

## Header.jsx

Файл Header.jsx є React-компонентом, який відображає шапку (header) веб-сайту.

Основні складові:

- **Імпорт:** Імпортує необхідні бібліотеки React (React), CSS-модуль (style), компонент для навігації (NavLink), іконку кошика (CiShoppingCart) та іконку ноутбука (FaLaptop).
- **Функціональний компонент Header:**
  - Містить блок logo для відображення логотипу, який складається з іконки ноутбука (FaLaptop) та назви сайту "LapTrade".
  - Містить навігаційне меню (nav) з трьома посиланнями (NavLink): "Home", "Categories" та "About us". Для кожного посилання застосовується умовний клас activeClassName для стилізації активного посилання.
  - Містить блок cart з посиланням (NavLink) на сторінку кошика, відображаючи іконку кошика (CiShoppingCart).

Як працює:

Компонент Header створює верхню панель веб-сайту з логотипом, навігаційним меню та посиланням на кошик. Використання NavLink забезпечує клієнтську навігацію без повного перезавантаження сторінки та автоматично додає клас activeClassName до поточного активного посилання, що дозволяє візуально виділити поточну сторінку в навігації. Стили компонента визначаються у файлі Header.module.css, який імпортується.

## Home.jsx

Цей файл React-компонента Home є головною сторінкою веб-сайту.

Основні складові:

- **Імпорт:** Імпортує необхідні бібліотеки React (useState, useNavigate, NavLink), CSS-модуль (style), іконку пошуку (ImSearch), зображення категорій та компонент Shop.
- **Стан (state):** Використовує useState для керування значенням пошукового інпуту (search).
- **Навігація:** Використовує useNavigate для програмної навігації.

- **Обробник подій:** Функція `handleKeyPress` перенаправляє користувача на сторінку пошуку (`/explore/${search}`) при натисканні клавіші `Enter` в інпуті, якщо поле пошуку не пусте.
- **JSX розмітка:**
  - Відображає банер з текстом та полем пошуку. При натисканні кнопки `"Search"` або клавіші `Enter` відбувається перехід на сторінку `/explore/${search}`. Кнопка `"Search"` деактивована, якщо поле пошуку пусте.
  - Відображає секцію `"Categories"` зі списком категорій. Кожна категорія є посиланням (`NavLink`) на відповідну сторінку категорії. Компонент `CategoryCard` відповідає за відображення кожної картки категорії (зображення та заголовок).
  - Включає компонент `Shop`, який, ймовірно, відображає список товарів.
- **Компонент `CategoryCard`:** Простий функціональний компонент, який приймає `img` та `title` як пропси та відображає зображення та заголовок категорії.

Як працює:

Користувач бачить банер із закликом до дії та полем пошуку. Вводячи текст у поле пошуку та натискаючи `"Search"` або `Enter`, відбувається навігація на сторінку з результатами пошуку. Нижче відображаються картки категорій, клік на які переводить користувача на сторінку з товарами цієї категорії. Також на головній сторінці відображається секція з товарами (`Shop`).

## Shop.jsx

Є `React`-компонентом, який відображає головну секцію магазину на веб-сайті.

Основні складові:

- **Імпорт:** Імпортує необхідні бібліотеки `React` (`React`), `CSS`-модуль (`style`), зображення (продавців, категорій, новинок), дані про ноутбуки (`laptops`), компонент для відображення карток ноутбуків (`CardLaptop`), компонент для навігації (`NavLink`), а також кастомні компоненти `CountdownTimer` та `Label`.
- **Компонент `Shop`:**
  - Має бічну панель (`aside`) з:
    - Компонентом `Label` для заголовків.

- Два компоненти SideCard з інформацією про спеціальні пропозиції та новинки, включаючи зображення, текст та посилання.
- Спеціальний блок з таймером зворотного відліку (CountdownTimer).
- Основну частину (shopMain) з:
  - Компонентом Label для заголовка "Trending laptops".
  - Відображення перших шести елементів з масиву laptops у вигляді карток (CardLaptop).
  - Компонентом Label для заголовка "Laptop Categories".
  - Два компоненти Category, що відображають великі блоки з зображеннями та текстом для категорій "Gaming laptops" та "Work laptops" з посиланнями на відповідні сторінки.
- **Компонент SideCard:** Відображає картку з заголовком, підзаголовком, зображенням, текстом у колі, текстом кнопки та посиланням.
- **Компонент Category:** Відображає великий блок з фоновим зображенням, заголовком, підзаголовком та кнопкою з посиланням.

