Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: Розробка інтернет-магазину Vortex

Виконав: студент IV курсу, групи СН-42
спеціальності 122 Комп’ютерні науки
(шифр і назва спеціальності)

(підпис) Шабля Р. А. (прізвище та ініціали)

Керівник (підпис) Гром’як Р. С. (прізвище та ініціали)

Нормоконтроль (підпис) Шимчук Г. В. (прізвище та ініціали)

Завідувач кафедри (підпис) Боднарчук І.О. (прізвище та ініціали)

Рецензент (підпис) (прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
«__» _червня_ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Шаблі Руслану Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інтернет-магазину Vortex

Керівник роботи Гром'як Роман Сильвестрович, к.ф.-м.н., доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «*29*» *квітня* 2024 року № *4/7-470*

2. Термін подання студентом завершеної роботи 24 червня 2024р.

3. Вихідні дані до роботи Літературні та інтернет джерела

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1. Аналіз предметної області. Постановка завдання та вибір середовища розробки. 1.1 Аналіз предметної. 1.2 Огляд існуючих рішень. 1.3 Постановка завдання. 1.4 Вибір середовища розробки. 1.5 Висновок першого розділу. 2.Проектування архітектури інтернет-магазину та бази даних. 2.1 Проектування архітектури інтернет-магазину. 2.2 Визначення акторів та варіанти використання інтернет-магазину. 2.3 Проектування архітектури бази даних. 2.4 Висновок другого розділу. 3. Розробка інтернет-магазину Vortex. 3.1 Розробка серверної частини на Базі Node.js. 3.2 Розробка клієнтської частини з використанням React.js. 3.3 Висновок третього розділу. 4. Безпека життєдіяльності. Основи охорони праці. 4.1 Долікарська допомога при ураженні електричним струмом. 4.2 Вимоги ергономіки до організації робочого місця оператора ПК. 4.3 Висновок четвертого розділу. Висновки. Список використаних джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В.С. к.т.н., доцент кафедри МТ		

7. Дата видачі завдання 29 січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	30.01.2024	*Виконано*
2.	Підбір джерел для розробки інтернет-магазину	31.01.2024-03.02.2024	*Виконано*
3.	Опрацювання джерел по темі кваліфікаційної роботи	04.02.2024-06.02.2024	*Виконано*
4.	Виконання дослідження щодо інтернет-магазинів.	07.02.2024-11.02.2024	*Виконано*
	Розроблення інтернет-магазину Vortex.		
5.	Оформлення розділу «Аналіз предметної області. Постановка завдання та вибір середовища розробки»		
6.	Оформлення розділу «Проектування архітектури інтернет-магазину та бази даних»		
7.	Оформлення розділу «Розробка інтернет-магазину Vortex»		
8.	Виконання завдання до підрозділу «Безпека життєдіяльності»		
9.	Виконання завдання до підрозділу «Основи охорони		
	праці»		
10.	Оформлення кваліфікаційної роботи		
11.	Нормоконтроль		
12.	Перевірка на плагіат		
13.	Попередній захист кваліфікаційної роботи		
14.	Захист кваліфікаційної роботи	28.06.2024	

Студент _____ Шабля Р. А.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Гром'як Р. С.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка інтернет-магазину Vortex // Кваліфікаційна робота освітнього рівня «Бакалавр» // Шабля Руслан Андрійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-42 // Тернопіль, 2024 // С. 63, рис. – 42, табл. – 11, слайди – 12, додат. – 2 , бібліогр. – 38.

Ключові слова: react, node, express, компоненти, api, mongodb, javascript, інтернет-магазин.

Мета кваліфікаційної роботи полягає в розробці інтернет-магазину Vortex з використанням сучасних технологій та дослідженні практичних аспектів впровадження, спрямованих на покращення зручності користування та ефективності бізнесу.

У першому розділі проведено глибокий аналіз предметної області, оглянуто різноманітні існуючі рішення на ринку, детально сформульовано постановку завдання та вибрано оптимальне середовище розробки для досягнення поставлених цілей.

Другий розділ присвячено детальному проектуванню архітектури системи, визначенню ролей акторів інтернет-магазину, розглянуті різні варіанти використання акторів та розроблено ефективну архітектуру бази даних для оптимального функціонування системи.

У третьому розділі детально описано реалізацію серверної та клієнтської частин інтернет-магазину з використанням сучасних технологій, таких як Node.js для серверу та React.js для клієнтської частини.

У четвертому розділі детально проаналізовано питання долікарської допомоги при ураженні електричним струмом та розглянуто вимоги ергономіки до організації робочого місця оператора ПК.

ANNOTATION

Development of Vortex Online Store // Qualification work of the educational level «Bachelor» // Shablia Ruslan Andriyovych // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-42 // Ternopil, 2024 // P. 63, fig. – 42, tabl. –11, chair. – 12, annexes. – 2, references – 38.

Keywords: react, Node, Express, components, api, mongodb, javascript, online store.

The goal of the qualification work is to develop the Vortex online store using modern technologies and explore practical aspects of its implementation aimed at improving user convenience and business efficiency.

In the first section, a thorough analysis of the subject area is conducted, various existing solutions on the market are reviewed, the task is formulated in detail, and the optimal development environment is selected to achieve the set goals.

The second section is dedicated to detailed system architecture design, defining the roles of the online store actors, considering various usage scenarios, and developing an efficient database architecture for the optimal functioning of the system.

The third section provides a detailed description of the implementation of the server and client parts of the online store using modern technologies such as Node.js for the server and React.js for the client.

The fourth section comprehensively analyzes issues related to medical assistance in cases of electric shock and the requirements of ergonomics for the organization of the workplace of the PC operator are considered.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ТЕРМІНІВ

API (Application Programming Interface) – інтерфейс прикладного програмування.

DOM (Document Object Model) – об'єктна модель документа.

JS – Java Script.

JWT (JSON Web Token) – стандарт представлення токенів безпеки.

NPM (Node Package Manager) – менеджер пакетів для Node.js.

БД – база даних.

ЗМІСТ

ВСТУП

У сучасному світі інформаційні технології є невід'ємною складовою розвитку суспільства та просування економіки. Зокрема, інтернет-технології стали потужним інструментом для ведення бізнесу, що значно змінює традиційні підходи до торгівлі. Сьогодні інтернет-магазини набувають все більшої популярності та стають невід'ємною частиною бізнесу.

Інтернет-магазини забезпечують значні переваги для покупців та продавців. Для покупців основною перевагою використання є зручність, оскільки з’являється можливість здійснювати покупки будь-де та будь-коли, використовуючи лише девайс з підключенням до інтернету, що дуже економить час та зусилля, враховуючи теперішній швидкий темп життя, а для продавців відкриваються нові можливості для бізнесу, дозволяючи залучати клієнтів з усього світу, розширюючи таким чином ринки збуту. Також перевагою для продавців є можливість знизити витрати на оренду фізичних приміщень та утримання великого штату працівників. Крім того, завдяки автоматизації багатьох процесів, таких як облік товарів, обробка замовлень та управління клієнтськими даними, підвищується ефективність бізнесу.

Розробка інтернет-магазину передбачає впровадження сучасних технологій, що забезпечить високий рівень користувацького досвіду, безпеку та стабільність системи. Це сприятиме підвищенню конкурентоспроможності на ринку та задоволенню потреб сучасних споживачів, які все частіше обирають онлайн-покупки як основний спосіб придбання товарів.

На основі цього можна зробити висновок що розробка інтернет-магазину є актуальним напрямом дослідження.

Метою даної роботи є розробка інтернет-магазину Vortex, який буде відповідати сучасним вимогам споживачів, забезпечуючи високу зручність користування, безпеку транзакцій та ефективне управління товарними запасами.

Щоб здійснити поставлену мету, потрібно виконати певні задачі, а саме:

– проаналізувати предметну область;

– розробити схеми для зберігання даних та спроектувати базу даних;

– провести проектування робочої архітектури;

– створити зручний користувацький інтерфейс;

– провести тестування працездатності інтернет-магазину.

Після виконання усіх перелічених пунктів, буде реалізовано готовий до використання інтернет-магазин. Впровадження інтернет-магазину надасть залучення клієнтів з різних регіонів, сприяючи збільшення продажів та розширення популярності бренду.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. ПОСТАНОВКА ЗАВДАННЯ ТА ВИБІР СЕРЕДОВИЩА РОЗРОБКИ

1.1 Аналіз предметної області

У сучасному світі, інтернет-магазин є потужним методом просування бізнесу чи бренду, забезпечуючи зручний доступ до товарів через інтернет. Зараз важко уявити компанію чи бренд без інтернет-магазину, оскільки це дозволяє залучити нових клієнтів та підвищити рівень продажів. Інтернет-магазини мають змогу забезпечувати цілодобову доступність товарів і послуг, спрощувати процес купівлі товарів та зменшувати витрати на утримання фізичних приміщень, крім того, інтернет-магазини сприяють підвищенню популярності бренду. Для покупців така тенденція є привабливою, оскільки не потрібно витрачати багато часу на пошук товару [1].

Для комфорту користувачів, кожен інтернет-магазин повинен включати в себе зрозумілий інтерфейс, що дозволить користувачам легко знаходити потрібні товари, здійснювати покупки та керувати аккаунтом. Так як, не кожна людина має у власності ПК чи ноутбук, то надзвичайно важливим аспектом є оптимізація інтернет-магазину під мобільні пристрої.

Крім того, структура інтернет-магазину повинна включати головну сторінку, яка містить основну інформацію про магазин, популярні товари, акції та новинки, каталог товарів, що поділений на категорії, щоб користувачі мали змогу швидко знаходити потрібні товари за допомогою фільтрів, пошуку та сортування. Також до основних компонентів інтернет-магазину включають картку товару, що містить в собі детальну інформацію про конкретний товар, включаючи опис, характеристики, фотографії, відгуки та ціни, кошик покупок, де користувачі можуть переглядати обрані товари, змінювати їх кількість або відмінити перед замовленням певний товар. Аккаунт користувача є не менш важливим за інші компоненти, оскільки це особистий кабінет у якому

користувач може переглядати історію замовлень, змінювати персональні дані, налаштовувати способи оплати та доставки [2].

1.2 Огляд існуючих рішень

На даний момент на ринку існує безліч різних інтернет-магазинів. Таким чином, щоб розробити хороший інтернет-магазин вкрай важливо провести аналіз найпопулярніших та найзручніших інтернет-магазинів на думку користувачів, щоб визначити основні моменти розробки. Аналіз існуючих рішень допоможе знайти важливу інформацію про різні характеристики, компоненти та функціональні можливості кожного інтернет-магазину.

Для аналізу були обрані такі українські інтернет-магазини як: «Staff», «Hector» та «LC Waikiki».

Staff – це український бренд, відомий своїм стильним та якісним одягом, що спеціалізується на молоду аудиторію. В інтернет-магазині Staff можна знайти різноманітний одяг, включаючи футболки, худі, джинси, куртки та аксесуари [3].

На рисунку 1.1 зображено головну сторінку інтернет-магазину Staff.

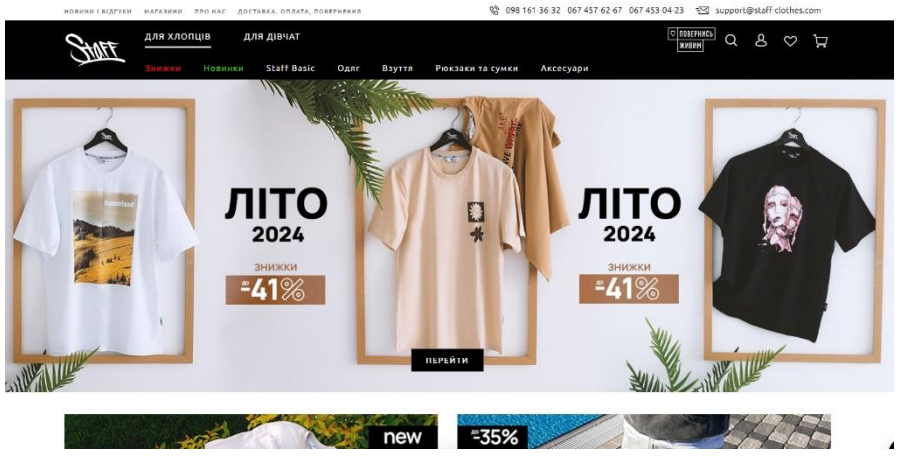

Рисунок 1.1 – Головна сторінка інтернет-магазину Staff

Hector – це український бренд одягу, який спеціалізується на стильному та зручному одязі для чоловіків. Інтернет-магазин Hector відомий яскравими та

незвичайними дизайнами [4]. На рисунку 1.2 зображено головну сторінку інтернет-магазину Hector.

Рисунок 1.2 – Головна сторінка інтернет-магазину Hector

LC Waikiki – це міжнародний бренд з турецьким корінням, що має значну присутність на українському ринку. На рисунку 1.3 зображено головну сторінку інтернет-магазину LC Waikiki.

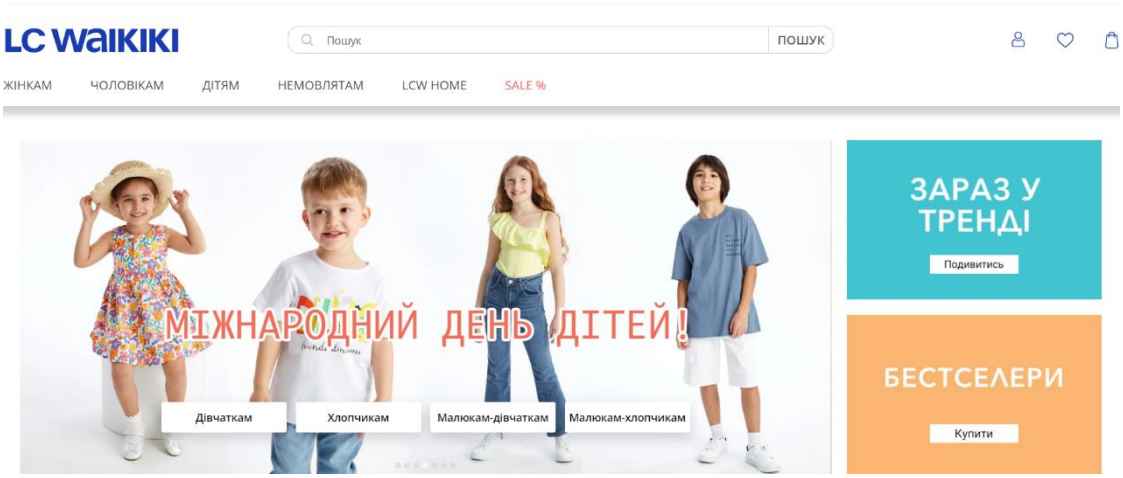

Рисунок 1.3 – Головна сторінка інтернет-магазину LC Waikiki

Для зручного порівняння даних інтернет-магазинів було вирішено створити таблицю, у якій містяться усі необхідні характеристики, що надають користувачам, комфортне та зручне користування інтернет-магазином, а саме:

зручність інтерфейсу, швидкість завантаження, оптимізація під мобільні пристрої, реєстрація та вхід, процес замовлення, оплата, підтримка клієнтів, відгуки клієнтів [5].

