reactNOtes

1=> what is react js ? Ans=> *react is an open source js library.
*react is used for building user interface(ui).
*react simplifies the creation of spa by using reusable components.

2=> key features of react? Ans=> 1virtual dom 2component based architecture 3 reusability and composition 4jsx javascript xml 5 declarative syntax 6 community and ecosystem 7 react hooks

3=> what is virtual dom? what is difference between dom and virtual dom? Ans=> react uses a virtual dom to efficiently update the ui without rerender the entire page which helps improve performance and make the application more responsive

difference between dom and virtual dom

dom 1 dom is the actual representation of the webpages. 2 dom rerender the entire page when updates occur 3 dom can be slower especially with frequent updates. 4 suitable for static website and simple application

virtual dom 1 vd is lightweight copy of the dom 2 vd rerenders only the changed parts efficiently. 3 vd optimized for faster rendering 4 vd ideal for dynamic and complex spa with frequent updates.

4=> what are components? what are the main element of it? Ans=> in react a components is a reusable building block for creating user interface.

5=> what is SPA(Single page application)? Ans=>

- spa is a web application that have only one single web page
- in spa whenever user do some action on the website, then response content is dynamically updated witout refreshing or loading a new page.

6=> what is the advantages of react? Ans=> 1 simple to build single page application by using components. 2 react is cross platform and open source 3light weight and very fast because of virtual dom 4 it is supported by large community and ecosystem 5 testing is easy

7=> what are the disadvantages of react? Ans=> react is not good choice for very small application

8=> what is the roll of jsx in react? Ans=> *jsx stand for js xml. *jsx is used by react to write html like code. *jsx is converted to js via tool like babel because browser understand js not jsx.

9=> what is difference between declarative and imperative syntax? Ans=> *Declarative syntax focuses on describing the desired result without specifying the step by step process

- jsx in react is used to write declarative syntax

function App(){ return

SHubh

}

- imperative syntax involves step by step process to achieve a particular goal
- js has an imperative syntax

function App(){ const element=document.createElement("h1"); element.textContent="interview happy"; document.body.appendChild(element) }

10=> what are arrow function expression in jsx? Ans=> the arrow function expression syntax is a concise way of defining functions.

not in interview 11=> how to setup react project? 1 install node.js 2 install vs code 3 in vs code terminal type cmd "npx create-react-app my-app" 4 click file open go to just created folder cd my-app 5 run cmd "npm start"

11=> how react app load and display the components in browser? Ans=> Index.js = single page which loads index.js by react library index.js=replace root element of index.html file by app components app.js= root component which is the container of all the child components childcomponents= custom child components placed over app components

12=> what is defference between react and angular? r and u both are used to create single page ui application using components. react *r is a js library *r uses a virtual dom which makes it faster. *react js smaller in size and lightweight and therefore faster sometime. *react depends on external library for many complex features so doveloper has to write many lines of code for complex functionality *react is simple to leart and more popular than angular

```
Angular
*a is a complete framework
*angular uses a real dom
*a is bigger because it is a complete framework
```

```
 *since angular is a complete framework  therefore it provides built in support for features like routing forms validat
 *angular is slighty difficult to learn as it has typescript oops concept and many more thing.
```

13=> what are others 5 js frameworks other than react? angular vue.js backbone.js ember.js

14=> whether react is a framework or a library? what is difference? Ans=> Library *developers import the libraries at the top and then used its functions in components *react is commonly reffered to as a js library

```
Framework
*developers need to follow a specific structure or patters difined by the framework
*angular is a framework
```

15=> how react provies reuability and composition? Ans=> *react provides reusability and composition through its component based architecture. *Reusability:=> once you create a componnts you can re-use it in different parts of your application or even in multiple project. *Composition:=> composition is creating new and big components by combining existing small components. its advantage is change to one small components will not impact other components.