Як працює:

Компонент Shop організовує відображення контенту магазину. Бічна панель пропонує спеціальні пропозиції та відображає таймер. Основна частина показує список популярних ноутбуків та великі блоки-посилання на основні категорії ноутбуків. Дані про ноутбуки імпортуються з зовнішнього файлу, а навігація між сторінками здійснюється за допомогою компонента NavLink. Кастомні компоненти SideCard, Category, Label та CountdownTimer використовуються для структурування та стилізації контенту.

## Footer.jsx

Файл Footer.jsx є React-компонентом, який відображає нижню частину (footer) веб-сайту.

Основні складові:

- **Імпорт:** Імпортує необхідні бібліотеки React (React), CSS-модуль (style), та іконки соціальних мереж (FaLinkedin, FaXTwitter, FaInstagram).
- **Функціональний компонент Footer:**
  - Має ліву частину (left) з назвою веб-сайту "LapTrade" та іконками соціальних мереж.
  - Має праву частину (right), яка містить кілька колонок (FooterColumn) з посиланнями на різні розділи сайту.

- **Функціональний компонент FooterColumn:** Приймає title (заголовок колонки) та масив links (посилання) як пропси. Відображає заголовок колонки (h5) та список посилань (a), які рендеряться за допомогою методу map.

Як працює:

Компонент Footer створює нижню панель веб-сайту. У лівій частині відображається назва сайту та іконки соціальних мереж для посилань на відповідні профілі. У правій частині розміщено кілька колонок з корисними посиланнями, такими як підтримка, інформація про сайт та контактні дані. Компонент FooterColumn відповідає за структурування кожної окремої колонки з заголовком та списком посилань. Стилi компонента визначаються у файлі Footer.module.css, який імпортується.

## CardLaptop.jsx

Файл CardLaptop.jsx є React-компонентом, який відображає картку окремого ноутбука.

Основні складові:

- **Імпорт:** Імпортує необхідні бібліотеки React (React), іконки кошика (CiShoppingCart) та закриття (IoMdClose), CSS-модуль (style), компонент для навігації (NavLink), компонент для відображення рейтингу зірками (StarRating) та хук для роботи з кошиком (useCart).
- **Функціональний компонент CardLaptop:** Приймає об'єкт laptop як пропс, що містить інформацію про ноутбук.
- **Хук useCart:** Використовує кастомний хук useCart для отримання стану кошика (cart), функції додавання до кошика (addToCart) та функції оновлення кошика (setCart).
- **Стан isInCart:** Логічна змінна, яка визначає, чи присутній поточний ноутбук у кошику.
- **Функція handleCartClick:** Обробник кліку на іконку кошика. Запобігає стандартній поведінці посилання. Якщо ноутбук вже є в кошику, він видаляється; інакше - додається. Після чого оновлюється стан isInCart.
- **JSX розмітка:**
  - Картка є посиланням (NavLink) на сторінку з детальною інформацією про ноутбук (/laptop/\${laptop.name}).
  - Відображає зображення ноутбука.
  - Має кнопку "Open", яка, ймовірно, також веде на сторінку деталей.

- Відображає назву ноутбука.
- Використовує компонент `StarRating` для відображення рейтингу ноутбука.
- У нижній частині відображається ціна ноутбука.
- Залежно від стану `isInCart`, відображається іконка кошика (для додавання) або іконка закриття (для видалення з кошика). При кліку на іконку викликається функція `handleCartClick`.

Як працює:

Компонент `CardLaptop` відображає інформацію про один ноутбук у вигляді інтерактивної картки. Користувач може клікнути на картку, щоб перейти на сторінку з детальною інформацією про товар. Іконка кошика дозволяє додавати або видаляти товар з кошика, використовуючи функціональність, надану хуком `useCart`. Стан наявності товару в кошику візуально відображається зміною іконки. Стили компонента визначаються у файлі `CardLaptop.module.css`, який імпортується.

## Explore.jsx

Файл `Explore.jsx` є React-компонентом, який відображає сторінку з результатами пошуку та фільтрації ноутбуків.

