

# ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

## «Розробка веб-орієнтованого додатку з продажу туристичного спорядження»

Студента 4 курсу, 6 групи,  
спеціальності 121 «Інженерія  
програмного забезпечення»  
освітньої програми «Інженерія  
програмного забезпечення»

\_\_\_\_\_

підпис студента

Марценюк Олексій  
Володимирович

Науковий керівник  
Кандидат технічних наук,  
Доцент

\_\_\_\_\_

підпис керівника

Харченко Олександр  
Анатолійович

Гарант освітньої програми  
кандидат технічних наук,  
доцент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

\_\_\_\_\_

підпис гаранта

Рзаєва Світлана  
Леонідівна

# Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

**Затверджую**

Зав. кафедри інженерії програмного  
забезпечення та кібербезпеки

Криворучко О. В.

«14» листопада 2022 р.

## Завдання

### на випускний кваліфікаційний проєкт студентів

Марценюка Олексія Володимировича

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Веб-орієнтований додаток з продажу туристичного спорядження»

Затверджена наказом ректора від «15» лютого 2023 р. № 464

2. Строк здачі студентом закінченого проєкту 5 червня 2023

3. Цільова установка та вихідні дані до проєкту

Мета проєкту розробити веб-орієнтований додаток з продажу туристичного спорядження

Об'єкт дослідження веб-орієнтований додаток з продажу туристичного спорядження

Предмет дослідження розробка веб-додатку з продажу туристичного спорядження

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ

1.1. Загальні відомості з предметної галузі

1.2. Аналіз методів реалізації

1.3. Призначення розробки та галузь застосування

1.4. Мета створення

1.5. Вимоги до системи

1.5.1. Вимоги до інформаційної безпеки

1.5.2. Вимоги до параметрів технічних засобів

1.6. Висновок до розділу

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА ВИБІР ПРОГРАМНИХ ЗАСОБІВ

2.1. Архітектура додатку

2.2. Функціональне призначення програми

2.3. Опис використаних технологій та програмних засобів

2.4. Опис структури системи

2.5. Висновок до розділу

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ «EXPLOREZONE»

3.1. Опис реалізації сайту

3.2. Опис функціональної частини сайту

3.3 Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

## 6. Календарний план виконання проекту

№ пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів проекту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проекту</i>	21.09.2022	21.09.2022
2.	<i>Розробка та затвердження завдання на проєкт</i>	14.11.2022	14.11.2022
3.	<i>Вступ та перелік літературних джерел</i>	23.12.2022	23.12.2022
4.	<i>Розділ 1. Аналіз предметної області та постановка задачі</i>	27.01.2023	27.01.2023
5.	<i>Розділ 2. Вибір програмних засобів та проєктування</i>	03.03.2023	03.03.2023
6.	<i>Розділ 3. Реалізація веб-додатку</i>	14.04.2023	14.04.2023
7.	<i>Висновки</i>	28.04.2023	28.04.2023
8.	<i>Здача випускного кваліфікаційного проекту на кафедру (перша перевірка)</i>	17.05.2023	17.05.2023
9.	<i>Підготовка автореферату та презентації доповіді</i>	26.05.2023	26.05.2023
10.	<i>Попередній захист випускного кваліфікаційного проекту</i>	29.05.2023 – 02.06.2023	
11.	<i>Зовнішнє рецензування випускного кваліфікаційного проекту</i>	05.06.2023	05.06.2023
12.	<i>Здача прошого випускного кваліфікаційного проекту на кафедру</i>	05.06.2023	05.06.2023
13.	<i>Публічний захист випускного кваліфікаційного проекту</i>		

7. Дата видачі завдання «14» листопада 2022 р.

8. Науковий керівник випускного кваліфікаційного проекту \_\_\_\_\_

Харченко О.А.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми \_\_\_\_\_

Рзаєва С.Л.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент \_\_\_\_\_

Марценюк О.В.

(прізвище, ініціали, підпис)



## АНОТАЦІЯ

Відповідно до мети дослідження випускна кваліфікаційна робота присвячена розробці та реалізації веб-орієнтованого додатку з продажу туристичного спорядження. Метою роботи є розробка веб-орієнтованого додатку з продажу туристичного спорядження, який дозволить клієнтам придбати необхідне спорядження за допомогою зручного та простого інтерфейсу.

**Ключові слова: ...**

## ABSTRACT

In accordance with the research objective, the final qualification work is devoted to the development and implementation of a web-based application for the sale of tourist equipment.

The aim of the work is to develop a web-based application for the sale of tourist equipment, which will allow customers to purchase the necessary equipment using a convenient and simple interface.

**Keywords: ...**

<b>ЗМІСТ</b>	
<b>ВСТУП.....</b>	<b>3</b>
<b>РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ.....</b>	<b>5</b>
1.1. Загальні відомості з предметної галузі.....	5
1.2. Аналіз методів реалізації.....	9
1.3. Призначення розробки та галузь застосування.....	11
1.4. Мета створення.....	12
1.5. Вимоги до системи.....	12
1.5.1. Вимоги до інформаційної безпеки.....	12
1.5.2. Вимоги до параметрів технічних засобів.....	12
1.6. Висновки до розділу.....	13
<b>РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА ВИБІР ПРОГРАМНИХ ЗАСОБІВ.....</b>	<b>15</b>
2.1. Аналіз та вибір архітектури додатку.....	15
2.2. Функціональне призначення програми.....	17
2.3. Опис використаних технологій та програмних засобів.....	18
2.4. Опис структури системи.....	21
2.5. Висновки до розділу.....	25
<b>РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ «ExploreZone».....</b>	<b>26</b>
3.1. Опис реалізації сайту.....	26
3.2. Опис функціональної частини сайту.....	28
3.3. Висновок до розділу 3.....	30
<b>ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....</b>	<b>31</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>32</b>
<b>КЕРІВНИЦТВО КОРИСТУВАЧА.....</b>	<b>.....</b>
<b>ДОДАТКИ.....</b>	<b>.....</b>

<i>ДТЕУ 121 6-1.БР</i>				
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
Зав. каф.		Криворучко О.В.		23.12.22
Керівник		Харченко О.А.		23.12.22
Гарант		Рзаєва С.Л.		23.12.22
Розробив		Марценюк О.В.		23.12.22
Зміст				
Веб-орієнтований додаток з продажу туристичного спорядження		Стадія	Аркуш	Аркушів
		3	2	50
Факультет інформаційних технологій 4 курс, 6 група				

## ВСТУП

*Актуальність:* Сучасний світ, який постійно змінюється та розвивається, вимагає від підприємств та організацій зміни своєї діяльності та способів продажу товарів та послуг. Одним із найбільш перспективних напрямків є онлайн-продажі через веб-орієнтовані додатки. Тому розробка веб-додатку для комерційного продажу є дуже актуальною та потрібною сьогоднішньому ринку.

*Мета дослідження:* розробка веб-орієнтованого додатку з продажу туристичного спорядження, який дозволить клієнтам придбати необхідне спорядження за допомогою зручного та простого інтерфейсу.

*Предмет дослідження:* розробка веб-додатку з продажу туристичного спорядження, технології що використовуються при розробці веб-додатків, та методи інтерфейсної оптимізації для зручності використання.

У відповідності з метою дослідження поставлені наступні завдання:

- Розробка інтерфейсу взаємодії;
- Розробка та реалізація нових або адаптація відомих моделей взаємодії;
- Розробка архітектури веб-додатку;

*Практичне значення дослідження:* полягає в розробці веб-орієнтованого додатку для продажу туристичного спорядження є актуальною та перспективною галуззю бізнесу. Результати дослідження та розробки даного додатку дозволить не тільки забезпечити клієнтів необхідним спорядженням, але і залучити нових користувачів до даної сфери бізнесу, що в свою чергу призведе до збільшення прибутку та розвитку компанії.

					<i>ДТЕУ 121 6-1.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Веб-орієнтований додаток з продажу туристичного спорядження  <i>Вступ</i>	<i>Стадія</i>	<i>Арку</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		23.12.22		<i>В</i>	<i>3</i>	<i>50</i>
Керівник		Харченко О.А.		23.12.22		<i>Факультет інформаційних технологій 4 курс, 6 група</i>		
Гарант		Рзаєва С.Л.		23.12.22				
Розробив		Марценюк О.В.		23.12.22				



У процесі розробки додатку будуть враховані потреби та побажання користувачів, що дозволить створити продукт, який буде зручним та привабливим для покупців.



Зм.	Аркуш	№ докум.	Підпис	Дата

ДТЕУ 121 6-1.БР

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ

#### 1.1. Загальні відомості з предметної галузі

Розробка веб-додатків онлайн магазинів є актуальною темою, оскільки в сучасному світі все більше людей віддає перевагу покупці товарів та послуг в Інтернеті. Онлайн магазини стають все популярнішими як для бізнесу, так і для клієнтів, оскільки вони забезпечують більш зручний та швидкий спосіб придбання товарів та послуг.

Створення веб-додатків вимагає знань з веб-програмування, дизайну, баз даних, а також знання специфіки бізнес-процесів відповідної компанії. Окрім того, при створенні веб-застосунків онлайн магазинів потрібно враховувати психологію покупців, їхні поведінкові звички, оскільки це важливо для успішної взаємодії з кінцевим продуктом.

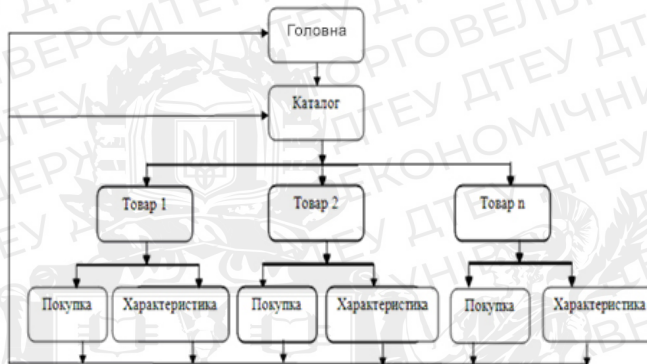
Взаємодія клієнтів з кінцевим продуктом відбувається через веб-інтерфейс онлайн магазину. Цей інтерфейс повинен бути зручним та інтуїтивно зрозумілим для користувачів, мати просту навігацію та забезпечувати швидку та безпечну оплату за товари та послуги. Окрім того, важливо забезпечити користувачам зручний пошук товарів, можливість фільтрувати товари за певними категоріями та збереження вибраних товарів до списку бажань.

Онлайн-магазини можна класифікувати за кількома критеріями, зокрема за способом продажу, товарною категорією, географічним простором тощо. За способом продажу розрізняють магазини, які власними силами здійснюють доставку та магазини, які працюють з доставкою через сторонні компанії.

					<i>ДТЕУ 121 6-1.БР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		23.12.22	Веб-орієнтований додаток з продажу туристичного спорядження	Стадія	Аркуш	Аркушів
Керівник		Харченко О.А.		23.12.22		P1	5	50
Гарант		Рзаєва С.Л.		23.12.22		Факультет інформаційних технологій 4 курс, 6 група		
Розробив		Марценюк О.В.		23.12.22				

Товарна категорія може бути дуже різноманітною, від одягу та взуття до продуктів харчування та електроніки. Інтернет-магазини можна класифікувати за такими видами:

- інтернет-вітрина;
- інтернет магазин;
- онлайн аукціон;



• Рис. 1.1 Структура інтернет вітрини

Інтернет-вітрина – це вид онлайн-магазину, в якому товари відображаються на одній сторінці відповідно до певних категорій або тематики. Інтернет-вітрина зазвичай пропонує товари з різних категорій, такі як електроніка, одяг, аксесуари, спортивний інвентар і т.д., але вони можуть також фокусуватися на конкретному виді товарів. Користувачі можуть переглядати товари на одній сторінці та здійснювати покупки безпосередньо з цієї сторінки. Інтернет-вітрина може бути створена як окремий веб-сайт або як частина більшого онлайн-магазину.

