

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА
ФРАНКА

ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ

Кафедра цифрової економіки та бізнес-аналітики

КУРСОВА РОБОТА

зі спеціалізації

на тему:

«Розробка інтернет-магазину студії флористики»

спеціальність: 051 «Економіка»

(код та найменування спеціальності)

спеціалізація: «Інформаційні технології в бізнесі»

(найменування спеціалізації)

освітній ступінь: бакалавр

(бакалавр/магістр)

Науковий керівник:

К. е. н., доц. Стадник Ю. А.

(науковий ступінь, посада, прізвище, ініціали)

“ ” 2021 р.
(підпис)

Виконавець:

студентка групи УФЕ-41с

Квятковська А. В

(прізвище, ініціали)

“ ” 2021 р.
(підпис)

Загальна кількість балів

(підпис, ППП членів комісії)

ЛЬВІВ 2021

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ОПИС ВИМОГ ДО ВЕБ-АПЛІКАЦІЇ	5
1.1 Техніко-економічне обґрунтування.....	5
1.2 Перелік функцій веб-сайту	9
1.3 Інтерфейс веб-сайту	10
РОЗДІЛ 2. РОЗРОБКА ВЕБ-АПЛІКАЦІЇ	12
2.1 Структура веб-сайту.....	12
2.1.1 Програмування клієнтської частини	13
2.1.2 Програмування серверної частини	16
2.2 Розробка бази даних.....	21
2.2.1 Опис моделі даних.....	22
2.2.2 Запити до таблиць бази даних	23
2.3 Тестування веб-сайту	25
ВИСНОВКИ	28
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	29
ДОДАТКИ	30

ВСТУП

Актуальність теми курсової роботи. На сьогоднішній день інтернет-магазини користуються популярністю, як і у звичайних користувачів, так і у фірм, що бажають купити або ж продати свій товар. Завдання звичайного користувача полягає в перегляді фотографій і характеристик товару, порівнянні цін і можливу покупку. З іншої сторони, персональний сайт або інтернет-магазин – ефективний та актуальний спосіб знайти нових та/або постійних клієнтів для підприємства у будь-якій сфері.

Метою даної роботи є поглиблення знань та навичок у веб-програмуванні та тестуванні сайту, як наслідок – реалізація інтернет-магазину студії флористики.

Виходячи з поставленої мети, сформовано наступні завдання:

- Визначення функціоналу сайту;
- Розробка інтерфейсу;
- Проектування бази даних та складання запитів;
- Забезпечення взаємодії веб-сторінок зі створеною базою даних, за допомогою форм та запитів.

Об'єктом даної роботи є використання засобів розробки PHP, MySQL, HTML, CSS, JavaScript для створення програмних аплікацій.

Предмет даної роботи – процес створення веб-аплікації.

Практичне значення курсової роботи. Створений, у результаті виконання курсової роботи, веб-сайт може бути розміщений на хостингу та використовуватися, як інтернет-магазин квітів та супутніх товарів і послуг.

Використане програмне забезпечення. Локальний сервер Denwer, програмне середовище phpMyAdmin, спеціалізований текстовий редактор Notepad++, мови веб-розробки PHP, HTML, CSS, JavaScript, система керування

реляційними базами даних MySQL, мова програмування для взаємодії з базами даних SQL.

Структура роботи. Курсова робота складається зі вступу, двох розділів, поділених, у свою чергу, на підрозділи, висновку, списку використаної літератури та додатків.

РОЗДІЛ 1. ОПИС ВИМОГ ДО ВЕБ-АПЛІКАЦІЇ

1.1 Техніко-економічне обґрунтування

Флористичний бізнес - справа перспективна, і залишається такою, навіть незважаючи на високу конкуренцію на ринку. Учасники ринку не задовольняють весь попит на квіти, що відкриває можливості для формування нових компаній в цій сфері. Також з'явилася тенденція продавати в квіткових магазинах не тільки рослини. Звичайно, можна зосередитися тільки на квітах, але краще надавати і додаткові послуги:

- продаж супутніх товарів: подарунків, листівок, шоколаду, цукерок і т.д.;
- співпраця з весільними організаціями для оформлення заходів квітами;
- кур'єрська доставка.

Очевидно, що це дозволить збільшити виручку і підвищити зручність для покупців. Магазин швидше стане відомим, якщо надавати оригінальні послуги - наприклад, підбирати букети на основі фотографії адресата, співпрацювати з місцевими майстрами для продажу унікальних листівок, солодоців та ін.

Проте, в сучасних умовах, коли практично все відбувається в Інтернеті, ніщо не приваблює клієнтів більше, ніж можливість придбати товар чи послугу онлайн.

Отже, практичним завданням даної роботи є створення програмного продукту у вигляді інтернет-магазину. За допомогою нього потенційні клієнти можуть ознайомитися з інформацією про студію флористики, послуги, які надаються нею, здійснити покупку, а також стати постійним клієнтом, зареєструвавшись на сайті.

У процесі розробки веб-сайту використано наступні технології:

- PHP (англ. PHP: Hypertext Preprocessor — PHP: гіпертекстовий препроцесор; попередня назва: Personal Home Page Tools) — скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP — проєкт відкритого програмного забезпечення. PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, тому що браузер отримує готовий html-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок.

- HTML (англ. HyperText Markup Language — мова розмітки гіпертексту) — це мова тегів, засобами якої здійснюється розмічання веб-сторінок для мережі Інтернет. Браузери отримують HTML-документи з веб-сервера або з локальної пам'яті й передають документи в мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично і спочатку включені сигнали для зовнішнього вигляду документа. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовану сторінку. HTML надає засоби для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML окреслені тегами, написаними з використанням кутових дужок, вони і виводять вміст на сторінку. Браузери не показують теги HTML, але використовують їх для інтерпретації вмісту сторінки. HTML може вбудовувати програми, написані на мові сценаріїв, наприклад JavaScript, що впливає на поведінку та вміст веб-сторінок. Включення CSS визначає вигляд і компонування вмісту. World Wide Web Consortium (W3C), який супроводжує стандарти HTML та CSS, заохочує використання CSS над явним презентаційним HTML з 1997 року.

- CSS (англ. Cascading Style Sheets, укр. Каскадні таблиці стилів) — це спеціальна мова стилю сторінок, що використовується для опису їхнього

зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних. CSS є основною технологією всесвітньої павутини, поряд із HTML та JavaScript. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів. CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо. CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

- JavaScript (JS) — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої) скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

- SQL (англ. Structured query language — мова структурованих запитів) — декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних. Сама по собі SQL не є ані системою керування базами даних, ані окремим програмним продуктом. На

відміну від дійсних мов програмування (C або Pascal), SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати як інструкції для керування даними. Окрім цього, стандарт SQL містить функції для визначення зміни, перевірки та захисту даних. SQL — це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також керування базами даних. Багато баз даних підтримує SQL з розширеннями до стандартної мови. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення і видалення даних за допомогою використання системи керування і адміністративних функцій.

- phpMyAdmin — веб-додаток з відкритим кодом на мові PHP із графічним веб-інтерфейсом для адміністрування бази даних MySQL або MariaDB. phpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати запити SQL, переглядати та редагувати вміст таблиць баз даних. Ця програма користується великою популярністю у веб-розробників, оскільки дозволяє керувати базу даних MySQL без вводу SQL команд через дружній інтерфейс і з будь-якого комп'ютера під'єданого до мережі Інтернет без необхідності встановлення додаткового програмного забезпечення. На сьогоднішній день phpMyAdmin широко застосовується на практиці. Останнє пов'язано з тим, що розробники інтенсивно розвивають свій продукт, з огляду на всі нововведення СКБД MySQL. Переважна більшість українських провайдерів використовують цей застосунок як панель керування для того, щоб надати своїм клієнтам можливість адміністрування виділених їм баз даних.

- Denwer — програмна оболонка, що використовується веб-розробниками для розробки сайтів на «домашній» (локальній) windows-машині без необхідності підключення до мережі Інтернет.

