

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Інститут електроенергетики  
(інститут)

Факультет інформаційних технологій  
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем  
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА  
кваліфікаційної роботи ступеня  
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента *Денисова Єгора Володимировича*  
(ПІБ)

академічної групи *122-19-2*  
(шифр)

спеціальності *122 Комп'ютерні науки*  
(код і назва спеціальності)

освітньої програми *Комп'ютерні науки*  
(назва освітньої програми)

на тему: *Розробка інтернет магазину з продажу одягу з  
використанням фреймворку React JavaScript*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи				
<b>розділів:</b>				
спеціальний	<i>доц. Кабак Л.В.</i>			
економічний	<i>проф. Вагонова О.Г.</i>			
<b>Рецензент</b>	<i>доц. Каштан В. Ю.</i>			
<b>Нормоконтролер</b>	<i>доц. Гуліна І.Г.</i>			

Дніпро  
2023

Міністерство освіти і науки України  
НТУ «Дніпровська політехніка»

**ЗАТВЕРДЖЕНО:**

завідувач кафедри  
програмного забезпечення комп'ютерних систем

(повна назва)

М.О. Алексєєв

(підпис)

(прізвище, ініціали)

«    »

2023 року

**ЗАВДАННЯ**

на кваліфікаційну роботу

*бакалавра*

(назва освітньо-кваліфікаційного рівня)

студента 122-19-2

(група)

Денисов Е.В.

(прізвище та ініціали)

тема кваліфікаційної роботи

Розробка інтернет магазину

*Розробка інтернет магазину з продажу одягу з використанням*

*фрейворку React JavaScript*

затверджена наказом ректора НТУ «ДП» від

16.05.2023

№ 350-с

Розділ	Зміст виконання	Термін виконання
Спеціальний	<i>На основі матеріалів проектно-технологічної практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i>	<u>13.05.2023 р.</u>
Економічний	<i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки</i>	<u>27.05.2023 р.</u>

Завдання видав

доц. Кабак Л.В.

(підпис)

(посада, прізвище, ініціали)

Завдання прийняв до виконання

Денисов Е.В.

(підпис)

(прізвище, ініціали)

Дата видачі завдання: 14.01.2023 р.

Термін подання кваліфікаційної роботи до ЕК: 12.06.2023 р.

## РЕФЕРАТ

Пояснювальна записка: 73 с., 19 рис., 5 дод., 24 джерел.

Об'єкт розробки: інтернет застосунок для продажі речей.

Мета дипломного проекту: створити веб-додаток для продажу товарів за допомогою модулів використовуючи React.

У вступі аналізується та уточнюється мета роботи, вказується на корисність цілей та наводиться обґрунтування актуальності теми. Також розкривається постановка завдання.

У першому розділі проводиться аналіз предметної області, визначається актуальність завдання та призначення розробки, формулюється постановка завдання та встановлюються вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі здійснюється аналіз існуючих рішень, обирається платформа для розробки, розробляється програма шляхом проектування і розробки, надається опис алгоритму та структури функціонування системи, визначаються вхідні та вихідні дані, наводяться характеристики технічних засобів, описується виклик та завантаження програмного застосунку, а також описується робота самої програми.

У економічному розділі визначається трудомісткість розробленої інформаційної підсистеми, проводиться підрахунок вартості роботи по створенню застосунку та розраховується час на його створення.

Список ключових слів: ВЕБ-ДОДАТОК, JAVASCRIPT, REACT, CSS, SCSS, HTML, JSON, DOM.

## ABSTRACT

Explanatory note: 73 pages, 19 pics, 5 apps, 24 sources.

Development object: an internet application for selling goods.

Objective of the diploma project: to create a web application for selling products using modules and utilizing React.

The introduction analyzes and specifies the objective of the work, highlights the usefulness of the objectives, and provides a justification for the relevance of the topic. The task formulation is also elaborated.

Chapter 1 conducts an analysis of the subject area, determines the relevance and purpose of the development, formulates the task, and establishes requirements for software implementation, technologies, and software tools.

Chapter 2 performs an analysis of existing solutions, selects the development platform, designs and develops the program, provides a description of the algorithm and system functioning structure, defines input and output data, presents technical characteristics of the tools, describes the application's invocation and loading, and outlines the program's operation.

The economic section determines the labor intensity of the developed information subsystem, calculates the cost of application development, and estimates the time required for its creation.

Keywords: web application, JavaScript, React, CSS, SCSS, HTML, JSON, DOM.

## ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА	
ЗАДАЧІ..... 10	
1.1 Загальні відомості с предметної галузі.....	10
1.2 Призначення розробки та область застосування.....	11
1.3 Підстава для розробки.....	11
1.4 Постановка завдання.....	13
1.5 Вимоги до програми або програмного виробу.....	14
1.5.1 Вимоги до функціональних характеристик .....	14
1.5.2 Вимоги до складу та параметрів технічних засобів.....	14
1.5.3 Вимоги інформаційної та програмної сумісності.....	15
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ	
СИСТЕМИ..... 16	
2.1 Функціональне призначення системи.....	16
2.2 Опис застосованих математичних методів.....	16
2.3 Опис використаних технологій та мов програмування.....	17
2.4 Опис структури системи та алгоритмів її функціонування.....	21
2.5 Обґрунтування та організація вхідних та вихідних даних програми.....	29

2.6	Опис роботи розробленої системи.....	29
2.6.1	Використані технічні засоби.....	29
2.6.2	Використані програмні засоби.....	30
2.6.3	Виклик та завантаження програми.....	30
2.6.4	Опис інтерфейсу користувача.....	30
	РОЗДІЛ 3. ЕКОНОМІКА.....	42
3.1	Розрахунок трудомісткості та вартості розробки програмного продукту .....	42
3.2	Рахунок витрат на створення програми.....	45
	ВИСНОВКИ.....	47
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48
	Додаток А. Код програми.....	49
	Додаток Б. Відзив керівника дипломного проекту.....	70
	Додаток В. Рецензія.....	71
	Додаток Г. Відзив керівника економічного розділу.....	72
	Додаток Д. Перелік файлів на диску.....	73

## СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

HTML – HyperText Markup Language.

ЕОМ – електронно-обчислювальна машина.

API – інтерфейс програмування додатків.

CSS – Cascading Styles Sheets.

SCSS – Syntactically Awesome Style Sheets

DOM – об'єктна модель документа.

HTTP – HyperText Transfer Protocol

HTTPS – HyperText Transfer Protocol Secure

JS – JavaScript

## ВСТУП

Зараз у кожного бізнесу використовується веб версія цього бізнесу яка має велике значення для розвитку та продажі будь якого товару, особливу одягу.

Інтернет-магазин дозволяє підприємствам розширити свій охоплюваний ринок та збільшити кількість клієнтів за рахунок доступності товарів в будь-який час та в будь-якому місці.

Окрім того, створення власного сайту з продажу товарів дозволяє бізнесу більш ефективно керувати своїм іміджем та брендом, дозволяючи клієнтам отримати більш повну інформацію про товари та послуги, які надає компанія.

З іншого боку, наявність власного інтернет-магазину може допомогти зменшити витрати на маркетинг та рекламу, оскільки дозволяє бізнесу прямо комунікувати зі своїми клієнтами та пропонувати їм персоналізовані пропозиції та знижки.

Крім того, наявність інтернет-магазину також дозволяє бізнесу отримати доступ до різноманітних аналітичних даних про поведінку користувачів, що дозволяє покращити ефективність маркетингових кампаній та оптимізувати пропозиції продуктів та послуг.

Отже, створення власного сайту з продажу товарів є важливим елементом для розвитку бізнесу та може допомогти підприємствам збільшити свій обсяг продажів та підвищити рівень впізнаваності бренду.

Додам, що наявність власного інтернет-магазину також дозволяє бізнесу знизити витрати на зберігання та управління запасами товарів, оскільки дозволяє ефективно контролювати інвентаризацію та керувати запасами за допомогою спеціальних програмних засобів.

Крім того, наявність власного інтернет-магазину дозволяє бізнесу пропонувати широкий асортимент товарів, що дозволяє клієнтам зручно та



швидко знайти потрібний товар, порівняти його з іншими варіантами та зробити відповідний вибір.

З іншого боку, відсутність власного інтернет-магазину може призвести до втрати бізнесу на користь конкурентів, які пропонують свої товари та послуги через інтернет. Більше того, сьогодні покупці все частіше віддають перевагу онлайн-покупкам, тому відсутність власного інтернет-магазину може призвести до зниження продажів та втрати позицій на ринку.

Отже, можна сказати, що наявність власного інтернет-магазину стала необхідністю для бізнесу в сучасних умовах. Це дозволяє бізнесу підвищити свою конкурентоспроможність, збільшити свій обсяг продажів та знизити витрати на управління бізнесом.

Метою дипломного проекту є створення веб-додатку, а точніше веб магазину який давав би змогу звичайному користувачеві купувати одяг тощо. За основу взятий React.js фреймворк. Все буде рендеритись на стороні клієнта.

Беручи це все до уваги було сформовано тему дипломної роботи: «Розробка інтернет магазину з продажу одягу з використанням фреймворку React JavaScript.»

# РОЗДІЛ 1

## АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1 Загальні відомості с предметної галузі

Предметна галузь – це область знань, яка вивчається в рамках певної дисципліни або наукового напрямку. У випадку інтернет-магазинів, предметною галуззю є електронна комерція (e-commerce) - це процес купівлі та продажу товарів і послуг через Інтернет.

Інтернет-магазини можуть пропонувати різноманітний асортимент товарів, від побутової техніки та електроніки до одягу та аксесуарів. Для забезпечення роботи інтернет-магазинів використовуються спеціальні платформи та програмні засоби, які дозволяють здійснювати операції з каталогами товарів, оформленням замовлень, оплатою, доставкою, обліком запасів та іншими аспектами електронної комерції.

Інтернет-магазини можуть бути різних розмірів та масштабів - від невеликих підприємств, які продають товари лише в одній країні, до міжнародних гігантів, які мають клієнтів по всьому світу. За останні роки інтернет-магазини значно зросли у популярності, оскільки це дозволяє зручно та швидко купувати товари не виходячи з дому.

У сучасних умовах, де пандемія COVID-19 спричинила поширення карантинних обмежень, інтернет-магазини стали ще більш актуальними для споживачів та бізнесів. Крім того, розвиток технологій та зменшення вартості доступу до Інтернету дозволяє забезпечити зростання кількості онлайн-покупок та розвитку інтернет-магазинів в майбутньому.

Для успішної роботи інтернет-магазинів необхідно мати не тільки якісний продукт, але й правильно зорієнтовану маркетингову стратегію, яка дозволяє привернути нових клієнтів та зберегти існуючих. До складу маркетингової

стратегії входить реклама в соціальних мережах, контекстна реклама, email-маркетинг, SMM-маркетинг та інші інструменти просування бренду.

Окрім цього, для успішної роботи інтернет-магазину дуже важлива швидкість та зручність роботи сайту, його безпека та надійність. Тому розробники та власники інтернет-магазинів активно використовують сучасні технології, такі як React фреймворк, для створення високопродуктивних та функціональних веб-додатків.

З огляду на те, що конкуренція в галузі електронної комерції дуже висока, важливо використовувати сучасні технології та інструменти для привернення нових клієнтів та збереження існуючих. Також важливо не забувати про важливість якості обслуговування та підтримки клієнтів, оскільки задоволені клієнти є найкращою рекламою для будь-якого бізнесу, в тому числі й інтернет-магазинів. [18]

## **1.2 Призначення розробки та область застосування.**

Темою бакалаврської дипломної роботи виступає: «Розробка інтернет магазину з продажу одягу з використанням фреймворку React JavaScript.». В якості функціональної основи взятий React фреймворк. Тобто головною метою роботи є створення веб-сайту, який представляє собою інтернет магазин для підтримки бізнеса та продажу товарів.

Основними критеріями розроблювального веб-додатку є:

1. Функціональність: Додаток повинен виконувати всі завдання, що від нього очікуються, і бути простим у використанні.
2. Надійність: Додаток має працювати стабільно і безперебійно, навіть при великому навантаженні.
3. Безпека: Додаток має бути захищеним від можливих атак, таких як віруси, хакерські атаки і т.д.

4. Швидкість: Додаток має завантажуватися швидко і працювати швидко, щоб користувачі не втратили терпіння і не перейшли до конкурентів.

5. Сумісність: Додаток повинен бути сумісним з різними браузерами та платформами.

6. Простота оновлення: Додаток має бути легко оновлювати без збоїв і простоїв в роботі.

7. Масштабованість: Додаток повинен бути готовим до збільшення обсягів роботи з часом і можливості розширення функціоналу.

8. Підтримка: Додаток має мати надійну підтримку та забезпечувати користувачів з необхідною допомогою у разі потреби.

9. Відкритий код: Відкритий код дозволяє розробникам та користувачам додатку вносити зміни та вдосконалювати його за потребою.

### **1.3 Підстава для розробки**

Відповідно до ОПП, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу (проект). Тема роботи узгоджується з керівником проекту, випускаючою кафедрою, та затверджується наказом ректора.

Отже, підставами для розробки (виконання кваліфікаційної роботи) є:

- наказ ректора Національного технічного університету «Дніпровська політехніка» № 350-с від 16.05.2023 р;
- ОПП за напрямом підготовки 122 «Комп'ютерні науки»;
- графік навчального процесу та навчальний план;
- завдання на дипломний проект на тему «Розробка інтернет магазину з продажу одягу з використанням фреймворку React JavaScript».

## 1.4 Постановка завдання

Постановка завдання на розробку веб-додатку для інтернет-магазину:

Замовник: Компанія, що займається продажем різних товарів в Інтернеті.

Мета: Розробка веб-додатку для інтернет-магазину, який забезпечить зручний і безпечний процес покупки та продажу товарів, а також забезпечить потрібну функціональність та надійність для користувачів.

### Опис завдання:

1. Розробити дизайн веб-додатку з урахуванням корпоративного стилю.
2. Розробити функціональність для додавання, редагування та видалення товарів.
3. Розробити функціональність для розрахунку вартості товарів, включаючи податки.
4. Розробити функціональність для реалізації оплати товарів різними способами, забезпечивши безпеку та надійність платежів.
5. Розробити функціональність для забезпечення зручного та швидкого пошуку товарів.
6. Розробити функціональність для забезпечення зручного та швидкого оформлення замовлення.
7. Забезпечити масштабованість веб-додатку, щоб забезпечити його працездатність при збільшенні обсягів роботи.
8. Забезпечити можливість розширення функціоналу веб-додатку з часом.

