

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

КВАЛІФІКАЦІЙНА РОБОТА

Тема: «Веб-сайт інтернет-магазину з продажу квітів, з використанням багатокритеріальних фільтрів та рекомендаційної системи»

Ступінь вищої освіти – бакалавр

Спеціальність – 122 «Комп'ютерні науки»

Освітня програма «Комп'ютерні науки»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконала: здобувачка 4 курсу
групи КН-21
Валентина ЛАБЗА

Керівник: зав. кафедри комп'ютерних наук
к.е.н., с.н.с, доцент
Сергій МІЧКІВСЬКИЙ

Засвідчую, що кваліфікаційна
робота оформлена відповідно
до ДСТУ 3008:2015 та не
містить запозичень з праць
інших авторів без відповідних
посилань.

Здобувач: _____
(підпис)

м. Київ – 2025 рік

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»**

ЗАТВЕРДЖУЮ:

завідувач кафедри комп'ютерних наук

_____ Сергій МІЧКІВСЬКИЙ

«_____» _____ 20__р

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Лабза Валентина Олександрівна

Тема роботи	Веб-сайт інтернет-магазину з продажу квітів, з використанням багатокритеріальних фільтрів та рекомендаційної системи
Номер та дата наказу про затвердження теми	№ 24-1 від 14 березня 2025 року
Коротка постановка завдання	Розробка веб-сайту інтернет-магазину з продажу квітів. На сайті реалізувати використання багатокритеріальних фільтрів та рекомендаційну систему підбору квітів.
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	Федько М.Р. Дослідження методів побудови рекомендаційних систем для вирішення задач в електронній комерції, методи колаборативної фільтрації. // International Electronic Scientific Journal “Science Online”, 2021. URL https://uk.legacy.reactjs.org/ (дата звернення: 20.12.2023) Шаповалов С.М., Мічківський С.М. Розробка рекомендаційної системи для підбору співрозмовника в соціальній мережі // XII Міжнародна науково-технічна конференція «Проблеми Інформатизації» (м. Київ, 13 грудня 2018 року). – К.: Державний університет телекомунікацій, 2018. URL: https://dut.edu.ua/uploads/1_1701_65845854.pdf (дата звернення: 20.12.2023)
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	Кваліфікаційна робота має передбачити теоретичне, системотехнічне або експериментальне дослідження складного спеціалізованого завдання або практичної проблеми в галузі комп'ютерних наук, яке характеризується комплексністю та невизначеністю умов і потребує застосування теорій і методів інформаційних технологій.

Дата видачі завдання 11 березня 2025 р.

Керівник

Сергій МІЧКІВСЬКИЙ

Здобувач освітнього ступеня бакалавра

Валентина ЛАБЗА

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання	Примітка
Підготовчий етап			
1	Вибір напрямку дослідження	01.03.2025 р.	<i>виконано</i>
2	Формування теми та призначення керівника	05.03.2025 р.	<i>виконано</i>
3	Затвердження теми кваліфікаційної роботи	14.03.2025 р.	<i>виконано</i>
4	Затвердження завдання на кваліфікаційну роботу	14.03.2025 р.	<i>виконано</i>
Основний етап			
5	Розробка концепції кваліфікаційної роботи	15.03.2025 р.	<i>виконано</i>
6	Підбір та вивчення джерел інформації з напрямку дослідження. Огляд існуючих аналогів	20.03.2025 р.	<i>виконано</i>
7	Затвердження розширеної постановки завдання. Підготовка та подання керівникові розділу 1 кваліфікаційної роботи	23.03.2025 р.	<i>виконано</i>
8	Проектування. Підготовка та подання керівникові розділу 2 кваліфікаційної роботи	24.03.2025 р.	<i>виконано</i>
9	Підготовка доповіді для експертизи стану виконання кваліфікаційної роботи (проміжний контроль)	31.03-04.04.2025 р.	<i>виконано</i>
10	Реалізація. Підготовка та подання керівникові розділу 3 кваліфікаційної роботи	07.04.2025 р.	<i>виконано</i>
11	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи	14.04.2025 р.	<i>виконано</i>
12	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника та представлення керівникові доопрацьованого варіанту кваліфікаційної роботи	21.04.2025 р.	<i>виконано</i>
Завершальний етап			
13	Представлення рукопису для перевірки на плагіат	28.04-04.05.2025 р.	<i>виконано</i>
14	Підготовка презентації та доповіді на передзахист	05.05-11.05.2025 р.	<i>виконано</i>
15	Передзахист кваліфікаційної роботи	12.05-16.05.2025 р.	<i>виконано</i>
16	Доопрацювання роботи за результатами передзахисту	19.05-06.06.2025 р.	<i>виконано</i>
17	Експертиза роботи керівником та зовнішнім експертом	09.06-15.06.2025 р.	<i>виконано</i>
18	Доопрацювання доповіді та презентації для захисту	09.06-15.06.2025 р.	<i>виконано</i>
19	Захист кваліфікаційної роботи	16.06-22.06.2025 р.	<i>виконано</i>

Керівник

Сергій МІЧКІВСЬКИЙ

Здобувач освітнього ступеня бакалавра

Валентина ЛАБЗА

Лабза В.О. Вебсайт інтернет-магазину з продажу квітів, з використанням багатокритеріальних фільтрів та рекомендаційної системи.

Пояснювальна записка кваліфікаційної роботи за спеціальністю 122 – Комп’ютерні науки (освітня програма – Комп’ютерні науки) СО Бакалавр. – ВНЗ .Університет економіки та права .КРОК., Навчально-науковий інститут інформаційних та комунікаційних технологій, кафедра комп’ютерних наук, Київ, 2025.

Описано розробку веб-застосунку інтернет-магазину з продажу квітів, який забезпечує візуалізація та управління веб-контентом інтернет магазину та реалізує багатокритеріальні фільтри та рекомендаційну систему.

Ключові слова: Web-розробка, інтернет-магазин, бази даних, React.

Рис. 20. Бібліограф.: 26 найм.

Labza V.O. Website of an online store for selling flowers with multi-criteria filters and a recommendation system.

Explanatory note of qualification work in specialty 122 - Computer Science (educational programme - Computer Science), Bachelor's degree - University of Economics and Law "KROK", Educational and Research Institute of Information and Communication Technologies, Department of Computer Science, Kyiv, 2025.

The development of a web application for an online flower shop is described, providing visualization and management of the store's web content while implementing multi-criteria filters and a recommendation system.

Keywords: Web development, online store, databases, React.

Fig. 20. Bibliography: 26 Items.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ВЕБ-САЙТУ	8
1.1 Опис предметної області.....	8
1.2 Визначення потенційних конкурентних переваг	9
1.3 Постановка завдання на кваліфікаційну роботу	12
Висновки до розділу 1	14
РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-САЙТУ	16
2.1 Моделювання поведінки продукту.....	16
2.2 Моделювання структури продукту.....	18
2.3 Опис архітектури продукту	21
Висновки до розділу.....	23
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБ-САЙТУ	24
3.1 Реалізація та конструювання програмного продукту.....	24
3.2 Тестування програмного продукту.....	30
3.3 Використання програмного продукту	32
Висновки по розділу 2.....	36
ВИСНОВКИ	37
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	38

ВСТУП

Актуальність теми. Сфера продажу квітів та кімнатних рослин продовжує активно розвиватися й адаптуватися до сучасних умов ринку. Відповідно до тенденцій останніх років [1], попит на квіткову продукцію стабільно зростає, особливо в періоди свят та важливих подій. Все більше людей обирають замовлення букетів та рослин онлайн, що зумовлено зручністю, швидкістю обслуговування та широким вибором продукції в інтернет-магазинах.

Попри це, значна частина локальних квіткових магазинів усе ще не має сучасного веб-сайту або використовує застарілі рішення, які не відповідають потребам користувачів. Крім того, більшість наявних онлайн-магазинів не пропонують достатньо гнучких засобів фільтрації, що ускладнює пошук товарів за конкретними критеріями: типом квітів, кольором, подією, бюджетом чи способом доставки.

Таким чином, створення веб-сайту інтернет-магазину з продажу квітів та кімнатних рослин, який враховує потреби користувачів завдяки багатокритеріальним фільтрам та системі персональних рекомендацій, є актуальним, інноваційним і важливим завданням у сучасних умовах розвитку електронної комерції.

Мета дослідження. Метою проекту є розробка програмного забезпечення веб-сайту інтернет-магазину з продажу квітів з використанням багатокритеріальних фільтрів та рекомендаційної системи, що дозволить користувачам замовляти товари та отримувати рекомендації.

Завдання дослідження:

- 1) провести аналіз аналогічних вебсайтів інтернет-магазинів з продажу квітів та кімнатних рослин;
- 2) визначити вимоги до системи;
- 3) дослідити підходи до реалізації рекомендаційних систем;
- 4) спроектувати архітектуру вебзастосунку з урахуванням сучасних принципів розробки;

5) розробити структуру бази даних для збереження інформації про товари, користувачів та їхню активність;

б) реалізувати програмне забезпечення застосунку.

Об'єктом дослідження є процес продажу квітів за допомогою інтернет магазину з використанням багатокритеріальних фільтрів і рекомендаційної системи.

Предметом дослідження є веб-сайти, інтернет-магазини, продаж квітів, багатокритеріальні фільтри та рекомендаційні системи.

Методи дослідження: теоретичною й методологічною основою дослідження є підходи та методи до розробки веб-сайтів, реалізації багатокритеріальних фільтрів, а також базові принципи побудови рекомендаційних систем на основі поведінки користувачів.

Практичне значення. Розробка веб-сайту інтернет-магазину з продажу квітів, з використанням багатокритеріальних фільтрів та рекомендаційної системи

Структура та обсяг пояснювальної записки. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків та списку посилань (26 найменувань). Пояснювальна записка містить 20 рисунків. Загальний обсяг пояснювальної записки складає 40 сторінки, основний зміст викладено на 37 сторінках.

РОЗДІЛ 1.

ПОСТАНОВКА ЗАВДАННЯ НА РОЗРОБКУ ВЕБ-САЙТУ

1.1 Опис предметної області

У сучасному світі все більше людей надає перевагу онлайн-покупкам [2] і ринок продажу квітів не є винятком. Інтернет-магазини квітів і кімнатних рослин стали важливим каналом продажу для флористичних компаній, оскільки дозволяють охопити ширшу аудиторію, забезпечити швидке обслуговування та зручний вибір продукції. Веб-сайт інтернет-магазину квітів надає користувачеві можливість у зручний для нього час переглядати каталог товарів, дізнаватися про характеристики букетів або рослин, здійснювати покупку, обирати спосіб доставки та оплати – усе це онлайн.

