

यह एक array है जिसमें सभी laptops का data (जैसे नाम, price, image) होगा।

js

CopyEdit

```
import { useDispatch } from 'react-redux';  
import { addToCart } from '../features/cartSlice';
```

👉 Redux का `useDispatch` hook और `addToCart` action import किया गया है ताकि products को cart में जोड़ा जा सके।

---

## ◆ Component Start

js

CopyEdit

```
const LaptopPage = ({ searchTerm }) => {
```

👉 `LaptopPage` एक functional component है जो `searchTerm` को prop के रूप में लेता है (user के search के लिए)।

js

CopyEdit

```
  const [sortType, setSortType] = useState("");
```

👉 State variable `sortType` यह track करता है कि sorting किस तरह से करनी है – Low to High या High to Low।

js

CopyEdit

```
  const dispatch = useDispatch();
```

👉 `dispatch` function Redux के action को भेजने के लिए इस्तेमाल किया जाता है।

---

## ◆ Search Filter

js

CopyEdit

```
  let filtered = laptopImages.filter(item =>  
    item.title.toLowerCase().includes(searchTerm.toLowerCase())  
  );
```

👉 यह logic search bar में लिखे गए शब्द के आधार पर laptops को filter करता है।

- अगर user ने "hp" लिखा है, तो सिर्फ HP वाले laptops दिखेंगे।

---

## ◆ Sorting Logic

js

CopyEdit

```
if (sortType === "low") {  
  filtered.sort((a, b) => a.price - b.price);  
} else if (sortType === "high") {  
  filtered.sort((a, b) => b.price - a.price);  
}
```

👉 अगर `sortType` "low" है, तो products को ascending order (कम कीमत से ज़्यादा) में sort करेंगे।

👉 अगर `sortType` "high" है, तो descending order (ज़्यादा कीमत से कम) में।

---

## ◆ Cart Handler

js

CopyEdit

```
const handleAddToCart = (product) => {  
  dispatch(addToCart(product));  
};
```

👉 जब user "Add to Cart" बटन दबाता है, तो Redux के ज़रिए उस product को cart में जोड़ दिया जाता है।

---

## ◆ UI Render

js

CopyEdit

```
return (  
  <div className="laptop-page">
```

👉 पूरा layout `laptop-page` नाम की CSS class से styled है।

## ◆ Sorting Buttons

js

CopyEdit

```
<div className="sort-buttons">
  <button onClick={() => setSortType("low")}>Price: Low to
High</button>
  <button onClick={() => setSortType("high")}>Price: High to
Low</button>
</div>
```

👉 दो बटन दिए गए हैं जो sorting type बदलते हैं।

- एक low-to-high करता है
- दूसरा high-to-low

---

### ◆ Product Grid

js

CopyEdit

```
<div className="laptop-product-grid">
```

👉 सभी laptops को grid layout में दिखाया जा रहा है।

js

CopyEdit

```
{filtered.map((item, idx) => (
  <div className="laptop-product-card" key={idx}>
```

👉 Filtered laptops को `.map()` करके एक-एक कार्ड में दिखाया जा रहा है। `key` unique ID के लिए है।

---

### ◆ Product Image & Overlay

js

CopyEdit

```
<div className="laptop-product-image-container">
  <img src={item.img} alt={item.title} />
  <div className="laptop-hover-overlay">
    <h3>{item.title}</h3>
    <p>₹{item.price}</p>
    <button
```

```
        className="laptop-add-to-cart-btn"
        onClick={() => handleAddToCart(item)}
      >
        Add to Cart
      </button>
    </div>
  </div>
```

👉 हर product का image दिखाया गया है, और hover करने पर एक overlay खुलता है जिसमें:

- Title
- Price
- और "Add to Cart" बटन होता है।

---

## ◆ Export

js

CopyEdit

```
export default LaptopPage;
```

👉 Component को export किया गया है ताकि इसे `App.js` या कहीं और इस्तेमाल किया जा सके।

