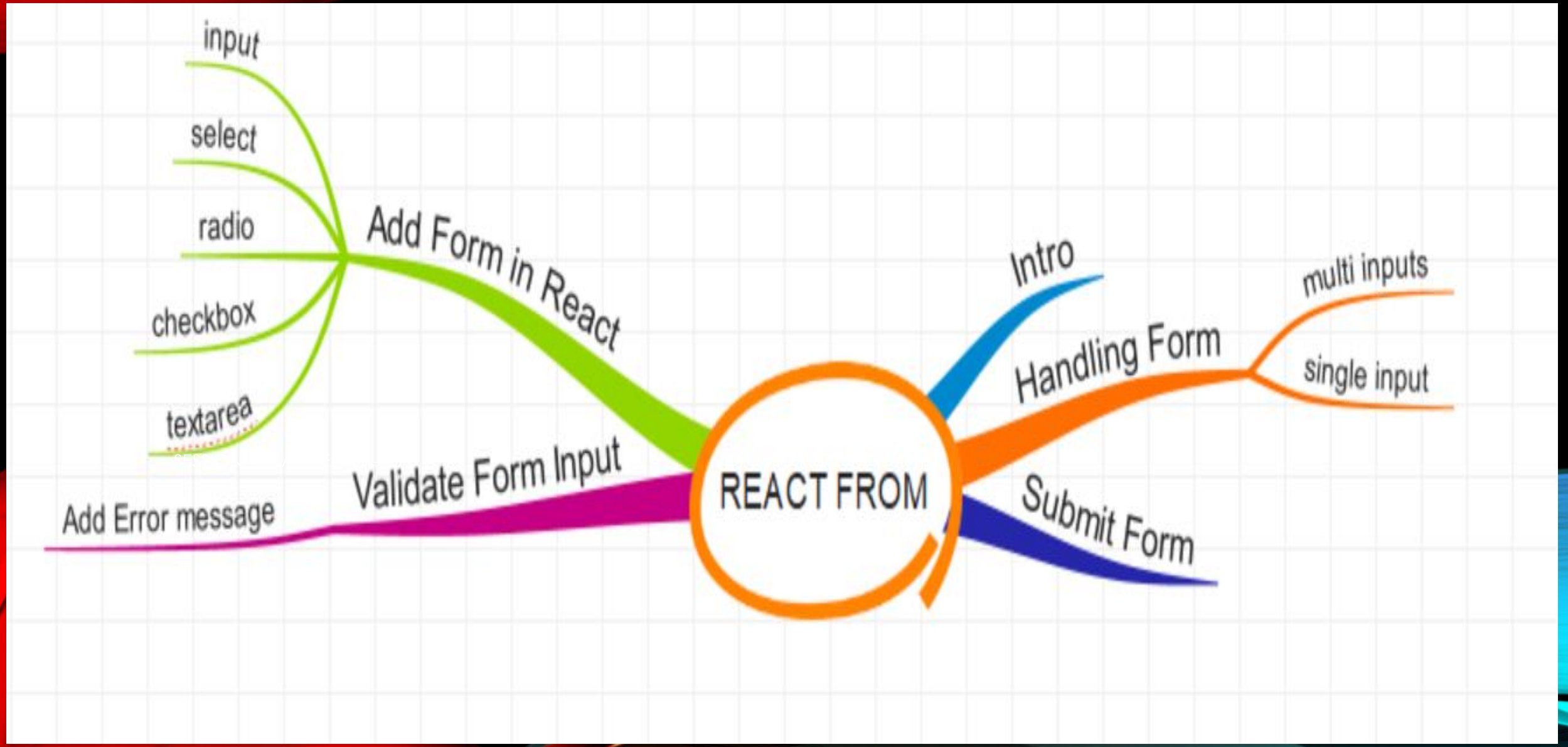
Session 4

REACT FORM

REACT FORM


input
select
radio
checkbox
textarea
Add Form in React
Add Error message
Validate Form Input
REACT FROM
Intro
Handling Form
Submit Form
multi inputs
single input

