এই word file ব্যবহার করার নিয়ম:

টপিক কে .... এর মধ্যে রাখতে হবে। লেখা গুছিয়ে সুন্দর করে উপস্থাপন করতে হবে।

npx create-react-app my-app

npm start

তবে git থেকে clone করলে nodeModule আনার জন্য ঃ npm install

html link --> style.css

html script src -->src/index.js type=”text/jsx”

reactDom.render

redner একটি মাত্র element কে নিয়ে কাজ করতে পারে।

index.js শুধু app কে render করবে।

jsx and js expression : html ও css এর মধ্যে javascript ব্যবহার করতে চাইলে js এর ভেরিয়েভল এর নাম second breack এর মধ্যে লিখতে হবে।

সবধান: প্রতিটি fileName এর প্রথম word capital রাখতে হবে। নয়তো কোড কাজ করবে না।

inline css ব্যবহার :

<h1 style = {{”}}> toto app </h1>

const headingStyle = {

color: “red”, fontSize : “3rem

}

css কোড এর জন্য class = “ “ ব্যবহার করা হয়। কিন্তু react এ css কোড এর জন্য className=” “ ব্যবহার করা হয়।

className = “first-class-name second-class-name ....... “

public folder এ css ফাইল কে html file এর সাথে link করে দিলে তাকে সব জায়গা থেকে access করা যাবে।

src folder এ css ফাইল তৈরি করলে তাকে App থেকে access করার নিয়ম: import ‘./fileName’

কোন ফাংশন রিটান করলে তাকে component বলা যায়।

export and import theory.

export default Card.

import ImportName from ‘./location’

index.js --> App.js --> controll all components

Props and destructuring

<a herf = “ “ > </a>

<Card titleText = “Call Mother” />

titleText, herf এগুলো হলো attributes Names.র

Card component এ props এর মাধ্যমে value রিসিভ করবো।

const {titletext, descText} = props;

Mapping data to components

json file কে নিয়ে আমার এই ভিডিওতে কিছুটা question তৈরি হলো।  
Data নামে import এবং maping

Data --> map --> item

map এক লাইনে লিখলে { ও return দেওয়ার প্রয়োজন নেই।

তবে একাধিক লাইনে লিখলে { ও return দিতে হবে।

অনেকগুলো data --> map --> যতগুলো ডাটা আছে সবগুলোই আসবে একটা একটা করে ।

for loop ব্যবহার করলে key={x} দিলে সমাধান হয়ে যাবে কিন্তু ম্যাপে কি করতে হবে:

{ Data.map((item, index) => <Card key = {index} titleText={item.title} descText= {item.desc} />}

map করলে একটা key ={index}

mapping data with a unique id

npm install uuid

import {v4 as uuidv4} from ‘uuid’; ES6 এখন ES7 চলছে।

<div key = {uuidv4()}>

বর্তমানে require ব্যবহার করা হয়।

nested map ব্যবহার :

Data --> map --> map

es7

reafc.............................. অ্যারে ফাংশন

rcc..................................

rconst ....................... for class component

react developer tools chrome extension

How to use bootstrap use in react

class component state.

constractor--> initialState && use this every state.

disable={count ===0 ? true : false}

conditional rendering | ternary, short Circuit

{isLoggedIn && <HomePage />}

Event Handler | onChange |onClick

Event handler binding

ES6 এর ফাংশন ব্যবহার করলে binding করতে হয় না। তবে normal function তৈরি করলে তার সাথে binding করতে হবে।

React Hooks | useState hook

useState Import করতে হয় react থেকে।

const [count, setCount] = useState(0);

dynamic styling in React

const error = false;

return (

<div className = “App”>

<h1 style={{color : error ? “red” : “green” }} > wellcome </h1>

</div>

useEffect এর ডিপেনডেন্সি হলো কোন state এর value পরিবর্তন হলে এটা সাথে সাথে কাজ করবে।

className = {`${validInput ? “valid” : “invalid”}`}

Controlled Component | Collect form data

FORM --> FORM.js

form

label htmlFor

input

**Name** --- **Email--password**

**button**

**একটি Component এর js ফাইল এর জন্য css ফাইলকে import : import style from “./form.module.css”; ব্যবহার করার নিয়ম: className={style.formGroup}**