Таблиця 1.1 – Огляд існуючих рішень

Характеристики	**Staff**	**Hector**	**LC Waikiki**
Зручність інтерфейсу	Інтуїтивний інтерфейс, простий пошук та фільтри	Простий та стильний дизайн, ефективна навігація	Функціональний, зручний пошук і фільтри
Швидкість завантаження	Висока швидкість	Висока швидкість	Середня швидкість
Оптимізація під мобільні пристрої	Оптимізований	Оптимізований	Оптимізований
Реєстрація та вхід	Проста реєстрація та швидкий вхід	Проста реєстрація	Проста реєстрація і можливість входу через соцмережі
Процес замовлення	Зручний, можливість швидкого замовлення	Зручний, легкий процес оформлення	Проста і зрозуміла процедура замовлення
Підтримка клієнтів	Оперативна підтримка через різні канали	Високий рівень підтримки	Оперативна підтримка

В результаті аналізу, можна зробити висновок що Staff це хороший інтернет-магазин з інтуїтивним інтерфейсом, простим пошуком, добре оптимізованим для мобільних пристроїв з простим входом та реєстрацією,

різноманітні способи оплати та оперативну підтримку клієнтів. Щодо недоліків, вузький асортимент, орієнтований на молодіжну аудиторію.

Hector має простий та стильний дизайн з ефективною навігацією, що робить пошук товарів зручним. Сайт завантажується швидко, добре оптимізований під мобільні пристрої. З недоліків можна віднести вузьку спеціалізацію на чоловічому одязі.

LC Waikiki має функціональний сайт з потужними інструментами пошуку та фільтрації, що легко дозволяє знаходити необхідні товари. Процес реєстрації простий та великим плюсом є можливість реєстрації через соціальні мережі, що додає зручності. До недоліків можна віднести нижчу швидкість завантаження на фоні конкурентів, що може свідчити про погану оптимізацію інтернет-магазину.

1.3 Постановка завдання

Після проведення детального аналізу предметної області та огляду існуючих рішень інтернет-магазинів на ринку, можна виділити основну структуру, яку повинен містити інтернет-магазин, а саме [6]:

– головна сторінка;
– сторінки категорій;
– сторінка товару;
– сторінки асортименту;
– сторінки авторизації та реєстрації.

А щодо функціоналу сайту, повинні бути присутні базові дії:

– реєстрація та авторизація в магазині;
– можливість перегляду усіх новинок на головній сторінці;
– вибір категорій товарів;
– корзина замовлень;
– фільтр товарів;
– пошук;
– можливість зберігання, редагування та видалення товарів з корзини;

– можливість оформити доставку;
– можливість вибору зручного методу оплати.

1.4 Вибір середовища розробки

На сьогоднішній день існує безліч інтегрованих середовищ розробки (IDE), що надають програмістам широкий вибір інструментів для створення програмного забезпечення. Вибір конкретного середовища розробки базується на ряді факторів, включаючи вимоги проекту, тип задач, мову програмування, зручність використання та специфічні можливості IDE. Кожне середовище розробки пропонує унікальний набір інструментів та функцій, які можуть бути оптимально підібрані для конкретних задач.

Основні типи задач, які можуть вирішуватись за допомогою IDE, включають розробку веб-додатків, інтернет-магазинів, мобільних додатків, настільного програмного забезпечення, системного програмного забезпечення, а також аналіз даних і машинне навчання. В залежності від задач, програмісти обирають різні IDE [7].

Для веб-розробки є кілька потужних IDE, таких як: Visual Studio Code, Atom, Sublime Text, WebStorm, IntelliJ IDEA.

Visual Studio Code, розроблений Microsoft, є одним з найпопулярніших і найпотужніших IDE для веб-розробки. Це безкоштовний і відкритий редактор коду, який підтримує широкий спектр мов програмування, включаючи JavaScript, TypeScript, HTML, CSS та багато інших [8].

Одна з головних переваг VS Code – це багата екосистема розширень, що дозволяє легко налаштовувати середовище під специфічні потреби розробника. Розширення для VS Code охоплюють інструменти для налагодження, тестування, автозавершення коду, інтеграцію з системами контролю версій, такими як Git, та багато іншого. Інтерфейс користувача є зручним та інтуїтивно зрозумілим, а також підтримує режим роботи з кількома вкладками та роздільними вікнами.

Незважаючи на всі свої переваги, VS Code може вимагати значних системних ресурсів, особливо при використанні численних розширень одночасно. Це може вплинути на продуктивність, особливо на старіших або менш потужних комп'ютерах. Крім того, налаштування та оптимізація середовища можуть зайняти певний час, особливо для новачків.

На рисунку 1.4 зображено інтерфейс середовища розробки VS Code.

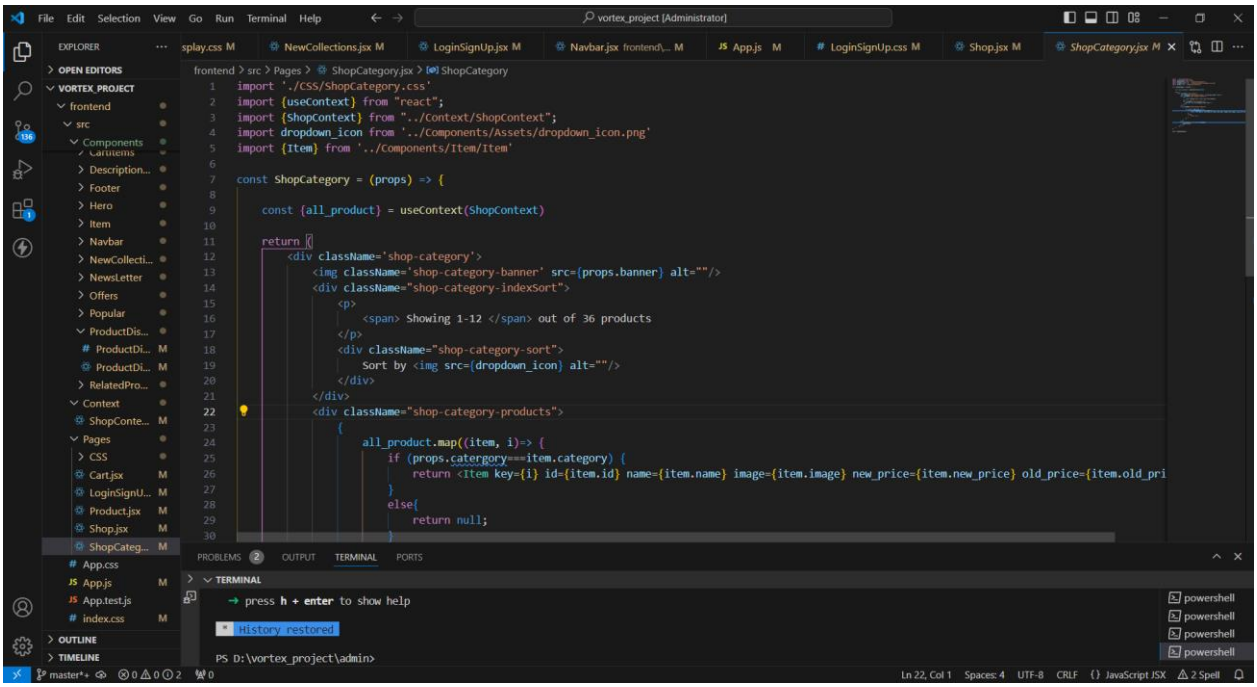

Рисунок 1.4 – Інтерфейс середовища розробки VS Code

Atom, розроблений GitHub, також є відкритим і безкоштовним редактором коду, що робить його популярним вибором серед веб-розробників. Atom підтримує безліч плагінів, які можна легко встановлювати для розширення функціональності редактора. Однією з унікальних особливостей Atom є його здатність до колаборативного редагування коду в режимі реального часу за допомогою пакета Teletype. Інтерфейс Atom є дуже настроюваним, що дозволяє розробникам створювати робоче середовище, яке відповідає їхнім індивідуальним потребам. Недоліками ж середовища розробки Atom є те, що він може бути повільнішим порівняно з іншими редакторами коду, особливо при роботі з великими файлами або проектами. Це пов'язано з тим, що Atom

побудований на Electron, що робить його менш ефективним у використанні системних ресурсів. Крім того, налаштування Atom може бути складним процесом для новачків, які тільки починають свою кар'єру у веб-розробці.

На рисунку 1.5 зображено інтерфейс середовища розробки Atom.

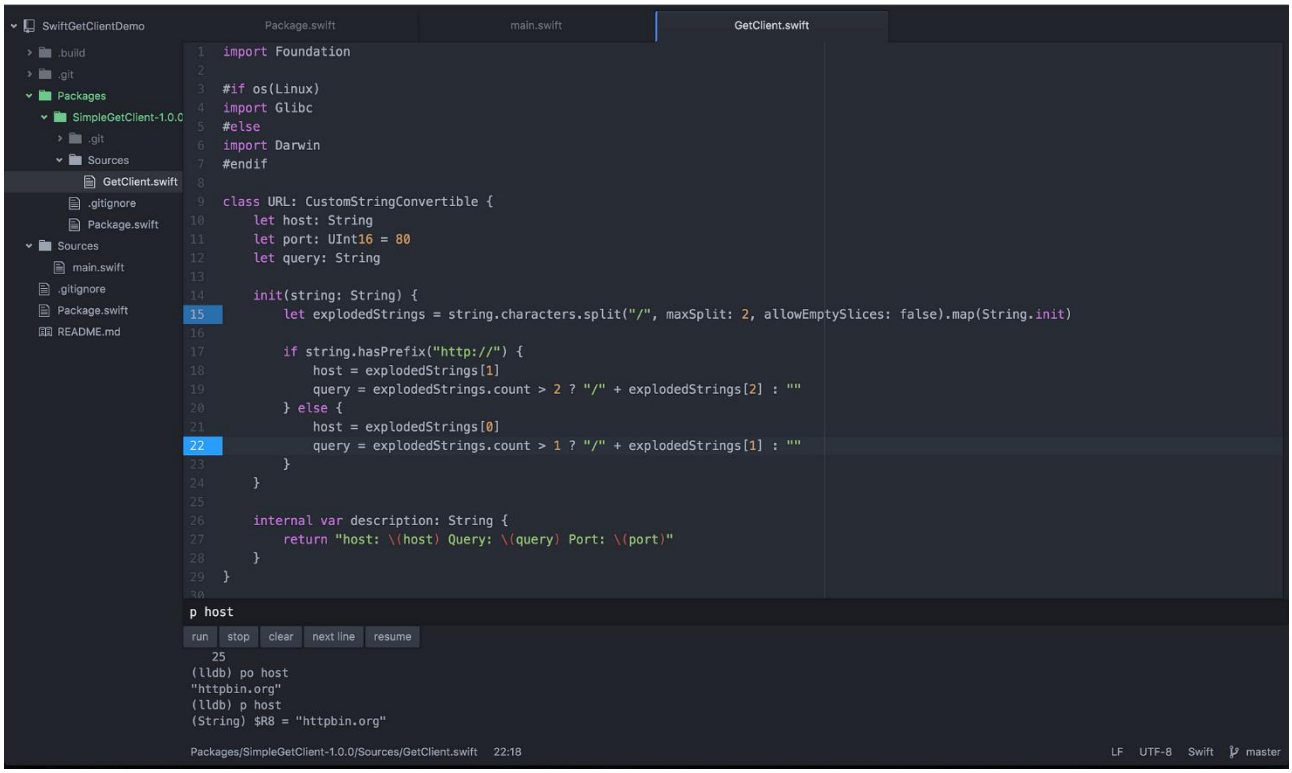

Рисунок 1.5 – Інтерфейс середовища розробки Atom

Sublime Text є одним з найшвидших та найефективніших текстових редакторів, що робить його популярним вибором серед веб-розробників. Sublime Text відомий своєю швидкістю роботи та мінімальним використанням системних ресурсів.

Редактор підтримує безліч плагінів, які можна встановити для розширення його функціональності, включаючи підтримку різних мов програмування, автозавершення коду та синтаксичний аналіз. Інтерфейс Sublime Text є чистим та зручним, що дозволяє розробникам зосередитися на коді без відволікань. Основним недоліком Sublime Text є обмежена безкоштовна версія, повна ліцензія вимагає покупки. Крім того, хоча Sublime Text підтримує плагіни,

екосистема може бути менш розвинутою порівняно з VS Code або Atom, що може обмежити можливості розширення функціональності редактора [9].

На рисунку 1.6 зображено інтерфейс середовища розробки Sublime Text.

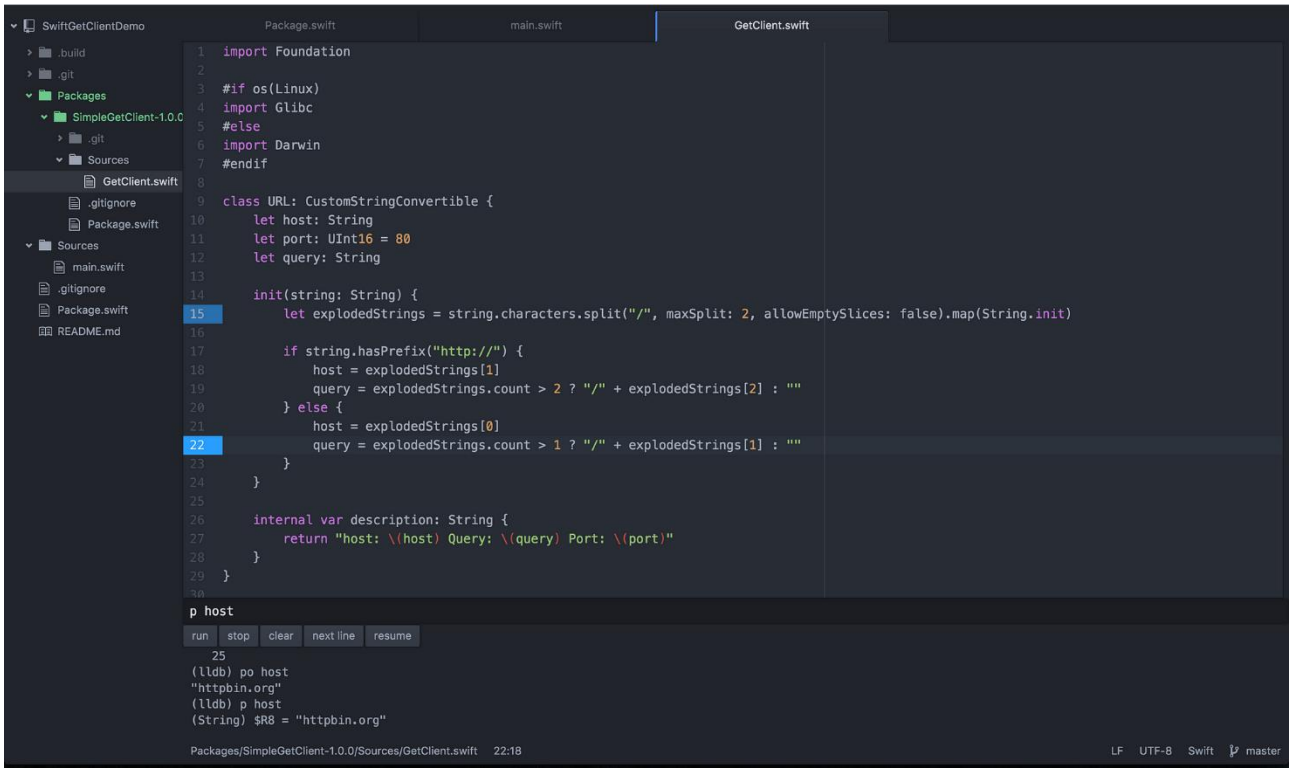

Рисунок 1.6 – Інтерфейс середовища розробки Sublime Text

WebStorm, що розроблений компанію JetBrains, є одним з найпотужніших IDE для веб-розробки, особливо для роботи з JavaScript та його фреймворками, такими як React, Angular, Vue.js та Node.js. WebStorm пропонує інтелектуальне автозавершення коду, потужний дебагінг, вбудоване тестування та підтримку контролю версій. Одна з ключових переваг WebStorm – це його глибока інтеграція з інструментами розробки, такими як Git, Docker, npm та інші, що робить його ідеальним для професійних розробників.