Зм.	Аркуш	№ докум.	Підпис	Дата

ДТЕУ 121 6-1.БР

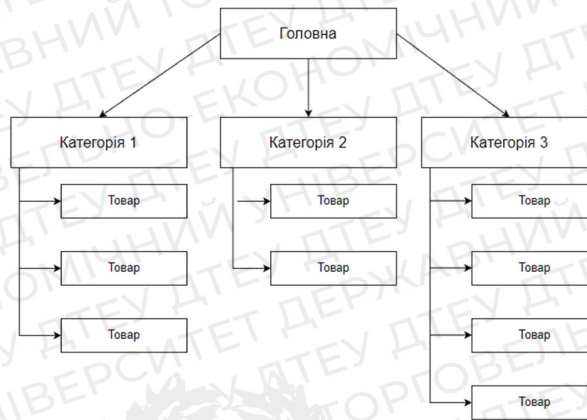


Рис. 1.2 Структура інтернет магазину

Інтернет-магазин – це електронний магазин, який дозволяє користувачам купувати товари та послуги через Інтернет. Він працює за допомогою веб-сайту або спеціальної програми, яка дозволяє вести торгівлю в Інтернеті.

В інтернет-магазині користувач може шукати товари, дізнаватися про їх характеристики, порівнювати ціни, зробити замовлення та оплатити його за допомогою різних електронних методів оплати. Зазвичай інтернет-магазини мають великий асортимент товарів, який може бути значно ширшим, ніж у звичайних крамницях. Також, завдяки можливості швидко та легко порівнювати ціни, користувач може знайти товари за найнижчою ціною, порівняно з офлайн-магазинами.

Інтернет-магазини зазвичай мають відкриту цілодобову доступність та є більш зручними для користувачів, які не мають можливості відвідати звичайний магазин у робочий час. Однак, вони також мають свої недоліки, такі як відсутність можливості дотикової взаємодії з продуктом перед покупкою та можливість отримати неякісний товар або послугу внаслідок шахрайства.

Онлайн аукціон – це електронний ресурс, що дозволяє користувачам купувати та продавати товари чи послуги шляхом участі в торгах, які

Зм.	Аркуш	№ докум.	Підпис	Дата

ДТЕУ 121 6-1.БР

проводяться через Інтернет. Такі аукціони можуть бути відкритими, де будь-який користувач може взяти участь, або закритими, де участь можуть брати тільки зареєстровані користувачі.

Онлайн аукціони зазвичай мають певні правила та умови, які регулюють процес торгів та забезпечують безпеку для продавців і покупців. Такі аукціони можуть бути використані для продажу різноманітних товарів, включаючи мистецтво, антикваріат, електроніку, одяг та інше. Також, онлайн аукціони можуть використовуватися для благодійних та інших громадських цілей.

Вигляд кожного інтернет-магазину має бути привабливим, щоб потенційним клієнтам було приємно відвідувати його та здійснювати покупки. Розробка дизайну інтернет-магазину - це складний процес, що включає не тільки створення зовнішнього вигляду сайту, але й розробку зручної та інтуїтивно зрозумілої навігації для відвідувачів. При розробці дизайну важливо враховувати потреби цільової аудиторії та особливості товарів, що пропонуються в магазині. Наприклад, для магазину з ексклюзивними товари можна використовувати більш креативний та незвичайний дизайн, тоді як для магазину з продуктами харчування важливо, щоб сайт був максимально простим та зрозумілим для клієнтів..

Дизайн сайту повинен відповідати останнім тенденціям та містити інтерактивні елементи, які забезпечують максимальний комфорт та простоту використання. Такі елементи можуть мати анімацію, інтерактивні кнопки та форми, а також інші методи взаємодії з користувачем. Більшість інтернет-магазинів також використовують різноманітні функції для забезпечення зручного та швидкого пошуку товарів, такі як фільтр за категорією або пошук за назвою товару.

Окрім того необхідно розробляти дизайн інтернет-магазину з урахуванням мобільних технологій. Оскільки все більше користувачів

Зм.	Аркуш	№ докум.	Підпис	Дата

*ДТЕУ 121 6-1.БР*

здійснюють покупки через мобільні пристрої, дизайн сайту повинен бути адаптивним та забезпечувати максимальний комфорт користування на будь-якому пристрої. Показ найкращих товарів на головній сторінці є дуже важливим. Цей метод дуже ефективний до відвідувачів які не завжди точно знають що їм необхідно. Показавши найкращі товари у всій красі, ви значно спростите їхній вибір. Також на сайті повинна бути інформація про всі товари у зручній для покупця формі.

Інформацію про товари краще подати у вигляді коротких описів, які містять основні переваги та характеристики. Також важливо не перевантажувати сторінку занадто багато інформацією, щоб не переповнити покупця непотрібною інформацією і не змусити його залишити сайт.

Якщо покупець зацікавиться товаром, має бути можливість перейти до детальної сторінки товару, де він зможе ознайомитись із повною інформацією про товар та його характеристиками. На цій сторінці також має бути можливість додати товар до кошика та оформити замовлення.

## 1.2. Аналіз методів реалізації

Розробникам доступний великий вибір мов програмування та технологій для створення веб додатків. Різні мови та технології мають свої переваги та недоліки, і вибір залежить від конкретних потреб та вимог проекту. Одними з найпоширеніших мов програмування для розробки веб додатків є: Ruby, Java, Python, PHP.

Ruby є динамічною, об'єктно-орієнтованою мовою програмування, яка заснована на принципах простоти та елегантності. Вона має зрозумілу синтаксичну структуру, що сприяє продуктивній розробці веб-додатків. Особливою рисою Ruby є фреймворк Ruby on Rails, який надає повноцінний стек для швидкої розробки веб-додатків. Ruby on Rails використовує принцип конвенції над конфігурацією, що спрощує розробку, прискорює процес і зменшує кількість коду, потрібного для реалізації певних функціональностей.

					<i>ДТЕУ 121 6-1.БР</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	

Завдяки цьому він забезпечує широкі можливості для розробки веб-орієнтованих додатків з продажу, включаючи обробку платежів, управління замовленнями, аутентифікацію користувачів та інші функції, необхідні для ефективної роботи онлайн-магазинів.

Загалом, використання Ruby для розробки дозволяє зосередитись на швидкому створенні функціональності та розробці бізнес-логіки, завдяки його зручному синтаксису та потужному фреймворку Ruby on Rails.

Java є потужною та універсальною мовою програмування, яка широко використовується для розробки веб-додатків. Вона пропонує велику кількість функціональних можливостей та багато інструментів для створення потужних і масштабованих додатків. Одним з найпопулярніших фреймворків для розробки веб-додатків на Java є Spring Framework. Spring надає ефективні інструменти для управління залежностями, обробки HTTP-запитів, обробки форм, аутентифікації, авторизації та багато іншого. Він також підтримує підхід "конфігурація над конвенцією", що дозволяє розробникам швидко налаштовувати і розгортати веб-додатки.

Java як мова для розробки веб додатків дозволяє забезпечити високу продуктивність, масштабованість і надійність додатків. та розширювати свої знання у розробці веб-додатків на Java.

Python також є популярною мовою програмування, яка знайшла широке застосування в розробці веб додатків завдяки тому, щонає чистий і зрозумілий синтаксис, що полегшує розробку і підтримку коду. Один з найпопулярніших фреймворків для розробки веб додатків на Python - Django. Django надає повноцінний набір інструментів для створення складних веб додатків, включаючи моделювання баз даних, обробку HTTP запитів, шаблони, аутентифікацію, авторизацію та інше. Він пропонує гнучкість і розширюваність, дозволяючи розробникам легко налаштовувати додатки під свої потреби.

					<i>ДТЕУ 121 6-1.БР</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	

JavaScript є однією з найпопулярніших мов програмування, яка широко використовується для розробки веб-додатків з продажу. Вона є мовою програмування, що працює на стороні клієнта (веб-браузері) і дозволяє створювати динамічні, інтерактивні та реактивні веб-додатки.

Найпопулярнішим фреймворком для створення веб додатків є React.js який дозволяє розробникам будувати ефективні та масштабовані інтерфейси користувача. Він базується на компонентній архітектурі, що дозволяє розбити інтерфейс на незалежні компоненти, які легко керувати та перевикористовувати.

Вибір JavaScript для розробки веб додатків дозволяє розробникам створювати потужні та динамічні інтерфейси, легко інтегруватися з бекендом та забезпечувати багатофункціональність.

### **1.3. Призначення розробки та галузь застосування**

Як об'єкт розробки системи відображення і адміністрування інтернет-магазину туристичного спорядження розглядається веб-орієнтований додаток інтернет-магазину «ExploreZone», в якому користувач може знайти різноманітне туристичне спорядження та передивитися каталог товарів з функціями сортування за категоріями. Розроблений веб-додаток може бути використано при організації продажів спорядження або інших товарів які мають подібне тематичне застосування.

Розроблена інформаційна система призначена для:

- підвищення продажів компанії;
- надання інформації для ознайомлення клієнтів з товарами;
- забезпечення користувачам сайту простоти і комфортного доступу до каталогу товарів;

Зм.	Аркуш	№ докум.	Підпис	Дата

*ДТЕУ 121 6-1.БР*



## 1.4. Мета створення

Метою розробки веб-орієнтованого додатку з продажу туристичного спорядження є виконання випускної кваліфікаційної роботи. Під час виконання необхідно виконати:

- здійснити збір та аналіз вимог;
- визначити функціональні та нефункціональні вимоги;
- розробити зручний та привабливий інтерфейс;

## 1.5. Вимоги до системи

### 1.5.1. Вимоги до інформаційної безпеки

Безпека веб-додатків - це сукупність заходів та технологій, що застосовуються для захисту веб-додатків від різних видів атак, таких як введення шкідливого коду, злам паролів, викрадення даних, SQL-ін'єкцій та інших. Безпека веб-додатків є критично важливою, оскільки веб-додатки зазвичай збирають, зберігають та обробляють конфіденційну інформацію користувачів, таку як особисті дані, фінансова інформація та інші види чутливої інформації. В даному проекті дані користувачів не зберігаються, тому для захищеності сайту достатньо того, що сайт був розміщений на безпечному та надійному хостингу.

### 1.5.2. Вимоги до параметрів технічних засобів

Ключовими складовими для взаємодії користувача з додатком є постійне інтернет з'єднання та додаток що дозволяє отримувати доступ до веб сайту. Постійне інтернет-з'єднання є необхідним, оскільки воно забезпечує передачу даних між користувачем і сервером, на якому розміщений веб-сайт. Це дозволяє користувачеві завантажувати сторінки, виконувати запити до сервера, отримувати актуальну інформацію та взаємодіяти з функціональністю додатка.

Зм.	Аркуш	№ докум.	Підпис	Дата

ДТЕУ 121 6-1.БР

Додаток, що надає доступ до веб-сайту, може бути веб-браузером на комп'ютері або мобільним додатком на смартфоні чи планшеті. Він служить інтерфейсом між користувачем і веб-сайтом, дозволяючи користувачеві переглядати сторінки, взаємодіяти з елементами інтерфейсу, вводити дані та виконувати різні дії.

