

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

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко
підпис

«__» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Інтернет-магазин меблів

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 408.22010106

Здобувачка

_____ В. Ю. Дзина
підпис

«__» _____ 2024 р.

Керівник канд. техн. наук, доцент

_____ Є. О. Давиденко
підпис

«__» _____ 2024 р.

Консультант канд. техн. наук, доцент

_____ А. О. Алексеева
підпис

«__» _____ 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Зав. кафедри _____ Є. О. Давиденко

« _____ » _____ 2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи бакалавра

Видано здобувачці групи 408 факультету комп'ютерних наук

Дзина Влада Юріївна

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

Інтернет-магазин меблів

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

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

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

Очікуваним результатом є вебзастосунок інтернет-магазину меблів

4. Перелік питань, що підлягають розробці:

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

5. Перелік графічних матеріалів

Презентація

6. Завдання до спеціальної частини

Аналіз охорони праці на робочих місцях фахівців з інформаційних технологій

7. Консультанти:

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

Керівник роботи канд. техн. наук, доцент Давиденко Євген Олександрович

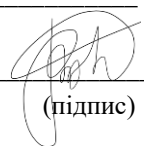
(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Дзина Влада Юріївна

(прізвище, ім'я, по батькові)



(підпис)

Дата видачі завдання « _____ » _____ 2024р.


КАЛЕНДАРНИЙ ПЛАН

виконання бакалаврської кваліфікаційної роботи

Тема: Інтернет-магазин меблів

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	20.02.2024	23.02.2024	Виконано
2.	Огляд літератури за темою роботи	24.02.2024	26.02.2024	Виконано
3.	Складання календарного плану КРБ	26.02.2024	27.02.2024	Виконано
4.	Аналіз предметної області	28.02.2024	01.03.2024	Виконано
5.	Розробка проєктних рішень	04.03.2024	10.03.2024	Виконано
6.	Моделювання та конструювання ПЗ	22.03.2024	26.03.2024	Виконано
7.	Кодування ПЗ, тестування та апробація розробленого ПЗ, аналіз результатів тестування	28.03.2024	22.05.2024	Виконано
8.	Розробка спеціальної частини з охорони праці	23.05.2024	25.05.2024	Виконано
9.	Оформлення КРБ та презентації	26.05.2024	27.05.2024	Виконано
10.	Відгук керівника КРБ	28.05.2024	29.05.2024	Виконано
11.	Попередній захист	03.06.2024	05.06.2024	Виконано
12.	Рецензування	13.06.2024	14.06.2024	Виконано
13.	Завершення оформлення КРБ та презентації	17.06.2024	21.06.2024	Виконано
14.	Захист кваліфікаційної роботи	24.05.2024	27.05.2024	Виконано

Розробила здобувачка Дзина Влада Юріївна
(прізвище, ім'я, по батькові)



(підпис)

«04» лютого 2024р.

Керівник роботи канд. техн. наук, доцент Давиденко Євген Олександрович
(посада, прізвище, ім'я, по батькові)



(підпис)

«04» лютого 2024р р.

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Інтернет-магазин меблів»

Здобувачка 408 гр.: Дзина Влада Юрійвна

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

Актуальність теми даного проєкту обумовлена стрімким розвитком сучасних технологій та постійними змінами в уподобаннях споживачів у сфері електронної комерції. Зростання популярності Інтернету та зручність його використання створюють попит на вебзастосунки інтернет-магазинів, які надають користувачам зручний та швидкий інструмент для пошуку та придбання товарів.

Вебзастосунки стають не лише зручними, але й ефективними інструментами для споживачів, що розширює можливості бізнесу та забезпечує конкурентні переваги.

Проєкт присвячений розробці вебзастосунку інтернет-магазину меблів, який забезпечує користувачам зручний та ефективний інструмент для пошуку та придбання меблів онлайн.

Об'єктом роботи є процес створення вебзастосунку інтернет-магазину з використанням сучасних технологій.

Предметом роботи є інструменти розробки сучасних вебзастосунків та технологій, які дозволяють реалізувати інтернет-магазин.

Метою кваліфікаційної роботи є розробка вебзастосунку для розширення аудиторії та залучення нових клієнтів до магазину через Інтернет.

Для досягнення мети необхідно виконати наступні завдання:

- провести аналіз сфери продажу меблів та сервісів для їх реалізації через мережу Інтернет;
- обґрунтувати вибір технологій та інструментальних засобів розробки інтернет-магазину;
- моделювання та проєктування програмного забезпечення;

– розробити дизайн інтернет-магазину з врахуванням усіх потреб користувача;

– розробити функціонал та сторінки інтернет-магазину.

Структура кваліфікаційної роботи бакалавра включає вступ, три розділи, висновки та перелік джерел посилань.

У вступі визначається актуальність обраної теми, формулюється основна мета, надається короткий огляд поставленої задачі, а також надається опис предмету та методів дослідження.

У першому розділі досліджено предметну область, проведено аналіз існуючих аналогів та розглянуто особливості їх роботи. В результаті цього аналізу було сформульовано завдання для виконання кваліфікаційної роботи бакалавра.

У другому розділі розглядаються проектування інтернет-магазину меблів. Проектуються такі діаграми як: діаграма варіантів використання, діаграма класів, діаграма послідовностей дій системи та діаграма станів та переходів.

У третьому розділі описується конструювання програмного забезпечення. Описуються технології розробки. Моделюється структура бази даних. А також проектується користувацький інтерфейс.

У четвертому розділі ідеться про програмну реалізацію. Спочатку реалізується клієнтська частина, а потім адміністративна панель.

Кваліфікаційна робота бакалавра викладена на 60 сторінок, вона містить 4 розділи, 37 ілюстрацій, 20 джерел в переліку посилань.

Ключові слова: *інтернет-магазин меблів, інтернет-магазин, вебзастосунок, React, Laravel.*

ABSTRACT

of the Bachelor's Thesis

"Online furniture store"

Student of group 408: Dzyna Vlada Yuryivna

Supervisor: candidate of technical Sciences, associate professor

Davydenko Y. O.

The relevance of the topic of this project is due to the rapid development of modern technologies and constant changes in consumer preferences in the field of electronic commerce. The growing popularity of the Internet and its ease of use create a demand for web applications of online stores that provide users with a convenient and fast tool for searching and purchasing goods. Web applications are becoming not only convenient, but also effective tools for consumers, which expands business opportunities and provides competitive advantages.

This project is dedicated to the development of a web application for an online furniture store, which provides users with a convenient and effective tool for searching and purchasing furniture online.

The subject of the work is the tools for developing modern web applications and technologies that allow you to implement an online store.

The purpose of the qualification work is to develop a web application to expand the audience and attract new customers to the store via the Internet.

To achieve the goal, it is necessary to complete the following tasks:

- to conduct an analysis of the sphere of sale of furniture and services for their implementation via the Internet;
- justify the choice of technologies and tools for developing an online store;
- software modeling and design;
- develop the design of the online store taking into account all the needs of the user;
- develop functionality and pages of the online store.

The structure of the bachelor's thesis includes an introduction, three chapters, conclusions and a list of reference sources.

The introduction determines the relevance of the chosen topic, formulates the main goal, provides a brief overview of the task, and also provides a description of the subject and research methods.

In the first chapter, the subject area was investigated, the analysis of existing analogues was carried out, and the peculiarities of their work were considered. As a result of this analysis, the tasks for the bachelor's qualification work were formulated.

The second section deals with the design of an online furniture store. The following diagrams are designed: a diagram of use cases, a diagram of classes, a diagram of sequences of system actions, and a diagram of states and transitions.

The third section describes software development. Development technologies are described. The structure of the database is modeled. A user interface is also being designed.

The fourth chapter deals with software implementation. First, the client part is implemented, and then the administrative panel. The bachelor's qualification work is laid out on 60 pages, it contains 4 sections, 37 illustrations, 20 sources in the list of references.

Keywords: *online furniture store, online store, web application, React, Laravel.*

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	6
1.1 Аналіз предметної області.....	6
1.2 Аналіз готових рішень	7
1.2.1 Аналіз Bedbathandbeyond	7
1.2.2 Аналіз Wayfair	8
1.2.3 Аналіз Ashley Furniture	10
1.3 Постановка задачі.....	11
1.4 Специфікація вимог до програмного забезпечення.....	11
Висновки до розділу 1	14
2 ПРОЄКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ МЕБЛІВ	15
2.1 Розробка діаграми варіантів використання	15
2.2 Розробка діаграми класів.....	16
2.3 Розробка діаграми послідовностей дій системи	18
2.4 Розробка діаграми станів та переходів.....	18
Висновки до розділу 2	19
3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	20
3.1 Опис технологій розробки.....	20
3.2 Моделювання бази даних	31
3.3 Розробка користувацького інтерфейсу	41
Висновки до розділу 3	49
4 ПРОГРАМНА РЕАЛІЗАЦІЯ	50
4.1 Реалізація клієнтської частини	50
4.2 Реалізація адмін панелі.....	55

	3
Висновки до розділу 4	57
ВИСНОВКИ.....	58
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	59

ВСТУП

Актуальність теми. У сучасному світі спостерігається значне збільшення інтересу до онлайн-покупок у різних сферах діяльності, включаючи придбання меблів. Зручність, доступність та широкий вибір товарів роблять інтернет-магазини привабливими для користувачів, особливо для тих, хто має обмежений час для фізичних візитів у магазини.

Під час пандемії COVID-19 було помічено збільшення попиту на меблі для дома, оскільки багато людей почали працювати з дому та проводити більше часу вдома. Це підвищило інтерес до онлайн-покупок меблів, а також підкреслило важливість зручних та ефективних інтернет-платформ для покупок.

В умовах війни та геополітичної нестабільності роль електронної комерції, зокрема інтернет-магазинів меблів, стає особливо важливою. Багато людей, які переживають війну або її наслідки, можуть шукати засоби для покращення свого життя та створення комфортного середовища для себе та своїх сімей. Онлайн-платформи меблів можуть стати важливим ресурсом для них у виборі необхідних предметів меблів для їхніх домівок чи тимчасових притулків. Тому розробка та вдосконалення таких інтернет-платформ є актуальним завданням для бізнесу в наші часи.

Практичне значення розроблюваного вебзастосунку полягає в тому, що такий застосунок створить зручну та ефективну платформу для клієнтів для ознайомлення з асортиментом меблів, що пропонується компанією. Надання повноцінної та стислої інформації про товари допоможе споживачам швидко зробити обдуманий вибір.

Об'єктом роботи є процес створення вебзастосунку інтернет-магазину з використанням сучасних технологій.

Предметом роботи є інструменти розробки сучасних вебзастосунків та технологій, які дозволяють реалізувати інтернет магазин.

Метою кваліфікаційної роботи є розробка вебзастосунку для розширення аудиторії та залучення нових клієнтів до магазину через Інтернет.

Для досягнення мети необхідно виконати наступні **завдання**:

- провести аналіз сфери продажу меблів та сервісів для їх реалізації через мережу Інтернет;
- обґрунтувати вибір технологій та інструментальних засобів розробки інтернет-магазину;
- моделювання та проєктування програмного забезпечення;
- розробити дизайн інтернет-магазину з врахуванням усіх потреб користувача;
- розробити функціонал та сторінки інтернет-магазину;
- виконати програмну реалізацію та тестування інтернет-магазину.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

Інтернет-магазини меблів представляють собою важливий сегмент електронної комерції, що набуває все більшої популярності серед споживачів. Зростаючий інтерес до цього типу торгівлі відображає не лише зміни в способах споживання, але й впливає на ринкові тенденції, конкуренцію та технологічний прогрес.

Попит на меблі постійно зростає, особливо з урахуванням розвитку житлової сфери та зміни в дизайні інтер'єрів. Люди все більше шукають унікальні та стильні рішення для свого житла, що робить інтернет-магазини меблів важливими платформами для їх задоволення потреб споживачів.

Конкуренція в галузі інтернет-магазинів меблів досить висока. Багато компаній вже присутні на ринку, проте завжди є місце для нових гравців, які пропонують інноваційні підходи до продажу меблів та послуг.

Технологічний прогрес впливає на розвиток інтернет-магазинів меблів у багатьох аспектах. Це включає використання віртуальної реальності для візуалізації меблів в просторі кімнати, розробку мобільних застосунків для зручного шопінгу, а також використання штучного інтелекту для персоналізованих рекомендацій та покращення користувацького досвіду.

Клієнтський досвід в інтернет-магазинах меблів відіграє ключову роль у залученні та утриманні клієнтів. Це включає в себе швидкий та зручний пошук товарів, простий процес замовлення, персоналізовані рекомендації, а також високий рівень обслуговування та підтримки клієнтів.

Ефективна логістика та доставка меблів є важливим елементом успішного інтернет-магазину. Клієнти очікують швидкої та безпечної доставки, яка забезпечить їм отримання замовлених товарів у найкоротший термін та у найкращому стані.

Розуміння та аналіз предметної області інтернет-магазинів меблів є важливим етапом у розробці та управлінні таким бізнесом. Постійне вдосконалення та адаптація до змін в ринкових умовах та технологічному прогресу допомагають інтернет-магазинам меблів зберігати конкурентоспроможність і задовольняти потреби сучасних споживачів [4].

1.2 Аналіз готових рішень

На сучасний момент на ринку вебзастосунків існує багато застосунків, що забезпечують продаж різноманітних меблів по містах. Тому перед початком розробки власного вебзастосунку був проведений аналіз вже існуючих систем.

Виявлено наступні критерії:

- назва;
- розробник (дистриб'ютор);
- архітектура (desktop application, client-server, 3tier web application);
- мова реалізації;
- перелік функцій, характеристик;
- аналіз переваг та недоліків даного ПЗ;
- джерело інформації (вебсайт).

Розглянуто наступні вебсервіси:

- Overstock;
- Wayfair;
- Ashley Furniture.

1.2.1 Аналіз Bedbathandbeyond [7]

Назва: Bedbathandbeyond

Виробник: Overstock.com, Inc.

Архітектура: client-server

Мова реалізації: JavaScript (React.js, Node.js), Python (Django), Go

Основні функції:

- широкий асортимент меблів та предметів декору з різних брендів та виробників;
- розширені можливості персоналізації та порівняння товарів;
- платформа для продажу використаних або відновлених меблів з вигідними цінами;
- кошук;
- особистий кабінет;
- обрані.

Переваги:

- великий вибір меблів;
- зручна система порівняння.

Недоліки:

- відсутність фільтрів та обцій сортування.

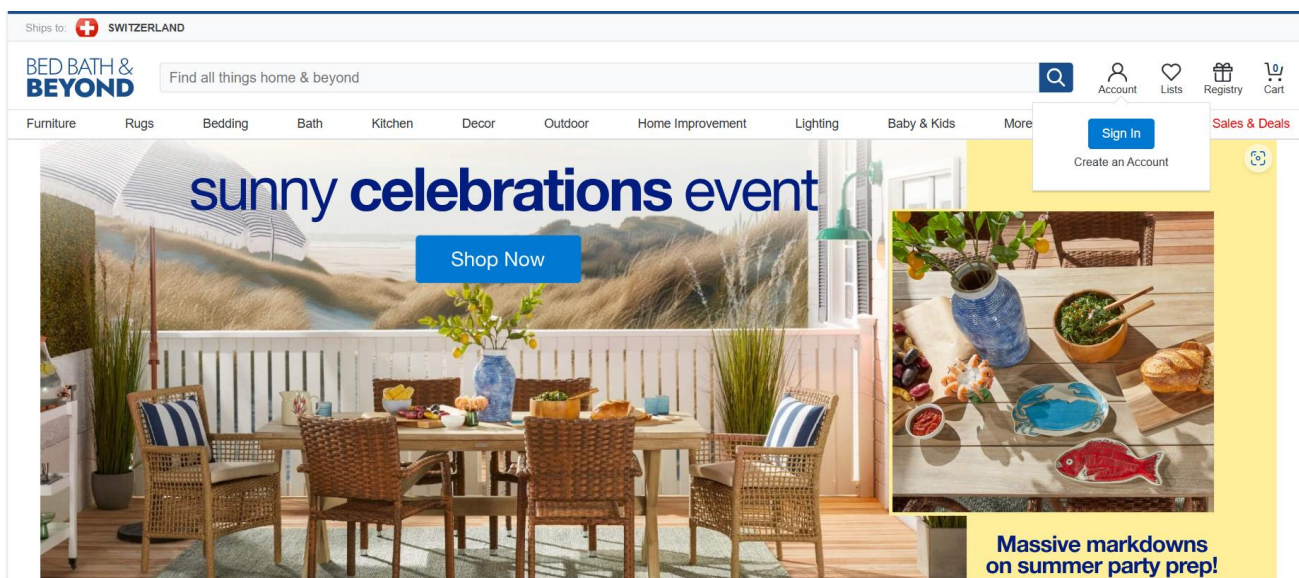


Рисунок 1.1 – Вигляд аналогу Bedbathandbeyond

Таким чином представлено вигляд аналогу Bedbathandbeyond.