Основним недоліком WebStorm є його вартість. Це комерційне IDE, що вимагає покупки ліцензії, хоча доступна безкоштовна пробна версія. Крім того, WebStorm може вимагати більше системних ресурсів порівняно з легшими редакторами, що може вплинути на продуктивність на менш потужних комп'ютерах.

Intellij IDEA, також розроблений компанію JetBrains, є універсальним IDE, яке підтримує численні мови програмування, включаючи Java, JavaScript, TypeScript, HTML, CSS та інші. IntelliJ IDEA відомий своїм потужним набором інструментів для розробки. Веб-розробники цінують Intellij IDEA за зручну інтеграцію з популярними фреймворками та платформами, а також за підтримку розширень і плагінів, які можуть значно розширити функціональність IDE.

На рисунку 1.7 зображено середовище розробки Intellij IDEA.

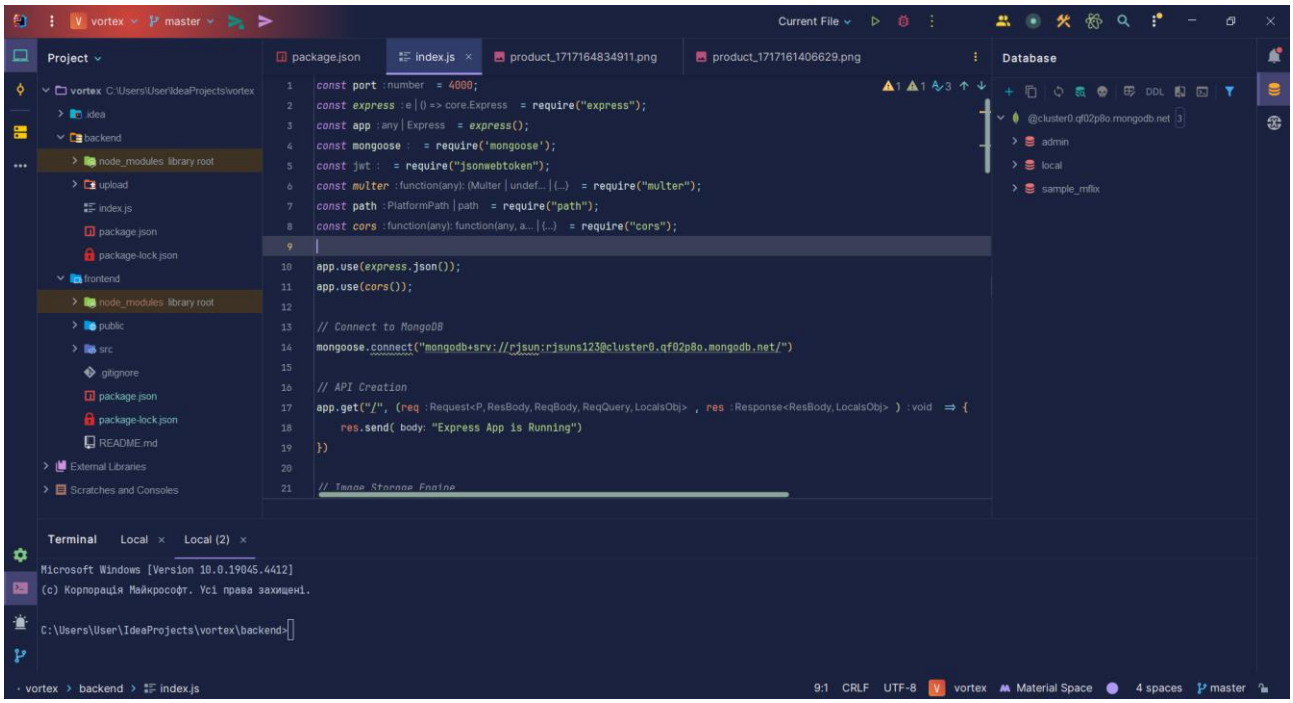

Рисунок 1.7 – Середовища розробки Intellij IDEA

Недоліками Intellij IDEA як і у WebStorm, є комерційним продуктом, що вимагає покупки ліцензії для використання повного набору функцій. Висока функціональність цього IDE також може призвести до високого використання системних ресурсів, що може вплинути на продуктивність на менш потужних комп'ютерах. Крім того, багатофункціональний інтерфейс може виявитися складним для початківців, які тільки починають працювати у веб-розробці [10].

Для даного проекту все ж таки було обрано використання середовища розробки Visual Studio Code. Такий вибір аргументовано більшою кількості плагінів та розширень за допомогою яких буде зручно та простіше налаштувати

середовище, саме під інтернет-магазин. Також одним з основних факторів такого вибору стало високе використання системних ресурсів продуктів WebStorm та Intellij IDEA, оскільки Visual Studio Code споживає набагато менше, а це є ключовим фактором швидкодії інтернет-магазину.

1.5 Висновок першого розділу

В першому розділі кваліфікаційної роботи проведено детальний аналіз предметної області, огляд існуючих рішень, постановку завдання та вибір середовища розробки для створення інтернет-магазину. Описано важливість інтернет-магазинів як ефективного інструменту для просування бізнесу та підвищення впізнаваності бренду, а також визначено ключові компоненти, необхідні для зручного користування.

Здійснено детальний аналіз існуючих рішень на ринку інтернет-магазинів на прикладі трьох магазинів: Staff, Hector та LC Waikiki. Описано їхні переваги та недоліки, що дозволило виявити найкращі практики та уникнути типових помилок при розробці власного проекту.

Визначено основні компоненти структури сайту, включаючи головну сторінку, сторінки категорій, сторінку товару, сторінки асортименту, а також сторінки авторизації та реєстрації. Описано базовий функціонал сайту, який включає реєстрацію та авторизацію користувачів, перегляд новинок, вибір категорій, корзину замовлень, фільтрацію товарів, пошук, управління товарним кошиком, оформлення доставки та вибір методів оплати.

Розглянуто різні середовища розробки, які можуть бути використані для створення інтернет-магазину. Проаналізовано переваги та недоліки середовищ Intellij IDEA, WebStorm, Sublime Text, Atom та VS Code. Обрано VS Code завдяки його великій бібліотеці плагінів та розширень, а також низькому використанню системних ресурсів.

РОЗДІЛ 2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ІНТЕРНЕТ-МАГАЗИНУ ТА БАЗИ ДАНИХ

2.1 Проектування архітектури інтернет-магазину

Одним з ключових етапів створення проекту є проектування архітектури. Воно визначає структуру, компоненти, взаємозв'язки та принципи взаємодії між елементами системи. Вдало спроектована архітектура гарантує ефективність, масштабованість, надійність і гнучкість програмного продукту. У контексті інтернет-магазинів правильний вибір архітектури є критичним для забезпечення швидкої роботи, високої доступності та здатності обробляти велику кількість користувачів і транзакцій.

Існує кілька основних типів архітектури, які застосовуються у проектуванні ПЗ і кожна з них використовується для різних вимог та задач, а саме: монолітна архітектура передбачає об'єднання всіх компонентів програмного забезпечення в один програмний зразок та усі функції, такі як обробка запитів, управління даними, автентифікація та інші, інтегровані в одну програму, мікросервісна архітектура, що розбиває програмне забезпечення на невеликі, незалежні сервіси, кожен з яких виконує окрему функцію, а сервіси взаємодіють між собою через визначені інтерфейси, такі як REST API або message queue, serverless архітектура, яка дозволяє розробникам зосередитися на написанні коду, тоді як управління інфраструктурою та автоматичне масштабування беруть на себе хмарні провайдери, проте serverless архітектура може потребувати оптимізації для забезпечення продуктивності та контролю за витратами [11].

Для інтернет-магазинів однією з найпопулярніших моделей є 3-рівнева архітектура. Дана архітектура складається з трьох основних рівнів: клієнтський рівень, що відправляє запити до серверного рівня та отримує відповіді, серверний рівень, що обробляє бізнес-логіку, обмін даними між клієнтом та

базою даних, автентифікацію користувачів та інші серверні задачі та база даних, що зберігає та керує даними, необхідними для функціонування додатка.

На рисунку 2.1 зображена 3-рівнева архітектура.

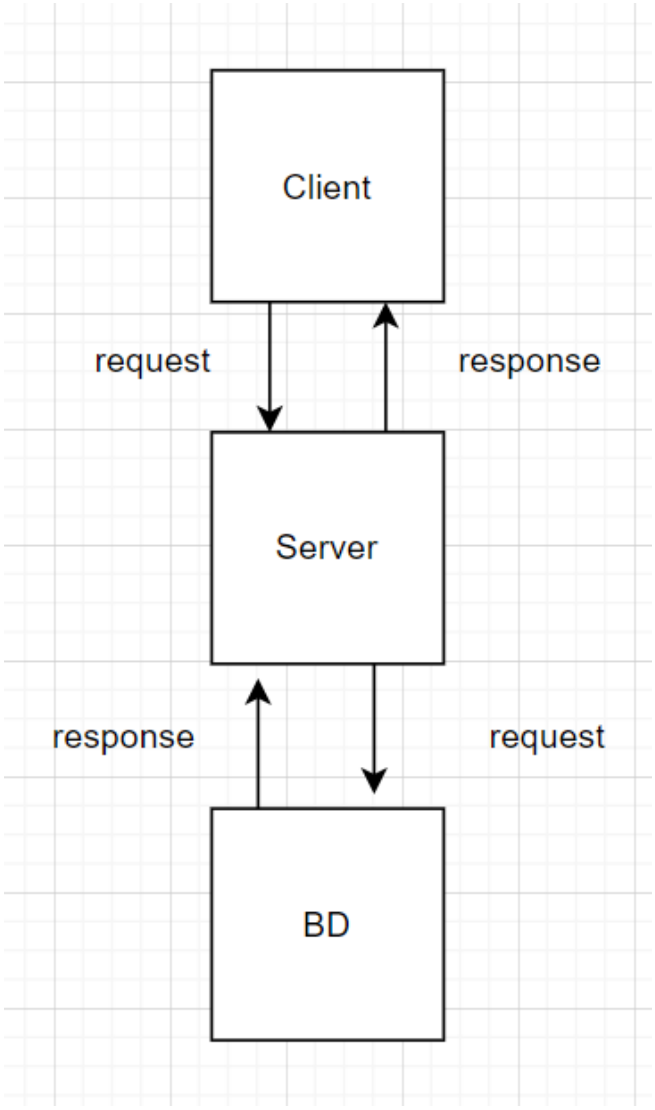

Рисунок 2.1 – Схема 3-рівневої архітектури

Переваги 3-рівневої архітектури для інтернет-магазинів включають розділення відповідальностей, що спрощує розробку та підтримку, можливість масштабувати кожен рівень незалежно, забезпечуючи ефективне використання ресурсів, гнучкість, оскільки зміни на одному рівні не впливають на інші, що спрощує впровадження нових функцій та оновлень, а також підвищення загальної безпеки системи через розділення даних та бізнес-логіки на різні рівні.

2.2 Визначення акторів та варіанти використання інтернет-магазину

Інтернет-магазин складається з кількох ключових акторів і виконує певні поставлені функції та має визначені ролі в системі. До цих учасників належать користувач, адміністратор, менеджер товарів [12].

На рисунку 2.2 зображено акторів інтернет-магазину.

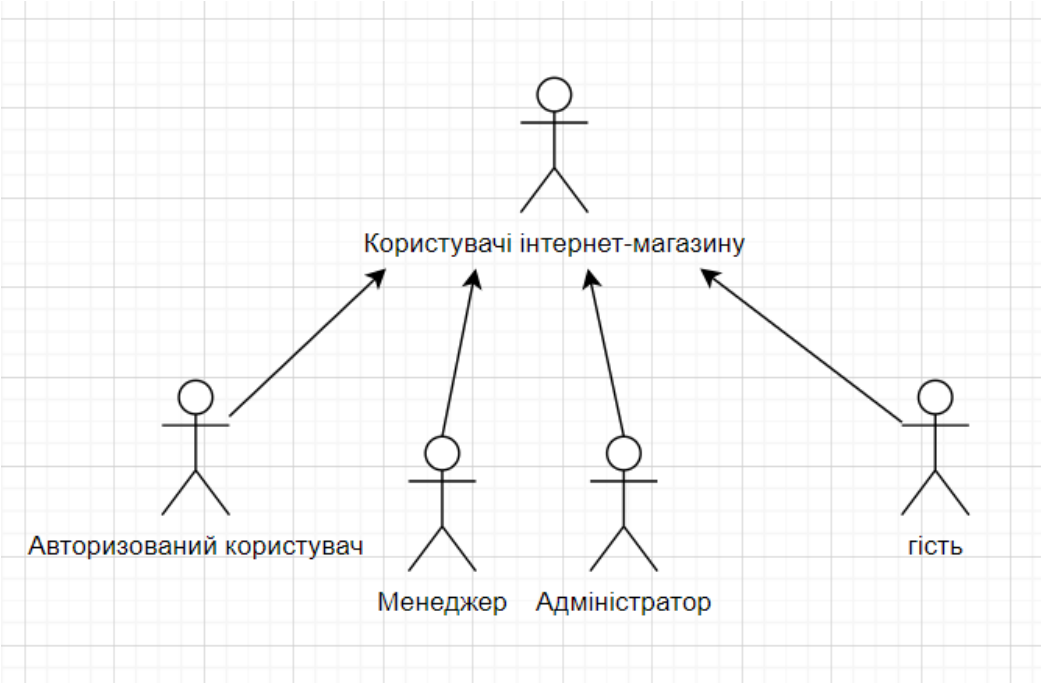

Рисунок 2.2 – Актори інтернет-магазину

Авторизовані покупці є основними користувачами інтернет-магазину. У сутності гість відсутні основні функції інтернет-магазину, все що доступно, це перегляд асортименту та пошук товарів. Адміністратори відповідають за загальне управління інтернет-магазином. Менеджери керують асортиментом товарів у магазині.

Кожен з цих акторів відіграє важливу роль у функціонуванні інтернет-магазину, забезпечуючи злагоджену роботу системи та задоволення потреб користувачів. Завдяки їхній взаємодії інтернет-магазин може ефективно обслуговувати клієнтів, підтримувати високу якість послуг та постійно вдосконалюватися.

Після створення діаграми акторів, наступним кроком є опис основних варіантів використання[13]. Опис варіантів використання актора гість наведено у таблиці 2.1.

Таблиця 2.1 – Опис варіантів використання актора гість

Актор	**Найменування**	**Формулювання**
Гість	Перегляд асортименту	Можливість переглядати доступні товари
	Реєстрація	Можливість реєстрації в інтернет-магазині
	Авторизація	Можливості використання усіх привілеїв авторизованого користувача

Опис варіантів використання актора авторизований користувач наведено у таблиці 2.2.

Таблиця 2.2 – Опис варіантів використання авторизованого користувача

Актор	**Найменування**	**Формулювання**
Авторизований користувач	Перегляд каталогу	Надання доступу до каталогу товарів.
	Дії з кошиком	Можливість додавання обраних товарів у кошик
	Оформлення замовлень	Можливість оформлення замовлення
	Здійснення оплати	Здійснення оплати за покупки

Продовження таблиці 2.2

Актор	Найменування	Формулювання
Авторизований користувач	Відстеження замовлення	Відстеження статусу замовлення та історії замовлень

Опис варіантів використання актора менеджера товарів наведено у таблиці 2.3.