Для зручного використання веб-інтерфейсу вимагається, щоб технічний засіб, на якому він використовується, відповідав таким вимогам:

- процесор з частотою не менше 2.1Ghz;
- оперативна пам'ять 4 Gb або більше;
- SSD накопичувач або жорсткий диск 120Gb;
- операційна система: Windows 7/8/10/11, Linux, MacOS, ChromeOs;
- доступ до мережі Internet
- веб браузер: Chrome 60+, Safari 10+ , Edge 12+, FirefoxESR+.

### 1.6. Висновки до розділу

У цьому розділі було проведено аналіз та постановка задачі для розробки веб-додатку. Розглянуті загальні відомості з предметної галузі, що включають інформацію про характеристики та особливості веб-орієнтованих додатків з продажу товарів.

Призначення розробки та галузь застосування веб-додатку були чітко визначені. Веб-додаток спрямований на забезпечення продажу товарів і використовується у галузі електронної комерції. Враховуючи потреби та вимоги ринку.

Детально проаналізовані вимоги до програми, що включають функціональні та нефункціональні вимоги. Це дозволяє зрозуміти, які функції повинен виконувати веб-додаток, його параметри продуктивності, безпеки, масштабованості та інші характеристики, які він повинен мати.

					<i>ДТЕУ 121 6-1.БР</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	

Під час аналізу методів реалізації веб-додатків у даному розділі було досліджено, що розробникам доступний широкий вибір мов програмування та технологій для створення веб-додатків. Кожна з розглянутих мов має свої переваги та недоліки, і вибір залежить від конкретних потреб і вимог проекту.

В результаті проведеного аналізу та постановки задачі, створено чітку основу для подальшої розробки веб-додатку. Дані висновки визначають основні напрямки роботи й встановлюють вимоги до розробки, що допоможуть забезпечити успішну реалізацію проекту та досягнення поставлених цілей.



Зм.	Аркуш	№ докум.	Підпис	Дата

*ДТЕУ 121 6-1.БР*

## РОЗДІЛ 2

### ПРОЄКТУВАННЯ ТА ВИБІР ПРОГРАМНИХ ЗАСОБІВ

#### 2.1. Аналіз та вибір архітектури додатку

Архітектура веб-додатку - це організація та структура компонентів, модулів і системних взаємозв'язків, які утворюють веб-додаток. Вона визначає, які компоненти будуть присутні, як вони взаємодіятимуть між собою і як буде організована логіка додатку. Архітектура додатку орієнтується на досягнення певних цілей, таких як масштабованість, надійність, безпека та продуктивність. Вона також враховує вимоги до функціональності додатку та специфіку бізнес-потреб. Ключовими компонентами архітектури веб-додатку є такі компоненти:

- логіка додатку;
- клієнтська сторона;
- серверна частина;
- База даних;

Логіка додатку - це частина програмного коду, яка визначає, як додаток працює і як він взаємодіє з різними компонентами та даними. Вона включає в себе алгоритми, правила, функції та процедури, що виконуються для досягнення функціональних та бізнес-потреб додатку.

Клієнтська сторона - це частина веб-додатку, яка виконується на стороні користувача. Вона відповідає за відображення інтерфейсу користувача (UI) і взаємодію з ним

Зм.	Аркуш	№ докум.	Підпис	Дата	<i>ДТЕУ 121 6-1.БР</i>			
Зав. каф.		Криворучко О.В.		23.12.22	Веб-орієнтований додаток з продажу туристичного спорядження	Стадія	Аркуш	Аркушів
Керівник		Харченко О.А.		23.12.22		P2	15	50
Гарант		Рзаєва С.Л.		23.12.22		Факультет інформаційних технологій		
Розробив		Марценюк О.В.		23.12.22		4 курс, 6 група		

Серверна частина - це частина веб-додатку, яка виконується на стороні сервера. Вона відповідає за обробку запитів від клієнтської сторони, взаємодію з базою даних, обробку бізнес-логіки та надання відповіді клієнту.

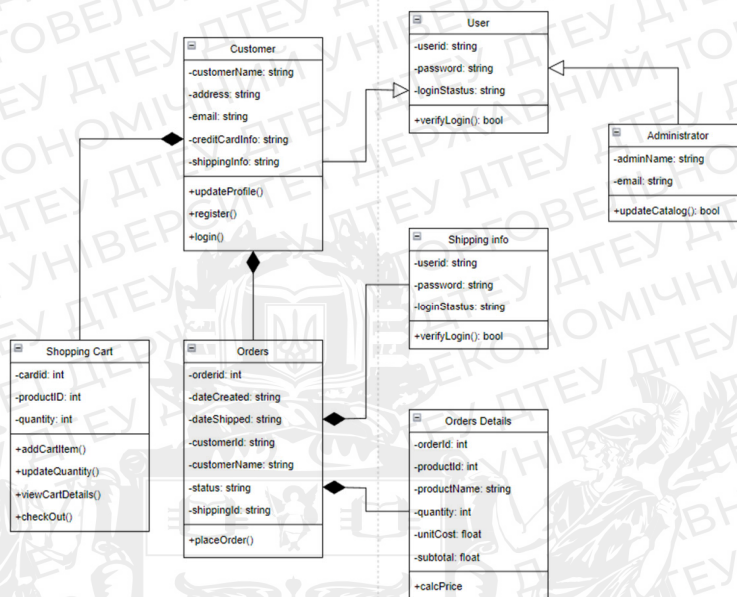


Рис. 2.1 Приклад схеми бази даних онлайн магазину

База даних - це структуроване сховище даних, яке використовується для збереження та організації великих обсягів інформації. Вона дозволяє ефективно зберігати, керувати, оновлювати та використовувати дані в додатках. БД відповідає за збереження, організацію, доступ і управління даними в додатку.

SQLite - це вбудована система управління базами даних, яка надає локальне зберігання та управління даними в одному файлі бази даних без потреби в окремому сервері. При порівнянні SQLite з іншими БД перевагою першої були її простота використання, надійність та стабільність, мала підтримка ресурсів, переносимість, відсутність необхідності в окремому сервері, гнучкість та розширюваність. Основні переваги SQLite:

- Простота використання;
- Надійність та стабільність;
- Відсутність окремого сервера;

Зм.	Аркуш	№ докум.	Підпис	Дата

ДТЕУ 121 6-1.БР

- Гнучкість та розширюваність;

Враховуючи ці переваги, SQLite є гарним вибором для розробки веб-додатків, особливо для проектів з обмеженими ресурсами, з необхідністю кросплатформеного розгортання, та потребою в простому та надійному зберіганні даних.

При виборі архітектури для розробки веб-додатку було вибрано SPA (Single-Page Application) через її переваги при розробці:

- Більш проста розробка та підтримка;
- Ефективне використання ресурсів;
- Покращений користувацький досвід;

Зважаючи на ці переваги, вибір SPA з використанням React.js може бути раціональним для створення ефективного та зручного веб-додатку.

Односторінковий додаток (SPA) з використанням React.js та бази даних SQLite є сучасним інтерактивним веб-додатком, який пропонує зручний і ефективний спосіб взаємодії з користувачем.

Односторінкова архітектура дозволяє додатку працювати на одній HTML-сторінці, яка завантажується лише один раз при початковому запуску, і подальша навігація відбувається без перезавантаження сторінки. Весь контент та функціонал сторінки оновлюються динамічно без перезавантаження сторінки, що забезпечує плавну та швидку взаємодію з користувачем.

## 2.2. Функціональне призначення програми

Під час виконання кваліфікаційної роботи було реалізовано розробку front-end частини системи інтернет магазину. Призначення розробленої системи:

- Перегляд асортименту та його характеристик;
- Легке та доступне керування інтернет магазином за рахунок модульності коду;

Зм.	Аркуш	№ докум.	Підпис	Дата

ДТЕУ 121 6-1.БР

- Оформлення замовлень з даного магазину;
- Зручний пошук необхідного спорядження за параметрами;
- Зручний та простий інтерфейс, що дозволяє користувачам швидко знайти необхідну інформацію та зробити покупку в декілька кліків.

### 2.3. Опис використаних технологій та програмних засобів

Web-додаток має front-end частину, яка була розроблена на мові програмування JavaScript з використанням фреймворку React JS.

HTML є мовою розмітки та CSS є мовою стилів, ці технології використовуються для розробки веб-сторінок і забезпечення їх зовнішнього вигляду, завдяки цьому симбіозу розробники мають можливість створювати привабливі, добре організовані та функціональні веб-сторінки.

HTML – це стандартна мова розмітки, використовувана для створення структури і вигляду веб-сторінок. Вона використовується для визначення ролі та організації елементів на сторінці, таких як заголовки, параграфи, таблиці, посилання, зображення та інші елементи контенту. HTML був розроблений в 1989 році Тімоті Бернерс-Лі та та Роберт Кайо, які створили першу версію HTML як спосіб обміну документами між науковцями у мережі Інтернет.

CSS – це мова стилізації, що використовується для оформлення та вигляду веб-сторінок, що написані за допомогою HTML або інших мов розмітки. CSS визначає зовнішній вигляд елементів на сторінці, включаючи кольори, шрифти, розташування, розміри, відступи, рамки та інші властивості.

SQLite є вбудованою, легко використовуваною та портативною базою даних, яка не потребує окремого сервера, що забезпечує надійність та безпеку даних для збереження інформації у додатку з використанням React.js.

					<i>ДТЕУ 121 6-1.БР</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	

JavaScript – є мовою програмування, яка дозволяє створювати динамічні веб-сторінки та реагувати на взаємодію користувача. JavaScript використовується для створення ефектів, анімацій, валідації форм, маніпулювання DOM-елементами, взаємодії з користувачем, виконання AJAX-запитів і багатьох інших функцій, які забезпечують багатofункціональність та зручність веб-додатків. Зазвичай використовується як мова програмування на стороні клієнта для реалізації динамічної взаємодії та інтерактивності на веб-сторінках.

Вам не потрібно перезавантажувати сторінку, щоб запустити програму, всі програми працюють у реальному часі. Наприклад, перед відправкою форми з даними ви можете перевірити значення і, якщо дані не відповідають вашим очікуванням, заборонити передачу інформації. Js також може використовуватись на стороні сервера (server-side JavaScript). Це відкриває можливості для створення серверних додатків, розробки API, роботи з базами даних та інших завдань, пов'язаних зі стороною сервера.

React.js - це декларативна, ефективна і гнучка JavaScript-бібліотека, призначена для створення інтерфейсів користувача. Вона дозволяє компонувати складні інтерфейси з невеликих окремих частин коду – «компонентів». Основна ідея React полягає в тому, що користувацький інтерфейс складається зі складових частин, які можуть бути перевикористовані в різних частинах додатку.

React використовує спеціальну мову розмітки JSX, яка поєднує в собі HTML-подібний синтаксис та JavaScript-код. Це дозволяє розробникам легко створювати компоненти та забезпечувати їхню взаємодію з іншими частинами додатка. Одна з ключових особливостей React - це використання віртуального DOM (Document Object Model). Він забезпечує швидкодію додатку, оскільки React виконує оновлення компонентів віртуального DOM та оновлює тільки ті частини реального DOM, які змінилися.

					<i>ДТЕУ 121 6-1.БР</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	



Під час розробки архітектури та вибору мови написання додатку також необхідно вибрати IDE для розробки. Для написання коду додатку було використане IDE Visual Studio Code.

