

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка та впровадження UI/UX дизайну для інтернет-магазину
«BLOOM»

Виконала: студентка IV курсу, групи СН-42

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Петрик О.Ю.

(прізвище та ініціали)

Керівник

(підпис)

Палка О.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Станько А.А.

(прізвище та ініціали)

Тернопіль
2025

«26» червня 2025 р.

Кваліфікаційна робота на тему: Розробка та впровадження UI/UX дизайну для інтернет-магазину «BLOOM», Мета і завдання роботи, Актуальність теми роботи, Об'єкт та предмет Дослідження, Практичне застосування ожержжаних результатів, Аналіз конкурентів інтернет-Магазинів, Поняття UI/UX, Обрано засоби розробки, Обрано хостинг, Розроблено структуру, Розроблено базу даних, Виділені кардинальності, Реалізація сайту, Розробка та тестування Інтерфейсу та адмін-панелі. Розробка бекенд частини, Висновки, Дякую за увагу

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин Віктор Степанович, кандидат техн. Наук кафедри МТ	08.05.2025	05.06.2025

7. Дата видачі завдання 29 жовтня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент

(підпис)

Петрик О.Ю.

(прізвище та ініціали)

Керівник роботи

(підпис)

Палка О.В.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка та впровадження UI/UX дизайну для інтернет-магазину «BLOOM»//
Кваліфікаційна робота освітнього рівня «Бакалавр» Петрик Олеся Юріївна //
Тернопільський національний технічний університет імені Івана Пулюя,
факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра
комп'ютерних наук, група СН-42 // Тернопіль, 2025 // С. 87, рис. – 67, табл. – 9,
додат. – 7, бібліогр. – 74.

Ключові слова: UI/UX, дизайн, інтерфейс користувача, користувацький досвід, інтернет-магазин, розробка, адаптивність, база даних.

Кваліфікаційна робота присвячена дослідженню процесів розробки та впровадження UI/UX дизайну для інтернет-магазину квітів «Bloom». У першому розділі кваліфікаційної роботи описано аналіз предметної області, особливості сучасних інтернет-магазинів, а також вимоги до створення інтерфейсу та вибору інструментів. Висвітлено актуальність теми та переваги застосування якісного UI/UX. Розглянуто аналіз конкурентів та засоби проєктування дизайну.

У другому розділі кваліфікаційної роботи змодельовано архітектуру магазину, описано ролі користувачів, варіанти використання та структуру бази даних. Досліджено трирівневу клієнт-серверну модель. Подано діаграми взаємодії та схеми структури таблиць.

У третьому розділі кваліфікаційної роботи описано реалізацію клієнтської та серверної частин, застосування HTML, CSS, JavaScript та PHP. Проаналізовано процес тестування, адаптацію під різні пристрої та роботу з базою даних.

Об'єкт дослідження: вебсайт для інтернет-магазину.

Предмет дослідження: інтерфейс користувача та користувацький досвід в інтернет-магазині.

ANNOTATION

Development and Implementation of UI/UX Design for the «Bloom» Online Store // Qualification work of the educational level «Bachelor» // Petryk Olesia // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-42 // Ternopil, 2025 // P. 87, fig. – 67, tabl. – 9, annexes. – 7, references – 74.

Keywords: UI/UX, design, user interface, user experience, online store, development, responsiveness, database.

This qualification work is dedicated to researching the processes of development and implementation of UI/UX design for the flower online store "Bloom".

The first chapter presents an analysis of the subject area, features of modern online stores, as well as the requirements for interface creation and tool selection. The relevance of the topic and the benefits of applying high-quality UI/UX design are highlighted. Competitor analysis and design prototyping tools are considered.

The second chapter models the architecture of the store, describes user roles, use cases, and the database structure. The three-tier client-server model is examined. Interaction diagrams and table structure schemes are provided.

The third chapter describes the implementation of the client-side and server-side components using HTML, CSS, JavaScript, and PHP. The processes of testing, responsiveness across various devices, and database interaction are analyzed.

Object of research: the website of an online store.

Subject of research: user interface and user experience in an online store.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ТЕРМІНІВ

AJAX (англ. *Asynchronous JavaScript and XML*) – технологія для асинхронної взаємодії з сервером без перезавантаження сторінки.

API (англ. *Application Programming Interface*) – інтерфейс прикладного програмування, що дозволяє програмам взаємодіяти між собою.

CSS (англ. *Cascading Style Sheets*) – каскадні таблиці стилів.

MySQL – вільна система керування реляційними базами даних.

HTML (англ. *HyperText Markup Language*) – мова розмітки гіпертексту.

JS (англ. *JavaScript*) – мова програмування для реалізації інтерактивності на сайтах.

PHP (англ. *Hypertext Preprocessor*) – серверна мова програмування, яка обробляє запити і генерує динамічний контент на сайті.

UI (англ. *User Interface*) – інтерфейс користувача, візуальна частина продукту.

UX (англ. *User Experience*) – досвід користувача, враження від продукту.

VS Code (англ. *Visual Studio Code*) – редактор коду.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	9
1.1 Аналіз предметної області інтернет-магазину «BLOOM»	9
1.2 Аналіз конкурентів інтернет-магазинів квітів	10
1.3 Формулювання вимог до інтернет-магазину «BLOOM»	12
1.4 Вибір засобів розробки для інтернет-магазину «BLOOM»	13
1.4.1 Визначення UI/UX	13
1.4.2 Figma.....	18
1.4.3 Порівняння програм для розробки дизайну	19
1.5 Програмні засоби для розробки.....	21
1.6 Розміщення сайту на хостингу	22
1.6.1 Netlify.....	22
1.6.2 AWARDSPACE	23
1.7 Постановка задач для інтернет-магазину «BLOOM».....	24
1.8 Висновок до першого розділу.....	25
РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ «BLOOM»	26
2.1 Моделювання архітектури та структури інтернет-магазину «BLOOM»	26
2.2 Моделювання архітектури інтернет-магазину квітів	26
2.3 Варіанти використання інтернет-магазину квітів.....	28
2.4 Розробка структури таблиць для інтернет-магазину «BLOOM»	31
2.5 Висновки до другого розділу	36
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ «BLOOM»	37
3.1 Розробка інтерфейсу та серверної частини для інтернет-магазину «BLOOM».....	37
3.1.1 Розробка frontend-частини для інтернет-магазину квітів	37
3.1.2 Розробка бекенд -частини для інтернет-магазину квітів	49

3.2 Тестування інтерфейсу сайту інтернет-магазину «BLOOM».....	54
3.2.1 Тестування адаптивності вебсайту інтернет-магазину	55
3.2.2 Кросплатформність інтерфейсу інтернет-магазину квітів	64
3.3 Висновки до третього розділу.....	70
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	71
4.1 Вплив діяльності людини на довкілля	71
4.2 Естетичне оформлення та ергономічне дослідження робочого місця оператора.....	72
4.3 Вплив кольору на покращення умов праці та підвищення продуктивності виробництва	74
4.4 Висновок до четвертого розділу	76
ВИСНОВКИ.....	77
ПЕРЕЛІК ДЖЕРЕЛ	79
ДОДАТКИ	

ВСТУП

Актуальність теми. Внаслідок глобалізації та цифрової торгівлі інтернет-магазини набули широкого поширення, а якість взаємодії користувача з інтерфейсом стала одним із головних чинників при виборі платформи. Тому розробка ефективного UI/UX дизайну є актуальним напрямком сучасних досліджень у галузі комп'ютерних наук та вебтехнологій.

Мета і задачі дослідження. Метою кваліфікаційної роботи освітнього рівня «Бакалавр» є підвищення якості надання послуг шляхом створення зручного, функціонального та привабливого інтерфейсу для інтернет-магазину «Bloom» на основі принципів UI/UX дизайну.

Для досягнення поставленої мети передбачено виконання таких завдань:

- проаналізувати стан наукових досліджень у сфері UI/UX дизайну вебзастосунків;
- дослідити інтерфейси існуючих інтернет-магазинів та виділити їх сильні та слабкі сторони;
- розробити структуру та архітектуру інтернет-магазину;
- спроектувати базу даних із підтримкою замовлень та облікових записів користувачів;
- реалізувати клієнтську та серверну частини вебзастосунку із використанням сучасних технологій;
- протестувати функціональність і адаптивність створеного ресурсу.

Практичне значення одержаних результатів. Результати даного дослідження можуть бути використані для впровадження повнофункціонального інтернет-магазину «Bloom» як комерційного продукту або шаблону для розробки аналогічних вебзастосунків. Реалізація проєкту демонструє можливість створення сучасного, адаптивного та зручного інтерфейсу без залучення складних фреймворків, що робить рішення доступним для малого бізнесу та індивідуальних розробників.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Аналіз предметної області інтернет-магазину «BLOOM»

Завданням кваліфікаційної роботи є розробка дизайну інтернет-магазину квітів «BLOOM».

Інтернет-магазини квітів стають популярнішими та створюють високу конкуренцію на ринку завдяки зручності замовлення [1]. Тому інтернет-покупці все частіше використовують нові технології, щоб приймати рішення про покупки [2].

Серед таких магазинів варто виділити «Million flowers», який пропонує стильні букети для різних подій із можливістю доставки, та «Sparkle Ternopil», що вирізняється індивідуальним підходом до клієнтів і оригінальними квітковими композиціями.

Інтернет-магазин має деякі переваги як для покупців, так і для продавців – зручність і доступність, так як робота магазину в інтернеті є цілодобовою, що дозволяє робити замовлення в будь який час та не виходячи з дому, що економить час [3].

Розробка інтернет-магазину передбачає впровадження сучасних технологій, які забезпечують високий рівень користувацького досвіду, безпеку та надійність системи. Це сприятиме підвищенню конкурентоспроможності на ринку та відповідатиме вимогам сучасних споживачів, які частіше обирають інтернет-шопінг як основний спосіб для покупки товарів [4].

Якщо говорити про переваги для продавців, то це залучення нових клієнтів через інтернет-магазин, що знижує витрати утримання звичайних магазинів. Це дозволяє їм значно розширити географію продажів, охопивши аудиторію далеко за межами свого фізичного розташування.

1.2 Аналіз конкурентів інтернет-магазинів квітів

Аналіз конкурентів інтернет-магазинів потрібний для дослідження теперішніх трендів та тенденцій ринку, а також сильних і слабких сторін інтерфейсу.

«Sparkle Ternopil» – один із найпопулярніших квіткових крамниць Тернополя. Сайт магазину розроблений в ніжних пастельних кольорах, з простим інтерфейсом, який не перевантажує сторінку і є легким у використанні для користувача. Є навігаційна палень, яка склалася з категорій «франшиза», «курси флористики», «майстер класи» та «про нас», також зліва є логотип та так зване випадаюче меню «каталог», де можна обрати категорію для замовлення [5].

На рисунку 1.1 зображений інтернет-магазин «Sparkle Ternopil».

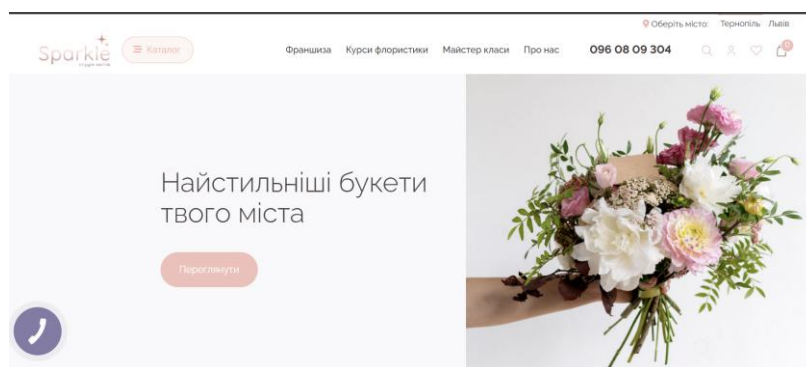


Рисунок 1.1 – Головна сторінка «Sparkle Ternopil» [5]

Переваги сайту магазину «Sparkle Ternopil»:

- Категорія контактів: Можливість швидко зв'язатися з магазином за допомогою телефону прямо із сайту.
- Інструмент пошуку: Зручна іконка «Пошук товарів» для швидкого знаходження потрібної продукції.
- Зрозуміла структура сайту: Зрозуміла панель меню, розташоване зверху, яке дозволяє легко переглядати категорії товарів.

– Мобільна версія: Сайт адаптований для зручного користування на смартфонах та планшетах.

Магазин «Million flowers» пропонує широкий асортимент квітів за цінами нижче ринкових. Даний сайт має реалізовано із дещо простішим інтерфейсом, що зображений на рисунку 1.2 [6].

Сама навігаційна панель складається із логотипу, місцезнаходження, корзини та випадаючого меню.

Крім того, в сайту магазину є такі недоліки:

- На самій панелі немає вкладки пошуку, що ускладнює користувачеві легко знайти товар.
- Погана адаптація сайту є незручно для використання на смартфонах або планшетах, що може відштовхувати сучасних користувачів.
- Відсутність вкладки пошуку, що ускладнює клієнтам швидко знаходження потрібного товару.

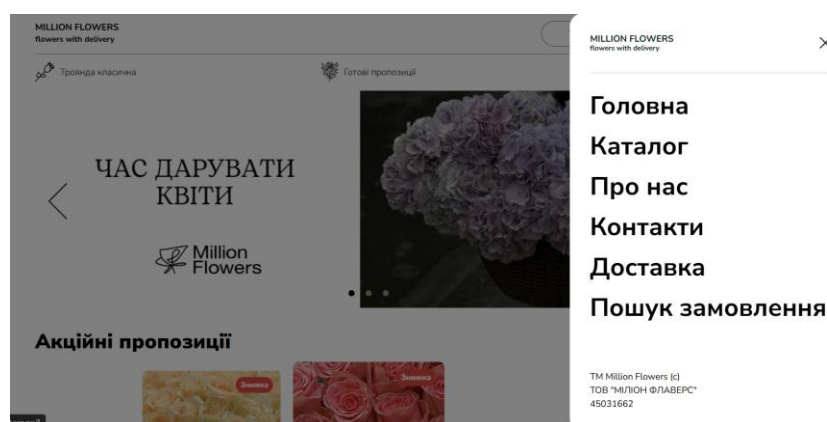


Рисунок 1.2 – Головна сторінка «Million flowers» [6]

В цьому випадку інтерфейс сайту для великих екранів не є дуже зручний. Для того, щоб переходити між сторінками, треба додатково відкривати меню, яке є, до того ж, великих розмірів і білого кольору – що дуже «ріже» очі.

Також недоліком сайту є перевантаженість зображеннями та великий розмір тексту, що робить сторінку більш громіздкою та ускладнює інтерфейс.

«Flora de luxe» пропонує великий вибір квітів з своєю базою в телеграмі. Завантажений логотип з основним елементом та якісним контентом, проте без свого сайту, як і ще один конкурент [7].

«FloArt» – магазин квітів з багатьма акційними пропозиціями, готовими букетами а також різними подарунковими наборами. Орієнтується на соц. мережі, тому немає сайту. Гарний основний елемент але перевантажений логотип [8].

1.3 Формулювання вимог до інтернет-магазину «BLOOM»

В ході кваліфікаційної роботи будуть такі завдання – розробка бренду інтернет-магазину та сайту, який повинен мати зручний та простий інтерфейс для покупців, з можливістю швидкого пошуку та іншими функціями, такими як додавання обраних товарів в кошик та перегляд наявного каталогу товарів.

Постановка завдань має базуватися виключно на чітко сформульованих вимогах, які визначають сутність кінцевого продукту. Кожне завдання повинно бути максимально конкретним та деталізованим [9].

Розробка дизайну для інтернет-магазину: створення детального візуального дизайну інтерфейсу користувача (UI) в Figma з урахуванням принципів UX; візуальна адаптація для різних пристроїв; проектування адаптивного дизайну для забезпечення коректного відображення та зручності використання сайту на різних пристроях.

Технічна реалізація: верстка сторінок інтернет-магазину за допомогою HTML та CSS; реалізація інтерактивних функцій за допомогою JavaScript; розробка серверної частини (бекенду) для обробки запитів та реалізації бізнес-логіки інтернет-магазину; проектування структури та підключення бази даних для зберігання інформації про товари, користувачів, замовлення тощо.

Тестування: проведення комплексного тестування функціональності сайту (робота каталогу, пошуку, кошика, форми замовлення); тестування адаптивності та коректності відображення на різних браузерях і пристроях [10].

Розміщення на хостингу: вибір та підготовка хостингу для розміщення вебсайту; завантаження файлів сайту на сервер та налаштування доменного імені.

Таким чином, робота спрямована на створення повноцінного та зручного інтернет-магазину з урахуванням всіх ключових аспектів розробки [11].

1.4 Вибір засобів розробки для інтернет-магазину «BLOOM»

1.4.1 Визначення UI/UX

UI (інтерфейс користувача) і UX (взаємодія з користувачем) є двома важливими аспектами дизайну цифрових продуктів, особливо програмного забезпечення, програм і вебсайтів [12].

Рисунок 1.3 ілюструє актуальність досліджуваної тематики в різних наукових галузях у вигляді діаграми розподілу публікацій за сферами знань відповідно до даних бази Scopus [13].

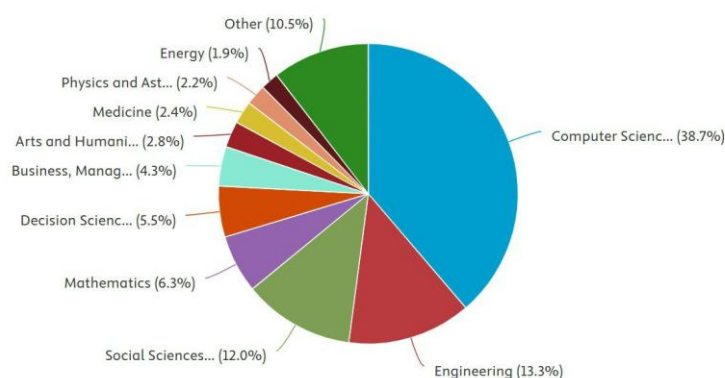


Рисунок 1.3 – Актуальність UI та UX у різних сферах [13]

З діаграми можна побачити, що найбільша частка публікацій припадає на комп'ютерні науки (38,7%), що свідчить про високу актуальність теми в цій сфері.

На рисунку 1.4 подана діаграма із статистикою актуальності UI/UX на базі Scopus [14].

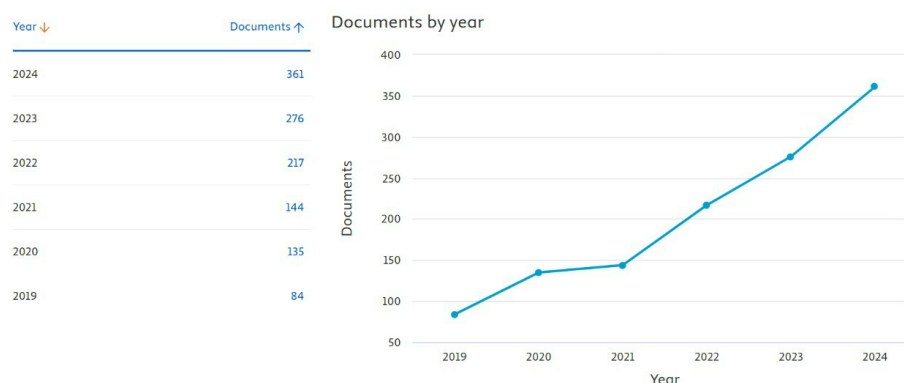


Рисунок 1.4 – Діаграма динаміки UI/UX [14]

Аналіз динаміки кількості документів за роками свідчить про стійку тенденцію до зростання. З 2019 року спостерігається поступове збільшення. У 2024 році кількість статей досягла 361, що є найвищим показником за весь період спостережень. Така тенденція вказує на зростаючу актуальність цієї сфери.

UI – це візуалізація частина цифрового продукту, який використовується користувачами для взаємодії з системою, отриманої після UX [12]. Він орієнтований на візуальний вигляд і функціональність продукту [15]. Можна сказати, що якщо UX-дизайн – це «тіло» сайту, то UI – це його «одяг» [16]. Не можна з упевненістю сказати, що UI - це лише графічне зображення точок, отриманих при створенні різних UX-моделей. для досягнення максимального комфорту взаємодії користувача з сайтом.

Значення UI набагато важливіше, ніж здається на перший погляд [17].

Хоча є певна схожість між UI та UX, та є основні відмінності, які слід враховувати:

- Зовнішній вигляд і відчуття: дизайн інтерфейсу користувача включає візуальні компоненти та інтерактивні елементи, які сприяють зручній взаємодії з користувачем. Дизайн UX зосереджується на створенні значущого та

актуального досвіду через загальне відчуття продукту чи послуги та його ключових компонентів.

– Дизайн та прототипування: дизайнери UX та UI працюють над одним продуктом, але дизайнери UX створюють каркаси та прототипи, щоб налагодити потік користувачів. Навпаки, UI дизайнери розробляють привабливий, візуально привабливий дизайн продуктів [18].

На рисунку 1.5 наведено взаємозв'язок між UX та UI, який демонструє основні аспекти кожного напрямку та їх спільні елементи, такі як співпраця, прототипування та дизайн мислення.

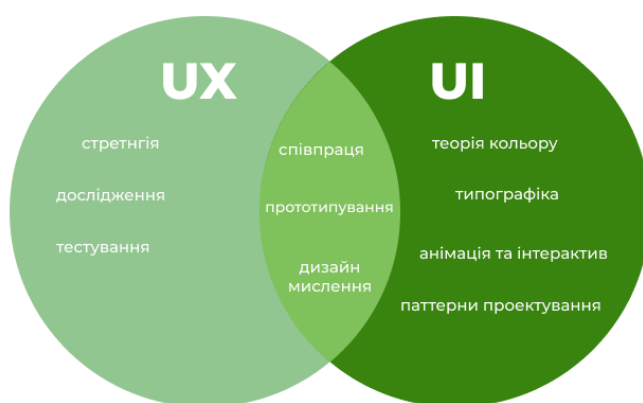


Рисунок 1.5 – Взаємозв'язок між UX та UI

UX зосереджується на загальному шляху користувача, охоплюючи все: від перших вражень до простоти використання продукту [19].

Він зосереджений на створенні бездоганного та змістовного досвіду користувача. Це все про дослідження, вирішення проблем і розуміння потреб користувачів, щоб забезпечити функціональність і задоволення [20].

Основна мета UX – забезпечити позитивний, ефективний і задовільний досвід для користувачів [21].

Створення дизайну за принципами UI/UX базується на принципах зручності використання, а саме: зрозумілість інтерфейсу (легкість першого знайомства), ефективність, можливість запам'ятовування та задоволення від проекту [12].

У таблиці 1.1 подано основні принципи зручності інтерфейсу.

Таблиця 1.1 – Основні принципи зручності інтерфейсу

Назва	Опис
Зрозумілість інтерфейсу	Система має бути простою для освоєння, щоб користувачі могли швидко працювати.
Ефективність	Швидкість та точність виконання завдань користувачами.
Запам'ятовуваність	Система повинна бути легкою для запам'ятовування, щоб навіть нерегулярні користувачі могли швидко відновити навички роботи без додаткового навчання.
Задоволення	Рівень комфорту та задоволення від використання проекту.

UI, з іншого боку, має справу з візуальними та динамічними частинами продукту, з якими безпосередньо взаємодіють відвідувачі вебсайту. Він стосується екранів, кнопок, перемикачів, значків та інших візуальних елементів, з якими ви взаємодієте під час використання вебсайту, програми чи іншого електронного пристрою, а також зосереджується на візуальних інтерактивних елементах продукту для створення естетично привабливих інтерфейсів [22].

Важливість користувацького інтерфейсу також виділяється у праці Палки Олега «Comparative Analysis of Smart City Platforms», де було проведено дослідження платформ таких як OneM2M, FIWARE та Cisco Kinetic for Cities. У висновку виявлено, що окрім схожої функціональності, характеристика UI важлива та повинна бути протестована на етапі реалізації [23].

Інтерфейс користувача зосереджується на зовнішньому вигляді, а UX – на фактичному досвіді користувача. Продукт може виглядати чудово, але без

тестування користувачами він може не задовольнити реальні потреби користувачів. На рисунку 1.6 подано приклад порівняння [24].

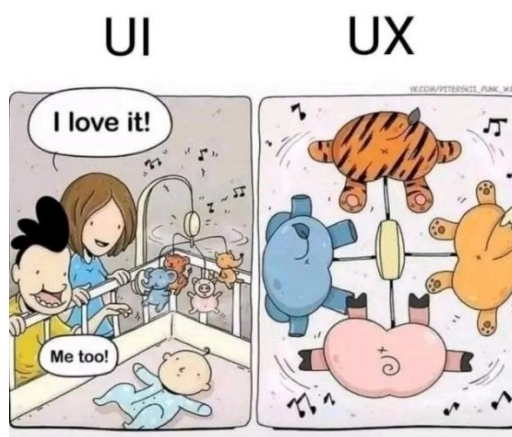


Рисунок 1.6 – Порівняння UX та UI [24]

Основні етапи, які вони виконують:

- Проведення дослідження користувачів для визначення будь-яких цілей, потреб, поведінки та проблемних моментів, пов'язаних із взаємодією продукту.
- Розробка персонажів користувача на основі цільових клієнтів.
- Створення «карт подорожі користувача» для аналізу взаємодії клієнта з продуктом.
- Створення каркасів і прототипів, щоб відточити, як виглядатиме кінцевий продукт.
- Виконання користувацького тестування для перевірки проектних рішень і виявлення проблем.
- Співпраця із зацікавленими сторонами, дизайнерами інтерфейсу користувача та розробниками [25].

На рисунку 1.7 зображена блок-схема UX, яка описує систему продажу насіння для рослин. Вона ілюструє, де знаходиться кожна сторінка та параметр навігації, як люди взаємодіють із нею та до чого призводить кожна взаємодія [26].

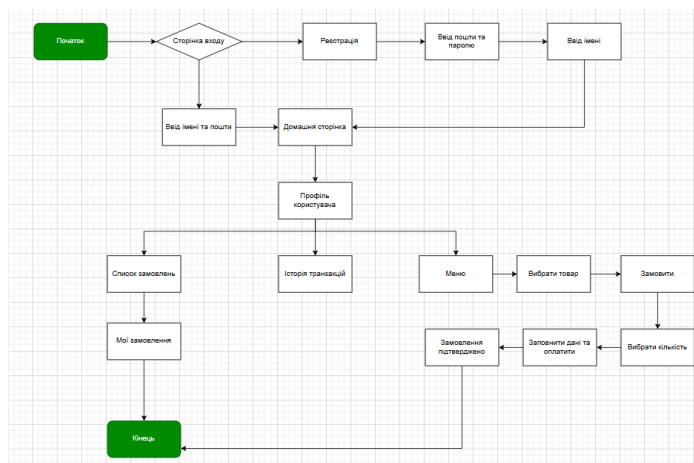


Рисунок 1.7 – Приклад UX блок-схеми для додатку

Подібні діаграми допомагають дизайнерам зрозуміти шлях користувачів через вебсайт. Завдяки цій інформації дизайнери оптимізують сайти, щоб користувачі могли виконати свої заплановані завдання [27].

Вибір засобів розробки для інтернет-магазину відіграють важливу роль, оскільки вони мають значний вплив на його ефективність, продуктивність і якість кінцевого продукту. Це робиться для того, щоб проект був кращим з точки зору і інтерфейсу користувача (UI) та взаємодії з користувачем (UX) [28].

1.4.2 Figma

Figma – це програма графічного дизайну для розробки прототипів і користувацьких інтерфейсів для цифрових продуктів, таких як програми для смартфонів або вебсайти [29].

Інструмент надає користувачам можливість додавати функції, включаючи маніпулювання текстом, зміну кольору, коригування зображення, наведення курсора, прокручування та анімацію до активів інтерфейсу.

Використовуючи інструменти векторної графіки, команди дизайнерів можуть створювати складні каркасні макети для вебсайтів, розмір яких можна змінювати для оптимізації для екранів будь-якого розміру.

Figma навіть дозволяє користувачам експортувати деяку кількість коду з проектів прототипу, щоб передати розробникам після вдосконалення прототипу [30].

Крім програми Figma розглядалися можливі варіанти, наприклад Canva, яка теж дозволяє створювати дизайн.

Canva є одним із найкращих прикладів платформи цифрового дизайну, яка розрізняє межу між професійним інструментом і простою у використанні системою навчання. Це інструмент графічного дизайну, який спрощує процес цифрового дизайну. Canva дозволяє редагувати зображення та навчатися на основі проектів за допомогою простого інтерфейсу. Вона пропонує безліч віртуальних шаблонів, щоб студенти могли працювати з дизайном і вносити власні зміни. Це дозволяє їм зосередитися на темі, що розглядається, не витрачаючи часу чи енергії на початок роботи з самим інструментом [31].

1.4.3 Порівняння програм для розробки дизайну

Щоб визначити, яка програма краще підійде для розробки дизайну, в університеті Бригама Янга, США, було проведено дослідження, де порівнювали два застосунки – Canva та Figma. Вісімнадцять студентів-дизайнерів з графіки брали участь у дослідженні. Вони зробили п'ять презентацій класу, двічі використовуючи Canva і двічі використовуючи Figma, а для п'ятої спільної презентації вони могли вибрати будь-який інструмент для презентації.

Їм було доручено створити 10-хвилинну презентацію для класу, яка включає факти про життя та навчання дизайнерів, зображення їхніх робіт, пояснення їхньої філософії дизайну і приклади того, як дизайнер вплинув на сучасний дизайн, якщо це можливо. Було надано 10-сторінковий шаблон Canva Design, як показано на рисунку 1.8, відформатований для використання в Instagram.



Рисунок 1.8 – Шаблон в Canva

Таке ж саме дослідження було з Figma, яка оптимізована для створення прототипів взаємодії вебсторінок і дозволяє створювати захоплюючі, спеціальні нелінійні презентації.

На рисунку 1.9 показано шаблон, який був наданий студентам.

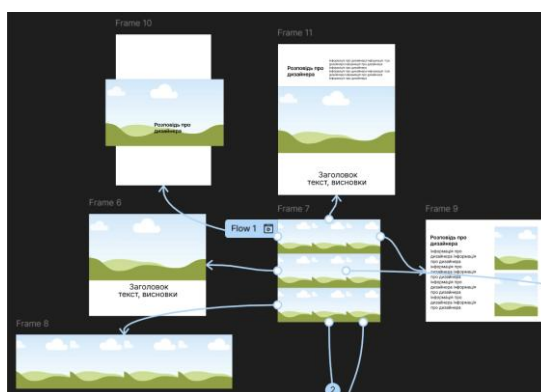


Рисунок 1.9 – Figma шаблон

Структура шаблону заохочувала взаємодію з аудиторією, де ведучі могли вибрати самі або запитати клас, який квадрат вмісту вони хочуть переглянути, таким чином даючи аудиторії можливість випробувати інформацію на основі своїх інтересів.

Протягом приблизно двох тижнів студенти досліджували, створювали та презентували п'ять різних презентацій.

У висновку було проведено опитування про програми та вибір студентів. Більшість вирішили використовувати Figma замість Canva для своєї остаточної презентації.

На запитання про функціонал студенти відповідали, що Canva легка у вивченні (87,5%) проте вважали, що вона стримує їх творчість. На відміну від першого опитування, всього 50% сказали, що Figma важка у вивченні проте

майже всі погодились, що будуть використовувати цю програму у майбутньому для, так як інтерактивні елементи відіграють важливу роль в презентаціях на вивченні інформації.

У висновку, яку б програму студенти рекомендували – 50% відповіли Figma аргументуючи це кращим та більш розширеним функціоналом порівняно з Canva [32].

1.5 Програмні засоби для розробки

Для написання коду використовувалася програма Microsoft Visual Studio Code – це безкоштовний, легкий редактор коду. Він включає в себе вбудовані мови, такі як JavaScript, TypeScript і Node.js та має розширення для інших мов програмування (таких як C++, C#, Python та PHP) [33].

Крім цієї програми, розглядалися кілька варіантів, які можуть бути аналогами.

Notepad++ – це редактор вихідного коду та заміна блокнота, який підтримує кілька мов у середовищі MS Windows із підсвічуванням і згортанням синтаксису та надає додаткові функції, яких немає в стандартному текстовому редакторі Windows, Notepad [34].

Sublime text 3 – це текстовий редактор для коду, який підтримує різні мови програмування. Основні переваги включають швидкість роботи, легкий інтерфейс, можливість налаштування через плагіни та теми, а також розширені функції, такі як «Goto Anything» для швидкого переходу між файлами і «Multiple Selections» для редагування кількох рядків одночасно. До того ж, Sublime має автозавершення коду – в тому числі для змінних, створених користувачем – і згортання коду. Останній робить великі фрагменти коду більш компактними для перегляду, приховуючи його частини [35].

У таблиці 1.2 було сформовано порівняння даних програм на основі сайту, який дозволяє порівнювати додатки за різними категоріями, такими як простота адміністрування, якість підтримки та загальна оцінка від

користувачів. Користувачі можуть ознайомитися з рейтингами та відгуками, щоб обрати найкраще програмне забезпечення для своїх потреб.

Таблиця 1.2 – Порівняльна характеристика програм [36]

Критерії/Програми	Notepad++	Sublime Text	Visual Studio Code
Простота використання	9.3	9.1	9.2
Якість підтримки	8.5	8.2	8.7
Функціональність	8.5	8.6	9.1
Кодування	8.8	8.9	9.1
Загальний рейтинг	8.6	8.6	9.1

Якщо порівнювати ці програми, то найпростішою є Notepad++, проте дивлячись на загальні характеристики, найкращою програмою буде Microsoft Visual Studio (9.1). В даній програмі найвищий рейтинг з критеріїв для кодування – авто виправлення, розширення, можливість взаємодіяти з Git [36].

Оновленням у Visual Studio Code є інтеграція з Copilot, яка допомагає швидше писати код за допомогою штучного інтелекту. Оновлення включає покращену підтримку для автозаповнення коду та створення функцій на основі коментарів, що значно підвищує продуктивність [37].

1.6 Розміщення сайту на хостингу

1.6.1 Netlify

Для того, щоб перевірити роботу сайту, буде використано безкоштовний хостинг Netlify.

Netlify – це платформа для хостингу та автоматизації розгортання вебпроектів. Він дає змогу розробникам для створення високопродуктивних і

динамічних вебсайтів, магазинів електронної комерції та вебдодатків на своїй платформі [38].

