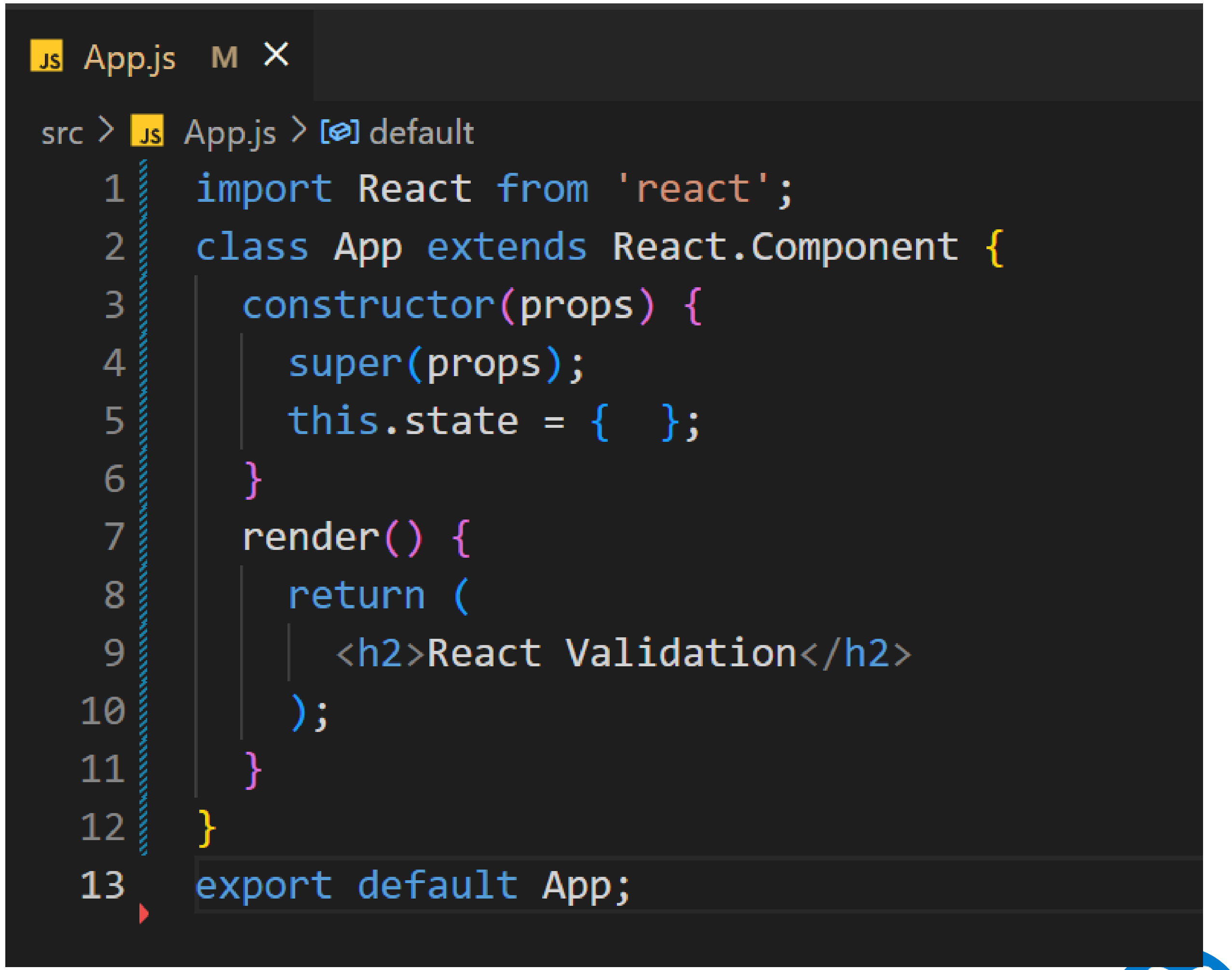
React JS

#React Notes

Validation

Create Component

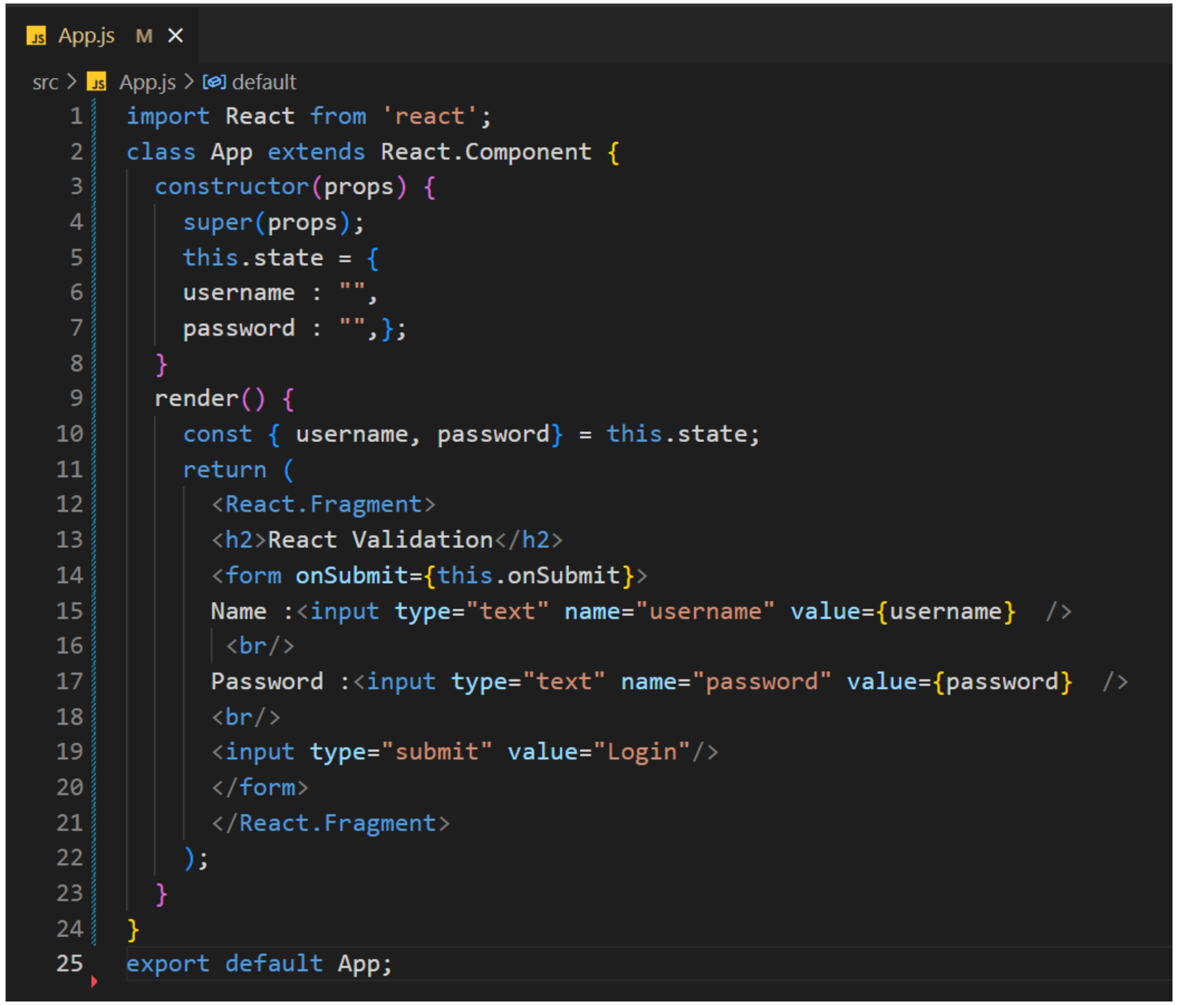
- Create Class Component



```
import React from 'react';
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {  };
  }
  render() {
    return (
      <h2>React Validation</h2>
    );
  }
}
export default App;
```

