

Lesson 03 Demo 02

Creating a Form with Three Input Fields: Name, Email, and Message

Objective: To develop a form with three input fields: Name, Email and Message

Tools Required: Node terminal, React app, and Visual Studio Code

Prerequisites: Knowledge of creating a React app and an understanding of the folder structure

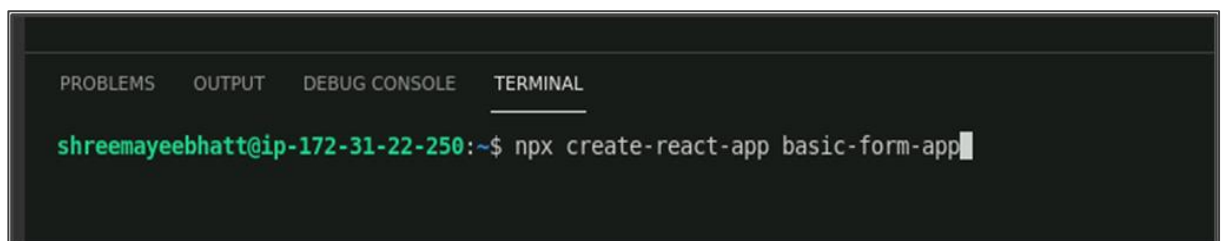
Steps to be followed:

1. Create a new React app
2. Configure the App.js file
3. Create a new file called BasicFormHandling.js
4. Run the app and verify the functionality

Step 1: Create a new React app

1.1 Open the terminal and run the following command to create a new **React** app:

npx create-react-app basic-form-app



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
shreemayeebhatt@ip-172-31-22-250:~$ npx create-react-app basic-form-app
```

Note: This will create a new **React** app in a directory named **basic-form-app**.

1.2 Move to the newly created project directory by running the **cd basic-form-app** command

Step 2: Configure the App.js file

2.1 Open the React project in the Visual Studio code and navigate through the project directory of **basic-form-app** to open the **src/App.js** file and replace the code in **App.js** with the following code:

```
import React from 'react';
import BasicFormHandling from './BasicFormHandling';
```

```
function App() {
  return (
    <div>
      <h2>Basic Form Handling</h2>

      <BasicFormHandling />
    </div>
  );
}
```

```
export default App;
```

A screenshot of a code editor with a dark background. The text is as follows:

```
src > JS App.js > ...
1  import React from 'react';
2  import BasicFormHandling from './BasicFormHandling';
3
4  function App() {
5    return (
6      <div>
7        <h2>Basic Form Handling</h2>
8        <BasicFormHandling />
9      </div>
10   );
11 }
12
13 export default App;
14
```

Note: This code defines the **App** component that renders the **BasicFormHandling** component.

Step 3: Create a new file called BasicFormHandling.js

3.1 In your code editor, create a new file called **BasicFormHandling.js** inside the **src** directory

3.2 Copy and paste the provided code into the **BasicFormHandling.js** file

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

