

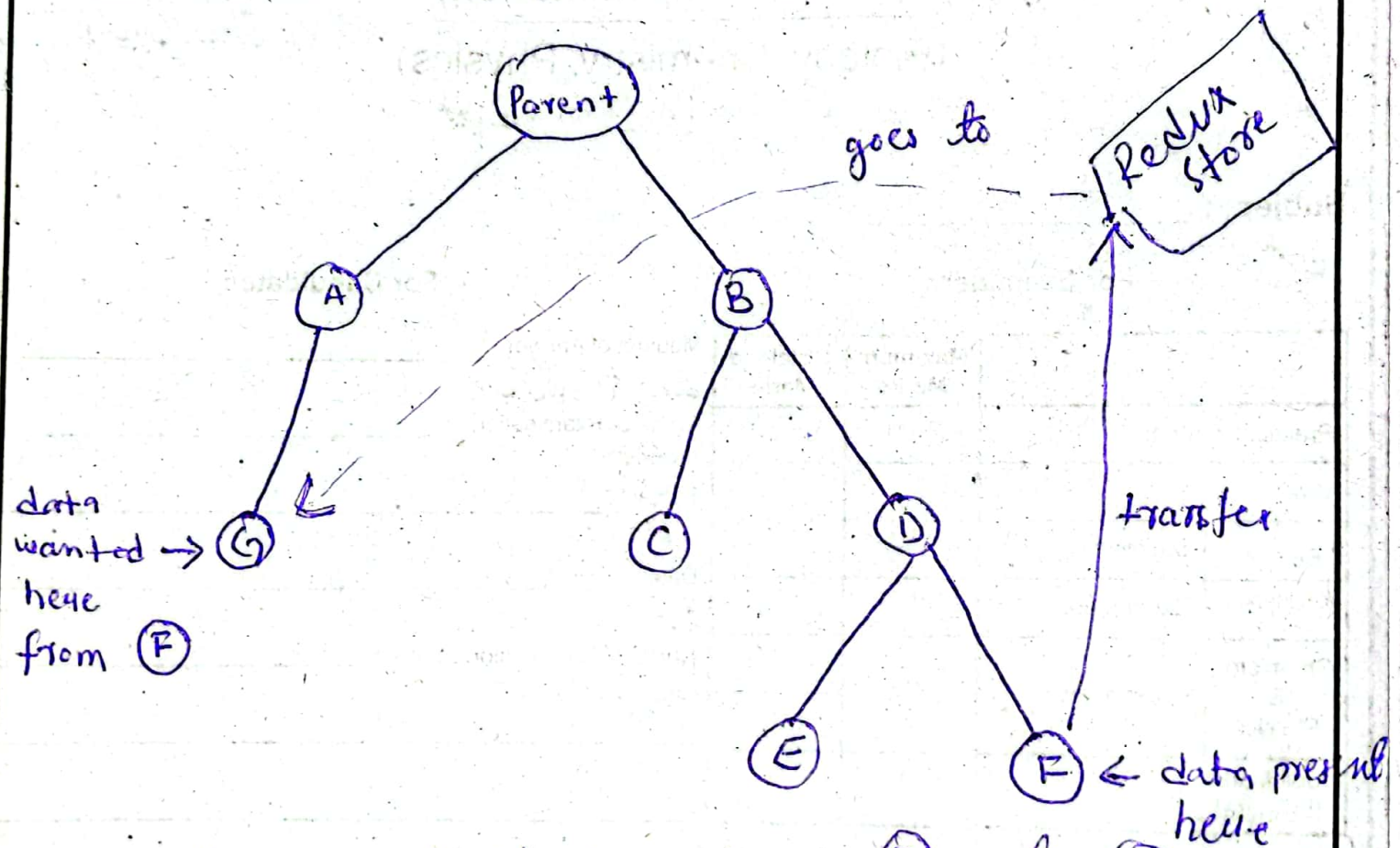


Why we need Redux toolkit?