Design State

- State Value will be Display in Textbox

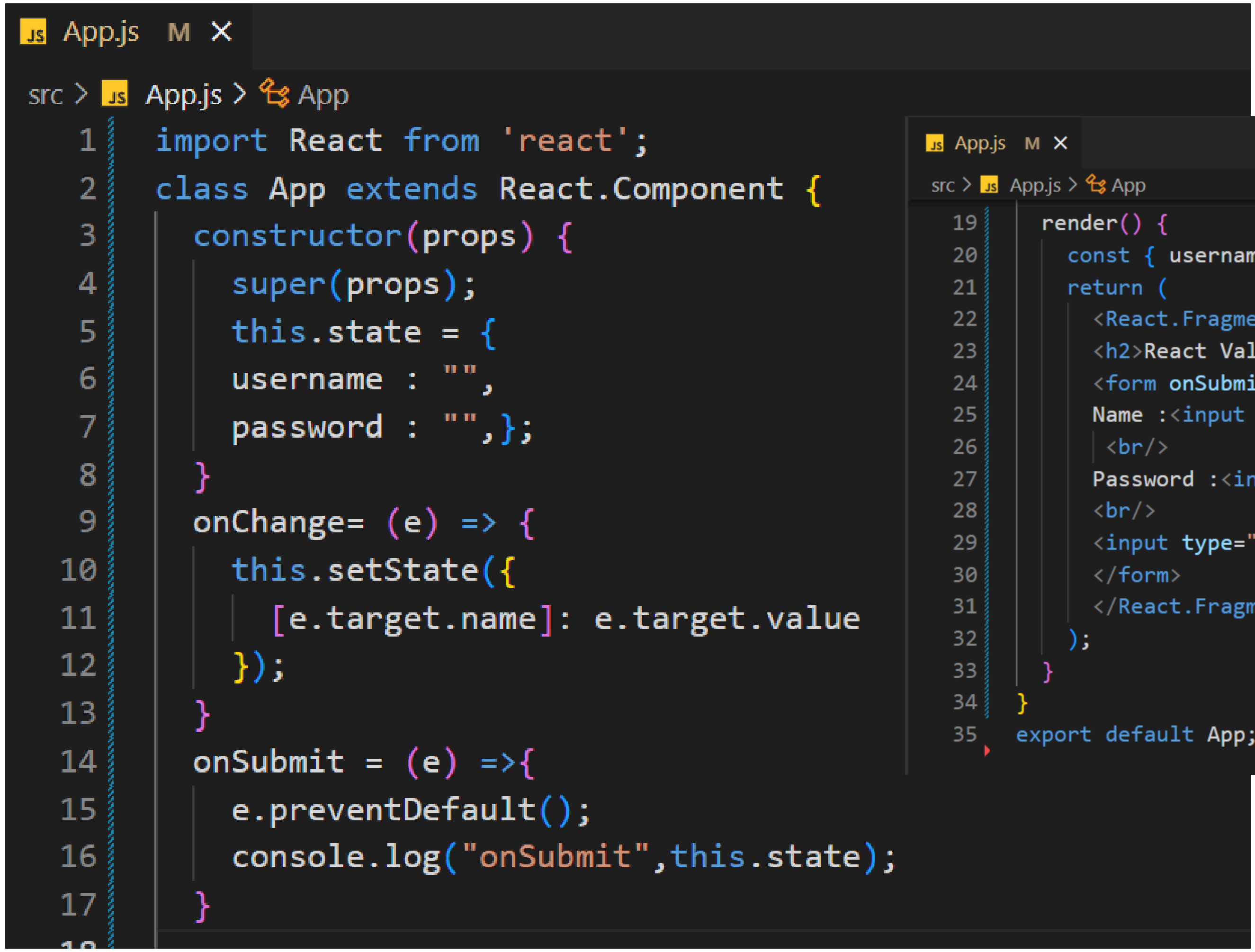


App.js M ×

src > App.js > default

```
import React from 'react';
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
    username : "",
    password : "",};
  }
  render() {
    const { username, password} = this.state;
    return (
      <React.Fragment>
      <h2>React Validation</h2>
      <form onSubmit={this.onSubmit}>
      Name :<input type="text" name="username" value={username}  />
       <br/>
      Password :<input type="text" name="password" value={password}  />
      <br/>
      <input type="submit" value="Login"/>
      </form>
      </React.Fragment>
    );
  }
}
export default App;
```

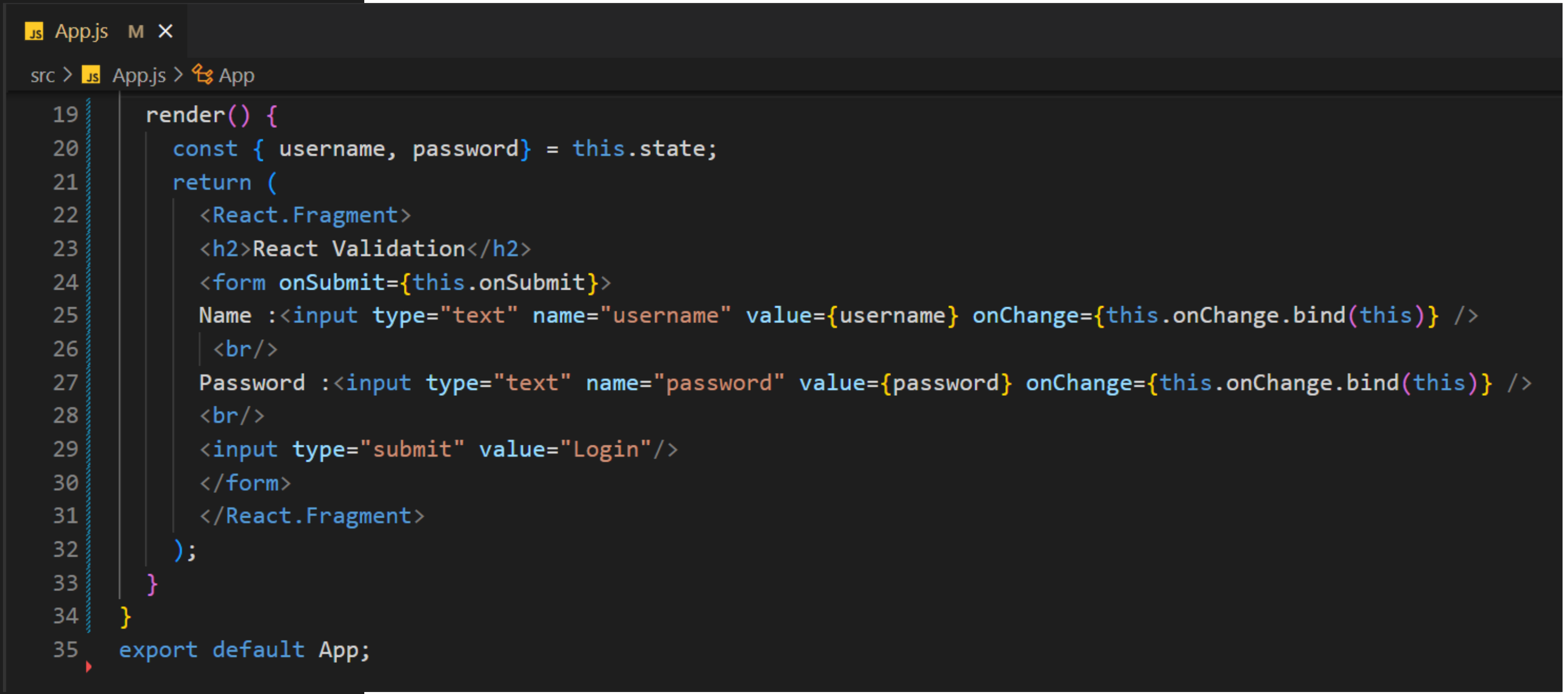
OnChange and OnSubmit



App.js M ×

src > App.js > App

```
import React from 'react';
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
    username : "",
    password : "",};
  }
  onChange= (e) => {
    this.setState({
      [e.target.name]: e.target.value
    });
  }
  onSubmit = (e) =>{
    e.preventDefault();
    console.log("onSubmit",this.state);
  }
```

