J. MANIVANNAN

1) React

=> JavaScript library
=> Build user interface
=> Component based Architecture

Why?

=> Handles only UI
=> React gives flexibility
=> It is not an framework everything built in.

2) Virtual DOM:

=> lightweight JavaScript object copy of real DOM.
=> update fast

Real DOM:

=> Manipulating it slow
=> Actual DOM rendered in browser

Difference:

=> Diffing Algorithm to find changes
=> Improve performance

3) Components in React

=> Reusable UI building blocks
=> Each component returns JSX

Difference:

Functional	class
use Hooks	ES6 class
Less code, easier to read	this.state
	life cycle method

4) Hooks:

=> Function let you use State and lifecycle features

=> Only used in functional component

Why Hook:

=> Avoid complex class st Component

=> logic shared easily between components

5) useState

State Management

Easy to use

useReducer

Complex state logic

multiple related values

6) Prop Drilling

=> props through multiple Component

=> Intermediate Components dont need them

Problems:

=> Hard to maintain

=>

How to avoid:

=> Component Composition

=> State Management libraries

7) Lazy Loading:

=> load Components only when needed

=> use React.lazy() and suspense

Why:

=> Reduce bundle size

=> Page load faster

8) Purpose of useEffect.

=> Handle side effects

-> API calls
-> Event listener
-> Dom manipulations

Dependency array Behaviour

=> [] -> runs once

[value] -> runs when value changes

No array -> runs on every render

9)

useRef

Does not trigger re-render

Store mutable values

useState

Trigger rerendered

used for UI related data

10) Role of keys in React list

=> Unique identifier for list item

=> Improve performance

=> Key must be unique

=> Avoid using array index if list changes.