Основні складові:

- **Імпорт:** Імпортує необхідні бібліотеки React (`React`), хуки для навігації (`useNavigate`, `useParams`), CSS-модуль (`style`), функцію фільтрації (`filter`), дані про ноутбуки (`laptops`), компонент "Назад" (`Back`), зображення нових надходжень (`newArrivals`), іконку "Далі" (`GrLinkNext`), компонент картки ноутбука (`CardLaptop`) та компонент заголовка (`Label`).
- **Хуки:**
  - `useParams`: Отримує параметр `query` з URL (пошуковий запит).
  - `useNavigate`: Надає функцію для програмної навігації.
- **Стани:**
  - `search`: Стан для керування значенням пошукового інпуту. Ініціалізується значенням `query` з URL або пустим рядком.
  - `categories`: Масив станів для відфільтрованих категорій.
  - `brands`: Масив станів для відфільтрованих брендів.
  - `maxPrice`: Стан для максимальної ціни фільтрації (за замовчуванням 4000).
- **Функція `filteredLaptops`:** Застосовує функцію `filter` до масиву `laptops`, використовуючи поточний пошуковий запит, вибрані категорії, бренди та максимальну ціну.
- **Обробники подій:**

- `handleSearch`: Здійснює навігацію на сторінку пошуку з поточним значенням `search`.
- `handleCategoryChange`: Додає або видаляє вибрану категорію з масиву `categories`.
- `handleBrandChange`: Додає або видаляє вибраний бренд з масиву `brands`.
- `handlePriceChange`: Оновлює значення `maxPrice` на основі зміни повзунка.
- **JSX розмітка:**
  - Відображає компонент "Назад" (`Back`).
  - Має секцію з фоновим зображенням нових надходжень та заголовком.
  - Містить поле пошуку з кнопкою для повторного пошуку.
  - Має основну частину (`shop`) з бічною панеллю (`aside`) для фільтрів та основною областю (`main`) для відображення результатів.
  - Бічна панель містить:
    - Заголовок `"Filters"` (`Label`).
    - Групи фільтрів для категорій, брендів та максимальної ціни. Кожен фільтр є інтерактивним чекбоксом або повзунком.
  - Основна область містить:
    - Заголовок `"Laptops"` (`Label`).
    - Список відфільтрованих ноутбуків, відображених за допомогою компонента `CardLaptop`.

Як працює:

Компонент `Explore` відображає сторінку для перегляду ноутбуків з можливістю фільтрації за категоріями, брендами та ціною. Початковий пошуковий запит береться з параметра `URL`. Користувач може змінювати пошуковий запит та застосовувати фільтри, що оновлює список відображених ноутбуків у реальному часі завдяки функції `filter`. Стан фільтрів керується за допомогою хуків `useState` та обробників подій. Компонент `CardLaptop` використовується для відображення кожної картки відфільтрованого ноутбука. Стили компонента визначаються у файлі `Explore.module.css`, а логіка фільтрації знаходиться у файлі `filters.js`.

## LaptopPage.jsx

Файл `LaptopPage.jsx` є `React`-компонентом, який відображає детальну інформацію про конкретний ноутбук.

Основні складові:

- **Імпорт:** Імпортує необхідні бібліотеки React (React), CSS-модуль (style), дані про ноутбуки (laptops), хуки для навігації (NavLink, useParams), компонент "Назад" (Back), зображення продавця (seller), компонент для відображення рейтингу зірками (StarRating), хук для роботи з кошиком (useCart), а також іконки характеристик ноутбука (CiShoppingCart, GoCpu, BsGpuCard, RiRam2Line, FaWeightHanging, TbArrowsDiagonalMinimize) та компонент заголовка (Label).
- **Хуки:**
  - useParams: Отримує параметр name з URL, який ідентифікує конкретний ноутбук.
  - useCart: Використовує кастомний хук useCart для отримання стану кошика (cart), функції додавання до кошика (addToCart) та функції оновлення кошика (setCart).
- **Пошук ноутбука:** За допомогою useParams отримує назву ноутбука з URL та знаходить відповідний об'єкт ноутбука в масиві laptops.
- **Функція handleCartButton:** Обробник кліку на кнопку "Add to Card". Перевіряє, чи є ноутбук вже в кошику. Якщо є, видаляє його; якщо немає, додає його до кошика.
- **JSX розмітка:**
  - Відображає компонент "Назад" (Back).
  - Має основний блок content, що містить:
    - Блок imgHolder для відображення зображення ноутбука.
    - Блок trade з інформацією про ноутбук: назва, ціна, посилання на продавця (з його зображенням та назвою), рейтинг (використовуючи StarRating), короткий опис та кнопку "Add to Card" з іконкою кошика, яка при кліку викликає handleCartButton.
  - Відображає заголовок "Details" (Label).
  - Має блок details, що відображає основні характеристики ноутбука (CPU, GPU, RAM, вага, діагональ) з відповідними іконками.