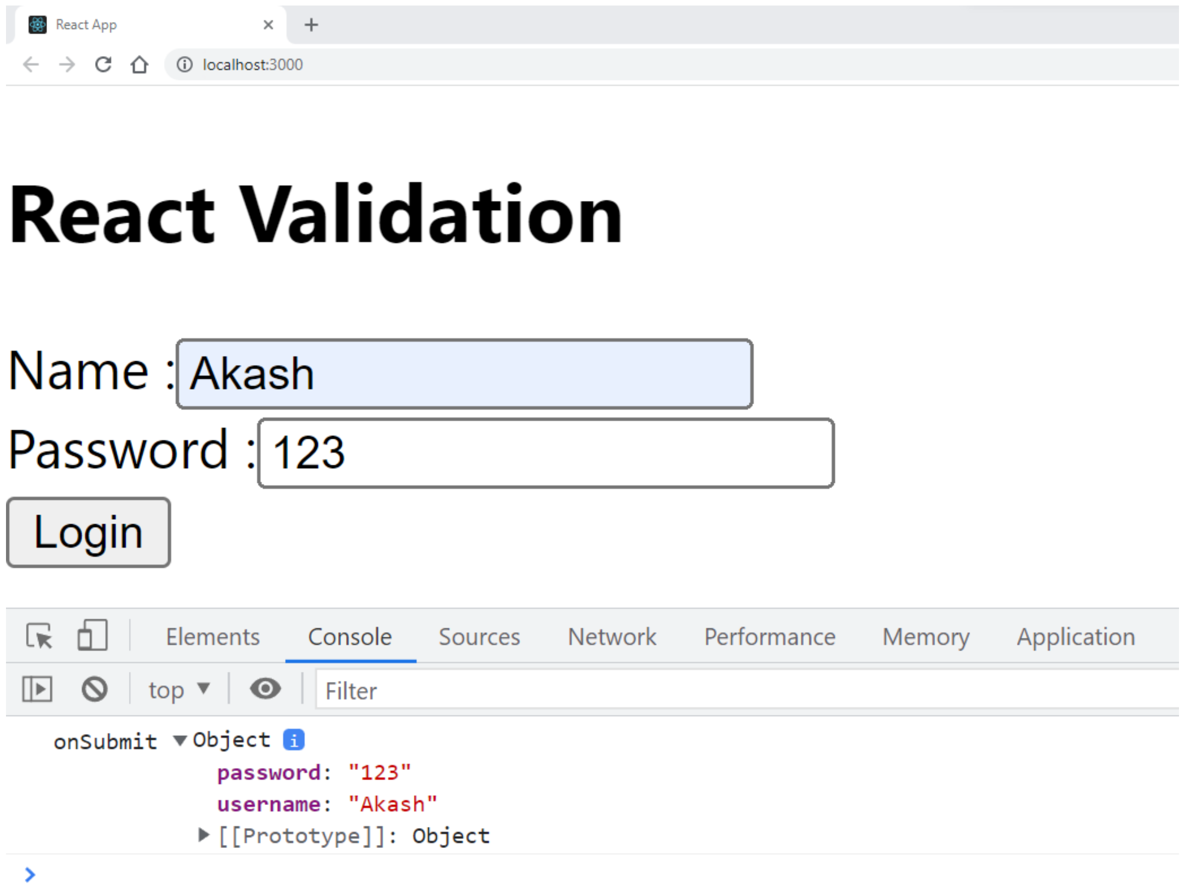


App.js M ×

src > App.js > App

```
  render() {
    const { username, password} = this.state;
    return (
      <React.Fragment>
      <h2>React Validation</h2>
      <form onSubmit={this.onSubmit}>
      Name :<input type="text" name="username" value={username} onChange={this.onChange.bind(this)} />
       <br/>
      Password :<input type="text" name="password" value={password} onChange={this.onChange.bind(this)} />
      <br/>
      <input type="submit" value="Login"/>
      </form>
      </React.Fragment>
    );
  }
}
export default App;
```

Basic Form is Ready with OnChange and Submit ☺



React Validation

Name : Akash

Password : 123

Login

```
onSubmit ▼Object
            password: "123"
            username: "Akash"
          ▶[[Prototype]]: Object
```

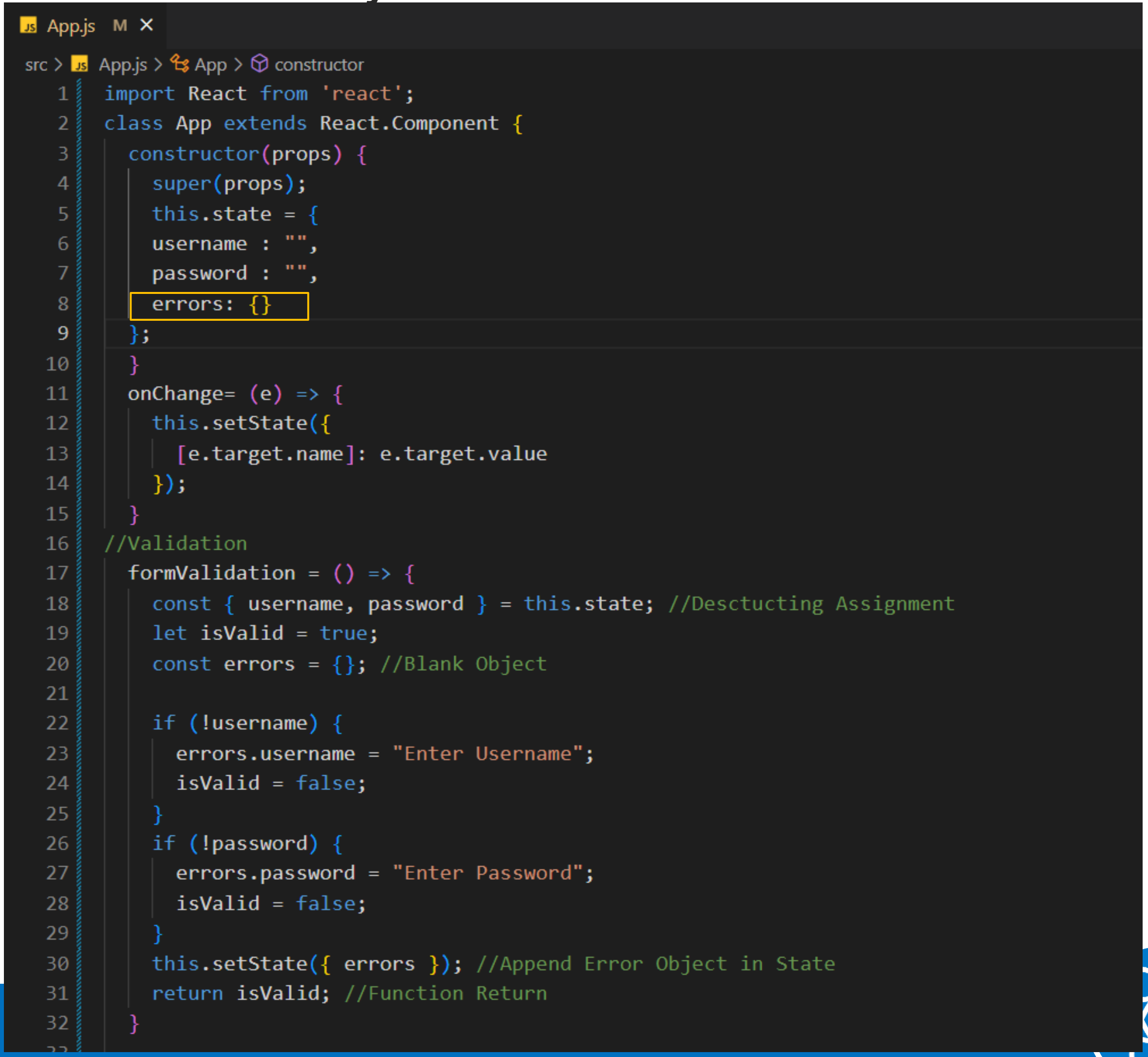
Validation Process

Define Error Object in State

- Create Blank Error Object which will stores all error in Object

```
this.state = {
  username : "",
  password : "",
  errors: {}
 };
```

```
formValidation = () => {
  const { username, password } = this.state; //Desctucting Assignment
  let isValid = true;
  const errors = {}; //Blank Object

  if (!username) {
   errors.username = "Enter Username";
   isValid = false;
  }
  if (!password) {
   errors.password = "Enter Password";
   isValid = false;
  }
  this.setState({ errors }); //Append Error Object in State
  return isValid; //Function Return
}
```


App.js M ×
src > App.js > App > constructor

```
import React from 'react';
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
    username : "",
    password : "",
    errors: {}
  };
  }
  onChange= (e) => {
    this.setState({
      [e.target.name]: e.target.value
    });
  }
//Validation
  formValidation = () => {
    const { username, password } = this.state; //Desctucting Assignment
    let isValid = true;
    const errors = {}; //Blank Object

    if (!username) {
      errors.username = "Enter Username";
      isValid = false;
    }
    if (!password) {
      errors.password = "Enter Password";
      isValid = false;
    }
    this.setState({ errors }); //Append Error Object in State
    return isValid; //Function Return
  }
```

