React-Outline

Raja Ata Ul Karim

Module 1:

> Understood how to initiate a React application (via Vite & CRA).

> Environment setup (VSCode + NodeJS)

> To create react app via CRA: *npm create-react-app app-name*

> To create react app via Vite: *npm create vite@latest*

> Components & Props:

```
import '../styles/basicComponent.css'

function list(props)
{
    return <li>Hello {props.listItems} </li>;
}

export default function Filler()
{
    const listItems = [
        'Pizza', 'Burger', 'Fries'
    ];

    const listedListItems = listItems.map((listItems)=>{
        return <li>{listItems}</li>
    });

    return (
        <>
            <div className="container">
                <h1 className='header'>Header Text</h1>
                <p>
                Lorem ipsum dolor sit amet, consectetur adipiscing elit.
                Nulla vulputate mattis elementum. Suspendisse sodales quam a enim
vulputate laoreet.
```

```
                Mauris porta vestibulum nisi, vitae consectetur massa volutpat
quis. Donec a dignissim leo.
                </p>
                <ul>
                    {listedListItems}
                </ul>

            </div>
        </>
    )
}
```

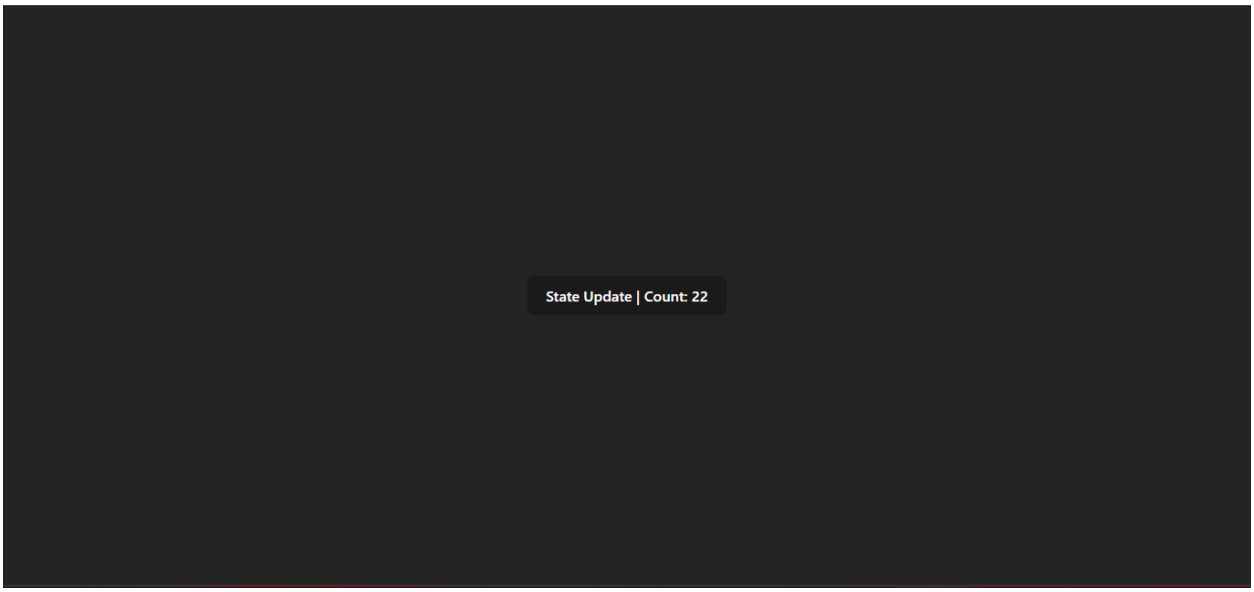
A stand-alone component with listItems being passed as a prop.

> State and Lifecycle Methods

State:

```
import { useState } from 'react'
import './App.css'

function App() {
  const [count, setCount] = useState(0)

  return (
    <>
        <button onClick={() => setCount((count) => count + 1)}>
          State Update Count: {count}
        </button>
    </>
  )
}

export default App
```

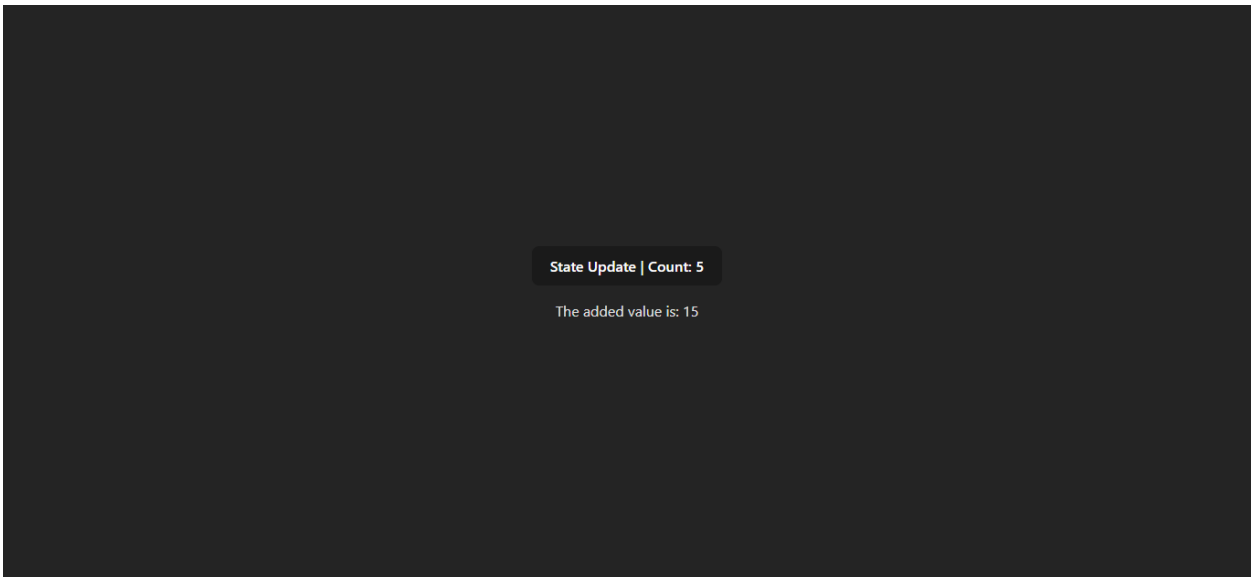
Implemented Output:



Lifecycle:

```
import { useEffect, useState } from 'react'
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'

function App() {
  const [count, setCount] = useState(0)
  const [addCount, setAddCount] = useState(0)

  useEffect(() => {
    setAddCount(() => addCount + count);
  }, [count]);

  return (
    <>
        <button onClick={() => setCount((count) => count + 1)}>
          State Update | Count: {count}
        </button>
        <p>The added value is: {addCount} </p>
    </>
  )
}

export default App
```

Implemented Output:



Module 2:

Class vs Functional Components:

> A functional component is a simple Javascript function as shown below:

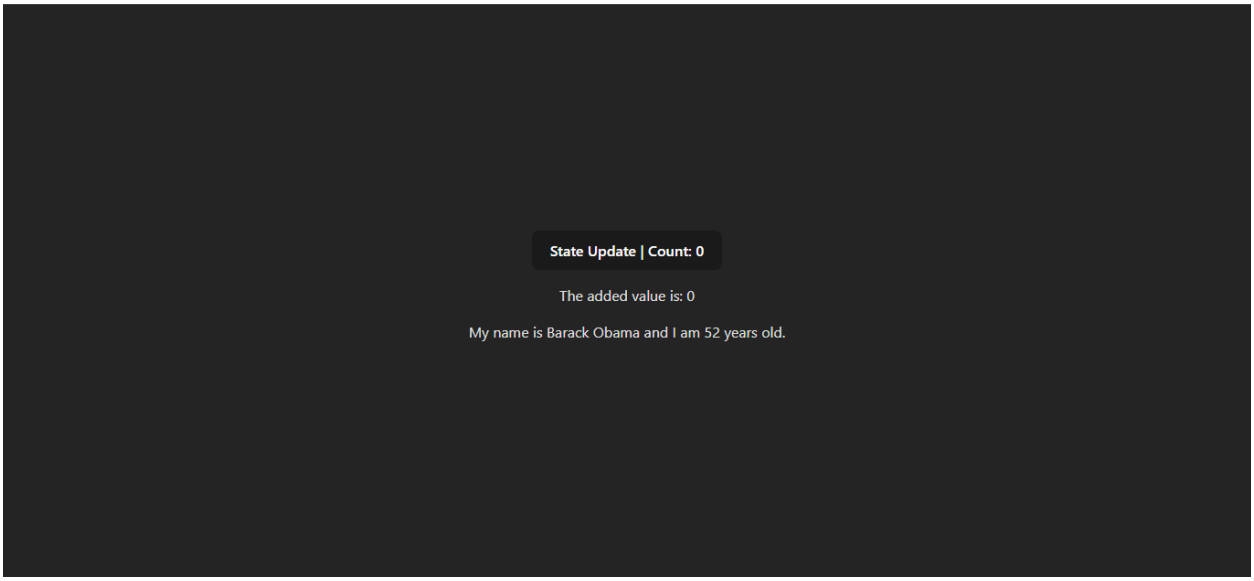
```
function list(props)
{
    return <li>Hello {props.listItems} </li>;
}
```

> A class component creates classes in React which can be utilized just like a normal component.

```
import React from "react";

class Object_1 extends React.Component {

    constructor(props) {
        super(props);
        this.state = {
```

```
            name: "Barack Obama",
            age: 52
        };
    }
    render() {
        return (
            <div>
                <p>My name is {this.state.name} and I am {this.state.age} years
old.</p>
            </div>
        );
    }
}

export default Object_1;
```

Implemented Output:



State Management in Functional Components:

Note: State Management has been utilized in the code for Lifecycle above ^

Handling Events:

Note: Event Handling has been performed in the code for Lifecycle above

Conditional Rendering:

```jsx
import { useEffect, useState } from 'react'
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'
import Object_1 from './Components/test'

function App() {
  const [count, setCount] = useState(0)
  const [addCount, setAddCount] = useState(0)
  const [clicked, setClicked] = useState(false)

  useEffect(() => {
    setAddCount(() => addCount + count);
  }, [count]);

  const handleClick = () =>
  {
    console.log(clicked)
    setClicked(!clicked);
    console.log(clicked)
  }

  function conditionalCheck()
  {
    if (handleClick == true)
    {
      return <p>The Earth is NOT flat.</p>

    }
    else
    {
      return <p>The Earth is flat.</p>
    }
  }

  return (
    <>
        <button onClick={() => setCount((count) => count + 1)}>
          State Update | Count: {count}
        </button>
        <p>The added value is: {addCount} </p>

        <button onClick={handleClick}>
```

```
          Click this for a fact.
        </button>

      { conditionalCheck()}



        {/* <Object_1/> */}
    </>
  )

}

export default App
```

Implemented Output: