Who am I?

Humera Tariq

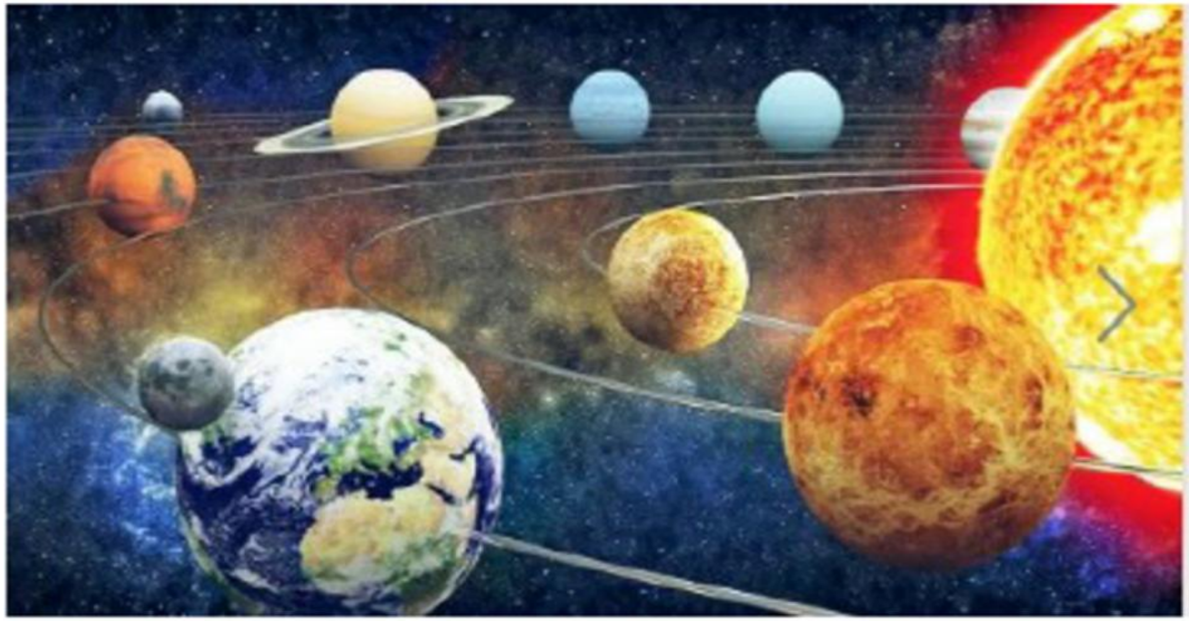
PhD, MS, MCS (Computer Science), B.E (Electrical)

Postdoc (Medical Image Processing, Deep Neural Networks)

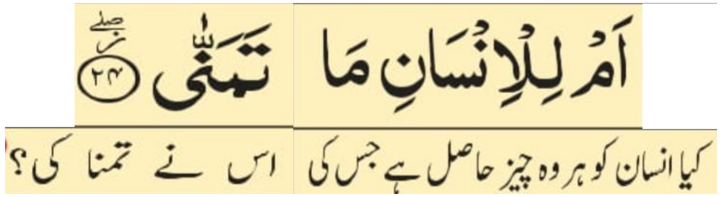
Email: humera@uok.edu.pk
Web: https://humera.pk/

Discord: https://discord.gg/xeJ68vh9

Starting in the name of Allah,



the most beneficial,
the most merciful.

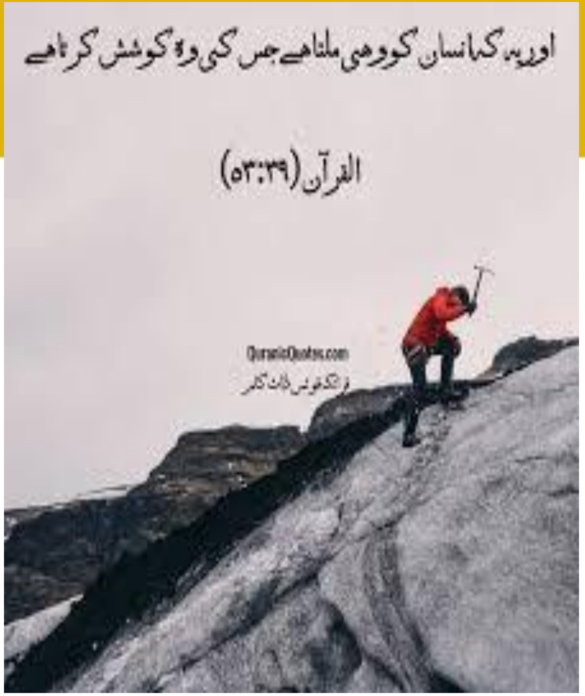


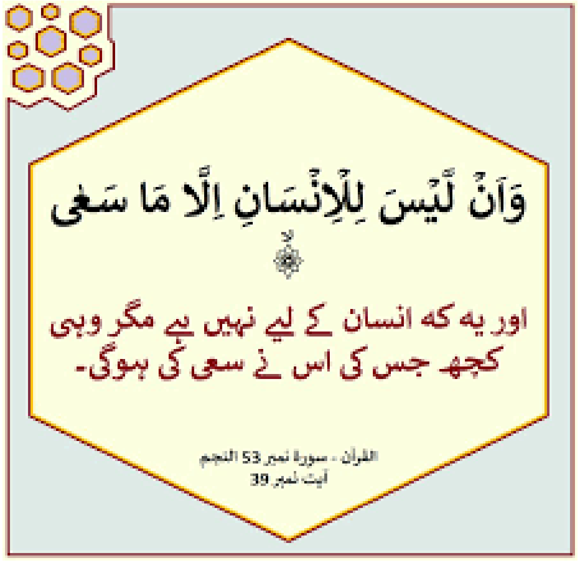
اَمۡ لِلۡاِنۡسَانِ مَا تَمَنّٰی ۫ۖ ﴿۲۴﴾

کیا انسان کو ہر وہ چیز حاصل ہے جس کی اس نے تمنا کی؟

Surah An-Najm Chapter 53 Verse 39

And there is not for man except that [good] for which he strives.




وَاَنْ لَّيْسَ لِلْاِنْسَانِ اِلَّا مَا سَعٰى

اور یہ کہ انسان کے لیے نہیں ہے مگر وہی
کچھ جس کی اس نے سعی کی ہوگی-

القرآن ، سورۃ نمبر 53 النجم
آیت نمبر 39

UNIVERSITY OF
KARACHI

Discussion

Week 08

Internet Application Development

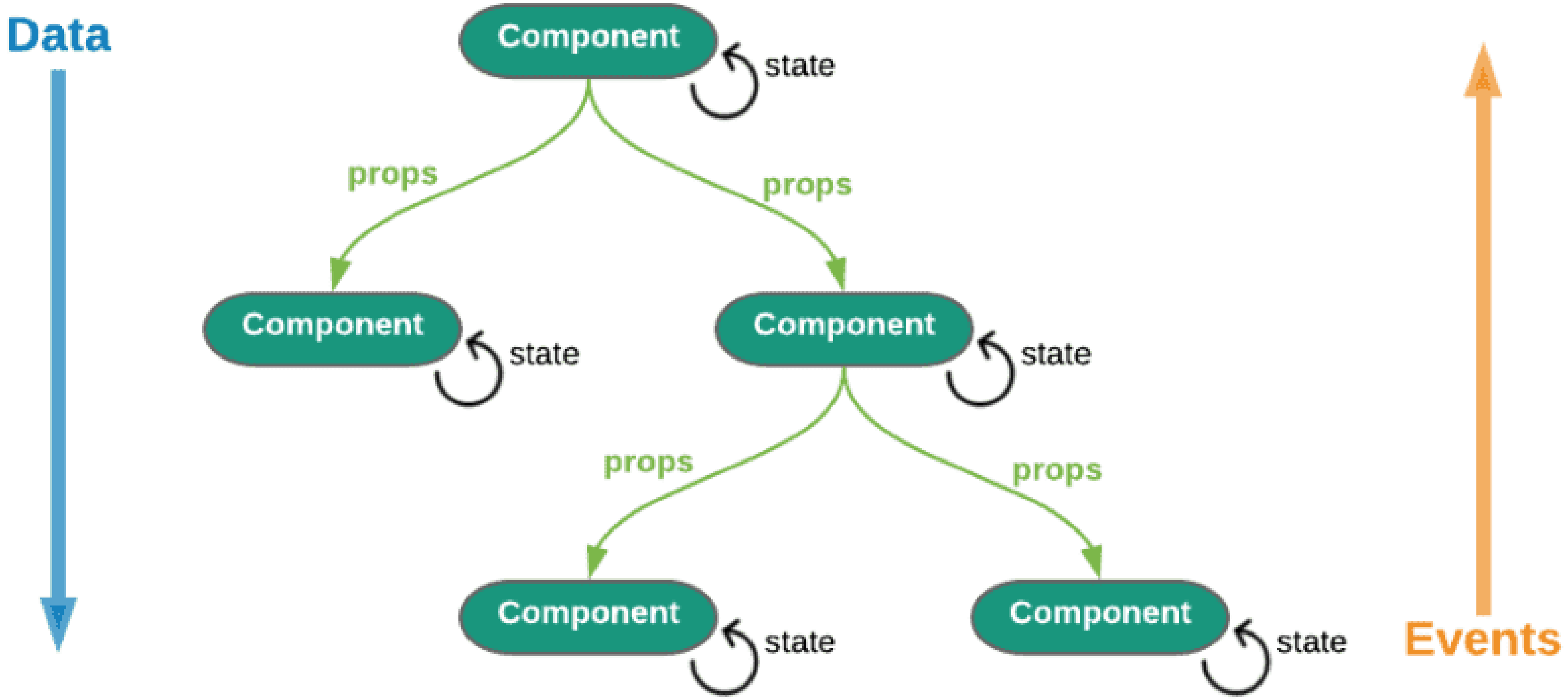
Department of Computer Science (DCS/UBIT)
University of Karachi

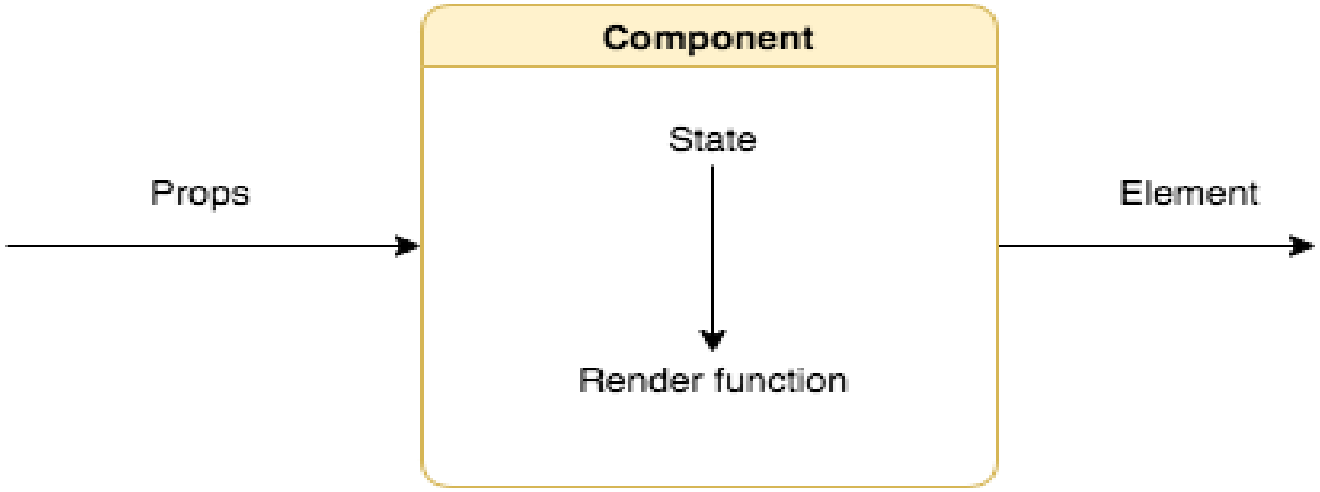
January 2025

React Approach/ Thinking Paradigm

State belongs to the component itself and props are used to pass data to child components.

React Data flow





Component

Props

State

Render function

Element

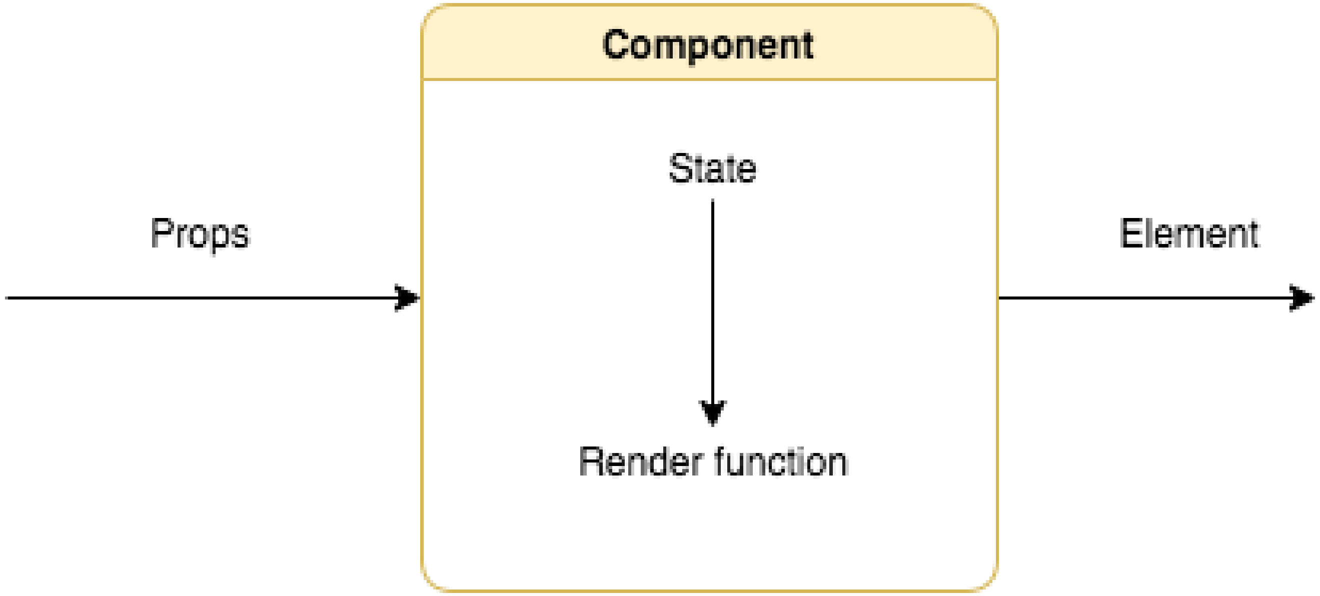
✓ Props are basically__________ that flows from one to another component as a parameters. Props can not be modified means in a component.

✓ React Props are like function arguments in JavaScript *and* attributes in HTML. Component use this data to generate dynamic html elements.

✓ Props are passed to components via ____________________(Hint: JSX/HTML) attributes .

- ✓ React components has a built-in state object which is __________(Hint: public /private) to a component. State can not be accessed from outside of the class. However it can be passed as an argument to another component.

- ✓ Whenever state is changed component calls the render function to render html elements.