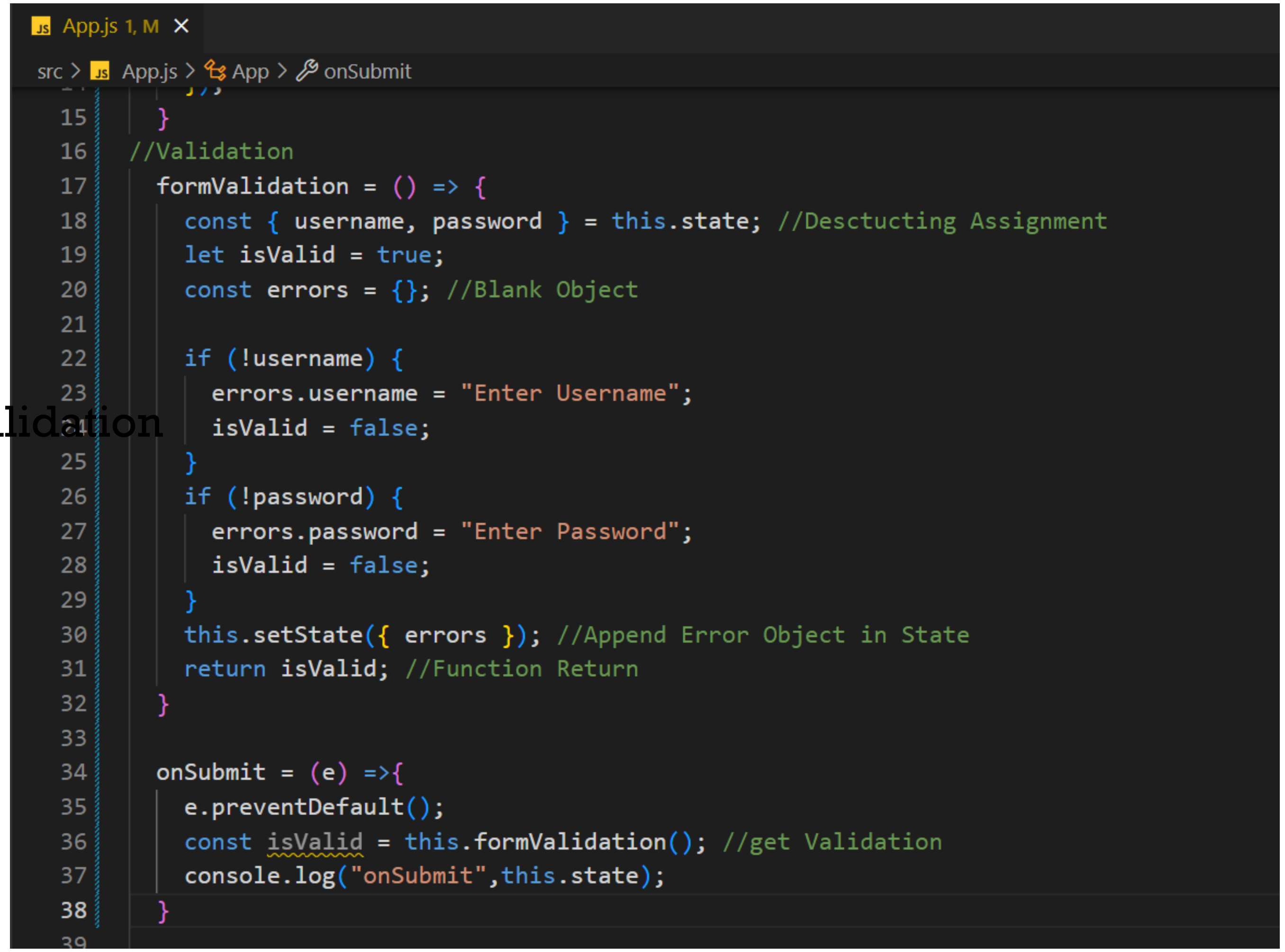
Statement

- First we will create blank object in State
  - errors:{}
- Create One Validation Functions which will return Boolean value.(True/False)
- First Define Boolean variable for validating Data.
- Extract /assign state data in variable
- Write if Condition to check state value is present or not
  - If state value is not present then return false in variable and Store Error message in Error Object.
  - So on.....
- At last Error Object data assign to Error State and Return Boolean value.

- If Data is validation Boolean value will return True and Error Object will return blank.
- Create Submit Event on Submit Print Error Value in Console

OnSubmit Call Validation

```
onSubmit = (e) =>{
  e.preventDefault();
  const isValid = this.formValidation(); //get Validation
  console.log("onSubmit",this.state);
 }
```

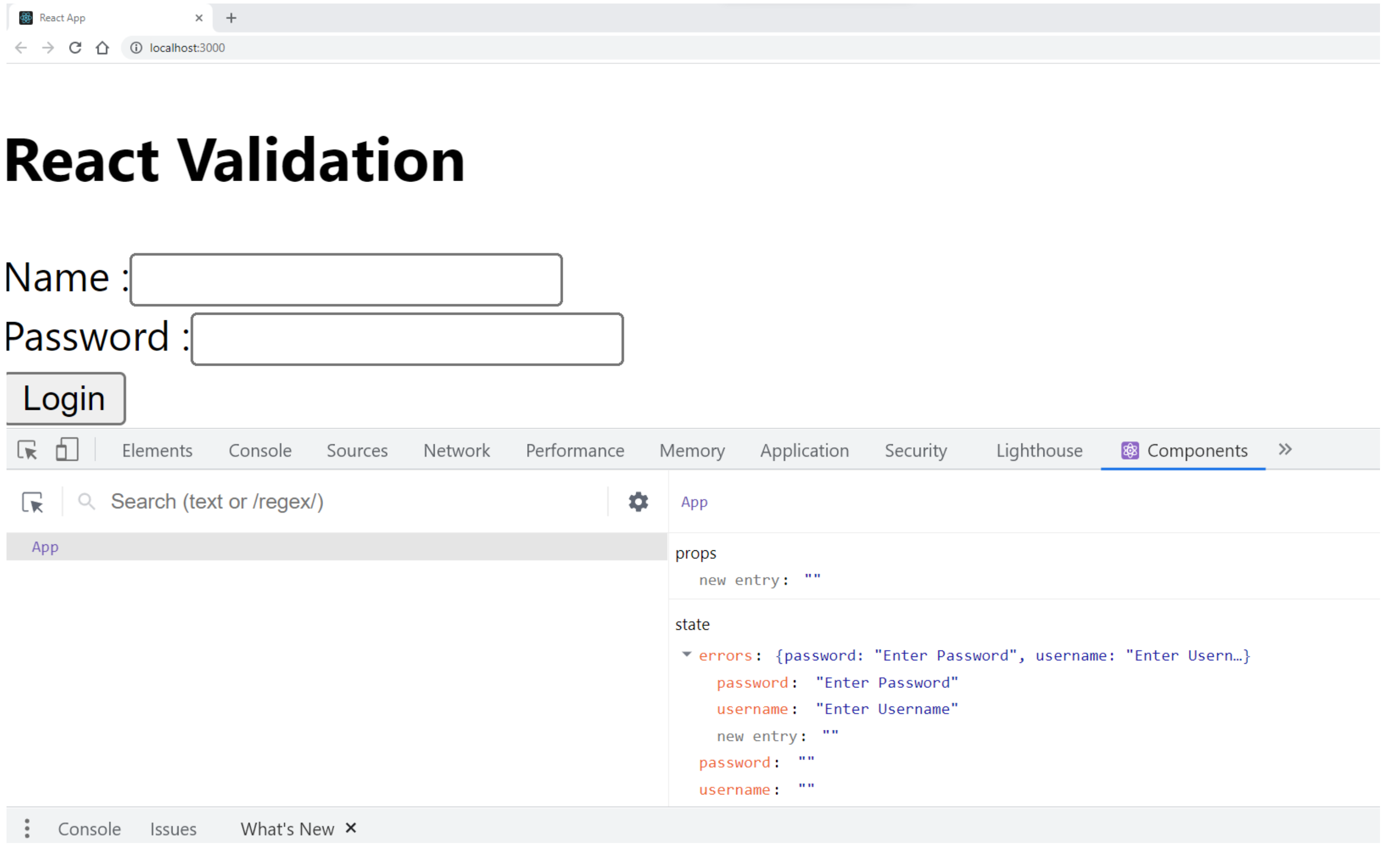

App.js 1, M

src > App.js > App > onSubmit

```
    }
  //Validation
    formValidation = () => {
      const { username, password } = this.state; //Desctucting Assignment
      let isValid = true;
      const errors = {}; //Blank Object

      if (!username) {
        errors.username = "Enter Username";
        isValid = false;
      }
      if (!password) {
        errors.password = "Enter Password";
        isValid = false;
      }
      this.setState({ errors }); //Append Error Object in State
      return isValid; //Function Return
    }

    onSubmit = (e) =>{
      e.preventDefault();
      const isValid = this.formValidation(); //get Validation
      console.log("onSubmit",this.state);
    }
```

OnSubmit

- OnSubmit Error Message will Append in Errors Object and we can access from React Developer Tool



