

Contents

General Interview Questions	3
Basic	3
Common.....	3
React	3
Hooks	3
Components.....	4
props	4
Redux	4
Pair programming: -	5
Unit Testing	5
Generators	5
Core JS.....	11
Security	11
Scenario	11
Design Patterns.....	11
Javascript	12
React	12
Html & CSS.....	14
Construct Connect	15
Interview Questions.....	15
Programming questions.....	16
IQVIA	21
Edward Jones	23
Shell.....	25
Interview Questions.....	25
Codility :	25
Interview Coding Questions.....	27
Mastercard.....	29
Interview questions	29
Walsgreen	30
Interview Topics.....	30

Expedia.....	31
Interview Topics.....	31
Cars24	33
EBSCO – Interview Questions	34
Macy’s - Interview questions.....	38
Jiffy’s - Interview questions	40
DBS.....	41

General Interview Questions

Basic

- How will you maintain code quality?
- Did you follow peer code review in your previous projects?
- Do you have experience in SonarQube?
- Are you comfortable writing integration tests using selenium?
- What we should do achieve 80% code coverage?
- Do you have experience in AWS?
- Do you have experience in CI-CD?

Common

- Difference between React and Angular?
- Performance wise Angular and react both give same results? Then why React.js?
- What is event loop
- When a setTimeout function is set for 1s and the function inside it takes longer (what happens)
- If we have three js files that are of different sizes then which one loads faster
- Js is single threaded then how does set interval and set timeout work?
- Lifecycle methods in react
- What is a higher order component
- What is redux/mobx
- Advantages of redux/mobx
- Performance Optimization
- learn design pattern
- Learn Solid Principle
- Reducer
- ReactJS / Redux
- Unit Testing

React

- Life cycle of React hook?
- Redux – what are middleware of redux?
- Difference in React and Angular?
- Performance wise Angular and react both give same results? Then why React.js?
- Components of Redux?
- What is Pure component in React?
- What is HBJs?

Hooks

- What are react hooks and why we need it
- How to use callback hooks and use state hook
- What is Use Effect hook and how it works and executed
- Life cycle of React hook
- How to achieve component didmount lifecycle hook in react hooks

- Uncontrolled components in hooks (useRef)
- how to use callback hooks and use state hook

Components

- why functional component
- container vs presentational
- What is Pure component in React?
- pure component vs functional
- component update

props

- Render props
- props drilling

Redux

- Components of Redux?
- What is your opinion with state management in react (is it complex and if so, is it necessary)?
- Redux-saga **generators**
- Why we need Redux when we already have React Context?

Pair programming: -

- For a given mockup create a small application using hooks (to-do application with delete)
- Design a Facebook like newsfeed
- Design a news aggregator page
- Design a chat interface

Mobx vs Flux vs redux
reducers

what are the methods execute before componentdidmount

service worker vs web worker

Unit Testing

- why jest why not jasmine
- jest & Enzyme

Generators

What are generators and where do we use them

how do u explain generator in simple words

- how to mock entire module in react
- how to find component performance issue
- new react testing library
- promises vs callbacks
- prototypical inheritance
- Refs
- Ng-If & Ng-For equivalent implementation in react
- Typescript
- Es6
- Web workers
- Service workers
- What is pure components
- What we will apply to functional component to make it pure
- What is cursor
- How we can run multiple git commands in one command
- Why we use npmrc file
- Webpack code splitting
- What is useLayout effect
- How to share data globally without props drilling
- When we will use class component if it is not feasible to use functional component
- Css Box model
- Specificity
- Opacity

- What is flex box
- How to center div in another div using flex box
- why we use render props instead we can use children
- Why hooks are better
- Difference between class n functional components
- What is useCallback hook
- boxsizing
 Css positioning like relative absolute
 How change detection works
 How Angular services are works
 when we do unit testing if call one method and that method have another method how we will know that second method is called.
 Difference between Service n Provider
 How to make div as round shape
 How we can separate two classes with name.
 What is peer dependencies
 what is package-lock.json file
 What is web security
- Difference between call and bind method,
 when we will use bind method
 why we need separate master n other branches
 Lazyloading in react n Angular

what is typescript interface

where we will use interfaces in other programming languages

What design pattern you have used

Where we will closures in daily

why we use closures

what is eventloop js

how eventloop works

why we use promises instead of callbacks

what is promisify

what is thunk and why we need thunk

what redux saga

what is viewchild angular

diff between controlled n uncontrolled components

Presentation n Container components

How we will use redux in container component.

