**Report**

**1. Introduction**

The Todo App is a simple task management application that allows users to add, remove, and mark tasks as completed. A key feature of the app is the **undo functionality**, which enables users to revert the most recent removal or completion of a task within 5 seconds.

**2. Design of the Undo Feature**

The undo feature is designed to provide users with a safety net for accidental actions, such as removing a task or marking it as completed. The design focuses on the following key aspects:

* **Time Constraint**: The undo option is available for only **5 seconds** after an action (removal or completion) is performed.
* **Single Undo**: Only the **most recent action** can be undone. Once the 5-second window expires, the undo option is no longer available.
* **No External Libraries**: The undo feature is implemented using only React's built-in state management and JavaScript's setTimeout function, adhering to the assignment constraints.

**3. Implementation**

The undo feature is implemented in the App component, which is the main component of the Todo App. Below is a detailed explanation of the implementation:

**3.1. State Management**

The app uses React's useState hook to manage the following state variables:

* **tasks**: An array that stores all the tasks in the app.
* **lastAction**: An object that stores the most recent action (either removal or completion) along with the task involved. This is used to implement the undo functionality.

**3.2. Handling Actions**

When a user performs an action (removing or toggling a task), the app stores the action in the lastAction state and sets a 5-second timeout to clear it. This ensures that the undo option is only available for 5 seconds.

**3.3. Undo Logic**

The undoLastAction function checks the type of the last action and reverts it:

* If the last action was a **removal**, the task is added back to the tasks array.
* If the last action was a **completion toggle**, the task's completion status is toggled again.

**3.4. User Interface**

The undo option is displayed as a notification at the bottom-right corner of the screen. It includes:

* A message indicating the type of action (e.g., "Task removed" or "Task toggled").
* An **Undo** button that triggers the undoLastAction function.

**4. Key Considerations**

* **Performance**: The app uses in-memory storage (an array) to manage tasks, which is efficient for small-scale applications. For larger applications, a more robust state management solution (e.g., Redux) or persistent storage (e.g., local storage) could be considered.
* **User Experience**: The undo notification is designed to be unobtrusive yet easily accessible, ensuring a smooth user experience.
* **Time Constraint**: The 5-second limit for the undo feature is implemented using JavaScript's setTimeout function, which is simple and effective for this use case.

**5. Conclusion**

The undo feature in the Todo App is implemented using React's state management and JavaScript's setTimeout function. It provides users with a safety net for accidental actions, enhancing the overall user experience. The feature is simple, efficient, and adheres to the constraints, demonstrating the effective use of React's built-in capabilities for state and lifecycle management.