

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ВСП «ЕКОНОМІКО-ПРАВНИЧИЙ ФАХОВИЙ КОЛЕДЖ**  
**ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ»**

Циклова комісія математичних дисциплін та інформаційних технологій

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: «РОЗРОБКА ДИНАМІЧНОГО ІНТЕРНЕТ- МАГАЗИНУ КВІТІВ З  
ВИКОРИСТАННЯМ JAVASCRIPT ДЛЯ ПОКРАЩЕННЯ  
КОРИСТУВАЦЬКОГО ДОСВІДУ ТА ЕФЕКТИВНОГО УПРАВЛІННЯ  
КОНТЕНТОМ»

Виконав:	<u>здобувач освіти 4 курсу, групи К121-20</u>
Спеціальність	<u>121 Інженерія</u> <u>програмного забезпечення</u> (шифр і спеціальність)
	<u>Олександр КАЛАШНИКОВ</u> (ім'я та ПРІЗВИЩЕ)
Керівник	<u>Катерина РЕШЕВСЬКА</u> (ім'я та ПРІЗВИЩЕ)
Рецензент	<u>завідувач кафедри програмної інженерії</u> <u>ЗНУ, доцент, к.ф.-м.н., Андрій ЛІСНЯК</u> (посада, вчене звання, науковий ступінь, ім'я та ПРІЗВИЩЕ)

Запоріжжя  
2024



6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.12.2023

**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи:		
	Постановка задачі	Грудень 2023	виконано
2.	Збір вихідних даних, обробка методичних та теоретичних джерел	Січень 2024	виконано
3.	Розробка першого розділу:		
	Огляд інформаційних систем та технологій для їхнього створення	Лютий 2024	виконано
4.	Розробка другого розділу:		
	Розробка проекту інформаційної системи та вибір технологій	Квітень 2024	виконано
5.	Розробка третього розділу:		
	Розробка застосунку	Травень 2024	виконано
6.	Оформлення і нормоконтроль кваліфікаційної роботи та перевірка на плагіат	Червень 2024	виконано
7.	Захист кваліфікаційної роботи	21.06.2024	виконано

Здобувач освіти \_\_\_\_\_  
(підпис)

Олександр КАЛАШНИКОВ  
(ім'я ПРІЗВИЩЕ)

Керівник роботи \_\_\_\_\_  
(підпис)

Катерина РЕШЕВСЬКА  
(ім'я ПРІЗВИЩЕ)

**Нормоконтроль пройдено**

Нормоконтролер \_\_\_\_\_  
(підпис)

Юлія БОРИСОВСЬКА  
(ім'я ПРІЗВИЩЕ)

## РЕФЕРАТ

Кваліфікаційна робота: 39 сторінок, 1 таблиця, 8 ілюстрацій, 4 позиції у переліку посилань.

Об'єкт дослідження – основні методи та засоби розробки інформаційних систем для інтернет-магазинів квітів з інтерактивним конструктором букетів.

Предмет дослідження – інформаційна система для інтернет-магазину квітів з можливістю інтерактивного створення та замовлення букетів.

Мета дослідження – розробка інформаційної системи для інтернет-магазину квітів з інтерактивним конструктором букетів.

У кваліфікаційній роботі розглянуті теоретичні аспекти створення інформаційних систем для онлайн-торгівлі, зокрема аналіз сучасних підходів та прикладів існуючих систем. На основі цих знань було розроблено концепцію інформаційної системи для інтернет-магазину квітів з можливістю інтерактивного конструювання букетів. Для реалізації проекту були використані HTML, CSS та JavaScript для фронтенду, а також PHP для бекенду. Для зберігання даних використовувався MySQL через інтерфейс PHPMyAdmin.

Цей підхід дозволив створити функціональну інформаційну систему, що дозволяє користувачам створювати і замовляти букети онлайн, забезпечуючи зручність та ефективність взаємодії.

ВЕБСАЙТ, ІНТЕРНЕТ-МАГАЗИН КВІТІВ, КОНСТРУКТОР БУКЕТІВ,  
JAVASCRIPT, HTML, CSS, PHP, MYSQL, PHPMYADMIN.

## SUMMARY

Thesis: 39 pages, 1 table, 8 illustrations, 4 references.

Research Object – primary methods and tools for developing information systems for online flower shops with an interactive bouquet constructor.

Research Subject – information system for an online flower shop enabling interactive creation and ordering of bouquets.

Research Goal – development of an information system for an online flower shop with an interactive bouquet constructor.

The thesis explores theoretical aspects of creating information systems for online commerce, including analysis of contemporary approaches and examples of existing systems. Based on this knowledge, a concept for an information system for an online flower shop with an interactive bouquet constructor was developed. The project utilized HTML, CSS, and JavaScript for the frontend, and PHP for the backend. MySQL was used via PHPMyAdmin for data storage.

This approach facilitated the creation of a functional information system allowing users to create and order bouquets online, ensuring convenience and effective interaction.

WEBSITE, ONLINE FLOWER SHOP, BOUQUET DESIGNER, JAVASCRIPT, HTML, CSS, PHP, MYSQL, PHPMYADMIN.

## ЗМІСТ

Завдання на кваліфікаційну роботу студенту .....	2
Реферат .....	4
Summary .....	5
Вступ.....	7
1 Огляд систем конструктору букетів в інтернет магазинах.....	9
1.1 Огляд доступних функціональностей та можливостей систем конструктора букетів у Інтернет-магазинах.....	9
1.2 Аналіз інтерфейсу користувача та процесу створення букетів з використанням цих систем.....	11
1.3 Висновки за розділом 1 .....	12
2 Розробка проекту.....	14
2.1 Технічне завдання .....	14
2.2 Етапи створення системи конструктору .....	15
2.3 Діаграма прецедентів .....	17
2.4. Проектування інтерфейсу конструктора .....	20
2.5 Висновки за розділом 2 .....	23
3 Розробка програмного продукту .....	26
3.1 Створення конструктора .....	26
3.2 Реалізація коду конструктора .....	28
3.3 Інструкція з використання конструктора .....	36
3.4 Висновки за розділом 3 .....	37
Висновки .....	38
Перелік посилань.....	39

## ВСТУП

Для кваліфікаційної роботи на тему розробки вебінтерфейсу для інтернет-магазину квітів на мові програмування JavaScript, важливо правильно визначити об'єкт та предмет дослідження. Об'єкт дослідження зазвичай є більш широким поняттям і включає в себе всі аспекти досліджуваного явища, в той час як предмет дослідження є конкретним аспектом або частиною об'єкта, на якій фокусується дослідження.

Об'єкт дослідження: Інтернет-магазини квітів як частина електронної комерції у сфері флористики.

Предмет дослідження: Розробка сервісу для збору та замовлення квітів онлайн з використанням JavaScript та базою даних PHPmyAdmin.

У цьому контексті, дослідження буде включати аналіз функціоналу, дизайну, зручності користування та впливу на споживачів вебсайту FLOWER DESIGN як парадигми для інших онлайн-платформ, присвячених флористичному мистецтву. Також буде розглянуто роль таких вебресурсів у розвитку сучасної флористичної індустрії.

Мета дослідження: Спроекувати та розробити ефективний та інтуїтивно зрозумілий вебінтерфейс для інтернет-магазину квітів, використовуючи можливості мови програмування JavaScript, з урахуванням сучасних тенденцій у вебдизайні та потреб користувачів.