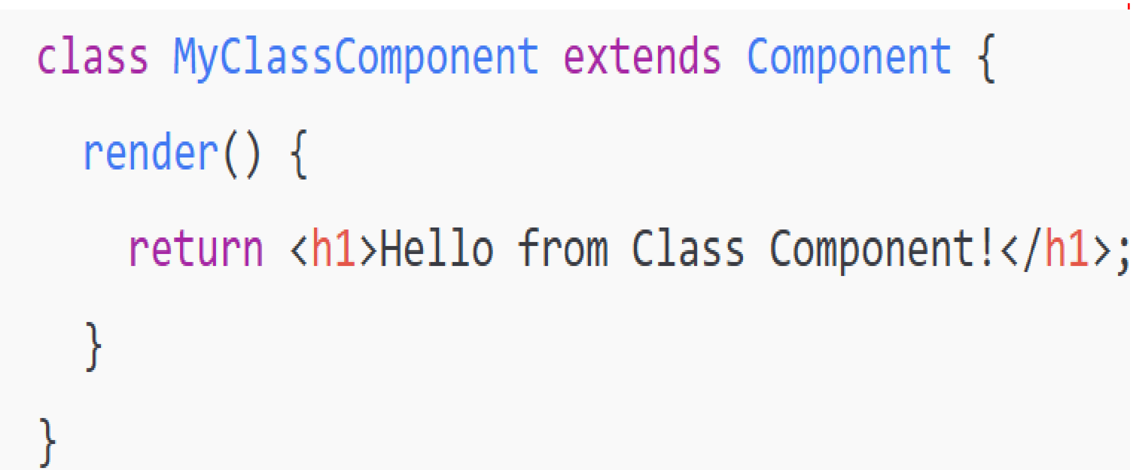
Class vs functional components

Class Component

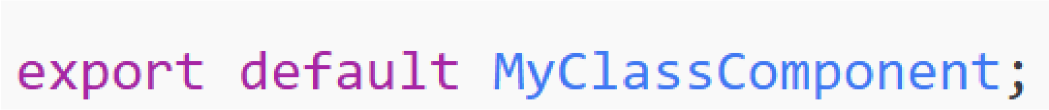
uses

↓

render() method



```
class MyClassComponent extends Component {
  render() {
    return <h1>Hello from Class Component!</h1>;
  }
}
```

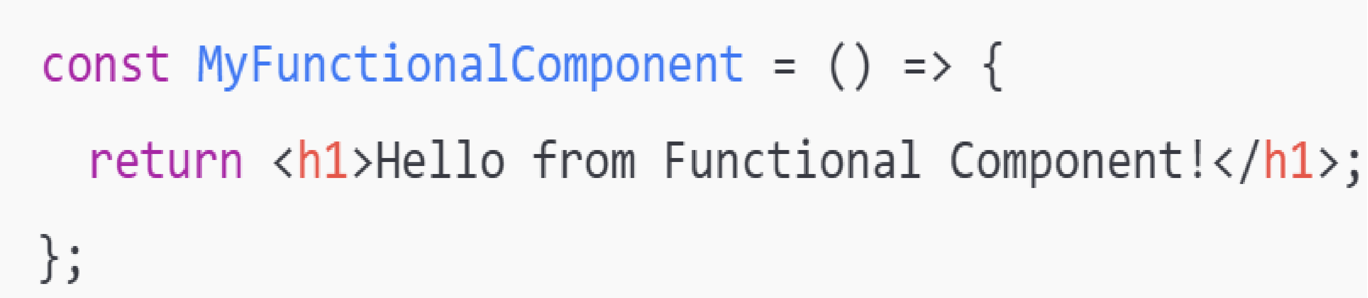


```
export default MyClassComponent;
```

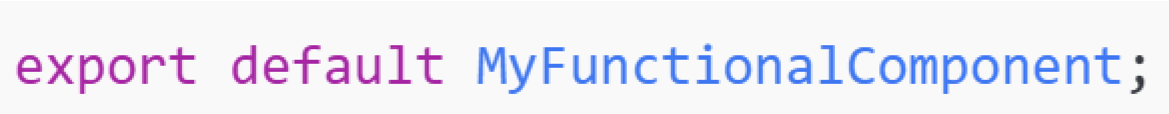
Functional Component

↓

Directly returns JSX



```
const MyFunctionalComponent = () => {
  return <h1>Hello from Functional Component!</h1>;
};
```



```
export default MyFunctionalComponent;
```

UNIVERSITY OF KARACHI

Props	State (Locally scoped, asynchronous)
props are passed in, and they cannot change (____________ /read only)	State can be changed (writable/mutable) using events or lifecycle methods within a component
props can be passed to component from outside	State can be passed to child components from inside a component but as prop
props are ________ (Hint: public/private)	State is _________ . (Hint: public/private)
have better performance	has worse performance

In React we can deal with data in 3 ways

✓Pass data from a parent to a child component by passing props in the child

✓Modify the data from within a component with

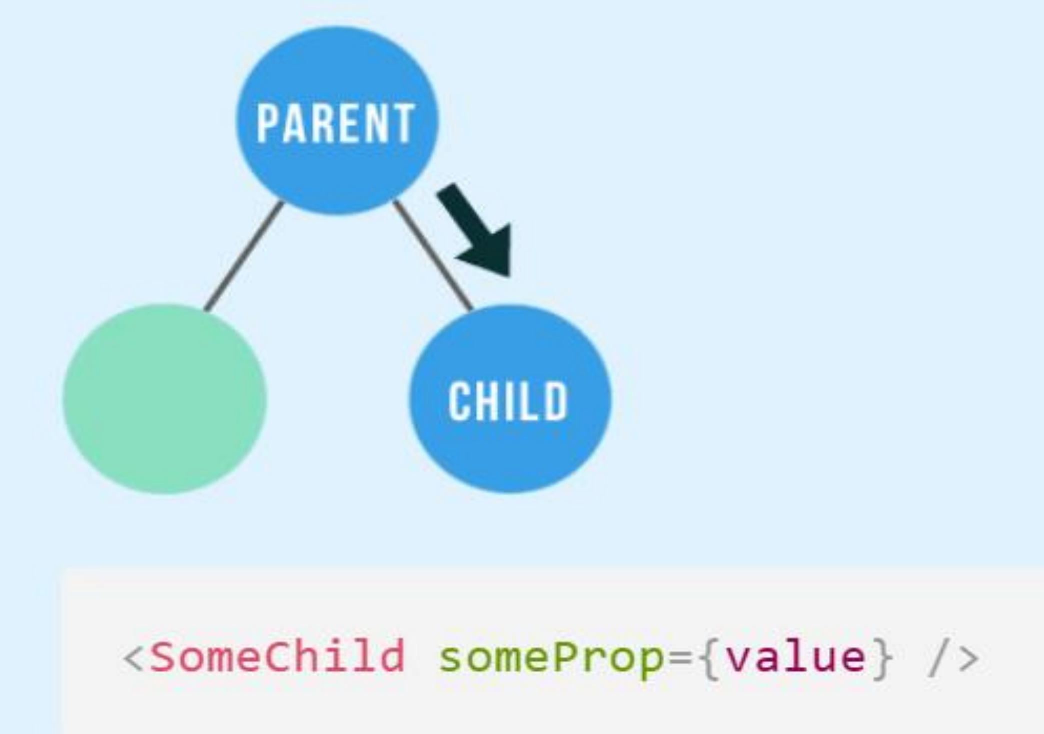
✓ Pass data up from a child component to a parent component by passing a handler function from the parent to the child as a prop.

UNIVERSITY OF KARACHI

"establishing communication Between parent and child"

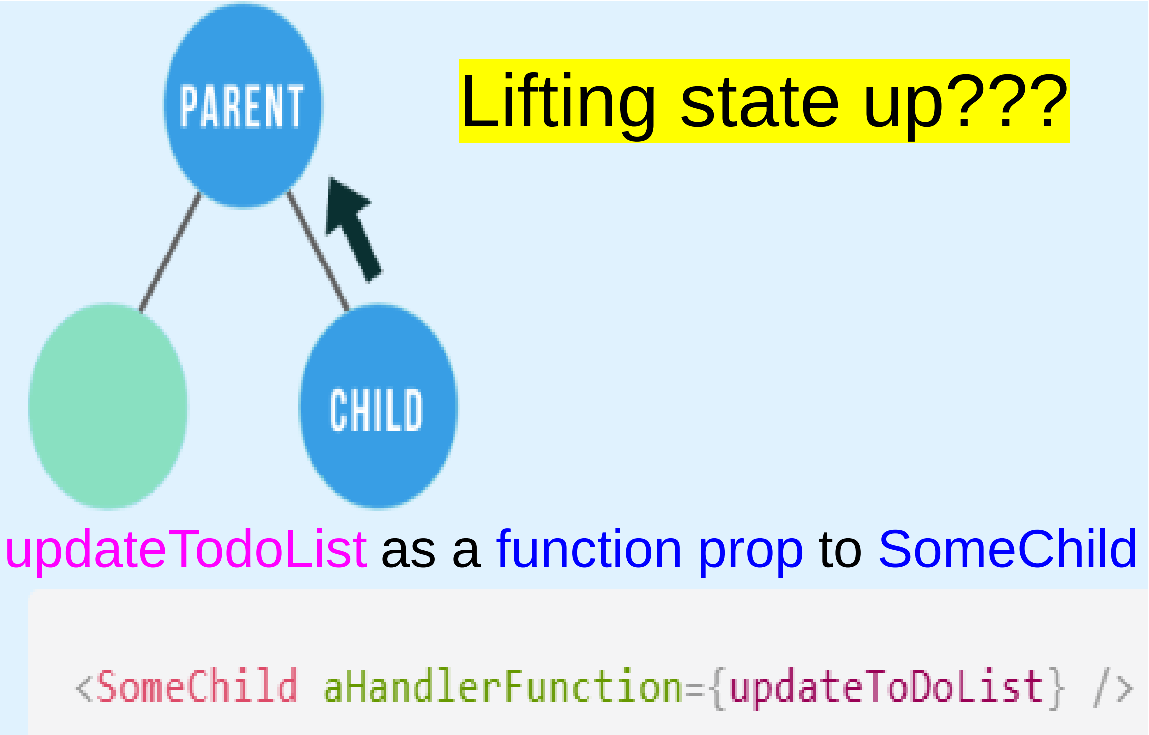
One way or two-way ??

Within component : setState({someStateKey: "someValue"});



```
<SomeChild someProp={value} />
```

- **Sending someProp as an attribute to SomeChild.**
- **Passing data to SomeChild via someProp**



Lifting state up???

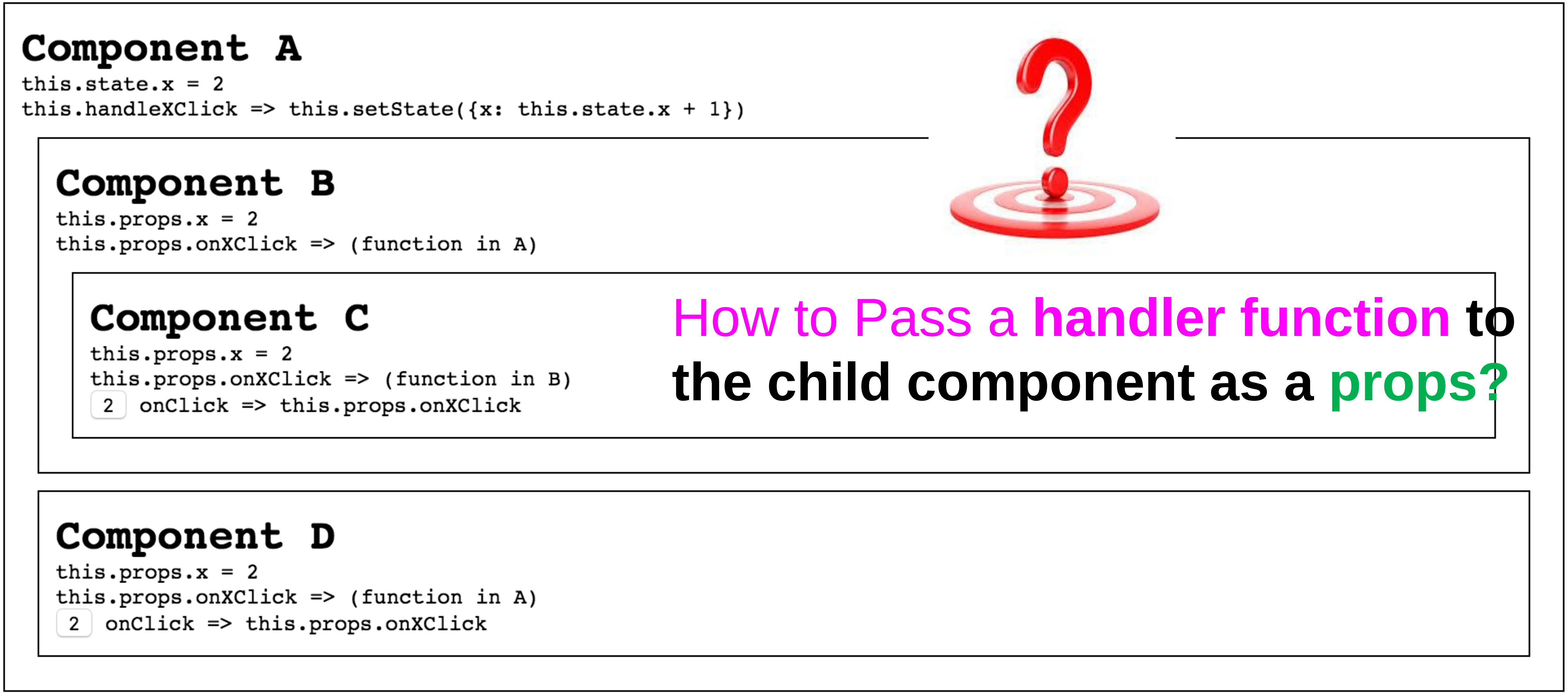
updateTodoList as a function prop to SomeChild

```
<SomeChild aHandlerFunction={updateToDoList} />
```

Passing updateTodoList for event handling
in SomeChild

Parent → Child

- **Definition**: The parent **owns the state** and passes it down as props to child components.
- **Example**: A has x and shares it with B, C, and D.



Component A

this.state.x = 2
this.handleXClick => this.setState({x: this.state.x + 1})

Component B

this.props.x = 2
this.props.onXClick => (function in A)

Component C

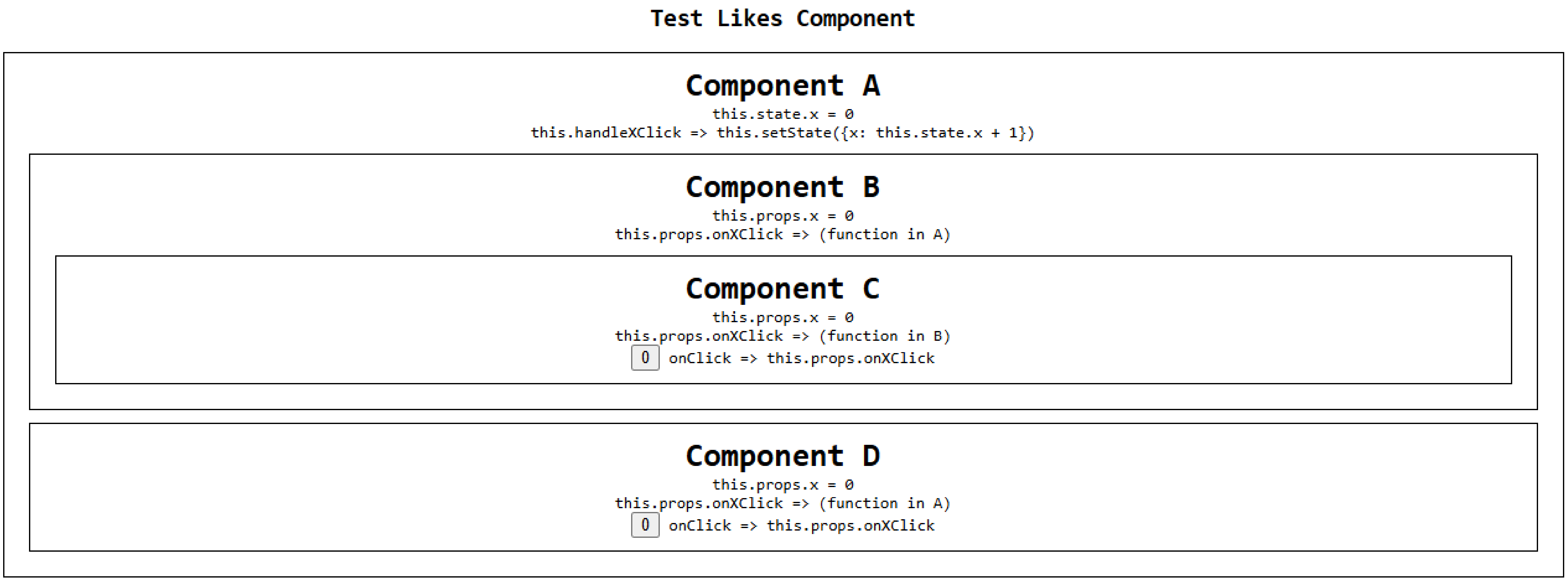
this.props.x = 2
this.props.onXClick => (function in B)
2 onClick => this.props.onXClick

Component D

this.props.x = 2
this.props.onXClick => (function in A)
2 onClick => this.props.onXClick

How to Pass a **handler function** to **the child component as a props?**

What should index.css hold if my UI looks like following?



Test Likes Component

Component A

this.state.x = 0

this.handleXClick => this.setState({x: this.state.x + 1})

Component B

this.props.x = 0

this.props.onXClick => (function in A)

Component C

this.props.x = 0

this.props.onXClick => (function in B)

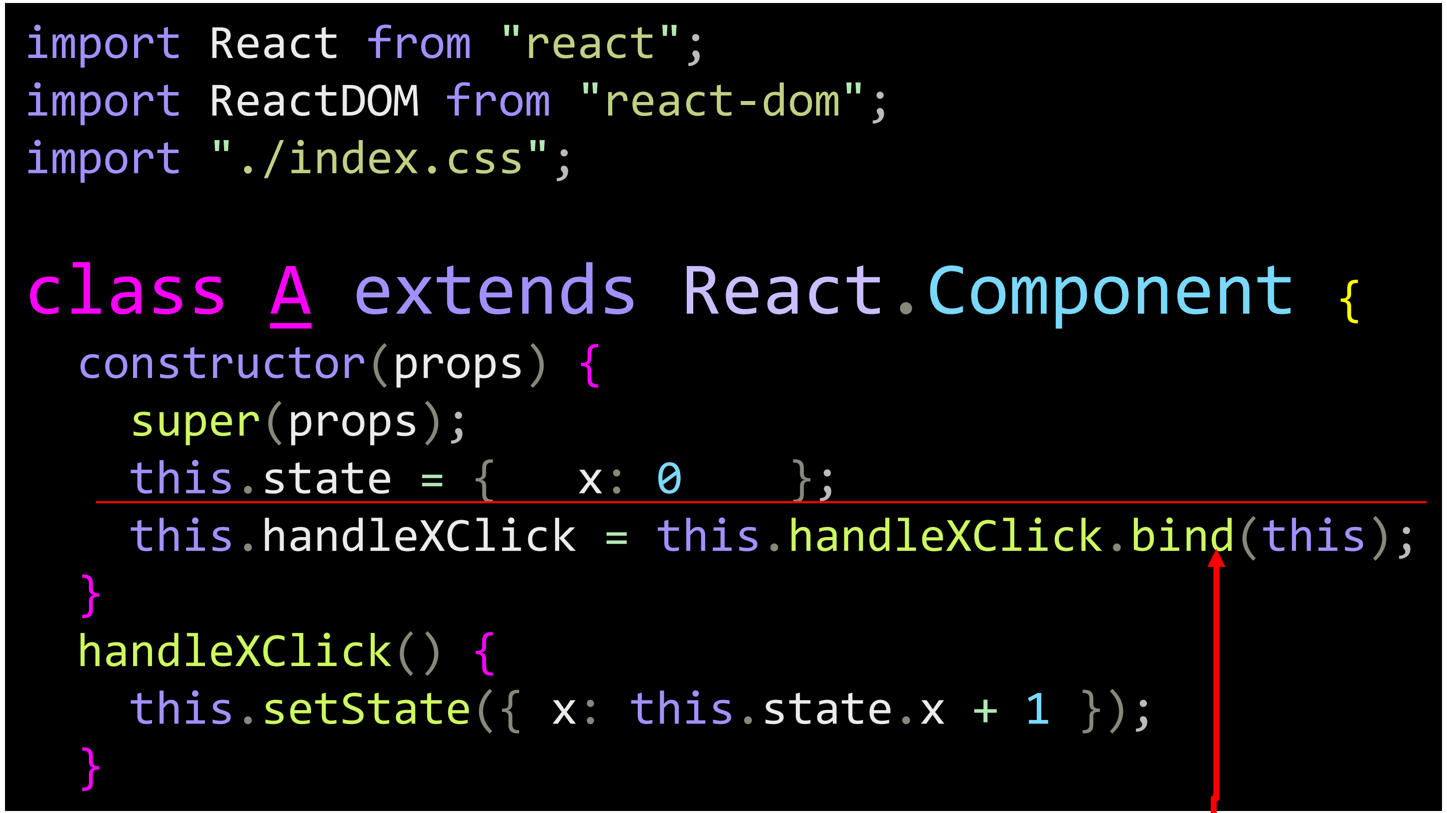
0 onClick => this.props.onXClick

Component D

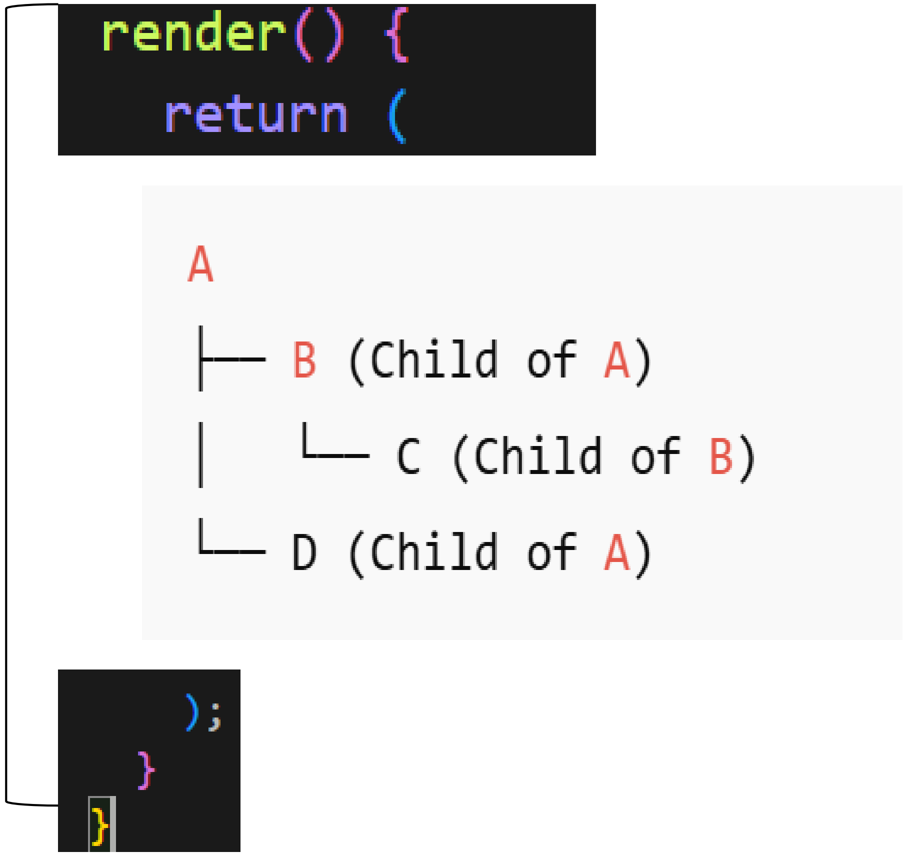
this.props.x = 0

this.props.onXClick => (function in A)

0 onClick => this.props.onXClick



```
import React from "react";
import ReactDOM from "react-dom";
import "./index.css";

class A extends React.Component {
  constructor(props) {
    super(props);
    this.state = {   x: 0    };
    this.handleXClick = this.handleXClick.bind(this);
  }
  handleXClick() {
    this.setState({ x: this.state.x + 1 });
  }
```

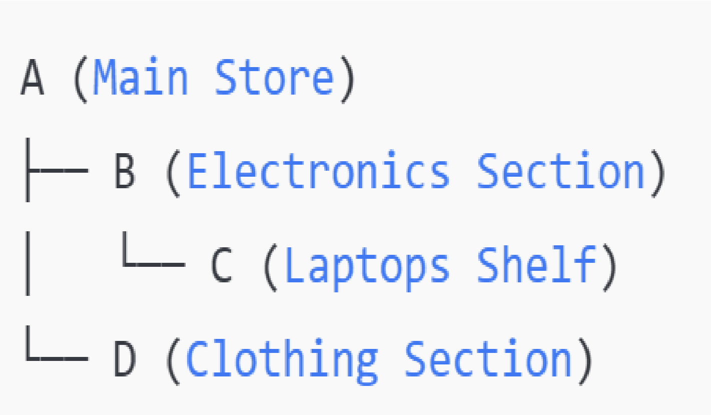


```
  render() {
    return (

A
├── B (Child of A)
│   └── C (Child of B)
└── D (Child of A)

    );
  }
}
```

JavaScript's this is **not automatically** **bound** inside ____________(Hint: class/object) methods.

The Problem of this in JavaScript

Why is bind(this) needed?



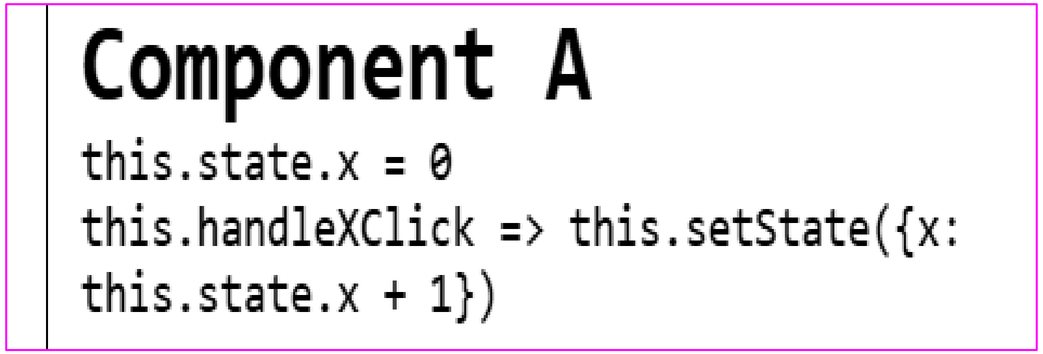
```
A (Main Store)
├── B (Electronics Section)
│    └── C (Laptops Shelf)
└── D (Clothing Section)
```

A customer walks into the **Electronics Section (B)** and browses laptops (C).

✓ Clicking a button in C (Laptops Shelf) calls handleXClick(), increasing the total customer count (x).

Another customer enters the **Clothing Section (D)** and looks at shirts.

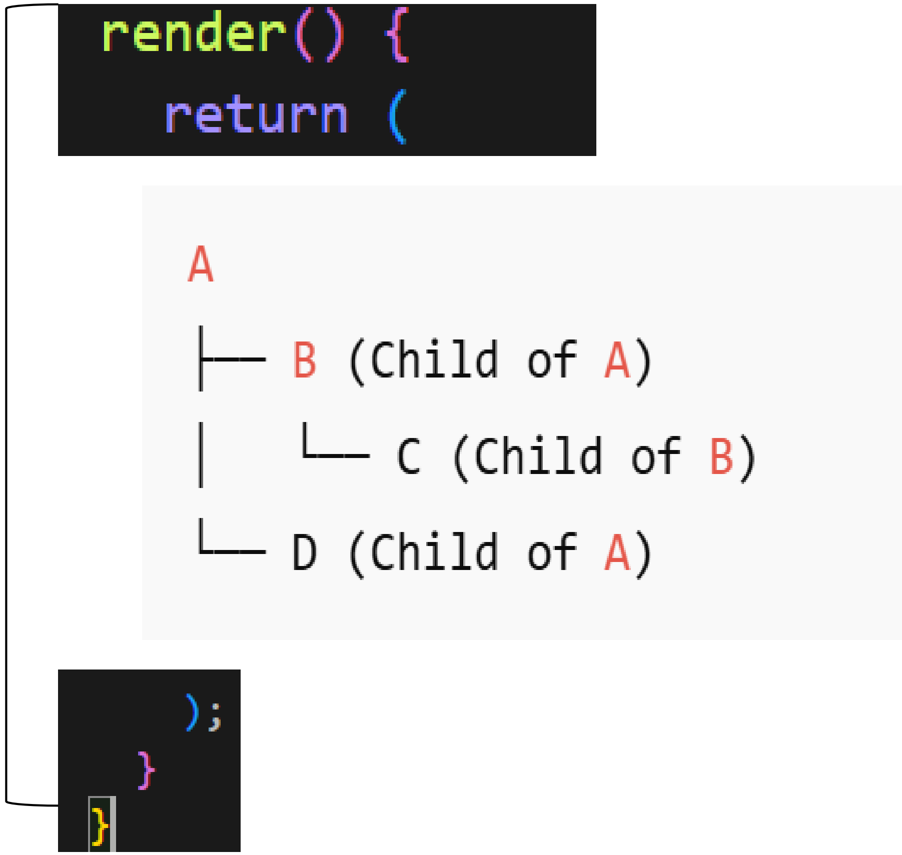
✓ Clicking a button in D also triggers handleXClick(), again increasing x.



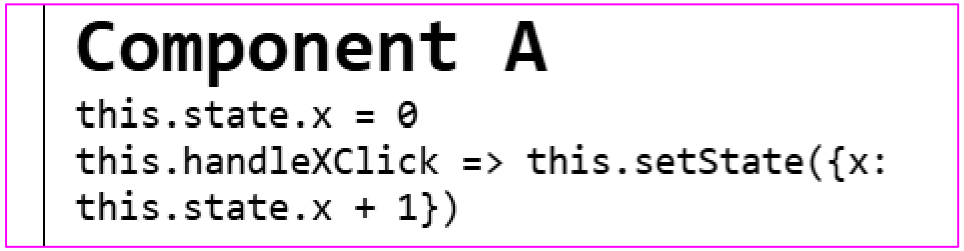
Component A

```
this.state.x = 0
this.handleXClick => this.setState({x:
this.state.x + 1})
```

Can you write render () for component A ?

