

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

*Житник Володимир Ігорович*

(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 веб сайту для підприємства що*

*виготовляє парфуми з використанням фреймворку JavaScript React*

| 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



## ABSTRACT

Explanatory note: 68- pp., 25 - img., 3 – table., 3 - addition, 12– sources

This qualification work is devoted to the development and implementation of an online perfume store. The main goal of the work was to create a functional and convenient online store that provides users with the opportunity to purchase perfume products in a convenient way via the Internet.

In the first chapter, an analysis of the subject industry is carried out, including the study of market trends and consumer demands in the field of perfumery. The task of developing the functionality of the site, which meets the needs of customers, is set.

The second section is devoted to the design and development of the software product. The layout of the site was carried out using HTML and CSS markup languages, as well as the logic of interaction between the user and the system was developed. The Visual Studio Code working environment and the GitHub repository are used for convenient development management.

The last section is devoted to the economic aspect of software product development. The cost of development was calculated, the costs of creating and maintaining the site were estimated, and the expected financial indicators were determined.

The result of the work is a functional and aesthetic online perfumery store that meets the needs of customers in a convenient and safe way to shop online. The developed site is the result of research in the field of web development and demonstrates knowledge of programming, design and project management technologies.

Keywords: Java-Script, React, WEB APPLICATION, SITE, COURSES, PERFUMES, HTML

## РЕФЕРАТ

Пояснювальна записка містить 68- с., 25 - рис., 3 – табл., 3 - додатка,  
12– джерел

Ключові слова: Java-Script, React, ВЕБ–ДОДАТОК, САЙТ, ПАРФУМИ,  
HTML

Тема кваліфікаційної роботи «розробка веб-сайту для підприємства,  
що виготовляє парфуми, з використанням фреймворку JavaScript React»

Кваліфікаційна робота складається зі вступу, трьох розділів, висновку,  
списку використаної літератури та додатків.

У першому розділі ми провели аналіз предметної галузі, пов'язаної зі  
створенням інтернет-магазину парфумерії. Визначили основні потреби та  
вимоги споживачів у цій галузі, а також розглянули конкурентну ситуацію. На  
основі цього аналізу було поставлено завдання для розробки програмного  
продукту.

У другому розділі ми зосередилися на проектуванні та розробці  
програмного продукту для інтернет-магазину парфумерії. Визначили  
архітектуру системи, розробили дизайн інтерфейсу користувача, реалізували  
основні функціональні можливості, такі як перегляд каталогу товарів,  
оформлення замовлення, оплата та доставка. Також ми використали технології  
HTML, CSS та React для створення фронтенду, а також використали мови  
програмування та фреймворки для розробки бекенду.

В останньому розділі ми провели розрахунок трудомісткості та вартості  
розробки програмного продукту для інтернет-магазину парфумерії. Визначили  
витрати на розробку, включаючи зарплату розробників, витрати на обладнання  
та програмне забезпечення. Також провели аналіз ринку та потенційних  
прибутків від роботи інтернет-магазину парфумерії.

Результатом проведеної роботи є створення програмного продукту для  
інтернет-магазину парфумерії, який відповідає потребам споживачів у цій галузі,  
а також аналіз його економічної ефективності.

## ЗМІСТ

|  |    |
|--|----|
| ABSTRACT .....   | 3  |
| РЕФЕРАТ .....  | 4  |
| ЗМІСТ .....  | 5  |
| Перелік умовних позначень .....  | 7  |
| Вступ.....   | 8  |
| РОЗДІЛ 1 .....   | 9  |
| АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА<br>ЗАВДАННЯ .....                 | 9  |
| 1.1 Загальні відомості з предметної галузі .....                         | 9  |
| 1.1.1. Особливості електронної комерції.....                             | 10 |
| 1.1.2. Історія парфумерної індустрії .....                               | 11 |
| 1.1.3. Сучасний стан парфумерної індустрії.....                          | 12 |
| 1.1.4. Ключові характеристики парфумів.....                              | 12 |
| 1.1.5. Конкурентна аналітика .....                                       | 13 |
| 1.2. Призначення розробки та галузь застосування .....                   | 13 |
| 1.3. Підстава для розробки .....   | 14 |
| 1.4. Постановка завдання.....  | 16 |
| 1.5. Вимоги до програми або програмного виробу.....                      | 16 |
| 1.5.1. Вимоги до функціональних характеристик: .....                     | 18 |
| 1.5.2. Вимоги до інформаційної безпеки:.....                             | 18 |
| 1.5.3. Вимоги до складу та параметрів технічних засобів: .....           | 19 |
| 1.5.4. Вимоги до інформаційної та програмної сумісності: .....           | 19 |
| 1.6 Актуальність проблеми.....   | 19 |
| 1.7 Аналіз існуючих аналогів.....  | 21 |
| РОЗДІЛ 2.....  | 26 |
| ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ.....                       | 26 |
| 2.1 Функціональне призначення програми інтернет магазину парфумерії..... | 26 |
| 2.2 Опис застосованих математичних методів.....                          | 27 |
| 2.3 Опис використаної архітектури та шаблонів проектування .....         | 28 |
| 2.4 Опис використаних технологій та мов програмування.....               | 29 |
| 2.5 Дизайн та верстка сайту.....   | 35 |
| 2.6 Створення карток товарів .....                                       | 41 |
| 2.7 Створення кошика.....  | 44 |
| 2.8 Розробка відправки замовлення.....                                   | 48 |
| 2.9 Імпортування веб-додатку на хостинг.....                             | 50 |
| РОЗДІЛ 3.....  | 52 |

|  |    |
|--|----|
| ЕКОНОМІЧНИЙ РОЗДІЛ .....   | 52 |
| 3.1 Розрахунок трудомісткості та вартості розробки програмного продукту..... | 52 |
| 3.2 Рахунок витрат на створення програми .....                               | 54 |
| Висновок .....   | 57 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....  | 58 |
| Додаток А.....   | 59 |
| Додаток Б .....  | 67 |
| Додаток В.....   | 68 |

## Перелік умовних позначень

- HTML: Мова розмітки гіпертексту (англ. Hypertext Markup Language)
- CSS: Каскадні таблиці стилів (англ. Cascading Style Sheets)
- JS: JavaScript, мова програмування
- API: Інтерфейс програмування додатків (англ. Application Programming Interface)
- UI: Користувацький інтерфейс (англ. User Interface)
- UX: Користувацький досвід (англ. User Experience)
- CMS: Система управління вмістом (англ. Content Management System)
- MVC: Модель-подання-контролер (англ. Model-View-Controller), архітектурний шаблон програмування
- CRUD: Створення, читання, оновлення, видалення (англ. Create, Read, Update, Delete), операції з базою даних
- IDE: Інтегроване середовище розробки (англ. Integrated Development Environment)
- Git: Система контролю версій, платформа для спільної роботи над кодом
- API: Інтерфейс програмування додатків (англ. Application Programming Interface)
- CMS: Система управління вмістом (англ. Content Management System)
- EDT: Туалетна вода (англ. Eau de Toilette)
- EDP: Парфумована вода (англ. Eau de Parfum)
- EDC: Колонійна вода (англ. Eau de Cologne)
- EDF: Фреш-вода (англ. Eau de Fraiche)
- Extrait: Екстракт парфуму (найконцентрованіша форма парфуму)
- Sillage: Відбиток запаху, його інтенсивність та тривалість
- Top notes: Верхні ноти, перші відчуття запаху
- Middle notes: Середні ноти, розвиток запаху після висихання верхніх нот
- Base notes: Базові ноти, стійкі основні складові запаху
- Projection: Проекція, відстань, на яку запах відчутний
- Longevity: Тривалість, скільки часу запах залишається на шкірі

## Вступ

Сучасний інтернет простір надає безліч можливостей для розвитку бізнесу. Інтернет-магазини стають все більш популярними, забезпечуючи зручність та доступність для покупців з усього світу. У такому контексті, створення власного інтернет-магазину стає важливим завданням для підприємців, які прагнуть розширити свій бізнес та привернути нових клієнтів.

Цей проект присвячений створенню інтернет-магазину парфумерії - платформи, яка дозволить клієнтам з легкістю придбати різноманітні парфуми. Головною метою проекту є створення зручного та привабливого інтерфейсу, який надасть користувачам можливість ознайомитися з асортиментом, зробити замовлення та отримати задоволення від покупки.

В цьому проекті я вивчатиму та використовуватиму різноманітні технології, мови програмування та інструменти, щоб створити функціональний та естетичний інтернет-магазин. Використання таких технологій, як HTML, CSS, React та Visual Studio, дозволить мені створити високоякісний дизайн та реактивну поведінку веб-сторінок.

В подальшому, я розгляну процес розробки, починаючи з аналізу вимог та вибору архітектури, продовжуючи розробкою та тестуванням, і завершуючи розгортанням на хостингу. Також буде проведений розрахунок витрат на створення програмного продукту, щоб забезпечити ефективне фінансове планування.

Цей проект відкриє для мене нові можливості в галузі веб-розробки та допоможе розвинути мої навички програмування та дизайну. В результаті, метою цього проекту є створення інтернет-магазину, який буде задовольняти потреби клієнтів та стимулювати розвиток бізнесу у сфері парфумерії.



# РОЗДІЛ 1

## АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА

### ЗАВДАННЯ

Основні завдання та цілі розробки веб-сайту для підприємства, що виготовляє парфуми, з використанням фреймворку JavaScript React. Завдання включають функціональні, технічні та дизайн-орієнтовані аспекти проекту. Також будуть визначені основні вимоги до розробки та очікувані результати.

#### 1.1 Загальні відомості з предметної галузі

Предметна галузь інтернет-магазину парфумерії є специфічною, оскільки вона пов'язана з продажем і розповсюдженням парфумів і косметичних товарів. Основні відомості з цієї галузі включають такі аспекти[1]:

1. Парфуми та косметика: Це включає різноманітні парфуми, туалетні води, духи, косметичні засоби для догляду за шкірою, волоссям і макіяжем. Розуміння різних типів продуктів, їх складу, властивостей та використання є важливим для ефективного функціонування інтернет-магазину.
2. Бренди та виробники: У парфумерній галузі існує багато відомих брендів і виробників, які пропонують свою продукцію. Розуміння різних брендів, їхньої репутації та якості є важливим для вибору правильного асортименту товарів у магазині.
3. Асортимент товарів: Інтернет-магазин парфумерії повинен мати широкий асортимент продуктів, який включає різні бренди, типи парфумів і косметики, а також варіанти упаковки і розмірів. Розуміння різноманітності товарів і вміння куриувати асортимент є важливими для привертання покупців.
4. Тренди та модність: Парфумерна галузь постійно змінюється, з'являються нові тренди, аромати та дизайн упаковки. Оновлення з трендами та

здатність адаптуватися до змін є важливими для успішності інтернет-магазину.

5. Законодавство та регулювання: В продажі парфумів і косметики існують певні правила та обмеження, пов'язані з безпекою, маркуванням, сертифікацією та іншими аспектами. Дотримання вимог законодавства та регулювання є важливим для запобігання проблемам і забезпечення безпеки покупців.

Ці загальні відомості з предметної галузі допоможуть розуміти основні аспекти, що пов'язані з розробкою та управлінням інтернет-магазину парфумерії.

### **1.1.1. Особливості електронної комерції**

Електронна комерція, або e-commerce, є сферою бізнесу, де купівля і продаж товарів та послуг здійснюються через Інтернет. Вона зазнала значного розквіту в останні роки та стала невід'ємною частиною багатьох галузей, включаючи парфумерну індустрію.

Особливості електронної комерції, які варто враховувати при розробці веб-сайту для підприємства, що виготовляє парфуми, включають:

1. **Онлайн-магазин:** Електронна комерція дає можливість створити власний онлайн-магазин, де клієнти зможуть переглядати асортимент парфумерної продукції, отримувати детальну інформацію про кожен продукт та здійснювати покупки безпосередньо через веб-сайт.
2. **Електронні платежі:** Важливим аспектом електронної комерції є можливість приймати онлайн-платежі в режимі реального часу. Інтеграція з платіжними шлюзами дозволить клієнтам зручно та безпечно здійснювати оплату за замовлення.
3. **Маркетингові можливості:** Електронна комерція надає широкі можливості для реклами, маркетингу та просування продуктів. Веб-сайт може бути

оптимізований для пошукових систем, реалізувати програми лояльності, надавати знижки та промокоди, що стимулюють покупку.

4. Клієнтський сервіс: Електронна комерція вимагає налагодження ефективної системи підтримки клієнтів. Це можуть бути онлайн-чати, електронна пошта, телефонна підтримка або інші засоби комунікації, що дозволяють клієнтам звертатися за допомогою та отримувати відповіді на свої запитання.
5. Логістика та доставка: Важливим аспектом електронної комерції є організація процесу доставки продукції клієнтам. Веб-сайт повинен включати можливість вибору способу доставки, відстеження замовлень та надання відповідної інформації про доставку.

Розуміння особливостей електронної комерції допоможе розробити веб-сайт для підприємства, що виготовляє парфуми, з урахуванням потреб та вимог сучасних онлайн-клієнтів.

### **1.1.2. Історія парфумерної індустрії**

Парфумерна індустрія має довгу і багатогранну історію, яка починається зі стародавніх цивілізацій. Відомості про використання ароматів і парфумів знаходяться в історичних джерелах різних епох та культур.

