Static routing (routes + route)

Nested and share routing = parent + child

Outlet = to share child data to parent (given to parent only)

Lazy routing:- to slowdown and show message when files is loding ..(when had too much files or routes) = const About = lazy(()=>import("./About")) =

 path : "/about",

        element : <>

        <Suspense fallback={<h1>Loding............</h1>}>

            <About/>

        </Suspense>

        </>

Main routing :- (createbrowserrouter, reactprovider)

1.class component = not use much now to do old project

Components means parts of the website

2.functional component

1.class component = use ( this )

01 : Class component Props :- small propertys of component

When you had 1 class into another and you want to send data to that class without destructuring original class …

{this.props.name}

02 : class component State :- An Object which you change on runtime…

For this use state(object) in Constructer Function with super metheod .

Ex:- constructor(){

        super()

        this.state = {

            num : 0

        }

    }

And after use setState for changes:-

Ex increment = () =>{

        this.setState({

            num : this.state.num + 1

        })

    }

State lifecycle:-

Mountain updating unmounting

1. constructor fun 1 New props() , setstate()
2. render 2 render
3. react dom 3 react dom
4. componant did mount 4 componant did update 1 componant will unmount

03 : Condition Rendering:- when you provide condition direct into render() function.

1 if else

2 let :- memory allocation

3 ternory opretor : only had two statement

4 && :- only show one value

04 : controlled and uncontrolled componanat :- to handle a form ( to out a value from input, select, textarea)

Controlled is handled by react & uncontrolled is handled by dom so in most case use Controlled.

1 uncontrolled : - to using ref{} method and createref() function.

2 Controlled :- to using state,value,and event(like onchanage)

05 : list and keys :- to give a list or li a sibling tags to a uniq identity or key .

Using map mathed to make a list and provide key by tostring() method.

Ex. const items = this.state.list.map((e)=><li key={e.toString()}>{e}</li>)

06.classcompotionandinheritance :-

Inheritance :- to load a child into parent and than you send a data from parent to child.

Classcomposition:- to load a child into parent and than you load child children to somewhere else that called …

07.HOC – higher order component – a patten where a function take a componanat as an argument and return a new component .

Ex :- const ironman = withsuit(tony stark)

(simply –to use same functionality create a one component with all functionality that we give that component to others.)

2.functional component

Hooks :- hooks are allow to use all react function without write a class. (

react 16.8 ,

don’t work in class,

call at top and only in react function,

don’t call inside loops,condition,nested function.

Every hooks name start with “use”

)

1 hook is = useState = allow to use state in functional component.

2 hook is = useEffect = allow to use react lifecycle(componantdidmount,didupdate,willunmount) combined

3 hook is =useLayouteffect = just like useEffect but useEffect call when mounting,unmountiog and useLayout is already call without react lifecycle although also add dependency”[]” in both is like where you want to. (ex. Doctor has app where his patients data and appointment is already load to apps before opening the app.)

4 hook is = useMemo = return a memorized value. ( simplay wrapper for function to use like function call when we call it otherwise not like keys in class compo listsandkeys.)

5 hook is = useCallback = same as useMemo but useMemo don’t work when you load a child give function as props there use useCallback instead of useMemo.

6 hook is = useRef = to hendal input (in form).

Api = application programming interface.

* Work as a Mediator (ex.zometo)
* What to do not how to do.
* With fatch method

7 hooks is = useContext = to send data to all child.(creater,provider,consumer)

8 hook is = useNavigate = to navigate after click button

9 hook is = useReducer = reduse funcnality ( all function into one function)(state manegment)

Redux = useReducer + useContext

Q&A

1) what is hooks :- hooks are allow to use all react function without writing a class.

2)what is functional component :- Functional component is just a simple javascript function; it accepts the data in the form of props and returns the react element. Whereas the class component will be created using the class keyword, and it extends the React.

3) difference between useeffect and usememo :- allow to use react lifecycle(componantdidmount,didupdate,willunmount) combined while useMemo return a memorized value.

4) what is render() :- render is a method in class componanat witch allow to read props and state and return our JSX code to the root component of our app.

5) describe state :- state is a object witch we can change on the runtime. State can be set using the **setState()** method.

6) what is api :- application programming interface.

* Work as a Mediator (ex.zometo)
* What to do not how to do.
* With fatch method

7) describes usereducer with example :- reduse funcnality ( all function into one function)(state manegment)

8) write a query solution of use context :- To use use context, you first need to create a context object. This can be done by calling the createContext function with an optional default value. Once you have created the context object, you can then use the Provider component to wrap your component tree. The Provider component takes the context object as a prop, and it will pass that value to all of its children.