Таблиця 2.3 – Опис варіантів використання менеджера товарів

Актор	Найменування	Формулювання
Менеджер товарів	Додавання нових товарів	Можливість додавання товарів до каталогу
	Оновлення інформації	Оновлення цін, опису товару
	Встановлення цін	Зміна та встановлення цін товарів

Опис варіантів використання адміністратора наведено у таблиці 2.4.

Таблиця 2.4 – Опис варіантів використання адміністратора

Актор	Найменування	Формулювання
Адміністратор	Управління користувачами	Додавання чи видалення користувачів
	Забезпечення безпеки	Здійснення моніторингу та забезпечення безпеки
	Моніторинг продуктивності	Аналіз та контроль продуктивності системи

Після загального опису варіантів використання можна зробити висновок, що гість має найменше прав, оскільки в його можливості входить лише перегляд асортименту та пошук товарів. В авторизованого користувача є більше можливостей аніж у гостя, а в менеджера є основні можливості по роботі з товарами. Найбільше привілеїв є у адміністратора, через повний доступ до інтернет-магазину [14].

Тепер можна перейти до детального опису кожного варіанту використання. На рисунку 2.3 зображено діаграму варіантів використання сутності гість.

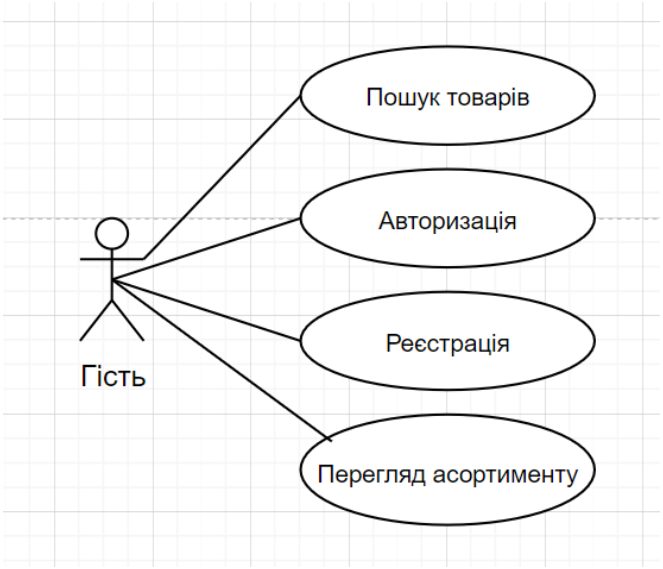

Рисунок 2.3 – Діаграма варіантів використання гостя

Гість з такими можливостями може користуватись базовими функціями реєстрації та авторизації для доступу до системи. Гість може переглядати широкий асортимент товарів, який доступний в магазині. Зручний пошук дозволить гостю швидко знайти необхідні товари. Цей функціонал дозволяє гостям магазину отримати огляд асортименту та зручно шукати товари, не обов'язково реєструючись або створюючи обліковий запис. Хоча такий доступ, не передбачає можливості здійснення покупок, але забезпечує гостям важливу інформацію для прийняття рішення про подальші дії.

В авторизованого користувача з’являється більше можливостей взаємодії з інтернет-магазином.

На рисунку 2.4 зображено діаграму використання сутності авторизований користувач.

Рисунок 2.4 – Діаграма варіантів використання авторизованого користувача

Авторизовані користувачі отримують ряд переваг, які значно полегшують та покращують процес покупок на сайті. Серед цих переваг – можливість переглядати ексклюзивний каталог товарів, доступ до якого відкривається лише зареєстрованим клієнтам. Це надає змогу вибирати з унікальних або обмежених в продажу продуктів. Крім того, авторизовані користувачі можуть додавати або видаляти товари у кошику, легко керуючи своїм замовленням та роблячи процес покупок більш організованим.

Оформлення замовлень також стає простішим, оскільки зареєстровані клієнти можуть швидко заповнювати необхідну інформацію, вказуючи адресу доставки та контактні дані, які зберігаються для майбутніх покупок. Це дозволяє уникнути повторного введення інформації при кожному новому замовленні.

Авторизовані користувачі можуть залишати відгуки на придбані товари, ділитися своїми враженнями та допомагати іншим клієнтам робити правильний

вибір. Оплата також стає зручнішою, оскільки зареєстровані користувачі можуть зберігати платіжні дані для швидкого проведення транзакцій у майбутньому.

Крім того, авторизовані користувачі можуть відстежувати статус своїх замовлень, що дозволяє завжди бути в курсі всіх етапів виконання замовлення. Також зареєстровані клієнти можуть переглядати історію попередніх покупок, що допомагає швидко знайти інформацію про попередні замовлення, повторно замовити ті ж товари або перевірити деталі минулих транзакцій.

Одним з основних рушіїв роботи інтернет-магазину є менеджер. На рисунку 2.5 зображено діаграму варіантів використання для менеджера.

Рисунок 2.5 – Діаграма варіантів використання для менеджера

Роль менеджерів магазину полягає в управлінні асортиментом товарів. Менеджери додають нові товари до каталогу, щоб клієнти могли знаходити та придбати якісні товари. Крім цього, менеджери оновлюють інформацію про наявні товари, таку як опис, характеристики та фотографії, щоб забезпечити точність та повноту інформації для клієнтів. Однією з основних функцій менеджерів є встановлення цін на товари. Крім того, менеджери можуть встановлювати акції та знижки на певні товари, щоб заохотити клієнтів до купівлі та підвищити продажі. Менеджери також контролюють запаси товарів,

задля забезпечення достатньої кількості товарів рівноцінному попиту клієнтів. Аналізуючи дані про продажі та попит, менеджери визначають, які товари потрібно замовляти та які можна вилучити з каталогу.

Але ключовим актором є адміністратор, оскільки у нього є всі можливості у керуванні інтернет-магазином. На рисунку 2.6 зображено діаграму варіантів використання для адміністратора.

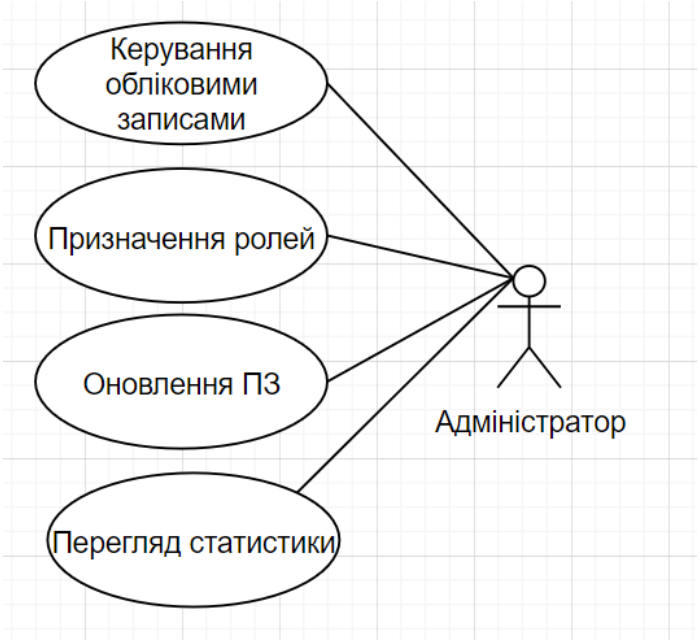

Рисунок 2.6 – Діаграма варіантів використання для адміністратора

Адміністратор інтернет-магазину має можливість управляти обліковими записами користувачів, включаючи їх створення, редагування та деактивацію за необхідності.

Адміністратор може призначати ролі користувачам на підставі їх обов’язків та зв’язків в інтернет-магазині. Аналізуючи статистику, що стосується інтернет-магазину та окремих співробітників, адміністратор може оцінити ефективність зусиль працівників та визначити загальну продуктивність.

Також адміністратор відповідає за забезпечення безпеки та конфіденційності документації, таких як паролі та електронні скриньки, а також контроль за діями інших користувачів.

Повну діаграму варіантів використання інтернет-магазину зображено на рисунку 2.7.

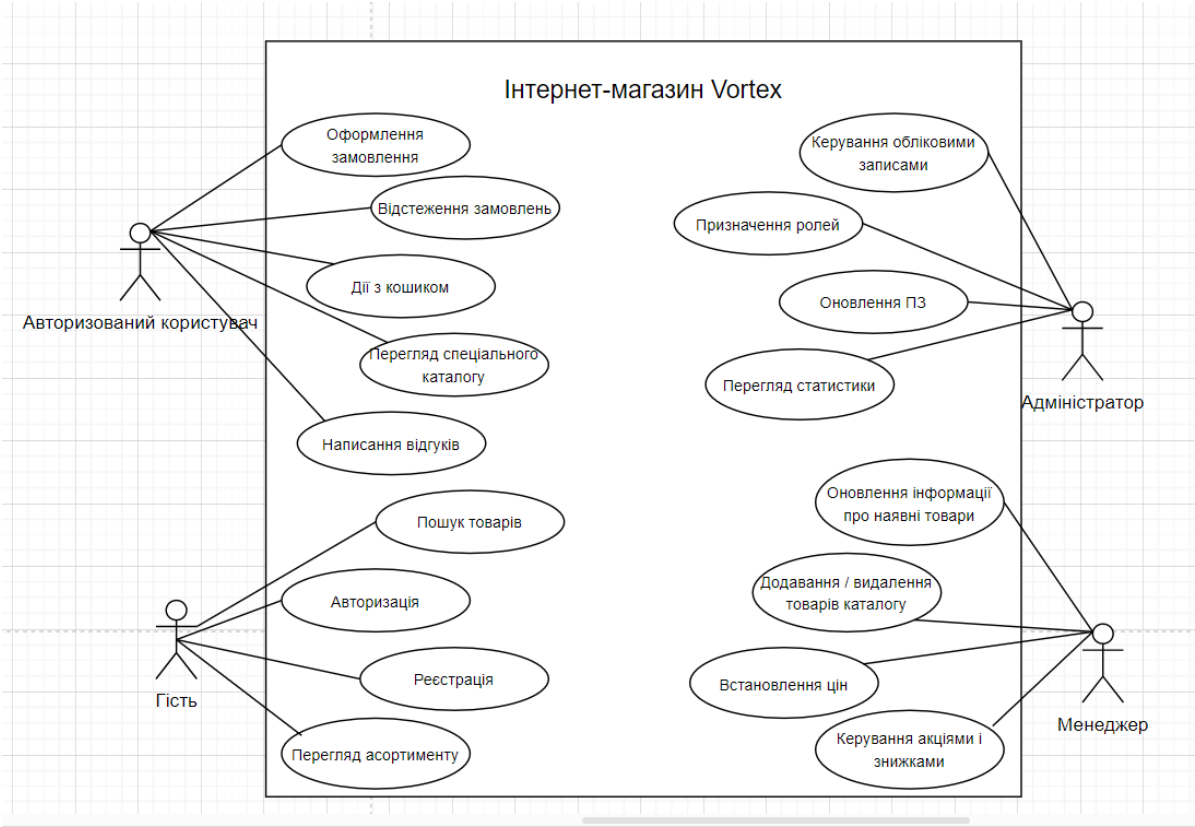

Рисунок 2.7 Діаграма варіантів використання

На діаграмі ВВ зображені актори та варіанти використання інтернет-магазину Vortex.

2.3 Проектування архітектури бази даних

У сучасному бізнесі обробка великих обсягів даних є важливою складовою. Архітектура бази даних визначає, як дані будуть зберігатися, організовані та оброблювані, що впливає на продуктивність і надійність системи [15]. Розглянемо реляційні та нереляційні бази даних та обгрунтуємо вибір MongoDB для інтернет-магазину. Реляційні бази даних (РБД) організовані у вигляді таблиць з рядками і стовпцями. РБД використовують мову запитів SQL та нормалізацію даних. Нереляційні бази даних (NoSQL) гнучкіші, мають різні типи, такі як документо-орієнтовані, колонкові, графові та ключ-значення [16].

MongoDB є документно-орієнтованою нереляційною СУБД з гнучкою моделлю даних, що підтримує горизонтальне масштабування.

Вибір MongoDB для інтернет-магазину зумовлений його гнучкістю та можливістю швидко адаптуватися до змін. MongoDB підтримує шардинг, що робить його ефективним для великих обсягів даних. Використання MongoDB гарантує ефективність роботи системи та легку масштабованість, що є ключовими факторами для успіху проекту [17].

БД буде складатись з наступних таблиць: Users, Products, Categories, Rewiews, Promocodes, Orders, OrderHistory.

Таблиця для зберігання даних про користувача складається з наступних полів: ім’я користувача, пароль, адреса електронної пошти та дані кошика користувача. Схему моделі в базі даних для сутності користувача можна побачити на рисунку 2.8.

Users	
id	objectId
name	string
email	string
password	string
cartData	array

Рисунок 2.8 – Схема таблиці для сутності користувача

В сутність користувач входять наступні поля:

– id – унікальний ідентифікатор користувача;
– name – ім’я користувача;
– email – електронна пошта;
– password – пароль;

– cartData – дані кошика користувача.

Також однією з основних таблиць інтернет-магазину, є таблиця товарів, що використовується для зберігання інформації про товари, а саме: назву, ім’я товару, зображення, до якої категорії відноситься даний продукт, ціни та дату створення товару в інтернет-магазині. Схему моделі в базі даних для сутності товар можна побачити на рисунку 2.9.

Products	
id	objectId
name	string
image	string
categoryId	objectId
new_price	number
old_price	number
createdAt	date

Рисунок 2.9 – Схема таблиці сутності товар

В сутність товарів входять наступні поля:

– id – унікальний ідентифікатор товару;

– name – назва товару;

– image – зображення товару;

– categoryId – ідентифікатор категорії товарів;

– new_price – нова ціна товару;

– old_price – стара ціна;

– createdAt – дата створення товару.

Також однією з головних таблиць є категорія товарів, що відповідає за категоризацію товарів. Схему моделі в базі даних для сутності категорія можна побачити на рисунку 2.10.

Categories	
id	objectId
name	string
parentCategory	objectId
description	string

Рисунок 2.10 – Схема таблиці сутності категорія

В сутність категорії входять наступні поля:

– id – унікальний ідентифікатор категорії;

– name – назва категорії;

– parentCategory – посилання на батьківську категорію;

– description – опис категорії.

Наступною таблицею є відгуки. Схему моделі в базі даних для сутності відгук можна побачити на рисунку 2.11.

Reviews	
id	objectId
userId	objectId
productId	objectId
rating	number
comment	string
createdAt	date

Рисунок 2.11 – Схема таблиці сутності відгуки

В сутність відгуки входять наступні поля:

– id – унікальний ідентифікатор відгуку;

– userId – ідентифікатор користувача, який залишив відгук;
– productId – ідентифікатор товару, до якого залишений відгук;
– rating – оцінка товару;
– comment – коментар користувача;
– createdAt – дата та час створення відгуку.

Наступною таблицею є сутність промокод, що використовується для зберігання інформації про промокоди, знижки та умови їх використання.

Схему моделі в базі даних для сутності промокод можна побачити на рисунку 2.12.

Promocodes	
id	objectId
code	string
discountType	string
discountAmount	number
expiryDate	date
conditions	string

Рисунок 2.12 – Схема таблиці сутності товари

В сутність промокод входять наступні поля:
– id – унікальний ідентифікатор промокоду;
– code – унікальний код промокоду;
– discountType – тип знижки;
– discountAmount – сума або відсоток знижки;
– expiryDate – термін дії промокоду;
– conditions – умови використання промокоду.

Одна з головних сутностей бази даних є таблиця замовлення, що відображає дані про замовлення користувачів. Схему моделі в базі даних для сутності замовлення можна побачити на рисунку 2.13.

