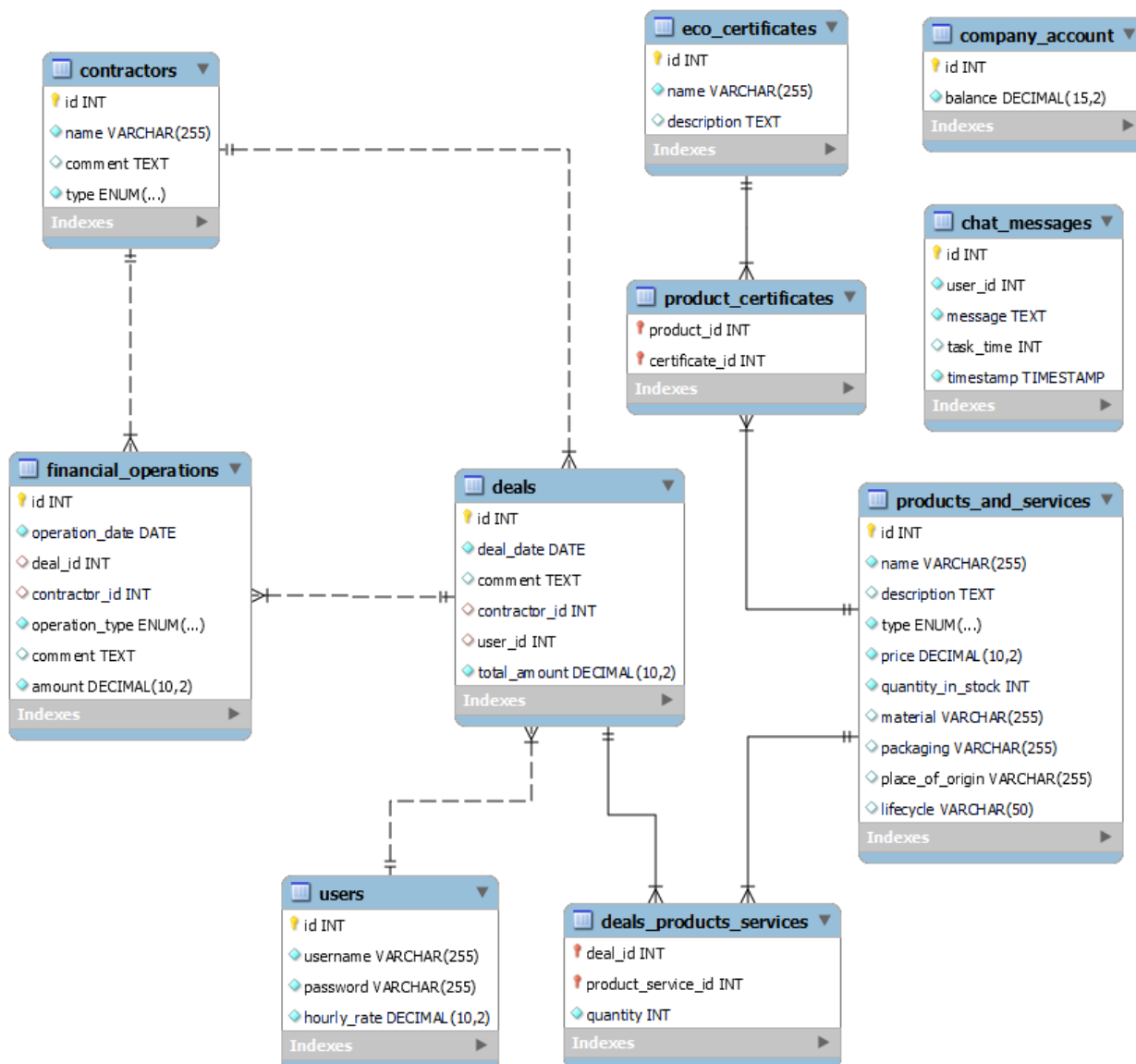


Рисунок 2.1 – Діаграма бази даних

2.3 Розробка дизайну користувацького інтерфейсу

Розробка дизайну користувацького інтерфейсу є ключовим етапом у створенні вебзастосунку. Головною метою є створення інтуїтивно зрозумілого, привабливого та зручного інтерфейсу, який забезпечить позитивний досвід взаємодії користувачів із системою.

Для досягнення цієї мети було вирішено використовувати сучасні технології та фреймворки, такі як HTML, CSS, JavaScript та Bootstrap. Ці інструменти

дозволяють швидко розробляти адаптивний та естетично привабливий інтерфейс, який буде однаково добре виглядати на різних пристроях та розмірах екрану [33].

Основним файлом стилів є "styles.css" (додаток А). Цей файл містить всі необхідні правила CSS для визначення зовнішнього вигляду та поведінки елементів інтерфейсу. Тут задаються кольори, шрифти, відступи, тіні, анімації та інші візуальні ефекти. Використання окремого файлу стилів дозволяє відокремити презентаційний код від структурного HTML, що полегшує подальшу підтримку та модифікацію застосунку.

Для забезпечення єдиного та консистентного дизайну на всіх сторінках застосунку було створено шаблони хедера (додаток Б) та футера (додаток В). Ці шаблони включаються на кожній сторінці, гарантуючи, що загальна структура та навігація залишаються незмінними. Хедер містить логотип застосунку, меню навігації та посилання на функціонал авторизації. Футер, в свою чергу, відображає інформацію про авторські права, контактні дані та посилання на соціальні мережі компанії [34].

Для побудови адаптивної сітки та розміщення елементів на сторінках використовується фреймворк Bootstrap. Він надає готові класи та компоненти, які дозволяють швидко створювати розмітку сторінок, яка автоматично адаптується під різні розміри екрану. Наприклад, на головній сторінці застосунку (додаток Г) використовуються класи Bootstrap для створення сітки з віджетами, які відображають загальну статистику по базі даних, фінансову інформацію та графік надходжень за останні 10 днів.

Кольорова схема інтерфейсу була ретельно підібрана з урахуванням специфіки екотоварів. Використовуються переважно природні, спокійні кольори, такі як зелений, блакитний та білий. Це створює асоціації з природою, свіжістю та чистотою, що відповідає тематиці застосунку. Для виділення важливих елементів, таких як кнопки та посилання, використовуються більш насичені відтінки, що привертають увагу користувача [35].

Особлива увага приділяється зручності використання застосунку. Всі елементи керування, такі як кнопки, поля вводу та випадаючі списки, мають достатній розмір та відступи, щоб користувачам було легко взаємодіяти з ними. Текст має достатній контраст з фоном, що забезпечує хорошу читабельність. Навігація по сторінках застосунку є логічною та послідовною, що дозволяє користувачам швидко знаходити потрібну інформацію.

Для покращення візуальної привабливості інтерфейсу використовуються різноманітні графічні елементи, такі як іконки, зображення та анімації. Наприклад, на сторінці товарів та послуг можуть відображатися фотографії продуктів, що допомагає користувачам краще зрозуміти, що вони купують. Анімації використовуються для плавних переходів між сторінками та для привернення уваги до важливих елементів.

Інтерфейс застосунку є адаптивним та реагує на зміну розміру екрану. Це означає, що розташування елементів автоматично підлаштовується під доступний простір, забезпечуючи оптимальне відображення контенту на різних пристроях – від настільних комп'ютерів до смартфонів. Завдяки цьому користувачі можуть зручно працювати з застосунком незалежно від того, який пристрій вони використовують.

Для забезпечення постійного вдосконалення дизайну інтерфейсу проводиться збір зворотного зв'язку від користувачів. Це дозволяє виявляти потенційні проблеми юзабіліті та вносити необхідні корективи. Регулярні оновлення та покращення інтерфейсу гарантують, що застосунок завжди залишається сучасним, привабливим та зручним для користувачів [36].

Підсумовуючи, можна сказати, що розробка дизайну користувацького інтерфейсу для нашого продукту – це комплексний процес, який враховує естетику, юзабіліті та адаптивність. Використання сучасних технологій, продумана кольорова схема, зручна навігація та увага до деталей дозволяють створити привабливий та ефективний інтерфейс, який забезпечує позитивний досвід користувачів та сприяє успіху застосунку на ринку.

Висновки до розділу 2

У другому розділі дипломної роботи було розглянуто ключові аспекти проєктування вебзастосунку. Цей етап є надзвичайно важливим, оскільки він закладає міцний фундамент для подальшої розробки та успішного функціонування застосунку.

Першим кроком було визначення структури та функціоналу розробки. Було проаналізовано вимоги до системи та виділено основні модулі:

- управління товарами та послугами;
- управління контрагентами;
- управління угодами;
- управління фінансовими операціями.

Для кожного модуля було детально описано функціональні можливості та сценарії використання. Це дозволило створити чітке бачення того, як застосунок буде працювати та які завдання він буде виконувати.

Наступним важливим етапом було проєктування моделі бази даних. Було розроблено ER-діаграму, яка відображає сутності системи та зв'язки між ними. На основі цієї діаграми було створено реляційну схему бази даних з детальним описом таблиць, їх полів та обмежень.

Особлива увага була приділена розробці дизайну користувацького інтерфейсу. Було створено привабливий та інтуїтивно зрозумілий інтерфейс, який забезпечує позитивний досвід взаємодії користувачів із системою. Використання сучасних технологій, таких як HTML, CSS та фреймворк Bootstrap, дозволило створити адаптивний дизайн, який однаково добре виглядає на різних пристроях. Продумана кольорова схема, зручна навігація та увага до деталей роблять інтерфейс естетично привабливим та ефективним у використанні.

Результатом проведеної роботи на етапі проєктування є детальний план розробки вебзастосунку з управління інтернет-магазином. Цей план включає в себе наступні моменти:

- визначені вимоги;

- архітектура системи;
- модель бази даних;
- дизайн інтерфейсу.

Він служить дорожньою картою для подальшої реалізації проєкту та забезпечує його успішне виконання.

Підсумовуючи, можна сказати, що етап проєктування є фундаментальним для створення якісного та ефективного застосунку. Ретельне планування, аналіз вимог, проєктування бази даних та розробка привабливого інтерфейсу – все це закладає міцну основу для подальшої реалізації проєкту. Завдяки проведеній роботі на цьому етапі, з'являється чітке розуміння того, що потрібно зробити, як це зробити та який результат очікується отримати.

3 РОЗРОБКА ВЕБЗАСТОСУНКУ

3.1 Вибір та обґрунтування технологій розробки

Вибір відповідних технологій є ключовим фактором успішної розробки системи. Правильно підібраний стек технологій дозволяє досягти високої продуктивності, масштабованості та зручності підтримки застосунку. При виборі технологій були враховані такі критерії, як функціональність, швидкість розробки, наявність документації та спільноти розробників.

Для розробки серверної частини вебзастосунку було розглянуто декілька мов програмування, зокрема PHP, Python та Ruby. Кожна з цих мов має свої переваги та недоліки.