1.2.2 Аналіз Wayfair [8]

Назва: Wayfair

Виробник: Wayfair LLC

Архітектура: client-server

Мова реалізації: JavaScript (React.js, Node.js), Python (Django), Ruby (Ruby on Rails)

Основні функції:

- широкий асортимент меблів та предметів декору від різних виробників;
- персоналізовані рекомендації товарів на основі попередніх покупок та переглядів;

- простий та зручний пошук товарів за різними параметрами;
- користувацька система оцінок та відгуків для кожного товару;
- кошик;
- особистий кабінет.

Переваги:

- великий вибір товарів;
- зручний пошук та фільтрація;
- персоналізовані рекомендації.

Недоліки:

- можливість перенасичення асортименту;
- завелика кількість рекламних елементів.

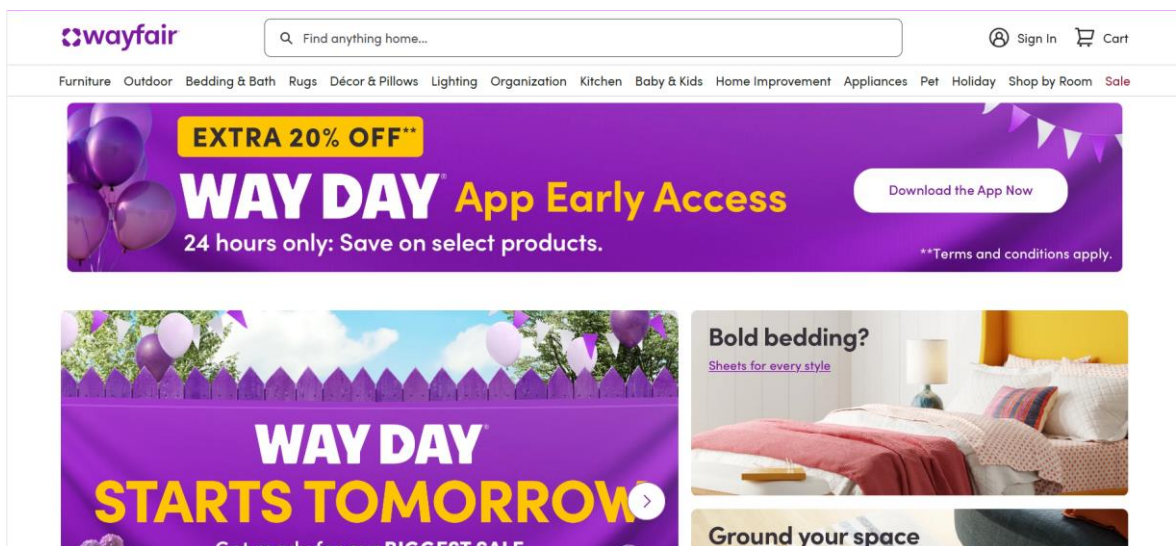


Рисунок 1.2 – Вигляд аналогу Wayfair

Таким чином представлено вигляд аналогу Wayfair.

1.2.3 Аналіз Ashley Furniture [6]

Назва: Ashley Furniture

Виробник: Ashley Furniture Industries, Inc.

Архітектура: client-server

Мова реалізації: JavaScript (Vue.js), PHP (Laravel)

Основні функції:

- можливість замовлення індивідуальних проєктів від власних дизайнерів;
- платформа для отримання консультацій та рекомендацій стосовно вибору меблів;
- система відгуків та оцінок товарів для допомоги користувачам у прийнятті рішення;
- кошик;
- пошук.

Переваги:

- унікальний дизайн;
- індивідуальні проєкти;
- програма лояльності.

Недоліки:

- обмежений вибір порівняно з більшими конкурентами;
- відсутність відгуків.

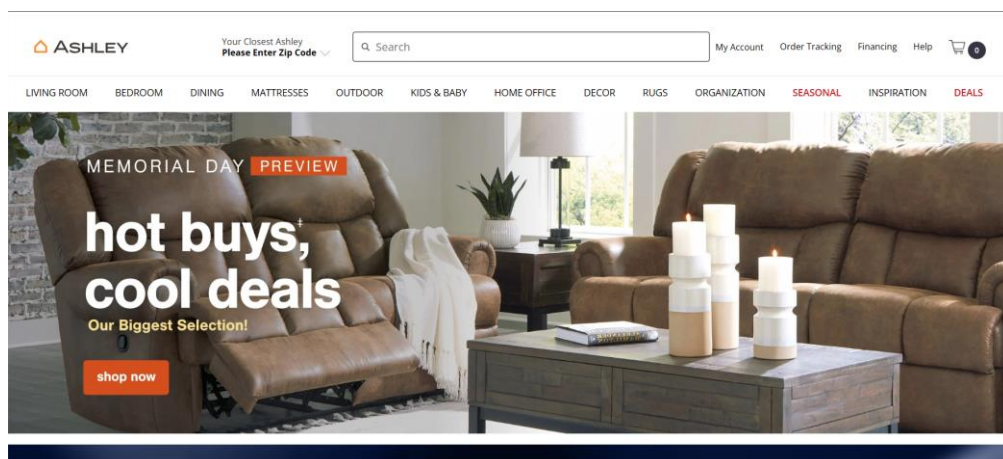


Рисунок 1.3 – Вигляд аналогу Ashley Furniture

Таким чином представлено вигляд аналогу Ashley Furniture. Визначено основні характеристики аналогових систем.

1.3 Постановка задачі

Потрібно розробити вебзастосунок, де користувач матиме змогу обрати потрібні меблі до свого помешкання, офісу чи просто приміщення швидко та інтуїтивно зрозуміло. Клієнтам будуть доступні наступні функції:

Для користувача:

- реєстрація користувачів;
- авторизація користувачів;
- перегляд товарів та колекцій;
- перегляд окремого товару та колекції;
- пошук товару;
- додавання товару у кошик;
- можливість залишити відгук;
- можливість використання фільтрації для полегшення пошуку потрібного товару;
- оформлення замовлення;
- здійснення платежів.

Для адміністратора:

- редагування, перегляд, видалення, додавання товарів та колекцій;
- підрахунок доходів;
- відстеження статусу замовлень.

1.4 Специфікація вимог до програмного забезпечення

ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

Призначення застосунку, для якої розробляється програмне забезпечення

Програмне забезпечення для інтернет-магазину меблів має на меті забезпечити ефективну та зручну платформу для продажу меблів через Інтернет.

Межі проєкту ПЗ

Кінцева дата завершення роботи над ПЗ – ...

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Сфера застосування інтернет-магазину меблів включає в себе електронну комерцію. Інтернет-магазин меблів дозволяє покупцям здійснювати покупки меблів онлайн, що робить процес зручним та доступним.

Характеристика користувачів

Адміністратор магазину для керування замовленнями, продуктами та колекціями.

Клієнти для пошуку необхідного товару, додавання у кошик, оформлення замовлення, оплати та відгуків.

Загальна структура та склад системи

Система складається з двох основних компонентів:

- клієнтська частина (вебзастосунок);
- серверна частина (база даних, вебсервер) забезпечує зберігання даних та обробку запитів: модуль бази даних, вебсервера.

Загальні обмеження

Головне обмеження – доступ до Інтернету.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Джерела і зміст вхідної інформації (даних)

В застосунку джерелом вхідної інформації є користувач. Користувач має вручну вводити такі дані: персональні дані, e-mail, пароль, адресу доставки, метод оплати.

Нормативно-довідкова інформація (класифікатори, довідники тощо)

Вимоги відсутні.

Вимоги до способів організації, збереження та ведення інформації

Для бази даних обрано MySQL.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Ноутбук з технічними характеристиками, що підтримує будь-який сучасний браузер.

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмної системи

Архітектура включає клієнтську частину, серверну частину та базу даних.

Системне програмне забезпечення

Для розробки вебзастосунку для back-end частини обрано використовувати фреймворк Laravel, а для front-end частини – React.

Мережне програмне забезпечення

ПЗ розроблено на операційній системі Windows 11. До програмних середовищ відносяться PhpStorm та WebStorm. Браузер – Google Chrome.

Програмне забезпечення ведення інформаційної бази

CRUD-операції виконуються через базу даних MySQL.

Мова розробки ПЗ

Мова розробки програмного забезпечення – PHP.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Інтерфейс користувача інтернет-магазину меблів має відповідати ряду вимог для забезпечення зручного та ефективного користувацького досвіду. До таких вимог відносяться простота використання, зручність навігації та інформативність.

Апаратний інтерфейс

Вебзастосунок має адаптивний інтерфейс.

Програмний інтерфейс

React – відкрита бібліотека JavaScript для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту вебсторінки.

Laravel – безкоштовний, з відкритим кодом PHP-фреймворк, призначений для розробки вебзастосунків відповідно до шаблону model view controller (MVC).

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

Інтернет-магазин меблів повинен бути доступний для будь-якого користувача, який хотів би здійснити покупку меблів.

Супроводжуваність

Супроводження вебзастосунку інтернет-магазину меблів не передбачається.

Переносимість

Вебзастосунок доступний на будь-якому браузері та операційній системі.

Продуктивність

Продуктивність ПЗ залежить від швидкості мережі Інтернет.

Надійність

Вхідні дані, що користувач передає, мають бути конфіденційними, і не повинна існувати можливість їх отримання будь-яким іншим способом. Користувач має доступ лише до власних даних та тільки після авторизації в системі.

Висновки до розділу 1

В першому розділі кваліфікаційної роботи бакалавра описано предметну область кваліфікаційної роботи.

Описано функціонал вебзастосунку, що спрямований на полегшення вибору меблів та оформлення замовлення.

В аналізі існуючих систем також виявлено ключові аспекти та вдалі рішення, які можна врахувати під час розробки власної платформи. Проаналізовані Overstock, Wayfair та Ashley Furniture як потенційні конкуренти на ринку продажу меблів.

Зазначено специфікації вимог до програмного забезпечення.

2 ПРОЄКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ МЕБЛІВ

2.1 Розробка діаграми варіантів використання

Діаграма варіантів використання (Use Case Diagram) – це вид діаграми Unified Modeling Language (UML), який використовується для візуалізації взаємодії між акторами (користувачами чи зовнішніми системами) та функціональними вимогами системи. Ця діаграма допомагає ідентифікувати та описувати всі можливі способи використання системи або програми з точки зору користувачів [2].

Діаграма варіантів використання допомагає створити загальне розуміння функціональних вимог до системи, ідентифікувати основні функціональність та сценарії взаємодії з користувачами. Вона є важливим інструментом при аналізі вимог та проєктуванні системи.

Діаграма варіантів використання інтернет-магазину, представлено на рисунку 2.1.

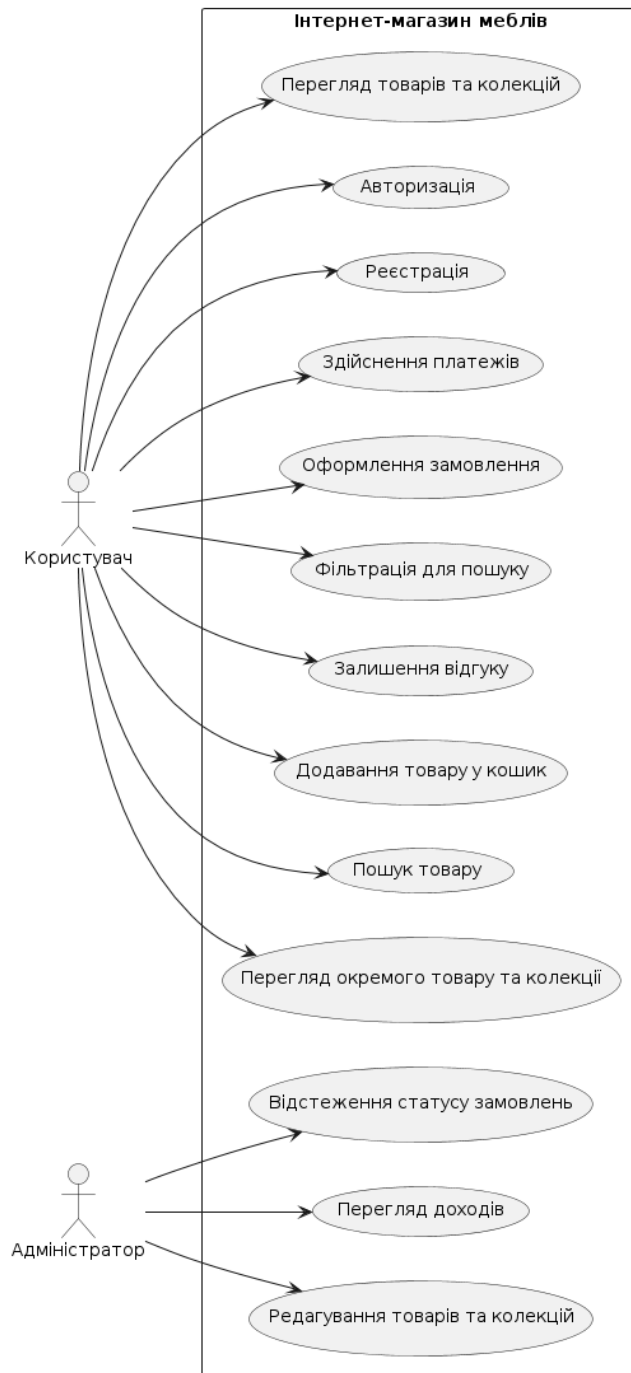


Рисунок 2.1 – Діаграма варіантів використання інтернет-магазину меблів

Таким чином представлено діаграму варіантів використання інтернет-магазину меблів.

2.2 Розробка діаграми класів

Діаграма класів – це вид діаграми Unified Modeling Language (UML), який використовується для візуалізації структури класів і їх взаємозв'язків у системі

програмного забезпечення. Вона надає графічне представлення класів, їх атрибутів, методів та спадковості.

Діаграми класів допомагають розуміти структуру програми, взаємозв'язки між класами та її архітектуру. Вони є важливим інструментом при проєктуванні та розробці програмного забезпечення.

Діаграму класів для інтернет-магазину меблів, представлено на рисунку 2.2.

