**Step 1: Setup**

First, create a new React application using create-react-app:

bash

Copy code

npx create-react-app t-shirt-store

cd t-shirt-store

**Step 2: Structure the Application**

Create the following file structure inside the src directory:

css

Copy code

src/

components/

Cart.js

ProductCard.js

ProductList.js

pages/

CartPage.js

HomePage.js

App.js

index.js

data.js

**Step 3: Create Components**

**data.js**

Fetch and store the catalogue data.

js

Copy code

export const fetchCatalogue = async () => {

const response = await fetch('https://geektrust.s3.ap-southeast-1.amazonaws.com/coding-problems/shopping-cart/catalogue.json');

const data = await response.json();

return data;

};

**ProductCard.js**

jsx

Copy code

import React from 'react';

const ProductCard = ({ product, addToCart }) => (

<div className="product-card">

<img src={product.image} alt={product.name} />

<h3>{product.name}</h3>

<p>{product.price}</p>

<button onClick={() => addToCart(product)}>Add to Cart</button>

</div>

);

export default ProductCard;

**ProductList.js**

jsx

Copy code

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

import ProductCard from './ProductCard';

const ProductList = ({ products, addToCart }) => (

<div className="product-list">

{products.map(product => (

<ProductCard key={product.id} product={product} addToCart={addToCart} />

))}

</div>

);

export default ProductList;

**Cart.js**

jsx

Copy code

import React from 'react';

const Cart = ({ cart, increaseQuantity, decreaseQuantity, removeItem }) => (

<div className="cart">

{cart.map(item => (

<div key={item.product.id} className="cart-item">

<img src={item.product.image} alt={item.product.name} />

<div>

<h4>{item.product.name}</h4>

<p>{item.product.price}</p>

<div>

<button onClick={() => decreaseQuantity(item.product)}>-</button>

<span>{item.quantity}</span>

<button onClick={() => increaseQuantity(item.product)}>+</button>

</div>

<button onClick={() => removeItem(item.product)}>Remove</button>

</div>

</div>

))}

</div>

);

export default Cart;

**Step 4: Create Pages**

**HomePage.js**

jsx

Copy code

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

import { fetchCatalogue } from '../data';

import ProductList from '../components/ProductList';

const HomePage = ({ addToCart }) => {

const [products, setProducts] = useState([]);

const [searchQuery, setSearchQuery] = useState('');

const [filters, setFilters] = useState({ gender: '', color: '', priceRange: '', type: '' });

useEffect(() => {

const loadProducts = async () => {

const data = await fetchCatalogue();

setProducts(data);

};

loadProducts();

}, []);

const handleSearch = () => {

// Implement search logic here

};

const handleFilterChange = (e) => {

setFilters({ ...filters, [e.target.name]: e.target.value });

};

return (

<div>

<input

type="text"

placeholder="Search"

value={searchQuery}

onChange={(e) => setSearchQuery(e.target.value)}

/>

<button className="search-button-container" onClick={handleSearch}>Search</button>

<div>

<label>Gender</label>

<select name="gender" onChange={handleFilterChange}>

<option value="">All</option>

<option value="male">Male</option>

<option value="female">Female</option>

</select>

<label>Colour</label>

<select name="color" onChange={handleFilterChange}>

<option value="">All</option>

<option value="red">Red</option>

<option value="blue">Blue</option>

</select>

<label>Price Range</label>

<select name="priceRange" onChange={handleFilterChange}>

<option value="">All</option>

<option value="0-50">$0 - $50</option>

<option value="50-100">$50 - $100</option>

</select>

<label>Type</label>

<select name="type" onChange={handleFilterChange}>

<option value="">All</option>

<option value="polo">Polo</option>

<option value="round">Round</option>

</select>

</div>

<ProductList products={products} addToCart={addToCart} />

</div>

);

};

export default HomePage;

**CartPage.js**

jsx

Copy code

import React from 'react';

import Cart from '../components/Cart';

const CartPage = ({ cart, increaseQuantity, decreaseQuantity, removeItem }) => {

const totalAmount = cart.reduce((sum, item) => sum + item.product.price \* item.quantity, 0);

return (

<div>

<h2>Shopping Cart</h2>

<Cart

cart={cart}

increaseQuantity={increaseQuantity}

decreaseQuantity={decreaseQuantity}

removeItem={removeItem}

/>

<h3>Total Amount: ${totalAmount.toFixed(2)}</h3>

</div>

);

};

export default CartPage;

**Step 5: Setup Routing and State Management**

**App.js**

jsx

Copy code

import React, { useState } from 'react';

import { BrowserRouter as Router, Route, Link, Switch } from 'react-router-dom';

import HomePage from './pages/HomePage';

import CartPage from './pages/CartPage';

const App = () => {

const [cart, setCart] = useState([]);

const addToCart = (product) => {

const existingItem = cart.find(item => item.product.id === product.id);

if (existingItem) {

if (existingItem.quantity < product.quantity) {

setCart(cart.map(item =>

item.product.id === product.id

? { ...item, quantity: item.quantity + 1 }

: item

));

} else {

alert('Cannot add more than available quantity');

}

} else {

setCart([...cart, { product, quantity: 1 }]);

}

};

const increaseQuantity = (product) => {

setCart(cart.map(item =>

item.product.id === product.id

? { ...item, quantity: item.quantity + 1 }

: item

));

};

const decreaseQuantity = (product) => {

const existingItem = cart.find(item => item.product.id === product.id);

if (existingItem.quantity === 1) {

removeItem(product);

} else {

setCart(cart.map(item =>

item.product.id === product.id

? { ...item, quantity: item.quantity - 1 }

: item

));

}

};

const removeItem = (product) => {

setCart(cart.filter(item => item.product.id !== product.id));

};

return (

<Router>

<div>

<nav>

<Link to="/">Home</Link>

<Link to="/cart">Cart</Link>

</nav>

<Switch>

<Route exact path="/" component={() => <HomePage addToCart={addToCart} />} />

<Route path="/cart" component={() => <CartPage cart={cart} increaseQuantity={increaseQuantity} decreaseQuantity={decreaseQuantity} removeItem={removeItem} />} />

</Switch>

</div>

</Router>

);

};

export default App;

**Step 6: Basic Styles**

Create a styles.css file to add some basic styling.

css

Copy code

/\* styles.css \*/

.product-card, .cart-item {

border: 1px solid #ccc;

padding: 16px;

margin: 16px;

text-align: center;

}

.product-card img, .cart-item img {

width: 100px;

height: 100px;

}

.product-list, .cart {

display: flex;

flex-wrap: wrap;

justify-content: space-around;

}

nav {

display: flex;

justify-content: space-between;

padding: 16px;

background-color: #eee;

}

nav a {

margin: 0 8px;

text-decoration: none;

color: #333;

}

nav a:hover {

text-decoration: underline;

}

**Step 7: Run the Application**

Finally, run your application to test it:

bash

Copy code

npm start

You should now have a basic e-commerce UI that allows browsing, searching, filtering, adding items to the cart, and viewing the cart. Adjust the search and filter logic as needed to match your exact requirements.

4o

Top of Form

Bottom of Form