**Data Flow Diagram (DFD) for Simple To-Do List App**

**Simple To-Do List App** allows users to add tasks, which are stored in state and displayed as a list. Below is the **DFD representation**:

**Level 0 (Context Diagram)**

At the highest level, the system consists of:

1. **External Entity: User** → Inputs a task and adds it to the list.
2. **Process: To-Do List System** → Manages task input and updates the list dynamically.
3. **Data Store: React State (useState Hook)** → Stores tasks and updates the UI.

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

| External Entity: |

| User |

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

|

v

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

| To-Do List System | (Process 1.0)

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

|

v

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

| useState Hook | (Stores task list)

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

**Level 1 DFD (Decomposed Processes)**

Breaking down **Process 1.0 (To-Do List System)** into detailed steps:

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

| External Entity: | | Process 1.1 - Input |

| User | -----> | (Enter Task) |

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

| |

v v

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

| Process 1.2 - | | Process 1.3 - Add Task|

| Capture Input | -----> | (Update useState) |

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

| |

v v

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

| Process 1.4 - | | Process 1.5 - Display |

| Store Task | -----> | (Show Task List) |

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

**Explanation of Level 1 DFD Processes**

1. **Process 1.1 - User Inputs Task**
   * The user types a new task in the input field.
2. **Process 1.2 - Capture Input**
   * The system listens for input changes (onChange event).
3. **Process 1.3 - Add Task (useState)**
   * The user clicks "Add Task," which updates the task list in state.
4. **Process 1.4 - Store Task**
   * The new task is stored in the tasks array using React state.
5. **Process 1.5 - Display Task List**
   * The updated list of tasks is displayed dynamically in the UI.

**Data Flow Summary**

✔ **User enters a task → System captures input → Updates state → Displays task list**  
✔ Uses **React state (useState Hook)** for real-time updates.  
✔ **No backend or database**—only front-end state management.