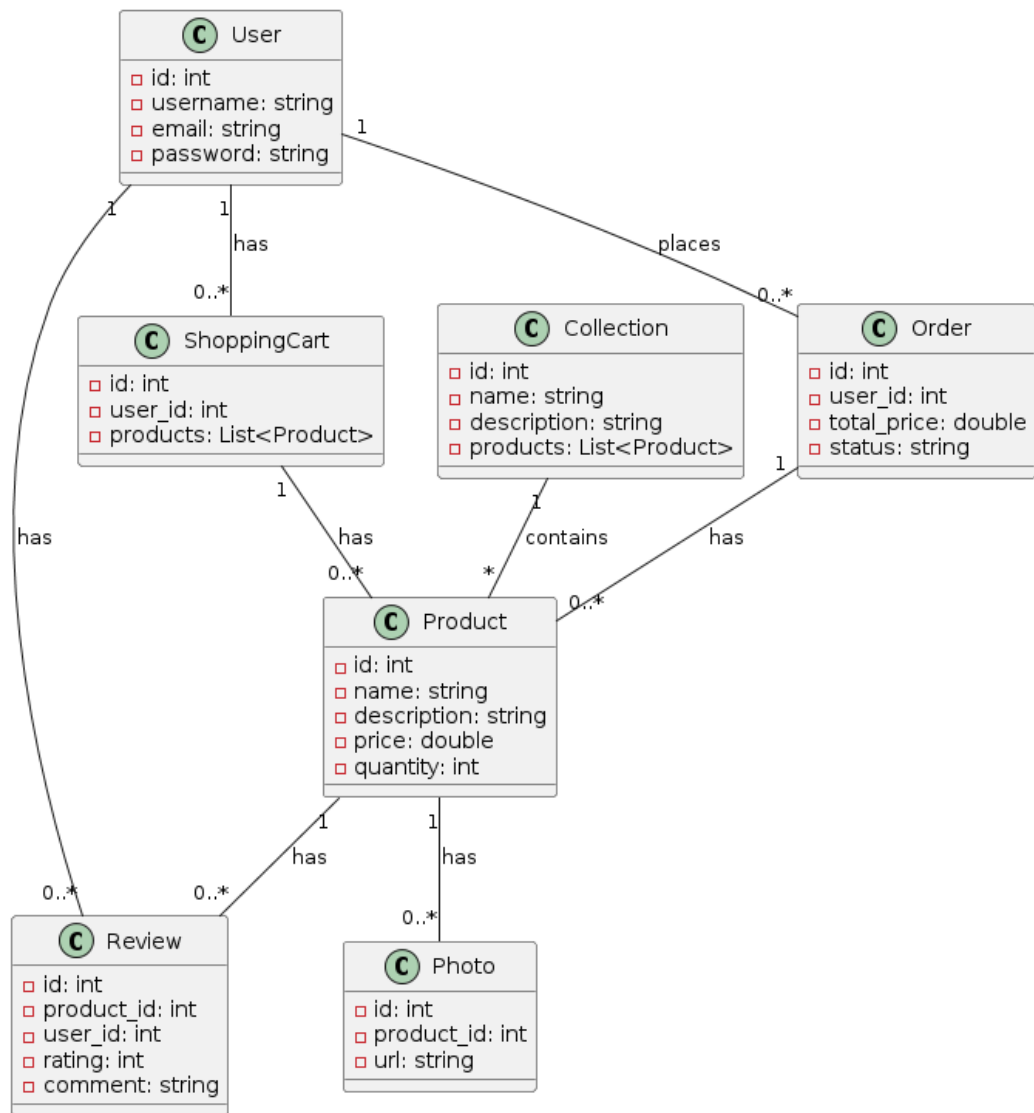


Рисунок 2.2 – Діаграма класів інтернет-магазину меблів

Таким чином розоблено діаграму класів інтернет-магазину меблів.

2.3 Розробка діаграми послідовностей дій системи

Діаграма послідовності – це вид діаграми UML (Unified Modeling Language), який використовується для візуалізації взаємодії між об'єктами у системі в рамках певного сценарію або послідовності подій. Вона показує послідовність викликів методів або повідомлень між об'єктами в рамках певного випадку використання або функціональної операції.

Діаграми послідовності допомагають розуміти порядок подій та взаємодію об'єктів у складних системах, що допомагає в їх аналізі, розробці та тестуванні.

Діаграма послідовності системи при додаванні товару до кошику представлена на рисунку 2.3.

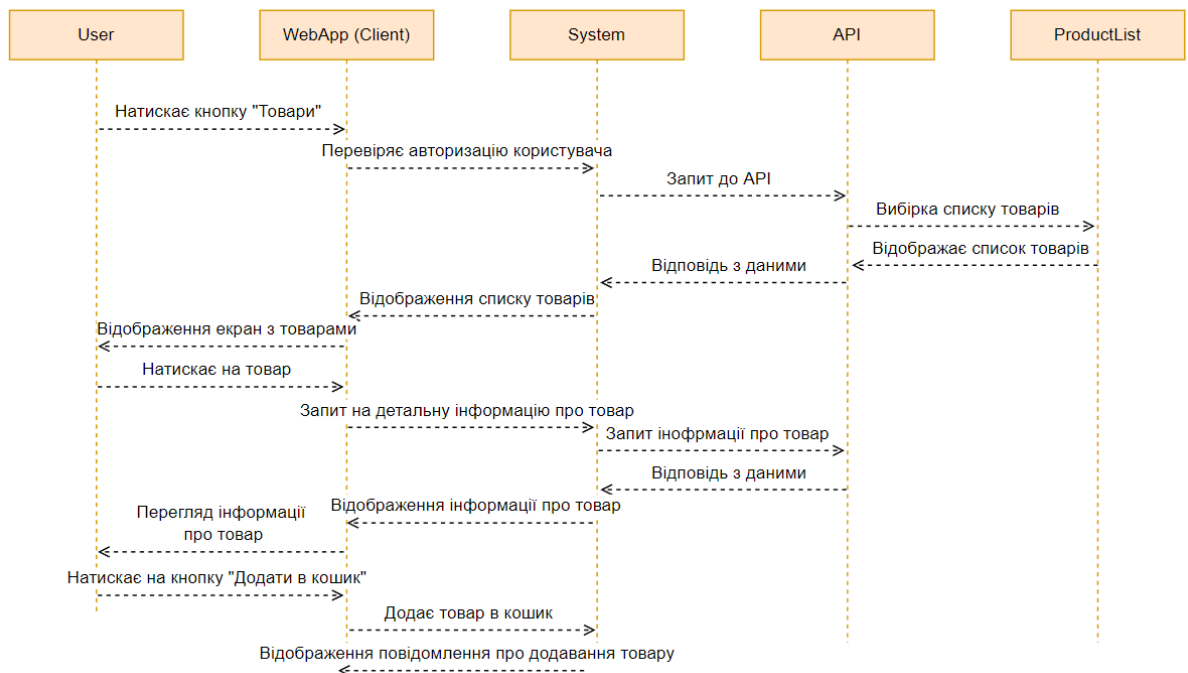


Рисунок 2.3 – Діаграма послідовності системи при додаванні товару до кошику

Таким чином представлено діаграму послідовності системи при додавання товару до кошику.

2.4 Розробка діаграми станів та переходів

Діаграма станів та переходів (State Diagram) є важливим інструментом в UML (Unified Modeling Language) для моделювання поведінки об'єкта протягом

його життєвого циклу. Вона відображає всі можливі стани об'єкта та переходи між цими станами, що викликані подіями.

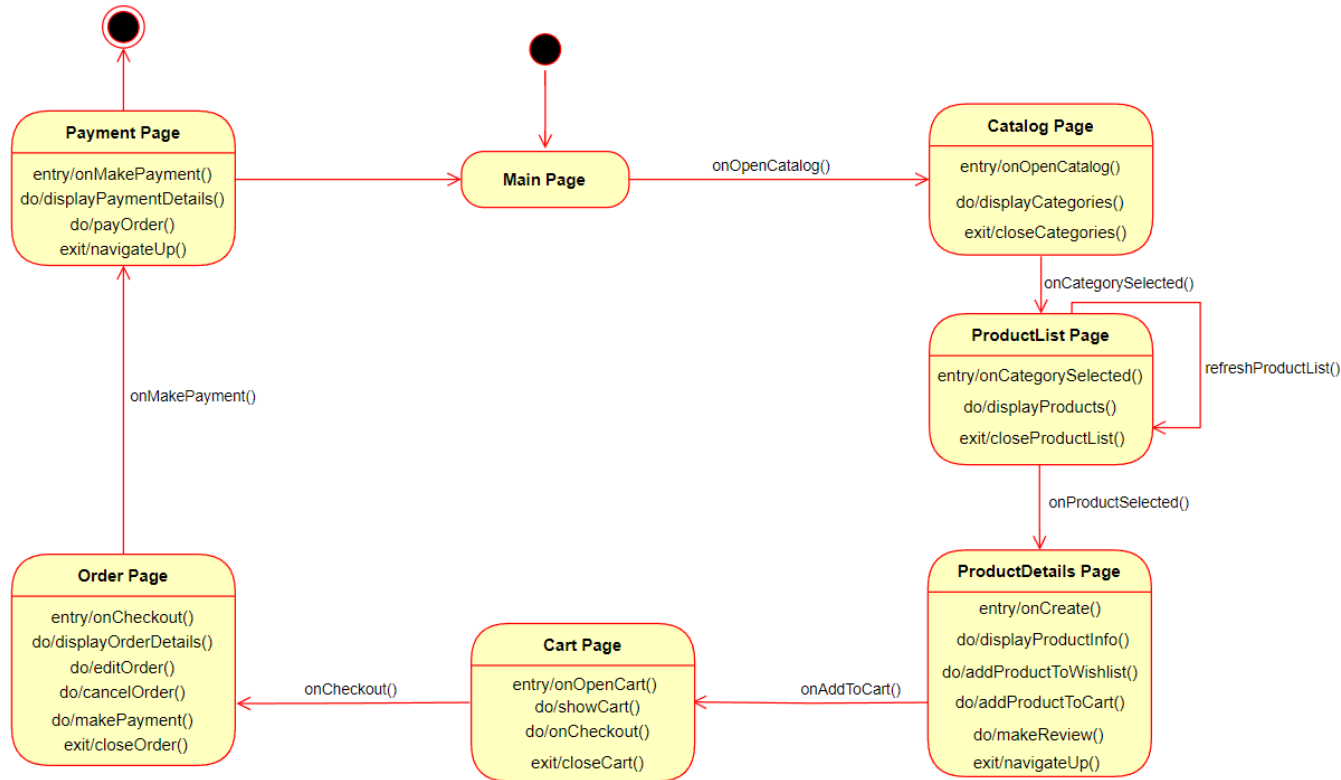


Рисунок 2.4 – Діаграма станів для створення та оплати замовлення

Таким чином представлено діаграму станів для створення та оплати замовлення.

Висновки до розділу 2

У другому розділі кваліфікаційної роботи бакалавра спроектовано інтернет-магазин меблів. Розроблено наступні моделі та діаграми:

- діаграму варіантів використання системи;
- діаграму класів системи;
- діаграму послідовностей оформлення замовлення;
- діаграму станів для створення та оплати замовлення.

3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Опис технологій розробки

PHP, Laravel, JavaScript, React, і MySQL – це важливі компоненти сучасної веброзробки, кожна з яких відіграє ключову роль у створенні інтерактивних, динамічних та ефективних вебзастосунків. Кожна з цих технологій має свої унікальні характеристики та області застосування, які визначають їх використання в проектах різних масштабів та складності.

PHP, або Hypertext Preprocessor, є однією з найбільш поширених серверних мов програмування, яка спеціально створена для розробки вебзастосунків. Вона була розроблена Расмусом Лердорфом у 1994 році як простий скрипт для відстеження відвідувань його онлайн-резюме. З того часу PHP значно еволюціонував і став однією з найбільш використовуваних мов програмування для створення динамічних вебсторінок. PHP є відкритим програмним забезпеченням, тобто його вихідний код доступний для перегляду, модифікації та поширення безкоштовно.

Однією з головних переваг PHP є його простота у використанні. Мова була спочатку розроблена для легкого навчання та швидкого освоєння, що дозволяє розробникам швидко почати створювати вебзастосунки. PHP інтегрується з HTML, що дозволяє вставляти PHP-код безпосередньо в HTML-сторінки. Це робить можливим швидке створення динамічних вебсторінок без потреби у складних налаштуваннях або додаткових інструментах.

PHP також підтримує широкий спектр баз даних, включаючи MySQL, PostgreSQL, Oracle, Microsoft SQL Server та інші. Це дозволяє розробникам створювати динамічні вебзастосунки, які можуть взаємодіяти з базами даних для зберігання та отримання даних. MySQL, зокрема, є дуже популярною у поєднанні з PHP завдяки своїй високій продуктивності, надійності та легкості у використанні. Разом ці технології формують основу багатьох сучасних вебзастосунків.

Крім того, PHP має розширену бібліотеку функцій, що робить його дуже гнучким і потужним інструментом. Вбудовані функції PHP дозволяють виконувати широкий спектр задач, від роботи з рядками та масивами до роботи з файлами і базами даних. Крім того, PHP підтримує розширення, які можна використовувати для додавання додаткових функціональностей, таких як робота з графікою, шифрування даних, інтеграція з іншими протоколами і сервісами.

Ще однією важливою характеристикою PHP є його кросплатформенність. PHP може працювати на різних операційних системах, включаючи Windows, Linux, macOS і Unix. Це дозволяє розробникам створювати вебзастосунки, які можуть бути розгорнуті на різних платформах без необхідності внесення значних змін до коду. Крім того, PHP підтримується більшістю вебсерверів, включаючи Apache, Nginx, IIS та інші.

З точки зору безпеки, PHP має вбудовані механізми для захисту вебзастосунків від різних видів атак, таких як SQL-ін'єкції, XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery) та інші. Розробники можуть використовувати ці механізми для створення безпечних вебзастосунків, знижуючи ризик експлуатації вразливостей. Крім того, активна спільнота PHP постійно працює над вдосконаленням безпеки мови та розробляє нові методи захисту.

PHP також має широку та активну спільноту розробників, яка створює та підтримує численні фреймворки, бібліотеки та інструменти. Це дозволяє розробникам легко знайти підтримку, приклади коду, документацію та інші ресурси, необхідні для успішної розробки вебзастосунків. Деякі з найпопулярніших фреймворків для PHP включають Laravel, Symfony, CodeIgniter, Yii та інші. Ці фреймворки надають розробникам готові рішення для типових задач, що спрощує процес розробки та скорочує час, необхідний для створення вебзастосунків.

PHP також підтримує об'єктно-орієнтоване програмування (ООП), що дозволяє розробникам створювати масштабовані та підтримувані вебзастосунки. Використання класів та об'єктів дозволяє структурувати код, розділяти логіку на

модулі та повторно використовувати код. Це робить PHP потужним інструментом для розробки великих та складних вебзастосунків.

Оновлення PHP також сприяють покращенню продуктивності та безпеки мови. З кожною новою версією додаються нові функції, вдосконалюється існуючий функціонал та виправляються помилки. Наприклад, PHP 7, випущений у 2015 році, приніс значне покращення продуктивності та нові можливості, такі як типізація скалярних типів, покращена обробка помилок та новий оператор `null coalescing`. Останні версії PHP продовжують розвивати мову, додаючи нові можливості та вдосконалюючи існуючий функціонал.

Підсумовуючи, PHP є потужною, гнучкою та простою у використанні мовою програмування, яка забезпечує розробникам всі необхідні інструменти для створення динамічних та інтерактивних вебзастосунків. Її інтеграція з базами даних, кросплатформенність, активна спільнота та підтримка об'єктно-орієнтованого програмування роблять PHP ідеальним вибором для широкого спектру вебзастосунків, від невеликих сайтів до великих корпоративних систем.

Laravel є одним з найпопулярніших фреймворків для PHP, який створений для забезпечення простоти, елегантності та ефективності у розробці вебзастосунків. Розроблений Тейлором Отвеллом у 2011 році, Laravel швидко здобув популярність серед розробників завдяки своїй інтуїтивній архітектурі, багатій функціональності та активній спільноті. Основною метою Laravel є спрощення процесу розробки, забезпечуючи розробників набором інструментів та ресурсів для створення сучасних вебзастосунків.

Однією з ключових особливостей Laravel є його архітектура на основі патерну MVC (Model-View-Controller). MVC розділяє застосунок на три основні компоненти: модель, представлення та контролер. Модель відповідає за взаємодію з базою даних, представлення відповідає за відображення даних користувачеві, а контролер обробляє запити користувачів і координує роботу моделі та представлення. Такий підхід дозволяє структурувати код, полегшуючи його підтримку та масштабування.

Laravel пропонує безліч вбудованих функцій, які спрощують розробку вебзастосунків. Однією з таких функцій є система маршрутизації, яка дозволяє визначати шляхи для обробки запитів користувачів. Розробники можуть легко створювати маршрути, які відповідають за різні дії у застосунку, такі як відображення сторінок, обробка форм та взаємодія з API. Laravel також підтримує маршрутизацію ресурсів, що забезпечує автоматичне створення маршрутів для стандартних CRUD (Create, Read, Update, Delete) операцій.

