



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

AY: 2024-25

Class:	TE	Semester:	
Course Code:		Course Name:	WC

Name of Student:	Sainath Khot
Roll No. :	20
Assignment No.:	6
Title of Assignment:	Advanced React
Date of Submission:	
Date of Correction:	

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Completeness	5	3
Demonstrated Knowledge	3	3
Legibility	2	1
Total	10	7

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Completeness	5	3-4	1-2
Demonstrated Knowledge	3	2	1
Legibility	2	1	0

Checked by

Name of Faculty : Kshitija Chharat

Signature : *Kshitija Chharat*

Date : 18/10/24

INC-6

- Q1. Write a functional component that combines use layout effect and use ref to measure and adjust the size of an element after rendering, ensuring the layout is updated synchronously.
- => import React { useLayoutEffect, useRef, useState } from 'react';

```
const ResizableComponent = () => {
  const ref = useRef(null);
  const [size, setSize] = useState({ width: 0, height: 0 });
```

```
  useLayoutEffect(() => {
    const { offsetWidth, offsetHeight } = ref.current;
    setSize({ width: offsetWidth, height: offsetHeight });
  }, [ref]);
```

```
  return (
    <div ref={ref} style={{ width: size.width, height: size.height }} >
      Resized Component
    </div>
  );
}
```

export default ResizableComponent

- Q2) Create a component that uses ReactDOM to render a tooltip or modal outside of its parent DOM hierarchy demonstrating how to manage event & styles.

different layers

→ import React from 'react'
import ReactDOM from 'react-dom';

const Modal = ({ children, onClose }) => {
 return ReactDOM.createPortal(
 <div style = { overlay style } onClick = { onClose } >
 <div style = { modal style } onClick = { e => e.stopPropagation() } >
 { children }
 </div>
 </div>
 document.body);
};

const overlayStyle = { position: 'fixed', top: 0, left: 0, right: 0, bottom: 0, background: 'rgba(0, 0, 0, 0.5)' };
const modalStyle = { background: '#fff', padding: '20px', margin: '100px auto', width: '300px' };
export default Modal;

Q3 Built a functional component that demonstrates controlled vs uncontrolled components, using forward Ref to pass ref to child components & implement an example with both type of input handling

=> import React, { useState, forwardRef } from 'react';

// uncontrolled component using forward Ref = {

<input ref = {ref} placeholder = "Uncontrolled Input" />


```
});
```

```
//controlled component
```

```
const controlled Input = () => {
```

```
  const [value, set value] = use state("");
```

```
  return (
```

```
    <input value = {value} onChange = {e} => set value
```

```
    (e.target.value)} placeholder = "Controlled Input" />
```

```
  );
```

```
}
```

```
return (
```

```
<div>
```

```
<uncontrolled Input ref = {uncontrolled Ref} />
```

```
<controlled Input />
```

```
<button on click = {handle submit} > Submit </button>
```

```
</div>
```

```
);
```

```
}
```

```
export default Parent Component;
```