В рамках цієї мети будуть вирішуватися наступні завдання:

- Аналіз існуючих вебресурсів, присвячених флористичному дизайну.
- Визначення ключових елементів та функцій, необхідних для успішного функціонування інтернет-магазину квітів.
- Проектування структури та дизайну вебінтерфейсу.
- Розробка та тестування вебінтерфейсу за допомогою JavaScript.
- Оцінка впливу розробленого інтерфейсу на досвід користувачів та його конкурентоспроможність на ринку.

Це дозволить створити інтернет-магазин квітів, який буде не тільки зручним для користувачів, але й сприятиме розвитку флористичної індустрії за допомогою передових технологій.

Структура роботи включає вступ, три розділи, висновки та список використаних джерел. Загальний обсяг магістерської роботи складає 39 сторінки і включає 11 рисунків. Список використаних джерел містить 4 джерела.



# 1 ОГЛЯД СИСТЕМ КОНСТРУКТОРУ БУКЕТІВ В ІНТЕРНЕТ МАГАЗИНАХ

## 1.1 Огляд доступних функціональностей та можливостей систем конструктора букетів у Інтернет-магазинах

Сучасні інтернет-магазини квітів дійсно надають унікальні можливості для створення власних букетів за допомогою інтерактивних конструкторів. Давайте розглянемо кілька прикладів таких магазинів:

- BloomThat: Цей магазин дозволяє клієнтам створювати власні букети, вибираючи види кольорів, їх кольори та додаючи аксесуари, такі як вази чи листівки. Додаткова інформація доступна [1].

- The Bouqs Co.: Магазин славиться свіжими та екологічно чистими квітами. Клієнти можуть складати власні букети, вибираючи квіти за кольором, типом та розміром. Додаткова інформація доступна [2].

- UrbanStems: UrbanStems пропонує стильні та сучасні квіткові композиції, які можна персоналізувати через їх конструктор. Додаткові прикраси та аксесуари також доступні. Додаткова інформація доступна [3].

Ці приклади ілюструють різноманітність та можливості, які надають сучасні інтернет-магазини квітів своїм клієнтам, дозволяючи їм створювати унікальні та персоналізовані букети онлайн. За допомогою цих систем, клієнти можуть створити букет, який відповідає їхнім уподобанням та потребам, змішуючи різні види квітів, кольори та композиції.

Системи конструктора букетів у Інтернет-магазинах представляють собою інноваційні інструменти, які надають користувачам можливість створювати власні унікальні композиції квітів. Основним завданням цих систем є надання клієнтам можливості індивідуалізувати свої покупки, враховуючи їхні особисті уподобання та потреби.

Однією з ключових функціональностей є можливість вибору квітів за різними критеріями, такими як вид квітів, колір, розмір тощо. Користувачі можуть переглядати доступний асортимент квітів та обирати ті, які вони бажають включити до свого букету.

Крім того, системи конструктора букетів зазвичай надають користувачам можливість додавати різноманітні додаткові елементи до своїх композицій, такі як зелень, красиві оформлення та прикраси. Це дозволяє клієнтам створювати унікальні та естетично привабливі букети, які відповідають їхнім індивідуальним смакам.

Системи конструкторів букетів зазвичай мають дуже зручний інтерфейс, що дозволяє користувачам легко орієнтуватися та використовувати всі наявні функції без особливих труднощів. Інтуїтивно зрозумілий і простий у використанні інтерфейс робить процес створення букетів приємним та захоплюючим для кожного клієнта.

Загалом, системи конструктора букетів у Інтернет-магазинах відкривають нові можливості для клієнтів, дозволяючи їм створювати індивідуальні та унікальні композиції квітів, які відповідають їхнім уподобанням та потребам.

- Завантаження даних про квіти з бази даних: Система використовує технологію AJAX для отримання інформації про доступні квіти у магазині. Ця функціональність дозволяє користувачам переглядати широкий асортимент квітів та обирати ті, які їм подобаються.

- Додавання товарів до корзини: Користувачі можуть додавати обрані квіти до корзини покупок за допомогою спеціальної кнопки "Додати в корзину". Після додавання товару, він автоматично з'являється у списку товарів у корзині, де можна переглянути загальну кількість товарів та їх сумарну вартість.

- Видалення товарів з корзини: Користувачі мають можливість видаляти товари з корзини, якщо вони змінили свої вимоги або вирішили

змінити склад свого замовлення. Ця функціональність дозволяє користувачам легко управляти вмістом своєї корзини покупок.

- Збільшення та зменшення кількості товарів: Для зручності користувачів, система надає можливість змінювати кількість квітів у корзині прямо на сторінці товару. Користувачі можуть збільшувати або зменшувати кількість товару за допомогою відповідних кнопок "плюс" та "мінус".

- Оформлення замовлення: Після того, як користувачі обрали всі необхідні товари, вони можуть перейти до сторінки оформлення замовлення. Тут вони можуть ввести свої контактні дані для доставки, обрати зручний спосіб оплати та переглянути підсумок свого замовлення перед його підтвердженням.

Загальний функціонал цих систем дозволяє користувачам максимально персоналізувати свої покупки та отримати задоволення від процесу створення унікальних букетів квітів прямо на вебсайті магазину.

## **1.2 Аналіз інтерфейсу користувача та процесу створення букетів з використанням цих систем**

Інтерфейс користувача систем конструктора букетів в Інтернет-магазинах відіграє ключову роль у забезпеченні зручності та ефективності процесу вибору та створення букетів. Він повинен бути інтуїтивно зрозумілим, зручним у використанні та естетично привабливим, щоб залучати користувачів та забезпечувати їм приємний досвід покупок.

Основні аспекти аналізу включають оцінку зручності навігації, доступність функціоналу, швидкість завантаження сторінок, а також унікальні функції, що відрізняють систему від конкурентів. Інтерфейс повинен бути адаптивним, щоб забезпечувати зручну роботу на різних пристроях, від комп'ютерів до мобільних телефонів.

Процес створення букетів зазвичай починається з вибору квітів та додаткових елементів з великого асортименту, доступного в системі. Користувачам надається можливість переглядати кожен елемент детально, переконуючись, що вони обирають саме те, що їм потрібно.

Після вибору всіх компонентів користувач може перейти до оформлення замовлення, вказавши кількість та інші деталі. При цьому система зазвичай автоматично розраховує загальну вартість замовлення та надає різні способи доставки та оплати.

Аналіз інтерфейсу користувача та процесу створення букетів в системах конструктора букетів у Інтернет-магазинах допомагає вдосконалювати їх функціональність та забезпечувати задоволення користувачів, роблячи покупки квітів онлайн простими та приємними.

### **1.3 Висновки за розділом 1**

У результаті аналізу доступних функціональностей та інтерфейсу систем конструктора букетів у Інтернет-магазинах можна зробити кілька висновків.

По-перше, зручний та інтуїтивно зрозумілий інтерфейс грає важливу роль у забезпеченні приємного досвіду користувачів під час вибору та створення букетів. Користувачі цінують зручність навігації та доступність функціоналу.

По-друге, важливою є широка можливість вибору квітів та додаткових елементів, що надає користувачам можливість створювати унікальні та індивідуальні букети за їхніми власними смаками та потребами.

По-третє, ефективний процес оформлення замовлення, який включає автоматичний розрахунок вартості, різні способи доставки та оплати, сприяє зручності та швидкості здійснення покупок.