1.2 Перелік функцій веб-сайту

Програмний продукт повинен забезпечувати основні функції інтернет-магазину. Веб-сайт дозволяє ознайомитися з брендом надання послуг, зробити покупку, використовуючи стандартні форми та надає можливість здійснення зворотного зв'язку. Також повинна існувати можливість додавання бажаних товарів корзину, накопичення вибраних товарів та їх видалення з корзини. До того ж, на сайті повинна бути можливість зареєструватися та/або авторизуватися, щоб зберігати та використовувати свої дані, в особистому кабінеті, для полегшеного здійснення наступних покупок.

Варіанти використання сайту ілюструються за допомогою однойменної діаграми, зображеної на рисунку 1.1. Діаграма варіантів використання складається з акторів «Гість», «Зареєстрований користувач» та доступних їм варіантів використання сайту, які зображені овалом.

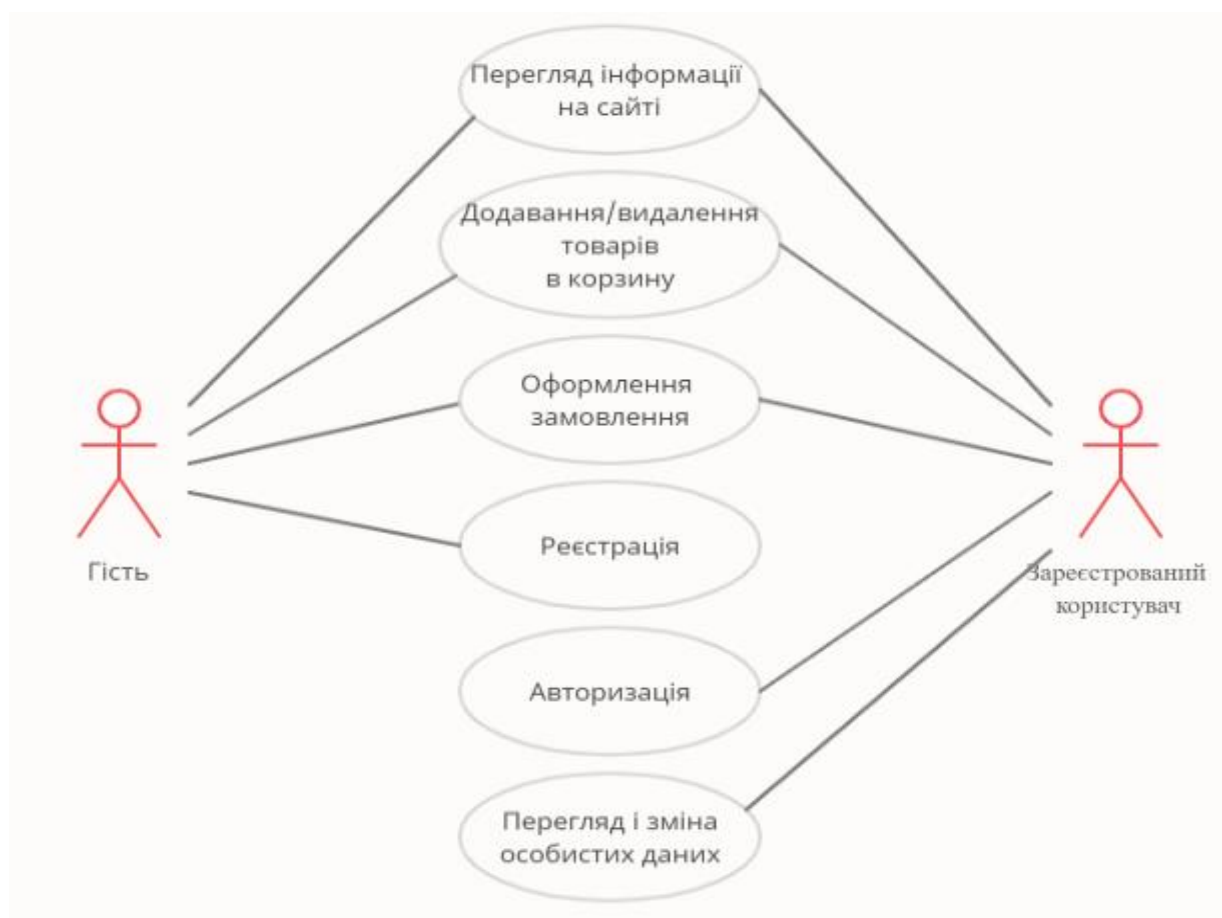


Рис. 1.1. Діаграма варіантів використання сайту

База даних, підключена до веб-сайту, повинна містити в собі таблиці з інформацією про всі товари і послуги, які надає студія флористики. Окрім того, інформація про зареєстрованих на сайті користувачів також має вноситися та зберігатися в окремій таблиці.

1.3 Інтерфейс веб-сайту

Одним із перспективних напрямів створення інтерфейсів користувача вважаються адаптивні інтерфейси. Адаптивним інтерфейсом користувача (AI) називають сукупність програмних та технічних засобів, які дозволяють користувачу ефективно використовувати всі можливості, які надає програма або сайт, та задаються конкретні налагодження для кожного користувача. Одним з головних критеріїв ефективності функціонування програмних продуктів є максимальне використання для роботи з ними людських ресурсів.

Адаптованість означає, що інтерфейс повинен бути:

- сумісним з потребами та можливостями користувача;
- забезпечувати простоту переходу від виконання однієї функції до іншої;
- забезпечувати користувача на високому рівні вказівками стосовно його можливих дій, а також генерувати належний зворотний зв'язок на його запити;
- забезпечувати користувача різними, взаємодоповнюючими формами представлення результатів в залежності від типу запиту;
- враховувати особливості користувачів різних рівнів;

Достатність інтерфейсу означає наступне:

- допустимі запити користувача повинні бути чіткими і однозначними для користувачів всіх рівнів;

– реакція системи на всі типи запитів також повинна бути однозначною і зрозумілою і, по можливості, простою.

Дружність інтерфейсу - це максимальна простота його використання і готовність в повній мірі задовольнити запити користувача при вирішенні певної задачі.

Гнучкість інтерфейсу – це можливість його адаптування до розв’язання конкретної задачі. Якщо розв’язувана задача дуже складна, то інтерфейс повинен полегшувати формулювання запитів і видавати результати у формі, яка легко і швидко сприймається користувачем.

При цьому простота означає, що:

– інтерфейс не повинен бути перевантажений деталями щодо представлення розв’язку поставленої задачі – користувач може не охопити всіх подробиць – тобто нічого зайвого, крім того, що необхідно для розуміння результату;

– він не повинен містити зайвих декоративних деталей, які відволікають від головної задачі;

– інтерфейс повинен ґрунтуватись на використанні відомих, загальноприйнятих методів і засобів представлення інформації;

– в ідеалі процес взаємодії користувача з системою не повинен представляти ніяких труднощів.

РОЗДІЛ 2. РОЗРОБКА ВЕБ-АПЛІКАЦІЇ

2.1 Структура веб-сайту

Дана програма реалізована у вигляді веб-сайту, який працює через будь-який доступний браузер. Доступ до сайту відбувається через адресу: «<http://www.flora/>». Кожен відвідувач має можливість оглянути сайт, здійснити бажану покупку та зареєструватися. Зареєстровані у системі користувачі можуть увійти в особистий кабінет, щоб скористатися своїми збереженими даними для замовлення.

Створений веб-сайт складається з наступних сторінок:

1. Web-сторінка «<http://www.flora/>» – головна сторінка, де можна ознайомитись із інформацією про студію флористики та скористатись навігацією сайту. Дана сторінка, в свою чергу, поділена на 5 секцій:
 - a. «<http://www.flora/#section-1>» - головна;
 - b. «<http://www.flora/#section-2>» - перелік послуг, які надає студія та посилання на сторінки з детальнішою інформацією;
 - c. «<http://www.flora/#section-3>» - умови оплати та доставки;
 - d. «<http://www.flora/#section-4>» - карта з фізичними магазинами;
 - e. «<http://www.flora/#section-5>» - контактна інформація.
2. Web-сторінка «<http://www.flora/flowers.php>» - сторінка з переліком букетів, які можна придбати.
3. Web-сторінка «<http://www.flora/flowerpots.php>» - сторінка з переліком кімнатних рослин, які можна придбати.
4. Web-сторінка «<http://www.flora/decor.php>» - сторінка, на якій можна ознайомитися з послугами декорування свят.
5. Web-сторінка «<http://www.flora/courses.php>» - сторінка з переліком та описом курсів флористики.
6. Web-сторінка «<http://www.flora/cart.php>» - корзина – сторінка, де можна переглянути обрані, для покупки, товари.

