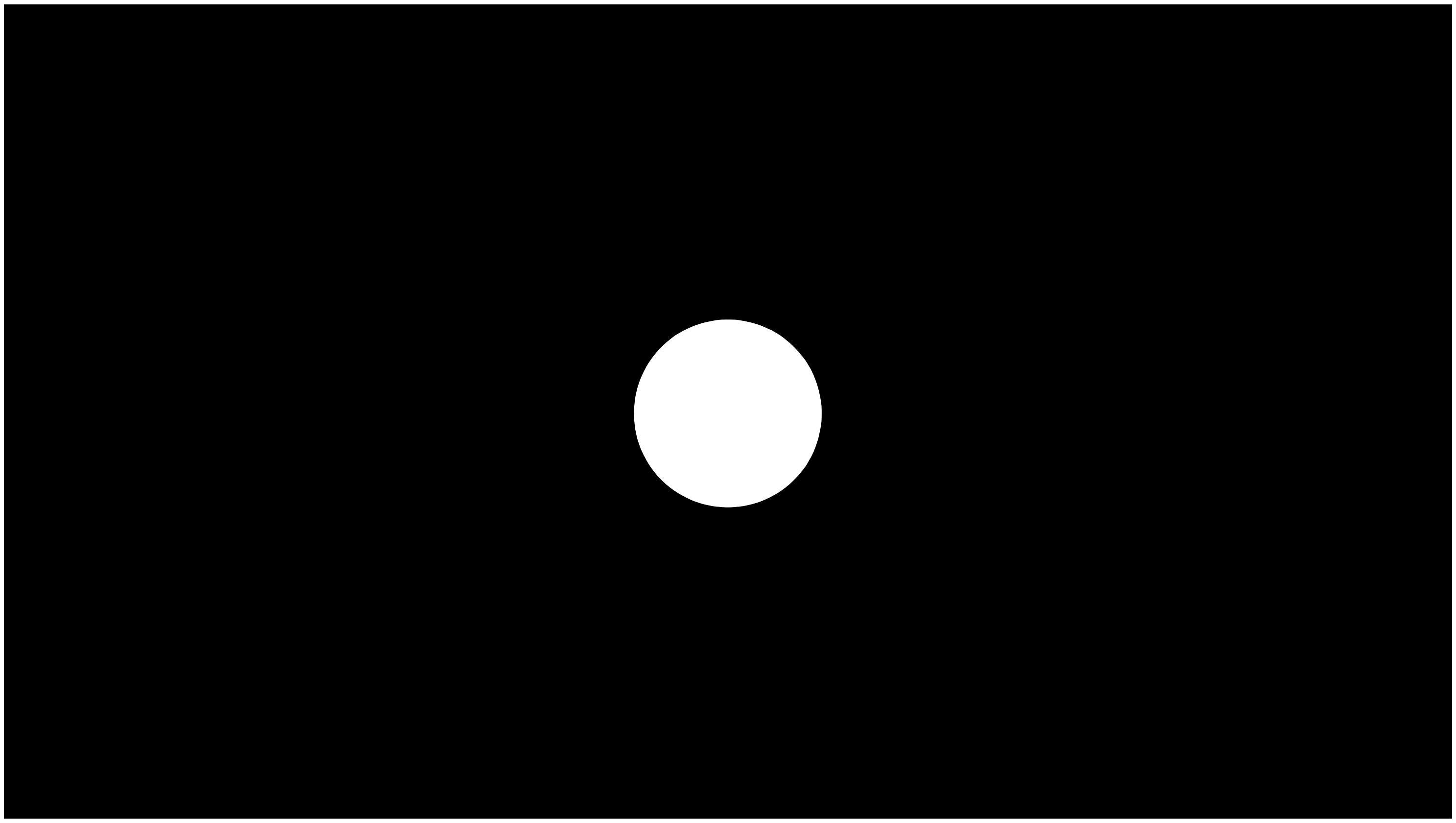
Complete React Developer in 2022 (w/ Redux, Hooks, GraphQL)



Course content Overview Q&A Notes Announcements Reviews Learr

Create a new note at 3:45

All lectures Sort by most recent

1:51 **3. React Basics** 23. Create React App - Everything Else

what is a component ?

A component is a self contained piece of code that returns some visual UI representatio of HTML UI being this that we can see and interact with.

