## **Flow Diagram for Drag-and-Drop Feature :**

**Start**

1. **User enters text and clicks "Add Card"**
   * User types text in the input field.
   * User clicks the "Add Card" button.
2. **addCard function validates input (Yes branch):**
   * The function checks if the input text is trimmed (has no leading/trailing whitespace).
3. **Valid input:**
   * A new card object is created with id and text properties.
   * The card’s state is updated by adding the new card object.
   * The inputText state is reset to an empty string.
4. **Cards are rendered:**
   * The component re-renders, and the cards array is mapped to create Draggable components.
5. **User interacts with a card (Draggable):**
   * The user drags a card using the draggable handle.
   * The react-draggable library handles the drag events.
6. **User drops the card within the canvas area:**
   * The react-draggable library handles the drop event.
   * (Currently, no specific logic is implemented for the drop target. You might need to add logic for specific drop zones in the future.)
7. **(Optional) User clicks "Show More" button:**
   * The button click triggers the handleShowMore function.
   * The function updates the selectedCard and showPopup states.
8. **Render popup (if showPopup is true):**
   * The component re-renders, and the popup modal is displayed with the full card text.
9. **User clicks "close" button in the popup:**
   * The button click triggers the closePopup function.
   * The function updates the showPopup and selectedCard states.
10. **Render updates:**
    * The component re-renders again, reflecting the changes in state.

**End**

**Thought process for designing a drag-and-drop feature:**

### **1. Understand Requirements**

* **Functionality: Enable users to create, drag, resize, and view detailed content in cards on a canvas.**
* **Flexibility & Visual Feedback: Ensure users can freely move and resize cards with clear visual cues.**

### **2. Choose Libraries**

* **React Draggable: Simplifies dragging functionality.**
* **React Resizable: Allows resizing of card components.**
* **React Flow: Enhances canvas visualisation with flow elements and controls.**

### **3. State Management**

* **Cards State: Store card objects in an array.**
* **Input State: Manage user input for adding new cards.**
* **Popup State: Control visibility and content of the detailed view popup.**

### **4. Implement Card Creation**

* **Input Handling: Capture and validate user input.**
* **Card Addition: Add new cards to the state and clear the input field.**

### **5. Drag-and-Drop & Resizing**

* **Draggable: Enable card movement across the canvas.**
* **ResizableBox: Allow dynamic resizing of cards.**

### **6. Popup Functionality**

* **Truncate Text: Show a preview on the card with a "Show More" button.**
* **Popup Display: Show detailed content in a popup on demand.**

### **7. User Experience**

* **Consistency: Ensure a cohesive and visually appealing UI.**
* **Flow Visualisation: Optional integration for future enhancements.**

### **8. Test & Refine**

* **Testing: Ensure smooth functionality across devices.**
* **Refinement: Improve based on feedback and testing.**

### **9. Scalability**

* **State Management: Efficiently handle a growing number of cards.**
* **Extensibility: Design for easy feature additions in the future.**