



React JS

React Forms

Agenda

01 Forms in HTML

03 Controlled Components

05 Form Validation

02 Forms in ReactJS

04 Uncontrolled Components

06 Displaying Errors

Forms in HTML

Forms in HTML

Forms are an essential part of HTML, as they are used to collect information from user



Forms in HTML

Forms are important because they are the most intuitive way that a user can interact and enter data into our application



Forms in HTML

Forms in HTML have several tags such as form, input, label, button etc.



Forms in HTML

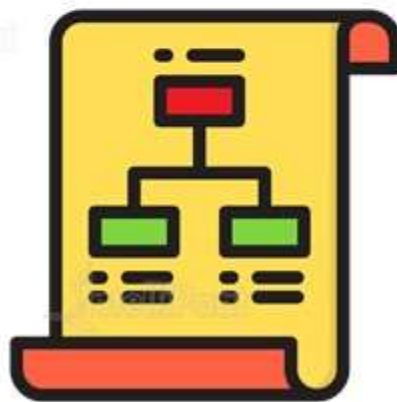
Each component in a form has their own attributes which modify how they look and/or how they work



Creating a form in ReactJS

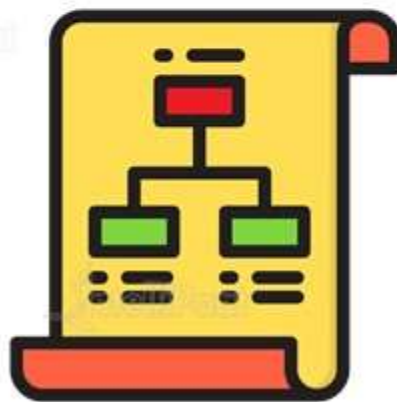
Forms in ReactJS

Just like HTML, ReactJS can also be used to create forms which can be used to capture user input



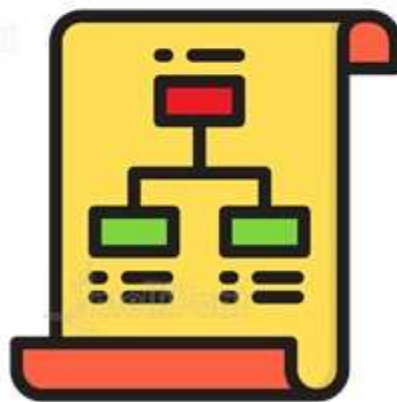
Forms in ReactJS

But with ReactJS we can have forms which can manage their own internal state and keep it in sync with the UI



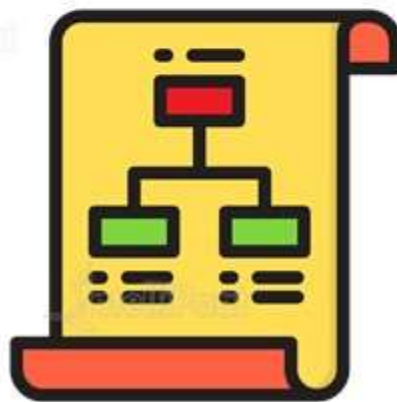
Forms in ReactJS

These forms can handle validation errors as well as asynchronous form submissions as well



Forms in ReactJS

There are two ways to create and manage a Form Components in ReactJS:
Controlled Components and Uncontrolled Components



Controlled Components

Controlled Components

In ReactJS a controlled component is an element that contains data that is handled by the state within the component



Controlled Components

In other words a component that contains data which is managed and controlled by it's internal state is a controlled component



Controlled Components

A Controlled Component is especially useful when building forms as each form element can contain its own internal state which can be managed by ReactJS



Controlled Components

In a Controlled Component the state acts as “single source of truth” as the data flow is uni-directional so it can only be updated one way



Hands On: Controlled Components

Uncontrolled Components

Uncontrolled Components

In ReactJS Uncontrolled Components act more like traditional HTML elements



Uncontrolled Components

In Uncontrolled Components we get access to underlying DOM elements and query their data instead of maintaining state



Uncontrolled Components

To get access to underlying HTML components we make extensive use of refs, which allow us to access DOM elements in ReactJS



Uncontrolled Components

Although Uncontrolled Components are easier to use and understand they should only be used in special cases where we need to manipulate underlying DOM elements



Hands On: Uncontrolled Components

Form Validation

In ReactJS since we can store data as state in our components we can also have validation of the same data in that component



When building a form we can store validation rules and when the form is submitted we can generate error messages



