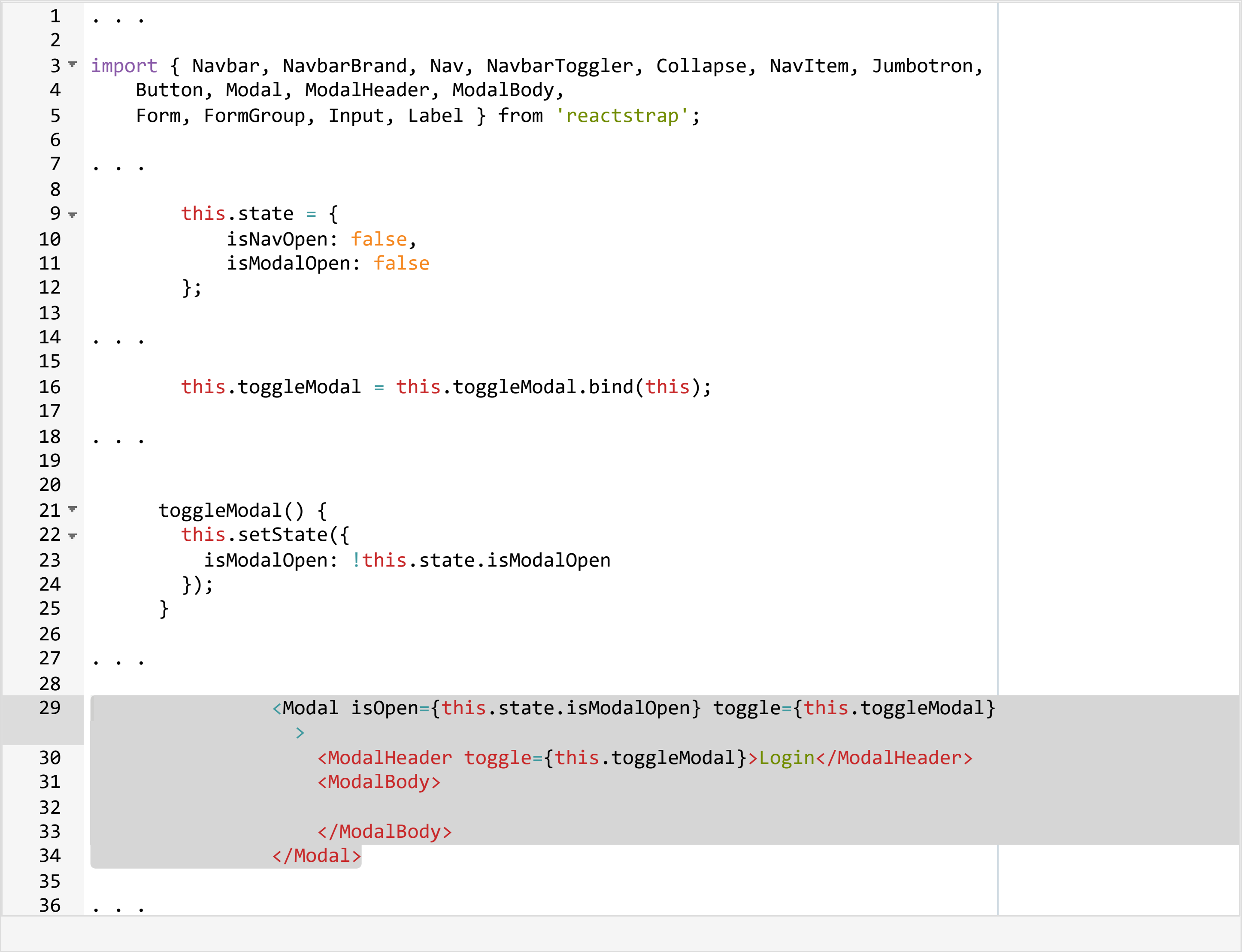
Exercise (Instructions): Uncontrolled Forms

In this exercise we will create an uncontrolled form within our React application using the uncontrolled component approach. At the end of this exercise you will be able to:

- Create uncontrolled forms through uncontrolled components in React
- Handle the form submission in the React application

