

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 122 «Комп'ютерні науки,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: «Web-додаток для підтримки продажу декоративних товарів для інтер'єру»

Здобувачки групи ІТ-91 Самочкіної Тетяни Валентинівни .
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ (підпис)

Тетяна САМОЧКІНА
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник доцент кафедри ІТ, к.т.н., доцент Юлія ПАРФЕНЕНКО

(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ)

_____ (підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології
проектування»

ЗАТВЕРДЖУЮ

В. о. зав. кафедри ІТ

_____Світлана ВАЩЕНКО

« _____ » _____ 2023 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Самочкіна Тетяна Валентинівна

1 Тема роботи Web-додаток для підтримки продажу декоративних товарів для інтер'єру .

керівник роботи Парфененко Юлія Вікторівна, к.т.н., доцент _____ ,

затверджені наказом по університету від «29» травня 2023 р. №0588-VI

2 Строк подання студентом роботи «09» червня 2023 р.

3 Вхідні дані до роботи текстові та графічні матеріали для наповнення web-додатку _____ .

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, проектування web-додатку, розробка web-додатку _____ .

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) дослідження продуктів-аналогів, контекстна діаграма в нотації IDEF0, декомпозиція контекстної діаграми, діаграма варіантів використання, фізична діаграма бази даних, структура проекту, представлення сторінок web-додатку.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 08.02.2023

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі проєкту	01.03.2023 -10.03.2023	
2	Аналіз додатків-аналогів	11.03.2023 -12.03.2023	
3	Складання технічного завдання	13.03.2023 -28.03.2023	
4	Проектування web-додатку	29.04.2023 - 17.04.2023	
5	Розробка web-додатку	18.04.2023 -02.05.2023	
6	Тестування web-додатку	03.05.2023 -15.05.2023	
7	Завантаження на хостинг	16.05.2023 -17.05.2023	
8	Оформлення документації	18.05.2023 -07.06.2023	

Студент

Тетяна САМОЧКІНА

(підпис)

Керівник роботи

т.н., доц. Юлія
ПАРФЕНЕНКО

(підпис)

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток для підтримки продажу декоративних товарів для інтер'єру».

Пояснювальна записка складається із таких розділів: вступ, аналіз предметної області, моделювання та проектування, програмна реалізація, висновки, список використаних джерел (17 посилань) та додатків А, Б, В, Г. Загальний обсяг роботи складає 96 сторінок: 57 сторінок основного тексту, 2 сторінки списку використаних джерел, 37 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці web-додатку для підтримки продажу декоративних товарів для інтер'єру.

У першому розділі проведено огляд останніх досліджень і публікацій, що стосуються web-додатків для підтримки продажу декоративних товарів. Також здійснено аналіз програмних продуктів-аналогів, які вже існують на ринку. Встановлено критерії для близького до ідеалу додатку.

У другому розділі здійснено моделювання і проектування web-додатку. Показано контекстну діаграму в нотації IDEF0 і її декомпозицію, а також діаграму варіантів використання. Розроблено модель бази даних.

У третьому розділі описано архітектуру інформаційної системи, її інтерфейс та функції. Розроблено серверну частину з використанням Laravel та мови програмування PHP.

Ключові слова: web-додаток, інтернет-додаток web-розробка, розробка web, Laravel, PHP, SQL, інформаційна система, підтримка продажів, декоративні товари.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Огляд останніх досліджень і публікацій.....	7
1.2 Аналіз програмних продуктів – аналогів	11
1.3 Постановка задачі.....	19
1.4 Вибір засобів реалізацій розробки web-додатку.....	20
2 Моделювання та проектування.....	23
2.1 Моделювання інформаційної системи.....	23
2.2 Діаграма варіантів використання	25
3 Програмна реалізація.....	33
3.1 Архітектура web додатку	33
3.2 Програмна реалізація.....	38
3.3 Використання програмного додатку.....	42
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
Додаток А.....	58
Додаток Б	70
Додаток В.....	76
Додаток Г	96

ВСТУП

У сучасному світі, де все більше людей віддає перевагу онлайн-торгівлі, web-додаток магазину є дуже актуальним. Онлайн-магазини дозволяють покупцям здійснювати покупки з будь-якого місця з доступом до інтернету, забезпечуючи зручність та доступність. Web-додаток магазину забезпечує можливість зберігати та обробляти великі обсяги даних, що дозволяє аналізувати поведінку покупців, відстежувати їхній вибір товарів та здійснювати персоналізовану рекламу і маркетингові акції.

Web-додаток для підтримки продажів є актуальним і корисним інструментом для бізнесу будь-якого розміру і галузі. Основні переваги web-додатків для підтримки продажів полягають у тому, що вони дозволяють бізнесу ефективно керувати своїми продажами та клієнтською базою, а також забезпечують швидкий та легкий доступ до інформації про продукти або послуги. Наприклад, web-додатки можуть допомогти збирати та аналізувати дані про клієнтів, здійснювати моніторинг замовлень та виконання доставок. Застосування web-додатків для підтримки продажів може допомогти бізнесу збільшити ефективність своєї роботи, підвищити якість обслуговування клієнтів та збільшити прибуток.

Отже, метою кваліфікаційної роботи бакалавра є розробка web-додатку, який буде сприяти підтримці продажу декоративних товарів для інтер'єру. Додаток має мати сучасний інтерфейс та функціонал, що допомагатиме автоматизувати робочі процеси, пов'язані з продажами. Такий підхід сприятиме поліпшенню ефективності та продуктивності бізнесу, а також покращенню взаємодії з клієнтами та забезпеченню зручності користування додатком.

Для досягнення мети проекту необхідно виконати наступні задачі:

- визначити актуальність роботи,
- провести моделювання та проектування інформаційної системи,
- реалізувати програмну частину web-додатку,
- провести протестування робочого web-додатку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Відповідно до теми кваліфікаційної роботи було досліджено та проаналізовано ряд вітчизняних та зарубіжних наукових досліджень і публікацій. Першочергово було досліджено матеріали про електронну комерцію, її види та переваги. Електронна комерція –це процес придбання і продажу товарів та послуг через Інтернет [1]. Однією з головних переваг електронної комерції є можливість здійснювати покупки безпосередньо з дому, зручніше та швидше, ніж традиційний шопінг у магазинах. Крім того, електронна комерція дозволяє зменшити витрати на оренду приміщень, утримання персоналу та інші операційні витрати, що може сприяти зниженню цін на товари та послуги.

У зв'язку зі швидким розвитком технологій, електронна комерція стає все більш автоматизованою. Наприклад, для підтримки продажів у онлайн-магазинах використовуються спеціалізовані програмні засоби – CRM-системи, ERP-системи, системи аналізу даних та інші. Вони дозволяють автоматизувати процеси замовлення, оплати, доставки та обліку товарів, що збільшує ефективність роботи та знижує ризики помилок.

Трішки детальніше про CRM- та ERP-системи. CRM (Customer Relationship Management) розроблена для відстеження та аналізу всіх взаємодій компанії з клієнтами для того, щоб оптимізувати обсяги продажів та зробити роботу з клієнтами більш ефективною. Типовими CRM-процесами є: обробка запитів, інформування, розв'язання проблем та керування скаргами клієнтів. ERP (Enterprise Resource Planning) створена для управління бізнесом, вона відстежує бізнес-ресурси та полегшує потік інформації полегшує між усіма бізнес-функціями та керує зв'язками із зовнішніми зацікавленими сторонами. Забезпечує такі функції, як фінансовий облік, людські ресурси, обробка замовлень та управління проектами [2].

Електронна комерція відкриває двері для інноваційних рішень і технологій. Наприклад, впровадження штучного інтелекту, машинного навчання та аналізу даних

дозволяє підприємствам вдосконалювати персоналізацію, прогнозувати попит та покращувати процеси маркетингу.

Розглядаючи питання розробки web-додатку продажу товарів, було досліджено, які саме налаштування потрібно провести для нього. У першу чергу потрібно обрати платформу для електронної комерції (наприклад, Shopify, WooCommerce (для WordPress), Magento або OpenCart), яка надає інструменти для створення та управління вашим інтернет-магазином. Потім – обрати домен та хостинг, хоч в деяких випадках його пропонують платформи, існує багато провайдерів доменних імен та хостингу, які пропонують різноманітні послуги. Ось кілька популярних доменів: GoDaddy, Namecheap, Google Domains, Bluehost, HostGator. І декілька хостингів: Bluehost, HostGator, SiteGround, DreamHost, InMotion Hosting [3]. Ці провайдери пропонують широкий спектр послуг, включаючи реєстрацію доменних імен, хостинг, SSL-сертифікати, електронну пошту та інші додаткові функції. Далі – визначитися з платіжною системою, яка дозволить клієнтам здійснювати оплату онлайн. Також обов'язково потрібно розглянути політику конфіденційності та захисту особистих даних [4].

На останній пункт варто звернути особливу увагу, адже безпека є важливими аспектами для будь-якого web-додатку. Ключові аспекти, на які варто звернути увагу:

- SSL-шифрування: встановлення SSL-сертифіката дозволяє шифрувати передачу даних між клієнтом і сервером, забезпечуючи конфіденційність та захист від перехоплення чи зловживання даними;
- захист особистих даних: розробка і впровадження політики конфіденційності, яка пояснює, які дані збираються, як вони використовуються та з ким можуть бути поділені;
- безпека платежів: використання безпечних платіжних шлюзів, що гарантують захищену обробку платежів;
- резервне копіювання даних: створення резервних копій web-додатку та бази даних, що допоможе запобігти втраті даних у разі випадкового видалення, вторгнення або інших проблем;

- оновлення програмного забезпечення: потрібно переконатися, що платформа електронної комерції, теми та плагіни завжди оновлені до останніх версій [5].

Узагальнивши все те, що описано вище: електронна комерція є потужним інструментом для розвитку бізнесу і приваблення клієнтів у світі онлайн. Вона надає можливість продавати товари і послуги широкій аудиторії незалежно від географічного розташування. Щоб забезпечити успішну електронну комерцію, важливо налаштувати інтернет-магазин, забезпечити зручний і безпечний процес покупок, враховувати потреби та вимоги клієнтів, залучати їх увагу за допомогою маркетингових стратегій та забезпечувати якість обслуговування.

Під час роботи велику увагу було приділено технологіям розробки web-додатків. Для створення повноцінного web-додатку потрібно реалізувати його front-end (стосується всього, що користувач може побачити у web-додатку та з чим може взаємодіяти) та back-end (включає приховані механізми, які відповідають за функціонал web-додатку [6]).

Розробка цих двох частин зазвичай зводиться до знання найбільш популярної зв'язки технологій для створення web-додатків: HTML, CSS, JavaScript. За допомогою них можна застосовувати для розробки найрізноманітніші web-технології.

Було проведено ознайомлення з таким переліком web-технологій:

– браузері – запитують інформацію, а потім інтерпретують її так, як ми можемо зрозуміти.

– HTML і CSS – знають що та яким чином покращити користувачеві після отримання запиту.

– Web Development Frameworks – є відправною точкою та існують для уникнення простих і буденних завдань.

Було розглянуто такі фреймворки для web-розробки:

– Angular – створений спеціально для розробки динамічних web-додатків. Його функції включають шаблони, архітектуру MVC, генерацію та поділ коду тощо;

- Ruby on Rails – технологія web-додатків на стороні сервера, робить розробку швидкою та більш простою. Цю структуру виділяє можливість повторного використання програмного коду;
- Laravel – web-фреймворк PHP з відкритим кодом, що дає змогу створювати web-додатки для різних платформ;
- Yii – побудований на PHP5 та застосовується для розробки web-застосунків з відкритим кодом. Він є продуктивний та включає в себе перелік хороших інструментів для налаштування та тестування програм;
- Meteor JS – web-фреймворк JS з відкритим кодом, що дає змогу створювати web-додатки для різних платформ;
- Zend – платформа з відкритим кодом, заснована на PHP та націлена на створенні безпечних та надійних web-застосунків;
- Джанго – фреймворк написаний на Python, який у якості інструментів надає шаблони, ORM, адміністратор Django, моделі, що значно полегшує процес розробки web-додатку [7].

Узагальнюючи вищеописане: фреймворки є потужними інструментами, які допомагають розробникам створювати ефективні та функціональні web-додатки з меншими зусиллями. Вибір фреймворку залежить від конкретних потреб та вимог проекту, але усі вони спрямовані на полегшення розробки і покращення продуктивності.

Було приділено увагу також і користі web-додатків для продажу. Розробка додатку може призвести до збільшення прибутку, залучення нових клієнтів, а також до спрощення більшості бізнес-процесів за рахунок їх автоматизації. Останній пункт з точки зору підприємництва є дуже важливим та має ряд переваг в електронній комерції: дозволяє заощадити час, підвищити ефективність продаж, зменшити кількість помилок та покращити якість обслуговування клієнтів [8].

Варто визначити, що саме розуміється під терміном «автоматизація». Отже, автоматизація – це перемикання ручних завдань на автоматизовані за допомогою програмного забезпечення [9]. Завдяки їй такі завдання, як виставлення рахунків,

перевірка електронної пошти, сповіщення та інші подібні завдання, виконуються легко та швидко.

Також цікаво розглянути й більш складні процеси автоматизації. Наприклад, система рекомендацій з алгоритмами навчання. Вона є частиною машинного навчання та використовується для того, щоб дослідити вподобання людей, їх ідеї та порекомендувати їм схожі продукти відповідно до їх пошуку, історії замовлень, географічних, демографічних показників тощо [10].

Ще одною з корисних є автоматизація адрес (використовуючи API карти Google для автоматизації адреси доставки та адреси платежу). Форма автозаповнення адреси місця отримує вибрані адреси з бази даних Google Місця на карті та використовує їх для автоматичного заповнення форми адреси. Автоматичний заповнювач форми адреси важливий і корисний, коли ви використовуєте адресу доставки для онлайн-замовлення, платіжну адресу кредитної картки, форму бронювання спільних поїздок тощо [11].

Для соціальної комерції важливий когнітивний поведінковий аналіз замовника. За допомогою автоматизації можна відстежити та проаналізувати відгуки клієнтів, що є важливим для розуміння своєї аудиторії та вдосконалення відповідно до неї.

Отже, web-додатки для комерції мають велике значення і є корисними для підтримки бізнесу та бізнес-процесів. Для створення таких застосунків існує велика кількість web-технологій, особливу увагу варто приділити web-фрейморкам, які є гарними помічниками у спрощенні та пришвидшенні розробки web-додатків.

1.2 Аналіз програмних продуктів – аналогів

Для створення якісного web-додатку важливим етапом є аналіз існуючих аналогів, в результаті якого буде виявлено їх слабкі та сильні сторони. Під час нього варто звернути увагу на такі критерії, як дизайн, функціональність та зручність у використанні.

Для визначення вимог майбутнього web-додатку було проведено дослідження існуючих аналогів web-додатків онлайн-магазинів декору “Décor el Home”.

Розглянемо web-додаток магазину «Décor el Home» [12].

На головній сторінці (рис. 1.1) розміщені зрозуміле навігаційне меню, категорії товарів, пошуковий рядок, контактна інформація, корисна інформація для покупця та графік роботи.

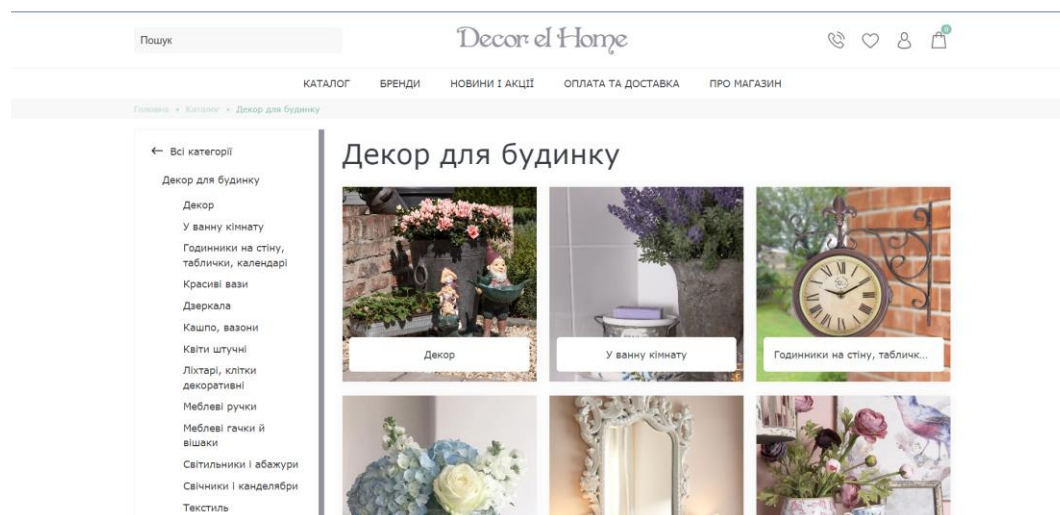


Рисунок 1.1 – Головна сторінка web-додатку “ Décor el Home ”

У каталог можна перейти з навігаційного меню (рис. 1.2).

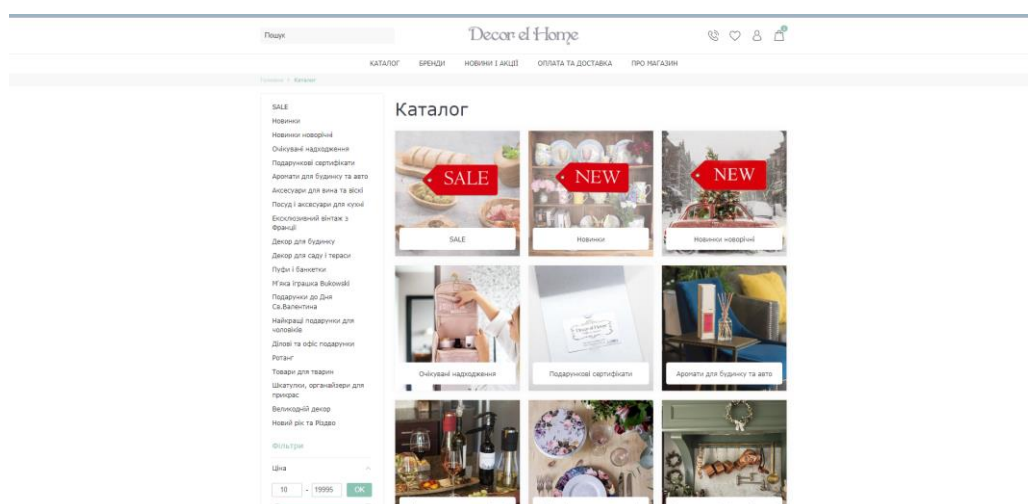


Рисунок 1.2– Сторінка каталогу web-додатку “ Décor el Home ”

Обравши категорію, можна обрати і товар. Доступно оформлена сторінка товару, є характеристики, багато якісних фото та інформація щодо замовлення та доставки.

Замовити продукцію можна, додавши товар до кошика, а потім на сторінці “Кошик” оформити замовлення (рис. 1.3).

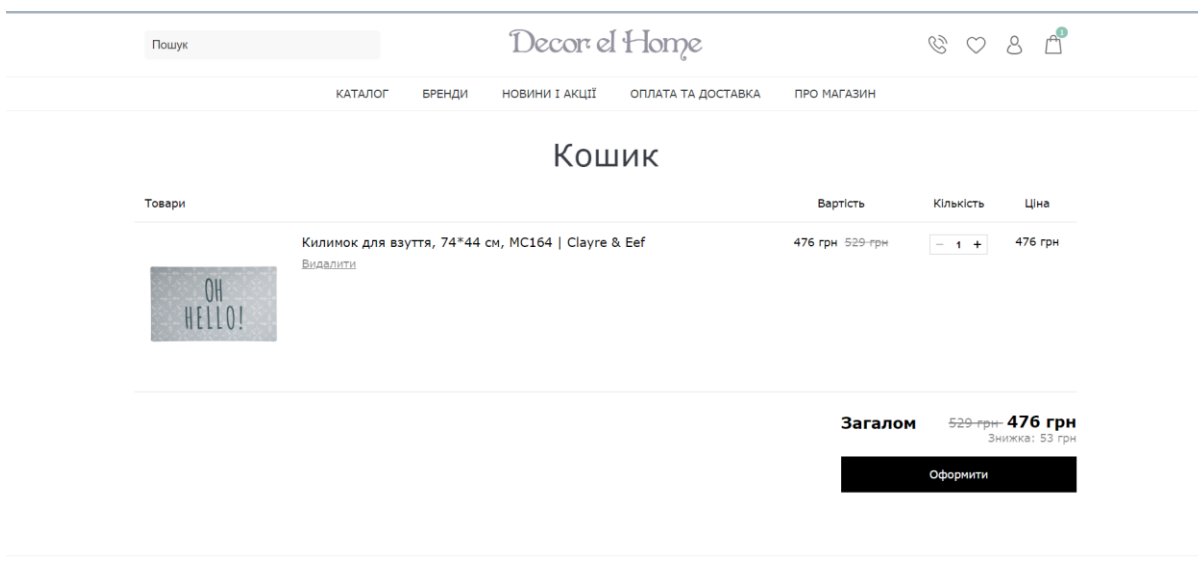


Рисунок 1.3 – Кошик web-додатку “Décor el Home”

На рис.1.4 представлена форма для оформлення замовлення

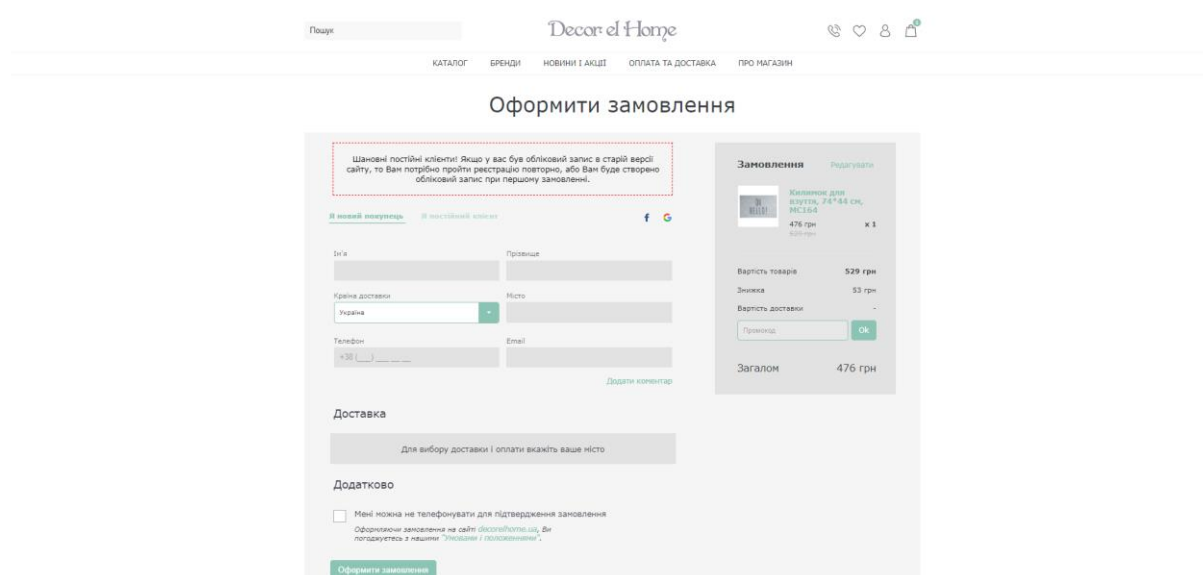


Рисунок 1.4 – Форма для оформлення товару у web-додатку “Décor el Home”

Із недоліків додатку-аналогу “Décor el Home” – відсутність інтерактивності.

Розглянемо web-додаток магазину «JYSK» [13]. На головній сторінці (рис. 1.5) розміщені навігаційне меню, категорії товарів, пошуковий рядок, контактна

інформація, корисна інформація для покупця. З першого погляду з'являється відчуття того, що всесвітньо відомий магазин має не дуже зручний web-додаток. Кидається в очі невдало оформлене навігаційне меню.

