

Here's how you can structure your `App.js`, `List.js`, and `Task.js` files using React components with `useState` and `props`.

App.js

```
import React, { useState } from 'react';
import List from './List';

function App() {
  const [lists, setLists] = useState([
    { id: 1, name: 'To Do', tasks: [] },
    { id: 2, name: 'In Progress', tasks: [] },
    { id: 3, name: 'Done', tasks: [] }
  ]);

  return (
    <div>
      {lists.map((list) => (
        <List key={list.id} list={list} />
      ))}
    </div>
  );
}

export default App;
```

List.js

```
import React, { useState } from 'react';
import Task from './Task';

function List({ list }) {
  const [tasks, setTasks] = useState([
    { id: 1, name: 'Task 1', completed: false },
    { id: 2, name: 'Task 2', completed: false },
    { id: 3, name: 'Task 3', completed: false }
  ]);

  return (
    <div>
      <h2>{list.name}</h2>
      {tasks.map((task) => (
        <Task key={task.id} task={task} />
      ))}
    </div>
  );
}

export default List;
```

Task.js

```

import React, { useState } from 'react';

function Task({ task }) {
  const [isCompleted, setIsCompleted] = useState(task.completed);
  const [isEditing, setIsEditing] = useState(false);
  const [taskName, setTaskName] = useState(task.name);

  const toggleCompletion = () => {
    setIsCompleted(!isCompleted);
  };

  const toggleEditing = () => {
    setIsEditing(!isEditing);
  };

  const handleNameChange = (e) => {
    setTaskName(e.target.value);
  };

  return (
    <div>
      {isEditing ? (
        <input
          type="text"
          value={taskName}
          onChange={handleNameChange}
        />
      ) : (
        <span
          style={{ textDecoration: isCompleted ? 'line-through' : 'none' }}
        >
          {taskName}
        </span>
      )}
      <button onClick={toggleCompletion}>
        {isCompleted ? 'Undo' : 'Complete'}
      </button>
      <button onClick={toggleEditing}>
        {isEditing ? 'Save' : 'Edit'}
      </button>
    </div>
  );
}

export default Task;

```

Explanation:

- **App.js** : Manages the lists of tasks. Each list is passed down to the **List** component as a prop.
- **List.js** : Manages tasks within each list. Each task is passed down to the **Task** component as a prop.

- **Task.js** : Manages the state of individual tasks, including completion status, editing state, and the task's name.

Each `List` and `Task` component has its own state managed by the `useState` hook. The states include:

- **List.js**:
 1. `tasks` - An array of tasks in the list.
- **Task.js**:
 1. `isCompleted` - Boolean for task completion status.
 2. `isEditing` - Boolean to toggle between editing and viewing the task name.
 3. `taskName` - The name of the task, which can be edited.

To Do

Task 1	Complete	Save
Task 2	Undo	Edit
Task 3	Complete	Save

In Progress

Task 1	Complete	Edit
Task 2	Undo	Edit
Task 3	Complete	Save

Done

Task 1	Complete	Edit
Task 2	Complete	Edit
Task 3	Complete	Edit