

REACT JS

LECTURE 2

By Mona Soliman

AGENDA

- Integrate with Ui libraries
- Conditional Rendering
- React Hooks
- Handling Forms

UI LIBRARIES

- Bootstrap
- React Bootstrap
- Material Ui
- Semantic Ui
- Ant design



CONDITIONAL RENDERING

Conditional rendering in React works the same way conditions work in JavaScript. Use JavaScript operators like `if` or the conditional operator to create elements representing the current state, and let React update the UI to match them.

You can use ternary operator or `&&` to implement conditional rendering

REACT HOOKS

Hooks are a new addition in React 16.8. They let you use state and other React features without writing a class.

REACT HOOKS >> USESTATE

Returns a stateful value, and a function to update it.

During the initial render, the returned state (state) is the same as the value passed as the first argument (initialState).

The `setState` function is used to update the state. It accepts a new state value and enqueues a re-render of the component

```
const [state, setState] = useState(initialState);
```

```
  setState(newState);
```

REACT HOOKS >> USEEFFECT

Accepts a function that contains imperative, possibly effectful code.

```
useEffect(() => {
```

```
  return () => {
```

```
    //clean your code on unmount component}
```

```
  }, arrayDependencies);
```

`[]` : empty array means to fire on mount only

`[dependencies]` : means to fire on mount and with every change in this dependency value

FORMS

There are two ways to handle form:

1-Native Way

2-using library

ex: <https://formik.org/>

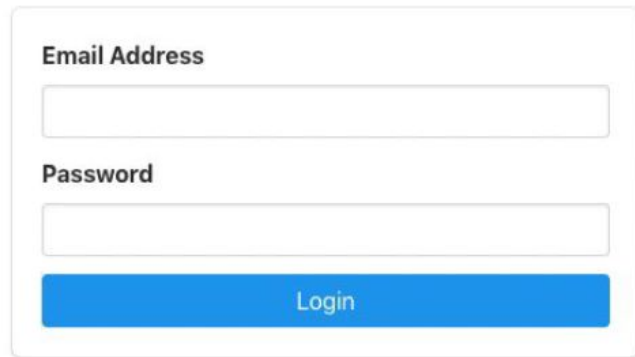
LAB

Create a TO DO react app contains of 2 pages
functional components with react hooks :

First step will contain login page with forms has fields :

- **Email address** - required - email format (REGEX) xxx@xxxx.com
- **Password** - required - password length not less than 8 characters
- show and hide password

You can use formik if you want in this form



The image shows a login form with a light gray border. It contains two text input fields, one for 'Email Address' and one for 'Password'. Below the password field is a blue button with the text 'Login' in white. The labels 'Email Address' and 'Password' are positioned above their respective input fields.

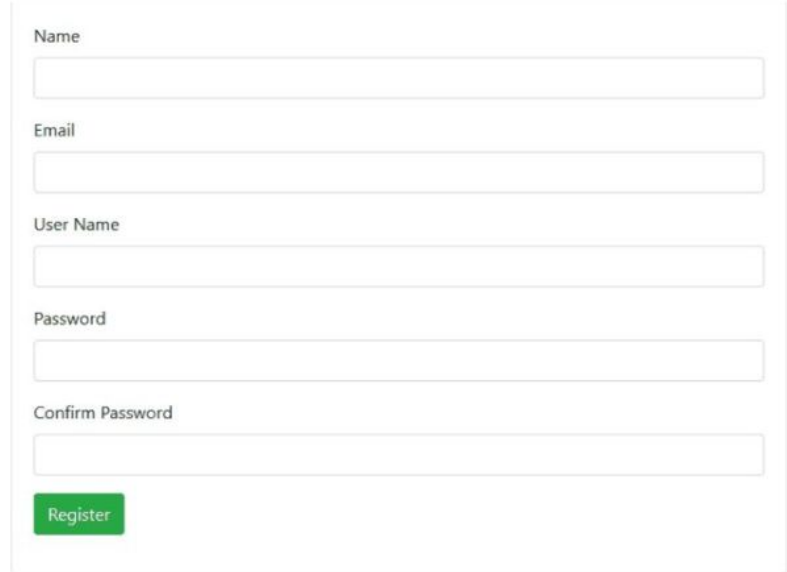
Email Address

Password

Login

Second page is register page with the following fields.

- Email address [required - email format]
- Name [required]
- Username [required - contains no spaces]
- Password [required - password length not less than 8 characters , contains at least one lowercase , one uppercase , at least one digit and special character [example : *@%\$#]]
- Example for valid Password : P@ssword1234
- Confirm password : [required - matches previous password]



The registration form consists of five input fields stacked vertically, each with a label above it: 'Name', 'Email', 'User Name', 'Password', and 'Confirm Password'. Below the 'Confirm Password' field is a green button with the text 'Register' in white.

Name

Email

User Name

Password

Confirm Password

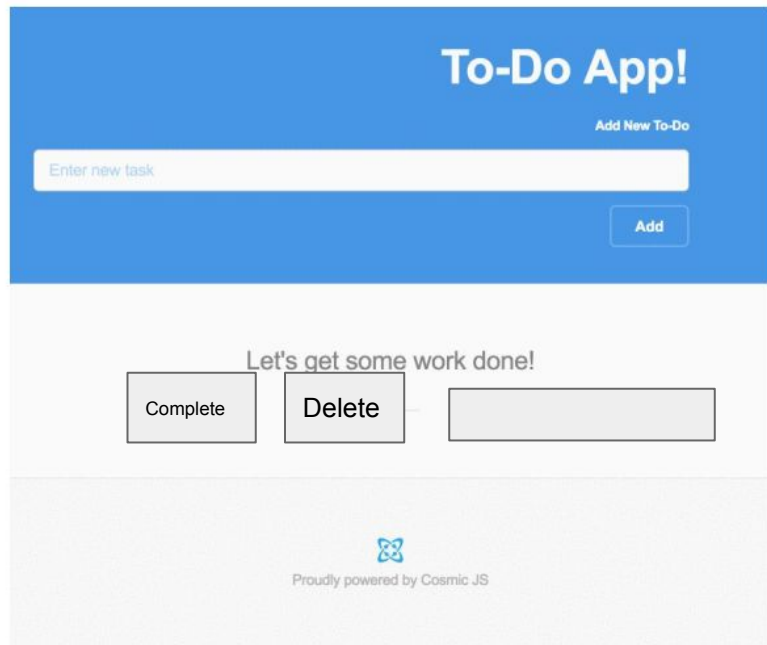
Register

Second step will contain To do form with a required input field with add button for tasks as shown in the image.

After adding task should be added in list with mark as done and delete buttons :

- Mark as done will add strikethrough on textc[Bonus]
- Delete will remove task from list

Using reusable components and passing data through components



THANK YOU