React Validation

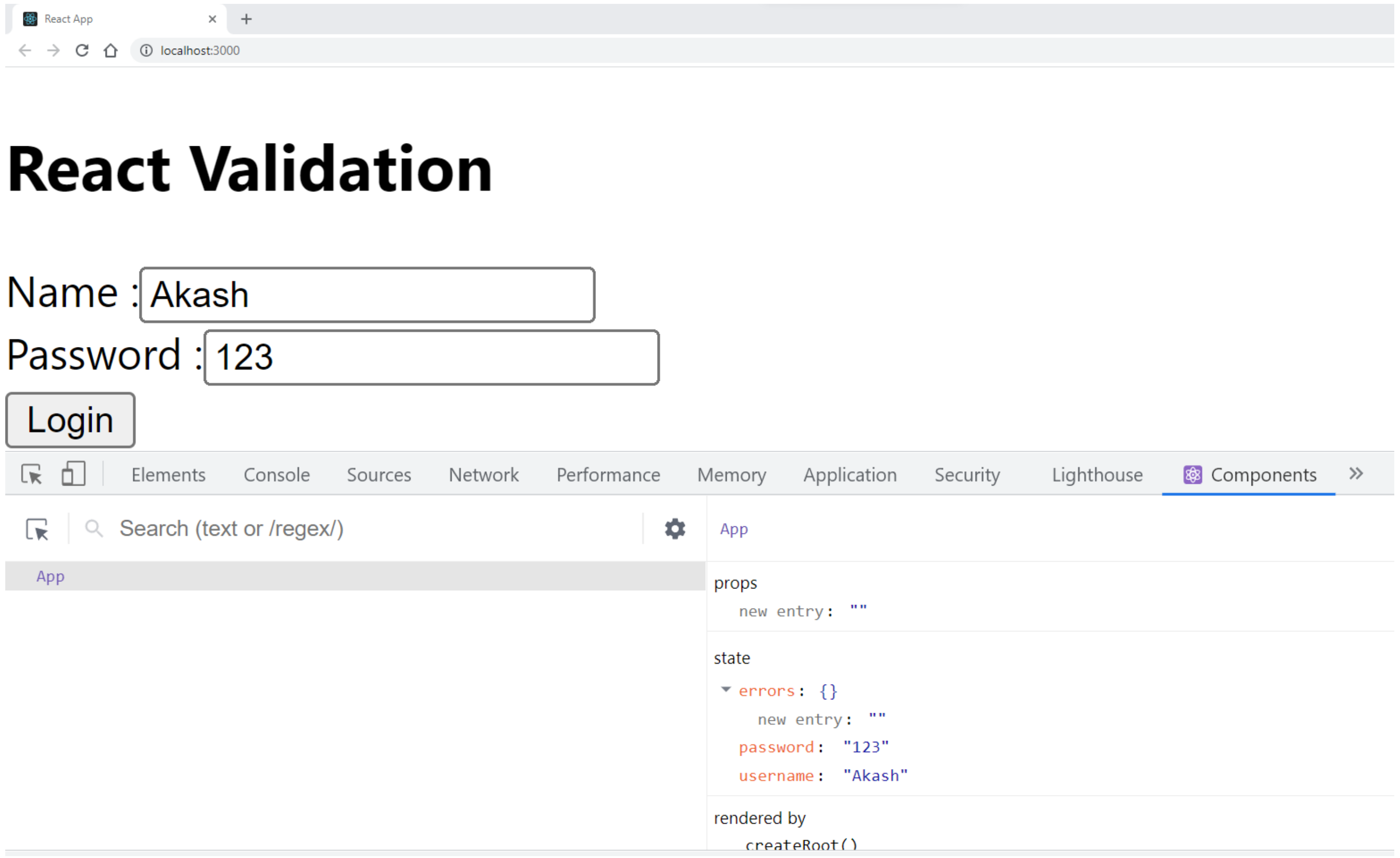
Name :
Password :
Login

Elements Console Sources Network Performance Memory Application Security Lighthouse Components
Search (text or /regex/)
App
App
Console Issues What's New

```
props
  new entry: ""
state
  ▾ errors: {password: "Enter Password", username: "Enter Usern…}
      password: "Enter Password"
      username: "Enter Username"
      new entry: ""
  password: ""
  username: ""
```

Validate Data

- On Valid Details Error Message will Hide



React Validation

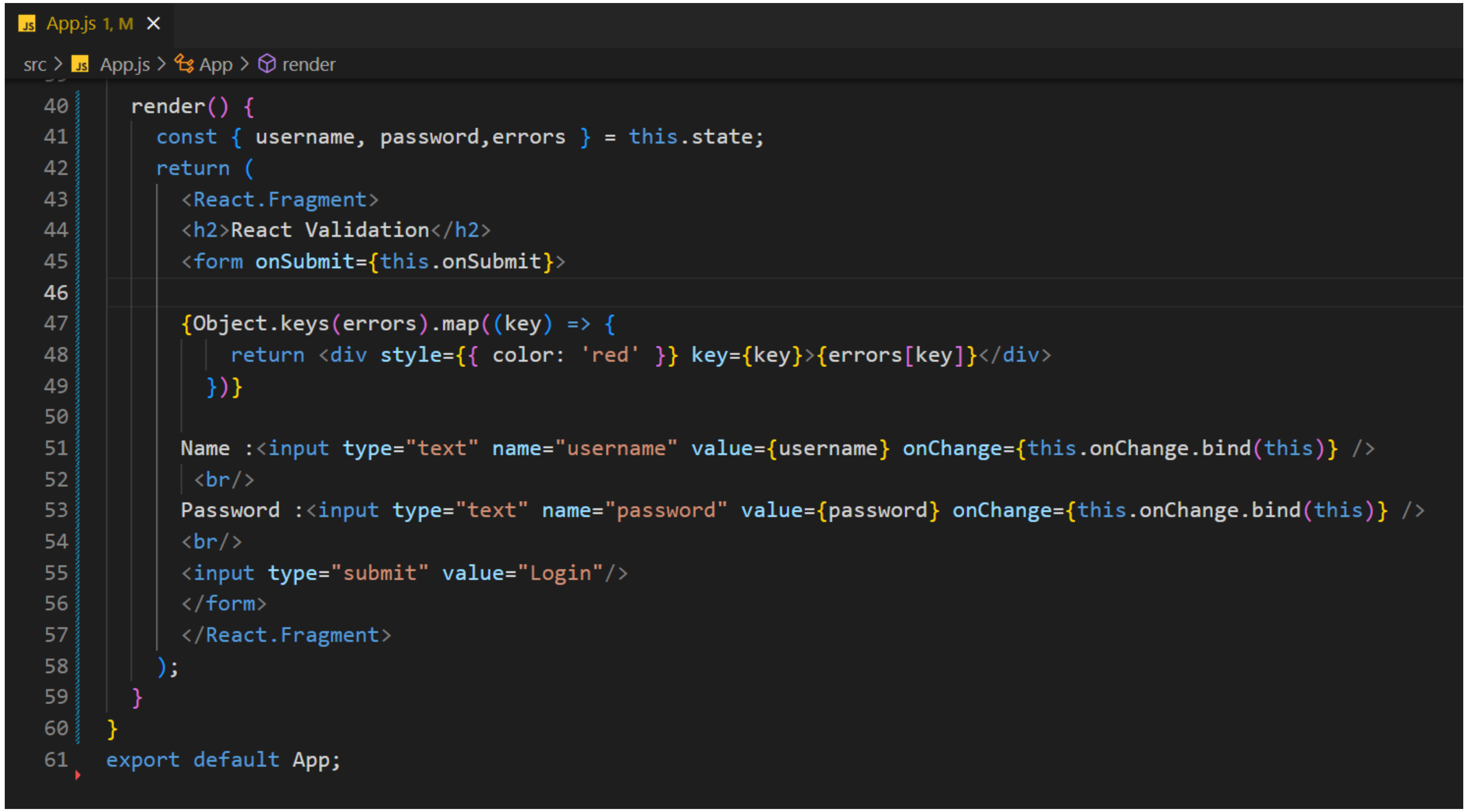
Name : Akash

Password : 123

Login

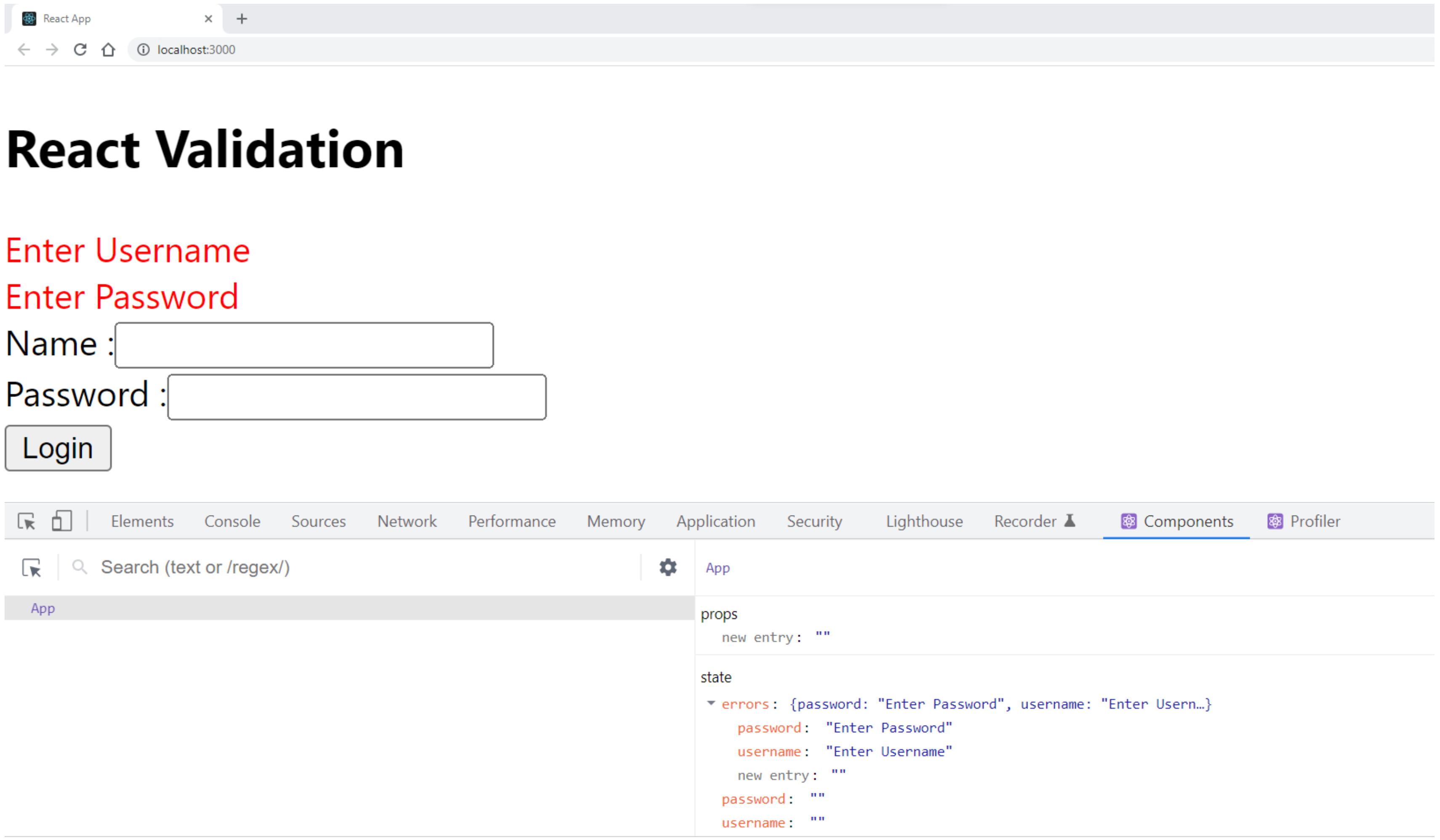
```
props
  new entry: ""
state
  ▾ errors: {}
     new entry: ""
  password: "123"
  username: "Akash"
rendered by
  createRoot()
```