Особливістю продажу квітів є їхня емоційна складова: покупка часто здійснюється для подарунка або важливої події, що вимагає швидкого та точного вибору. Через велику кількість варіантів користувачу складно швидко знайти відповідний товар без ефективного інтерфейсу та зручних інструментів для фільтрації. Багатокритеріальні фільтри дозволяють обрати товар за рядом параметрів – типом квітів, кольором, приводом, ціною, наявністю доставки тощо.

Крім того, зростає популярність кімнатних рослин [3] – як засобу створення затишку вдома або в офісі. Цей інтерес значною мірою зумовлений поширенням віддаленого формату роботи, коли люди прагнуть облаштувати комфортний простір у власному житті. Така тенденція розширює сферу застосування подібних веб-ресурсів і дозволяє залучати нову аудиторію, зацікавлену не лише у подарункових букетах, а й у декоративному озелененні інтер'єру.

Інтернет-магазин у цій сфері – це не просто засіб продажу, а інформаційна система зі зручним інтерфейсом, фільтрами, та рекомендаціями. Розробка таких рішень є актуальною для автоматизації продажів і покращення взаємодії з клієнтами.

1.2 Визначення потенційних конкурентних переваг

Для оцінки існуючих рішень було проаналізовано декілька веб-сайтів які спеціалізуються на продажі квітів.

Першим було розглянуто інтернет-магазин Flowers.ua (рис. 1.1) – національна служба доставки квітів і подарунків по Україні та світу.

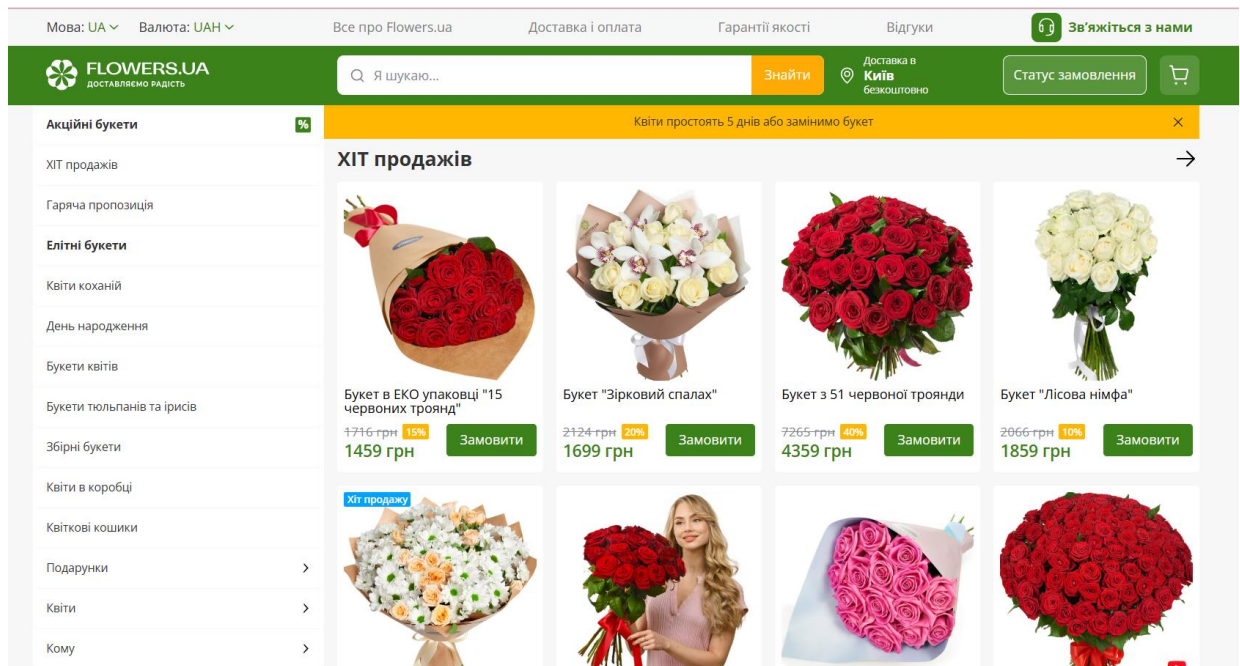


Рисунок 1.1 – Інтернет-магазин *Flowers.ua*

Джерело: [4]

З переваг можна виділити:

- широкий асортимент, є великий вибір букетів для різних подій;
- додаткові послуги, окрім квітів на сайті також доступні інші подарунки, такі як м'які іграшки, солодоші та повітряні кульки, що дозволяє клієнтам створювати більш комплексні подарункові набори;

- при перегляді окремого товару користувачу пропонуються схожі позиції, що може допомогти з вибором;

недоліки:

– користувач може лише обирати товари за загальними категоріями або сортувати за ціною, можливості відфільтрувати букети за параметрами, такими як колір, вид квітів, тематика чи розмір – відсутні;

– дизайн сайту виглядає дещо застарілим та перевантаженим, що може ускладнювати навігацію для нових відвідувачів.

Другим аналогом було розглянуто інтернет-магазин Ukrflora.ua (рис 1.2) – відомий український бренд, що спеціалізується на імпорті та продажу квітів, букетів і кімнатних рослин.

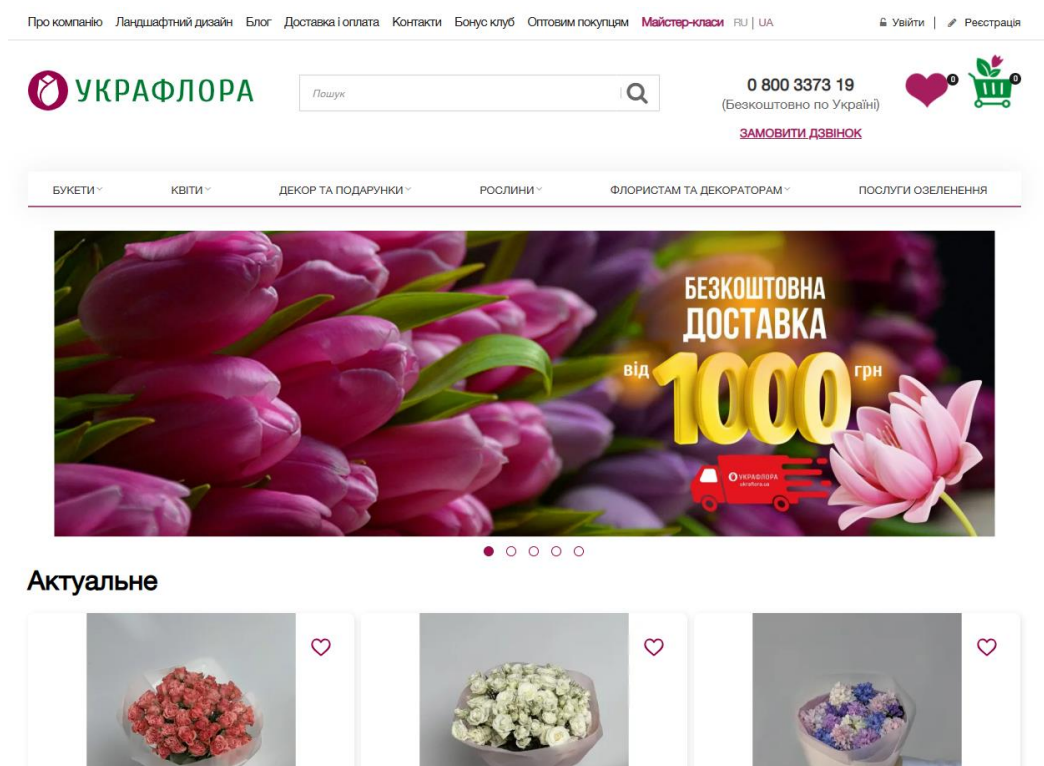


Рисунок 1.2 – Інтернет-магазин Ukrflora.ua

Джерело: [5]

Переваги:

– широкий асортимент – на сайті представлено не лише букети, а й кімнатні рослини, квіткові композиції, подарункові набори та супутні товари;

– товари структуровані за типами, кольорами, новинками та акційними пропозиціями, що полегшує пошук;

– пропонується кур'єрська доставка, оформлення листівок, упаковка тощо, що дозволяє формувати повноцінне замовлення для подарунку;

недоліки:

– попри добре структуровані категорії, фільтрації реалізовані не для всіх категорій товарів і не охоплюють усі важливі параметри (наприклад, умови освітлення чи режим поливу), що обмежує зручність підбору продукції;

– інтерфейс місцями незручний – на деяких елементах немає візуального відгуку при наведенні (hover-ефекту), тому користувачу не завжди зрозуміло, що елемент є клікабельним, це може ускладнювати взаємодію з сайтом;

– технічні проблеми – обрані фільтри не завжди коректно відображаються, що може викликати плутанину у користувача. Крім того, завантаження товарів при фільтрації відбувається повільно, що впливає на загальну швидкість взаємодії з сайтом;

– сайт не пропонує рекомендацій на основі попередніх переглядів користувача або схожих товарів, що могло б зробити процес вибору більш зручним.

Третім біло розглянуто сайт ProFlowers.ua (рис. 1.3) – інтернет-магазин доставки квітів та подарунків, що пропонує широкий асортимент букетів та композицій для різних подій.