## **1.5. Вимоги до програми або програмного виробу.**

### **1.5.1. Вимоги до функціональних характеристик.**

Функціональні характеристики веб-додатку повинні включати:

1. Можливість перегляду каталогу товарів з детальною інформацією про кожен товар.
2. Можливість реалізації знижок та промокодів для залучення нових клієнтів та збереження старих.
3. Можливість зв'язку з підтримкою користувачів та вирішення питань щодо товарів та доставки.
4. Можливість відгуків користувачів про товари та послуги.
5. Можливість встановлення інформаційних повідомлень для користувачів про новини, акції та знижки.
6. Можливість адміністрування веб-додатку з боку власника бізнесу, включаючи зміну цін, редагування контенту та інші функції.
7. Можливість підтримки міжнародного ринку шляхом розширення валют та мов, в яких доступний веб-додаток.

### **1.5.2 Вимоги до складу та параметрів технічних засобів.**

Технічні засоби є важливою складовою веб-додатку для інтернет-магазину. Нижче наведено деякі вимоги до складу та параметрів технічних засобів:

1. Сервер: сервер, на якому розміщується веб-додаток, повинен мати достатній рівень продуктивності та надійності. Для цього необхідно використовувати потужний процесор, достатньо об'єму оперативної пам'яті, а також використовувати швидкий твердотілий накопичувач.

2. База даних: база даних повинна мати достатній рівень продуктивності та надійності. Для цього необхідно використовувати потужний сервер бази даних, а також оптимізувати запити до бази даних.

3. Клієнтські комп'ютери: веб-додаток повинен бути оптимізований для використання на різних типах клієнтських комп'ютерів та мобільних пристроях. Для цього необхідно використовувати відповідні технології розробки, які забезпечують коректну роботу на різних платформах.

4. Мережеве забезпечення: веб-додаток повинен мати відповідну мережеву інфраструктуру, яка забезпечує надійну та швидку передачу даних між клієнтами та сервером.

5. Засоби безпеки: технічні засоби повинні мати вбудовані засоби безпеки, які забезпечують захист від різних видів кібератак, вірусів та шкідливих програм. [15, 16]

### **1.5.3 Вимоги інформаційної та програмної сумісності.**

Однією з ключових вимог при розробці веб-додатків є їх інформаційна та програмна сумісність з різними операційними системами та браузерами. Для забезпечення широкої аудиторії користувачів, веб-додатки повинні працювати стабільно та ефективно на різних операційних системах, таких як Windows, macOS, Linux тощо.

Також вимоги до програмної сумісності включають сумісність з різними версіями браузерів, такими як Google Chrome, Mozilla Firefox, Safari. Важливо

враховувати, що різні браузері можуть по-різному інтерпретувати код, тому необхідно тестувати веб-додаток на різних браузерах та їх версіях. [6]

## **РОЗДІЛ 2**

### **ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ**

#### **2.1. Функціональне призначення системи**

Метою даного проекту є розробка веб-магазину, основним метою якого буде онлайн продаж одягу та не тільки, для клієнтів.

Ця інформаційна система буде надавати зручний та доступний інтерфейс для клієнтів, де вони зможуть переглядати асортимент товарів, робити покупки та здійснювати оплату онлайн. Основними функціями системи будуть:

Перегляд каталогу товарів: Клієнти зможуть швидко переглядати доступні товари, використовуючи категорії. Вони зможуть переглядати детальну інформацію про кожен товар, таку як опис, ціна, розміри, кольори тощо.

Оформлення замовлення: Клієнти зможуть заповнити форму з доставковою інформацією, обрати метод оплати та підтвердити замовлення.

Оплата: Система буде підтримувати різні методи оплати, такі як кредитні карти, електронні гаманці, платіжні системи тощо. Клієнти зможуть обрати зручний для них метод оплати та ввести необхідні дані для оплати.

#### **2.2. Опис застосованих математичних методів**



Математичні методи не було використано, оскільки розв'язування задачі не передбачає застосування математичних методів, при розробці такого додатку математичні методи не застосовувалися.

### **2.3. Опис використаних технологій та мов програмування**

Застосунок має бути реалізовано на мові програмування JavaScript з використанням фреймворку React. Також використовував Sass модулі.

Що таке JavaScript:

JavaScript – це високорівнева, інтерпретована мова програмування, яка використовується для розробки веб-додатків та взаємодії з користувачем на стороні клієнта (в браузері). Вона є однією з найпоширеніших мов програмування веб-розробки і грає важливу роль у створенні динамічних та інтерактивних веб-сторінок.

JavaScript була розроблена компанією Netscape у 1995 році і вперше впроваджена в браузері Netscape Navigator. З тих пір вона значно еволюціонувала і стала стандартом для браузерної розробки. Сьогодні JavaScript підтримується всіма сучасними браузерами і використовується як для розробки клієнтських додатків, так і для серверної розробки за допомогою платформи Node.js.

JavaScript є мовою, орієнтованою на об'єкти, що означає, що програма складається з об'єктів, які взаємодіють між собою, шляхом виклику функцій і доступу до їхніх властивостей. Вона підтримує багато основних конструкцій програмування, таких як умови (if-else), цикли (for, while), функції, масиви та багато іншого.

Основна сфера застосування JavaScript – це веб-розробка. Вона дозволяє створювати динамічні елементи на веб-сторінках, реагувати на події користувача, валідувати дані, взаємодіяти з сервером за допомогою AJAX-запитів, анімувати елементи та багато іншого. За допомогою JavaScript також

можна розробляти мобільні додатки за допомогою фреймворків, таких як React Native та Ionic.

Загалом, JavaScript є потужним і гнучким інструментом для веб-розробки, який дозволяє створювати інтерактивні та захоплюючі додатки для вебу. JavaScript має багато вбудованих функцій і методів, які спрощують роботу з різними аспектами веб-розробки. Наприклад, вона надає можливість маніпулювати HTML-елементами і CSS-стилями, змінювати вміст веб-сторінок, обробляти події миші та клавіатури, виконувати асинхронні запити до сервера та багато іншого.

Крім того, JavaScript підтримує концепцію функціонального програмування і об'єктно-орієнтованого програмування. Вона дозволяє оголошувати функції, передавати їх як аргументи, повертати їх як значення і працювати з замиканнями. Об'єкти в JavaScript можуть мати властивості і методи, і їх можна використовувати для організації коду в логічні блоки і капсулювання пов'язаної функціональності.

JavaScript також підтримує велику кількість сторонніх бібліотек і фреймворків, які розширюють її можливості і полегшують розробку. Наприклад, jQuery допомагає спростити маніпуляцію DOM-елементами та роботу з AJAX-запитами, а Angular, React і Vue.js надають потужні інструменти для створення складних односторінкових додатків (SPA).

Нарешті, варто відзначити, що JavaScript не обмежується лише браузерною розробкою. За допомогою платформи Node.js, JavaScript може бути використана для створення серверних додатків і виконання коду на стороні сервера. Це дозволяє розробникам використовувати одну мову програмування як на клієнтській, так і на серверній стороні, що спрощує розробку та підтримку програмного забезпечення. [21]

Загалом, JavaScript є незамінним інструментом у світі веб-розробки, який дозволяє створювати потужні, динамічні та інтерактивні веб-додатки, які можуть працювати на різних пристроях та платформах. Вона продовжує

розвиватись і ставати ще потужнішою з кожним релізом, надаючи розробникам безліч можливостей для творчості та інновацій.

Що таке React.js:

React.js - це бібліотека JavaScript, яка використовується для побудови користувацьких інтерфейсів. Вона була розроблена компанією Facebook і використовується для створення веб-додатків з високою ефективністю і швидкодією.

Основна концепція React.js полягає в розбитті користувацького інтерфейсу на невеликі компоненти. Кожен компонент представляє собою частину веб-сторінки і має свої властивості та стан. Властивості передаються в компонент для визначення його вигляду та поведінки, а стан використовується для зберігання і оновлення даних в компоненті. Коли властивості або стан змінюються, React автоматично перерендерює компонент, оновлюючи відображення на сторінці.

React також використовує віртуальний DOM (Document Object Model) для оптимізації оновлення сторінки. Вміст сторінки представляється віртуальним DOM-деревом, яке зберігається в пам'яті. При зміні стану або властивостей компонента React порівнює віртуальний DOM з реальним DOM і знаходить мінімальний набір змін, які потрібно внести до реального DOM для оновлення сторінки. Це дозволяє покращити продуктивність додатка, оскільки оновлення відбуваються тільки в необхідних місцях, а не на всій сторінці.

React також має багато інших корисних функцій, таких як управління подіями, маршрутизація, управління формами і багато іншого. Він широко використовується в індустрії розробки веб-додатків і є одним з найпопулярніших інструментів для побудови користувацького інтерфейсу. [19]

Окрім основних концепцій, які я вже згадував, React також надає ряд інших функціональностей, що роблять його потужним інструментом для розробки веб-додатків.

Одна з цих функціональностей – це компонентний підхід до розробки. Замість писання великого монолітного коду, React дозволяє розбити його на

невеликі ізольовані компоненти. Це полегшує розуміння, підтримку і перевикористання коду. Компоненти можна вкладати один в одного, створюючи ієрархію компонентів і забезпечуючи зручну організацію коду.

React також має можливість використовувати JSX (JavaScript XML) - розширення JavaScript, яке дозволяє описувати структуру інтерфейсу у вигляді розмітки, схожої на HTML. JSX робить код більш зрозумілим і легким для написання та читання. React компілює JSX в JavaScript, що дозволяє використовувати його в браузерах безпосередньо.

Іншою важливою особливістю React є підхід "одностороннього потоку даних" (one-way data flow). Це означає, що дані рухаються вниз по ієрархії компонентів від батьківських компонентів до дочірніх. Це спрощує управління станом додатка та відстеження потоку даних.

React також підтримує відокремлення логіки та представлення. Зазвичай, логіка компонента може бути розділена від його візуального представлення, що дозволяє зручно маніпулювати даними і змінювати вигляд компонента незалежно одне від одного. [3, 5, 8, 9]

Що таке Sass:

Sass (Syntactically Awesome Style Sheets) - це метамова на мові CSS, яка надає розширений функціонал і покращує ефективність написання стилів для веб-сайтів. Sass дозволяє розробникам використовувати змінні, вкладені селектори, функції та інші функціональні можливості, які не доступні в звичайному CSS.

Основна перевага Sass полягає в його здатності до розширення і організації коду CSS. Він дозволяє використовувати змінні для зберігання значень, таких як кольори, розміри шрифтів, відступи та інші. Це дозволяє змінити значення змінної в одному місці і автоматично оновити всі місця, де вона використовується. [10]

Крім того, Sass дозволяє вкладати селектори, що спрощує структурування CSS-коду. За допомогою вкладених селекторів можна описувати стилі для

певних елементів, які знаходяться в межах інших елементів. Це допомагає уникнути повторення коду і полегшує зміну стилів в майбутньому.

Sass також надає функції, які дозволяють виконувати математичні операції, маніпулювати кольорами та працювати з рядками. Він підтримує умовні оператори, цикли та імпорт інших Sass-файлів, що дозволяє організувати код у більш структурованій та зручній формі.

Однак, важливо зазначити, що Sass не може працювати без спеціального процесора. Вихідний код Sass треба компілювати до звичайного CSS, який потім використовується веб-браузером. Існують різні інструменти та засоби збірки, такі як Sass Compiler або засіб збірки Gulp, які забезпечують автоматичну компіляцію Sass у CSS.

Узагальнюючи, Sass – це метамова на основі CSS, яка додає багато корисних функціональних можливостей, таких як змінні, вкладені селектори та функції. Він полегшує розробку та підтримку стилів веб-сайтів, допомагаючи зберегти час і зусилля розробника. [4]

## **2.4. Опис структури системи та алгоритмів її функціонування**

З часом Інтернет зазнав змін і перетворився на більш інтерактивне та динамічне середовище, де користувачі взаємодіють з веб-сторінками та додатками. Вже не маємо лише статичних веб-сторінок, які просто відображають інформацію. Замість цього, сучасні сайти реагують на дії користувача, обробляють запити та надають результати.

Для створення таких інтерактивних і динамічних сайтів часто використовується архітектурний патерн, відомий як MVC (Model-View-Controller). Цей патерн допомагає організувати структуру додатка, розділивши його на три основні компоненти: модель (Model), представлення (View) і контролер (Controller).[1]

Модель відповідає за управління даними та бізнес-логікою додатка. Вона зберігає дані, обробляє їх і забезпечує потрібні функції для взаємодії з даними.

Представлення відповідає за візуальне відображення даних користувачеві. Воно відповідає за структуру та вигляд веб-сторінки або додатка, включаючи HTML-розмітку, CSS-стилі та інші елементи, які стосуються зовнішнього вигляду.

Контролер відповідає за обробку запитів користувача та управління взаємодією між моделлю та представленням. Він приймає запити від користувача, виконує необхідні дії, взаємодіє з моделлю для отримання потрібних даних та оновлює представлення згідно з результатами.

Така архітектура дозволяє розділити логіку додатка на легко управляючі компоненти, що спрощує розробку, підтримку та розширення веб-додатків. Завдяки MVC інтерактивність та динамічність сайтів стають можливими, а користувачам надається більш потужна та приємна веб-досвід.

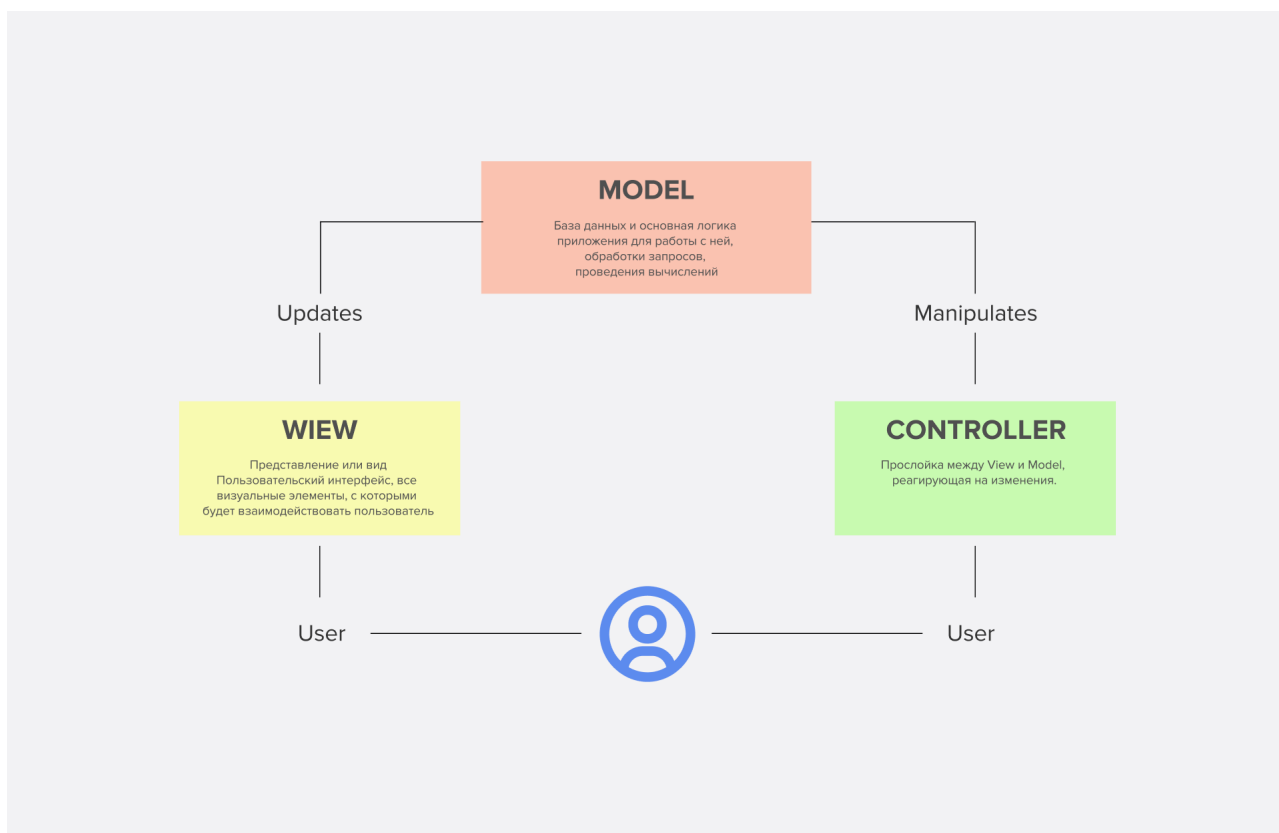


Рис. 2.1 – Структура MVC додатку

У 2011 році розробники Facebook стикалися з проблемою, коли користувачі бачили повідомлення в чаті, але коли натискати на них, нових повідомлень не відображалося. Це призводило до постійних скарг від користувачів, і команда розробників намагалась вирішити цю проблему. Однак, кожне рішення вирішувало лише конкретний випадок, і незалежно від зусиль розробників, проблема поверталася. Ця проблема отримала назву "повідомлення-зомбі".

