

Assignment

Learner Details

- **Name:** Shrayanth S
 - **Enrollment Number:** SU625MR011
 - **Batch / Class:** June 2025 MERN
 - **Assignment:** Contact Form
 - **Date of Submission:** 05/07/2025
-

Problem Solving Activity 1.1

1. Program Statement

Create a React-based contact form that accepts user input for Name, Email, and Contact Number. The form must display these values in an alert message upon submission. The form must be styled and centered on the screen using CSS. Inputs must be responsive to user changes using React's useState hook.

2. Algorithm

- Initialize three state variables for name, email, and contact.
 - Create input fields for each of the three values.
 - Bind each input field to its respective state variable and update them on change.
 - On form submission:
 - Prevent default form behavior.
 - Display an alert showing the collected values.
 - Apply CSS to center and style the form.
-

3. Pseudocode

Start

Import necessary modules from React

Define Form component

Initialize name, email, contact as state variables

Define functions to handle input changes

Define submit handler:

- Show alert with entered name, email, contact

Return JSX for:

- A styled form with input fields and submit button

Style the form using CSS:

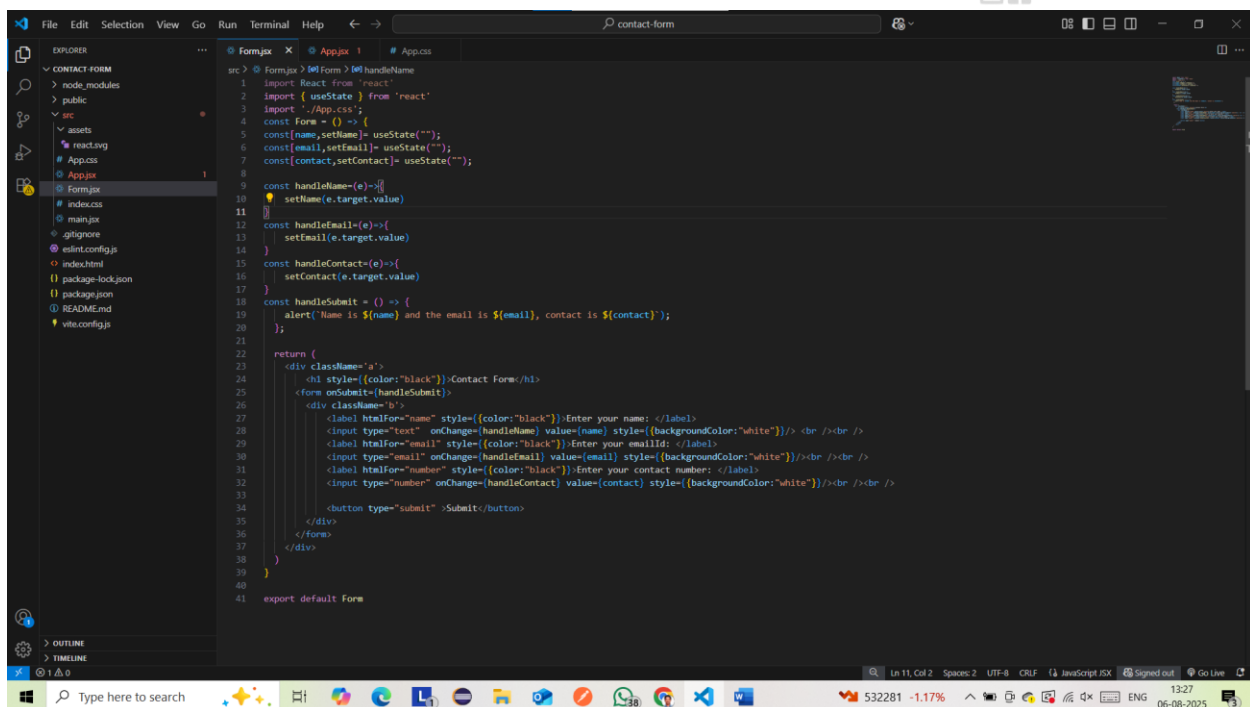
- Center the form
- Add padding, background, and borders