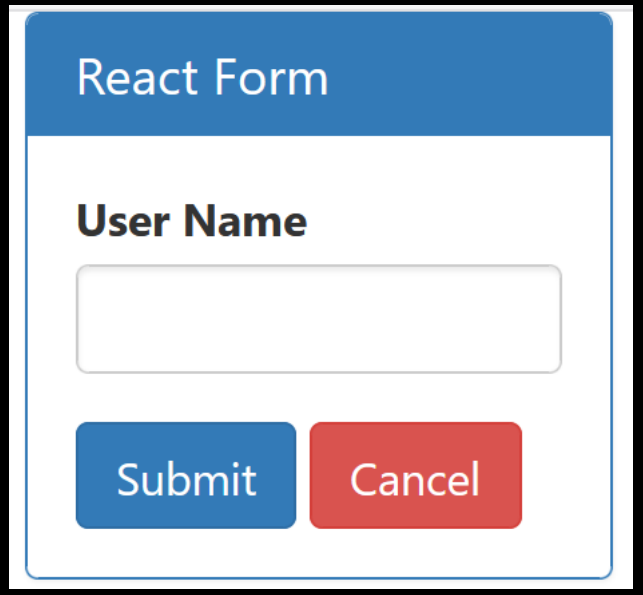
INTRO

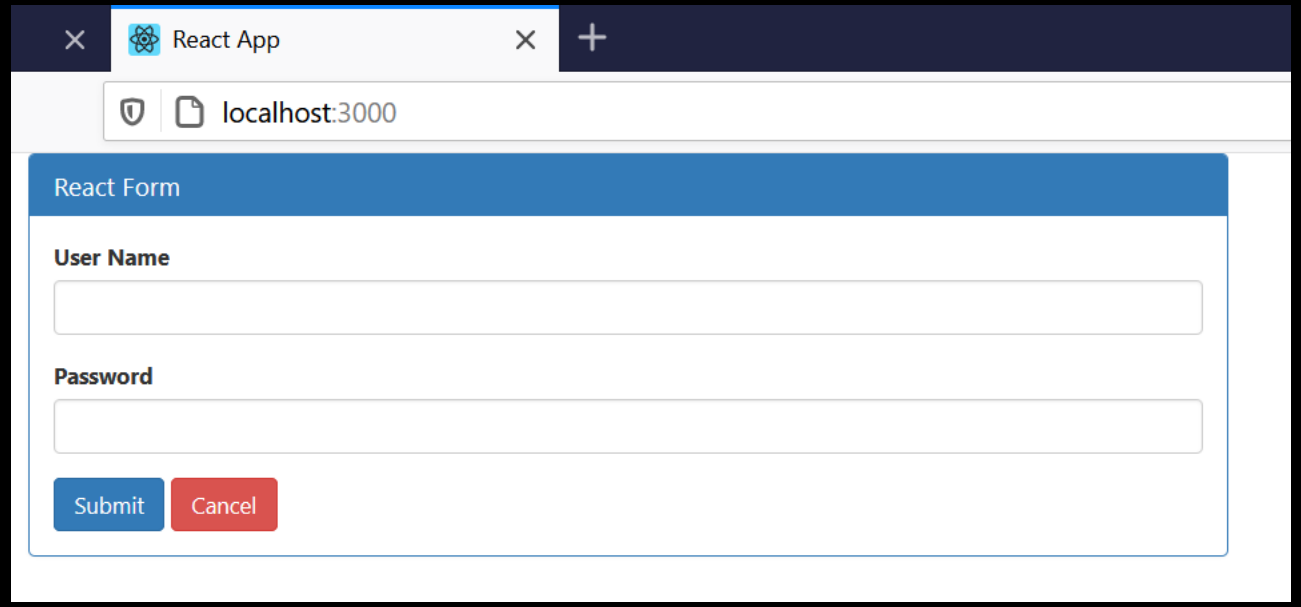
- Just like in HTML, React uses forms to allow users to interact with the web page.
- Forms are integral to any modern application. They serve as a basic medium for users to interact with your app.
- In this session, we will learn how to use forms in React.
- Developers rely on forms for everything: securely logging in the user, searching and filtering the product list, booking a product, and building a cart, etc.

INTRO

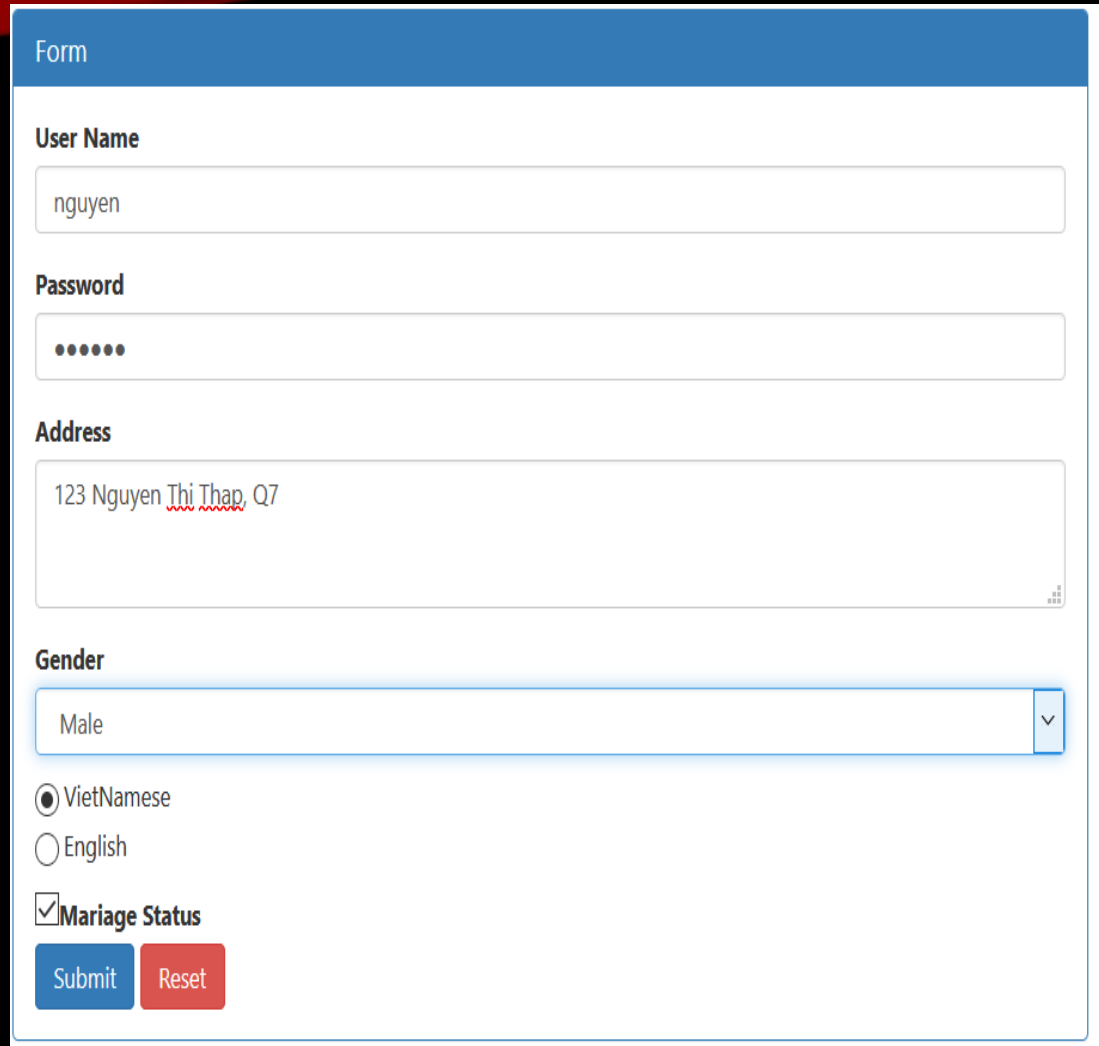
- Forms are very useful in any web application. Unlike with Angular and AngularJS, which give you form validation out of the box, you have to handle forms yourself in React. This brings about many complications, like how to get form values, how to manage form state, and how to validate a form on the fly and show validation messages.

