

Create a folder called “Experiment 7” on the desktop.

Open the Experiment 7 folder in VS code.

Open New terminal in VS code.

Check if node js is installed or not using “**node -v**” and “**npm -v**” commands.

Enter the following commands in the terminal-:

Creating “my-app” folder-:

**npx create-react-app my-app**

Get inside the my-app folder-:

**cd my-app**

In my-app → src → **App.js**:

Paste the following code-:

```
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;

```

Paste the following code in my-app → src → **App.css**:

```

.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;
}
```

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

### **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.