Проблема полягала в тому, що архітектура Facebook, яка використовувала MVC, стала крихкою. З часом додаток ставав все складнішим, і кількість моделей та представлень зростає. Взаємодія між ними стала складнішою для контролю. Крім того, код, який відповідав за взаємодію між моделями та представленнями, був переважно імперативним і мав тенденцію до поломок.

Для вирішення цих проблем була розроблена альтернативна архітектура під назвою Flux.

Flux – це тип архітектури, який Facebook використовує у сполученні з бібліотекою React. Вона відповідає за управління станом даних у JavaScript-додатках і для розробки серверної частини веб-додатків. Flux доповнює компоненти візуалізації (View) в React за допомогою одностороннього потоку даних.

Flux можна розглядати як не просто шаблон або фреймворк, а як архітектурний підхід, який має чотири основні компоненти (більш детально будуть розглянуті пізніше):

- Диспетчер (Dispatcher).
- Сховище (Stores).
- Відображення (Views) (компоненти React).
- Дії (Actions).

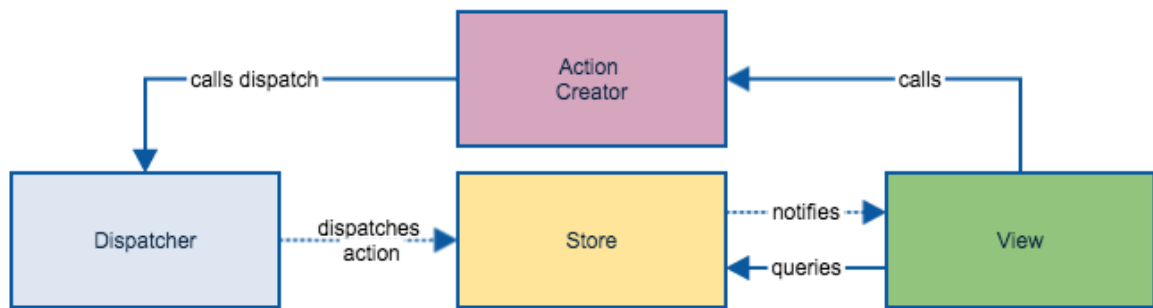


Рис. 2.2 – Структура Flux архітектури

Flux є відкритим вихідним кодом і може бути розглянутий як шаблон проектування, а не фреймворк, що дозволяє використовувати його безпосередньо. Flux допомагає зробити код більш передбачуваним порівняно з фреймворками MVC. Розробники можуть створювати додатки, не турбуючись про складні взаємодії між джерелами даних. Одна з ключових особливостей Flux полягає в односпрямованому потоці даних, що дозволяє краще організувати потік даних. У Flux зміни поширюються через диспетчер даних та дії, що спрощує взаємодію з користувачем.

Flux має вимогливий до потоку даних підхід. Диспетчер даних установлює строгі правила та виключення для управління потоком, що відрізняється від підходу MVC, де потоки реалізуються по-різному. У Flux односпрямований потік даних, тоді як у MVC потік є двонаправленим. У Flux всі зміни просочуються через диспетчер даних, і Store не може бути безпосередньо зміненим, так само як і інші дії. Зміни повинні пройти через диспетчер та дії. У Flux, на відміну від MVC, можна моделювати окремі об'єкти у Store для зберігання будь-яких даних, пов'язаних з додатком. [2]

Односпрямований потік даних у Flux розв'язує проблеми двонаправленого потоку даних. У MVC контролер має велику відповідальність за збереження стану та дані програми, що призводить до складності та помилок. У односпрямованому потоці даних Flux стан додатку передбачуваний, оскільки зміни в слої відображення впливають на дані, а не навпаки.

Стандартна архітектура Flux має такі компоненти:

- Дії (Actions) - допоміжні функції, які передають дані у диспетчер.



- Диспетчер (Dispatcher) - отримує дії та передає корисну навантаження зареєстрованим зворотним викликам (callbacks).

- Сховища (Stores) – діють як контейнери для стану програми та логіки. Робота додатку відбувається в сховищах, які прослуховують дії від диспетчера та оновлюють відображення.

- Відображення контролерів (Controller Views) – компоненти React, які отримують стан зі сховищ та передають його дочірнім компонентам.

Для створення веб-додатка та його архітектури необхідно завантажити всі необхідні пакетні файли. У даному випадку був реалізований додаток з використанням React, тому для використання цих технологій потрібно встановити Node.js, який має в собі пакетний менеджер npm. За допомогою npm можна завантажити всі необхідні пакети для роботи з React. [12, 13, 14]

Структура файлів веб-додатка, зображена на візуальному плануванні задач, виглядає наступним чином:

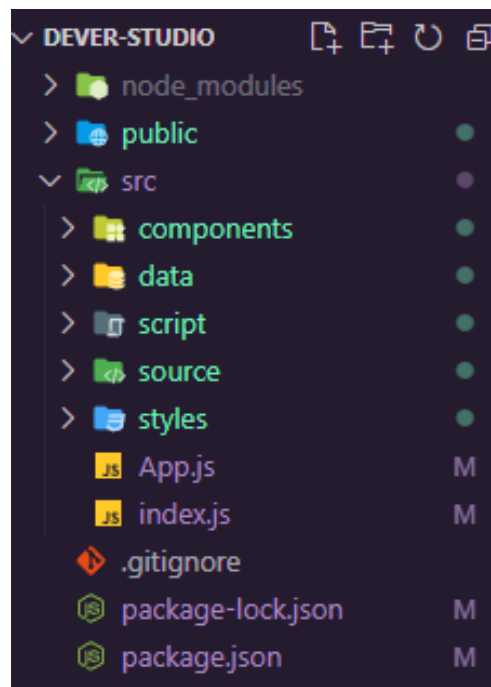


Рис. 2.3 – Файлова структура проекту

У папку `node_modules` зберігаються всі необхідні файли для роботи з React.

Сам додаток знаходиться у папці `src`, яка розділена на декілька підпапок, таких як `data`, `components`, `scripts`, `source`, `styles`. Кожна з цих папок має свою функціональність:

- Папка `scripts` містить файли - функції, що виконують певні дії або події у додатку.

- Папка `components` містить компоненти - перефразування веб-елементів, які використовуються для створення користувацького інтерфейсу додатку.

- Папка `data` містить файл, який генерує `item` для рендеру на сторінці.

- Папка `source` містить файли та фото, які рендерить наш веб-додаток.

- Папка `styles` містить файли, в яких визначаються стилі - налаштування вигляду додатка.

Розглянемо детально структуру `scripts`:

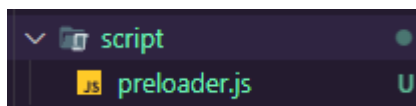


Рис. 2.4 – Структура `scripts`

`Scripts` має файл:

- `preloader.js` – файл робить імітацію завантаження сторінки коли отримує данні.

Розглянемо детально структуру `components`:

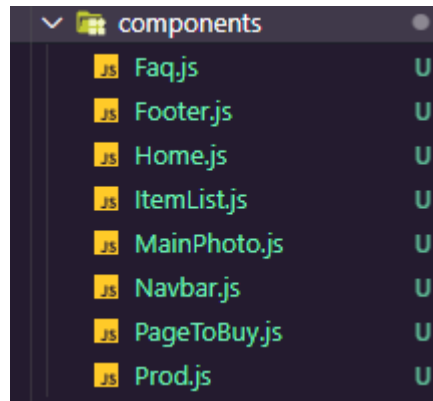


Рис. 2.5 – Структура components

Components має файли:

- Faq.js – файл з інформацією по різних питанням.
- Footer.js – файл який відноситься до футера проекту.
- Home.js – файл який є головною сторінкою проекту.
- ItemList.js – файл який генерує продукцію на сторінці.
- MainPhoto.js – файл який генерує задній фон на сторінках.
- Navbar.js – файл який відноситься до хедера проекту.
- PageToBuy.js – файл який генерує сторінку для придбання товару.
- Prod.js – файл який обгортає інші файли які генерують сторінку.

Розглянемо детально структуру data:

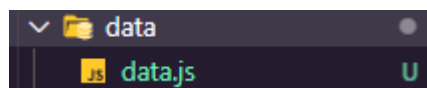


Рис. 2.6 – Структура data

Маємо один файл data.js - який в свою чергу має уся інформацію про товари які генеруються на сайті.

Розглянемо детально структуру source:

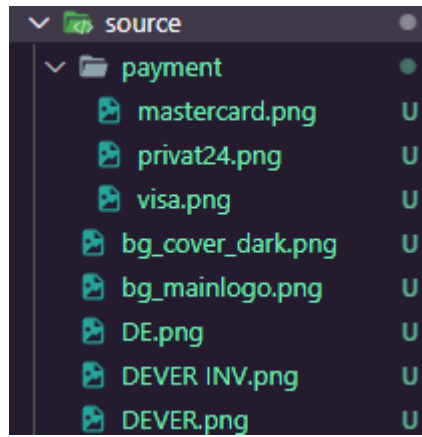


Рис. 2.7 – Структура source

Маємо багато файлів фото, які є типовими логотипами і задніми фоном.

Розглянемо детально структуру styles:

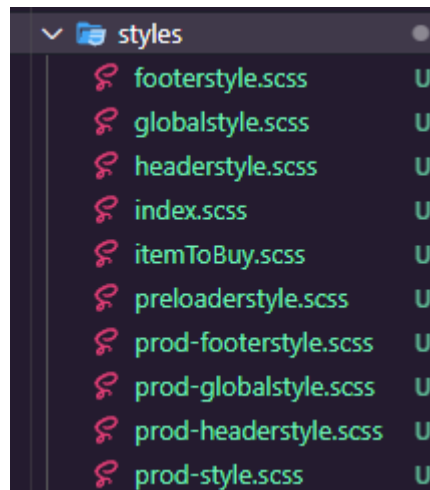


Рис. 2.8 – Структура styles

Styles має файли:

- footerstyle.scss - файл стилів для футера.
- globalstyle.scss - файл глобальних стилів.

- headerstyle.scss - файл стилів для хедера.
- index.scss - файл який угрупує усі стилі.
- itemToBuy.scss - файл для стилів на сторінці одного товару.
- preloaderstyle.scss - файл для відображення завантаження сторінки.
- prod-footerstyle.scss - файл стилів для футеру товару.
- prod-globalstyle.scss - файл глобальних стилів для товару.
- prod-headerstyle.scss - файл стилів для хедера товару.
- prod-style.scss - файл для угрупування стилів для самого товару.

Таким чином, додаток розділений на різні модулі, що дозволяє зручно відстежувати зміни в системі. Кожен файл відповідає за конкретну функціональність, що спрощує внесення змін до проекту. Файл index.js, що знаходиться в папці src, є точкою входу в веб-додаток.

## **2.5. Обґрунтування та організація вхідних та вихідних даних програми**

Веб-додаток отримує дані з форм, які заповнюються користувачем веб-компонентів, а також з джерел даних, що прикріплені до компонентів. [22]

## **2.6. Опис роботи розробленої системи**

### **2.6.1. Використані технічні засоби**

Нижче наведені характеристики є рекомендованими для клієнтського обладнання, оскільки вся інформація зберігається локально на стороні клієнта:

- Процесор Intel(R) Core(TM) i3 з тактовою частотою 2.0GHz.
- Оперативна пам'ять об'ємом 4 ГБ.
- Монітор з діагоналлю не менше 15 ".
- Мінімум 1 ГБ вільного місця на диску.
- Миша та клавіатура.
- Доступ до Інтернету.

Ці характеристики є рекомендованими, що означає, що при використанні обладнання з такими характеристиками, розроблений додаток буде забезпечувати необхідну надійність та швидкість обробки даних.

### **2.6.2. Використані програмні засоби**

Веб-додаток реалізований з використанням мови програмування JavaScript та залежить від бібліотеки React. Ці технології є достатніми для обробки даних на стороні клієнта. Для стилізації компонентів було використано SCSS.

Для коректної роботи додатка на стороні клієнта рекомендується мати наступні програмні засоби:

- Веб-браузер, такий як Google Chrome.
- Операційна система Windows 10 або вище.
- Швидкий інтернет.