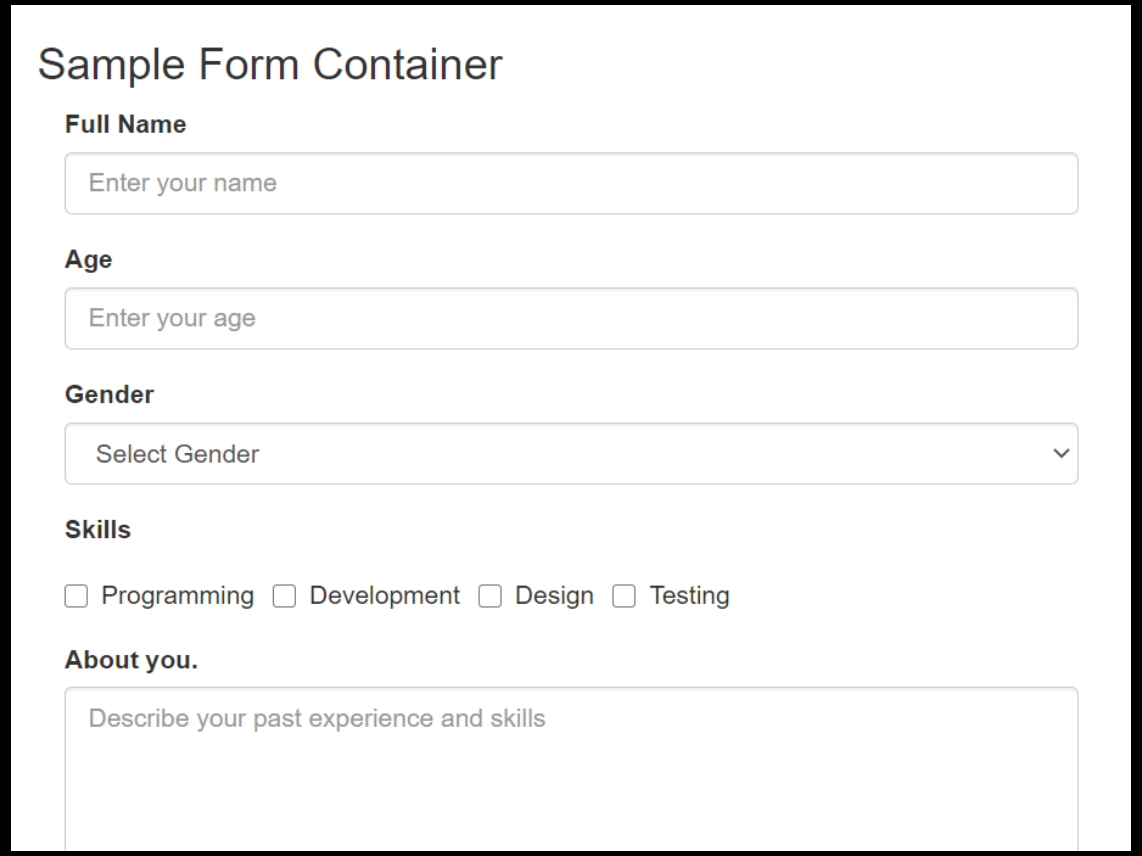
SOME REACT FORM





REACT FORM


Form
User Name
nguyen
Password
Address
123 Nguyen Thi Thap, Q7
Gender
Male
VietNamese
English
Mariage Status
Submit
Reset


Sample Form Container
Full Name
Enter your name
Age
Enter your age
Gender
Select Gender
Skills
Programming
Development
Design
Testing
About you.
Describe your past experience and skills

REACT FORM

- There are two types of form input in React. We have the uncontrolled input and the controlled input. The uncontrolled input are like traditional HTML form inputs, in that they remember what you typed. We will use ref to get the form values.

