Module 4

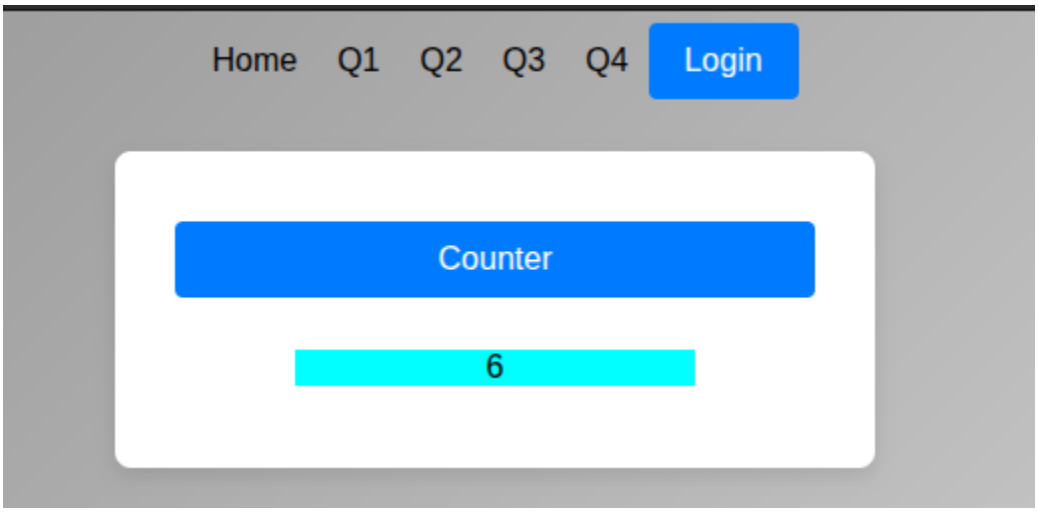
Exercise 2 (ReactJs States & Props)

Q 1.)Create a counter component with a button.

A 1.) React js

```
import {useState} from 'react';
import "./CounterButton.css"
function CounterButton(){

    const [count,counter] =useState(0)

    let incrementCounter=()=>{
        counter(count+1);
    }

    return(
        <div className="container">
            <button onClick={ incrementCounter}>
                Counter
            </button>

            <p id="display">
                {count}
            </p>
        </div>
    );
}

export default CounterButton;
```

Output



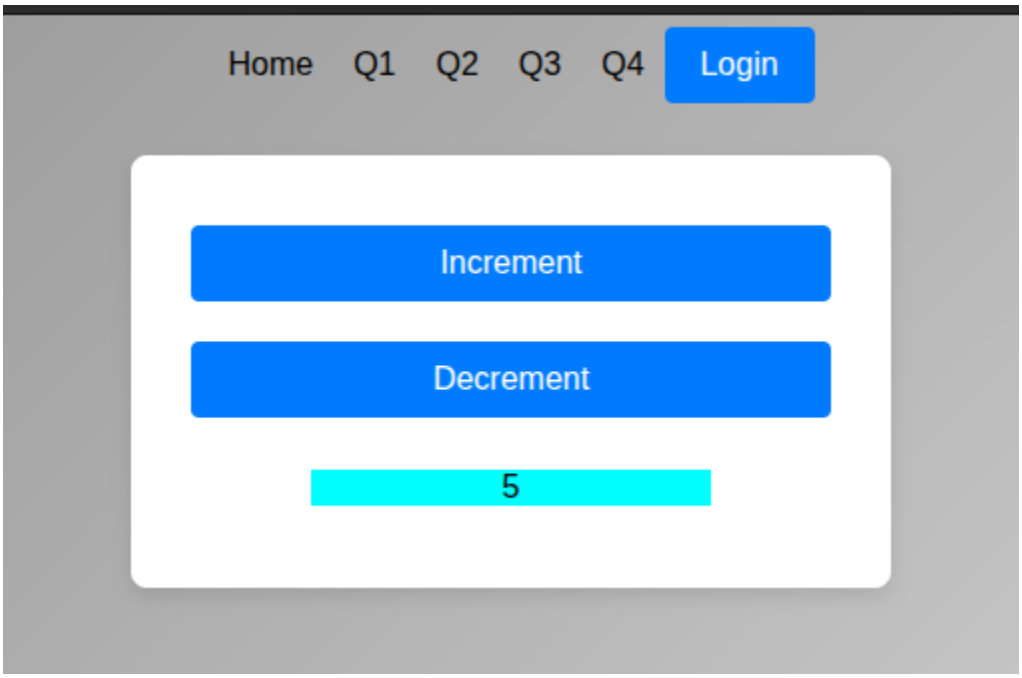
Q 2.)Create a counter with increment and decrement buttons.