Такі функції на платформі можуть використовуватися для різних цілей:

- Обробка запитів API: для аутентифікації користувачів або отримання даних із бази [39].
- Надсилання сповіщень: для надсилання електронних листів користувачам або створення сповіщення у месенджерах.

Для стартапів Netlify все частіше стає стандартним рішенням для швидкого та дешевого розгортання прототипів та MVP завдяки своїй простоті та сумісності з іншими технологіями, що робить його улюбленим для потенційних підприємців [40].

1.6.2 AWARDSPACE

У межах роботи, було обрано «AwardSpace» як хостинг і вебсервер для розробки інтернет-магазину. Awardspace.net не лише надає надійний хостинг, але й має великий набір програмних інструментів для створення та розвитку онлайн-ресурсів будь-якої величини. Відвідавши сайт цього хостера, можна з легкістю дізнатися про його переваги та актуальні пропозиції [41].

Netlify пропонує зручну функцію для обробки форм, яка дозволяє легко збирати дані від користувачів без необхідності налаштовувати окрему базу даних [42]. Проте в розробці інтернет-магазину буде підключена база даних, тому для подання функцій та її підключення краще обрати цей хостинг через такі переваги:

- Доступ до панелі керування: Awardspace пропонує зручну панель управління, де можна створювати та керувати базами даних без спеціальних знань програмування.
- Інтеграція з РНР: Можливість використовувати РНР для створення динамічних сторінок магазину, що інтегруються з базою даних для швидкого відображення інформації.

У таблиці 1.3 подано порівняння характеристик двох хостингів.

Таблиця 1.3 – Порівняльна характеристика хостингів [36]

Характеристика/Хостинг	Awardspace	Netlify
Об'єм пам'яті	1 Гб	Необмежена
Додаткові домени	так	ні
Бази даних	MySQL	Вбудована
Підтримка PHP	так	ні

На відміну від Netlify, який оптимізований для статичних сайтів, Awardspace забезпечує повноцінну серверну підтримку (PHP, MySQL). Це дозволяє реалізувати динамічні функції, необхідні для інтернет-магазину, такі як авторизація, пошук продуктів, обробка платежів [43].

1.7 Постановка задач для інтернет-магазину «BLOOM»

Для реалізації дипломного проєкту визначено такі ключові задачі:

- Провести аналіз сучасних тенденцій UI/UX – дослідити актуальні дизайнерські рішення, принципи зручності, адаптивності та візуальної привабливості користувацького інтерфейсу, це може суттєво вплинути на позицію сайту в пошукових системах [44].
- Обрати інструменти для проєктування сайту – визначити технології, мови програмування та середовища, що найкраще підходять для реалізації функціоналу та дизайну інтернет-магазину.
- Виділити акторів та побудувати варіанти використання (use case) – ідентифікувати основних користувачів системи та описати їх взаємодію з ключовими функціями вебсайту.

- Спроекувати базу даних – створити структуру таблиць, визначити зв'язки між ними, типи даних та забезпечити збереження й обробку інформації.
- Розробити логіку клієнт-серверної взаємодії – описати механізм обміну даними між користувацьким інтерфейсом (frontend) та серверною частиною (backend).
- Протестувати функціональність сайту – перевірити коректність роботи основних модулів, зокрема: реєстрації користувачів, кошика, оформлення замовлення, відображення товарів тощо.

Реалізація зазначених етапів спрямована на успішне досягнення поставленої мети проєкту.

1.8 Висновок до першого розділу

У першому розділі кваліфікаційної роботи проведено аналіз предметної області – інтернет-магазин, огляд його конкурентів, постановку завдання та вибір середовища розробки для його створення.

Проаналізовано рішення на ринку інтернет-магазинів на прикладі двох конкурентів: Sparkle Ternopil та Million flowers. Визначено їх сильні та слабкі сторони, що допомогло обрати один з найкращих підходів розробки та уникнути помилок при розробці власного проєкту.

Розглянуто варіанти середовищ розробки, які можуть бути використані для створення інтернет-магазину. Проаналізовано їх переваги та недоліки – Sublime Text, Notepad++ та VS Code. Обрано VS Code завдяки його великій бібліотеці плагінів та розширень, а також низькому використанню системних ресурсів.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ «BLOOM»

2.1 Моделювання архітектури та структури інтернет-магазину «BLOOM»

Як зазначено у меті кваліфікаційної роботи, наступний етап – розробка програмного забезпечення для інтернет-магазину. В другому розділі розглядаються такі етапи виконання, як проектування архітектури.

2.2 Моделювання архітектури інтернет-магазину квітів

Архітектура програмного забезпечення є надзвичайно важливим та основним етапом у розробці програмного забезпечення. Даний процес являється фундаментальною та її невід’ємною частиною. Було зроблено висновок, що архітектура – це розділення системи на структурні елементи [45].

У праці Дмитроци Лесі та Палки Олега «Аналіз мікросервісної архітектури «Розумного міста»» здійснено аналіз архітектур для порівняння алгоритмів та програмної реалізації міських рішень із можливістю їх подальшого впровадження та наявності вже реалізованих в Україні [46].

Мікросервісні архітектури народжуються як рішення цієї проблеми. Завдяки реалізації однієї функції, обмеженої в межах запущеного процесу, незалежні мікросервіси можуть бути розгорнуті окремо в розподіленій системі. У цій статті продемонстровано, як мікросервісна архітектура може бути використана для створення послуг інтернету речей для мультимобільності [46].

Для інтернет магазину буде використано архітектуру клієнт-сервер – це модель, у якій клієнт ініціює запити до сервера, що їх обробляє і повертає результати. Така архітектура забезпечує централізований доступ до даних і контроль безпеки, але може мати обмеження в масштабованості. Було обрано 3-рівневу архітектуру клієнт-сервер, у якій зазначені конкретні рівні: клієнт, що

подає запит, сервер, який обробляє ці запити та база даних, яка зберігає інформацію (див. рис. 2.1) [47].

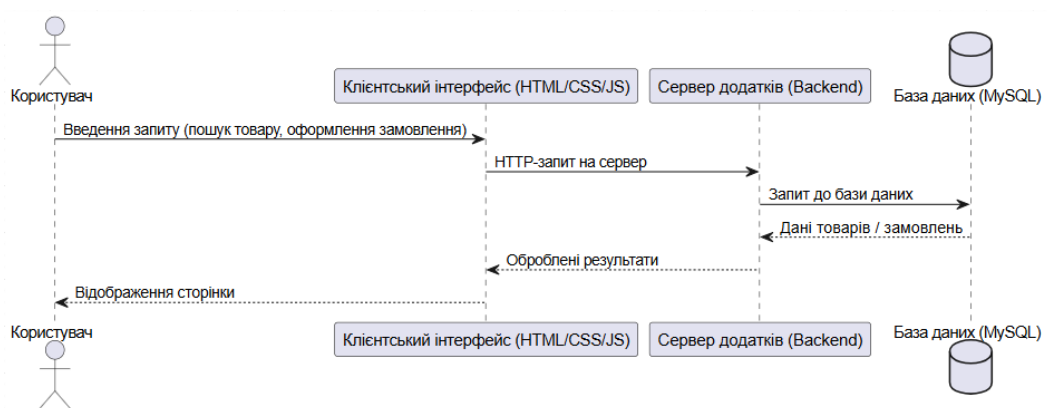


Рисунок 2.1 – Архітектура інтернет-магазину

Для кращого розуміння процесу взаємодії користувача з інтернет-магазином квітів «BLOOM» було розроблено алгоритм виконання замовлення. Алгоритм відображає основні етапи – від моменту вибору товару клієнтом до підтвердження замовлення менеджером і його подальшого виконання.

На рисунку 2.2 зображено алгоритм процесу замовлення в інтернет-магазині «BLOOM».



Рисунок 2.2 – Алгоритм процесу замовлення в інтернет-магазині «BLOOM»

Така схема дозволяє наочно представити послідовність дій і відповідальних осіб на кожному етапі процесу замовлення. Впровадження чітко структурованого алгоритму забезпечує ефективну обробку замовлень, мінімізує ризики помилок і сприяє підвищенню рівня задоволеності клієнтів

Здатність розробляти алгоритми дає можливість ефективно опрацювати процес створення замовлення в інтернет-магазині – від вибору товарів і формування кошика до підтвердження замовлення й обробки даних користувача. [48].

Якщо дотримуватись чіткого алгоритму дій, оптимізація бізнес-процесів відбуватиметься ефективно та без зайвих затримок [49].

2.3 Варіанти використання інтернет-магазину квітів

Для інтернет-магазину вибір акторів є важливою частиною, щоб зрозуміти які функції кожен може виконувати та який вплив мають один на одного.

На рисунку 2.3 зображено структуру акторів інтернет-магазину квітів «BLOOM», яка включає неавторизованого користувача, авторизованого користувача та адміністратора. Неавторизований користувач може переглядати каталог товарів, додавати товари до кошика, авторизуватися або зареєструватися. Авторизований користувач має розширений доступ: оформлення замовлень та перегляд історії замовлень. Адміністратор володіє правами управління системою: додавання, редагування та видалення товарів, а також управління замовленнями [50].

Після вибору акторів потрібно описати їх варіанти використання, що допомагає зрозуміти як система буде використовуватись різними користувачами [51].

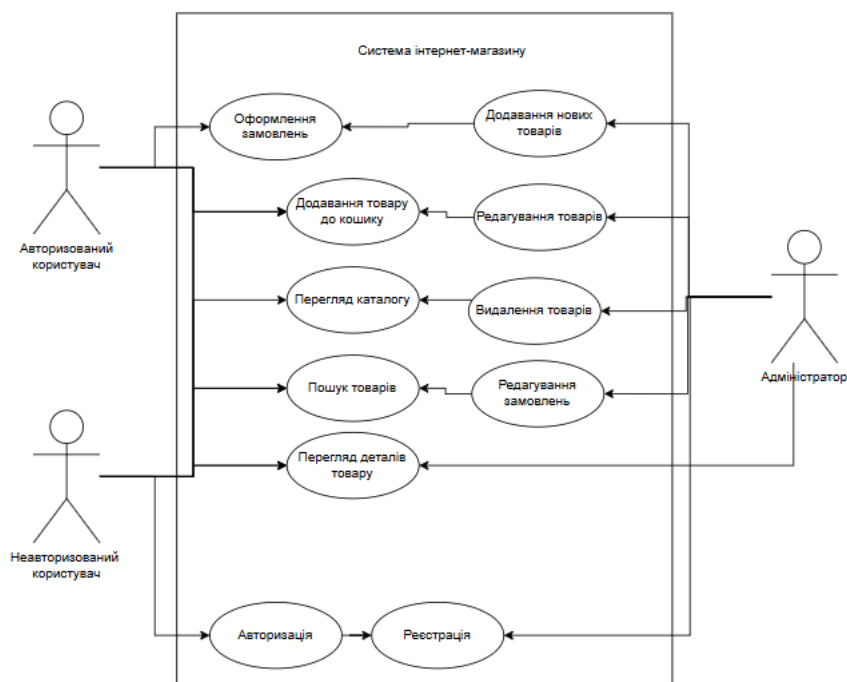


Рисунок 2.3 – Актори для інтернет-магазину «BLOOM»

Окрім загального опису системи інтернет-магазину, у цьому підрозділі також представлено діаграми використання, які ілюструють основні функції кожного типу користувача: неавторизованого користувача, авторизованого користувача та адміністратора.

На рисунку 2.4 подано діаграму використання для неавторизованого користувача.

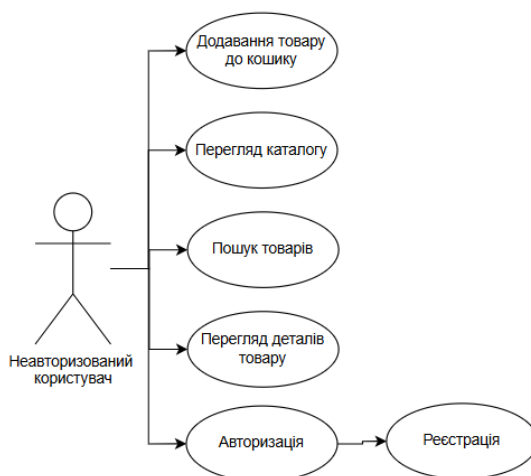


Рисунок 2.4 – Діаграма використання для неавторизованого користувача

Неавторизований користувач має дещо обмежений функціонал сайту, наприклад він може додати товар у кошик, але при оформленні замовлення – буде запрошено увійти в систему або зареєструватися. Це виконано для того щоб покращити безпеку та захистити інтернет-магазин і користувачів від неавторизованого доступу [52]. Тільки після того, як він авторизується, отримає функції авторизованого користувача. Проте якщо раніше такого не було, то доступна функція реєстрації у інтернет-магазині квітів. Після того як реєстрація була успішна, можна авторизуватись.

На рисунку 2.5 подана діаграма використання для авторизованого користувача.

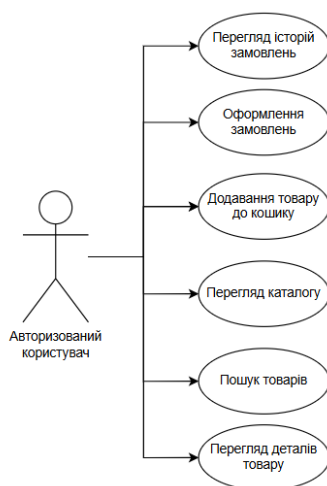


Рисунок 2.5 – Діаграма використання для авторизованого користувача

Авторизований користувач, в свою чергу, має більш розширений функціонал, що дає йому можливість оформити замовлення та переглядати їх історію.

Найбільше привілеїв має адміністратор. Діаграма використання адміністратора зображена на рисунку 2.6.

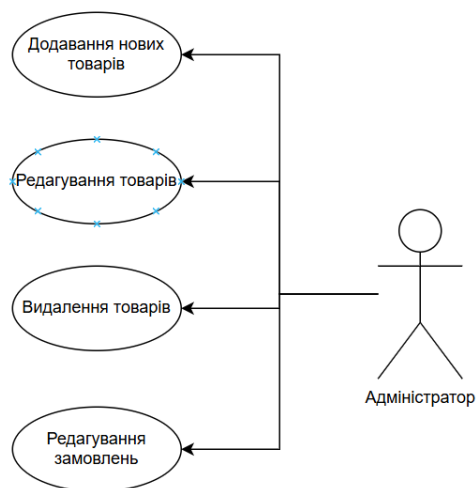


Рисунок 2.6 – Діаграма використання для адміністратора

Адміністратор має доступ до адміністративної панелі та товарів, він може створювати, редагувати та видаляти товари. Такий користувач потрібен для того, щоб перевіряти та слідкувати за даними у інтернет-магазині квітів [53].

2.4 Розробка структури таблиць для інтернет-магазину «BLOOM»

База даних для магазину квітів є важливою частиною, тому що вона визначає як дані будуть структуровані та збережені. Завдяки хорошій структурі запити працюють швидше та сайт тоді не «гальмує». Іншими аспектами хорошої структури є узгодженість (зменшуються дублювання та помилки, якщо дані про користувача зберігаються в одному місці – зміни відображаються скрізь) та дає змогу легко додавати новий функціонал, що розширює можливості магазину без значного перероблення бази даних [54].

На рисунку 2.7 подана створена база даних із таблицями.

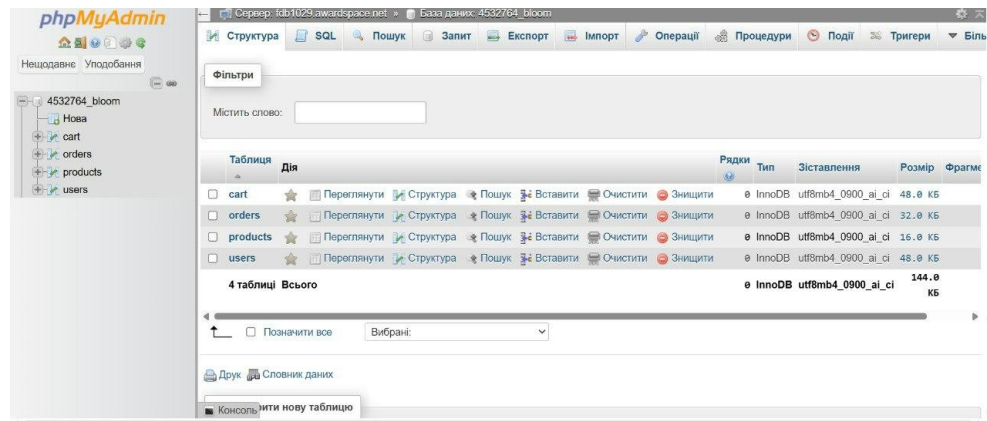


Рисунок 2.7 – Таблиці бази даних для інтернет-магазину

У таблиці 2.1 подана структура таблиці «cart» з переліком полів, ключів та значень. Вона містить список букетів та квіткової продукції, доступних для продажу.

Таблиця 2.1 – Структура таблиці «cart»

Ключ	Ім'я поля	Домен атрибуту	Значення
PPK	id	int	ID запису в кошику
FK	product_id	int	ID товару
	quantity	int	Кількість товару в кошику
FK	user_id	int	ID користувача (для авторизованих)

На рисунку 2.8 зображено структуру таблиці.

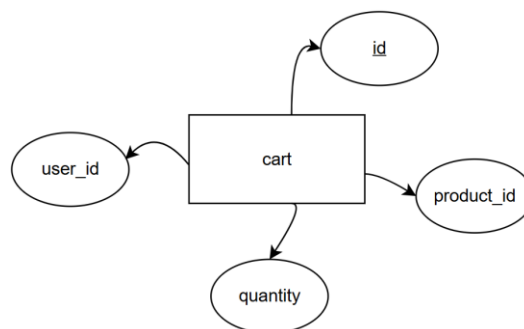


Рисунок 2.8 – Структура таблиці «cart»

У таблиці 2.2 подана структура таблиці «orders» з переліком полів, ключів та значень. Вона містить інформацію про оформлення замовлення в інтернет-магазині квітів, включаючи дані про товар, контактну інформацію покупця, адресу доставки, спосіб оплати, загальну вартість замовлення та дату його оформлення.

Таблиця 2.2 – Структура таблиці «orders»

Ключ	Ім'я поля	Домен атрибуту	Значення
PK	id	int	Унікальний ID замовлення
FK	product_id	int	ID букета, який був замовлений
	user_email	varchar(255)	Електронна пошта покупця
	user_phone	varchar(255)	Телефон покупця
	user_name	varchar(255)	Ім'я покупця
	user_address	text	Адреса доставки
	payment_method	varchar(50)	Спосіб оплати
	price	decimal(10,2)	Загальна сума замовлення
	order_date	datetime	Датата час оформлення замовлення

На рисунку 2.9 зображена структура таблиці «orders».

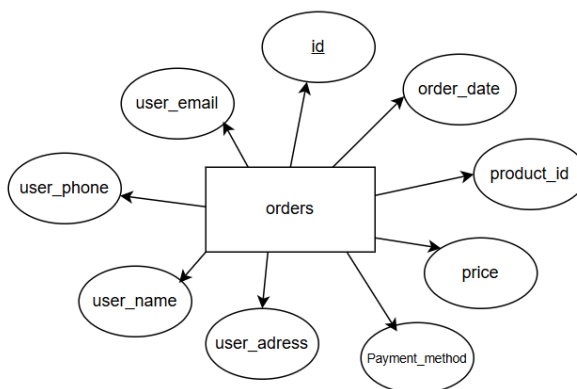


Рисунок 2.9 – Структура таблиці «orders»

У таблиці 2.3 подана структура таблиці «products» з переліком полів, ключів та значень. Вона містить містить список товарів, які продаються в інтернет-магазині. Кожен запис у цій таблиці – це один товар.

Таблиця 2.3 – Структура таблиці «products»

Ключ	Ім'я поля	Домен атрибуту	Опис
PK	id	int	ID товару
	name	varchar(255)	Назва товару
	description	text	Опис товару
	price	decimal(10,2)	Ціна товару
	image	varchar(255)	Назва зображення
	created_at	datetime	Дата та час створення запису

На рисунку 2.10 зображена структура таблиці «products».

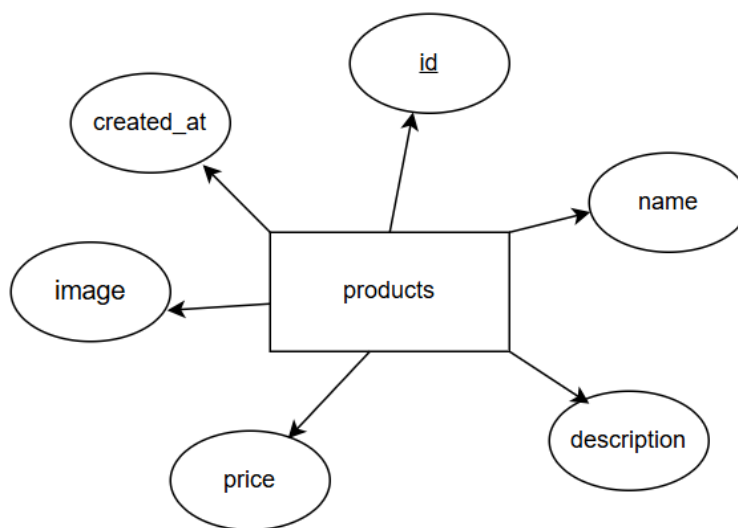


Рисунок 2.10 – Структура таблиці «products»

У таблиці 2.4 подана структура таблиці «users» з переліком полів, ключів та значень. Вона зберігає дані зареєстрованих користувачів інтернет-магазину.

Таблиця 2.4 – Структура таблиці «users»

Ключ	Ім'я поля	Домен атрибуту	Опис
PK	id	int	Унікальний ідентифікатор користувача
FK	email	varchar(255)	Електронна пошта
	password	varchar(255)	Пароль
	created_at	datetime	Дата та час реєстрації користувача

На рисунку 2.11 подано структуру таблиці.

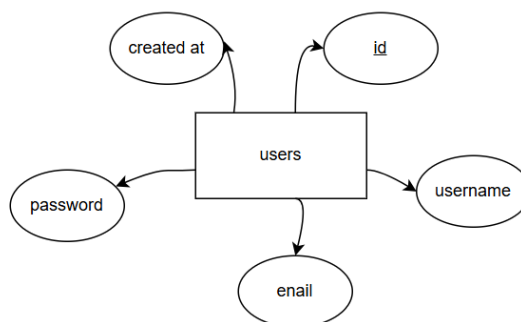


Рисунок 2.11 – Структура таблиці «users»

Було виділено наступні типи кардинальностей між таблицями:

- User-Cart – один до багатьох. Один користувач може вибрати кілька продуктів. У кошику користувача може бути кілька різних товарів.
- Products-Cart – один до багатьох. Один продукт може знаходитись у багатьох записах кошика.
- Products-Orders – один до багатьох. Один товар (Product) може бути замовлений у багатьох замовленнях (Orders). Одне замовлення (Order) зазвичай містить кілька товарів (Products).

На рисунку 2.12 зображено зв'язки між таблицями бази даних інтернет-магазину.

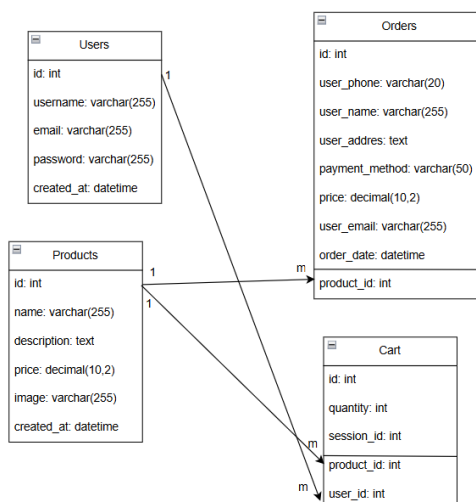


Рисунок 2.12 – Зв'язки між таблицями

Таблиця orders пов'язана з таблицею «products» через поле product_id. Таблиця «cart» пов'язана з products через поле product_id та з «users» через поле user_id. Таблиця «users» має унікальні поля username та email, які не повторюються.

Ці зв'язки дозволяють: зберігати інформацію про те, хто зробив замовлення, які саме товари додано до кошика та обробляти оплату та доставку.

2.5 Висновки до другого розділу

У другому розділі було розроблено архітектуру інтернет-магазину квітів «BLOOM», що базується на тришаровій моделі клієнт-серверної архітектури.

Проведено моделювання варіантів використання для кожного типу користувачів, що дозволяє чітко визначити функціональні можливості кожного.

Описано структуру бази даних із встановленням зв'язків між таблицями для ефективного збереження та обробки інформації про товари, користувачів і замовлення.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ «BLOOM»

3.1 Розробка інтерфейсу та серверної частини для інтернет-магазину «BLOOM»

У процесі розробки інтернет-магазину «BLOOM» було використано мови HTML, CSS та JavaScript для створення клієнтської частини – frontend (процеси, які помітні покупцю й здатні безпосередньо впливати на його вибір). За допомогою PHP реалізовано серверну частину – backend (процеси, які впливають на роботу інтернет магазину, але не помітні користувачам), яка відповідає за обробку запитів, взаємодію з базою даних, збереження інформації про замовлення та користувачів [43].

3.1.1 Розробка frontend-частини для інтернет-магазину квітів

Під час розробки сайту для інтернет-магазину «BLOOM» було реалізовано клієнтську частину (frontend). Основними задачами є розробка головної сторінки, сторінки каталогу товарів, інформація про товар та сторінку оформлення замовлення.

Для початку був створений прототип сайту у додатку Figma (див. рис. 3.1), що дозволив візуалізувати структуру майбутнього інтерфейсу та погодити розміщення основних блоків, кольорову гаму й типографіку до початку верстки. Прототип включав дизайн головної сторінки, каталогу, сторінку про товар а також оформлення замовлення. Крім того було спроектовано дизайн мобільної версії, що дало змогу ефективно спланувати адаптивність і послідовність елементів на різних пристроях.

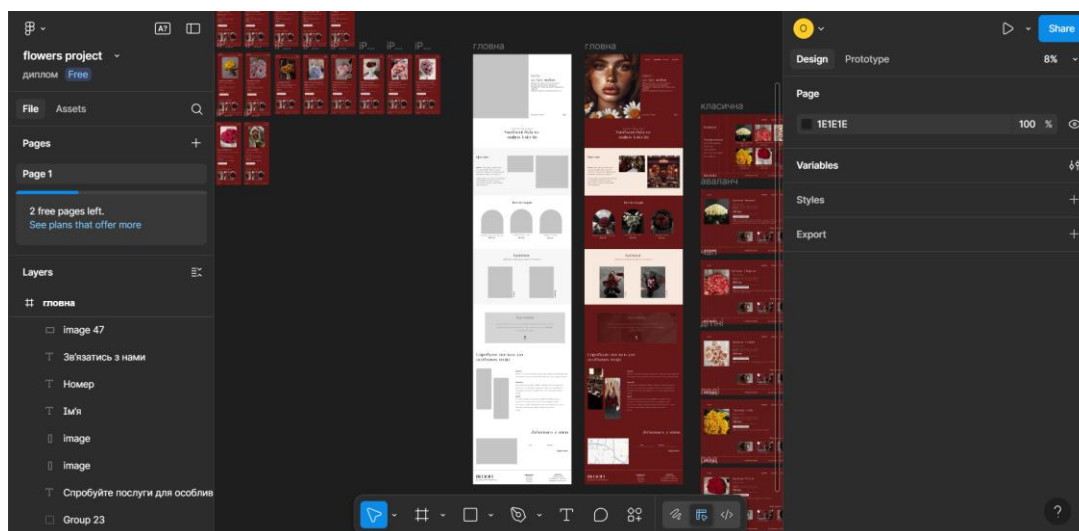


Рисунок 3.1 – Створення прототипу у Figma

Дизайн був побудований із застосуванням основних принципів UX/UI-дизайну, що передбачає логічне розміщення навігаційних елементів, візуальну рівновагу та комфортне сприйняття інформації.

Основною кольоровою палітрою сайту стали теплі винні відтінки, які асоціюються з емоціями, чуттєвістю та елегантністю, а також бежевий фон #FFF5ED, що додає легкості й балансу [55].

У типографіці використовувались два ключові шрифти:

- Montserrat для основного текстового контенту,
- Мак для акцентних заголовків, що підкреслюють унікальність бренду [56].

Головна сторінка має розділену структуру: навігаційне меню, рекламний банер із зображенням, навігаційна панель, що складається із кількох секцій: «Про нас», «Каталог», а також блоки «Доставка» та «Контакти». У лістингу 3.1 подано частину коду для реалізації навігаційного меню. Повний код подано в додатку Б.

Лістинг 3.1 – Частина коду навігаційного меню

```
<header>
  <div class="header-container">
    <div class="logo">
```

```

        
    </div>
    <nav>
        <ul>
            <li><a href="flower.php">Каталог</a></li>
            <li><a href="index.html">Головна</a></li>
            <li><a href="index.html#about">Про нас</a></li>
            <li><a
href="index.html#delivery">Доставка</a></li>
        </ul>
    </nav>
</div>
</header>

```

На рисунку 3.2 зображено навігаційне меню інтернет-магазину «BLOOM»

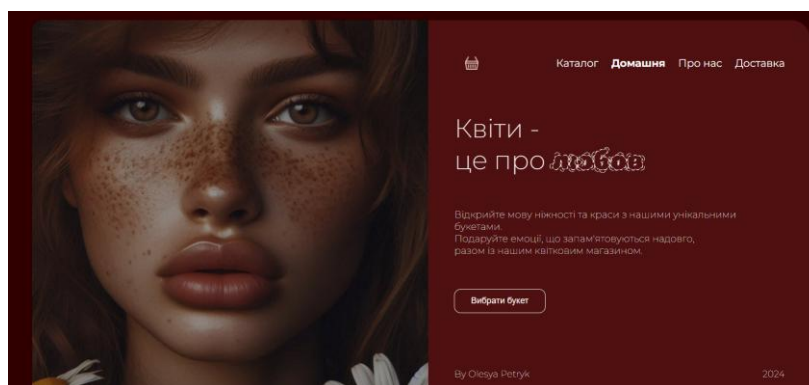


Рисунок 3.2 – Навігаційне меню

Окрім навігаційного меню, користувач також бачить зображення дівчини, яке є візуально яскравим акцентом сторінки. Праворуч від зображення розташований текстовий блок із заголовком, що відображає ідею магазину, «Квіти - це про любов».

Під заголовком знаходиться короткий опис та кнопка «Вибрати букет», яка веде користувача до каталогу товарів, що значно спрощує ознайомлення з асортиментом.

«Головна» – повертає користувача на головну сторінку;

«Про нас» – при натисканні виконує прокрутку до секції з інформацією про магазин (див. рис. 3.2);

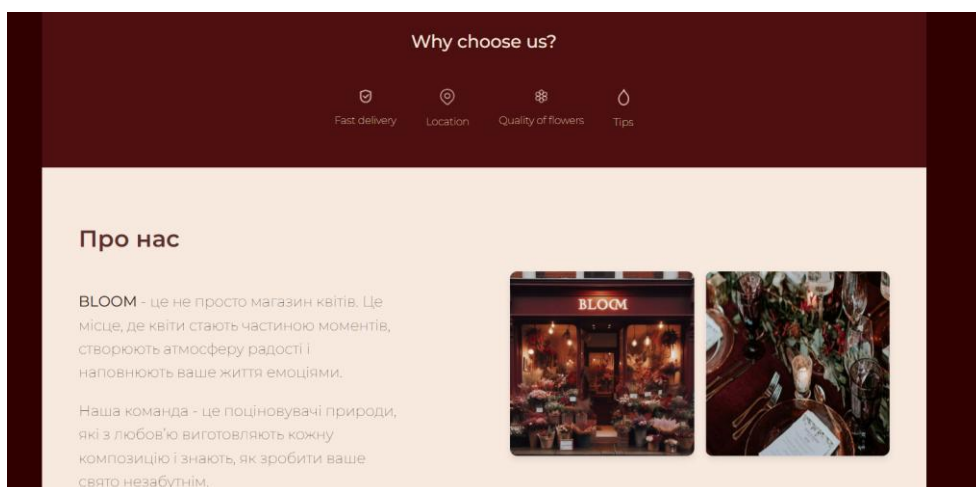


Рисунок 3.2 – Частина сторінки «Про нас»

«Доставка» – прокручує до секції з умовами доставки замовлень (див. рис. 3.3);

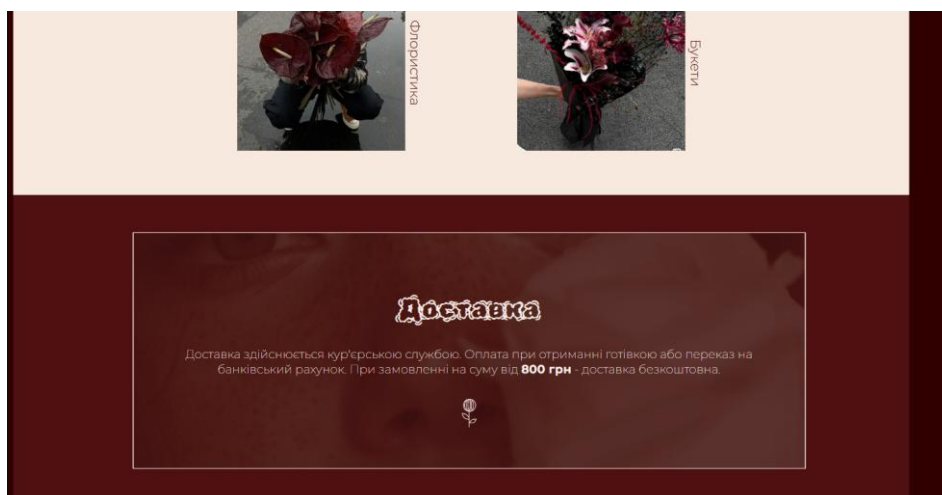


Рисунок 3.3 – Частина сторінки «Доставка»

«Каталог» – відкриває сторінку з добіркою букетів, де можна швидко знайти потрібний товар (див. рис. 3.4). Код сторінки поданий у додатку В.