7. Web-сторінка «<http://www.flora/account.php>» - форма авторизації/особистий кабінет.

8. Web-сторінка «<http://www.flora/registration.php>» - форма реєстрації на сайті.

2.1.1 Програмування клієнтської частини

Для розробки клієнтської частини сайту (frontend), використано стандарти розмітки сторінок HTML, стилі CSS та JavaScript. Для оформлення сайту було створено блоки header.php, footer.php та документ style.css зі стилями.

Клієнтська (гостьова) частина веб-сайту, має такі основні сторінки (зовнішній вигляд подано у додатках):

- Головна сторінка – поділена на секції, які містять інформацію про студію флористики, послуги, які надаються, умови оплати і доставки та контактну інформацію. (Додаток А-Є)
- Квіти – сторінка з переліком квітів, які можна придбати. (Додаток Ж)
- Кімнатні рослини – сторінка з переліком кімнатних рослин, які можна придбати. (Додаток З)
- Декор – сторінка з переліком та описом послуг декорування свят, які надає студія. (Додаток И)
- Курси флористики – сторінка з переліком та описом курсів флористики, які проводить студія. (Додаток К)
- Корзина – сторінка з переліком товарів, які обрав клієнт сайту. (Додаток Л)
- Оформлення замовлення – сторінка з формою, при заповненні якої клієнт оформлює своє замовлення. (Додаток М)

- Особистий кабінет – сторінка з формою авторизації для зареєстрованих на сайті користувачів/після авторизації відображається особистий кабінет того чи іншого користувача. (Додаток Н)
- Реєстрація – сторінка з формою реєстрації. (Додаток О)

Зареєстрований користувач має доступ до тих же сторінок, що і в гостьовій частині сайту, а також, після авторизації, може переглядати та змінювати свої дані в особистому кабінеті (рис. 2.1).

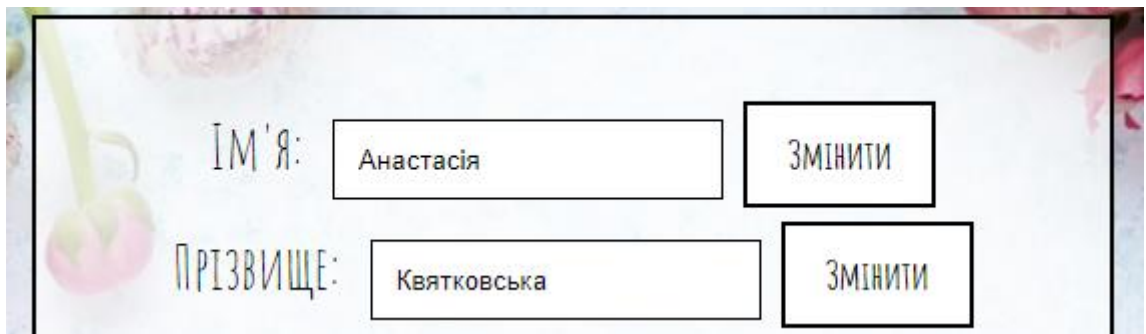
A screenshot of a user profile form. It features two input fields: 'ІМ'Я:' (Name) containing 'Анастасія' and 'ПРИЗВИЩЕ:' (Surname) containing 'Квятковська'. To the right of each field is a button labeled 'ЗМІНИТИ' (Change). The background is a light blue floral pattern.

Рис. 2.1 Дані користувача

На усіх сторінках сайту є блок header та на сторінках «Головна» та «Оформлення замовлення» – footer. Блок header (рис. 2.2) містить посилання на головну сторінку, меню, посилання на особистий кабінет та корзину. У блоці footer (рис. 2.3) розміщена форма зворотного зв'язку, де клієнт може залишити свої відгуки та побажання.

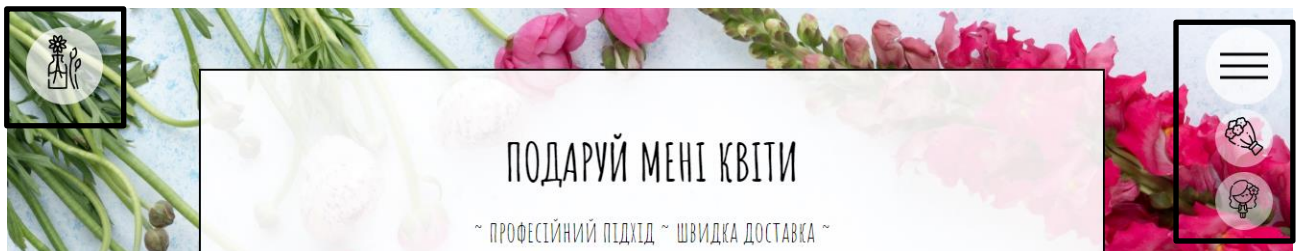
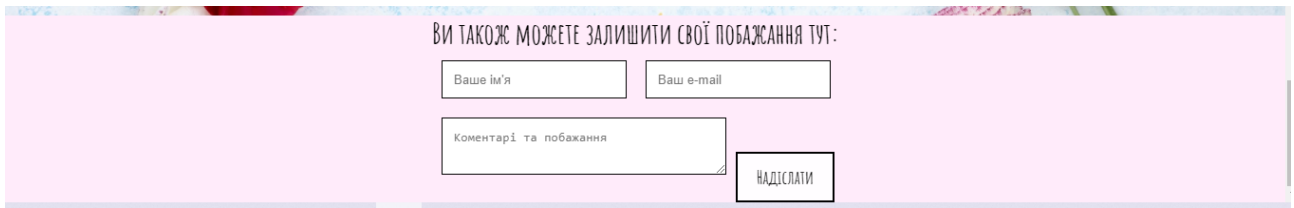


Рис. 2.2 Блок Header



ВІ ТАКОЖ МОЖЕТЕ ЗАЛИШИТИ СВОЇ ПОБАЖАННЯ ТУТ:

Ваше ім'я

Ваш e-mail

Коментарі та побажання

НАДІСЛАТИ

Рис. 2.3 Блок Footer

Задля реалізації динамічного меню сайту застосовано наступний скрипт:

```
function burgerMenu(selector) {

    let menu = $(selector);

    let button = menu.find('.burger-menu_button', '.burger-menu_lines');

    let links = menu.find('.burger-menu_link');

    let overlay = menu.find('.burger-menu_overlay');

    button.on('click', (e) => {

        e.preventDefault();

        toggleMenu();

    });

    links.on('click', () => toggleMenu());

    overlay.on('click', () => toggleMenu());

    function toggleMenu(){
```

```

        menu.toggleClass('burger-menu_active');

        if (menu.hasClass('burger-menu_active')) {

            $('body').css('overflow', 'hidden');

            } else {

            $('body').css('overflow', 'visible');

            }

        }

    }

}

burgerMenu('.burger-menu');

```

2.1.2 Програмування серверної частини

Код веб-сайту написаний в спеціалізованому текстовому редакторі Notepad++. Серверна частина сайту написана скриптовою мовою програмування PHP, а зовнішній вигляд сайту - мовами HTML, CSS, JavaScript.