So in order to validate form we need to keep track of data as state variables and generate and display errors using state variables as well



We just need to make sure that the forms are validated properly and if they are then we can submit them asynchronously and deal with the result

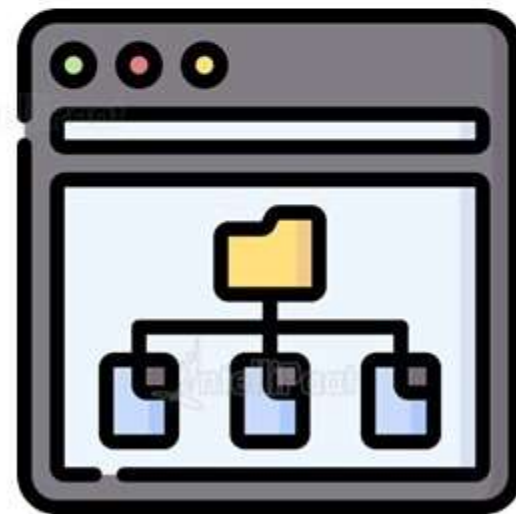


Hands On: Validating Forms

Displaying Errors

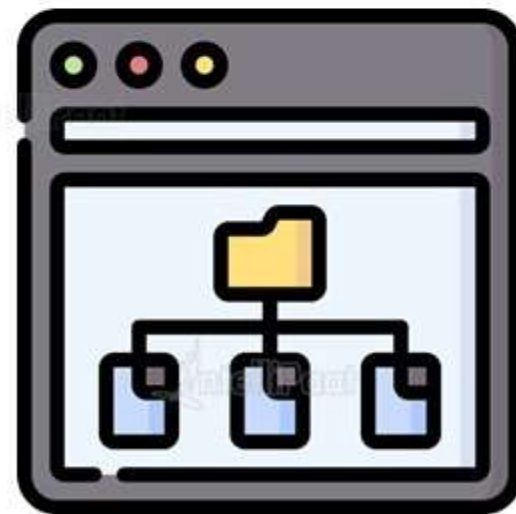
Displaying Errors

Once we have validated our forms we and we have gotten some errors we need to give feedback to the user to fix those errors



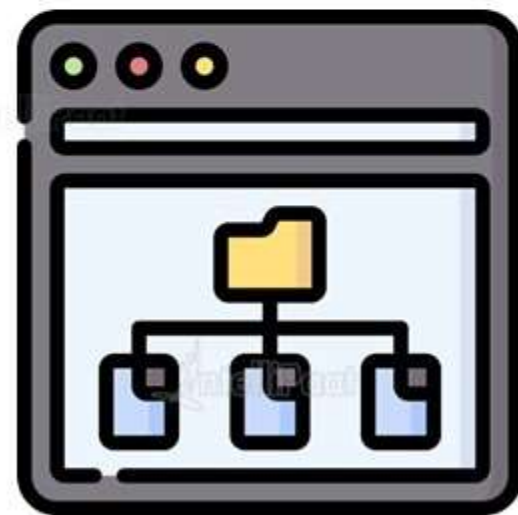
Displaying Errors

For this purpose we need to display errors and indicate that the data entered by the user is invalid and needs to be fixed



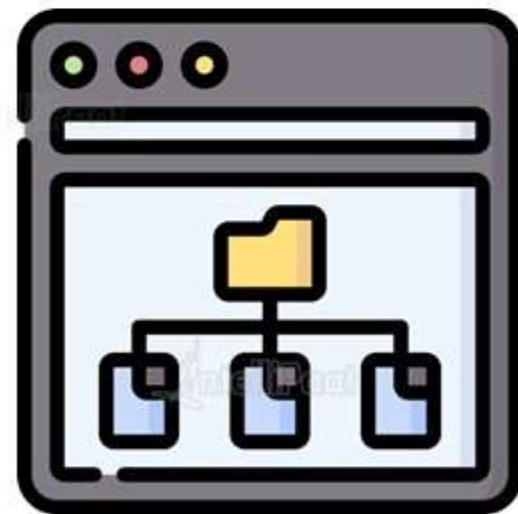
Displaying Errors

And we need to remove the errors once they are fixed and form is submitted to the users



Displaying Errors

There are many libraries available for form validation and error handling but only use them your forms become really complicated



Hands On: Displaying Errors



India: +91-7847955955

US: 1-800-216-8930 (TOLL FREE)



support@intellipaate.com



24/7 Chat with Our Course Advisor