**onchange ব্যবহার করলে ফাংশনের parameter হিসাবে e নিতে হবে।**

**e.preventDefault();**

**পরবর্তী ভিডিও টিউটিয়ালে আমরা শিখবো কিভাবে state এ object নিয়ে কাজ করতে হয়।**

**useState with object**

**const [user, setUser] = useState({name: ‘ ‘, email: ‘ ‘, password: ‘ ‘})**

**const {name, email, password}= user;**

**setUser({name: e.targe.value, email, password});**

**এখন handleSubmit এ user কে প্রিন্ট করবো।**

**এই ভিডিও থেকে কি কি শিখলাম?**

**১. useState with object**

**২. প্রতিটি input tag এ value এর যেকোন পরিবর্তন একটি function এর মধ্যে যাবে।**

**৩. ‍spreed oparetor use && name এর উপর ভিত্তি করে কিভাবে input এর value নিতে হয়।**

|  |  |  |
| --- | --- | --- |
|  |  | export default function USESTATE\_OBJECT() { |
|  |  | const [user, setUser] = useState({ userId: "", fullName: "" }); |
|  |  | const { userId, fullName } = user; |
|  |  | const handleChange = (e) => { |
|  |  | setUser({ ...user, [e.target.name]: e.target.value }); |
|  |  | }; |
|  |  | return ( |
|  |  | <div> |
|  |  | <form> |
|  |  | <input |
|  |  | type="text" |
|  |  | name="userId" |
|  |  | value={userId} |
|  |  | onChange={handleChange} |
|  |  | /> |
|  |  | <input |
|  |  | type="text" |
|  |  | name="fullName" |
|  |  | // value={user.fullName} |
|  |  | value={fullName} |
|  |  | onChange={handleChange} |
|  |  | /> |
|  |  | <h1> |
|  |  | User: {userId}, {fullName} |
|  |  | </h1> |
|  |  | </form> |
|  |  | </div> |
|  |  | ); |
|  |  | } |

**passing data from child to parent component**

**props এর মাধ্যমে যেই নামে data পাঠাবো সেই নামেই রিসিভ বা ডিসট্রাকচার করতে হবে।**

**child to parent data pass**

**const handleChildData = (childData) => {**

**}**

**<Child onChildData={ handleChildData } />**

**card component code**

**const data = “I am from child Data”;**

**props.onChildData(data);**

***এই ভিডিও থেকে আমারা কি শিখতে পারলাম:***

***কিভাবে props এর মাধ্যে methode pass করে child থেকে data parent এ আনতে হয়।***

***onযুক্ত props name ব্যবহার করা ভাল।***

***create a basic todo app:***

app.js--> HOme.js(todos=[“item1”, “item2”]) -> todos -> todo

components -> NewTodo

toggle.js

<p>

show and hide

useEffect

import করতে হবে এবং একটি অ্যানোনিমাস ফাংশন লাগবে।

প্রতিবার যখন রেন্ডারিং হবে তখন useEffect প্রতি বার কাজ করবে।

তবে [ ] ডিপেনডেন্সি সেট করলে সেই ‍state এর value পরিবর্তন হলেই কেবল useEffect কাজ করবে।

তবে একাধিক ডিপেনডেন্সি সেট করার জন্য (,)কমা ব্যবহার করা হয়।

react এর জন্য html tag এর মধ্যে onযুক্ত function লিখার নিয়ম:

<button onClick={handleClick}> + </button>

setCount((count) => count +1);

fetch data

fetch কে বলবো তোর ‍res কে json formate এ আমার সামনে পেশ কর। এই res এর মধ্যে থাকা data গুলোকে Todos এর মধ্যে পাঠিয়ে দে।

fetch(“url”)

.then(res) => {

return res.json()

})

.then(data) => {

setTodos(data);

})

setTimeOut function এর মধ্যে যে কোড থাকে সেগুলো background এ run হয়। নির্ধারিত সময় শেষ হলে তা front এ চলে আসে।

এখানে নির্ধারিত সময় এর মধ্যে data লোড হতে না পারলে কি হবে?

front-end এ error আসবে কি/ নাকি সময় নিয়ে তার পর প্রিন্ট হবে।

.catch((error) => {  
setError(error.message)

});

error কিভাবে আসবে। throw করলে।

how to create a custom hook

diff diff api এর জন্য hook কে ব্যবহার করা।

hook এর নামের ফাস্ট লেটার small ব্যবহার করলে সমস্যা হবে কি? আনিছুর স্যার ব্যবহার করেছে।

DataFetch ---> useFetch

useFetch এ আমারা url দিবো কিন্তু আমার useFetch থেকে কি কি পাব?