Загалом, системи конструктора букетів у Інтернет-магазинах є потужним інструментом, який допомагає споживачам отримувати унікальні та красиві композиції квітів за допомогою онлайн-платформ, забезпечуючи зручність та задоволення від покупок.

## **2 РОЗРОБКА ПРОЕКТУ**

### **2.1 Технічне завдання**

#### **2.1.1 Найменування і область застосування**

Даний вебсайт створений для того, щоб спростити процес складання букету та покупки квітів. Він надає користувачам унікальну можливість використовувати конструктор для створення саме того букету, який вони мріють мати. Тепер ви можете створити неповторний композиційний шедевр, вибравши кожен квітковий елемент з увагою до деталей. Завдяки цьому сайту, ви зможете дати вільний політ вашій фантазії та створити букет, що ідеально відповідає вашому смаку та настрою.

#### **2.1.2 Підстава для розробки**

Програма розробляється на підставі наказу № 2004-с «Про затвердження тем кваліфікаційних робіт студентів 4 курсу освітньо-професійного ступеня фаховий молодший бакалавр денної форми здобуття освіти ВСП «Економіко-правничий фаховий коледж ЗНУ» від 30.11.2023 року.

#### **2.1.3 Призначення розробки**

Цей проект був створений з метою вирішення низки завдань:

- Персоналізовані букети: Забезпечення можливості користувачам створювати унікальні букети, враховуючи їхні індивідуальні вподобання та смак.

- Спрощення процесу замовлення: Надання зручного і інтуїтивно зрозумілого інтерфейсу, який дозволяє легко скласти букети та здійснювати покупки онлайн.
- Розширення варіативності вибору: Дозвіл на вибір квітів, кольорів, форм і розмірів з метою створення різноманітних композицій, що відповідають потребам будь-якого клієнта.
- Створення унікального досвіду: Надання можливості клієнтам відчувати себе як дизайнерами, що втілюють в життя свої власні ідеї та концепції у вигляді квіткових композицій.
- Забезпечення якості та задоволення клієнтів: Забезпечення високої якості квітів і обслуговування, щоб кожен клієнт отримав задоволення від свого покупного досвіду і букет відповідав їхнім очікуванням.

#### **2.1.4 Технічні вимоги до програмного продукту**

Програмне забезпечення для цього проекту було реалізоване за допомогою HTML та CSS для створення вебсайту та його стилізації. Крім того, використовувалися JavaScript (JS) та PHP для додавання інтерактивності та обробки даних. Для зберігання та керування інформацією про квіти, користувачів та замовлення використовувалася база даних, яка була побудована за допомогою PhpMyAdmin. Такий комплекс технологій дозволив створити функціональний та зручний вебсайт для збирання букетів та покупки квітів.

#### **2.2 Етапи створення системи конструктору**

Проектування інтерфейсу конструктора є важливим етапом у розробці програмного забезпечення. Інтерфейс конструктора забезпечує користувачам

можливість створення, редагування та налаштування об'єктів або елементів у системі. У цьому розділі ми розглянемо основні аспекти та принципи, які слід враховувати при проектуванні інтерфейсу конструктора.

Основні принципи проектування інтерфейсу конструктора.

### **Простота та зручність використання:**

Інтерфейс має бути інтуїтивно зрозумілим та легким у використанні навіть для нових користувачів. Всі елементи управління повинні бути логічно розміщені та доступні.

### **Гнучкість та настроюваність:**

Користувачі повинні мати можливість налаштовувати інтерфейс відповідно до своїх потреб та вимог. Це може включати налаштування зовнішнього вигляду, розташування елементів, а також функціональних можливостей.

### **Візуальна привабливість:**

Інтерфейс має бути естетично привабливим. Використання кольорів, шрифтів та графічних елементів повинно бути гармонійним та сприяти зручності сприйняття інформації.

### **Забезпечення зворотного зв'язку:**

Система повинна надавати користувачам зворотний зв'язок на їхні дії. Це може включати сповіщення про успішне виконання операцій, помилки або попередження.

### **Доступність:**

Інтерфейс повинен бути доступним для всіх категорій користувачів, включаючи людей з обмеженими можливостями. Це може включати підтримку екранних читалок, налаштування контрасту та розміру шрифту.

### **Мінімізація кількості кроків:**

Кількість дій, необхідних для досягнення результату, має бути мінімізована. Користувачі повинні мати можливість виконувати основні операції за мінімальну кількість кроків.



## 2.3 Діаграма прецедентів

Діаграма прецедентів (або діаграма варіантів використання) (англ. Use case diagram) – в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі [4]. Така діаграма (див. табл. 2.1) демонструє різні сценарії використання системи і типи її користувачів.

Діаграма варіантів використання часто доповнюється іншими типами діаграм для більш повного представлення системи.

Таблиця 2.1 – Діаграма прецедентів

Актор	Прецедент
Користувач	Додати Квітку Видалити Квітку Збільшити кількість Зменшити кількість Створити замовлення Видалити замовлення

Варіанти використання представлені колами або еліпсами (див. рис. 2.1). Актори (дійові особи) часто зображуються у вигляді паличок.

### Прецедент «Додати Квітку»

Призначення: даний варіант використання надає можливість користувачу додати нову квітку до списку.

Основний потік подій:

Прецедент починає виконуватися, коли користувач надсилає команду «додати квітку» разом з інформацією про квітку.

Система перевіряє валідність введеної інформації про квітку.

Якщо інформація валідна, система додає квітку до списку.

Виняткова ситуація 1:

Якщо інформація невалідна, система відправляє повідомлення про помилку.

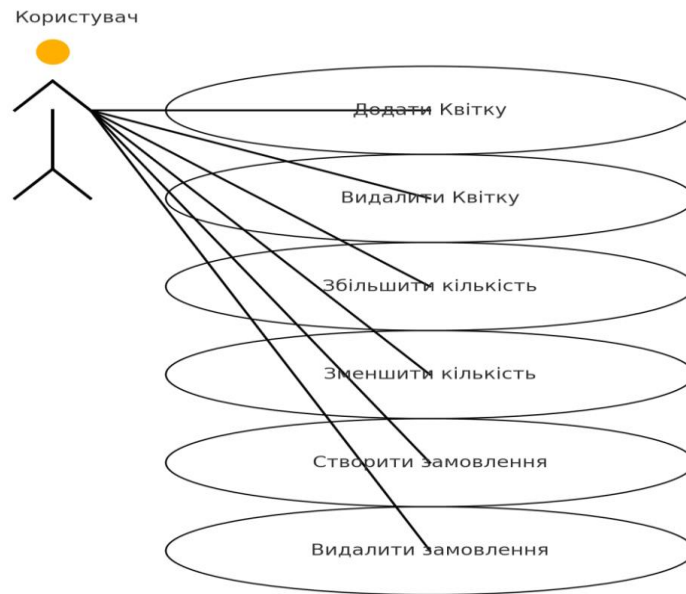


Рисунок 2.1 – Діаграма прецедентів

### **Прецедент «Видалити Квітку»**

Призначення: даний варіант використання надає можливість користувачу видалити квітку зі списку.

Основний потік подій:

Прецедент починає виконуватися, коли користувач надсилає команду «видалити квітку» разом з інформацією про квітку.

Система перевіряє, чи існує квітка у списку.

Якщо квітка існує, система видаляє квітку зі списку.