REACT FORM

- There are two main ways of handling forms in React, which differ on a fundamental level: **how data is managed**.
- if the data is handled by the DOM, we call them uncontrolled components
- if the data is handled by the components we call them controlled components

REACT FORM - EXAMPLE

```
class Form extends React.Component {
  constructor(props) {
    super(props)
    this.state = { username: '' }
  }

  handleChange(event) {}

  render() {
    return (
      <form>
        Username:
        <input
          type="text"
          value={this.state.username}
          onChange={this.handleChange}
        />
      </form>
    )
  }
}
```

1

2

3

REACT FORM - EXAMPLE

- With **class components,** in order to **set the new state**, **we must bind this to the handleChange method,** **otherwise this is not accessible from within that method:**

```
class Form extends React.Component {
  constructor(props) {
    super(props)
    this.state = { username: '' }
    this.handleChange = this.handleChange.bind(this)
  }

  handleChange(event) {
    this.setState({ username: event.target.value })
  }

  render() {
    return (
      <form>
        <input
          type="text"
          value={this.state.username}
          onChange={this.handleChange}
        />
      </form>
    )
  }
}
```

REACT FORM - EXAMPLE

```
class Form extends React.Component {
  constructor(props) {
    super(props)
    this.state = { username: '' }
    this.handleChange = this.handleChange.bind(this)
    this.handleSubmit = this.handleSubmit.bind(this)
  }

  handleChange(event) {
    this.setState({ username: event.target.value })
  }

  handleSubmit(event) {
    alert(this.state.username)
    event.preventDefault()
  }
```

1 2 3 4

```
  render() {
    return (
      <form onSubmit={this.handleSubmit}>
        <input
          type="text"
          value={this.state.username}
          onChange={this.handleChange}
        />
        <input type="submit" value="Submit" />
      </form>
    )
  }
}
```

5

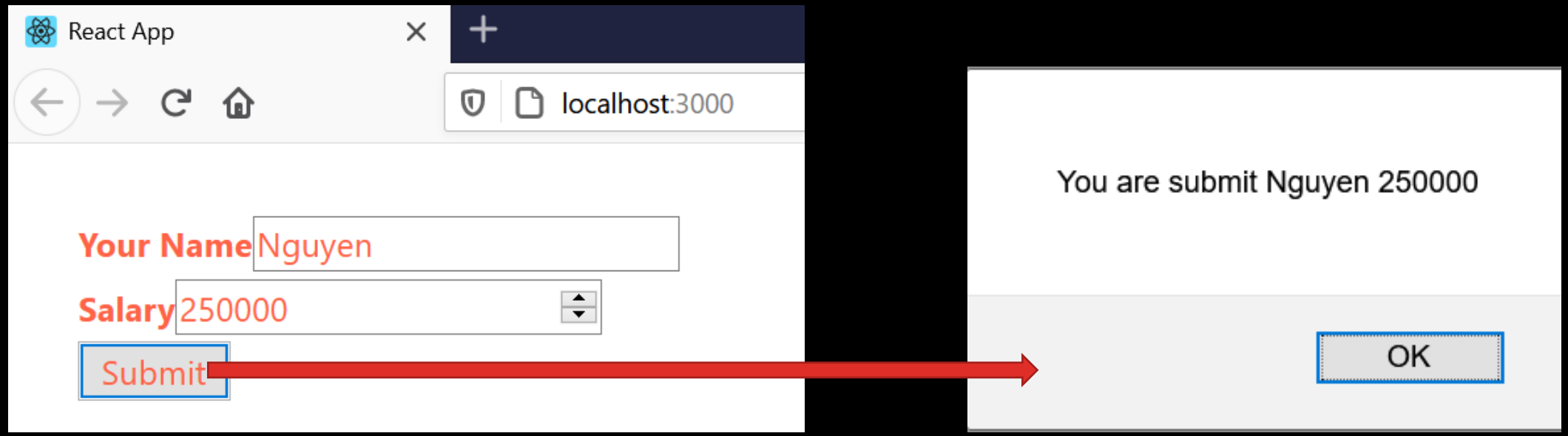
- Similarly, we use the onSubmit attribute on the form to call the handleSubmit method when the form is submitted:

REACT FORM - HOOKS

```
const Form = props => {
  const [username, setUsername] = useState()

  const handleChangeUsername = e => {
    setUsername(e.target.value)
  }

  const handleSubmit = event => {
    alert(username)
    event.preventDefault()
  }

  render() {
    return (
      <form onSubmit={handleSubmit}>
        Username:
        <input
          type="text"
          value={username}
          onChange={handleChangeUsername}
        />
      </form>
    )
  }
}
```

USING HOOKS IT'S ALL MUCH SIMPLER:

ACTIVITY


React App
localhost:3000
Your Name Nguyen
Salary 250000
Submit
You are submit Nguyen 250000
OK

SOLUTION

index.js > ...