Для забезпечення повноцінної роботи сайту створено наступні серверні скрипти:

1. для створення зв'язку між сайтом та базою даних

```

<?php

$servername = "localhost";

$username = "root";

$password = "";

```



```

$dbname = "floradb";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect_error) {

    die("Connection failed: " . $conn->connect_error);    }

?>

```

2. для обробки запитів до бази даних

```

<?php require_once "dbconnect.php";

$sql = "SELECT * FROM flowers";

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

if ($result->num_rows > 0) {

    echo "";

    while($row = $result->fetch_array()) {

        echo '<div class="product-wrap">

        <div class="product-item">

        <div class="product-buttons">

            <a          href="cart.php?fid=' . $row['flower_id'].'"
target="_blank" class="button">Купити</a>

        </div>

```

```

        </div>

        <div class="product-title">

        <a href="">'. $row['flower_name'].'</a>

        <span      class="product-price">'.      $row['flower_price'].'
UAN</span>

        </div>

    </div>'; }

```

```

}

```

```

?>

```

3. для відправлення заявки зворотного зв'язку

```

<?php

    header('Location: /footer.php');

    $name = $_POST['name'];

    $email = $_POST['email'];

    $message = $_POST['message'];

    $headers      =      "Form:      $email\r\nReply-To:      $email\r\nContent-
type:text/html;charset=utf-8\r\n";

    mail("th.gns61@gmail.com", $name, $message, $headers)

?>

```

4. для обробки форм реєстрації, авторизації та виходу

- а. код реєстрації

```

<?phprequire_once "dbconnect.php";

```

```

$firstname = mysqli_real_escape_string($conn, $_POST['firstname']);

$lastname = mysqli_real_escape_string($conn, $_POST['lastname']);

$phone = mysqli_real_escape_string($conn, $_POST['phone']);

$email = mysqli_real_escape_string($conn, $_POST['email']);

$city = mysqli_real_escape_string($conn, $_POST['city']);

$addres = mysqli_real_escape_string($conn, $_POST['addres']);

$password = mysqli_real_escape_string($conn, $_POST['password']);

$sql = "INSERT INTO users (user_firstname, user_lastname,
user_phone, user_email, user_city, user_addres, user_password) VALUES
('$firstname', '$lastname', '$phone', '$email', '$city', '$addres', '$password')";

if (mysqli_query($conn, $sql)) {

    echo '<div class="inputtext"><h3>Вхід:</h3><br>

    <p><a href="account.php">Увійдіть</a> в особистий
кабінет</p>';

    } else {

    echo "Error: " . $sql . "<br>" . mysqli_error($conn);    }

mysqli_close($conn); ?>

b. код авторизації

<?php

if(count($_POST)>0) {

    $conn = mysqli_connect("localhost", "root", "", "floradb");

```

```

$result = mysqli_query($conn, "SELECT user_email,
user_password FROM users WHERE user_email = " . $_POST["user_email"] . " and
user_password = " . $_POST["user_password"] . "");

```

```

$count = mysqli_num_rows($result);

```

```

if($count==0) {

```

```

    header ("Location: http://www.flora/");

```

```

} else {

```

```

    $_SESSION['user_email'] = $user_email;

```

```

    $_SESSION['success'] = "You are now logged in";

```

```

    header ("Location: http://www.flora/account.php?uid=" .
$_POST["user_email"] . ""); } } ?>

```

с. КОД ВИХОДУ

```

<?php

```

```

    session_unset();

```

```

    session_destroy();

```

```

?>

```

```

<div class="inputtext"><h3>Вхід:</h3><br>

```

```

<?php

```

```

    echo 'Ви вийшли з кабінету. <a
href="http://www.flora/account.php">Повернутись.</a>';

```

```

?>

```

```

</div>

```

2.2 Розробка бази даних

В програмному середовищі phpMyAdmin створено базу даних «floradb», яка міститиме таблиці з даними інтернет-магазину. При створенні бази даних необхідно обрати кодування, яке дозволяє зберігати в таблицях даних символи кирилиці - utf8_general_ci. У процесі створення бази даних її заповнено таблицями та полями, з інформацією необхідною для сторінок веб-сайту. Між таблицями встановлено зв'язки, визначено тип даних усіх полів, визначено первинні ключі – усе це зображено на рисунку 2.4.

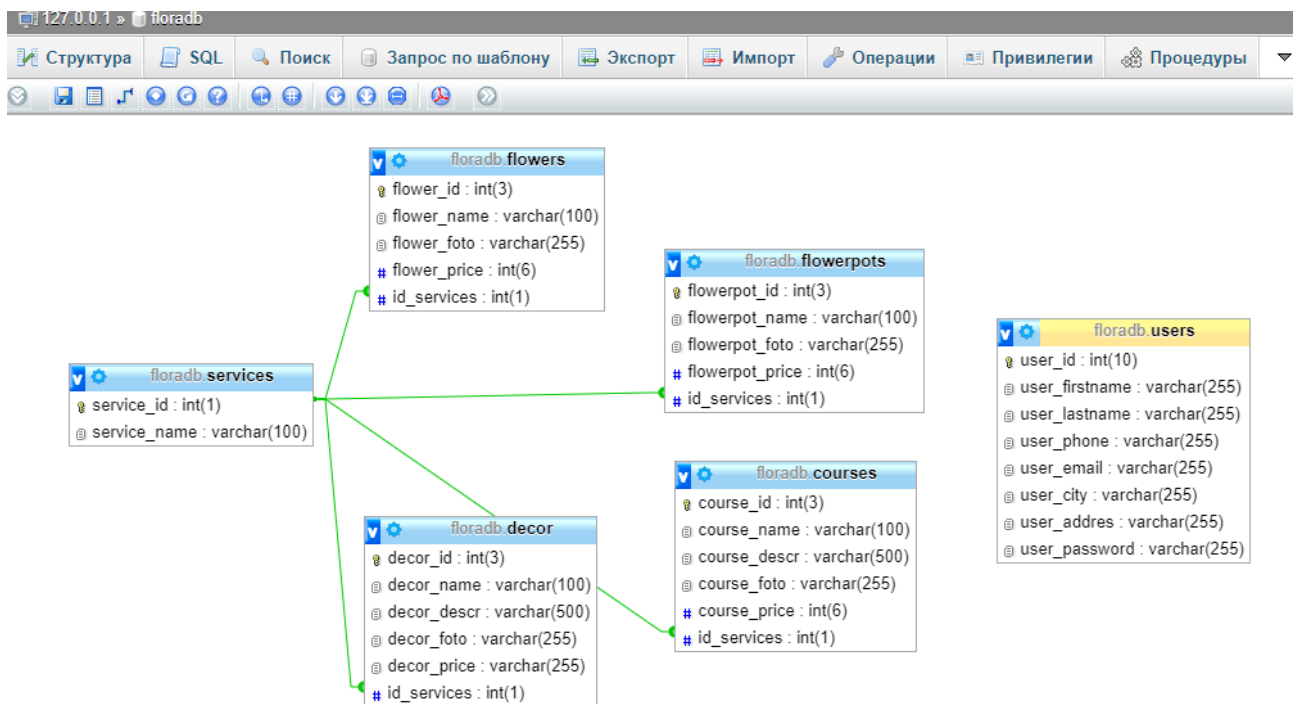


Рис. 2.4 Діаграма бази даних

Отже, в базі даних передбачено 6 таблиць. Для кожної таблиці передбачено первинні ключі: service_id для services, flower_id для flowers, flowerpot_id для flowerpot, decor_id для décor, course_id для courses, user_id для users – оскільки ці поля є ключовими, вони підпорядковують собі всі інші поля, таким чином у базі даних встановлено функціональні залежності.

У таблицях flowers, flowerpots, décor, courses зберігається інформація про товари та послуги студії флористики і є зовнішній ключ – id_services, встановлений з метою прив'язки цих таблиць до таблиці services.

Таблиця users – містить дані про зареєстрованих користувачів сайту і не має зв'язку з іншими таблицями, оскільки в ній немає даних, які відносяться до послуг, що надає студія.