### **2.6.3. Виклик та завантаження програми**

Додаток призначений для використання в онлайн-режимі і вимагає наявності веб-браузера з підтримкою JavaScript для його коректної роботи. На даний момент додаток розроблений для особистого використання, але робочу версію можна запустити, написавши `npm start` у консолі. [7]

#### **2.6.4. Опис інтерфейсу користувача**

Для того щоб розпочати роботу з додатком, потрібно запустити сам білд через `npm start` у консолі. Потім користувача буде направлено на головну сторінку сайту. Але це тільки для того щоб переконатися, що сайт працює, інакше треба білдити все в продакшн версію. Головна сторінка сайти виглядає наступним чином:

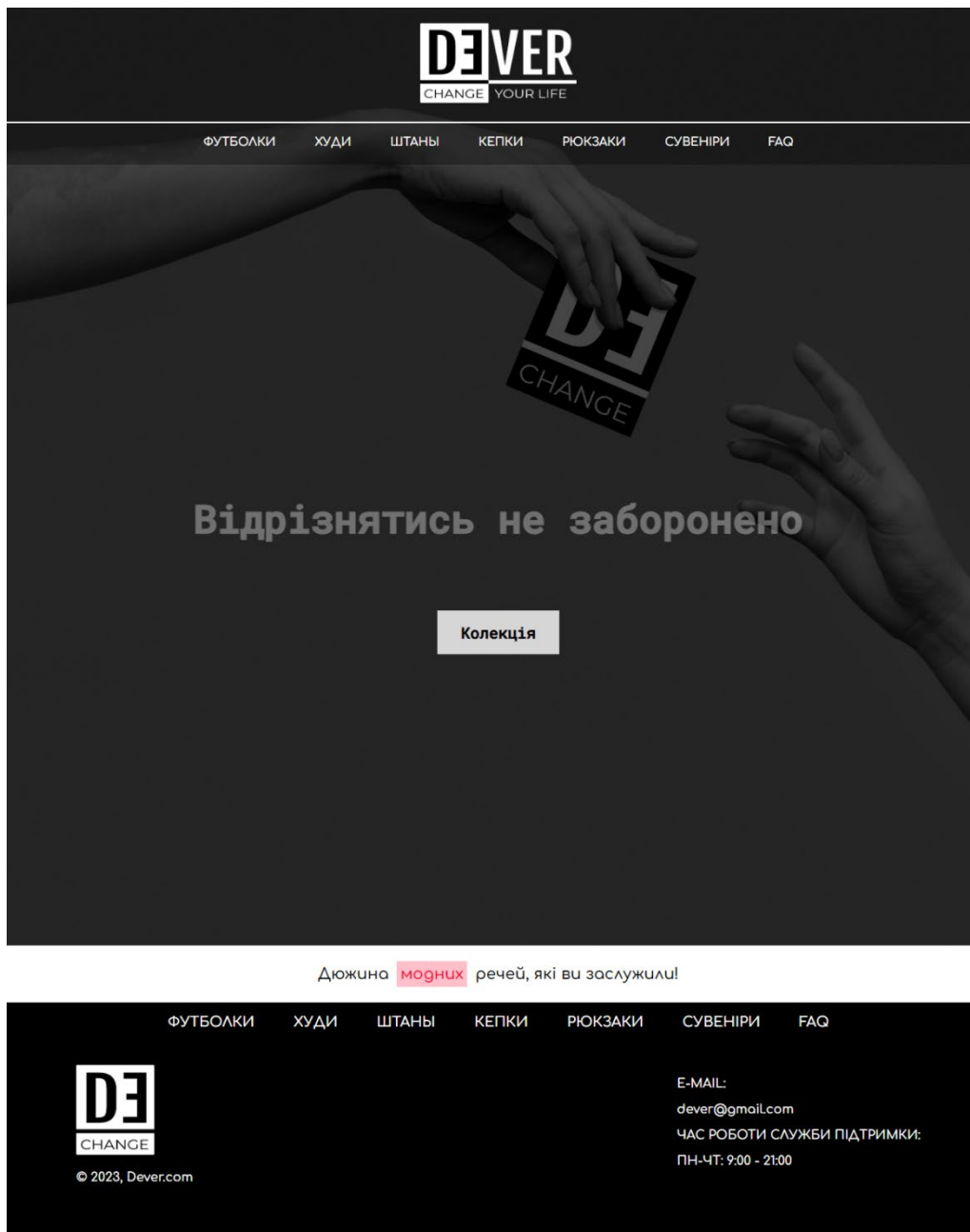


Рис. 2.9 – Головна сторінка

Натиснувши на “Колекція” нас направить до товарів:



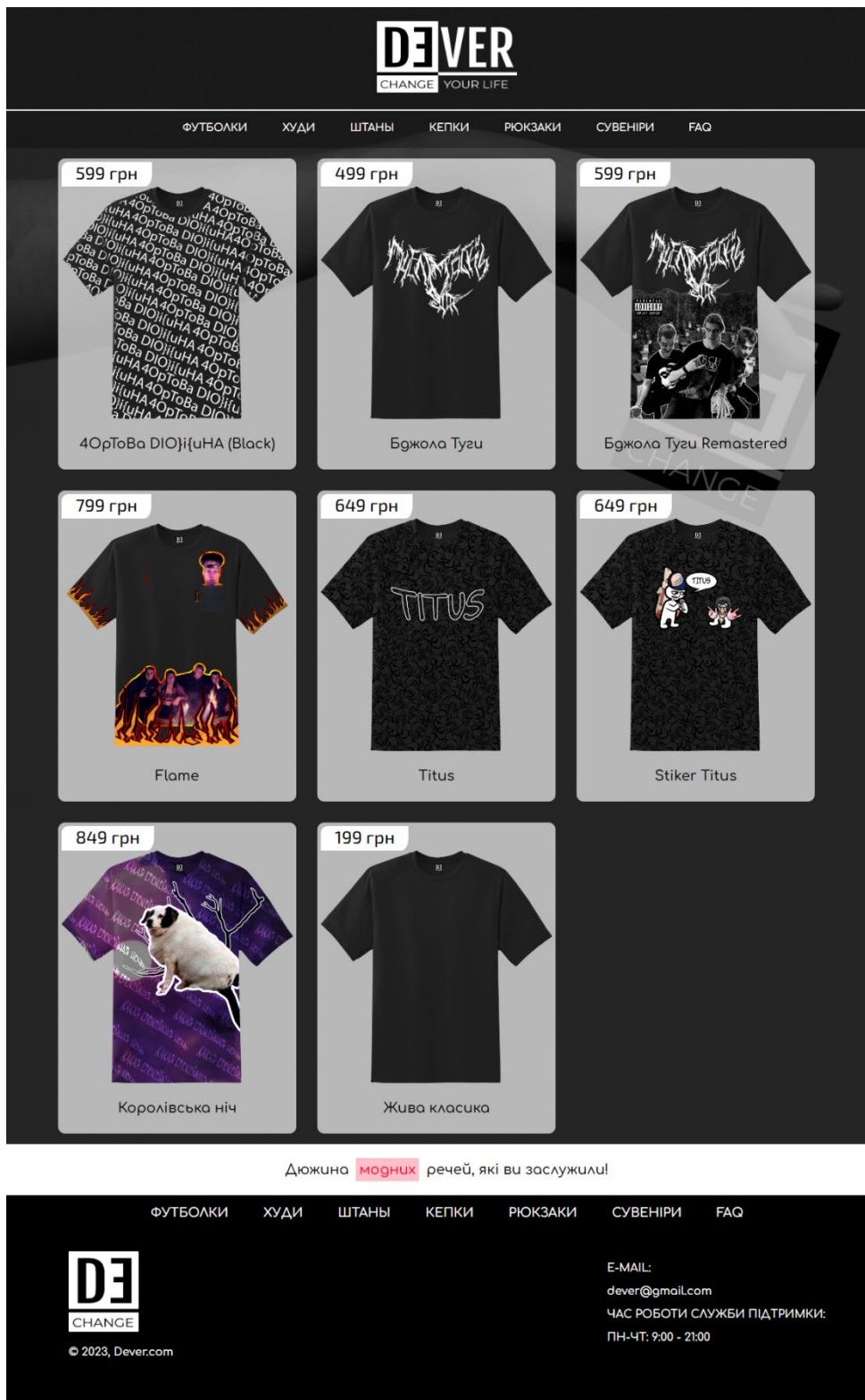


Рис. 2.10 – Каталог товарів

Ми також можемо вибрати по категоріям зверху та знизу, наприклад можемо перейти на “кепки”:

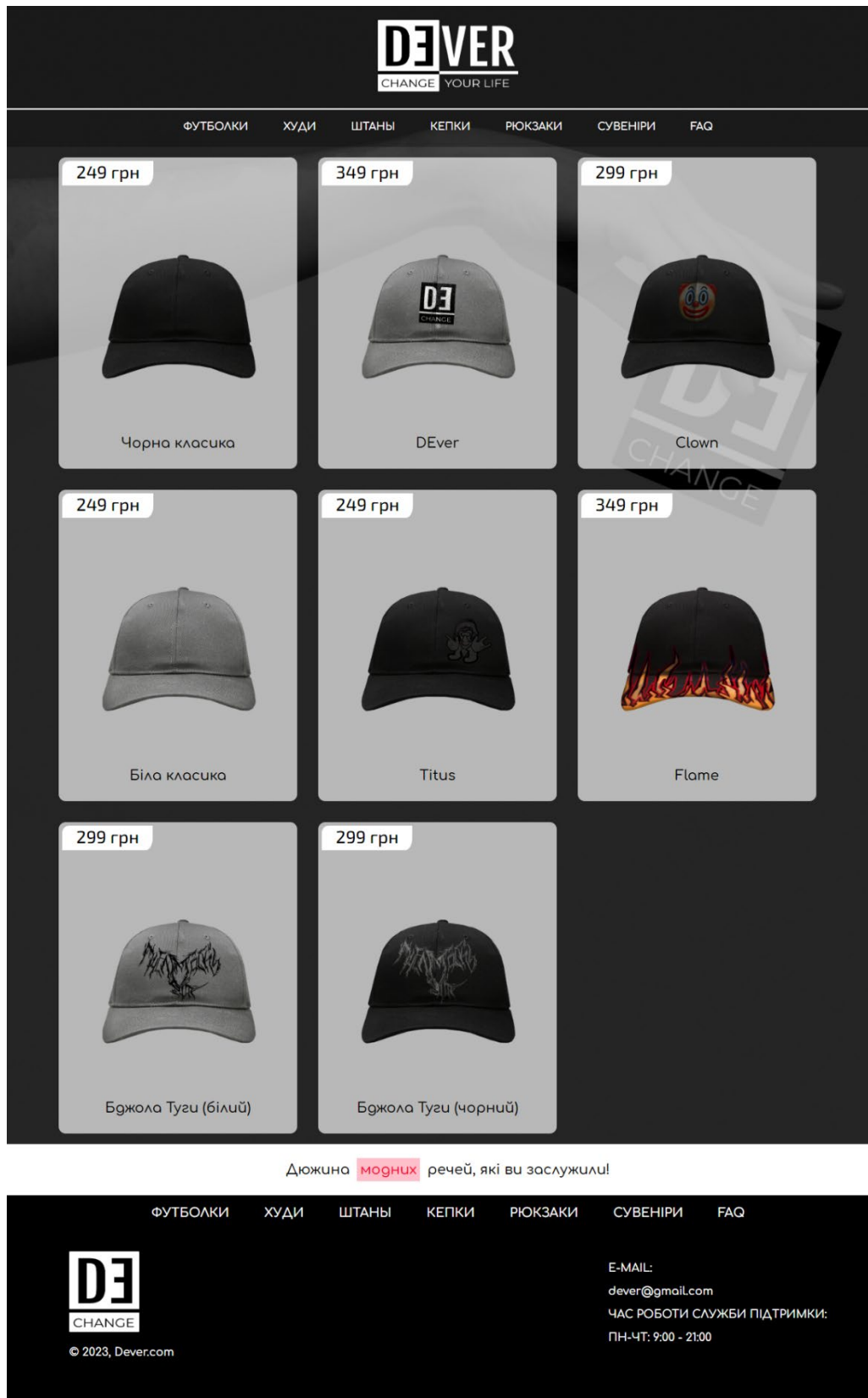




Рис. 2.11 – Каталог “Кепок”

А також інша продукція:




ФУТБОЛКИ
ХУДИ
ШТАНЫ
КЕПКИ
РЮКЗАКИ
СУВЕНІРИ
FAQ

799 грн




40pToBa DIOjifihHA (Black)

699 грн




Бджола Тузу

799 грн




Бджола Тузу Remastered

999 грн




Flame

849 грн




Titus

849 грн




Stiker Titus

1099 грн



Королівська ніч


399 грн



Жива класика

Дюжина модних речей, які ви заслужили!

ФУТБОЛКИ
ХУДИ
ШТАНЫ
КЕПКИ
РЮКЗАКИ
СУВЕНІРИ
FAQ



© 2023, Dever.com

E-MAIL:  
dever@gmail.com  
ЧАС РОБОТИ СЛУЖБИ ПІДТРИМКИ:  
ПН-ЧТ: 9:00 - 21:00

Рис. 2.12 – Каталог “Худи”

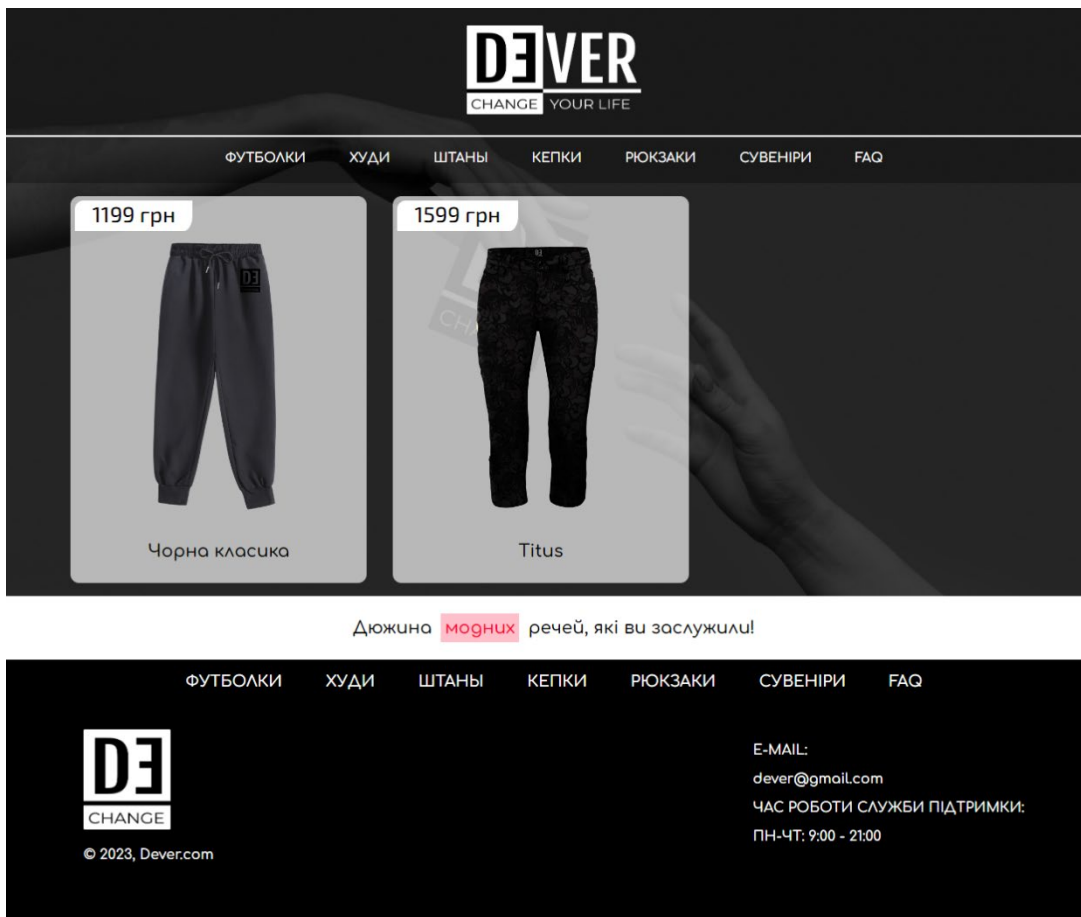




Рис. 2.13 Каталог штанів




ФУТБОЛКИ
ХУДИ
ШТАНЫ
КЕПКИ
РЮКЗАКИ
СУВЕНІРИ
FAQ

1499 грн




Чорна класика

1999 грн




Бджола Туго

1999 грн




Titus DEver

1799 грн




Flame

2199 грн




Titus

1999 грн




Stiker Titus

2199 грн




Frogus

1799 грн



DEver

Дюжина **модних** речей, які ви заслужили!