До переваг Python можна віднести:

- простота та читабельність коду;
- наявність потужних фреймворків – Django та Flask;
- широкі можливості для обробки даних та машинного навчання.

До недоліків Python відносяться:

- повільніша швидкість виконання в порівнянні з іншими мовами;
- менша популярність у сфері веброзробки, що може призвести до обмеженого вибору хостингів та ресурсів;
- більша вимогливість до ресурсів сервера.

Переваги Ruby наступні:

- популярний фреймворк Ruby on Rails, який дозволяє швидко створювати вебзастосунки з використанням принципів конвенції над конфігурацією;
- велика кількість готових бібліотек та модулів;
- зручність та ефективність у розробці складних застосунків.

Недоліки Ruby зазначені нижче:

- менша продуктивність порівняно з іншими мовами;
- менша популярність у порівнянні з PHP, що може призвести до обмеженого вибору хостингів та ресурсів;

- вищі витрати на хостинг та обслуговування.

PHP має широкий спектр переваг у цій сфері:

- широка поширеність у сфері веброзробки;
- велика кількість готових бібліотек та плагінів;
- простота інтеграції з різними системами управління базами даних;
- наявність великої спільноти розробників та підтримка;
- висока швидкість виконання для вебзастосунків.

Але PHP має й свої недоліки:

- більша вразливість до помилок безпеки при неналежному використанні;
- більш складний синтаксис у порівнянні з Python;
- обмежені можливості для обробки даних та машинного навчання

порівняно з Python.

Зважаючи усі переваги та недоліки, а також невеликий досвід роботи з деякими з цих мов, було вирішено використовувати PHP як основну мову для серверної частини застосунку (рис 3.1).



Рисунок 3.1 – Лого PHP

Для зберігання та управління даними розглядалися реляційні та нереляційні бази даних. Серед реляційних баз даних розглядалися MySQL та PostgreSQL. PostgreSQL відома своєю надійністю, підтримкою складних типів даних та розширеними можливостями запитів. Однак, враховуючи простоту використання, швидкість та широку сумісність з PHP [10], було обрано MySQL як основну базу даних для вебзастосунку.

Для розробки клієнтської частини вебзастосунку розглядалися різні технології та фреймворки. Окрім традиційних HTML та CSS, були розглянуті такі фреймворки, як React та Angular. React дозволяє створювати інтерактивні та швидкодіючі користувацькі інтерфейси з використанням компонентного підходу. Angular, в свою чергу, надає потужний інструментарій для створення складних односторінкових застосунків. Однак, враховуючи відносну простоту проєкту та необхідність швидкої розробки, було вирішено використовувати класичний стек HTML, CSS та JavaScript (рис. 3.2) з використанням бібліотеки jQuery для спрощення взаємодії з DOM та виконання AJAX запитів [11].



Рисунок 3.2 – Лого HTML, CSS та JS

Для забезпечення адаптивного дизайну та пришвидшення розробки інтерфейсу було обрано фреймворк Bootstrap. Bootstrap надає набір готових компонентів та стилів, що дозволяє швидко створювати привабливий та респонсивний інтерфейс. Серед альтернатив розглядалися такі фреймворки, як Foundation та Materialize. Foundation відомий своєю гнучкістю та модульністю, а Materialize надає стильний дизайн у стилі Material Design від Google. Однак,

враховуючи широку популярність та наявність великої кількості ресурсів та документації, було обрано Bootstrap як основний фреймворк для верстки.

Для забезпечення безпеки даних та користувачів були розглянуті різні підходи. Окрім хешування паролів та використання сесій для аутентифікації, були розглянуті додаткові заходи, такі як використання HTTPS для шифрування даних під час передачі та застосування механізмів захисту від SQL-ін'єкцій та міжсайтового скриптингу (XSS). Також розглядалася можливість використання готових рішень для аутентифікації, таких як OAuth або JWT (JSON Web Tokens). Однак, враховуючи відносну простоту проєкту та наявність вбудованих механізмів безпеки у PHP та MySQL, було вирішено використовувати класичний підхід з хешуванням паролів та сесіями.

Для підвищення продуктивності та зменшення навантаження на сервер розглядалися різні техніки кешування. Окрім кешування результатів запитів до бази даних, розглядалися такі підходи, як кешування на рівні фреймворку (наприклад, з використанням Memcached або Redis) та кешування на рівні вебсервера (наприклад, з використанням Nginx або Apache). Враховуючи масштаб проєкту та очікуване навантаження, було вирішено обмежитися кешуванням результатів запитів до бази даних з використанням вбудованих механізмів PHP.

Для забезпечення зручності користування та покращення візуальної привабливості інтерфейсу були розглянуті додаткові бібліотеки та плагіни. Для відображення графіків та діаграм розглядалися такі бібліотеки, як Chart.js, D3.js та Highcharts. Кожна з цих бібліотек має свої переваги та особливості. Chart.js відрізняється простотою використання та наявністю великої кількості готових прикладів, тому було вирішено використовувати саме її. Для реалізації функціоналу drag-and-drop розглядалися бібліотеки Dragula та Sortable.js. Враховуючи простоту інтеграції та наявність докладної документації, було обрано бібліотеку Dragula [12].

Також було приділено увагу зручності розробки та підтримки коду. Використовується модульний підхід до організації файлової структури проєкту, що

дозволяє легко знаходити та модифікувати необхідні компоненти. Код розділений на логічні частини, такі як моделі, контролери та представлення, що відповідає архітектурному шаблону MVC (Model-View-Controller). Це забезпечує чистоту та читабельність коду, а також полегшує його розширення та підтримку [13].

Для покращення користувацького досвіду та естетичної привабливості інтерфейсу використовуються сучасні техніки вебдизайну. Застосовуються принципи адаптивного дизайну, що забезпечує коректне відображення контенту на різних пристроях та розмірах екрану.

Для оптимізації швидкодії вебзастосунку застосовуються техніки оптимізації зображень, мінімізації CSS та JavaScript файлів, а також стиснення відповідей сервера. Це дозволяє зменшити час завантаження сторінок та покращити загальну продуктивність застосунку.

Таким чином, вибір технологій для розробки продукту базувався на ретельному аналізі вимог проєкту, власному досвіді та порівнянні переваг та недоліків різних варіантів (табл. 3.1). Обрані технології, такі як PHP, MySQL, HTML, CSS, JavaScript з використанням фреймворків Bootstrap та jQuery, а також додаткових бібліотек Chart.js та Dragula, дозволяють ефективно реалізувати поставлені задачі та створити високоякісний у використанні вебзастосунок.

Таблиця 3.1 – Використані технології для розробки вебзастосунку для управління інтернет-магазином екотоварів

Технологія	Опис
PHP	Серверна мова програмування, використовується для генерації динамічного контенту, обробки запитів та взаємодії з базою даних.
MySQL	Реляційна система управління базами даних, забезпечує ефективне зберігання та вибірку даних.
HTML	Мова розмітки гіпертексту, використовується для структурування контенту вебсторінок.

Закінчення таблиці 3.1

Технологія	Опис
CSS	Каскадні таблиці стилів, використовуються для візуального оформлення та стилізації елементів інтерфейсу.
JavaScript	Мова програмування для клієнтської частини, використовується для додавання інтерактивності та динамічної поведінки вебсторінок.
Bootstrap	Фреймворк для розробки адаптивних та респонсивних вебінтерфейсів, надає набір готових компонентів та стилів.
jQuery	Бібліотека JavaScript, спрощує маніпулювання елементами DOM, обробку подій та виконання AJAX запитів.
Chart.js	Бібліотека для створення інтерактивних та анімованих графіків та діаграм.
Dragula	Бібліотека для реалізації функціоналу drag-and-drop, спрощує розробку інтерфейсів з можливістю перетягування елементів.
PHP Storm	Інтегроване середовище розробки (IDE) для PHP, забезпечує зручні інструменти для редагування коду, налагодження та управління проектом.
phpMyAdmin	Вебінтерфейс для адміністрування бази даних MySQL, дозволяє виконувати операції з запитам та користувачами.
Кешування	Техніка збереження результатів обчислень та запитів до бази даних для швидкої генерації сторінок при повторних запитах.
Модульний підхід	Організація файлової структури проекту, розділення коду на логічні частини для зручності розробки та підтримки.
Адаптивний дизайн	Принципи вебдизайну, які забезпечують коректне відображення контенту на різних пристроях та розмірах екрану.
Оптимізація	Техніки оптимізації зображень, мінімізації CSS та JavaScript файлів, стиснення відповіді сервера для покращення швидкодії.

Такий вибір технологій дозволяє ефективно розробити вебзастосунок з управління інтернет-магазину екотоварів, який відповідає сучасним стандартам веброзробки та задовольняє потреби менеджерів.

3.2 Реалізація серверної частини вебзастосунку

Серверна частина вебзастосунку з продажу екотоварів реалізована на мові програмування PHP. Це широко використовувана мова для веброзробки, яка має багато вбудованих функцій та бібліотек, що спрощують роботу з базами даних, обробку форм та генерацію динамічного контенту.

Основна логіка серверної частини застосунку зосереджена у файлі `functions.php` (додаток Д). Цей файл містить набір функцій, які забезпечують взаємодію з базою даних, обробку даних та виконання різноманітних операцій, необхідних для функціонування застосунку [12].

Перш за все, файл `functions.php` підключає файл `database.php`, який містить налаштування підключення до бази даних. Це дозволяє встановити з'єднання з базою даних MySQL та виконувати запити до неї.

Наступним важливим аспектом є управління сесіями користувачів. Функції `isLoggedIn()`, `authenticateUser()`, `userExists()`, `registerUser()` та `logoutUser()` відповідають за аутентифікацію та авторизацію користувачів. Вони перевіряють, чи користувач увійшов у систему, виконують перевірку логіна та пароля, реєструють нових користувачів та здійснюють вихід з системи [16].

Для роботи з користувачами реалізовано функції `getUsers()`, `addUser()`, `getUserById()`, `updateUser()` та `deleteUser()`. Ці функції дозволяють отримувати список користувачів, додавати нових користувачів, отримувати інформацію про конкретного користувача, оновлювати дані користувача та видаляти користувачів з бази даних.

Для реалізації функціоналу чату між користувачами створено функції `getChatMessages()`, `addChatMessage()`, `deleteChatMessage()` та `updateChatMessage()`.

Вони забезпечують отримання списку повідомлень чату, додавання нових повідомлень, видалення та оновлення повідомлень.

Робота з контрагентами здійснюється за допомогою функцій `getContractorsWithDealCount()`, `createContractor()`, `getContractorById()`, `updateContractor()`, `deleteContractor()` та `getContractorDetails()`. Ці функції дозволяють отримувати список контрагентів разом з кількістю пов'язаних з ними угод, створювати нових контрагентів, отримувати інформацію про конкретного контрагента, оновлювати дані контрагента, видаляти контрагентів та отримувати детальну інформацію про контрагента, включаючи пов'язані з ним угоди.