Переваги:

– сайт пропонує великий вибір квіткових композицій, включаючи букети з троянд, тюльпанів, орхідей та інших квітів, а також подарункові кошики та солодощі;

– магазин надає безкоштовну доставку квітів у межах Києва, що є привабливим для клієнтів;

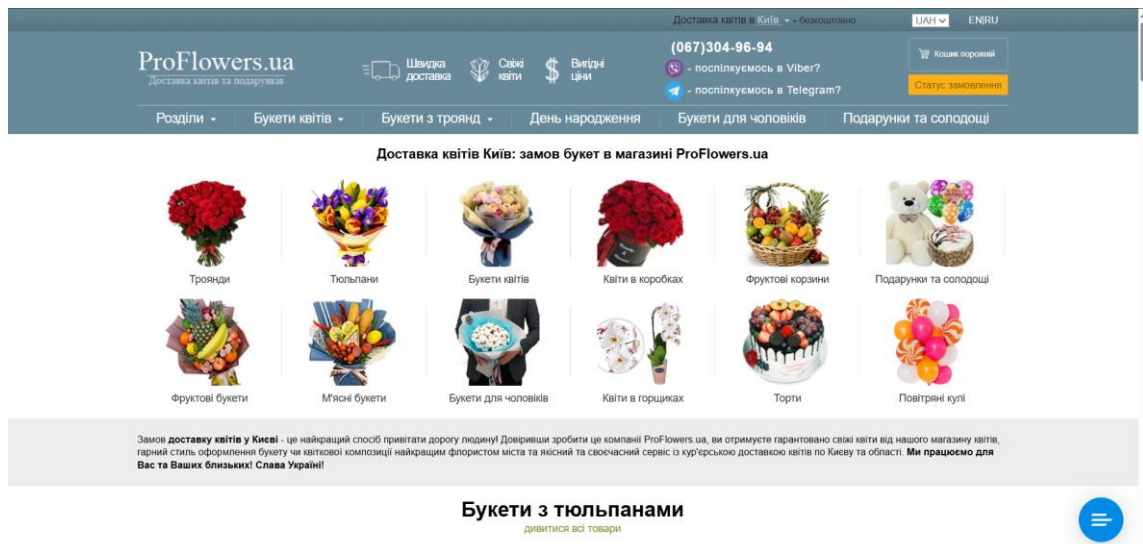


Рисунок 1.3 – Інтернет-магазин Proflowers.ua

Джерело: [6]

недоліки:

- сайт дозволяє вибрати лише один фільтр та встановити ціновий діапазон, що ускладнює пошук товарів за кількома параметрами одночасно;
- зображення товарів можуть завантажуватися повільно, що негативно впливає на зручність користування сайтом.

Отже, порівняння з наявними інтернет-магазинами показало, що більшість із них мають обмежені можливості фільтрації товарів, відсутність персоналізованих рекомендацій і не завжди зручний інтерфейс. Завдяки використанню сучасних технологій і продуманому дизайну, майбутній продукт буде конкурентоспроможним та комфортним для користувачів, забезпечуючи ефективний і приємний процес вибору та покупки товарів.

1.3 Постановка завдання на кваліфікаційну роботу

Основним завданням кваліфікаційної роботи є розробка веб-сайту інтернет-магазину з продажу квітів і кімнатних рослин з реалізацією багатокритеріальних фільтрів та рекомендаційної системи. Метою є створення зручного інструменту для онлайн-покупки, який забезпечить ефективний

пошук товарів за різними критеріями та покращить взаємодію з користувачами шляхом персоналізованих рекомендацій.

Функціональні вимоги:

- перегляд каталогу товарів;
- пошук і фільтрація за типом квітів, кольором, подією тощо;
- перегляд повної інформації про товар (опис, фото, характеристики, ціна);
- перегляд рекомендацій товарів;
- оформлення замовлення через зручну форму.

Нефункціональні вимоги:

- швидке завантаження сторінок;
- стабільна робота при великій кількості товарів;
- простий та зрозумілий інтерфейс;
- безпечне збереження та обробка введених даних при оформленні замовлення.

Вхідні дані:

- дані про товари: назва, опис, ціна, категорія, фотографії, наявність, характеристики ;
- дані замовлення: ім'я, прізвище, номер телефону, адреса доставки, коментар, обрані товари, спосіб доставки та оплати;
- критерії для фільтрації: категорії, тип квітів, колір, ціновий діапазон, подія, особливості догляду тощо.

Вихідні дані:

- список товарів, відібраний відповідно до вказаних фільтрів;
- дані про товари: назва, опис, ціна, категорія, фотографія, наявність, характеристики
- дані замовлення: номер замовлення, ім'я, прізвище, номер телефону, адреса доставки, коментар, обрані товари, спосіб доставки та оплати;
- рекомендовані товари на основі попередніх переглядів.

Основні функціональні можливості системи:

- пошук товарів;
- фільтрація товарів за критеріями;
- перегляд товарів у зручному форматі;
- перегляд рекомендацій товарів;
- оформлення замовлення через форму;
- формування номера замовлення.

Основні етапи реалізації веб-сайту:

1. аналіз аналогічних рішень та визначення вимог до системи;
2. проектування структури сайту;
3. моделювання даних;
4. моделювання процесів;
5. конструювання системи;
6. Розробка клієнтської та серверної частини;
7. проведення тестування;
8. підготовка документації та опис функціональних можливостей системи.

Висновки до розділу 1

У розділі проведено аналіз предметної області онлайн-торгівлі квітами та кімнатними рослинами. Вивчено особливості галузі, актуальні підходи до розробки веб-застосунків, а також сучасні вимоги користувачів до зручності пошуку та персоналізації товарів.

Розглянуто приклади існуючих інтернет-магазинів, виявлено їхні переваги й недоліки, зокрема відсутність повноцінних фільтрів, обмежений функціонал рекомендацій та недосконалий інтерфейс. Це дозволило чітко окреслити напрямки покращення в межах майбутньої системи.

Сформульовано постановку завдання на кваліфікаційну роботу: визначено вхідні та вихідні дані, описано основні етапи реалізації веб-сайту. Отримані результати створюють основу для подальшої практичної розробки

конкурентоспроможного інтернет-магазину, орієнтованого на зручність користувачів та ефективність онлайн-продажу.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-САЙТУ

2.1 Моделювання поведінки продукту

Одним із основних інструментів для моделювання функціональності програмного продукту є *діаграма прецедентів (Use Case Diagram)*. Ця діаграма (рис. 2.1) відображає основні сценарії взаємодії користувача з веб-сайтом [7] інтернет-магазину квітів. Основною метою такого моделювання є визначення ролі користувача, його потреб та способів досягнення цілей за допомогою системи.

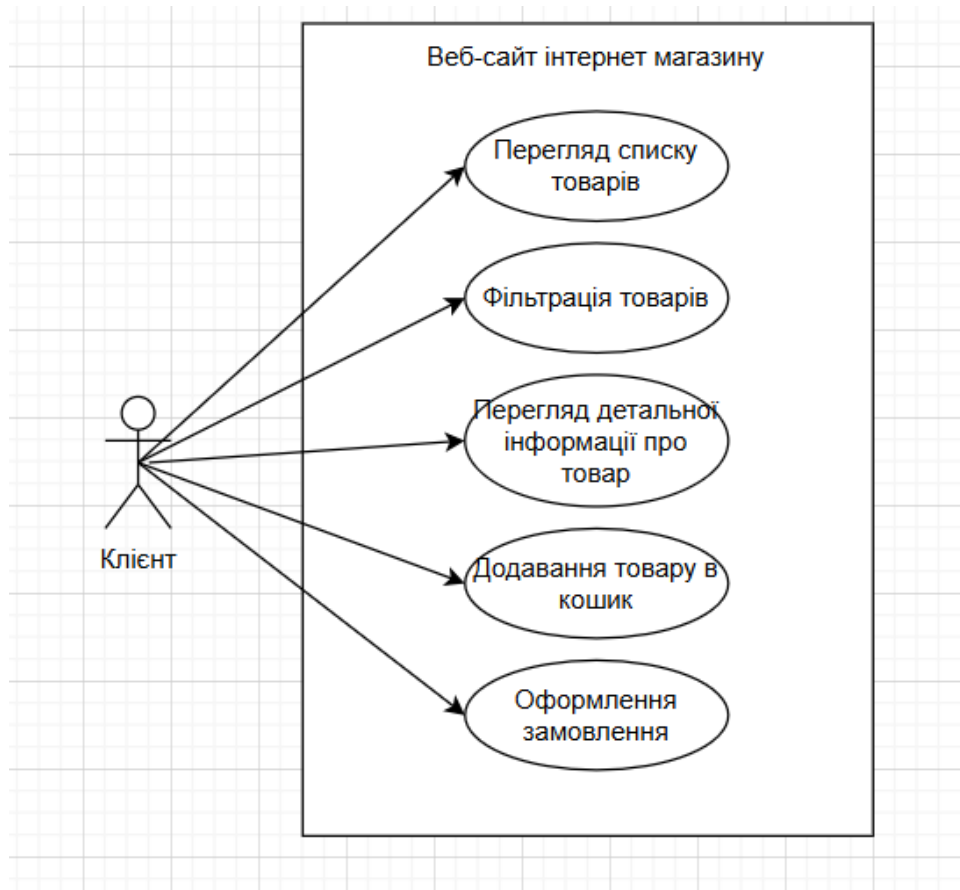


Рисунок 2.1 – Діаграма прецедентів

Джерело: розроблено автором

На діаграмі прецедентів зображено актора – Клієнта, який має доступ до основних функцій сайту:

- перегляд списку товарів – дозволяє клієнту ознайомитися з наявними букетами та кімнатними рослинами;
- фільтрація товарів – надає можливість застосовувати багатокритеріальні фільтри для швидкого пошуку потрібного товару (за типом, кольором, ціною тощо);
- перегляд детальної інформації про товар – відображає повну інформацію про конкретний товар: опис, зображення, ціну, доступність тощо;
- додавання товару в кошик – дає змогу обрати товар для подальшого замовлення;
- оформлення замовлення – процес надсилання форми замовлення без потреби у реєстрації, користувач залишає свої контактні дані та отримує номер замовлення.

Усі ці дії демонструють поведінкову модель взаємодії клієнта з системою, що є важливою частиною проектування функціональності веб-застосунку.

Діаграма діяльності (Activity Diagram) (рис. 2.2) є графічним представленням послідовності дій користувача під час взаємодії з веб-сайтом [8] інтернет-магазину квітів. Такий тип діаграм використовується для візуалізації процесів у системі, що дає змогу краще зрозуміти логіку поведінки користувача та внутрішню структуру дій, які виконує система у відповідь.

