**Day 7:**

**If we declare any state variable. That variable is local to that component.**

**Variable can number type, string type, object type, array type or array of object type.**

**Component communication:**

Component relationship

1. Parent to child for div, h1 and p are children tags
2. Child to parent for h1 div is parent and for p div is parent
3. Sibling relationship h1 and p are sibling tags.

<div>

<h1>This is heading tag</h1>

<p>First Paragraph tag </p>

</div>

Parent -🡪Component : inside a parent component JSX if we use child1 component as well as child2 component. Then child1 and child2 are children for Parent component. Child1 and Child2 are sibling.

Child1 -🡪 Component

Child2 -🡪 Component

1. Parent to child: -🡪 if parent component contains any state variable. If we want to pass the value from parent to child. We can use props concept. (props drilling)
2. Child to parent:-> if child component contains any state variable. If we want to pass this value to parent component. then we need to use props with callback function concept.
3. Child1 to child2 or vice-versa 🡪 if child1 contains some state variable. If we want to use that state variable in child2 component. Child1 need to pass the value to parent using props with callback and parent will pass this value to child2 using props. (props lifting).

npx create-react-app react-component-communication

cd react-component-communication

open the project in VS code

npm start

**Creating another react js project with component communication concept.**

**npx create-react-app react-rating-app**

cd **react-rating-app**

**open the project in VS code**

**npm start**

**App --🡪 root component**

**RatingApp** -🡪 Parent Component

**RatingSelector** --🡪 \* \* \* \* \* (user can select the rating)

This component send the user rating details to parent using props with callback

**RatingSummary** -🡪 display rating summary (receive the rating details from rating selector and pass to rating summary component using props concept.)

App ---🡪Root

Parent -🡪 id =100 as state variable

Using props we can pass parent to child1,child2 and child3

Child1 Child2 Child3

Child1 can pass value to child4 and child5 using props

Child4 Child5 Child6 Child7 Child8

Child9 Child10

When we want to pass the value from parent to child in n level we need to use props drilling concept. If any link lost we can’t access those values.

**React Context API:** React Context API is one of powerful tool which help to pass the value from parent to children till n level without props drilling. In Context API no need to keep the track about relationship level. But those component must be under one tree structure.

If we use Context API concept to pass the value we can avoid props drilling to pass the value for every component explicitly.

First we need to create the context api reference using createContext() hook.

In Parent component wrap component reference which provide one tag as Provider. Which help to provide any type of value on down level. So value can be any type like number, string, object, array even function also pass.

Then we need to use useContext() to access the value from parent to child on any level.

**npx create-react-app context-api-app**

cd context-api-app

open the project in VS code

npm start

**npx create-react-app context-api-operation**

cd **context-api-operation**

open the project in VS code

npm start

App -🡪root

UserComponent --🡪 Parent Component

AddUser DisplayUser -🡪Children

**Redux :** Redux is a predictable state container tool for JavaScript application.

Using Redux we can make local state variable as global state variable.

It helps to manage the local state variable of entire application in a single global store. If we use redux which help to track the changes on state variable, easy to debug and easy to test.

Redux Vs Context API

It is an external it is a part of react. Which help to

State management tool. Avoid props drilling and mainly

Use for basic state sharing.

It is use in enterprise or complex it is use to small application

application level

concept we use as

global state, action, dispatch minimum setup

and store, **reducer**

it support middleware features middleware not support any third party middleware

hook in redux hook in context api

useSelelector(), useDispatch() createContext(), useContext()

Redux we can use with plain JS, React as well as Angular (ngrx as redux module)

**npx create-react-app react-with-redux-app**

cd react-with-redux-app

**npm install redux react-redux**

open the project in VS code

npm start

**reducer :** reducer is a simple JavaScript function which takes two parameters.

1st parameter is initialState and 2nd parameter is action. Base upon action it will do the change on global state variable part of store.

Store : store is a part of redux module (which we installed). Store help use to make variable as global state variable. While creating store we need to take the help of createStore (legacy) and need to pass reducer reference. Store connect to reducer functions.

Redux provided one of the pre defined tag as Provider.

This tag is part of react-redux module which you need to wrap for parent

Component with store reference to enable redux store features in all child component.

We ned to use

useSelector() hook to access the global state variable part of store.

useDispatch() is use to dispatch the action to reducer function do to change on global state variable.