Walkthrough with React Part – 2

Assignment

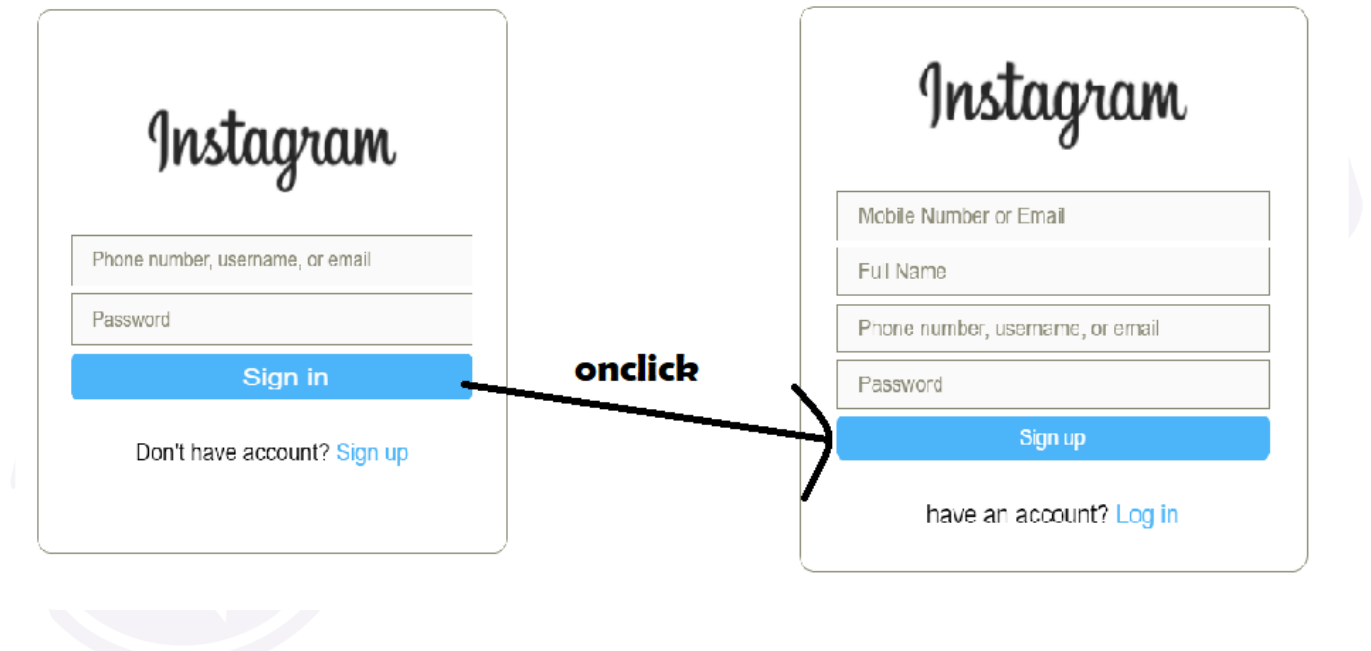**1: Creating a User Page like Instagram using React Functionalities**

Assignment Description:
In this assignment, you will create a user page similar to Instagram using React functionalities. The primary task is to implement conditional rendering, allowing users to either sign up or log in depending on their choice. The user interface should show different input fields based on their selection. If they choose to sign up, additional inputs for registration will be displayed. On the other hand, if they opt for login, only the basic username and password inputs should be visible.

Requirements:
1. Create a React application from scratch or use an existing one.
2. Implement conditional rendering to toggle between sign up and login views.
3. Design the user interface with a clean and intuitive layout similar to Instagram. Check below image for Your Reference

1. A well-organized React application with clear component structure.
2. The user interface should display the following components:
   a. Instagram Logo
   b. A button to switch between sign up and login views.
   c. Basic input fields for username and password.
   d. If the user chooses to sign up, additional input fields for name, email, etc., should be displayed

Guidelines:
1. Start by setting up a new React project or creating a new component for this assignment.
2. Use React hooks
3. Leverage CSS or a CSS framework to style the components and make the user interface visually appealing.
4. Test the application thoroughly to ensure all functionalities work as expected.
5. Document your code to explain any complex logic or reasoning behind your implementation.

2. Develop a web application using React that functions as a counter.
 Include two buttons in the UI:

1. **"Increment"** button:
• On clicking this button, the counter value should be incremented by one.

2. **"Decrement"** button:

• On clicking this button, the counter value should be decremented by one.
• Implement the counter logic using React's state management.
• Ensure that the counter value is displayed in the UI and updates in real-time when incremented or decremented.
• Use appropriate React components and hooks to manage the counter state and handle button click events.

See below for Reference