

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко
підпис

«__» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА
Вебсайт доставляння букетів

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 408.22010803

Здобувачка

_____ А. А. Гошева
підпис

«__» _____ 2024 р.

Керівник канд. техн. наук, доцент

_____ Є. О. Давиденко
підпис

«__» _____ 2024 р.

Консультант канд. техн. наук, доцент

_____ А. О. Алексєєва
підпис

«__» _____ 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Є. О. Давиденко

« _____ » _____ 2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи бакалавра

Видано здобувачу групи 408 факультету комп'ютерних наук

Гошева Аліна Андріївна

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи

Вебсайт доставляння букетів

Затверджена наказом по ЧНУ від «22» _____ грудня _____ 2023 р. № _____ 269 _____

2. Строк представлення кваліфікаційної роботи « _____ » _____ 20 _____ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом є вебсайт доставки
букетів

4. Перелік питань, що підлягають розробці:

- дослідження предметної області та аналіз існуючих аналогів;
- формування специфікації вимог до програмного забезпечення;
- визначення архітектури для проектування програмного забезпечення;
- моделювання та проектування програмного забезпечення;
- розробка програмного забезпечення;
- здійснення тестування роботи програмного забезпечення.

5. Перелік графічних матеріалів

Презентація

6. Завдання до спеціальної частини

Аналіз охорони праці на робочих місцях фахівців з інформаційних технологій

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алексеева А. О.	Кафедра екології	Спеціальна частина з охорони праці

Керівник роботи _____ канд. техн. наук, доцент Давиденко Є. О. _____
(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

_____ Гошева Аліна Андріївна _____
(прізвище, ім'я, по батькові)

(підпис)

Дата видачі завдання « _____ » _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН
виконання бакалаврської кваліфікаційної роботи

Тема: «Вебсайт доставляння букетів»

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КРБ	20.02.2024	23.02.2024	Виконано
2	Огляд літератури за темою роботи	24.02.2024	26.02.2024	Виконано
3	Складання календарного плану КРБ	26.02.2024	27.02.2024	Виконано
4	Аналіз предметної області	28.02.2024	01.03.2024	Виконано
5	Розробка проектних рішень	04.03.2024	10.03.2024	Виконано
6	Моделювання та конструювання ПЗ	22.03.2024	26.03.2024	Виконано
7	Кодування ПЗ, тестування та апробація розробленого ПЗ, аналіз результатів тестування	28.03.2024	22.05.2024	Виконано
8	Розробка спеціальної частини з охорони праці	17.04.2024	17.04.2024	Виконано
9	Оформлення КРБ та презентації	20.04.2024	20.04.2024	Виконано
10	Відгук керівника КРБ	20.05.2024	20.05.2024	Виконано
11	Попередній захист	03.06.2024	05.06.2024	Виконано
12	Рецензування	14.06.2024	15.06.2024	Виконано
13	Завершення оформлення КРБ та презентації	14.06.2024	15.06.2024	Виконано
14	Захист кваліфікаційної роботи	25.06.2024	27.06.2024	Виконано

Розробила здобувачка

Гошева Аліна Андріївна

(прізвище, ім'я, по батькові)

(підпис)

«26» лютого 2024 р.

Керівник роботи

канд. техн. наук, доцент Давиденко Є. О.

(посада, прізвище, ім'я, по батькові)

(підпис)

«27» лютого 2024 р.

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Вебсайт доставляння букетів»

Здобувачка 408 гр,: Гошева Аліна Андріївна

Керівник: зав. кафедри ПЗ, канд. техн. наук, доцент Давиденко Є. О.

Дана кваліфікаційна робота присвячена розробці вебсайту для доставки квіткових букетів, що відповідає сучасним вимогам інтернет-технологій та потребам користувачів.

Об'єктом роботи є процес розробки та впровадження вебсайту, який забезпечує замовлення та доставку квітів за допомогою Інтернету.

Предметом роботи є сучасні технології програмування та дизайну для аналізу, проектування та реалізації функціоналу вебсайту.

Метою даної роботи є створення ефективного та зручного інструменту для користувачів замовлення квітів через Інтернет, що дозволить їм швидко та зручно здійснювати покупки, а компанії — ефективно управляти процесом прийому та обробки замовлень.

У першому розділі кваліфікаційної роботи проаналізована предметна область вебзастосунку, проведено порівняльний аналіз готових рішень та визначено специфікацію вимог.

У другому розділі кваліфікаційної роботи розроблено діаграми варіантів використання, класів, послідовностей дій системи та потоків даних. Розписано сценарії роботи вебзастосунку.

У третьому розділі кваліфікаційної роботи проведено вибір технологій розробки, представлено мокап системи та представлено структуру бази даних.

У четвертому розділі кваліфікаційної роботи представлено результати розробки вебзастосунку.

КРБ викладена на 64 сторінки, вона містить 4 розділи, 48 ілюстрацій, 19 джерел в переліку посилань

Ключові слова: вебсайт, доставка квітів, PHP, Laravel, база даних, MySQL

ABSTRACT

to the qualification work of the bachelor

"Bouquet delivery website"

Student of group 408: Hosheva Alina Andriivna

Supervisor: head of the Software Engineering Department, candidate technical of Sciences, associate professor Davydenko Y. O.

This thesis is devoted to the development of a website for the delivery of flower bouquets, which meets the modern requirements of Internet technologies and the needs of users.

The object of the work is the process of developing and implementing a website that facilitates online ordering and delivery of flowers.

The subject of the work involves modern programming and design technologies for analyzing, designing, and implementing the website's functionality.

The purpose of this work is to create an effective and convenient tool for users to order flowers via the Internet, which will allow them to quickly and conveniently make purchases, and companies to effectively manage the process of receiving and processing orders.

In the first section of the qualification work, the subject area of the web application was analyzed, a comparative analysis of ready-made solutions was carried out.

In the second section of the qualification work, diagrams of use cases, classes, sequences of system actions and data flows were developed.

In the third section of the qualification work, a selection of development technologies was made, a mockup of the system was presented, and the structure of the database was presented.

The fourth section of the qualification work presents the results of the web application development.

QWB is laid out on 64 pages, it contains 4 sections, 48 illustrations, 19 sources in the list of references

Keywords: website, flower delivery, PHP, Laravel, database, MySQL

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	3
ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	5
1.1 Аналіз предметної області	5
1.2 Аналіз готових рішень	6
1.2.1 Аналіз Flowers.ua.....	7
1.2.2 Аналіз Buketland.ua	9
1.2.3 Аналіз Floristik.ua	10
1.3 Специфікація вимог	12
Висновки до розділу 1	13
2 ПРОЄКТУВАННЯ ВЕБСАЙТУ ДОСТАВКИ БУКЕТІВ	14
2.1 Розробка діаграми варіантів використання.....	14
2.2 Розробка діаграми класів	16
2.3 Розробка діаграми послідовностей дій системи	18
2.4 Розробка моделі потоків даних системи	19
Висновки до розділу 2	20
3 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ ТА МОДЕЛЮВАННЯ БАЗИ ДАНИХ	21
3.1 Вибір технологій розробки	21
3.2 Моделювання бази даних системи	31
3.3 Розробка мокапу системи.....	35
Висновки до розділу 3	42
4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	43
4.1 Реалізація моделей та контролерів.	43
4.2 Реалізація фронтенду вебсайту.....	49
Висновки до розділу 4	61
ВИСНОВКИ	62
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	63

ПЕРЕЛІК СКОРОЧЕНЬ

APC	– alternative php cache
API	– application programming interface
CSRF	– cross-site request forgery
DFD	– data flow diagrams
HTML	– hypertext markup language
MVC	–model view controller
OOP	– object-oriented programming
ORM	– object relational mapping
PHP	– hypertext preprocessor
SQL	– structured query language
XSS	– cross-site scripting

ВСТУП

Сьогоднішній світ заснований на швидкому темпі життя та технологічних інноваціях, що безперервно перетворюють спосіб, яким ми спілкуємося, працюємо та здійснюємо покупки. Однією з галузей, яка постійно розвивається в цьому контексті, є електронна комерція, яка забезпечує зручний доступ до товарів та послуг за допомогою Інтернету.

У контексті електронної комерції, сегмент квіткового бізнесу має великий потенціал для інновацій та покращень. З урахуванням зростаючої популярності онлайн-покупок та зручності отримання товарів безпосередньо до дверей, розробка вебсайту для доставки квітів стає актуальною та перспективною задачею.

Об'єктом роботи є процес розробки та впровадження вебсайту, який забезпечує замовлення та доставку квітів за допомогою Інтернету.

Предметом роботи є сучасні технології програмування та дизайну для аналізу, проєктування та реалізації функціоналу вебсайту.

Метою даної роботи є створення ефективного та зручного інструменту для користувачів та підприємств замовлення квітів через Інтернет, що сприятиме зручності та швидкості процесу покупки та доставки.

Завданням кваліфікаційної роботи є:

- аналіз потреб ринку квіткового бізнесу в онлайн-середовищі;
- визначення вимог;
- розробка та реалізація функціоналу вебсайту;
- визначення можливостей подальшого розвитку та покращення.

Сфера застосування розробленого вебсайту охоплює всіх зацікавлених учасників квіткового бізнесу: від покупців, які шукають зручний спосіб придбати квіти, до підприємств, які пропонують послуги доставки та реалізації квітів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

Квітковий бізнес є однією з найбільш традиційних та водночас динамічних галузей в економіці. Ця галузь має корені у культурі та мистецтві, що додає їй особливий символічний зміст. Ще з історичних часів квіти вважались виразом кохання, шанування та радості. Протягом століть вони використовуються як подарунки, символічні жести та вирази емоцій.

З появою Інтернету та розвитком електронної комерції квітковий бізнес відчув необхідність в адаптації до нових технологій та способів спілкування з клієнтами. Онлайн-продажі квітів стали популярними через свою зручність та доступність. Вони надають можливість замовити квіти з будь-якого місця та в будь-який час, не виходячи з дому. Це особливо зручно для тих людей, які не мають часу відвідувати фізичні квіткові магазини.

Основні чинники, що впливають на ринок квітів в онлайн-сфері, включають різноманітність товарів, зручність доставки, конкуренцію, дизайн та безпеку покупок. Зручність та доступність є ключовими факторами успіху онлайн-торгівлі квітами. Клієнти цінують можливість перегляду широкого асортименту товарів та замовлення їх з будь-якого пристрою. Доставка є ще однією важливою перевагою, оскільки вона дозволяє замовникам надсилати квіти своїм близьким та коханим, навіть якщо вони знаходяться в іншому місті чи країні.

Зростання популярності онлайн-торгівлі квітами призвело до збільшення конкуренції серед постачальників. Для успішного функціонування в цій галузі необхідно пропонувати якісний товар та ефективно просувати свої послуги в інтернеті. Окрім того, дизайн та представлення товарів грають важливу роль у приверненні уваги клієнтів. Привабливий та зручний інтерфейс дозволяє зробити покупку приємним та легким процесом.

Безпека та конфіденційність є ще однією важливою аспектом для клієнтів онлайн-торгівлі квітами. Передача особистих даних та інформації про кредитні картки вимагає надійного захисту від шахраїв та кіберзлочинців.

Урахування цих факторів у процесі розробки та впровадження вебсайту для доставки квітів дозволить забезпечити його успішне функціонування.

1.2 Аналіз готових рішень

На сучасний момент на ринку вебсайтів існує багато сайтів, що забезпечують доставку квітів по містах. Тому перед початком розробки власного вебсайту був проведений аналіз вже існуючих систем та виявлено наступні критерії:

Для аналізу готових рішень з доставки квітів можна використовувати такі критерії:

- 1) Асортимент квітів:
 - різноманітність та якість квітів, які доступні для замовлення;
 - наявність різних видів квітів (рожі, тюльпани, лілії тощо) та їх сортів.
- 2) Доставка:
 - швидкість та надійність доставки;
 - географічна доступність доставки (національна або міжнародна);
 - опції доставки (стандартна, експрес, вибір дати та часу).
- 3) Якість та свіжість квітів:
 - гарантія свіжості та довговічності квітів після доставки;
 - строк придатності та умови зберігання квітів під час доставки.
- 4) Цінова політика:
 - ціновий діапазон доступних квітів та букетів;
 - доступність акцій, знижок та спеціальних пропозицій.
- 5) Способи оплати:
 - різноманітність та зручність способів оплати (кредитні картки, платіжні системи, електронні гаманці тощо).
- 6) Сервіс та підтримка клієнтів:

- рівень сервісу та якість підтримки клієнтів (онлайн-консультації, живий чат, телефонна підтримка);

- реакція на запити та рішення проблем клієнтів.

7) Дизайн та інтерфейс користувача:

- зручність та інтуїтивність вебсайту або мобільного застосунку;

- естетика та привабливість дизайну.

8) Репутація та відгуки клієнтів:

- огляди та відгуки попередніх клієнтів;

- рейтинги та відгуки на незалежних вебсайтах або платформах.

Розглянуто наступні вебсервіси:

- Flowers.ua
- Buketland.ua
- Floristik.ua

1.2.1 Аналіз Flowers.ua

"Flowers.ua" — це онлайн-сервіс, який спеціалізується на доставці квітів та квіткових композицій у різних містах України та за її межами. Цей сервіс дозволяє клієнтам замовляти квіти через Інтернет та надсилати їх своїм рідним, друзям чи колегам на будь-який випадок чи свято. Flowers.ua пропонує широкий вибір квітів та букетів для різних ситуацій, включаючи святкування, вибачення, вирази кохання та інші події. Крім того, вони також надають послуги консультацій та підтримки клієнтів, щоб забезпечити задоволення та задоволення потреб кожного клієнта [1].

Інтерфейс вебсайту представлено на рис. 1.1.

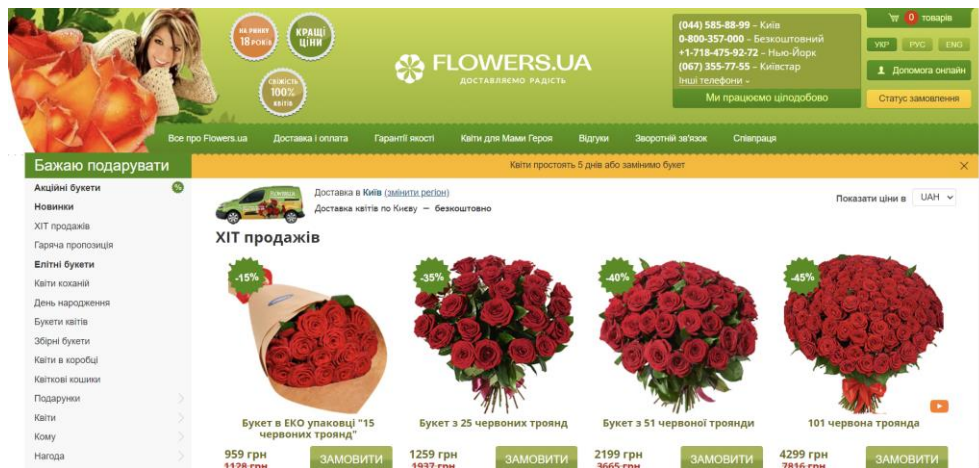


Рисунок 1.1 – Вебсайт Flowers.ua

Аналіз вебсайту "flowers.ua" за вказаними критеріями:

1) Асортимент квітів:

– "Flowers.ua" пропонує широкий асортимент квітів, включаючи різноманітні сорти рожевих, тюльпанів, хризантем, лілій та інших квітів. Також доступні різні варіанти композицій та букетів для різних випадків.

2) Доставка:

– сервіс пропонує доставку квітів по всій Україні та в деякі міжнародні країни. Доставка може бути здійснена як стандартно, так і експрес, з можливістю вказати дату та час доставки.

3) Якість та свіжість квітів:

– "Flowers.ua" гарантує свіжість та якість квітів, які використовуються в букетах. Квіти доставляються в спеціальних умовах для збереження свіжості.

4) Цінова політика:

– ціни на квіти на "Flowers.ua" варіюються в залежності від виду квітів та обраної композиції. Сайт регулярно пропонує акційні пропозиції та знижки на певні товари.

5) Способи оплати:

– вебсайт підтримує різноманітні способи оплати, включаючи оплату банківською картою, електронні гаманці та інші платіжні системи.

6) Сервіс та підтримка клієнтів:

– "Flowers.ua" надає швидку та якісну підтримку клієнтів через онлайн-

консультації, електронну пошту та телефонну підтримку. Відповіді на запитання надходять оперативно.

7) Дизайн та інтерфейс користувача:

– вебсайт має не зовсім сучасний та непривабливий дизайн, але зручний для користування та навігації. Простий інтерфейс дозволяє легко знайти потрібні товари та зробити замовлення.

8) Репутація та відгуки клієнтів:

– "Flowers.ua" має погану репутацію серед клієнтів через недоставку квітів, неповернення грошей і часте доставлення вже не свіжих букетів.

1.2.2 Аналіз Buketland.ua

Сайт "buketland.ua" є онлайн-сервісом з доставки квітів, який пропонує широкий асортимент квіткових композицій та букетів для різних випадків і подій. На цьому сайті клієнти можуть знайти квіткові аранжування з різних видів квітів, таких як рожі, тюльпани, хризантеми та інші, а також вибрати варіанти букетів з різних кольорів та стилів. Крім того, "buketland.ua" також пропонує послуги з доставки квітів у більшість міст України [2].

Інтерфейс вебсайту представлено на рис. 1.2.