2.2.1 Опис моделі даних

База даних сайту містить наступні таблиці та їх поля:

1. «users» – таблиця для даних зареєстрованих клієнтів:
 - a. «user_id»;
 - b. «user_firstname»;
 - c. «user_lastname»;
 - d. «user_phone»;
 - e. «user_email»;
 - f. «user_city»;
 - g. «user_addres»;
 - h. «user_password»;
2. «services» – таблиця для даних про послуги:
 - a. «service_id»;
 - b. «service_name»;
3. «flowers» – таблиця для даних про квіти:
 - a. «flower_id»;
 - b. «flower_name»;
 - c. «flower_foto»;
 - d. «flower_price»;
 - e. «id_services»;

4. «flowerpots» – таблиця для даних про кімнатні рослини:
 - a. «flowerpot_id»;
 - b. «flowerpot_name»;
 - c. «flowerpot_foto»;
 - d. «flowerpot_price»;
 - e. «id_services»;
5. «decor» - таблиця для даних про послуги декорування свят:
 - a. «décor_id»;
 - b. «décor_name»;
 - c. «décor_descr»;
 - d. «décor_foto»;
 - e. «décor_price»;
 - f. «id_services»;
6. «courses» – таблиця для даних про курси флористики:
 - a. «course_id»;
 - b. «course_name»;
 - c. «course_descr»;
 - d. «course_foto»;
 - e. «course_price»;
 - f. «id_services»;

2.2.2 Запити до таблиць бази даних

Web-сайт реалізовує наступні запити до бази даних:

1. Запити типу SELECT - оператор мови SQL, який повертає рядки з однієї чи багатьох таблиць. У більшості застосунків, SELECT — найчастіша команда Data Manipulation Language (DML). Оскільки, SQL не є процедурною мовою, запит SELECT описує кінцеві дані, однак, не вказує, які саме операції слід виконати для отримання цих даних: завдання

покладається на систему керування базами даних, яка здатна самостійно оптимізовувати необхідні для отримання результату операції.

Прикладом використання запиту SELECT у кодї є:

- a. `SELECT * FROM flowers` - запит відображає усі дані з таблиці «flowers» на сторінці «Квіти»;
 - b. `SELECT * FROM flowerpots` - запит відображає усі дані з таблиці «flowerpots» на сторінці «Кімнатні рослини»;
 - c. `SELECT * FROM courses` - запит відображає усі дані з таблиці «courses» на сторінці «Курси флористики»;
 - d. `SELECT * FROM decor` - запит відображає усі дані з таблиці «decor» на сторінці «Декор»;
 - e. `SELECT * FROM flowers WHERE flower_id = {$flowerid}` - запит відображає дані певного товару, з таблиці «flowers», у корзині;
 - f. `SELECT * FROM flowerpots WHERE flowerpot_id = {$flowerpotid}` - запит відображає дані певного товару, з таблиці «flowerpots», у корзині;
 - g. `SELECT user_email, user_password FROM users WHERE user_email = " .
$_POST["user_email"] . " and user_password = "
$_POST["user_password"] . "` - запит, який реалізовує авторизацію користувача на сайті;
 - h. `SELECT * FROM users WHERE user_email={$uid}` - запит відображає дані певного користувача, з таблиці users, в особистому кабінеті користувача;
2. Запити типу INSERT - оператор мови SQL, який додає рядки в таблицю.

Прикладом використання запиту INSERT у кодї є:

- i. `INSERT INTO users (user_firstname, user_lastname, user_phone,
user_email, user_city, user_addres, user_password) VALUES
('$firstname', '$lastname', '$phone', '$email', '$city', '$addres',
'$password')` - запит для внесення нових даних в таблицю «users», що забезпечує реєстрацію нового користувача на сайті;

3. Запити типу UPDATE - оператор мови SQL, що дозволяє оновити значення в заданих стовпцях таблиці.

Прикладом використання запиту UPDATE в коді є:

- ii. UPDATE users SET user_firstname='\$user_firstname',
user_lastname='\$user_lastname', user_phone='\$user_phone',
user_email='\$user_email', user_city='\$user_city',
user_addres='\$user_addres', user_password='\$user_password' WHERE
user_email={ \$uid } – запит, що забезпечує редагування даних в
таблиці «users»

2.3 Тестування веб-сайту

Враховуючи специфіку розробки веб-програм більш доречною для тестування продукту є функціональна модель тестування (або тестування «чорного ящика»).

Згідно ISTQB, тестування чорного ящика - це:

- тестування, як функціональне, так і нефункціональне, яка не передбачає знання внутрішнього устрою компонента або системи.
- тест-дизайн, заснований на техніці чорного ящика - процедура написання або вибору тест-кейсів на основі аналізу функціональної або нефункціональної специфікації компонента або системи.

Метою цієї техніки є пошук помилок в таких категоріях:

- неправильно реалізовані або відсутні функції;
- помилки інтерфейсу;
- помилки в структурах даних або організації доступу до зовнішніх баз даних;

- помилки поведінки або недостатня продуктивність системи;

Таким чином, ми не маємо уявлення про структуру та внутрішній устрій системи. Потрібно концентруватися на тому, що програма робить, а не на тому, як вона це робить.

У зв'язку з цим необхідно розбити програмний продукт на функціональні частини та розробити набір тестів для кожної із них.

Розроблений програмний продукт можна розбити на наступні частини:

- гостьова частина
- частина зареєстрованого користувача

Для гостьової частини можна запропонувати наступні тести:

1. Перевірка коректного доступу до сайту.
2. Перевірка коректності додавання, накопичення і видалення товарів в кошику.
3. Тест реєстрації на сайті.
4. Тестування форми зворотного зв'язку

Заданий набір тестів дозволяє протестувати всі компоненти програми та перевірити коректність обробки вхідних та вихідних даних за короткий термін, що є вкрай важливим фактором при веб-розробці.

Для частини зареєстрованого користувача можна запропонувати наступні тести:

1. Перевірка коректного доступу до сайту.
2. Перевірка коректності додавання, накопичення і видалення товарів в кошику.
3. Тестування форми зворотного зв'язку.

4. Перевірка авторизації на сайті.

5. Перевірка коректності інформації в особистому кабінеті.

6. Перевірка можливості оновлення інформації в особистому кабінеті.

Важливо зазначити, що веб-сайт потребує доопрацювання та оновлення, а саме:

- надання адміністратору можливості керувати замовленнями;
- удосконалити роботу кошика товарів;
- додати можливість надання знижки постійним клієнтам.

ВИСНОВКИ

У результаті виконання курсової роботи, досліджено можливості технологій PHP, MySQL, HTML та CSS. З метою закріплення отриманих знань, створено програмний продукт у вигляді інтернет-магазину студії флористики, визначено потреби та можливості використання створеного сайту, спроектовано базу даних, а також запити до неї, налаштовано взаємодію сайту та бази даних.

У результаті опрацювання Розділу 1 здійснено техніко-економічне обґрунтування веб-сайту, була проаналізована предметна область, визначені функціонал та інтерфейс сайту.

В ході виконання практичного завдання було створено програмний продукт – етапи розробки веб-сайту описано у Розділі 2. В першу чергу представлено програмування клієнтської частини, що складається з опису основних дій при роботі з сайтом та короткого опису інтерфейсу сторінок сайту, видимих користувачу. Також подано уривки коду – серверні скрипти, які є характерною складовою програмування серверної частини та описано розробку бази даних – визначено модель даних та складено запити до бази даних, які забезпечують повноцінну роботу сайту. Проектування бази даних реалізовано у програмному середовищі phpMyAdmin. Окрім того було здійснено тестування сайту, в результаті якого доведено коректну роботу основних функцій та визначено основні напрямки для доопрацювання сайту.