what is git flow

Diff between git fetch and pull

What is HBS

react lazy loading

transclusion

- redux vs context api
- why redux why not context api

- How react with redux works
- code splitting
- can we use redux directly
- diff between var and const
- Redux Thunk
- What is Promise and how it works
- React Lifecycle Hooks
- Higher Order components
- A component is having two sibling child components and how can we use one child state in another
- why we need redux?
- have you used middle ware and why we need it?
- diff var and let
- event bubbling in javascript
- generators in es6
- Box Model
- Sudo Classes like before and after
- How to make div center using flex
- explain about position relative and position absolute
- css specificity
- Pure components in react
- Why Redux?
- Redux default functions.
- How to access elements in react component?
- Why don't browser render JSX directly?
- Drawbacks of react
- What happens to child element if given "margin-left:auto and margin-right:0"?
- display block and display flex
- Semantic tags. Difference between div and header?
- Server Side Rendering in React.
- Best practices in Html
- Design patterns in js
- Find the output for a program based on the concept of hoisting
- Arrow function and normal function?
- Animation in html
- Can you rate yourself in HTML, CSS, Javascript, Object oriented on a scale of 1 to 5
- Do you know Inline element and block level element?
- Can we convert inline element to block level element, how?
- Which are HTML5 new tags?
- What is the Difference between Header, footer?
- Header footer will be in used in development area, what is that area?
- What are new main API's in HTML – web workers, service workers etc.

JavaScript:

- 0) What is Mutable and immutable variable, give few examples?
- 1) Few scenario based questions
- 2) Do you know restructuring? Please explain

Object Oriented programming:

- 1) What is Closure?
- 2) What is difference between Class and pure function
- 3) What is Constructor function in Javascript, how to define Javascript constructor?

React.js:

react lazy loading

transclusion

redux vs context api

why redux why not context api

- How react with redux works
- code splitting
- can we use redux directly
- diff between var and const
- Redux Thunk
- What is Promise and how it works
- reducers
- what are the methods execute before componentdidmount
- redux thunk vs redux saga
- service worker vs web worker
- why jest why not jasmine
- how to mock entire module in react
- how to find component performance issue
- new react testing library
- promises vs callbacks
- prototypical inheritance

- Refs
- Ng-If & Ng-For equivalent implementation in react
- Typescript
- Es6
- Web workers
- Service workers

- Unit Testing
jest & Enzyme

1. What is pure components
2. What is transclusion
3. What we will apply to functional component to make it pure
4. What is cursor
5. How we can run multiple git commands in one command
6. Why we use npmrc file
7. Webpack code splitting
8. What is useLayout effect
9. How to share data globally without props drilling
10. When we will use class component if it is not feasible to use functional component
11. Css Box model
12. Specificity
13. Opacity
14. What is flex box
15. How to center div in another div using flex box
16. why we use render props instead we can use children
17. Why hooks are better
18. Difference between class and functional components
19. What is useCallback hook
20. boxsizing
21. Css positioning like relative absolute
22. How change detection works

23. How Angular services are works
24. when we do unit testing if call one method and that method have another method how we will know that second method is called.
25. Difference between Service n Provider
26. How to make div as round shape
27. How we can separate two classes with name.
28. What is peer dependencies
- 29 what is packagelock.json file
30. What is web security
31. Difference between call and bind method
32. when we will use bind method
33. why we need separate master n other branches
34. Lazyloading in react n Angular
35. what is typescript interface
36. where we will use interfaces in other programming languages
37. What design pattern you have used
38. Where we will closures in daily
39. why we use closures
40. what is eventloop js
41. how eventloop works
42. why we use promises instead of callbacks
43. what is promisify
44. what is thunk and why we need thunk
45. what redux saga
46. what is viewchild angular
47. diff between controlled n uncontrolled components
48. Presentation n Container components
49. How we will use redux in container component.
50. what is git flow
51. Diff between git fetch and pull