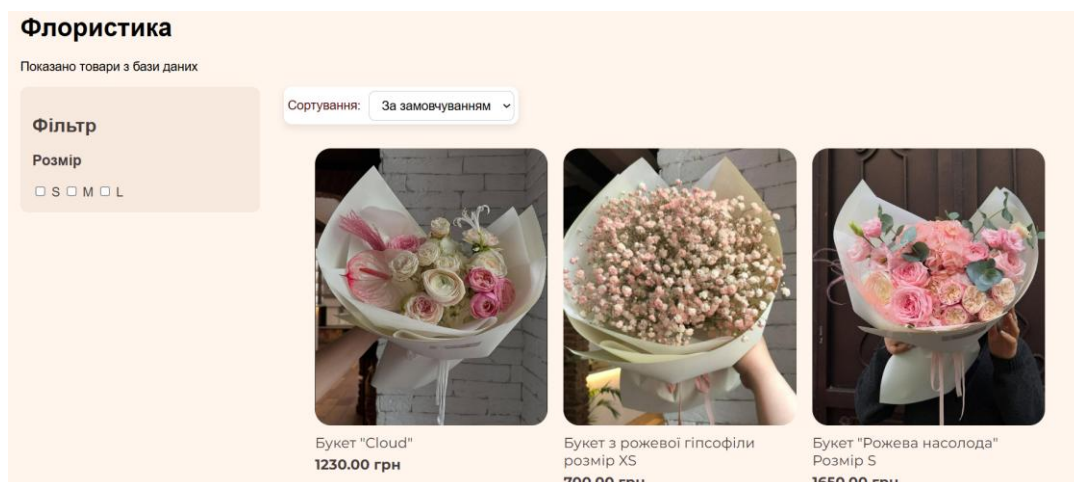


Рисунок 3.4 – Каталог інтернет-магазину

Користувач може обрати будь-який букет, і натиснувши на нього, відкриється сторінка з детальною інформацією. На рисунку 3.5 зображено інформацію про букет «Рожева насолода».

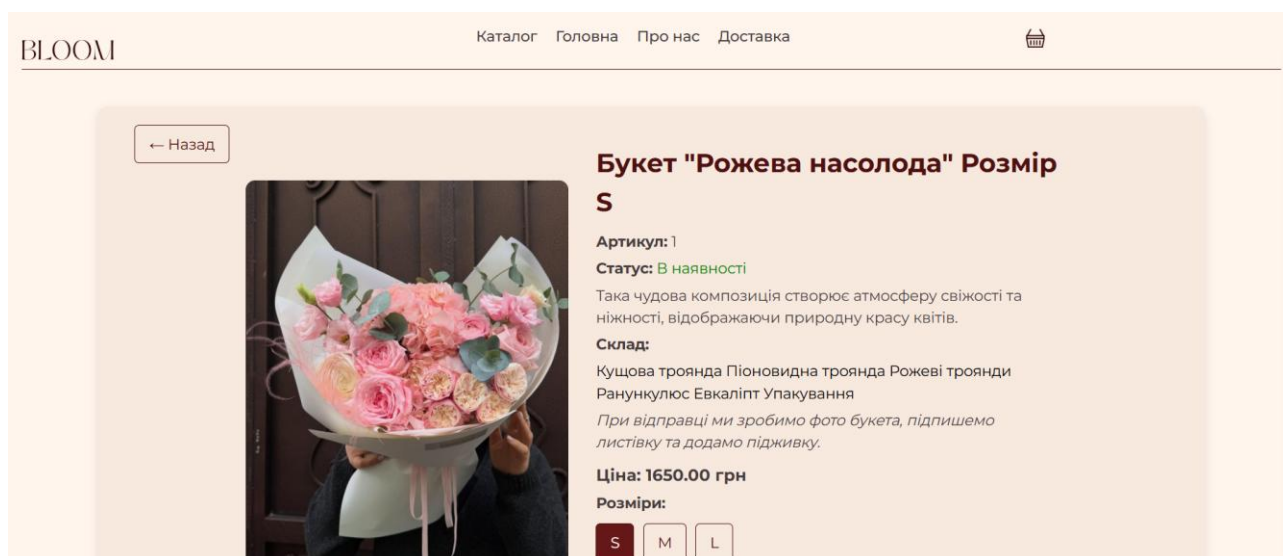


Рисунок 3.5 – Детальна інформація про букет

На цій сторінці розміщена інформація про букет, а також ціна до кожного розміру. Код реалізації поданий у лістингу 3.2. Повний код подано в додатку Д.

Лістинг 3.2 – Реалізація ціни та розміру букету

```
<div class="size-options">
    <?php foreach ($sizes as $size): ?>
```

```

        <button type="button" class="size-btn"
data-price="<?= $size['price'] ?>" data-size="<?= $size['size']
?>">

        <?= htmlspecialchars($size['size']) ?>
    </button>
    <?php endforeach; ?>
    <input type="hidden" name="size" id="selected-
size" value="<?= $sizes[0]['size'] ?? ' ' ?>" />
    <input type="hidden" name="price"
id="selected-price" value="<?= $sizes[0]['price'] ?? '0' ?>" />
</div>

```

Також можна додати товар у кошик, де показані букети які вибрав користувач – це той товар, який користувачеві сподобався і він планує купити (див. рис. 3.6), а також одразу купити.

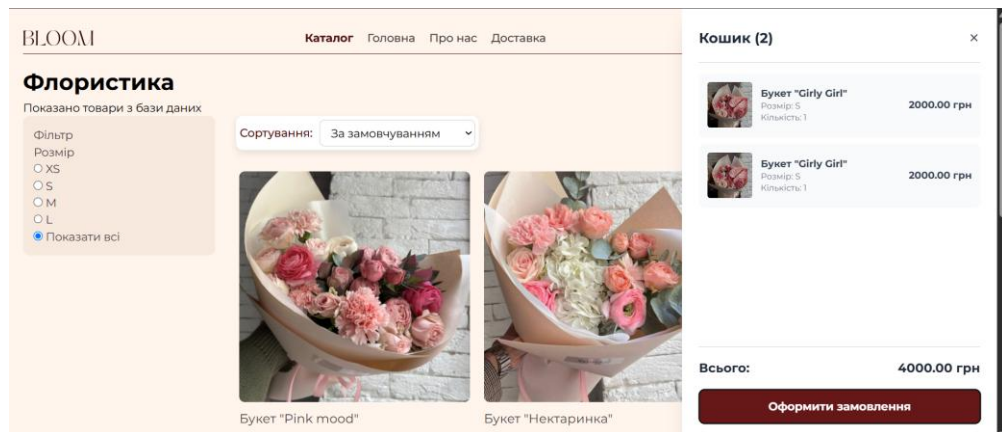
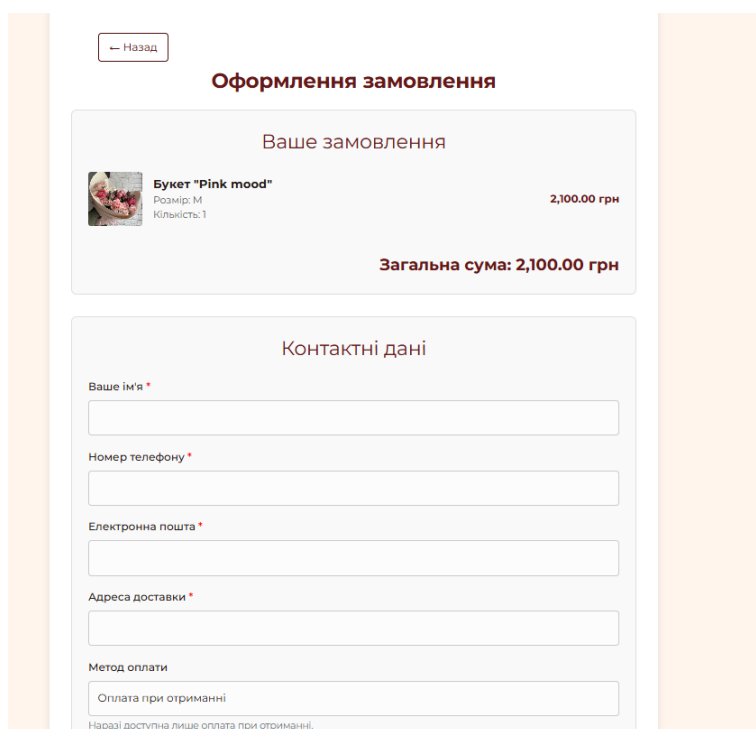


Рисунок 3.6 – Реалізація корзини

Після цього користувач може оформити замовлення. На рисунку 3.7 зображено детально який букет замовлено, яка сума та можна ввести дані відправника, також вказана адреса, оплата та можна ввести коментар для замовлення. Код реалізації сторінки подано в додатку Е.



← Назад

Оформлення замовлення

Ваше замовлення

 **Букет "Pink mood"**
Розмір: М
Кількість: 1

2,100.00 грн

Загальна сума: 2,100.00 грн

Контактні дані

Ваше ім'я *

Номер телефону *

Електронна пошта *

Адреса доставки *

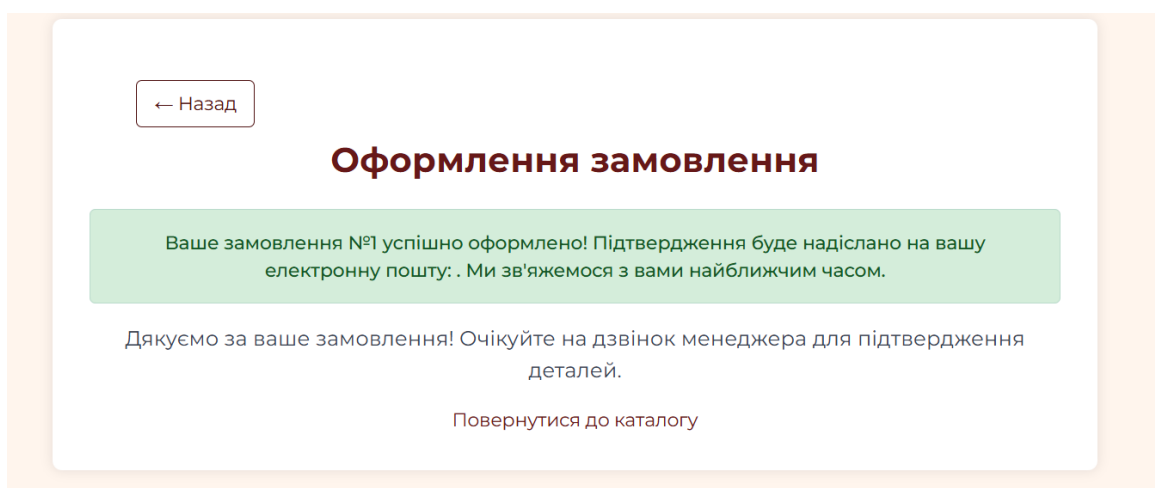
Метод оплати

Оплата при отриманні

Наразі доступна лише оплата при отриманні.

Рисунок 3.7 – Оформлення замовлення

При успішному оформленні з'явиться повідомлення, зображено на рисунку 3.8.



← Назад

Оформлення замовлення

Ваше замовлення №1 успішно оформлено! Підтвердження буде надіслано на вашу електронну пошту: . Ми зв'яжемося з вами найближчим часом.

Дякуємо за ваше замовлення! Очікуйте на дзвінок менеджера для підтвердження деталей.

[Повернутися до каталогу](#)

Рисунок 3.8 – Повідомлення про успішне оформлення замовлення

В іншому випадку, буде повідомлення про помилку (див. рис. 3.9).

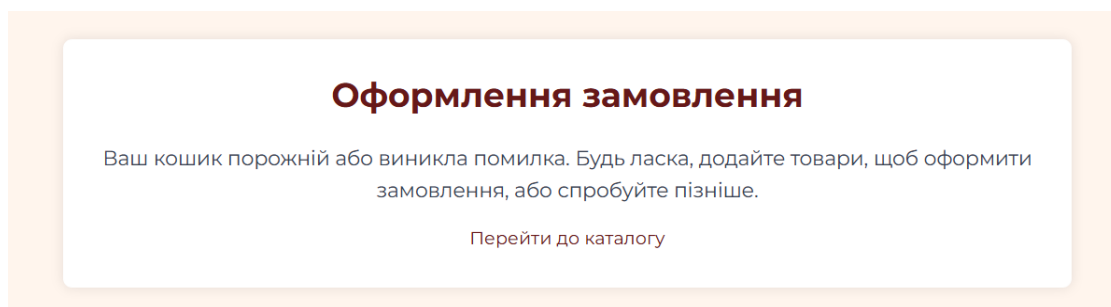


Рисунок 3.9 – Помилка оформлення замовлення

Для стилізації інтерфейсу застосовано таблиці стилів (CSS), зокрема сучасні підходи до компоновання елементів [58]. Усі стилі зібрано у зовнішньому файлі style.css. Приклад застосування подано у лістингу 3.3.

Лістинг 3.3 – Застосування стилів

```
body {  
    margin: 0;  
    font-family: 'Montserrat', sans-serif;  
    font-weight: 100;  
    display: flex;  
    flex-direction: column;  
    justify-content: center;  
    align-items: center;  
    background-color: #300000;
```

У цьому лістингу вказані стилі для головної сторінки інтерфейсу, що включають налаштування шрифтів, кольору фону та розміщення елементів. Властивості flexbox використовуються для гнучкого розташування елементів сторінки, де весь контент вирівнюється по центру як по вертикалі, так і по горизонталі, що надає інтерфейсу візуальну гармонію. Додатково, стиль для елемента body передбачає автоматичне регулювання висоти блоку в залежності від вмісту, а також додається відступ зверху, щоб забезпечити комфортний інтерфейс користувача.

Щоб забезпечити адаптивність дизайну під різні типи пристроїв (десктопи, планшети, смартфони), у таблиці стилів було використано медіа-запити. Вони дозволяють змінювати розміщення, розміри та зовнішній вигляд

елементів залежно від ширини екрана користувача. Це особливо важливо для збереження зручності навігації та читабельності на мобільних пристроях.

У лістингу 3.4 подано приклад медіа-запиту.

Лістинг 3.4 – Медіа-запит для головної сторінки

```
@media (max-width: 480px) {  
  .order-container {  
    width: 90%;  
    padding: 20px; }  
  .cart-item-image {  
    width: 70px;  
    height: 70px; }  
  .confirm-btn {  
    padding: 10px;  
    font-size: 16px; }  
}
```

У цьому медіа-запиті використано правило `@media (max-width: 480px)`, яке застосовує стилі лише тоді, коли ширина екрану пристрою не перевищує 480 пікселів, що зазвичай характерно для мобільних телефонів [57]. Для елемента `.order-container` задано ширину 90% екрану та внутрішній відступ 20px, що дозволяє вмісту займати більшу частину доступного простору і забезпечує комфортне відображення на малих екранах.

Зменшено розміри зображення товару в кошику через властивості `width: 70px` та `height: 70px`, що дозволяє краще використовувати обмежений простір на мобільних пристроях. Ці зміни дозволяють покращити адаптивність інтерфейсу та забезпечити зручність користування сайтом на мобільних пристроях.

На рисунку 3.10 зображено приклад адаптації сайту.

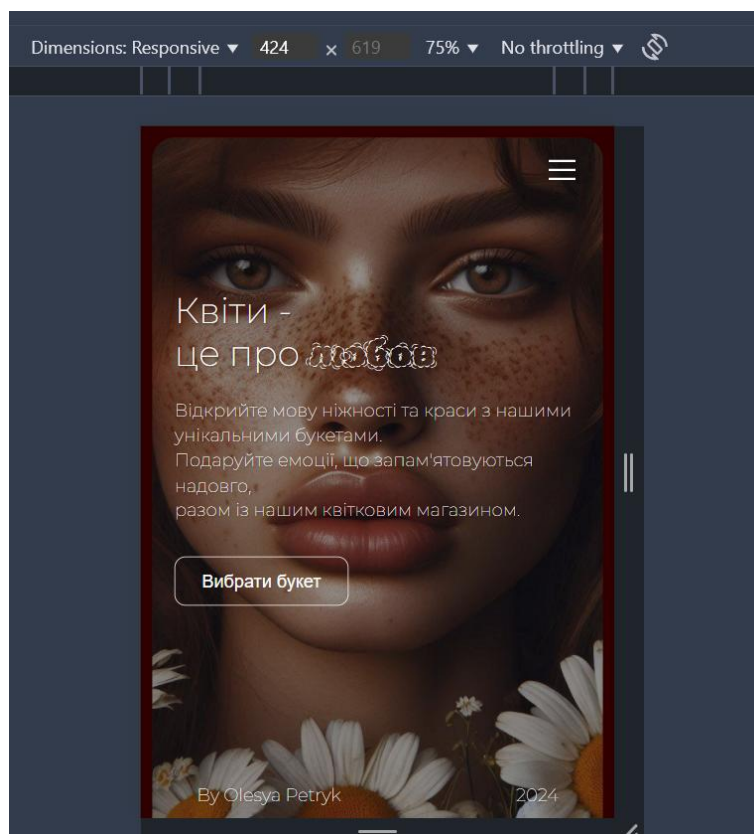


Рисунок 3.10 – Приклад адаптації головної сторінки сайту

У рамках розробки сайту для інтернет-магазину була реалізована адміністративна панель, що дозволяє адміністратору керувати каталогом товарів. Сторінка доступна лише після авторизації під обліковим записом адміністратора. Через цю панель можна додавати нові товари до бази даних, заповнюючи поля з назвою, описом, складом, наявністю, а також завантажуючи зображення товару.

Крім того, передбачена можливість додавати один або кілька розмірів з відповідними цінами. Всі введені дані перевіряються, що забезпечує цілісність записів у таблицях `products` та `product_sizes`. Інтерфейс панелі зручний і адаптований для роботи з кількома варіантами конфігурації одного товару (див. рис. 3.11).

← На головну ● Переглянути замовлення

Додати новий товар

Назва товару:

Опис:

Склад:

Зображення (фото букета):

Вибрати файл Файл не вибрано

Рисунок 3.11 – Реалізація адміністративної панелі

Коли товар додано, з'явиться повідомлення про успішне завершення (див. рис. 3.12).

Додати новий товар

Товар успішно додано!

Назва товару:

Рисунок 3.12 – Результат успішного додавання товару

Також є кнопка, де адміністратор може побачити замовлення (див. рис. 3.13). Код реалізації сторінки подано в додатку Ж.

← Назад до панелі адміністратора

Список замовлень

ID	Товар	Ім'я	Email	Телефон	Адреса	Оплата	Ціна
4	Букет "Girly Girl" x2	oleg	oleg@gmail.com	0966666684	Самовивіз	Оплата при отриманні	5300.00 грн
3	Букет "Pink mood"	любомир	olesyapetryk@gmail.com	0966936845	Shevchenka 25	Оплата при отриманні	2100.00 грн
2	Букет "Girly Girl"	Олеся Петрик	olesyapetryk@gmail.com	0966936845	Shevchenka 25	Оплата при отриманні	2100.00 грн
1	Букет "Pink mood"	Олеся Петрик	olesyapetryk96@gmail.com	+380966936845	Shevchenka 25	Оплата при отриманні	2100.00 грн

Рисунок 3.13 – Перегляд замовлень

Для зручності користувачів на сторінці каталогу реалізовано можливість сортування товарів за різними критеріями, зокрема за розміром та ціною. Такий функціонал дозволяє швидко знаходити необхідні позиції відповідно до індивідуальних потреб покупця.

У системі реалізовано два основні типи сортування: за зростанням та спаданням ціни. При цьому також враховується вибраний фільтр за розміром (наприклад, XS, S, M, L). Якщо користувач обирає конкретний розмір, до основного SQL-запиту приєднується таблиця `product_sizes`, де виконується фільтрація за відповідним значенням. На рисунку 3.14 зображено приклад фільтрації за обраним розміром XS.

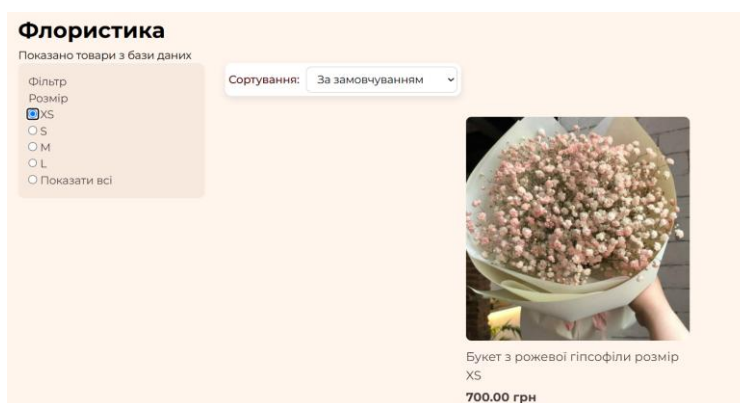


Рисунок 3.14 – Фільтрація товару за розміром

У разі, якщо фільтр розміру не встановлено, система автоматично обирає для кожного товару найменший доступний розмір із відповідною ціною. Це досягається за допомогою підзапиту, який використовує умовну логіку сортування розмірів ($XS < S < M < L$) та функцію `MIN(...)` для вибору мінімального значення.

Сортування за ціною дозволяє користувачеві переглядати товари в порядку зростання або спадання вартості. При виборі сортування за зростанням товари виводяться від найдешевших до найдорожчих, що зручно для користувачів, які шукають бюджетні варіанти. У випадку сортування за першими відображаються найдорожчі товари, що актуально для покупців, які

орієнтуються на преміум-продукцію або шукають найбільш насичені композиції.

На рисунку 3.15 зображено сортування за спаданням ціни.

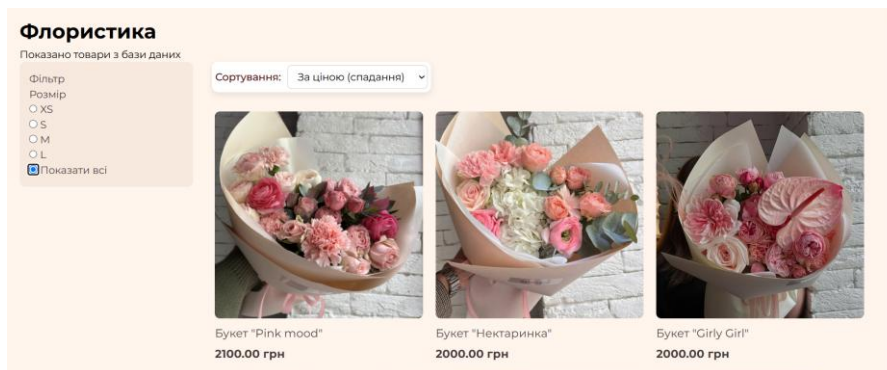


Рисунок 3.15 – Сортування за спаданням ціни

Така можливість надає гнучкість у взаємодії з каталогом та покращує користувацький досвід при виборі товарів.

3.1.2 Розробка бекенд -частини для інтернет-магазину квітів

Наступним етапом є підключення до бази даних, що є основою для взаємодії з інформацією на сервері. Для цього використовуємо функціонал PHP, зокрема об'єкт `mysqli`, який забезпечує зв'язок із MySQL базою даних.

У лістингу 3.5 подано код для підключення бази даних

Лістинг 3.5 – Код для підключення БД

```
$servername = "fdb1029.awardspace.net";
$username = "4532764_bloom";
$password = "olesyaa1020";
$dbname = "4532764_bloom";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Помилка з'єднання: " . $conn->connect_error);
}
```

Один з важливих аспектів бекенду інтернет-магазину – можливість додавати товари до бази даних. Це можна реалізувати через форму, яка дозволяє адміністраторам додавати нові товари з такими полями, як назва, опис, ціна, зображення та категорія. Відповідно до цього, наступним етапом є реалізація функції додавання товарів через форму в адмін-панелі.

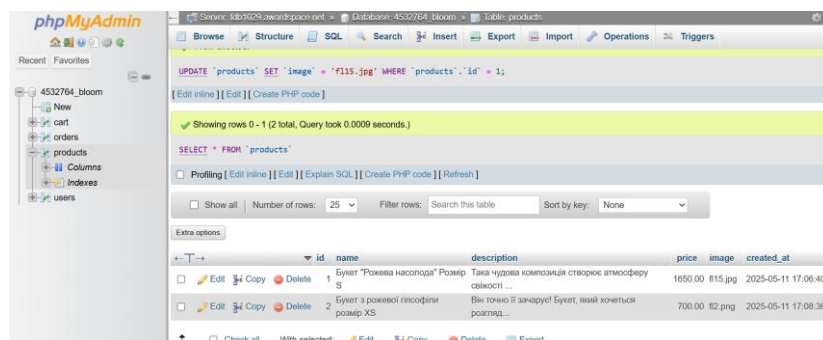
У лістингу 3.6 подано код для додавання товарів

Лістинг 3.6 – Додавання товарів

```
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $name = $_POST['name'] ?? '';
    $description = $_POST['description'] ?? '';
    $composition = $_POST['composition'] ?? '';
    $availability = $_POST['availability'] ?? '';
    $image_name = '';
    $conn->commit();
    $message = "Товар успішно додано!";
    $message_type = 'success';
} catch (Exception $e) {
}

?>
```

Коли адміністратор заповнює форму додавання товару (див. рис. 3.11), дані відправляються на сервер за допомогою методу POST. Спочатку перевіряється, чи була надіслана форма. Якщо це так, збережені значення з полів форми – назва, опис, ціна, категорія та зображення – передаються в базу даних за допомогою SQL-запиту INSERT (див рис.3.16).



id	name	description	price	image	created_at
1	Букет "Романтична нагода" Розмір S	Така чудова композиція створить атмосферу романтики...	1650.00	f115.jpg	2025-05-11 17:06:40
2	Букет з рожевої гірської рози XS	Він точно її закохає! Букет, який хочеться розглядати...	700.00	f2.png	2025-05-11 17:08:36

Рисунок 3.16 – Додані товари у базі даних

Зображення товару завантажується в папку images/ на сервері. Після успішного додавання товару, користувач отримує повідомлення про успіх. Всі додані товари відразу відображаються на головній сторінці інтернет-магазину, що дає можливість користувачам побачити актуальні пропозиції (див. рис. 3.17).

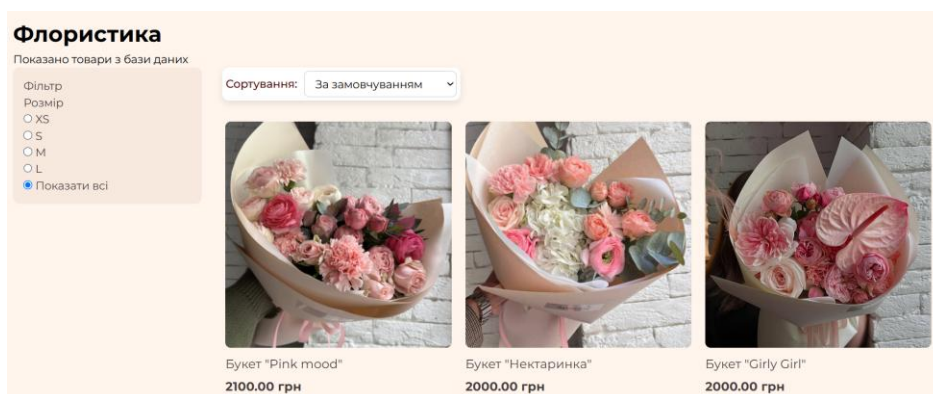


Рисунок 3.17 – Відображення товарів із бази даних

Також важливою частиною сайту є авторизація та реєстрація, тому що після неї користувач отримує більше можливостей. На рисунку 3.18 зображено вікно авторизації, яке з'являється після 10 секунд – це PHP-код, який реалізує авторизацію та реєстрацію користувачів. Користувачі можуть вводити свої дані, які перевіряються у базі даних. У разі успішної перевірки виводиться повідомлення про вхід.

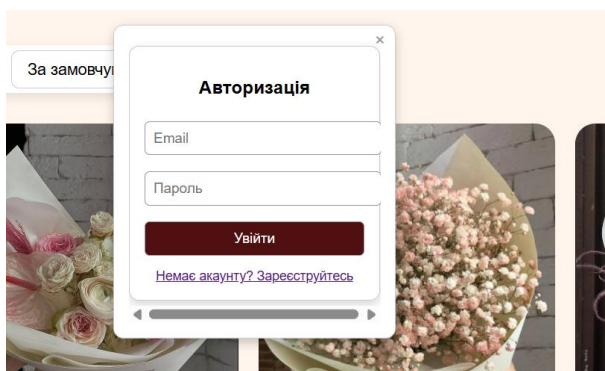


Рисунок 3.18 – Форма авторизації інтернет-магазину

Якщо користувач ще не має облікового запису, йому пропонується створити акаунт шляхом заповнення форми реєстрації, в якій обов'язковими є поля email і пароль.

У лістингу 3.7 подано частину коду реєстрації

Лістинг 3.7 – Частина коду реєстрації

```
$loginError = "Користувача не знайдено!";
    }
    $stmt->close();
} elseif ($_POST['form_type'] === 'register') {
    // Реєстрація

    $stmt = $conn->prepare("INSERT INTO users (email,
password) VALUES (?, ?)");
    $stmt->bind_param("ss", $email, $password);

    if ($stmt->execute()) {
        $registerError = "Реєстрація успішна! Ви можете
увійти.";
    } else {
        $registerError = "Помилка при реєстрації!";
    }
    $stmt->close();
```

Після надсилання форми, скрипт перевіряє, чи вже існує користувач із таким email. Якщо ні – дані зберігаються в таблицю users, де поля email та password фіксують облікові дані нового користувача. У разі успішної реєстрації з'являється повідомлення про можливість увійти в систему.

Авторизація відбувається шляхом перевірки email та пароля, введених користувачем, зі значеннями в базі даних. Якщо облікові дані правильні – користувач авторизується, і його ID або email зберігається в сесії для подальшої взаємодії із системою.

У системі також реалізований повноцінний інтерфейс кошика у вигляді висувної панелі (slide-out), яка з'являється з правого боку сторінки після натискання на іконку кошика.

Коли користувач відкриває кошик, система надсилає AJAX-запит до PHP-скрипта, який отримує з таблиці cart всі товари, що відповідають email або ID

авторизованого користувача. У лістингу 3.8 подано частину реалізації логік кошика.

Лістинг 3.8 – Реалізація кошика

```
<div id="cartItemsContainer" class="flex-grow overflow-y-auto
py-4 space-y-3">
  <p id="emptyCartMsg" class="text-gray-500 text-center mt-
10">Ваш кошик порожній.</p>
</div>
<div class="pt-4 border-t border-gray-300">
  <div class="flex justify-between items-center mb-4">
    <span class="text-lg font-semibold text-gray-
800">Всього:</span>
    <span id="cartTotal" class="text-lg font-semibold
text-gray-800">0.00 грн</span>
  </div>
  <button class="w-full btn-checkout hover:bg-opacity-90
text-white font-bold py-3 px-4 rounded-lg transition-colors
duration-150">
    Оформити замовлення
  </button>
</div>
```

Оформлення замовлення на сайті починається з того, що користувач додає потрібні товари до кошика. Коли кошик не порожній, у нижній його частині з'являється кнопка «Оформити замовлення». Натискання на цю кнопку відкриває форму, де користувач повинен вказати своє ім'я, email, номер телефону, адресу доставки, а також вибрати зручний спосіб оплати. Після заповнення форми і підтвердження замовлення дані надсилаються на сервер, де обробляються у PHP-скрипті. На сервері спочатку ініціюється транзакція, щоб забезпечити цілісність операції. В базу даних у таблицю `orders` записується основна інформація про замовлення – контактні дані покупця, обрана форма оплати та загальна сума замовлення.

Після успішного створення цього запису отримується унікальний ідентифікатор замовлення (`order_id`), який використовується для подальшого зв'язку з таблицею `order_items`, де зберігаються усі окремі позиції з кошика. Кожна позиція зберігається у вигляді окремого рядка, що містить ідентифікатор

товару, обраний розмір, кількість одиниць і ціну. Коли всі товари додані до бази, транзакція підтверджується, і замовлення вважається успішно створеним.

У лістингу 3.9 подано частину коду із додавання товарів до бази даних при оформленні замовлення.

Лістинг 3.9 – Реалізація оформлення замовлення

```
$conn->begin_transaction();
try {
    $insertOrderSql = "INSERT INTO orders (user_email,
user_phone, user_name, user_address, payment_method, price)
VALUES (?, ?, ?, ?, ?, ?)";
    $stmtOrder = $conn->prepare($insertOrderSql);
    $stmtOrder->bind_param("sssssd", $user_email,
$user_phone, $user_name, $user_address, $payment_method,
$totalOrderAmount); // Use $totalOrderAmount here
    $stmtOrder->execute();
    $order_id = $conn->insert_id;
```

Якщо ж на будь-якому з етапів виникає помилка, транзакція скасовується, щоб уникнути некоректного збереження даних.

Після завершення обробки замовлення кошик очищується, а користувач бачить повідомлення про успішну покупку або потрапляє на відповідну сторінку з підтвердженням. Таким чином, процес оформлення замовлення повністю автоматизований і забезпечує збереження всієї необхідної інформації в базі даних.

3.2 Тестування інтерфейсу сайту інтернет-магазину «BLOOM»

Для перевірки зручності та стабільності використання інтерфейсу сайту інтернет-магазину «BLOOM» було проведено тестування на різних пристроях і платформах. Особливу увагу приділено адаптивності – здатності інтерфейсу автоматично підлаштовуватись під розміри екрана користувача. Сайт було протестовано на стаціонарних комп'ютерах із операційними системами Windows 10 і macOS, а також на мобільних пристроях із Android та IOS. На кожній платформі перевірялась коректність відображення елементів – меню,

карток товарів, кошика, форм реєстрації та оформлення замовлення. Окремо тестувалась робота сайту у різних браузерях, зокрема Google Chrome та Microsoft Edge, щоб впевнитися у кросбраузерній сумісності.

Сторінки коректно масштабуються для різних розширень екранів – від великих моніторів до невеликих мобільних дисплеїв. Також було протестовано функціональність таких ключових компонентів, як кошик, форма замовлення, реєстрація та авторизація, які демонструють стабільну роботу на всіх протестованих платформах.

3.2.1 Тестування адаптивності вебсайту інтернет-магазину

Тестування адаптивності інтерфейсу має велике значення для зручності користувачів, адже сайт інтернет-магазину повинен однаково добре відображатися на будь-якому пристрої – від великого екрана комп'ютера до невеликого смартфона. Це особливо актуально, оскільки значна частина користувачів заходить на сайти саме з мобільних телефонів. Під час перевірки адаптивності було протестовано, як виглядає сайт на різних розмірах екранів: широкоформатних моніторах (1920×1080), планшетах (768×1024) і смартфонах (360×640).

