Experiment 6:
Create a component Counter with A state variable count initialized to 0. Create Buttons to increment and decrement the count. Simulate fetching initial data for the Counter component using useEffect (functional component) or componentid count (class component). Extend the Counter component to Double the count value when a button is clicked. Reset the count to 0 using another button.
Steps for reactjs:

Old approach

================
First install GIT

Ensure you are installing a compatible version of React, such as react@18. Run the following command to create the React app with a specific version:

npx create-react-app counter-app

.Or using Vite
npm create vite@latest counter-app
cd counter-app
npm install
npm run dev

modify App.jsx In src folder

```
import React from 'react';
import Counter from './Counter'; // Ensure the correct import path
import './App.css'; // Optional, only if you have this file
function App() {
return (
<div className="App">
<h1>Welcome to Counter App</h1>
<Counter />
</div>
);
}
export default App;
```

create src/Counter.jsx in src folder

```
import React, { useState, useEffect } from 'react';
const Counter = () => {
const [count, setCount] = useState(0);
// Simulate fetching initial data for the Counter component
useEffect(() => {
setTimeout(() => {
setCount(0); // Set initial value after 2 seconds
}, 2000);
}, []);
return (
<div>
<h1>Counter: {count}</h1>
<button onClick={() => setCount(count + 1)}>Increment</button>
<button onClick={() => setCount(count - 1)}>Decrement</button>
<button onClick={() => setCount(count * 2)}>Double</button>
<button onClick={() => setCount(0)}>Reset</button>
</div>
);
};
export default Counter;
```

steps for next js

1.npx create-next-app@latest counter-app

Options to Select:

1.**Would you like to use TypeScript? No** (Choose "Yes" if you want TypeScript, but if you prefer simplicity, select "No").

2.**Would you like to use ESLint? Yes** (Recommended for maintaining code quality, especially in team projects).

3.**Would you like to use Tailwind CSS? No** (Select "Yes" if you plan to style your app using Tailwind CSS, but it's unnecessary for this Counter Experiment).

4.**Would you like your code inside a src/ directory? Yes** (This organizes your files better, making the project structure clean and scalable).

5.**Would you like to use App Router? (recommended) Yes** (This uses the latest Next.js routing features introduced in version 13).

Would you like to use Turbopack for next dev? Yes (Recommended for faster development builds; it's experimental but stable for many use cases).
6.**Would you like to customize the import alias (@/* by default)? No** (Stick with the default unless you have specific aliasing requirements).

TypeScript? No
ESLint? Yes
Tailwind CSS? No
Code inside `src/`? Yes
App Router? Yes
Turbopack for `next dev`? Yes
Import alias customization? No

Procedure

1.put Counter.jsx code In the folder like by creating Counter.js in src/app/Counter.js. and also add 'use client'; as first line in this file.

2.Moify the page.js content with src/App.Jsx by removing importing of App.jsx

3.create a folder like mkdir 6

4.cd 6

5.npm create vite@latest count

6.npm install

7.under src folder

8.App.jsx

```
import { useState } from 'react'
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'
import Statemgtclass from './Statemgtclass'
function App() {
return (
<div>
{/*<Statemgt/>*/}
<Statemgtclass/>
</div>
)
}
export default App
```

using functional component

9.Statemgt.jsx

```
import React from 'react'
import { useState } from 'react'
const Statemgt = () => {
const [number, setNumber] = useState(0)
const incrememt = () => {
setNumber(number + 1)
}
const decrememt = () => {
if (number > 0)
setNumber(number - 1)
}
const reset = () => {
setNumber(0)
}
const double = () => {
setNumber(number * 2)
}
return (
<div><h1>number</h1>
<h1>{number}</h1>
<br></br>
<button onClick={incrememt}>increment</button>
<button onClick={decrememt}>decrement</button>
<button onClick={double}>double</button>
<button onClick={reset}>reset</button>
</div>
)
}
export default Statemgt
```

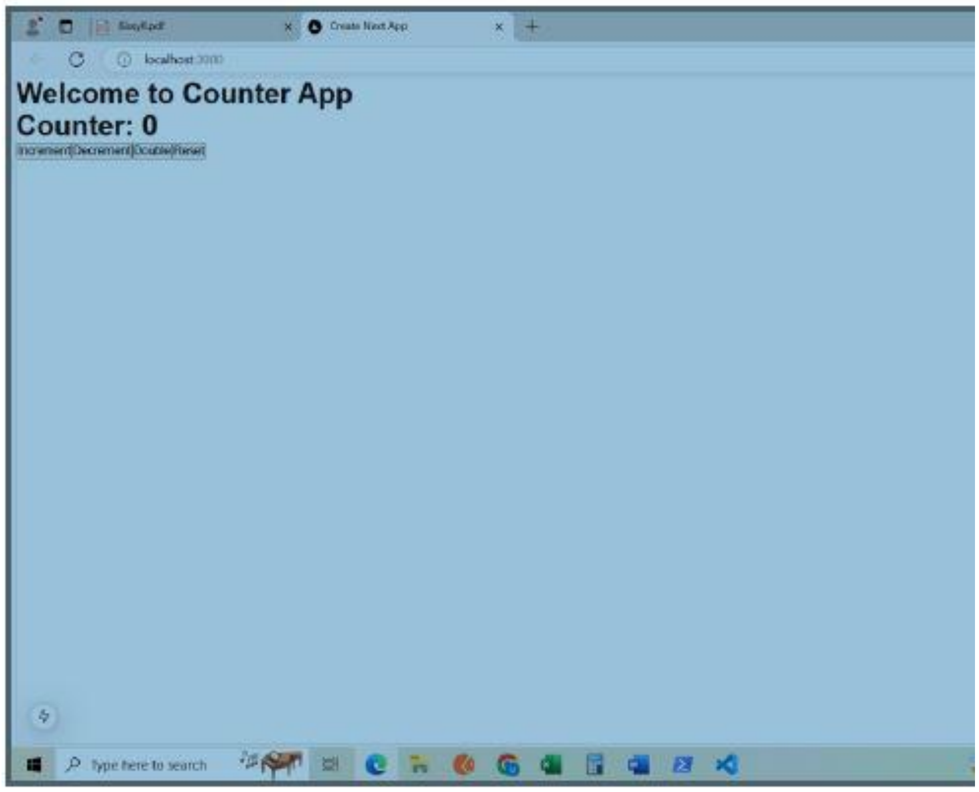
10.using class component

11.==================

12.Statemgtclass.jsx

```
import React, { Component } from 'react';
class Statemgtclass extends Component {
state = { number: 0 };
increment = () => {
const value = this.state.number + 1;
this.setState({ number: value });
};
decrement = () => {
if (this.state.number > 0) { }
const value = this.state.number - 1;
this.setState({ number: value });
};
reset = () => {
this.setState({ number: 0 });
};
double = () => {
const value = this.state.number * 2;
this.setState({ number: value });
};
render() {
return (
<div>
<h1>Number</h1>
<h1>{this.state.number}</h1>
<br />
<button onClick={this.increment}>Increment</button>
<button onClick={this.decrement}>Decrement</button>
<button onClick={this.double}>Double</button>
<button onClick={this.reset}>Reset</button>
</div>
);
}
}
export default Statemgtclass;
```

13.o/p: npm run dev



```
Welcome to Counter App
Counter: 0
```

Experiment 7: