1) GEC → Hoisting

2) Event loop

3) Closures → functional component
E, C, D, T, R

4) currying

5) promises → methods

6) prototype, polyfill

7) call, apply, Bind

8) This keyword → examples

9) LS vs SS

10) map vs obj

11) micro front ends

12) data types → tell also abt why type of null is obj

13) Shallow vs deep copy

14) Null vs undefined → legacy bug type tag 0x00
5 points for undefined

15) guess the o/p Questions

16) debounce, throttling

17) event ~~bubbling~~ propagation, event.stop propagation
bubbling delegation

18) filter, map, for each, reduce

19) var, let, const

20) scope chaining

21) shadowing

22) ES6 features

23) types of scopes

24) lexical scope

25) mo

26) HTTP methods

27) Fetch → async await
try{}
catch{}

29) axios

30) authorization vs authentication

31) slice, splice

32) pop, push, shift, unshift

33) includes, index of

34) arrow functions

35) destructing

36) == vs ===

37) implicit coersion

38) + £ - / *

JS Coding:

1) Flatten obj with prefix
2) Anagram → split. sort. join
3) currying for n elements
4) remove duplicates & sort
5) Sort → 2 for loop for(i=0, ...), for(j=i+1, ...) > → swap for ascending
   < → descending
6) Sort based on a value or date
7) fibonacci
8) Flattening with depth → depth > 0
   depth-1
9) debounce & throttling
10) find pairs which sum to target → 2 for loop like sort
11) min & max → sum → 1 for loop ~~Single~~
12) deep clone
13) alphabet sort → sort((a,b) ⇒ a.toLowerCase
   .localCompare(b.toLowerCase))
14) Substring ⇒ str.Substring(start, end)
   ↳ exclusive

React

1) Virtual DOM → Reconciliation → Diffing
2) usecontext → prop drilling, global state management
3) use callback, use Memo }
4) React.memo } improve performance
5) Lazy loding | code splitting }
6) Custom hooks
7) unidirectional flow
8) useEffect, mount, update, unmount
9) usereducer vs useState
10) class vs functional
~~11) React Saga → Thunk~~
12) React Fiber → Pawan's kalyan
13) React version → $\boxed{18}$
14) use Ref → controlled component, uncontrolled components
15) use Navigate
16) what is JSX
17) uses of React over other Vue,
18) Routing → use Navigate
19) why react can't return multiple elements

1) JSX, return element
2) reconsiliation
for root elements

Read coding:

1) fetch with url & display table
2) useEffect
3) useCallback & useMemo
4) useReducer
5) practice. fetch & Table
6) To Do list

19) React error boundaring
20) unit testing → Basic
21) Firebase auth - Basic
22) CI/CD - Basic
23) Internationalization - Basic
24) SSR - Next.JS → React based frameworks with ts/js & typescript
25) useLayoutEffect → Render → Browser paint → useEffect ↗ asynchronous
↳ Render → useEffect → Browser paint
↳ synchronous

CSS coding

1) Pseudo class → state or position :

2) Pseudo element → Part of element or insert ::

3) Box Model → describes how HTML elent → □

4) Flex Box → feature to align item inside a containe
1-dimention, row or column

5) media queries → help to determine what to apply
based on screen size, type, solution

6) selectors → pattern based on this styles are applied

7) positions → property to shift element in Dom based
on T, L, B, R

8) why mobile first.
↳ more people
↳ SEO
↳ avoid bloated mobile version

HTML

1) ✓ Image map
2) ✓ Semantic elements
3) ✓ HTML 5 features
4) accesibility → light house → aria- , ara-live
5) form validations
6) form Data API
7) LL, SS, Indexed DB

Redux:

✓1) Flow → architecture

$$\begin{tikzcd} store \arrow[r] & S,D \arrow[r, "A"] & R \arrow[ll, bend right] \end{tikzcd}$$

✓2) Redux thunk

✓3) Redux toolkit → create aync thunk

→ create slice

→ configure Store

4) Redux Saga

↳ redux Saga → is a library which we generator function from JS. to handle asyn code.

↳ we can pause the code & restart.

↳ Pause → retry → restart

↳ yield & put. [illegible], call methods.

↳ debouncing → add delay

↳