Як працює:

Компонент LaptopPage відображає детальну інформацію про ноутбук, назва якого передається через параметр URL. Він знаходить потрібний об'єкт ноутбука з масиву laptops. Користувач може переглянути зображення, ціну, рейтинг, опис та основні характеристики ноутбука. Кнопка "Add to Card" дозволяє додавати або видаляти товар з кошика, використовуючи функціональність, надану хуком useCart. Також є посилання на сторінку продавця. Стили компонента визначаються у файлі LaptopPage.module.css.

## SellerPage.jsx

Файл SellerPage.jsx є React-компонентом, який відображає сторінку конкретного продавця та список його товарів (ноутбуків).

Основні складові:

- **Імпорт:** Імпортує необхідні бібліотеки React (React), CSS-модуль (style), хук для отримання параметрів з URL (useParams), дані про ноутбуки (laptops), компонент "Назад" (Back), компонент заголовка (Label) та компонент картки ноутбука (CardLaptop), а також зображення продавця (sellerImg).
- **Хук useParams:** Отримує параметр seller з URL, який ідентифікує продавця.
- **Фільтрація товарів продавця:** За допомогою методу filter масиву laptops створює новий масив sellersLaptops, який містить лише ті ноутбуки, у яких поле seller (у нижньому регістрі) відповідає параметру seller з URL (також у нижньому регістрі).
- **JSX розмітка:**
  - Відображає компонент "Назад" (Back).
  - Має блок seller з інформацією про продавця: зображення продавця (sellerImg) та його ім'я (отримане з параметра URL).
  - Відображає заголовок "Laptops" (Label).
  - Має блок list, який відображає список ноутбуків, що належать даному продавцю (sellersLaptops). Кожен ноутбук відображається за допомогою компонента CardLaptop.

Як працює:

Компонент SellerPage відображає інформацію про конкретного продавця та всі ноутбуки, які він продає. Ім'я продавця отримується з URL. Потім відбувається фільтрація масиву всіх ноутбуків для відображення лише тих, які мають відповідне ім'я продавця. Ці відфільтровані ноутбуки відображаються у вигляді карток за допомогою компонента CardLaptop. Стили компонента визначаються у файлі SellerPage.module.css.

## CartPage.jsx

Файл CartPage.jsx є React-компонентом, який відображає сторінку кошика з доданими товарами.

Основні складові:



- **Імпорт:** Імпортує необхідні бібліотеки React (React), кастомний хук для роботи з кошиком (useCart), CSS-модуль (style), компонент "Назад" (Back), компонент заголовка (Label) та компонент картки ноутбука (CardLaptop).
- **Хук useCart:** Використовує кастомний хук useCart для отримання стану кошика (cart).
- **JSX розмітка:**
  - Відображає компонент "Назад" (Back).
  - Має основний блок main, що складається з двох секцій:
    - Секція sum відображає загальну суму товарів у кошику. Заголовок "Total" відображається за допомогою компонента Label. Загальна сума розраховується за допомогою методу reduce для підсумовування цін усіх товарів у масиві cart. Також є кнопка "Buy".
    - Секція cart відображає вміст кошика. Заголовок "Cart" відображається за допомогою компонента Label. Якщо кошик не порожній (cart.length > 0), то кожен товар у кошику відображається за допомогою компонента CardLaptop. Якщо кошик порожній, відображається повідомлення "Your cart is empty".

Як працює:

Компонент CartPage отримує дані про товари в кошику за допомогою кастомного хука useCart. Він відображає загальну вартість товарів у кошику та список самих товарів у вигляді карток, використовуючи компонент CardLaptop. Якщо кошик порожній, користувачеві відображається відповідне повідомлення. Стили компонента визначаються у файлі Cart.module.css.

## App.jsx

Файл App.jsx є головним React-компонентом, який відповідає за налаштування маршрутизації та загальну структуру веб-сайту.