→ Bcz of ~~state~~ management

(i) Less code

(ii) Easy async operation: it makes handling things like API-calls and other asynchronous task easier.



→ Transferring of data from (F) to (G) made easy with the help of redux.

→ If we are not using redux then we have to transfer data from (F) to (D) then (D) to (B) then (B) to Parent then Parent to (A) then (G) it take alot of code.

→ If Without -using redux we can transfer data with the help of context-API

but it is not good for large data.



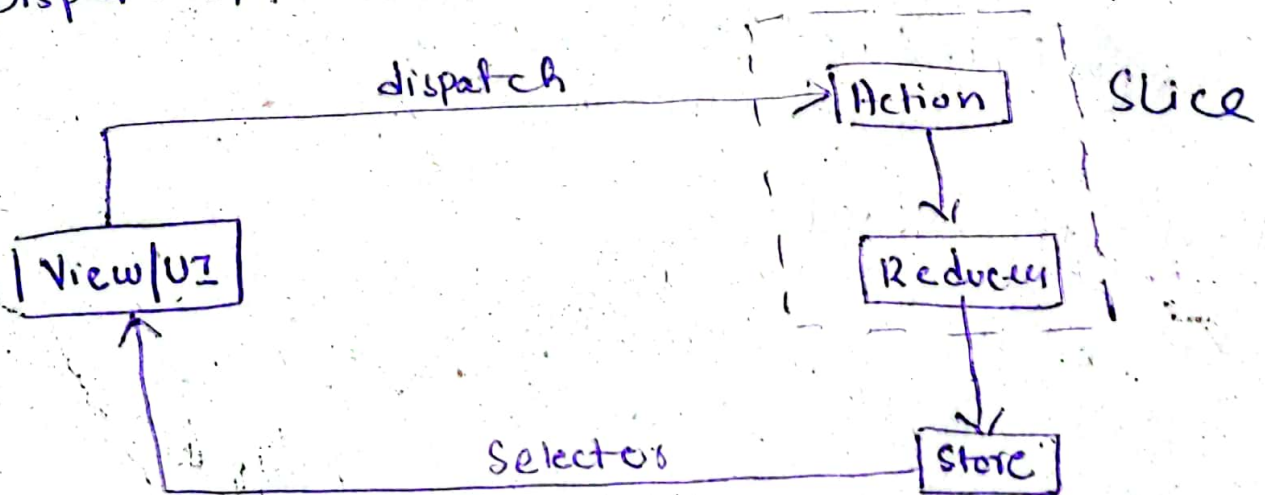
3

* REDUX Vs REDUX toolkit (RTK)

- Both are same purpose & use
- We need more code in redux
- In redux we create two file actions & reducers while in redux toolkit both these file comes under file slice.

* REDUX Main Part / Component:

- ① Action : Plain object / human data
- ② Reducer : Data ko save karwata hai
- ③ Store : store room
- ④ Dispatcher : Selected