Render Form component in App.jsx

End

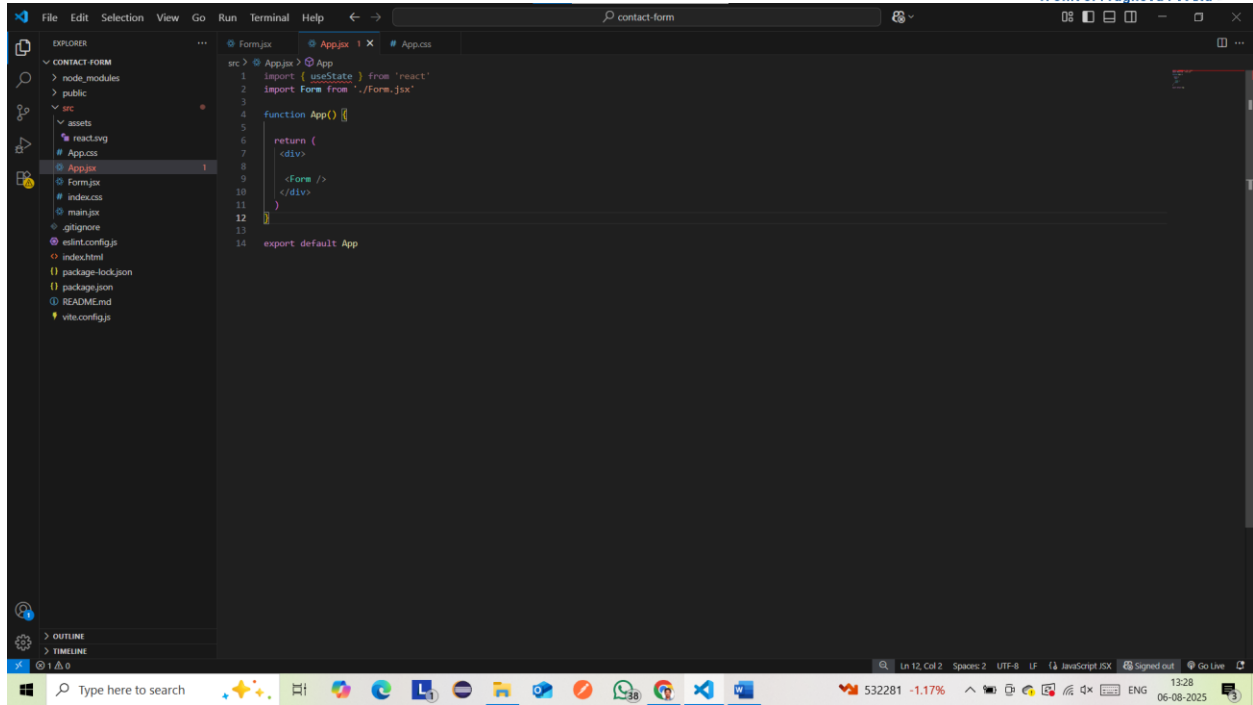


4. Program Code



```

1  import React from 'react'
2  import { useState } from 'react'
3  import './App.css'
4  const Form = () => {
5    const [name, setName] = useState('');
6    const [email, setEmail] = useState('');
7    const [contact, setContact] = useState('');
8
9    const handleName = (e) => {
10     setName(e.target.value)
11   }
12   const handleEmail = (e) => {
13     setEmail(e.target.value)
14   }
15   const handleContact = (e) => {
16     setContact(e.target.value)
17   }
18   const handleSubmit = () => {
19     alert('Name is ${name} and the email is ${email}, contact is ${contact}');
20   };
21
22   return (
23     <div className="a">
24       <div style={{color:'black'}}>Contact Form:</div>
25       <form onSubmit={handleSubmit}>
26         <div className="b">
27           <label htmlFor="name" style={{color:'black'}}>Enter your name:</label>
28           <input type="text" onChange={handleName} value={name} style={{backgroundColor:'white'}}/> <br /><br />
29           <label htmlFor="email" style={{color:'black'}}>Enter your email:</label>
30           <input type="email" onChange={handleEmail} value={email} style={{backgroundColor:'white'}}/> <br /><br />
31           <label htmlFor="number" style={{color:'black'}}>Enter your contact number:</label>
32           <input type="number" onChange={handleContact} value={contact} style={{backgroundColor:'white'}}/> <br /><br />
33           <button type="submit">Submit</button>
34         </div>
35       </form>
36     </div>
37   )
38 }
39
40
41 export default Form
  
