WEB TECHNOLOGIES

React JS – Form Handling

Prof. Vinay Joshi and Prof. Sindhu R Pai
Department of Computer Science and Engineering

ReactJS - Form Handling

Agenda

- Introduction
- Uncontrolled Components
- Controlled components
- Sample codes
- Handling multiple inputs
- Practice Program Demo

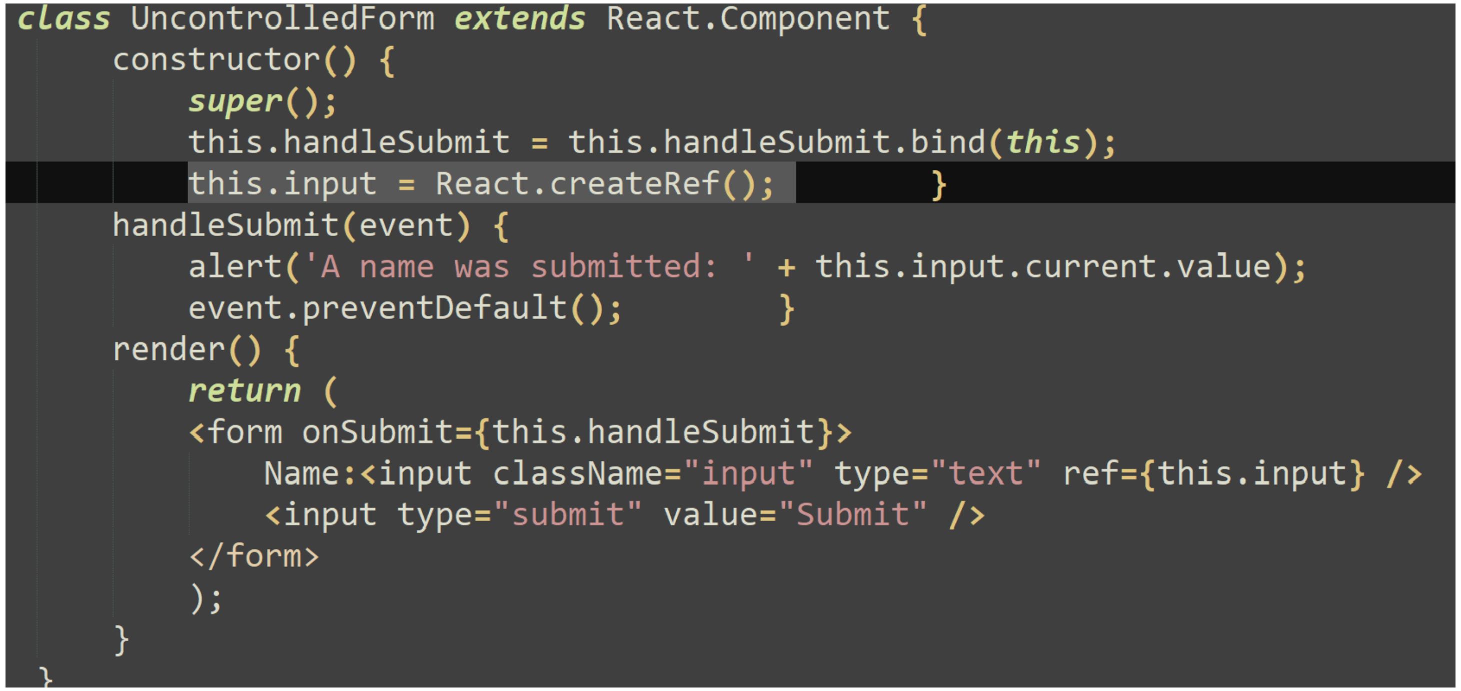
ReactJS - Form Handling
Introduction

- Deals with **how to handle the data when the input values are changed or when the form is submitted**
- Control these changes by adding **event handlers** to **onChange** and **onSubmit** respectively
- Two different ways
  - **Uncontrolled components :** Data handled by DOM.
  - **Controlled components:** Data handled by Components
    - When the data is handled by the components, all the data is stored in the **component state**
    - Form data is usually handled by the components.