You, a few seconds ago | 1 author (You)

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
```

You, a few seconds ago | 1 author (You)

```
class FormComponent extends React.Component {
  constructor(props) {
    super(props);
    //  this.state = {textData:""};
    this.state = {
      username: '',
      salary: 0
    }
  }
  handleChange = (event) => {
    this.setState(
      {
        [event.target.name]: [event.target.value]
      });
  }
  handleSubmit = (event) => {
    alert("You are submit " + this.state.username +" " + this.state.salary);
    event.preventDefault();
  }
```

index.html | index.js

index.js > FormComponent > render

```
  render() {
    return (
      <form className="fr" onSubmit={this.handleSubmit} >
        <label>Your Name</label>
        <input type="text" name="username" value={this.state.txtData} onChange={this.handleChange} />
        <br />
        <label>Salary</label>
        <input type="number" name="salary"
          value={this.state.salary}
          onChange={this.handleChange}
        />
        <br />
        <input type="submit"
          value="Submit" />
      </form>      You, 30 minutes ago • Uncommitted changes
    );
  }

}
ReactDOM.render(
  <FormComponent/>,
  document.getElementById('root')
);
```

SOLUTION

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
You, a few seconds ago | 1 author (You)
class FormComponent extends React.Component {
  constructor(props) {
    super(props);
    //  this.state = {textData:""};
    this.state = {
      username: '',
      salary: 0
    }
  }
  handleChange = (event) => {
    this.setState(
      {
        [event.target.name]: [event.target.value]
      });
  }
  handleSubmit = (event) => {
    alert("You are submit " + this.state.username +" " + this.state.salary);
    event.preventDefault();
  }
```

1

2

3

4

SOLUTION

```
render() {
  return (
    <form className="fr" onSubmit={this.handleSubmit} >
      <label>Your Name</label>
      <input type="text" name="username" value={this.state.txtData} onChange={this.handleChange} />
      <br />
      <label>Salary</label>
      <input type="number" name="salary"
        value={this.state.salary}
        onChange={this.handleChange}
      />
      <br />
      <input type="submit"
        value="Submit" />
    </form>
  );
  }

}
ReactDOM.render(
  <FormComponent/>,
  document.getElementById('root')
);
```