3:51 **3. React Basics** 31. Monsters Rolodex - setState and Secondary Callback `this.setState()` is asynchronous in nature

3:21 **3. React Basics** 32. Monsters Rolodex - Mapping Arrays to Elements

```
    this.setState(
            ()=>{
              return {name:'farhin'};
            },
            ()=>{
              console.log(this.state.name);

            });
```

9:38 **3. React Basics** 36. Monsters Rolodex - Lifecycle Method: componentDidMount

`componentDidMount() method`

Whenever we have a component , a class component specifically that needs to leverage
order to get data that it needs in order to display the appropriate UI, we need to put tha
componentDidMount() lifecycle method

4:33 **3. React Basics** 40. Monsters Rolodex - Searching & Filtering

advice :

never try to modify the existing array , always use generate new array like using filter me

1:31 **3. React Basics** 44. Monsters Rolodex - Understanding Components

react is a way to tie togethe the visual reporesntation of the UI and the functional repres

Component :

it aims to do tie togethe reusable portions of the code together into one segment

8:19 **3. React Basics** 48. Monsters Rolodex - SearchBox Component

Note : Whenever you change pass prop , if you are passing a string then no need of { } , e
need to use swigly brackets else it won't register

9:16 **3. React Basics** 49. Monsters Rolodex - CSS in React

NOTE: css of one component is applicable to the components of the entire page unless
isolation

1:06 **3. React Basics** 50. Monsters Rolodex - CardList Component

This below is called **string interpolation**

In jsx **correct format** :

```
<img  alt={`monster ${monster.id}`} />
```

Incorrect format :

```
<img  alt=`monster ${monster.id}` />
```

0:00 **3. React Basics** 52. Monsters Rolodex - Finishing Touches

Just a quick note to remember to use the back tick `` ` `` and NOT regular single or double
interpolation on our image `src`. This is an easy mistake to make as you cannot interpola
regular quotes, you must use back ticks!

4:56 **3. React Basics** 56. Pure & Impure Functions

Pure and Impure Functions

A pure function is a function whose return value is solely dependent on the arguments p
not on any external variables

```
let c= 5;
const impurefunc = (a, b) =>{
  return a  + b + c;
}
```

A pure function should also not create **side effects**

- when an external variables is changed through a function then this is called side effect

4:11 **3. React Basics** 56. Pure & Impure Functions

A side effect is when a function creates some kind of effect outside of its scope.

And in this particular case, it's setting some variable outside of the scope of its function.

And the scope is whatever is inside of this function.

The variable C does not live within the scope of this function.

It's accessible within the scope of this function, but it is also accessible outside of the sc

this function.

And as a result, if it were to change that variable, that is considered a side effect.

4:39 **3. React Basics** 57. Monsters Rolodex - Hooks: useState

To use useState() function:

```
[searchvalue , setSearchField] = useState('');   // here '' is the inti
const onSearchChange = (event)=>{
    const searchFieldString = event.target.value.toLocaleLowerCase();
    setSearchField(searchFieldString);
}
```

For every state variable we need to use a new useState() functions , as every useState is

2:54 **3. React Basics** 57. Monsters Rolodex - Hooks: useState

useState()

this gives us back an array of two values .

The first values if going to be the value that we want to store and teh second is going to

Difference between setState and useState is that setState is a object in the class compo

But in the functional component it's the individual values

6:44 **3. React Basics** 58. Monsters Rolodex - Functional Component Re-rendering

Note : **Strict mode** will render your components twice , disabling strict mode in react wi

4:13 **3. React Basics** 58. Monsters Rolodex - Functional Component Re-rendering

In functional component when the state value changes then only the whole component i

Even if the state values remains same after we submit still there wont be a re render trig

7:32 **3. React Basics** 59. Monsters Rolodex - Infinite Re-rendering

`useEffect(callback,[])` can be used to manage the infinite re-rendering cycle.

callback : callback is going to be the code or the effect that we want to happen inside of

[ ] = this contains different dependencies , these are likely to be the state values. or ma
passed to functional components.

3:57 **3. React Basics** 59. Monsters Rolodex - Infinite Re-rendering

Good note here :

Infinite re render caused by fetch as the data object received earlier has different referen

3:09 **3. React Basics** 60. Monsters Rolodex - Hooks: useEffect

IMPORTANT ABOUT useEFFECT Function:

Note : In the useEffect argument the second argument tells the useEffect function whe function passed as the first argument.

If we pass [] as second argument , it tells the useEffect function to run only once as ther which it must keep track on.

