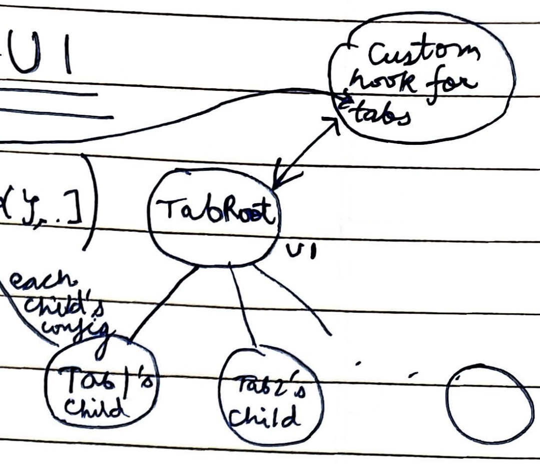
Tab - based FORM - UI

1) Config - driven UI (tabs = [{}, {},..])



```
{
    name : 'Profile',
    Component : Profile,
    validate : () => {. . . },
    validateWithoutSetting State : . . .
                          error
}
```

→ Also Custom Hook will have a State for
- data of all tabs for their fields
- error of all tabs's fields

2) We have a relative global State in Custom hook (or TabRoot) for all child States.

3) In TabRoot we have a state :
- activeTab for representing current childTab & a derived state which has current Component from tabs that we display

```
const ActiveComponent = tabs[activeTab].component;
   in JSX → <ActiveComponent props... />
```

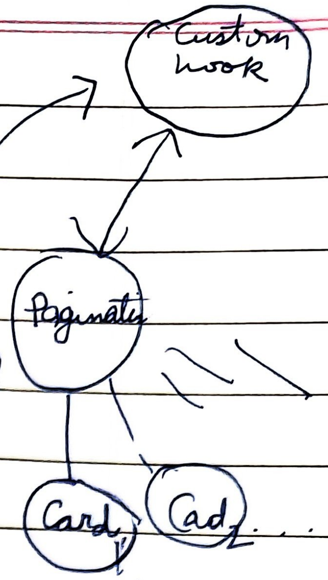
4) Based on events that happen in TabRoot & it's childs we manipulate the states in TabRoot & Custom hook.

→ Also basically we might do some things like setting a state directly or might put a condition.

→ We pass states & setters as props to child or we can use useContent or Redux Toolkit.

FE Pagination

⇒ We will define all states, derived states & useEffect in



⇒ We will import those states & derived state in (via custom hook)

⇒ Those cards will be rendered via .map

⇒ Also we will have states like these in hook:

→ data, error, load → States
[{},{}...]
all records

→ itemsPerPage

use Memo these

→ pages, dataOfSlices → derived States
[[{},{}..],[],...]

→ numberWindows (of a fixed size let be 10) or length of dataOfSlices < 10
(using dataOfSlices)
[[0,1 -- 9],[10,--],--]

→ currentWindow, currentItemInWindow both should be states ∵ they both depend on user events not on other states.