© 2023, Dever.com

E-MAIL:  
dever@gmail.com  
ЧАС РОБОТИ СЛУЖБИ ПІДТРИМКИ:  
ПН-ЧТ: 9:00 - 21:00

Рис. 2.14 – Каталог рюкзаків

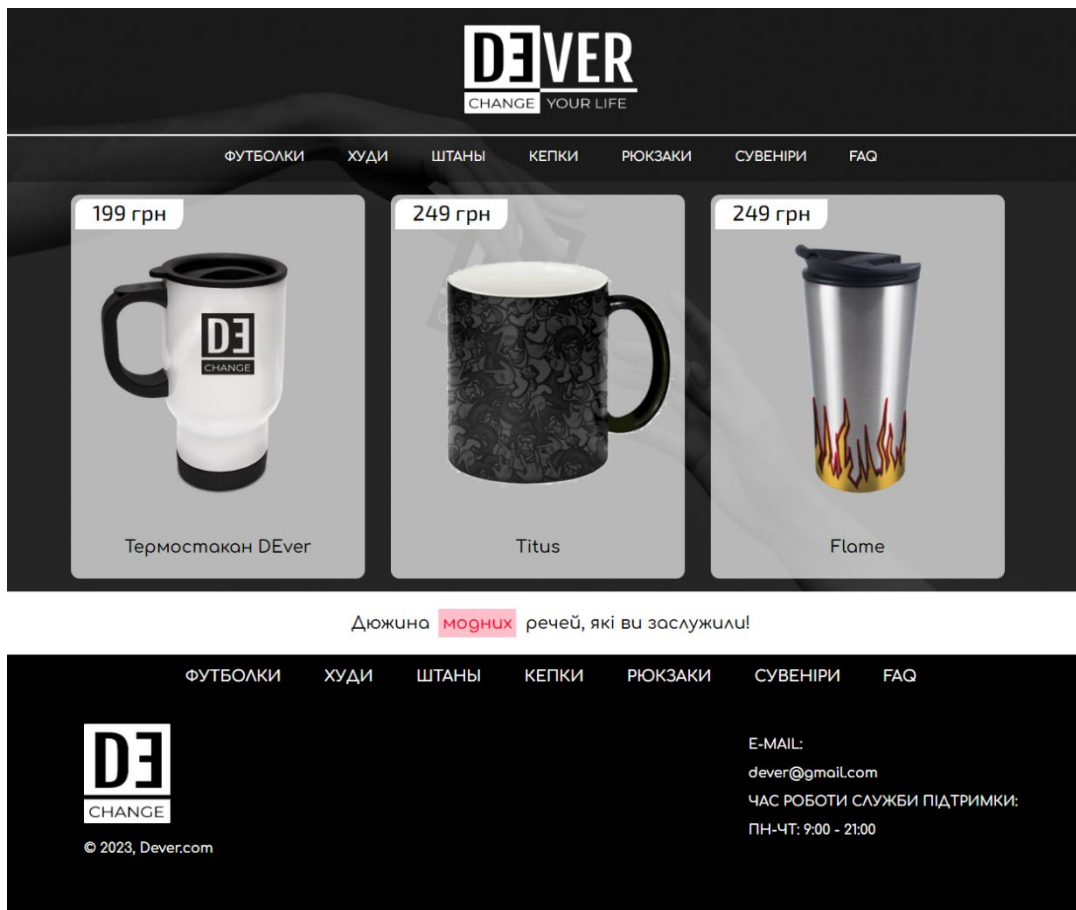


Рис. 2.15 Каталог сувенірів



Також є сторінка інформації для клієнтів, на неї можливо перейти якщо натиснути на “Faq”:

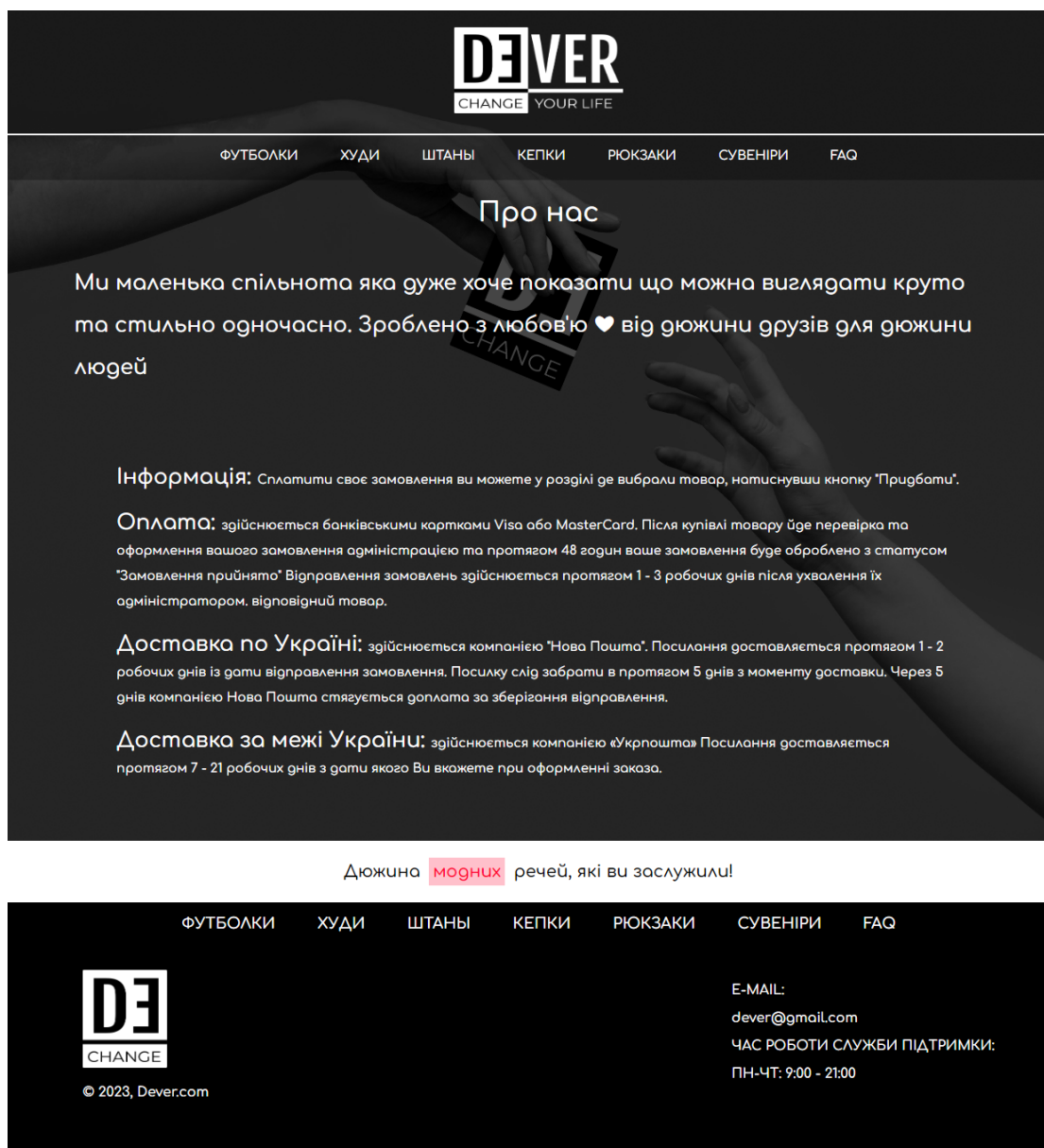


Рис. 2.16 – Сторінка інформації

Якщо треба купити товар, просто натискаємо на нього та відкривається його сторінка:

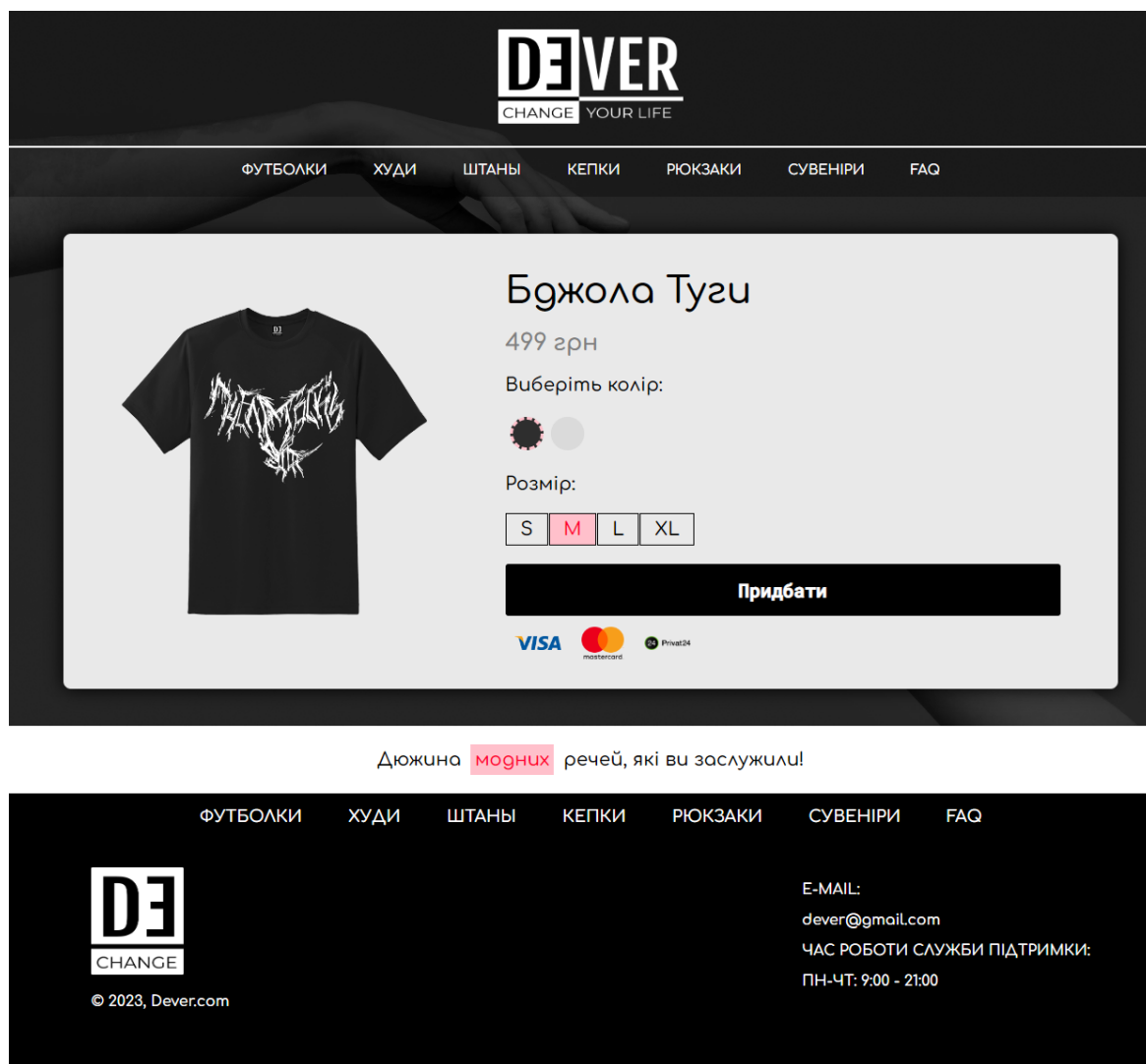
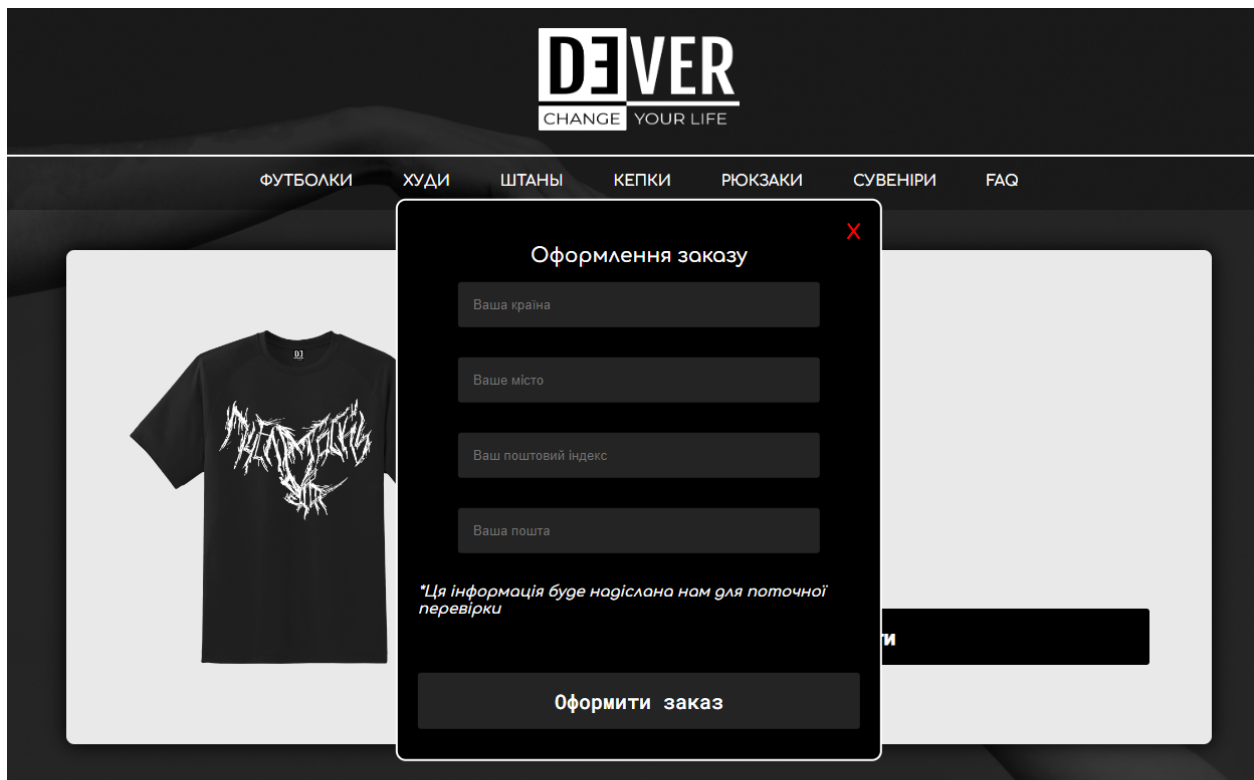


Рис. 2.17 – Сторінка певного товару

Якщо треба придбати його, натискаємо “Придбати” де у нас питають куди відправляти товар, а далі знімає кошти через сервіс.





