Create a simple App parent comp with methods constructor,getderivedstatefromprops,render(),componentdidmount()

import { useState } from 'react'

import reactLogo from './assets/react.svg'

import viteLogo from '/vite.svg'

import React from 'react'

import './App.css'

export default class App  extends React.Component{

  constructor(){

    super()

    console.log('component A: constructor () method')

  }

  static getDerivedStateFromProps() {

    console.log('component A: getDerivedStateFromProps () method')

  }

  componentDidMount(){

    console.log('Component A : ComponentDidmount method')

  }

  render(){

    console.log('Component A : render() method')

    return(

      <div>

        hello

      </div>

    )

  }

}

**CompB.jsx**

import { useState } from 'react'

import reactLogo from './assets/react.svg'

import viteLogo from '/vite.svg'

import React from 'react'

import './App.css'

export default class CompB  extends React.Component{

  constructor(){

    super()

    console.log('component B: constructor () method')

  }

  static getDerivedStateFromProps() {

    console.log('component B: getDerivedStateFromProps () method')

  }

  componentDidMount(){

    console.log('Component B : ComponentDidmount method')

  }

  render(){

    console.log('Component B : render() method')

    return(

      <div>

        hello compBs

      </div>

    )

  }

}

Call this CompB component inside App,jsx to make it a parent of it,

App.jsx

  render(){

    console.log('Component A : render() method')