Основні складові:

- **Імпорт:** Імпортує необхідні компоненти та бібліотеки React Router (Routes, Route), компонент шапки (Header), головну сторінку (Home), підвал (Footer), сторінку пошуку/фільтрації (Explore), сторінку детальної інформації про ноутбук (LaptopPage), сторінку всіх категорій (CategoriesPage), сторінку конкретної категорії (CategoryPage), сторінку кошика (CartPage) та сторінку продавця (SellerPage), а також основні стилі (index.css).

- **Функціональний компонент App:**

- Створює загальну структуру веб-сайту, обертаючи контент у контейнер (div з класом container).
- Відображає компонент Header у верхній частині сторінки.
- Використовує компонент Routes для визначення маршрутів веб-сайту.
- Кожен Route пов'язує певний шлях URL з відповідним React-компонентом, який буде відображено при переході на цей шлях:
  - /: Відображає головну сторінку (Home).
  - /explore: Відображає сторінку пошуку/фільтрації (Explore).
  - /explore/:query: Відображає сторінку пошуку/фільтрації (Explore) з урахуванням пошукового запиту, переданого як параметр URL.
  - /laptop/:name: Відображає сторінку детальної інформації про ноутбук (LaptopPage), де name є параметром URL, що ідентифікує ноутбук.
  - /categories: Відображає сторінку зі списком усіх категорій (CategoriesPage).
  - /categories/:category: Відображає сторінку конкретної категорії (CategoryPage), де category є параметром URL, що визначає категорію.
  - /cart: Відображає сторінку кошика (CartPage).
  - /seller/:seller: Відображає сторінку продавця (SellerPage), де seller є параметром URL, що ідентифікує продавця.
- Відображає компонент Footer у нижній частині сторінки.

Як працює:

Компонент App налаштовує навігацію веб-сайту за допомогою React Router. Залежно від URL-адреси, яку відвідує користувач, відображається відповідний компонент. Компоненти Header та Footer є статичними елементами, які відображаються на всіх сторінках завдяки їхньому розташуванню поза межами блоку Routes. Параметри URL (наприклад, :query, :name, :category, :seller) дозволяють передавати динамічні дані до відповідних компонентів для відображення конкретного контенту.

## **CartProvider.jsx**

Файл CartProvider.jsx створює та надає контекст React для керування станом кошика покупок у всьому застосунку.

Основні складові:

- **Імпорт:** Імпортує необхідні функції React (`createContext`, `useContext`, `useState`).
- **Створення контексту `CartContext`:** За допомогою `createContext()` створюється об'єкт контексту, який буде використовуватися для передачі та отримання даних кошика.
- **Компонент `CartProvider`:**
  - Це компонент-провайдер, який обертає частину дерева компонентів, яким потрібен доступ до стану кошика. Він приймає пропс `children`, який представляє обгорнуті компоненти.
  - Використовує хук `useState` для створення стану `cart` (масив товарів у кошику) та функції `setCart` для його оновлення. Початкове значення стану `cart` - порожній масив.
  - Визначає функцію `addToCart`, яка приймає `product` як аргумент і оновлює стан `cart`, додаючи цей товар до попереднього вмісту кошика за допомогою спред-оператора.
  - Повертає компонент `CartContext.Provider`. Властивість `value` цього провайдера є об'єктом, що містить поточний стан кошика (`cart`), функцію для додавання товару (`addToCart`) та функцію для безпосереднього оновлення кошика (`setCart`). Це значення буде доступне всім компонентам-споживачам, обгорнутим цим провайдером.
- **Кастомний хук `useCart`:**
  - Це кастомний хук, який використовує хук `useContext` для отримання значення контексту `CartContext`.
  - Повертає значення контексту, що робить стан кошика та функції для його керування легкодоступними в будь-якому функціональному компоненті, який використовує цей хук.

Як працює:

`CartProvider` створює централізоване місце для зберігання та оновлення стану кошика (`cart`). Завдяки `CartContext.Provider`, стан кошика та пов'язані з ним функції (`addToCart`, `setCart`) стають доступними для будь-якого дочірнього компонента, обгорнутого в `CartProvider`. Компонент `useCart` надає зручний спосіб для функціональних компонентів отримати доступ до цих значень контексту без необхідності використання `useContext(CartContext)` безпосередньо. Це полегшує обмін даними кошика між різними частинами застосунку без необхідності передачі пропсів через багато рівнів компонентів.