

1. Create new React app
2. Clean your react app
3. Update **App.js** to have class component
4. Create two new separated components: **Form** and **View**
5. In **Form** component have inputs (and labels) for:
 - First name (input)
 - Last name (input)
 - Phone number (input)
 - Role (select)
 - Message (textarea)
6. In **View** component includes:
 - Heading: "This is your input:"
 - Paragraphs with placeholders for:
 - First name
 - Last name
 - Phone number
 - Role
 - Message
7. Add **states** (firstname, lastname, phonenumber, role, message) to the App component and pass data to **View** component (example: *props.firstname* and *firstname={this.state.firstname}*)
8. Add an **event-handler method** to manipulate the state (*event.target.value*). Note, to handle multiple inputs add a name attribute to each element and let the handler function choose what to do based on the value of **event.target.name**
9. Pass the **event-handler method** reference to the **Form** component and bind it to the input-change event (onChange)
10. Make new component: **Popup**
11. In **Popup** component make placeholders for:
 - First name, last name, phone number, role and message
 - Add close button for reload application
 - Hide popup by default (state – showPopup: false)
12. Add **submit button** to **Form** component and add check for empty inputs and textarea (by default react don't know how to check select, so you can skip it).
13. Add an **event-handler method** for showing **Popup** component with submitted data.
14. Add styling of your choice. Here is example of one possible design:

First name Margit

Last name Tennosaar

Phone number 05012234

Message This is message from future...

Role Teacher

SEND

Check your input

First name	<i>Margit</i>
Last name	<i>Tennosaar</i>
Phonenumber	<i>05012234</i>
Message	<i>This is message from future...</i>
Role	<i>teacher</i>

15. Add screenshot from your application and add it to readme file.

You just learned how to **make controlled components** and **work with forms**.
Well done!