Mini Project Report - 13

Master of Computer Application – General
Semester – III

Sub: Web Technologies

Topic: UseState Node.js
By
Name: SANDRA B
Reg no.: 24110222500001

Faculty Name: VEERA RAGHAV K

Faculty Signature: _______________

Department of Computer Application
Alliance University
Chandapura - Anekal Main Road, Anekal
Bengaluru - 562 106

August 2025

Mini Project Report – React Login/Registration Form

Introduction

A login or registration form is an essential component of most web applications. It helps in collecting user details and authenticating access. This project demonstrates the creation of a simple login/registration page using **React.js** with fields such as **Registration Number, Name, Course, Email, and Password**. The project also implements features like form submission, clearing fields, displaying entered details, and showing an alert message.

Objective

The objectives of this project are:

- To design a simple and user-friendly login/registration form using React.js.
- To demonstrate the use of **React useState Hook** for dynamic data handling.
- To implement **form submission** and **clear functionality**.
- To display entered details in a structured box with styling.

Tools and Technologies Used

1. **Node.js** – JavaScript runtime environment.
2. **npm (Node Package Manager)** – For managing dependencies.
3. **React.js** – For building the UI.
4. **Visual Studio Code** – Code editor.
5. **Web Browser (Google Chrome/Edge)** – For running and testing the project.

Description of Code

1. **React Component (App.js)**
2. import React, { useState } from "react";
3.
4. export default function App() {
5.   const [regno, setRegno] = useState("");
6.   const [name, setName] = useState("");
7.   const [course, setCourse] = useState("");

```
const [email, setEmail] = useState("");
const [password, setPassword] = useState("");
const [showDetails, setShowDetails] = useState(false);

// Submit function
const handleSubmit = (e) => {
  e.preventDefault();
  setShowDetails(true);
  alert("Form submitted successfully!");
};

// Clear function
const handleClear = () => {
  setRegno("");
  setName("");
  setCourse("");
  setEmail("");
  setPassword("");
  setShowDetails(false);
};

return (
  <div
    style={{
      border: "2px solid black",
      width: "400px",
      margin: "50px auto",
      padding: "20px",
      borderRadius: "10px",
      textAlign: "center",
    }}
  >
    <h1><u>Alliance University</u></h1>
    <form onSubmit={handleSubmit}>
      <div style={{ margin: "10px" }}>
        <label>Reg. No: </label>
        <input type="text" value={regno} onChange={(e) => setRegno(e.target.value)} />
      </div>
      <div style={{ margin: "10px" }}>
        <label>Name: </label>
        <input type="text" value={name} onChange={(e) => setName(e.target.value)} />
      </div>
      <div style={{ margin: "10px" }}>
        <label>Course: </label>
        <input type="text" value={course} onChange={(e) => setCourse(e.target.value)} />
      </div>
      <div style={{ margin: "10px" }}>
        <label>Email: </label>
        <input type="email" value={email} onChange={(e) => setEmail(e.target.value)} />
      </div>
      <div style={{ margin: "10px" }}>
        <label>Password: </label>
        <input type="password" value={password} onChange={(e) => setPassword(e.target.value)} />
      </div>
      <div style={{ marginTop: "15px" }}>
        <button type="submit" style={{ marginRight: "10px" }}>Submit</button>
        <button type="button" onClick={handleClear}>Clear</button>
        </div>
      </form>

      {showDetails && (
        <div
          style={{
            marginTop: "20px",
            padding: "10px",
            border: "1px solid gray",
            borderRadius: "5px",
            backgroundColor: "#f9f9f9",
            textAlign: "left",
          }}
        >
          <h3>Entered Details:</h3>
          <p><b>Reg. No:</b> {regno}</p>
          <p><b>Name:</b> {name}</p>
          <p><b>Course:</b> {course}</p>
          <p><b>Email:</b> {email}</p>
          <p><b>Password:</b> {password}</p>
        </div>
      )}
    </div>
  );
}
```

Working of the Project

1. The user enters **Reg.No, Name, Course, Email, and Password** in the input fields.
2. Clicking **Submit** displays an alert (“Form submitted successfully!”) and shows the entered details inside a box below the form.