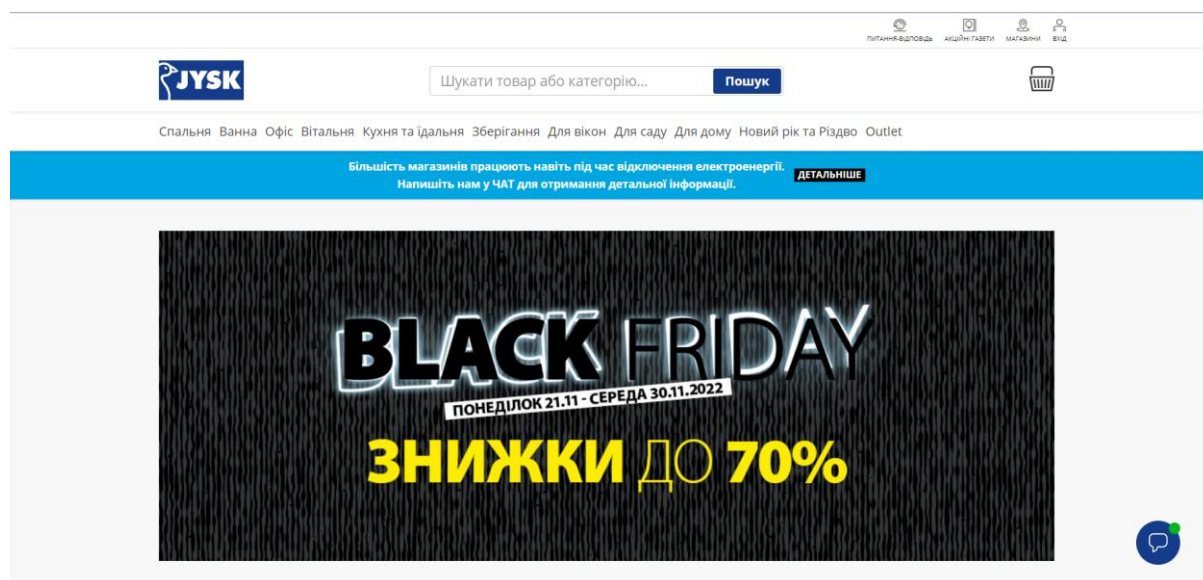


Рисунок 1.5 – Головна сторінка web-додатку “ JYSK ”

Навігаційне меню є каталогом, також категорії товарів є на головній сторінці (рис. 1.6).

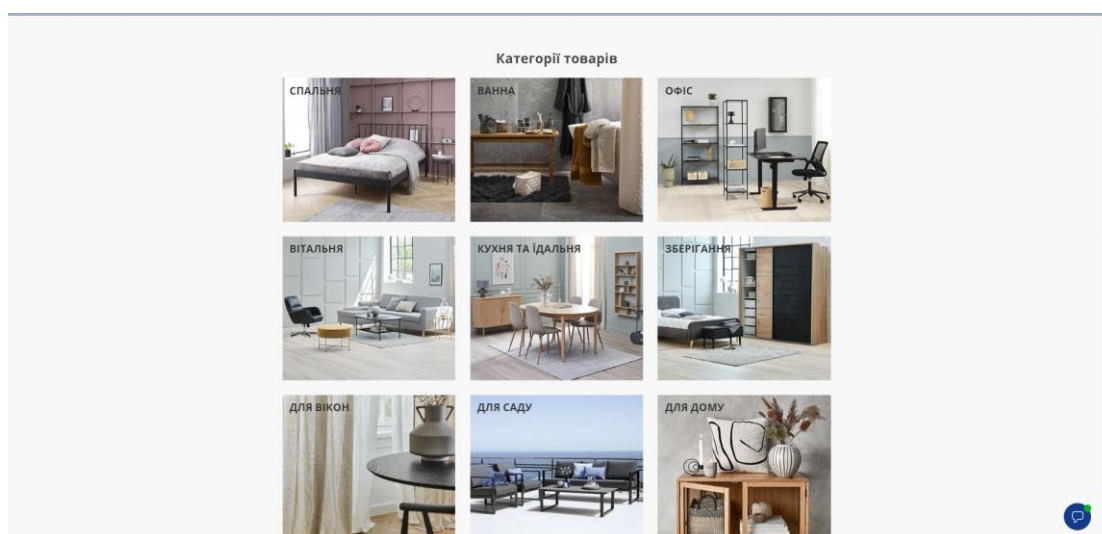


Рисунок 1.6 – Вигляд каталогу web-додатку “ JYSK ”

Обравши категорію, можна обрати і товар. Доступна оформлена сторінка товару, є характеристики, багато якісних фото та інформація щодо замовлення та доставки (рис. 1.7).

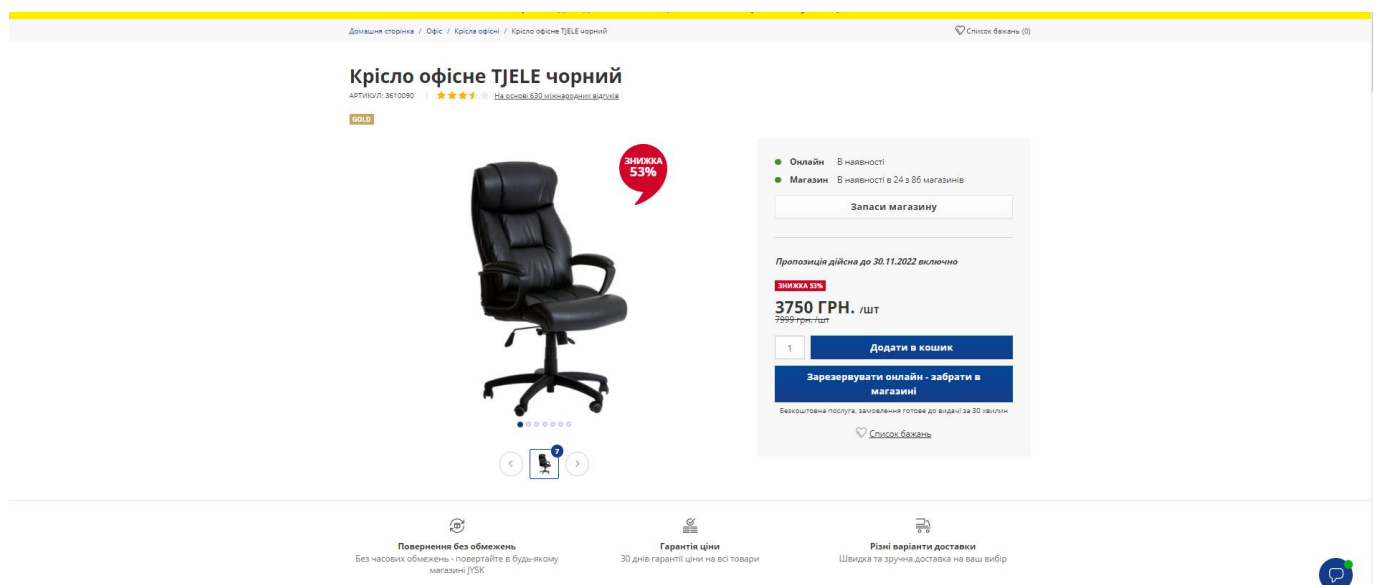


Рисунок 1.7– Сторінка товару web-додатку “ JYSK ”

Замовити продукцію можна, додавши товар до кошика, а потім на сторінці “Кошик” оформити замовлення (рис. 1.8).

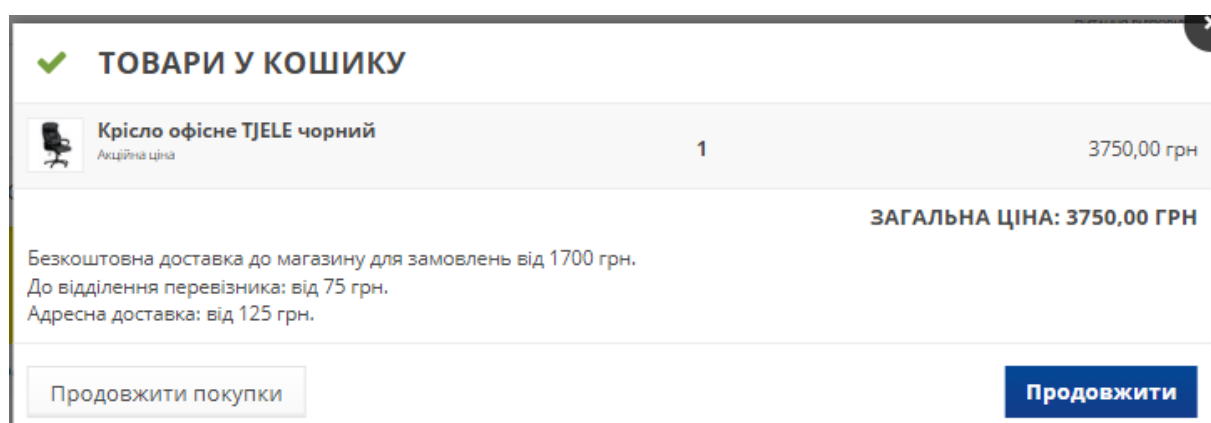


Рисунок 1.8– Кошик web-додатку “ JYSK ”

Так виглядає форма для оформлення замовлення (рис. 1.9).

The screenshot shows the 'Address' step of a purchase process on the JYSK website. The form contains the following fields:

- Ім'я *
- Но-батькові *
- Прізвище *
- Місто *
- Вулиця *
- Номер будинку *
- Поверх *
- Країна доставки - тільки Україна
- Ім'я *
- Мобільний телефон, наприклад, 0671234567 *

Additional information on the right side of the form:

- Кредитна картка ТРЕКІ червоний
- 1 шт. 3750,00 грн
- Доставка 0,00 грн
- Сума ПДВ 625,00 грн
- Усього 3750,00 грн**
- БЕЗПЕЧНИЙ ОНЛАЙН ПЛАТІЖ
- ГАРАНТІЯ ЦІНИ

Рисунок 1.9 – Форма для оформлення товару у web-додатку “ JYSK ”

Розглянемо web-додаток магазину «Décor Plus» [14]. На головній сторінці (рис. 1.10) розміщені навігаційне меню, категорії товарів, пошуковий рядок, контактна інформація, корисна інформація для покупця. Інтерфейс відштовхує. Також є проблеми з перекладом web-додатку: обираючи українську мову, можна все ж побачити багато слів російською.

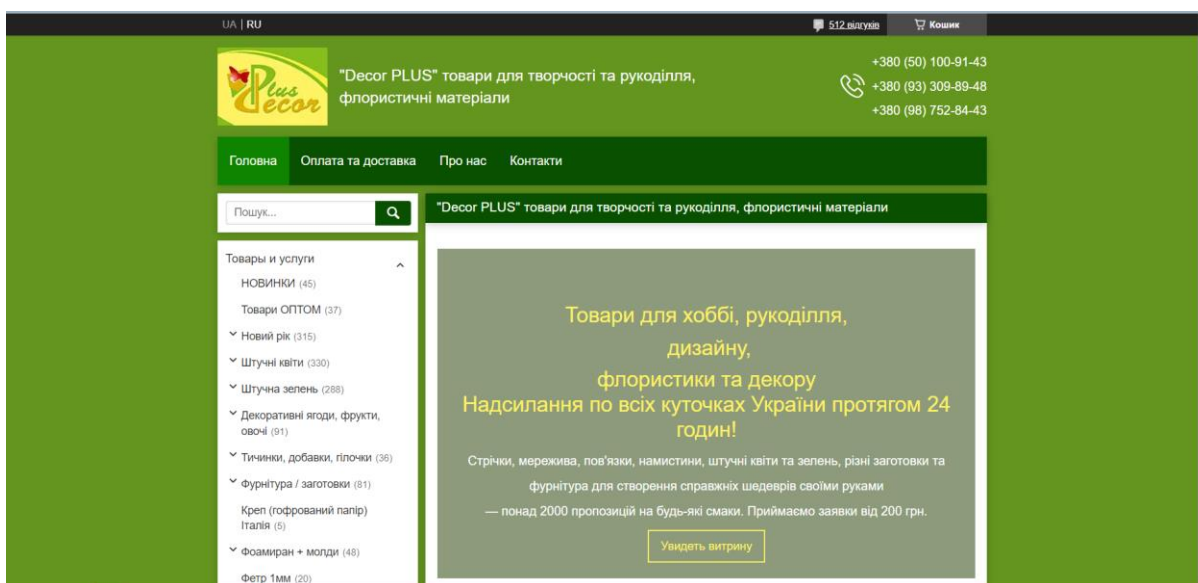


Рисунок 1.10– Головна сторінка web-додатку “ Décor Plus ”

Каталог знаходиться на головній сторінці (рис. 1.11).

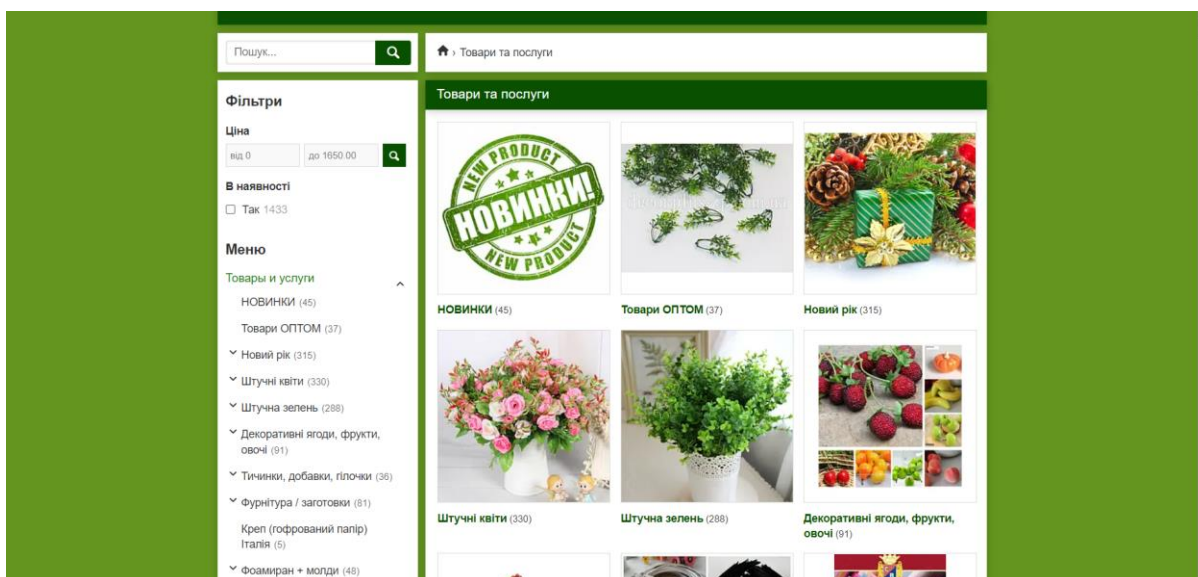


Рисунок 1.11 – Вигляд каталогу web-додатку “ Décor Plus ”

Обравши категорію, можна обрати і товар. Доступно оформлена сторінка товару, є характеристики та ціна (рис. 1.12).

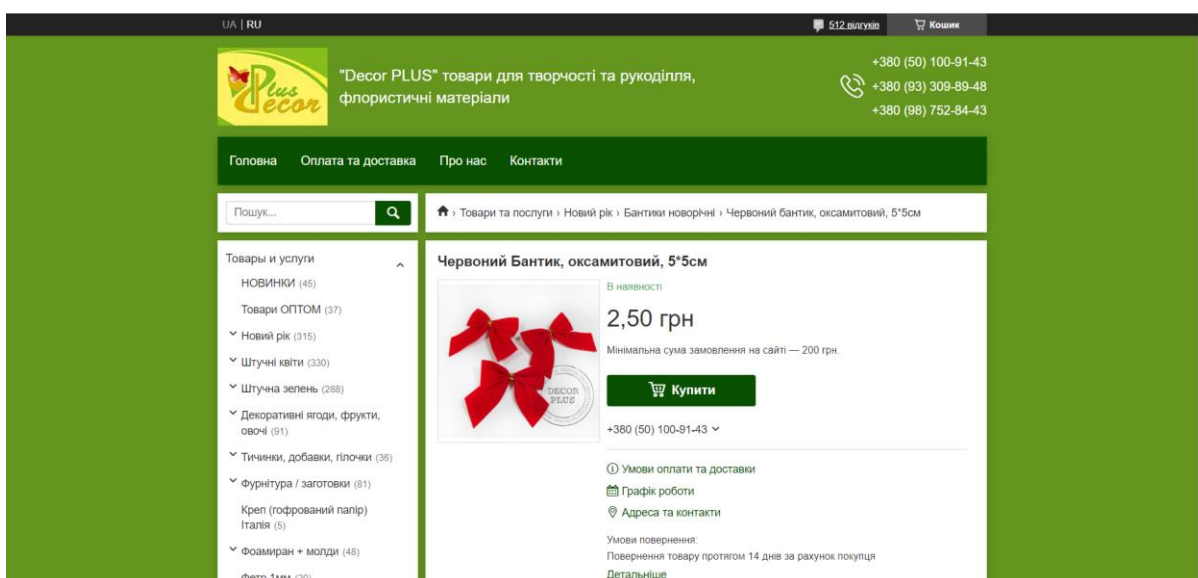


Рисунок 1.12 – Сторінка товару web-додатку “ Décor Plus ”

Замовити продукцію можна, додавши товар до кошика, а потім на сторінці “Кошик” оформити замовлення (рис. 1.13).

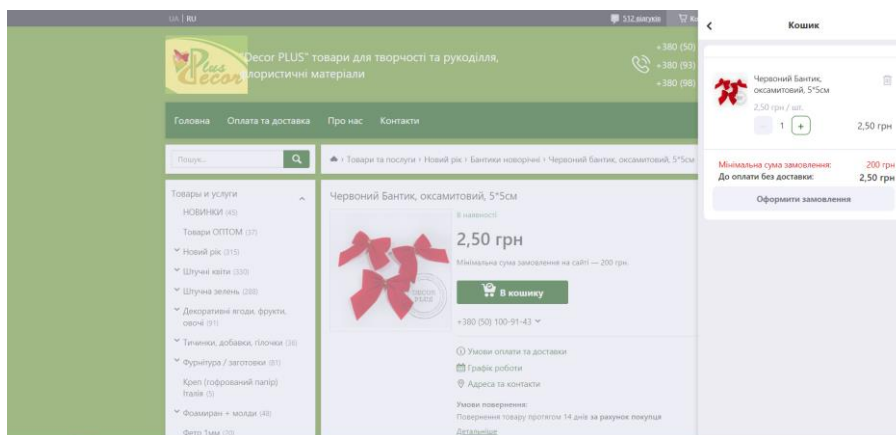


Рисунок 1.13– Кошик web-додатку “ DÉCOR PLUS ”
Форма для оформлення замовлення (1.14).

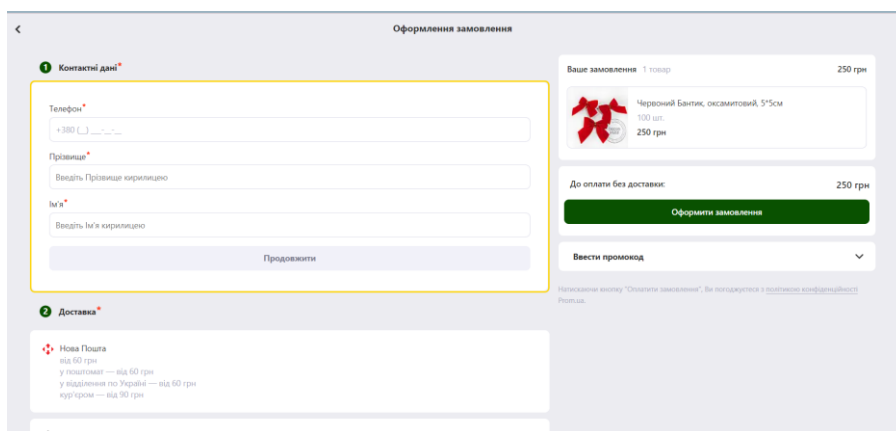


Рисунок 1.14 – Форма для оформлення товару у web-додатку “ DÉCOR PLUS ”

Провівши детальний аналіз аналогів web-додатків, було виділено їх переваги і недоліки.

Загалом, можна сказати, що “ DÉCOR el Home” випереджає своїх конкурентів у деяких позиціях. Результати аналізу представлені в таблиці 1.1.

Таблиця 1.1 – Порівняльна таблиця характеристик аналогів web-додатків

Характеристика	Магазин		
	“Décor el Home”	“JYSK”	“Décor Plus”
	+	+	-
Зручний інтерфейс	+	-	-
Інтерактивність	-	-	-
Функціональність	+	+	+
Навігація	+	+	+
Реєстрація користувачів	+	+	-
Замовлення товарів онлайн без обов'язкової реєстрації	+	+	+
Наявність онлайн зв'язку з тех. підтримкою	+	+	+

Під час аналізу даних таблиці 1.1, можна побачити очевидні переваги та недоліки web-додатків. Також скласти уявлення про близький до ідеалу програмний продукт. Він має мати сучасний, доречний дизайн, зрозумілий інтерфейс, зручну навігацію, інтерактивність web-сторінок.

1.3 Постановка задачі

Мета проекту полягає у розробці багатофункціонального web-додатку підтримки продажу декоративних товарів для інтер'єру з інтуїтивно зрозумілим інтерфейсом, що дозволить компанії автоматизувати частину робочих процесів, що значно покращить її організацію.

Web-додаток має бути доступним у мережі Інтернет. Права доступу мають бути розмежовані за групами користувачів: незареєстрований користувач, зареєстрований користувач та адміністратор.

Адміністратор має необмежений доступ до даних з правами перегляду, додавання, редагування та видалення. Варто зазначити, що створення сторінки для адміністратора є дуже важливим етапом у розробці будь-якого web-додатка, особливо якщо додаток повинен бути постійно наповнюваний новим контентом або даними. Сторінка для адміністратора має ряд переваг: зручність – сторінка для адміністратора дозволяє зробити процес редагування вмісту додатку більш зручним і простим,

ефективність роботи – адміністратор може додавати новий контент, оновлювати існуючий, а також видаляти неактуальний без затримок, зменшення витрат можна заощадити кошти на програмістів та розробників, адже адміністратор може самостійно додавати контент та оновлювати сторінки, що дозволяє зменшити витрати на внесення змін в додаток.

Повернемося до розмежування груп користувачів, а саме до відвідувачів та зареєстрованих клієнтів.

Незареєстрований користувач web-додатку може переглядати інформацію на web-сторінках, обирати та замовляти товари. Зареєстрований користувач може переглядати список раніше виконаних замовлень та редагувати свої облікові дані.

Основними вимогами до розроблюваного програмного продукту є:

- створення модулю реєстрації/авторизації – для розмежування прав доступу до деяких функцій та сторінок web-додатку;
- створення модулю для оформлення замовлення;
- організація зворотнього зв'язку із технічною підтримкою магазину за допомогою онлайн чату;
- створення панелі для адміністратора, де він матиме змогу додавати, змінювати та видаляти дані.

1.4 Вибір засобів реалізації розробки web-додатку

Вибір технічного стеку для розробки web-додатку є важливою частиною успішного завершення проекту. Тому було проведено дослідження інструментів декількох найпопулярніших мов програмування.

Для розробки клієнтської частини була обрана «велика трійця» web-програмування – HTML, CSS, JavaScript.

HTML (мова розмітки гіпертексту) є основою всіх web-додатків та є мінімумом того, що необхідно для їх створення. Найновішою версією є HTML5, вона підтримує крос-платформенну функціональність браузера, що робить його популярним у розробці мобільних додатків.

CSS (каскадні таблиці стилів) надає розмітці кращого вигляду за рахунок трансформування типографіки, кольорів та макетів. Він також є сумісним з усіма браузерами.

JavaScript використовується для додавання анімації, автоматизації завдань на певних сторінках і додавання інтерактивних функцій, які покращують взаємодію з користувачем. Зараз JS є напоширенішою мовою програмування у світі та завдяки Node.js є мовою внутрішнього коду [15].

Для створення бек-енд частини було розглянуто декілька інструментів. Варто зазначити, що «всередині» додаток складається з двох основних компонентів:

- база даних – відповідає за зберігання, упорядкування та обробку даних, щоб їх можна було отримати за запитами сервера;
- серверна мова – відповідає за надсилання, обробку та отримання запитів на дані.