Print Error Using Map



```
  render() {
    const { username, password,errors } = this.state;
    return (
      <React.Fragment>
      <h2>React Validation</h2>
      <form onSubmit={this.onSubmit}>

      {Object.keys(errors).map((key) => {
        return <div style={{ color: 'red' }} key={key}>{errors[key]}</div>
      })}

      Name :<input type="text" name="username" value={username} onChange={this.onChange.bind(this)} />
       <br/>
      Password :<input type="text" name="password" value={password} onChange={this.onChange.bind(this)} />
      <br/>
      <input type="submit" value="Login"/>
      </form>
      </React.Fragment>
    );
  }
}
export default App;
```

Error Display

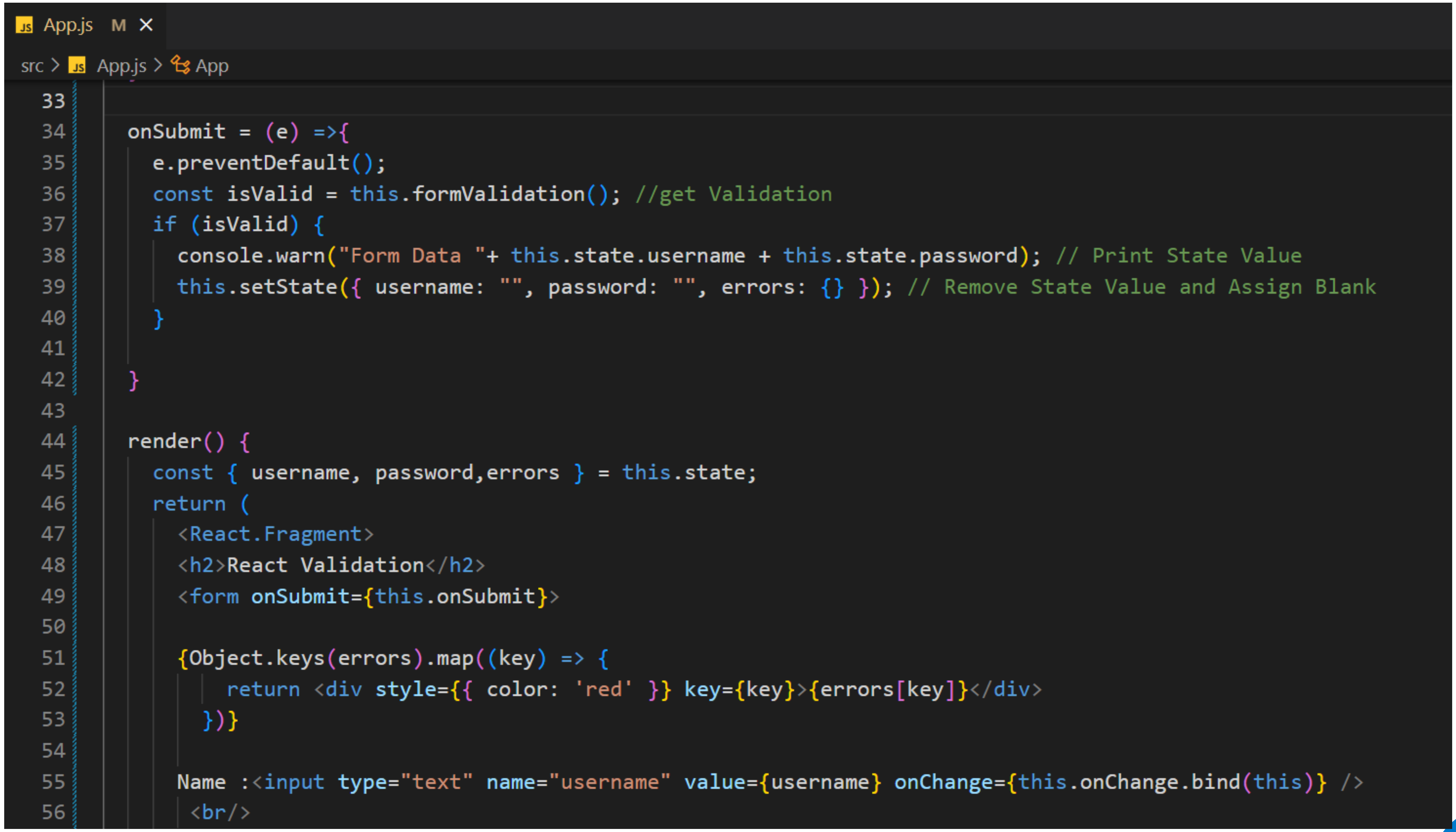


React Validation

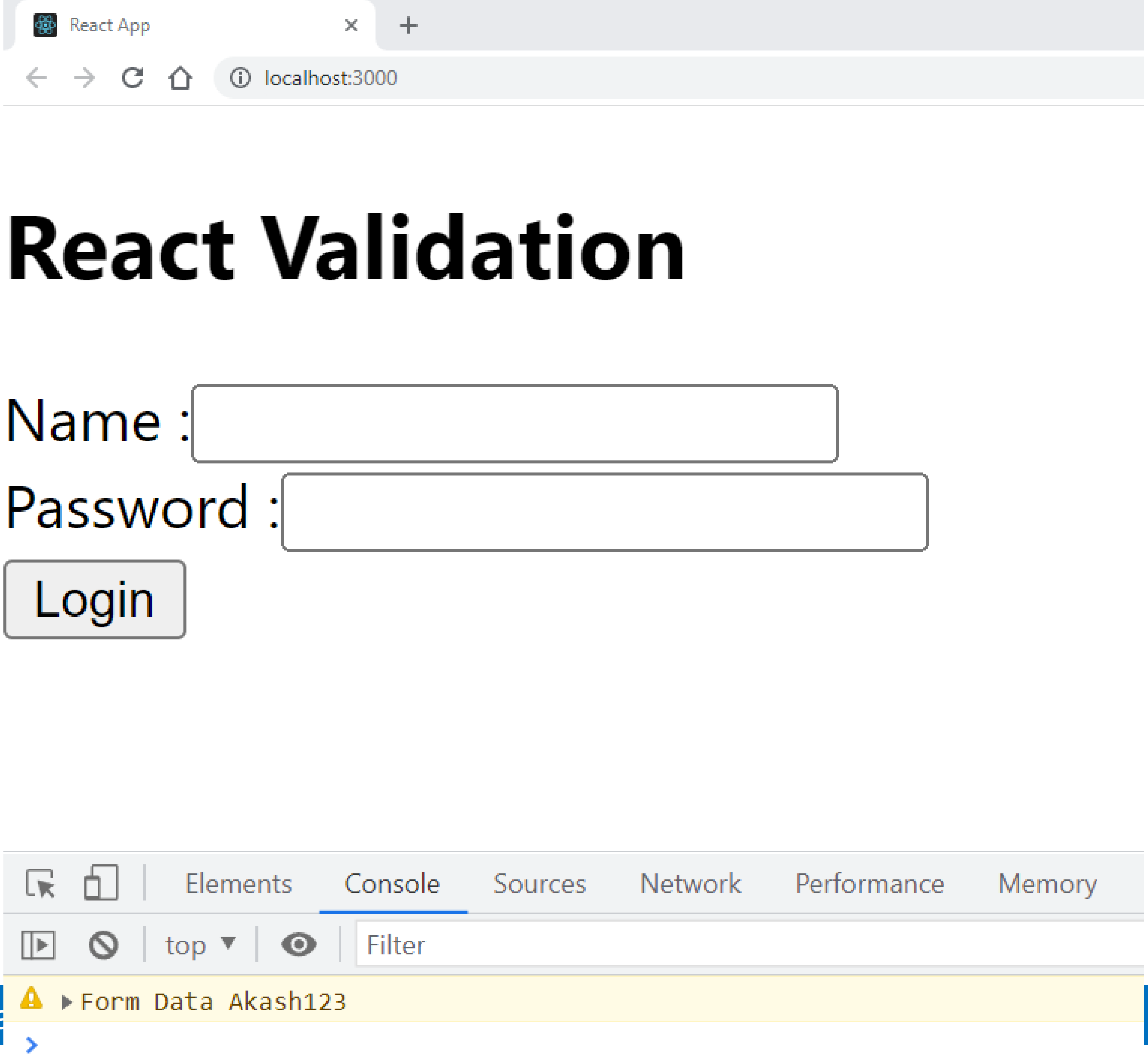
Enter Username
Enter Password
Name :
Password :
Login

```
props
  new entry: ""

state
  ▾ errors: {password: "Enter Password", username: "Enter Usern…}
      password: "Enter Password"
      username: "Enter Username"
      new entry: ""
  password: ""
  username: ""
```