ReactJS - Form Handling

Uncontrolled components

- Use a ref to get form values from the DOM
- Use the defaultValue property to specify initial value in React
- <input **defaultValue="Bob"** type="text" **ref={this.input}** />



```
class UncontrolledForm extends React.Component {
    constructor() {
        super();
        this.handleSubmit = this.handleSubmit.bind(this);
        this.input = React.createRef();       }
    handleSubmit(event) {
        alert('A name was submitted: ' + this.input.current.value);
        event.preventDefault();       }
    render() {
        return (
        <form onSubmit={this.handleSubmit}>
            Name:<input className="input" type="text" ref={this.input} />
            <input type="submit" value="Submit" />
        </form>
        );
    }
 }
```

ReactJS - Form Handling

Controlled components

- Has two aspects:
  - Have functions to govern the data going into them on every **onChange event,** rather than grabbing the data only once
    - Examples: When a user clicks a submit button. This 'governed' data is then saved to state
  - Data displayed by a controlled component is received through **props** passed down from it's parent/container component.
- Value attribute is set on our form element, the displayed value will always be **this.state.value**, making the React state the **source of truth.** Since handleChange runs on every keystroke to update the React state, **the displayed value will update as the user types**

ReactJS - Form Handling

Sample codes related to Controlled components

- text inputs, number inputs, radio inputs, checkbox inputs, textareas, selects

- Sample codes:
  - <textarea> Default Value </textarea> can be changed to

    <textarea value={this.state.value} />
  - <select>

    <option select value = “optionselected”> Default Value </option>

    </select> can be changed to

    <select value = {this.state.value}>

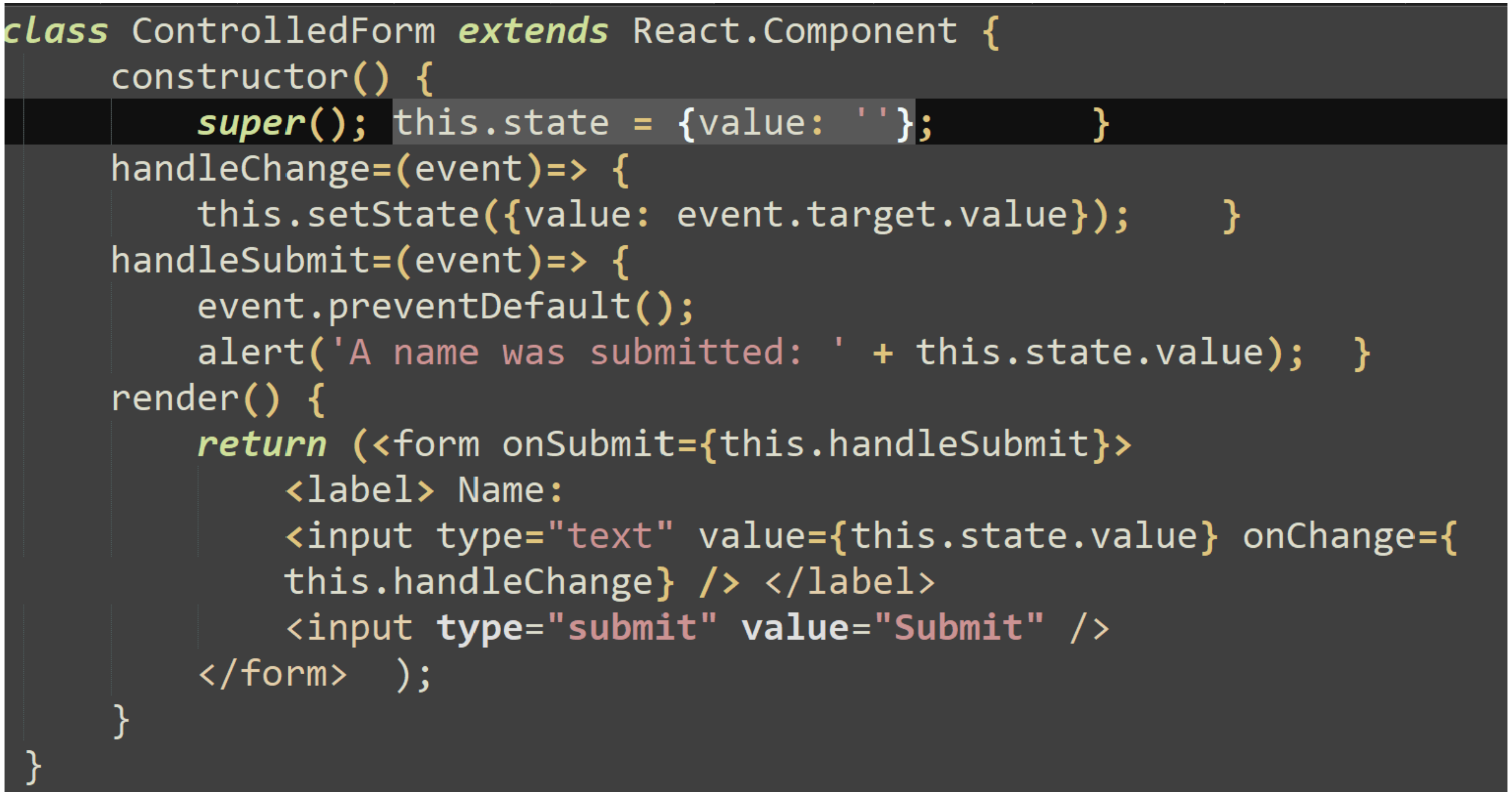
    <option value = “optionselected”>Default Value</option>

