React

React uses syntax extension of javascript known as JSX meaning Javascript XML

JSX => It allows us to write HTML code inside Javascript

Ex:

```
<footer> ⟶ HTML
  <hr> </hr>
  <p> {new Date().getFullYear()} </p>
</footer>
```

↓ Javascript

Javascript uses virtual DOM → which is a light weight version of real DOM

Use of Virtual DOM :=> It only updates the particular not fully

This reduces the rendering time

Requirements to learn React:

Learn JS:

- arrays
- classes
- objects
- arrow function

HTML:

We can write JS code inside HTML like,

```
<ul>
  <li> Apple </li>
  <li> {food1} </li>
  <li> {food2} </li>
</ul>
```

const food1 = "Banana"
const food2 = "Orange"

- Apple
- Banana
- Orange

We can make the word inside the JS as uppercase by using

```
<ul>
  <li> {food2.upperCase()} </li>
```

Note:

JavaScript codes are inside { }

Note:

Outside return dont need to write inside { }

but in return JS should be written inside { }

*In react everything is called as Component

Component:

Component is a section of reusable JSX code

Card Components:

It will be useful in image, title and description

Lets take an example:

We have to print one image, title, description.

Create a component Card() {

```
function Card() {
  return (
    <img src={"images"} alt="Myprofile" className="temp">
    </img>

    <h2> Bro code </h2>
    <p> This Card Component in React </p>
  );
```

box-shadow:
for shadow

Note
In css to make round image
make border-radius: 50%

Spread Operator

Pasing array into useState
Ib

Ex:

```
Spread operator example
const arr1 = [1,2,3];
const arr2 = [4,5,6];

const combinedarray = [...arr1,...arr2];

console.log('Array1:' arr1);
console.log('Array2:',arr2);
Console.log('combined array:', combinedarray);
```

UseEffect

Upon the condition or action we apply in our functional components monitoring the impact or side-effects can be done using useEffect hook.

Useeffect accepts two arguments
(i) One is callback function and dependency array

Note:
Callback function like constructor in Java

- To fetch the data in form;
e.target.value

* Usestate returns an array of two values,
(i) Initial state
(ii) Updated state

* state can be passed as props but in hierarchy

Note:
Totally we have 5 components Let them as C1, C2, C3, C4, C5
Every component should display the name as component1
C1 is parent
C1 child is C2
C2 child is C3
C3 child is C4
C4 child is C5

* Direct export:

```
export class
```

default export

```
export default class
```

Note:

Whenever we are using something inside { }
it can be either javascript object or react component

Note

props can be passed between components only by following the hierarchy

which means if we render the component then only it is possible.

To overcome this in terms of efficiency we are using hooks.

That hooks is useContext():

Conclusion:

If we want to use state from component to another component the only way to achieve is passing it as props in the hierarchy.

This is not efficient to make it efficient we have an exclusive hook called useContext()

Create 4 component

App.js

Container.js

Users.js

User.js

Create a useState in App and theme, setTheme as array variable

UseContext

Without following the hierarchy passing state from one component to another component in an effient way using the hook.

Two important things.

(i) Create the context

(ii) UseContext

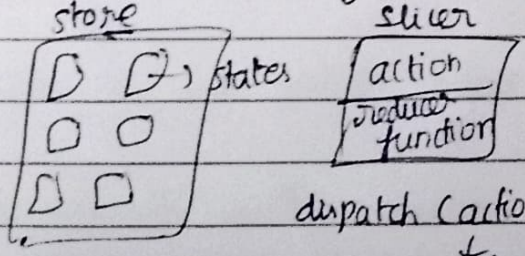
In the given example createContext will be done in App component that will be used in user component using UseContext will be done in user component

UseEffect

Synchronizing a component with external systems.

After our action monitoring or check in or see in the side effects happening in the functional component is possible using useEffect hook

Redux is a library or framework



dispatch() => it is an useReducer() hook.

To install redux, npm i @reduxjs/toolkit react-redux

↓ ~~Javascript~~ redux library

↓ Package (Linkage btw react and redux)

to create store, slices and all they given one library that is reduxjs/toolkit.

Router inside Redux

router
routes
route

Home
about
Contact

} Each picture

~~to~~ Work with redux

create two folder under src named as app and slices

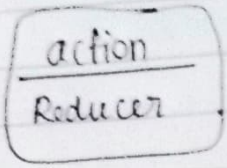
~~=~~ useSelector hook - to access data from redux

So, go and update users.js bcs u gonna get data from home.js to users.js
users.js [updated]

