

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

_____ проф. Приходько С.Б.

«___» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

на тему: Розробка програмного забезпечення інтернет-магазину одягу компанії

“Brand New Outfit”

Виконав: студент групи 4151

_____  Юрченко Д.С.

(підпис, ПІБ)

Керівник роботи:

Доцент, к.т.н. _____

(посада, науковий ступень вчене звання)

_____ Пухалевич А.В.

(підпис, ПІБ)

Миколаїв – 2023 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми

_____ доц. Макарова Л.М.

(підпис)

«___» _____ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти «бакалавр»

Студенту Юрченко Дар'ї Сергіївні

(Прізвище, ім'я, по батькові)

1. Тема роботи: розробка програмного забезпечення інтернет-магазину одягу компанії "Brand New Outfit"

Керівник роботи доцент, к.т.н Пухалевич Андрій Володимирович

Затверджені наказом ректора №257-уч від «17» березня 2023 р.

2. Термін подання роботи: 29.05.2023 р.

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____

Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Аналіз практичної задачі інженерії програмного забезпечення; Проект програмного забезпечення; Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів 1. Титульний слайд; 2. Мета і завдання роботи; 3. Функції програмного забезпечення; 4. Діаграма варіантів використання; 5. Діаграма класів; 6. Логічна модель; 7-8. Фізична модель; 9. Результати розробки; 10-11. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

7. Дата видачі завдання 20.03.2023 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу ВСТУП	24.03.2023	ПП*
2.	Аналіз практичної задачі інженерії ПЗ	31.03.2023	ПП*
3.	Аналіз вимог до ПЗ	07.04.2023	ПП*
4.	Розробка архітектури ПЗ	21.04.2023	
5.	Розробка проекту ПЗ	05.05.2023	
6.	Реалізація та тестування ПЗ	12.05.2023	
7.	Підготовка розділу Результати розробки ПЗ	17.05.2023	
8.	Підготовка розділу з охорони праці	19.05.2023	ПП*
9.	Підготовка розділу ВИСНОВКИ	24.05.2023	
10.	Оформлення списку використаних джерел та додатків	26.05.2023	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	29.05.2023	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
12.	Підготовка презентації та доповіді	02.06.2023	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	05.06.2023	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді, а також Авторського договору з додатком	12.06.2023	

* - за результатами переддипломної практики (ПП), яка була з 01.02.2023 до 19.03.2023 р.

Студент

(підпис)

Юрченко Д.С.

(ПІБ)

Керівник роботи

(підпис)

Пухалевич А.В.

(ПІБ)

АНОТАЦІЯ

Кваліфікаційна робота присвячена розв’язанню задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов, а саме розробці програмного забезпечення інтернет-магазину одягу компанії “Brand New Outfit”.

Проаналізовано практичну задачу розробки програмного забезпечення інтернет-магазину одягу. Проаналізовано існуюче програмне забезпечення інтернет-магазинів. Виконано постановку задачі на розробку програмного забезпечення інтернет-магазину компанії “Brand New Outfit”. Розроблено проєкт та проведено тестування і випробування програмного забезпечення інтернет-магазину компанії “Brand New Outfit”. Розглянуто питання з охорони праці в роботі інтернет-магазину компанії “Brand New Outfit”.

Кваліфікаційна робота викладена на 110 аркушах, містить 23 таблиці, 29 рисунків, 5 додатків, список використаних джерел із 6 найменувань.

ABSTRACT

The qualification work is dedicated to solving the engineering problem software characterized by complexity and the uncertainty of conditions, namely the development of software for the online clothing store of the company “Brand New Outfit”.

The practical task of developing software for an online clothing store is analyzed. The existing software of online stores was analyzed. The task of developing software for the online store of the company “Brand New Outfit” has been completed. The project was developed and software testing of the online store of the company “Brand New Outfit” was carried out. The issue of labor protection in the work of the online store of the company “Brand New Outfit” was considered.

The qualification work is presented on 110 sheets, contains 23 table, 29 drawings, 5 applications, list of used sources from 6 denominations.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РОБОТИ ІНТЕРНЕТ-МАГАЗИНУ ОДЯГУ, ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВІДПОВІДНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КОМПАНІЇ “BRAND NEW OUTFIT”.....	9
1.1 Аналіз практичної задачі розробки програмного забезпечення інтернет-магазину одягу	9
1.2 Аналіз аналогічного існуючого програмного забезпечення інтернет-магазинів	11
1.3 Постановка задачі на розробку програмного забезпечення інтернет-магазину компанії “Brand New Outfit”	14
2 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНУ ОДЯГУ ДЛЯ КОМПАНІЇ “BRAND NEW OUTFIT”.....	18
2.1 Аналіз вимог до програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”	18
2.1.1 Побудова моделі варіантів використання програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”	18
2.1.2 Специфікації варіантів використання	20
2.2 Розробка архітектури програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”	33
2.3 Проєкт програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”	34
2.3.1 Ескізний проєкт програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”	34
2.3.2 Технічний проєкт програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”	40
2.3.3 Робочий проєкт програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”	49
3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНУ ОДЯГУ ДЛЯ КОМПАНІЇ “BRAND NEW OUTFIT”.....	56
4 ОХОРОНА ПРАЦІ	57

ВИСНОВКИ	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	70
ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ...	71
ДОДАТОК Б – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	78
ДОДАТОК В – КЕРІВНИЦТВО КОРИСТУВАЧА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	80
ДОДАТОК Г – ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	85
ДОДАТОК Д – ТЕКСТ ПРОГРАМИНОГО ЗАБЕЗПЕЧЕННЯ	90

ВСТУП

Кваліфікаційна робота передбачає розв'язання практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов: розробку програмного забезпечення інтернет-магазину одягу компанії “Brand New Outfit”.

Існуючі аналоги не можуть повністю задовольнити вимоги до інтернет-магазину компанії “Brand New Outfit” щодо доставки, розміру комісії чи продуктивності вебсайту. Крім того, розробка власної платформи електронної комерції також може забезпечити більший контроль та гнучкість програмного забезпечення. На існуючих платформах можуть бути відсутні функції, параметри налаштування, які будуть необхідні в наступних версіях інтернет-магазину одягу компанії “Brand New Outfit”, а додати ці функції буде неможливо.

Отже, розробка програмного забезпечення інтернет-магазину одягу компанії “Brand New Outfit” дозволить отримати наявність усіх необхідних функцій для забезпечення конкурентної переваги на ринку.

Мета: розробка програмного забезпечення інтернет-магазину одягу компанії “Brand New Outfit”.

Завдання:

1. Проаналізувати практичну задачу розробки програмного забезпечення інтернет-магазину одягу.
2. Проаналізувати існуюче програмне забезпечення інтернет-магазинів.
3. Виконати постановку задачі на розробку програмного забезпечення інтернет-магазину компанії “Brand New Outfit”.
4. Розробити ескізний проєкт програмного забезпечення інтернет-магазину компанії “Brand New Outfit”.
5. Розробити технічний проєкт програмного забезпечення інтернет-магазину компанії “Brand New Outfit”.

6. Розробити робочий проєкт програмного забезпечення інтернет-магазину компанії “Brand New Outfit”.

7. Протестувати програмне забезпечення інтернет-магазину компанії “Brand New Outfit”.

1 АНАЛІЗ ПРАКТИЧНОЇ ЗАДАЧІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РОБОТИ ІНТЕРНЕТ-МАГАЗИНУ ОДЯГУ, ТА ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ ВІДПОВІДНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КОМПАНІЇ “BRAND NEW OUTFIT”

1.1 Аналіз практичної задачі розробки програмного забезпечення інтернет-магазину одягу

Інтернет-магазини одягу стають все популярнішими в останні роки через зростання електронної комерції та зручність покупок в Інтернеті. Ці магазини пропонують широкий вибір одягу, часто за конкурентними цінами, і покупці можуть робити покупки, не виходячи з власного будинку, у будь-який час. Однією з ключових переваг інтернет-магазинів одягу є їхня здатність охопити глобальну аудиторію. Вони не обмежені географічними межами та можуть легко розширити свою клієнтську базу за допомогою онлайн-маркетингу та реклами. Загалом інтернет-магазини, в тому числі інтернет-магазини одягу, зарекомендували себе успішною бізнес-моделлю в індустрії електронної комерції.

Кваліфікаційна робота передбачає розв'язання практичної задачі інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов: розробку програмного забезпечення інтернет-магазину одягу компанії “Brand New Outfit”.

Для вирішення цієї задачі потрібно створити вебсайт, який буде містити, в тому числі, сторінки, необхідні для роботи інтернет-магазину.

Сайт або вебсайт – це є місце розташування веб сторінок, які пов'язані між собою і доступні через головну сторінку вебсайту з допомогою браузера [1].

Зазвичай інтернет-магазини складаються з наступних сторінок:

- список продуктів;
- кошик;
- оформлення замовлення.

Сторінка зі списком продуктів є ключовим компонентом інтернет-магазину, оскільки саме тут покупці можуть переглядати та шукати продукти, доступні для покупки. Як правило, на сторінці зі списком продуктів відображається сітка або список продуктів з їх назвами, цінами та зображеннями. Кожен продукт також має короткий опис або список функцій, щоб допомогти клієнтам зрозуміти, що це за продукт і чи відповідає він їх потребам.

Сторінка зі списком продуктів також має фільтри або параметри сортування, щоб допомогти клієнтам звужити область пошуку та знайти продукти, які найкраще відповідають їхнім потребам. Ці фільтри можуть включати такі параметри, як ціновий діапазон, категорія продукту, бренд, розмір, колір або інші характеристики, характерні для типу продукту, що продається.

Список продуктів також містить посилання на докладніші сторінки продуктів або огляди. Це може допомогти покупцям дізнатися більше про продукт перед покупкою. В цілому, сторінка зі списком продуктів є важливим компонентом будь-якого інтернет-магазину, оскільки вона відіграє ключову роль у залученні та перетворенні потенційних клієнтів на клієнтів, що платять.

Кошик також є важливим компонентом інтернет-магазину, оскільки він дозволяє покупцям додавати товари у своє замовлення. Як правило, у кошику відображаються назва, зображення, ціна та кількість кожного товару в замовленні, а також проміжна сума замовлення.

Після того, як покупець додав усі бажані товари в кошик, він переходить на сторінку оформлення замовлення. На етапі оформлення замовлення покупець завершує покупку, вводячи інформацію про доставку, вибираючи спосіб оплати та переглядаючи деталі замовлення і загальну вартість замовлення перед його відправкою. Як правило, на етапі оформлення замовлення відображається форма, в якій покупець може ввести особисту інформацію, а також варіанти доставки та оплати.

1.2 Аналіз аналогічного існуючого програмного забезпечення інтернет-магазинів

Для розробки програмного забезпечення інтернет-магазину “Brand New Outfit” важливо провести аналіз існуючих аналогів, щоб визначити кращі практики та ключові функції, які можуть бути використані, щоб забезпечити конкурентну перевагу на висококонкурентному ринку онлайн-торгівлі.

Однією з найпопулярніших і найуспішніших платформ електронної комерції є Amazon (рисунок 1.1). Amazon надає широкий спектр продуктів та послуг, включаючи одяг, зручності та персоналізації. Він також включає методи аналізу даних і машинного навчання для надання персоналізованих рекомендацій і оптимізації цін. Ще однією ключовою особливістю Amazon є його широка мережа доставки, яка забезпечує швидку та надійну доставку, але наразі в Україні доставки немає.

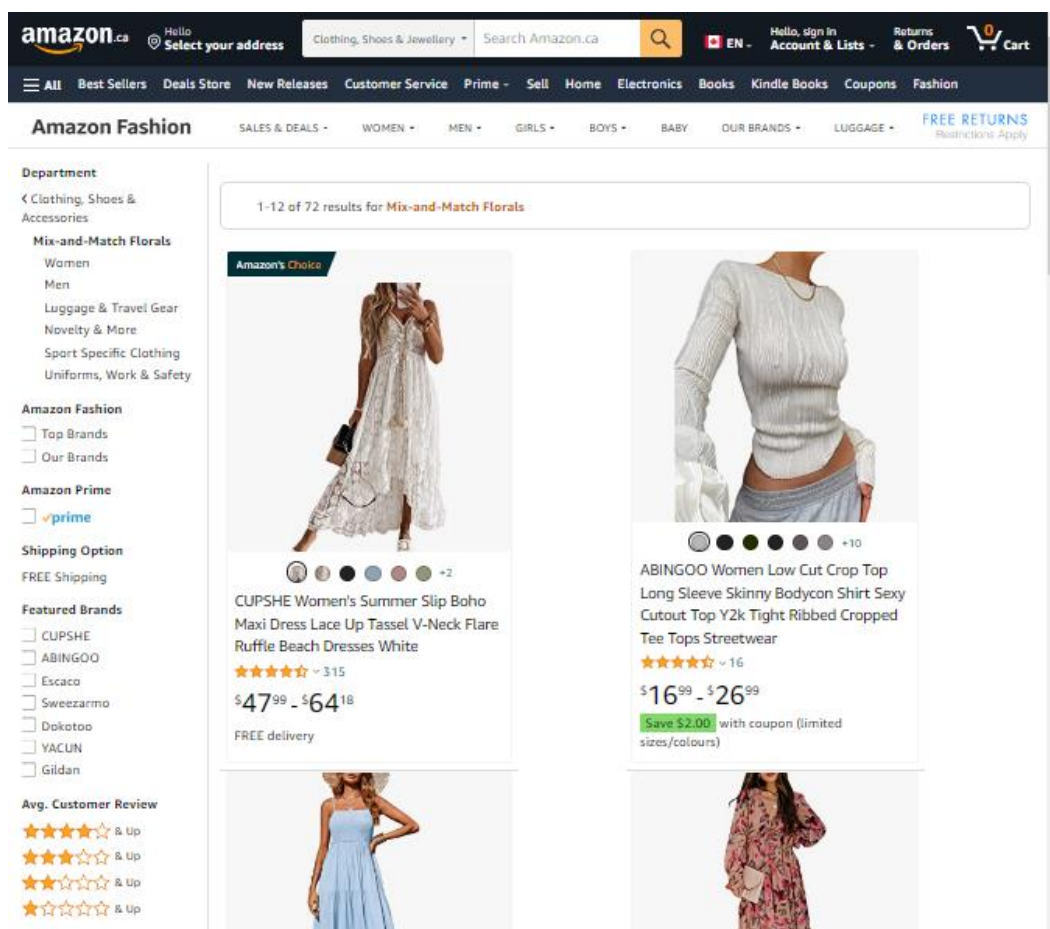


Рисунок 1.1 – Скріншот з інтернет-магазину Amazon

Rozetka - один із найбільших інтернет-магазинів України, що дозволяє компаніям продавати власний асортимент товарів, в тому числі і одяг (рисунк 1.2). Однак, є деякі недоліки у використанні цієї платформи. Наприклад, за розміщення власних товарів на платформі стягується комісія, що може стати на заваді малому бізнесу або приватним особам, які бажають продавати свою продукцію в Інтернеті. Ця плата може утримувати деяких продавців від використання платформи.

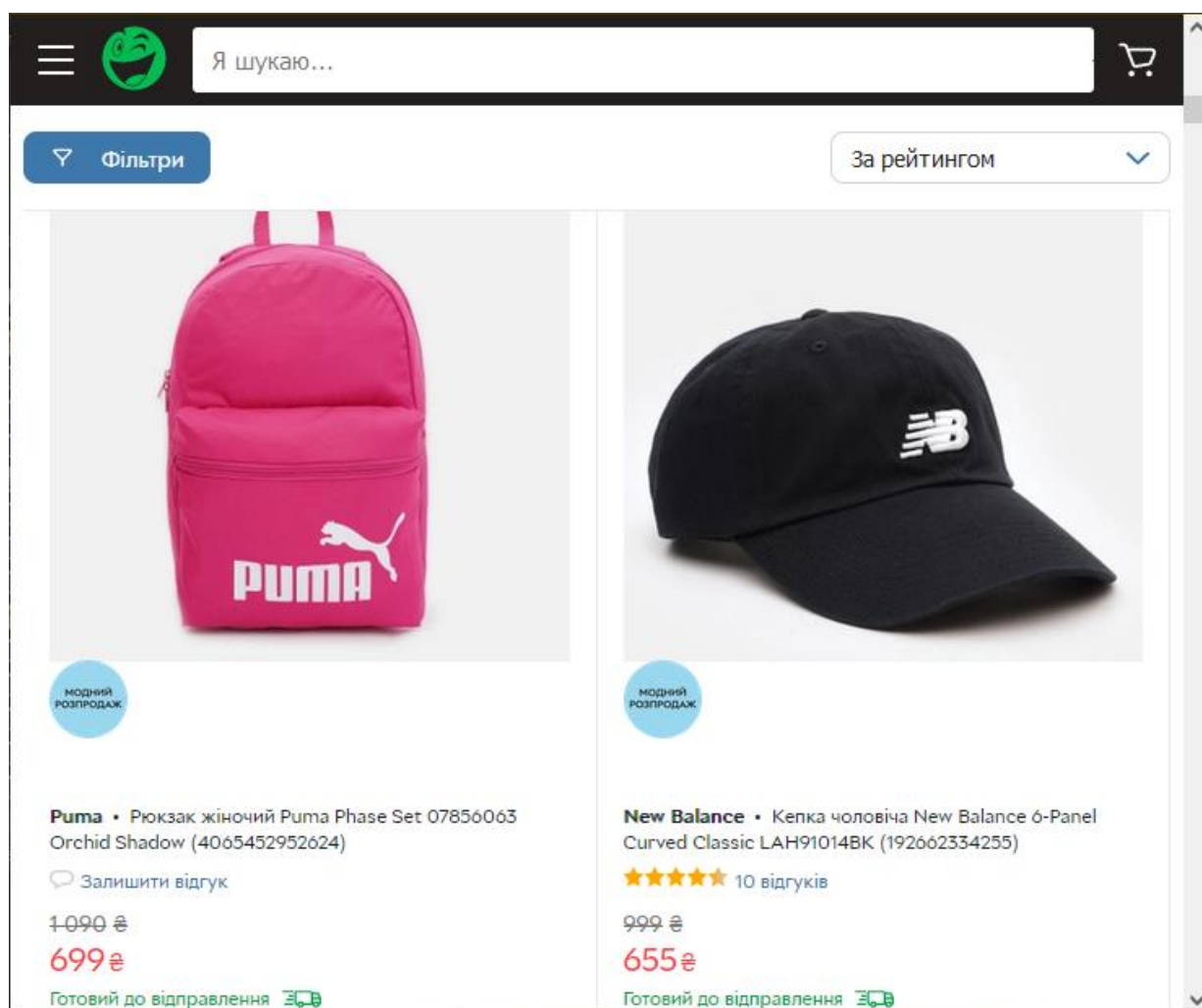


Рисунок 1.2 – Скріншот з інтернет-магазину Rozetka

WooCommerce – плагін електронної комерції з відкритим вихідним кодом для WordPress (рисунк 1.3). Це дозволяє користувачам створювати та керувати інтернет-магазином, який продає як фізичні, так і цифрові продукти. WooCommerce вимагає від користувачів певного рівня технічних знань для налаштування та використання. Незважаючи на те, що це зручна платформа, необхідне деяке знання концепцій WordPress, щоб скористатися всіма її можливостями та функціями. Основний недолік полягає в тому, що WooCommerce може стати повільним та

ресурсномістким, якщо на веб-сайті багато товарів та трафіку. Це може вплинути на продуктивність вебсайту та взаємодію з користувачем, вимагаючи додаткових заходів оптимізації для забезпечення безперебійної роботи. Хоча основний плагін WooCommerce є безкоштовним, користувачам може знадобитися заплатити за преміальні теми та розширення для доступу до деяких розширених функцій. Це може збільшити загальну вартість використання платформи, особливо підприємств з обмеженим бюджетом.