    </select>

ReactJS - Form Handling

Handling Multiple inputs

- First case: Single input

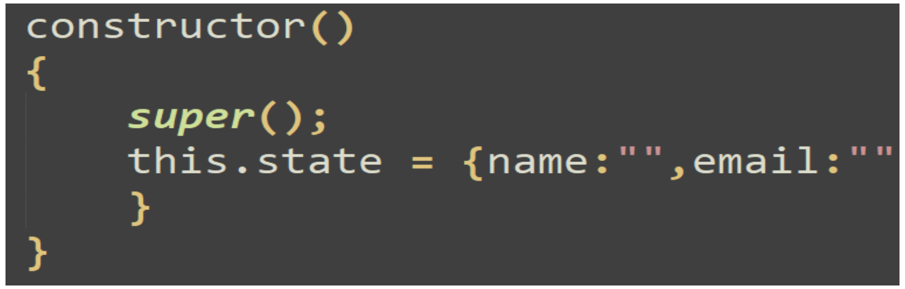


```
class ControlledForm extends React.Component {
    constructor() {
        super(); this.state = {value: ''};        }
    handleChange=(event)=> {
        this.setState({value: event.target.value});    }
    handleSubmit=(event)=> {
        event.preventDefault();
        alert('A name was submitted: ' + this.state.value);   }
    render() {
        return (<form onSubmit={this.handleSubmit}>
            <label> Name:
            <input type="text" value={this.state.value} onChange={
            this.handleChange} /> </label>
            <input type="submit" value="Submit" />
        </form>  );
    }
}
```