Для управління угодами реалізовано функції `getDeals()`, `getAllContractors()`, `getAllProductsAndServices()`, `createDeal()`, `getDealDetails()`, `deleteDeal()`, `updateDeal()`, `getDealMainDetails()` та `getDealProductsAndServices()`. Ці функції дозволяють отримувати список угод, список всіх контрагентів та товарів/послуг, створювати нові угоди, отримувати детальну інформацію про угоду, видаляти угоди, оновлювати дані угоди, отримувати основні деталі угоди та отримувати список товарів/послуг, пов'язаних з угодою.

Для сортування та фільтрування товарів та послуг створені функції `getAllUniqueValues()`, `countItemsByFilter()`, `getFilterCounts()`. Вони дозволяють отримати унікальні значення для конкретної колонки в базі даних, підраховують кількість елементів у таблиці, що відповідають певному фільтру та кількість товарів у кожній категорії фільтру, що дозволяє менеджерам бачити, скільки товарів відповідає конкретному фільтру.

Для роботи з товарами та послугами створено функції `addProductOrService()`, `getProductOrServiceById()`, `updateProductOrService()`, `isProductOrServiceUsedInDeals()` та `deleteProductOrService()`. Вони дозволяють додавати нові товари/послуги, отримувати інформацію про конкретний товар/послугу, оновлювати дані товару/послуги, перевіряти, чи використовується товар/послуга в угодах, та видаляти товари/послуги.

Для управління фінансовими операціями реалізовано функції `getCompanyAccount()`, `getAllFinancialOperations()`, `addFinancialOperation()`, `updateCompanyAccountBalance()`, `addExpense()`, `decreaseCompanyAccountBalance()`, `getAllDeals()`, `getDealById()`, `addIncomeByDeal()`, `increaseCompanyAccountBalance()`, `getFinancialOperationById()`, `updateFinancialOperation()`, `getFinancialOperationDetailsById()` та `deleteFinancialOperation()`. Ці функції дозволяють отримувати баланс компанії, список всіх фінансових операцій, додавати нові фінансові операції, оновлювати баланс компанії, додавати витрати, зменшувати баланс компанії, отримувати список всіх угод, отримувати інформацію про конкретну угоду, додавати надходження по угоді, збільшувати баланс компанії, отримувати інформацію про конкретну фінансову операцію, оновлювати фінансову операцію, отримувати детальну інформацію про фінансову операцію та видаляти фінансові операції.

Для реалізації віджетів на головній сторінці створено функції `getCount()`, `getTotalAmount()`, `getCompanyBalance()` та `getLastTenDaysIncomes()`. Вони дозволяють отримувати загальну кількість записів у таблицях, загальну суму фінансових операцій певного типу, баланс компанії та надходження за останні 10 днів [17].

Всі ці функції використовують підготовлені запити до бази даних з використанням PDO. Це забезпечує безпеку від SQL-ін'єкцій та дозволяє ефективно виконувати запити з параметрами.

Для обробки помилок та забезпечення цілісності даних використовуються транзакції бази даних. Наприклад, при створенні або оновленні угоди, всі операції з базою даних виконуються в рамках транзакції. Якщо виникає помилка, транзакція відкочується, щоб зберегти цілісність даних.

Функції в файлі `functions.php` організовані за логічними розділами, такими як робота з користувачами, контрагентами, угодами, товарами/послугами та фінансовими операціями. Це покращує читабельність коду та полегшує його підтримку.

Використання функцій дозволяє розділити логіку застосунку на окремі блоки, які можна викликати з різних частин коду. Це сприяє модульності та повторному використанню коду.

Параметри функцій дозволяють передавати дані, необхідні для виконання операцій. Наприклад, при створенні нового контрагента, функція `createContractor()` приймає параметри `$name`, `$type` та `$comment`, які містять відповідні дані для створення контрагента.

Функції повертають результати своєї роботи, такі як масиви з даними з бази даних, булеві значення, що вказують на успішність операції, або ідентифікатори створених записів. Це дозволяє використовувати результати роботи функцій в інших частинах коду.

Для забезпечення безпеки, паролі користувачів зберігаються в базі даних в хешованому вигляді з використанням функції `password_hash()`. При аутентифікації користувача, введений пароль порівнюється з хешем, збереженим в базі даних, за допомогою функції `password_verify()`.

Для оптимізації запитів до бази даних використовуються індекси на відповідних полях таблиць. Наприклад, поля `contractor_id` та `user_id` в таблиці `deals` мають індекси, що пришвидшує виконання запитів з фільтрацією за цими полями.

Код серверної частини вебзастосунку добре структурований, використовує значущі імена змінних та функцій, що покращує його читабельність та зрозумілість. Коментарі використовуються для пояснення призначення функцій та важливих ділянок коду.

Серверна частина застосунку забезпечує надійну та ефективну взаємодію з базою даних, обробку даних та виконання різноманітних операцій. Використання функцій, підготовлених запитів, транзакцій та хешування паролів гарантує безпеку, цілісність даних та зручність розробки та підтримки застосунку.

3.3 Розробка інтерфейсу

Інтерфейс розроблено з використанням HTML, CSS та фреймворку Bootstrap. Ці технології дозволяють створити привабливий, адаптивний та зручний у використанні інтерфейс, який забезпечує позитивний досвід взаємодії користувача з системою.

Розглянемо детальніше кожен з наданих файлів та опишемо інтерфейс та функціонал відповідних сторінок.

Почати можна з навігаційної панелі (рис. 3.3), де є усі необхідні функції для роботи з системою.

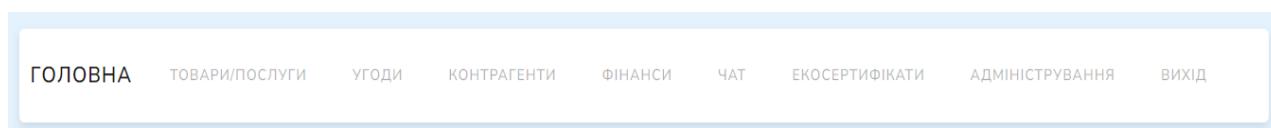


Рисунок 3.3 – Панель навігації системи

1. Сторінка керування товарами та послугами. Цей файл відповідає за відображення сторінки зі списком товарів та послуг (рис. 3.4). На сторінці присутній заголовок "Товари/Послуги" та кнопка "Додати товар чи послугу", яка веде на сторінку додавання нового товару або послуги.

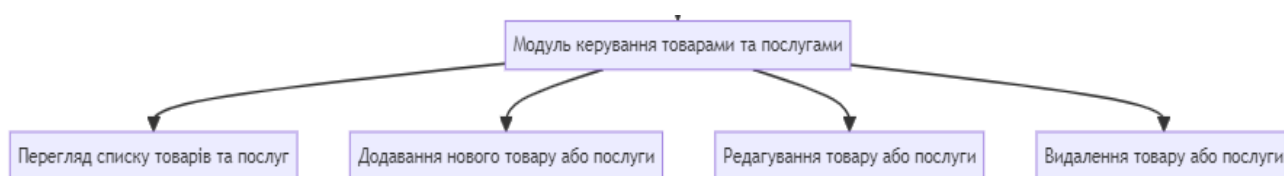


Рисунок 3.4 – Діаграма функціоналу модулю керування товарами

Нижче розташована таблиця (рис. 3.5), яка містить інформацію про кожен товар або послугу, таку як ID, назва, тип (товар або послуга), вартість, наявність (для товарів), матеріал, упаковка, місце походження, термін служби, екосертифікати та кнопки для редагування та видалення товару/послуги.

Кафедра інтелектуальних інформаційних систем
 Вебзастосунок для управління інтернет-магазином екотоварів

ФІЛЬТРИ

Ціна

від 0 до 100 грн

Матеріал

- Бамбук (8)
- Натуральні інгредієнти (13)
- Органічна бавовна (1)
- Органічні інгредієнти (19)
- Перероблене скло (1)
- Перероблені матеріали (5)
- Природні матеріали (1)
- Сонячні панелі (2)

Упаковка

- Без упаковки (12)
- Біорозкладна (17)
- Перероблена упаковка (20)
- Скло (1)

ТОВАРИ/ПОСЛУГИ

ID	НАЗВА	ТИП	ВАРТІСТЬ	НАЯВНІСТЬ	МАТЕРІАЛ	УПАКОВКА
1	Шампунь з квітковим екстрактом	Товар	7.00 грн.	75	Натуральні інгредієнти	Біорозкладна
2	Лосьйон для тіла з алое	Товар	10.00 грн.	90	Органічні інгредієнти	Біорозкладна
3	Бамбукові столові набори	Товар	12.00 грн.	100	Бамбук	Без упаковки
4	Органічна рідина для компосту	Товар	5.00 грн.	150	Органічні інгредієнти	Перероблена упаковка

Рисунок 3.5 – Скріншот сторінки products_and_services.php

2. Сторінка додавання товарів та послугами (рис. 3.6). Цей файл містить форму для додавання нового товару або послуги. Форма включає поля для введення назви, опису, вибору типу (товар або послуга), вартості, наявності (для товарів), екосертифікату (з наявних екосертифікатів), матеріала, упаковки, місця походження, терміна служби.

Після заповнення форми та натискання кнопки "Додати", дані відправляються на сервер, де за допомогою функції addProductOrService() з файлу functions.php відбувається додавання нового товару або послуги до бази даних.

ДОДАТИ ТОВАР ЧИ ПОСЛУГУ

Назва:

Опис:

Тип:

Вартість:

Наявність:

Екосертифікати:

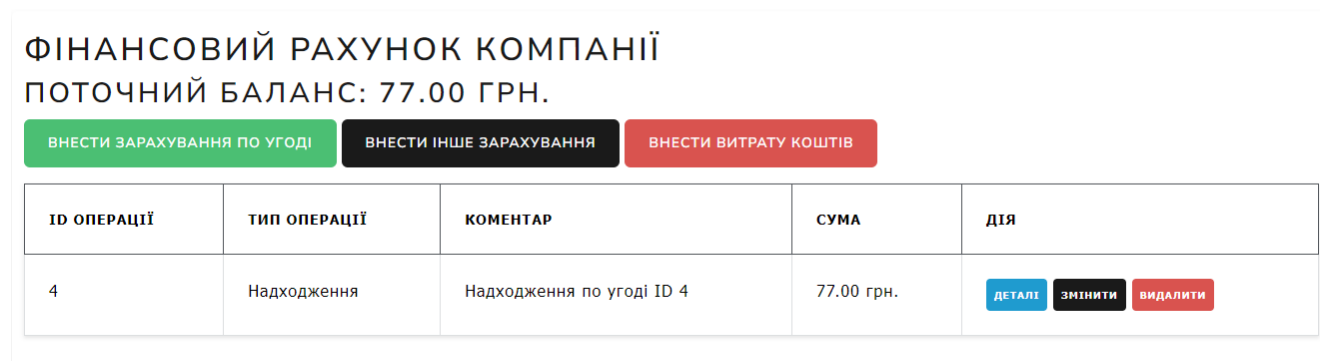
Виберіть екосертифікати

Рисунок 3.6 – Скріншот сторінки add_product_service.php

3. Сторінка керування фінансами компанії. Цей файл відображає фінансовий рахунок компанії та список всіх фінансових операцій (рис. 3.7). На сторінці присутній заголовок "Фінансовий рахунок компанії" та поточний баланс компанії.