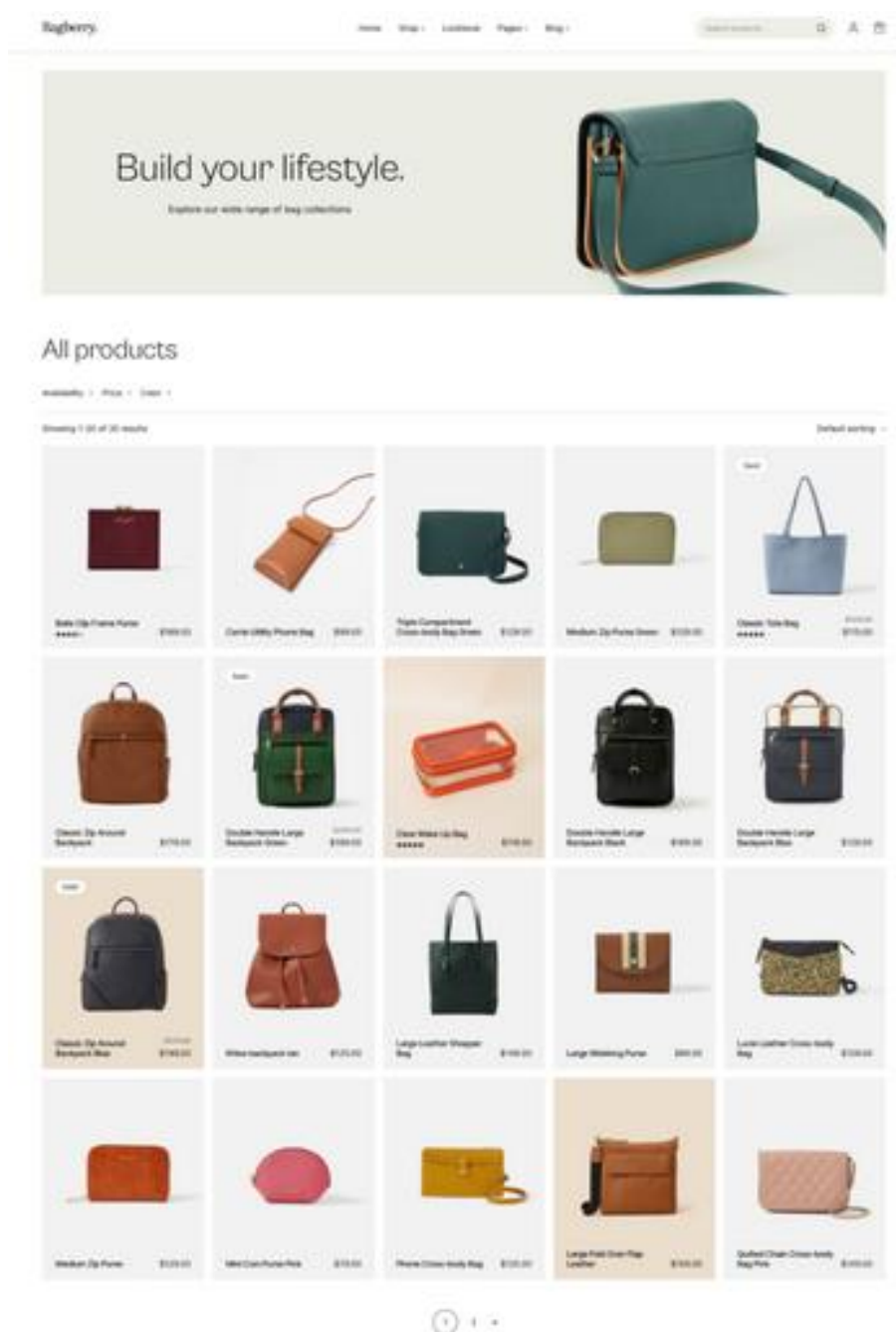


Рисунок 1.3 – Скріншот з інтернет-магазину створеного за допомогою WooCommerce плагіну

Розглянувши аналоги, було прийнято рішення розробити власний інтернет-магазин одягу компанії “Brand New Outfit”. Це рішення засноване на кількох обґрунтуваннях.

По-перше, незважаючи на те, що на ринку існує багато платформ електронної комерції, жодна з них не може повністю задовольнити вимоги компанії “Brand New Outfit” щодо доставки, розміру комісії чи продуктивності вебсайту.

По-друге, розробка власної платформи електронної комерції також може забезпечити більший контроль та гнучкість програмного забезпечення.

По-третє, на існуючих платформах можуть бути відсутні функції, параметри налаштування, які будуть необхідні в наступних версіях інтернет-магазину одягу компанії “Brand New Outfit”, а додати ці функції буде неможливо.

Отже, розробка програмного забезпечення інтернет-магазину одягу компанії “Brand New Outfit” дозволить отримати наявність усіх необхідних функцій для забезпечення конкурентної переваги на ринку.

1.3 Постановка задачі на розробку програмного забезпечення інтернет-магазину компанії “Brand New Outfit”

На підставі попереднього аналізу можна сформулювати практичну задачу інженерії програмного забезпечення, що характеризується комплексністю та невизначеністю умов: потрібно розробити програмне забезпечення інтернет-магазину одягу компанії “Brand New Outfit”, яке задовольняє наступним вимогам:

- реєстрація;
- авторизація;
- керування обліковим записом;
- вихід з облікового запису;
- перегляд товару;
- пошук товару;
- керування кошиком;

- оформлення замовлення;
- керування замовлення;
- керування товарами;
- керування брендами;
- керування типами.

Вимоги до організації вхідних та вихідних даних:

Реєстрація:

Вхідні дані: особиста інформація користувача (ім'я, адреса електронної пошти, пароль).

Вихідні дані: користувачу надіслано електронний лист із підтвердженням повідомлення про успішну реєстрацію.

Авторизація:

Вхідні дані: облікові дані користувача (електронна пошта, пароль).

Вихідні дані: повідомлення про успішний вхід.

Керування обліковим записом:

Вхідні дані: інформація про обліковий запис користувача (ім'я, адреса електронної пошти, пароль), оновлені дані облікового запису.

Вихідні дані: повідомлення про успішне оновлення інформації облікового запису, повідомлення про успішну зміну пароля, повідомлення про успішне видалення облікового запису.

Вихід з облікового запису:

Вхідні дані: ніяких конкретних даних не потрібно.

Вихідні дані: повідомлення про вихід користувача, перенаправлення на сторінку входу.

Перегляд товару:

Вхідні дані: ідентифікатор продукту або деталі продукту.

Вихідні дані: деталі товару (назва, опис, ціна тощо), зображення товару.

Пошук товару:

Вхідні дані: пошуковий запит (ключові слова, фільтри).

Вихідні дані: список відповідних товарів, деталі товару.

Керування кошиком:

Вхідні дані: ідентифікатор продукту або деталі продукту для додавання/вилучення з кошика.

Вихідні дані: поточний вміст кошика (перелік товарів, кількість, загальна ціна).

Оформлення замовлення:

Вхідні дані: вміст кошика, платіжні реквізити.

Вихідні дані: деталі підтвердження замовлення (ідентифікатор замовлення, загальна вартість)

Управління замовленнями:

Вхідні дані: ID замовлення або деталі замовлення.

Вихідні дані: деталі замовлення (статус замовлення, елементи, кількість, ціни), історія замовлень користувача.

Керування товаром:

Вхідні дані: деталі товару (назва, опис, ціна), інформація про бренд (назва, опис тощо), інформація про тип.

Вихідні дані: повідомлення про успішне створення/редагування/видалення товару, оновлений список товарів.

Керування брендом:

Вхідні дані: інформація про бренд (назва, опис).

Вихідні дані: повідомлення про успішне створення/редагування/видалення бренду, оновлений список брендів.

Керування типом:

Вхідні дані: введіть деталі (назва, опис тощо)

Вихідні дані: повідомлення про успішне створення/редагування/видалення типу, оновлений список типів.

Вимоги до складу й параметрів технічних засобів:

Вимоги до користувацького обладнання:

- процесор: будь-який сучасний процесор, здатний безперебійно працювати з веб-переглядачем, наприклад Intel Core i3;

- оперативна пам'ять: щонайменше 4 ГБ оперативної пам'яті для плавного перегляду та взаємодії з веб-сайтом;
- дисковий простір: достатньо вільного місця на диску для зберігання тимчасових файлів і кешу браузера;
- роздільна здатність екрана: для оптимального перегляду та зручності використання рекомендована мінімальна роздільна здатність екрана 1280x800 або вище.

Серверне обладнання для інтернет-магазину одягу повинно включати:

- 8-ядерний процесор (наприклад, Intel Xeon або AMD EPYC);
- не менше 32 ГБ оперативної пам'яті;
- жорсткий диск (SSD) з мінімальною ємністю 256 ГБ чи вище.

Ці специфікації забезпечують стабільність, продуктивність та можливості зберігання для операцій інтернет-магазину.

Вимоги до інформаційної та програмної сумісності:

Серверна операційна система для програмного забезпечення має бути Linux Ubuntu 22.04.2 LTS.

Програмне забезпечення має бути сумісним із популярними веббраузерами, такими як Google Chrome, Mozilla Firefox, Microsoft Edge, які підтримують HTML5 та JavaScript ES6.

2 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНУ ОДЯГУ ДЛЯ КОМПАНІЇ “BRAND NEW OUTFIT”

2.1 Аналіз вимог до програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”

У цьому підрозділі описані варіанти використання, їх специфікації, діаграми діяльності, а також інтерфейс програмного забезпечення.

2.1.1 Побудова моделі варіантів використання програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”

Виявлені актори програмного забезпечення показані у таблиці 2.1.

Таблиця 2.1 – Актори програмного забезпечення

Актор	Короткий опис
Користувач	фізична або юридична особа – фактичний одержувач товару
Адміністратор	технічний спеціаліст, який здійснює контроль та управління Інтернет-магазином

Виявлені прецеденти показані у таблиці 2.2.

Таблиця 2.2 – Прецеденти програмного забезпечення

Актор	Назва прецеденту	Короткий опис
Користувач	Реєстрація	Реєстрація користувача на сайті
	Авторизація	Вхід користувача в систему для доступу до відповідних функцій
	Перегляд товару	Користувач має можливість переглянути товар на сайті

Продовження таблиці 2.2

	Пошук товару	Пошук потрібних товарів
	Керування обліковим записом	Користувач може оновлювати особисту інформацію, змінювати паролі або видалити обліковий запис
	Керування кошиком	Керування кошиком для покупок, зокрема додавання товарів, видалення товарів
	Оформлення замовлення	Користувач вводить платіжні данні і підтверджує замовлення
	Вихід з облікового запису	Вихід користувача із системи
Адміністратор	Авторизація	Вхід адміністратора в систему для доступу до відповідних функцій
	Перегляд товару	Адміністратор має можливість переглянути товар на сайті
	Пошук товару	Пошук потрібних товарів
	Керування товарами	Дозволяє йому виконувати адміністративні завдання, пов'язані з товарами, наприклад додавати, редагувати або видаляти товар
	Керування брендами	Адміністратор може керувати брендами, пов'язаними з товарами, зокрема додавати, редагувати або видаляти бренди
	Керування типами	Адміністратор може керувати типами (категоріями) товарів, зокрема додавати, редагувати або видаляти типи
	Вихід з облікового запису	Вихід адміністратора із системи

Проаналізувавши функціональні характеристики та вимоги програмного забезпечення, нижче представлена діаграма варіантів використання для інтернет-магазину одягу “Brand New Outfit” (рисунок 2.1).

2.1.2 Специфікації варіантів використання

Проаналізувавши діаграму варіантів використання (рисунок 2.1), можна навести їх специфікації (таблиці 2.3-2.14).

Таблиця 2.3 – Специфікація варіанту використання “Реєстрація”

Варіант використання	Реєстрація користувача
Опис	Користувач створює новий обліковий запис
Актори	Користувач
Передумови	Немає
Базовий потік	- Користувач переходить на сторінку реєстрації. - Користувач вводить свою особисту інформацію. - Користувач надсилає реєстраційну форму. - Система перевіряє надану інформацію. - Система створює новий обліковий запис користувача. - Користувач отримує електронний лист із підтвердженням. - Користувач підтверджує свою адресу електронної пошти, натиснувши на посилання підтвердження. - Система підтверджує успішну реєстрацію.
Пост умови	Обліковий запис користувача створено, і користувач увійшов у систему
Альтернативний потік	- Користувача вже зареєстровано на сайті. - Користувачу відображається повідомлення.

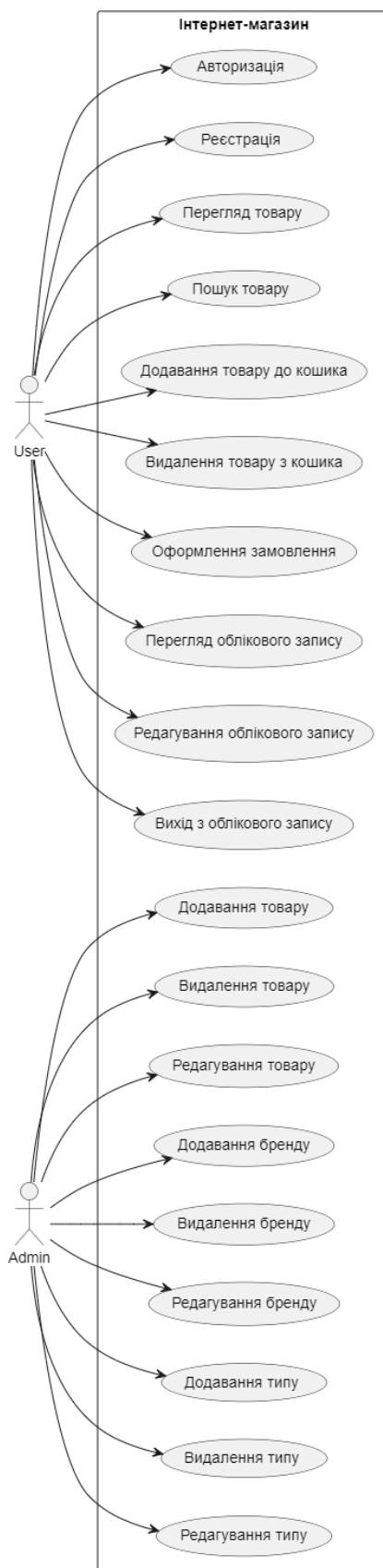


Рисунок 2.1 – Діаграма варіантів використання

Таблиця 2.4 – Специфікація варіанту використання “Авторизація”

Варіант використання	Авторизація користувача
Опис	Передбачається вхід користувача та адміністратора в систему для доступу до відповідних функцій
Актори	Користувач, Адміністратор
Передумови	Користувач повинен мати зареєстрований обліковий запис
Базовий потік	- Користувач вводить свою електронну адресу та пароль. - Користувач надсилає форму входу. - Система перевіряє надані облікові дані. - Система аутентифікує користувача. - Система перенаправляє користувача на домашню сторінку або сторінку, яку раніше запитували.
Пост умови	Успішний вхід у свій обліковий запис
Альтернативний потік	- Система відображає повідомлення про помилку, яке вказує на те, що облікові дані недійсні. - Користувачеві буде запропоновано повторно ввести облікові дані для входу. - Користувач повторно вводить облікові дані для входу. - Система перевіряє облікові дані. - Якщо облікові дані все ще неправильні, повторіть кроки 1-4.

Таблиця 2.5 – Специфікація варіанту використання “Керування обліковим записом”

Варіант використання	Керування обліковим записом
Опис	Дозволяє керувати своїм обліковим записом, зокрема оновлювати особисту інформацію, змінювати паролі або видалити обліковий запис
Актори	Користувач
Передумови	Користувач повинен бути авторизований
Базовий потік	- Користувач авторизується та отримує доступ до розділу керування обліковим записом. - Користувач вибирає опцію «Редагувати обліковий запис». - Користувач змінює свою особисту інформацію. - Користувач надсилає оновлену інформацію. - Система перевіряє зміни та оновлює інформацію облікового запису користувача. - Система сповіщає користувача про успішне оновлення.

Продовження таблиці 2.5

Пост умови	Інформація про обліковий запис оновлюється або обліковий запис видаляється
Альтернативний потік	- Користувач натискає на посилання «Забув пароль». - Система пропонує користувачеві ввести адресу електронної пошти. - Користувач вводить свою електронну адресу. - Система перевіряє адресу електронної пошти та надсилає посилання для скидання пароля на електронну адресу користувача. - Користувач отримує електронний лист для зміни пароля та натискає надане посилання. - Система перевіряє посилання для скидання пароля. - Система відображає форму скидання пароля для користувача. - Користувач вводить новий пароль і підтверджує його. - Система перевіряє новий пароль і оновлює обліковий запис користувача новим паролем. - Система відображає повідомлення про підтвердження, яке вказує на те, що пароль успішно скинуто. - Користувач перенаправляється на сторінку входу.

Таблиця 2.6 – Специфікація варіанту використання “Вихід з облікового запису”

Варіант використання	Вихід з облікового запису
Опис	Вихід зі свого облікового запису
Актори	Користувач. Адміністратор
Передумови	Користувач повинен бути авторизований
Базовий потік	- Користувач увійшов в систему. - Користувач вибирає опцію «Вийти». - Система завершує сеанс користувача та перенаправляє його на сторінку входу.
Пост умови	Користувач/Адміністратор вийшов із системи та перенаправлений на домашню сторінку
Альтернативний потік	- Система відображає повідомлення про помилку, яке вказує на проблему. - Користувачеві буде запропоновано спробувати ще раз або звернутися до служби підтримки по допомогу.

Таблиця 2.7 – Специфікація варіанту використання “Перегляд товару”

Варіант використання	Перегляд товару
Опис	Дозволяє Користувачеві та Адміністратору переглядати товари, доступні в каталозі системи
Актори	Користувач, Адміністратор
Передумови	Немає
Базовий потік	- Користувач відкриває інтерфейс системи. - Користувач вибирає опцію «Переглянути товар». - Система отримує список товарів із бази даних. - Система відображає список товарів для користувача.

Продовження таблиці 2.7

Пост умови	Користувач/Адміністратор бачить товари, доступні в каталозі системи
Альтернативний потік	- Система відображає повідомлення про помилку, яке вказує на те, що товар недоступний. - Користувачеві пропонуються альтернативні товари або пропонується виконати новий пошук.

Таблиця 2.8 – Специфікація варіанту використання “Пошук товару”

Варіант використання	Пошук товару
Опис	Користувач шукає конкретні товари
Актори	Користувач, Адміністратор
Передумови	Немає
Базовий потік	- Користувач відкриває інтерфейс системи. - Користувач вводить пошуковий запит у поле пошуку. - Користувач надсилає пошуковий запит. - Система виконує операцію пошуку на основі запиту. - Система отримує відповідні продукти з бази даних. - Система відображає користувачеві результати пошуку.
Пост умови	Користувачеві надаються результати пошуку за введеними ключовими словами
Альтернативний потік	- Система відображає повідомлення про те, що не знайдено товар, який відповідають пошуковому запиту. - Користувачеві пропонується уточнити пошуковий запит або дослідити інші категорії.

Таблиця 2.9 – Специфікація варіанту використання “Керування кошиком”

Варіант використання	Керування кошиком
Опис	Надає функції для керування кошиком для покупок, зокрема додавання, видалення товарів
Актори	Користувач
Передумови	Клієнт авторизований і має товари в кошику
Базовий потік	- Користувач шукає певний товар. - Користувач вибирає потрібний товар із результатів пошуку. - Користувач натискає кнопку «Додати до кошика». - Система перевіряє наявність товару та кількість на складі. - Система додає товар до кошика користувача. - Система оновлює проміжний підсумок кошика та загальну кількість.
Пост умови	Кошик оновлено з бажаними змінами
Альтернативний потік	- Користувач переглядає кошик для покупок. - Користувач визначає товар, який потрібно видалити. - Користувач натискає кнопку «Видалити» поруч із товаром. - Система видаляє товар із кошика користувача. - Система оновлює проміжний підсумок кошика та загальну кількість.

Таблиця 2.10 – Специфікація варіанту використання “Оформлення замовлення”

Варіант використання	Оформлення замовлення
Опис	Введення платіжних даних і підтвердження замовлення.
Актори	Користувач
Передумови	Клієнт авторизований і має товари в кошику
Базовий потік	- Користувач переглядає кошик для покупок. - Користувач перевіряє товари та кількість у кошику. - Користувач натискає кнопку «Оформити замовлення». - Система пропонує користувачу надати інформацію про доставку та рахунок. - Користувач вводить необхідні дані. - Користувач вибирає бажаний спосіб оплати. - Користувач підтверджує замовлення. - Система перевіряє інформацію та обробляє платіж. - Система генерує підтвердження замовлення з унікальним ідентифікатором замовлення.
Пост умови	Замовлення розміщено, і клієнт отримує підтвердження
Альтернативний потік	- Система відображає повідомлення про помилку, яке вказує на те, що кошик порожній. - Перед тим, як перейти до оформлення замовлення, користувачеві пропонується додати продукти в кошик.