Дюжина **можних** речей, які ви заслужили!

Рис. 2.18 – Оформлення заказу

Також при натисканні на розділи відпрацьовує preloader.js:

[20]

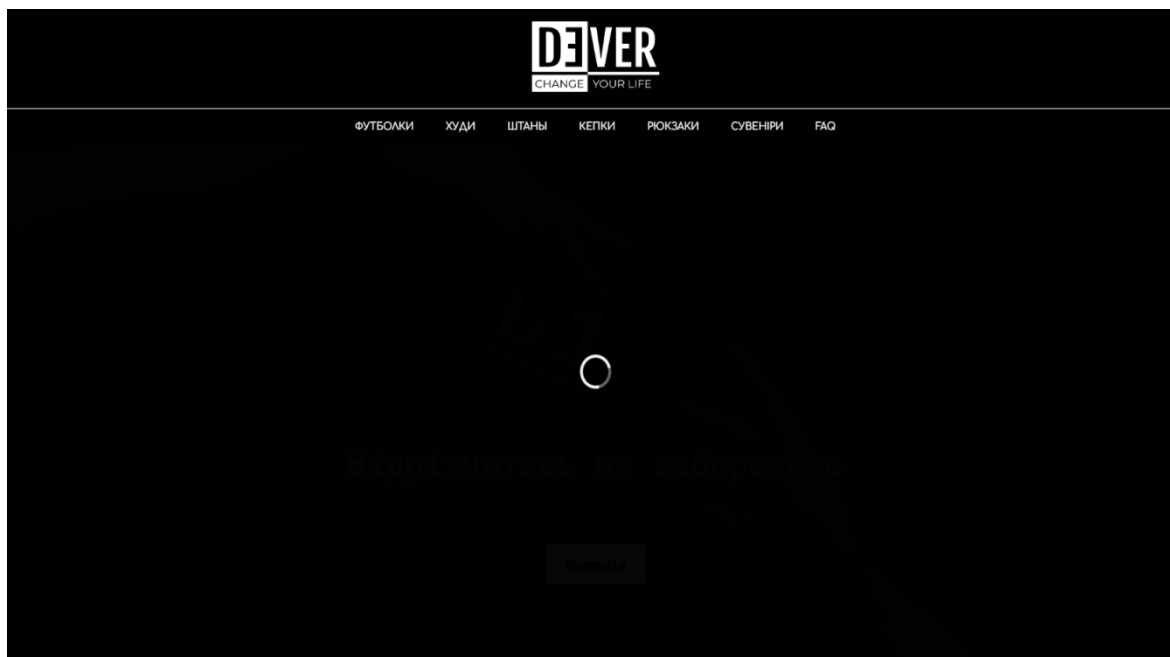


Рис. 2.19 – Preloader

## РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ

### 3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Вхідні дані:

- передбачуване число операторів – 2200;
- коефіцієнт складності програми – 1.4;
- коефіцієнт корекції програми в ході її розробки – 0,1;
- годинна заробітна плата програміста, грн / год – 150.

Годинна заробітна плата front-end розробника була вирахована виходячи з даних «Української спільноти програмістів (DOU)». Середньоукраїнська заробітна плата такого розробника дорівнює 1000 доларів США у місяць. При курсі валют НБУ на початок червня 2023 року один американський долар дорівнює 35,55 грн, тому середня зарплата в гривнях дорівнює 35000 грн. При 8-ми годинному робочому дні (176 робочих годин місяць в середньому) середня заробітна плата за годину буде становити приблизно ~ 170 грн. [11]

- коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,2;
- коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,5;
- вартість машино-години ЕОМ, грн/год – 15.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_{\partial}, \text{ людино-годин,} \quad (3.1)$$

де  $t_o$  – витрати праці на підготовку та опис поставленої задачі,

$t_u$  – витрати праці на дослідження алгоритму рішення задачі,

$t_a$  – витрати праці на розробку блок-схеми алгоритму,  
 $t_n$  – витрати праці на програмування за блок-схемою,  
 $t_{oml}$  – витрати праці на налагодження програми на ЕОМ,  
 $t_d$  – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \times C \times (1 + p), \text{ людино-годин,} \quad (3.2)$$

де  $q$  – передбачуване число операторів,

$C$  – коефіцієнт складності програми,

$p$  – коефіцієнт кореляції програми в ході її розробки.

$$Q = 2200 \times 1,4 \times (1 + 0,1) = 3388 \text{ людино-годин}$$

Для розрахунку витрат праці на дослідження алгоритму рішення задачі треба використати формулу:

$$t_u = \frac{QB}{(75 \dots 85)k}, \text{ людино-годин,} \quad (3.3)$$

де  $B$  – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі, ( $B = 1.2 \dots 1.5$ ),

$k$  – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності.

Витрати праці на розробку алгоритму рішення задачі:

$$t_u = \frac{2200 \cdot 1,2}{80 \cdot 1,5} = 22, \text{ людино-годин} \quad (3.4)$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20...25) \cdot k}, \text{ людино-годин,} \quad (3.5)$$

$$t_n = \frac{2200}{22 \cdot 1,5} = 66,6 \text{ людино-години}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20...25)k}, \text{ людино-годин,} \quad (3.6)$$

$$t_n = \frac{2200}{23 \cdot 1,5} = 63,7 \text{ людино-годин}$$

Витрати праці на налагодження програми на ЕОМ:

$$t_{oml} = \frac{Q}{(4..5) \cdot k}, \text{ людино-годин,} \quad (3.7)$$

$$t_{oml} = \frac{2200}{5 \cdot 1,5} = 293,3 \text{ людино-годин}$$

Витрати праці на підготовку документації:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \text{ людино-годин,} \quad (3.8)$$

де  $t_{\partial p}$  – трудомісткість підготовки матеріалів та рукописів,

$$t_{\partial p} = \frac{Q}{(15..20) \cdot k}, \text{ людино-годин,} \quad (3.9)$$

$$t_{\partial p} = \frac{2200}{18 \cdot 1,5} = 81,4 \text{ людино-годин}$$

$t_{до}$  – трудомісткість редагування, друку та оформлення документації.

$$t_{до} = 0,75 \cdot t_{др}, \text{ людино-годин,} \quad (3.10)$$

$$t_{до} = 0,75 \cdot 81,4 = 61,05 \text{ людино-годин}$$

Отримуємо трудомісткість розробки програмного забезпечення:

$$t = 61,05 + 81,4 = 142,45 \text{ людино-годин} \quad (3.11)$$

### 3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ  $K_{ПО}$  включають витрати на заробітну плату виконавця програми  $З_{зп}$  і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{ПО} = З_{зп} + З_{мв}, \text{ грн,} \quad (3.12)$$

Заробітна плата виконавців визначається за формулою:

$$З_{зп} = t \cdot C_{пр}, \text{ грн,} \quad (3.13)$$

де  $t$  – загальна трудомісткість, людино-годин,

$C_{пр}$  – середня годинна заробітна плата програміста, грн/година.

$$З_{зп} = 1097 \cdot 170 = 186490 \text{ грн.} \quad (3.14)$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$З_{MB} = t_{омл} \cdot C_M, \text{ грн}, \quad (3.15)$$

де  $t_{омл}$  – трудомісткість налагодження програми на ЕОМ, год,

$C_M$  – вартість машино-години ЕОМ, грн/год.

Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення АСУП.

$$З_{MB} = 620 \times 15 = 9300 \text{ грн}. \quad (3.16)$$

$$К_{по} = 186490 + 9300 = 195790 \text{ грн}. \quad (3.17)$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс.}, \quad (3.18)$$

де  $B_k$  – число виконавців,

$F_p$  – місячний фонд робочого часу (при 40 годинному робочому тижні  $F_p=176$  годин).

$$T = \frac{1097}{1 \cdot 176} = 6.2 \text{ міс.} \quad (3.19)$$

Висновок: трудомісткість розробленого додатку: 1097 люд-год, вартість роботи по створенню програми 195790 грн. та час на його створення 6,2 міс.

## **ВИСНОВКИ:**

Отже, розробка веб-додатків для інтернет-магазинів є важливим етапом в створенні успішного бізнесу в онлайн-середовищі. Вимоги до функціональності веб-додатків включають розширені можливості керування товарами та замовленнями, зручну сплату та доставку товарів, а також забезпечення безпеки та конфіденційності інформації про користувачів.

Для розробки веб-додатків в інтернет-магазинах широко використовуються фреймворки, такі як React, що дозволяють прискорити розробку та забезпечити високу продуктивність та якість веб-додатків. Однак, важливо враховувати вимоги до інформаційної безпеки та програмної сумісності, щоб забезпечити коректну та безперебійну роботу веб-додатка на різних операційних системах та браузерах.

Узагальнюючи, розробка веб-додатків для інтернет-магазинів має великий потенціал для розвитку бізнесу в онлайн-середовищі та дозволяє забезпечити зручний та безпечний досвід покупок для користувачів.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. MVC як шаблон проєктування: <https://brander.ua/technologies/mvc>
2. Fluxxor and the Flux Architecture: <https://softwareengineeringdaily.com/2015/09/20/fluxxor-and-the-flux-architecture-with-brandon-4.tilley/>
3. React documentation: <https://uk.legacy.reactjs.org/>
4. Sass documentation: <https://sass-lang.com/>
5. Node documentation: <https://nodejs.org/en>
6. Сумісність сайтів та браузерів: <https://sebweo.com/krosbrauzerna-sumisnist-sajtu-scho-tse-i-yak-yiyi-pereviriti/>
7. Референс сайту: <https://www.reserved.com/ua/uk/>
8. React: <https://blog.hubspot.com/website/react-js>
9. React Hooks: <https://dou.ua/lenta/articles/react-hooks-guide/>
10. Css: <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>
11. Salary Dou: <https://jobs.dou.ua/salaries/?period=2022-12&position=Junior%20SE>
12. JavaScript: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
13. Що таке JavaScript: <https://uk.wikipedia.org/wiki/JavaScript>
14. Чому JS перспективна мова: <https://dou.ua/forums/topic/35184/>
15. Архітектура веб-додатків: <https://codeguida.com/post/1394>
16. UI/UX дизайн: <https://dan-it.com.ua/uk/blog/vse-pro-profesiju-ui-ux-dizajnera/>
17. Методології розробок: <https://agile.yakubovsky.com/ua/2015/10/9-osnovnykh-metodolohiy-rozrobky-prohramnoho-zabezpechennya-agile/>
18. Методології: <https://wezom.com.ua/ua/blog/metodologija-razrobotki-programnogo-obespechenija>
19. Порівняння React з іншими фреймворками: <https://dou.ua/forums/topic/39933/>
20. Анімації в JS: [https://developer.mozilla.org/en-US/docs/Web/API/Canvas\\_API/Tutorial/Basic\\_animations](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Basic_animations)
21. Користувацькі скрипти: [https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%BA%D1%96%D0%BF%D0%B5%D0%B4%D1%96%D1%8F:%D0%9A%D0%BE%D1%80%D0%B8%D1%81%D1%82%D1%83%D0%B2%D0%B0%D1%86%D1%8C%D0%BA%D1%96\\_%D1%81%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%B8](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%BA%D1%96%D0%BF%D0%B5%D0%B4%D1%96%D1%8F:%D0%9A%D0%BE%D1%80%D0%B8%D1%81%D1%82%D1%83%D0%B2%D0%B0%D1%86%D1%8C%D0%BA%D1%96_%D1%81%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%B8)
22. Netscape: <https://isp.netscape.com/>
23. WWW: <https://www.techspot.com/article/788-world-wide-web-25-year-timeline/>
24. HTTP: <https://ru.wikipedia.org/wiki/HTTP>

## КОД ПРОГРАМИ

```

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

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

//App.js
import React from "react";
import {BrowserRouter, Route, Routes} from 'react-router-dom'
import './styles/index.scss'
import './styles/prod-footerstyle.scss'
import './styles/prod-globalstyle.scss'
import './styles/prod-headerstyle.scss'
import './styles/prod-style.scss'
import { Home } from "./components/Home";
import { Navbar } from "./components/Navbar";
import { Prod } from "./components/Prod";
import { Footer } from "./components/Footer";
import { Faq } from "./components/Faq";
import { PageToBuy } from "./components/PageToBuy";

function App() {
  return (
    <BrowserRouter>
      <Navbar />
      <div className="container">
        <Routes>
          <Route path="/" element={<Home/>} />
          <Route path='/ts' element={<Prod type={"ts"}/>} />
          <Route path='/hoodie' element={<Prod type={"hoodie"}/>} />
          <Route path='/cap' element={<Prod type={"cap"}/>} />
          <Route path='/bag' element={<Prod type={"bag"}/>} />
          <Route path='/pants' element={<Prod type={"pants"}/>} />
          <Route path='/souvenir' element={<Prod type={"souvenir"}/>} />
          <Route path='/faq' element={<Faq/>} />
          <Route path='/product/:type/:name/:id/:img/:price' element={<PageToBuy />} />
        </Routes>
      </div>
      <Footer />
    </BrowserRouter>
  );
}

export default App;

//Prod.js
import React, { Fragment } from "react";
import { ItemList } from "./ItemList";
import { Looper } from "../script/preloader";

export const Prod = (type) => {
  return (
    <Fragment>

```

```

    <Looper />
    <div style={{display:'flex', flexDirection:'row', flexWrap:'wrap'}}>
      <ItemList type={type} />
    </div>
  </Fragment>
)
}
//PageToBuy.js
import React, { useEffect, useRef, useState } from 'react';
import { useParams } from 'react-router-dom';
import Visa from './source/payment/visa.png'
import Mastercard from './source/payment/mastercard.png'
import Privat24 from './source/payment/privat24.png'

export const PageToBuy = () => {
  const { type, name, id, img, price } = useParams();
  const [isModalOpen, setIsModalOpen] = useState(false);
  const [currentSize, setCurrentSize] = useState('M');
  const modalRef = useRef(null);

  const toggleSize = (size) => {
    if (currentSize === size) {
      setCurrentSize(null);
    } else {
      setCurrentSize(size);
    }
  };

  // eslint-disable-next-line react-hooks/exhaustive-deps
  const toggleModal = () => {
    setIsModalOpen(!isModalOpen);
  };

  useEffect(() => {
    const handleOutsideClick = (event) => {
      if (modalRef.current && !modalRef.current.contains(event.target)) {
        toggleModal();
      }
    };

    document.addEventListener('mousedown', handleOutsideClick);

    return () => {
      document.removeEventListener('mousedown', handleOutsideClick);
    };
  }, [toggleModal]);

  return (
    <div className='wrap'>
      <div className='img_card img_card_item'>
        <img src={`/images/${type}/${img}`} alt="item" key={id} />
      </div>
      <div className='info_wrap'>
        <div className='title_item'>{name}</div>
        <div className='price_item'>{price}</div>
        <form>
          <p>Виберіть колір:</p>
          <input type="radio" name="color" id="color1" value="Button" />
          <input type="radio" name="color" id="color2" value="Button" />
        </form>
        <div>
          <p>Розмір:</p>
          <div className='size_item'>