Перші згадки про парфуми можна відстежити ще в Давньому Єгипті, де аромати використовувалися в релігійних обрядах та культових церемоніях. У Давній Греції і Римі парфуми отримали широке поширення як символ розкоші та статусу. Парфуми стали неодмінною частиною щоденної гігієни та особистого догляду. У середньовічній Європі парфумерна індустрія переживала зміну у підходах до виготовлення ароматів. Парфумери стали використовувати природні інгредієнти, такі як квіткові есенції, мускус, амбра та інші, для створення складних композицій. У 19-20 століттях парфумерна індустрія пережила період інтенсивного розвитку, з'явилися відомі парфумерні будинки, такі як Chanel,

Guerlain, Dior та інші. Були розроблені нові технології отримання есенцій та штучних ароматів, що значно розширило можливості створення парфумів.

### **1.1.3. Сучасний стан парфумерної індустрії**

Сьогодні парфумерна індустрія є глобальним ринком, який охоплює виробництво, дистрибуцію та продаж парфумерної продукції. У світі існує багато відомих брендів, які пропонують широкий спектр парфумів для різних цільових аудиторій. Сучасні тренди в парфумерній індустрії включають створення унісекс-ароматів, використання натуральних та органічних інгредієнтів, а також спеціалізовані лінії ароматів для конкретних груп споживачів, наприклад, аромати для спорту, вечірні парфуми тощо.

Онлайн-продажі парфумів та електронна комерція в цілому також займають все більшу частку ринку. Багато парфумерних компаній мають свої веб-сайти та інтернет-магазини, що дає змогу клієнтам замовляти продукцію онлайн та дізнаватись більше про аромати та їх характеристики.

### **1.1.4. Ключові характеристики парфумів**

Парфуми складаються з різних компонентів, які взаємодіють між собою, утворюючи унікальний аромат. Основні складові парфумів включають ароматичні есенції, розчинники, фіксативи та розведення.

Ароматичні есенції є основою парфумів і надають їм характерний запах. Вони можуть бути природного походження (отримані з рослин, квітів, фруктів) або штучні (створені хімічним шляхом). Розчинники використовуються для розчинення ароматичних есенцій і забезпечення стійкості аромату. Фіксативи допомагають зберегти аромат на шкірі тривалий час. Розведення включає воду та інші розчинники, які розріджують концентровані парфуми.

### **1.1.5. Конкурентна аналітика**

Для розробки веб-сайту для підприємства, що виготовляє парфуми, важливо вивчити конкурентну ситуацію на ринку. Розглядання веб-сайтів та онлайн-присутності конкурентів допоможе зрозуміти їхні стратегії, представлення продукції, дизайн та функціональні можливості.

Аналіз конкурентів дозволить виявити їхні переваги та слабкі місця, що може стати цінною інформацією для покращення веб-сайту підприємства і створення конкурентоздатного продукту.

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

Призначення розробки веб-сайту для підприємства, що виготовляє парфуми, полягає в створенні ефективного і привабливого інструменту для представлення компанії, її продукції та просування на ринку. Розробка веб-сайту на базі фреймворку JavaScript React дозволить створити веб-додаток з високою функціональністю, швидкістю та зручним інтерфейсом.

Галузь застосування розробленого веб-сайту охоплює парфумерну індустрію та електронну комерцію. Веб-сайт буде спрямований на просування та продаж продукції підприємства, залучення нових клієнтів та підтримку вже існуючих.

Розроблений веб-сайт має наступні головні функції та застосування:

1. Представлення продукції: Веб-сайт дозволить підприємству ефективно представити свою лінію парфумів, включаючи опис кожного аромату, його склад, фотографії та інші характеристики. Це допоможе клієнтам отримати повну інформацію про продукцію перед здійсненням покупки.

2. **Онлайн-продажі:** Веб-сайт забезпечить можливість клієнтам замовляти парфуми онлайн. Клієнти зможуть додавати товари до кошика, здійснювати онлайн-оплату та вибирати способи доставки. Це забезпечить зручність та швидкість процесу покупки.
3. **Клієнтський облік:** Веб-сайт може містити систему реєстрації та обліку клієнтів. Це дозволить клієнтам створювати особисті облікові записи, переглядати історію замовлень, зберігати улюблені аромати та використовувати інші персоналізовані функції.
4. **Маркетинг та реклама:** Веб-сайт може включати різноманітні маркетингові інструменти, такі як розсилка новин та акційних пропозицій, програми лояльності, відгуки клієнтів та можливості соціального ділінгу. Це допоможе підтримувати зв'язок з клієнтами та залучати нових покупців.
5. **Аналітика та звітність:** Веб-сайт може надавати засоби збору та аналізу даних про клієнтів, продажі, популярність продуктів та інші ключові метрики. Це дозволить підприємству здійснювати ефективний моніторинг ринку та приймати обґрунтовані рішення щодо розвитку бізнесу.

Розробка веб-сайту з використанням фреймворку JavaScript React сприятиме ефективній комунікації з потенційними клієнтами, розширенню аудиторії та покращенню конкурентоспроможності підприємства в сучасному цифровому середовищі.

### **1.3. Підстава для розробки**

Розробка веб-сайту для підприємства, що виготовляє парфуми, базується на деяких вагомих підставах, які впливають на необхідність і доцільність такого проекту. Основні підстави для розробки веб-сайту з використанням фреймворку JavaScript React включають:

1. **Інтернет-присутність:** В сучасному світі важко уявити бізнес без наявності в Інтернеті. Веб-сайт є візитною карткою підприємства та дозволяє досягти

більшої кількості потенційних клієнтів. Він надає можливість представити продукцію, надати детальну інформацію та залучити клієнтів для покупки.

1. Електронна комерція: Розробка веб-сайту з використанням фреймворку JavaScript React дозволяє підприємству впровадити електронну комерцію і створити онлайн-магазин для продажу своїх парфумерних виробів. Це дозволить розширити аудиторію та забезпечити клієнтам зручність покупок через Інтернет.
2. Покращення конкурентоспроможності: Застосування сучасних технологій, таких як фреймворк JavaScript React, дозволяє підприємству виграти конкурентну перевагу. Створення веб-сайту з високою швидкістю, привабливим дизайном та функціональністю сприятиме залученню і утриманню клієнтів.
3. Зручність для клієнтів: Веб-сайт надає клієнтам можливість переглядати асортимент продукції, отримувати детальну інформацію про аромати, порівнювати їх та здійснювати замовлення в будь-який зручний для них час. Це сприяє збільшенню задоволення клієнтів та покращенню їх досвіду покупок.
4. Маркетингові можливості: Розроблений веб-сайт надає підприємству можливості для проведення різноманітних маркетингових акцій, таких як розсилка новин, пропозиції для постійних клієнтів та залучення нових клієнтів через соціальні мережі. Це сприяє підвищенню уваги до бренду та збільшенню продажів.

Отже, розробка веб-сайту з використанням фреймворку JavaScript React є доцільною і має підстави на основі необхідності підприємства в ефективному засобі представлення, продажу та просування продукції у сучасному цифровому середовищі.

## **1.4. Постановка завдання**

Метою даного кваліфікаційної роботи є розробка веб-сайту для комерційної діяльності парфюмерного магазину "PerfumerShop". Цей веб-додаток має популярність серед широкого кола клієнтів, що шукають якісні парфуми.

Основним призначенням веб-сайту магазину "PerfumerShop" є створення зручного та доступного каналу замовлення парфюмерної продукції для клієнтів. Консультанти магазину зможуть приймати замовлення як по телефону, так і через Telegram, що забезпечить швидку відповідь та зручну комунікацію з покупцями.

Вимоги до функціональності веб-додатку включають можливість зручного оформлення замовлення через інтуїтивно зрозумілий інтерфейс, а також можливість перегляду товарів та акцій з мобільних пристроїв. Дизайн сайту буде сучасним та привабливим, адаптованим під різні пристрої.

Для забезпечення успішної реалізації проекту, необхідно врахувати сучасні тенденції веб-дизайну та розробити функціонал, що дозволить зручно переглядати та замовляти парфюмерну продукцію. Завдання проекту включають експертний аналіз предметної області, вибір необхідних інструментів для розробки веб-додатку, створення прототипу сайту для десктопної та мобільної версії, реалізацію функціоналу та забезпечення відповідності вимогам клієнтів.

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

Розробка веб-сайту для підприємства, що виготовляє парфуми з використанням фреймворку JavaScript React, вимагає деяких специфічних функціональних вимог. Основні вимоги до програмного виробу включають:



1. Корзина: Веб-сайт повинен мати функцію корзини, яка дозволяє клієнтам додавати товари до кошика покупок. Корзина має зберігати вибрані товари та їх кількість. Крім того, користувач повинен мати можливість змінювати кількість товарів або видаляти їх з корзини.
2. Створення товару за шаблоном: Для зручності адміністраторів веб-сайту має бути функціонал створення нових товарів за шаблоном. Це дозволить швидко створювати нові товари з попередньо заданими характеристиками, що спростить процес додавання нової парфумерної продукції.
3. Відправка замовлення у Telegram: Для забезпечення ефективного способу отримання замовлень, веб-сайт повинен мати можливість відправляти інформацію про нові замовлення у Telegram-канал або групу. Це дозволить оперативно отримувати і обробляти замовлення.
4. Сортування за категоріями товару: Для полегшення навігації та пошуку продукції, веб-сайт повинен мати функцію сортування товарів за категоріями. Клієнти повинні мати можливість переглядати продукцію за певними категоріями, такими як жіночі аромати, чоловічі аромати, набори тощо.
5. Зручний пошук: Веб-сайт повинен мати функцію пошуку, яка дозволяє клієнтам швидко знайти потрібний продукт за назвою, описом або характеристиками.
6. Адаптивний дизайн: Веб-сайт повинен бути адаптивним і коректно відображатися на різних пристроях, таких як комп'ютери, планшети та смартфони. Це забезпечить зручний доступ до веб-сайту незалежно від типу пристрою, який використовує користувач.
7. Безпека та захист даних: Веб-сайт повинен забезпечувати належний рівень безпеки та захисту клієнтських даних. Це включає використання шифрування для передачі даних, захист від несанкціонованого доступу та резервне копіювання інформації.

Враховуючи ці вимоги, розробка веб-сайту з використанням фреймворку JavaScript React дозволить створити функціональний, зручний та безпечний інтернет-магазин для підприємства з виробництва парфумів.

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

- Корзина: Система повинна мати функціональність для додавання товарів до корзини, видалення товарів з корзини та зміни кількості товарів.
- Створення товару за шаблоном: Має бути можливість швидкого створення нових товарів за певними шаблонами з попередньо заданими характеристиками.
- Відправка замовлення у Telegram: Система має мати функціональність для автоматичної відправки інформації про замовлення у Telegram-канал або групу.
- Сортування за категоріями товару: Клієнтам повинно бути зручно сортувати товари за категоріями, щоб полегшити пошук потрібного продукту.

### **1.5.2. Вимоги до інформаційної безпеки:**

- Шифрування даних: Система повинна забезпечувати шифрування передачі даних між клієнтом і сервером, забезпечуючи конфіденційність інформації.
- Захист від несанкціонованого доступу: Мають бути встановлені механізми захисту, такі як аутентифікація і авторизація користувачів, щоб забезпечити захист від несанкціонованого доступу до системи.
- Резервне копіювання даних: Система повинна мати механізм регулярного резервного копіювання даних, щоб уникнути втрати важливої інформації.

### **1.5.3. Вимоги до складу та параметрів технічних засобів:**

- **Масштабованість:** Система повинна бути спроектована таким чином, щоб мати можливість масштабування, якщо збільшується обсяг даних або кількість користувачів.
- **Швидкодія:** Технічні засоби повинні забезпечувати достатню швидкодію системи, щоб забезпечити плавну та ефективну роботу.
- **Доступність:** Система повинна бути доступною для користувачів без перебоїв та витоку часу.

### **1.5.4. Вимоги до інформаційної та програмної сумісності:**

- **Браузерна сумісність:** Система повинна працювати на різних веб-браузерах, таких як Google Chrome, Mozilla Firefox, Safari та інші популярні браузери.
- **Платформна сумісність:** Система повинна бути сумісною з різними операційними системами, такими як Windows, macOS, Linux і т.д.
- **Інтеграція зі сторонніми сервісами:** Може бути потрібна інтеграція зі сторонніми сервісами, такими як платіжні системи або системи доставки, для забезпечення повноцінної роботи системи.
- 

## **1.6 Актуальність проблеми**

З ростом швидкості розвитку Інтернету та соціальних мереж, підприємства все більше переходять на онлайн-платформи, щоб наблизитися до своїх потенційних клієнтів. Цей тренд охоплює різні сфери бізнесу, включаючи ринок пакувальної продукції. У сучасному світі, де якість та креативність стають ключовими чинниками у створенні торгових майданчиків, підприємства постійно змінюються, щоб відповідати новим вимогам та трендам.

Зростаюча конкуренція та зміни відбуваються в цій галузі зі швидкістю світла. Кожен фахівець, що працює у сфері пакувальної продукції, повинен постійно вдосконалюватися та підвищувати свою кваліфікацію, щоб відповідати сучасним вимогам та потребам клієнтів.

З кожним роком зростає кількість підприємств, які потребують пакувальної продукції для своїх товарів, а також з'являються нові види продукції. Тому, для того щоб залишатися конкурентоспроможними, необхідно бути на крок попереду і знати, які тренди пануватимуть на ринку пакувальної продукції.

Тільки ті, хто встигає впроваджувати нові ідеї та технології, можуть забезпечити своєму бізнесу стабільний розвиток і привабливість для клієнтів. У цьому контексті розробка веб-сайту для підприємства, що виробляє парфуми, з використанням фреймворку JavaScript React, є відповіддю на потреби ринку та сприяє залученню нових клієнтів, підвищенню продажів та підвищенню конкурентоспроможності.