Важливо, що всі елементи інтерфейсу правильно масштабуються, зберігається зручне розташування блоків, шрифти не стають занадто дрібними, а кнопки залишаються доступними для натискання пальцем. Така перевірка гарантує, що користувачам буде комфортно взаємодіяти з інтернет-магазином незалежно від типу пристрою.

На рисунку 3.19 подано частину головної сторінки інтернет-магазину для десктопу, де зображені фото бестселерів в рядок.

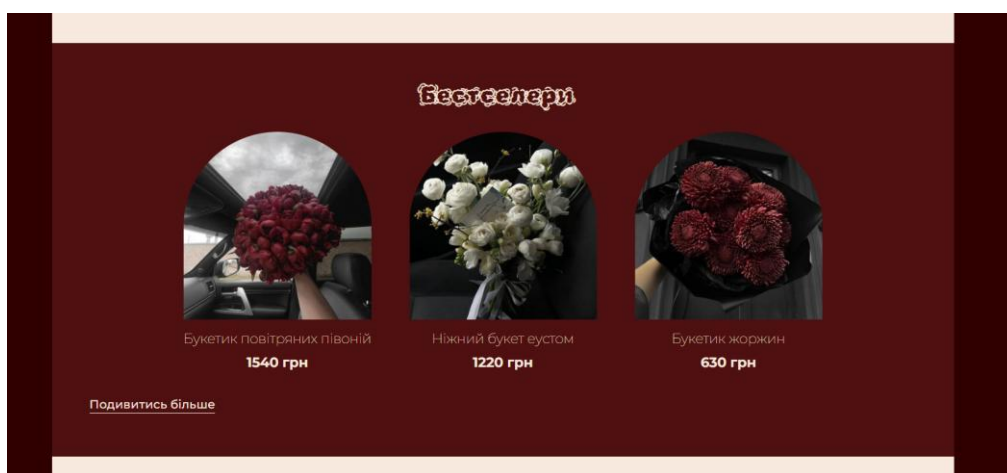


Рисунок 3.19 – Частина «Бестселери» на десктопі

При зменшенні розміру екрану (мобільна версія) відображається тільки одне фото, точніше, з рядка вони переміщуються у стовпець (див. рис. 3.20).

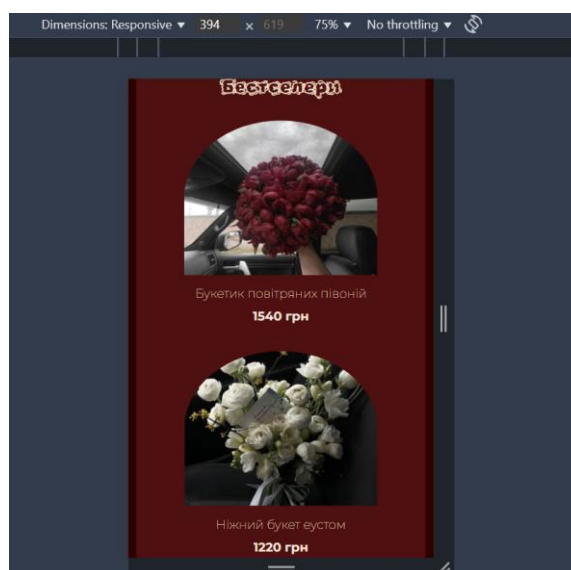
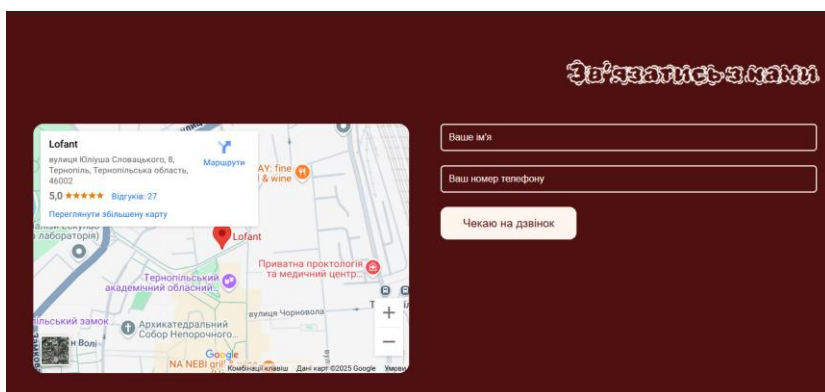
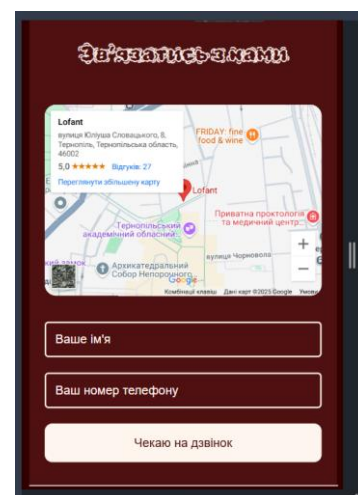


Рисунок 3.20 – Частина сторінки «Бестселери» на мобільній версії

За таким самим принципом реалізована частина сторінки «Контакти». Спочатку в рядку зображена карта зліва, а справа форма для заповнення зв'язатись (див. рис. 3.21. а), а при адаптації карта буде зверху, а форма переноситься вниз (див. рис. 3.21. б).



а



б

Рисунок 3.21 – Частина сторінки «Контакти» десктоп (а), мобільна версія (б)

Також реалізований footer сторінки (див. рис. 3.22). Він вміщає в себе логотип, інформацію про соціальні мережі та інші контакти, а також посилання на сторінки.

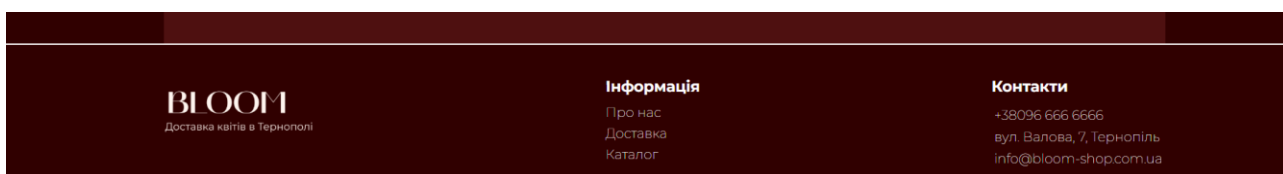


Рисунок 3.22 – Footer сторінки

При зменшенні екрану ця частина сторінки зсувається із рядка в стовпець та розташована по центру. (див. рис. 3.23).

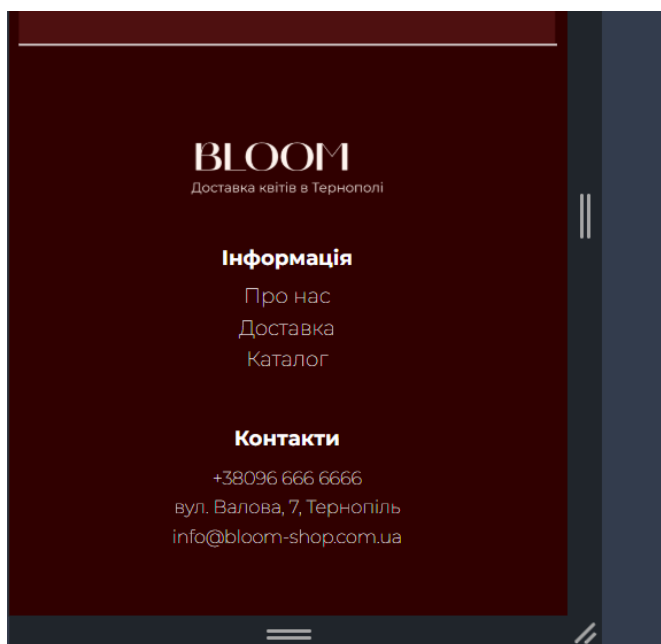


Рисунок 3.23 – Мобільна адаптація footer

Також було реалізовано адаптацію навігаційного меню. На рисунку 3.2 подано меню у десктоп версії. При зменшенні розміру екрану утворюється «бургер-меню», при натисканні чого зліва висувається панель (див. рис. 3.24)

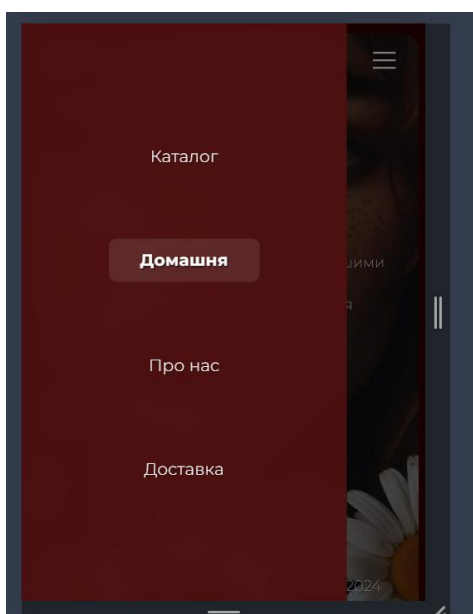


Рисунок 3.24 – Висувне навігаційне меню

На рисунку 3.25 представлено вигляд головної сторінки інтернет-магазину на великому екрані (1920×1080), де чітко видно повноцінне

горизонтальне меню навігації, великий банер та оптимально розміщені елементи товарів. Для забезпечення зручності користувачів на мобільних пристроях, інтерфейс було адаптовано до меншого розміру екрана.

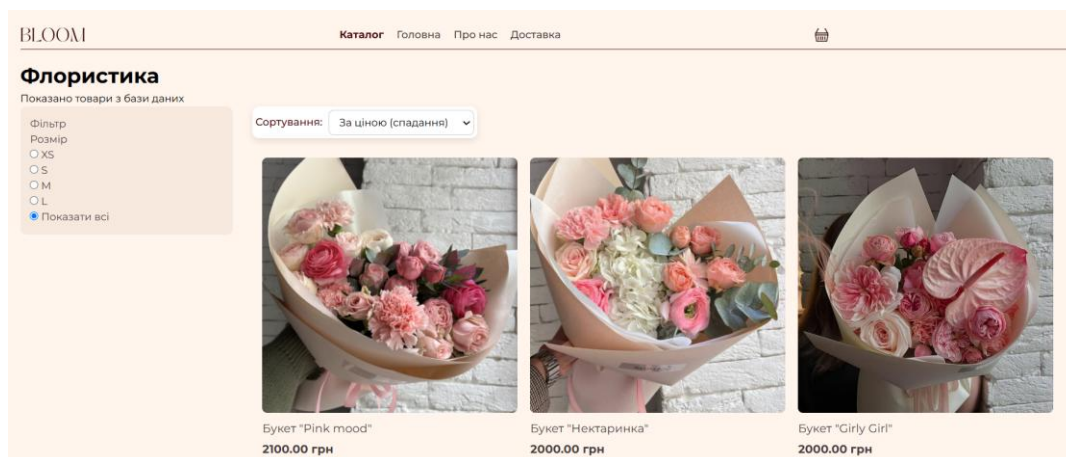


Рисунок 3.25 – Сторінка інтернет-магазину на великому екрані

Адаптований вигляд сторінки для смартфонів (розширення 360×640) наведено на рисунку 3.26. Як видно з прикладу, всі блоки перебудовуються у вертикальний формат, зберігаючи при цьому логіку розміщення контенту, що позитивно впливає на зручність користування сайтом на різних пристроях. Картинки товарів тепер по одній в стовпець.

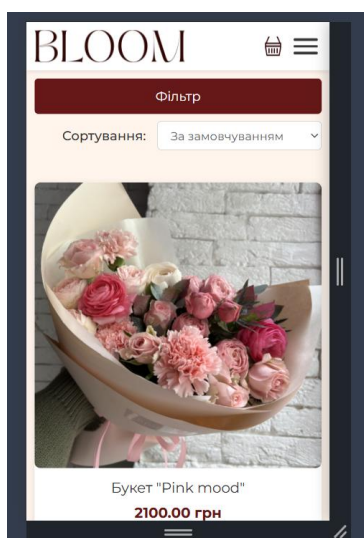


Рисунок 3.26 – Адаптований вигляд сторінки для смартфонів

Зокрема, при зменшенні ширини екрану навігаційне меню автоматично трансформується в так званий «гамбургер»-меню, що дозволяє зберегти простір та забезпечити зручний доступ до навігації (див рис. 3.27).

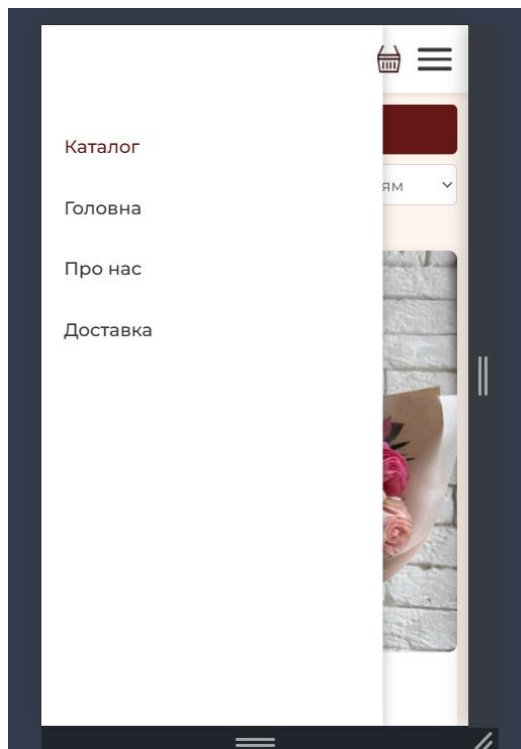


Рисунок 3.27 – Адаптивне навігаційне меню

Крім того, на сайті реалізовано функціонал висувного фільтра, який забезпечує зручну фільтрацію товарів за заданими критеріями. Фільтр розміщується зліва від основного контенту й у звичайному стані залишається прихованим поза межами екрану. При натисканні на кнопку «Фільтри» панель плавно висувається завдяки використанню CSS-анімації та JavaScript -функції, яка керує додаванням та видаленням відповідного класу (див. рис. 3.28).

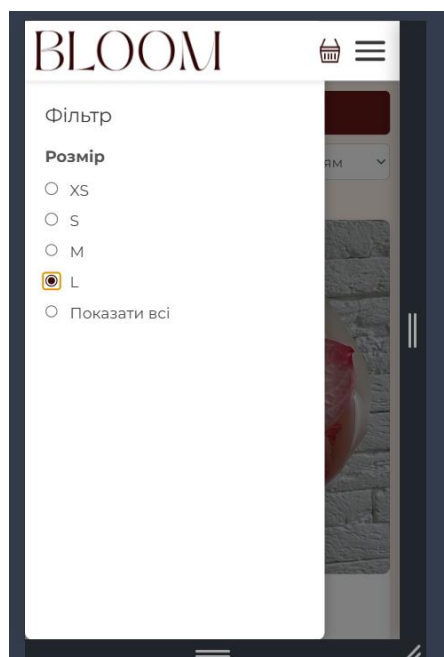


Рисунок 3.28 – Адаптація фільтру

На рисунку 3.29 зображено адаптивний інтерфейс сортування товарів у каталозі квіткового інтернет-магазину. Цей елемент інтерфейсу автоматично підлаштовується під розмір екрана користувача: на мобільних пристроях елемент <select> перетворюється на зручне випадаюче меню з варіантами сортування – «За замовчуванням», «За ціною (зростання)» та «За ціною (спадання)».

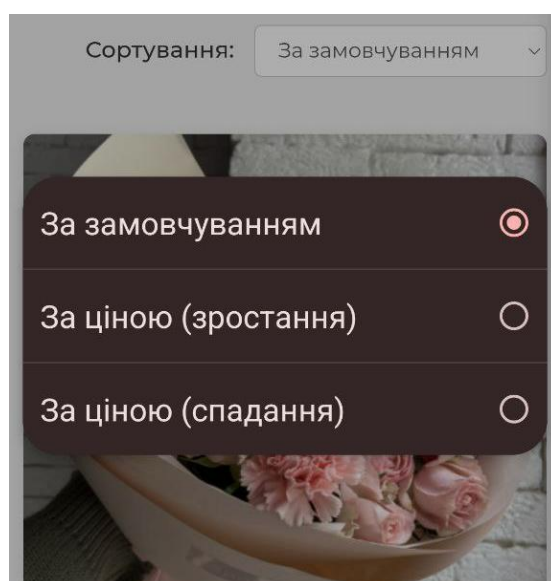


Рисунок 3.29 – Адаптація сортування

При виборі одного з варіантів користувач може миттєво переглянути оновлений список товарів відповідно до обраного критерію без перезавантаження сторінки, що досягається завдяки використанню AJAX.

Окрім адаптації головної сторінки та каталогу, було також реалізовано для сторінки інформації про букет (див. рис. 3.30). На зображенні видно, що при адаптації фото залишається зверху, а весь текст (опис і склад) залишається внизу, в стовпець.



Рисунок 3.30 – Мобільна адаптація сторінки про букет

На рисунку 3.31 зображено друга частина сайту про букет. Де видно як подано кнопки розміру, купити та додати в кошик. Крім того footer сторінки також адаптований в стовпець.

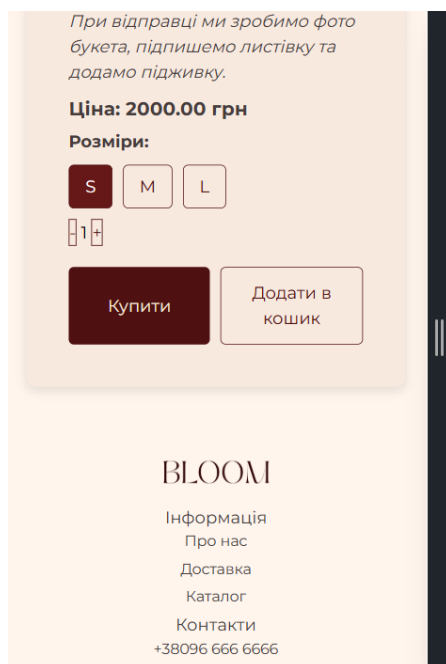
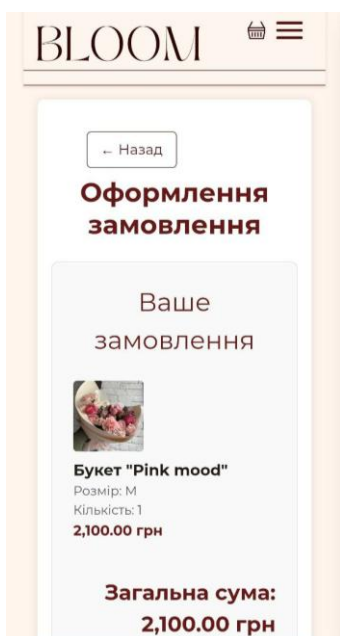
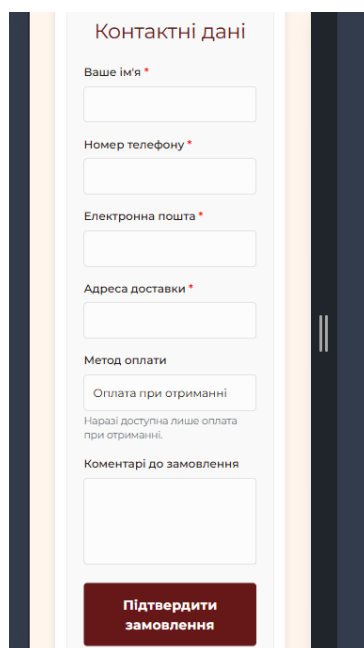


Рисунок 3.31 – Друга частина адаптованої сторінки

На рисунку 3.32, а подана сторінка оформлення замовлення на мобільних екранах. Форма для заповнення інформації щодо замовлення також адаптується і стовпець (див. рис. 3.32, б)



а



б

Рисунок 3.32 – Мобільна адаптація сторінки «оформлення замовлення» (а)
контактні дані (б)

Адаптація забезпечує стабільну та зручну роботу на всіх пристроях, що є невід'ємною частиною якісного користувацького досвіду [59].

3.2.2 Кросплатформність інтерфейсу інтернет-магазину квітів

Інтерфейс інтернет-магазину квітів розроблено з урахуванням кросплатформної сумісності, що забезпечує його коректне відображення та функціонування на різних операційних системах – зокрема, Windows, macOS, а також у популярних браузерях (Google Chrome, Microsoft Edge тощо). Завдяки використанню стандартів HTML, CSS та JavaScript, усі основні елементи, включаючи сортування товарів, фільтри, кнопки та шрифти, виглядають і працюють однаково на різних платформах.

На рисунку 3.33 представлено інтерфейс головної сторінки інтернет-магазину у браузері Microsoft Edge на операційній системі Windows.

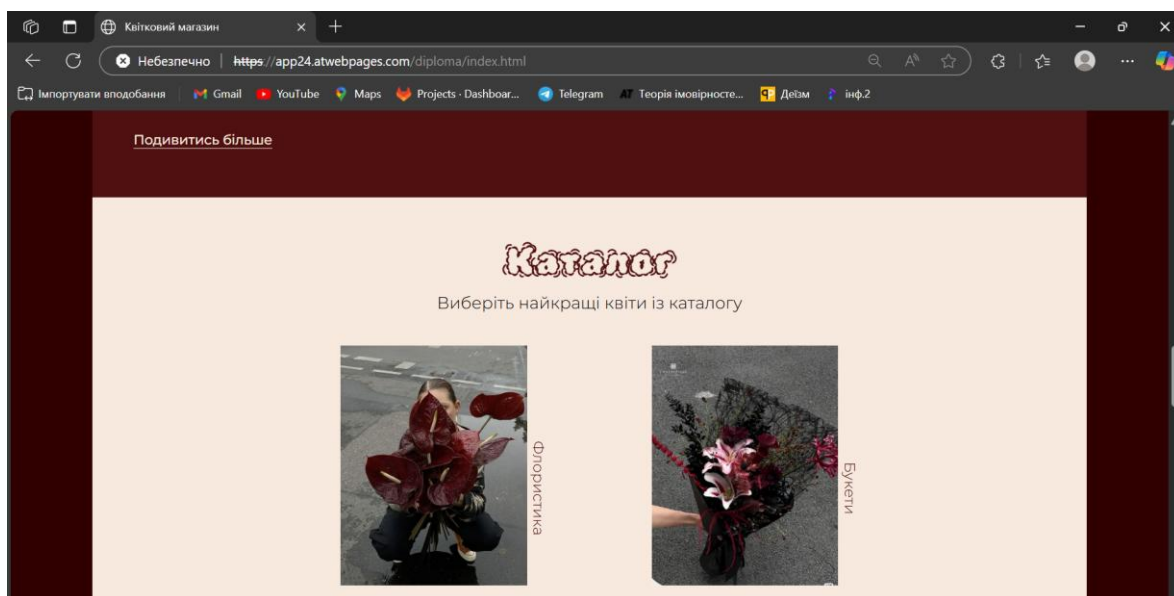


Рисунок 3.33 – Головна сторінка інтернет-магазину у браузері Microsoft Edge

На рисунку 3.34 подано сторінку каталогу у браузері Microsoft Edge.

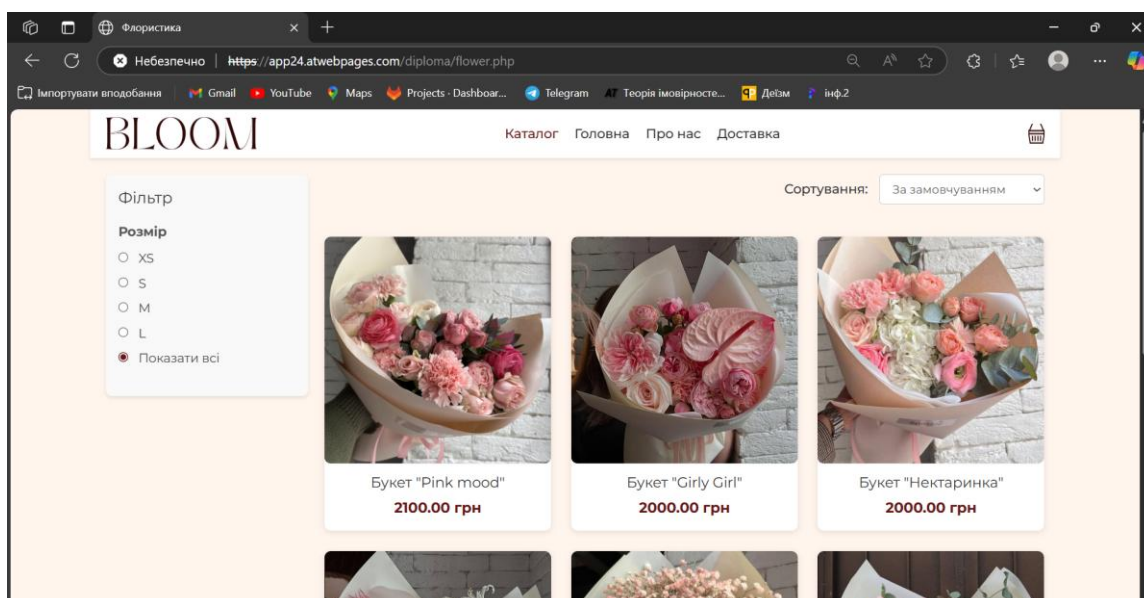


Рисунок 3.34 – «Каталог» інтернет-магазину у браузері Microsoft Edge

На рисунку 3.35 зображено сторінку інформації про букет у браузері Microsoft Edge.

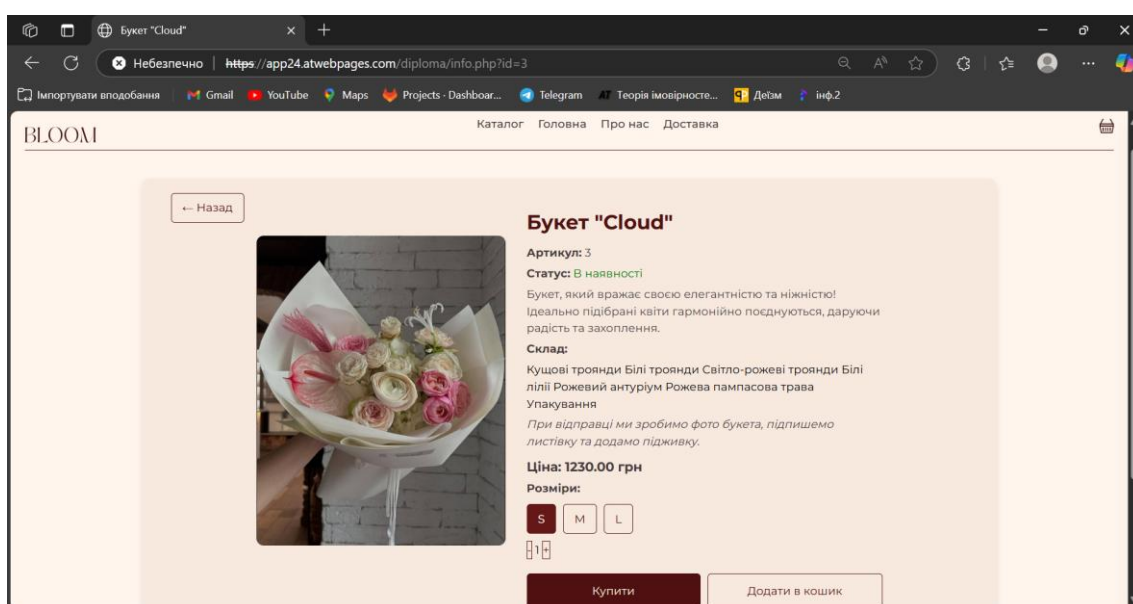


Рисунок 3.35 – Сторінка інформації про букет у браузері Microsoft Edge

На рисунку 3.36 зображено сторінку оформлення замовлення у браузері Microsoft Edge.

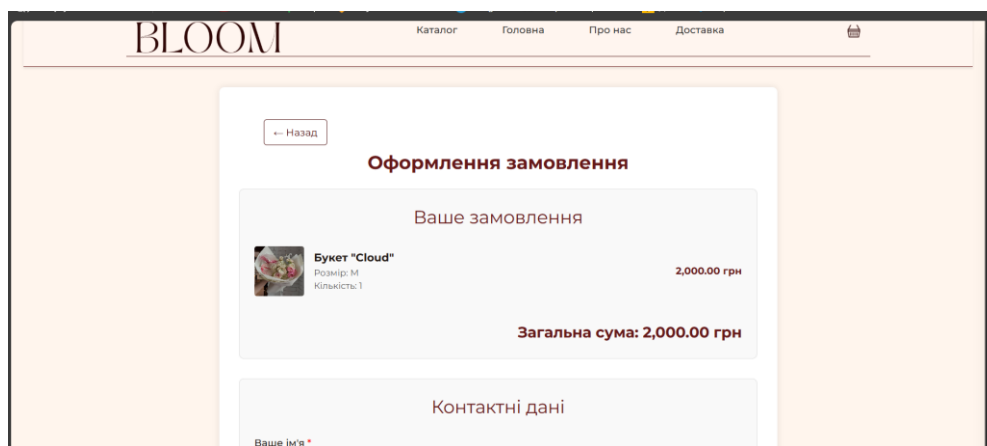


Рисунок 3.36 – «Оформлення замовлення» у браузері Microsoft Edge

Видно, що всі компоненти відображаються коректно, без спотворень або порушення верстки. Інтерактивність елементів збережена: список сортування відкривається плавно, вибрані опції миттєво застосовуються, а сторінка не потребує оновлення, завдяки використанню AJAX-запитів. Водночас деякі візуальні відмінності все ж наявні: наприклад, стилізація випадаючого списку в браузері Microsoft Edge дещо відрізняється від того, як вона реалізована у Google Chrome (див. рис. 3.11). У Chrome елементи форми мають інший відступ і обводку за замовчуванням, що трохи впливає на зовнішній вигляд, але не змінює загальну функціональність.

Також інтерфейс було протестовано у Firefox Developer Edition, який є розширеною версією звичайного Firefox, орієнтованою на потреби веброзробників.

Інтерфейс у відображається подібно до того, як це реалізовано в Microsoft Edge. Зокрема, структура сторінки, розміщення елементів каталогу, кнопки та загальний вигляд компонентів майже ідентичні. Список сортування також відкривається плавно, а обрані параметри застосовуються миттєво без перезавантаження сторінки. Незначні відмінності полягають у стилізації системних елементів, проте з функціонального погляду обидва браузери – Edge і Firefox Developer Edition – забезпечують стабільну, передбачувану та однаково зручну роботу інтернет-магазину (див. рис. 3.37).

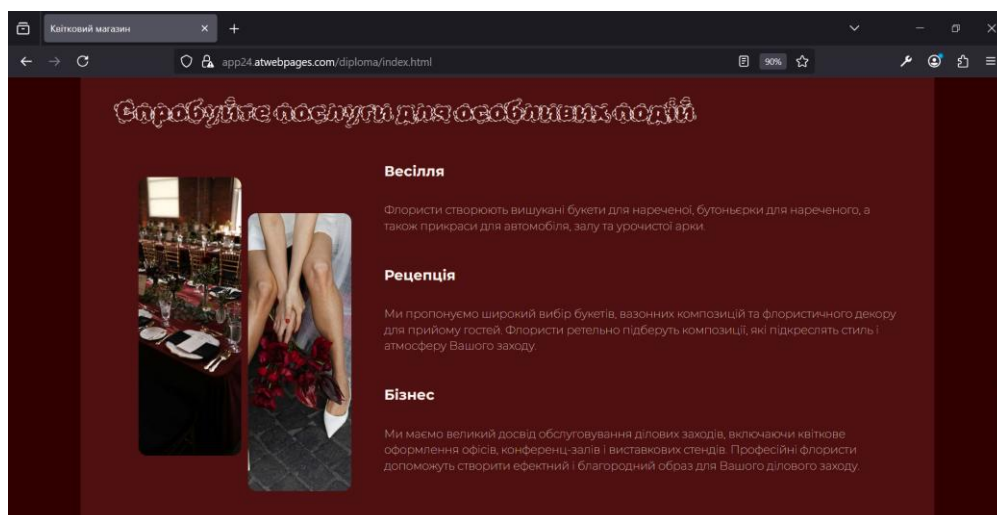


Рисунок 3.37 – Сторінка подій інтернет-магазину у браузері Firefox Developer Edition

На рисунку 3.38 подано сторінку каталогу у браузері Firefox Developer Edition.

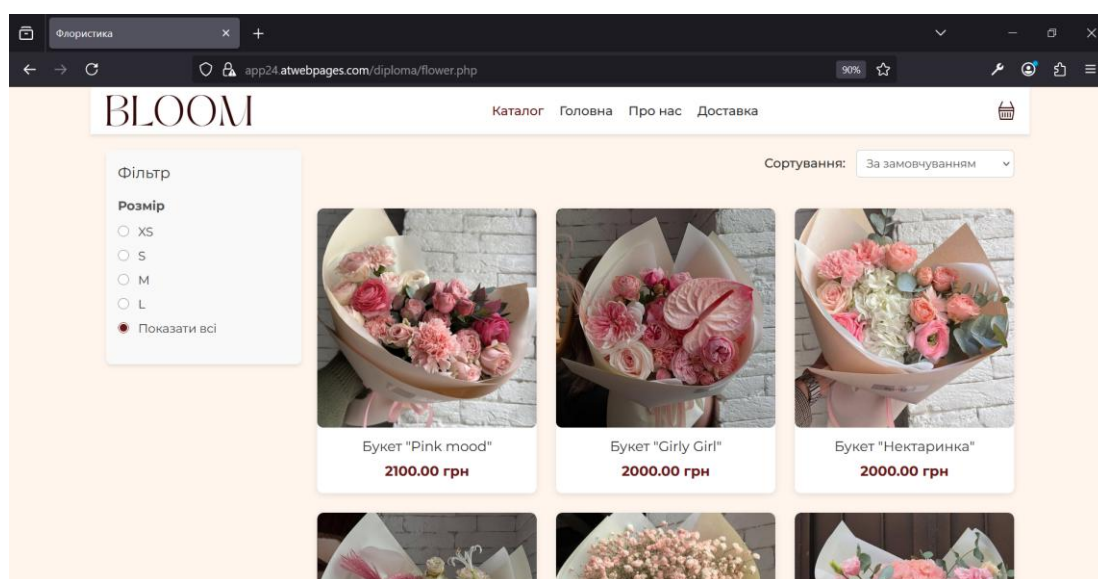


Рисунок 3.38 – Сторінка каталогу інтернет-магазину у браузері Firefox Developer Edition

На рисунку 3.39 зображено сторінку інформації про букет у браузері Firefox Developer Edition.