Таблиця 2.11 – Специфікація варіанту використання “Керування замовленнями”

Варіант використання	Керування замовленнями
Опис	Дозволяє керувати замовленнями користувачів, наприклад переглядати замовлення, оновлювати статус замовлення
Актори	Адміністратор
Передумови	Адміністратор авторизований
Базовий потік	- Адміністратор відкриває інтерфейс системи. - Адміністратор вибирає опцію «Керувати замовленнями». - Система отримує список замовлень із бази даних. - Система відображає список замовлень для адміністратора.
Пост умови	Інформація про замовлення оновлюється в системі
Альтернативний потік	- Адміністратор відкриває інтерфейс системи. - Адміністратор вибирає опцію «Керувати замовленнями». - Система отримує список замовлень із бази даних. - Система відображає список замовлень для адміністратора. - Адміністратор вибирає конкретне замовлення для оновлення. - Адміністратор змінює статус замовлення (наприклад, з «Обробляється» на «Відправлено»). - Адміністратор зберігає оновлений статус замовлення. - Система оновлює статус замовлення в базі даних.

Таблиця 2.12 – Специфікація варіанту використання “Керування товарами”

Варіант використання	Керування товарами
Опис	Дозволяє виконувати адміністративні завдання, пов’язані з товарами, наприклад додавати, редагувати або видаляти товар
Актори	Адміністратор
Передумови	Адміністратор увійшов
Базовий потік	- Адміністратор відкриває інтерфейс системи. - Адміністратор вибирає опцію «Керувати товарами». - Система отримує список товарів із бази даних. - Система відображає список товарів для адміністратора. - Адміністратор вибирає опцію «Додати товар». - Адміністратор надає необхідну інформацію для нового товару. - Адміністратор надсилає інформацію про новий товар. - Система додає новий товар до бази даних.
Пост умови	Адміністратор успішно керує вибраним товаром
Альтернативний потік	- Адміністратор відкриває інтерфейс системи. - Адміністратор вибирає опцію «Керувати товарами». - Система отримує список товарів із бази даних. - Система відображає список товарів для адміністратора. - Адміністратор вибирає товар для видалення. - Адміністратор підтверджує видалення. - Система видаляє товар із бази даних.

Таблиця 2.13 – Специфікація варіанту використання “Керування брендами”

Варіант використання	Керування брендами
Опис	Дозволяє виконувати адміністративні завдання, пов’язані з брендами, наприклад додавати, редагувати або видаляти бренд
Актори	Адміністратор
Передумови	Адміністратор увійшов
Базовий потік	- Адміністратор відкриває інтерфейс системи. - Адміністратор вибирає опцію «Керувати брендами». - Система отримує список брендів із бази даних. - Система відображає список брендів для адміністратора. - Адміністратор вибирає опцію «Додати бренд». - Адміністратор надає необхідну інформацію для нового бренду. - Адміністратор надсилає нові відомості про бренд. - Система додає новий бренд до бази даних.
Пост умови	Адміністратор успішно керує вибраним брендом
Альтернативний потік	- Адміністратор відкриває інтерфейс системи. - Адміністратор вибирає опцію «Керувати брендами». - Система отримує список брендів із бази даних. - Система відображає список брендів для адміністратора. - Адміністратор вибирає бренд для видалення. - Адміністратор підтверджує видалення. - Система видаляє бренд із бази даних.

Таблиця 2.14 – Специфікація варіанту використання “Керування типами”

Варіант використання	Керування типами
Опис	Може керувати типами (категоріями) товарів, зокрема додавати, редагувати або видаляти типи
Актори	Адміністратор
Передумови	Адміністратор увійшов
Базовий потік	- Адміністратор відкриває інтерфейс системи. - Адміністратор вибирає опцію «Керувати типами». - Система отримує список типів із бази даних. - Система відображає список типів для адміністратора. - Адміністратор вибирає опцію «Додати тип». - Адміністратор надає необхідну інформацію для нового типу. - Адміністратор надсилає деталі нового типу. - Система додає новий тип до бази даних.
Пост умови	Адміністратор успішно керує вибраним типом
Альтернативний потік	- Адміністратор відкриває інтерфейс системи. - Адміністратор вибирає опцію «Керувати типами». - Система отримує список типів із бази даних. - Система відображає список типів для адміністратора. - Адміністратор вибирає тип для видалення. - Адміністратор підтверджує видалення. - Система видаляє тип із бази даних.

2.2 Розробка архітектури програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”

Вибір та представлення архітектури програмного забезпечення має вирішальне значення для успішної розробки інтернет-магазину одягу “Brand New Outfit”. Вибрана програмна архітектура повинна забезпечувати масштабоване, гнучке та зручне у супроводі рішення, здатне впоратися зі складністю та невизначеністю проєкту.

Подання архітектури програмного забезпечення має включати докладний огляд загальної архітектури системи, включаючи компоненти, взаємодії та інтерфейси між різними модулями. Архітектура програмного забезпечення також повинна описувати стек технологій, включаючи мови програмування, фреймворки, бібліотеки та інструменти, що використовуються для розробки інтернет-магазину одягу.

Одним з підходів до архітектури програмного забезпечення є популярний і широкого використання шаблон архітектури – MVC тобто Model-View-Controller. Це гнучка та масштабована архітектура, яка розділяє додаток на три основні компоненти: модель, вид та контролер. Такий підхід забезпечує чіткий поділ завдань, спрощуючи розробку, обслуговування та тестування програми.

Принцип MVC полягає в тому, щоб розділити додаток на 3 основні частини, відомі як Модель, Вид (перегляд) і Контролер [2]. Компонент Model представляє дані та бізнес-логіку програми. Він взаємодіє з базою даних та надає необхідні дані компоненту View. Компонент View відповідає за відображення даних користувачеві та взаємодію з інтерфейсом користувача. Компонент Controller діє як посередник між компонентами Model та View, отримуючи вхідні дані від користувача, обробляючи їх та відповідним чином оновлюючи Model та View (рисунк 2.2).

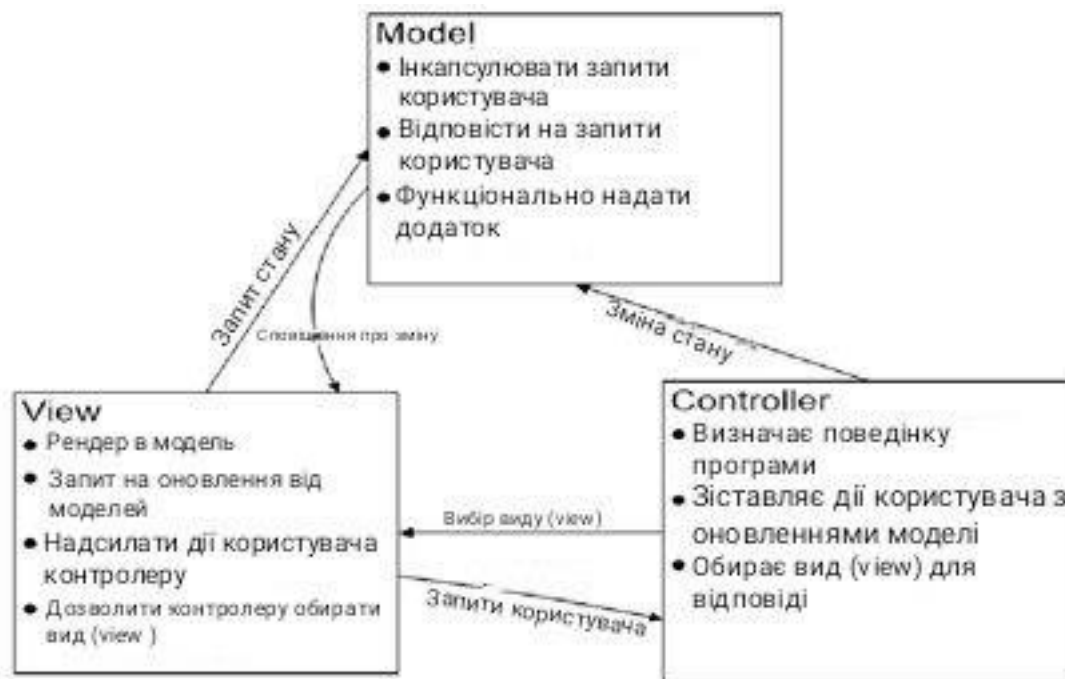


Рисунок 2.2 - Шаблон архітектури – MVC(Model-view-controller)

2.3 Проєкт програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”

2.3.1 Ескізний проєкт програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”

Ескізний проєкт програмного забезпечення є керівництвом для розробників, який забезпечує чітке розуміння структури системи, функцій та вимог до реалізації. Це забезпечує успішну розробку та впровадження програмної системи.

Діаграми діяльності

Діаграма діяльності - це інструмент для моделювання, аналізу та передачі потоку дій у системі чи процесі. Це допомагає виявляти залежності, приймати обґрунтовані рішення та підвищувати ефективність та результативність робочих процесів. Нижче представлені діаграми діяльності для інтернет-магазину компанії “Brand New Outfit”.

На наступній діаграмі діяльності (рисунок 2.3) користувач починає з введення своєї пошти та пароля. Система перевіряє, чи дійсні введені облікові дані. Якщо

вони дійсні, користувач автентифікується, і відображається домашня сторінка. Якщо автентифікація не вдається, відображається повідомлення про помилку, і користувачеві пропонується ввести свої облікові дані ще раз. Якщо введені облікові дані недійсні, відображається відповідне повідомлення про помилку, і користувачеві пропонується ввести свою пошту та пароль ще раз.

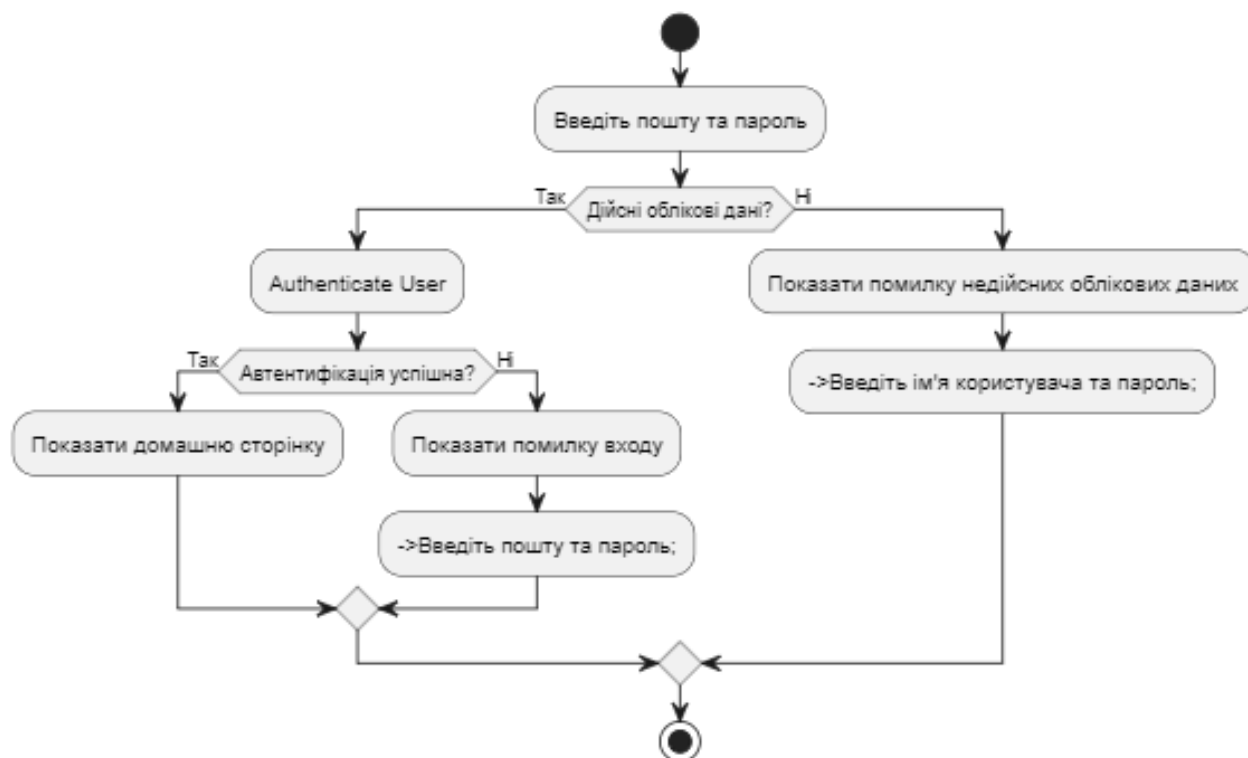


Рисунок 2.3 – Діаграма діяльності варіанту використання “Авторизація”

Нижче наведена діаграма діяльності (рисунок 2.4) відображає потік дій у процесі покупки. Користувач спочатку вибирає товар, потім додає його в кошик і переходить до оформлення замовлення. Якщо товар є на складі, їм буде запропоновано вказати адресу доставки. Якщо користувач авторизований, адреса отримується безпосередньо. В іншому випадку користувачеві пропонується увійти або зареєструватися. Після вказівки адреси доставки користувач переглядає замовлення. Якщо все правильно, замовлення відправляється, і починається процес оплати. Якщо платіж пройшов успішно, з’явиться підтвердження замовлення. У разі виникнення будь-яких проблем, таких як відсутність товару в наявності або помилка оплати, відображаються відповідні повідомлення.



Рисунок 2.4 – Діаграма діяльності процесу купівлі товару

На діаграмі діяльності (рисунок 2.5) користувач робить замовлення. Система приймає замовлення і підтверджує його. Якщо замовлення дійсне, система обробляє оплату. Якщо оплата пройшла успішно, система готує замовлення до відправлення та сповіщає про це клієнта. Якщо оплата не здійснилася, система скасовує замовлення та сповіщає клієнта про збій оплати. Якщо замовлення виявляється недійсним, система скасовує замовлення та сповіщає клієнта про недійсне замовлення.

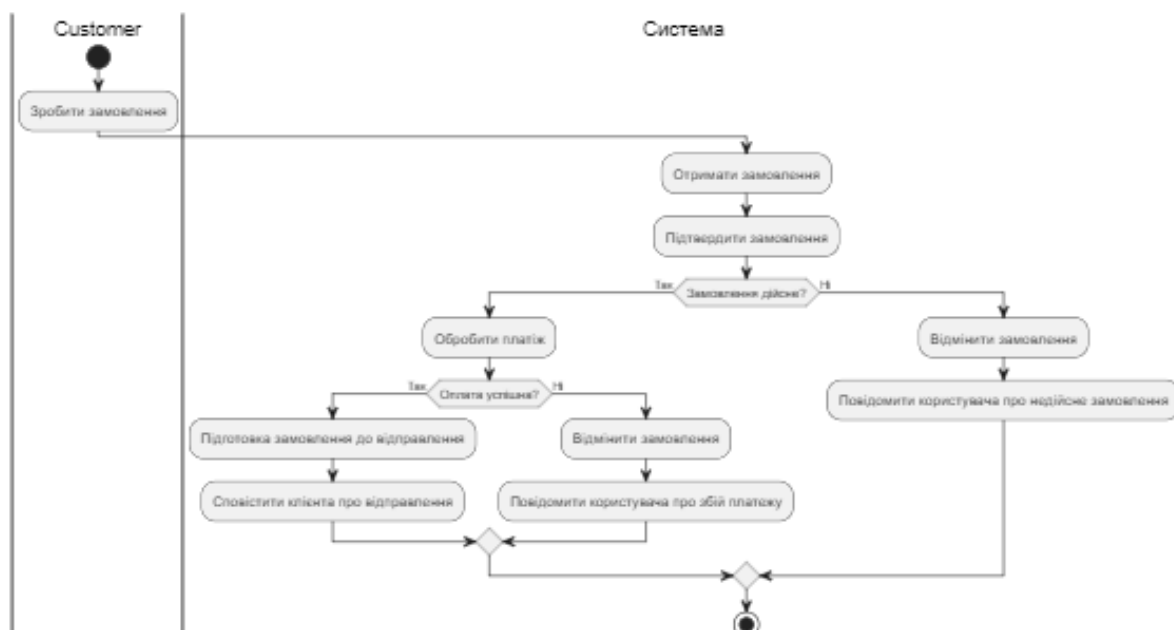


Рисунок 2.5 – Діаграма діяльності оформлення замовлення

Розробка інтерфейсу програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”

Розробка інтерфейсу має важливе значення для забезпечення ефективної взаємодії з користувачем, підвищення зручності використання, візуальної передачі інформації, надання зворотного зв'язку, а також забезпечення адаптивності на різних пристроях і платформах. Добре розроблений інтерфейс значною мірою сприяє загальному успіху та задоволенню користувачів програмним забезпеченням. Нижче наведено ескізи для кожної сторінки інтернет-магазину одягу для компанії “Brand New Outfit” (рисунки 2.6-2.11).

На рисунку 2.6 представлений ескіз сторінки “Реєстрація”:

Реєстрація

Введіть ваш email...

Введіть ваш пароль...

Є акаунт? [Увійти](#)

Зареєструватися

Рисунок 2.6 – Ескіз сторінки “Реєстрація”

На рисунку 2.7 представлений ескіз сторінки “Авторизація”:

Регістрація'. At the bottom right, there is a button labeled 'Увійти'." data-bbox="215 102 845 316"/>

Авторизація

Введіть ваш email...

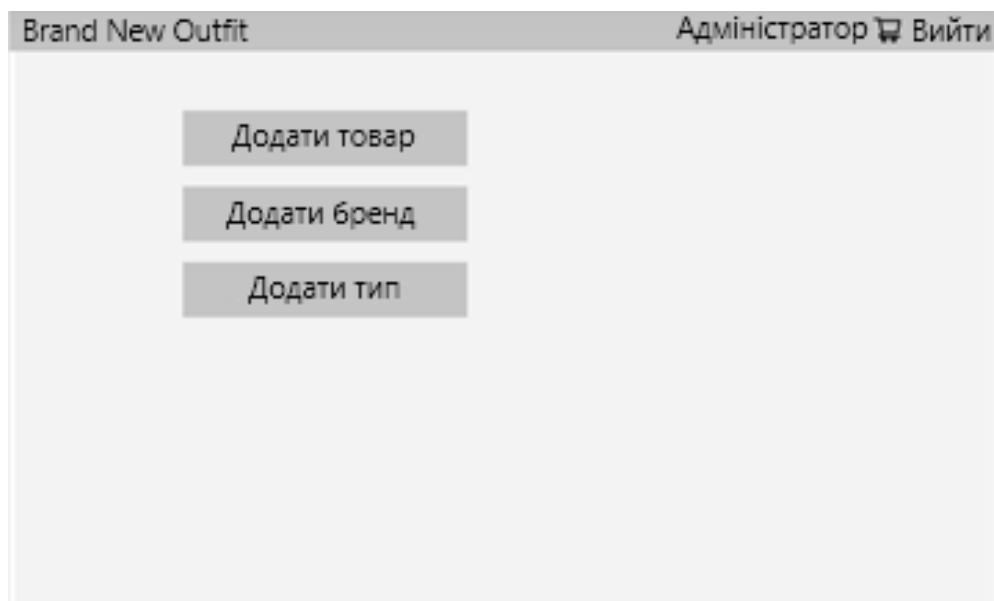
Введіть ваш пароль...

