**Data Flow Diagram (DFD) for Login Page App**

**Login Page App** allows users to enter their email and password, validates their credentials, and either logs them in or shows an error message. Below is the **DFD representation**:

**Level 0 (Context Diagram)**

At the highest level, the system consists of:

1. **External Entity: User** → Enters credentials (email and password).
2. **Process: Login System** → Validates credentials and logs the user in or shows an error.
3. **Data Store: React State (useState Hook)** → Stores user input and login status.

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

| External Entity: |

| User |

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

|

v

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

| Login System | (Process 1.0)

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

|

v

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

| useState Hook | (Stores email, password, login status)

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

**Level 1 DFD (Decomposed Processes)**

Breaking down **Process 1.0 (Login System)** into detailed steps:

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

| External Entity: | | Process 1.1 - Click |

| User | -----> | (Nav Link Selection) |

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

| |

v v

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

| Process 1.2 - | | Process 1.3 - Render |

| Route Matching | <-----> | (Display Page) |

| (React Router) | | (Home/Menu/Contact) |

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

|

v

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

| React State | (Manages Active Route)

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

**Explanation of Level 1 DFD Processes**

1. **Process 1.1 - User Clicks Navigation Link**
   * The user selects a link (Home, Menu, or Contact).
   * The browser updates the URL accordingly.
2. **Process 1.2 - Route Matching (React Router)**
   * The system checks which route (/, /menu, /contact) was selected.
   * It finds and matches the corresponding React Component.
3. **Process 1.3 - Page Rendering**
   * The matched component (Home.js, Menu.js, or Contact.js) is displayed dynamically.
   * React updates the UI based on the selected route.
4. **React State (Browser Router)**
   * Keeps track of which page is active and ensures smooth navigation without refreshing the page.

**Data Flow Summary**

✔ **User clicks a link → React Router processes the route → Loads and displays the corresponding page**  
✔ Uses **React Router (<Routes> and <Route> components)** to handle navigation.  
✔ No backend or database—only front-end state management with React.