```

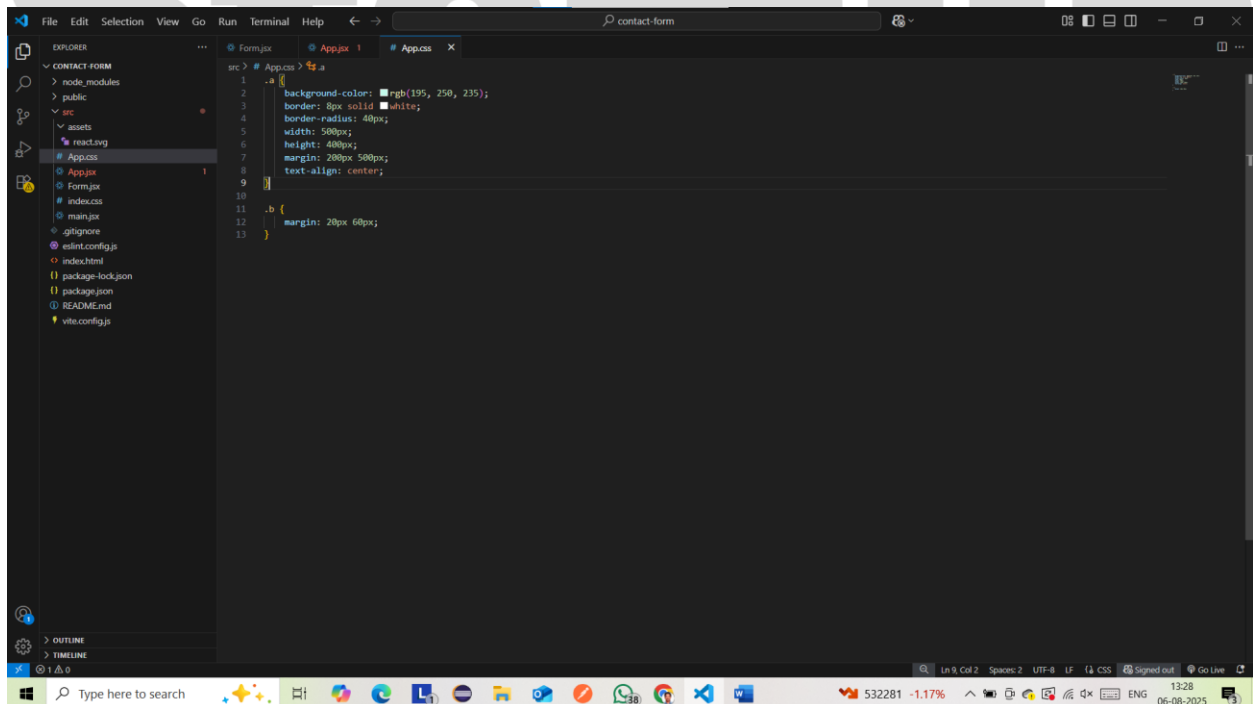


The screenshot shows the VS Code editor interface for a project named 'contact-form'. The Explorer sidebar on the left displays the file structure, including folders like 'node_modules', 'public', 'src', and 'assets', and files like 'App.jsx', 'Form.jsx', 'index.css', 'main.jsx', 'index.html', 'package-lock.json', 'package.json', 'README.md', 'vite.config.js', and 'eslint.config.js'. The main editor area shows the content of 'App.jsx' with the following code:

```

src > App.jsx > App
1 import { useState } from 'react'
2 import Form from './form.jsx'
3
4 function App() {
5
6   return (
7     <div>
8       <Form />
9     </div>
10   )
11 }
12
13
14 export default App
  
```

The status bar at the bottom indicates the file is 'Ln 12, Col 2', uses 'Spaces: 2', 'UTF-8' encoding, and 'JavaScript JSX' syntax. It also shows the user is 'Signed out' and a 'Go Live' button is available. The system tray at the very bottom shows the date as '06-08-2025' and time as '13:28'.



