

10. Create a Login and signup form in React

src/App.js

```
import React, { useState } from 'react';
import './App.css'; // Your custom CSS is imported here
import LoginForm from './components/LoginForm'; // Import the Login component
import SignupForm from './components/SignupForm'; // Import the Signup
component

function App() {
  const [isLoginView, setIsLoginView] = useState(true);
  const [isLoggedIn, setIsLoggedIn] = useState(false);

  const handleLogin = (success) => {
    if (success) {
      setIsLoggedIn(true);
    }
  };

  const handleSignup = (success) => {
    if (success) {
      alert('Sign up successful! Please log in.');
```

write this file only

```
      setIsLoginView(true); // Switch to login after successful sign-up
    }
  };

  const handleLogout = () => {
    setIsLoggedIn(false);
    alert('Logged out!');
  };

  // Helper functions to pass to the child components for switching views
  const switchToLogin = () => setIsLoginView(true);
  const switchToSignup = () => setIsLoginView(false);

  return (
    <div className="app-container">
      <h1>SIGN UP FORM</h1>

      { /* Conditional Rendering: Logged In vs. Forms */ }
      {isLoggedIn ? (
        // --- Logged In Dashboard ---
        <div className="dashboard">
          <h2>👋 Welcome! You are logged in.</h2>
          <p>This is your personalized dashboard content.</p>
          <button onClick={handleLogout} className="logout-btn">
            Log Out
```

```

        </button>
      </div>
    ) : (
      // --- Login/Signup Forms ---
      isLoginView ? (
        <LoginForm
          onLogin={handleLogin}
          onSwitchToSignup={switchToSignup} // Pass the switch function
        />
      ) : (
        <SignupForm
          onSignup={handleSignup}
          onSwitchToLogin={switchToLogin} // Pass the switch function
        />
      )
    )
  )}
</div>
);
}

export default App;

```

src/components/LoginForm.js

```

import React, { useState } from 'react';
// We assume App.css (or a global CSS file) is imported in App.js or index.js

const LoginForm = ({ onLogin, onSwitchToSignup }) => {
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    if (username && password) {
      alert(`Logging in with Username: ${username}`);
      onLogin(true); // Call the success handler in App.js
    } else {
      alert('Please enter both username and password.');
```

```

        type="text"
        id="login-username"
        placeholder="Name"
        value={username}
        onChange={(e) => setUsername(e.target.value)}
        required
      />
    </div>
    <div className="input-group-custom">
      <span className="icon">🔒</span> { /* Lock icon */}
      <input
        type="password"
        id="login-password"
        placeholder="Password"
        value={password}
        onChange={(e) => setPassword(e.target.value)}
        required
      />
    </div>
    <p className="forgot-password">
      New Member? <a href="#">SignUp 📧</a>
    </p>
    <div className="form-buttons-group">
      <button type="submit" className="submit-btn login-btn-custom">Login</button>
      <button type="button" onClick={onSwitchToSignup} className="switch-btn signup-switch-btn-custom">Sign Up</button>
    </div>
  </form>
);
};

export default LoginForm;

```

src/components/SignupForm.js

```

import React, { useState } from 'react';
// We assume App.css (or a global CSS file) is imported in App.js or index.js

const SignupForm = ({ onSignup, onSwitchToLogin }) => {
  const [name, setName] = useState('');
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    if (name && email && password) {
      alert(`Signing up with Name: ${name}, Email: ${email}`);
    }
  };

  return (
    <form>
      <div>
        <input type="text" value={name} onChange={e => setName(e.target.value)} />
        <input type="text" value={email} onChange={e => setEmail(e.target.value)} />
        <input type="password" value={password} onChange={e => setPassword(e.target.value)} />
      </div>
      <div>
        <button type="button" onClick={onSwitchToLogin}>Login</button>
        <button type="button" onClick={handleSubmit}>Sign Up</button>
      </div>
    </form>
  );
};

export default SignupForm;

