Roll No :- 48
Name:- Manav jain
Date:-25/08/2023

ASSIGNMENT – 6(a)

AIM:- WAP to implement concept of props and state . Create functional component and pass current date as props and with class component , on button display both date and time

LO MAPPED:- LO5

THEORY:-

In React, props and state are two fundamental concepts that are used to manage and pass data around in components. They play a crucial role in building dynamic and interactive user interfaces. Let's take a look at each of them with examples:

Props:

Props are read-only properties that are passed from a parent component to a child component. They allow you to pass data from one component to another and are used to make components reusable and modular. Props cannot be modified by the child component.

E.g. :-

```
import React from 'react';
import ChildComponent from './ChildComponent';

function ParentComponent() {
  const greeting = 'Hello,';

  return (
    <div>
      <ChildComponent message={greeting} />
```

```
    </div>
  );
}

export default ParentComponent;

// ChildComponent.js
import React from 'react';

function ChildComponent(props) {
  return <p>{props.message} World!</p>;
}

export default ChildComponent;
```

In this example, the ParentComponent passes the message prop to the ChildComponent, which then displays "Hello, World!" using the passed prop.

State:

State is used to manage the internal state of a component. It is mutable and can be modified by the component itself using the setState function. When state changes, React re-renders the component to reflect the updated state.

E.g. :-

```
import React, { Component } from 'react';
class CounterComponent extends Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0,
    };
  }

  incrementCount = () => {
```

```
    this.setState(prevState => ({
      count: prevState.count + 1,
    }));
  };

  render() {
    return (
      <div>
        <p>Count: {this.state.count}</p>
        <button onClick={this.incrementCount}>Increment</button>
      </div>
    );
  }
}

export default CounterComponent;
```

In this example,CounterComponent class component is created. The component's state includes a property called count which starts at 0. The incrementCount method uses setState to update the count when the "Increment" button is clicked. The component's render method displays the current count and the button.

When the button is clicked, React will automatically update the component's state, triggering a re-render of the component with the updated count value.

OUTPUT WITH CODE:-

1)classcomp.jsx :-

```
import React, { Component } from 'react';

class ClassComponent extends Component {
  constructor() {
```

```
    super();
    this.state = {
      color: 'black',
      datetime: new Date().toLocaleString()
    };
  }
  changeColor = () => {
    const colors = ['orange', 'cyan', 'purple'];
    const randomColor = colors[Math.floor(Math.random() * colors.length)];
    this.setState({ color: randomColor });
  }
  render(){
    const { color, datetime } = this.state;
    return(
        <div>
        <h1>Class Component</h1>
        <p> Current Date and Time:</p><p style={{color}}>{datetime}</p>
        <button onClick={this.changeColor}>Change Color</button>
        </div>
    );
  }
}
export default ClassComponent;
```

2)functional.jsx :-

```
import React from 'react';
function FunctionalComponent (props){
    const {datetime} = props;
    return(
        <div>
            <h1>Functional Component</h1>
            <p>Date and time = {datetime}</p>
        </div>
    );
}
```

```
export default FunctionalComponent;
```

3)App.js :-

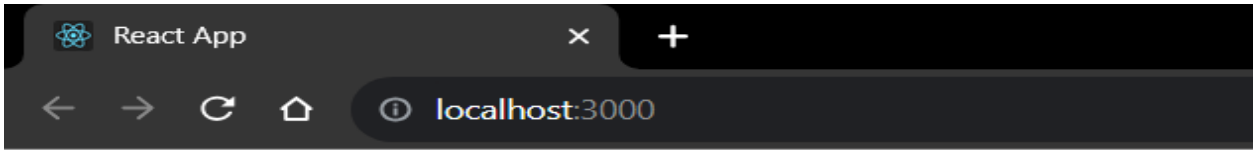
```
import React from 'react';
import FunctionalComponent from './FunctionalComp';
import ClassComponent from './ClassComp';

function App() {
  const currentDatetime = new Date().toLocaleString();

  return (
    <div>
      < FunctionalComponent datetime={currentDatetime} />
      < ClassComponent datetime={currentDatetime} />
    </div>
  );
}

export default App;
```

4)Output :-



Functional Component

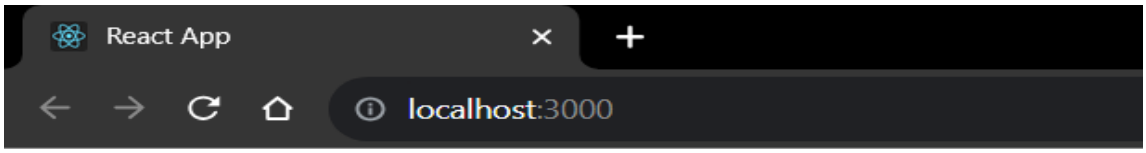
Date and time = 9/7/2023, 8:53:03 PM

Class Component

Current Date and Time:

9/7/2023, 8:53:03 PM

Change Color



Functional Component

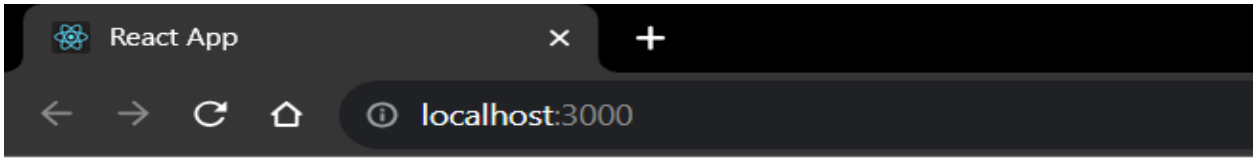
Date and time = 9/7/2023, 8:22:20 PM

Class Component

Current Date and Time:

9/7/2023, 8:22:20 PM

Change Color



Functional Component

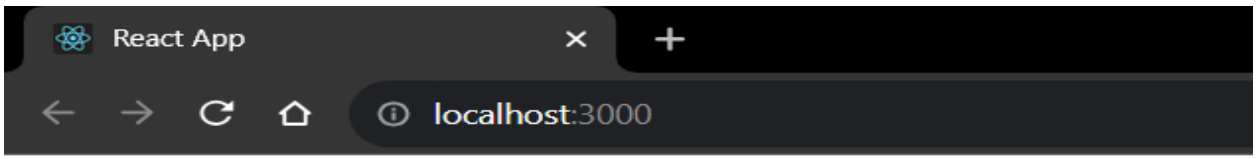
Date and time = 9/7/2023, 8:22:20 PM

Class Component

Current Date and Time:

9/7/2023, 8:22:20 PM

Change Color



Functional Component

Date and time = 9/7/2023, 8:22:20 PM

Class Component

Current Date and Time:

9/7/2023, 8:22:20 PM

Change Color

REFERENCES :-

1)https://www.geeksforgeeks.org/reactjs-state-vs-props/#:~:text=Props%20are%20read%2Donly.,data%20inside%20a%20component%20itself.
2)https://www.javatpoint.com/react-state-vs-props

CONCLUSION :- Learnt about usage of state and props in react , learnt about basic syntax of these and also discovered about use-case of these and implemented a program to demonstrate function of these two .