**React Basics Hooks Question**

Different different hooks has their own uses case. It makes things easier.

**1. useState -** Call back Method

const [value, setValue] = useState(5);

Here, First time 5 will pass to callback setValue.

After few seconds it will pass to value state.

5 will not directly pass to value state.

**2a. componentDidMount**() {

console.log('Hello World');

}

**2b. useEffect**(() => {

console.log('Hello World');

}, [])

**3a. componentDidUpdate(prevProps)** {

console.log(`Hello World ${prevProps}`);

}

**3b. useEffect**(() => {

console.log('Hello World');

}, [prevProps])

**4a. componentWillUnmount**() {

console.log('Hello World');

}

**4b. useEffect**(() => {

console.log('Hello World');

return () => {

console.log('this is also know as clean up methods in react');

}

}, [])

**5 useContext** ----> This hooks works like as wrapper in the root component to pass the data thourgh all roots. you can imagine kind of basic redux

i. createContext()

ii. useContext()

import React from 'react';

export const UserContext = React.createContext();

export default function App() {

const [value, setValue] = React.useState(2)

return (

<UserContext.Provider value={value, setValue}>

<User />

<Profile/>

</UserContext.Provider>

)

}

function User() {

const {value} = React.useContext(UserContext);

return <h1>{value}</h1>;

}

**6. How to access array**

const users = [

{ name: "Rohan", id: 1 , age: 19},

{ name: "Jane Doe", id: 2 , age: 9},

{ name: "Billy Doe", id: 3 , age:25},

{ name: “Deepak”, id: 4 , age:10}

];

**\*Bad mistake :\*** users.name or users.age

**\*Good habbit :\*** user.method (use predefine method to get each index object property, eg, map, filter, find, ....etc)

like users.filter((user)=> user.age> 18)

**7**. **Write an example for HOC Function (Call back uses case)**

2. const addition = (num1, num2) => {

return num1 + num2

}

3. const subtraction = (num1, num2) => {

return num1 - num2

}

4. const mul = (num1, num2) => {

return num1 \* num2

}

5. const divide = (num1, num2) => {

return num1 / num2

}

// Parent function :- Here calculator method is HOC

1. const calculator = (num1, num2, sign)=> {

return sign(num1, num2)

}

console.log(calculator(12,3,divide))

**8. Write a program to show fibonacci series**

const findFibonacci = (number) => {

let arr = [];

for(var i=0;i<=number;i++){

i < 2 ? arr.push(i) : arr.push(arr[i-1] + arr[i-2])

}

return arr

}

console.log('Length :', findFibonacci(6))

**Theory + Practical questions**

-> Why we use ReactJS :

-> features of react

-> Make hands on both function and class base component

-> Reusable component

-> Virtual Dom (Performance Due to Quicker rendering)

-> React framework for mobile app

-> why is It easy-to-learn

-> React vs Angular diff

-> When hooks introduce by react

-> ReactJS is SEO friendly

**=>>Well prepare below topics**

**=>> Major things are included in this**

1. Basics

• Html

• CSS

• Java Script

• ES6

• OOPs concepts (basics mechanism :- Inheritance, encapsulation, polymorphism….etc)

• SDLC Method (eg, agile, waterfall model….etc)

• Make hands on Bootstrap (special :- flex, container, grid, row, column, card, list….etc)

• Understand JSON Data

2. Basics of React

• What is React

• Features of react

• Life Cycle of React and same for functional base hooks

• Block Scope, Let, Constant, var

• Operator, Expression, (eg, ternary)

• Arrow Function

• Destructing (array, object)

• Solve 10 question on array. **(very very important)**

• Solve 10 question on object. **(very very important)**

• Promises

• State variable

3. JavaScript

• try, catch

• map vs foreach difference

• filter, find, findIndex, reducer (all method usecas)

• axios vs fetch api call

Redux

4. React

• Features of React v16

• useState

• Component

i. Function

ii. Class

iii. Pure

iv. Component styling

• Hooks, Memo

• Props

i. Understand Props

ii. Read, Write and validate props

• Communication between Component

i. Pass value from parent to child

ii. Pass value from Child to parent

iii. DOM Reference

5. Routing

• Types of Routing in React(Browser Router, Hash Router )

• Configure and Define routing

• Nested routes

• Navigate using nav links

• Redirect

6. Redux

• What is redux and uses

• Action, Reducer , Store

7. Lists, Map, Sets

8. API Calls(Axios)

• GET,POST,PUT,DELETE Methods

• Using above methods practice API calls

9. Prepare Basic CRUD(Create, Read, Update, Delete) Operation Application

**Hands on coding practise, js question**

-> Wap(Write a program) to find even odd number

-> Wap to swap two numbers without using 3rd variable.

-> Find all even number from given array.

-> Find greatest number from array

-> Fibanacci serious from 0 - 20

-> WAP to check given string is Palindrome Using for Loop (eg, madam)

-> Factorial of a number.

-> Factorial of a number using recursion.

-> Wap to Print table of 1- 10

-> Giving you string. Make first letter capital in every word

( “how are you today? Where are you going” )

-> Reverse a string. With method()

const str1= "how is the day"

-> Reverse a string. without using method.

const str1= "how is the day"

-> Sort the array value without using any array method

let arr = [4, 32, 2, 5, 8];

-> Sort the array value by using method

let arr = [4, 32, 2, 5, 8];

-> Create a method, it will take array of integers from its parameter and it will return reverse of that array

const arr1 = [2,4,5,6,7,8]

-> WAP(Write a program) to find out a palindrome string....

-> WAP to find out a given number is palindrome or not.

-> How to calculate the number of vowels and consonants in a string..

-> How to calculate the number of numerical digits in a string

-> Find out the missing number from array that contains integers from 1 to 100

-> Find out the largest value from array

-> Find out the second largest value from array

-> Find out the largest value from array without using sort method

-> How to remove special characters in a string that is in lowercase