```

```

    onSignup(true); // Call the success handler in App.js
  } else {
    alert('Please fill out all fields.');
```

```

  }
};
```

```

return (
  <form onSubmit={handleSubmit} className="auth-form signup-form-custom">
    <h2 className="form-title">Sign Up</h2>
    <div className="input-group-custom">
      <span className="icon">👤</span> { /* User icon */}
      <input
        type="text"
        id="signup-name"
        placeholder="Name"
        value={name}
        onChange={(e) => setName(e.target.value)}
        required
      />
    </div>
    <div className="input-group-custom">
      <span className="icon">✉️</span> { /* Email icon */}
      <input
        type="email"
        id="signup-email"
        placeholder="Email Id"
        value={email}
        onChange={(e) => setEmail(e.target.value)}
        required
      />
    </div>
    <div className="input-group-custom">
      <span className="icon">🔒</span> { /* Lock icon */}
      <input
        type="password"
        id="signup-password"
        placeholder="Password"
        value={password}
        onChange={(e) => setPassword(e.target.value)}
        required
      />
    </div>
    <p className="forgot-password">
      Existing Member? <a href="#">Login🔑!</a>
    </p>

    <div className="form-buttons-group">
```

```

        <button type="submit" className="submit-btn signup-btn-custom">Sign
Up</button>
        <button type="button" onClick={onSwitchToLogin} className="switch-btn
login-switch-btn-custom">Login</button>
    </div>
</form>
);
};

export default SignupForm;

```

src/App.css

```

body {
  margin: 0;
  font-family: 'Poppins', sans-serif; /* A font similar to the image */
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  background-color: #5d2b99; /* Deep purple background for the page */
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  padding: 20px;
}

.app-container {
  text-align: center;
  width: 100%;
}

h1 {
  color: white; /* Changed main title color */
  margin-bottom: 20px;
  font-size: 2.5em;
}

/* Auth Form Container (Card) */
.auth-form {
  background: white;
  padding: 40px 30px;
  border-radius: 20px; /* More rounded corners */
  margin: 40px auto 20px auto;
  box-shadow: 0 15px 30px rgba(0, 0, 0, 0.1); /* Stronger shadow */
  text-align: center; /* Center form content */
  position: relative; /* For potential absolute positioning later */
  max-width: 400px; /* Keep the max-width for desktop */
  width: 90%; /* <--- NEW: Ensures it shrinks on small screens */
}

```

```

    margin: 20px auto;
}

.form-title {
  color: #5d2b99; /* Purple title */
  font-size: 2.2em;
  margin-bottom: 30px;
  position: relative;
  display: inline-block; /* To allow pseudo-element underline */
}

.form-title::after { /* Underline effect */
  content: '';
  position: absolute;
  left: 50%;
  bottom: -10px;
  transform: translateX(-50%);
  width: 50px;
  height: 4px;
  background-color: #5d2b99;
  border-radius: 2px;
}

/* Input Groups (with icons) */
.input-group-custom {
  background-color: #f0f0f0; /* Light grey background for input fields */
  border-radius: 12px; /* Rounded input fields */
  margin-bottom: 25px;
  display: flex;
  align-items: center;
  padding: 0 15px;
  height: 60px; /* Taller input fields */
}

.input-group-custom .icon {
  font-size: 1.2em;
  color: #777; /* Icon color */
  margin-right: 15px;
}

.input-group-custom input {
  flex-grow: 1; /* Input takes up remaining space */
  border: none;
  background: transparent;
  padding: 10px 0;
  font-size: 1.1em;
  color: #333;
}

```

```

}

.input-group-custom input::placeholder {
  color: #a0a0a0;
}

.input-group-custom input:focus {
  outline: none;
}

/* Lost Password Link */
.forgot-password {
  text-align: right;
  margin-top: -10px; /* Move closer to inputs */
  margin-bottom: 30px;
  font-size: 0.9em;
}

.forgot-password a {
  color: #5d2b99; /* Purple link */
  text-decoration: none;
  font-weight: 500;
}

.forgot-password a:hover {
  text-decoration: underline;
}

/* Buttons Group */
.form-buttons-group {
  display: flex;
  justify-content: space-between; /* Space out the two buttons */
  margin-top: 40px;
}

.submit-btn, .switch-btn {
  padding: 15px 25px;
  border: none;
  border-radius: 30px; /* Heavily rounded buttons */
  cursor: pointer;
  font-size: 1.1em;
  font-weight: 600;
  transition: all 0.3s ease;
  flex-grow: 1; /* Allow buttons to grow */
  margin: 0 5px; /* Space between buttons */
}

/* Specific button styles from image */