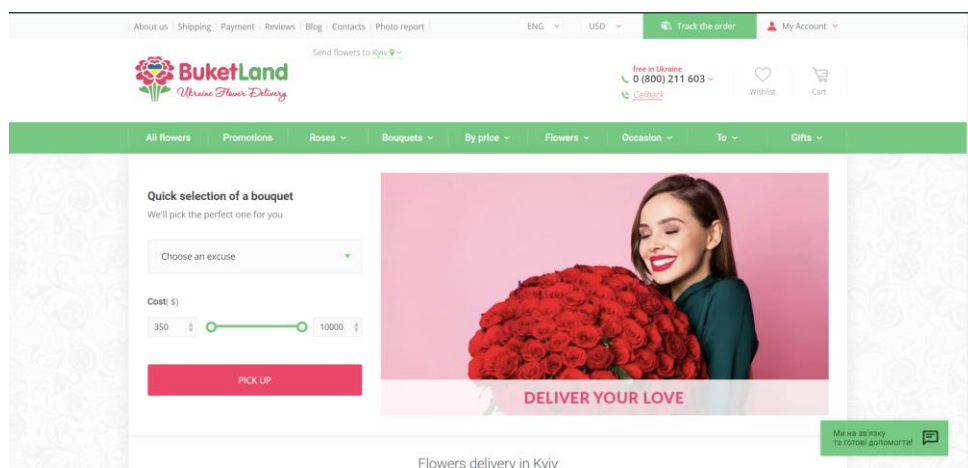


Рисунок 1.2 – Вебсайт buketland.ua

Тепер проаналізуємо сайт "buketland.ua" за раніше визначеними критеріями:

1) Асортимент квітів:

– "Buketland.ua" пропонує різноманіття квітів і букетів для різних ситуацій. В асортименті є як класичні букети, так і оригінальні квіткові композиції.

2) Доставка:

– сайт надає можливість доставки квітів в більшість міст України. Доставка може бути здійснена стандартним чи експрес-способом.

3) Якість та свіжість квітів:

– "Buketland.ua" гарантує свіжість та високу якість квітів у своїх композиціях. Квіти доставляються в спеціальних умовах, що дозволяє зберегти їх свіжість під час транспортування.

4) Цінова політика:

– ціни на квіти на "Buketland.ua" відповідають середньому рівню ринку. Сайт також пропонує акційні пропозиції та знижки на окремі товари.

5) Способи оплати:

– сервіс підтримує різноманітні способи оплати, включаючи оплату банківською картою, електронними гаманцями та інші платіжні системи.

6) Сервіс та підтримка клієнтів:

– "Buketland.ua" надає швидку та якісну підтримку клієнтів через онлайн-консультації, електронну пошту та телефонну підтримку.

7) Дизайн та інтерфейс користувача:

– вебсайт має привабливий та зручний дизайн, що спрощує процес пошуку квітів та замовлення.

8) Репутація та відгуки клієнтів:

– "Buketland.ua" має добру репутацію серед клієнтів і зазвичай отримує позитивні відгуки за якість квітів, швидку доставку та якісний сервіс.

1.2.3 Аналіз Floristik.ua

"Floristik.ua" — це онлайн-сервіс з доставки квітів, який пропонує великий вибір квіткових композицій, букетів та інших квіткових аксесуарів для різних випадків і подій. Сервіс надає можливість клієнтам замовляти та доставляти квіти

в різні міста України. Також "Floristik.ua" пропонує послуги консультацій та підтримки клієнтів, щоб забезпечити задоволення та задоволення потреб кожного клієнта [3].

Інтерфейс вебсайту представлено на рисунку 1.3.

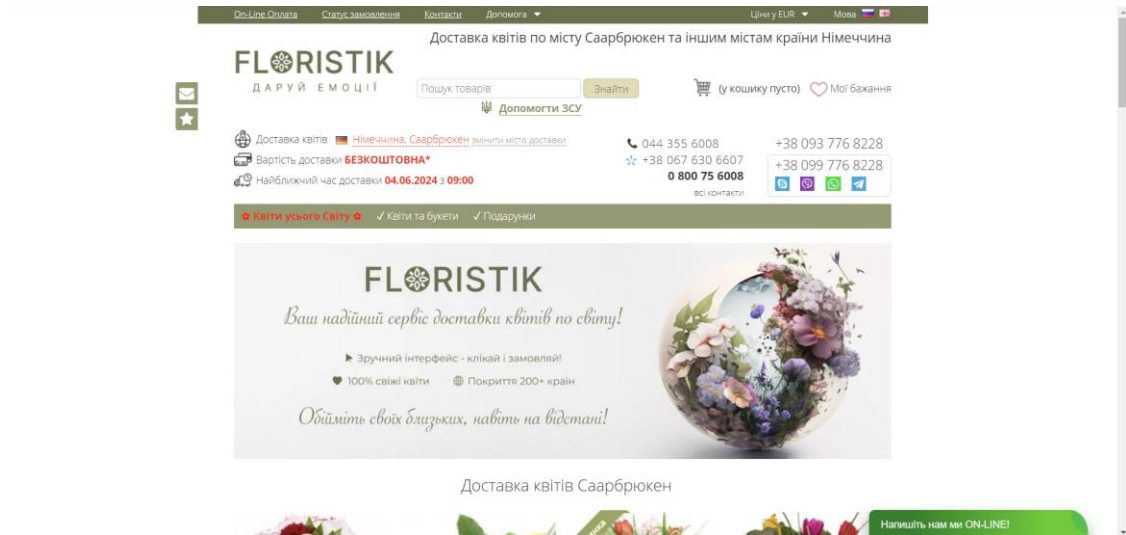


Рисунок 1.3 – Вебсайт buketland.ua

Тепер проаналізуємо сайт "floristik.ua" за раніше визначеними критеріями:

1) Асортимент квітів:

– "Floristik.ua" пропонує різноманіття квітів і букетів для різних ситуацій. В асортименті є як класичні букети, так і оригінальні квіткові композиції.

2) Доставка:

– сайт надає можливість доставки квітів в більшість міст України. Доставка може бути здійснена стандартним чи експрес-способом.

3) Якість та свіжість квітів:

– "Floristik.ua" гарантує свіжість та високу якість квітів у своїх композиціях. Квіти доставляються в спеціальних умовах, що дозволяє зберегти їх свіжість під час транспортування.

4) Цінова політика:

– ціни на квіти на "Floristik.ua" відповідають середньому рівню ринку. Сайт також пропонує акційні пропозиції та знижки на окремі товари.

5) Способи оплати:

– сервіс підтримує різноманітні способи оплати, включаючи оплату банківською картою, електронними гаманцями та інші платіжні системи.

6) Сервіс та підтримка клієнтів:

– "Floristik.ua" надає швидку та якісну підтримку клієнтів через онлайн-консультації, електронну пошту та телефонну підтримку.

7) Дизайн та інтерфейс користувача:

– вебсайт має привабливий та зручний дизайн, що спрощує процес пошуку квітів та замовлення.

8) Репутація та відгуки клієнтів:

– "Floristik.ua" має добру репутацію серед клієнтів і зазвичай отримує позитивні відгуки за якість квітів, швидку доставку та якісний сервіс.

1.3 Специфікація вимог

Потрібно розробити вебсайт, де користувач матиме змогу замовити доставку квітів улюбий час та улюбю локацію. Клієнтам будуть доступні наступні функції:

- реєстрація та авторизація користувачів;
- перегляд асортименту квітів;
- пошук квітів за різними критеріями;
- додавання квітів до кошика;
- оформлення замовлення;
- додавання квітів у улюблені;
- вибір методу доставки та оплати;
- підтвердження замовлення;
- відстеження статусу замовлення;
- опції зворотного зв'язку та підтримки клієнтів.

Вхідні дані:

- інформація для авторизації;
- інформація для реєстрації;

- інформація для замовлення.

Вихідні дані:

- результат реєстрації;
- результат входу;
- список квітів;
- список улюблених квітів користувача;
- історія замовлень користувача.

Висновки до розділу 1

В першому розділі кваліфікаційної роботи бакалавра розглянуто вебсайт доставки квітів, який пропонує зручний підхід до доставки та координації квітів.

Описано функціонал вебсайту, що спрямований на полегшення вибору квітів та оформлення доставки.

В аналізі існуючих систем також виявлено ключові аспекти та вдалі рішення, які можна врахувати під час розробки власної платформи. Проаналізовані Flowers.ua, Buketikland.ua та Floristik.ua як потенційні конкуренти на ринку продажу та доставки квітів.

2 ПРОЄКТУВАННЯ ВЕБСАЙТУ ДОСТАВКИ БУКЕТІВ

2.1 Розробка діаграми варіантів використання

Діаграма варіантів використання (Use Case Diagram) — це тип діаграми, яка показує функціональність системи, сценарії використання та взаємодію акторів (користувачів або інших систем) з цією системою. Основною метою діаграми варіантів використання є моделювання та візуалізація того, як користувачі взаємодіють з системою для досягнення певних цілей або виконання завдань.

Діаграма варіантів використання містить акторів (користувачі або зовнішні системи) та варіанти використання (use cases), які описують різні можливі взаємодії між акторами та системою. Варіанти використання відображаються у вигляді овалів, а актори - у вигляді піктограм людини або інших об'єктів.

Діаграма варіантів використання дозволяє команді на етапі проєктування системи краще зрозуміти потреби користувачів, ідентифікувати основні функції системи та визначити взаємодію між її частинами. Це корисний інструмент для аналізу та спілкування між розробниками, замовниками та іншими учасниками проєкту.

Діаграма варіантів використання вебсайту, представлено на рис. 2.1.



Рисунок 2.1 – Діаграма варіантів використання вебсайту доставки квітів

Діаграма демонструє ряд дій, які користувач може виконувати:

- реєстрація та авторизація користувачів: базовий процес для входу в систему, який дозволяє користувачам отримувати доступ до персоналізованих функцій;

- перегляд асортименту букетів: користувачі можуть переглядати різноманітність букетів, доступних для замовлення;
- пошук квітів за різними критеріями: це може включати пошук за типом квітів, ціною, нагодою тощо;
- додавання квітів до кошика та до улюблених: користувачі можуть зберігати букети, які їм подобаються, у кошику для покупок або в улюблених для швидкого доступу в майбутньому;
- оформлення замовлення: процес вибору квітів для покупки та введення даних для доставки;
- вибір методу оплати та доставки: користувачі можуть вибрати варіанти оплати та доставки, які найкраще відповідають їхнім потребам;
- підтвердження замовлення: завершальний крок процесу купівлі, де користувачі отримують підтвердження щодо деталей замовлення;
- встановлення статусу замовлення: система дозволяє користувачам перевіряти статус свого замовлення;
- опції зворотного зв'язку та підтримки клієнтів: можливість звернень до служби підтримки через особистий кабінет для швидкого вирішення будь-яких питань або проблем зі замовленням.

2.2 Розробка діаграми класів

Діаграма класів — це вид діаграми, в якій показано структуру системи або програми, що моделюється. Вона відображає класи системи, їх взаємозв'язки та властивості. Кожен клас представлений у вигляді прямокутника з розділенням на три частини: назва класу, його атрибути та методи. Стрілки між класами показують взаємозв'язки, наприклад, спадкування, агрегацію, композицію тощо. Діаграми класів широко використовуються під час аналізу, проєктування та реалізації програмних систем. Вони дозволяють команді розробників краще зрозуміти структуру системи, взаємозв'язки між її компонентами та спрощують процес розробки програмного забезпечення.

Діаграму класів для вебсайту доставки квітів, представлено на рис. 2.2.

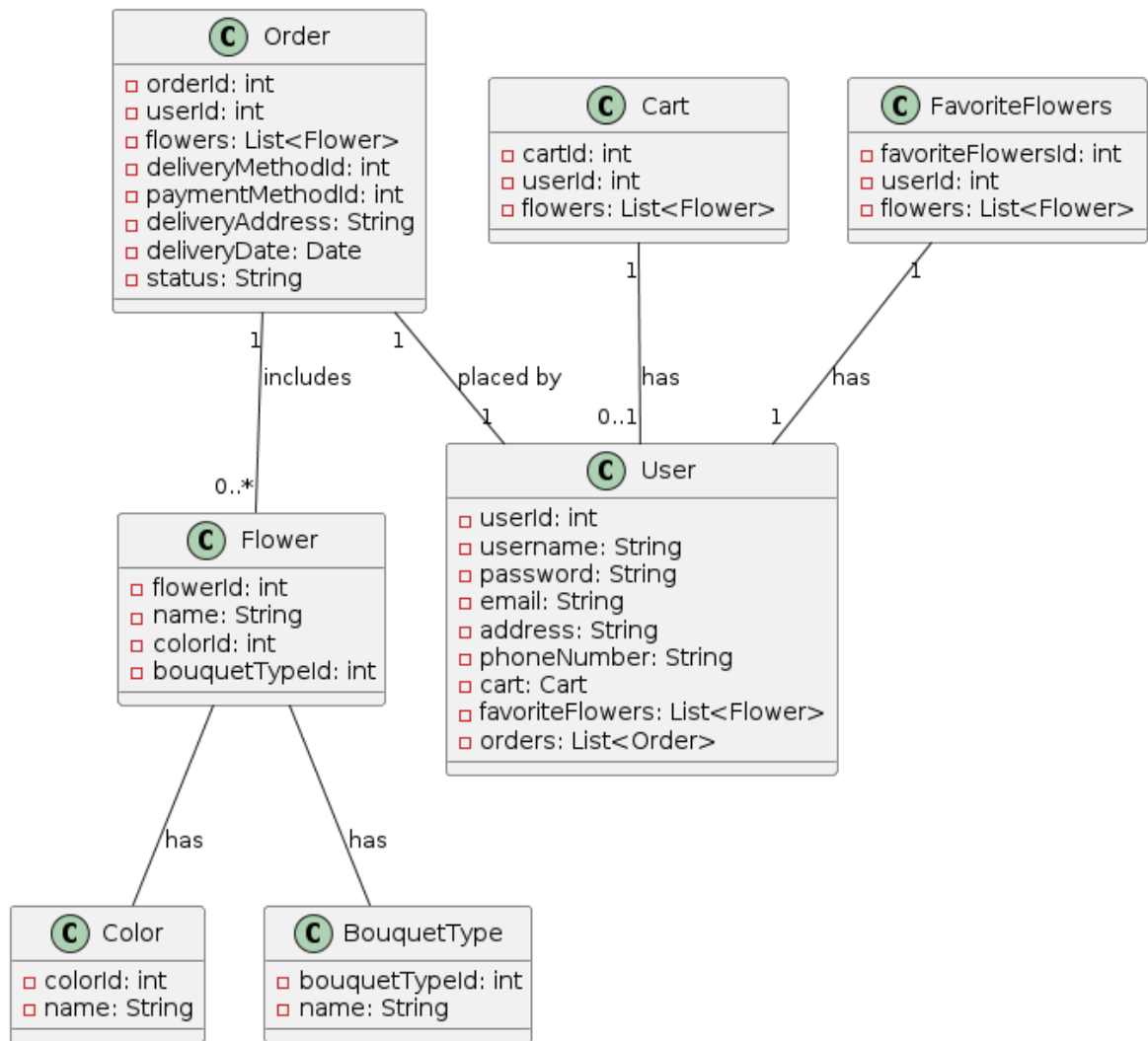


Рисунок 2.2 – Діаграма класів вебсайту

Розглянемо кожен клас окремо:

- 1) клас Flower: Представляє букет, його атрибути включають ідентифікатор, назву, ідентифікатор кольору та типу букету;
- 2) клас Color: Представляє колір квітки, містить інформацію про унікальний ідентифікатор та назву кольору;
- 3) клас BouquetType: Представляє тип букету, містить інформацію про унікальний ідентифікатор та назву типу;
- 4) клас Cart: Представляє кошик користувача, в якому можуть бути збережені квіткові композиції для подальшого замовлення;
- 5) клас FavoriteFlowers: Представляє список улюблених квітів користувача, де він може зберігати свої улюблені квіти для майбутнього перегляду та замовлення;

6) клас Order: Представляє замовлення користувача, включає інформацію про квітки, спосіб доставки, адресу доставки, дату, статус тощо;

7) клас User: Представляє користувача системи, містить інформацію про його ім'я, адресу електронної пошти, адресу, телефон, а також дані про його кошик, улюблені квіти та замовлення.

Класи мають наступні зв'язки:

- кожен об'єкт Flower має асоціації з об'єктами Color та BouquetType;
- кожен об'єкт Cart, FavoriteFlowers та Order має асоціацію з об'єктом User;
- кожен об'єкт Order має асоціацію з декількома об'єктами Flower.

2.3 Розробка діаграми послідовностей дій системи

Діаграма послідовності – це тип діаграми, який відображає послідовність взаємодій між об'єктами в системі в рамках конкретного сценарію або функціональності. Вона показує, як об'єкти взаємодіють між собою в певному порядку для виконання певного завдання або сценарію.

Діаграми послідовностей дозволяють моделювати взаємодію між об'єктами в системі в конкретних умовах та послідовностях. Вони широко використовуються в аналізі та проектуванні програмних систем для уточнення та визначення взаємодії між різними частинами системи.

Діаграма послідовності системи при оформленні замовлення представлена на рис. 2.3.

