Two approaches create ref :

* Ref could not attach to the functional component. It is attach in class component.

1. createRef

* With CreateRef you create reference using React.createRef()
* In this approach, we attach the reference to an element using the ref prop and assigning the property
* In this approach element can be accessed using this.referebcevariable.current

This.inputRef.current

1. CallbackRef

* it is old approach
* With Callback Ref first create property and then create method that assigns the property with Dom element passed as a parameter
* In callback approach we attach the ref to an element using the method that in turn assign the element to the property
* In this approach element can be accessed directly using this.referencevariable
* this.cbRef

import React, { Component } from 'react'

class Refdemo extends Component {

    constructor(props){

        super(props)

        this.inputRef=React.createRef()

    }

    componentDidMount(){

        this.inputRef.current.focus()

    }

  render() {

    return (

      <div>

        <input type="text" ref={this.inputRef}></input>

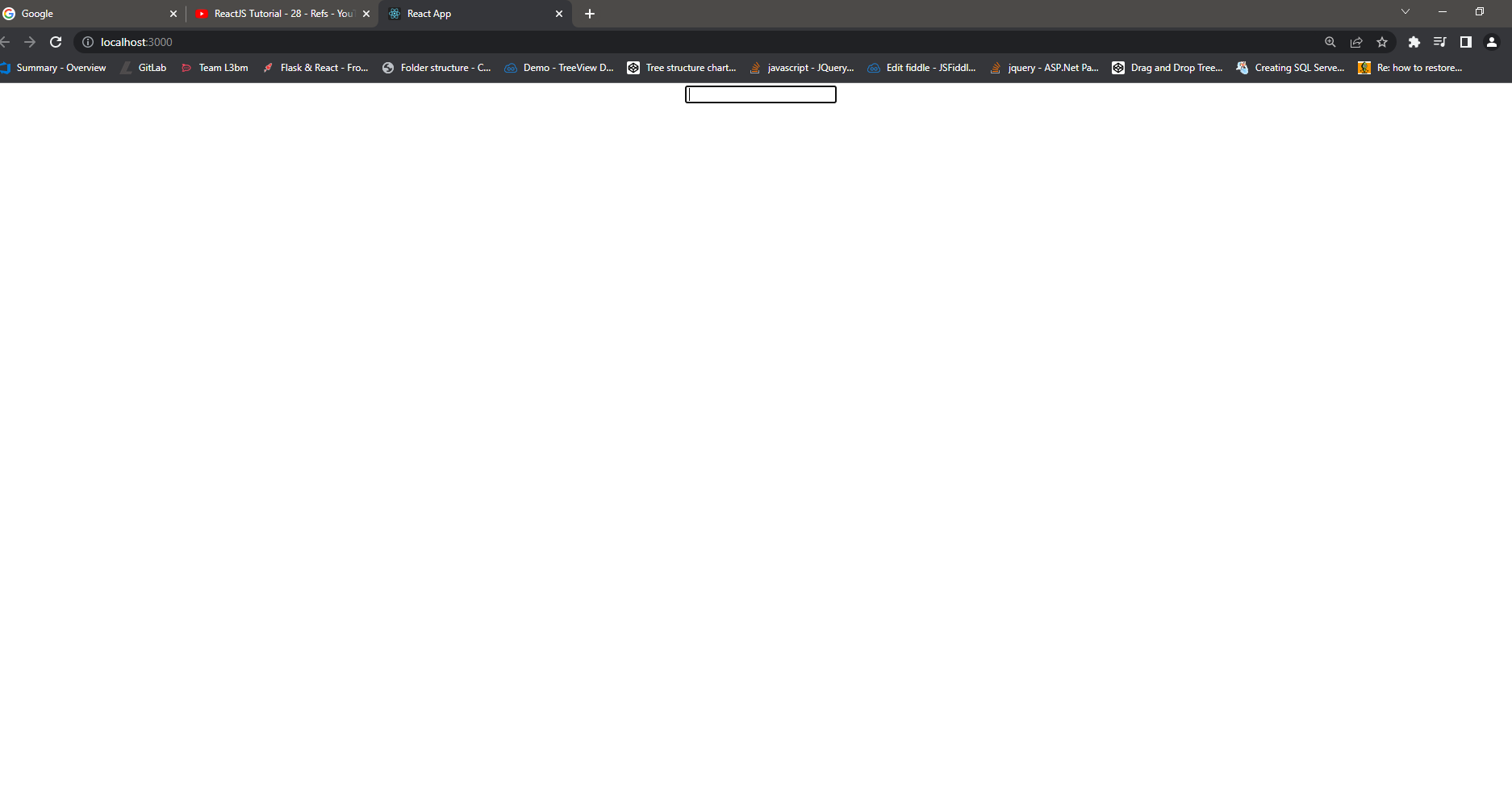
      </div>

    )