```

```

.signup-btn-custom {
  background-color: #5d2b99; /* Purple */
  color: white;
}

.signup-btn-custom:hover {
  background-color: #4a237a;
}

.login-switch-btn-custom { /* The "Login" button next to "Sign Up" */
  background-color: #f0f0f0; /* Light grey */
  color: #5d2b99; /* Purple text */
}

.login-switch-btn-custom:hover {
  background-color: #e0e0e0;
}

/* Dashboard View (Adjusted for new background) */
.dashboard {
  padding: 40px;
  background: white;
  border-radius: 12px;
  max-width: 500px;
  margin: 50px auto;
  box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);
  color: #333; /* Darker text for readability on white background */
}

.dashboard h2 {
  color: #5d2b99; /* Purple for dashboard heading */
}

.logout-btn {
  padding: 10px 20px;
  background-color: #dc3545; /* Red */
  color: white;
  border: none;
  border-radius: 6px;
  cursor: pointer;
  margin-top: 20px;
  font-weight: 600;
  transition: background-color 0.3s;
}

.logout-btn:hover {
  background-color: #c82333;
}

```



```

/* Login Form Specific adjustments if needed */
.login-form-custom .login-btn-custom {
    background-color: #5d2b99; /* Purple for Login submit button */
    color: white;
}

.login-form-custom .login-btn-custom:hover {
    background-color: #4a237a;
}

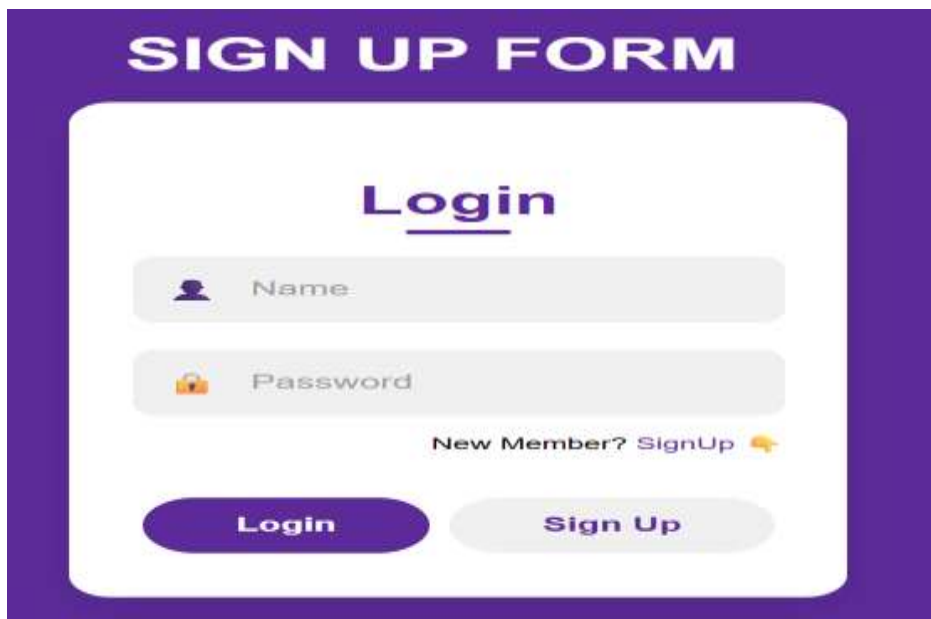
.login-form-custom .signup-switch-btn-custom {
    background-color: #f0f0f0; /* Light grey for Signup switch button */
    color: #5d2b99; /* Purple text */
}

.login-form-custom .signup-switch-btn-custom:hover {
    background-color: #e0e0e0;
}

/* If you need to use a custom font, add this to your index.html <head> or
import it in App.css */
/* <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&di
splay=swap" rel="stylesheet"> */


```


OUTPUT:




SIGN UP FORM

Login

 Name

 Password

New Member? [SignUp](#) 

Login **Sign Up**

SIGN UP FORM

Sign Up



Name



Email Id



Password

Existing Member? Login  !

Sign Up

Login