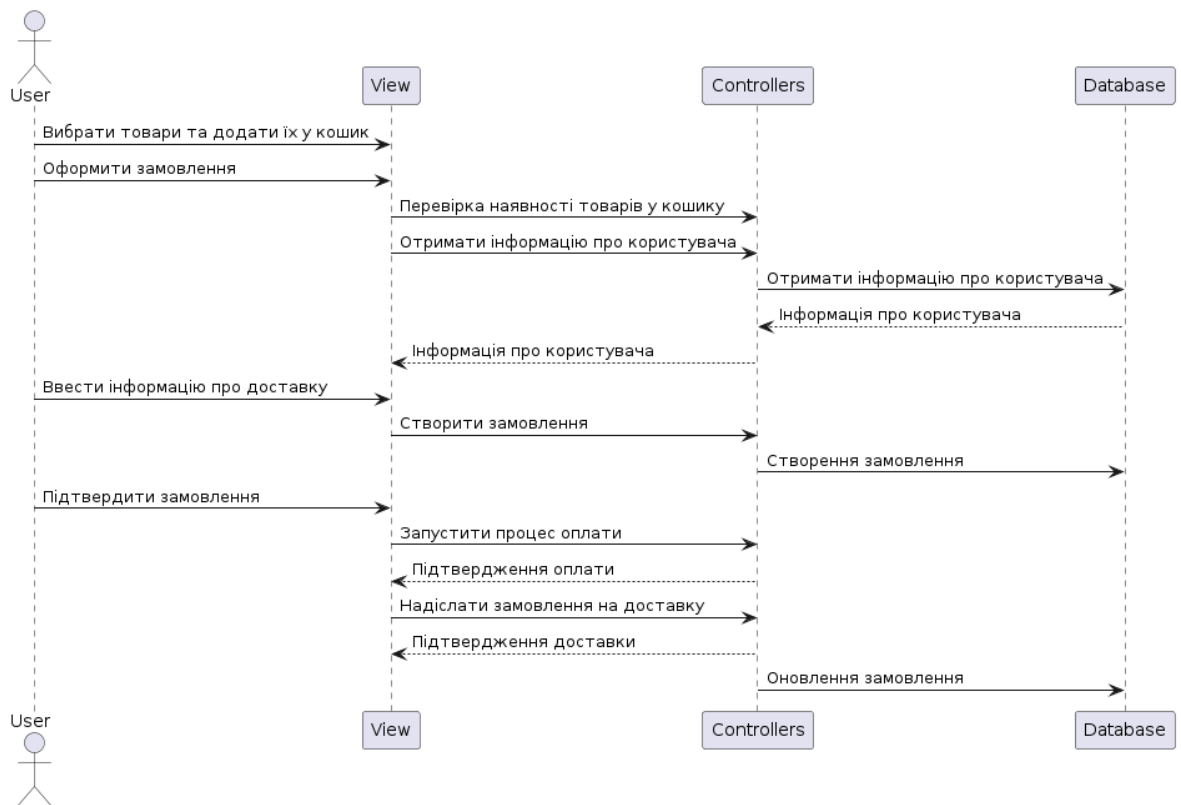


Рисунок 2.3 – Діаграма послідовності системи при оформленні замовлення

Діаграма послідовності відображає етапи обробки замовлення в системі вебсайту. Користувач взаємодіє з інтерфейсом, обираючи товари, вносячи інформацію про доставку та підтверджуючи замовлення. Процес оформлення замовлення включає перевірку наявності товарів у кошику, створення замовлення, ініціацію процесу оплати, та надсилання на доставку, що керується контролерами, які взаємодіють з базою даних для збереження та обробки даних. Ця діаграма демонструє інтеграцію між різними компонентами системи для забезпечення ефективного та злагодженого процесу покупки і доставки товарів.

2.4 Розробка моделі потоків даних системи

Модель потоків даних (DFD) — це методологія, що використовується для моделювання потоків даних та обробки інформації в системі. Вона дозволяє відобразити, як дані переміщуються через систему, як вони обробляються та взаємодіють з різними компонентами системи.

Діаграму потоків даних системи представлено на рис. 2.4.

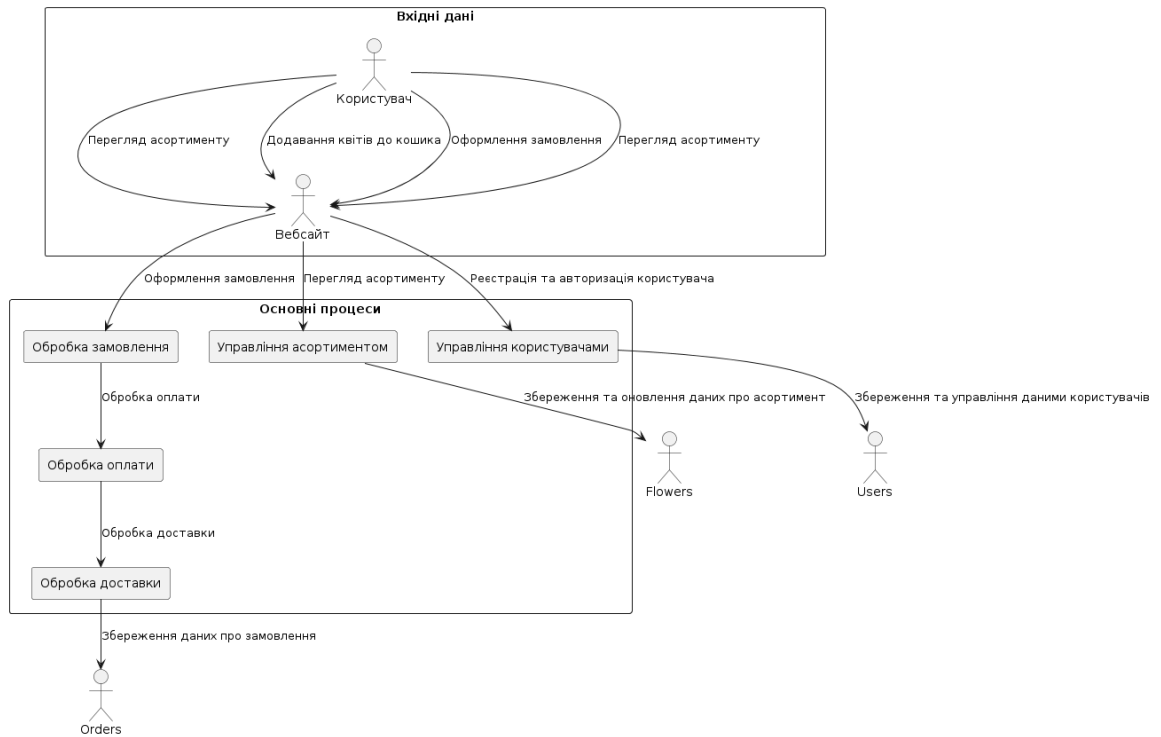


Рисунок 2.4 – Модель потоків даних системи

Діаграма чітко демонструє потоки даних між користувачами, базою даних та різними компонентами системи, що сприяє їх інтеграції та ефективному обміну інформацією, необхідною для функціональності вебсайту доставки букетів.

Висновки до розділу 2

У другому розділі кваліфікаційної роботи бакалавра спроектовано вебсайт доставки букетів.

- розроблено наступні моделі та діаграми:
- діаграму варіантів використання системи;
- діаграму класів системи;
- діаграму послідовностей оформлення замовлення;
- модель потоків даних системи.

3 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ ТА МОДЕЛЮВАННЯ БАЗИ ДАНИХ

3.1 Вибір технологій розробки

PHP (Hypertext Preprocessor) – це скриптова мова програмування, яка була створена для розробки динамічних вебсторінок і вебзастосунків. PHP розроблено Рамусом Лердорфом у 1994 році і спочатку називався "Personal Home Page Tools" (Інструменти для персональних домашніх сторінок), але з часом його можливості значно розширилися. Сьогодні PHP є основою мільйонів вебсайтів і є одним з найпоширеніших серверних мов програмування у світі [4].

Однією з ключових особливостей PHP є його інтеграція з HTML. PHP-код може бути включений безпосередньо у HTML-код сторінки, що дозволяє створювати динамічний контент безпосередньо у статичному шаблоні. Це робить PHP дуже зручним для швидкої розробки вебзастосунків, оскільки розробники можуть змішувати серверний та клієнтський код без необхідності в складній конфігурації [5].

PHP підтримує широкий спектр баз даних, включаючи MySQL, PostgreSQL, Oracle, Microsoft SQL Server, SQLite та інші. Саме це робить його універсальним інструментом для створення застосунків, які потребують зберігання та обробки великих обсягів даних. MySQL – одна з найпоширеніших баз даних, особливо чудово інтегрується з PHP. Тому ця комбінація використовується для створення багатьох популярних вебзастосунків, включаючи WordPress, Joomla та Drupal.

Окрім підтримки баз даних, PHP також має велику кількість вбудованих функцій та бібліотек для виконання різних задач. Це включає роботу з файлами, обробку форм, сесії, куки, управління користувачами та багато іншого. PHP також підтримує об'єктно-орієнтоване програмування (ООП), що дозволяє розробникам створювати складні та масштабовані додатки, використовуючи класи та об'єкти.

Однією з переваг PHP є висока продуктивність та можливість масштабування. PHP-код компілюється на льоту на сервері, що дозволяє швидко

обробляти запити і генерувати динамічний контент. PHP також підтримує різні механізми кешування, які можуть значно прискорити роботу додатків. Наприклад, APC (Alternative PHP Cache) та OPcache дозволяють зберігати зкомпільований PHP-код у пам'яті, зменшуючи навантаження на сервер.

PHP має активну спільноту розробників, яка постійно створює нові бібліотеки, фреймворки та інструменти, що розширюють можливості мови. Один з найпопулярніших фреймворків для PHP – Laravel, який забезпечує зручний та елегантний синтаксис для розробки вебзастосунків за архітектурним шаблоном MVC (Model-View-Controller). Laravel включає в себе багато корисних функцій, таких як маршрутизація, аутентифікація, міграції баз даних та багато іншого.

Безпека також є важливим аспектом PHP. Мова підтримує різні механізми для захисту додатків від поширених атак, таких як SQL-ін'єкції, XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery) та інші. Важливою частиною безпеки є належна валідація та фільтрація введених даних, а також використання безпечних механізмів зберігання паролів та інших конфіденційних даних.

PHP постійно розвивається, і нові версії мови виходять регулярно, додаючи нові функції та покращуючи продуктивність та безпеку. Наприклад, PHP 7, випущений у 2015 році, приніс значні покращення у продуктивності та використанні пам'яті порівняно з попередніми версіями. Останні версії PHP включають покращення в обробці помилок, підтримку типізації та багато інших корисних функцій.

У підсумку, PHP – це потужна, гнучка та універсальна скриптова мова, яка дозволяє швидко і ефективно створювати динамічні вебзастосунки. Завдяки своїй простоті, широкій підтримці баз даних, великій кількості вбудованих функцій та активній спільноті, PHP залишається однією з найпопулярніших мов для веброзробки у світі [6].

Laravel – це один з найпопулярніших і найпотужніших PHP-фреймворків, який був створений Тейлором Отвелом у 2011 році. Його основна мета – зробити процес розробки вебзастосунків більш елегантним, швидким та приємним для розробників. Laravel дотримується архітектурного шаблону Model-View-

Controller (MVC), що забезпечує розділення логіки додатка, його інтерфейсу користувача та контролерів, які обробляють вхідні запити та відповідають за бізнес-логіку.

Однією з найбільших переваг Laravel є його елегантний та зрозумілий синтаксис. Laravel надає безліч готових рішень для поширених задач, таких як аутентифікація користувачів, валідація введених даних, робота з базами даних, управління сесіями та кешування. Це дозволяє розробникам зосередитися на створенні унікального функціоналу свого додатка, не витрачаючи час на повторювані завдання.

Laravel включає в себе потужний ORM (Object-Relational Mapping) інструмент під назвою Eloquent. Eloquent дозволяє розробникам працювати з базами даних, використовуючи об'єктно-орієнтований підхід. Це значно спрощує взаємодію з базами даних, оскільки розробники можуть маніпулювати даними як об'єктами, а не писати складні SQL-запити. Eloquent також підтримує відносини між моделями, що дозволяє легко працювати з пов'язаними даними.

Ще однією важливою особливістю Laravel є його система маршрутизації. Laravel дозволяє легко визначати маршрути для свого додатка, прив'язуючи їх до відповідних контролерів та методів. Це забезпечує гнучкість та зручність у управлінні URL-адресами та обробці запитів. Крім того, Laravel підтримує RESTful маршрутизацію, що робить його ідеальним для створення API.

Laravel також має вбудовану систему міграцій баз даних, яка дозволяє розробникам легко керувати змінами у схемі бази даних. Міграції дозволяють створювати та змінювати таблиці бази даних за допомогою PHP-коду, що забезпечує контроль версій та полегшує спільну роботу над проектом. Крім того, Laravel підтримує модульні сидери, які дозволяють легко наповнювати базу даних тестовими даними.

Безпека є важливим аспектом Laravel. Фреймворк має вбудовані механізми захисту від поширених атак, таких як SQL-ін'єкції, XSS (Cross-Site Scripting) та CSRF (Cross-Site Request Forgery). Laravel забезпечує автоматичну валідацію та

фільтрацію введених даних, а також використовує захищені методи зберігання паролів та інших конфіденційних даних.

Laravel також підтримує модульне тестування, що дозволяє розробникам писати автоматизовані тести для свого додатка. Це значно покращує якість коду та забезпечує його стабільність при внесенні змін. Laravel включає інструменти для тестування як одиночних компонентів (unit testing), так і інтеграційних тестів, що охоплюють взаємодію між різними частинами додатка.

Ще однією потужною особливістю Laravel є його система черг (queues), яка дозволяє виконувати довготривалі задачі у фоновому режимі. Це забезпечує високу продуктивність додатка та зменшує час відгуку для користувачів. Laravel підтримує різні драйвери черг, включаючи Redis, Beanstalkd та інші, що робить його гнучким та масштабованим.

Laravel також має потужний механізм подій (events) та слухачів (listeners), які дозволяють розробникам легко керувати асинхронними задачами та взаємодіями у додатку. Це забезпечує високу гнучкість та розширюваність додатка, дозволяючи легко додавати нові функції та інтегрувати сторонні сервіси.

Завдяки своїй багатофункціональності, простоті використання та активній спільноті, Laravel став одним з найпопулярніших фреймворків для PHP. Він підходить як для створення невеликих вебсайтів, так і для розробки великих корпоративних додатків. Laravel продовжує активно розвиватися, додаючи нові можливості та покращуючи існуючі функції, що робить його надійним вибором для розробників вебзастосунків [7, 8].

Blade — це потужний шаблонний двигун, інтегрований у фреймворк Laravel, який забезпечує розробників зручним та елегантним способом створення динамічних HTML-шаблонів. Blade був створений, щоб зробити процес розробки вебінтерфейсів більш інтуїтивним та ефективним, пропонуючи простий у використанні синтаксис та велику кількість корисних функцій.

Одна з головних особливостей Blade полягає у його легкості та продуктивності. На відміну від інших шаблонних двигунів, Blade не вимагає попередньої компіляції шаблонів у PHP-код. Замість цього, Blade компілює

шаблони на льоту у звичайний PHP-код, що значно покращує продуктивність та забезпечує швидкий час відгуку. Це також означає, що Blade-шаблони легко інтегруються з існуючим PHP-кодом та бібліотеками.

Blade підтримує широкий спектр умовних виразів та циклів, що дозволяє розробникам легко управляти відображенням контенту на основі різних умов. Наприклад, розробники можуть використовувати умовні конструкції, такі як `@if`, `@else`, `@elseif` та `@endif` для контролю відображення блоків коду. Цикли, такі як `@for`, `@foreach`, `@forelse` та `@while`, дозволяють повторювати блоки коду для відображення списків даних чи інших повторюваних елементів.

Ще однією важливою особливістю Blade є його підтримка спадкування шаблонів. Blade дозволяє розробникам створювати основні шаблони, які містять загальні елементи інтерфейсу, такі як заголовки, підвали, навігаційні меню та інші компоненти, які повторюються на різних сторінках додатка. Ці основні шаблони можуть бути успадковані дочірніми шаблонами, які додають або змінюють конкретний контент для кожної сторінки. Це забезпечує кращу організацію коду та спрощує його підтримку.

Blade також підтримує компоненти та слоти, що дозволяє розробникам створювати повторно використовувані фрагменти коду. Компоненти Blade можуть містити як логіку, так і вигляд, і можуть бути використані у різних частинах додатка. Слоти дозволяють визначати області у компоненті, які можуть бути заповнені користувацьким контентом при його використанні. Це забезпечує високу гнучкість та розширюваність, дозволяючи легко створювати складні та адаптивні інтерфейси.

Однією з важливих функцій Blade є його здатність до безпечної вставки даних у HTML-код. Blade автоматично екранує дані, вставлені у шаблон, що забезпечує захист від XSS-атак (Cross-Site Scripting). Це значно покращує безпеку додатка та знижує ризик виконання шкідливого коду.

Blade також легко інтегрується з JavaScript-фреймворками, такими як Vue.js та React, що робить його дуже гнучким інструментом для розробки сучасних вебзастосунків. Blade дозволяє розробникам вставляти JavaScript-код у шаблони

та використовувати дані з PHP-коду у JavaScript, що забезпечує високу інтерактивність та динамічність інтерфейсів.

Завдяки своїй простоті, елегантності та потужності, Blade став одним з найпопулярніших шаблонних двигунів серед розробників Laravel. Він забезпечує високу продуктивність, безпеку та гнучкість, що робить його ідеальним вибором для створення динамічних та адаптивних вебінтерфейсів. Blade продовжує активно розвиватися, додаючи нові можливості та покращуючи існуючі функції, що робить його надійним та сучасним інструментом для розробки вебзастосунків [9].

TailwindCSS – це утилітарний CSS-фреймворк, який ставить собі за мету зробити процес стилізації вебзастосунків швидким та ефективним. Створений Адамом Ветаном і Стівом Шрагером, TailwindCSS відрізняється від традиційних CSS-фреймворків, таких як Bootstrap або Foundation, своєю філософією утилітарних класів, де кожен клас відповідає за конкретний стильовий атрибут.

TailwindCSS надає розробникам великий набір готових утилітарних класів, які дозволяють легко застосовувати стилі до HTML-елементів без необхідності писати власний CSS-код. Це значно знижує кількість написаного CSS та покращує його підтримку. Розробники можуть швидко створювати складні та адаптивні інтерфейси, використовуючи лише HTML та класи TailwindCSS. Наприклад, класи для управління відступами, кольорами, шрифтами, розмірами та іншими стилями роблять процес стилізації інтуїтивним та гнучким.

Однією з ключових переваг TailwindCSS є його гнучкість та можливість налаштування. Розробники можуть змінювати конфігураційний файл TailwindCSS, щоб налаштувати кольорові схеми, шрифти, відступи та інші параметри відповідно до своїх потреб. Це дозволяє створювати унікальні стилі, які відповідають дизайну проєкту, без необхідності переписувати існуючі класи. TailwindCSS також підтримує модульність, що дозволяє розробникам додавати або видаляти класи в залежності від вимог проєкту.

Ще однією важливою особливістю TailwindCSS є його підтримка JIT (Just-In-Time) компіляції. JIT-компілятор генерує CSS-код на льоту під час розробки, що

значно зменшує розмір кінцевого CSS-файлу та покращує продуктивність додатка. JIT-компіляція дозволяє розробникам використовувати тільки ті класи, які дійсно потрібні для проєкту, що знижує обсяг непотрібного CSS-коду.

