# EXPERIMENT 7

**AIM:** React JS – Basic components, forms, events, hook, styling.

A simple ReactJS application that demonstrates the use of **basic components**, **forms**, **events**, **hooks**, and **styling** using inline styles and CSS classes.

We'll break it down into:

1. **Basic Components** - Functional components in React.
2. **Forms** - Handling user input with forms.
3. **Events** - Handling events like button clicks.
4. **Hooks** - Using useState and useEffect.
5. **Styling** - Inline styles and CSS classes.

**Steps to Setup and Run the Example:**

**Create a new React App:**
If you haven't already, create a new React app using Create React App:

```
npx create-react-app my-app
cd my-app
```

1. **Replace src/App.js content:**
   Open src/App.js and replace its contents with the code below:

j

```
import React, { useState, useEffect } from 'react';
import './App.css';  // Import the CSS file
const App = () => {
  // States for handling form input and submitted data
  const [name, setName] = useState('');
  const [email, setEmail] = useState('');
  const [submittedData, setSubmittedData] = useState(null);
  // useEffect hook to log when the component mounts
  useEffect(() => {
    console.log('App component mounted.');
  }, []);
```

```
// Event handler for form submission
const handleSubmit = (e) => {
  e.preventDefault();  // Prevents page reload on form submission
  setSubmittedData({ name, email });
  setName(");  // Reset input fields
  setEmail(");
};
// Inline styles example
const inlineStyles = {
  container: {
    display: 'flex',
    flexDirection: 'column',
    alignItems: 'center',
    padding: '20px',
  },
  form: {
    display: 'flex',
    flexDirection: 'column',
    gap: '10px',
    marginBottom: '20px',
  },
  input: {
    padding: '8px',
    fontSize: '16px',
    borderRadius: '4px',
    border: '1px solid #ccc',
  },
  button: {
    padding: '10px 20px',
    backgroundColor: '#4CAF50',
    color: 'white',
    border: 'none',
    borderRadius: '5px',
    cursor: 'pointer',
  },
  greetingButton: {
    marginTop: '20px',
  },
};
return (
    <div style={inlineStyles.container}>
      <h1>React Form with Hooks</h1>
      {/* Form Section */}
      <form style={inlineStyles.form} onSubmit={handleSubmit}>
        <input
          type="text"
          placeholder="Enter your name"
          value={name}
          onChange={(e) => setName(e.target.value)}
          style={inlineStyles.input}
        />
        <input
          type="email"
          placeholder="Enter your email"
          value={email}
          onChange={(e) => setEmail(e.target.value)}
          style={inlineStyles.input}
        />
        <button type="submit" style={inlineStyles.button}>Submit</button>
      </form>
      {/* Show submitted data if available */}
      {submittedData && (
        <div className="submitted-data">
          <h3>Submitted Data:</h3>
          <p>Name: {submittedData.name}</p>
          <p>Email: {submittedData.email}</p>
        </div>
      )}
      {/* Handling Events - Greeting button */}
      <button
        onClick={() => alert(`Hello, ${name || 'Guest'}!`)}
        style={{ ...inlineStyles.button, ...inlineStyles.greetingButton }}
      >
        Greet Me
      </button>
    </div>
  );
};
export default App;
```

**3. Create and Edit src/App.css:**

Create a new file called App.css in the src folder, and paste this CSS code:

css

Copy code

```
.submitted-data {
  background-color: #f4f4f4;
  padding: 10px;
  border-radius: 5px;
  border: 1px solid #ccc;
  margin-top: 20px;
  width: 80%;
  max-width: 400px;
  text-align: center;
}
h1 {
  color: #333;
  font-size: 2rem;
  text-align: center;
}
```

**4. Run Your React Application:**

In your terminal, inside the project directory (my-app), run the following command to start the development server:

bash

Copy code

```
npm start
```

This will open a browser window showing your application. If it doesn't open automatically, navigate to http://localhost:3000.