A 2.) React js

Module4 > Exercise2 > q1 > src > components > IncrementDecrementCounter.jsx > IncrementDecrementCounter

```jsx
import {useState} from 'react';
import "./CounterButton.css"
function IncrementDecrementCounter(){

    const [count,setCount] =useState(0)

    let incrementCounter=()=>{
        setCount(count+1);
    }

    let decrementCounter=()=>{
        setCount(count-1);
    }

    return(
        <div className="container">
            <button onClick={ incrementCounter}>
                Increment
            </button>
            <button id="decrement-btn" onClick={ decrementCounter} disabled={count===0}>
                Decrement
            </button>

            <p id="display">
                {count}
            </p>
        </div>
    );
}

export default IncrementDecrementCounter;
```

Output



Q 3.)Create a parent() component that has a message in its state.

A 3.) React js

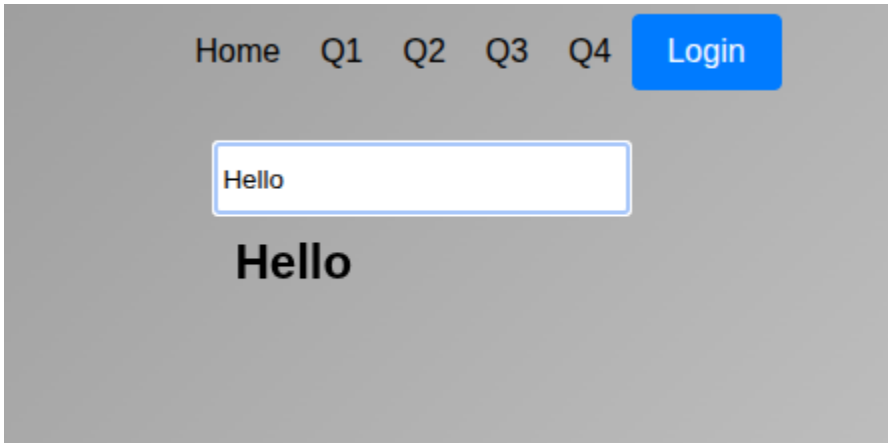
Parent()

```
import { useState } from "react";
import Child from "./Child";

function Parent(){
    const [message,setMessage]=useState("");
    let handleChange=(event)=>{
        setMessage(event.target.value);
    }

    return(
        <div className="q3-div">
            <input type="text" value={message} onChange={handleChange} placeholder="Type your messa
            <Child message={message}/>
        </div>

    );
}

export default Parent;
```

Child()

```jsx
function Child(props){
    return(
        <h2>{props.message}</h2>
    );
}

export default Child
```

Output



Q 4.)Create a **TodoList** component that stores a list of to-dos in its state.

