

Signup.jsx:

```
import React, { useState } from 'react';
import './signup.css'

const SignupForm = () => {
  // Form state
  const [formData, setFormData] = useState({
    username: '',
    email: '',
    password: '',
    confirmPassword: '',
  });

  const [errors, setErrors] = useState({});

  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData({ ...formData, [name]: value });
  };

  // Handle form submission
  const handleSubmit = (e) => {
    e.preventDefault();

    // Simple form validation
    const newErrors = {};
    if (!formData.username.trim()) {
      newErrors.username = 'Username is required';
    }
    if (!formData.email.trim()) {
      newErrors.email = 'Email is required';
    }
    if (!formData.password.trim()) {
      newErrors.password = 'Password is required';
    }
    if (formData.password !== formData.confirmPassword) {
      newErrors.confirmPassword = 'Passwords do not match';
    }
  }
}
```

```
// If there are errors, update the state and prevent submission
if (Object.keys(newErrors).length > 0) {
  setErrors(newErrors);
  return;
}

// If validation passes, you can proceed with form submission
// For example, you can send the form data to a server or perform
other actions
console.log('Form submitted:', formData);
};

return (
  <div className="aod">

    <form onSubmit={handleSubmit}>
      <div>
        <label htmlFor="username">Username:</label>
        <input
          type="text"
          id="username"
          name="username"
          value={formData.username}
          onChange={handleChange}
        />
        {errors.username && <span>{errors.username}</span>}
      </div>

      <div>
        <label htmlFor="email">Email:</label>
        <input
          type="email"
          id="email"
          name="email"
          value={formData.email}
          onChange={handleChange}
        />
        {errors.email && <span>{errors.email}</span>}
      </div>
    </form>
  </div>
);
```

```

<div>
  <label htmlFor="password">Password:</label>
  <input
    type="password"
    id="password"
    name="password"
    value={formData.password}
    onChange={handleChange}
  />
  {errors.password && <span>{errors.password}</span>}
</div>

<div>
  <label htmlFor="confirmPassword">Confirm Password:</label>
  <input
    type="password"
    id="confirmPassword"
    name="confirmPassword"
    value={formData.confirmPassword}
    onChange={handleChange}
  />
  {errors.confirmPassword && <span>{errors.confirmPassword}</span>}
</div>

  <button type="submit" Link to='#'className='but'>Sign Up</button>
</form>
</div>
);
};

export default SignupForm;

```

Signup.css:

```

.aod {
  font-family: 'Arial', sans-serif;
  background-color: #f5f5f5;

```

```
background-size: cover;
background-image:
url('https://img.freepik.com/free-vector/hospital-ward-with-patient-bed-do
ctor_107791-16654.jpg?w=1380&t=st=1706072423~exp=1706073023~hmac=3f276db5a
16df45c9210137aceac6039c0b27333575290ba988d466a5022910e');
display: flex;
align-items: center;
justify-content: center;
height: 100vh;
margin: 0;
position: fixed;
width: 100%;
margin-top: -.6cm;
margin-left: -.3cm;
margin-right: -.2cm;
}

form {
background-color: #fff;
border-radius: 8px;
box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
padding: 30px;
width: 400px;
max-width: 100%;
box-sizing: border-box;
}

div {
margin-bottom: 20px;
}

label {
display: block;
font-weight: bold;
margin-bottom: 8px;
color: #333;
}

input {
width: 100%;
```

```
padding: 12px;
border: 1px solid #ddd;
border-radius: 4px;
box-sizing: border-box;
font-size: 16px;
}

span {
  color: #e74c3c;
  font-size: 14px;
  margin-top: 5px;
  display: block;
}

.but {
  background-color: #3498db;
  color: #fff;
  padding: 12px 20px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-size: 16px;
  transition: background-color 0.3s ease;
}

button:hover {
  background-color: #2980b9;
}

/* Optional: Add media queries for responsiveness */
/* @media screen and (max-width: 600px) {
  form {
    width: 100%;
  }
}
*/
```