Ці компоненти працюють разом, створюючи основу для кожного web-застосування.

Під час вибору системи керування базами даних (скорочено СУБД) було розглянуто такі варіанти:

- СУБД Oracle – багатомодельна система керування базами даних, здебільшого використовується для обробки онлайн-транзакцій і сховищ даних. Дає можливість простого резервного копіювання та відновлення, єдина база для всіх типів даних, використовує методи шифрування, маскуванню даних, керування ключами та дії моніторингу для захисту бази даних;
- MySQL – безкоштовна реляційна система керування базами даних із відкритим вихідним кодом. Надійний, економічно ефективний інструмент, реалізований для підвищення безпеки та масштабованості бази даних. Його функціональні можливості включають високу швидкість обробки даних і відновлення даних, що збільшує його популярність

– Microsoft SQL Server – ефективна комерційна система керування реляційною базою даних. Забезпечує ефективне керування робочим навантаженням. Вона є безпечною та послідовною, легкою в налаштуванні та користуванні [16].

Було надано перевагу СУБД MySQL. Причинами цьому стали її надійність, підвищений рівень безпеки та здатність швидко обробляти дані.

Далі потрібно було обрати серверну мову. Серед розглянутих варіантів були:

– PHP – потужний інструмент для створення інтерактивних і динамічних веб-сторінок, він є безкоштовним, широко використовуваним та має C-подібний синтаксис, легко взаємодіє з сервером MySQL. Основні клієнти : Facebook, WhatsApp, Tesla Motors.

– Python – мова програмування високого рівня, яка є універсальною, інтерактивною та об'єктно-орієнтованою, сумісна з платформою Linux. Основні клієнти: Google, Pinterest, Netflix, Instagram, Spotify.

– JS – відкрита, сумісна з різними платформами мова, яку легко реалізувати, оскільки вона інтегрується з HTML. Основні клієнти :Reddit, Tinder, Alibaba [17].

Серед представлених мов програмування було обрано PHP. Вона є широковикористовуваною, досить швидкою, має велику кількість підтримки PHP для спільноти, легко поєднується з HTML та MySQL, її можна використовувати для створення різних за складністю веб-додатків.

2 Моделювання та проектування

2.1 Моделювання інформаційної системи

Для моделювання web-додатку було використано IDEF0 – методологію функціонального моделювання. На рис. 2.1 представлено контекстну діаграму, що моделюється в одному блоці та показує основні входи, елементи керування, виходи та механізми, пов'язані з системою.

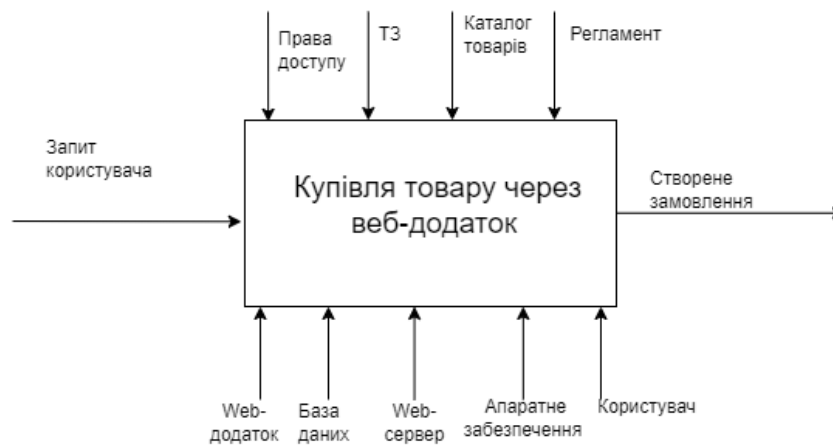


Рисунок 2.1 – Контекстна діаграма в нотації IDEF0

На рис. 2.2 показано ієрархічну декомпозицію, на якій деталізовано представлено розбиття складного процесу купівлі товару на складові його функції.

У декомпозиції моделі головної функції веб-сайту визначено п'ять блоків:

- перехід на сайт;
- навігація по сайту;
- вибір товару;
- оформлення замовлення.

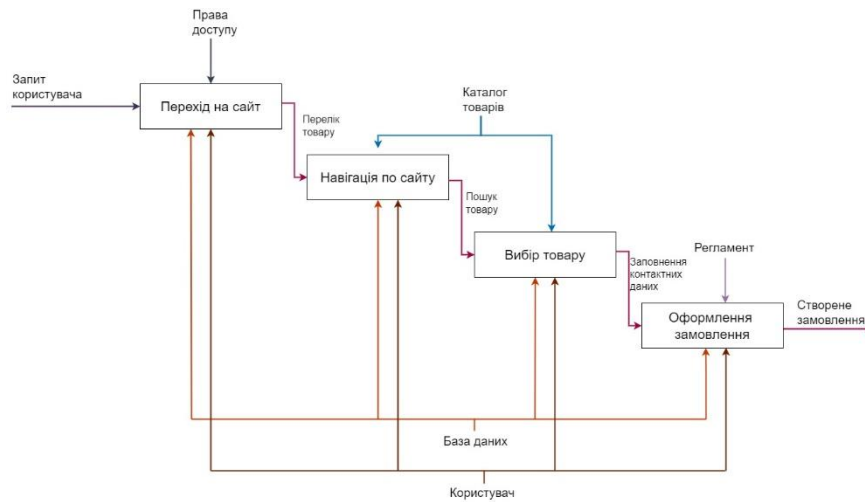


Рисунок 2.2 – Декомпозиція контекстної діаграми

Блок «Перехід на сайт»:

- Вхідна стрілка «Запит користувача»;
- Вихідна стрілка «Перелік товару»;
- Стрілка контролю «Права доступу»;
- Стрілки механізму: «База даних», «Користувач».

Блок «Навігація по сайту»:

- Вхідна стрілка «Перелік товару»;
- Вихідною стрілкою є «Пошук товару»;
- Стрілка контролю «Каталог товарів»;
- Стрілки механізму: «База даних», «Користувач».

Блок «Вибір товару»:

- Вхідна стрілка «Пошук товару»;
- Вихідна стрілка «Заповнення контактних даних»;
- Стрілка контролю «Каталог товарів»;
- Стрілки механізму: «База даних», «Користувач».

Блок «Оформлення замовлення»:

- Вхідна стрілка «Заповнення контактних даних»;
- Вихідна стрілка «Створене замовлення»;
- Стрілка контролю «Регламент»;
- Стрілки механізму: «База даних», «Користувач».

2.2 Діаграма варіантів використання

На рис. 2.3 представлено діаграму варіантів використання, на якій зображено безпосередньо варіанти використання, актори та відношення між ними.

Актори:

- адміністратор – особа, яка має доступ до адміністративної частини web-додатку, може коригувати бази даних, обробляти замовлення.
- користувач – особа, яка використовує web-додаток, ознайомлюючись з переліком товарів, маючи змогу оформити замовлення, переглянути замовлення та оновити персональні дані.

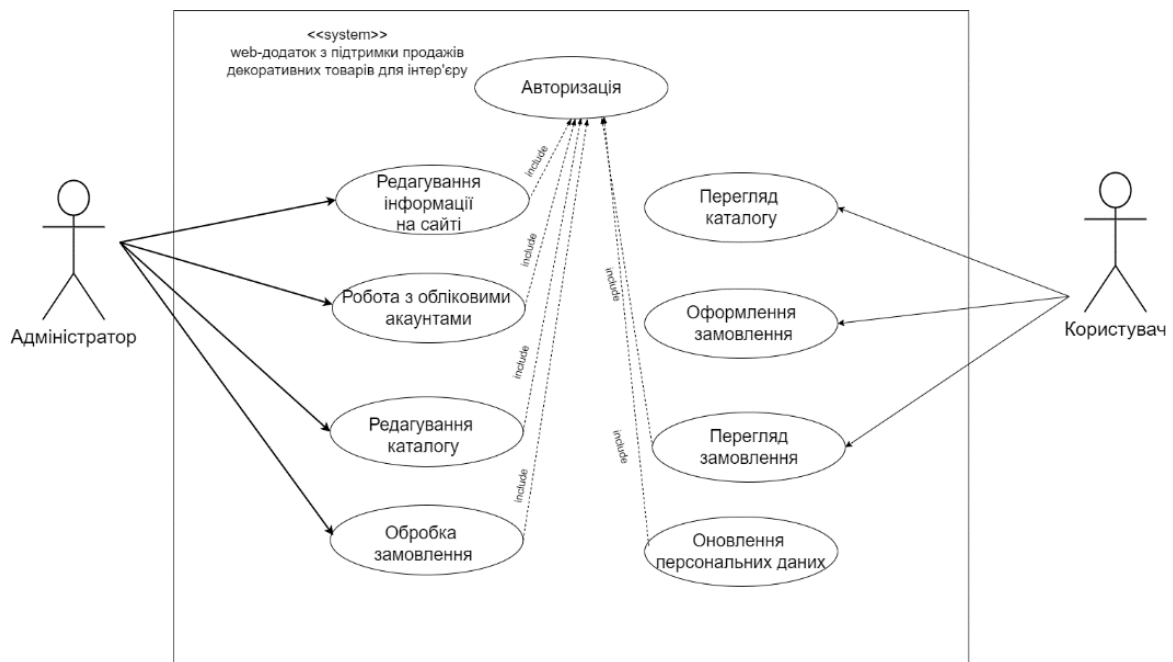


Рисунок 2.3 – Діаграма варіантів використання

Варіанти використання:

- ВВ Редагування інформації – надає дозвіл адміністратору редагувати інформацію у web-додатку;
- ВВ Робота з обліковими аккаунтами – надає дозвіл адміністратору переглядати контактні дані зареєстрованих користувачів;
- ВВ Редагування каталогу – надає дозвіл адміністратору редагувати каталог;
- ВВ Обробка замовлень – надає дозвіл адміністратору обробляти замовлення;
- ВВ Перегляд каталогу – надає дозвіл користувачу переглядати каталог;
- ВВ Оформлення замовлення – надає дозвіл користувачу оформляти замовлення;
- ВВ Оновлення персональних даних – надає дозвіл користувачу оновлювати свої дані;
- ВВ Перегляд замовлення – надає дозвіл користувачу переглядати свої замовлення

2.3 Проектування моделі бази даних

Етап проектування бази даних включає в себе розробку загального стратегічного плану, який має на меті забезпечити ефективну реалізацію етапів життєвого циклу бази даних. На цьому етапі необхідно розглянути ряд ключових питань, що включають:

1. Аналіз існуючих інформаційних систем: Дослідження та оцінка наявних інформаційних систем, включаючи їхні функціональні можливості, недоліки та потенційні проблеми.
2. Доцільність створення нової інформаційної системи: визначення необхідності створення нової системи на основі аналізу існуючих недоліків та вимог користувачів. Враховуючи економічні, технічні та організаційні фактори.
3. Обсяг робіт і ресурсів, вартість проекту: оцінка необхідного обсягу робіт, ресурсів (включаючи фінансові, людські та матеріальні ресурси) і вартості проекту.

4. Розробка методології збору даних, визначення їх формату: розробка плану збору даних, включаючи методи та інструменти для збору, організації та зберігання даних. Визначення формату даних, що найкраще відповідає потребам інформаційної системи.

5. Визначення послідовності проектування і реалізації застосувань: встановлення послідовності етапів проектування та реалізації бази даних, включаючи вибір відповідних технологій, розробку моделей та інтерфейсів.

На основі зібраної інформації було створено ERD-діаграму реляційної бази даних (рис. 2.4) та детальний опис таблиць (табл. 2.1) інформаційної системи, а також відповідних полів даних (табл.2.2) які містяться в таблицях, що відобразатимуть структуру та взаємозв'язки даних в системі.

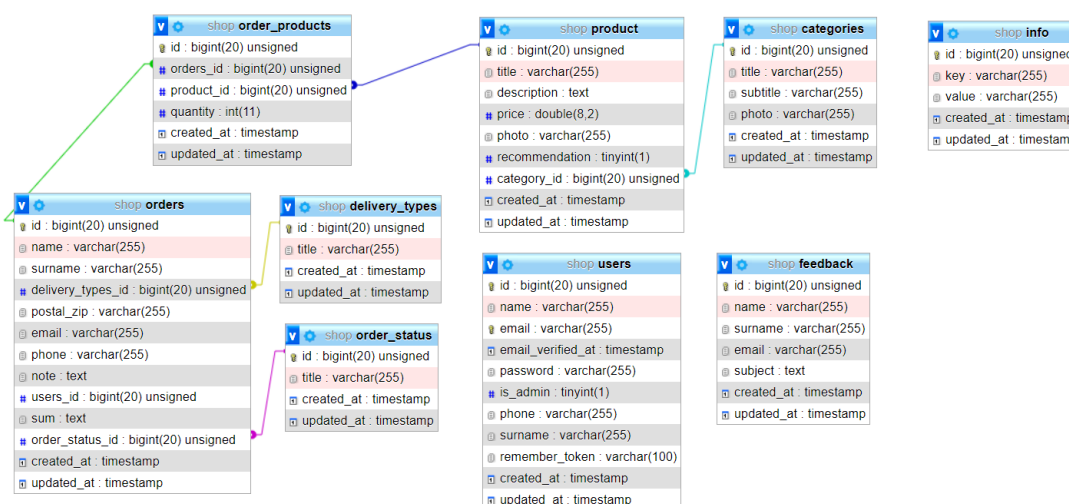


Рисунок 2.4 – Фізична модель бази даних

Таблиця 2.1 – Опис таблиць бази даних

№	Назва таблиці	Призначення
1	categories	Категорії товарів
2	delivery_types	Види доставки замовлень
3	Feedback	Запитання та відгуки користувачів

Продовження таблиці 2.1

4	Info	Загальна інформація web-додатку (контакти, опис, доставка)
5	Orders	Інформація про замовлення
6	order_products	Об'єднує товар який був доданий до замовлення і контактні дані користувача.
7	order_status	Статуси замовлень.
8	Product	Зберігає всю інформацію про товари.
9	Users	Таблиця з зареєстрованими користувачами додатку.

Таблиця 2.2 – Опис полів бази даних

Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
categories	id	Ідентифікатор категорії	INTEGER	PK	Не пустий
	title	Назва	VARCHAR(100)		Не пустий
	subtitle	Короткий опис	TEXT		Не пустий
	photo	Зображення	VARCHAR(100)		Не пустий
delivery_types	id	Ідентифікатор категорії	INTEGER	PK	Не пустий
	title	Назва	VARCHAR(100)		Не пустий
feedback	Name	Ім'я користувача	VARCHAR(100)		Не пустий
	surname	Прізвище користувача	VARCHAR(100)		Не пустий
	Email	Email користувача	VARCHAR(100)		Не пустий
	Subject	Текст повідомлення	TEXT		Не пустий
	Created_at	Дата створення	DATE		Не пустий
Info	Key	Ключ запису	VARCHAR(100)	PK	Не пустий
	Value	Значення запису	TEXT		Не пустий
Orders	id	Ідентифікатор категорії	INTEGER	PK	Не пустий

Продовження таблиці 2.2

Name	Ім'я користувача	VARCHAR(100)		Не пустий
surname	Прізвище користувача	VARCHAR(100)		Не пустий
Email	Email користувача	VARCHAR(100)		Не пустий
delivery_types _id	Тип доставки	INTEGER	FK	Не пустий
postal_zip	Відділення пошти	INTEGER		
phone	Телефон користувача	VARCHAR(100)		Не пустий
Note	Коментар	TEXT		
users_id	Ідентифікатор користувача	INTEGER	FK	Не пустий
Sum	Загальна сума	DOUBLE		Не пустий
order_status_id	Ідентифікатор статусу замовлення	INTEGER	FK	Не пустий
created_at	Дата створення	DATE		Не пустий

Продовження таблиці 2.2

order_products	orders_id	Ідентифікатор замовлення	INTEGER	FK	Не пустий
	product_id	Ідентифікатор товару	INTEGER	FK	Не пустий
	Quantity	Кількість товару	INTEGER		Не пустий
order_status	id	Ідентифікатор статусу	INTEGER	PK	Не пустий
	title	Назва	VARCHAR(100)		Не пустий
product	id	Ідентифікатор товару	INTEGER	PK	Не пустий
	title	Назва товару	VARCHAR(100)		Не пустий
	Description	Опис товару	TEXT		Не пустий
	Price	Ціна товару	DOUBLE		Не пустий
	photo	Фото товару	VARCHAR(100)		Не пустий
	Recommendation	Рекомендація товару	BOOLEAN		Не пустий
	category_id	Ідентифікатор категорії	INTEGER	FK	Не пустий
	created_at	Дата створення	DATE		Не пустий

Продовження таблиці 2.2

Users	d	Ідентифікатор користувача	INTEGER	PK	Не пустий
	Name	Ім'я користувача	VARCHAR(100)		Не пустий
	surname	Прізвище користувача	VARCHAR(100)		Не пустий
	Email	Email користувача	VARCHAR(100)		Не пустий
	phone	Телефон користувача	VARCHAR(100)		Не пустий
	Password	Пароль користувача	VARCHAR(100)		Не пустий
	Is_admin	Приналежність до адміністраторів	BOOLEAN		Не пустий

3 Програмна реалізація

3.1 Архітектура web додатку

Web-додаток складається з двох основних частин: адміністративної та користувацької. Адміністративна частина додатку забезпечує адміністраторам можливість додавати, редагувати та видаляти різноманітний вміст, такий як статичні та динамічні сторінки, розділи та блоки додатку. Це дозволяє змінювати та оновлювати вміст в зручний спосіб, забезпечуючи його актуальність.

Користувацький інтерфейс створений таким чином, щоб бути зрозумілим і легким у використанні для відвідувачів. Він інтуїтивно представлятиме структуру сайту та надаватиме можливість швидко та логічно переходити до розділів та сторінок. Таким чином користувачі зможуть легко знайти необхідну інформацію та користуватись додатком без зайвих зусиль.

Інформаційна система підтримки продажу декоративних товарів буде відповідати ряду вимог, які можна розділити на три основні категорії: візуальний вміст, інформаційне наповнення та структура. Візуальний вміст передбачає естетичний та привабливий дизайн web-додатку, що стимулює інтерес відвідувачів та сприяє позитивному сприйняттю товарів. Інформаційне наповнення включатиме якісний та інформативний матеріал про декоративні товари, їх характеристики, особливості та способи використання. Структура додатку повинна бути логічною та зручною для навігації, дозволяючи відвідувачам швидко знайти потрібну інформацію та здійснювати покупки з легкістю.

Детальніша інформація про вимоги до web-додатку підтримки продажу декоративних товарів наведена в таблиці 3.1. Вона містить докладний опис кожної вимоги, що відноситься до візуального наповнення, контенту та структури додатку, детально розкриваючи їх характеристики та важливість.

Таблиця 3.1 – Вимоги до web-додатку

№	Категорія	Назва	Опис
1	Структура	Верхній колонтитул	Кожна сторінка має містити колонтитул з навігаційним меню та логотипом додатку у верхній частині.
2		Меню навігації	Навігаційне меню містить список гіперпосилань на основні сторінки інформаційної системи.
3		Основний вміст	На сторінці розміщений блок, який займає більшу частину сторінки та містить інформаційне наповнення, відповідне тематиці.
4		Нижній колонтитул	В нижній частині сторінки розташований блок, призначений для виведення корисної, але не першорядної інформації. Тут можуть бути такі елементи, як посилання на соціальні мережі, розділи для зворотного зв'язку, дублювання основних пунктів навігації та інші.
5	Візуальний вміст	Висока читабельність шрифтів	Для забезпечення зручності використання інформаційної системи рекомендується використовувати шрифти, які легко читати. Використання непридатних для читання шрифтів може ускладнити процес перегляду.

Продовження таблиці 3.1

№	Категорія	Назва	Опис
6		Колірна гама	При виборі палітри кольорів важливо впевнитися, що вона має високий рівень доступності та легко сприймається всіма користувачами.
7		Адаптивний дизайн	При реалізації адаптивної верстки додатку будь-який користувач зможе зручно використовувати інформаційну систему як на телефоні, так і на інших пристроях.
8		Єдиноформатний стиль	Для зручного використання важливо дотримуватися єдиного стилю оформлення для блоків з інформацією, картинок, іконок та інших елементів.
9	Інформаційне наповнення	Ефективне використання	Текст повинен бути привабливим, його можна підкріплювати ілюстраціями для більшої ефективності.
10		Типографія	Варто звернути увагу на оформлення тексту, оскільки воно має великий вплив на сприйняття інформації звичайними користувачами.

Web-додаток підтримки продажу декоративних товарів можна логічно розділити на дві основні частини: перед авторизацією та після авторизації. Цей розподіл визначає доступ до різних функціональних модулів та сторінок в залежності від статусу користувача.

Перед авторизацією web-додаток надає доступ до загальної інформації та функцій, які не вимагають особистого облікового запису. На цьому етапі користувач може ознайомитися з основною інформацією про продукти, а також переглянути загальні категорії товарів. Однак, доступ до розширених функцій, таких як перегляд історії покупок, зміна особистої інформації, обмежений до авторизованих користувачів.

Після авторизації користувач отримує розширені можливості та привілеї. Він може увійти до особистого облікового запису, де отримує доступ до персоналізованої інформації, такої як історія замовлень, налаштування профілю тощо. Крім того, авторизованим користувачам надається можливість здійснювати замовлення товарів без повторного введення контактних даних.

Цей розподіл дозволяє забезпечити відповідний рівень функціональності та доступу до інформації в залежності від потреб та прав користувачів, створюючи зручне та персоналізоване середовище для продажу декоративних товарів.

Перелік функціональних можливостей для незареєстрованого користувача:

- реєстрація, або авторизація;
- пошук товару у web-додатку (за назвою, категорією);
- можливість оформлення онлайн-замовлень за допомогою форми;
- наявність повної інформації про товари;
- фільтрація товару за назвою та категорією;
- додання товару до обраного;
- форма зворотного зв'язку.

Перелік функціональних можливостей для зареєстрованого користувача:

- історія замовлень;
- пошук товару у web-додатку (за назвою, категорією);
- можливість оформлення онлайн-замовлень за допомогою форми;
- наявність повної інформації про товари;
- фільтрація товару за назвою та категорією;
- додання товару до обраного;
- форма зворотного зв'язку.

Перелік функціональних можливостей для адміністратора:

- додання, редагування, видалення товару;
- додання, редагування, видалення категорій товару;
- оновлення загальної інформації про web-додаток, доставку та контактні дані;
- перегляд зареєстрованих користувачів;
- перегляд записів форми «Зворотного зв'язку»;
- Перегляд статистики продажів;
- крім того, має доступ до всіх функцій звичайного користувача.

Розглянемо детальніше, які модулі доступні для кожного типу акаунту в таблиці 3.2.

Таблиця 3.2 – Таблиця рівнів доступу до сторінок інформаційної системи

№	Сторінки	Гість	Користувач	Адміністратор
1	«Замовлення товару»	+	+	+
2	«Історія замовлень»	–	+	+
3	«Перегляд товару»	+	+	+
4	«Пошук»	+	+	+
5	«Додання до обраного»	+	+	+
6	«Кошик»	+	+	+
7	«Оновлення інформації у web-додатку»	–	–	+
8	«Зворотній зв'язок»	+	+	+
9	«Сторінка редагування особистої інформації»	–	+	+

Розроблена інформаційна система підтримки продажу декоративних товарів забезпечує продуктивну взаємодію між користувачами різних типів.

3.2 Програмна реалізація

Програмна реалізація інформаційної системи підтримки продажу декоративних товарів виконана за допомогою PHP, а саме фреймворку Laravel.

Створення проекту на фреймворку Laravel може бути розділено на кілька етапів. Нижче наведено опис основних кроків розробки web-додатку.

1. Встановлення Laravel:

Початковий крок – встановлення фреймворку Laravel на локальному або віддаленому сервері. Для цього можна використовувати Composer, виконавши команду `composer global require laravel/installer`. Після успішної установки можна створити новий проект, використовуючи команду `laravel new mondo`.

