43. REACT MC QUESTIONS

Agenda:

- UseRef hook
- MC Questions
  - ↳ StopWatch
  - ↳ Image Carousel (Slider Component)
  - ↳ Custom hook (Create own hook)

- UseRef:
  - ↑ (Reference)
  - → For directly accessing & interacting with DOM elements & for persisting values across renders without causing re-renders.

eg

flow useRef works?

① Creates a Reference : useRef (initial Value).
   Returns mutable object; "It contains 'current' property.

② Persists the value : unlike state, changes the
   'current' property of the ref; do not trigger re-render

③ Accessing the DOM elements

$$<input \ ref = \{..\} \ />$$

Can assign 'ref' to a DOM element

Use Cases of useRef

⌐ ① Accessing DOM elements
   ② Storing mutable values
   ③ Avoid re-initializati

eg   let intervalRef = useRef (null)
                ↓
        Use this to store interval ID
                    ⌐ (by setInterval)

intervalRef. current = setInterval (() => {

                                    ,1000)

Return a Unique ID

ClearInterval (interval current)

StopWatch

Of

hh: mm: ss

00 : 00 : 00

(Display elapsed time)

[START] [RESET] [STOP]

State variables (Component/function)

time,

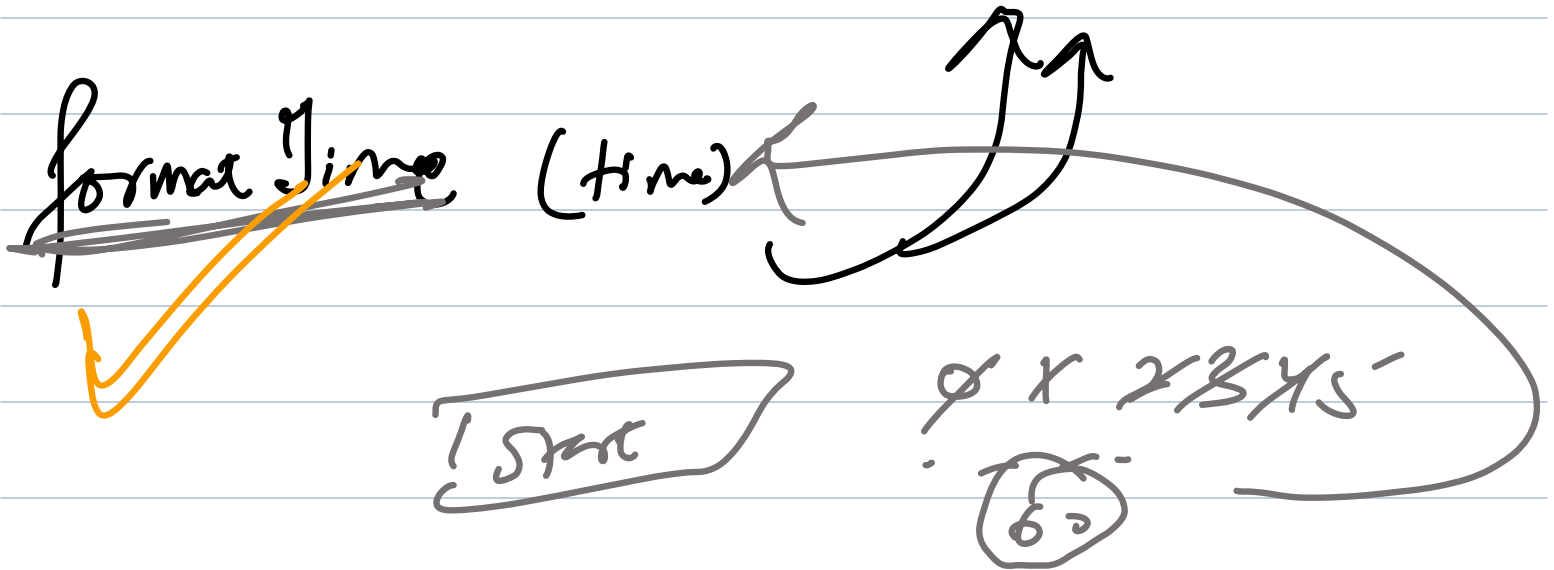setInterval

??

Manipulate with timerID (ClearInterval)

useRef

NO

To save timerID reference.