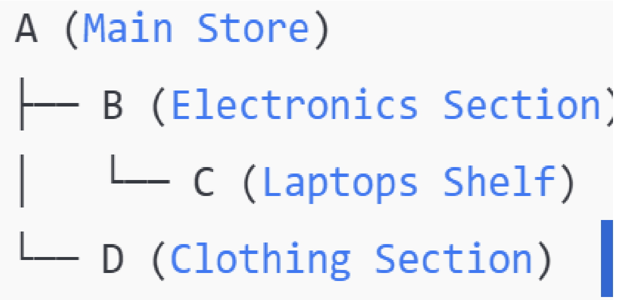


```
render() {
  return (

    A
    ├── B (Child of A)
    │     └── C (Child of B)
    └── D (Child of A)

    );
  }
}
```

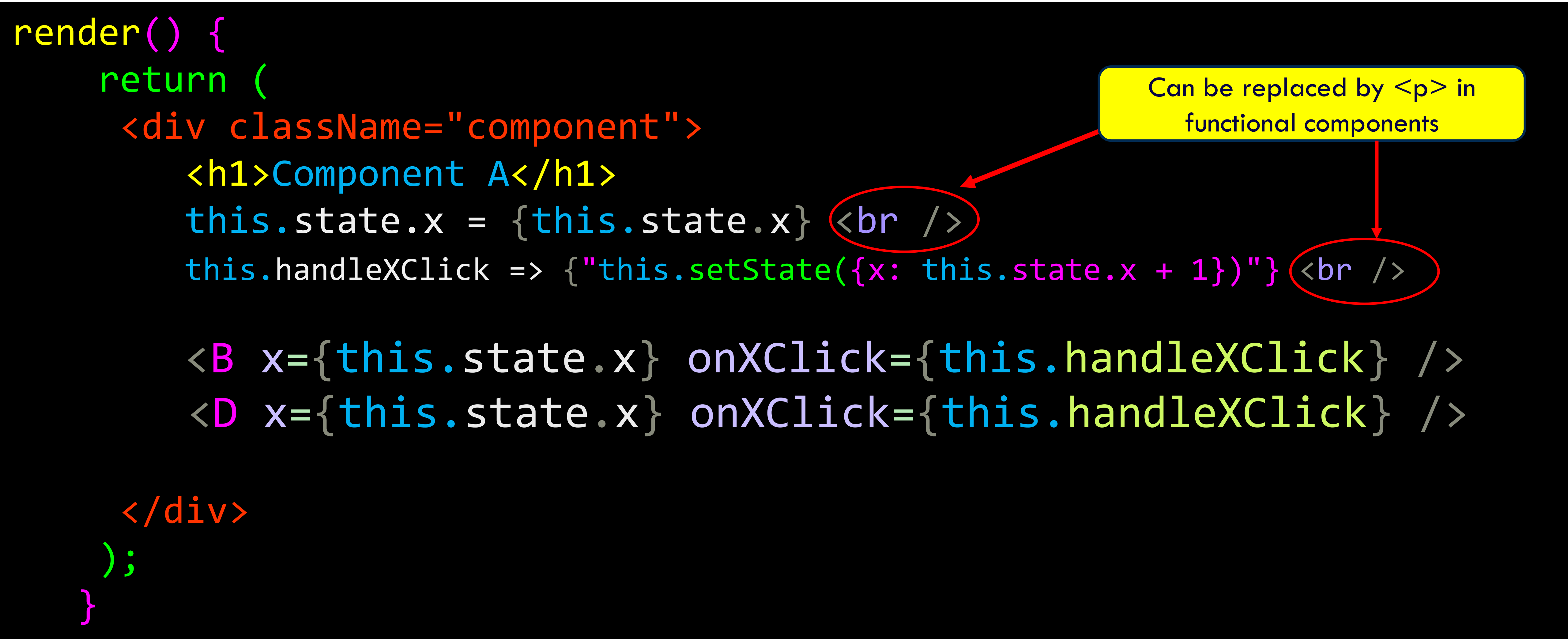


Component A

```
this.state.x = 0
this.handleXClick => this.setState({x:
this.state.x + 1})
```



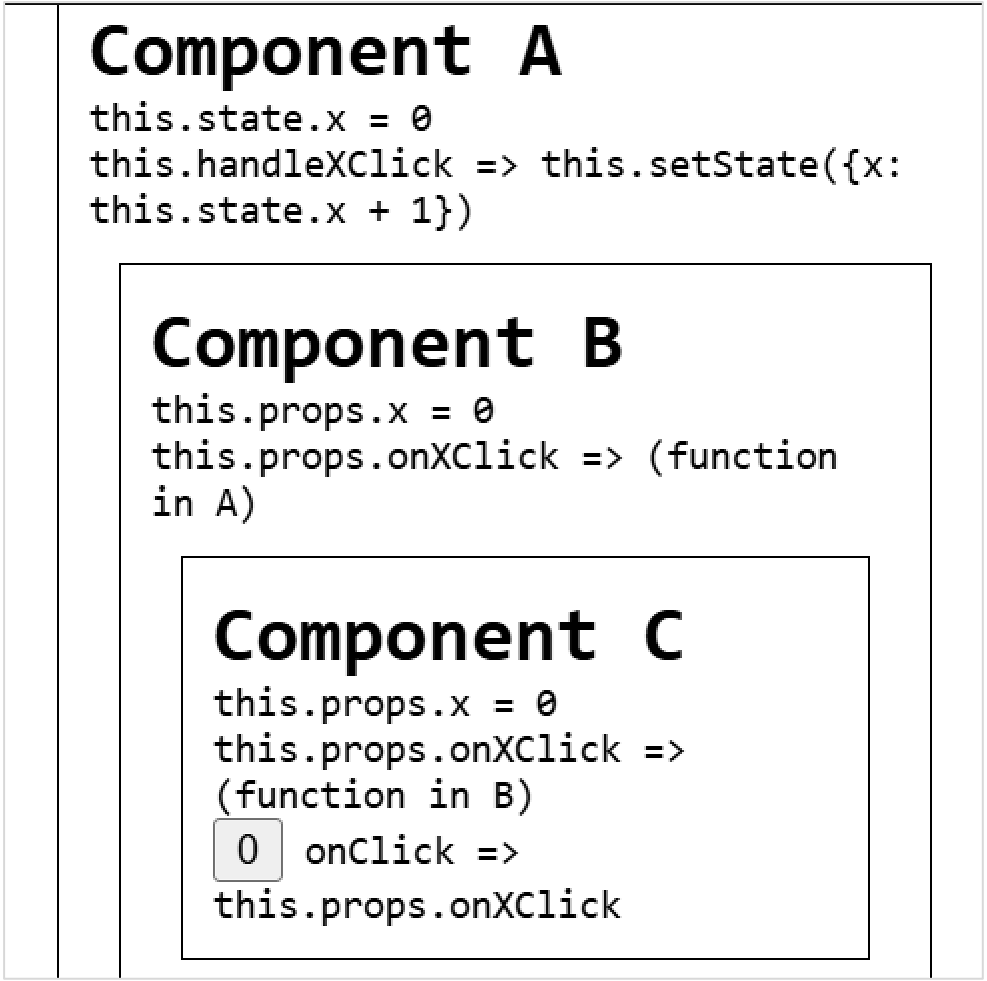
```
A (Main Store)
├── B (Electronics Section)
│    └── C (Laptops Shelf)
└── D (Clothing Section)
```



```
render() {
    return (
     <div className="component">
        <h1>Component A</h1>
        this.state.x = {this.state.x} <br />
        this.handleXClick => {"this.setState({x: this.state.x + 1})"} <br />

        <B x={this.state.x} onXClick={this.handleXClick} />
        <D x={this.state.x} onXClick={this.handleXClick} />

     </div>
    );
   }
```

Can be replaced by <p> in functional components

Can you write render () for component B ?



Component A

this.state.x = 0
this.handleXClick => this.setState({x: this.state.x + 1})

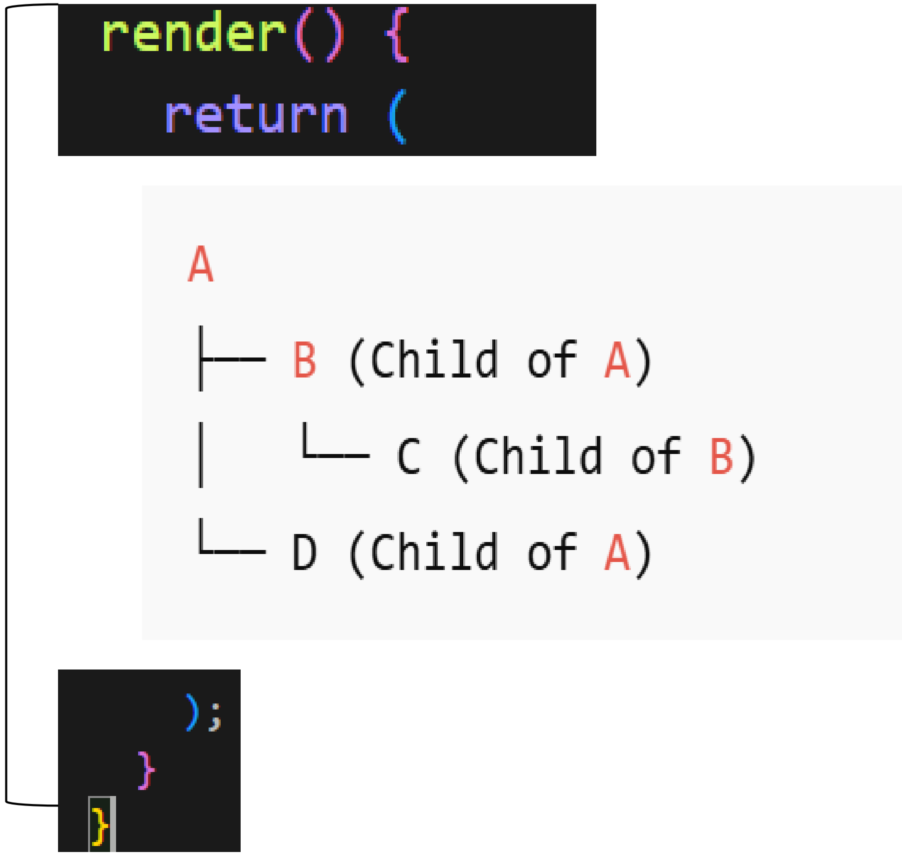
Component B

this.props.x = 0
this.props.onXClick => (function in A)

Component C

this.props.x = 0
this.props.onXClick => (function in B)

0 onClick => this.props.onXClick



```
render() {
  return (
```

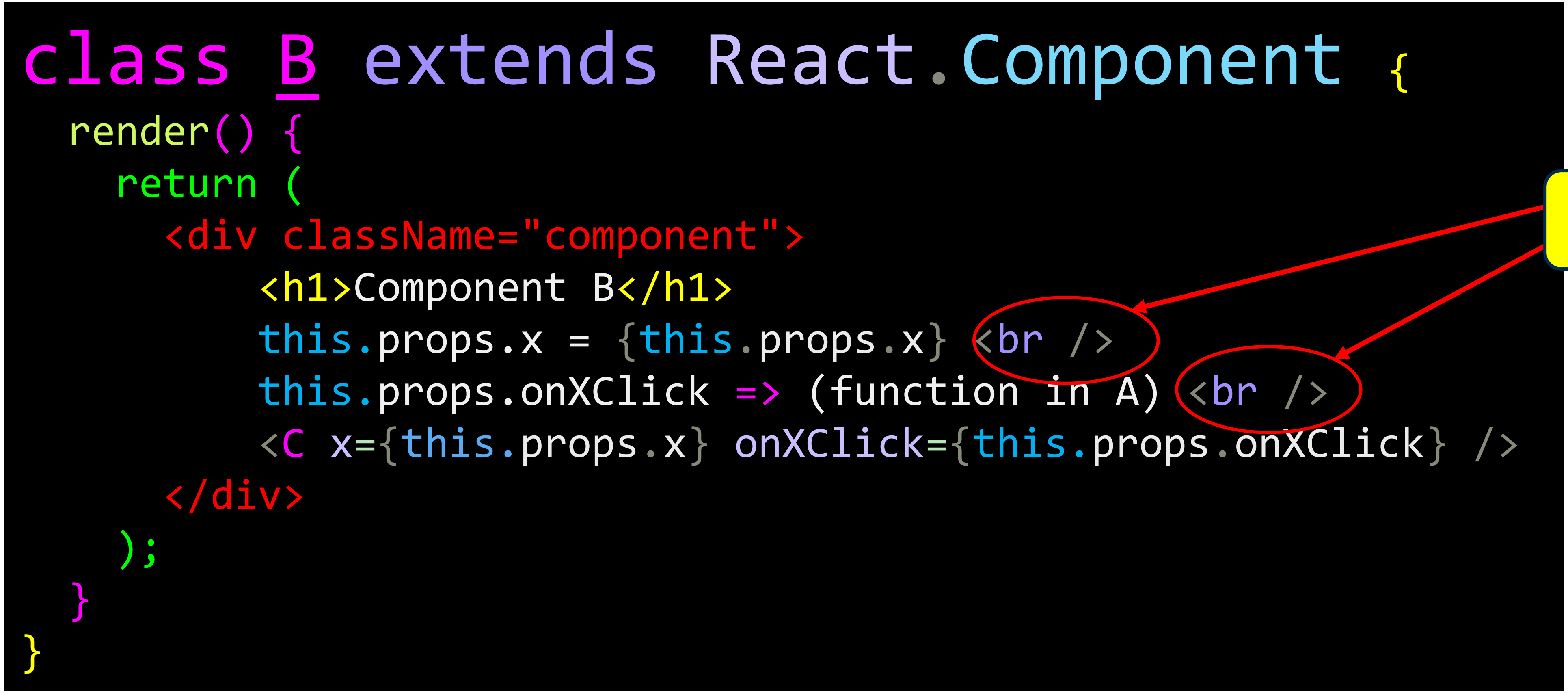
```
A
├── B (Child of A)
│    └── C (Child of B)
└── D (Child of A)
```

```
    );
  }
}
```

- A (Main Store) tracks **total customers (x)**.

- B (Electronics Section) doesn’t track customers but has access to the **store’s customer counter (x)**.

- B has a **"Notify Store" button (onXClick)** that tells the store to **increase total customers (setState)**.



```
class B extends React.Component {
  render() {
    return (
      <div className="component">
          <h1>Component B</h1>
          this.props.x = {this.props.x} <br />
          this.props.onXClick => (function in A) <br />
          <C x={this.props.x} onXClick={this.props.onXClick} />
      </div>
    );
  }
}
```

Can be replaced by <p> in functional components

Component A

this.state.x = 2

this.handleXClick => this.setState({x: this.state.x + 1})

DONE

Component B

this.props.x = 2

this.props.onXClick => (function in A)

DONE

Component C

this.props.x = 2

this.props.onXClick => (function in B)

2 onClick => this.props.onXClick

Component D

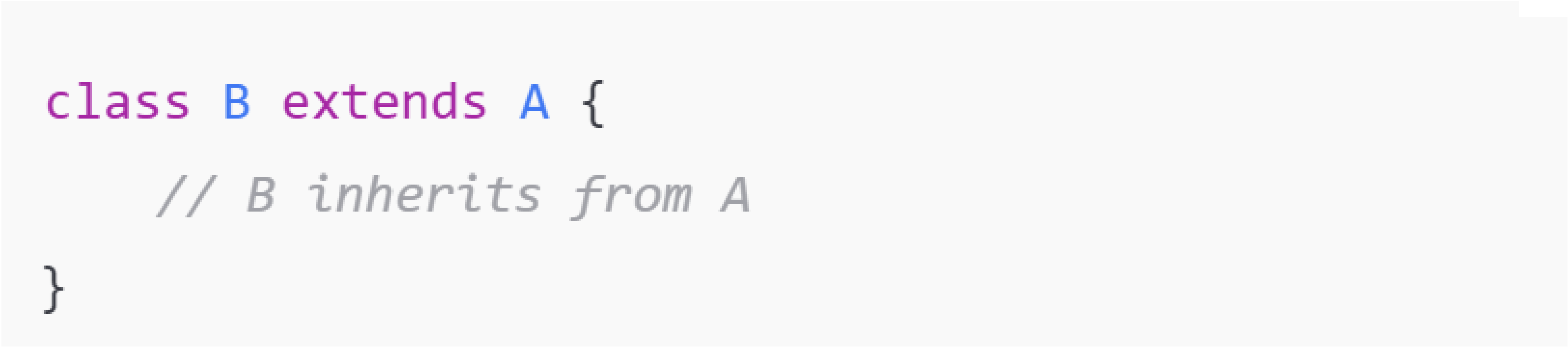
this.props.x = 2

this.props.onXClick => (function in A)

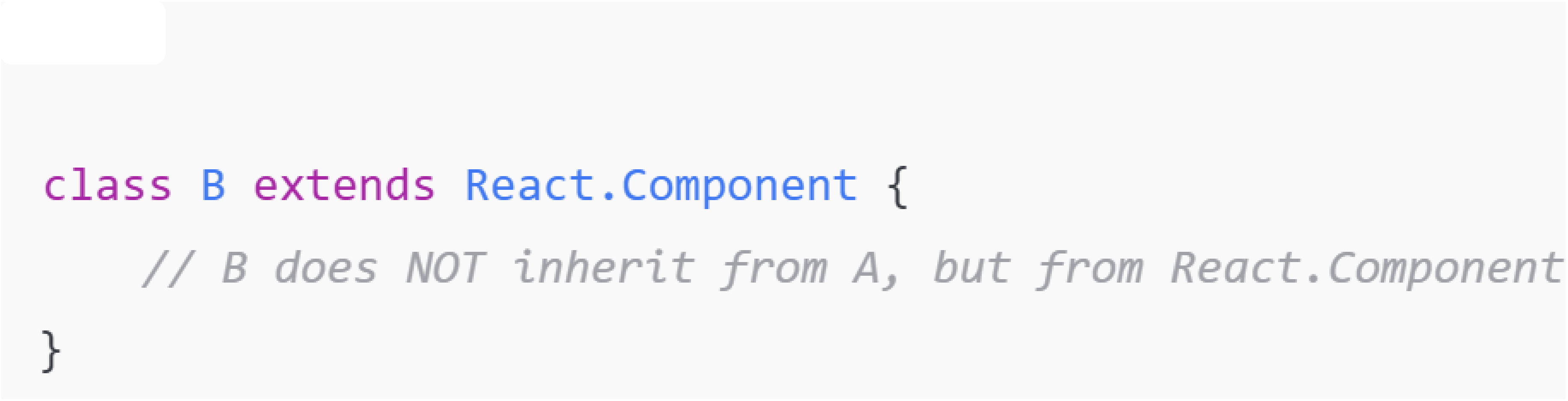
2 onClick => this.props.onXClick

Write 3-4 lines about your Understanding Lifting State Up ??

Which one is correct w.r.t. JSX

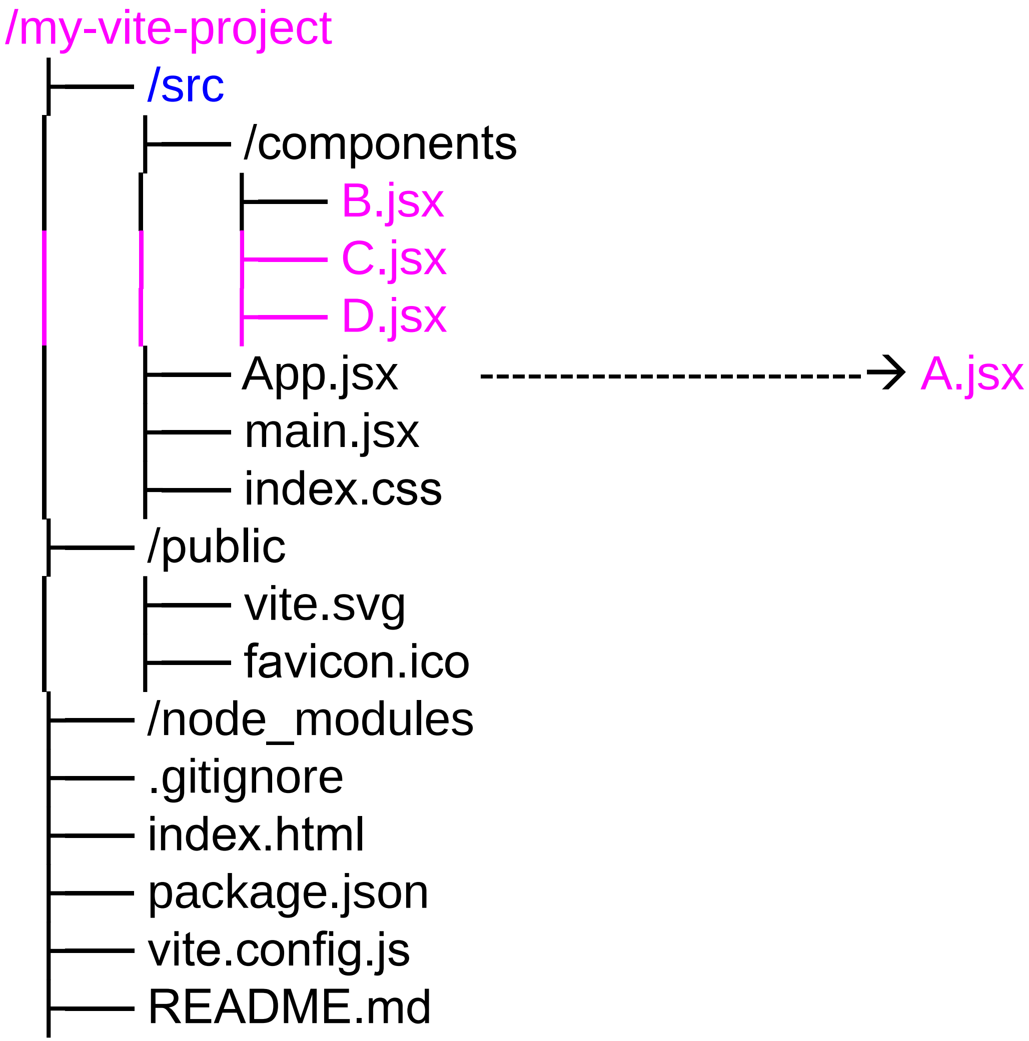


```
class B extends A {
    // B inherits from A
}
```



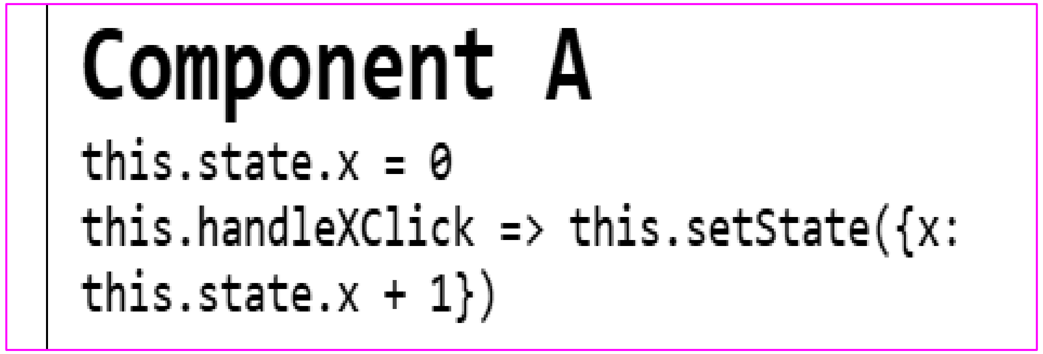
```
class B extends React.Component {
    // B does NOT inherit from A, but from React.Component
}
```

Should We Convert A,B,C,D example to Functional Components?



```
/my-vite-project
├── /src
│   ├── /components
│   │   ├── B.jsx
│   │   ├── C.jsx
│   │   ├── D.jsx
│   ├── App.jsx      ------------------------→ A.jsx
│   ├── main.jsx
│   ├── index.css
├── /public
│   ├── vite.svg
│   ├── favicon.ico
├── /node_modules
├── .gitignore
├── index.html
├── package.json
├── vite.config.js
├── README.md
```

function App() { ....}



Component A

this.state.x = 0

this.handleXClick => this.setState({x: this.state.x + 1})

```
import { useState } from "react";
import B from "./components/B";
import D from "./components/D";

function App() {
  const [x, setX] = useState(0);

  const handleXClick = () => {setX(x + 1);};

  return (
    <div className="component">
      <h1>Component A (App)</h1>
      this.state.x = {x} <br />
      this.handleXClick => {"setX(x + 1)"} <br />
      <B x={x} onXClick={handleXClick} />
      <D x={x} onXClick={handleXClick} />
    </div>
  );
}

export default App;};
```

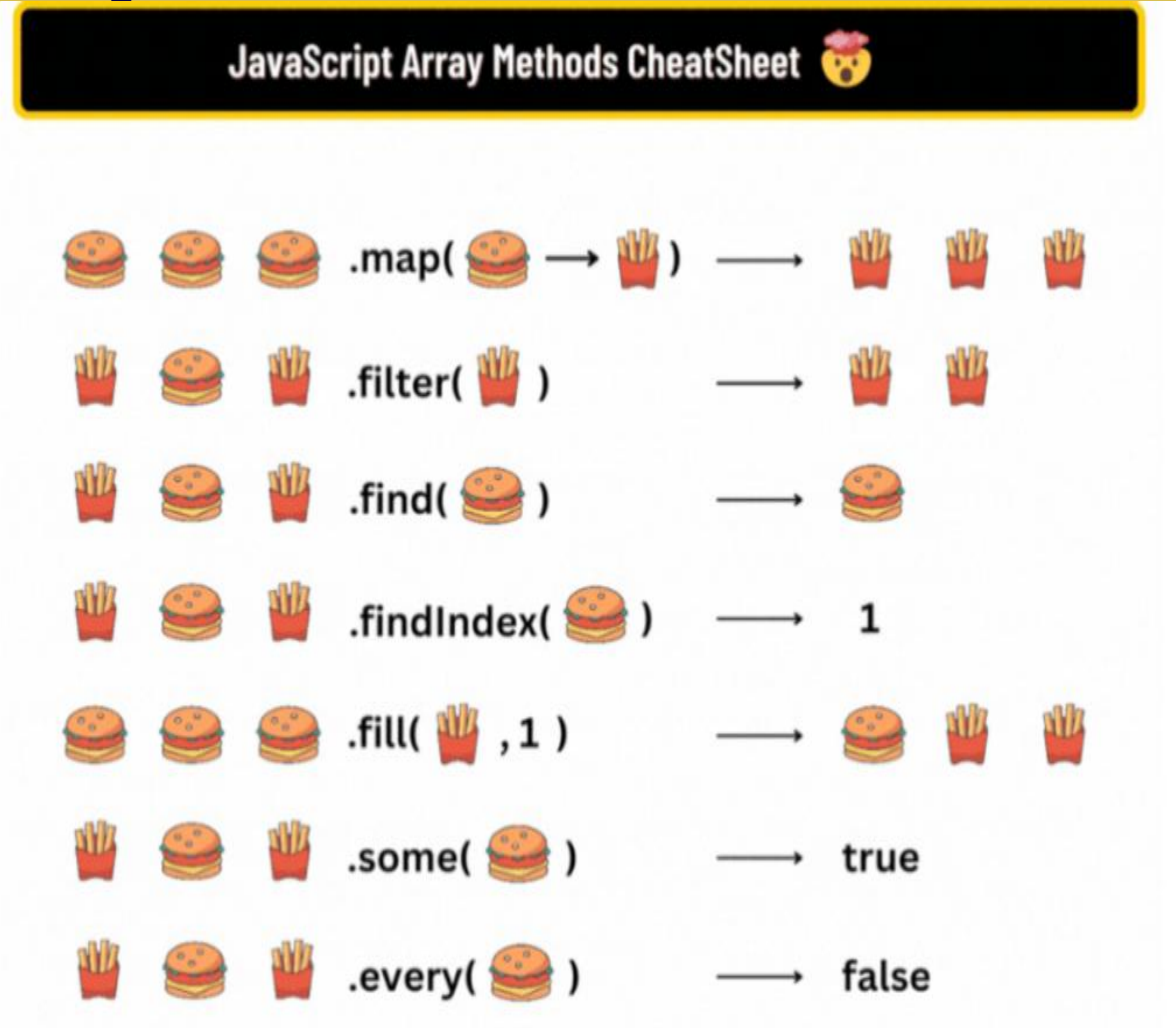
Toward React Large Scale

Multiple file .jsx structure

Let’s code a Capstone Project List

We will definitely need these!

An array in JavaScript is a dynamic list-like structure that can hold multiple values (including objects), and you can use: .map(), .filter(), .forEach(),.reduce(),.find(),. some(), .every(),.sort(),and .flatMap() to iterate over it.