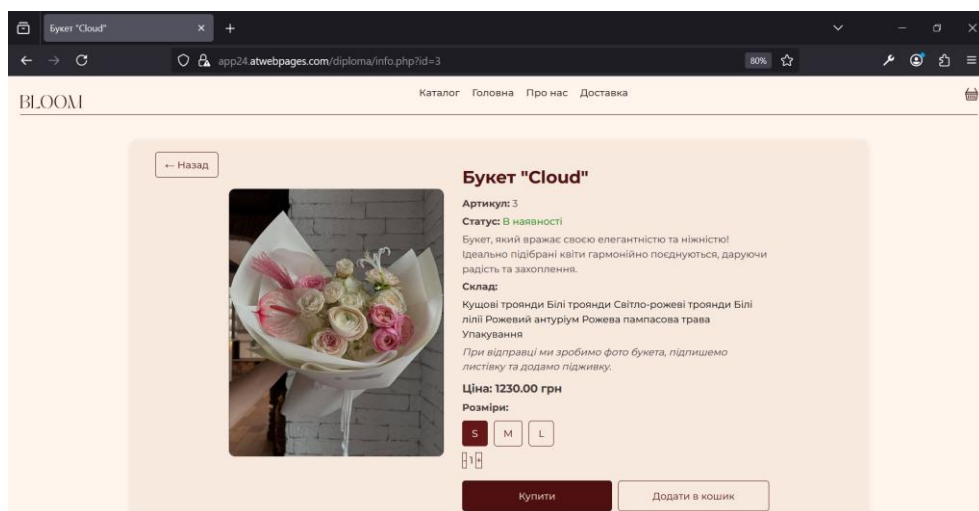


Рисунок 3.39 – Сторінка про букет інтернет-магазину у браузері Firefox Developer Edition

На рисунку 3.40 зображено сторінку оформлення замовлення у браузері Firefox Developer Edition.

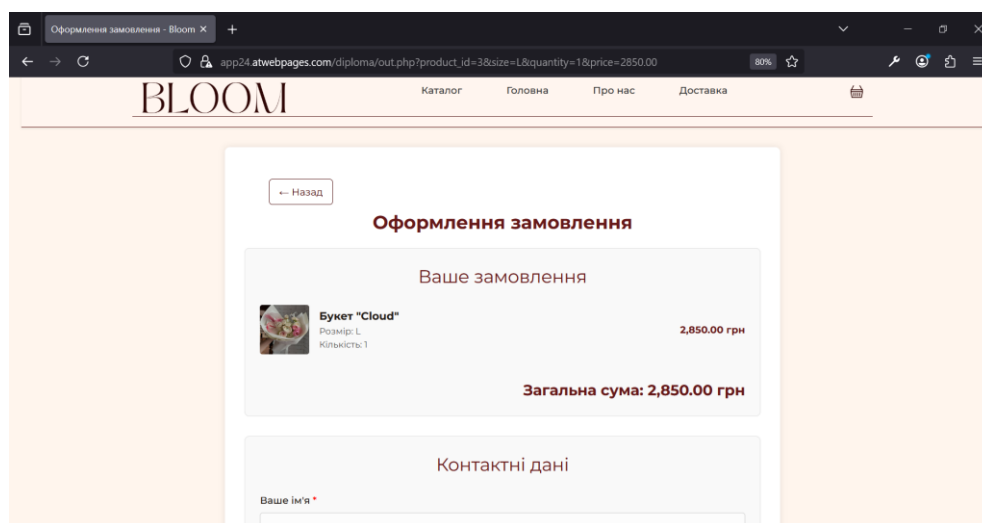


Рисунок 3.40 – «Оформлення замовлення» у браузері Firefox Developer Edition

Тестування у Firefox Developer Edition підтверджує відповідність інтерфейсу сучасним вебстандартам і його готовність до використання в різних середовищах, включно з браузерами, якими активно користуються розробники для перевірки та налагодження вебпроектів.

На рисунку 3.41 подано сторінку каталогу на операційній системі MacOS.

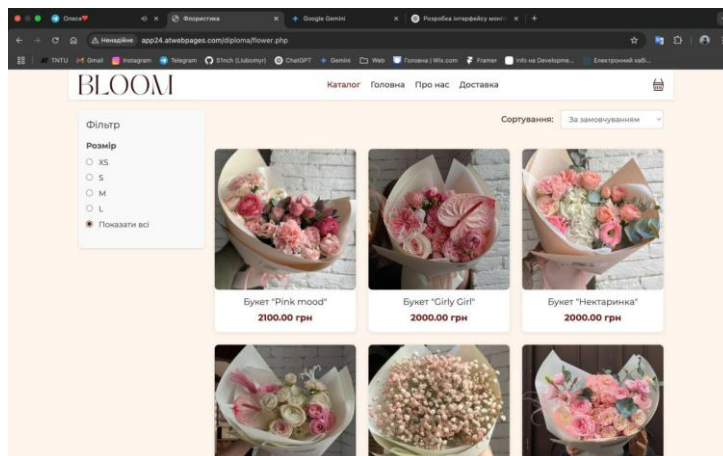


Рисунок 3.41 – Сторінка каталогу інтернет-магазину на MacOS

На рисунку 3.42 зображено сторінку інформації про товар.

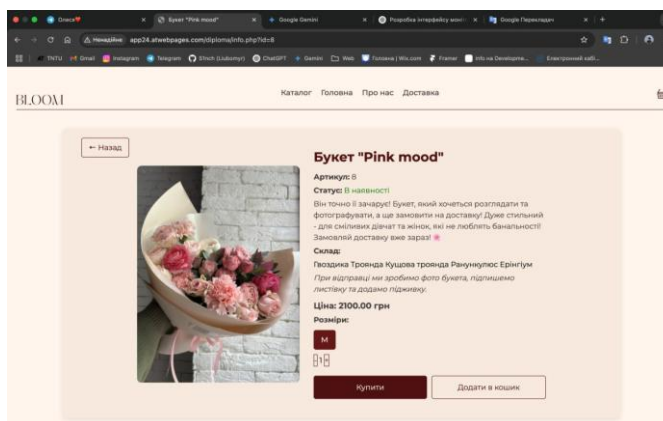


Рисунок 3.42 – Сторінка інформації про товар на MacOS

На рисунку 3.43 зображено сторінку оформлення замовлення.

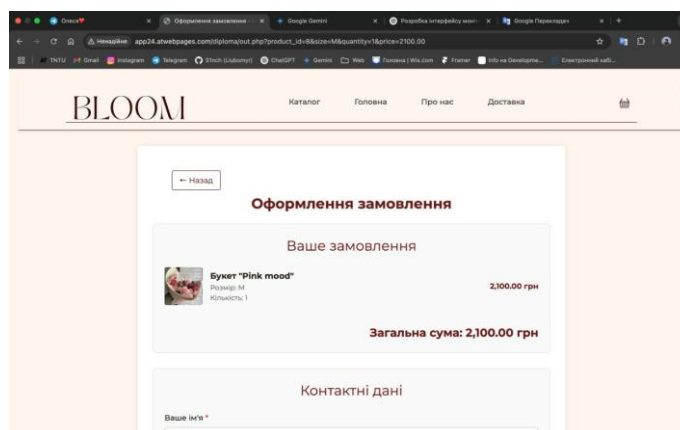


Рисунок 3.43 – Сторінка оформлення замовлення на MacOS

Крім сумісності з різними браузерми, інтерфейс інтернет-магазину показав стабільну роботу на кількох операційних системах, включаючи Windows 10/11, macOS Ventura та Ubuntu 22.04.

Усі елементи, зокрема адаптивна верстка, сортування товарів, фільтри та кнопки, відображаються коректно, незалежно від платформи. Інтерфейс залишається зручним як у класичному десктопному режимі, так і на пристроях із сенсорним керуванням. Завдяки дотриманню вебстандартів та уникненню залежності від специфічних компонентів ОС, сайт підтримує повноцінну взаємодію з користувачем на будь-якому пристрої.

3.3 Висновки до третього розділу

У третьому розділі кваліфікаційної роботи реалізовано практичну частину – створення сайту інтернет-магазину квітів “BLOOM”. Здійснено повну розробку клієнтської (frontend) та серверної (backend) частин сайту з урахуванням сучасних вимог до UI/UX-дизайну, адаптивності та зручності користування.

Розроблено повноцінний інтерфейс магазину з головною сторінкою, каталогом, сторінками товарів, кошиком і формою оформлення замовлення. Реалізовано систему авторизації та реєстрації користувачів, інтеграцію з базою даних, функціонал адміністративної панелі та фільтрацію товарів за параметрами.

Усі основні елементи системи протестовано на функціональність, адаптивність та коректність взаємодії з базою даних.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Вплив діяльності людини на довкілля

Діяльність людини є основним чинником, що спричиняє зміни в природному довкіллі. Це часто призводить до деградації екосистем та виникнення глобальних екологічних проблем [60].

Промислові процеси та виробництво енергії є одними з найзначніших джерел антропогенного впливу на довкілля. Вони є основними забруднювачами, що викидають в атмосферу різноманітні гази та токсини. Виробництво, наприклад, створює нові матеріали, такі як пластмаси, які наразі перевищують за вагою всю живу біомасу на Землі, причому прогнози вказують на те, що до 2050 року в океанах може бути більше пластику, ніж риби [61].

Виробництво фармацевтичних засобів та засобів особистої гігієни призводить до появи речовин, які виявляються у водних об'єктах по всьому світу, створюючи ризики токсичності, стійкості та біоаккумуляції для морського життя [62].

Величезний обсяг та нова хімічна природа промислових викидів (пластмаси, фармацевтичні засоби, важкі метали, радіоактивні ізотопи) вказують на фундаментальну зміну біогеохімічних циклів Землі. На відміну від природних відходів, багато антропогенних забруднювачів не піддаються біологічному розкладу або мають надзвичайно тривалий період напіврозпаду (пластмаси, плутоній з Чорнобиля), означає, що вони накопичуються в навколишньому середовищі та харчових мережах протягом тривалих періодів, створюючи стійкий, міжпоколінний тягар. Це ставить під сумнів лінійну економічну модель «виробництво-використання-утилізація» [63].

Надмірне споживання виникає, коли використання природних ресурсів перевищує здатність екосистеми до самовідновлення. За оцінками Global Footprint Network, поточний попит людства на 70% перевищує швидкість

регенерації екосистем планети, що призводить до деградації довкілля та, зрештою, до виснаження ресурсних баз [64].

У відповідь на ці виклики, дизайнери шукають рішення для зменшення негативного впливу на довкілля. Вони активно працюють над впровадженням принципів циркулярної економіки та сталого дизайну у всіх галузях промисловості [65].

Прикладом застосування принципів кругового дизайну є глобальна торгова платформа Loop, створена компанією з утилізації відходів TerraCycle (США). Loop пропонує популярні продукти від провідних брендів, таких як Nestle, Unilever та PepsiCo, у багаторазових контейнерах. Клієнти замовляють товари онлайн або купують у магазинах, а після використання повертають тару компанії для повторного наповнення [66].

4.2 Естетичне оформлення та ергономічне дослідження робочого місця оператора

Естетичне оформлення та ергономічне дослідження робочого місця оператора є ключовими аспектами для забезпечення комфорту, здоров'я та високої продуктивності праці. Для UI/UX дизайнера або розробника, який проводить значну частину часу перед екраном, це не просто про "красивий" вигляд, а про створення функціонального середовища, яке підтримує його у виконанні завдань, мінімізуючи втому, ризик професійних захворювань та підвищуючи якість розроблюваного продукту.

Загальні гігієнічні вимоги до приміщень для роботи з персональними комп'ютерами (ПК) за ДСТУ ISO 9241-5:2004, включаючи об'ємно-планувальні рішення, площу на одне робоче місце (не менше 6,0 м²), об'єм (не менше 20,0 м³), відстані між столами (не менше 2,5 м у ряду і 1,2 м між рядами), а також вимоги до кольору стін (пастельні тони з коефіцієнтом відбиття 0,5 – 0,6), викладені в ДСанПіН 3.3.2.007-98 та НПАОП 0.00-1.28-10. Розміщення робочих місць у підвальних приміщеннях заборонено [67].

Конструкцією робочого місця має бути забезпечено виконання трудових операцій у межах зони досяжності моторного поля. У таблиці 4.1 подано основні ергономічні параметри робочого столу та зон досяжності.

Таблиця 4.1 – Основні ергономічні параметри робочого столу та зон досяжності [68]

Параметр робочого місця	Рекомендоване значення
Висота робочої поверхні	Регулюється в межах 680-800 мм
Ширина та глибина столу	600-1400 мм, 800-1000 мм
Простір для ніг	Не менше 600 мм заввишки, 500 мм завширшки, 450 мм в глибину (на рівні колін) та 650 мм (на рівні простягнутої ноги)

Оптимальне розміщення монітора, клавіатури та миші є фундаментальним для ергономіки робочого місця відповідно до ДСТУ 8302:2015: оптимальна відстань від очей користувача: 600...700 мм, але не ближче 600 мм. Розташування екрана: висота від підлоги до нижнього краю екрана 95-100 см. Кут зору працюючого щодо екрана: 10-20°, не більше 40°. Екран повинен мати покриття антивідблиску [68].

Щодо естетичного оформлення – в оформленні об'єкта дослідження відсутні елементи стилізації, кольорове зонування, сучасні матеріали або декоративні елементи. Це знижує візуальний комфорт і може негативно впливати на психоемоційний стан оператора.

Відповідно до ДСТУ 7298:2013, показники естетичної якості включають: Декоративну та колірно-фактурну виразність, що стосується вибору кольорів, текстур та загального візуального стилю (використання пастельних тонів для стін, як це рекомендовано в ДСанПіН 3.3.2.007-98). Технологічну доцільність форми: кожен елемент робочого місця повинен бути не тільки привабливим, але й функціональним [69].

Оптимальне робоче місце для UI/UX дизайнера є результатом комплексного підходу, що поєднує строгі ергономічні вимоги з продуманим естетичним оформленням.

4.3 Вплив кольору на покращення умов праці та підвищення продуктивності виробництва

Колір є потужним інструментом, що виходить за межі простої естетики, прямо впливаючи на психологію людини. Він відіграє ключову роль у формуванні нашого настрою, впливі на рішення та навіть зміні рівня енергії.¹ Сприйняття кольору відбувається у зоровій корі мозку, що впливає на емоції, частоту серцевих скорочень і навіть рівень гормонів [70].

Окрім стандартів безпеки, в Україні також діє ДСТУ 4513:2006. Цей стандарт встановлює загальні вимоги до порядку розроблення, метрологічного атестування, затвердження та застосування стандартних зразків кольору.

Червоний: універсально позначає заборону, небезпеку та служить сигналом тривоги. Коефіцієнт яскравості β для звичайних матеріалів має бути більший ніж 0.07. Жовтий та помаранчевий: ці кольори призначені для попереджень та вказівки на можливі небезпеки або необхідність обережності. Коефіцієнт яскравості β для звичайних матеріалів має бути більший ніж 0.45. Синій: передає обов'язкові дії, інструкції або необхідність використання індивідуальних засобів захисту. Коефіцієнт яскравості β для звичайних матеріалів має бути більший ніж 0.05. Зелений: вказує на безпеку, безпечні умови, шляхи евакуації, місця надання першої допомоги або відсутність небезпеки. Коефіцієнт яскравості β для звичайних матеріалів має бути більший ніж 0.12. Білий: переважно використовується для фонів та символів для забезпечення контрасту та чіткості. Коефіцієнт яскравості β для звичайних матеріалів має бути більший ніж 0.75. Чорний: Служить контрастним кольором, особливо для жовтих та помаранчевих фонів, а також

для графічних символів. Коефіцієнт яскравості β для звичайних матеріалів має бути менший ніж 0.03 [71].

Для інтернет-магазину квітів «Bloom» вибір палітри має особливе значення, оскільки вона повинна передавати свіжість, красу та емоції, пов'язані з квітами, водночас забезпечуючи легкість використання та спонукаючи до покупки. Використання вишневого, коричневого, бежевого та білого кольорів для «Bloom» створює витончену, теплу та водночас сучасну естетику.

У Таблиці 4.2 подано детальний розбір обраних кольорів, їх психологічні асоціації, застосування в UI/UX дизайні та прогнозований вплив на користувача й ефективність сайту.

Таблиця 4.2 – Аналіз кольорів до вебсайту «Bloom»

Колір	Психологічні асоціації	Застосування в UI/UX для «Bloom»	Вплив на продуктивність
Вишневий	Емоції. Насичені відтінки квітів.	Акцентний колір для привернення уваги	Привертає увагу до ключових дій, спонукає до покупок.
Коричневий	Натуральність, затишок, тепло.	Для другорядного тексту	Створює відчуття стабільності
Бежевий	Спокій, нейтральність, мінімалізм.	Основний фон для сторінок.	зменшує напругу, спокій.
Білий	Чистота, простота, свіжість, простір	Для простору навколо елементів.	Забезпечує чіткість.

Для збільшення продажів (конверсії), колір кнопок «Купити» дуже важливий. Деякі дослідження показують, що червоні кнопки можуть збільшити кількість покупок більше, ніж зелені [72].

На рисунку 4.1 подано приклад червоної кнопки на сайті інтернет-магазину «Bloom».

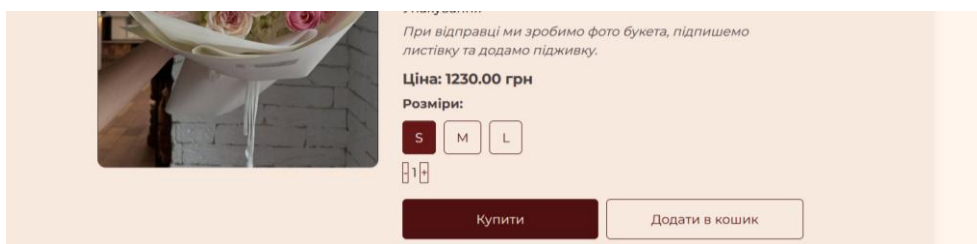


Рисунок 4.1 – Приклад кнопки червоного відтінку на сайті інтернет-магазину «Bloom»

Такі інноваційні гіганти, як Google та Airbnb, демонструють ефективне використання психології кольору. Вони оформлюють свої офіси яскравими палітрами та вільними просторами, тим самим заохочуючи креативність, співпрацю, підвищуючи задоволеність працівників та зміцнюючи позитивний образ компанії [73].

Колір є причиною до 85% рішень про покупку товару. Крім того, кольорова реклама в журналах впізнається на 26% частіше, ніж чорно-біла. Наприклад, Performable зміг збільшити свій коефіцієнт кліків на 21%, просто змінивши колір кнопки заклик до дії із зеленого на червоний. Це показує, як сильно колір впливає на сприйняття та поведінку [74].

4.4 Висновок до четвертого розділу

У четвертому розділі було розглянуто питання впливу діяльності дизайнерів на умови праці в процесі створення інтернет-магазину.

Проаналізовано естетичні та ергономічні особливості організації робочого місця, та запропоновано кольорову палітру інтерфейсу для покращення на психоемоційний стан спрямований на ефективність праці.

ВИСНОВКИ

У кваліфікаційній роботі було успішно вирішено поставлені завдання щодо розробки та впровадження UI/UX дизайну для інтернет-магазину «Bloom».

В першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- Проаналізовано сучасний стан досліджень у сфері UI/UX дизайну вебзастосунків, вивчено актуальні дизайнерські рішення, принципи зручності, адаптивності та візуальної привабливості користувацького інтерфейсу, а також їхній потенційний вплив на позиції сайту в пошукових системах.

- Вивчено інтерфейси існуючих інтернет-магазинів для виділення їхніх сильних та слабких сторін, що стало фундаментом для проектування «Bloom».

- Висвітлено ключові задачі для реалізації дипломного проєкту, включаючи вибір інструментів та технологій, а також ідентифікацію основних користувачів.

- Побудовано варіанти використання (use case), що описують взаємодію акторів із ключовими функціями вебсайту.

В другому розділі кваліфікаційної роботи:

- Досліджено та розроблено детальну структуру та архітектуру інтернет-магазину, забезпечуючи логічну організацію контенту та функціоналу.

- Обґрунтовано підходи до проектування бази даних, включаючи структуру таблиць, зв'язки та типи даних, для забезпечення збереження й обробки інформації.

- Сформовано логіку клієнт-серверної взаємодії, що описує механізм обміну даними між користувацьким інтерфейсом та серверною частиною.

- В третьому розділі кваліфікаційної роботи:

- Розроблено та реалізовано клієнтську та серверну частини інтернет-магазину «Bloom» з використанням сучасних вебтехнологій, створивши функціональний та інтерактивний ресурс.

- Запропоновано та впроваджено удосконалений UI/UX дизайн, спрямований на підвищення якості послуг, зручності та привабливості інтерфейсу.

- Протестовано функціональність та адаптивність розробленого ресурсу, перевіривши коректність роботи всіх ключових модулів: реєстрації користувачів, кошика, оформлення замовлення та відображення товарів.

- Підтверджено можливість створення функціонального інтерфейсу із застосуванням сучасних технологій без необхідності використання складних фреймворків, що робить розроблений продукт придатним для впровадження як комерційний проект або базовий шаблон для подібних вебзастосунків.

У розділі «Безпека життєдіяльності, основи охорони праці»:

- Висвітлено ключові аспекти ергономіки робочого місця, включаючи вплив освітлення та колірної гами на комфорт, здоров'я та продуктивність працівників.

- Проаналізовано психологічний вплив кольору в UI/UX дизайні, підкресливши його значення для візуального комфорту користувача, читабельності та доступності інтерфейсу.

- Обґрунтовано, як свідомий вибір кольорів сприяє зниженню втоми та підвищенню ефективності взаємодії як для розробників, так і для кінцевих користувачів, демонструючи, що колір є потужним інструментом у досягненні бізнес-цілей.

ПЕРЕЛІК ДЖЕРЕЛ

1. The digital transformation of business. Towards the datafication of the relationship with customers. / C. Fernandez-Rovira et al. Technological Forecasting and Social Change. 2021. Vol. 162. URL: <https://doi.org/10.1016/j.techfore.2020.120339>
2. Use of Online Shop Chatbots: How Trust in Seller Moderates Brand Preference and Purchase Intention. Springer Proceedings in Business and Economics / M. Illescas-Manzano et al. Cham: Springer Nature Switzerland, 2024. С. 151–171.
3. Переваги та недоліки Інтернет-Торгівлі [Електронний ресурс]. Режим доступу до ресурсу: https://pidru4niki.com/74049/marketing/perevagi_nedoliki_internet-_torgivli (дата звернення: 12.11.2024).
4. Чому створення інтернет-магазину це один з кращих методів розширення бізнесу [Електронний ресурс]. Режим доступу до ресурсу: https://lb.ua/tech/2023/10/23/579979_chomu_stvorenniya_internetmagazinu_tse.html (дата звернення: 13.11.2024).
5. Квітковий магазин «Sparkle Ternopil» [Електронний ресурс]. Режим доступу до ресурсу: <https://sparkle-shop.com.ua/?srsltid=AfmBOorSDltc4YxsSjC22MQm91DH983ysgkQqS0cYBsYBkzs3JBh5BU> (дата звернення: 13.11.2024).
6. Магазин квітів «Flora de luxe» [Електронний ресурс]. Режим доступу до ресурсу: https://www.instagram.com/floradeluxe_ternopil/ (дата звернення: 13.11.2024).
7. Магазин квітів «Million Flowers» [Електронний ресурс]. Режим доступу до ресурсу: <https://www.instagram.com/million.flowers.ternopil?igsh=MTc5aXhwdW02NTZnOQ%3D%3D> (дата звернення: 14.11.2024).
8. Магазин квітів «Floart» [Електронний ресурс]. Режим доступу до ресурсу: https://www.instagram.com/florart_ternopil/ (дата звернення: 15.11.2024).

9. Умови завдання та вимоги [Електронний ресурс]. Режим доступу до ресурсу: <https://itconsultant.com.ua/umovy-zavdannia-ta-vymohy-do-stvorennia-saytu> (дата звернення: 18.11.2024).

10. Види тестування сайтів [Електронний ресурс]. Режим доступу до ресурсу: <https://webtune.com.ua/statti/web-rozrobka/vydy-testuvannya-sajtiv/> (дата звернення: 19.12.2024).

11. Як UI/UX дизайн покращує позицію сайту в пошукових системах [Електронний ресурс]. Режим доступу до ресурсу: <https://www.seobanda.com/uk/blog/design-and-seo/> (дата звернення: 22.11.2024).

12. UI/UX Design of Web-Based for Outcome-based Education (OBE) Assessment Information System in Universitas Maritim Raja Ali Haji (UMRAH) Using Prototype Method / M. Pertiwi et al. BIO Web of Conferences, eds.: G. Jameson et al. Les Ulis : EDP Sciences, 2024. P. 134.

13. Статистика актуальності UI/UX у різних сферах на базі SCOPUS [Електронний ресурс]. Режим доступу до ресурсу: <https://www.scopus.com/term/analyzer.uri?sort=plf-f&src=s&sid=21aa0c025229b79857874fff9f8b2681&sot=a&sdt=a&sl=13&s=what+is+ui%2fux&origin=resultslist&count=10&analyzeResults=Analyze+results> (дата звернення: 22.11.2024).

14. Статистика щорічної актуальності UI/UX на базі SCOPUS [Електронний ресурс]. Режим доступу до ресурсу: <https://www.scopus.com/term/analyzer.uri?sort=plf-f&src=s&sid=21aa0c025229b79857874fff9f8b2681&sot=a&sdt=a&sl=13&s=what+is+ui%2fux&origin=resultslist&count=10&analyzeResults=Analyze+results> (дата звернення: 28.11.2024).

15. User Interface Redesign of E-Commerce Platform Mobile Application (Kudo) Through User Experience Evaluation to Increase User Attraction / E. Muslim та ін. IOP Conference Series: Materials Science and Engineering. Bristol : IOP Publishing, 2019. С. 508. URL: <https://doi.org/10.1088/1757-899x/508/1/012113>.

16. ЩО РОБИТЬ ДИЗАЙНЕР UI UX: обов'язки, зарплата, курс і відмінності [Електронний ресурс] Режим доступу до ресурсу: <https://businessyield.com/uk/%D0%BA%D0%B0%D1%80%27%D1%94%D1%80%D0%B0/ui-ux->

%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD%D0%B5%D1%80/

(дата звернення: 03.12.2024).

17. UX / UI design in 2021: How to create a comfortable website that brings profit [Електронний ресурс]. Режим доступу до ресурсу: <https://gl.ua/en/blog/ux-ui-design-2021-how-create-comfortable-website-brings-profit>

(дата звернення: 06.12.2024).

18. Unlocking the Power of UX Design for Ecommerce Businesses. [Електронний ресурс]. Режим доступу до ресурсу: <https://www.ecwid.com/blog/ux-design-for-ecommerce-businesses.html> (дата звернення: 09.12.2024).

19. UI vs. UX Design: What's the Difference?. [Електронний ресурс]. Режим доступу до ресурсу: <https://www.coursera.org/articles/ui-vs-ux-design> (дата звернення: 12.12.2024).

20. Have you ever wondered about the difference between UX (User Experience) and UI (User Interface)? - Ali Hassan. [Електронний ресурс]. Режим доступу до ресурсу: https://www.linkedin.com/posts/ali-hassan-tech_ux-vs-ui-activity-7279764680850526208-WYvG?utm_source=share&utm_medium=member_desktop (дата звернення: 13.12.2024).

21. Thooyib B., Zulfikar D. H. Desain UI/UX Website Referral untuk Program Gerakan Funding Culture Menggunakan Figma. Prosiding Seminar Nasional Teknologi Komputer dan Sains. 2023. T. 1. C. 232–241.

22. The Difference Between UX and UI Design: A Beginner's Guide. [Електронний ресурс]. Режим доступу до ресурсу: <https://careerfoundry.com/en/blog/ux-design/the-difference-between-ux-and-ui-design-a-laymans-guide/> (дата звернення: 18.12.2024).

23. Palka O., Kunanets N., Pasichnyk V., Matsiuk O. and Matsiuk S. Comparative Analysis of Smart City Platforms. Proceedings of the COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems (Kharkiv, Ukraine, April 20–21, 2023). CEUR Workshop Proceedings (CEUR-WS.org). 2023. Vol-3403. pp. 487-499.

24. B2B vs B2C - What can you learn from a badly designed toy?. [Электронный ресурс]. Режим доступа до ресурсу: <https://www.abilitymatrix.com/post/b2b-vs-b2c-what-can-you-learn-from-a-badly-designed-toy> (дата звернення: 20.12.2024).

25. UI/UX Design Process Stages and Deliverables Checklist. [Электронный ресурс]. Режим доступа до ресурсу: <https://mobidev.biz/blog/ui-ux-design-stages-deliverables-checklist> (дата звернення: 24.12.2024).

26. Muthia N. F., Setiawan A., Sukmasetya P. UI/UX Design of Plant Seedling Sales and Tracking System Using Design Thinking Method. E3S Web of Conferences. EDP Sciences, 2024. С. 500.

27. UI and UX are two distinct disciplines with remarkably similar goals. [Электронный ресурс]. Режим доступа до ресурсу: <https://webflow.com/blog/ux-vs-ui-design> (дата звернення: 25.12.2024).

28. Jampur, F. X. A. P., Santjojo D. J. Journal of Public Administration and Policy. 2022. Т. 14. С. 256.

29. Hidayanti P., Handayani R., Rifai B. UI/UX Design of Online Tickets for Situ Pasir Maung Tourism in Dago Village Using the Figma Application. Sinkron. 2023. Т. 8.

30. What is Figma?. [Электронный ресурс]. Режим доступа до ресурсу: <https://www.nobledesktop.com/learn/figma/what-is-figma> (дата звернення: 27.12.2024).

31. What Is Canva And How Does It Work? [Электронный ресурс]. Режим доступа до ресурсу: <https://www.techlearning.com/how-to/what-is-canva-and-how-does-it-work-for-education> (дата звернення: 28.12.2024).

32. 24. EXPLORING THE IMPACT OF LINEAR & NON-LINEAR PRESENTATION METHODS IN A DESIGN HISTORY COURSE. / Howell, B. F., Hemming, A. L., Kilbourn-Barber, G., & Christensen, S. Y. ін. Proceedings of the International Conference on Engineering and Product Design Education, EPDE 2023. The Design Society, 2023.

33. What is Visual Studio Code? Microsoft's extensible code editor [Електронний ресурс]. Режим доступу до ресурсу: <https://www.infoworld.com/article/2335960/what-is-visual-studio-code-microsofts-extensible-code-editor.html> (дата звернення: 29.12.2024).

34. What is Notepad++? [Електронний ресурс]. Режим доступу до ресурсу: <https://www.computerhope.com/jargon/n/notepad-plus-plus.htm> (дата звернення: 29.12.2024).

35. The Sublime Text Code Editor – An In-Depth Review [Електронний ресурс]. Режим доступу до ресурсу: <https://www.elegantthemes.com/blog/resources/the-sublime-text-code-editor-an-in-depth-review> (дата звернення: 29.12.2024).

36. Compare Sublime Text, Visual Studio and Notepad++ [Електронний ресурс]. Режим доступу до ресурсу: <https://www.g2.com/compare/sublime-text-vs-visual-studio-code-vs-notepad?source=search> (дата звернення: 29.12.2024).

37. GitHub Copilot in VS Code [Електронний ресурс]. Режим доступу до ресурсу: <https://code.visualstudio.com/docs/copilot/overview> (дата звернення: 03.01.2025).

38. About Netlify [Електронний ресурс]. Режим доступу до ресурсу: <https://www.Netlify.com/about/> (дата звернення: 07.01.2025).

39. Етапи створення сайту [Електронний ресурс]. Режим доступу до ресурсу: <https://webtune.com.ua/statti/web-rozrobka/etapy-stvorenniya-veb-sajtiv/> (дата звернення: 07.01.2025).

40. Особливості платформи Netlify [Електронний ресурс]. Режим доступу до ресурсу: <https://foxminded.ua/netlify-shcho-tse/> (дата звернення: 07.01.2025).

41. What is AwardSpace, Free Hosting no ads popular web hosting companies? URL <https://www.mindphp.com/en/meaning-phase/8750-what-is-awardspace-en.html> (дата звернення: 10.01.2025).

42. Forms setup [Електронний ресурс]. Режим доступу до ресурсу: <https://docs.Netlify.com/forms/setup/> (дата звернення: 10.01.2025).

43. Free Web Hosting Services: A Comprehensive Comparison. [Електронний ресурс]. Режим доступу до ресурсу: <https://medium.com/@KanakSengar/free-web-hosting-services-a-comprehensive-comparison-d770de3dffff> (дата звернення: 13.01.2025).

44. Бізнес-процеси інтернет-магазину та їх оптимізація [Електронний ресурс]. Режим доступу до ресурсу: <https://wezom.com.ua/ua/blog/biznes-processy-internet-magazina-i-ih-optimizaciya> (дата звернення: 13.01.2025).

45. Що таке архітектура програмного забезпечення [Електронний ресурс]. Режим доступу до ресурсу: <https://soandso.biz/blog/softwareengineering/arhitektura-programnogo-zabezpechennya.html> (дата звернення: 16.01.2025).

46. Палка О. та Дмитроца Л. Аналіз мікросервісної архітектури розумного міста. Вісник Хмельницького національного університету. Серія: «Технічні науки». 2023. №6(329). ст. 98-102.

47. Using a Three-Tier Architecture Model [Електронний ресурс]. Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/windows/win32/cosdk/using-a-three-tier-architecture-model> (дата звернення: 16.01.2025).

48. Побудова та розуміння алгоритмів: крок за кроком для новачків . [Електронний ресурс]. Режим доступу до ресурсу: <https://cloud.itstep.org/blog/building-and-understanding-algorithms-a-step-by-step-guide-for-beginners> (дата звернення: 20.01.2025).

49. Алгоритми в програмуванні: що потрібно знати? [Електронний ресурс]. Режим доступу до ресурсу: <https://denzadnem.com.ua/blogy/korysni-porady/168016> (дата звернення: 20.01.2025).

