

# PRACTICAL INTERVIEW SCENARIOS

## 1. Build a Todo App with React

jsx

```
import React, { useState, useEffect } from 'react';

function TodoApp() {
  const [todos, setTodos] = useState([]);
  const [inputValue, setInputValue] = useState('');
  const [filter, setFilter] = useState('all');

  // Load todos from localStorage
  useEffect(() => {
    const savedTodos = localStorage.getItem('todos');
    if (savedTodos) {
      setTodos(JSON.parse(savedTodos));
    }
  }, []);

  // Save todos to localStorage
  useEffect(() => {
    localStorage.setItem('todos', JSON.stringify(todos));
  }, [todos]);

  const addTodo = () => {
    if (inputValue.trim()) {
      const newTodo = {
        id: Date.now(),
        text: inputValue.trim(),
        completed: false,
        createdAt: new Date().toISOString()
      };
      setTodos([...todos, newTodo]);
      setInputValue('');
    }
  };

  const toggleTodo = (id) => {
    setTodos(todos.map(todo =>
      todo.id === id ? { ...todo, completed: !todo.completed } :
      todo
    ));
  };

  const deleteTodo = (id) => {
    setTodos(todos.filter(todo => todo.id !== id));
  };
}
```

```

};

const editTodo = (id, newText) => {
  setTodos(todos.map(todo => {
    todo.id === id ? { ...todo, text: newText } : todo
  }));
};

const clearCompleted = () => {
  setTodos(todos.filter(todo => !todo.completed));
};

const filteredTodos = todos.filter(todo => {
  if (filter === 'active') return !todo.completed;
  if (filter === 'completed') return todo.completed;
  return true;
});

const stats = {
  total: todos.length,
  active: todos.filter(t => !t.completed).length,
  completed: todos.filter(t => t.completed).length
};

return (
  <div className="todo-app">
    <header>
      <h1>Todo App</h1>
      <div className="add-todo">
        <input
          type="text"
          value={inputValue}
          onChange={(e) => setInputValue(e.target.value)}
          onKeyDown={(e) => e.key === 'Enter' && addTodo()}
          placeholder="Add a new todo..."
        />
        <button onClick={addTodo}>Add</button>
      </div>
    </header>

    <main>
      <div className="filters">
        {[ 'all', 'active', 'completed' ].map(filterType => (
          <button
            key={filterType}
            className={filter === filterType ? 'active' : ''}
            onClick={() => setFilter(filterType)}
          >

```

```

        {filterType.charAt(0).toUpperCase() +
filterType.slice(1)}
      </button>
    )}
  </div>

  <div className="todo-list">
    {filteredTodos.map(todo => (
      <TodoItem
        key={todo.id}
        todo={todo}
        onToggle={toggleTodo}
        onDelete={deleteTodo}
        onEdit={editTodo}
      />
    ))}
  </div>

  <footer className="stats">
    <p>Total: {stats.total} | Active: {stats.active} |
Completed: {stats.completed}</p>
    {stats.completed > 0 && (
      <button onClick={clearCompleted}>Clear
Completed</button>
    )}
  </footer>
</main>
</div>
);
}

```

```

function TodoItem({ todo, onToggle, onDelete, onEdit }) {
  const [isEditing, setIsEditing] = useState(false);
  const [editText, setEditText] = useState(todo.text);

  const handleEdit = () => {
    if (editText.trim() && editText !== todo.text) {
      onEdit(todo.id, editText.trim());
    }
    setIsEditing(false);
  };

  const handleKeyPress = (e) => {
    if (e.key === 'Enter') handleEdit();
    if (e.key === 'Escape') {
      setEditText(todo.text);
      setIsEditing(false);
    }
  }
}

```

```

    };

    return (
      <div className={`todo-item ${todo.completed ? 'completed' :
''}`}>
        <input
          type="checkbox"
          checked={todo.completed}
          onChange={() => onToggle(todo.id)}
        />

        {isEditing ? (
          <input
            type="text"
            value={editText}
            onChange={(e) => setEditText(e.target.value)}
            onBlur={handleEdit}
            onKeyDown={handleKeyPress}
            autoFocus
          />
        ) : (
          <span
            onClick={() => setIsEditing(true)}
            className="todo-text"
          >
            {todo.text}
          </span>
        )}

        <button onClick={() => onDelete(todo.id)}>Delete</button>
      </div>
    );
  }
}

export default TodoApp;

```