LAB ASSIGNMENT NO – 09

Aim: Write a program to implement

a. Create React refs

b. How to access Refs

c. Forward Refs

d. Callback Refs

Theory:

React refs provide a way to access and interact with DOM nodes or React components, either by directly accessing the current property or using callback functions. Forwarding refs allows passing refs from parent to child components for direct manipulation, enhancing component interaction. Callback refs facilitate dynamic assignment of refs using functions, enabling flexible handling of refs within components.

a. Create React Refs:

Definition: React refs are a way to access the properties or methods of React elements. They provide a means to reference DOM elements or React components created in a render method.

How to Create: Use React.createRef() or the useRef() hook to create a ref. Example: const myRef = React.createRef(); or const myRef = useRef();.

b. How to Access Refs:

After creating a ref, you can access the DOM node or React element it's associated with using the current property of the ref.

Example: myRef.current provides access to the DOM node or React element.

c. Forward Refs:

Definition: Forwarding refs is a technique in React that allows a parent component to pass down a ref to one of its children, enabling direct access to the child's DOM node or React element.

This is useful when you want to manipulate child components directly from the parent.

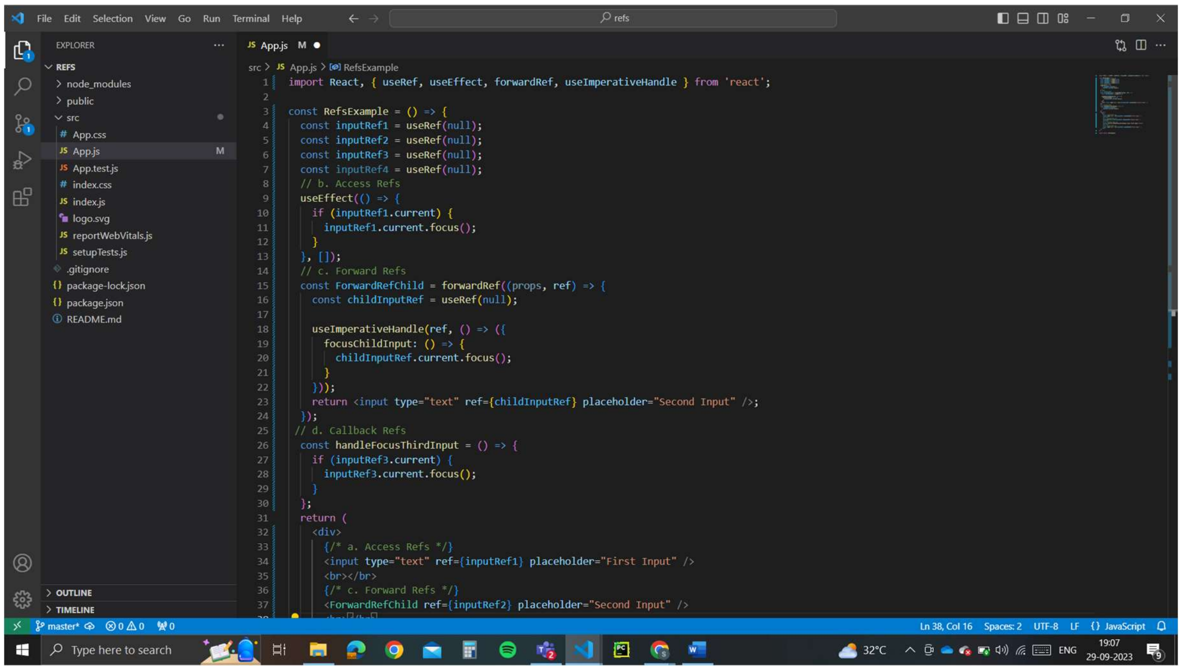
d. Callback Refs:

Definition: Callback refs are a way to access the properties or methods of a DOM node or React element by passing a function as the ref.