Core JS

1. Explain about event loop in javascript?
2. What is polyfilling?
3. What is the use of babel?
4. Let vs Var
5. Questions on Promises and Async-Await
6. Explain DOM event lifecycle(phases)
7. Do you know CORS? How will you handle it?
 - a. What is a polyfill
 - b. Write a polyfill for bind
 - c. What is prototypical inheritance
 - d. What is the difference between == and ===
 - e. What is variable hoisting?
 - f. What is a promise
 - g. What is observable

Security

- h. What is xss
- i. What is same origin policy
- j. What is http only cookie

Scenario

Design Patterns

Javascript

- Define various scopes - Global, Function and Block
- Variable hoisting with var and let. Asked output of a code on Hoisting
- What is a Context and how to bind this.
- Inheritance in Javascript (Prototypal Inheritance)
- Asked to write an algorithm for finding nth number in a fibonacci sequence with various approaches and discussed about which one is more efficient and why.
- If a client asks the website is slow enough, what would be your approach to resolve this.
- Describe the whole process how a website gets rendered under the hood and what is the critical rendering path.
- What is CORS and how do you resolve CORS error.
- What are XSS, Cross site scripting, Cross site request forgery.

Query selector and query selector all difference

map, reduce, filter.

closure and where did you, give example in real time.

What is asynchronous programming, callback, promises, async await?

Use strict in js

Block scoping let var const.

How can you achieve theme (different colors for button, text and body and other html elements) based on functionality? For a web page?

Styled components

React:

How can you pass data from parent to child and child to parent?

How can you improve performance of a react application?

What are hooks? Use effect,

Controlled and uncontrolled components

What do you prefer functional or class components?

If parent component render child components will re render, how can you stop unnecessary child component re rendering in react

Html & CSS

- What is box model?
- What are semantic elements?
- How can you do rectangle circle triangle (top bottom) in CSS?
- What is Flex Box? Did you use it?
- Pro's and Con's of SPA.
- What is semantic markup ? Why we need to consider it? How will you achieve?
- What css methodologies do you know? (like BEM,CSS Modules,Styled Components)
- How will you handle cross browser issues?
- Do you know Flex layout in CSS, what is the difference in display block and display flex?
- Due element – left 0 , right 0, what will happen?
- CSS scenario based questions
- Do you know Grouping/inheritance in CSS?
- CSS preprocessor: LESS, SCSS

Construct Connect

Interview Questions

JavaScript:

- Closures
- Inheritance
- var,let,const
- Event Loop
- Data types in js
- function Vs arrow function
- Hoisting
- scope in js
- Promises- practical (.all and .settledAll)
- static functions-practical
- super()
- Prototypal inheritance
- difference between let const var??
- hoisting (var & function)??
- spread & rest operators
- explain event loop with setTimeout example

CSS:

- Positioning of elements in css
- explain position properties in detail
- explain display properties in detail
- display flex
- CSS specificity with example

React:

- Lifecycle
- useEffect(): in depth
- refs
- optimization: in dept
- HOC and in built HOC examples

Testing:

Practical questions

- How to know the value of counter got changed on button click?
- Some questions on testing the rendered components

Redux:

Theoretically everything in redux

Other:

CI/CD

Code review

Programming questions

1) Predict the output

```
console.log(a)

var a = 10;

console.log(b)

let b = 10;
```

2) Predict the output

```
const a = [ 1, 2, 3 , 4];

a.push(5);

console.log(a)
```

3) Predict the output

```
console.log('hi')

setTimeout(() => console.log(' in timeout'), 0)

console.log('hello')
```

4) write code

write a function which mimics the behavior of fetch .

Example

```
mockFetch(10).then().catch();
```

write the implementation for mockFetch.

5) what is closure and explain with a code snippet example.

React js:

1) life cycle methods

2) writing setState in different ways

3) Hooks:

a) useEffect in detail and what are the matching lifecycle methods of class component w.r.t
useEffect

b) useContext ? what is its benefit?

c) useMemo and useCallback

4) explain basic concepts of redux (store, reducer, action)

5) unit testing in reactjs

good to have knowledge on:

Babel, webpack, create react app without using facebook's create-react-app.