50. Use Case Diagram Templates for Online Shopping Management [Електронний ресурс]. Режим доступу до ресурсу: <http://edrawmax.wondershare.com/diagram-tips/use-case-diagrams-online-shopping-management.html> (дата звернення: 22.01.2025).

51. Greimas' Actantial Model. [Електронний ресурс]. Режим доступу до ресурсу: https://www.researchgate.net/figure/Greimas-Actantial-Model-MA-Narrative-Environments-2018_fig1_356612598 (дата звернення: 24.01.2025).

52. Conventional User Authentication & Authorization in Web Application [Електронний ресурс]. Режим доступу до ресурсу: <https://siddhivinayak-sk.medium.com/conventional-user-authentication-authorization-in-web-application-5ba6882143e2> (дата звернення: 28.01.2025).

53. Що таке адмін-панель сайту та як туди зайти [Електронний ресурс]. Режим доступу до ресурсу: <https://hostiq.ua/blog/ukr/admin-panel/> (дата звернення: 03.02.2025).

54. The Role of the Database in E-Commerce. [Електронний ресурс]. Режим доступу до ресурсу: <https://trellis.co/blog/the-role-of-the-database-in-e-commerce> (дата звернення: 07.02.2025).

55. Як підібрати кольори? [Електронний ресурс]. Режим доступу до ресурсу: <https://trellis.co/blog/the-role-of-the-database-in-e-commerce> (дата звернення: 12.02.2025).

56. Як вибрати шрифт для сайту [Електронний ресурс]. Режим доступу до ресурсу: <https://www.site2b.ua/ua/web-blog-ua/yak-vibrati-shrift-dlya-sajtu.html> (дата звернення: 18.02.2025).

57. CSS Layout Commerce. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/CSS_layout (дата звернення: 21.02.2025).

58. Using media queries [Електронний ресурс]. Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_media_queries/Using_media_queries (дата звернення: 25.02.2025).

59. Як адаптувати свій сайт для мобільних пристроїв сайту [Електронний ресурс]. Режим доступу до ресурсу: <https://cityhost.ua/uk/blog/yak-optimizuvati-sviy-sayt-dlya-mobilnih-pristrov.html> (дата звернення: 28.02.2025).

60. Вплив навколишнього середовища на здоров'я людини [Електронний ресурс]. Режим доступу до ресурсу: <https://polyanytsya.if.ua/vplyv-navkolyshnogo-seredovyshha-na-zdorovya-lyudyny/> (дата звернення: 05.03.2025).

61. Найактуальніші екологічні проблеми Землі [Електронний ресурс]. Режим доступу до ресурсу: <https://eco.aep.kiev.ua/novini/15-najaktualnishihekologichnih-problem-zemli/> (дата звернення: 10.03.2025).

62. Human Impacts on the Environment [Електронний ресурс]. Режим доступу до ресурсу: <https://education.nationalgeographic.org/resource/resource-library-human-impacts-environment/> (дата звернення: 14.03.2025).

63. How has the environment been affected by the Chernobyl accident? [Електронний ресурс]. Режим доступу до ресурсу: <https://www.greenfacts.org/en/chernobyl/1-2/3-chernobyl-environment.htm> (дата звернення: 24.03.2025).

64. 1. Розділ 2.3: Вплив людської діяльності на екосистеми.. *Екосистеми та люди*. LibreTexts, 2022.

65. П'ять стратегій кругового дизайну? [Електронний ресурс]. Режим доступу до ресурсу: <https://info.ppv.net.ua/five-circular-design-strategies/> (дата звернення: 17.03.2025).

66. Чи впливають дизайнери на глобальні екологічні виклики [Електронний ресурс]. Режим доступу до ресурсу: <https://info.ppv.net.ua/global-environmental-challenges/> (дата звернення: 10.04.2025).

67. 1. ДСТУ ISO 9241-5-2004. ЕРГОНОМІЧНІ ВИМОГИ ДО РОБОТИ З ВІДЕОТЕРМІНАЛАМИ В ОФІСІ Частина 5. Вимоги до компонування робочого місця та до робочої пози. Київ, 2006. 18 с.

68. Гігієнічні вимоги до організації і обладнання робочих місць користувачів комп'ютерів [Електронний ресурс]. Режим доступу до ресурсу: <https://сро.stu.cn.ua/Oksana/posibnik/1010.html> (дата звернення: 15.04.2025).

69. Стандартизація в дизайнерській діяльності: навчальний посібник / В.О. Свірко, А.Л. Рубцов, О.В. Чемакіна, О.В. Кузьмін / За заг. ред. В.О. Свірка. – Херсон : ОЛДІ-ПЛЮС, 2020. – 232 с. ISBN 978-966-289-355-7

70. The Influence of Color on Human Behavior: A Deep Dive [Електронний ресурс]. Режим доступу до ресурсу: <https://imotions.com/blog/insights/color-and-human-behavior/> (дата звернення: 18.04.2025).

71. ДСТУ ISO 3864-1:2005. Графічні символи. Кольори та знаки безпеки. Частина 1. Принципи проектування знаків безпеки для робочих місць та місць громадського призначення. Київ, 2005. 24 с.

72. How To Use Color To Improve Your Conversion Rate [Електронний ресурс]. Режим доступу до ресурсу: <https://www.forbes.com/councils/theyec/2022/01/31/how-to-use-color-to-improve-your-conversion-rate/> (дата звернення: 24.04.2025).

73. The Role of Color Psychology in Workplace Productivity [Електронний ресурс]. Режим доступу до ресурсу: <https://onfra.io/blogs/color-psychology-and-its-role-in-workplace-productivity/> (дата звернення: 30.04.2025).

74. How Colors Affect Conversion Rate [Електронний ресурс]. Режим доступу до ресурсу: <https://www.quicksprout.com/how-colors-affect-conversion-rate/> (дата звернення: 05.05.2025).

ДОДАТКИ

Назва додатка А

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
імені ІВАНА ПУЛЮЯ
НАУКОВЕ ТОВАРИСТВО ім. ШЕВЧЕНКА**

МАТЕРІАЛИ

ХІІ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



18–19 грудня 2024 року

**ТЕРНОПІЛЬ
2024**

УДК 004.02

О. Ю. Петрик

(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ

UDC 004.02

O. Y. Petryk

UI/UX EFFECTIVENESS IN OPTIMIZING THE USER EXPERIENCE OF
ONLINE STORES

Дослідження користувацького досвіду є основним етапом у створенні ефективних онлайн-магазинів. Комфортна взаємодія з інтерфейсом сприяє зростанню конверсії та лояльності клієнтів.

Статистика показує, що добре продумана система копиків та інтуїтивно зрозуміла навігація можуть зменшити рівень відмов і заохотити клієнтів завершити свої покупки. Коли користувачі можуть легко прокручувати продукти та знаходити те, що вони шукають, вони, швидше за все, повернуться, незалежно від того, чи здійснюють покупки в Інтернеті, чи переходять на взаємодію офлайн.

Поширені проблеми UX на платформах електронної комерції можуть перешкодити вашій здатності підтримувати лояльність споживачів і стимулювати продажі. Наприклад, відсутність соціальних доказів, таких як відгуки клієнтів або рейтинги, може змусити потенційних покупців вагатися у прийнятті рішення про покупку. Крім того, розуміння важливості багатоканального досвіду, коли клієнти можуть плавно переходити між онлайн-покупками та офлайн-магазинами, є життєво важливим для підтримки взаємодії та задоволення.

Успіх діяльності онлайн магазину залежить від кількох факторів, зокрема:

- Якість пропонованого продукту або послуги
- Якість контенту, що представляє пропозицію клієнтам
- Якість дизайну електронної платформи — веб-сайту та/або мобільного додатку — за допомогою якої здійснюються продажі.

Тож, очевидно, що UI/UX дизайн є важливим: продумана логіка та переходи, інтуїтивно зрозумілі мікровзаємодії, швидкий зворотній зв'язок із системою, приваблива презентація продукту та плавні потоки платежів – усе це безпосередньо впливає на прибутковість бізнесу в конкурентному середовищі онлайн продажів.

Дизайнери та бізнес-експерти повинні співпрацювати, щоб покращити взаємодію з користувачами, що зрештою принесе користь цільовій аудиторії.

Література

1. Optimizing Online Stores With Essential UX Design Best Practices [Electronic resource] – 2024. URL: <https://finch.com/blog/optimizing-online-stores-with-essential-ux-design-best-practices>.
2. E-commerce UX and UI Design Guide for Best Practices [Electronic resource]. 2024. URL: <https://orthoplexolutions.com/web-development/e-commerce-ux-and-ui-design-best-practices-a-full-guide/>.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет імені Івана Пулюя (Україна)
Університет імені П'єра і Марії Кюрі (Франція)
Маріборський університет (Словенія)
Технічний університет у Кошице (Словаччина)
Вільнюський технічний університет ім. Гедимінаса (Литва)
Наукове товариство ім. Т.Шевченка

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник
тез доповідей

**XIII Міжнародної науково-практичної
конференції молодих учених та студентів**
11-12 грудня 2024 року



УКРАЇНА
ТЕРНОПІЛЬ – 2024

УДК 004.02

О.Ю. Петрик

Тернопільський національний технічний університет імені Івана Пулюя, Україна

АНАЛІЗ UI/UX РОЗРОБОК ПРИ СТВОРЕННІ ОНЛАЙН-МАГАЗИНІВ

O.Y. Petryk

UI/UX ANALYSIS OF CREATING ONLINE-STORES

Аналіз UI/UX розробок при створенні онлайн-магазинів є ключовим етапом, що впливає на успіх платформи. Зручний і інтуїтивно зрозумілий інтерфейс забезпечує легкість навігації для користувачів, зменшивши час пошуку потрібних товарів. Аналіз конкурентів та тестування прототипів допомагають виявити слабкі місця і вдосконалити кожен етап користувацького шляху. В результаті, якісний UI/UX дизайн підвищує лояльність клієнтів і конверсію, що має вплив на прибутковість онлайн-магазину [1].

UX-дизайн (User Experience) – це створення архітектури сайту, яка буде зручною для користувачів. Елементи розташовані інтуїтивно, користувач виконує цільову дію за мінімум часу та зусиль.

UI дизайн (User Interface) створення візуального вигляду сайту. Зовнішній вигляд сайту відповідає цільовій аудиторії, на ньому хочеться залишатися довше [2].

Коли мова заходить про розробку ефективного UI/UX для електронної комерції, є кілька ключових елементів, які не можна забувати (див. табл. 1).

Таблиця 1. Основні елементи ефективного UI/UX для онлайн-магазинів

Елемент	Ключові характеристики	Переваги
Навігація	Чіткі меню, фільтри, навігаційні "крихти"	Зменшує труднощі, підвищує коефіцієнт конверсії
Мобільна оптимізація	Адаптивний дизайн, швидке завантаження	Підвищує лояльність мобільних користувачів
Презентація продукту	Якісні зображення, інструменти перегляду товарів	Залучає увагу, полегшує ухвалення рішень
Персоналізація	Рекомендації, базовані на AI	Збільшує середній чек, підвищує задоволеність
Доступність	Підтримка читалок, інтуїтивна навігація	Розширює аудиторію, покращує інклюзивність
Аналітика	Теплові карти, A/B-тестування	Оптимізує шлях користувача, підвищує ефективність

Отже, аналіз UI/UX у створенні онлайн-магазинів має вирішальне значення, адже саме дизайн інтерфейсу та користувацький досвід впливають на лояльність покупців, швидкість прийняття рішень і загальну ефективність роботи платформи.

Література

1. Unlocking the Power of UX Design for Ecommerce Businesses [Electronic resource] – 2024. – <https://www.ecwid.com/blog/ux-design-for-ecommerce-businesses.html>.
2. 13 Tips From a UI/UX Designer To Make Any Online Store Better [Electronic resource] – 2021. – <https://cart-power.com/blog/13-tips-from-a-ui-ux-designer-to-make-any-online-store-better/>.

Головна сторінка інтернет-магазину квітів «index.html»

```

<!DOCTYPE html>
<html lang="uk">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Квітковий магазин</title>
    <link rel="stylesheet" href="styles.css">
    <link rel="preconnect"
href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wgh
t@0,100..900;1,100..900&display=swap" rel="stylesheet">
    <link rel="preconnect"
href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wgh
t@0,100..900;1,100..900&family=Pangolin&family=Rubik+Puddles&displ
ay=swap" rel="stylesheet">
  </head>
<body>

  <main>
    <div class="container">
      <div class="image-section">
        
      </div>
      <div class="text-section">
<nav>
  <div class="nav-container">
    <div class="burger-menu-icon" id="burger-icon">
      <span></span>
      <span></span>
      <span></span>
    </div>

</div>

    <ul class="nav-menu">
      <li><a href="flower.php">Каталог</a></li>
      <li><a href="#" class="active">Домашня</a></li>
      <li><a href="#about">Про нас</a></li>
      <li><a href="#delivery">Доставка</a></li>
    </ul>

```

```
</nav>
```

```

        <h1 style="font-weight: 200; font-size: 40px; margin-
bottom: 0; margin-top: 0;">Квіти - <br> це про
<span>любов</span></h1>
        <p style="font-size: 15px; margin-bottom: 0; margin-
top: 0;">Відкрийте мову ніжності та краси з нашими унікальними
букетами. <br>
        Подаруйте емоції, що запам'ятовуються надовго, <br>
        разом із нашим квітковим магазином.</p>
        <a href="flower.php">
        <button class="choose-bouquet-btn">Вибрати
букет</button>
        </a>

        <div class="foot">
            <p style="margin-top: 30px;">By Olesya Petryk</p>
            <p style="margin-top: 30px;">2024</p>
        </div>

    </div>
</div>
<div class="container2">
    
    <h1 style=" font-weight: 100; font-size: 20px; color:
white;margin: auto;margin-top: 10px; ">турбота в кожному
букеті</h1>
    <h1 style="font-weight: 500; font-size: 35px; color:
white;margin: auto;margin-top: 10px; margin-bottom: 50px; text-
align: center; font-family: Rubik Puddles; ">Улюблені букети <br>
наших клієнтів</h1>
</div>
<section class="why-choose-us">
    <h2>Why choose us?</h2>
    <div class="features">
        <div class="feature-item">
            
            <p>Fast delivery</p>
        </div>
        <div class="feature-item">
            
            <p>Location</p>
        </div>
        <div class="feature-item">
            
            <p>Quality of flowers</p>
        </div>
        <div class="feature-item">
            

```

```

        <p>Tips</p>
    </div>
</div>
</section>

<section id="about" class="about-section">
    <div class="about-text">
        <h2 style="font-family: Montserrat; font-weight: 600;">Про нас</h2>
        <p style="margin-top: 50px;"><strong>BLOOM</strong> - це не просто магазин квітів. Це місце, де квіти стають частиною моментів, створюють атмосферу радості і наповнюють ваше життя емоціями.</p>
        <p>Наша команда - це поціновувачі природи, які з любов'ю виготовляють кожну композицію і знають, як зробити ваше свято незабутнім.</p>
    </div>
    <div class="about-images">
        
        
    </div>
</section>
<section class="bestsellers">
    <h2 style="font-family: 'Rubik Puddles', regular; font-weight: 600;">Бестселери</h2>
    <div class="bestseller-items">
        <div class="bestseller-item">
            
            <p>Букетик повітряних півоній</p>
            <p class="price">1540 грн</p>
        </div>
        <div class="bestseller-item">
            
            <p>Ніжний букет еustom</p>
            <p class="price">1220 грн</p>
        </div>
        <div class="bestseller-item">
            
            <p>Букетик жоржин</p>
            <p class="price">630 грн</p>
        </div>
    </div>
    <a href="flower.html" class="see-more">Подивитись більше</a>
</section>

<section class="special-offer">
    <h2>Каталог</h2>
    <p style="margin-bottom: 40px; font-size: 20px; font-weight: 300; margin-top: 10px;">Виберіть найкращі квіти із каталогу</p>
    <div class="offer-content">
        <div class="offer-item">
            

```

```

        <a href="flower.html" class="offer-
label">Флористика</a>
    </div>
    <div class="offer-item">
        
        <a href="catalog.html" class="offer-label"
style="margin: 0 30px 0 0;" >Букети</a>
    </div>
</div>
</section>

<div id="delivery" class="delivery">
<div class="delivery-section">
    <div class="delivery-content">
        <h2 class="delivery-title">Доставка</h2>
        <p class="delivery-text">
            Доставка здійснюється кур'єрською службою. Оплата при
отриманні готівкою або переказ на банківський рахунок.
            При замовленні на суму від <strong>800 грн</strong> -
доставка безкоштовна.
        </p>
        
    </div>
</div>
</div>

<div class="service">
    <h2 style="font-family: 'Rubik Puddles', regular;">Спробуйте
послуги для особливих подій</h2>
    <div class="service-content">
        <div class="service-image">
            
            
        </div>
        <div class="service-text" >
            <h3>Весілля</h3>
            <p>Флористи створюють вишукані букети для нареченої,
бутоньєрки для нареченого, а також прикраси для автомобіля, залу
та урочистої арки.</p>
            <h3>Рецепція</h3>
            <p>Ми пропонуємо широкий вибір букетів, вазонних
композицій та флористичного декору для прийому гостей. Флористи
ретельно підберуть композиції, які підкреслять стиль і атмосферу
Вашого заходу.</p>
            <h3>Бізнес</h3>
            <p>Ми маємо великий досвід обслуговування ділових
заходів, включаючи квіткове оформлення офісів, конференц-залів і
виставкових стендів. Професійні флористи допоможуть створити
ефектний і благородний образ для Вашого ділового заходу.</p>
        </div>
    </div>

```

```

    </div>
</div>

<div class="contact">
    <h2 style="font-family: 'Rubik Puddles', regular; text-align:
right;">Зв'язатись з нами</h2>
    <div class="contact-columns">
        <div class="left-column">
            <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2588.27
59800913477!2d25.591944575114606!3d49.554815651447434!2m3!1f0!2f0!
3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x4730318d39122ced%3A0x7f663
7a4625fe208!2sLofant!5e0!3m2!1suk!2sua!4v1738620775036!5m2!1suk!2s
ua" width="100%" height="350" style="border-radius: 15px;"
allowfullscreen="" loading="lazy" referrerpolicy="no-referrer-
when-downgrade"></iframe>
        </div>
        <div class="right-column">
            <input type="text" class="form-field"
placeholder="Ваше ім'я" />
            <input type="tel" class="form-field" placeholder="Ваш
номер телефону" />
            <button class="form-submit">Чекаю на дзвінок</button>
        </div>
    </div>
</div>
</main>
<footer role="contentinfo">
    <hr style="border: 1px solid white; margin-bottom: 20px;
margin-top: 0; width: 100%;">
    <div class="footer-container">
        <div class="logo">
            
        </div>
        <div class="info">
            <h3>Інформація</h3>
            <a href="#">Про нас</a>
            <a href="#">Доставка</a>
            <a href="#">Каталог</a>
        </div>
        <div class="contacts">
            <h3>Контакти</h3>
            <p style="margin: 3px;">+38096 666 6666</p>
            <p style="margin: 3px;">вул. Валова, 7, Тернопіль</p>
            <p style="margin: 3px;"><a href="mailto:info@bloom-
shop.com.ua">info@bloom-shop.com.ua</a></p>
        </div>
    </div>
</footer>

<Script >

document.addEventListener('DOMContentLoaded', function () {

```

```

const burgerIcon = document.getElementById('burger-icon');
const navMenu = document.querySelector('.nav-menu');

// Створюємо overlay елемент
const overlay = document.createElement('div');
overlay.className = 'menu-overlay';
overlay.id = 'menu-overlay';
document.body.appendChild(overlay);

// Функція для переключення меню
function toggleMenu() {
    navMenu.classList.toggle('active');
    overlay.classList.toggle('active');

    if (navMenu.classList.contains('active')) {
        document.body.style.overflow = 'hidden';
    } else {
        document.body.style.overflow = '';
    }
}

// Функція для закриття меню
function closeMenu() {
    navMenu.classList.remove('active');
    overlay.classList.remove('active');
    document.body.style.overflow = '';
}

// Клік на бургер іконку
if (burgerIcon) {
    burgerIcon.addEventListener('click', toggleMenu);
}

// Клік на overlay для закриття
overlay.addEventListener('click', closeMenu);

// Клік на пункти меню для закриття
const navLinks = navMenu.querySelectorAll('a');
navLinks.forEach(link => {
    link.addEventListener('click', closeMenu);
});

// Закриття при натисканні Escape
document.addEventListener('keydown', function(e) {
    if (e.key === 'Escape' &&
navMenu.classList.contains('active')) {
        closeMenu();
    }
});

// Закриття при зміні розміру екрану
window.addEventListener('resize', function() {

```

```
        if (window.innerWidth > 480 &&
navMenu.classList.contains('active')) {
            closeMenu();
        }
    });
});

</Script >

</body>
</html>
```

Сторінка каталогу інтернет-магазину квітів «flower.html»

```

<?php
// Секція PHP коду

$servername = "fdb1029.Awardspace.net";
$username = "4532764_bloom";
$password = "olesyaa1020";
$dbname = "4532764_bloom";

// Створення з'єднання
$conn = new mysqli($servername, $username, $password, $dbname);

// Встановлення кодування UTF-8 для з'єднання
if (!$conn->set_charset("utf8mb4")) {
    // Обробка помилки, якщо потрібно
    // printf("Error loading character set utf8mb4: %s\n", $conn-
>error);
}

// Перевірка з'єднання
if ($conn->connect_error) {
    // У випадку AJAX-запиту, повертаємо помилку
    if (isset($_SERVER['HTTP_X_REQUESTED_WITH']) &&
    strtolower($_SERVER['HTTP_X_REQUESTED_WITH']) ===
    'xmlhttprequest') {
        http_response_code(500); // Встановлюємо HTTP статус
        помилки
        echo "Помилка з'єднання з базою даних.";
        exit; // Припиняємо виконання скрипта
    } else {
        // Якщо не AJAX
        die("Помилка з'єднання: " . $conn->connect_error);
    }
}

// Обробка AJAX-запиту для отримання даних кошика
if (isset($_SERVER['HTTP_X_REQUESTED_WITH']) &&
    strtolower($_SERVER['HTTP_X_REQUESTED_WITH']) === 'xmlhttprequest'
    && isset($_GET['action']) && $_GET['action'] === 'get_cart_data')
{
    header('Content-Type: application/json');
    $cartItems = [];

    // SQL запит для отримання даних кошика з таблиць cart,
    products та product_sizes
    $cartSql = "SELECT
                c.quantity,
                p.name,
                ps.size,
                ps.price,

```

```

        p.image
    FROM
        cart c
    JOIN
        products p ON c.product_id = p.id
    JOIN
        product_sizes ps ON c.product_id =
ps.product_id AND c.size = ps.size";

    $cartResult = $conn->query($cartSql);

    if ($cartResult && $cartResult->num_rows > 0) {
        while($row = $cartResult->fetch_assoc()) {
            $cartItems[] = $row;
        }
    }
    echo json_encode($cartItems);
    exit;
}

// Фільтрація за розміром: отримуємо значення з GET-параметра
'size'
$sizeFilter = isset($_GET['size']) ? $_GET['size'] : null;

// Перевіряємо, чи отримане значення є допустимим розміром або
порожнім рядком (для "Показати всі")
// Якщо значення фільтра некоректне і не пусте, скидаємо його
if ($sizeFilter !== null && $sizeFilter !== '' &&
!in_array($sizeFilter, ['XS', 'S', 'M', 'L'])) {
    $sizeFilter = null;
}

// Якщо size= '', це також вважається відсутністю фільтра (для
кнопки "Показати всі")
if ($sizeFilter === '') {
    $sizeFilter = null;
}

$sortOrder = isset($_GET['sort']) ? $_GET['sort'] : 'default'; //
За замовчуванням сортування 'default'

// Перевіряємо, чи отримане значення сортування є допустимим
if (!in_array($sortOrder, ['default', 'price_asc', 'price_desc']))
{
    $sortOrder = 'default'; // Якщо значення некоректне,
встановлюємо 'default'
}

$sql = ""; // Ініціалізуємо змінну для SQL-запиту

if ($sizeFilter) {

```

```

    $sql = "SELECT p.*, ps.size, ps.price
            FROM products p
            JOIN product_sizes ps ON p.id = ps.product_id
            WHERE ps.size = '" . $conn-
>real_escape_string($sizeFilter) . "'";

} else {
    // Якщо фільтр за розміром НЕ встановлено (включаючи випадок
    "Показати всі"),

    $sql = "SELECT p.*, ps_default.size, ps_default.price
            FROM products p
            JOIN (
                SELECT ps1.product_id, ps1.size, ps1.price
                FROM product_sizes ps1
                JOIN (
                    SELECT product_id, MIN(CASE size WHEN 'XS'
THEN 1 WHEN 'S' THEN 2 WHEN 'M' THEN 3 WHEN 'L' THEN 4 ELSE 5 END)
as size_order
                    FROM product_sizes
                    GROUP BY product_id
                ) as ps_min ON ps1.product_id = ps_min.product_id
                WHERE CASE ps1.size WHEN 'XS' THEN 1 WHEN 'S' THEN
2 WHEN 'M' THEN 3 WHEN 'L' THEN 4 ELSE 5 END = ps_min.size_order
                ) as ps_default ON p.id = ps_default.product_id";
}

$orderBy = "";
switch ($sortOrder) {
    case 'price_asc':
        $orderBy = ($sizeFilter) ? "ps.price ASC" :
"ps_default.price ASC";
        break;
    case 'price_desc':
        $orderBy = ($sizeFilter) ? "ps.price DESC" :
"ps_default.price DESC";
        break;
    case 'default':
    default:
        $orderBy = "p.created_at DESC"; // Сортуння за датою
створення (за замовчуванням)
        break;
}

$sql .= " ORDER BY " . $orderBy;

$sql .= " LIMIT 12"; // Ваша логіка пагінації потребуватиме
додаткових змін тут (OFFSET)

// Виконання запиту
$result = $conn->query($sql);

// Обробка AJAX-запиту для фільтрації товарів

```

```

if (isset($_SERVER['HTTP_X_REQUESTED_WITH']) &&
    strtolower($_SERVER['HTTP_X_REQUESTED_WITH']) === 'xmlhttprequest'
    && !isset($_GET['action'])) { // Додано перевірку
    !isset($_GET['action'])
        $output = "";
        if ($result && $result->num_rows > 0) {
            while($row = $result->fetch_assoc()) {
                $output .= '<div class="product">';
                $output .= '    ';
                $output .= '    <p class="product-info">';
                $output .= '        <a href="info.php?id=' .
$row['id'] . '"> . htmlspecialchars($row['name']) . '</a>';
                $output .= '        <span class="price">' .
htmlspecialchars($row['price']) . ' грн</span>';
                $output .= '    </p>';
                $output .= '</div>';
            }
        } else {
            $output = "<p>Товари не знайдено.</p>";
        }
        echo $output;
        exit;
    }
?>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Флористика</title>
    <link rel="stylesheet" href="style-flower.css">
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400
;500;700&display=swap" rel="stylesheet">
    <Script src="https://cdn.tailwindcss.com"></Script >
    <style>

        /* Додаткові стилі для сумісності та покращення вигляду */
        body {
            font-family: 'Montserrat', sans-serif;
        }

        header h1 {
            font-weight: 600; /* Жирний шрифт для Montserrat */
            font-size: 2rem; /* Більший розмір, можна налаштувати */
            margin-top: 1rem; /* Відступ зверху */
            background-color: #FFF5ED;

```

```

    }
    /* Стили для панелі кошика, якщо Tailwind класи
конфліктують або недостатні */
    #slideOutCart {
        transition: right 0.3s ease-in-out;
    }
    .product img { /* Силь для зображень товарів у каталозі
*/
        max-width: 100%;
        height: auto;
        aspect-ratio: 1 / 1; /* Робить зображення квадратними
*/
        object-fit: cover; /* Обрізає зображення, щоб
заповнити контейнер */
        border-radius: 8px; /* Округлені кути */
    }
    .cart-item-image { /* Силь для зображень товарів у кошику
*/
        width: 64px; /* 4rem */
        height: 64px; /* 4rem */
        object-fit: cover;
        border-radius: 4px; /* md */
    }
    /* Додатковий стиль для кнопки "Оформити замовлення" */
    .btn-checkout {
        background-color: #661818; /* Заданий колір */
    }
    .btn-checkout:hover {
        background-color: #7b2020; /* Трохи темніший колір при
наведенні */
    }
</style>
</head>
<body style= "background-color: #FFF5ED;">
<div class="header-container">
    <div class="logo">
        
    </div>

    <div class="menu-toggle" onclick="toggleMenu()">
        <span></span><span></span><span></span>
    </div>

<nav id="navMenu">
    <button class="close-menu-btn" style =
"display:none;">&times;</button> <ul>
        <li class="active"><a href="#">Каталог</a></li>
        <li><a href="index.html">Головна</a></li>
        <li><a href="index.html#about">Про нас</a></li>
        <li><a href="index.html#delivery">Доставка</a></li>
    </ul>

```

```

</nav>

    <div class="cart" id="cartIconTrigger" style="cursor:
pointer;">
        
    </div>
</div>

<main>
<aside class="filters">
    <div class="filter-content">
        <h2>Фільтр</h2>
        <form id="sizeFilterForm" method="GET">
            <div class="filter-group">
                <h3>Розмір</h3>
                <label><input type="radio" name="size" value="XS"
<?= ($sizeFilter === 'XS') ? 'checked' : '' ?>> XS</label>
                <label><input type="radio" name="size" value="S"
<?= ($sizeFilter === 'S') ? 'checked' : '' ?>> S</label>
                <label><input type="radio" name="size" value="M"
<?= ($sizeFilter === 'M') ? 'checked' : '' ?>> M</label>
                <label><input type="radio" name="size" value="L"
<?= ($sizeFilter === 'L') ? 'checked' : '' ?>> L</label>
                <label><input type="radio" name="size" value=""
<?= ($sizeFilter === null) ? 'checked' : '' ?>> Показати
всі</label>
            </div>
        </form>
    </div>
</aside>

    <div class="filter-rect">Фільтр</div>

    <section class="product-list">
        <div class="sorting">
            <label for="sortSelect">Сортування:</label>
            <select id="sortSelect">
                <option value="default" <?= ($sortOrder ===
'default') ? 'selected' : '' ?>>За замовчуванням</option>
                <option value="price_asc" <?= ($sortOrder ===
'price_asc') ? 'selected' : '' ?>>За ціною (зростання)</option>
                <option value="price_desc" <?= ($sortOrder ===
'price_desc') ? 'selected' : '' ?>>За ціною (спадання)</option>
            </select>
        </div>

        <div class="products">

```

```

        <?php
        if ($result && $result->num_rows > 0):
            // Скидаємо показчик результату, якщо він був
            використаний раніше (хоча AJAX виходить через exit)
            $result->data_seek(0);
            while($row = $result->fetch_assoc()):
                ?>
                    <div class="product">
                        "
                        onerror="this.onerror=null;this.src='https://placeholder.co/300x300/
                        E0E0E0/B0B0B0?text=Image+Not+Found';">
                        <p class="product-info">
                            <a href="info.php?id=<?= $row['id']
                            ?>"><?= htmlspecialchars($row['name']) ?></a>
                            <span class="price"><?=
                            htmlspecialchars($row['price']) ?> грн</span>
                        </p>
                    </div>
                <?php
                endwhile;
            else:
                ?>
                    <p>Товари не знайдено.</p>
                <?php endif; ?>
            </div>

            <div class="pagination">
                <a class="active">1</a>
                <a href="#">2</a>
                <a href="#">3</a>
            </div>
        </section>
    </main>

    <div id="slideOutCart" class="fixed top-0 right-[-100%] sm:right-
    [-450px] w-full sm:max-w-md h-full bg-white shadow-xl z-50 p-6
    flex flex-col">
        <div class="flex justify-between items-center pb-4 border-b
        border-gray-300">
            <h2 class="text-xl font-semibold text-gray-800">Кошик
            (<span id="cartItemCount">0</span>)</h2>
            <button id="closeCartButton" class="text-gray-600
            hover:text-gray-800 text-3xl">&times;</button>
        </div>
        <div id="cartItemsContainer" class="flex-grow overflow-y-auto
        py-4 space-y-3">
            <p id="emptyCartMsg" class="text-gray-500 text-center mt-
            10">Ваш кошик порожній.</p>
        </div>
        <div class="pt-4 border-t border-gray-300">
            <div class="flex justify-between items-center mb-4">

```

```

        <span class="text-lg font-semibold text-gray-800">Всього:</span>
        <span id="cartTotal" class="text-lg font-semibold text-gray-800">0.00 грн</span>
    </div>
    <button class="w-full btn-checkout hover:bg-opacity-90 text-white font-bold py-3 px-4 rounded-lg transition-colors duration-150">
        Оформити замовлення
    </button>
</div>
</div>

<div id="authModal" style="display:none; position:fixed; top:50%; left:50%; transform:translate(-50%,-50%); background:#fff; padding:15px; border:1px solid #ccc; border-radius:10px; box-shadow:0 4px 10px rgba(0,0,0,0.2); z-index:9999; width:280px; height:320px;">
    <span onclick="closeModal()" style="position:absolute; top:5px; right:10px; cursor:pointer; font-size:18px; color:#999;">&times;</span>
    <iframe src="log.php" style="width:100%; height:100%; border:none;"></iframe>
</div>

<footer>
    <div class="footer-container">
        <div class="logo">
            
        </div>
        <div class="info">
            <h3>Інформація</h3>
            <a href="#">Про нас</a>
            <a href="#">Доставка</a>
            <a href="#">Каталог</a>
        </div>
        <div class="contacts">
            <h3>Контакти</h3>
            <p>+38096 666 6666</p>
            <p>вул. Валова, 7, Тернопіль</p>
            <p><a href="mailto:info@bloom-shop.com.ua">info@bloom-shop.com.ua</a></p>
        </div>
    </div>
</footer>

<Script >
// Функція для перемикання меню (з вашого коду)
function toggleMenu() {
    document.querySelector('nav').classList.toggle('active');
}