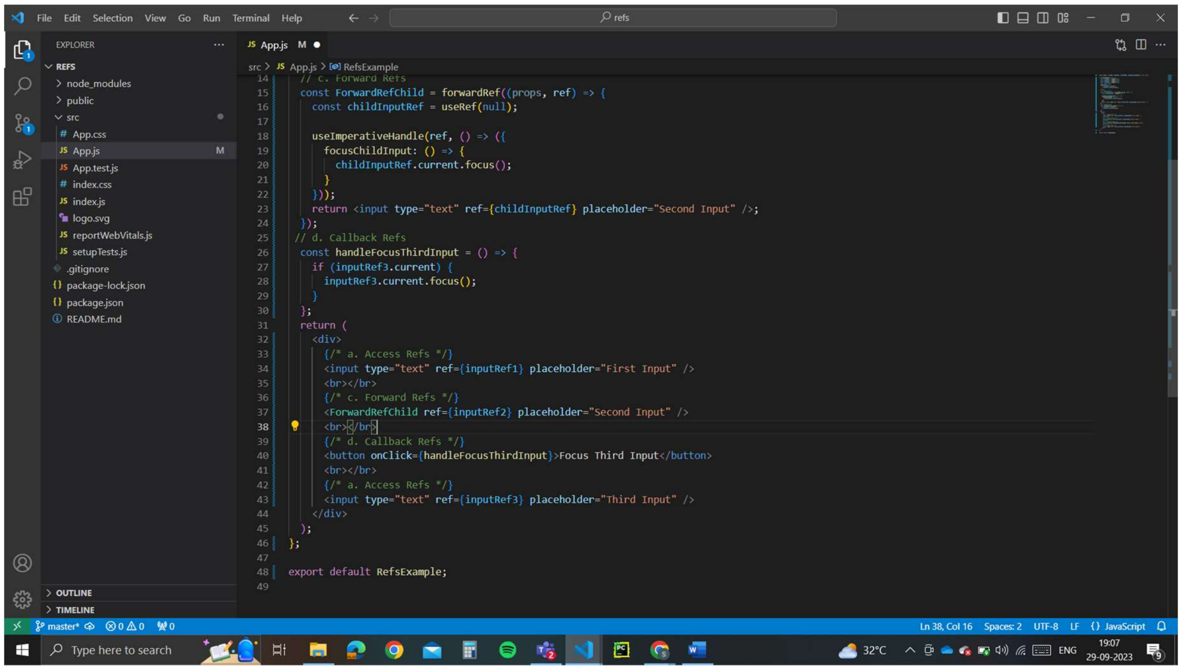
The function is called with the DOM node or React element as an argument when the component mounts or updates.

Conclusion: The experiment with React refs showcased their fundamental role in facilitating direct interaction with DOM elements and components. By leveraging refs, we obtained a means to access and manipulate elements, enhancing user interface control and behavior. This experiment emphasized their versatility, enabling smooth communication between parent and child components.