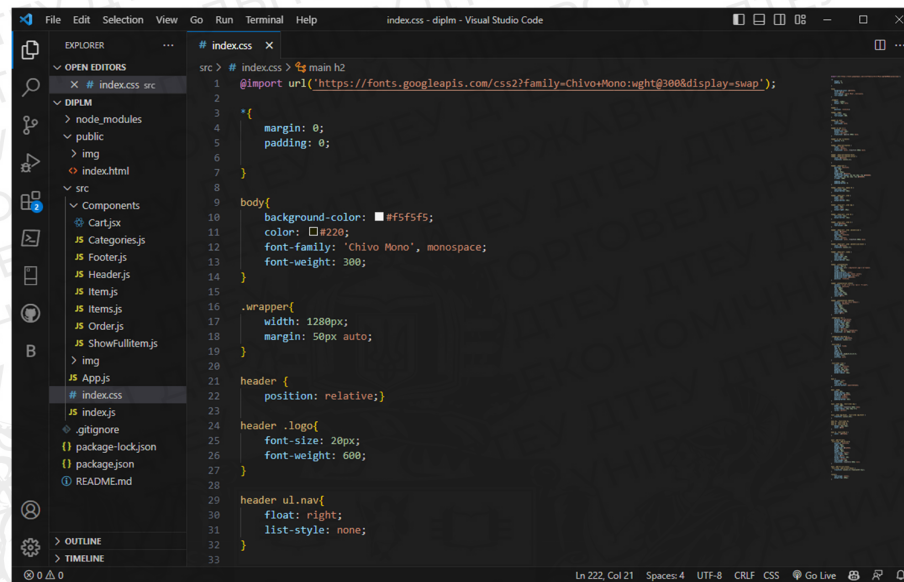


Рис. 2.1 Інтерфейс середовища розробки VS Code

IDE - це програмна система, яка надає розробникам зручне середовище для написання, відлагодження та виконання програмного коду. Вона забезпечує інструменти для розробки, тестування і налагодження програм, а також надає інтерфейс для управління проектами.

Visual Studio Code (VS Code) - це безкоштовне, легке та потужне інтегроване середовище розробки, (розроблене компанією Microsoft). Воно призначене для роботи з різними мовами програмування та підтримується на різних операційних системах, включаючи Windows, macOS та Linux.

VS Code підтримує широкий спектр розширень, які дозволяють налаштовувати середовище розробки для різних потреб. Розширення включають підтримку різних мов програмування, інструменти для роботи з Git, відлагоджувальники та інтеграцію з хмарними сервісами.

					<i>ДТЕУ 121 6-1.БР</i>
Зм.	Аркуш	№ докум.	Підпис	Дата	

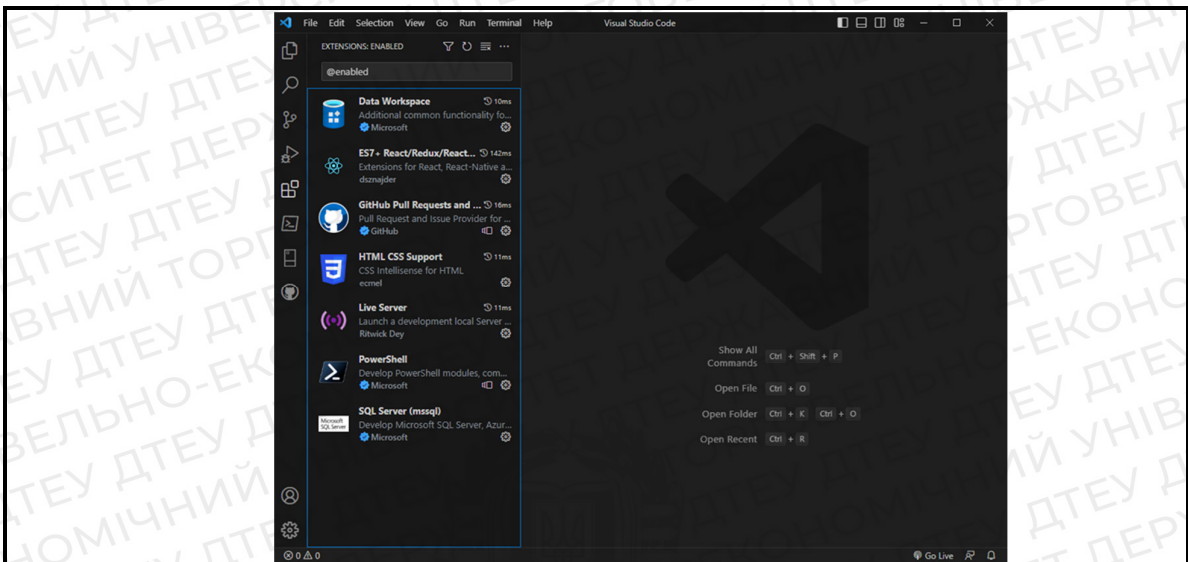


Рис. 2.2 Приклад використаних розширень.

ES7+ React/Redux/React-Native snippets - це розширення для редакторів коду, таких як VS Code, яке надає набір скорочень для розробки веб та мобільних додатків з використанням React, Redux і React Native. Це допомагає розробникам писати код швидше та ефективніше, використовуючи готові шаблони для типових завдань.

Розширення HTML CSS Support є додатком для редакторів коду, таких як Visual Studio Code, яке надає розширену підтримку HTML та CSS. Воно спрощує написання коду і полегшує розробку веб-сторінок та інтерфейсів, надаючи додаткові функції та можливості.

#### 2.4. Опис структури системи

Структура веб-сайту визначає спосіб, яким користувач може навігувати між сторінками та взаємодіяти з інтерфейсом сайту. Для залучення та зацікавлення відвідувачів, структура сайту повинна бути простою та легко зрозумілою для користувача. Всі користувачі цінують, коли вони можуть швидко зорієнтуватись на сайті, знаючи, де знаходяться необхідні їм елементи, щоб не витратити час на пошук потрібної інформації.

					<i>ДТЕУ 121 6-1.БР</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	

СУ веб-додатку, який буде розроблено, використана проста структура, яка дозволяє користувачеві швидко переходити з будь-якої сторінки в будь-який розділ. Це значно зберігає час для користувача.

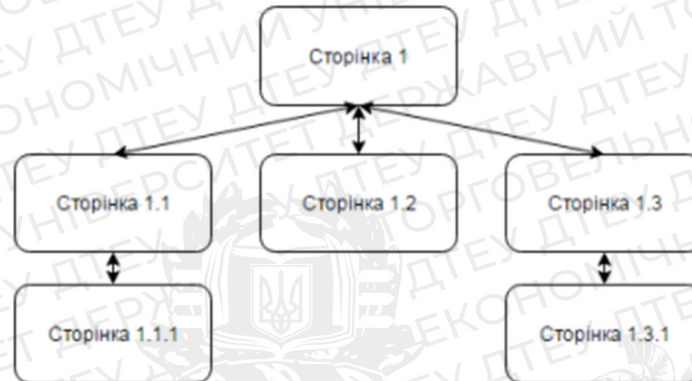


Рис. 2.3. Структура сайту

У відповідності з метою дослідження поставлені наступні завдання:

- головна сторінка;
- каталог товарів з можливістю сортуванням за типом товарів;
- сторінка товару з описом;
- сторінка с кошиком;
- сторінка замовлення.

Мінімальна структура додатку на React створюється автоматично після введення команди в консоль `create-react-app`.

Create React App автоматично створює основну структуру проекту, яка містить файли: `.gitignore`, `package.json`, `README.md`, а також створює папки: `public`, `src` та `node_modules`. А також встановлює необхідні залежності та налаштовує середовище розробки. Він також надає вбудований сервер розробки, який дозволяє переглядати та тестувати додаток під час розробки.

					<i>ДТЕУ 121 6-1.БР</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	

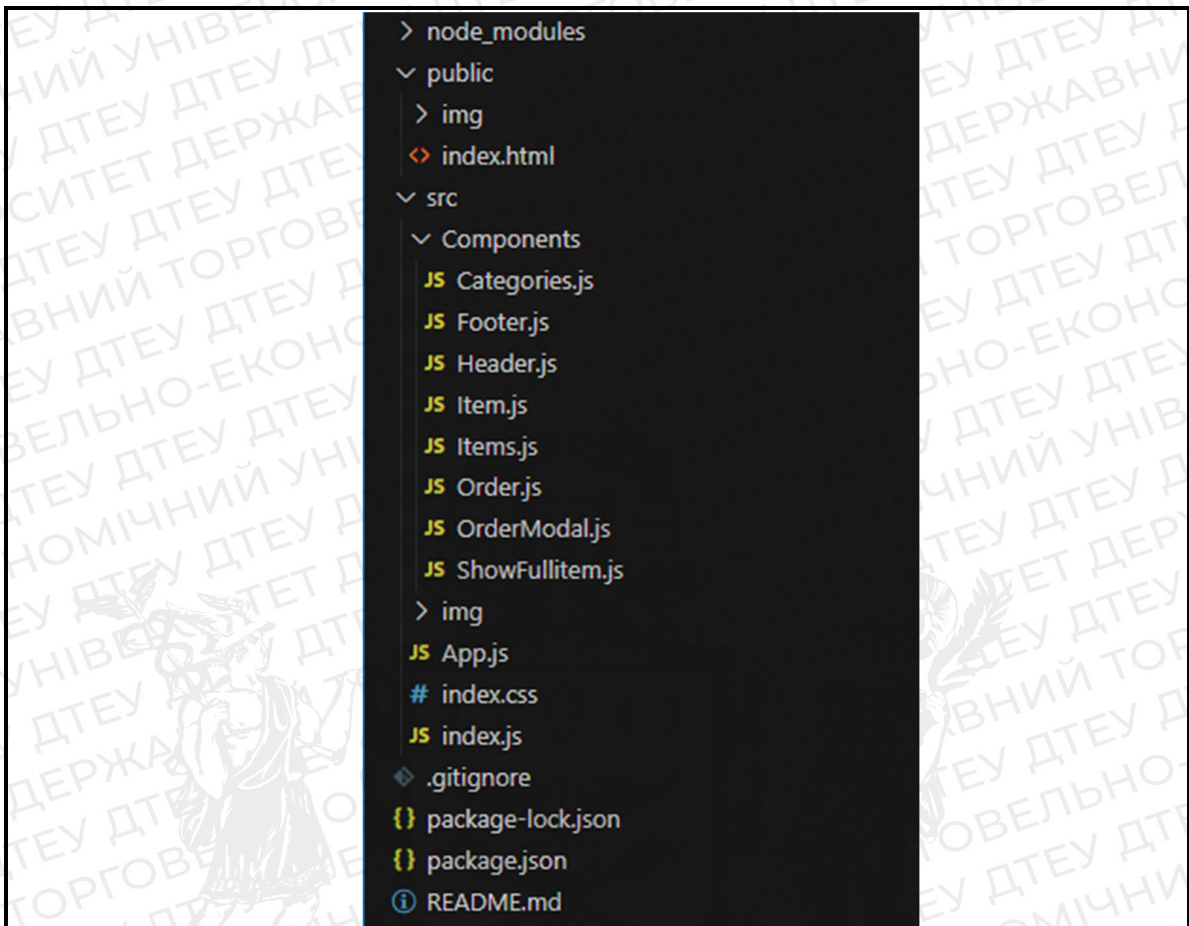


Рис. 2.4. Структура файлової системи

Папка `public` містить стандартний HTML файл який налаштовується під час розробки за необхідності. Тег `<script>` буде додано до нього автоматично під час процесу збірки.

