

Forms in React

Make Component

Make 2 input field and button

Use state for every input field

Update State on Change

Print State on Button Click

Create Form.js:-

```
import React from 'react'
export default class Forms extends React.Component{

  constructor()
  {
    super()
    this.state={

    }
  }
}
```

```
render()  
{  
  return(  
    <div>  
      <h1>Form Handling</h1>  
      <input type="text" name="user" />  
      <br /><br />  
      <input type="password" name="password" />  
      <br /><br />  
      <button>Submit</button>  
    </div>  
  )  
}
```

2

OPEN EDITORS 2 UNSAVED

● JS App.js src 3, M

JS Forms.js src U

● ≡ Untitled-1

4

BLOG

▸ node_modules

▸ public ●

▸ src ●

App.css

JS App.js 3, M

JS App.test.js

JS Forms.js U

index.css

src ▸ JS App.js ▸ App

```
1 import React from 'react';
2 import logo from './logo.svg';
3 import './App.css';
4 import Forms from './Forms'
5 function App() {
6   return (
7     <div className="App">
8       <Forms />
9     </div>
10   );
11 }
12
13 export default App;
14
```

← → ↻ ⓘ localhost:3000

📁 Apps 📁 Getting Started 📁 Imported Fro... 📁 para 📁 Imported 📁 YouTube Bann... 📁 English and In...

Form Handling

Submit

Create two state for user and password:-

```
constructor()  
{  
  super()  
  this.state={  
    user:null,  
    password:null  
  }  
}
```

```
r() {  
  return (  
    <div>  
      <h1>Form Handling</h1>  
      <input  
        type="text"  
        name="user"  
        onChange={(e)=>{this.setState({user:e.target.value})}}  
      />  
      <br /><br />  
      <input  
        type="password"  
        name="password"  
        onChange={(e)=>{this.setState({password:e.target.value})}}  
      />  
      <br /><br />  
      <button>Submit</button>  
    </div>  
  )  
}
```

Create a function and fetch print the states:-

```
submit()  
{  
  console.warn(this.state)  
}
```

1/2

The {@link console.warn}(
console.error()).

```
console.warn(this.state)
```

```
<br /><br />
```

submit(): void

```
<button onClick={() => this.submit()}>Submit</button>
```

```
</div>
```


Form Handling

Elements Console Sources Network Performance

top Filter

Download the React DevTools for a better development experience

Warning: ./src/App.js
Line 2: 'logo' is defined but never used no-unused-vars

Warning: {user: null, password: null}

Warning: {user: "sam", password: "123@sam"}
password: "123@sam"
user: "sam"
__proto__: Object

>

Validation

```
super();
this.state={
  name:"",
  password:"",
}
}
submit()
{
  alert("submit")
}
render() {
  return (
    <div className="App">
      <h1>Form Validation</h1>
      <input type="text" onChange={(event)=>{this.setState({name:event.target.value})}} />
      <p></p>
      <input type="password" onChange={(event)=>{this.setState({password:event.target.value})}} />
      <p></p>
      <button onClick={()=>this.submit()}>Submit</button>
    </div>
  );
}
```

Declare the nameError,passwordError

```
{
  super();
  this.state={
    name: "",
    password: "",
    nameError: "",
    passwordError: ""
  }
}
valid()
{

}
submit()
{
  if(this.valid())
  {
    alert("form has been submitted")
  }
}
```

<p>{this.state.nameError}</p>

```
render() {  
  return (  
    <div className="App">  
      <h1>Form Validation</h1>  
      <input type="text" onChange={(event)=>{this.setState({name:event.target.value})}} />  
      <p>{this.state.nameError}</p>  
      <input type="password" onChange={(event)=>{this.setState({password:event.target.value})}} />  
      <p></p>  
      <button onClick={()=>this.submit()}>Submit</button>  
    </div >  
  );  
}
```



```
valid()
{
  if(!this.state.name.includes("@") && this.state.password.length<5)
  {
    this.setState(
      {nameError:"Invalid Name",passwordError:"Password lenght should be more than 5 "}
    )
  }
}

submit()
{
  if(this.valid())
  {
    alert("form has been submited")
  }
}
```

```
valid() {  
  if (!this.state.name.includes("@") && this.state.password.length < 5) {  
    this.setState(  
      { nameError: "Invalid Name", passwordError: "Password length should be more than 5" }  
    )  
  }  
}  
  
submit() {  
  if (this.valid()) {  
    alert("form has been submitted")  
  }  
}  
  
render() {  
  return (  
    <div className="App">  
      <h1>Form Validation</h1>  
      <input type="text" onChange={(event) => { this.setState({ name: event.target.value }) }} />  
      <p style={{ color: "red", fontSize: "14px" }} >{this.state.nameError}</p>  
      <input type="password" onChange={(event) => { this.setState({ password: event.target.value }) }} />  
      <p style={{ color: "red", fontSize: "14px" }} >{this.state.passwordError}</p>  
      <button onClick={() => this.submit()}>Submit</button>  
    </div >  
  );  
}
```

```
valid() {  
  if (!this.state.name.includes("@") && this.state.password.length < 5) {  
    this.setState(  
      { nameError: "Invalid Name", passwordError: "Password lenght should be more than 5" }  
    )  
  }  
  else if (!this.state.name.includes("@")) {  
    this.setState(  
      { nameError: "Invalid Name" }  
    )  
  }  
  else if (this.state.password.length < 5) {  
    this.setState(  
      { passwordError: "Password lenght should be more than 5" }  
    )  
  }  
  else{  
    return true  
  }  
}
```

Form Validation

Invalid Name

Password length should be more than 5


```
submit() {  
  this.setState(  
    { nameError: "", passwordError: "" }  
  )  
  if (this.valid()) {  
    alert("form has been submitted")  
  }  
}  
  
render() {
```