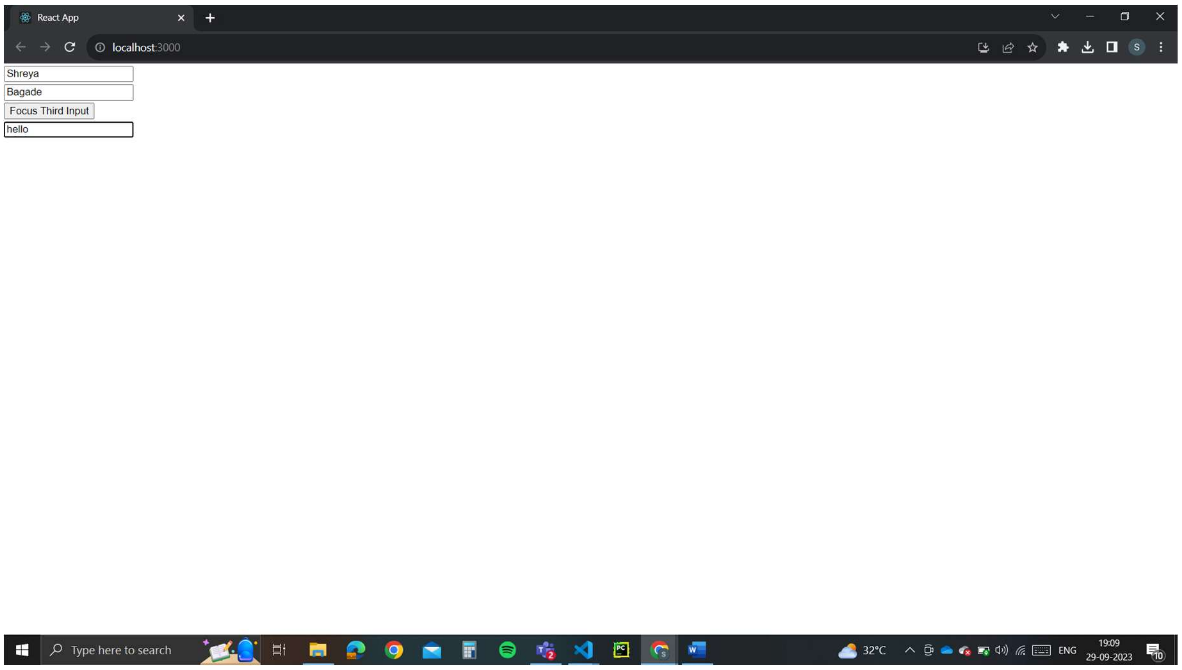
Code:



```
import React, { useRef, useEffect, forwardRef, useImperativeHandle } from 'react';

const RefsExample = () => {
  const inputRef1 = useRef(null);
  const inputRef2 = useRef(null);
  const inputRef3 = useRef(null);
  const inputRef4 = useRef(null);
  // b. Access Refs
  useEffect(() => {
    if (inputRef1.current) {
      inputRef1.current.focus();
    }
  }, []);
  // c. Forward Refs
  const ForwardRefChild = forwardRef((props, ref) => {
    const childInputRef = useRef(null);

    useImperativeHandle(ref, () => ({
      focusChildInput: () => {
        childInputRef.current.focus();
      }
    }));
    return <input type="text" ref={childInputRef} placeholder="Second Input" />;
  });
 // d. Callback Refs
  const handleFocusThirdInput = () => {
    if (inputRef3.current) {
      inputRef3.current.focus();
    }
  };
  return (
    <div>
      {/* a. Access Refs */}
      <input type="text" ref={inputRef1} placeholder="First Input" />
      <br></br>
      {/* c. Forward Refs */}
      <ForwardRefChild ref={inputRef2} placeholder="Second Input" />
```



```
  // c. Forward Refs
  const ForwardRefChild = forwardRef((props, ref) => {
    const childInputRef = useRef(null);

    useImperativeHandle(ref, () => ({
      focusChildInput: () => {
        childInputRef.current.focus();
      }
    }));
    return <input type="text" ref={childInputRef} placeholder="Second Input" />;
  });
 // d. Callback Refs
  const handleFocusThirdInput = () => {
    if (inputRef3.current) {
      inputRef3.current.focus();
    }
  };
  return (
    <div>
      {/* a. Access Refs */}
      <input type="text" ref={inputRef1} placeholder="First Input" />
      <br></br>
      {/* c. Forward Refs */}
      <ForwardRefChild ref={inputRef2} placeholder="Second Input" />
      <br></br>
      {/* d. Callback Refs */}
      <button onClick={handleFocusThirdInput}>Focus Third Input</button>
      <br></br>
      {/* a. Access Refs */}
      <input type="text" ref={inputRef3} placeholder="Third Input" />
    </div>
  );
};

export default RefsExample;
```



```
Shreya
Bagade
Focus Third Input
hello
```

<u>Lab Outcome :</u> LO-5: Construct front end applications using React.