Js:

What Is closure?

What is arrow function?

Map, filter, reduce.

Let const var.

Promises, async await, generators.

Observables

Which is fast, promise or set Timeout?

Context in JS?

React JS:

What is static in react?

Can we call this in static `getDerivedStateFromProps`?

How can you extend properties and methods? In class

React Life cycle methods?

What is pure component?

What are functional components?

What are hooks?

What is `usememo`, `useEffect`?

What is `redux`?

Context api In reactJS? `UseRef` and `ref` ?

What will it return when you write `ref` on components?

What are higher order components?

What is `saga`? What do you prefer `saga` or `thunk`?

Is `redux` by default synchronous? Why how can we achieve asynchronous?

Write a timer function using hooks which can decrease `<Timer sec = {35}>` should print 34 33 32.. it should stop what `sec` equal to 0?

Performance in react js?

CSS:

How can we arrange a div center to page? Using CSS (flexbox)

What are positions?

General topics:

What is continuous integration?

Unit testing how do you test react component explain?

JS:

Closures

Inheritance

var,let,const

Event Loop

Data types in js

function Vs arrow function

Hoisting

scope in js

Promises- practical (.all and .settledAll)

static functions-practical

super()

Prototypal inheritance

CSS:

Positioning of elements in CSS

React:

Lifecycle

useEffect(): in depth

refs

optimization: in dept

HOC and in built HOC examples

Testing:

Practical questions

-- How to know the value of counter got changed on button click?

-- Some questions on testing the rendered components

Redux:

Theoretically everything in redux

Other:

CI/CD

Code review

IQVIA

- Why hooks are implemented?
- What is Redux-thunk and why we need it?
- What is React.memo()?
- What are sockets?
- What is challenging task you have done in previous company?
- What are selectors?
- What do you invoke arrow functions and functions in JSX?
- Explain Redux architecture?
- How do u create redux store and in which file you pass it?
- Why we pass MapStateToProps and MapDispatchToProps ?
- Is MapDispatchToProps is Mandatory for connect function?
- How do u update Object content in a local State and do it using spread operator?
- How to achieve component life cycle in hooks?
- Lifecycle methods – Phases
- Context API – Provider and consumer
- Higher order component
- React-Redux - Why mapStateToProps
- Router concept – BrowserRouter, HashRoute
- history.push()
- Dynamic routing
- Functional hooks - useSelector
- hydrate method
- Logical based question to pass data
- Render props
- Props validation
- What is the benefit of virtual DOM?
- What is the benefit of using class components?
- What lifecycle methods that hooks substitute?
- What is the use of useCallback and useMemo hooks?
- What strategies do you follow while optimizing react app?
- How will you measure performance of react app?
- How will you avoid unnecessary renderings?
- What is shouldComponentUpdate ? what it will return?
- How will you share state between components?

Edward Jones

1. Explain about recent interesting bug that you came across.
2. Given a react component and asked to do code review.
3. Gave sample JavaScript code samples and asked to give the output.
4. Given the below code, re-write without "try/catch".
5. General strategy for validating the fields in UI.
6. What are the things that you look while doing a code review?
7. What will you do when client comes with an emergency requirement in the middle of the sprint?
8. What will you do when someone reviewed your code and said the approach is totally wrong and code has to be rewritten?

1. Interesting/Challenging tasks done in past projects
2. One least favourite feature of Javascript and ReactJs
3. Cookie related questions
4. If get a chance to work on past projects what would you like to change
5. When would you use a class component versus a functional component?
6. What will be output?

```
console.log('first');  
setTimeout(() => console.log('second'), 0);  
console.log('third');
```

7. What will be output?
save().then(callback1) // 5 s to execute
update().then(callback2) // 1 s to execute

```
function callback1() {  
  throw Error()  
}
```

```
function callback2() {
```

```
}
```

```
8. const employees = [  
  { id: 11, name: 'A', managerId: 11 },  
  { id: 100, name: 'B', managerId: 11 },  
  { id: 50, name: 'C', managerId: 100 },  
  { id: 200, name: 'D', managerId: 100 },  
  { id: 300, name: 'E', managerId: 200 }  
]
```

write getManager(employees, 50) function which will return list of managers for given employee id

