

## Requirement: Create Login and Registration Pages for E-Commerce Project

### Objective

To build and implement **Login** and **Registration** pages for an e-commerce project using React, while integrating with APIs. Interns will write their own code by understanding the requirements and taking reference from the provided instructions.

---

### Assignment Requirements

#### 1. Login Page

- **Functional Requirements:**
    - Create a LoginForm component with the following:
      - Input fields for **Username** and **Password**.
      - A **Submit** button to send user credentials to:
        - POST `http://localhost:8080/api/users/login`
      - Display appropriate feedback:
        - **Success message:** "Login successful!".
        - **Error message:** "Invalid username or password".
      - Pass the logged-in username to the parent component (App.js) to manage user sessions.
  - **Validation:**
    - Ensure both fields are filled before submission.
    - Display error messages for empty fields.
- 

#### 2. Registration Page

- **Functional Requirements:**
  - Create a SignUpForm component with the following:
    - Input fields for **Username**, **Password**, and optionally a **Role** (default: "USER").
    - A **Submit** button to send user data to:
      - POST `http://localhost:8080/api/users/register`
    - Display appropriate feedback:
      - **Success message:** "Registration successful!".
      - **Error message:** "Error registering user. Try again."

- **Validation:**
    - Username must have at least 4 characters.
    - Password must have at least 6 characters.
    - Display error messages for invalid input.
- 

### 3. App Integration

- **State Management:**
    - Use useState to manage:
      - The logged-in user (username).
      - Whether to show the login or registration page.
  - **Features:**
    - If no user is logged in:
      - Display the login page by default.
      - Allow switching to the registration page with a button (e.g., "Go to Register").
    - If a user is logged in:
      - Display a welcome message with their username.
      - Provide a logout button to clear the user session.
- 

### 4. Styling

- Use CSS to enhance the UI:
    - Center the forms on the page.
    - Add padding, margins, and colors to make the application visually appealing.
    - Use responsive design for mobile compatibility.
- 

### Development Steps

1. **Set Up Your Environment:**
  - Initialize a React project using create-react-app
  - Install axios for API integration:  
npm install axios
2. **Implement the Login Form:**

- Create a new component LoginForm.js
  - Write your own code to handle:
    - Input state management.
    - API integration using axios.
    - Feedback display (success or error messages).
  - 3. Implement the Registration Form:**
    - Create a new component SignUpForm.js
    - Write your own code to:
      - Handle form inputs.
      - Validate input data before submission.
      - Integrate with the registration API.
  - 4. Integrate with the Main Application:**
    - Update App.js to:
      - Toggle between login and registration forms.
      - Manage the logged-in user state.
      - Display appropriate content based on user login state.
  - 5. Test the Application:**
    - Ensure that:
      - Login works correctly and displays feedback.
      - Registration works correctly and displays feedback.
      - Logged-in state is maintained until logout.
- 

### Submission Guidelines

- Submit the project via a GitHub repository.
- Ensure the following:
  - Your code is clean and well-commented.
  - The app runs without errors when cloned.
  - Include a README.md with instructions to set up and run the application.