Актуальність проблеми розробки веб-сайту для підприємства, що виготовляє парфуми, з використанням фреймворку JavaScript React, визначається декількома факторами:

1. Зростання електронної комерції: Електронна комерція стала одним з найважливіших способів продажу товарів у сучасному світі. Попит на онлайн-покупки продовжує зростати, що створює потребу у розробці веб-сайтів, які забезпечують зручну та безпечну покупку товарів.
2. Важливість онлайн-присутності: Для підприємств, що виробляють парфуми, важливо мати ефективну онлайн-присутність, щоб привернути нових клієнтів і розширити свою аудиторію. Розробка веб-сайту дозволяє підприємству представляти свою продукцію із зручним способом покупки.
3. Зручність для клієнтів: Розробка веб-сайту з корзиною, можливістю створення товару за шаблоном, відправкою замовлення у Telegram та сортуванням за категоріями товару забезпечує зручність і легкість

використання для клієнтів. Це може збільшити їх задоволеність та залучити нових покупців.

4. Конкуренентоспроможність: У сучасному бізнес-середовищі, особливо в галузі парфумерії, конкуренція є дуже високою. Мати сучасний, функціональний та зручний веб-сайт може допомогти підприємству виділитися на ринку, залучити увагу клієнтів та створити конкурентну перевагу.

Отже, розробка веб-сайту для підприємства, що виготовляє парфуми, з використанням фреймворку JavaScript React, має велику актуальність у сучасному бізнес-середовищі, де електронна комерція та онлайн-присутність стають все більш важливими для успіху підприємств.

### **1.7 Аналіз існуючих аналогів**

Створення парфюмерного сайту розпочинається з вивчення дизайну та врахування бажань замовника. Одним з інструментів, які ми використовуємо на початковому етапі, є Mood board. Mood board (дошка настрою) - це візуальна презентація, що складається з зображень, тексту та зразків об'єктів, що виражають загальну ідею або почуття з певної тематики. Спільно з замовником, ми аналізуємо існуючі веб-сайти з парфюмерною тематикою, щоб визначити, які елементи нам будуть потрібні на нашому сайті. Один з аналогів, який варто розглянути - [perfumer.ua](http://perfumer.ua). Цей веб-ресурс представляє інтернет-магазин парфюмерної продукції в Дніпрі. Сайт включає в себе крамницю, де можна ознайомитися з асортиментом товару та проектами компанії, а також отримати консультацію від фахівців та зробити замовлення.

На основі аналізу існуючих сайтів та спільної роботи зі замовником, ми зможемо розробити наш парфюмерний сайт, враховуючи відповідні елементи та функціонал, що задовольняють потреби нашої мети.

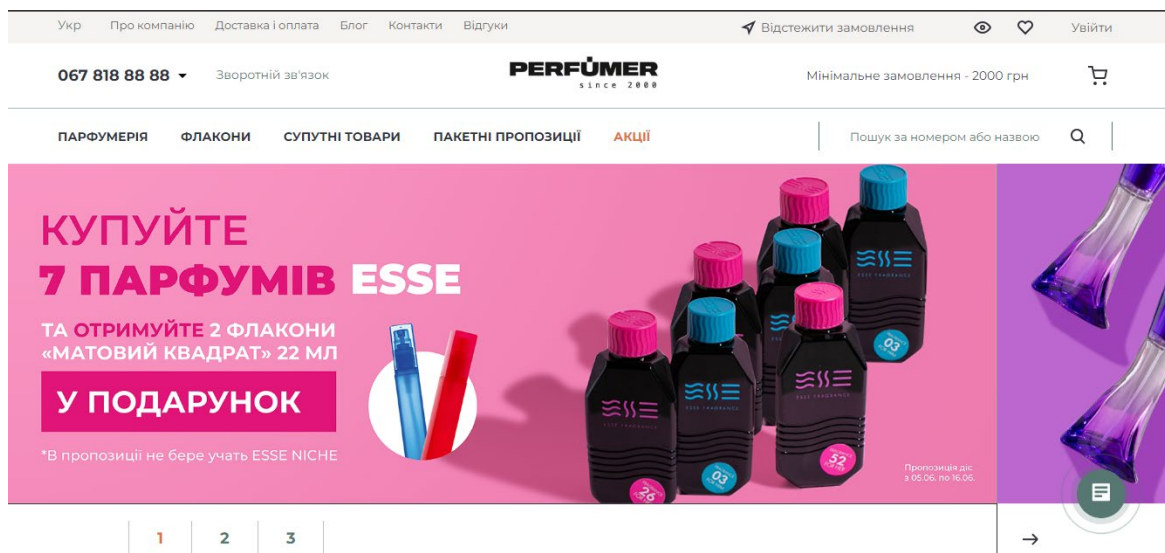


Рис 1.1. – Початкова сторінка сайту

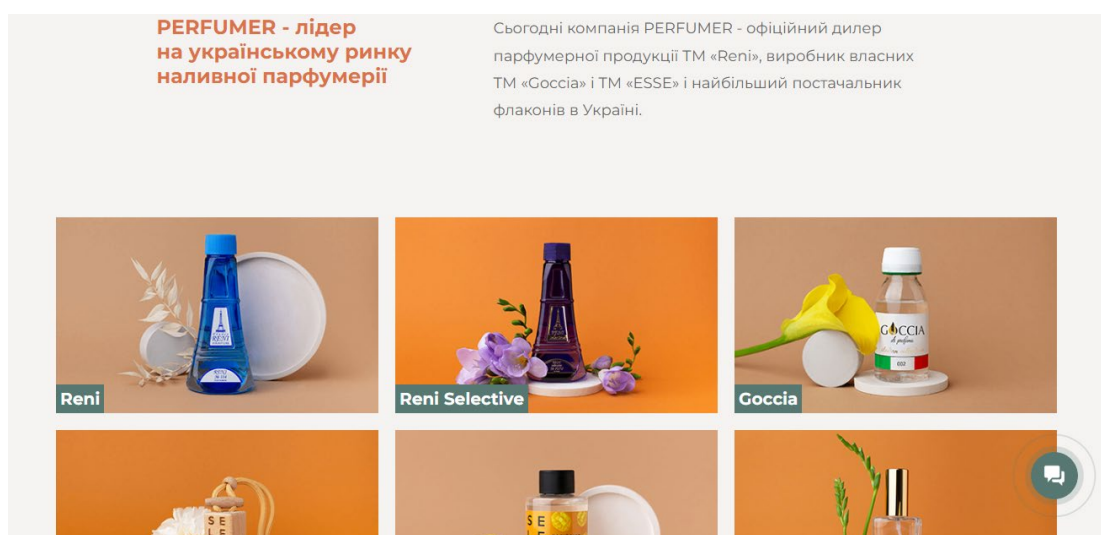


Рис 1.2. – Інформація про компанію

Аналог – [parfum.dp.ua](http://parfum.dp.ua). На цьому сайті продукція представлена у вигляді віртуальної вітрини, що дозволяє відвідувачам ознайомитися з асортиментом товарів та магазином в цілому. Крім того, на ресурсі є можливість зробити замовлення, враховуючи індивідуальні потреби клієнта. На рис. 1.3–1.5 представлені перша сторінка сайту, форма замовлення та віртуальна вітрина товарів.



Рис 1.3. – Головна сторінка «parfum.kh.ua»

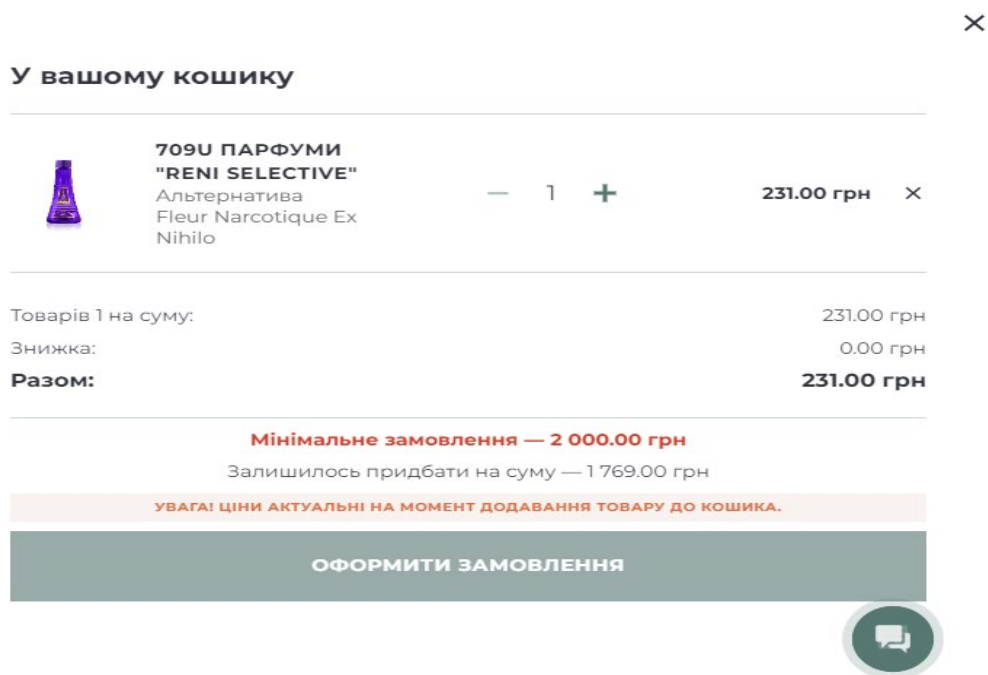


Рис. 1.4. – Кошик



Рис. 1.5. – Товари

Цей сайт відрізняється від типових інтернет-магазинів завдяки широкому вибору продукції, яка охоплює як харчові, так і промислові товари. Особливістю цього магазину є те, що вони самі виробляють свою продукцію, що забезпечує розмаїття товарів.

Серед основних елементів, які ми використовували на нашому сайті, варто виділити наступні:

1. Кошик, до якого можна додавати товари та безпосередньо оформляти замовлення.
2. Односторінкову вітрину товарів, яка уникне потреби відкривати кожен товар на окремій сторінці.
3. Картки товарів з повною інформацією та кнопками для зміни кількості товару та оформлення замовлення.
4. Вертикальне меню для легкого переходу між різними групами товарів.
5. Навігаційне меню, яке допомагає користувачам орієнтуватися на сайті.
6. Відправлення замовлення клієнта до телеграму.
7. Використання модальних вікон для зручного інформування користувачів.
8. Мобільна версія сайту з гамбургер-меню для зручності навігації.

Ці елементи допомагають нам створити функціональний та зручний сайт для наших клієнтів.



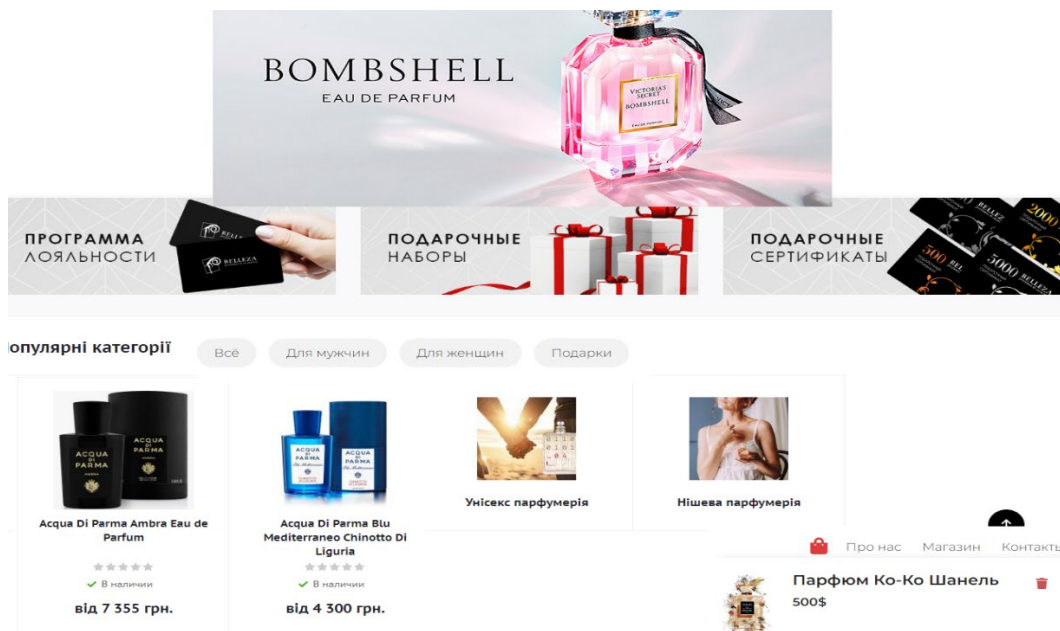


Рис. 1.6. – Дошка настрою

Таблиця 1.1

### Порівняльна характеристика аналогів

| Вимоги                           | Сайт         |             |              |                  |
|----------------------------------|--------------|-------------|--------------|------------------|
|                                  | ParfumerShop | perfumer.ua | parfum.kh.ua | parfum-city.shop |
| Сучасний дизайн                  | +            | +           | +            | +                |
| Landing Page                     | +            | -           | -            | +                |
| Можливість оформлення замовлення | +            | +           | +            | +                |
| Інформація та соц. мережі        | +            | +           | +            | +                |
| Наявність мобільної версії       | +            | -           | +            | +                |

## РОЗДІЛ 2.

### ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

#### 2.1 Функціональне призначення програми інтернет магазину парфумерії

Функціональне призначення програми інтернет-магазину парфумерії полягає в наданні користувачам зручного та ефективного середовища для покупки парфумерних товарів через Інтернет. Основна мета програми полягає в тому, щоб надати користувачам можливість швидкого перегляду, вибору та придбання парфумерних продуктів за допомогою веб-інтерфейсу.