Нижче розташовані кнопки "Внести зарахування по угоді", "Внести інше зарахування" та "Внести витрату коштів", які ведуть на відповідні сторінки для додавання фінансових операцій.

Далі йде таблиця з інформацією про кожну фінансову операцію, таку як ID операції, тип операції (надходження або витрата), коментар, сума та кнопки для перегляду деталей, зміни та видалення операції.



ID ОПЕРАЦІЇ	ТИП ОПЕРАЦІЇ	КОМЕНТАР	СУМА	ДІЯ
4	Надходження	Надходження по угоді ID 4	77.00 грн.	ДЕТАЛІ ЗМІНИТИ ВИДАЛИТИ

Рисунок 3.7 – Скріншот сторінки finance.php

4. Адміністративна частина. Цей файл представляє панель адміністратора, доступну лише для користувачів з ID менше або рівним 10 (рис. 3.8). На сторінці відображається список всіх користувачів системи з інформацією про їх ID, ім'я користувача (логін), ставку оплати за годину та кнопки для редагування та видалення користувача.

Також присутня кнопка "Додати користувача", яка веде на сторінку додавання нового користувача.

ПАНЕЛЬ АДМІНІСТРАТОРА

ДОДАТИ КОРИСТУВАЧА

ID	ІМ'Я КОРИСТУВАЧА (ЛОГІН)	СТАВКА (ГРН./ГОДИНА)	ДІЇ
1	admin	0.00	РЕДАГУВАТИ ВИДАЛИТИ

Рисунок 3.8 – Скріншот сторінки admin.php

5. Корпоративний чат. Цей файл містить інтерфейс чату для спілкування між користувачами системи. На сторінці відображається вікно чату з повідомленнями та форма для відправки нового повідомлення (рис. 3.9).

Форма включає поле для введення тексту повідомлення та поле для вказання витраченого часу на виконання завдання (в хвиликах). При відправці повідомлення воно додається до чату за допомогою функції `addChatMessage()` з файлу `functions.php`.

Також присутнє модальне вікно для редагування повідомлення, яке відкривається при натисканні на кнопку редагування біля відповідного повідомлення.

ЧАТ

[15.05.2024, 12:50] demo : перевірка 123 **РЕДАКТИРОВАТИ** **УДАЛИТИ**

Повідомлення:

|

Витрачений час (хв):

ВІДПРАВИТИ

Рисунок 3.9 – Скріншот сторінки chat.php

6. Сторінка детальної інформації про контрагента. Цей файл відображає детальну інформацію про конкретного контрагента (рис. 3.10). На сторінці присутній заголовок "Деталі контрагента" та таблиця з інформацією про

контрагента, такою як ID, назва, тип (юридична або фізична особа), кількість угод, загальна сума угод та коментар.

Нижче таблиці розташовані кнопки "Редагувати", "Видалити" та "Згенерувати акт звірки". Кнопка "Редагувати" веде на сторінку редагування даних контрагента, кнопка "Видалити" дозволяє видалити контрагента після підтвердження дії, а кнопка "Згенерувати акт звірки" генерує акт звірки для даного контрагента.

ДЕТАЛІ КОНТРАГЕНТА	
ІД	2
НАЗВА	Березанський Олександр Петрович
ТИП	Фізична особа
КІЛЬКІСТЬ УГОД	1
ЗАГАЛЬНА СУМА УГОД	77.00
КОМЕНТАР	

РЕДАГУВАТИ **ВИДАЛИТИ** **ЗГЕНЕРУВАТИ АКТ ЗВІРКИ**

Рисунок 3.10 – Сторінка contractor_details.php

7. Сторінка додавання контрагентів. Цей файл містить форму для створення нового контрагента (рис. 3.11). На сторінці присутній заголовок "Додати нового контрагента" та форма з полями для введення назви контрагента, вибору типу (юридична або фізична особа) та введення коментаря.

Якщо при відправці форми виникають помилки (наприклад, не заповнені обов'язкові поля), вони відображаються над формою у вигляді повідомлень про помилки.

Після успішного заповнення форми та натискання кнопки "Додати", новий контрагент додається до бази даних, і користувач перенаправляється на головну сторінку.

ДОДАТИ НОВОГО КОНТРАГЕНТА

Назва

Тип

Юридична особа

Коментар

ДОДАТИ

Рисунок 3.11 – Скріншот сторінки create_contractor.php

8. Сторінка додавання угоди. Цей файл містить форму для створення нової угоди. На сторінці присутній заголовок "Створити нову угоду" та форма з полями для вибору дати угоди, контрагента, товарів/послуг (з можливістю вказати кількість та ціну), введення коментаря та загальної суми угоди (рис. 3.12).

При виборі товарів/послуг відображається таблиця з доступними товарами/послугами, де можна вибрати необхідні позиції, вказати кількість та побачити ціну за одиницю та доступну кількість на складі. Загальна сума угоди розраховується автоматично на основі вибраних товарів/послуг та їх кількості за допомогою JavaScript.

Після заповнення форми та натискання кнопки "Створити угоду", нова угода додається до бази даних, і користувач перенаправляється на сторінку зі списком угод.

Для зручного керування товарами та послугами реалізовано динамічне відображення таблиці з доступними товарами/послугами при створенні нової угоди. Таблиця автоматично оновлюється після кожної угоди, зменшуючи кількість товарів у наявності і на сторінці товарів, що видно нижче (рис. 3.13)

СТВОРИТИ НОВУ УГОДУ

Дата угоди:
18.06.2024

Контрагент:
Березанський Олександр Петрович

Товари/Послуги:

ТОВАР/ПОСЛУГА	КІЛЬКІСТЬ	ЦІНА	ДОСТУПНО
<input checked="" type="checkbox"/> Шампунь з квітковим екстрактом	<input type="text" value="5"/>	7.00	75
<input type="checkbox"/> Лосьйон для тіла з алое	<input type="text"/>	10.00	90
<input checked="" type="checkbox"/> Бамбукові столові набори	<input type="text" value="3"/>	12.00	100

Рисунок 3.12 – Скріншот сторінки create_deals.php

ТОВАРИ/ПОСЛУГИ

ДОДАТИ ТОВАР ЧИ ПОСЛУГУ

ID	НАЗВА	ТИП	ВАРТІСТЬ	НАЯВНІСТЬ
1	Шампунь з квітковим екстрактом	Товар	7.00 грн.	<u>70</u>
2	Лосьйон для тіла з алое	Товар	10.00 грн.	90
3	Бамбукові столові набори	Товар	12.00 грн.	<u>97</u>

Рисунок 3.13 – Зміна кількості товарів у наявності після створення угоди

9. Сторінка керування контрагентами. Цей файл відображає список усіх контрагентів з основною інформацією про них (рис. 3.14). На сторінці присутній заголовок "Список контрагентів" та таблиця з інформацією про кожного контрагента, такою як ID, назва, тип, кількість угод та кнопки для виконання дій (деталі, редагування, видалення).

Кнопка "Створити нового контрагента" веде на сторінку створення нового контрагента.

При натисканні на кількість угод контрагента, користувач перенаправляється на сторінку зі списком угод, відфільтрованих за даним контрагентом.

СПИСОК КОНТРАГЕНТІВ

СТВОРИТИ НОВОГО КОНТРАГЕНТА

ID	НАЗВА	ТИП	УГОДИ	ДІЇ
2	Березанський Олександр Петрович	Фізична особа	1	ДЕТАЛІ РЕДАГУВАТИ ВИДАЛИТИ
3	ТОВ "Зелений двір"	Юридична особа	0	ДЕТАЛІ РЕДАГУВАТИ ВИДАЛИТИ

Рисунок 3.14 – Сторінка contractor_details.php

10. Сторінка генерування рахунку по угоді. Цей файл генерує рахунок на оплату для конкретної угоди (рис. 3.15). На сторінці відображається заголовок "Рахунок на оплату", інформація про компанію-відправника та контрагента-одержувача.

Нижче розташована таблиця з переліком товарів/послуг, їх кількістю, ціною за одиницю та загальною ціною. В кінці таблиці виводиться загальна сума рахунку.

Сторінка також містить поле для підпису директора та кнопки "Друк" і "Експорт в PDF". Кнопка "Друк" дозволяє роздрукувати рахунок, а кнопка "Експорт в PDF" генерує PDF-файл з рахунком за допомогою бібліотеки html2pdf.js.

РАХУНОК НА ОПЛАТУ

Відправник: Назва вашої компанії
Адреса: Адреса вашої компанії
Реквізити: Банківські реквізити вашої компанії

Одержувач: Березанський Олександр Петрович

НАЗВА ТОВАРУ/ПОСЛУГИ	КІЛЬКІСТЬ	ЦІНА ЗА ОДИНИЦЮ	ЗАГАЛЬНА ЦІНА
Шампунь з квітковим екстрактом	5	7.00	35.00
Освіжувач повітря з ефірними оліями	2	6.00	12.00
Кавовий крем	3	10.00	30.00
		Загальна сума:	77.00

Директор: (Підпис) _____

ДРУК

ЕКСПОРТ В PDF

Рисунок 3.15 – Скріншот сторінки generate_invoice.php

11. Сторінка генерації видаткової накладної. Цей файл генерує акт звірки взаєморозрахунків між компанією та певним контрагентом (рис. 3.16). На сторінці відображається заголовок з назвами компанії та контрагента, а також пояснювальний текст щодо призначення акту звірки.

Нижче розташована таблиця, яка містить дані про взаємні розрахунки за даними обліку компанії та контрагента. У таблиці відображається дата, номер

документа (видаткової накладної), дебет та кредит за даними компанії, а також аналогічні дані за даними контрагента.

Після таблиці вказується дата, на яку складено акт звірки, та передбачено місця для підписів директорів та головних бухгалтерів компанії та контрагента. Сторінка містить кнопку "Друк акта звірки", яка дозволяє роздрукувати акт звірки. При друку кнопка приховується завдяки використанню CSS-класу no-print.

ВИДАТКОВА НАКЛАДНА

Відправник: Назва вашої компанії
 Адреса: Адреса вашої компанії
 Реквізити: Банківські реквізити вашої компанії

Одержувач: Березанський Олександр Петрович

НАЗВА ТОВАРУ/ПОСЛУГИ	КІЛЬКІСТЬ	ЦІНА ЗА ОДИНИЦЮ	ЗАГАЛЬНА ЦІНА
Шампунь з квітковим екстрактом	5	7.00	35.00
Освіжувач повітря з ефірними оліями	2	6.00	12.00
Кавовий крем	3	10.00	30.00
Загальна сума:			77.00

Директор: (Підпис) _____

ДРУК

Рисунок 3.16 – Скріншот сторінки generate_reconciliation_act.php

Акт звірки взаєморозрахунків є важливим документом, який допомагає узгодити стан розрахунків між компанією та контрагентом. Він дозволяє виявити можливі розбіжності в обліку та забезпечує прозорість фінансових операцій між сторонами.