```
function BasicFormHandling() {  
  const [formValues, setFormValues] = useState({  
    name: "",  
    email: "",  
    message: "",  
  });  
  const [errors, setErrors] = useState({  
    name: false,  
    email: false,  
    message: false,  
  });  
  const handleChange = (event) => {  
    const { name, value } = event.target;  
    setFormValues({ ...formValues, [name]: value });  
    setErrors({ ...errors, [name]: false });  
  };  
  const handleSubmit = (event) => {  
    event.preventDefault();  
    // Check if any input fields are empty  
    let formValid = true;  
    const errorsCopy = { ...errors };  
    for (const field in formValues) {  
      if (!formValues[field]) {  
        formValid = false;  
        errorsCopy[field] = true;  
      }  
    }  
    setErrors(errorsCopy);  
  }  
}
```

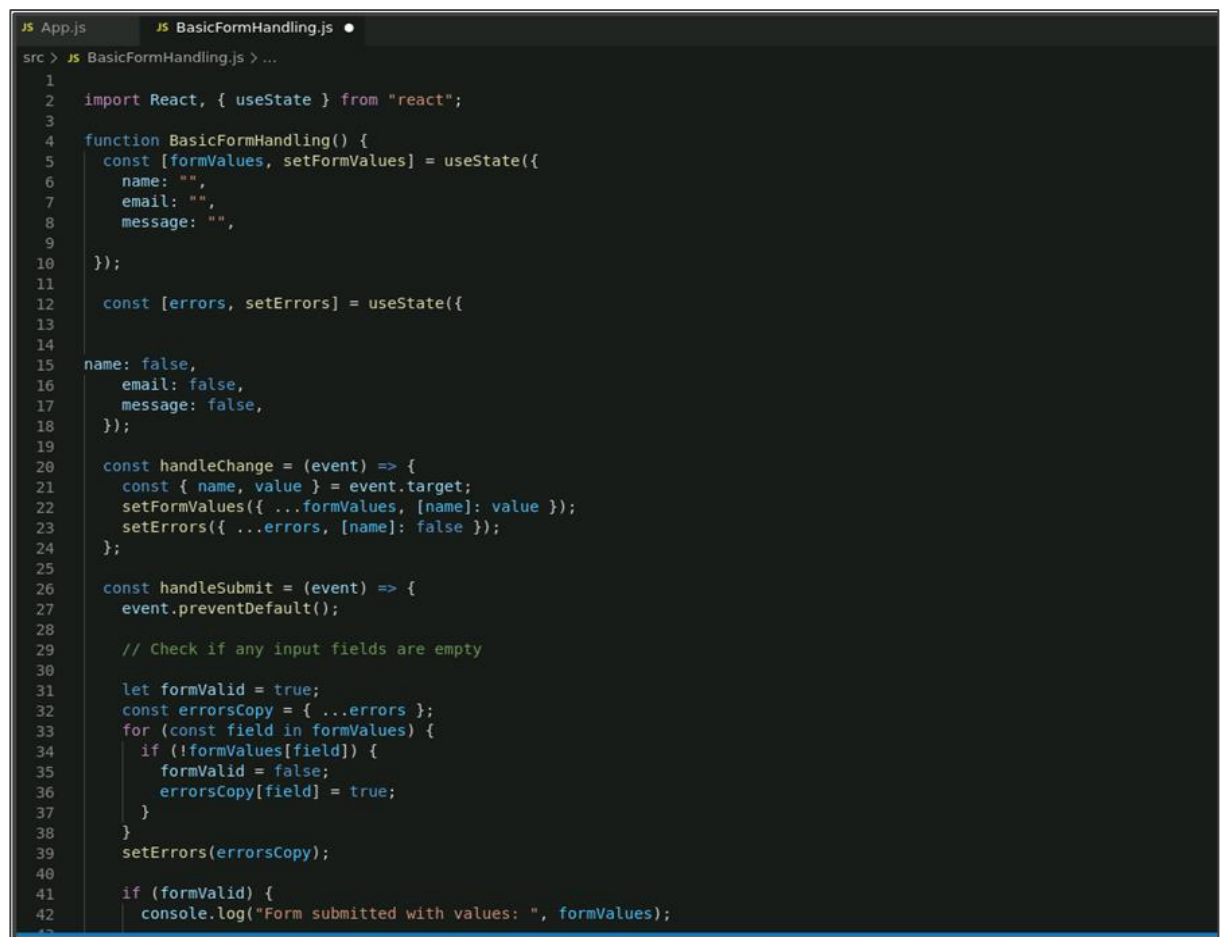
```
if (formValid) {
  console.log("Form submitted with values: ", formValues);

  // Add code to send form data to server or perform the desired action
}
};
return (
  <form onSubmit={handleSubmit}>
    <div>
      <label htmlFor="name">Name:</label>
      <input
        type="text"
        id="name"
        name="name"
        value={formValues.name}
        onChange={handleChange}
      />
      {errors.name && <div className="error">Please enter your name</div>}
    </div>
    <div>
      <label htmlFor="email">Email:</label>
      <input
        type="email"
        id="email"
        name="email"
        value={formValues.email}
        onChange={handleChange}
      />
      {errors.email && <div className="error">Please enter a valid email
address</div>}
    </div>
    <div>
      <label htmlFor="message">Message:</label>
      <textarea
        id="message"
        name="message"
        value={formValues.message}
      />
    </div>
  </form>
);
```