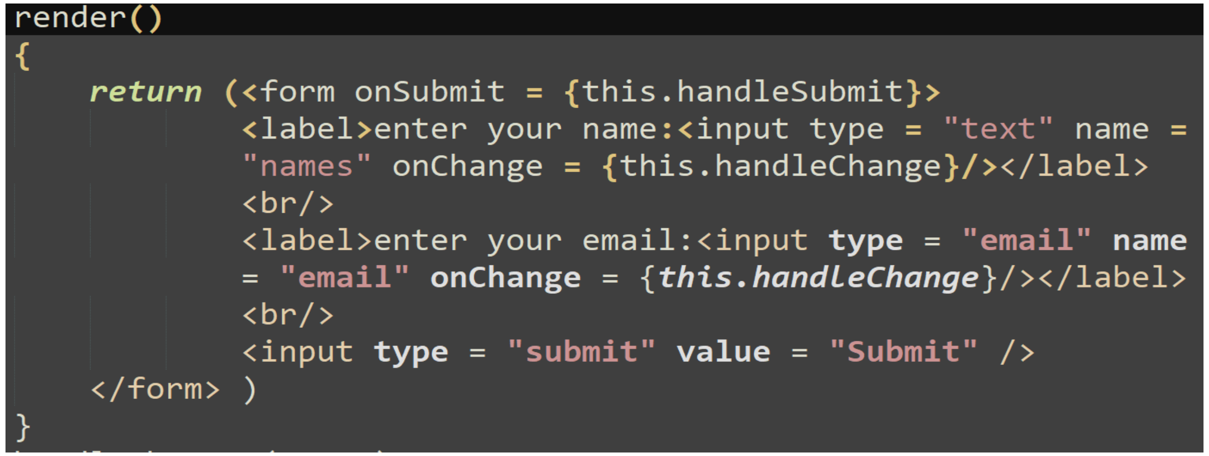
ReactJS - Form Handling

Handling Multiple inputs contd

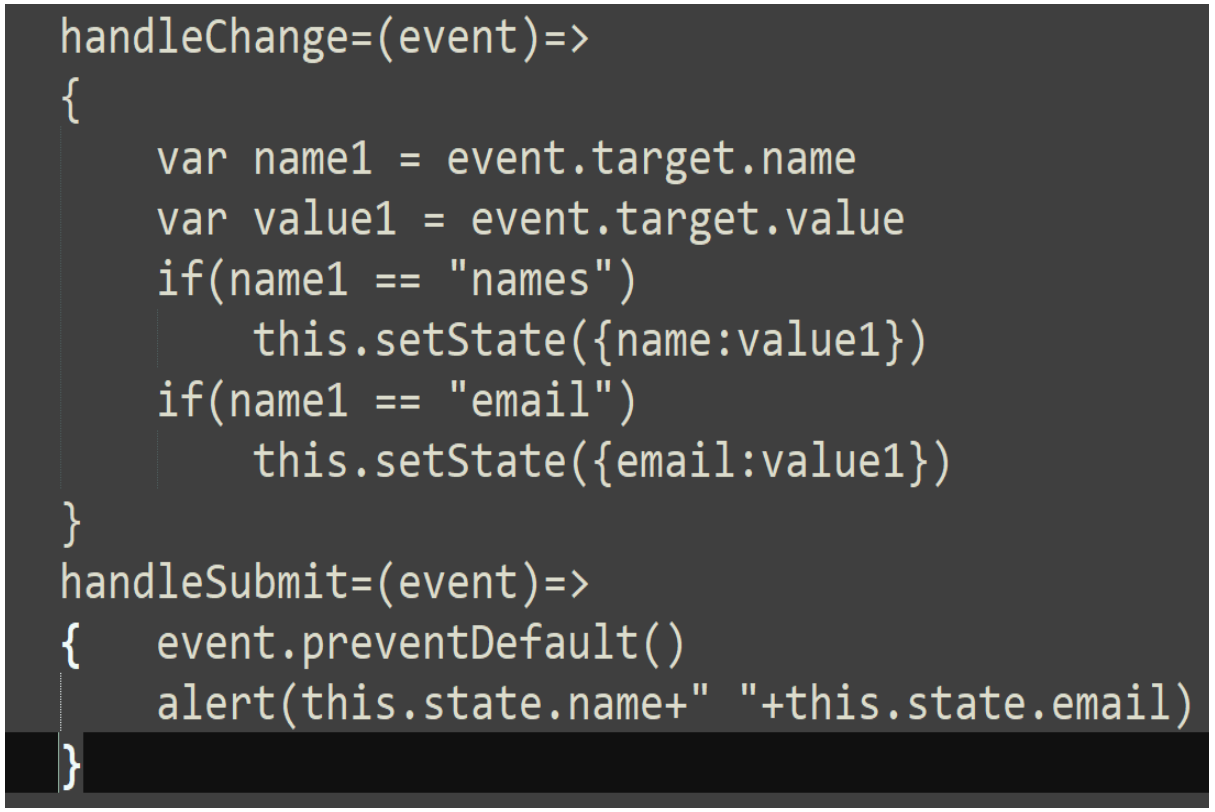
- Second case: Multiple input



```
constructor()
{
    super();
    this.state = {name:"",email:""
    }
}
```



```
render()
{
    return (<form onSubmit = {this.handleSubmit}>
            <label>enter your name:<input type = "text" name =
            "names" onChange = {this.handleChange}/></label>
            <br/>
            <label>enter your email:<input type = "email" name
            = "email" onChange = {this.handleChange}/></label>
            <br/>
            <input type = "submit" value = "Submit" />
    </form> )
}
```