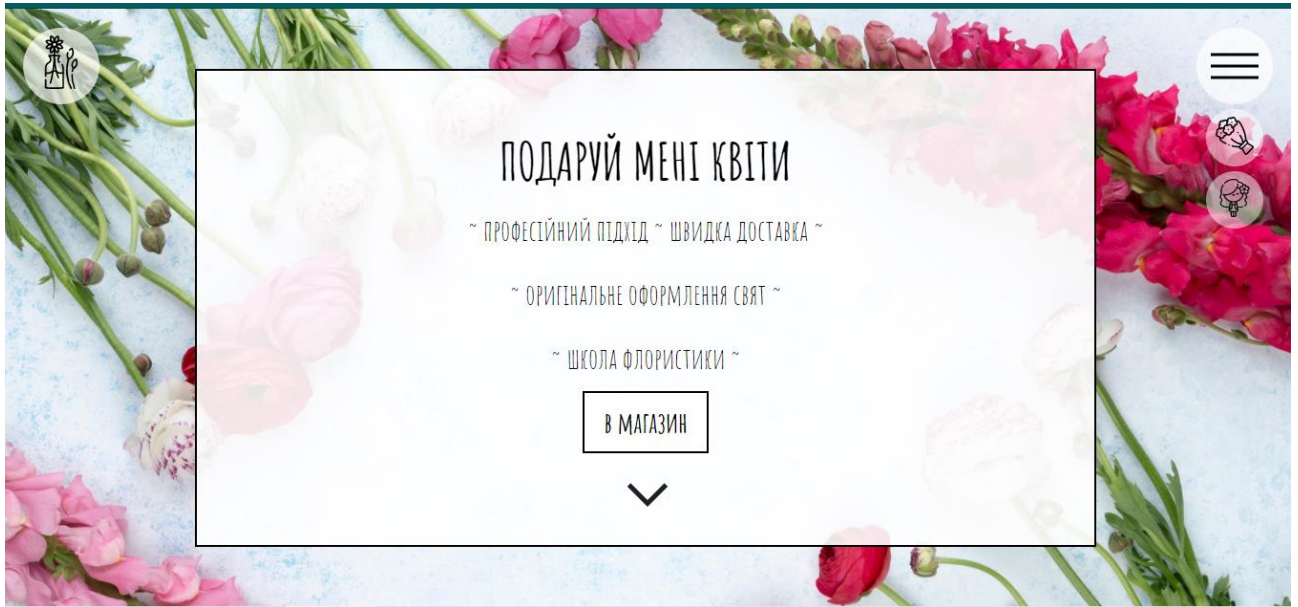
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Terry Felke-Morris. Web Development and Design Foundations with HTML5 / Terry Felke-Morris., 2018. – 720 с. – (9th Edition).
2. Jennifer Robbins. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics / Jennifer Robbins., 2018. – 810 с.
3. Mike O'Kane. A Web-Based Introduction to Programming / Mike O'Kane., 2017. – 628 с.
4. Larry Ullman. PHP and MySQL for Dynamic Web Sites / Larry Ullman., 2017. – 704 с.
5. Carlos Coronel. Database Systems: Design, Implementation, & Management / Carlos Coronel, Steven Morris., 2019. – 816 с.
6. Thomas Nield. Getting Started with SQL: A Hands-On Approach for Beginners / Thomas Nield., 2016. – 134 с.
7. .Фрейн Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Б. Фрейн. – СПб.: Питер, 2014. – 304 с.
8. CSS Tutorial [Электронный ресурс] – Режим доступа до ресурсу: <https://www.w3schools.com/css/default.asp>.
9. HTML Tutorial [Электронный ресурс] – Режим доступа до ресурсу: <https://www.w3schools.com/html/default.asp>.
10. SQL Tutorial [Электронный ресурс] – Режим доступа до ресурсу: <https://www.w3schools.com/sql/default.asp>.
11. PHP Tutorial [Электронный ресурс] – Режим доступа до ресурсу: <https://www.w3schools.com/php/default.asp>.
12. JavaScript Tutorial [Электронный ресурс] – Режим доступа до ресурсу: <https://www.w3schools.com/js/default.asp>
13. Создание PHP+MySQL сайта [Электронный ресурс] – Режим доступа до ресурсу: <https://habr.com/ru/sandbox/14484/>.
14. phpMyAdmin Tutorials [Электронный ресурс] – Режим доступа до ресурсу: <https://www.siteground.com/tutorials/phpmyadmin/>

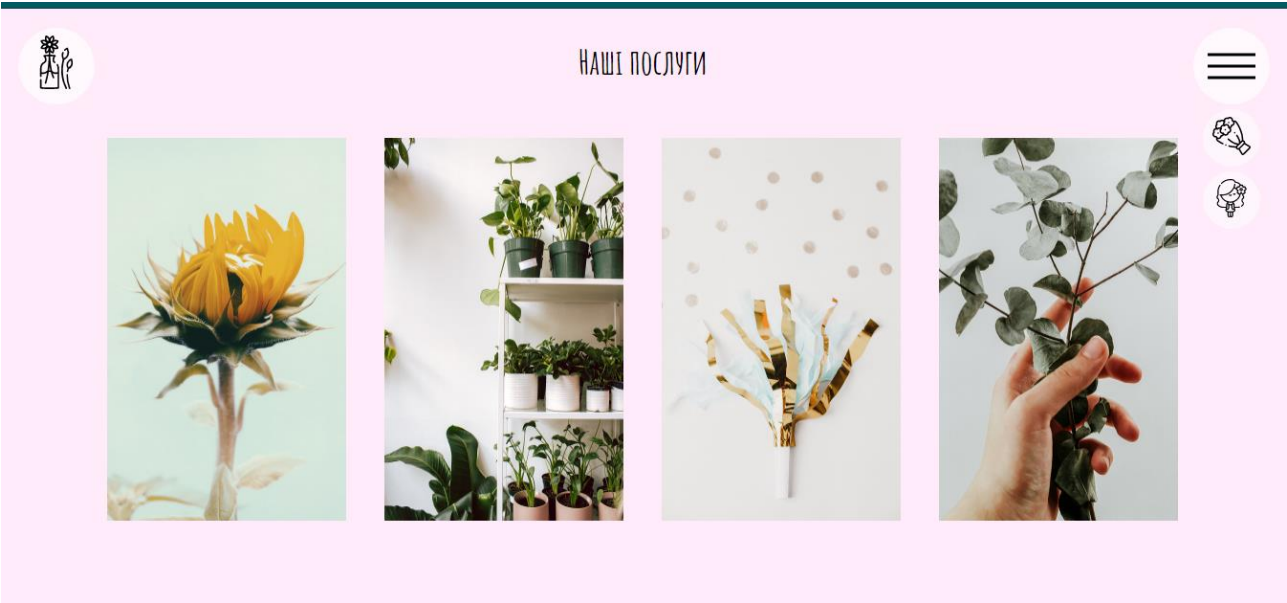
ДОДАТКИ

Додаток А

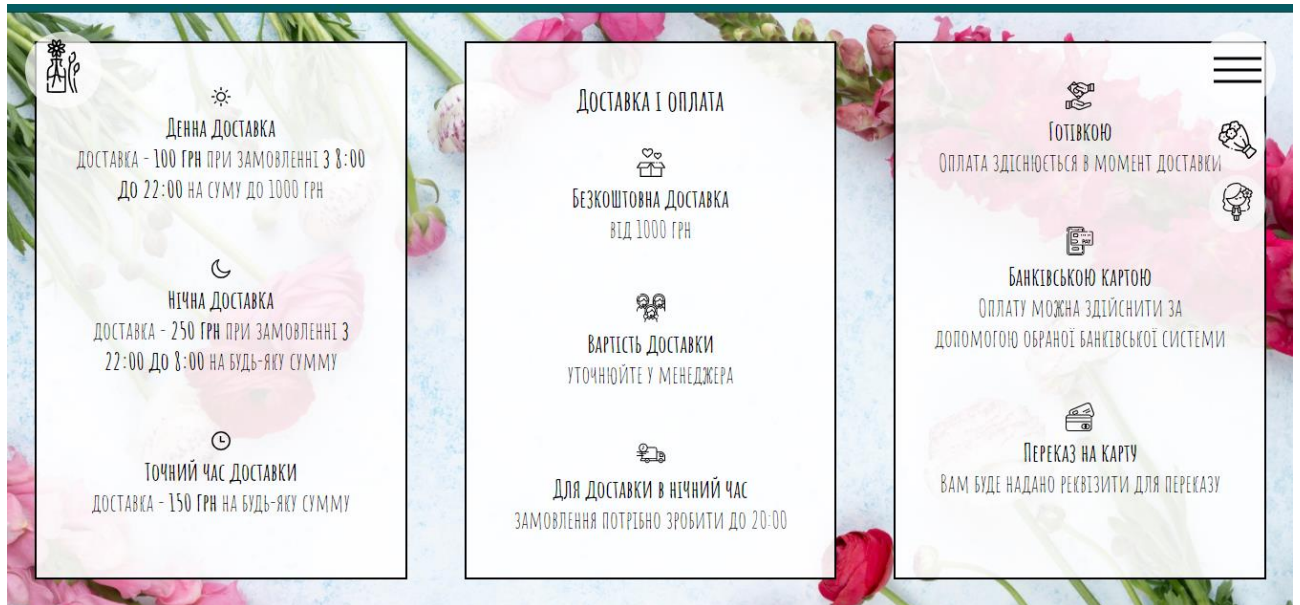
Головна сторінка – секція «Головна»