Немає акаунту? [Регістрація](#)

Увійти

Рисунок 2.7 – Ескіз сторінки “Авторизація”

На рисунку 2.8 представлений ескіз сторінки Адміністратора:



Brand New Outfit

Адміністратор  Вийти

Додати товар

Додати бренд

Додати тип

Рисунок 2.8 – Ескіз сторінки Адміністратора

На рисунку 2.9 представлений ескіз сторінки “Кошик”:

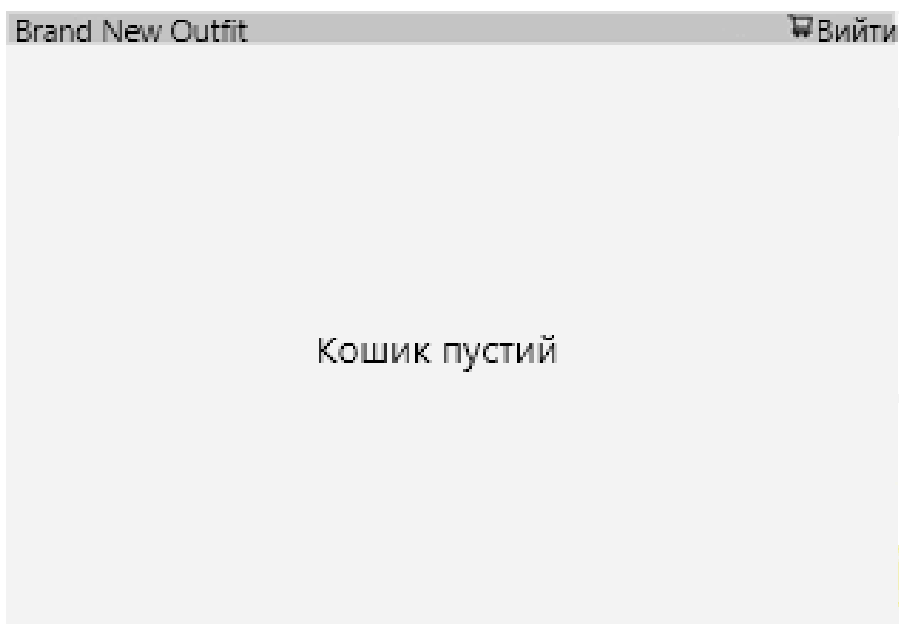


Рисунок 2.9 – Ескіз сторінки “Кошик”

На рисунку 2.10 представлений ескіз сторінки головної сторінки:



Рисунок 2.10 – Ескіз головної сторінки

На рисунку 2.6 представлений ескіз сторінки товару:

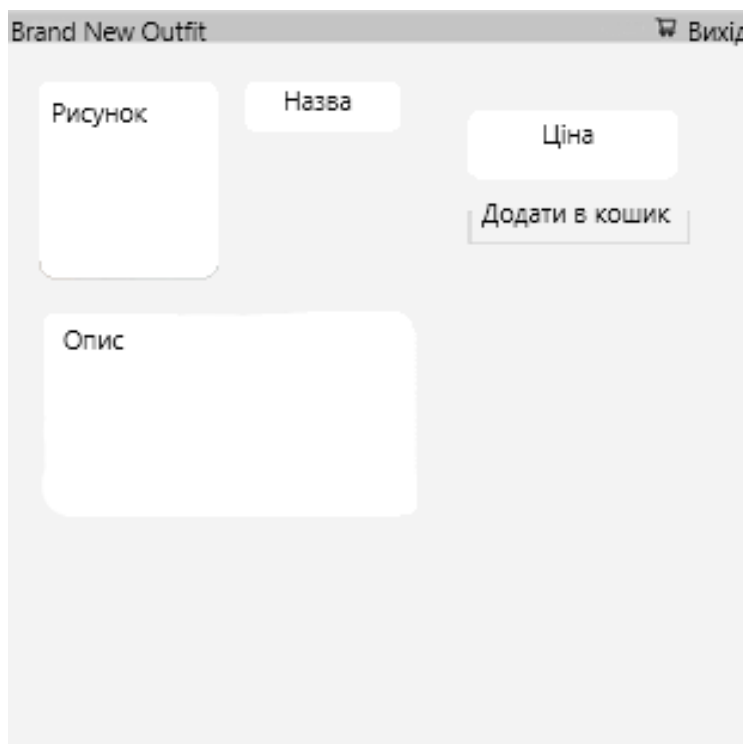


Рисунок 2.11 – Ескіз сторінки товару

2.3.2 Технічний проєкт програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”

Статична модель

Діаграма класів - це тип статичної моделі, яка використовується в інженерії програмного забезпечення для зображення структури та зв'язків між класами в системі. Вона надає огляд класів, їхніх атрибутів, методів і асоціацій між ними.

Наведена діаграма класів (рисунок 2.12) включає класи Користувач, Товар, Кошик, Тип, Бренд і Адміністратор, а зв'язки між класами представлені стрілками. Користувач керує Кошиком, а Кошик належить Користувачеві. Кошик містить об'єкти Товару, і кожен Товар належить до Типу та Бренду. Клас Адміністратор має асоціації з брендом, типом і товаром, що вказує на те, що він має можливість створювати екземпляри цих класів.

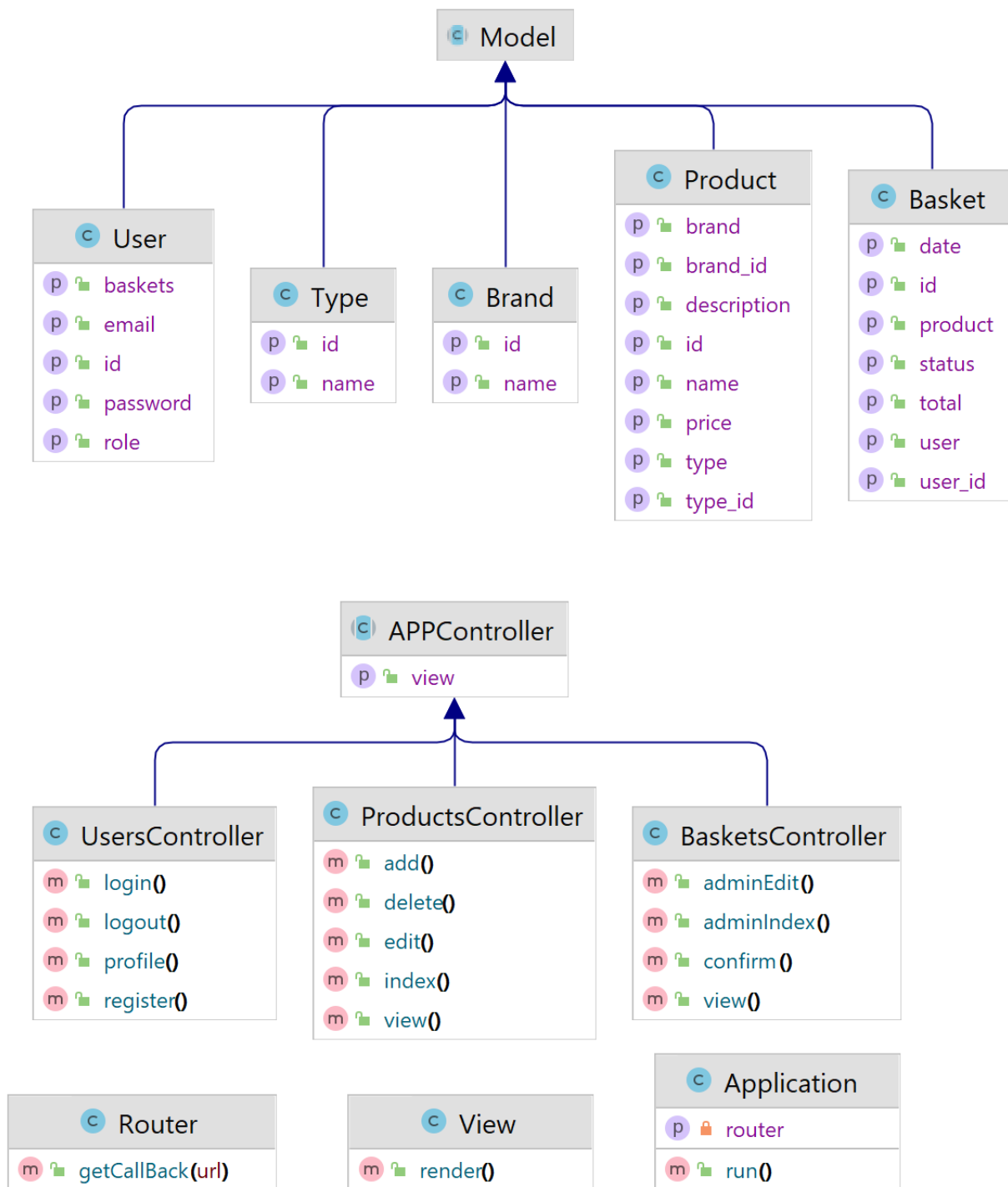


Рисунок 2.12 - Діаграма класів програмного забезпечення

Специфікації класів

Специфікації класів програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit” приведена нижче.

Клас: Model.

Відповідальність: служить абстрактним базовим класом для всіх моделей у програмі.

Клас: User.

Відповідальність: представляє користувача в інтернет-магазині.

Успадковується від: Model.

Атрибути:

- id: ідентифікатор користувача;
- email: електронна адреса користувача;
- password: пароль користувача;
- role: роль користувача;
- baskets: кошики, пов'язані з користувачами.

Клас: Type.

Відповідальність: представляє тип товару в інтернет-магазині.

Успадковується від: Model.

Атрибути:

- id: ідентифікатор типу;
- name: ім'я типу.

Клас: Brand.

Відповідальність: представляє бренд товару в інтернет-магазині.

Успадковується від: Model.

Атрибути:

- id: ідентифікатор бренду;
- name: назва бренду.

Клас: Product.

Відповідальність: представляє товар в інтернет-магазині.

Успадковується від: Model.

Атрибути:

- id: ідентифікатор товару;
- name: назва товару;
- description: опис товару;
- price: ціна товару;
- brand_id: ідентифікатор бренду, пов'язаного з товаром;
- brand: бренд, пов'язаний із товаром;
- type_id: ідентифікатор типу, пов'язаного з товаром;
- type: тип, пов'язаний із товаром.

Клас: Basket.

Відповідальність: представляє замовлення в інтернет-магазині.

Успадковується від: Model.

Атрибути:

- id: ідентифікатор замовлення;
- user_id: ідентифікатор користувача, пов'язаного з замовленням;
- user: користувач, пов'язаний із замовленням;
- date: дата замовлення;
- total: загальна ціна кошика;
- status: статус замовлення;
- product: товар, пов'язаний із замовленням.

Клас: APPController.

Відповідальність: служить абстрактним базовим класом для всіх контролерів у програмі.

Атрибути:

- view: представляє відображення, пов'язане з контролером.

Клас: UsersController.

Відповідальність: здійснює операції, пов'язані з користувачами.

Успадковується від: APPController.

Функції:

- register(): обробляє реєстрацію користувачів;
- login(): обробляє вхід користувача;
- logout(): обробляє вихід користувача з системи;
- profile(): відображає профіль користувача.

Клас: ProductsController.

Відповідальність: здійснює операції, пов'язані з товарами.

Успадковується від: APPController.

Функції:

- index(): відображає індекс товарів;
- view(): відображає вигляд товарів;
- add(): додає новий товар;
- edit(): редагує товар;
- delete(): видаляє товар.

Клас: BasketsController

Відповідальність: виконує операції з кошиками.

Успадковується від: APPController.

Функції:

- view(): відображає вигляд кошика;
- confirm(): підтвердження замовлення;
- adminIndex(): перегляд списку замовлень адміністратором;
- adminEdit(): зміна статусу замовлення адміністратором.

Клас: Router.

Відповідальність: обробляє запити маршрутизації.

Функції:

- getCallBack(url): отримує зворотний виклик, пов'язаний із вказаною URL-адресою.

Клас: View.

Відповідальність: представляє відображення в шаблоні MVC.

Функції:

- render(): рендерить відображення.

Клас: Application.

Відповідальність: представляє точку входу програми та керує виконанням програми.

Атрибути:

- router: представляє маршрутизатор, відповідальний за маршрутизацію запитів.

Функції:

- run(): запускає програму.

Динамічна модель

Діаграми послідовності використовуються для візуального представлення взаємодії та порядку повідомлень між різними об'єктами чи компонентами в системі. Вони надають динамічне уявлення про те, як об'єкти співпрацюють для виконання певної функції або сценарію використання. Діаграми послідовності допомагають зрозуміти потік керування, послідовність подій і взаємодію між об'єктами під час виконання певного сценарію. Їх можна використовувати для проектування, документування та передачі інформації про поведінку системи або для аналізу та усунення несправностей складних взаємодій.

Нижче представлена діаграма послідовностей для реєстрації та авторизації користувача (рисунки 2.13):

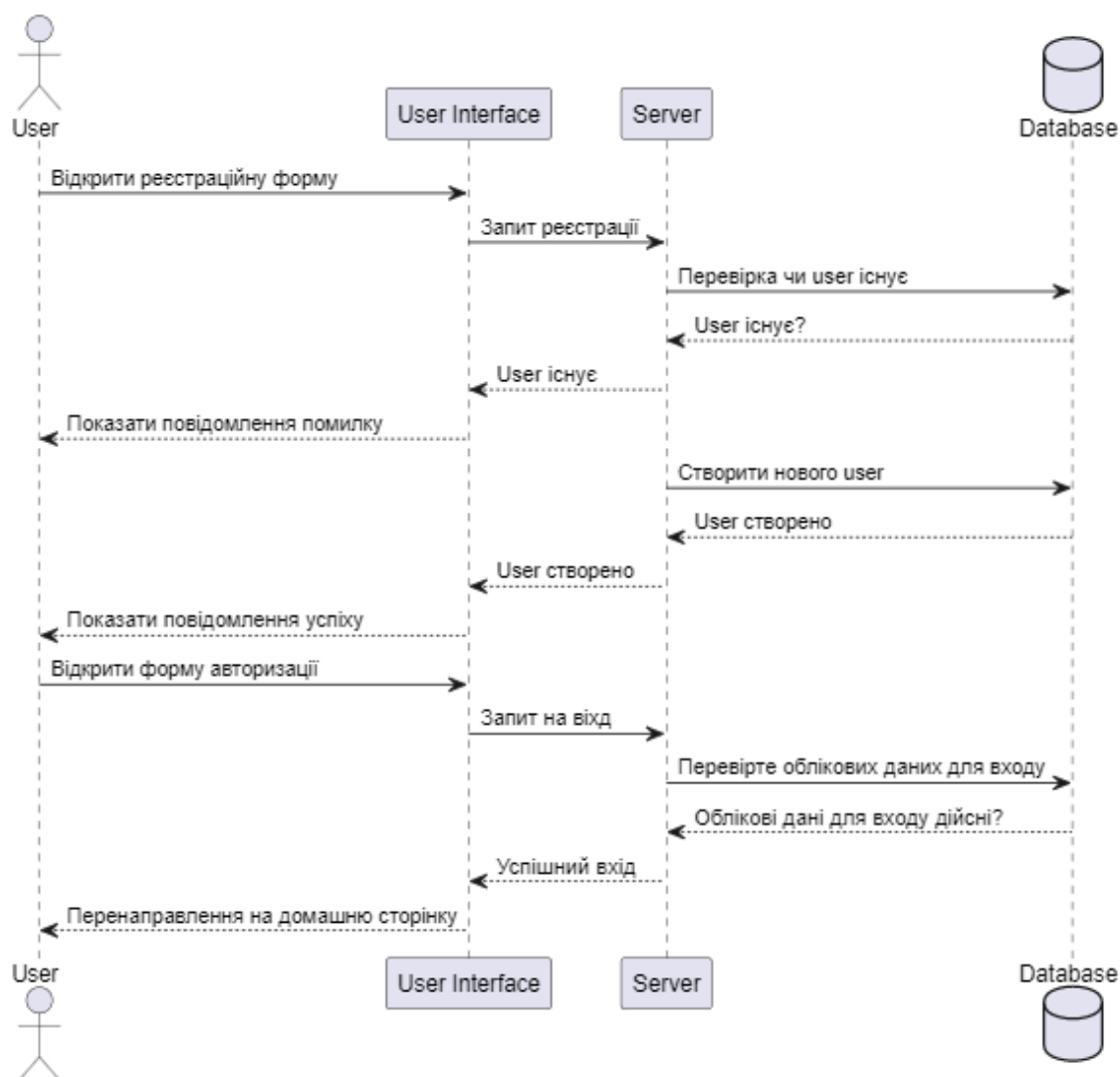


Рисунок 2.13 - Діаграма послідовності для реєстрації та авторизації користувача

Наступна діаграма послідовностей (рисунок 2.14) ілюструє взаємодію між користувачем, сторінкою магазину, сторінкою товару, магазином товарів, API товару, магазином кошика та API кошика. Вона показує потік дій, коли користувач переглядає товар, шукає товар, вибирає товар і додає його у кошик.

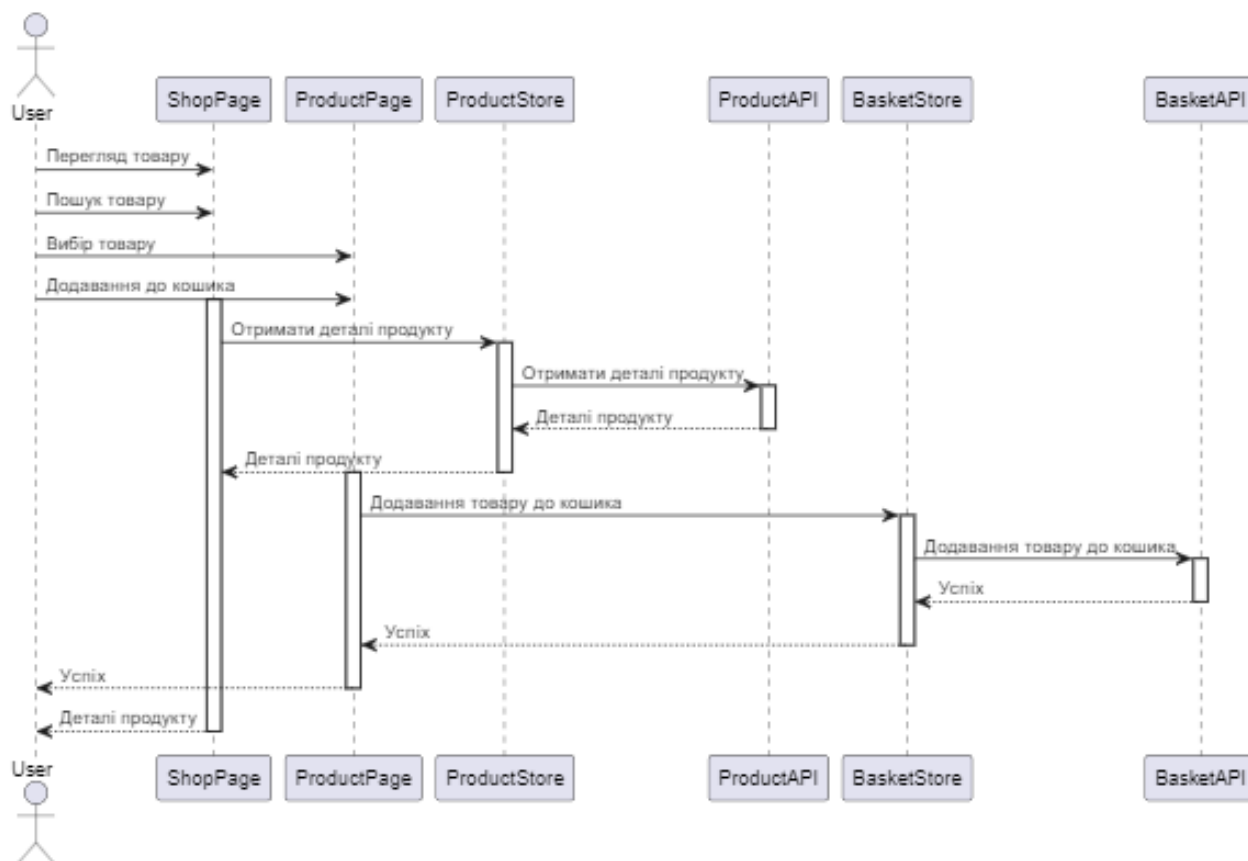


Рисунок 2.14 - Діаграма послідовності для перегляду, пошуку, вибору та додаванню товару до кошика

Для оформлення замовлення та оплати користувачем (рисунок 2.15):

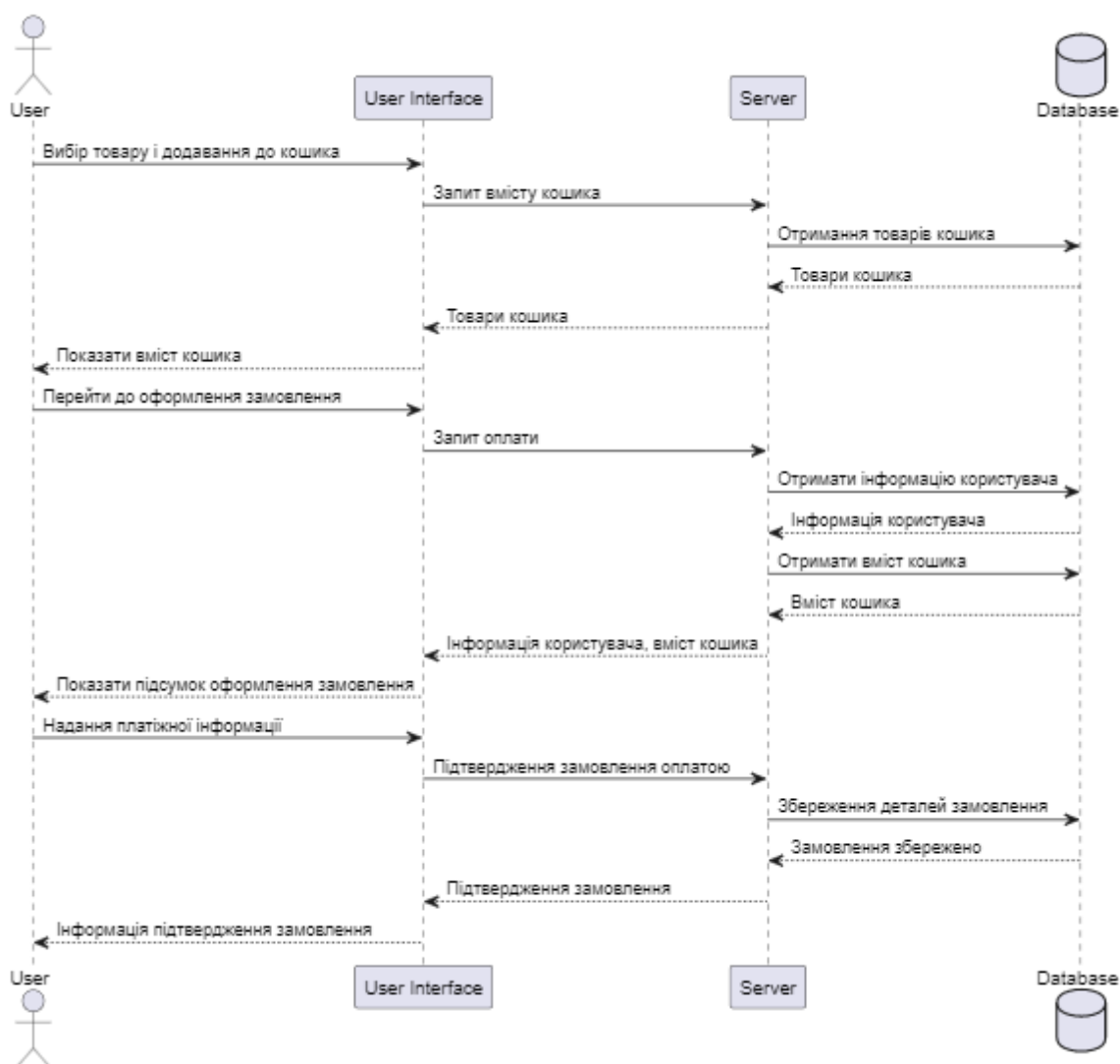


Рисунок 2.15 - Діаграма послідовності для оформлення замовлення та оплати

На даній діаграмі зображений процес керування товаром або брендом або типом. Оскільки послідовність дій однакова, я об'єднала їх в одну діаграму (рисунок 2.16):