9. how to provide validations for form after user enters the data/ based on user entered data
10. Async await

Shell

Interview Questions

- What are the datatypes in Javascript?
- What is prototypal inheritance?
- What is Closure?
- What is an Event bubbling?
- How to stop event propagation?
- Difference between stopPropagation and preventDefault?
- Can we load specific module in angular based on requirement?
- Which framework(Angular or React) will you choose?
- Were you using any repositories to share your codebase?
- Do you know GIT commands?
- Any idea about JIRA?
- How to resolve a Promise
- Saga effects
- Generators
- Useeffect lifecycle methods
- Write a functional component with hooks for a login component
- Rest/spread operators
- What is webpack?
- Explain about webpack configuration
- React vs other frameworks/libraries, which is better and why?
- CI/CD knowledge
- SonarQube and fortify
- Array methods map reduce filter
- Prototypal inheritance
- How do we restrict routes when use is not logged in
- What happens if we do not write yield in saga before call/put
- HOC and examples
- Explain React testing library
- Unit testing
- Saga vs thunk, which is better
- Redux vs context API
- What factors should be considered when converting React app to React Native mobile app
- Typescript interfaces

Codility :

Test 1

1. Create a component with Todo List items.
 - Case 1: Input field with add button, When user will fill and press add button should be added in the list
 - Case 2: Once user will click on any item a class should be added which presents line-through the text
 - Case 3: Show the total added list and total line through items
2. Write a function that, given string like 'aaaa' and array like [3,4,5,6] of integers, both of length N=12, returns the minimum cost of all necessary deletions.

Test 2

React.js

1. Users can like a post using LIKE button => the like count should increase by one.
2. Users can cancel their like by clicking again on the LIKE button => like count should decrease by one, "liked" class should be removed from the button.
3. Users can dislike a post using DISLIKE button => the dislike count should increase by one.
4. Users can cancel their dislike by clicking again on the DISLIKE button => dislike count should decrease by one, "disliked" class should be removed from the button.

Nodejs

1. 2 API post and get.
2. Post API to add the unique data of the candidate to the database (in memory) with unique id, unique name and array of skills.
3. GET API which retrieve the candidate with the most number of skills set passed as parameters.
4. Make sure to pass all the response status such as 200,404,400.

Interview Coding Questions

Prototype in JS

Inherit a object into another

=====

replace values of one object from another

```
obj1 = {  
  name: 'abc',  
  age: 23  
}
```

```
obj2 = {  
  name: 'xyz',  
  age: 45,  
  print: function(){  
    console.log(name);  
  }  
}
```

=====

write a fn to print hello world 5 times, with interval of 1 sec.

=====

write a fn to print hello world 5 times, with interval of 1 sec.

=====

promises in React.js

=====

```
function print() {  
  console.log('1');  
  delay(1000);  
  console.log('2');  
}  
function delay(x){  
}  
print();
```

//write the delay function

=====

Div grid in react for collapsable functionality

=====

table implementation in react and getting data from api

=====

Mastercard

Interview questions

1. What is react ?
2. What are lifecycle methods?
3. what are hooks and why do we use them?
4. what are functional and class based components?
5. What are generator functions and where have you used them in your project previously?
6. What is event loop?
7. Promises vs call back, which one is better and why?
8. What do you understand by async and await?
9. What are render props?
10. What is context API?
11. What is redux and why its needed?
12. What are the components of redux?
13. What is the significance of middleware? Which one have you used?
14. What is redux-saga?
15. How do you optize the performance of React app ?
16. Which tools have you used for optimizing the application?
17. What is unit testing and which tools have you used for doing unit testing?
18. What are HOCs and why they are required?
19. Can you share a real life scenario where HOCs are used?
20. HOW do you implement routing in react ?
21. What is destructiong and how do u do it?
22. Use of UselayoutEffect and useMemo hooks in react?
23. HOW do you implement different lifecycle methods using react hooks?
24. What is suspense in react?
25. Why do we use UseRef hook?
26. How do you center a div using flex box?
27. ES6 features
28. React Hook in depth
29. Use of functional component.
30. How to test basic components using JEST.
31. React optimization techniques knowledge