У даному випадку змодельовано сценарій покупки товару. Користувач починає взаємодію з системою з головної сторінки сайту, де переходить до перегляду товарів. При потребі він може скористатися фільтрами або пошуком для точнішого вибору. Після ознайомлення з характеристиками обраного товару користувач має можливість додати його до кошика. Далі система пропонує вибір: продовжити перегляд інших товарів або перейти до оформлення замовлення. Якщо користувач обирає оформлення, він заповнює

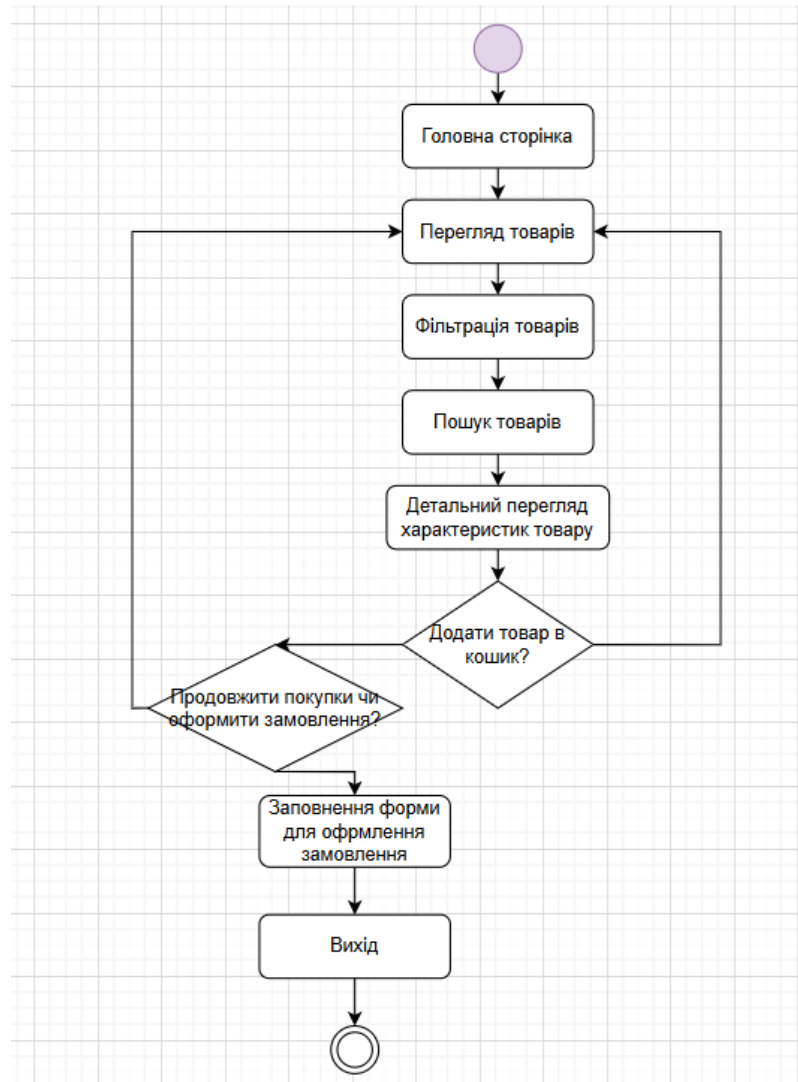


Рисунок 2.2 – Діаграма діяльності

Джерело: розроблено автором

форму з контактними даними. У протилежному випадку повертається до каталогу товарів для подальшого перегляду.

Ця діаграма дозволяє наочно представити основні кроки у процесі покупки, починаючи від першого контакту з сайтом і завершуючи оформленням замовлення.

2.2 Моделювання структури продукту

Діаграма класів (рис. 2.3) використовується для моделювання структури програмного продукту та візуалізації взаємозв'язків між основними

сутностями системи [9]. Вона показує класи, їх атрибути та зв'язки, що відображають логіку зберігання та обробки даних в інтернет-магазині квітів.

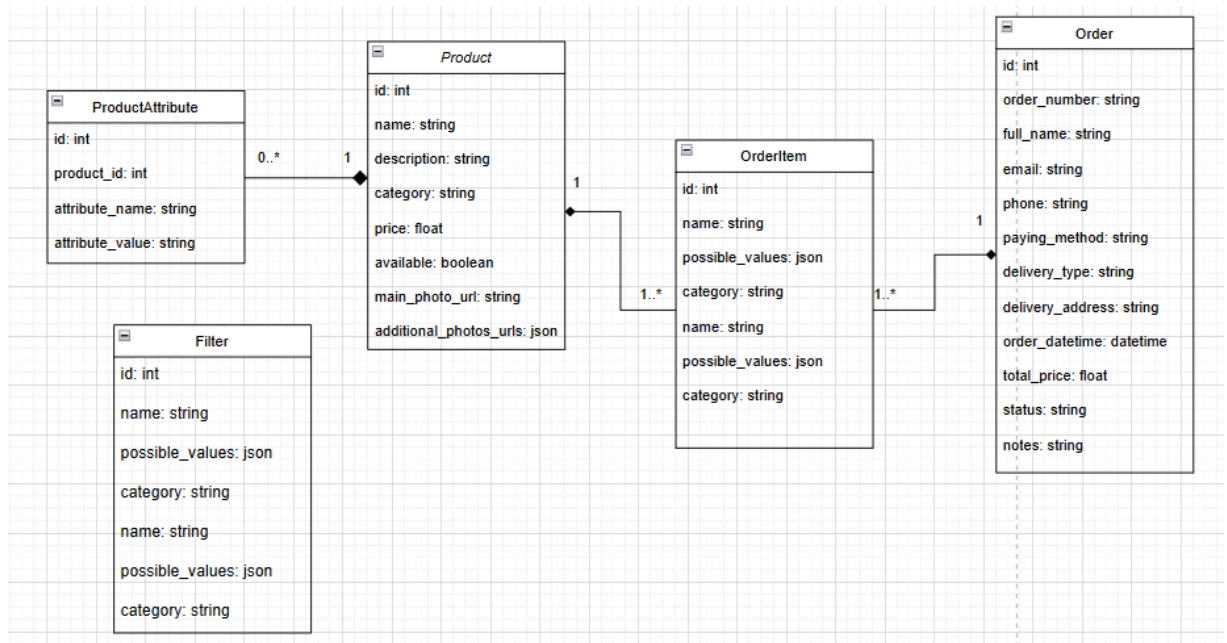


Рисунок 2.3 – Діаграма класів

Джерело: розроблено автором

У проекті реалізовано архітектуру з використанням ORM (Object-Relational Mapping) – зокрема, SQLAlchemy [10]. Це дало змогу представити сутності бази даних у вигляді Python-класів, які безпосередньо відображаються на діаграмі класів.

На діаграмі представлені ключові класи:

- Product, що містить інформацію про товар;
- ProductAttribute – зберігає характеристики товару;
- Filter – набір можливих фільтрів для багатокритеріального пошуку;
- Order – інформація про замовлення;
- OrderItem – окремі позиції в замовленні.

Кожен клас має набір атрибутів, які відповідають полям таблиць у базі даних. Зв'язки між класами показують логічну залежність: один товар може мати багато атрибутів, одне замовлення – багато позицій, і кожна позиція в

замовленні пов'язана з певним товаром. Таким чином, діаграма класів відображає логічну модель даних, що лежить в основі функціонування веб-застосунку, та є основою для побудови взаємодії між компонентами системи.

Діаграма об'єктів (рис. 2.4) це тип UML-діаграм, який відображає об'єкти системи у конкретний момент часу разом з їхніми атрибутами та значеннями [11]. Вона є конкретизацією діаграми класів і дозволяє наочно побачити, як саме взаємодіють елементи системи в реальному сценарії.

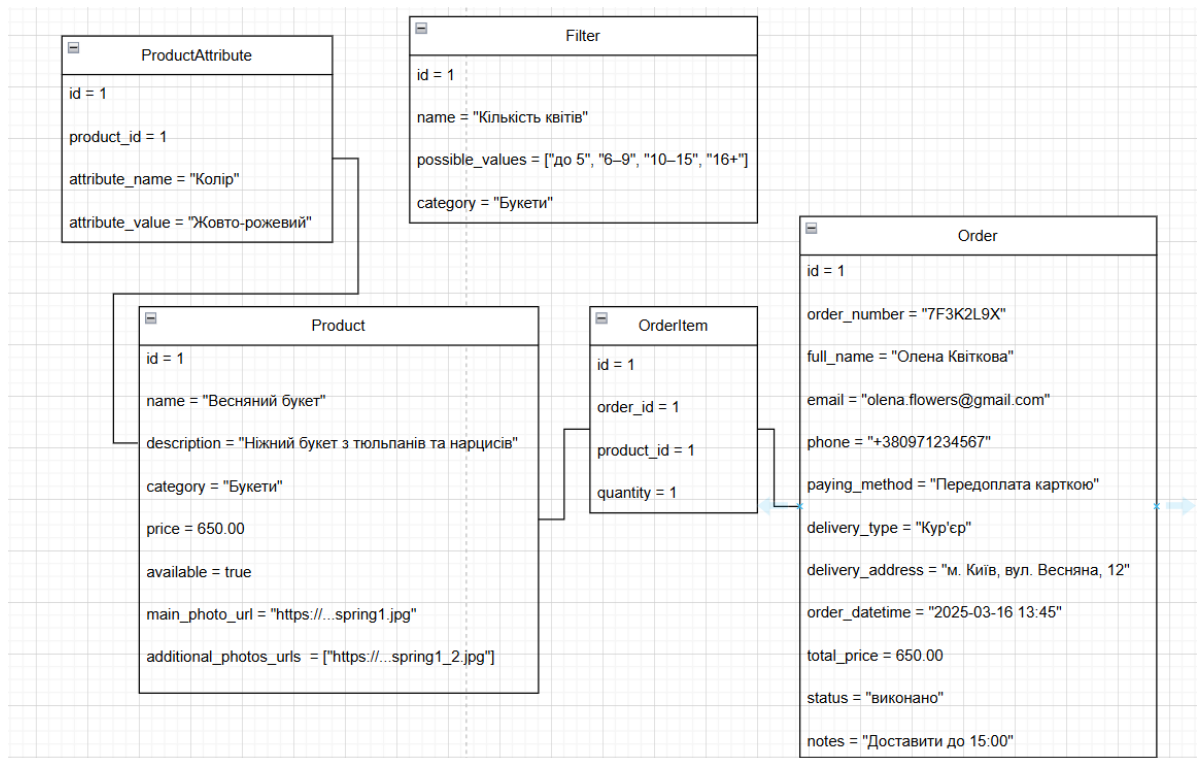


Рисунок 2.4 – Діаграма об'єктів

Джерело: розроблено автором

У представленій діаграмі об'єктів змодельовано типовий сценарій оформлення замовлення в інтернет-магазині квітів. Зокрема, показано об'єкт товару (букет), його атрибут (наприклад, колір), об'єкт замовлення з відповідними даними користувача, позицію замовлення та об'єкт фільтра для кількості квітів. Це дозволяє краще зрозуміти структуру даних та взаємозв'язки між об'єктами в процесі оформлення замовлення.

2.3 Опис архітектури продукту

Архітектура веб-сайту інтернет-магазину квітів побудована за класичною моделлю клієнт-серверної взаємодії [12]. Така архітектура передбачає чітке розділення функцій між клієнтською та серверною частинами, а також централізоване зберігання даних у базі даних. Це рішення є оптимальним для забезпечення масштабованості, стабільності роботи системи та зручності її подальшого супроводження.