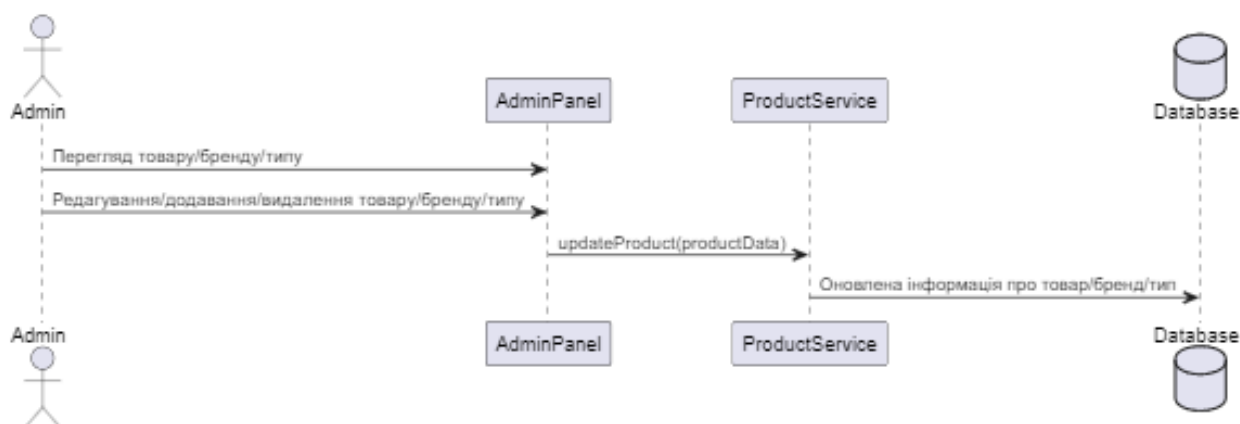


Рисунок 2.16 - Діаграма послідовності для керування товаром/брендом/типом

Логічна модель даних

Логічна модель даних представляє структуру та організацію даних у системі без зазначення деталей фізичної реалізації. Основна увага приділяється сутностям, атрибутам та відносинам. Нижче на рисунку 2.17 представлена логічна модель даних.

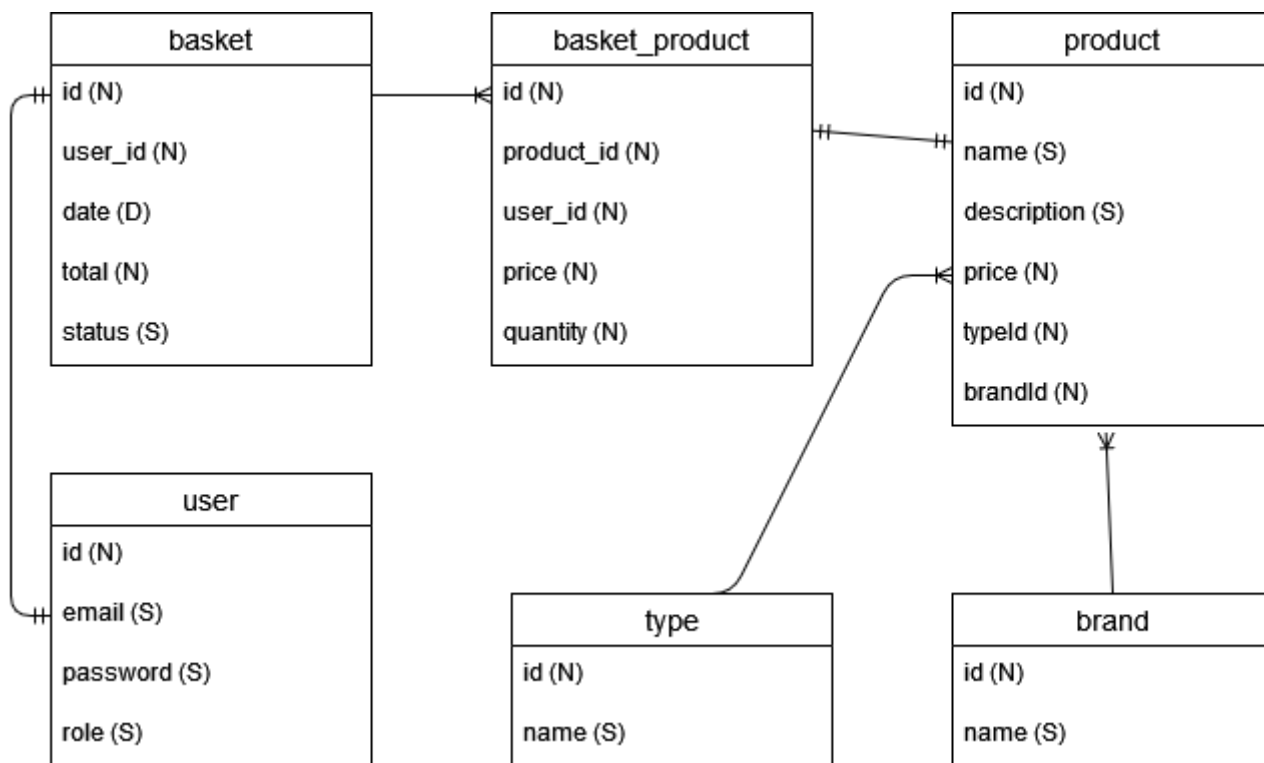


Рисунок 2.17 - Логічна модель даних

2.3.3 Робочий проєкт програмного забезпечення інтернет-магазину одягу для компанії “Brand New Outfit”

Обґрунтування вибору мови програмування та засобів розробки

Для розробки програмного забезпечення інтернет-магазину для компанії “Brand New Outfit” я обрала мову програмування JavaScript, а середовище розробки Visual Studio Code.

На сьогоднішній день JavaScript - єдина мова програмування для браузерів [6]. JavaScript є популярною мовою програмування та вибором для веброзробки, включно з онлайн-магазинами. Її можна використовувати як для зовнішньої розробки (на стороні клієнта), так і для серверної (на стороні сервера). Фреймворки JavaScript, такі як React, Angular і Vue.js, часто використовуються для створення

інтерактивних інтерфейсів користувача, тоді як Node.js дозволяє розробляти JavaScript на стороні сервера.

JavaScript (зокрема Node.js) часто вибирають для онлайн-магазинів через такі переваги:

- можна використовувати як для зовнішньої, так і для внутрішньої розробки, що дозволяє працювати з однією мовою в усьому стеку. Це зменшує перемикання контексту та полегшує підтримку та масштабування кодової бази.

- має одну з найбільших і найактивніших спільнот розробників, а це означає, що існує велика кількість ресурсів, бібліотек, фреймворків і підтримки спільноти для створення онлайн-магазинів.

- JavaScript, зокрема з Node.js, підтримує асинхронне програмування, що є корисним для обробки одночасних запитів і створення масштабованих і адаптивних онлайн-магазинів.

- фреймворки JavaScript, такі як React, дозволяють розробку на основі компонентів, сприяючи повторному використанню коду та модульній архітектурі. Це може оптимізувати розробку та покращити підтримку коду.

Visual Studio Code є популярним вибором для веб-розробки через кілька зручностей:

- має багату екосистему розширень, які підвищують продуктивність і забезпечують підтримку мови, інструменти налагодження та інтеграцію з різними фреймворками та технологіями веб-розробки.

- пропонує функції завершення коду, IntelliSense та навігації по коду, які допомагають розробникам писати код швидше та з меншою кількістю помилок.

- є інтегрований термінал, який дозволяє розробникам запускати команди та сценарії, не виходячи з редактора, забезпечуючи бездоганний досвід розробки.

- має вбудовану інтеграцію Git, що полегшує керування версіями та співпрацює з іншими користувачами в кодовій базі.

- відомий своєю продуктивністю та легким характером, що робить його придатним для завдань веб-розробки без споживання надмірних системних ресурсів.

Ці фактори сприяють популярності Visual Studio Code серед розробників, які працюють над онлайн-магазинами та веб-додатками загалом.

Обґрунтування вибору системи керування бази даних

PostgreSQL часто вважають кращим вибором для певних сценаріїв завдяки його розширеним функціям, сумісності з SQL, розширюваності, підтримці паралелізму, цілісності даних, активній спільноті з відкритим кодом і крос-платформній сумісності.

Однак найкращий вибір бази даних залежить від конкретних вимог та інших факторів. Наприклад, Visual Studio Code та PostgreSQL - комбінація для розробки та керування базами даних PostgreSQL. За допомогою цих інструментів можна писати код, керувати підключеннями до бази даних і ефективно взаємодіяти з базою даних PostgreSQL.

У Visual Studio Code можна встановити розширення, які забезпечують підтримку PostgreSQL, наприклад "PostgreSQL" від Microsoft. Це розширення дозволяє підключатися до бази даних PostgreSQL, виконувати запити та керувати об'єктами бази даних безпосередньо з редактора.

Фізична модель даних

Фізична модель даних представляє деталі конкретної реалізації системи баз даних. Визначає як логічна модель даних буде перетворена на фактичну схему бази даних, включаючи таблиці, стовпці, типи даних, індекси. Нижче представлена фізична модель даних (таблиці 2.15-2.20).

Таблиця 2.15 - Структура таблиці user

Назва	Тип даних	Ключ
id	integer	Primary Key
email	varchar	
password	varchar	
role	varchar	

Таблиця 2.16 - Структура таблиці product

Назва	Тип даних	Ключ
id	integer	Primary Key
name	varchar	
description	text	
price	decimal	
brand_id	integer	Foreign Key
type_id	integer	Foreign Key

Таблиця 2.17 - Структура таблиці brand

Назва	Тип даних	Ключ
id	integer	Primary Key
name	varchar	

Таблиця 2.18 - Структура таблиці type

Назва	Тип даних	Ключ
id	integer	Primary Key
name	varchar	

Таблиця 2.19 - Структура таблиці basket

Назва	Тип даних	Ключ
id	integer	Primary Key
user_id	integer	Foreign Key
date	timestamp	
total	decimal	
status	varchar	

Таблиця 2.20 - Структура таблиці basket_product

Назва	Тип даних	Ключ
id	integer	Primary Key
basket_id	integer	Foreign Key
product_id	integer	Foreign Key
price	decimal	
quantity	integer	

Тестування програмного забезпечення інтернет-магазину компанії “Brand New Outfit”

Мета тестування кожного варіанту використання інтернет-магазину компанії “Brand New Outfit” полягає у перевірці працездатності та справності роботи програмного забезпечення. Тестування приведено нижче.

1) Реєстрація:

Вхідні дані: дані користувача (адреса електронної пошти, пароль).

Тест: реєстрування нового користувача з наданими даними.

Результат: користувач успішно зареєстрований в системі.

1.1) Реєстрація – альтернативний потік.

Вхідні дані: дані користувача (адреса електронної пошти користувача, якого вже зареєстровано на сайті, пароль).

Результат: користувачу відображається повідомлення.

2) Авторизація:

Вхідні дані: облікові дані користувача (електронна адреса, пароль).

Тест: вхід за допомогою наданих облікових даних.

Результат: користувач успішно автентифікований і увійшов у систему.

2.1) Авторизація - – альтернативний потік.

Вхідні дані: облікові дані користувача (електронна адреса незареєстрованого користувача, пароль).

Результат: система відображає повідомлення про помилку, яке вказує на те, що облікові дані недійсні.

3) Управління обліковим записом:

Вхідні дані: дані облікового запису користувача (адреса електронної пошти, пароль).

Тест: оновлення даних облікового запису користувача.

Результат: дані облікового запису користувача успішно оновлено.

4) Вихід з облікового запису:

Тест: вихід з поточного сеансу користувача.

Результат: користувач успішно вийшов із системи.

5) Перегляд товарів:

Тест: перегляд списку доступних товарів.

Результат: користувачеві відображається список товарів.

6) Пошук товару:

Вхідні дані: пошуковий запит або фільтри (наприклад, назва товару, бренд, тип).

Тест: виконання пошуку товарів на основі наданого запиту або фільтрів.

Результат: відповідний товар відображається користувачеві.

7) Управління кошиком:

Тест: додавання товарів у кошик, видалення товарів із кошика, оновлення кількості товарів у кошику.

Результат: кошик оновлюється відповідно до доданих, видалених або оновлених товарів.

8) Оформлення замовлення:

Вхідні дані: товари в кошику, реквізити користувача, інформація про доставку, реквізити для оплати.

Тест: розміщення замовлення з наданою інформацією.

Результат: замовлення успішно розміщено та підтверджено.

9) Зміна статусу замовлення:

Тест: оновлення статусу замовлення.

Результат: інформація про замовлення відображається та оновлюється правильно.

10) Додавання товару:

Вхідні дані: назва, опис, ціна, тип, бренд товару.

Тест: додавання нового товару.

Результат: товар додається коректно.

11) Редагування товару:

Вхідні дані: назва, опис, ціна, тип, бренд, ідентифікатор товару.

Тест: редагування наявного товару.

Результат: інформація про товар відображається та оновлюється правильно.

12) Видалення товару:

Вхідні ідентифікатор товару.

Тест: видалення товарів.

Результат: товар видалено.

3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕРНЕТ-МАГАЗИНУ ОДЯГУ ДЛЯ КОМПАНІЇ “BRAND NEW OUTFIT”

В кваліфікаційній роботі було розроблене програмне забезпечення інтернет-магазину одягу компанії “Brand New Outfit”. Реалізовувався проект використовуючи стек PERN (PostgreSQL, Express.js, React.js і Node.js.). Програмне забезпечення складається із 3176 рядків.

Технічне завдання наведено у Додатку А.

Було проведено випробування інтернет-магазину згідно з програмою та методикою випробувань, наведеною у Додатку Б. Всі випробування було успішно пройдено, що свідчить про працездатність програмного забезпечення інтернет-магазину компанії “Brand New Outfit”.

У Додатку В представлено керівництво користувача для програмного забезпечення інтернет-магазину одягу компанії “Brand New Outfit”.

Опис програмного забезпечення представлено у Додатку Г.

Частина тексту реалізованої програми представлена у Додатку Д.

4 ОХОРОНА ПРАЦІ

Охорона праці. Визначення поняття. Основні завдання. Правове регулювання охорони праці в Україні

Визначення поняття охорони праці дається в ст. 1 Закону України від 14 жовтня 1992 р. «Про охорону праці». Охорона праці — це система правових, соціально-економічних, організаційно-технічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі праці. В поняття охорони праці входять і всі ті заходи, що спеціально призначені для створення особливих полегшених умов праці для жінок і неповнолітніх, а також працівників зі зниженою працездатністю.

Охорону праці і здоров'я громадян віднесено до пріоритетних напрямків соціальної політики України. Так, Конституція України одним з основних соціальних прав громадян визначає право кожного на належні, безпечні й здорові умови праці, встановлює, що використання праці жінок і неповнолітніх на небезпечних для їхнього здоров'я роботах забороняється.

Основні законодавчі акти про охорону праці

Конституція України, Закони України “Про охорону праці”, “Про охорону здоров'я”, “Про пожежну безпеку”, “Про використання ядерної енергії та радіаційний захист”, “Про забезпечення санітарного та епідемічного благополуччя населення”, “Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності”, Кодекс законів про працю України.

В Основному Законі України – Конституції питанням охорони праці присвячено статті 43, 45 та 46.

У ст. 43 Конституції України записано: “Кожен має право на працю, що включає можливість заробляти собі на життя працею, яку він вільно обирає або на яку вільно погоджується”, “Кожен має право на належні, безпечні і здорові умови

праці, на заробітну плату, не нижчу від визначеної законом”; “Використання праці жінок і неповнолітніх на небезпечних для їхнього здоров’я роботах забороняється”.

“Кожен, хто працює, має право на відпочинок” (ст. 45 Конституції України). Це право забезпечується наданням днів щотижневого відпочинку, а також оплачуваної щорічної відпустки, встановленням скороченого робочого дня щодо окремих професій і виробництв, скороченої тривалості роботи у нічний час.

У тексті ст. 46 Конституції України наголошено на тому, що “громадяни мають право на соціальний захист, що включає право на забезпечення їх у разі повної, часткової або тимчасової втрати працездатності, втрати годувальника, безробіття з незалежних від них обставин, а також у старості та в інших випадках, передбачених законом”.

Основним законодавчим документом у галузі охорони праці є Закон України “Про охорону праці”, дія якого поширюється на юридичних та фізичних осіб, які відповідно до законодавства використовують найману працю, та на всіх працюючих.

Завдання охорони праці:

- проектування підприємств, технологічних процесів і конструювання обладнання з обов’язковим виконанням вимог охорони праці;
- знаходження оптимальних співвідношень між різними факторами виробничого середовища, що дозволяє забезпечити мінімум несприятливого впливу їх на здоров’я працівників;
- встановлення, законодавче оформлення визначених норм кожного з несприятливих або небезпечних факторів, систематичний контроль за їх застосуванням;
- розробка конкретних заходів щодо покращення умов праці та забезпечення її безпеки на основі застосування у виробництві новітніх досягнень науки і техніки;
- застосування раціональних засобів захисту працівників від впливу несприятливих факторів виробничого середовища, а також втілення організаційних заходів, які нейтралізують або послаблюють ступінь їх впливу на організм людини;

– розробка та застосування методів і засобів оцінки ефективності заходів з охорони праці, що плануються і здійснюються [3].

При роботі на персональному комп'ютері

1. Загальні положення

1.1. Дія інструкції поширюється на всі підрозділи підприємства, де виконують роботи з персональним комп'ютером (ПК) .

1.2. Інструкція розроблена відповідно до Положення про розробку інструкцій з охорони праці, затвердженого наказом Держнаглядохоронпраці від 29.01.1998 № 9, Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці від 26.01.2005 № 15, Правил охорони праці під час експлуатації електронно-обчислювальних машин, затверджених наказом Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду від 26.03.2010 № 65, Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98, затверджених постановою Головного державного санітарного лікаря України від 10.12.1998 № 7, Загальних вимог стосовно забезпечення роботодавцями охорони праці працівників, затверджених наказом Міністерства надзвичайних ситуацій України від 25.01.2012 № 67 (НПАОП 0.00-7.11-12).

1.3. За цією інструкцією працівника, який використовує персональний комп'ютер (далі — користувач), інструктують перед початком роботи (первинний інструктаж), а потім через кожні 6 місяців (повторний інструктаж). Результати інструктажу заносять до Журналу реєстрації інструктажів з питань охорони праці на робочому місці (у журналі має бути підпис особи, яка інструктує, та користувача).

1.4. Користувач зобов'язаний дбати про особисту безпеку і здоров'я, а також про безпеку і здоров'я довколишніх при виконанні будь-яких робіт, а також під час перебування на території підприємства.

1.5. До роботи на персональному комп'ютері допускають осіб, які пройшли інструктажі з питань охорони праці та пожежної безпеки.