After 60s, ⟶ 00:01:00

60m ⟶ 01:00:00

formatTime (time)

Start

$\phi \times x/s/y/s$

60

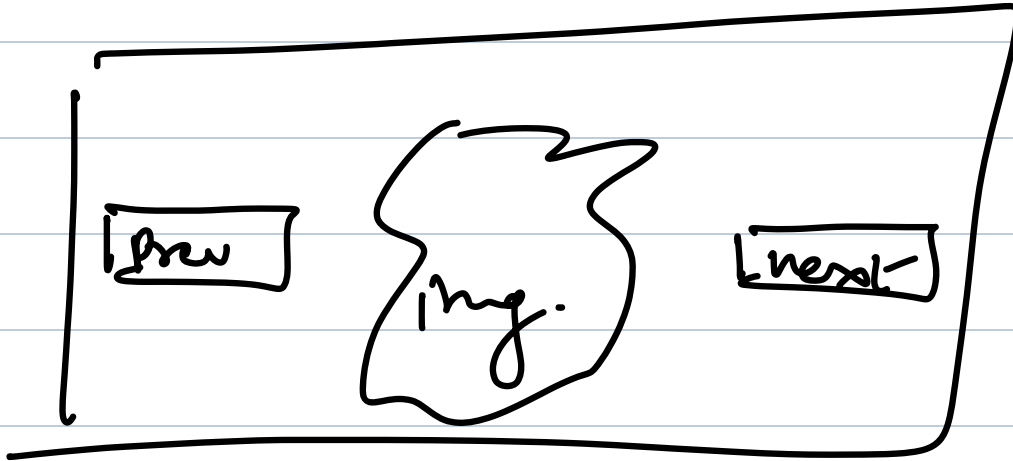< 60 ⟹ 00:00: {time}

> 60 ⟹

61 ⟶ 00: 01: 01

:02

62

63 :03

64 :04

Automatic Carousel

- Every 2s image change: (Automatically)
- Manual Change of image.
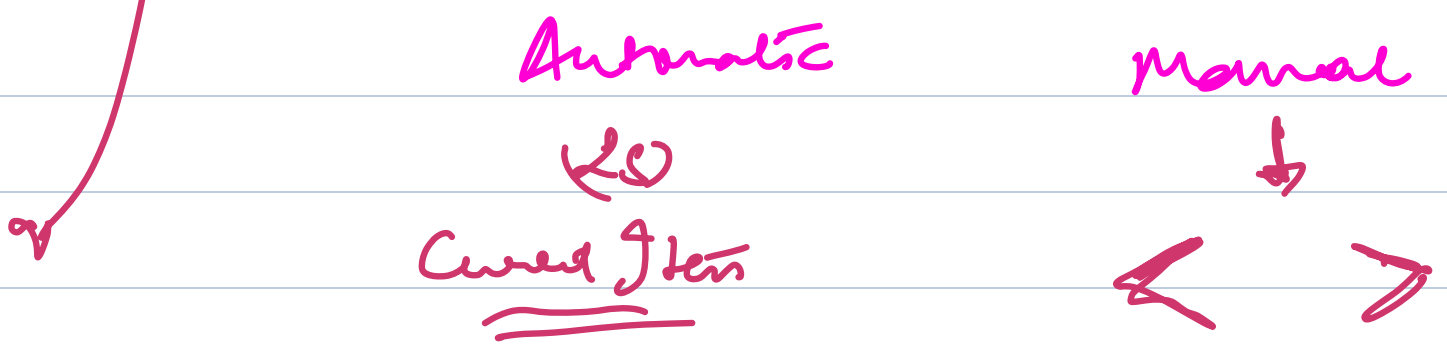- Image, title, description of image.

State → Current Item

function

↓○

Current Items

< >

Next Item → +1
(Prev Item → -1   next Fun
  Use Effect ( ) ⇒ +1 ', 2000)

{
  items · map (l) ⇒   {
                    - -
                    ↳
}

[Custom Hook]

↳ Function that uses React
                    hooks.

Eg: Write a React hook that manages visibility of an element, such as model, dropdown

useVisibility

∟ provide simple interface to show / hide the element
• reusable across different components.

Features

set initial
↗ visibility state

1) Initial Visibility State

2) Toggle ↘ visibility [show ⇆ hide]
  ∟ hook should provide method to toggle visibility.

3) Show & Hide Methods
  ∟ explicitly set show/hide

4) Current Visibility State

└→ return current
            state

<App>   →props change
                              here
  ⟨ <Parent> ⟩
                         } Also
        <Child> re-
                         render
        </Child>

  </Parent>

</App>

React. memo (( ) =>   }

[...] .)