**Introduction to Redux & Redux toolkit**

**Redux:-**

* Redux means "brought back" or "bringing back.".
* Redux is an open-source JavaScript library for managing and centralizing application state. Redux is a predictable state container for JavaScript apps.   
    
  It helps you write applications that behave consistently, run in different environments (client, server, and native), and are easy to test. On top of that, it provides a great developer experience, such as live code editing combined with a time traveling debugger.   
    
  You can use Redux together with React, or with any other view library.
* As the application grows, it becomes difficult to keep it organized and maintain data flow. Redux solves this problem by managing application’s state with a single global object called **Store**.   
  Redux fundamental principles help in maintaining consistency throughout your application, which makes debugging and testing easier. More importantly, it gives you live code editing combined with a time-travelling debugger. It is flexible to go with any view layer such as React, Angular, Vue, etc.
* For Example:  
  Let consider state is value of a variable.  
  By default React only allow single flow of data transfer and it would be Top to Bottom.  
  Let say,  
  <App/> 🡪 <Home/> 🡪 <Post/>  
  There are 3 components. A react Application having Outer Component App containing Home Component and just like Home containing Post. Flow Outer Component 🡪 Inner Component.  
  We can easily transfer data i.e. (a variable c= 20;) from App 🡪 Home 🡪 Post.  
  But we can’t transfer data i.e. (a variable c= 20;), from Post 🡪Home or Home 🡪App or Post 🡪 App.  
  So, this is where, Redux comes in Handy.  
    
  So, here variable 20 i.e. the value of c is a State.   
  Changing value of c == Changing state

**Redux Terminologies:-**

Store, Reducer, Action & State

**Store**: A global Object, which is going to store data and can be access from anywhere throughout the application.

**Reducer**: Reducer is a function that helps in changing the state or state manipulation.

**Action**: Action helps in triggering the reducer.  
  
**State**: Value or any Data

**Redux Toolkit:-**

* **Redux Toolkit** is our official standard approach for writing Redux logic.  
  It wraps around the Redux core, and contains packages and functions that we think are essential for building a Redux app. Redux Toolkit builds in our suggested best practices, simplifies most Redux tasks, prevents common mistakes, and makes it easier to write Redux applications.
* Redux Toolkit is available as a package on NPM for use with a module bundler or in a Node application:  
  *“npm install @reduxjs/toolkit”*For React Project:  
  *“npm install @reduxjs/toolkit react-redux”*