В проектах, що використовують React, у більшості випадків, файл `index.html` залишається стандартним, без необхідності змінювати його вручну. Оскільки це початковий шаблон, що генерується автоматично під час створення проекту.

Зм.	Аркуш	№ докум.	Підпис	Дата

ДТЕУ 121 6-1.БР

```
index.html
public > index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1" />
6     <title>React App</title>
7   </head>
8   <body>
9     <div id="root"></div>
10  </body>
11 </html>
12
```

Рис. 2.5. Файл index.html

В папці src зберігається основний код додатку. Папка src має декілька розділів такі як:

- api;
- components;
- img;

React дозволяє визначати компоненти як класи або функції, що дає розробникам гнучкість у виборі підходу до написання коду. Коли компонент визначається як клас, він наслідується від базового класу React.Component в розділі component розміщенні всі головні компоненти з їхніми стилями.

Модулем який відіграє ключову роль у навігації користувача є Header (шапка сайту). Він допомагає користувачеві розібратись, де знайти потрібну інформацію та перейти до наступної сторінки.

Footer(підвал сайту) – це нижня частина веб сайту, яка призначена для логічного завершення сторінки де користувачі можуть знайти додаткову інформацію, умови використання, та інші релевантні посилання.

					<i>ДТЕУ 121 6-1.БР</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	

## 2.5. Висновки до розділу

Отже, підсумовуючи вище викладене, була обрана архітектура додатку SPA (Single Page Application), яка дозволяє створювати веб-додатки, які загружаються один раз, а далі переходять між різними розділами без перезавантаження сторінки. Це забезпечує більш швидку та плавну роботу додатку, знижує навантаження на сервер та поліпшує користувацький досвід.

Функціональне призначення програми визначено як продаж товарів. Веб-додаток дозволить користувачам переглядати, вибирати та здійснювати покупки різних товарів онлайн.

Описані використані технології та програмні засоби, зокрема фреймворк React.js, який є популярним та потужним фреймворком для розробки користувацького інтерфейсу веб-додатків. Використання цього фреймворку дозволяє створити ефективну та гнучку фронтенд частину додатку з високою швидкістю реакції.

Описана структура системи, передбачає розбиття додатка на компоненти, модулі та шари. Це сприяє організації коду, полегшує розробку та підтримку, а також сприяє повторному використанню коду та розширенню функціональності.

Завдяки вибору react.js та SPA архітектури, веб-додаток отримує сучасну, швидку та гнучку фронтенд реалізацію, що відповідає потребам сучасних веб-додатків з продажу товарів. Використання цих технологій та підходів дозволяє забезпечити ефективну та зручну роботу користувачів і досягти успіху в галузі електронної комерції.

Зм.	Аркуш	№ докум.	Підпис	Дата

*ДТЕУ 121 6-1.БР*

### РОЗДІЛ 3

#### РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ «ExploreZone»

В попередніх розділах була вибрана архітектуру веб-додатку, а також програмні компоненти, які будуть необхідні для реалізації проекту.

#### 3.1. Опис реалізації сайту

Проект сайт «ExploreZone» - це інтернет магазин при розробці якого реалізовані сучасні методи розробки та адміністрування для забезпечення ефективності та зручності користування.

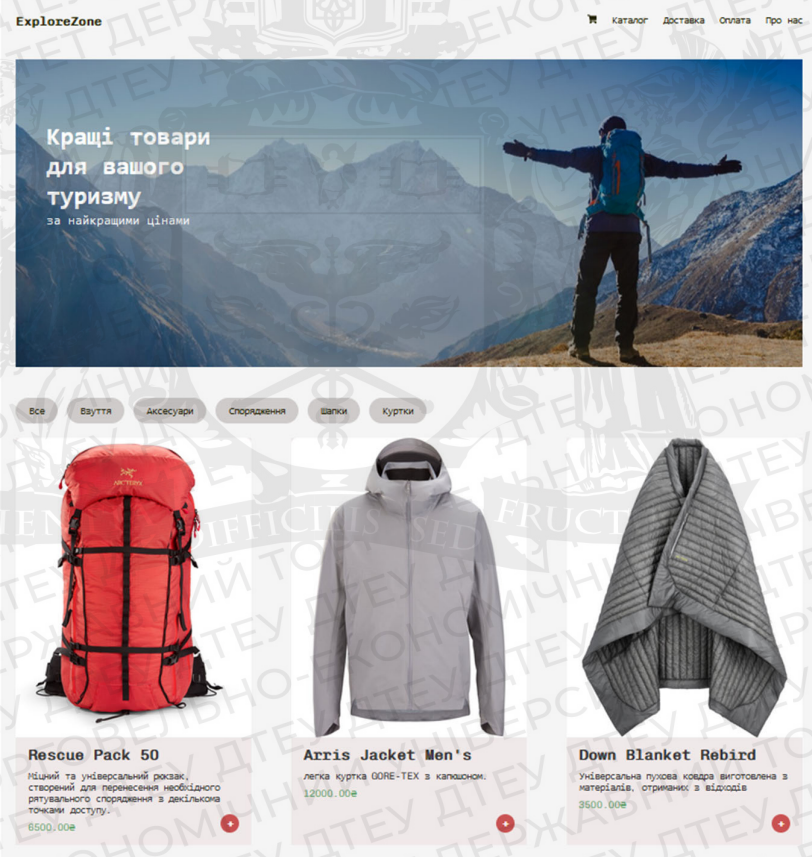


Рис. 3.1 Головна сторінка сайту

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 6-1.БР			
Зав. каф.	Криворучко О.В.			14.04.23	Веб-орієнтований додаток з продажу туристичного спорядження	Стадія	Аркуш	Аркушів
Керівник	Харченко О.А.			14.04.23		РЗ	26	50
Гарант	Рзаєва С.Л.			14.04.23		Факультет інформаційних технологій		
Розробив	Марценюк О.В.			14.04.23		4 курс, 6 група		

На головній сторінці розташовані основні елементи, які спрямовані для забезпечення легкої навігації та залучення користувачів. Потенційний покупець може одразу відсортувати необхідну йому товарну групу, та додати товар у кошик за допомогою інтуїтивно зрозумілих значків.

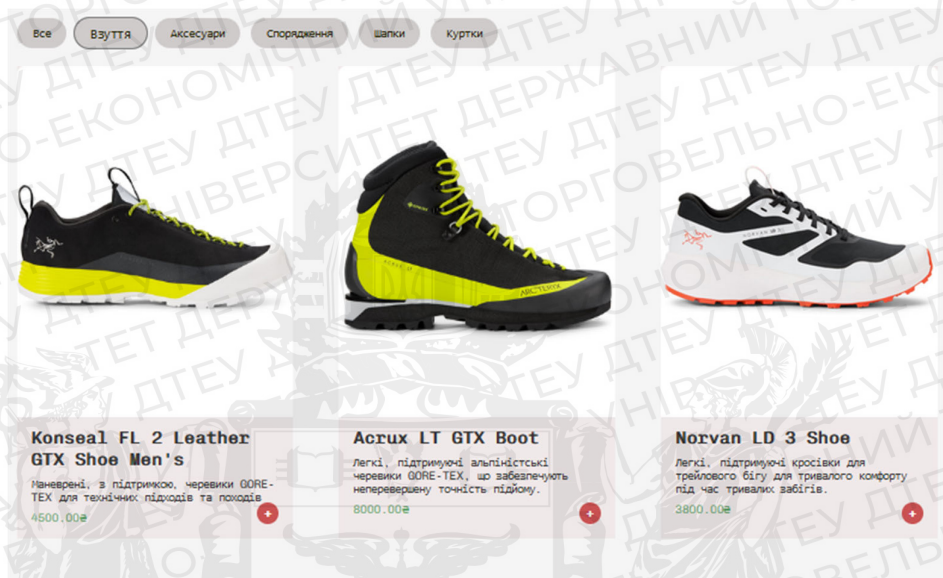


Рис. 3.2 Приклад сортування за товарною групою

Для отримання детальної інформації про товар була реалізована базова структура картки товару, яка характеризується наступними елементами:

- **Зображення товару:** Включає зображення товару, що допомагає візуалізувати його зовнішній вигляд або особливості,;
- **Назва товару:** Містить назву або заголовок товару, що ідентифікує його;
- **Опис товару:** Включає короткий або докладний опис, що пояснює особливості, властивості або використання товару;
- **Ціна:** Вказує вартість товару в державній валюті;
- **Кнопка «+»:** що надає можливість покупцю легко додати товар до кошика або здійснити покупку безпосередньо з картки товару.

Зм.	Аркуш	№ докум.	Підпис	Дата

ДТЕУ 121 6-1.БР





Рис. 3.3 Приклад картки товару

### 3.2. Опис функціональної частини сайту

В розробленому додатку реалізована функція кошику у вигляді модального вікна яке слугує підтвердженням дії для користувача. На сайті присутня можливість оформити замовлення одного або декілька товарів з різних категорій, користувач має можливість видаляти товари з кошику.

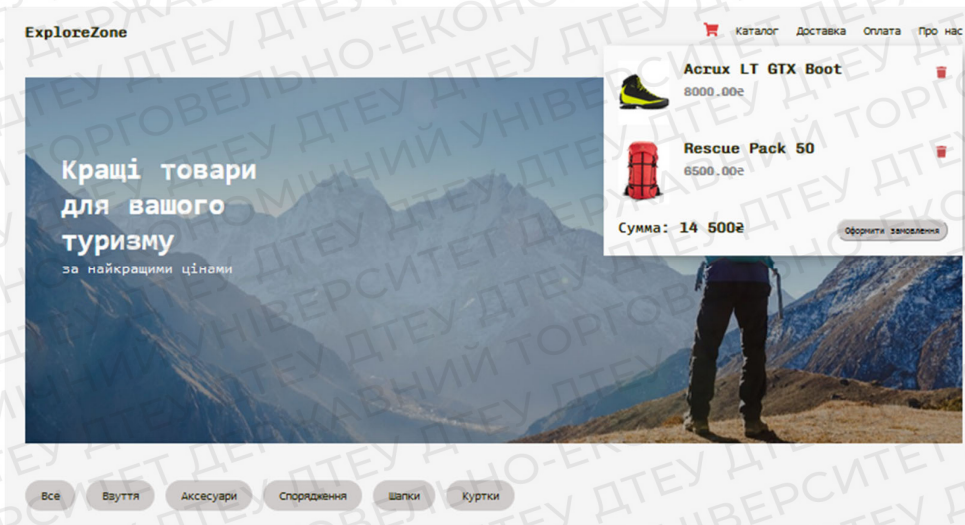


Рис. 3.4 Реалізація кошику

					<i>ДТЕУ 121 6-1.БР</i>
Зм.	Аркуш	№ докум.	Підпис	Дата	

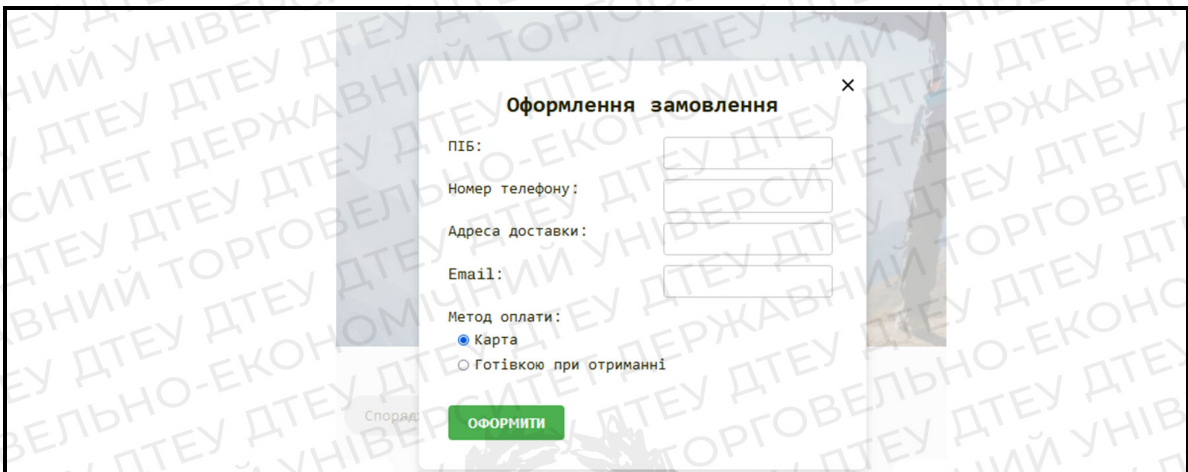


Рис. 3.5 Оформлення замовлення

Після перегляду та заповнення форми замовлення, покупець натискає кнопку «Оформити», відкривається вікно, де відображається повідомлення про те, що замовлення було успішно оформлене й знаходиться в обробці.

