**REACT JS CODING NOTES DOCUMENT**

**[05-05-2023]**

**#REACT JS**

ReactJs - developed by facebook(Meta) - 2011

component - function , it will return html elements

reactjs - to create reusable UI(user interface) components

Nodejs - run time environment for JS, you can run outside the browser

cmd - node -v

DOM – Document Object Model

**#Reacts Js Benefits:**

\*build UI

\*reusable components

\*virtual DOM

\*fast and responsive

\*single page application

It is used to create the project - npx create-react-app projectname

to run or start the project - npm start

ctrl + C - to stop the react application

===========================================================================

**[08-05-2023]**

npm - node package manager - it is use to install the packages.

to create the project - npx create-react-app my-firstapp

to run project - npm start or npm run start

jsx - Javascript extension - used to write html inside JS

**#Use this code in App.js**

function App(){

return (

<div>

<h1> Welcome to React JS</h1>

</div>

)

}

export default App;

===========================================================================

**#keep following code inside index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render

(

<App />

);

**[09-05-2023]**

**# App.js**

import Mobile from "./Mobile/Mobile";

function app(){

return(

<div>

<Mobile/>

<Mobile/>

<Mobile/>

</div>

)

}

export default app;

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**# Mobile.js**

import './Mobile.css';

function Mobile(){ //function-keyword //Mobile-component name [folder name]

return (<div className="box">

<h1>Mobile Name:Sony</h1>

<h2>Amount:40000</h2>

<h3>Discount:10%</h3>

</div>)

}

export default Mobile;

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**# Mobile.css**

.box(

border:1px solid black;

width:500px;

height:200px;

)

===========================================================================

**[10-05-2023]**

**# Props – Properties**

it allows you to pass data from one component to other components (argument)

it also used to store date that can be accessed by component

**# Mobile.js**

import './Mobile.css';

function Mobile(props){

return (<div className='box'>

<img src={props.image}/>

<h1>Mobile Name:{props.name}</h1>

<h2>Amount:{props.amount}</h2>

<h2>Discount:{props.discount}</h2>

</div>)

}

export default Mobile;

===========================================================================

**# App.js**

import Mobile from "./Mobile/Mobile";

import pict1 from "./assets/phone1.png";

import pict2 from "./assets/phone4.png";

import pict3 from "./assets/phone5.png"

function App(){

return (

<div>

<Mobile image={pict1} name="Apple" amount="65000" discount = "23%"/>

<Mobile image={pict2} name="Samsung" amount="55000" discount = "13%"/>

<Mobile image={pict3} name="Nokia" amount="25000" discount = "3%"/>

</div>

)

}

export default App;

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**[11-05-2023]**

**# BOOTSTRAP USING INSIDE REACT JS**

Use 4 important links of bootsrap:

-This has to be pasted inside index.html (head tag)

-UseState: - API/hook - Special function

\*UseState & \*Without UseState

===========================================================================

**# Without use state:**

function App(){

let fullName = "Kumar";

const changeDetails=(event)=>{

console.log(event.target.value);

fullName = event.target.value;

console.log(fullName);

}

return(

<div>

<h6>Input Value: {fullName}</h6>

<input type="text" onChange={changeDetails}/>

</div>

);

}

export default App;

===========================================================================

import {useState} from "react";

const [currentvalue,Updatevalue]= useState('initial value');

Onchange:

import { useState } from "react";

function App(){

const[inputValue,updateinputValue] = useState('Welcome');

const changeDetails=(event)=>{

updateinputValue(event.target.value);

}

return(

<div>

<h6>Input Value: {inputValue}</h6>

<input type="text" onChange={changeDetails}/>

</div>

);

}

export default App;

===========================================================================

**# OnClick:**

import { useState } from "react";

function App(){

const[click,setClick] = useState(0);

const updateclick=()=>{

setClick(click + 1);

}

return(

<div>

<h6>You pressed {click} times</h6>

<button onClick={updateclick}> Click Me </button>

</div>

);

}

export default App;

===========================================================================

**[16-05-2023]**

**# App.js**

import Form from "./Form/Form";

function App()

{

return (<div>

<Form/>

</div>)

}

export default App;

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**# Form.js**

import { useState } from "react";

import './Form.css';

const Form=()=>{

let [getchangeInput, setchangeInput] = useState(0);

let [getchangeInput1, setchangeInput1] = useState();

let [getchangeInput2, setchangeInput2] = useState();

let [getFlag,setFlag] = useState();

const onInputChangeHandler1=(event)=>{

setchangeInput1(Number(event.target.value));

}

const onInputChangeHandler2=(event)=>{

setchangeInput2(Number(event.target.value));

}

const onAdditionHandler=()=>{

setchangeInput(getchangeInput1 + getchangeInput2);

}

const onSubtractionHandler=()=>{

setchangeInput(getchangeInput1 - getchangeInput2);

}

const onMultiplicationHandler=()=>{

setchangeInput(getchangeInput1 \* getchangeInput2);

}

const onDivisionHandler=()=>{

setchangeInput(getchangeInput1 / getchangeInput2);

}

const onSubmitHandler=()=>{

setFlag(true);

}

const onResetHandler=()=>{

setFlag(false);

setchangeInput1('');

setchangeInput2('');

}

return (<div>

Value of A <input type ="text" name = "Value of A" value={getchangeInput1} onChange={onInputChangeHandler1}/>

Value of B <input type ="text" name = "Value of B" value={getchangeInput2} onChange={onInputChangeHandler2}/>