  }

}

export default Refdemo



2nd Example :

import React, { Component } from 'react'

class Refdemo extends Component {

    constructor(props){

        super(props)

        this.inputRef=React.createRef()

    }

    componentDidMount(){

        this.inputRef.current.focus()

    }

    clickHandler =() =>{

alert(this.inputRef.current.value)

    }

  render() {

    return (

      <div>

        <input type="text" ref={this.inputRef}></input>

        <button onClick={this.clickHandler}>click</button>

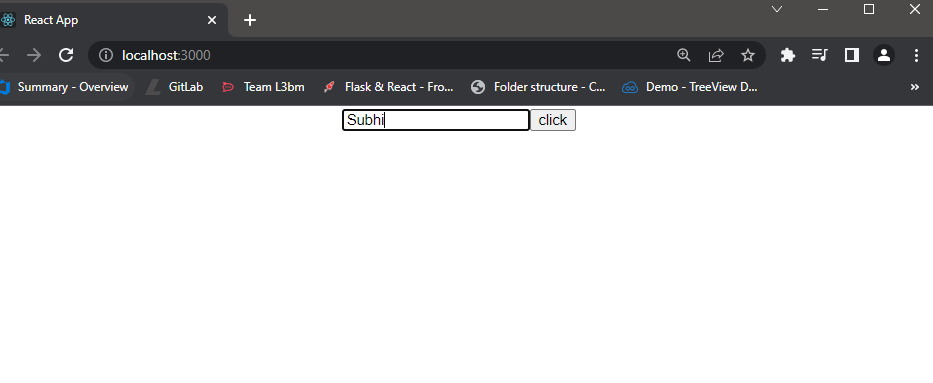
      </div>

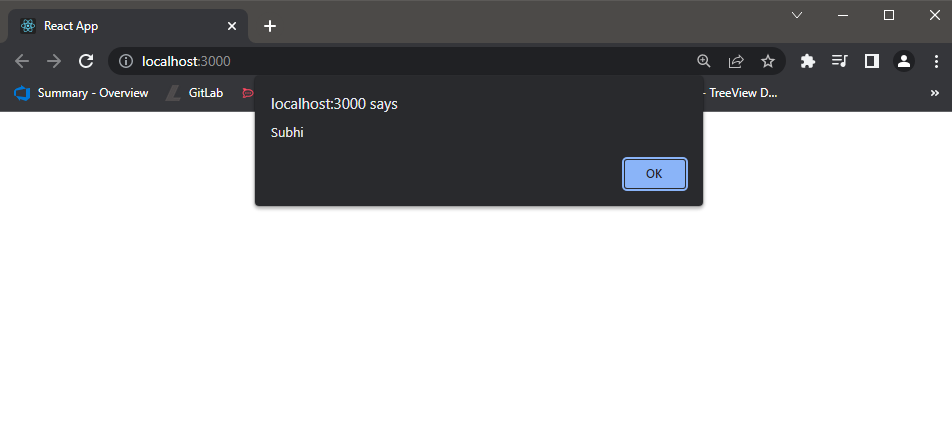
    )

  }

}

export default Refdemo





Old Approach to create ref :

Call back ref :

import React, { Component } from 'react'

 class Callbackrefdemo extends Component {

    constructor(props){

        super(props)

        this.inputRef=React.createRef()

        this.cbRef=null

        this.setCbRef=element=> {

            this.cbRef=element

        }

    }

    componentDidMount(){

        // this.inputRef.current.focus()

        if(this.cbRef){

            this.cbRef.focus()

        }

    }

    clickHandler =() =>{

alert(this.inputRef.current.value)

    }

  render() {

    return (

      <div>

        <input type="text" ref={this.inputRef}></input>

        <input type="text" ref={this.setCbRef}></input>

        <button onClick={this.clickHandler}>click</button>

      </div>

    )

  }

}

export default Callbackrefdemo