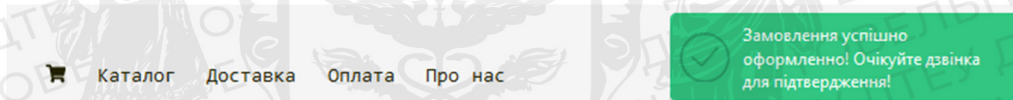


Рис. 3.6 Повідомлення про замовлення

Лист, що містить інформацію про замовлення та контактні дані покупця, надіслано на електронну адресу інтернет-магазину. Це зроблено з метою, щоб менеджер магазину міг зв'язатися з покупцем.

Зм.	Аркуш	№ докум.	Підпис	Дата

ДТЕУ 121 6-1.БР

### 3.3. Висновок до розділу 3

Отже, використання односторінкової архітектури (SPA) у розробленому веб додатку дозволяє забезпечити швидку та ефективну навігацію для користувачів. Використання фреймворку React.js дозволяє нам побудувати користувацький інтерфейс, використовуючи компонентний підхід.

Завдяки компонентному підходу React.js, ми можемо розділити інтерфейс на невеликі ізольовані компоненти, що спрощує розробку, підтримку та розширення нашого веб-додатку. Кожен компонент може мати свою внутрішню логіку, стан та методи, що дозволяє нам зручно управляти різними аспектами користувацького інтерфейсу.

Крім того, використання React.js дозволяє нам перевикористовувати компоненти, що дозволяє нам зменшити дублювання коду та покращити підтримку та розширюваність нашого додатку. комбінація SPA з React.js на фронтенді та використання бекенду дозволяють нам створити потужний, швидкий та масштабований веб-додаток з продажу туристичного спорядження.

					<i>ДТЕУ 121 6-1.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		30

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В ході виконання випускного кваліфікаційного проєкту було реалізовано інтернет-магазин з продажу туристичного спорядження на мові програмування JavaScript та JavaScript-бібліотеки React. В результаті виконання кваліфікаційної роботи було виконано такі задачі як:

1. Проведено аналіз ринку веб-орієнтованих додатків з продажу товарів та виявлено ключові тенденції та конкурентні переваги. Були враховані потреби та вимоги цільової аудиторії, що дозволило створити сайт "ExploreZone" з урахуванням їхніх очікувань.
2. Проведено аналіз методів реалізація веб-додатків за допомогою різних мов програмування та їх фреймворків. Було вивчено і порівняно різні варіанти реалізації веб-додатків, такі як використання Java з фреймворком Spring, Python з фреймворком Django, Ruby з фреймворком Ruby on Rails та JavaScript з фреймворком React.js.
3. В процесі виконання проєкту розроблено веб-додаток за допомогою обраного методу реалізації, який включав в себе використання React.js та архітектури SPA. Додаток дозволяє користувачам переглядати товари, здійснювати пошук, додавати їх до кошика та оформляти замовлення. Для реалізації функціональності були використані різноманітні компоненти, такі як форми, списки, кнопки та інші елементи інтерфейсу.
4. Розроблена система повністю задовольняє потреби користувачів надаючи їм зручний та інтуїтивно зрозумілий спосіб придбання туристичного спорядження.

Зм.	Аркуш	№ докум.	Підпис	Дата	ДТЕУ 121 6-1.БР			
Зав. каф.		Криворучко О.В.		28.04.23	Веб-орієнтований додаток з продажу туристичного спорядження	Стадія	Аркуш	Аркушів
Керівник		Харченко О.А.		28.04.23		ВП	31	50
Гарант		Рзаєва С.Л.		28.04.23		Факультет інформаційних технологій		
Розробив		Марценюк О.В.		28.04.23		4 курс, 6 група		
					Висновки та пропозиції			

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Архітектура програмного забезпечення [Електроний ресурс] Режим доступу: <https://javarush.com/groups/posts/2519-chastjh-2-pogovorim-nemnogo-ob-arkhitekture-po>
2. Кайл Сімпсон «Objects & Classes», 2015. 296 с. Режим доступу: <https://github.com/getify/You-Dont-Know-JS/blob/2nd-ed/objects-classes/README.md>
3. Стоян Тэфанов «React.js Швидкий старт», 2017. 304с.
4. Документація React.js [Електроний ресурс] Режим доступу: <https://react.dev/learn>
5. Документація Node.js [Електроний ресурс] Режим доступу: <https://nodejs.org/uk/docs>
6. Документація SQLite [Електроний ресурс] Режим доступу: <https://www.sqlite.org/docs.html>
7. Документація JavaScript [Електроний ресурс] Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
8. Різновиди інтерфейсів [Електроний ресурс] Режим доступу: <https://cases.media/article/sho-take-interfeis-riznovidi-interfeisiv>
9. Структури інтернет магазинів [Електроний ресурс] Режим доступу: <https://blogchain.com.ua/stvorennia-stryktyri-internet-magazyny-shema-kategorii/>
10. Документація IDE VS code [Електроний ресурс] Режим доступу: <https://code.visualstudio.com/docs>

<i>ДТЕУ 121 6-1.БР</i>				
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
Зав. каф.		Криворучко О.В.		28.04.23
Керівник		Харченко О.А.		28.04.23
Гарант		Рзаєва С.Л.		28.04.23
Розробив		Марценюк О.В.		28.04.23
Веб-орієнтований додаток з продажу туристичного спорядження				
Список використаних джерел				
		<i>Стадія</i>		<i>Аркуш</i>
		<i>СВД</i>		<i>32</i>
		<i>Аркушів</i>		<i>50</i>
Факультет інформаційних технологій 4 курс, 6 група				

11. С. Теленик, В. Войналович, Д. Смаковський Архітектура веб-додатків для кластера Kubernetes на хмарній платформі Google із горизонтальним автоматичним масштабуванням // Міжвідомчий науково-технічний збірник "Адаптивні системи автоматичного управління". - 2021-12-15. - №2. - С. 98-105.
12. Malokhvii E. Аналітичний огляд та порівняння сучасних javascript рішень для розробки веб-додатків / E. Malokhvii, V. Buhai, H. Molchanov, O. Chernykh // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2021. – Т. 4 (66). – С. 55-58. – doi:<https://doi.org/10.26906/SUNZ.2021.4.055>.



<i>ДТЕУ 121 6-1.БР</i>				
Зм.	Аркуш	№ докум.	Підпис	Дата
Зав. каф.		Криворучко О.В.		28.04.23
Керівник		Харченко О.А.		28.04.23
Гарант		Рзаєва С.Л.		28.04.23
Розробив		Марценюк О.В.		28.04.23
<i>Список використаних джерел</i>				
			Веб-орієнтований додаток з продажу туристичного спорядження	
			<i>Стадія</i>	<i>Аркуш</i>
			<i>СВД</i>	<i>33</i>
			<i>50</i>	
			Факультет інформаційних технологій	
			<i>4 курс, 6 група</i>	

## КОД ПРОГРАМИ

## ДОДАТОК А

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

@import url('https://fonts.googleapis.com/css2?family=Chivo+Mono:wght@300&display=swap');

* {
  margin: 0;
  padding: 0;
}

body {
  background-color: #f5f5f5;
  color: #220;
  font-family: 'Chivo Mono', monospace;
  font-weight: 300;
}

.wrapper {
  width: 1280px;
  margin: 50px auto;
}

header {
  position: relative;
}

header .logo {
  font-size: 20px;
  font-weight: 600;
}

header ul.nav {
  float: right;
  list-style: none;
}

header ul.nav li {
  display: inline;
  margin-left: 25px;
}
```

```
cursor: pointer;
transition: opacity 500ms ease;
}

header ul.nav li:hover {
  opacity: 0.5;
}

header .shop-cart-button {
  float: right;
  cursor: pointer;
  transition: color, transform 500ms ease;
}

header .shop-cart-button:hover,
header .shop-cart-button.active {
  color: #dc3d3d;
  transform: scale(1.3);
}

header .shop-cart {
  position: absolute;
  top: 30px;
  right: 0;
  width: 450px;
  background: #fafafa;
  -webkit-box-shadow: 4px 5px 15px -7px #606060;
  box-shadow: 4px 5px 15px -7px #606060;
  z-index: 1000;

  padding: 20px;
  padding-bottom: 0;
}

header .shop-cart .empty h2 {
  font-size: 20px;
  margin-bottom: 20px;
}

header .shop-cart .item {
  width: 100%;
  float: left;
  margin-bottom: 20px;
}

header .shop-cart .item img {
  width: 70px;
  float: left;
  margin-right: 20px;
}

header .shop-cart .item h2 {
```





```
font-size: 20px;
margin-bottom: 10px;
}

header .shop-cart .item b {
color:#797979;
font-weight: 600;
}

header .shop-cart .item .delete-icon {
color: #ca5252;
float: right;
position: relative;
top: -25px;
cursor: pointer;
transition: color, transform 500ms ease;
}

header .shop-cart .item .delete-icon:hover {
color: #e41b1b;
transform: scale(1.5);
}

header .shop-cart .summa {
float: left;
width: 100%;
font-weight: 600;
font-size: 20px;
margin-bottom: 20px;
}

header .presentation{
margin: 50px 0;
background: url('./img/banner.jpg') no-repeat;
width: 100%;
height: 500px;
background-size: cover;
background-position: center center;
background-blend-mode: multiply;
background-color: #bcbcbc;
position: relative;
}

header .presentation::after{
content: 'Кращі товари для вашого туризму';
position: absolute;
top: 100px;
left: 50px;
width: 300px;
font-size: 40px;
font-weight: 600;
color: #fff;
}
```



```
header .presentation::before {
  content: 'за найкращими цінами';
  position: absolute;
  top: 250px;
  left: 50px;
  width: 300px;
  font-size: 20px;
  font-weight: 300;
  color: #fff;
}
```

```
.categories div {
  display: inline-block;
  background: #cfcfcb;
  border-radius: 50px;
  padding: 10px 20px;
  margin-bottom: 25px;
  margin-right: 15px;
  cursor: pointer;
  border: 1px solid transparent;
  transition: all 500ms ease;
}
```

```
.categories div:hover {
  border-color: #797979;
  transform: scale(1.1);
}
```

```
.full-item {
  position: fixed;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background: rgba(0,0,0,0.5);
  z-index: 7777;
  overflow: auto;
}
```