A 4.) React js

TodoList

Module4 > Exercise2 > q1 > src > components > TodoList.jsx > TodoList

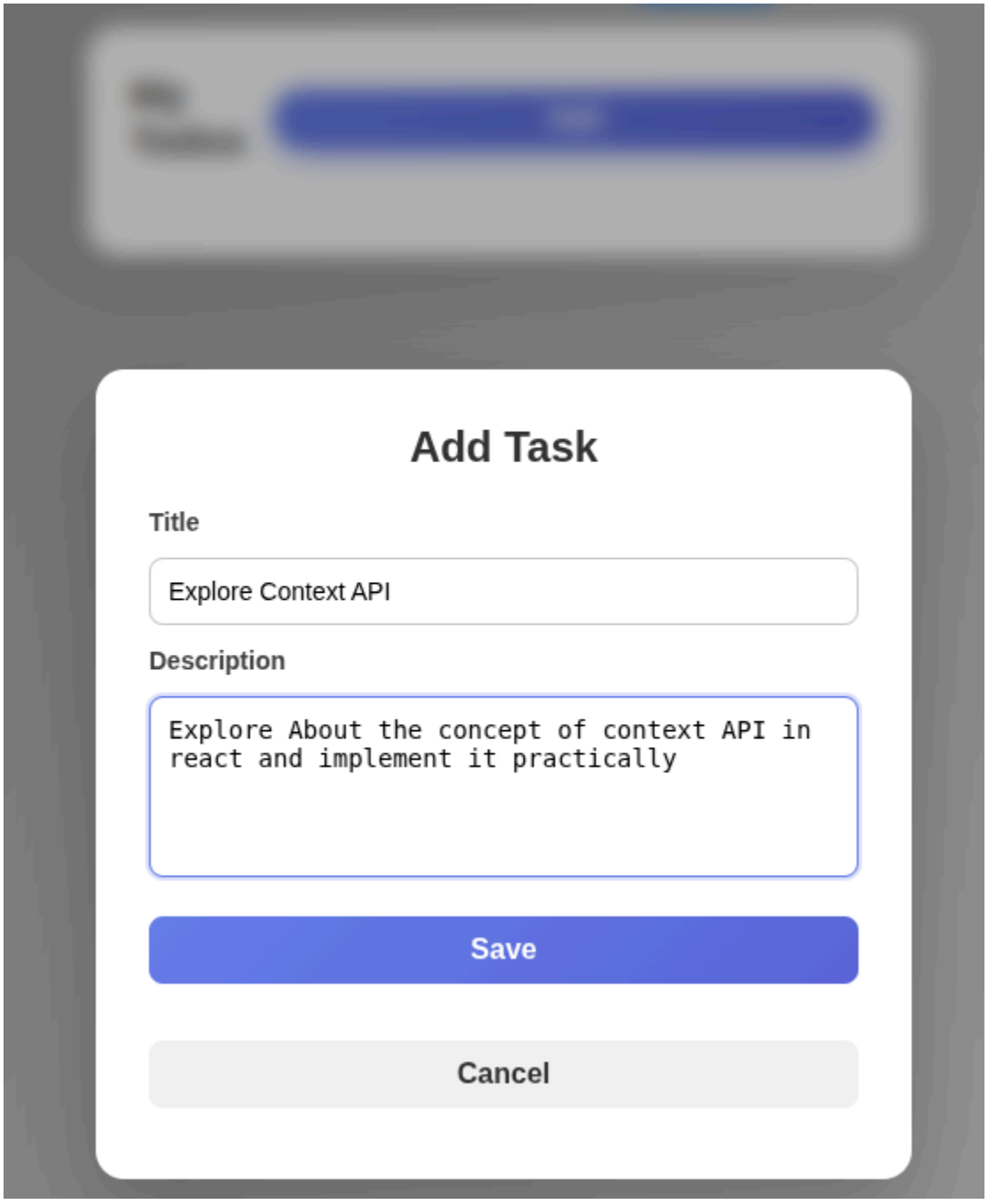
```jsx
import { useState } from "react";
import TodoForm from "./TodoForm";

function TodoList() {
    const [showForm, setShowForm] = useState(false);
    const [todo, setTodo] = useState([]);

    const handleTodo = (obj) => {
        setTodo((prevTodos) => [...prevTodos, obj]);
    }

    return (
        <div>
            <div className="todo">

                <div className="todo-header">
                    <h2>My Todos</h2>
                    <button className="add-btn" onClick={() => setShowForm(true)}>
                    Add
                    </button>
                </div>

                <ul>
                    {todo.map((item, index) => (
                        <li key={index}>
                            <h3>{item.title}</h3>
                            <p>{item.description}</p>
                        </li>
                    ))}
                </ul>
            </div>

            {showForm && (
                <div className="modal-overlay">
                    <TodoForm
                    onClose={() => setShowForm(false)}
                    handleTodo={handleTodo}
                    />
                </div>
            )}
```

