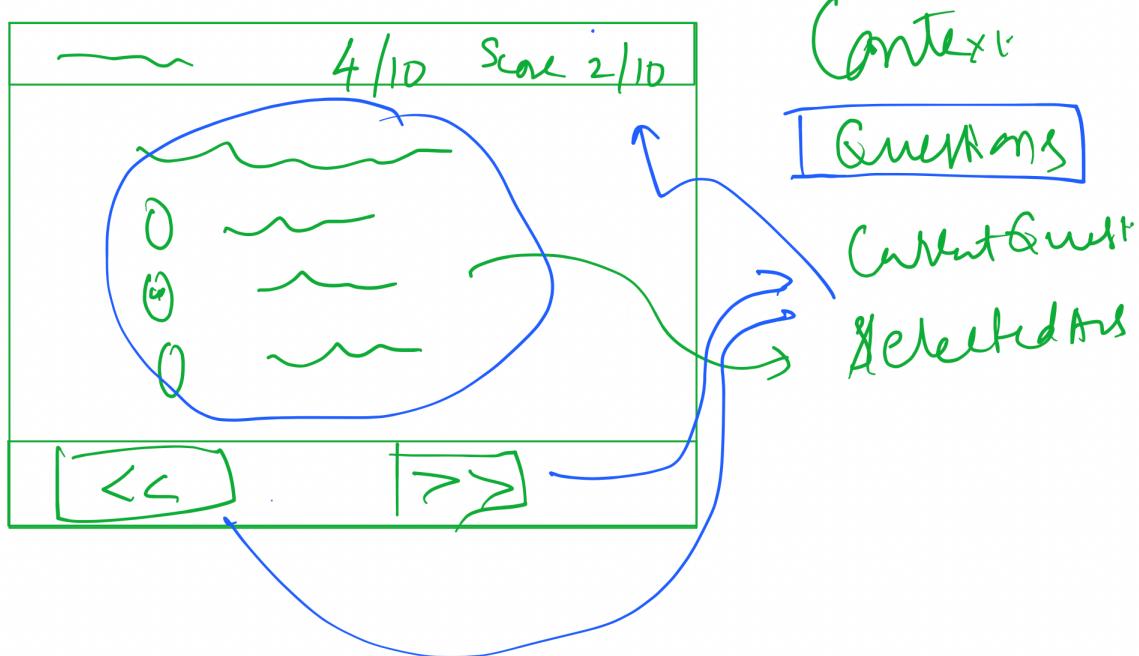


MULTI STEP PROGRESSIVE FORM



Feature	Redux	Context API
Primary Purpose	A standalone pattern and library for managing complex global application state with predictable updates.	A built-in React feature for "dependency injection" to avoid prop drilling for static or infrequently updated data.
Complexity & Setup	Requires more initial setup and boilerplate code, but modern Redux Toolkit (RTK) simplifies this significantly.	Minimal setup, easier to use for simple scenarios.
Performance	Highly optimized; components only re-render when the specific data they select changes.	Can cause performance issues in large applications because all consuming components re-render whenever <i>any</i> part of the context value changes.
Debugging	Powerful Redux DevTools offer time-travel debugging and detailed action tracing.	Limited debugging tools; difficult to trace state changes over time.
Middleware	Supports middleware (e.g., Redux Thunk, Saga) for handling side effects like API calls and complex logic.	Does not natively support middleware; side effects must be managed with <code>useEffect</code> or other hooks.
Scope	UI-agnostic; can be used with React, Angular, Vue, or vanilla JavaScript.	Limited to React applications as it is a core React feature.

