**Assignment 3: State and Events**

**Introduce state in React components.**

**Handle events and update the state accordingly.**

In this assignment, we'll introduce state in React components and handle events to update the state accordingly. Let's modify our existing App.js file to include a state and handle events.

Step 1: Update App.js to Use State

Open the src/App.js file and modify it to include state. We'll use the useState hook to manage the state.

| // App.js  import React, { useState } from 'react';  import './App.css'; // Keep this line for styling, if needed  import Header from './Header';  import Footer from './Footer';  function App() {  const [appTitle, setAppTitle] = useState("React State and Events Assignment");  const [appAuthor, setAppAuthor] = useState("Your Name"); // Replace with your name  const [buttonText, setButtonText] = useState("Click me!");  // Event handler to update the state when the button is clicked  const handleButtonClick = () => {  setButtonText("Button clicked!");  };  return (  <div className="App">  <Header title={appTitle} />  <p>Main content of the app goes here.</p>  <button onClick={handleButtonClick}>{buttonText}</button>  <Footer author={appAuthor} />  </div>  );  }  export default App; |
| --- |

Step 2: Run Your React App

Save the changes and return to the terminal. Make sure you are still in the root directory of your React app (my-react-app). Start the development server with the following command:

npm start

Visit http://localhost:3000/ in your web browser. You should now see your React app displaying the header, main content, a button, and the footer. Clicking the button will update its text, demonstrating the handling of events and state updates.

Step 3: Explore Further

Feel free to explore further with state and events. You can add more interactive elements, handle different types of events, and experiment with different state values. Understanding how to manage state and handle events is crucial in React development and forms the basis for building dynamic and interactive user interfaces.

Congratulations! You have successfully introduced state in React components and handled events to update the state accordingly.