**Project Description:**

The **Todo List Application System** efficiently handles task management by leveraging React state for real-time updates. Users can add and delete tasks, with future improvements including local storage for persistence and edit functionality.

### **Level 0 (Context Diagram)**

At the highest level, we have just one process, which represents the system as a whole.

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

| External Entity |

| (User) |

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

|

v

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

| Todo App System |

| (Process: 1.0) |

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

|

v

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

| Task Data Store |

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

**Explanation**:

* **External Entity (User):** The user interacts with the Todo App.
* **Process (Todo App System):** The app processes user actions like adding and deleting tasks.
* **Task Data Store:** Stores the list of tasks that the user has added**.**

### **Level 1 DFD (Decomposition of Process)**

Now, let's break down the **Todo List Application System** process (Process 1.0) into more detailed steps.

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

| External Entity | | Data Store |

| (User) | | (Task List) |

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

| |

v v

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

| Process: 1.1 - Input | | Process: 1.3 - Store |

| Task (User Enters) | | Task in Task List |

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

|

v

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

| Process: 1.2 - Add |

| Task to Task List |

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

|

v

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

| Process: 1.4 - Delete |

| Task from Task List |

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

**Explanation**:

### **Process 1.1 (Input Task):** The user types a task into the input field.

### **Process 1.2 (Add Task):** The task is added to the task list when the user clicks "Add Task".

### **Process 1.3 (Store Task):** The task list is updated and stored in the component's state.

### **Process 1.4 (Delete Task):** The user can delete a task, updating the task list.

### **Data Flow**

* The user enters a task in the input field.
* The user clicks "Add Task," and the task is added to the list.
* The task is stored in the application's state.
* The list of tasks is displayed to the user.
* If the user clicks "Delete," the selected task is removed from the list.

### **Notes:**

* The task list is stored in the React state (use State), meaning it does not persist after a page refresh.
* The application processes two main actions: adding and deleting tasks.
* A future enhancement could be adding local storage to store tasks persistently.