1.6. Користувач зобов'язаний: виконувати правила внутрішнього трудового розпорядку; не допускати за своє робоче місце сторонніх осіб; не виконувати вказівок, які суперечать правилам охорони праці та пожежної безпеки; знати правила надання домедичної допомоги; знати розташування та вміти користуватись первинними засобами пожежогасіння; вміти працювати з ПК.

1.7. Основні небезпечні та шкідливі виробничі фактори, що можуть впливати на користувача: підвищений рівень статичної електрики; нерівномірність розподілу яскравості в полі зору; підвищена яскравість світлового зображення; ураження електричним струмом; напруга зору та уваги; тривалі статичні навантаження.

1.8. У приміщеннях із ПК має бути природне і штучне освітлення.

1.9. При розміщенні робочих місць необхідно унеможливити пряме засвічування екрана природним освітленням.

1.10. При природному освітленні слід передбачити наявність сонцезахисних засобів (плівка, жалюзі, штори тощо).

1.11. Світлові відблиски із клавіатури, екрана та інших частин ПК у напрямку очей користувача неприпустимі.

1.12. Основним обладнанням робочого місця є ПК або ноутбук, монітор, клавіатура, маніпулятор, робочий стіл, стілець (крісло).

1.13. При розміщенні елементів робочого місця слід враховувати: робочу позу користувача; простір для розміщення користувача; можливість огляду елементів робочого місця; можливість огляду простору поза межами робочого місця; можливість робити записи, розміщувати на робочому столі документацію та матеріали, які використовує користувач.

1.14. Розміщення елементів робочого місця не має заважати рухам та переміщенню для експлуатування ПК.

1.15. Монітор встановлюють так, щоб відстань від поверхні екрана до очей користувача була 600-700 мм залежно від розміру екрана.

1.16. Клавіатуру розміщують на робочому або окремому столі на відстані 100-300 мм від краю з боку користувача. Положення клавіатури та кут її нахилу

залежить від побажання користувача (як правило, в межах 5-15°). Не допускати хитання клавіатури.

1.17. Конструкція робочого столу має бути такою, щоб оптимально розмістити на робочій поверхні обладнання, що використовують, з урахуванням кількості, розмірів, конструктивних особливостей і характеру його роботи.

1.18. Крісло має забезпечувати підтримування раціональної робочої пози під час виконання основних виробничих операцій та можливість зміни пози. Тип робочого крісла обирають залежно від характеру та тривалості роботи.

1.19. Раціональна поза користувача: ступні розташовані на підлозі або на підставці для ніг; стегна зорієнтовані у горизонтальній площині; верхні ділянки рук вертикальні; кут ліктьового суглоба у межах 70-90°; зап'ястя зігнуті під кутом не більше ніж 20°; нахил голови у межах 15-20°, а часті її повороти виключені.

1.20. Для забезпечення оптимальної робочої пози користувача необхідно: засоби праці, з якими користувач має тривалий або найбільш частий зоровий контакт, розмістити у центрі зони зорового спостереження та моніторного поля; забезпечити відстань близько 500 мм між найважливішими засобами праці, з якими користувач працює найчастіше.

1.21. ПК встановлювати на рівній твердій поверхні (столі). Не дозволено встановлювати ПК та оргтехніку на хитких підставках чи на похилій поверхні.

1.22. ПК не встановлювати впритул до стіни, перегородки тощо. Не допускати загородження вентиляційних отворів ПК сторонніми предметами.

1.23. Розетка біля ПК має бути в доступному місці, щоб в аварійних випадках можна було своєчасно його відімкнути. Не рекомендовано використовувати подовжувачі.

1.24. Під час переміщення ПК, периферійних пристроїв витягти вилку живлення з розетки.

1.25. Не допускати ушкодження чи модифікування шнура живлення. Заборонено ставити важкі речі на шнур живлення, тягнути чи надмірно перегинати його, скручувати та перев'язувати шнур живлення вузлом.

1.26. ПК під'єднувати до електромережі лише за допомогою справних штепсельних з'єднань та електророзеток заводського виробництва.

1.27. Штепсельні з'єднання та електророзетки мають бути зі спеціальними контактами для під'єднання нульового захисного провідника. Їхня конструкція має забезпечувати з'єднання нульового захисного провідника раніше, ніж з'єднання фазового та нульового робочого провідників. Порядок роз'єднань при вимкненні має бути зворотнім.

1.28. Заборонено під'єднувати електрообладнання до звичайної двошнурової електромережі.

1.29. За невиконання цієї інструкції працівники несуть відповідальність згідно з чинним законодавством [4].

Загальні вимоги до умов праці на підприємствах встановлено законодавством про працю. Відповідно до ч. 1 ст. 6 Закону України «Про охорону праці» від 14.10.92 р. № 2694-XII (далі – Закон про охорону праці) умови праці на робочому місці, безпека технологічних процесів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам законодавства.

Більшість нормативів щодо умов праці офісних працівників встановлено на рівні державних стандартів. Основними з них є:

- Державні санітарні норми виробничого шуму, ультразвуку та інфразвуку ДСН 2.3.6.037-99, затверджені постановою Головного державного санітарного лікаря України від 01.12.99 р. № 37;

- Державні санітарні норми виробничої загальної та локальної вібрації ДСН 3.3.6.039-99, затверджені постановою Головного державного санітарного лікаря України від 01.12.99 р. № 39;

- Державні санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99, затверджені постановою Головного державного санітарного лікаря України від 01.12.99 р. № 42;

- Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98, затверджені постановою Головного державного санітарного лікаря України від 10.12.98 р. № 7;

Правила охорони праці під час експлуатації електронно-обчислювальних машин, затверджені наказом Держгірпромнагляду від 26.03.2010 р. № 65 (далі – Правила № 65);

Загальні вимоги стосовно забезпечення роботодавцями охорони праці працівників, затверджені наказом МНС від 25.01.2012 р. № 67.

Відповідно до ч. 1 ст. 13 Закону про охорону праці роботодавець зобов'язаний створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів.

Площу приміщень, в яких розташовують персональні комп'ютери, визначають згідно з чинними нормативними документами. Відповідно до ДСанПіН 3.3.2.007-98 з розрахунку на одне робоче місце, обладнане ПК, встановлено такі норми:

- площа – не менше 6,0 кв. м;
- об'єм – не менше 20,0 куб. м.

Заземлені конструкції, що знаходяться у приміщеннях (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном тощо), мають бути надійно захищені діелектричними щитками або сітками від випадкового дотику.

Також в цих приміщеннях повинні бути медичні аптечки першої допомоги, системи автоматичної пожежної сигналізації та переносні вуглекислотні вогнегасники. Підходи до засобів пожежогасіння повинні бути вільними.

Робочі місця, згідно з п. 4.3 ДСанПіН 3.3.2.007-98, слід розташовувати відносно світлових прорізів так, щоб природне світло падало переважно з лівого боку.

Конструкція робочого місця користувача персонального комп'ютера має забезпечити підтримання оптимальної робочої пози офісного працівника. Конструкція робочого столу має відповідати сучасним вимогам ергономіки і

забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів.

Правилами встановлюються висота робочої поверхні робочого столу, параметри ширини і глибини для робочих столів, які мають забезпечувати можливість виконання операцій у зоні досяжності моторного поля.

Відповідно до п. 4.8 ДСанПіН 3.3.2.007-98 робочий стілець має бути підйомно-поворотним, регульованим за висотою, з кутом нахилу сидіння та спинки, від спинки до переднього краю сидіння поверхня сидіння має бути плоскою, передній край – заокругленим. Регулювання за кожним із параметрів має здійснюватися незалежно, легко і надійно фіксуватися.

Поверхня сидіння і спинки стільця має бути напівм'якою з нековзним, повітронепроникним покриттям, що легко чиститься і не електризується (п. 4.12 ДСанПіН 3.3.2.007-98).

Робочий стіл для ПК, як правило, має бути обладнаним підставкою для ніг, вимоги до її розмірів та конструкції також прописані в правилах. Застосування підставки для ніг тими, у кого ноги не дістають до підлоги, є обов'язковим.

Приміщення можуть обладнуватись шафами для зберігання документів, магнітних дисків, полицями, стелажми, тумбами тощо з урахуванням вимог до площі приміщень.

Поверхня підлоги має бути рівною, неслизькою, з антистатичними властивостями. Забороняється для оздоблення інтер'єру приміщень з персональними комп'ютерами застосовувати полімерні матеріали (деревинно-стружкові плити, шпалери, що миються, рулонні синтетичні матеріали, шаруватий паперовий пластик тощо), що виділяють у повітря шкідливі хімічні речовини.

Відповідно до розд. VI Правил № 65 щодня перед початком роботи необхідно очищати монітор від пилу та інших забруднень. Після закінчення роботи персональний комп'ютер і периферійні пристрої повинні бути відключені від електричної мережі. У разі виникнення аварійної ситуації необхідно негайно відключити персональний комп'ютер і периферійні пристрої від електричної мережі.

Персональні комп'ютери, периферійні пристрої повинні підключатися до електромережі тільки з допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення (п. 2.9 Правил № 65). Штепсельні з'єднання та електророзетки, окрім контактів фазового та нульового робочого провідників, повинні мати спеціальні контакти для підключення нульового захисного провідника. Конструкція їх має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Необхідно унеможливити з'єднання контактів фазових провідників з контактами нульового захисного провідника. Неприпустимим є підключення комп'ютерів, периферійних пристроїв до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв.

Є неприпустимими:

- експлуатація кабелів та проводів з пошкодженою або такою, що втратила захисні властивості за час експлуатації, ізоляцією;
- застосування саморобних подовжувачів, застосування для опалення приміщення нестандартного (саморобного) електронагрівального обладнання або ламп розжарювання;
- користування пошкодженими розетками, розгалужувальними та з'єднувальними коробками, вимикачами та іншими електровиробами, а також лампами, скло яких має сліди затемнення або випинання;
- використання електроапаратури та приладів в умовах, що не відповідають вказівкам (рекомендаціям) підприємств-виготовлювачів.

Приміщення для роботи з персональними комп'ютерами мають бути обладнані системами опалення, кондиціонування повітря, або припливно-втяжною вентиляцією. У приміщеннях на робочих місцях мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості та рухливості повітря відповідно до норм та правил, а також ДБН В.2.5-67:2013 «Опалення, вентиляція та кондиціонування», затверджених наказом Мінрегіону від 25.01.2013 р. № 24.

Відповідно до санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99 в офісних приміщеннях температура повітря повинна становити 22-25°C, відносна вологість повітря – 40-60 %, швидкість руху повітря – не більше 0,1 м/с.

Під час перевищення припустимих значень робочий день співробітників повинен бути скорочений мінімум на 10 %.

Для підтримки допустимих значень мікроклімату та концентрації позитивних і негативних іонів необхідно передбачати установки або прилади зволоження та/або штучної іонізації, кондиціонування повітря. В Україні відсутні затверджені на законодавчому рівні гранично допустимі норми вмісту вуглекислого газу в повітрі для житлових, офісних та громадських споруд. Проте, враховуючи його вплив на працівників, а саме суттєве зниження їх працездатності, роботодавцям варто приділяти цьому питанню увагу та вживати заходи профілактики.

Як вже зазначалося, відносно вікон робоче місце необхідно організовувати так, щоб природне світло було з лівого боку (п. 4.3 ДСанПіН 3.3.2.007-98). Робоче місце необхідно розміщувати таким чином, щоб уникнути попадання прямого світла в очі. Для забезпечення захисту і досягнення нормованих рівнів комп'ютерних випромінювань необхідне застосування приєкраних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, що пройшли випробування в акредитованих лабораторіях і мають щорічний гігієнічний сертифікат (п. 4.19 ДСанПіН 3.3.2.007-98).

Штучне освітлення приміщення має здійснюватись системою загального рівномірного освітлення (п. 3.2.2 ДСанПіН 3.3.2.007-98). У приміщеннях при переважній роботі з документами допускається використання системи комбінованого освітлення, тобто встановлення світильників місцевого освітлення додатково до загального.

Як джерела штучного освітлення необхідно використовувати люмінесцентні лампи. Згідно з п. 3.2.5 ДСанПіН 3.3.2.007-98 система загального освітлення має

бути у вигляді суцільних або переривчатих ліній світильників, що розташовані збоку від робочих місць (зазвичай ліворуч) паралельно лінії зору працівників.

Допускається застосування ламп розжарювання у світильниках місцевого освітлення та, у разі влаштування відбитого освітлення у виробничих чи адміністративно-громадських приміщеннях, металогалогенних ламп потужністю 250 Вт.

Коефіцієнт пульсації не повинен перевищувати 5 % (п. 3.2.14 ДСанПіН 3.3.2.007-98). Рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300 – 500 лк. Світильники місцевого освітлення слід встановлювати таким чином, щоб не створювати відблисків на поверхні екрана, а освітленість екрана має не перевищувати 300 лк.

Для забезпечення нормованих значень освітленості у приміщеннях відповідно до п. 3.2.15 ДСанПіН 3.3.2.007-98 необхідно мити вікна і світильники не рідше 2 разів на рік, а також своєчасно замінювати лампи, що перегоріли.

Рівні шуму та вібрації на робочих місцях осіб, що працюють з ПК, визначаються відповідно до ДСанПіН 3.3.2.007-98.

Для забезпечення дотримання допустимих рівнів шуму на робочих місцях застосовуються засоби звукопоглинання, вибір яких обґрунтовується спеціальними інженерно-акустичними розрахунками (п. 3.3.3 ДСанПіН 3.3.2.007-98).

Перелік організаційно-технічних заходів щодо обмеження несприятливого впливу шуму та вібрації на працюючих наведено в ДСН 2.3.6.037-99 та ДСН 3.3.6.039-99, серед яких зменшення шуму та вібрації на шляху розповсюдження засобами ізоляції та поглинання, наприклад, за рахунок використання гумових, поролонових, інших шумо- чи вібропоглинаючих матеріалів, або інших матеріалів аналогічного призначення, що дозволені для оздоблення приміщень органами державного санітарно-епідеміологічного нагляду.

Допустимі параметри неіонізуючого електромагнітного випромінювання

Вимоги щодо рівня неіонізуючих електромагнітних випромінювань, електростатичних та магнітних полів встановлюються відповідно до ДСанПіН 3.3.2.007-98, а також Вимог до роботодавців щодо захисту працівників від

шкідливого впливу електромагнітних полів, затверджених наказом Міненергетики від 05.02.2014 р. № 99, ДСанПіН 3.3.6.096-2002.

Значення напруженості електростатичного поля на робочих місцях (як у зоні екрана дисплея, так і на поверхнях обладнання, клавіатури, друкувального пристрою) мають не перевищувати гранично допустимих відповідно до встановлених норм [5].

ВИСНОВКИ

В кваліфікаційній роботі було досягнуто поставленої мети: розроблене програмне забезпечення інтернет-магазину одягу компанії “Brand New Outfit”.

Для цього було виконано всі поставлені завдання:

1. Проаналізовано практичну задачу розробки програмного забезпечення інтернет-магазину одягу.

2. Проаналізовано існуюче програмне забезпечення інтернет-магазинів.

3. Виконано постановку задачі на розробку програмного забезпечення інтернет-магазину компанії “Brand New Outfit”. Технічне завдання наведено у Додатку А.

4. Розроблено ескізний проєкт програмного забезпечення інтернет-магазину компанії “Brand New Outfit”.

5. Розроблено технічний проєкт програмного забезпечення інтернет-магазину компанії “Brand New Outfit”.

6. Розроблено робочий проєкт програмного забезпечення інтернет-магазину компанії “Brand New Outfit”.

7. Реалізовано, протестовано та випробувано програмне забезпечення інтернет-магазину компанії “Brand New Outfit”. Програму та методику випробувань наведено у Додатку Б.

8. Розглянуто питання з охорони праці в роботі інтернет-магазину компанії “Brand New Outfit”.

У Додатку В представлено керівництво користувача для програмного забезпечення інтернет-магазину одягу компанії “Brand New Outfit”.

Опис програмного забезпечення представлено у Додатку Г.

Частина тексту реалізованої програми представлена у Додатку Д.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке вебсайт. URL: <https://futurenow.com.ua/shho-take-veb-sajt-yaka-istoriya-yih-vynyknennya-ta-vydy-veb-sajtiv-shho-take-veb-storinka/> (дата звернення: 05.03.2023).
2. Що таке MVC. URL: <https://hi-news.pp.ua/kompyuteri/14628-programa-model-view-controller-mvc-scho-ce-osoblivost-opis.html> (дата звернення: 08.03.2023).
3. Охорона праці. Визначення поняття. Основні завдання. Правове регулювання охорони праці в Україні. URL: https://studies.in.ua/bjd_seminar/1044-ohorona-prac.html (дата звернення: 09.05.2023).
4. При роботі на персональному комп'ютері. URL: <https://services.uteka.ua/ua/publication/zrazky-34-trudovi-vidnosyny-ta-oplata-pratsi-138-instrukciya-po-oxrane-truda-pri-rabote-na-personalnom-kompyutere-obrazec> (дата звернення: 09.05.2023).
5. Організація та обладнання робочого місця. URL: <https://oppb.com.ua/news/ohorona-praci-ofisnyh-pracivnykiv-organizaciya-ta-obladnannya-robochogo-miscya> (дата звернення: 09.05.2023).
6. Що таке JavaScript. URL: <https://wezom.academy/ua/chto-takoe-javascript-i-zachem-ego-uchit/> (дата звернення: 14.02.2023).

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1 Вступ

1.1 Назва програмного забезпечення

Інтернет-магазину одягу компанії “Brand New Outfit”.

1.2 Сфера застосування

Онлайн торгівля одягом.

2 Підстава для розробки програмного забезпечення

Розробка програмного забезпечення виконується на підставі завдання на кваліфікаційну роботу, виданого кафедрою ПЗАС НУК імені адмірала Макарова.

3 Призначення розробки

3.1 Функціональне призначення

Функціональним призначенням програмного забезпечення є створення платформи для перегляду та купівлі одягу.

3.2 Експлуатаційне призначення

Надає інформацію про товари - зберігання, пошук та проведення різних операцій з даними, що зберігаються.

4 Технічні вимоги до програми

4.1 Вимоги до функціональних характеристик

Програма повинна забезпечувати можливість виконання нижче перерахованих функцій:

- реєстрація;

- авторизація;
- керування обліковим записом;
- вихід з облікового запису;
- перегляд товару;
- пошук товару;
- керування кошиком;
- оформлення замовлення;
- керування замовлення;
- керування товарами;
- керування брендами;
- керування типами.

Вимоги до організації вхідних та вихідних даних:

Реєстрація:

Вхідні дані: особиста інформація користувача (ім'я, адреса електронної пошти, пароль).

Вихідні дані: користувачу надіслано електронний лист із підтвердженням повідомлення про успішну реєстрацію.

Авторизація:

Вхідні дані: облікові дані користувача (електронна пошта, пароль).

Вихідні дані: повідомлення про успішний вхід.

Керування обліковим записом:

Вхідні дані: інформація про обліковий запис користувача (ім'я, адреса електронної пошти, пароль), оновлені дані облікового запису.

Вихідні дані: повідомлення про успішне оновлення інформації облікового запису, повідомлення про успішну зміну пароля, повідомлення про успішне видалення облікового запису.

Вихід з облікового запису:

Вхідні дані: ніяких конкретних даних не потрібно.

Вихідні дані: повідомлення про вихід користувача, перенаправлення на сторінку входу.

Перегляд товару:

Вхідні дані: ідентифікатор продукту або деталі продукту.

Вихідні дані: деталі товару (назва, опис, ціна тощо), зображення товару.

Пошук товару:

Вхідні дані: пошуковий запит (ключові слова, фільтри).

Вихідні дані: список відповідних товарів, деталі товару.

Керування кошиком:

Вхідні дані: ідентифікатор продукту або деталі продукту для додавання/вилучення з кошика.

Вихідні дані: поточний вміст кошика (перелік товарів, кількість, загальна ціна).

Оформлення замовлення:

Вхідні дані: вміст кошика, платіжні реквізити.

Вихідні дані: деталі підтвердження замовлення (ідентифікатор замовлення, загальна вартість)

Управління замовленнями:

Вхідні дані: ID замовлення або деталі замовлення.

Вихідні дані: деталі замовлення (статус замовлення, елементи, кількість, ціни), історія замовлень користувача.

Керування товаром:

Вхідні дані: деталі товару (назва, опис, ціна), інформація про бренд (назва, опис тощо), інформація про тип.

Вихідні дані: повідомлення про успішне створення/редагування/видалення товару, оновлений список товарів.

Керування брендом:

Вхідні дані: інформація про бренд (назва, опис).

