React-16

### ****Project Title****: MailRegisterApp – React Application with Form Validation

### ****Objectives****

* Explain **React Forms validation** mechanisms.
* Identify key differences between **React Forms** and **HTML Forms**.
* Define and utilize **controlled components**.
* Recognize various **React Form input controls** such as textboxes, buttons, and textareas.
* Understand how to **handle form events** in React.
* Explain the **submission process** of forms in a React environment.

### ****Learning Outcomes****

By completing this lab, the learner will be able to:

* Implement real-time and submission-based **form validations** in React.
* Use **controlled form inputs** effectively.
* Handle form events with onChange and onSubmit.
* Provide proper **user feedback** for invalid form entries.

### ****Prerequisites****

To execute this lab, the following are required:

* Node.js
* NPM
* Visual Studio Code

### ****Estimated Time****

* Approximately 60 minutes.

### ****Implementation Instructions****

1. **Project Setup**
   * Create a new React application named mailregisterapp using:
   * npx create-react-app mailregisterapp
2. **Component Creation**
   * Inside the src folder, create a file named register.js.
   * Define a React functional or class component named Register.
3. **Form Layout**
   * Design a form with the following fields:
     + **Name** (textbox)
     + **Email** (textbox)
     + **Password** (textbox)
     + **Submit** button
4. **Validation Rules**
   * Implement validation logic as follows:
     + **Name** must be at least **5 characters**.
     + **Email** must include both **“@”** and **“.”** symbols.
     + **Password** must be at least **8 characters**.
5. **Event Handling**
   * Use onChange to update state and show validation feedback in real time if needed.
   * Use onSubmit (e.g., handleSubmit) to prevent default form submission and validate inputs before further processing.
6. **Expected Output**
   * If validation fails, display appropriate **error messages**.
   * If validation passes, show a **success message** or proceed with form submission logic (e.g., displaying an alert, console output, etc.).