PRACTICAL 4
Form Handling in React Using Controlled Components

App.js

```
import React, { useState } from 'react';

export default function App() {
  const [formData, setFormData] = useState({
    name: '',
    email: '',
    password: ''
  });

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

  const handleSubmit = (event) => {
    event.preventDefault();
    console.log('Form data submitted:', formData);
    setFormData({
      name: '',
      email: '',
      password: ''
    });
  };

  return (
    <div>
      <style>{`
        body {
          font-family: Arial, sans-serif;
          background-color: #f4f4f9;
          display: flex;
          justify-content: center;
          align-items: center;
          height: 100vh;
          margin: 0;
        }
        .form-container {
          background-color: white;
          padding: 2rem;
          border-radius: 10px;
          box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
          width: 100%;
          max-width: 400px;
          box-sizing: border-box;
        }
        h2 {
          text-align: center;
          color: #333;
          margin-bottom: 1.5rem;
```

```
      }
    .form-group {
      margin-bottom: 1rem;
    }
    .form-group label {
      display: block;
      margin-bottom: 0.5rem;
      color: #555;
    }
    .form-group input {
      width: 100%;
      padding: 0.75rem;
      border: 1px solid #ccc;
      border-radius: 5px;
      box-sizing: border-box;
    }
    .form-group input:focus {
      outline: none;
      border-color: #007bff;
      box-shadow: 0 0 0 3px rgba(0, 123, 255, 0.25);
    }
    button {
      width: 100%;
      padding: 0.75rem;
      border: none;
      border-radius: 5px;
      background-color: #007bff;
      color: white;
      font-size: 1rem;
      cursor: pointer;
      transition: background-color 0.2s;
    }
    button:hover {
      background-color: #0056b3;
    }
  `}</style>

  {/* Form */}
  <div className="form-container">
    <h2>Register</h2>
    <form onSubmit={handleSubmit}>
      {/* Name Input Field */}
      <div className="form-group">
        <label htmlFor="name">Name</label>
        <input
          type="text"
          id="name"
          name="name"
          value={formData.name}
          onChange={handleChange}
          required
        />
      </div>

      {/* Email Input Field */}
      <div className="form-group">
        <label htmlFor="email">Email</label>
```

```jsx
        <input
          type="email"
          id="email"
          name="email"
          value={formData.email}
          onChange={handleChange}
          required
        />
      </div>

      {/* Password Input Field */}
      <div className="form-group">
        <label htmlFor="password">Password</label>
        <input
          type="password"
          id="password"
          name="password"
          value={formData.password}
          onChange={handleChange}
          required
        />
      </div>

      {/* Submission Button */}
      <button type="submit">Submit</button>
    </form>
  </div>
</div>
  );
}
```
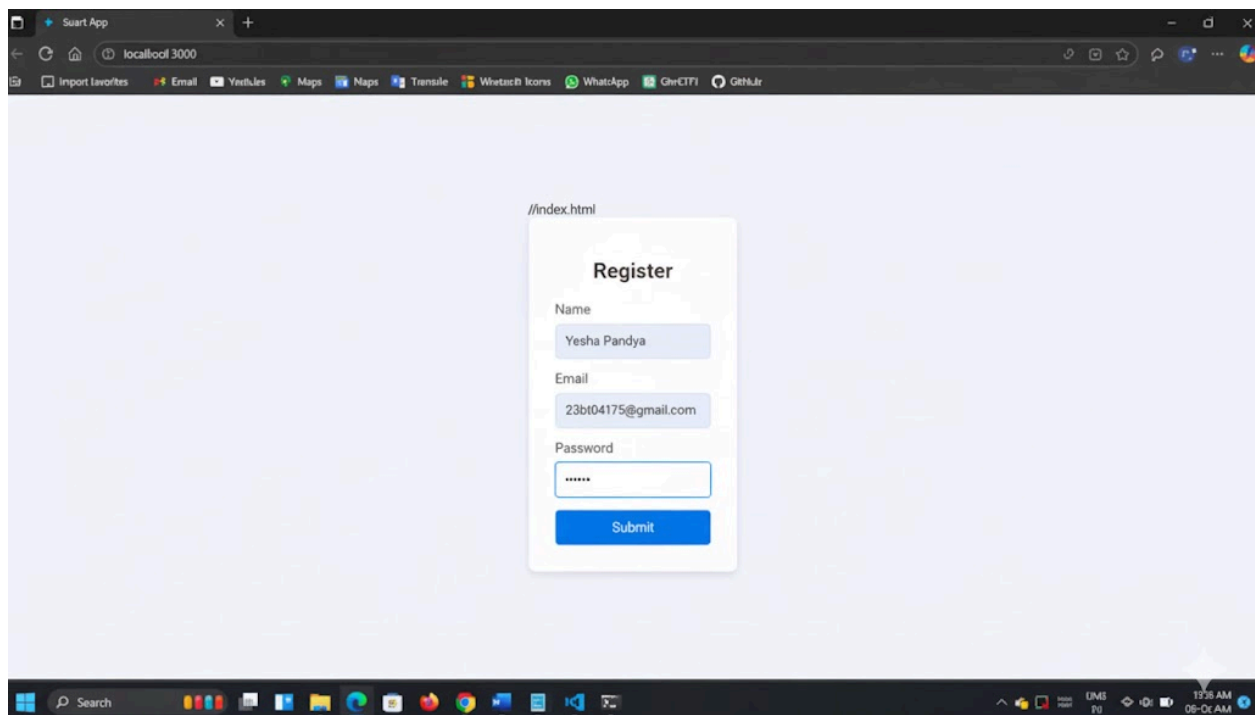

****