**Ex. No : 4 A**

**Date : 13.03.2025**

**Develop client-side application using React API**

**A. Form Design and Validation**

**AIM:**

To Develop application to demonstrate form design and validation of forms using React Library.

**PROCEDURE:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **React** | **DESCRIPTION** |
| **1** | **useState()** | **Manages state inside a functional component (e.g., storing form data).** |
| **2** | **useEffect()** | **Runs side effects (e.g., fetching data, form validation on load).** |
| **3** | **onChange()** | **Handles user input changes in form fields.** |
| **4** | **onSubmit()** | **Triggers form submission and validation logic.** |
| **5** | **setState()** | **Updates the state of the component dynamically.** |
| **6.** | **preventDefault()** | **Prevents the default form submission behavior.** |
| **7** | **map()** | **Loops through an array (e.g., displaying validation errors).** |

**SOURCE CODE :**

App.js

import React, { useState } from "react";

import "./App.css";

export default function Form() {

const [formData, setFormData] = useState({

name: "",

email: "",

password: "",

});

const [errors, setErrors] = useState({});

const validate = () => {

let newErrors = {};

if (!formData.name.trim()) newErrors.name = "Name is required.";

if (!formData.email.includes("@")) newErrors.email = "Enter a valid email.";

if (formData.password.length < 6)

newErrors.password = "Password must be at least 6 characters.";

return newErrors;

};

const handleSubmit = (e) => {

e.preventDefault();

const newErrors = validate();

setErrors(newErrors);

if (Object.keys(newErrors).length === 0) alert("Form submitted!");

};

const handleChange = (e) => {

setFormData({ ...formData, [e.target.name]: e.target.value });

};

return (

<div className="container">

<h2>Form Validation</h2>

<form onSubmit={handleSubmit}>

<div className="form-group">

<label>Name:</label>

<input type="text" name="name" onChange={handleChange} />

{errors.name && <small className="error">{errors.name}</small>}

</div>

<div className="form-group">

<label>Email:</label>

<input type="email" name="email" onChange={handleChange} />

{errors.email && <small className="error">{errors.email}</small>}

</div>

<div className="form-group">

<label>Password:</label>

<input type="password" name="password" onChange={handleChange} />

{errors.password && <small className="error">{errors.password}</small>}

</div>

<button type="submit">Submit</button>

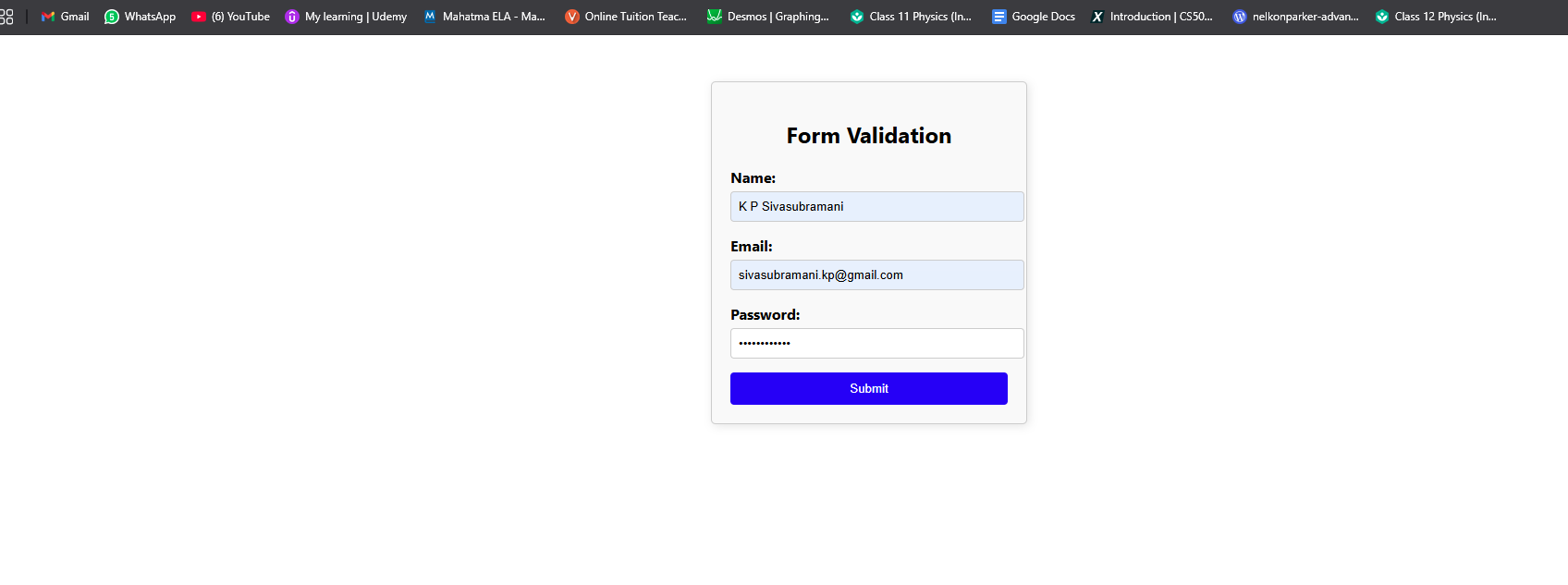
</form>

</div>

);

}

**OUTPUT 1 :**



**RESULT :**

The program has been successfully executed and the output is verified.