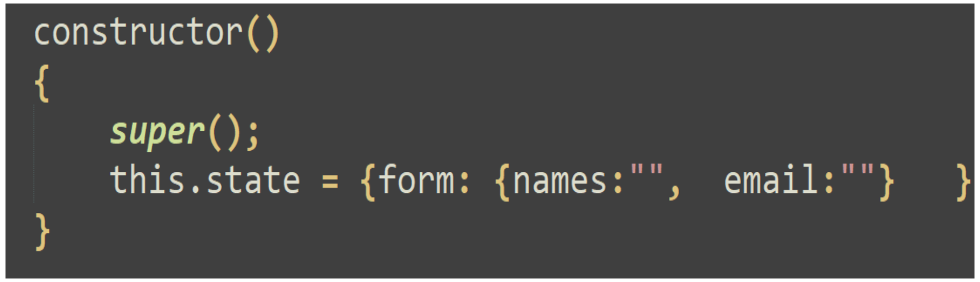
```
handleChange=(event)=>
{
    var name1 = event.target.name
    var value1 = event.target.value
    if(name1 == "names")
        this.setState({name:value1})
    if(name1 == "email")
        this.setState({email:value1})
}
handleSubmit=(event)=>
{   event.preventDefault()
    alert(this.state.name+" "+this.state.email)
}
```

Note: If the form contains many fields, this code might not be ideal to use. Reason being event handler contains the code to check the event.target.name on every field before updating the state