Orders	
id	objectId
userId	objectId
products	array
totalAmount	number
shippingAddress	string
status	string
createdAt	date

Рисунок 2.13 – Схема таблиці сутності замовлення

В сутність замовлення входять наступні поля:

– id – унікальний ідентифікатор замовлення;

– userId – ідентифікатор користувача, що зробив замовлення;

– products – масив об'єктів з інформацією про товари в замовленні;

– totalAmount – загальна сума замовлення;

– shippingAddress – адреса доставки;

– status – статус замовлення (нове, оброблене, відправлене, доставлене);

– createdAt – дата та час створення замовлення.

Сутність що напряму пов’язана з сутність замовлення є історія замовлень, що зберігає історію замовлень для кожного користувача.

Схему моделі в базі даних для сутності історія замовлень можна побачити на рисунку 2.14.

Order History	
id	objectId
userId	objectId
orders	array

Рисунок 2.14 – Схема таблиці сутності історія замовлень

В сутність історія замовлень входять наступні поля:

– id – унікальний ідентифікатор запису історії замовлень;

– userId – унікальний ідентифікатор користувача, до якого належить історія замовлень;

– orders – масив об'єктів з інформацією про кожне замовлення.

Також не менш важливою за інші є таблиця платіжні дані, що використовується для зберігання інформації про платіжні дані користувачів. Схему моделі в базі даних для сутності платіжні дані можна побачити на рисунку 2.15.

Payment Details	
id	objectId
userId	objectId
cardNumber	string
expiryDate	date
cvv	string

Рисунок 2.15 – Схема таблиці сутності платіжні дані

В сутність платіжні дані входять наступні поля:

– id – унікальний ідентифікатор запису платіжних даних;

– userId – ідентифікатор користувача, до якого належать платіжні дані;

– cardNumber – номер кредитної картки;

– expiryDate – термін дії картки;

– cvv – cvv-код картки.

Після визначення та опису усіх сутностей системи, можна розробити повну діаграму бази даних [18]. На рисунку 2.16 зображено діаграму бази даних.

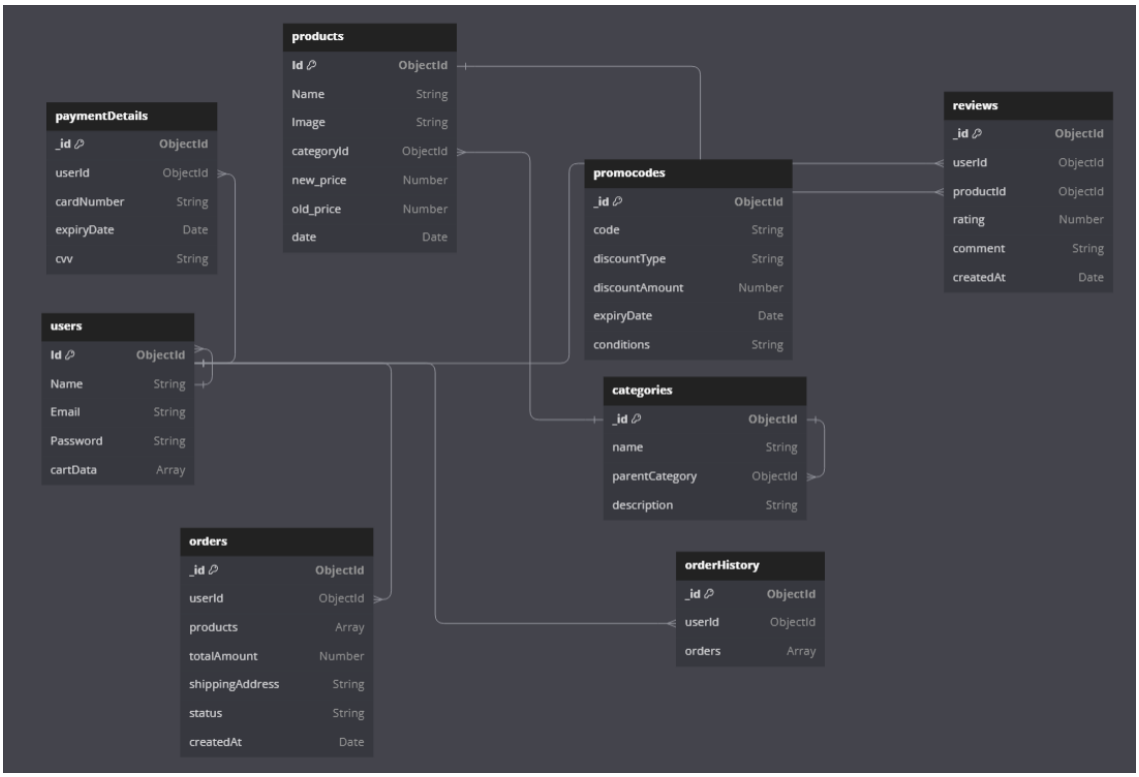

Рисунок 2.16 – Діаграма бази даних

Діаграма бази даних відображає вісім сутностей і зв'язки між ними, що допоможе краще усвідомити структуру цієї бази даних [19].

З діаграми можна побачити наступні відношення сутностей:

– Кожен користувач може мати багато відгуків, що створює відношення «один до багатьох» між користувачем і відгуками.

– Товари можуть мати багато відгуків, тому мають відношення «один до багатьох» з відгуками.

– Оскільки категорії можуть містити багато товарів, вони також мають відношення «один до багатьох» з товарами.

– Користувачі можуть здійснювати багато замовлень, тому вони мають відношення «один до багатьох» з замовленнями.

– Кожен користувач має один набір платіжних даних, що встановлює відношення «один до одного» між користувачем і платіжними даними.

– Також кожен користувач має лише одну історію замовлень, утворюючи відношення «один до одного» з історією замовлень.

– Замовлення можуть бути частиною історії замовлень, що створює відношення «один до багатьох» між замовленням і історією замовлень.

2.4 Висновок другого розділу

У другому розділі було виконано детальне проектування інтернет-магазину, зосереджене на виборі 3-рівневої архітектури, яка забезпечує гнучкість, масштабованість та легкість у підтримці системи. Було визначено ключових акторів: авторизований користувач, гість, менеджер та адміністратор, для кожного з яких розроблено та описано варіанти використання, що охоплюють усі необхідні функціональні можливості. Додатково, була складена діаграма варіантів використання, що візуально демонструє взаємодії користувачів із системою та полегшує розуміння процесів та взаємозв’язків.

Окрім цього, значна увага була приділена проектуванню бази даних. Було визначено основні сутності, їх поля та зв'язки між таблицями, що дозволяє забезпечити цілісність даних та ефективний доступ до них. Діаграма бд, що була створена під час дослідження сутностей, відображає всі необхідні відношення між таблицями, закладаючи основу для надійної та функціональної системи управління даними.

РОЗДІЛ 3. РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ VORTEX

3.1 Розробка серверної частини на базі Node.js

Вибір правильної технології є ключовим фактором успіху будь-якого продукту. Він суттєво впливає на ефективність, продуктивність та масштабованість. Для створення серверної частини було вирішено використовувати Node.js, технологію, що базується на мові програмування JavaScript і дозволяє виконувати її код поза межами браузера [20].

JavaScript – одна з найпоширеніших мов програмування, яка в основному застосовується для створення веб-сайтів, зокрема інтернет-магазинів. Серед переваг JavaScript – широкий вибір бібліотек і фреймворків, які додають додаткові функції та оптимізують процес розробки, роблячи її потужним інструментом для створення інтернет-магазинів [21].

Node.js – це серверне середовище виконання JavaScript, яке працює асинхронно і використовує механізм JavaScript V8. Воно відоме своєю неблокуючою моделлю вводу-виводу, що забезпечує ефективність при розробці серверних додатків. Крім того, Node.js має велику кількість модулів і пакетів, які можна легко інтегрувати завдяки Node Package Manager (NPM), що робить розробку швидшою і ефективнішою [22].

Для початку роботи над проектом, потрібно створити теку і відкрити її у терміналі. Потім запускаємо команду 'npm init' у командному рядку, щоб почати ініціалізацію проекту. Ця команда створить файл 'package.json' у теці, який буде містити інформацію про проект та його залежності. Далі, ми встановлюємо необхідні залежності, такі як Express і MongoDB. На рисунку 3.1 зображено процес встановлення залежностей Express та mongoose для роботи з базою даних [23].

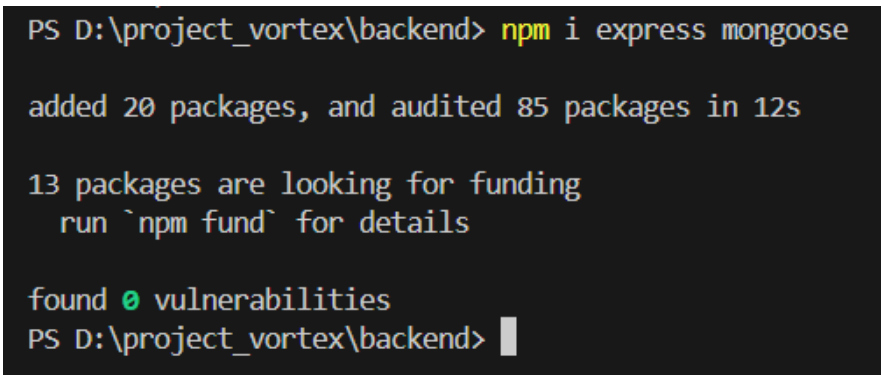

Рисунок 3.1 – Встановлення залежностей Express та mongoose

Командою «npm install express mongoose» у терміналі встановлюються необхідні залежності для проекту [24].

Для налаштування сервера, у кореневій теці проекту створюється файл «index.js». У цьому файлі відбувається підключення необхідних модулів та налаштування базового сервера Express. Потім використовуються схеми для опису типів та запитів. Після цього встановлюється порт 4000, це порт де буде виконуватись запуск сервер. Далі потрібно запустити сервер ввівши команду «node . \index.js» після чого у терміналі можна побачити повідомлення про успішний запуск сервера. На рисунку 3.2 зображено повідомлення у терміналі, що підтверджує успішний запуск сервера.

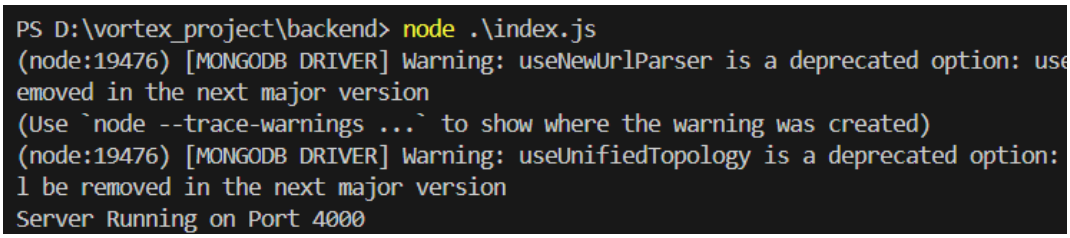

Рисунок 3.2 – Запуск сервера у терміналі

Наступним що треба зробити після запуску сервера, розробити основну логіку та виконати підключення до бази даних.

Для належного функціонування сервера необхідно налаштувати зв'язок з базою даних, що у цьому випадку є MongoDB. Підключення до бази відбувається на етапі ініціалізації сервера в файлі 'index.js', де також обробляються можливі

помилки. Після успішного підключення, наступним, до чого потрібно перейти, це створення моделей даних, які описують структуру та зв'язки між даними. У цих схемах визначається структура та типи даних, які будуть зберігатися у MongoDB.

Для ефективної розробки серверної частини проекту, була створена структура файлів і каталогів. Кожен компонент цієї структури призначений для конкретного функціоналу та відіграє важливу роль у роботі системи. На рисунку 3.3 зображено каталог серверної частини інтернет-магазину Vortex.

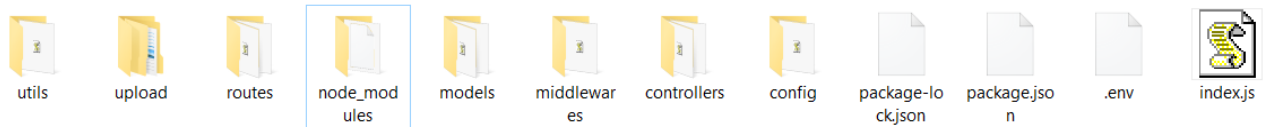

Рисунок 3.3 – Каталог серверної частини інтернет-магазину

Тепер можна перейти до повного огляду каталогу та файлів.

Каталог config містить конфігураційні файли проекту, а саме файл db.js що містить логіку підключень до бази даних MongoDB. Також файл db.js використовується для встановлення з’єднання з базою даних і обробки помилок підключення.

Каталог controllers містить контролери, які обробляють логіку запитів та відповідають за взаємодію між моделями та маршрутами.

Назви файлів та опис каталогу controllers наведено в таблиці 3.1.

Таблиця 3.1 – Опис файлів каталогу controllers

Назва файлу	**Опис файлу**
productController.js	Містить функції для обробки запитів, пов’язаних з товарами.
userController.js	Містить функції для обробки запитів, пов’язаних з реєстрацією.

Продовження таблиці 3.1

Назва файлу	Опис файлу
cartController.js	Містить функції для обробки запитів, пов’язаних з кошиком.

Каталог middlewares містить мідлвари – функції, які виконуються перед основними обробниками запитів. Каталог містить файл authMiddleware.js що перевіряє наявність та валідність JWT токену у запиті.

Каталог models містить моделі Mongoose для взаємодії з MongoDB. Назви файлів та опис каталогу models наведено в таблиці 3.2.

Таблиця 3.2 – Опис файлів каталогу models

Назва файлу	Опис файлу
productModel.js	Містить схему товарів, визначає структуру документа в колекції товарів MongoDB.
userModel.js	Містить схему користувача, визначає структуру документа в колекції користувачів MongoDB.

Каталог routes містить маршрути, які визначають, які контролери повинні обробляти певні HTTP-запити. Назви файлів та опис каталогу routes наведено в таблиці 3.3.

Таблиця 3.3 – Опис файлів каталогу routes

Назва файлу	Опис файлу
productRoutes.js	Містить маршрути, пов’язані з товарами.

Продовження таблиці 3.3

Назва файлу	Опис файлу
cartRoutes.js	Містить маршрути дій з кошиком
userRoutes.js	Містить маршрути, пов’язані з користувачами

Каталог utils містить утиліти – допоміжні функції та конфігурації. Назви файлів та опис каталогу routes наведено в таблиці 3.4.

Таблиця 3.4 – Опис файлів каталогу utils

Назва файлу	Опис файлу
storage.js	Містить конфігурацію для зберігання зображено за допомогою multer, визначає куди зберігати файли та як їх іменувати.

.env – файл конфігурації для змінних середовища. Містить конфіденційні дані та налаштування, такі як порт сервера та URL підключення до MongoDB.

index.js – головний файл для запуску сервера. Відповідає за налаштування сервера та підключення до бази даних. Імпортує необхідні модулі та середовища, налаштовує Express додаток, підключає мідлвари та маршрути, встановлює з’єднання з MongoDB за допомогою конфігурації з файлу db.js, налаштовує статичну папку для зображень, запускає сервер на зазначеному порту [25].

3.2 Розробка клієнтської частини з використанням React.js

Після завершення серверної частини проекту, наступним кроком є розробка клієнтської частини за допомогою бібліотеки React [26]. Бібліотека JavaScript, відома як React, використовується для розробки інтерфейсів користувача. Вона славиться своїм декларативним підходом до створення

компонентів інтерфейсу та користується значною популярністю серед веб-розробників. Ця бібліотека базується на архітектурі, де компоненти інтерфейсу користувача створюються блоками, які можна легко комбінувати разом.

React також використовує віртуальний DOM для ефективного оновлення лише тих частин інтерфейсу, які потребують змін. Це підвищує продуктивність та взаємодію з користувачем, та робить React популярним серед веб-розробників [27].