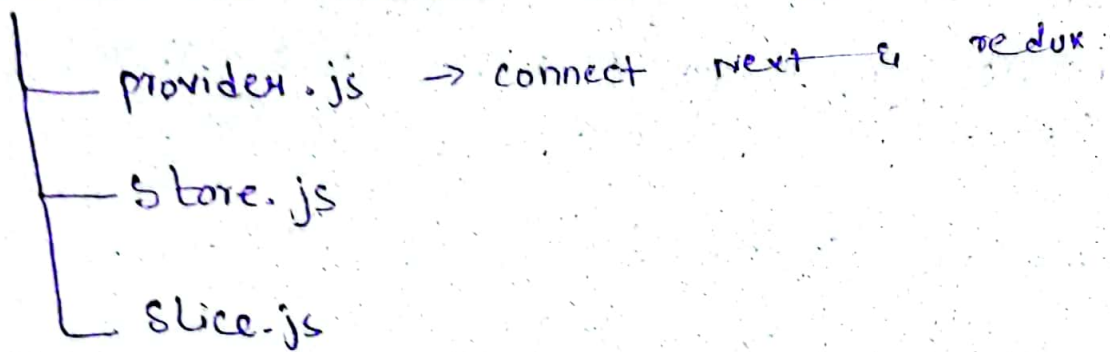


① Jb be human page ko refresh karty hein to redux store se be data chala jata. Agr ap ny redux ka data local storage me store karwana hai to ap Library use kar skty



* REDUX - Tool Kit = FOLDER STRUCTURE

redux ← folder in app



* Redux ka data flow kya hota hai?

Ans: (A) Bi-directional → two ways
 (B) Uni-directional → one way

⊕ What is slice?

→ A slice is a collection of Redux reducer logic and actions for a single feature in your app

→ In slice we define action & reducer

→ Action: Data which we are going to store in redux-store

→ Reducer: Thing which store action data in store is reducer

Slice code



5

```
const initialState = {  
  users: []  
}
```

```
const slice = createSlice({ name: "todo",  
  initialState,  
  reducers: {
```

```
    addUser: (state, action) => {
```

```
      const data = {
```

```
        id: nanoid(),
```

```
        name: action.name  
                payload
```

```
      }
```

```
      state.users.push(data)
```

```
    }  
  }
```

Action

```
export const {addUser} = slice.actions;
```

```
export default slice.reducer;
```

Store.js

reducer: reducer

Component is not data store data

```
const [name, setName] = useState("");
```

```
const dispatch = useReducer();
```

```
const handleSubmit = () => {  
  dispatch(addUser(name))  
}
```

```
onclick = {handleSubmit}
```




GET IDATA FROM REDUX to NEXTJS

→ Import useSelector hook

Jis mei display karwana us mei

Const UserData = useSelector ((data: AppState) => data.user)

```
data.map((item) => {
  <div <div> }
```

Q: Kia Nextjs mei multi-store ho skti

Ans: No

REMOVE DATA FROM REDUX

→ Make action for remove data

→ make remove button with each user

*PASS REDUX IDATA BETWEEN PAGES

Const UserData = useSelector ((data) => data.user)

Const dispatch = use dispatch ();

return

```
{ UserData.map((item) => (
```

```
<div>
```

```
<div> Item name
```

```
<div> <btn> onclick =
```

```
{ () => dispatch(remove User {id}) }
```

* REDUX WITH LOCAL - STORAGE

slice.js



7

addUser (state, action) => {

const data = {

id: nanoid(),

name: action.payload

}

state.user.push(data)

~~localStorage.set~~

let userData = JSON.stringify(currentState.user)

localStorage.setItem('user', userData)

TO GET BACK

initialState

user: JSON.parse(localStorage.getItem('user'))

MAKING SECOND SLICE

→ Define initial state

→ make slice

→ Add Reducer

→ Check flow

Q: Kia Use Selector or hook
ka hook hein?

Use dispatch React

Ans: Nahi, ye dono
K connector K hook

React-redux naam
hein.



Q Core principles of Redux?

- ① Single source of truth (Jo be data hoga store mei single obj) hoga
- ② State is read only
- ③ Changes are made with pure function
Reducers

Q What is Redux?

- State management Library
- Redux we can share data b/w components

Q What is useDispatch?

- Hook in Redux-react
- Used for pass data from component to Redux store

Q What is difference b/w store & state?

- The store is an object that holds the entire state of the app.
- The state is the data that the app need to render its UI

Q What is Provider in React-Redux?

make redux store available to any components.

Q: Alternative of Redux

- MobX
- Recoil
- Context API

Q: How to Debug Redux?

- Use the Redux Devtools extension.
- Log action and state changes.
- Use unit test.
- Browser debugger.