**Assignment Title**: Understanding React.js: State, Props and useEffect

**Objective**: To deepen understanding of React.js key features: State, Props, and useEffect.

**Project Details**:

Develop a simple "Post Viewer" app using React.js, showcasing the use of State, Props, and useEffect. The app should fetch posts from a mock API (like JSONPlaceholder), display them, and allow the user to select a post to view its details.

**Part 1 - Creating the Post List Component:**

1. Create a new React project using Create React App.
2. Create a **Posts** component. This component should have a state variable, **posts**, that will be used to store the list of posts.
3. Use the **useEffect** hook to fetch the list of posts from the API (<https://jsonplaceholder.typicode.com/posts>) when the component mounts. The fetched posts should be stored in the **posts** state variable.
4. Render a list of posts. Each post should be rendered as a button with the post's title. When a post's button is clicked, its **id** should be stored in a new state variable, **selectedPostId**.

**Part 2 - Creating the Post Detail Component:**

1. Create a **PostDetail** component. This component should accept a prop, **postId**.
2. The **PostDetail** component should have a state variable, **postDetail**, to store the detail of a post.
3. Use the **useEffect** hook in **PostDetail** to fetch the details of the post whenever the **postId** prop changes. Fetch the post detail from the API ([https://jsonplaceholder.typicode.com/posts/{postId}](https://jsonplaceholder.typicode.com/posts/%7BpostId%7D)). The fetched post detail should be stored in the **postDetail** state variable.
4. Render the post details.

**Part 3 - Combining the Components:**

1. In your **App** component, render the **Posts** component.
2. Also in the **App** component, conditionally render the **PostDetail** component only if **selectedPostId** is defined. Pass **selectedPostId** as a prop to **PostDetail**.

**Evaluation Criteria:**

1. Correct usage of state and props in React.
2. Correct and appropriate usage of the **useEffect** hook.
3. Successful fetching and rendering of data from the API.
4. Clean code and proper code structure.
5. Correct interaction between **Posts** and **PostDetail** components via state and props.