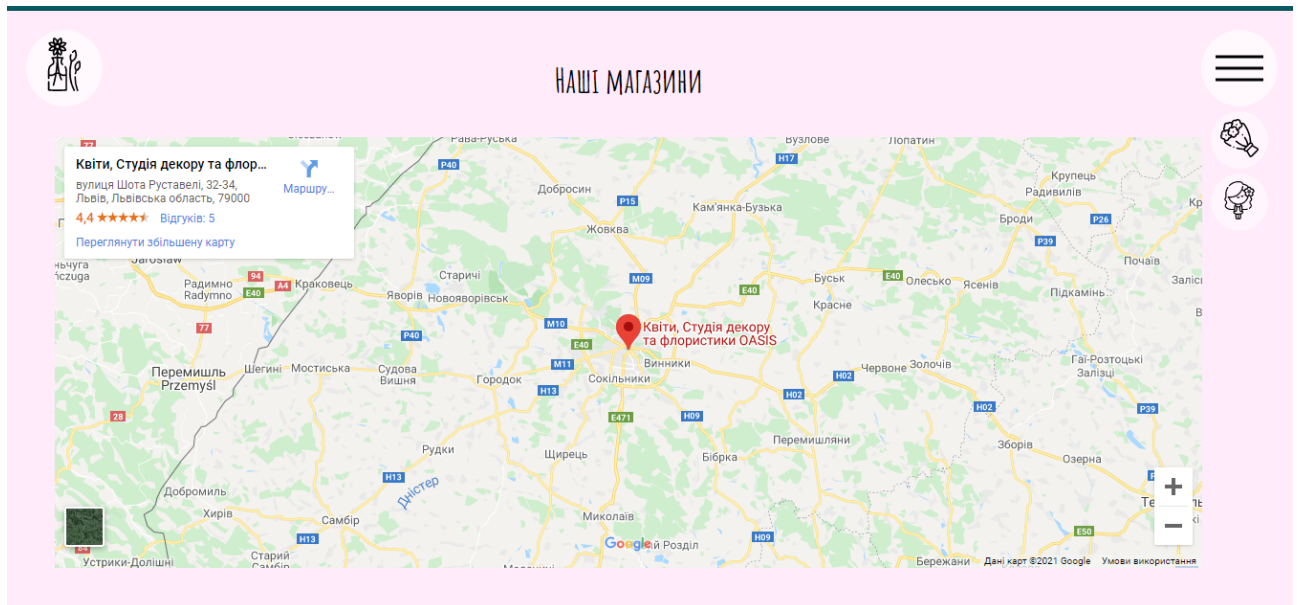
Головна сторінка – секція «Послуги»



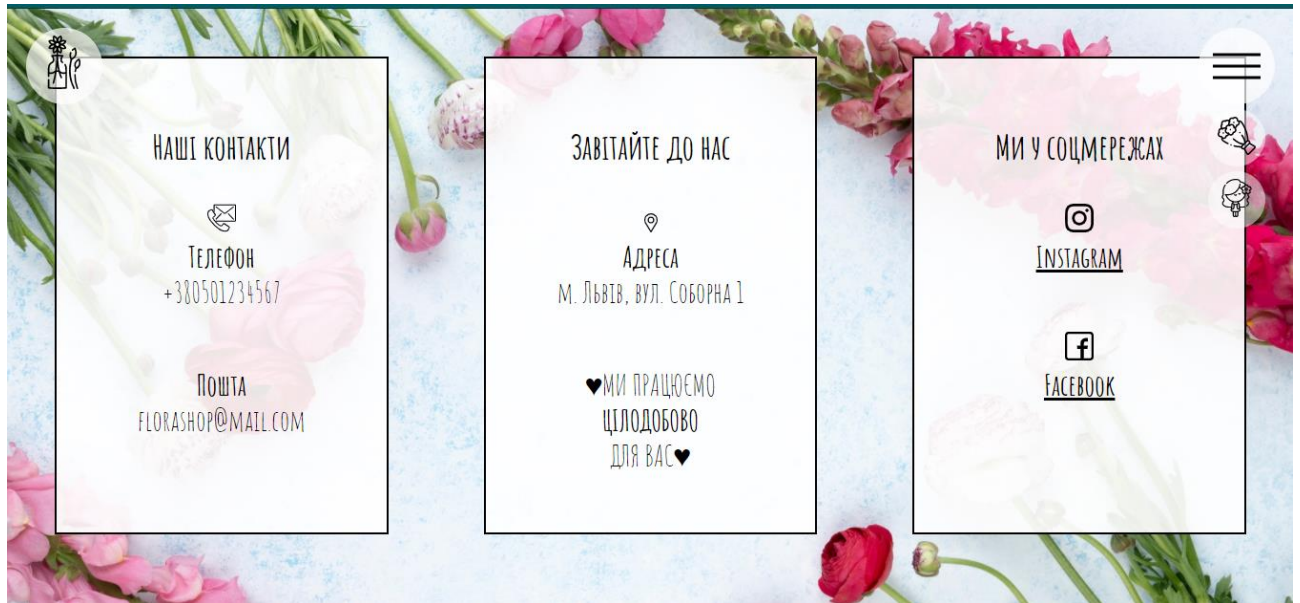
Головна сторінка – секція «Доставка і оплата»




Головна сторінка – секція «Наші магазини»




Головна сторінка – «Контакти»