OnSubmit Print Data and Clear State



```

  onSubmit = (e) =>{
    e.preventDefault();
    const isValid = this.formValidation(); //get Validation
    if (isValid) {
      console.warn("Form Data "+ this.state.username + this.state.password); // Print State Value
      this.setState({ username: "", password: "", errors: {} }); // Remove State Value and Assign Blank
    }

  }

  render() {
    const { username, password,errors } = this.state;
    return (
      <React.Fragment>
      <h2>React Validation</h2>
      <form onSubmit={this.onSubmit}>

      {Object.keys(errors).map((key) => {
        return <div style={{ color: 'red' }} key={key}>{errors[key]}</div>
      })}

      Name :<input type="text" name="username" value={username} onChange={this.onChange.bind(this)} />
       <br/>
```



React App

localhost:3000

React Validation

Name :

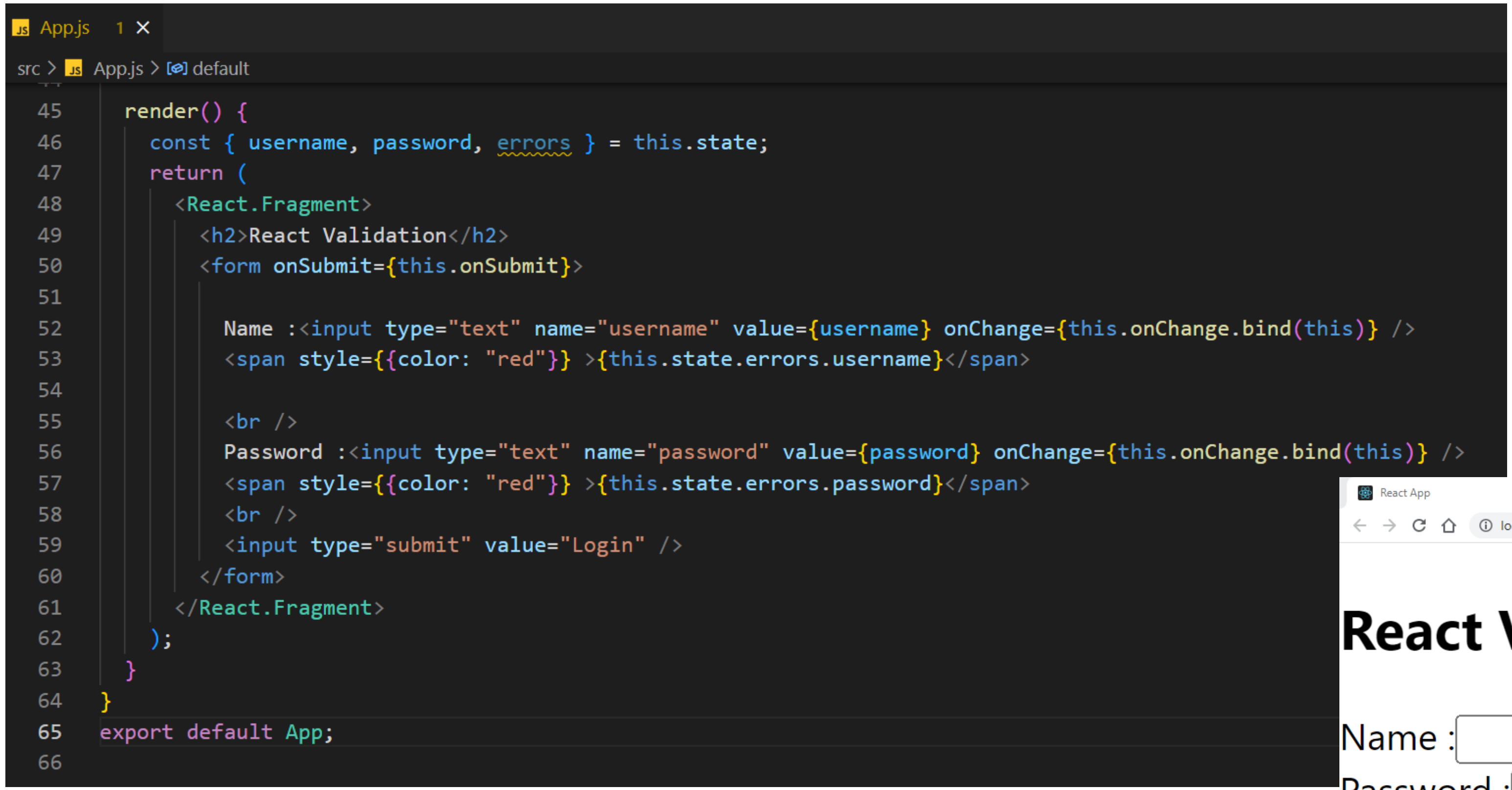
Password :

Login

```
Form Data Akash123
```

Example

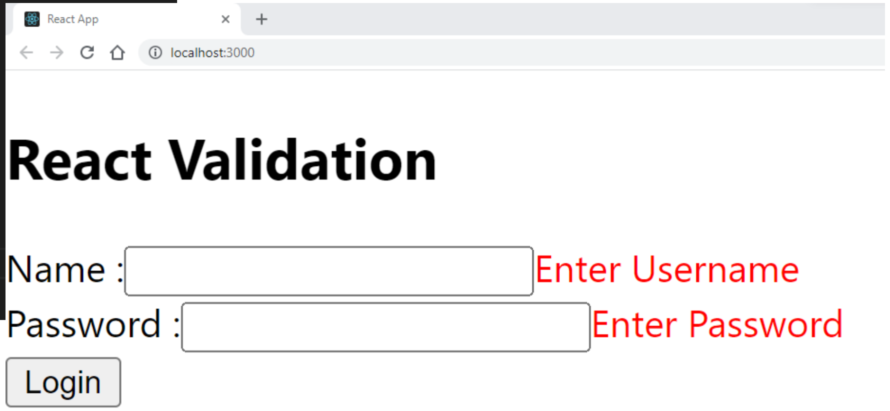
- <span style={{color: "red"}} >{this.state.errors.username}</span>



App.js 1

src > App.js > default

```
  render() {
    const { username, password, errors } = this.state;
    return (
      <React.Fragment>
        <h2>React Validation</h2>
        <form onSubmit={this.onSubmit}>

          Name :<input type="text" name="username" value={username} onChange={this.onChange.bind(this)} />
          <span style={{color: "red"}} >{this.state.errors.username}</span>

          <br />
          Password :<input type="text" name="password" value={password} onChange={this.onChange.bind(this)} />
          <span style={{color: "red"}} >{this.state.errors.password}</span>
          <br />
          <input type="submit" value="Login" />
        </form>
      </React.Fragment>
    );
  }
}
export default App;
```