16=> what are state steteless stateful and state management? Ans=> *state refers to the current data of the components.
*stateful or statement means when user perform some action on the ui then the react application should be able to update and rereder that data or state on the ui.

17=> what are props in jsx? Ans=> props(properties) are a way to pass data from a parent component to child component.

18=> what is npm> what is the role of node_module folder? Ans=> npm(node package manager) is used to manage the dependencies for your react project including the react library itself.

node_modules folder contain all the dependencies of the project, including the react libraries.

19=> what is the role of public folder in react? Ans=> public folder contains static assets that are served directly to the user's browser such as images fonts and the index.html/

20=> what is the role of src folder in react? Ans=> src folder is used to store all the source code of the application which is then responsible for dynamic changes in your web application.

21=> what is the role of index.js page in react? Ans=> *index.html file is the main html file and spa in react application. *here the div with id root will be replaced by the components inside index.js file.

22=> what is the role of index.js file and react dom in react? Ans=> *react dom is a js library that renders components to the dom or browser. *index.js file is the js file that replaces the root elements of the index.html file with the newly rendered components/

23=> what is the role of App.js file in react? Ans=> App.js contains the root components of react application. App.js components is like a container for other components App.js define the structure layout and routing in the application

24=> what is role of function and return inside App.js? Ans=> *the function keyword is used to define a js function that represent your react components. *function is like a placeholder which contain all the code or logic of components. *the function takes in props as its arguments (if needed) and return jsx.

return is used to return the elements from the function

25=> can we have a function without a retrn inside App.js? Ans=>

- yes a function without a return statement is possible. *in that case your components will not render anything in ui. *the common use case for logging purpose.

26=> what is the role of export default insde App.js? Ans=> export statement is used to make a components available for importing using "import" statement in other files.

27=> does the file name and the components name must be same in react? Ans=> *no the file name and the components name dont have to be same *however it is recommended to keep them same for easier to organize and understand your code.

28=> what is role of jsx in react? Ans=> *jsx stands for javascript XML. *jsx is used by react to write html like code. *jsx is converted to js via tool like babel because browser does not understand js not jsx.

29=> what are the advantages of jsx? Ans=> *improve code readability and writability. *error checking in advance(type safety) *support js expression. *improve performance
*code reusability

30=> what is babel? Ans=> babel in react is used to transpile jsx syntax into regular js which browser can understand.

31=> what is roll of fragment in jsx? Ans=>

- in react a fragement is a way to group multiple children elements. *using a fragement prevent the addition of unnessary nodes to the dom. <></>,

32=> what is spread operator in jsx? Ans=> the so is used to expand or spread an array or object.

33=> what are the types of conditional rendering in jsx? Ans=> 1 if/else stm. 2 ternary operator 3&& operatot 4switch statement

34=> how do iterate over a list in jsx? what is map() method? Ans=> map() method allows you to iterate over an array and modify its element using callback function

35=> can a browser read jsx fiie? Ans=> *NO browser cannot directly understand jsx file. *babel takes jsx and converts it into equivalent js code that browser can understand

36=> what is transpiler? what is difference between compiler and transpiler? ANS=>

- a transpiler is a tool that converts source from one high level programming language(jsx) to another high level programming language (js) ex babel

*a compiler is a tool that converts high level programming language(java) into a lower level language(machine code or bytecode)

37=> is it possible to use jsx without react? Ans=> *yes its possible to use jsx without react by creating your own transpiler like babel *however this is not recommended since jsx is tightly integrated with react and relies on many specific features.

38=> what are react components? what are the main elements if it? Ans=> *in react a component is a reusable building block for creating user iterfaces. element of components 1 import the react library 2define a functional component 3return jsx to describe the component's ui 4 export the component to make it avaibable for use in other

39=> what are the type of react components? what are the functional components? Ans=> functional components class components *functional components are declared as a js function. *they are stateless component but with the help of hooks they can now manage state also.