Вихідні дані: повідомлення про успішне створення/редагування/видалення бренду, оновлений список брендів.

Керування типом:

Вхідні дані: введіть деталі (назва, опис тощо)

Вихідні дані: повідомлення про успішне створення/редагування/видалення типу, оновлений список типів.

4.2 Вимоги до надійності

Програмне забезпечення має працювати без збоїв за умови стабільного підключення до мережі Інтернет.

4.3 Умови експлуатації

Експлуатація програмного забезпечення має задовольняти санітарним правилам та нормам. Стежити за працездатністю програмного забезпечення повинен адміністратор, до обов'язків якого також входять:

- перевірка програмного забезпечення на наявність вірусів не рідше одного разу на день;
- регулярне оновлення антивірусного програмного забезпечення робочої станції.

4.3.1 Кліматичні умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні відповідати вимогам, що занесені до технічних засобів в частині умов їх експлуатації.

4.3.2 Вимоги до чисельності та кваліфікації користувачів

Користувач повинен мати базові навички роботи з комп'ютером.

4.4 Вимоги до складу і параметрів технічних засобів

Для роботи з програмою необхідний наступний склад технічних засобів з мінімальними параметрами.

4.4.1 Вимоги до користувацького обладнання:

- процесор: будь-який сучасний процесор, здатний безперебійно працювати з веб-переглядачем, наприклад Intel Core i3;
- оперативна пам'ять: щонайменше 4 ГБ оперативної пам'яті для плавного перегляду та взаємодії з веб-сайтом;

- дисковий простір: достатньо вільного місця на диску для зберігання тимчасових файлів і кешу браузера;
- роздільна здатність екрана: для оптимального перегляду та зручності використання рекомендована мінімальна роздільна здатність екрана 1280x800 або вище.

4.4.2 Вимоги до серверного обладнання:

- 8-ядерний процесор (наприклад, Intel Xeon або AMD EPYC);
- не менше 32 ГБ оперативної пам'яті;
- жорсткий диск (SSD) з мінімальною ємністю 256 ГБ чи вище.

Ці специфікації забезпечують стабільність, продуктивність та можливості зберігання для операцій інтернет-магазину.

4.5 Вимоги до інформаційної та програмної сумісності

4.5.1 Вимоги до вихідних кодів і мов програмування

Вихідні коди програми повинні бути реалізовані на мові JavaScript. В якості інтегрованого середовища розробки програми може бути використане середовище Visual Studio Code.

4.5.2 Вимоги до програмних засобів

Серверна операційна система для програмного забезпечення має бути Linux Ubuntu 22.04.2 LTS.

Програмне забезпечення має бути сумісним із популярними веббраузерами, такими як Google Chrome, Mozilla Firefox, Microsoft Edge, які підтримують HTML5 та JavaScript ES6.

4.6 Вимоги до маркування та упаковки

Програма поставляється у вигляді коду в електронному вигляді.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання відсутні.

4.8 Спеціальні вимоги

Спеціальні вимоги відсутні.

5 Вимоги до програмної документації

Склад програмної документації повинен включати в себе:

- технічне завдання;
- програма і методика випробувань;
- керівництво користувача;
- опис програми;
- текст програми.

6 Техніко-економічні показники

У даній кваліфікаційній роботі техніко-економічні показники не розраховуються.

7 Стадії та етапи розробки

Етапи розробки програмного забезпечення вказано в таблиці А.1.

Таблиця А.1 – Етапи розробки програмного забезпечення

Номер	Назва етапів роботи	Терміни виконання
1.	Підготовка розділу ВСТУП	24.03.2023
2.	Аналіз практичної задачі інженерії ПЗ	31.03.2023
3.	Аналіз вимог до ПЗ	07.04.2023
4.	Розробка архітектури ПЗ	21.04.2023
5.	Розробка проекту ПЗ	05.05.2023
6.	Реалізація та тестування ПЗ	12.05.2023
7.	Підготовка розділу Результати розробки ПЗ	17.05.2023
8.	Підготовка розділу з охорони праці	19.05.2023
9.	Підготовка розділу ВИСНОВКИ	24.05.2023
10.	Оформлення списку використаних джерел та додатків	26.05.2023
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи	29.05.2023
12.	Підготовка презентації та доповіді	02.06.2023
13.	Попередній захист роботи на засіданні кафедри ПЗАС	05.06.2023
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи; презентації доповіді, а також Авторського договору з додатком	12.06.2023

8 Порядок контролю і прийомки

Контроль здійснюється керівником кваліфікаційної роботи на кожному етапі розробки програмного забезпечення. Приймання здійснюється за програмою та методикою випробувань. Випробування проводиться в зазначений термін.

ДОДАТОК Б – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1 Об'єкт випробувань

Об'єктом випробувань є програмне забезпечення інтернет-магазину компанії “Brand New Outfit”.

2 Мета випробувань

Метою проведення випробувань є перевірка роботи програмного забезпечення інтернет-магазину одягу компанії “Brand New Outfit ”, а саме правильність виконання та роботи функцій, вказаних в документації.

3 Вимоги до програми

При проведенні випробувань програмного забезпечення, функціональні характеристики повинні відповідати вимогам, які були викладенні у підрозділі “Постановка задачі”.

4 Вимоги до документації програми

До складу програмної документації входить:

- технічне завдання програми;
- програма та методика випробувань;
- керівництво користувача;
- опис програми;
- текст програми.

5 Склад та порядок випробувань

5.1 Засоби випробувань

Вимоги до складу й параметрів технічних засобів для проведення випробувань:

- процесор: Intel Core i3 2.30GHz;
- оперативна пам'ять: 8 ГБ;
- роздільна здатність екрана: 1280x800.

Склад програмних засобів:

- операційна система Windows 10;
- браузер Chrome версії 109;
- браузер Firefox версії 112.

5.2 Порядок проведення випробувань

Проведення випробувань починається з перевірки вимог до програмної документації та функціональних характеристик. Потім виконується тестування та аналіз отриманих результатів тестування, яке закінчується перевіркою програмного забезпечення на відповідність програмного продукту вимогам.

ДОДАТОК В – КЕРІВНИЦТВО КОРИСТУВАЧА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

При переході на сторінку інтернет-магазину компанії “Brand New Outfit”, перше, що побачить користувач, буде головна сторінка незареєстрованого користувача (рисунок В.1).

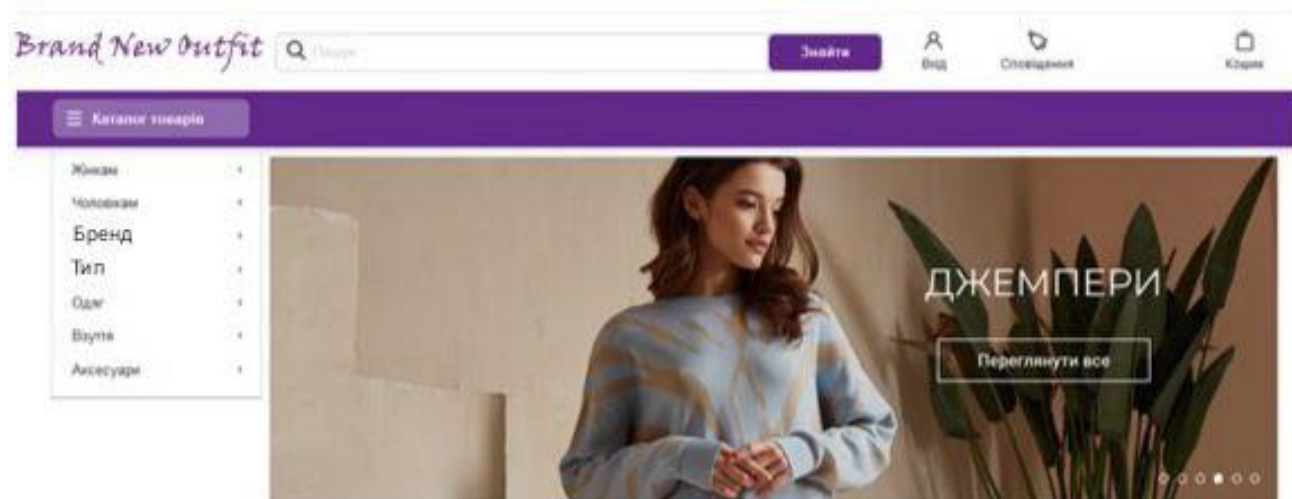


Рисунок В.1 – Головна сторінка інтернет-магазину

На цій сторінці розміщена коротка інформація про сайт, яка потрібна для розуміння куди потрапив користувач, а також розташовані основні функціональні можливості, а саме Реєстрація/Авторизація користувача, пошук товару за фільтром та без, та перегляд товару.

Для реєстрації/авторизації потрібно натиснути на відповідну кнопку Вхід.

Пошук товару здійснюється у полі для пошуку, а також по категоріям. Вибравши якийсь товар його можна переглянути.

Для того щоб мати усі функції доступні користувачу, потрібно увійти у систему, рисунок В.2:

Вхід

Введіть свій E-mail

example@gmail.com

Пароль

мінімум 10 символів

Забули пароль?

Увійти

Реєстрація

Рисунок В.2 - Сторінка входу користувача

Як видно з рисунку В.2, користувачу потрібно авторизуватися або зареєструватися.

Після авторизації користувачу надається додатковий функціонал. При пошуку(рисунок В.3) та перегляді товару (рисунок В.4) користувач має змогу додавати його до кошику. Перед додаванням товару, у кошику зазначено, що він пустий (рисунок В.4).

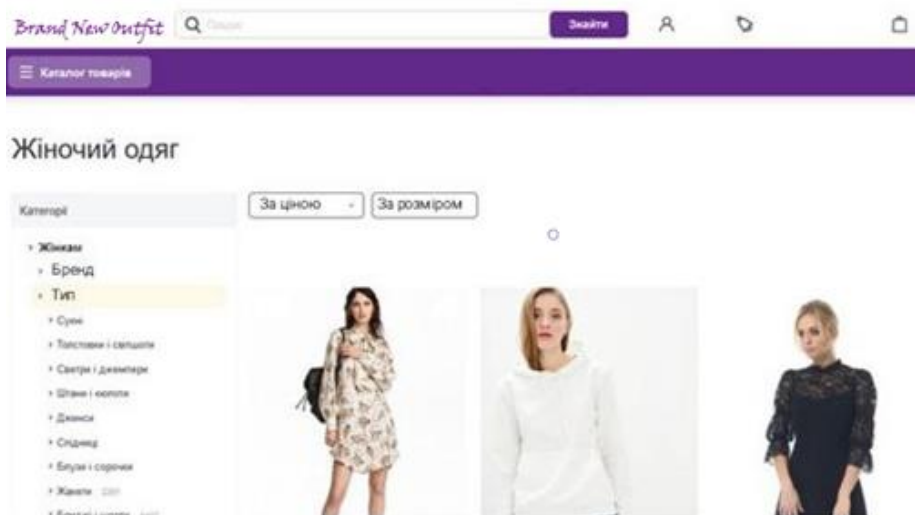


Рисунок В.3 – пошук товару в інтернет-магазині

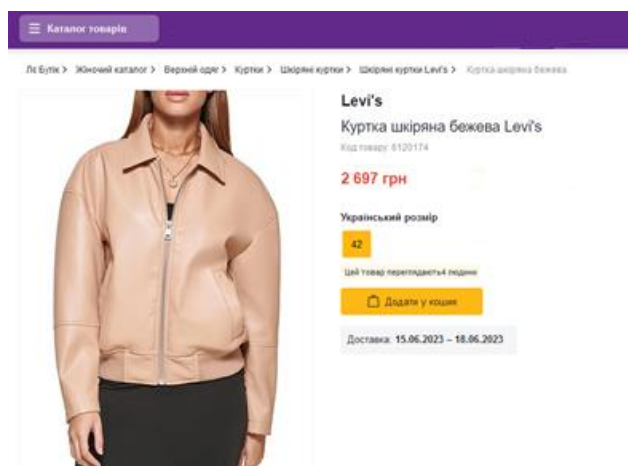


Рисунок В.4 – перегляд товару в інтернет-магазині

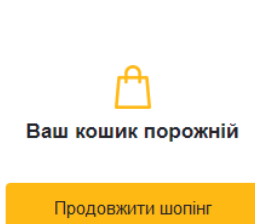


Рисунок В.5 – кошик, перед додаванням товару в нього

Усі товари можна сортувати за брендом (рисунок В.5) та типом. Додавання товару виглядає, як продемонстровано на рисунку В.6.

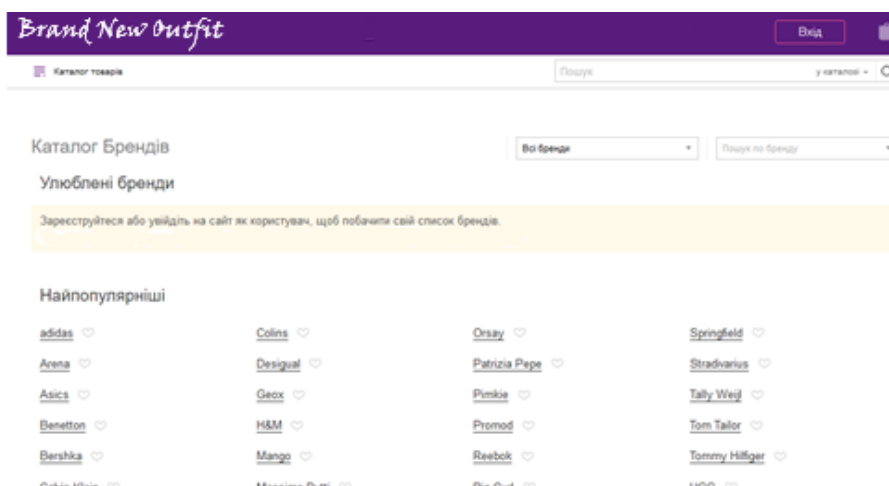


Рисунок В.6 – каталог брендів інтернет-магазину

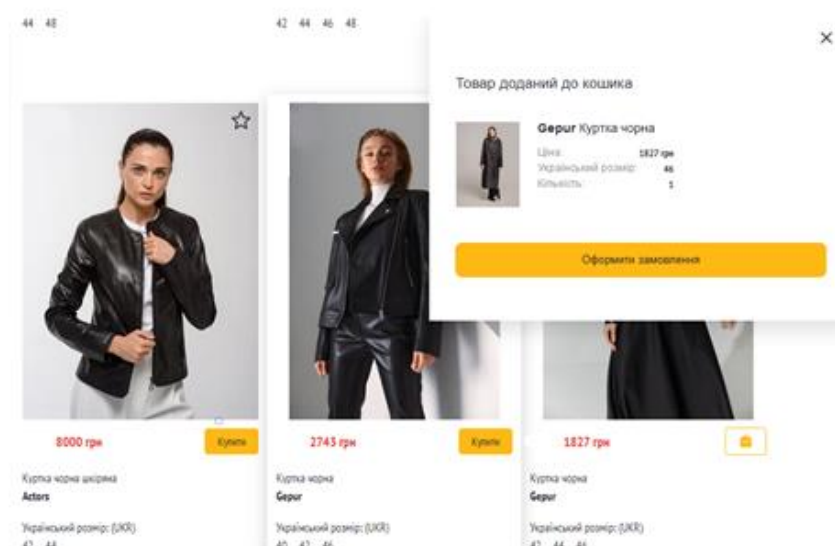


Рисунок В.7 – сповіщення про додавання товару до кошика

Для того, щоб користувач міг оформити замовлення, спочатку потрібно перейти до сторінки Кошика (рисунок В.7). У ньому зазначені усі додані до кошика товари, їх кількість та сума. Товар можна видаляти з кошика, але якщо користувача задовільняють усі товари, то також зазначена загальна сума замовлення.

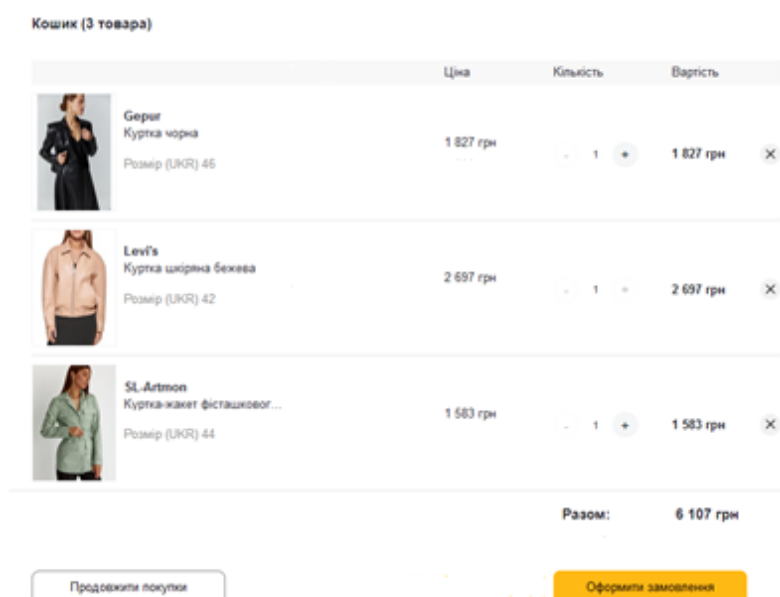


Рисунок В.8 – сторінка кошика з доданими товарами

Якщо користувач хоче оформити замовлення, то побачить сторінку оформлення замовлення, на якій потрібно ввести особисті дані, а також відображаються деталі замовлення, рисунок В.8:

Brand New Outfit

1

Контактні дані одержувача

Ім'я одержувача:

Дар'я

Телефон:

+38

Оформити замовлення

Замовлення №1



Серий

Купити новий

46 (L) 1 шт.

1 827 грн

17:49

Доставка: 16 — 19 червня 2023

Вартість товарів в замовленні: 1 827 грн

Сума: 1 827 грн

Замовлення №2



Levi's

Купити цілий бейкер

42 (XS) 1 шт.

2 697 грн

17:18

Доставка: 15 — 18 червня 2023

Рисунок В.9 – сторінка оформлення замовлення

ДОДАТОК Г – ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вступ

Позначення і найменування програми

Найменування: Програмне забезпечення інтернет-магазину компанії “Brand New Outfit”.

Позначення: інтернет-магазин одягу.

1 Функціональне призначення

Функціональним призначенням програмного забезпечення є створення платформи для перегляду та купівлі одягу.

Сторінка зі списком продуктів є ключовим компонентом інтернет-магазину, оскільки саме тут покупці можуть переглядати та шукати продукти, доступні для покупки. Як правило, на сторінці зі списком продуктів відображається сітка або список продуктів з їх назвами, цінами та зображеннями. Кожен продукт також має короткий опис або список функцій, щоб допомогти клієнтам зрозуміти, що це за продукт і чи відповідає він їх потребам.

Сторінка зі списком продуктів також має фільтри або параметри сортування, щоб допомогти клієнтам звужити область пошуку та знайти продукти, які найкраще відповідають їхнім потребам. Ці фільтри можуть включати такі параметри, як ціновий діапазон, категорія продукту, бренд, розмір, колір або інші характеристики, характерні для типу продукту, що продається.

Список продуктів також містить посилання на докладніші сторінки продуктів або огляди. Це може допомогти покупцям дізнатися більше про продукт перед покупкою. В цілому, сторінка зі списком продуктів є важливим компонентом будь-якого інтернет-магазину, оскільки вона відіграє ключову роль у залученні та перетворенні потенційних клієнтів на клієнтів, що платять.

Кошик також є важливим компонентом інтернет-магазину, оскільки він дозволяє покупцям додавати товари у своє замовлення. Як правило, у кошику

відображаються назва, зображення, ціна та кількість кожного товару в замовленні, а також проміжна сума замовлення.

Після того, як покупець додав усі бажані товари в кошик, він переходить на сторінку оформлення замовлення. На етапі оформлення замовлення покупець завершує покупку, вводячи інформацію про доставку, вибираючи спосіб оплати та переглядаючи деталі замовлення і загальну вартість замовлення перед його відправкою. Як правило, на етапі оформлення замовлення відображається форма, в якій покупець може ввести особисту інформацію, а також варіанти доставки та оплати.

2 Опис логіки

2.1 Опис структури

Програмне забезпечення інтернет-магазину компанії “Brand New Outfit” складається з двох частин: клієнтської та серверної.

Клієнтська частина це графічний інтерфейс, який відображається у браузері. Серверна частина - обробляє запити користувача на сервері, а також включає в себе взаємодію з базою даних, яка забезпечує зберігання даних та їх видачу в момент запиту.

2.2 Мова програмування

Реалізовувався проект використовуючи стек PERN, тобто чотири технології: PostgreSQL, Express.js, React.js і Node.js. Для розробки програмного забезпечення використовувалося середовище розробки Visual Studio Code.