```

```

// JavaScript для AJAX фільтрації та сортування (з вашого коду)
const sizeFilterForm = document.getElementById('sizeFilterForm');
const sortSelect = document.getElementById('sortSelect');
const productsContainer = document.querySelector('.product-list
.products');

function fetchProducts(filtersAndSort) {
    const currentUrl = new URL(window.location.href);
    currentUrl.pathname = currentUrl.pathname.substring(0,
currentUrl.pathname.lastIndexOf("/") + 1) + '<?php echo
basename($_SERVER["PHP_SELF"]); ?>'; // Ensure it points to the
current Script

    if (filtersAndSort.size) {
        currentUrl.searchParams.set('size', filtersAndSort.size);
    } else {
        currentUrl.searchParams.delete('size');
    }

    if (filtersAndSort.sort && filtersAndSort.sort !== 'default')
{
        currentUrl.searchParams.set('sort', filtersAndSort.sort);
    } else {
        currentUrl.searchParams.delete('sort');
    }

    fetch(currentUrl.toString(), {
        headers: {
            'X-Requested-With': 'XMLHttpRequest'
        }
    })
    .then(response => {
        if (!response.ok) {
            return response.text().then(text => {
                throw new Error('Помилка HTTP: ' + response.status
+ ' - ' + text);
            });
        }
        return response.text();
    })
    .then(html => {
        if (productsContainer) {
            productsContainer.innerHTML = html;
        }
    })
    .catch(error => {
        console.error('Помилка під час AJAX-запиту для товарів:',
error);
        if (productsContainer) {
            productsContainer.innerHTML = "<p style='color:
red;'>Виникла помилка під час завантаження товарів.</p>";
        }
    });
}

```

```

}

if (sizeFilterForm && sortSelect && productsContainer) {
    sizeFilterForm.addEventListener('change', function(event) {
        if (event.target.name === 'size') {
            const selectedSize = event.target.value;
            const currentSort = sortSelect.value;
            fetchProducts({ size: selectedSize, sort: currentSort
});
            updateUrl({ size: selectedSize, sort: currentSort });
        }
    });

    sortSelect.addEventListener('change', function() {
        const selectedSort = this.value;
        const sizeInputs =
sizeFilterForm.querySelectorAll('input[name="size"]');
        let currentSize = null;
        sizeInputs.forEach(input => {
            if (input.checked) {
                currentSize = input.value;
            }
        });
        fetchProducts({ size: currentSize, sort: selectedSort });
        updateUrl({ size: currentSize, sort: selectedSort });
    });

    window.addEventListener('popstate', function(event) {
        const currentUrl = new URL(window.location.href);
        const sizeParam = currentUrl.searchParams.get('size');
        const sortParam = currentUrl.searchParams.get('sort');

        fetchProducts({ size: sizeParam, sort: sortParam });

        const sizeInputs =
sizeFilterForm.querySelectorAll('input[name="size"]');
        sizeInputs.forEach(input => {
            input.checked = (input.value === (sizeParam || ''));
        });
        sortSelect.value = sortParam || 'default';
    });
}

function updateUrl(params) {
    const currentUrl = new URL(window.location.href);
    currentUrl.pathname = currentUrl.pathname.substring(0,
currentUrl.pathname.lastIndexOf("/") + 1) + '<?php echo
basename($_SERVER["PHP_SELF"]); ?>';

    if (params.size) {
        currentUrl.searchParams.set('size', params.size);
    } else {
        currentUrl.searchParams.delete('size');
    }
}

```

```

    }
    if (params.sort && params.sort !== 'default') {
        currentUrl.searchParams.set('sort', params.sort);
    } else {
        currentUrl.searchParams.delete('sort');
    }
    history.pushState(null, '', currentUrl.toString());
}

// Логіка відображення модального вікна авторизації (з вашого коду)
const authModal = document.getElementById('authModal');
if (authModal) {
    setTimeout(function() {
        // Тут має бути перевірка, чи користувач авторизований
        authModal.style.display = 'block';
    }, 10000);
}

function closeModal() {
    if (authModal) {
        authModal.style.display = 'none';
    }
}

const cartIcon = document.getElementById('cartIconTrigger');
const slideOutCart = document.getElementById('slideOutCart');
const closeCartButton =
document.getElementById('closeCartButton');
const cartItemsContainer =
document.getElementById('cartItemsContainer');
const cartItemCount = document.getElementById('cartItemCount');
const cartTotal = document.getElementById('cartTotal');
const emptyCartMsg = document.getElementById('emptyCartMsg');

function openCart() {
    if (slideOutCart) {
        slideOutCart.classList.remove('right-[-100%]', 'sm:right-[-450px]');
        slideOutCart.classList.add('right-0');
        loadCartData();
    }
}

function closeCart() {
    if (slideOutCart) {
        slideOutCart.classList.remove('right-0');
        slideOutCart.classList.add('right-[-100%]', 'sm:right-[-450px]');
    }
}

```

```

async function loadCartData() {
  try {
    const currentScript Path =
window.location.pathname.substring(0,
window.location.pathname.lastIndexOf("/") + 1) + '<?php echo
basename($_SERVER["PHP_SELF"]); ?>';
    const response = await fetch(`${currentScript
Path}?action=get_cart_data`, {
      headers: {
        'X-Requested-With': 'XMLHttpRequest'
      }
    });
    if (!response.ok) {
      throw new Error(`HTTP error! status:
${response.status}`);
    }
    const items = await response.json();

    renderCartItems(items);
  } catch (error) {
    console.error('Помилка завантаження даних кошика:',
error);
    if (cartItemsContainer) {
      cartItemsContainer.innerHTML = '<p class="text-red-500
text-center">Не вдалося завантажити кошик.</p>';
    }
    if (emptyCartMsg) emptyCartMsg.style.display = 'none';
  }
}

function renderCartItems(items) {
  if (!cartItemsContainer || !cartItemCount || !cartTotal ||
!emptyCartMsg) return;

  cartItemsContainer.innerHTML = ''; // Очистити попередні
товари
  let totalAmount = 0;
  let itemCount = 0;

  if (items && items.length > 0) {
    emptyCartMsg.style.display = 'none';
    items.forEach(item => {
      const itemElement = document.createElement('div');
      itemElement.className = 'flex items-center space-x-3
p-3 bg-gray-50 rounded-lg shadow-sm';

      const imageName = item.image ? `img2/${item.image}` :
'https://placeholder.co/64x64/E0E0E0/B0B0B0?text=N/A';
      const itemName = item.name || 'Без назви';
      const itemSize = item.size || 'N/A';
      const itemQuantity = parseInt(item.quantity) || 0;

```

```

        const itemPrice = parseFloat(item.price) || 0;

        itemElement.innerHTML = `
            
            <div class="flex-grow">
                <h3 class="text-sm font-semibold text-gray-
700">${htmlspecialchars(itemName)}</h3>
                <p class="text-xs text-gray-500">Розмір:
${htmlspecialchars(itemSize)}</p>
                <p class="text-xs text-gray-500">Кількість:
${itemQuantity}</p>
            </div>
            <div class="text-right">
                <p class="text-sm font-semibold text-gray-
700">${itemPrice.toFixed(2)} грн</p>
            </div>
        `;
        cartItemsContainer.appendChild(itemElement);
        totalAmount += itemPrice * itemQuantity;
        itemCount += itemQuantity;
    });
} else {
    emptyCartMsg.style.display = 'block';
}

cartItemCount.textContent = itemCount;
cartTotal.textContent = `${totalAmount.toFixed(2)} грн`;
}

// Функція для екранування HTML (запобігання XSS)
function htmlspecialchars(str) {
    if (typeof str !== 'string') return '';
    return str.replace(/([<>"']/g, function (match) {
        const replacements = {
            '&': '&amp;',
            '<': '&lt;',
            '>': '&gt;',
            '"': '&quot;',
            "'": '&#39;',
        };
    });
    return replacements[match];
});
}

if (cartIcon) {
    cartIcon.addEventListener('click', openCart);
}
if (closeCartButton) {
    closeCartButton.addEventListener('click', closeCart);
}

```

```

}

// Закриття кошика при кліку поза ним (опціонально)
document.addEventListener('click', function(event) {
    if (slideOutCart && cartIcon) {
        const isClickInsideCart =
slideOutCart.contains(event.target);
        const isClickOnCartIcon = cartIcon.contains(event.target);
        if (!isClickInsideCart && !isClickOnCartIcon &&
!slideOutCart.classList.contains('right-[-100%]')) {
            // closeCart(); // Розкоментуйте, якщо хочете таку
поведінку
        }
    }
});

</Script >
<Script >

function toggleMenu() {
    document.querySelector('nav').classList.toggle('active');
}

// Отримуємо посилання на кнопку закриття меню
const closeMenuBtn = document.querySelector('.close-menu-btn');

const navMenu = document.querySelector('nav');

if (closeMenuBtn && navMenu) { // Перевіряємо, чи існують елементи
    closeMenuBtn.addEventListener('click', function() {
        navMenu.classList.remove('active'); // Закриваємо меню
    });
}

const sizeFilterForm = document.getElementById('sizeFilterForm');
const sortSelect = document.getElementById('sortSelect');
const productsContainer = document.querySelector('.product-list
.products');

const filterRect = document.querySelector('.filter-rect');
const filtersAside = document.querySelector('aside.filters');
const closeFilterButton =
document.getElementById('closeFilterButton'); // If you added it
const mobileFilterOverlay =
document.getElementById('mobileFilterOverlay'); // If you added it

if (filterRect && filtersAside) {
    filterRect.addEventListener('click', function() {
        filtersAside.classList.add('active'); // Add 'active' to
show filter
        if (mobileFilterOverlay)
mobileFilterOverlay.classList.add('active'); // Show overlay

```

```

    });

    // Event listener for closing the filter
    if (closeFilterButton) {
        closeFilterButton.addEventListener('click', function() {
            filtersAside.classList.remove('active');
            if (mobileFilterOverlay)
mobileFilterOverlay.classList.remove('active');
        });
    }

    // Close filter when clicking on the overlay
    if (mobileFilterOverlay) {
        mobileFilterOverlay.addEventListener('click', function() {
            filtersAside.classList.remove('active');
            mobileFilterOverlay.classList.remove('active');
        });
    }

    // Optional: Close filter if window resized to desktop from
mobile while filter is open
    window.addEventListener('resize', function() {
        if (window.innerWidth > 480 &&
filtersAside.classList.contains('active')) {
            filtersAside.classList.remove('active');
            if (mobileFilterOverlay)
mobileFilterOverlay.classList.remove('active');
        }
        // Also close main nav menu if it's open on desktop resize
        if (window.innerWidth > 480 &&
document.querySelector('nav').classList.contains('active')) {
document.querySelector('nav').classList.remove('active');
        }
        // Also close main nav toggle if window resized to desktop
from mobile while nav menu is open
        if (window.innerWidth > 480 &&
document.querySelector('.menu-
toggle').classList.contains('active')) {
            document.querySelector('.menu-
toggle').classList.remove('active');
        }
    });
}

const sizeFilterForm = document.getElementById('sizeFilterForm');
const sortSelect = document.getElementById('sortSelect');
const productsContainer = document.querySelector('.product-list
.products');

function fetchProducts(filtersAndSort) {
    const currentUrl = new URL(window.location.href);

```

```

    currentUrl.pathname = currentUrl.pathname.substring(0,
currentUrl.pathname.lastIndexOf("/") + 1) + '<?php echo
basename($_SERVER["PHP_SELF"]); ?>';

    if (filtersAndSort.size) {
        currentUrl.searchParams.set('size', filtersAndSort.size);
    } else {
        currentUrl.searchParams.delete('size');
    }

    if (filtersAndSort.sort && filtersAndSort.sort !== 'default')
{
        currentUrl.searchParams.set('sort', filtersAndSort.sort);
    } else {
        currentUrl.searchParams.delete('sort');
    }

    fetch(currentUrl.toString(), {
        headers: {
            'X-Requested-With': 'XMLHttpRequest'
        }
    })
    .then(response => {
        if (!response.ok) {
            return response.text().then(text => {
                throw new Error('Помилка HTTP: ' + response.status
+ ' - ' + text);
            });
        }
        return response.text();
    })
    .then(html => {
        if (productsContainer) {
            productsContainer.innerHTML = html;
        }
    })
    .catch(error => {
        console.error('Помилка під час AJAX-запиту для товарів:',
error);
        if (productsContainer) {
            productsContainer.innerHTML = "<p style='color:
red;'>Виникла помилка під час завантаження товарів.</p>";
        }
    });
}

if (sizeFilterForm && sortSelect && productsContainer) {
    sizeFilterForm.addEventListener('change', function(event) {
        if (event.target.name === 'size') {
            const selectedSize = event.target.value;
            const currentSort = sortSelect.value;
            fetchProducts({ size: selectedSize, sort: currentSort
});
}
}

```

```

        updateUrl({ size: selectedSize, sort: currentSort });
    }
});

sortSelect.addEventListener('change', function() {
    const selectedSort = this.value;
    const sizeInputs =
sizeFilterForm.querySelectorAll('input[name="size"]');
    let currentSize = null;
    sizeInputs.forEach(input => {
        if (input.checked) {
            currentSize = input.value;
        }
    });
    fetchProducts({ size: currentSize, sort: selectedSort });
    updateUrl({ size: currentSize, sort: selectedSort });
});

window.addEventListener('popstate', function(event) {
    const currentUrl = new URL(window.location.href);
    const sizeParam = currentUrl.searchParams.get('size');
    const sortParam = currentUrl.searchParams.get('sort');

    fetchProducts({ size: sizeParam, sort: sortParam });

    const sizeInputs =
sizeFilterForm.querySelectorAll('input[name="size"]');
    sizeInputs.forEach(input => {
        input.checked = (input.value === (sizeParam || ''));
    });
    sortSelect.value = sortParam || 'default';
});
}

function updateUrl(params) {
    const currentUrl = new URL(window.location.href);
    currentUrl.pathname = currentUrl.pathname.substring(0,
currentUrl.pathname.lastIndexOf("/") + 1) + '<?php echo
basename($_SERVER["PHP_SELF"]); ?>';

    if (params.size) {
        currentUrl.searchParams.set('size', params.size);
    } else {
        currentUrl.searchParams.delete('size');
    }
    if (params.sort && params.sort !== 'default') {
        currentUrl.searchParams.set('sort', params.sort);
    } else {
        currentUrl.searchParams.delete('sort');
    }
    history.pushState(null, '', currentUrl.toString());
}
</Script >

```

```

<Script >
document.addEventListener('DOMContentLoaded', function () {
    const filterBtn = document.querySelector('.filter-rect');
    const filterPanel = document.querySelector('.filters');

    // Створення та додавання overlay, якщо ще немає
    let overlay = document.getElementById('mobileFilterOverlay');
    if (!overlay) {
        overlay = document.createElement('div');
        overlay.id = 'mobileFilterOverlay';
        overlay.classList.add('overlay');
        document.body.appendChild(overlay);
    }

    // Кнопка відкриття фільтра
    filterBtn?.addEventListener('click', () => {
        filterPanel?.classList.add('active');
        overlay.classList.add('active');
    });

    // Кнопка закриття (додається лише раз)
    if (filterPanel && !filterPanel.querySelector('.close-filter-
btn')) {
        const closeBtn = document.createElement('button');
        closeBtn.className = 'close-filter-btn';
        closeBtn.innerHTML = '&times;';
        filterPanel.appendChild(closeBtn);

        closeBtn.addEventListener('click', () => {
            filterPanel.classList.remove('active');
            overlay.classList.remove('active');
        });

        overlay.addEventListener('click', () => {
            filterPanel.classList.remove('active');
            overlay.classList.remove('active');
        });
    }
});
</Script >

</body>
</html>

<?php
// Закриття з'єднання з базою даних в кінці файлу
if ($conn) {
    $conn->close();
}
?>

```

Сторінка інформації інтернет-магазину квітів «info.html»

```

<?php
session_start();

// Підключення до бази даних
$servername = "fdb1029.awardspace.net";
$username = "4532764_bloom";
$password = "olesyaa1020";
$dbname = "4532764_bloom";

$conn = new mysqli($servername, $username, $password,
$dbname);
if ($conn->connect_error) {
    die("Помилка з'єднання: " . $conn->connect_error);
}

// Встановлення кодування UTF-8 для з'єднання
if (!$conn->set_charset("utf8mb4")) {
    // Обробка помилки, якщо потрібно
    // printf("Error loading character set utf8mb4: %s\n",
    $conn->error);
}

$product_id = $_GET['id'] ?? 0;
$product = null;
$sizes = [];

// Отримання інформації про товар
$sql = "SELECT * FROM products WHERE id = $product_id";
$result = $conn->query($sql);
if ($result && $result->num_rows > 0) {
    $product = $result->fetch_assoc();
}

$sql_sizes = "SELECT * FROM product_sizes WHERE product_id =
$product_id";
$result_sizes = $conn->query($sql_sizes);
if ($result_sizes && $result_sizes->num_rows > 0) {
    while ($row = $result_sizes->fetch_assoc()) {
        $sizes[] = $row;
    }
}

// Обробка додавання в кошик
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Логіка для кнопки "Купити" (buy_now)
    if (isset($_POST['buy_now'])) {
        if ($product['availability'] !== 'В наявності') {
            echo "<script>alert('На жаль, цього товару немає в
наявності і його не можна придбати.');

```

```

    } else {
        $selected_size = $_POST['size'] ?? '';
        $quantity = max(1, (int)$_POST['quantity']);
        $price = $_POST['price'] ?? 0;

        // Перенаправлення на out.php з параметрами товару
        header("Location:
out.php?product_id={$product_id}&size=" .
urlencode($selected_size) .
"&quantity={$quantity}&price={$price}");
        exit;
    }
}
// Логіка для кнопки "Додати в кошик" (add_to_cart)
elseif (isset($_POST['add_to_cart'])) {
    if ($product['availability'] !== 'В наявності') {
        echo "<script>alert('На жаль, цього товару немає в
наявності і його не можна додати до кошика.');

```

```

        </script>";
    }
}

// Обробка AJAX-запиту для отримання даних кошика
if (isset($_SERVER['HTTP_X_REQUESTED_WITH']) &&
    strtolower($_SERVER['HTTP_X_REQUESTED_WITH']) ===
    'xmlhttprequest' && isset($_GET['action']) && $_GET['action']
    === 'get_cart_data') {
    header('Content-Type: application/json');
    $cartItems = [];
    $session_id = session_id(); // Отримуємо поточний ID сесії
    $user_id = null; // Припустимо, що user_id null, якщо
    користувач не авторизований

    // SQL запит для отримання даних кошика з таблиць cart,
    products та product_sizes
    // Фільтруємо за session_id
    $cartSql = "SELECT
                c.id as cart_item_id,
                c.quantity,
                p.name,
                ps.size,
                ps.price,
                p.image
            FROM
                cart c
            JOIN
                products p ON c.product_id = p.id
            JOIN
                product_sizes ps ON c.product_id =
ps.product_id AND c.size = ps.size
            WHERE
                c.session_id = ?"; // Додаємо фільтр по
    сесії

    $stmt = $conn->prepare($cartSql);
    $stmt->bind_param("s", $session_id); // Прив'язуємо
    session_id як рядок
    $stmt->execute();
    $cartResult = $stmt->get_result();

    if ($cartResult && $cartResult->num_rows > 0) {
        while($row = $cartResult->fetch_assoc()) {
            $cartItems[] = $row;
        }
    }
    echo json_encode($cartItems);
    exit; // Важливо вийти, щоб не відправляти HTML
}

$conn->close();

```

```

?>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1" />
    <title><?= $product ? htmlspecialchars($product['name']) :
"Товар не знайдено" ?></title>
    <link rel="stylesheet" href="style-info.css?v=<?= time()
?>" />
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght
@400;500;700&display=swap" rel="stylesheet">
    <script src="https://cdn.tailwindcss.com"></script>
    <style>
        /* Додаткові стилі для сумісності та покращення
вигляду */
        body {
            font-family: 'Montserrat', sans-serif;
        }
        header h1 {
            font-weight: 600; /* Жирний шрифт для Montserrat
*/
            font-size: 2rem; /* Більший розмір, можна
налаштувати */
            margin-top: 1rem; /* Відступ зверху */
            color: #211A16; /* Змінено колір з 211A16 на
#211A16 */
        }
        /* Стилi для панелi кошика, якщо Tailwind класи
конфліктують або недостатні */
        #slideOutCart {
            transition: right 0.3s ease-in-out;
        }
        .cart-item-image { /* Стилi для зображень товарів у
кошику */
            width: 64px; /* 4rem */
            height: 64px; /* 4rem */
            object-fit: cover;
            border-radius: 4px; /* md */
        }
        /* Додатковий стиль для кнопки "Оформити замовлення"
*/
        .btn-checkout {
            background-color: #661818; /* Заданий колір */
        }
        .btn-checkout:hover {
            background-color: #7b2020; /* Трохи темніший колір
при наведенні */
        }

```

```

/* === ПОЧАТОК ЗМІН ДЛЯ ОДНАКОВОЇ ШИРИНИ КНОПОК === */
.button-container {
    display: flex; /* Вмикаємо Flexbox для контейнера
кнопок */
    gap: 10px; /* Проміжок між кнопками (налаштуйте за
потреби, наприклад, 1rem) */
}

.button-container > a, /* Ціль для тегу <a>,
що огортає кнопку "Купити" */
.button-container > button { /* Ціль для кнопки
"Додати в кошик" */
    flex: 1; /* Обидва елементи (<a> та <button>)
займатимуть однакову частку доступного простору */
}

.buy-now-btn, /* Власне кнопка "Купити" */
.buy-btn { /* Власне кнопка "Додати в кошик" */
    width: 100%; /* Кожна кнопка розтягується на всю
ширину свого батьківського елемента (<a> або самого себе, якщо
вона є flex-елементом) */
    text-align: center; /* Центрування тексту
всередині кнопки */
    /* Для ідеальної візуальної відповідності
переконайтеся, що відступи (padding)
та рамки (border) однакові для обох кнопок,
або додайте 'box-sizing: border-box;'
(Tailwind зазвичай встановлює це глобально).
Приклад однакових відступів (можете
налаштувати під ваш дизайн):
padding-top: 0.75rem;
padding-bottom: 0.75rem;
padding-left: 1rem;
padding-right: 1rem;
*/
}
/* === КІНЕЦЬ ЗМІН ДЛЯ ОДНАКОВОЇ ШИРИНИ КНОПОК === */

</style>
</head>
<body>
<header>
    <div class="header-container">
        <div class="logo">
            
        </div>
        <nav>
            <ul>
                <li><a href="flower.php">Каталог</a></li>
                <li><a href="index.html">Головна</a></li>
            </ul>
        </nav>
    </div>
</header>

```

```

        <li><a href="index.html#about">Про
нає</a></li>
        <li><a
href="index.html#delivery">Доставка</a></li>
    </ul>
</nav>
<div class="cart" id="cartIconTrigger" style="cursor:
pointer;">
    
</div>
<div class="menu-toggle" onclick="toggleMenu()">
    <span></span><span></span><span></span>
</div>
</div>
</header>

<main>
<?php if ($product): ?>
    <div class="product-container">
        <button class="back-btn"
onclick="window.location.href='flower.php'">← Назад</button>

        <div class="product-image">
            " />
        </div>
        <div class="product-info">
            <h1><?= htmlspecialchars($product['name']) ?></h1>
            <p><strong>Артикул:</strong> <?=
htmlspecialchars($product['id']) ?></p>
            <p><strong>Статус:</strong>
                <span style="color: <?=
$product['availability'] == 'В наявності' ? 'green' : 'red';
?>;">
                    <?=
htmlspecialchars($product['availability']) ?>
                </span>
            </p>
            <p><?= htmlspecialchars($product['description'])
?></p>
            <p><strong>Склад:</strong></p>
            <ul>
                <?php foreach (explode(',',
$product['composition']) as $item): ?>
                    <li><?= htmlspecialchars(trim($item))
?></li>
                <?php endforeach; ?>
            </ul>

```

<p>При відправці ми зробимо фото букета, підпишемо листівку та додамо підживку.</p>

```

    <form method="post">
        <p id="price-display">Ціна: <strong><?=
$size[0]['price'] ?? '0.00' ?> грн</strong></p>

        <p><strong>Розміри:</strong></p>
        <div class="size-options">
            <?php foreach ($size as $size): ?>
                <button type="button" class="size-btn"
data-price="<?= $size['price'] ?>" data-size="<?=
$size['size'] ?>"
                    <?=
htmlspecialchars($size['size']) ?>
                </button>
            <?php endforeach; ?>
            <input type="hidden" name="size"
id="selected-size" value="<?= $size[0]['size'] ?? '' ?>" />
            <input type="hidden" name="price"
id="selected-price" value="<?= $size[0]['price'] ?? '0' ?>"
/>
        </div>

        <div class="quantity">
            <button type="button" class="quantity-btn"
onclick="changeQuantity(-1)">-</button>
            <span id="quantity-value">1</span>
            <button type="button" class="quantity-btn"
onclick="changeQuantity(1)">+</button>
            <input type="hidden" name="quantity"
id="quantity-input" value="1" />
        </div>

        <div class="button-container">
            <button type="submit" name="buy_now"
class="buy-now-btn" <?= $product['availability'] !== 'В
наявності' ? 'disabled' : '' ?>>Купити</button>
            <button type="submit" name="add_to_cart"
class="buy-btn" <?= $product['availability'] !== 'В наявності'
? 'disabled' : '' ?>>Додати в кошик</button>
        </div>
    </form>
</div>
</div>

<div id="cart-modal" style="
display: none;
position: fixed;
top: 20%;
left: 50%;
transform: translate(-50%, -50%);
background-color: #FFF5EF;

```

```

        border: 2px solid #661818;
        padding: 20px 30px;
        border-radius: 10px;
        font-size: 1.2em;
        color: #661818;
        z-index: 10000;
        text-align: center;
    ">
        Товар додано до кошика!
    </div>

<?php else: ?>
    <h2 style="text-align:center; color:#4e1010;">Товар не
    знайдено.</h2>
<?php endif; ?>
</main>

<div id="slideOutCart" class="fixed top-0 right-[-100%]
sm:right-[-450px] w-full sm:max-w-md h-full bg-white shadow-xl
z-50 p-6 flex flex-col">
    <div class="flex justify-between items-center pb-4 border-
b border-gray-300">
        <h2 class="text-xl font-semibold text-gray-800">Кошик
        (<span id="cartItemCount">0</span>)</h2>
        <button id="closeCartButton" class="text-gray-600
hover:text-gray-800 text-3xl">&times;</button>
    </div>
    <div id="cartItemsContainer" class="flex-grow overflow-y-
auto py-4 space-y-3">
        <p id="emptyCartMsg" class="text-gray-500 text-center
mt-10">Ваш кошик порожній.</p>
    </div>
    <div class="pt-4 border-t border-gray-300">
        <div class="flex justify-between items-center mb-4">
            <span class="text-lg font-semibold text-gray-
800">Всього:</span>
            <span id="cartTotal" class="text-lg font-semibold
text-gray-800">0.00 грн</span>
        </div>
        <button onclick="window.location.href='out.php';"
class="w-full btn-checkout hover:bg-opacity-90 text-white
font-bold py-3 px-4 rounded-lg transition-colors duration-
150">
            Оформити замовлення
        </button>
    </div>
</div>

<div id="authModal" style="display:none; position:fixed;
top:50%; left:50%; transform:translate(-50%,-50%);
background:#fff; padding:15px; border:1px solid #ccc; border-
radius:10px; box-shadow:0 4px 10px rgba(0,0,0,0.2); z-
index:9999; width:280px; height:320px;">

```

```

        <span onclick="closeModal()" style="position:absolute;
top:5px; right:10px; cursor:pointer; font-size:18px;
color:#999;">&times;</span>
        <iframe src="log.php" style="width:100%; height:100%;
border:none;"></iframe>
    </div>

<footer>
    <div class="footer-container">
        <div class="logo">
            
        </div>
        <div class="info">
            <h3>Інформація</h3>
            <a href="#">Про нас</a>
            <a href="#">Доставка</a>
            <a href="#">Каталог</a>
        </div>
        <div class="contacts">
            <h3>Контакти</h3>
            <p>+38096 666 6666</p>
            <p>вул. Вало́ва, 7, Терно́піль</p>
            <p><a href="mailto:info@bloom-shop.com.ua">info@bloom-shop.com.ua</a></p>
        </div>
    </div>
</footer>

<script>
// Зміна ціни та розміру
const sizeButtons = document.querySelectorAll('.size-btn');
const selectedSizeInput = document.getElementById('selected-size');
const selectedPriceInput = document.getElementById('selected-price');

sizeButtons.forEach(button => {
    button.addEventListener('click', () => {
        // Відміна виділення в інших кнопок
        sizeButtons.forEach(btn =>
            btn.classList.remove('selected'));
        // Вибрана кнопка - активна
        button.classList.add('selected');
        // Оновлення ціни і прихованого інпуту
        document.getElementById('price-display').innerHTML =
'Ціна: <strong>' + button.dataset.price + ' грн</strong>';
        selectedSizeInput.value = button.dataset.size;
        selectedPriceInput.value = button.dataset.price; //
Оновлюємо ціну тут!
    });
});

function changeQuantity(change) {

```

```

        const valueElem = document.getElementById('quantity-
value');
        const inputElem = document.getElementById('quantity-
input');
        let current = parseInt(valueElem.textContent);
        current = Math.max(1, current + change);
        valueElem.textContent = current;
        inputElem.value = current;
    }

    // Відмічаємо за замовчуванням перший розмір як вибраний
    if (sizeButtons.length > 0) {
        sizeButtons[0].classList.add('selected');
    }

    function toggleMenu() {
        document.querySelector('nav').classList.toggle('active');
    }

    const authModal = document.getElementById('authModal');
    if (authModal) {
        setTimeout(function() {
            // Тут має бути перевірка, чи користувач авторизований
            // authModal.style.display = 'block'; // Закоментуйте,
якщо не потрібно автоматичне відображення
        }, 10000);
    }

    function closeModal() {
        if (authModal) {
            authModal.style.display = 'none';
        }
    }

    const cartIcon = document.getElementById('cartIconTrigger');
    const slideOutCart = document.getElementById('slideOutCart');
    const closeCartButton =
document.getElementById('closeCartButton');
    const cartItemsContainer =
document.getElementById('cartItemsContainer');
    const cartItemCount =
document.getElementById('cartItemCount');
    const cartTotal = document.getElementById('cartTotal');
    const emptyCartMsg = document.getElementById('emptyCartMsg');

    function openCart() {
        if (slideOutCart) {
            slideOutCart.classList.remove('right-[-100%]',
'sm:right-[-450px]');
            slideOutCart.classList.add('right-0');
            loadCartData();
        }
    }

```

```

}

function closeCart() {
  if (slideOutCart) {
    slideOutCart.classList.remove('right-0');
    slideOutCart.classList.add('right-[-100%]', 'sm:right-[-450px]');
  }
}

async function loadCartData() {
  try {
    // Отримуємо поточний шлях до файлу для формування
    коректного URL
    const currentScriptPath = window.location.pathname;
    const url = new URL(currentScriptPath,
window.location.origin);
    url.searchParams.set('action', 'get_cart_data');

    const response = await fetch(url.toString(), {
      headers: {
        'X-Requested-With': 'XMLHttpRequest'
      }
    });
    if (!response.ok) {
      throw new Error(`HTTP error! status:
${response.status}`);
    }
    const items = await response.json();

    renderCartItems(items);
  } catch (error) {
    console.error('Помилка завантаження даних кошика:',
error);
    if (cartItemsContainer) {
      cartItemsContainer.innerHTML = '<p class="text-
red-500 text-center">Не вдалося завантажити кошик.</p>';
    }
    if (emptyCartMsg) emptyCartMsg.style.display =
'none';
  }
}

function renderCartItems(items) {
  if (!cartItemsContainer || !cartItemCount || !cartTotal ||
!emptyCartMsg) return;

  cartItemsContainer.innerHTML = ''; // Очистити попередні
товари
  let totalAmount = 0;
  let itemCount = 0;

  if (items && items.length > 0) {

```

```

        emptyCartMsg.style.display = 'none';
        items.forEach(item => {
            const itemElement = document.createElement('div');
            itemElement.className = 'flex items-center space-
x-3 p-3 bg-gray-50 rounded-lg shadow-sm';

            const imageName = item.image ?
`img2/${item.image}` :
'https://placeholder.co/64x64/E0E0E0/B0B0B0?text=N/A';
            const itemName = item.name || 'Без назви';
            const itemSize = item.size || 'N/A';
            const itemQuantity = parseInt(item.quantity) || 0;
            const itemPrice = parseFloat(item.price) || 0;

            itemElement.innerHTML = `
                
                <div class="flex-grow">
                    <h3 class="text-sm font-semibold text-
gray-700">${htmlspecialchars(itemName)}</h3>
                    <p class="text-xs text-gray-500">Розмір:
${htmlspecialchars(itemSize)}</p>
                    <p class="text-xs text-gray-
500">Кількість: ${itemQuantity}</p>
                </div>
                <div class="text-right">
                    <p class="text-sm font-semibold text-gray-
700">${itemPrice.toFixed(2)} грн</p>
                </div>
            `;
            cartItemsContainer.appendChild(itemElement);
            totalAmount += itemPrice * itemQuantity;
            itemCount += itemQuantity;
        });
    } else {
        emptyCartMsg.style.display = 'block';
    }

    cartItemCount.textContent = itemCount;
    cartTotal.textContent = `${totalAmount.toFixed(2)} грн`;
}

function htmlspecialchars(str) {
    if (typeof str !== 'string') return '';
    return str.replace(/([<>"']/g, function (match) {
        const replacements = {
            '&': '&amp;',
            '<': '&lt;',
            '>': '&gt;',
            '"': '&quot;',
            "'": '&#39;';
        }
    }

```

```

        };
        return replacements[match];
    });
}

if (cartIcon) {
    cartIcon.addEventListener('click', openCart);
}
if (closeCartButton) {
    closeCartButton.addEventListener('click', closeCart);
}

document.addEventListener('click', function(event) {
    if (slideOutCart && cartIcon) {
        const isClickInsideCart =
slideOutCart.contains(event.target);
        const isClickOnCartIcon =
cartIcon.contains(event.target);
        if (!isClickInsideCart && !isClickOnCartIcon &&
!slideOutCart.classList.contains('right-[-100%]')) {
            // closeCart(); // Розкоментуйте, якщо хочете таку
поведінку
        }
    }
});

</script>

</body>
</html>

