

## React Assessment: E-Commerce Product Dashboard

### Objective:

Build a basic e-commerce dashboard where users can:

1. View a list of products (with React Router for navigation).
2. Add and remove products from a shopping cart (using React Context and Reducers).
3. Save cart information in localStorage.
4. Manage user login/logout using cookies (React Cookies).
5. Fetch product data from an API (using Axios).
6. Show notifications using React Toastify.

### Requirements:

#### 1. Project Setup

Create a React app using `create-react-app` or Vite. Install the necessary packages:

```
npm install react-router-dom axios react-cookie react-toastify
```

#### 2. Functionalities:

##### 1. Product List Page (/products):

- Fetch a list of products from a fake API (e.g., <https://fakestoreapi.com/products>).
- Display the products in a grid with details:
  - Product name
  - Price
  - "Add to Cart" button
- If a product is added to the cart, show a success notification using React Toastify.

##### 2. Shopping Cart Page (/cart):

- Display all products added to the cart (use React Context to manage cart state).
- Implement a "Remove from Cart" button for each product.
- Display the total price of items in the cart.

##### 3. User Login (with Cookies):

- Create a basic login form on /login with the following fields:
  - Username
  - Password
- On login, save the username in cookies.
- If the user is logged in, show their username at the top of every page.
- Add a "Logout" button that clears the cookie and redirects to /login.

##### 4. Persist Cart Data (Local Storage):

- Save the cart data in localStorage so it persists after a page refresh.
- Load cart data from localStorage when the app starts.

### 5. React Router:

- Set up routes for:
  - /products (Product List)
  - /cart (Shopping Cart)
  - /login (User Login)

### 6. React Toastify:

- Show a toast notification for:
  - Adding a product to the cart.
  - Removing a product from the cart.
  - Successful login/logout.

### 3. Folder Structure (Example):

src

```
├── components
│   ├── Header.js
│   ├── ProductCard.js
│   └── CartItem.js
├── pages
│   ├── Login.js
│   ├── Products.js
│   └── Cart.js
├── context
│   ├── CartContext.js
│   └── CartReducer.js
├── App.js
├── index.js
├── utils
│   └── api.js
```

### 4. Key Guidelines:

- Use React Context and Reducers for managing the cart state.
- Use Axios for fetching products from the API.
- Use localStorage to persist the cart state across page reloads.
- Use React Cookies to handle user login/logout state.
- Use React Toastify for notifications.

### 5. Grading Criteria:

1. Functional and responsive UI.
2. Proper usage of React Router for navigation.
3. Effective state management using React Context and Reducers.
4. Correct integration of Axios for API calls.
5. Persistent state using localStorage and cookies.
6. Clean, readable code with comments.

## 6. Stretch Goals (Optional):

- Add a "Search" feature for products.
- Include a product details page (`/products/:id`) for more information about a selected product.
- Implement dark mode using React Context.

 **Wishing You All the Best! Code With Confidence, Learn With Curiosity, and Create With Passion. Every Step You Take is Progress. Go Shine! **