TailwindCSS також підтримує адаптивний дизайн завдяки своїй системі брейкпоінтів. Розробники можуть легко створювати адаптивні інтерфейси, використовуючи класи TailwindCSS для визначення стилів на різних розмірах екранів. Це дозволяє створювати інтерфейси, які добре виглядають і працюють на різних пристроях, від мобільних телефонів до настільних комп'ютерів.

TailwindCSS інтегрується з багатьма популярними інструментами та фреймворками, що робить його універсальним інструментом для веброзробників. Наприклад, TailwindCSS легко інтегрується з React, Vue.js, Angular та іншими JavaScript-фреймворками, що дозволяє розробникам використовувати утилітарні класи для стилізації компонентів. TailwindCSS також підтримує роботу з CSS-in-JS рішеннями, такими як Emotion та Styled Components, що робить його дуже гнучким та зручним у використанні.

Завдяки своїй простоті, гнучкості та потужності, TailwindCSS став одним з найпопулярніших CSS-фреймворків серед веброзробників. Він забезпечує високу продуктивність, адаптивність та можливість налаштування, що робить його ідеальним вибором для створення сучасних вебінтерфейсів. TailwindCSS продовжує активно розвиватися, додаючи нові можливості та покращуючи існуючі функції, що робить його надійним та сучасним інструментом для розробки вебзастосунків[10].

MySQL – це одна з найпоширеніших та найпопулярніших систем керування базами даних (СКБД), яка використовується для зберігання та управління даними у багатьох вебзастосунках. Вперше розроблена шведською компанією MySQL AB у 1995 році, MySQL здобула велику популярність завдяки своїй продуктивності, надійності та простоті у використанні. З 2008 року MySQL належить Oracle Corporation, яка продовжує розвивати та підтримувати цю СКБД.

MySQL є системою керування реляційними базами даних (РСУБД), що означає, що дані зберігаються у таблицях, які можуть бути пов'язані між собою за

допомогою ключів. Це забезпечує структуроване зберігання даних та дозволяє ефективно виконувати складні запити. MySQL підтримує мову SQL (Structured Query Language), яка використовується для створення, читання, оновлення та видалення даних у базах даних.

Однією з головних переваг MySQL є її висока продуктивність та масштабованість. MySQL здатна обробляти великі обсяги даних та високі навантаження, що робить її ідеальним вибором для вебзастосунків з великим трафіком, таких як соціальні мережі, електронна комерція та інші. MySQL підтримує різні механізми зберігання даних, включаючи InnoDB та MyISAM, що дозволяє налаштовувати базу даних відповідно до вимог проєкту.

InnoDB є основним механізмом зберігання даних у MySQL, який забезпечує підтримку транзакцій, зовнішніх ключів та інших важливих функцій для забезпечення цілісності та надійності даних. MyISAM, інший популярний механізм зберігання, забезпечує високу продуктивність для читання даних, але не підтримує транзакції. Це робить його підходящим для додатків, де читання даних є пріоритетом, а вимоги до цілісності даних не є критичними.

MySQL підтримує різні типи індексів, включаючи первинні ключі, унікальні індекси та індекси на основі колонок, що значно покращує продуктивність запитів. Індекси дозволяють швидко знаходити та сортувати дані, що особливо важливо для великих баз даних з великою кількістю записів. MySQL також підтримує повнотекстовий пошук, що дозволяє виконувати складні пошукові запити за текстовими даними.

MySQL має вбудовані механізми для забезпечення безпеки даних, включаючи контроль доступу, шифрування даних та резервне копіювання. Контроль доступу дозволяє налаштовувати права користувачів та груп користувачів, обмежуючи доступ до бази даних лише авторизованим користувачам. Шифрування даних забезпечує захист конфіденційних даних як у стані спокою, так і під час передачі. Резервне копіювання дозволяє створювати резервні копії бази даних для відновлення у випадку втрати або пошкодження даних.

MySQL є кросплатформеною системою, яка підтримує роботу на різних операційних системах, включаючи Windows, Linux, macOS та інші. Це робить MySQL універсальним інструментом для розробників, які можуть використовувати його у своїх проектах незалежно від вибору платформи. MySQL також легко інтегрується з різними мовами програмування, включаючи PHP, Java, Python, C# та інші, що робить його популярним вибором для веброзробників.

Завдяки своїй надійності, продуктивності та гнучкості, MySQL став однією з найпопулярніших СКБД у світі. Вона використовується у багатьох відомих проектах, включаючи Facebook, Twitter, YouTube, WordPress та багато інших. MySQL продовжує активно розвиватися, додаючи нові функції та покращуючи існуючі, що робить її надійним вибором для розробки та підтримки сучасних вебзастосунків.

OpenServer Panel – це програмний комплекс для розробки, тестування та налагодження вебзастосунків на локальному комп'ютері. Створений російськими розробниками, OpenServer Panel об'єднує у собі кілька важливих компонентів, включаючи вебсервер Apache або Nginx, базу даних MySQL або PostgreSQL, інтерпретатор PHP та інші інструменти, необхідні для створення та тестування вебзастосунків.

Однією з ключових особливостей OpenServer Panel є його простота у використанні та налаштуванні. Він надає зручний графічний інтерфейс, який дозволяє розробникам легко керувати сервісами, створювати нові віртуальні хости, керувати базами даних та виконувати інші задачі, пов'язані з розробкою вебзастосунків. OpenServer Panel також підтримує різні версії PHP, що дозволяє тестувати сумісність додатків з різними версіями інтерпретатора.

OpenServer Panel включає у себе додаткові інструменти, такі як phpMyAdmin для управління базами даних MySQL та PgAdmin для PostgreSQL, що забезпечує зручне керування базами даних. Він також підтримує різні мови програмування та фреймворки, що робить його універсальним інструментом для веброзробників, які бажають швидко розгорнути та тестувати свої проекти на локальному сервері.

Однією з переваг OpenServer Panel є його гнучкість та можливість налаштування. Розробники можуть налаштовувати конфігураційні файли Apache, Nginx, PHP та інших компонентів відповідно до своїх потреб. Це дозволяє створювати середовище, яке максимально наближене до виробничого, що полегшує процес розробки та тестування.

OpenServer Panel також має вбудовані інструменти для резервного копіювання та відновлення даних, що забезпечує безпеку та надійність даних під час розробки. Розробники можуть створювати резервні копії баз даних та файлів проекту, а також легко відновлювати їх у випадку втрати або пошкодження. Це забезпечує додатковий рівень захисту даних та знижує ризик втрати важливої інформації.

Ще однією важливою особливістю OpenServer Panel є його підтримка роботи з різними версіями компонентів. Розробники можуть легко перемикатися між різними версіями PHP, MySQL, Apache або Nginx, що дозволяє тестувати сумісність додатків з різними конфігураціями серверного середовища. Це особливо корисно для великих проектів, які мають підтримуватися на різних платформах та версіях програмного забезпечення.

OpenServer Panel також підтримує інтеграцію з іншими інструментами для розробки, такими як системи контролю версій (наприклад, Git), що забезпечує зручну спільну роботу над проектом. Розробники можуть легко керувати версіями коду, створювати гілки та злиття, а також відслідковувати зміни у коді. Це забезпечує кращу організацію роботи та підвищує продуктивність команди.

Завдяки своїй простоті, гнучкості та багатофункціональності, OpenServer Panel став популярним серед веброзробників, які шукають простий та ефективний спосіб налаштування локального серверного середовища. Він забезпечує зручний інтерфейс для керування сервісами, підтримку різних версій програмного забезпечення, резервне копіювання та інтеграцію з іншими інструментами, що робить його надійним вибором для розробки та тестування вебзастосунків. OpenServer Panel продовжує активно розвиватися, додаючи нові можливості та

покращуючи існуючі функції, що робить його сучасним та зручним інструментом для веброзробників.

3.2 Моделювання бази даних системи

Фізична модель бази даних (Physical Database Model) – це детальне представлення структури бази даних, яке описує, як дані будуть зберігатися на фізичному рівні. Вона містить точні визначення всіх таблиць, стовпців, типів даних, індексів, обмежень (ключів) і інших елементів бази даних. Фізична модель бази даних зазвичай створюється на основі логічної моделі бази даних, яка є більш абстрактною і зосереджена на структурі та відносинах даних.

Фізична модель вебсайту доставляння букетів представлено на рис. 3.1.

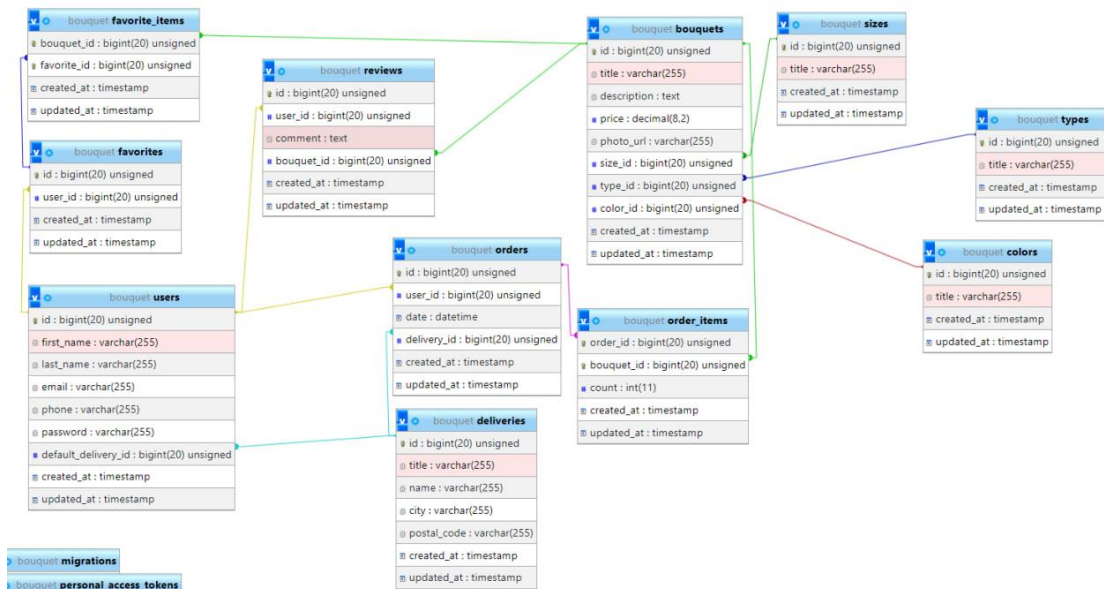


Рисунок 3.1 – Фізична модель бази даних

Розглянемо кожну таблицю окремо:

Таблиця bouquets зберігає інформацію про букети, доступні в магазині.

Структура таблиці Вид відносин

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	3 description	text	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	4 price	decimal(8,2)			Ні	Немає			Більше
<input type="checkbox"/>	5 photo_url	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	6 size_id	bigint(20)		UNSIGNED	Ні	Немає			Більше
<input type="checkbox"/>	7 type_id	bigint(20)		UNSIGNED	Ні	Немає			Більше
<input type="checkbox"/>	8 color_id	bigint(20)		UNSIGNED	Ні	Немає			Більше
<input type="checkbox"/>	9 created_at	timestamp			Так	NULL			Більше
<input type="checkbox"/>	10 updated_at	timestamp			Так	NULL			Більше

Позначити все Вибрані:

Рисунок 3.2 – Структура таблиці букетів

Таблиця colors зберігає інформацію про доступні кольори.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	3 created_at	timestamp			Так	NULL			Більше
<input type="checkbox"/>	4 updated_at	timestamp			Так	NULL			Більше

Рисунок 3.3 – Структура таблиці кольори

Таблиця deliveries зберігає інформацію про доступні способи доставки.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	3 name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	4 city	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	5 postal_code	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
<input type="checkbox"/>	6 created_at	timestamp			Так	NULL			Більше
<input type="checkbox"/>	7 updated_at	timestamp			Так	NULL			Більше

Рисунок 3.4 – Структура таблиці доставки

Таблиця favorites

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id 🔑	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2 user_id 🔑	bigint(20)		UNSIGNED	Ні	Немає			Більше ▾
<input type="checkbox"/>	3 created_at	timestamp			Так	NULL			Більше ▾
<input type="checkbox"/>	4 updated_at	timestamp			Так	NULL			Більше ▾

Рисунок 3.5 – Структура таблиці улюблені

Таблиця favorite_items зберігає інформацію про елементи в списках улюблених букетів.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 bouquet_id 🔑	bigint(20)		UNSIGNED	Ні	Немає			Більше ▾
<input type="checkbox"/>	2 favorite_id 🔑	bigint(20)		UNSIGNED	Ні	Немає			Більше ▾
<input type="checkbox"/>	3 created_at	timestamp			Так	NULL			Більше ▾
<input type="checkbox"/>	4 updated_at	timestamp			Так	NULL			Більше ▾

Рисунок 3.6 – Структура таблиці улюблені елементи

Таблиця orders зберігає інформацію про замовлення користувачів.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id 🔑	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2 user_id 🔑	bigint(20)		UNSIGNED	Ні	Немає			Більше ▾
<input type="checkbox"/>	3 date	datetime			Ні	Немає			Більше ▾
<input type="checkbox"/>	4 delivery_id 🔑	bigint(20)		UNSIGNED	Ні	Немає			Більше ▾
<input type="checkbox"/>	5 created_at	timestamp			Так	NULL			Більше ▾
<input type="checkbox"/>	6 updated_at	timestamp			Так	NULL			Більше ▾

Рисунок 3.7 – Структура таблиці замовлення

Таблиця order_items зберігає інформацію про елементи в замовленнях.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 order_id 🔗	bigint(20)		UNSIGNED	Ні	Немає			Більше ▾
<input type="checkbox"/>	2 bouquet_id 🔗 🔗	bigint(20)		UNSIGNED	Ні	Немає			Більше ▾
<input type="checkbox"/>	3 count	int(11)			Ні	Немає			Більше ▾
<input type="checkbox"/>	4 created_at	timestamp			Так	NULL			Більше ▾
<input type="checkbox"/>	5 updated_at	timestamp			Так	NULL			Більше ▾

Рисунок 3.8 – Структура таблиці елементи замовлення

Таблиця reviews зберігає відгуки користувачів про букети.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id 🔗	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2 user_id 🔗	bigint(20)		UNSIGNED	Ні	Немає			Більше ▾
<input type="checkbox"/>	3 comment	text	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	4 bouquet_id 🔗	bigint(20)		UNSIGNED	Ні	Немає			Більше ▾
<input type="checkbox"/>	5 created_at	timestamp			Так	NULL			Більше ▾
<input type="checkbox"/>	6 updated_at	timestamp			Так	NULL			Більше ▾

Рисунок 3.9 – Структура таблиці відгуки

Таблиця sizes зберігає інформацію про розміри букетів.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id 🔗	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	3 created_at	timestamp			Так	NULL			Більше ▾
<input type="checkbox"/>	4 updated_at	timestamp			Так	NULL			Більше ▾

Рисунок 3.10 – Структура таблиці розміри

Таблиця types зберігає інформацію про типи букетів.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id 🔗	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	3 created_at	timestamp			Так	NULL			Більше ▾
<input type="checkbox"/>	4 updated_at	timestamp			Так	NULL			Більше ▾

Рисунок 3.11 – Структура таблиці типи

Таблиця users зберігає інформацію про користувачів.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id 🔑	bigint(20)		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше ▾
<input type="checkbox"/>	2 first_name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	3 last_name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	4 email	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	5 phone	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	6 password	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше ▾
<input type="checkbox"/>	7 default_delivery_id 🔑	bigint(20)		UNSIGNED	Так	NULL			Більше ▾
<input type="checkbox"/>	8 created_at	timestamp			Так	NULL			Більше ▾
<input type="checkbox"/>	9 updated_at	timestamp			Так	NULL			Більше ▾

Рисунок 3.12 – Структура таблиці користувачі

3.3 Розробка мокапу системи

Мокап системи – це імітація вигляду та функціональності програмного забезпечення, що створюється. Він служить інструментом для візуалізації та концептуалізації інтерфейсу користувача, функцій та потоків роботи системи.

Створено наступні мокапи сторінок:

- головну сторінку застосунку, представлено на рис. 3.13;
- сторінку каталогу букетів, представлено на рис. 3.14;
- сторінку про заклад, представлено на рис. 3.15;
- сторінку з поширеними питаннями, представлено на рис. 3.16;
- сторінку для певного букету, представлено на рис. 3.17;
- сторінку кошику, представлено на рис. 3.18;
- сторінку оформлення замовлення, представлено на рис. 3.19;
- сторінку улюблених букетів, представлено на рис. 3.20;
- сторінку контактів, представлено на рис. 3.21.

