**JAVASCRIPT**

1. What are promises in JavaScript?

2. What is Asynchronous code?

3. Console questions

4. What is destructuring?

5. What is setTimeout and setInterval?

23. What is JavaScript and what is ES6?

24. What are the features introduced in ES6?

25. What is an arrow function?

26. What are the different ways to create a function in JavaScript?

27. Difference b/w let, var and const.

28. How to create an event in JavaScript? addEvenetListner use?

29. What is Promise?

30. What is Pyramid of DOM in JavaScript? in Promise.

31. What are the methods in Promise?

32. Promise.raise, Promise.any?

33. What is async await?

34. What is callback?

35. How to create an Object using a function?

36. const change in object?

37. How to add one more property to Object?

38. How to get the length/depth of the Object?

39. What is high order function?

40. Write a function for the promise method. async await and try catch promise completely?

3. 2 div with the same id, document,getElementById targets which div?

4. What do you understand by hoisting in JavaScript?

5. What is the basic difference b/w var and let keyword?

6. What are closures in JavaScript?

7. Explain JavaScript in OOPS with an example.

8. What are Prototypes in JavaScript?

9. What are high order functions in JavaScript.

10. What are callbacks in JavaScript?

11. What are promises

12. Why is the promise introduced as the same thing can be done with a callback?

13. There are 4 tasks, we have to do 1 by 1 how to do?

14. What are filters and maps in JS.

15. What is basic difference is map and reduce.

16. Filter the object with given city/name Pune.

17. Call by Property Tom and jerry JavaScript

18. Make a copy of an array.

19. What is spread operators.

20. By cloning a spread array A to array B, by changing the value of A array the value B array is also changes, how to overcome this problem and why is behave like that.

1. What is Map (allocate a memory) and WeakMap in JavaScript?

2. What is the use of Promises?

3. When do you use async/await or promise?

4. What is the use of mutation observor?

5. What is the use of yield?

6. What is generator function?

7. How yield works, ++ operator?

8. What is the best way to load the JavaScript files when there are multiple files and some import in the exist?

9. Best way to do import?

10. How lazy loading works?

1. What is IIFI?

2. What is hoisting in JavaScript?

3. What is function binding in JavaScript? What is bind? What is the behaviour of the bind?

4. What is this keyword in JavaScript?

5. Difference b/w methods and functions?

6. Difference b/w call, bind, apply?

7. What is the purpose of the inbuilt function?

8. function name person and variable name with value abc and I have a function get name and typing this.name and outof this person function we are creating an object of person name and then calling the method what is the output?

9. What is constructor?

10. What is the difference b/w the named function and arrow function?

11. What is temporal dead zone?

12. console b/w settimeout and promise timing?

16. What are closures in JavaScript?

17. What is scope chaining?

18. What is pure function?

19. What is the difference b/w var, let and const?

20. What are high order function? Explain with example?

21. What are array function? Explain them?

22. What is reduce function in Array?

23. What is promise?

24. What is the difference between promise and callback?

25. What are the disadvantages of callback?

26. What are different optimization use in JavaScript?

25. Fiter the array having the name with having skill will be JavaScript...

10. Combine both array in a single array, after that sort the array in ascending order of ID.

11. Filter the object which have student access is TRUE? Output will be array having student access is TRUE. without pre defined functions?

12. Mention git command use in git commit/fetch.

13. Find the index of object within the Array where the name of the book is four.

12. Create a one to do list from the text with having button to add and button to remove individually in React.

13. Add active class & style in JavaScript in react and JavaScript by select an list of element?

14. Initially in each data need to add one more key, once click this array item.

15. How to add new key to object?

16. What are data types in JavaScript?

17. What are inbuilt methods in Array?

18. What is regular expression in JavaScript?

19. What is closure in JavaScript? and write a example.

20. What is JavaScript hoisting?

21. What is the difference b/w var, let and const?

22. How to flatten nested object output?

23. How to flatten nested array output?

18. Write a program of revering the string.

11. What is closure?

12. What is == and ===?

15. What is typescript? Any advantage?

1. What is ES6 and its concept?

2. What is string interpolation?

3. Declare a var with number?

4. Prepare a snippet that count the each character of string how many times it is present in it.

8. What is hoisting?

9. What is closure?

10. What is recurssion function?

27. What is event loop.

28. What is a promise and how does a promise work.

29. What is the difference b/w Promise and Async await.

30. What is anonymous function.

31. 0.1 + 0.2 == 0.3 console = Because of the limited precision of floating point numbers round-off errors can occur during calculations.