Виняткова ситуація 1:

Якщо квітка не знайдена у списку, система відправляє повідомлення про помилку.

### **Прецедент «Збільшити кількість»**

Призначення: даний варіант використання надає можливість користувачу збільшити кількість певної квітки.

Основний потік подій:

Прецедент починає виконуватися, коли користувач надсилає команду «збільшити кількість» разом з інформацією про квітку та кількість.

Система перевіряє валідність введеної інформації.

Якщо інформація валідна, система збільшує кількість вказаної квітки у списку.

Виняткова ситуація 1:

Якщо інформація невалідна, система відправляє повідомлення про помилку.

### **Прецедент «Зменшити кількість»**

Призначення: даний варіант використання надає можливість користувачу зменшити кількість певної квітки.

Основний потік подій:

Прецедент починає виконуватися, коли користувач надсилає команду «зменшити кількість» разом з інформацією про квітку та кількість.

Система перевіряє валідність введеної інформації.

Якщо інформація валідна, система зменшує кількість вказаної квітки у списку.

Виняткова ситуація 1:

Якщо інформація невалідна або кількість перевищує наявну, система відправляє повідомлення про помилку.

### **Прецедент «Створити замовлення»**

Призначення: даний варіант використання надає можливість користувачу створити нове замовлення.

Основний потік подій:

Прецедент починає виконуватися, коли користувач надсилає команду «створити замовлення» разом з інформацією про квітки та кількість.

Система перевіряє валідність введеної інформації.

Якщо інформація валідна, система створює нове замовлення.

Виняткова ситуація 1:

Якщо інформація невалідна, система відправляє повідомлення про помилку.

### **Прецедент «Видалити замовлення»**

Призначення: даний варіант використання надає можливість користувачу видалити існуюче замовлення.

Основний потік подій:

Прецедент починає виконуватися, коли користувач надсилає команду «видалити замовлення» разом з інформацією про замовлення.

Система перевіряє наявність замовлення.

Якщо замовлення існує, система видаляє замовлення.

Виняткова ситуація 1:

Якщо замовлення не знайдено, система відправляє повідомлення про помилку.

## **2.4. Проектування інтерфейсу конструктора**

Проектування інтерфейсу конструктора є важливим етапом у розробці вебдодатку, оскільки від нього залежить зручність користування сайтом та взаємодія користувачів з системою. В даному проєкті інтерфейс конструктора був реалізований за допомогою HTML, CSS, JavaScript, PHP та бази даних, що керується через phpMyAdmin. На рисунку 2.2 можливо побачити що вийшло в результаті.

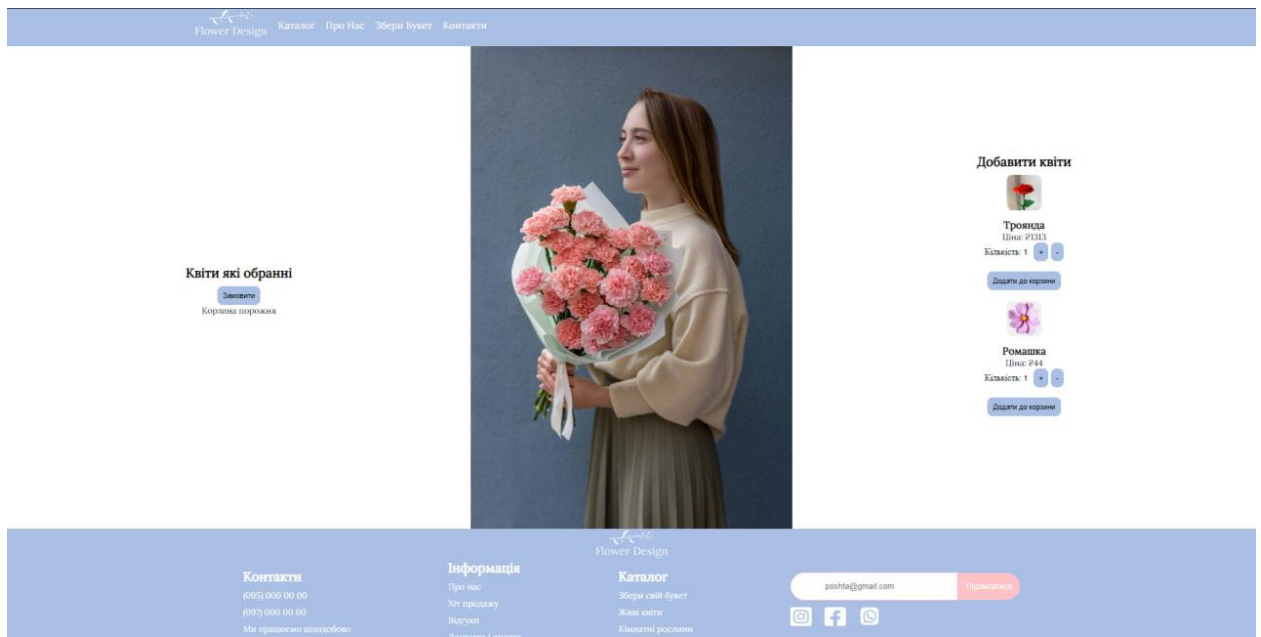


Рисунок 2.2 – Інтерфейс конструктора

На рисунку 2.3 можливо побачити, як буде виглядати, якщо вибрати вподобані квіти.

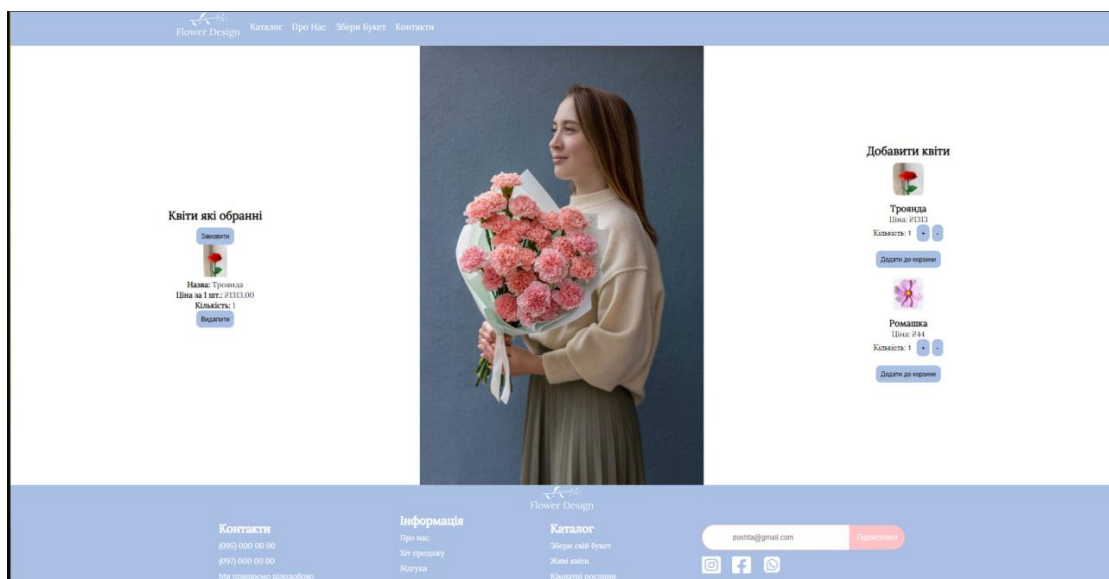



Рисунок 2.3 – Інтерфейс конструктора в якому вибрані вподобані квітки

На рисунку 2.4 можливо побачити, як буде виглядати, якщо почати оформлювати замовлення.