রিটান করলে আমরা যা যা রিটান করা হবে সবকিছু DataFetch এ ব্যবহার করতে পারবো।

useEfftect : component এর প্যারামিটার হিসাবে url দিতে হবে।

এখানে

১. const [data, setData] =useFtech(“https:..............................”);

১.১. url pass করা হলো।

২. পরে useEfftect থেকে object আকারে

return { data, isLoading, error }; রিটান করা হয়। তাই DataFetch এ

 const { data, isLoading, error } = useFetch(

    "https://jsonplaceholder.typicode.com/todos"

  );

রিসিভ করা হয়।

৩. ডিপেনডেন্সি হিসাবে এখানে url ব্যবহার করা যায়।

৪. hook এর সাহায্যে এভাবে data pass করার জন্য first letter small case হয়।

ধন্যবাদ custom hooks কে।

fragment

<>

life cycle of a class component

Mounting --> constructor --> render --> componentDidMount

updating --> state/props – ‍shouldcomponentUpdate --> render {শুধু render call হবে}

ref in a class component with example.

ref directed dom কে ম্যনিপুলেট করতে সাহায্য করে।

ref তৈরি করা:constructor এর মধ্যে  this.userNameRef = createRef();

এবার লিংক করে দিতে হবে।

<input type="text" id="userName" ref={this.userNameRef} />

console.log(this.userNameRef.current.value);

    this.userNameRef.current.style.color = "green";

**React Hooks | useRef Hook**

**react থেকে import করতে হবে।**

ref ব্যবহার করা হয় class component এর জন্য । কিন্তু useRef ব্যবহার করা হয় function component এর জন্য। ধন্যবাদ।

useref ব্যবহার করে dom কে access ও modify করতে পারি।

তৈরি করা: const userNameRef = useRef();

<input type="text" id="userName" ref={userNameRef} />

এখানে ref attribute হিসাবে কাজ করবে।

useRef দিয়ে শুধু মাত্র dom এর value খুঁজে নিবো। এটা দিয়ে dom কে ম্যনিপুলেট করবো না। তাহলে প্রোসেস slow হয়ে যাবে। এটাকে control করা হয় না।

const handleSubmit = (event) => {

    event.preventDefault();

    const userName = userNameRef.current.value;

    const password = passwordRef.current.value;

    userNameRef.current.style.color = "red";

    console.log({ userName, password });

  };

useReducer

index.js ফাইল এর সুবিধাঃ একটি পাথ কম হবে। index.js পাথ না দিলেও react auto নিয়ে নিবে। স্যার পরে অন্য ফাইল এর চেন্স করলেন তাতে কোন সমস্যা নাই। UseReducer

books, modalText, isModalOpen তিনটি স্টেট তৈরি না করে একটি মাত্র reducer দিয়ে তিনটির কাজ করতে পারি। আরে বেশি থাকলে তাও করতে পারি।

প্রথমে ‍state ব্যবহার করে কাজটি করবো। তারপর useReducer

booksData

books -> id, name

books --> book --> name

bookName

form get book name

map এর রিটার্ন এর পূর্বে distracture করা যায়।

handleSubmit\*\*\*

setBookName এর মাধ্যমে BookName এর মধ্যে value দিয়ে দিছি। এখন এই value কারো প্রয়োজন হলে সে BookName থেকে নিবে।

setBookName এর মধ্যে অ্যানোনিমাস ফাংশন ব্যবহার করলে একলাইনে করলে return ও { দিতে হবেনা একাধিক লাইন ব্যবহার করলে দিতে হবে।

const handleSubmit = (e) => {

e.preventDefault ();

setBooks((prevState) => {

const newBook = {id: newDate().getTime().toString(), name: bookName};

return [...prevState, newBook];

});

setIsModalOpen(true);

setModalText(“book is added”)

}:

পূর্বের value গুলো ‍spread operator এর মধ্যে থাকবে এবং নতুন কোন value আসতে ‍singe package হয়ে যুক্ত হবে।

প্রশ্নঃ কোন value গুলো spread operator এ থাকবে?