40=> how do you pass data between functional components in react? Ans=> props(properties) are a way to pass data from a parent components to a child components

41=> what are the prop drilling in react? Ans=> prop drilling is process of passing down props through multiple layers of components

42=> why to avoid prop drilling ? in how many ways can avoid prop drilling? Ans=> avoid reason 1Maintenance:=> prop drilling can make code harder to mantain as changes in data flow require updates across multiple components

2 complexity:=> it increases code complexity and reduces code readability

3:=> Debugging:=> bebugging becomes challenging when props need to be traced through numerous components.

5 way to avoid prop drilling 1 using context api 2using redux 3using components composition 4 using callback functions 5using custom hooks

42=> what are class components in react? Ans=> * class components are defined using js classes. * they are stateful components by using lifecycle method * the render method in a class components is responsible for returning jsx.

43=> how to pass data between class components? this. props can be used in child components to access property/data passed from parent components

45=> what are the role of this keyword in class components? Ans=> this keyword is used to refer to the instance of the class

46=> what are the 5 difference between functional and class components? Ans=> functional components 1 syntax difined as a js function 2 state originally stateless but can now maintain state using hooks 3 there are no lifecycle method 4 more readable and concise 5 do not have render method

class components 1 difined as a js(es6) class 2 can manage local state with this.state 3 there have lifecycle method 4more complex 5have render method

47=> what is routing and router in react? Ans=> Routing allows you to create single page web application with navigation without the need for a full page refresh

react router is a library for handling routing and enables navigation and rendering of differenct components based on the url

48=> how to implement routing in react? Ans=> install react router create navigation creates routes

49=> what are the roles of and component in react routing? Ans=> the components is used as the root container for declaring your collection of routes.

the components is used to define a route and specify the component that should render when the route matches for example in this code if user enter "www.sh.com/about" in url then matching about components will be rendered

50=> what are route parameters in react routing?

Ans=> route parameters in react router are a way to pass dynamic values(data) to the components as part of the url path.

51=> what is the role of switch components in react routing? ans=> switch components ensure that only the first matching is rendered and rest are

for example switch is commonly used to handle 404 or not found routes

52=> what is the role of exact prop in react routing? Ans=> exact prop is used with the components to match exactly to the provided path.

53=> what are react hooks > what are the top react hooks? Ans=> *react hooks are inbuilt functions provided by react that allows functional components to use state and lifecycle features *before hooks class components lifecycle method were used to maintain state in react application *to use react hook first we have to import it from react library

useState(): Manages state in functional components.

useEffect(): Handles side effects, like data fetching or subscriptions.

useContext(): Accesses the value of a React context within a component.

useReducer(): Manages state logic with a reducer function.

const [state, dispatch] = useReducer(reducer, initialState); state: The current state value. dispatch: A function to dispatch actions to update the state. reducer: A function that specifies how the state should be updated based on the action type. initialState: The initial state value.

useCallback(): Memoizes a callback function to prevent unnecessary re-renders.

useMemo(): Memoizes the result of a computation to avoid re-computing.

useRef(): Creates a mutable object that persists across renders, often used for accessing DOM element

54=> what is role of useState() hook and how it works? Ans=> *the usestate hooks enables functional components to mange state *usestate working: usestate function accept the initial state value as the parameter and return an array with two elements *the first element is the current state value. *second element is the function that is used to update the state.

55=> what is role of useEffect? how it works and what is its use?

The useEffect() hook in React is used for handling side effects in functional components. It works by allowing you to perform operations, such as data fetching, subscriptions, or manual DOM manipulations, after the component has rendered.