TodoForm

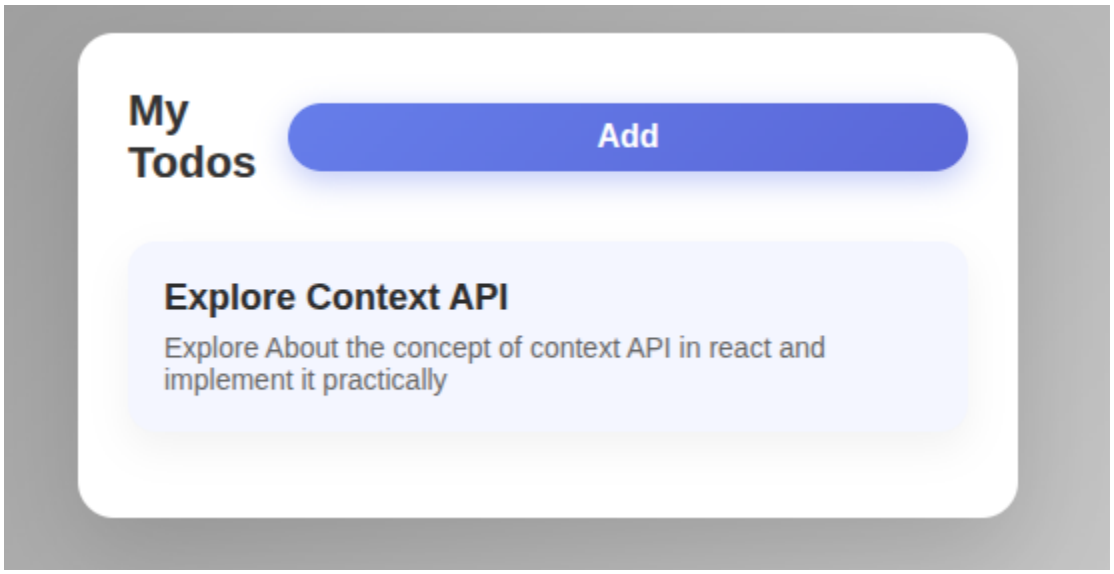

Module4 > Exercise2 > q1 > src > components > TodoForm.jsx > TodoForm
1 import { useState } from 'react';
2 import './TodoForm.css'
3
4 function TodoForm({ onClose, handleTodo }) {
5 const [title, setTitle] = useState("")
6 const [description, setDescription] = useState("")
7
8 const onSave = (e) => {
9 e.preventDefault();
10
11 const todo = {
12 title: title,
13 description: description
14 };
15
16 handleTodo(todo);
17 onClose();
18
19 setTitle("");
20 setDescription("");
21 }
22
23 return (
24
25 <h2>Add Task</h2>
26 <form onSubmit={onSave}>
27 <label htmlFor="title">Title</label>
28 <input
29 type="text"
30 name="title"
31 id="title"
32 value={title}
33 onChange={(e) => setTitle(e.target.value)}
34 required
35 />
36
37 <label htmlFor="description">Description</label>
38 <textarea
39 name="description"
40 id="description"

Output

TodoForm



TodoLIst



Q 5.)Create a component that toggles between two views: **"Login"** and **"Logout"**.