Основні функції інтернет-магазину парфумерії можуть включати:

1. Перегляд асортименту: Користувачі можуть переглядати доступний асортимент парфумерних товарів, який може бути розподілений за категоріями, брендами, типами ароматів тощо. Кожен товар представлений зображенням, описом, ціною та іншою необхідною інформацією.
2. Пошук товарів: Користувачі можуть використовувати пошукову функцію для швидкого знаходження конкретного товару за назвою, брендом або іншими параметрами. Це дозволяє ефективно орієнтуватися в широкому асортименті продукції.
3. Додавання до кошика: Користувачі можуть додавати понравившись товари до віртуального кошика, щоб зберігати їх для подальшого придбання. В кошику відображається загальна сума покупок та кількість товарів.
4. Оформлення замовлення: Користувачі можуть перейти до процесу оформлення замовлення, де вони надають необхідну інформацію про доставку, контактні дані та інші деталі. Після підтвердження замовлення, користувачі можуть здійснити оплату.
5. Керування обліковим записом: Користувачі можуть створити обліковий запис, де вони можуть зберігати свої особисті дані, переглядати історію

замовлень, відстежувати статуси доставки тощо.

6. Опції доставки та оплати: Користувачі можуть обрати підходящий спосіб доставки (кур'єрська служба, поштова доставка, самовивіз тощо) та спосіб оплати (готівкою, онлайн-переказом, платіжними картками тощо).
7. Відгуки та рейтинги: Користувачі можуть залишати відгуки та оцінки для придбаних товарів, що дозволяє іншим користувачам отримати додаткову інформацію та рекомендації перед покупкою.
8. Контактна інформація та підтримка: Користувачі можуть знайти контактну інформацію для зв'язку з представниками магазину та отримати підтримку щодо питань про товари, доставку або оплату.

Функціональне призначення програми інтернет-магазину парфумерії забезпечує зручну та ефективну платформу для покупки парфумерних товарів з будь-якого пристрою, надаючи користувачам доступ до широкого асортименту продукції, зручність у виборі, оформленні замовлення та оплаті, а також забезпечує зручну комунікацію та підтримку для задоволення потреб покупців.

## **2.2 Опис застосованих математичних методів**

У контексті розробки програми для інтернет-магазину парфумерії, застосування математичних методів може бути обмеженим. Проте, деякі математичні методи можуть бути використані для покращення деяких аспектів функціонування магазину. Ось кілька прикладів:

1. Алгоритми рекомендацій: Математичні методи, такі як алгоритми колаборативного фільтрування або алгоритми на основі машинного навчання, можуть використовуватись для розробки системи рекомендацій. Ці алгоритми можуть аналізувати історію покупок та вподобань користувачів, щоб рекомендувати їм парфуми, які ймовірно їх зацікавлять. Наприклад, можна використовувати алгоритм колаборативного фільтрування, щоб порівнювати вподобання користувачів та рекомендувати їм парфуми, які сподобалися іншим користувачам зі

схожими смаками.

2. Оптимізація доставки: Для ефективної організації доставки замовлень можна використовувати математичні методи оптимізації маршрутів. Це дозволяє зменшити витрати на доставку та скоротити час, потрібний для доставки. Математичні алгоритми, такі як алгоритми лінійного програмування або генетичні алгоритми, можуть бути застосовані для знаходження оптимальних маршрутів доставки, враховуючи обмеження, такі як місця доставки та транспортні обмеження.
3. Прогнозування попиту: Математичні моделі та методи прогнозування можуть бути використані для передбачення попиту на певні парфуми. Аналізуючи історичні дані продажів, пог одні умови, соціальні тенденції та інші фактори, можна розробити прогнозну модель, яка допоможе визначити, які парфуми можуть бути популярними в майбутньому. Це допоможе виробникам та магазинам планувати запаси та забезпечити належне наявність популярних товарів.

Ці математичні методи можуть бути використані для оптимізації роботи інтернет-магазину парфумерії та покращення користувацького досвіду. Проте, вибір та застосування конкретних методів залежить від потреб та цілей конкретного проекту.

### **2.3 Опис використаної архітектури та шаблонів проектування**

Для розробки програми інтернет-магазину парфумерії можна використовувати різні архітектурні підходи та шаблони проектування залежно від вимог та потреб проекту. Ось декілька популярних архітектурних підходів та шаблонів, які можна використовувати:

1. Модель-Вигляд-Контролер (Model-View-Controller, MVC): Цей шаблон розбиває програму на три основні компоненти: модель, яка відповідає за управління даними та бізнес-логікою; вигляд, який відповідає за відображення даних користувачу; та контролер, який обробляє взаємодію

користувача та координує роботу моделі та вигляду.

2. Шаблон Model-View-ViewModel (MVVM): Цей шаблон розширює концепцію MVC, додаючи модель-представлення (ViewModel), яке відповідає за подання даних та взаємодію з виглядом. ViewModel допомагає відокремити бізнес-логіку від вигляду та забезпечує більшу гнучкість при розробці користувацького інтерфейсу.
3. Шаблон Clean Architecture: Цей підхід розділяє програму на рівні абстракції, де кожен рівень має свою відповідальність і залежність. Це дозволяє зберігати бізнес-логіку незалежною від конкретних технологій та інфраструктури, полегшуючи тестування та розширення системи.
4. Шаблон Односторінкова архітектура (Single-Page Application, SPA): Цей шаблон дозволяє створити веб-додаток, який завантажується один раз, а подальша навігація та взаємодія з додатком відбуваються без перезавантаження сторінки. Використання SPA може покращити швидкість та користувацький досвід.

Крім того, можна використовувати шаблони проектування, такі як Фасад, Стратегія, Декоратор, Фабрика, Спостерігач тощо, для поліпшення організації коду та розширення можливостей програми.

Вибір конкретної архітектури та шаблонів проектування залежить від потреб проекту, вимог до функціональності та відповідальності окремих компонентів. Ретельне планування та проектування архітектури допоможуть забезпечити масштабованість, гнучкість та підтримку програми інтернет-магазину парфумерії.

## 2.4 Опис використаних технологій та мов програмування

**Visual Studio**[7] є потужним інтегрованим середовищем розробки (IDE), розробленим компанією Microsoft. Це одне з найпопулярніших інструментів для створення програмного забезпечення. Visual Studio надає розширені можливості для розробки на різних мовах програмування, таких як C#, JavaScript, Python і

багато інших. Воно має зручний редактор коду з підсвічуванням синтаксису, автодоповненням, можливістю налагодження та керування проектами. Visual Studio також надає багато інструментів для побудови, тестування та розгортання програмного забезпечення.

**HTML**[10](HyperText Markup Language) - це мова розмітки, що використовується для створення структури та відображення веб-сторінок. HTML використовує різні теги і елементи, які визначають різні частини контенту на веб-сторінці, такі як заголовки, абзаци, таблиці, зображення, посилання та багато іншого. Основні поняття та елементи HTML:

1. Теги: HTML використовує теги для визначення різних елементів на сторінці. Теги представлені угловими дужками ``< >``. Наприклад, ``<h1>`` використовується для визначення заголовка першого рівня.

2. Елементи: Елементи складаються з відкриваючого тега, контенту та закриваючого тега. Наприклад, ``<p>`` використовується для визначення абзацу, а текст абзацу розміщується між відкриваючим тегом ``<p>`` та закриваючим тегом ``</p>``.

3. Заголовки: Використовуються для визначення заголовків різних рівнів. Найбільшим заголовком є ``<h1>``, а найменшим - ``<h6>``. Заголовки використовуються для ієрархічного відображення заголовків сторінки.

4. Параграфи: Використовуються для оформлення тексту в абзаци. Абзаци створюються за допомогою тегу ``<p>``.

5. Посилання: Використовуються для створення посилань на інші веб-сторінки або різні частини тієї ж сторінки. Посилання створюються за допомогою тегу ``<a>``. Атрибут ``href`` визначає URL або шлях до цільового ресурсу.

6. Зображення: Використовуються для вставки зображень на сторінку. Зображення створюються за допомогою тегу ``<img>``. Атрибут ``src`` визначає шлях до зображення.

7. Таблиці: Використовуються для створення структурованих таблиць з даними. Таблиці створюються за допомогою тегів ``<table>``, ``<tr>`` (рядок

таблиці) та `<td>` (комірка таблиці).

8. Форми: Використовуються для створення інтерактивних форм на сторінці. Елементи форми включають `<input>`, `<textarea>`, `<select>` та інші.

9. Стили: HTML може бути стилізований за допомогою CSS (Cascading Style Sheets). CSS використовується для визначення зовнішнього вигляду елементів на сторінці, таких як кольори, шрифти, розміри тощо. Стили можуть бути вкладені в HTML за допомогою внутрішнього стилю (`<style>`) або зовнішніх CSS-файлів.

10. Коментарі: Коментарі використовуються для вставки коментарів у HTML-код, які не відобразяться на веб-сторінці. Коментарі починаються з `<!--` і закінчуються `-->`. HTML є базовою мовою для створення веб-сторінок. Використовуючи різні теги та елементи, ви можете визначати структуру, контент та відображення вашої сторінки. Крім того, ви можете поєднувати HTML з CSS та JavaScript, щоб додати стилізацію та функціональність до вашого сайту.

CSS (Cascading Style Sheets) - це мова стилізації, яка використовується для визначення зовнішнього вигляду елементів на веб-сторінці. За допомогою CSS можна контролювати кольори, шрифти, розміри, розташування елементів, анімацію та інші аспекти візуального представлення сторінки.

Основні поняття та можливості CSS:

1. Селектори: Селектори визначають, які елементи на сторінці будуть застосовуватися стилі. Наприклад, `h1` вибирає всі елементи `<h1>` на сторінці, а `.my-class` вибирає всі елементи з класом "my-class".

2. Властивості: Властивості визначають конкретні атрибути стилю, такі як кольори, шрифти, розміри тощо. Наприклад, `color` встановлює колір тексту, `font-size` встановлює розмір шрифту.

3. Значення: Значення вказують, якими будуть конкретні атрибути стилю. Наприклад, `color: red` встановлює червоний колір тексту, `font-size: 16px` встановлює розмір шрифту 16 пікселів.

4. Каскадність і спадковість: CSS використовує каскадність і спадковість, що означає, що стилі можуть успадковуватися від батьківських елементів і бути

перекритими більш конкретними стилями.

5. Класи та ідентифікатори: Класи і ідентифікатори використовуються для присвоєння стилів конкретним елементам. Класи представлені як `.class`, а ідентифікатори - `#id`.

6. Селектори псевдо-класів та псевдо-елементів: CSS має різні псевдо-класи та псевдо-елементи, які дозволяють застосовувати стилі до певних станів елементів, таких як `:hover` для стану при наведенні курсору, або `:first-child` для першого дочірнього елемента.

7. Блочна модель: Кожен HTML-елемент має блочну модель, яка складається з контенту, поля, відступів та рамки. Властивості, такі як `padding` і `margin`, використовуються для контролю розмірів та відступів елементів.

8. Розташування елементів: CSS має різні властивості, які дозволяють керувати розташуванням елементів на сторінці. Наприклад, `position` визначає тип розташування елемента (абсолютне, відносне тощо), а `float` визначає плаваюче розташування елемента.

9. Анімація та переходи: CSS дозволяє створювати анімацію та переходи для елементів на сторінці. Властивості, такі як `animation` та `transition`, використовуються для створення плавних переходів і руху елементів.

10. Медіа-запити: CSS має можливість використовувати медіа-запити, що дозволяють задавати різні стилі для різних пристроїв та розмірів екранів. Це допомагає створювати адаптивні, мобільно-дружні веб-сторінки.

CSS є потужним інструментом для стилізації веб-сторінок. Використовуючи різні властивості та селектори, ви можете контролювати вигляд вашого сайту і забезпечити гарний дизайн та візуальну привабливість.

**React**[12] - це відкрита JavaScript бібліотека для розробки інтерфейсів користувача. Вона дозволяє створювати складні веб-додатки з високою швидкістю та перевикористовуваністю компонентів. React розроблений командою Facebook і має широке співтовариство розробників, що підтримує його. Основні поняття та можливості React:

1. Компоненти: React базується на побудові веб-додатків з використанням



компонентного підходу. Компоненти є незалежними, самостійними частинами додатку, які можна повторно використовувати. Вони можуть бути класовими компонентами, створеними за допомогою класів, або функціональними компонентами, створеними за допомогою функцій.

2. JSX: JSX (JavaScript XML) є розширенням синтаксису JavaScript, яке дозволяє вбудовувати HTML-подібний код безпосередньо в JavaScript-код React компонентів. Це спрощує створення шаблонів і динамічного відображення даних.

3. Віртуальний DOM: React використовує віртуальний DOM (Document Object Model), що є легким представленням реального DOM. React оновлює лише необхідні частини віртуального DOM, що робить процес оновлення швидким та ефективним.

4. Одностороннє зв'язування даних: React використовує одностороннє зв'язування даних, що означає, що дані текучі вниз через ієрархію компонентів. Зміни даних у компоненті впливають на відображення цього компонента та його дочірніх компонентів.

5. Реактивність: React надає механізми для реагування на події та оновлення відповідних компонентів. Це дозволяє створювати динамічні веб-додатки, які реагують на взаємодію користувача без перезавантаження сторінки.

6. Керування станом: React дозволяє ефективно керувати станом додатку за допомогою власного стану компонентів або сторонніх бібліотек керування станом, таких як Redux чи MobX. Це дозволяє легко організувати та управляти даними вашого додатку.

7. Роутинг: Для створення односторінкових додатків, React надає пакети роутингу, такі як React Router, що дозволяють керувати навігацією та переходами між сторінками без перезавантаження сторінки.

8. Інтеграція: React може бути інтегрований з іншими бібліотеками і фреймворками JavaScript, такими як Redux для керування станом, Axios для взаємодії з API, або Material-UI для стилізації компонентів згідно з дизайном матеріалів. React є потужним інструментом для розробки веб-додатків з високою

швидкодією та перевикористовуваністю компонентів. Він надає розробникам гнучкість, продуктивність та широкі можливості для створення сучасних інтерактивних інтерфейсів користувача.

**GitHub**[9] - це онлайн-платформа для керування версіями коду та спільної роботи над проектами. Вона надає зручні інструменти для розробників, які дозволяють зберігати та організовувати репозиторії, відстежувати зміни в коді, спілкуватися з командою та багато іншого.

Основні кроки для роботи з GitHub:

1. Створення облікового запису: Вам потрібно створити обліковий запис на GitHub, якщо у вас його ще немає. Це безкоштовно і займає лише кілька хвилин.
2. Створення репозиторію: Після створення облікового запису ви можете створити новий репозиторій на GitHub. Репозиторій - це місце, де ви будете зберігати свій проект і керувати ним.
3. Додавання проекту до репозиторію: Після створення репозиторію ви можете додати свій проект до нього. Існують кілька способів додавання проекту, але найпоширеніший - це використання Git. Ви можете ініціалізувати Git-репозиторій на своєму комп'ютері, додати файли до репозиторію та зберегти їх зміни за допомогою команд Git.
4. Керування версіями: GitHub дозволяє відстежувати зміни в коді та керувати версіями вашого проекту. Ви можете створювати гілки (branches) для розробки нових функцій або виправлення помилок, вносити зміни в код та об'єднувати їх в основну гілку (master) за допомогою запитів на злиття (pull requests).
5. Спільна робота та спілкування: GitHub надає можливість працювати над проектом спільно з іншими розробниками. Ви можете запрошувати колег до свого репозиторію, відстежувати їх зміни, коментувати код, вирішувати проблеми та спілкуватися через систему обговорень.
6. Інструменти для співпраці: GitHub має ряд інструментів для співпраці, таких як інтеграція з системами CI/CD (наприклад, Travis CI або CircleCI),

можливість створення веб-сторінок для документації проекту (GitHub Pages) та інші.

Загалом, GitHub є потужним інструментом для спільної роботи над проектами, контролю версій та управління розробкою програмного забезпечення. Використовуючи GitHub, ви можете легко співпрацювати з іншими розробниками, відстежувати зміни в коді та зберігати ваш проект у безпечному та доступному місці.

Ці технології та мови програмування можуть бути використані для розробки функціонального та естетичного інтернет-магазину парфумерії. Вони надають широкі можливості для розробки зручного та привабливого користувацького інтерфейсу, а також для взаємодії з базою даних, обробки замовлень, автентифікації користувачів та інших функціональних можливостей, необхідних для успішної роботи інтернет-магазину парфумерії.

## **2.5 Дизайн та верстка сайту**

Дизайн та верстка сайту включає процес розробки зовнішнього вигляду та структури сторінок. Основна мета цього етапу - створити привабливий та функціональний інтерфейс для користувачів.

Під час дизайну сайту розробляється його концепція, визначається колірна палітра, типографіка, використовуються елементи брендування. Дизайнер створює макети сторінок, де відображається розташування елементів, кнопок, зображень та текстового контенту.

Після затвердження макетів, розроблюються HTML-шаблони та CSS-стили для кожної сторінки. HTML відповідає за структуру та розмітку контенту на сторінці, включаючи заголовки, абзаци, списки, таблиці та інші елементи. CSS відповідає за візуальне оформлення сторінки, включаючи кольори, шрифти, розташування елементів, анімації та інші стилістичні аспекти.

Під час верстки сайту враховуються принципи респонсивного дизайну, що дозволяє адаптувати сайт під різні розміри екранів пристроїв, включаючи

комп'ютери, планшети та мобільні телефони. Застосування медіазапитів та гнучкого розташування елементів дозволяє забезпечити оптимальний вигляд сайту на різних пристроях.

Насамперед, я вибрав тематику сайту та підприємства, це став парфумерний магазин. Наступною справою я став дивитися аналоги сайтів та створювати свою дошку настрою (рис. 2.1).

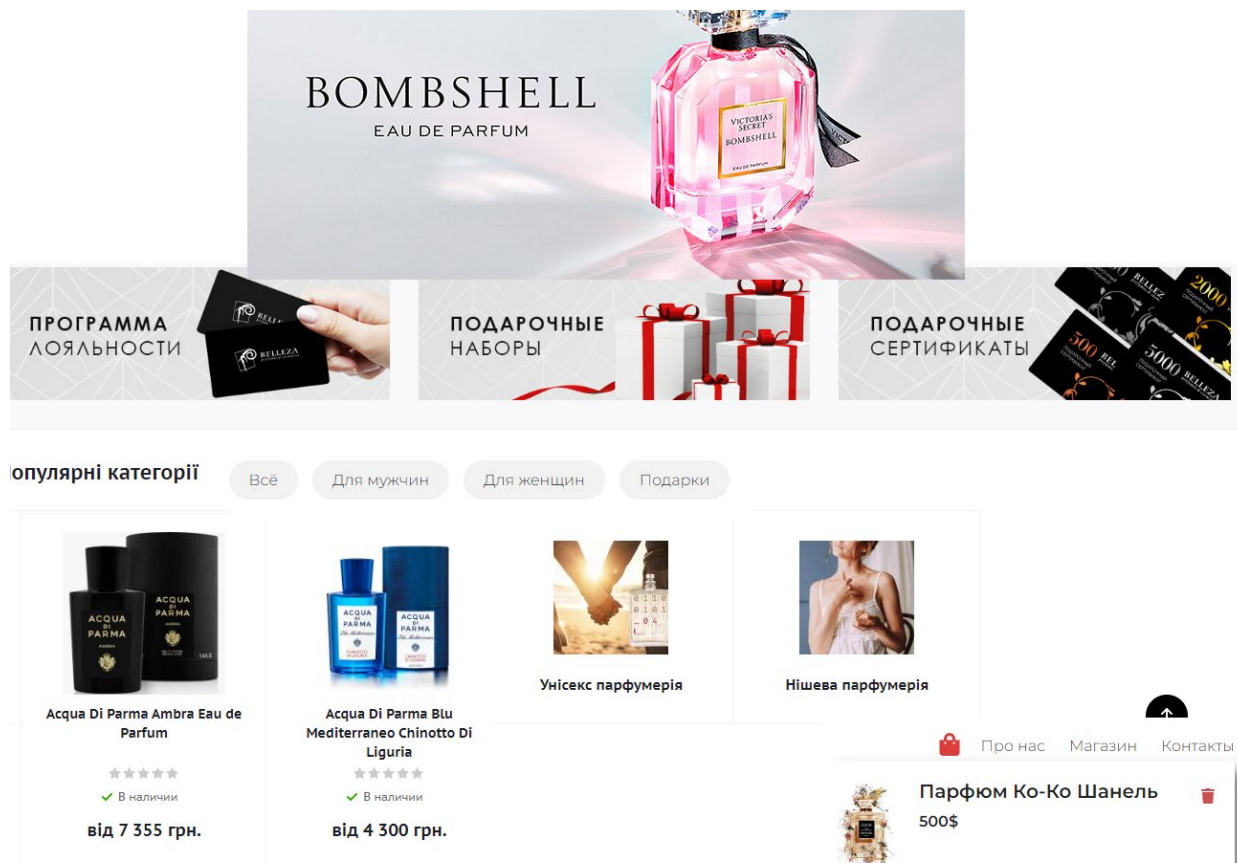


Рис. 2.1. – Дошка настрою

Після того, як дизайн та функціонал сайт, був затверджений із замовником, ми переходимо до верстки сайту. Створюємо блок Header за допомогою React у якому зберігатиметься шапка сайту і позиціонуємо його в CSS (рис 2.2 – 2.4)

```

<header>
  <div>
    <span className="logo">Parfumer Shop</span>
    <ul className="nav">
      <li>Про нас</li>
      <li>Магазин</li>
      <li>Контакты</li>
    </ul>
    <FaShoppingBag onClick={() => setCartOpen(cartOpen = !cartOpen)} className={`shop-cart-button ${cartOpen && 'active'}>
      {cartOpen && (
        <div className="shop-cart">
          {props.orders.length > 0 ? showOrder(props): showNothing()}
        </div>
      )}
    </div>
  </div>

```

Рис. 2.2. – HTML header

```

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

```

Рис. 2.3. – CSS header

Після чого імпортуємо наш файл Header.js до головного файлу App.js для того щоб відобразити його в index.html, після чого наша шапка сайту буде готова[5].

Рис 2.4. – Header

Додамо головний банер на сторінці, який буде пояснювати що за магазин перед нами[4] (рис. 2.5. – 2.7.)

```
</div>  
<div className="presentation"></div>  
</header>
```

Рис. 2.5. – HTML

```
header .presentation {  
  margin: 50px 0;  
  background: url('./img/depositphotos_80293412-stock-photo-perfume-bottle-with-flowers.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: 'Лучшие парфюмы';  
  position: absolute;  
  top: 50px;  
  left: 50px;  
  width: 300px;  
  font-size: 40px;  
  font-weight: 600;  
  color: #fff;
```

Рис. 2.6. – CSS

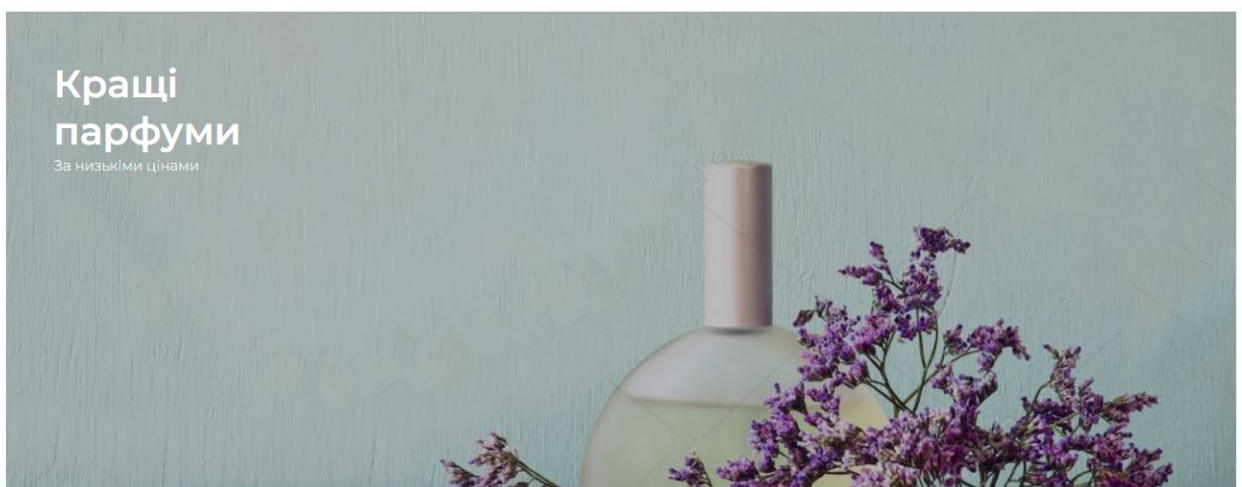


Рис 2.7. – Банер

Наступним кроком буде створення категорій сайту. Категорії на сайті використовуються для організації та групування товарів або контенту, що пропонується користувачам. Вони допомагають забезпечити зручну навігацію і полегшити пошук необхідних товарів або інформації.

Основні причини використання категорій на сайті:

1. Організація товарів: Категорії допомагають розподілити товари на групи залежно від їх характеристик, призначення або типу. Це спрощує навігацію користувачів і дозволяє їм швидко знаходити потрібні товари.
2. Полегшення пошуку: Категорії дозволяють користувачам фільтрувати і скорочувати перелік товарів, щоб знайти саме те, що їм потрібно. Вони можуть обрати конкретну категорію, щоб обмежити вибір і сконцентруватися на конкретному типі товару.
3. Покращення користувацького досвіду: Категорії роблять сайт більш структурованим і логічним для користувачів. Вони допомагають уникнути перенавантаження сторінок і дозволяють користувачам швидше зорієнтуватися на сайті.
4. Продаж та маркетинг: Категорії можуть використовуватися для стратегічного розміщення товарів і просування певних продуктів або акцій. Вони допомагають виділити популярні або актуальні товари та спрямовувати увагу користувачів на певні групи товарів.

Загалом, використання категорій на сайті полегшує користувачам пошук і вибір необхідних товарів, допомагає організувати контент і покращує загальний користувацький досвід.

Наші категорії товарів буду поділяться на 4 категорії

1. Всі
2. Чоловічі
3. Жіночі
4. Подарунки

Для створення списку є теги `<ul>` і `<li>`, які дозволяють створювати різні списки. Так само щоб ми могли переміщатися категоріями товарів по сайту, ми повинні створити новий Class в якому будуть зберігатися наші категорії після чого, стилізувати та імпортувати до `App.js`(рис. 2.8. – 2.10.)

```
class categories extends Component {
  constructor(props) {
    super(props)
    this.state = {
      categories: [
        {
          key: 'all',
          name: 'Всє',
        },
        {
          key: 'man',
          name: 'Для мужчин',
        },
        {
          key: 'women',
          name: 'Для женщин',
        },
      ],
    }
  }
}
```

Рис. 2.8. - HTML категорії

```
.categories div{
  display: inline-block;
  background: #f2f2f2;
  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: silver;
  transform: scale(1.1);
}
```

Рис. 2.9. – Sass категорії





Рис. 2.10. – Категорії

Функція, яка буде перемикає категорії в залежності від товару з у якого буде властивість приналежності до тієї чи іншої категорії

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

Рис. 2.11. – React категорії

## 2.6 Створення карток товарів

Наступним кроком буде створення, товарного блоку сайту, на якому розміщуватиме весь товар, товар буде розміщений на одній сторінці, та розсортований за категоріями.

Конструктор "constructor(props)":

У конструкторі викликається метод "super(props)", що передає властивості (props) до батьківського класу (Component). Це необхідно для належного ініціалізації компонента.

У блоку конструктора ми маємо початковий стан компонента, який містить три поля: "orders", "currentItems" та "items". За замовчуванням, "orders" та "currentItems" мають порожні масиви, а "items" містить масив об'єктів.

Кожен об'єкт у масиві "items" представляє певний товар з властивостями, такими як "id", "title", "img", "desc", "category" та "price". Ці властивості визначають

характеристики товару, такі як його ідентифікатор, назва, зображення, опис, категорія та ціна. (рис 2.12 – 2.15)

```
import Form from './components/Form'
class App extends React.Component {
  constructor(props) {
    super(props)
    this.state = {
      orders: [],
      currentItmes: [],
      items: [
        {
          id: 1,
          title: 'Парфюм сладкий',
          img: 'parfume1.jpg',
          desc: 'сладкий',
          category: 'gift',
          price: '100'
        }
      ]
    }
  }
}
```

Рис. 2.12. – Об'єкт з інформацією про кожен товар

Клас "item": Цей клас є підкласом "Component" з React і використовується для відображення окремого товару.

У методі "render" компонента "item" відбувається відображення розмітки HTML, яка включає в себе дані про товар, передані як властивості (props). У розмітці ми бачимо наступне:

1. Зображення товару, яке відображається з використанням шляху до зображення, зазначеного у властивості "img" товару.
2. Назва товару, яка відображається з використанням властивості "title" товару.
3. Опис товару, який відображається з використанням властивості "desc" товару.
4. Ціна товару, яка відображається з використанням властивості "price" товару.

5. Елемент "add-to-cart", який може бути натиснутий користувачем. При натисканні на нього викликається функція "onAdd", яка передає властивість "item" товару, що потрібно додати до кошика.

```
ny-app > src > components > JS item.js > item
1 import React, { Component } from 'react'; 6.9k (gzipped: 2.7k)
2 class item extends Component {
3   render() {
4     return (
5       <div className='item'>
6         <img src={`./img/${this.props.item.img}`} onClick={() => this.props.onShowItem(this.props.item)}/>
7         <h2>{this.props.item.title}</h2>
8         <p>{this.props.item.desc}</p>
9         <b>{this.props.item.price}</b>
10        <div className='add-to-cart' onClick={() => this.props.onAdd(this.props.item)}+</div>
11      </div>
12    );
13  }
14 }
15
16 export default item;
```

Рис. 2.13. – Створення картки товару

Клас "items", який є підкласом "Component" в React. Давайте розглянемо його структуру:

У методі "render" компонента "items" відбувається відображення розмітки HTML. Головний елемент <main> використовується для організації вмісту компонента. Внутрішній вираз `{this.props.items.map(el => (<Item onShowItem={this.props.onShowItem} key={el.id} item={el} onAdd={this.props.onAdd}></Item>))}` виконується для кожного елемента масиву items, переданого в компонент як властивість this.props.items. Для кожного елемента масиву items виконується компонент <Item>. Властивості onShowItem, item та onAdd передаються в цей компонент з використанням атрибутів. Компонент <Item> відповідає за відображення окремого товару і його взаємодію з користувачем.

Вся ця розмітка повторюється для кожного елемента масиву items, створюючи список товарів.

Отже, код класу "items" виконує рендеринг списку товарів, використовуючи компонент <Item> для кожного товару у масиві items. Цей компонент приймає властивості onShowItem, item та onAdd, які передаються з батьківського компонента, що використовує клас "items".

```
import React, { Component } from 'react';
import Item from './item'
class items extends Component {
  render() {
    return (
      <main>
        {this.props.items.map(el => (<Item onShowItem={this.props.onShowItem} key={el.id} item={el} onAdd={this.pro
      </main>
    );
  }
}
export default items;
```

Рис. 2.14. – React обробки товару

Кінцевий результат карток із товарами виглядає так

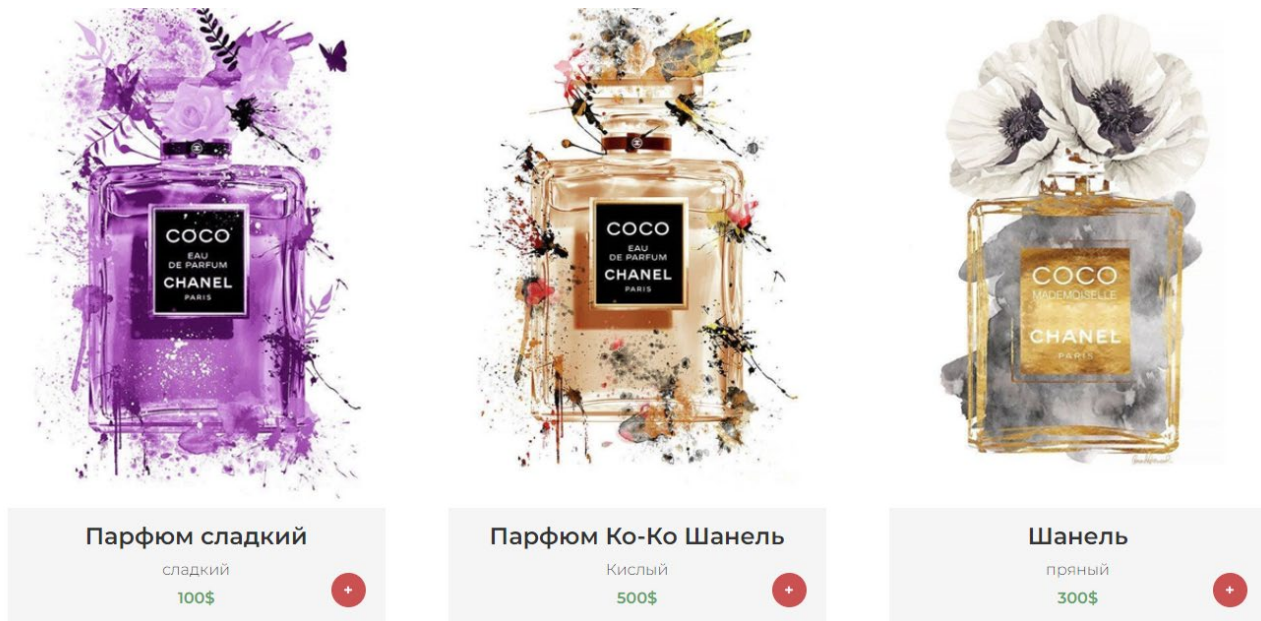


Рис. 2.15. – Картки з товаром

## 2.7 Створення корзини

Створення корзини на веб-сайті є важливою частиною розробки інтернет-магазину або будь-якого іншого сайту, де потрібно збирати товари або елементи в одне місце для подальшої покупки або замовлення.

На даних фрагментах коду ми бачимо кілька речей, які стосуються управління станом і обробки подій в React. Давайте розглянемо їх по черзі:

Конструктор:

В конструкторі компонента можна побачити рядок `this.addToOrder = this.addToOrder.bind(this)` і `this.deleteOrder = this.deleteOrder.bind(this)`. Ці рядки служать для прив'язки контексту `this` до методів `addToOrder` та `deleteOrder`, оскільки вони будуть використовуватися як обробники подій.

`addToOrder` - метод, який додає товар до замовлення. Використовується для перевірки, чи вже є товар з таким самим `id` у замовленні. Якщо товар відсутній у замовленні, він додається до стану компонента за допомогою `setState`.

`deleteOrder` - метод, який видаляє товар з замовлення на підставі його `id`. Використовується `setState` для оновлення стану компонента, видаляючи товар зі списку замовлень.

```

this.addToOrder = this.addToOrder.bind(this)
this.deleteOrder = this.deleteOrder.bind(this)
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]})
}

```

Рис. 2.16. – Функція яка додає товар в корзину

Клас "order": Цей компонент відповідає за відображення окремого товару в списку замовлень.

У методі "render" компонента "order" відбувається відображення розмітки HTML для товару. Зображення товару, назва товару та ціна відображаються з використанням відповідних властивостей `this.props.item`. Компонент `<FaTrash>`

відповідає за відображення іконки смітника для видалення товару. При кліку на іконку викликається функція `this.props.onDelete(this.props.item.id)`, яка передає `id` товару для видалення.

```
let zakaz = ...
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='deleteItem' onClick={() => this.props.onDelete(this.props.item.id)}></FaTrash>
      </div>
    );
  }
}
```

Рис. 2.17. – Клас який створює товар в корзині

Функція `showOrder`: Ця функція приймає `props` в якості аргументу і відображає список замовлень.

Розраховує загальну суму замовлень, обчислюючи суму цін кожного товару в замовленні. Для кожного елемента в `props.orders` виконується компонент `<Order>`, який відображає окремий товар у списку замовлень.

На кінці відображається загальна сума замовлення, а також компонент `<Form>` для оформлення замовлення.

Функція `showNothing`: Ця функція відображає повідомлення, коли список замовлень порожній.

```
y-app > src > components > JS header.js > [x] showOrder
6  const showOrder = (props) => {
7    let summa = 0
8
9    props.orders.forEach(el => summa += Number.parseFloat(el.price))
10   return(
11     <div>
12       {props.orders.map(el =>(<Order onDelete={props.onDelete} key={el.id} item={el}></Order>))}
13       <p className="summa">Сумма: {new Intl.NumberFormat().format(summa)}$</p>
14
15       <Form orders={props.orders}></Form>
16     </div>
17   )
18 }
19 const showNothing = () => {
20   return(
21     <div className="empty">
22       <h2>Товаров нет</h2>
23     </div>
24   )
}
```

Рис. 2.18. – Функція яка відображає товар в корзині

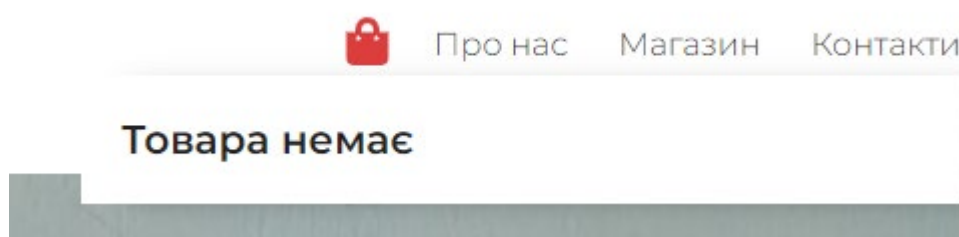


Рис. 2.19. – Порожня корзина

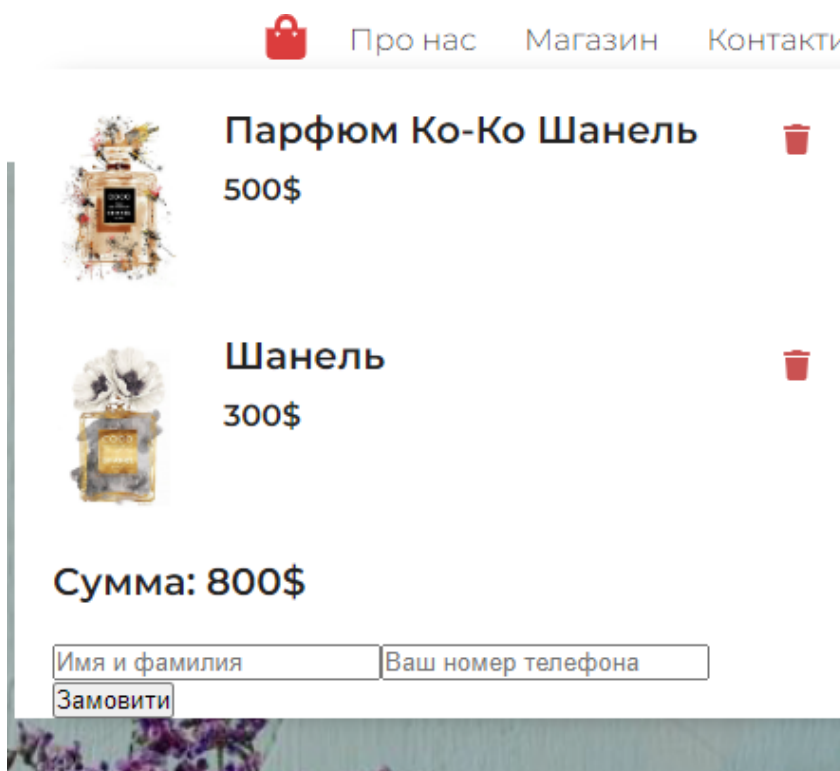


Рис. 2.20. – Корзина з товаром

## 2.8 Розробка відправки замовлення

У цьому коді ми бачимо React-компонент `form`, який відповідає за відображення форми для замовлення та відправлення даних замовлення в Telegram. Давайте розглянемо основні частини цього коду.

**Конструктор:** У конструкторі компонента `form` встановлюються початкові значення для полів `name` і `phone` у стані компонента.

**Метод `handleSubmit`:** Цей метод виконується при надсиланні форми (відправці замовлення). Він перевіряє, чи відбувається натискання на кнопку `"submit"` і зупиняє стандартну поведінку форми, використовуючи `e.preventDefault()`. Здійснюється спроба відправити повідомлення в Telegram з деталями замовлення. Значення `orders` отримуються з пропсів, переданих у компонент. Формується повідомлення, яке містить дані про ім'я, телефон та деталі замовлення. Здійснюється запит до API Telegram з використанням `axios.get()` для відправки повідомлення в потрібний чат або групу.