**Сторінка замовлення**

[Головна](#)

**Обрані товари:**


Троянда - 1 шт. - #1313

Усього товарів: 1 шт.  
Загальна сума замовлення: #1313

**Оформити замовлення:**

ІМ'Я:

ПІЛЬСЬКЕ:

Номер телефону:

Місто:

Спосіб оплати:

Надомовній платіж  Спосіб оплати:

Рисунок 2.4 – Оформлення замовлення

## 1. HTML

HTML (HyperText Markup Language) є основою будь-якого вебдодатку. Для створення структури конструктора було використано наступні елементи:

**Форма:** основний елемент інтерфейсу, який дозволяє користувачам обирати квіти та формувати букет.

**Кнопки:** для додавання квітів до корзини та замовлення обраного букету.

**Зображення:** для візуалізації квітів, що додаються до букету.

HTML забезпечує структурну розмітку сторінки, яка є основою для подальшого стилізації та інтерактивності.

## 2. CSS

CSS (Cascading Style Sheets) використовується для стилізації HTML-елементів, щоб надати інтерфейсу привабливого вигляду. В даному проєкті було використано наступні стилі:

**Основні стилі:** задають загальний вигляд сторінки, такі як кольори фону, шрифтів та відступів.

Сітка: для розташування елементів конструктора та квітів.

Анімації: для плавних переходів при додаванні квітів до корзини.

### 3. JavaScript

JavaScript (JS) додає інтерактивність до вебдодатку. Для даного проєкту були реалізовані наступні функціональні можливості:

Додавання квітів до корзини: користувач може додавати квіти до корзини, натискаючи на кнопку "Додати до корзини".

Оновлення вмісту корзини: після додавання квітів відображається актуальний стан корзини.

### 4. PHP

PHP (Hypertext Preprocessor) використовується для обробки серверної логіки. В даному проєкті PHP відповідає за обробку замовлень та взаємодію з базою даних.

### 5. phpMyAdmin

phpMyAdmin використовується для управління базою даних MySQL. В даному проєкті за допомогою phpMyAdmin було створено базу даних з таблицею orders, яка зберігає інформацію про замовлення квітів.

Проектування інтерфейсу конструктора з використанням HTML, CSS, JavaScript, PHP та phpMyAdmin дозволило створити зручний та функціональний вебдодаток для замовлення квітів. Це забезпечує користувачам легкість у використанні та задоволення від взаємодії з системою.

## 2.5 Висновки за розділом 2

Розробка інтернет-магазину квітів з інтерактивним конструктором букетів має за мету створення зручної платформи для покупців, де вони зможуть самостійно складати букети за своїми уподобаннями та перевагами. Інтерактивний конструктор дозволить користувачам експериментувати з

різними комбінаціями квітів, кольорів та дизайну, забезпечуючи індивідуальний підхід до кожного замовлення.

Зручний інтерфейс у розробці інтернет-магазину квітів з інтерактивним конструктором букетів відіграє критичну роль у забезпеченні позитивного досвіду користувачів. При розробці такого інтернет-магазину необхідно враховувати інтуїтивність та легкість навігації для кожного користувача.

Користувачі повинні легко знаходити потрібні їм квіти та мати можливість самостійно скласти букети за допомогою інтерактивного конструктора. Чіткий та простий інтерфейс допоможе забезпечити зручність вибору квітів, кольорів та дизайну, забезпечуючи задоволення від купівлі кожному клієнтові.

Крім того, важливо враховувати адаптивність інтерфейсу під різні пристрої, забезпечуючи однаково зручний доступ до магазину як на комп'ютерах, так і на мобільних пристроях. Це дозволить залучити більше клієнтів та забезпечити їм комфортне користування вашим сервісом незалежно від обраного пристрою.

Процес тестування та впровадження є ключовим етапом у розробці інтернет-магазину квітів з інтерактивним конструктором букетів. Перед тим, як приступити до впровадження, необхідно провести ретельне тестування для виявлення і усунення всіх можливих помилок та недоліків.

Тестування повинно охоплювати всі аспекти функціональності та взаємодії користувачів з інтерфейсом і магазином. Це включає перевірку роботи інтерактивного конструктора букетів, коректне відображення сторінок на різних пристроях та веббраузерах, а також перевірку безпеки та захисту персональних даних користувачів.

Після успішного завершення тестування можна приступати до впровадження інтернет-магазину. Перед цим важливо ретельно підготувати всі необхідні матеріали та ресурси, включаючи інструкції для користувачів, документацію та підтримку.



Після впровадження необхідно продовжувати вдосконалювати магазин на основі отриманого зворотного зв'язку від користувачів. Це включає вносити зміни в інтерфейс та функціональність, вирішувати виникаючі проблеми та надавати підтримку користувачам для забезпечення їхнього задоволення від використання магазину.

## 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

Під час розробки інформаційної системи було визначено, що оптимальним рішенням буде створення конструктора для вебсайту з використанням JavaScript, а також інтеграції PHP та PHPmyAdmin. Для демонстрації роботи цього конструктора було обрано відповідний хостинг.

### 3.1 Створення конструктора

У цьому розділі ми розглянемо процес створення конструктора на JavaScript для взаємодії з користувачами через сайт. Основна мета полягає в розробці функціоналу конструктора для складання букетів, який надає можливість користувачам збирати товар, що їм подобається, через вебсайт. Використання баз даних PHPmyAdmin дозволить реалізувати необхідний функціонал для обробки товарів для клієнтів.

#### 3.1.1 Вибір технологій

- HTML - це мова розмітки, яка використовується для створення структури вебсторінок.
- CSS - це мова стилів, яка використовується для візуального оформлення вебсторінок, включаючи кольори, шрифти та розміщення елементів.
- JavaScript - це мова програмування, яка використовується для створення інтерактивності на вебсторінках, таких як анімація, обробка подій та взаємодія з користувачем.

- PHP - це мова програмування, яка використовується для розробки серверних застосунків та обробки даних на стороні сервера, що дозволяє створювати динамічний вебконтент.
- PHPmyAdmin - це інструмент для адміністрування баз даних MySQL через вебінтерфейс, який надає можливість керувати базами даних, таблицями, запитами та іншими аспектами баз даних.

### **3.1.2 Створення конструктора**

Створення конструктора є важливим етапом у розробці вебдодатку, що дозволяє користувачам створювати індивідуальні букети з квітів. В цьому розділі буде розглянуто приклад реалізації конструктора на основі HTML, CSS, JavaScript, PHP та бази даних MySQL.

Ініціалізація проєкту:

Для початку роботи створимо новий проєкт та налаштуємо його структуру:

- Створення структури проєкту: Створимо папку для проєкту та створимо в ній базові файли: index.html, styles.css, scripts.js, config.php.
- Налаштування вебсервера: Налаштуємо локальний вебсервер для роботи з PHP і MySQL.
- Налаштування бази даних: Створимо базу даних для зберігання інформації про квіти та замовлення.

### **3.1.3 Налаштування конструктора**

- Основний файл сторінки конструктора – buket.html. До сторінки також додані стилі в файлах style.css та mobile.css, а також JavaScript для функціональності.