Ще однією важливою особливістю Laravel є його ORM (Object-Relational Mapping) під назвою Eloquent. Eloquent дозволяє розробникам працювати з базами даних за допомогою об'єктів, а не безпосередньо через SQL-запити. Це спрощує роботу з базами даних, роблячи код більш читабельним та зрозумілим. Eloquent підтримує різні типи відносин між моделями, такі як один до одного, один до багатьох, багато до багатьох, а також поліморфні відносини. Це дозволяє легко визначати та управляти складними зв'язками між даними.

Laravel також пропонує потужну систему аутентифікації та авторизації. Вбудовані механізми дозволяють розробникам швидко реалізовувати реєстрацію, вхід, вихід та управління доступом до різних частин застосунку. Laravel підтримує різні методи аутентифікації, включаючи стандартні логін-пароль, соціальні мережі через OAuth та двофакторну аутентифікацію. Це забезпечує високий рівень безпеки та зручність для користувачів.

Однією з найбільш корисних функцій Laravel є система міграцій баз даних. Міграції дозволяють розробникам відстежувати зміни в структурі бази даних, зберігаючи їх у вигляді коду. Це робить процес управління базами даних більш контрольованим та передбачуваним, дозволяючи легко розгортати зміни на різних середовищах, таких як локальна розробка, тестування та продуктивне середовище. Крім того, міграції можуть бути використані для відновлення бази даних до попереднього стану у разі помилок або необхідності відкату змін.

Laravel також включає потужну систему черг для обробки фонових задач. Черги дозволяють розробникам відправляти задачі на обробку у фоновому режимі, що допомагає зменшити час відгуку застосунку та забезпечити кращу

продуктивність. Наприклад, завдання, такі як надсилання електронної пошти, обробка зображень або виконання довготривалих обчислень, можуть бути відправлені у чергу для обробки окремо від основного потоку застосунку. Це дозволяє зберегти швидкість та чуйність інтерфейсу користувача.

Крім того, Laravel має вбудовану підтримку для тестування застосунків. Фреймворк включає інструменти для написання та запуску тестів, що дозволяє розробникам перевіряти функціональність застосунку та забезпечувати його стабільність. Laravel підтримує як модульне тестування, так і функціональне тестування, що дозволяє перевіряти як окремі частини коду, так і взаємодію між ними. Це допомагає виявляти помилки на ранніх етапах розробки та забезпечувати високу якість коду.

Laravel має активну та залучену спільноту, яка постійно працює над покращенням фреймворку та створенням нових інструментів і розширень. Це забезпечує розробникам доступ до багатого набору ресурсів, включаючи документацію, навчальні матеріали, відеоуроки та форуми для обговорення питань. Крім того, Laravel має велику екосистему пакетів та розширень, які можуть бути легко інтегровані у застосунок для додавання нових функціональностей або оптимізації існуючих процесів.

Загалом, Laravel є потужним та гнучким інструментом для розробки вебзастосунків, який забезпечує розробників всіма необхідними ресурсами для створення масштабованих, безпечних та ефективних рішень. Його архітектура, багатий набір функцій, активна спільнота та підтримка сучасних методологій розробки роблять Laravel одним з найкращих виборів для створення сучасних вебзастосунків.

JavaScript є однією з найпоширеніших мов програмування, яка використовується для створення інтерактивних та динамічних вебзастосунків. Створена Бренданом Айком у 1995 році, JavaScript спочатку використовувалася для додавання інтерактивності до вебсторінок. З часом мова значно еволюціонувала та стала однією з ключових технологій для фронтенд-розробки, а з появою Node.js – також і для серверної розробки.

Однією з основних переваг JavaScript є його можливість виконувати код безпосередньо в браузері. Це дозволяє створювати динамічні інтерфейси користувача, які реагують на дії користувачів у реальному часі. Завдяки цьому JavaScript став незамінним інструментом для створення сучасних вебзастосунків, які забезпечують високий рівень взаємодії та зручності використання. Багато сучасних вебсайтів та застосунків, такі як соціальні мережі, онлайн-магазини та інші, використовують JavaScript для забезпечення багатого користувацького досвіду.

JavaScript є мовою з динамічною типізацією, що означає, що змінні можуть змінювати свій тип під час виконання програми. Це надає розробникам більшу гнучкість, але також вимагає уважності при написанні коду, щоб уникнути помилок. Однак, JavaScript має потужну систему обробки винятків, яка дозволяє виявляти та обробляти помилки, забезпечуючи стабільність роботи застосунків.

JavaScript підтримує об'єктно-орієнтоване програмування (ООП), що дозволяє створювати масштабовані та підтримувані застосунки. Мова підтримує прототипне наслідування, яке є більш гнучким, ніж класичне наслідування, що використовується в інших мовах програмування. Це дозволяє розробникам створювати складні структури даних та повторно використовувати код. В останніх версіях JavaScript була додана підтримка класів, що полегшує розробку об'єктно-орієнтованих програм та робить код більш читабельним та зрозумілим.

JavaScript має багату екосистему бібліотек та фреймворків, які спрощують розробку вебзастосунків. Деякі з найпопулярніших бібліотек включають jQuery, яка спрощує роботу з DOM, та D3.js, яка використовується для створення складних візуалізацій даних. Серед фреймворків для розробки фронтенд-частини застосунків особливо виділяються React, Angular та Vue.js. Кожен з цих фреймворків має свої унікальні особливості та переваги, але всі вони спрямовані на спрощення процесу розробки та забезпечення високої продуктивності застосунків.

React, розроблений компанією Facebook, є бібліотекою для створення користувацьких інтерфейсів. Його основною концепцією є компонентний підхід,

який дозволяє створювати багаторазові та ізольовані компоненти. Це спрощує розробку та підтримку складних інтерфейсів користувача. Angular, розроблений Google, є фреймворком для побудови масштабованих вебзастосунків. Він підтримує двосторонню прив'язку даних, що забезпечує автоматичне оновлення інтерфейсу при зміні даних. Vue.js, розроблений Еваном Ю, є прогресивним фреймворком, який можна легко інтегрувати у вже існуючі проекти або використовувати для створення нових застосунків.

Node.js, створений Райаном Далем у 2009 році, дозволяє виконувати JavaScript на сервері. Це розширило можливості мови, дозволивши використовувати її для повного стеку розробки. Node.js базується на асинхронній моделі введення-виведення, що забезпечує високу продуктивність та масштабованість застосунків. Node.js часто використовується для створення серверів реального часу, RESTful API та мікросервісів. Завдяки своїй високій продуктивності та легкості у використанні, Node.js став популярним вибором для створення сучасних вебзастосунків та сервісів.

Однією з важливих особливостей JavaScript є його широка підтримка різними браузером. Всі основні веббраузери, включаючи Google Chrome, Mozilla Firefox, Microsoft Edge, Safari та інші, мають вбудовані двигуни JavaScript, які забезпечують виконання коду без необхідності додаткових плагінів або налаштувань. Це забезпечує високу сумісність та доступність вебзастосунків на різних пристроях та платформах.

JavaScript також має потужну систему модулів, яка дозволяє розробникам організовувати та управляти кодом. Система модулів дозволяє розбивати код на окремі файли та компоненти, які можуть бути легко імпортовані та використовувані у різних частинах застосунку. Це сприяє кращій структурі коду, полегшує його підтримку та повторне використання. У останніх версіях JavaScript була додана підтримка нативних модулів, що дозволяє використовувати стандартизовану систему модулів без необхідності у зовнішніх бібліотеках або інструментах.

JavaScript є мовою з відкритим кодом, що означає, що її розвиток та підтримка здійснюються спільнотою розробників. Це забезпечує постійне вдосконалення мови, додавання нових можливостей та виправлення помилок. Активна спільнота також створює численні інструменти, бібліотеки та фреймворки, які спрощують процес розробки та забезпечують високу продуктивність застосунків.

Підсумовуючи, JavaScript є потужною, гнучкою та популярною мовою програмування, яка забезпечує розробникам всі необхідні інструменти для створення сучасних вебзастосунків. Його можливість виконуватись безпосередньо в браузері, підтримка об'єктно-орієнтованого програмування, багата екосистема бібліотек та фреймворків, а також підтримка серверної розробки з використанням Node.js роблять JavaScript незамінним інструментом для фронтенд- та бекенд-розробки.

React, розроблений компанією Facebook, є однією з найпопулярніших бібліотек JavaScript для створення користувацьких інтерфейсів. Вперше представлений у 2013 році, React швидко здобув популярність серед розробників завдяки своєму компонентному підходу, високій продуктивності та легкості у використанні. Основною метою React є забезпечення ефективного та гнучкого способу створення динамічних та інтерактивних інтерфейсів користувача.

Однією з ключових концепцій React є компонентний підхід до розробки інтерфейсів. Компоненти в React є багаторазовими та ізольованими елементами, які можуть бути легко поєднані для створення складних інтерфейсів. Кожен компонент може мати власний стан (state) та властивості (props), що дозволяє зберігати та управляти даними незалежно від інших компонентів. Це робить розробку більш модульною та зручною для підтримки, оскільки зміни в одному компоненті не впливають на інші компоненти.

React використовує віртуальний DOM (Document Object Model), що забезпечує високу продуктивність при оновленні інтерфейсу. Віртуальний DOM є легкою копією реального DOM, яка зберігається в пам'яті. Коли стан компонента змінюється, React порівнює новий віртуальний DOM з попереднім і

визначає, які зміни потрібно внести до реального DOM. Це значно зменшує кількість операцій з реальним DOM, що є однією з головних причин високої продуктивності React. Такий підхід також дозволяє уникнути непотрібних оновлень інтерфейсу, що забезпечує більш плавний та швидкий користувацький досвід.

Однією з важливих особливостей React є підтримка одностороннього потоку даних. Це означає, що дані передаються від батьківських компонентів до дочірніх через властивості (props), але не навпаки. Такий підхід забезпечує кращий контроль за потоком даних та зменшує ймовірність помилок, пов'язаних з неправильним оновленням стану компонентів. Крім того, односторонній потік даних робить застосунки більш передбачуваними та легкими для дебагування.

React також підтримує JSX (JavaScript XML), що дозволяє писати HTML-подібний код безпосередньо в JavaScript. JSX спрощує процес створення інтерфейсів, роблячи код більш читабельним та зрозумілим. Відмінною рисою JSX є можливість використовувати всю потужність JavaScript для створення динамічних інтерфейсів, що робить його потужним інструментом для розробників. Крім того, JSX дозволяє легко інтегрувати сторонні бібліотеки та компоненти, що робить React гнучким та розширюваним.

Однією з найбільших переваг React є його широка екосистема та активна спільнота. Існує безліч бібліотек та інструментів, які можуть бути використані разом з React для спрощення розробки та забезпечення високої продуктивності застосунків. Деякі з найпопулярніших бібліотек включають React Router для управління маршрутизацією у застосунках з однією сторінкою (SPA), Redux для управління глобальним станом застосунка та Material-UI для створення адаптивних інтерфейсів з використанням готових компонентів.

React також підтримує серверний рендеринг, що дозволяє генерувати HTML на сервері перед тим, як він буде відправлений клієнту. Це забезпечує швидше завантаження сторінок та покращує індексацію застосунків пошуковими системами. Серверний рендеринг особливо корисний для застосунків з великою кількістю контенту, де швидкість завантаження та SEO є критично важливими.

React Native, розроблений на основі React, дозволяє створювати мобільні застосунки для iOS та Android з використанням тих самих принципів та методів, що і для веброзробки. React Native дозволяє розробникам використовувати JavaScript та React для створення нативних мобільних застосунків, що забезпечує високу продуктивність та зручність використання. Це робить React ще більш привабливим вибором для розробників, які хочуть створювати застосунки для різних платформ з використанням однієї технології.

React постійно розвивається та вдосконалюється завдяки активній спільноті розробників та внеску компанії Facebook. Регулярні оновлення забезпечують додавання нових функцій, виправлення помилок та покращення продуктивності. Наприклад, у версії React 16 було додано підтримку фрагментів, що дозволяє створювати компоненти без необхідності додавання зайвих елементів у DOM. Крім того, було впроваджено новий механізм обробки помилок, що дозволяє краще контролювати стан застосунка у випадку помилок.

MySQL є однією з найпопулярніших систем управління реляційними базами даних (СУРБД) у світі. Розроблена компанією MySQL AB у середині 1990-х років, MySQL стала основою для багатьох вебзастосунків завдяки своїй продуктивності, надійності та простоті у використанні. Сьогодні MySQL використовується мільйонами вебсайтів та застосунків, включаючи деякі з найбільших та найбільш відвідуваних у світі.

Однією з ключових особливостей MySQL є її реляційна структура, яка використовує таблиці для зберігання даних. Ця структура дозволяє ефективно організовувати та керувати даними, забезпечуючи високий рівень цілісності та узгодженості. Реляційна модель також дозволяє виконувати складні запити до бази даних за допомогою SQL (Structured Query Language), що робить MySQL потужним інструментом для аналізу та обробки даних.

MySQL підтримує широкий спектр типів даних, включаючи числові, текстові, часові та бінарні типи. Це дозволяє зберігати різноманітні дані та забезпечує гнучкість у розробці застосунків. Крім того, MySQL підтримує

складні структури даних, такі як індекси, ключі та зв'язки між таблицями, що дозволяє оптимізувати продуктивність та забезпечити цілісність даних.

Однією з найбільших переваг MySQL є її висока продуктивність. MySQL використовує оптимізований механізм обробки запитів, який дозволяє швидко виконувати складні запити до бази даних. Крім того, MySQL підтримує реплікацію баз даних, що дозволяє створювати копії бази даних на різних серверах для забезпечення високої доступності та навантаження. Це особливо корисно для великих вебзастосунків, які обробляють великий обсяг запитів та потребують високої продуктивності.

MySQL також відома своєю надійністю та стійкістю до збоїв. Система забезпечує механізми для резервного копіювання та відновлення даних, що дозволяє мінімізувати втрати даних у випадку аварійних ситуацій. Крім того, MySQL підтримує транзакції, які забезпечують цілісність даних та дозволяють виконувати кілька операцій як одну атомарну операцію. Це гарантує, що база даних завжди знаходиться у консистентному стані, навіть у випадку збоїв.

Безпека є ще однією важливою характеристикою MySQL. Система підтримує різні механізми аутентифікації та авторизації, що дозволяє контролювати доступ до бази даних. Розробники можуть визначати різні рівні доступу для користувачів, забезпечуючи захист від несанкціонованого доступу. Крім того, MySQL підтримує шифрування даних як на рівні зберігання, так і під час передачі, що забезпечує додатковий рівень захисту даних.

Однією з ключових причин популярності MySQL є її відкритий код та активна спільнота розробників. Відкритий код дозволяє розробникам переглядати, модифікувати та поширювати MySQL безкоштовно, що робить її доступною для широкого кола користувачів. Активна спільнота забезпечує постійну підтримку, вдосконалення та розвиток MySQL, що гарантує стабільність та актуальність системи. Крім того, існує велика кількість ресурсів, таких як документація, форуми, навчальні матеріали та відеоуроки, які допомагають розробникам швидко освоїти MySQL та ефективно використовувати її у своїх проектах.

MySQL також підтримує різні програмні інтерфейси (API), що дозволяють легко інтегрувати її з іншими мовами програмування та фреймворками. Це включає підтримку таких мов, як PHP, Java, Python, Ruby, C# та інші, що робить MySQL універсальним інструментом для розробників. Інтеграція з популярними фреймворками, такими як Laravel для PHP, Django для Python та Hibernate для Java, забезпечує швидку та ефективну розробку застосунків.

MySQL є кросплатформеною системою, яка може працювати на різних операційних системах, включаючи Windows, Linux, macOS та Unix. Це дозволяє розробникам використовувати MySQL на різних платформах та забезпечує високу сумісність з існуючими системами. Крім того, MySQL підтримується більшістю хостинг-провайдерів, що робить її зручним вибором для розгортання вебзастосунків.

Підсумовуючи, MySQL є потужною, надійною та продуктивною системою управління реляційними базами даних, яка забезпечує розробникам всі необхідні інструменти для створення ефективних та масштабованих застосунків. Її реляційна структура, підтримка складних запитів, висока продуктивність, надійність, безпека та активна спільнота роблять MySQL одним з найкращих виборів для зберігання та управління даними у сучасних вебзастосунках. Використання MySQL у поєднанні з іншими популярними технологіями, такими як PHP та фреймворк Laravel, дозволяє створювати потужні та гнучкі рішення, що відповідають потребам бізнесу та користувачів.