```

        onChange={handleChange}
      />
      {errors.message && <div className="error">Please enter a message</div>}
    </div>
    <button type="submit">Submit</button>
  </form>
);
}
export default BasicFormHandling;

```



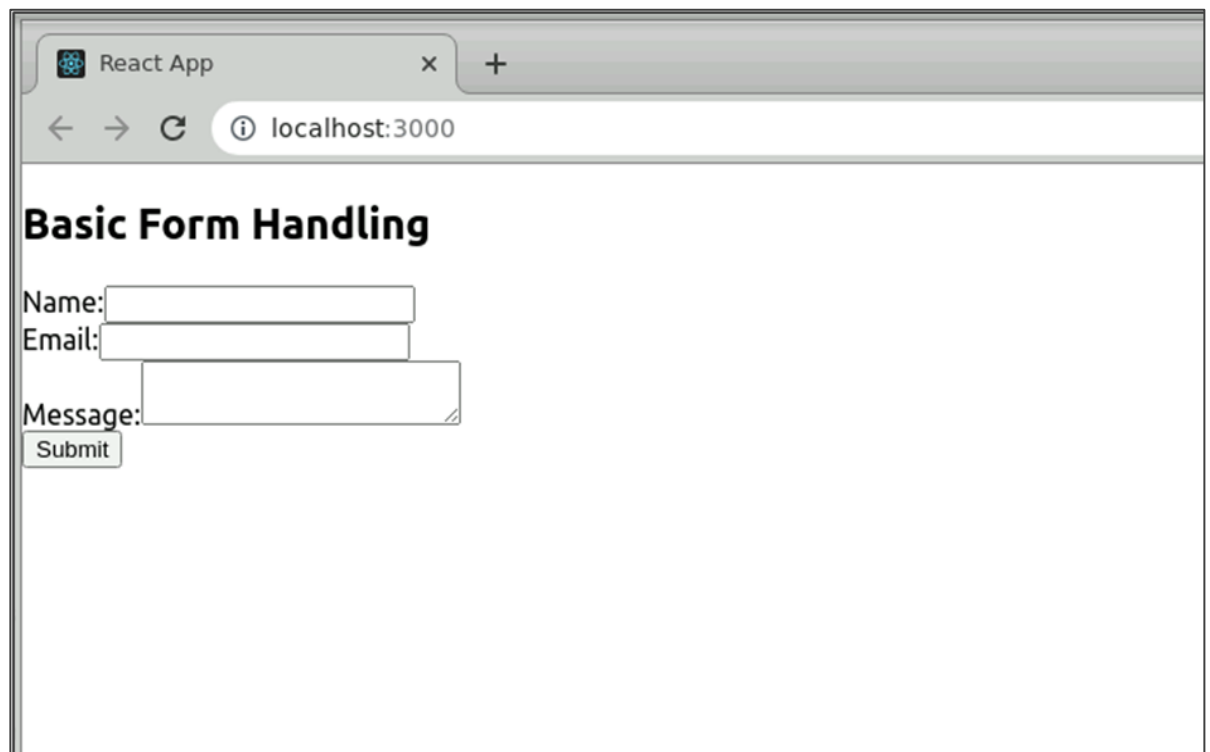
```

JS App.js JS BasicFormHandling.js
src > JS BasicFormHandling.js > ...
1
2 import React, { useState } from "react";
3
4 function BasicFormHandling() {
5   const [formValues, setFormValues] = useState({
6     name: "",
7     email: "",
8     message: "",
9   });
10
11   const [errors, setErrors] = useState({
12     name: false,
13     email: false,
14     message: false,
15   });
16
17   const handleChange = (event) => {
18     const { name, value } = event.target;
19     setFormValues({ ...formValues, [name]: value });
20     setErrors({ ...errors, [name]: false });
21   };
22
23   const handleSubmit = (event) => {
24     event.preventDefault();
25
26     // Check if any input fields are empty
27
28     let formValid = true;
29     const errorsCopy = { ...errors };
30     for (const field in formValues) {
31       if (!formValues[field]) {
32         formValid = false;
33         errorsCopy[field] = true;
34       }
35     }
36     setErrors(errorsCopy);
37
38     if (formValid) {
39       console.log("Form submitted with values: ", formValues);
40     }
41   };
42 }

```

Step 4: Run the app and verify the functionality

- 4.1 Go to the terminal and execute the **npm start** command within the project directory **basic-form-app** to run the app
- 4.2 Once the server starts successfully, open **http://localhost:3000** in your browser to view the app



The screenshot shows a web browser window with a single tab titled "React App". The address bar displays "localhost:3000". The main content area of the browser shows a web page with the heading "Basic Form Handling". Below the heading, there is a form with three input fields: "Name:", "Email:", and "Message:". Each field is followed by a text input box. Below the "Message:" field, there is a "Submit" button.

In the browser, you will see the heading **Basic Form Handling** and a form with input fields for **Name**, **Email**, and **Message**, along with a **Submit** button. Enter the values in the input fields and submit the form. If any required field is left empty or an invalid email address is entered, appropriate error messages will be displayed. If all the required fields are filled correctly, the form is submitted, and the form values will be logged into the console.

With this, you have successfully created a form in React to collect values from the user and log them into the console.