2. Створення маршрутів:

Laravel використовує маршрути для визначення, який код виконувати при запитах до вашого додатку. Маршрути знаходяться в файлі `routes/web.php` та `routes/api.php`. Можна визначати маршрути за допомогою методів, таких як `get`, `post`, `put`, `delete` та інших. Наприклад, для створення маршруту GET, який відображає web-сторінку, Використовуємо наступний код:

```
Route::get('/', [HomeController::class, 'home'])->name('home');
```

3. Створення контролерів:

Контролери в Laravel використовуються для групування логіки, пов'язаної з обробкою запитів. Створити контролер можна за допомогою команди `php artisan make:controller HomeController`. У контролері описуємо методи, які відповідають за обробку різних запитів. Наприклад:

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class ProductController extends Controller{
    public function index() {
        // Код для відображення списку користувачів
    }
    public function store(Request $request){
        // Код для збереження нового користувача
    }
}
```

```

    }

    public function show($id){
        // Код для відображення конкретного користувача
    }
    public function update(Request $request, $id){
        // Код для оновлення існуючого користувача
    }
    public function destroy($id){
        // Код для видалення користувача
    }
}

```

4. Робота з міграціями та моделями:

Laravel має вбудовану підтримку для роботи з базами даних. Для цього використовується міграції для створення та зміни структури бази даних. Міграції знаходяться в директорії `database/migrations`. Міграції створювались за допомогою команди `php artisan make:migration create_table_products`. Крім того, створені моделі, які відображають таблиці бази даних. Моделі знаходяться в директорії `app/Models`.

Наприклад:

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class User extends Model{
    protected $fillable = ['name', 'surname', 'email', 'password',
'is_admin'];
}

```

5. Використання шаблонів та видів:

Laravel використовує шаблони та види для відображення сторінок вашого додатку. Шаблони визначають загальну структуру сторінки, тоді як види відповідають за вміст конкретних сторінок. Види знаходяться в директорії `resources/views`. Всі шаблони створювались з розширенням `.blade.php`. Наприклад:

```

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

```

```
<head>
  <title>Про нас</title>
  @include('inc.header')
</head>
<body>
  <div class="site-wrap">
    @include('inc.head', [
      'categories' => $categories
    ])
    <div class="bg-light py-3">
      <div class="container">
        {{$data['about']}}
      </div>
    </div>
    @include('inc.footer', [
      'categories' => $categories
    ])
  </div>
  @include('inc.scripts')
</body>
</html>
```

Розглянемо структуру додатку на рисунку 3.1 та детальний опис кожного із каталогів в таблиці 3.3


```

Mode                               LastWriteTime           Length Name
----                               -
d----- 16.05.2023 23:09          app
d----- 16.05.2023 23:09        bootstrap
d----- 16.05.2023 23:09        config
d----- 16.05.2023 23:09        database
d----- 16.05.2023 23:09        lang
d----- 18.05.2023 0:27        node_modules
d----- 22.05.2023 23:14        public
d----- 17.05.2023 23:18        resources
d----- 16.05.2023 23:09        routes
d----- 16.05.2023 23:09        storage
d----- 16.05.2023 23:09        tests
d----- 22.05.2023 17:55        vendor
-a----- 02.09.2022 7:10           258 .editorconfig
-a----- 22.05.2023 19:11          1119 .env
-a----- 02.09.2022 7:10          1069 .env.example
-a----- 02.09.2022 7:10           179 .gitattributes
-a----- 21.05.2023 16:17          219 .gitignore
-a----- 15.09.2022 2:26           89 .htaccess
-a----- 02.09.2022 7:10          1686 artisan
-a----- 17.05.2023 23:11          1881 composer.json
-a----- 17.05.2023 23:11         297714 composer.lock
-a----- 22.05.2023 22:27         9498934 modno.zip
-a----- 18.05.2023 0:27          180621 package-lock.json
-a----- 18.05.2023 0:34           380 package.json
-a----- 02.09.2022 7:10          1175 phpunit.xml
-a----- 02.09.2022 7:10          3977 README.md
-a----- 18.05.2023 0:34           312 vite.config.js

```

Рисунок 3.1 – Структура проекту

Таблиця 3.3 – Опис структури проекту

№	Назва	Опис
1	«app»	Код ядра додатку на Laravel.
2	«bootstrap»	Файли, що використовуються для розширення можливостей Vue.js за допомогою VueBootstrap.
3	«config»	Файли налаштування додатку
4	«database»	Міграційні файли моделей бази даних та класи для наповнення бази даних інформацією
5	«public»	Центральна точка доступу для всіх запитів, яка включає всі ресурси.
6	«resources»	Некомпільовані ресурси, такі як SASS, LESS і JavaScript
7	«routes»	Маршрути додатку, які налаштовані для визначення шляхів та обробки запитів, що надходять до додатку.
8	«storage»	Містить скомпільовані шаблони, файли кешу, файли сесії та інші файли, що створені в процесі роботи з Laravel.

Отже, результатом є підготовлена структура проекту із всіма потрібними шаблонами, файлами та налаштуваннями.

3.3 Використання програмного додатку

Web-додаток підтримки продажу декоративних товарів для інтер'єру складається з двох основних частин: клієнтської та адміністративної.

Клієнтська частина додатку призначена для перегляду інформації та доступна всім користувачам. Тут відвідувачі можуть ознайомитися з асортиментом декоративних товарів, переглянути їх характеристики, зображення та ціни. Крім того, користувачі можуть здійснювати пошук товарів за різними критеріями, додавати їх до кошика та здійснювати покупки.

Адміністративна панель є іншою частиною web-додатку та доступна тільки для адміністраторів. Вона надає спеціальні функції та можливості для додавання та редагування інформації, що відображається у додатку. Адміністратори можуть керувати асортиментом товарів, оновлювати їх характеристики, завантажувати зображення, встановлювати ціни та проводити інші необхідні зміни. Крім того, вони мають можливість керувати замовленнями та відстежувати стан виконання замовлень.

Такий розподіл на клієнтську та адміністративну частини дозволяє забезпечити відокремлену функціональність для користувачів та адміністраторів, сприяючи зручному та ефективному управлінню декоративними товарами та забезпечуючи задоволення потреб усіх сторін.

3.3.1 Клієнтська частина web-додатку

При завантаженні додатку у браузері з'являється головна сторінка (рис. 3.6). У верхній частині розміщено шапку, де розташоване навігаційне меню, що містить посилання на різні розділи та сторінки web-додатку. Також у шапці знаходяться посилання на сторінку авторизації, де користувачі можуть увійти до своїх облікових записів, перейти до обраного, а також на кошик, який відображає кількість товарів, що додані для покупки.

Поміж навігаційним меню та пошуковою формою, також розміщено фірмову назву web-додатку, що дозволяє користувачам легко впізнати його та відразу ж

зорієнтуватися в контексті. Форма пошуку дозволяє швидко знайти потрібний товар, вводячи назву товару або вибираючи конкретну категорію.

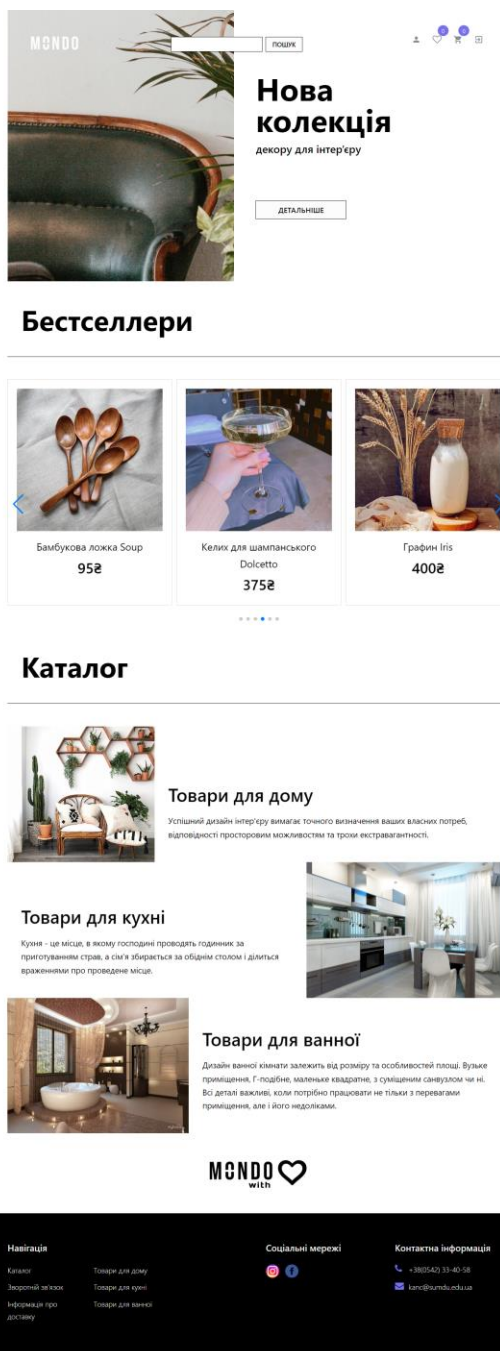


Рисунок 3.2 – Головна сторінка web-додатку

Після натискання на кнопку «Детальніше», або введення назви бажаного товару в форму пошуку відкриється сторінка каталогу (рис. 3.3).

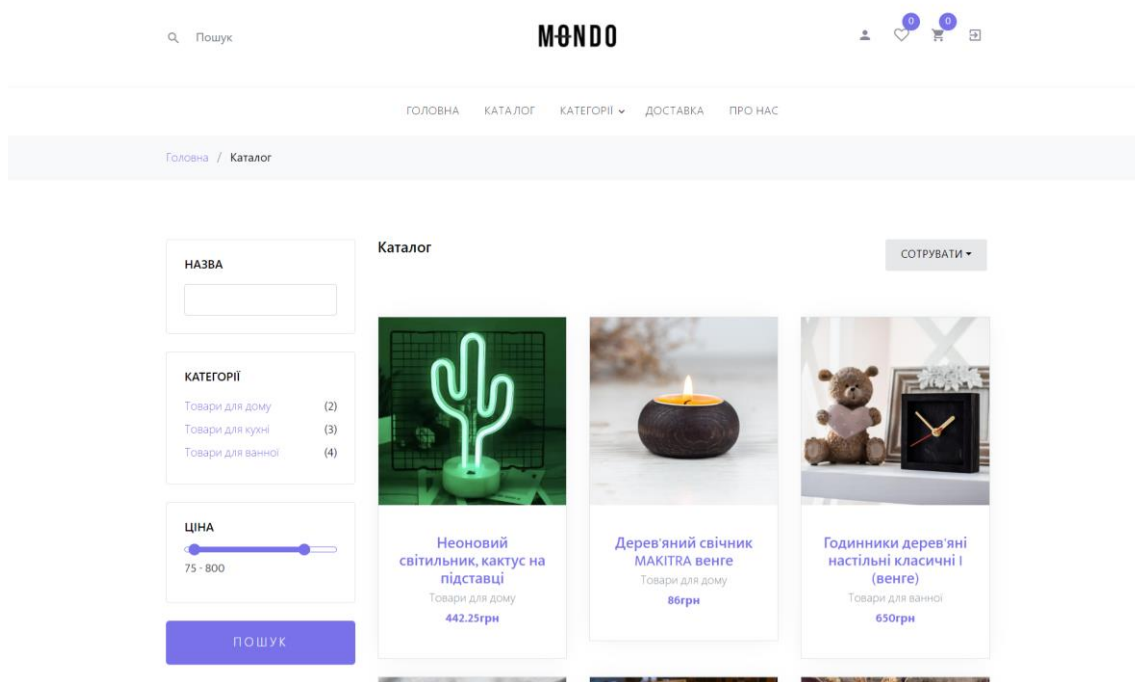


Рисунок 3.3 – Сторінка каталогу

Справа на сторінці розташована навігаційна панель, яка включає в себе декілька корисних елементів. Перш за все, тут розміщена форма пошуку, де користувачі можуть вводити ключові слова для знаходження конкретного товару. Крім того, навігаційна панель містить фільтр для товарів за ціною. В полі ціни користувачі можуть вказати діапазон цін, обмежуючи пошук від найнижчої до найвищої ціни. Це дозволяє користувачам знайти товари, які відповідають їх бюджету та вимогам.

Також сторінка має можливість сортування товарів за назвою та ціною. Користувачі можуть вибрати відповідний спосіб сортування, щоб впорядкувати товари в алфавітному порядку або в порядку зростання або спадання ціни.

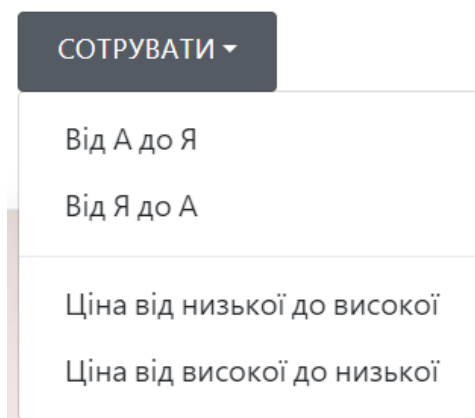
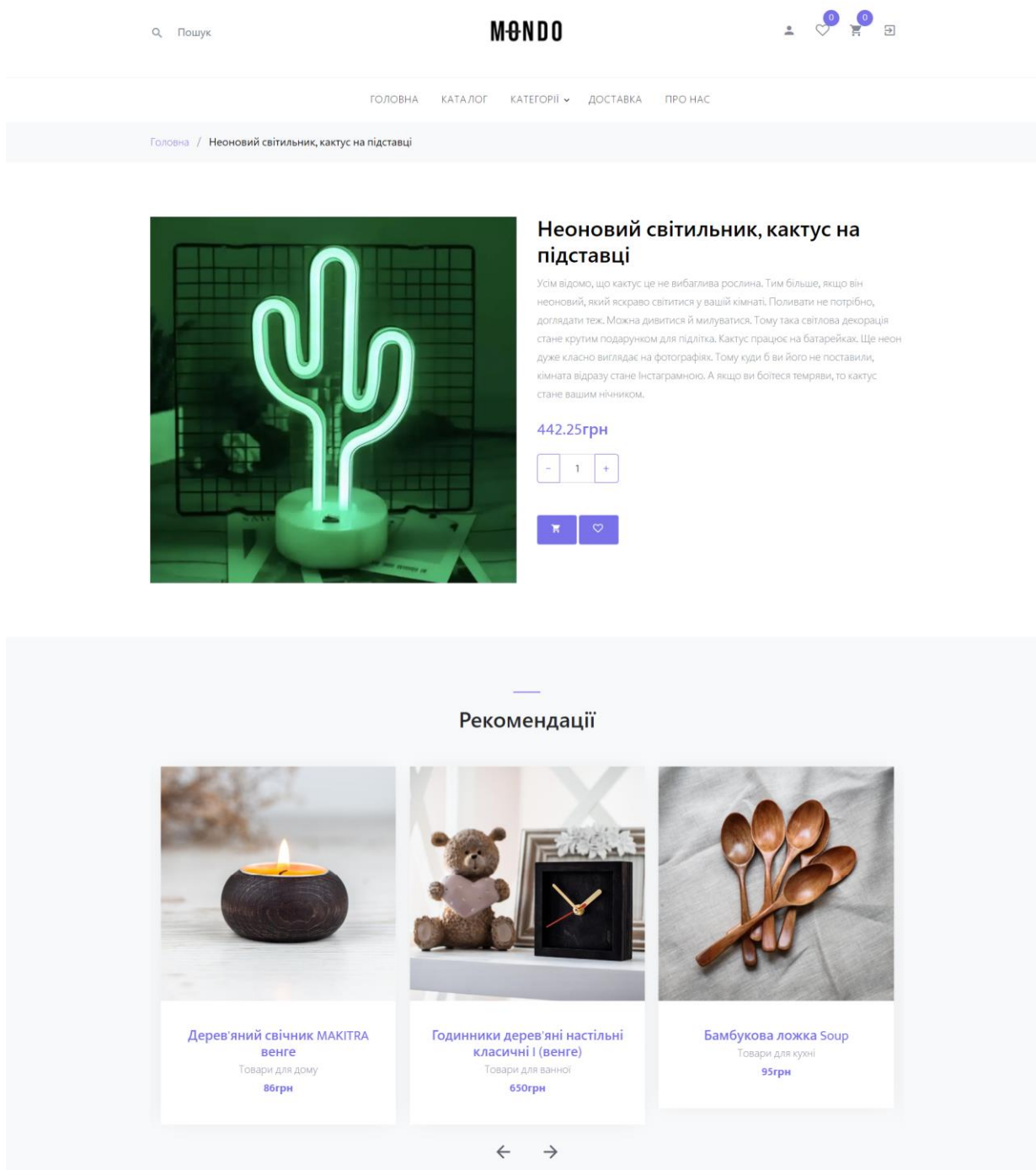


Рисунок 3.4 – Сторінка каталогу

При натисканні на картку товару відбувається перехід на сторінку з детальною інформацією про товар (рис. 3.5). На цій сторінці користувач може вказати кількість товару, який він бажає замовити, а також додати товар до списку обраних. При натисканні кнопки "В кошик" товар з вказаними розміром, кольором і кількістю автоматично додається до кошика.



Пошук

MENDO

ГОЛОВНА КАТАЛОГ КАТЕГОРІЇ ДОСТАВКА ПРО НАС

Головна / Неоновий світильник, кактус на підставці

Неоновий світильник, кактус на підставці

Усім відомо, що кактус це не вибаглива рослина. Тим більше, якщо він неоновий, який яскраво світитися у вашій кімнаті. Поливати не потрібно, доглядати теж. Можна дивитися й милуватися. Тому така світлова декорація стане крутим подарунком для підлітка. Кактус працює на батарейках. Ще неон дуже класно виглядає на фотографіях. Тому куди б ви його не поставили, кімната відразу стане Інстаграмною. А якщо ви боїтеся темряви, то кактус стане вашим нічним.

442.25грн

- 1 +

🛒 ❤️

Рекомендації

Дерев'яний свічник МАКІТРА венге
Товари для дому
86грн

Годинники дерев'яні настільні класичні I (венге)
Товари для ванної
650грн

Бамбукова ложка Soup
Товари для кухні
95грн

← →

Рисунок 3.5 – Сторінка з товаром

Нижче блока з інформацією про товар розташований слайдер з картками рекомендованих товарів.

Для додавання товару до кошика необхідно натиснути на кнопку з зображенням кошика, яка розташована поруч з кнопкою "Додати до обраного". На сторінці кошика користувач знайде список обраних товарів, які можна видалити або змінити кількість. Нижче списку товарів розташований блок зі загальною ціною замовлення та кнопка для оформлення замовлення.

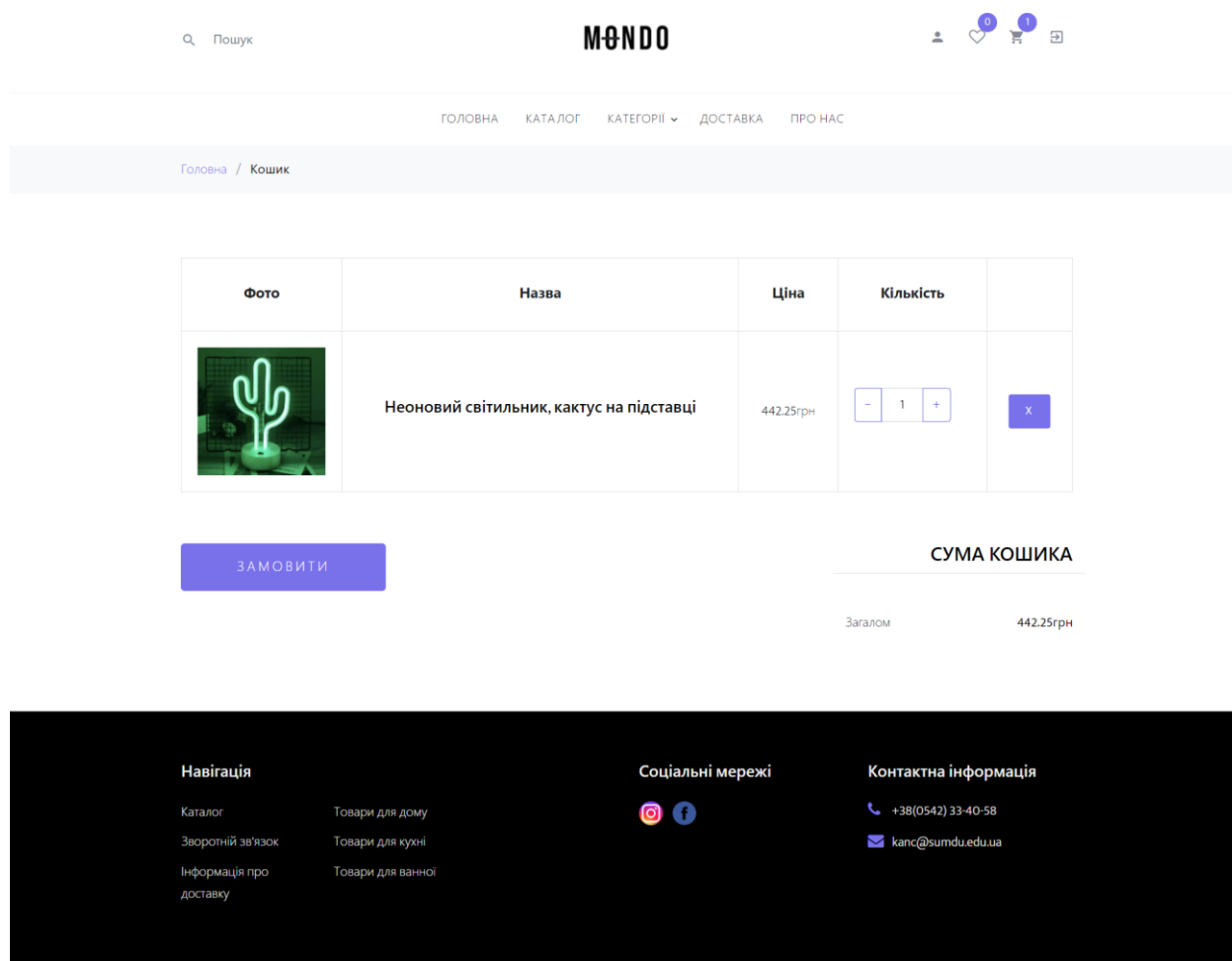


Рисунок 3.6 – Сторінка кошику

Сторінка оформлення замовлення включає форму для введення контактних даних замовника, вибору способу доставки і можливості додати коментар (за бажанням). На сторінці також відображена загальна сума замовлення та кількість обраних товарів.

Після заповнення всіх необхідних полів користувач натискає кнопку "Оформити замовлення". Після цього з'являється повідомлення про успішне замовлення і детальна інформація про замовлення (рис 3.8).

Пошук

MENDO

ГОЛОВНА КАТАЛОГ КАТЕГОРІЇ ДОСТАВКА ПРО НАС

Home / Кошик / Замовлення

Контактні дані

Ім'я * Прізвище *

Спосіб доставки * Відділення *

Email * Телефон *

Коментар

Замовлення

Товар	Ціна
Неоновий світильник, кактус на підставці	442.25грн
Всього	442.25

ОФОРМИТИ ЗАМОВЛЕННЯ

Навігація

- Каталог
- Зворотній зв'язок
- Інформація про доставку
- Товари для дому
- Товари для кухні
- Товари для ванної

Соціальні мережі

Контактна інформація

- +38(0542) 33-40-58
- kanc@sumdu.edu.ua

Рисунок 3.7 – Сторінка з підтвердженням замовленням

Пошук

MENDO

ГОЛОВНА КАТАЛОГ КАТЕГОРІЇ ДОСТАВКА ПРО НАС

✓

Дякуємо за замовлення!

Ваше замовлення успішно виконано.

ПРОДОВЖИТИ ПОКУПКИ

Рисунок 3.8 – Сторінка з підтвердженим замовленням

Сторінка «Про нас» містить детальний опис інтернет-магазину, а також його переваги серед конкурентів.