JavaScript Array Methods CheatSheet 🤯
🍔 🍔 🍔 .map(🍔 → 🍟) → 🍟 🍟 🍟
🍟 🍔 🍟 .filter(🍟) → 🍟 🍟
🍟 🍔 🍟 .find(🍔) → 🍔
🍟 🍔 🍟 .findIndex(🍔) → 1
🍔 🍔 🍔 .fill(🍟 , 1) → 🍔 🍟 🍟
🍟 🍔 🍟 .some(🍔) → true
🍟 🍔 🍟 .every(🍔) → false

What does the body element of the *index.html* file generated by Vite look ?

```
<body>




</body>
```

What does a Vite-generated React project have in its *main.jsx* file?

import ______________________________ ;

import ______________________________ ;

import ______________________________ ;

import ______________________________ ;

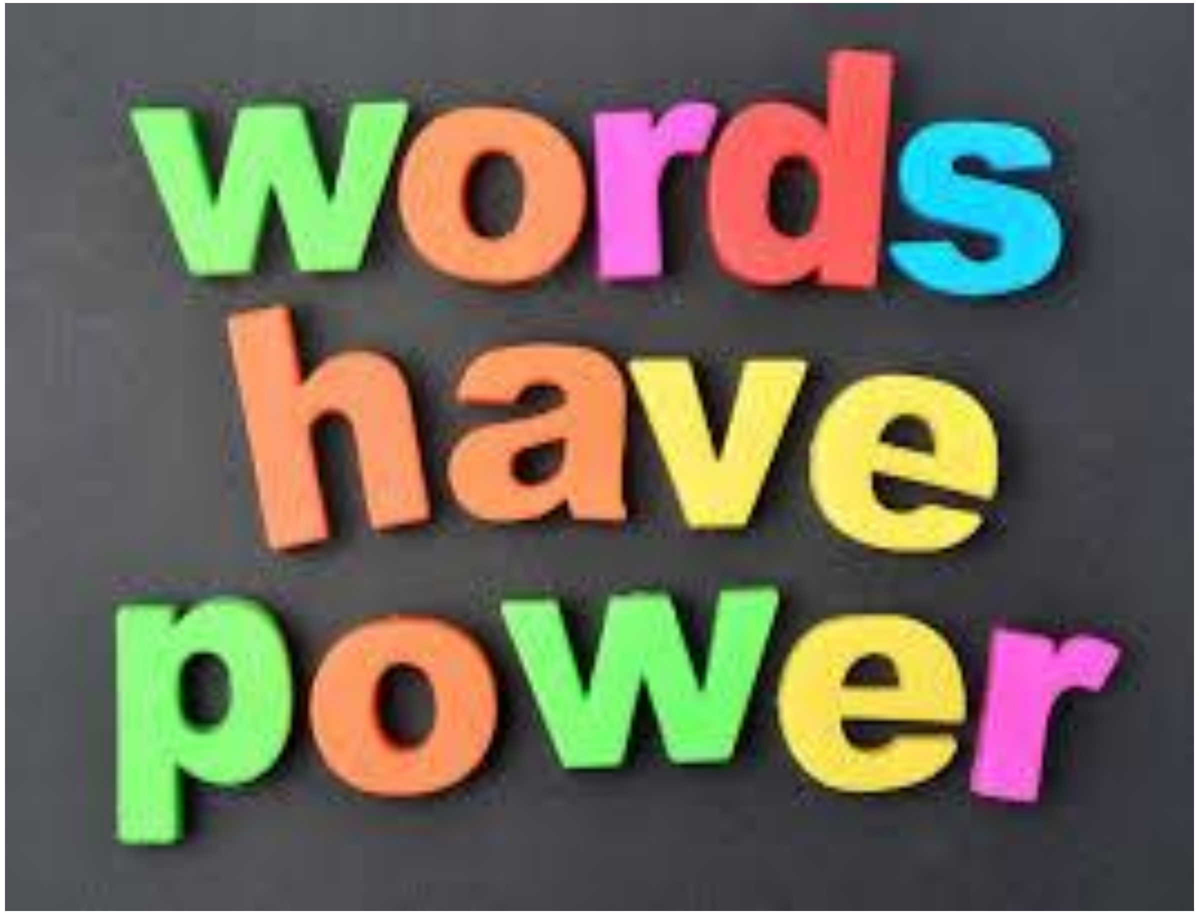
createRoot(document.getElementById("root")).render(

____,);

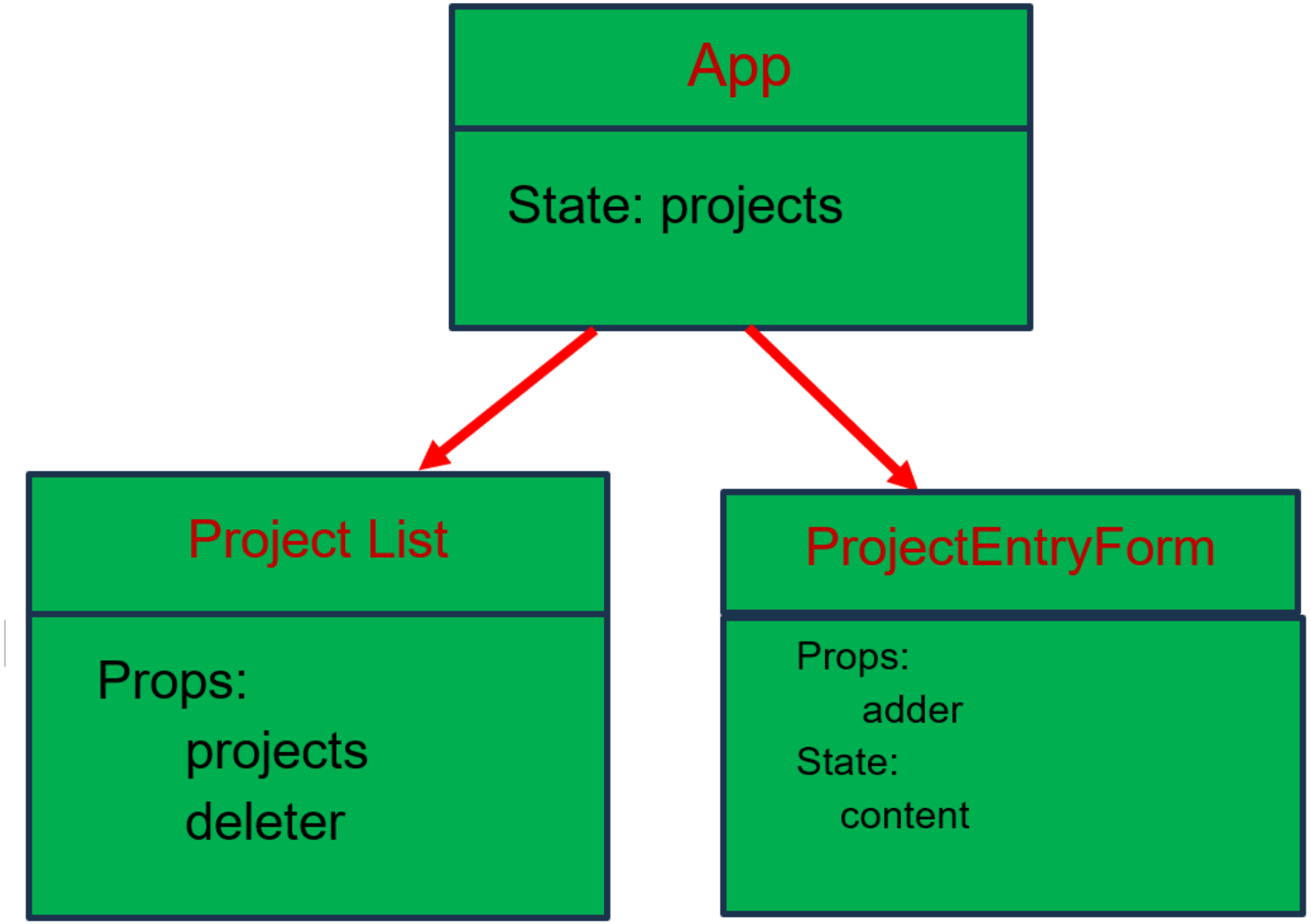
When customizing a Vite-generated React app, what are some things you need to do?

Change the title in ____________________ , remove the React SVG file, clean up the ______________ file, replace the contents of the ______________ file, and update the _________________ file.

Define/ Write 3-4 lines using appropriate vocabulary!



Use given word bank and explain the architectural plan in 3-4 lines!



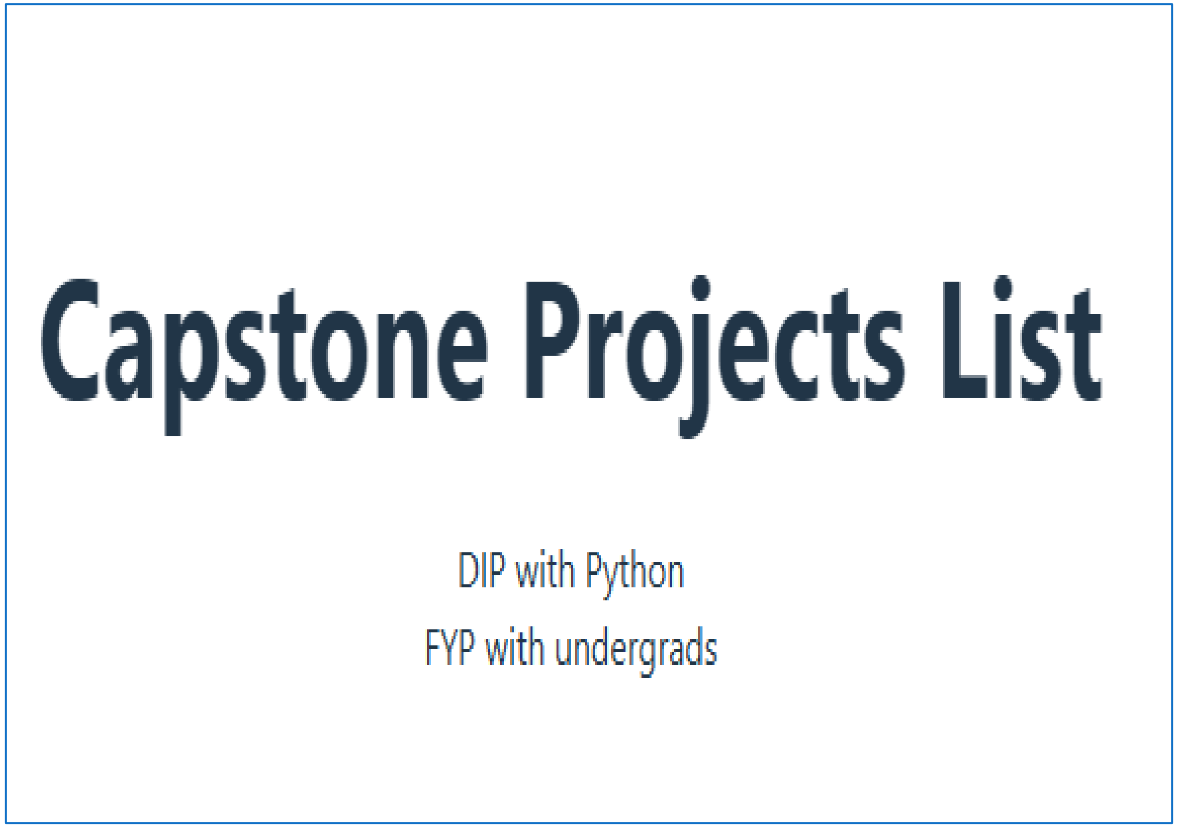
- ✓Components
- ✓.jsx
- ✓App.jsx
- ✓Parent ,child
- ✓List of projects
- ✓State,Prop
- ✓Passing ?
- ✓Render
- ✓Respond to click
- ✓Lift state

App.jsx

ProjectList.jsx

ProjectEntryForm.jsx

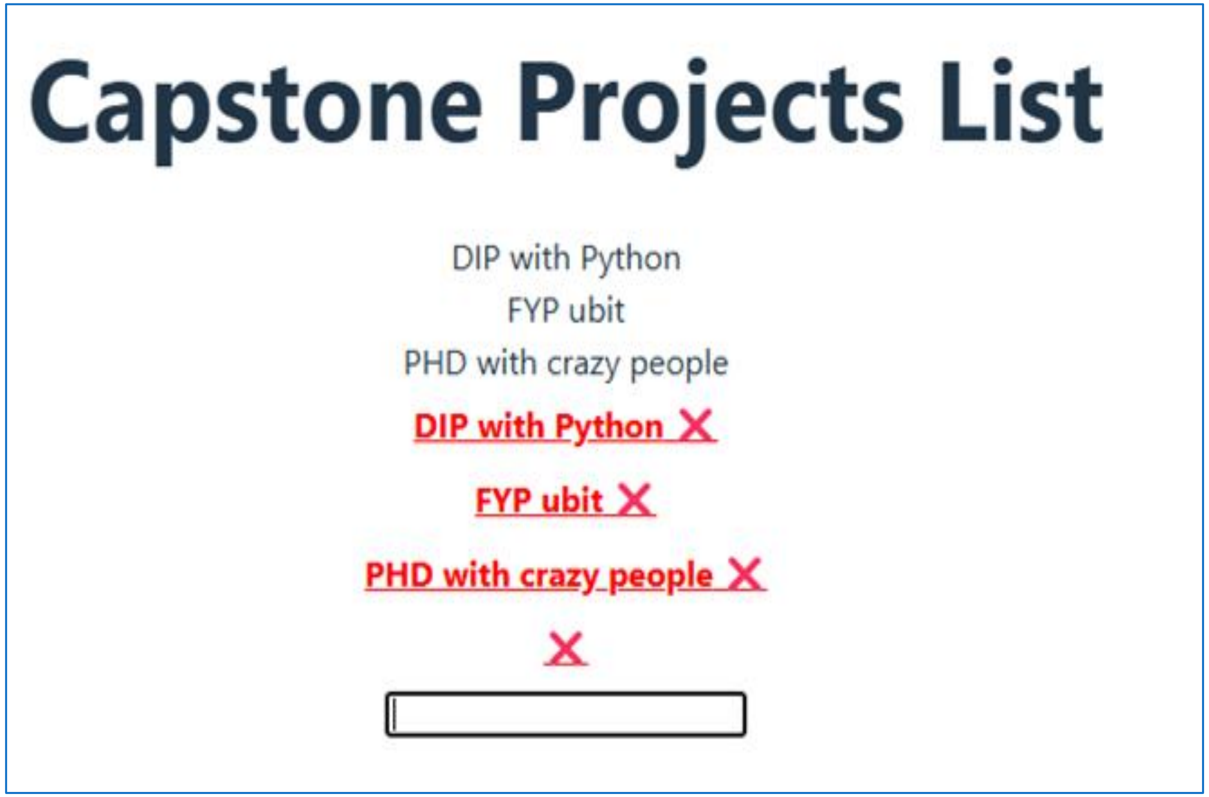
User Interface!



Capstone Projects List

DIP with Python

FYP with undergrads



Capstone Projects List

DIP with Python

FYP ubit

PHD with crazy people

DIP with Python ✗

FYP ubit ✗

PHD with crazy people ✗

✗

Please complete the App() component and answer accordingly

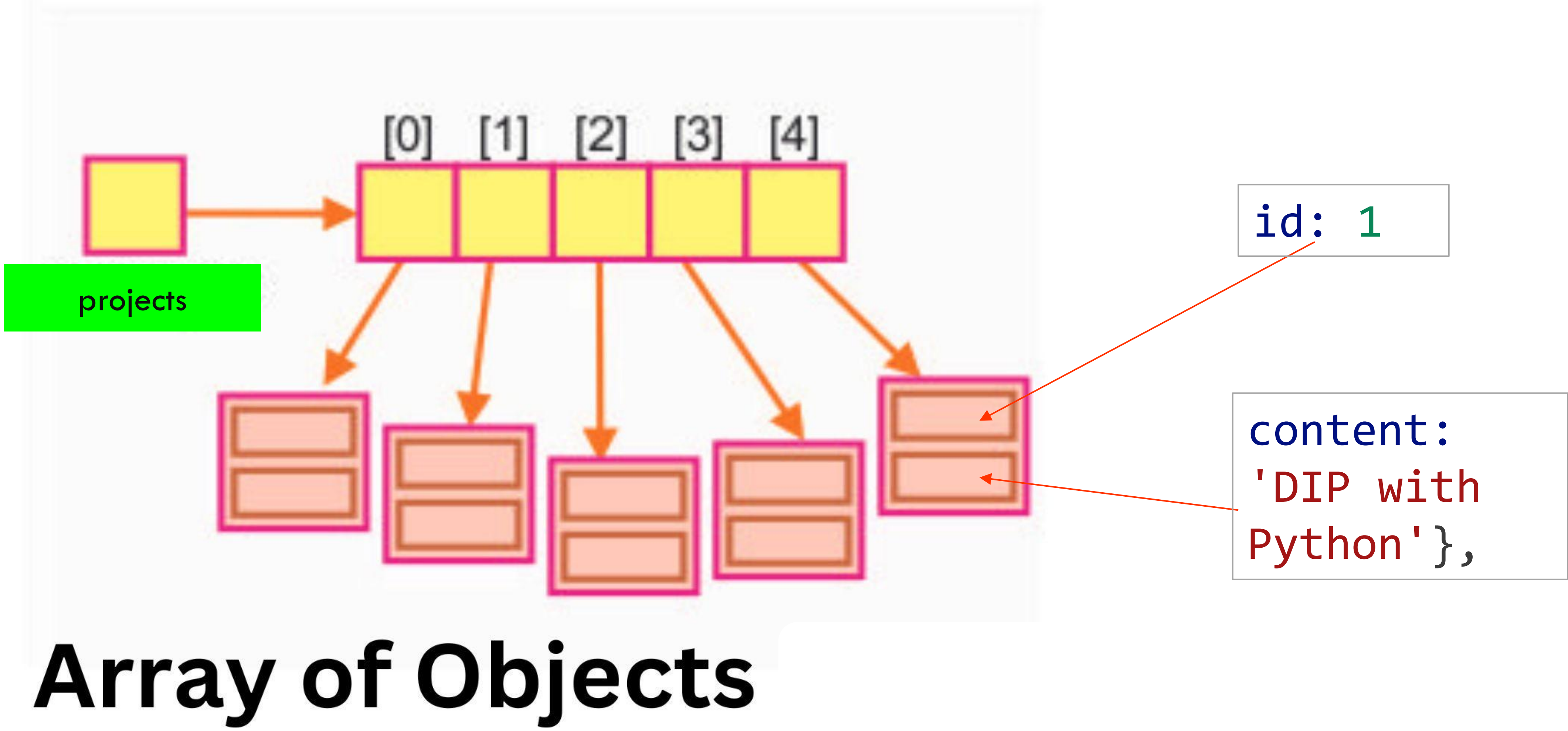
```
import './App.css'
export default function App() {

  const [projects, setProjects] = useState([
    {id: 1, content: 'DIP with Python'},
    {id: 2, content: 'FYP with undergrads'},
    { ____________________________________ },
  ]);

  return (
    <>
      <h1> ______________________________ </h1>
    </>
  );
}
```

This line initializes a **state variable** in a React functional component using the useState hook.

```
const [projects, setProjects] = useState([
    {id: 1, content: 'DIP with Python'},
    {id: 2, content: 'FYP with undergrads'},
    { ___________________________________  },
  ]);
```

Here’s the shortest yet clear interpretation:

projects holds an array of project objects, and setProjects updates it.



```
id: 1
```

```
content:
'DIP with
Python'},
```

- ✓ The React way is to keep the state high up in the tree and pass it down to ________________ that will do the rendering.