2.3 Вхідні та вихідні дані

Вхідні дані - введені користувачем дані і запити, надіслані від користувача до сервера.

Вихідні дані - відповіді сервера, запити до бази даних і відповіді, надіслані назад клієнту.

2.4 Алгоритм програми

Алгоритм взаємодії клієнта з сервером та сервера з базою даних:

- 1) користувач заходить на сторінку інтернет-магазину у браузері;
- 2) браузер відсилає запит на сервер;
- 3) отримавши запит від браузера користувача, сервер підтверджує вхідний запит і починає його обробку;
- 4) сервер визначає відповідний маршрут або кінцеву точку на основі запитаної URL-адреси та методу HTTP;
- 5) сервер виконує код, пов'язаний із запитуваним маршрутом;
- 6) якщо серверному коду потрібні дані з бази даних, він з нею взаємодіє, для отримання або оновлення необхідних даних;
- 7) код на стороні сервера перевіряє введені користувачем дані та готує необхідні дані для створення відповіді;
- 8) на основі виконання коду на стороні сервера, генерує відповідь. Ця відповідь зазвичай містить код HTML, CSS і JavaScript, який буде відтворено браузером користувача;
- 9) сервер надсилає згенеровану відповідь назад у браузер користувача;
- 10) браузер користувача отримує відповідь від сервера та починає її обробку;
- 11) візуалізація на стороні клієнта, тобто відображення вмісту та елементів інтерфейсу користувача за призначенням;
- 12) користувач може взаємодіяти з відтвореною вебсторінкою, натискаючи посилання, надсилаючи форми або виконуючи інші дії;
- 13) повторення: користувач може продовжувати перегляд онлайн-магазину, і процес повторюється для кожного нового запиту та циклу відповіді. Сервер обробляє наступні запити, виконує необхідний серверний код, взаємодіє з базою даних, генерує відповіді, а браузер відтворює оновлений вміст.

3 Склад та функції

3.1 Склад програмного забезпечення

Інтернет-магазин складається з наступних сторінок:

- список продуктів;
- кошик;
- оформлення замовлення.

3.2 Функції програмного забезпечення

Програмне забезпечення інтернет-магазину компанії “Brand New Outfit” реалізовує наступні функції:

- реєстрація;
- авторизація;
- керування обліковим записом;
- вихід з облікового запису;
- перегляд товару;
- пошук товару;
- керування кошиком;
- оформлення замовлення;
- керування замовлення;
- керування товарами;
- керування брендами;
- керування типами.

4 Умови застосування

4.1 Вимоги до складу й параметрів технічних засобів:

Серверне обладнання повинно включати:

- 8-ядерний процесор (наприклад, Intel Xeon або AMD EPYC);
- не менше 32 ГБ оперативної пам'яті;

- жорсткий диск (SSD) з мінімальною ємністю 256 ГБ чи вище.

Вимоги до користувацького обладнання:

- процесор: будь-який сучасний процесор, здатний безперебійно працювати з веб-переглядачем, наприклад Intel Core i3;
- оперативна пам'ять: щонайменше 4 ГБ оперативної пам'яті для плавного перегляду та взаємодії з веб-сайтом;
- дисковий простір: достатньо вільного місця на диску для зберігання тимчасових файлів і кешу браузера;
- роздільна здатність екрана: для оптимального перегляду та зручності використання рекомендована мінімальна роздільна здатність екрана 1280x800 або вище.

Також має бути встановлений браузер з наведених у програмних вимогах та доступ до інтернету.

4.2 Функціональні обмеження на застосування

Найголовнішим обмеженням на застосування програмного забезпечення є необхідність підключення до мережі інтернету.

4.3 Виклик та завантаження програмного забезпечення

Виклик та завантаження програмного забезпечення інтернет-магазину компанії “Brand New Outfit” починається з встановлення чи використання вже встановленого браузера, в якому користувачу потрібно ввести URL адресу сайту.

ДОДАТОК Д – ТЕКСТ ПРОГРАМИНОГО ЗАБЕЗПЕЧЕННЯ

App.js - цей файл є точкою входу програми React

```
import React, {useContext, useEffect, useState} from 'react';
import {BrowserRouter} from "react-router-dom";
import AppRouter from "./components/AppRouter";
import NavBar from "./components/NavBar";
import {observer} from "mobx-react-lite";
import {Context} from "./index";
import {check} from "./http/userAPI";
import {Spinner} from "react-bootstrap";

const App = observer(() => {
  const {user} = useContext(Context)
  const [loading, setLoading] = useState(true)

  useEffect(() => {
    check().then(data => {
      user.setUser(true)
      user.setIsAuth(true)
    }).finally(() => setLoading(false))
  }, [])

  if (loading) {
    return <Spinner animation={"grow"}/>
  }

  return (
    <BrowserRouter>
```

```

    <NavBar />
    <AppRouter />
  </BrowserRouter>
);
});

```

```
export default App;
```

Admin.js – цей компонент представляє сторінку адміністратора та містить кнопки для додавання типів, брендів і товарів.

```

import React, { useState } from "react";
import { Button, Container } from "react-bootstrap";
import CreateBrand from "../components/modals/CreateBrand";
import CreateProduct from "../components/modals/CreateProduct";
import CreateType from "../components/modals/CreateType";

const Admin = () => {
  const [brandVisible, setBrandVisible] = useState(false);
  const [typeVisible, setTypeVisible] = useState(false);
  const [productVisible, setProductVisible] = useState(false);

  return (
    <Container className="d-flex flex-column">
      <Button
        variant={"outline-dark"}
        className="mt-4 p-2"
        onClick={() => setTypeVisible(true)}
      >
        Додати тип
      </Button>

```

```

<Button
  variant={"outline-dark"}
  className="mt-4 p-2"
  onClick={() => setBrandVisible(true)}
>
  Додати бренд
</Button>

<Button
  variant={"outline-dark"}
  className="mt-4 p-2"
  onClick={() => setProductVisible(true)}
>
  Додати товар
</Button>

<CreateBrand show={brandVisible} onHide={() => setBrandVisible(false)} />
<CreateProduct
  show={productVisible}
  onHide={() => setProductVisible(false)}
/>
<CreateType show={typeVisible} onHide={() => setTypeVisible(false)} />
</Container>
);
};

export default Admin;

```

Auth.js – цей компонент представляє сторінку автентифікації та містить форму для входу та реєстрації.

```

import React, { useContext, useState } from "react";
import { Container, Form } from "react-bootstrap";

```

```

import Card from "react-bootstrap/Card";
import Button from "react-bootstrap/Button";
import Row from "react-bootstrap/Row";
import { NavLink, useLocation, useHistory } from "react-router-dom";
import { LOGIN_ROUTE, REGISTRATION_ROUTE, SHOP_ROUTE } from
"../utils/consts";
import { login, registration } from "../http/userAPI";
import { observer } from "mobx-react-lite";
import { Context } from "../index";

```

```

const Auth = observer(() => {
  const { user } = useContext(Context);
  const location = useLocation();
  const history = useHistory();
  const isLogin = location.pathname === LOGIN_ROUTE;
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

```

```

const click = async () => {
  try {
    let data;
    if (isLogin) {
      data = await login(email, password);
    } else {
      data = await registration(email, password);
    }
    user.setUser(user);
    user.setIsAuth(true);
    history.push(SHOP_ROUTE);
  } catch (e) {

```

```

    alert(e.response.data.message);
  }
};

return (
  <Container
    className="d-flex justify-content-center align-items-center"
    style={{ height: window.innerHeight - 54 }}
  >
    <Card style={{ width: 600 }} className="p-5">
      <h2 className="m-auto">{isLoggedIn ? "Авторизація" : "Реєстрація"}</h2>
      <Form className="d-flex flex-column">
        <Form.Control
          className="mt-3"
          placeholder="Введіть ваш email..."
          value={email}
          onChange={(e) => setEmail(e.target.value)}
        />
        <Form.Control
          className="mt-3"
          placeholder="Введіть ваш пароль..."
          value={password}
          onChange={(e) => setPassword(e.target.value)}
          type="password"
        />
        <Row className="d-flex justify-content-between mt-3 pl-3 pr-3">
          {isLoggedIn ? (
            <div>
              Немає акаунта? {" "}
              <NavLink to={REGISTRATION_ROUTE}>Зареєструйтеся!</NavLink>
            </div>
          ) : (
            <div>
              <NavLink to={LOGIN_ROUTE}>Вхід</NavLink>
            </div>
          )}
        </Row>
      </Form>
    </Card>
  </Container>
);

```

```

    </div>
  ) : (
    <div>
      Є акаунт? <NavLink to={LOGIN_ROUTE}>Увійдіть!</NavLink>
    </div>
  )}
  <Button variant={"outline-success"} onClick={click}>
    {isLogin ? "Увійти" : "Реєстрація"}
  </Button>
</Row>
</Form>
</Card>
</Container>
);
});

export default Auth;

```

Shop.js - цей компонент представляє сторінку магазину та відображає список товарів. Він включає панель типу та панель бренду для фільтрації.

```

import React, { useContext, useEffect } from "react";
import { Container } from "react-bootstrap";
import Row from "react-bootstrap/Row";
import Col from "react-bootstrap/Col";
import TypeBar from "../components/TypeBar";
import BrandBar from "../components/BrandBar";
import ProductList from "../components/ProductList";
import { observer } from "mobx-react-lite";
import { Context } from "../index";
import { fetchBrands, fetchProduct, fetchTypes } from "../http/productAPI";

```

```

import Pages from "../components/Pages";

const Shop = observer(() => {
  const { product } = useContext(Context);

  useEffect(() => {
    fetchTypes().then((data) => product.setTypes(data));
    fetchBrands().then((data) => product.setBrands(data));
    fetchProduct(null, null, 1, 2).then((data) => {
      product.setProduct(data.rows);
      product.setTotalCount(data.count);
    });
  }, []);

  useEffect(() => {
    fetchProducts(
      product.selectedType.id,
      product.selectedBrand.id,
      product.page,
      2
    ).then((data) => {
      product.setProducts(data.rows);
      product.setTotalCount(data.count);
    });
  }, [product.page, product.selectedType, product.selectedBrand]);

  return (
    <Container>
      <Row className="mt-2">
        <Col md={3}>

```

```

        <TypeBar />
    </Col>
    <Col md={9}>
        <BrandBar />
        <ProductList />
        <Pages />
    </Col>
</Row>
</Container>
);
});

```

```
export default Shop;
```

Basket.js - цей компонент представляє сторінку кошика

```

import React from 'react';
import React, { useEffect } from 'react';
import { useContext } from 'react';
import { Context } from '..';
import { getBasket } from '../http/productAPI';

const Basket = () => {
    import { Card, Col, Container, Row } from 'react-bootstrap'
    import { observer } from 'mobx-react-lite';
    //import close from '../assets/close.svg'
    const Basket = observer(() => {
        const {product} = useContext(Context)
        useEffect(() => {
            getBasket().then(data => product.setBaskets(data))
        }, [])
    })
}

```

```

let prices = 0;
{product.basket.map(price =>
  prices += Number(price.product.price)
)}
return (
  <div>
    basket
  </div>
  <Container
    className="d-flex flex-sm-column justify-content-center align-items-center mt-
3"
  >
    <h1 className="pb-2">Кошик</h1>
    <Card className="d-flex flex-row p-2 justify-content-between align-items-
center mb-2">
      <h1 className="pr-2">Итого:</h1>
      <h3 className="pl-2">{prices}<span className="font-weight-light pl-
2">рублей</span></h3>
    </Card>
    {product.basket.map(product =>
      <Card className="d-flex w-100 p-2 justify-content-center mb-2"
key={product.id}>
        <Row className="d-flex w-100">
          <Col>
            <div className="d-flex flex-row align-items-center">
              <img src={process.env.REACT_APP_API_URL + product.img}
width={50} />
              <h1 className="pl-3">{product.name}</h1>
            </div>
          </Col>

```

```

        <Col>
          <div className="d-flex h-100 flex-row justify-content-end align-items-center">
            <h2 className="font-weight-light">{product.price} грн</h2>
          </div>
        </Col>
      </Row>
    </Card>
  )}
</Container>
);
};
});

```

export default Basket;

ProductPage.js – цей компонент представляє сторінку товару та відображає детальну інформацію про один товар.

```

import React, { useEffect, useState } from "react";
import { Button, Card, Col, Container, Image, Row } from "react-bootstrap";
import bigStar from "../assets/bigStar.png";
import { useParams } from "react-router-dom";
import { fetchOneProduct } from "../http/productAPI";

const ProductPage = () => {
  const [product, setProduct] = useState({ info: [] });
  const { id } = useParams();
  useEffect(() => {
    fetchOneProduct(id).then((data) => setProduct(data));
  }, []);

```

```

return (
  <Container className="mt-3">
    <Row>
      <Col md={4}>
        <Image
          width={300}
          height={300}
          src={process.env.REACT_APP_API_URL + product.img}
        />
      </Col>
      <Col md={4}>
        <Row className="d-flex flex-column align-items-center">
          <h2>{product.name}</h2>
          <div
            className="d-flex align-items-center justify-content-center"
            style={{
              background: `url(${bigStar}) no-repeat center center`,
              width: 240,
              height: 240,
              backgroundSize: "cover",
              fontSize: 64,
            }}
          >
            {product.rating}
          </div>
        </Row>
      </Col>
      <Col md={4}>
        <Card

```

```

        className="d-flex flex-column align-items-center justify-content-around"
        style={{
            width: 300,
            height: 300,
            fontSize: 32,
            border: "5px solid lightgray",
        }}
    >
        <h3>От: {product.price} грн.</h3>
        <Button variant={"outline-dark"}>Додати в кошик</Button>
    </Card>
</Col>
</Row>
<Row className="d-flex flex-column m-3">
    <h1>Характеристики</h1>
    {product.info.map((info, index) => (
        <Row
            key={info.id}
            style={{
                background: index % 2 === 0 ? "lightgray" : "transparent",
                padding: 10,
            }}
        >
            {info.title}: {info.description}
        </Row>
    ))}
</Row>
</Container>
);
};

```

```
export default ProductPage;
```

CreateBrand.js - цей компонент представляє сторінку створення бренду адміністратором

```
import React, { useState } from "react";
import Modal from "react-bootstrap/Modal";
import { Button, Form } from "react-bootstrap";
import { createBrand, createType } from "../http/productAPI";

const CreateBrand = ({ show, onHide }) => {
  const [value, setValue] = useState("");

  const addBrand = () => {
    createBrand({ name: value }).then((data) => {
      setValue("");
      onHide();
    });
  };

  return (
    <Modal show={show} onHide={onHide} centered>
      <Modal.Header closeButton>
        <Modal.Title id="contained-modal-title-vcenter">Додати тип</Modal.Title>
      </Modal.Header>
      <Modal.Body>
        <Form>
          <Form.Control
            value={value}
            onChange={(e) => setValue(e.target.value)}
            placeholder="Введіть назву типу">

```

```

    />
  </Form>
</Modal.Body>
<Modal.Footer>
  <Button variant="outline-danger" onClick={onHide}>
    Закрити
  </Button>
  <Button variant="outline-success" onClick={addBrand}>
    Додати
  </Button>
</Modal.Footer>
</Modal>
);
};

```

```
export default CreateBrand;
```

CreateType.js - цей компонент представляє сторінку створення типу адміністратором

```

import React, { useState } from "react";
import Modal from "react-bootstrap/Modal";
import { Form, Button } from "react-bootstrap";
import { createType } from "../../http/productAPI";

```

```

const CreateType = ({ show, onHide }) => {
  const [value, setValue] = useState("");

  const addType = () => {
    createType({ name: value }).then((data) => {
      setValue("");
    });
  };

```

```

    onHide();
  });
};

return (
  <Modal show={show} onHide={onHide} centered>
    <Modal.Header closeButton>
      <Modal.Title id="contained-modal-title-vcenter">Додати тип</Modal.Title>
    </Modal.Header>
    <Modal.Body>
      <Form>
        <Form.Control
          value={value}
          onChange={(e) => setValue(e.target.value)}
          placeholder={"Введіть назву типу"}
        />
      </Form>
    </Modal.Body>
    <Modal.Footer>
      <Button variant="outline-danger" onClick={onHide}>
        Закрити
      </Button>
      <Button variant="outline-success" onClick={addType}>
        Додати
      </Button>
    </Modal.Footer>
  </Modal>
);
};

```

```
export default CreateType;
```

CreateProduct.js - цей компонент представляє сторінку створення товару адміністратором

```
import React, { useContext, useEffect, useState } from "react";
import Modal from "react-bootstrap/Modal";
import { Button, Dropdown, Form, Row, Col } from "react-bootstrap";
import { Context } from "../../index";
import {
  createProduct,
  fetchBrands,
  fetchProducts,
  fetchTypes,
} from "../../http/productAPI";
import { observer } from "mobx-react-lite";

const CreateProduct = observer(({ show, onHide }) => {
  const { product } = useContext(Context);
  const [name, setName] = useState("");
  const [price, setPrice] = useState(0);
  const [file, setFile] = useState(null);
  const [info, setInfo] = useState([]);

  useEffect(() => {
    fetchTypes().then((data) => product.setTypes(data));
    fetchBrands().then((data) => product.setBrands(data));
  }, []);

  const addInfo = () => {
    setInfo([...info, { title: "", description: "", number: Date.now() }]);
```

```

};

const removeInfo = (number) => {
  setInfo(info.filter((i) => i.number !== number));
};

const changeInfo = (key, value, number) => {
  setInfo(
    info.map((i) => (i.number === number ? { ...i, [key]: value } : i))
  );
};

const selectFile = (e) => {
  setFile(e.target.files[0]);
};

const addProduct = () => {
  const formData = new FormData();
  formData.append("name", name);
  formData.append("price", `${price}`);
  formData.append("img", file);
  formData.append("brandId", product.selectedBrand.id);
  formData.append("typeId", product.selectedType.id);
  formData.append("info", JSON.stringify(info));
  createProduct(formData).then((data) => onHide());
};

return (
  <Modal show={show} onHide={onHide} centered>
    <Modal.Header closeButton>
      <Modal.Title id="contained-modal-title-vcenter">
        Додати товар

```

```

</Modal.Title>
</Modal.Header>
<Modal.Body>
  <Form>
    <Dropdown className="mt-2 mb-2">
      <Dropdown.Toggle>
        {product.selectedType.name || "Виберіть тип"}
      </Dropdown.Toggle>
      <Dropdown.Menu>
        {product.types.map((type) => (
          <Dropdown.Item
            onClick={() => product.setSelectedType(type)}
            key={type.id}
          >
            {type.name}
          </Dropdown.Item>
        ))}
      </Dropdown.Menu>
    </Dropdown>
    <Dropdown className="mt-2 mb-2">
      <Dropdown.Toggle>
        {product.selectedBrand.name || "Виберіть тип"}
      </Dropdown.Toggle>
      <Dropdown.Menu>
        {product.brands.map((brand) => (
          <Dropdown.Item
            onClick={() => product.setSelectedBrand(brand)}
            key={brand.id}
          >
            {brand.name}

```

```

        </Dropdown.Item>
      )}}
    </Dropdown.Menu>
  </Dropdown>
  <Form.Control
    value={name}
    onChange={(e) => setName(e.target.value)}
    className="mt-3"
    placeholder="Введіть назву товару"
  />
  <Form.Control
    value={price}
    onChange={(e) => setPrice(Number(e.target.value))}
    className="mt-3"
    placeholder="Введіть вартість товару"
    type="number"
  />
  <Form.Control className="mt-3" type="file" onChange={selectFile} />
  <hr />
  <Button variant={"outline-dark"} onClick={addInfo}>
    Додати нову властивість
  </Button>
  {info.map((i) => (
    <Row className="mt-4" key={i.number}>
      <Col md={4}>
        <Form.Control
          value={i.title}
          onChange={(e) =>
            changeInfo("title", e.target.value, i.number)
          }

```

```

        placeholder="Введіть назву властивості"
    />
</Col>
<Col md={4}>
    <Form.Control
        value={i.description}
        onChange={(e) =>
            changeInfo("description", e.target.value, i.number)
        }
        placeholder="Введіть опис властивості"
    />
</Col>
<Col md={4}>
    <Button
        onClick={() => removeInfo(i.number)}
        variant={"outline-danger"}
    >
        Видалити
    </Button>
</Col>
</Row>
    )}
</Form>
</Modal.Body>
<Modal.Footer>
    <Button variant="outline-danger" onClick={onHide}>
        Закрити
    </Button>
    <Button variant="outline-success" onClick={addProduct}>
        Додати

```

```
        </Button>
      </Modal.Footer>
    </Modal>
  );
});

export default CreateProduct;
```