5:20 **3. React Basics** 60. Monsters Rolodex - Hooks: useEffect

use `useEffect()` to cope with the issue of infinite re rendering with functional compone

1:06 **3. React Basics** 61. Monsters Rolodex - Remaining Components

For functional component we pass props like the below

```
const  CardList = (props , forwardRef) =>{};
```

the forwardRef parameter won't be used mostly , asit's used around 3-7 % of the time

2:32 **3. React Basics** 63. React v18: Strict Mode Changes

Strict Mode :

React strict mode renders everything twice so as to ensure there is no error .

5:31 **4. Capstone Project: Intro + Setup** 78. Category Item Component

key = {x.id} should be always set within map function

```
monsters.map((x) = > (
<Card key = {x.id} props = {prop} />
));
```

8:09 **4. Capstone Project: Intro + Setup** 80. Adding Fonts

Lock Files:

These lock files are essentially generated by our package managers , npm or yarn respe
lock file is a way for us to make sure that all of the libraries that we use as a team are loc
Note: people using yarn will have package.lock file as well as yarn.lock file

5:47 **5. Routing + React-Router** 82. Updating/Upgrading Libraries

react router version 6 requires react v 16.8 or greater.

6:21 **5. Routing + React-Router** 84. React Router Outlet

`<Outlet/>` helps us dynamically change our code output

2:32 **5. Routing + React-Router** 86. React Router Link

`React.Fragment` renders to nothing when it's rendered into the DOM
`<></>` is a fragment is we want to manually write :

4:08 **6. Authentication + Firebase** 88. Setting Up Firebase

`yarn add firebase`

3:58 **6. Authentication + Firebase** 92. Authenticating With Firebase

firebase SDK : A developer kit

0:52 **6. Authentication + Firebase** 92. Authenticating With Firebase always store backend database related files in seperate folder called **utils**

11:48 **6. Authentication + Firebase** 94. Setting Up User Documents

firebase Concept :

doc() method creates a connection with the firebase database .

getDoc() method returns the document object from the firebase database.

2:14 **6. Authentication + Firebase** 94. Setting Up User Documents

Firestore is a different service so we need to

```
import {
getFirestore,
doc, // retreives docuemnts from firestore data base,
getDoc, // get document data
setDoc
} from 'firesbase/firestore'
```

7:57 **6. Authentication + Firebase** 96. Sign In With Redirect

Firebase `auth` keeps track of all the authentication states that are happening internall

```
import getAuth from 'firebase/auth';
export const auth = getAuth();
```

This auth is helpful in case we use googleSignInWithRedirect

5:06 **6. Authentication + Firebase** 96. Sign In With Redirect

With SignInWithRedirct we need to use useEffect or useState as signinwithredirect re-re
the site lose the data

14:15 **6. Authentication + Firebase** 97. Sign Up Form Pt.1

Note about dynamic update of objects

3:46 **6. Authentication + Firebase** 98. Sign Up Form Pt.2

imporant thing said

2:29 **6. Authentication + Firebase** 98. Sign Up Form Pt.2

Firebase concept:
Native providers in firebase authentication doesn't require any Provider.

so email authentication with user and pass doesn't require any providers and we can dir

12:22 **6. Authentication + Firebase** 100. Generalizing Form Input Component

better approach to pass input attributes

10:06 **6. Authentication + Firebase** 100. Generalizing Form Input Component

~ symbol in scss looks for the nearest sibling with given class name .

```
$main-color :black;
@mixin shrinkLabel{
top: -14px;
font-size:12px;
color : $main-color
}
For example :
&:focus ~ .form-input-label{
@include shrinkLabel();
}
```

10:49 **6. Authentication + Firebase** 100. Generalizing Form Input Component

scss & and ~ explanation

8:02 **6. Authentication + Firebase** 100. Generalizing Form Input Component

method to add class name :

```
className  = {`${otherProps.value.length?'shrink':''} form-input-label`}
```

2:37 **7. React Context For State Management** 105. User Context

`useContext()` : whenever the values inside the context updates, re-render of page hap

`useContext(<ContextName>)` returns the value for the context

5:10 **7. React Context For State Management** 106. Re-rendering From Context

IMPORTANT NOTE WHILE USING useCONTEXT.

One drawback of using context is that , suppose you use useContext() in the sign up con
updates in the sign in component , because we have useContext() in the signup compon
keep re-rendering causing performance issues if there are alot of functions within the s
Although this function calls will cause performance issues but if you try paint flashing te
there is no re-render of the signup component. This is because there is no changes in th
of virtual dom).

English

Get the app

About us

Help and Support

Help and Support

Terms

Privacy policy

Sitemap

Accessibility statement

business