- ✓ The useState line will create a ____________variable called "projects" and a function called "setProjects"

- ✓ When setProjects updates this array, React will __________ the component to reflect the changes.
- ✓ The stateful_____?____Hint (Array/variable) of objects is stored in state.
- ✓ The left-hand side (projects) is just a ____________ to that stateful data.

Create the file for first child component ***src/ProjectList.jsx***

```
@@@@@@@@@@@@@@@@@@@ Version 1: Using if-else @@@@@@@@@@@@@@@@@@@@
```

```
export default function _______________ ({ projects }) {

let pList;

  if ( _____________________ ) {


    pList = projects.map( ______________=> <div __________ </div>);
  } else {

   // If no projects exist, show a message
    pList = <p> _____________________ </p>;

}
  // Return a <div> containing either the project list or the message
  return <div>  ______________ </div>;
}
```

Why do we need a new array each time?

What happens if we don’t create a new array?

Write equivalent (forEach, push) version of ***ProjectList.jsx.*** ***will it work?***

Why Modifying the original array (forEach, push) is not recommended in React.

✓ **.map()** **is used to transform data** (convert an array of _____ to array of ________).

✓ React’s rendering **requires a new** _____________________ **on** every update.

✓ In **functional programming**, we **avoid** ___________________ **(mutating)** and instead **create** __________________ with updates. This is crucial in React because React's **state management relies on** ____________________________________ to detect changes and trigger re-renders efficiently.

@@@@@@@@@@@@@@@@@@@ Version 2: src/ProjectList.jsx @@@@@@@@@@@@@@@@@@@@

Return Early (Slightly Shorter), Avoids storing pList in a variable,
//More readable than the previous version, but still uses if-else.

```
export default function ________________ ({ projects }) {

  if ( _________ === ____________) {

     return <p> _____________________ </p>;
    }


   return (

       ____________________________________________________

     );
}
```

@@@@@@@@@@@@@@ Version 3: src/ProjectList.jsx @@@@@@@@@@@@@@@@@@

@@@@@@@@@ Inline JSX with Ternay Operator @@@@@@@@

Uses **inline JSX , no pList variable**

□ projects.length ? … : … → This is the **ternary operator**.

□ **Condition:** projects.length (If there are projects).

□ **If true:** map() over the projects and display them.

□ **If false:** Show a <p>No Projects left</p>.

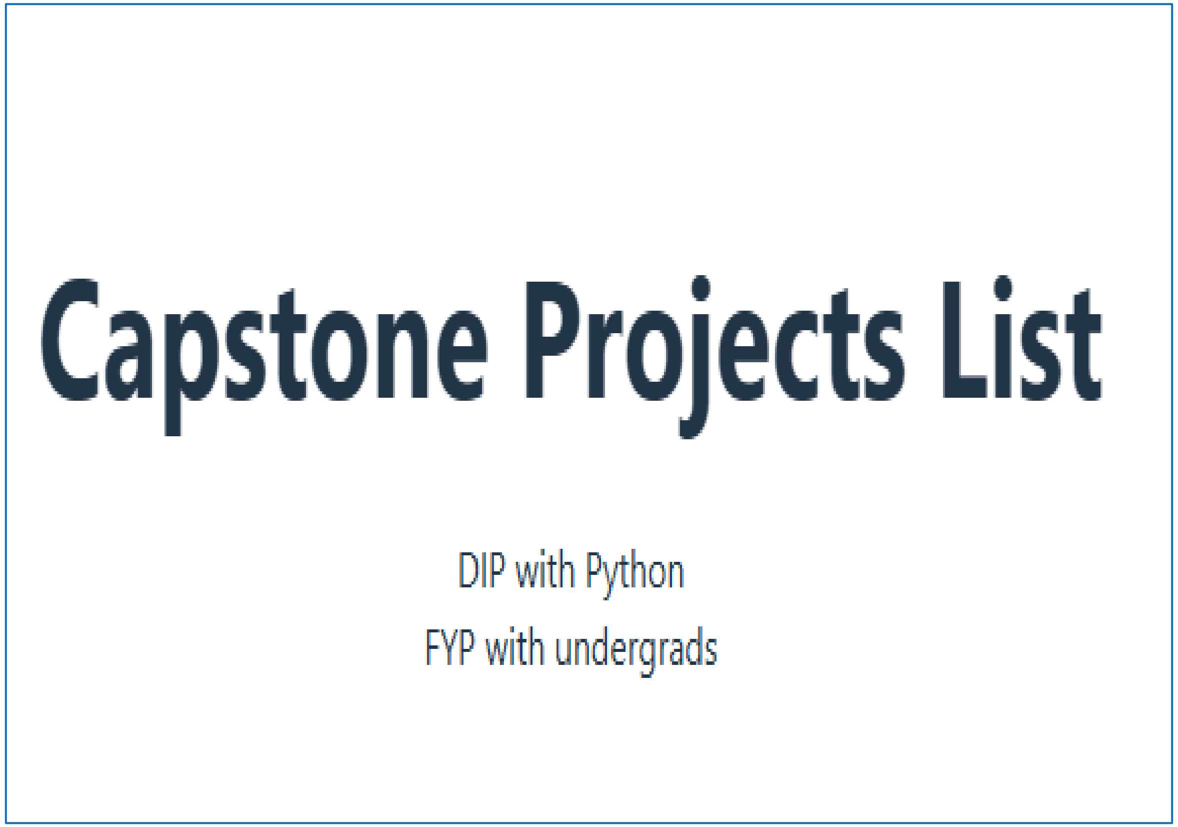
```
export default function  ____________________ ({ projects }) {
    const pList =  _______________________   ? (



   ) : (
      <p>No Projects left</p>
    );
    return <div>{pList}</div>;   }
```

Let’s add **delete** functionality. We want that when we click on a project, it is deleted.

One way to do this deletion is for App to pass ____________________ as a prop, but this gives the ______________ component too much power.

Instead, we create a ______________________ in App called ___________ and pass just *that* down as a prop.

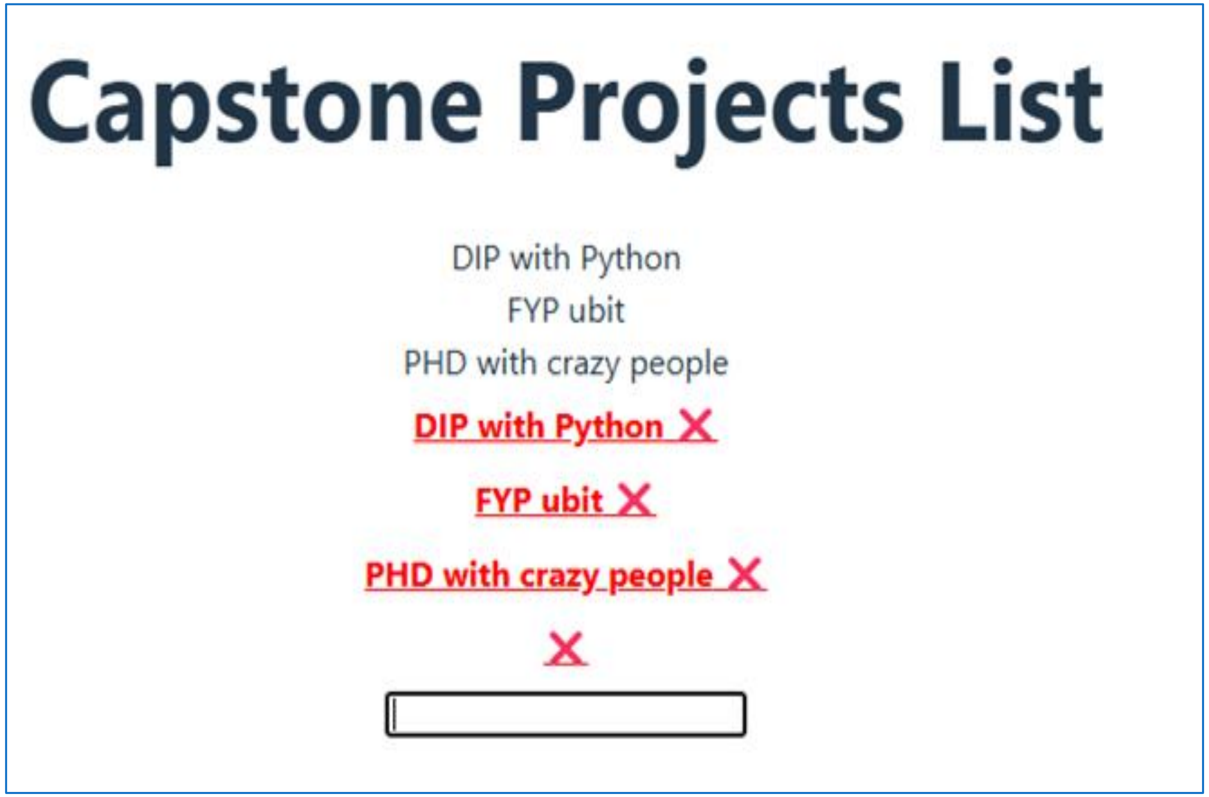
Recall DesiredUser Interface!