32. What is data type of null. => Object

**REACT**

6. What is useEffect?

7. What is states and props?

8. What is fetch and axios?

9. What is redux?

10. What is useMemo?

11. Where we use React.Memo?

12. Given an api, and fetch title image and rating of the product?

21. What is useEffect in react.

22. What dependency is passed in useEffect.

23. If we pass an object to useEffect how does it behave.

24. Why we use useMemo in React.

25. What is difference b/w useEffect and useMemo.

26. How do you call API in React.

27. What are interceptors in React.

28. What is useContext in React.

29. In useContext we have variable a, b, c and all these are siblings to each other. after made a change in parent for c only, what will happen on these component, will it re-render.

30. What is useState. Why we can't use let or var instead of useState.

31. We have 1 parent component and inside 1 child called A and inside A we have another child B. Variables are passed from Parent to A and B. In parent we are changing values used in child B but not used in child A, what is behavior, will A re-render and B re-render or both will re-render. vice versa.

32. Difference b/w useContext and Redux Toolkit.

33. What is Redux.

11. What is shadow DOM and what is Virtual DOM?

12. What is tree shaking in React? How it will happen?

13. How do shadow DOM works?

14. What is encapsulations in React?

15. Which is better functional or class component? Why?

16. Is is possible to impplent redux thung or saga with the functional approach?

17. What is redux toolkit and the middleware and how it works?

18. How store works?

19. Where do wrap the store?

20. What are build tools we use?

21. When you want to migrate the images from local to stage than prod sever how do you configure the any webpack file?

22. What is context api in react?

23. Unit testing, jest?

24. How to check the element is present of the DOM using jest?

2. What is React? What are its features?

3. What is components in React? What are different types of components?

4. What is JSX? How does the browser reads it?

5. What is custom hooks in react?

6. What are the various vision of optimize react code?

7. What is difference b/w useRef, createRef in React?

8. What id Redux and where & why do we use in React?

9. Increment and Decrement of Text size on click in React?

6. What is the lazy loading in React?

13. What is virtual DOM in react? What is the purpose of creating?

14. What is difference b/w useState and Props?

15. What is the lifecycle hooks in React?

16. RXJS?

17. Execution Process of Redux?

24. What are hooks in React?

25. What is the difference b/w useState and useReducer?

26. For unit testing in react what we use and elaborate?

27. What is used for SSR?

28. What is prop drilling? What is disadvantage and how to fix?

29. Use filter then use arrow to return value greater than asked from object?

1. What are components in React and what are class components and functional components?

2. What is react hooks? Give an brief example?

3. What is useMemo? Explain with ann example.

4. What is context API and redux?

5. What is despatcher in redux?

6. What is prop drilling?

7. How to you do the testing in React?

8. What will be your best approach to fetch the data from API in react? Give an example.

9. In React component how it differs from another component?

10. What is skeleton loader?

1. What are HOC and functional architucture? Please elaborate and explain it with example.

2. Create a hoc component to resue the resuable table component

3. What is functional component why its uses over the class component?

4. What will be the benefits of using functional component dev vs prod?

4. Why we use Memomization in React?

5. What are the techniques uses in Memozization, what problems we face what are their principles?

6. If we are reutilising the value with memozization then why we don't use cache?

7. Simple break the array without using loop, reduce not using any array methods, print array in console 1 by 1.

10. In react what is pure component.

11. What is state-full and state-less component.

12. What is the difference b/w virtual DOM and shadow DOM.

13. What are react life cycle hooks, life cycle methods or class based and life cycle hooks or function based.

14. How do you implement mount, update and unmount with functional components.

15. What is useMemo in React.

16. What is React.Memo

17. What are error boundaries.

18. How do you implement error boundaries and what are predefined for that.

19. What are synthetic events is React JS.

20. How do you implemented authentication in React.

21. How do you validate user is logged in OR valid user or not.

22. How do you use private protector routes in react with router. Is this is a part of react route?

23. What is state management.

24. What are core principals/design principals on which Redux built up on.

25. What are the use case of Redux use.

26. Explain the flow of data through Redux.

0. SOLID principles

0. What is JSX?

1. What is the virtual DOM?

1. function vs class component

2. What are the different phases of the component lifecycle?

2. What are the lifecycle methods of React?

2. Explain React state and props

3. Hooks, Rules of Hooks

4. Custom Hook

4. Pure component

5. Controlled vs Uncontrolled component

5. What is prop drilling in React?

6. Context - Ask more on it

7. Error Boundaries

8. Virtual and Shadow DOM

9. Client side rendering and Server side rendering (CSR & SSR)

10. Higher-Order Components

11. Fragments

12. Error Boundaries

13. Babel

14. What is React Router?

15. How to perform automatic redirect after login?

14. Name a few techniques to optimize React app performance.

0. Lazy loading (Suspense and fallback)

1. Code splitting

How to improve react performance.