Adding a Modal to Host the Form

- Update *HeaderComponent.js* as follows to add a new Modal to the application to host the form:



```
. . .

import { Navbar, NavbarBrand, Nav, NavbarToggler, Collapse, NavItem, Jumbotron,
    Button, Modal, ModalHeader, ModalBody,
    Form, FormGroup, Input, Label } from 'reactstrap';

. . .

        this.state = {
            isNavOpen: false,
            isModalOpen: false
        };

. . .

        this.toggleModal = this.toggleModal.bind(this);

. . .


      toggleModal() {
        this.setState({
          isModalOpen: !this.state.isModalOpen
        });
      }

. . .

                <Modal isOpen={this.state.isModalOpen} toggle={this.toggleModal}>
                    <ModalHeader toggle={this.toggleModal}>Login</ModalHeader>
                    <ModalBody>

                    </ModalBody>
                </Modal>

. . .
```

- Then, add a button to the Navbar to enable toggling the modal:

```
. . .

                            <Nav className="ml-auto" navbar>
                                <NavItem>
                                    <Button outline onClick={this.toggleModal}><span className="fa fa-sign-in fa-lg"></span> Login</Button>
                                </NavItem>
                            </Nav>

. . .
```

Adding the Uncontrolled Form

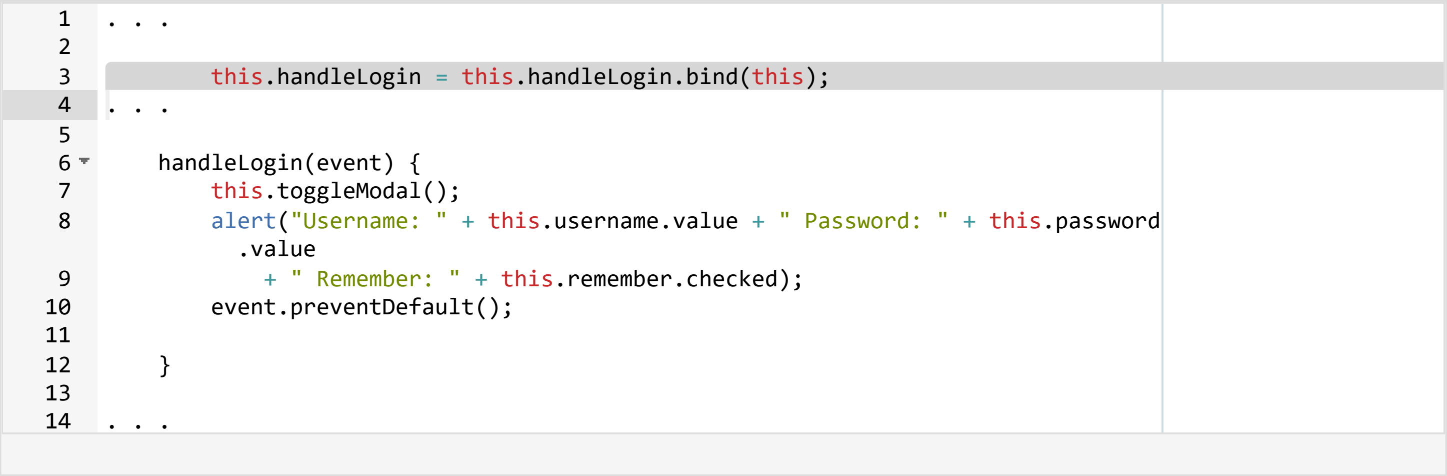
- Add the form to the modal body as shown below:

```
. . .

                        <Form onSubmit={this.handleLogin}>
                            <FormGroup>
                                <Label htmlFor="username">Username</Label>
                                <Input type="text" id="username" name="username"
                                    innerRef={(input) => this.username = input} />
                            </FormGroup>
                            <FormGroup>
                                <Label htmlFor="password">Password</Label>
                                <Input type="password" id="password" name="password"
                                    innerRef={(input) => this.password = input}  />
                            </FormGroup>
                            <FormGroup check>
                                <Label check>
                                    <Input type="checkbox" name="remember"
                                    innerRef={(input) => this.remember = input}  />
                                    Remember me
                                </Label>
                            </FormGroup>
                            <Button type="submit" value="submit" color="primary">Login</Button>
                        </Form>

. . .
```

- Then, add the following function to the class to handle the form submission:



```
. . .

        this.handleLogin = this.handleLogin.bind(this);
. . .

    handleLogin(event) {
        this.toggleModal();
        alert("Username: " + this.username.value + " Password: " + this.password.value
            + " Remember: " + this.remember.checked);
        event.preventDefault();

    }

. . .
```

- Save all the changes and do a Git commit with the message "Uncontrolled Form"

Conclusions

In this exercise we learnt to use an uncontrolled component approach to add a form to our React application.