2 imp points 1 useeffect is called after the components renders ex: side effect 1 useeffect function will accept two parameter (effect function,dependency array)

import React, { useState, useEffect } from 'react';

const ItemList = () => { const [items, setItems] = useState([]);

useEffect(() => { // Function to fetch data const fetchData = async () => { try { const response = await fetch('https://api.example.com/items'); const data = await response.json(); setItems(data); // Update state with fetched data } catch (error) { console.error('Error fetching data:', error); } };

```
fetchData(); // Call the fetch data function when the component mounts

return () => {
  // Cleanup code (optional)
  // This function runs when the component is unmounted
  console.log('Component will unmount, cleanup here if needed');
};
```

}, []); // Empty dependency array means this effect runs only once after the initial render

return (

Item List

{items.map(item => (
- {item.name}
))}

); };

export default ItemList;

56=> what is dependency array in useeffect hook? Ans= dependency arrays(optional) act as trigger for useeffect to rerun meaining if any of dependencies values change the code inside useeffect will be executed again.

57=> what is the meaning of the empty array in the useeffect? Ans=> the empty array indicates that the effect function should only run once

58=> what is the role of useContext hook? Ans=> use context in react provides a way to pass data from parent to child components with out using props.

59=> what is the createContext() method? what are provider and consumer properties? Ans=> *createContext() functions returns an object with provided and consumer property.

*the provider property is responsible for providing the context value to all its child components

*use context() method ir consumer property can be used to consume the context value in child components

59=> when to useContext() hooks instead of props in real application. Ans=> *use usecontext instead of props when you want to avoid prop drilling and access context values directly withiin deeply nested components. 1 theme switching 2 localization (language selection) 3centralize configration setting 4 notification system

60=> what are components life cycle phases? And=> Mounting Phase:

Definition: The phase when a component is created and inserted into the DOM. Key Methods: constructor, render, componentDidMount Updating Phase:

Definition: The phase when a component re-renders due to changes in state or props. Key Methods: shouldComponentUpdate, render, componentDidUpdate Unmounting Phase:

Definition: The phase when a component is removed from the DOM. Key Method: componentWillUnmount Error Handling:

Definition: The phase for catching and handling errors within a component tree. Key Method: componentDidCatch (introduced in React 16)

61=> what are component life cycle methods? Ans=> component lifecycle methods are special methods that get called at various stages of component life.

```
1Mounting Phase
constructor() getDerivedFromProps()  render()  componentDidMount()

2 updating phase
shouldComponentUpdate, render, componentDidUpdate
```

3 unmounting phase componentwillunmount()

62=> what are constructors in class components ? when to use them? Ans=> *constructor is a special method that is called when an instance of the class in created

*constructor is used for initializing the components state or performing any setup that is needed before the components is rendedred

63=> what is role of super keyword in constructor? Ans=> super keyword is used in the constructure of a class components to call the constructor of the parent class

this is necessary to ensure that the initialization logic of the parent class is executed

64=> what is role of render() method in components life cycle? Ans=> render() method returns the react elements that will be rendered to the dom

65=> how the state can be maintained in a class components? Ans=> this.setState() method is used to update the state this.state property is used to render the update state in dom.

66=> what are controlled components in react? Ans=> a controlled components is a component whose form elements(like input fields or checkboxes) are controlled by the state of the application

67=> what are the difference btw controlled and uncontrolled components? Ans=> Controlled components

- values are controlled by react state.
- event handlers update react state
- dont't depend on useRef()
- re-renders on state changes
- a recommended and standard practice for form handling in react

Uncontrolled components *values are not controlled by react state *no explicit state update values can be accessed directly from the dom *commonly uses useRef() to access form element values *less rerendering since values are not directly tied to react state *useful in certain scenarios but less commonly considered a best practice

68=> what are characteristics of controlled components? Ans=> 1 state control:= the value of the form elements is stored in the components state.
2 Event Handling:= changes to the form element trigger an event(onchange) 3state update:= the event handlers updates the components states with the new values of the form element 4 rerendering: the component are rerenders with the update state and the form elements reflects the values.

69=> what are advantages of using controlled components in react form? Ans=> in controlled components form elements have their values managed by react state, ensuring a single source of truth.

