**Level 0 (Context Diagram)**

At the highest level, the system consists of a single process that represents the Login Form App.

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

| External Entity |

| (User) |

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

|

v

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

| Login Form App |

| (Process: 1.0) |

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

|

v

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

| Data Store: Users |

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

**Explanation:**

* **External Entity (User):** The user interacts with the **Login Form App** by entering their email and password.
* **Process (Login Form App):** The system processes the login credentials and validates them.
* **Data Store (Users):** Stores the predefined valid user credentials.

**Level 1 DFD (Decomposition of Process 1.0 - Login Form App)**

Now, let’s break down the **Login Form App** process into more detailed steps.

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

| External Entity |

| (User) |

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

|

v

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

| Process: 1.1 - Enter Credentials |

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

|

v

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

| Data Store: Input State |

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

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

| Process: 1.2 - Validate Credentials |

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

|

v

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

| Decision: Credentials Valid? |

| (Yes) | (No) |

|-------------|---------------|

| Redirect to | Show Error |

| Dashboard | Message |

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

**Explanation:**

1. **Process 1.1 (Enter Credentials):**
   * The user inputs their **email** and **password** into the form.
   * The **useState hook** updates the state dynamically.
2. **Process 1.2 (Validate Credentials):**
   * The system checks if both fields are filled; otherwise, it displays an error message.
   * The credentials are compared with stored values (user@example.com and password).
3. **Decision (Credentials Valid?):**
   * **If valid**, the system clears any error messages and **alerts "Login successful!"**.
   * **If invalid**, an error message **"Invalid credentials."** is displayed.

**Data Flow**

* The **User** enters credentials in the form.
* The **React State** temporarily holds the input values.
* The **System** validates credentials:
  + If valid → **Success message (or redirect)**
  + If invalid → **Error message displayed**

**Additional Notes:**

* This **React-based form** handles input state dynamically.
* **Error handling ensures users provide input before submission.**
* Future enhancements could include:
  + **API authentication**
  + **User session storage**
  + **Enhanced UI with form validation messages**