Jest tool

Code to remove duplicate items from array.

Web worker

Flux architecture

How to avoid re-rendering

Concept of redux

File structure for react

Component life cycle methods

CSS preprocessor

CSS frameworks

Difference between Typescript and JavaScript

SOLID principles of react

Client side rendering and Server side rendering (CSR & SSR)

Draw diagram for login and save account channel functionality ( Mostly how you will structure your component)

If login to dashboard, will we redirect to login screen if page is refreshed.

Debugging tools

One way data binding

Code coverage tools

What is the process that you will follow if you assign a new task and you don’t have any knowledge regarding that?

Lazy loading (Suspense and fallback)

Code splitting

How you will identify which component is taking more time to load?

Virtual and Shadow DOM

ES6 features

Coding standard in react

Context API

How to pass data between components

Why to use keys when creating list of components

differences between class component and functional components in react

what state management we used in react and explain why we use them

differences between old react (before 16.8) and new react version (with hook)

two most used lifecycle method in hooks

explain useEffect

what are the use of arguments for ueseEffect, how they work?

explain useContext

differences between SSR and CSR

how to optimize SEO in CSR

explain promises in javascript

explain Async Await

explain some es6 changes

differences between merge and assign function

explain differences of flexbox and grid

how to overlap div on top of each other

how to center a div horizontally and vertically using flex properties

ability to use webflow

explain parallex effect

explain context api in react

Redux:

1. Redux workflow

2. Hooks in a React Redux App - useSelector, useDispatch

3. connect() - The connect() function connects a React component to a Redux store.

4. connect() Parameters

mapStateToProps?: (state, ownProps?) => Object

If a mapStateToProps function is specified, the new wrapper component will subscribe to Redux store updates. This means that any time the store is updated, mapStateToProps will be called.

mapDispatchToProps?: Object | (dispatch, ownProps?) => Object

If your mapDispatchToProps is declared as a function taking one parameter, it will be given the dispatch of your store.

5. middlewares

6. combineReducers

1. Explain the building blocks of React?

The five main building blocks of React are:

• Components: These are reusable blocks of code that return HTML.

• JSX: It stands for JavaScript and XML and allows you to write HTML in React.

• Props and State: props are like function parameters and State is similar to variables.

• Context: This allows data to be passed through components as props in a hierarchy.

• Virtual DOM: It is a lightweight copy of the actual DOM which makes DOM manipulation easier.

What is the use of ref in React?

Explain the core components of react-redux?

There are four fundamental concepts of redux in react which decide how the data will flow through components

• Redux Store: It is an object that holds the application state

• Acrtion Creators: These are unctions that return actions (objects)

• Actions: Actions are simple objects which conventionally have two properties- type and payload

• Reducers: Reducers are pure functions that update the state of the application in response to actions

What is context API?

How to pass state data from one component to another component?

• Parent to Child (via props)

Child to Parent (via callback functions)

Sibling to Sibling

Using Context API

Using State Management Libraries

What are hooks in React?

What is custom hooks in React?

What is the Prop drilling in reactjs?

1. How to create components in ReactJS?

2. What are the advantages of ReactJS?

3. Differentiate between real DOM and virtual DOM?

4. What are forms in ReactJS?

5. How is React different from React Native?

6- What is JSX?

7-Explain React state and props.

7- What is thunk & Saga ?

1. How to improve react performance.

2. Jest tool – unit test cases (If function having third party dependency how you write unit test cases for this. Will you use mock/spi)

3. Code to remove duplicate items from array.

4. Web worker

5. Flux architecture

6. How to avoid re-rendering

7. Concept of redux

8. File structure for react

9. Explain life cycle methods of React components

10. What are Higher Order Components(HOC)

11. CSS preprocessor

12. CSS frameworks

13. Difference between Typescript and JavaScript

14. SOLID principles of react

15. Client side rendering and Server side rendering (CSR & SSR)

16. Debugging tools

17. One way data binding

18. Code coverage tools

19. Lazy loading (Suspense and fallback)

20. Code splitting

21. How you will identify which component is taking more time to load?

22. Virtual and Shadow DOM

23. ES6 features

24. Coding standard in react

25. Context API

26. How to pass data between components

27. Why to use keys when creating list of components

28. Shallow copy vs Deep copy

HTML

1. What is the use case of HTML? Why we need it?

2. What is the use of HTML tag and HTML attribute and difference b/w them and what actually they do or what does it mean.