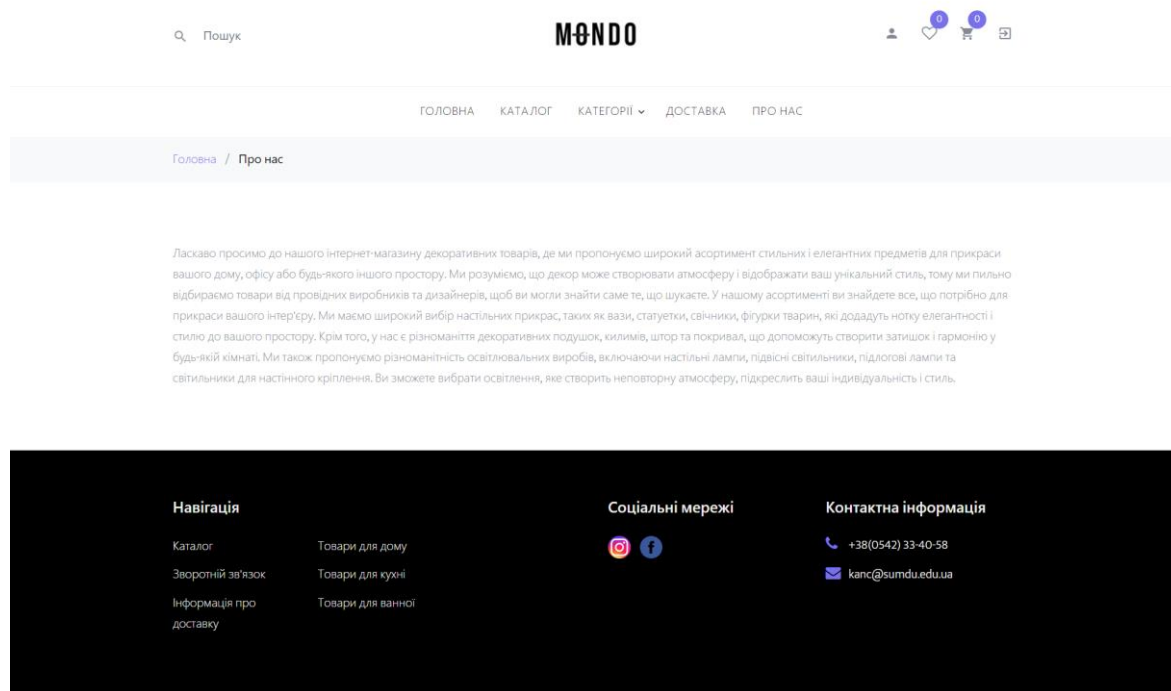


Рисунок 3.9 – Сторінка «Про нас»

На сторінці «Доставка и оплата» знаходиться інформація про умови доставки і оплати замовленого товару.

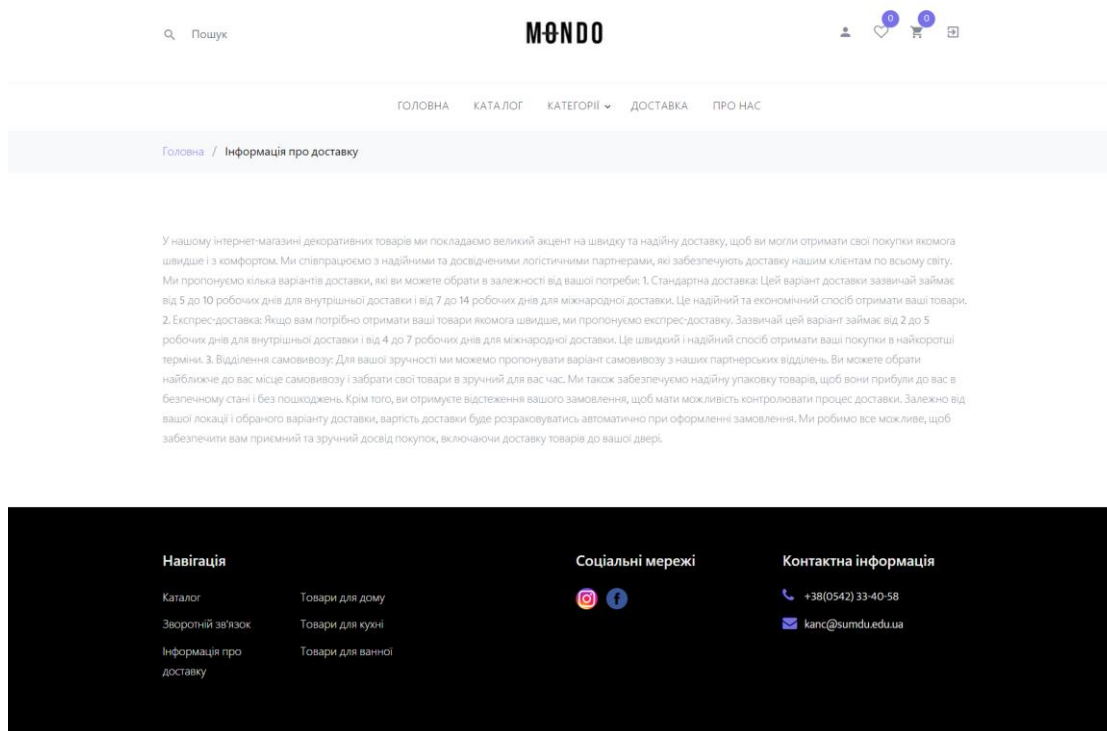


Рисунок 3.10 – Сторінка «Інформація про доставку»

Користувачам web-додатку надається можливість звернутися до адміністраторів у разі виникнення питань. Для цього була створена окрема сторінка під назвою "Зворотній зв'язок". На цій сторінці користувачі можуть надіслати електронного листа або скористатися формою зворотного зв'язку для зв'язку з адміністраторами.

Пошук

MENDO

ГОЛОВНА КАТАЛОГ КАТЕГОРІЇ ДОСТАВКА ПРО НАС

Головна / Зворотній зв'язок

Зворотній зв'язок

Ім'я*

Прізвище*

Email*

Повідомлення

ВІДПРАВИТИ

Рисунок 3.11 – Сторінка «Зворотній зв'язок»

3.3.2 Адміністративна панель

Web-додаток має вбудовану адміністративну панель, яка надає можливість управляти додатком, додавати нові товари та обробляти замовлення. Для доступу до адміністративної панелі необхідно спочатку пройти процес авторизації. Для цього користувач повинен ввести свою електронну адресу та пароль, які використовуються для ідентифікації (рис. 3.18). Після успішної авторизації користувач отримує доступ до всіх функцій адміністративної панелі, що дозволяє йому здійснювати редагування вмісту web-додатку, керувати товарним асортиментом і ефективно виконувати обробку замовлень.

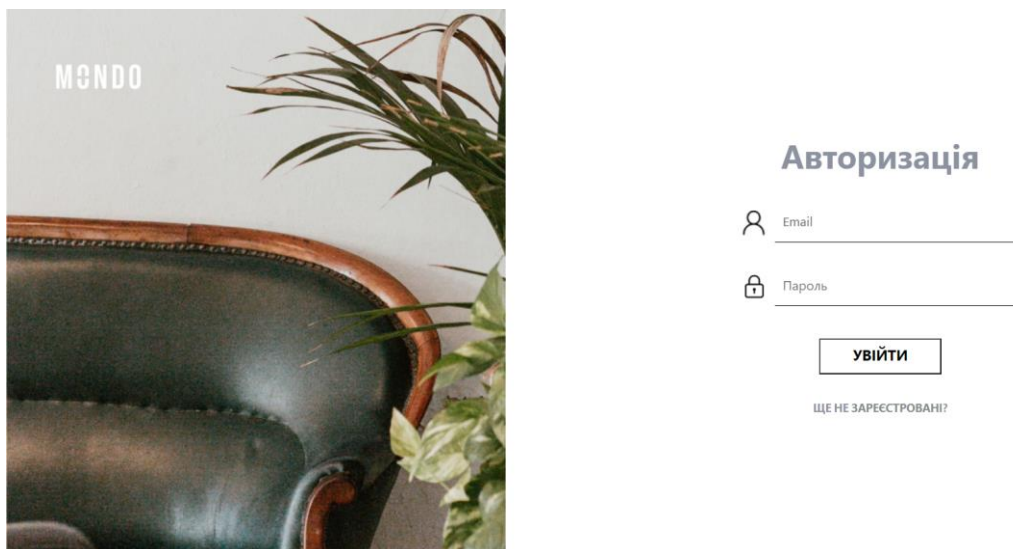


Рисунок 3.12 – Форма авторизації в адміністративній панелі

Після успішної авторизації як адміністратор, користувач перенаправляється на головну сторінку адміністративної панелі. На цій сторінці відображається важлива інформація, включаючи статистику замовлень і список останніх замовлень. Статистика замовлень надає зведену інформацію про загальну кількість замовлень, що дозволяють адміністратору моніторити ефективність продажів. У списку останніх замовлень відображаються недавні замовлення з вказанням деталей, таких як ідентифікатор замовлення, дата та статус. Ця інформація допомагає адміністратору бути в курсі актуальних замовлень та вчасно реагувати на них.

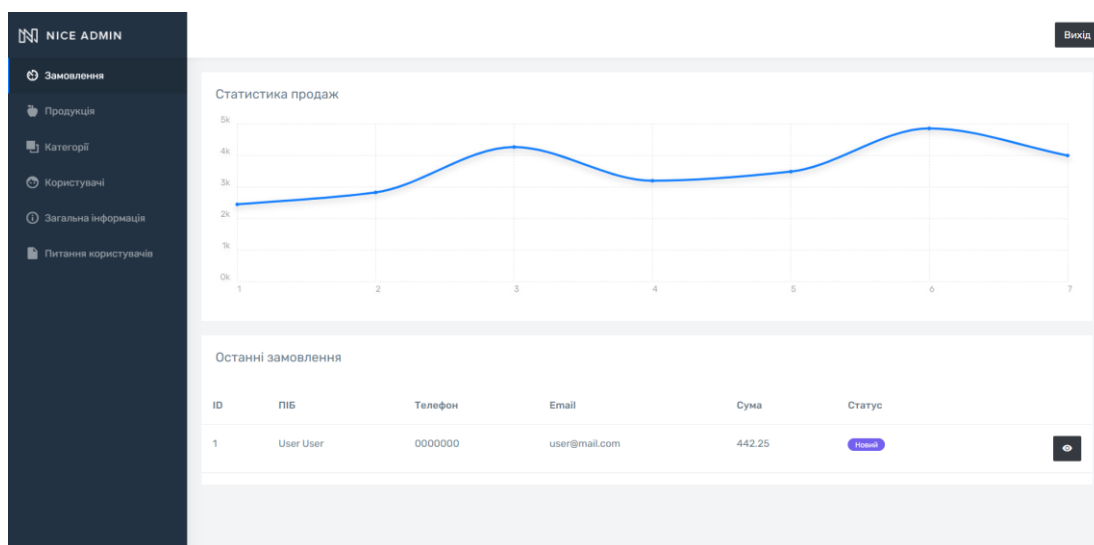


Рисунок 3.13 – Головна сторінка адміністративної панелі

Перед початком додавання товарів до web-додатку необхідно спочатку створити категорії. Це означає, що ви повинні створити структуру категорій перед тим, як почати додавати конкретні продукти. Категорії мають ієрархічну структуру, що означає, що ви можете створювати категорії в межах інших категорій. Хоча технічно глибина категорій не має обмежень, фактична глибина залежить від вимог вашої системи. При створенні категорій немає жорстких правил, і ви можете групувати продукти за логічними та практичними критеріями, які відповідають вашим потребам і стратегії.

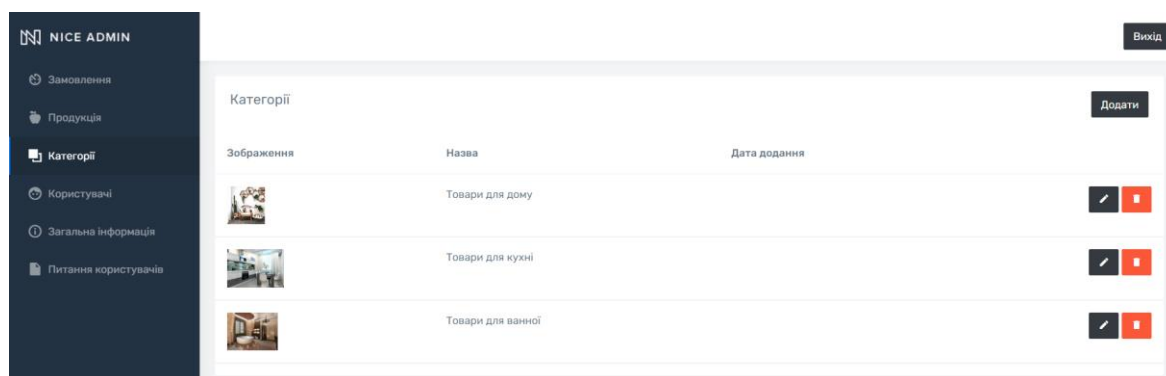


Рисунок 3.14 – Сторінка категорій

Щоб додати товар, необхідно перейти на сторінку «Продукція» і натиснути кнопку «Додати». На сторінці можна задати опис товару, додати зображення, визначити приналежність до категорії та вказати ціну.

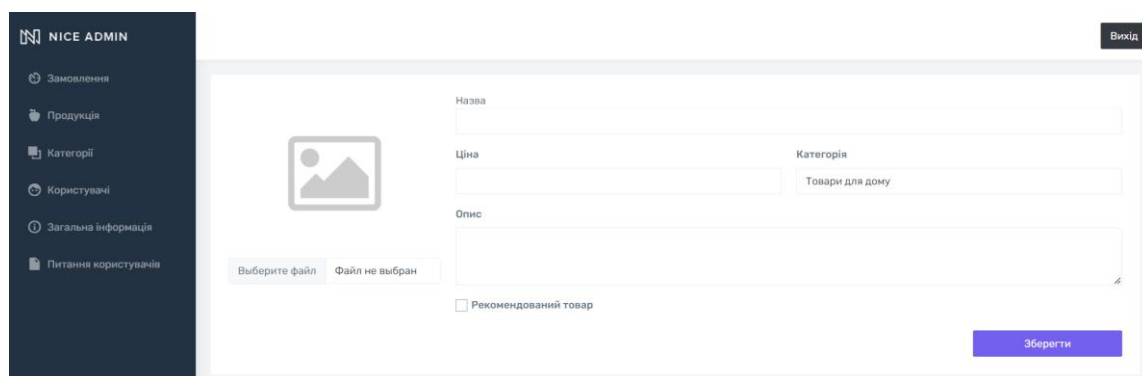


Рисунок 3.15 – Сторінка додання товару

Після додавання товарів їх можна подивитися і змінити, перейшовши на сторінку «Продукція» (рис. 3.16)

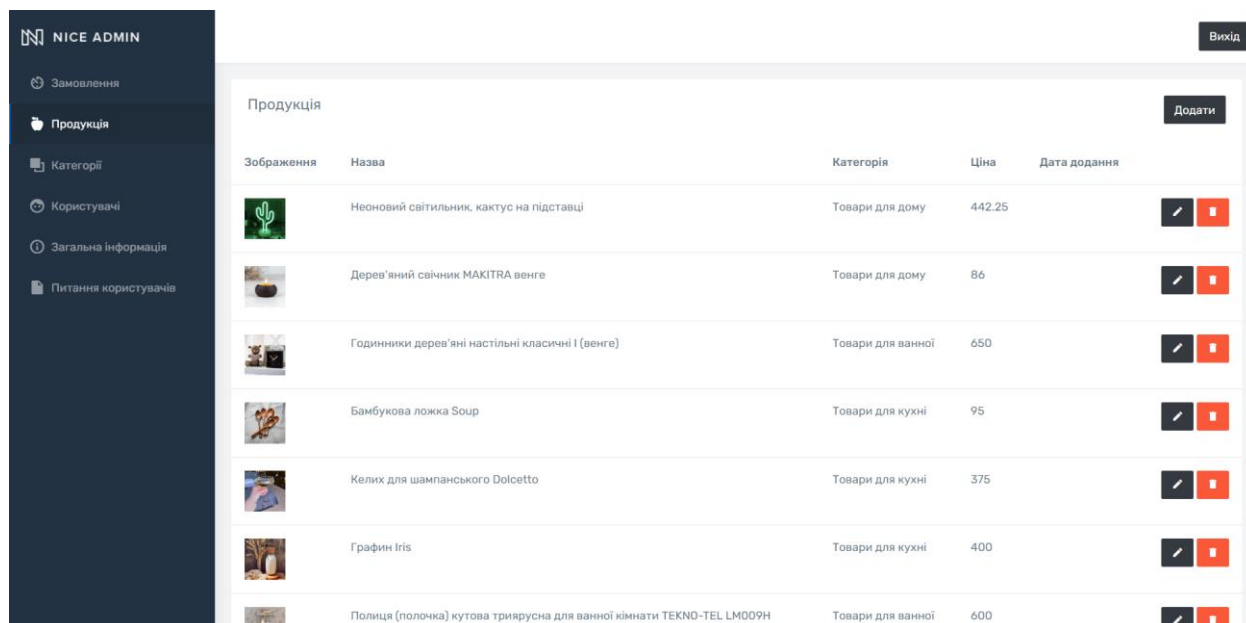


Рисунок 3.16 – Сторінка з товаром

Для перегляду детальної інформації про замовлення переходимо на сторінку «Замовлення» і натиснути піктограму «Ока» біля обраного замовлення.

Останні замовлення					
ID	ПІБ	Телефон	Email	Сума	Статус
1	User User	0000000000	user@mail.com	442.25	Новий

Рисунок 3.17 – Сторінка з замовленнями

На сторінці можна підтвердити замовлення, відхилити або роздрукувати детальну інформацію.

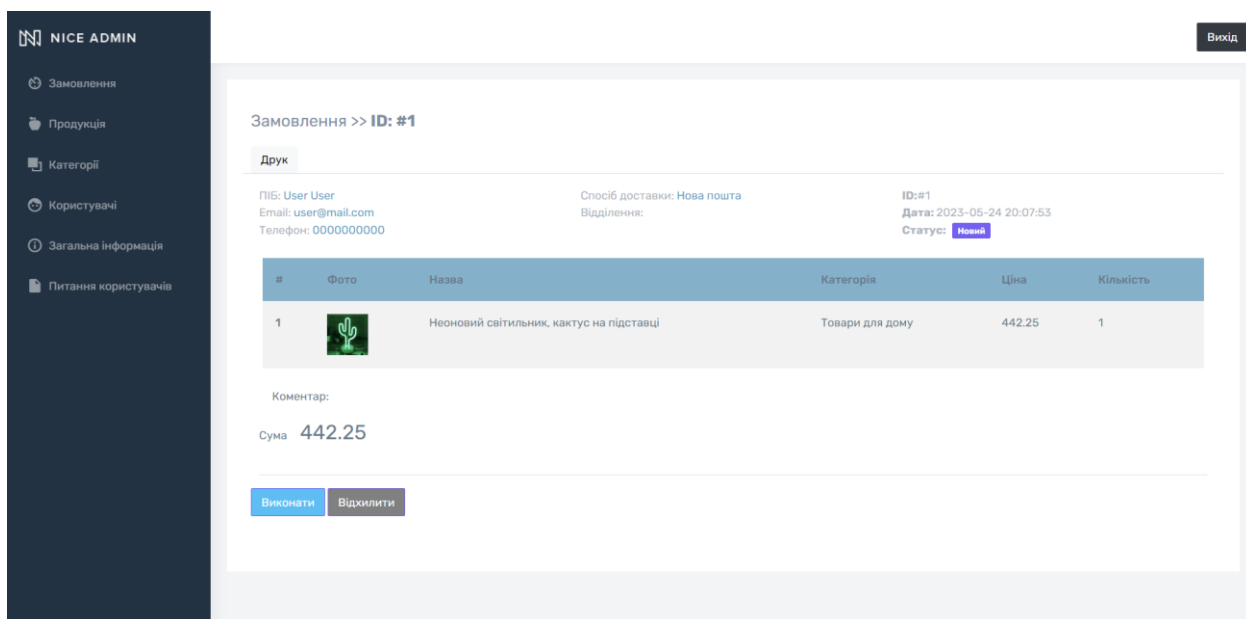


Рисунок 3.18 – Сторінка з замовленнями

Якість і рівень оптимізації було перевірено за допомогою сайту <https://pagespeed.web.dev/>. Результати оптимізації наведено на рис. 3.19.

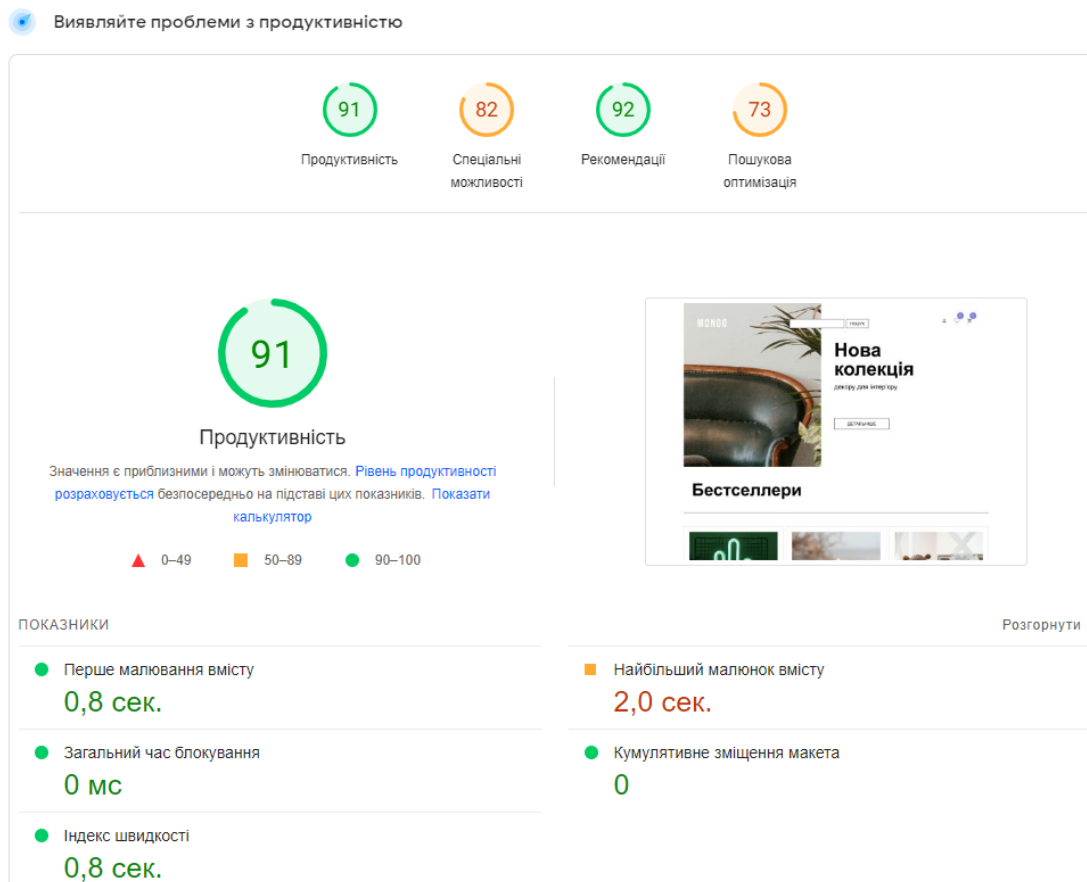


Рисунок 3.19 – Відсоток оптимізації web-додатку

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було проведено аналіз останніх досліджень і публікацій, присвячених web-додаткам для бізнесу та їх розробці. Виявлено, що комерційні web-додатки мають значення для бізнесу та бізнес-процесів. Досліджено різні методи розробки web-додатків та ознайомлено з сучасними web-фреймворками, що полегшують та прискорюють процес розробки.

Також було проведено порівняльний аналіз аналогічних програмних продуктів та встановлено критерії для ідеального додатку. Визначено, що він повинен мати сучасний та релевантний дизайн, зрозумілий інтерфейс, зручну навігацію та інтерактивність web-сторінок.

Сформульовано постановку задачі для кваліфікаційної роботи, в якій були визначені та описані основні функціональні вимоги. Було спроектовано структуру web-додатку, створено базу даних та розроблено серверну частину для реалізації функціоналу.