**What Should You See?**

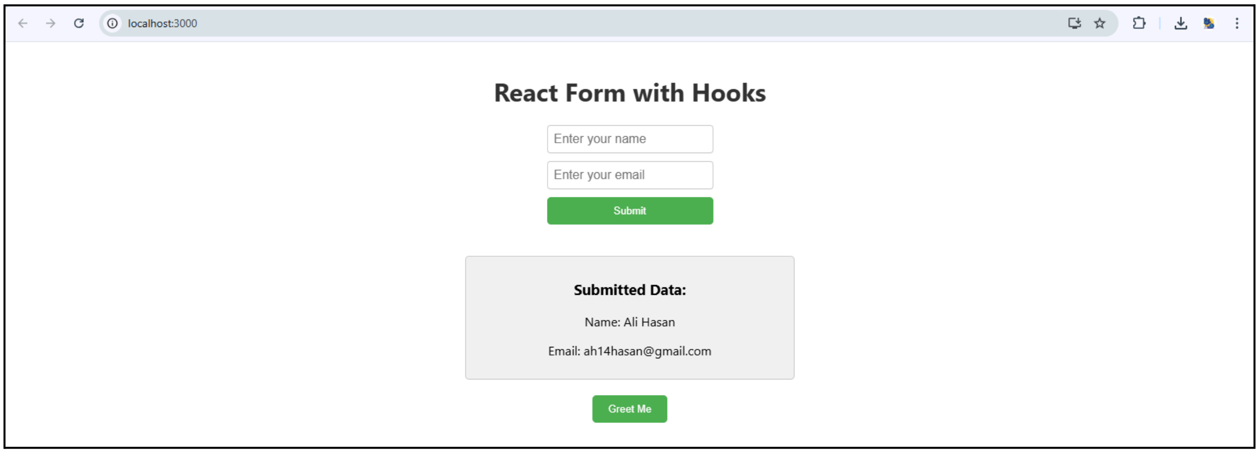
- **The title**: "React Form with Hooks".
- **Two input fields**: one for entering a name and another for entering an email.
- **A submit button**: Clicking this will store the data and display it below the form.
- **A greet me button**: Clicking this will show an alert with the entered name (or "Guest" if no name is provided).

If you submit the form, the submitted data will appear below the form with the name and email you entered.

**Common Issues:**

- **No data displayed after form submission**: Ensure that the handleSubmit function correctly sets the state (setSubmittedData) with the correct data and that submittedData is properly rendered.
- **CSS not applying**: Make sure you have imported the CSS file correctly (import './App.css'; in App.js).
- **Inputs not resetting after form submission**: Make sure setName(") and setEmail(") are called after the form is submitted to reset the form fields.

**OUTPUT:**

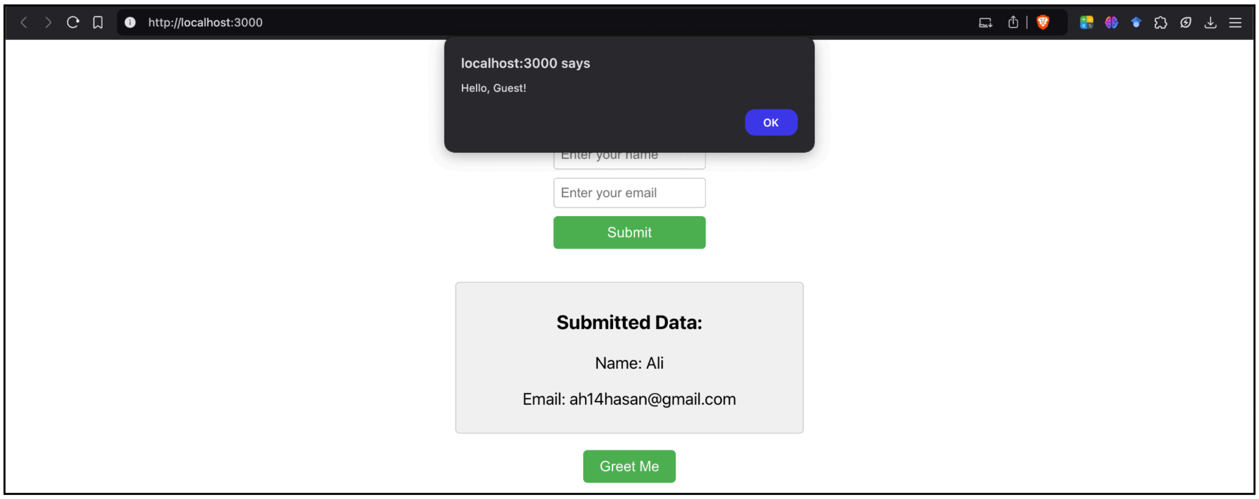


React Form with Hooks

Submitted Data:

Name: Ali Hasan

Email: ah14hasan@gmail.com



localhost:3000 says

Hello, Guest!

Submitted Data:

Name: Ali

Email: ah14hasan@gmail.com