A 5.) React js

Login

```
import './login.css'
function Login({login}){

    return(
        <div className="login-container">
            <h2>Login Form</h2>

            <form>
                <label htmlFor="email">Email</label>
                <input type="email" name="email" id="email" required/>
                <label htmlFor="password">Password</label>
                <input type="password" name="password" id="password" required/>

                <button type="button" className="login-btn" onClick={login}>Login</button>
            </form>
        </div>
    );
}

export default Login;
```

Logout

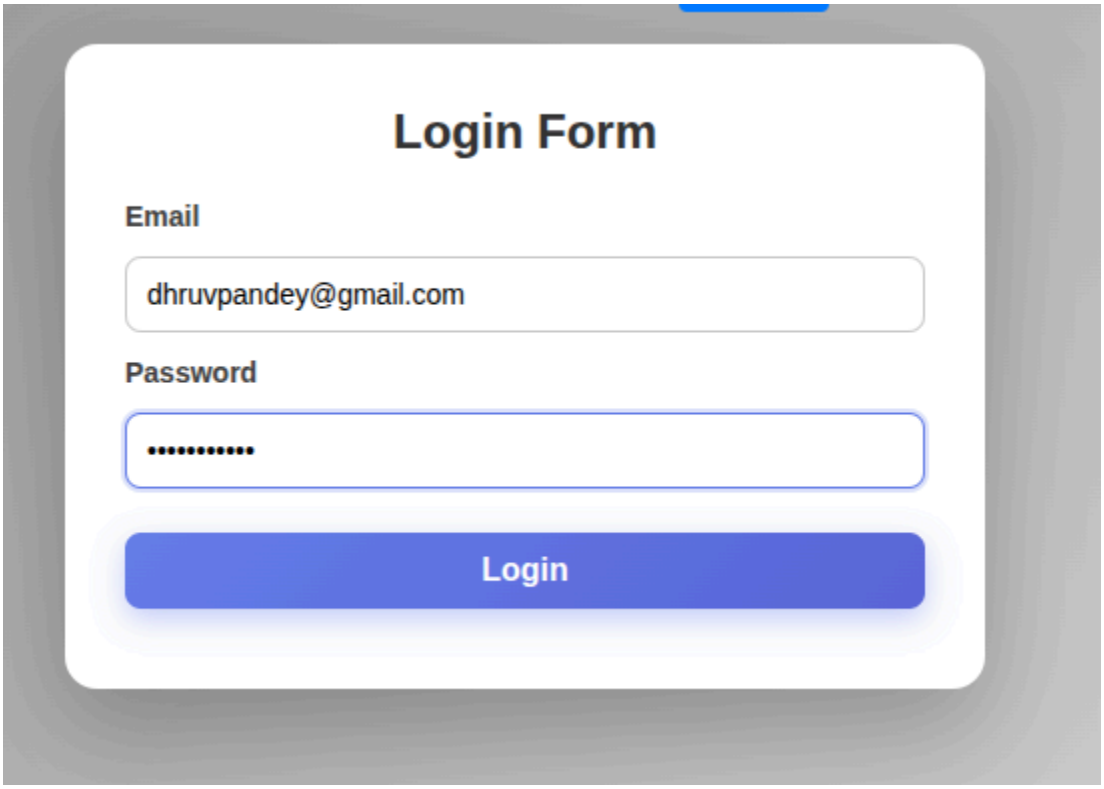
```jsx
function Logout(){
    return(
        <div>
            <h2>Logout</h2>
            <p>You have been logout please login again!</p>
        </div>
    );
}

export default Logout
```

App.jsx

```jsx
<div>
  <nav className="home">
    <Link to="/">Home</Link>
    <Link to="/q1">Q1</Link>
    <Link to="/q2">Q2</Link>
    <Link to="/q3">Q3</Link>
    <Link to="/q4">Q4</Link>
    {isLogin?(
    <button type="button" onClick={navigateLogout} className="btn">Logout</button>
    ):(<button type="button" onClick={navigateLogin} className="btn">Login</button>)
    }
  </nav>
```

Output

Login



Logout