Walsgreen

Interview Topics

event loop: +
execution context: +-
arrow function: +
hoisting: +
temporal dead zone: +
immutable: +
Promise vs Observable: +
Async await: +
Prototypical inheritance: -+
Fibonacci via array: +
Fibonacci via recursion: -+
Algorithm complexity: mentioned recursive approach is not optimized due to the necessity to remember the execution context (+)
Big O notation: +-
Shadow DOM: +
CSS position: +

Expedia

Interview Topics

- curry functions, context(this), array prototype methods
- polyfills for native javascript functions- promise.all, promise.allsettled, array methods.
- React - design flow, high level components, context api in react.
- React hooks, custom hooks, lifecycle methods in react.
- virtual dom vs shadow dom, react router
- lazyloading, tree shaking

Javascript questions –

1. Variable scoping and hoisting.
2. What is reduce function and describe how to use it. Also implement your own reduce function?
3. Scenario based question – if there are 1000 buttons on a webpage, how would you register click event handler for all without using loop?
4. Outputs for following code snippets –

a).

```
Console.log(a); //undefined
```

```
Console.log(b); // error
```

```
Var a;
```

```
Let b;
```

b).

```
var b = 10;
```

```
function outer(){  
    console.log(b);  
    b++;  
}
```

```
        console.log(b);  
        var b = 3;  
        console.log(b);  
    }  
    Outer();
```

c).

```
var p = Promise.resolve('foo');  
async function callback(){  
    let result = await p;  
    console.log(result);  
}  
callback();  
console.log(p.then(data => console.log('bar')));
```

CSS Questions –

1. Can an app performance be improved using CSS. What are the best practices for that?
2. Design a Toggle button with animated slide transition on click using HTML and CSS only.

Cars24

JS Questions :

Scopes

Hoisting

Closures and examples.

new keyword

Bindings

Reactjs

Unit testing related to react.

EBSCO – Interview Questions

ReactJS:

1. What is the benefit of virtual DOM?
2. What is the benefit of using class components?
3. What lifecycle methods that hooks substitute?
4. What is the use of useCallback and useMemo hooks?
5. What strategies do you follow while optimizing react app?
6. How will you measure performance of react app?
7. How will you avoid unnecessary renderings?
8. What is shouldComponentUpdate ? what it will return?
9. How will you share state between components?

Redux:

1. Why we need Redux when we already have React Context?

HTML & CSS:

1. Pro's and Con's of SPA.
2. What is semantic markup ? Why we need to consider it? How will you achieve?
3. What css methodologies do you know?(like BEM,CSS Modules,Styled Componenets)
4. How will you handle cross browser issues?
5. What is box model?
6. What is Flex Box? Did you use it?

Javascript:

8. Explain about event loop in javascript?
9. What is polyfilling?
10. What is the use of babel?
11. Let vs Var
12. Questions on Promises and Async-Await
13. Explain DOM event lifecycle(phases)
14. Do you know CORS? How will you handle it?

NodeJS&Express:

1. Why we need express js when we already have node js?
2. Some questions on express js features like Routing, Middleware

General Questions:

1. How will you maintain code quality?
2. Did you follow peer code review in your previous projects?
3. Do you have experience in SonarQube?

4. Are you comfortable writing integration tests using selenium?
5. Why we should achieve 80% code coverage?
6. Do you have experience in AWS?
7. Do you have experience in CI-CD?

1. Tell me about yourself?
2. Which version of React have you used?
3. Mention newly added features to 16.8?
4. What do you like/dislike about React hooks?
5. Explain how useState can be used?
6. Is Javascript synchronous?
7. How does Javascript handles Asynchronous code ?
8. How is Async await different from promises?
9. Explain React lifecycle hooks?
10. Why getDerivedStateFromProps is important?
11. Explain Redux pattern?
12. What is a pure component?
13. What is Memoization?
14. Latest array methods in ES6?
15. What Array.from does?
16. What reduce method does?
17. What array.some does?
18. Which unit testing frameworks have you used?
- 19. Scenario :-** We have an input field and submit button. After entering text in input field and clicking on submit button, API call happens. How to write unit test for this component?
20. What is the purpose of Webpack?
21. Why module bundling is important?
22. You mentioned RxJs in your resume what is the purpose of Rxjs (used with Angular)?
23. You mentioned D3.js in your resume what visualization you have created, what kind of data was used?

