THEORY Assignment

- ✓ **Props/state**

• Question 1: What are props in React.js? How are props different from state?

Props (short for "properties") are used to **send data from one component to another**, usually from a **parent component to a child**. They help in making components reusable. Props are **immutable**, meaning the child component **cannot change** the data it receives.

State, on the other hand, is used to **manage data inside a component**. It can **change over time** due to user actions, API calls, etc., and when it changes, the component **automatically re-renders**.

- ✓ **Props = Passed from parent, Read-only**
  State = Managed within component, Can change

Example:

```
function Welcome(props) {
  return <h1>Hello, {props.name}!</h1>;
}
```

• Question 2: Explain the concept of state in React and how it is used to manage component data.

State is a special object in React that holds information about a component's current situation — like a user's input, a counter value, or data fetched from an API.

When you update the state, React **re-renders the component automatically**, showing the updated data on the screen.

In functional components, you use the useState() hook:

```
const [count, setCount] = useState(0);
function increase() {
  setCount(count + 1);
}
```

Use state when you want the UI to **change based on user actions or other events.**

• **Question 3: Why is this.setState() used in class components, and how does it work?**

In **class components**, you can't change state directly. Instead, you use this.setState() to **request a state update**.

React then **merges the new state** with the old one and **re-renders the component** to reflect the change.

Example:

```
class Counter extends React.Component {
  constructor() {
    super();
    this.state = { count: 0 };
  }

  increaseCount = () => {
    this.setState({ count: this.state.count + 1 });
  };

  render() {
    return <button onClick={this.increaseCount}>Count: {this.state.count}</button>;
  }
}
```

- this.setState() helps React keep track of changes and update the UI properly.

Let me know if you want a visual chart comparing props vs state, or examples using real-world UI elements like buttons, forms, etc.