Для створення нового проекту з використанням React.js спершу потрібно відкрити термінал у теці, де планується розміщення проекту. Потім виконується команда «npx create-react-app .», щоб ініціалізувати всі необхідні файли для початку роботи проекту з бібліотекою React.

Після успішного виконання цієї команди, у вказаній теці буде створено базову структуру проекту, готову для подальшого розвитку та розширення. На рисунку 3.3 продемонстровано процес створення React проекту.

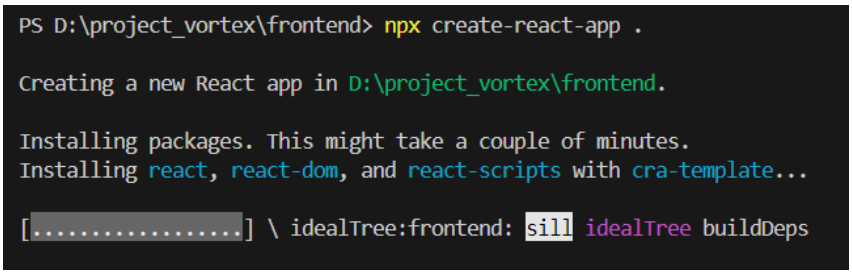

Рисунок 3.3 – Створення React проекту

Далі для розробки клієнта, у цьому проекті необхідно встановити декілька бібліотек, що полегшують роботу з роутингом, керуванням станом, анімаціями та іншими аспектами. На рисунку 3.4 демонструється процес встановлення однієї з ключових бібліотек, яка відповідає за роутінг - «react-router-dom».

Ця бібліотека дозволяє легко налаштовувати маршрути та навігацію у веб-додатках React, для завантаження бібліотеки застосовується команда 'npm i react-router-dom' [28].

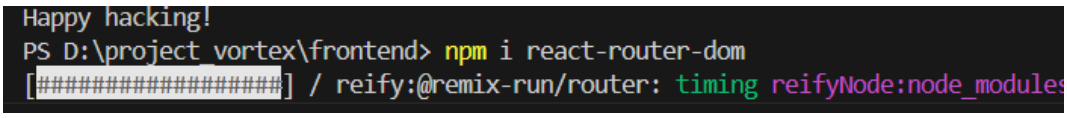

Рисунок 3.4 – Процес інсталяції бібліотеки «react-router-dom»

Починаючи роботу над клієнтською частиною, першим кроком є налаштування взаємодії з сервером. Для цього створюються сервісні файли, які керують різними типами запитів до сервера, такими як GET, POST, PUT, DELETE.

Потім використовується бібліотека Axios для виконання цих запитів, і базові URL-адреси налаштовуються в сервісах для звернень до сервера з клієнтської частини. Після запуску проекту результати цих запитів будуть відображатися у консолі браузера [29].

Наступним етапом розробки є створення компонентів з яких і буде складатись інтернет-магазин. На початку створюється файл 'App.js' в якому оголошується функція App(), що буде відмальовувати навігаційне меню з посиланнями на певні сторінки та підвал інтернет-магазину. На рисунку 3.5 зображено навігаційне меню інтернет-магазину Vortex.

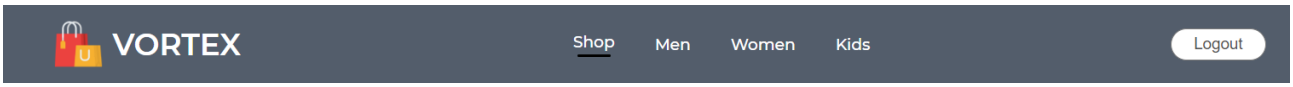

Рисунок 3.5 – Навігаційне меню інтернет-магазину Vortex

Далі потрібно зробити головну сторінку інтернет-магазину, що буде містити інформацію про магазин.

Було вирішено розмістити на головній сторінці таку інформацію, як банер з можливістю перейти на розділ останньої колекції.

На рисунку 3.6 зображено банер переходу до розділу останньої колекції.

Рисунок 3.6 – Банер головної сторінки

Далі було прийнято рішення розмістити на головній сторінці інформацію про найбільш популярні товари серед жінок, що значно підвищує зручність використання інтернет-магазину та допомагає користувачам швидко знайти бажані товари. Для кожного товару в цьому розділі передбачена можливість перейти на сторінку з детальною інформацією про обраний продукт.

На рисунку 3.7 зображено приклад інформації з цього розділу, що показує популярні товари серед жінок.

Рисунок 3.7 – Розділ популярні товари серед жінок

Далі, для залучення ще більшої уваги користувачів, на головній сторінці інтернет-магазину було розміщено банер ексклюзивних пропозицій. Цей банер має можливість перенести користувача на відповідну сторінку, де можна

придбати унікальні товари, що не доступні в інших магазинах. На рисунку 3.8 зображено банер ексклюзивних пропозицій.

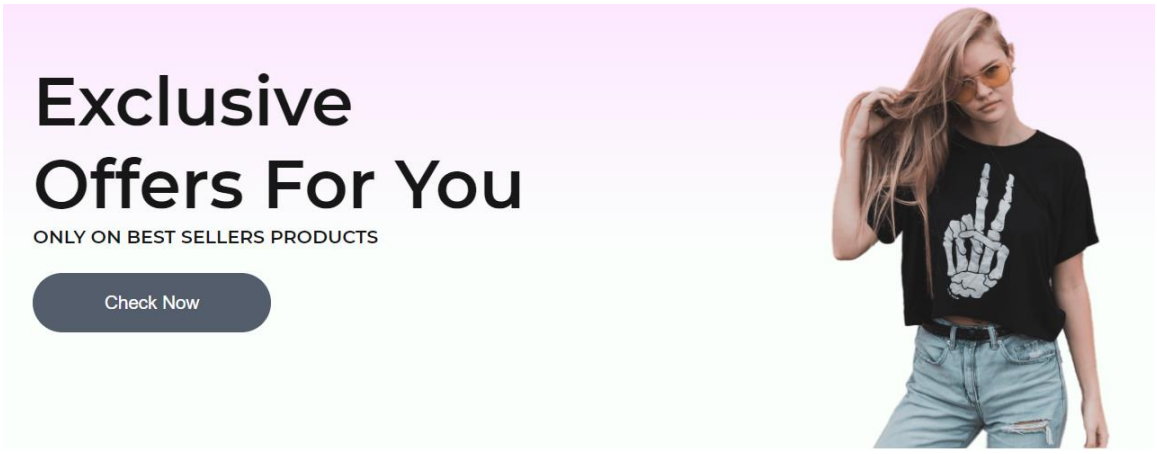

Рисунок 3.8 – Банер ексклюзивних пропозицій

Далі було створено розділ з інформацією про нову колекцію, де можна переглянути новинки, що з’явились в інтернет-магазині Vortex та можливістю перейти на вподобаний товар. На рисунку 3.9 зображено розділ з інформацією про нову колекцію.

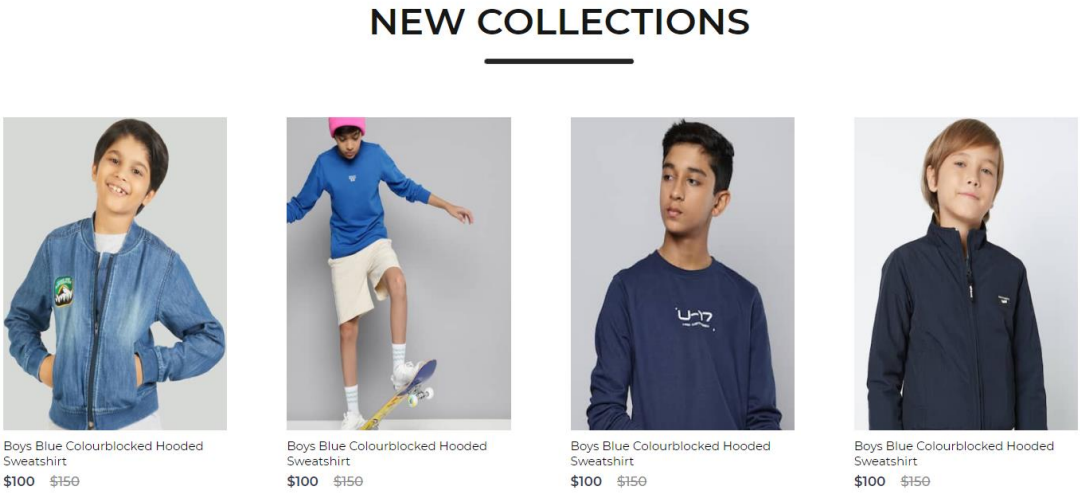

Рисунок 3.9 – Розділ з інформацією про нову колекцію

Далі було створено спеціальну форму у якій користувач може вказати електронну пошту. Завдяки цій формі користувачі можуть легко підписатися на

розсилку та отримувати актуальну інформацію про новинки, акції та спеціальні пропозиції, що сприяє підвищенню зацікавленості та лояльності клієнтів.

На рисунку 3.10 зображено форму для отримання розсилки інформації.

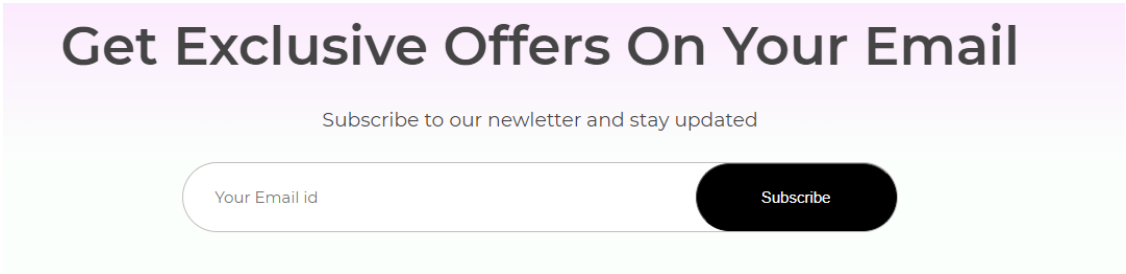

Рисунок 3.10 – Форма для отримання розсилки інформації

Останнім що потрібно додати на головну сторінку є футер інтернет-магазину, в якому міститься інформація про компанію, офіси, контакти та посилання на соціальні мережі. На рисунку 3.11 зображено футер інтернет-магазину Vortex.

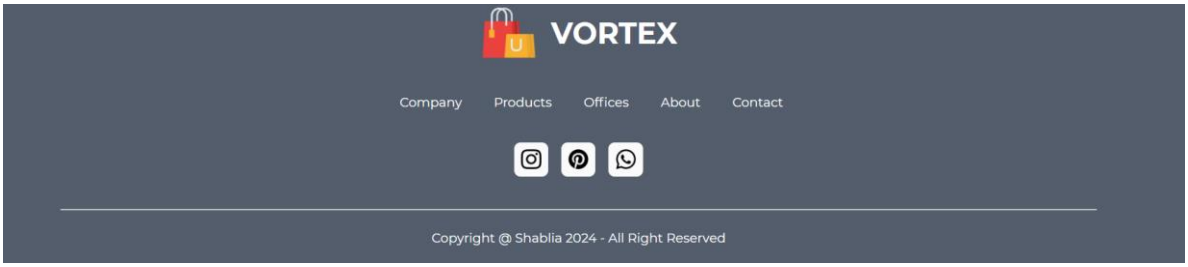

Рисунок 3.11 – Футер інтернет-магазину

Далі було створено відповідні сторінки категорій, а саме категорії для чоловіків, жінок та дітей. На кожній сторінці категорій міститься інформаційний банер, каталог товарів, що містить зображення товару, ціну та назву.

Також на сторінках категорій є можливість сортування товарів за ціною, назвою та популярністю.

На рисунку 3.12 зображено сторінку категорії для чоловіків.

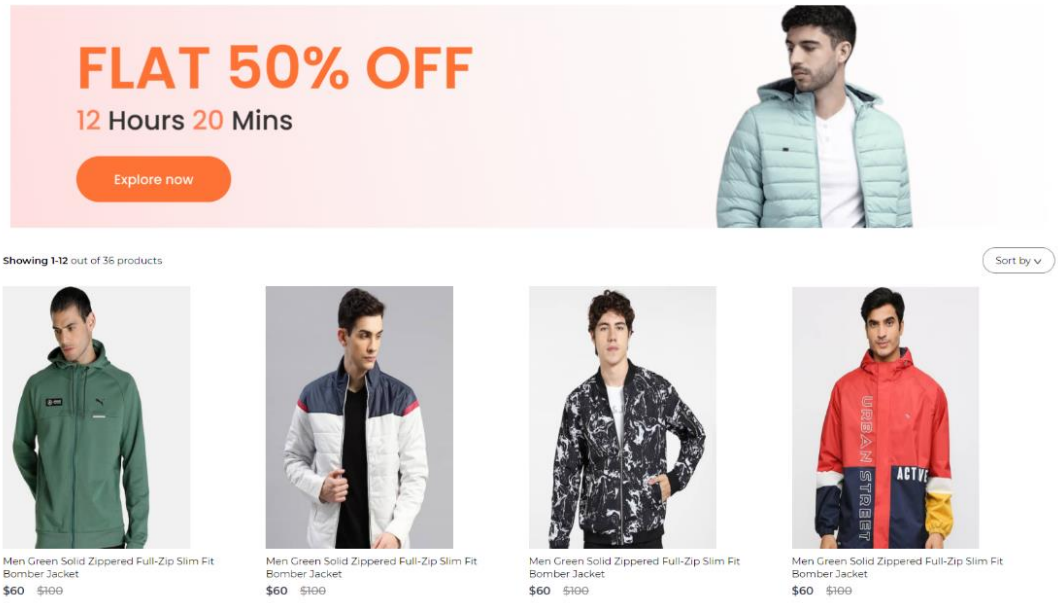

Рисунок 3.12 – Сторінка категорії для чоловіків

Наступним кроком було створено сторінку для детального огляду товару, що забезпечує користувачам повну інформацію про кожен товар та сприяє прийняттю рішення щодо покупки.

На цій сторінці було розміщено назву товару, рейтинг у вигляді зірочок, що демонструє середню оцінку товару на основі відгуків інших покупців, стару і нову ціну товару, що дозволяє побачити знижку, короткий опис товару з основною інформацією про його особливості та переваги, галерею детальних фотографій з різних ракурсів, опцію вибору розмірів від S до XXL для зручного підбору, кнопку «Додати до кошика» для швидкого додавання товару до замовлення, а також категорію товару для полегшення навігації по сайту.

Нижче на сторінці розміщено секцію з відгуками користувачів, де покупці можуть залишати свої коментарі та оцінки товару, що надає додаткову інформацію про реальний досвід використання продукту, а також зверху розміщено коротку навігацію по сайту [30].

На рисунку 3.13 зображено сторінку детального огляду товару.