```
.full-item > div {
  width: 600px;
  position: relative;
  margin: 10% auto;
  padding: 40px 30px;
  background: #fff;
  border-radius: 10px;
}
```

```
main {
  display: flex;
}
```



```
width: 100%;
flex-wrap: wrap;
justify-content: space-between;
}

main .item {
width: 30%;
margin-bottom: 50px;
background: #f0e9e9;
overflow: hidden;
position: relative;
padding-bottom: 20px;
}

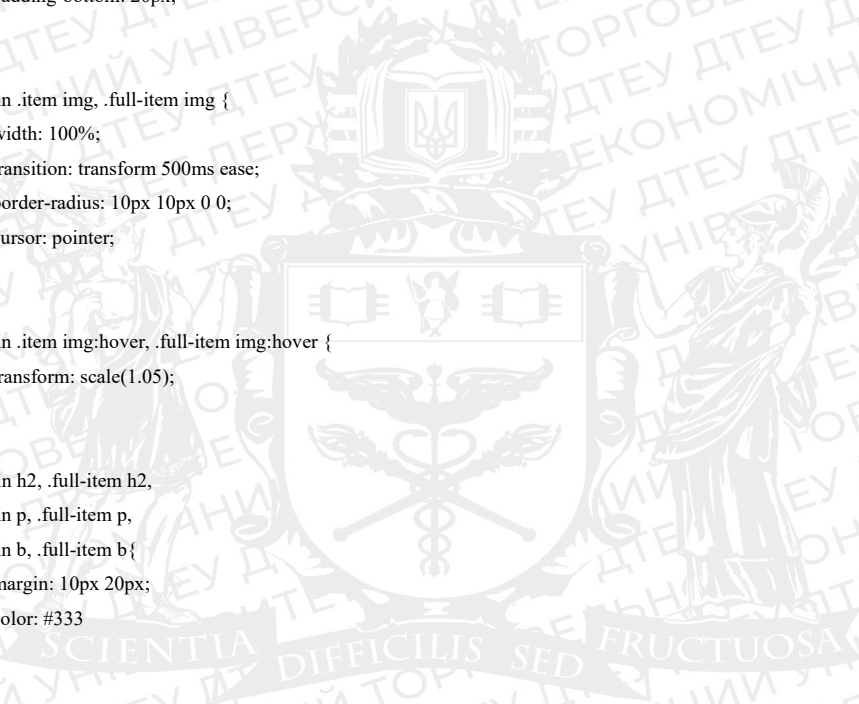
main .item img, .full-item img {
width: 100%;
transition: transform 500ms ease;
border-radius: 10px 10px 0 0;
cursor: pointer;
}

main .item img:hover, .full-item img:hover {
transform: scale(1.05);
}

main h2, .full-item h2,
main p, .full-item p,
main b, .full-item b {
margin: 10px 20px;
color: #333
}

main b, .full-item b {
color: #5fa36a;
}

main .add-to-cart,
.full-item .add-to-cart {
position: absolute;
right: 20px;
bottom: 20px;
background: #ca5252;
width: 30px;
height: 30px;
text-align: center;
line-height: 30px;
color: #fff;
border-radius: 50%;
cursor: pointer;
font-weight: 600;
transition: transform 500ms ease;
}
```



```
main .add-to-cart:hover,
.full-item .add-to-cart:hover {
  transform: scale(1.3) translateY(-5px);
}

footer {
  text-align: center;
  margin-top: 100px;
}

.order_btn {
  display: inline-block;
  background: #cfcfcb;
  border-radius: 25px;
  padding: 5px 10px;
  cursor: pointer;
  float: right;
  font-family: 'Chivo Mono', monospace;
}

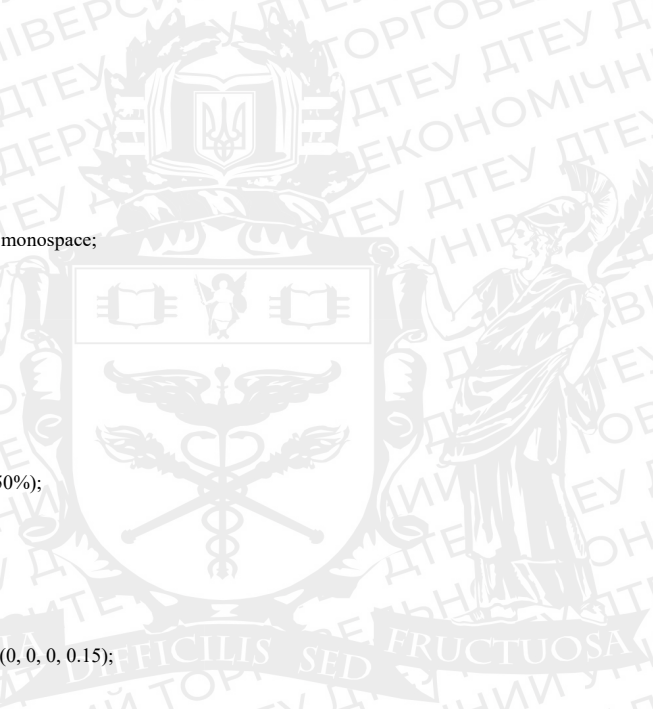
.modal {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  background-color: #fff;
  width: 400px;
  padding: 30px;
  border-radius: 8px;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.15);
}

.modal h2 {
  margin-top: 0;
  margin-bottom: 20px;
  font-size: 24px;
  text-align: center;
}

.modal form {
  display: flex;
  flex-direction: column;
}

.modal label {
  margin-bottom: 5px;
}

.modal input {
  padding: 8px;
  margin-bottom: 10px;
  border-radius: 4px;
}
```



```
border: 1px solid #ccc;
margin-right: 5px;
}

.modal_text_input {
float: right;
}

.modal-buttons {
display: flex;
justify-content: space-between;
margin-top: 20px;
}

.modal-buttons button {
padding: 10px 20px;
border-radius: 4px;
border: none;
cursor: pointer;
font-weight: bold;
text-transform: uppercase;
}

.modal-buttons button[type="submit"] {
background-color: #4caf50;
color: #fff;
}

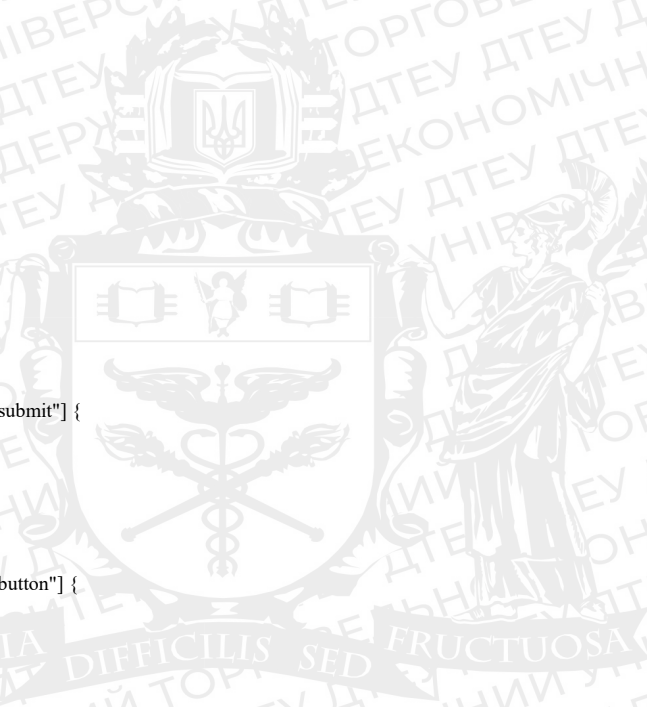
.modal-buttons button[type="button"] {
background-color: #ccc;
color: #fff;
}

.close-button {
position: absolute;
top: 10px;
right: 10px;
background-color: transparent;
border: none;
cursor: pointer;
}

.close-button span {
font-size: 24px;
line-height: 1;
}

.radio_label {
padding-left: 5px;
}

.radio_box {
padding: 5px;
}
```



```
}

import React from 'react';
import Header from './Components/Header';
import Footer from './Components/Footer';
import Items from './Components/Items';
import Categories from './Components/Categories';
import ShowFullitem from './Components/ShowFullitem';
import OrderModal from './Components/OrderModal';

class App extends React.Component {
  constructor(props) {
    super(props)
    this.state = {
      orders: [],
      currentItems: [],
      items: [
        {
          id: 1,
          title: 'Rescue Pack 50',
          img: 'Rescue-Pack-50.jpg',
          desc: 'Міцний та універсальний рюкзак, створений для перенесення необхідного рятувального спорядження з декількома точками доступу.',
          category: 'Equipment',
          price: '6500.00'
        },
        {
          id: 2,
          title: 'Arris Jacket Men's',
          img: 'ArrisJacket.jpg',
          desc: 'Легка куртка GORE-TEX з капшоном.',
          category: 'Jackets',
          price: '12000.00'
        },
        {
          id: 3,
          title: 'Down Blanket Rebird',
          img: 'blanket.jpg',
          desc: 'Універсальна пухова ковдра виготовлена з матеріалів, отриманих з відходів',
          category: 'Equipment',
          price: '3500.00'
        },
        {
          id: 4,
          title: 'Konseal FL 2 Leather GTX Shoe Men's',
          img: 'Krossu.jpg',
          desc: 'Маневрені, з підтримкою, червики GORE-TEX для технічних підходів та походів',
          category: 'Boots',
          price: '4500.00'
        },
        {
          id: 5,
```

title:'Acrux LT GTX Boot',  
img: '4erevuku.jpg',  
desc: 'Легкі, підтримуючі альпіністські черевики GORE-TEX, що забезпечують неперевершену точність підйому.',  
category: 'Boots',  
price: '8000.00'

id: 6,  
title:'Small Bird Cap',  
img: 'Керпка.jpg',  
desc: 'Цілорічна експлуатація в горах і поза ними робить цю шапку ідеальною для трейлових пробіжок, походів, гірськолижних турів та відпочинку в таборі',  
category: 'Caps&Hats',  
price: '1100.00'

id: 7,  
title:'Шапка',  
img: 'Shappka.jpg',  
desc: 'Тепла, зручна шапка із суміші вовни та переробленого поліестеру.',  
category: 'Caps&Hats',  
price: '1300.00'

id: 8,  
title:'Бафф',  
img: 'Wool-Neck.jpg',  
desc: 'забезпечує тепло і має безліч варіантів носіння',  
category: 'Accessories',  
price: '700.00'

id: 9,  
title:'Рукавички',  
img: 'Glove-Black.jpg',  
desc: 'Утеплені та захищені від погодних умов GORE-TEX, найтепліші рукавиці',  
category: 'Accessories',  
price: '1300.00'

id: 10,  
title:'Кепка з вигнутим козирком',  
img: 'Керпкаblue.jpg',  
desc: 'Класичний вигнутий козирок, технічні матеріали - ця модель ідеально підходить для повсякденного носіння, гірських мандрівок',  
category: 'Caps&Hats',  
price: '1100.00'

id: 11,  
title:'Сумка через плече',  
img: 'symka.jpg',  
desc: 'Водонепроникна та міцна, сумка через плече доставить вас туди, куди ви прямуєте',



