**Level 0 (Context Diagram)**

At the highest level, the **React App** loads and displays a simple UI with an image, text, and a link.

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

| External Entity |

| (User) |

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

|

v

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

| React App |

| (Process: 1.0) |

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

|

v

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

| Data Store: App.js |

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

**Explanation:**

* **External Entity (User):** The user interacts with the React app by viewing the content.
* **Process (React App):** The system loads the app components and displays them.
* **Data Store (App.js):** Stores the React component structure and logic.

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

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

| External Entity |

| (User) |

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

|

v

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

| Process: 1.1 - Load App |

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

|

v

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

| Process: 1.2 - Display Content |

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

|

v

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

| Data Store: App.js |

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

**Explanation:**

1. **Process 1.1 (Load App):**
   * The browser loads the App.js component.
2. **Process 1.2 (Display Content):**
   * The UI is rendered, showing:
     + A **logo image** (logo.svg).
     + A **text message** (Edit src/App.js and save to reload.).
     + A **link** (Learn React button) that directs users to the React website.
3. **Data Store (App.js):**
   * Stores the React component logic and JSX structure.

**Data Flow**

* **User opens the app** → React loads the **App component**.
* The **logo** and **text** are displayed from App.js.
* The **external link** (ReactJS.org) allows further learning.

**Additional Notes:**

* Uses **React Functional Components**.
* **CSS Styling (App.css)** is applied to format the page.
* Future Enhancements:
  + **Dynamic Content** (e.g., fetching API data).
  + **User Interactions** (e.g., buttons, input forms).
  + **State Management** (e.g., useState, Redux).