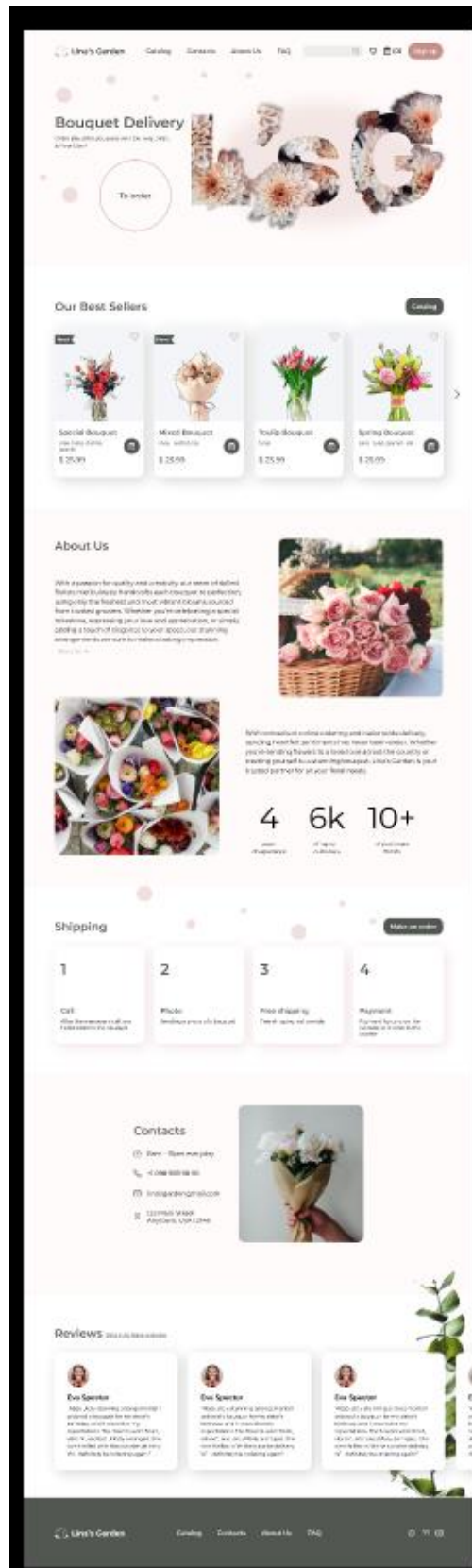


Рисунок 3.13 – Мокап головної сторінки застосунку

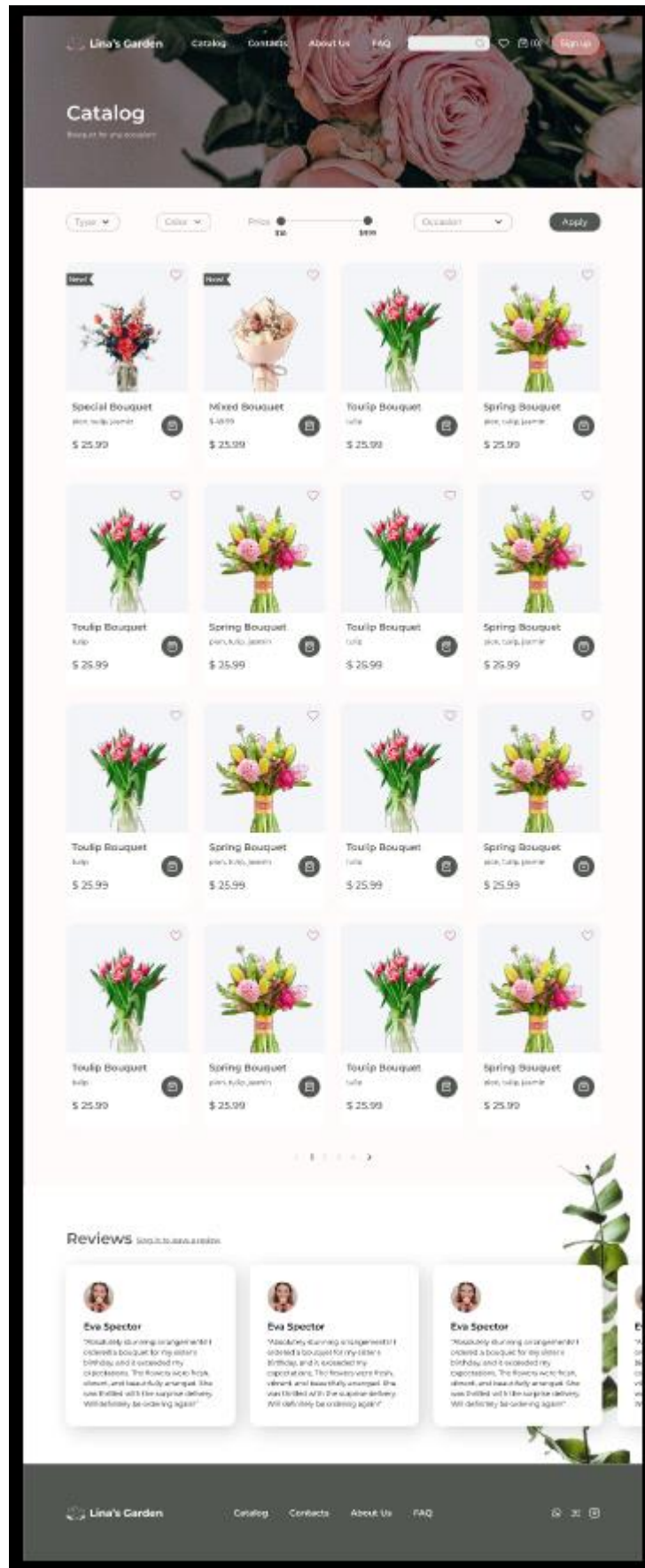


Рисунок 3.14 – Мокап сторінки каталогу букетів

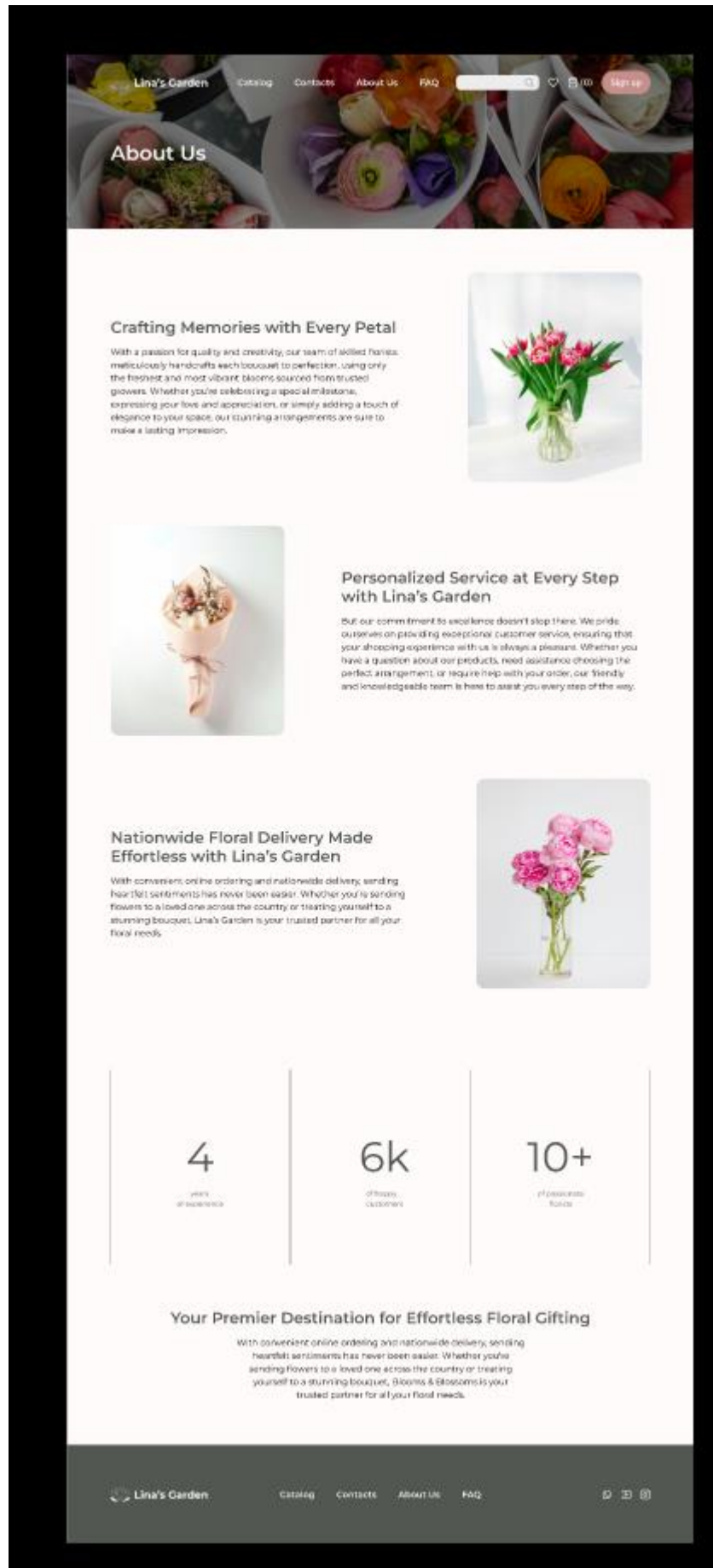


Рисунок 3.15 – Мокап сторінки про заклад

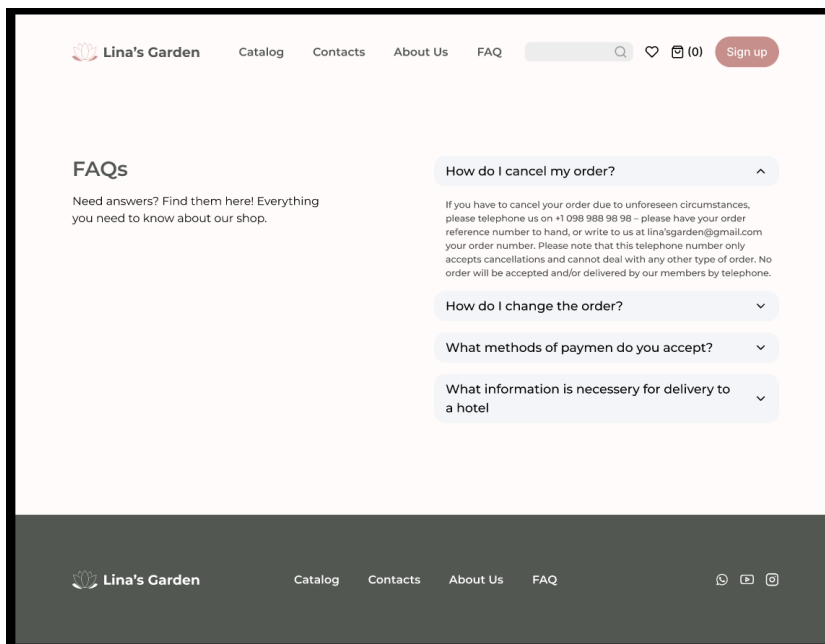


Рисунок 3.16 – Мокап сторінки з поширеними питаннями

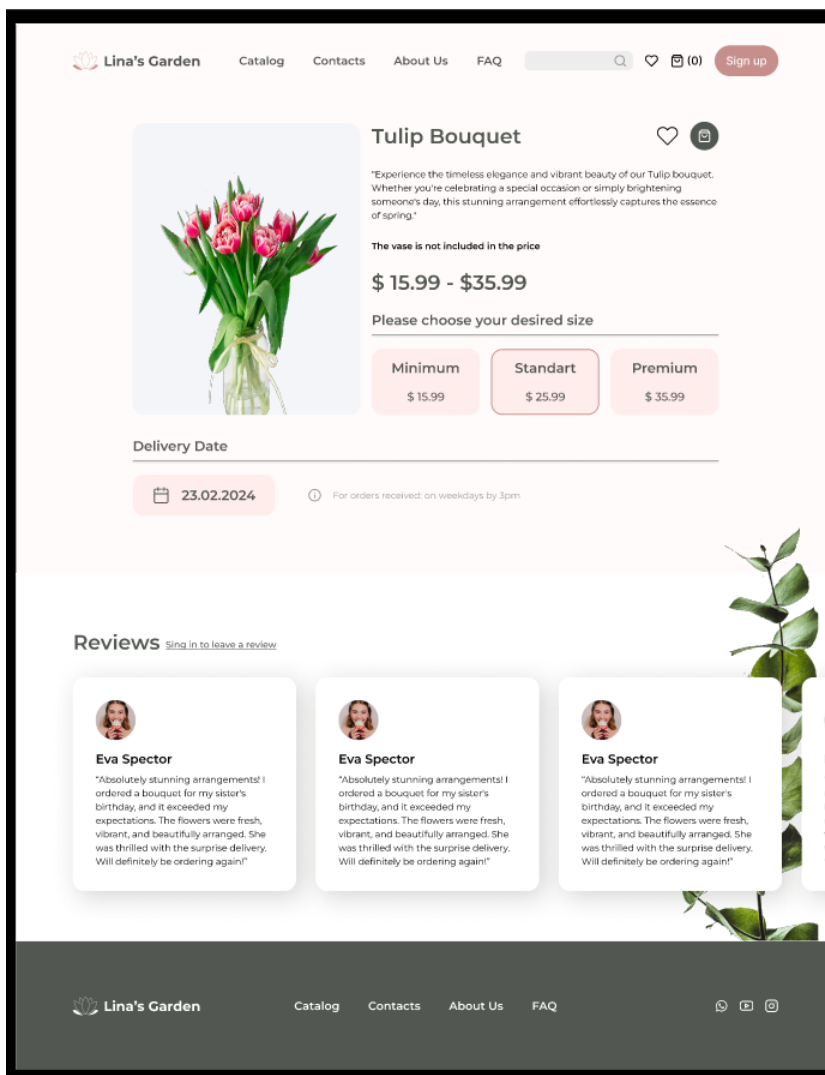


Рисунок 3.17 – Мокап сторінки для певного букету

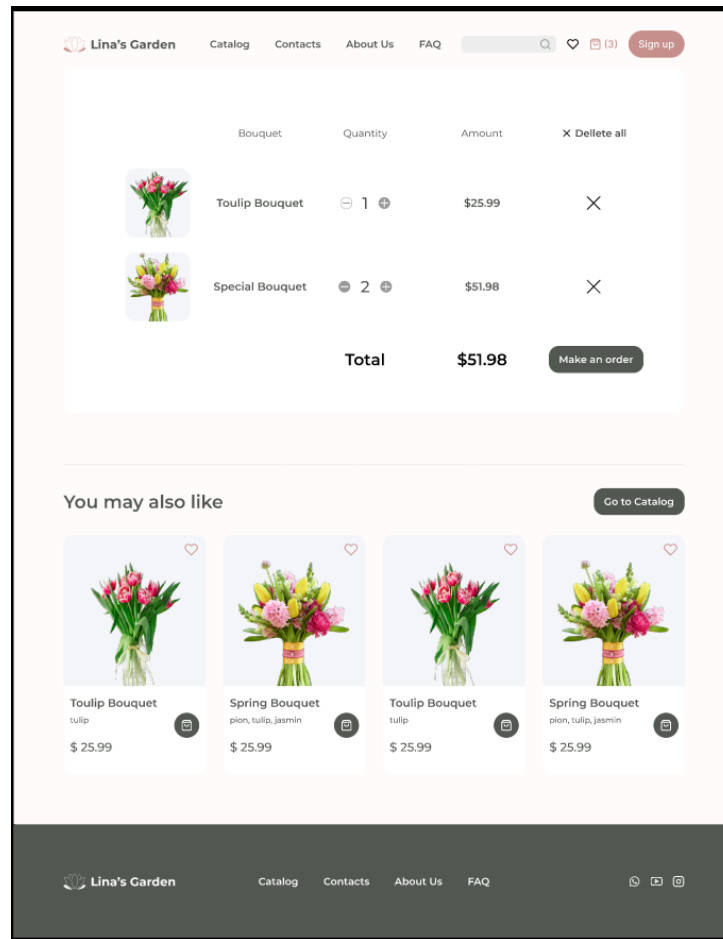


Рисунок 3.18 – Мокап сторінки кошику

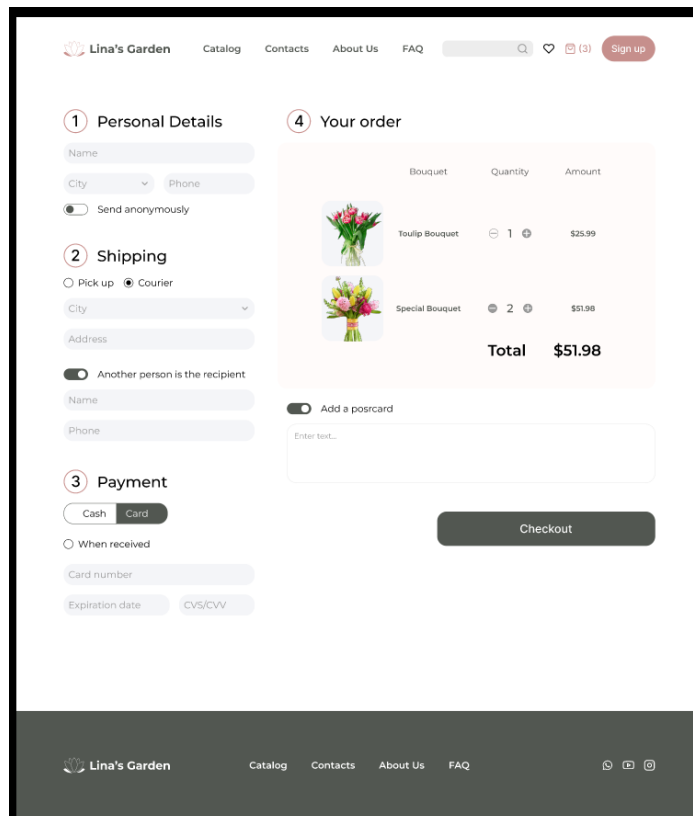


Рисунок 3.19 – Мокап сторінки оформлення замовлення

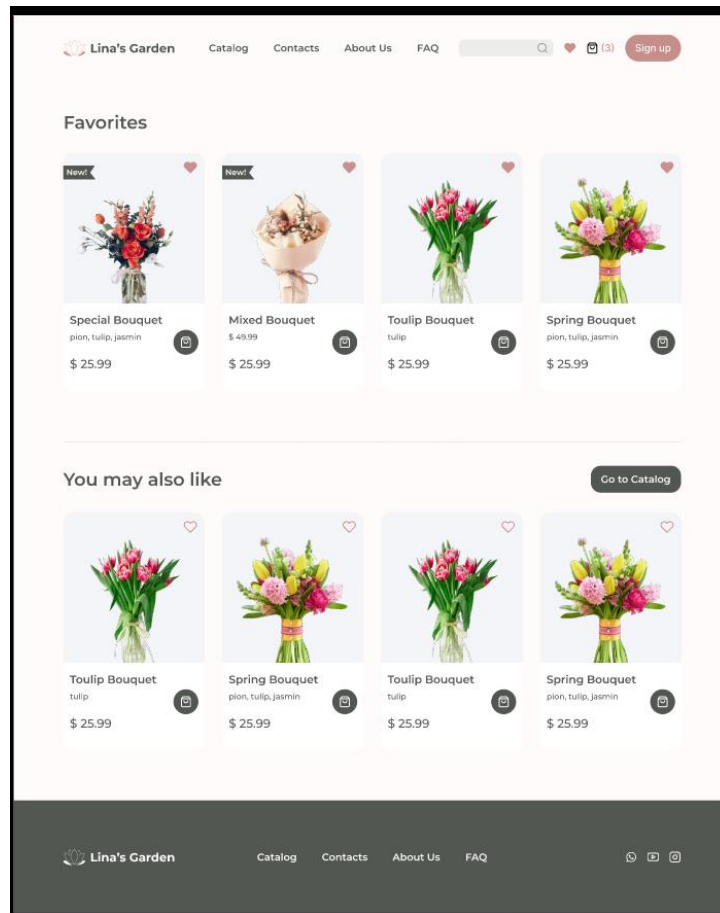


Рисунок 3.20 – Мокап сторінки улюблених букетів

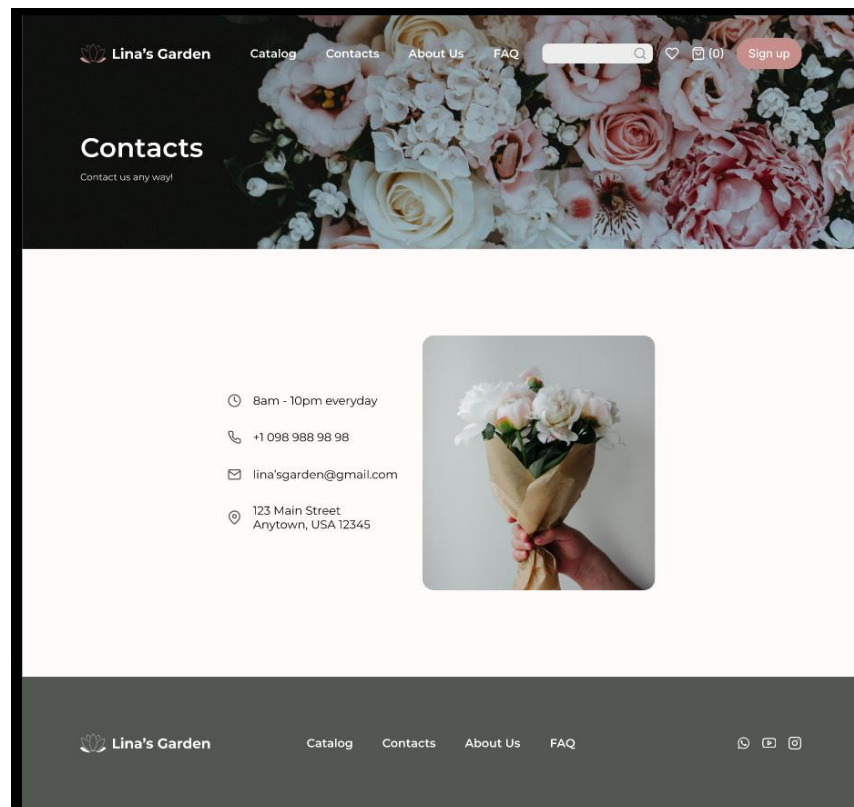


Рисунок 3.21 – Мокап сторінки контактів

Розробка цих мокапів забезпечила чітке розуміння структури та навігації сайту, а також допомогла визначити ключові елементи інтерфейсу, які забезпечать користувачам приємний досвід. Це також вирішує можливі проблеми зручності використання на ранній стадії проєктування. Використання мокапів також дозволяє візуально оцінити і погодити концепцію перед тим, як розпочати розробку, що є ключовим для успішного реалізування проєкту. Завдяки цим мокапам, проєкт стає більш гнучким і адаптованим до змін, що можуть виникнути в процесі розробки.

Висновки до розділу 3

У третьому розділі представлено та обґрунтовано вибір технологій розробки, а саме:

- Php;
- Laravel;
- Blade;
- TailwindCss;
- MySQL.

Спроектовано та представлено модель та структуру бази даних.

Представлено мокап системи.

4 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Реалізація моделей та контролерів.

Під час розробки вебсайту реалізовано моделі для зв'язку бази даних та вебсайту.

Модель таблиці букетів представлена на рис. 4.1.

```
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Bouquet extends Model
{
    use HasFactory;

    protected $fillable = ['title', 'description', 'price', 'photo_url', 'size_id', 'type_id',
'color_id'];

    public function size()
    {
        return $this->belongsTo(Size::class);
    }

    public function type()
    {
        return $this->belongsTo(Type::class);
    }

    public function color()
    {
        return $this->belongsTo(Color::class);
    }

    public function reviews()
    {
        return $this->hasMany(Review::class);
    }

    public function orderItems()
    {
        return $this->hasMany(OrderItem::class);
    }
}
```

Рисунок 4.1 – Модель до таблиці букетів

Модель Bouquet у Laravel представляє букет квітів і має такі основні властивості та взаємозв'язки:

Основні атрибути

- title: назва букета;
- description: опис букета;
- price: ціна букета;
- photo_url: URL фотографії букета;
- size_id, type_id, color_id: зовнішні ключі для зв'язку з відповідними

моделями.

Взаємозв'язки

- size(): належить одному розміру (Size::class);
- type(): належить одному типу (Type::class);
- color(): належить одному кольору (Color::class);

- reviews(): має багато відгуків (Review::class);
- orderItems(): має багато елементів замовлення (OrderItem::class).

Ці взаємозв'язки дозволяють моделі Bouquet інтегруватися з іншими моделями в додатку, забезпечуючи зручне управління даними про букети.

Модель для керування таблицею з замовленнями представлено на рис. 4.2.

```
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Order extends Model
{
    use HasFactory;

    protected $fillable = ['user_id', 'date', 'delivery_id'];

    public function user()
    {
        return $this->belongsTo(User::class);
    }

    public function delivery()
    {
        return $this->belongsTo(Delivery::class);
    }

    public function orderItems()
    {
        return $this->hasMany(OrderItem::class);
    }
}
```

Рисунок 4.2 – Модель для керування таблицею замовлень

Модель Order у Laravel представляє замовлення і має такі основні атрибути та взаємозв'язки:

Основні атрибути

- user_id: ідентифікатор користувача;
- date: дата замовлення;
- delivery_id: ідентифікатор способу доставки.

Взаємозв'язки

- user(): належить одному користувачу (User::class);
- delivery(): належить одному способу доставки (Delivery::class);
- orderItems(): має багато елементів замовлення (OrderItem::class).

Ці взаємозв'язки дозволяють моделі Order інтегруватися з іншими моделями, забезпечуючи зручне управління даними про замовлення.

Модель для керування таблицею з користувачами представлено на рис. 4.3.

```
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class User extends Model
{
    use HasFactory;

    protected $fillable = ['first_name', 'last_name', 'email', 'phone', 'password',
'default_delivery_id'];

    public function reviews()
    {
        return $this->hasMany(Review::class);
    }

    public function orders()
    {
        return $this->hasMany(Order::class);
    }

    public function favorite()
    {
        return $this->hasOne(Favorite::class);
    }

    public function defaultDelivery()
    {
        return $this->belongsTo(Delivery::class, 'default_delivery_id');
    }
}
```

Рисунок 4.3 – Модель для керування таблицею з користувачами

Модель User у Laravel представляє користувача і має такі основні атрибути та взаємозв'язки:

Основні атрибути

- first_name: ім'я користувача;
- last_name: прізвище користувача;
- email: електронна пошта користувача;
- phone: телефон користувача;
- password: пароль користувача;
- default_delivery_id: ідентифікатор способу доставки за замовчуванням.

Взаємозв'язки

- reviews(): має багато відгуків (Review::class);
- orders(): має багато замовлень (Order::class);
- favorite(): має один улюблений об'єкт (Favorite::class);
- defaultDelivery(): належить одному способу доставки за замовчуванням (Delivery::class).

Ці взаємозв'язки дозволяють моделі User інтегруватися з іншими моделями, забезпечуючи зручне управління даними про користувачів.

Модель для керування таблицею з відгуками представлено на рис. 4.4.

```
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Review extends Model
{
    use HasFactory;

    protected $fillable = ['user_id', 'comment', 'bouquet_id'];

    public function user()
    {
        return $this->belongsTo(User::class);
    }

    public function bouquet()
    {
        return $this->belongsTo(Bouquet::class);
    }
}
```

Рисунок 4.4 – Модель для керування таблицею з відгуками

Модель Review у Laravel представляє відгук і має такі основні атрибути та взаємозв'язки:

Основні атрибути

- user_id: ідентифікатор користувача, який залишив відгук
- comment: текст відгуку
- bouquet_id: ідентифікатор букета, до якого відноситься відгук

Взаємозв'язки

- user(): належить одному користувачу (User::class)
- bouquet(): належить одному букету (Bouquet::class)