У результаті виконання кваліфікаційної роботи був розроблений web-додаток для підтримки продажу декоративних товарів. Були розроблені сторінки авторизації/реєстрації, модулі пошуку товарів та оформлення онлайн-замовлень, а також сторінки з повною інформацією про товари та адаптивним дизайном.

У рамках кваліфікаційної роботи були підготовлені додатки, такі як додаток А з технічним завданням, додаток Б з повним списком робіт проекту та додаток В з програмним кодом.

Результати роботи були оприлюднені на науково-технічній конференції ІМА-2023. Тези доповіді розміщені у додатку Г.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. E-Commerce: Purchasing and Selling Online – What You Need to Consider. <http://www.gov.pe.ca>. URL: http://www.gov.pe.ca/photos/original/IPEI_ebiz_ecomm.pdf (дата звернення: 15.05.2023).
2. Kütz M. Introduction to e-commerce combining business and information technology//2016, – с. 209.
3. Васвані Дж. 10 найкращих послуг web-хостингу 2023 (поглиблені огляди) Який найкращий постачальник web-хостингу?. <https://www.bloggersideas.com/>. URL: <https://www.bloggersideas.com/uk/best-rated-web-hosting-services/> (дата звернення: 15.05.2023).
4. Kumar B. How to start an online store in 2023 (step-by-step guide). <https://www.shopify.com/>. URL: <https://www.shopify.com/blog/start-online-store> (дата звернення: 15.05.2023).
5. Bakhur N. How to protect an online store. <https://neklo.com/>. URL: <https://neklo.com/how-to-protect-online-store/> (дата звернення: 15.05.2023).
6. Browser automation: automate website actions like clicks or data extraction. <https://www.fortra.com/>. URL: <https://www.fortra.com/resources/articles/browser-automation-automate-website-actions-clicks-or-data-extraction> (дата звернення: 10.02.2023).
7. Web технології – що це таке та які найпопулярніші? URL: <https://futurenow.com.ua/veb-tehnologiyi-shho-tse-take-ta-yaki-najpopulyarnishi/> (дата звернення: 10.02.2023).
8. How ecommerce automation works and benefits your online business. <https://www.bigcommerce.com/>. URL: <https://www.bigcommerce.com/articles/ecommerce/ecommerce-automation/> (дата звернення: 10.02.2023).

9. Rупак N. 10 ecommerce automation ideas to implement on the website in 2022. <https://webocreation.com/>. URL: <https://webocreation.com/10-ecommerce-automation-ideas-to-implement-on-the-website-in-2022/> (дата звернення: 10.02.2023).
10. Левус Є. В. «Алгоритм вироблення комплексних рекомендацій клієнтам туристичної галузі»./ Левус Є. В., А. О. Полянська// *Науковий вісник НЛТУ України*, вип. 30, вип. 5. – Львів: Національний університет "Львівська політехніка", 2020, - с. 122-127.
11. Browser automation: automate website actions like clicks or data extraction. <https://www.fortra.com/>. URL: <https://www.fortra.com/resources/articles/browser-automation-automate-website-actions-clicks-or-data-extraction> (дата звернення: 10.02.2023).
12. Головна сторінка web-додатку «Decorel Home» [URL:https://decorelhome.ua](https://decorelhome.ua). (дата звернення: 12.03.2023).
13. Головна сторінка web-додатку «JYSK». [URL:https://jysk.ua](https://jysk.ua) (дата звернення: 12.03.2023).
14. Головна сторінка web-додатку «DecorPlus». [URL:https://decorplus-zp.com.ua/ua](https://decorplus-zp.com.ua/ua) (дата звернення: 12.03.2023).
15. Kolowich Cox L. Coding for Web Design 101: How HTML, CSS, and JavaScript Work. <https://blog.hubspot.com/>. URL: <https://blog.hubspot.com/marketing/web-design-html-css-javascript> (дата звернення: 15.05.2023).
16. Бондаренко С. СУБД: які бувають, як вибрати. <https://highload.today>. URL: <https://highload.today/uk/subd-yaki-buvayut-yak-vibrati/> (дата звернення: 15.05.2023).
17. SunTec. Python vs PHP vs javascript: which is best for your next project?. <https://hackernoon.com/>. URL: <https://hackernoon.com/python-vs-php-vs-javascript-which-is-best-for-your-next-project-i33v31y1> (дата звернення: 16.03.2023).

Додаток А

ТЕХНІЧНЕ ЗАВДАННЯ

на розробку

«Web-додаток підтримки продажу декоративних товарів для інтер'єру»

1. Призначення й мета web-додатку підтримки продажу декоративних товарів для інтер'єру

1.1 Призначення web-додатку

Web-додаток призначений для підтримки продажу декоративних товарів для інтер'єру.

1.2 Мета створення web-додатку

Мета проекту – це створення web-додатку для підтримки продажу декоративних товарів для інтер'єру, автоматизації процесів формування та доставки замовлень.

1.3 Цільова аудиторія

Цільовою аудиторією даного проекту є замовник та клієнти інтернет-магазину, які зацікавлені в онлайн-покупці декоративних товарів для інтер'єру.

2 Вимоги до проекту

2.1 Вимоги до проекту в цілому

2.1.1 Вимоги до структури й функціонування

Web-додаток повинен бути реалізований за допомогою таких web-інструментів, як HTML, CSS, JS та PHP, і забезпечувати можливість перегляду, пошуку, замовлення товару, перегляду історії замовлень.

Кінцевий продукт даного проекту має бути представлений web-додатком, який містить якісне інформаційне наповнення та графічні матеріали.

2.1.2 Вимоги до персоналу

Персонал не повинен мати особливих технічних навичок для роботи з web-додатком і його підтримкою. Єдиною вимогою є наявність навичок користування персональним комп'ютером та web-браузером.

2.1.3 Вимоги до збереження інформації

Уся інформація надана у web-додатку повинна зберігатися у базі даних реалізованій засобами системи управління базами даних MySQL.

2.1.4 Вимоги до розмежування доступу

Розроблюваний web-додаток має бути загальнодоступним у мережі Інтернет. Права доступу до інформації розмежовані за групами користувачів: адміністратор, відвідувач та клієнт. Адміністратор має необмежений доступ до даних з правами перегляду, додавання, редагування та видалення. Доступ до адміністративної панелі надається за спеціальним логіном та паролем.

Відвідувач web-додатку може переглядати інформацію на web-сторінках, обирати та замовляти товари. Зареєстрованому клієнту web-додатку спектр доступу до інформації ширший за відвідувача, але менший за адміністратора. До переліку його можливостей входять ті, які визначені в групі користувачів «Зареєстровані» та доступ до особистого кабінету з переглядом історії замовлень та вподобань.

2.2 Структура web-додатку

2.2.1 Загальна інформація про структуру web-додатку

До структури web-додатку входять усі його web-сторінки, які є загальнодоступними, та адміністративна панель для користування персоналом інтернет-магазину.

Перелік сторінок web-додатку наступний:

- «Головна» сторінка містить хедер, рекламний банер, розділ з бестселерами, каталог і футер;
- web-сторінка «Каталог», на якій знаходяться блоки з товарами за категоріями;
- web-сторінка «Категорії», містить товари розподілені за категоріями, фільтр;
- web-сторінка реєстрації, яка містить форму реєстрації;
- web-сторінка авторизації слугує для авторизації користувача та адміністратора каталогу;
- адміністративна панель з web-сторінкою, що слугує для додавання, редагування та видалення інформації про товари та категорії товарів.

2.2.2 Навігаційне меню

Для зручної навігації повинно бути створене меню, що забезпечить швидке переміщення користувача по всім доступним сторінкам web-додатку. Меню має бути закріплене і розташовуватися зверху (у шапці) на кожній сторінці. У ньому мають бути такі пункти:

- каталог (перехід на розділ додатку);
- бестселери (перехід на розділ додатку);
- оплата та доставка (перехід на сторінку додатку з інформацією про умови оплати та доставки);
- особистий кабінет (перехід на сторінку авторизації/реєстрації або перехід на сторінку з інформацією користувача та його історією замовлень);
- вішлист (перехід на сторінку з уподобаннями);
- кошик (перехід на сторінку для замовлення товарів).

2.2.3 Управління контентом

Управління контентом web-додатку має здійснюватися за допомогою адміністративної панелі. Усе інформаційне наповнення web-додатку має міститися у базі даних. Графічні матеріали та інформацію для наповнення надає Замовник.

2.2.4 Дизайн web-додатку

Дизайн web-додатку має бути виконаний у мінімалістичному та сучасному стилі. Корпоративні кольори та шрифти зображені на рис. А.1. Під час розробки web-додатку треба використовувати саме їх.

Види і розміри шрифтів повинні бути комфортними для перегляду. Інформаційні блоки, графічні матеріали та інші елементи web-сторінок повинні мати зручне і логічне розташування.



Рисунок А.1 – Мудборд, кольори, шрифти проекту

Майбутній web-додаток включатиме в себе такі розділи:

- хедер з бургер-меню, логотипом, пошуковим рядком та посиланнями на особистий кабінет, уподобання та кошик (рис. А.2);



Рисунок А.2 – Хедер проекту

- рекламний банер (рис. А.3);

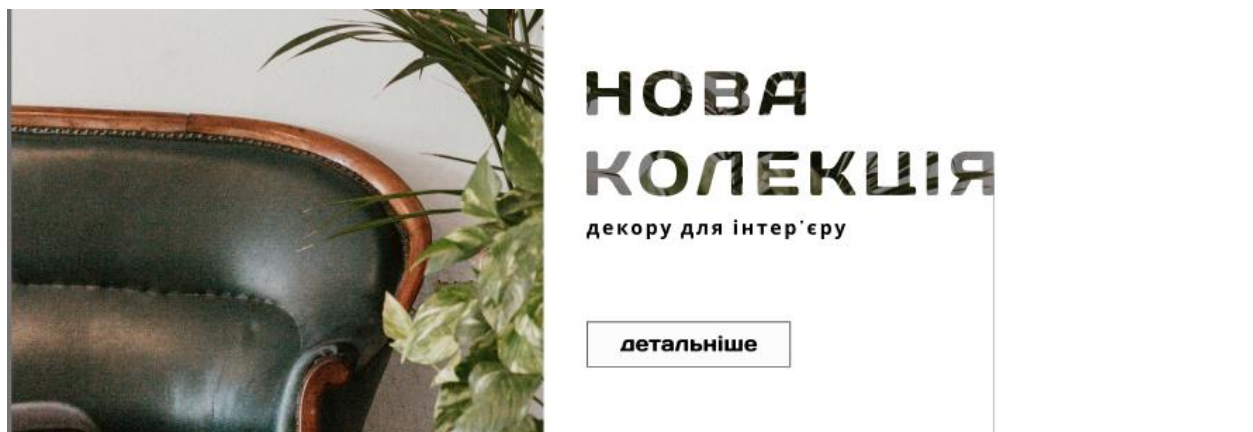


Рисунок А.3 – Рекламний банер

– бестселлери (товари, що користуються найбільшим попитом) (рис. А.4);

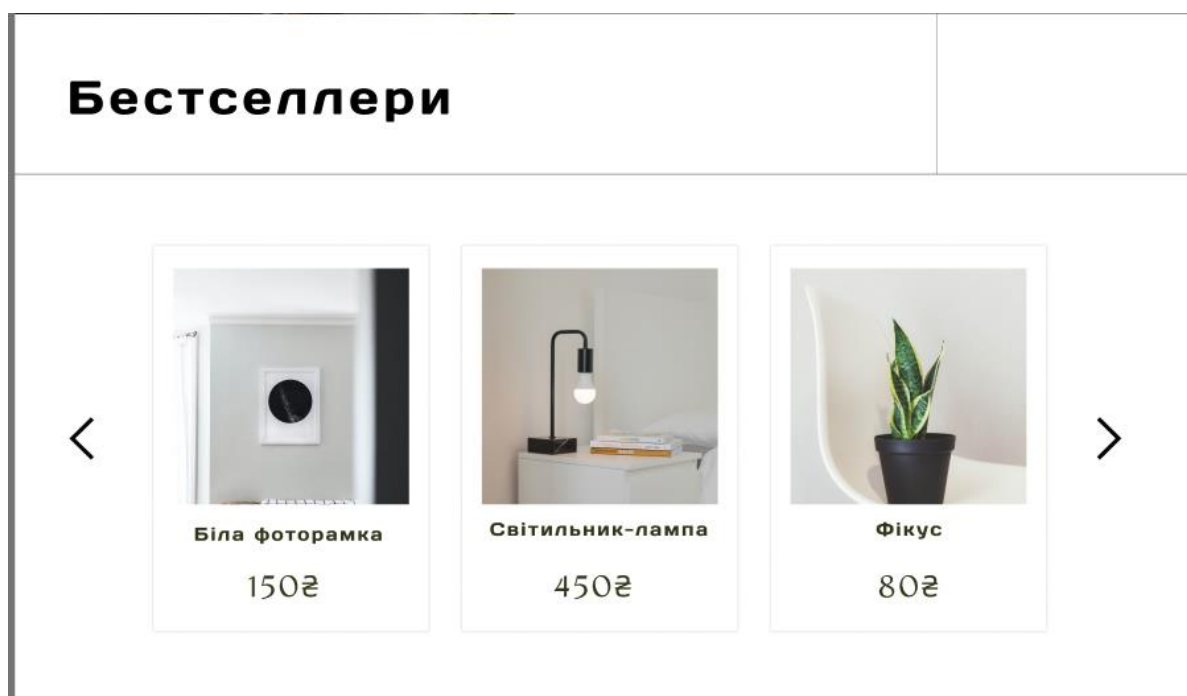


Рисунок А.4 – Бестселлери

– каталог з категоріями товарів (рис. А.5);

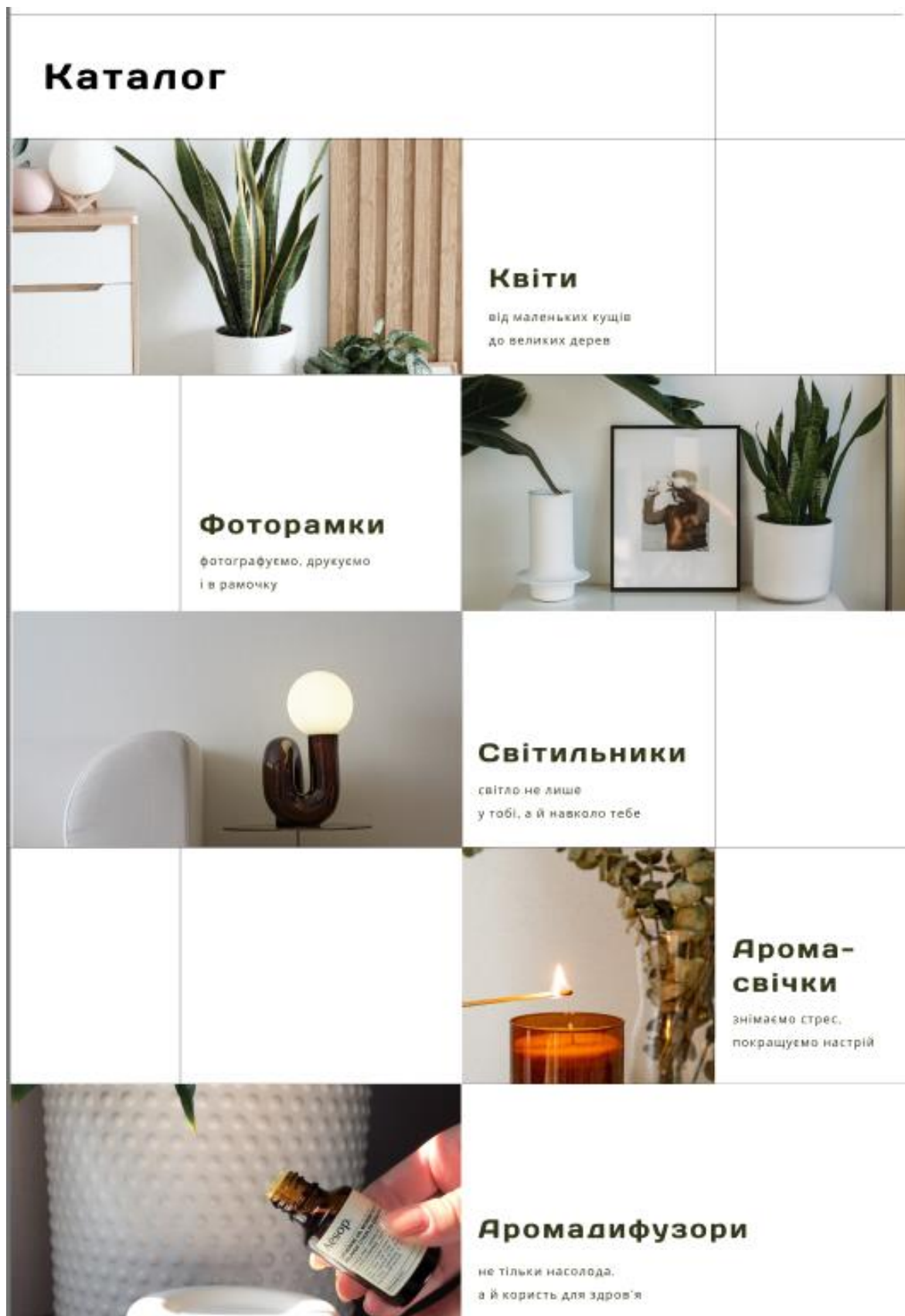


Рисунок А.5 – Каталог

- футер з контактами, посиланнями на корисну інформацію та соц.мережі (рис. А.6).



Рисунок А.6 – Футер

- вигляд сторінки авторизації (рис. А.7).

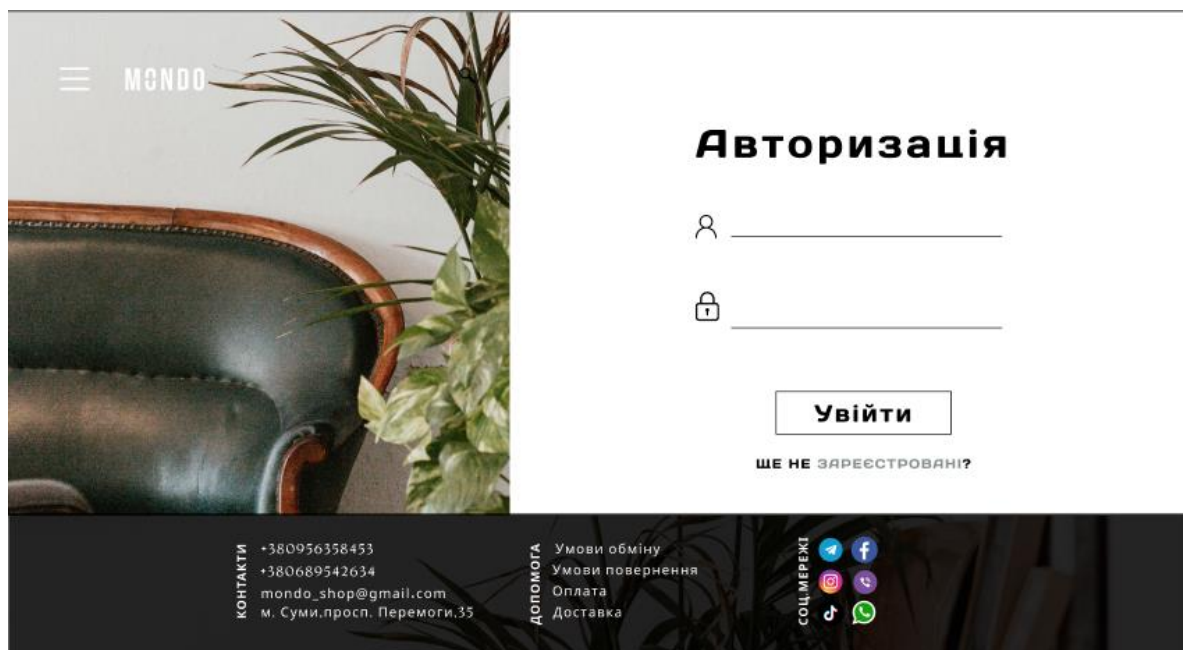


Рисунок А.7 – Сторінка авторизації

– вигляд сторінки реєстрації (рис. А.8).

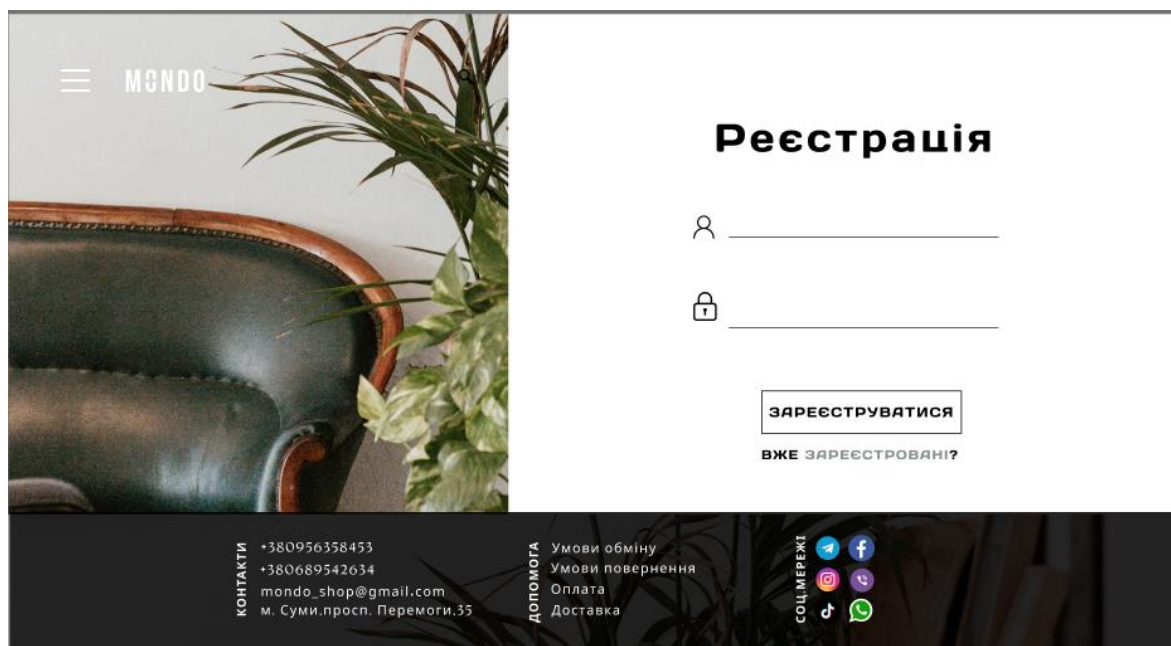


Рисунок А.8 – Сторінка реєстрації

– вигляд сторінки товару (рис. А.9).