3.2 Моделювання бази даних

Фізична модель бази даних є найнижчим рівнем моделювання даних, який описує, як дані зберігаються в пам'яті комп'ютера. Вона визначає конкретні деталі зберігання даних, структури файлів, індекси, способи доступу та інші аспекти фізичного зберігання, необхідні для забезпечення ефективного доступу до даних.

Фізичну модель бази даних вебсайту магазину меблів представлено на рисунку 3.1.

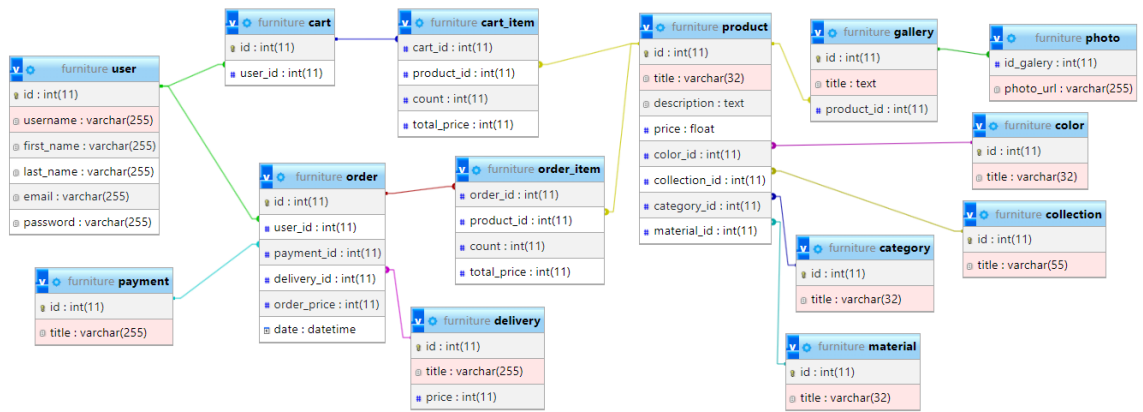


Рисунок 3.1 – Фізична модель бази даних магазину меблів

Давайте розглянемо кожну таблицю окремо.

Таблиця cart

Ця таблиця представляє корзину користувача.

- id (int, NOT NULL): Унікальний ідентифікатор корзины.
- user_id (int, NOT NULL): Ідентифікатор користувача, якому належить ця корзина.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	user_id	int(11)			Ні	Немає			Більше

Друк Запропонувати структуру таблиці Перемістити стовпці Упорядкувати

Додати 1 стовпець(ів) після user_id Виконати

Індекси

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
PRIMARY		BTREE	Так	Ні	id	0	A	Ні	
	user_id	BTREE	Ні	Ні	user_id	0	A	Ні	

Рисунок 3.2 – Структура таблиці корзины

Таблиця cart_item

Ця таблиця містить товари, додані до корзины.

- cart_id (int, NOT NULL): Ідентифікатор корзины, до якої додано товар.
- product_id (int, NOT NULL): Ідентифікатор товару, доданого до корзины.
- count (int, NOT NULL): Кількість одиниць товару.

- total_price (int, NOT NULL): Загальна ціна для цих одиниць товару.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	cart_id	int(11)			Ні	Немає			Більше
2	product_id	int(11)			Ні	Немає			Більше
3	count	int(11)			Ні	Немає			Більше
4	total_price	int(11)			Ні	Немає			Більше

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	cart_item_ibfk_1	BTREE	Ні	Ні	cart_id	0	A	Ні	
	cart_item_ibfk_2	BTREE	Ні	Ні	product_id	0	A	Ні	

Рисунок 3.3 – Структура таблиці елементів корзини

Таблиця category

Ця таблиця містить категорії товарів.

- id (int, NOT NULL): Унікальний ідентифікатор категорії.
- title (varchar(32), NOT NULL): Назва категорії.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	title	varchar(32)	utf8mb3_general_ci		Ні	Немає			Більше

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	6	A	Ні	

Рисунок 3.4 – Структура таблиці категорій

Таблиця collection

Ця таблиця містить колекції товарів.

- id (int, NOT NULL): Унікальний ідентифікатор колекції.

- title (varchar(55), NOT NULL): Назва колекції.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	title	varchar(55)	utf8mb3_general_ci		Ні	Немає			Більше

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	9	A	Ні	

Рисунок 3.5 – Структура таблиці колекцій

Таблиця color

Ця таблиця містить кольори товарів.

- id (int, NOT NULL): Унікальний ідентифікатор кольору.
- title (varchar(32), NOT NULL): Назва кольору.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	title	varchar(32)	utf8mb3_general_ci		Ні	Немає			Більше

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	8	A	Ні	

Рисунок 3.6 – Структура таблиці кольорів

Таблиця delivery

Ця таблиця містить інформацію про способи доставки.

- id (int, NOT NULL): Унікальний ідентифікатор способу доставки.
- title (varchar(255), NOT NULL): Назва способу доставки.
- price (int, NOT NULL): Вартість доставки.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	title	varchar(255)	utf8mb3_general_ci		Ні	Немає			Більше
3	price	int(11)			Ні	Немає			Більше

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	0	A	Ні	

Рисунок 3.7 – Структура таблиці видів доставки

Таблиця gallery

Ця таблиця містить галереї товарів.

- id (int, NOT NULL): Унікальний ідентифікатор галереї.
- title (text, NOT NULL): Назва або опис галереї.
- product_id (int, NOT NULL): Ідентифікатор товару, до якого належить ця галерея.

галерея.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	title	text	utf8mb3_general_ci		Ні	Немає			Більше
3	product_id	int(11)			Ні	Немає			Більше

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	0	A	Ні	
	product_id	BTREE	Ні	Ні	product_id	0	A	Ні	

Рисунок 3.8 – Структура таблиці галереї

Таблиця material

Ця таблиця містить матеріали товарів.

- id (int, NOT NULL): Унікальний ідентифікатор матеріалу.

- title (varchar(32), NOT NULL): Назва матеріалу.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	title	varchar(32)	utf8mb3_general_ci		Ні	Немає			Більше

↑ Позначити все Вибрані:

Друк Запропонувати структуру таблиці Перемістити стовпці Упорядкувати

Додати стовпець(ів)

Індекси

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	4	A	Ні	

Рисунок 3.9 – Структура таблиці матеріалів

Таблиця order

Ця таблиця містить замовлення.

- id (int, NOT NULL): Унікальний ідентифікатор замовлення.
- user_id (int, NOT NULL): Ідентифікатор користувача, який зробив замовлення.
- payment_id (int, NOT NULL): Ідентифікатор способу оплати.
- delivery_id (int, NOT NULL): Ідентифікатор способу доставки.
- order_price (int, NOT NULL): Загальна вартість замовлення.
- date (datetime, NOT NULL, DEFAULT current_timestamp()): Дата та час створення замовлення.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
<input type="checkbox"/>	2 user_id	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	3 payment_id	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	4 delivery_id	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	5 order_price	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	6 date	datetime			Ні	current_timestamp()			Більше

↑ Позначити все Вибрані:

[Друк](#) [Запропонувати структуру таблиці](#) [Перемістити стовпці](#) [Упорядкувати](#)

Додати стовпець(ів) [Виконати](#)

Індекси

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	0	A	Ні	
	delivery_id	BTREE	Ні	Ні	delivery_id	0	A	Ні	
	payment_id	BTREE	Ні	Ні	payment_id	0	A	Ні	
	user_id	BTREE	Ні	Ні	user_id	0	A	Ні	

Рисунок 3.10 – Структура таблиці заказів

Таблиця order_item

Ця таблиця містить товари в замовленні.

- order_id (int, NOT NULL): Ідентифікатор замовлення.
- product_id (int, NOT NULL): Ідентифікатор товару.
- count (int, NOT NULL): Кількість одиниць товару.
- total_price (int, NOT NULL): Загальна ціна для цих одиниць товару.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	order_id	int(11)			Ні	Немає			Більше
2	product_id	int(11)			Ні	Немає			Більше
3	count	int(11)			Ні	Немає			Більше
4	total_price	int(11)			Ні	Немає			Більше

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	order_id	BTREE	Ні	Ні	order_id	0	A	Ні	
	product_id	BTREE	Ні	Ні	product_id	0	A	Ні	

Рисунок 3.11 – Структура таблиці елементів заказів

Таблиця payment

Ця таблиця містить способи оплати.

- id (int, NOT NULL): Унікальний ідентифікатор способу оплати.
- title (varchar(255), NOT NULL): Назва способу оплати.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	title	varchar(255)	utf8mb3_general_ci		Ні	Немає			Більше

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	0	A	Ні	

Рисунок 3.12 – Структура таблиці видів оплати

Таблиця photo

Ця таблиця містить фото в галереї.

- id_galery (int, NOT NULL): Ідентифікатор галереї.
- photo_url (varchar(255), NOT NULL): URL адреса фото.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id_gallery	int(11)			Ні	Немає			Більше
2	photo_url	varchar(255)	utf8mb3_general_ci		Ні	Немає			Більше

Індекси																				
<table border="1"> <thead> <tr> <th>Дія</th> <th>Назва ключа</th> <th>Тип</th> <th>Унікальне</th> <th>Запакований</th> <th>Стовпець</th> <th>Кількість елементів</th> <th>Зіставлення</th> <th>Нуль</th> <th>Коментар</th> </tr> </thead> <tbody> <tr> <td></td> <td>id_gallery</td> <td>BTREE</td> <td>Ні</td> <td>Ні</td> <td>id_gallery</td> <td>0</td> <td>A</td> <td>Ні</td> <td></td> </tr> </tbody> </table>	Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар		id_gallery	BTREE	Ні	Ні	id_gallery	0	A	Ні	
Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар											
	id_gallery	BTREE	Ні	Ні	id_gallery	0	A	Ні												

Рисунок 3.13 – Структура таблиці фотографій

Таблиця product

Ця таблиця містить товари.

- id (int, NOT NULL): Унікальний ідентифікатор товару.
- title (varchar(32), NOT NULL): Назва товару.
- description (text, NOT NULL): Опис товару.
- price (float, DEFAULT NULL): Ціна товару.
- color_id (int, NOT NULL): Ідентифікатор кольору товару.
- collection_id (int, NOT NULL): Ідентифікатор колекції, до якої належить товар.
- category_id (int, NOT NULL): Ідентифікатор категорії, до якої належить товар.
- material_id (int, NOT NULL): Ідентифікатор матеріалу, з якого виготовлений товар.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id 🔑	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
<input type="checkbox"/>	2 title	varchar(32)	utf8mb3_general_ci		Ні	Немає			Більше
<input type="checkbox"/>	3 description	text	utf8mb3_general_ci		Ні	Немає			Більше
<input type="checkbox"/>	4 price	float			Так	NULL			Більше
<input type="checkbox"/>	5 color_id 🔑	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	6 collection_id 🔑	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	7 category_id 🔑	int(11)			Ні	Немає			Більше
<input type="checkbox"/>	8 material_id 🔑	int(11)			Ні	Немає			Більше

↑ Позначити все Вибрані:

Друк Запропонувати структуру таблиці Перемістити стовпці Упорядкувати

Додати стовпець(ів)

Індекси ⓘ

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	34	A	Ні	
	category_id	BTREE	Ні	Ні	category_id	17	A	Ні	
	product_ibfk_2	BTREE	Ні	Ні	collection_id	34	A	Ні	
	color_id	BTREE	Ні	Ні	color_id	17	A	Ні	
	material_id	BTREE	Ні	Ні	material_id	11	A	Ні	

Рисунок 3.14 – Структура таблиці продуктів

Таблиця user

Ця таблиця містить інформацію про користувачів.

- **id** (int, NOT NULL): Унікальний ідентифікатор користувача.
- **username** (varchar(255), DEFAULT NULL): Логін користувача.
- **first_name** (varchar(255), NOT NULL): Ім'я користувача.
- **last_name** (varchar(255), NOT NULL): Прізвище користувача.
- **email** (varchar(255), NOT NULL): Електронна адреса користувача.
- **password** (varchar(255), NOT NULL): Пароль користувача.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	id	int(11)		Ні	Немає		AUTO_INCREMENT	Більше
<input type="checkbox"/>	2	username	varchar(255)	utf8mb3_general_ci	Так	NULL			Більше
<input type="checkbox"/>	3	first_name	varchar(255)	utf8mb3_general_ci	Ні	Немає			Більше
<input type="checkbox"/>	4	last_name	varchar(255)	utf8mb3_general_ci	Ні	Немає			Більше
<input type="checkbox"/>	5	email	varchar(255)	utf8mb3_general_ci	Ні	Немає			Більше
<input type="checkbox"/>	6	password	varchar(255)	utf8mb3_general_ci	Ні	Немає			Більше

↑ Позначити все Вибрані:

Друк Запропонувати структуру таблиці Перемістити стовпці Упорядковувати

Додати стовпець(ів) після password

Індекси

Дія	Назва ключа	Тип	Унікальний	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	0	A	Ні	

Рисунок 3.15 – Структура таблиці користувачів

Таким чином представлено структури таблиць бази даних.

3.3 Розробка користувацького інтерфейсу

Мокап системи – це модель або прототип інтерфейсу користувача, який використовується для візуалізації та тестування концепцій програмного забезпечення перед його розробкою. Це своєрідна «макетна» версія системи, яка дозволяє команді розробників та зацікавленим сторонам краще розуміти функціонал програми або вебзастосунку [1].

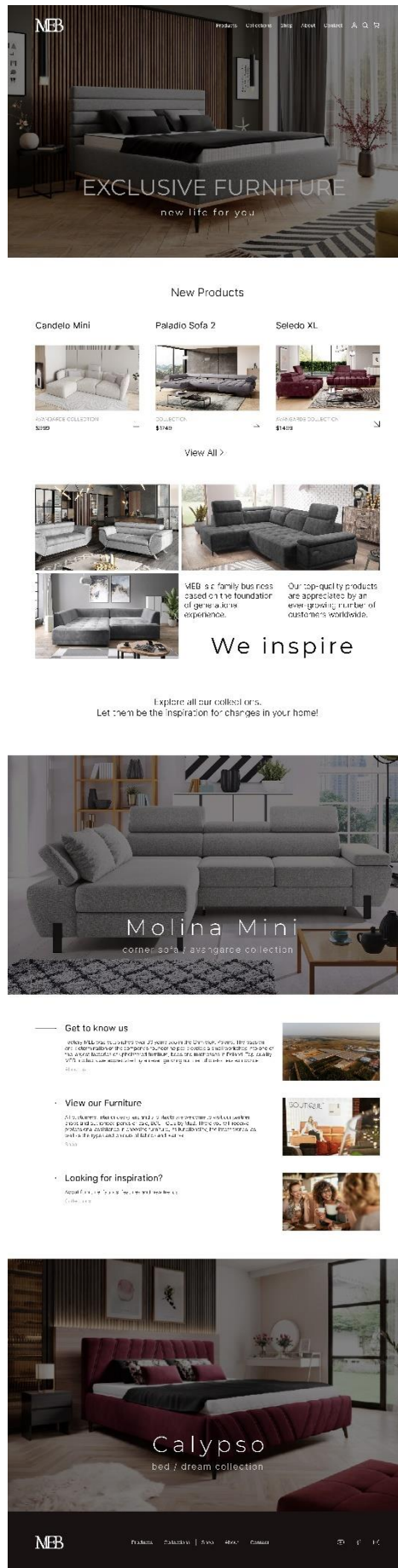


Рисунок 3.16 – Головна сторінка

Header. На верхній частині сторінки розташований хедер, який містить логотип магазину та основні навігаційні елементи, такі як «Каталог» «Колекції», «Магазин», «Про нас», «Контакти», «Особистий кабінет», «Пошук» та «Кошик».

Останні нові додані продукти. Цей блок представляє собою сітку з останніми доданими продуктами. Кожен товар відображається зі зображенням, назвою, колекцією, до якої належить, та ціною. В розділі також розміщена кнопка «Переглянути всі», яка спрямовує користувачів на повний каталог товарів.

Блок з посиланням на окремий випадковий продукт. У цьому блоку випадково вибирається один продукт з каталогу і відображається з назвою та зображенням, при натисканні на нього користувач може швидко перейти до деталей цього конкретного товару.