Ці взаємозв'язки дозволяють моделі Review інтегруватися з іншими моделями, забезпечуючи зручне управління даними про відгуки.

Далі розглянемо контролери для керування даними у вебзастосунку.

Контролер для керування даними з таблиці букетів представлено на рис. 4.5.

```
<?php
namespace App\Http\Controllers;

use App\Models\Bouquet;
use Illuminate\Http\Request;

class BouquetController extends Controller
{
    public function index()
    {
        $bouquets = Bouquet::all();
        return view('bouquets.index', compact('bouquets'));
    }

    public function create()
    {
        return view('bouquets.create');
    }

    public function store(Request $request)
    {
        $bouquet = Bouquet::create($request->all());
        return redirect()->route('bouquets.index')->with('success', 'Bouquet created successfully.');
```

Рисунок 4.5 – Контролер для керування даними у таблиці букетів

Контролер для керування даними у таблиці заказів представлено на
рис. 4.6.

```
<?php
namespace App\Http\Controllers;

use App\Models\Order;
use Illuminate\Http\Request;

class OrderController extends Controller
{
    public function index()
    {
        $orders = Order::all();
        return view('orders.index', compact('orders'));
    }

    public function create()
    {
        return view('orders.create');
    }

    public function store(Request $request)
    {
        $order = Order::create($request->all());
        return redirect()->route('orders.index')->with('success', 'Order created successfully.');
```

Рисунок 4.6 – Контролер для керування даними у таблиці заказів

Контролер для керування даними у таблиці користувачів представлено на рис. 4.7.

```
<?php
namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;

class UserController extends Controller
{
    public function index()
    {
        $users = User::all();
        return view('users.index', compact('users'));
    }

    public function create()
    {
        return view('users.create');
    }

    public function store(Request $request)
    {
        $user = User::create($request->all());
        return redirect()->route('users.index')->with('success', 'User created successfully.');
```

Рисунок 4.7 – Контролер для керування даними у таблиці користувачів

Контролер для керування даними у таблиці відгуків представлено на рис. 4.8.

```
<?php
namespace App\Http\Controllers;

use App\Models\Review;
use Illuminate\Http\Request;

class ReviewController extends Controller
{
    public function index()
    {
        $reviews = Review::all();
        return view('reviews.index', compact('reviews'));
    }

    public function create()
    {
        return view('reviews.create');
    }

    public function store(Request $request)
    {
        $review = Review::create($request->all());
        return redirect()->route('reviews.index')->with('success', 'Review created successfully.');
```

Рисунок 4.8 – Контролер для керування даними у таблиці відгуків

Дані моделі та контролери забезпечують цілісну передачу даних з бази до застосунку.

4.2 Реалізація фронтенду вебсайту

Під час проектування вебсайту доставляння букетів розроблено наступні мокапи сторінок:

- головної сторінки застосунку;
- сторінки каталогу букетів;
- сторінки про заклад;
- сторінки з поширеними питаннями;
- сторінки для певного букету;
- сторінки кошику;
- сторінки оформлення замовлення;
- сторінки улюблених букетів;
- сторінки контактів.

Головна сторінка вебсайту доставляння букетів має вишуканий та зрозумілий дизайн, орієнтований на зручність користувачів. В хедері сторінки розташоване меню навігації, яке включає розділи сайту, а також кнопки для входу в обліковий запис, переглядання кошику та пошук товарів. Головний банер сторінки містить назву сервісу "Bouquet Delivery" з слоганом, який складається з флористичних елементів.

Нижче, у розділі "Our Best Sellers", представлено слайдер з найпопулярнішими букетами, що дає змогу користувачам швидко ознайомитися з топовими продуктами. Розділ "About Us" включає короткий опис місії компанії, тобто легенду бренду, а також інфографіку з основними досягненнями, такими як кількість клієнтів та замовлень. Внизу сторінки знаходиться інформація про процес доставки, поділений на чотири кроки, які детально пояснюють, як відбувається обслуговування замовлень.

Розділ "Contacts" містить основні контактні дані для зв'язку з компанією, включаючи телефон та електронну пошту. В футері сторінки є відгуки задоволених клієнтів, що підкреслюють надійність та якість сервісу.

Дизайн сторінки розроблений у світлих та лагідних тонах з простими, але ефективними візуальними елементами, що робить навігацію інтуїтивно зрозумілою і приємною.

Реалізацію головної сторінки представлено на рис. 4.9.

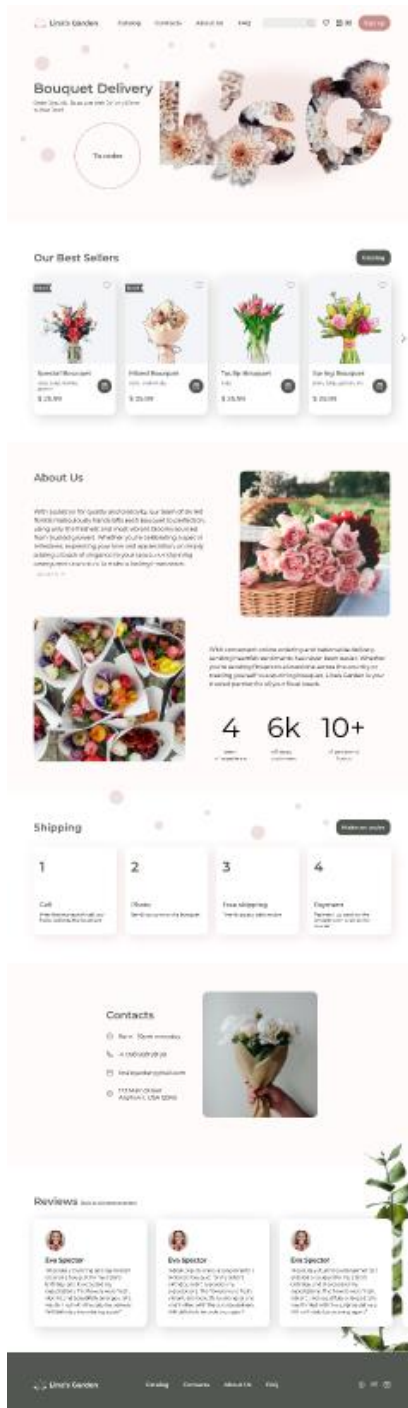


Рисунок 4.9 – Головна сторінка застосунку

Представлено сторінку каталогу вебсайту доставляння букетів. Розташовано список букетів, організований у вигляді сітки, де кожен продукт представлений зі

світлиною, назвою та ціною. Це дозволяє користувачам легко переглядати і вибирати букети.

Кожен букет у каталозі має кнопки для швидкого перегляду деталей та додавання товару до кошика, що робить процес покупки більш зручним і швидким. Внизу сторінки розміщено розділ з відгуками клієнтів, який включає фотографії та коментарі задоволених покупців, надаючи відвідувачам додаткову впевненість у якості продукції та сервісу.

Реалізацію сторінки каталогу представлено на рис. 4.10.

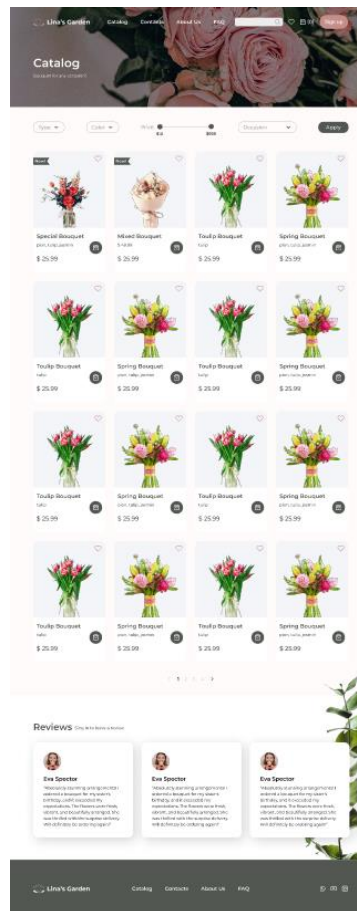


Рисунок 4.10 – Сторінка каталогу букетів

Ця сторінка вебсайту присвячена розділу "Про нас". Вона розроблена для детального представлення послуг, якості та особливостей компанії, створення емоційного зв'язку з потенційними клієнтами та підкреслення переваг, які вони отримують, обираючи цей бренд.

На сторінці використано яскраві та кольорові зображення квітів, які акцентують увагу на креативності та різноманітності асортименту. Текстовий

контент розділений на три основні блоки: "Crafting Memories with Every Petal", де йдеться про те, як кожна квітка в композиції створює незабутні спогади; "Personalized Service at Every Step", що наголошує на індивідуальному підході до кожного клієнта; та "Nationwide Floral Delivery Made Effortless", що описує легкість та зручність доставки квітів по всій країні.

В нижній частині сторінки розміщено інфографіку з ключовими показниками успішності бізнесу, такими як кількість замовлень та задоволених клієнтів, що підсилює довіру до бренду. Завершує сторінку гасло "Your Premier Destination for Effortless Floral Gifting", яке підкреслює позиціонування компанії як кращого вибору для подарунків, пов'язаних із квітами.

Реалізацію сторінки про заклад представлено на рис. 4.11.