উত্তরঃ যেহেতু setBooks এ মধ্যে ব্যবহার করেছে যা books এর অংশ তাই books এর Data গুলোই থাকবে।

প্রশ্নঃ prevState ছাড়া অন্য কোন কিছু ব্যবহার করলে কি কাজ করতো?

এখানে প্রশ্ন হলোঃ ‍setBooks এ একাধিক লাইন ব্যবহার করা হয়েছে কিন্তু এখানে দ্বিতীয় লাইন return করা হয় নাই কেন?

উত্তর: এটা শুধু একটা ভেরিয়েভল।

প্রশ্নঃ ফাংশন এর ভেতরে ভেরিয়েভল ডিক্লেয়ার করলে তা কে কে ব্যবহার করতে পারবে?

উত্তরঃ শুধু ফাংশন ই তাকে ব্যবহার করতে পারবে। অন্য কোথায় ব্যবহার করতে চাইলে সম্ভবত ভেরিয়েভল কে return করে দিলেই হবে।

js এ ভেরিয়েভল কে return করলে তা ফাংশনে চলে যেতে। কিভাবে ?

useReducer

import form react

const [booksState, dispatch] = useReducer(reducer, {

booksরিলেটেড state নিয়ে কাজ করবে এবং action কে dispatch করবে।

তিনটি state কে একটা ‍state এর মধ্যে নিয়ে আসা হয়েছে।

books: booksData,

isModalOpen: false,

modalText: “ “

})

useReducer একটি reducer ফাংশন তৈরি করে এবং ‍state গুলোর inisital value নেয়।

const reducer = (state, action) => {

বর্তমানে initial যেই state গুলো আছে তা reducer এর এই state এ চলে আসছে ।

return state;

}

অবর্শই কোন না কোন কিছু রিটান করতে হবে। যেমনঃ ‍state.

রেডিউছার ফাংশন দুইটি প্যারামিটার গ্রহন করবে present state and action.

যখন কোন action dispatch হবে তখন reducer তাকে নিয়ে কাজ করবে।

<button

              onClick={() => {

                removeBook(id);

              }}

            >

              Remove

            </button>

অ্যানোনিমাস ফাংশন হিসাবে এর মধ্যে একটা ফাংশনকে call করা হয়েছে এবং তাতে একটা id parameter হিসাবে pass করা হয়েছে।

   const filteredBooks = [...state.books].filter(

        (book) => book.id !== action.payload

      );

সেগুলোই থাকবে যাদের id ম্যাচ করবে না।

if (action.type === "ADD") {

      const allBooks = [...state.books, action.payload];

      return {

        ...state,

        books: allBooks,

        isModalOpen: true,

        modalText: "book is added",

      };

    }

প্রতিবার state কে রিটান করে দিতে হবে।

Prop drilling

com1--> com2--> com3 --> com4

এখানে com2 && com3 তে কোন ধরকার নাই com4 এ দরকার।

props এর মাধ্যমে কোন টেক্স পাঠালে propsName= “”

object পাঠালে propsName= { }

PropsName হলো যেই নামে পাঠাতে চাচ্ছি।

useContext Hook

parente এ useContext import করে ঃ import { UserContext } from "./UserContext";

তৈরি করা ঃ export const UserContext = React.createContext();

তৈরি করে UserContext ভেরিয়েভলে রেখে দিলাম।

|  |
| --- |
| step1: create context |
| // step2: from parent wrap childs with context provider |
| // setp3: state access useContext hook |

<UserContext.Provider value={{ user, text }}>

<Component2 />

</UserContext.Provider>

parente থেকে provide করলে সকল child ব্যবহার করতে পারবে।

value = { } object pass করতে হলে {এর মধ্যে অবজেক্ট গুলো দিতে হবে একাধিক হলে , ব্যবহার করতে হবে}

যেহেতু object তাই

|  |
| --- |
|  |
| import React, { useContext } from "react"; | |
| import { UserContext } from "./UserContext"; | |
| const Component4 = () => { | |
| const { user, text } = useContext(UserContext); | |
| return ( | |
| <div> | |
| <h2>{text}</h2> | |
| <h3>{user.id}</h3> | |
| <h3>{user.name}</h3> | |
| </div> | |
| ); | |
| }; | |
| export default Component4; | |

useContext কে এবং যে UserContext কে import করতে হবে।