CET-223

Web Technologies

Experiment # 18

Experiment Title

React JS - II

Assessment of CLO(s): 03

Performed on ____________________

Student Name:			
Roll No.		**Group**	
Semester		**Session**	

Total (Max)	**Performance (03)**	**Viva (03)**	**File (04)**	**Total (10)**
Marks Obtained				
Remarks (if any)				

Experiment evaluated by

Instructor's Name	**Engr. Bilal Iqbal**		
Date		**Signature**	

Objective:

This lab builds on the basics of React by introducing advanced concepts, including state management and to understand and practice the usage of two important React hooks: useState and useEffect.

Understanding React State and useState Hook:

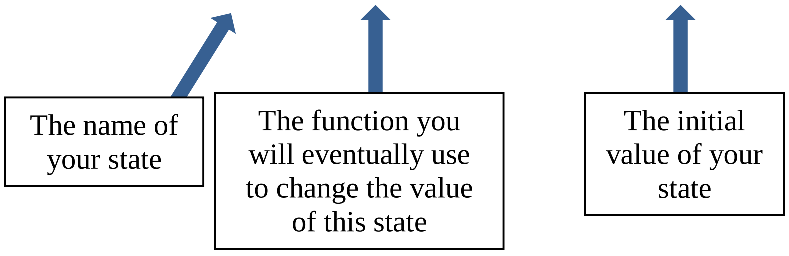
What is State?

- State is a built-in object that stores data that changes over time in a component.
- Changes to state trigger re-renders, updating the UI with the new data.

Using `useState` Hook:

- The `useState` hook is used to create state in functional components.
- `useState` is a React hook used to declare state variables in functional components. It allows us to store and update values that are used in the component.

Const [state, setstate] = useState (initial State)


The name of your state
The function you will eventually use to change the value of this state
The initial value of your state

Example: Creating a Counter with useState

```
// src/Counter.js
import React, { useState } from 'react';
function Counter() {
  const [count, setCount] = useState(0);
  const handleClick = () => setCount(count + 1);
  return (
    <div>
      <h2>Counter: {count}</h2>
      <button className='eg-button-inc' onClick={handleClick}>Increase</button>
     </div>
  );
}
export default Counter;
```

```
// src/App.js
import Counter from './Counter';
function App() {
  return (
    <>
    <Counter/>
    </>
  );
}
export default App;
```

- The `useState` hook initializes `count` to `0`.
- `setCount` updates `count`, triggering a re-render.
- In this example, the component maintains a count state and provides buttons to increment or decrease the count.

Example: Input Field State

```
// src/TextInput.js
import React, { useState } from 'react';
function TextInput() {
    const [text, setText] = useState("");
    const handleInput = (event) =>  setText(event.target.value);
    return (
        <div>
            <input type="text" placeholder="Type
something..."  value={text}  onChange={handleInput}/>
            <p>You typed: {text}</p>
        </div>
    );
}

export default TextInput;
```

```
import './App.css';
import TextInput from './TextInput';
function App() {
  return (
    <>
    <TextInput/>
    </>
  );
}
export default App;
```

- **`handleInput`**: A function that gets triggered whenever the user types in the input box (via the `onChange` event).

- **`event.target.value`**: Captures the current value of the input field and updates the `text` state using `setText`.

Lab Tasks:

1. Create a counter application that allows users to increase or decrease the count using buttons.
2. Create a button that changes the background color of the page when clicked.