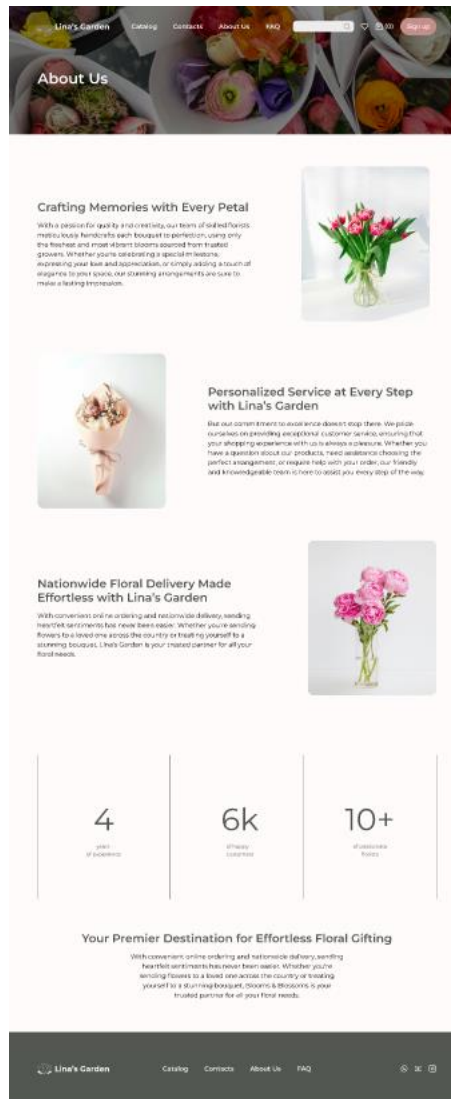


Рисунок 4.11 – Сторінка про заклад

Наступна сторінка вебсайту присвячена часто задаваним питанням (FAQs), де користувачі можуть знайти відповіді на звичайні питання щодо послуг магазину. Цей розділ структурований у формі розкритих списків, що дозволяє ефективно організувати інформацію і зробити її легкодоступною для користувачів. Відвідувачі сайту можуть клікнути на будь-яке питання, щоб розкрити детальну відповідь.

Зображена сторінка включає такі питання як "How do I cancel my order?", "How do I change the order?", "What methods of payment do you accept?" та "What information is necessary for delivery to a hotel?". Кожна відповідь налаштована на те, щоб допомогти користувачам розуміти політики і процедури магазину, тим самим підвищуючи користувацький досвід і сприяючи підвищенню довіри до бренду.

FAQ не тільки спрощує навігацію по сайту, але й допомагає зменшити кількість звернень до служби підтримки, надаючи користувачам відповіді на їхні запитання в режимі реального часу.

Реалізацію сторінки про поширені питання представлено на рис. 4.12.

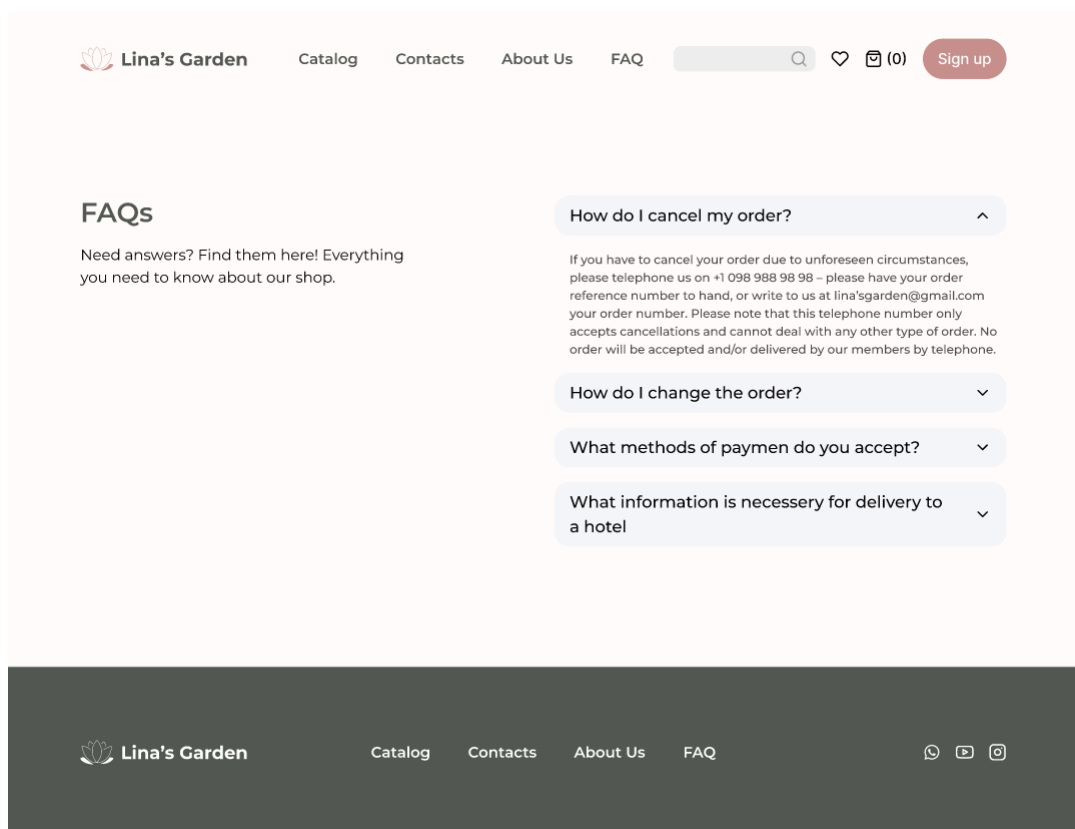


Рисунок 4.12 – Сторінка з поширеними питаннями

Наступна сторінка вебсайту призначена для детального перегляду та замовлення конкретного букету. Опис продукту розміщено поруч із зображенням, де надається короткий опис характеристик та унікальності букета, включно з інформацією про ціновий діапазон від \$15.99 до \$35.99.

Користувачі можуть обрати розмір букету (мінімум, стандарт, преміум) за вказаними цінами. Також є можливість вибрати дату доставки з зручного календаря, що дозволяє планувати покупку на конкретний день. Внизу сторінки розміщено розділ з відгуками клієнтів, де реальні користувачі діляться своїми враженнями від букетів, що додатково підсилює довіру до якості продукції.

Реалізацію сторінки певного букету представлено на рис. 4.13.

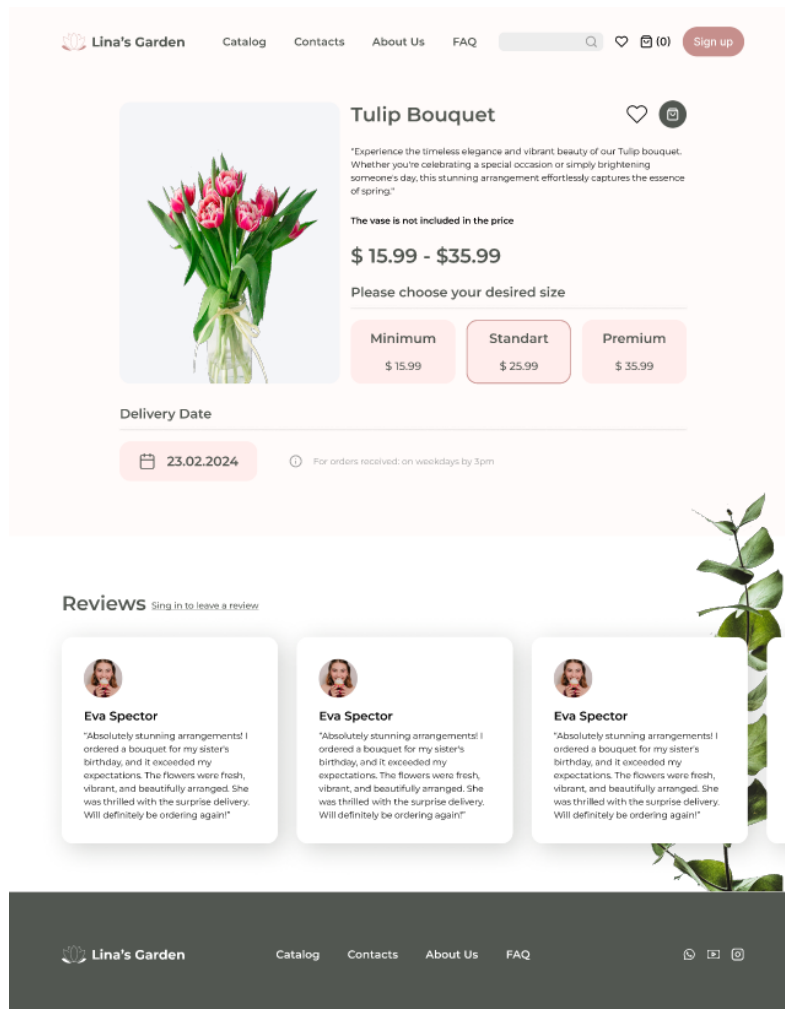


Рисунок 4.13 – Сторінка для певного букету

Наступна сторінка кошика призначена для перегляду та управління товарами, які користувач додає до свого кошика перед оформленням замовлення. На сторінці зображено детальний список вибраних товарів, кожен з яких має

2024 р. Гошева А. А. 121-КРБ.1 – 408.22010803

власне зображення, назву, кількість та ціну. Користувачі мають змогу змінювати кількість товарів або видаляти їх з кошика за допомогою інтерфейсу.

Під списком товарів відображено загальну суму замовлення, а також кнопка "Make an order", яка дозволяє перейти до процесу оформлення покупки. Нижче розташований розділ "You may also like", який пропонує додаткові товари, схожі на ті, що вже вибрані користувачем. Цей розділ може спонукати до додаткових покупок, показуючи релевантні та популярні товари, які можуть зацікавити покупця.

Реалізацію сторінки кошику представлено на рис. 4.14.

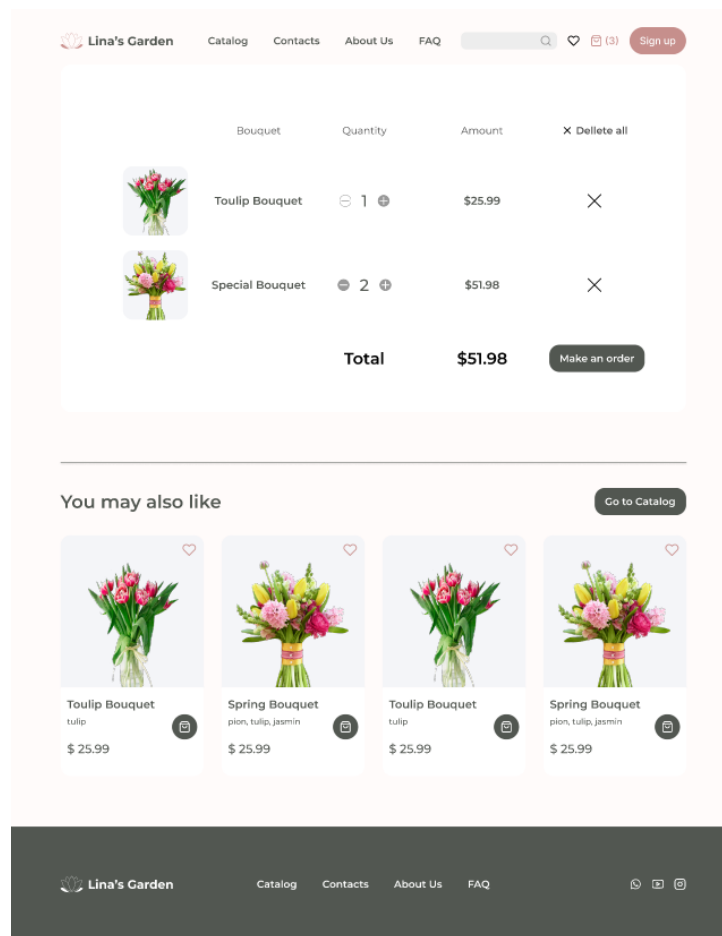


Рисунок 4.14 – Сторінка кошику

Наступна сторінка вебсайту призначена для оформлення замовлення і включає всі необхідні поля та опції, щоб спростити процес покупки для користувачів. Сторінка оформлення замовлення поділена на декілька секцій для зручності



- особисті дані (Personal Details): Користувачам пропонується ввести своє ім'я, місто та номер телефону. Є можливість відправити замовлення анонімно або обрати варіант отримання кур'єром;
- доставка (Shipping): Тут можна вибрати місто та вказати адресу доставки. Є опція вказати, що одержувачем є інша особа, та ввести її дані;
- оплата (Payment): Користувачі можуть вибрати між готівкою та картою, вказати деталі кредитної картки, якщо це необхідно;
- ваше замовлення (Your order): В цій секції відображаються обрані товари, їх кількість та загальна сума до оплати. Є можливість додати листівку до замовлення.

Під всіма полями розміщена кнопка "Checkout", що ініціює процес перевірки та завершення покупки. Сторінка оформлена у чистому та простому стилі, що робить процес замовлення інтуїтивно зрозумілим і мінімалістичним, забезпечуючи приємний досвід користувача.

Реалізацію сторінки про оформлення замовлення представлено на рис. 4.15.

The screenshot shows the checkout interface for 'Lina's Garden'. At the top, there is a navigation bar with links for 'Catalog', 'Contacts', 'About Us', and 'FAQ', along with a search bar, a heart icon, a shopping cart icon with '3' items, and a 'Sign up' button. The main content area is divided into four numbered sections:

- 1 Personal Details:** Includes input fields for 'Name', 'City', and 'Phone', a 'Send anonymously' toggle, and a 'City' dropdown.
- 2 Shipping:** Includes radio buttons for 'Pick up' and 'Courier', an 'Address' input field, and a toggle for 'Another person is the recipient' with 'Name' and 'Phone' input fields.
- 3 Payment:** Includes radio buttons for 'Cash' and 'Card', a 'When received' radio button, and input fields for 'Card number', 'Expiration date', and 'CVS/CVV'.
- 4 Your order:** A table showing the selected items:

Bouquet	Quantity	Amount
 Toulip Bouquet	1	\$25.99
 Special Bouquet	2	\$51.98
Total		\$51.98

Below the table is an 'Add a postcard' toggle and an 'Enter text...' input field. A large 'Checkout' button is positioned at the bottom right of this section.

At the bottom of the page, there is a dark footer bar with the 'Lina's Garden' logo, navigation links, and social media icons.

Рисунок 4.15 – Сторінка оформлення замовлення

Ця сторінка вебсайту представляє розділ "Favorites", де користувачі можуть переглядати свій список улюблених букетів. Сторінка виконана у вигляді галереї з чіткими та яскравими зображеннями букетів, кожен з яких має підпис, що включає назву та ціну. Відвідувачі мають можливість швидко перейти до покупки будь-якого букету зі списку, що робить вебсайт більш зручним для користувачів, які бажають зберегти або повторно замовити свої улюблені композиції.

У нижній частині сторінки розташований розділ "You may also like", який пропонує альтернативні варіанти букетів, які можуть зацікавити користувача. Це створює додаткову вартість для покупців, пропонуючи їм подібні товари на основі їхніх уподобань. Також присутня кнопка "Go to Catalog", яка спрощує перехід до повного каталогу букетів для детальнішого вибору.

Реалізацію сторінки про улюблені букети представлено на рис. 4.16.

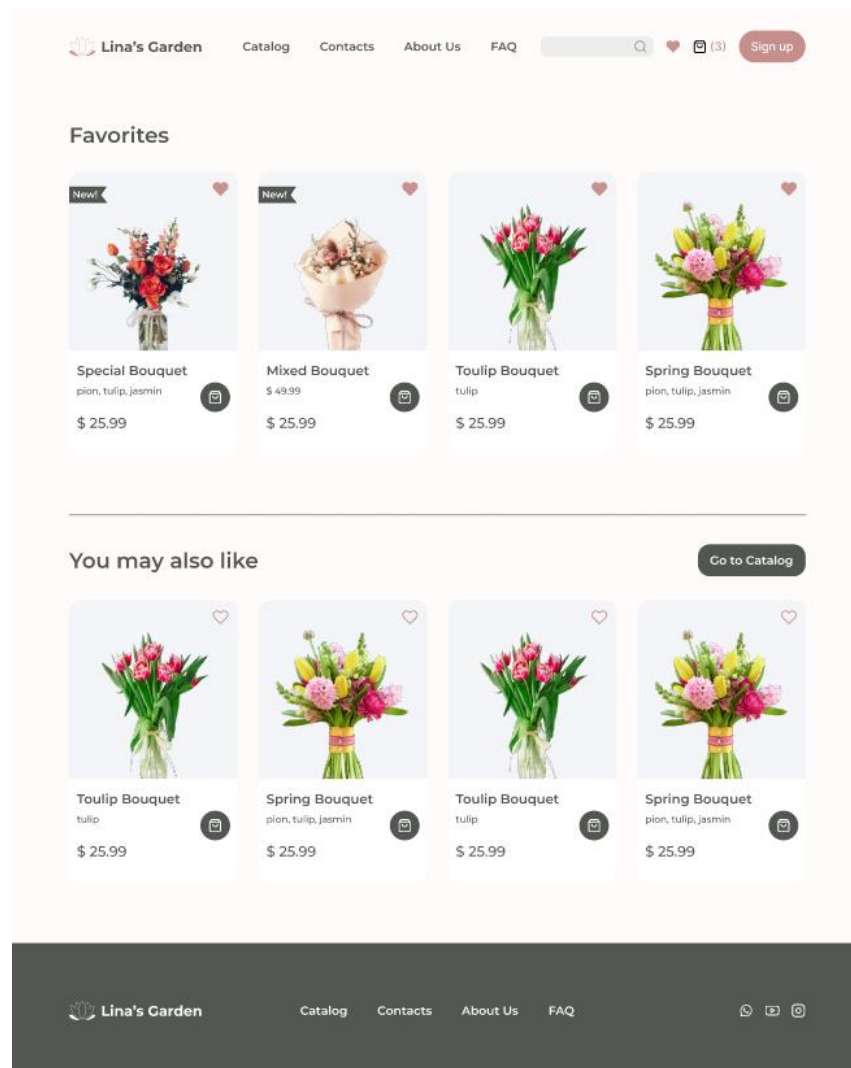


Рисунок 4.16 – Сторінка улюблених букетів

На зображенні представлено сторінку контактів, яка забезпечує користувачам всю необхідну інформацію для зв'язку з компанією. Сторінка має простий та елегантний дизайн, відповідає загальному стилю вебсайту.

У верхній частині сторінки розташоване головне зображення з квітами, що додає естетичного вигляду та створює приємне відчуття для відвідувачів. Нижче на сторінці вказані основні контактні дані:

- час роботи: зазначено, що компанія працює з 8 ранку до 10 вечора щоденно;
- телефонний номер: надається номер для звернення з питань та замовлень;
- електронна пошта: для електронних звернень;
- фізична адреса: адреса головного офісу компанії.

Реалізацію сторінки контактів представлено на рис. 4.17.