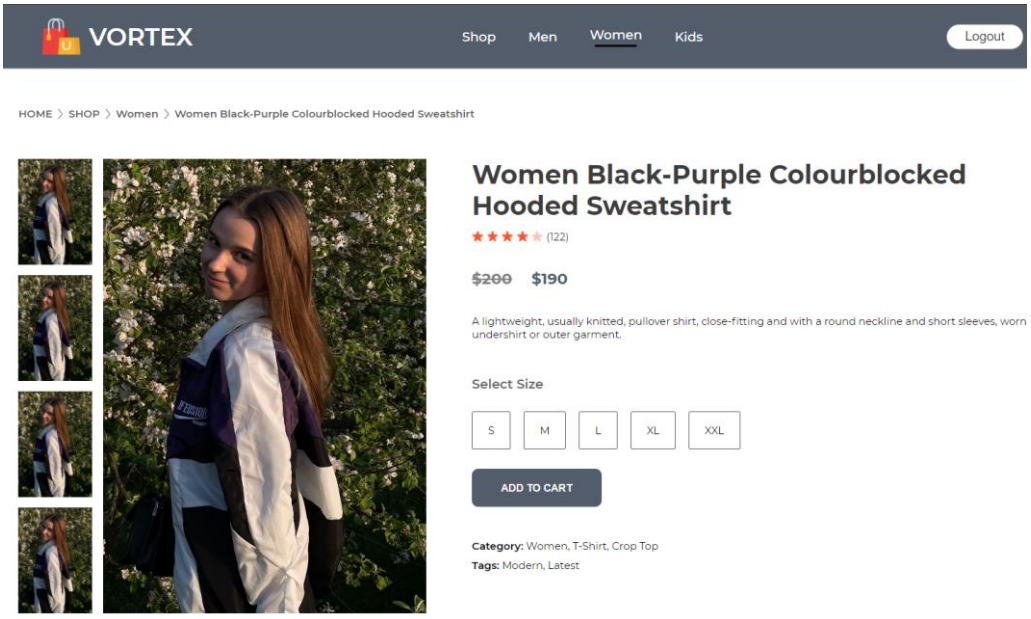

Рисунок 3.13 – Сторінка про товар

Далі, для того щоб отримати можливість замовити товар, гостю потрібно пройти форму реєстрації. У формі реєстрації гостю необхідно ввести ім'я, електронну пошту та пароль. Після успішного заповнення форми реєстрації, користувач отримує доступ до кошика та може продовжити процес оформлення замовлення. На рисунку 3.14 зображено сторінку реєстрації.

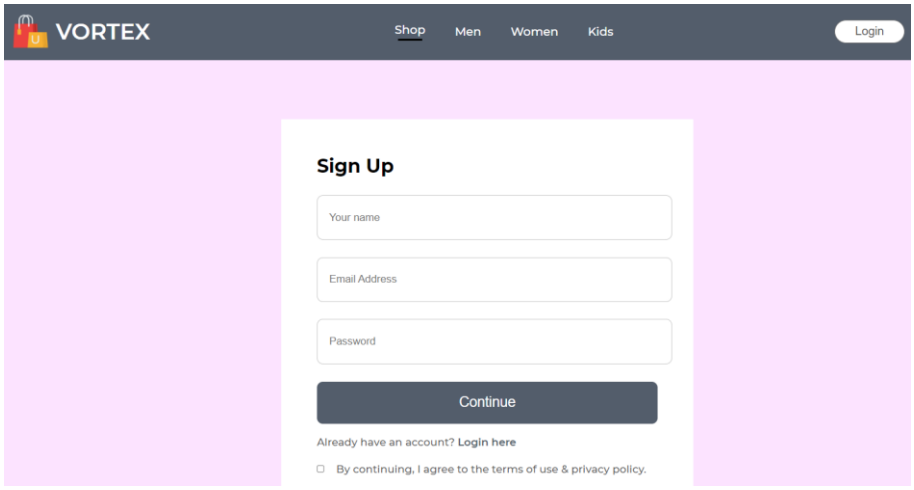

Рисунок 3.14 – Сторінка реєстрації

Послідовно, після створення сторінки реєстрації, було створено сторінку авторизації, де користувачу потрібно ввести електронну пошту та пароль, після

чого користувач зможе увійти у свій обліковий запис, обрати вподобаний товар та здійснити замовлення, використовуючи кошик. На рисунку 3.15 зображено сторінку авторизації.

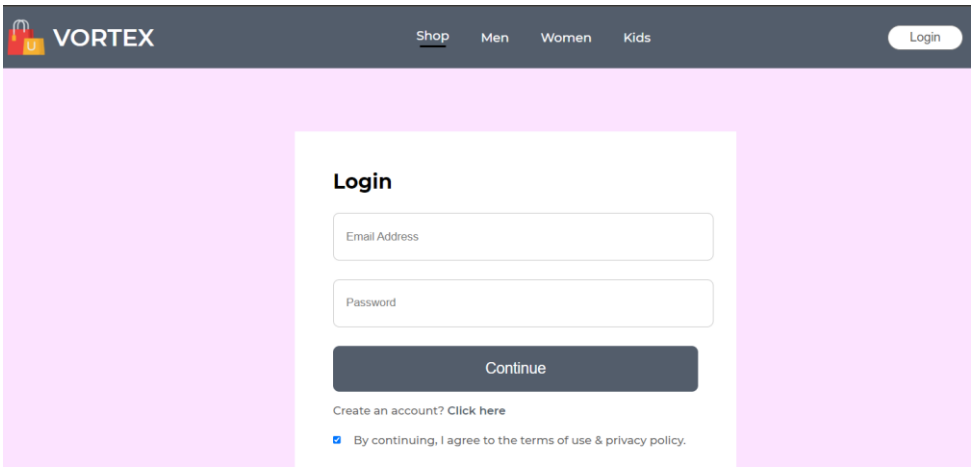

Рисунок 3.15 – Сторінка авторизації

Після авторизації у користувача з'являється зображення кошика в навігаційному меню. Це зображення кошика забезпечує зручний доступ до всіх доданих товарів, дозволяючи користувачам швидко переглядати вміст кошика незалежно від того, на якій сторінці сайту вони знаходяться. На рисунку 3.16 продемонстровано зображення кошика.

Рисунок 3.16 – Додавання до навігаційного меню зображення кошика

Після того, як користувач натисне на зображення кошика, його перенаправить на сторінку корзини. Тому наступним кроком є створення сторінки корзини.

На рисунку 3.17 зображено сторінку корзини.

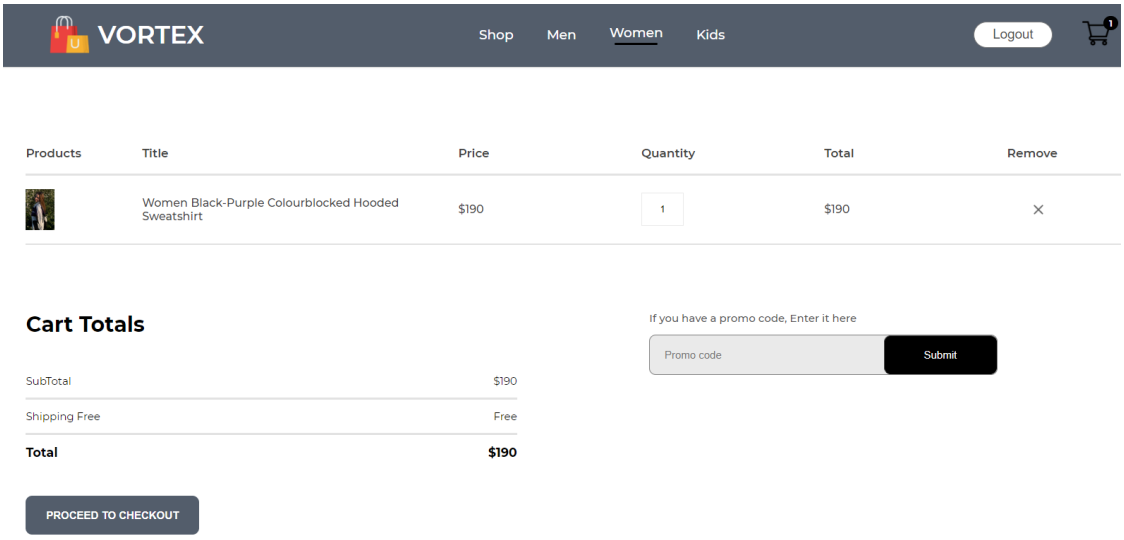

Рисунок 3.17 – Сторінка корзини

Було створено блок, де відображаються всі товари, що були додані до кошика, з усіма необхідними характеристиками та кнопка видалення товару з кошика. Нижче розташовано блок з інформацією про кошик користувача, де вказується загальна сума, а також окремо зазначена ціна доставки. Також у цьому блоці розміщена кнопка оформлення замовлення.

Крім того, було створено форму для введення промокоду, що дозволить користувачам отримати знижку на товар.

Для розробки клієнтської частини була створена структура файлів і каталогів. Кожен елемент цієї структури відповідає певному функціоналу та відіграє важливу роль у роботі проекту. На рисунку 3.18 зображено каталог клієнтської частини інтернет-магазину Vortex.

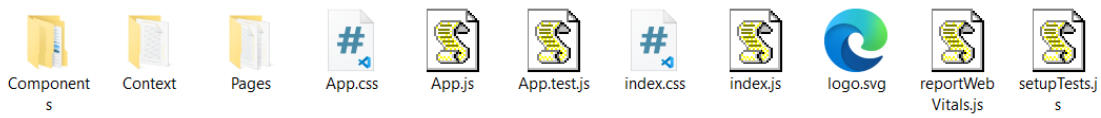

Рисунок 3.18 – Каталог клієнтської частини

Тепер можна перейти до повного огляду каталогу та файлів, що розміщені в клієнтській частині інтернет-магазину Vortex.

Каталог Components містить всі React компоненти, що використовуються в інтернет-магазині. Далі наведено компоненти, що знаходяться в каталозі Components:

– BreadCrumbs.jsx – відображає шлях до поточної сторінки, допомагаючи користувачам зрозуміти її контекст у ієрархії веб-сайту;

– CartItems.jsx – відображає вміст кошика покупок користувача, дозволяючи йому переглядати, редагувати та видаляти товари перед оформленням замовлення;

– DescriptionBox.jsx – демонструє опис продукту або іншої інформації на сторінці, допомагаючи клієнтам зрозуміти його характеристики;

– Footer.jsx – відповідає за відображення футера інтернет-магазину;

– Hero.jsx – відповідає за відображення головного банеру;

– Item.jsx – представляє окремий товар на сторінці, зображуючи його зображення, назву та ціни;

– Navbar.jsx – містить навігаційний блок, допомагаючи користувачам переміщатися по веб-сайту, а також кнопку кошика та кнопку входу;

– NewCollections.jsx – показує нові колекції товарів на сторінці, допомагаючи користувачам ознайомитися з останніми надходженнями;

– NewsLetter.jsx – надає можливість користувачам підписатися на розсилку новин та спеціальні пропозиції за допомогою електронної пошти;

– Offers.jsx – відображає ексклюзивні пропозиції на сторінці, привертаючи увагу користувачів до акційних товарів;

– Popular.jsx – показує популярні товари серед жінок, допомагаючи користувачам вибрати популярні продукти;

– ProductDisplay.jsx – відображає деталі конкретного продукту, включаючи його зображення, опис, ціни та інші характеристики;

– RelatedProducts.jsx – відображає пов'язані продукти на сторінці, допомагаючи користувачам знайти схожі товари.

Каталог Context забезпечує централізоване керування станом та логікою інтернет-магазину. Функції що використовуються в ShopContextProvider:

– getDefaultCart() – функція, яка ініціалізує початковий стан кошика з пустими значеннями;

– useEffect() – використовується для завантаження даних про всі продукти;

– addToCart() – функція, яка додає товар до кошика та оновлює стан кошика, як в локальному сховищі, так і на сервері, якщо користувач є авторизованим;

– removeFromCart() – функція, яка видаляє товар із кошика;

– getTotalCartAmount() – функція, яка обчислює загальну суму товарів у кошику;

– getTotalCartItems() – функція, яка обчислює загальну кількість товарів у кошику.

Каталог Pages містить сторінки інтернет-магазину, такі як сторінку товарів, сторінки авторизації та реєстрації, головну сторінку та сторінки категорій.

Файл App.js є головним компонентом інтернет-магазину і відповідає за роутинг та розміщення основних компонент на сторінці.

3.3 Висновок третього розділу

У третьому розділі було описано розробку серверної та клієнтської частини інтернет-магазину. Для серверної частини було використано технологію Node.js, такий вибір був зроблений через високу продуктивність та широку екосистему модулів. За допомогою Express та MongoDB, було налаштовано сервер, підключено до бази даних та реалізовано логіку обробки запитів.

Після завершення серверної частини, розпочалась розробка клієнтської частини, де було використано технологію React.js. Були створені компоненти, з яких складається інтернет-магазин та описані всі функції, для швидкої та комфортної роботи інтернет-магазину.

Також були описані структури каталогів та файлів клієнтської і серверної частини інтернет-магазину Vortex.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Долікарська допомога при ураженні електричним струмом

Долікарська допомога – це сукупність медичних заходів, спрямованих на надання допомоги при невідкладних станах, які виникають на роботі, у повсякденному житті, під час дорожньо-транспортних пригод, катастроф, техногенних аварій, а також при гострих неврологічних, терапевтичних, хірургічних та критичних станах. Відсутність долікарської допомоги у разі нещасних випадків або раптових гострих захворювань може призвести до тяжких наслідків, включаючи летальні. Своєчасне надання допомоги відіграє важливу роль у подальшому лікуванні постраждалих і хворих, сприяючи скороченню строків їх медичної та трудової реабілітації [34].

Менеджери та адміністратори онлайн-магазинів постійно працюють за персональними комп'ютерами або ноутбуками. Їхня діяльність включає взаємодію з різними електронними приладами та обладнанням, що необхідне для роботи інтернет-магазину. Незважаючи на сучасні засоби захисту, існує ймовірність ураження електричним струмом через несправність обладнання, неправильне використання приладів або технічні збої. Тому важливо знати, як надати долікарську допомогу при ураженні електричним струмом, щоб забезпечити безпеку працівників та мінімізувати наслідки таких випадків. Усвідомлення ризиків та вміння правильно реагувати на них є ключовими аспектами підтримання здоров'я та безпеки на робочому місці.

Ураження електричним струмом виникає при контакті людини з електричним ланцюгом, що викликає протікання струму через тіло. Електричний струм може бути різної сили та напруги, і навіть короткочасний контакт з високовольтним струмом може призвести до серйозних наслідків або смерті. Загалом електротравма становить 1-2 % усіх видів травм. Близько 80 % ураження електричним струмом виникає при контакті зі струмом низької напруги (до 1000 В), до 20 % - струмом високої напруги. У першому випадку летальність

становить 3 %, при контакті зі струмом високої напруги – до 30 %. Побутовий струм у 50 Гц особливо небезпечний для нормального функціонування міокарду [35].

Електротравми можна розділити на легкі, середні та тяжкі форми. При легких ураженнях постраждалий може втратити свідомість. У випадку середніх уражень спостерігаються загальні м'язові судоми, непритомність, розлади дихання та серцевої діяльності. Тяжкі ураження призводять до такого пригнічення дихання та серцевої діяльності, що звичайні методи реанімації не є ефективними, і постраждалий перебуває в стані клінічної смерті.

Ознаки ураження електричним струмом включають втрату свідомості у 70% випадків, яка може тривати від кількох хвилин до години, а іноді й більше доби. Ураження головного мозку проявляються клонічними та тонічними судомами, головним болем, сонливістю та ретроградною амнезією. Рідше виникають локальні ураження головного та спинного мозку. Ці симптоми часто з'являються, коли струм проходить через голову або верхні кінцівки. Якщо струм впливає на головний мозок або серце, наслідки можуть бути смертельними – виникає клінічна смерть, де у 80% випадків спостерігається фібриляція шлуночків, а у 20% – асистолія [36].

Порядок надання першої медичної допомоги при ураженні електричним струмом:

– Перш за все, необхідно забезпечити безпеку як рятувальника, так і потерпілого. Вимкнути джерело живлення або використати ізольовані інструменти, щоб усунути контакт постраждалого з електричним струмом.

– Після того, як потерпілий опиниться в безпечному середовищі, слід провести первинну оцінку його стану. Перевірити свідомість, дихання та наявність пульсу. Якщо потерпілий не реагує, не дихає або у нього відсутній пульс, слід негайно розпочати серцево-легеневу реанімацію (СЛР).