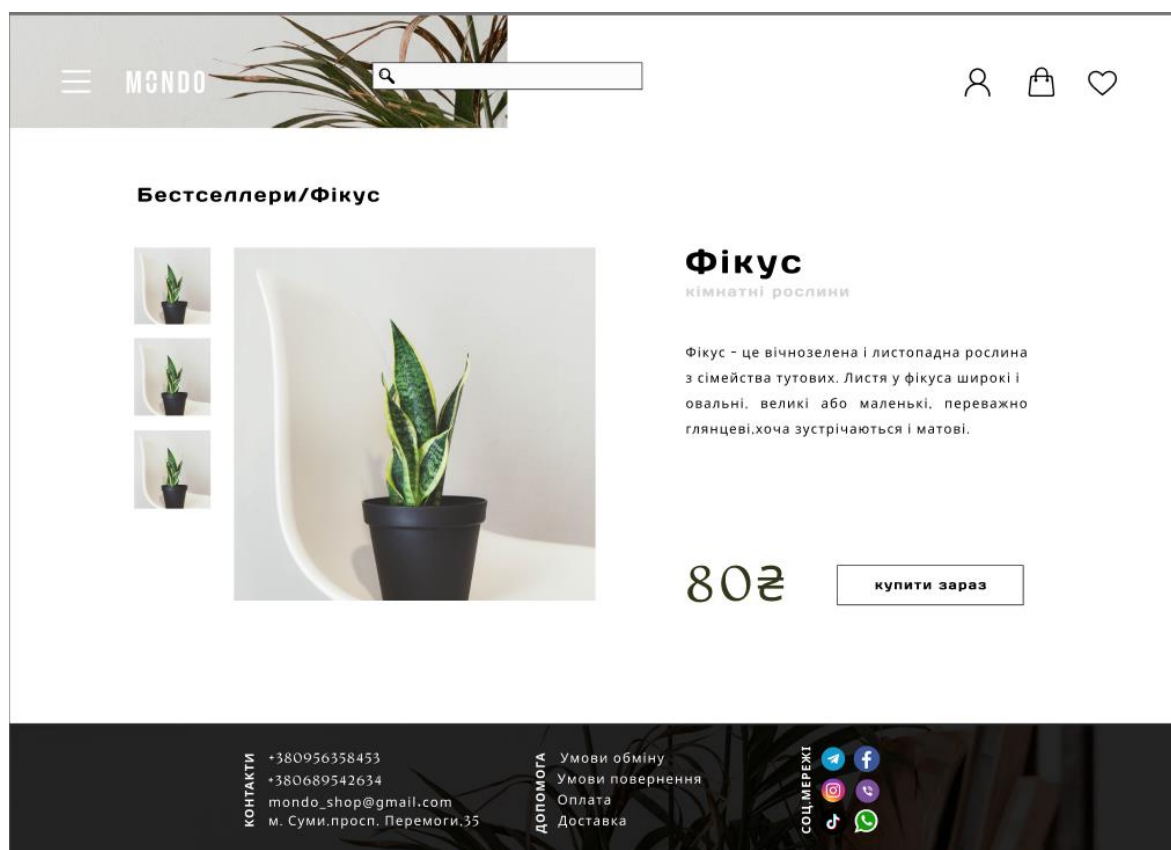


Рисунок А.9 – Сторінка товару

2.2.4 Система навігації (карта web-додатку)

Карта web-додатку зображена на рисунку А.10.

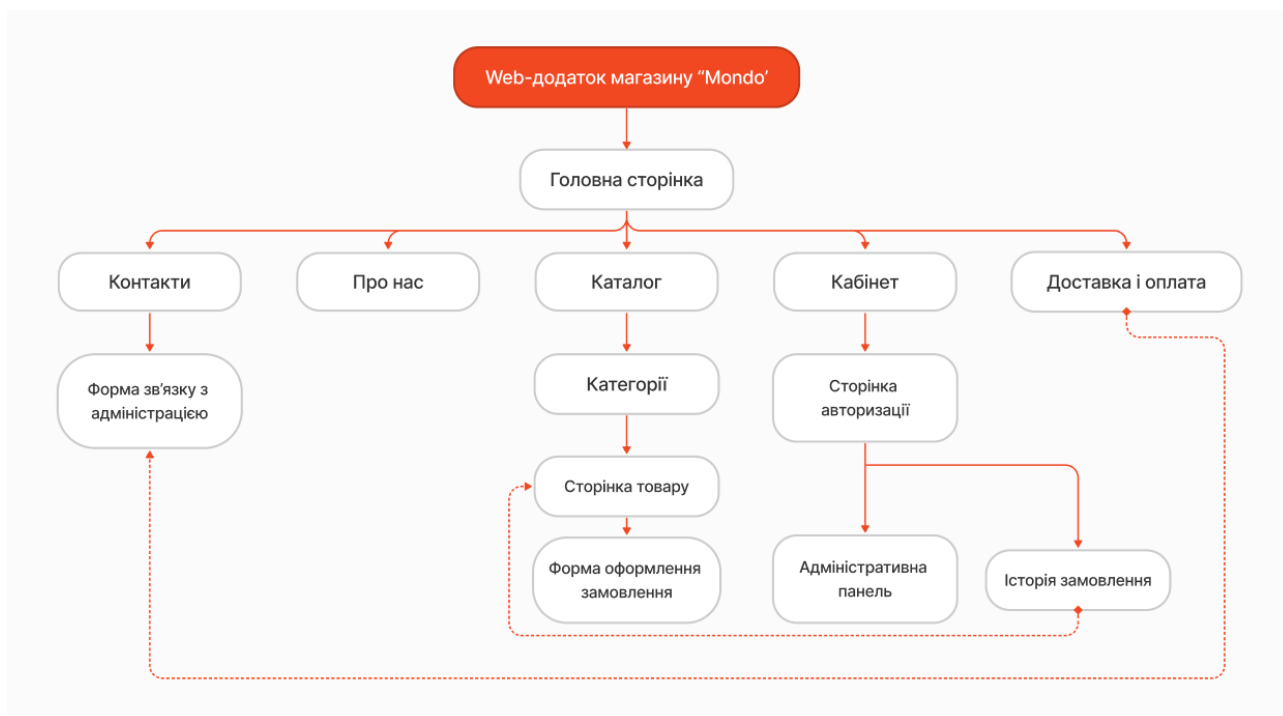


Рисунок А.10 – Система навігації

2.3 Вимоги до видів забезпечення

2.3.1 Вимоги до лінгвістичного забезпечення

Весь текст у web-додатку має бути виконаний українською мовою.

2.3.2 Вимоги до програмного забезпечення

Для забезпечення стабільної роботи web-додатку web-браузер має бути Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище.

2.4 Вимоги до функціонування системи

2.4.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ID	Потреби користувача	Джерело
UN-01	Перегляд продукції магазину декоративних товарів для інтер'єру	Клієнт, Відвідувач
UN-02	Замовлення товару онлайн	Клієнт, Відвідувач
UN-03	Перегляд історії своїх замовлень	Клієнт
UN-04	Перегляд опису товару	Клієнт, Відвідувач
UN-05	Перегляд оголошень	Клієнт, Відвідувач
UN-06	Перегляд інформації про інтернет-магазин	Клієнт, Відвідувач
UN-07	Редагування даних	Адміністратор
UN-08	Зворотній зв'язок	Клієнт, Відвідувач
UN-09	Вибір світлої/темної версії додатку	Клієнт, Відвідувач
UN-10	Статистика відвідувань додатку	Адміністратор
UN-11	Пошук товарів у web-додатку	Клієнт, Відвідувач

2.4.2 Системні вимоги

Проаналізувавши потреби користувачів та персоналу закладу було визначено наступні вимоги:

- наявність реєстрації та авторизації клієнтів;
- пошук товару у web-додатку (за назвою, категорією);
- можливість оформлення онлайн-замовлень за допомогою форми;
- наявність повної інформації про товари;
- наявність панелі адміністратора для додавання, редагування та видаленні інформації з web-додатку;
- наявність адаптивного дизайну.

3 Склад і зміст робіт зі створення web-додатку

Детальний опис етапів створення web-додатку наведено в таблиці А.2.

Таблиця А.2 – Етапи створення web-додатку

№	Склад і зміст робіт	Строк розробки
1	Розробка шаблону web-додатку	5 дні
2	Задання верстки сторінок web-додатку	15 днів
3	Розробка бази даних	7 днів
4	Розробка модулю каталогу товарів	10 днів
5	Розробка модулю оформлення замовлень	5 день
6	Розробка панелі адміністратора	2 дні
7	Розробка модулю авторизації та реєстрації	2 дні
8	Бета-тестування	5 дні
9	Alpha-тестування	10 дні
10	Розміщення на хостингу	1 день
11	Перевірка працездатності	3 дні
12	Написання супровідної документації	7 дні
13	Реліз web-додатку	1 день
	Загальна тривалість робіт	73 дня

4 Вимоги до складу й змісту робіт із введення web-додатку в експлуатацію

Web-додаток має бути затверджено та розміщено на web-хостингу.

Додаток Б

Планування робіт

Деталізація мети проекту методом SMART.

Щоб проект був успішним та конкурентоспроможним треба на концептуальному етапі правильно визначити його мету за допомогою SMART-методу. Результати деталізації методом SMART розміщені у таблиці Б.1.

Таблиця Б.1 - Деталізація мети проекту методом SMART

Specific	Web-додаток для підтримки продажу декоративних товарів для інтер'єру.
Measurable	Збільшення прибутку магазину на 20%.
Achievable	Мета досяжна, оскільки є кваліфіковані спеціалісти (дизайнери, програмісти, тестувальники), є необхідна техніка та ліцензійні програмні середовища для розробки web-додатків.
Relevant	Для підвищення рейтингу та конкурентоспроможності web-додатку.
Time-framed	Розробка програмного проекту буде завершена до 20 травня 2022 р.

Планування змісту робіт

WBS (Work Breakdown Structure – Ієрархічна структура робіт) – це графічний вигляд елементів проекту, які згруповані ієрархією єдине ціле з продуктом проекту. Структура декомпозиції робіт орієнтована на досконале виконання робіт по частинам і сама є ключовою частиною проекту, яка спрямована на організацію командної роботи. Елементами декомпозиції можуть бути продукти, дані та послуги. Більше того, WBS забезпечує необхідним каркасом для ретельної оцінки термінів та контролю та графіків роботи.

На найвищому (першому) рівні розміщений продукт проекту. Основні дії та заходи, що забезпечують досягнення мети проекту, зафіксовані на другому рівні декомпозиції. Декомпозиція робіт виконується до тих пір, поки вони не стануть елементарними (простими).

Елементарні роботи – це дії, які мають однозначний чіткий результат, на які призначена відповідальному одна конкретна особа, для якої можна обчислити витрати праці і тривалість виконання. На рисунку Б.1 представлено WBS з розробки

web-додатку.

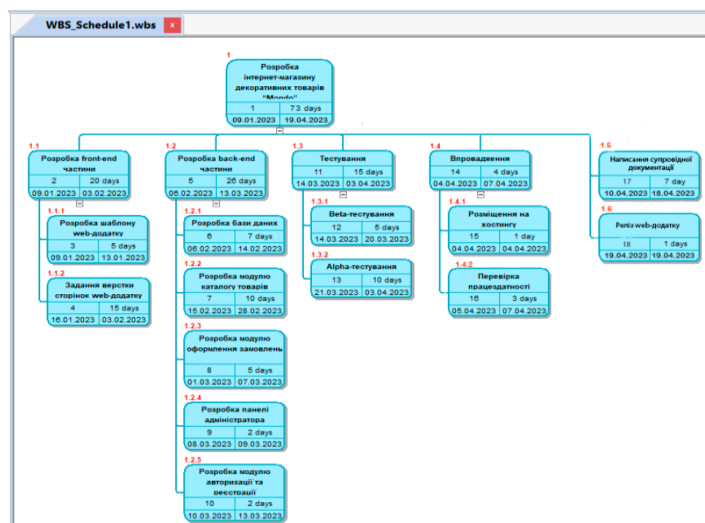


Рисунок Б.1 – WBS-структура робіт проекту

Планування структури виконавців

Наступним етапом після декомпозиції процесів є розробка організаційної структури виконавців або OBS, яка визначається як графічна структура відображення учасників або відповідальних осіб, які беруть участь у реалізації проекту.

У ролі відповідальних осіб виступають співробітники, що відповідають за організацію і виконання елементарної роботи, що зазначена у WBS. Кожну елементарну роботу можна розглядати як окремий проект.

На рисунку Б.2 представлено організаційну структуру планування проекту.

Список виконавців, що функціонують в проекті описано в таблиці Б.2

Таблиця Б.2 – Список виконавців

Роль	Ім'я	Проектна роль
Розробник	Самочкіна Т.В.	Виконує front-end та back-end розробку
Проектувальник	Самочкіна Т.В.	Виконує проектування бази даних та розробляє структуру web-додатку.
Тестувальник	Самочкіна Т.В.	Відповідає за тестування функціоналу та дизайну web-додатку.
Керівник проекту	Парфененко Ю.В.	Формує завдання на розробку проекту.
Менеджер проекту	Самочкіна Т.В., Парфененко Ю.В.	Відповідає за виконання термінів. Виконує збір та аналіз даних.

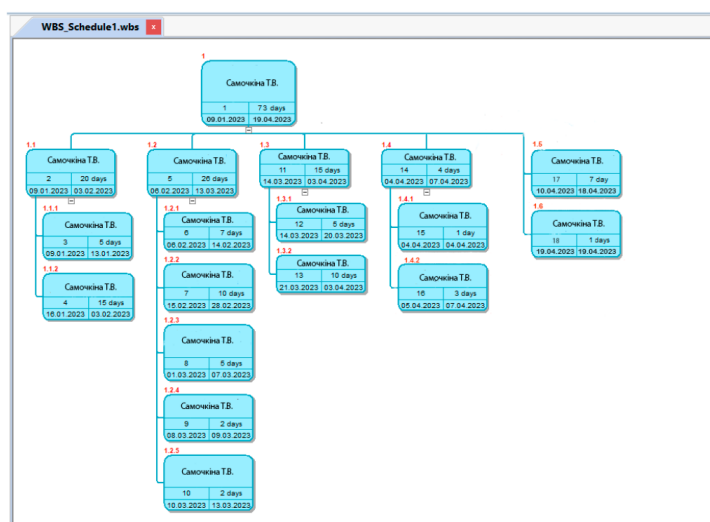


Рисунок Б.2 – OBS-структура робіт проекту

Діаграма Ганта

Побудова календарного графіку (діаграми Ганта) є одним з важливих етапів планування проекту, що виглядає як розклад виконання робіт з реальним розподілом дат. Завдяки йому можна отримати достовірне уявлення про тривалість процесів з обмеженнями у ресурсах, урахуванням вихідних днів та свят.

Календарний графік проекту представлено на рисунках Б.3.

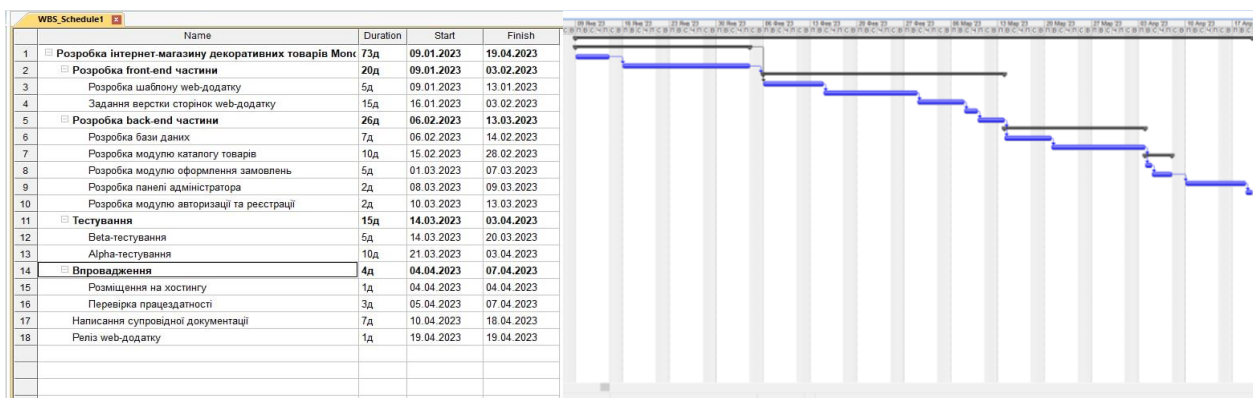


Рисунок Б.3 – Календарний графік проекту

Управління ризиками проекту

Під час виконання якісної оцінки ризиків треба визначити ризики, які мають бути усунені якнайшвидше. В залежності від ступеня важливості ризику – реагування буде відповідне. Наступним етапом є виконання кількісного оцінювання ризиків. Кількісне та якісне оцінювання можуть виконувати одночасно або окремо, що

залежить від ступеня забезпечення проекту. У таблиці Б.3 представлено шкалу для класифікації ризиків за величиною впливу на проект та ймовірністю виникнення.

Таблиця Б.3 – Шкала оцінювання ризиків за ймовірністю виникнення та величиною впливу.

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Класифікація ризиків за рівнем, відповідно до отриманого значення індексу, представлена на рисунку Б.4. У таблиці Б4 представлено шкалу оцінювання за рівнем ризику. У таблиці Б.5 описано ризики та стратегії реагування накожен з них.

Ймовірність ризику (Й)	Вплив загрози (ризик)				
	Дуже малий	Малий	Середній	Великий	Дуже великий
	0,05	0,1	0,2	0,4	0,8
0,9	0,045	0,09	0,18	0,36	0,72
0,7	0,035	0,07	0,14	0,25	R5(0,56)
0,5	0,025	0,05	0,10	R1(0,2)	R4(0,40)
0,3	0,015	0,03	R2,R6(0,06)	R3(0,12)	0,24
0,1	0,05	0,01	0,02	0,04	0,08

Рисунок Б.4 – Класифікація ризиків за рівнем

Таблиця Б.4 – Шкала оцінювання за рівнем ризику.

Назва	Межі	Ризики які входять (номера)
Прийнятні	$0,005 \leq R \leq 0,05$	-
Виправдані	$0,05 < R \leq 0,14$	2,3,6
Недопустимі	$0,14 < R \leq 0,72$	1,4,5

Таблиця Б.5 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Нечітке завдання на розробку	0,4	0,5	0,2	1. Грамотно поставити технічне завдання проекту. 2. Звітування перед замовником на різних етапах роботи. 3. Уникати нечітких, та невимірюваних характеристик продукту.	Попередження	Якщо є розбіжності між характеристиками продукту та ТЗ - виконати виправлення.
RS_2	Відкритий	Низька кваліфікація розробників	0,4	0,5	0,2	1. Провести ретельний відбір серед кваліфікованих претендентів за допомогою рекрутерів..	Попередження	Якщо робітник або ціла команда виявилися недостатньо компетентними - передати проект іншій команді, або знайти нового робітника за допомогою рекрутера.
RS_	Відкритий	Зміни у ТЗ	0,3	0,2	0,06	1. Ясно і однозначно обговорити із замовником усі види вимог. 2. Скласти глосарій для запобігання розбіжностей у розумінні слів та термінів. 3. Періодичний контроль замовником етапів роботи.	Ухилення	При виявленні невідповідностей деяких характеристик продукту заявленим вимогам потрібно уважно та чітко окреслити те, що було виконано невірно та зробити правки.

Продовження таблиці Б.5

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_4	Відкритий	Неоптимальний розподіл часу	0,3	0,2	0,06	1. Розрахувати час виконання замовлення із запасом	Попередження	Повне перепланування з урахування усіх обтяжуючих обставин.
RS_5	Відкритий	Повітряні тривоги	0,5	0,9	0,45	1. Перемістити офіс подалі від бойових дій. Допомогти учасникам проекту покинути обстрілювану територію.	Ухилення	Перенести строки виконання робіт. Перепланування.
RS_6	Відкритий	Перебої електропостачання	0,7	0,7	0,7	1. Виконувати роботу на ноутбуках, або інших пристроях з автономними джерелами живлення. 2. Закупити переносні генератори та потужні павербанків 3. Закласти в календарний план додаткові резерви. 4. Мати резервне джерело мобільного інтернету.	Ухилення	Переміщення офісу або робітників у населений пункт з стабільної енергосистемою

Додаток В

КОД РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

OrdersController.php

Контролер замовлень.

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Order;
```

```
use App\Models\OrderProducts;
```

```
use App\Models\DeliveryTypes;
```

```
use App\Models\OrderStatus;
```

```
use App\Models\Category;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Support\Facades\Auth;
```

```
class OrdersController extends Controller
```

```
{
    /**
     * Show the form for creating a new resource.
     *
     */
    public function create()
    {
        $cartId = isset($_COOKIE['cart_id']) ? $_COOKIE['cart_id'] : null;
        $cart = \Cart::session($cartId);
        $data = $cart->getContent();

        $sum = 0;

        foreach ($data as $item) {
            $sum += ($item['price'] * $item['quantity']);
        }

        $deliveryTypes = DeliveryTypes::get();
        $categories = Category::get();

        return view('checkout', compact('data', 'sum', 'deliveryTypes',
'categories'));
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     */
    public function store(Request $request)
    {
        $data = $request->all();
        $model = new Order();
        $order = $model->create($data);

        $cartId = isset($_COOKIE['cart_id']) ? $_COOKIE['cart_id'] : null;
        $cart = \Cart::session($cartId);
    }
}
```

```

        $cartContent = $cart->getContent();

        foreach ($cartContent as $item) {
            OrderProducts::create([
                'orders_id' => $order->id,
                'product_id' => $item['id'],
                'quantity' => $item['quantity']
            ]);
        }

        $cart->clear();

        return redirect('/thankyou');
    }

    public function edit($id)
    {
        return view('admin.updateOrder', [
            'order' => Order::with('delivery')->find($id),
            'products' => OrderProducts::with('product.category')-
>where('orders_id', $id)->get(),
            'statuses' => OrderStatus::get()
        ]);
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     */
    public function update(Request $request, $id)
    {
        $data = $request->all();
        $model = Order::find($id);
        $model->update($data);
        return response()->json();
    }

    /**
     * Remove the specified resource from storage.
     *
     */
    public function destroy($id)
    {
        Order::find($id)->delete();
        return response()->json();
    }
}

```

ProductController.php

Контролер товару.

<?php

```

namespace App\Http\Controllers;

use App\Models\Product;
use App\Models\Category;
use Illuminate\Http\Request;

class ProductController extends Controller
{
    protected $publicStorage = '/uploads/';
    /**
     * Display a listing of the resource.

```

```

*
*/
public function index()
{
    return view('admin.products', [
        'data' => Product::with('category')->get()
    ]);
}

/**
 * Show the form for creating a new resource.
 *
 */
public function create()
{
    return view('admin.createProduct', [
        'categories' => Category::get()
    ]);
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 */
public function store(Request $request)
{
    $data = $request->all();
    $photo = $request->file('photo');

    if ($photo) {
        $filename = date('YmdHi') . $photo->getClientOriginalName();
        $photo->move(public_path($this->publicStorage), $filename);
        $data['photo'] = $this->publicStorage . $filename;
    }

    Product::create($data);

    return redirect('/dashboard/product');
}

/**
 * Display the specified resource.
 *
 * @param \App\Models\Product $product
 */
public function show($id)
{
    $data = Product::find($id);
    $recommendation = Product::where('recommendation', 1)->with('category')-
>get();
    return view('shop-single', [
        'data' => $data,
        'recommendation' => $recommendation,
        'categories' => Category::get()
    ]);
}

/**
 * Show the form for editing the specified resource.
 *
 * @param \App\Models\Product $product
 */
public function edit($id)

```

```

{
    return view('admin.updateProduct', [
        'categories' => Category::get(),
        'product' => Product::find($id)
    ]);
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Models\Product $product
 */
public function update(Request $request, $id)
{
    $data = $request->all();
    $photo = $request->file('photo');

    if ($photo) {
        $filename = date('YmdHi') . $photo->getClientOriginalName();
        $photo->move(public_path($this->publicStorage), $filename);
        $data['photo'] = $this->publicStorage . $filename;
    }

    $model = Product::findOrFail($id);
    $model->update($data);

    return redirect('/dashboard/product');
}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Models\Product $product
 */
public function destroy($id)
{
    Product::find($id)->delete();
    return response()->json();
}
}

```

WishlistController.php

Контролер вподобань.

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\Models\Category;
```

```
use App\Models\Product;
```

```
class WishlistController extends Controller
```

```

{
    public function index() {
        if(isset($_COOKIE['wishlist_id'])) {
            \Cart::session($_COOKIE['wishlist_id']);
        }
        $data = \Cart::getContent();

        return view('wishlist', [
            'data' => $data,
            'categories' => Category::get()
        ]);
    }
}

```

```

public function store(Request $request) {

    $product = Product::find($request->id);

    if(!isset($_COOKIE['wishlist_id'])) {
        $uniqid = uniqid();
        setcookie('wishlist_id', $uniqid);
        $wishlist_id = $uniqid;
    } else {
        $wishlist_id = $_COOKIE['wishlist_id'];
    }

    if(isset($_COOKIE['wishlist_id'])) {
        \Cart::session($_COOKIE['wishlist_id']);
    }

    $items = \Cart::getContent();

    foreach ($items as $value) {
        if($value->id == $request->id) {
            return response()->json([
                'result' => 'Товар вже додано до обраного.',
                'concat' => 0
            ]);
        }
    }

    \Cart::session($wishlist_id)->add([
        'id' => $product->id,
        'name' => $product->title,
        'price' => $product->price,
        'quantity' => 1,
        'attributes' => [
            'photo' => $product->photo,
            'category' => $product->category['title']
        ],
        'associatedModel' => $product
    ]);

    return response()->json([
        'result' => 'Товар додано до обраного!',
        'concat' => 1
    ]);
}

public function destroy($id) {
    $wishlist_id = $_COOKIE['wishlist_id'];
    \Cart::session($wishlist_id)->remove($id);
    return response()->json();
}
}