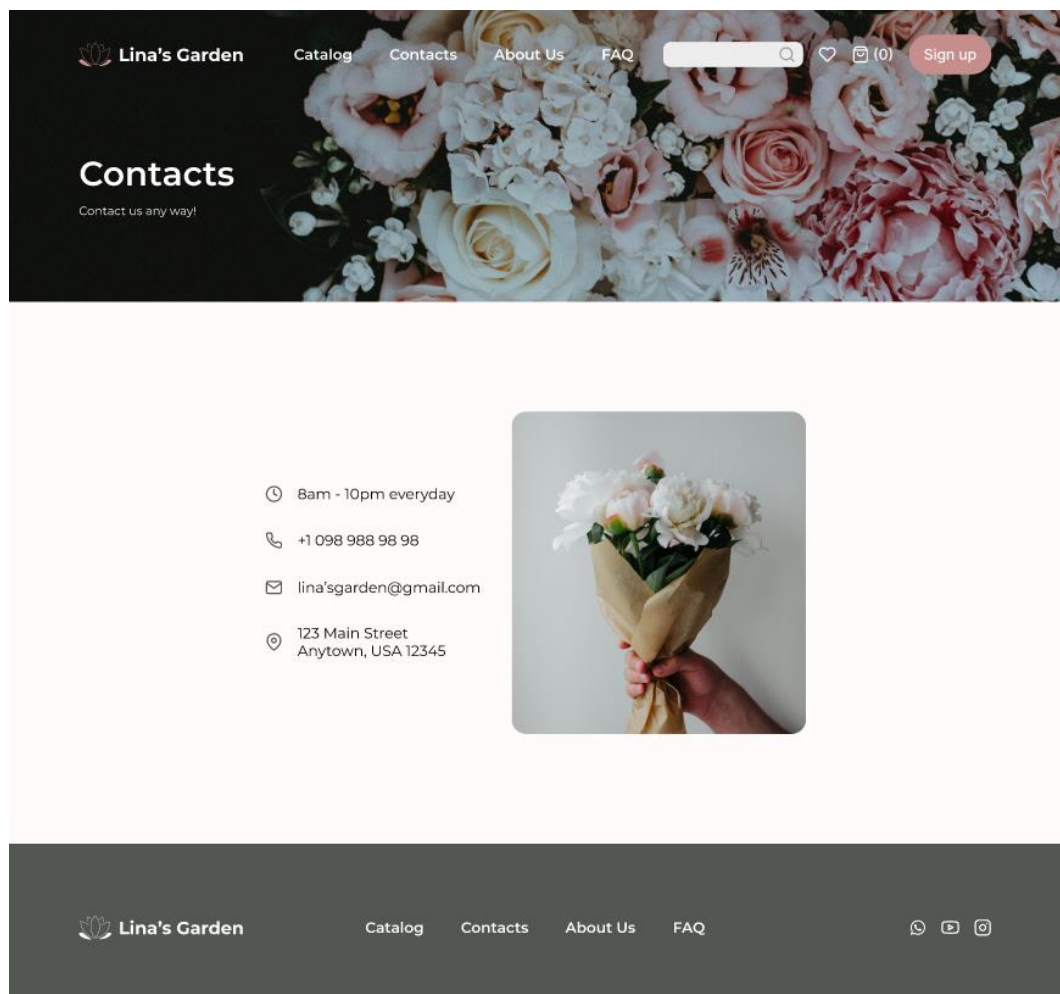


Рисунок 4.17 – Сторінка контактів

Ця сторінка реєстрації на вебсайті є ключовою для залучення нових користувачів і їхньої взаємодії з сайтом. Формуляр включає стандартні поля: ім'я, прізвище, електронна пошта та пароль, що необхідно для створення нового облікового запису.

Кнопка "Register" розташована під полями для введення, що дозволяє легко завершити процес реєстрації. Також на сторінці є посилання для користувачів, які вже мають обліковий запис, з пропозицією перейти до входу в систему.

Реалізацію сторінки реєстрації представлено на рис. 4.18.

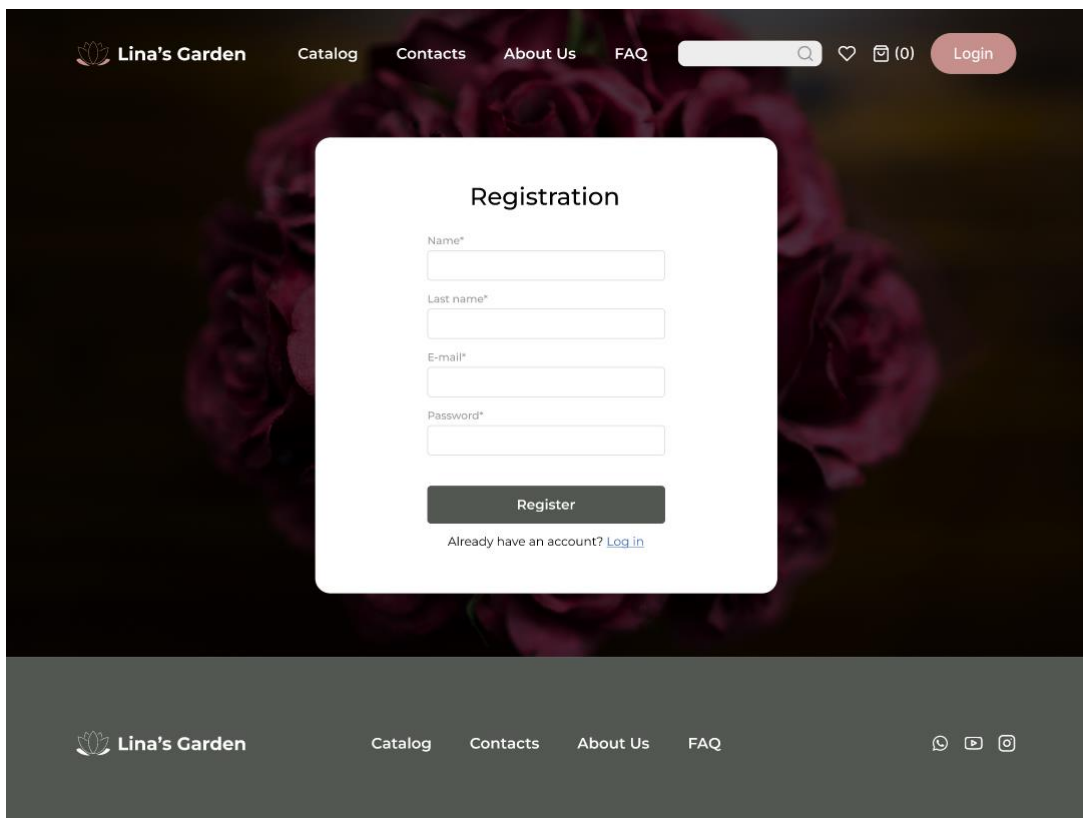


Рисунок 4.18 – Сторінка реєстрації

На зображенні представлена сторінка авторизації на вебсайті, що дозволяє користувачам увійти в свій обліковий запис для здійснення покупок та управління своїми замовленнями.

Форма входу включає поля для введення електронної пошти та пароля, а також кнопку "Login", що спрощує процес входу. Під формою входу є посилання "Register", яке надає новим користувачам можливість швидко перейти до сторінки реєстрації, якщо вони ще не створили обліковий запис.

Реалізацію сторінки авторизації представлено на рис. 4.19.

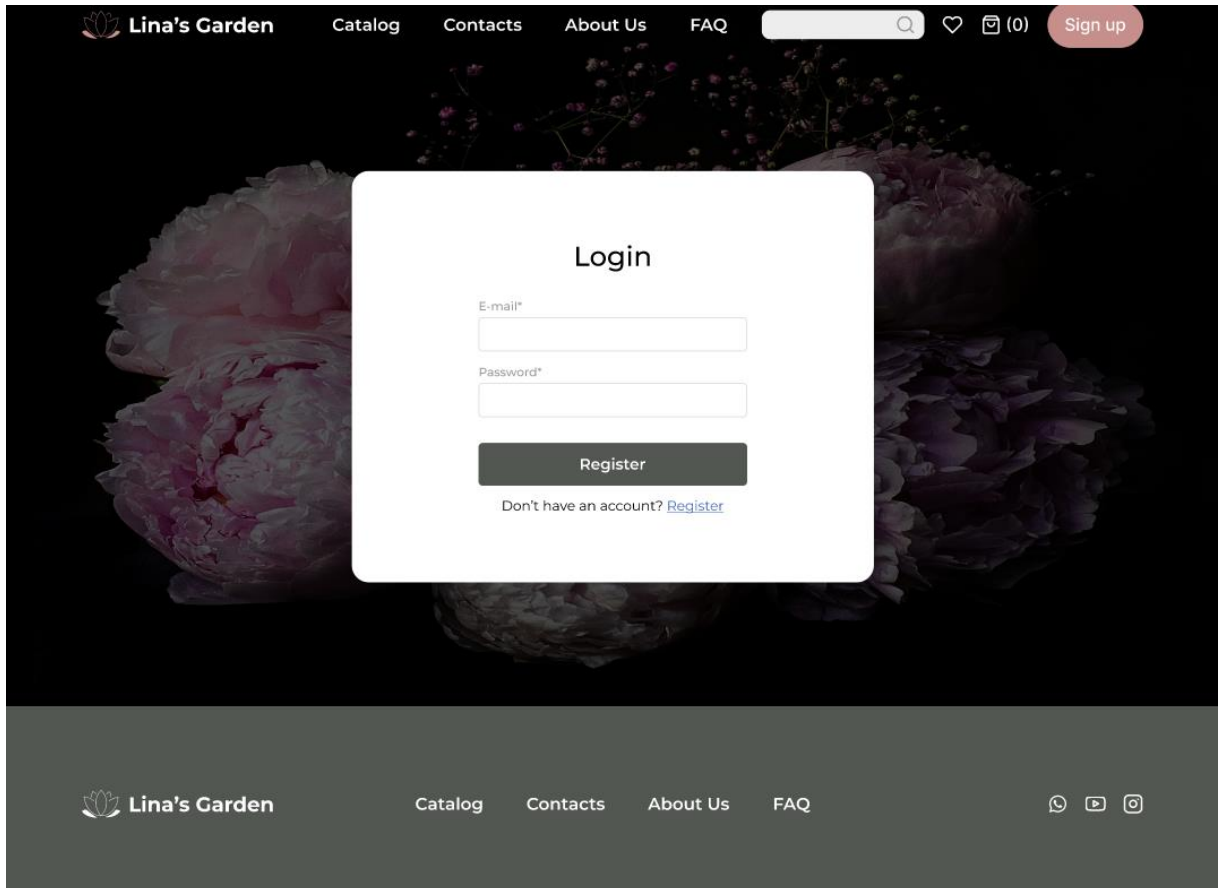


Рисунок 4.19 – Сторінка авторизації

На зображенні представлена сторінка особистого кабінету користувача на вебсайті. Ця сторінка надає зручний доступ до розділів, таких як історія замовлень, адреси, деталі облікового запису та вихід із системи. На цій сторінці користувач може переглядати деталі своїх замовлень, включаючи інформацію про товари, які були замовлені, їхні ціни, а також подробиці замовлення, такі як ім'я покупця, дата замовлення, одержувач, адреса доставки та контактна інформація.

Зображено два букети: "Tulip Bouquet" та "Spring Bouquet", кожен з яких має вказану ціну. Нижче надається інформація про конкретне замовлення, що включає дані про покупця, дату замовлення, одержувача та адресу доставки.

Реалізацію сторінки профілю представлено на рис. 4.20.

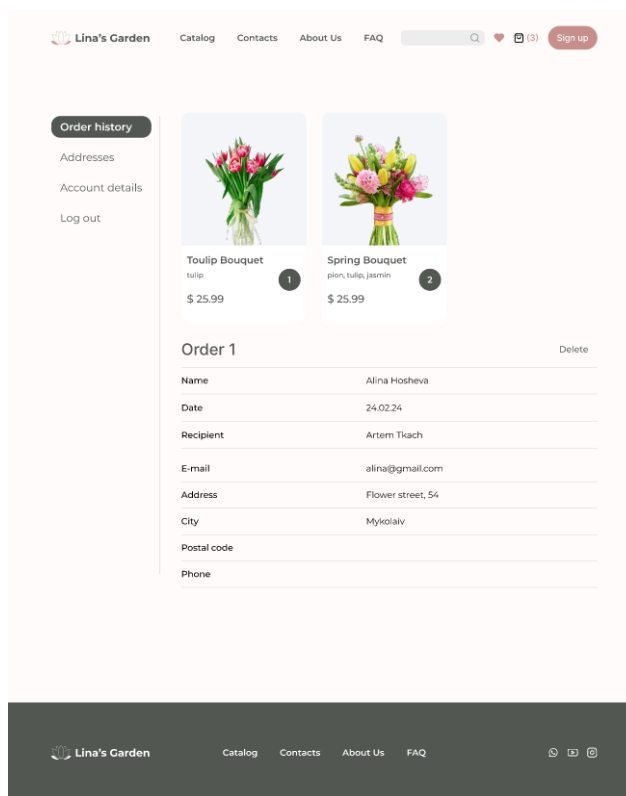


Рисунок 4.20 – Сторінка профілю

Реалізовано сторінки таким чином, щоб вони загалом були інтуїтивно зрозумілими і логічними для користувачів. Забезпечено чітку структуру та просту навігацію по вебсайту.

Висновки до розділу 4

У четвертому розділі кваліфікаційної роботи представлено реалізацію вебзастосунку для доставки букетів. Реалізовано наступні сторінки:

- головної сторінки застосунку;
- сторінки каталогу букетів;
- сторінки про заклад;
- сторінки з поширеними питаннями;
- сторінки для певного букету;
- сторінки кошику;
- сторінки оформлення замовлення;
- сторінки улюблених букетів;
- сторінки контактів.

ВИСНОВКИ

Під час аналізу предметної області та поставлення задачі було виявлено ключові потреби та вимоги до проєктування вебсайту, що включають визначення його основних функціональних можливостей та інтерфейсів взаємодії з користувачами.

Для досягнення визначеної мети вирішено наступні завдання:

- проаналізовано потреби ринку квіткового бізнесу в онлайн-середовищі;
- спроектовано та представлено модель та структуру бази даних;
- представлено мокап системи;
- реалізовано функціонал вебсайту.

Аналіз готових рішень дозволив оцінити існуючі продукти на ринку, їх переваги та недоліки. Це дало змогу визначити стратегічні напрями розробки власного продукту, що забезпечує унікальність та конкурентоспроможність.

Розробка діаграми варіантів використання допомогла структурувати можливі сценарії взаємодії користувачів з системою, визначити ключові функції та акторів системи.

Діаграму класів створено для визначення структури даних і взаємозв'язків між компонентами системи, що сприяє її модульності та масштабованості.

Розробка діаграми послідовностей дій системи дала змогу детально проробити логіку обробки запитів і реакції системи, забезпечивши надійність та ефективність реагування на дії користувача.

Загалом, аналіз і проєктування вебсайту демонструють системний підхід до створення рішення, що здатне задовольнити специфічні потреби користувачів та ефективно взаємодіяти в заданих контекстах. Ретельна підготовка у фазах аналізу та проєктування забезпечує міцну основу для подальшої реалізації та впровадження вебсайту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1) Flowers.ua. URL: <https://flowers.ua/> (дата звернення: 15.01.2024).
- 2) Buketland.ua. URL: <https://buketland.ua/> (дата звернення: 15.01.2024)
- 3) Floristik.ua. URL: <https://floristik.ua/city/kiev> (дата звернення: 15.01.2024).
- 4) Історія PHP. <https://sergeivl.ru/history-of-php.html> (дата звернення: 20.01.2024).
- 5) Інтеграція PHP з HTML та CSS. URL: <https://webcraftingcode.com/uk/pochatok-roboty/intehratsiia-php-z-html-ta-css-praktychnuu-posibnyk/> (дата звернення: 20.01.2024).
- 6) Advantages and disadvantages of PHP. URL: <https://anywhere.epam.com/en/blog/pros-and-cons-of-php> (дата звернення: 20.01.2024).
- 7) Фреймворки у мовах програмування. URL: https://cloud.itstep.org/blog_3/frameworks-in-programming-languages-what-are-they-for-and-how-to-choose-them#16 (дата звернення: 20.01.2024).
- 8) Why Laravel Framework is the Best Choice for PHP Web Development. URL: <https://www.netsolutions.com/insights/laravel-framework-benefits/> (дата звернення: 20.01.2024).
- 9) Laravel Blade Basics. URL: <https://www.inmotionhosting.com/support/edu/laravel/laravel-blade-basics/> (дата звернення: 20.01.2024).
- 10) Introduction to Tailwind CSS. URL: <https://www.geeksforgeeks.org/introduction-to-tailwind-css/> (дата звернення: 20.01.2024).
- 11) Іванов І. І. Сучасні технології в електронній комерції. Київ: Логос, 2019. 234 с.
- 12) Петренко П. П., Сидоренко С. С. Веб-дизайн: від ідеї до реалізації. Львів: Афіша, 2020. 192 с.

- 13) Технології та інновації в квітковому бізнесі: матеріали міжнародної конференції. Одеса: ОНУ ім. Мечникова, 2019, 156 с.
- 14) Морозова О. В. Основи програмування веб-додатків. Харків: Фактор, 2018, 310 с.
- 15) Кузнєцова Т. О. Інновації в онлайн-торгівлі: глобальні тенденції. Дніпро: Баланс Бізнес Букс, 2022, 278 с.
- 16) Шевченко Г. Г., Бондаренко В. В. Цифровий маркетинг: стратегії та інструменти. Київ: КНЕУ, 2021. 196 с.
- 17) Лисенко Л. Л. UX/UI дизайн: створення ефективних вебсайтів. Київ: Університет "Україна", 2021, 212 с.
- 18) Федоренко В. Г. Розробка веб-сайтів: від прототипу до готового продукту. Київ: Наукова думка, 2019, 288 с.
- 19) Інтерфейси користувача і їх вплив на продажі в інтернет-магазинах / за ред. Краснова Є. В. Суми: Університетська книга, 2020, 178 с.