Блок «Про нас». У цьому розділі представлена коротка інформація про магазин. Також цей блок містить фотографії команди та заводу, що працює над створенням меблів, щоб підкреслити професіоналізм та експертність.

Footer. У нижній частині сторінки розташований футер, який містить логотип магазину, основні навігаційні елементи та посилання на соціальні мережі.

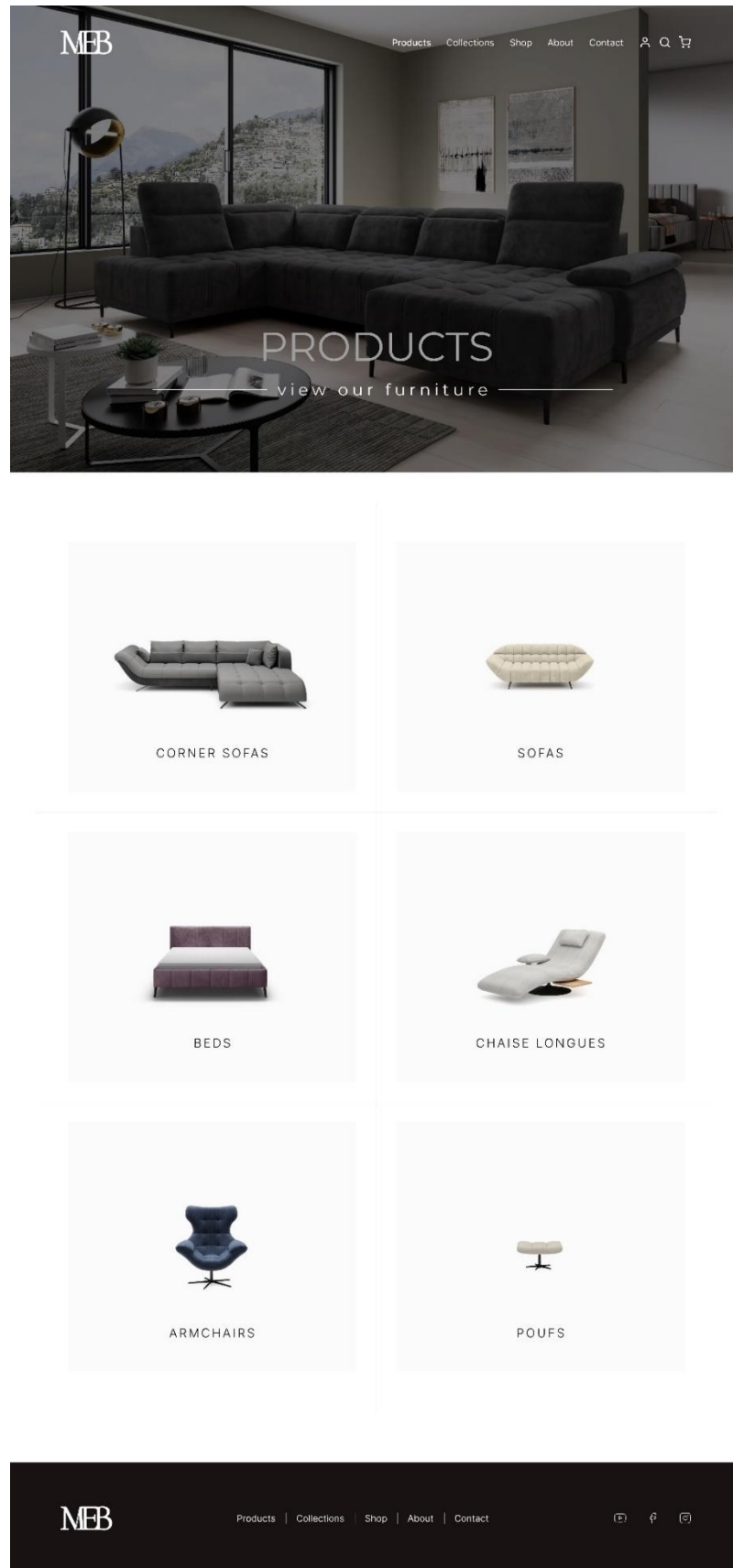


Рисунок 3.17 – Каталог

Сторінка каталогу містить 6 блоків з різними типами меблів, які виробляються.

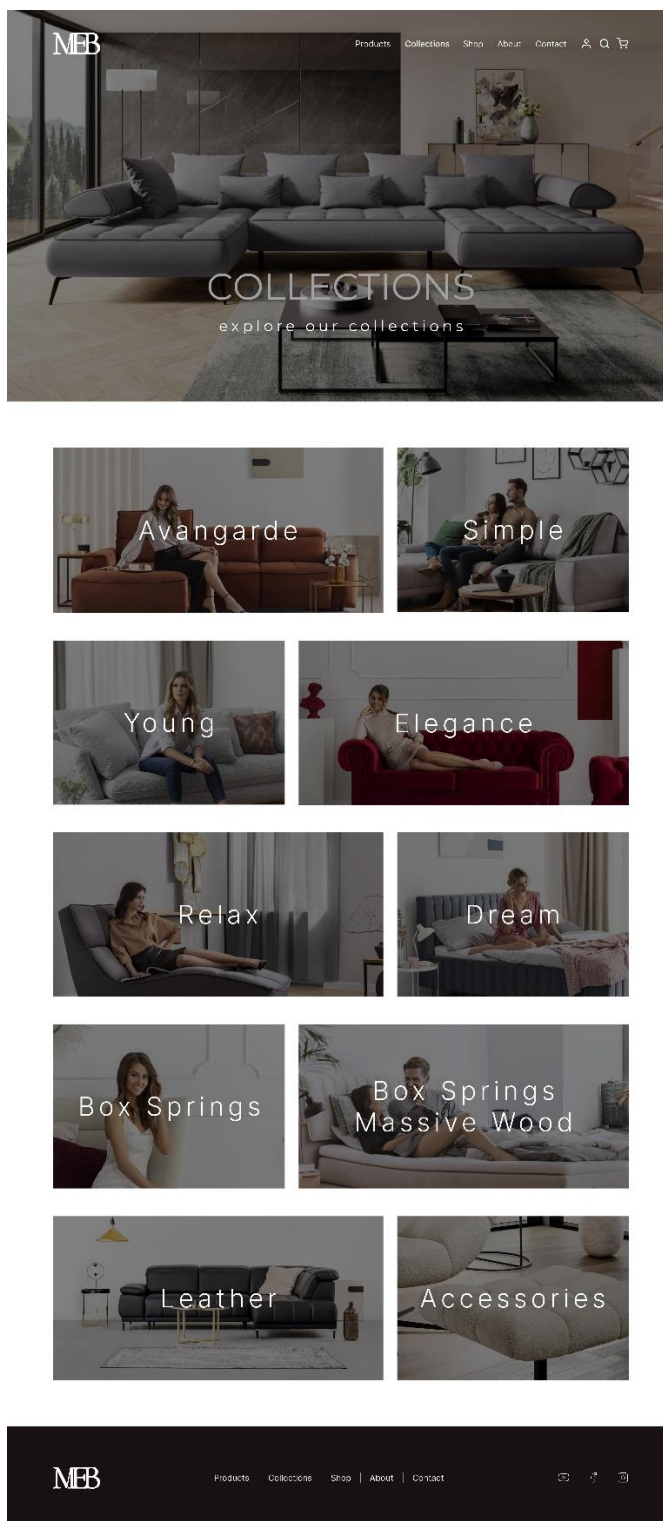


Рисунок 3.18 – Колекції

Сторінка колекцій містить 10 колекцій, продукти в яких схожі за стилем відповідно у кожній колекції.



Рисунок 3.19 – Продукти

Сторінка продуктів містить продукти, які відносяться до окремого типу меблів.



Рисунок 3.19 – Сторінка продукту

Сторінка товару містить зображення, назву, тип та колекцію, до яких відноситься, матеріал, з якого вироблений, ціну, кнопку «Додати у кошик», розміри та функції, галерею зображень, відгуки та оцінку (незарєєстрований користувач має зарєєструватися, щоб залишити відгук) та рекомендації з іншими системними елементами продукту.

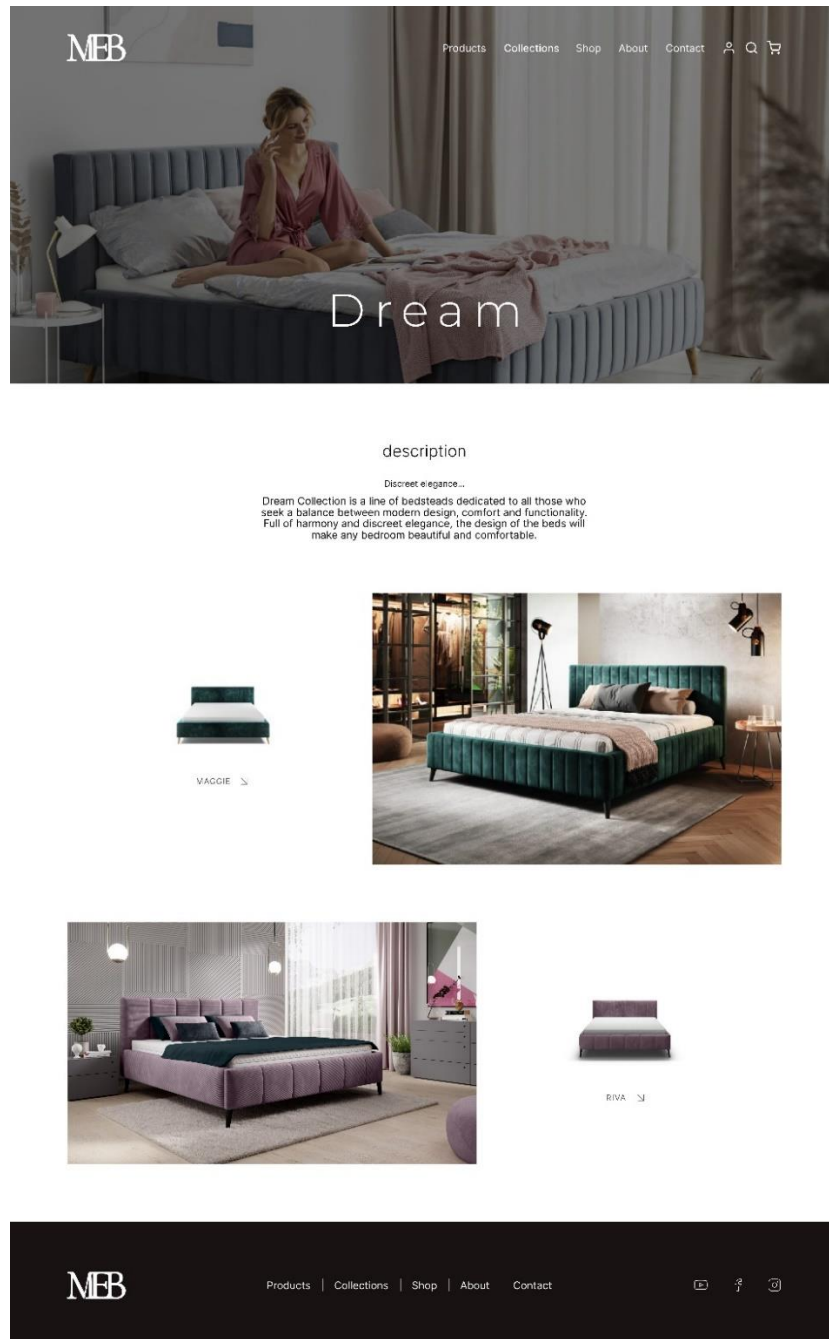


Рисунок 3.20 – Сторінка колекції

Сторінка колекції містить опис та продукти, схожі за стилем.

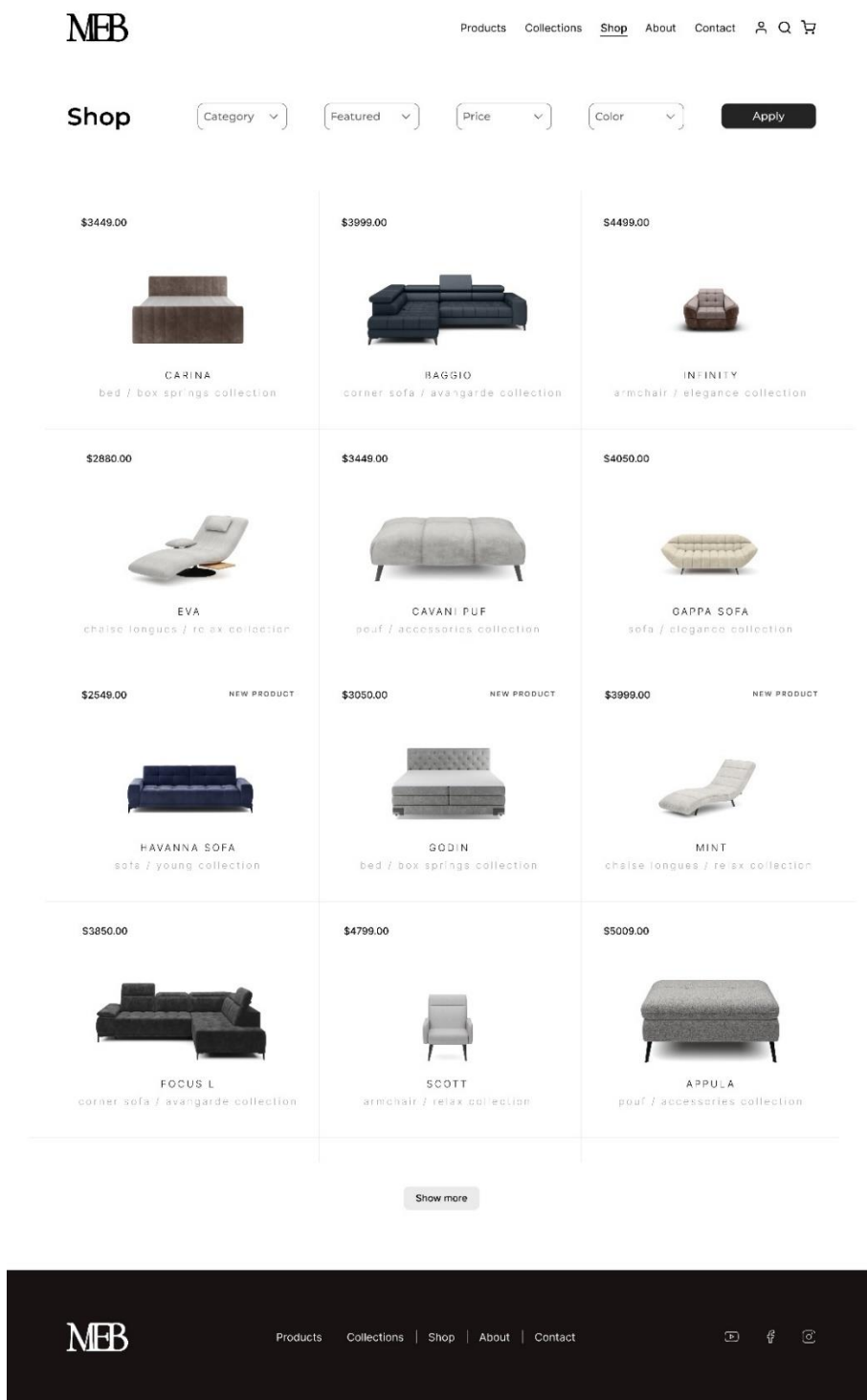


Рисунок 3.21 – Сторінка магазину

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи представлено опис технологій розробки вебзастосунку магазину меблів. Розроблено та розглянуто фізичну модель бази даних вебзастосунку, а також описано структуру кожної таблиці окремо.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Реалізація клієнтської частини

У цьому підрозділі представлено результати розробки клієнтської частини вебзастосунку інтернет-магазину меблів.