Загалом, сторінка `generate_reconciliation_act.php` забезпечує ефективне формування акту звірки взаєморозрахунків між компанією та контрагентом, що є важливим елементом фінансового контролю та співпраці між сторонами.

12. Фільтрування на сторінці з товарами. Цей розділ у файлі `products_and_services.php` дає можливість фільтрувати товари за різними показниками, такими як ціна (спеціальний зручний повзунок для цього), матеріал, упаковка, місце походження, термін служби. Є можливість вибору товару потрібної категорії через галочки. На скріншоті нижче (рис. 3.17) видно використання 2 фільтрів, цей скріншот можна порівняти з рисунком товарів без фільтрів вище (рис. 3.5) та побачити, що функція працює.

ФІЛЬТРИ

Ціна

від 0 до 100 грн

Матеріал

- Бамбук (8)
- Натуральні інгредієнти (13)**
- Органічна бавовна (1)
- Органічні інгредієнти (19)
- Перероблене скло (1)
- Перероблені матеріали (5)
- Природні матеріали (1)
- Сонячні панелі (2)

Упаковка

- Без упаковки (12)
- Біорозкладна (17)**
- Перероблена упаковка (20)
- Скло (1)

ТОВАРИ/ПОСЛУГИ

ДОДАТИ ТОВАР ЧИ ПОСЛУГУ

ID	НАЗВА	ТИП	ВАРТІСТЬ	НАЯВНІСТЬ	МАТЕРІАЛ	УПАКОВКА
1	Шампунь з квітковим екстрактом	Товар	7.00 грн.	75	Натуральні інгредієнти	Біорозкладна
5	Захисний крем від сонця	Товар	8.00 грн.	66	Натуральні інгредієнти	Біорозкладна
8	Крем для обличчя з алое	Товар	15.00 грн.	60	Натуральні інгредієнти	Біорозкладна
10	Освіжувач повітря з ефірними оліями	Товар	6.00 грн.	118	Натуральні інгредієнти	Біорозкладна

Рисунок 3.17 – Скріншот фільтрації на сторінці `products_and_services.php`

13. Сторінка зі списком екосертифікатів. Ця сторінка у вебзастосунку призначена для управління екологічними сертифікатами, що присвоюються товарам або послугам (рис. 3.18). Ця сторінка дозволяє менеджерам магазину переглядати всі доступні сертифікати, додавати нові, редагувати існуючі та видаляти сертифікати.

Сертифікати, що додані або відредаговані на цій сторінці, можуть бути призначені конкретним товарам чи послугам через сторінку управління товарами. Інформація про сертифікати відображається на сторінці товарів. Це допомагає підкреслити екологічну відповідальність товарів та сприяє покращенню їх репутації серед споживачів.

СПИСОК СЕРТИФІКАТІВ

ДОДАТИ СЕРТИФІКАТ

ID	НАЗВА	опис	дії
1	Organic Certification	Підтверджує, що продукт виготовлений з органічних інгредієнтів.	РЕДАГУВАТИ ВИДАЛИТИ
2	Eco-Friendly	Продукт, виготовлений без шкоди для навколишнього середовища.	РЕДАГУВАТИ ВИДАЛИТИ
3	Fair Trade	Гарантує справедливі умови праці та торгівлі.	РЕДАГУВАТИ ВИДАЛИТИ

Рисунок 3.18 – Скріншот списку сертифікатів на сторінці edit_certificates.php

14. Також вебзастосунок відображається належним чином на телефоні завдяки використанню адаптивного дизайну (рис. 3.19). Адаптивний дизайн дозволяє вебсторінкам автоматично підлаштовуватися під різні розміри екранів та типи пристроїв. Це досягається за допомогою технологій, таких як медіа-запити у CSS, які змінюють стилі залежно від ширини екрана, та використанням відносних одиниць виміру. Також часто використовуються CSS-фреймворки, такі як Bootstrap, які забезпечують готові компоненти для створення адаптивних макетів. Завдяки цим підходам система забезпечує зручний та привабливий інтерфейс для користувачів на різних пристроях, включаючи мобільні телефони, планшети та настільні комп'ютери.



Рисунок 3.19 – Відображення застосунку на телефоні

15. На головній сторінці інформаційної панелі вебзастосунку відображається поточний стан бази даних та фінансова інформація (рис. 3.20).

Цей інтерфейс дозволяє швидко отримати загальну інформацію про стан бізнесу, зокрема щодо фінансів та кількості різних елементів у системі. Також тут використовується адаптивний дизайн, що забезпечує зручний перегляд на різних пристроях.

ІНФОРМАЦІЙНА ПАНЕЛЬ

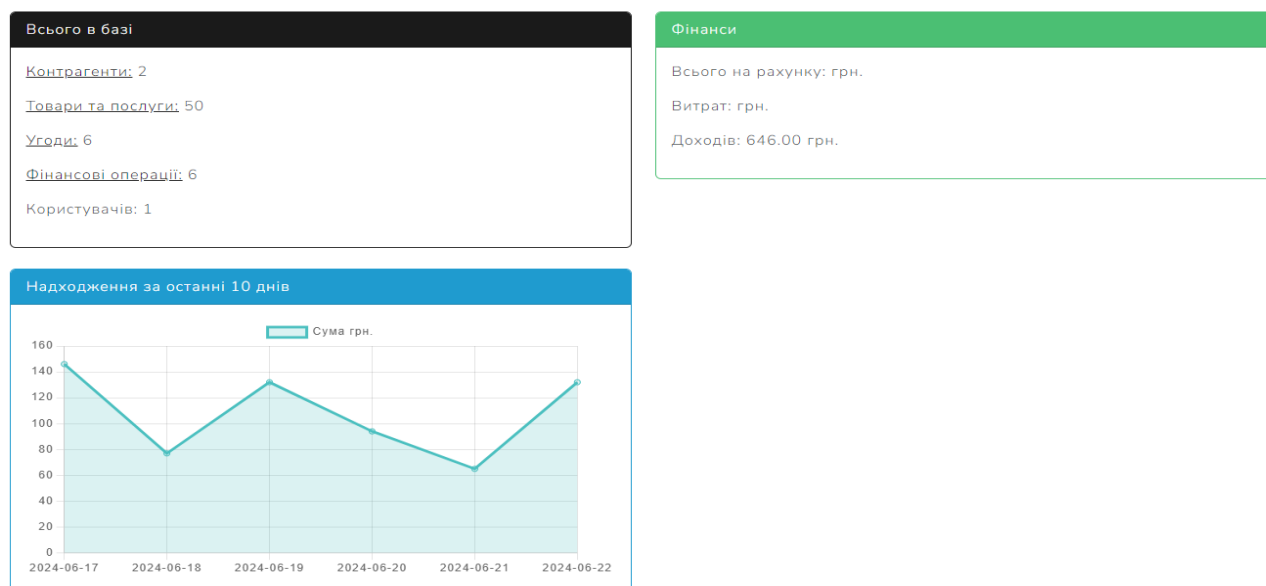


Рисунок 3.20 – Відображення інформаційної панелі

Всі ці сторінки використовують спільні шаблони `header.php` та `footer.php`, які забезпечують єдиний вигляд та навігацію по розробці.

Для стилізації сторінок використовується фреймворк `Bootstrap`, який надає готові класи та компоненти для створення адаптивного та привабливого інтерфейсу. Додаткові стилі застосовуються за допомогою файлу `styles.css`.

Інтерфейс вебзастосунку розроблений з урахуванням зручності використання та естетичної привабливості. Використовуються чіткі заголовки, таблиці для структурованого відображення даних, кнопки для виконання дій та форми для введення інформації.

Адаптивний дизайн забезпечує коректне відображення сторінок на різних пристроях та розмірах екрану. Кольорова схема інтерфейсу є приємною та гармонійною, використовуються кольори, які асоціюються з екотематикою, такі як зелений та блакитний. Інтерактивні елементи, такі як кнопки та посилання, мають чіткі стани (наведення, натискання), що покращує зворотній зв'язок з користувачем та полегшує навігацію по сторінках.

Модальні вікна використовуються для відображення додаткової інформації або для виконання певних дій, таких як редагування повідомлення в чаті. Це

дозволяє зберегти основний контент сторінки незмінним та зосередити увагу користувача на конкретній дії.

Підсумовуючи, можна сказати, що інтерфейс вебзастосунку є привабливим, адаптивним та зручним у використанні. Він забезпечує користувачам легкий доступ до необхідних функцій та інформації, наприклад:

- управління товарами/послугами;
- управління фінансовими операціями;
- адміністрування користувачів;
- спілкування через чат.

Використання сучасних технологій та дизайнерських рішень робить інтерфейс естетично привабливим та приємним для взаємодії.

3.4 Тестування вебзастосунку

Тестування є невід'ємною частиною процесу розробки вебзастосунку. Воно дозволяє перевірити функціональність, надійність, продуктивність та зручність використання застосунку, а також виявити та виправити потенційні помилки та недоліки.

Для забезпечення якості вебзастосунку було проведено різні види тестування, зокрема:

1) функціональне тестування:

- перевірка коректності роботи всіх функціональних модулів застосунку, таких як авторизація та аутентифікація, керування товарами та послугами, керування контрагентами, керування угодами, керування фінансами, адміністрування та чат;
- тестування форм введення даних, перевірка валідації та обробки коректних та некоректних вхідних даних;
- перевірка навігації між сторінками, коректності посилань та переходів;

- тестування функцій додавання, редагування та видалення записів у базі даних;

- перевірка генерації звітів, рахунків та актів звірки;
- тестування пошуку та фільтрації даних.

2) *тестування зручності використання (Usability Testing):*

- оцінка інтуїтивності та зрозумілості інтерфейсу для користувачів;
- перевірка логічності розташування елементів керування та навігаційних елементів;

- тестування часу, необхідного для виконання основних задач користувачами;

- аналіз та врахування зворотнього зв'язку для покращення зручності використання.

3) *тестування безпеки:*

- перевірка механізмів авторизації та аутентифікації, запобігання несанкціонованому доступу до захищених розділів вебзастосунку;

- тестування безпеки введення даних, запобігання атакам типу SQL-ін'єкцій та міжсайтовому скриптингу (XSS);

- перевірка шифрування конфіденційних даних, таких як паролі користувачів;

- тестування безпеки зберігання та передачі даних між клієнтом та сервером;

- перевірка захисту від несанкціонованого доступу до файлів та директорій на сервері.