```

```

    <span
      onClick={() => toggleSize('S')}
      className={currentSize === 'S' ? 'current_item' : ''}
    >
      S
    </span>
    <span
      onClick={() => toggleSize('M')}
      className={currentSize === 'M' ? 'current_item' : ''}
    >
      M
    </span>
    <span
      onClick={() => toggleSize('L')}
      className={currentSize === 'L' ? 'current_item' : ''}
    >
      L
    </span>
    <span
      onClick={() => toggleSize('XL')}
      className={currentSize === 'XL' ? 'current_item' : ''}
    >
      XL
    </span>
  </div>
</div>
<div>
  <button className='btn_item' onClick={toggleModal}>Придбати</button>
  {isModalOpen && (
    <div className='modal' ref={modalRef}>
      <div className='exit_item' onClick={toggleModal}> X </div>
      <div className='inputs_item'>
        <div style={{color:'#fff'}}>Оформлення заказу</div>
        <input type='text' placeholder='Ваша країна' />
        <input type='text' placeholder='Ваше місто' />
        <input type='text' placeholder='Ваш поштовий індекс' />
        <input type='text' placeholder='Ваша пошта' />
        <p style={{color:'#fff', fontSize:'15px', fontStyle:'italic'}}>*Ця інформація буде надіслана нам для
        поточної перевірки</p>
        <button className='btn_item_modal'>Оформити заказ</button>
      </div>
    </div>
  )}
  <div className='payment_item'>
    <img style={{width:'50px', padding:'10px'}} src={Visa} alt='Visa' />
    <img style={{width:'50px', padding:'10px'}} src={Mastercard} alt='MasterCard' />
    <img style={{width:'50px', padding:'10px'}} src={Privat24} alt='Privat24' />
  </div>
</div>
</div>
);
};
//Navbar.js
import React, { Fragment } from "react";
import Dever_logo from '../source/DEVER.png';

export const Navbar = () => {
  return (
    <Fragment>
      <header>
        <div className="header_first">
          <div className="header_main_logo">

```

```

    <a href="/"
      ><img src={Dever_logo} alt="logo_mini_black"
    /></a>
  </div>
</div>
<div className="header_second">
  <ul className="header_main">
    <li><a href="/ts" className="header_main_item">Футболки</a></li>
    <li><a href="/hoodie" className="header_main_item">Худи</a></li>
    <li><a href="/pants" className="header_main_item">Штаны</a></li>
    <li><a href="/cap" className="header_main_item">Кепки</a></li>
    <li><a href="/bag" className="header_main_item">Рюкзаки</a></li>
    <li><a href="/souvenir" className="header_main_item">Сувеніри</a></li>
    <li><a href="/faq" className="header_main_item">Faq</a></li>
  </ul>
</div>
</header>
</Fragment>
);
};

```

```

//MainPhoto.js
import React from 'react'

```

```

export const MainPhoto = () => {
  return (
    <main>
      { /* <!-- //ФОТО --> */ }
      <div className="wrapper" style={{position:'relative', height:'1000px'}}>
        <h1 className="main_tag">Відрізнятись не заборонено</h1>
        <div className="main_button">
          <a href="/ts">Колекція</a>
        </div>
      </div>
      <div className="push"></div>
    </main>
  )
}

```

```

//ItemList.js
import React from 'react';
import { prodList } from '../data/data';
import { Link } from 'react-router-dom';

```

```

export const ItemList = (type) => {
  const typeOf = type.type.type;

  // Filter the items based on the type
  const filteredItems = prodList.filter((item) => {
    switch (typeOf) {
      case 'ts':
        return item.price.ts !== "";
      case 'hoodie':
        return item.price.hoodie !== "";
      case 'pants':
        return item.price.pants !== "";
      case 'cap':
        return item.price.cap !== "";
      case 'bag':
        return item.price.bag !== "";
      case 'souvenir':
        return item.price.souvenir !== "";
      default:

```

```

    return false;
  }
});

// Render two items for "pants" and three items for "souvenir"
let renderedItems = filteredItems;
if (typeof === 'pants') {
  renderedItems = filteredItems.slice(0, 2);
} else if (typeof === 'souvenir') {
  renderedItems = filteredItems.slice(0, 3);
}

return (
  renderedItems.map((item) => {
    let price, imgSrc, name;

    switch (typeof) {
      case 'ts':
        price = item.price.ts;
        imgSrc = item.img.img_ts;
        name = item.name.main;
        break;
      case 'hoodie':
        price = item.price.hoodie;
        imgSrc = item.img.img_hoodie;
        name = item.name.main;
        break;
      case 'pants':
        price = item.price.pants;
        imgSrc = item.img.img_pants;
        name = item.name.pants;
        break;
      case 'cap':
        price = item.price.cap;
        imgSrc = item.img.img_cap;
        name = item.name.cap;
        break;
      case 'bag':
        price = item.price.bag;
        imgSrc = item.img.img_bag;
        name = item.name.bag;
        break;
      case 'souvenir':
        price = item.price.souvenir;
        imgSrc = item.img.img_souvenir;
        name = item.name.souvenir;
        break;
      default:
        price = "";
        imgSrc = "";
        name = "";
    }

    return (
      <div className="card_hover" key={item.id}>
        <div className="card">
          <div className="price_container">
            <span className="left_side" />
            <span className="price">{price}</span>
            <span className="right_side" />
          </div>
          <Link to={`/product/${typeof}/${name}/${item.id}/${imgSrc}/${price}`} className="img_card">
            <img src={`/images/${typeof}/${imgSrc}`} alt={item.img_alt} />
          </Link>
        </div>
      </div>
    );
  })
);

```

```

    </Link>
    <div className="card_name">{name}</div>
  </div>
</div>
);
})
);
}

```

//Home.js

```

import React, { Fragment } from "react";
import { Looper } from "../script/preloader";
import { MainPhoto } from "./MainPhoto";

```

```

export const Home = () => {
  return (
    <Fragment>
      <Looper />
      <MainPhoto />
    </Fragment>
  )
}

```

//Footer.js

```

import React, { Fragment } from "react";
import Dever_logo_mini from "../source/DE.png";

```

```

export const Footer = () => {
  return (
    <Fragment>
      <footer>
        <div className="quote_footer">
          Дюжина
          <p className="pink">модних</p>
          речей, які ви заслужили!
        </div>
        <div className="main_footer">
          <div className="container">
            <div className="el_list_footer">
              <ul className="el_ul_footer">
                <li><a href="/ts" className="el_footer">Футболки</a></li>
                <li><a href="/hoodie" className="el_footer">Худи</a></li>
                <li><a href="/pants" className="el_footer">Штаны</a></li>
                <li><a href="/cap" className="el_footer">Кепки</a></li>
                <li><a href="/bag" className="el_footer">Рюкзаки</a></li>
                <li><a href="/souvenir" className="el_footer">Сувеніри</a></li>
                <li><a href="/faq" className="el_footer">FAQ</a></li>
              </ul>
            </div>
            <div className="info_footer">
              <div className="logo_footer">
                <img src={Dever_logo_mini} alt="DE_logo" />
                <p>© 2023, Dever.com</p>
              </div>
              <div className="text_footer">
                <div>E-MAIL:</div>
                <div className="email_footer">dever@gmail.com</div>
                <div>ЧАС РОБОТИ СЛУЖБИ ПІДТРИМКИ:</div>
                <div>ПН-ЧТ: 9:00 - 21:00</div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </Fragment>
  )
}

```

```

    </footer>
  </Fragment>
)
}
//Faq.js
import React from 'react'
import { Looper } from '../script/preloader'

export const Faq = () => {
  return (
    <
      <Looper />
      <div className="container" style={{color:'white'}}>
      <h1>Про нас</h1>
      <section>
        <div className="info_text">
          Ми маленька спільнота яка дуже хоче показати що можна
          виглядати круто та стильно одночасно. Зроблено з любов'ю ❤️ від дюжини
          друзів для дюжини людей
        </div>
        <div className="info_text_mini">
          <div>
            <span className="bold">Інформація: </span>
            Сплатити своє замовлення ви можете у розділі де вибрали товар, натиснувши
            кнопку "Придбати".
          </div>
          <div>
            <span className="bold">Оплата: </span>
            здійснюється банківськими картками Visa або MasterCard. Після
            купівлі товару йде перевірка та оформлення вашого замовлення
            адміністрацією та протягом 48 годин ваше замовлення буде оброблено з
            статусом "Замовлення прийнято" Відправлення замовлень здійснюється протягом
            1 - 3 робочих днів після ухвалення їх адміністратором.
            відповідний товар.
          </div>
          <div>
            <span className="bold">Доставка по Україні: </span>
            здійснюється компанією "Нова Пошта". Посилання доставляється протягом 1 - 2
            робочих днів із дати відправлення замовлення. Посилку слід забрати в
            протягом 5 днів з моменту доставки. Через 5 днів
            компанією Нова Пошта стягується доплата за зберігання відправлення.
          </div>
          <div>
            <span className="bold">Доставка за межі України: </span>
            здійснюється компанією «Укрпошта» Посилання доставляється протягом
            7 - 21 робочих днів з дати якого Ви вкажете при оформленні
            заказа.
          </div>
        </div>
      </section>
    </div>
    </>
  )
}
//data.js
export const prodList= [
  {
    id: 1,
    name: {
      main: "4OpToBa DIO}i{uHA (Black)",
      cap: "Чорна класика",
      bag: "Чорна класика",
      pants: "Чорна класика",
    }
  }
]

```



```

    souvenir: "Термостакан DEver"
  },
  price: {
    ts: "599 грн",
    hoodie: "799 грн",
    cap: "249 грн",
    bag: "1499 грн",
    pants: "1199 грн",
    souvenir: "199 грн"
  },
  img: {
    img_ts: "t-s_black-4OpToBa DIO}{i{uHA.png",
    img_hoodie: "black-4OpToBa DIO}{i{uHA_hoodie.png",
    img_cap: "cap_black.png",
    img_bag: "classic_bag.png",
    img_pants: "classic_pants.png",
    img_souvenir: "logo_cup.png"
  },
  img_alt:"Black_classic",
},
{
  id: 2,
  name: {
    main: "Бджола Туги",
    cap: "DEver",
    bag: "Бджола Туги",
    pants: "Titus",
    souvenir: "Titus"
  },
  price: {
    ts: "499 грн",
    hoodie: "699 грн",
    cap: "349 грн",
    bag: "1999 грн",
    pants: "1599 грн",
    souvenir: "249 грн"
  },
  img: {
    img_ts: "t-s_pchela_toski.png",
    img_hoodie: "pchela_toski_hoodie.png",
    img_cap: "cap_white_logo.png",
    img_bag: "pchela_toski_bag.png",
    img_pants: "titus_pants.png",
    img_souvenir: "titus_cup.png"
  },
  img_alt:"Pchela_toski",
},
{
  id: 3,
  name: {
    main: "Бджола Туги Remastered",
    cap: "Clown",
    bag: "Titus DEver",
    souvenir: "Flame"
  },
  price: {
    ts: "599 грн",
    hoodie: "799 грн",
    cap: "299 грн",
    bag: "1999 грн",
    souvenir: "249 грн"
  },
  img: {

```

```

    img_ts: "t-s_pchela_toski_remastered.png",
    img_hoodie: "pchela_toski_remastered_hoodie.png",
    img_cap: "cap_clown.png",
    img_bag: "logo2_bag.png",
    img_souvenir: "flame_cup.png"
  },
  img_alt:"Pchela_toski",
},
{
  id: 4,
  name: {
    main: "Flame",
    cap: "Біла класика",
    bag: "Flame",
  },
  price: {
    ts: "799 грн",
    hoodie: "999 грн",
    cap: "249 грн",
    bag: "1799 грн"
  },
  img: {
    img_ts: "t-s_dimarik_flame.png",
    img_hoodie: "flame_hoodie.png",
    img_cap: "cap_white.png",
    img_bag: "falme_bag.png"
  },
  img_alt:"Flame",
},
{
  id: 5,
  name: {
    main: "Titus",
    cap: "Titus",
    bag: "Titus",
  },
  price: {
    ts: "649 грн",
    hoodie: "849 грн",
    cap: "249 грн",
    bag: "2199 грн",
  },
  img: {
    img_ts: "t-s_titus_logo.png",
    img_hoodie: "titus_hoodie.png",
    img_cap: "cap_titus.png",
    img_bag: "titus_bag.png"
  },
  img_alt:"Titus",
},
{
  id: 6,
  name: {
    main: "Stiker Titus",
    cap: "Flame",
    bag: "Stiker Titus"
  },
  price: {
    ts: "649 грн",
    hoodie: "849 грн",
    cap: "349 грн",
    bag: "1999 грн",
  },
},

```

```

img: {
  img_ts: "t-s_titus_logo_stikers.png",
  img_hoodie: "stiker_titus_hoodie.png",
  img_cap: "cap_flame.png",
  img_bag: "stiker_titus_bag.png"
},
img_alt:"Stiker_titus",
},
{
id: 7,
name: {
  main: "Королівська ніч",
  cap: "Бджола Туги (білий)",
  bag: "Frogus"
},
price: {
  ts: "849 грн",
  hoodie: "1099 грн",
  cap: "299 грн",
  bag: "2199 грн ",
},
img: {
  img_ts: "t-s_king_night.png",
  img_hoodie: "king_night_hoodie.png",
  img_cap: "cap_pchela_toski_white.png",
  img_bag: "frog_bag.png",
},
img_alt:"Stiker_titus",
},
{
id: 8,
name: {
  main: "Жива класика",
  cap: "Бджола Туги (чорний)",
  bag: "DEver",
},
price: {
  ts: "199 грн",
  hoodie: "399 грн",
  cap: "299 грн",
  bag: "1799 грн",
},
img: {
  img_ts: "t-s_classic.png",
  img_hoodie: "classic_hoodie.png",
  img_cap: "cap_pchela_toski_black.png",
  img_bag: "logo_bag.png",
},
img_alt:"Classic",
},
]
//preloader.js
export const Looper = () => {

setTimeout (function(){
  var preloader = document.getElementById("page_loader")
  if(!preloader.classList.contains("done")){
    preloader.classList.add("done");
  }
}, 1000);

return (
<div id="page_loader" className="preloader">