Capstone Projects List

DIP with Python

FYP with undergrads



Capstone Projects List

DIP with Python

FYP ubit

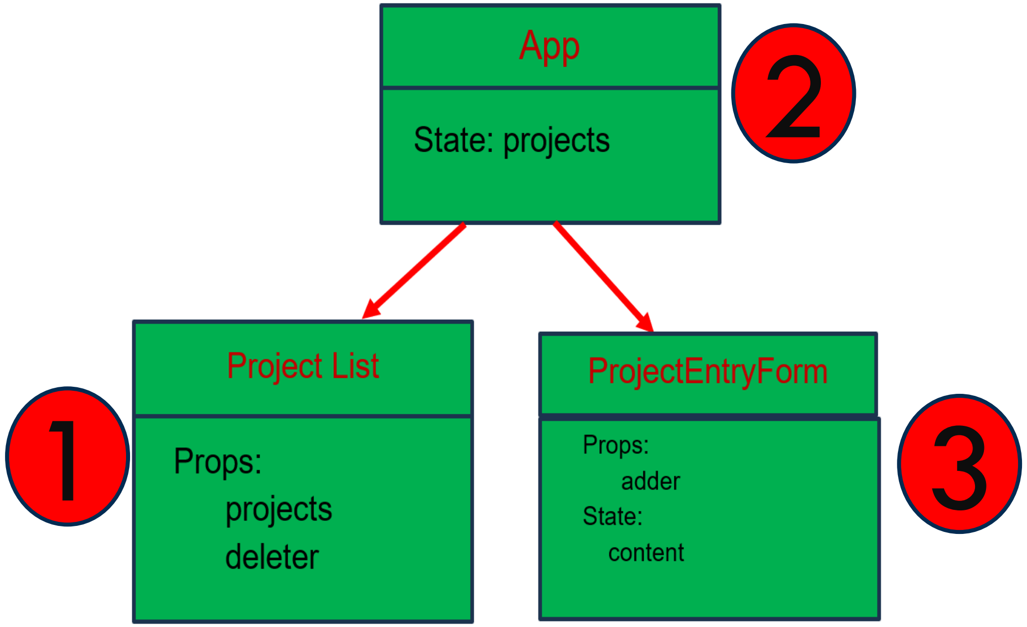
PHD with crazy people

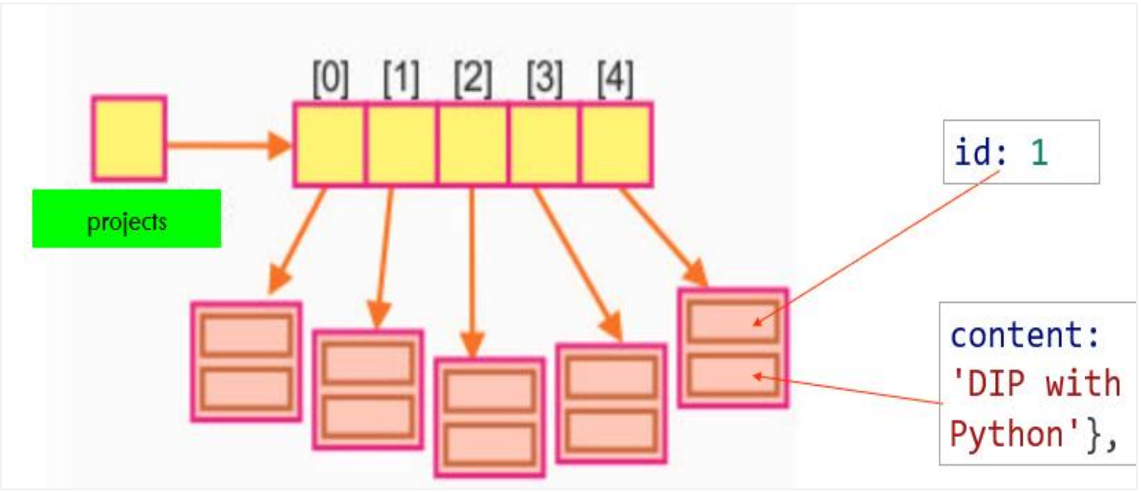
DIP with Python ✗

FYP ubit ✗

PHD with crazy people ✗

✗





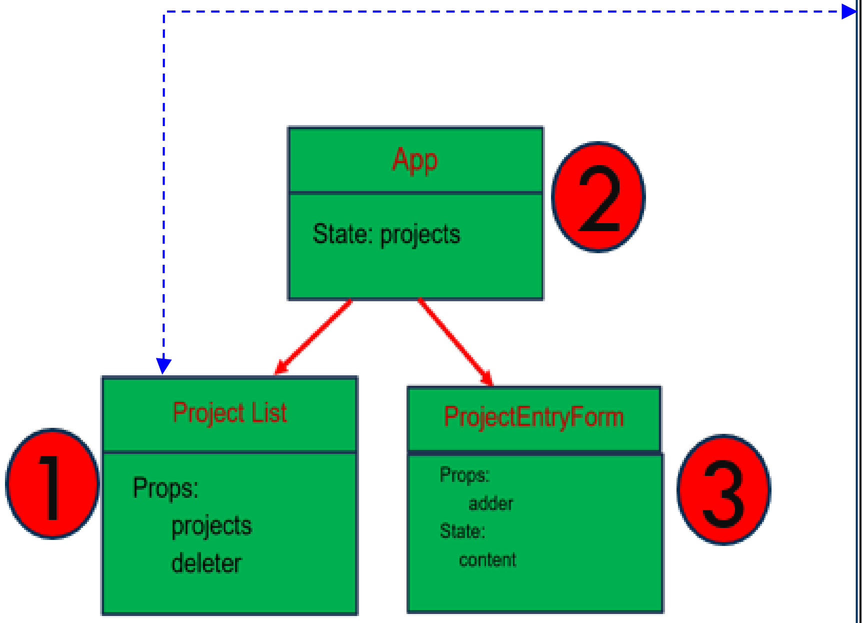
Array of Project

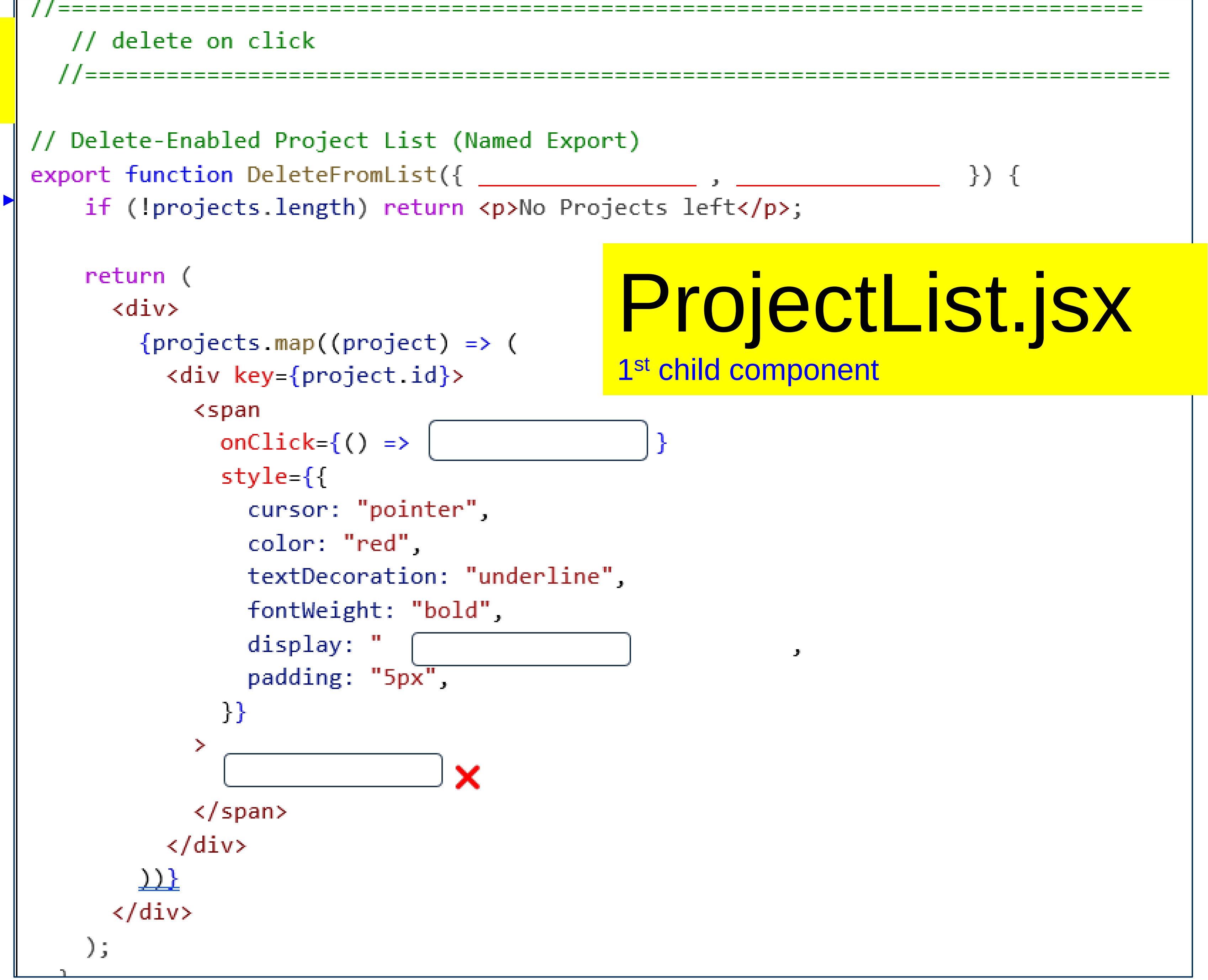
ProjectList.jsx

1st child component

```
// File ProjectList.jsx
export default function ProjectList({ projects }) {
    const pList = _________________   ? (

      projects.map( _________________________________________________ )
    ) : (
      <p>No Projects left</p>
    );
    return <div> __________________  </div>;
  }
```

DeleteFromList ()





ProjectList.jsx

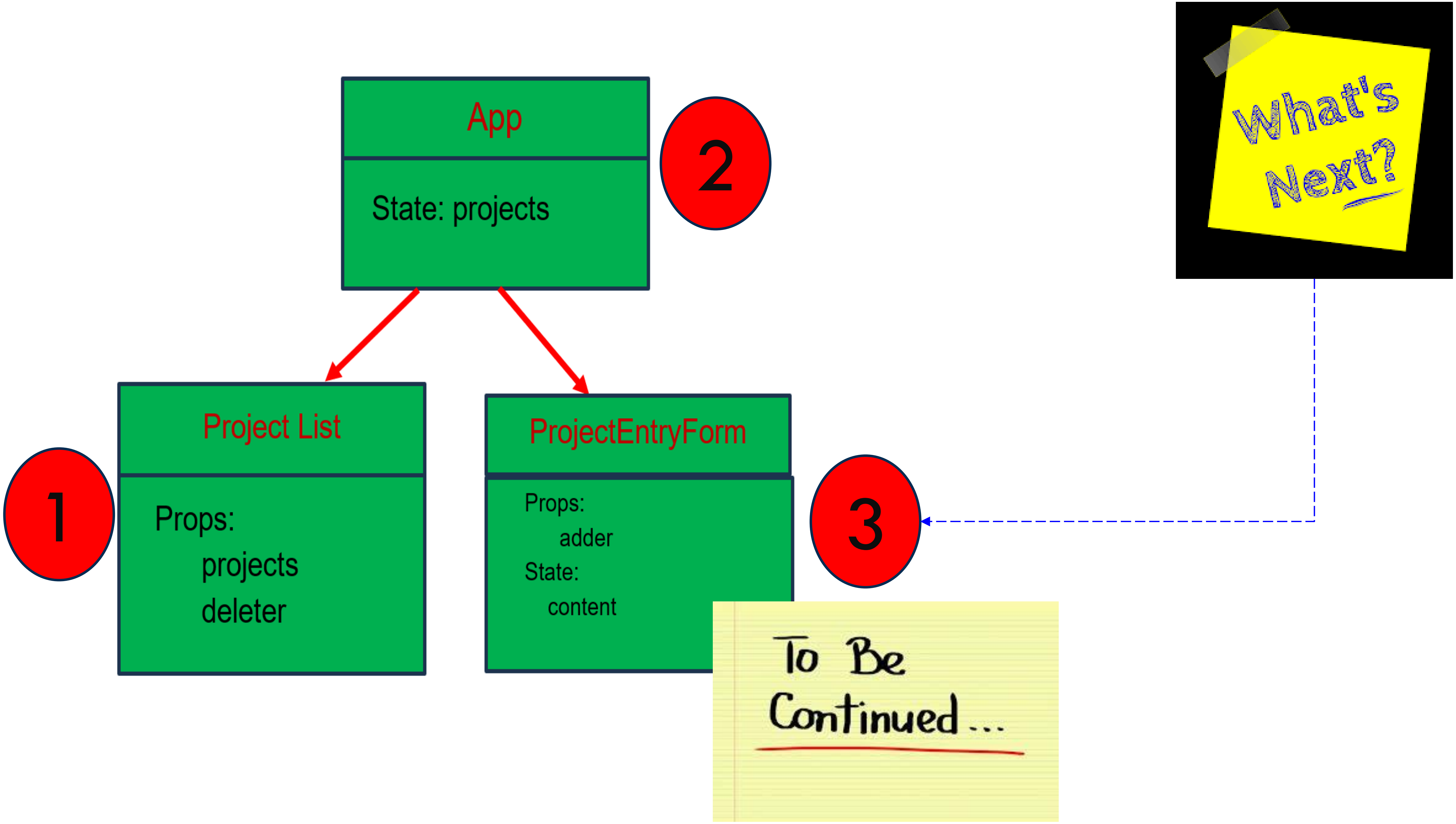
1st child component

```
//========================================================================
   // delete on click
  //========================================================================

// Delete-Enabled Project List (Named Export)
export function DeleteFromList({ ________________ , _______________  }) {
    if (!projects.length) return <p>No Projects left</p>;

    return (
      <div>
        {projects.map((project) => (
          <div key={project.id}>
            <span
              onClick={() =>                  }
              style={{
                cursor: "pointer",
                color: "red",
                textDecoration: "underline",
                fontWeight: "bold",
                display: "                              ,
                padding: "5px",
              }}
            >
                                ✗
            </span>
          </div>
        ))}
      </div>
    );
  }
```

App.jsx

```
// File App.jsx
import './App.css'
import { useState } from 'react'; // Import useState
import ProjectList from './ProjectList';
import { DeleteFromList } from "./ProjectList"; // Named export (Correct!)

export default function App() {  // parent component
   {/*Create a state variable called "projects" and a setter function called
   "setProjects". This stores array of objects, each with an id and content*/}

// Delete function: Removes a project by filtering out the matching ID
  function deleteProject(_________ ) {

      setProjects(  ___________ => projects.filter( _______________________ );
  }
  return (
    <>
      <h1>Capstone Projects List</h1>
      {/* / Only Display Projects → Pass only projects.
       Display + Delete Projects → Pass both projects and deleter */}
      <ProjectList projects={projects} />
      <DeleteFromList projects={projects} deleter={ _______________ } />
    </>
  );
}
```

Passing projects and deleteProject() as props to child component DeleteFromList



App

State: projects

2

Project List

Props:
projects
deleter

1

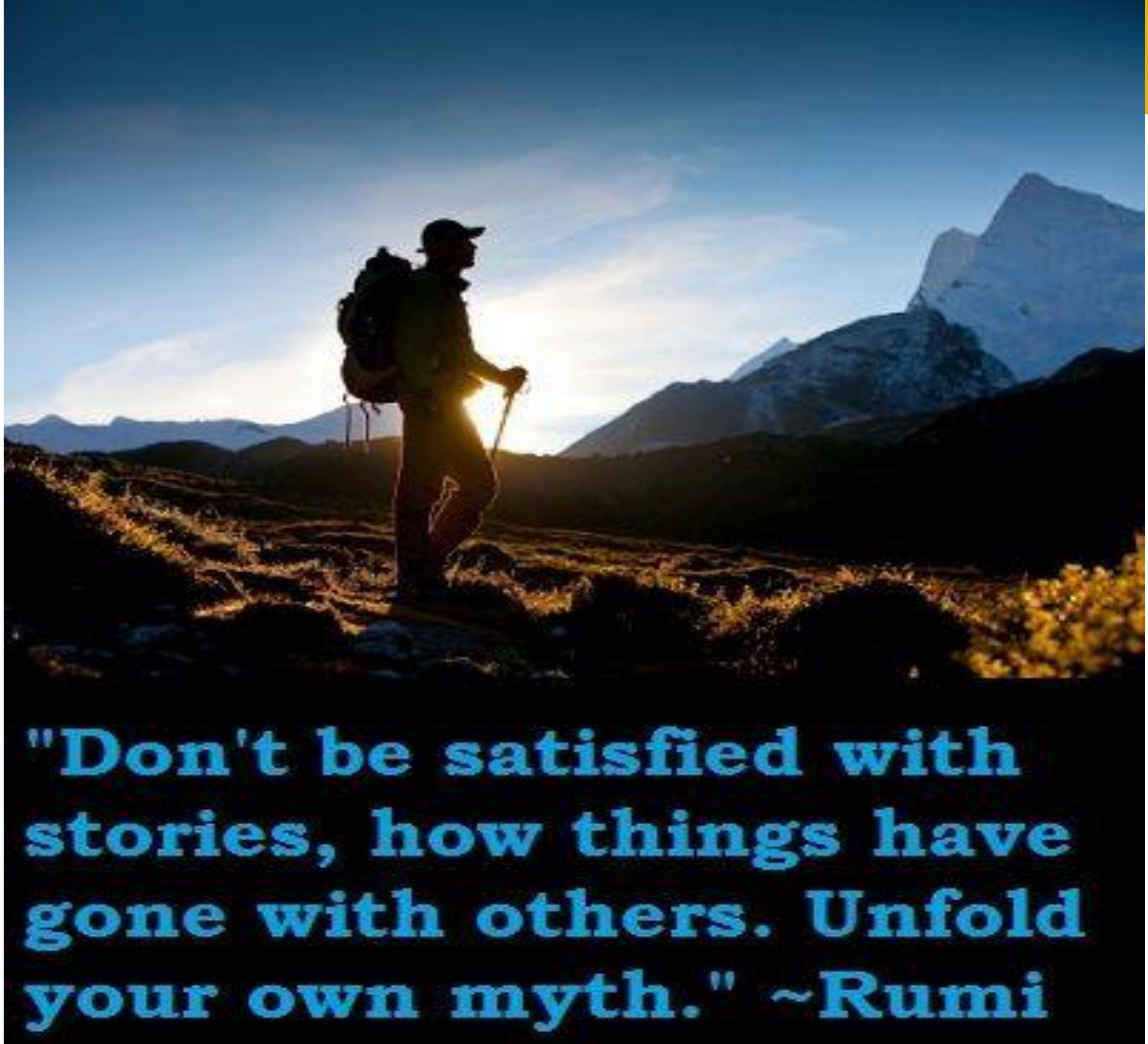
ProjectEntryForm

Props:
adder
State:
content

3

THANKS
for your
ATTENTION

Any questions
?



"Don't be satisfied with stories, how things have gone with others. Unfold your own myth." ~Rumi

Department of Compute Science (UBIT Building), Karachi, Pakistan.

1200 Acres (5.2 Km sq.)
53 Departments
19 Institutes
25000 Students

UNIVERSITY OF KARACHI

My Homeland Pakistan