4) *тестування продуктивності:*

- оцінка швидкості завантаження сторінок та реакції продукту на дії користувача;

- перевірка роботи застосунку під навантаженням, тестування з великою кількістю одночасних користувачів;

- аналіз використання ресурсів сервера, оптимізація продуктивності бази даних та серверної частини;
- тестування швидкості обробки запитів та генерації звітів;
- перевірка роботи вебзастосунку при великих обсягах даних у базі даних.

5) *тестування сумісності:*

- перевірка коректності відображення та функціонування вебсистеми в різних браузерах (Chrome, Firefox, Safari, Edge тощо);
- перевірка адаптивності дизайну та коректності відображення на різних розмірах екрану (настільні комп'ютери, ноутбуки, планшети, смартфони);
- тестування сумісності з різними версіями використовуваних бібліотек та фреймворків.

б) *регресійне тестування:*

- повторне виконання тестів після внесення змін або виправлення помилок у вебзастосунку;
- перевірка того, що нові зміни не призвели до появи нових помилок або несправностей у раніше протестованих частинах застосунку;
- забезпечення стабільності та надійності роботи вебзастосунку при його постійному розвитку та оновленні.

Процес тестування вебзастосунку виконувався як вручну, так і з використанням автоматизованих інструментів. Ручне тестування дозволило перевірити функціональність та зручність використання з точки зору користувача, тоді як автоматизовані тести забезпечили швидку та ефективну перевірку великої кількості сценаріїв та граничних випадків.

Для автоматизації тестування використовувалися такі інструменти, як Selenium WebDriver для функціонального тестування, JMeter для тестування продуктивності та навантаження, а також PHPUnit для модульного тестування серверної частини.

Під час тестування особлива увага приділялася критичним функціям вебзастосунку, таким як обробка замовлень, керування товарами та послугами, фінансові операції та безпека даних. Були розроблені детальні тестові сценарії, які охоплювали різні варіанти використання та можливі виняткові ситуації.

Результати тестування документувалися, включаючи опис виявлених помилок, кроки для їх відтворення та очікувані результати. На основі цих результатів розробники вносили необхідні виправлення та вдосконалення у проєкт.

Окрім функціональних аспектів, тестування також охоплювало перевірку зручності використання та візуальної привабливості інтерфейсу. Було проведено тестування користувачами (User Acceptance Testing), під час якого реальні користувачі взаємодіяли з вебзастосунком та надавали зворотний зв'язок щодо своїх вражень та можливих покращень.

Тестування безпеки включало перевірку захищеності від поширених вразливостей, таких як SQL-ін'єкції, міжсайтовий скриптинг (XSS) та підробка міжсайтових запитів (CSRF). Були проведені тести на проникнення (penetration testing) для виявлення потенційних слабких місць у системі безпеки розробки.

Тестування продуктивності дозволило оцінити швидкість завантаження сторінок, час відгуку сервера та здатність вебзастосунку обробляти велику кількість одночасних запитів. Були проведені стрес-тести для визначення меж продуктивності та виявлення потенційних вузьких місць.

Регресійне тестування виконувалося після кожного внесення змін або виправлення помилок у системі. Це гарантувало, що нові зміни не вплинули негативно на існуючу функціональність та не призвели до появи нових помилок.

Процес тестування був ітеративним та тривав протягом усього циклу розробки вебзастосунку. Кожна ітерація включала планування тестів, їх виконання, аналіз результатів та внесення необхідних змін. Це дозволило забезпечити високу якість та надійність продукту на момент його випуску.

Завдяки ретельному та всебічному тестуванню вдалося виявити та усунути більшість помилок, оптимізувати продуктивність та покращити загальний

користувацький досвід. Це дозволило створити вебзастосунок, який відповідає очікуванням користувачів та забезпечує ефективне управління продажами екотоварів.

Висновки до розділу 3

У третьому розділі дипломної роботи було детально розглянуто процес розробки вебзастосунку з продажу екотоварів. Цей процес включав вибір та обґрунтування технологій розробки, реалізацію серверної частини, розробку інтерфейсу, а також ретельне тестування застосунку.

Вибір технологій розробки був ключовим етапом, який визначив успішність та ефективність створення вебзастосунку. Було проаналізовано та обґрунтовано використання мови програмування PHP для серверної частини, реляційної бази даних MySQL для зберігання даних, а також HTML, CSS та фреймворку Bootstrap для розробки клієнтської частини. Ці технології були обрані завдяки їх популярності, наявності широкого набору інструментів та бібліотек, а також можливості швидкого та ефективного створення вебзастосунків.

Реалізація серверної частини дозволила забезпечити надійну та ефективну обробку запитів, взаємодію з базою даних та генерацію динамічного контенту. Було створено набір функцій у файлі `functions.php`, які відповідають за різні операції, такі як авторизація та аутентифікація користувачів, керування товарами та послугами, контрагентами, угодами, фінансами та адмініструванням.

Розробка інтерфейсу вебзастосунку забезпечила створення привабливого, адаптивного та зручного у використанні інтерфейсу. Було розроблено ряд сторінок, таких як головна сторінка, сторінки керування товарами та послугами, контрагентами, угодами, фінансами, адміністрування та чату. Використання Bootstrap дозволило швидко створити адаптивний дизайн, який коректно відображається на різних пристроях та розмірах екрану.

Тестування вебзастосунку було невід'ємною частиною процесу розробки і дозволило забезпечити його якість, надійність та зручність використання. Було

проведено різні види тестування, включаючи функціональне тестування, тестування зручності використання, тестування безпеки, тестування продуктивності, тестування сумісності та регресійне тестування.

Розроблений вебзастосунок дозволяє ефективно керувати товарами, контрагентами, угодами та фінансовими операціями. Він забезпечує зручний доступ до інформації, автоматизує рутинні завдання та спрощує прийняття рішень.

Подальший розвиток може включати нові функціональні можливості, оптимізацію продуктивності, покращення користувацького досвіду та забезпечення більшої персоналізації. Регулярне оновлення підтримуватиме актуальність продукту.

Загалом, розробка вебзастосунку для управління інтернет-магазином екотоварів виявилася успішною. Завдяки ретельному плануванню, реалізації та тестуванню вдалося створити функціональний та надійний інструмент, який відповідає сучасним стандартам веброзробки та задовольняє потреби користувачів.

ВИСНОВКИ

У рамках кваліфікаційної роботи бакалавра було успішно розроблено повнофункціональний вебзастосунок для управління інтернет-магазином екотоварів. Вебзастосунок покликаний забезпечити ефективно ведення онлайн-бізнесу в сфері продажу екологічних товарів та оптимізувати ключові бізнес-процеси компанії.

Розробці вебзастосунку передували аналіз предметної сфери, дослідження діяльності інтернет-магазинів екотоварів, вивчення існуючих рішень та формування вимог до системи. На основі зібраних вимог було розроблено архітектуру вебзастосунку, що забезпечує розширення та масштабування системи.

Функціонал вебзастосунку охоплює наступні аспекти:

- управління каталогом товарів: додавання, редагування, категоризація та пошук товарів;
- керування контрагентами: ведення бази постачальників та клієнтів, відстеження історії взаємодії, формування персональних пропозицій;
- формування та відстеження угод: створення угод, контроль статусу, генерація супровідних документів;
- управління фінансовими операціями: реєстрація надходжень та витрат, формування фінансових звітів;
- генерація звітів: автоматичне формування рахунків, актів звірки та інших документів;
- адміністрування користувачів: керування обліковими записами, розподіл прав доступу;
- внутрішній чат: можливість оперативної комунікації між співробітниками.

Цей функціонал забезпечує ефективно управління всіма аспектами діяльності інтернет-магазину екотоварів.

Вибір технологій для розробки вебзастосунку базувався на аналізі їх можливостей та відповідності вимогам проекту. Для серверної частини обрано

PHP, популярну мову для веброзробки з широким набором бібліотек і фреймворків. Система управління базами даних – MySQL, надійна та швидка для зберігання структурованих даних. Клієнтська частина реалізована за допомогою HTML, CSS та JavaScript, з використанням фреймворку Bootstrap для швидкої розробки адаптивного інтерфейсу.

Особливу увагу приділено безпеці та оптимізації роботи з базою даних. Використано підготовлені запити для захисту від SQL-ін'єкцій та кешування запитів для підвищення швидкодії. Транзакції забезпечують цілісність даних, а паролі зберігаються у хешованому вигляді для захисту конфіденційності.

Інтерфейс розроблено з урахуванням принципів юзабіліті та естетичної привабливості. Модулі системи мають інтуїтивно зрозумілі сторінки, форми забезпечені валідацією, а навігація адаптована під роль користувача. Кольорова гамма і стиль оформлення підкреслюють природність та екологічність екотоварів. Адаптивна верстка на основі Bootstrap забезпечує зручну роботу на різних пристроях.

Невід'ємним етапом розробки вебзастосунку було ретельне тестування, яке включало перевірку функціональності, надійності, безпеки та зручності використання. Функціональне тестування охопило всі модулі, підтвердивши коректність виконання ключових сценаріїв. Юзабіліті-тестування із залученням користувачів надало цінний зворотний зв'язок щодо зручності інтерфейсу. Тестування безпеки включало перевірку стійкості до поширених вразливостей, таких як SQL-ін'єкції та XSS. Аналіз продуктивності виявив потенційні вузькі місця, які було оптимізовано. Кросбраузерне тестування підтвердило коректність роботи у різних браузерах та на різних операційних системах.

Розроблений вебзастосунок для управління інтернет-магазином екотоварів має значну практичну цінність. Він надає власникам потужний інструмент для автоматизації та оптимізації бізнес-процесів, знижуючи операційні витрати та підвищуючи ефективність роботи персоналу. Зручний інтерфейс мінімізує час на навчання співробітників і знижує ймовірність помилок. Модульна архітектура

забезпечує можливість подальшого розширення та інтеграції зі сторонніми сервісами, такими як системи онлайн-платежів, служби доставки та програми лояльності.

Подальший розвиток вебзастосунку може включати розширення функціональності відповідно до потреб бізнесу. Перспективними напрямками є інтеграція з системами управління складом та бухгалтерського обліку, впровадження елементів штучного інтелекту для персоналізації взаємодії з клієнтами та оптимізації маркетингових кампаній. Регулярний збір зворотного зв'язку від користувачів та аналіз статистики дозволить вдосконалювати вебзастосунок та підтримувати його актуальність.