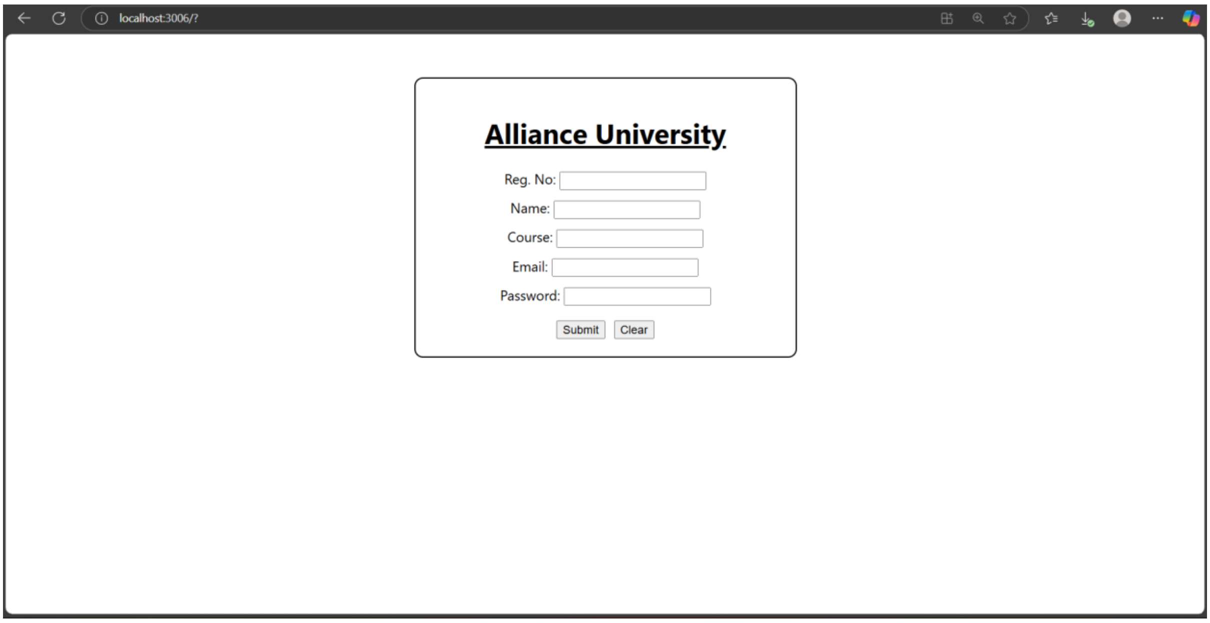
3. Clicking **Clear** resets all input fields and hides the details box.
4. The entire login/registration form is enclosed within a **bordered box** for better visibility.

Features

- Simple and structured registration/login form.
- Uses React's **useState Hook** for dynamic data updates.
- **Submit button** → Displays details and triggers alert.
- **Clear button** → Clears inputs and hides details.
- Clean layout with bordered form container.

Output

- A bordered login form with fields for **Reg.No, Name, Course, Email, Password**.
- After clicking **Submit**, entered details appear in a box below.
- Alert popup confirms successful submission.
- **Clear button** resets the form.



Advantages

- Simple and easy to understand for beginners.
- Demonstrates React **state management** effectively.
- User-friendly interface.
- Can be extended to include validation and backend integration.

Future Scope

- Add **form validation** (check empty fields, valid email format, password strength).
- Connect with a **backend (Node.js, Express, MongoDB)** for storing user details.
- Use **CSS frameworks (Bootstrap/Tailwind)** for better design.
- Add features like **Forgot Password** and **Register/Login toggle**.
- Make the form **responsive for mobile devices**.

Conclusion

This project demonstrates how to create a simple login/registration form in **React.js** using useState for managing input fields. It highlights form submission, clearing inputs, and displaying entered details dynamically. The project lays the foundation for building more advanced authentication systems by integrating with backend services.