---

## ✅ Summary (Quick Table)

लाइन	क्या करता है
<code>useState</code>	Sorting track करता है
<code>useDispatch</code>	Redux action भेजने के लिए
<code>filter()</code>	Search के हिसाब से items छांटता है
<code>sort()</code>	Price के हिसाब से order करता है
<code>map()</code>	हर item को UI में दिखाता है
<code>dispatch(addToCart)</code>	Cart में item जोड़ता है

---

अगर आप चाहें तो मैं इसका UI कैसा दिखेगा उसका rough sketch भी समझा सकता हूँ, या CSS में कैसे hover effect दिया गया है वो भी बता सकता हूँ। बताएं?

40

**You said:**

```
// src/pages/CartPage.js import React from 'react'; import { useDispatch, useSelector } from
'react-redux'; import { useNavigate } from 'react-router-dom'; // for navigation import {
removeFromCart, increaseQuantity, decreaseQuantity } from '../features/cartSlice'; import
'../style/cartPage.css'; const CartPage = () => { const dispatch = useDispatch(); const
navigate = useNavigate(); const cartItems = useSelector((state) => state.cart.items); const
totalPrice = cartItems.reduce((total, item) => total + (item.price * item.quantity), 0); const
handleBuyNow = () => { navigate('/success'); }; const handleCloseCart = () => { navigate(-1);
// Goes back to the previous page, like closing }; return ( <div className="cart-page"> {/*
Close Button */} <button className="close-cart-btn"
onClick={handleCloseCart}>X</button> <h2>Your Cart</h2> <div className="cart-items">
{cartItems.length === 0 ? ( <p>Your cart is empty!</p> ) : ( cartItems.map((item) => ( <div
key={item.id} className="cart-item"> <img src={item.img} alt={item.title}
className="cart-item-img" /> <div className="cart-item-details"> <h3>{item.title}</h3>
<p>₹{item.price}</p> <div className="quantity-control"> <button onClick={() =>
dispatch(decreaseQuantity(item.id))}>-</button> <span>{item.quantity}</span> <button
onClick={() => dispatch(increaseQuantity(item.id))}>+</button> </div> <p>Total: ₹{item.price
* item.quantity}</p> </div> <button className="remove-btn" onClick={() =>
dispatch(removeFromCart(item.id))}> Remove </button> </div> )) } </div> {cartItems.length
> 0 && ( <div className="cart-total"> <h3>Total Price: ₹{totalPrice}</h3> <button
className="buy-now-btn" onClick={handleBuyNow}> Buy Now </button> </div> )} </div> );
}; export default CartPage;
```

**ChatGPT said:**

बिल्कुल! नीचे आपके दिए गए `CartPage.js` को लाइन-बाय-लाइन हिंदी में समझाया गया है, ताकि आपको पूरा logic और flow अच्छे से क्लियर हो जाए। चलिए शुरू करते हैं:

---

## ◆ Imports

js

CopyEdit

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

👉 React library को import किया गया है ताकि आप JSX और component बना सकें।

js

CopyEdit

```
import { useDispatch, useSelector } from 'react-redux';
```

👉 Redux से:

- `useDispatch` – actions dispatch करने के लिए।
- `useSelector` – Redux state से cart items लेने के लिए।

js

CopyEdit

```
import { useNavigate } from 'react-router-dom';
```

👉 `useNavigate` एक hook है जो pages के बीच navigate करने में काम आता है।

js

CopyEdit

```
import { removeFromCart, increaseQuantity, decreaseQuantity } from  
'../features/cartSlice';
```

👉 ये Redux actions हैं:

- किसी item को हटाने के लिए।
- quantity बढ़ाने/घटाने के लिए।

js

CopyEdit

```
import '../style/cartPage.css';
```

👉 Cart page का styling यहाँ से आएगा।

---

## ◆ Component Start

js

CopyEdit

```
const CartPage = () => {
```

👉 CartPage एक functional React component है।

js

CopyEdit

```
const dispatch = useDispatch();  
const navigate = useNavigate();
```