Система складається з трьох основних логічних рівнів [24]:

- 1) клієнтська частина (інтерфейс користувача);
- 2) серверна частина (бізнес-логіка);
- 3) база даних (сховище інформації).

Користувач взаємодіє з інтерфейсом, переглядає каталог товарів, застосовує фільтри, формує замовлення. Усі ці дії формують запити, які передаються на сервер для обробки. Серверна частина отримує запити, виконує відповідні обчислення або вибірку даних, після чого формує відповіді для клієнта. Сервер взаємодіє з базою даних [13] для зберігання та отримання інформації про товари, замовлення, характеристики тощо.

Передача даних між клієнтом і сервером здійснюється через стандартизовані запити, що дозволяє забезпечити ефективний обмін інформацією.

Клієнтська частина містить набір основних компонентів, які визначають функціональні можливості сайту:

- 1) каталог товарів – відображає список доступної продукції з можливістю перегляду основних характеристик;
- 2) панель фільтрації – забезпечує застосування багатокритеріальних фільтрів для зручного пошуку товарів;
- 3) сторінка товару – надає детальну інформацію про вибраний товар.
- 4) форма замовлення – дозволяє користувачу оформити покупку, залишивши необхідні контактні дані;

5) блок рекомендацій – відображає товари, які можуть зацікавити користувача на основі його попередніх переглядів.

Ці компоненти формують основу інтерфейсу, забезпечують зручну навігацію та оптимізують процес вибору товарів.

Серверна частина системи забезпечує обробку запитів, реалізацію бізнес-логіки та взаємодію з базою даних. В її складі можна виділити такі основні модулі:

1) список товарів з фільтрацією – приймає параметри пошуку та повертає користувачеві результати відповідно до заданих критеріїв;

2) детальна інформації про товар – надає повний опис вибраного товару для відображення на клієнтській частині;

3) створення замовлень – відповідає за прийом та збереження інформації про нові замовлення;

4) рекомендаційний модуль – формує перелік рекомендованих товарів для користувача на основі його взаємодії з сайтом.

Ці модулі дозволяють забезпечити цілісну роботу веб-застосунку, гарантуючи коректну обробку запитів та взаємодію з даними.

Сховище даних виконує роль центрального джерела збереження інформації про товари, характеристики, замовлення. Структура бази даних розроблена таким чином, щоб забезпечити ефективну роботу з великими обсягами даних, швидку вибірку інформації та збереження цілісності даних.

Усі компоненти системи працюють у тісній взаємодії:

– клієнтська частина формує запити згідно з діями користувача;

– серверна частина обробляє запити, звертається до бази даних, здійснює необхідні обчислення та повертає результати;

– база даних зберігає структуровану інформацію та забезпечує доступ до актуальних даних для інших компонентів.

Така архітектурна модель забезпечує високу продуктивність, гнучкість у масштабуванні та полегшує інтеграцію нових функціональних можливостей у майбутньому.

Висновки до розділу 2

У другому розділі виконано проектування веб-сайту інтернет-магазину квітів, що включає моделювання поведінки користувачів, структури даних та архітектури системи.

За допомогою діаграми прецедентів визначено основні сценарії взаємодії користувача з системою: вибір товарів, фільтрація, перегляд детальної інформації, оформлення замовлення. Діаграма діяльності описала логіку процесу покупки – від перегляду товарів до оформлення замовлення.

Структурне моделювання виконано за допомогою діаграми класів та об'єктів. Ці моделі показали, які дані зберігаються в системі (товари, замовлення, фільтри), як вони пов'язані та використовуються у роботі сайту. Це дозволило визначити склад і організацію даних, необхідних для функціонування інтернет-магазину.

Описано архітектуру сайту на основі клієнт-серверної моделі з розподілом функцій між клієнтською частиною, сервером та базою даних. Виділено основні компоненти інтерфейсу та ключові модулі сервера, що відповідають за виконання функцій, обробку запитів і роботу з даними.

Результати проектування створюють основу для подальшої реалізації веб-сайту інтернет-магазину квітів, орієнтованого на зручність користування та ефективність пошуку товарів.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ВЕБСАЙТУ

3.1 Реалізація та конструювання програмного продукту

Розробка веб-сайту інтернет-магазину квітів здійснена з використанням сучасних технологій, які забезпечують високу продуктивність, масштабованість та зручність у розробці й супроводженні програмного продукту. При виборі засобів реалізації враховувалися актуальні тенденції веб-розробки, специфіка завдання та вимоги до функціональності системи.

Клієнтська частина сайту реалізована за допомогою бібліотеки React , яка є однією з найпопулярніших для створення інтерактивних користувацьких інтерфейсів [14]. Основною перевагою React є компонентний підхід, що дозволяє розділити інтерфейс на незалежні частини та забезпечує їх повторне використання.

Для організації маршрутизації сторінок використано бібліотеку React Router [15], яка забезпечує плавну навігацію між секціями сайту без перезавантаження сторінки (рис 3.1).

```
import { Routes, Route } from 'react-router-dom';

function App() {
  return (<>
    <SnackbarProvider>
      <Header/>
      <Routes>
        <Route path="/" element={<ShopPage/>}/>
        <Route path="/search" element={<SearchPage/>}/>
        <Route path="/product/:productId" element={<Product/>}/>
        <Route path="/cart" element={<CartPage/>}/>
        <Route path="/checkout" element={<CheckoutPage/>}/>
        <Route path="/thanks_for_order" element={<ThanksForOrderPage/>}/>
      </Routes>
      <Footer/>
    </SnackbarProvider>
  </>);
}

export default App;
```

Рисунок 3.1 – Маршрутизація сайту

Джерело: розроблено автором

Зберігання стану кошика реалізовано локально у браузері користувача, що дозволяє знизити навантаження на сервер та забезпечити стабільну роботу навіть при тимчасовій відсутності підключення до мережі.

Взаємодія між клієнтською та серверною частинами веб-сайту реалізована через API (Application Programming Interface) – інтерфейс, який дозволяє програмам обмінюватися даними між собою за допомогою стандартизованих запитів [16]. У контексті веб-застосунку API виступає посередником між користувацьким інтерфейсом та серверною логікою, забезпечуючи передачу даних і виконання запитів.

Серверна частина сайту реалізована на основі FastAPI – сучасного асинхронного веб-фреймворку для мови програмування Python [17]. FastAPI обрано завдяки високій швидкості розробки та підтримці автоматичної генерації документації до запитів. Фреймворк забезпечує зручне створення інтерфейсів для обміну даними між клієнтською частиною сайту та сервером.

У рамках серверної частини реалізовані такі основні модулі:

1) модуль отримання фільтрів за категорією – відповідає за динамічне формування списку доступних фільтрів залежно від вибраної категорії товару: «усі», «букети» або «кімнатні рослини».

На сервер надсилається запит із зазначеною категорією, після чого формується відповідний запит до бази даних. Якщо передана категорія відрізняється від значення "усі", застосовується фільтрація, і повертаються лише ті фільтри, що відповідають заданій категорії (рис. 3.2);

```
@router.post("/filters")
def get_filters(filter_params: FiltersForm, db: Session = Depends(get_db)):
    query = db.query(Filter)

    if filter_params.category != 'all':
        query = query.filter(Filter.category == filter_params.category)

    filters = query.all()

    return {'filters': filters}
```

Рисунок 3.2 – Модуль отримання фільтрів за категорією

Джерело: розроблено автором

2) модуль отримання списку товарів з фільтрацією – забезпечує формування списку товарів з урахуванням вибраних користувачем параметрів пошуку, фільтрації, категорії та цінового діапазону. Він є ключовим для реалізації багатокритеріального пошуку на сайті.

Алгоритм фільтрації побудований за такими правилами:

– усі фільтри різних типів поєднуються за логікою AND – тобто товар повинен відповідати всім вибраним категоріям фільтра (наприклад, основна квітка і певний колір);

– значення в межах одного типу фільтра поєднуються за логікою OR – тобто товар може відповідати будь-якому з вибраних значень (наприклад, червоний або білий колір);

– також реалізована фільтрація за назвою (пошук), ціною (мінімум і максимум) та категорією.

Для кожного фільтра, який обрав користувач, додається окрема умова до запиту яка перевіряє, чи має товар атрибут з відповідною назвою і значенням. Таким чином, система підтримує масштабовану та гнучку логіку підбору товарів (рис. 3.3).

```

@router.post("/products/")
def list_products(search_params: ProductsList, db: Session = Depends(get_db)):
    query = db.query(Product)
    if search_params.search_query:
        query = query.filter(Product.name.ilike(f"%{search_params.search_query}%"))
    for applied_filter in search_params.filters:
        if applied_filter.values:
            query = query.filter(
                Product.attributes.any(
                    (ProductAttribute.attribute_name == applied_filter.name) &
                    ProductAttribute.attribute_value.in_(applied_filter.values)
                )
            )
    if search_params.price_max:
        query = query.filter(Product.price <= search_params.price_max)
    if search_params.price_min:
        query = query.filter(Product.price >= search_params.price_min)
    if search_params.category != 'all':
        query = query.filter(Product.category == search_params.category)
    page_size = search_params.page_size
    total = query.with_entities(func.count(Product.id)).scalar()
    query = query.order_by(Product.id)
    paged_q = (query.order_by(Product.id).offset((search_params.page - 1) * page_size).limit(page_size))
    products = paged_q.all()
    return {'products': products, 'total': total}

```

Рисунок 3.3 – Модуль отримання списку товарів з фільтрацією

Джерело: розроблено автором

3) модуль отримання детальної інформації про товар – відповідає за отримання інформації про окремий товар на основі його ідентифікатора. Під час запиту до бази даних, окрім основної інформації про товар (назва, опис, ціна, зображення тощо), також завантажуються характеристики з таблиці атрибутів. Це дозволяє на сторінці товару відобразити повну характеристику – наприклад, колір, висоту або привід.

```

@router.post("/products/{product_id}")
def get_product_detail(product_id: int, db: Session = Depends(get_db)):
    product = (
        db.query(Product)
        .options(joinedload(Product.attributes))
        .filter(Product.id == product_id)
        .first()
    )
    if not product:
        raise HTTPException(status_code=404, detail="Product not found")
    return product

```

Рисунок 3.4 – Модуль отримання детальної інформації про товар

Джерело: розроблено автором

4) модуль оформлення замовлення – отримує дані про замовлення (список товарів, адреса, номер телефону тощо) та створює запис про замовлення в базі даних разом із пов’язаними позиціями товарів.