- Зберігання даних товари: зберігаються в базі даних PhpMyAdmin.
- Оформлення замовлення: після створення букету клієнт переходить до оформлення замовлення, яке реалізоване за допомогою PHP.

### 3.2 Реалізація коду конструктора

Реалізація необхідного коду, що керує системою конструктора, відбувається в основному блоку, де відбувається весь процес:

```
<div class="sborbyket">
  <div class="buy-flowers1">
<h2 id="h222">Квіти які обранні</h2>
<div class="cart" id="cartContainer">
  <!-- Сюди додаватимуться вибрані елементи -->
  <button onclick="placeOrder()" id="butf">Замовити</button>
  <div id="cartItemsContainer">
    <!-- Тут буде відображатися вміст кошика -->
    <p>Корзина порожня</p>
  </div>
</div>
</div>
</div>

<div class="buy-flowers">
  <h2 id="h222">Добавити квіти</h2>
  <div id="flowersContainer">
</div>
</div>
</div>
```

Блок, який відповідає за корзину та оформлення замовлення, можна побачити на рисунку 3.1.

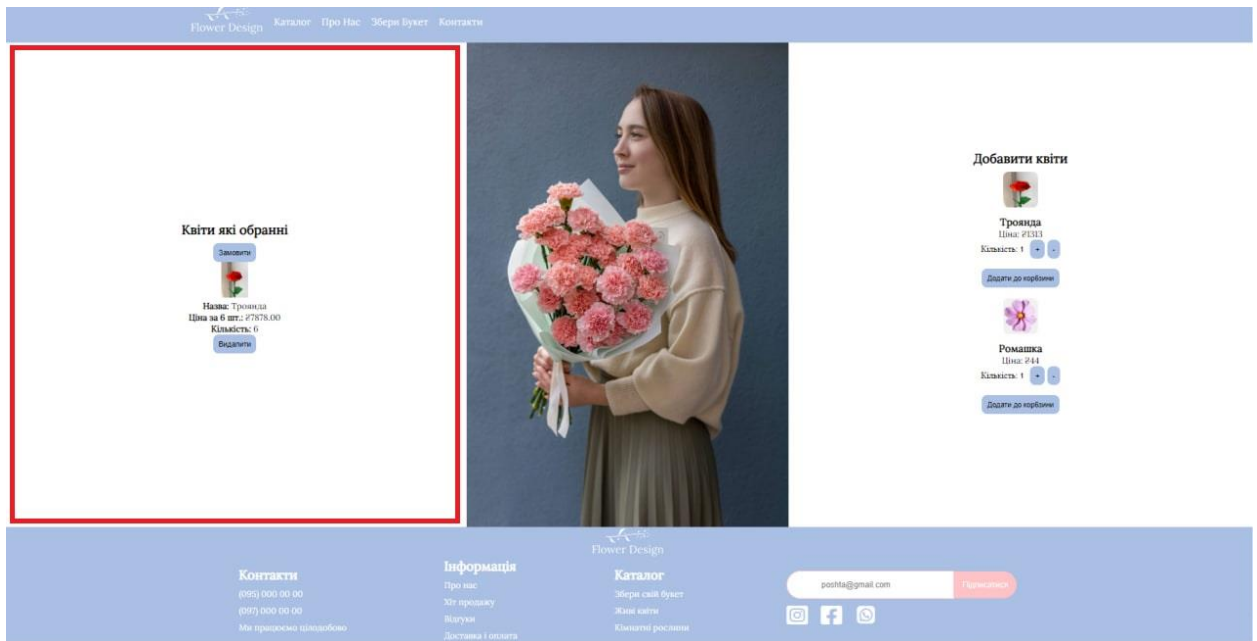


Рисунок 3.1 – Блок корзини в конструкторі

```
function addToCart(productId, quantityId) {

    var product = document.getElementById(productId);
    var quantityInput = document.getElementById(quantityId);
    var quantity = parseInt(quantityInput.value, 10);
    var pricePerUnit = parseFloat(product.querySelector('.price').textContent.replace('Ціна: ₴',
    ''));
    var totalPrice = pricePerUnit * quantity;
    var existingItem = Object.values(cartItems).find(item => item.name ===
    product.querySelector('h3').textContent);
    if (existingItem) {

        existingItem.quantity += quantity;
        existingItem.price = (parseFloat(existingItem.price) + totalPrice).toFixed(2);
        var cartItem = document.getElementById(existingItem.id);
        cartItem.querySelector('p:nth-of-type(3)').innerHTML = '<strong>Кількість:</strong> ' +
        existingItem.quantity;
        cartItem.querySelector('p:nth-of-type(2)').innerHTML = '<strong>Ціна за ' +
        existingItem.quantity + ' шт.:</strong> ₴' + existingItem.price;
    } else {
        var itemId = 'cartItem' + Object.keys(cartItems).length;
        var cartItem = document.createElement('div');
        cartItem.id = itemId;
        cartItem.innerHTML = '<img src="" + product.querySelector('img').src + ""
        width="50px">' +
            '<p><strong>Назва:</strong> ' + product.querySelector('h3').textContent +
            '</p>' +
```

```

        '<p><strong>Ціна за ' + quantity + ' шт.:</strong> ₪' + totalPrice.toFixed(2)
+ '</p>' +
        '<p><strong>Кількість:</strong> ' + quantity + '</p>' +
        '<button    onclick="removeFromCart(\'"    +    itemId    +    '\')"
id="butf">Видалити</button>';
    var cartItemsContainer = document.getElementById('cartItemsContainer');
    cartItemsContainer.appendChild(cartItem);
    cartItems[itemId] = {
        id: itemId,
        name: product.querySelector('h3').textContent,
        price: totalPrice.toFixed(2),
        quantity: quantity
    };
}
quantityInput.value = 1;
updateCartEmptyMessage();
}

```

Цей код демонструє основні функціональні можливості системи корзини

```

function loadFlowers() {
    var flowersContainer = document.getElementById('flowersContainer');

    var xhr = new XMLHttpRequest();
    xhr.open('GET', 'get_flowers.php', true);
    xhr.onload = function () {
        if (xhr.status >= 200 && xhr.status < 400) {

            flowersContainer.innerHTML = xhr.responseText;
        } else {
            console.error('Request failed');
        }
    };
    xhr.send();
}

```

А тут показано, як саме передається замовлення і після чого вас перенаправляє на сторінку оформлення замовлення. Екран оформлення замовлення можна побачити на рисунку 2.4.

Товар, який відображається на сайті, береться з бази даних PHPMyAdmin. Це можна побачити на рисунку 3.2.