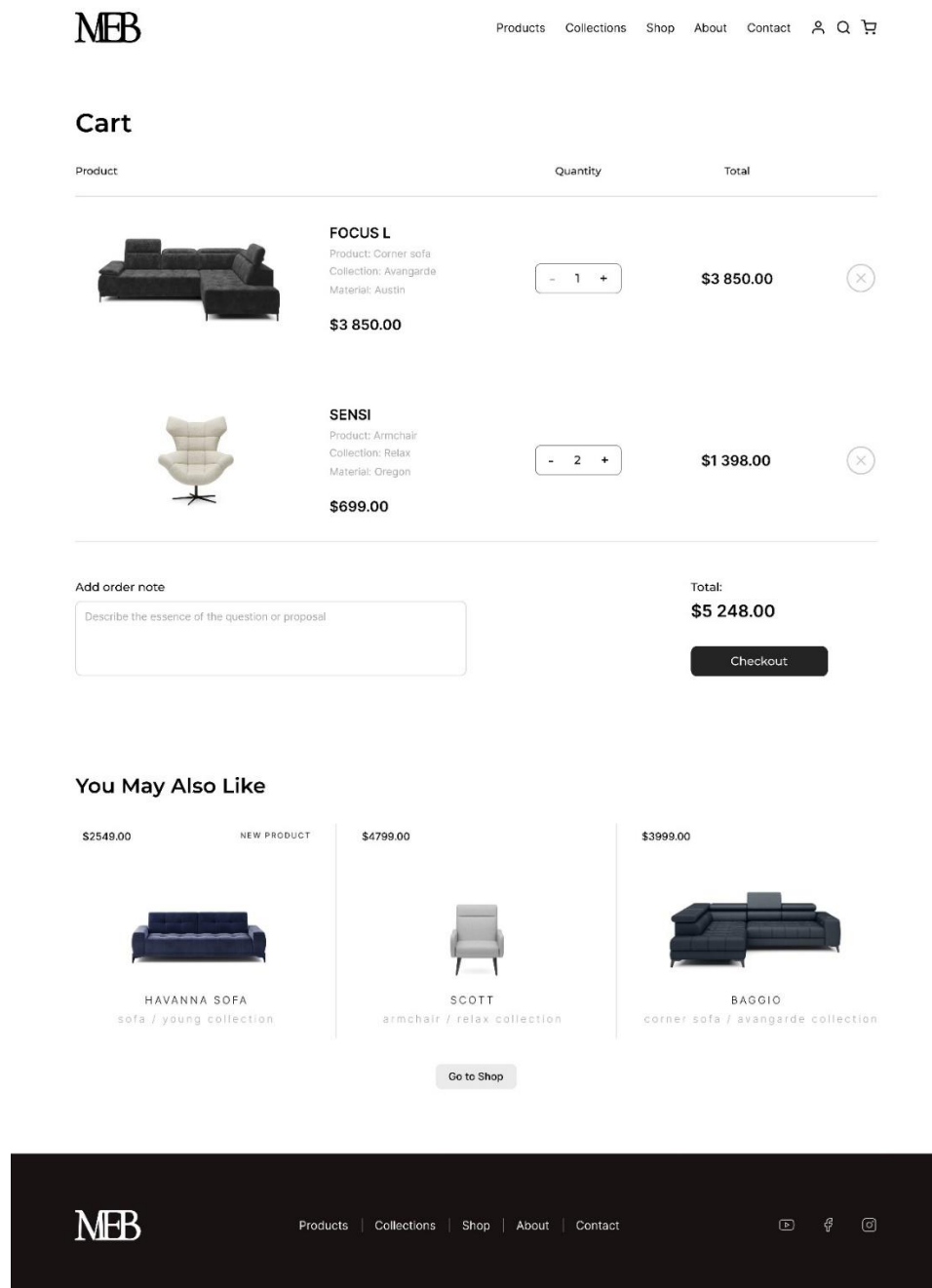
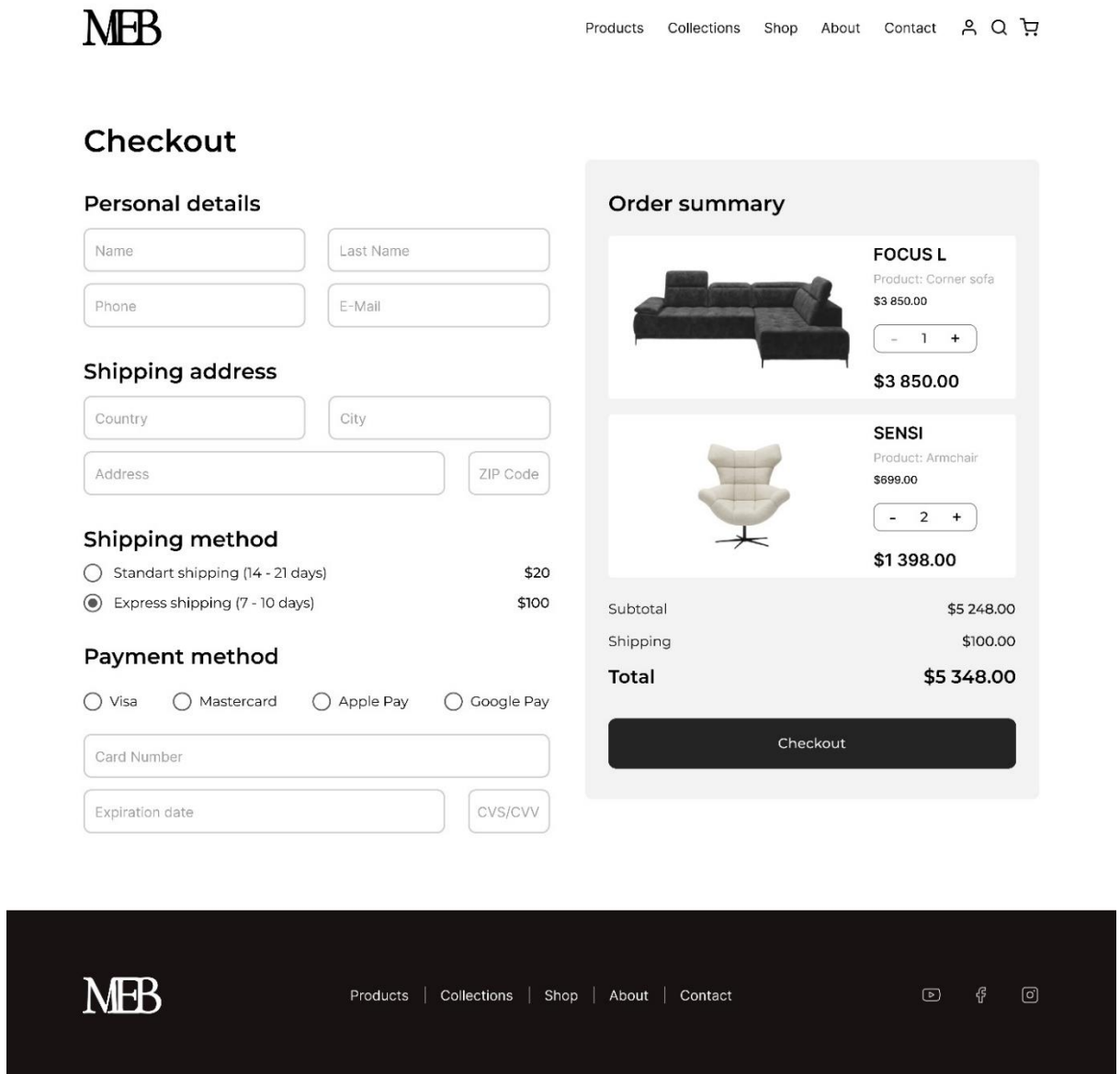


Рисунок 4.1 – Кошик

Усі вибрані користувачем продукти додані у **кошик**, з можливістю видалити та змінити кількість товару. Поле для додавання примітки, кнопка

«Оформити замовлення» та рекомендації з товарами, які можуть сподобатись також наявні.



MEB Products Collections Shop About Contact

Checkout

Personal details

Name Last Name
Phone E-Mail

Shipping address

Country City
Address ZIP Code

Shipping method

Standart shipping (14 - 21 days) \$20
 Express shipping (7 - 10 days) \$100

Payment method

Visa Mastercard Apple Pay Google Pay

Card Number
Expiration date CVS/CVV

Order summary

FOCUS L
Product: Corner sofa
\$3 850.00
- 1 +
\$3 850.00

SENSI
Product: Armchair
\$899.00
- 2 +
\$1 398.00

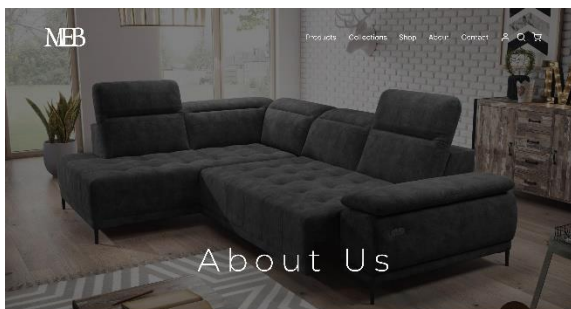
Subtotal \$5 248.00
Shipping \$100.00
Total \$5 348.00

Checkout

MEB Products | Collections | Shop | About | Contact

Рисунок 4.2 – Оформлення замовлення

Сторінка з оформленням замовлення являє собою форму для заповнення важливих даних, таких як персональні данні, адреса доставки, метод доставки, заповнення даних для сплати. Підсумок замовлення та кнопка «Сплатити» також наявні.



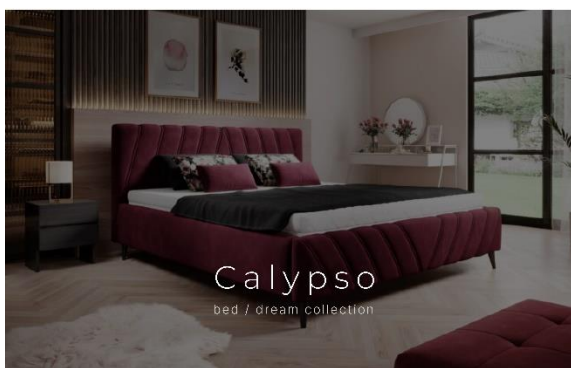
We inspire

MEB is a family business based on the decades long generational experience. For over 30 years, it has consistently pursued its vision of creating furniture that combines elegance, functionality and the latest trends with a unique awareness of market needs. Our top quality products are appreciated by an ever-growing number of customers worldwide. The company believes in continuous development to make sure that the furniture created by MEB is created in response to the real needs of users. Explore all our collections. Let them be the inspiration for changes in your home!

- avangarde collection
- simple collection
- young collection
- elegance collection
- relax collection



- dream collection
- box springs collection
- box springs massive collection
- leather collection
- accessories collection



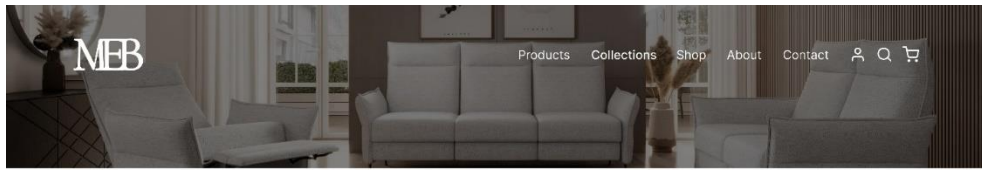
Passion to create

The high-quality products are appreciated by a numerous group of people worldwide. It is reflected in a positive response, recognition at industry trade fairs and industry exhibitions. The company values customer feedback. The production team works in accordance with the Design Thinking principles to ensure that the furniture created by MEB is made in response to the real needs of the users.



Рисунок 4.3 – Сторінка «Про нас»

Сторінка «Про нас» з описом магазину та посиланнями на колекції для ознайомлення.



Contact Us

contact us any way you like

Adress

Pl. Croix-Blanche 11,
1066 Epalinges

Phone

+41794441010

E-Mail

meb@gmail.com

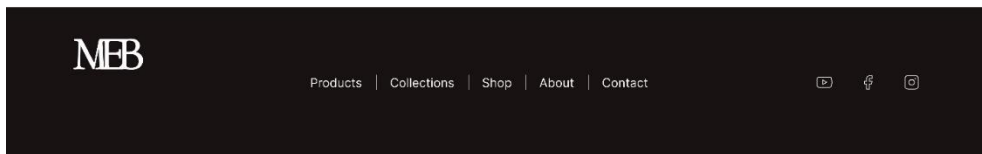
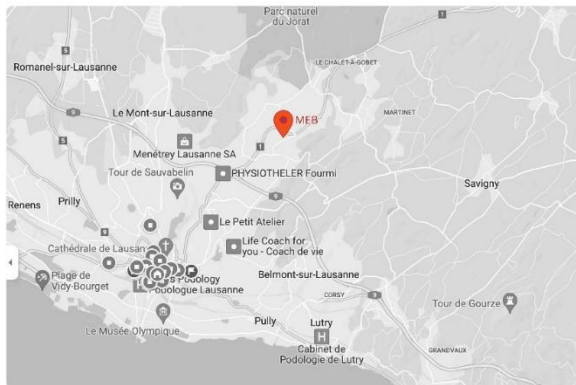


Рисунок 4.4 – Сторінка «Контакти»

Сторінка «Контакти» з контактною інформацією для зв'язку.

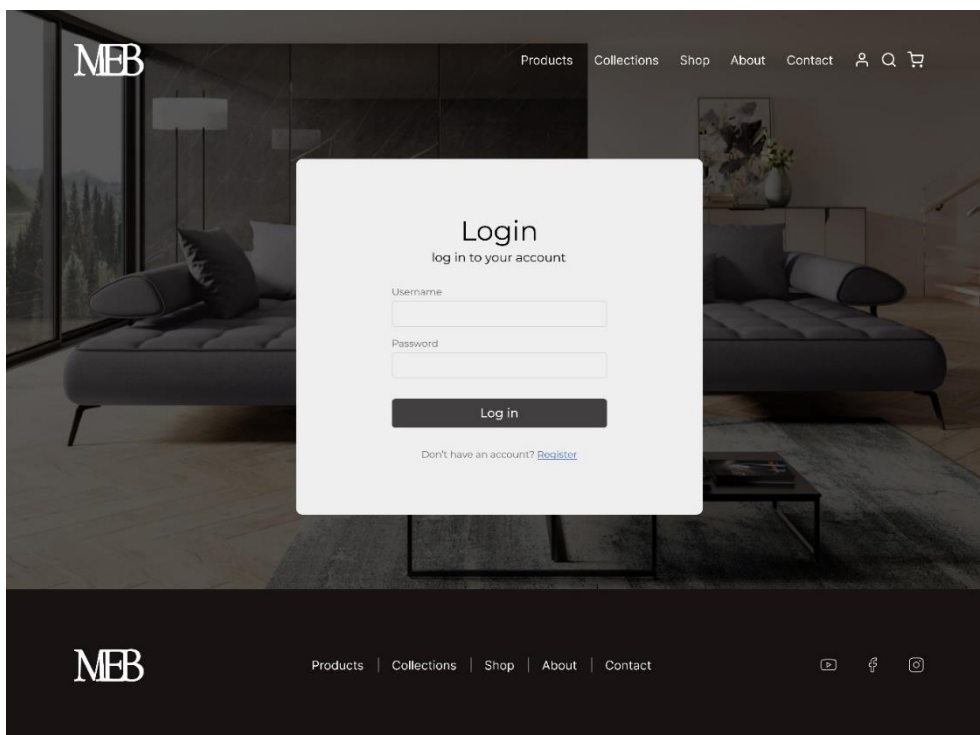


Рисунок 4.5 –Авторизація

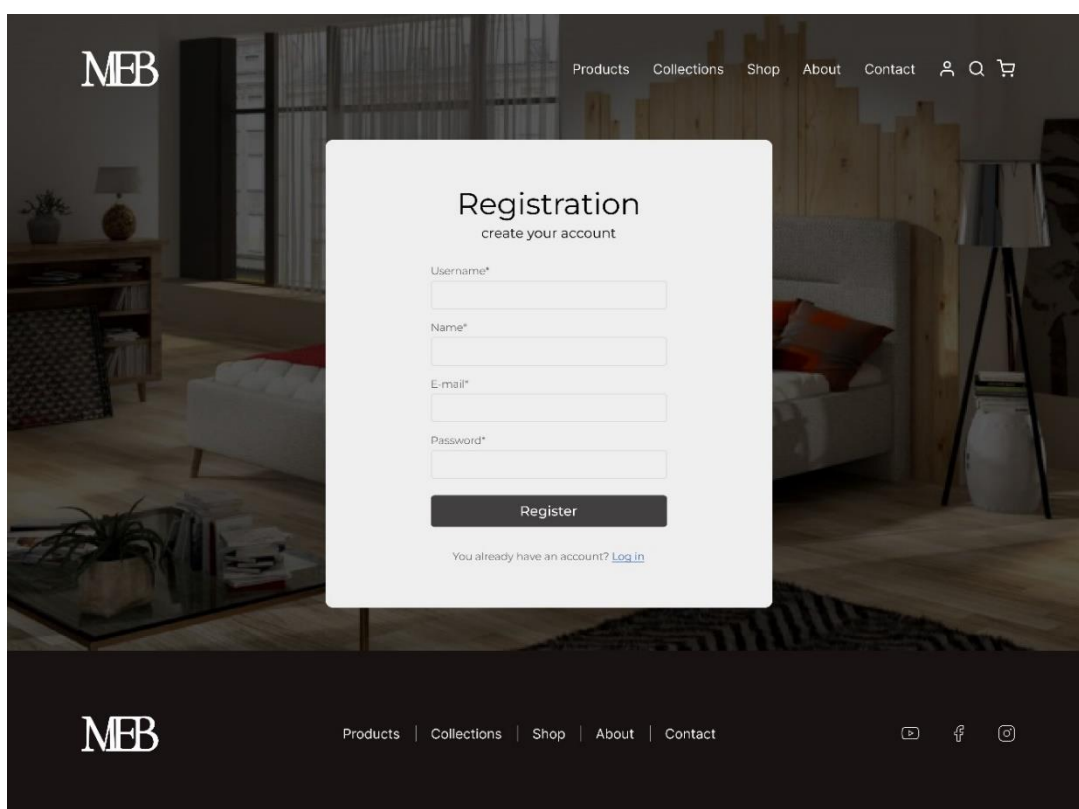


Рисунок 4.6 –Реєстрація

Форми для авторизації та реєстрації з відповідними полями для заповнення та посиланнями на реєстрацію або авторизацію відповідно.