Coder

- Shopping cart

Add

Macy's - Interview questions

- React component lifecycle methods.
- Is setState method synchronous or asynchronous?
- Can we directly set the state?
- Where in a component, we can initialise the state?
- Explain virtual DOM?
- Why to choose React JS over other technologies available?
- Why SPA is better?
- Explain Redux principle
- Scenarios based question - There is a page with a text field and submit button. When user types in the text field and clicks submit button, it should validate the text in the text field and based on the validation result, show a new component saying either "Thank you" or "Text is invalid".
- What is promise and how to handle it? What are different ways to handle it?
- Async and await
- Multiple promises are there, how to handle those and improvise
- hooks in react, why were they introduced?
- What all things are possible with hooks
- Can we use one hook inside another hook?
- How to write custom hook?
- useEffect hook different scenarios
- Styled components in react
- CSS themes and why they are used
- CSS modules
- React hooks rules
- Custom hook conventions and restrictions
- Average of a value of an array of objects - Which technique would you use?
- Array reduce method, how does it work, what arguments does it take?
- How to handle asynchronous code - Promises, promise chaining, how is chaining achieved
- What does a Promise return?
- API fetching - fetch/axios.
- How is React hooks different from class based react?
- Can we use componentDidMount in react hooks? If not, why?
- Which technique would you use to find the average of a value of an array of objects?
- How does array reduce method work and what arguments does it take?
- How to handle asynchronous code in JavaScript?
- What is a promise. How is promise chaining achieved?
- If one asynchronous call is dependent on another, how to handle with .then() and .catch(). What does the .then() method take?
- How to handle parallel execution of multiple asynchronous calls which are not dependent on each other?
- How to get the results irrespective of promise rejected/success from multiple async calls?

- How to handle parallel execution of async calls with `async/await`? How to order them if one call takes longer time than another?
- What is a difference between class and functional component?
- Why can't we have lifecycle methods in functional component?
- What is the use of `useReducer()` hook?
- How the session is active if a user doesn't log out of his account [ex: facebook] and returns next day. How is it achieved?
- In an e-commerce site, explain the process that happens under the hood from the moment a user types in a particular category and press Enter in search bar. [no need of backend details]
- How the user is navigated to the products he viewed even after page refresh/page back.
- What is a JWT and what is contained in the payload?
- Did you get a chance to work with authentication before?
- How to pass data from one component to another which doesn't have parent child relationship?
- How is CSS theming achieved for a component using styled components?
- Which technique would you use to find the average of a value of an array of objects?
- How does array reduce method work and what arguments does it take?
- How to handle asynchronous code in JavaScript?
- What is a promise. How is promise chaining achieved?
- If one asynchronous call is dependent on another, how to handle with `.then()` and `.catch()`. What does the `.then()` method take?
- How to handle parallel execution of multiple asynchronous calls which are not dependent on each other?
- How to get the results irrespective of promise rejected/success from multiple async calls?
- How to handle parallel execution of async calls with `async/await`? How to order them if one call takes longer time than another?
- What is a difference between class and functional component?
- Why can't we have lifecycle methods in functional component?
- What is the use of `useReducer()` hook?
- How the session is active if a user doesn't log out of his account [ex: facebook] and returns next day. How is it achieved?
- In an e-commerce site, explain the process that happens under the hood from the moment a user types in a particular category and press Enter in search bar. [no need of backend details]
- How the user is navigated to the products he viewed even after page refresh/page back.
- What is a JWT and what is contained in the payload?
- Did you get a chance to work with authentication before?
- How to pass data from one component to another which doesn't have parent child relationship?
- How is CSS theming achieved for a component using styled components?

