

## Code review - ListComponent.tsx

The code was reviewed by Tomer Shor - 325511541

### Areas for Improvement

#### 1. Improve Code Readability

- a. Rename functions and variables to make their purpose clearer. For example, the function that retrieves a random item should specify that it only selects incomplete items. That means – changing the function “getRandomItem” to “getRandomIncompleteItem”.

```
// 🚫 Change Here: Rename function to clarify that it only picks incomplete items
const getRandomItem = () => {
  if (list.items.length === 0) {
    Alert.alert("Info", "No items in the list!");
    return;
  }
}
```

- b. Lack of documentation is very present in this code. I would suggest adding notes next to each move, especially next to untrivial functions.

#### 2. Reduce Code Duplication

-Extract repetitive logic into reusable helper functions. This will improve maintainability and make updates easier.

--Common operations such as updating the list and progress should be handled in a single function instead of being repeated across multiple actions.

#### 3. Fix State Mutation Issues

- Avoid modifying state variables directly. Instead, use immutable updates to ensure data integrity and prevent unexpected bugs.
- Functions modifying the list, such as marking an item as completed or deleting an item, should return a new state rather than altering the existing one. This way, the program could prevent some unexpected errors.

```
// 🚫 Change Here: Avoid direct mutation of state
const markAsCompleted = (index: number) => {
  const updatedItems = list.items.map((item, i) =>
    | i === index ? { ...item, completed: !item.completed } : item
  );
  💡 updateList(updatedItems);
};
```