* Main work of REDUX is to manage the passing of data in an organized form
* Context API ka main kaam yhi hai ki ek global variable hoga aur usime sara data jaega
* React is a library
* Frameworks, on the other hand, typically provide a **full-fledged** structure for building applications, including routing, state management, and backend communication.
* For Step1: make the Context folder and UserContext.js then

import React from "react";

const userContext = React.createContext()

export default userContext;

* When we write like

<userContext>

    <Data />

    <Contact />

<userContext/>

Then userContext become a provider and all the inside elements like Data and Contact will have access to the context

* Context is just a global variable
* Hm chahate to Global file bnake usme sare variables fit krdete but usse problem ye hoti that if we try to update one variable then it might occur unwanted changes in other variables

**REDUX**

* Redux is an independent library not related with just React
* useDispatch reducers ko use krke store me value dispatch krta hai
* useSelector is just a method with the advantage that we get the access of state in a callback
* In general whenever we have to pass the parameters, we generally use callback function i.e. () => {}
* Redux is a core library while redux-toolkit is an implementation or wiring so that redux and react can communicate easily
* It allows us to not manually change the state
* One application **should have** **only one store**
* In Context API, the state is not preserved that’s why we have to use spread operator while adding the todo.

Whereas, in Redux, the state is preserved that’s why we can directly push the new todo in the initial state