This screenshot shows the same VS Code editor interface, but with the 'App.css' file open in the main editor. The Explorer sidebar remains the same. The content of 'App.css' is as follows:

```

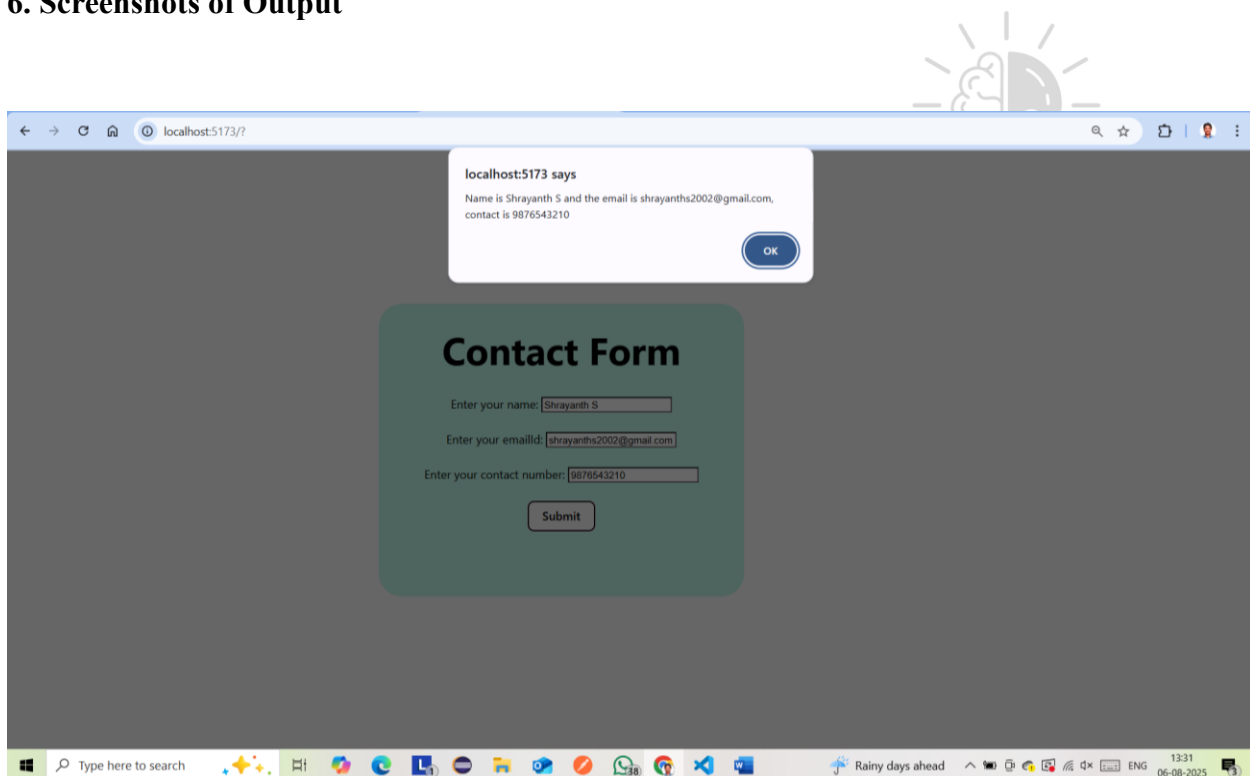
src > App.css > a
1
2 .a {
3   background-color: #rgb(195, 250, 235);
4   border: 8px solid #white;
5   border-radius: 40px;
6   width: 500px;
7   height: 400px;
8   margin: 200px 500px;
9   text-align: center;
10
11 .b {
12   margin: 20px 60px;
13 }
  
```

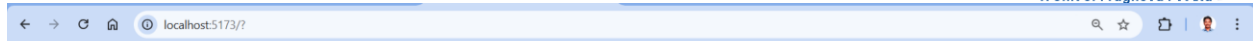
The status bar at the bottom now indicates the file is 'Ln 9, Col 2' and uses 'CSS' syntax. The system tray at the bottom remains the same, showing '06-08-2025' and '13:28'.

5. Test Cases

Test Case	Input (Name, Email, Contact)	Expected Output (Alert)
TC1	Shrayanth, shrayanth2002@gmail.com, 9876543210	"Name is Shrayanth, email is shrayanth2002@gmail.com, contact is 9876543210"

6. Screenshots of Output





Contact Form

Enter your name:

Enter your emailid:

Enter your contact number:



7. Observation / Reflection

The main challenge was keeping the form in the center of the screen using CSS, and handling the input values properly with state. I learned how to use useState, handle forms, and apply basic CSS in React. Next time, I will try to add form validation, show a success message on the page, and make the form work better on mobile.

A Unit of Pragnova Pvt Ltd