Таким чином, у ході виконання кваліфікаційної роботи було розроблено високоякісний вебзастосунок для управління інтернет-магазином екотоварів. Він забезпечує високий рівень безпеки, продуктивності та зручності використання. Ретельне проєктування, сучасний технологічний стек та всебічне тестування дозволили створити надійний та ефективний інструмент для онлайн-бізнесу. Вебзастосунок готовий до впровадження та має значний потенціал для подальшого розвитку. Отриманий досвід та навички будуть корисними для майбутніх проєктів у сфері веброзробки та електронної комерції.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Мельник С. Розробка веб-застосунків для електронної комерції. Київ: КПШ ім. Ігоря Сікорського, 2020. 120 с.
2. Чубукова О.Ю. Інтернет-маркетинг та електронна комерція. Київ: КНЕУ, 2019. 280 с.
3. Гужва В.М. Інформаційні системи і технології на підприємствах. Київ: КНЕУ, 2018. 400 с.
4. Олійник А.В., Оліфер В.Г. Комп'ютерні мережі. Принципи, технології, протоколи: Підручник для ВНЗ. 4-те вид. Львів: Магнолія, 2010. 944 с.
5. Буров Є.В. Комп'ютерні мережі. Львів: Магнолія, 2010. 262 с.
6. Berners-Lee T., Fielding R., Frystyk H. Hypertext Transfer Protocol -- HTTP/1.0. RFC 1945. 1996. URL: <https://tools.ietf.org/html/rfc1945> (дата звернення: 16.05.2024).
7. Філіппов І. Рітейл майбутнього: 7 трендів, які змінять торгівлю. RAU. 2021. URL: <https://rau.ua/novyni/riteyl-7-trendiv/> (дата звернення: 16.05.2024).
8. Organic Monitor. The Global Market for Organic Food & Drink: Trends & Future Outlook. 2021.
9. Центр екологічної сертифікації та маркування. Органічна сертифікація. 2023. URL: <https://www.ecolabel.org.ua/organichna-sertifikatsiya> (дата звернення: 16.05.2024).
10. Гриценко Т.В. Веб-технології та веб-дизайн. Львів: Вид-во ЛПУ, 2021. 226 с.
11. Брокшмідт К. Довідник веб-розробника. К.: Фабула, 2021. 752 с.
12. Мейксін М. PHP 8 для початківців. Львів: Магнолія, 2023. 592 с.
13. Домінік С. JavaScript и jQuery. Вичерпне керівництво. К.: Видавнича група BHV, 2019. 688 с.
14. Вандер Вер Д. Вчасно та в рамках бюджету. Управління проектами за методом критичного ланцюга. К.: Фабула, 2019. 360 с.
15. Рамел Д. Joomla! Для професіоналів. Львів: Магнолія, 2021. 448 с.

16. Кузнецов М., Симдянов І. PHP 8. Об'єктно-орієнтоване програмування. Львів: Магнолія, 2023. 608 с.
17. Скляр Д., Треттер А. PHP Cookbook. Рецепти програмування. 3-тє вид. Львів: Магнолія, 2020. 832 с.
18. Кантелон М., Хартер М., Головайчук Т., Райлі Н. Node.js у дії. К.: Фабула, 2019. 432 с.
19. Авінаш К. Веб-аналітика 2.0 на практиці. Тонкощі та найкращі практики. К.: Фабула, 2018. 528 с.
20. Вирин Ф. Інтернет-маркетинг. Повний збірник практичних інструментів. К.: Видавнича група ВНУ, 2021. 320 с.
21. Векшинський А.А., Тивін Л.Ф. Інтернет-маркетинг як новий напрямок у сучасній концепції маркетингу взаємодії. К.: КНЕУ, 2022.
22. Мельник А.І. Особливості розробки веб-застосунків для електронної комерції. Київ: КПП ім. Ігоря Сікорського, 2020.
23. Чубукова О.Ю. Інформаційні системи та технології для електронної комерції. Київ: КНЕУ, 2021.
24. Пономаренко В.С., Клебанова Т.С., Чавалах О.В. Інформаційні системи в економіці. Харків: ХНЕУ ім. С. Кузнеця, 2019. 424 с.
25. Федорчук А.Г. Маркетингові дослідження в Інтернеті. К.: Університет "Україна", 2020. 216 с.
26. Тардаскіна Т.М., Стрельчук Є.М., Терешко Ю.В. Електронна комерція. Одеса: ОНАЗ ім. О.С. Попова, 2021. 244 с.
27. Гаврилов Л.П. Електронна комерція. К.: Центр учбової літератури, 2022. 364 с.
28. Шалева О.І. Електронна комерція. Львів: Новий Світ-2000, 2020. 216 с.
29. Зосімов В.В., Берко А.Ю. Проектування інформаційних систем електронної комерції. Львів: Новий Світ-2000, 2021. 330 с.
30. Заборовський В.П. Основи електронного бізнесу. Ірпінь: Університет ДФС України, 2019. 316 с.

31. Стрій Л.О. Маркетинг ХХІ століття. Концептуальні зміни та тенденції розвитку. Одеса: ВМВ, 2020. 320 с.
32. Дубовик Т.В. Інтернет-торгівля і електронна комерція. К.: Київ. нац. торг. екон. ун-т, 2022. 396 с.
33. Левицька О.О. Сучасні інформаційні системи та технології. Рівне: НУВГП, 2019. 268 с.
34. Воробієнко С.П., Нікітюк Л.А. Проектування інформаційних систем. Одеса: ОНАЗ ім. О.С. Попова, 2021. 124 с.
35. Сендзюк М.А. Інформаційні системи в державному управлінні. К.: КНЕУ, 2020. 288 с.
36. Гужва В.М. Інформаційні системи і технології в міжнародному бізнесі. К.: КНЕУ, 2021. 164 с.
37. Шквір В.Д., Загородній А.Г., Височан О.С. Інформаційні системи і технології в обліку та аудиті. Львів: Львівська політехніка, 2019. 404 с.
38. Марченко Д.М., Марченко О.С. Основи веб-дизайну та HTML. К.: Університет "Україна", 2020. 294 с.

ДОДАТОК А

Програмний код styles.css

```
body {
  background-color: #f5f5f5;
}

.navbar {
  margin-bottom: 15px;
}

.container {
  background-color: #ffffff;
  padding: 20px;
  border-radius: 5px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

footer {
  background-color: #ffffff;
  padding: 10px 0;
  border-top: 1px solid #ddd;
}

.table {
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

.btn-primary {
  background-color: #007bff;
  border-color: #007bff;
  width: auto;
}

.btn-secondary {
  background-color: #6c757d;
  border-color: #6c757d;
}

.nav-link {
  color: #007bff;
}

.navbar-dark .navbar-toggler-icon {
  background-color: #007bff;
}

.task-container {
  display: flex;
  flex-wrap: wrap;
  gap: 15px;
}

.task {
  width: 350px;
  padding: 15px;
  background-color: #f8f9fa;
  border-radius: 5px;
}
```

```
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    margin-bottom: 10px;
    margin-top: 10px;
}
```

```
.low-priority {
    background-color: #dff0d8;
}
```

```
.medium-priority {
    background-color: #fcf8e3;
}
```

```
.high-priority {
    background-color: #f2dede;
}
```

```
.task-bin {
    position: fixed;
    bottom: 20px;
    right: 20px;
    width: 60px;
    height: 60px;
    background-color: #dc3545;
    border-radius: 50%;
    display: flex;
    justify-content: center;
    align-items: center;
    cursor: pointer;
}
```

```
.task-bin i {
    font-size: 24px;
    color: #ffffff;
}
```

```
.task-placeholder {
    height: 20px;
}
```

```
.task-urgent {
    background-color: #f44336;
    color: white;
    font-weight: bold;
}
```

```
html, body {
    height: 100%;
}
```

```
.wrapper {
    display: flex;
    flex-direction: column;
    min-height: 100%;
}
```

```
.content {
    flex: 1;
}
```

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для управління інтернет-магазином екотоварів

```
footer {
  background-color: #f8f9fa;
  border-top: 1px solid #e9ecef;
}

.chat-window {
  height: 400px;
  overflow-y: scroll;
  border: none;
  border-radius: .25rem;
}

.chat-message {
  padding: 0.5rem;
  border-bottom: 1px solid #f0f0f0;
}

.task-message {
  color: #007bff;
}

.task-time {
  color: #28a745;
  font-weight: bold;
}

.chat-form textarea {
  resize: none;
}

.chat-message {
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

ДОДАТОК Б

Програмний код header.php

```

<?php
<!doctype html>
<html lang="uk">
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="css/styles.css">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link rel="stylesheet" href="css/dragula.min.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/select2@4.1.0-
rc.0/dist/css/select2.min.css">
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script> <!-- jQuery
UI -->
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.css">
  <link rel="stylesheet" href="https://bootswatch.com/5/lux/bootstrap.min.css">
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-
rc.0/dist/js/select2.min.js"></script>
  <title>ecoCRM</title>
  <link rel="stylesheet"
href="https://cdn.datatables.net/1.11.3/css/dataTables.bootstrap5.min.css">
  <script src="https://cdn.jsdelivr.net/clipboard.js/1.5.12/clipboard.min.js"></script>
  <script>
    $(document).ready(function() {
      $('.select2').select2();
    });
  </script>
</head>
<body>
  <div class="wrapper">
    <?php if (isLoggedIn()): ?>
    <nav class="navbar navbar-expand-lg navbar-light" style="background-color:
#e3f2fd;">
      <div class="container">
        <a class="navbar-brand" href="index.php">Головна</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
          <ul class="navbar-nav mr-auto">
            <li class="nav-item">
              <a class="nav-link"
href="products_and_services.php">Товари/послуги</a>
            </li>
            <li class="nav-item">

```

```

        <a class="nav-link" href="deals.php">Угоди</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="contactors.php">Контрагенти</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="finance.php">Фінанси</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="chat.php">Чат</a>
    </li>
    <li class="nav-item">
        <a class="nav-link"
href="edit_certificates.php">Екосертифікати</a>
    </li>
    <?php if ($_SESSION['user_id'] <= 10): ?>
        <li class="nav-item">
            <a class="nav-link" href="admin.php">Адміністрування</a>
        </li>
    <?php endif; ?>
</ul>
<ul class="navbar-nav">
    <li class="nav-item">
        <a class="nav-link" href="logout.php">Вихід</a>
    </li>
</ul>
</div>
</div>
</nav>
<?php endif; ?>
<div class="content">

```

ДОДАТОК В

Програмний код footer.php

```
<?php
</div>

<footer class="mt-auto py-3" style="background-color: #e3f2fd;">
  <div class="container">
    <div class="row">
      <div class="col-md-6">
        <h5>ecoCRM</h5>
        <p>&copy; <?php echo date("Y"); ?> Всі права захищено</p>
      </div>

      <div class="col-md-6 text-md-right">
        <h5>Зв'яжіться з нами</h5>
        <a href="#" class="text-dark mr-2"><i class="fab fa-facebook-f"></i></a>
        <a href="#" class="text-dark mr-2"><i class="fab fa-twitter"></i></a>
        <a href="#" class="text-dark mr-2"><i class="fab fa-linkedin-in"></i></a>
        <a href="#" class="text-dark"><i class="fab fa-instagram"></i></a>
      </div>
    </div>
  </div>
</footer>

</div> <!-- Закрyтие div.wrapper -->

<script src="js/bootstrap.bundle.min.js"></script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css" />
<script src="js/main.js"></script>
<script src="js/dragula.min.js"></script>
<script src="js/drag_and_drop.js"></script>
<script src="https://cdn.datatables.net/1.11.3/js/jquery.dataTables.min.js"></script>
<script src="https://cdn.datatables.net/1.11.3/js/dataTables.bootstrap5.min.js"></script>