```

```

    <div className="loader"></div>
  </div>
)
}
//footerstyle.scss
.main_footer {
  background-color: rgb(0, 0, 0);
}
%flexcenter {
  display: flex;
  justify-content: center;
}
.el_ul_footer {
  @extend %flexcenter;
  margin: 0px;
  padding: 15px;
}
footer {
  position: relative;
  left: 0;
  bottom: 0;
  width: 100%;
  height: 0px;
}
.el_footer {
  text-transform: uppercase;
  text-decoration: none;
  color: rgb(255, 255, 255);
  font-size: 18px;
  background-image: linear-gradient(currentColor, currentColor);
  background-position: 50% 100%;
  background-repeat: no-repeat;
  background-size: 0% 2px;
  transition: background-size 0.5s;
}
.el_footer:hover {
  background-size: 100% 3px;
}
.el_ul_footer li {
  display: block;
  margin: 0px 25px;
}
.main_footer {
  height: 300px;
}
.info_footer {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
}
.logo_footer img {
  width: 100px;
}
.info_footer {
  margin: 30px 30px 0px 30px;
}
.logo_footer p {
  color: rgb(255, 255, 255);
}
.text_footer div {
  margin-top: 15px;
}
.text_footer {

```

```

    color: white;
  }
  .email_footer {
    cursor: pointer;
  }
  .email_footer:hover {
    color: rgb(128, 128, 128);
    transition: 0.3s;
  }
  .wrapper {
    min-height: 100%;
    margin-bottom: -50px;
  }

//globalstyles.scss
$img-path: "../source/bg_mainlogo.png";
body {
  margin: 0px;
  background-color: black;
  background-image: url($img-path);
  background-size: cover;
  font-family: "Comfortaa", cursive;
  height: 100%;
}
.main_button {
  justify-content: center;
  text-align: center;
}
.wraper {
  display: flex;
  flex-direction: column;
  justify-items: center;
  justify-content: center;
  align-content: center;
  align-items: center;
  position: absolute;
  height: 100%;
  width: 100%;
  left: 0;
  bottom: 0;
}
.main_tag {
  color: #ffffff59;
  font-size: 50px;
  margin-bottom: 100px;
  font-family: 'Roboto Mono', monospace;
  user-select: none
}
.main_tag:hover {
  color: #ffffffb0;
  transition: 0.3s;
  cursor: default;
}
.main_button a {
  color: black;
  font-family: 'Roboto Mono', monospace;
  text-decoration: none;
  padding: 15px 30px;
  background-color: #ffffffcf;
  font-size: 20px;
  font-weight: 600;
}
.main_button a:hover {

```

```

background-color: black;
color: white;
transition: 0.3s;
cursor: pointer;
}
.container {
  max-width: 1140px;
  margin: 0 auto;
}

//headerstyle.scss
header,
footer {
  padding-bottom: 5px;
  z-index: 10;
}
%flexcenter {
  display: flex;
  justify-content: center;
}
.header_main {
  @extend %flexcenter;
}
.header_main li {
  display: block;
  margin: 0px 25px;
}
.header_main_item {
  text-transform: uppercase;
  text-decoration: none;
  color: rgb(255, 255, 255);
  font-size: 15px;
  background-image: linear-gradient(currentColor, currentColor);
  background-position: 50% 100%;
  background-repeat: no-repeat;
  background-size: 0% 2px;
  transition: background-size 0.5s;
}
.header_main_item:hover {
  background-size: 100% 2px;
}
.header_main_logo {
  @extend %flexcenter;
  padding: 20px 0px;
}
.header_main_logo img {
  width: 200px;
}
.header_first {
  border-bottom: 2px white solid;
}
.quote_footer {
  padding: 20px;
  background-color: white;
}
.quote_footer {
  @extend %flexcenter;
  align-items: center;
  font-size: 20px;
}
.pink {
  background-color: pink;
  color: #ff002d;
}

```

```
padding: 5px;
margin: 0px 10px;
}
```

```
//index.scss
@import url("https://fonts.googleapis.com/css2?family=Comfortaa:wght@600&display=swap");
@import url("https://fonts.googleapis.com/css2?family=Amatic+SC:wght@700&display=swap");
@import url("https://fonts.googleapis.com/css2?family=Amatic+SC:wght@700&display=swap");
@import url("https://fonts.googleapis.com/css2?family=Comfortaa:wght@600&display=swap");
@import url("https://fonts.googleapis.com/css2?family=Roboto+Mono:wght@500&display=swap");
%flexcenter {
  display: flex;
  justify-content: center;
}
```

```
// Глобальные стили
@import "globalstyle.scss";
// стили для прелоадера
@import "preloaderstyle.scss";
//стили для хэдера
@import "headerstyle.scss";
// стили для футера
@import "footerstyle.scss";
// стили для товара
@import "itemToBuy.scss";
```

```
//itemToBuy.scss
.wrap {

  display: flex;
  flex-direction: row;
  justify-content: space-around;
  margin: 40px 0px;
  border-radius: 8px;
  padding: 20px;
  background-color: #e9e9e9;
  box-shadow: 0px 0px 20px black;
}
```

```
.info_wrap{
  display: flex;
  flex-direction: column;
  font-size: 20px;
  width: 600px;
}
```

```
.title_item {
  font-size: 40px;
  margin: 20px 0px;
}
```

```
.price_item {
  color: grey;
  font-size: 25px;
}
```

```
.img_card_item{
  display: flex;
  align-items: center;
}
```

```
input[id=color1] {
  appearance: none;
  padding: 18px;
  background-color: rgb(46, 46, 46);
  border-radius:50%;
}
```

```
input[id=color2] {
```

```

    appearance: none;
    padding: 18px;
    background-color: rgb(218, 218, 218);
    border-radius:50%;
}

input[id=color1]:checked {
    border: 3px dashed pink;
    padding: 15px;
}
input[id=color2]:checked {
    border: 3px dashed pink;
    padding: 15px;
}

.size_item {
    display: flex;
    flex-direction: row;
}
.size_item span{
    padding: 5px 15px;
    border: 1px solid black;
    cursor: pointer;
    margin: 0px 0.5px;
}
.current_item {
    background-color: pink;
    color: #ff002d;
}
.current_item:hover {
    border-radius: 4px;
    transition: 0.3s;
}

.btn_item{
    color: #fff;
    width: 100%;
    font-family: 'Roboto Mono', monospace;
    text-decoration: none;
    padding: 15px 30px;
    background-color: #000;
    font-size: 20px;
    font-weight: 600;
    border: none;
    border-radius: 3px;
    margin-top: 20px;
}
.btn_item:hover {
    background-color: #fff;
    color: #000;
    transition: 0.3s;
    cursor: pointer;
}
.payment_item{
    display: flex;
    flex-direction: row;
    align-items: center;
}
.modal {
    position: fixed;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
}

```



```

background-color: rgb(0, 0, 0);
border: 2px solid #fff;
width: 35%;
height: 55%;
padding: 20px;
border-radius: 10px;
z-index: 50;
}
.modal input {
border: none;
background-color: #242424;
color: #fff;
padding: 15px;
border-radius: 3px;
margin: 15px 0px;
width: 75%;
}
.exit_item {
color: red;
display: flex;
justify-content: end;
user-select: none;
cursor: pointer;
}
.inputs_item {
display: flex;
justify-content: center;
flex-direction: column;
align-items: center;
}
.btn_item_modal {
color: #fff;
width: 100%;
font-family: 'Roboto Mono', monospace;
text-decoration: none;
padding: 15px 30px;
background-color: #242424;
font-size: 20px;
border: none;
border-radius: 3px;
margin-top: 40px;
}
.btn_item_modal:hover {
background-color: #c7c7c7;
color: #000;
transition: 0.3s;
cursor: pointer;
}
//preloader.scss
.preloader {
position: fixed;
top: 0px;
left: 0px;
width: 100%;
height: 100%;
background: rgb(0, 0, 0);
z-index: 5;
transition: 1s all;
opacity: 1;
visibility: visible;
}
.loader {
position: absolute;

```

```

top: 50%;
left: 50%;
transform: translate(-50%, -50%);
width: 40px;
height: 40px;
border: rgb(255, 255, 255) solid 5px;
border-radius: 50%;
border-top: rgb(105, 105, 105) 5px solid;
animation: 0.7s spin infinite linear;
}

.done {
opacity: 0;
visibility: hidden;
}

@keyframes spin {
from {
transform: translate(-50%, -50%) rotate(0deg);
}
to {
transform: translate(-50%, -50%) rotate(360deg);
}
}

//prod-footerstyle.scss
/* ФУТЕР */
.main_footer {
background-color: rgb(0, 0, 0);
}
.el_ul_footer {
display: flex;
justify-content: center;
margin: 0px;
padding: 15px;
}
.el_footer {
text-transform: uppercase;
text-decoration: none;
color: rgb(255, 255, 255);
font-size: 18px;
background-image: linear-gradient(currentColor, currentColor);
background-position: 50% 100%;
background-repeat: no-repeat;
background-size: 0% 2px;
transition: background-size 0.5s;
}
.el_footer:hover {
background-size: 100% 3px;
}
.el_ul_footer li {
display: block;
margin: 0px 25px;
}
.main_footer {
height: 300px;
}
.info_footer {
display: flex;
flex-direction: row;
justify-content: space-between;
}
.logo_footer img {

```

```

    width: 100px;
  }
  .info_footer {
    margin: 30px 30px 0px 30px;
  }
  .logo_footer p {
    color: rgb(255, 255, 255);
  }
  .text_footer div {
    margin-top: 15px;
  }
  .text_footer {
    color: white;
  }
  .email_footer {
    cursor: pointer;
  }
  .email_footer:hover {
    color: rgb(128, 128, 128);
    transition: 0.3s;
  }
  .wrapper {
    min-height: 100%;
    margin-bottom: -50px;
  }

//prod-globalstyle.scss
/* ФУТЕР */
.main_footer {
  background-color: rgb(0, 0, 0);
}
.el_ul_footer {
  display: flex;
  justify-content: center;
  margin: 0px;
  padding: 15px;
}
.el_footer {
  text-transform: uppercase;
  text-decoration: none;
  color: rgb(255, 255, 255);
  font-size: 18px;
  background-image: linear-gradient(currentColor, currentColor);
  background-position: 50% 100%;
  background-repeat: no-repeat;
  background-size: 0% 2px;
  transition: background-size 0.5s;
}
.el_footer:hover {
  background-size: 100% 3px;
}
.el_ul_footer li {
  display: block;
  margin: 0px 25px;
}
.main_footer {
  height: 300px;
}
.info_footer {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
}

```

```

.logo_footer img {
  width: 100px;
}
.info_footer {
  margin: 30px 30px 0px 30px;
}
.logo_footer p {
  color: rgb(255, 255, 255);
}
.text_footer div {
  margin-top: 15px;
}
.text_footer {
  color: white;
}
.email_footer {
  cursor: pointer;
}
.email_footer:hover {
  color: rgb(128, 128, 128);
  transition: 0.3s;
}
.wrapper {
  min-height: 100%;
  margin-bottom: -50px;
}

//prod-headerstyle.scss
.header_main_item {
  text-transform: uppercase;
  text-decoration: none;
  color: rgb(255, 255, 255);
  font-size: 15px;
  background-image: linear-gradient(currentColor, currentColor);
  background-position: 50% 100%;
  background-repeat: no-repeat;
  background-size: 0% 2px;
}
.header_main_item:hover {
  transition: background-size 0.5s;
  background-size: 100% 2px;
}
.header_main_logo img {
  width: 200px;
}
.header_main_logo {
  display: flex;
  justify-content: center;
  padding: 20px 0px;
}
.header_main {
  display: flex;
  justify-content: center;
  margin: 0px;
  padding: 15px;
}
.header_main li {
  display: block;
  margin: 0px 25px;
}
.header_first {
  border-bottom: 2px white solid;
}

```

```

header {
  background-color: #0000004d;
  padding-bottom: 5px;
  z-index: 10;
  position: relative;
}

//prod-style.scss
@import url("https://fonts.googleapis.com/css2?family=Comfortaa:wght@600&display=swap");
@import url("https://fonts.googleapis.com/css2?family=Amatic+SC:wght@700&display=swap");
@import url("https://fonts.googleapis.com/css2?family=Amatic+SC:wght@700&display=swap");
@import "prod-globalstyle.scss";
@import "preloaderstyle.scss";
@import "prod-headerstyle.scss";

%flexcenter {
  display: flex;
  justify-content: center;
}

/* //МЕИИИ КОНТЕНТ */
.card_hover {
  box-shadow: none;
  border-radius: 10px;
  margin: 15px;
}
.card_hover:hover {
  box-shadow: 0px 0px 10px rgba(255, 255, 255, 0.521);
  background-color: rgba(255, 255, 255, 0.521);
  transition: 0.3s;
}
.card {
  border-radius: 10px;
  cursor: pointer;
  padding: 5px;
  background-color: #efefefba;
}
.price {
  background-color: #fff;
  display: block;
  font-size: 25px;
  padding-bottom: 5px;
  font-family: "Exo 2", sans-serif;
}
.img_card img {
  width: 330px;
  height: 332px;
}
.container h1 {
  display: flex;
  justify-content: center;
}
.left_side {
  padding: 10px;
  background-color: #fff;
  border-radius: 10px 0px 0px;
}
.right_side {
  padding: 10px;
  background-color: #fff;
  border-radius: 0% 0% 50% 0%;
}
.price_container {

```

```

display: flex;
flex-direction: row;
}
.card_name {
display: flex;
justify-content: center;
font-size: 20px;
padding: 20px;
}

.flex_direction {
display: flex;
flex-direction: row;
flex-wrap: wrap;
justify-content: center;
}
// подключаем футер
@import "prod-footerstyle.scss";

//info-content
.info_text {
@extend %flexcenter;
font-size: 25px;
padding: 20px;
line-height: 50px;
}
.info_text_mini {
font-size: 15px;
padding: 50px;
}
.info_text_mini div {
margin: 20px;
line-height: 30px;
}
.bold {
font-size: 25px;
}

```

**ВІДЗИВ**

**на дипломний проект бакалавра  
на тему:  
«Розробка інтернет магазину з продажу одягу з використанням  
фреймворку React JavaScript.»  
студента групи 122-19-2 Денисова Єгора Володимировича**

**РЕЦЕНЗІЯ**

**на дипломний проект бакалавра**

**на тему:**

**«Розробка інтернет магазину з продажу одягу з використанням  
фреймворку React JavaScript.»  
студента групи 122-19-2 Денисова Єгора Володимировича**



**Відзив керівника економічного розділу**

## ПЕРЕЛІК ФАЙЛІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файла	Опис
Пояснювальні документи	
Диплом.doc	Пояснювальна записка до дипломного проекту. Документ Word.
Диплом.pdf	Пояснювальна записка до дипломного проекту в форматі PDF
Програма	
Dever.zip	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація.ppt	Презентація дипломного проекту