```

AccountController.php

Контроллер профілю користувача.

```
<?php
```

```

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use App\Models\User;
use App\Models\Category;
use App\Models\OrderProducts;

```



```

class AccountController extends Controller
{
    public function show() {
        $userId = Auth::id();
        $orders = OrderProducts::with('product')->whereHas('order',
function($query) use ($userId) {
    $query->where('users_id', $userId);
})->limit(15)->get();
        return view('account', [
            'user' => User::find($userId),
            'orders' => $orders,
            'categories' => Category::get()
        ]);
    }

    public function update(Request $request) {
        $data = $request->all();
        $model = User::find(Auth::id());
        $model->update($data);
        return view('account');
    }
}

```

AdminController.php

Контроллер адміністративної панелі.

```
<?php
```

```

namespace App\Http\Controllers;

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

class AdminController extends Controller
{
    public function dashboard()
    {
        $model = Order::select(\DB::raw("DATE_FORMAT(created_at, '%Y-%m-%d') AS
created_at"), \DB::raw('count(*) as count'))
->groupBy('created_at')
->pluck('count', 'created_at')
->all();
        $statistic = [
            'labels' => [],
            'series' => []
        ];
        foreach ($model as $key => $value) {
            array_push($statistic['labels'], "' . $key . '");
            array_push($statistic['series'], Order::whereDate('created_at',
$key)->count());
        }
        return view('admin.dashboard', [
            'statistic' => $statistic,
            'orders' => Order::with('status')->get()
        ]);
    }

    public function users()
    {
        return view('admin.users', [
            'users' => User::get()
        ]);
    }
}

```

```

public function delivery()
{
    $data = Info::pluck('value', 'key');
    return view('admin.delivery', [
        'data' => $data
    ]);
}

public function products()
{
    return view('admin.products', [
        'users' => User::get()
    ]);
}
}

```

welcome.blade.php

Шаблон головної сторінки.

```

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

<head>
    <title>{{ config('app.name', 'Laravel') }}</title>
    <link rel="stylesheet" href="{{ asset('css/bootstrap.min.css') }}">
    <link rel="stylesheet" href="{{ asset('css/custom.css') }}">
    <link rel="stylesheet" href="{{ asset('css/style.css') }}">
    <link rel="stylesheet" href="{{ asset('css/jquery-ui.css') }}">
    <link rel="stylesheet" href="{{ asset('fonts/icomoon/style.css') }}">

    <link rel="stylesheet" href="https://cdn.jsdelivrivr.net/npm/swiper@9/swiper-
bundle.min.css" />
    <link rel="stylesheet" href="https://unpkg.com/swiper@8/swiper-
bundle.min.css" />
    <style>
        .slide {
            height: 500px;
        }
    </style>
</head>

<body>
    <div class="container">
        <header class="header">
            <div class="navigation">
                
                <form action="/shop">
                    <input type="search" name="title">
                    <button>Пошук</button>
                </form>
                <ul>
                    <li>
                        <a href="/account"><span class="icon icon-
person"></span></a>
                    </li>
                    <li class="ml-3">
                        <a href="/wishlist" class="site-cart">
                            <span class="icon icon-heart-o"></span>
                            <span class="count" id="wishlist-count">{{
isset($_COOKIE['wishlist_id']) ? \Cart::session($_COOKIE['wishlist_id'])-
>getTotalQuantity() : 0 }}</span>
                        </a>
                    </li>
                    <li class="ml-3">
                        <a href="/cart" class="site-cart">
                            <span class="icon icon-shopping_cart"></span>

```

```

                <span class="count" id="basket-count">{{
isset($_COOKIE['cart_id']) ? \Cart::session($_COOKIE['cart_id'])-
>getTotalQuantity() : 0 }}</span>
                </a>
            </li>
            <li class="d-inline-block d-md-none ml-3">
                <a href="#" class="site-menu-toggle js-menu-
toggle"><span class="icon-menu"></span></a>
            </li>
            @if(Auth::user())
            <li class="ml-3">
                <a href="{{ route('logout') }}"
onclick="event.preventDefault(); document.getElementById('logout-
form').submit();">
                    <span class="icon icon-exit_to_app"></span>
                </a>
                <form id="logout-form" action="{{ route('logout') }}"
method="POST" class="d-none">
                    @csrf
                </form>
            </li>
            @endif
        </ul>
    </div>
    <div class="banner" data-aos="fade-up">
        <div class="title">
            <h1>Нова<br> колекція</h1>
            <h2>декору для інтер'єру</h2>
            <a href="/shop">Детальніше</a>
        </div>
    </div>
</header>
<main>
    <div class="title-block">
        Бестселлери
    </div>
    <div class="swiper">
        <div class="swiper-wrapper mb-4">
            @foreach($data as $item)
            <div class="swiper-slide">
                <div class="slide">
                    <a class="title" href="/product/{{ $item['id'] }}">
                        
                    </a>
                    <a class="title" href="/product/{{ $item['id'] }}">{{
$item['title'] }}</a>
                    <div class="price">{{ $item['price'] }}</div>
                </div>
            </div>
            @endforeach
        </div>
        <div class="swiper-button-prev"></div>
        <div class="swiper-button-next"></div>
        <br>
        <div class="swiper-pagination"></div>
    </div>
    <div class="title-block">
        Каталог
    </div>
    @foreach($categories as $key => $item)
    <div class="catalog-item <?= $key % 2 == 0 ? 'left' : 'right' ?>">
        
        <div class="content">
            <div class="title">

```

```

                <a href="/shop?category_id={{ $item['id'] }}">{{
$item['title'] }}</a>
            </div>
            <p>{{ $item['subtitle'] }}</p>
        </div>
    </div>
    @endforeach

    <div class="text-center my-5">
        
    </div>
</main>
</div>
@include('inc.footer2', [
    'categories' => $categories
])
<script src="https://unpkg.com/swiper@8/swiper-bundle.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.j
s"></script>

<script>
    const swiper = new Swiper('.swiper', {
        slidesPerView: 3,
        spaceBetween: 10,
        loop: true,
        autoplay: {
            delay: 5000,
        },
        pagination: {
            el: '.swiper-pagination',
        },
        navigation: {
            nextEl: '.swiper-button-next',
            prevEl: '.swiper-button-prev',
        },
    });
</script>
</body>

</html>

```

cart.blade.php

Шаблон сторінки кошику.

```

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

<head>
    <title>{{ config('app.name', 'Laravel') }}</title>
    @include('inc.header')
</head>

<body>
    <div class="site-wrap">
        @include('inc.head', [
            'categories' => $categories
        ])
        <div class="bg-light py-3">
            <div class="container">
                <div class="row">
                    <div class="col-md-12 mb-0"><a href="/">Головна</a> <span class="mx-2
mb-0"></span> <strong class="text-black">Кошик</strong></div>
                </div>
            </div>
        </div>
    </div>

```

```

@if(count($data) > 0)
<div class="site-section">
  <div class="container">
    <div class="row mb-5">
      <form class="col-md-12" method="post">
        <div class="site-blocks-table">
          <table class="table table-bordered">
            <thead>
              <tr>
                <th class="product-thumbnail">Фото</th>
                <th class="product-name">Назва</th>
                <th class="product-price">Ціна</th>
                <th class="product-quantity">Кількість</th>
                <th class="product-remove"></th>
              </tr>
            </thead>
            <tbody>
              @foreach ($data as $item)
                <tr>
                  <td class="product-thumbnail">
                    
                  </td>
                  <td class="product-name">
                    <h2 class="h5 text-black">{{ $item['name'] }}</h2>
                  </td>
                  <td>{{ $item['price'] }}грн</td>
                  <td>
                    <div class="input-group mb-3" style="max-width: 120px;">
                      <div class="input-group-prepend">
                        <button index="{{ $item['id'] }}" class="btn btn-
outline-primary js-btn-minus quantity" type="button">&minus;</button>
                      </div>
                      <input index="{{ $item['id'] }}" type="number"
class="form-control text-center quality-count" value="{{ $item['quantity'] }}"
min="0" max="100" aria-label="Example text with button addon" aria-
describedby="button-addon1">
                      <div class="input-group-append">
                        <button index="{{ $item['id'] }}" class="btn btn-
outline-primary js-btn-plus quantity" type="button">&plus;</button>
                      </div>
                    </div>
                  </td>
                  <td><a href="#" index="{{ $item['id'] }}" class="btn btn-
primary btn-sm remove-cart">X</a></td>
                </tr>
              @endforeach
            </tbody>
          </table>
        </div>
      </form>
    </div>

    <div class="row">
      <div class="col-md-6">
        <div class="row mb-5">
          <div class="col-md-6 mb-3 mb-md-0">
            <button class="btn btn-primary btn-lg py-3 btn-block"
onclick="window.location='/checkout'">Замовити</button>
          </div>
        </div>
      </div>
      <div class="col-md-6 pl-5">
        <div class="row justify-content-end">

```



```

    @include('inc.header')
</head>

<body>

    <div class="site-wrap">
        @include('inc.head', [
            'categories' => $categories
        ])
        <div class="bg-light py-3">
            <div class="container">
                <div class="row">
                    <div class="col-md-12 mb-0"><a href="/">Головна</a> <span class="mx-2
mb-0">/</span> <strong class="text-black">{{ $data['title'] }}</strong></div>
                </div>
            </div>
        </div>

        <div class="site-section">
            <div class="container">
                <div class="row">
                    <div class="col-md-6">
                        
                    </div>
                    <div class="col-md-6">
                        <h2 class="text-black">{{ $data['title'] }}</h2>
                        <p class="mb-4">{{ $data['description'] }}</p>
                        <p><strong class="text-primary h4">{{ $data['price']
}}рпН</strong></p>
                        <input id="index-product" type="hidden" value="{{ $data['id'] }}">
                        <div class="mb-4">
                            <div class="input-group mb-2" style="max-width: 120px;">
                                <div class="input-group-prepend">
                                    <button class="btn btn-outline-primary js-btn-minus"
type="button">&minus;</button>
                                </div>
                                <input id="quality" type="number" min="0" max="100" class="form-
control text-center" value="1" aria-label="Example text with button addon" aria-
describedby="button-addon1">
                                <div class="input-group-append">
                                    <button class="btn btn-outline-primary js-btn-plus"
type="button">&plus;</button>
                                </div>
                            </div>
                        </div>
                        <p>
                            <a href="#" class="buy-now btn btn-sm btn-primary"><span
class="icon icon-shopping_cart"></span></a>
                            <a href="#" class="wishlist-now btn btn-sm btn-primary"><span
class="icon icon-heart-o"></span></a>
                        </p>
                    </div>
                </div>
            </div>
        </div>

        <div class="site-section block-3 site-blocks-2 bg-light">
            <div class="container">
                <div class="row justify-content-center">
                    <div class="col-md-7 site-section-heading text-center pt-4">
                        <h2>Рекомендації</h2>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="col-md-12">
            <div class="nonloop-block-3 owl-carousel">
                @foreach ($recommendation as $item)
                    <div class="item">
                        <div class="block-4 text-center">
                            <figure class="block-4-image">
                                
                            </figure>
                            <div class="block-4-text p-4">
                                <h3><a href="/product/{{ $item['id'] }}">{{ $item['title']
}}</a></h3>
                                <p class="mb-0">{{ $item['category']['title'] }}</p>
                                <p class="text-primary font-weight-bold">{{ $item['price']
}}грн</p>
                            </div>
                        </div>
                    </div>
                @endforeach
            </div>
        </div>
    </div>
    @include('inc.footer2', [
        'categories' => $categories
    ])
</div>
@include('inc.footer')
<script>
    $("#quality").change(function() {
        if($("#quality").val() == '' || $("#quality").val() <= 0) {
            $("#quality").val(1)
        } else {
            $("#quality").val($("#quality").val())
        }
    })
</script>
</body>
</html>

```

checkout.blade.php

Шаблон сторінки замовлення.

```

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

<head>
    <title>Кошик</title>
    @include('inc.header')
</head>

<body>
    <div class="site-wrap">
        @include('inc.head', [
            'categories' => $categories
        ])
        <div class="bg-light py-3">
            <div class="container">
                <div class="row">
                    <div class="col-md-12 mb-0"><a href="/">Home</a> <span class="mx-2 mb-
0"></span> <a href="/cart">Cart</a> <span class="mx-2 mb-0"></span> <strong
class="text-black">Checkout</strong></div>
                </div>
            </div>
        </div>
    </div>

```



```

</div>

<div class="site-section">
  <div class="container">
    <form method="POST" action="/orders">
      <input type="hidden" name="users_id" value="{{ auth()->check() ?
auth()->user()->id : ' ' }}">
      <input type="hidden" name="sum" value="{{ $sum }}">
      <div class="row">
        <div class="col-md-6 mb-5 mb-md-0">
          <h2 class="h3 mb-3 text-black">Контактні дані</h2>
          <div class="p-3 p-lg-5 border">
            <div class="form-group row">
              <div class="col-md-6">
                <label for="name" class="text-black">Ім'я <span class="text-
danger">*</span></label>
                <input type="text" class="form-control" id="name"
name="name" value="{{ auth()->check() ? auth()->user()->name : ' ' }}" required>
                </div>
                <div class="col-md-6">
                  <label for="surname" class="text-black">Прізвище <span
class="text-danger">*</span></label>
                  <input type="text" class="form-control" id="surname"
name="surname" value="{{ auth()->check() ? auth()->user()->surname : ' ' }}"
required>
                </div>
              </div>
            </div>
            <div class="form-group row">
              <div class="col">
                <div class="form-group">
                  <label for="delivery" class="text-black">Спосіб доставки
<span class="text-danger">*</span></label>
                  <select id="delivery" class="form-control"
name="delivery_types_id" onchange="deliveryType()">
                    @foreach($delivery_types as $item)
                      <option value="{{ $item['id'] }}">{{ $item['title']
}}</option>
                    @endforeach
                  </select>
                </div>
              </div>
              <div class="col postal_zip">
                <label for="postal_zip" class="text-black">Відділення <span
class="text-danger">*</span></label>
                <input type="text" class="form-control" id="postal_zip"
name="postal_zip">
              </div>
            </div>
            <div class="form-group row">
              <div class="col-md-6">
                <label for="email" class="text-black">Email <span
class="text-danger">*</span></label>
                <input type="email" class="form-control" id="email"
name="email" value="{{ auth()->check() ? auth()->user()->email : ' ' }}"
required>
              </div>
              <div class="col-md-6">
                <label for="phone" class="text-black">Телефон <span
class="text-danger">*</span></label>
                <input type="text" class="form-control" id="phone"
name="phone" value="{{ auth()->check() ? auth()->user()->phone : ' ' }}"
required>
              </div>
            </div>
          </div>
        </div>
      </div>
    </form>
  </div>
</div>

```



```

<html dir="ltr" lang="en">

@include('inc.admin.head')

<link href="/admin/dist/chartist.min.css" rel="stylesheet">

<body>
  <div id="main-wrapper" data-navbarbg="skin6" data-theme="light" data-
  layout="vertical" data-sidebartype="full" data-boxed-layout="full">
    @include('inc.admin.header')
    <div class="page-wrapper">
      <div class="container-fluid">
        <div class="row">
          <div class="col-12">
            <div class="card">
              <div class="card-body">
                <h4 class="card-title">Статистика продаж</h4>
                <div class="sales ct-charts mt-3"></div>
              </div>
            </div>
            <div class="card">
              <div class="card-body">
                <h4 class="card-title">Останні замовлення</h4>
              </div>
              <div class="table-responsive">
                <table class="table table-hover">
                  <thead>
                    <tr>
                      <th class="border-top-0">ID</th>
                      <th class="border-top-0">ПІБ</th>
                      <th class="border-top-0">Телефон</th>
                      <th class="border-top-0">Email</th>
                      <th class="border-top-0">Сума</th>
                      <th class="border-top-0">Дата</th>
                      <th class="border-top-0">Статус</th>
                      <th class="border-top-0"></th>
                    </tr>
                  </thead>
                  <tbody>
                    @foreach($orders as $item)
                      <tr>
                        <td class="txt-overflow">{{ $item['id'] }}</td>
                        <td class="txt-overflow">{{ $item['name'] }} {{ $item['surname'] }}</td>
                        <td class="txt-overflow">{{ $item['phone'] }}</td>
                        <td class="txt-overflow">{{ $item['email'] }}</td>
                        <td class="txt-overflow">{{ $item['sum'] }}</td>
                        <td class="txt-overflow">{{ $item['created_at'] }}</td>
                        <td>
                          @if($item['order_status_id'] == 1)
                            <span class="label label-primary label-rounded">Новий</span>
                          @endif
                          @if($item['order_status_id'] == 2)
                            <span class="label label-success label-rounded">Виконаний</span>
                          @endif
                        </td>
                      </tr>
                    @endforeach
                  </tbody>
                </table>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

3)
label-rounded">Відхилений</span>
width="190">
}}/edit" class="btn btn-dark text-white">
    @endif
    @if($item['order_status_id'] ==
    <span class="label label-danger
    @endif
    </td>
    <td class="txt-oflo text-end"
    <a href="/orders/{{ $item['id']
    <i class="mdi mdi-eye"></i>
    </a>
    </td>
    </tr>
    @endforeach
    </tbody>
    </table>
    </div>
    </div>
    </div>
    </div>
    </div>
    </div>
    </div>
</div>
<include('inc.admin.footer')
<script src="/admin/dist/chartist.min.js"></script>
<script src="/admin/dist/chartist-plugin-tooltip.min.js"></script>
<script>
    $(function() {
        "use strict";
        var chart = new Chartist.Line('.sales', {
            labels: [<? = implode(", ", $statistic['labels']) ?>],
            series: [[<? = implode(", ", $statistic['series']) ?>]]
        }, {
            low: 0,
            high: 8,
            showArea: true,
            fullWidth: true,
            plugins: [
                Chartist.plugins.tooltip()
            ],
            axisY: {
                onlyInteger: false,
                scaleMinSpace: 40,
                offset: 20,
            },
        });
        var chart = [chart];
    });
</script>
</body>
</html>

```

account.blade.php

```

Шаблон сторінки профілю користувача.
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Профіль користувача</title>
    @include('inc.header')
</head>

```

```

<body>
  <div class="site-wrap">
    @include('inc.head', [
      'categories' => $categories
    ])
    <div class="bg-light py-3">
      <div class="container">
        <div class="row">
          <div class="col-md-12 mb-0"><a href="/">Головна</a> <span
class="mx-2 mb-0"/></span> <strong class="text-black">Профіль
користувача</strong></div>
        </div>
      </div>
    </div>

    <div class="site-section">
      <div class="container">
        <div class="row">
          <div class="col-md-6">
            <h2 class="h3 mb-3 text-black">Персональні дані</h2>
            <form action="/account" method="post">
              <div class="p-3 p-lg-5 border">
                <div class="row">
                  <div class="col">
                    <div class="form-group row">
                      <div class="col-md-12">
                        <label for="name" class="text-
black">Ім'я <span class="text-danger">*</span></label>
                        <input type="text" class="form-
control" id="name" name="name" value="{{ $user['name'] }}" required>
                      </div>
                    </div>
                  </div>
                  <div class="col">
                    <div class="form-group row">
                      <div class="col-md-12">
                        <label for="surname"
class="text-black">Прізвище <span class="text-danger">*</span></label>
                        <input type="text" class="form-
control" id="surname" name="surname" value="{{ $user['surname'] }}" required>
                      </div>
                    </div>
                  </div>
                </div>
                <div class="form-group row">
                  <div class="col-md-12">
                    <label for="email" class="text-
black">Email <span class="text-danger">*</span></label>
                    <input type="email" class="form-control"
id="email" name="email" value="{{ $user['email'] }}" required>
                  </div>
                </div>
                <div class="form-group row">
                  <div class="col-md-12">
                    <label for="phone" class="text-
black">Телефон <span class="text-danger">*</span></label>
                    <input type="text" class="form-control"
id="phone" name="phone" value="{{ $user['phone'] }}" required>
                  </div>
                </div>
                <div class="form-group row">

```

```

        <div class="col-md-12">
            <label for="password" class="text-
black">Новий пароль <span class="text-danger">*</span></label>
            <input type="password" class="form-
control" id="password" name="password">
        </div>
    </div>

    <div class="form-group row">
        <div class="col-md-12">
            <label for="repeat_password"
class="text-black">Повторіть пароль <span class="text-danger">*</span></label>
            <input type="password" class="form-
control" id="repeat_password" name="repeat_password">
        </div>
    </div>

    <div class="form-group row">
        <div class="col-lg-12">
            <input type="submit" class="btn btn-
primary btn-lg btn-block" value="Зберегти">
        </div>
    </div>
</form>
</div>
<div class="col-md-6 ml-auto">
    <h2 class="h3 mb-3 text-black">Історія замовлень</h2>
    @foreach($orders as $item)
    <div class="p-4 border mb-3">
        <div class="row">
            <div class="col-3">
                
            </div>
            <div class="col-9">
                <span class="d-block text-primary h6 text-
uppercase">{{ $item['product']['title'] }} - {{ $item['quantity'] }}x</span>
                <p class="mb-0">{{ $item['product']['price']
}}грн</p>
                <p class="mb-0">{{ $item['created_at']
}}</p>
            </div>
        </div>
    </div>
    @endforeach
    @if(count($orders) == 0)
    <div class="py-5 text-center">
        
    </div>
    @endif
</div>
</div>
</div>
<div class="col-md-6 ml-auto">
    @include('inc.footer2', [
        'categories' => $categories
    ])
</div>
<div class="col-md-6 ml-auto">
    @include('inc.footer')
</div>
</body>
</html>

```


Додаток Г

ІМА :: 2023

СЕКЦІЯ 2: Інформаційні технології проєктування

Web-додаток підтримки продажу декоративних товарів для інтер'єру «Mondo»

Самочкіна Т.В., студент ІТ-91; Парфененко Ю.В., доцент
Сумський державний університет, м. Суми, Україна

У сучасному світі, де все більше людей віддає перевагу онлайн шопінгу, web-додаток магазину є дуже актуальним. Онлайн-магазини дозволяють покупцям здійснювати покупки з будь-якого місця з доступом до інтернету, забезпечуючи зручність та доступність. Web-додаток магазину також може дозволяти покупцям швидко та зручно знайти потрібні товари, скориставшись фільтрами та іншими зручностями пошуку. Крім того, web-додатки магазину забезпечують можливість зберігати та обробляти великі обсяги даних, що дозволяє аналізувати поведінку покупців, відстежувати їхній вибір товарів та здійснювати персоналізовану рекламу і маркетингові акції.

Усе це робить web-додаток магазину необхідним для сучасного бізнесу, який бажає залишатися конкурентоспроможним та надавати своїм клієнтам найкращі сервіси та можливості для зручного та безпечного онлайн-шопінгу.

Метою роботи є розробка багатофункціонального web-додатку підтримки продажу декоративних товарів для інтер'єру «Mondo» зі зручним та інтуїтивно зрозумілим інтерфейсом, який забезпечить належну організацію компанії за рахунок автоматизації деяких робочих процесів.

Для реалізації даного web-додатку було обрано такі технології як мова розмітки гіпертексту HTML, каскадні таблиці стилів CSS і скриптову мову програмування JavaScript. Для реалізації бази даних розроблюваного програмного продукту було обрано СКБД MySQL. Для реалізації бекенду використана мова програмування PHP.

Основними функціями розробленого web-додатку є перегляд інформації про товари, пошук товару у web-додатку, можливість оформлення онлайн-замовлень за допомогою форми.

Практична значимість розробленого web-додатку полягає у підтримці продажу декоративних товарів для інтер'єру, автоматизації процесів формування замовлень.