</body>
</html>
```


ДОДАТОК Г

Програмний код index.php

```

<?php
session_start();
require_once 'php/functions.php';

$incomes = getLastTenDaysIncomes();
$dates = array_column($incomes, 'operation_date');
$amounts = array_column($incomes, 'total_amount');

if (!isLoggedIn()) {
    header('Location: login.php');
    exit;
}

include 'templates/header.php';
?>

<div class="container mt-5">
    <h1 class="mb-4">Інформаційна панель</h1>
    <div class="row">
        <!-- Віджет "Всього в базі" -->
        <div class="col-md-6 mb-4">
            <div class="card border-primary">
                <div class="card-header bg-primary text-white">Всього в базі</div>
                <div class="card-body">
                    <p><a href="contactors.php">Контрагенти:</a> <?=
getCount('contractors') ?></p>
                    <p><a href="products_and_services.php">Товари та послуги:</a> <?=
getCount('products_and_services') ?></p>
                    <p><a href="deals.php">Угоди:</a> <?= getCount('deals') ?></p>
                    <p><a href="finance.php">Фінансові операції:</a> <?=
getCount('financial_operations') ?></p>
                    <p>Користувачів: <?= getCount('users') ?></p>
                </div>
            </div>
        </div>
        <!-- Віджет "Фінанси" -->
        <div class="col-md-6 mb-4">
            <div class="card border-success">
                <div class="card-header bg-success text-white">Фінанси</div>
                <div class="card-body">
                    <p>Всього на рахунку: <?= getCompanyBalance() ?> грн.</p>
                    <p>Витрат: <?= getTotalAmount('Expense') ?> грн.</p>
                    <p>Доходів: <?= getTotalAmount('Income') ?> грн.</p>
                </div>
            </div>
        </div>
        <!-- Віджет графіка -->
        <div class="col-md-6 mb-4">
            <div class="card border-info">
                <div class="card-header bg-info text-white">Надходження за останні 10
днів</div>
                <div class="card-body">
                    <canvas id="incomeChart"></canvas>
                </div>
            </div>
        </div>
    </div>

```


ДОДАТОК Д

Програмний код functions.php

/Функції товарів та послуг

```
function addProductOrService($name, $description, $type, $price, $quantity_in_stock) {
    global $db;

    $query = "INSERT INTO products_and_services (name, description, type, price,
quantity_in_stock)
        VALUES (:name, :description, :type, :price, :quantity_in_stock)";

    $stmt = $db->prepare($query);
    $stmt->execute([
        'name' => $name,
        'description' => $description,
        'type' => $type,
        'price' => $price,
        'quantity_in_stock' => $type == 'Product' ? $quantity_in_stock : 0 // якщо це
послуга, то кількість на складі завжди буде 0
    ]);

    return true;
}

function getProductOrServiceById($id) {
    global $db;

    $query = "SELECT * FROM products_and_services WHERE id = :id";
    $stmt = $db->prepare($query);
    $stmt->execute(['id' => $id]);
    $product_or_service = $stmt->fetch(PDO::FETCH_ASSOC);

    $certificatesQuery = "SELECT certificate_id FROM product_certificates WHERE product_id
= :id";
    $certificatesStmt = $db->prepare($certificatesQuery);
    $certificatesStmt->execute(['id' => $id]);
    $certificates = $certificatesStmt->fetchAll(PDO::FETCH_COLUMN);

    $product_or_service['certificates'] = $certificates;

    return $product_or_service;
}

function updateProductOrService($id, $name, $description, $type, $price,
$quantity_in_stock) {
    global $db;

    $query = "UPDATE products_and_services SET name = :name, description = :description,
type = :type, price = :price, quantity_in_stock = :quantity_in_stock WHERE id = :id";

    $stmt = $db->prepare($query);
    $stmt->execute([
        'id' => $id,
        'name' => $name,
        'description' => $description,
        'type' => $type,
        'price' => $price,
```

```

        'quantity_in_stock' => ($type == 'Product') ? $quantity_in_stock : 0 // якщо це
        послуга, то кількість на складі завжди 0
    ]);

    return true;
}

function isProductOrServiceUsedInDeals($id) {
    global $db;

    $query = "SELECT COUNT(*) as count FROM deals_products_services WHERE product_service_id
= :id";

    $stmt = $db->prepare($query);
    $stmt->execute(['id' => $id]);

    $result = $stmt->fetch(PDO::FETCH_ASSOC);

    return $result['count'] > 0;
}

function deleteProductOrService($id) {
    global $db;

    $query = "DELETE FROM products_and_services WHERE id = :id";

    $stmt = $db->prepare($query);
    $stmt->execute(['id' => $id]);

    return true;
}

//фінанси

function getCompanyAccount() {
    global $db;

    $query = "SELECT
                (SELECT COALESCE(SUM(amount), 0) FROM financial_operations WHERE
operation_type = 'Income')
                -
                (SELECT COALESCE(SUM(amount), 0) FROM financial_operations WHERE
operation_type = 'Expense')
                as balance";

    $stmt = $db->prepare($query);
    $stmt->execute();

    return $stmt->fetch(PDO::FETCH_ASSOC);
}

function getAllFinancialOperations() {
    global $db;

    $query = "SELECT * FROM financial_operations ORDER BY operation_date DESC";
    $stmt = $db->prepare($query);
    $stmt->execute();

    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

```

```

}

function addFinancialOperation($operation_date, $comment, $amount) {
    global $db;

    $query = "INSERT INTO financial_operations (operation_date, operation_type, comment,
amount) VALUES (:operation_date, 'Income', :comment, :amount)";

    $stmt = $db->prepare($query);
    $stmt->execute([
        'operation_date' => $operation_date,
        'comment' => $comment,
        'amount' => $amount
    ]);

    updateCompanyAccountBalance($amount);
    return true;
}

function updateCompanyAccountBalance($amount) {
    global $db;

    $query = "UPDATE company_account SET balance = balance + :amount WHERE id = 1";

    $stmt = $db->prepare($query);
    $stmt->execute(['amount' => $amount]);
}

function addExpense($operation_date, $comment, $amount) {
    global $db;

    $query = "INSERT INTO financial_operations (operation_date, operation_type, comment,
amount) VALUES (:operation_date, 'Expense', :comment, :amount)";

    $stmt = $db->prepare($query);
    $stmt->execute([
        'operation_date' => $operation_date,
        'comment' => $comment,
        'amount' => $amount
    ]);

    decreaseCompanyAccountBalance($amount);
    return true;
}

function decreaseCompanyAccountBalance($amount) {
    global $db;

    $query = "UPDATE company_account SET balance = balance - :amount WHERE id = 1";

    $stmt = $db->prepare($query);
    $stmt->execute(['amount' => $amount]);
}

function getAllDeals() {
    global $db;

    $query = "SELECT * FROM deals";

    $stmt = $db->prepare($query);

```

```

$stmt->execute();

return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

function getDealById($id) {
    global $db;

    $query = "SELECT * FROM deals WHERE id = :id";

    $stmt = $db->prepare($query);
    $stmt->execute(['id' => $id]);

    return $stmt->fetch(PDO::FETCH_ASSOC);
}

function addIncomeByDeal($deal_id, $operation_date) {
    global $db;

    $deal = getDealById($deal_id);
    $amount = $deal['total_amount'];

    // Генеруємо коментар на основі ID угоди
    $comment = "Надходження по угоді ID " . $deal_id;

    // Додаємо поле коментару в SQL-запит
    $query = "INSERT INTO financial_operations (operation_date, deal_id, contractor_id,
operation_type, comment, amount) VALUES (:operation_date, :deal_id, :contractor_id,
'Income', :comment, :amount)";

    $stmt = $db->prepare($query);
    $stmt->execute([
        'operation_date' => $operation_date,
        'deal_id' => $deal_id,
        'contractor_id' => $deal['contractor_id'],
        'comment' => $comment, // Додаємо коментар
        'amount' => $amount
    ]);

    increaseCompanyAccountBalance($amount);
    return true;
}

function increaseCompanyAccountBalance($amount) {
    global $db;

    $query = "UPDATE company_account SET balance = balance + :amount WHERE id = 1";

    $stmt = $db->prepare($query);
    $stmt->execute(['amount' => $amount]);
}

function getFinancialOperationById($id) {
    global $db;

    $query = "SELECT * FROM financial_operations WHERE id = :id";
    $stmt = $db->prepare($query);
    $stmt->execute(['id' => $id]);
}

```

```

    return $stmt->fetch(PDO::FETCH_ASSOC);
}

function updateFinancialOperation($id, $operation_date, $operation_type, $comment, $amount)
{
    global $db;

    $query = "UPDATE financial_operations SET operation_date = :operation_date,
operation_type = :operation_type, comment = :comment, amount = :amount WHERE id = :id";
    $stmt = $db->prepare($query);
    $stmt->execute([
        'id' => $id,
        'operation_date' => $operation_date,
        'operation_type' => $operation_type,
        'comment' => $comment,
        'amount' => $amount
    ]);

    return true;
}

function getFinancialOperationDetailsById($id) {
    global $db;

    $query = "SELECT * FROM financial_operations WHERE id = :id";
    $stmt = $db->prepare($query);
    $stmt->execute(['id' => $id]);

    return $stmt->fetch(PDO::FETCH_ASSOC);
}

function deleteFinancialOperation($id) {
    global $db;

    $query = "DELETE FROM financial_operations WHERE id = :id";
    $stmt = $db->prepare($query);
    $stmt->execute(['id' => $id]);

    return true;
}

//Віджети на головній
function getCount($tableName) {
    global $db;

    $stmt = $db->query("SELECT COUNT(*) FROM $tableName");
    return $stmt->fetchColumn();
}

function getTotalAmount($operationType) {
    global $db;

    $stmt = $db->prepare("SELECT SUM(amount) FROM financial_operations WHERE operation_type
= :operationType");
    $stmt->execute(['operationType' => $operationType]);
    return $stmt->fetchColumn();
}

function getCompanyBalance() {
    global $db;

```

Кафедра інтелектуальних інформаційних систем
Вебзастосунок для управління інтернет-магазином екотоварів

```
$stmt = $db->query("SELECT balance FROM company_account WHERE id = 1");  
return $stmt->fetchColumn();  
}  
  
function getLastTenDaysIncomes() {  
    global $db;  
  
    $query = "  
        SELECT operation_date, SUM(amount) as total_amount  
        FROM financial_operations  
        WHERE operation_type = 'Income' AND operation_date > DATE_SUB(CURDATE(), INTERVAL  
10 DAY)  
        GROUP BY operation_date  
        ORDER BY operation_date ASC  
    ";  
  
    $stmt = $db->query($query);  
    return $stmt->fetchAll(PDO::FETCH_ASSOC);  
}
```