4.2 Реалізація адмін панелі

У цьому підрозділі представлено результати розробки адмін панелі вебзастосунку інтернет-магазину меблів.

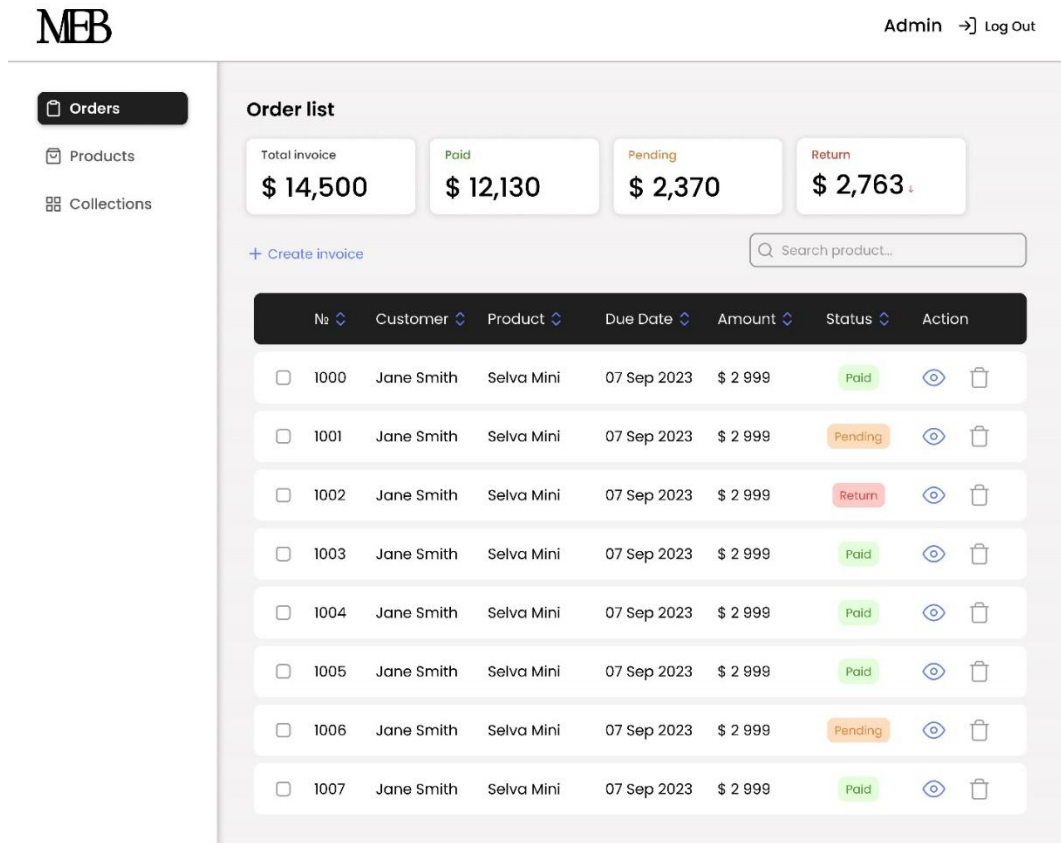


Рисунок 4.7 – Адмін панель (Список замовлень)

Адмін панель має 3 модулі (Замовлення, Каталог та Колекції). **Модуль замовлень** відображає список замовлень та загальний дохід, рахунок відповідно до статусу доставки.

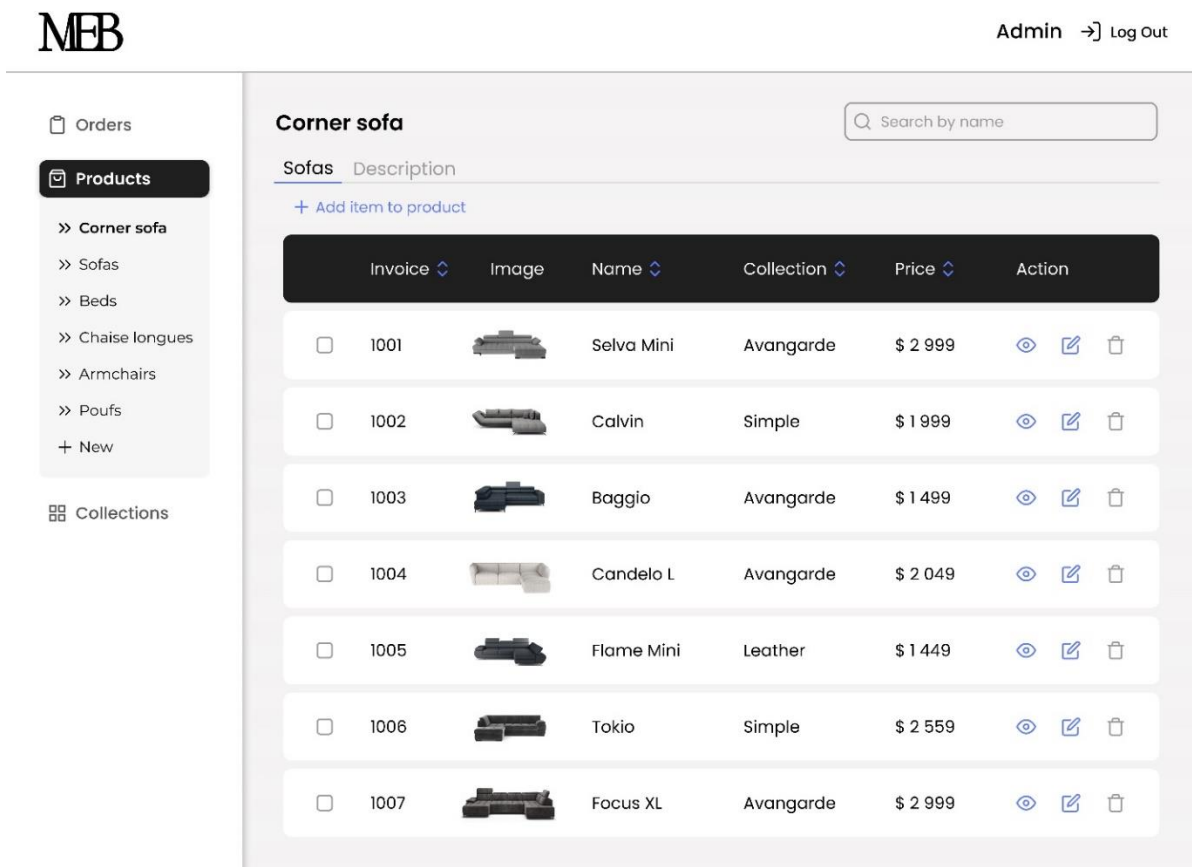


Рисунок 4.8 –Адмін панель (Каталог)

Модуль каталогу відображає список товарів, які відносяться до відповідної категорії. Має можливість додавання нової категорії, товару у категорію, перегляду, редагування та видалення.

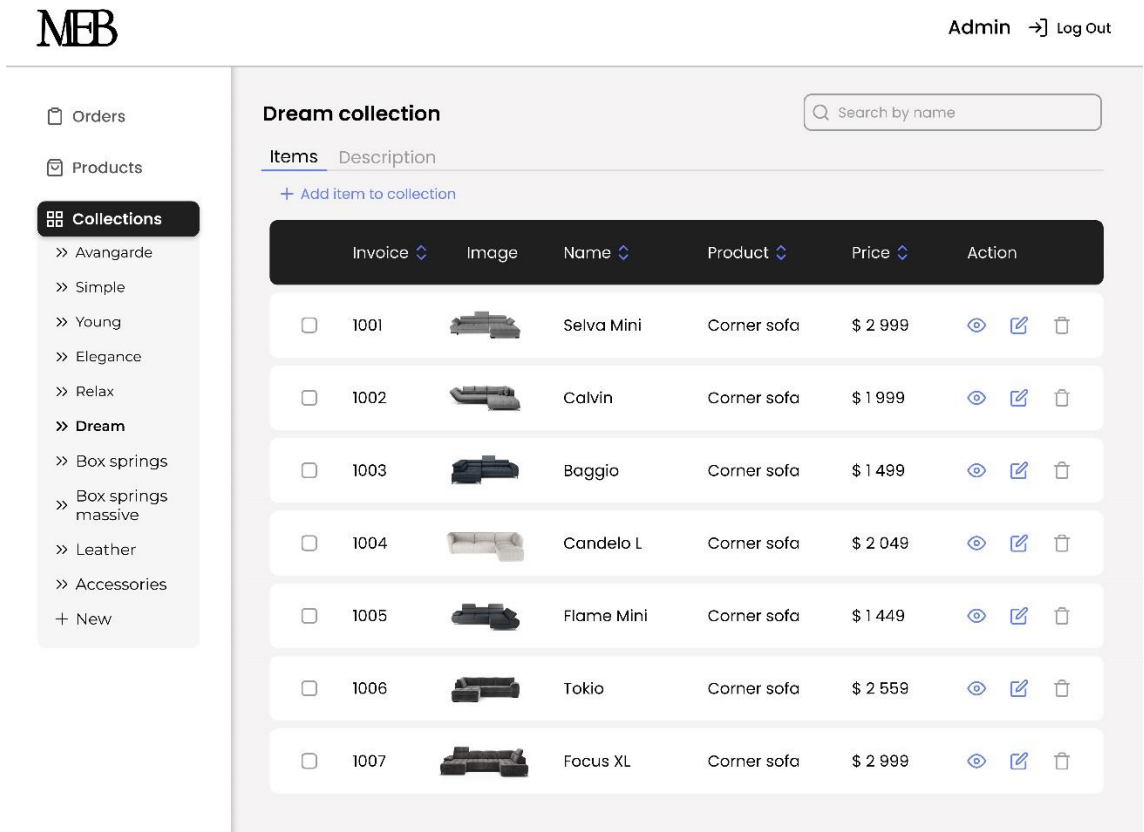


Рисунок 4.9 –Адмін панель (Колекції)

Модуль колекцій відображає список товарів, які відносяться до відповідної колекції. Має можливість додавання нової колекції, товару у колекцію, перегляду, редагування та видалення.

Висновки до розділу 4

У четвертому розділі описано результат розробки програми, а також візуалізовано кожну розроблену сторінку. Розробку описано у двох підрозділах, спочатку реалізацію клієнтської частини, а потім адмін панель вебзастосунку інтернет-магазину меблів.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи розроблено вебзастосунок магазину меблів та сформовано звіт, який складається з чотирьох розділів. Мету досягнуто та поставлені завдання виконано.

Описано предметну область кваліфікаційної роботи, функціонал вебзастосунку, що спрямований на полегшення вибору меблів та оформлення замовлення та розглянуто аналогічні системи. В аналізі існуючих систем виявлено ключові аспекти та вдалі рішення, які можна врахувати під час розробки власної платформи. Проаналізовані Overstock, Wayfair та Ashley Furniture як потенційні конкуренти на ринку продажу меблів за такими критеріями:

- назва;
- розробник (дистриб'ютор);
- архітектура (desktop application, client-server, 3tier web application);
- мова реалізації;
- перелік функцій, характеристик;
- аналіз переваг та недоліків даного ПЗ;
- джерело інформації (вебсайт).

Спроектовано інтернет-магазин меблів. Розроблено такі діаграми як діаграма варіантів використання системи, діаграма класів системи, діаграма послідовностей оформлення замовлення.

Представлено опис технологій розробки вебзастосунку магазину меблів. Розроблено та розглянуто фізичну модель бази даних вебзастосунку, а також описано структуру кожної таблиці окремо. Спроектовано користувацький інтерфейс вебзастосунку.

Описано результат розробки програми, а також візуалізовано кожен розроблену сторінку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Дж. Джеффри, С. Макьюен, Р. Бейкер. Веб-дизайн. Дизайн користувацького інтерфейсу. К.: Вид. група ВНУ. 2017. 496 с.
2. Пітер Морвіль, Джессіка Джеймс, Холлі Розенталь. Інформаційна архітектура для Всесвітньої павутини. 2010. 320 с.
3. Джекі Сонг, Фатима Захеді. Веб-дизайн в електронній комерції: теорія та емпіричний аналіз. 2012. 209 с.
4. Ільченко О. С. Інтернет-маркетинг. Київ: Видавництво "КНЕУ", 2021. 560 с.
5. E-commerce Website Design Best Practices. URL: <https://www.shopify.com/enterprise/ecommerce-website-design> (дата звернення: 09.04.2024).
6. How to Create an E-commerce Website in 8 Easy Steps. URL: <https://www.websitebuilderexpert.com/building-online-stores/how-to-create-an-ecommerce-website/> (дата звернення: 09.04.2024).
7. Bedbathandbeyond URL: <https://www.bedbathandbeyond.com/?ostk=true> (дата звернення: 09.04.2024).
8. Wayfair URL: <https://www.wayfair.com/> (дата звернення: 09.04.2024).
9. Ashley Furniture URL: <https://www.ashleyfurniture.com/> (дата звернення: 09.04.2024).
10. Online Payment Fraud Prevention Techniques. URL: <https://www.paypal.com/us/brc/article/fraud-prevention> (дата звернення: 09.04.2024).
11. HTML Tutorial. URL: <https://www.w3schools.com/html/> (дата звернення: 09.04.2024).
12. CSS Tutorial. URL: <https://www.w3schools.com/css/> (дата звернення: 09.04.2024).
13. JavaScript Tutorial. URL: <https://www.w3schools.com/js/> (дата звернення: 09.04.2024).
14. Bootstrap. URL: <https://getbootstrap.com/docs/> (дата звернення: 09.04.2024).

15. jQuery. URL: <https://jquery.com/> (дата звернення: 09.04.2024).
16. OpenCart. URL: <https://www.opencart.com/> (дата звернення: 09.04.2024).
17. PrestaShop. URL: <https://prestashop.com/> (дата звернення: 09.04.2024).
18. MySQL Tutorial. URL: <https://www.w3schools.com/MySQL/default.asp> (дата звернення: 09.04.2024).
19. Google Trends. URL: <https://trends.google.com/trends/> (дата звернення: 09.04.2024).
20. Google Developers. URL: <https://developers.google.com/> (дата звернення: 09.04.2024).