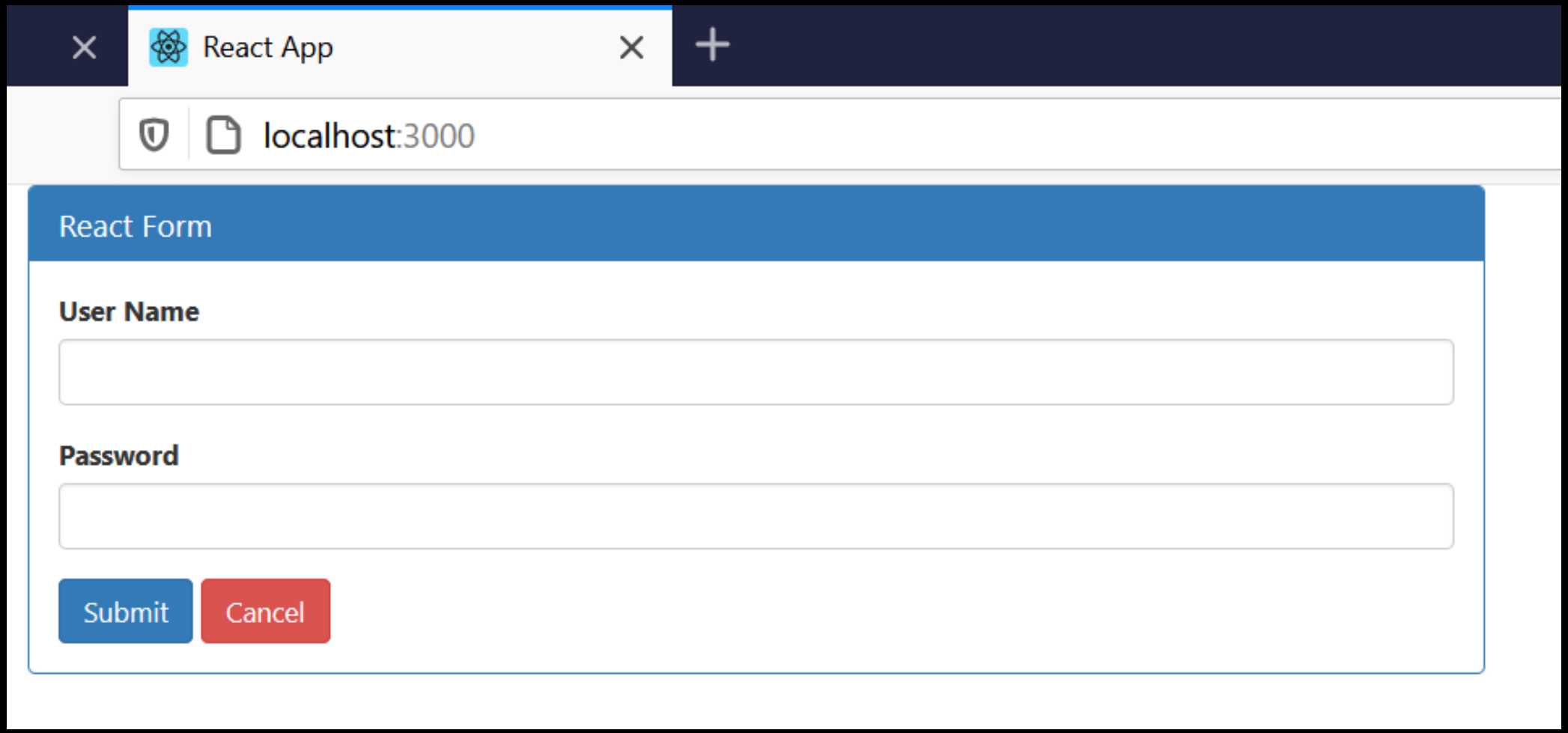
1

2

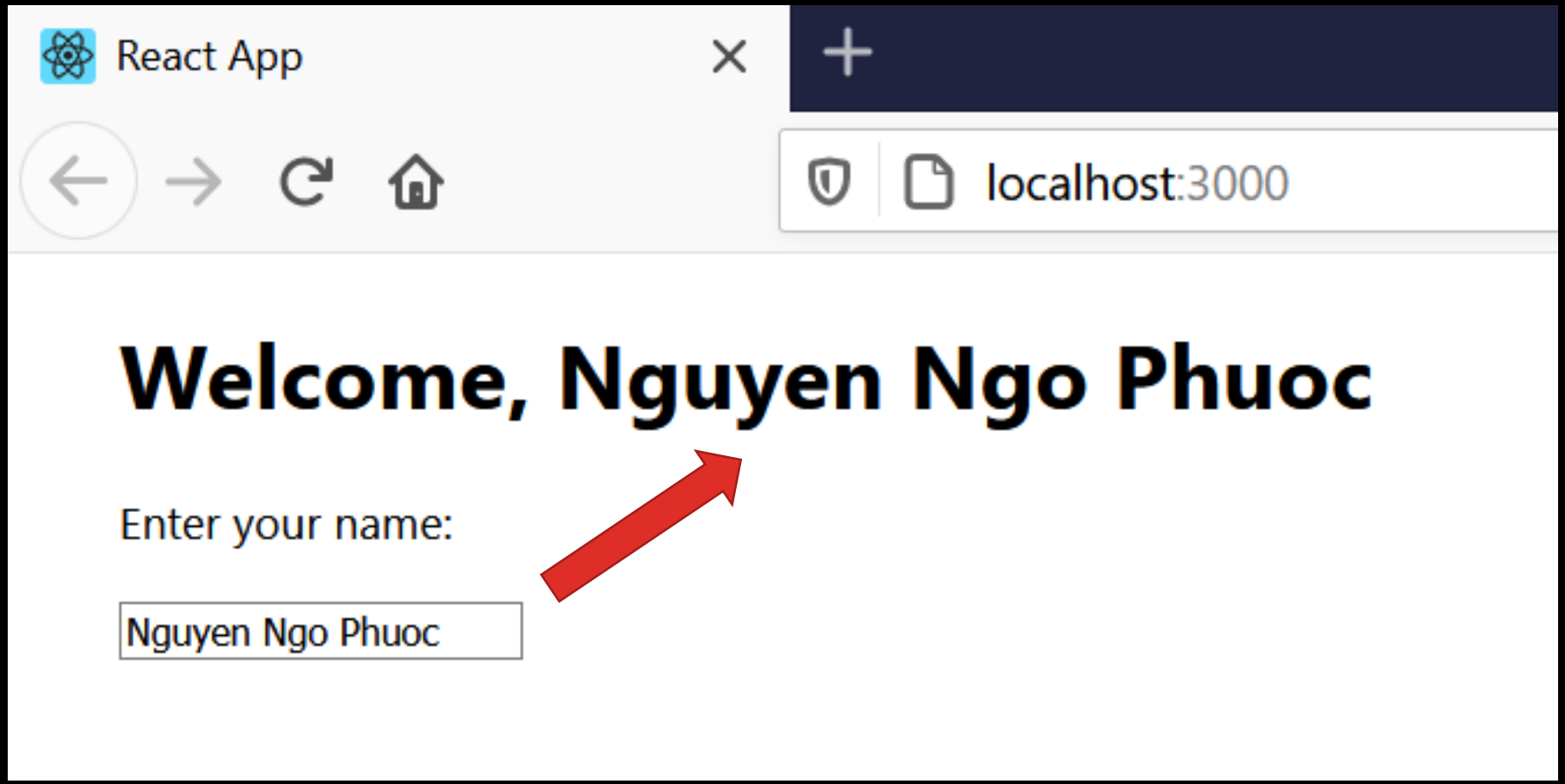
3

4

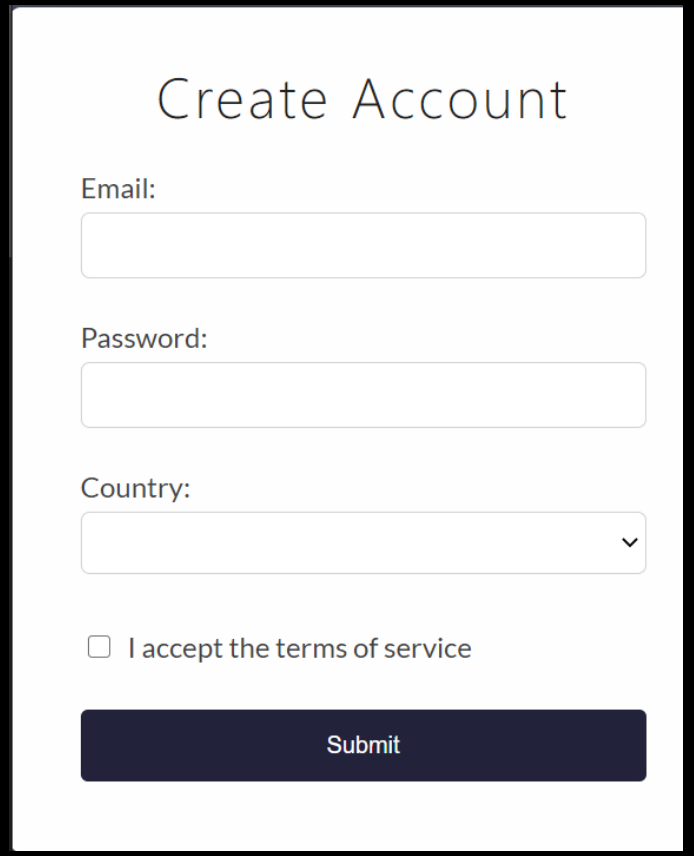
DEMO

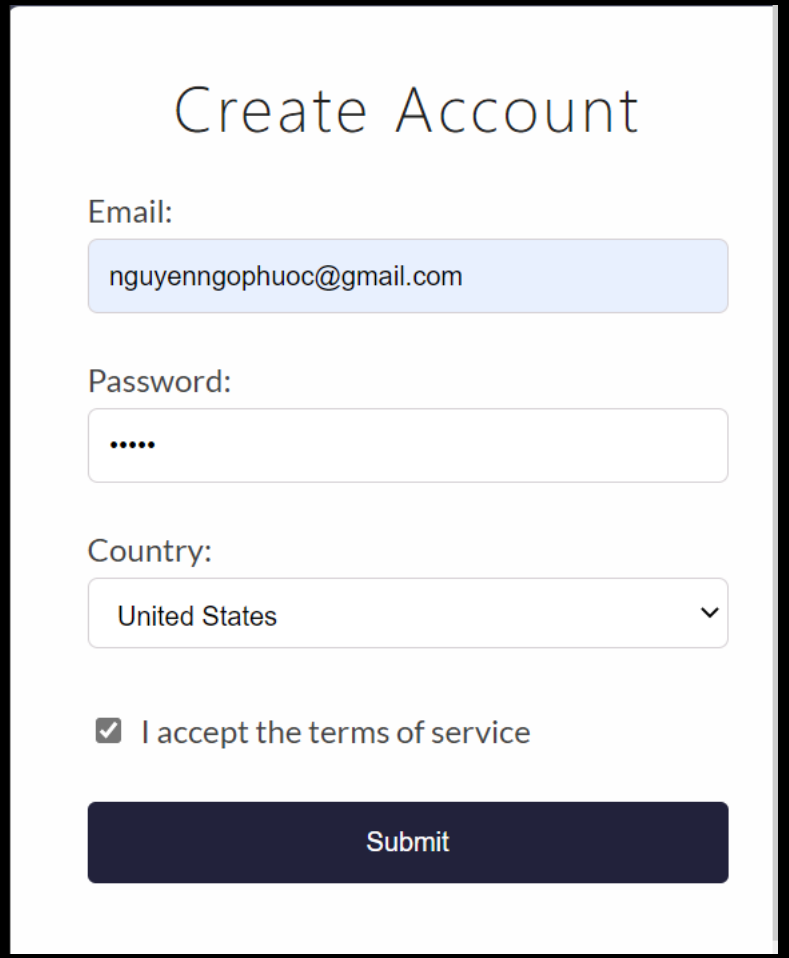

React App
localhost:3000
React Form
User Name
Password
Submit
Cancel

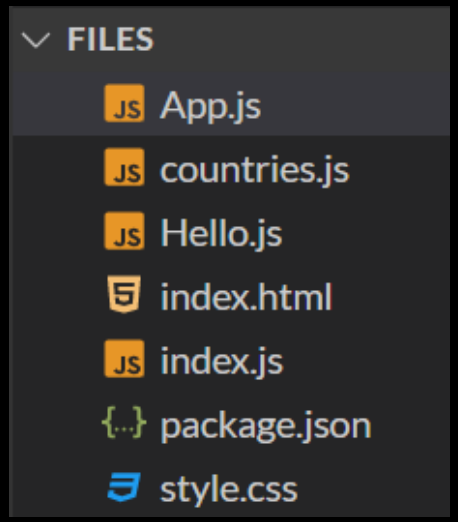
DEMO


React App
localhost:3000
Welcome, Nguyen Ngo Phuoc
Enter your name:
Nguyen Ngo Phuoc

HANDLING MULTIPLE FORM INPUTS







HANDLING MULTIPLE FORM INPUTS

App.js

```
import React, { Component } from "react";
import countries from "./countries";

export default function App() {
  const [email, setEmail] = React.useState("");
  const [password, setPassword] = React.useState("");
  const [country, setCountry] = React.useState("");
  const [acceptedTerms, setAcceptedTerms] = React.useState(false);

  const handleSubmit = (event) => {
    console.log(`
      Email: ${email}
      Password: ${password}
      Country: ${country}
      Accepted Terms: ${acceptedTerms}
    `);

    event.preventDefault();
  }

```

```
  return (
    <form onSubmit={handleSubmit}>
      <h1>Create Account</h1>

      <label>
        Email:
        <input
          name="email"
          type="email"
          value={email}
          onChange={e => setEmail(e.target.value)}
          required />
      </label>

      <label>
        Password:
        <input
          name="password"
          type="password"
          value={password}
          onChange={e => setPassword(e.target.value)}
          required />
      </label>
```