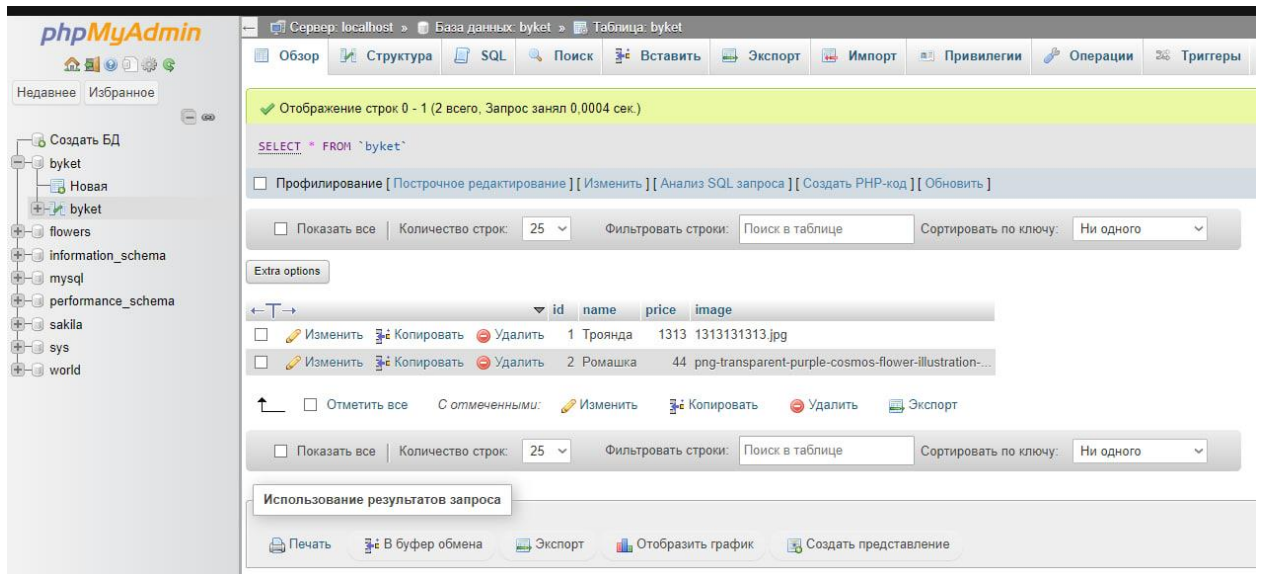


Рисунок 3.2 – Вікно бази даних в конструкторі

```

$sql = "SELECT id, name, price, image FROM byket";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        echo "<div class='product' id='product{".$row["id"]."}'>
            <img src='images/{.$row["image"].}' width='150px'>
            <h3>{".$row["name"]."}</h3>
            <p class='price'>Ціна: €{".$row["price"]."}</p>
            <div class='butprod'>
                <label for='quantity{".$row["id"]."}'>Кількість:</label>
                <input type='text' id='quantity{".$row["id"]."}' value='1' readonly class='inpflow'>
                <button onclick='increaseQuantity(\"quantity{".$row["id"]."}\")'+</button>
                <button onclick='decreaseQuantity(\"quantity{".$row["id"]."}\")'-</button>
            </div>
            <br>
            <button onclick='addToCart(\"product{".$row["id"]."}\",
            \"quantity{".$row["id"]."}\")'>Додати до корзини</button>
            </div>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>

```

У цьому коді показаний функціонал товару, зокрема: кількість, ім'я товару, ціна товару, картинка товару, кнопки для збільшення та зменшення кількості товару, а також кнопка "Додати до корзини". Вся ця інформація про товари береться з бази даних. Це можна побачити на рисунку 3.3.

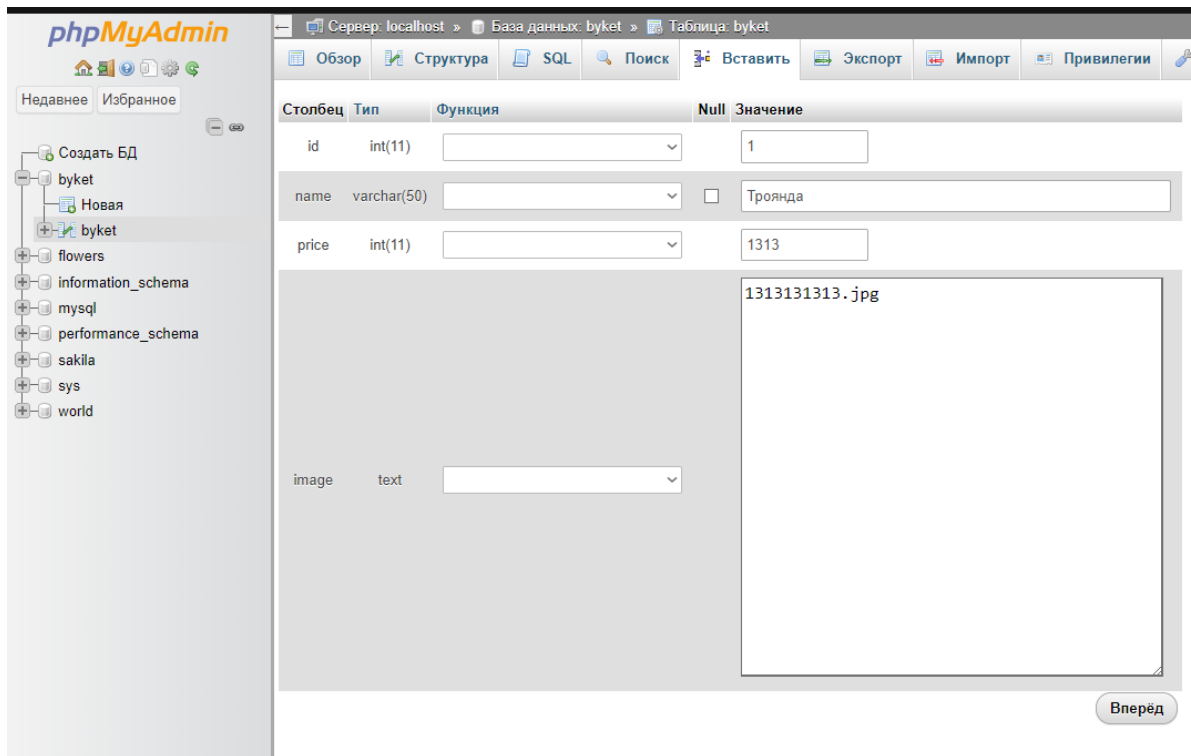


Рисунок 3.3 – Інформація про товар в базі даних

І по середині цього основного блоку sborbyket знаходиться відображення:

```

```

А також, коли ми перемикаємося на мобільну версію, це зображення зникає для кращої оптимізації під мобільні пристрої. Це можна побачити на рисунку 3.4.





Рисунок 3.4 – Мобільна версія блоку

Також, після того як ви зібрали букет і бажаєте оформити замовлення, вас перекидає на сторінку оформлення. Її вигляд можна побачити на рисунку 2.4. Ось основний код сторінки замовлення:

```
<div class="container">
  <h1>Сторінка замовлення</h1>
  <a href="index.html" style="color:black;">Головна</a>
  <?php
  if (isset($_GET['items'])) {
    $items = json_decode($_GET['items'], true);
    $totalQuantity = 0;
    $totalPrice = 0;
    if (!empty($items)) {
      foreach ($items as $item) {
        $totalQuantity += intval($item['quantity']);
```

```

        $totalPrice += intval($item['price']) * intval($item['quantity']);
    }
}

if (!empty($items)) {
    echo '<h2>Обрані товари:</h2>';
    echo '<ul>';
    foreach ($items as $item) {
        echo "<div class='cart-load'>";
        if (isset($item['image'])) {
            echo "<img src='images/{ $item['image'] }' alt='{ $item['name'] }' >";
        } else {
            echo "<img src='images/' alt='{ $item['name'] }' style='width: 100px; height:
100px;'>";
        }
        echo "{ $item['name'] } - { $item['quantity'] } шт. - €" . ((int)$item['price'] *
(int)$item['quantity']);
        echo "</div>";

    }
    echo '</ul>';
    echo '<br>';
    echo '<p>Усього товарів: ' . $totalQuantity . ' шт.</p>';
    echo '<p>Загальна сума замовлення: €' . $totalPrice . '</p>';

    echo '<h2>Оформити замовлення:</h2>';
    echo '<div class="form-container">';
    echo '<div class="form-group">';
    echo '<form action="send_order.php" method="post">';
    echo ' <label for="name">ПІБ:</label>';
    echo ' <input type="text" id="name" name="name" required><br>';
    echo '</div>';
    echo '<div class="form-group">';
    echo ' <label for="email">Пошта:</label>';
    echo ' <input type="email" id="email" name="email" required><br>';
    echo '</div>';
    echo '<div class="form-group">';
    echo ' <label for="number">Номер телефону:</label>';
    echo ' <input type="text" id="number" name="number" required><br>';
    echo '</div>';
    echo '<div class="form-group">';
    echo ' <label for="city">Місто:</label>';
    echo ' <input type="text" id="city" name="city" required><br>';
    echo '</div>';
    echo '<div class="form-group">';