```

    category: 'Equipment',
    price: '2100.00'
  },
  {
    id: 12,
    title: 'Norvan LD 3 Shoe',
    img: 'kroc.jpg',
    desc: 'Легкі, підтримуючі кросівки для трейлового бігу для тривалого комфорту під час тривалих забігів.',
    category: 'Boots',
    price: '3800.00'
  },
],
showFullitem: false,
showOrderModal: false,
fullItem: {}
}

this.state.currentItems = this.state.items
this.addToOrder = this.addToOrder.bind(this);
this.deleteOrder = this.deleteOrder.bind(this);
this.chooseCategory = this.chooseCategory.bind(this);
this.onShowItem = this.onShowItem.bind(this);
this.onChangeModal = this.onChangeModal.bind(this);
this.cleanupBucket = this.cleanupBucket.bind(this);
}
render() {
  return (
    <div className="wrapper">
      {this.state.showOrderModal} && <OrderModal onClose={() => this.onChangeModal(false)}
      cleanupBucket={this.cleanupBucket} />

      <Header orders={this.state.orders} onDelete={this.deleteOrder} onShowModal={() => this.onChangeModal(true)} />
      <Categories chooseCategory={this.chooseCategory} />
      <Items onShowItem={this.onShowItem} items={this.state.currentItems} onAdd={this.addToOrder} />
      {this.state.showFullitem} && <ShowFullitem onAdd={this.addToOrder} onShowItem={this.onShowItem}
      item={this.state.fullItem} />
      <Footer />
    </div>
  )
}

onChangeModal(state) {
  this.setState({ showOrderModal: state });
  if (state === true) {
    this.setState({ showFullitem: false })
  }
}

onShowItem(item) {
  this.setState({ fullItem: item })
}

```



```
    this.setState({ showFullItem: !this.state.showFullItem })
  }

  cleanupBucket() {
    this.setState({ orders: [] })
  }

  chooseCategory(category) {
    if (category === 'all') {
      this.setState({ currentItems: this.state.items })
      return;
    }

    this.setState({
      currentItems: this.state.items.filter(el => el.category === category)
    })
  }

  deleteOrder(id) {
    this.setState({ orders: this.state.orders.filter(el => el.id !== id) })
  }

  addToOrder = (item) => {
    let isInArray = false;
    this.state.orders.forEach(el => {
      if (el.id === item.id)
        isInArray = true;
    })
    if (!isInArray)
      this.setState({ orders: [...this.state.orders, item] })
  }
}
```

```
export default App;
```

```
import React, { Component } from 'react'
```

```
export class Categories extends Component {
```

```
  constructor(props) {
```

```
    super(props)
```

```
    this.state = {
```

```
      categories: [
```

```
        {
```

```
          key: 'all',
```

```
          name: 'Bce'
```

```
        },
```

```
        {
```

```
          key: 'Boots',
```

```
        },
```

```
      ],
```

```
    }
  }
```

```
      name: 'Взуття'
    },
    {
      key: 'Accessories',
      name: 'Аксессуары'
    },
    {
      key: 'Equipment',
      name: 'Спорядження'
    },
    {
      key: 'Caps&Hats',
      name: 'Шапки'
    },
    {
      key: 'Jackets',
      name: 'Куртки'
    }
  ]
}
render() {
  return (
    <div className='categories'>
      {this.state.categories.map(el => (
        <div key={el.key} onClick={() => this.props.chooseCategory(el.key)}>{el.name}</div>
      ))}
    </div>
  )
}
```

```
export default Categories
```

```
import React from 'react'
```

```
export default function Footer() {
```

```
  return (
```

```
    <footer>
```

```
      ВКР Марценюка О.В. &copy;2023
```

```
    </footer>
```

```
  )
```

```
}
```

```
import React from 'react'
```

```
import { FaShoppingCart } from "react-icons/fa";
```

```
import Order from './Order';
```

```
const showOrders = (props, setCartOpen) => {
```

```
  let summa = 0
```

```
  props.orders.forEach(el => summa += Number.parseFloat(el.price))
```

```
  return (<div>
```

```

    {props.orders.map(el => (
      <Order onDelete={props.onDelete} key={el.id} item={el} />
    ))}
    <div className='summa'>
      Сумма: {new Intl.NumberFormat().format(summa)}$
      <button className='order_btn' onClick={() => {
        props.onShowModal();
        setCartOpen(false);
      }}>Оформити замовлення</button>
    </div>
  </div>
}

const showNothing = () => {
  return (<div className='empty'>
    <h2>Кошик порожній</h2>
  </div>)
}

export default function Header(props) {
  let [cartOpen, setCartOpen] = React.useState(false)
  return (
    <header>
      <div>
        <span className='logo'>ExploreZone</span>
        <ul className='nav'>
          <li>Каталог</li>
          <li>Доставка</li>
          <li>Оплата</li>
          <li>Про нас</li>
        </ul>
        <FaShoppingCart onClick={() => setCartOpen(cartOpen !== cartOpen)} className={`shop-cart-button ${cartOpen && 'active'} ` />
      </div>
      {cartOpen && (
        <div className='shop-cart'>
          {props.orders.length > 0 ?
            showOrders(props, setCartOpen) : showNothing()}
        </div>
      )}
    </div>
    <div className='presentation'></div>
  </header>
)
}

import React, { Component } from 'react'

```

```
export class Item extends Component {
  render() {
    return (
      <div className='item'>
        <img src={`./img/" + this.props.item.img} onClick={() => this.props.onShowItem(this.props.item)} />
        <h2>{this.props.item.title}</h2>
        <p>{this.props.item.desc}</p>
        <b>{this.props.item.price}&</b>
        <div className='add-to-cart' onClick={() => this.props.onAdd(this.props.item)}></div>
      </div>
    )
  }
}
```

export default Item

```
import React, { Component } from 'react'
import Item from './Item'
```

```
export class Items extends Component {
  render() {
    return (
      <main>
        {this.props.items.map(el => (
          <Item onShowItem={this.props.onShowItem} key={el.id} item={el} onAdd={this.props.onAdd} />
        ))}
      </main>
    )
  }
}
```

export default Items

```
import React, { Component } from 'react'
import { FaTrash } from 'react-icons/fa'
```

```
export class Order extends Component {
  render() {
    return (
      <div className='item'>
        <img src={`./img/" + this.props.item.img} />
        <h2>{this.props.item.title}</h2>
        <b>{this.props.item.price}&</b>
        <FaTrash className='delete-icon' onClick={() => this.props.onDelete(this.props.item.id)} />
      </div>
    )
  }
}
```

export default Order

```
import React, { useState } from 'react';
```

```
import Modal from 'react-modal';
import { Notify } from 'notiflix/build/notiflix-notify-aio';

const OrderModal = ({ onClose, cleanupBucket }) => {
  const [name, setName] = useState("");
  const [address, setAddress] = useState("");
  const [phoneNumber, setPhoneNumber] = useState("");
  const [email, setEmail] = useState("");
  const [paymentMethod, setPaymentMethod] = useState('card');

  const handleSubmit = (e) => {
    e.preventDefault();

    setName("");
    setAddress("");
    setPhoneNumber("");
    setPaymentMethod("");

    onClose();
    cleanupBucket();
    Notify.success("Замовлення успішно оформлено! Очікуйте дзвінка для підтвердження!");
  };

  return (
    <Modal isOpen={true} onRequestClose={onClose} className="modal">
      <button className="close-button" onClick={onClose}>
        <span>&times;</span>
      </button>
      <h2>Оформлення замовлення</h2>
      <form onSubmit={handleSubmit}>
        <div>
          <label>ПІБ:</label>
          <input
            type="text"
            value={name}
            className="modal_text_input"
            onChange={(e) => setName(e.target.value)}
            required
          />
        </div>

        <div>
          <label>Номер телефону:</label>
          <input
            type="text"
            value={phoneNumber}
            className="modal_text_input"
            onChange={(e) => setPhoneNumber(e.target.value)}
            required
          />
        </div>

        <div>
```

```
<label>Адреса доставки:</label>
<input
  type="text"
  value={address}
  className="modal_text_input"
  onChange={(e) => setAddress(e.target.value)}
  required
/>
</div>
<div>
<label>Email:</label>
<input
  type="text"
  value={email}
  className="modal_text_input"
  onChange={(e) => setEmail(e.target.value)}
  required
/>
</div>
<div>
<label>Метод оплати:</label>
<div className='radio_box'>
<label className='radio_label'>
<input
  type="radio"
  name="paymentMethod"
  value="card"
  checked={paymentMethod === "card"}
  onChange={(e) => setPaymentMethod(e.target.value)}
  required
/>
Карта
</label>
<br />
<label className='radio_label'>
<input
  type="radio"
  name="paymentMethod"
  value="cash"
  checked={paymentMethod === "cash"}
  onChange={(e) => setPaymentMethod(e.target.value)}
  required
/>
Готівкою при отриманні
</label>
</div>
</div>
<div className="modal-buttons">
<button type="submit">Оформити</button>
</div>
</form>
```

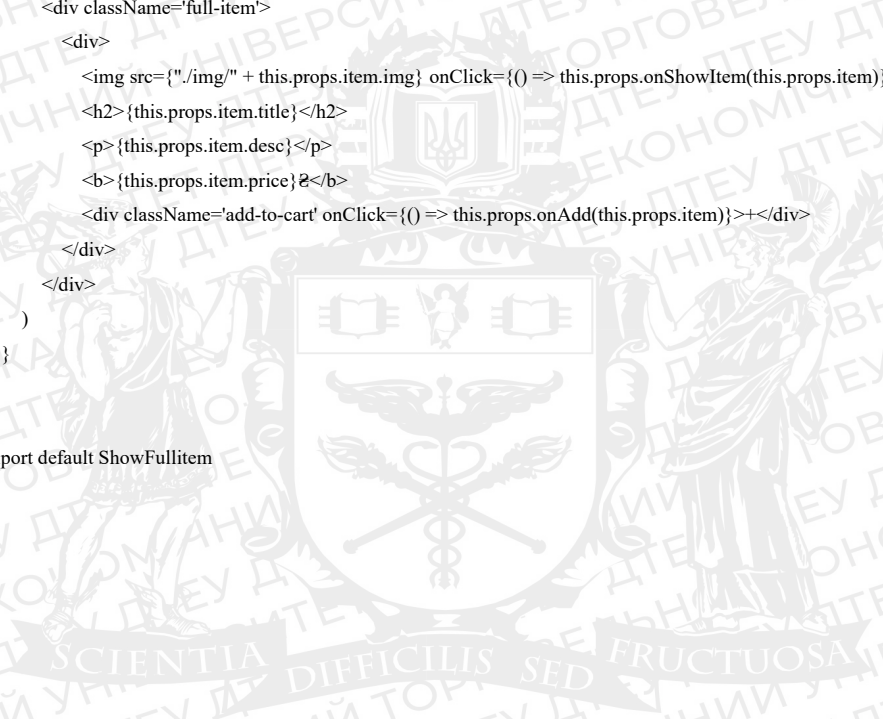
```
</Modal>
);
};

export default OrderModal;

import React, { Component } from 'react'

export class ShowFullitem extends Component {
  render() {
    return (
      <div className='full-item'>
        <div>
          <img src='"/img/" + this.props.item.img' onClick={() => this.props.onShowItem(this.props.item)} />
          <h2>{this.props.item.title}</h2>
          <p>{this.props.item.desc}</p>
          <b>{this.props.item.price}</b>
          <div className='add-to-cart' onClick={() => this.props.onAdd(this.props.item)}+</div>
        </div>
      </div>
    )
  }
}

export default ShowFullitem
```



## Продовження дод. А

```
runOnUiThread(new Runnable() {  
    public void run() {  
        _imageView.setImageDrawable(draw);  
    }  
});  
}  
private void SetTextInfo(final String str) {  
    runOnUiThread(new Runnable() {  
        public void run() {  
            _textViewInfo.setText(str);  
        }  
    }  
}
```