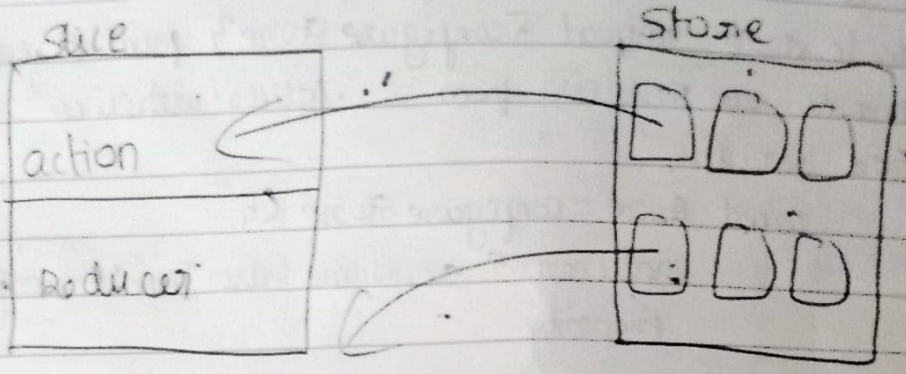
---

state is reducer userinfo from store users from slice

UseReducer



Redux:



Redux contain 2 parts:
(i) Store
(ii) slice

Store contain multiple state
That each state contain seprate slice
Each slice contain 2 things
(i) action
(ii) Reducer

Reducer: is a function that updates the state
But in Reducer we can't directly call the reducer function by using actions we call (trigger) the reducer through dispatch (actions)

Note: we can create more than one reducer in slice

to export each reducer,
export const { setUser, deleteUser } = userSlice.actions;
↓ reducer function
↓ It will take the all reducers

to export whole reducer
export default userslice.reducer;

To createSlice, use, createSlice( )

To use those slice in store,
inside store import {configureStore} from '@reduxjs/toolkit';
import userReducer from '../slices/userslice';
Store variable,
const store = configureStore ({
reducer: { anyName (userInfo): userReducer, }
property

//Then we have to export them
export default store;

To connect with React.,
go to main.js or App Js
import {Provider} from "react-redux";
import store from "./app/store.js";

ReactDOM.
<Provider store={store}>
<App />
</Provider>
);

To use the redux in our application.
if we enter data in one component, that will reflect in another component is possible using Redux.

if we want to see the changes that made in Home.js in see in User.js
in User.js,
import {useSelector} from "react-redux";

```
const Users = () => {
  const users = useSelector((state) => state.userInfo.users);
```

state represent the store
This store's component
users is initial value in the object

```
  console.log(users);
```

How to store data in redux:

create one functions in where you want to send data

useSelector() => get data from redux
useDispatch() => send data to redux

```
const dispatch = useDispatch();
const add = () => {
  event.preventDefault();
  dispatch(setUsers(payload))
```

send input

It is a function (reducer in slice)

When use Redux:

When there is no connection b/w component but we have to send data.

If To avoid refresh, type command as

event.preventDefault();

Now, we just print them in console, to show them in screen, use map function of JS

(1) Go to the component where we have to ~~add~~ display the data

~~return~~ (
<h1>{user.name}

return (

{users?.map((user, index) => {
return (

<h1>{user.name}</h1>
<h2>{user.age}</h2>
<h3>{user.email}</h3>
<h4>{user.phone}</h4>

)})}

<input type=text value={name} ></input>

here value is an javascript object

Bootstrap

CARD → It's a container with below things
card
card.img

MongoDB

MongoDB is act like server
Mongosh is act like user.

connecting MongoDB with react we need mongosh

JSON → It's noSQL
We use NoSQL to handle unstructured data.
Ex: JSON object (JSON) → is a Javascript object (looks like)

Compass:

Compass helps to fetch data from mongo db server
means compass helps us to reach mongoDB server. like its client

Mongosh:

Mongo Shell was replaced by Mongo.Sh

Mongosh gives us the interactive environment where you can run your queries, manage database, and perform administrative tasks.

Data Modeling:

Fix structure of your data, planning the structure.

ex:

name
id
Password

Schema:

actual blue print or structure of ur data / which you created by using data modeling.

Ex:

Employees
Emp.name: String
Emp.age : Intego
Salary : float

No SQL

SQL - Record
Mongo Document

SQL Table
Mongo - collection

Collections are stored in DB

Mongo will have multiple DB

Go to Mongosh
Then go click Enter
It will show tests
Then type command [use aiml]
↳ collection Name

db.createCollection("emp")

db.emp.insert

Qu
Create db computers
One collection name laptops
documents: name, model, color, status, price, (available, Notavailable)
vendar: {v-name, v-price}

(i) list out particular model laptops.
(ii) change its status to available if it is unavailable, i.e. and vice versa.
(iii) To delete all data. // : drop()

to delete database => db.dropDatabase()

Que

Filter all the records from computers database 32000 to 50000 and Brand as Dell and then update them add the shipping document as and make them as yes

Create collection named details, and the data modeling are id, name, age, city
Schema: number, string,

$nin : not in
db.customer.find({hobbies : {$nin : ['cooking']}})

This query will return all documents where customers not having hoobies as cooking

---

$and $or
db.customer.find({$and : [{price: {$gt: 5000}}, {brand: 'Apple'}]})
It will fetch the data if both of them are true

$or => It will fetch data if any one of them is true.

$exists : false

It will fetch data who didn't have one particular field.
db.customers.find({$and: [{hobbies: {$exists: false}}, {age: {$gt: 40}}]})