    return(

      <div>

        hello

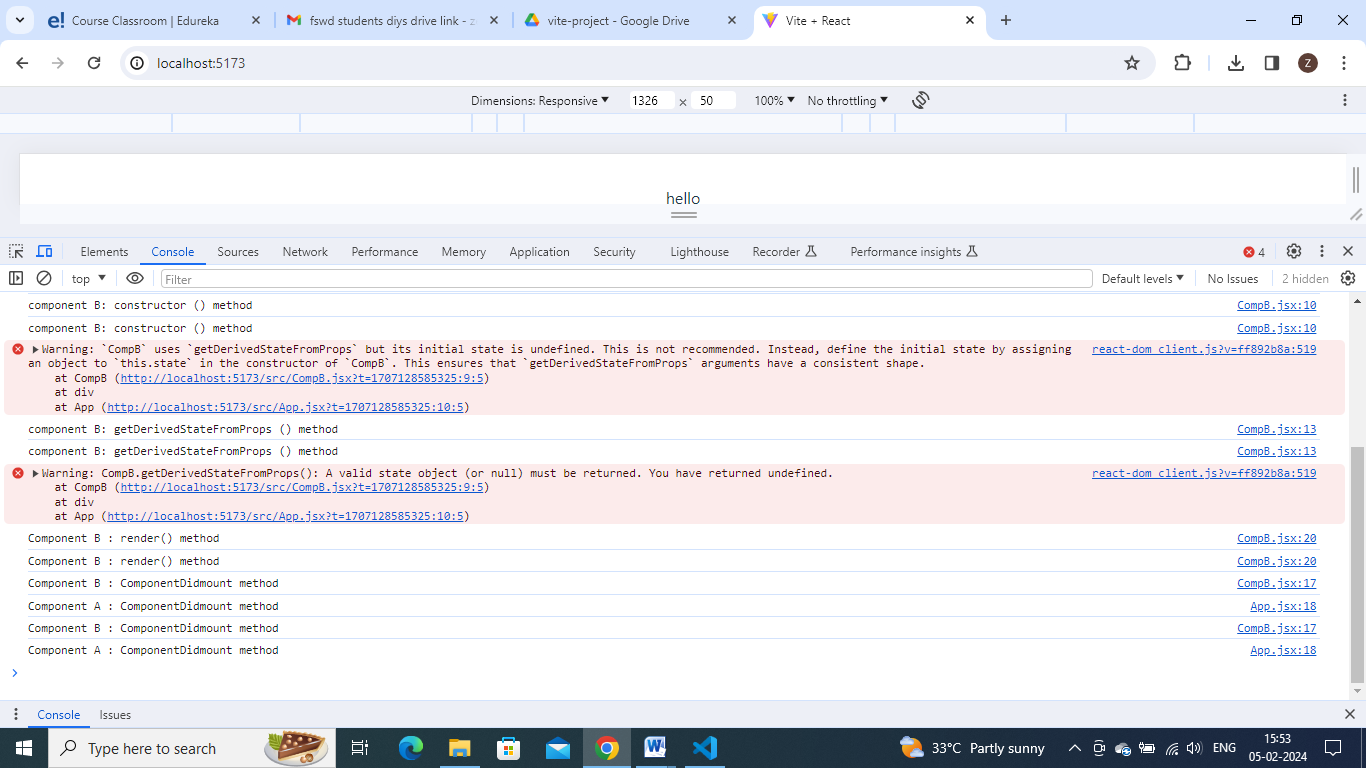
        <CompB/>

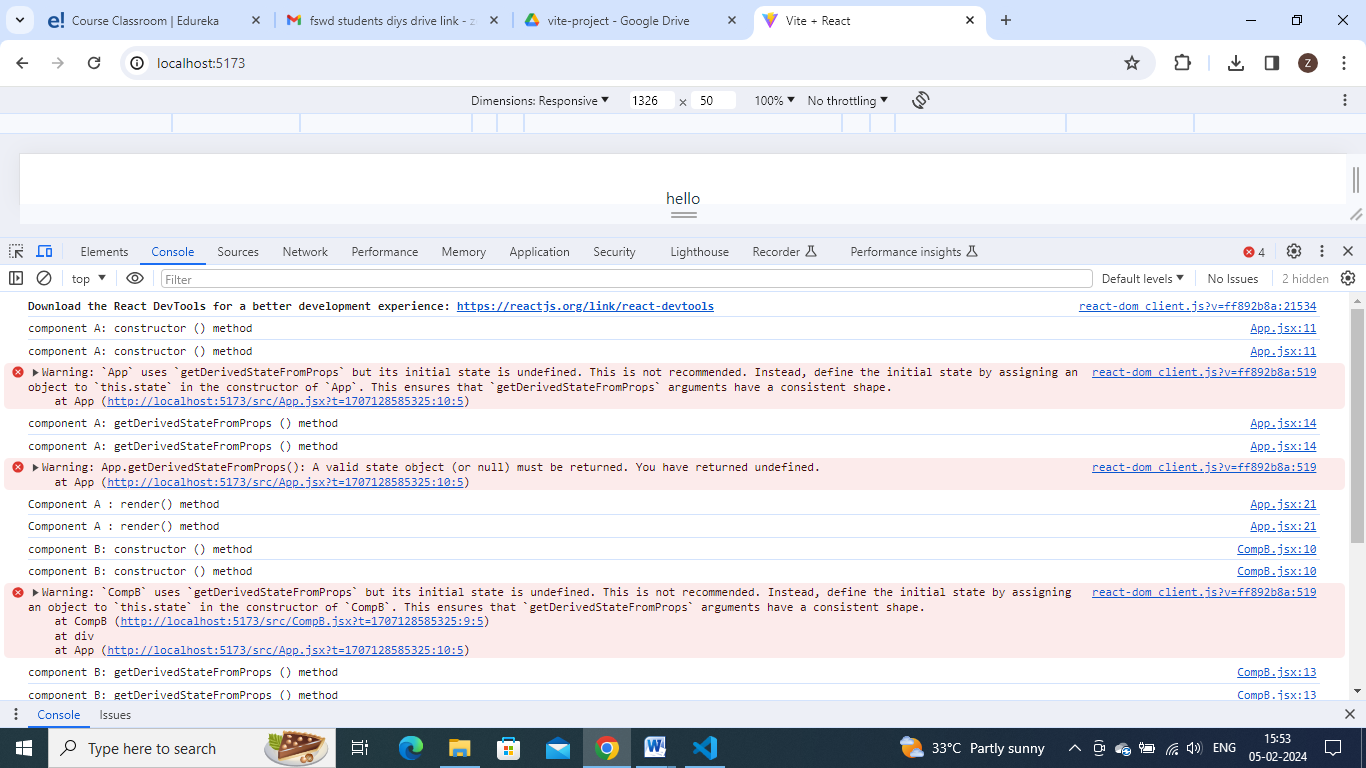
      </div>

    )

  }

}





2)

Here we need to show shouldcomponentupdate and componentdidupdate method. So we need to have some state to get updated to get call these methods. So create some state

Create state in child CompB.js

// import { useState } from 'react'

// import reactLogo from './assets/react.svg'

// import viteLogo from '/vite.svg'

// import React from 'react'

// import './App.css'

// export default class CompB  extends React.Component{

//   constructor(){

//     super()

//     console.log('component B: constructor () method')

//   }

//   static getDerivedStateFromProps() {

//     console.log('component B: getDerivedStateFromProps () method')

//   }

//   componentDidMount(){

//     console.log('Component B : ComponentDidmount method')

//   }

//   render(){

//     console.log('Component B : render() method')

//     return(

//       <div>

//         hello compBs

//       </div>

//     )

//   }

// }

import React from 'react'

export default class CompB  extends React.Component{

  constructor(){

    super()

    this.state={

        data:'dummy data'

    }

    console.log('component B: constructor () method')

  }

  static getDerivedStateFromProps() {

    console.log('component B: getDerivedStateFromProps () method')

  }

  componentDidUpdate(){

    console.log('Component B : ComponentDidUpdate method')

  }

  getSnapshotBeforeUpdate(){

    console.log('Component B: getSnapshotBeforeUpdate method')

  }

  shouldComponentUpdate(){

    console.log('Component B: shouldComponentUpdate() method')

