React-redux- Day -12

Basics Of Redux : group of state transition

State Management Library

It provides a centralized way to manage the state of your application.

To handle complex state management needs

Why Redux:

Complex State Management

Predictable Data Flow

Large-Scale Applications.

Steps for creating Redux:

1. **CreateSlice->** {createSlice} from **'@redux/toolkit'**

   name, initialState,reducers (3 things important)
2. **Creating store** -> {configureStore -> reducer{}} from '**@redux/toolkit'**
3. **Provider** -> {Provider} from **'react-redux**' and **pass store** as a property
4. **UseSelector()->** {useSelector} from 'react-redux' – **fetch purpose**
5. **UseDispatch()**-> {useDispatch} from 'react-redux' – **update purpose**

Redux Toolkit:

Redux Toolkit is designed for **complex state management** in larger applications. It is suitable for applications with **multiple components** that need to share and update global state, and where state changes can become complex.

Context API:

The Context API is more straightforward and is typically used for **simpler state management** needs within a subtree of your component hierarchy. It's useful when you want to pass data **down to deeply nested components without the need for prop drilling.**

Example-1 -> simple operation

Example-2 -> API Calling real time working

Explanation Task with Demo/ existing Code

Note: Advanced Concept: **extraReducer, createAsyncThunk(), Api's call workout**.