5) рекомендаційний модуль – дозволяє формувати персоналізовані пропозиції для користувача на основі його попередніх переглядів [26]. При перегляді товарів, їх ідентифікатор зберігається у браузері користувача. Після цього сайт використовує ці дані, щоб підібрати схожі по характеристиках товари, у результаті формується добірка персональних рекомендацій.

Для реалізації рекомендаційної логіки використовується бібліотека scikit-learn – популярний інструмент у Python для машинного навчання та обробки даних [18].

Щоб мати змогу порівнювати товари між собою, їх опис разом з назвою та атрибутами перетворюється у текстове подання. Ці текстові дані

обробляються за допомогою інструмента `TfidfVectorizer` з бібліотеки `scikit-learn`, який формує числові вектори для кожного товару (рис 3.5). Метод TF-IDF (Term Frequency – Inverse Document Frequency) оцінює важливість кожного терміна в описі товару [19]. Цей метод використовує формулу: $TF\text{-}IDF(t, d) = TF(t, d) * IDF(t)$, де $TF(t, d)$ – частота терміна t у товарі d , $IDF(t)$ – зворотна частота документа, яка показує, наскільки рідко цей термін зустрічається серед усіх товарів

```
def product_to_feature_vector(product):
    attr_text = " ".join([f"{attr.attribute_name} {attr.attribute_value}" for attr in product.attributes])
    full_text = f"{product.name} {product.category} {attr_text}"
    return full_text

3 usages
def build_tfidf_matrix(products):
    texts = [product_to_feature_vector(p) for p in products]
    vectorizer = TfidfVectorizer()
    tfidf_matrix = vectorizer.fit_transform(texts)
    return tfidf_matrix, vectorizer
```

Рисунок 3.5 – Модуль для формування TF-IDF векторів для товарів

Джерело: розроблено автором

Таким чином, найбільшу вагу отримують ті слова, які часто зустрічаються в описі одного товару, але рідко в інших. Це дозволяє точно передати унікальні ознаки кожного товару у векторній формі. У результаті формується TF-IDF матриця, яка зберігається в пам'яті та використовується для пошуку схожих товарів (рис. 3.5).

```
@app.on_event("startup")
async def load_tfidf():
    products = db.query(Product).options(joinedload(Product.attributes)).all()
    tfidf_matrix, vectorizer = build_tfidf_matrix(products)
    app.state.products = products
    app.state.tfidf_matrix = tfidf_matrix
```

Рисунок 3.6 – Формування матриці TF-IDF та збереження в пам'яті

Джерело: розроблено автором

Коли користувач надсилає запит на отримання рекомендацій, система використовує збережену TF-IDF матрицю та список ідентифікаторів переглянутих товарів. На основі цих даних обчислюється схожість між товарами за змістом. Для цього використовується метод cosine similarity, реалізований у бібліотеці scikit-learn.

Косинусна схожість (cosine similarity) – це метрика, яка визначає, наскільки подібними є два вектори (товари) на основі кута між ними [20]. Формула має вигляд: $\cos(A,B)=\frac{A \cdot B}{\|A\| \cdot \|B\|}$, де A і B – TF-IDF вектори двох товарів, $A \cdot B$ – скалярний добуток, $\|A\|$ і $\|B\|$ – довжини векторів. Значення близьке до 1 означає високу схожість, а близьке до 0 – майже повну відмінність.

Для кожного товару, який не був переглянутий, обчислюється середня схожість з тими, які вже були переглянуті. Найбільш схожі товари включаються до фінального списку, в даному випадку 5 товарів (рис. 3.7).

```
def recommend_products(products, viewed_product_ids, tfidf_matrix, top_n=5):
    id_to_index = {p.id: idx for idx, p in enumerate(products)}
    viewed_indices = [id_to_index[pid] for pid in viewed_product_ids if pid in id_to_index]
    if not viewed_indices:
        return []
    similarity_matrix = cosine_similarity(tfidf_matrix)

    avg_similarities = np.mean(similarity_matrix[viewed_indices], axis=0)

    for idx in viewed_indices:
        avg_similarities[idx] = -1

    recommended_indices = np.argsort(avg_similarities)[::-1][:top_n]
    recommended_products = [products[i] for i in recommended_indices]
    return recommended_products
```

Рисунок 3.7 – Формування рекомендованих товарів на основі переглядів

Джерело: розроблено автором

Для зберігання інформації про товари, замовлення та інші сутності використано систему управління базами даних PostgreSQL. Вона була обрана завдяки своїй стабільності, підтримці складних запитів, транзакцій та широким можливостям масштабування [21].

Взаємодія між серверною частиною та базою даних реалізована через ORM (Object-relational mapping, Об'єктно-реляційне відображення) SQLAlchemy. Завдяки цьому сутності бази даних описуються у вигляді Python-класів, що спрощує роботу з даними, забезпечує читабельність коду та прискорює розробку.

Основними сутностями у структурі бази даних є:

- товари (з основними характеристиками: назва, ціна, категорія, опис, наявність, тощо),
- атрибути товарів (колір, тип, інші характер),
- замовлення,
- позиції замовлення,
- фільтри для багатокритеріального пошуку.

Ця структура забезпечує зручну організацію даних для ефективної роботи функціональних модулів сайту.

Логіка взаємодії компонентів:

- 1) користувач взаємодіє з інтерфейсом (React), надсилаючи запити до API;
- 2) серверна частина (FastAPI) обробляє запити, виконує вибірку з бази даних через SQLAlchemy і повертає дані у форматі JSON;
- 3) усі зміни даних (наприклад, створення замовлення) фіксуються у базі даних PostgreSQL.

3.2 Тестування програмного продукту

Якість веб-сайту напряму залежить від коректної роботи всіх його функцій, тому обов'язковим етапом розробки є проведення тестування. Тестування веб-проектів включає перевірку роботи основного функціоналу,

навігації, інтерфейсів, інтерактивних елементів та користувальницьких форм [22]. Це дозволяє своєчасно виявити помилки, які можуть завадити комфортному користуванню сайтом, зокрема перегляду та пошуку товарів чи оформленню замовлення.

У рамках даного проекту основну увагу було приділено функціональному виду тестування. Його метою є перевірка відповідності реалізованих функцій сайту заявленим вимогам та очікуваній поведінці [23].

В рамках функціонального тестування було перевірено такі аспекти роботи сайту:

- тестування користувацького інтерфейсу – було проведено перевірку коректності навігації між сторінками сайту, роботи кнопок, форм та полів введення. Особливу увагу приділено відображенню інформації про товари: перевірено коректність виводу назв, цін, атрибутів та зображень;

- тестування функціональності пошуку та фільтрації – особливу увагу приділено перевірці багатокритеріальних фільтрів, було протестовано застосування фільтрів за категорією, кольором, приводом та ціновим діапазоном. Перевірено, що результати пошуку коректно відображаються відповідно до заданих критеріїв, а сам процес фільтрації відбувається швидко та без помилок;

- тестування кошика та оформлення замовлення – було протестовано процес додавання товарів до кошика, зміни їх кількості, видалення товарів та остаточне оформлення замовлення через форму. Перевірено коректність формування номера замовлення, а також передачу контактних даних користувача на сервер для збереження у базі даних;

- тестування рекомендаційного блоку – окремо перевірено коректну роботу блоку з персональними рекомендаціями. Тестування включало перевірку відображення товарів, пов'язаних з попередніми переглядами користувача, та актуальність рекомендованих позицій.

3.3 Використання програмного продукту

Для початку роботи користувачу необхідно відкрити сайт інтернет-магазину квітів за відповідним посиланням. Після завантаження сайту відображається головна сторінка (рис. 3.8), де представлено контактні дані, панель пошуку, кошик та каталог товарів.

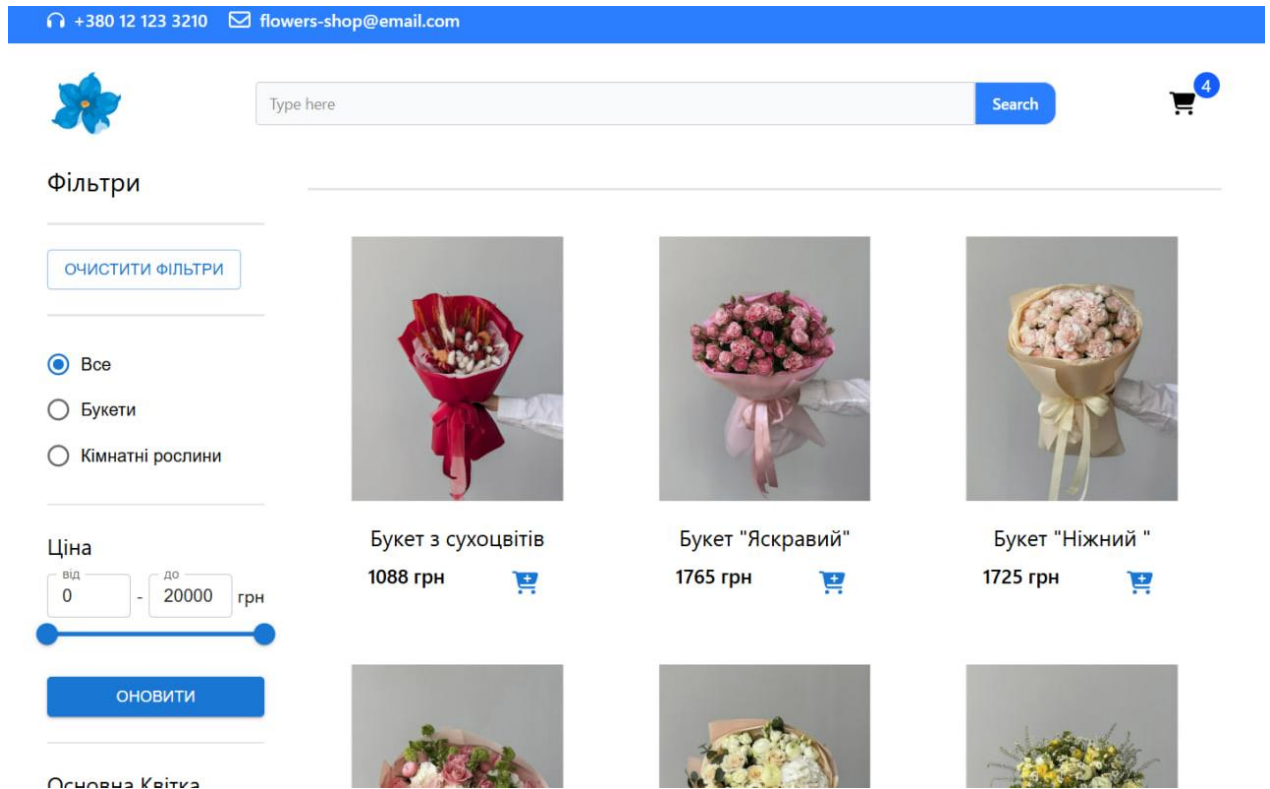


Рисунок 3.8 – Головна сторінка сайту

Джерело: розроблено автором

На головній сторінці користувач може обрати одну з категорій товарів (букети або кімнатні рослини), скористатися пошуком, а також перейти до кошика. При виборі категорії оновлюється панель фільтрів, яка відображає тільки ті характеристики, що актуальні для вибраного типу товарів (рис. 3.9), що дає змогу уточнити пошук. Користувач може вказати бажану категорію, діапазон цін та додаткові фільтри – наприклад, тип поливу чи висоту рослини (для кімнатних рослин).