  }

  render(){

    console.log('Component B : render() method')

    return(

      <div>

        hello compBs

        <h2>{this.state.data}</h2>

        <button onClick={()=>{this.setState({data:'dummy2'})}} >click2</button>

      </div>

    )

  }

}

App.jsx

import React from 'react'

import CompB from './CompB'

export default class App  extends React.Component{

  constructor(){

    super()

    console.log('component A: constructor () method')

  }

  static getDerivedStateFromProps() {

    console.log('component A: getDerivedStateFromProps () method')

  }

  getSnapshotBeforeUpdate(){

    console.log('Component A: getSnapshotBeforeUpdate method')

  }

  shouldComponentUpdate(){

    console.log('Component A: shouldComponentUpdate() method')

  }

  componentDidUpdate(){

    console.log('Component A : ComponentDidUpdate method')

  }

  render(){

    console.log('Component A : render() method')

    return(

      <div>

        hello

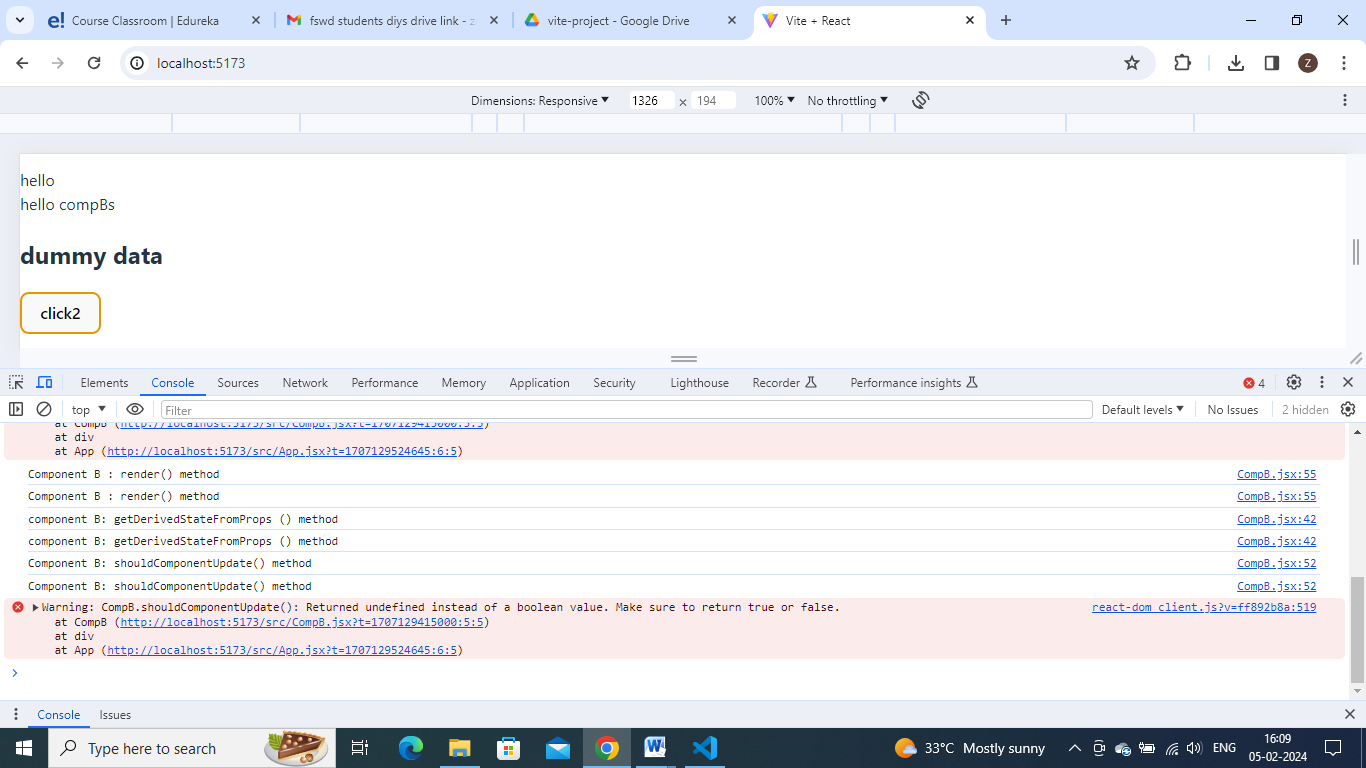
        <CompB />

      </div>

    )

  }

}



Onclicking the button shouldcomponent update getting called.

But componentdidmount is not getting called and data is not updating in ui

Bcoz shouldcomponentupdate () method return false default

So if it returns false – indicated don’t update UI

Retuns true means – indicates update UI

Add return stmt in both components

shouldComponentUpdate(){

    console.log('Component A: shouldComponentUpdate() method')

    return true

  }

