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

**Project Description**

The Simple To-Do List is a React-based web application that allows users to add tasks dynamically. Users can input tasks, view them in a list format, and manage their daily activities efficiently. The system processes user input, updates the task list in state, and renders UI changes dynamically.

**Level 0 DFD (Context Diagram)**

The Level 0 DFD provides a high-level overview of the system, illustrating how users interact with the To-Do List App and how data flows between various components.

+-----------------------+  
| External Entities |  
| |  
| - User |  
+-----------------------+  
 |  
 v  
+---------------------------+  
| To-Do List App |  
| (Process: 1.0) |  
+---------------------------+  
 |  
 v  
+---------------------------+  
| Data Store: Application |  
| State (Task List) |  
+---------------------------+

**Explanation:**

**• External Entity (User):**The user interacts with the To-Do List App by adding tasks.

•**Process (To-Do List App):** The system processes user input and updates the task list.

• **Data Store (Application State - Task List):** Stores and manages the list of tasks.

**Level 1 DFD (Detailed Breakdown)**

The Level 1 DFD further decomposes the To-Do List App process into sub-processes, showing how user interactions trigger different functionalities.

+-----------------------+  
| External Entity |  
| (User) |  
+-----------------------+  
 |  
 v  
+----------------------------+  
| Process: 1.1 - Add Task |  
| (Input & Click Add) |  
+----------------------------+  
 |  
 v  
+----------------------------+  
| Process: 1.2 - Update List |  
| (Modify Application State)|  
+----------------------------+  
 |  
 v  
+----------------------------+  
| Process: 1.3 - Display List|  
| (Render Task List in UI) |  
+----------------------------+  
 |  
 v  
+----------------------------+  
| Data Store: Application |  
| State (Task List) |  
+----------------------------+

**Explanation:**

1**. Process 1.1 (Add Task - Input & Click Add):** The user enters a task and clicks the 'Add Task' button.

2. **Process 1.2 (Update List - Modify Application State):** The new task is stored in the application state and displayed in the UI.

3**. Process 1.3 (Display List - Render Task List in UI):** The UI updates dynamically, showing the task list.

**Data Flow**

• The User enters a task and clicks 'Add Task'.

• The System updates the task list in the application state.

• The UI dynamically updates, displaying the new task.

**Additional Notes**

• This system currently operates as a simple React To-Do List with local state management.

• Future enhancements could include task deletion, completion tracking, and backend storage for persistence.