```

```

echo ' <label for="payment">Спосіб оплати:</label>';
echo ' <select id="payment" name="payment" required>';
echo ' <option value="Наложений платіж">Наложений платіж</option>';
echo ' <option value="Оплата картою">Оплата картою</option>';
echo ' <option value="Оплата за рахунком">Оплата за рахунком</option>';
echo ' </select><br>';
echo '</div>';
echo ' <input type="hidden" name="items" value="" . htmlentities($_GET['items']) .
">';

echo ' <button type="submit" class="form-submit-btn">Замовити</button>';
echo '</form>';
echo '</div>';
} else {
    echo '<p>Вибраних товарів немає.</p>';
}
} else {
    echo '<p>Дані про товари не передані.</p>';
}
?>
</div>

```

Після того, як ви вказали правильні реквізити, замовлення надсилається на електронну пошту компанії.

```

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST['name']) && isset($_POST['email']) && isset($_POST['items'])) {

        $name = $_POST['name'];
        $email = $_POST['email'];
        $items = json_decode($_POST['items'], true);

        $subject = "Нове замовлення від $name";
        $message = "ІМЯ: $name\n";
        $message .= "Email: $email\n\n";
        $message .= "Товари:\n";
        foreach ($items as $item) {
            $message .= "- {$item['name']} (кількість: {$item['quantity']}, ціна за одну річ:
{$item['price']})\n";
        }

        $to = "@gmail.com"; // почта
    }
}

```

```
$headers = "From: $email";

if (mail($to, $subject, $message, $headers)) {
    echo "Замовлення успішно відправлено.";
} else {
    echo "Помилка при надсиланні замовлення.";
}
} else {
    echo "Неможливо отримати дані про замовлення.";
}
} else {
    echo "Неприпустимий метод запиту.";
}
?>
```

У цьому коді можна побачити, що інформація відправляється на електронну пошту, як було сказано раніше. Крім того, кожному замовленню присвоюється індивідуальний номер.

### 3.3 Інструкція з використання конструктора

Інструкція з використання конструктора букетів в інтернет-магазині:

Крок 1: Вибір квітів

- Вибір квітів: Зайдіть на сайт магазину, виберіть розділ "Збери Букет".
- Персоналізація: Оберіть види квітів, їх кольори та кількість. Додайте додаткові аксесуари, такі як ваза або листівка.

Крок 2: Оформлення замовлення обраних квітів

- Перевірка та замовлення: Перегляньте свій букет, переконайтеся, що все правильно, і додайте його до кошика.
- Оформлення замовлення: Заповніть дані для доставки та оплатіть замовлення.

### **3.4 Висновки за розділом 3**

У цьому розділі було описано процес розробки програмного продукту для створення конструктора букетів на вебсайті. Було вибрано відповідні технології, такі як HTML, CSS, JavaScript, PHP та PhpMyAdmin, для забезпечення інтерактивності, візуального оформлення та обробки даних. Після створення структури проєкту та налаштування бази даних розроблено конструктор, який дозволяє користувачам персоналізувати свої букети та оформляти замовлення онлайн. Такий підхід забезпечує ефективну взаємодію з користувачами та зручність у використанні системи. В результаті, було створено ефективну інформаційну систему, яка забезпечує зручність для користувачів і оптимізує процес покупки квітів онлайн.

## ВИСНОВКИ

На основі проведеного дослідження та реалізації проєкту інтернет-магазину квітів з інтерактивним конструктором букетів можна зробити наступні висновки:

1. Аналіз та проектування: Проведено детальний аналіз існуючих систем конструкторів букетів, що дозволило визначити ключові елементи та функції, необхідні для успішного функціонування інтернет-магазину. Було розроблено структуру та дизайн вебінтерфейсу, що забезпечує зручність використання та ефективну взаємодію з користувачами.

2. Технічна реалізація: Використано сучасні вебтехнології, такі як HTML, CSS, JavaScript, PHP та PhpMyAdmin, що дозволило створити інтерактивний та функціональний конструктор букетів. Система забезпечує персоналізацію букетів, вибір аксесуарів та оформлення замовлень онлайн.

3. Тестування та впровадження: Проєкт пройшов етапи тестування, що дозволило виявити та виправити можливі недоліки. Розроблений інтерфейс позитивно впливає на досвід користувачів та підвищує конкурентоспроможність інтернет-магазину на ринку.

4. Ефективність системи: Результати впровадження показали, що створена інформаційна система є ефективною та зручною для користувачів, оптимізує процес покупки квітів онлайн та сприяє зростанню продажів.

5. Перспективи розвитку: Проєкт має потенціал для подальшого розвитку та вдосконалення. Можливими напрямками є інтеграція додаткових функцій, таких як рекомендаційні системи, покращення алгоритмів персоналізації та розширення асортименту товарів та послуг.

Загалом, реалізація інтернет-магазину квітів з інтерактивним конструктором букетів підтвердила доцільність та ефективність такого підходу, забезпечуючи високий рівень задоволеності користувачів та досягнення поставлених цілей проєкту.

## ПЕРЕЛІК ПОСИЛАНЬ

1. BloomThat. URL: <https://bloomthat.com> (дата звернення: 15.06.2024).
2. Bouqs. URL: <https://bouqs.com> (дата звернення: 15.06.2024).
3. UrbanStems. URL: <https://urbanstems.com> (дата звернення: 15.06.2024).
4. Діаграма прецедентів. URL:  
[https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0\\_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D1%96%D0%B2](https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D1%96%D0%B2) (дата звернення: 24.04.2024).

**Декларація**  
**академічної доброчесності**  
**здобувача освіти ВСП «Економіко-правничого фахового коледжу ЗНУ»**


Я, Калашников Олександр Дмитрович, здобувач(-ка) освіти 4 курсу, спеціальності/освітньо-професійної програми Інженерія програмного забезпечення, групи K121-20, адреса електронної пошти erfk.121.20kalashnikov@gmail.com

- підтверджую, що написана мною кваліфікаційна робота на тему «Розробка динамічного інтернет-магазину квітів з використанням JavaScript для покращення користувацького досвіду та ефективного управління контентом» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений/ознайомлена;

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

- згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата 17.06.2024 Підпис \_\_\_\_\_

Дата 17.06.2024 Підпис 

Олександр КАЛАШНИКОВ

Катерина РЕШЕВСЬКА