

Forms in React

When the user submits the form in HTML, the browser navigates to a new page. React exhibits the same behavior. However, in the majority of situations, we want to have a JavaScript function that manages the form submission and has access to the information that the user entered into the form. This is typically accomplished via a method known as "controlled components."

We can use the `setState()` function in case of Class component or `useState()` hook in case of Function Component to maintain the states of elements in the form.

Let's see example of each.

Class Form Component

```
class UserRemarks extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = {value: ""};  
  
    this.changeHandlerFunc = this.changeHandlerFunc.bind(this);  
    this.submitHandlerFunc = this.submitHandlerFunc.bind(this);  
  }  
  
  changeHandlerFunc(event) {  
    this.setState({value: event.target.value});  
  }  
  
  submitHandlerFunc(event) {  
    alert('User Remarks: ' + this.state.value);  
    event.preventDefault();  
  }  
}
```

```
render() {  
  return (  
    <form onSubmit={this.submitHandlerFunc}>  
      <label>  
        User Remarks:  
        <textarea value={this.state.value} onChange={this.changeHandlerFunc} />  
      </label>  
      <input type="submit" value="Submit" />  
    </form>  
  );  
}
```

In the above example, we can see that the `setState()` method is called in the `changeHandlerFunc()` method to maintain the current value of the `textarea` element.

Function Form Component

```
function UserRemarksForm(){  
  const [inputVal, setInputVal] = useState("");  
  
  const changeHandler = (e) => {  
    setInputVal(e.target.value);  
  };  
  
  const submitHandler = (e) => {  
    e.preventDefault();  
    alert('User Remarks' + inputVal);  
  };  
  
  return (  
    <form onSubmit={submitHandler}>
```

```
    <textarea onChange={changeHandler}></textarea>
    <input type="submit"></input>
  </form>
);
}
```

The same result can be achieved by a function form component using the `useState()` hook.