Сторінка з товарами




КВІТКОВІ КОМПОЗИЦІЇ








БУЗОК
1590 UAN




ТРОЯНДА ВІКТОРІЯ
2215 UAN




ГОРТЕНЗІЯ
5355 UAN




Сторінка з товарами




РОСЛИНИ ДЛЯ ВАШОГО ДОМУ








АНТУРІУМ
350UAN



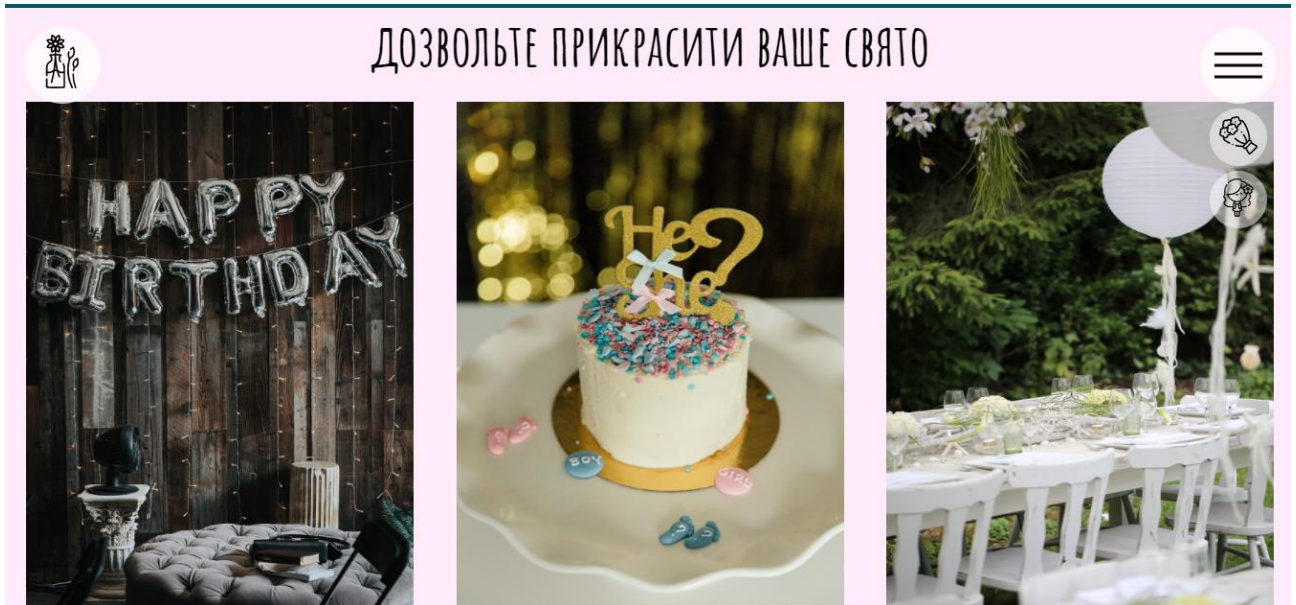
КАЛАНХОЄ
225UAN



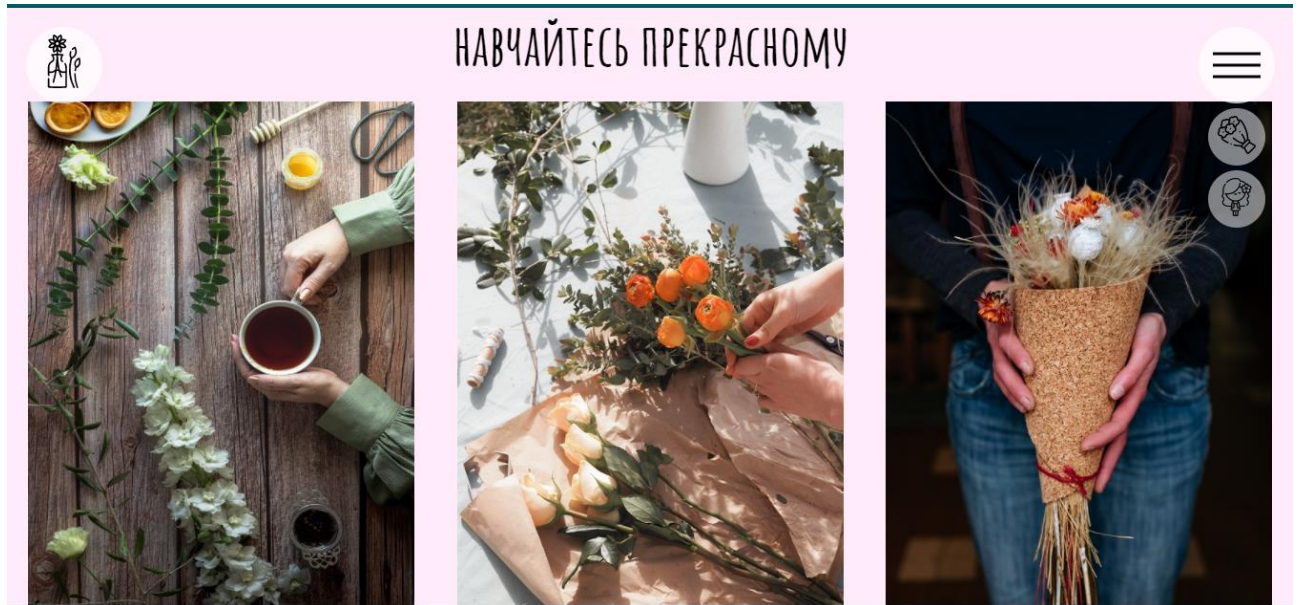
БЕГОНІЯ
350UAN

Сторінка з послугами



Сторінка з послугами



Корзина з товарами




ПЕРЕГЛЯД ОБРАНИХ ТОВАРІВ







У ВАШІЙ КОРЗИНІ
НЕМАЄ ТОВАРІВ

В МАГАЗИН



ПЕРЕГЛЯД ОБРАНИХ ТОВАРІВ






ТЮЛЬПАНИ
850 UAN

ОФОРМИТИ ЗАМОВЛЕННЯ

Сторінка з формою оформлення замовлення



ВВЕДІТЬ НАСТУПНІ ДАНІ, ЩОБ ОФОРМИТИ ЗАМОВЛЕННЯ



Ім'я:

Прізвище:

Телефон:

ЕМАІЛ:

МІСТО:

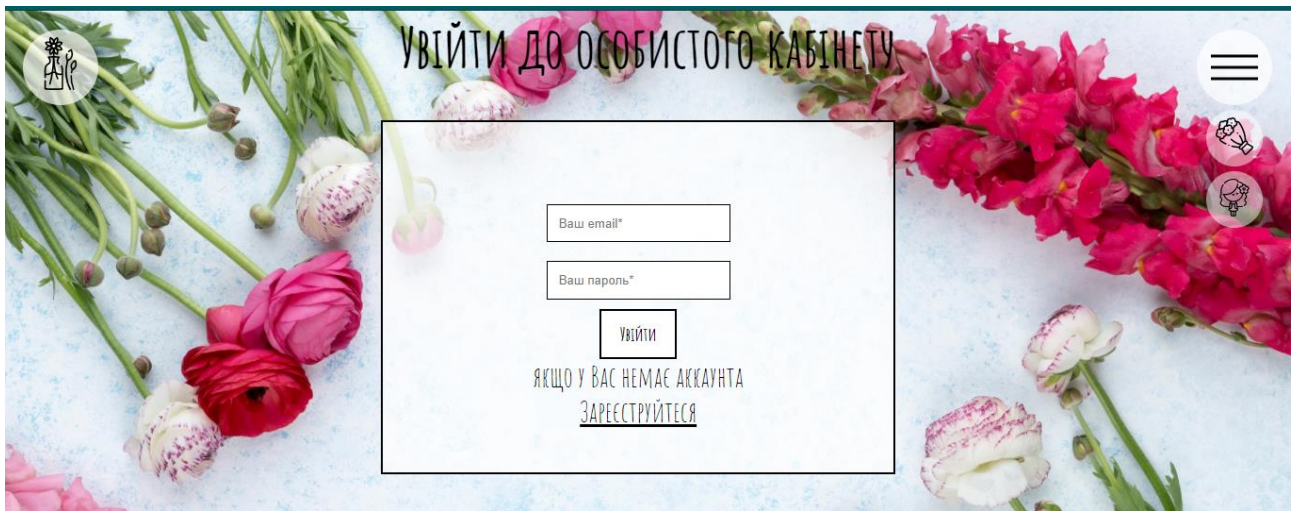
Адреса:

Спосіб оплати:

Замовити

[www.flora](#)

Сторінка з формою авторизації

The image shows a login form overlaid on a background of pink and white flowers. The form is titled "УВІЙТИ ДО ОСОБИСТОГО КАБІНЕТУ" (Log in to personal cabinet). It contains two input fields: "Ваш email*" (Your email) and "Ваш пароль*" (Your password). Below these fields is a "УВІЙТИ" (Log in) button. At the bottom of the form, there is a link that says "ЯКЩО У ВАС НЕМАЄ АКАУНТА ЗАРЕЄСТРУЙТЕСЯ" (If you do not have an account, register). The background features a variety of flowers, including tulips and ranunculus, in shades of pink and white, set against a light blue textured surface.

УВІЙТИ ДО ОСОБИСТОГО КАБІНЕТУ


Ваш email*

Ваш пароль*




УВІЙТИ

ЯКЩО У ВАС НЕМАЄ АКАУНТА
ЗАРЕЄСТРУЙТЕСЯ

Сторінка з формою реєстрації



ВВЕДІТЬ НАСТУПНІ ДАНІ, ЩОБ ЗАРЕЄСТРУВАТИСЯ



ІМ'Я:

ПРИЗВИЩЕ:

ТЕЛЕФОН:

EMAIL:

МІСТО:

АДРЕСА:

ПАРОЛЬ:

ЗАРЕЄСТРУВАТИСЯ