👉 Redux और React Router के functions usable बनाए गए।

js

CopyEdit

```
const cartItems = useSelector((state) => state.cart.items);
```

👉 Redux store से cart items को access किया गया है।

---

## ◆ Total Price Calculation

js

CopyEdit

```
const totalPrice = cartItems.reduce((total, item) => total +  
(item.price * item.quantity), 0);
```

👉 सभी items के price और quantity से total bill निकाला गया है।

---

## ◆ Navigation Handlers

js

CopyEdit

```
const handleBuyNow = () => {  
  navigate('/success');  
};
```

👉 जब user "Buy Now" दबाता है तो success page पर भेजा जाता है।

js

CopyEdit

```
const handleCloseCart = () => {  
  navigate(-1); // Back to previous page  
};
```

👉 Cart को बंद करने पर पिछली page पर वापस ले जाता है।

---

## ◆ Return JSX

js

CopyEdit

```
return (  
  <div className="cart-page">
```

👉 Cart Page का main container है।

### ◆ Close Button

js

CopyEdit

```
    <button className="close-cart-btn"  
    onClick={handleCloseCart}>X</button>
```

👉 "X" बटन जिसे दबाकर user वापस जा सकता है।

### ◆ Heading

js

CopyEdit

```
    <h2>Your Cart</h2>
```

👉 Cart heading।

---

## ◆ Cart Items Display

js

CopyEdit

```
    <div className="cart-items">  
      {cartItems.length === 0 ? (  
        <p>Your cart is empty!</p>
```

👉 अगर cart खाली है, तो message दिखेगा।

js

CopyEdit

```
      ) : (  
        cartItems.map((item) => (  
          <div key={item.id} className="cart-item">
```

👉 Cart में item हो तो `.map()` करके display करेंगे।

---



## ◆ Each Cart Item

js

CopyEdit

```
      <img src={item.img} alt={item.title}
className="cart-item-img" />
```

👉 Product की image।

js

CopyEdit

```
      <div className="cart-item-details">
        <h3>{item.title}</h3>
        <p>₹{item.price}</p>
```

👉 Title और Price दिखाया गया है।

## ◆ Quantity Controls

js

CopyEdit

```
      <div className="quantity-control">
        <button onClick={() =>
dispatch(decreaseQuantity(item.id))}>-</button>
        <span>{item.quantity}</span>
        <button onClick={() =>
dispatch(increaseQuantity(item.id))}>+</button>
      </div>
```

👉 Quantity को घटाने और बढ़ाने के buttons।

js

CopyEdit

```
      <p>Total: ₹{item.price * item.quantity}</p>
    </div>
```

👉 उस item का total price।

js

CopyEdit

```
      <button className="remove-btn" onClick={() =>
dispatch(removeFromCart(item.id))}>
        Remove
      </button>
```

👉 Item हटाने का button।

---

### ◆ Total Price + Buy Button

js

CopyEdit

```
{cartItems.length > 0 && (  
  <div className="cart-total">  
    <h3>Total Price: ₹{totalPrice}</h3>
```

👉 अगर cart में items हैं, तो total price और "Buy Now" दिखाया जाएगा।

js

CopyEdit

```
    <button className="buy-now-btn" onClick={handleBuyNow}>  
      Buy Now  
    </button>  
  </div>  
  )}
```

### ◆ Export

js

CopyEdit

```
export default CartPage;
```

👉 इस component को दूसरे files में use करने के लिए export किया गया।

---

### ✅ Summary Table

भाग	काम
<code>useSelector</code>	Cart से items लाता है
<code>useDispatch</code>	Redux actions dispatch करता है
<code>navigate()</code>	Routing करता है (success या वापस)
<code>reduce()</code>	Total bill निकालता है

`map()` हर item को display करता है

`dispatch(add/remove/increase)` Redux को update करता है