70=> how to handle forms in react? Ans=> the prefirred and recommended approach for handling forms in react is by using controlled components

71=> how can you handle multiple input fields in a controlled form? Ans=> maintain seprate state variables for each input field and update them individually using the onchange event

72=> how do you handle form validation in a controlled components? Ans=> by using conditional rendering based on the state and validate input values before updating the state.

73=> in what scenarios might using uncontrolled components be advantages? Ans=> uncontrolled components can be benificial when integrating with non react libraries or when dealing with forms where controlled components are not possible

74=> what is code splitting in react? Ans=> code splitting is a technique to split the js bundle into smaller chunks which are loded on demand

75=> how to implement code splitting in react? Ans=> 1 use React.lazy() import components 2 wrap components with suspence to handle loading 3 configure your build tool (eg webpack) for dynamic imports

76=> what are the pros and cons of code splitting? Ans=> pros of code splitting faster initial load time optimized bandwidth usage improved caching parallel loading easier maintenance

cons of code splitting complexity tooling dependencies potential for runting error increassed no of request

77=> what is higher order components in react? Ans=> a higher order components is component is a component which takes another components as an arguments and adds extra features to another components

78=> what are the ways to style react components ? Ans=> 1 inline styles 2 css stylesheets 3 css module 4 global stylesheets 5 css frameworks

79=> what are the difference between react and react native? Ans=> react *react is library
*react is used for building web interfaces
*run on web browser
*html and css are used for ui
*deployed as web application

react native *react native is frameworks *react native is used for building mobile application *runs on ios and android plateform *native ui components(eg view text) are used for ui *deployed through app stores(eg app store and google play)

80=> what are the top 3 ways to achieve state management ? when to use what in react? Ans=> 1 use state hooks when to use simple components level state

2 context api use prop drilling avoidance for shaing global data

3 redux large scale application with complex state

81=> how can you implement authentication in a react application? Ans=>

1 POST(username, password) 2 autenticate and create jwt token 3return response (jwt token) 4store jwt token at local storage 5request data 6 validate token signature 7send data 8 display data on browser

82=> what is react use of react profiler? Ans=> react profiler is a set of tools in react that allows developers to profile(anlyze) the performance of a react application

83=> what is different between fetch and axios for api call in react? Ans=> fetch *fetch is a built in js function so it doesn't require any additional libraries. *fetch return promises making it easy to work with asynchronous code using async await syntax *if you want to keep http request simple fetch is good choice ex fetch("https://api.com"). then((response) => response.json()) .then((data) => console.log(data)). catch((error) => console.log("error))

Axios *Axios is third party library that simplifies the process of making http request *Axios allows you to use interceptors which can be good for tasks like requests response logging , authentication and error handling *if you want to intercept http request response or improve error handling then azios has more features to do it ex installtion command npm install axios import axios from "axios" axios .get("https://api.com") .then((response) => console.log(response.data)) .catch((error) => console.log("error))

84=> what are the popular testing libraries for react? Ans=> jest,react testing libraries,enzyme,cupress

85=> how can you optimize performance in react application. Ans=> 1 optimizing render with react.fragement = use it to aviod unnecessary wrapper elements that could case additional dom nodes 3code splitting = employee code splitting to divide your application into smaller chunks that are loaded on demand improving initial load times
4 optimized images and assets = compress and optimized image use responsive images and leverage lazy loading for images to reduce network and rendering overhead

86=> explain reactive programming with example? Ans=> react programming is a programming approach that focus on reacting to changes and event in a declarative and asynchronous manner

87=> in how many ways can we implement reactive programming in react. 1 state and props= reacting to changes in local components state and passing data reactively through props 2react hooks= leverage usestate and useeffect hook for managing state and side effect in functional components 3 context api and redux

88=> parent provides a callbacks function to child and then child components can then invoke this callback to pass data back to the parent