Jiffy's - Interview questions

- React component lifecycle methods.
- What is React?
- Which other JS library did you work on?
- What is virtual DOM
- What is server side rendering
- Explain some React hooks
- Class and functional components
- Component lifecycle methods.
- What is Redux and why we use it?
- What are new ES6 features?
- What is async await and promises?
- What is browser event loop?
- Version of react used and Class/functional components
- Knowledge of react hooks.
- Explain react lifecycle methods.
- What is the purpose of `getSnapshotBeforeUpdate` lifecycle method?
- Explain the redux flow.
- How actions are dispatched.
- How to connect react and redux.
- How to do shallow copy and deep copy of objects.
- Write a recursive function for performing deep copy.
- What is the difference between `async`, `defer` and `prefetch`?
- What is the difference between `React.memo` and `useMemo`?
- What is `debounce` and `throttling`?
- Explain `reduce` function.
- What are synthetic events?
- CSS box model.
- CSS Flexbox.
- Explain Context API.
- Which library do you use for promises/API calls? How to attach headers to each request using `Axios`?
- Do you have experience in Code reviews and micro services?

DBS

List of topics for DBS:

1) REACT

- React.js – Importance and Use Cases, when to use
- Virtual DOM in React
- Basic concepts like:
 - o State & Props
 - o Events in React
 - o Functional vs Class based components
 - o Lifting state up
 - o Error Boundaries, Fragments, Portals
 - o Context API
 - o React Forms
 - o Higher Order Components
- Component Lifecycle methods
- Render props
- Props drilling
- Create a Todo List application in react and demonstrate the below actions:
 - o Create a Todo
 - o Edit a Todo
 - o Delete a Todo
- Hooks in React
- Client Side Routing
- PropTypes, Typechecking
- Advanced React Concepts
- Testing Components in React: React Testing Library

2) TYPESCRIPT

- Why use typescript
- Advantages
- Implementation use cases

3) STYLED COMPONENTS

- What are they
- Advantages
- Use Cases

4) STATE MANAGEMENT LIBRARIES (Redux, Redux Saga, Mobx)

- Need of state management libraries
- Redux Thunk
- What are Actions, Dispatchers, Reducers, Store, Provider
- Difference between Redux & Redux Saga
- What are generators in Redux Saga

5) UNIT TESTING LIBRARIES/ FRAMEWORKS (eg: react testing library)

- Why should we do testing
- Advantages
- Use Cases

6) JAVASCRIPT

- Define various scopes - Global, Function and Block
- Variable hoisting with var and let.
- Hoisting
- Operations on Arrays and Objects
- What is a Context and how to bind this.
- Callback, Memoization
- Inheritance in Javascript (Prototypal Inheritance)
- ES6 (let, const, classes, arrow functions, imports, exports, destructuring, spread and rest operators etc)
- Write an algorithm for finding nth number in a fibonacci sequence.
- Optimizing website performance.
- How a webpage gets rendered under the hood and what is critical rendering path.
- What is CORS and how do you resolve CORS error.
- What are XSS, Cross site scripting, Cross site request forgery.
- Query selector and query selector all difference map, reduce, filter.
- Closure
- Asynchronous programming, callback, promises, async await?
- Use strict in js
- Block scoping let var const.

- How can you achieve theme (different colors for button, text and body and other html elements) based functionality? For a web page? Styled components
- Regex in Javascript
- Errors (try, catch)
- Window Object
- DOM events
- Timers (setTimeout, setInterval etc)
- Web Storage

7) NEXT.JS

- Server side rendering
- SEO
- Next js basics
- getServerSideProps and getStaticProps

8) HTML5

9) CSS3

- Flexbox
- CSS Grids
- Preprocessors (LESS, SASS)
- All other CSS3 features

Design 2 web pages with following requirements.

- Show a button/link containing "Click to view color palette" in first page
- When clicked on above link, make an http API call(endpoint provided by client) which returns a color list items and then navigate to second webpage and show them as a grid/tile layout view(a div box containing the applied color code and name below it).
- Use responsiveness based on user device(IPAD/desktop).

Theory Questions:

- i. Redux Sagas basics
- ii. Web Components/Reusable components.
- iii. How to achieve responsiveness in IPAD/Desktop.