9) what is components in react :- Components means parts of the website

10) write a code of use ref :-

import React, { useRef } from "react";

const Useref = () => {

    const inputvalue = useRef(null)

    const showinpute = () => {

        // console.log("showinpute");

        console.log(inputvalue.current.value);

        // inputvalue.current.value="100"

        inputvalue.current.focus()

        inputvalue.current.style.color = "green"

    }

    return (

        <>

        <input type="text" ref={inputvalue}/>

        <button onClick={showinpute}>upadte</button>

        </>

    );

}

export default Useref;

11) use layout :- = just like useEffect but useEffect call when mounting,unmountiog and useLayout is already call without react lifecycle although also add dependency”[]” in both is like where you want to. (ex. Doctor has app where his patients data and appointment is already load to apps before opening the app.)

12) what is function describe ani 4 function of js :- function is a set of statements that performs a task or calculates a value

13) what is spa and describe :- Single page applications, often called SPAs, are becoming increasingly popular among web developers. React is a JavaScript library that allows for the creation of complex and dynamic user interfaces – making it an ideal choice for single page applications.

14) what is use callback :- same as useMemo but useMemo don’t work when you load a child give function as props there use useCallback instead of useMemo.

**1. What is React Js?** React.js, or simply React, is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies. React allows developers to create reusable UI components and build complex user interfaces by composing these components.

**2. What is NPM in React Js?** NPM (Node Package Manager) is a package manager for JavaScript, and it is commonly used in React.js development. NPM allows you to install and manage packages (libraries, tools, etc.) that can be used in your React projects. It simplifies the process of handling project dependencies.

**3. What is the Role of Node Js in React Js?** Node.js is not required to build React applications, but it is often used in the React development environment. Node.js provides a JavaScript runtime environment outside the browser, and it's used for various development tasks like setting up a development server, running build scripts, and managing dependencies using NPM.

**4. What is CLI command In React Js?** CLI stands for Command Line Interface. In React.js development, the most common CLI is **create-react-app**, which is a tool that sets up a new React project with a default configuration. To create a new React app, you would use a command like:

bashCopy code

npx create-react-app my-react-app

**5. What are Components in React Js?** Components are the building blocks of a React application. They are reusable, self-contained pieces of code that define a specific part of a user interface. React applications are typically composed of multiple components, each responsible for rendering a part of the UI.

**6. What are Header and Content Components in React Js?** Header and Content components are examples of specific components in a React application. A Header component might contain the navigation bar or the top section of a webpage, while a Content component could include the main content area. These are conceptual divisions to organize your UI components.

**7. How to install React Js on Windows, Linux Operating System? How to Install NPM and How to check the version of NPM?** To install React on Windows or Linux, you can use the **create-react-app** tool. First, ensure that you have Node.js and NPM installed. Then run the following commands in your terminal or command prompt:

bashCopy code

# Install create-react-app globally npm install -g create-react-app # Create a new React app npx create-react-app my-react-app # Navigate to the app's directory cd my-react-app # Start the development server npm start

To check the version of NPM, you can use:

bashCopy code

npm -v

**8. How to check the version of React Js?** To check the version of React in your project, you can look in the **package.json** file in your project's root directory. The React version is specified in the **dependencies** or **devDependencies** section.

**9. How to make changes in components of React Js?** To make changes to React components, you edit the corresponding JavaScript or JSX files. Components can have state and props that determine how they render. You modify the component's code, save the file, and the React development server will automatically update the application with your changes.

**10. How to Create a List View in React Js?** To create a list view in React, you can use the **map** function to iterate over an array and render a component for each item. For example:

jsxCopy code

const MyList = () => { const items = ["Item 1", "Item 2", "Item 3"]; return ( <ul> {items.map((item, index) => ( <li key={index}>{item}</li> ))} </ul> ); };

**11. Create Increment-Decrement state change by button click?** To create an increment-decrement functionality in React, you can use state to keep track of a counter. Here's a simple example:

jsxCopy code

import React, { useState } from 'react'; const Counter = () => { const [count, setCount] = useState(0); const increment = () => { setCount(count + 1); }; const decrement = () => { setCount(count - 1); }; return ( <div> <p>Count: {count}</p> <button onClick={increment}>Increment</button> <button onClick={decrement}>Decrement</button> </div> ); }; export default Counter;

In this example, the **useState** hook is used to create a state variable **count**, and the **increment** and **decrement** functions modify the state based on button clicks. The current count is displayed, and the buttons trigger the state changes.