3. What is void element in HTML?

4. What are the features of HTML5?

5. What are the difference b/w iframe and embed?

6. What is accessibility in HTML?

7. What is DOM?

8. What is the relation b/w the DOM and DOM tree?

9. Compile the application in JS fiddle example?

10. Relationship b/w the html body and h1 & h2 tag in DOM tree?

11. Explain DOM methods or what are methods in DOM?

12. How do you fetch the element context in HTML DOM?

13. Difference b/w class and id?

1. What do you understand by class and id?

2. By mistake, I added 2 the same id, how HTML behaves?

5. What is the used of semantic tags in HTML?

6. What are meta tags?

7. What are data attributes in HTML?

8. What is picture tags?

9. What are other tactics we use to optimize images for responsive layouts?

10. What are storages in HTML?

11. What are the use case of local storages and session storages?

1. What is HTML5 new tags?

2. What is HTML5 APIs?

3. What is the difference b/w local storage and session storage?

4. What is the difference b/w Canvas and SVG?

1. Features

2. <audio>, <video>

3. LocalStorage vs SessionStorage

5. HTTP Methods

6. data-\* Attributes

7. target attribute : \_blank

8. <datalist>

9. Web Worker

Web Storage

CSS

14. What is CSS?

15. What are the differnt ways to use the CSS in HTML5?

16. How can you define external file CSS in HTML?

17. What is box model?

18. What is important in CSS?

19. Features of pre processors in CSS?

20. What is mixins?

21. What is flex? - Center an element.

22. What is grid? - Center an element.

34. Positions in CSS.

35. What is display flex.

36. How to middle a div center using flex.

37. What are viewports in CSS. Why we use.

38. What are Doc types in HTML.

39. If we don't give any doctype what will happen to document.

12. What are features of CSS pre processors?

13. What are pseudo classes and pseudo selectors in CSS?

14. What is CSS specificity?

15. What is flexbox in CSS?

5. How many ways we use CSS in our application?

6. What is box model in CSS?

7. What is opacity is CSS?

8. Difference b/w display block and inline block?

9. What are the pseudo classes in CSS?

10. What is flex? What are major properties in flex?

11. What is viewport height and width?

12. How we achieve responsive design in CSS?

13. What is better way to write CSS for better approach?

14. What is bootstrap grid system?

15. What is tailwind CSS?

13. What are preprocessors?

14. Diff b/w less and sass?

17. What is source map?

5. What are pre processors?

6. Material UI using SCSS built , can affect the container on your custom resposive break point?

0. Responsive Web Design - The Viewport

1. Flexbox vs Grid

2. vertically and horizontally centered

display: flex;

justify-content: center;

align-items: center;

3. The position Property

static

relative

fixed

absolute

sticky

4. Margin vs Padding

5. CSS Preprocessors

Differences between SASS & LESS

Sass: Syntactically Awesome Style Sheet

LESS: Learner Style Sheets

6. Media Queries

7. display: none vs visibility: hidden

8. CSS Selectors

9. box-sizing: border-box;

GENERAL

1. What are the techniques you providing the estimations.

2. How do you managing the sytemation?

3. On what basis you give estimation, how you check the complexity of the screen?

4. Tell me the some good engineering practices that we can apply while creating a page in the React?

5. On performance basis how we decide this component will be best?

7. How we do the unit testing on the code?

10. How you deploy your code? CI/CD or Manually?

11. How you deploy the code to the QA team.

1. What is the difference b/w tailwind and bootstrap?

2. Have you worked with RWD?

4. To develop HTML pages what are the best practices and standards with seo?

27. In terms of HTML & CSS what kind of optimization we use?

16. What is the advantage of using the bootstrap?

1. Explain the entire flow for the login and authentication in React? Explain scenarios in brief?

2. What are the roles of JWT tokens?

3. What happen when token gets expired and how user maintain the login? So that user having an seamless experience?

NEXT JS

4. Core principal of Next JS.

5. Difference b/w page based routing and app based routing.

6. How can we implement dynamic routing with Next JS.

7. Types of rendering in Next JS.

8. How do you define it is client side or server side.

9. How Next JS handles code splitting.

NextJS-

Features->

• Hot Code Reload − Next.js server detects modified files and reloads them automatically.