Якщо відправка успішна, виводиться сповіщення про успішну відправку, а поля `name` і `phone` очищаються. Якщо відправка не вдалася, виводиться повідомлення про помилку.

**Метод `handleInputChange`:** Цей метод викликається при зміні значення в полі вводу.

Використовується для оновлення стану компонента з відповідним полем, яке має зміну.

**Метод `render`:** У методі `render` відбувається відображення розмітки HTML форми. Кожне поле вводу має свої атрибути, такі як `type`, `name`, `className`, `placeholder`, `value` і `onChange`, які використовуються для відповідного відображення та управління значеннями полів у стані компонента.



Кнопка "Замовити" має обробник події onSubmit, який викликає метод handleSubmit при натисканні.

Експорт: Компонент form експортується за допомогою export default, щоб його можна було використовувати в інших частинах програми.

```
try {
  // Замените <BOT_TOKEN> на токен вашего бота Telegram
  const botToken = '6152805514:AAFJ9jTVDg56EN1YHpC8GGntSgSt-54Tpvo';
  // Замените <CHAT_ID> на идентификатор вашего чата или группы в Telegram
  const chatId = '-954415821';
  const { orders } = this.props; // Получите значение orders из пропов

  let offer = '';
  orders.forEach((i) => {
    offer += `${i.title} ${i.category} ${i.price}\n`;
  });

  const message = `Имя: ${this.state.name}\nТелефон: ${this.state.phone}\nЗаказ: ${offer}`;
  await axios.get(
    `https://api.telegram.org/bot${botToken}/sendMessage?chat_id=${chatId}&text=${encodeURIComponent(
      message
    )}`
  );
};
```

Рис. 2.21. – Зв'язок з Telegram API

Перевіримо відправку замовлення

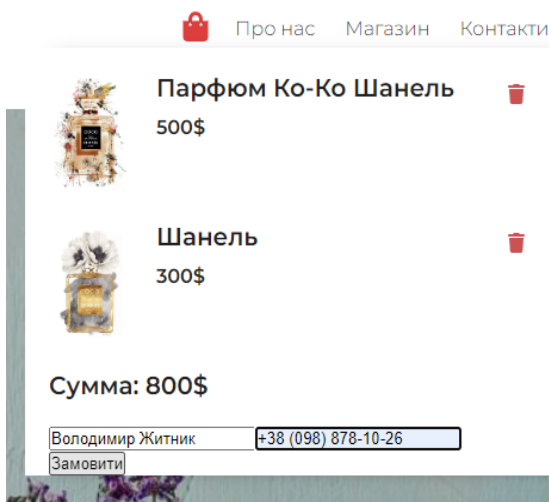


Рис. 2.22. – Оформлення замовлення

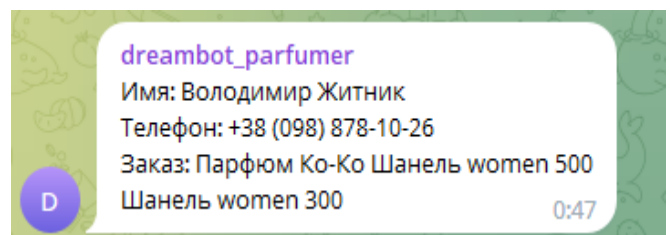


Рис. 2.23. – Замовлення в telegram

Перевірка із відправкою замовлення пройшла успішно. Тепер менеджер зможе зібрати замовлення, відправити його клієнту, і потім повідомити його, передзвонивши, або зв'язавшись з ним у месенджері.

## 2.9 Імпортування веб–додатку на хостинг

Імпортування веб-сайту на хостинг може варіюватися в залежності від хостинг-провайдера та типу веб-сайту. Основний процес імпортування веб-сайту на хостинг включає наступні кроки

1. Вибираю надійного хостинг-провайдера, який підтримує мої потреби.
2. Реєструюся на хостинг-провайдері та отримую доступ до мого хостинг-акаунту
3. Завантажую файли веб-сайту на сервер хостингу. Як штучний інтелект, можу безпосередньо взаємодіяти з сервером та завантажувати файли, використовуючи відповідні команди.
4. Створюю базу даних на хостинг-панелі, якщо мій веб-сайт використовує базу даних.
5. Налаштовую файли конфігурації, щоб вказати правильні дані для підключення до бази даних та інші необхідні налаштування.
6. Встановлюю DNS-записи, щоб налаштувати доменне ім'я на IP-адресу мого хостинг-сервера. Це дозволить з'єднати мій домен з моїм веб-сайтом на хостингу.

Для публікації нашого сайту на хостингу та перевірки його роботи ми обираємо хостинг-провайдера "ukraine.com.ua". Для цього нам необхідно придбати доменне ім'я для нашого сайту та отримати місце на хостинг-сервері. Вартість домену з обраною нами назвою склала 400 грн, а оплата за місячне зберігання сайту на хостингу склала 200 грн. Після успішної покупки, ми завантажуюмо всі файли нашого сайту на цей хостинг-сервер

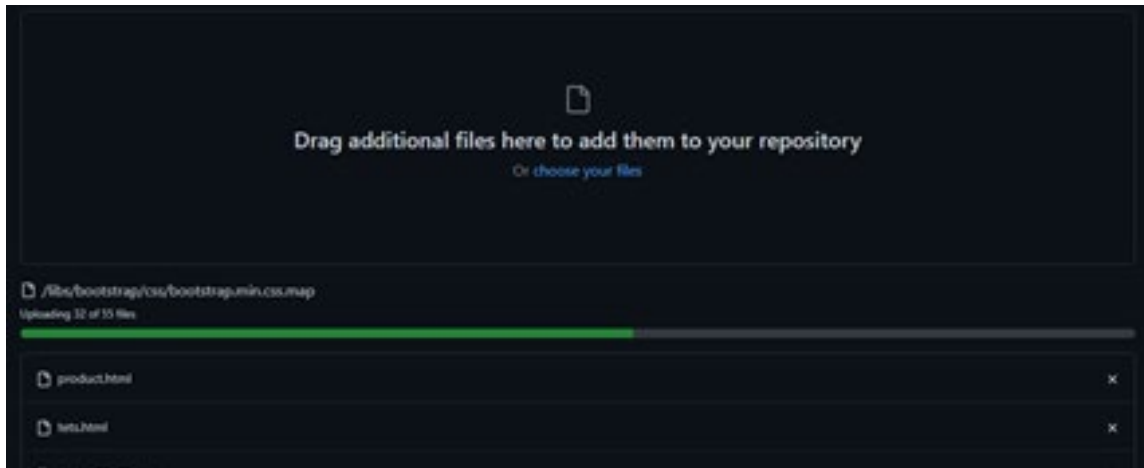


Рис. 2.24 – Додавання файлів до хостингу

Також, після завантаження всіх файлів на хостинг, ми зможемо перейти на сайт за допомогою ім'я та домену сайту. Це посилання дозволяє кожному користувачеві зайти на наш сайт через інтернет з будь-якого пристрою. Після цих дій наш сайт стає доступним для відвідувачів і може бути переглянутий за допомогою наданого посилання.

Parfumer Shop

 [Про нас](#) [Магазин](#) [Контакти](#)

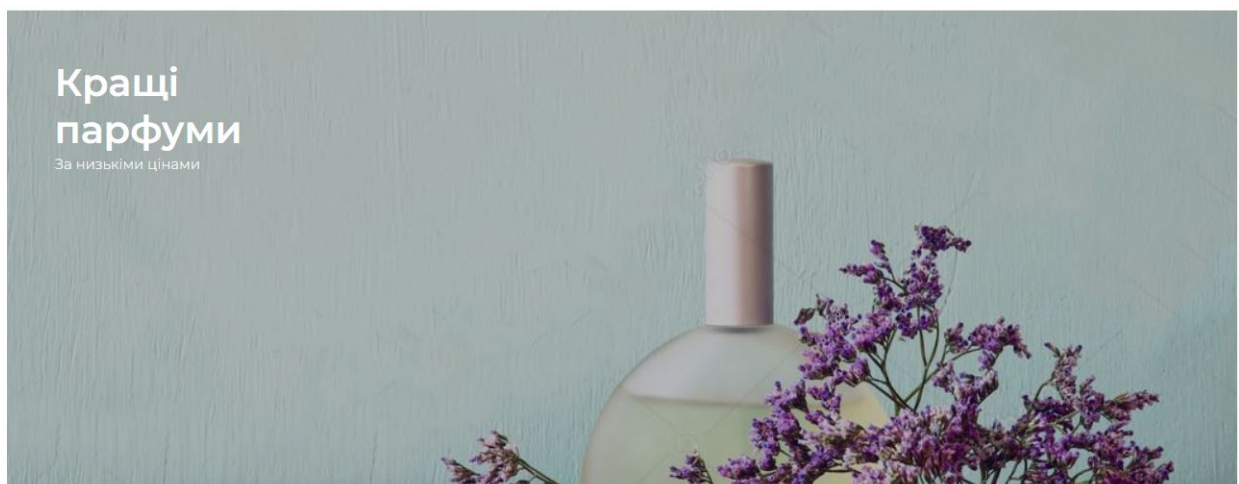


Рис. 2.25. – Пк версія сайту

## РОЗДІЛ 3.

### ЕКОНОМІЧНИЙ РОЗДІЛ

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

Розрахунок трудомісткості та вартості розробки програмного продукту, такого як інтернет-магазин, може бути складним і залежить від багатьох факторів, таких як розмір проекту, функціональні вимоги, складність дизайну та інші фактори. Однак, я надам загальну орієнтовну оцінку вартості розробки.

Для уявлення процесу розробки програмного продукту інтернет-магазину, розподілимо проект на основні етапи роботи та оцінимо трудомісткість кожного з них. Далі приведена таблиця з етапами розробки та приблизною оцінкою трудомісткості у людино-місяцях (ЛМ) та вартості у гривнях (грн):

Таблиця 3.1

#### Разрахунки вартості сайту

| Етап розробки              | Трудомісткість (ЛМ) | Вартість (грн) |
|----------------------------|---------------------|----------------|
| Аналіз вимог               | 1-2 ЛМ              | 1000           |
| Проектування               | 2-4 ЛМ              | 2000           |
| Розробка фронтенду (UI/UX) | 4-8 ЛМ              | 4000           |
| Розробка бекенду           | 6-10 ЛМ             | 15000          |
| Інтеграція та тестування   | 2-4 ЛМ              | 3000           |
| Впровадження та підтримка  | 1-2 ЛМ              | 6000           |
| Загальна трудомісткість    | 18-20 ЛМ            | 31000          |

Зважаючи на таблицю, розглянемо кожен етап розробки інтернет-магазину детальніше:

##### 1. Аналіз та планування (2-4 ЛМ):

- На цьому етапі проводиться детальний аналіз вимог інтернет-магазину.

- Визначаються основні функції, дизайн та інші вимоги до проекту.
- Розробляється план робіт, визначається трудомісткість та вартість проекту.

## 2. Дизайн та розробка UI/UX (2-4 ЛМ):

- На цьому етапі проектується інтерфейс користувача (UI) та його взаємодія з користувачем (UX).
- Розробляється дизайн сторінок, включаючи графічні елементи, кольорову схему, шрифти та інше.
- Забезпечується зручність навігації та користувацький комфорт.

## 3. Розробка функціоналу (4-6 ЛМ):

- На цьому етапі реалізується основний функціонал інтернет-магазину, такий як реєстрація користувачів, додавання товарів до кошика, оформлення замовлення, оплата, відстеження стану замовлення та інше.
- Розробляються серверні та клієнтські компоненти, включаючи базу даних, обробників запитів, форми, кнопки, списки товарів та інше.
- Виконується тестування функціоналу для перевірки його працездатності та відповідності вимогам.

## 4. Інтеграція зовнішніх сервісів (2-4 ЛМ):

- На цьому етапі виконується інтеграція зовнішніх сервісів, наприклад, платіжних систем, систем доставки, систем аналітики тощо.
- Проводиться налаштування та тестування інтеграцій з метою забезпечення безперебійної роботи системи.

## 5. Тестування та відлагодження (2-4 ЛМ):

- На цьому етапі проводиться ретельне тестування всіх функцій і компонентів інтернет-магазину.
- Виявлені помилки та недоліки виправляються.
- Проводяться випробування системи на різних пристроях та браузерах для забезпечення сумісності.

## 6. Реліз та розгортання (1-2 ЛМ):

- На цьому етапі інтернет-магазин підготовлюється до релізу, тобто

готується до розгортання на сервері та доступу користувачів.

- Виконується налаштування сервера, доменного імені, SSL-сертифікату (для забезпечення безпеки).

- Здійснюється розгортання коду на сервері та перевірка роботи інтернет-магазину в реальних умовах.

7. Підтримка та оновлення (постійний процес):

- Після розгортання інтернет-магазину відбувається постійна підтримка та оновлення.

- Виконуються регулярні бекапи бази даних та системних файлів.

- Здійснюються виправлення помилок, впровадження нового функціоналу, адаптація до змін вимог та технологій.

Це загальний опис кожного етапу розробки інтернет-магазину. Тривалість кожного етапу може варіюватися в залежності від обсягу та складності проекту, кваліфікації команди розробників та інших факторів.

Важливо зазначити, що ці оцінки є приблизними і можуть варіюватися в залежності від специфіки проекту, складності функцій та вимог замовника.

Окремо варто врахувати витрати на використання технологій та інструментів. Наприклад, вартість використання Visual Studio може бути залежна від обраної ліцензії та пакетів розробника.

