**Emp Id : 46190961**

**Emp Name.: Dharmadhipathi Sathvik Krishna**

**Email :** [**dharmadhipathi.sathvik-krishna@capgemini.com**](mailto:dharmadhipathi.sathvik-krishna@capgemini.com)

**React Forms**

HTML form elements work a bit differently from other DOM elements in React, because form elements naturally keep some internal state.This form has the default HTML form behavior of browsing to a new page when the user submits the form. If you want this behavior in React, it just works. But in most cases, it’s convenient to have a JavaScript function that handles the submission of the form and has access to the data that the user entered into the form. The standard way to achieve this is with a technique called “controlled components”.

## Controlled Components

In HTML, form elements such as <input>, <textarea>, and <select> typically maintain their own state and update it based on user input. In React, mutable state is typically kept in the state property of components, and only updated with [setState()](https://reactjs.org/docs/react-component.html" \l "setstate).

We can combine the two by making the React state be the “single source of truth”. Then the React component that renders a form also controls what happens in that form on subsequent user input. An input form element whose value is controlled by React in this way is called a “controlled component”.

Since handleChange runs on every keystroke to update the React state, the displayed value will update as the user types.With a controlled component, the input’s value is always driven by the React state. While this means you have to type a bit more code, you can now pass the value to other UI elements too, or reset it from other event handlers.

## The textarea Tag

In HTML, a <textarea> element defines its text by its children.

In React, a <textarea> uses a value attribute instead.

## The select Tag

In HTML, <select> creates a drop-down list.React, instead of using this selected attribute, uses a value attribute on the root select tag. This is more convenient in a controlled component because you only need to update it in one place.

## The file input Tag

In HTML, an <input type="file"> lets the user choose one or more files from their device storage to be uploaded to a server or manipulated by JavaScript via the [File API](https://developer.mozilla.org/en-US/docs/Web/API/File/Using_files_from_web_applications).Because its value is read-only, it is an uncontrolled component in React. It is discussed together with other uncontrolled components [later in the documentation](https://reactjs.org/docs/uncontrolled-components.html#the-file-input-tag).

## Handling Multiple Inputs

When you need to handle multiple controlled input elements, you can add a name attribute to each element and let the handler function choose what to do based on the value of event.target.name.

## Controlled Input Null Value

Specifying the value prop on a [controlled component](https://reactjs.org/docs/forms.html#controlled-components) prevents the user from changing the input unless you desire so. If you’ve specified a value but the input is still editable, you may have accidentally set value to undefined or null.