• Automatic Routing − No need to configure any url for routing. files are to be placed in pages folder. All urls will be mapped to file system. Customization can be done.

• Component specific styles − styled-jsx provides support for global as well as component specific styles.

• Server side rendering − react components are prerendered on server hence loads faster on client.

• Node Ecosystem − Next.js being react based gels well with Node ecosystem.

• Automatic code split − Next.js renders pages with libraries they need. Next.js instead of creating a single large javascript file, creates multiples resources. When a page is loaded, only required javascript page is loaded with it.

• Prefetch − Next.js provides Link component which is used to link multiple components supports a prefetch property to prefetch page resources in background.

• Dynamic Components − Next.js allows to import JavaScript modules and React Components dynamically.

• Export Static Site − Next.js allows to export full static site from your web application.

• Built-in Typescript Support − Next.js is written in Typescripts and provides excellent Typescript support.

In Next.js, we know it generates HTML for a page called pre-rendering. Next.JS supports two types of pre-rendering.

• Static Generation − This method generates the HTML page at build time. This pre-rendered HTML is sent on each request. This method is useful for marketing websites, blogs, e-commerce products listing wesites, helps, documentation websites.

• Server Side Generation − This method generates the HTML page on each request. This method is suitable when an html page contents can vary with each request.

• Routing-

• Index Routes − An index.js file present in a folder maps to root of directory. For example −

• pages/index.js maps to '/'.

• pages/posts/index.js maps to '/posts'.

• Nested Routes − Any nested folder structure in pages directory because router url automatically. For example −

• pages/settings/dashboard/about.js maps to '/settings/dashboard/about'.

• pages/posts/first.js maps to '/posts/first'.

• Dynamic Routes − We can use named parameter as well to match url. Use brackets for the same. For example −

• pages/posts/[id].js maps to '/posts/:id' where we can use URL like '/posts/1'.

• pages/[user]/settings.js maps to '/posts/:user/settings' where we can use URL like '/abc/settings'.

• pages/posts/[...all].js maps to '/posts/\*' where we can use any URL like '/posts/2020/jun/'.

How to do page based routing in next js?

Jest:

Can you explain what mock functions are? How are they created?

What do you understand about snapshot testing in context with Jest?

EXERCISE

28. let str = "Welcome to this TCS Drive!";

"out put : "emocleW ot siht SCT !evirD"

str.split(' ').map((e) => e.split('').reverse().join('')).join(' ');

29. Reverse string:

const a = 'sachin';

a.split('').reverse().join('')

30.

const a = [1,2];

const b = [3,4];

// output: [1,2,3,4]

31.

const arr = [1, 2, 3];

const sum = (x, y) => x + y;

arr.reduce(sum, 0); // 6

32.

const nums = [2, 4, 6, 8, 1, 3, 5, 7];

Math.max(...nums); // 8

Math.min(...nums); // 1

29. Write a function to remove duplicate and sort

const a = [9,4,1,3,2,5,3,1,5,6,7,8,1];

// Output: [1, 2, 3, 4, 5, 6, 7, 8, 9]

[...new Set(a.sort((b, c) => b - c))]

[...new Set(a.sort((b, c) => c - b))]

// Output: [9, 8, 7, 6, 5, 4, 3, 2, 1]

34. const items = [

{ id: 1, name: 'a' },

{ id: 2, name: 'b' },

{ id: 3, name: 'c' },

{ id: 1, name: 'd' },

{ id: 3, name: 'f' },

{ id: 1, name: 'a' },

{ id: 3, name: 'c' },

];

/\*

expected result:

[

{ id: 1, names: ['a', 'd']},

{ id: 2, names: ['b']},

{ id: 3, names: ['c', 'f']}

]

\*/

35. merge two arrays and sort accordingly with unique numbers only

const arr1 = [5, 2, [3], 4, 1, ‘test’, { a:1 }];

const arr2 = [6, 5, 1, ‘test’];

// expected result: [1, 2, 3, 4, 5, 6]

36.

(function(x) {

return (function(y) {

console.log(x);

})(2)

})(1);

Snippets:

1. Concate 2 arrays

2. Remove duplicate from array

3. return the number greater than five from defined array

4. return each value of an array multiplied by 2

5. improve call using async await

6.

console.log(false == '0')

console.log(false === '0')

7. Reverse a string