```

Сторінка оформлення замовлення інтернет-магазину квітів «out.html»

```

<?php
session_start();

// Підключення до бази даних
$servername = "fdb1029.awardspace.net";
$username = "4532764_bloom";
$password = "olesyaa1020";
$dbname = "4532764_bloom";

$conn = new mysqli($servername, $username, $password,
$dbname);
if ($conn->connect_error) {
    die("Помилка з'єднання: " . $conn->connect_error);
}

if (!$conn->set_charset("utf8mb4")) {
    // Обробка помилки, якщо потрібно
}

$orderItems = [];
$totalOrderAmount = 0;
$orderStatus = null; // Додаємо змінну для статусу замовлення
$customer_email_for_message = ''; // Initialize to avoid
undefined variable notice

// Перевіряємо, чи був зроблений прямий перехід з кнопки
"Купити"
if (isset($_GET['product_id']) && isset($_GET['size']) &&
isset($_GET['quantity']) && isset($_GET['price'])) {
    $product_id = (int)$_GET['product_id'];
    $selected_size = $_GET['size'];
    $quantity = (int)$_GET['quantity'];
    $price = (float)$_GET['price'];

    // Отримуємо додаткову інформацію про товар з бази даних
    $sql = "SELECT name, image FROM products WHERE id = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("i", $product_id);
    $stmt->execute();
    $result = $stmt->get_result();
    $product_info = $result->fetch_assoc();

    if ($product_info) {
        $orderItems[] = [
            'product_id' => $product_id,
            'name' => $product_info['name'],
            'size' => $selected_size,
            'quantity' => $quantity,
            'price' => $price,

```

```

        'image' => $product_info['image']
    ];
    $totalOrderAmount = $price * $quantity;
}
} else {
    // Якщо не прямий перехід "Купити", завантажуюмо товари з
    кошика
    $session_id = session_id();

    $cartSql = "SELECT
                c.id as cart_item_id,
                c.quantity,
                p.name,
                ps.size,
                ps.price,
                p.image
            FROM
                cart c
            JOIN
                products p ON c.product_id = p.id
            JOIN
                product_sizes ps ON c.product_id =
ps.product_id AND c.size = ps.size
            WHERE
                c.session_id = ?";

    $stmt = $conn->prepare($cartSql);
    $stmt->bind_param("s", $session_id);
    $stmt->execute();
    $cartResult = $stmt->get_result();

    if ($cartResult && $cartResult->num_rows > 0) {
        while($row = $cartResult->fetch_assoc()) {
            $orderItems[] = $row;
            $totalOrderAmount += $row['price'] *
$row['quantity'];
        }
    }
}

// Обробка форми оформлення замовлення
if ($_SERVER['REQUEST_METHOD'] === 'POST' &&
isset($_POST['place_order'])) {
    $user_name = htmlspecialchars(trim($_POST['name'] ?? ''));
    $user_phone = htmlspecialchars(trim($_POST['phone'] ??
''));
    $user_email = htmlspecialchars(trim($_POST['email'] ??
''));
    $user_address = htmlspecialchars(trim($_POST['address'] ??
''));
    $payment_method = "Оплата при отриманні"; // Завжди
"Наліком"

```

```

        $user_email_for_message = $user_email; // Store for
        success message

        if (empty($user_name) || empty($user_phone) ||
        empty($user_email) || empty($user_address)) {
            $orderMessage = "Будь ласка, заповніть усі обов'язкові
        поля: ім'я, телефон, email та адресу.";
            $orderStatus = "error";
        } elseif (!filter_var($user_email, FILTER_VALIDATE_EMAIL))
        {
            $orderMessage = "Будь ласка, введіть коректний
        email.";
            $orderStatus = "error";
        } elseif (empty($orderItems)) {
            $orderMessage = "Немає товарів для оформлення
        замовлення.";
            $orderStatus = "error";
        } else {
            // Початок транзакції
            $conn->begin_transaction();
            try {
                $insertOrderSql = "INSERT INTO orders (user_email,
        user_phone, user_name, user_address, payment_method, price)
                VALUES (?, ?, ?, ?, ?, ?)";
                $stmtOrder = $conn->prepare($insertOrderSql);
                $stmtOrder->bind_param("sssss", $user_email,
        $user_phone, $user_name, $user_address, $payment_method,
        $totalOrderAmount); // Use $totalOrderAmount here
                $stmtOrder->execute();
                $order_id = $conn->insert_id;

                // Очищення кошика після успішного замовлення,
        якщо товари були з кошика
                if (!isset($_GET['product_id'])) {
                    $deleteCartSql = "DELETE FROM cart WHERE
        session_id = ?";
                    $stmtDeleteCart = $conn-
        >prepare($deleteCartSql);
                    $session_id_to_clear = session_id(); // ensure
        session_id is available
                    $stmtDeleteCart->bind_param("s",
        $session_id_to_clear);
                    $stmtDeleteCart->execute();
                }

                $conn->commit();
                $orderMessage = "Ваше замовлення №" . $order_id .
        " успішно оформлено! Підтвердження буде надіслано на вашу
        електронну пошту: " .
        htmlspecialchars($customer_email_for_message) . ". Ми
        зв'яжемося з вами найближчим часом.";
                $orderStatus = "success"; // Встановлюємо статус
        успіху
    
```

```

        $orderItems = []; // Очищаємо товари, щоб вони не
відображались після оформлення
        $totalOrderAmount = 0;
    } catch (mysqli_sql_exception $exception) {
        $conn->rollback();
        $orderMessage = "Виникла помилка під час
оформлення замовлення: " . $exception->getMessage();
        $orderStatus = "error";
    }
}

$conn->close();
?>

<!DOCTYPE html>
<html lang="uk">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Оформлення замовлення - Bloom</title>
    <link rel="stylesheet" href="style-info.css?v=<?= time()
?>">
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght
@400;500;700&display=swap" rel="stylesheet">
    <script src="https://cdn.tailwindcss.com"></script>
    <style>
        /* Ваші існуючі стилі */
        body {
            font-family: 'Montserrat', sans-serif;
            background-color: #FFF5ED;
            color: #211A16;
            line-height: 1.6;
        }

        header {
            background-color: #FFF5EF;
            padding: 1rem 0;
            border-bottom: 1px solid #661818;
            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
            position: relative;
            /* Додано для коректного позиціонування меню-
бургера */
        }

        .header-container {
            display: flex;
            justify-content: space-between;
            align-items: center;
            max-width: 1200px;

```

```

        margin: 0 auto;
        padding: 0 1rem;
    }

    .header-container .logo img {
        height: 50px;
        width: auto;
    }

    .header-container nav ul {
        list-style: none;
        margin: 0;
        padding: 0;
        display: flex;
        gap: 20px;
    }

    .header-container nav ul li a {
        color: #211A16;
        text-decoration: none;
        padding: 0.5rem 1rem;
        transition: color 0.3s ease;
        white-space: nowrap;
        font-size: 16px;
    }

    .header-container nav ul li a:hover {
        color: #661818;
    }

    .header-container nav ul li.active a {
        font-weight: bold;
        color: #4e1010;
    }

    .header-container .cart img {
        width: 20px;
        height: auto;
    }

    /* Меню-гамбургер для мобільних (глобальні стилі) */
    /* Меню-гамбургер для мобільних */
    .menu-toggle {
        display: none; /* Приховано за замовчуванням на десктопі */
        /*
        cursor: pointer;
        flex-direction: column;
        justify-content: space-around;
        width: 30px;
        height: 25px;
        position: absolute;
        right: 20px;
        top: 50%;
        */
    }

```

```

        transform: translateY(-50%);
    }

    .menu-toggle span {
        display: block;
        height: 3px;
        width: 100%;
        background-color: #333;
        border-radius: 2px;
        transition: all 0.3s ease-in-out;
    }

    /* Навігаційне меню - Десктоп */
    #navMenu ul {
        list-style: none;
        margin: 0;
        padding: 0;
        display: flex;
    }

    #navMenu ul li {
        margin-left: 20px;
    }

    #navMenu ul li a {
        text-decoration: none;
        color: #333;
        font-weight: 500;
        transition: color 0.3s ease;
    }

    #navMenu ul li a:hover,
    #navMenu ul li.active a {
        color: #661818;
    }

    /* Кнопка закриття меню (глобальні стилі) */
    .close-menu-btn {
        position: absolute;
        top: 10px;
        right: 10px;
        font-size: 24px;
        background: none;
        border: none;
        color: #fff;
        cursor: pointer;
        z-index: 101;
        /* Над nav та перемикачем */
    }

    .back-btn {
        display: inline-block;
        position: relative;
        text-decoration: none;

```

```

        font-size: 16px;
        color: #4e1010;
        background: transparent;
        border: 1px solid #4e1010;
        padding: 8px 15px;
        border-radius: 5px;
        margin-bottom: 30px;
        transition: background-color 0.3s ease, color 0.3s
ease;
    }

    .back-btn:hover {
        background-color: #4e1010;
        color: #FFF5ED;
    }

    .main-content {
        max-width: 900px;
        margin: 2rem auto;
        padding: 2rem;
        background-color: #fff;
        border-radius: 8px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }

    .order-summary,
    .customer-details {
        margin-bottom: 2rem;
        border: 1px solid #e0e0e0;
        border-radius: 8px;
        padding: 1.5rem;
        background-color: #f9f9f9;
    }

    .order-summary h2,
    .customer-details h2 {
        color: #661818;
        font-size: 1.8rem;
        margin-bottom: 1.5rem;
        text-align: center;
    }

    .order-item {
        display: flex;
        align-items: center;
        margin-bottom: 1rem;
        padding-bottom: 1rem;
        border-bottom: 1px dashed #e0e0e0;
    }

    .order-item:last-child {
        border-bottom: none;
    }

```

```
.order-item img {
    width: 80px;
    height: 80px;
    object-fit: cover;
    border-radius: 4px;
    margin-right: 1rem;
}

.item-details {
    flex-grow: 1;
}

.item-details h3 {
    font-size: 1.1rem;
    font-weight: 600;
    color: #211A16;
}

.item-details p {
    font-size: 0.9rem;
    color: #555;
}

.item-price {
    font-weight: 700;
    color: #661818;
}

.order-total {
    text-align: right;
    font-size: 1.5rem;
    font-weight: 700;
    color: #661818;
    margin-top: 1.5rem;
}

.form-group {
    margin-bottom: 1.2rem;
}

.form-group label {
    display: block;
    margin-bottom: 0.5rem;
    font-weight: 500;
    color: #211A16;
}

.form-group input[type="text"],
.form-group input[type="email"],
.form-group textarea {
    width: 100%;
    padding: 0.8rem;
}
```

```

        border: 1px solid #ccc;
        border-radius: 5px;
        font-size: 1rem;
        color: #211A16;
        background-color: #fcfcfc;
        box-sizing: border-box;
        /* Важливо для коректної ширини */
    }

    .form-group textarea {
        resize: vertical;
        min-height: 100px;
    }

    .form-group input[type="text"]:focus,
    .form-group input[type="email"]:focus,
    .form-group textarea:focus {
        border-color: #661818;
        outline: none;
        box-shadow: 0 0 5px rgba(102, 24, 24, 0.3);
    }

    .place-order-btn {
        display: block;
        width: 100%;
        padding: 1rem;
        background-color: #661818;
        color: #fff;
        border: none;
        border-radius: 5px;
        font-size: 1.2rem;
        font-weight: 700;
        cursor: pointer;
        transition: background-color 0.3s ease;
    }

    .place-order-btn:hover {
        background-color: #7b2020;
    }

    .message-box {
        padding: 1rem;
        margin-bottom: 1rem;
        border-radius: 5px;
        text-align: center;
        font-weight: 500;
    }

    .message-box.success {
        background-color: #d4edda;
        color: #155724;
        border: 1px solid #badbcc;
    }

```

```

.message-box.error {
    background-color: #f8d7da;
    color: #721c24;
    border: 1px solid #f5c6cb;
}

footer {
    background-color: #FFF5EF;
    /* Consistent with header style */
    padding: 2rem 0;
    margin-top: 2rem;
    border-top: 1px solid #e0e0e0;
    /* Consistent with previous footer style */
    text-align: center;
    color: #555;
}

.footer-container {
    display: flex;
    justify-content: space-around;
    align-items: flex-start;
    max-width: 1200px;
    margin: 0 auto;
    flex-wrap: wrap;
}

.footer-container>div {
    flex-basis: 30%;
    /* Adjust as needed, this is from original */
    margin-bottom: 1rem;
}

.footer-container .logo img {
    height: 60px;
    /* As per original */
    width: auto;
}

.footer-container h3 {
    color: #211A16;
    margin-bottom: 1rem;
    font-size: 1.1rem;
}

.footer-container a,
.footer-container p {
    display: block;
    color: #555;
    text-decoration: none;
    margin-bottom: 0.5rem;
    font-size: 0.9rem;
}

```

```

.footer-container a:hover {
    color: #661818;
}

/* Mobile specific styles from provided CSS (ensure
they work with new header) */
@media (max-width: 768px) {
    .header-container {
        flex-direction: row;
        /* Stack header items */
        align-items: center;
        padding: 10px;
        justify-content: space-between;
    }

    .header-container .logo {
        margin-bottom: 0;
    }

    .header-container .logo img {
        height: 40px;
        /* Adjust for mobile if needed */
    }

    .header-container nav {
        width: 100%;
        margin-left: 0;
        /* Reset margin */
        margin-bottom: 0;
        /* Reset margin */
    }

    .header-container nav ul {
        justify-content: center;
        /* Center nav items */
        flex-wrap: wrap;
        /* Allow wrapping if too many items */
    }

    .header-container .cart {
        margin-left: 0;
        /* Reset margin */
        margin-top: 0;
        /* Reset margin */
        right: 15%;
    }

    .header-container .menu-toggle {
        display: flex;
        /* Assuming menu-toggle from style-info.css
takes over for smaller screens if needed */
    }
}

```

```

        .main-content {
            margin: 1rem;
            padding: 1rem;
        }

        .order-item {
            flex-direction: column;
            align-items: flex-start;
        }

        .order-item img {
            margin-bottom: 0.5rem;
        }

        .footer-container>div {
            flex-basis: 100%;
            text-align: center;
        }
    }

    /* Styles for menu-toggle from flower.php, adapted for
out.php */
    @media (max-width: 480px) {
        /* Hides original nav, menu-toggle shows it */
        .header-container nav {
            /* This will hide the nav defined in the new
header */
            display: none;
        }

        /* Show the menu toggle on small screens */
        .header-container .menu-toggle {
            display: flex;
            flex-direction: column;
            justify-content: space-around;
            width: 30px;
            height: 25px; /* Adjusted for better visual */
            cursor: pointer;
            position: absolute; /* Position it relative to
header-container */
            right: 1rem;
            top: 50%;
            transform: translateY(-50%);
        }

        .menu-toggle span {
            height: 3px;
            background-color: #4e1010;
            border-radius: 2px;
            display: block; /* ensure spans are block */
        }
    }

```

```

/* Mobile Navigation Menu */
#navMenu {
  display: none; /* Initially hidden */
  position: fixed;
  top: 0;
  left: -100%; /* Hidden off-screen to the left
*/
  width: 80%;
  max-width: 300px;
  height: 100%;
  background-color: #333;
  z-index: 1000; /* Ensure it's on top */
  transition: left 0.3s ease-in-out; /* Smooth
slide-in */
  padding-top: 60px; /* Space for close button
*/
  overflow-y: auto;
}

#navMenu.active {
  left: 0; /* Slide in when active */
  display: block; /* Make sure it's visible */
}

#navMenu ul {
  flex-direction: column; /* Vertical menu items
*/
  align-items: flex-start; /* Align items to the
left */
  padding: 0 20px; /* Padding for menu items */
  height: auto; /* Allow content to dictate
height */
}

#navMenu ul li {
  margin: 10px 0; /* Spacing between menu items
*/
}

#navMenu ul li a {
  color: #fff;
  font-size: 1.2rem;
  display: block;
  width: 100%;
  padding: 10px 0;
}

/* Close button for mobile menu */
.close-menu-btn {
  position: absolute;
  top: 15px; /* Adjust as needed */
  right: 15px; /* Adjust as needed */
  font-size: 28px; /* Larger for touch */
}

```

```

        background: none;
        border: none;
        color: #fff; /* White color for visibility on
dark menu */
        cursor: pointer;
        z-index: 1001; /* Above the menu */
    }
}
</style>
</head>

<body>
    <header>
        <div class="header-container">
            <a href="flower.php" class="logo">
                
            </a>
            <nav id="navMenu">
                <button class="close-menu-
btn">&times;</button>
                <ul>
                    <li><a href="flower.php">Каталог</a></li>
                    <li><a href="index.html">Головна</a></li>
                    <li><a href="index.html#about">Про
нас</a></li>
                    <li><a
href="index.html#delivery">Доставка</a></li>
                </ul>
            </nav>
            <div class="menu-toggle">
                <span></span>
                <span></span>
                <span></span>
            </div>
            <div class="cart" id="cartIconTrigger"
style="cursor: pointer;">
                
            </div>
        </div>
    </header>

    <main class="main-content">

        <button class="back-btn"
onclick="window.location.href='flower.php'">← Назад</button>
        <h1 class="text-3xl font-bold mb-6 text-center text-
[#661818]">Оформлення замовлення</h1>

```

```

        <?php if (isset($orderMessage)): ?>
            <div class="message-box <?=
htmlspecialchars($orderStatus ?? '') ?>">
                <?= $orderMessage // This now includes email
if successful ?>
            </div>
        <?php endif; ?>

        <?php if ($orderStatus !== "success" &&
!empty($orderItems)): // Показуємо форму, якщо замовлення ще
не оформлено успішно і є товари ?>
            <div class="order-summary">
                <h2>Ваше замовлення</h2>
                <div class="items-list">
                    <?php foreach ($orderItems as $item): ?>
                        <div class="order-item">
                            "
                            <div class="item-details">
                                <h3><?=
htmlspecialchars($item['name'] ?? 'Назва товару') ?></h3>
                                <p>Розмір: <?=
htmlspecialchars($item['size'] ?? 'N/A') ?></p>
                                <p>Кількість: <?=
htmlspecialchars($item['quantity'] ?? 0) ?></p>
                            </div>
                            <div class="item-price">
                                <?=
number_format((float)($item['price'] ?? 0) *
(int)($item['quantity'] ?? 0), 2) ?> грн
                            </div>
                        </div>
                    <?php endforeach; ?>
                </div>
                <div class="order-total">
                    Загальна сума: <?=
number_format((float)$totalOrderAmount, 2) ?> грн
                </div>
            </div>

            <div class="customer-details">
                <h2>Контактні дані</h2>
                <form method="post">
                    <div class="form-group">
                        <label for="name">Ваше ім'я <span
style="color: red;">*</span></label>
                        <input type="text" id="name"
name="name" required value="<?=
htmlspecialchars($_POST['name'] ?? '') ?>">
                    </div>
                    <div class="form-group">

```

```

        <label for="phone">Номер телефону
<span style="color: red;">*</span></label>
        <input type="text" id="phone"
name="phone" required value="<?=
htmlspecialchars($_POST['phone'] ?? '') ?>">
        </div>
        <div class="form-group">
            <label for="email">Електронна пошта
<span style="color: red;">*</span></label>
            <input type="email" id="email"
name="email" required value="<?=
htmlspecialchars($_POST['email'] ?? '') ?>">
            </div>
            <div class="form-group">
                <label for="address">Адреса доставки
<span style="color: red;">*</span></label>
                <input type="text" id="address"
name="address" required value="<?=
htmlspecialchars($_POST['address'] ?? '') ?>">
                </div>
                <div class="form-group">
                    <label for="payment_method">Метод
оплати</label>
                    <input type="text" id="payment_method"
name="payment_method" value="Оплата при отриманні" readonly
                    class="bg-gray-100 cursor-not-
allowed">
                    <p class="text-sm text-gray-500 mt-
1">Наразі доступна лише оплата при отриманні.</p>
                </div>
                <div class="form-group">
                    <label for="comments">Коментарі до
замовлення</label>
                    <textarea id="comments"
name="comments" rows="4"><?=
htmlspecialchars($_POST['comments'] ?? '') ?></textarea>
                </div>
                <button type="submit" name="place_order"
class="place-order-btn">Підтвердити замовлення</button>
            </form>
        </div>
        <?php elseif ($orderStatus === "success"): // Якщо
замовлення успішно оформлено, показуємо тільки повідомлення
(додаткове) ?>
            <p class="text-center text-gray-700 text-lg mt-
4">Дякуємо за ваше замовлення! Очікуйте на дзвінок менеджера
для
                підтвердження деталей.</p>
            <p class="text-center mt-4"><a href="flower.php"
class="text-[#661818] hover:underline">Повернутися до
                каталогу</a></p>
        <?php else: // Якщо кошик порожній і немає
повідомлення про успіх/помилку обробки ?>

```

```

        <p class="text-center text-gray-700 text-lg">Ваш
кошик порожній або виникла помилка. Будь ласка, додайте
        товари, щоб оформити замовлення, або спробуйте
пізніше.</p>
        <p class="text-center mt-4"><a href="flower.php"
class="text-[#661818] hover:underline">Перейти до
        каталогу</a></p>
        <?php endif; ?>
    </main>

    <div id="slideOutCart"
        class="fixed top-0 right-[-100%] sm:right-[-450px] w-
full sm:max-w-md h-full bg-white shadow-xl z-50 p-6 flex flex-
col transition-all duration-300 ease-in-out">
        <div class="flex justify-between items-center pb-4
border-b border-gray-300">
            <h2 class="text-xl font-semibold text-gray-
800">Кошик (<span id="cartItemCount">0</span></h2>
            <button id="closeCartButton" class="text-gray-600
hover:text-gray-800 text-3xl">&times;</button>
        </div>
        <div id="cartItemsContainer" class="flex-grow
overflow-y-auto py-4 space-y-3">
            <p id="emptyCartMsg" class="text-gray-500 text-
center mt-10">Ваш кошик порожній.</p>
        </div>
        <div class="pt-4 border-t border-gray-300">
            <div class="flex justify-between items-center mb-
4">
                <span class="text-lg font-semibold text-gray-
800">Всього:</span>
                <span id="cartTotal" class="text-lg font-
semibold text-gray-800">0.00 грн</span>
            </div>
            <button onclick="window.location.href='out.php';"
                class="w-full bg-[#661818] hover:bg-[#7b2020]
text-white font-bold py-3 px-4 rounded-lg transition-colors
duration-150 <?php if (empty($orderItems) && $orderStatus !==
'success') echo 'opacity-50 cursor-not-allowed'; ?>"
                <?php if (empty($orderItems) && $orderStatus
!== 'success') echo 'disabled'; ?>>
                Оформити замовлення
            </button>
        </div>
    </div>

    <div id="authModal"
        style="display:none; position:fixed; top:50%;
left:50%; transform:translate(-50%,-50%); background:#fff;
padding:15px; border:1px solid #ccc; border-radius:10px; box-
shadow:0 4px 10px rgba(0,0,0,0.2); z-index:9999; width:280px;
height:320px;">
        <span onclick="closeModal()"

```

```

        style="position:absolute; top:5px; right:10px;
        cursor:pointer; font-size:18px; color:#999;">&times;</span>
        <iframe src="log.php" style="width:100%; height:100%;
        border:none;"></iframe>
    </div>

    <footer>
        <div class="footer-container">
            <div class="logo">
                
            </div>
            <div class="info">
                <h3>Інформація</h3>
                <a href="index.html#about">Про нас</a> <a
href="index.html#delivery">Доставка</a> <a
                href="flower.php">Каталог</a>
            </div>
            <div class="contacts">
                <h3>Контакти</h3>
                <p>+38096 666 6666</p>
                <p>вул. Вало́ва, 7, Терно́піль</p>
                <p><a href="mailto:info@bloom-
shop.com.ua">info@bloom-shop.com.ua</a></p>
            </div>
        </div>
    </footer>

    <script>
        // Функція для перемикання меню
        function toggleMenu() {
            const navMenu =
document.getElementById('navMenu');
            navMenu.classList.toggle('active');
        }

        // Отримуємо посилання на кнопку закриття меню
        const closeMenuBtn = document.querySelector('.close-
menu-btn');

        // Отримуємо посилання на меню
        const navMenu = document.getElementById('navMenu');

        // Отримуємо посилання на кнопку menu-toggle
        const menuToggle = document.querySelector('.menu-
toggle');

        // Додаємо обробник події для кнопки закриття
        if (closeMenuBtn && navMenu) {
            closeMenuBtn.addEventListener('click', function()
{

```

```

        navMenu.classList.remove('active'); //
Закриваємо меню
    });
}

// Додаємо обробник події для menu-toggle
if (menuToggle && navMenu) {
    menuToggle.addEventListener('click', toggleMenu);
}

// Логіка відображення модального вікна авторизації
const authModal =
document.getElementById('authModal');

function closeModal() {
    if (authModal) {
        authModal.style.display = 'none';
    }
}

// --- JavaScript для кошика (оновлений для out.php) -
--
    const cartIcon =
document.getElementById('cartIconTrigger');
    const slideOutCart =
document.getElementById('slideOutCart');
    const closeCartButton =
document.getElementById('closeCartButton');
    const cartItemsContainer =
document.getElementById('cartItemsContainer');
    const cartItemCount =
document.getElementById('cartItemCount');
    const cartTotal =
document.getElementById('cartTotal');
    const emptyCartMsg =
document.getElementById('emptyCartMsg');

    function openCart() {
        if (slideOutCart) {
            slideOutCart.classList.remove('right-[-100%]',
'sm:right-[-450px]');
            slideOutCart.classList.add('right-0');
            loadCartData(); // Завантажуємо дані кошика
при відкритті
        }
    }

    function closeCart() {
        if (slideOutCart) {
            slideOutCart.classList.remove('right-0');
            slideOutCart.classList.add('right-[-100%]',
'sm:right-[-450px]');
        }
    }

```

```

    }

    async function loadCartData() {
      try {
        // Determine the correct path to info.php.
        This assumes info.php is in the same directory.
        const basePath =
window.location.pathname.substring(0,
window.location.pathname.lastIndexOf("/") + 1);
        const response = await
fetch(`${basePath}info.php?action=get_cart_data`, {
          // Adjusted path if needed
          headers: {
            'X-Requested-With': 'XMLHttpRequest'
          }
        });
        if (!response.ok) {
          throw new Error(`HTTP error! status:
${response.status}`);
        }
        const items = await response.json();

        renderCartItems(items);
      } catch (error) {
        console.error('Помилка завантаження даних
кошика:', error);
        if (cartItemsContainer) {
          cartItemsContainer.innerHTML = '<p
class="text-red-500 text-center">Не вдалося завантажити кошик.
Спробуйте оновити сторінку.</p>';
        }
        if (emptyCartMsg) emptyCartMsg.style.display =
'none';
      }
    }

    function renderCartItems(items) {
      if (!cartItemsContainer || !cartItemCount ||
!cartTotal || !emptyCartMsg) return;

      cartItemsContainer.innerHTML = '';
      let totalAmount = 0;
      let itemCount = 0;

      if (items && items.length > 0) {
        emptyCartMsg.style.display = 'none';
        items.forEach(item => {
          const itemElement =
document.createElement('div');
          itemElement.className = 'flex items-center
space-x-3 p-3 bg-gray-50 rounded-lg shadow-sm';

```

```

        const imageName = item.image ?
`img2/${item.image}` :
'https://placeholder.co/64x64/E0E0E0/B0B0B0?text=N/A';
        const itemName = item.name || 'Без назви';
        const itemSize = item.size || 'N/A';
        const itemQuantity =
parseInt(item.quantity) || 0;
        const itemPrice = parseFloat(item.price)
|| 0;

        itemElement.innerHTML = `
            
            <div class="flex-grow">
                <h3 class="text-sm font-semibold
text-gray-700">${htmlspecialchars(itemName)}</h3>
                <p class="text-xs text-gray-
500">Розмір: ${htmlspecialchars(itemSize)}</p>
                <p class="text-xs text-gray-
500">Кількість: ${itemQuantity}</p>
            </div>
            <div class="text-right">
                <p class="text-sm font-semibold
text-gray-700">${(itemPrice * itemQuantity).toFixed(2)}
грн</p>
            </div>`;

        cartItemsContainer.appendChild(itemElement);
        totalAmount += itemPrice * itemQuantity;
        itemCount += itemQuantity;
    });
} else {
    emptyCartMsg.style.display = 'block';
}

cartItemCount.textContent = itemCount;
cartTotal.textContent = `${totalAmount.toFixed(2)}
грн`;

    // Disable checkout button in slide-out cart if
    cart is empty
    const checkoutButtonSlideOut =
document.querySelector('#slideOutCart
button[onclick*="out.php"]');
    if (checkoutButtonSlideOut) {
        if (itemCount === 0) {

            checkoutButtonSlideOut.classList.add('opacity-50', 'cursor-
not-allowed');

            checkoutButtonSlideOut.disabled = true;

```

```

        } else {

checkoutButtonSlideOut.classList.remove('opacity-50', 'cursor-
not-allowed');
            checkoutButtonSlideOut.disabled = false;
        }
    }

    function htmlspecialchars(str) {
        if (typeof str !== 'string') return String(str);
// Ensure it's a string
        const map = {
            '&': '&amp;',
            '<': '&lt;',
            '>': '&gt;',
            '"': '&quot;',
            "'": '&#39;';
        };
        return str.replace(/([<>"']/g, function(m) {
            return map[m];
        });
    }

    if (cartIcon) {
        cartIcon.addEventListener('click', openCart);
    }
    if (closeCartButton) {
        closeCartButton.addEventListener('click',
closeCart);
    }

    document.addEventListener('click', function(event) {
        if (slideOutCart && cartIcon) {
            const isClickInsideCart =
slideOutCart.contains(event.target);
            const isClickOnCartIcon =
cartIcon.contains(event.target) || (cartIcon.parentElement &&
cartIcon.parentElement.contains(event.target));
            if (!isClickInsideCart && !isClickOnCartIcon
&& !slideOutCart.classList.contains('right-[-100%]')) {
                // closeCart(); // You can uncomment this
if you want click-outside-to-close behavior
            }
        }
    });

    // Initial load of cart data when the page loads, for
the slide-out cart icon count.
    // This part is tricky as the main page (out.php)
already loads order items via PHP.
    // We only need to update the small cart icon counter,
perhaps.

```

```

        // For now, `loadCartData` is called when cart is
opened.
        // If you want the cart count (in header) to be
accurate on page load without opening cart,
        // you might need another small AJAX call or pass
count from PHP.
        // However, `loadCartData` itself gets full cart data.

        // To initialize cart count on page load for the
header icon (if needed separately from PHP part)
        async function updateHeaderCartCount() {
            try {
                const basePath =
window.location.pathname.substring(0,
window.location.pathname.lastIndexOf("/") + 1);
                const response = await
fetch(`${basePath}info.php?action=get_cart_data`, {
                    headers: {
                        'X-Requested-With': 'XMLHttpRequest'
                    }
                });
                if (!response.ok) throw new Error('Failed to
fetch cart count');
                const items = await response.json();
                let itemCount = 0;
                if (items && items.length > 0) {
                    items.forEach(item => itemCount +=
(parseInt(item.quantity) || 0));
                }
                if (cartItemCount) cartItemCount.textContent =
itemCount; // Updates count in slide-out
                // If you have a separate counter in the fixed
header, update it here too.
                // Example: const headerCartBadge =
document.getElementById('headerCartBadge');
                // if(headerCartBadge)
headerCartBadge.textContent = itemCount;

            } catch (error) {
                console.warn("Could not update header cart
count on page load:", error);
                if (cartItemCount) cartItemCount.textContent =
'0'; // Default on error
            }
        }
    </script>
</body>

</html>

```

Сторінка адміністратора для перегляду замовлень у інтернет-магазині

квітів «order.html»

```

<?php
$servername = "fdb1029.awardspace.net";
$username = "4532764_bloom";
$password = "olesyaa1020";
$dbname = "4532764_bloom";

$conn = new mysqli($servername, $username, $password, $dbname);
$conn->set_charset("utf8mb4");

$sql = "SELECT
        o.id,
        o.product_id,
        o.name AS order_name,    -- назва з orders
        p.image AS product_image, -- фото з products
        o.user_name,
        o.user_email,
        o.user_phone,
        o.user_address,
        o.payment_method,
        o.price
    FROM orders o
    LEFT JOIN products p ON o.product_id = p.id
    ORDER BY o.id DESC";

$result = $conn->query($sql);
?>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Список замовлень</title>
    <style>
        body {
            background: #FFF5ED;
            font-family: sans-serif;
            padding: 20px;
        }
        table {
            width: 100%;
            border-collapse: collapse;
            table-layout: fixed;
            background: white;
        }
        th, td {
            padding: 12px;
            border: 1px solid #e0cfc0;
            text-align: left;

```

```

        word-wrap: break-word;
        vertical-align: top;
    }
    th {
        background: #f7e1d6;
    }
    img.product-img {
        max-width: 100px;
        height: auto;
        display: block;
        margin-bottom: 5px;
    }
    .back-button {
        display: inline-block;
        margin-bottom: 20px;
        background: #5a1a12;
        color: white;
        padding: 10px 20px;
        text-decoration: none;
        border-radius: 6px;
    }
    @media screen and (max-width: 768px) {
        table, thead, tbody, th, td, tr {
            display: block;
        }
        td {
            margin-bottom: 10px;
        }
        th {
            display: none;
        }
    }
</style>
</head>
<body>
    <a href="admin.php" class="back-button">← Назад до панелі
адміністратора</a>

    <h2>Список замовлень</h2>

    <table>
        <thead>
            <tr>
                <th>ID</th>
                <th>Товар</th>
                <th>Ім'я</th>
                <th>Email</th>
                <th>Телефон</th>
                <th>Адреса</th>
                <th>Оплата</th>
                <th>Ціна</th>
            </tr>
        </thead>

```

```

<tbody>
  <?php while($row = $result->fetch_assoc()): ?>
    <tr>
      <td><?php echo $row['id']; ?></td>
      <td>
        <?php if (!empty($row['product_image'])):
?>
          "
          <?php endif; ?>
          <?php echo
htmlspecialchars($row['order_name'] ?? 'Невідомо'); ?>
        </td>
        <td><?php echo
htmlspecialchars($row['user_name'] ?? ''); ?></td>
        <td><?php echo
htmlspecialchars($row['user_email'] ?? ''); ?></td>
        <td><?php echo
htmlspecialchars($row['user_phone'] ?? ''); ?></td>
        <td><?php echo
htmlspecialchars($row['user_address'] ?? ''); ?></td>
        <td><?php echo
htmlspecialchars($row['payment_method'] ?? ''); ?></td>
        <td><?php echo
number_format((float)$row['price'], 2, '.', '') . ' грн'; ?></td>
      </tr>
    <?php endwhile; ?>
  </tbody>
</table>
</body>
</html>

```