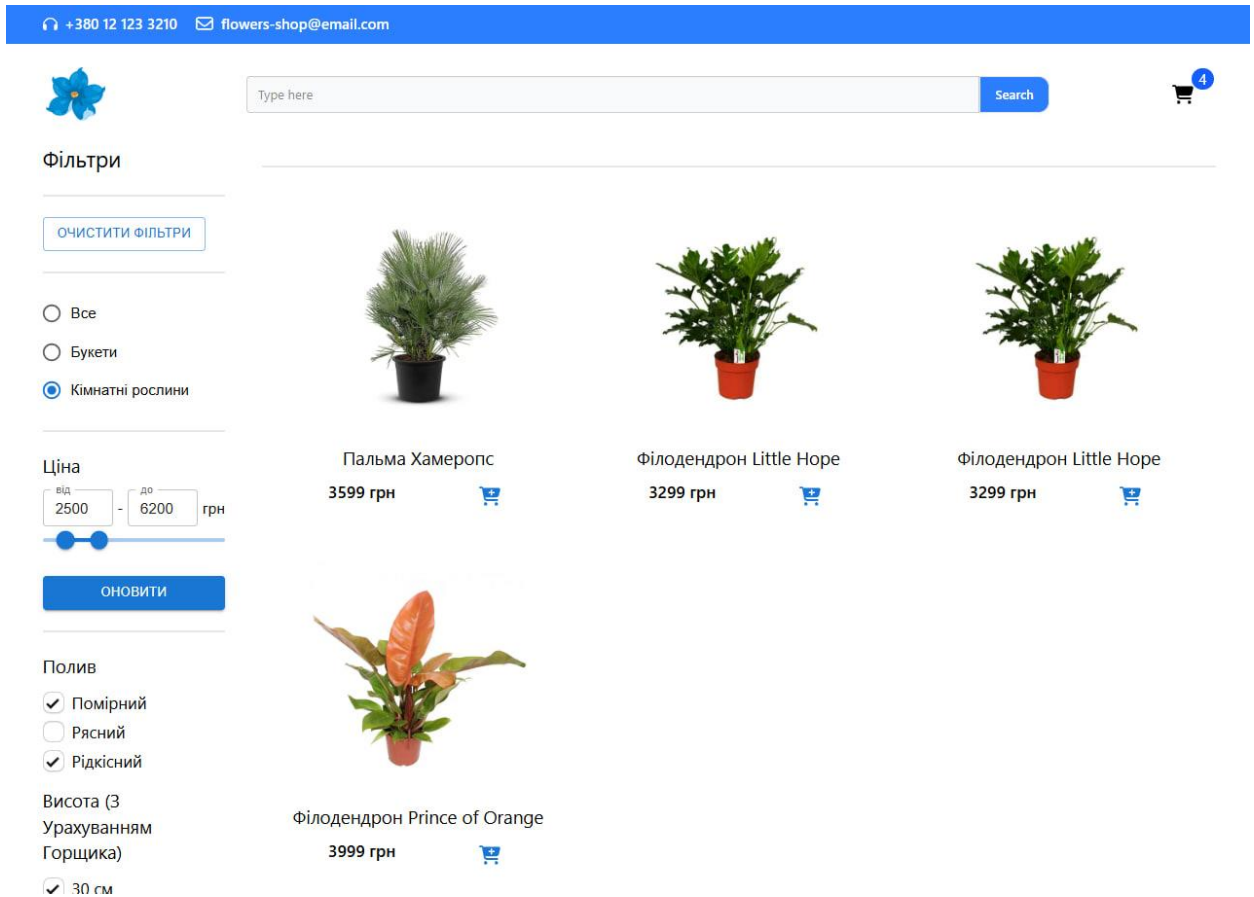


Рисунок 3.9 – Фільтрація товарів у категорії Кімнатні рослини

Джерело: розроблено автором

Товари на сторінці відображаються у вигляді карток із зображенням, назвою, ціною та кнопкою додавання до кошика. При натисканні на товар відкривається сторінка детального перегляду (рис. 3.5), де подано повний опис, таблицю характеристик, ціну, можливість вибору кількості товару та кнопки «Додати до кошика» або «Купити зараз».

Також відображається блок персональних рекомендацій – товари, які можуть зацікавити користувача на основі його попередніх переглядів.

При натисканні на іконку кошика користувач переходить на сторінку оформлення замовлення (рис. 3.11), де зібрано обрані товари. Тут можна змінити кількість товарів, видалити позиції та переглянути загальну суму замовлення.

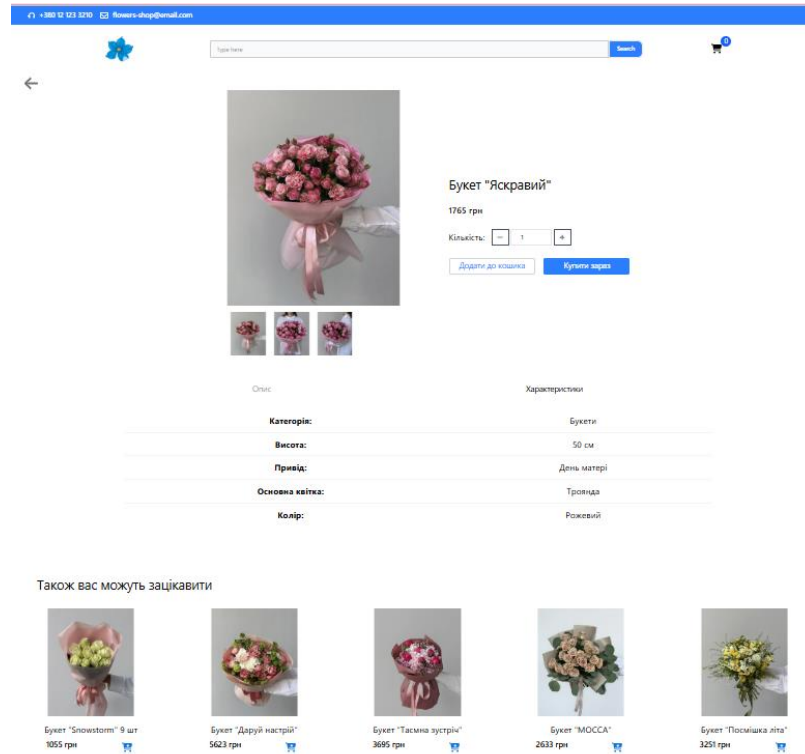


Рисунок 3.10 – Сторінка товару з детальними характеристиками

Джерело: розроблено автором

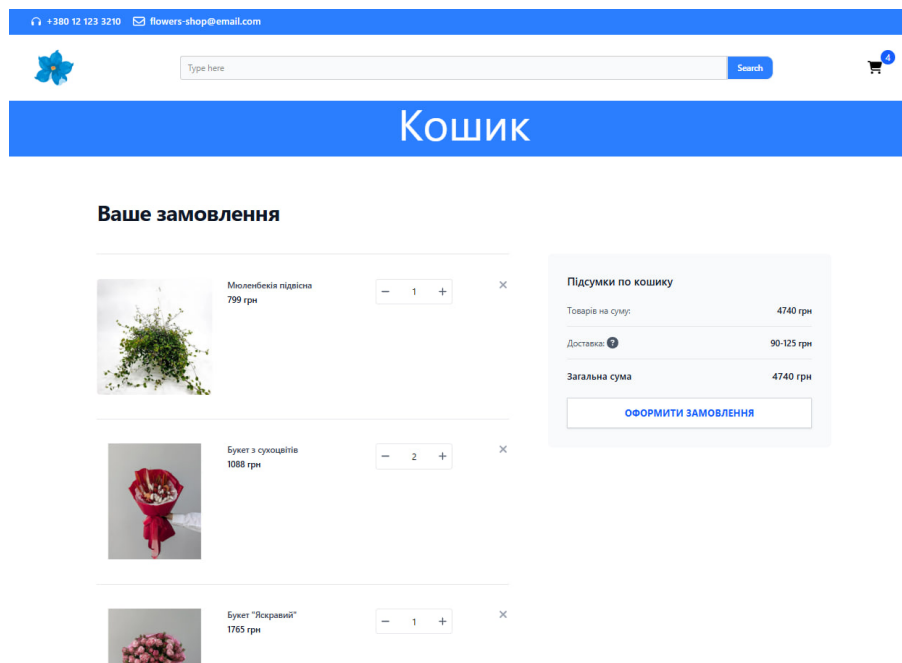


Рисунок 3.11 – Сторінка кошика з переліком обраних товарів

Джерело: розроблено автором

Натиснувши кнопку «Оформити замовлення», користувач потрапляє на сторінку оформлення замовлення (рис. 3.12). Тут потрібно вказати контактні дані, обрати тип доставки (відділення або кур'єр), спосіб оплати (передоплата чи при отриманні), за потреби залишити коментар до замовлення.

Процес замовлення спрощений і не вимагає реєстрації – після натискання кнопки «Оформити» користувач отримує підтвердження із номером замовлення та повідомлення про те, що з ним зв'яжеться менеджер для уточнення деталей (рис. 3.13).

