Refactor the previous project to use Hooks and ContextAPI for state management.

Let's refactor the previous project to use Hooks and Context API for state management. We'll manage the cart state globally using Context API and use Hooks for component state management.

Step 1: Setup the Project (If not already done)

1. **Install Node.js and npm:**
   - Download and install Node.js from here.
   - npm (Node Package Manager) comes bundled with Node.js.
2. **Create a New React Project:**
   - Open your terminal/command prompt.
   - Run the following command to create a new React app (replace `ecommerce-site` with your desired project name):

```
npx create-react-app ecommerce-site
```

3. **Navigate to the Project Directory:**

```
cd ecommerce-site
```

4. **Install React Router:**

```
npm install react-router-dom
```

5. **Start the Development Server:**

```
npm start
```

Step 2: Setup Context API for State Management

1. **Create Context for Cart State:**
   - Inside the `src` folder, create a new folder called `context`.
   - Inside the `context` folder, create a file called `CartContext.js`.
2. **Edit `CartContext.js` to Setup Context and Provider:**

```
import React, { createContext, useState } from 'react';

export const CartContext = createContext();

export const CartProvider = ({ children }) => {
  const [cart, setCart] = useState([]);

  const addToCart = (product) => {
    setCart([...cart, product]);
  };

  const removeFromCart = (productId) => {
    setCart(cart.filter(product => product.id !== productId));
  };

  return (
    <CartContext.Provider value={{ cart, addToCart, removeFromCart
}}>
      {children}
    </CartContext.Provider>
  );
};
```

3. **Wrap the App with CartProvider in `index.js`:**

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import { CartProvider } from './context/CartContext';

ReactDOM.render(
  <CartProvider>
    <App />
  </CartProvider>,
  document.getElementById('root')
);
```

Step 3: Refactor Components to Use Context API

1. **Edit `Products.js` to Use Cart Context:**

```
import React, { useContext } from 'react';
import { Link } from 'react-router-dom';
import { CartContext } from '../context/CartContext';

const products = [
  { id: 1, name: 'Product 1' },
  { id: 2, name: 'Product 2' },
  { id: 3, name: 'Product 3' },
];

function Products() {
  const { addToCart } = useContext(CartContext);

  return (
    <div>
      <h1>Products</h1>
      <ul>
        {products.map((product) => (
          <li key={product.id}>
            <Link
to={`/products/${product.id}`}>{product.name}</Link>
            <button onClick={() => addToCart(product)}>Add to
Cart</button>
          </li>
        ))}
      </ul>
    </div>
  );
}

export default Products;
```

2. **Edit `ProductDetails.js` to Use Cart Context:**

```
import React, { useContext } from 'react';
import { useParams } from 'react-router-dom';
import { CartContext } from '../context/CartContext';

const products = [
  { id: 1, name: 'Product 1', description: 'Description for product
1' },
  { id: 2, name: 'Product 2', description: 'Description for product
2' },
  { id: 3, name: 'Product 3', description: 'Description for product
3' },
];

function ProductDetails() {
  const { productId } = useParams();
  const { addToCart } = useContext(CartContext);
  const product = products.find((p) => p.id === parseInt(productId));

  if (!product) {
    return <div>Product not found</div>;
  }

  return (
    <div>
      <h1>{product.name}</h1>
      <p>{product.description}</p>
      <button onClick={() => addToCart(product)}>Add to Cart</button>
    </div>
  );
}

export default ProductDetails;
```

3. **Edit `Cart.js` to Use Cart Context:**

```
import React, { useContext } from 'react';
import { CartContext } from '../context/CartContext';

function Cart() {
  const { cart, removeFromCart } = useContext(CartContext);

  return (
    <div>
      <h1>Cart</h1>
      {cart.length === 0 ? (
        <p>Your cart is empty.</p>
      ) : (
        <ul>
          {cart.map((product) => (
            <li key={product.id}>
              {product.name}
              <button onClick={() =>
removeFromCart(product.id)}>Remove</button>
            </li>
          ))}
        </ul>
      )}
    </div>
  );
}

export default Cart;
```

Step 4: Setup Routes with React Router

1. **Edit `App.js` to Setup Routes:**

```
import React from 'react';
import { BrowserRouter as Router, Route, Routes, Link } from 'react-
router-dom';
import Home from './pages/Home';
import Products from './pages/Products';
import ProductDetails from './pages/ProductDetails';
import Cart from './pages/Cart';
import './App.css';

function App() {
  return (
    <Router>
      <div className="App">
        <nav>
          <Link to="/">Home</Link>
          <Link to="/products">Products</Link>
          <Link to="/cart">Cart</Link>
        </nav>
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/products" element={<Products />} />
          <Route path="/products/:productId" element={<ProductDetails
/>} />
          <Route path="/cart" element={<Cart />} />
        </Routes>
      </div>
    </Router>
  );
}

export default App;
```

Step 5: Add Basic Pages (If not already done)

1. **Edit `Home.js` to Add the Home Page Component:**

```
import React from 'react';
import { Link } from 'react-router-dom';

function Home() {
  return (
    <div>
      <h1>Welcome to Our E-commerce Site</h1>
      <Link to="/products">View Products</Link>
    </div>
  );
}

export default Home;
```

2. **Add Basic Styling (Optional):**
   - Open `src/App.css` and add some basic styling:

```
.App {
  text-align: center;
}

nav {
  display: flex;
  justify-content: center;
  gap: 20px;
  margin: 20px 0;
}

a {
  text-decoration: none;
  color: #61dafb;
}

a:hover {
  text-decoration: underline;
}

ul {
  list-style-type: none;
  padding: 0;
}

li {
  margin: 10px 0;
}

button {
  margin-left: 10px;
}
```

Step 6: Run Your App

1. **Start the Development Server Again (if it's not running):**

```
npm start
```

2. **Open Your Browser:**
   - Navigate to `http://localhost:3000/` to see your E-commerce site in action with state management using Hooks and Context API.

By following these steps, you should have a multi-page E-commerce site with state management using React Hooks and Context API.