**Data Flow Diagram (DFD) for E-Commerce App**

**E-Commerce App** consists of a **frontend (React.js with React Router)** and a **backend (Express.js with MongoDB)**. It includes authentication (login/signup), a product catalog, a shopping cart, and a payment system. Below is the **DFD representation**:

**Level 0 (Context Diagram)**

At the highest level, the system consists of:

1. **External Entity: User** → Interacts with the app by browsing products, adding items to the cart, logging in, and making payments.
2. **Process: E-Commerce System** → Manages user authentication, cart operations, checkout, and payment processing.
3. **Data Store: MongoDB & React State (useState)** → Stores user accounts, cart items, and product details.

+--------------------+

| External Entity: |

| User |

+--------------------+

|

v

+--------------------+

| E-Commerce App | (Process 1.0)

+--------------------+

|

v

+--------------------+

| MongoDB Database | (Stores Users, Products, Cart Items)

+--------------------+

**Level 1 DFD (Decomposed Processes)**

Breaking down **Process 1.0 (E-Commerce System)** into detailed steps:

+--------------------+ +-------------------------+

| External Entity: | | Process 1.1 - User |

| User | -----> | Authentication/Login |

+--------------------+ +-------------------------+

| |

v v

+--------------------+ +-------------------------+

| Process 1.2 - | | Process 1.3 - Browse |

| View Products | -----> | & Search Products |

+--------------------+ +-------------------------+

| |

v v

+--------------------+ +-------------------------+

| Process 1.4 - | | Process 1.5 - Cart |

| Add to Cart | -----> | (Add/Remove Items) |

+--------------------+ +-------------------------+

| |

v v

+--------------------+ +-------------------------+

| Process 1.6 - | | Process 1.7 - Checkout |

| Store Cart Data | -----> | & Payment Processing |

+--------------------+ +-------------------------+

|

v

+--------------------+

| Process 1.8 - |

| Fetch Products |

| from MongoDB API |

+--------------------+

**Explanation of Level 1 DFD Processes**

1. **Process 1.1 - User Authentication (Login/Signup)**
   * The user enters **email and password** on the **Login page**.
   * The app sends a **request to the backend (/api/login)**.
   * The backend **verifies credentials** using **bcrypt** and returns a **JWT token** if valid.
2. **Process 1.2 - View Products (Product Catalogue)**
   * The app fetches **product data from MongoDB** and displays it dynamically in the **ProductCatalogue component**.
3. **Process 1.3 - Browse & Search Products**
   * Users can navigate between products and filter/search items.
4. **Process 1.4 - Add to Cart**
   * The user selects a product and clicks **"Add to Cart"**.
   * The selected product is **stored in React state (useState)**.
5. **Process 1.5 - Manage Cart (Add/Remove Items)**
   * Users can view their cart and remove items before checkout.
   * Cart updates are stored in **React state**.
6. **Process 1.6 - Store Cart Data**
   * The cart data is **stored in local storage or state** to persist until checkout.
7. **Process 1.7 - Checkout & Payment Processing**
   * The user enters **payment details** on the **Payment page**.
   * The app processes the payment (this could integrate with an external payment API like Stripe).
8. **Process 1.8 - Fetch Product Data (MongoDB API Call)**
   * The **React frontend** makes an API request to the backend to fetch the **product list**.
   * The **Express.js backend** retrieves data from MongoDB and sends it back to the frontend.

**Data Flow Summary**

✔ **User navigates → System loads products dynamically → User logs in/signs up → Products are displayed → User adds to cart → Proceeds to payment**  
✔ Uses **React Router for navigation, MongoDB for data storage, Express.js for backend API, and JWT for authentication**.  
✔ **Secure authentication with bcrypt and JWT tokens**.  
✔ **React State (useState) and localStorage** are used for cart management.