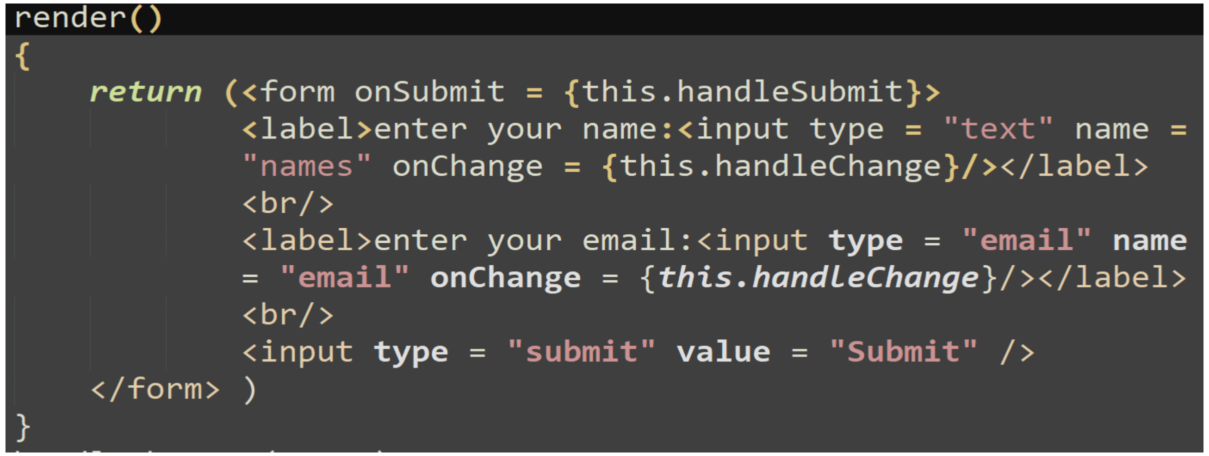
ReactJS - Form Handling

Handling Multiple inputs contd

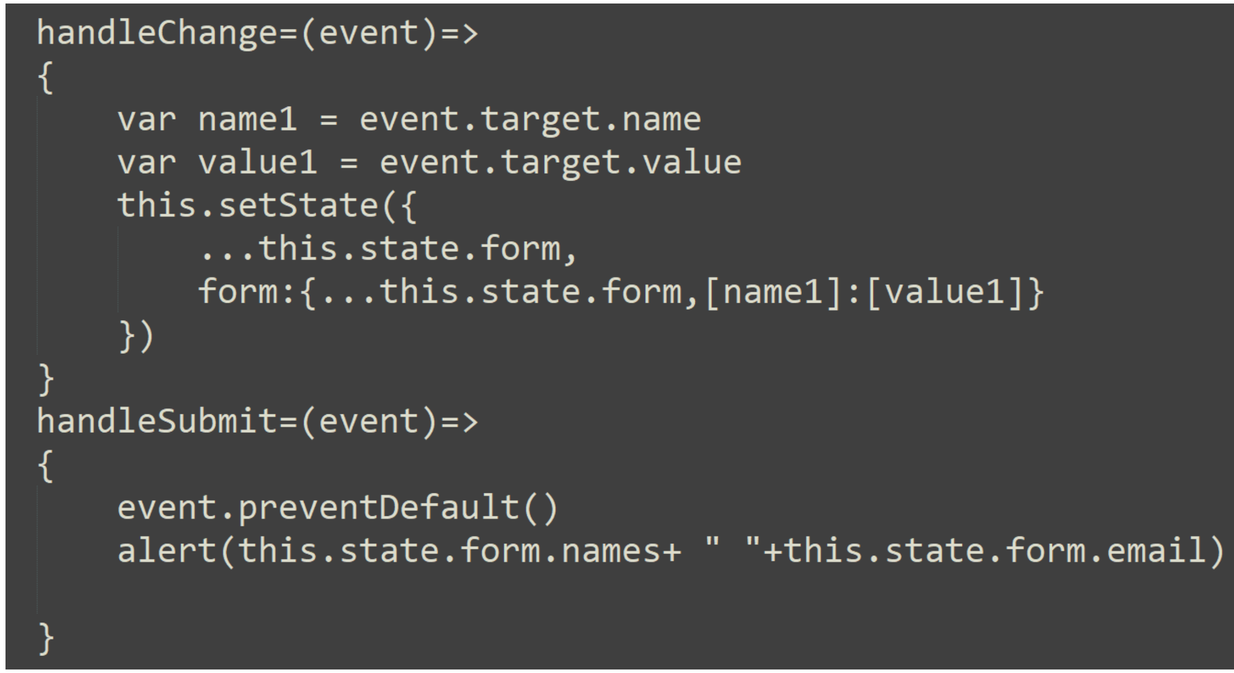
- Solution code to previous Problem



```
constructor()
{
    super();
    this.state = {form: {names:"",  email:""}   }
}
```



```
render()
{
    return (<form onSubmit = {this.handleSubmit}>
            <label>enter your name:<input type = "text" name =
            "names" onChange = {this.handleChange}/></label>
            <br/>
            <label>enter your email:<input type = "email" name
            = "email" onChange = {this.handleChange}/></label>
            <br/>
            <input type = "submit" value = "Submit" />
    </form> )
}
```



```
handleChange=(event)=>
{
    var name1 = event.target.name
    var value1 = event.target.value
    this.setState({
        ...this.state.form,
        form:{...this.state.form,[name1]:[value1]}
    })
}
handleSubmit=(event)=>
{
    event.preventDefault()
    alert(this.state.form.names+ " "+this.state.form.email)

}
```

ReactJS - Form Handling

Practice Program Demo

- Calculate the Body Mass Index of a person, given the height in meters and weight in kilograms. Also display appropriate message.
  - bmi = weight/(height*height)
    - If bmi < 19, display “underweight”
    - If bmi is between 20 and 24, display “Normal”
    - Else display “overweight”

PES UNIVERSITY ONLINE

THANK YOU

Vinay Joshi and Sindhu R Pai
Department of Computer Science and Engineering

vinayj@pes.edu
+91 80 2672 6622

sindhurpai@pes.edu
+91 8277606459