React App

localhost:3000

React Validation

Name : Enter Username

Password : Enter Password

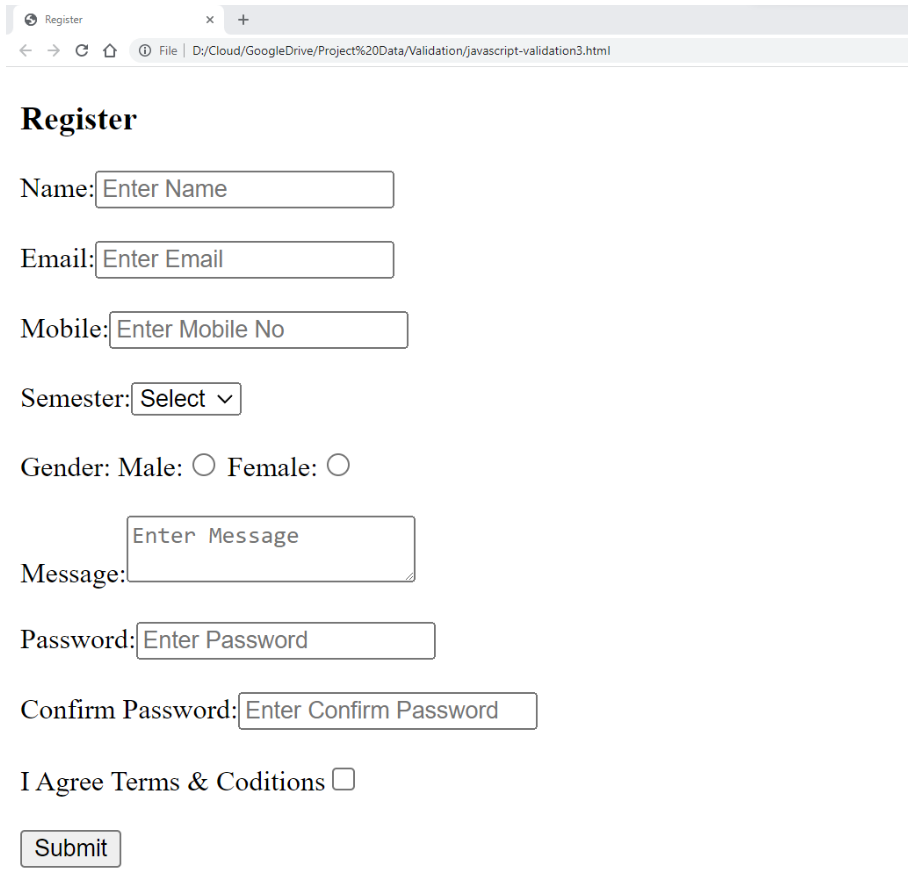
Login

```
import React from 'react';
class App extends React.Component {
 constructor(props) {
  super(props);
  this.state = {
  username : "",
  password : "",
  errors: {}
 };
 }
 onChange= (e) => {
  this.setState({
    [e.target.name]: e.target.value
  });
 }
//Validation
 formValidation = () => {
  const { username, password } = this.state; //Desctucting Assignment
  let isValid = true;
  const errors = {}; //Blank Object

  if (!username) {
    errors.username = "Enter Username";
    isValid = false;
  }
  if (!password) {
    errors.password = "Enter Password";
    isValid = false;
  }
  this.setState({ errors }); //Append Error Object in State
  return isValid; //Function Return
 }
```

```
 onSubmit = (e) =>{
  e.preventDefault();
  const isValid = this.formValidation(); //get Validation
  if (isValid) {
    console.warn("Form Data "+ this.state.username + this.state.password); // Print State Value
    this.setState({ username: "", password: "", errors: {} }); // Remove State Value and Assign Blank
  }

 }

 render() {
  const { username, password,errors } = this.state;
  return (
    <React.Fragment>
    <h2>React Validation</h2>
    <form onSubmit={this.onSubmit}>

    {Object.keys(errors).map((key) => {
       return <div style={{ color: 'red' }} key={key}>{errors[key]}</div>
     })}

    Name :<input type="text" name="username" value={username} onChange={this.onChange.bind(this)} />
     <br/>
    Password :<input type="text" name="password" value={password} onChange={this.onChange.bind(this)} />
    <br/>
    <input type="submit" value="Login"/>
    </form>
    </React.Fragment>
  );
 }
}
export default App;
```

Task

- Make Same Form in Class with Proper Validation
- Create Same program using Hooks


Register

File | D:/Cloud/GoogleDrive/Project%20Data/Validation/javascript-validation3.html

Register

Name: Enter Name

Email: Enter Email

Mobile: Enter Mobile No

Semester: Select

Gender: Male: ○ Female: ○

Message: Enter Message

Password: Enter Password

Confirm Password: Enter Confirm Password

I Agree Terms & Coditions ☐

Submit

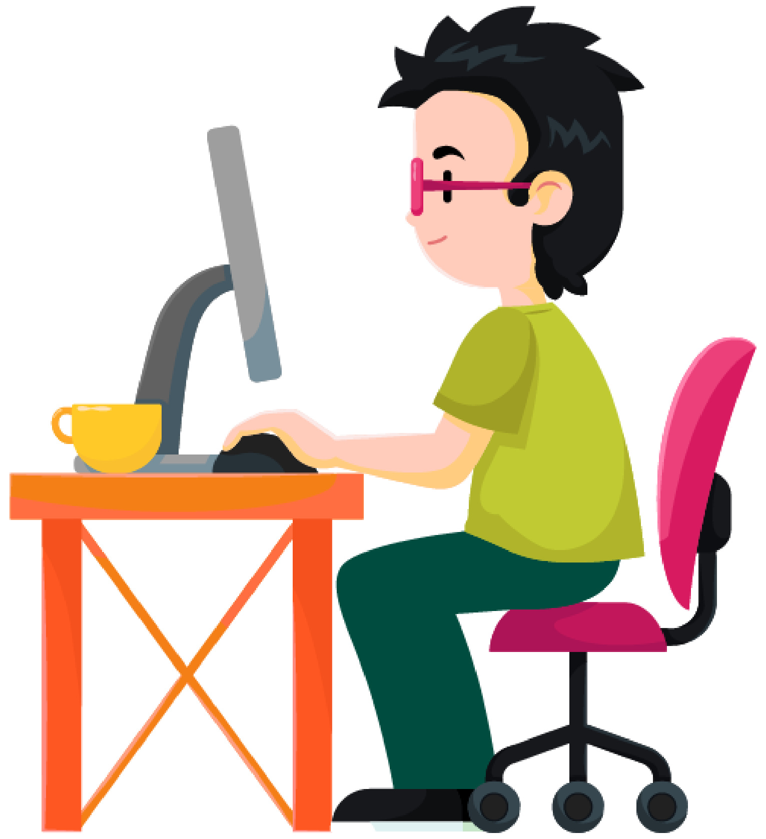
```
 if (!values.name) {
         errors.name = 'Name is required';
       } else if (values.name.length > 15) {
         errors.name = 'Must be 15 characters or less';
       }


       if (!values.email) {
         errors.email = 'Email is required';
       } else if (!/^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-
Z]{2,4}$/i.test(values.email)) {
         errors.email = 'Invalid email address';
       }
```

```
const validateEmployee = empData => {
  const errors = {};

  if (!empData.Name) {
    errors.Name = 'Please Enter Employee Name';
  } else if (empData.Name.length > 20) {
    errors.Name = 'Name cannot exceed 20 characters';
  }

  if (!empData.Location) {
    errors.Location = 'Please Enter Employee Location';
  }

  if (!empData.EmailId) {
    errors.EmailId = 'Please Enter Email ID';
  } else if (!/^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-
Z]{2,4}$/i.test(empData.EmailId)) {
    errors.EmailId = 'Invalid email address';
  }

  return errors;
};
```

Validation Rules

- All Fields must be required
- Name must be Character ,
- Minimum 2 Character required in Name and Maximum 10 Characters.
- Email must be Valid Format
- Mobile Number must be == 10 Digits only
- Password and Confirm Password Must be Same.
- Take a Radio,CheckBox and Dropdown Field in Form with Proper Validation.

Get Exclusive Video Tutorials



www.aptutorials.com

https://www.youtube.com/user/Akashtips

Get More Details

www.akashsir.com

If You Liked It !

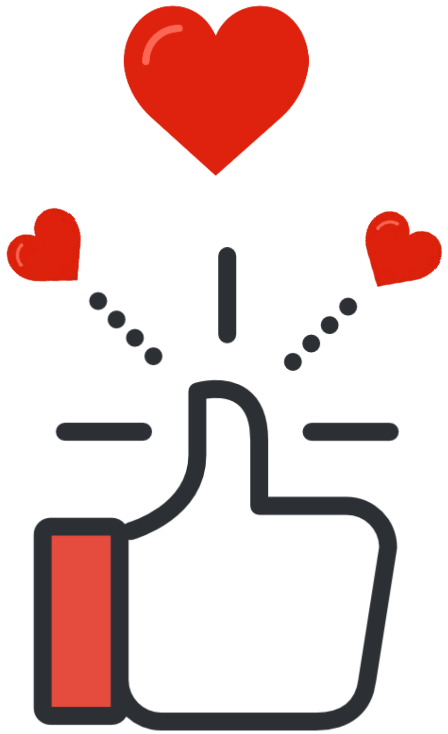
Rating Us Now

Just Dial

https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4_BZDET

Sulekha

https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad



Connect With Me

Akash Padhiyar
#AkashSir

www.akashsir.com
www.akashtechnolabs.com
www.akashpadhiyar.com
www.aptutorials.com

# Social Info

Akash.padhiyar

Akashpadhiyar

Akash_padhiyar

+91 99786-21654

#Akashpadhiyar
#aptutorials