<button className = "operation" onClick={onAdditionHandler}>Addition</button>

<button className = "operation" onClick={onSubtractionHandler}>Substraction</button>

<button className = "operation" onClick={onMultiplicationHandler}>Multiplication</button>

<button className = "operation" onClick={onDivisionHandler}>Division</button>

<div>

<button className ="submit" onClick={onSubmitHandler}>Submit</button>

<button className ="submit" onClick={onResetHandler}>Reset</button>

{getFlag ?<h1>Result is :{getchangeInput}</h1>:null}

</div>

</div>

)

}

export default Form;

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**# Form.css**

h1{

text-align: center;

color: green;

margin-top: 10px;

}

input{

display: block;

margin: auto;

width: 100%;

height: 25px;

text-align:center;

}

.operation{

display: block;

margin: auto;

width: 120px;

height: 50px;

text-align: center;

color: gold;

background-color: gray;

margin-top: 10px;

}

.submit{

display: block;

margin: auto;

width: 120px;

height: 50px;

text-align: center;

color: green;

background-color: white;

margin-top: 10px;

}

===========================================================================

**[19-05-2023]**

register .js import './Register.css';

import Header from '../Header';

import { useState } from 'react';

import { useNavigate } from 'react-router-dom';

const Register=()=>{

const[getForm,setForm] =useState({

FirstName:'',

LastName:'',

Email:'',

Password:''

}

);

const navigate = useNavigate();

const onChangeHandler=(event)=>{

setForm({...getForm,[event.target.name]:event.target.value})

}

const emptyValidation =(value)=>{

if(value){

return true

}

else

{

return false;

}

}

const onSubmitHandler=(event)=>{

event.preventDefault();

if(!emptyValidation(getForm.FirstName))

{

alert("First name cannot be empty");

return;

}

if(!emptyValidation(getForm.LastName))

{

alert("Last name cannot be empty");

return;

}

if(!emptyValidation(getForm.Email))

{

alert("Email name cannot be empty");

return;

}

if(!emptyValidation(getForm.Password))

{

alert("Password name cannot be empty");

return;

}

navigate('/login');

}

return (<div>

<Header/>

<div class="container">

<div class="row">

<div class="col-4"></div>

<div class="col-4">

<h1>Sign Up</h1>

<form>

<div class="form-group">

<label>First Name</label>

<input type="text" onChange={onChangeHandler} class="form-control" name="FirstName"/>

</div>

<div class="form-group">

<label>Last Name</label>

<input type="text"onChange={onChangeHandler} class="form-control" name="LastName"/>

</div>

<div class="form-group">

<label>Email</label>

<input type="text" onChange={onChangeHandler} class="form-control" name="Email"/>

</div>

<div class="form-group">

<label>Password</label>

<input type="password" onChange={onChangeHandler} class="form-control" name="Password"/>

</div>

<button type="submit" onClick={onSubmitHandler} class="btn btn-primary">Submit</button>

</form>

</div>

<div class="col-4"></div>

</div>

</div>

</div>)

}

export default Register;

===========================================================================

Dummy backend set up with help of Nodejs - package to be installed (npm install -g json-server)

To Start JSON Server type (json-server --watch db.json) in cmd prompt inside backend

axios - communicate with API - promised based library

promise function in es6 -

npm install axios

import axios from 'axios';

promise - it has 3 states - pending , fulfilled and rejected

promise - 2 methods

then - used to handle successful fulfillment

catch - used to handle errors or rejections

backend will run on port 3000

frontend will run on port 3001

axios.post('http://localhost:3000/registration',getForm).then((result)=>{

console.log(result);

navigate('/login');

})

.catch((error)=>{

console.log(error);

})

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

ensure to use tilde symbol ` before ! mark on your keyboard for template string ${}

above src - class based component

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

\*Json-server

npm install -g json-server

json-server --watch db.json

\*Install JSON Server

npm install -g json-server

Create a db.json file with some data

{

"posts": [

{ "id": 1, "title": "json-server", "author": "typicode" }

],

"comments": [

{ "id": 1, "body": "some comment", "postId": 1 }

],

"profile": { "name": "typicode" }

}

Start JSON Server

json-server --watch db.json

Now if you go to http://localhost:3000/posts/1, you'll get

{ "id": 1, "title": "json-server", "author": "typicode" }

===========================================================================

-Remaining topics to be covered are redux , interceptor and lazy loading

Usecontext - UserContext.Provider - It has to be used in App.js .

UserContext.Consumer - wherever you want to consume the data (it could be any component)

{/\* <UseCallBack/> \*/} component based re render

<UseMemo/> - value based re render

const onChangeHandler=(event)=>{

if(event.target.name === "available"){

setform({...getform,[event.target.name]:event.target.checked})

}

else{

setform({...getform,[event.target.name]:event.target.value})

}

}

U have use event.target.value instead of name

===========================================================================

**[19-06-2023]**

**# Revision Overview REACT JS**

\*props - property - to pass data from one component to another component

\*useContext - we can pass state data to any of the component

\*provider and consumer - to improve performance

\*Redux - to handle complex application where there will be hundred of component will be accessing to state

1. view - User Interface component

2. Action - methods to update state

3. dispatcher - calling action/method and passing state value / to trigger action

4. store - data

Counterslice :

1. initiaize state

2. state name

3. reducers - multiple action

4. asyncthunk - to call API

5. extrareducers - fullfiled, pending, rejected