– При проведенні СЛР, необхідно переконатися, що дихальні шляхи відкриті: для цього нахилити голову потерпілого назад та підняти підборіддя. Виконувати штучне дихання та компресії грудної клітки.

– Під час проведення СЛР обов'язково викликати екстрену медичну допомогу або попросити когось із присутніх зробити це. Прибуття кваліфікованих медиків є критичним для подальшого обстеження та лікування постраждалого.

– Очікуючи на прибуття екстреної медичної допомоги, важливо постійно контролювати стан потерпілого та надавати йому емоційну підтримку. Якщо потерпілий починає приходити до тями, необхідно допомогти йому зайняти зручне положення та підтримувати його морально.

Надання долікарської допомоги при ураженні електричним струмом є важливим елементом забезпечення безпеки на робочому місці, особливо в умовах роботи з електронними приладами. Володіння навичками першої допомоги може суттєво знизити ризик серйозних ускладнень та врятувати життя.

4.2 Вимоги ергономіки до організації робочого місця оператора ПК

Проектування та розробка інтернет-магазину вимагає постійної роботи за персональним комп’ютером, тому важливо правильно організувати робоче місце оператора ПК. Ергономічно оптимізоване робоче місце сприяє здоров’ю, комфорту та продуктивності тих, хто проводить багато часу за комп’ютером.

Робочі місця з комп’ютерами повинні відповідати вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце під час виконання робіт сидячи» [37] та ДСанПІН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електроннообчислювальних машин» [38].

Основні положення ДСТУ 8604:2015:

– Конструкція робочого місця та розташування всіх його елементів (сидіння, органи керування, засоби відображення інформації тощо) повинні відповідати антропометричним, фізіологічним і психологічним вимогам, а також характеру роботи.

– Конструкція робочого місця має забезпечувати виконання трудових операцій у межах досяжності моторного поля.

– Висота сидіння та підставки для ніг (якщо висота робочої поверхні не регулюється) повинні бути встановлені за номограмою для людей зі зростом 1800 мм. Для менших працівників висоту сидіння та підставки для ніг потрібно збільшувати на різницю між висотою робочої поверхні для зросту 1800 мм і висотою, оптимальною для зросту конкретного працівника.

– Мінімальна товщина стільниці залежить від міцності матеріалу та технічних вимог і не повинна перевищувати 30 мм.

– Підставка для ніг має бути регульованою по висоті, шириною не менше 300 мм і довжиною не менше 400 мм, поверхня має бути рифленою. По передньому краю слід передбачити бортик висотою 10 мм.

Основні положення ДСанПІН 3.3.2.007-98:

– Площа на одне робоче місце повинна бути не менше 6,0 м^2, а об'єм - не менше 20,0 м^3.

– Приміщення повинні мати природне та штучне освітлення.

– Природне освітлення має здійснюватись через вікна, орієнтовані переважно на північ чи північний схід і забезпечувати коефіцієнт природної освітленості (КПО) не нижче 1,5%.

– Для внутрішнього оздоблення приміщень слід використовувати дифузновідбивні матеріали з коефіцієнтами відбиття для стелі 0,7-0,8, для стін 0,5-0,6.

– Яскравість світильників загального освітлення в зоні кутів випромінювання від 50 до 90 градусів з вертикаллю має бути не більше 200 кд/м^2, а захисний кут світильників - не менше 40 градусів.

– Показник засліплення при використанні джерел загального штучного освітлення не повинен перевищувати 20.

– Необхідно обмежувати нерівномірність розподілу яскравості в полі зору працюючих з екранами. Співвідношення яскравості робочих поверхонь не

повинно перевищувати 3:1, а співвідношення яскравості робочих поверхонь і поверхонь стін, обладнання – 5:1.

4.3 Висновок четвертого розділу

У четвертому розділі кваліфікаційної роботи детально розглянуто питання безпеки життєдіяльності та основ охорони праці в контексті ураження електричним струмом та вимог ергономіки до організації робочого місця оператора ПК. Аналізуючи відповідність вимогам стандартів безпеки, виділено ключові аспекти, які включають надання долікарської допомоги при ураженні електричним струмом та вимоги ергономіки до організації робочого місця оператора ПК.

В контексті ураження електричним струмом було проаналізовано його можливі наслідки, зокрема різні форми ураження, ознаки та порядок надання першої медичної допомоги. Зазначено, що своєчасне та правильне реагування на такі ситуації є важливим для забезпечення безпеки працівників та мінімізації можливих наслідків.

Висвітлено важливість оптимізації робочого місця операторів ПК з точки зору ергономіки. Розглянуто ключові вимоги до організації робочих місць відповідно до стандартів ДСТУ 8604:2015 та ДСанПІН 3.3.2.007-98. Зокрема дані вимоги стосуються конструкції робочого місця, що включають в собі ергономічні вимоги до крісла, підставки для ніг, стільниці, освітленості та площі робочого простору. Вказані вимоги дозволять підвищити продуктивність праці працівників та забезпечити комфортні умови роботи.

Передбаченні в роботі ергономічні вимоги сприятимуть зниженню ризику травм та захворювань, пов'язаних з професійною діяльністю.

ВИСНОВКИ

У першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр» проведено детальний аналіз предметної області, що дозволило визначити ключові аспекти та вимоги до розроблюваного інтернет-магазину. Під час аналізу існуючих рішень було виявлено недоліки та можливості для подальшого вдосконалення інтернет-магазину серед конкурентів. Висвітлено основні завдання проекту, включаючи визначення його функціональності та ключові вимоги до реалізації. Проаналізовано можливі варіанти середовища розробки з урахуванням придатності та відповідності поставленим завданням.

У другому розділі кваліфікаційної роботи досліджено можливі варіанти архітектури продукту, що дозволило визначити оптимальну модель його побудови. Обґрунтовано вибір акторів та сценаріїв використання продукту, що забезпечило зручність та ефективність використання. Сформовано концепцію архітектури бази даних, що відповідає потребам та вимогам розроблюваного інтернет-магазину.

У третьому розділі кваліфікаційної роботи розроблено та реалізовано серверну частину інтернет-магазину на базі Node.js, що забезпечило його стабільну та ефективну роботу. Запропоновано та реалізовано клієнтську частину з використанням React.js, що забезпечило зручний та зрозумілий інтерфейс для простого користувача.

У розділі «Безпека життєдіяльності, основи охорони праці» висвітлено важливі аспекти забезпечення безпеки під час експлуатації інтернет-магазину, включаючи процедури долікарської допомоги при ураженні електричного струму та вимог ергономіки до організації робочого місця оператора ПК.

Таким чином, досліджено, розроблено та реалізовано інтернет-магазин, який відповідає сучасним вимогам ефективності, зручності використання та безпеці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Чому створення інтернет-магазину це один з кращих методів розширення бізнесу. [Електронний ресурс] – Режим доступу до ресурсу: https://lb.ua/tech/2023/10/23/579979_chomu_stvorennya_internetmagazinu_tse.html

2. Які сторінки мають бути на сайті інтернет-магазину ? [Електронний ресурс] – Режим доступу до ресурсу: https://www.fishdigital.agency/blog-yaki-storinki-mayut-buti-na-sayti-internet-magazinu.

3. Інтернет-магазин Staff. [Електронний ресурс] – Режим доступу до ресурсу: https://www.staff-clothes.com/

4. Інтернет-магазин Hector. [Електронний ресурс] – Режим доступу до ресурсу: https://hector.ua/

5. Інтернет-магазин LC Waikiki. [Електронний ресурс] – Режим доступу до ресурсу: https://www.lcwaikiki.ua/uk-UA/UA

6. Структура інтернет-магазину : [Електронний ресурс] – Режим доступу до ресурсу: https://wezom.com.ua/ua/blog/struktura-internet-magazina

7. Найкращі IDE та текстові редактори. [Електронний ресурс] – Режим доступу до ресурсу: https://javarush.com/ua/groups/posts/uk.177.naykrajsh-ide-ta-tekstov-redaktori-dlja-frontendnika

8. Documentation for Visual Studio Code. [Електронний ресурс] – Режим доступу до ресурсу: https://code.visualstudio.com/docs

9. Documentation for Sublime Text. [Електронний ресурс] – Режим доступу до ресурсу: https://www.sublimetext.com/docs/

10. Documentation for Intellij Idea. [Електронний ресурс] – Режим доступу до ресурсу: https://www.jetbrains.com/help/idea/getting-started.html

11. Проектування архітектури ПЗ. [Електронний ресурс] – Режим доступу до ресурсу: https://javarush.com/ua/groups/posts/uk.2519.chastina-2-pogovorimo-trokhi-pro-arkhtekturu-pz

12. Key Actors in an Ecommerce Website's. [Електронний ресурс] – Режим доступу до ресурсу: https://www.nextwebi.com/blog/use-case-diagrams-for-ecommerce-websites-in-2023

13. Use Case Scenarios for Administrators in Ecommerce Websites. [Електронний ресурс] – Режим доступу до ресурсу: https://itsourcecode.com/uml/e-commerce-website-use-case-diagram-uml/

14. Діаграма варіантів використання. [Електронний ресурс] – Режим доступу до ресурсу: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/

15. Архітектура бази даних. [Електронний ресурс] – Режим доступу до ресурсу: https://medium.com/tag/database-design

16. Реляційні та нереляційні бази даних. [Електронний ресурс] – Режим доступу до ресурсу: https://alternativescience.net/programming/242-sql-chy-nosql-os-v-chomu-pytannya/

17. What is MongoDB? [Електронний ресурс] – Режим доступу до ресурсу: https://www.mongodb.com/company/what-is-mongodb

18. Schema Design for MongoDB. [Електронний ресурс] – Режим доступу до ресурсу: https://blog.det.life/the-inheritance-schema-design-pattern-for-mongodb-data-modelling-03540a484a93

19. Relation in ER Diagram. [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/dbms-er-model-concept

20. Introduction to Node.js. [Електронний ресурс] – Режим доступу до ресурсу: https://nodejs.org/en/learn/getting-started/introduction-to-nodejs

21. What is JavaScript. [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

22. Introduction to NPM. [Електронний ресурс] – Режим доступу до ресурсу: https://nodejs.org/en/learn/getting-started/an-introduction-to-the-npm-package-manager

23. Основи роботи з фреймворком Express.js. [Електронний ресурс] – Режим доступу до ресурсу: https://foxminded.ua/express-js/

24. How works with Mongoose. [Електронний ресурс] – Режим доступу до ресурсу: https://docs.nestjs.com/recipes/mongodb

25. Middleware in Express. [Електронний ресурс] – Режим доступу до ресурсу: https://www.geeksforgeeks.org/middleware-in-express-js/

26. Intro to React. [Електронний ресурс] – Режим доступу до ресурсу: https://legacy.reactjs.org/tutorial/tutorial.html

27. DOM. [Електронний ресурс] – Режим доступу до ресурсу: https://uk.legacy.reactjs.org/docs/react-dom.html

28. React Router DOM. [Електронний ресурс] – Режим доступу до ресурсу: https://www.npmjs.com/package/react-router-dom

29. What is Axios? [Електронний ресурс] – Режим доступу до ресурсу: https://axios-http.com/docs/intro

30. React Components. [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/react/react_components.asp

31. Additive mathematical model of gas consumption process / Iaroslav Lytvynenko, Serhii Lupenko, Oleh Nazarevych, Hryhorii Shymchuk, Volodymyr Hotovych // Scientific Journal of TNTU. – Tern. : TNTU, 2021. – Vol 104. – No 4. – P. 87–97.

32. Approach to gas consumption process forecasting on the basis of a mathematical model in the form of a random cyclic process / Serhii Lupenko, Iaroslav Lytvynenko, Oleg Nazarevych, Grigorii Shymchuk, Volodymyr Hotovych // ICAAEIT 2021, 15-17 December 2021. – Tern. : TNTU, Zhytomyr «Publishing house „Book-Druk“» LLC, 2021. – P. 213–219. – (Mathematical modeling in power engineering and information technologies).

33. Hromyak R., Nemish V. (2023). Estimation of the structural p parameter for a number of structural materials. Scientific Journal of TNTU (Tern.), vol 112, no 4, pp. 67-72.

34. Перша долікарська допомога [Електронний ресурс] – Режим доступу до ресурсу: https://www.pharmencyclopedia.com.ua/article/790/persha-dolikarskadopomoga

35. Діагностика та лікування уражень електричним струмом [Електронний ресурс] – Режим доступу до ресурсу: https://pidru4niki.com/76919/meditsina/diagnostika_likuvannya_urazhennya_elektrichnim_strumom

36. Жидецький В. Ц., Джигирей В. С., Мельников О. В. Основи охорони праці. 2- ге вид. Львів: Афіша, 2000. 348 с.

37. ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце під час виконання робіт сидячи» [Електронний ресурс] – Режим доступу до ресурсу: https://zakon.rada.gov.ua/rada/show/va042282-99#Text

38. ДСанПІН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електроннообчислювальних машин» [Електронний ресурс] – Режим доступу до ресурсу: https://zakon.rada.gov.ua/rada/show/v0007282-98#Text

ДОДАТКИ

Додаток А

Програмний код JS для запуску сервера «index.js»

```
require('dotenv').config();
const express = require("express");
const cors = require("cors");
const connectDB = require('./config/db');
const productRoutes = require('./routes/productRoutes');
const userRoutes = require('./routes/userRoutes');
const cartRoutes = require('./routes/cartRoutes');
const upload = require('./utils/storage');

const app = express();
const port = process.env.PORT || 4000;

app.use(express.json());
app.use(cors());

connectDB();

app.get("/", (req, res) => {
    res.send("Express App is Running");
});

app.use('/images', express.static('upload/images'));
app.post("/upload", upload.single('product'), (req, res) => {
    res.json({
        success: 1,
        image_url:
`http://localhost:${port}/images/${req.file.filename}`
    });
});

app.use('/api/products', productRoutes);
app.use('/api/users', userRoutes);
app.use('/api/cart', cartRoutes);

app.listen(port, (error) => {
    if (!error) {
        console.log("Server Running on Port " + port);
    } else {
        console.log("Error: " + error);
    }
});
```

Додаток Б

Програмний код JS для побудови сторінок та навігації «App.js»

```
import './App.css';
import { Navbar } from "./Components/Navbar/Navbar";
import { BrowserRouter, Route, Routes } from "react-router-dom";
import { Shop } from "./Pages/Shop";
import { ShopCategory } from "./Pages/ShopCategory";
import { Product } from "./Pages/Product";
import { LoginSignUp } from "./Pages/LoginSignUp";
import { Cart } from "./Pages/Cart";
import { Footer } from "./Components/Footer/Footer";
import men_banner from './Components/Assets/banner_mens.png'
import women_banner from './Components/Assets/banner_women.png'
import kid_banner from './Components/Assets/banner_kids.png'

function App() {
    return (
        <div>
            <BrowserRouter>
                <Navbar />
                <Routes>
                    <Route path='/' element={<Shop />} />
                    <Route path='/mens' element={<ShopCategory
banner={men_banner} catergory="men" />} />
                    <Route path='/womens' element={<ShopCategory
banner={women_banner} catergory="women" />} />
                    <Route path='/kids' element={<ShopCategory
banner={kid_banner} catergory="kid" />} />
                    <Route path='product' element={<Product/>}>
                   <Route path=':productId' element={<Product/>}/>
                     </Route>
                    <Route path='/cart' element={<Cart/>} />
                   <Route path='/login' element={<LoginSignUp/>}/>
                </Routes>
                <Footer />
            </BrowserRouter>
        </div>
    );
}

export default App;
```