Experiment-7

ProductCard Component Using Props

Code Implementation

```
import React from 'react';

// The reusable ProductCard component
// It accepts props: name, price, and stockStatus
function ProductCard({ name, price, stockStatus }) {
  // Determine text color for stock status
  const statusColor = stockStatus === 'In Stock' ? 'text-green-500' : 'text-red-500';

  return (
    <div className="bg-gray-800 shadow-lg rounded-xl p-6 m-4 w-64 text-center border border-gray-700 hover:shadow-2xl hover:shadow-indigo-500/20 transition-shadow duration-300">
      <h2 className="text-xl font-bold mb-3 text-white">{name}</h2>
      <p className="text-gray-300 mb-2">Price: ${price.toFixed(2)}</p>
      <p className={`font-semibold ${statusColor}`}>Status: {stockStatus}</p>
    </div>
  );
}

// The main App component that uses the ProductCard
export default function App() {
  // An array of product data to demonstrate reusability
  const products = [
    { name: 'Wireless Mouse', price: 25.99, stockStatus: 'In Stock' },
    { name: 'Keyboard', price: 45.50, stockStatus: 'Out of Stock' },
    { name: 'Monitor', price: 199.99, stockStatus: 'In Stock' },
  ];

  return (
    <div className="bg-gray-900 min-h-screen flex flex-col items-center justify-center font-sans p-4">
      <div className="bg-gray-800 border-2 border-gray-700 rounded-2xl p-8 shadow-md">
        <h1 className="text-4xl font-bold text-center mb-8 text-white">Products List</h1>
        <div className="flex flex-wrap justify-center">
          {/* Map over the products array and render a ProductCard for each item */}
```

```
     {products.map((product, index) => (
      <ProductCard
       key={index} // Using index as a key for this simple example
       name={product.name}
       price={product.price}
       stockStatus={product.stockStatus}
      />
     ))}
    </div>
   </div>
  </div>
 );
}
```