HANDLING MULTIPLE FORM INPUTS

```
44
45          <label>
46            Country:
47            <select
48              name="country"
49              value={country}
50              onChange={e => setCountry(e.target.value)}
51              required>
52              <option key=""></option>
53              {countries.map(country => (
54                <option key={country}>{country}</option>
55              ))}
56            </select>
57          </label>
58
59          <label>
60            <input
61              name="acceptedTerms"
62              type="checkbox"
63              onChange={e => setAcceptedTerms(e.target.value)}
64              required />
65            I accept the terms of service
66          </label>
67
68          <button>Submit</button>
69        </form>
70      );
71    }
```

1

2

3

countries.js

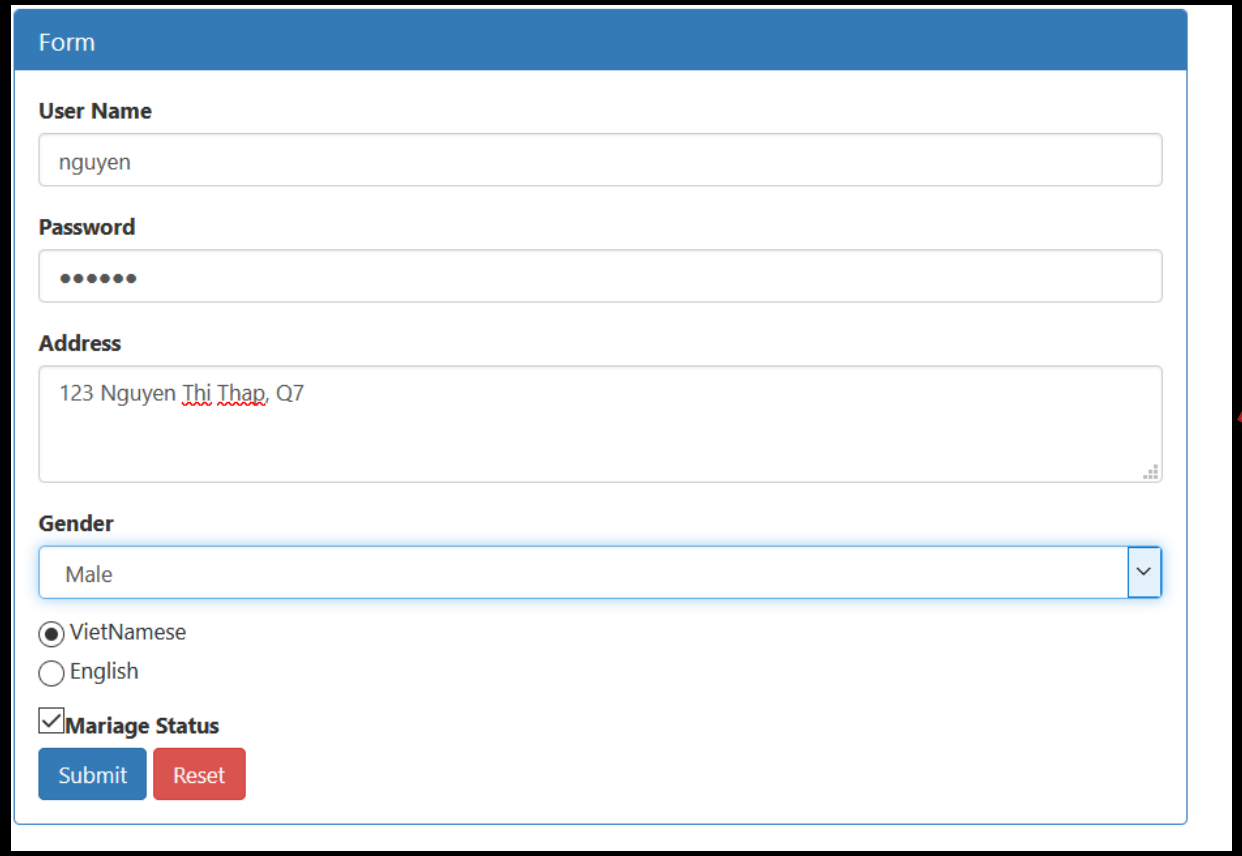
```
1   export default [
2     'Albania',
3     'Andorra',
4     'Armenia',
5     'Austria',
6     'Azerbaijan',
7     'Belarus',
8     'Belgium',
9     'Bosnia & Herzegovina',
10    'Bulgaria',
11    'Croatia',
12    'Cyprus',
13    'Czech Republic',
14    'Denmark',
15    'Estonia',
16    'Finland',
17    'France',
18    'Georgia',
19    'Germany',
20    'Greece',
```

HANDLING MULTIPLE FORM INPUTS

index.js

```
import React from "react";
import ReactDOM from "react-dom";
import App from "./App";

ReactDOM.render(<App />, document.getElementById("root"));
```

DEMO


Form
User Name
nguyen
Password
••••••
Address
123 Nguyen Thi Thap, Q7
Gender
Male
VietNamese
English
Mariage Status
Submit
Reset


Object
chkMariage: true
rdLang: "vi"
sGender: "1"
txtAddress: "123 Nguyen Thi Thap"
txtPassword: "123"
txtUsername: "Nguyen"
__proto__: Object

THE END