CLIENT SIDE ----------
HTML,CSS
How will the data go from HTML element to JS (process data)
How will the data go from JS to HTML

CLIENT SERVER -------------
How will the data go from Client side to serverside

1. As query param
2. Path param
3. Form method post submit = request body
4. JSON object

How will the data go from server side to client side

5. Res.send(data)
6. Res.render (html)

End Module Exam : ----

7. HTML, CSS, Client Side JS(use in <script> tag or external JS)
   How will you send data from input to JS variable = document obj
   Can you process that variable (String , Date , array)
   How will you return DATA from JS to HTML (innerText,innerHTML)

8. HTML , CSS , Jquery ($(document).ready(()=>{})
   How will you send data from input to JS variable =
   selector,val,checked
   Can you process that variable (String , Date , array)
   How will you return DATA from JS to HTML (text() , html())

9. React
   How will you send data from input to JS variable = (onBlur)
   Can you process that variable (String , Date , array)
   How will you return DATA from JS to HTML (rerender- state {})
   Routing
   Props

Hooks = useState, useParam

React **Ref** = Type of variable that is created only
when the component is mounted
It is not created on RERENDER

This is also true for state variable !!
State has to be set using setState() or setter
---- every time we change the state the page is rerender

Each Ref variable has a property called as **current**
To access the ref value we must say refobj.current

Ref usage in Function Component	Ref usage in Class Component

How to create a ref variable
 	
Function CompName()
{
Let refv = useRef()
}
	How to create a ref variable
	
	Class CompName extends Com..
	{
	Constructor()
	{
	This.refv = createRef()
	}
	}

<input type="text" ref={x} />	<input type="text" ref={this.refv} />
x.current does not take the Value of input type	this.refv.current = The input tag to which it is attached this.refv.current.value = to get the value in the tag
Use the useRef only when you need a Variable that survives RERENDER	

How to use BootStrap in React ?
Add the scripts and CDN links in index.html

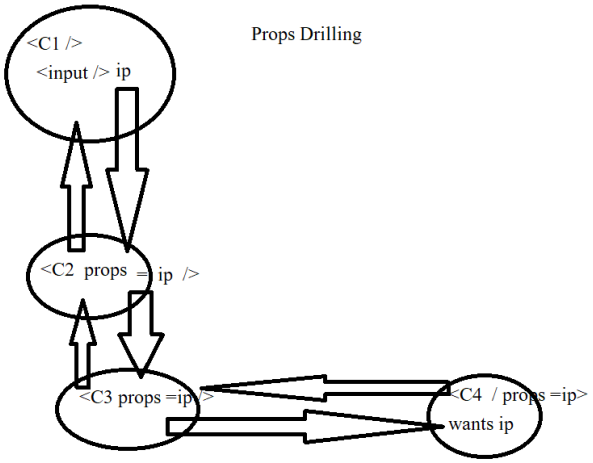
DATA SHARING BETWEEN REACT COMPONENTS ----------

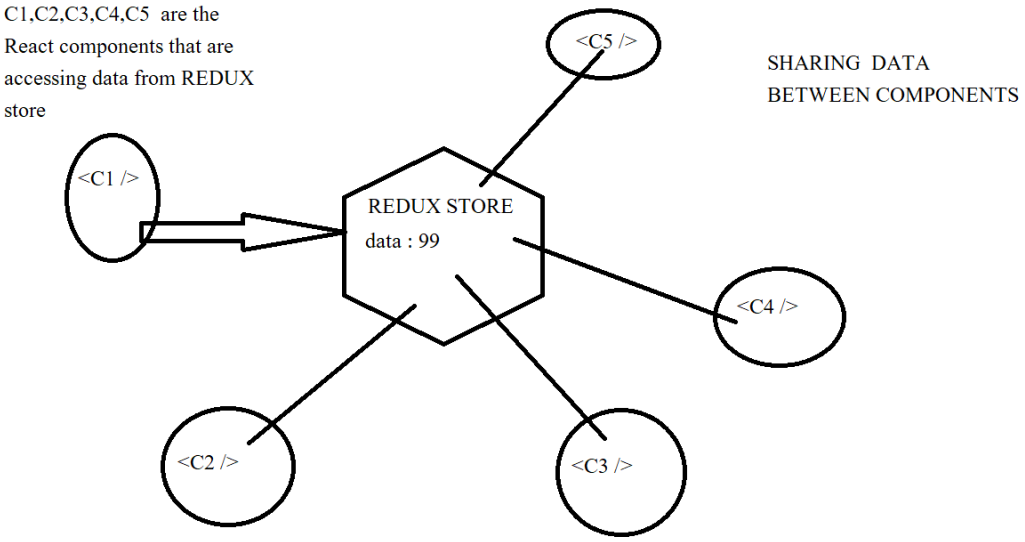
10. Use Props = The communication happens between Outer component (Parent)
and Nested component (INNER component)

This cannot be used if two components are not outer-inner

11. Use localStorage /sessionStorage = Browser maintains a space ,GLOBAL DATA
12. Redux = Third party library to maintain the store of shared data

Redux is solving the problem of PROPS drilling





We may want to **see** the CURRENT STATE of the shared DATA in REDUX STORE
We may want to **change** the CURRENT STATE of the shared DATA in REDUX STORE

The REDUX Observable is defined as a REDUCER function !!

REDUCER function tells ?
What state is stored ? **State parameter**
What operations can be done on the state by the Observers ? **Action parameter**
The job of **dispatch** is to change the state !! What do we want to send ? WE want to send the action on the reducer .

Reducer function returns the CURRENT STATE (after applying action , if any)

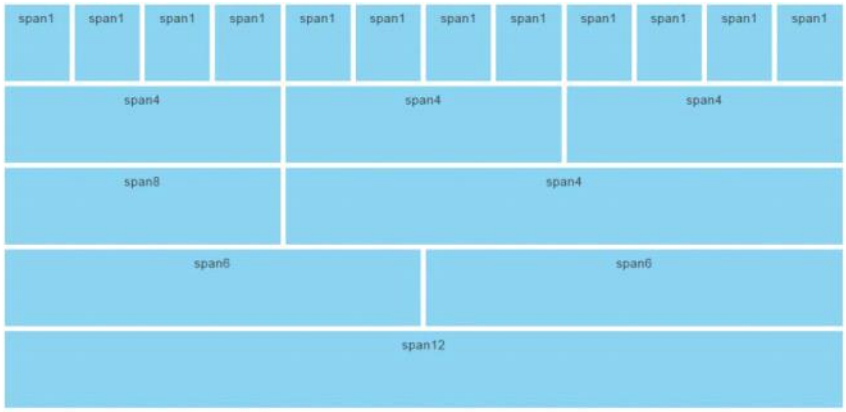
13. Go to the React project folder
14. npm install react-redux
15. npm install @reduxjs/toolkit

16. These will get added in the node_modules
    5. We create a store.js (it will have Reducer --(State + ACTION)
    6. Wrap our <App> in a <Provider> in index.js
    7. Write Component to access State and to Dispatch action

_____

Responsive UI = Screen Layout Changes with changing viewport size automatically
Bootstrap Grid classes are used to provide responsive UI

ViewPort size = mobile , tablet, laptop, desktop



- **.col-xs** (extra small devices - screen width less than 576px)
- **.col-sm-** (small devices - screen width equal to or greater than 576px)
- **.col-md-** (medium devices - screen width equal to or greater than 768px)
- **.col-lg-** (large devices - screen width equal to or greater than 992px)
- **.col-xl-** (xlarge devices - screen width equal to or greater than 1200px)