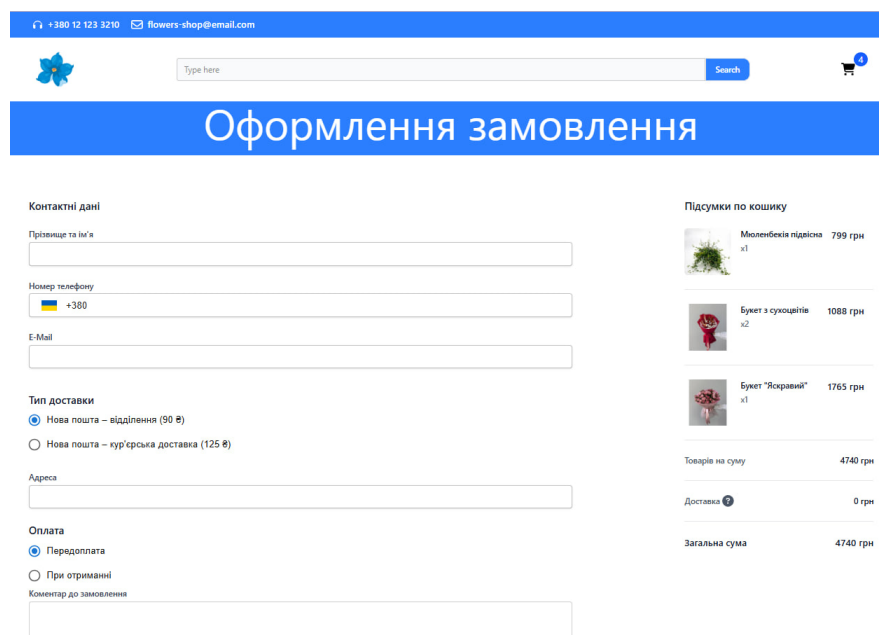


Рисунок 3.12 – Сторінка оформлення замовлення

Джерело: розроблено автором

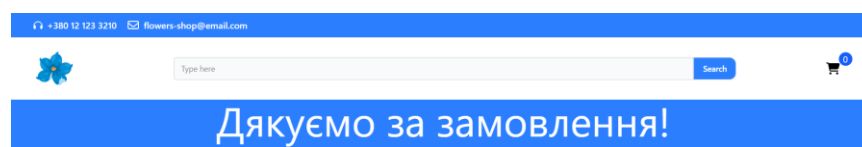


Рисунок 3.13 – Повідомлення про успішне оформлення замовлення

Джерело: розроблено автором

Висновки по розділу 3

У ході реалізації програмного забезпечення було проведено технологічний аналіз створення веб-сайту інтернет-магазину квітів відповідно до сучасних підходів у веб-розробці. Визначено та обґрунтовано вибір засобів реалізації. Для клієнтської частини використано бібліотеку React, що дозволяє будувати компонентну архітектуру та реалізувати зручний інтерфейс користувача. Для серверної частини обрано FastAPI – фреймворк на мові Python, який забезпечує швидку розробку та зручну роботу з API. Збереження даних реалізовано з використанням реляційної бази PostgreSQL, а для взаємодії з базою даних застосовано ORM-бібліотеку SQLAlchemy, що спростило обробку запитів і структурування даних у вигляді класів.

Реалізовано ключові програмні модулі, зокрема: формування списку товарів із фільтрацією, перегляд детальної інформації, оформлення замовлень та побудова персональних рекомендацій на основі алгоритмів TF-IDF та Cosine Similarity.

Всі модулі протестовано в рамках функціонального тестування – перевірено коректність навігації, відображення товарів, роботи фільтрів, кошика та рекомендацій. Описано особливості використання сайту з точки зору кінцевого користувача,

В результаті реалізації було створено повноцінну програмну систему – веб-сайт інтернет-магазину квітів, який відповідає поставленим вимогам і є готовим до практичного використання.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було створено вебсайт інтернет-магазину з продажу квітів і кімнатних рослин, що поєднує в собі сучасні технології веб-розробки, багатокритеріальні фільтри і персоналізовану систему рекомендацій.

У першому розділі було досліджено предметну область, виявлено актуальні проблеми у сфері онлайн-продажу квітів, проаналізовано аналогічні сервіси-конкуренти. Визначено їхні недоліки, такі як обмежена фільтрація, відсутність рекомендацій та незручний інтерфейс. Це дозволило сформулювати вимоги до майбутнього рішення та поставити завдання на розробку.

У другому розділі виконано проектування веб-застосунку. Змодельовано сценарії взаємодії користувача з системою (діаграма прецедентів, діаграма діяльності), структуру даних (діаграми класів і об'єктів), а також архітектуру клієнт-серверної взаємодії. Це дозволило створити чітке технічне підґрунтя для реалізації функціональних модулів.

У третьому розділі описано процес реалізації програмного продукту. Реалізовано клієнтську частину за допомогою React та серверну частину на основі FastAPI. Використано PostgreSQL для збереження даних і SQLAlchemy як ORM. Створено модулі для фільтрації товарів, перегляду деталей, оформлення замовлень і формування рекомендацій. Проведено функціональне тестування системи та описано сценарії її використання кінцевим користувачем.

Розроблений веб-сайт відповідає сформульованим на початку роботи вимогам. Він забезпечує зручний інтерфейс для вибору квіткової продукції, дозволяє швидко знаходити потрібні товари за допомогою фільтрів, переглядати детальну інформацію, отримувати персоналізовані рекомендації та оформлювати замовлення без потреби реєстрації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Аналіз ринку квітів в Україні: які квіти купують частіше і на які свята – URL <https://pro-consulting.ua/ua/pressroom/analiz-rynka-cvetov-v-ukraine-kakie-cvety-pokupayut-chashe-i-na-kakie-prazdniki> (дата звернення 25.03.2025)
2. Уподобання користувачів у інтернет-маркетингу.– URL: https://www.researchgate.net/publication/332403531_Customer_Preferences_towards_Digital_Marketing_Strategies(дата звернення 25.03.2025)
3. Глобальний ринок кімнатних рослин –тенденції в галузі та прогноз до 2031 року – URL: <https://www.databridgemarketresearch.com/reports/global-indoor-plants-market> (дата звернення 25.03.2025)
4. Інтернет-магазин Flowers.ua –доставка квітів по Україні та світу – URL: <https://flowers.ua/> (дата звернення 25.03.2025)
5. Інтернет-магазин Укрфлора – букети, рослини і квіти в Києві – URL: <https://ukraflora.ua> (дата звернення 25.03.2025)
6. Інтернет-магазин Proflowers.ua – доставка квітів Київ – URL: <https://proflowers.ua/ua> (дата звернення 25.03.2025)
7. Опис діаграм UML та діаграма прецедентів – стаття Evergreens – URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html> (дата звернення 25.03.2025)
8. Діаграма діяльності в UML: символи, компоненти та приклад – Guru99 – URL: <https://www.guru99.com/uk/uml-activity-diagram.html> (дата звернення 25.03.2025)
9. Підручник з діаграми класів UML: абстрактний клас із прикладами – URL: <https://www.guru99.com/uk/uml-class-diagram.html> (дата звернення 25.03.2025)
10. Офіційна документація SQLAlchemy – URL: <https://docs.sqlalchemy.org/en/20/> (дата звернення 25.03.2025)

11. Підручник з діаграми класів UML: Що таке діаграма об'єктів в UML – URL: <https://www.guru99.com/uk/uml-object-diagram.html> (дата звернення 25.03.2025)
12. Стаття про клієнт-серверну архітектуру – навчальний блог QATestLab – URL: <https://training.qatestlab.com/blog/technical-articles/client-server-architecture/> (дата звернення 25.03.2025)
13. Архітектура бази даних у моделі клієнт-сервер – стаття InformatеcDigital – URL: <https://informatecdigital.com/uk/%D0%90%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0%D0%B1%D0%B0%D0%B7%D0%B8%D0%B4%D0%B0%D0%BD%D0%B8%D1%85%D0%BA%D0%BB%D1%96%D1%94%D0%BD%D1%82%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BF%D0%BE%D0%B2%D0%BD%D0%B8%D0%B9%D0%BF%D0%BE%D1%81%D1%96%D0%B1%D0%BD%D0%B8%D0%BA/> (дата звернення 25.03.2025)
14. Офіційна документація React українською – URL: <https://uk.react.dev/learn> (дата звернення 25.03.2025)
15. Підручник з React Router українською – W3Schools – URL: https://w3schoolsua.github.io/react/react_router.html (дата звернення 25.03.2025)
16. Що таке API: пояснення та приклади – GoIT – URL: <https://goit.global/ua/articles/application-programming-interface-api-shcho-tse-iak-i-de-pratsiuiu/> (дата звернення 25.03.2025)
17. Офіційна документація FastAPI українською – URL: <https://fastapi.tiangolo.com/uk/> (дата звернення 25.03.2025)
18. Офіційна документація бібліотеки scikit-learn – URL: <https://scikit-learn.org/stable/index.html> (дата звернення 25.03.2025)

19. Пояснення принципу TF-IDF у Natural Language Processing – Medium – URL: <https://medium.com/@abhishekjainindore24/tf-idf-in-nlp-term-frequency-inverse-document-frequency-e05b65932f1d> (дата звернення 25.03.2025)
20. Cosine Similarity – принцип роботи та приклад реалізації – GeeksforGeeks – URL: <https://www.geeksforgeeks.org/cosine-similarity/> (дата звернення 25.03.2025)
21. Вступ до PostgreSQL українською – W3Schools – URL: https://w3schoolsua.github.io/postgresql/postgresql_intro.html (дата звернення 25.03.2025)
22. Тестування веб-проектів: основні етапи та поради – QALight – URL: <https://qalight.ua/baza-znaniy/testuvannya-veb-proektiv-osnovni-etapi-ta-poradi/> (дата звернення 25.03.2025)
23. Функціональне тестування – QALight – URL: <https://qalight.ua/baza-znaniy/funktsionalne-testuvannya/> (дата звернення 25.03.2025)
24. Розробка програмного забезпечення з використанням баз даних: навчальний посібник / Ю. В. Шамарін, С. М. Мічківський, К. В. Смоктій, Д. В. Шевцов. – Донецьк: ДонНУ, 2013. – 201 с.
25. Мічківський С. Microsoft Office (Word, Excel, Outlook ...) : навч. посіб. / С. Мічківський, Д. Балдик, В. Головань; Східноукр. нац. ун-т ім. В. Даля, Аграр. ф-т. – Київ : [Вид-во Східноукр. нац. ун-т ім. В. Даля], 2023. – 128 с. – URL: <https://dspace.snu.edu.ua/handle/123456789/1723>
26. Шаповалов С.М., Мічківський С.М. Розробка рекомендаційної системи для підбору співрозмовника в соціальній мережі // XII Міжнародна науково-технічна конференція «Проблеми Інформатизації» (м. Київ, 13 грудня 2018 року). – К.: Державний університет телекомунікацій, 2018. URL: https://dut.edu.ua/uploads/1_1701_65845854.pdf (дата звернення: 20.12.2023)