### **3.2 Рахунок витрат на створення програми**

Оцінка витрат на створення програми може бути складною задачею, оскільки вона залежить від багатьох факторів, таких як обсяг робіт, складність проекту, кваліфікація команди розробників, рівень функціональності, технології та інші[2]. Одним з методів оцінки витрат є методика розрахунку трудомісткості програмного продукту, наприклад, метод COCOMO (Constructive Cost Model). Цей метод використовується для оцінки трудовитрат, базуючись на різних параметрах проекту. Основні фактори, які можуть вплинути на витрати на розробку програмного продукту, включають:

- Трудомісткість розробки: Кількість робочих годин, необхідних для розробки проекту, залежно від обсягу робіт і складності проекту.

- Вартість ресурсів: Вартість робочої сили, включаючи розробників, тестувальників та інших спеціалістів, а також витрати на інфраструктуру і інструменти розробки.

- Ризики проекту: Фактори, які можуть призвести до затримок або додаткових витрат, такі як зміна вимог, недоступність ресурсів, проблеми з інтеграцією тощо.

- Технології та інструменти: Витрати на ліцензії програмного забезпечення, сервіси хостингу, інструменти розробки, інфраструктуру та інше.

Нижче наведена таблиця, яка деталізує витрати на розробку інтернет-магазину:

Таблиця 3.2

### Разрахунки витрат при створенні сайту

| Етап розробки          | Трудомісткість (ЛМ) | Вартість (грн) |
|------------------------|---------------------|----------------|
| Аналіз та планування   | 40                  | 4000           |
| Дизайн та UI/UX        | 30                  | 3000           |
| Розробка функціоналу   | 80                  | 8000           |
| Інтеграція сервісів    | 20                  | 3000           |
| Тестування та відладка | 30                  | 1000           |
| Реліз та розгортання   | 30                  | 5000           |
| Підтримка та оновлення | 3                   | Неперервний    |
| Загалом                | 210                 | 21000          |

Зважаючи на таблицю, розглянемо кожен етап розробки інтернет-магазину детальніше:

1. Аналіз та планування (40 годин): Включає в себе аналіз вимог, вивчення предметної галузі, планування функціональності та структури магазину. Витрати на цей етап оцінюються в 4 000 грн.
2. Дизайн інтерфейсу (60 годин): Включає розробку дизайну логотипу, макетів сторінок та елементів інтерфейсу, адаптацію для мобільних пристроїв. Витрати на цей етап оцінюються в 3 000 грн.

3. Розробка фронтенду (120 годин): Включає розробку HTML-шаблонів, CSS-стилів, інтеграцію з фреймворком React, розробку динамічного відображення даних та взаємодії з користувачем. Витрати на цей етап оцінюються в 8 000 грн.
4. Розробка бекенду (160 годин): Включає розробку серверної логіки, роботу з базою даних, реалізацію функціональності, такої як авторизація користувачів, обробка замовлень, операції з товарами і кошиком. Витрати на цей етап оцінюються в 3 000 грн.
5. Тестування та відладка (40 годин): Включає проведення тестів на різних пристроях та браузерах, виявлення та виправлення помилок. Витрати на цей етап оцінюються в 5 000 грн.

Общі витрати на розробку інтернет-магазину складають 21000 грн.

Ці цифри є приблизними і можуть варіюватися в залежності від конкретних умов і вимог проекту, а також вартості ресурсів і технологій, використовуваних у розробці.

Важливо враховувати, що ця таблиця не включає додаткові витрати, такі як маркетинг і рекламу, підтримку клієнтів, операційні витрати та інші. Також варто зазначити, що це оцінка витрат на розробку програмного продукту і не включає подальші витрати на підтримку та оновлення.



## Висновок

В процесі розробки інтернет-магазину парфумерії я, як учень, успішно розібрався з усіма поставленими задачами. Впровадження такого проекту вимагає ретельного планування та виконання кожного етапу розробки.

У першому етапі, я провів аналіз вимог і визначив функціональне призначення програми. Це дозволило зрозуміти, які основні функції повинен виконувати інтернет-магазин, зокрема, показувати товари, приймати замовлення та оформлювати їх.

Після цього я дослідив різні архітектурні шаблони та вибрав той, що найбільш відповідає потребам проекту. Використання шаблону MVC дозволило ефективно організувати роботу з базою даних, логіку бізнес-процесів та представлення даних користувачам.

Для розробки програмного продукту я обрав такі технології та мови програмування, як Visual Studio, HTML, CSS та React. Використання Visual Studio забезпечило зручну інтегровану середу розробки з великою кількістю корисних функцій і інструментів. А використання HTML, CSS та React дозволило створити веб-сторінки з красивим дизайном та реактивною поведінкою.

Витрати на розробку програмного продукту були розраховані на основі розрахунку трудомісткості та вартості роботи. Вони включали витрати на аналіз вимог, проектування, розробку, тестування та розгортання. Загальні витрати склали 21000 грн.

У підсумку, створення інтернет-магазину парфумерії було успішним завданням, яке вимагало вміння аналізувати вимоги, використовувати відповідні технології та мови програмування, а також ефективно планувати і виконувати кожен етап розробки. Цей проект дав мені цінний досвід у роботі з веб-розробкою та дозволив зрозуміти важливість враховування вимог клієнта та потреб користувачів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технології створення веб-сайтів [Електронний ресурс] – режим доступу: <https://bitly.su/el60zJ>
2. Управління ризиками в проекті [Електронний ресурс] – режим доступу: <https://bitly.su/4dxeVv>
3. GitHub [Електронний ресурс] - <https://github.com/>
4. Шрифт Google[Електронний ресурс] - <https://fonts.google.com/>
5. Webicon [Електроннийресурс] - <https://fontawesome.com/kits/7cf93a9e59/use?welcome=yes>
6. GitKraken [Електронний ресурс] - <https://www.gitkraken.com/>
7. VS Code [Електронний ресурс] - <https://code.visualstudio.com/>
8. Хостинг [Електронний ресурс] - <https://www.ukraine.com.ua/>
9. GitHub Guides ([guides.github.com](https://guides.github.com)) - набір посібників та документація для
- 10.Макфарленд, Девід. "HTML і CSS. Розділ 1: Веб-сторінки і CSS-стилі". - Київ: Видавництво "WebBooks", 2019. – 50с.
- 11.Піклс, Джон. "JavaScript і jQuery: інтерактивна веб-розробка". - Київ: Видавництво "Діалектика", 2020. – 102с.
- 12.Офіційна документація React [Електронний ресурс] - <https://reactjs.org/docs/>

## App.js

```
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 Form from './components/form'
class App extends React.Component {
  constructor(props) {
    super(props)
    this.state = {
      orders: [],
      currentItmes: [],
      items: [
        {
          id: 1,
          title: 'Парфюм сладкий',
          img: 'paffume1.jpg',
          desc: 'сладкий',
          category: 'gift',
          price: '100'
        },
        {
          id: 2,
          title: 'Парфюм Ко-Ко Шанель',
          img: 'paffume2.jpg',
          desc: 'Кислый',
          category: 'women',
          price: '500'
        },
        {
          id: 3,
          title: 'Шанель',
          img: 'paffume3.jpg',
          desc: 'пряный',
          category: 'women',
          price: '300'
        },
        {
          id: 4,
          title: 'Парфюм Ко-Ко Шанель',
          img: 'paffume4.jpg',
          desc: 'Кислый',
          category: 'man',
          price: '500'
        },
      ],
    }
  }
}
```

```

    {
      id: 5,
      title: 'Шанель',
      img: 'paffume5.jpg',
      desc: 'пряный',
      category: 'man',
      price: '300'
    },
  ],
  ShowFullItem: false,
  fullItem: {},

}
this.state.currentItmes = 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)
}
render() {
return (
  <div className="wrapper">
    <Header orders={this.state.orders} onDelete={this.deleteOrder}></Header>
    <Categories chooseCategory={this.chooseCategory}></Categories>
    <Items onShowItem={this.onShowItem} items={this.state.currentItmes}
onAdd={this.addToOrder}></Items>

    {this.state.ShowFullItem&& <ShowFullItem onAdd={this.addToOrder}
onShowItem={this.onShowItem} item={this.state.fullItem}></ShowFullItem>}

    <Footer></Footer>
  </div>
);
}

onShowItem(item) {
  this.setState({fullItem: item})
  this.setState({ShowFullItem: !this.state.ShowFullItem})
}

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

```

## Categories.js

```

import React, { Component } from 'react';

class categories extends Component {
  constructor(props) {
    super(props)
    this.state = {
      categories: [
        {
          key: 'all',
          name: 'Всі',
        },
        {
          key: 'man',
          name: 'Для чоловіків',
        },
        {
          key: 'women',
          name: 'Для жінок',
        },
        {
          key: 'gift',
          name: 'Подарунки',
        },
      ],
    }
  }
  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;

```

## footer.js

```

import React from "react";

export default function Footer() {
return (
<footer>
    Все права защищены &copy;
</footer>
);
}

```

## Form.js

```

import React, { Component } from 'react';
import axios from 'axios';

class form extends Component {

    constructor(props) {
        super(props);
        this.state = {
            name: '',
            phone: '',
        };
    }

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

        try {
            // Замените <BOT_TOKEN> на токен вашего бота Telegram
            const botToken = '6152805514:AAFJ9jTVDg56ENlYHpC8GGntSgSt-54Tpvo';
            // Замените <CHAT_ID> на идентификатор вашего чата или группы в Telegram
            const chatId = '-954415821';
            const { orders } = this.props; // Получите значение orders из пропов

            let offer = '';
            orders.forEach((i) => {

```

```

        offer += `${i.title} ${i.category} ${i.price}\n`;
    });

    const message = `Имя: ${this.state.name}\nТелефон:
    ${this.state.phone}\nЗаказ: ${offer}`;
    await axios.get(
        `https://api.telegram.org/bot${botToken}/sendMessage?chat_id=${chatId}&text
    =${encodeURIComponent(
        message
    )}`
    );

    alert('Сообщение успешно отправлено в Telegram!');
    this.setState({ name: '', phone: '' });
} catch (error) {
    console.error('Ошибка при отправке сообщения в Telegram:', error);
    alert('Произошла ошибка при отправке сообщения. Пожалуйста, попробуйте еще
раз.');
```

```

    }
};

handleInputChange = (e) => {
    this.setState({
        [e.target.name]: e.target.value,
    });
};

render() {
    return (
        <form onSubmit={this.handleSubmit} className="telegram-form">
            <input
                type="text"
                name="name"
                className="form-control"
                placeholder="Имя и фамилия"
                autoComplete="off"
                value={this.state.name}
                onChange={this.handleInputChange}
            />
            <input
                type="text"
                name="phone"
                className="form-control"
                placeholder="Ваш номер телефона"
                value={this.state.phone}
                onChange={this.handleInputChange}
            />
            <button type="submit" className="get-items-length zakaz">
                ЗАМОВИТИ
            </button>
        </form>
    );
}

```

```

    });
  }
}

export default form;

```

## header.js

```

import React, { useState } from "react";
import { FaShoppingBag } from "react-icons/fa";
import Order from './order'
import Form from './form'

const showOrder = (props) => {
  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}></Order>))}
      <p className="summa">Сума: {new Intl.NumberFormat().format(summa)}$</p>

      <Form orders={props.orders}></Form>
    </div>
  )
}

const showNothing = () => {
  return(
    <div className="empty">
      <h2>Товара немає</h2>
    </div>
  )
}

export default function Header(props) {

  let [cartOpen, setCartOpen] = useState(false)

  return (
    <header>
      <div>
        <span className="logo">Parfumer Shop</span>
        <ul className="nav">
          <li>Про нас</li>
          <li>Магазин</li>
          <li>Контакти</li>

```



```

    </ul>
    <FaShoppingBag onClick={() => setCartOpen(cartOpen = !cartOpen)}
className={`shop-cart-button ${cartOpen && 'active'}`></FaShoppingBag>

    {cartOpen && (
      <div className="shop-cart">
        {props.orders.length > 0 ? showOrder(props): showNothing()}

      </div>
    )}
  </div>
  <div className="presentation"></div>
</header>

);
}

```

## Item.js

```

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

```

## Order.js

```

import React, { Component } from 'react';
import { FaTrash } from "react-icons/fa";
let zakaz = ''
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='deletItem' onClick={() =>
this.props.onDelete(this.props.item.id)}></FaTrash>
    </div>

    );
}
}

export default order;

```

## ShowFullItems.js

```

import React, { Component } from 'react';

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;

```

**ВІДГУК**

керівника економічного розділу  
на кваліфікаційну роботу бакалавра

на тему:

**«Розробка програмного додатку «розробка веб-сайту для підприємства, що виготовляє парфуми, з використанням фреймворку JavaScript React»**

студента групи 122-19-2 Житник Володимир Ігорович

**Керівник економічного розділу**  
**проф каф. ПЕП та ПУ, д.е.н**

**Вагонова О.Г.**

## ПЕРЕЛІК ФАЙЛІВ НА ОПТИЧНОМУ НОСІЇ

| Ім'я файла                                      | Опис   |
|---|--|
| Пояснювальні документи                          |  |
| Пояснювальна записка кваліфікаційної_роботи.doc | Пояснювальна записка до кваліфікаційної роботи. Документ Word. |
| Пояснювальна записка кваліфікаційної_роботи.doc | Пояснювальна записка до кваліфікаційної роботи в форматі PDF.  |
| При дипломна практика Житник 122-19-2           | При дипломна практика  |
| програма  |  |
| React Js.rar                                    | Архів. Містить коди програми і откомпільовану програму         |
| Презентація                                     |  |
| Презентація.ppt                                 | Презентація кваліфікаційної роботи.                            |